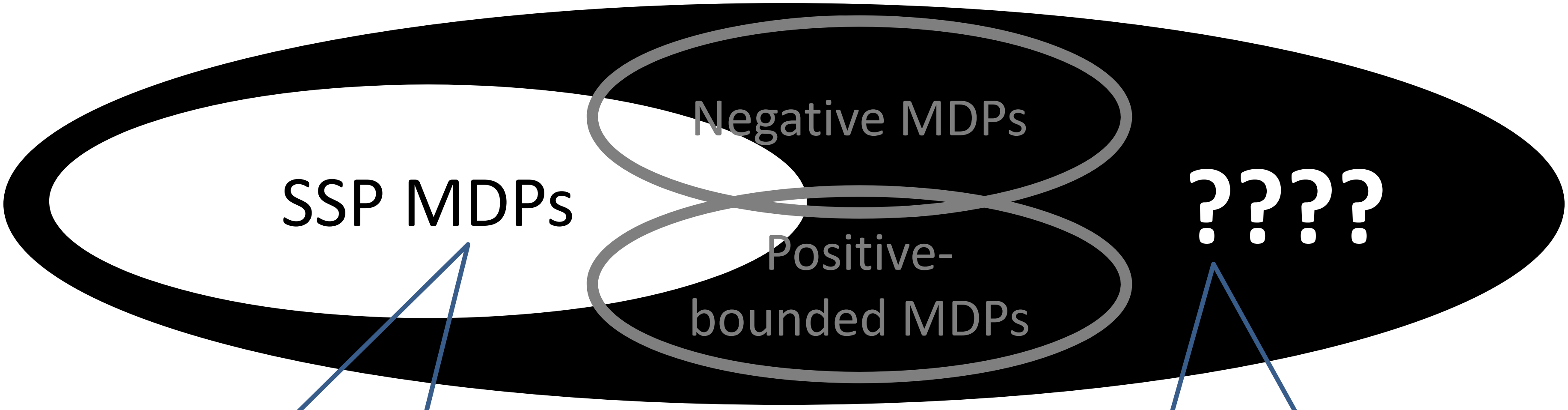
Heuristic Search for Generalized Stochastic Shortest Path MDPs

Andrey Kolobov, Mausam,
Daniel S. Weld, Hector Geffner

Discrete MDP Research So Far

Goal-oriented MDPs (GOMDPs)

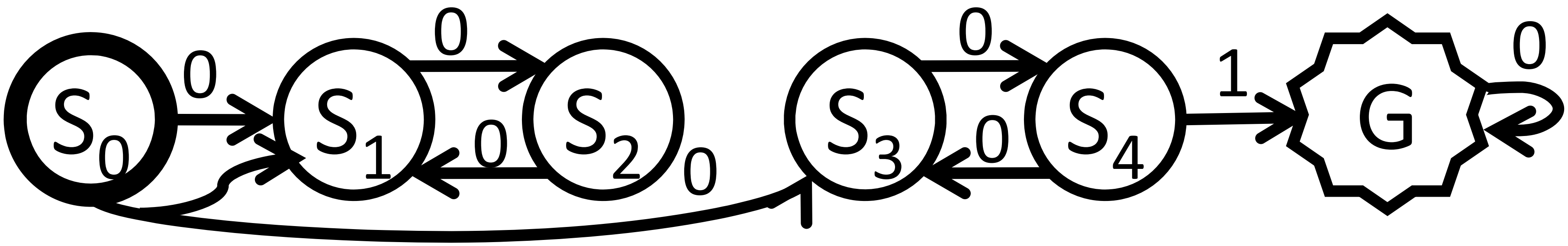


-Model many interesting scenarios
-Efficiently* solvable by heuristic search

-What interesting problems are here?
-How do we solve them efficiently?

Interesting Problems Outside SSP

- MAXPROB – maximize the *probability* of reaching the goal
  - Action rewards are 0 (they are irrelevant)
  - Reaching the goal yields reward = 1
  - Past IPPC problems are of this kind
  - Heuristic search doesn't work on them!



Outline

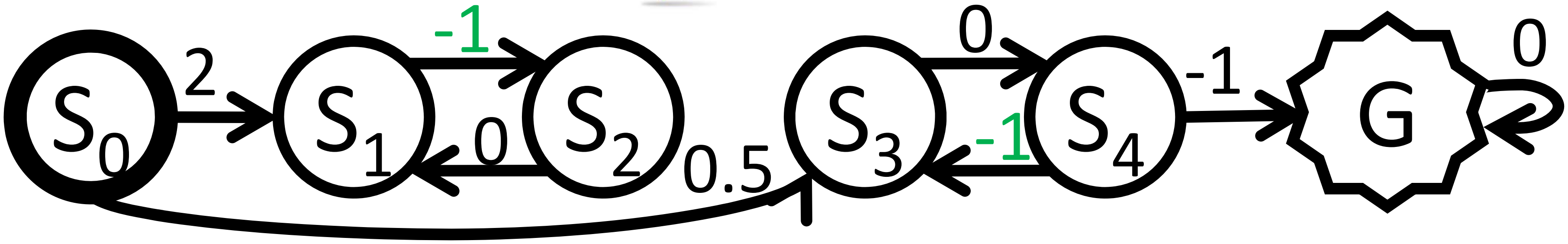
- ✓ Motivation
- ➢ Generalized SSP MDPs – Definition & Examples
- ➢ Heuristic Search for GSSPs: **FRET**
- ➢ Experiments
- ➢ Future Work
- ➢ Q&A

Why Is SSP≠GOMDP?

