ENGINEERING
HONORS PROGRAM
UNIVERSITY OF MICHIGAN

REGAL Revisited: Reinforcement Learning in Unknown MDPs

Anh Tuan Tran
Advisor: Prof. Ambuj Tewari

Introduction

- Markov Decision Processes (MDPs) describe how an agent interacts with its environment.
- MDPs specify what possible states an agent can be in, what its possible actions are, and what the effects of taking those actions may be.
- Current algorithms have assume that the agent knows all the information it needs about the environment.
- The REGAL algorithm removes this assumption, and has good theoretical performance, but cannot be implemented due to requiring unknown information.

The algorithm constructs a set of possible MDPs, using information the agent has seen so far. It then finds the MDP which the agent will perform the best in, and assumes this MDP will resemble the real environment. The agent then acts under this assumption.

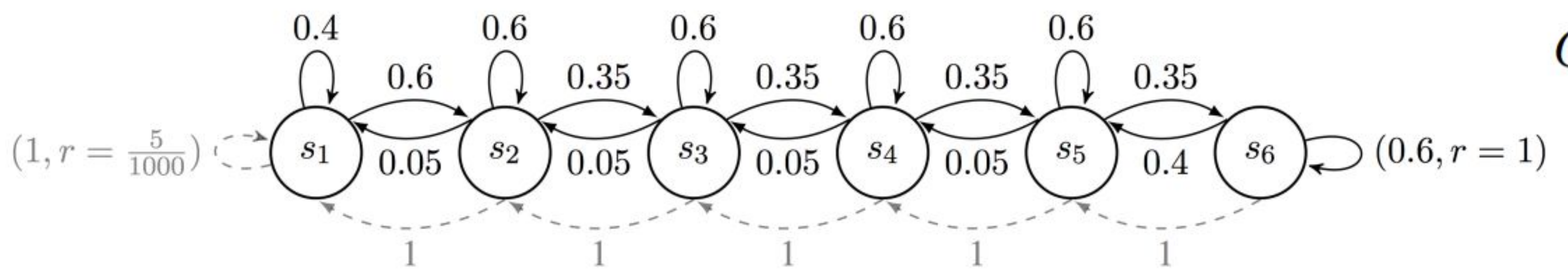
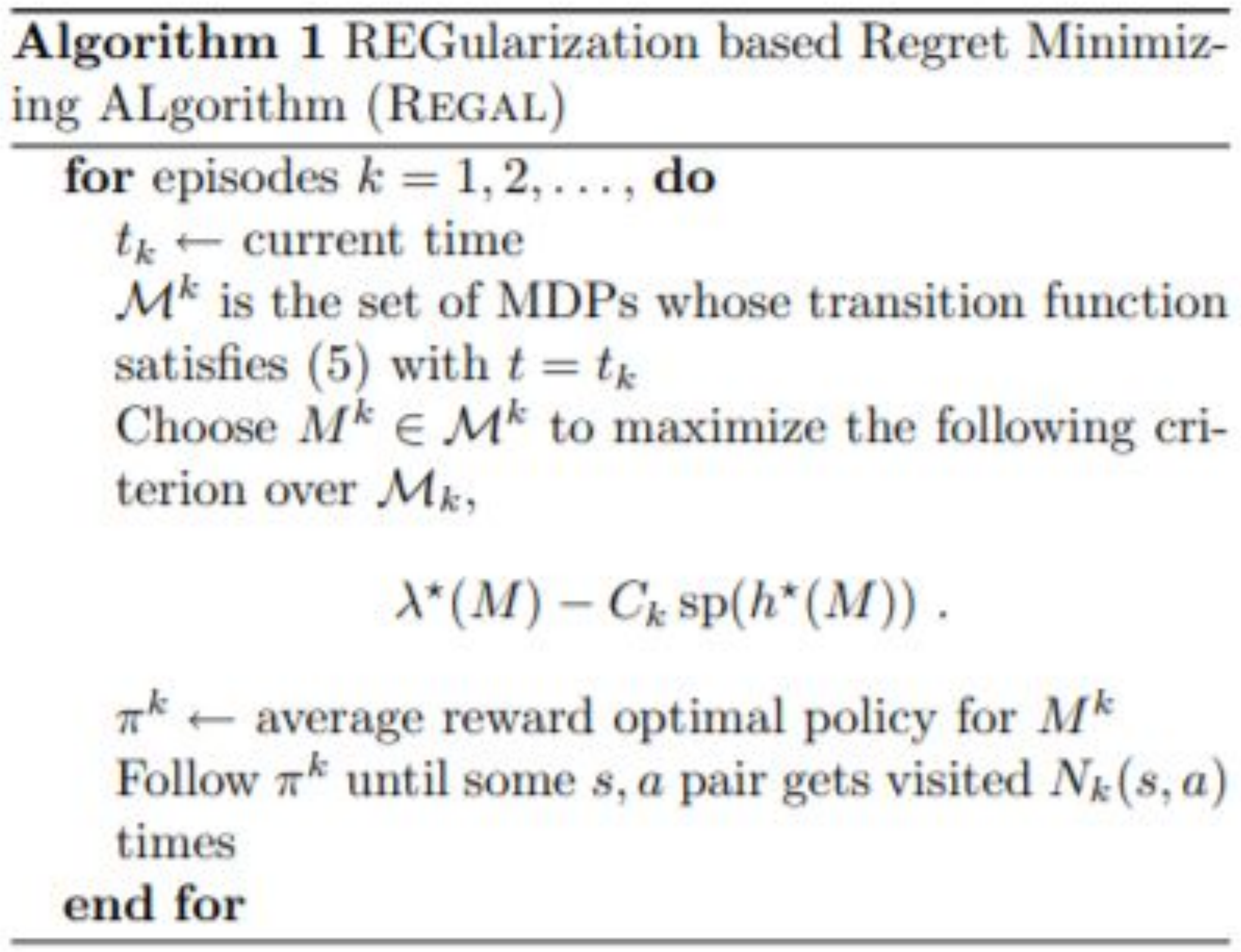


