บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีพื้นฐานที่เกี่ยวข้องกับการพัฒนาเรื่อง การจัดการคลังสินค้าสำหรับค้นหาเส้นทาง ซึ่งอาศัยหลักการค้นหาเส้นทางที่สั้นที่สุด ได้นำทฤษฎีและงานวิจัยที่เกี่ยวข้องเข้ามาช่วยในการพัฒนาโปรแกรมเพื่อเพิ่มประสิทธิภาพของการทำงาน โดยงานวิจัยได้นำวิธีการและเทคนิควิธีการต่างๆ ดังนี้

1) อัลกอริทึมค้นหาเส้นทางที่สั้นที่สุด (Dijkstra's Algorithm)

2) อัลกอริทึมของระบบมด (Ant Colony Algorithm)

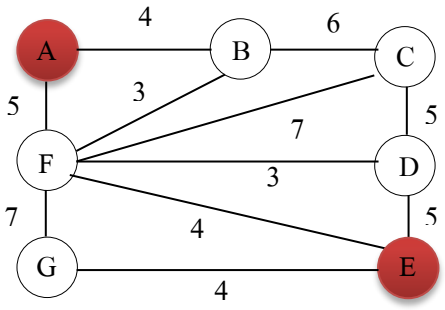
3) การแทนที่กราฟด้วยเมตริกซ์ (Adjacency Matrix)

4) งานวิจัยที่เกี่ยวข้อง

อัลกอริทึมค้นหาเส้นทางที่สั้นที่สุด (Dijkstra's Algorithm)

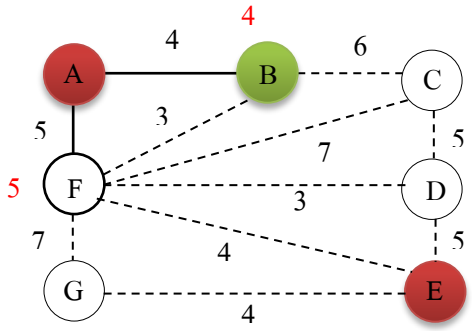
เป็นขั้นตอนวิธีที่ใช้ในการหาเส้นทางที่สั้นที่สุด ซึ่งใช้เวลาในการค้นหาเส้นทางที่สั้นที่สุดและมีขั้นตอนที่ไม่ซับซ้อน โดยสามารถนำ Dijkstra’s Algorithm มาประยุกต์ใช้ในงานต่างๆได้ เช่นการค้นหาเส้นทางที่สั้นที่สุดบนแผนที่ หรือไปประยุกต์ใช้ในงานทางด้านเครือข่ายคอมพิวเตอร์ Network ได้เป็นต้น (รศ.ดร.อนงนาฏ ศรีวิหค และคณะ, 2548.)

ตัวอย่าง การหาเส้นทางที่สั้นที่สุด จากจุด A ไปยัง จุด E โดยวิธี Dijkstra's Algorithm



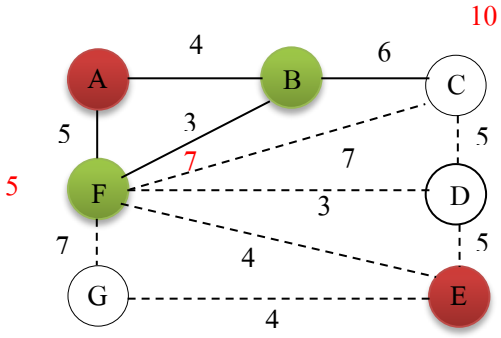
ภาพที่ 2.1 กำหนดจุดเริ่มต้น และจุดปลายทาง

จากภาพที่ 2.1 กำหนดจุดเริ่มต้น คือจุด A และจุดปลายทางคือ จุด E ด้วยวงกลมทึบซึ่งได้กำหนดค่าระยะทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง



