Program Equilibria and Discounted ComputationTime

Lance Fortnow*
Northwestern University

December 4, 2008

Abstract

Tennenholtz (GEB 2004) developed Program Equilibrium to model play in a finite two-player game where each player can base their strategy on the other player's strategies. Tennenholtz's model allowed each player to produce a "loop-free" computer program that had access to the code for both players. He showed a folk theorem where any mixed-strategy individually rational play could be an equilibrium payoff in this model even in a one-shot game. Kalai et al. gave a general folk theorem for correlated play in a more generic commitment model.

We develop a new model of program equilibrium using general computational models and discounting the payoffs based on the computation time used. We give an even more general folk theorem giving correlated-strategy payoffs down to the pure minimax of each player. We also show equilibrium in other games not covered by the earlier work.

1 Introduction

Consider two players Alice and Bob who play a finite game like Prisoner's Dilemma. Instead of just choosing an action, suppose they try to reason about what the other player will do. For example, Alice may be willing to cooperate as long as Alice believes Bob will also cooperate and vice-versa. Cooperation may become an equilibria, even in a one-shot game, if both players can share their reasoning mechanisms.

Tennenholtz [Ten04] suggests modeling the reasoning processes as computer programs where each player submits a program that can see the code of the other player (as well as its own code). To avoid infinite regression (each player simulating the other with neither ever willing to halt first) Tennenholtz limits the programs to be "loop-free" or straight-line programs in the CS vernacular that must pick some action in the action space.

Tennenholtz defines a notion of *Program Equilibrium* in this model and shows that dual cooperation can indeed be achieved: Both players submit programs that will cooperate if both programs are identical and defect otherwise.

Tennenholtz proves a general "Folk Theorem" for these games where every individually rational mixed strategy can be achieved as an equilibrium in the program equilibrium model, even for one-shot games.

We broaden Tennenholtz's work by allowing the players to play arbitrary Turing machines. We avoid infinite regression by discounting the running time, i.e., for some $\delta < 1$, the payoff of

*Supported in part by NSF grants CCF-0829754 and DMS-0652521.

each player is discounted by a multiplicative factor of δ^t where the player used t steps in their computation. If a player's program doesn't halt, the player gets a payoff of zero. If one player halts while the opponent doesn't, the player gets the best possible (discounted) payoff for his action.

The idea of discounting utility comes from a standard assumption in economic theory that people value an (inflation-adjusted) dollar a year from now less than receiving that dollar today. The discount δ for a specific time-period is chosen so that an agent is indifferent between receiving δ dollars at the beginning of the period and one dollar at the end of the period.

Not only does our model allow for arbitrary computer programs, we also get a stronger folk theorem than Tennenholtz or his successors. We can achieve any correlated strategy individual rational play down to the pure-strategy minimax as a program equilibrium in our model.

We describe our model in Section 3 and the main Folk theorem in Section 4. In Section 5, we consider the largest-integer game where each player announces an integer and the one who announces the largest integer receives \$100 and splitting the pot if they give the same number. This game has no Nash equilibrium and no program equilibrium in Tennenholtz or later models. We give a simple equilibrium that splits the pot in our model and can also achieve a general Folk theorem for this game.

In Section 6.1 we look at Chess. The Nash equilibrium and previous program equilibriums would require the players to play best play despite the fact that it is assumably computationally hard. We argue that in our discounted time model, best play is not an equilibrium and make the case (but don't prove) that the right equilibrium better matches how computers and humans play chess.

Finally in Section 6.2 we discuss discounted time in traditional computational complexity outside of game theory, arguing that it is more natural than our traditional definitions, can be used to capture the standard classes and may give us a different way to understand average-case complexity, expected running time and approximation algorithms.

2 Related Work

2.1 Tennenholtz Program Equilibrium

In his seminal paper, Tennenholtz [Ten04] developed the first program equilibrium model. Fix a finite 2-play normal-form game G. Instead of playing actions in G, each player produces a program to compute the action. The twist is that each program can look at the code of the other player's program (as well as its own). This twist captures the idea of reasoning about the other player's strategy.

He shows how to achieve cooperation in a 1-shot Prisoner's Dilemma game.

	C	D
C	$3,3$	$0,4$
D	$4,0$	$1,1$

Figure 1: Prisoner's Dilemma

The Equilibrium Strategies work as follows:

$P_1(\langle P_1\rangle, \langle P_2\rangle)$:

If $\langle P_2 \rangle$ is as below
   Cooperate.
Otherwise Defect.