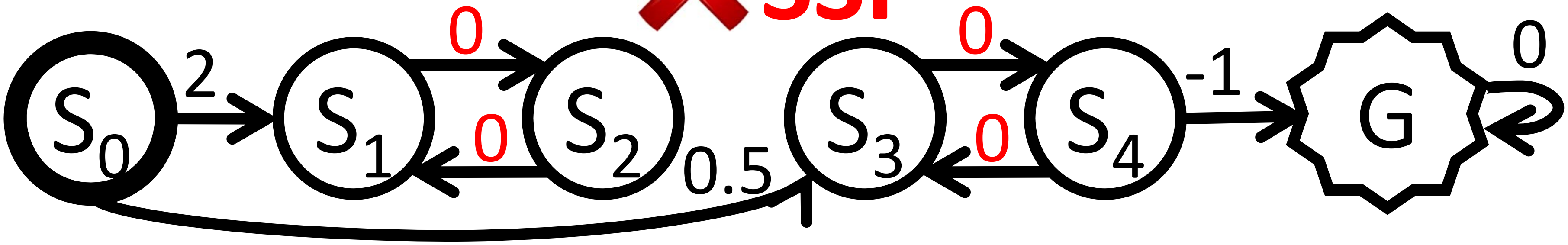
- An MDP M = <S, A, T, R, G, s_0> for which
  - There is a proper policy (reaches the goal with P=1)
  - Every *improper* policy has V(s) = -∞
- Solving an SSP = finding a reward-maximizing (cost-minimizing) policy
- SSP can't contain "free loops"!

