9th International Conference on Prestigious Applications of Intelligent Systems – PAIS@ECAI2020
Santiago de Compostela, Spain

Entropy-based adaptive exploit-explore coefficient for Monte-Carlo path planning

Ana Raquel Carmo[1] and **Jean-Alexis Delamer**[2] and **Yoko Watanabe**[3] and **Rodrigo Ventura**[1] and **Caroline P. C. Chanel**[4]

Abstract.

Efficient path planning for autonomous vehicles in cluttered environments is a challenging sequential decision-making problem under uncertainty. In this context, this paper implements a partially observable stochastic shortest path (PO-SSP) planning problem for autonomous urban navigation of Unmanned Aerial Vehicles (UAVs). To solve this planning problem, the POMCP-GO algorithm is used, which is goal oriented variant of POMCP, one of the fastest online state-of-the-art solvers for partially observable environments based on Monte Carlo Planning. This algorithm relies on the Upper Confidence Bounds (UCB1) algorithm as action selection strategy. UCB1 depends on an exploration constant typically adjusted empirically. Its best value varies significantly between planning problems, and hence, an exhaustive search to find the most suitable value is required. This exhaustive search applied to a complex path planning problem may be extremely time consuming. Moreover, considering real applications where online planning is needed, this extensive search is not suitable. Thereby this paper explores the use of an adaptive exploration coefficient for action selection during planning. Monte-Carlo value backup approximation is also applied which empirically demonstrates to accelerate the policy value convergence. Simulation results show that the use of the adaptive exploration coefficient within a user-defined interval achieves better convergence and success rates when compared with most hand-tuned fixed coefficients in said interval, although never achieving the same results as the best fixed coefficient. Therefore, a compromise must be made between the desired quality of the results and the time one is willing to spend on the exhaustive search for the best coefficient value before planning.

1 Introduction

Navigation of autonomous vehicles becomes challenging in a cluttered environment, e.g. urban areas, in which the precision or even the availability of some onboard navigation sensors may vary significantly. For instance, the localization precision of a Global Positioning System (GPS) depends on the satellite constellation visibility, which in turn depends on the geo-localization, time and surrounding obstacle configurations [2]. However, because the GPS satellite constellation is known, a prediction of its localization precision can be given as a metric called Dilution of Precision (DOP) [3]. Such information can be used as prior knowledge in the path planning problem, in order to guarantee safety despite possible degradation of navigation capability.

In this context, this paper addresses the safe path planning for autonomous vehicles in urban environments, by taking into consideration probabilistic onboard sensor availability maps and path execution error propagation. Previous work [4] [5] dealt with the problem of deterministic and discrete path planning by considering the vehicle localization uncertainty propagated along a calculated path according to the environment. Delamer et al. (2019) [6] have recently integrated a closed-loop vehicle motion model with Guidance, Navigation and Control (GNC) modules in the sequential decision-making process to propagate the influence of sensor availabilities on the uncertainty of the trajectory executed.

Building upon [6], the path planning problem is modeled as a Mixed-Observability Markov Decision Process (MOMDP) [7, 8]. This mathematical formalism is an extension of the classical Partially Observable Markov Decision Process (POMDP) [9] to allow the factorization of the belief state space through the definition of fully and partially observable (or even hidden) state variables. Thus it benefits from a smaller belief state space dimension, for accelerating policy computation. Applied to the problem here addressed, the state transition and observation functions of the MOMDP model are built based on the vehicle's GNC motion model defined in a continuous state space as well as on a priori knowledge of the environment given by probability grid maps of obstacles and on-board sensor availability. Therefore, the resulting function issues the belief state function to have a complex (non-Gaussian) form.

To tackle this problem, [6] proposed to use the Partially Observable Monte Carlo Planning (POMCP) algorithm [10], in a goal oriented configuration, here called POMCP-GO [7]. The POMCP, currently one of the fastest online state-of-the-art POMDP solvers, simplifies the representation of the belief state by approximating it to a set of particles. Furthermore, POMCP extends the Upper Confidence Bound Applied to Trees (UCT) algorithm [11] to partially observable environments. As in UCT, POMCP applies the Upper Confidence Bounds (UCB1) action selection strategy [2] during value and policy optimization to deal with the exploration-exploitation dilemma, while minimizing the regret of choosing wrong actions [1]. UCB1 strategy depends on an exploration coefficient c, whose value is typically constant and adjusted empirically. The most suitable value of this parameter varies significantly between planning domains, requiring an exhaustive search to find it.

[1] Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal, email: raquelcarmo@tecnico.ulisboa.pt, rodrigo.ventura@isr.tecnico.ulisboa.pt

[2] Queen's University, Kingston, Canada, email: name.surname@cs.queensu.ca

[3] ONERA - The French Aerospace Lab, Toulouse, France, email: name.surname@onera.fr

[4] ISAE-SUPAERO, Université de Toulouse, France, email: name.surname@isae-supaero.fr

Motivated by the difficulty of finding the best parameter value, some researchers have been striving to refine the UCB1 formula with heuristic properties to facilitate the action selection process during search. For instance, [12] proposes Progressive Bias to linearly combine the standard UCB1 evaluation with an heuristic evaluation with a weight proportional to the number of simulations. The more simulations are performed, the more statistical confidence, and therefore, the higher weight is assigned to the standard UCB1 formula (still depending on a constant parameter value). Another approach [13] explores the use of simple regret minimizing bandit algorithms at the root, while using UCB1 throughout the tree, which has shown the potential to overcome some weaknesses of the UCT algorithm. A variant of the UCT algorithm, called PUCT [14] [15], used in deep Reinforcement Learning (RL), exploits the neural network to predict the next action. [18] attempted to dynamically tune the exploration coefficient from the UCB1 formula, without however being able to demonstrate an enhancement in the performance relatively to the fixed value for the Arimaa game problem. [21] also explored an adaptive coefficient, whose value decays with the depth of the search tree, considering only the tree depth but does not the planning domain.

Besides, it is known that the Q-value approximation proposed by UCT and POMCP includes a bias. Keller and Helmert (2013) [17] proposed the trial-based heuristic tree-search framework, which incorporates ingredients from Monte-Carlo Tree Search, Dynamic Programming (DP) and Heuristic Search. Within their framework, they derive three novel algorithms: MaxUCT, that merges action-value Monte-Carlo back-propagation function and state-value Full Bellman back-propagation function; DP-UCT, which considers (model) probabilities in the backups of action-value estimates; and UCT*, that incorporates trial length in DP-UCT. Such variants of UCT are proven to perform significantly better and in less time than the standard UCT algorithm.

