








# Q-Learning in a Multidimensional Maze Environment

Oscar Chang<sup>1,2</sup> , Stadyn Román Niemes<sup>1</sup> , Washington Pijal<sup>1</sup> ,  
Arianna Armijos<sup>1,3</sup> , and Luis Zhinin-Vera<sup>1,2</sup> 

<sup>1</sup> School of Mathematical and Computational Sciences, Yachay Tech University,  
100650 Urcuquí, Ecuador

{ochang, stadyn.roman, washington.pijal, arianna.armijos,  
luis.zhinin}@yachaytech.edu.ec

<sup>2</sup> MIND Research Group - Model Intelligent Networks Development,  
Urcuquí, Ecuador

<sup>3</sup> LoUISE Research Group, I3A, University of Castilla-La Mancha, Albacete, Spain

**Abstract.** Experiments with rodents in mazes demonstrate that, in addition to visual cues, spatial localization and olfactory sense play a key role in orientation, foraging and eventually survival. Simulation at some level and understanding of this unique behavior is important for solving optimal routing problems. This article proposes a Reinforcement Learning (RL) agent that learns optimal policies for discovering food sources in a 2D maze using space location and olfactory sensors. The proposed Q-learning solution uses a dispersion formula to generate a cheese smell matrix  $S$ , tied in space time to the reward matrix  $R$  and the learning matrix  $Q$ . RL is performed in a multidimensional maze environment, in which location and odor sensors cooperate in making decisions and learning optimal policies for foraging activities. The proposed method is computationally evaluated using location and odor sensor in two different scenarios: random and Deep-Search First (DFS), showing positive results in both cases.

**Keywords:** Q-learning · Agent · Multi-dimensional environment · Maze solving

## 1 Introduction

In the brain of a real-world rat trying to find a food source in a difficult maze, a great deal of parallel data processing occurs. Even if the maze is brought into total darkness, the animal will still go about its daily survival routines of foraging, shifting its attention to senses other than sight such as the sense of place [9] and the sense of smell [6]. It is evident that in total darkness the rat will keep its learning ability intact and will be able to quickly learn a strategy that defines its decision-making behavior and optimizes its path to the food source, using only place and smell sense. Although some interesting theories have been established over the years, no one knows exactly how the sense of place operates in the rat brain [1, 9, 16].

The rat’s sense of smell is highly complex. It has been demonstrated that the utilization of stereo cues is critical for the detection of odor sources, as a rat can distinguish whether an odor is coming from the right or left in only 50 milliseconds with just one sniff. The rat’s olfactory system appears to satisfy both the independent sampling and neural mechanisms criteria for stereo odor localization. According to the scientists, smelling in stereo has a number of evolutionary advantages, including the ability to swiftly detect the presence of a predator or prey with high precision [22,33]

On the other hand, in a totally dark environment, other senses come into play, such as the use of whiskers, since rats have a rather poor vision system. Whiskers change direction and allow the rat to move quickly in places it already knows or to explore new territories if the environment is new [2]. One type of neurons in the hippocampus are activated, the so-called *place cells* that respond maximally when the animal is in a specific location in an environment [17]. From these studies it is concluded that with little visual information, the rat activates the senses of localization and smell to the maximum.

In terms of computation, the rats quickly learn a decision-making policy that optimizes their way from anywhere in the maze to the food supply using only position (place) and odor detection information, even in complete darkness.

This paper proposes an extended 2D maze model in which a new dimension of odor is introduced to the environment in addition to the traditional location information (R coordinate matrix). The aim is to construct RL agents that emulate the learning behavior of rats operating in complete darkness while also incorporating the senses of “place” and “smell”. A dispersion formula provides a cheese odor gradient in the maze space, which serves as the odor dimension. Additionally, the agent is equipped with odor sensors that are assembled into a gradient detection mechanism which complements an olfactory system.

For the implementation of this approach, the maze is one of the most important parts since using a modified Q-learning strategy generates an ideal scenario for the agent to learn to optimize routes and generate the expected results.

