# Pitfalls of Goal-Oriented Planning

Andrey Kolobov
*(Joint work with Mausam, Daniel S. Weld, and Hector Geffner)*
Computer Science and Engineering
University of Washington, Seattle

# SSP –The Main Goal-Oriented MDP Class

[Bertsekas, 1995]

SSP MDP is a tuple <*S, A, T, C, G, ($s_0$)*>, where:

- *S* is a finite state space
- *A* is a finite action set
- *T* is a stationary transition function
- *C* is a stationary cost function
- *G* is a set of absorbing cost-free goal states
- $s_0$ is an optional initial state

**Under two conditions:**

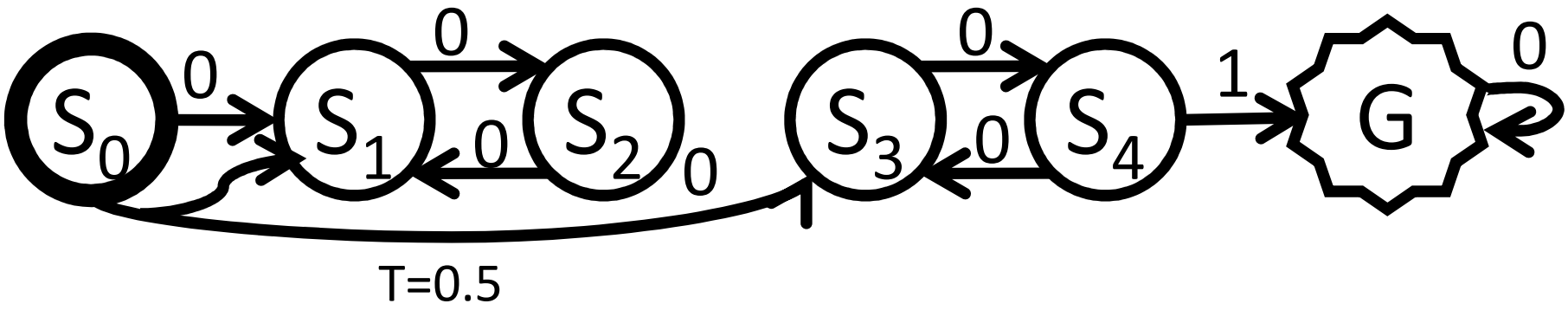
- There is a *proper policy* (reaches a goal with P=1 from *all* states)
- Every *improper policy* incurs a cost of ∞ from every state from which it does not reach the goal with $P_G = 1$

Disallows dead ends

Disallows 0-reward "loops"

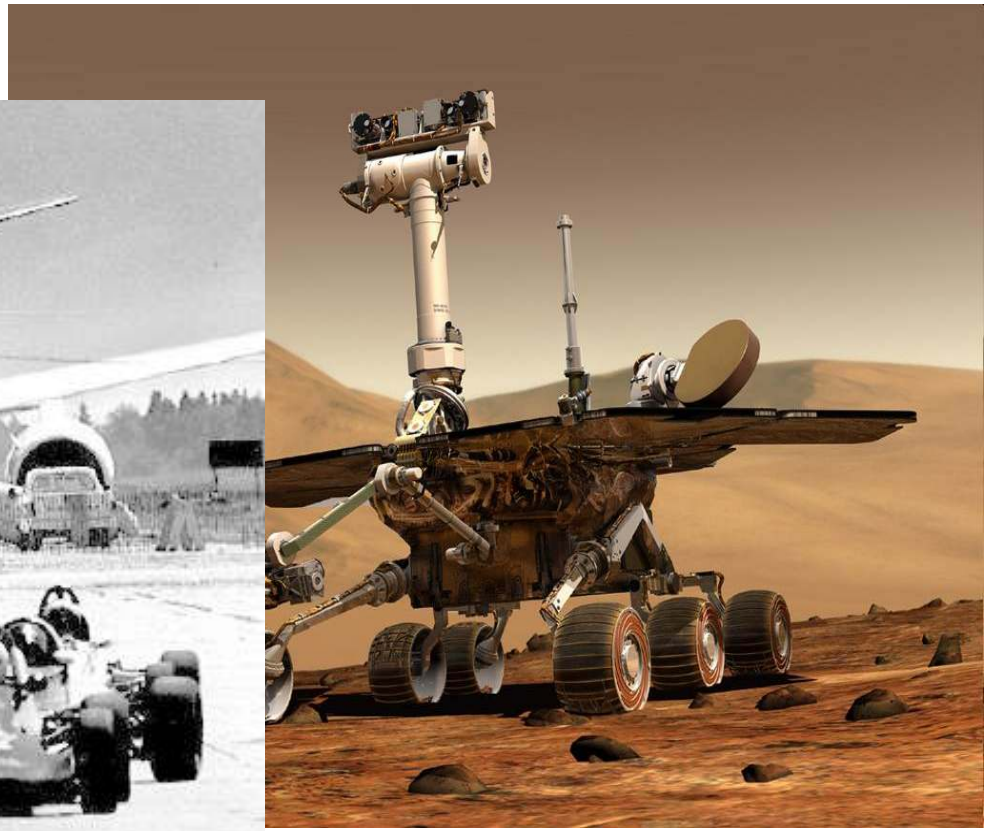
# Why Might We Want Zero-Reward Loops?

