Lecture 22: Solving directional dynamic games for all Markov perfect equilibria

Australian Summer School in Dynamic Structural Econometrics

Fedor Iskhakov, Australian National University
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

ANU, Canberra
December 19, 2022

ROAD MAP

1. Collusion of Australian corrugated fibre packaging (CFP) producers
   - Collusion between Amcor and Visy
   - Bertrand pricing and investment game
   - Solution concept: Markov perfect equilibrium (MPE)
2. Experiment with the model
3. State recursion algorithm
   - Theory of directional dynamic games (DDGs)
4. Recursive lexicographical search (RLS) algorithm
5. Full solution for the leapfrogging game
6. Structural estimation of directional dynamic games with Nested RLS method

Estimation of directional dynamic games:
Full solution nested MLE estimation

Nested Recursive Lexicographic Search algorithm

Markov Perfect Equilibria

- MPE is a pair of strategy profile and value functions
- In compact notation

$$\begin{aligned} \mathsf{V} &= \Psi^{\mathsf{V}}(\mathsf{V}, \mathsf{P}, \theta) \\ \mathsf{P} &= \Psi^{\mathsf{P}}(\mathsf{V}, \mathsf{P}, \theta) \end{aligned}$$

- Set of all Markov Perfect Equilibria

$$SOL(\Psi, \theta) = \left\{ (\mathsf{P}, \mathsf{V}) \,\middle|\, \begin{array}{l} \mathsf{V} = \Psi^{\mathsf{V}}(\mathsf{V}, \mathsf{P}, \theta) \\ \mathsf{P} = \Psi^{\mathsf{P}}(\mathsf{V}, \mathsf{P}, \theta) \end{array} \right\}$$

- $\Psi^{\mathsf{V}}: \ \mathsf{V}, \mathsf{P} \longrightarrow \mathsf{V}$ Bellman operator
- $\Psi^{\mathsf{P}}: \ \mathsf{V}, \mathsf{P} \longrightarrow \mathsf{P}$ Choice probability formulas (logit)
- $\Gamma: \ \mathsf{P} \longrightarrow \mathsf{V}$ Hotz-Miller inversion

Estimation methods for *dynamic* stochastic games

- Two step (CCP) estimators
  - Fast, potentially large finite sample biases

Hotz, Miller (1993); Altug, Miller (1998); Pakes, Ostrovsky, and Berry (2007); Pesendorfer, Schmidt-Dengler (2008)

1. Estimate CCP $\rightarrow \hat{\mathsf{P}}$
2. Method of moments • Minimal distance • Pseudo likelihood

$$\min_{\theta} \left[\hat{\mathsf{P}} - \Psi^{\mathsf{P}}(\Gamma(\theta, \hat{\mathsf{P}}), \hat{\mathsf{P}}, \theta)\right]' W \left[\hat{\mathsf{P}} - \Psi^{\mathsf{P}}(\Gamma(\theta, \hat{\mathsf{P}}), \hat{\mathsf{P}}, \theta)\right]$$

$$\max_{\theta} \mathcal{L}(\mathsf{Z}, \Psi^{\mathsf{P}}(\Gamma(\theta, \hat{\mathsf{P}}), \hat{\mathsf{P}}, \theta))$$

- Nested pseudo-likelihood (NPL)
  - Recursive two step pseudo-likelihood
  - Bridges the gap between efficiency and tractability
  - Unstable under multiplicity

Aguirregabiria, Mira (2007); Pesendorfer, Schmidt-Dengler (2010); Kasahara and Shimotsu (2012); Aguirregabiria, Marcoux (2021)

Estimation methods for *dynamic* stochastic games

- Equilibrium inequalities (BBL)
  - Minimize the one-sided discrepancies
  - Computationally feasible in large models

  Bajari, Benkard, Levin (2007)

- Math programming with equilibrium constraints (MPEC)
  - MLE as constrained optimization
  - Does not rely on the structure of the problem
  - Much bigger computational problem

  Su (2013); Egesdal, Lai and Su (2015)

$$\max_{(\theta,\mathsf{P},\mathsf{V})} \mathcal{L}(\mathsf{Z},\mathsf{P}) \text{ subject to } \mathsf{V} = \Psi^{\mathsf{V}}(\mathsf{V},\mathsf{P},\theta), \mathsf{P} = \Psi^{\mathsf{P}}(\mathsf{V},\mathsf{P},\theta)$$

- All solution homotopy MLE

  Borkovsky, Doraszelsky and Kryukov (2010)

Overview of NRLS

- Robust and *computationally feasible*$^{(?)}$ MLE estimator for directional dynamic games (DDG)
- Rely of full solution algorithm that provably computes all MPE under certain regularity conditions
- Employ smart discrete programming method to maximize likelihood function over the finite set of equilibria
- Fully robust to multiplicity of MPE
- Relax single-equilibrium-in-data assumption
- Path to estimation of equilibrium selection rules

Nested Recursive Lexicographical Search (NRLS)

- Data from M independent markets from T periods
  $\mathrm{Z} = \{\bar{\mathrm{a}}^{mt}, \bar{\mathrm{x}}^{mt}\}_{m \in \mathcal{M}, t \in \mathcal{T}}$
  Usually assume only one equilibrium is played in the data.
- Denote θ $\left(\mathrm{P}^{\ell}(\theta), \mathrm{V}^{\ell}(\theta)\right) \in SOL(\Psi, \theta)$ the ℓ-the equilibrium

1. Outer loop
   Maximization of the likelihood function w.r.t. to structural parameters θ
   $$\theta^{ML} = \arg\max_{\theta} \mathcal{L}(Z, \theta)$$
2. Inner loop
   Maximization of the likelihood function w.r.t. equilibrium selection
   $$\mathcal{L}(Z, \theta) = \max_{(\mathrm{P}^{\ell}(\theta), \mathrm{V}^{\ell}(\theta) \in SOL(\Psi, \theta)} \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P_i^{\ell}(\bar{a}_i{}^{mt} | \bar{\mathrm{x}}^{mt}; \theta)$$

Max of a function on a discrete set organized into RLS tree

Branch and bound (BnB) method

Land and Doig, 1960 *Econometrica*

- Old method for solving discrete programming problems

1. Form a tree of subdivisions of the set of admissible plans
2. Specify a bounding function representing the best attainable objective on a given subset (branch)
3. Dismiss the subsets of the plans where the bound is below the current best attained value of the objective

Theory of BnB: branching

$$\max f(x) \text{ s.t. } x \in \Omega$$

$f(x)$ objective function

Ω set of feasible x

$\mathcal{P}_j(\Omega)$ partition of Ω into k_j subsets, $\mathcal{P}_1(\Omega) = \Omega$

$$\mathcal{P}_j(\Omega) = \{\Omega_{j1}, \ldots, \Omega_{jk_j} : \ \Omega_{ji} \cap \Omega_{ji'} = \varnothing, i \neq i', \ \cup_{i=1}^{k_j} \Omega_{ji} = \Omega\}$$

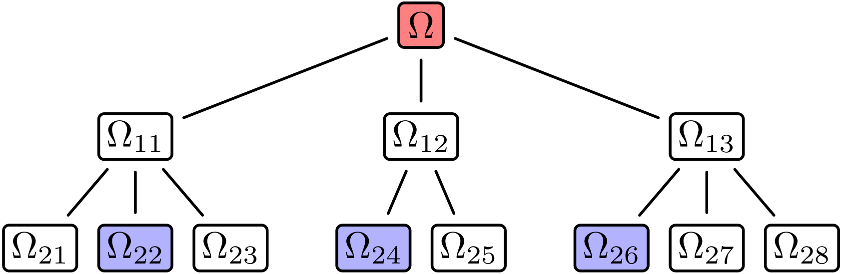
$\{\mathcal{P}_j(\Omega)\}_{j=1,\ldots,J}$ a sequence of J gradually refined partitions

$$k_1 \leq \cdots \leq k_j \leq \cdots \leq k_J$$

$$\forall j = 1, \ldots, J, \forall i = 1, \ldots, k_j : \ \forall j' < j \ \exists i'_{j'} \text{ such that } \Omega_{ij} \subset \Omega_{i'j'}$$

