Multiagent Stochastic Shortest Path Problem

Martin Jonáš , Antonín Kučera , Vojtěch Kůr , Jan Mačák , Vojtěch Řehák
Faculty of Informatics, Masaryk University, Czechia
martin.jonas@mail.muni.cz, tony@fi.muni.cz, vojtech.kur@mail.muni.cz, macak.jan@mail.muni.cz, rehak@fi.muni.cz

Abstract

We introduce and study the multi-agent stochastic shortest path (MSSP) problem, in which k agents strive to reach a target state, aiming to minimize the expected time to reach the target by any agent. We analyze the computational and strategy-complexity of the problem in both autonomous and coordinated settings, and we design efficient strategy-synthesis algorithms. The algorithms are experimentally evaluated on instances of increasing size against natural baselines.

1 Introduction

Stochastic Shortest Path (SSP) is a fundamental optimization problem deeply studied in planning and control. The task is to navigate an agent to a target state in a given explicitly represented Markov decision process (MDP) at the minimum expected path length (the expected path length can also be interpreted as the expected time to reach a target). The SSP problem captures a variety of realistic scenarios, including car navigation, drone flying, game playing, etc.

In this work, we initiate the study of the *Multiagent Stochastic Shortest Path (MSSP)* problem, where k agents strive to reach a target so that the expected time of reaching the target by some agent is minimized. As in the classical SSP, we consider application scenarios where the MDP is known and represented *explicitly* (hence, there is no need to *learn* the MDP structure). As a simple example, consider the problem of urgent blood transport from a blood bank to a distant hospital by a car driving through a city. The current traffic density determines the probability distribution of the time required to cross individual sections of the road and can be modeled by an MDP whose size is proportional to the number of road sections. Hence, planning an optimal route for a *single* car is an instance of the SSP problem, and an optimal moving strategy σ can be quickly computed by existing algorithms. In order to increase the chance of timely delivery, *multiple cars* (from the same or different blood banks) can be dispatched simultaneously. Clearly, the aim is to minimize the expected time of the *first arrival* (the blood doses delivered by the cars arriving later can be stored for later use). A natural approach to solving this optimization problem is to compute the above strategy σ for each car, i.e., to solve the SSP problem independently for each car. However, this is *not* optimal in general, and in some cases it is even the *worst* possible solution. A simple illustrative example is given in Fig. 1.

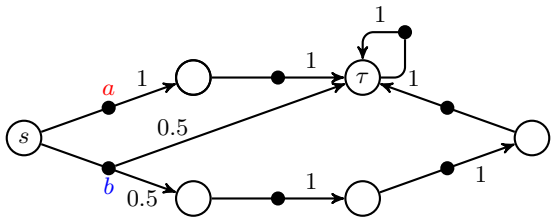

Figure 1: The initial state is s, the target is τ, and the numbers denote transition probabilities. If a single agent chooses the action a (or b), the expected time to reach τ is 2 (or 2.5, resp.). Hence, a single agent should choose a. Now consider two agents. If both of them choose a, the expected time of reaching τ by some agent is 2. However, if one of them chooses a and the other b, the expectation decreases to 1.5. (The b action selects between the paths of length 1 and 4 uniformly at random. In the first case, the "blue" agent arrives to τ after one step. In the second case, the "red" agent arrives to τ already after two steps. Hence, the expectation is $0.5{\cdot}1{+}0.5{\cdot}2 = 1.5$.) Also note that if both agents choose b, the expectation is 1.75.

More generally, MSSP models a class of *time-critical delivery problems* where the item can be replicated and sent across multiple paths to improve time performance and reliability. The primary research question considered in this work is *"How to compute an optimal solution for a given MSSP instance?"* In addition to designing efficient algorithms, we analyze the *computational complexity* of MSSP and determine the *type of strategies* required for optimal solutions.

In the above application scenario, the cars can be autonomous or coordinated by a central control room. When the agents are UAVs operating in a hostile environment, the coordination may be unachievable. Therefore, we consider the MSSP problem for both *autonomous* and *coordinated* agents.

Even if agent coordination is achievable, it can be expensive. Therefore, we also analyze *the benefits of coordination for solving MSSP*, i.e., the ratio between the best solutions achievable in the autonomous and coordinated settings.

Our Contribution. We start by analyzing the computational complexity of the MSSP problem and identifying the type of strategies required for optimal solutions.

For the *Coordinated MSSP* problem, we show that an optimal coordinated strategy for k agents in a given MDP M is computable in time *polynomial* in the size of M and *exponential* in k. For every *fixed* k, the problem is solvable in polynomial time, but the degree of the bounding polynomial grows linearly in k. Furthermore, an optimal strategy can be constructed so that it is *memoryless* (the decision depends only on the current positions of the agents) and *deterministic* (the choice of actions is uniquely determined in every step). We also show that the problem is PSPACE-*hard*, and hence the exponential blowup in k is *unavoidable*. More precisely, we prove it is PSPACE-hard to decide the existence of a coordinated strategy such that the expected time to reach a target by some agent is bounded by a given constant.

For the *Autonomous MSSP* problem, we show that an optimal autonomous strategy profile for k agents (i.e., a k-tuple of independent single-agent strategies) exists, but each agent may require memory to "remember" some information about the history of previous moves. We also prove that *finite-memory randomized* strategy profiles are sufficient for achieving ε-optimal solutions for an arbitrarily small $\varepsilon > 0$. Hence, for practical purposes, it suffices to consider only finite-memory randomized strategy profiles. Then, we show that the problem of deciding the existence of an autonomous strategy profile for k agents such that the expected time to reach a target by some agent is bounded by a given constant is NP-hard *even for* $k = 2$ *and even if the class of admissible profiles is restricted to memoryless profiles*. Hence, the Autonomous MSSP problem is not fixed-parameter tractable in k, and it is also *conceptually* harder (in particular, there is no apparent way of computing an ε-optimal profile for a given $\varepsilon \geq 0$). Our algorithmic solution is based on the following:

(1) We show that for every memory size m, the problem of synthesizing the best randomized profile where every strategy in the profile uses at most m memory states is reducible to the problem of computing the best randomized memoryless profile. Hence, the algorithmic core of the problem is the synthesis of the best memoryless randomized profile (this problem is still NP-hard even for two agents, see above).

(2) In the next step, we show how to express the expected time to reach a target by some agent as a differentiable function of the parameters representing memoryless randomized profiles. This is the crucial step enabling the application of state-of-the-art techniques of differentiable programming, which is the core of our AUTOHIT algorithm. The efficiency of AUTOHIT is bought by the loss of optimality guarantees (this is unavoidable due to the NP-hardness of the problem).

In the *experimental part* of our paper, we evaluate our synthesis algorithms on instances of increasing size against natural baselines, and we compare the efficiency of solutions obtained in the coordinated and autonomous settings. The instances resemble city maps with a grid-like structure with random delays at certain locations. To avoid bias towards specific MDP classes, the instances are generated randomly.