- Identifying a policy with the highest P of reaching the goal.
  - Action rewards are 0 (they are irrelevant)
  - Reaching the goal yields reward = 1



# Why Might We Want Dead Ends?

- Because many scenarios have them!



# Contributions of This Work

- Goal-oriented MDP classes that allow zero-reward loops
  - MAXPROB MDPs
  - Generalized SSP MDPs (GSSP)
- Goal-oriented MDP classes that allow dead ends:
  - SSP MDPs with Avoidable Dead Ends (SSPADE)
  - SSP MDPs with Unavoidable Dead Ends
    - With finite dead-end penalty (fSSPUDE)
    - With infinite penalty (iSSPUDE)
- Optimal heuristic search algorithms for all of them

# Outline

- Motivation
- Background
- Dealing with zero-reward loops
  - MAXPROB MDPs
  - Generalized SSP MDPs (GSSP)
- Dealing with dead ends
  - SSP MDPs with Avoidable Dead Ends (SSPADE)
  - SSP MDPs with Unavoidable Dead Ends
    - With finite dead-end penalty (fSSPUDE)
    - With infinite penalty (iSSPUDE)
- Class equivalences and experiments

# Background: Heuristic Search

[Bonet, Geffner, IJCAI'03]

- The most efficient optimal technique for SSP MDPs with $s_0$
  - If equipped with a good heuristic

- Find-and-Revise: HS for *SSP* [Bonet & Geffner, 2003]
  - 1 Start with a heuristic value function $V \leftarrow h$
    2 **while** *$V$'s greedy graph $G^V_{s_0}$ contains a state $s$ with $Res^V(s) > \epsilon$* **do**
    3 | FIND a state $s$ in $G^V_{s_0}$ with $Res^V(s) > \epsilon$
    4 | REVISE $V(s)$
    5 **end**
    6 return a $\pi^V$

    Bellman backup

    $V'(s) \leftarrow \min_{a \text{ in } A} [\Sigma_{s' \text{ in } S} T(s,a,s') [C(s,a,s') + V(s')]]$
  - Includes LRTDP, LAO*, etc.

# Outline

- Motivation
- Background
- Dealing with zero-reward loops
  - MAXPROB MDPs
  - Generalized SSP MDPs (GSSP)
- Dealing with dead ends
  - SSP MDPs with Avoidable Dead Ends (SSPADE)
  - SSP MDPs with Unavoidable Dead Ends
    - With finite dead-end penalty (fSSPUDE)
    - With infinite penalty (iSSPUDE)
- Class equivalences and experiments

# MAXPROB MDPs

[Kolobov, Mausam, Weld, Geffner, ICAPS'11]

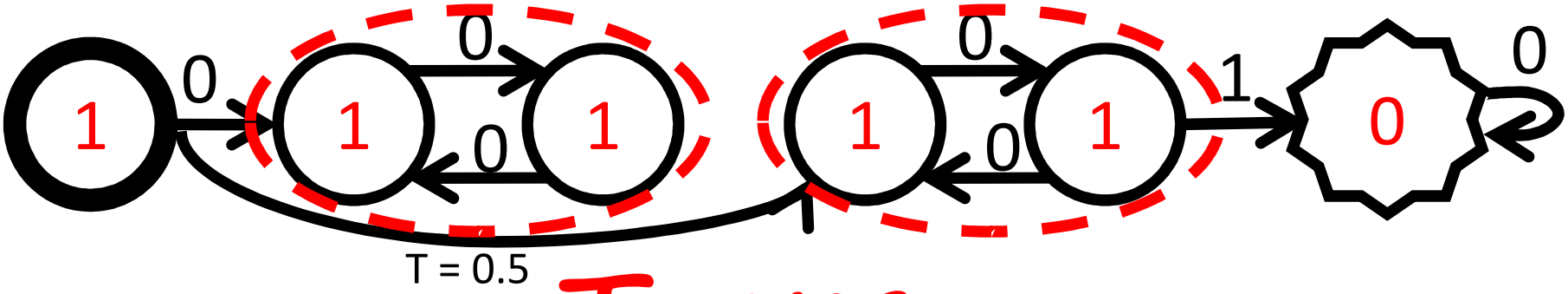
MAXPROB MDP is a tuple <*S, A, T, G*>, where:

- *S* is a finite state space
- *A* is a finite action set
- *T* is a stationary transition function
- Action rewards are 1 for transitions to goal, 0 otherwise
- *G* is a set of absorbing cost-free goal states

