Algorithmic Rationality: Adding Cost of Computation to Game Theory

JOSEPH Y. HALPERN and RAFAEL PASS
Cornell University

We summarize our work on a general game-theoretic framework for reasoning about strategic agents performing possibly costly computation. In this framework, many traditional game-theoretic results (such as the existence of a Nash equilibrium) no longer hold. Nevertheless, we can use the framework to provide psychologically appealing explanations to observed behavior in well-studied games (such as finitely repeated prisoner's dilemma and rock-paper-scissors).

Categories and Subject Descriptors: F.m [**Theory of Computation**]: Miscellaneous; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Economics, Theory

Additional Key Words and Phrases: Nash equilibrium, costly computation

1. INTRODUCTION

Consider the following game. You are given a random odd n-bit number x and you are supposed decide whether x is prime or composite. If you guess correctly you receive \$2, if you guess incorrectly you instead have to pay a penalty of \$1000. Additionally you have the choice of "playing safe" by giving up, in which case you receive \$1. In traditional game theory, computation is considered "costless"; in other words, players are allowed to perform an unbounded amount of computation without it affecting their utility. Traditional game theory suggests that you should compute whether x is prime or composite and output the correct answer; this is the only Nash equilibrium of the one-person game, no matter what n (the size of the prime) is. Although for small n this seems reasonable, when n grows larger most people would probably decide to "play safe"—as eventually the cost of computing the answer (e.g., by buying powerful enough computers) outweighs the possible gain of \$1.

The importance of considering such computational issues in game theory has been recognized since at least the work of Simon [1955]. There have been a number of attempts to capture various aspects of computation. Perhaps most relevant to our work is the work of Rubinstein [1986], who assumed that players choose a finite automaton to play the game rather than choosing a strategy directly; a player's utility depends both on the move made by the automaton and the complexity of the automaton (identified with the number of states of the automaton). Intuitively, automata that use more states are seen as representing more complicated procedures. (See [Kalai 1990] for an overview of the work in this area in the 1980s, and [Ben-Sasson et al. 2007] for more recent work.)

As in Rubinstein's work, we view players as choosing a machine, but for us the machine is a Turing machine, rather than a finite automaton. We associate a complexity, not just with a machine, but with the machine and its input. The complexity could represent the running time of or space used by the machine on that input. The complexity can also be used to capture the complexity of the machine itself (e.g., the number of states, as in Rubinstein's case) or to model the cost of searching for a new strategy to replace one that the player already has. For example, if a mechanism designer recommends that player i use a particular strategy (machine) M, then there is a cost for searching for a better strategy; switching to another strategy may also entail a psychological cost. By allowing the complexity to depend on the machine *and* the input, we can deal with the fact that machines run much longer on some inputs than on others. A player's utility depends both on the actions chosen by all the players' machines and the complexity of these machines.

In this setting, we can define Nash equilibrium in the obvious way. However, as we show by a simple example (rock-paper-scissors), if randomizing is costly, a Nash equilibrium may not always exist. Other standard results in game theory, such as the *revelation principle* (which, roughly speaking, says that there is always an equilibrium where players truthfully report their types, i.e., their private information [Myerson 1979; Forges 1986]) also do not hold. We view this as a feature. We believe that taking computation into account should force us to rethink a number of basic notions.

On the positive side, we show that in a precise sense, if randomizing is free, then (under minimal assumptions) a Nash euqilibrium is guaranteed to exist. Moreover, as we show by example, despite the fact that Nash equilibrium may not always exist, taking computation into account leads to Nash equilibria that give insight into and explain behavior in a number of games, such as cooperation in finitely repeated prisoner's dilemma, and biases in information processing, in a psychologically appealing way. Indeed, as shown by our results, many concerns expressed by the emerging field of *behavioral economics* (pioneered by Kahneman and Tversky [1981]) can be accounted for by simple assumptions about players' cost of computation, without resorting to ad hoc cognitive or psychological models.

In the remainder of this note, we briefly sketch the definition and give a few examples of its use. We refer the reader to [Halpern and Pass 2008; 2009; 2010a; 2010b] for further details.

2. THE DEFINITION AND SOME EXAMPLES

Recall that in a standard *Bayesian game*, each agent i has a type t_i chosen from a type space T_i. An agent's type can be viewed as describing the agent's private information. It is assumed that there is a commonly known distribution Pr on the type space $T_1 \times \ldots \times T_n$. Each agent i then chooses a strategy σ_i which is a function from T_i to A_i, the set of possible actions for agent i; intuitively, what agent i does depends on her type. Then agent i's utility u_i depends on the the profile of types $\vec{t}$ and the profile of actions chosen $\vec{a}$. Thus, a Bayesian game is characterized by a tuple $([m], T, A, \Pr, \vec{u})$, where $[m]$ is the set of players, T is the type space, $A = A_1 \times \ldots \times A_n$ is the set of action profiles, and $\vec{u}$ is the utility function profile, where $u_i : T \times A \to I\!R$.