$P_2(\langle P_1 \rangle, \langle P_2 \rangle)$:
If $\langle P_1 \rangle$ is as above
   Cooperate.
Otherwise Defect.

If both players play the equilibrium programs the programs will both cooperate. If one player deviates the other will defect.

To guarantee that the programs always play an action, Tennenholtz allows only "loop-free" or straight-line programs. In his model, the players must use pure strategies to choose a program but the programs themselves can randomize. Tennenholtz shows the following Folk theorem for his model. A program equilibrium is a choice of programs for each player such that neither player has incentive to choose a different program.

Theorem 1 (Tennenholtz) *Fix a finite two-player game G and mixed strategies of the players on that game such that are individually rational, that is the payoff for each player is better than their worst-possible payoff for some mixed-strategy. Then there is a program equilibrium where each player picks a program that has the same payoffs for each player.*

2.2 Later Work

Monderer and Tennenholtz [MT06] consider "mediated equilibrium" where agents can optionally use a mediator that behaves in a pre-specified way based on messages received from agents. They show that this model generalizes Tennenholtz and show a number of Folk theorems for arbitrary numbers of players.

Kalai, Kalai, Lehrer and Samet [KKLS07] consider a model of "commitments." Each player has a set of devices that represent a total function from the other player's device to an action of a finite two-player game. There are additional "voluntary devices" that allow a player to play an action of the original game and the other player cannot react to that choice of action.

In this model Kalai et al. prove a stronger version of Tennenholtz's Theorem 1. A correlated play is a arbitrary distribution on the set of pairs of actions of both players (whereas a mixed strategy only allows product distributions on the sets of pairs).

Theorem 2 (Kalai-Kalai-Lehrer-Samet) *Fix a finite two-player game G and correlated strategies of the players on that game such that are individually rational, that is the payoff for each player is better than their worst-possible payoff for some mixed-strategy. Then there is a set of devices (including voluntary devices) for both players and a mixed-strategy equilibrium over devices with the same payoffs for each player.*

Note the players choose mixed-strategies over devices yet still can achieve equilibrium for correlated strategies in the original game. One can get a similar result for Tennenholtz's model if one allows mixed-strategies and apply the techniques of the proof of Theorem 2.

Peters and Szentes [PS08] consider the model where each player can play a first-order formula over the integers which gets the Gödel-numbering of the formula for the other player as well as its own. This is equivalent to allowing the players to play total functions in the arithmetic hierarchy (computable in the halting problem, the halting problem relative to the halting problem, etc.) They get results similar to Theorem 1 and 2 in their model even when allowing more than two players.

Halpern and Pass [HP08] consider mediation based on cryptographic assumptions with players of limited computational power.

3 Our Model

Let $G = (A_1, A_2, u_1, u_2)$ be a two-player finite normal-form game. Each A_i is a finite set of actions and $u_i : A_1 \times A_2 \rightarrow \Re^{\geq 0}$.

Player i has action space A_i and receives utility $u_i(a_1, a_2)$ when each player i chose action a_i.

In the program equilibria model, each player does not directly choose an action, rather they choose a program P_i. We allow a program to be an arbitrary Turing machine. Let $\langle P_i \rangle$ represent the code for program P_i. Alternatively we can allow program to be chosen from any Turing-complete programming language (includes nearly all common languages including C, Java, Basic as long as there is no a priori bound on memory).

Each P_i will receive two inputs, $\langle P_1 \rangle$ and $\langle P_2 \rangle$, the codes for the programs produced by the players denoted by $P_i(\langle P_1 \rangle, \langle P_2 \rangle)$. Since we allow arbitrary Turing machines, $P_i(\langle P_1 \rangle, \langle P_2 \rangle)$ may have an arbitrary output or may not output at all because it doesn't halt. Let a_i be the output of $P_i(\langle P_1 \rangle, \langle P_2 \rangle)$, where we say $a_i = \perp$ if there was no output or the output was not in A_i. Let t_i be the number of computation steps used by $P_i(\langle P_1 \rangle, \langle P_2 \rangle)$ before it outputs a_i where we say $t_i = \infty$ if the machine didn't halt or didn't output an action in A_i.

We discount the utility by the running time. Fix δ with $0 < \delta < 1$. In current technology, think of δ about $1 - 10^{-12}$.

