

Specification of Dynamic Strategy Switching

(Extended Abstract)

Soumya Paul

The Institute of Mathematical Sciences, Chennai

Consider the game of cricket. During the course of the game there are many strategic reasonings and inferences that take place on the part of the players. From the bowler's point of view, she has to strategise what line and length to bowl to a batter during the course of her over. If the batter is weak playing deliveries of certain line and length, the bowler has to try and exploit this weakness. But there is also a danger of becoming too predictable in the process. Hence she has to continuously keep varying her bowling strategies. On the other hand, the batter also applies her own reasoning. She may be able to hit a weak bowler for a boundary every ball of the over, but she may just be a bit conservative and hit her only for enough runs so that the bowler is not replaced by another and the batter can reap more benefits in her next over. Such decisions and strategising take place dynamically during the course of the game based on the observation of the outcome of the game so far.

In traditional game theory, strategies are viewed as black boxes (functions from histories to actions) and nothing much is considered about the 'structure' of strategies. The planning element in strategies is completely missing from such an approach. But if we look at the above example and almost every other game in day-to-day life, a player playing the game does not start with a rigid strategy in mind but keeps revising the same as and when the game progresses based on what she can observe of the game being played so far. Thus arises the motivation of looking at structured strategies.

We propose a model of unbounded duration games in which players dynamically switch strategies not only based on observed outcomes but also on anticipated switching by other players. Strategies for us are partial functions from the set of finite histories of the game to the set of actions of the players. At the outset of a game, a player may start playing with a small (possibly finite) set of 'atomic' strategies and combine them, as the game progresses, to generate more and more complex strategies. To describe the above process we suggest the following syntax. If σ is an atomic strategy (partial function from the set of histories to the set of actions) of the player, then the following are also valid strategies:

$$\Pi ::= \sigma \mid \pi_1 \cup \pi_2 \mid \pi_1 \cap \pi_2 \mid \pi_1 \wedge \pi_2 \mid (\pi_1 + \pi_2) \mid \psi? \pi$$

where ψ is an observable property of the game or a boolean combination of such observables. The intuitive meaning of the above operators are as follows:

- $\pi_1 \cup \pi_2$ means that the player plays according to the strategy π_1 or the strategy π_2 .

- $\pi_1 \cap \pi_2$ means that if at a history of the game, π_1 is defined then the player plays according to π_1 ; else if π_2 is defined at that history then the player plays according to π_2 . If both π_1 and π_2 are defined then the moves that π_1 and π_2 specify at that history must be the same.
- $\pi_1 \frown \pi_2$ is the critical operator in our specification. We call it the ‘chop’ operator. It means that the player plays according to the strategy π_1 and then after some history, switches to playing according to π_2 . The position at which she makes the switch is not fixed in advance.
- $(\pi_1 + \pi_2)$ is similar to the chop operator and says that at every point, the player can choose to follow either π_1 or π_2 .
- $\psi? \pi$ says at every history, the player tests if the property ψ holds of that history. If it does then she plays according to π .

If now the atomic strategies are bounded memory then they can be simulated by finite state transducers (FSTs). From these FSTs, we construct FSTs for the complex strategies as follows. Let \mathcal{A}_π denote the FST for the strategy π . Then

- $\mathcal{A}_{\pi_1 \cup \pi_2}$ is just the disjoint union of \mathcal{A}_{π_1} and \mathcal{A}_{π_2} .
- $\mathcal{A}_{\pi_1 \cap \pi_2}$ is the intersection of \mathcal{A}_{π_1} and \mathcal{A}_{π_2} .
- $\mathcal{A}_{\pi_1 \frown \pi_2}$ is more interesting. It simulates \mathcal{A}_{π_1} and \mathcal{A}_{π_2} in parallel. It starts by mirroring the output of \mathcal{A}_{π_1} and then non-deterministically switches to mirroring the output of \mathcal{A}_{π_2} .
- $\mathcal{A}_{(\pi_1 + \pi_2)}$ at every point non-deterministically decides whether to mirror the output of \mathcal{A}_{π_1} or that of \mathcal{A}_{π_2} .
- $\mathcal{A}_{\psi? \pi}$ at every history tests the property ψ of that history. If ψ holds then it mimics \mathcal{A}_π .

As an example, for our game of cricket, consider the initial strategy set for a bowler, $\Sigma_b = \{\sigma_5, \sigma_2, \dots\}$ where σ_5 says “bowl a slower delivery on the 5th ball of every over” and σ_2 says “bowl a slower delivery on the 2nd ball of every over”. Then a bowler may adopt the strategy $\sigma_5 \frown \sigma_2$ in the above syntax, which means that he nondeterministically switches from bowling a slower ball on the 5th ball of every over to bowling a slower ball on the 2nd ball of every over from the fear of becoming too predictable!

We call a strategy ‘switch-free’ if it does not have any of the operators \frown , $+$ or $?$.

Given a game arena G (which is just a directed graph) with a valuation of a set of observable propositions at every node, a subarena R of G and strategy specifications π_1, \dots, π_n of players 1 to n , we may ask the questions:

1. Does the game settle down to R ?
2. If so, then does the strategy of player i become eventually stable with respect to switching?

We show that for a finite arena and bounded memory strategies, the above questions are decidable and give the complexity of our decision procedures in terms of the size of the arena and the given specifications.

This is joint work with R. Ramanujam and Sunil Simon.