

An Approach to Bounded Rationality

Eli Ben-Sasson
Technion

Adam Tauman Kalai
TTI Chicago

Ehud Kalai
Northwestern University

July 13, 2006

Abstract

A central question in game theory, learning, and other fields is how a rational intelligent agent should behave in a complex environment, given that it cannot perform unbounded computations. We study strategic aspects of this question by formulating a simple model of a game with additional costs (computational or otherwise) for each strategy. While a zero-sum game with strategy costs is no longer zero-sum, we show that its Nash equilibria have an interesting structure and the game has a new type of “value.” We also show that potential games with strategy costs remain potential games. Both zero-sum and potential games with strategy costs maintain a very appealing property: simple learning dynamics converge to Nash equilibrium.

1 The Approach and Basic Model

How should an intelligent agent play a complicated game like chess, given that it does not have unlimited time to think? This question reflects one fundamental aspect of “bounded rationality,” a term coined by Herbert Simon [2]. However, bounded rationality has proven to be a slippery concept to formalize (prior work has focused largely on finite automata playing simple repeated games such as prisoner’s dilemma, e.g. [3, 4, 5, 6]). This paper focuses on the *strategic* aspects of decision-making in complex multi-agent environments, i.e., on how a player should choose among strategies of varying complexity, given that its opponents are making similar decisions. Our model applies to general strategic games and allows for a variety of complexities that arise in real-world applications. For this reason, it is applicable to one-shot games, to extensive games, and to repeated games, and it generalizes existing models such as repeated games played by finite automata.

Let us motivate with an example of two players preparing to play a computerized chess game for \$100K prize. Suppose, for simplicity, the players simultaneously choose among two options: to use a \$10K program A or an advanced program B, which costs \$50K. We refer to the row chooser as white and to the column chooser as black, with the corresponding advantages reflected by the win probabilities of white described

in Table 1a. For example, when both players use program A, white wins 55% of the time and black wins 45% of the time (we ignore draws). The players naturally want to choose strategies to maximize their *expected net* payoffs, i.e., their expected payoff minus their cost. Each cell in Table 1b contains a pair of payoffs in units of thousands of dollars; the first is white's net expected payoff and the second is black's.

a)	A	B	b)	A (-10)	B (-50)
A	55%	13%	A (-10)	45, 35	3, 37
B	93%	51%	B (-50)	43,-3	1,-1

Figure 1: a) Table of first-player winning probabilities based on program choices. b) Table of expected net earnings in thousands of dollars. The unique equilibrium is (A,B) which strongly favors the second player.

This simple example illustrates how computational costs can dramatically change the outcome of a constant-sum game. Everything about the game seems to favor white. Yet due to the (symmetric) costs, at the unique Nash equilibrium (A,B) of Table 1b, black wins 87% of the time and nets \$34K more than white. In fact, it is a dominant strategy for white to play A and for black to play B. To see this, note that playing B increases white's probability of winning by 38%, independent of what black chooses. Since the pot is \$100K, this is worth \$38K in expectation, but B costs \$40K more than A. On the other hand, black enjoys a 42% increase in probability of winning due to B, independent of what white does, and hence is willing to pay the extra \$40K.

Before formulating the general model, we comment on some important aspects of the chess example. First, traditional game theory states that chess can be solved in "only" two rounds of elimination of dominated strategies [11], and the outcome with optimal play should always be the same: either a win for white or a win for black. This theoretical prediction is in stark contrast to reality: in grandmaster play, the outcome is very nondeterministic with the white winning about 4% more often than black. The game is too large and complex to be solved by brute force.

Second, we have been able to analyze the above chess program selection example exactly because we formulated as a game with a small number of strategies per player. We elaborate on the advantages and disadvantages of our simplistic approach in Section 5. Note that our model also applies to the full game of chess, though it is beyond our means to analyze such a large game.¹

Third, in the example above we used monetary software cost to illustrate a type of strategy cost. But the same analysis could accommodate many other types of costs that can be measured numerically and subtracted from the payoffs, such as time or effort involved in the development or execution of a strategy, and other resource costs. Additional examples in this paper include the number of states in a finite automaton, the number of gates in a circuit, and the number of turns on a commuter's route. Our analysis is limited, however, to cost functions that depend only on the strategy of the player and not the strategy chosen by its opponent. For example, if our players above

¹We also discuss an integer factorization game which has a huge number of strategies, but for which we can analyze the game under standard cryptographic assumptions.

were renting computers A or B and paying for the time of actual usage, then the cost of using A would depend on the choice of computer made by the opponent.

