การหาผลเฉลยของปัญหาเส้นทางที่สั้นที่สุดในโครงข่ายสื่อสาร ด้วยวิธี Q-learning

A solution to the shortest path problem in communication networks based on Q-learning

ดนุพร ยิ่งยงค์[1] ยศนันท์ จันทร์ดำ[1] และ กฤษฎา มามาตร[1*]

Danuphon Yingyong[1], Yotsanan Chandam[1] and Kritsada Mamat[1*]

บทคัดย่อ

บทความนี้นำเสนอการประยุกต์ใช้วิธี Q-learning เพื่อแก้ปัญหาเส้นทางที่สั้นที่สุดในโครงข่ายสื่อสาร ซึ่งมีการใช้งานทั้งในการสื่อสารแบบมีสายและการสื่อสารแบบไร้สาย เพื่อลดต้นทุนในการสื่อสาร เช่น กำลังส่งหรือการหน่วงเวลา บทความนี้ได้เปรียบเทียบวิธี Q-learning กับวิธีแก้ปัญหาระยะทางที่สั้นที่สุดที่มีอยู่แล้ว ได้แก่ วิธีของไดก์สตรา วิธีของเบลล์แมน-ฟอร์ด และวิธีของฟลอยด์-วอร์แชล การดำเนินการเริ่มจากการศึกษาวิธีการแก้ปัญหาเส้นทางที่สั้นที่สุดทั้งสามวิธีข้างต้นเพื่อทราบถึงข้อดีและข้อเสียของแต่ละวิธี ในการนำเสนอการประยุกต์ใช้วิธี Q-learning นั้น มีการแปลงปัญหาระยะทางที่สั้นที่สุดให้เป็นองค์ประกอบของวิธี Q-learning ได้แก่ เอเจนท์ แอคชัน สเตจ สิ่งแวดล้อม และรางวัล ผลการจำลองการแก้ปัญหาด้วยวิธีทั้งสี่แสดงให้เห็นว่าทุกวิธีให้ผลเฉลยการแก้ปัญหาที่เท่ากัน ในส่วนของวิธี Q-learning นั้นพบว่า วิธีดังกล่าวใช้จำนวนครั้งของการเรียนรู้ไม่มาก ซึ่งในตัวอย่างที่สนใจนั้นใช้การเรียนรู้ จำนวน 35 ครั้ง นอกจากนั้นยังพบด้วยว่า วิธีดังกล่าวให้ผลเฉลยสำหรับทุกโหนดปลายทางใด ๆ ไปยังโหนดต้นทางซึ่งเทียบเท่ากับวิธีของฟลอยด์-วอร์แชล อย่างไรก็ตามที่วิธี Q-learning นั้น มีความซับซ้อนในการคำนวณน้อยกว่า เมื่อพิจารณาจากผลการศึกษาพบว่า วิธี Q-learning สามารถนำไปประยุกต์ใช้ในกรณีที่มีคู่โหนดต้นทาง-โหนดปลายทางต้องการส่งข้อมูลพร้อมกันมากกว่าหนึ่งคู่โหนด

คำสำคัญ: โครงข่ายสื่อสาร ปัญหาเส้นทางที่สั้นที่สุด Q-learning

Abstract

This paper proposes an application of a Q-learning algorithm to solve the shortest path problem in communication networks, which are widely used in both wireline and wireless communications. The objective is to minimize the cost occurred in a transmission process. The cost may refer to transmit power and time delay. To bench mark such a proposed method, Dijkstra's, Bellman-Ford's, and Floy-Warshall's algorithms, which are existing methods, are taken into a comparison. The research method begins with studying Dijkstra's, Bellman-Ford's, and Floy-Warshall's algorithms to obtain some insights of advantages and disadvantages. In applying the Q-learning, the shortest path

[1] วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

[1] College of Industrial Technology, King Mongkut's University of Technology North Bangkok

[*] Corresponding author. E-mail: kritsada.m@cit.kmutnb.ac.th

Received: July 8, 2023; Revised: September 26, 2023; Accepted: November 7, 2023

problem is decomposed into Q-learning's components such as agent, action, state, environment, and reward which can be completely used in training process. Experimental results show that all methods provide the same solution. For the Q-learning method, it can be stated that this method requires a small number of episodes in training procedure e.g., 35 episodes for the interested example. Furthermore, it can also be found that the Q-learning algorithm offers all paths for any sources to the destination node. This is comparable with the Floy-Warshall's algorithm. However, the Q-learning method requires less complexity in calculation. Furthermore, it can be concluded from the study results that the Q-learning method can be applied in multiple source-destination pairs.

Keywords: communication network, shortest path problem, Q-learning

บทนำ

เทคโนโลยีโครงข่ายสื่อสาร (communication network) จัดเป็นเทคโนโลยีที่มีความสำคัญอย่างมากต่อการสื่อสาร ตัวอย่างของโครงข่ายสื่อสาร ได้แก่ โครงข่ายคอมพิวเตอร์ โครงข่ายเส้นใยแก้วนำแสง (fiber optic) การเชื่อมต่อถึงกันของสถานีฐาน (base station) ในระบบโทรศัพท์เคลื่อนที่ และการใช้รีเลย์ (relay) สำหรับการติดต่อสถานีเคลื่อนที่ (mobile station) ของสถานีฐาน (Raza, & Muhamad, 2015) ในการส่งข้อมูลข่าวสารผ่านโครงข่ายสื่อสารจากภาคส่งต้นทางไปยังภาครับปลายทางมักจะต้องมีการกำหนดเลือกเส้นทางที่เหมาะสมเพื่อให้ต้นทุน (cost) ที่ต้องใช้มีค่าต่ำที่สุด ต้นทุนดังกล่าวอาจเป็นค่าใช้จ่ายในด้านของกำลังส่งหรือเวลาแฝง (latency) ยกตัวอย่าง เช่น การส่งข้อมูลในโครงข่ายคอมพิวเตอร์ซึ่งต้องมีการแบ่งข้อมูลออกเป็นแพ็กเกจ (packet) เพื่อส่งจากต้นทางไปยังปลายทาง (Forouzan, 2007) เนื่องจากโครงข่ายคอมพิวเตอร์นั้นมีอุปกรณ์ที่เชื่อมต่อกันอยู่มากมายทำให้เกิดเส้นทางการส่งข้อมูลหลายเส้นทาง วิธีการหาเส้นทางที่เหมาะสมในการส่งข้อมูล (routing) ที่เหมาะสมจึงเป็นที่สนใจและได้รับการปรับปรุงแก้ไขมาตลอด ยกตัวอย่างเช่น งานวิจัยของ Li, Zheng, & Zhang (2011) ได้นำเสนอวิธีการหาเส้นทางโดยคำนึงถึงกำลังส่งที่ใช้ในโครงข่ายเซนเซอร์ไร้สาย (wireless sensor network) งานวิจัยของ Chanda, Rout, & Lingam (2018) ได้นำเสนอวิธีการหาเส้นทางเพื่อลดการหน่วงเวลา (delay) ในโครงข่ายโทรศัพท์เคลื่อนที่ งานวิจัยของ Aneja (2001) ได้นำเสนอวิธีการหาเส้นทางในโครงข่ายใยแก้วนำแสงเพื่อลดความแออัดในโครงข่าย จากการศึกษาวิธีการค้นหาเส้นทางพบว่า ปัญหาเส้นทางที่สั้นที่สุด (shortest path problem) เป็นวิธีการหาเส้นทางที่ได้รับความสนใจเนื่องจากเป็นแนวคิดการแก้ปัญหาที่ไม่ซับซ้อนและมีผลต่อการประหยัดต้นทุนเป็นอย่างมาก หลักการของปัญหาดังกล่าวคือ การหาเส้นทางที่ใช้ต้นทุนน้อยที่สุดในการส่งข้อมูลโดยหนังสือของ Wuttisittikulkij (2016) ได้รวบรวมวิธีการแก้ปัญหาดังกล่าวเอาไว้ 3 วิธี ได้แก่ วิธีของไดก์สตรา (Dijkstra) วิธีของเบลล์แมน-ฟอร์ด (Bellman-Ford) และวิธีของฟลอยด์-วอร์แชล (Floy-Warshall)