**The objective is to maximize the probability $P_G$ of reaching the goal from every state**

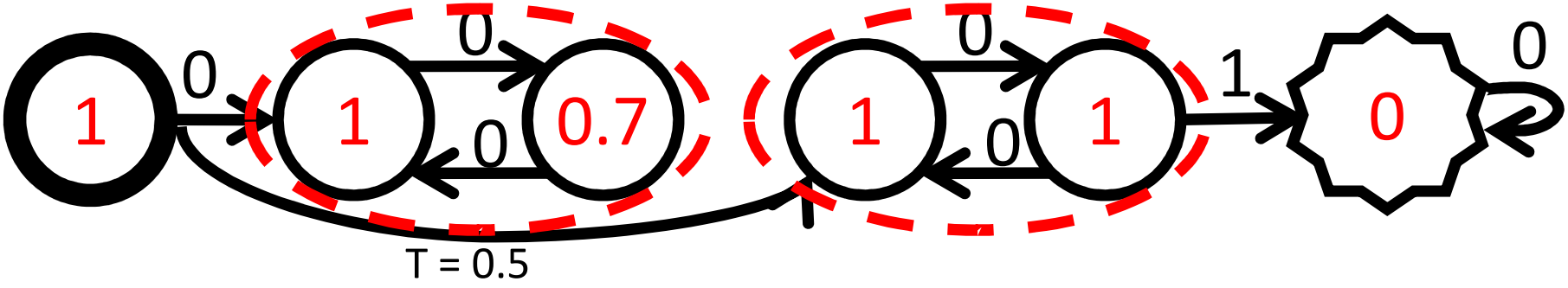
# Zero-Reward Loops: What Breaks?

- *V** isn't the unique fixed point of Bellman backup:




Traps

- Full Bellman backup may never converge:



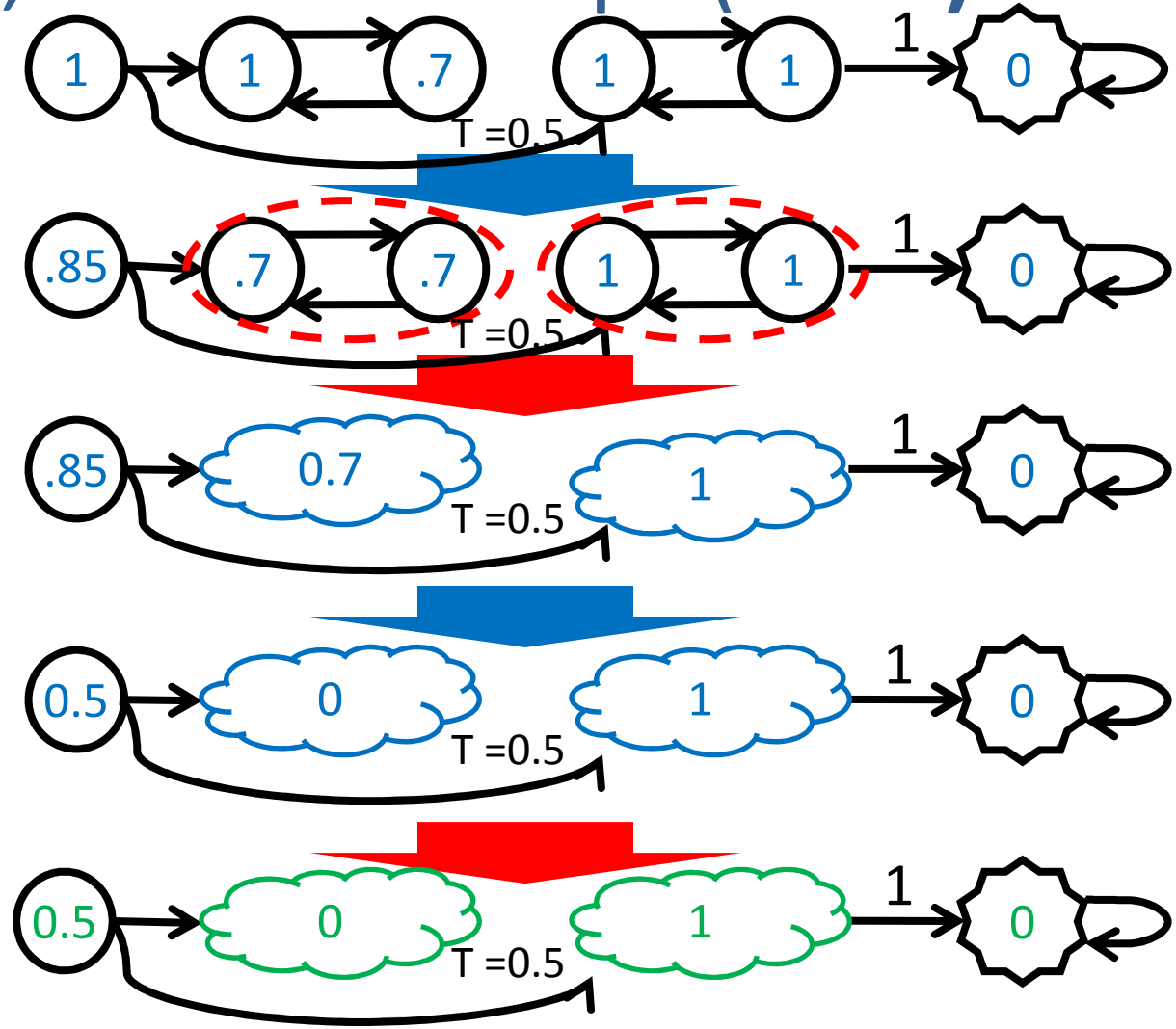
# Heuristic Search for MAXPROB: Find, Revise, Eliminate Traps (**FRET**)

