# *Repeated Goofspiel*: A Game of Pure Strategy

Moshe Dror and Graham Kendall, *Senior Member, IEEE*

***Abstract*—In this paper, we examine a pure strategy game known as *Goofspiel* and report on the results of round-robin competitions between 14 programs designed to play this game. *Goofspiel* is a two-person card game that is easy to play. However, playing this game successfully has proven to be a difficult task. There is no known "good" strategy for *Goofspiel*. This is the first time that playing *Goofspiel* has been examined in a context of a round-robin competition between programs. None of the participating programs won consistently against its rivals. Thus, no clear dominating strategy of play has emerged. In this respect, *Goofspiel* is similar to the *Prisoner's Dilemma* where *Tit-for-Tat* has proven to be a good strategy against many but not against all. This paper introduces *Repeated Goofspiel* and presents preliminary experimental results. We hope it will motivate further research into this fascinating game.**

***Index Terms*—*Goofspiel*, strategy.**

## I. Introduction

In THIS paper, we examine a game called *Goofspiel*—a two-person card game introduced by Merrill Flood [1], [2] in his student days at the Mathematics Department, Princeton University, Princeton, NJ, USA, in the 1930s. At the time, Merrill Flood was known to be a very good card player. Perhaps this the reason why it is difficult to analyze *Goofspiel* for a "winning" strategy. Thus, one has to resort to experimentation. We report here on computational experiments, with a set of 14 computer programs designed to compete against each other in round-robin tournaments with a varying number of rounds in each. The participating programs were written by computer science or management information systems doctoral students or faculty members. No two programs have been written by the same individual. To our knowledge, it is the first time that such a *Goofspiel* competition has been proposed and presented in the scientific literature. The software framework is described in the Appendix.

An earlier, well-known, competition in search of *good* strategies in two-person games is Axlerod's *(Iterated) Prisoner's Dilemma (IPD)* competition [3], which was repeated in 2007 [4]. *Repeated Goofspiel* is a game that is far more complex than the *Prisoner's Dilemma*.

*IPD* is probably the most famous two-person strategy game but there are a number of others that have received significant research attention, and continue to do so. Earlier work on *RoShambo (Rock/Paper Scissors)* was reported in [5] and [6], with a recent book giving more up to date information [7]. This game now has its own world championship. More recently, *Spoof* (a game where you have to guess the sum of the number of concealed coins held by all players) has received attention from the evolutionary computation community [8], [9]. A more recent introduction into the scientific literature is the *Lemonade Stand Game* [10], which is worthy of further investigation.

Manuscript received September 09, 2012; revised January 29, 2013; accepted March 12, 2013. Date of publication April 12, 2013; date of current version December 11, 2013.

M. Dror is with the University of Arizona, Tucson, AZ 85004 USA (e-mail: mdror@eller.arizona.edu).

G. Kendall is with the ASAP Research Group, School of Computer Science, University of Nottingham, Nottingham NG7 2NR, U.K. and the University of Nottingham Malaysia Campus, Selangor, Malaysia (e-mail: graham.kendall@nottingham.ac.uk; graham.kendall@nottingham.edu.my).

Digital Object Identifier 10.1109/TCIAIG.2013.2257773

## II. *Goofspiel*

*Goofspiel* is a two-person card game. Though simple to describe, it has wider implications and lessons for competitive positioning of limited resources. The rules of the game are simple but the mathematical analysis is not. In standard *Goofspiel* there are two players and three full suits of 13 cards from a regular deck (one randomly selected suit is discarded for the purposes of this game).

One of the suits (e.g., the 13 hearts) is given to player I. Another suit (say, the 13 diamonds) is given to player II. The remaining suit (say spades) is shuffled and placed face down between the players. In this case, the 13 club cards are not used and play no role in the game.

At the start of the game, the top spade card is turned face up in full view of both players. Each of the two players chooses one of his cards and presents them simultaneously. The player who presents the higher value card wins the face up spade card. The winning player retains the spade card, and the other two cards are discarded from the game.

If both players present cards of the same value (a tie), the three exposed cards are discarded and in that turn no player wins any points. A new top spade is exposed, and the game is repeated until no cards are left. This represents one round, and the players can play as many rounds as they decide at the start of the game.

The player with the highest total value of spade cards is the winner of a round, and the winning margin is the difference between the two players' total values. In a *Repeated Goofspiel* game, the player with a higher total value at the end of the game (after the final round) is the winner by the corresponding winning margin. *Goofspiel* is a zero-sum game. In this context, there are a set number of points (91) to be won in each game. Some of these points are won by one of the players in each round, unless the players show the same cards, in which case, the points are *wasted*. At the end of each game, the total points shared by the players, added to any wasted points, sum to the 91 points available.

In ranking the cards, Ace is considered the lowest (with a score of one) and King is the highest (with a score of 13). The other cards take on the "usual" values, that is, $A = 1, 2 = 2, \ldots, 10 = 10, J = 11, Q = 12, K = 13$.

*Goofspiel* is related to an important family of two-person zero-sum multistage games called games with finite resources [11]. However, it is not the matrix game described in [11].

Adopting a more recent notation of games with finite resources from [12], a game with finite resources is described as follows. Player I has a vector of resources $A = \{a_1, \ldots, a_n\} \in \mathbf{R}^n$, and player II has a vector of resources $B = \{b_1, \ldots, b_n\} \in \mathbf{R}^n$. We are also given an $n \times n$ payoff matrix $M = (m(a_i, b_j))$ associated with $A$ and $B$. The payoff matrix is assumed to be known to both players. In stage $k, (k = 1, \ldots, n)$, players I and II simultaneously select $a_k \in A \setminus \{a_1, \ldots, a_{k-1}\}$ and, respectively, $b_k \in B \setminus \{b_1, \ldots, b_{k-1}\}$ (a row and a column, respectively) and realize a row and a column payoff of $m_r(a_k, b_k)$ and $m_c(a_k, b_k)$, respectively $(a_0 = b_0 = \emptyset)$. It is assumed that the players know (can observe) what resources were used up to stage $k$—the remaining payoff matrix can be observed. At the last stage, the players use their remaining resources $(a_n, b_n)$ and obtain the corresponding payoffs $m_r(a_n, b_n)$ and $m_c(a_n, b_n)$. The total payoffs for the row and column players are the corresponding sums of stage payoffs—$\sum_{k=1}^{n} m_r(a_k, b_k)$ and $\sum_{k=1}^{n} m_c(a_k, b_k)$. Note that we introduce some abuse of notation. The payoff $m(a_k, b_k)$ does not correspond to the $(k, k)$ element of $M$ but to the elements of $A$ and $B$ of the $k$th stage (row and column) selection denoted here as $a_k$ and $b_k$.

Gale [11] proved that the value of the game with finite resources is $V = 1/n \sum_{i=1}^{n} \sum_{j=1}^{n} m(i, j)$ and the optimal strategy for player I is to select a permutation $(a_1, \ldots, a_n)$ of $\{1, \ldots, n\}$ at random with probability $1/n!$ each, and to use resource $a_k$ at stage $k$. Similarly for player II. Gale's result has been extended by [12].

In contrast to games with finite resources, in *Goofspiel*, the payoffs are revealed at each game turn, one at a time before the next play. There is no static payoff matrix like the matrix in the above game with finite resources since the pair (combination) of the cards selected in a play turn is stochastic with respect to the payoff value. *Goofspiel* has been described in [13] and studied in [14]. However, it dates back to the early 1930s [2] as testified by Tucker in his transcribed version of an interview published in "The Princeton mathematics community in the 30's," The Trustees of Princeton University, 1985, where *Goofspiel* is said to be a 1930s invention of Merrill Flood.

*Repeated Goofspiel* is a difficult game to analyze and presently there are no known optimal winning strategies (i.e., the best move to make at any point, against *any* given strategy) against an opponent using an arbitrary strategy. The optimal strategy (if it exists) must include randomization, since any deterministic strategy can easily be defeated for a sufficiently large number of rounds and payoff values that are not too skewed.

To demonstrate the computational challenge of attempting to construct "good" strategies for *Goofspiel* we note that just the number of pure strategies for each player is

$$n^n \times \Pi_{k=1}^{n-1} k^{k(k+1)}.$$

That is, for $n = 4$ card *Goofspiel*, the number of pure strategies for a single player is 8.4 billion (an 8.4 billion by 8.4 billion game). For $n = 5$, the number of pure strategies goes up to about $10^{23}$ [14].

A few results for restricted play versions of *Goofspiel* are known [14]: if player I knows that player II plays his cards in a random fashion, to maximize his/her expected payoff in a *Repeated Goofspiel*, player I should match the upturned value of the spade card with their own card. For a simple proof of this result see [15]. We call this *the upcard-matching strategy*.

