Computer Science and Game Theory

Yoav Shoham
Computer Science Department
Stanford University

April 18, 2008

Abstract

Game theory has been playing an increasingly visible role in computer science, in areas as diverse as artificial intelligence, theory, and distributed systems, among others. I take stock of where most of the action has been in the past decade or so, and suggest that going forward, the most dramatic interaction between computer science and game theory could be around what might be called game theory pragmatics.

1 Introduction

Game theory has influenced many fields, from economics (historically its initial focus) to political science to biology, and many others. In recent years its presence in computer science has become impossible to ignore. It features routinely in the leading conferences and journals of artificial intelligence (AI), theory, certainly electronic commerce, as well as in networking and other areas of computer science. There are several reasons for this. One is application pull; the Internet calls for analysis and design of systems that span multiple entities with diverging information and interests. Game theory, for all its limitations, is by far the most developed theory of such interactions. Another is technology push; the mathematics and scientific mindset of game theory are similar to those of many computer scientists. Indeed, it is interesting to note that modern computer science and modern game theory in large measure originated at the same place and time, namely at Princeton under the leadership of John von Neumann.[1]

In this paper I would like to do two things. First, to identify the main areas of interaction between computer science and game theory so far. Second, to point to where the most interesting interaction yet may lie, an area which is still relatively under-explored.

The first part aims to be an unbiased survey, but since this area – the interaction between CS and GT – is so dynamic, two caveats are in order. First, given the diversity of work in the area, it is impossible to avoid bias altogether. Ten researchers in the area would probably write ten different types of survey. Indeed, several already have (see below). Second, in this very short survey I cannot possibly do justice to all the work taking place in the area. I will

[1] I thank Moshe Tennenholtz for this observation, which is especially true of GT and AI.

try to compensate for both limitations in two ways. First, I will attempt to provide a balanced set of initial pointers into the different subareas of the CS-GT interaction, without regard to the amount or nature of work that takes place in each subarea. Second, I will point the reader to other relevant surveys, each having its own take on things.

The second part is decidedly subjective, but still meant to be relevant broadly to both computer scientists and game theorists interested in the interaction between the disciplines.

2 Lessons from Kalai (1995)

My departure point will be a thirteen-year-old survey paper by E. Kalai [16]. Written by a game theorist with algorithmic and optimization sensibilities, and geared primarily towards computer scientists, the paper took stock of the interaction among game theory, operations research and computer science at the time.[2] The paper points to the following areas:

1. Graphs in games.
2. The complexity of solving a game.
3. Multi-person operations research.
4. The complexity of playing a game.
5. Modelling bounded rationality.

The reason I start with this paper, beside the fact that it's interesting to start with the perspective of a non-computer-scientist, is the comparison with current CS-GT interaction, since both the match and the mismatch are instructive. When one looks at the interaction between CS and GT taking place today, broadly speaking one can identify the following foci of action:

(a) Compact game representations.

(b) Complexity of, and algorithms for, computing solution concepts.

(c) Algorithmic aspects of mechanism design.

(d) Game theoretic analysis inspired by specific applications.

(e) Multiagent learning.

(f) Logics of knowledge and belief and other logical aspects of games.

The crude mapping between this list and Kalai's is as follows:

[2] This current survey originated in a presentation made at a festschrift in honor of E. Kalai in December, 2007.

1995		2008
1.	———	(a)
2.	———	(b)
3.	———	(c)
4.		(d)
5.		(e)
		(f)

I will first discuss the areas that match up ($1-a, 2-b, 3-c$), then the currently active areas that were not discussed over by Kalai (d, e, f), and finish with the orphans on the other side, ($4, 5$), discussed by Kalai but not yet been picked up as vigorously as they might.

There has been substantial work on compact and otherwise specialized game representations. Some of them are indeed graph-based, for example graphical games [18], local-effect games [21], MAIDS [19], and Game networks [20]. The graph-based representations extend also to coalition game theory [7]. But specialized representations exist that are not graph based, for example multi-attribute based [5] and logic based [15]. I believe this area is ripe for additional work, for example one that has the strategy space of agents described using constructs of programming languages.