Generalizing the example above, we consider a normal form game with the addition of strategy costs, a player-dependent cost for playing each strategy. Our main results regard two important classes of games: constant-sum and potential games. We show that despite changes due to strategy costs, some important properties are preserved. First, in the class of two-person constant-sum games the equilibria sets have simple elegant structures. Second, learning dynamics converge in both classes of games.

2 Definition of strategy costs

We first define an N -person normal-form game $G = (N, S, p)$ consisting of finite sets of *pure strategies* $S = (S_1, \dots, S_N)$ for the N players, and a payoff function $p : S_1 \times \dots \times S_N \rightarrow \mathbf{R}^N$. Players simultaneously choose strategies $s_i \in S_i$ after which player i is rewarded with $p_i(s_1, \dots, s_N)$. A randomized or *mixed strategy* σ_i for player i is a probability distribution over its pure strategies S_i ,

$$\sigma_i \in \Delta_i = \left\{ x \in \mathbf{R}^{|S_i|} : \sum x_j = 1, x_j \geq 0 \right\}.$$

We extend p to $\Delta_1 \times \dots \times \Delta_N$ so that $p_i(\sigma_1, \dots, \sigma_N) = \mathbf{E}[p_i(s_1, \dots, s_N)]$ where each s_i is drawn from σ_i , independently. Denote by $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_N)$ and similarly for σ_{-i} . A *best response* by player i to σ_{-i} is $\sigma_i \in \Delta_i$ such that $p_i(\sigma_i, \sigma_{-i}) = \max_{\sigma'_i \in \Delta_i} p_i(\sigma'_i, \sigma_{-i})$. A (mixed strategy) *Nash equilibrium* of G is a vector of strategies $(\sigma_1, \dots, \sigma_N) \in \Delta_1 \times \dots \times \Delta_N$ such that each σ_i is a best response to σ_{-i} .

We now define G^{-c} , the game G with strategy costs $c = (c_1, \dots, c_N)$, where $c_i : S_i \rightarrow \mathbf{R}$. It is simply an N -person normal-form game $G^{-c} = (N, S, p^{-c})$ with the same sets of pure strategies as G , but with a new payoff function $p^{-c} : S_1 \times \dots \times S_N \rightarrow \mathbf{R}^N$ where,

$$p_i^{-c}(s_1, \dots, s_N) = p_i(s_1, \dots, s_N) - c_i(s_i), \text{ for } i = 1, \dots, N.$$

We similarly extend c_i to Δ_i in the natural way.

3 Two-person constant-sum games with strategy costs

Recall that a game is constant-sum (k-sum for short) if at every combination of individual strategies, the players' payoffs sum to some constant k . Two-person k-sum games have some important properties, not shared by general sum games, which result in more effective game-theoretic analysis. Since incorporating strategy costs into a k-sum game destroys the constant-sum property, it is not clear which of the properties survive.

One property that is violated by the chess example could be called the ‘‘advantage of strength’’ property. Formally, we say Player 1 is *stronger* than Player 2 in a square game if $p_1(s_1, s_2) \geq p_2(s_2, s_1)$ for all strategies s_1, s_2 . At equilibrium of a k-sum

game, a stronger player must have a payoff at least as large as its opponent. This is no longer the case after incorporating strategy costs, as seen in the chess example, where Player 1 is stronger (even including strategy costs), yet his equilibrium payoff is smaller than 2's.

The remainder of the section covers properties of k-sum games that are preserved. The min-max theorem states that every two-person k-sum game has a *value* v such that the payoffs to the players are $(v, k-v)$ at any equilibrium. Moreover, each player has a set of mixed *optimal strategies*, and the set of equilibria is exactly the cross-product of these two sets. This condition is often referred to as *exchangeability* because it states that if strategy pairs (σ_1, σ_2) and (σ'_1, σ'_2) are both equilibria, then so are (σ_1, σ'_2) and (σ'_1, σ_2) . Lastly, it is well-known that equilibria in two-person k-sum games can be *learned* in repeated play by simple dynamics that are guaranteed to converge [18].