In this context, this paper proposes an Entropy-based adaptive exploration coefficient to be applied in the UCB1 formula during planning. This adaptive coefficient is proportional to a measure of the uncertainty of possible action outcomes, which enables to explore more when uncertainty is higher. This newly proposed adaptive coefficient relates directly to the observation probability distribution. In our view it constitutes a better option for an online planning framework because it depends only on the uncertainty modeled in the observation function exempting the need for an exhaustive coefficient value search. Moreover, this paper adapts MaxUCT [17] to POMCP-GO, leading to a new Q-value and value approximation strategy. A combination of the Entropy-based adaptive exploration coefficient and this new value approximation strategy proposed in this paper is expected to accelerate policy value convergence during planning.

This paper is organized as follows: firstly the MOMDP model for the safe path planning application case is presented in Section 2. This is followed by the POMCP-GO algorithm, with specific modifications in order to fit the MOMDP problem. Section 3 introduces the proposed adaptive coefficient and the value backup strategy. Simulation results are presented in Section 4 to show the impact of the adaptive coefficient and the value backup strategy on the convergence of the value function as well as on the success rates. Lastly, conclusions and future work are discussed.

2 Partially Observable Shortest Path (PO-SSP) planning problem for an autonomous UAV

PO-SSP problem for UAV navigation considered in this paper deals with a problem of planning safe (avoiding obstacles) and efficient (minimum distance or time) trajectories towards a goal under uncertainty, taking into account the availability of onboard navigation sensors that depends on the environment.An a priori knowledge on the environment is assumed to be given as a set of probability grid maps of obstacles and availability of each of the sensors. These maps are used during the path planning task to propagate the path execution uncertainty through the vehicle GNC motion models, given the probabilistic sensor's availability.

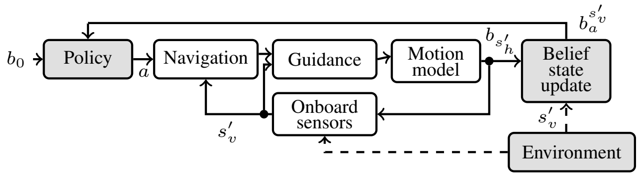


Figure 1: Modules architecture. The white blocks represent the closed-loop GNC vehicle motion model.

2.1 PO-SSP Planning Model

The PO-SSP planning problem addressed is modeled as a MOMDP [8]. Assuming that a vehicle always knows if a given sensor measurement can be used or not at the current decision time step, the sensor availabilities are considered as a fully observable state variables of the model. On the other hand, the vehicle state vector (e.g. position) is non-observable from the planning model point of view, because the planning module does not have a direct access to it. However, thanks to the closed-loop vehicle motion model based on the GNC modules, for a given action, it is possible to propagate the probability density (i.e. belief state) of the vehicle state vector. The planning model architecture is illustrated in Fig. 1.

The MOMDP is defined as a tuple $(\mathcal{S}_v, \mathcal{S}_h, \mathcal{A}, \Omega, \mathcal{T}, \mathcal{O}, \mathcal{C}, \mathcal{G}, b_0)$, where $\mathcal{S}_v$ is the space of fully observable states, $\mathcal{S}_h$ is the space of hidden continuous states, thus one has $s = (s_v, s_h) | s \in \mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$. $\mathcal{A}$ is the set of actions; Ω is the set of observations; $\mathcal{T}$ is the state transition function, such as $\mathcal{T}(s'_h, s'_v, a, s_v, s_h) = \Pr(s'_h, s'_v | a, s_h, s_v)$; $\mathcal{O}$ is the observation function, such as $\mathcal{O}(o, a, s'_h, s'_v) = \Pr(o | s'_h, s'_v, a)$; $\mathcal{C} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_0^+$ is the cost function; G the set of goal states, $b_0 = (s_v^0, b_{\mathcal{S}_h}^0)$, where $b_{\mathcal{S}_h}^0 \in \mathcal{B}_h$ is the initial probability distribution over the initial hidden continuous states, conditioned to $s_v^0 \in \mathcal{S}_v$, the initial fully observable state. The figure 2 schematizes the MOMDP model being considered.

The visible state $s_v \in \mathcal{S}_v$ is defined as a tuple $s_v = (F_{S_1}, F_{S_2}, ..., F_{S_N}, F_{Col}, \mathbf{P}, \Theta)$, where F_{S_i} defines the fully observable Boolean state variable for the availability of sensor S_i, and F_{Col} represents a fully observable Boolean variable for a collision flag. $\mathbf{P}$ is the localization error covariance matrix computed by the navigation module. Θ represents the total flight time from b_0 until s.

The hidden continuous state $s_h \in \mathcal{S}_h$ is defined as $s_h = \mathbf{x}$, the vehicle state vector defined by the position, the velocity and the accelerometer bias, such as $\mathbf{x} = \begin{bmatrix} \mathcal{X}^T & \mathcal{V}^T & \beta_a^T \end{bmatrix}^T$. Apart from the vehicle position $\mathcal{X}$, it is necessary to consider the velocity $\mathcal{V}$ and the accelerometer bias β_a in the state s_h, as they are considered in the transition function to estimate the next state.

An action $a \in \mathcal{A}$ is defined as a tuple $a = (\{\mathcal{V}_{ref}\}, m_n)$, where $\mathcal{V}_{ref}$ defined the reference velocity given to the guidance module, $\{\mathcal{V}_{ref}\}$ defines a finite set of possible $\mathcal{V}_{ref}$ and $m_n \in \{S_1, ..., S_N\}$ is the navigation mode to select a subset of sensors to be used in the navigation module at each planning epoch t, depending on their availabilities.

Given the specificity of the planning model addressed, the set of observations Ω is equal to $\mathcal{S}_v$ (see Fig. 2). Although the agent receives no direct observation on the state s_h. The execution error prop-

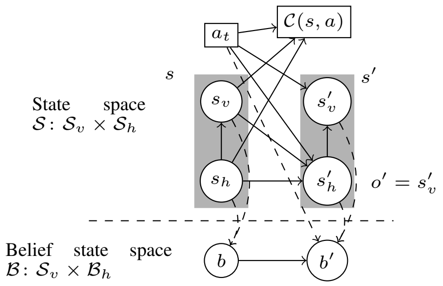

a_t
$\mathcal{C}(s,a)$
s
s'
State space
$\mathcal{S}$: $\mathcal{S}_v \times \mathcal{S}_h$
s_v
s'_v
s_h
s'_h
$o' = s'_v$
Belief state space
$\mathcal{B}$: $\mathcal{S}_v \times \mathcal{B}_h$
b
b'

Figure 2: MOMDP transition model being considered.

agation dynamics depending on the action selected (reference velocity and navigation mode) are known. This avoids considering sensor measurements, which are not accessible at the moment of planning, and therefore it also avoids working with a continuous observation space.

Following the vehicle (GNC) transition model described in [6], the complete transition function is given as: $\mathcal{T}(s'_h, s'_v, a, s_v, s_h) = \Pr(s'_v|s'_h)\Pr(s'_h|s_h, s_v, a)$ where, $\Pr(s'_v|s'_h)$ represents the transition function for s'_v, which depends only on the probabilistic sensors' availability maps, and $\Pr(s'_h|s_h, s_v, a) \sim \mathcal{N}(\bar{s}'_h, \sum')$, which is based on the GNC closed-loop vehicle motion model. It gives the probability density of a predicted state s'_h as a normal distribution $\mathcal{N}(\bar{s}'_h, \sum')$, which in turn, is a function of the previous state s_h, the previous visible state s_v and the action a.