Repeat

Start with an admissible $V_0$

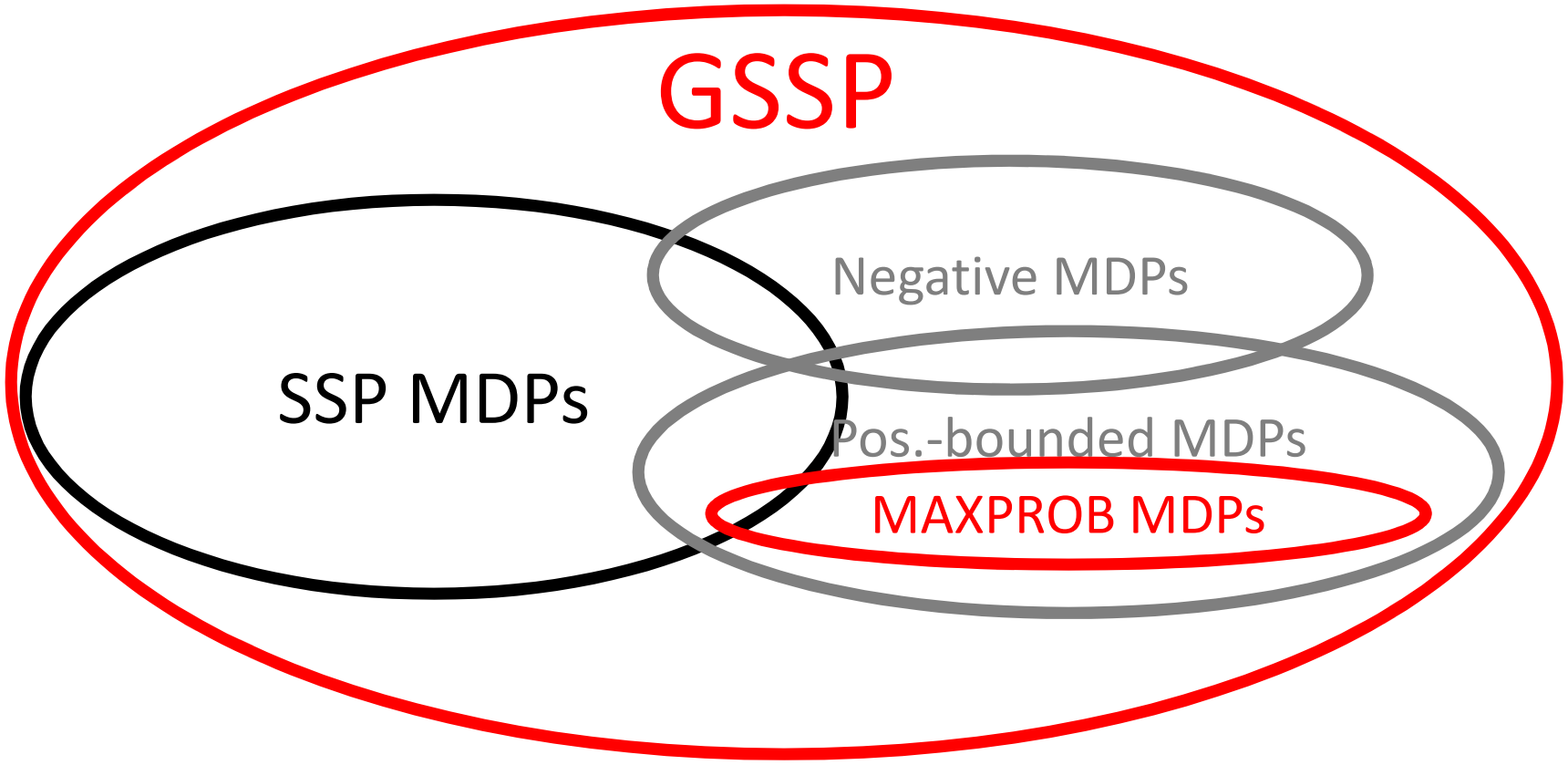
Run **F&R** until convergence

**Eliminate Traps** in the resulting $V_i$



# Generalized SSPs

An extension of *SSP* that allows zero-reward loops




Like MAXPROB MDPs, can be solved with FRET

# Outline

- Motivation
- Background
- Dealing with zero-reward loops
  - MAXPROB MDPs
  - Generalized SSP MDPs (GSSP)
- Dealing with dead ends
  - SSP MDPs with Avoidable Dead Ends (SSPADE)
  - SSP MDPs with Unavoidable Dead Ends
    - With finite dead-end penalty (fSSPUDE)
    - With infinite penalty (iSSPUDE)
- Class equivalences and experiments

# Dead Ends: What Breaks?

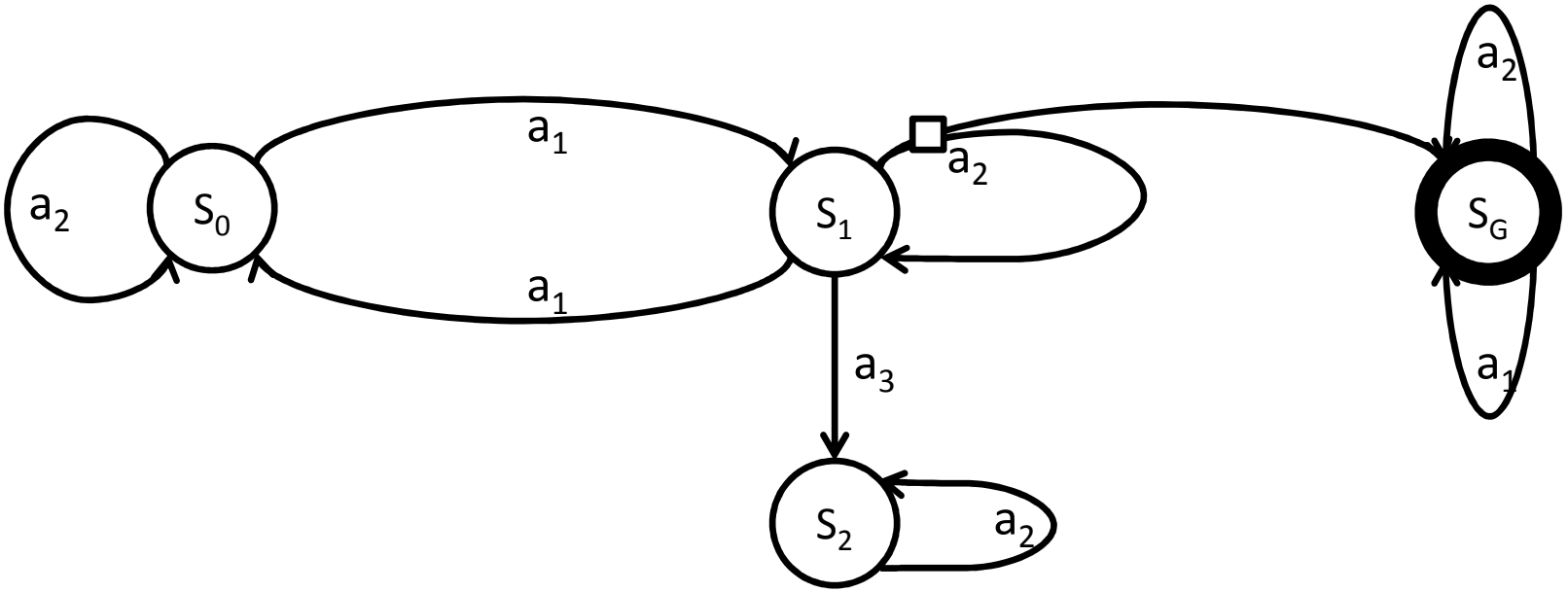
- Value iteration & policy iteration never halts
  - Since some states have $V(s) = \infty$

- If a dead end $s$ is unavoidable from $s_0$, comparing policies' expected cost is meaningless
  - Because $V^{\pi}(s) = \infty$ for all $\pi$

# SSPADE: SSP with Avoidable Dead Ends

[Kolobov, Mausam, Weld, UAI'12]

- D.e.s may be avoidable *from $s_0$* via optimal policy




- Need only "relevant" states to get the "right" value

# SSPADE: SSP with Avoidable Dead Ends

SSP MDP with avoidable dead ends (SSPADE) is a tuple $<S, A, T, C, G, s_0>$, where:

- $S$ is a finite state space
- $A$ is a finite action set
- $T$ is a stationary transition function
- $C$ is a stationary cost function
- $G$ is a set of absorbing cost-free goal states
- $s_0$ is a known initial state