The above properties do not hold for general-sum games. There may be a multiplicity of equilibria with very different payoffs, and players may face coordination problems among the multiple equilibria. Unfortunately, there are no known learning dynamics that efficiently converge to Nash equilibrium in general-sum games. In fact, recent results suggest that even efficiently computing one equilibrium in an $n \times n$ two-person general-sum game is computationally intractable in the worst case [10, 9]. We show that games with strategy costs are more similar to k-sum games, in these senses.

Theorem 1. *Let G be a finite two-person k-sum game and G^{-c} be the game with strategy costs $c = (c_1, c_2)$.*

1. *The equilibria of G^{-c} are exchangeable: the set of mixed-strategy Nash equilibria is a cross product $OPT_1 \times OPT_2$, where the sets OPT_1 and OPT_2 are called the optimal strategies for players 1 and 2.*
2. *Among equilibria, each player's choice of strategy affects only its **opponent's** payoff. More precisely, the expected net payoffs at equilibrium (σ_1, σ_2) are $(v - c_2(\sigma_2), k - v - c_1(\sigma_1))$, where $v \in \mathbf{R}$ is a fixed number which we call the value of the game G^{-c} .*
3. *The set of net payoffs possible at equilibrium is an axis-parallel rectangle in \mathbf{R}^2 .*

The proof of the above theorem is based on the following simple observation. Consider the k-sum game $H = (N, S, q)$ with the following payoffs:

$$q_1(s_1, s_2) = -q_2(s_1, s_2) = p_1(s_1, s_2) - c_1(s_1) + c_2(s_2) = p_1^{-c}(s_1, s_2) + c_2(s_2).$$

That is to say, Player 1 pays its strategy cost to Player 2 and vice versa. It is easy to verify that,

$$\forall \sigma_1, \sigma'_1 \in \Delta_1, \sigma_2 \in \Delta_2 \quad q_1(\sigma_1, \sigma_2) - q_1(\sigma'_1, \sigma_2) = p_1^{-c}(\sigma_1, \sigma_2) - p_1^{-c}(\sigma'_1, \sigma_2) \quad (1)$$

This means that the relative advantage in switching strategies in games G^{-c} and H are the same. In particular, σ_1 is a best response to σ_2 in G^{-c} if and only if it is in H . A similar equality holds for player 2's payoffs. Note that these conditions imply that the games G^{-c} and H are *strategically equivalent* in the sense defined by Moulin and Vial [17].

Proof of Theorem 1. Since the best responses of G^{-c} and H are the same, the Nash equilibria of the two games are the same. Since H is a k -sum game, its Nash equilibria are exchangeable, and thus we have part 1. (This holds for any game that is strategically equivalent to k -sum.)

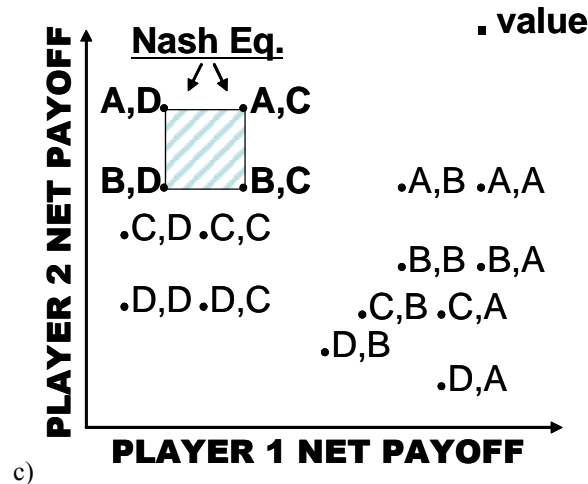
Let v be the value of the game H . The payoffs at any Nash equilibrium (σ_1, σ_2) in H are $(v, -v)$. Since $q_i(\sigma_1, \sigma_2) - p_i^{-c}(\sigma_1, \sigma_2) = c_{-i}(\sigma_{-i})$, we have that $p^{-c}(\sigma_1, \sigma_2) = (v - c_2(\sigma_2), k - v - c_1(\sigma_1))$, as required for part 2.