The complexity of computing a sample Nash equilibrium (as well as other solution concepts) has indeed been the focus of much interest in CS, especially within the theory community. A new complexity class – PPAD – was proposed to handle problems for which a solution is known to exist [27], the computation of a sample Nash equilibrium was shown to be complete for this class [2], and the problem of computing Nash equilibria with specific properties was shown to be NP-hard [10, 4]. At the same time, algorithms – some quite sophisticated, and all exponential in the worst case – have been proposed to compute Nash equilibria [41, 11]. Somewhat surprisingly, it was recently shown experimentally that a relatively simple search algorithm significantly outperforms more sophisticated algorithms [31]. This is a very active area that promises many additional results.

The third match is somewhat less tight than the first two. There are at least two kinds of optimization one could speak about in a game theoretic setting. The first is computing a best response to a fixed decision by the other agents; this is of course the quintessential single-agent optimization problem of operations research and AI, among others. The second is the optimization by the designer of the mechanism with an eye towards inducing games with desirable equilibria. This so-called *mechanism design* has been the focus of much work in computer science. One reason is the interesting interaction between traditional CS problems such as optimization and approximation, and traditional mechanism design issues such as incentive compatibility, individual rationality, and social welfare maximization. Good examples include the interaction between the Vickrey-Clarke-Groves (VCG) mechanism and shortest-path computation [26], and the literature on combinatorial auctions [6] which combines a weighted-set-packing-like NP-hard optimization problem with incentive issues. In this connection one might also mention the interplay between mechanism design and cryptography. They are both in the business of controlled dissemination of information, but are different in significant ways. First, they are dual in the following sense: Mechanism design attempts to *force* the *revelation* of information, while cryptography attempts to *allow* its *hiding*. Second, traditionally they embody quite different models of paranoia; game theory assumes an even-keeled expected utility maximization on the part of all agents, while cryptography is more simple minded: it assumes that "good" agents act as instructed, while "bad" agents are maximally harmful. Recent work, however, has begun to bridge these gaps.

This third category blends into the fourth one, which is research motivated by specific applications that have emerged in the past decade. For example, the domain of networking has given rise to a literature on so-called "price of anarchy" (which captures the inefficiency of equilibria in such domains), games of routing, networking-formation games, and peer-to-peer networks. Other domains include sponsored search auctions, information markets, and reputation systems. This combination of third and fourth areas is arguably the most active area today at the interface of CS and GT, and many aspects of it are covered in [25], an extensive edited collection of surveys. The popularity of this area is perhaps not surprising. The relevance of specific applications speak for itself (although arguments remain about whether the traditional game theoretic analysis is an appropriate one). More generally, it is not surprising that mechanism design struck a nerve in CS, since much of the focus in CS is on the design of algorithms and protocols, and mechanism design is the one area within GT that adopts such a design stance.

The fifth active today is multiagent learning, also called *interactive learning* in the game theory literature.[3] Multiagent learning, long a major focus within game theory, has been discovered with something of a vengeance in computer science and in particular AI, witness special issues devoted to it in the Journal of Artificial Intelligence [39] and the Machine Learning Journal [12]. For computer science, the move from single-agent learning to multi-agent learning is interesting not only because it calls for new solutions, but because the very questions change. When multiple agents learn concurrently one cannot distinguish between learning and teaching, and the question of "optimal" learning is no longer well defined (just as the more general notion of an "optimal policy" ceases to be meaningful when one moves to the multiagent setting). For a discussion of this see the aforementioned [39].

The sixth and final major area of focus, also one not discussed in [16], is what is called in game theory *interactive epistemology*, and in computer science usually simply reasoning about knowledge and belief. Starting in the mid-80s, this area was for a while the most active focus of interaction between computer science (including distributed systems, AI and theory) and game theory. Beside game theory, it made deep ties with philosophy and mathematical logic, and culminated in the seminal [8].[4] It is interesting to speculate why this area was omitted from Kalai's list even although it predates his paper by a decade, and why today it is not as active as the other areas. I think this is because the subject matter is more foundational, primarily non-algorithmic, and appeals to a smaller sliver of the two communities. Be that as it may, it remains a key area of interaction between the two fields, as well as logic and other areas.

These six areas are where most of the action has been in the past years, but by listing only them, and being brief about each one, I have by necessity glossed over some important areas. The references compensate for this to an extent. In addition, the reader is referred

[3]Kalai's omission of this area is ironic in light of the fact that he co-authored one of the area's seminal papers.