Figure 1: An example MDP called RiverSwim. The agent starts at state s_1. It can choose to go left, which is always successful. Alternatively, it can choose to move right, which has a 0.35 probability of moving it right, 0.6 probability of staying still, and 0.05 probability of moving to the left. The rightmost state s_6 has the highest reward of 1, while the leftmost state has very low reward.

Objectives

- Identify how to eliminate the need for unavailable information
- Develop an alternate algorithm which can be implemented
- Prove its theoretical performance is still preserved
- Implement it and compare its performance to other algorithms

Original REGAL Algorithm



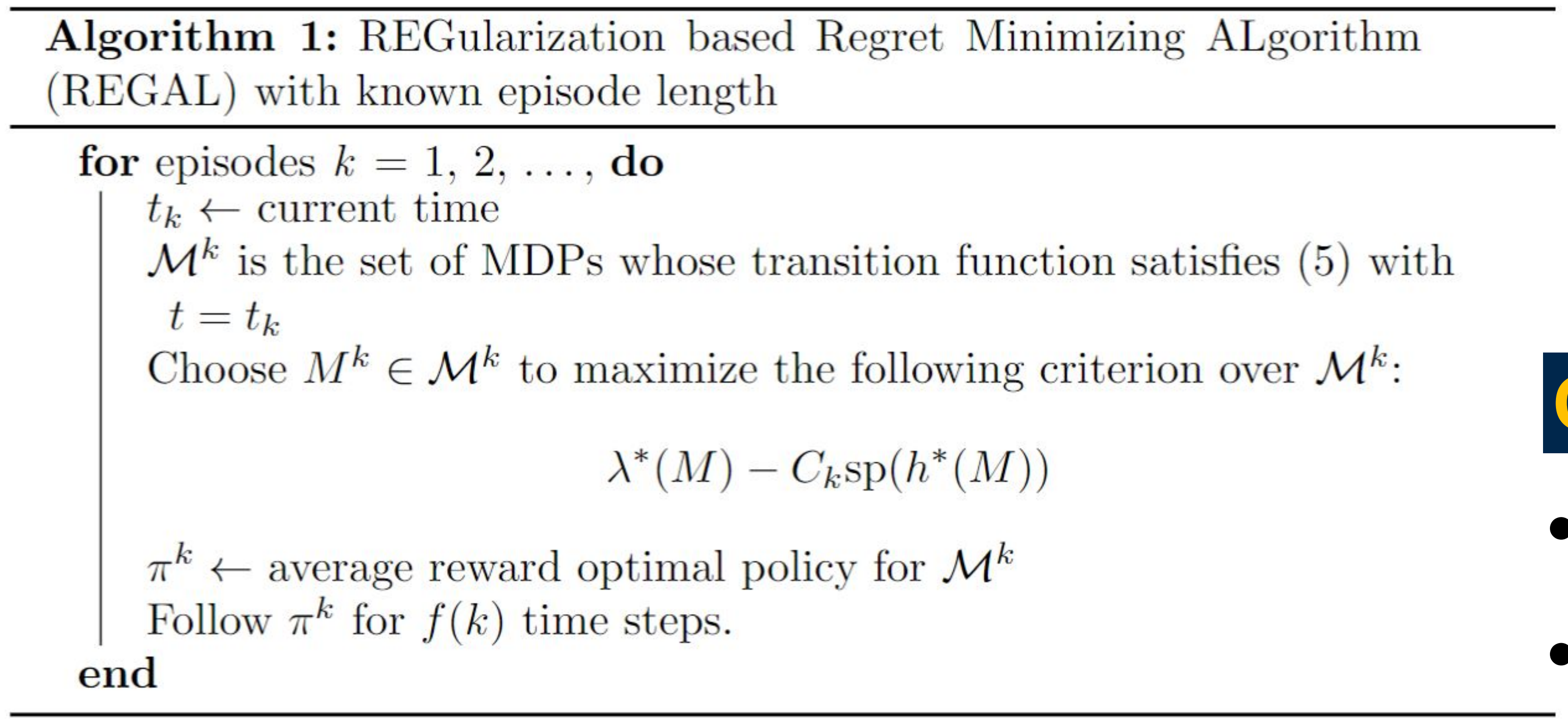
Algorithm 1 REGularization based Regret Minimizing ALgorithm (REGAL)

```
for episodes $k = 1, 2, \dots,$ do
    $t_k \leftarrow$ current time
    $\mathcal{M}^k$ is the set of MDPs whose transition function satisfies (5) with $t = t_k$
    Choose $M^k \in \mathcal{M}^k$ to maximize the following criterion over $\mathcal{M}_k$,
        $\lambda^\star(M) - C_k \,\mathrm{sp}(h^\star(M))$ .
    $\pi^k \leftarrow$ average reward optimal policy for $M^k$
    Follow $\pi^k$ until some $s, a$ pair gets visited $N_k(s,a)$ times
end for
```

$$C_k = \frac{2\sum_{s,a} v_k(s,a)\sqrt{\frac{12S\log\frac{2AT}{\delta}}{N_k(s,a)}} + \sqrt{2\ell_k \log\frac{1}{\delta}}}{\ell_k}$$

The algorithm has to choose C_k, which requires knowledge of $v_k(s,a)$, the number of times each state-action pair will be visited, before the episode begins.