Finally, the optimal mixed strategies OPT_i of any k -sum game are convex sets. If we look at the achievable costs of the mixed strategies in OPT_i , by the definition of the cost of a mixed strategy, this will be a convex subset of \mathbf{R} , i.e., an interval. By parts 1 and 2, the set of achievable net payoffs at equilibria of G^{-c} are therefore the cross-product of intervals. \square

To illustrate Theorem 1 graphically, let us give a 4×4 example with costs of 1, 2, 3, and 4, respectively:

2a)	A	B	C	D
A	6, 4	5, 5	3, 7	2, 8
B	7, 3	6, 4	4, 6	3, 7
C	7.5, 2.5	6.5, 3.5	4.5, 5.5	3.5, 6.5
D	8.5, 1.5	7, 3	5.5, 4.5	4.5, 5.5

2b)	A (-1)	B (-2)	C (-3)	D (-4)
A (-1)	5, 3	4, 3	2, 4	1, 4
B (-2)	5, 2	4, 2	2, 3	1, 3
C (-3)	4.5, 1.5	3.5, 1.5	1.5, 2.5	0.5, 2.5
D (-4)	4.5, 0.5	3, 1	1.5, 1.5	0.5, 1.5



In table a) Payoffs in 10-sum game G . b) Expected net earnings in G^{-c} . OPT_1 is

any mixture of A and B, and OPT_2 is any mixture of C and D. Each player’s choice of equilibrium strategy affects only the opponent’s net payoff. c) A graphical display of the payoff pairs. The shaded region shows the rectangular set of payoffs achievable at mixed strategy Nash equilibria.

Example 2 illustrates a situation with multiple optimal strategies. Notice that player 1 is completely indifferent between its optimal choices A and B, and player 2 is completely indifferent between C and D. Thus the only question is how kind they would like to be to their opponent. The (A,C) equilibrium is perhaps most natural as it yields the highest payoffs for both parties.

3.1 Learning in repeated two-person k-sum games with strategy costs

Another desirable property of k-sum games is that, in repeated play, natural learning dynamics converge to the set of Nash equilibria. Before we state the analogous conditions for k-sum games with costs, we briefly give a few definitions. A *repeated game* is one in which players chooses a sequence of strategies vectors s^1, s^2, \dots , where each $s^t = (s_1^t, \dots, s_N^t)$ is a strategy vector of some fixed *stage game* $G = (N, S, p)$. Under *perfect monitoring* at each stage, the players know the actions of all players from previous stages. As we shall discuss, it is possible to learn to play without perfect monitoring as well.

Perhaps the most intuitive dynamics are best-response: at each stage, each player selects a best response to the opponent’s previous stage play. Unfortunately, these naive dynamics fails to converge to equilibrium in very simple examples. The *fictitious play* dynamics prescribe, at stage t , selecting any strategy that is a best response to the empirical distribution of opponent’s play during the first $t - 1$ stages. It has been shown that fictitious play converges to equilibrium (of the stage game G) in k-sum games [18].

However, fictitious play requires perfect monitoring. One can learn to play a two-person k-sum game with no knowledge of the payoff table or anything about the other players actions. Using experimentation, the only observations required by each player are its own payoffs in each period (in addition to the number of available actions). So-called *bandit algorithms* [8] must manage the exploration-exploitation tradeoff. The proof of their convergence follows from the fact that they are *no-regret* algorithms. (No-regret algorithms date back to Hannan in the 1950’s [13], but his required perfect monitoring). The *regret* of a player i at stage T is defined to be,

$$\text{regret of } i \text{ at } T = \frac{1}{T} \max_{s_i \in S_i} \sum_{t=1}^T (p_i(s_i, s_{-i}^t) - p_i(s_i^t, s_{-i}^t)),$$

that is, how much better in hindsight player i could have done on the first T stages had it used one fixed strategy the whole time (and had the opponents not changed their strategies). Note that regret can be positive or negative. A no-regret algorithm is one in which each player’s asymptotic regret converges to $(-\infty, 0]$, i.e., is guaranteed to approach 0 or less. It is well-known that no-regret condition in two-person k-sum games implies convergence to equilibrium (see, e.g., [14]). In particular, the pair of

mixed strategies which are the empirical distributions of play over time approaches the set of Nash equilibrium of the stage game.

