

บทที่ 3

การออกแบบขั้นตอนวิธีเชิงพันธุกรรมในการแก้ปัญหาการจัดสรรความถี่ของ ช่องสัญญาณเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็ม

ในบทนี้จะกล่าวถึงหลักการและสาระสำคัญของการออกแบบขั้นตอนวิธีเชิงพันธุกรรมในการแก้ปัญหาการจัดสรรความถี่ของช่องสัญญาณเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็ม

3.1 เหตุผลในการเลือกขั้นตอนวิธีเชิงพันธุกรรมมาใช้ในการแก้ปัญหาการจัดสรรความถี่

การเลือกขั้นตอนวิธีเชิงพันธุกรรมมาใช้ในการแก้ปัญหาการจัดสรรความถี่ เนื่องจากขั้นตอนวิธีเชิงพันธุกรรม สามารถคำนวณหาค่าคำตอบได้อย่างรวดเร็ว สอดคล้องกับการแก้ปัญหาการจัดสรรความถี่ที่ต้องการความรวดเร็วเพื่อให้สามารถแก้ไขปัญหาให้กับผู้ใช้งานโทรศัพท์เคลื่อนที่ได้ทันที ไม่กระทบกับความเชื่อมั่นในการใช้งานโทรศัพท์เคลื่อนที่

3.2 การกำหนดเงื่อนไขในการจัดสรรความถี่

การจัดสรรความถี่ของช่องสัญญาณเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็มเป็นการจัดสรรความถี่ที่มีอยู่อย่างจำกัดให้กับแต่ละเซลล์ไซต์ภายในเครือข่ายเพื่อให้ค่าสัญญาณรบกวนในระบบน้อยที่สุด โดยเงื่อนไขหรือข้อจำกัดของสัญญาณรบกวน แบ่งออกเป็น 3 ประเภทดังนี้

1) เงื่อนไขของช่องสัญญาณที่เหมือนกัน (Co-channel constraint : CCC)

ไม่สามารถกำหนดช่องสัญญาณเดียวกันให้กับคู่ของเซลล์ไซต์ที่กำหนดว่าห้ามจัดสรรช่องสัญญาณที่เหมือนกันได้

2) เงื่อนไขของช่องสัญญาณข้างเคียง (Adjacent channel constraint : ACC)

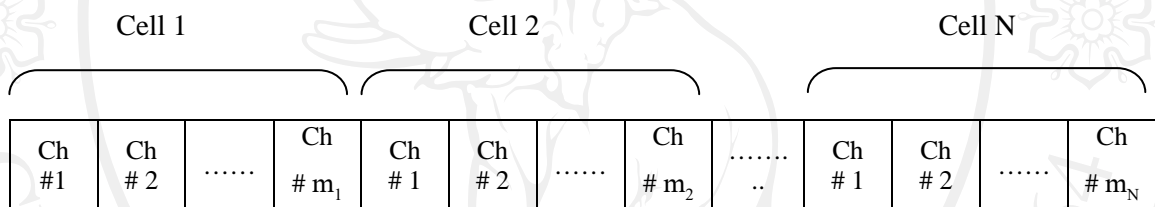
ไม่สามารถกำหนดช่องสัญญาณข้างเคียงกันในแถบความถี่ให้กับเซลล์ไซต์ที่อยู่ข้างเคียงกัน

3) เงื่อนไขของเซลล์ไซต์เดียวกัน (Co-site constraint : CSC)

ในการกำหนดช่องสัญญาณให้กับเซลล์ไซต์เดียวกันจะต้องมีระยะห่างระหว่างความถี่ตามที่กำหนดไว้

