Large State Space Techniques for Markov Decision Processes

Ron Parr
Duke University

Bob Givan
Purdue University

Outline for first half (Bob)

Backward search (regression) techniques

- Model minimization
- Structured dynamic programming

Forward search techniques

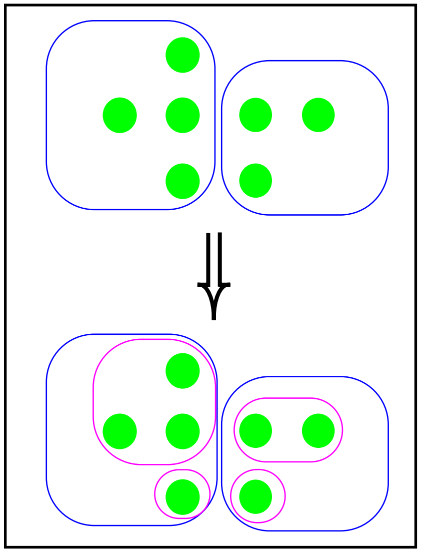
- Nondeterministic conformant planning
- Monte-Carlo Sampling

As time allows: Relational factoring

<u>Second half</u> (Ron): Value function approximation
Hierarchical abstraction

Backward Search Techniques

<u>Idea</u>: start with immediate reward definition and regress through action dynamics



- Initially group states with similar immediate reward
- Separate states with different horizon one value
- Separate states with different horizon two value....etc.

Model minimization carries this process to quiescence and then aggregates the resulting groups to form an explicit aggregate model amenable to traditional solution.

Backward Search Techniques

- Structured dynamic programming

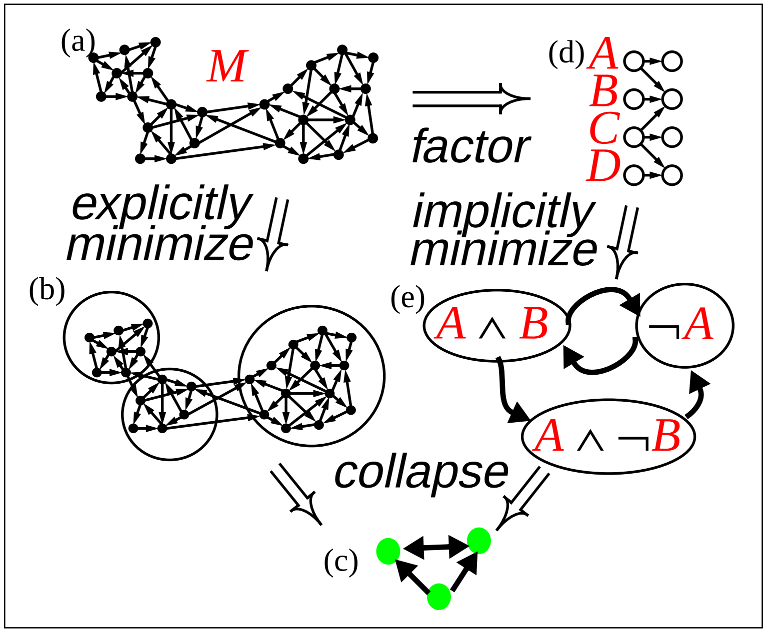
  [Boutilier, Dearden, and Goldszmidt, AIJ-2000]

- Model minimization

  [Dean and Givan, AAAI-97]