Related Work. The SSP problem was introduced in [Eaton and Zadeh, 1962] in the framework of pursuit-evasion games. The SSP problem is solvable efficiently by the standard MDP strategy synthesis techniques (i.e., value iteration, policy iteration, and linear programming) under some structural conditions [Puterman, 1994; Bertsekas and Tsitsiklis, 1991; de Alfaro, 1999]. The solution algorithms and their complexity depend also on the properties of the reward function (see, e.g., [Baier *et al.*, 2018]). More recent works study SSP under some additional conditions on the probability distribution of the accumulated reward [Piribauer *et al.*, 2022; Haase and Kiefer, 2015; Křetínský and Meggendorfer, 2018; Ahmadi *et al.*, 2021; Mannor and Tsitsiklis, 2011]. See also [Randour *et al.*, 2015] for an overview. The SSP problem in an *unknown* environment has been studied in various variants, and the solution is mostly constructed by appropriate reinforcement learning techniques. A recent overview of these works is given in [Rosenberg, 2022].

Multi-agent problems related to SSP have been studied only recently. In [Nawaz and Ornik, 2023], an algorithm for minimizing the expected time of visiting *all* targets in a given set by a swarm of autonomous agents is proposed. A multi-agent extension of pursuit-evasion games is introduced in [Zhang *et al.*, 2021]. The problem of minimizing the *total time* of reaching all targets by autonomous agents in an *unknown* environment is studied in [Trivedi and Hemachandra, 2023; Chavan *et al.*, 2025]. Recent works also study multi-agent variants of infinite-horizon objectives (such as mean-payoff or steady-state objectives) in explicitly presented MDPs [Jonáš *et al.*, 2025; Díaz-García *et al.*, 2023; Klaška *et al.*, 2023]. Related objectives such as expected average reward have also been studied for decentralized POMDPs (see, e.g., [Jiang *et al.*, 2017]). To the best of our knowledge, there are no previous works on MSSP.

2 Preliminaries

We assume familiarity with basic notions of probability theory and Markov chain theory. We use $\mathbb{N}$ to denote the set of all non-negative integers and $\mathbb{N}_\infty$ to denote the set $\mathbb{N} \cup \{\infty\}$. For a given finite or countably infinite set A, the set of all probability distributions over A is denoted by $Dist(A)$. We say that $\mu \in Dist(A)$ is *Dirac* if $\mu(a) = 1$ for some $a \in A$.

Markov Chains. A *Markov chain* is a pair $C = (S, A)$ where S is a finite or countably infinite set of *states* and $A : S \times S \to [0, 1]$ is a *stochastic matrix* such that $\sum_{t \in S} A(s, t) = 1$ for every $s \in S$.

A *path* in C is a finite or infinite sequence of states. Infinite paths are called *runs*, where a run is also understood as a mapping $\omega : \mathbb{N} \to S$. For each finite path w, we use Run_w to denote the set of all runs starting with w. To every $s \in S$ we associate the probability space $(Run_s, \mathcal{F}_s, Prob_s)$ such that $\mathcal{F}_s$ is the σ-algebra generated by all Run_w where w is a finite path initiated in s, and $Prob_s$ is the unique probability measure satisfying $Prob_s(Run_w) = \prod_{i=0}^{n-1} A(s_i, s_{i+1})$ for all finite paths $w = s_0, \ldots, s_n$ such that $s_0 = s$.

Markov Decision Processes (MDPs). A Markov decision process (MDP) is a triple $M = (S, Act, P)$ where S and Act are non-empty finite sets of *states* and *actions*, and $P : S \times Act \times S \to [0, 1]$ is a *transition function*. We require that, for all $s \in S$ and $a \in Act$, the sum $\sum_{t \in S} P(s, a, t)$ is equal either to 0 or 1. In the latter case, we say that a is *enabled* in s. The set of all actions enabled in s is denoted by

$\mathit{En}(s)$. We require that $\mathit{En}(s) \neq \emptyset$ for all $s \in S$. We say that $t \in S$ is *reachable* from $s \in S$ if there is a finite sequence $s_0, \ldots, s_n$ such that $s_0 = s$, $s_n = t$, and for every $i < n$ there is $a_i \in \mathit{Act}$ such that $P(s_i, a_i, s_{i+1}) > 0$.

Strategies and Strategy Profiles. Let $M = (S, \mathit{Act}, P)$ be an MDP. Consider $k \geq 1$ agents moving among the vertices of M by performing the actions of M. The agents are either *autonomous* or *coordinated*, i.e., each agent acts either independently or coordinates its decisions with the other agents, respectively (the coordination is usually implemented by a central controller instructing the agents).

We start by defining a coordinated strategy for $k \geq 1$ agents. Let Mem be a non-empty set of *memory states* where some information about the history of the agents' previous moves is stored. A *configuration* is a tuple $(s_1, \ldots, s_k, m)$ where $s_1, \ldots, s_k \in S$ are the states currently visited by the agents and $m \in \mathit{Mem}$ is the current memory state. We use Conf to denote the set of all configurations. A *coordinated strategy* for $k \geq 1$ agents is a triple $\sigma = (\mu, \mathit{next}, \mathit{update})$ where $\mu \in \mathit{Mem}$ is an *initial memory state* and

$$\begin{aligned} \mathit{next} &: \mathit{Conf} \to \mathit{Dist}(\mathit{Act}^k), \\ \mathit{update} &: \mathit{Conf} \times \mathit{Act}^k \to \mathit{Dist}(\mathit{Mem}) \end{aligned}$$

are functions defining the next actions taken by the agents and the associated memory update. We require that for all $\gamma \in \mathit{Conf}$ and $\alpha \in \mathit{Act}^k$ such that $\mathit{next}(\gamma)(\alpha) > 0$, we have that the action α_i is enabled in the underlying state s_i of the configuration γ for all $i \in \{1, \ldots, k\}$.

Note that the choice of the next tuple of actions is randomized. The current memory state is updated according to the current configuration and the chosen tuple of actions. We say that σ is *deterministic* if next and update use only Dirac distributions. Furthermore, we say that σ is *finite-memory* if the underlying set Mem is finite, and *memoryless* if Mem is a singleton. Note that if σ is memoryless, then update is trivial and next can be seen as a function $S^k \to \mathit{Dist}(\mathit{Act}^k)$. Slightly abusing our notation, we formally define memoryless strategies as functions $\sigma : S^k \to \mathit{Dist}(\mathit{Act}^k)$, i.e., we avoid using the next and update functions completely.

An *autonomous strategy profile* for k agents is defined as a k-tuple $\pi = (\sigma_1, \ldots, \sigma_k)$ where every σ_i is a coordinated strategy for *one* agent (see above). Since each σ_i controls only the agent i, the behaviour of each agent is completely *autonomous*. We use $\mathit{Conf}_i = S \times \mathit{Mem}_i$ to denote the set of all configurations of agent i, where Mem_i is the set of memory states of σ_i. We say that π is *finite-memory* or *memoryless* if every Mem_i is finite or a singleton, respectively.