## III. Learning in Games in Relation to *Goofspiel*

Consider two players and a finite stage game with full information disclosure, like *Goofspiel*, that is played in an infinite (very long) sequence of repeated rounds. In such repeated games, in order to anticipate the opponent's projected "moves," the players learn their opponent's perception of the game as the game progresses by forming beliefs about how their opponent appears to view the game based on his/her previous "moves"—the history of the game. The question raised in numerous papers on the topic of learning to play strategies concerns the existence of learning strategies that guarantee, or come close, to equilibrium play in repeated games without assuming any prior knowledge of the opponent's strategies (see [16]–[18] and their references). The process of learning your opponent's strategies and responding to them has been examined in the scientific literature for some time. Brown [19] proposed a "fictitious play" learning and response process that converges to the Nash equilibrium set after a sufficient number of plays for certain classes of games, such as zero-sum games, dominance solvable games, games with strategic complementarities and diminishing returns, and potential games. However, even though *Goofspiel* is a zero-sum game, it does not seem to fit into the "fictitious play" learning classes because of the randomness of the payoff in each turn and the nonstationarity (that is, a player does not have to select a strategy and play the same strategy throughout the game—he/she could choose a different strategy on every play) of the opponent's strategies. A representative work on convergence in Bayesian learning leading to convergence to Nash equilibrium is the work of Kalai and Lehrer [20]. Since no pure strategy is likely to consistently outperform general strategies in *Goofspiel*, one ought to examine learning mixed strategies (see [21] for an analysis). However, the lack of plausible stationarity in an opponent's strategies in *Goofspiel* derails the known classic approaches in the literature on learning in games. In game theory, the existence of pure strategy Nash equilibrium in two-person zero-sum games is claimed by referring to Zermelo's theorem [22]. However, for *Goofspiel*, Zermelo's theorem does not apply because of the uncertainty introduced by the simultaneous card selection of both players.

Since no one plays an infinite sequence of rounds of any kind of game, the challenge is to play a game successfully in a finite number, not necessarily known *a priori*, of game repetitions—rounds. Given a finite number of game rounds played against the same opponent, what should a player's strategy be? That is, what should be the plan for learning your opponent's strategies and your subsequent actions that will capitalize on that knowledge by improving the likelihood of winning more points than your opponent over a finite number of games, say 50. In *Repeated Goofspiel* you would want to update your game strategy after each one of the first 11 cards played in a game. It is easy to determine an optimal strategy for the last two card plays.

One perspective based on experimental findings (see [23], [24], and their references) is that in strategic settings without clear precedences, players' initial responses often deviate systematically from equilibrium and the so-called "level-$k$ thinking" can outpredict equilibrium. Level-$k$ thinking assumes that players are heterogeneous in their strategic sophistication. For instance, level-0 players choose at random across all strategies. Level-1 players assume that all the other players are level-0 and select the best response strategy, etc.

In *Goofspiel*, a level-0 player chooses the card to play at random. An optimizing strategy against such a random play is to play a card matching the upturned spade. Thus, the spade matching players are level-1. How does the level-$k$ thinking translate to a repeated game scenario where the players observe their opponents past plays and are allowed to modify their strategic behavior at each/current move? This is not at all clear in the context of playing *Goofspiel* and, in fact, a static heterogeneous strategic sophistication performs poorly (see Section VI).

There is only scarce scientific literature on devising strategies for playing *Repeated Goofspiel*. In their working paper, Chan and Craft [25] describe a limited experimentation with a computer program for learning card playing strategies, and applying their program to *Goofspiel*. They stipulate three strategies: level-0—random, level-1—matching, and good ("G"). Their "G" strategy contains both randomized and deterministic strategies. The details of the "G" strategy are not provided. The "G" strategy selects a card by evaluating the current state of play and making a decision whether to play aggressively, while considering the value of the upturned card. The card selection algorithm for "G" makes a probabilistic decision for some states of system parameters and a deterministic selection for others as the parameters are updated after each play. Playing "G" against level-0 for 100 game rounds resulted in "G" winning 95 games, losing four games, and tying one game. Playing "G" versus level-1 resulted in "G" winning 74 games, losing 24 games, and tying two games. With level-1 against level-0, level-1 won 96 games and lost four games. No total points won by the strategies are provided.

## IV. Experimental Framework

In this section, we provide a description of the competition structure and the computer programs in our *Repeated Goofspiel* competition. The participating programs could also play against human opponents, but this option was only used for debugging purposes. The programs were submitted by computer science or management information systems doctoral students from the University of Arizona, Arizona, Tucson, AZ, USA and the University of Nottingham, Nottingham, U.K., and one computer science faculty. The round-robin competitions were held with 10, 20, 30, and 50 repeated games (rounds) against each opponent in each competition.

We maintain a record of the total points won by each player for each set of repeated games. We feel that maintaining a total points tally is fairer than simply recording the number of wins and losses, as this is a more fine grained way of assessing the player. We note that, in this respect, our competition setup is similar to many *IPD* competitions [3], [4], which also records the points won (or lost) rather than the number of individual games won/lost.

Competitions with an increasing number of rounds aim to assess the learning "speed" and subsequent strategy adjustments of the players—the programs. In addition, we paired the two most successful programs in a competition comprising 100 rounds.

The competition was conducted in two stages, three years apart. This was due to the low number of entries in the first competition. In the first stage, we offered monetary rewards and were able to solicit only five participants. We added eight more competitors three years later as a result of it being a compulsory requirement of a Ph.D. course at the University of Arizona. For completeness, we begin with a description of the dollar incentive in the first round-robin *Repeated Goofspiel* competition with five players.

The dollar reward structure was set as follows. The players were told that there is a fixed prize fund of $500 to be divided based on their performance in proportion to their total winning values. It is not a zero-sum game setting—the players do not risk any monetary losses. It is our understanding that the $500 prize did not constitute a significant enticement for the participating players. Our perception was that the primary incentive for the players was to perform well and to win the competition—gaining prestige for being the person with the highest score.

The eight Ph.D. students who submitted their computer programs for the competition three years later (December 2011) were told that their final grade in a Ph.D. course might be affected by the performance of their entries. Their participation and success in the *Repeated Goofspiel* competition would constitute a fraction of their final grade. Again, the primary incentive was to perform well and try to win the competition.

## V. Description of the Players' Strategies

In this section, we present an edited version of the players' strategies, based on their own writeup, that describes the logic of their algorithm for selecting play moves. Some of the players provided more detail than others. Subsequently, the card selection (playing) strategies below are not presented with the same level of detail. Ideally, we would ask the strategy authors to refine their descriptions, and provide more detail, but they have now all graduated and left the institutions, so it is not possible.

### A. Strategy 0101

This strategy attempts to detect the opponent's static strategy and counter it with a best response strategy. It assumes only two options for the opponent's strategy. The opponent's strategy is analyzed by computing the Pearson correlation between the indices of the cards played by the opponent and the upcard indices. If the opponent tends to play high-value cards for high-value upcards (and correspondingly low-value cards for low-value upcards), i.e., the Pearson correlation is higher than a certain threshold-value (0.2), the strategy is labeled as "nonrandom," otherwise it is labeled as "random." As long as the opponent's strategy is labeled as a "random" strategy, 0101 will play a card matching the value of the upcard. Otherwise, if the opponent's cards have a high a correlation with the upcards, 0101 strategy

will play cards with a slightly higher value for the upcards with values greater-or-equal than seven (case 1), and cards with low random values for upcards with values less than seven (case 2). This is done by playing the lowest unused card with a value greater than the upcard value in case 1 or, if that cannot be done, the highest unused card with a value smaller than the upcard value. For upcards with values less than seven, 0101 will play cards with a lower value than the upcard and, if that cannot be done, the lowest unused card with the highest value.

### B. Strategy 0102

This strategy attempts to identify and exploit the opponent's play by using a simple static rule-based mechanism.

The opponent's strategies are classified into four categories. Category 1 plays the same card as the upcard on every move; category 2 plays the card one point higher than the upcard; category 3 plays the card two points higher than the upcard; and category 4 contains all strategies that are not contained in other categories.

0102 identifies the opponent's strategy in the first round and assigns it into one of the four categories, treating it as a fixed strategy. If the opponent's strategy is labeled as category 1, 0102 plays the card one point higher than the upcard on every move. If the opponent's strategy is labeled as category 2, 0102 plays the card two points higher than the upcard on every move. If the opponent's strategy is labeled as category 3, 0102 plays the card three points higher than the upcard on every move. If the opponent's strategy is labeled as category 4, 0102 plays the card equivalent to the upcard on every move. If 0102 wins less than 30 points in any round after round 1, it switches to a random strategy.

### C. Strategy 0103

This strategy has three main components. First, 0103 attempts to detect obvious patterns, such as an opponent playing the same card in response to a certain *upcard*, or cards that an opponent always uses in a certain order. Second, 0103 follows a preset strategy when failing to detect the opponent's playing pattern. Finally, when there are two cards left in each hand, 0103 computes, and plays, the best response.

*1) 0103 Strategy Description:* Note that one can win any round of *Goofspiel* by winning just a few high-value cards. 0103 initially selects a subset of cards "to win" and plays a "high-value" card when each of these cards appears. For the other cards, it selects randomly from its non-"high-value" cards.

In each subsequent round, the game plan is modified when an opponent's pattern is detected. Table I shows the strategy of 0103.