ในปัจจุบันเทคโนโลยีปัญญาประดิษฐ์ (artificial intelligence: AI) หรือการเรียนรู้ของเครื่อง (machine learning: ML) ได้รับความสนใจและมีการประยุกต์ใช้งานที่หลากหลาย หลักการของ ML คือการเรียนรู้ผ่านกระบวนการลองผิดลองถูก (trial-and-error) ของเอเจนท์

(agent) ต่อสิ่งแวดล้อม (environment) เพื่อสร้างกระบวนการตัดสินใจที่เหมาะสม Sutton, & Barto (2018) งานวิจัยที่ผ่านมามีการประยุกต์ใช้งาน ML ในงานเชิงวิศวกรรมสื่อสารเป็นจำนวนมากโดยเฉพาะวิศวกรรมสื่อสารไร้สาย ตัวอย่างเช่น Chen et al. (2021) และ Nguyen et al. (2021) ได้รวบรวมการประยุกต์ใช้ ML ในการสื่อสารไร้สายชั้นฟิสิคัล (physical layer) ตัวอย่างเช่น การระบุคุณลักษณะของช่องสัญญาณ การตรวจจับสัญญาณ การประมาณช่องสัญญาณ งานวิจัยของ Pryshchenko, Dumin, Plakhtii, Shyrokorad, & Pochanin (2021) ได้นำเสนอการประยุกต์ใช้ ML ในการตรวจจับวัตถุด้วยสัญญาณเรดาร์ งานวิจัยของ Mete, & Girici (2020) ได้นำเสนอการประยุกต์ใช้ ML สำหรับการตรวจจับสัญญาณในระบบสื่อสารไร้สายยุคที่ 5 และในงานวิจัยของ Al-Tam, Correia, & Rodriguez (2020) ได้นำเสนอการประยุกต์ใช้ ML เพื่อจัดลำดับการส่งสัญญาณในระบบสื่อสารไร้สาย

บทความนี้นำเสนอการประยุกต์ใช้ปัญญาประดิษฐ์เพื่อแก้ปัญหาระยะทางที่สั้นที่สุดโดยสามารถสรุปที่สิ่งนำเสนอได้ดังนี้

1. ศึกษาวิธีแก้ปัญหาระยะทางที่สั้นที่สุดที่มีอยู่ก่อนหน้า ได้แก่ วิธีของไดก์สตรา วิธีของเบลล์แมน-ฟอร์ด และวิธีของฟลอยด์-วอร์แชล จากผลการศึกษาพบว่า วิธีของไดก์สตราและวิธีของเบลล์แมน-ฟอร์ดสามารถหาเส้นทางที่สั้นที่สุดได้เพียงคู่โหนดเดียวเท่านั้นและวิธีของฟลอยด์-วอร์แชลสามารถแก้ปัญหาเส้นทางที่สั้นที่สุดสำหรับทุกคู่โหนดได้ อย่างไรก็ตามวิธีดังกล่าวมีการคำนวณเส้นทางที่ซับซ้อนเนื่องจากต้องคำนวณระยะทางสำหรับทุกคู่โหนด

2. นำเสนอวิธีการประยุกต์ใช้งาน Q-learning ซึ่งจัดเป็น ML วิธีหนึ่งเพื่อแก้ปัญหาระยะทางสั้นที่สุดโดยการแปลงโครงข่ายสื่อสารให้เป็นองค์ประกอบของ Q-learning ซึ่งประกอบไปด้วย เอเจนท์ สิ่งแวดล้อม แอคชัน (action) สเตจ (stage) และรางวัล จากการศึกษาและการออกแบบผลการทดลองพบว่าวิธี Q-learning ให้ผลเฉลยเป็นระยะทางที่สั้นที่สุดของทุกคู่โหนดเทียบเท่าวิธีของฟลอยด์-วอร์แชล อย่างไรก็ตามวิธี Q-learning มีขั้นตอนการทำงานที่ซับซ้อนน้อยกว่าวิธีของของฟลอยด์-วอร์แชลและสามารถนำวิธี Q-learning ไปประยุกต์ใช้แก้ปัญหาระยะทางที่สั้นที่สุดที่มีคู่ของโหนดต้นทางและโหนดปลายทางมากกว่า 1 คู่ต้องการส่งข้อมูลพร้อมกันได้

วิธีการศึกษา

บทความนี้สนใจการแก้ปัญหาระยะทางที่สั้นที่สุดในโครงข่ายสื่อสารรูปแบบ mesh ดัง (Figure 1)

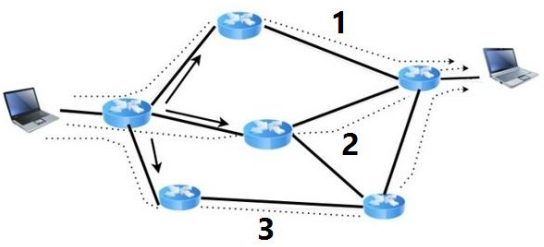

Figure 1 An example of data transmission routes a mesh network.

เมื่อพิจารณา (Figure 1) พบว่าการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางนั้น สามารถมีได้หลายเส้นทาง อย่างไรก็ตามมีเพียงเส้นทางเดียว

เท่านั้นที่มีต้นทุนต่ำที่สุด การศึกษาการแก้ปัญหาเส้นทางที่สั้นที่สุดในโครงข่ายสื่อสารด้วยวิธีต่าง ๆ ที่กล่าวถึงในบทความนี้ เริ่มต้นได้แทนโครงข่ายสื่อสารด้วยรูปแบบกราฟ (graph) ดัง (Figure 2) หนังสือ Wilson (1996) แสดงให้เห็นว่าการแทนโครงข่ายด้วยกราฟนั้นสามารถใช้ทดสอบวิธีการแก้ปัญหาระยะทางที่สั้นที่สุดที่นำเสนอในบทความนี้ได้อย่างแม่นยำ กราฟที่แสดงดัง (Figure 2) นั้นเป็นโครงข่ายที่มีความซับซ้อนมากว่ารูปโครงข่ายใน (Figure 1)

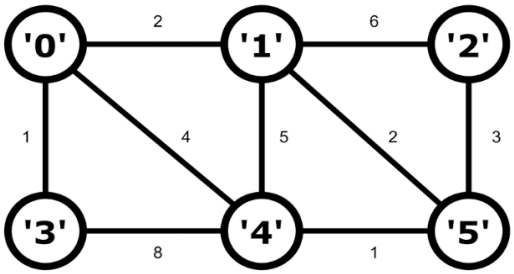

Figure 2 A graph representing the communication network.

