

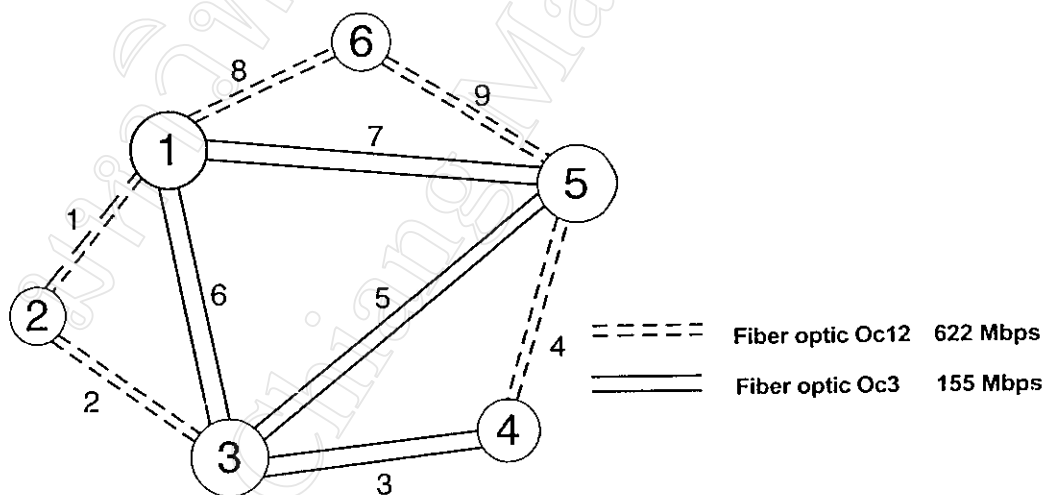
บทที่ 3

ระบบการหาตารางเส้นทางโดยใช้จูนิตัลกอริทึม และเครือข่ายประสาท

ระบบการหาตารางเส้นทางโดยใช้จูนิตัลกอริทึมและเครือข่ายประสาท แบ่งการทำงานออกเป็นสองส่วนคือ การหาตารางเส้นทางที่เหมาะสมโดยใช้จูนิตัลกอริทึม และการช่วยลดเวลาการคำนวณของกระบวนการจูนิตัลกอริทึมทั้งแบบที่ใช้วิธีเครือข่ายประสาท และแบบที่ใช้วิธี Past Selection เพื่อการเปรียบเทียบ โดยระบบทั้งหมดถูกจำลองการทำงานไว้บนโปรแกรมคอมพิวเตอร์ (MATLAB)

3.1 การแทนตารางเส้นทางในรูปโครงโมโซม

การแทนตารางเส้นทางให้อยู่ในรูปโครงโมโซม เริ่มต้นจากการหาเส้นทางที่เป็นไปได้ทั้งหมดระหว่างคูโหนดทั้งหมด ในงานวิจัยนี้ใช้เครือข่ายทดสอบที่ 6 โหนด 9 ลิงค์ ดังรูปที่ 1.2 โดยการกำหนดหมายเลขโหนดและหมายเลขลิงค์แสดงดังรูปที่ 3.1 เครือข่ายทดสอบนี้จะมีเส้นทางที่เป็นไปได้ทั้งหมดระหว่างคูโหนดใด ๆ แสดงดังตารางที่ 3.1 – 3.6



รูปที่ 3.1 การกำหนดหมายเลขโหนดและหมายเลขลิงค์ของเครือข่ายทดสอบ

ตารางที่ 3.1 เส้นทางระหว่างโหนด 1 ไปยังโหนดใดๆ

	1-2	1-3	1-4	1-5	1-6
1	1-2	1-3	1-2-3-4	1-5	1-2-3-4-5-6
2	1-6-2	1-2-3	1-2-3-5-4	1-2-3-4-5	1-2-3-5-6
3	1-5-4-3-2	1-5-4-3	1-3-4	1-3-4-5	1-3-4-5-6
4	1-5-3-2	1-5-3	1-3-5-4	1-3-5	1-3-5-6
5	1-6-5-3-2	1-6-5-4-3	1-5-4	1-2-3-5	1-5-6
6	1-6-5-4-3-2	1-6-5-3	1-5-3-4	1-6-5	1-6
7	-	-	1-6-5-4	-	-
8	-	-	1-6-5-3-4	-	-

ตารางที่ 3.2 เส้นทางระหว่างโหนด 2 ไปยังโหนดใดๆ

	2-1	2-3	2-4	2-5	2-6
1	2-1	2-3	2-3-4	2-1-3-4-5	2-1-6
2	2-3-1	2-1-3	2-3-5-4	2-1-5	2-1-5-6
3	2-3-4-5-1	2-1-5-3	2-1-5-4	2-1-3-5	2-1-3-4-5-6
4	2-3-5-1	2-1-5-4-3	2-1-3-4	2-1-6-5	2-1-3-5-6
5	2-3-5-6-1	2-1-6-5-3	2-1-5-3-4	2-3-5	2-3-4-5-6
6	2-3-4-5-6-1	2-1-6-5-4-3	2-1-6-5-4	2-3-4-5	2-3-5-6
7	-	-	2-1-6-5-3-4	-	-