All the results from previously played cards are recorded. These include: what upcards have been used, what opponent cards have been used, how many times a certain card has been used in response to a certain upcard, etc.

If the opponent's pattern is detected, the game plan will be updated. 0103 responds with a card according to the modified game plan. In 0103, there are five characteristics of cards that the strategy handles:

1) an opponent usually plays a card that is $x > 0$ points above an upcard;
2) an opponent usually plays cards in a certain order;
3) an opponent usually plays a certain card in response to a certain upcard;
4) our highest card is strictly better than the opponent's highest card, and the player is in a position to win the highest upcard; therefore, the game plan will be modified to play the highest card when the highest upcard is shown;
5) the last two plays for the round. When there are two cards left in the round, 0103 computes and plays the best response strategy.

TABLE I
STRATEGY FOR 0103

| Upcards | Rules for assigning cards from the player's hand |
|---|---|
| 12 ,13 | Assign randomly to one of the two upcards the 13 value card.<br>To the other upcard assign 8. |
| 8, 9, 10, 11 | Assign any three of these 4 upcards 12, 11, 10, in relation<br>to their values.<br>Assign 7 the remaining upcard. |
| 6, 7 | If upcard 12 is assigned 13, then upcard 7 is assigned 9.<br>Otherwise, assign 9 to any one of these two upcards and assign<br>the other upcard at random a value between 1 to 6. |
| 1, 2, 3, 4, 5 | Assign randomly the remaining cards. |

### D. Strategy 0104

The 0104 strategy combines random and match upcard strategies. It uses the ideas of *interval* and *balance*.

The *interval* determines, based on the value of the upcard, the range of the cards to play in each turn. *Balance* determines the midpoint of the return value. The average value of all possible return values is equal to upcard+balance.

The card 0104 plays is a random number in the range of upcard and interval+balance $-1$ modulo(13) of the available cards.

If 0104 loses too many points in a row, then the strategy changes by altering the *balance* and the *interval*. The player only changes the strategy after 30 cards have been played since the last strategy change.

### E. Strategy 0105

This strategy is a combination of the following three ideas.

1) Order the current upcard (the presently exposed spade) among the spade cards not yet played—the face down spades. Then, select the card in hand that meets the position of the upcard. Thus, the higher cards will be matched with the higher upcards.
2) Average the opponent's previous bids on the same upcard, and bid according to that average.
3) Bid according to the upcard's value.

The three simple strategies are selected at random in each turn—mixed strategy with 1/3 likelihood. After strategy selection, a bid is chosen by random card selection based on a normal distribution centering on the bid intention.

0105 also checks the current game status. If it finds that it has lost four bids or 30 points out of five bids then it will assign a random shift to the current bid intention.

### F. Strategy 0106

0106 is based on the assumption that, in every turn, the opponent will attempt to anticipate the next card to be played. Thus, the 0106 play strategy involves randomization in every aspect of selection of the card to be played. In addition, 0106 selects a goal for a score ranging between 35 and 50 in every round and determines the bids corresponding to achieving that goal.

The different card combinations that add up to the selected goal score are selected at random. Then, a list is created that contains the set of cards to bid for each card in the selected card combination. Every card selection is randomized over the remaining cards.

### G. Strategy 0107

0107 is based on its belief that its opponent's play is based on past play. The key to this strategy is the computation of a "probability matrix" for the prediction of the cards most likely to be played. It uses weights that are biased toward the more recently played rounds. Once the opponent's most likely card is determined, 0107 uses card $+1$ (or $+2$, $+3$, if still available) to respond. As a default option a random card is selected. The strategy assumes that the opponent uses a static strategy and essentially responds with its own static strategy.

### H. Strategy 0108

0108 attempts to predict the opponent's play for each upcard. However, recognizing that playing a value greater than the predicted opponent's play might fail when playing for high value cards, 0108 attempts to "neutralize" the high value cards and to focus on winning the seven low value cards by playing value $+3$. In addition, to limit the opponent's learning ability, 0108 uses randomization when playing for the cards it does not target to win. The assumption is that the opponent will mainly play a card value equal to the exposed spade card value $+x$ strategy and will attempt to learn/anticipate 0108's strategy.

### I. Strategy 0109

0109 is based on the assumption that one can predict/discover/learn the opponent's strategy and then apply a set of static (+random) strategies against the "discovered" opponent's strategy. The strategy learning process is quite involved and based on tabulating all past plays, aided by threshold calculations that trigger perceived strategy detection. Essentially, this strategy is based on the fact that if one can successfully predict the next card that an opponent is to play, one can choose a card that has a value one above that of the opponent's card. So the strategy is built around *mining* the information encrypted in the game's history. Cards played by the opponent are tabulated by the upcard and the number of times the opponent played one of his remaining cards in the present game setting. This leads to an assignment of a likelihood to each card in the opponent's hand. If none of these likelihood values is greater than a preselected threshold value (say, 0.35), but still higher than another preselected threshold values (say, 0.20), then a play card is selected at random. If none of these likelihood values is greater than the second threshold, say 0.20, then it is assumed that the opponent is selecting his cards at random, in which case a matching strategy is used. In case the "right" card is not available, then the lowest card is played. The first two rounds are played is reverse order to a card matching strategy. In addition, the play history is weighted by a scheme that resembles exponential smoothing and the threshold values are periodically adjusted.

### J. Strategy 0110

0110 is built on a set of four static strategies. The idea is to play the strategy that continues to win against the opponent's strategy. If 0110 loses two rounds in a row, then it activates the next strategy in this set of four. The idea is based on the notion that the most successful strategy against a given opponent ought to play most often. In the set of four strategies, strategy 1 is a static strategy designed to win against strategies of the type value $+x$, where $x$ is fixed. Strategy 2 does not directly take into account the up value card but only responds to the opponent's previous last play for the upcard. Strategy 3 is a static strategy that targets specific cards. Strategy 4 is activated in all cases by playing at random after a win is assured (points $>45$).

### K. Strategy 0111

0111's play algorithm is based on anticipating/predicting the opponent's play and responding "appropriately," via a set of static rules for each case. In the learning phase, 0111 focuses on learning the opponent's value $+x$ strategies for a fixed $x$ and detecting if the opponent is using a random strategy. Then, its strategy employs a simple defensive logic and a set of rules.

### L. Strategy 0112

0112 has three components: greedy, balanced, and random plays. Greedy focuses on wining the cards of value 10, and higher, by always playing the highest of the remaining cards. For the other cards, the play is based on selecting the cards as ordered by a dynamic ordering scheme generated by the *balanced* component. The balanced component selects the playing cards in proportion to the points of the upcard without considering the low valued cards. The random component selects a random card from the remaining cards to play. The three component strategies are used together in each round. No learning or prediction are incorporated in the 0112 algorithm.

### M. Strategy 0113

The 0113 algorithm is based on assigning probabilities to each opponent's remaining cards calculated from past plays and responding by playing a card of value $+1$ with equal probability. In cases where the card is no longer available, it assigns this probability to the closest higher card modulo 13. This is a fixed response probabilistic strategy that incorporates some learning.

### N. $K$-Level Thinking

In our initial experiment in 2009, we also implemented a set of 14 strategies that mimic decision rules for 14 fictitious players, each executing one of the $k$-level strategies. Level-0 strategy mimics a player who does not assume anything about the opponent's strategy and merely chooses to play his next card according to some probability distribution, say, uniform (i.e., a random player). Level-1 assumes that the opponent is a level-0 player and thus plays the card matching the upturned card—the

TABLE II
RESULTS FOR SIX STRATEGIES PLAYING TEN ROUND-ROBIN ROUNDS

| | Rand. | 0101 | 0102 | 0103 | 0104 | 0105 |
|---|---|---|---|---|---|---|
| Rand. | X | 298/531 | 296/541 | 321/479 | 370/461 | 388/471 |
| 0101 | | X | 152/750 | 153/737 | 154/512 | 413/384 |
| 0102 | | | X | 346/510 | 323/468 | 404/427 |
| 0103 | | | | X | 376/413 | 454/362 |
| 0104 | | | | | X | 347/463 |
| 0105 | | | | | | X |

TABLE III
NUMBER OF WINS AND LOSSES FOR FIVE STRATEGIES PLAYING TEN ROUND-ROBIN ROUNDS

| | W | L |
|---|---|---|
| Rand. | 0 | 5 |
| 0101 | 2 | 3 |
| 0102 | 2 | 3 |
| 0103 | 4 | 1 |
| 0104 | 4 | 1 |
| 0105 | 3 | 2 |

best response to a random player. We introduce players/strategies from level-0 to level-13 as follows.

- Level-0: Always randomize with uniform probability on each card (a random strategy).
- Level-1 (the best response to level-0): Use the upcard matching strategy.
- Level-2 (the best response to level-1): A-on-K, K-on-Q, Q-on-J, . . ., 2-on-A.
  Note: To verify level-2, note that it is impossible to win the play turn where the opponent plays K. However, the strategy level-2 wins EVERY other card in that round, hence it must be the best response. One now realizes that analogous logic repeats itself for level-$k$ with $k \geq 2$, e.g.
- Level-3 (the best response to level-2): A-on-Q, K-on-J, Q-on-T, . . ., 3-on-A, 2-on-K, etc.