Inverse-polynomial rates of convergence (that are polynomial also in the size of the game) can be given for such algorithms. Hence no-regret algorithms provide arguably reasonable ways to play a k-sum game of moderate size. Note that in general-sum games, no such dynamics are known. Fortunately, the same algorithm that work for learning in k-sum games seem to work for learning in such games with strategy costs.

Theorem 2. *Fictitious play converges to the set of Nash equilibria of the stage game in a two-person k-sum game with strategy costs, as do no-regret learning dynamics.*

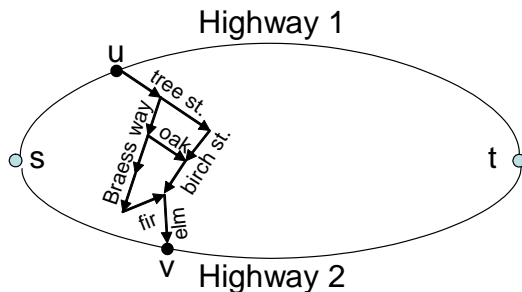
Proof. The proof again follows from equation (1) regarding the game H . Fictitious play dynamics are defined only in terms of best response play. Since G^{-c} and H share the same best responses, fictitious play dynamics are identical for the two games. Since they share the same equilibria and fictitious play converges to equilibria in H , it must converge in G^{-c} as well.

For no-regret algorithms, equation (1) again implies that for any play sequence, the regret of each player i with respect to game G^{-c} is the same as its regret with respect to the game H . Hence, no regret in G^{-c} implies no regret in H . Since no-regret algorithms converge to the set of equilibria in k-sum games, they converge to the set of equilibria in H and therefore in G^{-c} as well. \square

4 Potential games with strategic costs

Let us begin with an example of a *potential game*, called a *congestion games* [19]. There is a fixed directed graph with n nodes and m edges. Commuters $i = 1, 2, \dots, N$ each decide on a route π_i , to take from their home s_i to their work t_i , where s_i and t_i are nodes in the graph. For each edge, uv , let n_{uv} be the number of commuters whose path π_i contains edge uv . Let $f_{uv} : \mathbb{Z} \rightarrow \mathbf{R}$ be a non-negative monotonically increasing congestion function. Player i 's payoff is $-\sum_{uv \in \pi_i} f_{uv}(n_{uv})$, i.e., the negative sum of the congestions on the edges in its path.

Consider a congestion game with 50 players with the graph below, where all players route from s to t , with congestion functions $f_{su}(k) = f_{vt}(k) = k$, $f_{ut}(k) = f_{sv}(k) = 55$, and the congestion is some small $\epsilon > 0$ elsewhere. Without complexity costs, all players will travel from $s \rightarrow u \rightarrow$ (side streets) $\rightarrow v \rightarrow t$, with a commute time just over 100. Increasing the complexity cost of turns forces the traffic to split between the two highway routes $s \rightarrow u \rightarrow t$ and $s \rightarrow v \rightarrow t$ with a decreased commute of 80 for every player.



An N -person normal form game G is said to be a potential game [16] if there is some potential function $\Phi : S_1 \times \dots \times S_N \rightarrow \mathbf{R}$ such that changing a single player's action changes its payoff by the change in the potential function. That is, there exists a single function Φ , such that for all players i and all pure strategy vectors $s, s' \in S_1 \times \dots \times S_N$ that differ only in the i th coordinate,

$$p_i(s) - p_i(s') = \Phi(s) - \Phi(s'). \quad (2)$$

Potential games have appealing learning properties: simple better-reply dynamics converge to pure-strategy Nash equilibria, as do the more sophisticated fictitious-play dynamics described earlier [16]. In our example, this means that if players change their individual paths so as to selfishly reduce the sum of congestions on their path, this will eventually lead to an equilibrium where no one can improve. (This is easy to see because Φ keeps increasing.) The absence of similar learning properties for general games presents a frustrating hole in learning and game theory.