ตารางที่ 3.3 เส้นทางระหว่างโหนด 3 ไปยังโหนดใดๆ

	3-1	3-2	3-4	3-5	3-6
1	3-1	1-3	3-4	3-5	3-1-6
2	3-2-1	3-1-2	3-5-4	3-4-5	3-4-5-6
3	3-4-5-1	3-5-1-2	3-2-1-5-4	3-2-1-5	3-5-6
4	3-5-1	3-4-5-1-2	3-2-1-6-5-4	3-2-1-6-5	3-1-5-6
5	3-4-5-6-1	3-5-6-1-2	3-1-5-4	3-1-5	3-2-1-6
6	3-5-6-1	3-4-5-6-1-2	3-1-6-5-4	3-1-6-5	3-2-1-5-6

ตารางที่ 3.4 เส้นทางระหว่างโหนด 4 ไปยังโหนดใดๆ

	4-1	4-2	4-3	4-5	4-6
1	4-3-2-1	4-3-2	4-3	4-5	4-5-6
2	4-5-3-2-1	4-5-3-2	4-5-3	4-3-5	4-3-5-6
3	4-3-1	4-5-1-2	4-5-1-2-3	4-3-1-5	4-3-1-6
4	4-5-3-1	4-3-1-2	4-5-6-1-2-3	4-3-1-6-5	4-3-2-1-6
5	4-5-6-1	4-3-5-1-2	4-5-1-3	4-3-2-1-5	4-3-2-1-5-6
6	4-3-5-1	4-5-6-1-2	4-5-6-1-3	4-3-2-1-6-5	4-3-1-5-6
7	4-5-6-1	4-3-5-6-1-2	-	-	-
8	4-3-5-6-1	-	-	-	-

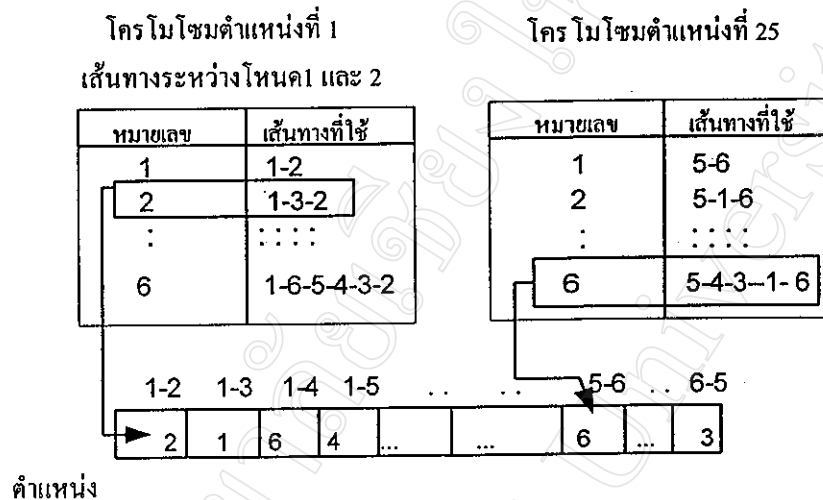
ตารางที่ 3.5 เส้นทางระหว่างโหนด 5 ไปยังโหนดใดๆ

	5-1	5-2	5-3	5-4	5-6
1	5-1	5-4-3-1-2	5-3	5-4	5-6
2	5-4-3-2-1	5-1-2	5-4-3	5-3-4	5-1-6
3	5-4-3-1	5-3-1-2	5-1-2-3	5-1-3-4	5-3-2-1-6
4	5-3-1	5-6-1-2	5-6-1-2-3	5-6-1-3-4	5-3-1-6
5	5-3-2-1	5-3-2	5-1-3	5-1-2-3-4	5-4-3-2-1-6
6	5-6-1	5-4-3-2	5-6-1-3	5-6-1-2-3-4	5-4-3-1-6

ตารางที่ 3.6 เส้นทางระหว่างโหนด 6 ไปยังโหนดใดๆ

	6-1	6-2	6-3	6-4	6-5
1	6-5-4-3-2-1	6-1-2	6-1-3	6-5-4	6-5
2	6-5-3-2-1	6-5-1-2	6-5-4-3	6-5-3-4	6-1-5
3	6-5-4-3-1	6-5-4-3-1-2	6-5-3	6-1-3-4	6-1-2-3-5
4	6-5-3-1	6-5-3-1-2	6-5-4-3	6-1-2-3-4	6-1-3-5
5	6-5-1	6-5-4-3-2	6-4-2-3	6-5-1-2-3-4	6-1-2-3-4-5
6	6-1	6-5-3-2	6-5-1-2-3	6-5-1-3-4	6-1-3-4-5

จากตารางที่ 3.1-3.6 เครือข่ายนี้มีเส้นทางทั้งหมด 30 เส้นทางแต่ละเส้นทางมีทางเลือกตั้งแต่หกถึงแปดวิธี เพื่อความเหมาะสมกับปัญหงานวิจัยนี้ใช้การแทนโครโมโซมแบบจำนวนเต็มเนื่องจากในแต่ละเส้นทางมีจำนวนวิธีการเลือกเส้นทางไม่เท่ากัน ถ้าใช้การแทนแบบไบนารีอาจเกิดปัญหาในขั้นตอนของการมิวเตชัน ดังนั้นโครโมโซมที่ใช้เป็นจำนวนเต็มมีค่าตั้งแต่หนึ่งถึงแปดความยาว 30 ตำแหน่ง ซึ่งสามารถแทนเส้นทางทั้งหมดของเครือข่าย การแทนโครโมโซมแสดงดังรูปที่ 3.2