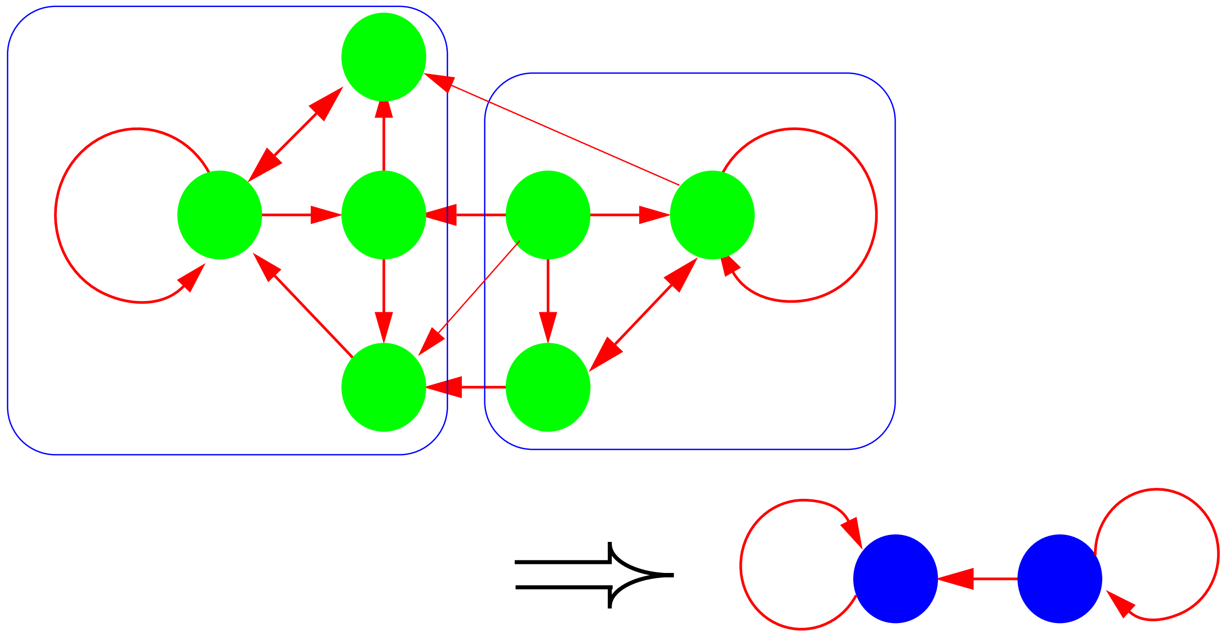
Model Minimization Overview

1. Constructing aggregate-state MDPs
2. Operating directly on factored representations



Our methods are inspired by work in the model checking community on reducing non-deterministic systems, in particular [Lee and Yannakakis, STOC 1992].

State Space Partitions & Aggregation



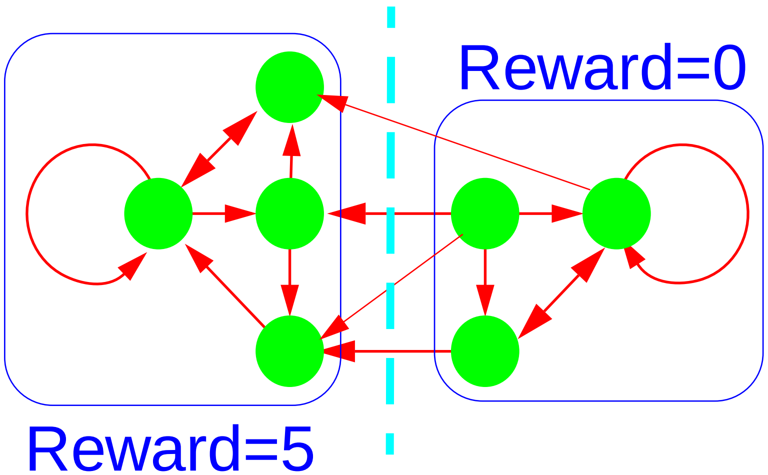
aggregate MDP

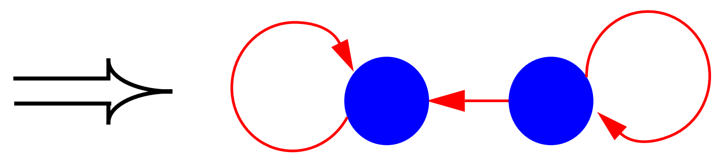
Under what conditions does the aggregate MDP capture what we want to know about the original MDP?