It is clear that the theoretically clean commuting example above misses some realistic considerations. One issue regarding complexity is that most commuters would not be willing to take a very complicated route just to save a short amount of time. To model this, we consider potential games with strategy costs. In our example, this would be a cost associated with every path. For example, suppose the graph represented streets in a given city. We consider a natural strategy complexity cost associated with a route π , say $\lambda(\#\text{turns}(\pi))^2$, where there is a parameter $\lambda \in \mathbf{R}$ and $\#\text{turns}(\pi)$ is defined as the number of times that a commuter has to turn on a route. (To be more precise, say each edge in the graph is annotated with a street name, and a turn is defined to be a pair of consecutive edges in the graph with different street names.) Hence, a best response for player i would minimize:

$$\min_{\pi \text{ from } s_i \text{ to } t_i} (\text{total congestion of } \pi) + \lambda(\#\text{turns}(\pi))^2.$$

The example in Figure 3 is inspired by Braess' paradox, which was used to illustrate the surprising property that closing a road can decrease the commute time for all commuters. In our model, increasing λ sufficiently will cause fewer commuters to travel on the small streets, and will again decrease the commute time for all.

While adding strategy costs to potential games allows for much more flexibility in model design, one might worry that appealing properties of potential games, such as

having pure strategy equilibria and easy learning dynamics, no longer hold. This is not the case. We show that strategic costs fit easily into the potential game framework:

Theorem 3. *For any potential game G and any cost functions c , G^{-c} is also a potential game.*

Proof. Let Φ be a potential function for G . It is straightforward to verify that the G^{-c} admits the following potential function Φ' :

$$\Phi'(s_1, \dots, s_N) = \Phi(s_1, \dots, s_N) - c_1(s_1) - \dots - c_N(s_N). \quad \square$$

5 Additional remarks

Part of the reason that the notion of bounded rationality is so difficult to formalize is that understanding enormous games like chess is a daunting proposition. That is why we have narrowed it down to choosing among a small number of programs.

A game theorist might begin by examining the *complete* payoff table of Figure 1a, which is prohibitively large. Instead of considering only the choices of programs A and B, each player considers all possible chess strategies. In that sense, our payoff table in 1a would be viewed as a reduction of the “real” normal form game. A computer scientist, on the other hand, may consider it reasonable to begin with the existing strategies that one has access to. Regardless of how you view the process, it is clear that for practical purposes players in real life do simplify and analyze “smaller” sets of strategies. Even if the players consider the option of engineering new chess-playing software, this can be viewed as a third strategy in the game, with its own cost and expected payoffs.

Again, when considering small number of strategies, like the two programs above, it may still be difficult to assess the expected payoffs that result when (possibly randomized) strategies play against each other. An additional assumption made throughout the paper is that the players share the same assessments about these expected payoffs. Like other common-knowledge assumptions made in game theory, it would be desirable to weaken this assumption. In the special families of games studied in this paper, and perhaps in additional cases, learning algorithms may be employed to reach equilibrium without knowledge of payoffs.

5.1 Finite automata playing repeated games

There has been a large body of interesting work on repeated games played by finite automata (see [15] for a survey). Much of this work is on achieving cooperation in the classic prisoner’s dilemma game (e.g., [3, 4, 5, 6]). Many of these models can be incorporated into the general model outlined in this paper.

For example, to view the Abreu and Rubinstein model [7] as such, consider the normal form of an infinitely repeated game with discounting, but restricted to strategies that can be described by finite automata (the payoffs in every cell of the payoff table are the discounted sums of the infinite streams of payoffs obtained in the repeated game). Let the cost of a strategy be an increasing function of the number of states it employs.

For Neyman’s model [4], consider the normal form of a finitely repeated game with a known number of repetitions. You may restrict the strategies in this normal form to be only ones with a bounded number of states, as required by Neyman, and assign zero cost to all strategies. Alternatively, you may allow all strategies but assign zero cost to ones that employ number of states below Neyman’s bounds, and an infinite cost to strategies that employ a number of states that exceeds Neyman’s bounds.