จาก (Figure 2) กำหนดให้วงกลมแต่ละวงแทนโหนดแต่ละโหนดของโครงข่ายสื่อสาร แต่ละโหนดมีชื่อเรียกเป็นตัวเลขอยู่ในเครื่องหมาย ‘’ และเส้นเชื่อมวงกลมแต่ละเส้นแทนเส้นทางจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง แต่ละเส้นทางมีต้นทุนเป็นตัวเลขที่แตกต่างกัน ตัวอย่างเช่น เส้นทางจากโหนด ‘0’ ไปยังโหนด ‘3’ มีต้นทุนเท่ากับ 1 และเส้นทางจากโหนด ‘3’ ไปยังโหนด ‘4’ มีต้นทุนเท่ากับ 8 ตามลำดับ สมมติฐานของการแก้ปัญหาเส้นทางที่สั้นที่สุดในบทความนี้คือการหาเส้นทางที่สั้นที่สุดของทุกคู่โหนดต้นทาง-ปลายทาง ศึกษาการแก้ปัญหาเส้นทางที่สั้นที่สุดด้วยวิธีของไดก์สตรา เบลล์แมน-ฟอร์ด และฟลอยด์-วอร์แชลล์ ที่นำเสนอในบทความนี้นั้นได้อ้างอิงจากการศึกษาใน Wuttisittikulkij (2016)

1. การศึกษาการแก้ปัญหาเส้นทางที่สั้นที่สุดด้วยวิธีของไดก์สตรา เบลล์แมน-ฟอร์ด และฟลอยด์-วอร์แชลล์

วิธีแก้ปัญหาเส้นทางที่สั้นที่สุดวิธีแรกที่ศึกษาในบทความนี้คือวิธีของไดก์สตรา ข้อจำกัดของวิธีดังกล่าวคือ ต้นทุนของแต่ละเส้นทางมีค่าเป็นบวกหรือศูนย์เท่านั้นโดยสามารถสรุปได้ ดัง (Algorithm 1)

Algorithm 1 Dikstra algorithm.

1. define $C_{k,l}$ to be budget of path from node k to node l, set $V[k]$ to be budget of the path with minimum budget from the origin node to node k, set $d[k]$ to be node's number located before node k in the shortest path from the origin node to node k
2. assuming that all nodes are temporary node and store all nodes in set T, set the origin node to be starting node S
3. assuming $V[i] = \begin{cases} 0; & i = s \\ +\infty; & \text{otherwise} \end{cases}$
4. **WHILE** the number of entries in set T does not equal to 0 **DO**
5. $p \leftarrow s$ % change starting node to be permanent node
6. assuming that vector N to be vector of the adjacent node with S
7. **FOR** i in N **DO**
8. $V[i] \leftarrow min\{V[i], V[p] + C_{p,i}\}$

Algorithm 1 Dikstra algorithm (continue).

9. IF $V[i]$ change THEN
10. $d[k] \leftarrow p$
11. ENDIF
12. ENDFOR
13. update $s = \text{argmin}_{s \in T} V$
14. update $T \leftarrow T \backslash s$ % subtract s from set T
15. ENDWHILE
16. RETURN $V[k]$ and $d[k]$

วิธีการแก้ปัญหาเส้นทางที่สั้นที่สุดวิธีที่สอง ที่ได้ศึกษาในบทความนี้คือวิธีของเบลล์แมน-ฟอร์ด ซึ่งมีข้อจำกัดน้อยกว่าวิธีของไดก์สตราเพราะอนุญาต ให้ต้นทุนของแต่ละโครงข่ายเชื่อมโยงมีค่าเป็นลบได้ ซึ่งสามารถสรุปได้ ดัง (Algorithm 2)

Algorithm 2 Bellman-Ford algorithm.

1. define K to be the number of all nodes, $c_{k,l}$ to be budget of the path from node k to node l, set $V[k]$ to be the budget of the path with minimum budget from the origin node to node k, $V^{(t)}[k]$ represents $V[k]$ obtained from the t iteration, set $d[k]$ to be the node's number located before node k in the shortest path from the origin node to node k
2. set the origin node to be starting node s
3. assuming $V^{(0)}[k] = \begin{cases} 0; & k = s \\ +\infty; & \text{otherwise} \end{cases}$
4. FOR $t = 1:K$ DO
5. $V^{(0)}[k] \leftarrow min\{V^{(t-1)}[i] + c_{i,k}\}$ % for all nodes i connecting with node k
6. IF $V^{(0)}[k] < V^{(t-1)}[k]$; For all k THEN
7. $d[k] \leftarrow i$
8. ELSEIF $V^{(0)}[k] = V^{(t-1)}[k]$ THEN
9. BREAK
10. ENDIF
11. ENDFOR
12. RETURN $V[k]$ and $d[k]$

วิธีการแก้ปัญหาเส้นทางที่สั้นที่สุดวิธีที่สาม ที่ได้ศึกษาคือวิธีของฟลอยด์-วอร์แชลล์ วิธีดังกล่าวได้ลบข้อด้อยของสองวิธีก่อนหน้าที่ไม่สามารถหา ระยะทางสั้นที่สุดของทุกคู่โหนดได้ ในการคำนวณครั้งเดียวโดยสรุปได้ ดัง (Algorithm 3)

Algorithm 3 Floyd-Warshall algorithm.

1. define K to be the number of all nodes, $c_{k,l}$ to be budget of the path from node k to node l, set $V^{(t)}[k,l]$ to be the budget of the path with minimum budget from node k to node l by providing nodes along the route with numbers less than or equal to t, set $d[k,l]$ to be node located before node l currently located on the shortest path from node k to node l
2. sort number of nodes in ascending
3. IF k is connected with l THEN % for all nodes $k, l \in [1:K]$
4. $V^{(0)}[k,l] = c_{k,l}$
5. $d[k,l] = k$
6. ELSE
7. $V^{(0)}[k,l] = \begin{cases} 0; & k = l \\ +\infty; & \text{otherwise} \end{cases}$
8. ENDIF
9. FOR $t = 1:K$ DO
10. $V^{(t)}[k,l] \leftarrow min\{V^{(t-1)}[k,l], V^{(t-1)}[k,t], V^{(t-1)}[t,l]\}$
    % for all node $k, l \in [1:K],\ k, l \neq t$
11. IF $V^{(t)}[k,l] < V^{(t-1)}[k,l]$ THEN
12. $d[k,l] \leftarrow d[t,l]$
13. ENDIF
14. IF $V^{(t)}[k,k] < 0$ THEN
15. BREAK
16. ENDIF
17. ENDFOR
18. RETURN $V^{(t)}[k,l]$ and $d[k,l]$