Random Variable *MHit*. Let $M = (S, \mathit{Act}, P)$ be an MDP, $k \geq 1$ the number of agents, and let $\iota_i \in S$, $T_i \subseteq S$ be the initial/target states of agent i for every $i \in \{1, \ldots, k\}$.

Let $\sigma = (\mu, \mathit{next}, \mathit{update})$ be a strategy for k agents, and let $\mathit{init} = (\iota_1, \ldots, \iota_k, \mu)$ be the *initial configuration*. We define a Markov chain $M_\sigma = (\mathit{Conf}, A_\sigma)$ where the stochastic matrix A_σ is defined as follows: for all $\gamma, \delta \in \mathit{Conf}$ such that $\gamma = (s_1, \ldots, s_k, n)$ and $\delta = (t_1, \ldots, t_k, m)$, we put

$$A_\sigma(\gamma, \delta) = \sum_{\alpha \in \mathit{Act}^k} \mathit{next}(\gamma)(\alpha) \cdot \mathit{update}(\gamma, \alpha)(m) \cdot \prod_{i=1}^{k} P(s_i, \alpha_i, t_i) \,.$$

Let $\mathit{MHit} : \mathit{Run}_{\mathit{init}} \to \mathbb{N}_\infty$ be a random variable over the runs in M_σ initiated in init such that $\mathit{MHit}(\omega)$ is either the least $j \in \mathbb{N}$ such that $\omega(j)_i \in T_i$ for some $i \in \{1, \ldots, k\}$, or ∞ if there is no such j. We use $\mathbb{E}_\sigma[\mathit{MHit}]$ to denote the expected value of MHit. Hence, if the agents are controlled by σ, the expected number of actions executed before some agent visits a target state is equal to $\mathbb{E}_\sigma[\mathit{MHit}]$.

Now consider an autonomous strategy profile $\pi = (\sigma_1, \ldots, \sigma_k)$ for k agents. For every $i \in \{1, \ldots, k\}$, let μ_i be the initial memory state of σ_i and $\mathit{init}_i = (\iota_i, \mu_i)$ the initial configuration of agent i. Each σ_i determines a Markov chain M_{σ_i} and the probability space over $\mathit{Run}_{\mathit{init}_i}$ in the way described above. Let Prob_π be the probability measure in the product probability space over the set $\mathit{Run}_{\mathit{init}_1} \times \cdots \times \mathit{Run}_{\mathit{init}_k}$ of all *multiruns*. Slightly overloading our notation, we define a random variable MHit over the multiruns such that $\mathit{MHit}(\omega_1, \ldots, \omega_k)$ is either the least $j \in \mathbb{N}$ such that $\omega_i(j) \in T_i \times \mathit{Mem}_i$ for some $i \in \{1, \ldots, k\}$, or ∞ if there is no such j. We use $\mathbb{E}_\pi[\mathit{MHit}]$ to denote the expected value of MHit. Hence, if the agents move according to the profile π, then $\mathbb{E}_\pi[\mathit{MHit}]$ is the expected number of actions executed before some agent visits a target state.

Multiagent Stochastic Shortest Path (MSSP) Problem. In this work, we study the *Coordinated MSSP* and *Autonomous MSSP* problems. In both cases, a problem instance consists of

- an MDP $M = (S, \mathit{Act}, P)$,
- an integer $k \geq 1$ (the number of agents),
- initial/target states $\iota_i \in S$, $T_i \subseteq S$ such that $\iota_i \notin T_i \neq \emptyset$ for every $i \in \{1, \ldots, k\}$.

The task is to construct a coordinated strategy σ (in the case of Coordinated MSSP) or an autonomous strategy profile π (in the case of Autonomous MSSP) for k agents minimizing the expected value of MHit.

For a given $\varepsilon \geq 0$, we say that σ^* and π^* are *ε-optimal* if $\mathbb{E}_{\sigma^*}[\mathit{MHit}]$ and $\mathbb{E}_{\pi^*}[\mathit{MHit}]$ are ε-close to $\inf_\sigma \mathbb{E}_\sigma[\mathit{MHit}]$ and $\inf_\pi \mathbb{E}_\pi[\mathit{MHit}]$, where σ and π range over all coordinated strategies and autonomous strategy profiles for k agents, respectively. 0-optimal σ^* and π^* are called *optimal*.

Price of Autonomy. For every MSSP instance, the *price of autonomy* is defined as the ratio of $\inf_\pi \mathbb{E}_\pi[\mathit{MHit}]$ to $\inf_\sigma \mathbb{E}_\sigma[\mathit{MHit}]$. Hence, the price of autonomy measures the benefits of coordination for a given MSSP instance, where a higher value means higher benefits.

3 The Complexity of MSSP

In this section, we analyze the computational/strategy complexity of the Coordinated and Autonomous MSSP problems.

Single-Agent Case. We begin by recalling known results about the single-agent case, i.e., the SSP problem (see, e.g., [Puterman, 1994]).

Let $M = (S, \mathit{Act}, P)$ be an MDP, $k = 1$ the number of agents, $\iota \in S$ an initial state and $T \subseteq S$ a set of target states such that $\iota \notin T \neq \emptyset$. We have the following:

$$\begin{aligned} \textbf{maximize} \quad & \sum_{s \in S_{\mathcal{R}}} x_s \\ \text{subject to} \quad & x_s = 0, && s \in T, \\ & x_s \leq 1 + \sum_{t \in S_{\mathcal{R}}} P(s,a,t) \cdot x_t, && s \in S_{\mathcal{R}} \smallsetminus T, \\ & && a \in \mathit{En}_{\mathcal{R}}(s) \end{aligned}$$

Figure 2: A linear program for computing an optimal strategy for a single agent.

$$\begin{aligned} & y_s = 0 && s \in T, \\ & y_s = 1 + \sum_{a \in \mathit{En}(s)} \left(\sigma(s)(a) \cdot \sum_{t \in S_\sigma} P(s,a,t) \cdot y_t \right) && s \in S_\sigma \smallsetminus T \end{aligned}$$

Figure 3: A system of linear equations evaluating a given memoryless strategy σ for one agent.

(a) There exists an optimal memoryless deterministic strategy for the agent computable in polynomial time by the LP of Fig. 2.

(b) For a given memoryless strategy σ for one agent, we have that $\mathbb{E}_\sigma[\mathit{MHit}]$ is computable in polynomial time by solving the system of linear equations of Fig. 3.

More precisely, let $S_{\mathcal{R}}$ be the maximal $C \subseteq S$ satisfying the following conditions:

- if $s \in C$, then there is $t \in T$ reachable from s,
- if $s \in C \smallsetminus T$, then there is $a \in \mathit{En}(s)$ such that $t \in C$ for all $t \in S$ where $P(s,a,t) > 0$.

