Lecture 15: Nested Recursive Lexicographical Search: Structural Estimation of Dynamic Directional Games with Multiple Equilibria

Fedor Iskhakov, Australian National University
Anders Munk-Nielsen, University of Copenhagen
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

Empirical IO: Dynamic Structural Models
Department of Economics (NHH)
29 Nov - 3 Dec., 2021

Estimation of stochastic dynamic games

1. Several decision makers (*players*)
2. Maximize discounted expected lifetime utility
3. Anticipate consequences of their current actions
4. Anticipate actions by other players in current and future periods (*strategic interaction*)
5. Operate in a stochastic environment (*state of the game*) whose evolution depend on the collective actions of the players

- Estimate structural parameters of these models
- Data on M independent markets over T periods
- Multiplicity of equilibria

Markov Perfect Equilibrium

- MPE is a pair of strategy profile and value functions:
- **Bellman Optimality**
  Each players solves their Bellman equiation for values V taking other players choice probabilities P into account
- **Bayes-Nash Equilibrium**
  The choice probabilities P are determined by the values V
- 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

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

Maximum Likelihood

- 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.
- For a given θ, let
  $\left(\mathrm{P}^{\ell}(\theta), \mathrm{V}^{\ell}(\theta)\right) \in SOL(\Psi, \theta)$ denote the ℓ-the equilibrium
- Log-likelihood function is

$$\mathcal{L}(Z, \theta) = \max_{(\mathrm{P}^{\ell}(\theta), \mathrm{V}^{\ell}(\theta) \in SOL(\Psi, \theta)} \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P_i^{\ell}(\bar{a}_i{}^{mt} | \bar{\mathrm{x}}^{mt}; \theta)$$

- The ML estimator is $\theta^{ML} = \arg\max_{\theta} \mathcal{L}(Z, \theta)$

Estimation methods for stochastic games

Maximum likelihood estimator

- Efficient, but expensive: need full solution method
- No problem with multiple equilibria

Borkovsky, Doraszelsky and Kryukov (2010) All solution homotopy; Iskhakov, Rust and Schjerning (2016) RLS

Two-step estimators

- Fast, but potentially large finite sample biases

Bajari, Benkard, Levin (2007); Pakes, Ostrovsky, and Berry (2007); Pesendorfer and Schmidt-Dengler (2008)

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

Estimation methods for stochastic games

Nested psuedo-likelihood (recursive two-step)

- Bridges the gap between efficiency and tractability
- Unstable under multiplicity

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

Math Programming with Equilibrium Constraints (MPEC)

- Reformulates ML problem as constrained optimization
- Should not be affected by multiplicity

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

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

Summary of this paper

- Propose robust and computationally feasible MLE estimator for directional dynamic games (DDG), finite state stochastic games with particular transition structure
- 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
- Provide Monte Carlo evidence of the performance

- Fully robust to multiplicity of MPE
- Relax single-equilibrium-in-data assumption
- Path to estimation of equilibrium selection rules

Nested Recursive Lexicographical Search (NRLS)

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, $\mathcal{P}_1(\Omega) = \Omega$

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

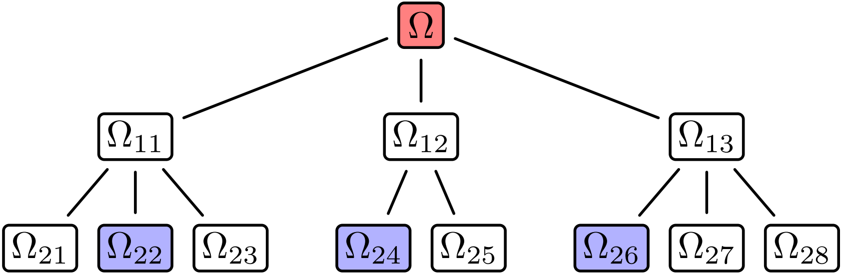
$\{\mathcal{P}_j(\Omega)\}_{j=1,\dots,J}$ a sequence of J gradually refined partitions

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

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

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

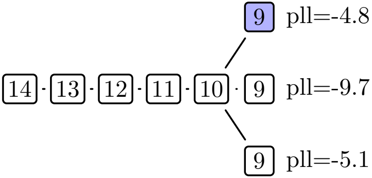
$$\mathcal{L}^{\text{Part}}(Z,\theta,\mathcal{S}) \quad = \quad \frac{1}{M}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T}\log P_i^{\ell}(\bar{a}_i{}^{mt}|\bar{\mathrm{x}}^{mt};\theta)$$
$$\text{s.t. } (\bar{\mathrm{x}}^{mt},\bar{a}_i{}^{mt}) \in \mathcal{S}$$