## 1.1 Q-learning

The Q-learning algorithm is a well known Reinforcement Learning technique first introduced in 1989 by J. Watkins for solving the Markov decision problems with incomplete information. It works with an agent in an environment that has to learn an action-value function that gives the expected utility for taking a given action in a given state [19]. It can also be thought of as an asynchronous dynamic method (DP). In other words, it enables agents to learn how to act optimally in Markovian domains by observing the effects of their actions instead of requiring them to build domain maps [32].

This agent-environment duo is widely used in data structures, educational algorithms, and research [3]. In this paper, an agent uses Q-learning to learn an efficient strategy by exploring and using place and odor sensors in a coordinated manner. As usual in this algorithm, exploration requires taking into account future events during the reward capture process.

**Table 1.** Overview of the main related works results that suggest solutions to solve the shortest path (STP) problem with Q-Learning, a RL algorithm.

Proposed method	Problem	Main results	Reference
Multi-Q-table Q-learning	Shortest path (STP) problem in a maze with considerable sub-tasks such as gathering treasures and evading traps	Manage the trouble of the lower average sum of compensations	[13]
$\epsilon$ -Q-learning	Slowly convergence speed during the location of the optimal paths in a given environment	The suggested $\epsilon$ -Q-Learning can find out more useful optimal paths with lower costs of searching, and the agent successfully evade all barriers or traps in an unfamiliar environment	[8]
ERTS-Q	The interaction between the environment and the agent for collecting real experiences is time-consuming and expensive	An adaptive tree structure integrating with experience replay for Q-Learning called ERTS-Q	[11]
Q-learning	Loss of resources to solve mobile robot maze	The robot can locate the briefest way to solve the maze	[12]
Multi-agent DQN system (N-DQN) model	Characteristics and conditions that are associated with the performance of reinforcement learning	N-DQN offers approximately 3.5 times more elevated learning performance compared to the Q-Learning algorithm in the reward-sparse environment in the performance evaluation	[14]
Improved Q-learning (IQL)	Even though numerous studies report the successful execution of Q-learning, its slow convergence related to the curse of dimensionality could restrict the performance in practice	The suggested techniques accelerate the learning speed of Improved Q-learning (IQL) compared to traditional Q-learning	[18]
A algorithm and Q-learning	Path planning for wheeled mobile robots on somewhat understood irregular landscapes is challenging since robot motions can be affected by landscapes with insufficient environmental information, such as impassable terrain areas and locally detected obstacles	The experimental results and simulation demonstrate that the developed path planning approach provides paths that bypass locally detected impassable areas and obstructions in a somewhat known irregular terrain	[34]
Bees Algorithm (BA) and Q-learning algorithm	Discover an optimal path in a two-dimensional environment for a mobile robot	The experimental results on various maps to validate the suggested method in the static and dynamic cases demonstrate the effectiveness and robustness of the presented method in discovering the optimal path	[4]

## 2 Related Work

Several ideas and methods have emerged from the study of biological agents' behavior and self-learning capacity in agent research, based on Q-learning and agent automatic learning in an unknown environment.

In [15] it is shown how a group of children learn from a totally unknown environment taking into account 2 conditions: Exploring freely into the maze and find a reward within the maze. Children were divided into 3 groups related to low, medium and high explorers. It was found that the later achieved a high percentage of exploration of the environment and a better performance when searching for rewards. An important conclusion is that children seem to have and innate behavior oriented toward DFS algorithm.

In the work of [23] the behavior of 20 different agents (mice) was analyzed. The mice were kept inside a cage next to a labyrinth. Half group have food and water at all times, while the rest were deprived from them. With this research it was appreciated how a biological agent is able to learn an efficient strategy from an unknown environment with the help of experience and exploration.

In previous work [5] a robotic structure that imitates an amino acid chain was proposed. Subsequently an agent uses reinforcement learning to explore new forms of folding that will lead toward rewards in terms of energy stability. Here the combination of two sensors, self bending and nearby molecules forces, is used by the agent to learn how to fold into proteins looking shapes. The agent was implemented with neural networks with a noise balance training algorithm.