Furthermore, for every $s \in S_{\mathcal{R}}$, we use $\mathit{En}_{\mathcal{R}}(s)$ to denote the set of all $a \in \mathit{En}(s)$ such that $P(s,a,t) > 0$ implies $t \in S_{\mathcal{R}}$. If $\iota \notin S_{\mathcal{R}}$, then the expected time to reach a target state from the initial state is infinite for every strategy σ. Otherwise, let x^*_s, $s \in S_{\mathcal{R}}$ be the solution to the LP of Fig. 2, and let σ^* be a memoryless deterministic strategy where $\sigma^*(s)(a) = 1$ for an action $a \in \mathit{En}_{\mathcal{R}}(s)$ such that $x^*_s = 1 + \sum_{t \in S_{\mathcal{R}}} P(s,a,t) \cdot x^*_t$. Then σ^* is optimal and $\mathbb{E}_{\sigma^*}[\mathit{MHit}] = x^*_\iota$.

The set S_σ used in the system of Fig. 3 is the maximal $C \subseteq S$ such that

- if $s \in C$, then a state of T can be reached from s with positive probability in the Markov chain M_σ,
- if $s \in C \smallsetminus T$, $\sigma(s)(a) > 0$, and $P(s,a,t) > 0$, then $t \in C$.

If $\iota \notin S_\sigma$, then $\mathbb{E}_\sigma[\mathit{MHit}] = \infty$. Otherwise, $\mathbb{E}_\sigma[\mathit{MHit}] = y^*_\iota$, where y^*_s, $s \in S_\sigma$ is the unique solution of the system of linear equations of Fig. 3.

Coordinated MSSP Problem. The problem of constructing an optimal coordinated strategy for $k \geq 1$ agents is easily reducible to the SSP problem discussed above. For an MSSP instance $M = (S, \mathit{Act}, P)$, k, ι_i, T_i where $i \in \{1, \ldots, k\}$, we construct an SSP instance consisting of an MDP $M' = (S^k, \mathit{Act}^k, P')$ where

$$P'((s_1, \ldots, s_k), (a_1, \ldots, a_k), (t_1, \ldots, t_k)) = \prod_{i=1}^{k} P(s_i, a_i, t_i),$$

the initial state is $(\iota_1, \ldots, \iota_k)$, and the target states are all $(t_1, \ldots, t_k) \in S^k$ such that $t_i \in T_i$ for some $i \in \{1, \ldots, k\}$. There is a natural one-to-one correspondence between memoryless coordinated strategies for k agents in M and memoryless strategies for one agent in M', and this correspondence preserves the expected value of *MHit*. Hence, an optimal coordinated strategy for k agents in M corresponds to an optimal strategy for one agent in M' computable by the LP of Fig. 2. Observe that the size of this LP is polynomial in the size of M and exponential in k. We obtain the following:

Theorem 1. *For every MSSP instance, there exists an optimal memoryless deterministic coordinated strategy computable in time polynomial in the size of the MDP M and exponential in the number of agents k. For every* fixed *k, the strategy is computable in polynomial time.*

In the following, the above-described algorithm for computing an optimal memoryless deterministic strategy for a given MSSP instance is referred to as CoorHit.

Now we show that the exponential blowup in k is unavoidable (assuming P $\neq$ PSPACE), because the Coordinated MSSP problem is PSPACE-hard. More precisely, we have the following:

Theorem 2. *Let M, k, ι_i, T_i where $1 \leq i \leq k$ be an MSSP instance and B a rational bound. The problem of whether there exists a coordinated strategy σ for k agents such that $\mathbb{E}_\sigma[\mathit{MHit}] \leq B$ is* PSPACE*-hard.*

A proof of Theorem 2 is non-trivial and can be found in [Jonáš *et al.*, 2026].

Autonomous MSSP Problem. We start by observing that optimal autonomous profiles exist and may require strategies with memory of arbitrarily large size.

The class of all autonomous strategy profiles is denoted by Π. Furthermore, for every $n \geq 1$, we use Π_n to denote the class of all profiles $\pi = (\sigma_1, \ldots, \sigma_k)$ such that the underlying Mem_i of every σ_i contains at most n memory states. In particular, Π_1 is the class of all memoryless profiles. Recall that π^* is *optimal* if $\mathbb{E}_{\pi^*}[\mathit{MHit}] = \inf_{\pi \in \Pi} \mathbb{E}_\pi[\mathit{MHit}]$. Furthermore, we say that π^* is *Π_n-optimal* if $\pi^* \in \Pi_n$ and $\mathbb{E}_{\pi^*}[\mathit{MHit}] = \inf_{\pi \in \Pi_n} \mathbb{E}_\pi[\mathit{MHit}]$. Note that the existence of an optimal and a Π_n-optimal profile is not immediately clear. It is established in the next theorem.

Theorem 3. *For every MSSP instance, there exist an optimal profile and a Π_n-optimal profile for every $n \geq 1$.*

The existence of an optimal profile is proven by repeatedly selecting converging subsequences of distributions occurring in an infinite sequence of profiles $\pi_1, \pi_2, \ldots$ such that $\lim_{i \to \infty} \mathbb{E}_{\pi_i}[\mathit{MHit}] = \inf_{\pi' \in \Pi} \mathbb{E}_{\pi'}[\mathit{MHit}]$. A similar technique is used to prove the existence of a Π_n-optimal profile. The details are in [Jonáš *et al.*, 2026].

Now we show that optimal autonomous profiles may require strategies with memory of arbitrarily large size. However, for every $\varepsilon > 0$, there exists an ε-optimal profile with finite memory.

Theorem 4. *For every $n \geq 1$, there exist an instance of the Autonomous MSSP problem with two agents and a profile*

$\pi \in \Pi_{n+1}$ *such that* $\mathbb{E}_\pi[\mathit{MHit}] < \inf_{\pi' \in \Pi_n} \mathbb{E}_{\pi'}[\mathit{MHit}]$. *Furthermore, for every instance of the Autonomous MSSP problem with k agents and every $\varepsilon > 0$, there exists an ε-optimal finite-memory profile.*

Furthermore, in [Jonáš *et al.*, 2026] we show that randomized profiles are *strictly more powerful* than deterministic profiles when the agents use memory of a given size.

Now we analyze the computational complexity of Autonomous MSSP. The next theorem says that the problem of computing a profile $\pi \in \Pi_n$ for an MDP M minimizing $\mathbb{E}_\pi[\mathit{MHit}]$ can be efficiently reduced to the problem of computing a *memoryless* profile $\bar{\pi}$ minimizing $\mathbb{E}_{\bar{\pi}}[\mathit{MHit}]$ in an effectively constructible MDP $\bar{M}$.