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

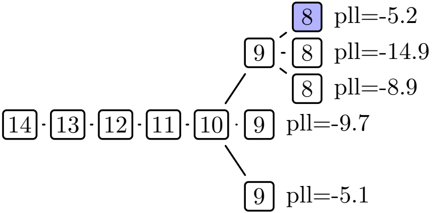
BnB on RLS tree, step 1

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

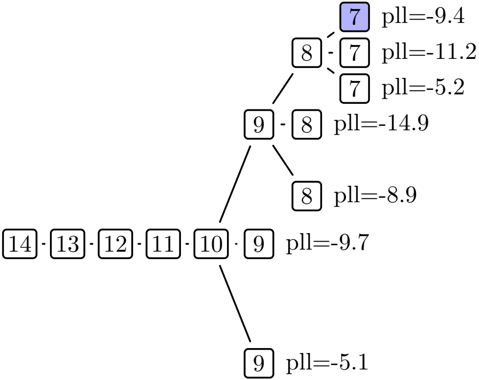
BnB on RLS tree, step 2



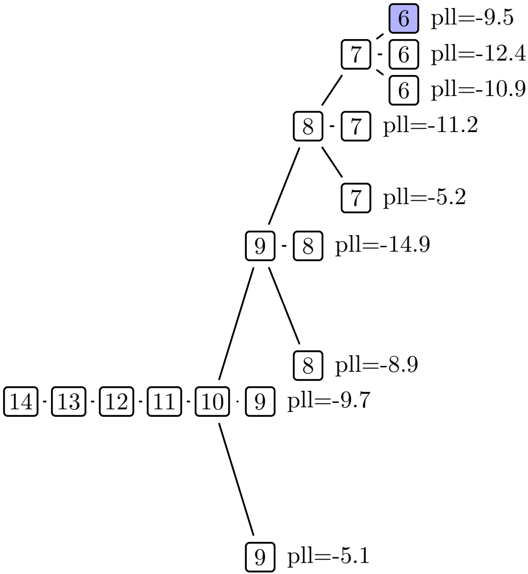
BnB on RLS tree, step 3



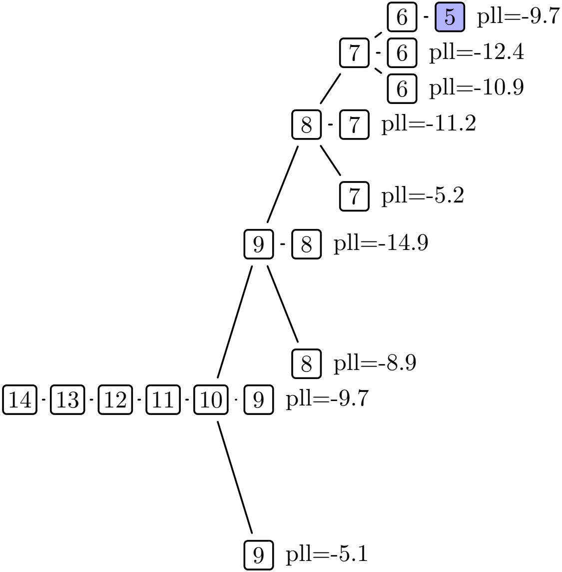
BnB on RLS tree, step 4



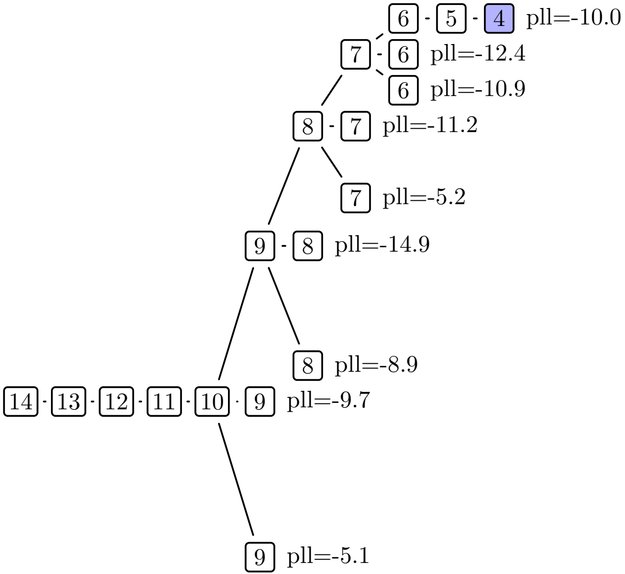
BnB on RLS tree, step 5



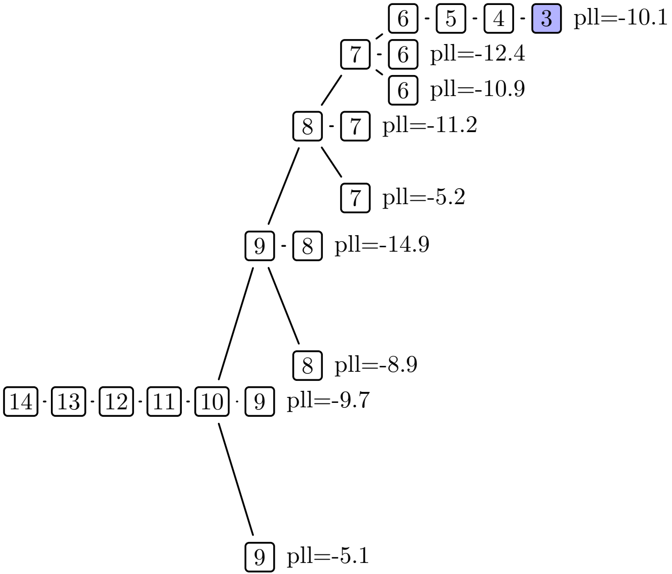
BnB on RLS tree, step 6



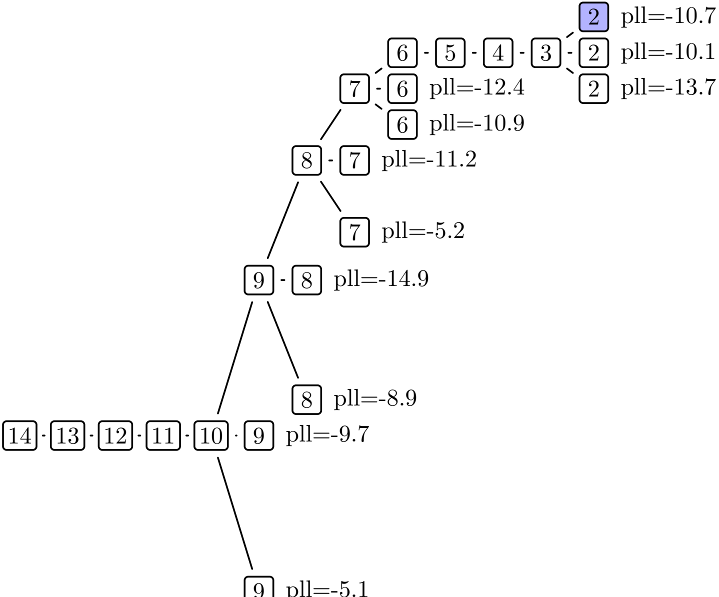
