Recursive Lexicographical Search: Finding all Markov Perfect Equilibria in Directional Dynamic Games

Fedor Iskhakov, University of New South Wales
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

National University of Singapore
August 11, 2015

Recursive Lexicographic Search Algorithm

Building blocks of RLS algorithm:

1. State recursion algorithm solves the game *conditional on* equilibrium selection rule (ESR)
2. RLS algorithm efficiently cycles through *all feasible* ESRs

Properties of RLS algorithm:

- Complete: Computes all MPE equilibria of the game
- Fast: time spent of search of feasible ESRs is negligible in comparison to time spent on solving the game
  - Efficiently skip infeasible ESRs
  - Re-use results of previously computed subgames

Assumptions of RLS

1. There is a method to find *all* MPE in every d-stage game (i.e. equilibria within the class of continuation strategies)
2. The number of equilibria in every d-stage game is finite
3. $\Rightarrow$ RLS finds *all* MPE of the DDG $\mathcal{G}$.

- RLS also works if this algorithm can only find *some* of the equilibria of d-stage games.
- In the latter case RLS is not guaranteed to find *all* MPE of $\mathcal{G}$, but it can still find, potentially, *very many* MPE of $\mathcal{G}$.

Efficiency of RLS

1. **[Decompose]** State recursion decomposes a large, complex game into a sequence of much simpler component stage games
2. **[Reuse]** RLS minimizes the computational cost when if traverses the set of all MPE
3. **[Jump]** RLS is caplable to jump from one feasible ESR to next in one step

Coming up next: RLS algorithm

1. Formal representation of equilibrium selection rules
2. Feasible equilibrium selection rules and jump function
3. Variable base arithmetic and successor function
4. Relationship between jump and successor functions

$\Rightarrow$ Recursive lexicographical search algorithm

Equilibrium Selection Strings (ESS)

ESS formalizes the ESR Γ

- K – the least upper bound on the number of possible equilibria in any stage game of $\mathcal{G}$.
  Implementation does not require the prior knowledge of K
- N – the total number of d-substages of the DDG $\mathcal{G}$.
  The state recursion algorithm must loop over all N of these substages to find a MPE in the stage games that correspond to each of these N d-stages to construct a MPE of $\mathcal{G}$.
- Equilibria in every d-stage games are indexed $\{0, 1, \ldots, K-1\}$

Equilbrium Selection Strings (ESS), continued

Definition (Equilibrium Selection Strings)

An *equilibrium selection string* (ESS), denoted by γ, is a vector in Z_+^N whose components are integers expressed in base K arithmetic, i.e. each coordinate (or "digit") of γ takes values in the set $\{0, 1, \dots, K-1\}$.

$$\gamma = (\gamma_{\mathcal{T}}, \gamma_{\mathcal{T}-1}, \dots, \gamma_1),$$

$$\gamma_\tau = (\gamma_{1,\tau}, \dots, \gamma_{n_\tau,\tau})$$

$\gamma^0 = (0, \dots, 0)$ – the initial ESS that consists of N zeros, always feasible assuming at least one equilibrium exists in every d-stage game

Ordering of stages/digits in ESSs

- Elements of γ are ordered in a particular way
- γ_τ are ordered from right to left corresponding to stages of $\mathcal{G}$ from τ to $\mathcal{T}$
- Higher digits of ESS correspond to later stages of $\mathcal{G}$
- $\Rightarrow$ when digit j is changed in the loop over ESS, only stages to the right ($\tau < j$) have to be resolved

Enumerating all possible Equilibrium Selection Strings

- K^N possible equilibrium strings for the DDG $\mathcal{G}$
  this is an upper bound on the number of possible MPE of $\mathcal{G}$

- The loop starts from $\gamma^0 = (0, \ldots, 0)$
- One step in the loop is mod(K) addition $+1$
- Second ESR is $\gamma^1 = (0, \ldots, 0, 1) = 1$ in base-K representation

Does γ^1 correspond to an MPE of $\mathcal{G}$?

- $\gamma_1 = (0, 0, \ldots, 0, 1)$ may or may not correspond to a MPE of $\mathcal{G}$ because there may be only a *single* MPE at the d_{1,n_1}-stage game of $\mathcal{G}$.
- If there is only a single MPE in this substage $\rightarrow$ the equilibrium string γ_1 is *infeasible* because the only feasible equilirbrium index for the first d-stage is 0

Definition (Feasible Equilibrium Selection String)

An equilibrium string γ is *feasible* if all of its digits index a MPE that exists at each of the corresponding d-stage games of $\mathcal{G}$.

Tracking the number of MPE at each substage

Definition ($ne(\gamma)$ string)

Let the $N \times 1$ vector $ne(\gamma)$ be the maximum number of MPE at each stage game of $\mathcal{G}$ under the ESR implied by the equilibrium string γ. Define $ne(\gamma)$ using the same format as the equilibrium string, so that the digits of the equilibrium string γ are in one to one correspondence with the elements of the vector $ne(\gamma)$ as follows:

$$ne(\gamma) = \Big(ne_{\mathcal{T}}, ne_{\mathcal{T}-1}(\gamma_{>\mathcal{T}-1}), \dots, ne_1(\gamma_{>1}) \Big),$$

where $\gamma_{>\tau} = (\gamma_{\tau+1}, \dots, \gamma_{\mathcal{T}})$ is a $\mathcal{T} - \tau \times 1$ vector listing the equilibrium selection sub-string for stages of $\mathcal{G}$ higher than τ.