Theorem 5. *Let $M = (S, \mathit{Act}, P)$ be an MDP, $k \geq 1$, and Mem a set of memory states of size $n \geq 1$. Then there is an MDP $\bar{M} = (S \times \mathit{Mem}, \mathit{Act} \times \mathit{Mem}, \bar{P})$ such that*

- *for every profile $\pi = (\sigma_1, \ldots, \sigma_k)$ for M where every σ_i is a strategy with memory Mem, there exists a memoryless profile $\bar{\pi} = (\bar{\sigma}_1, \ldots, \bar{\sigma}_k)$ for $\bar{M}$ such that the Markov chains M_{σ_i} and $\bar{M}_{\bar{\sigma}_i}$ are identical for every $i \in \{1, \ldots, k\}$,*
- *for every memoryless profile $\bar{\pi} = (\bar{\sigma}_1, \ldots, \bar{\sigma}_k)$ for $\bar{M}$, there exists a profile $\pi = (\sigma_1, \ldots, \sigma_k)$ for M where every σ_i is a strategy with memory Mem such that the Markov chains $\bar{M}_{\bar{\sigma}_i}$ and M_{σ_i} are identical for every $i \in \{1, \ldots, k\}$.*

Intuitively, the MDP $\bar{M}$ is obtained from M by encoding the elements of Mem into the states of $\bar{M}$. Hence, instead of synthesizing a finite-memory profile for M, we may synthesize a memoryless profile $\bar{\pi}$ for $\bar{M}$ and then "translate" $\bar{\pi}$ into a finite-memory profile for M so that $\mathbb{E}_{\bar{\pi}}[\mathit{MHit}]$ in $\bar{M}$ is equal to $\mathbb{E}_\pi[\mathit{MHit}]$ in M. In other words, the algorithmic core of the MSSP problem is the *synthesis of memoryless profiles*. Unfortunately, the synthesis problem is NP-hard even for two agents and memoryless profiles.

Theorem 6. *Let M be an MDP and ι_i, T_i where $i \in \{1, 2\}$ initial/target states for two agents. Let B be a rational bound. The problem of whether there exists an autonomous strategy profile π for two agents such that $\mathbb{E}_\pi[\mathit{MHit}] \leq B$ is NP-hard. The problem is NP-hard even if the class of admissible profiles is restricted to memoryless profiles.*

A proof is obtained by a non-trivial reduction from a variant of the Boolean satisfiability problem; see [Jonáš *et al.*, 2026]. Finally, we show that the price of autonomy can be arbitrarily large, even for two agents.

Theorem 7. *For every $B \in \mathbb{N}$, there exists an MSSP instance with two agents such that the price of autonomy is at least B.*

4 Minimizing $\mathbb{E}_\pi[\mathit{MHit}]$

The results of the previous section show that the problem of computing an optimal coordinated strategy for a given MSSP instance is fixed-parameter tractable in the number of agents k. However, there is no straightforward algorithmic solution for the Autonomous MSSP. In this section, we address this challenge by designing AUTOHIT, an efficient profile synthesis algorithm for the Autonomous MSSP. We begin by summarizing the principal limitations.

Due to Theorems 4 and 5, we can safely restrict our attention to the synthesis of *memoryless* profiles. Since synthesizing optimal memoryless profiles is NP-hard even for two agents (see Theorem 6), it cannot be done efficiently unless P = NP. Hence, *efficiency inevitably leads to losing optimality guarantees*. Furthermore, the synthesis should not be limited to deterministic profiles because randomized profiles are strictly more powerful (see the remarks after Theorem 4). On the other hand, the synthesis may take advantage of the explicit MDP representation by analyzing and utilizing relevant structural properties of M. The design of AUTOHIT reflects these observations.

We start by designing an efficient *evaluation procedure* computing $\mathbb{E}_\pi[\mathit{MHit}]$ for a *given* memoryless profile π. Note that a naive evaluation based on constructing a product Markov chain and solving the linear system of Fig. 3 is exponential in k. A more efficient procedure is given below.

For the rest of this section, we fix an MSSP instance consisting of an MDP $M = (S, \mathit{Act}, P)$, $k \geq 1$, and $\iota_i \in S$, $T_i \subseteq S$ for every $i \in \{1, \ldots, k\}$.

Evaluating Memoryless Profiles. Let $\pi = (\sigma_1, \ldots, \sigma_k)$ be a memoryless profile. Recall that each σ_i is formally a function $S \to \mathit{Dist}(\mathit{Act})$ and determines a Markov chain $M_{\sigma_i} = (S, A_i)$. For the sake of clarity, we use Hit_i to denote a random variable over the runs of M_{σ_i} such that $\mathit{Hit}_i(\omega)$ is either the least $j \in \mathbb{N}$ such that $\omega(j) \in T_i$, or ∞ if there is no such j. Since the agents are independent, we have that

$$\mathbb{E}_\pi[\mathit{MHit}] = \sum_{\ell=1}^{\infty} \mathit{Prob}_\pi[\mathit{MHit} \geq \ell] = \sum_{\ell=1}^{\infty} \prod_{i=1}^{k} \mathit{Prob}_{\iota_i}[\mathit{Hit}_i \geq \ell] \quad (1)$$

For every $i \in \{1, \ldots, k\}$, we fix a target state $\tau_i \in T_i$ and define a transition matrix $\bar{A}_i$ obtained from A_i by changing τ_i into a sink and modifying transitions ending in a state of T_i so that they end in τ_i. More precisely, we put

- $\bar{A}_i(s,t) = A_i(s,t)$ for all $s, t \notin T_i$,
- $\bar{A}_i(s, \tau_i) = \sum_{t \in T_i} A_i(s,t)$ for all $s \notin T_i$,
- $\bar{A}_i(s,t) = 0$ for all $s \notin T_i$ and $t \in T_i \smallsetminus \{\tau_i\}$,
- $\bar{A}_i(t,t) = 1$, and $\bar{A}_i(t,s) = 0$ for all $t \in T_i$ and $s \neq t$.

Then, for every $\ell \geq 0$, we have that $\mathit{Prob}_{\iota_i}[\mathit{Hit}_i \leq \ell] = \bar{A}_i^\ell(\iota_i, \tau_i)$, where $\bar{A}_i^0$ is the identity matrix. Hence,

$$\mathit{Prob}_{\iota_i}[\mathit{Hit}_i \geq \ell] = 1 - \bar{A}_i^{\ell-1}(\iota_i, \tau_i) \quad (2)$$

for all $\ell \geq 1$, and thus we obtain

$$\mathbb{E}_\pi[\mathit{MHit}] = \sum_{\ell=1}^{\infty} \prod_{i=1}^{k} \left(1 - \bar{A}_i^{\ell-1}(\iota_i, \tau_i)\right) \quad (3)$$

Since (3) is an infinite sum, it is not algorithmically workable. Therefore, we also consider a truncated version of (3) where the range of ℓ is restricted to $\{1, \ldots, \gamma\}$ for a suitable constant $\gamma \in \mathbb{N}$. Formally, we define

$$\mathbb{E}\langle\gamma\rangle_\pi[\mathit{MHit}] = \sum_{\ell=1}^{\gamma} \prod_{i=1}^{k} \left(1 - \bar{A}_i^{\ell-1}(\iota_i, \tau_i)\right) \quad (4)$$

Note that $\mathbb{E}\langle\gamma\rangle_\pi[\mathit{MHit}]$ is computable in time *polynomial* in k, γ, and the size of M. Furthermore, we show that for every $\varepsilon > 0$, there is an efficiently computable γ_ε such that

$\mathbb{E}_\pi[\mathit{MHit}] - \mathbb{E}\langle\gamma_\varepsilon\rangle_\pi[\mathit{MHit}] \leq \varepsilon$. Thus, the value of $\mathbb{E}_\pi[\mathit{MHit}]$ can be efficiently ε-approximated by evaluating the right-hand side of (4).