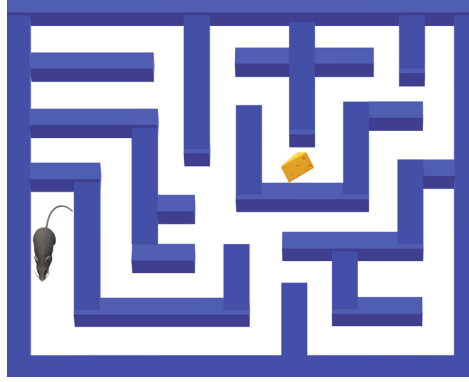
In addition to these approaches, Table. 1 presents an overview of the most representative recent articles that use Q-Learning to solve maze and optimization problems. An initial search in Scopus using the keywords: "Maze, RL, Q learning" gave us 93 documents, of which 16 have been published in the last three years. This demonstrates the scientific interest in developing this type of work.

## 3 Methodology

The methodology to develop this approach requires an adequate generation of learning scenarios, the implementation of search algorithms and the RL approach for the agent to explore and learn.

### 3.1 Maze Design: Environment

A maze is a puzzling way that consists of a different branch of passages where the agent intends to reach his destination by finding the most efficient route in the shortest possible time. By definition, the agent can only move in 4 quadrants (up, down, left, and right), and walls cannot be passed through. The agent is evaluated to be the best in this procedure based on the least amount of time or steps required to reach the destination. There were various studies to automatically execute maze search even before reinforcement learning was discovered and explored. Figure 1 shows an illustration of a maze design.



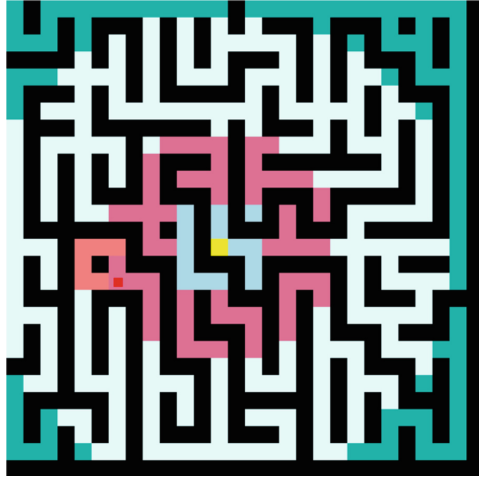
**Fig. 1.** Graph of a 3D maze showing the proposed idea represented in this experiment, where as an illustration, the agent using RL is the rat and the target is represented by a cheese.

There are many peculiarities to consider when simulating this scenario. In real world, environments are multidimensional; for example, in a real-world cheese maze, a piece of cheese will react instantaneously with the surrounding air molecules, and a gradient of cheese odor will eventually penetrate the entire volume of the maze due to natural rules governing the environment [21]. On the other hand, odor information is hard to process because its high-dimensional data, and large amounts of computation are required to distinguish or separate odors. Odors are made up of a variety of odorant molecules (about 200–400 thousand). Most odor discrimination devices have been built for specific odors to decrease the dimensions of odor information. [20,27].

Rats, on the other hand, have an intrinsic ability to extract information from their environment using a highly developed set of sensors, including a strong sense of smell and olfactory gradients, which complicates the experiment. These rodents utilize it to improve their abilities to locate food sources fast in a maze [7]. It's important to note that if the food source is steady, the rat will utilize its sense of smell to develop a policy that optimizes its path to the largest reward in the shortest time possible [31]. This is the type of scenario addressed in this work, in which odor plays an important part in the agent's self-learning process.

To simulate the environment, we create a matrix  $R$  that depicts the maze's space as well as the distribution of rewards. Then, using an exponential decay algorithm, a new dimension is provided by simulating the dispersion of the cheese (reward) odor throughout the maze. This dispersion formula can be as complicated as needed, and it can even incorporate a time variation.

Odor information is stored in a matrix  $S$  that has the same dimension as and is bounded in space-time by the matrix  $R$ . The greatest value of odor in  $S$  is in the same row-column location as the cheese in this arrangement. When agent is so far away from the cheese, the intensity of the odor reduces exponentially. Figure 2 shows a graphical representation of the environment with colors representing the odor intensity.