Theory of BnB: branching

$$\max f(x) \text{ s.t. } x \in \Omega$$



Theory of BnB: bounding

$$\max f(x) \text{ s.t. } x \in \Omega$$

$g(\Omega_{ij})$ bounding function: from subsets of Ω to real line

$g(\Omega_{ij}) = f(x)$ for singletons, i.e. when $\Omega_{ij} = \{x\}$

Monotonicity of bounding function

$$\forall j \; \forall \Omega_{i_1 1} \supset \Omega_{i_2 2} \supset \cdots \supset \Omega_{i_j j}$$

$$g(\Omega_{i_1 1}) \geq g(\Omega_{i_2 2}) \geq \cdots \geq g(\Omega_{i_j j})$$

- Inequalities are reversed for the minimization problem

BnB with NRLS

- **Branching**: RLS tree
- **Bounding**: The bound function is partial likelihood calculated on the subset of states that

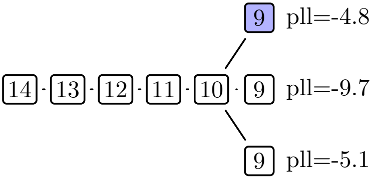
$$\mathcal{L}^{\text{Part}}(Z, \theta, \mathcal{S}) \quad = \quad \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P_i^{\ell}(\bar{a}_i{}^{mt} | \bar{\mathrm{x}}^{mt}; \theta)$$
$$\text{s.t. } (\bar{\mathrm{x}}^{mt}, \bar{a}_i{}^{mt}) \in \mathcal{S}$$

- Monotonically declines as more data is added
- Equals to the full log-likelihood at the leafs of RLS tree

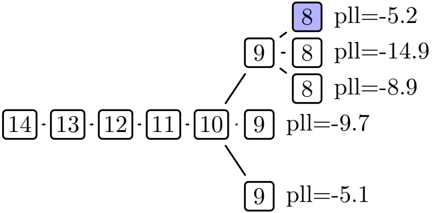
BnB on RLS tree, step 1

$\boxed{14} \cdot \boxed{13} \cdot \boxed{12} \cdot \boxed{11} \cdot \boxed{10}$ Partial loglikelihood = -3.2