The constant γ_ε is computed as follows. Recall that for every $i \in \{1, \ldots, k\}$, the value of $\mathbb{E}_{\sigma_i}[\mathit{Hit}_i]$ is computable in polynomial time by solving the system of linear equations of Fig. 3. For a given $\varrho \in \mathbb{N}$, we put

$$\delta\langle\varrho\rangle_i = \mathbb{E}_{\sigma_i}[\mathit{Hit}_i] - \sum_{\ell=1}^{\varrho}(1 - \bar{A}_i^{\ell-1}(\iota_i, \tau_i)) \,.$$

Furthermore, let $X_i = \{1, \ldots, k\} \smallsetminus \{i\}$. Then

$$\mathbb{E}_\pi[\mathit{MHit}] - \mathbb{E}\langle\varrho\rangle_\pi[\mathit{MHit}] \;\leq\; \delta\langle\varrho\rangle_i \cdot \prod_{j \in X_i}(1 - \bar{A}_j^{\varrho}(\iota_j, \tau_j)) \quad (5)$$

Note that (5) follows immediately from definitions and the fact that $\bar{A}_j^{\ell}(\iota_j, \tau_j) \geq \bar{A}_j^{\varrho}(\iota_j, \tau_j)$ for all $\ell \geq \varrho$. Hence, γ_ε can be set to the least ϱ such that the right-hand side of (5) is bounded by ε for some $i \in \{1, \ldots, k\}$. Since the right-hand side of (5) decreases exponentially in ϱ, the value of γ_ε is small for most instances.

Representing Memoryless Profiles. Now we show how to represent memoryless strategies by vectors of real-valued parameters so that $\mathbb{E}\langle\gamma\rangle_\pi[\mathit{MHit}]$ becomes a differentiable function of these parameters.

Let $i \in \{1, \ldots, k\}$ be an agent index. For all $s \in S$ and $a \in \mathit{En}(s)$, we fix a fresh parameter (variable) $X_{i,s,a}$ ranging over $\mathbb{R}$. We use Par_i to denote the vector of all $X_{i,s,a}$. Furthermore, for every $s \in S$, we use $\mathit{Par}_{i,s}$ to denote the vector of all $X_{i,s,a}$ where $a \in \mathit{En}(s)$.

Let SOFTMAX be the standard softmax function transforming a vector of real values into a probability distribution, i.e., a vector of the same dimension over $(0, 1]$ whose sum is equal to one. The values of Par_i represent a memoryless strategy $\boldsymbol{\sigma}_i$ such that $\boldsymbol{\sigma}_i(s)(a) = \textsc{SoftMax}(\mathit{Par}_{i,s})(a)$ for every $a \in \mathit{En}(s)$. In other words, for every $s \in S$, the values of $\mathit{Par}_{i,s}$ are transformed into a probability distribution over $\mathit{En}(s)$ using SOFTMAX. Note that $\boldsymbol{\sigma}_i$ can be seen as a function of Par_i. We also use $\bar{\boldsymbol{A}}_i$ and $\bar{\boldsymbol{A}}_i^{\ell}$ to denote the matrices $\bar{A}_i$ and $\bar{A}_i^{\ell}$ determined by $\boldsymbol{\sigma}_i$ (see the previous paragraphs). Again, $\bar{\boldsymbol{A}}_i$ and $\bar{\boldsymbol{A}}_i^{\ell}$ are seen as functions of Par_i. More concretely, for all $s, t \in S$, the element $\bar{\boldsymbol{A}}_i^{\ell}(s, t)$ is a function of Par_i involving SOFTMAX, multiplication, addition, and some constants. Note that $\bar{\boldsymbol{A}}_i^{\ell}(s, t)$ is differentiable.

Let $\mathit{Par}_1, \ldots, \mathit{Par}_k$ be parameters whose values represent a memoryless profile $\boldsymbol{\pi} = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k)$ in the way described above. We use $\mathbb{E}\langle\boldsymbol{\gamma}\rangle_{\boldsymbol{\pi}}[\mathit{MHit}]$ to denote the value of $\mathbb{E}\langle\gamma\rangle_\pi[\mathit{MHit}]$ for the profile $\boldsymbol{\pi}$. Note that $\mathbb{E}\langle\boldsymbol{\gamma}\rangle_{\boldsymbol{\pi}}[\mathit{MHit}]$ is a differentiable function of $\mathit{Par}_1, \ldots, \mathit{Par}_k$.

AUTOHIT Algorithm. The AUTOHIT algorithm (Algorithm 1) is based on minimizing $\mathbb{E}\langle\boldsymbol{\gamma}\rangle_{\boldsymbol{\pi}}[\mathit{MHit}]$ by gradient descent from a suitable initial profile.

More precisely, AUTOHIT starts by invoking an abstract function $\textsc{InitParams}(M, k)$ initializing the parameters $\mathit{Par}_1, \ldots, \mathit{Par}_k$ to values representing an initial profile π_0 (line 4). In our experiments, π_0 is either a *random profile* or a *precomputed profile* (in the latter case, π_0 is computed by some efficient method based on analyzing the structure of M, see Section 5). For the random profile π_0, the value of each parameter $X_{i,s,a}$ is sampled from the normal distribution with expected value 0 and variance 1.

Algorithm 1 The AUTOHIT Synthesis Algorithm

1: **Inputs:**
   MDP $M = (S, \mathit{Act}, P)$,
   the number of agents $k \geq 1$,
   $\iota_i \in S, T_i \subseteq S$ for all $i \in \{1, \ldots, k\}$
2: **Outputs:**
   A memoryless profile π for M and k agents
   Val such that $\mathbb{E}_\pi[\mathit{MHit}] - \varepsilon \leq \mathit{Val} \leq \mathbb{E}_\pi[\mathit{MHit}]$
3: **Numerical hyperparameters:**
   Steps, ε, γ
4: $\mathit{Par}_1, \ldots, \mathit{Par}_k \leftarrow \textsc{InitParams}(M, k)$
5: **for all** *index* $\in \{1, \ldots, \mathit{Steps}\}$ **do**
6: $\quad \mathit{grad} \leftarrow \nabla\, \mathbb{E}\langle\boldsymbol{\gamma}\rangle_{\boldsymbol{\pi}}[\mathit{MHit}](\mathit{Par}_1, \ldots, \mathit{Par}_k)$
7: $\quad \mathit{Par}_1, \ldots, \mathit{Par}_k \leftarrow \textsc{Optimize}(\mathit{grad}, \mathit{Par}_1, \ldots, \mathit{Par}_k)$
8: *Val* $\leftarrow \textsc{Evaluate}(\boldsymbol{\pi}, \varepsilon)$
9: **return** $\boldsymbol{\pi}$, *Val*

