Balancing Exploration and Exploitation in Reinforcement Learning

Youssef Achbany\textsuperscript{1}, Francois Fouss\textsuperscript{1}, Luh Yen\textsuperscript{1}, Alain Pirotte\textsuperscript{1} & Marco Saerens\textsuperscript{1}

\textsuperscript{1}Information Systems Research Unit (ISYS), IAG, Université catholique de Louvain, Place des Doyens 1, B-1348 Louvain-la-Neuve, Belgium
\{achbany, fouss, yen, pirotte, saerens\}@isys.ucl.ac.be

Abstract. In this work, we present a model that integrates both exploration and exploitation in a common framework. First of all, we define the concept of degree of exploration from a state as the entropy of the probability distribution on the set of admissible actions in this state. This entropy value allows to control the degree of exploration linked to this state, and should be provided by the user. Then, we restate the exploration/exploitation dilemma as a global optimization problem: define the best exploration strategy that minimizes the expected cumulated cost, while maintaining fixed degrees of exploration. This formulation leads to a set of nonlinear updating rules reminiscent from the “value iteration” algorithm. Interestingly enough, when the degree of exploration is zero for all states (no exploration), these equations reduce to Bellman’s equations for finding the shortest path while, when it is maximum, a full “blind” exploration is performed. The theoretical results are confirmed by simple simulations showing that the model behaves as expected.

1. Introduction

Balancing exploration and exploitation is an important issue in reinforcement learning. Exploration aims to continually try new ways of solving the problem, while exploitation aims to capitalize on already well-established solutions. Exploration is of course especially important when the environment is changing, i.e. non-stationary. In this case, good solutions can deteriorate over time, or new, better, solutions can appear over time. Without exploration, the system only sends agents along the best, optimal, path, without exploring alternative paths. It would therefore not be aware of the changes, and its performances would inevitably deteriorate over time. One of the key features of reinforcement learning is that it explicitly considers the exploration/exploitation problem in an integrated way; this is one of its specificities [17], in contrast with, for instance, Markov decision processes [4], [5], [14].

Usually, one tackle this exploration problem by using a probabilistic approach for selecting actions (choice randomization). In other words, it is common to assign a probability distribution on the set of admissible actions so that, at some fixed interval of time, an action will be chosen stochastically with, however, a clear preference for “good solutions” [12]. This way, the system readjusts its policy periodically by exploring up-to-now sub-optimal actions. Generally, the agent chooses action \( u_j \) with a probability that is a function of the immediate cost associated with this action (usually, a softmax action selection; see [12], [17]). However, as shown in this paper, this
strategy is sub-optimal because it is accounted for in an ad-hoc manner: it does not inte- 
grate the exploration issue within the optimization process. Stated in other words, the same optimization strategy is used whether or not exploration is performed.

The objective of this paper is precisely to propose a model that integrates both 
exploration and exploitation in a common framework. We therefore assume that the 
environment (both the costs and the states) is continuously explored, while minimizing 
the current average cost to destination. In order to simplify the analysis, we concentrate 
on a “stochastic shortest-path problem”, as defined in [5] and described later in the 
next section; we are currently working on extensions to “average cost per state” as 
well as full stochastic problems. To this end, we define the concept of **degree of 
exploration** from a state as the entropy [10] related to the probability distribution 
of the set of admissible actions on this state.

This entropy value allows to control the degree of exploration linked to this state, 
and is provided a priori by the user. When the entropy is zero, no exploration is per-
formed from this state, while when the entropy is maximal, a full, blind, exploration, 
with equal probability of choosing any action, is performed. Then, we restate the ex-
ploration/exploitation dilemma as a global optimization problem: define the best ex-
ploration strategy (the probability distribution of choosing an action in a given state) 
that minimizes the expected cumulated cost from the initial state while maintaining a 
fixed degree of exploration. This problem leads to a set of nonlinear equations defin-
ing the optimal solution. These equations, which are similar to Bellman’s equations, 
can be solved by iterating them until convergence, which is proved for a particular 
initialization strategy. They provide the action policy (the probability distribution of 
choosing an action in a given state) that minimizes the average cost from the initial 
state to the destination, or terminal, state, for a given degree of exploration.