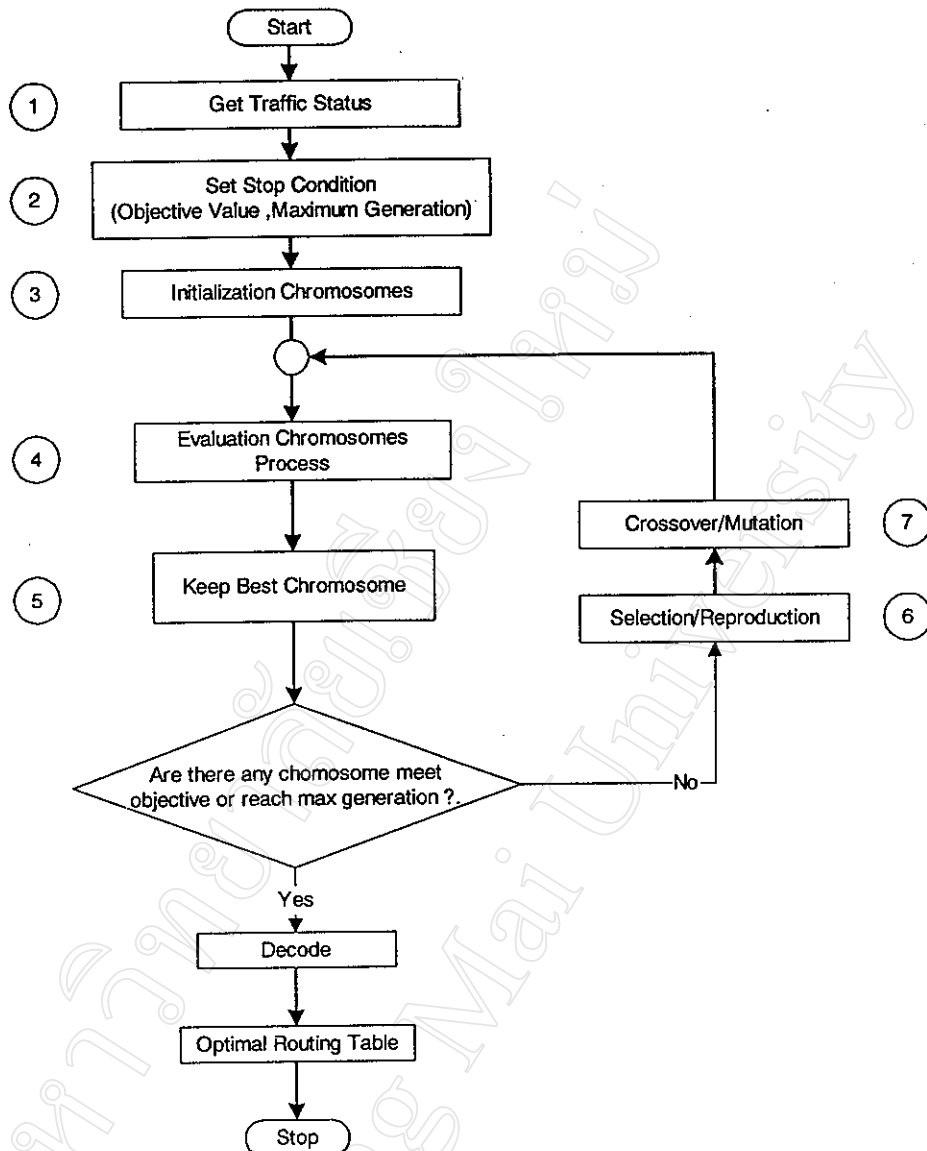
รูปที่ 3.2 การแทนตารางเส้นทางให้อยู่ในรูปโครโมโซม

จากรูปที่ 3.2 ตำแหน่งของโครโมโซมจะแทนหมายเลขเส้นทาง และค่าตัวเลขของแต่ละโครโมโซมจะแทนหมายเลขเส้นทางที่ใช้ ซึ่งมีค่าตั้งแต่หนึ่งถึงแปดแล้วแต่จำนวนวิธีของแต่ละเส้นทาง สำหรับการถอดรหัสโครโมโซมกลับมาเป็นตารางเส้นทางก็สามารถทำกลับกันได้โดยง่ายเช่นกัน

3.2 การจำลองระบบการหาตารางเส้นทางโดยวิธีจีเน็ติกอัลกอริทึมโดยโปรแกรมคอมพิวเตอร์ (MATLAB)

ระบบจำลองที่ใช้ในงานวิจัยนี้ สร้างโดยโปรแกรมคอมพิวเตอร์ MATLAB โดยมีไฟล์ชาร์ตดังรูปที่ 3.3 รายละเอียดการทำงานของแต่ละส่วนเป็นดังนี้

