Probabilistic Guided Exploration for Reinforcement Learning in Self-Organizing Neural Networks

Peng Wang*†, Weigui Jair Zhou*, Di Wang‡ and Ah-Hwee Tan*‡
*School of Computer Science and Engineering,
‡Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly,
Nanyang Technological University, Singapore
†School of Electrical and Information Engineering, Tianjin University, Tianjin, China
Email: ytwp327@gmail.com, {jairzhou,wangdi,asahtan}@ntu.edu.sg

***Abstract*—Exploration is essential in reinforcement learning, which expands the search space of potential solutions to a given problem for performance evaluations. Specifically, carefully designed exploration strategy may help the agent learn faster by taking the advantage of what it has learned previously. However, many reinforcement learning mechanisms still adopt simple exploration strategies, which select actions in a pure random manner among all the feasible actions. In this paper, we propose novel mechanisms to improve the existing knowledge-based exploration strategy based on a probabilistic guided approach to select actions. We conduct extensive experiments in a Minefield navigation simulator and the results show that our proposed probabilistic guided exploration approach significantly improves the convergence rate.**

***Index Terms*—Reinforcement learning, self-organizing neural networks, guided exploration**

I. Introduction

Exploration is of vital importance and necessity in reinforcement learning, wherein the agent chooses an action that might not be the optimal choice for exploration so as not to miss the opportunity of finding a competent solution.

In [1], the authors proposed three heuristic search-based exploration strategies for reinforcement learning. These three strategies record the outcomes of the performed actions in different states and subsequently use the learned knowledge to guide the exploration. In neighborhood search-based exploration, the agent selects an action from the neighborhood sets, including all the actions it has selected under this states. Simulated annealing-based exploration is similar to neighborhood search-based exploration, except reducing the chance of getting stuck in local optima. Tabu search-based exploration can help the agent to avoid the most recently visited action for the current state.

In [2], the knowledge-based exploration categorizes all the feasible actions into three non-overlapping groups: positive, negative and unexplored. Subsequently, the agent randomly selects an action from the reduced action space that comprise of the positive and unexplored actions. In order to take better advantage of what the agent has learned previously and further improve the efficiency during exploration, in this paper, we propose a novel knowledge-based exploration strategy named **probabilistic guided exploration**. We adopt a probabilistic guided approach to better select an action from the reduced action space. Furthermore, we assign each remaining action a probability, which is regulated by both the associated Q-value and the matching degree between the current situation and the identified knowledge. An action with higher Q-value and higher matching degree gets higher probability of being selected and executed subsequently. In addition, we also discuss how to leverage the weights associated to these two factors. The above-mentioned directed exploration strategies in [1], [2] are selected as the benchmarking models for the performance comparisons. The undirected exploration strategy with ϵ-greedy policy [3] is also selected as a benchmarking model for performance comparisons.

To realize our proposed probabilistic guided exploration approach for performance evaluations, we use a self-organizing neural network named Temporal Difference-Fusion Architecture for Learning and Cognition (TD-FALCON) [3]. TD-FALCON is a 3-channel fusion Adaptive Resonance Theory (ART) network that incorporates temporal difference methods into the ART models for reinforcement learning. The four mentioned directed exploration method and ϵ-greedy policy are similarly incorporated in TD-FALCON. Moreover, for benchmarking purposes, evaluation of performance are conducted using the Minefield navigation simulator [3]. Comparisons show that our proposed probabilistic guided exploration approach significantly improves the convergence rate.

II. Reinforcement Learning Model

Although our proposed probabilistic guided exploration strategy is applicable to various reinforcement learning algorithms, in this paper, we illustrate its usage through a self-organizing neural network model named TD-FALCON [3]. By learning three-dimensional mappings across state, action and reward in an online and incremental manner, TD-FALCON enables reinforcement learning of both value and action policies in a real time. Readers are directed to [3] for the detailed implementation.