Desired Partition Properties

- Reward Homogeneity
- Dynamic Homogeneity

} *stochastic bisimulation*





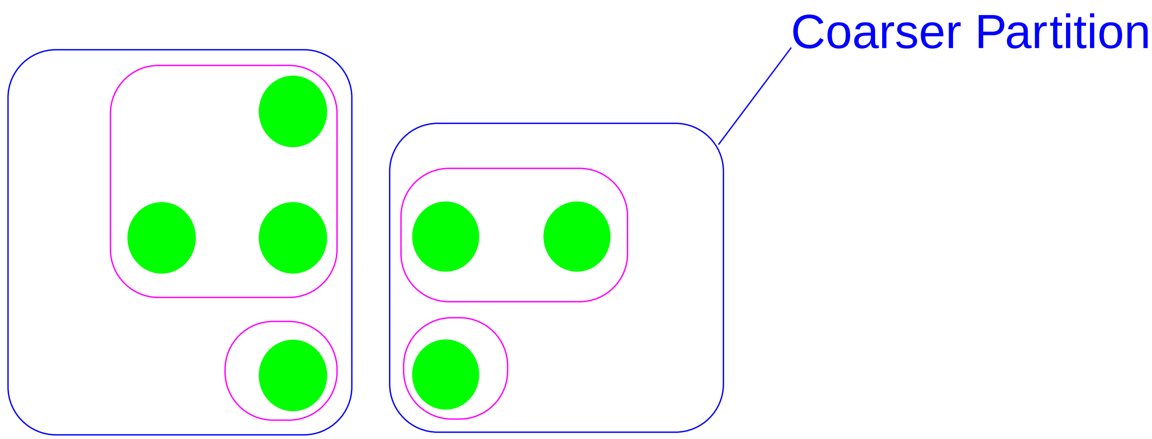
equivalent aggregate MDP

Theorem: *Each equivalent aggregate MDP*[1] *has the same policy values and optimal policies as original MDP.*

[1] *There can be many...*

Constructing Homogeneous Partitions

Definition: We say P_1 refines P_2, written $P_1 \ll P_2$, if P_2 can be constructed from P_1 by splitting blocks.

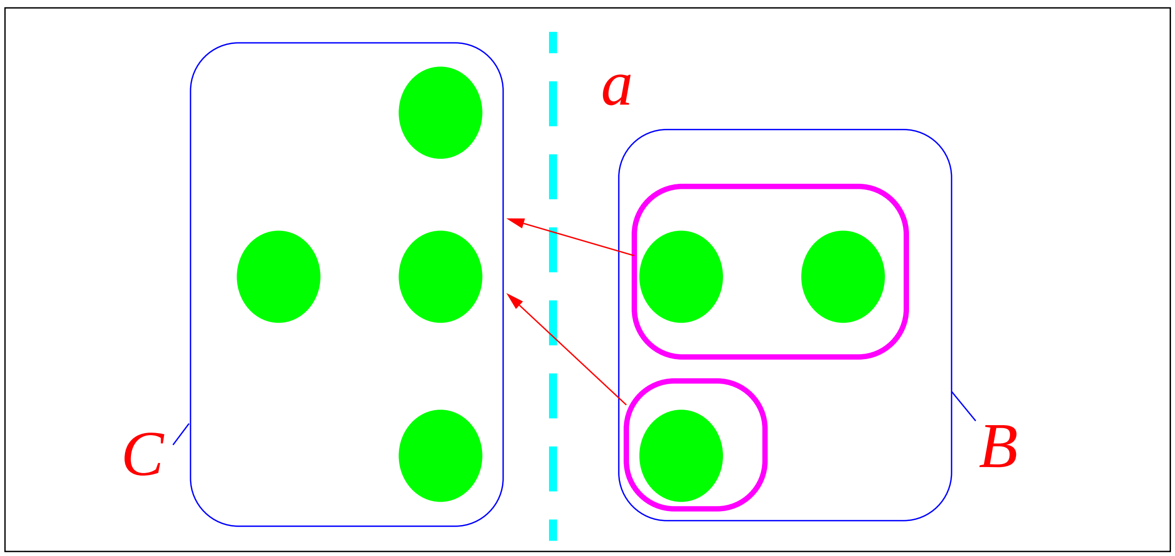


Every homogeneous partition refines the reward partition.

Refining a Partition

Let P be a partition which every homogeneous partition refines. How can we refine P maintaining this property?

$\mathrm{SPLIT}(P, B, C, a)$ is a new partition with this property:



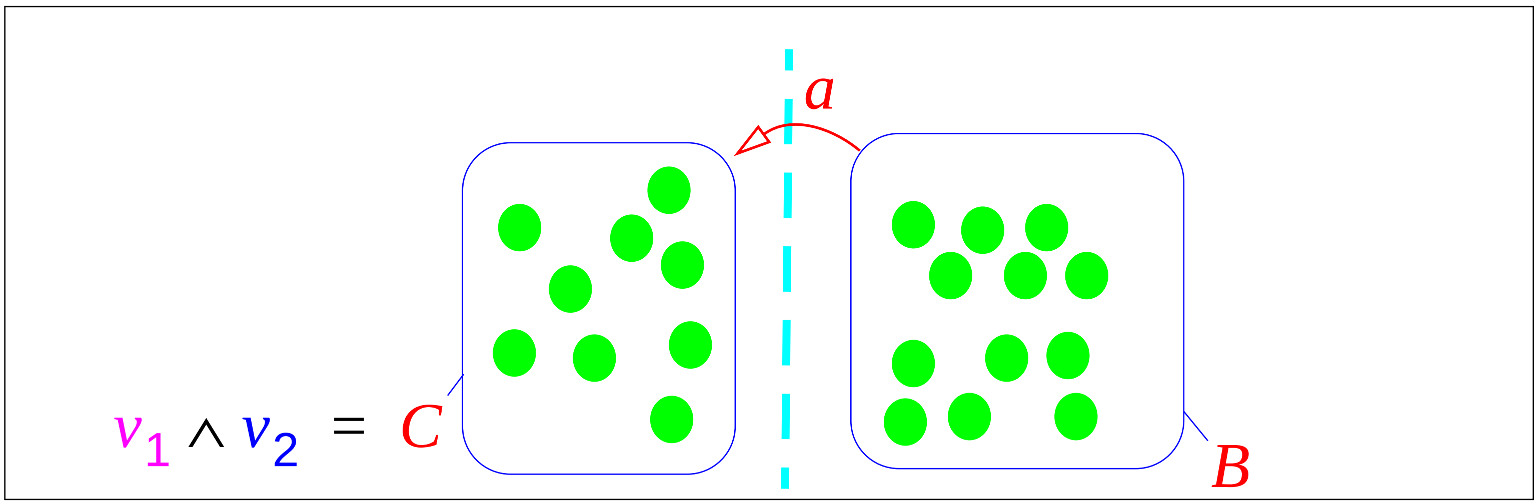
Thm: Repeating SPLIT derives *smallest homogeneous* P

Complexity

- Number of calls to SPLIT is quadratic in the number of states in the resulting minimal model.
- Cost of each call to SPLIT depends on the representation for both the MDP and the partitions.
- [Goldsmith&Sloan AIPS-2000] – SPLIT is NP^{PP}-hard for factored representations

The Factored SPLIT Operation

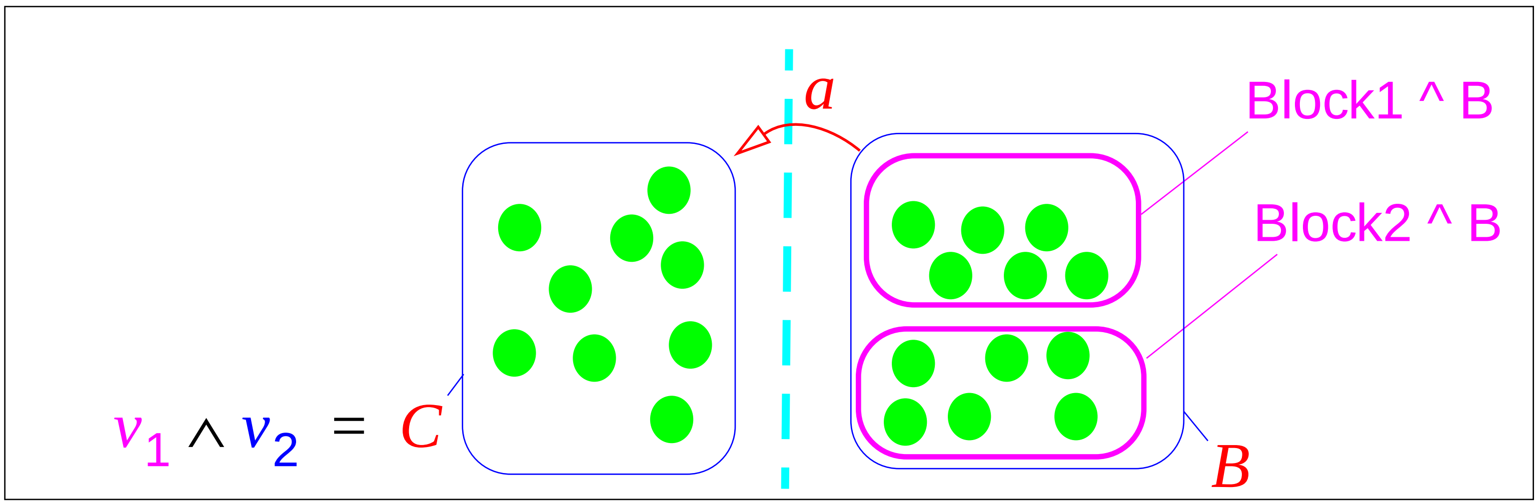
Each variable in destination block formula induces a (factored) partition of source block:



A clustering of the intersection of these partitions is the desired splitting of B.