BnB on RLS tree, step 7



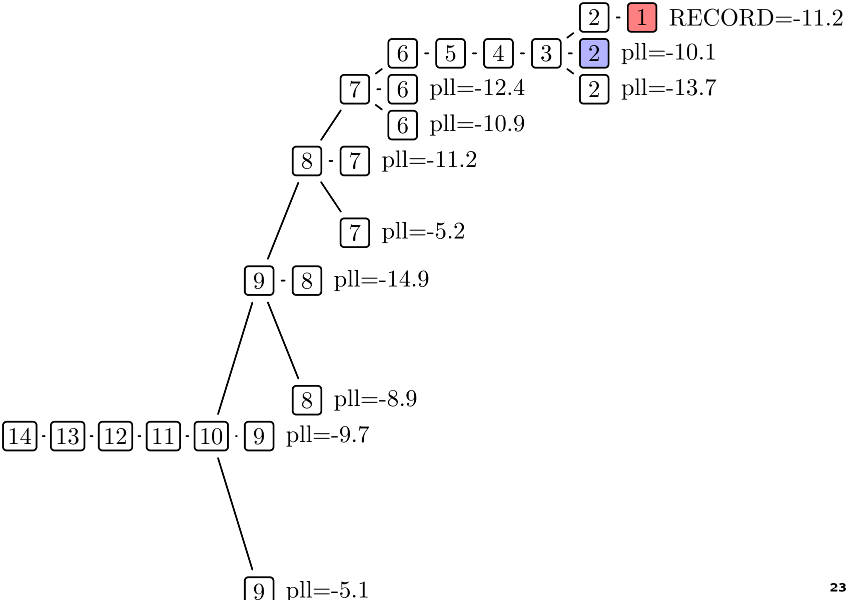
BnB on RLS tree, step 8



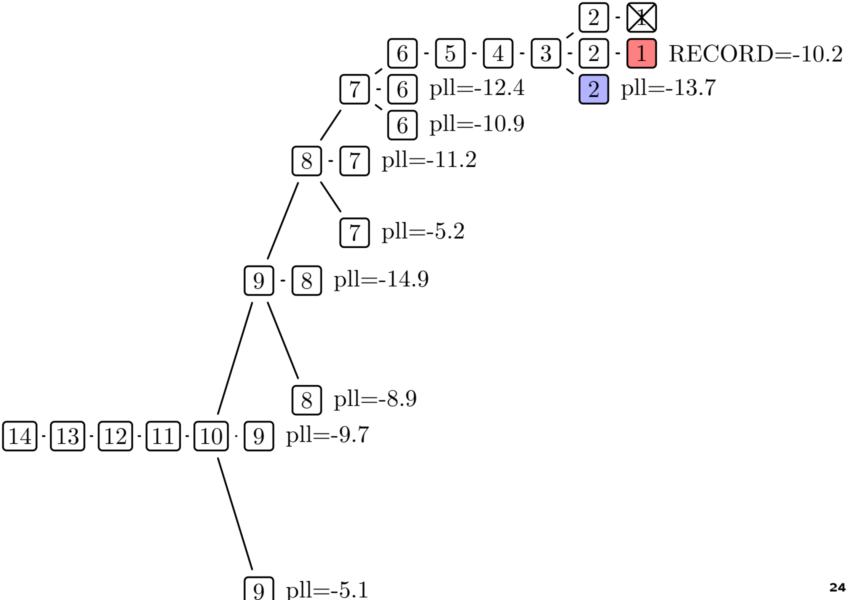
BnB on RLS tree, step 9



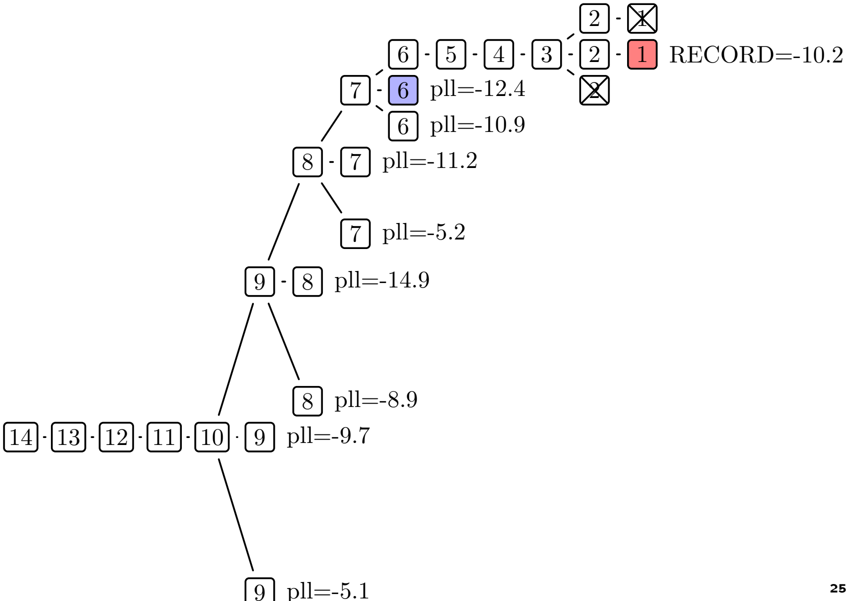
BnB on RLS tree, step 10



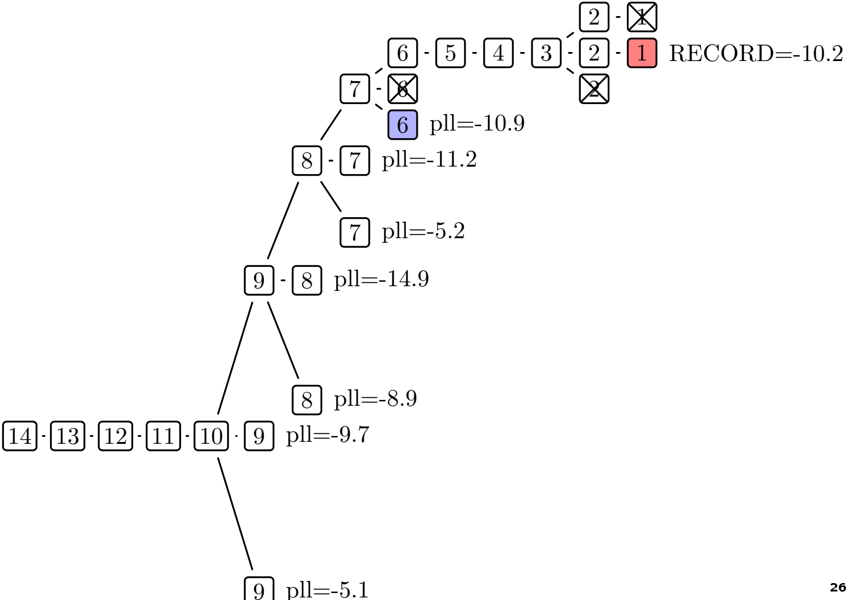
BnB on RLS tree, step 11



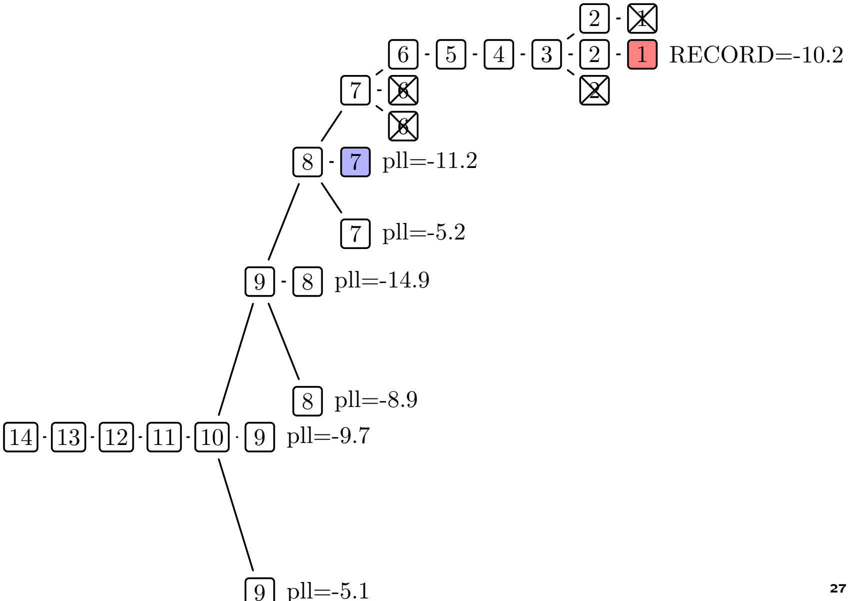
BnB on RLS tree, step 12



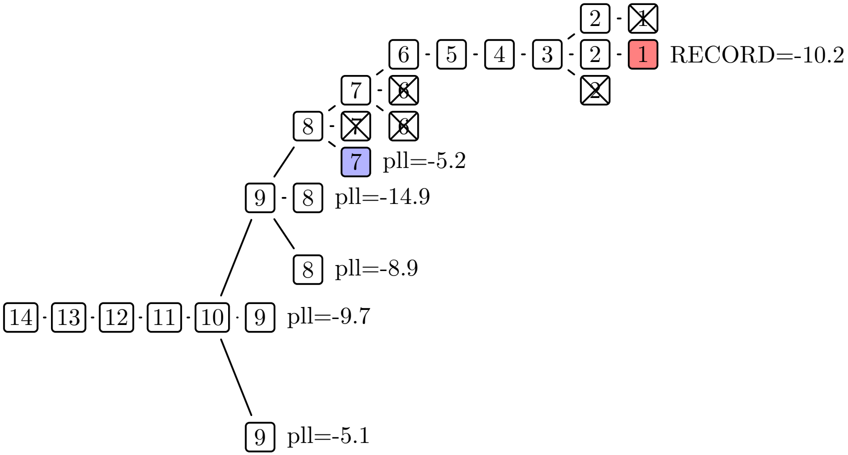
BnB on RLS tree, step 13



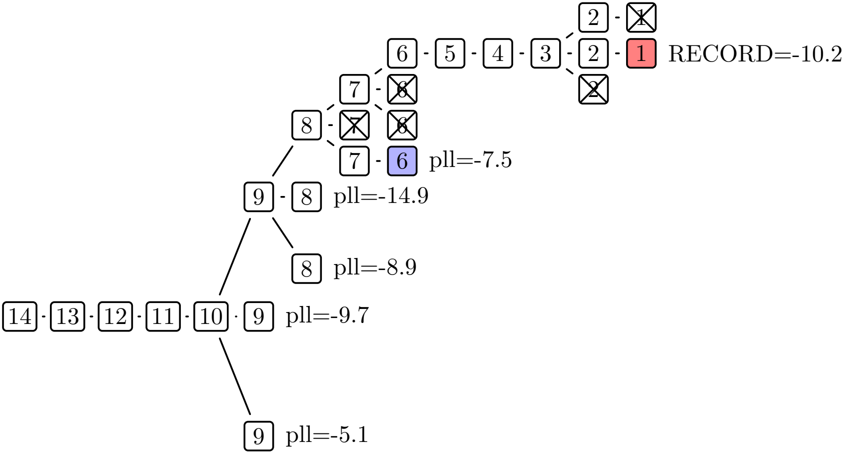
BnB on RLS tree, step 14



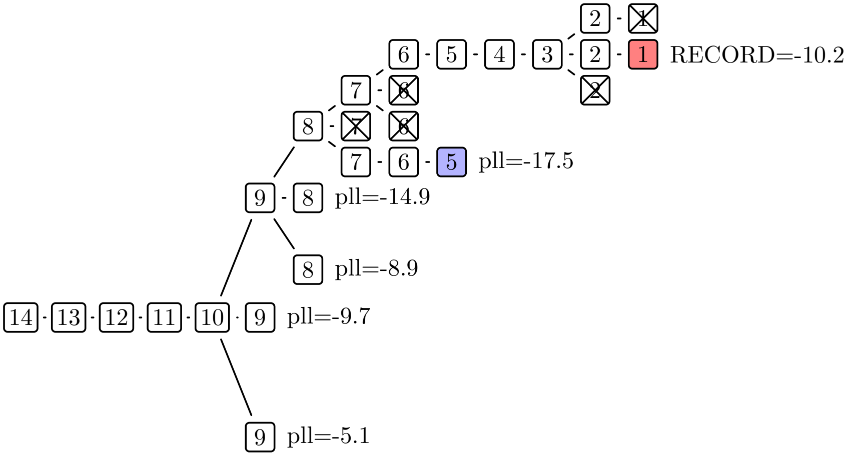
BnB on RLS tree, step 15