Interestingly enough, when the degree of exploration is zero for all states, the 
nonlinear equations reduce to Bellman’s equations for finding the shortest path from 
the initial state to the destination (solution) state. On the other hand, when the degree 
of exploration is maximum for all states, the nonlinear equations reduce to the linear 
equations allowing to compute the average cost for reaching the solution from the 
initial state in a Markov chain with transition probabilities equal to the inverse of the 
number of admissible actions for each state (full blind exploration).

The main drawback of this method is that it is computationally demanding since it 
relies on iterative algorithms alike to the value iteration algorithm. We however 
show in [1] that if the graph of states is directed and acyclic, the equations can easily 
be solved by performing a single backward pass from the destination state. One way 
to obtain such a directed, acyclic, graph is to use Dial’s procedure, popular in the 
transportation networks field [8].

Section 2 introduces the notations, the standard stochastic shortest-path problem 
and the way we manage exploration. Section 3 describes our procedure for solving 
the stochastic shortest-path problem with exploration. Some numerical examples are 
shown in section 4, while section 5 is the conclusion.
2. Statement of the problem and notations

In this work, we analyze reinforcement learning problems involving exploration. For the sake of simplicity, we will concentrate here on what is called “stochastic shortest-path problems”.

In a stochastic shortest-path problem, we assume that during every state transition a bounded cost, given by $C(u(s_t)|s_t)$, is incurred where $s_t$ is a variable that contains the current state at time $t$ and $u(s_t)$ contains a control action selected from a set of admissible actions, or choices, $U(s_t)$, available in state $s_t$. The cost $C(u(s_t)|s_t)$ can be viewed as the cost of performing action $u(s_t)$ given that the agent is in state $s_t$, and is assumed to be bounded. It can be positive (penalty), negative (reward) or zero provided that no cycle exists whose total cost is negative. This is a standard requirement in shortest paths problems [6]; indeed, if such a cycle exists, then following this cycle an arbitrary large number of times will result in a path with an arbitrary small cost so that a best path is not properly defined. This implies that if the graph of the states contains nondirected arcs, these cannot have negative costs.

The control action $u(s_t)$ at any time $t$ is determined according to a policy $\pi$ that maps every state $k$ on the set of admissible actions $U(k)$ with a certain probability distribution, $P(u(k)|s = k)$, with $u(k) \in U(k)$. Thus the policy associates to each state $k$ a probability distribution on the set of admissible actions $U(k)$: $\pi \equiv \{P(u(1)|s = 1), P(u(2)|s = 2), \ldots \}$ with each $u(k) \in U(k)$. For instance, if the admissible choices in state $k$ are $U(k) = \{u^k_1, u^k_2, u^k_3\}$, the probability distribution $P(u(k)|s = k)$ involves three values, $P(u(k) = u^k_1|s = k)$, $P(u(k) = u^k_2|s = k)$, and $P(u(k) = u^k_3|s = k)$. Under a so-called stationary policy $\pi$, the control depends only on the current state $s$.

The decision process is thus randomized because, for each state, the agent chooses the next action according to the probability distribution $P(u(k)|s = k)$. Randomization is introduced in order to guarantee a given degree of exploration that will be controlled by the entropy of the probability distribution; randomized choices are very common in a variety of fields; for instance game theory (called mixed strategies in this context; see for instance [13]) or decision sciences [15]. We do not authorize, however, that an agent returns to the initial state; in other words, we do eliminate all the control actions that let us return to the initial state $k_0$.