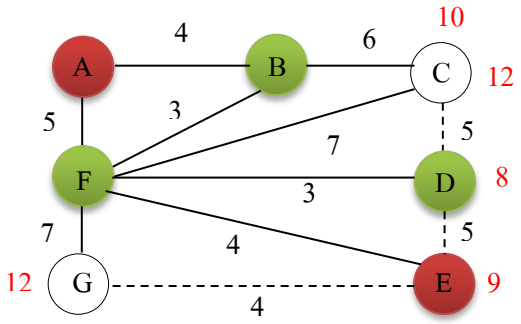
ภาพที่ 2.2 พิจารณาเส้นทางจากจุดเริ่มต้น

จากภาพที่ 2.2 พิจารณาเส้นทางที่อยู่ใกล้กับจุดเริ่มต้น โดยเส้นทางที่ได้พิจารณาจะเป็นเส้นทึบ และเส้นทางที่ไม่ได้พิจารณาจะเป็นเส้นประ ซึ่งมีเส้นทางจากจุด A ไปยังจุด B, จุด A ไปยังจุด F ซึ่งมีค่าระยะทางรวม 4 และ 5 ตามลำดับ ดังนั้นจึงเลือกเส้นทางจากจุด A ไปยังจุด B โดยกำหนดเป็นวงกลมทึบ



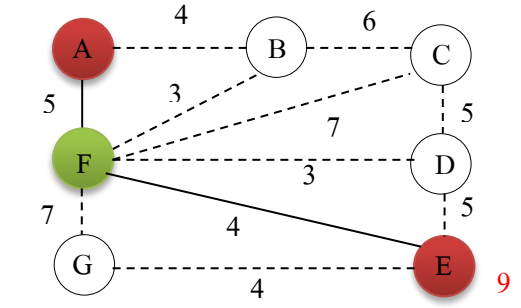
ภาพที่ 2.3 พิจารณาเส้นทางใหม่ต่อจากจุด B

จากภาพที่ 2.3 พิจารณาเส้นทางต่อจากจุด B ซึ่งได้กำหนดเส้นทางที่พิจารณาเป็นเส้นทึบ และเส้นทางที่ไม่ได้พิจารณาเป็นเส้นประ คือ เส้นทางจากจุด B ไปยังจุด C, เส้นทางจากจุด B ไปยังจุด F และเส้นทางจากจุด A ไปยังจุด F ซึ่งได้ค่าระยะทางรวมแต่ละเส้นทาง ดังนี้ 10, 7 และ5 ตามลำดับ ดังนั้น จึงเลือกเส้นทางจากจุด A ไปยังจุด F โดยกำหนดเป็นวงกลมทึบ



ภาพที่ 2.4 พิจารณาเส้นทางใหม่ต่อจากจุด F

จากภาพที่ 2.4 พิจารณาเส้นทางจากจุด F ซึ่งได้กำหนดเส้นทางที่พิจารณาเป็นเส้นทึบ และเส้นทางที่ไม่ได้พิจารณาเป็นเส้นประ คือ เส้นทางจากจุด F ไปยังจุด B และจุด C, เส้นทางจากจุด F ไปยังจุด C, เส้นทางจากจุด F ไปยังจุด D, เส้นทางจากจุด F ไปยังจุด E และเส้นทางจากจุด F ไปยังจุด G ซึ่งมีค่าระยะทางรวมแต่ละเส้นทาง คือ 14, 12, 8, 9, 12 ตามลำดับ



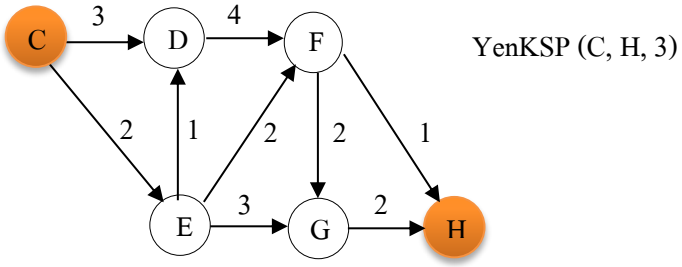
ภาพที่ 2.5 แสดงเส้นทางที่สั้นที่สุด

จากภาพที่ 2.5 ดังนั้นจึงเลือกเส้นทางจากจุด A ไปยังจุด F และไปยังจุด E เป็นเส้นทางที่สั้นที่สุด เนื่องจากเป็นการหาเส้นทางจุดเริ่มต้นไปยังจุดปลายทางที่สั้นที่สุดซึ่งได้ค่าระยะทางรวมเท่ากับ 9