Why Is SSP≠ GOMDP: Example

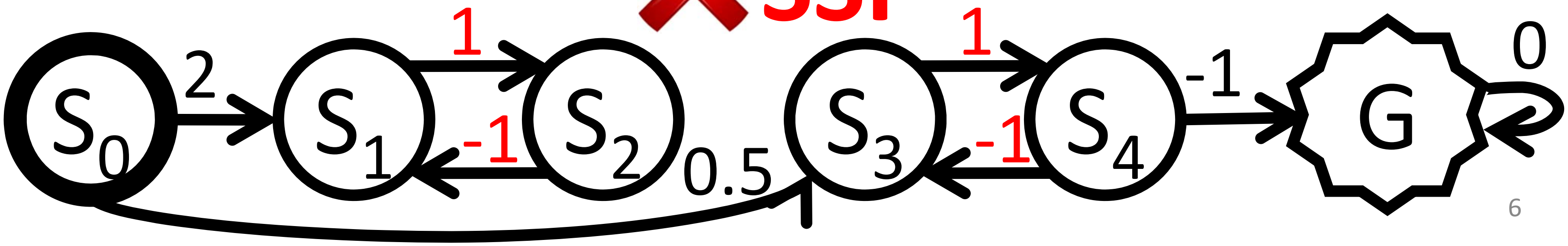
✓ **SSP**



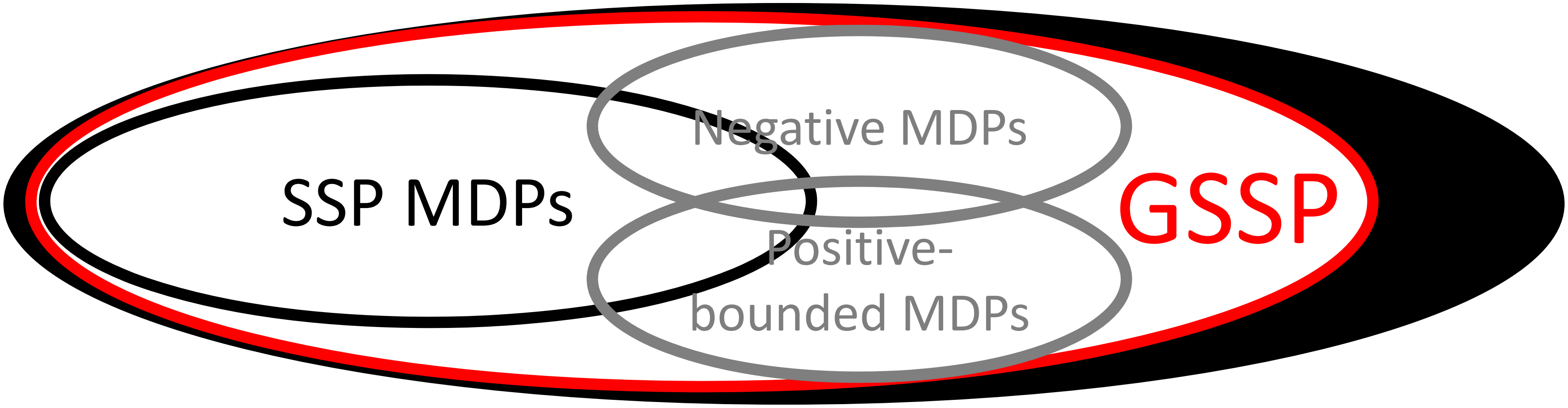
✗ **SSP**



✗ **SSP**



Introducing Generalized SSPs

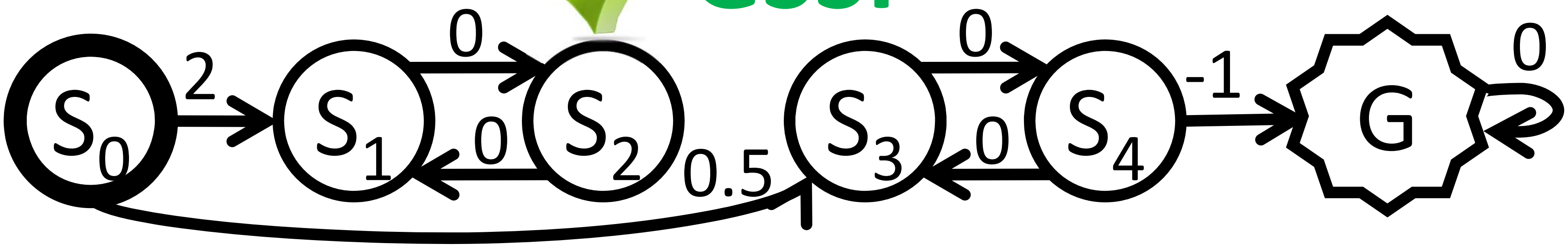


Generalized SSPs: Definition

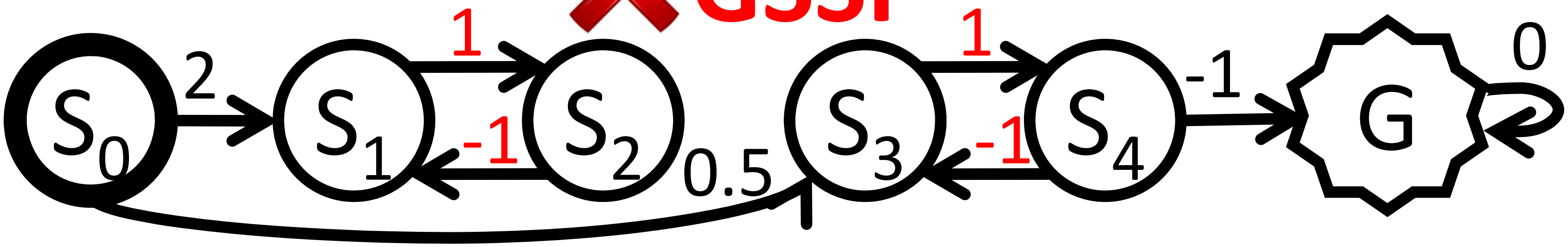
- An MDP M = <S, A, T, R, G, s_0> for which
  - There is a proper policy (reaches the goal with P=1)
  - Sum of *non-negative* rewards accumulated by any policy starting at s_0 is bounded from above
- Solving a GSSP = finding a reward-maximizing Markovian policy *that reaches the goal*