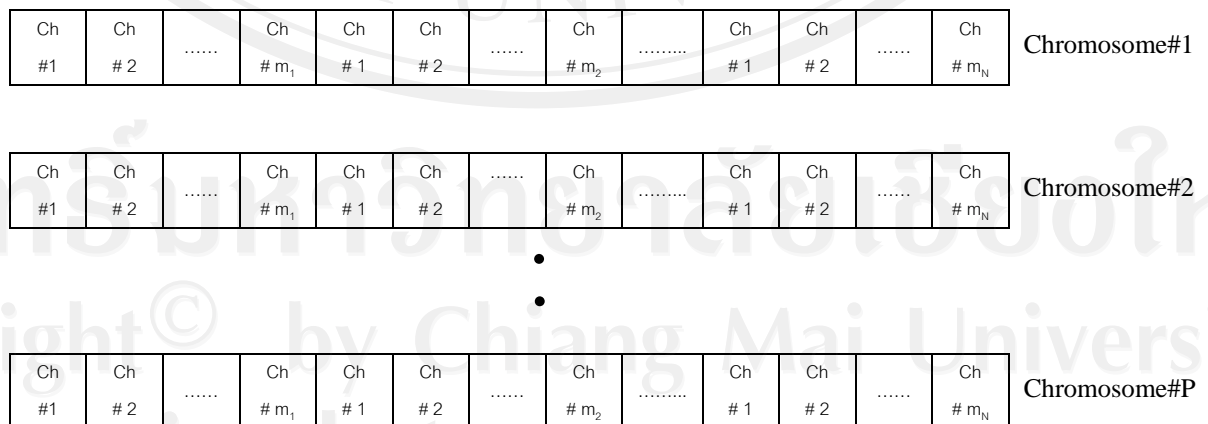
3.3 การออกแบบโครโมโซมเพื่อใช้ในการจัดสรรความถี่

การกำหนดรูปแบบของโครโมโซมเพื่อใช้ในการแก้ปัญหาในการจัดสรรความถี่ของช่องสัญญาณเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็ม กำหนดให้ หนึ่งยีนในโครโมโซมเป็นหนึ่งช่องสัญญาณความถี่ ของหนึ่งเซลล์ไซต์ โดยความยาวของโครโมโซมคือจำนวนช่องสัญญาณรวมทั้งหมดของทุกเซลล์ไซต์ในเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็มที่เราต้องการจัดสรรความถี่



รูป 3.1 รูปแบบโครโมโซมในการจัดสรรความถี่

เมื่อทำการรวมเอาหลายๆ โครโมโซม จะเป็นกลุ่มของโครโมโซมที่เรียกว่าประชากรของโครโมโซม



รูป 3.2 รูปแบบโครงสร้างของประชากรโครโมโซม

ขั้นตอนวิธีเชิงพันธุกรรมเป็นกระบวนการทำซ้ำโดยจะทำการเลือกประชากร $P(t)$ ในแต่ละรอบการทำซ้ำ (t) จะมีประชากรใหม่ $P(t+1)$ ที่ถูกสร้างมาจากประชากรเก่า $P(t)$ โดยตัวปฏิบัติการของขั้นตอนวิธีเชิงพันธุกรรม ได้แก่ การไขว้เปลี่ยน การกลายพันธุ์ และการคัดเลือก

ประชากรประกอบไปด้วยโครโมโซมจำนวน P โครโมโซมเรียกว่าขนาดของประชากร (Population Size) และในแต่ละโครโมโซมจะมีจำนวน Q ช่องสัญญาณทั้งหมดที่ต้องการจัดสรรในเครือข่าย อธิบายได้ดังสมการที่ 3.1

$$Q = \sum_{i=1}^N m_i \quad (3.1)$$

โครโมโซม i จะประกอบไปด้วยจำนวน N เซลไซต์ที่ต้องการจัดสรรช่องสัญญาณในเครือข่าย และในแต่ละเซลล์ไซต์ i จะต้องการจำนวนช่องสัญญาณทั้งหมด m_i

