Decision Theory II
Open-Source Game Theory, Commitment Races, and Safe Pareto Improvements

Daniel C

Where we are: agents that read each other's code

Cooperation and conflict between embedded agents

- Deck I ended at **fair computational worlds**: agent and environment are both programs that may inspect, simulate, or predict each other. Embedded agents must reason about other agents – including agents as capable as themselves.
- This deck takes the strategic question: **when agents can read each other's source code, how do they cooperate without conflict, and without being exploitable?**
- The thread:
  - **Open-source game theory** – conditional commitment (FairBot, Löbian cooperation, simulation-based cooperation).
  - **Commitment races** – the incentive to commit first, its connection to s-risks, and entanglement.
  - **Safe Pareto improvements** – and entanglement-free SPI, which sidesteps the commitment race.

A 60-second refresher on game theory

Nash, and the Prisoner's Dilemma

- A (normal-form) game: players, action sets, and payoffs $u_i(a_1, \ldots, a_n)$ depending on the *joint* action.
- **Nash equilibrium:** a profile where no player gains by deviating *alone*.
- **Prisoner's Dilemma:** (C, C) is better for both than (D, D), yet D *dominates* C for each player. The unique Nash is (D, D): rational players defect, and both end up worse.
- **Root cause:** your payoff depends on the *joint* action, but you must choose *without knowing or affecting* the other's choice. You cannot *condition* on what they do.

		Player 2	
		C	D
P1	C	$3, 3$	$0, 5$
	D	$5, 0$	$1, 1$

(D, D) *is the only Nash, though* (C, C)

beats it for both.

Löbian cooperation: FairBot

When players can read each other's code

- **Open-source (program) game** (Tennenholtz): each player submits a *program* that takes the opponent's *source code* as input and outputs an action. Now you *can* condition on the opponent.
- **FairBot:** "cooperate iff I can *prove* my opponent cooperates with me."
- **FairBot vs. FairBot:** by **Löb's theorem**, each proves mutual cooperation and so cooperates. (Here Löb is a *feature*: $\Box(\Box C \to C) \to \Box C$, the self-referential proof search succeeds.) Two mutually suspicious agents cooperate, just by reading each other.
- **Unexploitable:** FairBot cooperates with CooperateBot, but *defects* against DefectBot (it cannot prove a defector cooperates). It is never the sucker.
- Caveat: proof-based FairBot is *brittle* – it depends on syntactic provability, can fail to prove cooperation with a differently-written but behaviourally-cooperative bot, and proof search may not terminate.

Robust cooperation by simulation: ε-grounded bots

Replacing proofs with simulation (Oesterheld)

- ε-**grounded FairBot:** with small probability ε, cooperate *unconditionally* ("ground out"); with probability $1 - \varepsilon$, *run the opponent's program* on your own code and *copy* its action.
- **Why it terminates:** each level of mutual simulation grounds out with probability ε, so the recursion halts with probability 1 – no infinite regress (contrast the naive "run each other" that hangs).
- Two ε-grounded FairBots (almost always) cooperate, and this is a **Nash equilibrium** of the program game. It is **robust**: programs that *behave* the same are treated the same (unlike syntactic proof-based bots).
- (The ε-grounded πBot and its generalisations, arXiv:2412.14570, prove folk theorems for such simulation-based equilibria.)

What open-source game theory is really doing

Conditional commitment

- Two desiderata for a good decision theory here:
  (1) **cooperate with your own copy**, and
  (2) **unexploitability** – an opponent cannot become better off by making you worse off.
- Why ordinary game theory fails: payoff depends on the joint action, but you are uncertain about the opponent's action and they about yours – a chicken-and-egg of *mutual uncertainty*.
- **Open-source game** = **conditional commitment:** "*I commit to X conditional on you committing to Y.*"
  - the **conditional** part keeps you *unexploitable* (you cooperate only if they do);
  - the **commitment** part makes your cooperation *legible*, so you can satisfy the conditions of the opponent's conditional commitment.
- FairBot is exactly this: "I commit to C conditional on a proof that you commit to C."