The main optimization loop at lines 5–7 is terminated after *Steps* iterations, where *Steps* is a hyperparameter. In each step, the gradient of $\mathbb{E}\langle\boldsymbol{\gamma}\rangle_{\boldsymbol{\pi}}[\mathit{MHit}]$ at $\mathit{Par}_1, \ldots, \mathit{Par}_k$ is computed (line 6) and then passed to a gradient-based optimizer that computes modified parameter values (line 7). Our implementation uses the PYTORCH library [Paszke *et al.*, 2019] to compute the gradient, and $\textsc{Optimize}(\mathit{grad}, \mathit{Par}_1, \ldots, \mathit{Par}_k)$ is implemented by the ADAM optimizer [Kingma and Ba, 2015]. At line 8, the value of $\mathbb{E}_\pi[\mathit{MHit}]$ for the resulting profile $\boldsymbol{\pi}$ is computed up to the precision ε by the function $\textsc{Evaluate}(\boldsymbol{\pi}, \varepsilon)$ whose implementation is described in the paragraph "Evaluating Memoryless Profiles" above.

5 Experiments

Our experimental evaluation aims to answer the following fundamental research questions: *What is the quality of strategies and strategy profiles synthesized by our* COORHIT *and* AUTOHIT *algorithms? What is the efficiency and scalability of these algorithms?* We also aim to compare the quality of solutions obtained in the coordinated and autonomous settings (i.e., estimate the price of autonomy).

Benchmarks. We designed a scalable family of benchmarks resembling cities with congestion at some crossroads. The benchmarks, parameterized by their length l and a "congestion factor" p_c, consist of states $s_{x,y}$ for all $1 \leq x \leq l$, $1 \leq y \leq 5$ and actions *left*, *right*, *up*, *down*. Each state is *congested* with probability p_c (to avoid bias towards specific instances, the subset of congested states is determined randomly). In non-congested states, the actions work as expected, moving the agent left, right, up, and down, respectively. In congested states, the actions work as expected only with probability $p \in \left[\frac{1}{8}, \frac{1}{2}\right]$ and with probability $1-p$ do not move the agent (i.e., they lead back to the same state). Thus, congested states model random delays at road segments. The goal is to reach the target state $s_{l,3}$ from $s_{1,3}$.

Setup. We used a Linux machine with AMD Ryzen 7 PRO 5750G CPU and 32 GB of RAM. All experiments were executed without using GPUs. Each execution has wall clock

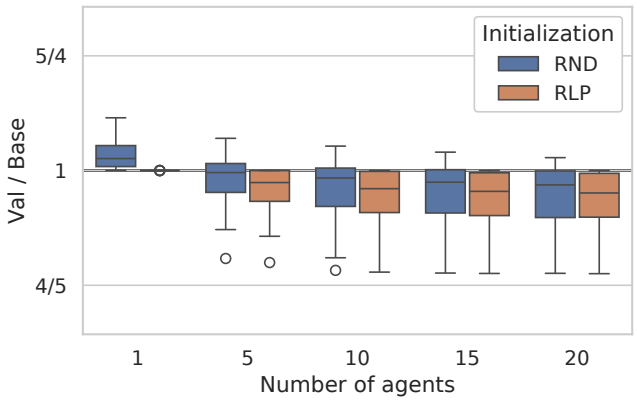

Figure 4: We report the $Val_{\text{RND}}/Base$ (blue) and $Val_{\text{RLP}}/Base$ (brown) ratios. The points below the line $y = 1$ correspond to benchmarks where the ratio is smaller than 1, i.e., the profiles computed by AUTOHIT outperform the baseline.

time limit of 5 minutes. In all experiments, the hyperparameters are set to $Steps = 1000$ and $\varepsilon = 10^{-9}$. The hyperparameter γ is set to the number of states of the MDP. This ensures that each simple path from the initial state to the target state is reflected in the objective function[1] $\mathbb{E}\langle\gamma\rangle_{\pi}[\mathit{MHit}]$. A reproduction package is available as [Jonáš *et al.*, 2026].

Results for AUTOHIT. For each $l \in \{10, 20, 30, 40, 50\}$, we generated 10 random benchmarks with p_c=0.2. Thus, we considered 50 benchmarks in total. As a baseline, we use the profile π_{LP} consisting of the *optimal single-agent strategies* obtained by solving the LP of Fig. 2. We ran AUTOHIT for every $k \in \{1, 5, 10, 15, 20\}$. All AUTOHIT executions took less than 85 seconds of wall time, even for 20 agents and grids with l=50 containing 250 states and 890 actions.

We executed AUTOHIT with two initial parameter settings representing a random profile π_{RND} and a precomputed profile π_{RLP}, which is a "randomized version" of π_{LP}. Note that since π_{LP} is deterministic, it cannot be precisely translated into the parameter values due to the use of SOFTMAX. Instead, each parameter $X_{i,s,a}$ representing π_{RLP} is sampled from the normal distribution with the expected value equal to 10 or 0, depending on whether the strategy σ_i in the profile π_{LP} selects the action a in s or not, respectively (the variance is 1). For each MSSP instance, we generate 5 randomly sampled versions of π_{RND} and π_{RLP}, and we use Val_{RND} and Val_{RLP} to denote the *average Val* returned by AUTOHIT for these 5 versions of π_{RND} and π_{RLP}, respectively. We compare Val_{RND} and Val_{RLP} against the baseline $Base = \text{EVALUATE}(\pi_{\text{LP}}, \varepsilon)$.

The results are presented in Fig. 4. The value Val_{RND} is on average 97.5 % of *Base* and is even 81.9 % in some cases. For Val_{RLP}, this improves to 96.0 % on average and 81.8 % in the best case. Hence, AUTOHIT computes strategy profiles that *outperform* the baseline, decreasing the expected value of *MHit* by almost 20 % in some cases. When initialized with π_{RLP}, the profile computed by AUTOHIT outperforms the baseline *in the vast majority of the benchmarks*.

[1]We also experimented with other values of γ, the results are available in [Jonáš *et al.*, 2026].

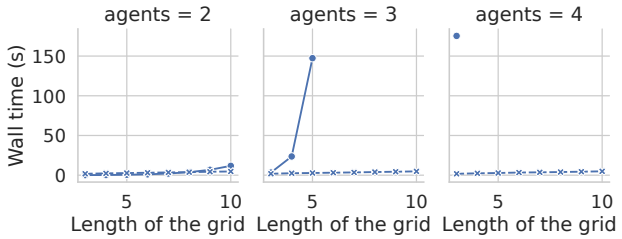

Figure 5: Average execution times of COORHIT (solid line) and AUTOHIT (dashed line).

We also ran similar experiments with another precomputed initial profile π_{SP} consisting of strategies following a graph-theoretic shortest path in the MDP. However, the quality of the resulting profiles computed by AUTOHIT tends to be worse than that for π_{RND} and π_{RLP}. We refer to [Jonáš *et al.*, 2026] for details.

