Economic Equilibria in Type Theory

Cezar Ionescu (PIK) with Paul Flondor (Politechnica University of Bucharest), Edwin Brady (University of St. Andrews), and members of the Cartesian Seminar (PIK & Uni Potsdam)

Motivation

Integrated assessment models are used to answer questions such as "how will phasing out nuclear power plants affect Germany's unemployment?".

We focus on the economic models because

- 1. Most integrated assessment models contain an economic component.
- Economic models have more structure than some of the other components. (More opportunity to reuse software components).
- 3. There is more need for improving economic theory than physical theory.
- 4. My boss is an economist.

Game theory 101

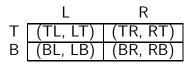
Example: A two-person finite game.

Strategy sets: first player can choose from $\{T, B\}$, second player from $\{L, R\}$.

A *play* consists of each agent making a choice from their strategy set.

Agents have *preferences* on plays (often induced by a *payoff function*).

Typical representation:



Essential question: what are "good" plays?

Game context

module GameContext" (nA' : Nat) where

nA : Nat nA = S nA'Agent : Set Agent = Fin nA

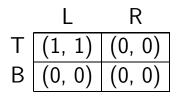
module GameContext' (*Strategy* : *Agent* \rightarrow *Set*) where

$$Play$$
 : Set
 $Play = (a : Agent) \rightarrow Strategy a$

module *GameContext* (*prefs* : (*a* : *Agent*) → *TotalPreorder Play*) **where**

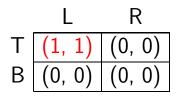
Some examples

Sometimes, it's clear what a good play is ...



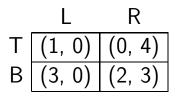
Some examples

Sometimes, it's clear what a good play is



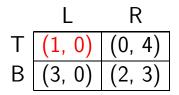
Some examples

Other times, it's clear what a bad play is ...



Some examples

Other times, it's clear what a bad play is ...



Pareto optimality (efficiency)

$$\begin{array}{ll} Pareto' & : \ Play \to Set \\ Pareto' \ p * & = \\ (\forall (p : Play) \to \\ \neg ((\ \forall (a : Agent) \ \to (prefs \ a) \vdash p \ge p*) \land \\ \exists (\lambda(a : Agent) \to (prefs \ a) \vdash p > p*))) \end{array}$$

$$\begin{array}{ll} \textit{Pareto} & : \ \textit{Play} \rightarrow \textit{Set} \\ \textit{Pareto} \ p * & = \\ & \forall \ (p : \textit{Play}) \rightarrow \\ & (\forall (a : \textit{Agent}) \rightarrow \quad (\textit{prefs } a) \vdash p * \geq p) \lor \\ & (\exists (\lambda(a : \textit{Agent}) \rightarrow (\textit{prefs } a) \vdash p * > p)) \end{array}$$

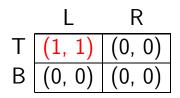
Computing Pareto points

For finite games, we can use brute-force enumeration or somewhat cleverer methods such as sieving.

◆□▶ ◆□▶ ◆ □▶ ★ □▶ = □ ● の < @

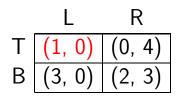
Pareto points

As expected ...



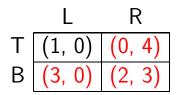
Pareto points

As expected ...



Pareto points

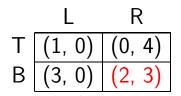
There's something unsatisfactory here ...



◆□▶ ◆□▶ ◆三▶ ◆三▶ - 三 - のへぐ

Pareto points

Not all Pareto points are equally plausible ...



Nash equilibria

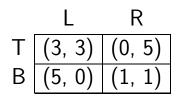
$$egin{aligned} \mathsf{Nash}': \mathsf{Play} &
ightarrow \mathsf{Set} \ \mathsf{Nash}' \ \mathsf{p*} = & \ & \forall \ (a: \mathsf{Agent}) &
ightarrow \forall (p: \mathsf{Play}) &
ightarrow & \ & \neg \ (\mathsf{prefs} \ \mathsf{a} dash (\mathsf{change} \ p * \mathsf{at} \ \mathsf{a} \ \mathsf{to} \ (p \ \mathsf{a})) > p) \end{aligned}$$

$$egin{array}{rll} Nash & : \ Play
ightarrow Set \ Nash \ p \ st & = \ orall \ (a : Agent)
ightarrow orall (p : Play)
ightarrow \ prefs \ a dash \ p \ st \geqslant (change \ p \ st \ a \ to \ (p \ a)) \end{array}$$

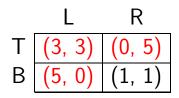
Computing Nash equilibria

For finite games, by enumeration etc.

