3.4.5 Correlated equilibrium

The correlated equilibrium is a solution concept that generalizes the Nash equilibrium. Some people feel that this is the most fundamental solution concept of all.[7]

In a standard game, each player mixes his pure strategies independently. For example, consider again the Battle of the Sexes game (reproduced here as Figure 3.18) and its mixed-strategy equilibrium.

	LW	WL
LW	$2, 1$	$0, 0$
WL	$0, 0$	$1, 2$

Figure 3.18: Battle of the Sexes game.

As we saw in Section 3.3.3, this game's unique mixed-strategy equilibrium yields each player an expected payoff of $2/3$. But now imagine that the two players can observe the result of a fair coin flip and can condition their strategies based on that outcome. They can now adopt strategies from a richer set; for example, they could choose "WL if heads, LW if tails." Indeed, this pair forms an equilibrium in this richer strategy space; given that one player plays the strategy, the other player only loses by adopting another. Furthermore, the expected payoff to each player in this so-called correlated equilibrium is $.5 * 2 + .5 * 1 = 1.5$. Thus both agents receive higher utility than they do under the mixed-strategy equilibrium in the uncorrelated case (which had expected payoff of 2/3 for both agents), and the outcome is fairer than either of the pure-strategy equilibria in the sense that the worst-off player achieves higher expected utility. Correlating devices can thus be quite useful.

The aforementioned example had both players observe the exact outcome of the coin flip, but the general setting does not require this. Generally, the setting includes some random variable (the "external event") with a commonly-known probability distribution, and a private signal to each player about the instantiation of the random variable. A player's signal can be correlated with the random variable's value and with the signals received by other players, without uniquely identifying any of them. Standard games can be viewed as the degenerate case in which the signals of the different agents are probabilistically independent.

To model this formally, consider n random variables, with a joint distribution over these variables. Imagine that nature chooses according to this distribution, but

7. A Nobel-prize-winning game theorist, R. Myerson, has gone so far as to say that "if there is intelligent life on other planets, in a majority of them, they would have discovered correlated equilibrium before Nash equilibrium."

reveals to each agent only the realized value of his variable, and that the agent can condition his action on this value.[8]

correlated equilibrium

Definition 3.4.12 (Correlated equilibrium) *Given an n-agent game $G = (N, A, u)$, a* correlated equilibrium *is a tuple (v, π, σ), where v is a tuple of random variables $v = (v_1, \dots, v_n)$ with respective domains $D = (D_1, \dots, D_n)$, π is a joint distribution over v, $\sigma = (\sigma_1, \dots, \sigma_n)$ is a vector of mappings $\sigma_i : D_i \mapsto A_i$, and for each agent i and every mapping $\sigma'_i : D_i \mapsto A_i$ it is the case that*

$$\sum_{d \in D} \pi(d) u_i \left(\sigma_1(d_1), \dots, \sigma_i(d_i), \dots, \sigma_n(d_n)\right) \geq \sum_{d \in D} \pi(d) u_i \left(\sigma_1(d_1), \dots, \sigma'_i(d_i), \dots, \sigma_n(d_n)\right).$$

Note that the mapping is to an action—that is, to a pure strategy rather than a mixed one. One could allow a mapping to mixed strategies, but that would add no greater generality. (Do you see why?)

For every Nash equilibrium, we can construct an equivalent correlated equilibrium, in the sense that they induce the same distribution on outcomes.

Theorem 3.4.13 *For every Nash equilibrium σ^* there exists a corresponding correlated equilibrium σ.*

The proof is straightforward. Roughly, we can construct a correlated equilibrium from a given Nash equilibrium by letting each $D_i = A_i$ and letting the joint probability distribution be $\pi(d) = \prod_{i \in N} \sigma^*_i(d_i)$. Then we choose σ_i as the mapping from each d_i to the corresponding a_i. When the agents play the strategy profile σ, the distribution over outcomes is identical to that under σ^*. Because the v_i's are uncorrelated and no agent can benefit by deviating from σ^*, σ is a correlated equilibrium.

On the other hand, not every correlated equilibrium is equivalent to a Nash equilibrium; the Battle-of-the-Sexes example given earlier provides a counter-example. Thus, correlated equilibrium is a strictly weaker notion than Nash equilibrium.

Finally, we note that correlated equilibria can be combined together to form new correlated equilibria. Thus, if the set of correlated equilibria of a game G does not contain a single element, it is infinite. Indeed, any convex combination of correlated equilibrium payoffs can itself be realized as the payoff profile of some correlated equilibrium. The easiest way to understand this claim is to imagine a public random device that selects which of the correlated equilibria will be played; next, another random number is chosen in order to allow the chosen equilibrium to be played. Overall, each agent's expected payoff is the weighted sum of the payoffs

8. This construction is closely related to two other constructions later in the book, one in connection with Bayesian Games in Chapter 6, and one in connection with knowledge and probability (KP) structures in Chapter 13.

from the correlated equilibria that were combined. Since no agent has an incentive to deviate regardless of the probabilities governing the first random device, we can achieve any convex combination of correlated equilibrium payoffs. Finally, observe that having two stages of random number generation is not necessary: we can simply derive new domains D and a new joint probability distribution π from the D's and π's of the original correlated equilibria, and so perform the random number generation in one step.