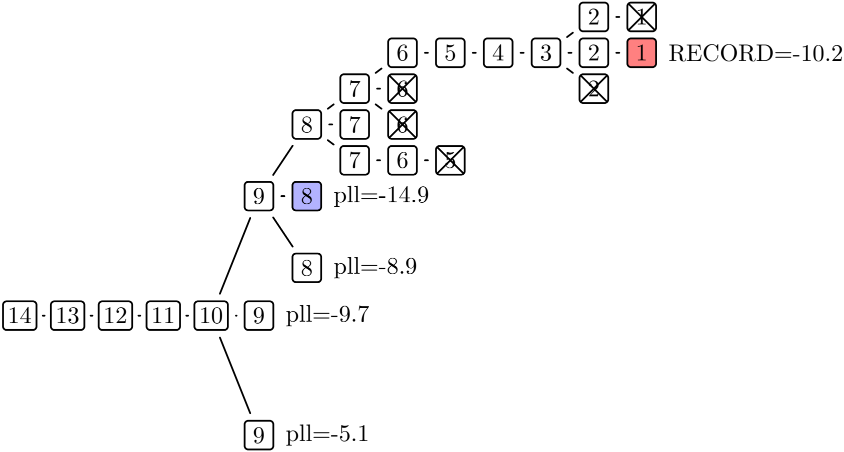
BnB on RLS tree, step 16



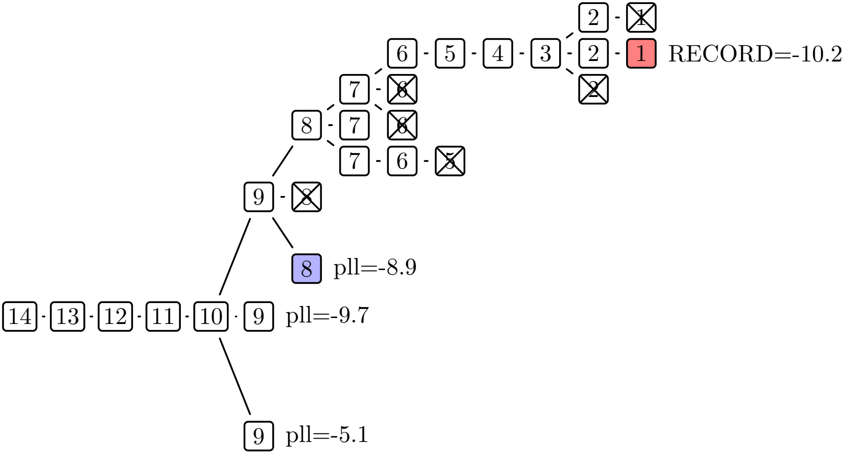
BnB on RLS tree, step 17



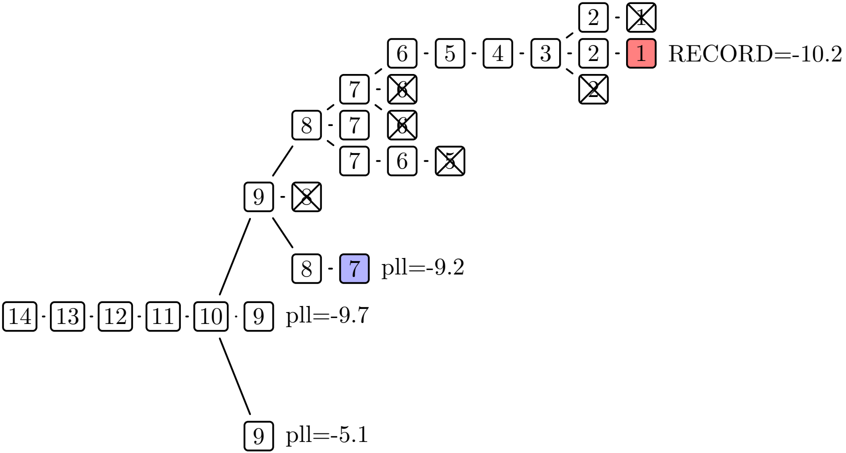
BnB on RLS tree, step 18



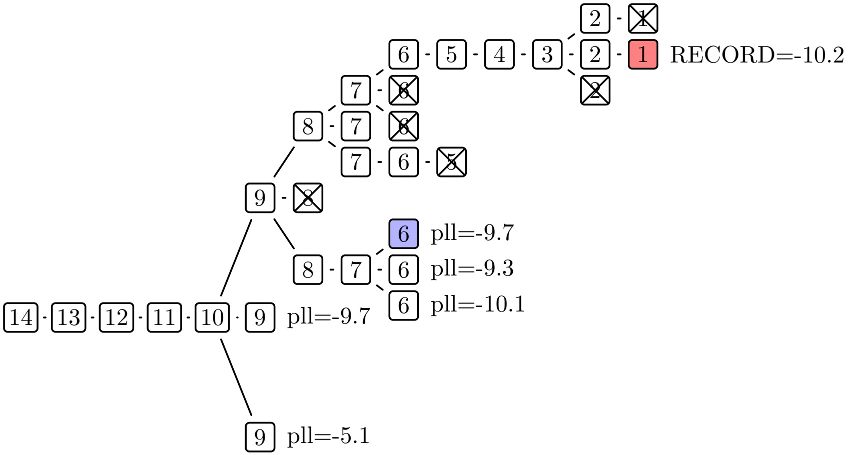
BnB on RLS tree, step 19



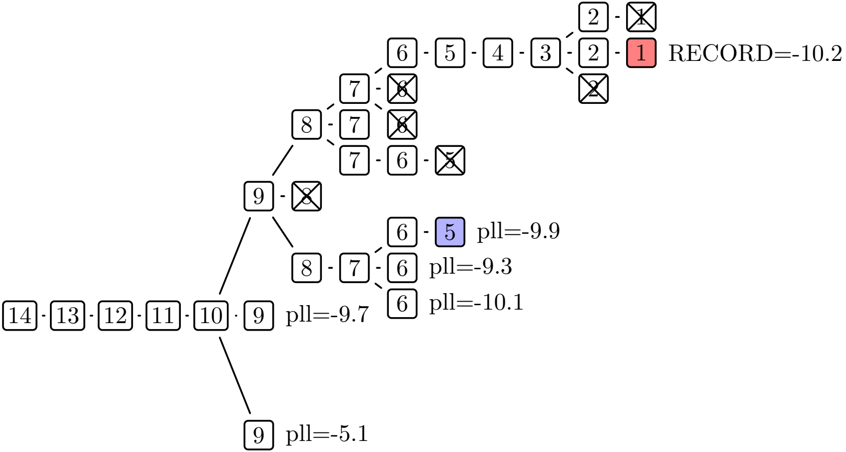
BnB on RLS tree, step 20



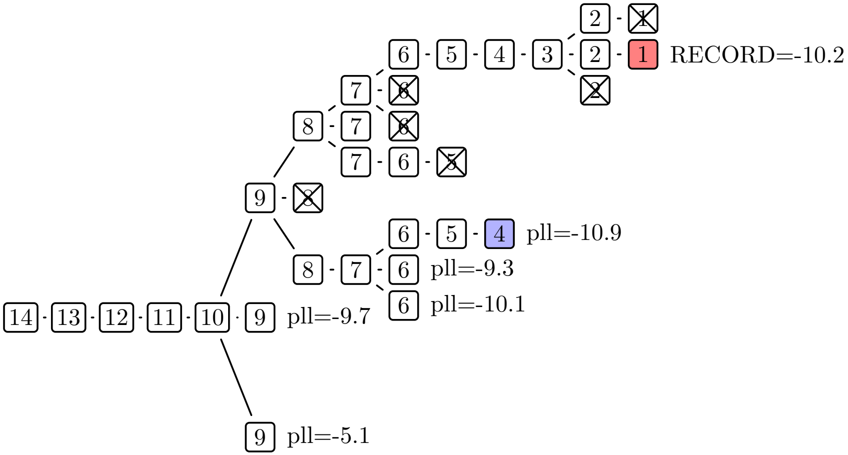
BnB on RLS tree, step 21



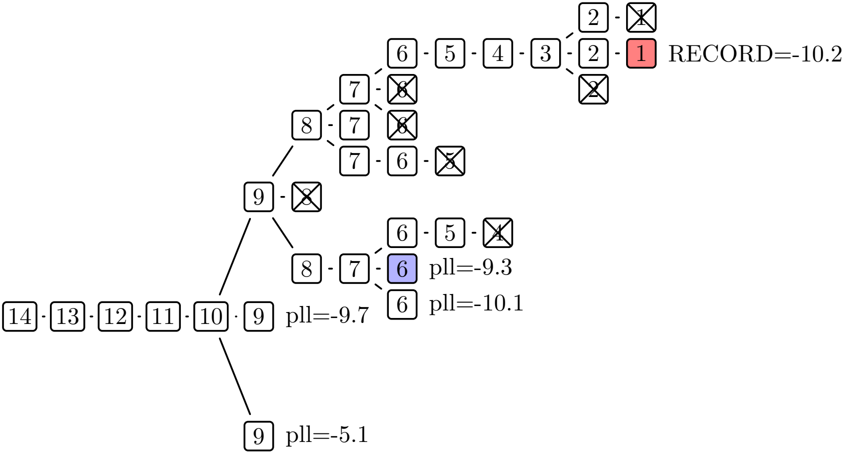
BnB on RLS tree, step 22



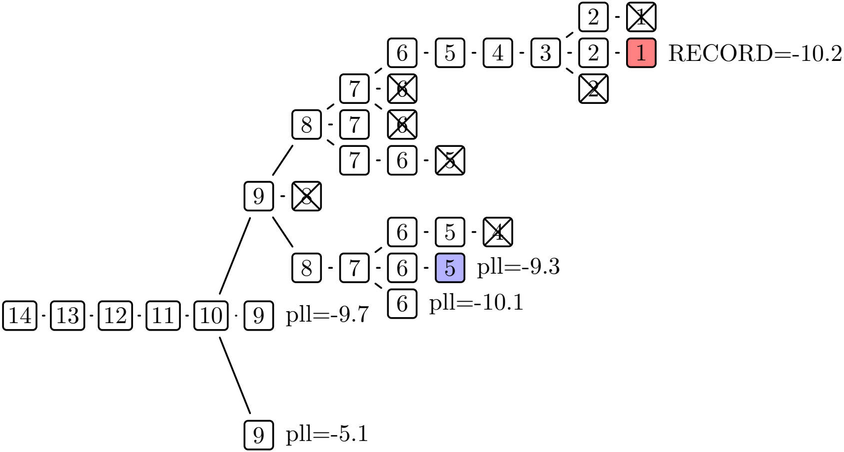
BnB on RLS tree, step 23