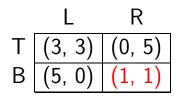
The prisoner's dilemma



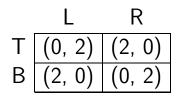
Pareto points for the prisoner's dilemma



Nash equilibrium for prisoner's dilemma

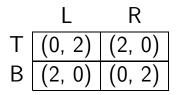


A game with no Nash equilibrium



Introducing mixed Nash equilibrium

We change the game: agents can choose probability distributions over their strategy sets.



Mixed Nash equilibrium

module *MixedStrategyContext* (*Strategy* : *Agent* \rightarrow *Set*; *payoff* : (*a* : *Agent*) \rightarrow ((*a* : *Agent*) \rightarrow *Strategy a*) \rightarrow *Float*) **where**

 $MixedStrategy : (a : Agent) \rightarrow Set$ MixedStrategy a = SimpleProb (Strategy a)

MixedPlay : Set $MixedPlay = (a : Agent) \rightarrow MixedStrategy a$

Mixed Nash equilibrium

 $expected_payoff : (a : Agent) \rightarrow MixedPlay \rightarrow Float$ $expected_payoff a mp = expected (fmap ((payoff a)))$ (mixedPlayToProbPlay mp))

prefs : (a: Agent) → TotalPreorder MixedPlay prefs a = InducedPreorder floatPreorder (expected_payoff a)

open GameContext' MixedStrategy open GameContext prefs renaming (Nash to MixedNash)

Computing mixed Nash equilibrium

... is hard! In fact, NP-hard (Goldberg and Papadimitriou, 2005).

◆□▶ ◆□▶ ◆ □▶ ★ □▶ = □ ● の < @

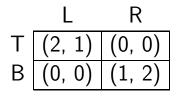
Mixed Nash equilibrium

$$p_L = 0.5, \ p_T = 0.5$$

◆□▶ ◆□▶ ◆ □▶ ◆ □▶ ● □ ● ● ● ●

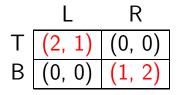
Nash equilibrium example

Example: BoS



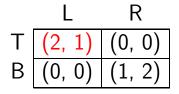
Nash equilibrium example

Pure Nash equilibria:



Nash equilibrium example

Problem: Pure equilibria are unfair!

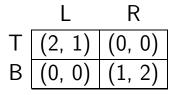


Nash equilibrium example

$$p_T = 2 / 3$$
, $p_R = 2 / 3$
Expected payoffs: $2 / 3$ each.

Nash equilibrium example

 $p_R = 2 / 3$, $p_T = 2 / 3$ Expected payoffs: 2 / 3 each: we cannot avoid miscoordination, if we want fairness.

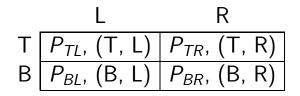


Introducing correlated equilibrium

We change the game again.

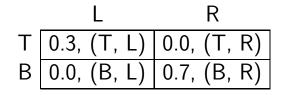
A new object: A simple probability distribution on plays.

Interpretation: A coordinator advises agents how to play.



Introducing correlated equilibrium

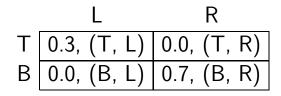
Example: avoiding miscoordination in BoS.



◆□> ◆□> ◆三> ◆三> ・三 のへで

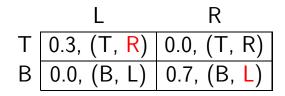
Introducing correlated equilibrium

Strategies: Agent *a* now chooses a function ϕ *a* from *Strategy a* \rightarrow *Strategy a*.



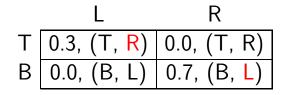
Introducing correlated equilibrium

Example: The first agent chooses $id : \{T, B\} \rightarrow \{T, B\}$, the second chooses $\neg : \{L, R\} \rightarrow \{L, R\}$.



Introducing correlated equilibrium

Payoffs are computed in the expected way.