Generalized SSPs: Example

✓ **GSSP**

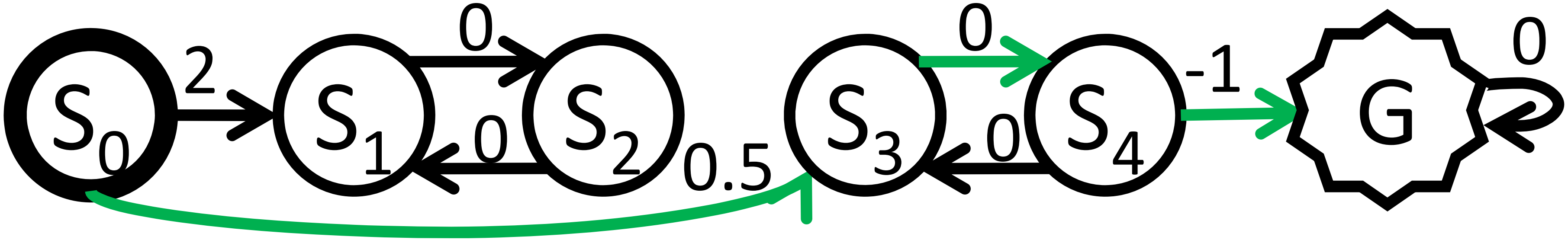


✗ **GSSP**

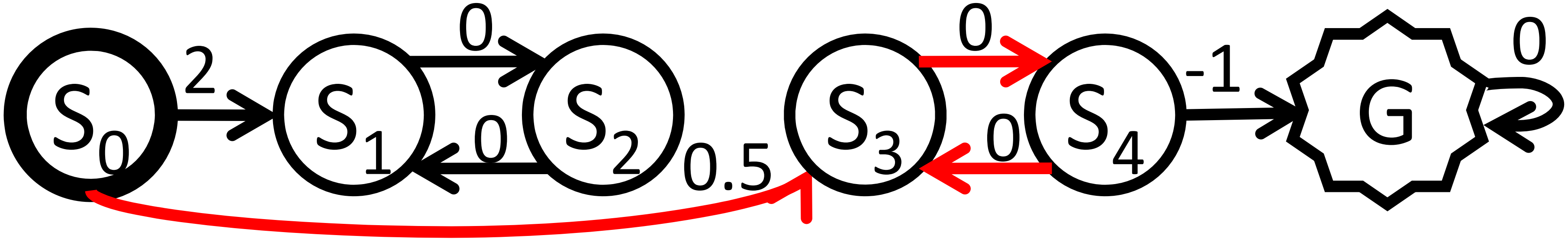


Generalized SSPs: Example

Solution



Not a solution



Outline

- ✓ Motivation
- ✓ Generalized SSP MDPs – Definition & Examples
- ➢ Heuristic Search for GSSPs: **FRET**
- ➢ Experiments
- ➢ Future Work
- ➢ Q&A

Digression: Heuristic Search for SSPs

- Reminder: in SSPs, V* = *B* V*, where
  - *B* is the *Bellman backup operator*
  - *B* V(s) = $\max_a$ {R(s, a) + $\sum_{s' \text{ in } succ(s,a)}$T(s, a, s')V(s')}
- In SSPs, V* is the unique fixed point of *B*
  - I.e., V* = *B* o *B* o … *B* V_0, V_0 is a *heuristic value function*

Digression: Heuristic Search for SSPs

- Find-and-Revise framework (Bonet & Geffner, IJCAI 2003) – LRTDP, LAO*, etc:
  - Start with an *admissible* V_0
  - Iteratively, **find** an unconverged state reachable by the current greedy policy, **revise** its value with *B*
  - Extract the greedy policy from V*

Digression: Heuristic Search for SSPs

- F&R is optimal & resource-efficient. Why?
  - V_0 admissible => $V_0 \geq V^*$ => $V_i \geq V^*$
  - F&R “smartly chooses” states to apply *B* to
  - V* is the unique fixed point of *B*
  - Any V*-greedy policy is optimal

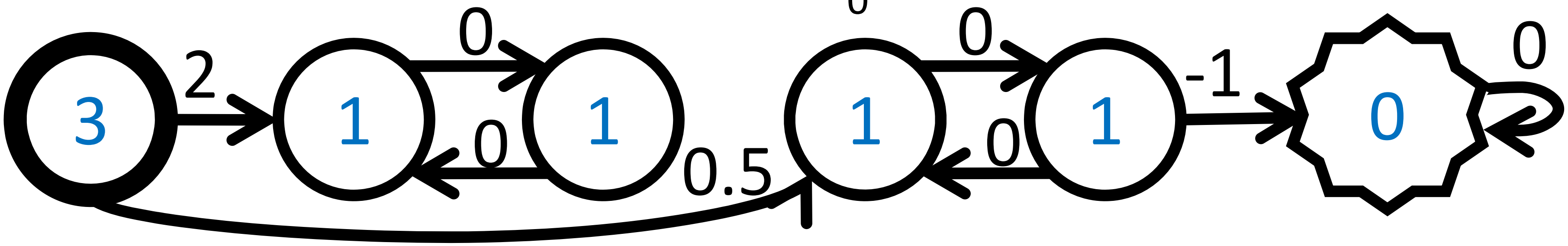
Efficiently Solving GSSPs: Attempt #1

- **Remove “free loops”, solve SSP with F&R**
  - Find loops via transition graph traversal

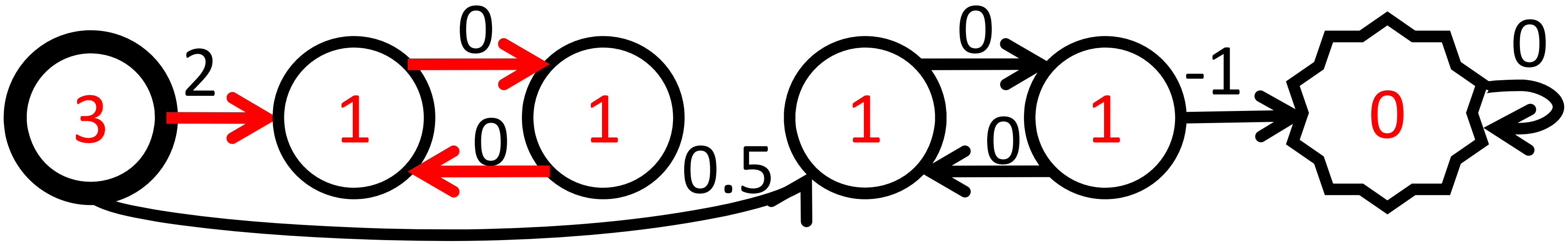
- But… consider a MAXPROB problem
  - The problem “consists” of 0-reward loops
  - Defeats the point of using heuristic search (F&R)