[4]That book focused on static aspects of knowledge and belief, which, notwithstanding the substantial computer science credentials of the authors, raises an interesting contrast between the computer science and game theory literature on knowledge and belief. In game theory static theories are indeed the primary focus, whereas in computer science – in particular, in database theory and artificial intelligence – belief revision and other dynamic theories [30] (including the entire mini-industry of nonmonotonic logics [9]) play an equal if not greater role. Indeed, recent work at the interface of logic and game theory [37] extends the static treatment of [8] in a dynamic direction.

to the following additional surveys, all by computer scientists, each with a slightly different slant, and most of which go in considerably more detail into some of the topics.

- The earliest relevant survey is probably [22]. Geared primarily towards game theorists, this 58-page survey has deep coverage of game theoretic aspects of distributed systems, fault-tolerant computing, and cryptography, and also touches on other topics, including computation of game theoretic concepts, games and logic, and others.

- [29], geared towards computer scientists, is a concise, 5-page paper summarizing the main complexity and algorithmic issues at the interface of CS and GT circa 2001.

- The 21-page [13] is similar to [22] in that it is geared towards game theorists and its main focus is distributed systems, but it is more current than the latter given the decade separating the two. [13] later evolved into [14], a 17-page survey with an abbreviated discussion of distributed computing, but with additional material on complexity considerations, price of anarchy, mediators, and other topics.

- The 30-page [32] is a detailed survey of a specific topic, namely the complexity of computing a sample Nash equilibrium. Geared mostly towards economists, it includes ample background material on relevant concepts from complexity theory.

The material discussed so far features prominently in computer science conferences and journals, and is also beginning to find its way into textbooks (cf. [25, 35]). These areas will undoubtedly continue to flourish, but I want to turn the attention to the two closely related areas – 4 and 5 – listed by Kalai that have *not* been looked as closely by the community at large, and CS in particular. I want to do this for two reasons: I believe that they are critical to the future success of game theory, and I believe that CS can play an important role in them. They both have to do with incorporating practical considerations within the model of rationality inherent in game theory. To repeat the caveat mentioned earlier: Unlike the material so far, the remaining material is future directed, speculative, and subjective.

3 Lessons from Linguistics

The field of linguistics distinguishes among syntax, semantics, and pragmatics. Syntax defines the form of language, semantics its meaning, and pragmatics its use. While the three interact in important ways, the distinction has proved very useful. I believe that game theory may do well to make similar distinctions, and that CS can help in the process. Just as in the case in linguistics, it is unlikely that game theory pragmatics will yield to unified clean theories as do syntax and semantics. But by the same token I expect game theory pragmatics to be as important to applying game theory in practice as language pragmatics are to analyzing full human discourse, or understanding language by computers.

The distinction between the syntax and semantics of games is I think quite important. I feel that some of the disputes within game theory regarding the primacy of different game representations (for example, the strategic and extensive forms), suffer from the lack of this distinction. It is perhaps presumptuous for CS to intrude on this debate, except insofar as it lends logical insights, since I do think that logic is a useful lens through which to look at these issues (cf. [38]). But perhaps indeed this is more the role of mathematical logic than of CS *per se*.

Where CS can truly lead the way is I think on the pragmatics of game theory. Game theory as we know it embodies radical idealizations, which include the infinite capacity of agents to reason and the infinite mutually-recursive modelling of agents. Backing off from these strong assumptions has proven challenging. A fairly thin strand of work under the heading of "bounded rationality" studies games played by automata (cf. [33]). This is an important area of research, and sometimes makes deep connections between the two fields. Early results, for example, showed that one of the well known pesky facts in game theory – namely that constant 'defection' is the only subgame-perfect equilibrium in the finitely-repeated prisoner's dilemma game – ceases to hold true if the players are finite automata with sufficiently few states [24, 28]. A more recent result shows that when players in a game are computer programs, one obtains phenomena akin to the Folk Theorem for repeated games [36].

This connection between theoretical models of computation and game theory is I believe quite important and beautiful, but a fairly narrow interpretation of the term 'bounded rationality'. The term should perhaps should be reserved to describe a much broader research agenda, an agenda which may encourage more radical departures from the traditional view in game theory. Let me mention two directions that I think would be profitable (and hard) to pursue under this broader umbrella.