The cost function is defined as: $\mathcal{C}(s,a) = \{0$ if $s \in \mathcal{G}, K - \Theta,$ if s in collision, f_t otherwise$\}$, with, f_t is the flight time for a given action a at decision step t and K is a fixed cost in case of collision. When a collision occurs, the cost of any action is a fixed penalty subtracted with the total flight time (Θ) from the initial belief state until the collision state. This trick avoids penalizing more if the collision occurs after a longer flight time or near the goal.

The aim of solving a MOMDP problem is to find a policy $\pi : \mathcal{B} \to \mathcal{A}$, where $\mathcal{B}$ defines the belief state space (i.e. $\mathcal{B} : \mathcal{S}_v \times \mathcal{B}_h$), which optimizes a given criterion usually defined by a value function. In the PO-SSP planning problem addressed, the value function $V^{\pi}(b)$ is defined as the expected total cost when starting from b_0 and following policy π. Therefore, the value function takes the following form:

$$V^{\pi}(b) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \mathcal{C}(b_t, \pi(b_t)) | \, b_0 = b\right] \qquad (1)$$

where $\mathcal{C}(b_t, \pi(b_t) = a)$ is the expected cost of executing an action $a \in \mathcal{A}$ in belief state $b_t \in \mathcal{B}$. The optimal policy π^* is defined by the optimal value function V^*, given by:

$$V^*(b) = \min_{a \in \mathcal{A}} \left[\mathcal{C}(b,a) + \sum_{s_v \in \mathcal{S}_v} \Pr(s_v|b,a) V^*(b_a^{s_v})\right] \qquad (2)$$

with $\mathcal{C}(b,a) = \mathbb{E}\left[\mathcal{C}(s,a)|\forall s \in b, a \in \mathcal{A}\right]$. It is expected that by minimizing the expected cost, the algorithm will minimize the expected flight time (for efficiency) and the probability of collision (for safety) at the same time.

The action's Q-value can be defined as the value of performing an action a in belief state b, assuming the optimal policy will be followed afterwards, as in:

$$Q(b,a) = \left[\mathcal{C}(b,a) + \gamma \sum_{s_v \in \mathcal{S}_v} \Pr(s_v|b,a) V^*(b_a^{s_v})\right] \qquad (3)$$

Solving the MOMDP problem here proposed (and in general) is not a trivial task. The process of keeping and updating the belief states is challenging. Furthermore, in the path planning problem here addressed, the state distribution can not be represented by a general form (e.g. non-Gaussian). More precisely, as the hidden state space is continuous and the fully observable state space is discrete, the resulting computation of the probability distribution over $\mathcal{S}_h$ and correction by s_v would be computationally expensive. Moreover, the computation of $\Pr(s_v|b,a)$ necessary to the value approximation is also a time consuming step. For this reason, approaches based on Monte-Carlo Tree Search (MCTS), like the Partially Observable Monte Carlo Planning (POMCP) algorithm [10], that do not require to explicitly update the belief state in each decision step, become a promising solution to apply.

2.2 POMCP Algorithm

POMCP is a Monte Carlo Tree Search algorithm for partially observable environments [10] and works by sampling a state s from the initial belief state b_0, and simulating sequences of action-observation (by a trial procedure) to construct a tree of history (belief) nodes. Each tree node h represents an history of action-observation pairs since the initial belief state. POMCP calculates for a given node h of the tree the average cost observed for all trials that have started from this node. Rather than updating the belief state after each action-observation pair, POMCP keeps in memory the number of times a node was explored $N(h)$ as well as the number of times a given action a was chosen $N(ha)$ in this node, allowing to approximate the Q-value $Q(h,a)$ by the mean return from all trajectories started from the history h when action a was selected. Note it differs from the Q-value definition, as introduced in Eq. 3.

During planning, POMCP relies on the Upper Confidence Bounds (UCB1) action selection strategy [1] to deal with the exploration-exploitation dilemma. This action selection strategy is based on a combination of two terms, the action Q-value and a measure of how well explored an action is, as follows:

$$\bar{a}_{\text{UCB}} = \arg\min_{a \in \mathcal{A}} \left\{ Q(h,a) - c\sqrt{\frac{\log N(h)}{N(h,a)}} \right\} \qquad (4)$$

While the first term in Eq. (4) vouches for the exploitation of the previously visited choices with the highest reward values, the second encourages the exploration of undiscovered nodes. The exploration coefficient c forces the algorithm to explore actions that seem less promising in order to avoid falling into a local optimum policy. The larger the c is, the more the exploration is prioritized over the exploitation. Hence the value of c directly influences the policy convergence.

2.3 POMCP-GO Algorithm

The POMCP-GO algorithm [6], recalled in Algorithm 1, is a goal oriented variant of the POMCP algorithm for the PO-SSP problem. The main differences between the original POMCP and POMCP-GO are hereafter discussed. In the classical POMCP algorithm, the value of a tree node is estimated based on sequences of UCB1 greedy action selections until a leaf node is reached, while POMCP-GO the sequences end only when a terminal state is reached (either a goal or a collision), applying a depth-first search as proposed in [23]. When a new node needs to be created in the classical POMCP, a rollout method is introduced to estimate its value by simulating sequences of random action-observation pairs starting from this new node, whereas POMCP-GO estimates its value using an heuristic

Algorithm 1: POMCP-GO (adapted from [6])

```
1  Function POMCP-GO($h, b_0, c$):
2      $h \leftarrow b_0$
3      while $nbTrial < nb_{max}$ do
4          $s_h \sim b_0$
5          Trial($h, s_h, s_v, c, 0$)
6          $nbTrial += 1$
7      return $a^* \leftarrow \arg\min_{a \in \mathcal{A}} Q(b_0, a)$
8  Function Trial($h, s_h, s_v, c, d$):
9      if $s_h \in \mathcal{G}$ then
10         return 0
11     if $F_{col} == 1$ then
12         return $K - \Theta$
13     if $h \notin T$ then
14         for $a \in \mathcal{A}$ do
15             Creating $ha$ node
16             $T(ha) \leftarrow (N_{init}(ha), Q_{init}(h,a), \emptyset)$
17         $V_{init}(h) \leftarrow \min_{a \in \mathcal{A}} Q(h,a)$
18     $\bar{a} \leftarrow$ ActionSelection($Q(h,a), N(h), N(ha), c$)
19     $(s'_h, s'_v, \mathcal{C}(s_h, \bar{a})) \sim \mathcal{G}(s_h, \bar{a})$
20     Creating $hao$ node (if necessary) with $s'_v = o, a = \bar{a}$
21     $Q(h,\bar{a})' \leftarrow \mathcal{C}(s_h, \bar{a}) +$ Trial($hao, s'_h, s'_v, c, d+1$)
22     $N(h) \leftarrow N(h) + 1$
23     $N(h\bar{a}) \leftarrow N(h\bar{a}) + 1$
24     $V(h),\ Q(h,\bar{a}) \leftarrow$ Backup($\mathcal{C}(s_h, \bar{a}), Q(h,\bar{a})'$)
```

value that explores the a pre-computed trivial solution, given by the Dijkstra algorithm [20] without considering uncertainties. It gives an optimistic estimated flight time. To be noted that this value initialization gives an informative value approximation in this goal-oriented path planning problem, for a given state in a given grid cell, when compared to the rollout policy.