Moreover, we assume here that once the action has been chosen, the next state $s_{t+1}$ is fixed univocally in a deterministic way, $s_{t+1} = f(u(s_t)|s_t)$ where $f$ is a function. It provides the next state $s_{t+1}$, given that we are in state $s_t$ and we choose action $u(s_t)$. Notice that, since each action automatically leads to a precise state, the introduction of control actions could be avoided by directly computing state transition probabilities. The more general formalism introduced here is justified by the fact that we will generalize this work to full stochastic problems for which, once a specific action is chosen, the transition to a state occurs according to some probability distribution.

The goal is to minimize

$$V_\pi(s = k_0) = E_\pi \left[ \sum_{t=0}^{\infty} C(u(s_t)|s_t) | s_0 = k_0 \right]$$

which is the total expected cost accumulated over an infinite horizon, given the initial (or source) state $k_0$. The expectation is taken on the policy, that is, on all the random variables $u(k)$ associated to the states $k$. 


As in Bertsekas [5], we assume that there is a special cost-free destination state (the solution state); once the system has reached that state, it remains there at no further cost. We also consider a problem structure such that termination is inevitable, at least under an optimal policy. Thus, the horizon is in effect finite, but its length is random and may be affected by the policy being used. The conditions for which this is true are, basically, linked to the fact that the destination state can be reached in a finite number of steps from any potential initial state; for a rigorous treatment, see [3], [5].

2.1. Computation of the total expected cost for a given policy

The essence of the problem is thus to reach the destination state \( s = d \) with minimal expected cost from the initial state \( k_0 \); as already stated before, this problem is usually called the stochastic shortest-path problem. The deterministic shortest-path problem is obtained as a special case where, for each state, the control action is determined univocally.

This problem can be represented as a Markov chain where each original state and each action is a state. The destination state is then considered as an absorbing state with no outgoing link. In this framework, the problem of computing the expected cost (2.1) from any state \( k \) is closely related to the computation of the average first-passage time in the associated Markov chain [11]. The average first-passage time is the average number of steps a random walker starting from the initial state \( k_0 \) will take in order to reach destination state \( d \), and can be easily generalized in order to take the cost of the transitions into account. By first-step analysis (see for instance [18]), we show in [1] that, once the policy is fixed, \( V_\pi(k) \) can be computed through the following equations

\[
\begin{aligned}
V_\pi(s = k) &= \sum_{i \in U(k)} P(u(k) = i | s = k) \left[ C(u(k) = i | s = k) + V_\pi(s = k') \right], \\
V_\pi(s = d) &= 0, \text{ where } d \text{ is the destination state}
\end{aligned}
\]  

(2.2)

Thus, in equation (2.2), \( k' \) is the state resulting from the application of control action \( u(k) = i \) in state \( k \). In the sequel, we will use the following shortcuts for the different quantities of interest: \( V_\pi(s = k) = V_\pi(k) \), \( C(u(k) = i | s = k) = C(i | k) \), \( f(u(k) = i | s = k) = f(i | k) \) and \( P(u(k) = i | s = k) = P(i | k) \).

(2.2) is a system of linear equations that can be solved by iterating the equations or by inverting the so-called fundamental matrix [11]; it is analogous to Bellman’s equations in Markov decision processes.

Notice that the same framework can easily handle multiple destinations problems by defining one absorbing state for each destination. If the destination states are \( d_1, d_2, \ldots, d_m \), we have

\[
\begin{aligned}
V_\pi(k) &= \sum_{i \in U(k)} P(i | k) \left[ C(i | k) + V_\pi(k') \right], \\
\text{where } k' &= f(i | k) \text{ and } k \neq d_1, d_2, \ldots, d_m \\
V_\pi(d_j) &= 0 \text{ for all destination states } d_j, j = 1, \ldots, m
\end{aligned}
\]  

(2.3)
We now have to address two questions: (1) how do we control the randomized choices, i.e. exploration, and (2) how do we compute the optimal policy for a given degree of exploration.