In [2], the knowledge-based exploration strategy is incorporated in TD-FALCON to better use the previously learned knowledge. Specifically, after finding the most similar node to the current state-action pair, the reward can be represented as $R = \{L_J, U_J\}$, where J denotes the most similar node and $L_J = w_J^{c3}$, $U_J = 1 - w_J^{c3}$ denote the lower and upper bounds

of Q-values associated to node J, respectively. Subsequently, each feasible action at the current state can be dichotomized into either a positive, negative, or unexplored action as follows.

Definition 1: Action a is a positive action if the lower bound of its Q-value (for performing a at the current state s) $L_J > u^+$, where u^+ is a heuristically assigned threshold value to distinguish positive actions.

Definition 2: Action a is a negative action if the upper bound of its Q-value (for performing a at the current state s) $U_J < u^-$, where u^- is a heuristically assigned threshold value to distinguish negative actions.

Definition 3: Action a is an unexplored action, if it is neither a positive action nor a negative one. This assignment of "unexplored" implies that the reward information about the corresponding state-action pair is not deterministic, yet.

As a consequence, the reduced action space only consists of both the positive and unexplored actions. Therefore, during exploration, the agent randomly chooses an action only from the reduced action space. As such, the agent may save much effort in re-examining the actions that are known as leading to undesirable outcomes and the overall learning efficiency is expected to increase. The dynamics of the knowledge-based exploration are summarized in Algorithm 1.

Algorithm 1 Dynamics of knowledge-based exploration

Require: State s
1: **for** each feasible action a at s **do**
2: Based on the current $\{s, a\}$ pair, identify its most similar node and retrieve the associated reward r.
3: **if** this action is positive (see Definition 1) **then**
4: action $a \in A^+$;
5: **else if** this action is negative (see Definition 2) **then**
6: action $a \in A^-$;
7: **else**
8: action $a \in A^u$ (unexplored, see Definition 3).
9: **end if**
10: **end for**
11: Create the reduced action space $A^r = A^+ \cup A^u$.
12: Randomly select an action a from A^r for exploration.

III. Probabilistic Guided Exploration

As our contributions of this paper, we propose novel mechanisms to further take the advantage of previously learned knowledge when selecting an action from the reduced action space. Specifically, we propose two mechanisms to select an action from the reduced action space, wherein each action is associated with a probability proportional to its reward value and to a combination of both the reward value and its matching degree in the current situation.

A. Probability based on Q-value

For each state-action pair, the associated Q-value represents the estimated reward to be received if the corresponding action is performed at the corresponding state. With a higher Q-value, the corresponding action is expected to receive a better outcome. Following this rationale, in knowledge-based exploration, for all the actions in the reduced action space, their expected outcomes are still not the same. Therefore, instead of choosing a random action from the reduced action space with equal probability, it might be beneficial to select the action based on their expected rewards. Intuitively, we assign each action i in the reduced action space with a probability P_i^Q proportional to its Q-value, which is defined as follows:

$$P_i^Q = \frac{Q_i}{\sum_{j \in A^r} Q_j}, i \in A^r, \tag{1}$$

where A^r denotes the reduced action space (see Step 11 in Algorithm 1), and Q_i denotes the Q-value of action i.

B. Probability based on Both Q-value and Matching Degree

In reinforcement learning, due to the possible huge state space, state generalization is commonly performed, e.g., fuzzy AND operation is applied in FALCON [5] to group similar states together. Therefore, when evaluating the possible outcome of a state-action pair, the matching degree between the current situation and the retrieved learned knowledge should be taken into consideration other than the associated Q-value. Similar to (1), we define the probability P_i^M of selecting an action from the reduced action space A^r according to the aforementioned matching degree M_i as follows:

$$P_i^M = \frac{M_i}{\sum_{j \in A^r} M_j}, i \in A^r. \tag{2}$$

In different reinforcement learning algorithms, the matching degree may be computed in various ways. In this work, as we implement the probability guided exploration strategy using FALCON, we use its template matching function to compute the matching degree. In ART theory, the matching degree is computed to examine whether the current situation and the retrieved learned knowledge have a resonance, i.e., similar enough to be grouped together. Therefore, in this paper, M_i is computed as

$$M_i = \frac{|x \wedge w_i|}{|x|}. \tag{3}$$

where x represents the current situation and w_i is the state representation of action i of the retrieved learned knowledge.