ส่วนที่ 1. การทำงานเริ่มต้นจากการสร้างข้อมูลกราฟฟิกขนาดหกคูณหกให้มีค่าระหว่าง 0 ถึง 40 Mbps ซึ่งสมมุติว่าเป็นกราฟฟิกที่เกิดขึ้นบนเครือข่ายขณะนั้น รายละเอียดโปรแกรมแสดงดังรูป 3.4



รูปที่ 3.3 โฟลว์ชาร์ตการทำงานของระบบจำลองระบบการหาเส้นทางโดยใช้จีนีติกอัลกอริทึม

```

TRFX=round(rand(6,6)*40);
for x = 1:6
    TRFX(x,x)=0;
end
end
  
```

รูปที่ 3.4 โปรแกรมการสุ่มค่ากราฟฟิกขนาดหกคูณหก

จากรูปที่ 3.4 โปรแกรมสุ่มกราฟฟิค ใช้คำสั่ง Rand ในการสุ่มค่าเมตริกซ์ขนาดหกคูณหก แล้วคูณด้วยค่าน้ำหนัก ซึ่งในกรณีนี้คือ 40 จากนั้นทำการตัดกราฟฟิคที่มีเส้นทางและปลายทางเดียวกันออก

ส่วนที่ 2. เป็นการตั้งค่าเป้าหมายของตารางเส้นทางที่ต้องการ ซึ่งในงานวิจัยนี้ตั้งค่าเป้าหมายของค่าประวิงเวลารวมไว้ที่ต่ำกว่า 0.9 วินาที และค่าอัตราการบล็อกเซลล์เฉลี่ยไว้ที่ต่ำกว่า 10^{-8} ส่วนเงื่อนไขค่ายุคสูงสุดตั้งไว้ที่ 30 ยุคเพื่อความเร็วในการคำนวณ

ส่วนที่ 3. การสร้างโครโมโซมเริ่มต้น เพื่อเป็นประชากรยุคแรกให้กับ จีนิติกอัลกอริทึม ทำโดยการสุ่มเลขจำนวนเต็ม ความยาว 30 ตำแหน่ง มีค่าหนึ่งถึงหก หรือหนึ่งถึงแปดขึ้นอยู่กับจำนวนเส้นทางที่เป็นไปได้ในแต่ละตำแหน่ง การสร้างโครโมโซมเริ่มต้นแบบสุ่มจะถูกเปรียบเทียบกับประสิทธิภาพกับ โครโมโซมเริ่มต้นที่สร้างโดยวิธีเครือข่ายประสาทและวิธี Past Selection ภายหลังในงานวิจัยนี้ใช้จำนวนประชากร 500 ตัว โปรแกรมการทำงานของการสร้างโครโมโซมเริ่มต้นแบบสุ่ม แสดงดังรูปที่ 3.5

```
pop_size = 500 ;
for x =1:pop_size %generate 500 Chromosome
    for i = 1:30 %30 length for each Chromosome
        a = 6;
        if i == 4 | i ==16 % 1-4,4-1 have 8 route
            a = 8;
        elseif i ==8 | i==17 %2-4,4-2 have 7 route
            a =7;
        end
        temp = Randperm(a);
        cc(1,i,x)=temp(1,1);

    end %end i loop
end %end x loop
```

รูปที่ 3.5 โปรแกรมการสร้างประชากรเริ่มต้นแบบสุ่ม

ส่วนที่ 4. การประเมินค่าความเหมาะสมของโครโมโซมใช้ ค่าการประวิงเวลาเฉลี่ย (APD:Average Packet Delay) และ ค่าอัตราการบล็อกเซลล์เฉลี่ย (ACB:Average Cell Blocking) ซึ่งมีค่าตามสมการที่ 2.1 และ 2.3 ตามลำดับ นอกจากนี้ยังใช้ค่าฮอป (Hop) มาประกอบเป็นค่าความเหมาะสม โดยกำหนดให้เวลาการคำนวณที่แต่ละโหนดประมาณ 0.001 วินาที แล้วนำไปรวมกับค่าการประวิงเวลารวม เนื่องจากปกติค่าอัตราการบล็อกเซลล์มีค่าน้อยกว่าค่าการประวิงเวลามาก

ดังนั้นจึงต้องมีการควบน้ำหนักให้แก่อัตราการบล็อกเซลล์ด้วย ดังนั้นสมการที่ใช้ ในการคำนวณค่าความเหมาะสมมีค่าเป็น

$$OBJ = \frac{1}{APD + (0.001 * H) + (1000000 * ACB)} \quad (3.1)$$

เมื่อ

APD เป็น ค่าการประวิงเวลา มีค่าตามสมการ 2.1

H เป็น จำนวน ฮอป รวมของตารางเส้นทางนั้น

ACB เป็นค่าการบล็อกเซลล์ของสายส่งทั้งหมด

ส่วนที่ 5. นำโครโมโซมมาเปรียบเทียบค่าเป้าหมายแล้วเก็บโครโมโซมที่มีค่าเป้าหมายดีที่สุดไว้ โดยโครโมโซมที่ดีที่สุดจะถูกคัดลอกเข้าสู่ยุคถัดไป เพื่อให้แน่ใจว่าโครโมโซมที่ดีที่สุดจะไม่ถูกทำลายโดยการครอสโอเวอร์หรือมิวเตชัน