3 Adaptive exploit-explore coefficient and backup value approximation

POMCP-GO relies on Eq. 4 (called by Alg. 1 line 18) as an action selection strategy. The exploration factor c is a typically a constant value adjusted manually. The best value of this parameter varies significantly between planning domains requiring an exhaustive search to it. Given the complexity of the PO-SSP planning problem addressed, this exhaustive search is extremely time consuming. Moreover, in real flight settings where the online planning is needed, this previous coefficient value search is not suitable. Therefore, this paper proposes a new entropy-based adaptive coefficient (EBC).

3.1 Entropy-based Coefficient (EBC)

Similarly to the approach proposed in [18], this EBC method dynamically tunes the UCB1 exploration coefficient within an interval of values $[c_{min}, c_{max}]$ that needs to be specified a priori by the user. The proposed method determines the coefficient based on a score related to a measure of the uncertainty about the possible actions outcomes (e.g observations) so that the planner explores more thoroughly in areas where the uncertainty is higher. In other words, it adjusts the coefficient value according to the entropy of the probability related to the navigation sensors' availabilities. Thus, c_{EBC} replaces c in Eq. 4, and is defined such as:

$$c_{EBC}(s_v, s_h) = e_n(s_v, s_h) \max_{\substack{s \in \mathcal{S} \\ a \in \mathcal{A}}} |\mathcal{C}(s,a)| \tag{5}$$

where, $e_n(s_v, s_h) = (c_{max} - c_{min})\, e(s_v, s_h) + c_{min}$ is the normalized entropy to fit in the user-defined interval $[c_{min}, c_{max}]$, with $e(s_v, s_h) = -\sum_{s_v \in \int_v} \Pr(s_v|s_h) \log_2(\Pr(s_v|s_h))$ being the value of the entropy according to the probability grid map of the navigation sensors' availability. In this case, $\int_v = (F_{S_1}, F_{S_2}, ..., F_{S_N})$ is a subset of $\mathcal{S}_v$ that only considers the fully observable Boolean state variables F_{S_i}, i.e. neither $\mathbf{P}$, F_{Col} nor even Θ are accounted for in this computation.

3.2 Backup strategies

The backup function of the POMCP-GO algorithm (see Alg. 1 line 24) defines how the knowledge on action-value estimates $Q(h,a)$ is propagated through the tree [17].

Classical POMCP [10] and POMCP-GO [7] approaches The action-value estimates are updated based on the mean return from all trials started when action a was selected in history h. Let $Q(h,a)'$ be the current return (line 21 in Alg. 1), then the state-value and action-value estimates are calculated as:

$$\begin{cases} V(h) \leftarrow \min_{a \in \mathcal{A}} Q(h,a) \\ Q(h,a) \leftarrow Q(h,a) + \dfrac{Q(h,a)' - Q(h,a)}{N(ha)} \end{cases} \tag{6}$$

As the action-value estimate $Q(h,a)$ averages over all trajectories started in that action a, and not over trials starting with a and following the current best policy, this might cause a potential pitfall: if a trajectory yields a very high cost compared to an optimal one, a single trial over said course can bias $Q(h,a)$ disproportionally over many trials [17].

MinPOMCP-GO approach This paper proposes to apply the value update strategy of the MaxUCT algorithm [17]. In this approach, the estimation of the action-values is based on the value of its best successor, rather than on all trials, as follows:

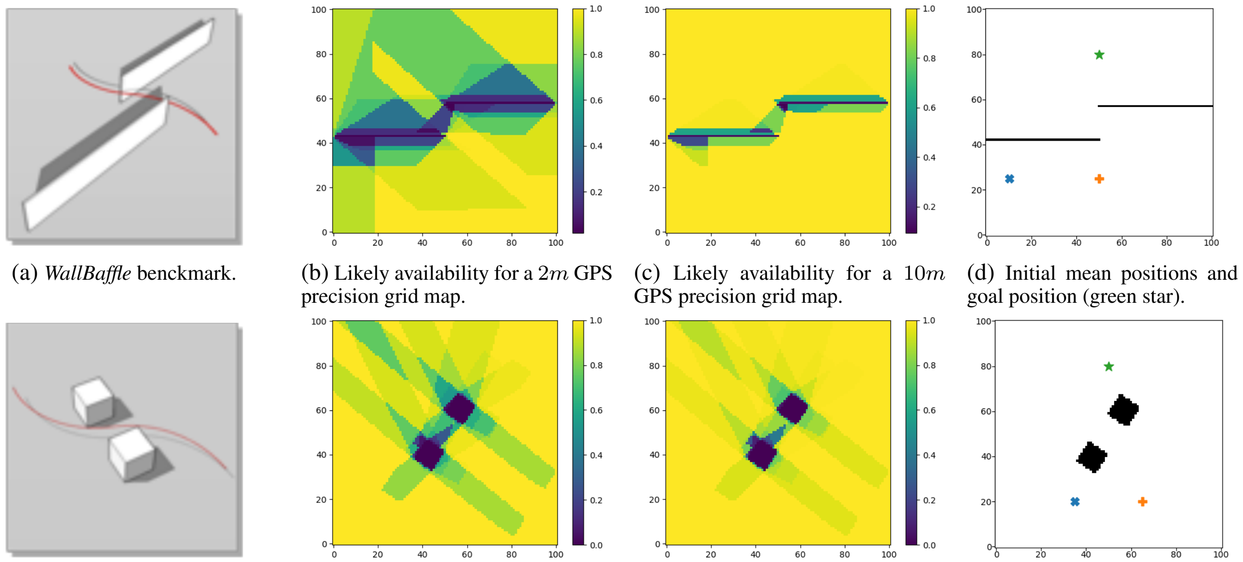
$$\begin{cases} V(h) \leftarrow \min_{a \in \mathcal{A}} Q(h,a) \\ Q(h,a) \leftarrow C(h,a) + \dfrac{\sum_{hao} N(hao)\, V(hao)}{N(ha)} \end{cases} \tag{7}$$

where,

$$C(h,a) \leftarrow C(h,a) + \frac{\mathcal{C}(s_h,a) - C(h,a)}{N(ha)} \tag{8}$$

is the mean immediate cost of executing action a in history h. As a result of applying this value backup strategy, the contributing subtree in the action-value approximation is identical to the best partial solution tree. Therefore, the pitfall discussed in the Classical POMCP strategy no longer applies, because the best approximated value is back-propagated. It is important to mention that this method can be applied only after certain number of trials are performed.