Let a_1, a_2, t_1 and t_2 be the actions and times as described above where each Player i plays program P_i. Here are the payoffs in the program equilibria game:

- If $a_1 \neq \perp$ and $a_2 \neq \perp$ then each Player i receives utility $\delta^{t_i} u_i(a_1, a_2)$.
- If $a_1 = a_2 = \perp$ both players receive zero utility.
- If $a_1 \neq \perp$ and $a_2 = \perp$ then Player 2 receives zero utility and Player 1 receives utility $\delta^{t_1} \max_{b \in A_2} u_1(a_1, b)$.
- If $a_1 = \perp$ and $a_2 \neq \perp$ then Player 1 receives zero utility and Player 2 receives utility $\delta^{t_2} \max_{b \in A_1} u_2(b, a_2)$.

One could also consider using discounting functions that use both running times such as $\delta^{t_1+t_2}$ or $\delta^{\max(t_1,t_2)}$ or use min instead of max when one machine doesn't give an action. Our results hold in those models as well but we don't want to punish one player because the other player's program used a large or infinite amount of time.

We allow players to use mixed strategies to choose the programs P_1 and P_2 but the programs themselves must be deterministic.

4 Folk Theorem

For a game G the pure minimax utility for player 1 is

$$\alpha_1 = \min_{a_2 \in A_2} \max_{a_1 \in A_1} u_2(a_1, a_2)$$

and the pure maximin utility for player 2 is

$$\alpha_2 = \min_{a_1 \in A_1} \max_{a_2 \in A_2} u_1(a_1, a_2).$$

A game G has a non-empty interior of individually rational strategies if each player can possibly achieve better than their pure minimax, specifically for $i \in \{1, 2\}$,

$$\alpha_i < \max_{a_1 \in A_1} \max_{a_2 \in A_2} u_i(a_1, a_2).$$

A correlated play of a game G is when the actions a_1 and a_2 are drawn from some joint distribution $\mathcal{D}$ over $A_1 \times A_2$. A correlated play is purely individually rational if the expected utilities of each player over $\mathcal{D}$ is at least their pure maximin utility.

Theorem 3 *For every game G with nonnegative payoffs and a non-empty interior of individually rational strategies, and any purely individually rational correlated play on G with distribution $\mathcal{D}$ and every $\epsilon > 0$ there is some $\delta < 1$ such that there is a Nash Equilibrium in the mixed program equilibrium game where each player's expected utility is within ϵ of their expected utility over $\mathcal{D}$.*

Like Kalai et al. [KKLS07], we achieve correlated actions with only mixed strategies. We use techniques similar to Kalai et al. to achieve the correlation though our programs are otherwise quite different than those used by Kalai et al. or Tennenholtz [Ten04].

Note we achieve a true Nash equilibrium, not merely an ϵ-Nash. However we can achieve equilibrium on only a dense subset of the correlated strategies for two reasons:

1. There are an uncountable number of distribution $\mathcal{D}$ but only a countable number of distributions possibly generated by the countable set of Turing machines.

2. In equilibrium the programs use a small but non-zero amount of computation time which leads to a tiny amount of discounting.

In Section 4.1 we give an example to show why we need a non-empty interior.

Proof of Theorem 3:

Assume that under $\mathcal{D}$, both players achieve an expected payoff more than ϵ higher than their minimax payoffs. We can achieve this, if needed, by modifying $\mathcal{D}$ to put a small amount of weight on the action pairs that give each player their maximum utility.

Since the expected utilities of the players are continuous as a function of $\mathcal{D}$ choose n and a distribution $\mathcal{E}$ such that the each probability of choosing any action pair (a_1, a_2) according to $\mathcal{E}$ is a multiple of 2^{-n} and the expected utilities of both players according to $\mathcal{D}$ and $\mathcal{E}$ differ by at most $\epsilon/2$. Fix a function $f : \{0,1\}^n \rightarrow A_1 \times A_2$ such that

$$\Pr_r(f(r) = (a_1, a_2)) = \Pr_{\mathcal{E}}(a_1, a_2)$$

where r is chosen uniformly over $\{0,1\}^n$.

In equilibrium each player i chooses a string r_i uniformly over $\{0,1\}^n$ and plays the following programs.

$P_1(\langle P_1\rangle, \langle P_2\rangle)$:
If $\langle P_2\rangle$ is not as below
  Simulate $P_2(\langle P_1\rangle, \langle P_2\rangle)$
  If $P_2(\langle P_1\rangle, \langle P_2\rangle)$ outputs action a_2
    output action a_1 that minimizes $u_2(a_1, a_2)$.
Otherwise
  Pull r_2 from $\langle P_2\rangle$.
  Play action a_1 where $f(r_1 \oplus r_2) = (a_1, a_2)$.