**Fig. 2.** In the context of the analyzed environment, this 2D graphical representation shows the odor gradient, the cheese (yellow) and the agent (red). (Color figure online)

For the purpose of this work a  $28 \times 28$  cell maze is used, where the agent has to learn efficient policies using an expanded version of Q-learning where the sense of location (place) and the sense of smell cooperate to learn efficient food location policies. The agent's learning capabilities are tested with Random Search and Deep-Search First (DSF) modalities [24, 30]. Finally, both implementations' performance and outcomes are analyzed and compared.

### 3.2 Depth-First Search (DFS)

Reachability in a directed graph is frequently determined using depth-first search in sequential algorithms. A depth-first spanning tree is built by recursively exploring all successors from a given vertex. Each vertex is marked before visiting its successors to avoid looping, and a marked vertex is not searched again [28].

DFS is a technique for traversing a graph that uses a last-in, first-out (LIFO) scheme and a stack as the underlying data structure [26]. Following the LIFO concept, insertion (push) and removal (pop) are performed at the top or front [10].

DFS on a graph with  $n$  vertices and  $m$  edges takes  $O(m + n)$  runtime. DFS traversal starts at one vertex and branches out to corresponding vertices until it reaches the final or destination point. DFS traversal of a graph performs the following [26]:

- Visits all vertices and edges of G
- Determine whether G is connected

- Computes the connected components
- Computes the spanning forest of  $G$

This algorithmic approach together with a random search are used in this work to compare the effectiveness of our approach in combination with the new dimension we incorporated: odor.

### 3.3 How the Agent Explores and Learns

Efficient maze solving plays a key role in some branches of Artificial Intelligence [25]. The Q-learning algorithm, in particular, is an effective way for enabling agents to capture rewards and learn an optimal policy in maze path solutions. The agent main goal is to interact with the environment (maze) by trial and error, and use evaluative feedback systems (rewards and penalties) to achieve decision-making optimization [29].

The learning process begins after the multidimensional environment has been prepared. The first step is for the agent to begin exploring and determining the optimal policies by itself. This is accomplished by a modified version of the conventional Q-learning method, in which the agent takes input from both the  $R$  and  $S$  matrices to make a choice. The outcome of these choices is stored in a  $Q$  matrix, which finally becomes the optimal policy.

In order to fill the knowledge matrix  $Q$ , the Bellman equation is used, which is defined as:

$$Q(s, a) = R(s, a) + \gamma \cdot \max_{a'} Q(s', a') \quad (1)$$

The concept is that when the agent finds the cheese,  $Q$  gets filled depending on the immediate reward in  $R$  as well as the highest possible reward from  $Q$  based on future states. The *gamma* parameter, often defined as the discount rate, determines the contribution of future steps.

In our model, the smell matrix  $S$  is the one that supplies the data that will be used, thereby transforming it into a reward matrix that considers odor gradient. As a result, the Bellman equation is as follows:

$$Q(s, a) = S(s, a) + \gamma \cdot \max_{a'} Q(s', a') \quad (2)$$

Using both Random Search and Depth-First Search algorithms, the impacts of having a new odor dimension in the maze environment and an improved agent with odor sensing skills are evaluated. When odor capacities are activated, the agent now considers data from  $R$  or  $S$  in its decision-making, and the learning process becomes more efficient and closer to biological processes.

The Random Search algorithm is an adaptation of the original method with minor changes. Normally, Random Search would be unconcerned by odor, but in this case, with a gradient to take, the agent's behavior is more greedy and odor-oriented. To prevent gradient traps, the agent's decision-making is also Markovian. When the odor gradient is insufficient to cover the entire maze, the agent reverts to random decision-making. In this way, the agent's search strategy

---

**Algorithm 1:** Odor Random Search Pseudocode
 

---

```

do
  if there is no odor around then
    | Move to a random neighbor.
  else
    | Search the best neighbor according to  $S$ .
    | Move to that neighbor.
  end
  Update  $Q$  according to (2).
while reward not captured;

```

---

devolves into a random search aided by odor. Algorithm 1. shows the algorithm that controls the behavior of this type of agent.