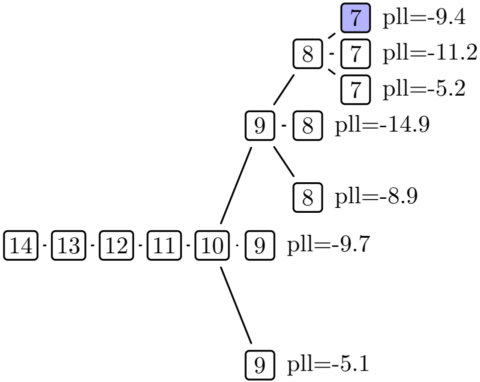
BnB on RLS tree, step 2



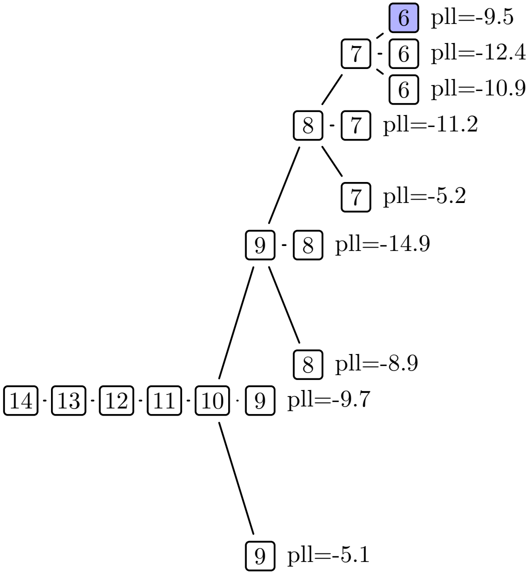
BnB on RLS tree, step 3



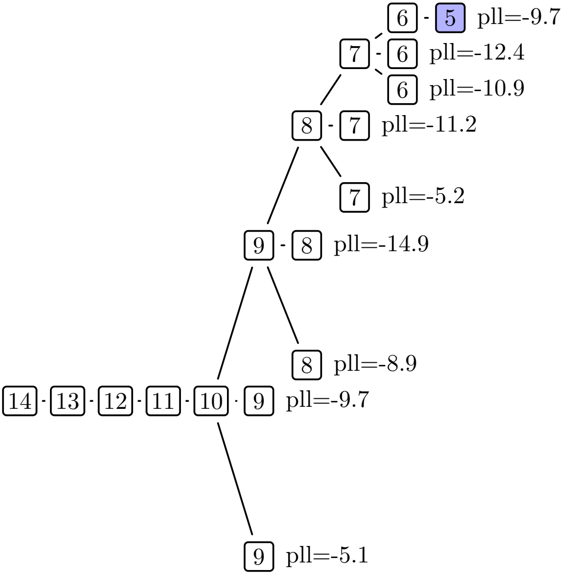
BnB on RLS tree, step 4



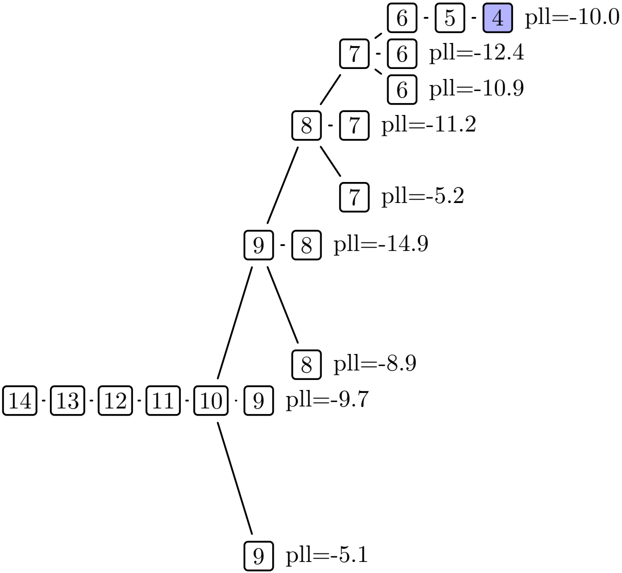
BnB on RLS tree, step 5



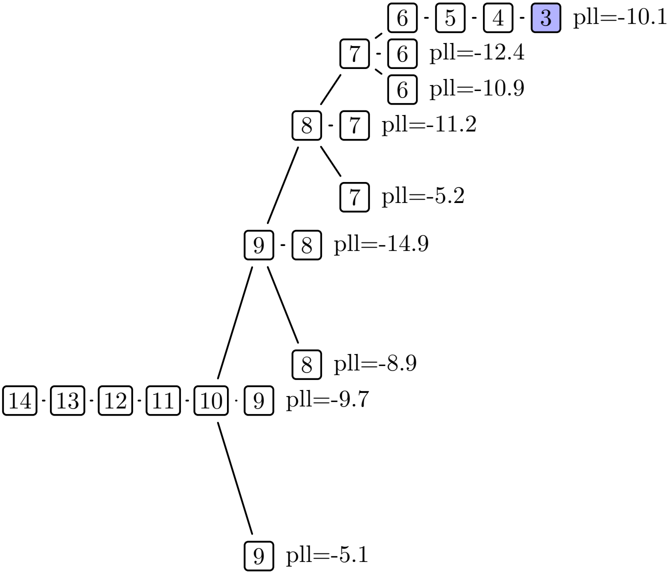
BnB on RLS tree, step 6



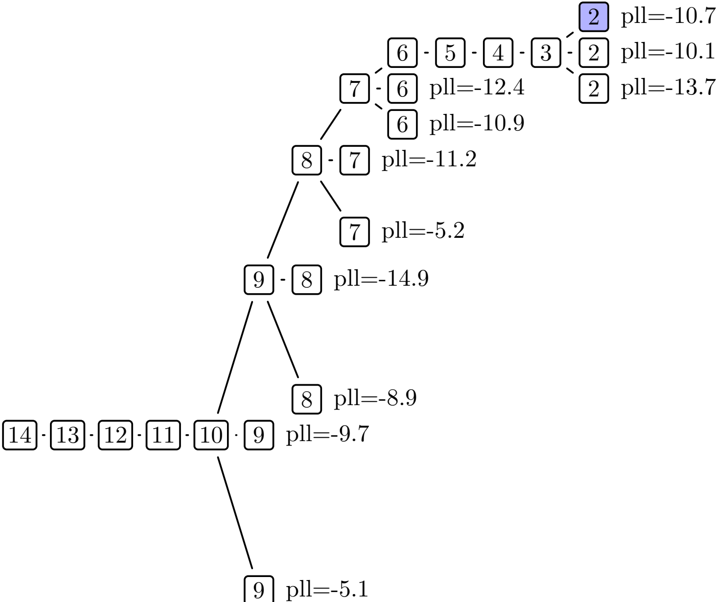
BnB on RLS tree, step 7



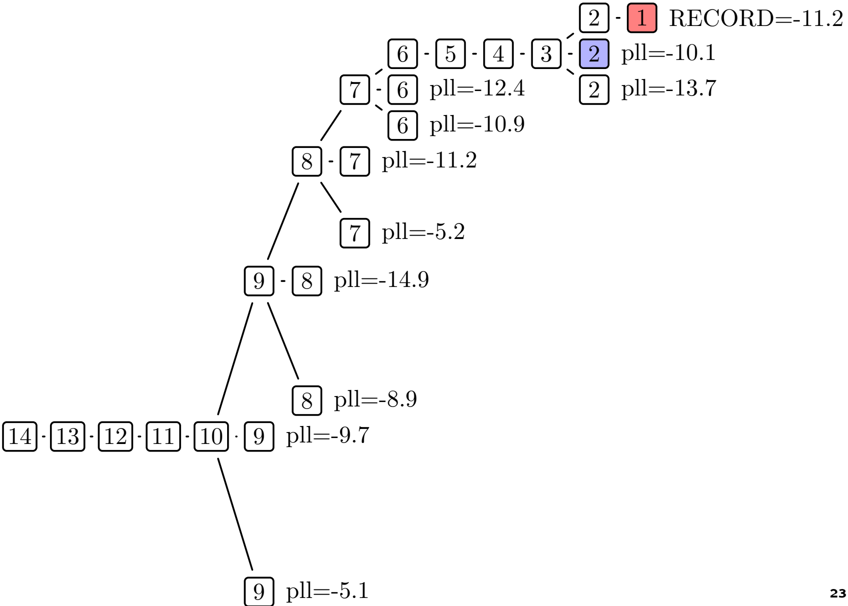
BnB on RLS tree, step 8



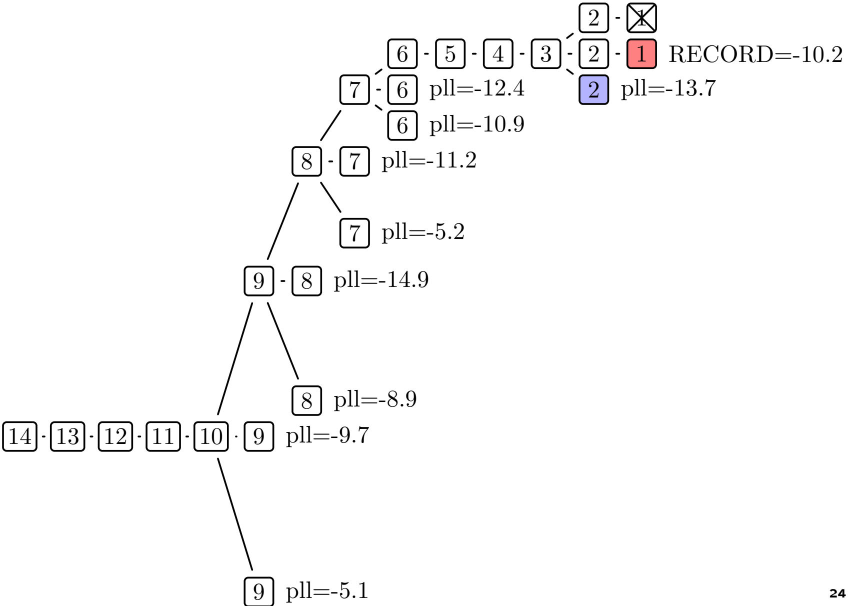
BnB on RLS tree, step 9



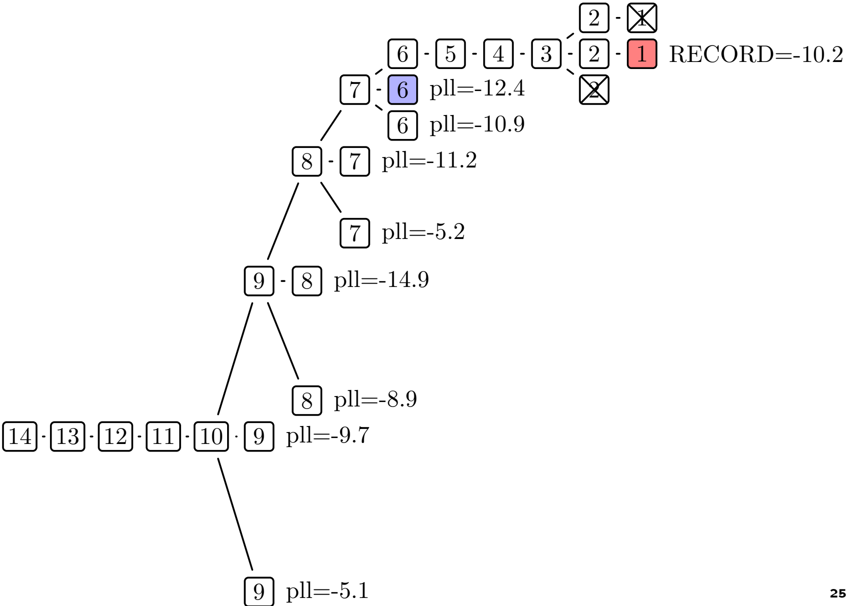
BnB on RLS tree, step 10



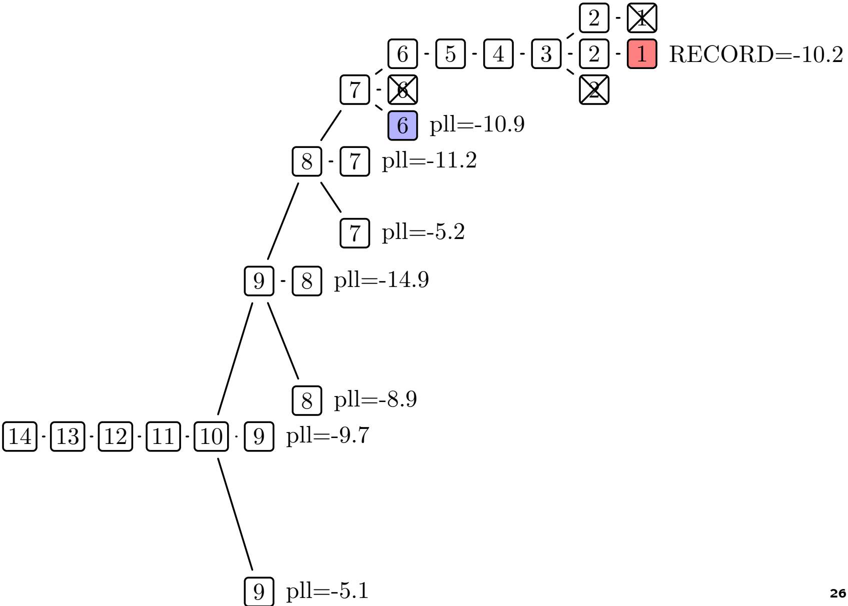
BnB on RLS tree, step 11



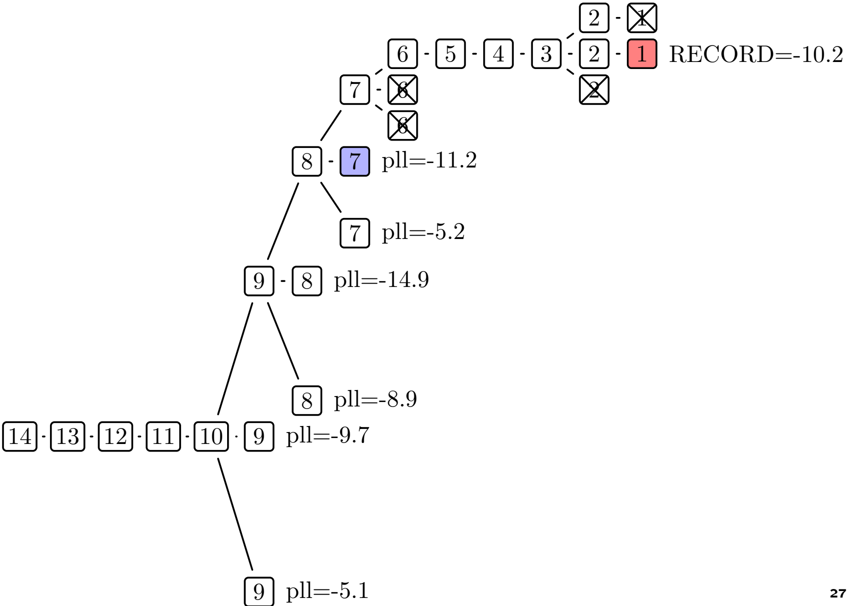
BnB on RLS tree, step 12



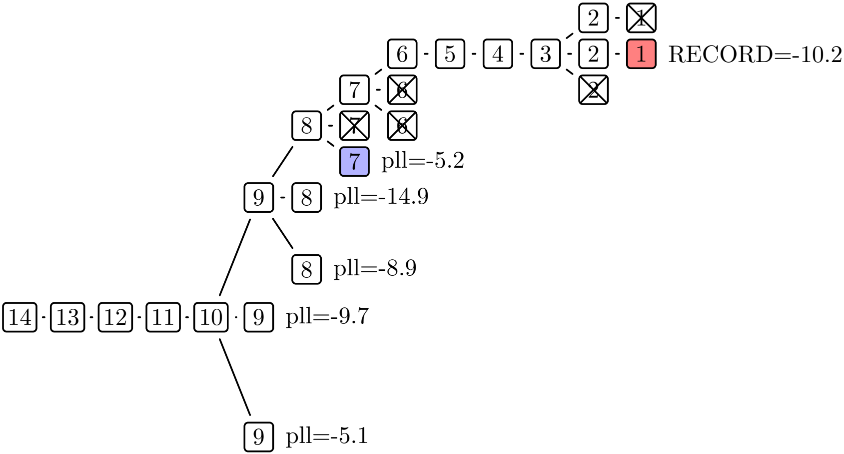
BnB on RLS tree, step 13



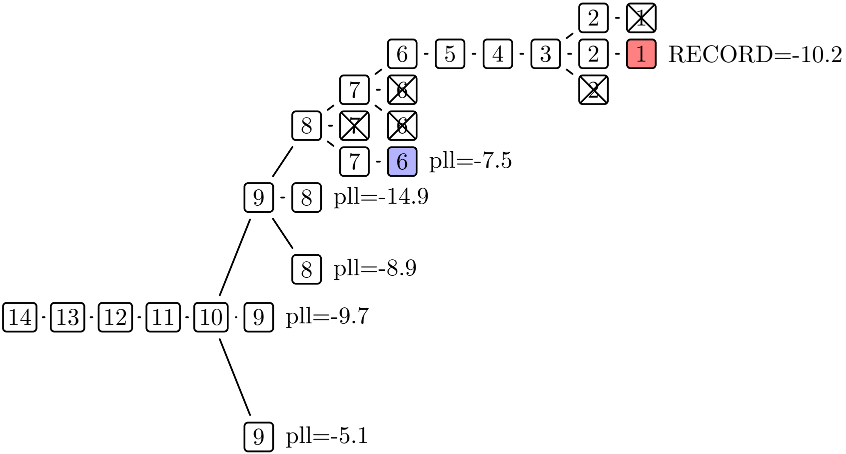
BnB on RLS tree, step 14



