

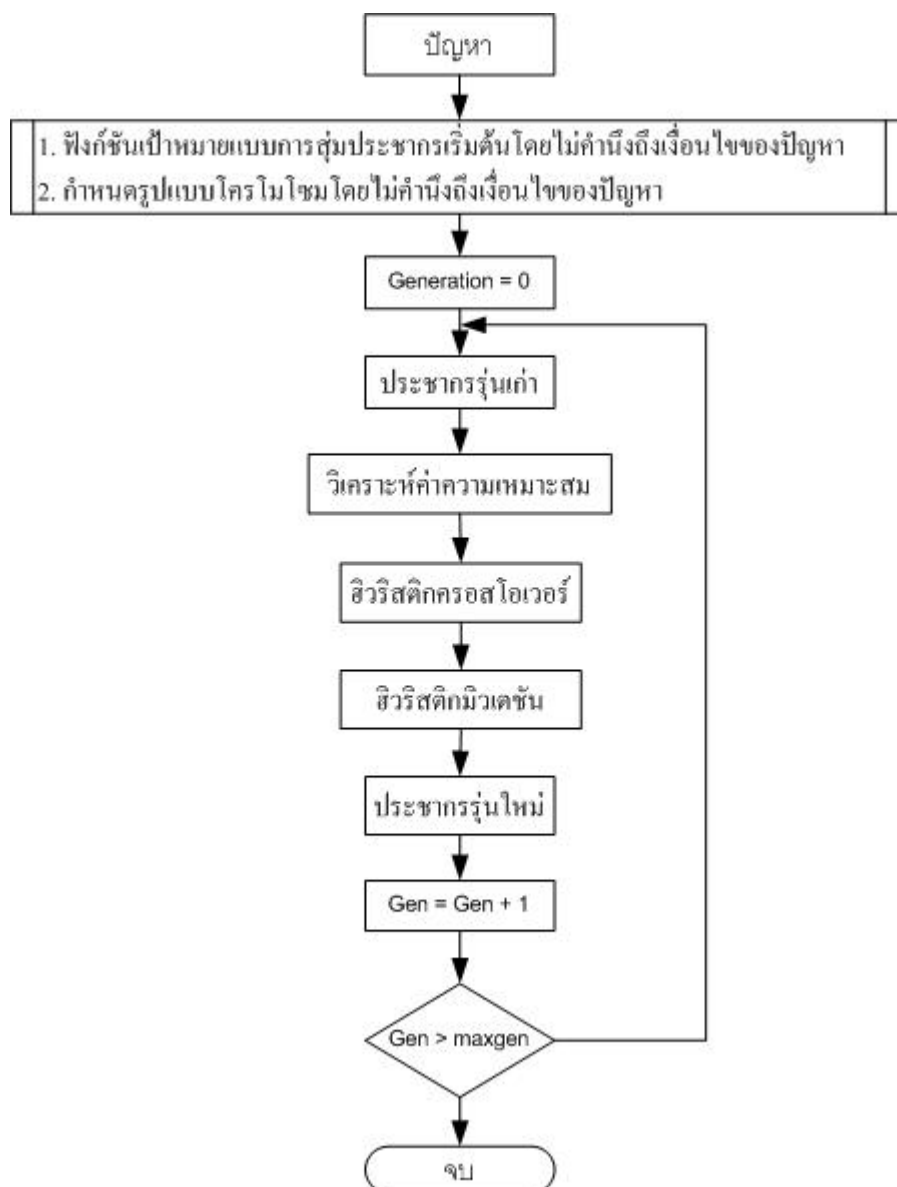
บทที่ 5

การปรับปรุงประสิทธิภาพการทำงานของจีเนติกอัลกอริทึม

เนื่องจากการแก้ปัญหาการเลือกเส้นทางแบบการใช้ทางออกทางเดียวของบีจีพี ด้วยวิธีจีเนติกแบบง่าย ยังไม่สามารถหาค่าต้นทุนที่เหมาะสมได้เมื่อเราเทียบกับอัลกอริทึม MPPF ดังนั้นเราจึงเริ่มการปรับปรุงอัลกอริทึมจีเนติกแบบง่าย ด้วยการนำวิธีฮิวริสติกเพื่อที่จะช่วยให้ประชากรของจีเนติกมีการพัฒนาไปในทางที่ดีกว่าเดิมเสมอ ในงานวิจัยเราได้ปรับปรุงการทำงานของจีเนติกแบบง่าย โดยการปรับปรุงการครอสโอเวอร์และมิวเตชัน ซึ่งผลจากการปรับปรุงทำให้ได้ผลลัพธ์ที่ดีกว่าเดิม และงานวิจัยเรื่อง “A Combined Heuristic and Genetic Approach to the BGP Route Configuration Problem” [18] ได้ตีพิมพ์ในวารสาร ITC-CSCC

5.1 การใช้ฮิวริสติกกับจีเนติกในการแก้ปัญหาการเลือกเส้นทางแบบใช้ทางออกทางเดียว

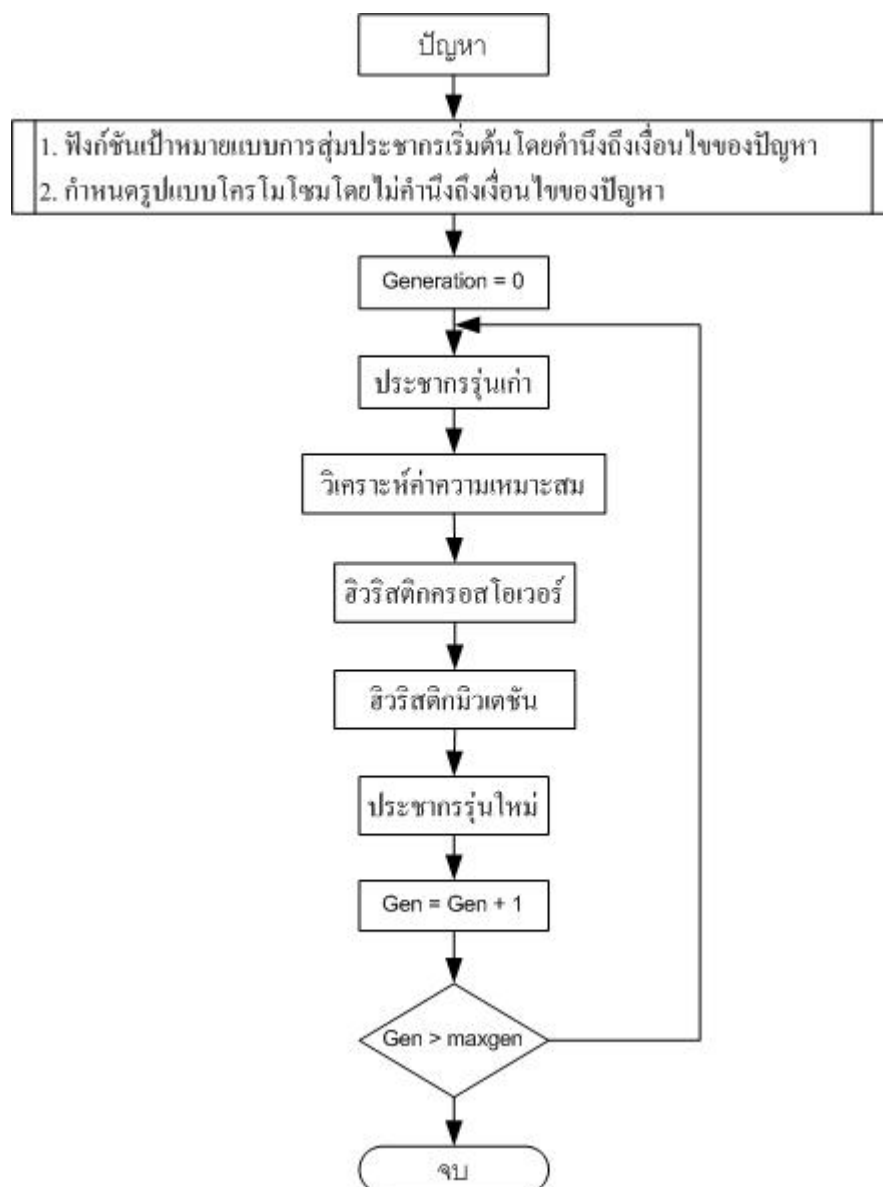