In the case of the DFS approach, the algorithm performs as expected, that is, it creates the DFS path and visited lists. The main loop then utilizes DFS and the odor to determine where to move, then executes the move and stores the information. This is shown in Algorithm 2.

---

**Algorithm 2:** Odor DFS Main Pseudocode
 

---

```

Initialize the path list  $P$ .
Initialize the visited list  $V$ .
Put the agent in a random starting position.
do
  Use Odor DFS to get the next move.
  Make the move.
  Search the best neighbor (according to  $Q$ ).
  Update  $Q$  according to (2).
while reward not captured;

```

---

The environment, the path and visited sets, the knowledge matrix, and the agent's initial position are used by the Odor DFS method. The ideal route for solving the maze determined by the algorithm is  $P$ , whereas  $V$  represents all the cells visited by the agent during the procedure. This method functions similarly to a standard DFS, with the exception that it makes decisions using the  $S$  matrix and chooses the agent's next step in a markovian way rather than traversing the entire maze at once. Algorithm 3 illustrates this approach.

### 3.4 Implementation Details

The algorithm was implemented using C++ and the Borland C++ graphic libraries. The code was run in an Intel Core i5 processor of 10th generation @ 1.00 GHz. The programs used for the project are hosted in this GitHub repository: <https://github.com/StadynR/q-learning-multidim-maze>.



**Algorithm 3:** Odor DFS Pseudocode

---

```

Save the current position in both  $P$  and  $V$ .
Initialize the unvisited list  $U$ .
Get the unvisited neighbors from the current position and save them in  $U$ .
if  $U$  is empty then
    | Remove the current position from  $P$ .
    | Backtrack to get the next move.
else
    | Search the best unvisited neighbor according to  $S$ .
    | Get the next move from the previous step.
end
return the next move.

```

---

## 4 Results

The simulation was performed with and without the odor gradient to determine the efficiency of the additional dimension. This means that the following four scenarios were explored: Random Search, Odor Random Search, DFS, and Odor DFS. In the non-odor cases, equation (1) was used to fill  $Q$ , while in the opposite cases, Equation (2) is used. For every case, the simulation was run in a total of 10 instances. An instance is the period of time that the agent takes to learn, i.e., the time in which the  $Q$  matrix is stabilized (does not change between iterations). Total execution time and total steps were used to determine the duration of the instances.

**Table 2.** Results of the runs of the algorithms: Random Search, Odor Random Search, DFS, and Odor DFS.

Instance	Random Search		Odor R. Search		DFS		Odor DFS	
	Time (s)	Steps	Time (s)	Steps	Time (s)	Steps	Time (s)	Steps
1	4023.777	135869	1540.11	44341	<b>134.283</b>	<b>3710</b>	208.202	5667
2	3427.708	115328	1648.718	45788	311.473	9730	262.389	7656
3	<b>1566.016</b>	<b>59395</b>	3257.644	56041	235.346	6673	208.767	5594
4	5010.259	192674	1508.734	42475	270.83	7807	228.717	6431
5	3932.311	126950	1755.688	47486	147	4211	273.018	7746
6	6673.376	184338	1665.358	46345	285.029	8948	129.092	3656
7	6403.862	215333	932.847	<b>25821</b>	220.893	6569	205.488	5859
8	4941.812	150560	<b>860.624</b>	28770	274.576	7783	<b>78.892</b>	<b>2267</b>
9	2683.155	94152	1420.08	39247	259.253	7747	259.264	7392
10	4120.635	137179	1869.393	53112	240.848	6848	289.283	8236