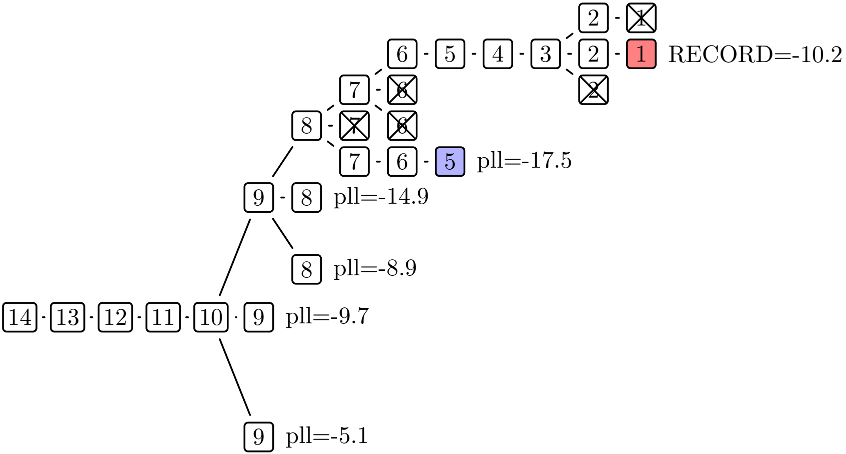
BnB on RLS tree, step 15



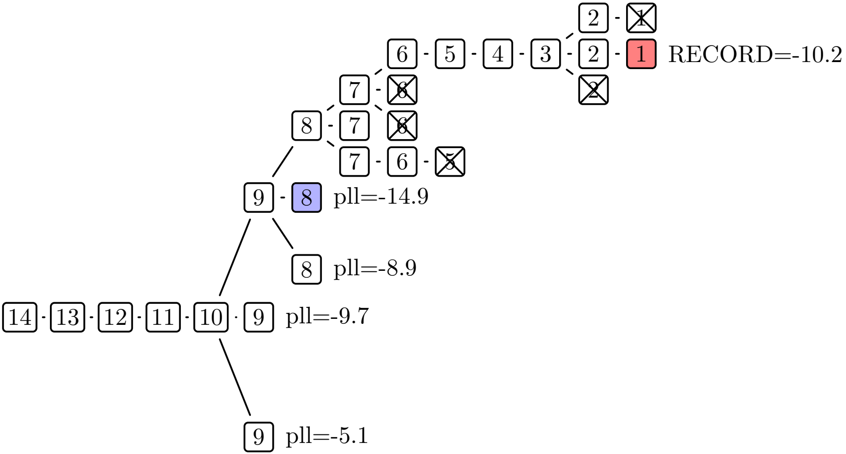
BnB on RLS tree, step 16



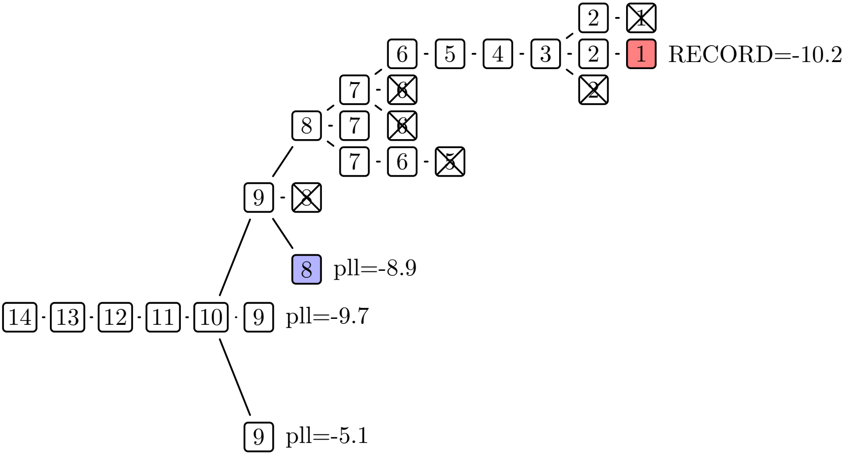
BnB on RLS tree, step 17



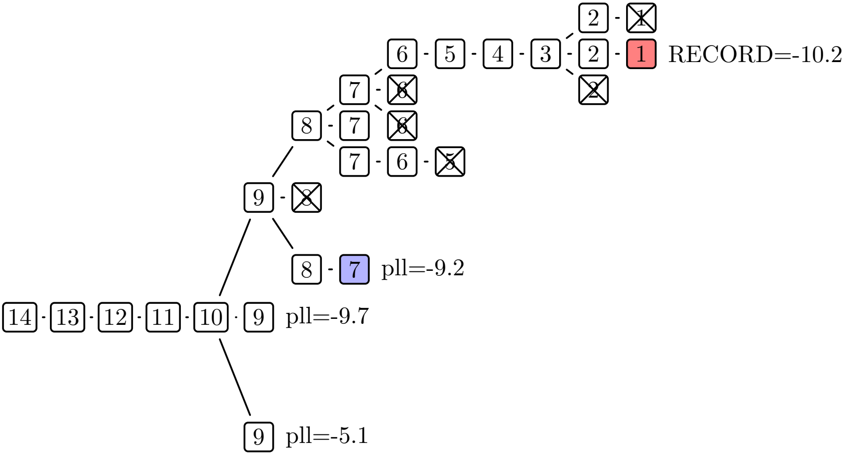
BnB on RLS tree, step 18



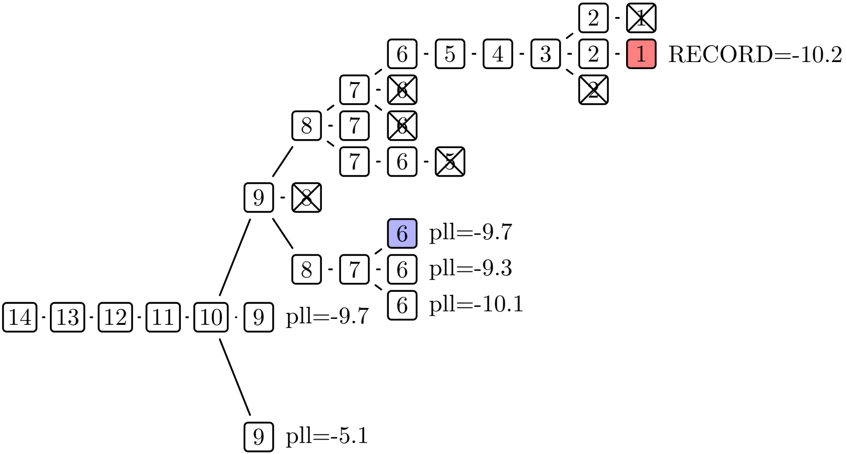
BnB on RLS tree, step 19



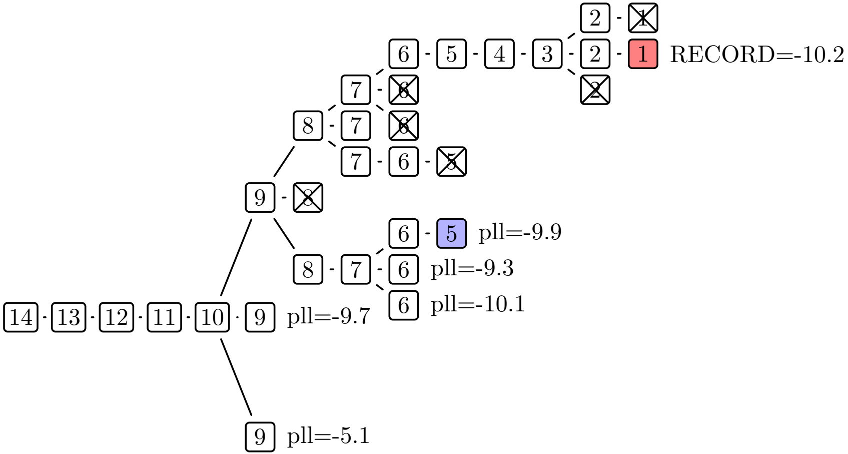
BnB on RLS tree, step 20



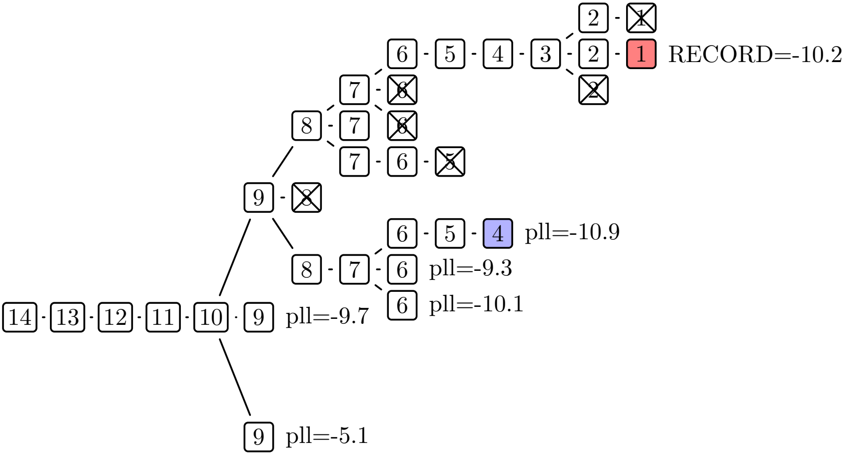
BnB on RLS tree, step 21



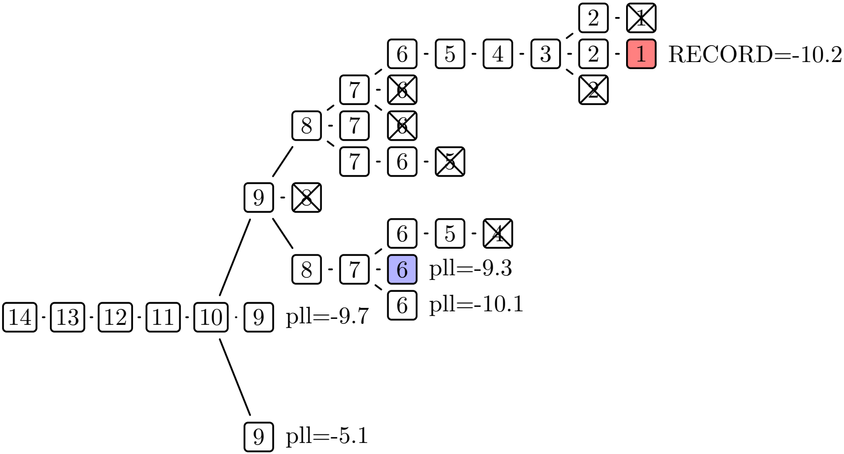
BnB on RLS tree, step 22



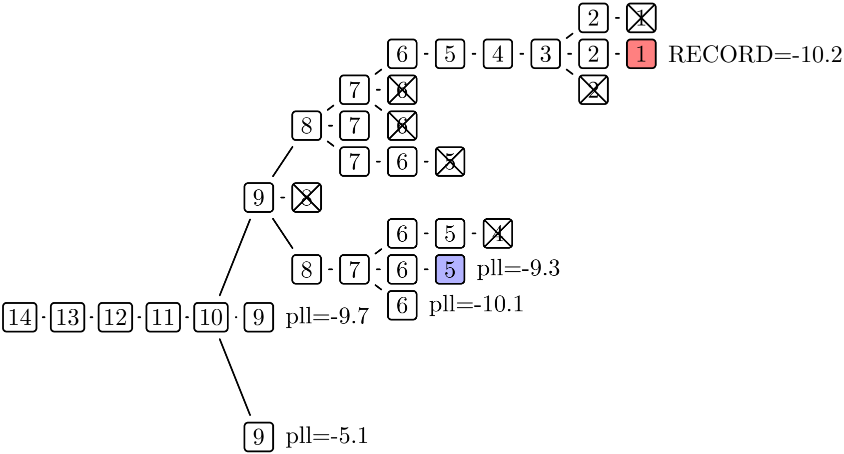
BnB on RLS tree, step 23



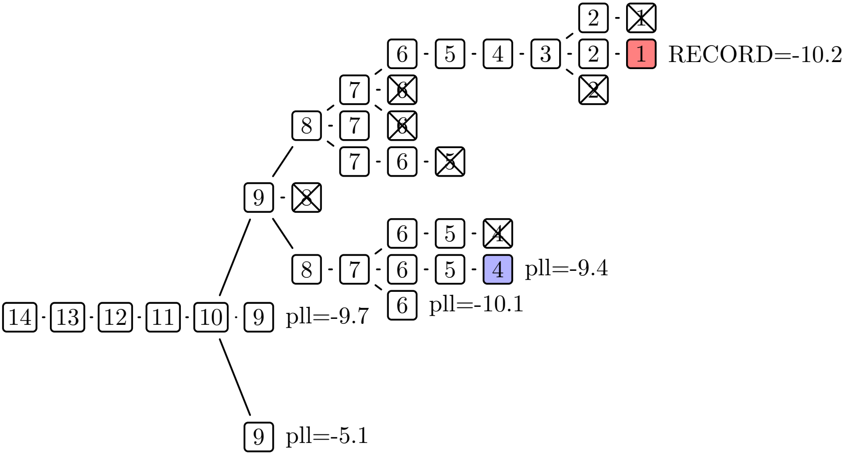
BnB on RLS tree, step 24



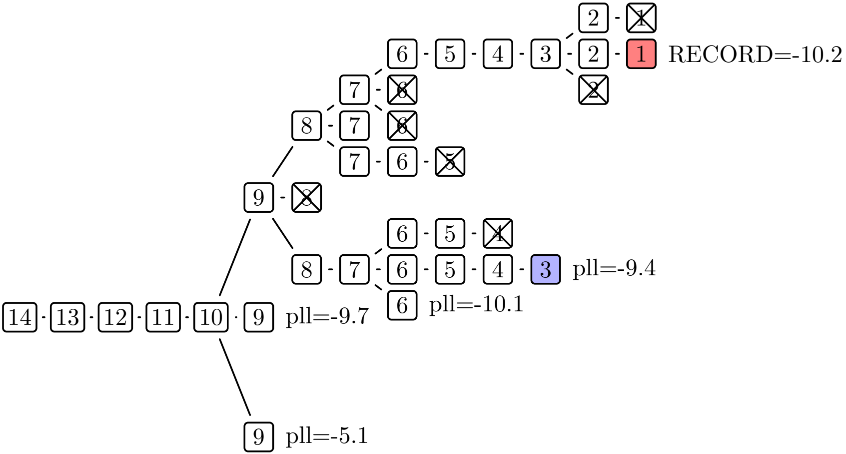
BnB on RLS tree, step 25



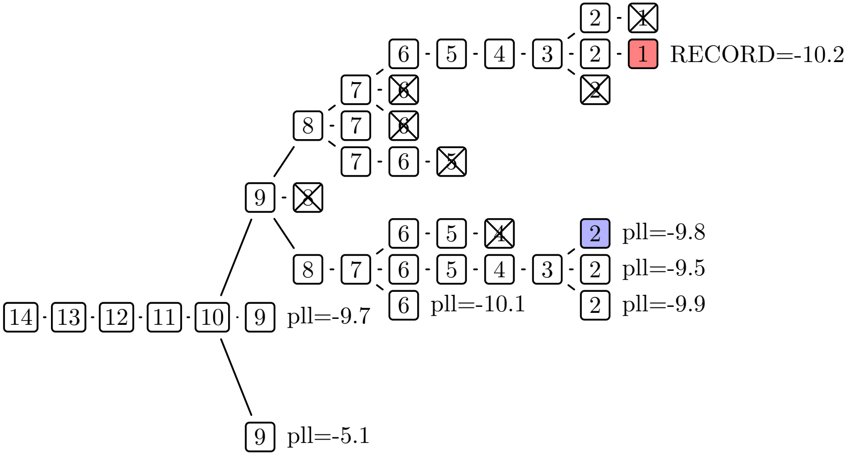
BnB on RLS tree, step 26



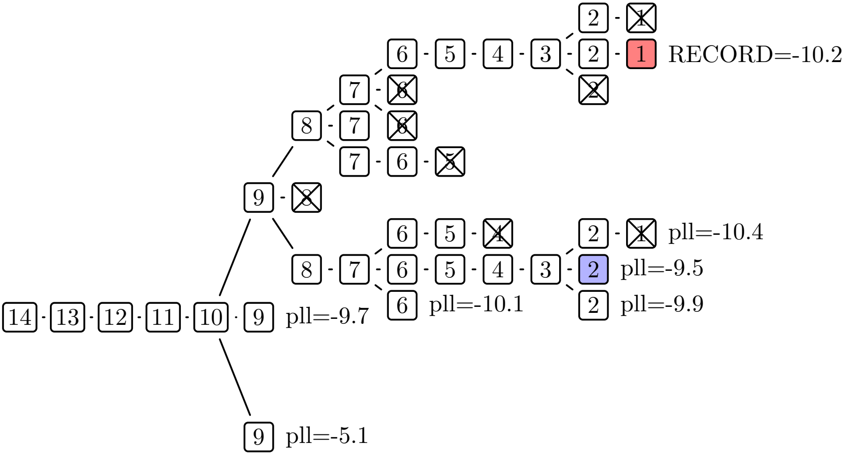
BnB on RLS tree, step 27



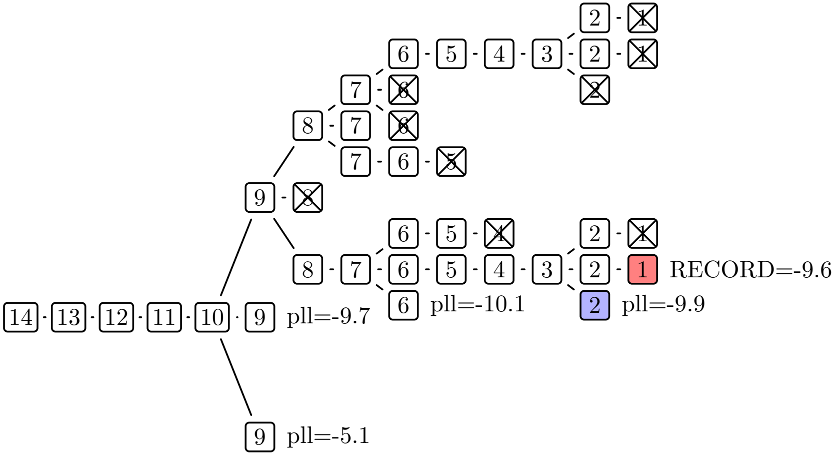