การนำฮิวริสติกมาปรับปรุงการทำงานของ SGA โดยการปรับปรุงการทำงานภายในอัลกอริทึมครอสโอเวอร์และมิวเตชัน ซึ่งการทำงานของวิธี SGA เป็นการทำงานแบบสุ่มทั้งวิธีครอสโอเวอร์และวิธีมิวเตชัน ซึ่งผลที่ได้จากสองวิธีนี้ผลที่ได้อาจไม่มีความแน่นอนว่าโครโมโซมที่ได้จากการทำงานจะมีวิวัฒนาการไปในทางใด อาจจะได้ผลที่ดีหรือไม่ดีก็ได้ขึ้นอยู่กับวิธีการใช้วิธีฮิวริสติกเข้ามาช่วยปรับปรุงการทำงานให้ส่วนนี้ โดยหลักการทำงานที่สำคัญคือโครโมโซมลูกที่ได้จากการครอสโอเวอร์และมิวเตชันจะต้องได้ผลที่ดีกว่าโครโมโซมรุ่นพ่อแม่ การปรับปรุงโดยการนำฮิวริสติกเข้ามาใช้กับจีเนติก ได้มีการแบ่งการทำงานออกเป็นสองแบบตามลักษณะของฟังก์ชันวัตถุประสงค์ โดยแบบแรกเป็นฟังก์ชันเป้าหมายแบบการสุ่มประชากรเริ่มต้นโดยไม่คำนึงถึงเงื่อนไขของปัญหา ซึ่งมีชื่อเรียกว่า HGA (L) หรือ HGA แบบ Loose โดยมีการปรับปรุงการทำงานของ SGA (L) ดังนี้