3.4.6 Trembling-hand perfect equilibrium

Another important solution concept is the *trembling-hand perfect equilibrium*, or simply *perfect equilibrium*. While rationalizability is a weaker notion than that of a Nash equilibrium, perfection is a stronger one. Several equivalent definitions of the concept exist. In the following definition, recall that a fully mixed strategy is one that assigns every action a strictly positive probability.

trembling-hand perfect equilibrium

Definition 3.4.14 (Trembling-hand perfect equilibrium) *A mixed-strategy profile s is a* (trembling-hand) perfect equilibrium *of a normal-form game G if there exists a sequence $s^0, s^1, \ldots$ of fully mixed-strategy profiles such that $\lim_{n\to\infty} s^n = s$, and such that for each s^k in the sequence and each player i, the strategy s_i is a best response to the strategies s^k_{-i}.*

proper equilibrium

Perfect equilibria are relevant to one aspect of multiagent learning (see Chapter 7), which is why we mention them here. However, we do not discuss them in any detail; they are an involved topic, and relate to other subtle refinements of the Nash equilibrium such as the *proper equilibrium*. The notes at the end of the chapter point the reader to further readings on this topic. We should, however, at least explain the term "trembling hand." One way to think about the concept is as requiring that the equilibrium be robust against slight errors—"trembles"—on the part of players. In other words, one's action ought to be the best response not only against the opponents' equilibrium strategies, but also against small perturbation of those. However, since the mathematical definition speaks about arbitrarily small perturbations, whether these trembles in fact model player fallibility or are merely a mathematical device is open to debate.

3.4.7 ϵ-Nash equilibrium

Our final solution concept reflects the idea that players might not care about changing their strategies to a best response when the amount of utility that they could gain by doing so is very small. This leads us to the idea of an ϵ-Nash equilibrium.

Definition 3.4.15 (ϵ-Nash) *Fix $\epsilon > 0$. A strategy profile $s = (s_1, \ldots, s_n)$ is an ϵ-Nash equilibrium if, for all agents i and for all strategies $s'_i \neq s_i$, $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) - \epsilon$.*

2. **(Reduction identity)** Given action subsets $A'_i \subseteq A_i$ for each player i, does there exist a maximally reduced game where each player i has the actions A'_i?

3. **(Reduction size)** Given constants k_i for each player i, does there exist a maximally reduced game where each player i has exactly k_i actions?

It turns out that the complexity of answering these questions depends on the form of domination under consideration.

Theorem 4.5.1 *For iterated strict dominance, the* strategy elimination*,* reduction identity*,* uniqueness *and* reduction size *problems are in P. For iterated weak dominance, these problems are NP-complete.*

The first part of this result, considering iterated strict dominance, is straightforward: it follows from the fact that iterated strict dominance always arrives at the same set of strategies regardless of elimination order. The second part is tricker; indeed, our statement of this theorem sweeps under the carpet some subtleties about whether domination by mixed strategies is considered (it is in some cases, and is not in others) and the minimum number of utility values permitted for each player. For all the details, the reader should consult the papers cited at the end of the chapter.

4.6 Computing correlated equilibria

The final solution concept that we will consider is correlated equilibrium. It turns out that correlated equilibria are (probably) easier to compute than Nash equilibria: a sample correlated equilibrium can be found in polynomial time using a linear programming formulation. It is not hard to see (e.g., from the proof of Theorem 3.4.13) that every game has at least one correlated equilibrium in which the value of the random variable can be interpreted as a recommendation to each agent of what action to play, and in equilibrium the agents all follow these recommendations. Thus, we can find a sample correlated equilibrium if we can find a probability distribution over pure action profiles with the property that each agent would prefer to play the action corresponding to a chosen outcome when told to do so, given that the other agents are doing the same.

As in Section 3.2, let $a \in A$ denote a pure-strategy profile, and let $a_i \in A_i$ denote a pure strategy for player i. The variables in our linear program are $p(a)$, the probability of realizing a given pure-strategy profile a; since there is a variable for every pure-strategy profile there are thus $|A|$ variables. Observe that as above the values $u_i(a)$ are constants. The linear program follows.

$$\sum_{a \in A | a_i \in a} p(a) u_i(a) \geq \sum_{a \in A | a_i \in a} p(a) u_i(a'_i, a_{-i}) \quad \forall i \in N, \forall a_i, a'_i \in A_i \tag{4.52}$$

$$p(a) \geq 0 \qquad \forall a \in A \tag{4.53}$$

$$\sum_{a \in A} p(a) = 1 \tag{4.54}$$

Constraints (4.53) and (4.54) ensure that p is a valid probability distribution. The interesting constraint is (4.52), which expresses the requirement that player i must be (weakly) better off playing action a when he is told to do so than playing any other action a'_i, given that other agents play their prescribed actions. This constraint effectively restates the definition of a correlated equilibrium given in Definition 3.4.12. Note that it can be rewritten as $\sum_{a \in A | a_i \in a} [u_i(a) - u_i(a'_i, a_{-i})] p(a) \geq 0$; in other words, whenever agent i is "recommended" to play action a_i with positive probability, he must get at least as much utility from doing so as he would from playing any other action a'_i.