อัลกอริทึมค้นหาเส้นทาง (K-Shortest Paths)

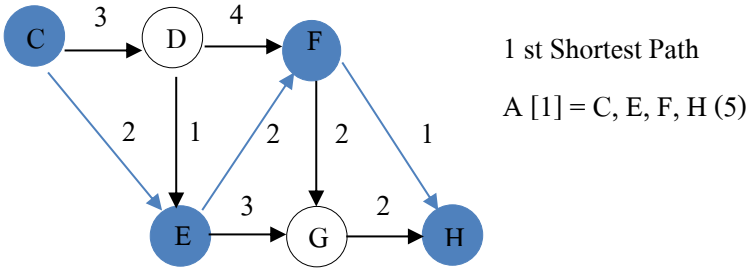
การศึกษาครั้งนี้อัลกอริทึมเส้นทางที่สั้นที่สุด (มาร์ติน, EQ และ MM Pascoal, 2003) ได้รับการดำเนินการสร้างการเดินทาง มันเป็นขั้นตอนวิธีการขยายของเส้นทางเส้นทางที่สั้นที่สุดอัลกอริทึม (Cherkassky, BV et al., 1996) ในเครือข่ายที่ได้รับ อัลกอริทึมไม่เพียง แต่พบเส้นทางที่

สั้นที่สุดระหว่างสองจุด แต่ยังสามารถหาเส้นทางที่สั้นที่สุดได้หลายเส้นทาง อัลกอริทึมจึงถูกนำมาใช้อย่างแพร่หลายในด้านของการสื่อสารโทรคมนาคมการดำเนินงานวิจัยด้านวิทยาการคอมพิวเตอร์และวิทยาศาสตร์การขนส่งเครือข่าย แสดงดังตัวอย่าง



ภาพที่ 2.6 จุดเริ่มต้นและจุดปลายทาง

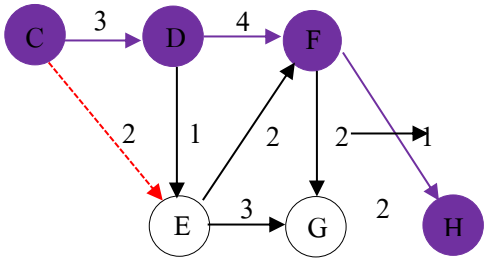
จากภาพที่ 2.6 กำหนดจุดเริ่มคือ จุด C และจุดปลายทางคือ จุด H ด้วยวงกลมทึบซึ่งได้กำหนดค่าระยะทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง



ภาพที่ 2.7 เส้นทางที่สั้นที่สุดลำดับที่ 1

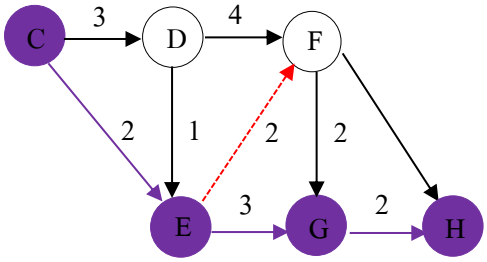
จากภาพที่ 2.7 ทำการหาเส้นทางที่สั้นที่สุดโดยวิธีการ Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุดลำดับที่ 1โดยกำหนดเป็นวงกลมทึบ คือ C, E, F, H รวมระยะทางเท่ากับ 5

ทำการหาเส้นทางที่สั้นที่สุดลำดับที่ 2 โดย K - Shortest Paths โดยใช้วิธีลบเส้นทางทีละเส้นจาก 1’ st Shortest Paths โดยกำหนดเส้นทางที่ลบออกเป็นเส้นประ ดังนั้น จึงได้เส้นทางใหม่อีก 3 เส้นทาง แสดงดังรูป 2.8, 2.9 และ 2.10 ตามลำดับ