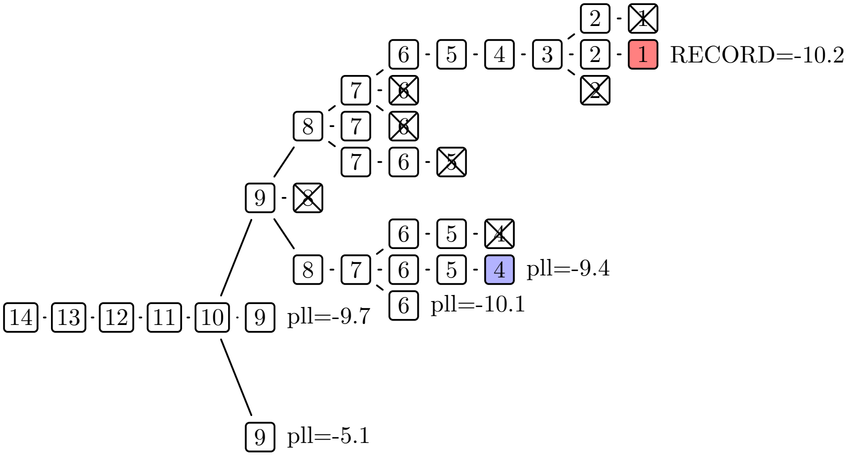
BnB on RLS tree, step 24



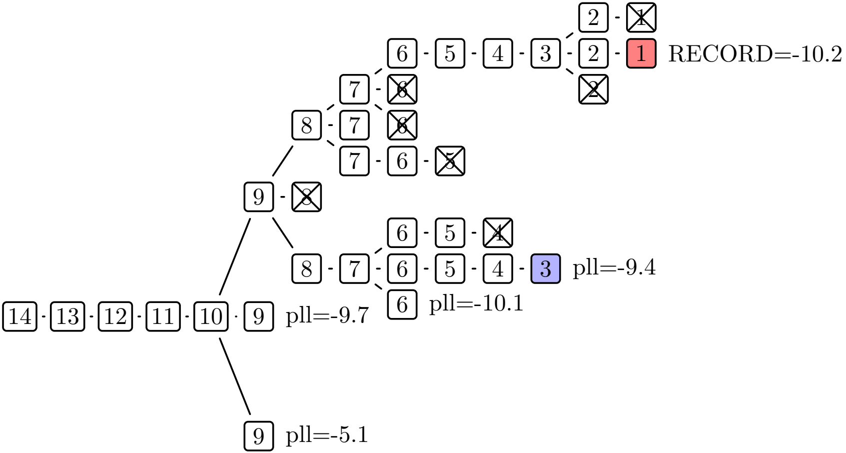
BnB on RLS tree, step 25



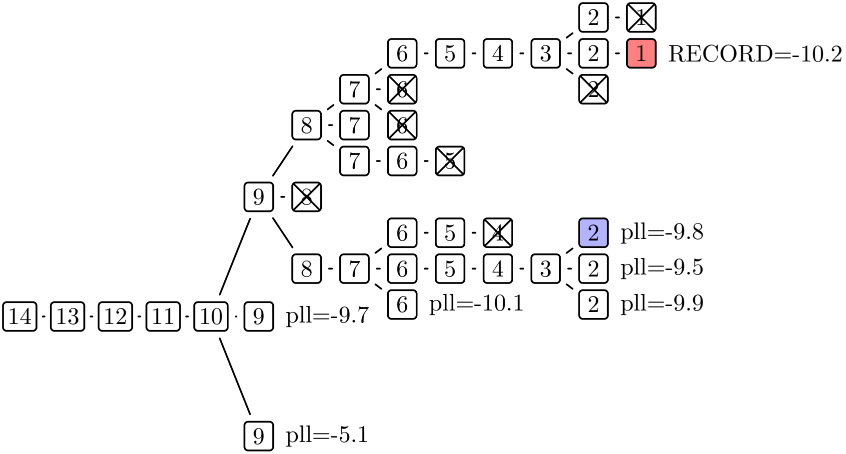
BnB on RLS tree, step 26



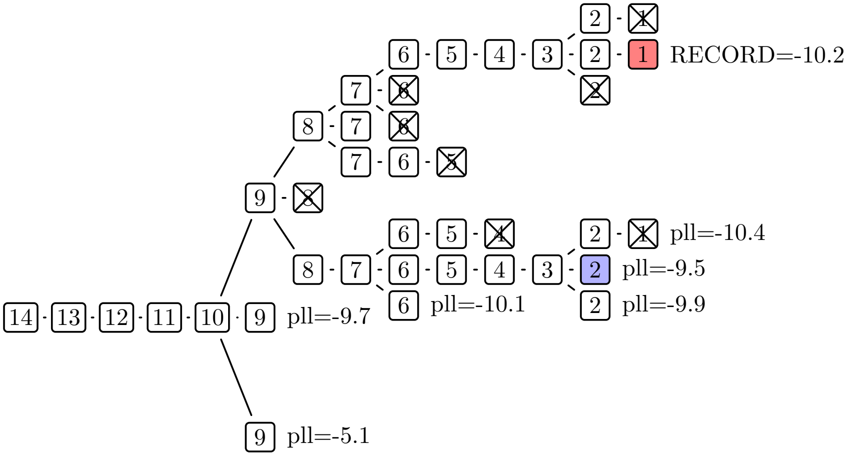
BnB on RLS tree, step 27



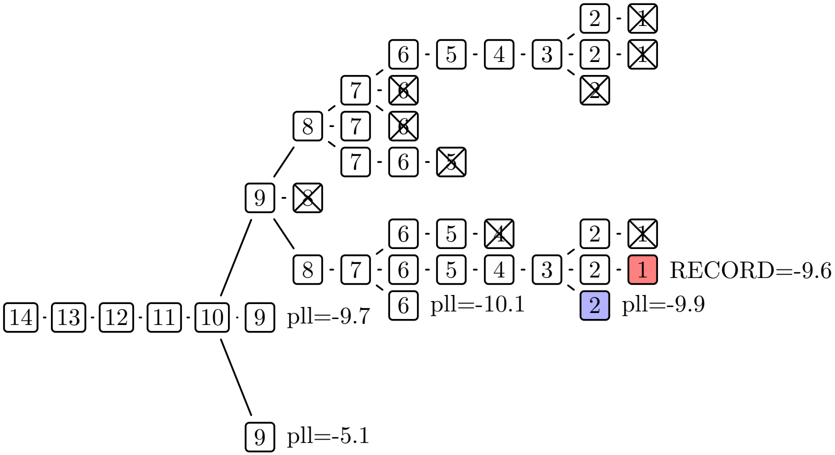
BnB on RLS tree, step 28



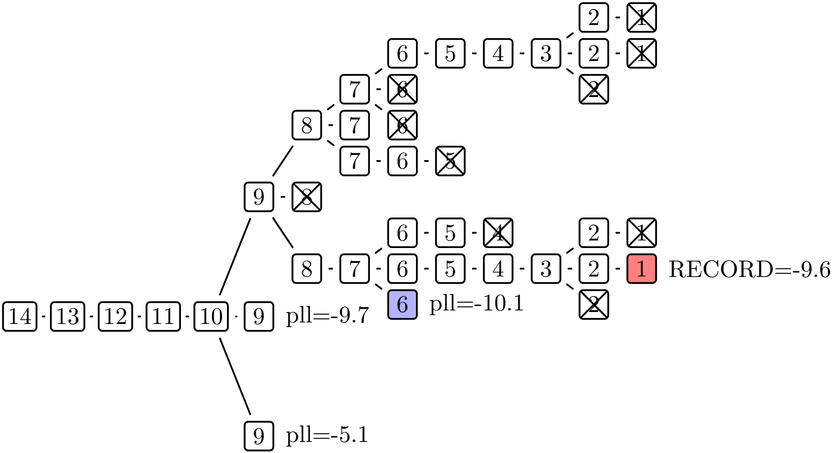
BnB on RLS tree, step 29



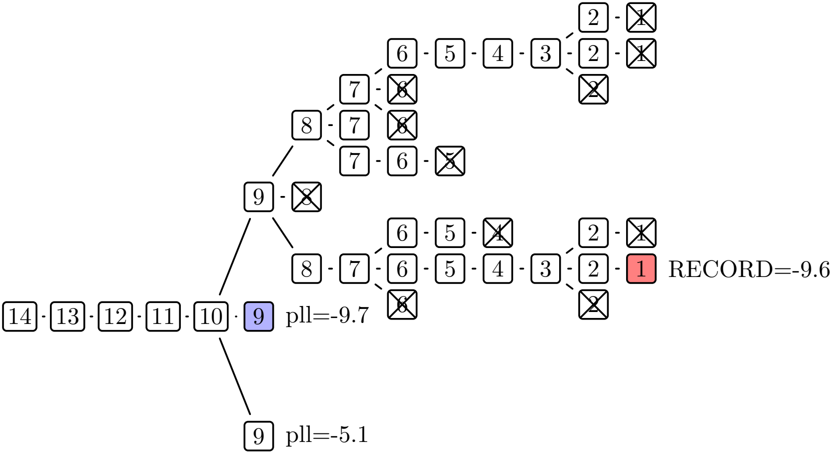
BnB on RLS tree, step 30



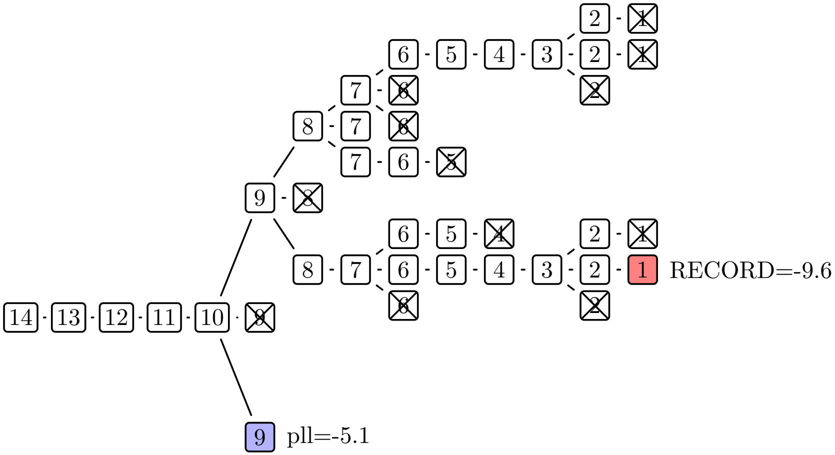
BnB on RLS tree, step 31