### 2.2. Controlling exploration by fixing entropy at each state

Now that we have introduced the problem, we will explain how we manage exploration. At each state $k$, we define the **degree of exploration** $E_k$ by

$$E_k = - \sum_{i \in U(k)} P(i|k) \log(P(i|k)) \quad (2.4)$$

which is simply the entropy of the probability distribution of the control actions in this state [7], [10]. This degree of exploration may vary from state to state, and is provided by the user. It measures the uncertainty about the choice at each state and is equal to zero when there is no uncertainty at all (the distribution $P(i|k)$ reduces to a Kronecker delta), or is equal to $\log(n_k)$, where $n_k$ is the number of admissible choices at node $k$, in the case of maximum uncertainty, $P(i|k) = 1/n_k$ (a uniform distribution).

Furthermore, the **exploration rate** $E_r^*_{k}$ at state $k$ is defined as the ratio between the degree of exploration and the maximum degree for that state, $E_r^*_{k} = E_k / \log(n_k)$, and takes its values in the interval $[0, 1]$. Fixing the entropy at a state sets the exploration level from this state; increasing the entropy increases exploration up to the maximal value, in which case there is no more exploitation since the next action is chosen completely at random, with a uniform distribution, without taking the costs into account.

### 3. Optimal policy with exploration constraints

#### 3.1. Optimal policy and expected cost

We now turn to the problem of determining the optimal policy under exploration constraints. More precisely, we will seek the policy, that is, the probability distributions of the actions within the states, $\pi = \{P(u(1)|s = 1), P(u(2)|s = 2), \ldots\}$, for which the expected cost $V^*_\pi(k_0)$ from initial state $k_0$ is minimal while maintaining a given degree of exploration on the states, $E_k$. Before going into the details, we make the following important preliminary observation: whatever the chosen policy, the system will remain a Markov chain and consequently Equation (2.2) remains valid for any admissible policy. The problem is thus to find the transition probability leading to the minimal expected cost, and can be formulated as a constrained optimization problem involving a Lagrange function.

In [1], we derive the optimal probability distribution of control actions within a state $k$, which appears to be a logit distribution:

$$P(i|k) = \frac{\exp[-\theta_k (C(i|k) + V^*(k'_0))]}{\sum_{j \in U(k)} \exp[-\theta_k (C(j|k) + V^*(k'_0))]}, \quad (3.1)$$

where $k'_i = f(u(k) = i|s = k)$ is a following state and $V^*$ is the optimal (minimum) expected cost given by

$$
\begin{align*}
V^*(k) &= \sum_{i \in U(k)} P(i|k) [C(i|k) + V^*(k'_i)], \text{ with } k'_i = f(i|k) \text{ and } k \neq d \\
V^*(d) &= 0, \text{ for the destination state } d
\end{align*}
$$

(3.2)

In Equation (3.1), $\theta_k$ must be chosen in order to satisfy

$$
\sum_{i \in U(k)} P(i|k) \log(P(i|k)) = -E_k
$$

(3.3)

for each state $k$, and takes its values in $[0, \infty]$. This last equation guarantees a fixed exploration level (entropy) at each state. Of course if, for some state, the number of possible control actions reduces to one (no choice), no entropy constraint is introduced.

Equation (3.1) has a nice appealing interpretation: choose preferably (with highest probability) action $i$ leading to the state $k'_i$ of lowest expected cost, including the cost of performing the action, $C(i|k) + V^*(k'_i)$. Thus, the agent is routed preferably to the state which is nearest (in average) to the destination state.

Since the Equation (3.3) has no analytical solution, $\theta_k$ must be computed numerically in terms of $E_k$. This is in fact quite easy since it can be shown that the function $\theta_k(E_k)$ is strictly monotonic decreasing, so that a line search algorithm (such as the bisection method, see [2]) can efficiently find the $\theta_k$ corresponding to a $E_k$.

### 3.2. Computation of the optimal policy

Equations (3.1) and (3.2) suggest an iterative procedure very similar to the well-known value iteration algorithm for the computation of both the expected cost and the policy.