By following the above logic, you will see the following. Let cards be called by numbers 1–13. Level-$k$ for $k \geq 0$ will play the Ace on card number $1 + (14 - [1 + (k - 1) \bmod 13])$ mod 13 and then $K, Q, \ldots, 2$ on the next lower cards where 13 is "below" 1. That describes level-$k$, for all $k$. So there are exactly 14 patterns of play.

## VI. THE FIVE-PLAYER COMPETITION

In this section, we present the round-robin results of the competition between six (five players+Rand.) computer programs from the initial submission in 2009, with and without the 14 level-$k$ strategies. Table II shows the results of the round-robin play for these six strategies (0101, . . ., 0105)+Rand.

The results in Table II were generated by playing ten rounds for each strategy against every other strategy.

Table III summarizes the wins and losses. Table IV shows the total points accumulated by each player.

The results in Table II should be interpreted as follows. When the *column* plays the *row* player, the first figure shows the points awarded to the *row* player. For example, when Rand. played 0101, Rand. received 298 points and 0101 received 531 points. The strategies do not play themselves.

It is interesting to compare the results shown in Tables II and III. For example, strategy 0102 lost against two of the other strategies, beating Rand. and 0101. Strategy 0103 won the most points in total (2556; see Table IV) even though it lost to 0104. None of the five strategies submitted in the 2009 competition empirically dominated[1] all the others. All the strategies were beaten by at least one of the other strategies. The results in Tables II–IV indicate that the results (relative success of the different strategies) can be measured in a number of dimensions like the total number of wins, (cumulative) total points won, and pairwise comparison of strategies with points won against points lost. The competitors were told that they would be ranked by the total number of points won. Nevertheless, the other two dimensions of strategic success add more clarity to our understanding of the working of a strategy.

TABLE IV
CUMULATIVE RESULTS FOR EACH STRATEGY

| **Rand.** | **0101** | **0102** | **0103** | **0104** | **0105** |
|---|---|---|---|---|---|
| 298 | 531 | 541 | 479 | 461 | 471 |
| 296 | 152 | 750 | 737 | 512 | 384 |
| 321 | 153 | 346 | 510 | 468 | 427 |
| 370 | 154 | 323 | 376 | 413 | 362 |
| 388 | 413 | 404 | 454 | 347 | 463 |
| **1673** | **1403** | **2364** | **2556** | **2201** | **2107** |

We reran the competition and included the 14 level-$k$ strategies (see Table V). Unlike Table II, in this table, we show the point difference from the perspective of the *row* player (this is done in the interest of space). For example, when 0105 plays L-3, 0105 wins by a points difference of 76, but when L-9 plays L-10, L-10 wins with a points difference of 810.

The 14 level-$k$ strategies performed uniformly poorly against the strategies submitted by human participants. For this reason, we withdrew the 14 level-$k$ strategies from further competitions. We did, however, retain Rand., which is actually the same as L-0,

The initial round-robin 2009 *Repeated Goofspiel* competition had only five participants. It would be difficult to draw conclusions based on this limited number of players. Thus, we expanded the number of participating programs, adding another eight algorithms which were submitted in fall 2011 by eight doctoral students from the University of Arizona by making it part of a course requirement. Our main experimental report focuses on these 5+8+Rand.=14 strategies and examines the results that arose.

## VII. THE 14-PLAYER COMPETITION

### A. Results

We present the results in several tables.

Table VI shows the wins and losses for each of the 14 strategies. Each column shows the number of wins and losses for a given strategy, for each type of round-robin tournament. For example, the column labeled "W:20" shows the number of wins for each strategy in the round-robin tournament that comprised 20 rounds. Each pair of columns should add up to 13 (as each of the 14 strategies played against the other 13. Where this does not happen (for example, for 0106/0113. W:10), this means that

[1]In this paper, we use the terms empirically dominant and empirically dominated, rather than just dominant and dominated in order to avoid any confusion with the standard way that these terms are used in game theory.

TABLE V
RESULTS FROM TEN ROUNDS OF A ROUND-ROBIN COMPETITION WITH FIVE PLAYERS AND THE LEVEL-$k$ STRATEGIES

| | Rand. | 0101 | 0102 | 0103 | 0104 | 0105 | L-1 | L-2 | L-3 | L-4 | L-5 | L-6 | L-7 | L-8 | L-9 | L-10 | L-11 | L-12 | L-13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rand. | X | -300 | -210 | -277 | -294 | 24 | -277 | -48 | -93 | 17 | 49 | 112 | 118 | 181 | 82 | 102 | 88 | 22 | -128 |
| 0101 | | X | -638 | -597 | -224 | -13 | 0 | -650 | -310 | -190 | 10 | 190 | 350 | 490 | 610 | 710 | 790 | 850 | 890 |
| 0102 | | | X | -219 | 369 | 60 | 640 | 638 | 635 | -190 | 16 | 168 | 350 | 472 | 610 | 685 | 790 | 836 | 882 |
| 0103 | | | | X | 35 | -13 | 589 | 464 | 393 | 340 | 334 | 503 | 582 | 657 | 654 | 677 | 693 | 702 | 680 |
| 0104 | | | | | X | -43 | 285 | -62 | -302 | -297 | -128 | 28 | 167 | 242 | 356 | 450 | 526 | 582 | 610 |
| 0105 | | | | | | X | 105 | -4 | 76 | 30 | 103 | 48 | 199 | 191 | 424 | 325 | 258 | 302 | 111 |
| L-1 | | | | | | | X | -650 | -410 | -190 | 10 | 190 | 350 | 490 | 610 | 710 | 790 | 850 | 890 |
| L-2 | | | | | | | | X | -670 | -450 | -250 | -70 | 90 | 230 | 350 | 450 | 530 | 590 | 630 |
| L-3 | | | | | | | | | X | -690 | -490 | -310 | -150 | -10 | 110 | 210 | 390 | 350 | 390 |
| L-4 | | | | | | | | | | X | -710 | -530 | -370 | -230 | -90 | -10 | 70 | 130 | 170 |
| L-5 | | | | | | | | | | | X | -730 | -570 | -410 | -310 | -190 | -130 | -70 | -30 |
| L-6 | | | | | | | | | | | | X | -750 | -610 | -490 | -390 | -310 | -250 | -210 |
| L-7 | | | | | | | | | | | | | X | -770 | -650 | -550 | -470 | -410 | -370 |
| L-8 | | | | | | | | | | | | | | X | -790 | -690 | -610 | -550 | -490 |
| L-9 | | | | | | | | | | | | | | | X | -810 | -730 | -670 | -630 |
| L-10 | | | | | | | | | | | | | | | | X | -830 | -770 | -730 |
| L-11 | | | | | | | | | | | | | | | | | X | -850 | -810 |
| L-12 | | | | | | | | | | | | | | | | | | X | -870 |
| L-13 | | | | | | | | | | | | | | | | | | | X |

TABLE VI
NUMBER OF WINS AND LOSSES FOR THE 13 STRATEGIES+ RAND. PLAYING 10, 20, 30, AND 50 ROUND-ROBIN ROUNDS

| | W:10 | L:10 | W:20 | L:20 | W:30 | L:30 | W:50 | L:50 |
|---|---|---|---|---|---|---|---|---|
| Rand. | 1 | 12 | 0 | 13 | 0 | 13 | 1 | 12 |
| 0101 | 5 | 8 | 5 | 8 | 3 | 10 | 5 | 8 |
| 0102 | 4 | 9 | 3 | 10 | 4 | 9 | 1 | 12 |
| 0103 | 11 | 2 | 10 | 3 | 10 | 3 | 9 | 4 |
| 0104 | 5 | 8 | 10 | 3 | 8 | 5 | 9 | 4 |
| 0105 | 5 | 8 | 4 | 9 | 6 | 7 | 8 | 5 |
| 0106 | 6 | 6 | 3 | 10 | 3 | 10 | 4 | 9 |
| 0107 | 4 | 9 | 4 | 9 | 3 | 10 | 2 | 11 |
| 0108 | 11 | 2 | 11 | 2 | 11 | 2 | 11 | 2 |
| 0109 | 9 | 4 | 12 | 1 | 12 | 1 | 11 | 2 |
| 0110 | 10 | 3 | 12 | 1 | 11 | 2 | 13 | 0 |
| 0111 | 6 | 7 | 6 | 7 | 7 | 6 | 6 | 7 |
| 0112 | 5 | 8 | 6 | 7 | 5 | 8 | 6 | 7 |
| 0113 | 9 | 3 | 7 | 6 | 9 | 4 | 8 | 5 |

TABLE VII
TOTAL NUMBER OF POINTS (WINS/LOSSES) BY EACH OF THE 13 STRATEGIES+ RAND. PLAYING 10, 20, 30, AND 50 ROUND-ROBIN ROUNDS

| | 10 rounds | 20 rounds | 30 rounds | 50 rounds | Total |
|---|---|---|---|---|---|
| Rand. | -1687 | -4663 | -5359 | -8714 | -20423 |
| 0101 | -123 | -931 | -3173 | -2298 | -6525 |
| 0102 | -482 | -1590 | -1516 | -7075 | -10663 |
| 0103 | 1185 | 1916 | 3490 | 4848 | 11439 |
| 0104 | -598 | 1218 | 354 | -634 | 340 |
| 0105 | -254 | -462 | 125 | 681 | 90 |
| 0106 | 69 | -1169 | -1918 | -1830 | -4848 |
| 0107 | -435 | -757 | -3234 | -6469 | -10895 |
| 0108 | 764 | 1010 | 1084 | 2249 | 5107 |
| 0109 | 828 | **3251** | **5659** | **10001** | **19739** |
| 0110 | **1033** | 2808 | 3878 | 7154 | 14873 |
| 0111 | 103 | 1170 | 1581 | 1580 | 4434 |
| 0112 | -117 | 336 | 99 | 1071 | 1389 |
| 0113 | -100 | -323 | -378 | -219 | -1020 |

there was a draw. This case can be verified in Tables XIV and XXI.