2. การประยุกต์ใช้วิธี Q-learning เพื่อการแก้ปัญหาเส้นทางที่สั้นที่สุด

ตามที่ได้กล่าวไว้ในหัวข้อก่อนหน้าว่าโครงข่ายสื่อสารสามารถแทนได้ด้วยกราฟที่มีโหนดแทนอุปกรณ์สื่อสารและมีเส้นทางการเชื่อมต่อที่มีต้นทุนแตกต่างกัน การแทนโครงข่ายสื่อสารดังกล่าวเป็นแนวทางการแก้ปัญหาเส้นทางที่สั้นที่สุดด้วย ML เนื่องจาก ML สามารถจดจำเส้นทางและต้นทุนของแต่ละเส้นทางรวมถึงหาเส้นทางที่สั้นที่สุดจากโหนดหนึ่งไปยังอีกโหนดหนึ่งได้ โดยผ่านกระบวนการเรียนรู้ (training) สำหรับวิธี ML ที่เลือกใช้ในบทความนี้คือวิธี Q-learning เนื่องจากเป็นวิธี ML พื้นฐานที่มีกระบวนการทำงานไม่ซับซ้อนและสามารถใช้แก้ปัญหาลักษณะนี้ได้ วิธี Q-learning มีส่วนประกอบที่สำคัญคือเอเจนท์ (agent) สิ่งแวดล้อม (environment) สเตจ (state) การกระทำหรือแอคชัน

(action) และรางวัล (reward) การทำงานของวิธี Q-learning นั้นอยู่บนพื้นฐานของกระบวนการตัดสินใจมาร์คอฟ (Markov decision process: MDP) เริ่มต้นจากเอเจนท์ดำเนินการแอคชัน a_t กับสิ่งแวดล้อมที่สเตจปัจจุบัน s_t จากนั้นสิ่งแวดล้อมจะคืนค่ารางวัล r_t และย้ายไปที่สเตจ s_{t+1} ในสเตจถัดไป นั้นเอเจนท์จะดำเนินการแอคชัน a_{t+1} เพื่อรับรางวัล r_{t+1} การกระทำดังกล่าวจะดำเนินการไปเรื่อย ๆ เพื่อหาแอคชันที่เหมาะสมสำหรับแต่ละสเตจ ความหมายของแอคชันที่เหมาะสมคือแอคชันที่กระทำต่อสิ่งแวดล้อมในสเตจปัจจุบันแล้วได้รางวัลคืนค่ากลับมาสูงที่สุด ในการแก้ปัญหาระยะทางที่สั้นที่สุดนั้นบทความนี้ได้เสนอการแปลงปัญหาดังกล่าวให้เป็นส่วนประกอบของวิธี Q-learning ดังนี้

1. **เอเจนท์**: เนื่องจากบทความนี้สนใจปัญหาระยะทางที่สั้นที่สุดในโครงข่ายสื่อสาร ดังนั้นเอเจนท์ในปัญหานี้คือข้อมูลข่าวสารที่ต้องการส่งจากโหนดต้นทางไปยังโหนดปลายทาง

2. **แอคชัน**: แอคชันในที่นี้คือการเดินทางของข้อมูลข่าวสารจากโหนดหนึ่งไปยังโหนดที่มีเส้นทางเชื่อมต่อถึงกัน แต่เนื่องจากโครงข่ายสื่อสารที่สนใจเป็นโครงข่ายแบบ mesh ที่มีรูปแบบการเชื่อมต่อที่ไม่ตายตัว ดังนั้นเพื่อให้ครอบคลุมกับรูปแบบการเชื่อมต่อทั้งหมด จึงกำหนดให้แอคชันมีลักษณะเป็นการเดินทางในตารางทั้งหมด 8 ทิศทาง ได้แก่ N (north), NE (north east), E (east), SE (south east), S (south), SW (south west), W (west) และ NW (north west) โดยสามารถอธิบายเอเจนท์และแอคชันได้อย่างง่าย ดัง (Figure 3)

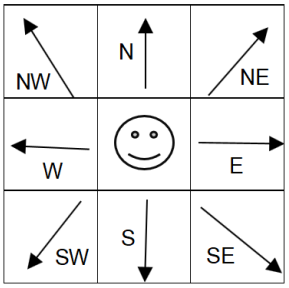

Figure 3 Agent and actions.

จาก (Figure 3) กำหนดให้รูปหน้ายิ้มแทนเอเจนท์หรือข้อมูลข่าวสารและแอคชันคือทิศทางการเดินทั้ง 8 ทิศทางตามที่ได้กล่าวมาแล้ว

3. **สเตจ**: ในที่นี้สเตจคือหมายเลขประจำโหนดที่เอเจนท์หรือข้อมูลข่าวสารอยู่ในขณะนั้น

4. **รางวัล**: ตามที่กล่าวไว้ก่อนหน้าว่าการแก้ปัญหาระยะทางที่สั้นที่สุดคือการหาระยะทางที่มีต้นทุนรวมน้อยที่สุด ดังนั้นในการกำหนดรางวัลสำหรับวิธี Q-learning ในบทความนี้จึงกำหนดให้รางวัลเป็นต้นทุนที่ติดลบในแต่ละเส้นทาง กล่าวคือหากต้องการรางวัลที่สูงที่สุดก็คือการไปในเส้นทางที่ติดลบน้อยที่สุดนั่นเอง ดังนั้นเมื่อเอเจนท์ต้องการเดินทางจากโหนดต้นทางไปยังโหนดปลายทางในระยะทางที่สั้นที่สุดต้องได้รับรางวัลมากที่สุดในแต่ละสเตจ และเนื่องจากกำหนดให้แอคชันมีลักษณะเป็นการเดินทางในตาราง เพื่อให้กระบวนการดังกล่าวเป็นไปอย่างถูกต้องจำเป็นต้องมีการสร้างตารางเพื่อบันทึกค่า Q (Q-value) สำหรับทุกสเตจและแอคชันทั้งหมดที่เป็นไปได้ ตารางดังกล่าวมีชื่อ Q-table จำนวนของเซลล์ใน Q-table ทั้งหมดเท่ากับจำนวนสเตจทั้งหมดคูณกับจำนวนแอคชันทั้งหมด ดังนั้นจึงกำหนดให้รางวัลมีลักษณะเป็นตารางเช่นกัน ดัง (Table 1) แสดงรางวัล ดัง (Figure 2)

Table 1 Reword of each node pair for (Figure 2).

node pair	0	1	2	3	4	5
0	-100	-2	-100	-1	-4	-100
1	-2	-100	-6	-100	-5	-2
2	-100	-6	-100	-100	-100	-3
3	-1	-100	-100	-100	-8	-100
4	-4	-5	-100	-8	-100	-1
5	-100	-2	-3	-100	-1	-100

จาก (Table 1) กำหนดให้หลักแรกที่มีตัวเลขตั้งแต่ 0-5 และแทนสเตจปัจจุบันและแถวแรกที่มีตัวเลขตั้งแต่ 0-5 แทนสเตจถัดไป ตัวเลขติดลบอื่น ๆ ในตารางแทนรางวัลของการเดินทางจากสเตจปัจจุบันไปยังสเตจถัดไป ตัวอย่างเช่น แถวที่ 1 หลักที่ 2 มีรางวัลเท่ากับ -2 มีความหมายคือการเดินทางจากโหนด '0' ไปยังโหนด '1' ในโครงข่ายจาก (Figure 3) มีรางวัลเท่ากับ -2 ซึ่งตัวเลข -2 นี้มาจากต้นทุนของเส้นทางจากโหนด '0' ไปยังโหนด '1' มีค่าเท่ากับ 2 แต่เนื่องจากวิธี Q-learning ต้องการรางวัลที่มีค่ามากในขณะที่ปัญหาเส้นทางที่สั้นที่สุดต้องการต้นทุนที่มีค่าน้อย ดังนั้นการให้รางวัลเป็นต้นทุนที่ติดลบจึงทำให้ทั้งสองปัญหามีความสอดคล้องกัน ในส่วนของตัวเลข -100 นั้นแทนการเดินทางไปที่โหนดเดิมหรือการเดินทางไปยังโหนดที่ไม่มีเส้นทางเชื่อมต่อ ทั้งสองกรณีนี้ไม่อาจเกิดขึ้นได้จริงจึงต้องให้รางวัลที่มีค่าต่ำ เพื่อให้เอเจนท์เกิดการเรียนรู้ว่าไม่ควรทำทั้งสองกรณีนี้

5. **สิ่งแวดล้อม**: เพื่อให้สอดคล้องกับเอเจนท์ แอคชัน และรางวัลที่ได้อธิบายไปก่อนหน้า สิ่งแวดล้อมที่สนใจได้ ดัง (Algorithm 4)

Algorithm 4 Environment $\boldsymbol{env}(\cdot)$.

