REAL-TIME DYNAMIC PROGRAMMING

BARTO ET AL., 1993

Jonathan Campbell
COMP-767
February 3, 2017

MAIN IDEA

- Like value iteration, but better in certain conditions
  - Update values of states that have higher transition probabilities.
- This presentation:
  - Value iteration
  - Gauss-Seidel DP
  - Asynchronous DP + RTDP

IMPLEMENTATION

- Uses Gridworld RL framework from UC Berkeley AI course
    - http://ai.berkeley.edu/reinforcement.html
- Grid with goal states in middle
    - Outer 2 rows/columns have only 1% probability to be entered from neighbouring states

DP: VALUE ITERATION

Update <u>all states</u> in each iteration until convergence.
(including unlikely or bad states)

$$f_{k+1}(i) = \min_{u \in U(i)} \left[c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f_k(j) \right]$$

For n states and m actions: O(mn^2) operations

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

GAUSS-SEIDEL DP

Update all states in each iteration until convergence, with each state update using most recent values.

$$f_{k+1}(i) = \min_{u \in U(i)} \left[c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f(j) \right]$$

$$\text{where} \quad f(j) = \begin{cases} f_{k+1}(j), & \text{if } j < i \\ f_k(j), & \text{otherwise} \end{cases}$$

Generally converges faster than regular value iteration (depends on state ordering)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	$\boxed{0.00}$	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ASYNCHRONOUS DP

Update <u>a subset of all states</u> in each iteration until convergence.

$$f_{k+1}(i) = \begin{cases} \min_{u \in U(i)} \left[c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f_k(j) \right], & \text{if } i \in S_k \\ f_k(i), & \text{otherwise} \end{cases}$$

Each iteration updates a minimum of one state value.

(Multi-core implementation:
each processor handles a certain subset)

RTDP

- Execute asynchronous DP concurrently with control process
- Start at an initial state.
  - Update value of current state.
  - Choose action w.r.t. greedy policy and go to next state.
  - Repeat.

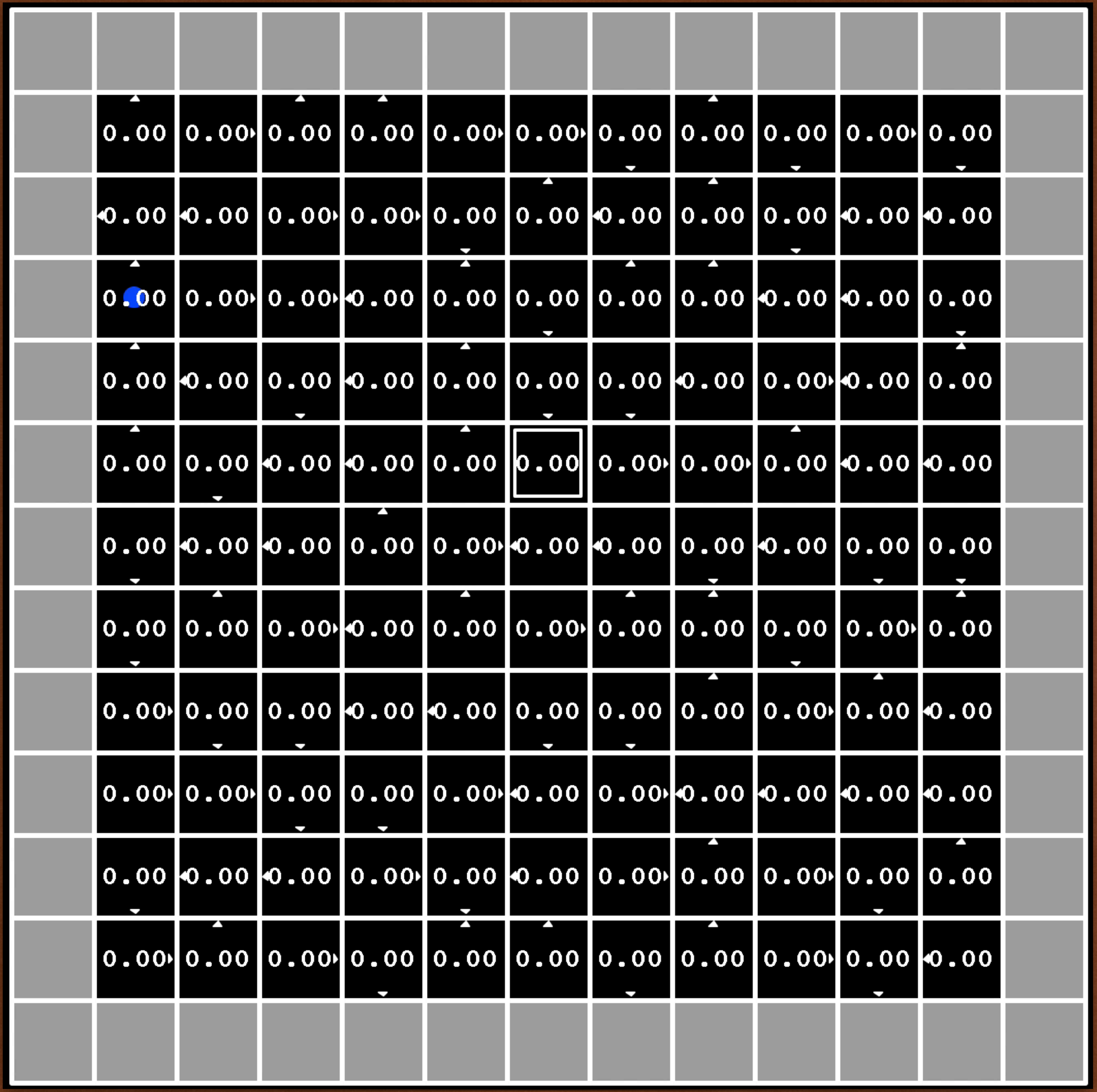
RTDP

Use sample (bounded) trajectories through MDP
to determine which states to update.

$$f_{k+1}(i) = \begin{cases} \min_{u \in U(i)} \left[c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f_k(j) \right], & \text{if } i \in S_k \\ f_k(i), & \text{otherwise} \end{cases}$$

$$\text{where } s_t \in S_k$$

Each iteration updates a minimum of one state value;
computation is focused on relevant states.