By examining the results from the 14 program competitions we can clearly note that some strategies are better than others. The indication is that 0103, 0108, 0109, and 0110 seem to be the four strongest strategies in this set.

Table VII shows the total number of points accumulated by each strategy, for each round-robin tournament (of 10, 20, 30, and 50 rounds). The points are calculated by taking the difference in the number of points for each game and summing them for each round-robin tournament, and then summing across all the tournaments to arrive at the total. The strategy with the highest points total is shown in bold for each tournament. From these data, it would appear that 0109 is the strongest strategy, with 0110 also performing well. The other strategies we identify above (0103 and 0108) also perform relatively well, being in the third and fourth place with respect to points total. For instance, strategy 0108 was the best strategy in terms of the pairwise strategy comparison. It empirically dominated ten other strategies while being empirically dominated only by one. Perhaps this was due to the randomization component when not targeting to win a card. When 0108 won it did not win by high margins. Thus, it did not capture the most total points. It is quite hard to pinpoint why the four most successful strategies won most of their competitions. This will remain a topic for further research.

In order to try and determine if there is a difference between the two most successful strategies (0109 and 0110) we played them against each other in two separate 100 round competitions. In the first competition, 0109 won 4459 points and 0110 won 3870 points. In the second competition, 0109 won 4401 points and 0110 won 3998 points. It appears that for a many rounds *Repeated Goofspiel* play 0109 outperforms 0110, but we cannot say for certain that 0109 is superior to 0110. Table VIII shows the number of times that a strategy was *empirically dominant* or was *empirically dominated*. By *empirically dominant*, we mean that it beat all other strategies in all rounds of the round-robin tournament. As an example, strategy 0101 had a positive score in each of the four round-robin tournaments against 0112, so this would increment its dominant *counter* by one. Being *empirically dominated* means that the strategy loses to some strategy in every tournament. So 0101 empirically dominates 0112, and

TABLE VIII
NUMBER OF TIMES STRATEGIES WERE EMPIRICALLY DOMINANT OR BEING EMPIRICALLY DOMINATED, FOR THE 13 STRATEGIES+RAND

| | Dominant | Dominated |
|---|---|---|
| Rand. | 0 | 11 |
| 0101 | 2 | 6 |
| 0102 | 1 | 7 |
| 0103 | 9 | 2 |
| 0104 | 5 | 2 |
| 0105 | 2 | 4 |
| 0106 | 1 | 4 |
| 0107 | 1 | 7 |
| 0108 | 10 | 1 |
| 0109 | 9 | 1 |
| 0110 | 9 | 0 |
| 0111 | 4 | 4 |
| 0112 | 3 | 6 |
| 0113 | 6 | 3 |

TABLE IX
0101: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0101:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0102 | 167 | 212 | -1103 | 1155 |
| 0103 | -124 | -258 | -480 | -755 |
| 0104 | -53 | -237 | -439 | -249 |
| 0105 | 89 | 142 | -4 | -153 |
| 0106 | -4 | 153 | 29 | 54 |
| 0107 | -203 | -111 | -58 | -207 |
| 0108 | -54 | -200 | -165 | -484 |
| 0109 | -113 | -246 | -506 | -1061 |
| 0110 | -208 | -327 | -500 | -860 |
| 0111 | 85 | -588 | -938 | -1589 |
| 0112 | 88 | 122 | 279 | 346 |
| 0113 | -5 | -158 | -82 | 291 |
| Rand. | 212 | 565 | 794 | 1214 |

TABLE X
0102: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0102:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | -167 | -212 | 1103 | -1155 |
| 0103 | -220 | -329 | -739 | -1015 |
| 0104 | 334 | 51 | -301 | -607 |
| 0105 | 48 | -16 | -294 | -445 |
| 0106 | -115 | 40 | 12 | -223 |
| 0107 | -96 | -244 | -100 | -562 |
| 0108 | -117 | -239 | -300 | -507 |
| 0109 | -129 | -144 | -308 | -682 |
| 0110 | -150 | -342 | -506 | -872 |
| 0111 | -209 | -334 | -475 | -797 |
| 0112 | 160 | -401 | 12 | -1015 |
| 0113 | -64 | -20 | -363 | -551 |
| Rand. | 243 | 600 | 743 | 1356 |

TABLE XI
0103: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0103:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 124 | 258 | 480 | 755 |
| 0102 | 220 | 329 | 739 | 1015 |
| 0104 | 3 | 8 | 311 | -55 |
| 0105 | 111 | 9 | 170 | 278 |
| 0106 | 4 | 345 | 304 | 524 |
| 0107 | 124 | 275 | 575 | 739 |
| 0108 | 57 | -124 | -61 | -27 |
| 0109 | -96 | -61 | -399 | -831 |
| 0110 | -33 | -210 | -301 | -421 |
| 0111 | 245 | 363 | 328 | 847 |
| 0112 | 52 | 230 | 121 | 384 |
| 0113 | 174 | 225 | 418 | 477 |
| Rand. | 200 | 269 | 805 | 1163 |

TABLE XII
0104: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0104:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 53 | 237 | 439 | 249 |
| 0102 | -334 | 51 | 301 | 607 |
| 0103 | -3 | -8 | -211 | 55 |
| 0105 | -7 | 73 | -85 | -215 |
| 0106 | 51 | 109 | 305 | 434 |
| 0107 | -24 | 45 | 207 | 286 |
| 0108 | -190 | -262 | -262 | -740 |
| 0109 | -211 | -413 | -660 | -1445 |
| 0110 | -264 | 599 | -935 | -1512 |
| 0111 | 174 | 212 | 334 | 327 |
| 0112 | 14 | 159 | 221 | 365 |
| 0113 | -24 | 45 | 207 | 286 |
| Rand. | 167 | 371 | 493 | 669 |

TABLE XIII
0105: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0105:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | -89 | -142 | 4 | 153 |
| 0102 | -48 | 16 | 294 | 445 |
| 0103 | -111 | -9 | -170 | -278 |
| 0104 | 7 | -73 | 85 | 215 |
| 0106 | -15 | 11 | -34 | 167 |
| 0107 | 116 | 94 | 190 | 446 |
| 0108 | -148 | -91 | -55 | -477 |
| 0109 | 5 | -73 | -147 | -26 |
| 0110 | -90 | -146 | -113 | -79 |
| 0111 | 78 | -122 | -124 | 111 |
| 0112 | -55 | -145 | 115 | 21 |
| 0113 | -52 | -106 | -67 | -278 |
| Rand. | 148 | 324 | 147 | 261 |

0112 is empirically dominated by 0101. We provide the details of who empirically dominates who, in Section VIII and part A of the Appendix.

Tables IX–XXII present the detailed results for each of the 14 strategies that formed part of this simulation. We show these in the Appendix, with a brief commentary on each.