BnB on RLS tree, step 28



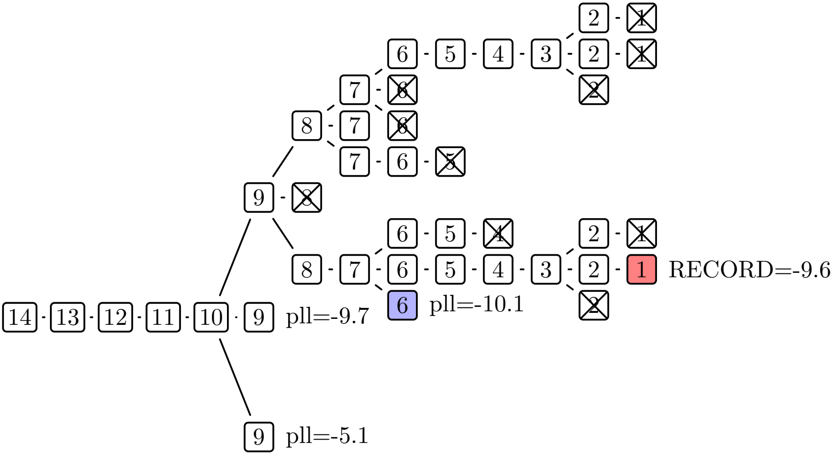
BnB on RLS tree, step 29



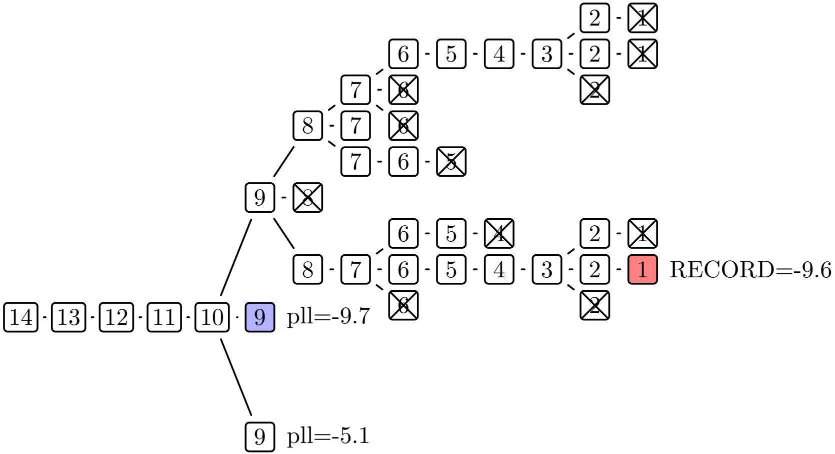
BnB on RLS tree, step 30



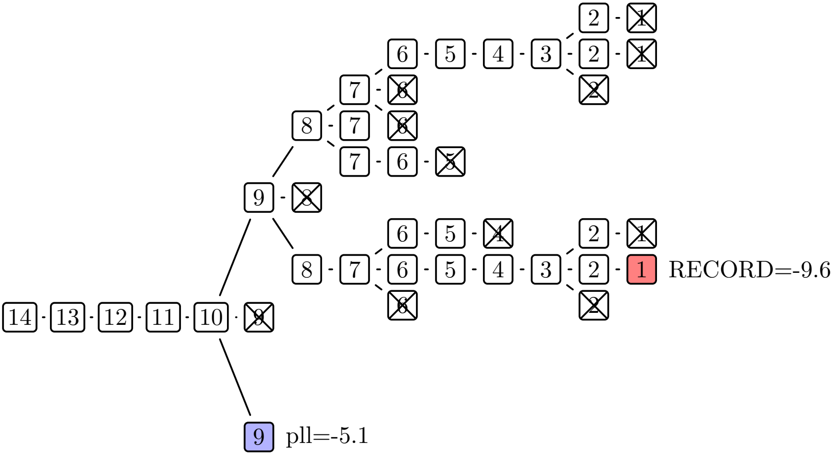
BnB on RLS tree, step 31



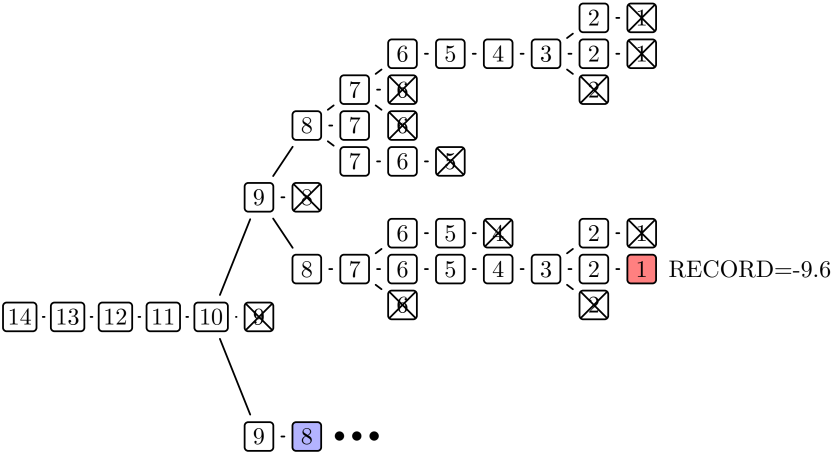
BnB on RLS tree, step 32



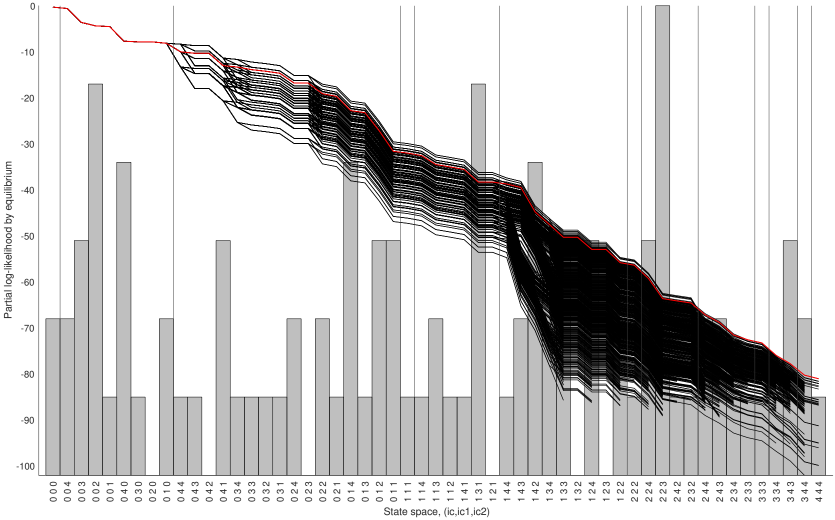
BnB on RLS tree, step 33



BnB on RLS tree, step 34



BnB and numerical performance of NRLS estimator



Numerical performance and refinements

- Numerical performance of NRLS estimator depends crucially on how the data is able to distinguish between different equilibria

- Bounding criterion is deterministic $\rightarrow$ may use statistical criterion to decide whether to extend a given branch or not
- Have to assess potential likelihood contribution of the branches that are not fully extended $\rightarrow$ Vuong closeness test (LR-type test to assess how different two equilibria are given already computed partial likelihood)