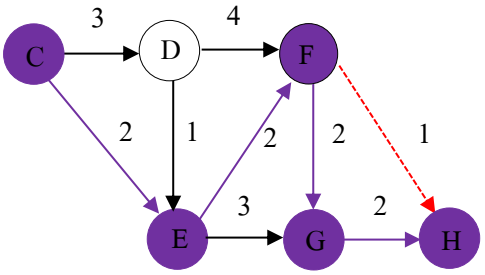
ภาพที่ 2.8 การตัดเส้นทาง C ไปยัง E

จากภาพที่ 2.8 ลบเส้นทางจากจุด C ไป E โดยกำหนดเป็นเส้นประแล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, D, F, H รวมระยะทางเท่ากับ 8 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ



ภาพที่ 2.9 การตัดเส้นทาง E ไปยัง F

จากภาพที่ 2.9 ลบเส้นทางจากจุด E ไป F แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, E, G, H รวมระยะทางเท่ากับ **7** โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ

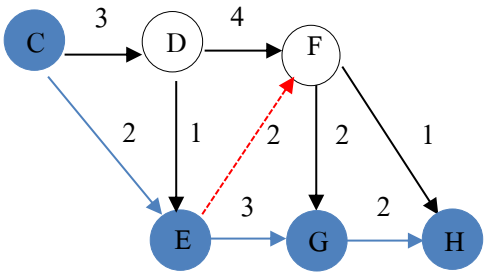


B = C, D, F, H (8)

B+ = C, E, G, H (7)

ภาพที่ 2.10 การตัดเส้นทาง F ไปยัง H

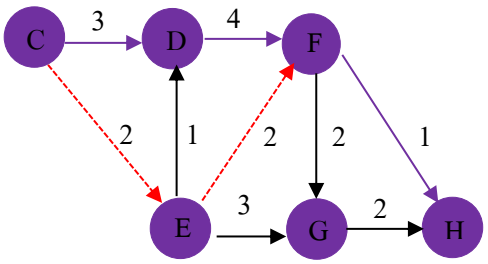
จากภาพที่ 2.10 ลบเส้นทางจากจุด F ไป H แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, E, F, G, H รวมระยะทางเท่ากับ 8 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ



ภาพที่ 2.11 เส้นทางที่สั้นที่สุดลำดับที่ 2

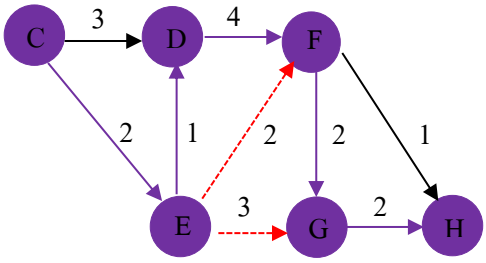
จากภาพที่ 2.11 ลบเส้นทางจากจุด E ไป F แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุดลำดับที่ 2 คือ C, E, G, H รวมระยะทางเท่ากับ 7 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ

จากนั้นนำเส้นทางที่สั้นที่สุดลำดับ 2 มาทำการหา 2'nd Shortest Paths เส้นทางที่สั้นที่สุดลำดับที่ 3 โดย K- Shortest Path โดยใช้วิธีลบเส้น 2 เส้น โดยลบเส้นทาง E ไป F เป็นหลัก ซึ่งกำหนดเส้นทางที่ลบออกให้เป็นเส้นประ ดังนั้น จึงได้เส้นทางใหม่อีก 3 เส้นทาง แสดงดังภาพที่ 2.12 2.13 และ 2.14 ตามลำดับ



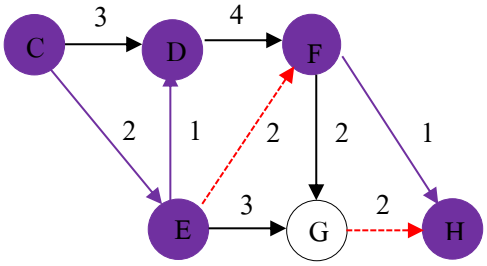
ภาพที่ 2.12 การตัดเส้นเส้นทาง E ไป F, C ไป E