รูปที่ 5.1 แสดงรูปอัลกอริทึม HGA (L)

อัลกอริทึมที่ได้ทำการปรับปรุงจินเนติก โดยการใช้อีวิริสติกแบบฟังก์ชันเป้าหมายแบบการสุ่มประชากรเริ่มต้นโดยไม่คำนึงถึงเงื่อนไขของปัญหา HGA (L) ที่แสดงอยู่ในรูปที่ 5.1 โดยภาพรวมโครงสร้างการทำงานของอัลกอริทึมยังคงเดิม รวมถึง ฟังก์ชันวัตถุประสงค์และฟังก์ชันความเหมาะสม ส่วนที่เปลี่ยนแปลงไป จะเป็นส่วนของการพัฒนาอัลกอริทึมขึ้นมาใหม่นั้นได้ทำการเปลี่ยนแปลงเฉพาะในส่วนของการครอสโอเวอร์และมิวเตชันเท่านั้น

สำหรับแบบที่สองนั้นเป็นการปรับปรุงจาก SGA (S) ซึ่งใช้ฟังก์ชันเป้าหมายแบบการสุ่มประชากรเริ่มต้นโดยคำนึงถึงเงื่อนไขของปัญหา มีชื่อแบบย่อว่า HGA (S) หรือ HGA แบบ Strict โดยมีการปรับปรุงการทำงานของ SGA (S) ดังนี้



รูปที่ 5.2 แสดงรูปอัลกอริทึม HGA (S)