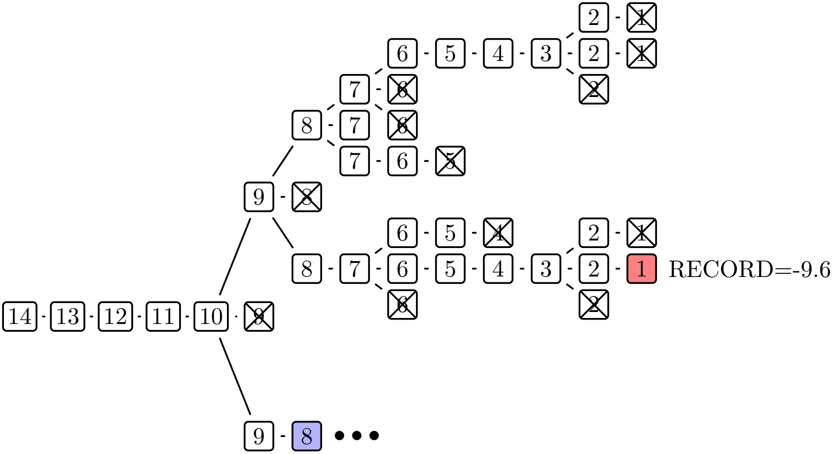
BnB on RLS tree, step 32



BnB on RLS tree, step 33



BnB on RLS tree, step 34



Refinements of the BnB for NRLS

- 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 $\rightarrow$ choice probabilities are not known (the goal is not to compute them)

1. Bounds for choice probabilities? Model specific
2. Yet, comparing two equilibria based on the already extended parts is possible $\rightarrow$ LR type test

$\Rightarrow$ Poly-algorithm with dichotomous decision rule

Battery of MC tests

A	B	C
Single equilibrium in the model Single equilibrium in the data	Multiple equilibria in the model Single equilibrium in the data	Multiple equilibria in the model Multiple equilibria in the data

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

Battery of MC tests: preliminary results

A

1. Fastest, small sample bias
2. Approaching MLE
3. MLE

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

B

1. Small sample bias
2. Failing due to multiplicity
3. Local extrema

C

1. Huge data requirements
2. Failing due to multiplicity
3. Curse of dimensionality

Monte Carlo setup (A and B)