When one takes seriously the notion of agent's limited reasoning powers, it is not only some of the answers that begin to change; the questions themselves come into question. Consider the basic workhorses of game theory – the Nash equilibrium, and its various variants. They have so far served as the very basic analysis tool of strategic interactions. Questioning the role of equilibrium analysis will be viewed by some in GT as act of heresy, but real life suggests that perhaps we have no choice. For example, in the trading agent competition (TAC), Nash equilibrium of the game did not play a role in almost any participating program [42], and this is certainly true of the more established chess and checkers competitions. It is premature to write off the Nash equilibrium as irrelevant. For example, one program (see chapter 8 of [42]) did in fact make use of what can be viewed as approximate empirical NE. Another striking example is the computation of equilibria in a simplified game tree by a top scoring program in a poker competition [43]. It could be argued that maxmin strategies, which coincide with equilibrium strategies in zero-sum games, do play an important pragmatic role. But certainly computation of either maxmin or equilibrium strategies in competitions has been the exception to the rule. The more common experience is that one spends the vast majority of the effort on traditional AI problems such as designing a good heuristic function, searching, and planning. Only a little time – albeit, important time – is spent reasoning about the opponent. The impact of such pragmatic considerations on game theory can be dramatic. Rather than start from very strong idealizing assumptions and awkwardly try to back off from them, it may prove more useful and/or accurate to start from assumptions of rather limited reasoning and mutual modeling, and judiciously add those as is appropriate for the situation being modeled. What exact incremental modeling approach will win out is yet to be seen, but the payoff for both CS and game theory can be substantial.

The second direction is radical in a different way. Game theory adopts a fairly terse vocabulary, inheriting it from decision theory and the foundations of statistics.[5] In particular, agents have "strategies" which have minimal structure, and motivations which are encapsulated in a simple real-valued utility function (which in fact carries even less information than is suggested by the use of numbers, since the theory is unchanged by any positive affine transformation of the numbers). In real life, and in computer programs attempting to behave intelligently, we find use for a much broader vocabulary. Agents are *able* to take certain actions and not others, have *desires*, *goals* and *intentions* (the belief-desire-intention combination giving rise to the pun 'BDI agent architecture'), make *plans*, and so on. Apparently these abstract notions are useful to both effect intelligent behavior and reason about it. Philosophers have written about it (e.g., [1]), and there have been attempts – albeit preliminary ones – to formalize these intuition (starting with [3]). Some in AI have advocated embracing an even broader vocabulary of emotions (e.g., the recent provocative albeit informal [23]). Is game theory missing out by not considering these concepts?

4 Concluding Remarks

Science operates at many levels. For some, it is sufficient that scientific theories be clever, beautiful and inspirational. Others require that any science eventually make contact with compelling applications and be subjected to empirical evaluation. Without personally weighing in on this emotional debate, I note that in his presidential address at the quad-annual congress of the International Game Theory Society [17], Kalai reprised the three stages of any science as discussed by von Neumann and Morgenstern:

> [W]hat is important is the gradual development of a theory, based on a careful analysis of the ordinary everyday interpretation of economic facts. The theory finally developed must be mathematically rigorous and conceptually general. Its first applications are necessarily to elementary problems where the result has never been in doubt and no theory is actually required. At this early stage the application serves to corroborate the theory. The next stage develops when the theory is applied to somewhat more complicated situations in which it may already lead to a certain extent beyond the obvious and the familiar. Here theory and application corroborate each other mutually. Beyond this lies the field of real success: genuine predictions by theory. It is well known that all mathematized sciences have gone through these successive phases of evolution. ([40], pp. 7-8)

So at least von Neumann, the father of modern-day game theory and computer science, attached importance to spanning the spectrum from the theoretical to the applied. Pragmatics may be key to achieving von Neumann and Morgenstern's third stage, and may prove to be a joint endeavor between computer science and game theory.

[5] Parenthetically it can be remarked that Savage's setting [34] on which the modern Bayesian framework is based does not have an obvious extension to the multi-agent case. However this is not the focus of the point I'm making here.

Acknowledgements

I thank Alon Altman, Joe Halpern, Sam Ieong, Daphne Koller, Tim Roughgarden, Moshe Vardi, Mike Wellman and anonymous referees for useful help and suggestions. Of course all opinions, errors, and erroneous opinions are mine alone. This work has been supported by NSF grant ITR-0205633.

References

[1] M. E. Bratman. *Intention, Plans, and Practical Reason*. CSLI Publications, Stanford University, 1987.

[2] X. Chen and X. Deng. Settling the complexity of 2-player Nash-equilibrium. In *FOCS*, 2006.