3.4 การกำหนดฟังก์ชันความเหมาะสม

ในเครือข่ายโทรศัพท์เคลื่อนที่ระบบจีเอสเอ็มประกอบไปด้วยจำนวน N เซลไซต์ และจำนวน M ช่องสัญญาณที่สามารถใช้งานได้ โดยจำนวนช่องสัญญาณที่ต้องการของเซลล์ไซต์ i กำหนดให้เป็น m_i โดยที่ $1 \leq i \leq N$ การกำหนดความสัมพันธ์ของระยะห่างของช่องสัญญาณของแต่ละเซลล์ไซต์ เพื่อการันตีระดับสัญญาณต่อสัญญาณรบกวน (signal to noise ratio) ภายในพื้นที่ที่ต้องการมีการจัดสรรช่องสัญญาณจะแทนด้วย $N \times N$ เมตริกซ์ เรียกว่า คอมแพทIBILITY เมตริกซ์ Compatibility matrix C เป็นเมตริกซ์ที่กำหนดระยะห่างของช่องสัญญาณแต่ละเซลล์ไซต์ในเมตริกซ์ C ประกอบด้วย

- แต่ละสมาชิกในแนวเส้นทแยงมุมหลัก C_{ii} แทนด้วยเงื่อนไขของเซลล์ไซต์เดียวกัน โดย C_{ii} เป็นระยะห่างน้อยที่สุดระหว่างสองช่องสัญญาณที่เซลล์ไซต์ i
- แต่ละสมาชิกที่ไม่อยู่ในแนวเส้นทแยงมุม C_{ij} แทนด้วยระยะห่างน้อยที่สุดของช่องสัญญาณที่ถูกจัดสรรให้กับเซลล์ไซต์ i และ j
- เงื่อนไขของช่องสัญญาณที่เหมือนกันแทนด้วย $C_{ij} = 1$
- เงื่อนไขของช่องสัญญาณข้างเคียงแทนด้วย $C_{ij} = 2$

- เซลล์ที่เป็นอิสระต่อกันซึ่งสามารถใช้ช่องสัญญาณเดียวกันได้จะแทนด้วย $C_{ij} = 0$

ตัวอย่างที่ 1 ให้จำนวนของเซลล์ในเครือข่าย $N = 4$ จำนวนช่องสัญญาณที่สามารถใช้งานได้ $M = 15$ และจำนวนช่องสัญญาณที่ต้องการในแต่ละเซลล์กำหนดให้เป็น $D = (2,2,2,3)$ โดยมี Compatibility matrix C เป็น

$$C = \begin{bmatrix} 3 & 2 & 0 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

สมาชิกแนวเส้นทแยงมุม $C_{ii} = 3$ ช่องสัญญาณที่ถูกจัดสรรให้กับ Cell i จะต้องมียุทธศาสตร์ของช่องสัญญาณอย่างน้อย 3 ช่องสัญญาณเพื่อไม่ทำให้เกิดสัญญาณรบกวนในเซลล์ที่เดียวกันตรงกับเงื่อนไขของเซลล์ที่เดียวกัน และสมาชิกที่ไม่อยู่ในแนวเส้นทแยงมุม $C_{ij} = 1$ และ $C_{ij} = 2$ จะตรงกับเงื่อนไขของช่องสัญญาณที่เหมือนกัน และเงื่อนไขของช่องสัญญาณข้างเคียงตามลำดับ โดยการจัดสรรช่องสัญญาณตามเงื่อนไขที่กำหนดดังรูป 3.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															

รูป 3.3 การจัดสรรความถี่ที่ไม่เกิดสัญญาณรบกวน สำหรับ 4 เซลล์ และ 15 ช่องสัญญาณ

เมตริกซ์ F ขนาด $N \times M$ เป็นไบนารีเมตริกซ์สำหรับการจัดสรรช่องสัญญาณให้กับแต่ละเซลล์ โดยที่ N คือจำนวนของเซลล์ทั้งหมดในเครือข่าย และ M คือจำนวนของช่องสัญญาณที่สามารถใช้งานได้ ในแต่ละสมาชิก u_{ik} ในเมตริกซ์ F มีค่าเป็น 0 หรือ 1 โดยมีเงื่อนไขดังนี้

$$u_{ik} = \begin{cases} 1 & \text{if channel } k \text{ is } \left\{ \begin{array}{l} \text{assigned} \\ \text{not assigned} \end{array} \right\} \text{ to cell } i \\ 0 & \end{cases}$$

u_{ik} แทนด้วยสมาชิกในเมตริกซ์ F
เมตริกซ์ F สามารถแสดงได้ด้วยรูป 3.4

	Channel Number						
	1	2	3	...			M
1	0	1	0	0	0	0	0
2	0	0	0	0	1	0	1
3	0	1	0	0	0	0	0
...	0	0	1	0	0	1	0
N	1	0	1	0	0	1	0