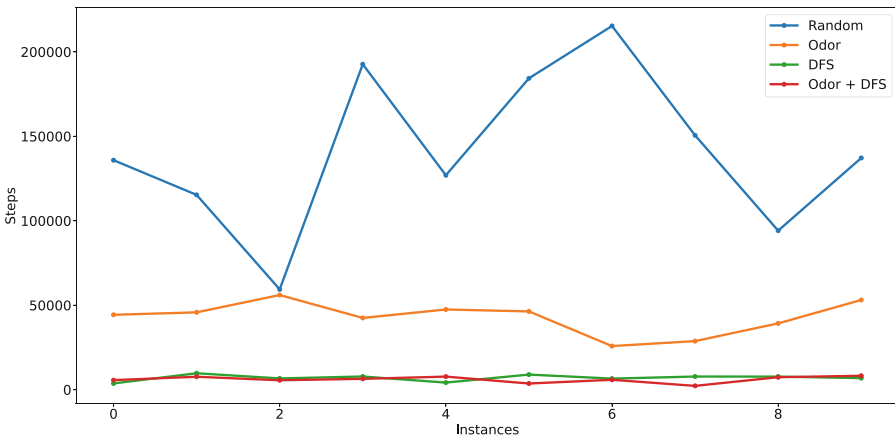
Table 2 shows the results when the agent uses Random Search to fill the  $Q$  matrix. It takes an average of 141178 steps and 4278.291s to reach a stable

$Q$  matrix. The values obtained are very high due to the fact that the agent uses a totally Random Search. In the same table, the improvement in learning efficiency when the agent uses Random Search supported by the Odor gradient (Odor Random Search), which information is taken from the  $S$  matrix. It takes an average of 42942 steps and 1645.92s to reach a stable  $Q$  matrix. In this search, the results improve greatly due to the odor gradient that guides the agent.

When the agent uses DFS to fill the  $Q$  matrix, the utilized stop criteria is the same as the rest. It takes an average of 7002 steps and 237.953s to stabilize the matrix  $Q$ . With this type of search, the agent has a great advantage over all the previous techniques. It is one of the most stable compared to the others.

In addition, the improvement in learning efficiency when the agent uses DFS assisted by the odor gradient is shown in the same table (Odor DFS). This technique presents the lowest average in both number of steps and seconds to achieve stabilization of the  $Q$  matrix, with 6050 steps and 214.311 s. This last type of search turns out to be the best of all, giving very low search averages, therefore being the fastest method for learning.

Finally, Fig. 3 shows the comparison between the four types of search, taking as parameters the instances and the steps needed to stabilize the  $Q$  matrix in each of them. It is clearly noticeable that the most unstable results are obtained when the agent explores its environment in a random manner. At the same time there is a great difference between pure random search with the other three methods, with Odor DFS being the most stable and appropriate technique for the agent studied.



**Fig. 3.** Comparison of the evolution of instances vs steps of the different results of the executed algorithms.

## 5 Discussion

As can be observed from the results, introducing an odor matrix aids the agent's decision-making while also complicating it. In particular, if we compare each pair, we can clearly see the improvement. The difference between random search and odor random search is significant, with odor random search learning 3 times faster (using average steps as a measure) than its non-odor counterpart. Odor random search appears to be a viable option for more complex maze traversal strategies.

When we look at the DFS-odor pair, the improvement is a small but significant 1.16 ratio. It's astonishing that odor matrix information can increase the performance of a top contender, given that DFS is one of the most efficient search algorithms known, used by biological entities with millions of years of evolution.

Random search (both odor and non-odor) frequently stabilizes  $Q$  in less episodes than DFS, which is worth noting. This is because DFS is highly direct and prioritizes speed above maze coverage, whereas random search usually ends up traversing the majority, if not all, of the maze cells. This suggests that random search fills  $Q$  with more information in a single episode than DFS. Even so, on a wide scale, this fact is immaterial since, while DFS requires more episodes, the steps and time spent in each episode are significantly lower than in random search, making DFS solutions a clear winner.

In general, it is evident that adding odor as a new dimension not only allows for more realistic maze models to be created, but it also improves the agent's behavior and learning speed. In fact, by including more matrices into the model, additional dimensions can be added to the simulation.

## 6 Conclusions

This paper added a new point of view to the classical *rat in a maze* scenario, in which an agent must learn a policy that optimizes reward capturing paths during exploitation. The additional dimension reflects a cheese odor gradient that occurs naturally in real-life scenarios; it is represented by the matrix  $S$  and constructed using an exponential decay dispersion algorithm. Along from its sense of location, the used agent has a rodent-like sense of smell, which allows it to identify odor gradients that aid in decision-making and efficient learning.