In the next section, simulation experiments are introduced in order to evaluate the entropy-based coefficient c_{ECB} proposed for replacing c in the UCB1 action selection strategy, along with the backup value approximation proposed called MinPOMCP-GO.



(a) *WallBaffle* benckmark.

(b) Likely availability for a $2m$ GPS precision grid map.

(c) Likely availability for a $10m$ GPS precision grid map.

(d) Initial mean positions and goal position (green star).

(e) *CubeBaffle* benchmark.

(f) Likely availability for a $1m$ GPS precision grid map.

(g) Likely availability for a $2m$ GPS precision grid map.

(h) Initial mean positions and goal position (green star).

Figure 3: Obstacle maps from [22] and examples of GPS probabilistic availability grid maps with different precision thresholds. The 2 initial belief states are illustrated by the mean initial position: $(10, 25, 5)m$ (blue cross) and $(50, 25, 5)m$ (orange plus) for the *WallBaflle* map; $(35, 20, 5)m$ (blue cross) and $(65, 20, 5)m$ (orange plus) for the *CubeBaffle*. Goal position was set at $(50, 80, 5)m$ (green star) for both maps.

4 Simulations results

Configuration A total of 8 study cases were examined with 2 benchmarks environment maps from [22] (see Fig. 3(a) and Fig. 3(e)). These environment maps contain a grid size of $100 \times 100 \times 20$ cells, where each grid cell has the size of $2m \times 2m \times 2m$. It is assumed that only IMU and GPS are equipped as onboard navigation sensors. IMU is considered to be available all the time. For each environment map, 2 distinct a priori knowledge on the probabilistic GPS availability depending on the minimum required GPS precision were supposed (see Fig. 3(b)-(c) and Fig. 3(f)-(g)). For the *WallBaffle* map, $2m$ and $10m$ GPS precision availability were examined, while in the *CubeBaffle* map, $1m$ and $2m$ GPS precision availability were chosen. The initial belief state is defined as $b_0 = (s_v^0, b_{\mathcal{S}_h^0} = (\bar{s}_h^0, \tilde{\Sigma}_0))$, where $s_v^0 = [1, 1, 0, \mathbf{P}]$ is the initial visible state; $\mathbf{P} = \tilde{\Sigma}_0 = diag(1, 1, 1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01)$. Two initial mean states $\bar{s}_h^0$ are considered for each map (see Fig. 3(d) and 3(h)). For *WallBaffle*: $\bar{s}_h^0 = [10, 25, 5, 0, 0, 0, 0, 0, 0]$ and $\bar{s}_h^0 = [50, 25, 5, 0, 0, 0, 0, 0, 0]$; and for *CubeBaffle*: $\bar{s}_h^0 = [35, 20, 5, 0, 0, 0, 0, 0, 0]$ and $\bar{s}_h^0 = [65, 20, 5, 0, 0, 0, 0, 0, 0]$.The target state is given at the position $s_g = (50, 80, 5)m$ (see green stars in Fig. 3). The set of $\{\mathcal{V}_{ref}\}$ is composed by 26 reference velocity directions the two navigation modes with and without GPS, thus comprising a total of 52 possible actions. Parameters in the GNC model are configured to ensure an action duration of 4 seconds (constant action cost $f_t = 4$). The collision cost is fixed at $K = 450$.

Evaluated action selection strategies several constant c values ($c = (0.01, 0.05, 0.1, 0.5, 1, 5, 10)$) were considered for the *UCB1* strategy (UCB(c)), while *Entropy-based Coefficient* (EBC) considered the interval $[c_{min}, c_{max}] = [0, 0.0222]$, so that c_{EBC} lies within an interval of $[0, 10]$, i.e. same range of coefficient values examined for the constant exploration coefficient approach. Additionally, the state-of-the-art approaches were tested: the adaptive coefficient that *Decays With Depth* (DWD) proposed by [21]; and, the approach explored by [13] (UCB$_{\sqrt{\cdot}}$) that requires extensive search for the definition of the exploration factor's value as in UCB1 and, therefore, is used only for the best UCB1 coefficient (UCB$^*_{\sqrt{\cdot}}$).

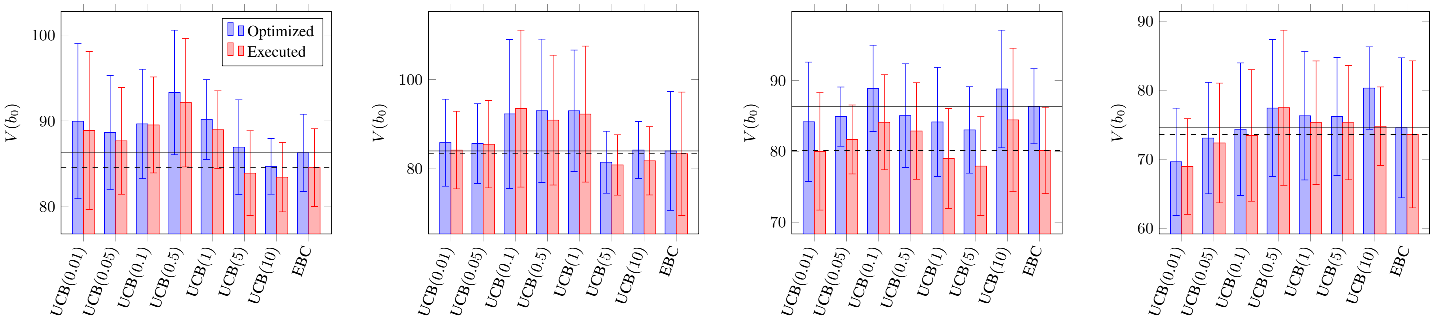
Evaluated backup strategies Classical POMCP backup value approximation was tested for each of the above-listed 4 action selection strategies, while MinPOMCP-GO backup value approximation was performed only for the EBC and for the coefficient value that showed the best results in UCB1.

Solving Ten value and policy optimizations were performed for each map configuration, action selection strategy and backup strategy. The total number of trials (always starting from b_0) for each optimization process is set to 50000. Moreover, 1000 simulations were performed to evaluate the policy being currently optimized.

Evaluation Metrics The evaluation metrics include the value of the initial belief state $V(b_0)$ of the optimized policy, the value of the initial belief state reached $V(b_0)$ (executed) during the simulations, the success rate (in %), the average flight time T for the successful simulations and the computation time per optimization process.