Results for COORHIT. Our COORHIT implementation is based on Gurobi [Gurobi Optimization, LLC, 2024] LP solver. Since COORHIT does not scale so well in the number of agents, we use smaller grids. We generated 10 random benchmarks for each l between 3 and 10. To ensure that there are some congested crossroads on the shortest path, we used $p_c = 1$. We ran the algorithms with 2, 3, and 4 agents.

The average runtimes of COORHIT and AUTOHIT (initialized with the π_{RLP} profile) are shown in Fig. 5. Consistently with our theoretical results, AUTOHIT scales significantly better than COORHIT. For 4 agents, COORHIT did not terminate within 5 minutes for any grid with l>3. Hence, AUTOHIT is significantly more efficient than COORHIT.

In the cases where both algorithms terminated, the expected values of *MHit* for the resulting strategies and strategy profiles differed by 0.01 on average and at most by 0.08. This shows that the price of autonomy is rather low in the considered benchmarks.

6 Conclusions

Our initial study of the MSSP problem reveals the fundamental complexity barriers for designing efficient algorithmic solutions in both coordinated and autonomous setting. Our AUTOHIT algorithm successfully overcomes the barriers in the (more challenging) autonomous setting and produces high-quality solutions.

For our randomly generated benchmarks, AUTOHIT *consistently* improves the expected value of *MHit* over the baseline $\mathbb{E}_{\pi_{\text{LP}}}[\mathit{MHit}]$ where π_{LP} is the profile consisting of optimal single-agent strategies. In some cases, this improvement is *significant*. This shows that despite the high computational complexity of the MSSP problem, there are efficient synthesis algorithms producing solutions outperforming the natural baseline $\mathbb{E}_{\pi_{\text{LP}}}[\mathit{MHit}]$.

The questions whether a similarly efficient synthesis algorithm exists also for the coordinated setting and whether alternative approaches can lead to even better synthesis algorithms certainly deserve an adequate research effort left for future work.

Acknowledgements

The work received funding from the Czech Science Foundation, grant No. 26-23441S.

References

[Ahmadi *et al.*, 2021] M. Ahmadi, A. Dixit, J.W. Burdick, and A.D. Ames. Risk-averse stochastic shortest path planning. In *Proceedings of 60th IEEE Conference on Decision and Control (CDC 2021)*, pages 5199–5204. IEEE Computer Society Press, 2021.

[Baier *et al.*, 2018] C. Baier, N. Bertrand, C. Dubslaff, D. Gburek, and O. Sankur. Stochastic shortest paths and weight-bounded properties in Markov decision processes. In *Proceedings of LICS 2018*, pages 86–94. ACM Press, 2018.

[Bertsekas and Tsitsiklis, 1991] D.P. Bertsekas and J.N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[Chavan *et al.*, 2025] U. Chavan, P. Trivedi, and H. Hemachandra. Regret lower bounds for decentralized multi-agent stochastic shortest path problems. In *Advances in Neural Information Processing Systems (NeurIPS 2025)*, volume 38, pages 173948–173992. Curran Associates, Inc., 2025.

[de Alfaro, 1999] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *Proceedings of CONCUR'99*, volume 1664 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1999.

[Díaz-García *et al.*, 2023] G. Díaz-García, F. Bullo, and J.R. Marden. Distributed Markov chain-based strategies for multi-agent robotic surveillance. *IEEE Control Systems Letters*, 7:2527–2532, 2023.

[Eaton and Zadeh, 1962] J.H. Eaton and L.A. Zadeh. Optimal pursuit strategies in discrete-state probabilistic systems. *Journal of Basic Engineering*, 84(1):23–29, 1962.

[Gurobi Optimization, LLC, 2024] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024.

[Haase and Kiefer, 2015] Ch. Haase and S. Kiefer. The odds of staying on budget. In *Proceedings of ICALP 2015*, volume 9135 of *Lecture Notes in Computer Science*, pages 234–246. Springer, 2015.

[Jiang *et al.*, 2017] X. Jiang, X. Wang, H. Xi, and F. Liu. Centralized optimization for Dec-POMDPs under the expected average reward criterion. *IEEE Transactions on Automatic Control*, 62(11):6032–6038, 2017.

[Jonáš *et al.*, 2025] M. Jonáš, A. Kučera, V. Kůr, and J. Mačák. Steady-state strategy synthesis for swarms of autonomous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2025)*, pages 135–142, 2025.

[Jonáš *et al.*, 2026] M. Jonáš, A. Kučera, V. Kůr, J. Mačák, and V. Řehák. Multiagent stochastic shortest path problem. *arXiv*, 2605.06056 [cs.MA], 2026.

[Jonáš *et al.*, 2026] M. Jonáš, A. Kučera, V. Kůr, J. Mačák, and V. Řehák. Reproduction Package for IJCAI 2026 Paper "Multiagent Stochastic Shortest Path Problem" , Zenodo, May 2026.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR 2015*, 2015.

[Klaška *et al.*, 2023] D. Klaška, A. Kučera, M. Kurečka, P. Novotný, V. Musil, and V. Řehák. Synthesizing resilient strategies for infinite-horizon objectives in multiagent systems. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2023)*, pages 171–179, 2023.

[Křetínský and Meggendorfer, 2018] J. Křetínský and T. Meggendorfer. Conditional value-at-risk for reachability and mean payoff in Markov decision processes. In *Proceedings of LICS 2018*, pages 609–618. ACM Press, 2018.

[Mannor and Tsitsiklis, 2011] S. Mannor and J.N. Tsitsiklis. Mean-variance optimization in Markov decision processes. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 177–184. Omnipress, 2011.

[Nawaz and Ornik, 2023] F. Nawaz and M. Ornik. Multiagent, multitarget path planning in Markov decision processes. *IEEE Transactions on Automatic Control*, 68(12):7560–7574, 2023.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Piribauer *et al.*, 2022] J. Piribauer, O. Sankur, and C. Baier. The variance-penalized stochastic shortest path problem. In *Proceedings of ICALP 2022*, volume 229 of *Leibniz International Proceedings in Informatics*, pages 129:1–129:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.

[Puterman, 1994] M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[Randour *et al.*, 2015] M. Randour, J.-F. Raskin, and O. Sankur. Variations on the stochastic shortest path problem. In *Proceedings of VMCAI 2015*, volume 8931 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2015.

[Rosenberg, 2022] A. Rosenberg. *Regret minimization in reinforcement learning*. PhD thesis, Tel Aviv University, Israel, 2022.

[Trivedi and Hemachandra, 2023] P. Trivedi and N. Hemachandra. Multi-agent congestion cost minimization with linear function approximations. In

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, pages 7611–7643, 2023.

[Zhang *et al.*, 2021] L. Zhang, A. Prorok, and S. Bhattacharya. Pursuer assignment and control strategies in multi-agent pursuit-evasion under uncertainties. *Frontiers in Robotics and AI*, 8:691637, 2021.