Efficiently Solving GSSPs: Attempt #2

- **Just Run F&R!**
  - Start with an admissible V_0

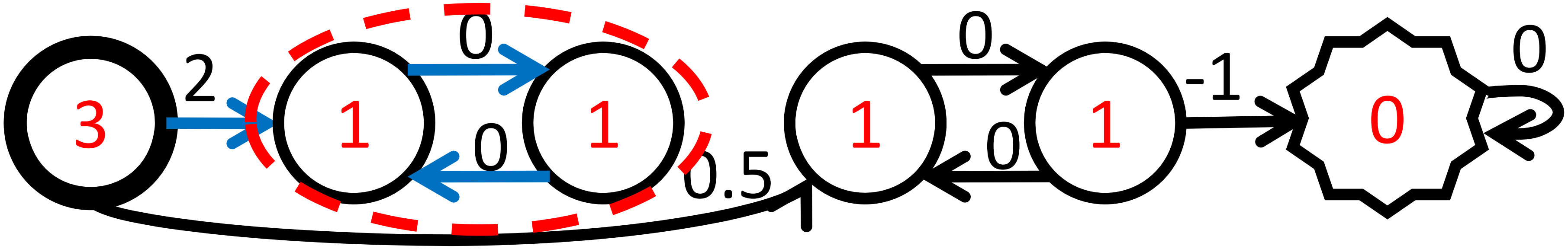


  - Done!



Attempt #2: What Went Wrong?

- **In GSSPs, there are multiple suboptimal admissible fixed points!**
  - When starting with $V_0 \geq V^*$, F&R hit one of them.
  - *B* can't change V over ***traps*** – strongly-connected leaf components in V's greedy transition graph



- SSP-style F&R can yield an arbitrarily poor solution

Efficiently Solving GSSPs: **FRET**

- **F**ind, **R**evise, **E**liminate **T**raps
  - First heuristic search algorithm for MDPs beyond SSP
  - Provably optimal if the heuristic is admissible

- Main idea
  - Run F&R until convergence
  - Eliminate traps in the policy envelope
  - Repeat until no more traps