$P_2(\langle P_1\rangle, \langle P_2\rangle)$:
If $\langle P_1\rangle$ is not as above
  Simulate $P_1(\langle P_1\rangle, \langle P_2\rangle)$
  If $P_1(\langle P_1\rangle, \langle P_2\rangle)$ outputs action a_1
    output action a_2 that minimizes $u_1(a_1, a_2)$.
Otherwise
  Pull r_1 from $\langle P_1\rangle$.
  Play action a_2 where $f(r_1 \oplus r_2) = (a_1, a_2)$.

Suppose both players play according to the equilibrium strategy. In which case the "Otherwise" clause will kick, $r_1 \oplus r_2$ will be uniformly distributed over $\{0,1\}^n$ and the expected payoff for player i will be $\delta^t \beta_i$ where t is the number of steps used by each program and β_i is the expected payoffs according to $\mathcal{E}$. Choose δ so that $(1-\delta^t)\beta_i \leq \epsilon/2$ for each i and the payoffs for each player is at most ϵ worse than the expected payoffs in G under $\mathcal{D}$.

Suppose Player 1 plays the equilibrium strategy but Player 2 deviates (the reverse case is similar). We have two cases.

If Player 2 plays a different program than above than either Player 2's program doesn't halt and Player 2 receives zero utility, or Player 2's program halts with some action a_2 and Player 1's program will play the a_1 that minimizes $u_2(a_1, a_2)$. At best Player 2 will achieve her pure minimax payoff which by assumption is more than ϵ less than Player 2's payoff in G under $\mathcal{D}$ and thus strictly less than Player 2's program equilibrium payoff.

Player 2 can also deviate by playing the program above but not choosing r_2 according to the uniform distribution. But since r_1 is chosen uniformly, $r_1 \oplus r_2$ will be uniform independent of the distribution chosen by Player 2. Since the running time of the program is independent of the choice of r_2, Player 2 will not gain (or lose) by deviating in this way. □

4.1 Empty Interior

We give an example as to why we need non-trivial strategies to achieve for our Folk Theorem (Theorem 3). Consider the game in Figure 2.

	L	R
U	$1, 5$	$1, 4$
D	$0, 1$	$1, 1$

Figure 2: A Game with an Empty Interior of Individually Rational Strategies

An expected payoff of $(1, 3)$ can be achieved by the correlated play of playing (U, L) and (D, R) each with probability 1/2 or the mixed strategy of player 2 always playing R and player 1 playing U with with probability 2/3 and D with probability 1/3 or some combination of the two.

In either case if we used the programs in the proof of Theorem 3, player 1 could deviate by just immediately playing U since it still receives a utility one 1 but with less time and thus less of a discount. Any program equilibrium achieving expected payoff $(1, 3)$ would have Player 1's program either immediately playing U or D with a positive probability of the program playing U. However in that case Player 2 could benefit by just playing L and thus there is no program equilibrium achieving an expected discounted payoff close to $(1, 3)$.

5 Largest Integer Game

Our Folk theorem can be extended to some infinite games that don't even have a Nash Equilibrium.

Consider the following 2-player infinite game, Largest Integer. Each Player i choose an integer a_i.

- If $a_1 > a_2$ then Player 1 receives 100 and Player 2 receives 0.
- If $a_2 > a_1$ then Player 2 receives 100 and Player 1 receives 0.
- If $a_1 = a_2$ then both players receive 50.

Largest Integer has no Nash equilibrium since for any mixed strategy of Player 1, player 2 can play a large enough integer to achieve $100 - \epsilon$ for any $\epsilon > 0$ and vice-versa. For similar reasons there is no equilibrium in Tennenholtz's model or any of the other models described in Section 2.

We can achieve a discounted equilibrium of $50 - \epsilon$ as follows:

$P_1(\langle P_1\rangle, \langle P_2\rangle)$:
If $\langle P_2\rangle$ is not as below
  Simulate $P_2(\langle P_1\rangle, \langle P_2\rangle)$
  If $P_2(\langle P_1\rangle, \langle P_2\rangle)$ outputs action a_2
    output action $a_1 = a_2 + 1$.
Otherwise Output 1.

$P_1(\langle P_1\rangle, \langle P_2\rangle)$:
If $\langle P_1\rangle$ is not as above
  Simulate $P_1(\langle P_1\rangle, \langle P_2\rangle)$
  If $P_2(\langle P_1\rangle, \langle P_2\rangle)$ outputs action a_1
    output action $a_2 = a_1 + 1$.
Otherwise Output 1.

If a player deviates from the equilibrium strategy they will receive a payoff of zero, either because they ran forever or because the other player will play a larger number.