$\Rightarrow$ Poly-algorithm with statistical decision rule

Monte Carlo simulations

A

Single equilibrium in the model
Single equilibrium in the data

B

Multiple equilibria in the model
Single equilibrium in the data

C

Multiple equilibria in the model
Multiple equilibria in the data

1. Two-step CCP estimator
2. Nested pseudo-likelihood vs. NRLS estimator
3. MPEC

Implementation details

- Two-step estimator and NPL
  - Matlab unconstraint optimizer (numerical derivatives)
  - CCPs from frequency estimators
  - For NPL max 30 iterations
- MPEC
  - Matlab constraint optimizer (interior-point algorithm)
  - MPEC-VP: Constraints on both values and choice probabilities (as in Egesdal, Lai and Su, 2015)
  - MPEC-P: Constraints in terms of choice probabilities + Hotz-Miller inversion
  - Starting values from two-step estimator
- Estimated parameters $\theta = (k_1, k_2)$
- Sample size: 1000 markets in 5 time periods
- Initial state drawn uniformly over the state space

Monte Carlo A, run 1: no multiplicity

Maximum number of equilibria in the model: 1

Number of equilibria in the data: 1

	PML2step	NPL	MPEC-VP	MPEC-P	NRLS
k1=3.5	3.51893	3.51022	3.50380	3.50380	3.50380
Bias	0.01893	0.01022	0.00380	0.00380	0.00380
MCSD	0.12087	0.12635	0.11573	0.11573	0.11573
k2=0.5	0.50860	0.50658	0.50452	0.50452	0.50452
Bias	0.00860	0.00658	0.00452	0.00452	0.00452
MCSD	0.06460	0.06247	0.05939	0.05939	0.05939
log-likelihood	-1958.176	-1953.406	-1953.327	-1953.327	-1953.327
$\|\|\Psi^{\mathbf{P}}(P) - P\|\|$	0.25285	0.00001	0.00000	0.00000	0.00000
$\|\|\Psi^{\mathbf{V}}(v) - v\|\|$	0.50038	0.00001	0.00000	0.00000	0.00000
Converged,%	100	100	100	100	100
K-L divergence	0.131139	0.005020	0.006770	0.006770	0.006770