Results Figures 4 and 5 compare the average (and standard deviation) results of the metrics $V(b_0)$, optimized (after the 50000 trials) and executed (1000 simulations), for the several fixed coefficient values for the UCB1 strategy with the EBC strategy. Recalling that these experiments use the Classical POMCP backup value approximation. Moreover, Table 1 presents the average (and standard deviation) of the success rates, the average flight time T (s) and the computation time (min) achieved after 50000 trials for the best fixed UCB1 coefficient, EBC, DWD and UCB$^*_{\sqrt{\cdot}}$ action selection strategies. And finally, Figures 6 and 7 compare the average values of the best UCB1 fixed coefficient (UCB*) and the EBC action selection strategy, for both backup value strategies. These average results are also summarized in Table 1.

The results obtained allow to draw some observations. Firstly, it is noticeable that the UCB1's best fixed coefficient varies significantly (see Figures 4-5), not only across the different probabilistic availability GPS grid maps considered, but also when changing initial belief



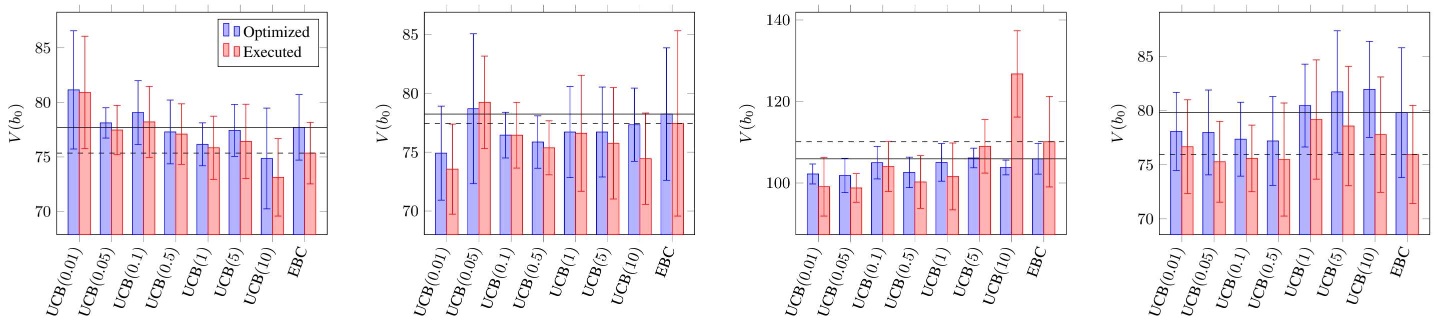
(a) Initial belief state mean position (blue cross) at $(10, 25, 5)m$, for a probability grid map considering 2-meter GPS precision.

(b) Initial belief state mean position (blue cross) at $(10, 25, 5)m$, for a probability grid map considering 10-meter GPS precision.

(c) Initial belief state mean position (orange plus) at $(50, 25, 5)m$, for a probability grid map considering 2-meter GPS precision.

(d) Initial belief state mean position (orange plus) at $(50, 25, 5)m$, for a probability grid map considering 10-meter GPS precision.

Figure 4: Results obtained for *WallBaffle* map, for the several UCB1 fixed coefficient values and the EBC approach, using the classical POMCP backup value approximation.



(a) Initial belief state mean position (blue cross) at $(35, 20, 5)m$, for a probability grid map considering 1-meter GPS precision.

(b) Initial belief state mean position (blue cross) at $(35, 20, 5)m$, for a probability grid map considering 2-meter GPS precision.

(c) Initial belief state mean position (orange plus) at $(65, 20, 5)m$, for a probability grid map considering 1-meter GPS precision.

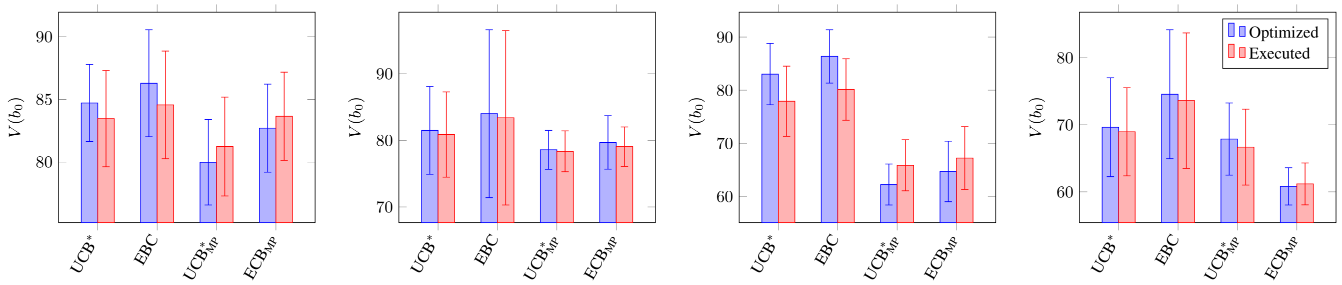
(d) Initial belief state mean position (orange plus) at $(65, 20, 5)m$, for a probability grid map considering 2-meter GPS precision.

Figure 5: Results obtained for *CubeBaffle* map, for the several UCB1 fixed coefficient values and the EBC approach, using the classical POMCP backup value approximation.

Table 1: Performance comparison between action selection strategies and backup value approximation. The data is organized as Average (Standard deviation). In **bold** are represented the best values of the three metrics (success rate, flight time and computation time per optimization process) for each backup value approximation. The MinPOMCP-GO backup value approximation results are identified with an extra subscript MP.