FRET Example: Finding V*

R
e
p
e
a
t

Start with an admissible V_0

Run **F&R** until convergence

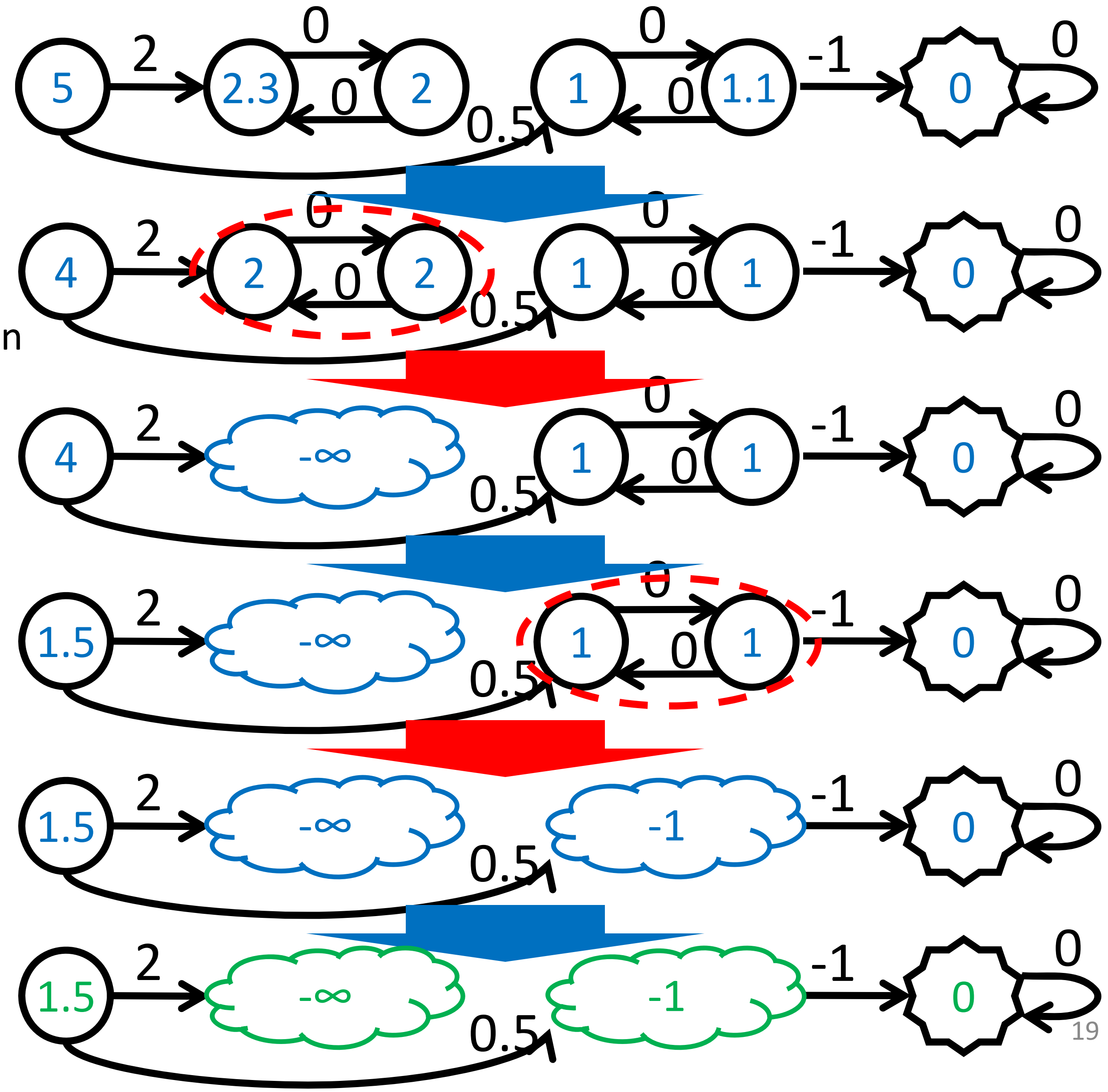
Eliminate Traps in the resulting V_i

Find-and-Revise

Eliminate Traps

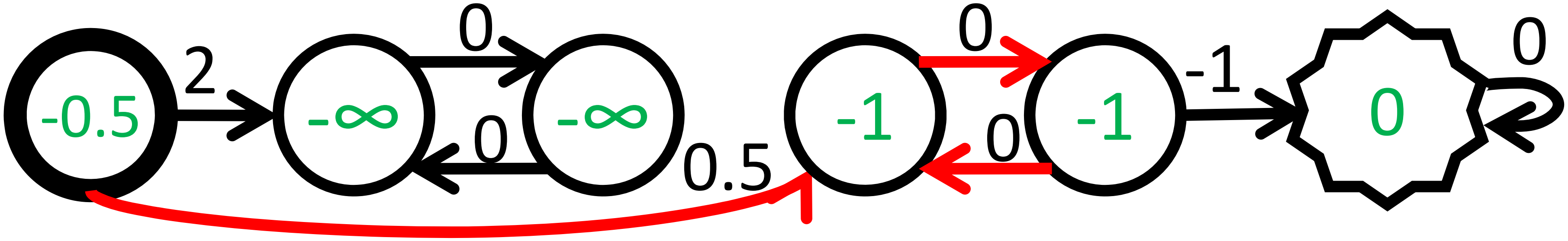
Find-and-Revise

No traps left – done!



FRET Example: Extracting ∏*

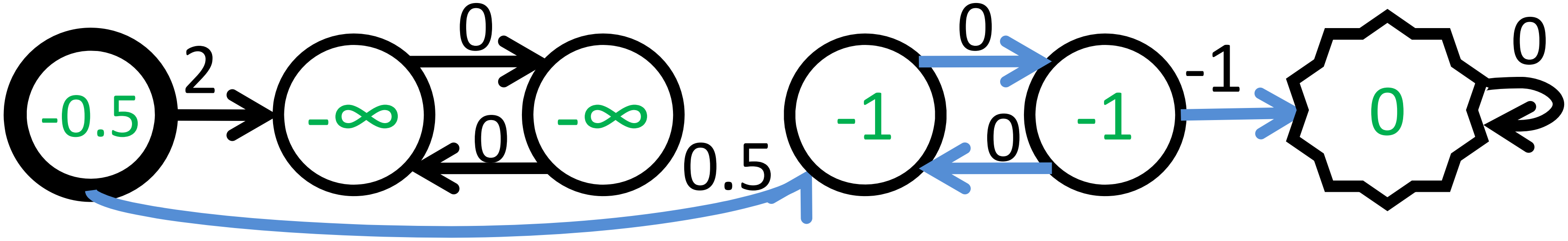
- Greedy attempt:



- **In GSSPs, not every V*-greedy policy is optimal!**

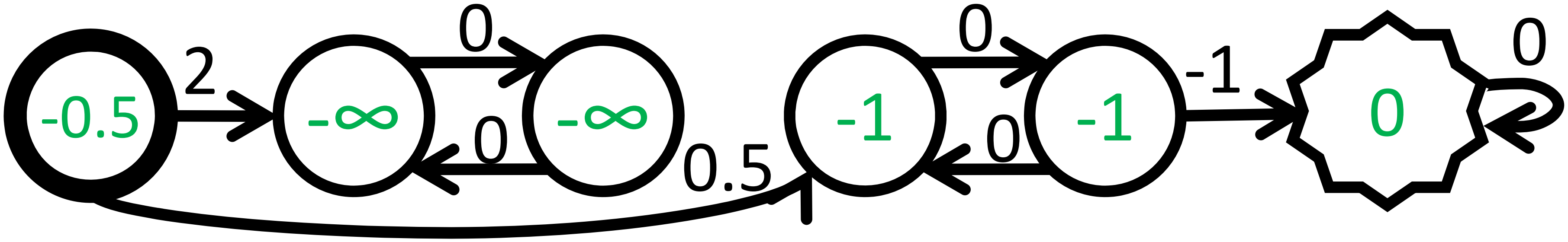
FRET Example: Extracting ∏*

- Iteratively “connect” states to the goals
  - Using optimal actions
  - Until s_0 is connected



Why Does **FRET** It Work?