รูป 3.4 เมตริกซ์ F ไบนารีเมตริกซ์ขนาด $N \times M$

เราจะกำหนดฟังก์ชันความเหมาะสมสำหรับแต่ละเงื่อนไขในการจัดสรรความถี่ดังนี้

- 1) ฟังก์ชันความเหมาะสมสำหรับเงื่อนไขของเซลล์ไซต์ที่เดียวกัน F_{CSC}

สำหรับเงื่อนไขของเซลล์ไซต์ที่เดียวกัน F_{CSC_i} ฟังก์ชันความเหมาะสมของแต่ละเซลล์ไซต์ i ได้ดังสมการที่ 3.2

$$F_{CSC_i} = \sum_{k=1}^{m_i} f_{ik} \quad (3.2)$$

$$f_{ik} = \begin{cases} 1 & : \text{if } |v_{ik} - v_{i(k+1)}| < C_{ii} \text{ or } |v_{ik} - v_{i(k-1)}| < C_{ii} \\ 0 & : \text{otherwise} \end{cases}$$

f_{ik} แทนด้วยสถานะของการเป็นไปตามเงื่อนไขของเซลล์ไซต์เดียวกัน

i แทนด้วยหมายเลขของเซลล์ไซต์ในเครือข่าย

k แทนด้วยหมายเลขของช่องสัญญาณในแต่ละเซลล์ไซต์

v_{ik} แทนด้วยความถี่ของช่องสัญญาณที่กำหนด โดยที่เมื่อไม่เป็นไปตามเงื่อนไขของเซลล์ไซต์เดียวกัน $f_{ik} = 1$ แต่ถ้าเป็นไปตามเงื่อนไขของเซลล์ไซต์เดียวกัน $f_{ik} = 0$ ฟังก์ชันความเหมาะสมสำหรับแต่ละโครโมโซม i ที่รวมเซลล์ไซต์ทั้งหมด ได้ดังสมการที่ 3.3

$$F_{CSC} = \sum_{i=1}^N \sum_{k=1}^{m_i} f_{ik} \quad (3.3)$$

N คือจำนวนของเซลล์ไซต์ทั้งหมดในเครือข่าย

m_i คือจำนวนช่องสัญญาณแต่ละเซลล์ไซต์ i

2) ฟังก์ชันความเหมาะสมสำหรับเงื่อนไขของช่องสัญญาณข้างเคียง และช่องสัญญาณเดียวกัน

F_{AC}

ฟังก์ชันความเหมาะสมสำหรับเงื่อนไขของช่องสัญญาณข้างเคียง และช่องสัญญาณเดียวกันแต่ละ โครโมโซม ได้ดังสมการที่ 3.4

$$F_{AC} = \sum_{i=1}^N \sum_{k=1}^{m_i} \sum_{j=1}^N \sum_{l=1}^{m_j} f_{ijkl} \quad (3.4)$$

ที่

$$f_{ijkl} = \begin{cases} 1 & : \text{if } |v_{ik} - v_{jl}| < C_{ij} \\ 0 & : \text{otherwise} \end{cases}$$

f_{ijkl} แทนด้วยสถานะของการเป็นไปตามเงื่อนไขของช่องสัญญาณข้างเคียง และช่องสัญญาณเดียวกัน

i และ j แทนด้วยหมายเลขของเซลล์ในเครือข่าย

k และ l แทนด้วยหมายเลขของช่องสัญญาณในแต่ละเซลล์

สุดท้ายเราจะพิจารณาฟังก์ชันความเหมาะสมโดยรวมเอาฟังก์ชันความเหมาะสมสำหรับเงื่อนไขของเซลล์เดียวกัน และเงื่อนไขของช่องสัญญาณข้างเคียงในแต่ละโครโมโซม ได้ตั้งสมการที่ 3.5

$$\begin{aligned}
 F_{Chromosome} &= F_{CSC_{Chromosome}} + F_{AC_{Chromosome}} \\
 &= \sum_{i=1}^N \sum_{k=1}^{m_i} f_{ik} + \sum_{i=1}^N \sum_{k=1}^{m_i} \sum_{j=1}^N \sum_{l=1}^{m_j} f_{ijkl}
 \end{aligned} \tag{3.5}$$