1. define current state s_t
2. choose 1 from 8 actions a_t
3. go to the next state s_{t+1}
4. calculate reward $r(s_t, a_t)$ from (Table 1)
5. IF $r(s_t, a_t) = -100$ THEN
6. $terminal = T$ (true)
7. ELSE
8. $terminal = F$ (false)
9. ENDIF
10. RETURN s_{t+1}, $r(s_t, a_t)$ and $terminal$

เมื่อพิจารณา (Algorithm 4) พบว่าเมื่อมีการทำแอคชันกับสเตจปัจจุบันจะได้รางวัลกลับมา ซึ่งสอดคล้องหลักการของวิธี Q-learning ตามที่ได้อธิบายไปแล้วข้างต้น ดังนั้นในส่วนของกระบวนการเรียนรู้จึงต้องมีการสร้าง Q-table เพื่อบันทึกค่า Q หรือรางวัลสำหรับแต่ละคู่ สเตจ-แอคชัน สำหรับการแก้ปัญหาระยะทางสั้นที่สุดนั้น มีการออกแบบ Q-table เอาไว้ดัง (Table 2)

Table 2 Q-table.

stage-action	N	NE	E	SE	S	SW	W	NW
0	Q (0, N)	Q (0, NE)	Q (0, E)	Q (0, SE)	Q (0, S)	Q (0, SW)	Q (0, W)	Q (0, NW)
1	Q (1, N)	Q (1, NE)	Q (1, E)	Q (1, SE)	Q (1, S)	Q (1, SW)	Q (1, W)	Q (1, NW)
2	Q (2, N)	Q (2, NE)	Q (2, E)	Q (2, SE)	Q (2, S)	Q (2, SW)	Q (2, W)	Q (2, NW)
3	Q (3, N)	Q (3, NE)	Q (3, E)	Q (3, SE)	Q (3, S)	Q (3, SW)	Q (3, W)	Q (3, NW)
4	Q (4, N)	Q (4, NE)	Q (4, E)	Q (4, SE)	Q (4, S)	Q (4, SW)	Q (4, W)	Q (4, NW)
5	Q (5, N)	Q (5, NE)	Q (5, E)	Q (5, SE)	Q (5, S)	Q (5, SW)	Q (5, W)	Q (5, NW)

จาก (Table 2) เป็น Q-table สำหรับ (Figure 2) จาก (Table 2) หลักแรกคือจำนวนสเตจซึ่งเท่ากับจำนวนโหนดและแถวแรกคือแอคชันทั้งหมดที่เป็นไปได้ของแต่ละสเตจ สำหรับค่าอื่น ๆ ในตารางคือค่า Q สำหรับแต่ละคู่ สเตจ-แอคชัน $Q(s_t, a_t)$ โดยสามารถคำนวณได้ตามสมการของ Bellman อ้างอิงจาก Sutton, & Barto (2018) ดังสมการ (1)

$$Q^{new}(s_t, a_t) \leftarrow Q^{old}(s_t, a_t) + \eta((r(s_t, a_t) + \beta \max_a Q^{new}(s_{t+1}, a) - Q^{old}(s_t, a_t)) \quad (1)$$

โดยที่ η คืออัตราการเรียนรู้ (learning rate) และ β แทนปัจจัยส่วนลด (discount factor) เมื่อพิจารณาสมการ (1) พบว่าค่าที่อัพเดทใน Q-table หรือ $Q^{new}(s_t, a_t)$ ขึ้นอยู่กับค่าเดิม $Q^{old}(s_t, a_t)$ รางวัล $r(s_t, a_t)$ และรางวัลสูงสุดที่คาดหวังสามารถแทนได้ด้วย $\max_a Q^{new}(s_{t+1}, a)$ เมื่อการอัพเดท Q-table เสร็จสมบูรณ์แล้วเอเจนท์ก็จะทราบแอคชันที่เหมาะสมสำหรับแต่ละสเตจ บทความนี้สนใจการประยุกต์ใช้ Q-learning เพื่อแก้ปัญหาเส้นทางที่สั้นที่สุดในโครงข่ายสื่อสารตามที่ได้กล่าวมาแล้วข้างต้น กระบวนการเรียนรู้เพื่อหาค่าใน Q-table ดัง (Table 2) สามารถสรุปได้ ดัง (Algorithm 5)

Algorithm 5 Training process.

1. define N to be the number of training iterations, E to be the number of episodes and probability p
2. FOR $episode = 1{:}\,E$ DO
3. initialize state s_t, $terminal = F$ and $counter = 0$
4. WHILE $terminal = F$ DO
5. $counter = counter + 1$
6. IF $rand(\cdot) < p$ THEN % function $rand(\cdot)$ generates a random number in range 0-1
7. randomly choose 1 from 8 actions a_t
8. ELSE
9. choose action from $a_t = argmax_a Q(s_t, a_t)$
10. ENDIF

Algorithm 5 Training process (continue).

11. $s_{t+1}, r_t(s_t, a_t), terminal = Env(a_t)$
12. update Q-table using equation (1)
13. IF $counter \geq N$ THEN
14. BREAK
15. ENDIF
16. ENDWHILE
17. ENDFOR

ผลการศึกษาและอภิปรายผล

ส่วนนี้เป็นการนำเสนอผลการศึกษาวิธีการแก้ปัญหาระยะทางที่สั้นที่สุดในหัวข้อก่อนหน้า การนำเสนอนั้นจะสอดคล้องกับหัวข้อก่อนหน้า เพื่อให้ง่ายต่อการทำความเข้าใจ การจำลองวิธีการแก้ปัญหาระยะทางที่สั้นที่สุดทั้งสี่วิธีที่ได้ศึกษาในบทความนี้ ได้แก่ วิธีของไดก์สตรา วิธีของเบลล์แมน-ฟอร์ด วิธีของฟลอยด์-วอร์แชลล์ และวิธี Q-learning นั้นใช้ภาษา Python ซึ่งเป็นภาษาที่นิยมใช้งานในปัจจุบัน เนื่องจากมีไลบรารี่ที่สนับสนุนการคำนวณทางคณิตศาสตร์โดยเฉพาะการสร้าง ML บทความนี้ได้จำลองการทำงานของทั้งสี่วิธีและเปรียบเทียบเพื่อให้เห็นความแตกต่างของวิธีต่าง ๆ การแก้ปัญหาระยะทางสั้นที่สุดของทั้ง 4 วิธีนั้นจะทดสอบกับโครงข่ายสื่อสาร ดัง (Figure 2)