The commitment races problem

Why consequentialists race to commit (Kokotajlo)

- A consequentialist (acts purely on expected consequences) is both a **bully and a coward**: it threatens when threats work, and *caves* to credible threats. So **whoever commits first controls the one who commits later**.
- This creates a **race**: it pays to commit *before* you have thought or learned – to "move first". For agents that read or simulate each other, even *thinking longer* leaks exploitable information about what you will do.
- Two catastrophe modes:
  - **Incompatible commitments lock in:** both players throw out the steering wheel in a game of Chicken ⇒ a crash neither wanted (e.g. grim-trigger protocols keyed to incompatible bargaining solutions).
  - **Wasteful racing:** everyone commits before learning – a coordination failure like an arms race.

Moving first in logical time; and s-risks

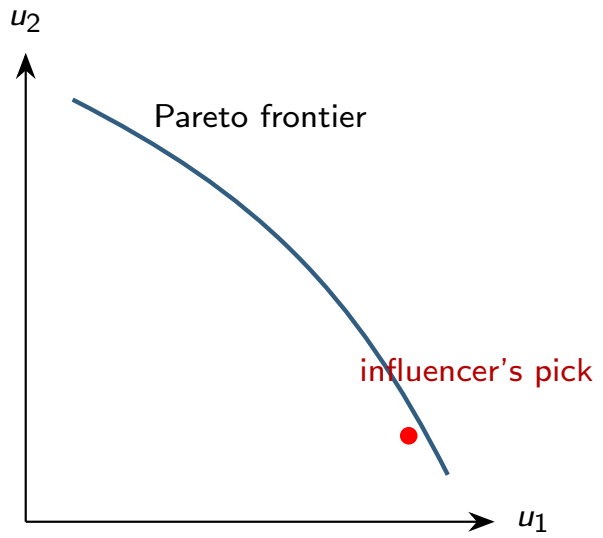
Why this is a safety priority

- "First" is not clock time but **logical time**: who, in the structure of mutual prediction, effectively decides *before* the other. An agent that is *predicted* to have committed *is* committed – even simultaneously in clock time.
- **Connection to s-risks (risks of astronomical suffering).** The most dangerous commitments are *threats*: "do X or I make things very bad for you." Commitment races can lock superintelligences into *carrying out* catastrophic threats (or executing them to build credibility), i.e. suffering on an astronomical scale.
- So a decision theory that avoids commitment races – letting agents reach good outcomes *without* a destructive scramble to pre-commit – is a genuine safety goal, not a curiosity.

Influencer vs. responder

The first-mover's advantage, and the race it triggers

- **Responder:** run the opponent's program, observe its action, then arg max your own utility against that action (a plain *best response*).
- **Influencer:** *assume the opponent is a responder* (will best-respond to you); arg max your utility over that – i.e. pick the point on the Pareto frontier **best for you and worst for the opponent**, knowing they will cave.
- **"The best response is not the best response":** being a naive best-responder is *dominated* by being the influencer. So everyone wants to be the influencer $\Rightarrow$ a race to "move first in logical time".



moves first $\Rightarrow$ grabs its favourite corner of the frontier

The generalized commitment race: entanglement

Why agents are not incentivised to make Pareto improvements

- When you choose your program in an open-source game, you optimise against your *subjective distribution* over the opponent's possible programs – including **counterfactual** ones (programs they might have submitted but didn't).
- So you cannot simply maximise against the opponent's *actual* program: doing so might make you worse off against *counterfactual* versions. Your incentive is **entangled** with those counterfactual programs.
- Equivalently: whenever a rational agent "*punishes*" an opponent (for, say, proposing a Pareto improvement), it is *forgoing utility against the actual program* to gain utility against some *counterfactual* program.
- **Entanglement is the obstacle:** it is exactly what disincentivises accepting a Pareto improvement (accepting might look "soft" and cost you against a counterfactual hawk). This is the *generalized* commitment race – it persists even with full conditional commitment.