In a random search situation, a computer simulation shows that coordinating the senses of place and smell greatly improves the Q-learning process, which becomes up to three times more efficient. The DFS ambient also shows an increase in learning efficiency, which is a notable feat inside a high-performance method. The proposed technique enables the addition of extra dimensions and the creation of more realistic maze models.

## 7 Future Work

In principle, other dimensions could be incorporated to obtain more realistic maze models. For example, our research team has performed initial tests with an additional matrix  $U$  of self-generated odor, typical in the rodent world and created by urination, special glands, among others. Another proposal would be to implement this type of approach to solve more complex optimization problems such as finding optimal routes in situations that require moderate use of computational power. Another additional approach would be to compare our algorithm with metaheuristic algorithms focused on solving the rat in a maze problem, to know how better or worse our algorithm performs in comparison. Ultimately, this research can be used as a way to improve on the methods exposed, and create simulations closer and closer to the real world.

## References

1. Abbott, A.: Brains of norway. *Nature* **514**(7521), 154–157 (2014)
2. Arkley, K., Grant, R., Mitchinson, B., Prescott, T.: Strategy change in vibrissal active sensing during rat locomotion. *Curr. Biol.* **24**(13), 1507–1512 (2014). <https://doi.org/10.1016/j.cub.2014.05.036>
3. Bakale, V.A., Kumar VS, Y., Roodagi, V.C., Kulkarni, Y.N., Patil, M.S., Chickerur, S.: Indoor navigation with deep reinforcement learning. In: 2020 International Conference on Inventive Computation Technologies (ICICT), pp. 660–665. IEEE (2020)
4. Bonny, T., Kashkash, M.: Highly optimized q-learning-based bees approach for mobile robot path planning in static and dynamic environments. *J. Field Robot.* **39**(4), 317–334 (2022)
5. Chang, O., Gonzales-Zubiarte, F.A., Zhinin-Vera, L., Valencia-Ramos, R., Pineda, I., Diaz-Barrios, A.: A protein folding robot driven by a self-taught agent. *Biosystems* **201**, 104315 (2021)
6. Deschenes, M., Moore, J.D., Kleinfeld, D.: Sniffing and whisking in rodents. *Curr. Opin. Neurobiol.* **22**, 243–250 (2012)
7. Findley, T., et al.: Sniff-synchronized, gradient-guided olfactory search by freely-moving mice. *eLife* 10 (05 2021). <https://doi.org/10.7554/eLife.58523>
8. Gu, S., Mao, G.: An improved Q-learning algorithm for path planning in maze environments. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) *IntelliSys 2020. AISC*, vol. 1251, pp. 547–557. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-55187-2\\_40](https://doi.org/10.1007/978-3-030-55187-2_40)
9. Hafting, T., Fyhn, M., Molden, S., Moser, M.B., Moser, E.: Microstructure of a spatial map in the entorhinal cortex. *Nature* 436, 801–6 (09 2005). <https://doi.org/10.1038/nature03721>
10. Hsu, L.H., Lin, C.K.: *Graph theory and interconnection networks* (2008)
11. Jiang, W.C., Hwang, K.S., Lin, J.L.: An experience replay method based on tree structure for reinforcement learning. *IEEE Trans. Emerg. Topics Comput.* **9**(2), 972–982 (2019)