จากภาพที่ 2.12 ลบเส้นทางจากจุด C ไป E แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, D, F, H รวมระยะทางเท่ากับ 8 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ



ภาพที่ 2.13 การตัดเส้นทาง E ไป F, E ไป G

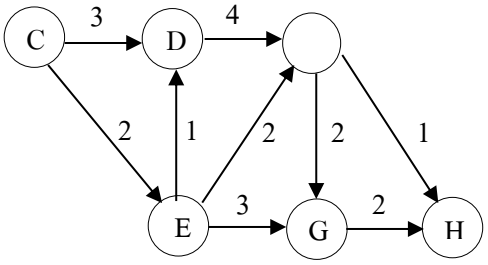
จากภาพที่ 2.13 ตัดเส้นทางจากจุด E ไป G แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, E, D, F, G, H รวมระยะทางเท่ากับ 11 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ



ภาพที่ 2.14 การตัดเส้นทาง E ไป F, G ไป H

จากภาพที่ 14 ตัดเส้นทางจากจุด G ไป H แล้วทำการหาเส้นทางใหม่โดยวิธี Dijkstra's Algorithm จึงได้เส้นทางที่สั้นที่สุด คือ C, E, D, F, H รวมระยะทางเท่ากับ 8 โดยกำหนดจุดเส้นทางใหม่เป็นวงกลมทึบ

ทำการหาเส้นทางที่สั้นที่สุดลำดับที่ 3 โดยการนำเส้นทางที่เหลือจากการหาเส้นทางลำดับที่ 2 จากภาพที่ 2.8 และภาพที่ 2.10 ไปทำการเปรียบเทียบกับเส้นทางใหม่ที่ได้จากการหาเส้นทางที่สั้นที่สุดลำดับที่ 3



ภาพที่ 2.15 เส้นทางที่สั้นที่สุดลำดับ 3

ดังนั้นจึงได้เส้นทางที่สั้นที่สุด 3 ลำดับเส้นทาง คือ

เส้นทางที่สั้นที่สุดลำดับที่ 1 C, E, F, H ระยะทางรวมเท่ากับ 5

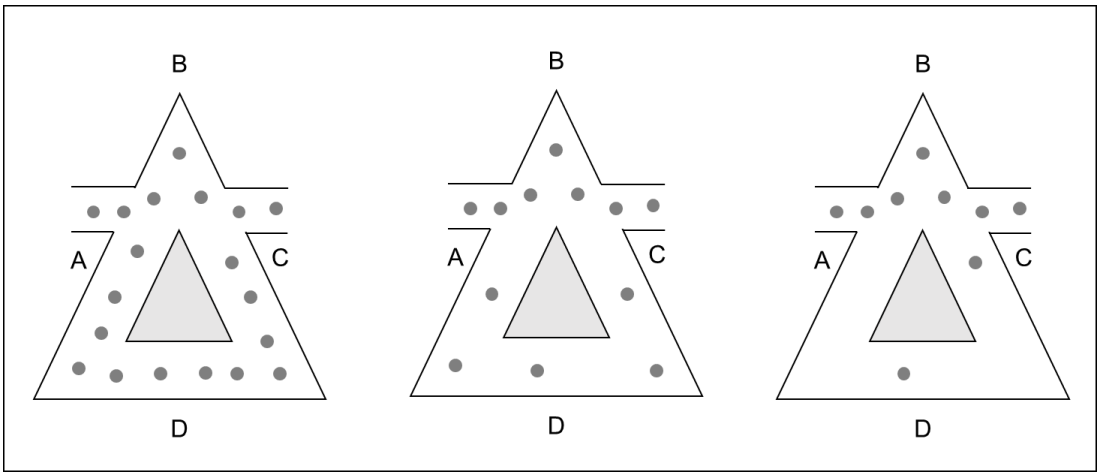
เส้นทางที่สั้นที่สุดลำดับที่ 2 C, E, G, H ระยะทางรวมเท่ากับ 7