0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

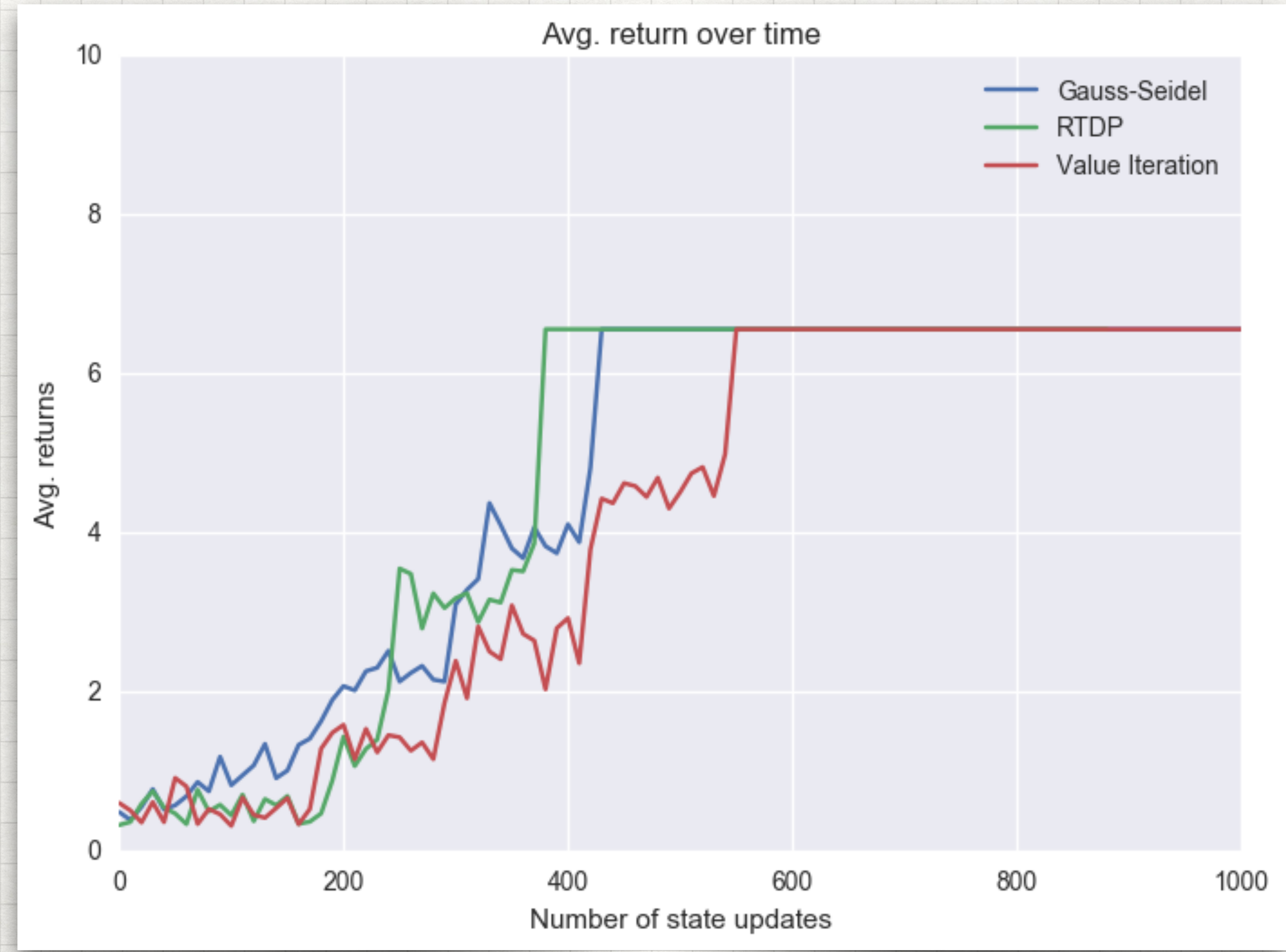
RTDP - STATE UPDATES

- Which states to include in updates?
  - Current state (mandatory)
  - States based on prior knowledge (guided exploration)
  - Neighbors of current states
  - Lots of other suggestions
- Good choice of these states speeds convergence.

CONVERGENCE

- Value iteration / Gauss-Seidel DP:
  - Repeated iterations will converge to optimal policy (when $\gamma < 1$)
  - (Infeasible for large state spaces.)
- RTDP
  - Repeated trajectories will converge to optimal policy on the set of states reachable under an optimal policy from initial state(s).
  - Depends on selection of initial state.
    - (Could randomize initial state, so all states would be reachable.)

COMPARISON



Avg. return over time
Avg. returns
Number of state updates
0
2
4
6
8
10
0
200
400
600
800
1000
Gauss-Seidel
RTDP
Value Iteration