In a *Bayesian machine game*, we still have each player i chooses a Turing machine (TM) M_i from some set $\mathcal{M}_i$ of Turing machines. The output $M_i(t_i)$ that results from running M_i on input t_i, player i's type, is an action in A_i. (We assume that there is a special action ω in A_i that is played if M_i does not halt on input t_i.) In addition, each player i has a *complexity function* $\mathcal{C}_i$ that associates with each TM M_i and input t_i its *complexity*, which is just a natural number. As we said earlier, the complexity could represent, among other things, the running time of M_i on t_i, the space used, the number of states in M_i, or the cost of "finding" strategy M_i. Player i's utility depends on the the profile $\vec{t}$, the action profile $(M_1(t_1), \ldots, M_n(t_n))$ (where $\{1, \ldots, n\}$ is the set of players), and the tuple of complexities $(\mathcal{C}_1(M_1, t_1), \ldots, \mathcal{C}_n(M_n, t_n))$.[1] The fact that i's utility can depend on the whole tuple of complexitities allows for the possibility that, for example, i gains high utility by having a faster running time than j.

In this setting, a strategy associates with each type a TM. Then i's expected utility if the profile of TM's $(\sigma_1, \ldots, \sigma_n)$ is chosen is just

$$\sum_{\vec{t} \in T} \Pr(\vec{t}) u_i((\sigma_1(t_1))(t_1), \ldots, (\sigma_n(t_n))(t_n)), (\mathcal{C}_1(\sigma_1(t_1), t_1), \ldots, \mathcal{C}_n(\sigma_n(t_n), t_n))).$$

Note that $\sigma_i(t_i)(t_i)$ is the action output by the TM $\sigma(t_i)$. The definition of Nash equilibrium is just as usual: a tuple of strategies $(\sigma_1, \ldots, \sigma_n)$ is a Nash equilibrium if no player i can improve her expected utility by deviating to a different strategy.

The following example shows that a Nash equilibrium may not always exist.

EXAMPLE 2.1 *Rock-paper-scissors.* Consider the 2-player Bayesian game of roshambo (rock-paper-scissors). Here the type space has size 1 (the players have no private information). We model playing rock, paper, and scissors as playing 0, 1, and 2, respectively. The payoff to player 1 of the outcome (i, j) is 1 if $i = j \oplus 1$ (where $\oplus$ denotes addition mod 3), -1 if $j = i \oplus 1$, and 0 if $i = j$. Player 2's playoffs are the negative of those of player 1; the game is a zero-sum game. As is well known, the unique Nash equilibrium of this game has the players randomizing uniformly between 0, 1, and 2.

Now consider a machine game version of roshambo. Suppose that we take the complexity of a deterministic strategy to be 0, and the complexity of a strategy that uses randomization to be 1, and take player i's utility to be his payoff in the underlying Bayesian game minus the complexity of his strategy. Intuitively, programs involving randomization are more complicated than those that do not randomize. With this utility function, it is easy to see that there is no Nash equilibrium. For suppose that (M_1, M_2) is an equilibrium. If M_1 uses randomization, then 1 can do better by playing the deterministic strategy $j \oplus 1$, where j is the action that gets the highest probability according to M_2 (or is the deterministic choice of player 2 if M_2 does not use randomization). Similarly, M_2 cannot use randomization. But it is well known (and easy to check) that there is no equilibrium for roshambo with deterministic strategies. $\square$

Charging for randomization does not seem so unreasonable. It is well known that people have difficulty simulating randomization; we can think of the cost for

[1]We are simplifying a bit here. A TM can randomize, so also gets as input a random string. The complexity can thus depend on the TM chosen, the input, and the random string.

randomizing as capturing this difficulty. Interestingly, there are roshambo tournaments (indeed, even a Rock Paper Scissors World Championship), and books written on roshambo strategies [Walker and Walker 2004]. Championship players are clearly not randomizing uniformly (they could not hope to get a higher payoff than an opponent by randomizing). Our framework provides a psychologically plausible account of this lack of randomization.

Example 2.1 shows that a Nash equilibrium may not always exist. In [Halpern and Pass 2008], we show that, in a sense, the fact that we charge for randomization is the reason for this. Among other things, we show that if the cost of a randomized strategy is the convex combination of the cost of the pure strategies involved, and the game itself is computable in a precise show, then a Nash equilibrium is guaranteed to exist.