More concretely, we consider that agents are sent from the initial state and are choosing an action in each state $k$ with probability distribution $P(u(k)|s = k)$, according to the current policy. The agent then performs the chosen action, say action $i$, and observes the associated cost, $C(i|k)$ (which, in a non-stationary environment, may vary over time), together with the new state, $k'$. This allows him to update the cost, the policy and the average cost until destination estimates; these estimates will be denoted by $\hat{C}(i|k)$, $\hat{P}(i|k)$ and $\hat{V}(k)$.

1. **Initialization phase:**
   - Choose (and fix) any initial policy, $\hat{P}(i|k), \forall i, k$, verifying the exploration rate constraints (3.3) and
   - Compute the corresponding expected cost until destination by using an iterative procedure for solving the set of linear equations (2.2). Any standard asynchronous iterative procedure (for instance, a Gauss-Seidel like algorithm) for computing the expected cost until absorption in a Markov chain could be used (see [11]).

2. **Computation of the policy and the expected cost under exploration constraints:**
   For each visited state $k$, do until convergence:
– Choose an action \( i \) with current probability estimate \( \hat{P}(i|k) \), observe/update the current cost \( \hat{C}(i|k) \) for performing this action, and jump to the next state, \( k'_i \).

– Update the probability distribution of the state \( k \) by:

\[
\hat{P}(i|k) \leftarrow \frac{\exp \left[ -\theta_k \left( \hat{C}(i|k) + \hat{V}(k'_i) \right) \right]}{\sum_{j \in U(k)} \exp \left[ -\theta_k \left( \hat{C}(j|k) + \hat{V}(k'_j) \right) \right]},
\]

(3.4)

where \( k'_i = f(u(k) = i|s = k) \) and \( \theta_k \) is set in order to respect the prescribed degree of entropy (see Equation (3.3)).

– Update the expected cost asynchronously:

\[
\begin{align*}
\hat{V}(k) &\leftarrow \sum_{i \in U(k)} \hat{P}(i|k) [\hat{C}(i|k) + \hat{V}(k'_i)], \text{ with } k'_i = f(i|k) \text{ and } k \neq d \\
\hat{V}(d) &\leftarrow 0, \text{ where } d \text{ is the destination state}
\end{align*}
\]

(3.5)

The convergence of these updating equations is proved in [1]. However, the described procedure is computationally demanding since it relies on iterative procedures alike to the value iteration algorithm in Markov decision processes.

Notice also that, while the initialization phase is necessary in our convergence proof, other simpler initialization schemes could also be applied. For instance, set initially \( \hat{C}(i|k) = 0 \), \( \hat{P}(i|k) = 1/n_k \), \( \hat{V}(k) = 0 \), where \( n_k \) is the number of admissible actions in state \( k \); then proceed by directly applying updating rules (3.4) and (3.5). While convergence is not proved in this case, we observed that this updating rule works well in practice; in particular, we did not observe any convergence problem – it is indeed this rule that will be used in our experiments.

### 3.3. Some limiting cases

We will now show that when the degree of exploration is zero for all states, the nonlinear equations reduce to Bellman’s equations for finding the shortest path from the initial state to the destination (solution) state.

Indeed, from Equations (3.4)-(3.5), if the parameter \( \theta_k \) is very large, which corresponds to a near-zero entropy, the probability of choosing the action with the lowest value of \( (\hat{C}(i|k) + \hat{V}(k'_i)) \) dominates all the others. In other words, \( \hat{P}(u(k) = j|s = k) \approx 1 \) for the action \( j \) corresponding to the lowest average cost (including the action cost), while \( \hat{P}(u(k) = i|s = k) \approx 0 \) for the other alternatives \( i \neq j \). Equations (3.5) can therefore be rewritten as

\[
\begin{align*}
\hat{V}(k) &\leftarrow \min_{i \in U(k)} [\hat{C}(i|k) + \hat{V}(k'_i)], \text{ with } k'_i = f(i|k) \text{ and } k \neq d \\
\hat{V}(d) &\leftarrow 0, \text{ where } d \text{ is the destination state}
\end{align*}
\]

(3.6)

which are Bellman’s updating equations for finding the shortest path to the destination state [4], [5].
On the other hand, when $\theta_k = 0$, the choice probabilities reduce to $\hat{P}(i|k) = 1/n_k$, and the degree of exploration is maximum for all states. In this case, the nonlinear equations reduce to the linear equations allowing to compute the average cost for reaching the destination state from the initial state in a Markov chain with transition probabilities equal to $1/n_k$. In other words, we then perform a “blind” exploration, without taking the costs into consideration.