## VIII. DISCUSSION

By examining Tables VI and VII and also the more detailed Tables IX–XXII in the Appendix, we could draw the conclusion that strategy 0109 is the superior strategy out of the 14 that we have tested. 0109 achieved more points than any other strategy (by some margin with a value of 19739 as compared to its closest rival (0110) with a value of 14873). However, although it empirically dominated nine other strategies, 0108 (which only obtained an overall total number of points of 5107) empirically dominated even more strategies. We also have to note that 0109 was itself empirically dominated (by 0112). This is particularly interesting as 0112 was empirically dominated by six other strategies (see Table XX). This infers that, although 0109 was successful, it can still be empirically dominated by a

TABLE XIV
0106: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0106:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 4 | -153 | -29 | -54 |
| 0102 | 115 | -40 | -12 | 233 |
| 0103 | -4 | -345 | -304 | -524 |
| 0104 | 51 | -109 | -305 | -434 |
| 0105 | 15 | -11 | 34 | -167 |
| 0107 | 51 | 30 | 62 | 419 |
| 0108 | -16 | -161 | -378 | -384 |
| 0109 | 74 | -165 | -249 | -272 |
| 0110 | -88 | -86 | -132 | -281 |
| 0111 | -55 | 16 | -116 | 66 |
| 0112 | -71 | -233 | -441 | -463 |
| 0113 | 0 | -81 | -120 | -150 |
| Rand. | -7 | 169 | 72 | 181 |

TABLE XV
0107: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0107:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 54 | 111 | -58 | 207 |
| 0102 | 96 | 244 | 100 | 562 |
| 0103 | -124 | -275 | -575 | -739 |
| 0104 | 24 | -45 | -207 | -286 |
| 0105 | -116 | -94 | -190 | -446 |
| 0106 | -51 | -30 | -62 | -419 |
| 0108 | -15 | -43 | -143 | -448 |
| 0109 | -147 | -460 | -762 | -1620 |
| 0110 | -133 | -297 | -774 | -1152 |
| 0111 | -59 | -199 | 149 | -535 |
| 0112 | -10 | 28 | -325 | -456 |
| 0113 | -8 | -262 | -484 | -1032 |
| Rand. | 54 | 565 | 97 | -105 |

TABLE XVI
0108: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0108:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 54 | 200 | 168 | 484 |
| 0102 | 117 | 239 | 300 | 567 |
| 0103 | -57 | 124 | 61 | 27 |
| 0104 | 190 | 262 | 262 | 740 |
| 0105 | 148 | 91 | 55 | 477 |
| 0106 | 16 | 161 | 378 | 384 |
| 0107 | 15 | 43 | 143 | 448 |
| 0109 | -120 | -616 | -1053 | -1891 |
| 0110 | 39 | -62 | -97 | -268 |
| 0111 | 169 | 206 | 337 | 484 |
| 0112 | 52 | 60 | 123 | 190 |
| 0113 | 32 | 128 | 120 | 236 |
| Rand. | 109 | 174 | 287 | 371 |

TABLE XVII
0109: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0109:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 113 | 246 | 506 | 1061 |
| 0102 | 128 | 144 | 308 | 682 |
| 0103 | 96 | 261 | 399 | 831 |
| 0104 | 211 | 413 | 660 | 1445 |
| 0105 | -5 | 73 | 147 | 26 |
| 0106 | -74 | 165 | 249 | 272 |
| 0107 | 147 | 460 | 762 | 1620 |
| 0108 | 120 | 616 | 1053 | 1891 |
| 0110 | -66 | 32 | 168 | -18 |
| 0111 | 46 | 232 | 432 | 617 |
| 0112 | -71 | -59 | -160 | -193 |
| 0113 | 124 | 469 | 702 | 1290 |
| Rand. | 59 | 199 | 433 | 477 |

TABLE XVIII
0110: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0110:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 208 | 327 | 500 | 860 |
| 0102 | 150 | 342 | 506 | 872 |
| 0103 | 33 | 210 | 301 | 421 |
| 0104 | 264 | 599 | 935 | 1512 |
| 0105 | 90 | 146 | 113 | 79 |
| 0106 | 88 | 86 | 132 | 281 |
| 0107 | 133 | 297 | 774 | 1152 |
| 0108 | -39 | 62 | 97 | 268 |
| 0109 | 66 | -32 | -168 | 18 |
| 0111 | -128 | 126 | 209 | 222 |
| 0112 | 137 | 339 | 459 | 817 |
| 0113 | -22 | 39 | -198 | 42 |
| Rand. | 53 | 267 | 218 | 610 |

TABLE XIX
0111: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0111:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | -85 | 588 | 938 | 589 |
| 0102 | 209 | 334 | 475 | 797 |
| 0103 | -245 | -363 | -328 | -847 |
| 0104 | -174 | -212 | -334 | -327 |
| 0105 | -78 | 122 | 124 | -111 |
| 0106 | 55 | -16 | 116 | -66 |
| 0107 | 59 | 199 | 149 | 535 |
| 0108 | -169 | -206 | -337 | -484 |
| 0109 | -46 | -232 | -432 | -617 |
| 0110 | 128 | -126 | -209 | -222 |
| 0112 | -20 | -29 | -66 | 80 |
| 0113 | 293 | 656 | 938 | 1529 |
| Rand. | 176 | 455 | 547 | 724 |

strategy that did not perform very well in the overall competition but which, presumably, has found a way to exploit a weakness in the strategy adopted by 0109. This suggests that there is still a lot of scope for further research into this area. We would like to acknowledge an important point observed by one of the referees: For this competition, we simply asked the participants to submit their programs and report on the logic of their entry. We did not specify any time constraints between the consecutive card plays. It was perhaps for this reason that we did not have many strategies that involved anticipating future developments of the game, e.g., by using search or Monte Carlo simulations of the game, as the contestants might have been worried about the computational resources that would have been required. The one exception (0103) computed the best response when the number of remaining cards was very small.

## IX. FUTURE RESEARCH DIRECTIONS

We hope that this paper prompts further research into *Goofspiel*. As this is only one of a small number of papers that have been reported in the scientific literature there are many opportunities for research. We present just a few here.

1) As mentioned in Section VIII, the vast majority of the submitted strategies only considered a limited number of previous cards (often just one). It was only toward the end of

TABLE XX
0112: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0112:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | -88 | -122 | -279 | -346 |
| 0102 | -160 | 401 | -12 | 1015 |
| 0103 | -52 | -230 | -121 | -384 |
| 0104 | -14 | -159 | -211 | -365 |
| 0105 | 55 | 145 | -115 | 21 |
| 0106 | 71 | 233 | 441 | 463 |
| 0107 | -10 | -28 | 325 | 456 |
| 0108 | -52 | -60 | -123 | -190 |
| 0109 | 71 | 59 | 160 | 193 |
| 0110 | -137 | -339 | -459 | -817 |
| 0111 | 20 | 29 | 66 | -80 |
| 0113 | -75 | -93 | -202 | -193 |
| Rand. | 254 | 500 | 629 | 1298 |

TABLE XXI
0113: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **0113:** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | 5 | 158 | 83 | 291 |
| 0102 | 64 | -20 | 363 | 551 |
| 0103 | -174 | -225 | -418 | -477 |
| 0104 | 85 | 219 | 301 | 454 |
| 0105 | 52 | 106 | 67 | 152 |
| 0106 | 0 | 81 | 120 | 150 |
| 0107 | 8 | 262 | 484 | 1032 |
| 0108 | -32 | -128 | -120 | -236 |
| 0109 | 75 | -469 | -702 | -1290 |
| 0110 | 22 | -39 | 108 | -42 |
| 0111 | -293 | -656 | -938 | -1529 |
| 0112 | 75 | 93 | 202 | 193 |
| Rand. | 13 | 295 | 72 | 532 |

TABLE XXII
RAND.: COMPETITION RESULTS AGAINST THE OTHER 13 STRATEGIES IN 10, 20, 30, AND 50 ROUNDS IN TOTAL POINTS WON

| **Rand.** | 10 rounds | 20 rounds | 30 rounds | 50 rounds |
|---|---|---|---|---|
| 0101 | -212 | -369 | -805 | -1165 |
| 0102 | -243 | -606 | -743 | -1356 |
| 0103 | -206 | -369 | -805 | -1165 |
| 0104 | -167 | -371 | -493 | -669 |
| 0105 | -148 | -324 | -147 | -261 |
| 0106 | 7 | -169 | -73 | -181 |
| 0107 | -54 | -565 | -97 | 105 |
| 0108 | -109 | -174 | -287 | -371 |
| 0109 | -59 | -199 | -433 | -477 |
| 0110 | -53 | -267 | -218 | -610 |
| 0111 | -176 | -455 | -547 | -734 |
| 0112 | -254 | -500 | -639 | -1298 |
| 0113 | -13 | -295 | -72 | -532 |

the game that some strategies calculated optimal moves. It would be interesting to investigate strategies where more previous cards were considered in deciding their next play. This could also be extended to previous games against the same opponent to try and work out what their play had been and, therefore, how to respond.

2) As mentioned in Section II, *Goofspiel* suffers from the combinatorial explosion of the number of pure strategies. This could be a fruitful line of research. Is it possible to find some of the *good* pure strategies, without resorting to having to search the entire search space of candidate strategies? The recent successes in Monte Carlo search [26] might provide inspiration.
3) Related to the above point, it would be interesting to not limit the number of cards to $N = 13$. This is an arbitrary value set by the makeup of a standard deck of cards. There is no reason why $N$ cannot be increased in order to further investigate the complexity of the game or explore how learning strategies adapt to larger values of $N$.
4) As a further development for *Repeated Goofspiel*, there has been some interest in the *IPD* in recent years, where strategies cooperate with *partner* strategies to maximize their partner's score, to the detriment of their own [27]–[29]. This could be extended to *Repeated Goofspiel* and introduces many research directions such as handshaking protocols, identification mechanisms, etc.

## X. CONCLUSION

Unlike the *IPD*, *Repeated Goofspiel* has many more decision options at each decision point. Research into the *IPD* has been active for at least 25 years, and continues to this day (see, for example, [27] and [30]–[35] for just a few of the many recent papers). Although *Goofspiel* was introduced about 80 years ago, it has never been fully investigated or analyzed. We hope that this paper will bring this game to the attention of a much wider body of AI researchers and that it will motivate others to study this fascinating game.