- **In GSSPs, V* is a fixed point of *B***



- FRET is optimal if the heuristic is admissible

Outline

- ✓ Motivation
- ✓ Generalized SSP MDPs – Definition & Examples
- ✓ Heuristic Search for GSSPs: FRET
- ➢ Experiments
- ➢ Future Work
- ➢ Q&A

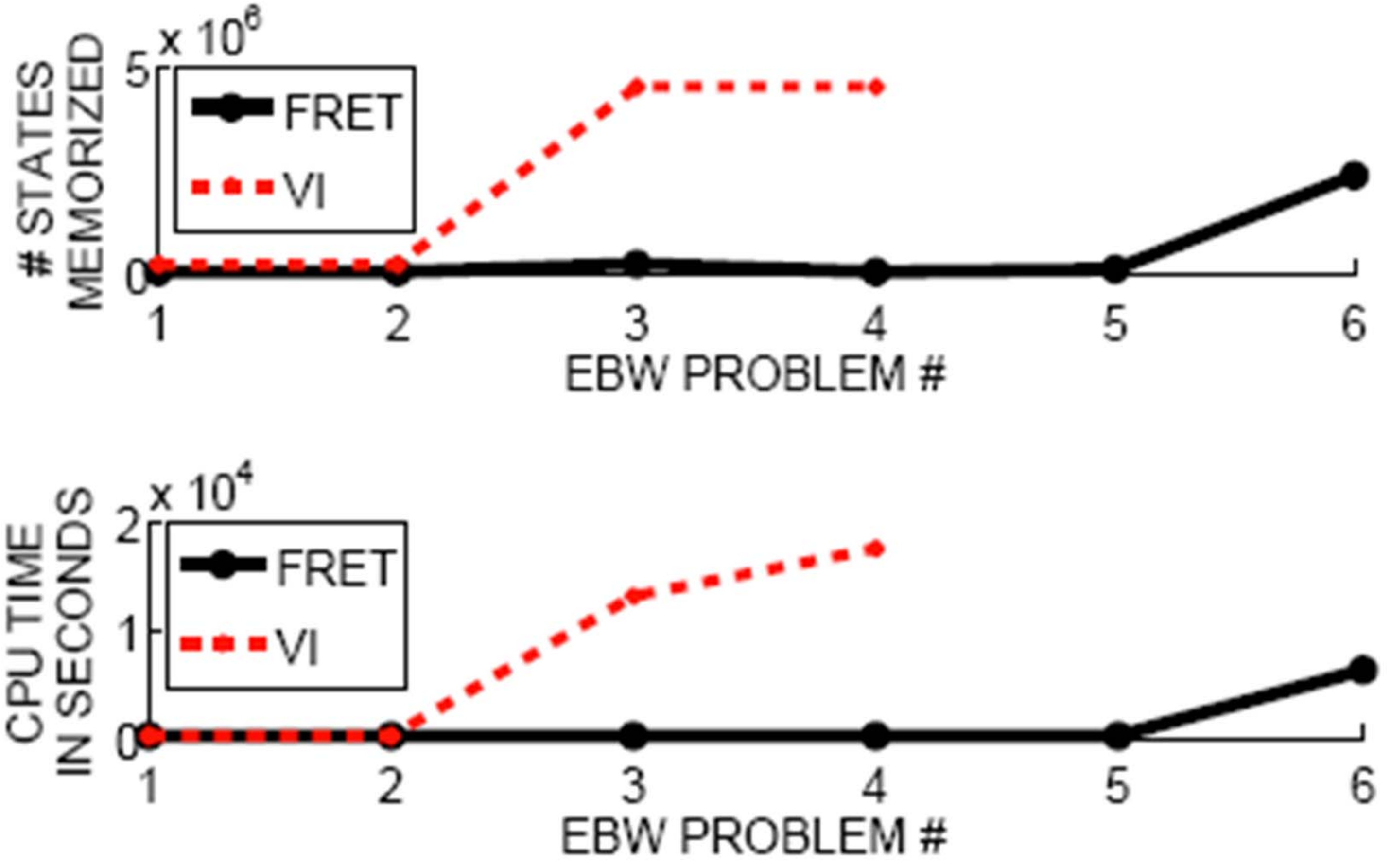
Experimental Setup

- **Problems**: MAXPROB versions of EBW

- **Planners**: VI vs FRET

- **Heuristics**: Zero for VI, One+SixthSense for FRET
  - SixthSense (Kolobov et al., AAAI 2010) soundly identifies some of the “dead ends”; their values are set to 0