We can select a desired correlated equilibrium by adding an objective function to the linear program. For example, we can find a correlated equilibrium that maximizes the sum of the agents' expected utilities by adding the objective function

$$\text{maximize: } \sum_{a \in A} p(a) \sum_{i \in N} u_i(a). \tag{4.55}$$

Furthermore, all of the questions discussed in Section 4.2.4 can be answered about correlated equilibria in polynomial time, making them (most likely) fundamentally easier problems.

Theorem 4.6.1 *The following problems are in the complexity class P when applied to correlated equilibria:* uniqueness, Pareto optimal, guaranteed payoff, subset inclusion*, and* subset containment*.*

Finally, it is worthwhile to consider the reason for the computational difference between correlated equilibria and Nash equilibria. Why can we express the definition of a correlated equilibrium as a linear constraint (4.52), while we cannot do the same with the definition of a Nash equilibrium, even though both definitions are quite similar? The difference is that a correlated equilibrium involves a single randomization over action profiles, while in a Nash equilibrium agents randomize separately. Thus, the (nonlinear) version of constraint (4.52) which would instruct a feasibility program to find a Nash equilibrium would be

$$\sum_{a \in A} u_i(a) \prod_{j \in N} p_j(a_j) \geq \sum_{a \in A} u_i(a'_i, a_{-i}) \prod_{j \in N \setminus \{i\}} p_j(a_j) \quad \forall i \in N, \forall a'_i \in A_i.$$

This constraint now mimics constraint (4.52), directly expressing the definition of Nash equilibrium. It states that each player i attains at least as much expected utility from following his mixed strategy p_i as from any pure strategy deviation a'_i, given the mixed strategies of the other players. However, the constraint is nonlinear because of the product $\prod_{j \in N} p_j(a_j)$.

4.7 History and references

The complexity of finding a sample Nash equilibrium is explored in a series of articles. First came the original definition of the class TFNP [Megiddo and Papadimitriou, 1991], a super-class of PPAD, followed by the definition of PPAD by Papadimitriou [1994]. Next, Goldberg and Papadimitriou [2006] showed that finding an equilibrium of a game with any constant number of players is no harder than finding the equilibrium of a four-player game, and Daskalakis et al. [2006b] showed that these computational problems are PPAD-complete. The result was almost immediately tightened to encompass two-player games by Chen and Deng [2006]. The NP-completeness results for Nash equilibria with specific properties are due to Gilboa and Zemel [1989] and Conitzer and Sandholm [2003b]; the inapproximability result appeared in Conitzer [2006].

A general survey of the classical algorithms for computing Nash equilibria in 2-person games is provided in von Stengel [2002]. Another good survey is McKelvey and McLennan [1996]. Some specific references, both to these classical algorithms and to the newer ones discussed in the chapter, are as follows. The Lemke–Howson algorithm [Lemke and Howson, 1964] can be understood as a a specialization of Lemke's pivoting procedure for solving linear complementarity problems [Lemke, 1978]. The graphical exposition of the Lemke–Howson algorithm appeared first in Shapley [1974], and then in a modified version in von Stengel [2002]. Our description of the Lemke–Howson algorithm is based on the latter. An example of games for which *all* Lemke–Howson paths are of exponential length appears in Savani and von Stengel [2004]. Scarf's simplicial-subdivision-based algorithm is described in Scarf [1967]. Homotopy-based approximation methods are covered, for example, in García and Zangwill [1981]. Govindan and Wilson's homotopy method was presented in Govindan and Wilson [2003]; its path-following procedure depends on topological results due to Kohlberg and Mertens [1986]. The support-enumeration method for finding a sample Nash equilibrium is described in Porter et al. [2004a]. The complexity of iteratedly eliminating dominated strategies is described in Gilboa et al. [1989] and Conitzer and Sandholm [2005].

GAMBIT Two online resources are of particular note. *GAMBIT* [McKelvey et al., 2006] (`http://econweb.tamu.edu/gambit`) is a library of game-theoretic algorithms for finite normal-form and extensive-form games. It includes many different algorithms for finding Nash equilibria. In addition to several algorithms that can be used on general sum, n-player games, it includes implementations of algorithms designed for special cases, including two-player games, zero-sum games, and finding
GAMUT all equilibria. Finally, *GAMUT* [Nudelman et al., 2004] (`http://gamut.stanford.edu`) is a suite of game generators designed for testing game-theoretic algorithms.