Using similar ideas from the proof of Theorem 3, we can get a general Folk Theorem for the Largest Integer game.

6 Future Directions

In Section 5 we gave a Folk theorem for a countable game. Can we prove a Folk theorem for a general class of games.

What happens if we have more than two players? What if we look at a family of games paramaterized by some value n? In the parameterized setting, the number of players could also depend on the parameter.

Can we use the Kleene recursion theorem [Kle38] to eliminate the need for each program to have its own code as input?

6.1 Computationally Hard Games

Consider a standard chess game and say the payoff is 100 for winning, 50 for a draw and 1 for losing (so it is better to lose than to keep playing indefinitely). The only Nash equilibrium is best play and this is also the only equilibrium in previous models of program equilibriums and commitment. But neither man nor machine plays best play in chess because of the computationally difficulty in searching the full game tree. We can't prove that there is some unknown quick algorithm that does play perfectly but for the sake of the argument let us assume no such algorithm exists.

For a reasonable choice of δ, say $\delta = 1 - 10^{-12}$, best play no longer becomes an equilibrium in the discounted time model since searching the game tree would drop the discounted payoff to less than just immediately resigning.

Instead discounting suggests a method of playing that more closely matches how humans and computers play chess: Doing an carefully pruned search of small number of levels of the game tree and then apply some evaluation function that gives a belief of winning from that position as well as a belief in a possible increase in value from further exploration, optimizing the time with the amount of discount given up in the search.

It would be extremely difficult if not impossible to exactly characterize such an equilibrium. One could try to characterize equilibria of other simpler but still computationally difficult games.

6.2 Discounted Time in Computational Complexity

Are there interesting applications of discounted time outside of game theory? In computational complexity when we talk about time it usually represents a hard limit in the running time, solving the problem in time $t(n)$. So we are happy, say, if we can solve the problem in one hour and miserable if it takes 61 minutes. But our real gradation of happiness over the running time is not so discontinuous.

Let us consider discounting the value of a solution by a δ^t factor for an algorithm that uses t steps, so a small increase in the running time yields only a small decrease in utility. When t is small, δ^t is about $1 - \epsilon t$, a linear decrease. For t large, δ^t is about $e^{-\epsilon t}$, an exponential decrease.

There is also a time-independent flavor to this notion. After time t the additional discount for continuing for another r steps is δ^r, independent of t.

We can also recover traditional complexity classes. $\mathrm{DTIME}(O(m(n)))$ is the set of languages computable in time t such that for some constant $c > 0$, $\delta^t > c$ for $\delta = 1 - \frac{1}{m(n)}$.

Abbott and Garcia-Molina [AG88] had considered non-hard deadlines in databases though not specifically using a discount factor.

Some possible applications of discounted time in computational complexity and algorithms.

- What does average case and expected time mean in the discounted time model?
- What if you take the value of the solution of some approximation problem and discount it with the time taken? Can you determine the optimal point to stop?

7 Acknowledgments

I thank in particular Ehud Kalai, Balázs Szentes and Rakesh Vohra for their many useful insights as well as the many great comments I received from early presentations at Dagstuhl and Northwestern.

I presented the ideas on discounted time in complexity in Section 6.2 at a rump session at the 2008 Conference on Computational Complexity and in my weblog [For08] and thank those who game me comments. In particular thanks to Alex Lopez-Ortiz for pointing me to the work of Abbott and Garcia-Molina [AG88].

References

[AG88] R. Abbott and H. Garcia-Molina. Scheduling real-time transacations. *ACM SIGMOD Record*, 17(1):71–81, March 1988.

[For08] L. Fortnow. Discounted time. *Computational Complexity Weblog*, August 2008. http://weblog.fortnow.com/2008/08/discounted-time.html.

[HP08] J. Halpern and R. Pass. Game theory with costly computation. Technical Report arXiv:0909.0024v1, arXiv, 2008.

[KKLS07] A. Kalai, E. Kalai, E. Lehrer, and D. Samet. A commitment folk theorem. Manuscript., 2007.

[Kle38] S. Kleene. On notation for ordinal numbers. *Journal of Symbolic Logic*, 3:150–155, 1938.

[MT06] D. Monderer and M. Tennenholtz. Strong mediated equilibrium. Manuscript., 2006.

[PS08] M. Peters and B. Szentes. Definable and contractible contracts. Manuscript., 2008.

[Ten04] M. Tennenholtz. Program equilibrium. *Games and Economic Behavior*, 49(2):363–373, November 2004.