		WallBaffle, $2m$ GPS precision		*WallBaffle*, $10m$ GPS precision		*CubeBaffle*, $1m$ GPS precision		*CubeBaffle*, $2m$ GPS precision	
	Initial position	$(10, 25, 5)m$	$(50, 25, 5)m$	$(10, 25, 5)m$	$(50, 25, 5)m$	$(35, 20, 5)m$	$(65, 20, 5)m$	$(35, 20, 5)m$	$(65, 20, 5)m$
UCB1*	Success (%)	97.42 (1.09)	**97.35 (1.45)**	98.45 (1.83)	98.54 (1.12)	98.65 (0.54)	**92.44 (0.81)**	97.59 (0.85)	97.76 (1.48)
	$T\,(s)$	**73.75 (0.81)**	**67.80 (3.33)**	75.05 (2.52)	63.32 (3.98)	67.97 (3.66)	70.06 (0.96)	64.77 (1.38)	66.88 (1.56)
	Time per opt. (min)	153.03 (4.50)	**80.80 (9.23)**	88.33 (3.47)	65.30 (8.08)	155.33 (1.25)	**94.34 (8.29)**	66.34 (3.82)	173.33 (2.36)
EBC	Success (%)	97.59 (1.02)	97.04 (0.85)	97.19 (3.28)	97.64 (2.51)	98.02 (0.58)	89.34 (2.89)	96.82 (1.42)	**98.35 (0.67)**
	$T\,(s)$	75.54 (1.42)	68.87 (2.95)	**72.81 (1.08)**	64.51 (2.55)	**67.78 (1.64)**	**69.58 (0.91)**	65.23 (3.58)	69.67 (2.46)
	Time per opt. (min)	**113.80 (3.34)**	119.60 (4.29)	75.45 (4.63)	79.33 (2.05)	**110.45 (17.39)**	177.30 (5.09)	98.67 (3.94)	86.67 (9.42)
DWD	Success (%)	**98.68 (1.04)**	96.52 (1.26)	**99.91 (0.08)**	**99.69 (0.45)**	**99.57 (0.67)**	86.91 (3.32)	**99.66 (0.52)**	95.39 (3.36)
	$T\,(s)$	76.46 (0.88)	69.80 (3.88)	77.34 (2.31)	69.82 (3.50)	70.58 (1.64)	70.10 (2.10)	70.64 (3.48)	69.66 (3.67)
	Time per opt. (min)	196.00 (28.26)	266.51 (29.06)	101.67 (4.71)	164.90 (17.55)	233.75 (25.34)	356.34 (4.32)	199.30 (34.32)	285.76 (18.90)
$\text{UCB}^*_{\sqrt{(\cdot)}}$	Success (%)	97.99 (0.90)	96.69 (1.42)	99.00 (0.78)	97.83 (1.50)	97.96 (0.82)	91.51 (1.77)	96.69 (1.30)	97.43 (0.74)
	$T\,(s)$	74.71 (1.58)	67.97 (3.93)	73.85 (1.06)	**62.40 (3.03)**	68.12 (3.41)	70.46 (0.89)	**64.69 (2.61)**	**66.54 (1.07)**
	Time per opt. (min)	203.30 (14.73)	144.50 (28.32)	**66.33 (2.62)**	**55.56 (3.82)**	333.67 (10.53)	407.53 (8.99)	**51.33 (6.18)**	**53.36 (1.70)**
$\text{UCB1}^*_{\text{MP}}$	Success (%)	**98.25 (0.94)**	98.28 (1.03)	99.00 (0.89)	98.35 (1.25)	**97.20 (0.66)**	97.11 (0.43)	**97.70 (1.64)**	98.09 (1.26)
	$T\,(s)$	**74.68 (1.24)**	**59.13 (1.33)**	**74.59 (1.52)**	60.25 (1.72)	65.08 (0.74)	66.32 (0.98)	64.37 (1.60)	**65.23 (0.61)**
	Time per opt. (min)	277.67 (9.67)	161.00 (17.47)	124.30 (4.61)	86.34 (3.47)	199.33 (7.59)	**113.75 (4.97)**	106.67 (1.25)	215.42 (4.08)
EBC_{MP}	Success (%)	97.81 (0.80)	**98.29 (1.19)**	**99.07 (0.92)**	**99.75 (0.31)**	96.89 (0.87)	**97.49 (0.53)**	97.27 (1.76)	**99.36 (0.95)**
	$T\,(s)$	75.46 (1.71)	60.57 (2.07)	75.56 (2.15)	**60.21 (2.31)**	**64.41 (0.80)**	**66.10 (0.21)**	**63.70 (0.81)**	65.67 (0.82)
	Time per opt. (min)	**119.01 (6.31)**	**98.30 (4.63)**	**101.20 (5.35)**	**85.67 (2.87)**	**107.66 (6.13)**	121.63 (1.25)	**97.68 (3.09)**	**100.03 (9.27)**

state mean positions in the same map. It confirms that the most suitable value for the exploration coefficient to be used in the UCB1 formula varies with the planning problem, and cannot be determined in general. Additionally from Table 1, one can verify that even the single best fixed coefficient in UCB1 takes more computational effort than the EBC in some cases, meaning that the whole extensive



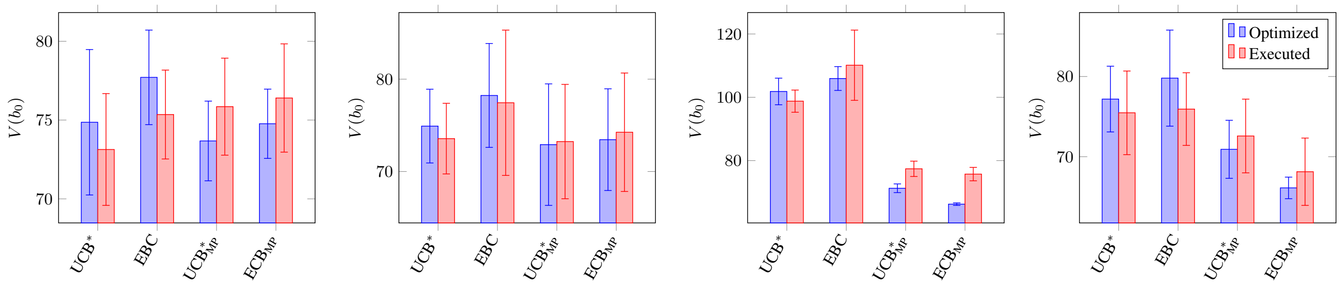
(a) Initial belief state position (blue cross) at $(10, 25, 5)m$, for a probability grid map considering 2-meter GPS precision.

(b) Initial belief state mean position (blue cross) at $(10, 25, 5)m$, for a probability grid map considering 10-meter GPS precision.

(c) Initial belief state mean position (orange plus) at $(50, 25, 5)m$, for a probability grid map considering 2-meter GPS precision.

(d) Initial belief state mean position (orange plus) at $(50, 25, 5)m$, for a probability grid map considering 10-meter GPS precision.

Figure 6: Comparison between the results obtained for *WallBaffle* map, using the classical POMCP backup value approximation and the MinPOMCP-GO backup value approximation (MP subscript). For both cases, the best UBC1 (UCB*) and ECB coefficients were considered.



(a) Initial belief state mean position (blue cross plus) at $(35, 20, 5)m$, for a probability grid map considering 1-meter GPS precision.

(b) Initial belief state mean position (blue cross) at $(35, 20, 5)m$, for a probability grid map considering 2-meter GPS precision.

(c) Initial belief state mean position (orange plus) at $(65, 20, 5)m$, for a probability grid map considering 1-meter GPS precision.

(d) Initial belief state mean position (orange plus) at $(65, 20, 5)m$, for a probability grid map considering 2-meter GPS precision.

Figure 7: Comparison between the results obtained for *CubeBaffle* map, using the classical POMCP value backup approximation and the MinPOMCP-GO backup value approximation (MP subscript). For both cases, the best UBC1 (UCB*) and ECB coefficients were considered.

search process could be computationally heavy. For this reason, the use of a coefficient that can be dynamically tuned to fit all the different planning problems becomes a very promising solution to avoid the extensive search. From Figures 4 and 5, one can verify that both $V(b_0)$ (optimized and executed) computed for EBC lie within the values obtained for the fixed coefficients, although never reaching the values of the best fixed coefficient.