Any intermediary setting $0 < E_k < \log(n_k)$ leads to an optimal exploration vs. exploitation strategy minimizing the expected cost, and favoring short paths to the solution. In [1], we further show that if the graph of states is directed and acyclic, the nonlinear equations can easily be solved by performing a single backward pass from the destination state.

4. Simulation results

Our experiments were performed on a graph composed of 14 nodes connected by edges of different weights, representing costs.

The algorithm described in this paper is used in each experiment. It searches an optimal path in a given graph, by going from a starting node to a destination node, while fixing the exploration rate. We will investigate how the algorithm reacts when we vary the value of the entropy (exploration rate), and the impact of these values on the total expected cost.

Two simple experiments are performed, testing the algorithm’s capacity to find the optimal paths in two different settings: (1) a static environment (i.e. an environment where the weight of the edges does not change over time); (2) a dynamic (non-stationary) environment.

In the first experiment, we analyze the paths used by the algorithm to convey agents from one node to another in a static environment, fixing various values for the exploration rate at each node. In the second experiment, we observe how the algorithm reacts in a dynamic environment, when the weights of the edges vary over time.

In each experiment, the matrices of expected costs and transition probabilities are updated at each transition according to Equations (3.4) and (3.5) while a step
represents the complete routing of an agent from its source to its destination. At the
beginning of each simulation, the $V(k)$ and $P(i|k)$ are initialized according to the
procedure introduced at the end of section 3.2. The exploration rate is fixed to a
specified value common to each state.

4.1. First experiment

Description. During this first experiment, we will send 15,000 agents through the
network shown in Figure 4.1. The initial and destination nodes are node 1 and node
13. The costs of the external edges (i.e. the edges on the paths $[1,2,6,12,13]$ and
$[1,5,9,10,14,13]$) are initialized to 1, while the costs of all the other edges are initialized
to 2.

The goal of this simulation is to observe the paths followed by the routed agents, in
function of various values for the entropy. We simulate the agents transfer by assigning
the same value of the exploration rate at each node. Each node corresponds to a state
and the actions correspond to the choice of the next edge to follow. We repeat the
experiment four times with, for each of them, a fixed value of the entropy corresponding
to a percentage of the maximum entropy (i.e. exploration rates of 0%, 30%, 60% and
90%).

Results. The following graphs, displayed in Table 4.1, show the behaviour of the
algorithm when varying the values of the exploration rate. For each tested value, a
graph (with the same topology as described above) is used to represent the results: the
more an edge is used in the routing, the more its grey level and width are important.

The first graph shows the results when sending the 15,000 agents by using an
exploration rate fixed at 0%. When exploration rate is zero, the algorithm finds the
shortest path from node 1 to node 13 (path $[1,2,6,12,13]$), and no exploration is carried
out: $[1,2,6,12,13]$ is the only path used for sending the agents.

The second graph shows the followed paths for an entropy of 30% of the maximum
value for each node. The path $[1,2,6,12,13]$, corresponding to the shortest path, is still
the most used, while other edges not belonging to the shortest path are now also
investigated. Moreover, we observe that the second shortest path, i.e. path $[1,5,9,10,14,13]$ also
conveys a significant traffic here. Notice however that, despite the rise of explo-
ration, the path mostly used by the algorithm remains the shortest path. Exploitation
thus remains a priority while exploration is nevertheless present.

The third graph shows the paths used for an exploration rate of 60%. As for the
previous experiment, the algorithm uses more and more edges that are not part of the
shortest path, but still prefers the shortest path.