Improved REGAL Algorithm

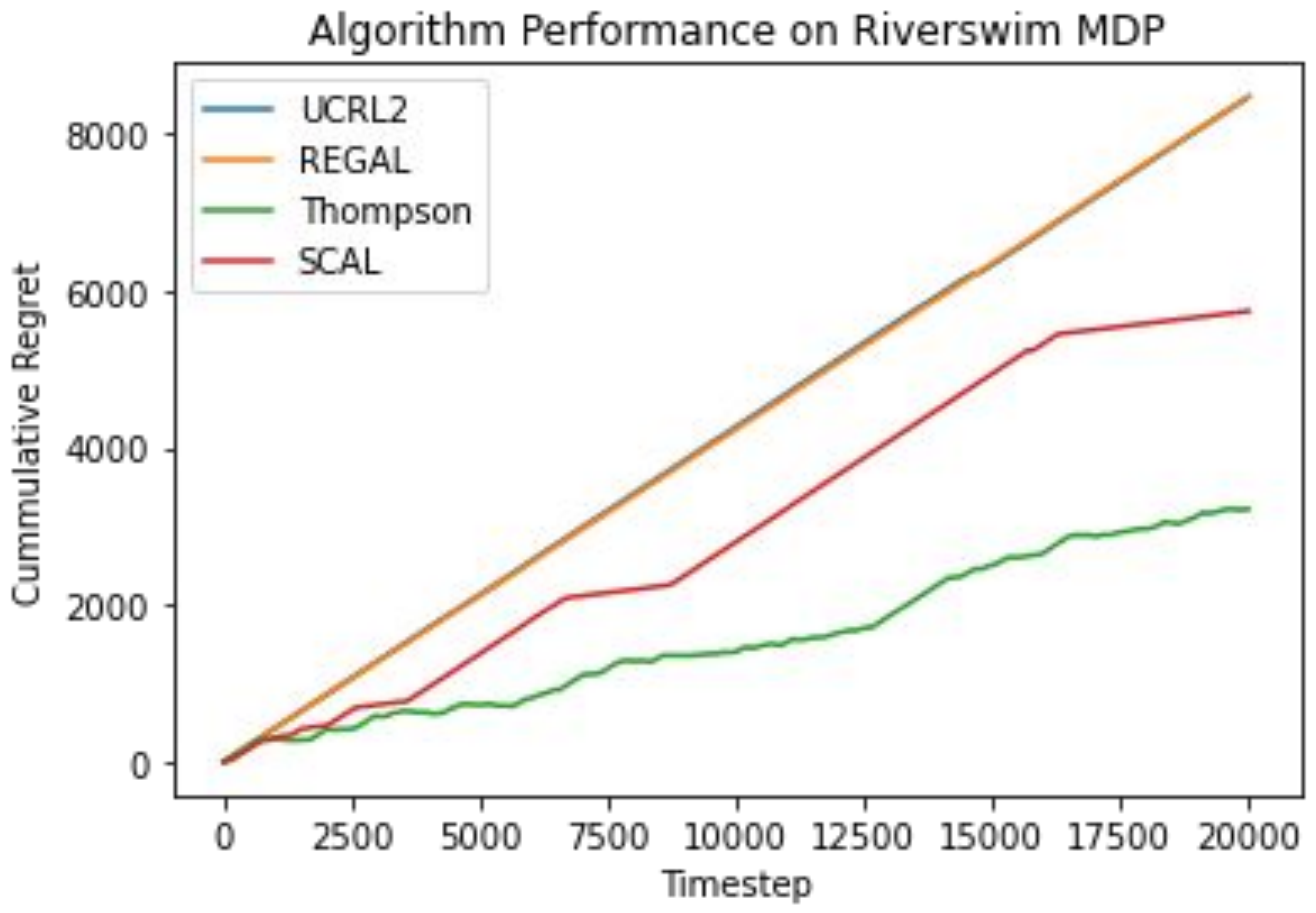


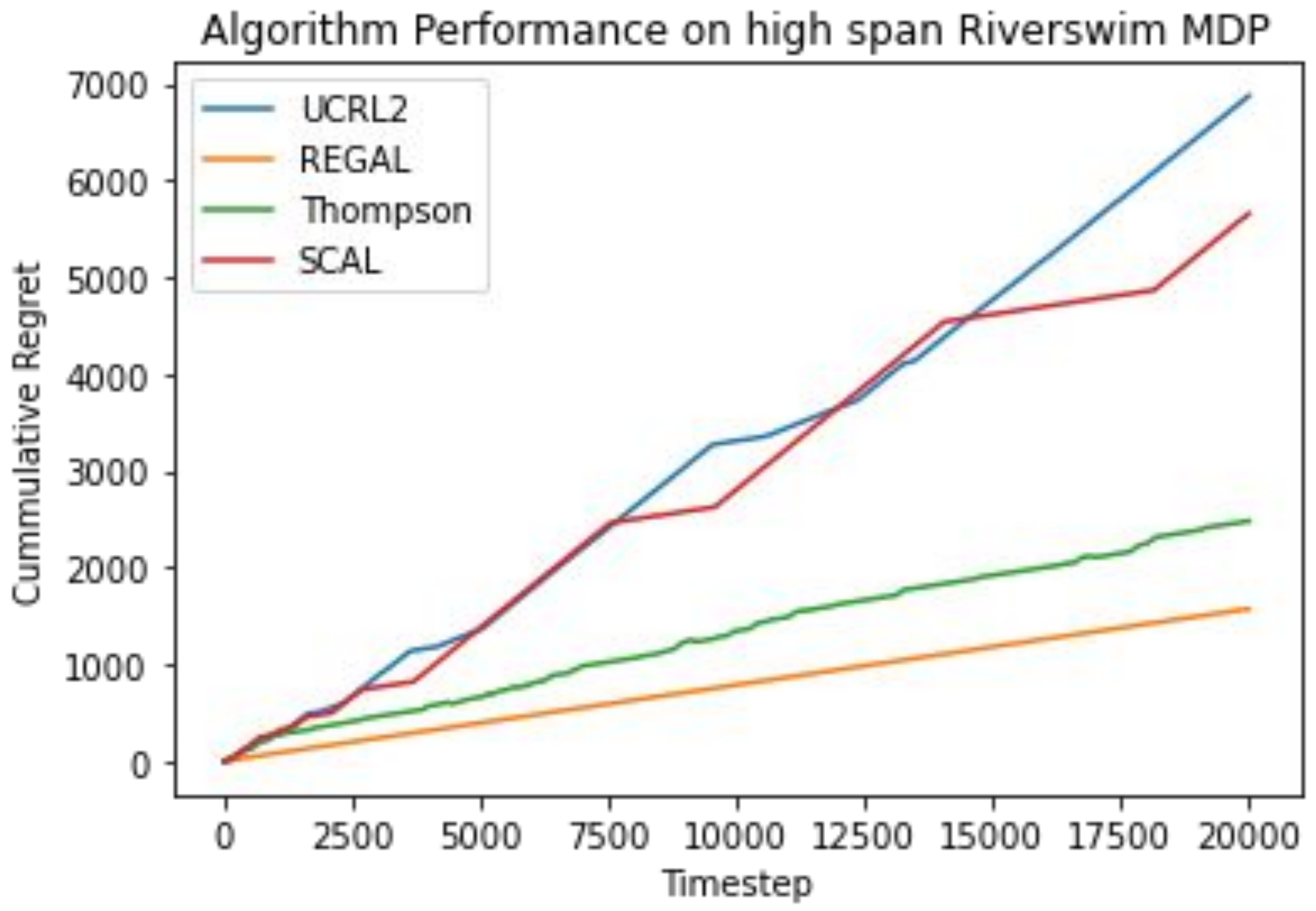
Algorithm 1: REGularization based Regret Minimizing ALgorithm (REGAL) with known episode length

```
for episodes $k = 1, 2, \dots,$ do
    $t_k \leftarrow$ current time
    $\mathcal{M}^k$ is the set of MDPs whose transition function satisfies (5) with $t = t_k$
    Choose $M^k \in \mathcal{M}^k$ to maximize the following criterion over $\mathcal{M}^k$:
        $\lambda^*(M) - C_k \mathrm{sp}(h^*(M))$
    $\pi^k \leftarrow$ average reward optimal policy for $\mathcal{M}^k$
    Follow $\pi^k$ for $f(k)$ time steps.
end
```

$$C_k = \frac{2\sqrt{12S\log(2AT/\delta)}(\sqrt{SAT} + \sqrt[4]{SAT}) + \sqrt{2l_k\log(1/\delta)}}{l_k}$$

Results





Conclusion

- The new algorithm has the same theoretical performance, and is now implementable.
- It performs on par with UCRL2, a popular algorithm for the same problem, however is outperformed by other methods.
- In high span environments, REGAL outperforms the other algorithms.