Table 1 shows that both DWD and UCB$^*_{\sqrt{\cdot}}$ strategies do not perform with a consistent behaviour across the different scenarios, yielding the best results for particular case studies, while the worst results for others. Furthermore, DWD is on average the heaviest computational taking the longest time to minimize the flight time. Therefore, a compromise must be made between the desired quality of the results and the time one is willing to spend on the exhaustive search for the best UCB1 coefficient value. Moreover, it is worth to say that for an online planning configuration (real flights), such exhaustive search is not suitable (or even not feasible). Thereby, an adaptive coefficient, such as ECB, should propose a promising applicable solution.

When applying the backup value approximation proposed, MinPOMCP-GO (see Figures 6 and 7), a substantial improvement on the initial belief state value approximation is verified in all cases studied, except for the *CubeBaffle* blue position with probabilistic GPS availability grid map for 1-meter precision. In which, the executed $V(b_0)$ is higher for both coefficients. The values registered in Table 1 follow this outcome, as the success rate also increases with the use of this value backup strategy for all cases, except for the same case. Additionally, the combination of EBC with the MinPOMCP-GO strategy is a promising approach for online planning, since it offers no need for extensive coefficient value search, yields better success rates and accelerates value convergence, when compared with the best fixed coefficient that needs to be defined before flight.

5 Conclusion and future work

This paper presents a novel adaptive coefficient, called EBC, based on a measure of the uncertainty about action outcomes (e.g. probabilistic sensors' availability onboard a UAV), to be used during action selection in the POMCP-GO algorithm. Such an adaptive coefficient pushes the algorithm to explore more where uncertainty is higher. Comparative results show this adaptive coefficient approach, which avoids the extensive parameter tuning associated with the UCB1 formula, guarantees a satisfying level of performance across different planning scenarios considered in this particular study case of safe urban vehicle navigation. In our view, this approach is more suited for an online planning configuration, dismissing the need of previous parameter tuning, because it only relates on the model probability functions. Moreover, this paper also proposes the combination of the Entropy-based adaptive coefficient with an alternative backup value approach, resulting in better success rates and convergence values on the initial belief state in most study cases considered.

Further work will study the possibility of the adaptive coefficient EBC approach generalization for Monte-Carlo Tree Search based algorithms to enrich UCB1 action selection strategy. It includes a wide range of probabilistic planning models (MDP, POMDP, etc) and solvers (e.g. UCT [11] and variants [17]). On another hand, the online planning configuration of the MinPOMCP-GO algorithm exploiting the EBC action selection strategy will also be evaluated in a planning while executing paradigm (such as [24]) during simulated and real flights.

REFERENCES

[1] Auer, P., Cesa-Bianchi, N., and Fischer, P., ”Finite-time Analysis of the Multiarmed Bandit Problem”, *Machine Learning 47(2-3):235-256*, 2002.

[2] Delamer, J.-A., Watanabe, Y., and Chanel, C. P. C., ”MOMDP solving algorithms comparison for safe path planning problems in urban environments”, *9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2017.

[3] Kleijer, F., Odijk, D., and Verbree, E., ”Prediction of GNSS availability and accuracy in urban environments case study Shiphol airport”, *Location Based Services and TeleCartography II*, Springer, 2009.

[4] Watanabe, Y., Dessus, S., and Fabiani, S., ”Safe Path Planning with Localization Uncertainty for Urban Operation of VTOL UAV”, *AHS Annual Forum*, 2014.

[5] Achtelik, M. W., Lynen, S., Weiss, S., Chli, M., and Siegwart, R., ”Motion- and Uncertainty-aware Path Planning for Micro Aerial Vehicles”, *Journal of Field Robotics*, 2014.

[6] Delamer, J.-A., Watanabe, Y., and Chanel, C. P. C., ”Solving path planning problems in urban environments based on a priori sensor availability and execution error propagation”, *AIAA Scitech 2019 Forum*, 2019.

[7] Ong, S. C., Png, S. W., Hsu, D., and Lee, W. S., ”Planning under uncertainty for robotic tasks with mixed observability”, *The International Journal of Robotics Research*, 2010.

[8] Araya, M., Thomas, V., Buffet, O., and Charpillet, F., ”A closer look at MOMDPs”, *22nd International Conference on Tools with Artificial Intelligence - ICTAI*, 2010.

[9] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., ”Planning and acting in partially observable stochastic domains”, *Artificial Intelligence*, 1998.

[10] Silver, D., and Veness, J., ”Monte-Carlo planning in large POMDPs”, *Advances in neural information processing systems*, 2010.

[11] Kocsis, L., and Szepesvri, C., ”Bandit based Monte-Carlo planning”, *Machine Learning: ECML*, 2006.

[12] Chaslot, G. M. J.-B., Winands, M. H. M., van den Herik, H. J., Uiterwijk, J. W. H. M., and Bouzy, B., ”Progressive Strategies for Monte-Carlo Tree Search”, *New Mathematics and Natural Computation 04(03):343-357*, 2008.

[13] Tolpin, D., and Shimony, S., ”MCTS based on Simple Regret”, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[14] Rosin, C., ”Multi-armed Bandits with Episode Context”, *Annals of Mathematics and Artificial Intelligence, 61(3):203-230*, 2010.

[15] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., ”Mastering the Game of Go with Deep Neural Networks and Tree Search”, *Nature. 529. 484-489*, 2016.

[16] Vien, N., and Toussaint, M., ”Hierarchical Monte-Carlo Planning”, *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[17] Keller, T., and Helmert, M., ”Trial-based Heuristic Tree Search for Finite Horizon MDPs”, *ICAPS 2013 - Proceedings of the 23rd International Conference on Automated Planning and Scheduling*, 2013.

[18] Kozelek, T., ”Methods of MCTS and the game Arimaa”, Masters thesis, Charles University in Prague, 2009.

[19] Kolobov, A. and Mausam, *Planning with Markov decision processes: An AI perspective*, Vol. 6, Morgan & Claypool Publishers, 2012.

[20] Dijkstra, Edsger W., ”A note on two problems in connexion with graphs”, *Numerische mathematik 1.1*, p. 269271, 1959.

[21] Delamer, J.-A., *Planification de stratégies de navigation et de guidage pour des drones autonomes dans des milieux encombrés* (Unpublished doctoral dissertation). ISAE-SUPAERO, Université de Toulouse, 2019.

[22] Mettler, B., Kong, Z., Goerzen, C., and Whalley, M., Benchmarking of obstacle field navigation algorithms for autonomous helicopters, 66th Forum of the American Helicopter Society: ”Rising to New Heights in Vertical Lift Technology”, AHS Forum 66, 2010. URL: `www.aem.umn.edu/people/mettler/projects/AFDD/AFFDwebpage.htm`.

[23] Bonet, B., and Geffner, H., ”Solving POMDPs: RTDP-Bel versus point-based algorithms. *In Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[24] Chanel, C. P. C., Albore, A., THooft, J., Lesire, C., Teichteil-Knigsbuch, F., ”AMPLE: an anytime planning and execution framework for dynamic and uncertain problems in robotics”. *Autonomous Robots*, 43(1), p. 37-62. 2019.