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

- 100 random samples from a single equilibrium (one market)
- 10,000 observations per market/equilibrium
- Uniform distribution over state space $\leftrightarrow$ "ideal" data
- Data contains simulated discrete investment choices only

- Estimating one parameter in cost function

Monte Carlo A: no multiplicity

Number of equilibria in the model: 1
Number of equilibria in the data: 1
True value of parameter k1: 1.0

	mle-mpec	pml2step	npl	mle-bnb
Parameter: k1	4.0745	1.0146	1.0203	1.0206
MCSD	3.4974	0.0221	0.0237	0.0212
Bias	3.0745	0.0146	0.0203	0.0206
Log-likelihood	-12,989.73	-12,991.88	-12,987.10	-12,987.37
$\|\|\Psi(P) - P\|\|$	0.00	0.04	0.01	0.00
$\|\|\Gamma(v) - v\|\|$	0.00	0.27	0.15	0.00
Runs converged,	92.00	100.00	100.00	100.00
CPU time, sec	4.03	0.05	0.17	7.22
K-L divergence	4.78	0.00	0.00	0.00
abs deviation	0.38	0.02	0.01	0.00

Monte Carlo B, run 1: moderate multiplicity

Number of equilibria in the model: 5
Number of equilibria in the data: 1
True value of parameter k1: 1.0

	mle-mpec	pml2step	npl	mle-bnb
Parameter: k1	1.1090	0.9985	1.0009	0.9815
MCSD	0.0000	0.0000	0.0000	0.0000
Bias	0.1090	-0.0015	0.0009	-0.0185
Log-likelihood	-11,102.91	-11,102.91	-11,101.19	-11,092.05
$\|\|\Psi(P) - P\|\|$	0.00	0.03	0.01	0.00
$\|\|\Gamma(v) - v\|\|$	0.00	0.54	0.13	0.00
Runs converged,	100.00	100.00	100.00	100.00
CPU time, sec	2.35	0.04	0.23	11.13
K-L divergence	0.01	0.01	0.01	0.00
Abs deviation	0.03	0.03	0.04	0.01

Monte Carlo B, run 2: larger multiplicity

Number of equilibria in the model: 95
Number of equilibria in the data: 1
True value of parameter k1: 1.0

	mle-mpec	pml2step	npl	mle-bnb
Parameter: k1	1.0910	0.9948	1.0045	0.9970
MCSD	0.3202	0.0113	0.0094	0.0065
Bias	0.0910	-0.0052	0.0045	-0.0030
Log-likelihood	-6,714.92	-6,714.32	-6,722.46	-6,695.74
$\|\|\Psi(P) - P\|\|$	0.00	0.10	0.12	0.00
$\|\|\Gamma(v) - v\|\|$	0.00	1.22	0.94	0.00
Runs converged,	100.00	100.00	5.00	100.00
CPU time, sec	7.72	0.04	0.32	13.79
Mean K-L divergence	0.27	0.01	0.02	0.00
Mean abs deviation	0.06	0.04	0.06	0.00

Monte Carlo B, run 3: moderate multiplicity, bad start points

Number of equilibria in the model: 5
Number of equilibria in the data: 1
True value of parameter k1: 1.0

	mle-mpec	pml2step	npl	mle-bnb
Parameter: k1	6.7685	0.9948	1.0045	0.9970
MCSD	11.3401	0.0113	0.0094	0.0065
Bias	5.7685	-0.0052	0.0045	-0.0030
Log-likelihood	-6,709.38	-6,714.32	-6,722.46	-6,695.74
$\|\|\Psi(P) - P\|\|$	0.00	0.10	0.12	0.00
$\|\|\Gamma(v) - v\|\|$	0.00	1.22	0.94	0.00
Runs converged,	30.00	100.00	5.00	100.00
CPU time, sec	10.43	0.06	0.37	18.47
Mean K-L divergence	8.63	0.01	0.02	0.00
Mean abs deviation	0.25	0.04	0.06	0.00

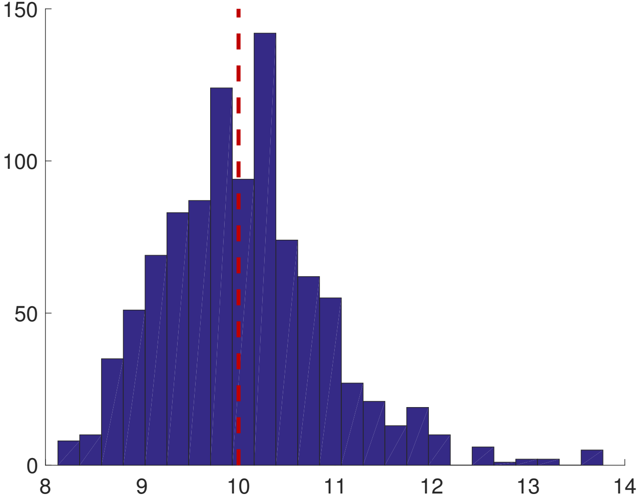
NRLS Monte Carlo setup (C)

- $n = 3$ 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
- Data contains simulated discrete investment choices only

- 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

$$0.321 \text{ (std=0.11635)}$$

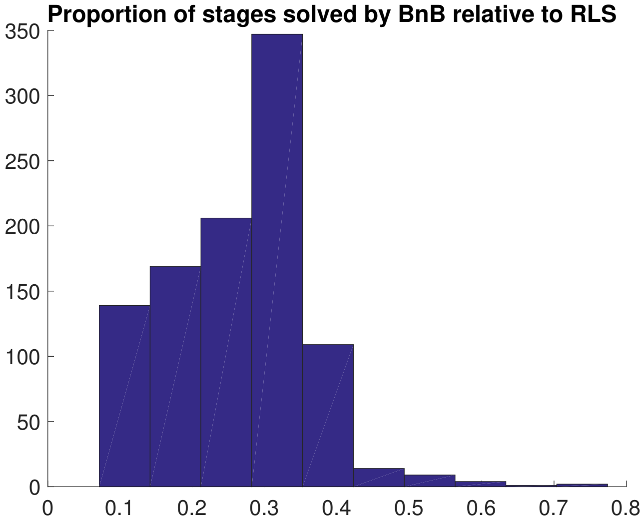
3. Average fraction of stages solved by BnB relative to RLS

$$0.263 \text{ (std=0.09725)}$$

4. All 3 MPE correctly identified by BnB in

98.4% of runs

Distribution of computational reduction factor



Conclusions

- Full solution MLE estimator for dynamic games of a particular type, namely directional dynamic games (DDGs)
- Nested loop: outer likelihood max + inner model solver
- Need to maximize over the set of all equilibria $\leftrightarrow$ daunting computational task
- Smart BnB algorithm not to waste time on unlikely MPE
- Further refinement of BnB bounding function based on statistical argument
- Horse race with existing estimators