เส้นทางที่สั้นที่สุดลำดับที่ 3 C, D, F, H ระยะทางรวมเท่ากับ 8

อัลกอริทึมของระบบมด (Ant Colony Algorithm)

เป็นอัลกอริทึมที่จำลองหรือได้แนวคิดมาจากการเดินหาอาหารของมด โดยมดจะอาศัยสารเคมีที่เรียกวาฟีโรโมน (Pheromone) ที่มดแต่ละตัวก่อนหน้านี้ปล่อยลงบนพื้นและเมื่อมดตัวหลังเดินตามมาก็จะปรับปรุงฟีโรโมนลงไปบนพื้นอีก ซึ่งเหตุนี้เองฟีโรโมนจึงเป็นข้อมูลที่สำคัญในการหาเส้นทางจากแหล่งอาหารกลับไปยังรัง ซึ่งอธิบายพฤติกรรมของมดโดยใช้ข้อมูเรื่องปริมาณของฟีโรโมนในการหาเส้นทางเดินจากรังไปยังแหล่งอาหาร จุดเด่นของระบบมด ก็คือ เป็นขั้นตอนวิธีที่สามารถหาผลเฉลยได้มากกว่า 1คำตอบ ทำให้ได้เส้นทางที่หลากหลาย เมื่อเส้นทางที่เหมาะสมที่สุดเกิดมีปัญหา เช่น เนื่องจากเส้นทางชำรุด หรือการจราจรหนาแน่นจะสามารถรู้ว่าเส้นทางที่เหมาะสมเส้นรองต่อไปคือเส้นทางใด โดยที่ไม่ต้องทำการคำนวณหาอีกครั้ง ต่ละวิธีนั้นก็จะมีขั้นตอนการปรับค่าฟีโรโมน หรือการค้นหาเส้นที่จะเดินต่อแตกต่างกันไป และคำตอบที่มดแต่ละตัวหาได้นั้นจะเรียกว่า ส่วนของคำตอบ เมื่อได้ข้อมูลมากเพียงพอเราสามารถนำข้อมูลภายในเซตของส่วนของคำตอบมาจัดเรียงต่อกัน ก็จะได้เส้นทางที่มีฟีโรโมนมากที่สุดถือเป็นเส้นทางที่เหมาะสมที่สุดในการทำงาน

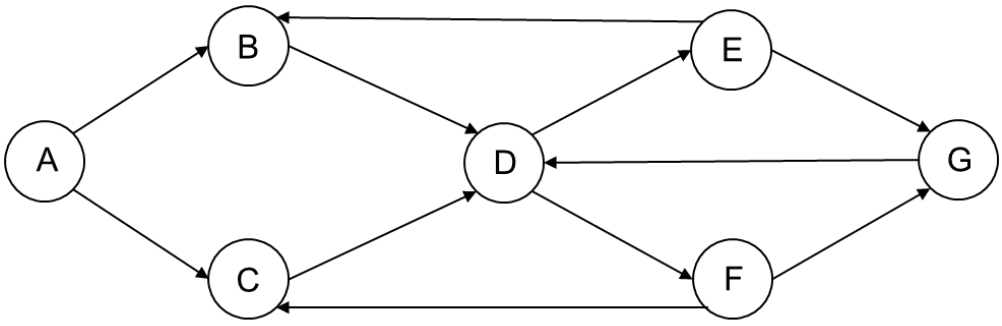
โดยวิธีการนี้สามารถนำไปประยุกต์ใช้เพื่อแก้ไขปัญหาได้ค่อนข้างหลากหลาย เช่น การหาเส้นทางที่สั้นที่สุดในเครื่องข่าย การจัดตารางสอน และจัดลำดับความสำคัญของรายการหรือข้อมูลที่เราต้องการ



ภาพที่ 2.16 รูปภาพแสดงวิธีการเดินของมด

การแทนที่กราฟด้วยเมตริกซ์ (Adjacency Matrix)