To regulate the appropriateness of the selected action from A^r, M_i is considered in additional to Q_i. As aforementioned in this subsection that the learned knowledge, especially the state representation, is generalized in many reinforcement learning algorithms. Hence, only considering Q_i when assigning the selection probability may introduce bias if the current situation deviates much from the retrieved knowledge. In initial learning phase, few state-action pairs have been evaluated and the overall learned knowledge is far from convergence. It is highly likely that a selected high Q_i value does not represent the current situation well. In such cases, it is probably more appropriate for the agent to perform more random explorations, rather than only relying on the seemingly high Q-values that may not well represent the current situation.

Therefore, we design the following definition of the overall selection probability P_i, which simultaneously considers both the reward value and the matching degree:

$$P_i = \lambda P_i^M + (1 - \lambda) P_i^Q \tag{4}$$

where $\lambda \in [0, 1]$ denotes the leverage between the reward value and the matching degree.

C. Leverage between reward value and matching degree

As shown in (4), the overall probability of selecting an action from the reduced action space is regulated by the leverage parameter λ, which determines the relative weightage between the reward value and the matching degree. Here, we further discuss how to systematically determine the value of λ rather than setting it heuristically by trial-and-error.

Without doubt, Q-value is a dominating factor in action selection in the ϵ-greedy policy. When the agent chooses to exploit learned knowledge, it will perform the action that may possibly lead to the highest reward value at the current state. Moreover, as discussed in the previous subsection that it might be biased to select merely based on the Q-value in the initial phase. However, in the converging phase, to maximize the performance by greatly relying on the learned knowledge (towards pure exploitation), the agent should heavily rely on the Q-value to fine tune the learned knowledge. Therefore, the corresponding coefficient $(1-\lambda)$ should be initialized to a reasonable value and kept increasing to a maximal value when the exploration phase terminates.

On the other hand, the agent better considers the matching degree between the current situation and the learned knowledge in the initial reinforcement learning phase. However, as the agent progresses towards performing more exploitations than explorations, the state-action space gets discovered better. As such, the reliance on the matching degree should be decrease gradually. Therefore, the corresponding coefficient should be initialized to a reasonable value and kept decreasing to a minimal value when the exploration phase terminates.

As a result, the relationship between λ and $(1 - \lambda)$ is conceptually the same as the linear ϵ decay policy [6] that as ϵ gradually decays, the agent will perform more exploitations than explorations. Therefore, in this paper, we intuitively set λ to be the same value as ϵ.

IV. Experimental Results

In each experimental run, the Minefield navigation simulator [3] generates an emulating minefield map. The primary usage of this simulator is to examine the navigation capability of an autonomous vehicle (AV) travelling through the minefield to a randomly selected target location within a specified time frame without hitting any mine. In each trial, the AV starts from a randomly selected starting location in the minefield and repeats the cycles of sense, act, and learn. A trial ends when the AV reaches the target (success), hits a mine (failure), or exceeds 30 sense-act-learn cycles (time out). Note that the target location and the mine deployment locations remain stationary during each trial. All results reported in this paper are based on a