The structure of equilibria proven in Theorem 1 applies to all the above models when dealing with repeated k -sum games.

6 Future work

There are very interesting questions to answer about bounded rationality in truly large games that we did not touch upon. For example, consider the following zero-sum factoring game, which has applications to cryptography. Player 1 chooses an n -bit number and sends it to Player 2, who attempts to find its prime factorization. If Player 2 is correct, he is paid 1 by Player 1, otherwise he pays 1 to Player 1. A pure strategy for Player 1 would be outputting a single n -bit number. A pure strategy for Player 2 would be any factoring program, described by a circuit that takes as input an n -bit number and attempts to output a representation of its prime factorization. The complexity of such a strategy would be an increasing function of the number of gates in the circuit. Ignoring complexity costs, the game is a trivial win for Player 2. However, with appropriate complexity costs² the game should be essentially a win for Player 1, who can easily output a large random number that Player 2 cannot factor with a small circuit. It would be interesting to see how such games fit into the model.

Another direction regards an elegant line of work on learning to play correlated equilibria by repeated play [12]. It would be interesting to consider how strategy costs affect correlated equilibria. Another question that arises in the congestion example is how the strategy cost affects the so-called “price of anarchy” [20] in the congestion game, which is a measure of social inefficiency. It would also be interesting to see if existing bounds [21] still apply.

There is no question that a good model of bounded rationality could have far-reaching applications across a broad range of fields, just as other game-theoretic notions of equilibrium have impacted a great many subjects. Finding such a model seems like a problem perfectly suited for the intersection of game theory and computer science.

References

[1]

[2] H. Simon. *The sciences of the artificial*. MIT Press, Cambridge, MA, 1969.

²It is believed that the problem of factoring an n -bit number has computational costs that are prohibitively large as n grows without bound.

- [3] E. Ben-Porath. Repeated games with finite automata, *Journal of Economic Theory* 59: 17–32, 1993.
- [4] A. Neyman. Bounded Complexity Justifies Cooperation in the Finitely Repeated Prisoner’s Dilemma. *Economic Letters*, 19: 227–229, 1985.
- [5] A. Rubenstein. Finite automata play the repeated prisoner’s dilemma. *Journal of Economic Theory*, 39:83–96, 1986.
- [6] C. Papadimitriou, M. Yannakakis: On complexity as bounded rationality. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 726–733, 1994.
- [7] D. Abreu and A. Rubenstein. The Structure of Nash Equilibrium in Repeated Games with Finite Automata. *Econometrica* 56:1259-1281, 1988.
- [8] P. Auer, N. Cesa-Bianchi, Y. Freund, R. Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32(1):48-77, 2002.
- [9] X. Chen, X. Deng, and S. Teng. Computing Nash Equilibria: Approximation and smoothed complexity. Electronic Colloquium on Computational Complexity Report TR06-023, 2006.
- [10] K. Daskalakis, P. Goldberg, C. Papadimitriou. The complexity of computing a Nash equilibrium. Electronic Colloquium on Computational Complexity Report TR05-115, 2005.
- [11] C. Ewerhart. Chess-like Games Are Dominance Solvable in at Most Two Steps. *Games and Economic Behavior*, 33:41-47, 2000.
- [12] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40-55, 1997.
- [13] J. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pp. 97–139. Princeton University Press, 1957.
- [14] S. Hart and A. Mas-Colell. A General Class of Adaptive Strategies. *Journal of Economic Theory* 98(1):26–54, 2001.
- [15] E. Kalai. Bounded rationality and strategic complexity in repeated games. In T. Ichiishi, A. Neyman, and Y. Tauman, editors, *Game Theory and Applications*, pp. 131–157. Academic Press, San Diego, 1990.
- [16] D. Monderer, L. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
- [17] H. Moulin and P. Vial. Strategically Zero Sum Games: the Class of Games Whose Completely Mixed Equilibria Cannot Be Improved Upon. *International Journal of Game Theory*, 7:201–221, 1978.
- [18] J. Robinson, An iterative method of solving a game, *Ann. Math.* 54:296–301, 1951.

- [19] R. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [20] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 404–413, 1999.
- [21] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM* 49(2): 236–259, 2002.