จากรูปที่ 5.2 โครงสร้างหลักของอัลกอริทึมยังใช้วิธีเดียวกับ SGA (S) เริ่มต้นตั้งแต่การสุ่มสร้างโครโมโซมเริ่มต้น การคัดเลือกประชากร การรักษาสภาพประชากรรุ่นเก่า ด้านอัลกอริทึมใหม่ที่ได้พัฒนาขึ้นมานั้นได้ทำการเปลี่ยนแปลงเฉพาะในส่วนของการครอสโอเวอร์และมิวเตชันเท่านั้น โดยที่ทั้งสองอัลกอริทึม HGA (L) และ HGA (S) มีอัลกอริทึมในการครอสโอเวอร์และมิวเตชันเหมือนกัน ซึ่งได้ทำการปรับปรุงดังต่อไปนี้

5.1.1 การปรับปรุงการครอสโอเวอร์แบบฮิวริสติกโดยใช้วิธีฮิวริสติก

การปรับปรุงการครอสโอเวอร์ของ SGA โดยการนำวิธีฮิวริสติกเข้ามาช่วยนั้น ซึ่งจุดประสงค์หลักคือ การต้องให้โครโมโซมลูกมีการพัฒนาไปในทางที่ดีขึ้นกว่าเดิม ในทางปฏิบัติแล้ว การทำงานของ SGA จะไม่คำนึงถึงผลการของการมีวิวัฒนาการว่าโครโมโซมจะมีวิวัฒนาการที่ดีขึ้นหรือไปจากจุดนี้ เราได้ทำการปรับปรุงการทำงานวิธีครอสโอเวอร์แบบสุ่ม โดยมีลำดับการทำงานแสดงอยู่ในรูปที่ 5.3 ดังนี้

Heuristic Crossover Algorithm

1: HCrossover($CMS1$, $CMS2$)

2: Check the lower cost solution between two chromosomes ($CMS1$ and $CMS2$). Then the lower cost chromosome is called CMS_L , otherwise is CMS_H .

3: for prefix number $k=1$ to K do {

4: $CMS_COST(k) = \text{Infinity}$

5: $CMS_{L,k} = \text{Result of replacing of column } k \text{ on } CMS_L \text{ by column } k \text{ of } CMS_H$

6: If capacity constraint of $CMS_{L,k}$ is satisfied by Eq.(4.2) then {

7: calculate cost of $CMS_{L,k}$ chromosome and assign it to $CMS_COST[k]$

8: }

9: }

10: The $CMS_{L,k}$ chromosome which has lowest cost of $CMS_COST[k]$ is then selected as a child chromosome for next generation

11: end

รูปที่ 5.3 แสดงขั้นตอนการทำงานของฟังก์ชันครอสโอเวอร์แบบวิธีฮิวริสติก

การทำงานของฟังก์ชันครอสโอเวอร์แบบฮิวริสติก เริ่มต้นจากการนำโครโมโซมพ่อและแม่ ที่ได้จากการจับคู่จากการสุ่มมาเข้ากระบวนการครอสโอเวอร์ จากรูปจะเห็นได้จากโครโมโซม $CMS1$ และ $CMS2$ เป็นตัวแปรของโครโมโซมพ่อและแม่จะทำการครอสโอเวอร์ โดยเริ่มต้นที่

บรรทัดที่ 2 ทำการเปรียบเทียบค่าต้นทุนของโครโมโซมพ่อกับแม่ โครโมโซมตัวใดมีค่าต้นทุนต่ำกว่าให้เรียกเป็นโครโมโซม CMS_L ส่วนตัวที่เหลือให้เป็น CMS_H

บรรทัดที่ 3 เป็นการเริ่มต้นของลูปของตัวแปร k โดยลูปนี้จะทำงาน K ครั้ง หรือตามจำนวนหมายเลขเครือข่าย

บรรทัดที่ 4 กำหนดค่าเริ่มต้นให้กับตัวแปร $CMS_COST[k]$ ของแต่ละ k โดยให้ค่าเป็นอนันต์