- All MLE estimators identical to the last digit
- NPL estimator is approaching MLE

Monte Carlo A, run 2: no multiplicity at true parameter

Maximum number of equilibria in the model: 3

Number of equilibria at true parameter value: 1

Number of equilibria in the data: 1

	PML2step	NPL	MPEC-VP	MPEC-P	NRLS
k1=3.5	3.50467	3.51307	3.49485	3.49318	3.49318
Bias	0.00467	0.01307	-0.00515	-0.00682	-0.00682
MCSD	0.11252	0.00000	0.10193	0.10177	0.10177
k2=0.5	0.50035	0.47394	0.50265	0.50157	0.50157
Bias	0.00035	-0.02606	0.00265	0.00157	0.00157
MCSD	0.05009	0.00000	0.04154	0.04205	0.04205
log-likelihood	-4106.771	-3940.158	-4091.873	-4093.040	-4093.04
$\|\|\Psi^{\mathbf{P}}(P) - P\|\|$	0.41453	0.00001	0.00000	0.00000	0.00000
$\|\|\Psi^{\mathbf{V}}(v) - v\|\|$	1.90182	0.00005	0.00000	0.00000	0.00000
Converged,%	100	1	98	100	100
K-L divergence	0.188551	0.004546	0.002921	0.002921	0.002920

- NPL estimator fails to converge
- MPEC is not affected by “nearby” equilibria with good starting values (PML2step)

Monte Carlo B, run 1: moderate multiplicity

Number of equilibria in the model (at true parameter): 3

Number of equilibria in the data: 1

	PML2step	NPL	MPEC-VP	MPEC-P	NRLS
k1=3.5	3.50081	-	3.72713	3.94941	3.49624
Bias	0.00081	-	0.22713	0.44941	-0.00376
MCSD	0.12050	-	0.85934	1.16633	0.09537
k2=0.5	0.49478	-	0.56166	0.62361	0.49381
Bias	-0.00522	-	0.06166	0.12361	-0.00619
MCSD	0.04317	-	0.25552	0.32488	0.03510
log-likelihood	-4070.035	-	-4080.989	-4121.102	-4049.647
$\|\|\Psi^{\mathbf{P}}(P) - P\|\|$	0.50375	-	0.00000	0.00000	0.00000
$\|\|\Psi^{\mathbf{V}}(v) - v\|\|$	2.83611	-	0.00000	0.00000	0.00000
Converged,%	100	0	100	100	100
K-L divergence	0.304411	-	0.018636	2.302525	0.006314

- NPL estimator fails to converge
- MPEC fails to identify the equilibrium that generated the data (converges to a different MPE) as seen from MCSD and K-L divergence

Monte Carlo B, run 2: higher multiplicity

Number of equilibria in the model (at true parameter): 81

Number of equilibria in the data: 1

	PML2step	NPL	MPEC-VP	MPEC-P	NRLS
k1=3.5	3.51468	-	3.48740	3.49007	3.47786
Bias	0.01468	-	-0.01260	-0.00993	-0.02214
MCSD	0.04844	-	0.02802	0.02929	0.02731
k2=0.5	0.53780	-	0.49197	0.48944	0.49252
Bias	0.03780	-	-0.00803	-0.01056	-0.00748
MCSD	0.03894	-	0.00850	0.01033	0.00404
log-likelihood	-4038.78471	-	-4007.45663	-4010.18139	-3996.45223
$\|\|\Psi^{\mathbf{P}}(P) - P\|\|$	0.68907	-	0.00000	0.00000	0.00000
$\|\|\Psi^{\mathbf{V}}(v) - v\|\|$	5.44052	-	0.00000	0.00000	0.00000
Converged,%	100	0	100	100	100
K-L divergence	0.453917	-	0.278263	0.356678	0.000750

- NPL estimator fails to converge
- MPEC fails to identify the DGP equilibrium (converges to a different MPE)
- With good starting values, does not suffer more with higher multiplicity

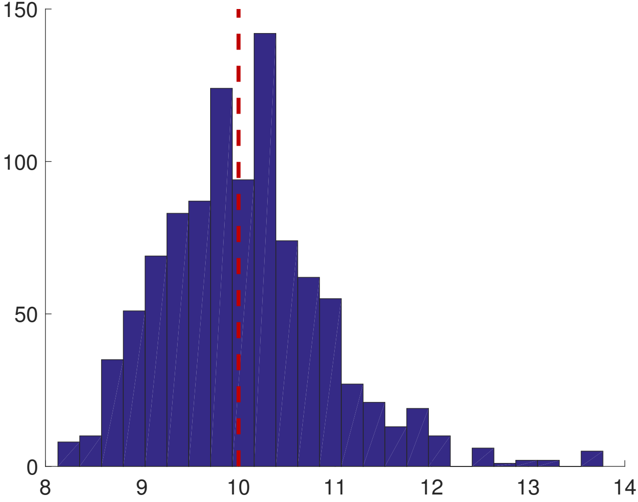
NRLS Monte Carlo setup (C)

- $n = 3$ points on the grid on the grid of costs
- 14 points in state space of the model
- 109 MPE in total

- 1000 random samples from 3 different equilibria (3 markets)
- 100 observations per market/equilibrium
- Uniform distribution over state space $\leftrightarrow$ "ideal" data

- Estimating one parameter in cost function

Distribution of estimated k_1 parameter



MC results and numerical performance of NRLS

1. Average bias and RMSE of the estimates of the cost of investment parameter (true value is 10.0)

$$\text{Bias} = 0.0737$$
$$\text{RMSE} = 0.8712$$

2. Average fraction of MPE computed by BnB relative to RLS

   0.321 (std=0.11635)

3. Average fraction of stages solved by BnB relative to RLS

   0.263 (std=0.09725)

4. All 3 MPE correctly identified by BnB in

   98.4% of runs

Identification of multiple equilibria in the data (C)

- 100 random samples
- 3 market clusters with different equilibria
- 1000 observations per market cluster/equilibrium in 3 time periods

- Among all runs, 93% of equilibria were pin-pointed exactly
- Among the misidentified equilibria, all had deviation in one point of the state space

Conclusions: Bertrand investments model

- Many types of endogenous coordination is possible in equilibrium
  - Leapfrogging (alternating investments)
  - Preemption (investment by cost leader)
  - Duplicative (simultaneous investments)
- Full rent dissipation and monopoly outcomes are supported as MPE.
- Numerous MPE equilibria and "Folk theorem"-like result
- The equilibria are generally inefficient due to over-investment
  - Duplicative or excessively frequent investments

Conclusions: Solution of dynamic games

- When equilibrium is not unique the computation algorithm inadvertently acts as an equilibrium selection mechanism
- When directionality in the state space is present, state recursion algorithm is preferred to time iterations
- Plethora of Markov perfect equilibria poses new challenges:
  - How firms manage to coordinate on a particular equilibrium?
  - Increased difficulties for empirical applications.
  - Daunting perspectives for identification of equilibrium selection rule from the data.
- Estimation of dynamic games with multiple equilibria
  Nested Recursive Lexicographical Search (NRLS)

Contributions and further developments

- NRLS is MLE estimator for dynamic games of a particular type, directional dynamic games (DDGs)
  - Fully robust to multiplicity of equilibria
  - Able to identify multiple equilibria in the data
- Further work on and tests of numerical performance
  - Refinements of the implementation of NRLS (optimization of BnB algorithm)
  - Statistical bounding criterion
- More detailed comparison of existing estimators using leapfrogging game
  - Refine the implementation of MPEC
  - Include recent estimators into the battery (Aguirregabiria and Marcoux, 2019, Bugni and Bunting, 2020)