## APPENDIX

### A. Points Scored by Each Strategy

Tables IX–XXII provide the total points accumulated in a competition against the other 13 strategies for the 10, 20, 30, and 50 round competitions.

From Table VI, we can see that strategy 0101 won 18 games and lost 34. It is empirically dominated by six other strategies (0103, 0104, 0107, 0108, 0109, 0110). However, it empirically dominated two strategies (0112 and Rand).

Strategy 0102 won 12 games and lost 42. The only strategy it empirically dominated was the random strategy. It was empirically dominated by seven strategies (0103, 0107, 0108, 0109, 0110, 0111, and 0113)

Strategy 0103 (Table XI) won 40 games and lost 12. It is one of the strongest strategies empirically dominating nine other strategies (0101, 0102, 0105, 0106, 0107, 0111, 0112, 0113, and Rand). However, it is empirically dominated by two strategies (0109 and 0110).

Strategy 0104 (Table XII) won 32 games and lost 20. It empirically dominated 5 strategies (0101, 0106, 0111, 0112, and Rand), but was empirically dominated by 2 strategies (0108 and 0109).

Strategy 0105 (Table XIII) won 23 games and lost 29. It was empirically dominated by four strategies (0103, 0108, 0110, and 0113) and empirically dominated by two strategies (0107 and Rand).

Strategy 0106 (Table XIV) is a weak strategy that won only 16 games and lost 35 with one tie (against 0113). It was empirically dominated by four strategies (0103, 0108, 0110, and

0112) and empirically dominated only one strategy (0107). It is notable that it lost to the random strategy.

Strategy 0107 (Table XV) is essentially a losing strategy. It won 13 games and lost 39. The only strategy it empirically dominated was 0102. It was empirically dominated by seven strategies (0103, 0105, 0106, 0108, 0109, 0110, and 0113). It lost one game to the random strategy.

Strategy 0108 (Table XVI) won 44 games and lost only eight. It was empirically dominated only by 0109 and empirically dominated ten other strategies (0101, 0102, 0104, 0105, 0106, 0107, 0111, 0112, 0113, and Rand).

Strategy 0109 (Table XVII) won 44 games and lost eight. It was only empirically dominated by 0112 and empirically dominated nine other strategies (0101, 0102, 0103, 0104, 0107, 0108, 0111, 0113, and Rand).

Strategy 0110 (Table XVIII) won 46 games and lost only six. It empirically dominated nine other strategies (0101, 0102, 0103, 0104, 0105, 0106, 0107, 0112, and Rand). This strategy was not empirically dominated by any other strategy, the only strategy to achieve this.

Strategy 0111 (Table XIX) wins about the same number of games that it loses (25 wins, 27 losses). It was empirically dominated by four strategies (0103, 0104, 0108, and 0109) and also empirically dominated four strategies (0102, 0107, 0113, and Rand).

Strategy 0112 (Table XX) was not very successful. It won 22 games and lost 30. It was empirically dominated by six strategies (0101, 0103, 0104, 0108, 0110, and 0113) and empirically dominated three strategies (0106, 0109, and Rand).

Strategy 0113 (Table XXI) won 32 games, while losing 21. It was empirically dominated by three strategies (0103, 0108, and 0111) and empirically dominated six strategies (0101, 0104, 0105, 0107, 0112, and Rand).

The random strategy (Table XXII) only won two games, lost 49, and drew one. It is interesting to observe that it wins by seven points against 0106 in the ten-round competition, ties with 0107 in the 20-round competition, and wins by 105 points against 0107 in the 50-round competition. One ought not draw conclusions from the seven-point win in the ten-round competition against 0106. However, for 0107, losing by 105 and drawing once implies that 0107 frequently resorts to its default substrategy of random play when it cannot "believably" predict/classify the opponent's play strategy given an opponent selecting his cards at random.

### B. The Format of the Computer Program

All the participants were required to submit two items: 1) a computer program in C++ in the format described in the following; and 2) an English description of their logic (their ideas) as implemented in their program/code.

The computer program entry in this competition comprises two main software components.

1) The entries from the competitors (submitted as a C++ class computer programs). In essence, the participants had to supply a function which specifies how much they are willing to bid for the current card.
2) The competition organizers provided a framework that enabled the players/competitors to develop their entry without having to write the necessary supporting code. This also ensured that every competitor was using the same set of rules.

Both these elements are described in further detail in the following.

### C. Competition Entries

To enter the competition, the entrants had to supply a C++ class. The class had to include the following function definition (shown in C++ syntax):

*int nextPlay(int upCard, int round, int noOfRounds)*

where the *upCard* variable is the value of the card being bid for. It is defined as an integer as we are only concerned with the value of the card, not with the suit. The *round* variable is the current round being played (starting from zero). The *noOfRounds* variable tells the competitors how many rounds there will be. In the competition reported here, $noOfRounds = 10$. The function expects a value between 1 and 13 to be returned. This is the amount the competitor is bidding for the current *upCard*.

The class submitted by the entrants had to be inherited from a strategy class. To provide a concrete example, an entry could have the following form:

```
class exampleStrategy: public strategy {
public:
exampleStrategy();
virtual exampleStrategy();
int nextPlay(int upCard, int round, int noOfRounds);
}.
```

Note that the *nextPlay* function has to be defined, as we insist (with the C++ framework) that any class inheriting from strategy has to supply a definition of this function. Of course, the competitors are able to supply any other data members and functions that they require to support their strategy.

### D. Software Framework

The software framework provides two services. The first is to play each entrant against every other entrant. This is undertaken once all the entries have been received. It is not discussed in detail here as it is purely part of the competition administration. The framework also provides a set of support functions, which can be called by the competitors. They could be developed independently by every competitor. Providing them as part of the framework not only saves development time but also ensures that each competitor has access to tried and tested code. The support functions we provided were as follows (again, they are presented in C++ syntax).

- *Void displayHand(int round, bool short = true)* displays the status of your hand, for the round you are interested in, where *round* is the round to report, and *short* specifies whether to display the cards in long form (e.g., Ten of Diamonds) or a short form (e.g., TD). By default, the short form is used.
- *Void displayHandFull(int round, bool opponent = false, bool short = true)* displays the cards that have been played by you or your opponent, for a specified round, where *round* is the round to report, *opponent* specifies which hand

to report (yours or your opponent; the default is to display your own hand), and *short* specifies whether to display the cards in long form (e.g., Ten of Diamonds) or a short form (e.g., TD). By default, the short form is used.

- *Void displayLastCardPlayed(int round, bool short = true)* displays the last card that was played, where *round* is the round to report, and *short* specifies whether to display the cards in long form (e.g., Ten of Diamonds) or a short form (e.g., TD). By default, the short form is used.
- *Int getOpponentsPoints(int round)* returns the number of points of your opponent for the round that provided, where *round* is the round to report.
- *Int getOpponentsTotalPoints(int noOfRounds)* returns the number of points of your opponent for all the rounds played so far, where *noOfRounds* is the total number of rounds that are being played.
- *Int getPoints(int round, bool own = true)* returns the number of points for you or your opponent, for the round provided, where *round* is the round to report, and *own* is used to determine whether you want to find out the points for you or your opponent. The default is to return your own points. Note that the *getOpponentsPoints* simply calls this function, but a specific function was implemented in order to make the use of the functions slightly easier.
- *Int getTotalPoints(int noOfRounds, bool own = true)* returns the total number of points for you or your opponent, considering all the rounds played so far, where *noOfRounds* is the total number of rounds that are being played, and *own* is used to determine whether you want to find out the points for you or your opponent. The default is to return your own points. Note that the *getOpponentsTotalPoints* simply calls this function, but we implemented a specific function in order to make the use of the functions slightly easier.
- *Bool opponentPlayedCard(int round, int val)* is used to find out if your opponent has played a particular card in a given round, where *round* is the round of interest, and *val* is the card of interest. The function returns true if the card has been played in that round, false otherwise.
- *Bool playedCard(int round, int val)* is used to find out if you have played a particular card in a given round, where *round* is the round of interest, and *val* is the card of interest. The function returns true if the card has been played in that round, false otherwise.
- *Int unusedRandomCard(int currentRound)* returns a random number between $1, \ldots, 13$, representing a card that you have NOT played so far. You need to specify the round of interest (usually the current round).

## ACKNOWLEDGMENT

The authors would like to thank M. Dufwenberg for his early involvement and his advice and gratefully acknowledge Amnon Rapoport's advice of conducting the round-robin competition. They would also like to thank E. Hanani for pointing out to them that Zermelo's theorem does not apply to *Goofspiel* and directing them to [22]. Special thanks go to S. Zeng and N. Suakkaphong from the University of Arizona, Tucson, AZ, USA, for their help in running the computerized round-robin competitions and to the 13 participants who submitted entries to the competition.