Characterization of Feasible ESSs

- We use the notation $ne_\tau(\gamma_{>\tau})$ to emphasize that the number of MPE at stage τ depends only on the equilibria selected at higher stages of $\mathcal{G}$.
- Notice that in the endgame $\mathcal{T}$ there are no further stages of the game, so the maximum number of MPE in this stage, $n_\mathcal{T}$ does not depend on any substring of the equilibrium string γ.

Lemma

The ESS γ is feasible if and only if

$$\gamma_{i,\tau} < ne_{i,\tau}(\gamma_{>\tau}), \quad \tau = 1, \ldots, \mathcal{T}, \quad i = 1, \ldots, n_\tau$$

The Jump Function

$\mathcal{J}(\gamma)$ is the "smallest" ESS *after* γ that is also a feasible ESS.

Definition (Jump function)

Let $\mathcal{J} : Z_+^N \to Z_+^N \cup [\mathrm{STOP}]$ be defined by

$$\mathcal{J}(\gamma) = \begin{cases} \mathit{argmin}_{\gamma'}\{\gamma' : \ \gamma' > \gamma \ \text{ and } \gamma' \text{ is feasible}\} \\ [\mathrm{STOP}] \ \text{ if } \nexists \gamma' : \ \gamma' > \gamma \ \text{ and } \gamma' \text{ is feasible} \end{cases}$$

Variable base arithmetics

- Replace the base-K ($mod(K)$) arithmetics with variable base arithmetics
- Let (3 1 2) be bases $\rightarrow$
- Allowed digits in the numbers are $\{0, 1, 2\}$, $\{0\}$ and $\{0, 1\}$
- The 3-digit numbers in this system are:

$$\begin{array}{lll} 0\ 0\ 0 & +1 & \rightarrow \\ 0\ 0\ 1 & +1 & \rightarrow \\ 1\ 0\ 0 & +1 & \rightarrow \\ 1\ 0\ 1 & +1 & \rightarrow \\ 2\ 0\ 0 & +1 & \rightarrow \\ 2\ 0\ 1 & & \end{array}$$

ESS γ in variable base arithmetics

- Bases: $ne(\gamma) = \Big(ne_{\mathcal{T}}, ne_{\mathcal{T}-1}(\gamma_{>\mathcal{T}-1}), \dots, ne_1(\gamma_{>1})\Big)$
- Successor function $\mathcal{S} : Z_+^N \to Z^N : \; S(\gamma) = \gamma + 1$
- Defined correctly in variable bases:

$$\gamma+1 = \begin{cases} (\quad \dots \quad , \gamma^{(1)} + 1) & \text{if } \gamma^{(1)} + 1 < ne^{(1)}, \\ (\dots, \gamma^{(j-k)+1}, 0, \dots, 0) & \text{otherwise,} \end{cases}$$

where $k : \; \gamma^{(j-k)+1} < ne^{(j-k)}$

- In the latter case dependent bases change to $\big(ne^{(1)}, \dots, ne^{(j-k)}, \tilde{ne}^{(j-k+1)}, \dots, \tilde{ne}^{(N)}\big)$
- But $\gamma + 1$ is still a well-defined number in this new bases because all digits with new bases are zeros

Relation between the Successor and Jump Function

- $\mathcal{S}'$ – augmented with $[STOP]$ signal when more than N digits are needed for the successor result

Theorem

Assume that ESS are expressed in variable bases as above.
If γ is a feasible ESS, then $\mathcal{J}(\gamma) = \mathcal{S}'(\gamma)$.

- In other words, when the bases for the equilibrium selection string are chosen in the right way, the function that returns next feasible ESS is just a successor function
- The number of steps RLS takes is the same as number of MPE in the model!

RLS Algorithm

1. Set $\gamma^0 = (0, \ldots, 0)$, $k = 0$
2. Solve for an MPE given the ESS γ^k with State Recursion
3. Fix the bases for the ESS at computed $ne(\gamma)$
4. Apply a successor function to find next feasible ESS

$$\gamma^{k+1} = S'(\gamma^k)$$

5. Stopping rule: $\gamma^{k+1} \stackrel{?}{=} [\mathrm{STOP}]$
6. Return to step 2

Main result of the RLS Algorithm

Theorem (Decomposition theorem)

Assume there exists an algorithm that can find all MPE of every stage game of the DDG $\mathcal{G}$, and that the number of these equilibria is finite in every stage game.
Then the RLS algorithm finds all MPE of DDG $\mathcal{G}$ in at most $|\mathcal{E}(\mathcal{G})|$ steps, which is the total number of MPE of the DDG $\mathcal{G}$.