โครงสร้างของกราฟเป็นโครงสร้างที่ประกอบไปด้วยโหนดและเส้นเชื่อมต่อที่บอกถึงเส้นทางของการเดินทาง หรือความสัมพันธ์ในทิศทางซึ่งสามารถนำมาแทนความสัมพันธ์นั้นด้วยเมตริกซ์ได้ด้วยการกำหนดเมตริกซ์ n x n



ภาพที่ 2.17 กราฟแบบไดกราฟ (Direct Graph)

จากภาพที่ 2.17 หากนำมาแทนที่ด้วยเมตริกซ์จะแทนที่ได้ด้วยจำนวน n x n ซึ่ง n ก็คือ A,B,C,D,E,F เป็น 6 จำนวน โหนดทำให้ได้ค่าเมตริกซ์ จำนวน 6 x 6 = 36 ช่อง และเมื่อพิจารณาตามทิศทางของการเชื่อมต่อ หากโหนดใดมีเส้นเชื่อมต่อไปยังโหนดอื่นให้ระบุตัวเลขลง ไปในเมตริกซ์เป็น 1 แต่ถ้าโหนดใดไม่ได้มีการระบุว่าทำการเชื่อมต่อก็ให้ระบุเป็น 0 ดังภาพ

	A	B	C	D	E	F	G
A	0	0	0	1	0	1	0
B	0	0	0	0	1	0	0
C	0	0	0	0	1	0	0
D	0	0	0	0	0	1	0
E	0	0	0	0	0	0	1
F	0	0	0	0	1	0	0
G	0	0	0	0	0	0	0

ภาพที่ 2.18 การแทนที่กราฟด้วยเมตริกซ์

จากภาพที่ 2.18 แทนที่เมตริกซ์ด้วยจำนวนโหนดทั้งด้านแนวตั้งและแนวนอนโดยกำหนดให้แนวนอนเป็นโหนดต้นทางและแนวตั้งเป็นโหนดปลายทาง สามารถเขียนเลขลำดับเมตริกซ์โดยระบุค่าดังนี้ พิจารณาโหนด A มีเส้นเชื่อมต่อระบุไป B ระบุหมายเลข 1 นอกนั้นระบุค่าเป็น 0 พิจารณาโหนด B มีเส้นเชื่อมต่อระบุไป C, E ระบุหมายเลข 1 นอกนั้นระบุค่าเป็น 0 พิจารณาโหนด C มีเส้นเชื่อมต่อระบุไป D, E ระบุหมายเลข 1 นอกนั้นระบุค่าเป็น 0 พิจารณาโหนด D ไม่มีเส้นเชื่อมต่อระบุไปยังโหนดอื่นระบุหมายเลข 0 ทุกช่อง พิจารณาโหนด E มีเส้นเชื่อมต่อระบุไป D, F ระบุหมายเลข 1 นอกนั้นระบุค่าเป็น 0 พิจารณาโหนด F ไม่มีเส้นเชื่อมต่อระบุไปยังโหนดอื่นระบุหมายเลข 0 ทุกช่อง

งานวิจัยที่เกี่ยวข้อง

การเปรียบเทียบหาเส้นทางที่เหมาะสมโดยวิธีระบบมดและ Dijkstra’s Algorithm งานวิจัยนี้มีวัตถุประสงค์เพื่อนำวิธีระบบมด (Ant system algorithm) มาเปรียบเทียบกับ Dijkstra’s algorithm ว่าวิธีระบบทำงานใดมีความเหมาะสมที่สามารถนำมาใช้ในการ ค้นหาเส้นทางที่