Experimental Setup

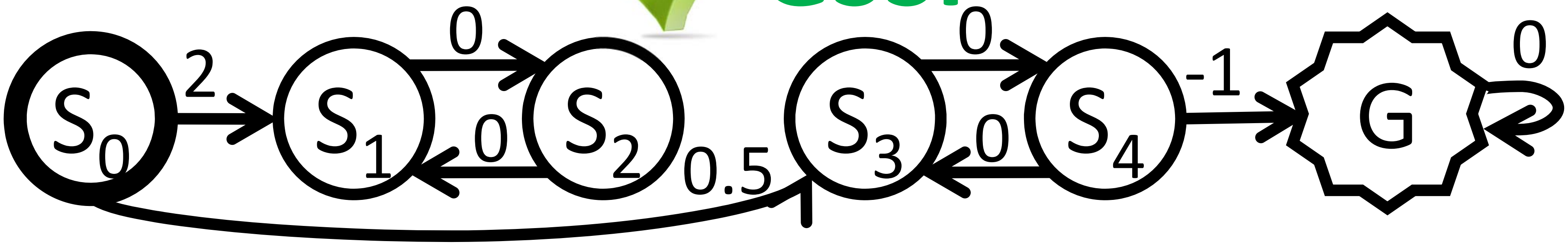


Outline

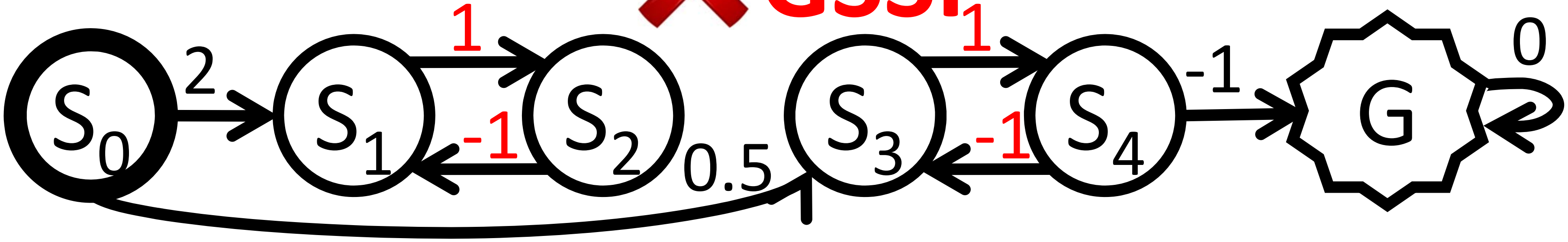
- ✓ Motivation
- ✓ Generalized SSP MDPs -- Definition & Examples
- ✓ Heuristic Search for GSSPs: **FRET**
- ✓ Experiments
- ➢ Future Work
- ➢ Q&A

Future Work

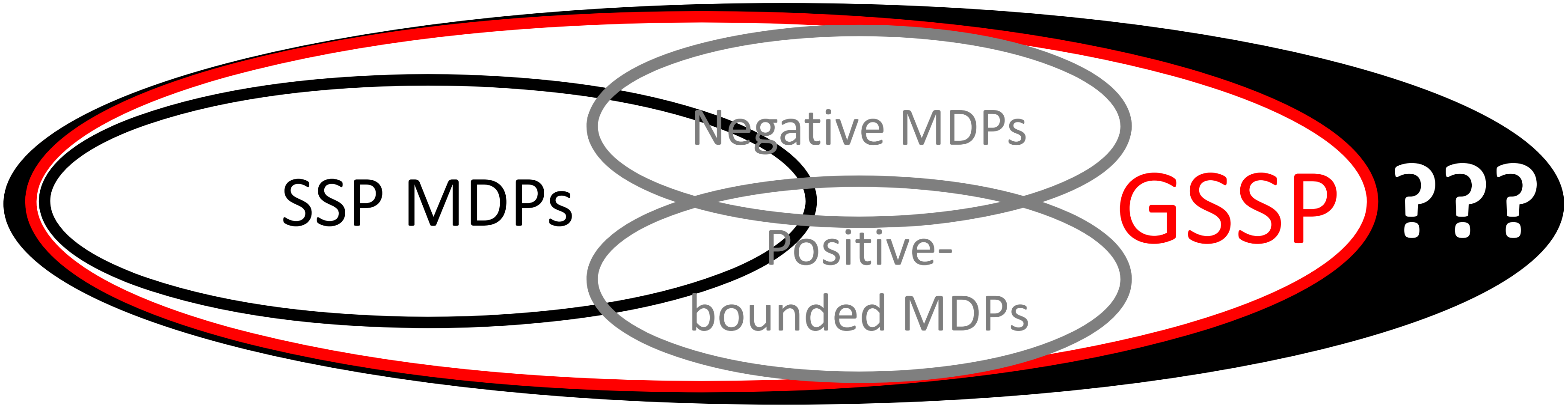
✓ GSSP



✗ GSSP



Future Work



Conclusions

- SSP MDPs exclude interesting planning scenarios
- GSSP contains SSP and several other MDP classes
- SSP heuristic search algorithms fail on GSSPs
- FRET is an optimal heuristic search algorithm for solving GSSPs
- What is beyond GSSPs and how do we solve it?

Questions?