1. ผลการศึกษาการแก้ปัญหาเส้นทางที่สั้นที่สุดด้วยวิธีของไดก์สตรา เบลล์แมน-ฟอร์ด และฟลอยด์-วอร์แชลล์

เมื่อพิจารณา (Algorithm 1) พบว่าหลักการของวิธีของไดก์สตราคือการเลือกเส้นทางที่มีต้นทุนต่ำที่สุดจากโหนดเริ่มต้นไปยังโหนดข้างเคียง จากนั้นก็ย้ายไปโหนดข้างเคียงที่มีต้นทุนต่ำที่สุดและกำหนดให้เป็นโหนดเริ่มต้นในรอบถัดไป กระบวนการนี้จะถูกทำซ้ำไปเรื่อย ๆ จนไม่เหลือโหนดข้างเคียงสำหรับโหนดเริ่มต้นตัวสุดท้าย เมื่อสิ้นสุด (Algorithm 1) แล้วจะได้ค่า $V[k]$ และ $d[k]$ ของโหนดเริ่มต้นในข้อที่ 2 ของ (Algorithm 1) ซึ่งทั้งสองค่าเป็นตัวบอกต้นทุนและระยะทางที่สั้นที่สุดจากโหนดเริ่มต้นไปยังแต่ละโหนด ในส่วนของ (Algorithm 2) นั้นถูกพัฒนาขึ้นเพื่อลดข้อจำกัดเรื่องการอนุญาตให้สามารถใช้ต้นทุนที่เป็นลบได้ อย่างไรก็ตามวิธีดังกล่าวไม่สามารถใช้กับโครงข่ายที่ประกอบด้วยวงจรปิดมีทิศทางที่มีค่าต้นทุนเป็นลบอยู่ ผลดังกล่าวมีการสนับสนุนจาก Javaid (2013)

จาก (Figure 4-5) แสดงผลการแก้ปัญหาระยะทางสั้นที่สุดจากโหนด '3' ไปยังโหนด '5' ด้วยวิธีของไดก์สตราและเบลล์แมน-ฟอร์ด ตามลำดับ จากรูปทั้งสองพบว่าทั้งสองวิธีให้ผลเฉลยเป็นระยะทางที่สั้นที่สุดจากโหนดต้นทางไปยังโหนดปลายทางได้เท่ากัน ผลการแก้ปัญหาที่ได้จะแสดงโหนดต้นทาง (src) โหนดปลายทาง (dst) ต้นทุนหรือระยะทาง (distance) และเส้นทางหรือโหนดทั้งหมดที่ข้อมูลข่าวสารต้องเดินทางผ่านเพื่อไปยังโหนดปลายทาง (visited nodes) จากรูปทั้งสองเมื่อกำหนดให้โหนดต้นทางเป็นโหนด '3' และโหนดปลายทางเป็นโหนด '5' แล้วโหนดที่ข้อมูลข่าวสารต้อง

เดินทางผ่านทั้งหมดได้แก่โหนด ‘3’, ‘0’, ‘1’ และ ‘5’ ซึ่งมีต้นทุนเท่ากับ 5 ซึ่งเป็นคำตอบที่ถูกต้อง อย่างไรก็ตาม เมื่อพิจารณารูปทั้งสองประกอบกับ (Algorithm 1) และ (Algorithm 2) พบว่าทั้งสองวิธีต้องมีการคำนวณเส้นทางใหม่ทุกครั้งสำหรับทุกคู่โหนดต้นทาง-ปลายทาง ทำให้ต้องมีการคำนวณที่ซับซ้อนเมื่อต้องหาเส้นทางสำหรับทุกคู่โหนดต้นทาง-ปลายทาง จึงเป็นข้อด้อยของทั้งสองวิธีตามที่กล่าวเอาไว้ในหัวข้อวิธีการศึกษา

```
src=3
dst=5
Distance= 5
Visited nodes= ['3', '0', '1', '5']
```

Figure 4 A Dijkstra’s solution for the shortest path problem from node ‘3’ to node ‘5’.

```
src =3
dst =5
Distance =5
Visited nodes=['3', '0', '1', '5']
```

Figure 5 A Bellman-Ford’s solution for the shortest path problem from node ‘3’ to node ‘5’.

ในส่วนของวิธีของฟลอยด์-วอร์แชลล์ นั้น เมื่อพิจารณา (Algorithm 3) พบว่าสิ่งที่ทำให้วิธีดังกล่าวแตกต่างจากวิธีของไดก์สตราและเบลล์แมน-ฟอร์ดคือวิธีของของฟลอยด์-วอร์แชลล์มีการพิจารณาเส้นทางของทุกคู่โหนดไปพร้อมกัน ในขณะที่ทั้งสองวิธีก่อนหน้าพิจารณาเฉพาะโหนดที่สนใจเท่านั้น ดังนั้นวิธีของของฟลอยด์-วอร์แชลล์จึงสามารถหาเส้นทางของทุกคู่โหนดต้นทางและโหนดปลายทางได้ในการคำนวณเพียงครั้งเดียว ผลดังกล่าวมีการสนับสนุนจาก Mukhlif, & Safi (2020)

จาก (Figure 6) แสดงตัวอย่างผลการแก้ปัญหาระยะทางสั้นที่สุดของทุกคู่โหนดด้วยวิธีของฟลอยด์-วอร์แชลล์ เมื่อพิจารณาผลของรูปดังกล่าวประกอบกับ (Algorithm 3) พบว่าวิธีดังกล่าวนั้นให้ผลเฉลยเป็นระยะทางสั้นที่สุดของทุกคู่โหนด ใน (Figure 6) นั้นแสดงตัวอย่างผลเฉลย 6 จากทั้งหมด 36 ผลเฉลย (จำนวนผลเฉลยทั้งหมดมาจากจำนวนโหนดทั้งหมดยกกำลังสองใน (Figure 4) คือ $6^2 = 36$) ผลเฉลยดังกล่าวรวมถึงผลเฉลยที่โหนดต้นทางและโหนดปลายทางเป็นโหนดเดียวกันด้วย (กรอบสีน้ำเงินในรูป) เมื่อเปรียบเทียบผลเฉลยของระยะทางที่สั้นที่สุดจากโหนด ‘3’ ไปยังโหนด ‘5’ ใน (Figure 6) (กรอบสีแดงในรูป) กับผลเฉลยใน (Figure 4) และ (Figure 5) พบว่าให้ผลเฉลยที่เท่ากัน

```
Shortest path from 3 to 0
Distance: 1
Visited nodes: [3, 0]
Shortest path from 3 to 1
Distance: 3
Visited nodes: [3, 0, 1]
Shortest path from 3 to 2
Distance: 8
Visited nodes: [3, 0, 1, 5, 2]
Shortest path from 3 to 3
Distance:0
Visited nodes: 3
Shortest path from 3 to 4
Distance: 5
Visited nodes: [3, 0, 4]
Shortest path from 3 to 5
Distance: 5
Visited nodes: [3, 0, 1, 5]
```

Figure 6 An example of Floy-Warshall’s solutions for the shortest path problem.