ส่วนที่ 6 การคัดเลือกพ่อแม่พันธุ์ ใช้กลไกการคัดเลือกพันธุ์แบบวงล้อรูเล็ตต์(Roulette Wheel) โครโมโซมตัวใดมีความเหมาะสมสูงก็มีอัตราที่จะถูกเลือกเข้าไปเป็นประชากรยุคถัดไป หรือถูกเลือกเข้าสู่อ่างผสมพันธุ์ (Mating Pool) สูง นอกจากนี้โครโมโซมใดที่มีการใช้งานถึงกั้นความจุของลิงก์นั้น จะถูกทำโทษ(Penalty) โดยการเพิ่มค่าของการประวิงเวลาไปจำนวนหนึ่ง เพื่อลดโอกาสในการถูกเลือกเข้าไปสู่ยุคถัดไป

ส่วนที่ 7 .ใช้อัตราการครอสโอเวอร์ ที่ 25 % และการมิวเตชันที่ 5 % โดยเลือกพ่อแม่พันธุ์จากอ่างผสมพันธุ์ที่ผ่านการเลือกจากวิธี Roulette Wheel นอกจากนี้เพื่อลดเวลาการคำนวณมีการเก็บแฟล็กของการครอสโอเวอร์และมิวเตชัน โดยโครโมโซมใดไม่มีการครอสโอเวอร์หรือมิวเตชัน จะสามารถลดขั้นตอนการคำนวณค่าความเหมาะสมเนื่องจากการคำนวณแล้วในยุคก่อนหน้า

หลังจากได้โครโมโซมที่ตรงตามเงื่อนไข หรือกระบวนการดำเนินการมาจนถึงยุคสูงสุดที่ตั้งค่าไว้ นำโครโมโซมที่ดีที่สุด ไปถอดออกเป็นตารางเส้นทางและนำไปใช้งานต่อไป

3.3 การลดเวลาการคำนวณของจีเนติกอัลกอริทึม

การลดเวลาการคำนวณของจีเนติกอัลกอริทึมโดยการสร้างโครโมโซมเริ่มต้นที่เหมาะสมจะช่วยลดจำนวนยุคที่ใช้ในการเข้าสู่ค่าเป้าหมาย งานวิจัยนี้ศึกษาการสร้างโครโมโซมเริ่มต้นโดยใช้วิธีเครือข่ายประสาท และเปรียบเทียบกับวิธีการสร้างโครโมโซมเริ่มต้นโดยวิธี Past Selection

ที่กล่าวในเอกสารอ้างอิง [6] - [8] การเตรียมข้อมูลสำหรับทั้งสองวิธีทำโดยการสุ่มทราฟฟิกจำนวน 200 รูปแบบ แล้วส่งเข้าสู่กระบวนการจินตคณิตอัลกอริทึม ตั้งค่ายุคสูงสุดที่ 300 ยุค แล้วเก็บค่าทราฟฟิกและโครโมโซมที่ดีที่สุดไว้เป็นข้อมูลสำหรับวิธีเครือข่ายประสาทและวิธี Past Selection ต่อไป

สำหรับรูปแบบทราฟฟิกที่เก็บไว้ ก่อนที่จะนำไปใช้งานต้องนำมาปรับอัตราส่วนเพื่อลดความซับซ้อนของข้อมูล โดยแบ่งค่าทราฟฟิกออกเป็นสี่ระดับ โดยโปรแกรมปรับระดับทราฟฟิก ภาษา MATLAB เป็นดังรูป 3.6

```
%convert Traffic matrix to 0 1 2 3
TRFS = zeros(6,6);
TRFF = TRF ;
for k = 1:6
    for j = 1:6
        if TRFF(k,j)<7
            x = 0;
        elseif 7 >=TRFF(k,j) & TRFF(k,j)<=14
            x = 1;
        elseif 15>=TRFF(k,j) & TRFF(k,j)<=23
            x = 2;
        elseif 24>=TRFF(k,j)
            x = 3;
        end % end if
        TRFS(k,j) = x;
    end %j loop
end %k loop
```