[3] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[4] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. In *IJCAI*, pages 761–771, 2003.

[5] V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *AAAI*, 2004.

[6] P. C. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[7] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19:257, 1994.

[8] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[9] I. Niemela G. Brewka and M. Truszczynski. Nonmonotonic reasoning. In *Handbook of Knowledge Representation*. Elsevier, 2007. Frank van Harmelen and Vladimir Lifschitz and Bruce Porter (eds.).

[10] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[11] S. Govindan and R. Wilson. A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, 110:65–86, 2003.

[12] A. Greenwald and M. L. Littman (Eds.). Special issue on learning and computational game theory. *Machine Learning*, 67(1-2), 2007.

[13] J. Y. Halpern. A computer scientist looks at game theory. *Games and Economic Behavior*, 45(1):114–132, 2003.

[14] J. Y. Halpern. Computer science and game theory: A brief survey. In *The New Palgrave Dictionary of Economics*. Palgrave MacMillan, 2008. S. N. Durlauf and L. E. Blume (eds.).

[15] S. Ieong and Y. Shoham. Multi-attribute coalitional games. In *EC'06: Proceedings of the 7th ACM conference on electronic commerce*, pages 170–179, 2006.

[16] E. Kalai. Games, computers, and O.R. In *ACM/SIAM Symposium on Discrete Algorithms*, 1995.

[17] E. Kalai. Presidential address, the second World Congress of the Game Theory Society, Marseille 2004. *Games and Economic Behavior*, 2008. Special issue, to appear (P. Reny, ed.).

[18] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *UAI*, 2001.

[19] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *IJCAI*, 2001.

[20] P. LaMura. Game networks. In *UAI*, 2000.

[21] K. Leyton-Brown and M. Tennenholtz. Local-effect games. In *IJCAI*, 2003.

[22] N. Linial. Game theoretic aspects of computing. In *Handbook of Game Theory*, volume 2, pages 1339–1395. Elsevier Science, 1994. R. J. Aumann and S. Hart (eds.).

[23] M. Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon and Shuster, 2007.

[24] A. Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, pages 227–229, 1985.

[25] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[26] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proc. 31st ACM Symp. on Theory of Computing*, pages 129–140, 1999.

[27] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[28] C. H. Papadimitriou and M. Yannakakis. On bounded rationality and computational complexity. In *STOC: Proceedings of the Symposium on the Theory of Computing*, pages 726–733, 1994.

[29] Christos H. Papadimitriou. Algorithms, games, and the Internet. *Lecture Notes in Computer Science*, 2076, 2001.

[30] P. Pappas. Belief revision. In *Handbook of Knowledge Representation*. Elsevier, 2007. Frank van Harmelen and Vladimir Lifschitz and Bruce Porter (eds.).

[31] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. In *AAAI'04: Proceedings of the National Conference on Artificial Intelligence*, pages 664–669, 2004.

[32] T. Roughgarden. Computing equilibria: A computational complexity perspective. *Economic Theory*, 2008.

[33] A. Rubinstein. *Modeling Bounded Rationality*. MIT Press, 1998.

[34] L. J. Savage. *The Foundations of Statistics*. John Wiley and Sons, New York, 1954. (Second Edition, Dover Press, 1972).

[35] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, game theoretic, and logical foundations*. Cambridge University Press, 2008.

[36] M. Tennenholtz. Program equilibrium. *Games and Economic Behavior*, 49:363–373, 2004.

[37] J. van Benthem. *Exploring Logical Dynamics*. Center for the Study of Language and Information, Stanford, CA, USA, 1997.

[38] J. van Benthem. When are two games the same? In *LOFT-III*, 1998. (ILLC preprint, 1999).

[39] R. Vohra and M. P. Wellman (eds.). Special issue on foundations of multiagent learning. *Artificial Intelligence*, 171(1), 2007.

[40] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior, Second Edition*. Princeton University Press, second edition, 1947.

[41] B. von Stengel. Computing equilibria for two-person games. In Robert Aumann and Sergiu Hart, editors, *Handbook of Game Theory*, volume III, chapter 45, pages 1723–1759. Amsterdam: Elsevier, 2002.

[42] M. P. Wellman, A. Greenwald, and P. Stone. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. MIT Press, 2007.

[43] M. Zinkevich, M. Bowling, , and N. Burch. A new algorithm for generating equilibria in massive zero-sum games. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 788–793, 2007.