The remaining examples give other instances where thinking computationally can give psychologically plausible explanations for behaviors in games.

Example 2.2. Consider finitely repeated prisoner's dilemma (FRPD), where prisoner's dilemma is played for some fixed number N of rounds. As is well known, the only Nash equilibrium is to always defect; this can be seen by a backwards induction argument. This seems quite unreasonable. And, indeed, in experiments, people do not always defect [Axelrod 1984]. In fact, quite often they cooperate throughout the game. There have been many attempts to explain cooperation in FRPD in the literature; see, for example, [Kreps et al. 1982; Neyman 1985; Papadimitriou and Yannakakis 1994]. In particular, [Neyman 1985; Papadimitriou and Yannakakis 1994] demonstrate that if players are restricted to using a finite automaton with bounded complexity, then there exist equilibria that allow for cooperation. However, the strategies used in those equilibria are quite complex, and require the use of large automata. By using our framework, we can provide a straightforward explanation.

Consider the *tit for tat* strategy, which proceeds as follows: a player cooperates at the first round, and then at round $m + 1$, does whatever his opponent did at round m. Thus, if the opponent cooperated at the previous round, then you reward him by continuing to cooperate; if he defected at the previous round, you punish him by defecting. If both players play tit for tat, then they cooperate throughout the game. The best response to tit for tat is easily seen to be cooperating for the first $N - 1$ rounds, then defecting at the last round. But this requires counting up to N, which in turn requires some use of computation. Suppose that the cost of this computation is more than the gain from defection. In that case, the best response to tit for tat is tit for tat. □

Example 2.3 *Biases in information processing.* Psychologists have observed many systematic biases in the way that individuals update their beliefs as new information is received (see [Rabin 1998] for a survey). In particular, a "first-impressions-matter" bias has been observed: individuals put too much weight on initial signals and less weight on later signals. As they become more convinced that their beliefs are correct, many individuals even seem to simply ignore all information once they reach a confidence threshold. Several papers in behavioral economics have focused on identifying and modeling some of these biases (see, e.g., [Rabin 1998] and the references therein, [Mullainathan 2002], and [Rabin and

Schrag 1999]). In particular, Mullainathan [2002] makes a potential connection between memory and biased information processing, in a model which makes several explicit (psychology-based) assumptions on the memory process (e.g., that the agent's ability to recall a past event depends on how often he has recalled the event in the past.) More recently, Wilson [2002] has presented an elegant model of bounded rationality (where agents are described by finite automata) that (among other things) can explain why agents eventually choose to ignore new information; her analysis, however, is very complex and holds only in the limit (specifically, in the limit as the probability ν that a given round is the last round goes to 0). As we now show, the first-impression-matters bias can be easily explained if we assume that here is a small cost for "absorbing" new information.

Consider the following simple game (which is very similar to one studied in [Mullainathan 2002; Wilson 2002]). The state of nature is a bit b which is 1 with probability 1/2. An agent receives as his type a sequence of independent samples $s_1, s_2, \ldots, s_n$ where $s_i = b$ with probability $\rho > 1/2$. The samples corresponds to signals the agents receive about b. An agent is supposed to output a guess b' for the bit b. If the guess is correct, he receives $1 - mc$ as utility, and $-mc$ otherwise, where m is the number of bits of the type he read, and c is the cost of reading a single bit (c should be thought of the cost of absorbing/interpreting information). It seems reasonable to assume that $c > 0$; signals usually require some effort to decode (such as reading a newspaper article, or attentively watching a movie). If $c > 0$, it easily follows by the Chernoff bound that after reading a certain (fixed) number of signals $s_1, \ldots, s_i$, the agents will have a sufficiently good estimate of ρ that the marginal cost of reading one extra signal s_{i+1} is higher than the expected gain of finding out the value of s_{i+1}. That is, after processing a certain number of signals, agents will eventually disregard all future signals and base their output guess only on the initial sequence. In fact, doing so stricly dominates reading more signals. □