Safe Pareto improvements (SPI)

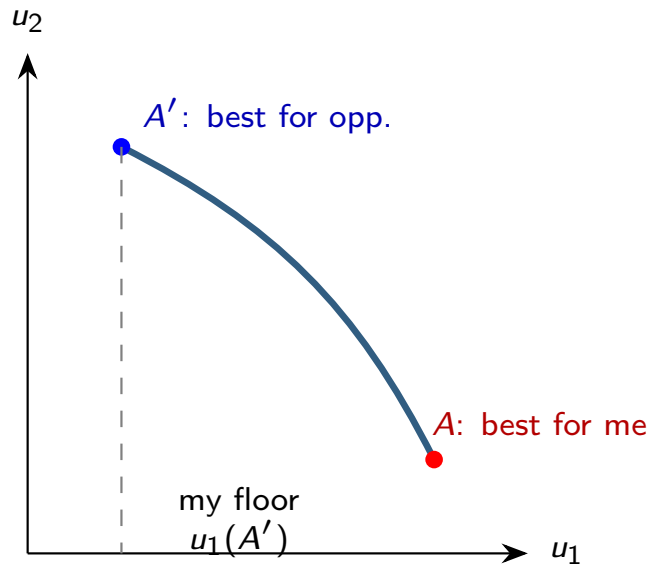
Improving on a default that risks conflict

- Setup (Oesterheld–Conitzer; DiGiovanni–Clifton–Macé, arXiv:2403.05103): players delegate to *representative* programs. The *default* instruction is "play the original game as best you can".
- An **SPI** is a re-instruction (restrict the strategy set, or re-map utilities) that is **guaranteed to make every player weakly better off** – with *no* probabilistic assumptions about how the representatives play. A Pareto improvement that is *safe* because it improves *regardless* of the opponent.
- Example: a game heading for costly conflict (Chicken-style). Agree on a transformation that strips out the mutually destructive outcome; each player does at least as well, whatever happens.
- The paper proves a clean guarantee – the **Pareto Meet Minimum (PMM)**: each player gets at least their *lowest payoff over efficient (Pareto-optimal) outcomes*, and this bound is *tight* (renegotiation cannot promise more). Next slide: why. Caveat: it needs a strong assumption.

Why an SPI guarantees a floor: the Pareto Meet Minimum

The shape of the renegotiation set controls exploitation

- Each agent's *renegotiation set* is the Pareto-improvements it will accept. On the frontier, "better for the opponent" *means* "worse for me".
- Let A be the frontier point **best for me** (worst for the opponent). I have every reason to include points *up to* A: pushing the opponent below A's level cannot help me – by definition A already maximises *my* payoff on the frontier.
- But including points *beyond* A (where I would accept being worse than at A) enables **exploitation**: the opponent could make herself better off by making me worse off.
- So rational renegotiation sets stop at each player's A-point. The worst the agreed outcome can be for me is the *opponent's* A-point – my **lowest efficient payoff**. That guaranteed floor is the **PMM** (and it is tight).



accept the frontier between A' and A, never beyond

The assumption SPI relies on – and the cheating problem

Participation independence

- **Participation independence:** representatives behave the *same* in the SPI-transformed game as in the default game – their play does not change just because an SPI is in effect. This independence is what lets you *guarantee* the improvement.
- But it is unrealistic. **The cheating problem:** if I know that an SPI will Pareto-improve away any conflict, then conflict is *less costly* for me, so I am incentivised to submit a *more hawkish* default program.
- Anticipating my hawkishness, my opponent's incentive to offer the SPI in the first place is undermined. **The SPI can defeat itself** – precisely the entanglement effect, reappearing.