บรรทัดที่ 5 ทำการคัดลอกค่าของแถวที่ k จากโครโมโซม CMS_H แทนที่แถวที่ k ของโครโมโซม CMS_L

บรรทัดที่ 6 ทำการตรวจสอบค่าต้นทุนของโครโมโซมใหม่ที่ได้ ว่ามีแบบวัดที่เกินความจุของท่อส่งข้อมูลหรือไม่ ในสมการที่ 4.2 หรือ 4.5 ถ้าไม่เกินให้นำค่าต้นทุนที่ได้มาเก็บไว้ในตัวแปร $CMS_COST[k]$

บรรทัดที่ 10 เมื่อดำเนินการทำครบทุก k แล้วให้นำค่าต้นทุนที่ทำการครอสโอเวอร์ที่เก็บไว้ในตัวแปร $CMS_COST[k]$ มาหาค่าต่ำสุด ซึ่งโครโมโซมลูกที่มีค่าต้นทุนที่ต่ำที่สุด คือโครโมโซมลูกที่ได้จากการครอสโอเวอร์

โครโมโซมพ่อ						โครโมโซมแม่					
	P_1	P_2	P_3	P_4	P_5		P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0	b_1	0	1	0	0	0
b_2	1	0	0	0	0	b_2	0	0	0	0	0
b_3	0	1	0	0	0	b_3	1	0	1	0	0
b_4	0	0	0	1	0	b_4	0	0	0	1	0
b_5	0	0	0	0	1	b_5	0	0	0	0	1

รูปที่ 5.4 โครโมโซมพ่อแม่

ตัวอย่างการนำโครโมโซมพ่อและแม่จาก mating pool ในรูปที่ 5.4 มาทำการครอสโอเวอร์โดยใช้วิธีฮิวริสติก เริ่มต้นจากการนำโครโมโซมพ่อและแม่มาทำการเปรียบเทียบค่าต้นทุนรวมของทั้งคู่ โดยโครโมโซมพ่อ มีค่าต้นทุนรวมเป็น 900 และโครโมโซมแม่มีค่าต้นทุนรวมเป็น 950 จากนั้น เนื่องจากโครโมโซมพ่อมีค่าต้นทุนรวมที่ต่ำกว่า จึงเลือกโครโมโซมพ่อเป็นโครโมโซมตั้งต้นเป็นโครโมโซมลูก

โครโมโซมพ่อ						โครโมโซมแม่					
	P_1	P_2	P_3	P_4	P_5		P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0	b_1	0	1	0	0	0
b_2	1	0	0	0	0	b_2	0	0	0	0	0
b_3	0	1	0	0	0	b_3	1	0	1	0	0
b_4	0	0	0	1	0	b_4	0	0	0	1	0
b_5	0	0	0	0	1	b_5	0	0	0	0	1

โครโมโซมลูก					
	P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0
b_2	0	0	0	0	0
b_3	1	1	0	0	0
b_4	0	0	0	1	0
b_5	0	0	0	0	1

CMS_COST[1] = 900

	P_1	P_2	P_3	P_4	P_5
b_1	0	1	1	0	0
b_2	1	0	0	0	0
b_3	0	0	0	0	0
b_4	0	0	0	1	0
b_5	0	0	0	0	1

CMS_COST[2] = 850

	P_1	P_2	P_3	P_4	P_5
b_1	0	0	0	0	0
b_2	1	0	0	0	0
b_3	0	1	1	0	0
b_4	0	0	0	1	0
b_5	0	0	0	0	1

CMS_COST[3] = 800

	P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0
b_2	1	0	0	0	0
b_3	0	1	0	0	0
b_4	0	0	0	1	0
b_5	0	0	0	0	1

CMS_COST[4] = 900

	P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0
b_2	1	0	0	0	0
b_3	0	1	0	0	0
b_4	0	0	0	1	0
b_5	0	0	0	0	1