3. DISCUSSION

We have defined a general approach to taking computation into account in game theory that subsumes previous approaches. This opens the door to a number of exciting research directions; see [Halpern and Pass 2008] for a list. Perhaps the key issue is that, in the framework described above, we have assumed that the agents understand the costs associated with each Turing machine. That is, they do not have to do any "exploration" to compute the costs. In [Halpern and Pass 2010b], we model uncertainty regarding complexity by letting the complexity function take also the state of nature as input. However, even if we do this, we are assuming that agents can compute the probability of (or, at least, are willing to assign a probability to) events like "TM M will halt in 10,000 steps" or "the output of TM M solves the problem I am interested in on this input". But calculating such probabilities itself involves computation, which might be costly. Similarly, we do not charge the players for computing which machine is the best one to use in a given setting, or for computing the utility of a given machine profile. It would be relatively straightforward to extend our framework so that the TMs computed probabilities and utilities, as well as actions; this would allow us to "charge" for the cost of computing probabilities and utilities. However, once we do this, we need to think about what counts as an "optimal" decision if a decision maker does not have a probability and utility, or has a probability only on a coarse space. A related problem is that we have assumed that the players have all options available "for free". That is, the players choose among a set of TMs which is given *ex ante*, and does not change. But, in practice, it may be difficult to generate alternatives. (Think of a chess player trying to generate interesting lines of play, for example.) Again, we can imagine charging for generating such alternatives. We are currently exploring charging for computation costs in extensive-form games in ways that take these concerns into account, using ideas related to awareness in games (see, e.g., [Heifetz et al. 2007; Halpern and Rêgo 2006]).

Acknowledgments:

Halpern supported in part by NSF grants ITR-0325453, IIS-0534064, IIS-0812045, and IIS-0911036, and by AFOSR grants FA9550-08-1-0438 and FA9550-09-1-0266, and ARO grant W911NF-09-1-0281. Pass supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197, and BSF Grant 2006317.

REFERENCES

Axelrod, R. 1984. *The Evolution of Cooperation*. Basic Books, New York.

Ben-Sasson, E., Kalai, A., and Kalai, E. 2007. An approach to bounded rationality. In *Advances in Neural Information Processing Systems 19 (Proc. of NIPS 2006)*. 145–152.

Forges, F. 1986. An approach to communication equilibria. *Econometrica 54,* 6, 1375–85.

Halpern, J. Y. and Pass, R. 2008. Algorithmic rationality: Game theory with costly computation. Unpublished manuscript.

Halpern, J. Y. and Pass, R. 2009. Sequential equilibrium and perfect equilibrium in games of imperfect recall. Unpublished manuscript.

Halpern, J. Y. and Pass, R. 2010a. Game theory with costly computation. In *Proc. First Symposium on Innovations in Computer Science*.

Halpern, J. Y. and Pass, R. 2010b. I don't want to think about it now: Decision theory with costly computation. In *Principles of Knowledge Representation and Reasoning: Proc. Twelfth International Conference (KR '10)*.

Halpern, J. Y. and Rêgo, L. C. 2006. Extensive games with possibly unaware players. In *Proc. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. 744–751. Full version available at arxiv.org/abs/0704.2014.

Heifetz, A., Meier, M., and Schipper, B. 2007. Unawareness, beliefs and games. In *Theoretical Aspects of Rationality and Knowledge: Proc. Eleventh Conference (TARK 2007)*. 183–192. Full paper available at www.econ.ucdavis.edu/faculty/schipper/unawprob.pdf.

Kalai, E. 1990. Bounded rationality and strategic complexity in repeated games. In *Game Theory and Applications*. Academic Press, San Diego, 131–157.

Kreps, D., Milgrom, P., Roberts, J., and Wilson, R. 1982. Rational cooperation in finitely repeated prisoners' dilemma. *Journal of Economic Theory 27,* 2, 245–252.

Mullainathan, S. 2002. A memory-based model of bounded rationality. *Quarterly Journal of Economics 117,* 3, 735–774.

Myerson, R. 1979. Incentive-compatibility and the bargaining problem. *Econometrica 47*, 61–73.

Neyman, A. 1985. Bounded complexity justifies cooperation in finitely repated prisoner's dilemma. *Economic Letters 19*, 227–229.

Papadimitriou, C. H. and Yannakakis, M. 1994. On complexity as bounded rationality. In *Proc. 26th ACM Symposium on Theory of Computing*. 726–733.

RABIN, M. 1998. Psychology and economics. *Journal of Economic Literature XXXVI*, 11–46.

RABIN, M. AND SCHRAG, J. 1999. First impressions matter: A model of confirmatory bias. *Quaterly Journal of Economics 114,* 1, 37–82.

RUBINSTEIN, A. 1986. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory 39*, 83–96.

SIMON, H. A. 1955. A behavioral model of rational choice. *Quarterly Journal of Economics 49*, 99–118.

TVERSKY, A. AND KAHNEMAN, D. 1981. The framing of decisions and the psychology of choice. *Science 211*, 453–58.

WALKER, G. AND WALKER, D. 2004. *The Official Rock Paper Scissors Strategy Guide*. Simon & Schuster, Inc., New York.

WILSON, A. 2002. Bounded memory and biases in information processing. Manuscript.