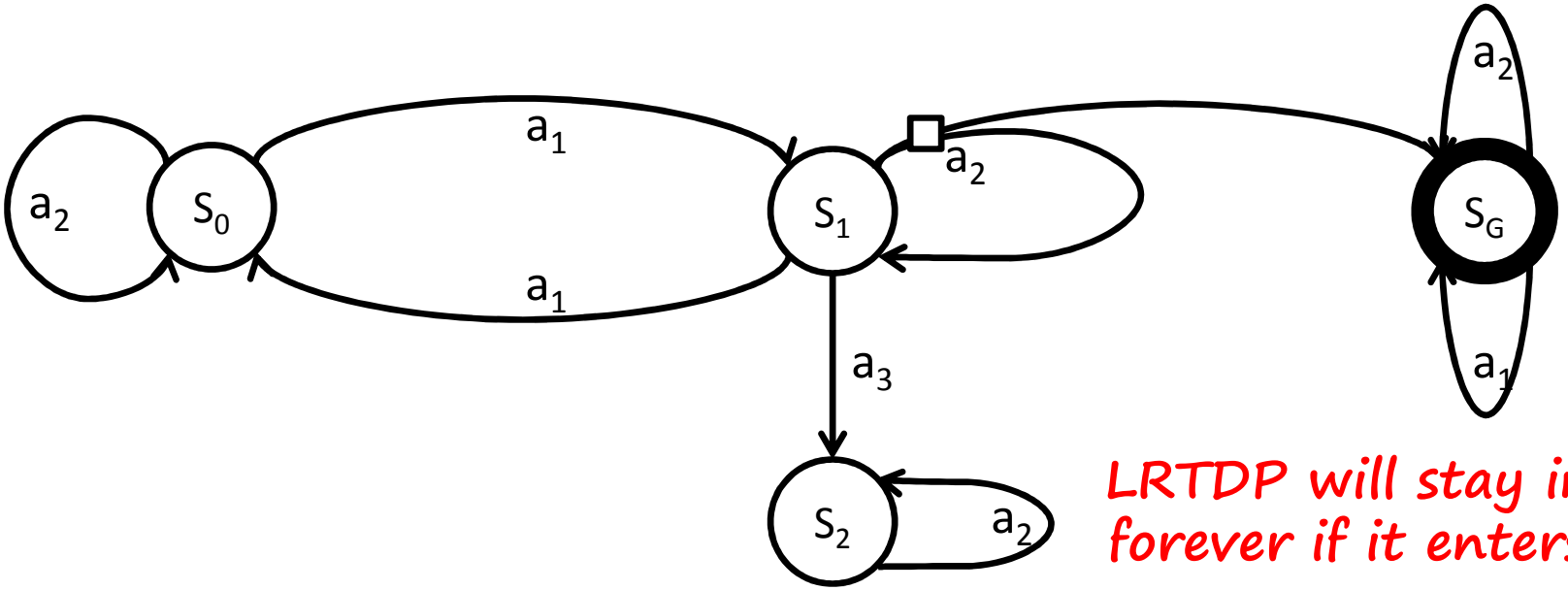
} Same as for SSP

**Under two conditions:**

- There is a *proper policy* closed w.r.t. $s_0$ (reaches a goal with $P_G = 1$ from $s_0$)
- Every *improper policy* incurs a total expected cost of ∞ from every state from which it does not reach the goal with $P_G = 1$

# Heuristic Search for SSPADE

- The F&R schema works
  - Need to modify some F&R algorithms, e.g., LRTDP:




  - For LRTDP: limit the trial length to some *N*

# If Dead Ends are Unavoidable from $s_0$…

- Alternatives:
  - Assume there is a finite penalty on reaching a dead end

  *or*

  - Fix the optimization criterion to deal with ∞ penalties

# fSSPUDE MDPs: Finite Penalty on Dead Ends

[Kolobov, Mausam, Weld, UAI'12]

- Agent pays penalty *P* when hitting a dead end
  - Kind of
  - Yields counterintuitive results, but can be fixed
- Equivalent to SSP with special $a_{stop}$ action
  - Leads directly to a goal and costs *P*
- F&R for SSP MDPs works on fSSPUDE MDPs too

# Infinite Penalty on Unavoidable Dead Ends

- Raises several issues:
  - Comparison between policies becomes uninformative
  - Fixed-point methods don't converge
  - Etc.