2. ผลการประยุกต์ใช้วิธี Q-learning เพื่อการแก้ปัญหาเส้นทางที่สั้นที่สุด

สิ่งแวดล้อมที่นำเสนอใน (Algorithm 4) นั้นแสดงความสัมพันธ์ระหว่างแอคชันและรางวัลในแต่ละสเตจ รางวัลที่ได้จากแต่ละสเตจนั้นเชื่อมโยงกับ (Table 1) และเอเจนท์ต้องการรางวัลที่สูงที่สุดในแต่ละสเตจซึ่งต้องมีกระบวนการเรียนรู้จาก (Algorithm 5) เมื่อพิจารณา (Algorithm 5) พบว่าในแต่ละรอบของกระบวนการเรียนรู้นั้น มีการสุ่มทำแอคชัน (บรรทัดที่ 7) หรือเลือกแอคชันที่ให้รางวัลสูงที่สุด (บรรทัดที่ 9) ขึ้นอยู่กับค่าความน่าจะเป็น p เมื่อกระบวนการเรียนรู้สิ้นสุดลงจะได้ Q-table ที่มีค่ารางวัลสูงที่สุดสำหรับแต่ละคู่ สเตจ-แอคชัน Q-table ดังกล่าวสามารถบอกได้ว่าข้อมูลต้องเดินทางไปยังโหนดใดจากโหนดที่อยู่ในสเตจปัจจุบัน เพื่อไปให้ถึงโหนดปลายทางด้วยต้นทุนที่ต่ำที่สุด ผลการศึกษานี้สอดคล้องกับ Buysse, Mets, & Latre (2022)

(Figure 7) แสดงรางวัลเฉลี่ยของวิธี Q-learning ต่อจำนวนรอบของการเรียนรู้ (episode) จาก (Algorithm 5) จาก (Figure 7) พบว่ารางวัลเฉลี่ยของวิธี Q-learning มีค่าเพิ่มขึ้นเมื่อจำนวนรอบของการเรียนรู้เพิ่มขึ้น ซึ่งหมายความว่าทุก ๆ รอบของการเรียนรู้วิธี Q-learning สามารถจดจำแอคชันที่ทำให้รางวัลของแต่ละสเตจมีค่าสูงขึ้น รางวัลมีค่าสูงสุดและคงที่เมื่อผ่านไปประมาณ 35 รอบของการเรียนรู้ หมายความว่าวิธี Q-learning สามารถจดจำแอคชันที่ทำให้รางวัลของแต่ละสเตจมีค่าสูงสุดและสามารถแก้ปัญหาระยะทางสั้นที่สุดได้

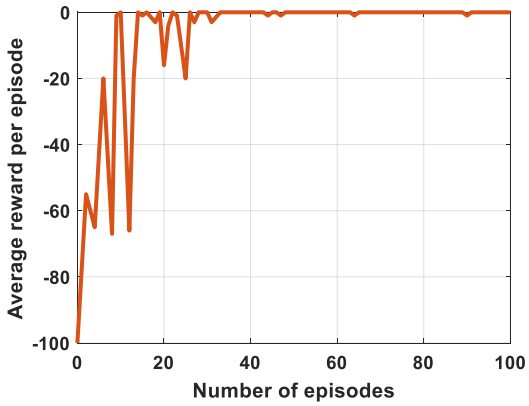

Figure 7 Average reward per episode of Q-learning.

(Figure 8) แสดงผลการแก้ปัญหาระยะทางสั้นที่สุดจากโหนด '3' ไปยังโหนด '5' ด้วยวิธี Q-learning จาก (Figure 8) พบว่าวิธี Q-learning ให้ผลเฉลยที่ถูกต้องเช่นเดียวกับทั้งสามวิธีที่ผ่านมา นอกจากนี้วิธี Q-learning ยังมีคุณลักษณะพิเศษคือสามารถจดจำเส้นทางที่สั้นที่สุดจากทุกโหนดต้นทางไปยังโหนดปลายทางได้ คุณลักษณะดังกล่าวเทียบเท่ากับการหาผลเฉลยสำหรับทุกคู่โหนดด้วยวิธีของฟลอยด์-วอร์แชลล์ อย่างไรก็ตามเมื่อเปรียบเทียบขั้นตอนการทำงานของทั้งสองวิธีจาก (Algorithm 3, 4 and 5) พบว่าวิธี Q-learning มีขั้นตอนการทำงานและการคำนวณที่น้อยกว่าวิธีของฟลอยด์-วอร์แชลล์ นอกจากนี้วิธี Q-learning ยังสามารถพัฒนาเพื่อประยุกต์ใช้ในสถานการณ์ที่การส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางมากกว่าหนึ่งคู่โหนดได้ แนวทางดังกล่าวสอดคล้องกับผลในงานวิจัย Abdoos, Mozayani, & Bazzan (2011)

```
dst node: 5
Input src node:3
Visited nodes = ['3', '0', '1', '5']
```

Figure 8 A Q-learning's solution for the shortest path problem from node '3' to node '5'.

(Figure 9) แสดงการเผยแพร่โค้ดโปรแกรมภาษา Python จำลองการทำงานของวิธีการแก้ปัญหาเส้นทางที่สั้นที่สุดทั้งสี่วิธีในเว็บไซต์ github.com ซึ่งเป็นเว็บไซต์ที่ได้รับความนิยมในหมู่นักพัฒนาภาษาคอมพิวเตอร์ ผู้ที่สนใจสามารถเข้าไปดาวน์โหลดและทดลองใช้โปรแกรมจำลองการทำงานดังกล่าวได้ตามเอกสารอ้างอิง Danuphon1999 (2023)

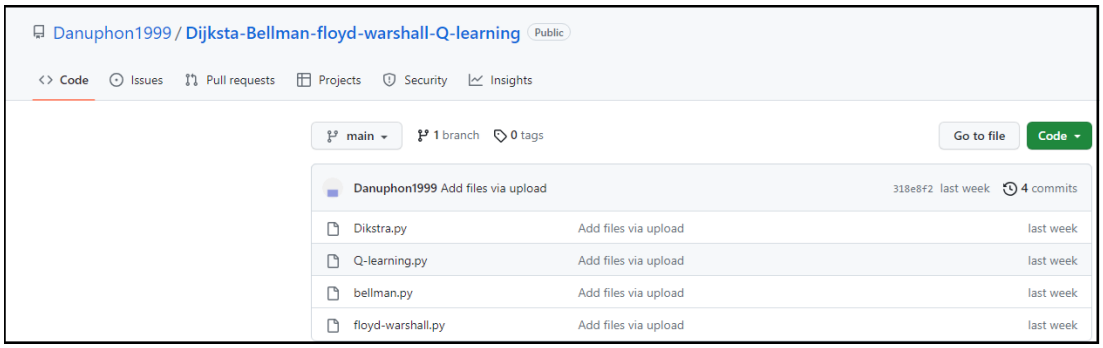

Figure 9 Publishing of Python codes for 4 algorithms on github.com.