จากสมการฟังก์ชันความเหมาะสมข้างต้นสามารถนำมาเขียนอัลกอริทึม ได้ดังนี้

Algorithm EvaluateIndividual (chromosome,C,D,M,N)

Input : chromosome : array of frequency allocation

C : Compatibility Matrix size NxN

D : array of require of frequency channel

M : number of available channel

N : number of Cell

Output : Return Fitness value each chromosome

Variables : ECSC : summary of not satisfied CSC constraint

EAC : summary of not satisfied ACC and CCC constraint

diffCh : different of channel

// หมายเลขบรรทัด 1-13 อ้างอิงสมการ 3.3 //

// หมายเลขบรรทัด 14-28 อ้างอิงสมการ 3.4 //

// หมายเลขบรรทัด 29 อ้างอิงสมการ 3.5 //

```

1. ECSC = 0
2. i = 0
3. while i < N
4.   for k = 1 to D(i)-1
5.     for j = k+1 to D(i)
6.       diffCh = chromosome(k) - chromosome(j)
7.       if diffCh < C(i,i) then
8.         ECSC = ECSC+1
9.       end if
10.    end for
11.  end for
12.  i = i + 1
13. end while
14. EAC = 0
15. i = 0
16. while i < N
17.   for k = 1 to D(i)
18.     j = i+1
19.     while j <= N
20.       for l = 1+D(i) to D(j)
21.         diffCh = chromosome(k) - chromosome(j)
22.         if diffCh < C(i,j) then
23.           EAC = EAC + 1
24.         end if

```


25. end for
26. j = j+1
27. end while
28. i = i+1
29. end while
30. fitness value = ECSC + EAC
31. return fitness value

3.5 การดำเนินการของขั้นตอนวิธีเชิงพันธุกรรมในการจัดสรรช่องสัญญาณ

3.5.1 สร้างประชากรเริ่มต้น (Initial population)

สร้างประชากรเริ่มต้น โดยการสุ่มช่องสัญญาณความถี่ที่มีอยู่ในระบบให้กับยีนในแต่ละโครโมโซม โดยจำนวนโครโมโซมที่จะสร้างทั้งหมดจะเท่ากับขนาดของประชากรที่กำหนดไว้

Algorithm InitializePopulation (populationSize,D,M,N)

Input : populationSize : number of population chromosome

D : array of require of frequency channel

M : number of available channel

N: number of Cell

Output : Return population

Variables : sumF : array for summary '1' in F Matrix

pop : index of populationSize

chromosome : array of frequency allocation

1. Create Binary Matrix F array NxM

2. for pop = 1 to populationSize

3. Matrix F = 0

4. sumF = 0

5. for i = 1 to N

```

6.     for j = 1 to M
7.         if sumF != D(i) then
8.             F(i,j) = random '1' or '0'
9.             if F(i,j) == '1' then
10.                sumF(i) = sumF(i)+1
11.            end if
12.        end if
13.    end for
14.    count = 1
15.    for i = 1 to N
16.        for j = 1 to M
17.            if F(i,j) == '1' then
18.                chromosome(pop,count) = j
19.                population(pop,count) = chromosome(pop,count)
20.                count = count+1
21.            end if
22.        end for
23.    end for
24.    return population

```

3.5.2 การคัดเลือก (Selection)