- Proposed solution: new optimization criterion
  - *Among policies reaching a goal with the highest possible probability, choose those that do so with the lowest expected cost*

# iSSPUDE Definition

[Kolobov, Mausam, Weld, UAI'12]

SSP MDP with unavoidable dead ends and infinite dead-end penalty (iSSPUDE) is a tuple $<S, A, T, C, G, s_0>$, where:

- $S$ is a finite state space
- $A$ is a finite action set
- $T$ is a stationary transition function
- $C$ is a stationary cost function
- $G$ is a set of absorbing cost-free goal states
- $s_0$ is a known initial state

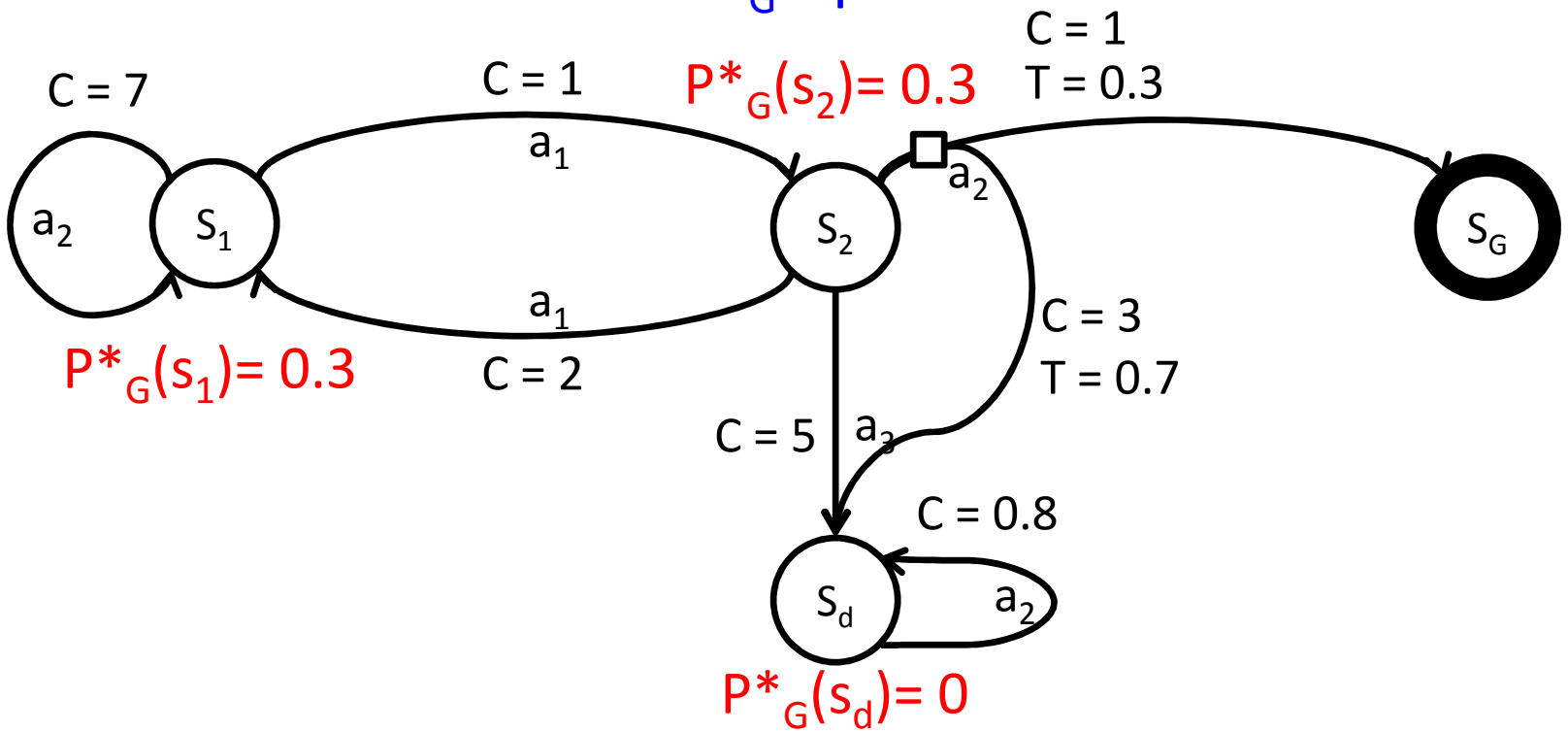
Same as for SSP

**Under a condition:**

- ~~There is a *proper policy* (reaches a goal with $P_G = 1$)~~
- Every *improper policy* incurs the total expected cost of ∞ from every state from which it does not reach the goal with $P_G = 1$