CMS_COST[5] = 900

รูปที่ 5.5 แสดงตัวอย่างการทำการครอสโอเวอร์แบบฮิวริสติก

เมื่อได้โครโมโซมดั้งเดิมที่จะเป็นโครโมโซมลูกแล้ว จากนั้นจึงเริ่มนำคอลัมน์แรกของโครโมโซมแม่ ไปแทนที่ในคอลัมน์แรกของโครโมโซมพ่อ จากนั้นทำการคำนวณหาค่าต้นทุนของโครโมโซมดั้งเดิม นำไปเก็บค่าไว้ใน $CMS_COST[k]$ ซึ่งจะทำการคำนวณทุกๆ คอลัมน์ ดังตัวอย่างในรูปที่ 5.5 จะเห็นได้ว่าค่า $CMS_COST[3]$ ได้ค่าต้นทุนเป็น 800 ซึ่งเป็นค่าที่ต่ำที่สุดในรอบที่ 3 แล้วจึงนำโครโมโซมตัวนี้มาเป็น โครโมโซมลูกในรอบต่อไป

โดยสรุปหลักการของการครอสโอเวอร์โดยใช้วิธีฮิวริสติก เป็นการคัดลอกส่วนที่ดีของแต่ละโครโมโซมของในแต่รุ่นไว้ เพื่อให้ได้โครโมโซมในรุ่นใหม่ที่มีการพัฒนาไปในทางที่ดีขึ้นในรุ่นต่อไป

5.1.2 การปรับปรุงการมิวเตชันแบบฮิวริสติกโดยใช้วิธีฮิวริสติก

การมิวเตชันแบบฮิวริสติก เป็นการปรับปรุงการมิวเตชันของ SGA ซึ่งโดยปกติแล้วการทำงานของ SGA จะไม่คำนึงถึงผลการของการมิวเตชันว่าโครโมโซมจะมีวิวัฒนาการที่ดีขึ้นหรือไม่จากจุดนี้ เราได้ทำการปรับปรุงการทำงานวิธีมิวเตชันแบบสุ่ม โดยมีลำดับการทำงานแสดงอยู่ในรูปที่ 5.6 ดังนี้

Heuristic Mutation Algorithm <pre> 1: HMutation(CMS, k) 2: $Mincost = Cost(CMS)$ 3: $chk = 0;$ 4: for $i = 1$ to total edge link in $OUT(k)$ { 5: $CMS_i = CMS$ 6: on each CMS_i move '1' on the column k to the row which is the member of $OUT(k)$, but has not been selected by previous steps. 7: if (($Cost(CMS_i) < Mincost$) AND (capacity constraint of CMS_i is satisfied by Eq.(4.2))) then { 8: $Mincost = Cost(CMS_i)$ 9: Child chromosome = CMS_i 10: $chk = 1;$ 11: } 12: } 13: if ($chk = 1$) 14: return(Child chromosome) 15: else 16: return (CMS) 17: end </pre>
--

รูปที่ 5.6 แสดงขั้นตอนการทำงานของฟังก์ชันมิวเตชันแบบวิธีฮิวริสติก

การเริ่มต้นการทำงานของมิวเตชันแบบฮิวริสติก เหมือนกับวิธี SGA ในการสุ่มเลือกโครโมโซมที่มีความเหมาะสมสูงมาจาก mating pool จากนั้นใช้วิธีสุ่มเลือกค่าหมายเลขเครือข่าย P_k เพื่อที่จะนำมาเข้าฟังก์ชัน HMutation(CMS, k) เริ่มต้นโดย

บรรทัดที่ 1 จะเห็นได้ว่าพารามิเตอร์ที่รับเข้ามาในฟังก์ชันคือค่า CMS หรือ โครโมโซม และค่า k ซึ่งคือหมายเลขแถวของโครโมโซมที่จะใช้เป็นตำแหน่งของการมิวเตชัน