การคัดเลือกจะใช้วิธีการคัดเลือกแบบวงล้อรูเล็ตต์ โดยการคัดเลือกจะทำการเลือกโครโมโซมตามค่าความเหมาะสมของแต่ละโครโมโซม โดยที่โครโมโซมใดมีค่าความเหมาะสมน้อยจะมีโอกาสที่จะถูกคัดเลือกสูง โอกาสที่จะถูกคัดเลือกของแต่ละโครโมโซมหาได้จากสมการดังนี้

$$P_{select\ Chromosome\ i} = \frac{F_{Chromosome\ i}}{\sum_{i=1}^P F_{Chromosome\ i}}$$

โดยที่ P คือจำนวนประชากร

Algorithm SelectionRoulette (fitness, populationSize)

Input : fitness : fitness function array of each chromosome

populationSize : number of population chromosome

Output : Return index of chromosome in population

Variables : wheel : cumulative summary of fitness

1. wheel = cumulative summary / summary all fitness
2. r = random selection within value of wheel
3. for j = 1 to length(wheel)
4. if r < wheel(j) then
5. return value j
6. end if
7. end for

3.5.3 การไขว้เปลี่ยน (Crossover)

จะดำเนินการไขว้เปลี่ยน เมื่อค่าความน่าจะเป็นของการไขว้เปลี่ยน (Crossover probability)

มากกว่าค่าจากการสุ่มการไขว้เปลี่ยนจะทำการสุ่มจุดที่จะทำการไขว้เปลี่ยนของโครโมโซม 2 ตัวที่ถูกคัดเลือก โดยได้ทำการเลือกใช้การไขว้เปลี่ยนแบบหนึ่งส่วน

1) การไขว้เปลี่ยนแบบหนึ่งส่วน (Single-Point Crossover)

นำโครโมโซม 2 ตัวที่ถูกคัดเลือก แล้วสุ่มจุดที่สลับยีนจำนวน 1 จุดจากนั้นทำการสลับยีนระหว่าง 2 โครโมโซม

// Single-Point Crossover

Algorithm CrossOver_one point (chromosome1, chromosome2)

Input : chromosome1 : chromosome is selected by roulette wheel selection

chromosome2 : chromosome is selected by roulette wheel selection

Output : return new chromosome

Variables : crossOverPoint : random point of crossover

nGenes : size of chromosome

1. nGenes = size of chromosome
2. crossOverPoint = random point of crossover
3. for j = 1 to nGenes
4. if j > crossOverPoint then
5. new chromosome1(j) = chromosome1(j)
6. new chromosome2(j) = chromosome2(j)
7. else
8. new chromosome1(j) = chromosome2(j)
9. new chromosome2(j) = chromosome1(j)
10. end if
11. end for
12. return new chromosome1 and new chromosome2

2) การไขว้เปลี่ยนแบบสองส่วน (Two-Point Crossover)

นำโครโมโซม 2 ตัวที่ถูกคัดเลือก แล้วสุ่มจุดที่สลับยีนจำนวน 2 จุดจากนั้นทำการสลับยีนระหว่าง 2 โครโมโซม

// Two-Point Crossover

Algorithm CrossOver_two point (chromosome1, chromosome2)

Input : chromosome1 : chromosome is selected by roulette wheel selection

chromosome2 : chromosome is selected by roulette wheel selection

Output : return new chromosome

Variables : crossOverPoint1,2 : random point of crossover

nGenes : size of chromosome

1. nGenes = size of chromosome
2. crossOverPoint1 = random 1st point of crossover
3. crossOverPoint2 = random 2nd point of crossover
4. if (crossOverPoint1 < crossOverPoint2) then
5. for j = 1 to nGenes
6. if j < crossOverPoint1 then
7. new chromosome1(j) = chromosome2(j)
8. new chromosome2(j) = chromosome1(j)
9. else
10. if j > crossOverPoint1 and j < crossOverPoint2 then
11. new chromosome1(j) = chromosome1(j)
12. new chromosome2(j) = chromosome2(j)
13. else
14. new chromosome1(j) = chromosome2(j)
15. new chromosome2(j) = chromosome1(j)
16. end if
17. end if
18. end for
19. else
20. for j = 1 to nGenes
21. if j < crossOverPoint2 then

```

22.     new chromosome1(j) = chromosome2(j)
23.     new chromosome2(j) = chromosome1(j)
24.   else
25.     if j>crossOverPoint2 and j<crossOverPoint1 then
26.       new chromosome1(j) = chromosome1(j)
27.       new chromosome2(j) = chromosome2(j)
28.     else
29.       new chromosome1(j) = chromosome2(j)
30.       new chromosome2(j) = chromosome1(j)
31.     end if
32.   end if
33. end for
34. end if
35. return new chromosome1 and new chromosome2