เหมาะสมในสภาพปัจจุบันได้ซึ่งแนวคิดได้ถูกนำมาใช้ในการค้นหาเส้นทางที่เหมาะสมในสภาพปัจจุบัน ถึงแม้ว่าเส้นทางนั้นจะเป็นเส้นทางที่สั้นที่สุดก็ตามแต่ไม่ได้หมายถึงว่าจะเป็นเส้นทางที่ดีได้เสมอไป ซึ่งการเปรียบเทียบ ได้จำลองสถานการณ์ของเส้นทางเป็นในรูปแบบของกราฟ แบบไม่มีทิศทาง (Undirected graph) ซึ่งจุดเชื่อม (Arcs) โหนดจะมีน้ำหนัก (Weighted graph) ด้วย ซึ่งและมีค่าเป็นบวกเสมอโดย จำลองสถานการณ์กราฟจำนวน 20 แบบ โดยแต่ละแบบถูก จำลองมาจากสภาพถนนและลักษณะกราฟต่างๆโดยโหนดจะหมายถึงสถานที่สำคัญหรือทางแยก ซึ่งการเปรียบเทียบการ ประเมินคุณภาพโดยดูจากการหาผลเฉลยของคำตอบระหว่างวิธี ระบบมดกับ Dijkstra's algorithm โดยดูจากการทดลองจาก กราฟจำนวน 20 แบบ ว่าวิธีแต่ละวิธีให้ผลเฉลยการค้นหาเส้นทางแบบใดบ้าง จากการทดลองสรุปได้ว่า Dijkstra's algorithm ให้ผลเฉลยการค้นหาเส้นทางที่สั้นที่สุดเพียงเส้นทาง เดียวซึ่งเส้นทางที่ค้นหาได้เป็นเส้นทางที่สั้นที่สุดเท่านั้น ส่วนวิธีระบบมดมีการค้นหาให้ผลเฉลยการค้นหาเส้นทางที่หลากหลายซึ่งให้ผลเฉลยได้เส้นทางที่เหมาะสมที่สุดและได้ผล เฉลยที่เป็นเส้นทางรองด้วย

การเลือกเส้นทางปลอดภัยที่เหมาะที่สุดสำหรับวางแผนการเดินทาง ในแต่ละปีอุบัติเหตุบนท้องถนนในประเทศไทยได้ก่อให้เกิดจำนวน ผู้เสียชีวิตและบาดเจ็บจำนวนมาก และไม่ได้มีแนวโน้มที่จะลดลง แต่อย่างใด ดังนั้นหากสามารถวางแผนการเดินทางเพื่อหลีกเลี่ยง ถนนที่มีความเสี่ยงสูง ก็อาจทำให้อัตราการเกิดอุบัติเหตุบนท้อง ถนนลดลงได้ การศึกษานี้เป็นการพัฒนาวิธีการเพื่อเลือกเส้นทางที่ ปลอดภัยที่ดีที่สุดสำหรับการเดินทางจากจุดหนึ่งไปยังจุดอื่นด้วย การใช้อัลกอริทึมของ Dijkstra เป็นพื้นฐานแล้วเพิ่มปัจจัยอื่น ๆ ที่เกี่ยวกับอุบัติเหตุบนท้องถนน คือค่าดัชนีความรุนแรงของถนนและ ทางแยก จากนั้นจึงทำการวิเคราะห์เพื่อให้ได้เส้นทางที่เหมาะสม ที่สุด เส้นทางนี้อาจถือได้ว่าเป็นเส้นทางที่น่าจะปลอดภัยที่สุด เพราะถนนและทางแยกที่มีความรุนแรงมากกว่าจะถูกตัดทิ้งไป นอกจากนี้ ยังได้ทำการดัดแปลง อัลกอริทึมของ Dijkstra ให้รวม เอาทางแยกและถนนที่อยู่ถัดไปที่เชื่อมต่อกับทางแยกที่กำลัง พิจารณา การทดสอบด้วยโครงข่ายขนาด 6 โหนด พบว่า อัลกอริทึมที่ปรับปรุงใหม่ทั้งสองให้ผลลัพธ์ไม่ต่างกันนัก แต่เมื่อ โครงข่ายมีขนาดใหญ่ขึ้น พบว่า ทั้ง 2 วิธีให้ผลลัพธ์ที่แตกต่างกัน ขึ้นอยู่กับลักษณะของโครงข่าย