12. Jin, C., Lu, Y., Liu, R., Sun, J.: Robot path planning using q-learning algorithm. In: 2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT), pp. 202–206. IEEE (2021)
13. Kantasewi, N., Marukatat, S., Thainimit, S., Manabu, O.: Multi q-table q-learning. In: 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), pp. 1–7. IEEE (2019)
14. Kim, K.: Multi-agent deep Q network to enhance the reinforcement learning for delayed reward system. *Appl. Sci.* **12**(7), 3520 (2022)
15. Kosoy, E., et al.: Exploring exploration: Comparing children with RL agents in unified environments. arXiv preprint [arXiv:2005.02880](https://arxiv.org/abs/2005.02880) (2020)
16. Krupic, J., Bauza, M., Burton, S., Barry, C., O’Keefe, J.: Grid cell symmetry is shaped by environmental geometry. *Nature* **518**, 232–5 (2015). <https://doi.org/10.1038/nature14153>
17. Kulvicius, T., Tamosiunaite, M., Ainge, J., Dudchenko, P., Wörgötter, F.: Odor supported place cell model and goal navigation in rodents. *J. Comput. Neurosci.* **25**(3), 481–500 (2008)
18. Low, E.S., Ong, P., Low, C.Y., Omar, R.: Modified q-learning with distance metric and virtual target on path planning of mobile robot. *Expert Syst. Appl.* **199**, 117191 (2022)
19. Namalomba, E., Feihu, H., Shi, H.: Agent based simulation of centralized electricity transaction market using bi-level and Q-learning algorithm approach. *Int. J. Electr. Power Energy Syst.* **134**, 107415 (2022)
20. Okuhara, K., Nakamura, T.: Explore algorithms in olfactory system of mice. *Softw. Biol.* **3**, 20–25 (2005)
21. Radvansky, B.A., Dombeck, D.A.: An olfactory virtual reality system for mice. *Nature Commun.* **9**(1), 1–14 (2018)
22. Rajan, R., Clement, J.P., Bhalla, U.S.: Rats smell in stereo. *Science* **311**(5761), 666–670 (2006). <https://doi.org/10.1126/science.1122096>
23. Rosenberg, M., Zhang, T., Perona, P., Meister, M.: Mice in a labyrinth: Rapid learning, sudden insight, and efficient exploration (2021). <https://doi.org/10.1101/2021.01.14.426746>
24. Russell, S., Norvig, P.: *Artificial intelligence: a modern approach* (2002)
25. Sadik, A.M., Dhali, M.A., Farid, H.M., Rashid, T.U., Syeed, A.: A comprehensive and comparative study of maze-solving techniques by implementing graph theory. In: 2010 International Conference on Artificial Intelligence and Computational Intelligence, vol. 1, pp. 52–56. IEEE (2010)
26. Sagming, M., Heymann, R., Hurwitz, E.: Visualising and solving a maze using an artificial intelligence technique. In: 2019 IEEE AFRICON, pp. 1–7 (2019). <https://doi.org/10.1109/AFRICON46755.2019.9134044>
27. Soh, Z., Suzuki, M., Tsuji, T., Takiguchi, N., Ohtake, H.: A neural network model of the olfactory system of mice: computer simulation of the attention behavior of mice for some components in an odor. *Artif. Life Robot.* **12**(1–2), 75–80 (2008)
28. Steier, D.M., Anderson, A.P.: *Depth-First Search*, pp. 47–62. Springer, US, New York, NY (1989). [https://doi.org/10.1007/978-1-4613-8877-7\\_5](https://doi.org/10.1007/978-1-4613-8877-7_5)
29. Sutton, R.S., Barto, A.G.: *Reinforcement learning: An introduction*. MIT press (2018)
30. Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1**(2), 146–160 (1972)
31. Wallace, D.G., Gorny, B., Whishaw, I.Q.: Rats can track odors, other rats, and themselves: implications for the study of spatial behavior. *Behav. Brain Res.* **131**(1), 185–192 (2002). [https://doi.org/10.1016/S0166-4328\(01\)00384-9](https://doi.org/10.1016/S0166-4328(01)00384-9)

32. Watkins, C., Dayan, P.: Technical note: Q-learning. *Machine Learning* 8, 279–292 (1992). <https://doi.org/10.1007/BF00992698>
33. Wolfe, J., Mende, C., Brecht, M.: Social facial touch in rats. *Behav. Neurosci.* **125**(6), 900 (2011)
34. Zhang, B., Li, G., Zheng, Q., Bai, X., Ding, Y., Khan, A.: Path planning for wheeled mobile robot in partially known uneven terrain. *Sensors* **22**(14), 5217 (2022)