The Factored SPLIT Operation

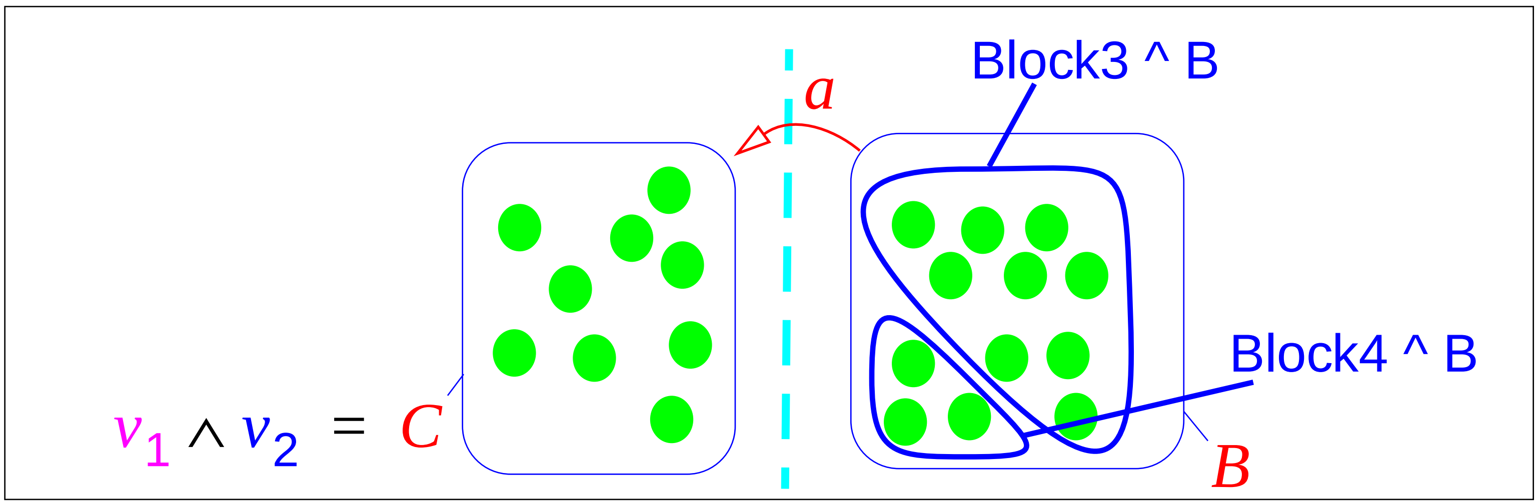
Each variable in destination block formula induces a (factored) partition of source block:



A clustering of the intersection of these partitions is the desired splitting of B.

The Factored SPLIT Operation

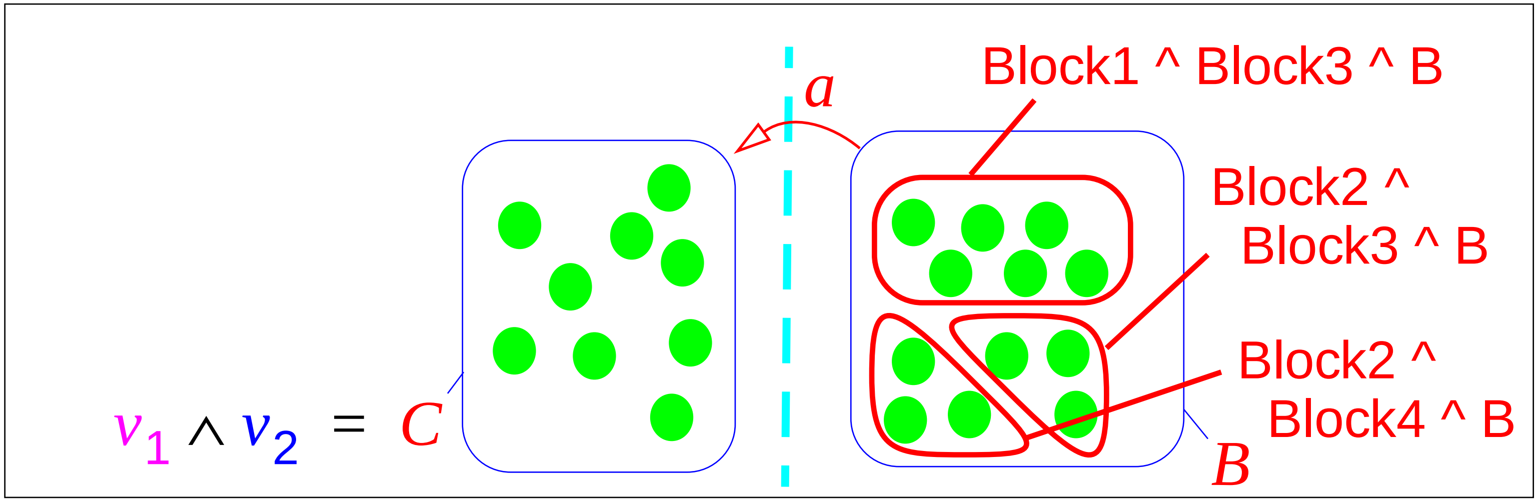
Each variable in destination block formula induces a (factored) partition of source block:



A clustering of the intersection of these partitions is the desired splitting of B.

The Factored SPLIT Operation

Each variable in destination block formula induces a (factored) partition of source block:



A clustering of the intersection of these partitions is the desired splitting of B.

Algorithm Summary

Input: A factored MDP.

Output: An explicit MDP, possibly with much smaller state space. Suitable for traditional MDP algorithms.

Pseudocode: While some a, B, C remain untried
Select untried a and blocks B, C in P
$P \leftarrow \text{SPLIT}(P, B, C, a)$

Complexity: Polynomial number of SPLIT calls in size of resulting MDP. Block formulas may grow in size exponentially—simplification is NP-hard. *Finding the minimal equivalent aggregate MDP is NP-hard.*

Extensions

- Relaxation of homogeneity requirement allows approximate minimization
- Large factored action spaces can be automatically incorporated, forming a partition of S A.
  [Dean, Givan, Kim AIPS-98]
  - Yields an automatic detection of symmetry, e.g. finds circular symmetry in dining philosophers [Ravindran&Barto, 2001]

Structured Dynamic Programming

Predates model minimization

Basic MDP review:

- Finite horizon value functions approximate true value
- Approximation improves as horizon increases
- Horizon $n+1$ values from horizon n by regression

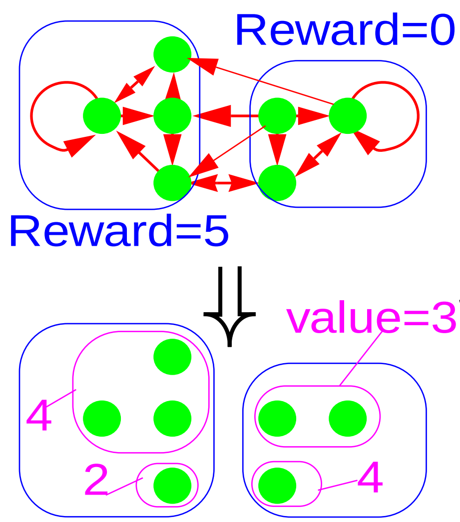
Critical observations:

- Value functions can be kept as labelled partitions
- Regression can be computed directly on partitions using provided factored action representation

Comparison to Model Minimization

Similarities

- Start with reward partition
- Split blocks using factored action dynamics



Differences

- Value computations interleaved with block splitting
- Splitting not "opportunistic" but follows horizon
- Can reaggregate to exploit "coincidences"
- No reduced equivalent model formed

Forward Search Methods

Nondeterministic BDD-based methods[Bertoli+, IJCAI-01]

Sampling methods surveyed/evaluated in my later talk

- Unbiased sampling [Kearns et al., IJCAI-99]
- Policy rollout [Bertsekas&Castanon, Heuristics 1999]
- Parallel Policy Rollout [Givan et al., under review]
- Hindsight Optimization [Givan et al., CDC 2000]

Nondeterministic BDD-based Methods

Nondeterministic domains

- [Cimatti, Roveri, Traverso, AAAI-98][1] Universal plans
- [Bertoli, Cimatti, Roveri, IJCAI-01] Conformant plans
- [Bertoli et al., IJCAI-01] Partial observability

Basic idea:

- represent state sets as BDDs.
- heuristically expand a tree of reachable state sets
- tree arcs correspond to actions

---

1. Proceeds backward from goal

Relational Factoring

[Boutilier et al., IJCAI-01]

- State space is set of first-order models
- Represent each deterministic realization of each action using the situation calculus
    - downside: could be one per state in worst case
- SPLIT can be worked out using classical planning regression
- Current implementation solves very small problems relying on human hand simplification of formulas

- Ron Parr spoke at this point for half an hour on value function approximation methods.