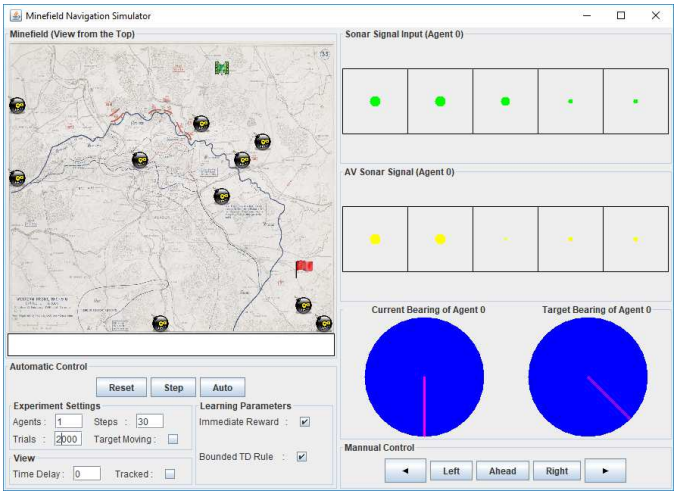

Fig. 1: Interface of the Minefield navigation simulator.

TABLE I: Parameter values of TD-FALCON

FALCON parameters for $k = 1, 2, 3$	
Choice parameters α^{ck}	0.1, 0.1, 0.1
Learning rates β^{ck}	1.0, 1.0, 1.0
Contribution parameters γ^{ck}	0.5, 0.5, 0.0
Vigilance parameters ρ_l^{ck} in learning	0.2, 0.2, 0.5
Vigilance parameters ρ_p^{ck} in performing	0.0, 0.0, 0.0
TD learning parameters	
Learning rate α	0.5
Initial Q-value (Immediate reward)	0.5
Discount factor γ	0.1

16 by 16 minefield map with 10 randomly located mines as illustrated in Fig. 1.

1) Input space in Minefield navigation simulator: An AV in the Minefield navigation simulator has a rather coarse sensory capability with a 180-degree forward view based on 5 sonar sensors that are 45 degrees apart from each other. For each direction of sonar i, its signal is measured by $s_i = 1/d_i$, where d_i is the distance to the nearest obstacle, i.e., a mine or the boundary of the minefield in the ith direction. Other input attributes of the sensory vector include the bearing of the target from the AV's current position. In each step, the AV can choose one out of the five possible actions, namely: move left, diagonally left, straight ahead, diagonally right, and right. At the end of a trial, a reward of 1 is given if the AV reaches the target. A reward of 0 is given if the AV hits a mine or the trial runs out of time. At each step of the trial, an estimated immediate reward is computed as:

$$r = \frac{1}{1 + rd} \tag{5}$$

where rd is the remaining distance between the current position to the target position.

2) Model parameter settings: The parameter values of TD-FALCON and those for probabilistic guided exploration are shown in Tables I and II, respectively. Note that our probabilistic guided exploration extends the action selection strategy defined by the knowledge-based exploration. As aforementioned, λ and its decay rate are set to the same values as ϵ and its decay rate.

TABLE II: Parameter values of probabilistic guided exploration

Knowledge-based exploration parameters	
Initial ϵ value	0.5
ϵ decay rate	0.0005
Positive action threshold u^+	0.55
Negative action threshold u^-	0.45

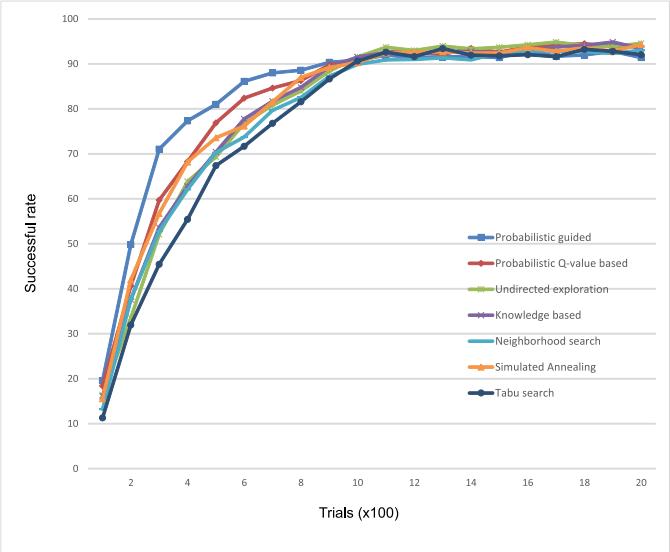

Fig. 2: Performance comparisons of various reinforcement learning models.