```

3) การไขว้เปลี่ยนแบบยูนิฟอร์ม (Uniform Crossover)

นำโครโมโซม 2 ตัวที่ถูกคัดเลือก แล้วสุ่มหน้าต่าง (Mask) จากนั้นทำการสลับยีนระหว่าง 2 โครโมโซม โดยจะสลับยีน ณ จุดที่หน้าต่างเป็นบิต '1'

```
// Uniform Crossover
```

Algorithm CrossOver_uniform (chromosome1,chromosome2)

Input : chromosome1 : chromosome is selected by roulette wheel selection

chromosome2 : chromosome is selected by roulette wheel selection

Output : return new chromosome

Variables : crossUnifMask : array size 1 x nGenes

nGenes : size of chromosome

1. nGenes = size of chromosome

2. crossUnifMask = array size 1 x nGenes
3. for i = 1 to nGenes
4. crossUnifMask(j) = random '1' or '0'
5. for j = 1 to nGenes
6. if crossUnifMask(j) == '0' then
7. new chromosome1(j) = chromosome1(j)
8. new chromosome2(j) = chromosome2(j)
9. else
10. new chromosome1(j) = chromosome2(j)
11. new chromosome2(j) = chromosome1(j)
12. end if
13. end for
14. end for
15. return new chromosome1 and new chromosome2

3.5.4 การกลายพันธุ์ (Mutation)

การกลายพันธุ์จะดำเนินการเมื่อค่าความน่าจะเป็นของการกลายพันธุ์ (Mutation probability) มากกว่าค่าจากการสุ่มการดำเนินการกลายพันธุ์จะทำการเปลี่ยนการจัดสรรช่องสัญญาณในยีนของโครโมโซม ไปเป็นช่องสัญญาณอื่นภายในช่องสัญญาณความถี่ที่มีอยู่ในระบบการทำงานจะนำเอาโครโมโซม มาทำการสุ่มตัวเลขในแต่ละยีนภายในโครโมโซม เช่นถ้าค่าตัวเลขที่สุ่ม ของยีน ณ ตำแหน่งที่ j น้อยกว่าความน่าจะเป็นของการกลายพันธุ์ ก็จะนำเอายีนของโครโมโซม ณ ตำแหน่งที่ j ไปรวมกับค่าจากการสุ่มในช่วงตัวเลขระหว่าง 1 ถึง (จำนวนของช่องสัญญาณที่สามารถใช้ได้ไปลบกับ ยีนของโครโมโซม ณ ตำแหน่งที่ j) แต่ถ้าตัวเลขที่สุ่มมากกว่าความน่าจะเป็นของการกลายพันธุ์ จะนำเอายีนของโครโมโซม ณ ตำแหน่งที่ j ไปลบกับค่าจากการสุ่มในช่วงตัวเลขระหว่าง 1 ถึง ค่ายีนของโครโมโซม ณ ตำแหน่งที่ j)

Algorithm Mutation (original chromosome, mutationProb, M)

Input : originalChromosome : array of original chromosome

mutationProb : probability value of mutation

M : number of available channel

Output : return mutated chromosome

Variables : nGenes : size of original chromosome

1. nGenes = size of original chromosome
2. mutated chromosome = original chromosome
3. for j = 1 to nGenes
4. r = random
5. if r < mutationProb then
6. mutated chromosome(j) = mutated chromosome(j) + random value between 1
to (M – mutated chromosome(j))
7. else
8. mutated chromosome(j) = mutated chromosome(j) - random value between 1
to mutated chromosome(j)
9. end if
10. end for
11. return mutated chromosome