รูปที่ 3.6 โปรแกรมปรับระดับทราฟฟิก

การปรับทราฟฟิกเป็นสี่ระดับ แสดงดังตารางที่ 3.7

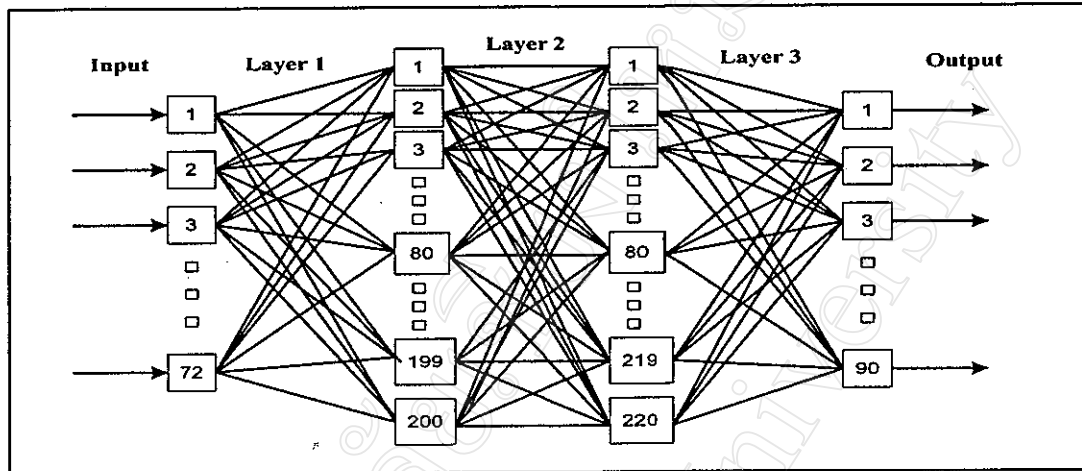
ตารางที่ 3.7 การปรับระดับทราฟฟิก

ขนาดทราฟฟิก	ทราฟฟิกหลังจากแปลง
ทราฟฟิก < 7	0
7 < ทราฟฟิก < 14	1
15 < ทราฟฟิก < 23	2
24 < ทราฟฟิก	3

การปรับระดับของทราฟฟิกให้เป็นระดับ 0 - 3 ทำให้ง่ายต่อการเปรียบเทียบทราฟฟิก และช่วยลดความซับซ้อนในการในการสร้างโครโมโซมโดยเครือข่ายประสาทและวิธี Past Selection

3.3.1 การสร้างโครงข่ายประสาทเทียมเริ่มต้นโดยวิธีเครือข่ายประสาท

เครือข่ายประสาทที่ใช้ในการสร้างค่าเริ่มต้นเป็นแบบ Perceptron แบบมีสองชั้นซ่อน และใช้การเรียนรู้แบบแพร่กลับ (Backpropagation) ในการสอนเครือข่ายประสาท โดยโครงสร้างของเครือข่ายประสาทที่ใช้แสดงดังรูป 3.7



รูป 3.7 โครงสร้างของเครือข่ายประสาทที่ใช้ในการสร้างโครงข่ายประสาทเทียมเริ่มต้น

โดยปกติเครือข่ายประสาทจะทำงานได้ดีกับข้อมูลแบบไบนารี ดังนั้นก่อนที่จะนำข้อมูลกราฟิกเข้ามาสอนให้กับเครือข่ายประสาท ต้องนำข้อมูลกราฟิกมาแปลงให้เป็นข้อมูลแบบไบนารี เนื่องจาก กราฟฟิกมีค่า - 3 ดังนั้นสามารถใช้เลขฐานสองสองบิตแทนได้ ดังนั้นกราฟฟิกขนาด 800x800 สามารถแทนได้โดยเลขฐานสองขนาด 72 บิต ดังนั้น Input Layer มีขนาดเป็น 72 โหนด ส่วน Output Layer มี 90 โหนด เนื่องจากค่าของเส้นทางมีขนาดหนึ่งถึงแปด นั่นคือ มีเจ็ดค่า สามารถใช้เลขฐานสองจำนวนสามบิตแทนได้ ดังนั้นโครงข่ายความยาวสามสิบตำแหน่งสามารถแทนได้โดยเลขฐานสองจำนวน 90 บิต สำหรับชั้นซ่อน จากการทดลองพบว่าการใช้ชั้นซ่อนขนาด 200 และ 220 ตามลำดับ จะทำให้เครือข่ายประสาททำงานได้ดีที่สุด การสร้างเครือข่ายประสาทตามรูปที่ 3.7 โดยใช้ โปรแกรม MATLAB แสดงดังรูปที่ 3.8

```
net = newff([0 1],[200 220 90],{'purelin'
'logsig','logsig'});
net.inputs{1}.size = 72;
net.trainFcn = 'trainrp';
net.trainParam.lr = 0.1;
net.trainParam.mc = 0.7;
net.trainParam.show = 10;
net.trainParam.epochs = 400;
net.trainParam.goal = 1e-3;
```

รูปที่ 3.8 โปรแกรมสร้างเครือข่ายประสาทด้วยภาษา MATLAB

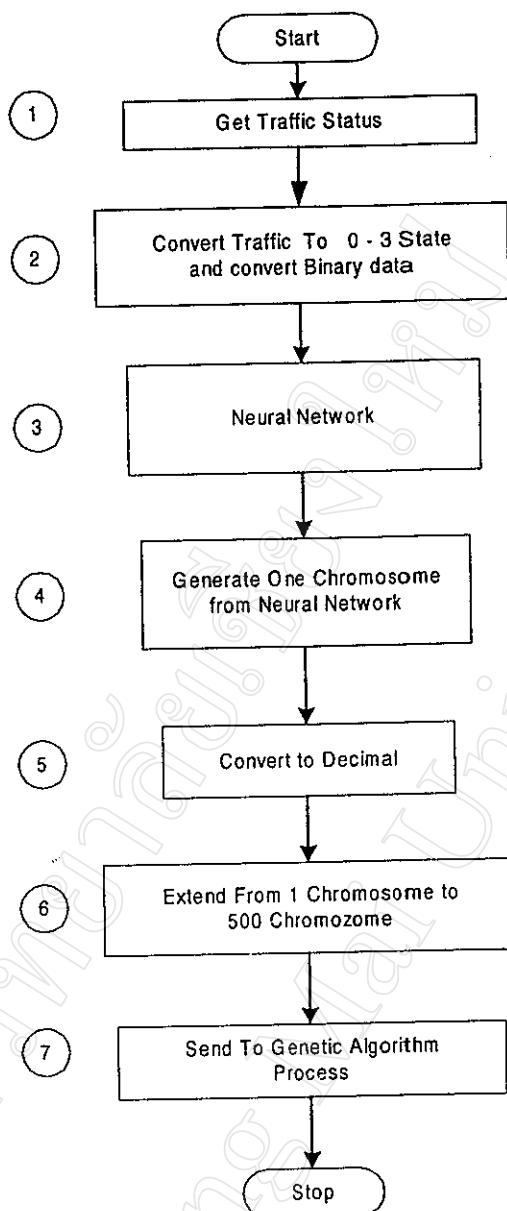
สำหรับค่าพารามิเตอร์อื่นๆ ที่สำคัญของเครือข่ายประสาทที่ใช้ในงานวิจัยนี้ แสดงในตารางที่ 3.8