Correlated equilibrium

```
module CorrelatedStrategyContext' (Strategy : Agent \rightarrow Set;
payoff : (a : Agent) \rightarrow ((a : Agent) \rightarrow Strategy a) \rightarrow Float)
where
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

open GameContext' Choice

Correlated equilibrium

module *CorrelatedStrategyContext* (*coord* : *SimpleProb Play'*) **where**

 $\begin{array}{ll} expected_payoff & : Agent \rightarrow Play \rightarrow Float \\ expected_payoff a p = expected (fmap (payoff a) \\ (fmap (change p) coord)) \\ \textbf{where} \\ change : Play \rightarrow Play' \rightarrow Play' \\ change p p' a = (p a) (p' a) \end{array}$

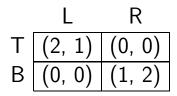
prefs : (a: Agent) → TotalPreorder Play prefs a = InducedPreorder floatPreorder (expected_payoff a)

open GameContext prefs renaming (Nash to CorrelatedEquilibrium)

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Correlated equilibrium example

 $P_{TL} = 1 / 2$, $P_{BR} = 1 / 2$ Expected payoffs: 3 / 2 each.



▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Games of exchange

The quintessential economic situation: exchange of goods.

- 1. Two agents, two goods, X units of the first good, Y units of the second.
- 2. Agent *i* has x_i unit of the first good, and y_i units of the second.
- 3. A distribution of goods to agents, such as $((x_1, y_1), (x_2, y_2))$ is called an *allocation*. Agents have preferences over allocations.
- 4. Agents are allowed to exchange their goods in order to find a better allocation: no throwing goods away, and no creation of goods: $x_1 + x_2 = X$, $y_1 + y_2 = Y$.

What is a good allocation?

Example: Cobb-Douglas economy

A typical example is the Cobb-Douglas economy, in which the agents preferences induced by the utility functions

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

$$u_1(x,y) = x^a y^{(1-a)}$$

$$u_2(x,y) = x^b y^{(1-b)}$$

where 0 < a, b < 1.

Introducing prices

If goods have prices p_x , p_y then an initial allocation gives each agent a *budget*:

$$B_i = p_x x_i + p_y y_i.$$

An agent has to solve:

maximize u(x, y) such that

$$p_x x + p_2 y = B_i$$

Whether the resulting allocation is feasible depends on the prices.

Walrasian equilibrium

An allocation-price pair (\mathbf{x}, \mathbf{p}) is a **Walrasian equilibrium** if (1) the allocation is feasible, and (2) each agent is making an optimal choice from its budget set. In equations:

1.
$$\sum_{i=1}^{n} \mathbf{x}_i = \sum_{i=1}^{n} \omega_i$$

2. If $\mathbf{x'}_i$ is preferred by agent *i* to \mathbf{x}_i , then $\mathbf{px'}_i > \mathbf{p}\omega_i$.

Varian, p. 325

Walrasian equilibria

 $\begin{array}{ll} \textit{params} (\textit{omega}:\textit{Vect}(\textit{Vect}\textit{Float}\textit{nG})\textit{nA},\\ \textit{prices}:\textit{Vect}\textit{Float}\textit{nG},\\ \textit{prefs}:\textit{Fin}\textit{nA} \rightarrow \textit{TotalPreorder}(\textit{Vect}\textit{Float}\textit{nG})) \end{array}$

Feasible : Vect (Vect Float nG) $nA \rightarrow Set$; *Feasible* xss = SumCols xss === SumCols omega;

```
Optimal : Vect (Vect Float nG) nA → Set;

Optimal xss = forall (i : Fin nA, xss' : Vect (Vect Float nG) nA) →

gt (prefs i) (vlookup i xss')

(vlookup i xss) →

gt floatOrder (prices .* (vlookup i xss'))

(prices .* (vlookup i xss));
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○三 のへ⊙

WalrasEq : Vect (Vect Float nG) $nA \rightarrow Set$; WalrasEq xss = And (Feasible xss) (Optimal xss);

Computing Walrasian equilibria

How can we compute Walrasian equilibria?

For the special case of the Cobb-Douglas economy, the solution can be computed analytically:

$$\frac{p_{y}}{p_{x}} = \frac{(1-a)x_{1}+(1-b)x_{2}}{ay_{1}+by_{2}}$$
$$x_{1}^{*} = \frac{B_{1}a}{p_{x}}$$
$$y_{1}^{*} = \frac{B_{1}(1-a)}{p_{y}}$$

In general, however, computing Walrasian equilibria involves a lot of numerical methods (optimization, solving linear systems, etc.).

Conclusions

We continue to develop the formalization of economic theory: local Nash equilibria (P. Flondor), general equilibrium, etc.

Work has begun on a DSL for numerical methods (E. Brady).

Ideally, one would like to have numerical methods implemented in terms of constructive reals, used in a constructive economic theory.