The last graph shows the result with an exploration rate of 90%. With such a
value, the exploitation of the shortest path is no more the priority; exploration is now
clearly favored over exploitation.

4.2. Second experiment

Description. In this second experiment, we will reiterate the previous simulations
on the same graph with various values of the exploration rate. The difference lies in
the dynamic aspect of the environment used in this experiment. Indeed, the goal of
Table 4.1. Traffic through the edges for the four different exploration rates. We clearly observe that the exploration progressively increases with the exploration rate.

this second experiment is to analyze the behaviour of the algorithm when a change in the environment occurs (i.e. a change in the cost of some edges).

Except the change in the environment, the context remains the same as in the first experiment, namely to convey 15,000 agents from a source node (1) to a destination node (13) by using the suggested algorithm. Thus, at each time step, an agent is sent from node 1 to reach node 13. At the beginning, the costs of the external edges (i.e. the edges on the paths [1,2,6,12,13] and [1,5,9,10,14,13]) are initialized to 3, while the cost of all the others (internal edges) are initialized to 6. In this configuration, the shortest path from node 1 to node 13 is the path [1,2,6,12,13] with a total cost of 12.

After having sent 7,500 agents (i.e. the middle of the simulation), the costs of all the internal edges were set to 1, all other things being equal. This change creates new shortest paths, all of them with a total cost of 4, passing through internal nodes [3,4,7,8,11].

Results. The first figure (Figure 4.2) contains five graphs representing the total costs of the transfer of all agents from their source to their destination, averaged on the first 7500 agents. We display the results for five levels of exploration rate: 0%, 10%, 20%, 30% and 50%.

The first graph (on the bottom) shows the total cost for an exploration rate of 0%. We observe that, despite the environment change (which leads to the creation of shorter paths with a total cost of 4), the cost remains equal to 12 throughout simulation. This is due to the fact that, once the algorithm has discovered the shortest path, it does not explore anymore.
Fig. 4.2. Average cost needed to reach destination node (13) from the initial node (1) in terms of time step. We observe that when no exploration is performed (exploration rate of 0%), the system misses a path that becomes (at time step 7500) much lighter (in terms of cost) than the optimal one up to now.

Fig. 4.3. Average, maximal and minimal costs needed to reach the destination node (13) from the initial node (1) in terms of exploration rate.

The four remaining graphs show the results for exploration rates of 10%, 20%, 30% and 50%. Each of these settings is able to find the new optimal path after the change in environment – the total cost is updated from about 12 to 4. This is due to the fact that the algorithm carries out exploration and exploitation in parallel. This also explains the slight rise in total cost when increasing the value of the entropy.

The second figure (Figure 4.3) shows the evolution of the average cost, the maximal cost as well as the minimal cost (computed on the first 7,500 agents) in terms of the exploration rate. Increasing the entropy induces a growth in the average total costs, which is completely foreseeable since the rise in this entropy causes an enhancement in exploration, therefore producing a reduction in the exploitation and increasing the average total costs. We also notice that the difference between the minimum and the maximum total cost grows with the increase in entropy but remains quite weak.
5. Conclusions
In this work, we presented a model integrating both exploration and exploitation in a common framework. The exploration rate is controlled by the entropy of the choice probability distribution defined on the states of the system. When no exploration is performed (zero entropy on each node), the model reduces to the common value iteration algorithm computing the minimum cost policy. On the other hand, when full exploration is performed (maximum entropy on each node), the model reduces to a “blind” exploration, without considering the costs. Further work will aim to investigate full “stochastic shortest-path” problems, as well as alternative cost formulation, such as the “average cost per step”. Moreover, since the optimal average cost is obtained by taking the minimum among all the potential policies, it can be shown that it defines a distance measure between the states of the process. Further work will investigate the properties of this distance, which generalizes the Euclidean commute time distance between nodes of a graph, as introduced and investigated in [9], [16].

References