ตารางที่ 3.8 ค่าพารามิเตอร์ของเครือข่ายประสาท

Neural Parameter	Value
Input Layer Node	70
Output Layer Node	90
Hidden(1) Layer Node	200
Hidden(2) Layer Node	220
Layer 1 Transfer Function	Linear Transfer Function
Layer 2 Transfer Function	Log-Sigmoid Transfer Function
Layer 3 Transfer Function	Log-Sigmoid Transfer Function
Training Algorithm	Resilient Backpropagation
Learning Rate	0.1
Momentum	0.7

ก. การเตรียมข้อมูลสำหรับสอนให้กับเครือข่ายประสาท

หลังจากได้ข้อมูลกราฟฟิคและตารางเส้นทางที่เหมาะสมในรูปโครโมโซมจากกระบวนการจินตนาการ อัลกอริทึม เพื่อลดความผิดพลาดในการสร้างโครโมโซมเอาท์พุท จากคุณสมบัติเจเนอราไลเซชัน (Generalization) ของเครือข่ายประสาท โดยนำข้อมูลกราฟฟิคมาเปลี่ยนแปลงบางส่วน แล้วให้คู่กับโครโมโซมตัวเดิม ซึ่งจะช่วยให้ข้อมูลที่ใช้ในการสอนให้มากขึ้น และช่วยให้เครือข่ายประสาทสร้างโครโมโซมที่เหมาะสมได้ดี โดยงานวิจัยนี้ใช้การขยายไป 10 เท่า ดังนั้นจำนวนชุดข้อมูลที่ใช้ในการสอนเครือข่ายประสาททั้งหมดมี 2000 ชุด แปลงให้เป็นข้อมูลไบนารี แล้วสอนเครือข่ายประสาทต่อไป หลังจากทำการสอนเครือข่ายประสาทเสร็จสิ้น เครือข่ายประสาทก็พร้อมที่จะสร้างโครโมโซมเริ่มต้น โฟวชาร์ตของการสร้างโครโมโซมเริ่มต้นโดยเครือข่ายประสาทแสดงดังรูปที่ 3.9



รูปที่ 3.9 โฟลว์ชาร์ตการสร้างโครโมโซมเริ่มต้นโดยวิธีเครือข่ายประสาท

ส่วนที่ 1. รับค่ากราฟฟิคที่เกิดขึ้นบนเครือข่าย โดยกราฟฟิคนี้เป็นกราฟฟิคเดียวกับที่จะส่งเข้าสู่กระบวนการจินตคณิตอัลกอริทึม

ส่วนที่ 2. เปลี่ยนอัตราส่วนกราฟฟิคให้อยู่ในช่วง 0 - 3 แล้วเปลี่ยนเป็นข้อมูลไบนารี

ส่วนที่ 3. เครือข่ายประสาทที่ใช้ในการสร้างโครโมโซม

ส่วนที่ 4. เครือข่ายประสาทสร้างโครโมโซมออกมา 1 โครโมโซมในรูปข้อมูลไบนารี

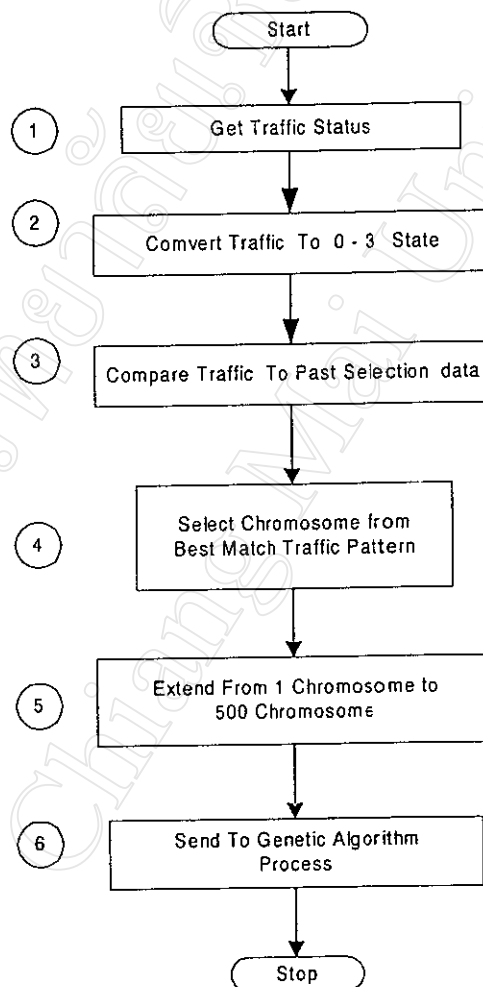
ส่วนที่ 5. แปลงข้อมูลโครโมโซมไบนารีเป็น โครโมโซมเลขฐานสิบ