Entanglement-free SPI: the idea

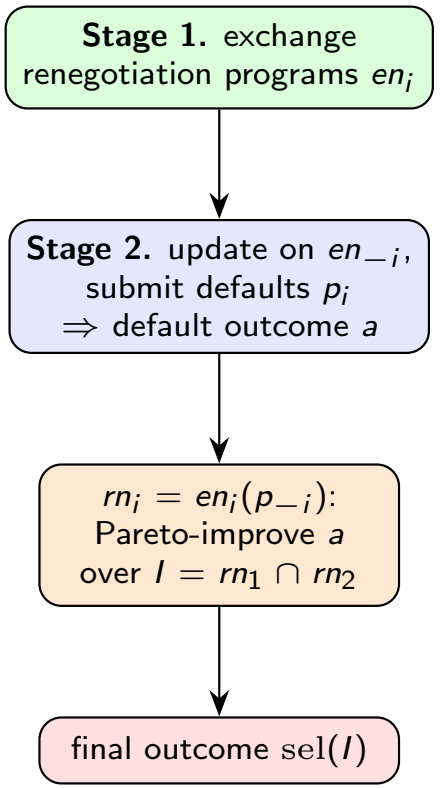
Make the improvement with an entanglement-free program

- Diagnosis: *entanglement* (caring about counterfactual opponent programs) is what disincentivises Pareto improvements. So make the improvement using a separate program that is **entanglement-free** – the **renegotiation program**.
- **Sequential structure:** agents *first* exchange renegotiation programs and *update on the actual one*; *then* submit their default programs.
- Because you update on the opponent's *actual* renegotiation program, you optimise against *it*, not against counterfactual versions $\Rightarrow$ *no entanglement* $\Rightarrow$ no incentive to punish/reject improvements.
- The renegotiation program may *only propose Pareto improvements* (conditional on the opponent's default program). Two consequences:
  - it can *never make the opponent worse off* $\Rightarrow$ no influencer-responder exploitation, *even though* it moves first;
  - it *conditions on the default program* $\Rightarrow$ it can *detect* hawkish "cheating" defaults and withhold improvements, restoring unexploitability and **removing reliance on participation independence**.
- You keep your full action space: you can still play the open-source game with your default program; the renegotiation program only *adds* the option to Pareto-improve.

Entanglement-free SPI: the protocol

A clean formalism

- Open-source game: actions $A = A_1 \times A_2$, utilities $u_i : A \to \mathbb{R}$, programs $p_i : P_{-i} \to A_i$, joint action $a = (p_1(p_2),\ p_2(p_1))$.
- **Renegotiation function** $rn_i : RN_{-i} \times A \to \mathcal{C}(A)$: conditional on the opponent's renegotiation function, it proposes a *set* of Pareto-improvements on the default outcome a. A rule sel picks one point of the agreed intersection.
- **Entanglement-free renegotiation program** $en_i : P_{-i} \to RN_i$ reads the opponent's *default* program, outputs a renegotiation function.
- The intersection $I = rn_1(rn_2, a) \cap rn_2(rn_1, a)$ holds the Pareto-improvements *both* agents will make; $\mathrm{sel}(I)$ is the final outcome. (Choosing sel adds no coordination problem, under mild conditions – DiGiovanni.)



Why first-mover is safe here – and what we trade away

Resolving the race without creating a new one

- Usually, a sequential structure creates an influencer-responder race (the first mover grabs the frontier). Here the first move (renegotiation) can *only Pareto-improve*, so it **cannot exploit** – no new commitment race is introduced.
- Net result: entanglement-free SPI **solves the cheating problem** and **drops the participation-independence assumption** (the renegotiation program conditions on the defaults to catch cheating), while the sequential structure **avoids introducing its own commitment race**.
- **The trade-off – no tight PMM.** The renegotiation program conditions on the default program, so it can create an *incentive gradient*: "more cooperative default $\Rightarrow$ I offer more Pareto-improvement; more hawkish $\Rightarrow$ less". Because it may *withhold* improvement to shape the opponent's default, the agreed set is no longer the full "up-to-A" set, and the tight **Pareto Meet Minimum** floor is lost. Weaker (more realistic) assumptions buy a weaker guarantee.

(Entanglement-free SPI: Daniel C; selection-function coordination after DiGiovanni.)