# Heuristic Search for iSSPUDE: Step 1

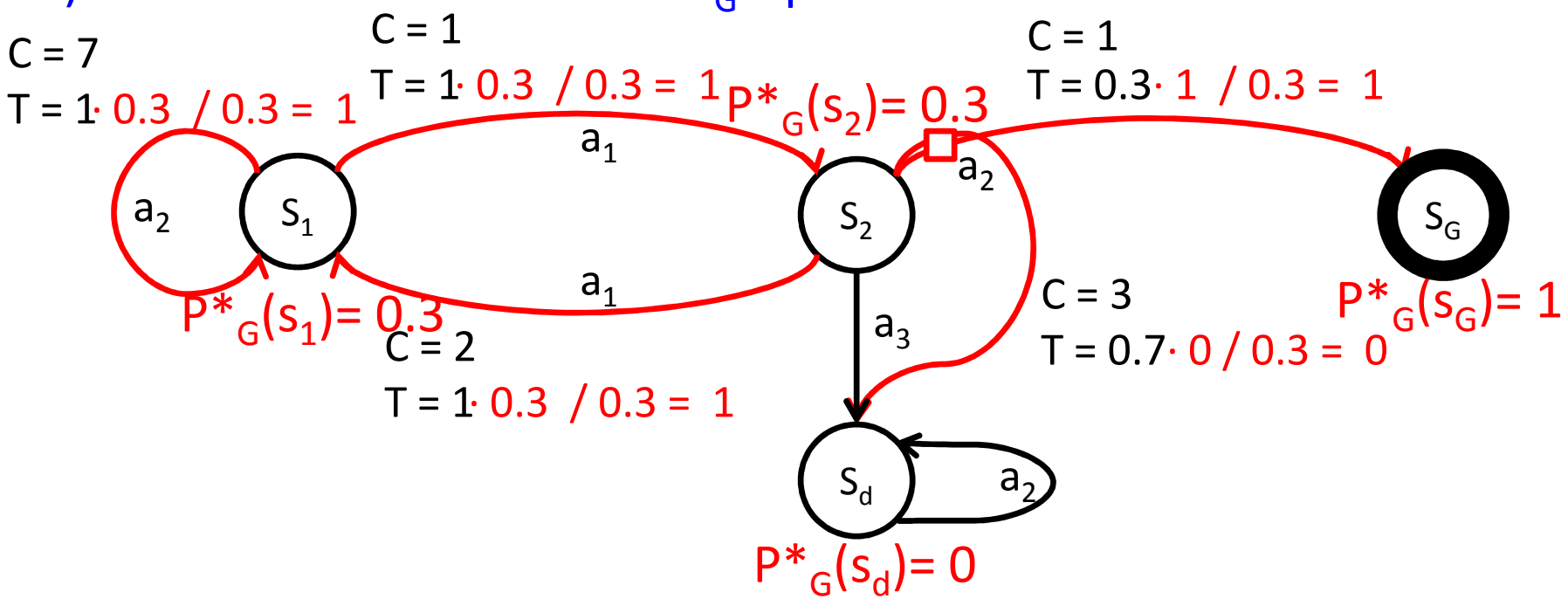
1) Solve MAXPROB to find $P^*_G$-optimal actions




- All $P^*_G$ -optimal actions are greedy w.r.t. $P^*_G$
- Solve MAXPROB using FRET

# Heuristic Search for iSSPUDE: Step 2

2) Solve an MDP over the $P^*_G$-optimal actions




- For each s, a, s', let $T'(s,a,s') = T(s,a,s')P^*_G(s')/P^*_G(s)$
- Intuitively: bias T so that only trajectories to goal remain
- The resulting MDP is an SSP!
- Solve this SSP MDP with F&R

# Heuristic Search for iSSPUDE: Summary

1) Identify the actions used in policies with the optimal $P^*_G$ using FRET

2) Out of the $P^*_G$-optimal actions, construct a policy with the lowest expected cost of trajectories that reach the goal using F&R

# Outline

- Motivation
- Background
- Dealing with zero-reward loops
  - MAXPROB MDPs
  - Generalized SSP MDPs (GSSP)
- Dealing with dead ends
  - SSP MDPs with Avoidable Dead Ends (SSPADE)
  - SSP MDPs with Unavoidable Dead Ends
    - With finite dead-end penalty (fSSPUDE)
    - With infinite penalty (iSSPUDE)
- Class equivalences and experiments

# Equivalence

**Theorem 1:** For a sufficiently large finite penalty $P_{thres}$, optimal policies for an fSSPUDE MDP are also optimal for the corresponding MAXPROB MDP.

# Equivalence

**Theorem 2:** For a $P < P_{thres}$, an optimal policy for an fSSPUDE MDP can be arbitrarily worse in terms of $P^*_G$ than an optimal policy of the corresponding MAXPROB MDP.

# Experimental Results

- Which is harder to solve, MAXPROB or fSSPUDE with a high penalty?
  - If dead-end-identification heuristics are used, MAXPROB
  - If they aren't used, depends on fSSPUDE's penalty *P*
- Solving MAXPROB takes much memory if $P^*_G = 1$
  - Computing $P^*_G$ then visits all states reachable from $s_0$

# Conclusions

- Proposed two MDP classes that admit zero-reward loops
  - MAXPROB
  - GSSP
- Proposed three MDP classes that admit dead ends
  - SSPADE
  - fSSPUDE
  - iSSPUDE
- Proposed heuristic search algorithms for each
- Next step: solving Stochastic Longest-Path problems, a class with no constraints on action costs and proper policy existence.