ส่วนที่ 6. นำโครโมโซมที่ได้ไปขยายโดยสุ่มปรับเปลี่ยนค่าบางตำแหน่งให้มีจำนวน 500 โครโมโซมเท่ากับจำนวนประชากรเริ่มต้น

ส่วนที่ 7. ส่งเข้าสู่กระบวนการจินตคณิตอัลกอริทึมเพื่อหาตารางเส้นทางที่เหมาะสมต่อไป

3.3.2 การสร้างโครโมโซมเริ่มต้นโดยวิธี Past Selection

วิธีการ Past Selection เป็นการใช้ประโยชน์จากข้อมูลกราฟฟิกและโครโมโซมตารางเส้นทางที่เหมาะสมที่เก็บไว้จากกระบวนการจินตคณิตอัลกอริทึมโดยตรง กลไกหลักของวิธี Past Selection คือวิธีการเปรียบเทียบ กราฟฟิกปัจจุบันกับข้อมูลที่เก็บไว้ แล้วเลือกใช้โครโมโซมตารางเส้นทางของกราฟฟิกที่ใกล้เคียงที่สุด มาสร้างประชากรยุคแรก ในงานวิจัยนี้ใช้วิธีการแปลงกราฟฟิกออกเป็น 4 ระดับ ระดับ คือ 0 - 3 ตามตารางที่ 3.7 เพื่อง่ายต่อการเปรียบเทียบ ขั้นตอนการสร้างโครโมโซมเริ่มต้นโดยวิธี Past Selection แสดงดังรูป 3.10



รูป 3.10 โฟลว์ชาร์ตการสร้างโครโมโซมเริ่มต้นโดยวิธี Past Selection

ส่วนที่ 1. รับค่ากราฟฟิกที่เกิดขึ้นบนเครือข่าย โดยกราฟฟิกนี้เป็นกราฟฟิกเดียว
 กับที่จะส่งเข้าสู่กระบวนการจินตคณิตอัลกอริทึม

ส่วนที่ 2. เปลี่ยนอัตราส่วนกราฟฟิคให้อยู่ในช่วง 0-3

ส่วนที่ 3. เปรียบเทียบกราฟฟิคที่รับเข้ามากับกราฟฟิคที่เก็บไว้ โดยโปรแกรม

MATLAB เป็นดังนี้

```
convert % convert traffic to our traffic level
load data1.mat
example = 200 ;
score = zeros(1,example);
Best_score = 0 ; % initialize best score
position = 1; % initial best position
for k=1:6
    for j=1:6
        for i=1:example
            data = TRFS(k,j); %get value from Traffic State
            % TRFB = past selection example that we use
            if data == TRFB(k,j,i)
                score(1,i)=score(1,i)+ 4;
            else
                score(1,i)=score(1,i)+(4-abs(TRFS(k,j)-TRFB
(k,j,i)));
            end % end if
            if score(1,i) > Best_score
                Best_score = score(1,i);
                position = i;
            end % end if score
        end %i loop
    end %j loop
end % k loop
```

รูป 3.11 โปรแกรม MATLAB สำหรับเปรียบเทียบกราฟฟิค

จากรูปที่ 3.11 ใช้วิธีการเปรียบเทียบ กราฟฟิคทีละจุด (ทั้งหมด 36 จุด) โดยมีการให้คะแนนดังนี้

- ให้คะแนน 4 คะแนน ในกรณีที่จุดนั้นมีค่ากราฟฟิคเท่ากัน
- ให้คะแนน 3 คะแนน ในกรณีที่จุดนั้นมีค่ากราฟฟิคต่างกัน 1 ระดับ
- ให้คะแนน 2 คะแนน ในกรณีที่จุดนั้นมีค่ากราฟฟิคต่างกัน 2 ระดับ
- ให้คะแนน 1 คะแนน ในกรณีที่จุดนั้นมีค่ากราฟฟิคต่างกัน 3 ระดับ

จากนั้นนำคะแนนทั้ง 36 จุดมารวมกัน หลังจากเสร็จสิ้นการเปรียบเทียบรูปแบบกราฟฟิคมีคะแนนมากที่สุดก็คือกราฟฟิคที่ใกล้เคียงมากที่สุดและใช้โครโมโซมตารางเส้นทางที่คู่กับกราฟฟิคนั้น มาใช้ในการสร้างประชากรยุคแรกต่อไป

ส่วนที่ 4. เครือข่ายประสาทสร้างโครโมโซมออกมาหนึ่งโครโมโซม

ส่วนที่ 5. นำโครโมโซมที่ได้ไปขยายโดยสุ่มปรับเปลี่ยนค่าบางตำแหน่งให้มีจำนวน 500 โครโมโซม

ส่วนที่ 6. ส่งเข้าสู่กระบวนการจูนิตัลกอริทึมเพื่อหาตารางเส้นทางที่เหมาะสมต่อไป