บรรทัดที่ 2, 3 ทำการหาค่าต้นทุนของโครโมโซมเก็บไว้ในตัวแปร *Mincost* เพื่อที่ใช้ในการเปรียบเทียบกับค่าต้นทุนใหม่ที่จะได้จากการมิวเตชัน และการตั้งค่าเริ่มต้นให้กับตัวแปร *chk* เพื่อไว้สำหรับการตรวจสอบว่าการมิวเตชันให้ผลลัพธ์ที่ดีกว่าเดิมหรือไม่

บรรทัดที่ 4 เป็นรูปในการหาตำแหน่งที่ดีที่สุด โดยการย้าย “1” ไปยังทุก b_i ภายในเซตของ $Out(k)$ ซึ่งอยู่ในแถว k ของโครโมโซม CMS_i เพื่อที่จะหาค่าที่ต้นทุนที่ต่ำที่สุดในการย้าย

บรรทัดที่ 5 ทำการคัดลอกตัวแปร CMS ที่เป็นโครโมโซมพ่อไปยังโครโมโซมชั่วคราว CMS_i ซึ่งจะเป็นโครโมโซมที่จะนำมามิวเตชันในแต่ละรอบของ i

บรรทัดที่ 6 แล้วจึงทำการมิวเตชันโครโมโซม CMS_i จะทำโดยการย้ายตำแหน่งบิตของทอส่งข้อมูล b_i ให้เป็น “1” ภายในแถวของ P_k

บรรทัดที่ 7 ทำการตรวจสอบเงื่อนไข โดยนำค่าต้นทุนของโครโมโซมใหม่ (CMS_i) แล้วนำมาเปรียบเทียบกับ ค่าต้นทุนเดิม (*Mincost*) ว่ามีค่าน้อยกว่าเดิมหรือไม่ และโครโมโซมใหม่ (CMS_i) ที่ทำการย้ายตำแหน่ง “1” ไปแล้ว จะต้องไม่ทำให้แบนวิดท์ของทอส่งข้อมูล b_i ที่ย้ายไปเกินความจุทอส่งที่เหลืออยู่ ถ้าเงื่อนไขเป็นจริงให้ดำเนินการบรรทัดที่ 8 และ 9 ต่อไป

บรรทัดที่ 8 ทำการกำหนดค่า *Mincost* ใหม่ด้วยค่าต้นทุนของโครโมโซมใหม่ (CMS_i)

บรรทัดที่ 9 คัดลอกโครโมโซมใหม่ไปยังโครโมโซมลูก (Child chromosome)

บรรทัดที่ 10 ทำการกำหนดค่า *chk* ให้เป็นค่า 1 เพื่อใช้ในการตรวจสอบว่าการทำมิวเตชันประสบความสำเร็จ

บรรทัดที่ 12 ให้ทำบรรทัดที่ 5 ถึง 11 ใหม่จนกว่าจะทำการย้าย “1” ไปยังทุกๆ b_i ในแถวของ P_k

บรรทัดที่ 13 ถ้า *chk* มีค่าเท่ากับ 1 แสดงว่ามีการมิวเตชันสำเร็จให้ ส่งค่าโครโมโซมลูก (Child chromosome) กลับไป

บรรทัดที่ 14 ถ้า *chk* มีค่าไม่เท่ากับ 1 แสดงว่าการมิวเตชันทำไม่สำเร็จให้ ส่งค่าโครโมโซมพ่อกลับไป ตัวอย่างการมิวเตชัน ของประชากรตัวที่ 1 ดูในรูปที่ 5.7

โครโมโซมพ่อ						ค่าต้นทุนรวมของ P_2 ในแต่ละ b_i						
	P_1	P_2	P_3	P_4	P_5	แบนวิทท่งเหลือ		P_1	P_2	P_3	P_4	P_5
b_1	0	0	1	0	0	80	b_1		50			
b_2	1	0	0	0	0	95	b_2		50			
b_3	0	1	0	0	0	95	b_3		100			
b_4	0	0	0	1	0	90	b_4		0			
b_5	0	0	0	0	1	95	b_5		0			