## REFERENCES

[1] A. Tucker, "Merrill Flood (interview): The Princeton mathematics community in the 1930s," Transcript 30.
[2] A. Tucker, "The Princeton mathematical community in the 30's: An oral-history project," The Trustees of Princeton University, Transcript 11 (PMC11).
[3] R. Axelrod, *The Evolution of Cooperation*. New York, NY, USA: BASIC Books, 1984.
[4] G. Kendall, X. Yao, and S. Chong, *The Iterated Prisoner's Dilemma: 20 Years On*. Singapore: World Scientific, 2007.
[5] D. Billings, "Thoughts on Roshambo," *Int. Comput. Games Assoc. J.*, vol. 23, no. 1, pp. 3–8, 2000.
[6] D. Billings, "The first international Roshambo programming competition," *Int. Comput. Games Assoc. J.*, vol. 23, no. 1, pp. 42–50, 2000.
[7] L. Fisher, *Rock, Paper, Scissors: Game Theory in Everyday Life*. Carlsbad, CA, USA: Hay House, 2010.
[8] M. Wittkamp and L. Barone, "Evolving adaptive play for the game of spoof using genetic programming," in *Proc. IEEE Symp. Comput. Intell. Games*, 2006, pp. 164–172.
[9] M. Wittkamp, L. Barone, and L. While, "A comparison of genetic programming and look-up table learning for the game of spoof," in *Proc. IEEE Symp. Comput. Intell. Games*, 2007, pp. 63–71.
[10] M. Zinkevich, M. Bowling, and M. Wunder, "The Lemonade stand game competition: Solving unsolvable games," *ACM SIGecom Exchanges*, vol. 10, no. 1, pp. 35–38, 2011.
[11] D. Gale, "Information in games with finite resources," *Ann. Math. Studies*, vol. 39, pp. 141–145, 1957.
[12] T. S. Ferguson and C. Melolidakis, "Games with finite resources," *Int. J. Game Theory*, vol. 29, no. 2, pp. 289–303, 2000.
[13] R. D. Luce and H. Raiffa, *Games and Decisions: Introduction and Critical Survey*. New York, NY, USA: Wiley, 1957.
[14] S. Ross, "Goofspiel: The game of pure strategy," *J. Appl. Probab.*, vol. 8, no. 3, pp. 621–625, 1971.
[15] M. Dror, "Simple proof for Goofspiel: The game of pure strategy," *Adv. Appl. Probab.*, vol. 8, pp. 711–712, 1989.
[16] D. P. Foster and H. Young, "Learning, hypothesis testing, and Nash equilibrium," *Games Econ. Behav.*, vol. 45, no. 1, pp. 73–96, 2003.
[17] P. Battigalli, M. Gilli, and M. C. Molinari, *Learning and Convergence to Equilibrium in Repeated Strategic Interactions: An Introductory Survey*. Milan, Italy: Università Commerciale, 1992.
[18] P. Battigalli, M. Galli, and M. C. Molinari, "Learning and convergence to equilibrium in repeated strategic interactions: An introductory survey," *Ricerche Economiche*, vol. 46, pp. 335–378, 1992.
[19] G. W. Brown, "Iterative solution of games by fictitious play," in *Activity Analysis of Production and Allocation*. New York, NY, USA: Wiley, 1951, pp. 374–376.
[20] E. Kalai and E. Lehrer, "Rational learning leads to Nash equilibrium," *Econometrica*, vol. 61, no. 5, pp. 1019–1045, 1993.
[21] D. Fudenberg and D. M. Kreps, "Learning mixed equilibria," *Games Econ. Behav.*, vol. 5, no. 3, pp. 320–367, 1993.
[22] U. Schwalbea and P. Walker, "Zermelo and the early history of game theory," *Games Econ. Behav.*, vol. 34, no. 1, pp. 123–137, 2001.
[23] C. Camerer, T.-H. Ho, and J. Chong, "Behavioral game theory: Thinking learning and teaching," in *Advances in Understanding Strategic Behavior: Game Theory, Experiments and Bounded Rationality; Essays in Honor of Werner Guth*. Basingstoke, Hampshire, U.K.: Palgrave Macmillan, 2004, pp. 119–179.
[24] M. Costa-Gomes and V. P. Crawford, "Cognition and behavior in two-person guessing games: An experimental study," *Amer. Econ. Rev.*, vol. 96, no. 5, pp. 1737–1768, 2006.
[25] T. Chan and D. Craft, "Learning card playing strategies with support vector machines," Massachusetts Inst. Technol. (MIT), Cambridge, MA, USA, Tech. Rep., 2003.
[26] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, S. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.
[27] J. Li and G. Kendall, "Collective behavior and kin selection in evolutionary IPD," *J. Multiple-Valued Logic Soft Comput.*, vol. 16, no. 6, pp. 509–525, 2010.

[28] A. Rogers, R. Dash, S. Ramchurn, P. Vytelingum, and N. Jennings, "Coordinating team players within a noisy iterated prisoner's dilemma tournament," *Theor. Comput. Sci.*, vol. 377, no. 1–3, pp. 243–259, 2007.
[29] J. Li, *How to Design a Strategy to Win an IPD Tournament*. Singapore: World Scientific, 2007, ch. 4, pp. 89–104.
[30] D. Ashlock and E.-Y. Kim, "Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 647–659, Oct. 2008.
[31] D. Ashlock, E.-Y. Kim, and W. Ashlock, "Fingerprint analysis of the noisy prisoner's dilemma using a finite-state representation," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 2, pp. 154–167, Jun. 2009.
[32] S. Mittal and K. Deb, "Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 554–565, Jun. 2009.
[33] J. Li and G. Kendall, "A strategy with novel evolutionary features for the iterated prisoner's dilemma," *Evol. Comput.*, vol. 17, no. 2, pp. 257–274, 2009.
[34] H.-Y. Quek, K. C. Tan, C.-K. Goh, and H. Abbass, "Evolution and incremental learning in the iterated prisoner's dilemma," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 303–320, Apr. 2009.
[35] J. Li, P. Hingston, and G. Kendall, "Engineering design of strategies for winning iterated prisoner's dilemma competitions," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 4, pp. 348–360, Dec. 2011.

**Moshe Dror** received the M.Sc. degree in mathematical methods and the I.E. degree from Columbia University, New York, NY, USA, in 1974 and 1975, respectively, and the Ph.D. degree in management science from University of Maryland, College Park, MD, USA, in 1983.

He has conducted well-documented research on a variety of topics in logistics, scheduling, manufacturing, cooperative game theory/cost allocation, and on applications of agent theory in project management. He has published numerous book chapters and over 100 papers in the primary refereed journals which focus on management of operations—*Management Science*, *Operations Research*, *Naval Research Logistics*, *IIE Transactions*, *Transportation Science*, among others. He has also published in a number of mathematical and computer science journals such as the *SIAM Journal on Discrete Mathematics*, the *Journal of Complexity*, *ORDER*, *Information Processing Letters*, *Discrete Mathematics*, *Discrete Applied Mathematics*, *Discrete Optimization*, the *International Journal of Computational Geometry & Applications*, the *Journal of Applied Probability*, in addition to publications in a major economics journals such as the *International Journal of Game Theory* and *Games & Economic Behavior*. He has edited a book *Arc Routing: Theory, Solutions, and Applications* (New York, NY, USA: Springer-Verlag, 2000) and coedited a book *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications* (New York, NY, USA: Springer-Verlag, 2002). He has also edited three special issues in *Operations Research*, the *European Journal of Operational Research*, and *IIE Transactions*.

**Graham Kendall** (M'03–SM'10) received the B.S. degree in computation (first class honors) from the Institute of Science and Technology, University of Manchester, Manchester, U.K., in 1997 and the Ph.D. degree in computer science from the University of Nottingham, Nottingham, U.K., in 2001.

His previous experience includes almost 20 years in the information technology industry where he held both technical and managerial positions. He is a Professor of Computer Science at the University of Nottingham and is currently based at their Malaysia Campus, Selangor, Malaysia, where he holds the position of Vice-Provost (Research and Knowledge Transfer). He is a Director of two companies (Aptia Solutions Ltd., Nottingham, U.K. and Nottingham MyRIAD Solutions Sdn Bhd, Malaysia). During his career, he has edited or authored ten books, published over 150 refereed papers, and has guest edited special issues of journals including *Annals of Operations Research*, the *Journal of Scheduling*, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and the *International Computer Games Association Journal*. His research interests include adaptive learning (with an emphasis on evolving games), heuristic development (particularly hyperheuristics), optimization, scheduling (particularly sports), and artificial intelligence.

Prof. Kendall is a Member of the Automated Scheduling, Optimization, and Planning Research Group. He is a Fellow of the Operational Research Society. He is an Associate Editor of seven international journals, including two IEEE journals: the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES. He chaired the Multidisciplinary International Conference on Scheduling: Theory and Applications in 2003, 2005, 2007, 2009, and 2011, and has chaired several other international conferences, which included establishing the IEEE Symposium on Computational Intelligence and Games. He chaired this symposium in 2005 and 2006. He has been a member of the program (or refereeing) committees of over 160 international conferences over the last few years. He has been awarded externally funded grants worth over £6 million from a variety of sources including the Engineering and Physical Sciences Research Council (EPSRC) and commercial organizations.