We use the Minefield navigation simulator to evaluate the performance of various directed or undirected reinforcement learning models, namely: (i) our proposed probabilistic guided exploration (see (4)), (ii) probabilistic Q-value based (see (1)), (iii) undirected exploration [3], (iv) knowledge based directed exploration [2], (v) neighborhood search-based exploration [1], (vi) simulated annealing-based exploration [1], and (vii) Tabu search-based exploration [1]. Furthermore, we run each model for 20 times and report the averaged mission successful rate in Fig. 2. As suggested in Table II that ϵ decays to 0 at the 1000th trial. Therefore, all the models learn from scratch and are expected to converge since 1000th trial.

As shown in Fig. 2, after the 1000th trial, all the models converge and achieve high successful rate around 90%, without significant difference. This observation suggests that in a stable environment such as the Minefield navigation simulator, all models are able to learn effectively after enough training.

The performance comparison before the 1000th trial is more distinguishable as shown in Fig. 2. It can be observed that our proposed probabilistic guided exploration strategy significantly outperforms the others in terms of learning speed. Moreover, our proposed probabilistic Q-value based exploration strategy also seemingly perform better than the other benchmarking models. When we look at the accumulated number of successful trials obtained by each model, at the 500th trial, our proposed models obtain at least 28% and 13% more number of successes than any other model does, respectively. The similar improvements at the 1000th trial are 17% and 8%, respectively. These findings suggest that our proposed probabilistic based approaches learn efficiently in the exploration phase.

As shown in Fig. 2, the performance of knowledge-based exploration has no significant difference from the performance of the original TD-FALCON (undirected exploration). The most probable reason is due to the difference in the application domain that the knowledge-based exploration was applied to. To be more specific, in the Pursuit-Evasive problem defined in [2], a pursue agent has eight choices (360 degree movement) as feasible actions to choose from. When the target is in the forward direction of the pursue agent, by using the exploration strategy, four or five actions can be excluded from the reduced action space (see Algorithm 1). However, in the Minefield navigation problem, an agent only has five actions (180 degree movement), and on average it can only exclude one negative action as revealed from the investigation on the experimental results. Therefore, different application domain may have affected the performance of the knowledge-based exploration strategy.

V. Conclusion

In this paper, we proposed a novel probabilistic Q-value based exploration strategy, which only takes the actions' Q-values into consideration when selecting an action from the reduced action space, and a novel probabilistic guided exploration strategy, which takes both the Q-value and the matching degree into consideration. The experimental results show that our probabilistic guided exploration outperforms the other benchmarking models in terms of learning speed. Going forward, we plan to apply our proposed strategies to other more challenging problem domains.

Acknowledgment

This research is supported in part by the DSO National Laboratories under research grant DSOCL16006. This research is also supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative.

References

[1] N. A. Vien, N. H. Viet, S. G. Lee, and T. C. Chung, "Heuristic search based exploration in reinforcement learning," in *Lecture Notes in Computer Science*, vol. 4507, 2007, pp. 110–118.

[2] T.-H. Teng and A.-H. Tan, "Knowledge-based exploration for reinforcement learning in self-organizing neural networks," in *Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2012, pp. 332–339.

[3] A.-H. Tan, N. Lu, and X. Dan, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 230–244, 2008.

[4] T.-H. Teng, A.-H. Tan, and J. Zurada, "Self-organizing neural networks integrating domain knowledge and reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 889–902, 2015.

[5] D. Wang, B. Subagdja, and A.-H. Tan, "Creating human-like autonomous players in real-time first person shooter computer games," in *Proceedings of Conference on Innovative Applications of Artificial Intelligence*, 2009, pp. 173–178.

[6] D. Wang and A.-H. Tan, "Creating autonomous adaptive agent in a real-time first-person shooter computer game," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 2, pp. 123–138, 2015.