Correlated Strategy And EqB.

Suppose there are two cars approaching a crossroads. The "game" in this case can be represented as follows :-

	Wait	Go
Wait	0,0	1,2
Go	2,1	-10,-10

Highlighted are PSNE.
However, in practice both players trust a mediator to guide them - the traffic light

Correlated Strategy

A correlated strategy is a mapping π such that (prob. distribution over strategy profiles)

$$\pi : S \to [0,1] \quad \wedge \quad \sum_{s \in S} \pi(s) = 1$$

- Note that the correlated strategy is common knowledge

filling space lol

Correlated Equilibrium

A strategy π s.t

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i})\, u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i})\, u_i(s_i', s_{-i}) \quad \forall i \in N,\ \forall s_i, s_i' \in S_i$$

You were suggested s_i by the mediator, but picked s_i' to become worse-off

You will never be better by deviating from π! (assuming others follow it)

Computing Correlated Equilibrium

Finding CE is simply solving a set of linear equations.
There are two sets of constraints :-

(Quite literally the defn)

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i})\, u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i})\, u_i(s_i', s_{-i}) \quad \forall s_i, s_i' \in S_i, \forall i \in N$$

$$\pi(s) \geq 0, \forall s \in S$$
$$\sum_{s \in S} \pi(s) = 1$$

π must be a prob. distribution

Theorem For every MSNE σ^*, there exists a CE π^*

Proof Simple, given σ^* we construct an equivalent π^*.

$$\forall s \in S,\ \pi^*(s) = \prod_{s_i \in s} \sigma^*(s_i)$$

SDSE
↓
WDSE
↓
PSNE
↓
MSNE
↓
CE

Extensive Form Games

Normal form games are not enough in cases such as chess, where players use history of the game so far in decision making as well.

That is, more suitable for multi-stage games

Perfect Information Extensive Form Games (PIEFG)

All players can look at history and movement of other players. Chess ✓ but Cards ✗

PIEFG - $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N - Set of players
- A - set of all possible actions
- H - set of histories satisfying
  - Empty $\phi \in H$
  - If $h \in H$, then every suffix also $\in H$
  - $h = (a^0, a^1, \dots, a^T)$ is terminal IF $\nexists (a^0, \dots, a^T, a^{T+1}) \in H$

  ($Z \subseteq H$, set of all terminal histories)
- X - $H \setminus Z \to 2^A$, the action selection function
- P - $H \setminus Z \to N$, player function
- u_i - $Z \to \mathbb{R}$, utility function defined only for terminal!

Strategy

The strategy of a player for an EFG is a tuple of actions at every history where the player plays.

$$S_i = X(h) : h \in H \setminus Z, P(h) = i$$

- A PIEFG can be converted into an NFG where player i's actions are given by a sequence $a^{n_1} a^{n_2} \dots$ where a^{n_1} - player i's action at node n_1,
  a^{n_2} - " node n_2

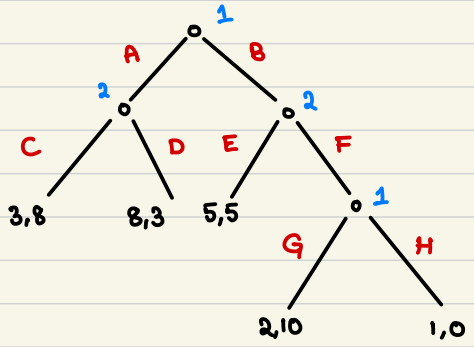
This has huge redundancy, and the PSNE computed need not be applicable

⇓

PIEFGs are here to stay!

Example

Consider the following PIEFG with two players:-



⇒ Strategies for player 1 - AG, AH, BG, BH
player 2 - CE, CF, DE, DF

SUBGAME

A game rooted at intermediate vertex

⇒ The subgame of G rooted at history h is limited to descendants of h.

SUBGAME PERFECTION - Finding best move at every subgame

SUBGAME PERFECT NASH EQUILIBRIUM

The SPNE of G are all strategy profiles $s \in S$ such that for any subgame G'

The restriction of s to G' is PSNE for G'

That is, we convert the PIEFG G to an NFG and find out if a common PSNE for all subgames exists

From the previous example :-

PIEFG converted to NFG, PSNE highlighted

	CE	CF	DE	DF
AG	3,8	3,8	8,3	8,3
AH	3,8	3,8	8,3	8,3
BG	5,5	2,10	5,5	2,10
BH	5,5	1,0	5,5	1,0

⇒ (AG, CF) – Is a PSNE ! ← This is SPNE

(AH, CF) – X for subgame $G-\overset{1}{o}-H$

(BH, CE) – "

ALGORITHM TO COMPUTE SPNE

```
func BACK_IND (history h):
    if h ∈ Z then
        return u(h), ∅
    best_util_P(h) = −∞
    for all a ∈ X(h):
        util_at_child_P(h) = BACK_IND((h,a))      ← Recurse to child's utility
        if util_at_child_P(h) > best_util_P(h):
            best_util_P(h) = util_at_child_P(h)   ┐
            best_action_P(h) = a                  ┘ better child found, replace!
    return best_util_P(h), best_action_P(h)
```

ADVANTAGES

- SPNE is guaranteed to exist in finite PIEFG
- SPNE is PSNE ⇒ PIEFG's guaranteed to have PSNE
- Algo. to find SPNE is quite simple

DISADVANTAGES

- Traversing whole tree is terrible
- Not representative of human cognitive limit

- SPNE traverses all nodes and subgames recursively. However, there may be multiple subgames that are not reached at all because of how the eqb. works.