สรุป

บทความนี้นำเสนอการประยุกต์ใช้งาน Q-learning ซึ่งจัดเป็นการเรียนรู้ของเครื่องหรือ ML วิธีหนึ่งในการแก้ปัญหาระยะทางที่สั้นที่สุดในโครงข่ายสื่อสาร การแก้ปัญหาดังกล่าวนั้นมีความสำคัญมาในการลดต้นทุนการสื่อสารซึ่งอาจเป็นกำลังส่งหรือระยะเวลาในการส่งข้อมูล ในขั้นตอนการดำเนินงานนั้นเริ่มต้นจากการศึกษาวิธีแก้ปัญหาดังกล่าวสามวิธีก่อนหน้า ได้แก่ วิธีของไดก์สตรา วิธีของเบลล์แมน-ฟอร์ด และวิธีของฟลอยด์-วอร์แชล หลังจากนั้นได้มีการนำเสนอการแปลงปัญหาระยะทางสั้นที่สุดให้เป็นองค์ประกอบของวิธี Q-learning ได้แก่ เอเจนท์ แอคชัน สเตจ รางวัล และสิ่งแวดล้อม ผลการจำลองวิธีการทั้งสี่วิธีด้วยภาษา Python แสดงให้เห็นว่าวิธีทั้งสี่สามารถแก้ปัญหาระยะทางที่สั้นที่สุดได้อย่างถูกต้อง วิธี Q-learning นั้นสามารถแก้ปัญหาระยะทางที่สั้นที่สุดจากโหนดต้นทางใด ๆ ไปยังโหนดปลายทาง ซึ่งเทียบเท่ากับผลเฉลยของวิธีของฟลอยด์-วอร์แชล อย่างไรก็ตามวิธี Q-learning นั้นมีขั้นตอนการทำงานที่ซับซ้อนน้อยกว่า ในตัวอย่างที่สนใจนั้นวิธีของฟลอยด์-วอร์แชลต้องใช้การคำนวณระยะทางของทุกคู่โหนด จำนวน 36 คู่โหนด ในขณะที่วิธี Q-learning นั้นใช้การเรียนรู้ทั้งสิ้น 35 ครั้ง การทดสอบใช้กราฟแทนโครงข่ายซึ่งมีความแม่นยำในการทดสอบวิธีต่าง ๆ นอกจากผลเฉลยและความซับซ้อนแล้ว วิธี Q-learning ยังสามารถพัฒนาต่อเพื่อใช้แก้ปัญหาระยะทางที่สั้นที่สุดในกรณีที่มีการรับ-ส่งข้อมูลมากกว่าหนึ่งคู่โหนดพร้อมกัน

แนวทางการพัฒนาวิธี Q-learning ในอนาคตคือการแก้ปัญหาเดิมโดยการเพิ่มเงื่อนไขว่ามีการรับ-ส่งข้อมูลมากกว่าหนึ่งคู่โหนดพร้อมกันในโครงข่ายสื่อสารตามที่ได้กล่าวมาข้างต้น ถึงแม้ว่าวิธี Q-learning จะมีขั้นตอนการทำงานที่ซับซ้อนน้อยกว่าวิธีของฟลอยด์-วอร์แชล อย่างไรก็ตามเมื่อพิจารณาวิธี Q-learning พบว่าต้องมีการสร้างตาราง Q-table ซึ่งตารางดังกล่าวมีขนาดที่เติบโตในเชิงเอกซ์โพเนนเชียลกับจำนวนคู่ของสเตจ-แอคชันนำมาซึ่งความซับซ้อนที่เพิ่มขึ้นตามที่ได้มีการศึกษาในงานวิจัย Aermsa-Ard, Wangsamad, & Mamat (2023) การลดความซับซ้อนดังกล่าวเป็นปัญหาที่น่าสนใจและสามารถพัฒนาต่อได้ในอนาคต

เอกสารอ้างอิง

Abdoos, M., Mozayani, N., & Bazzan, A. L. C. (2011). Traffic light control in non-stationary environments based on multi agent Q-learning. *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 1580-1585). Washington, D.C.: IEEE.

Aermsa-Ard, P., Wangsamad, C., & Mamat, K. (2023). On applying Q-learning to optimize power allocation in 2-user NOMA system. *The journal of Industrial Technology*, 19(1), 104-116.

Al-Tam, F., Correia, N., & Rodruguez., J. (2020). Learn to schedule (LEACH): A deep reinforcement learning approach for radio resource scheduling in the 5G MAC layer. *IEEE Access, 8*, 108088-108101.

Aneja, Y. P. (2001). Routing in wavelength routed optical networks. *IEEE Workshop on High Performance Switching and Routing* (pp. 155-158). Dallas, TX: IEEE.

Buysse, M. H., Mets, K., & Latre, S. (2022). Hierarchical reinforcement learning: A survey and open research challenges. *Machine Learning and Knowledge extraction*, *4*(1), 172-221.

Chanda, A., Rout, R. R., & Lingam, G. (2018). An Improvement Friendship-based Routing Algorithm in Mobile Social Networks. *13th International Conference on Industrial and Information System (ICIIS)* (pp. 286-291). Rupnagar, India: IEEE.

Chen, W., He, R., Wang, G., Zhang, J., Wang, F., Xiong, K., Ai, B., & Zhong, Z. (2021). AI assisted PHY in future wireless systems: Recent developments and challenges. *China Communications*, *18*(5), 285-297.

Danuphon1999. (2023). *Dijksta-Bellman-floyd-warshall-Q-learning.* Retrieved 16 October 2023, from https://github.com/Danuphon1999/Dijksta-Bellman-floyd-warshall-Q-learning?search=1

Forouzan, B. A. (2007). *Data communications and networking* (4th ed.). Boston Burr Ridge: McGraw-Hill.

Javaid, A. (2013). Understanding Dijkstra algorithm. *Electronic Journal*, *1*, 1-27.

Li, R., Zheng, C., & Zhang, Y. (2011). Study of power-aware routing protocol in wireless sensor networks. *International Conference on Electrical and Control Engineering* (pp. 3173-3176). Yichang: IEEE.

Mete, E., & Girici, T. (2020). Q-learning based scheduling with successive interference cancellation. *IEEE Access, 8*, 172034-172042.

Mukhlif, F., & Safi, A. (2020). Comparative study on Bellman-Ford and Dijkstra algorithms. *International Conference on Communication, Electrical and Computer Networks (ICCECN 2020)* (pp.1-5). Kuala Lumpur: University of Malaya.

Nguyen, D. C., Peng, C., Ming, D., Lopez-Perez, D. Pathirana, P. N., Li, J., Aruna, S., Younghui, L., & Poor, H. V. (2021). Enabling AI in Future Wireless Networks : A Data Life Cycle Perspective. *IEEE Communication Surveys & Tutorial,* 23(1), 553-595.

Pryshchenko, O., Dumin, O., Plakhtii, V., Shyrokorad, D., & Pochanin, G. (2021). Collective artificial intelligence approach for the problem of object classification with UWB GPR. *IEEE 26th International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED)* (pp. 185-190). Tbilisi: IEEE.

Raza, A. D., & Muhammad, S. S. (2015). Achievable capacity region of a Gaussian optical wireless relay channel. *Journal of Optical Communications and Networking*, *7*(12), 83-95.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). Cambridge, MA: MIT Press.

Wilson, R. J. (1996). *Introduction to graph theory* (4th ed.). Harlow: Addison Wesley Longman.

Wuttisittikulkij, L. (2016). *Principle of Telecommunication Engineering* (2nd ed.). Bangkok: Chulalongkorn University Press.