ต้นทุนรวม 900

โครโมโซมลูก												
	P_1	P_2	P_3	P_4	P_5	แบนวิทท่งเหลือ						
b_1	0	1	1	0	0	75						
b_2	1	0	0	0	0	95						
b_3	0	0	0	0	0	100						
b_4	0	0	0	1	0	90						
b_5	0	0	0	0	1	95						

ต้นทุนรวม 850

รูปที่ 5.7 แสดงตัวอย่างวิธีมิวเตชันแบบฮิวริสติก

จากรูปที่ 5.7 เริ่มต้นจาก HGA ได้ทำการเลือกโครโมโซมตัวที่ 1 จาก mating pool มามิวเตชัน จากนั้นทำการสุ่มเลือก P_k มาหนึ่งแถวเพื่อที่จะใช้ในการมิวเตชัน ซึ่งการสุ่มในตัวอย่างเราได้แถวของ P_2 มาทำการมิวเตชัน จากนั้นค้นหาค่าต้นทุนที่ต่ำที่สุดของแต่ละทอส่งข้อมูล b_i ที่กระแสข้อมูลทั้งหมดมีจุดหมายปลายทางที่ P_2 แต่ยังคงต้องคำนึงถึงกฎในการย้ายค่า “1” ไปในแต่ละทอส่งข้อมูล b_i จะต้องเป็นสมาชิกอยู่ในเซตของ $Out(2)$ เสมอ ซึ่งสมาชิกของ $Out(2)$ ประกอบไปด้วย b_1, b_2 และ b_3 จากรูปที่ 5.7 จะเห็นได้ว่าโครโมโซมพ่อมีตำแหน่ง “1” อยู่ที่ b_3 ถ้าย้ายไปที่ตำแหน่ง b_1 หรือ b_2 ซึ่งจะให้ค่าต้นทุนรวมที่ต่ำกว่า จากนั้นทำการสลับที่ตำแหน่ง “1” ของ b_3 ไป b_1 ทำให้ได้ต้นทุนรวมที่ต่ำกว่าโครโมโซมพ่อ คือ 850 หน่วย และแบนวิทท่งของทอส่งข้อมูล b_3 กลับมาเป็นค่าเริ่มต้นคือ 100 หน่วย และค่าแบนวิทท่งของ b_1 ลดลงเหลือ 75 ซึ่งยังไม่เกินค่าความจุทอส่งข้อมูล ถ้าหากมีการย้ายมาแล้วเกิดความจุทอส่งข้อมูลเต็ม หรือไม่มีค่าที่ต่ำกว่าค่าเดิมที่มีอยู่ให้ยกเลิกการย้ายตำแหน่งแล้วทำการสุ่มค่า P_k ขึ้นมาใหม่จนกว่าจะครบตามจำนวนครั้งของ k จะเห็นได้ว่าวิธีฮิวริสติกมิวเตชันจะให้ผลของโครโมโซมลูกที่ดีกว่าเก่าเสมอ

โดยสรุปการมิวเตชันและครอสโอเวอร์โดยการใช้วิธีฮิวริสติก เป็นการดัดแปลงการมิวเตชันและครอสโอเวอร์แบบเดิม ซึ่งเป็นแบบสุ่มที่ไม่มีการคำนึงว่าผลที่ได้จากการทำงานจะดีขึ้นหรือแย่ลง การนำวิธีฮิวริสติกเข้ามาประยุกต์ใช้ในการแทนการสุ่ม โดยที่ทำให้เมื่อมีการทำมิวเตชันหรือครอสโอเวอร์แล้ว ผลของโครโมโซมลูกที่จะได้ออกมาจะต้องมีค่าที่ดีขึ้นแทน