บทที่ 3

อัลกอริทึมฮิวริสติก

งานวิจัย [11] พบว่าข้อดีของการใช้แบบจำลองทางคณิตศาสตร์ให้ค่าคำตอบผลเฉลยที่เหมาะสมที่สุดหรือใช้จำนวนเส้นใยนำแสงในโครงข่ายน้อยที่สุด แต่ข้อเสียของแบบจำลองนี้ก็คือเมื่อปัญหาของโครงข่ายที่ออกแบบมีขนาดที่ใหญ่หรือซับซ้อนมาก เช่น โครงข่ายมีขนาดโทโพโลยีมัลติคาสต์เซสชันหรือปริมาณทราฟฟิกขนาดใหญ่ขึ้น ย่อมส่งผลให้เวลาที่ใช้ในการคำนวณหาผลเฉลยค่อนข้างกินเวลานานหรืออาจหาคำตอบไม่ได้ในบางคำตอบ อันเนื่องมาจากจำนวนพารามิเตอร์และตัวแปรที่ไม่ทราบค่าเพิ่มจำนวนมากขึ้นตามลักษณะปัญหาแบบ NP-hard แบบจำลองทางคณิตศาสตร์นี้จึงเหมาะกับการนำไปใช้ออกแบบโครงข่ายที่มีขนาดเล็กและไม่ซับซ้อนมากนัก ในบทนี้จะกล่าวถึงอัลกอริทึมฮิวริสติก (Heuristic) ซึ่งเป็นอีกวิธีการหนึ่งที่มีความเหมาะสมกับการนำมาใช้แก้ปัญหาการออกแบบโครงข่ายที่มีขนาดใหญ่ได้ดี โดยอาศัยหลักการสุ่มหาคำตอบผลเฉลย วิธีนี้เป็นการหาค่าผลเฉลยที่เหมาะที่สุดแบบวิธีประมาณค่า ดังนั้นคำตอบของผลเฉลยที่ได้จะได้ต้นทุนความต้องการใช้เส้นใยนำแสงของโครงข่ายใกล้เคียงผลเฉลยที่เหมาะที่สุด (Sub-optimum)

ก่อนที่จะกล่าวถึงการนำอัลกอริทึมฮิวริสติกไปประยุกต์ใช้สำหรับการออกแบบการจัดสรรเส้นทางและกำหนดความยาวคลื่นบนโครงข่าย WDM หัวข้อถัดไปจะกล่าวถึงหลักการพื้นฐานของอัลกอริทึมฮิวริสติกที่ใช้ในวิทยานิพนธ์ฉบับนี้ ได้แก่ อัลกอริทึมฮิวริสติกแบบ Local Search อัลกอริทึมฮิวริสติกแบบ Simulated Annealing และอัลกอริทึมฮิวริสติกแบบ Tabu Search ซึ่งมีรายละเอียดดังนี้

3.1 อัลกอริทึมฮิวริสติกแบบ LS

อัลกอริทึมฮิวริสติกแบบ LS [13,14] มีหลักการพิจารณาคำตอบอย่างง่าย คือ หากกระบวนการค้นหาผลเฉลยสามารถค้นพบคำตอบใหม่ที่พัฒนาขึ้นหรือมีต้นทุนลดลง จะยอมรับคำตอบของผลเฉลยที่พัฒนาขึ้นนั้นเสมอ ซึ่งถือว่าเป็นจุดเริ่มต้นของกระบวนการค้นหาผลเฉลยใหม่ในรอบถัดไป ส่วนในกรณีที่ต้นทุนของผลเฉลยมีค่าเท่าเดิม จะยอมรับค่าใหม่และถือเอาเป็นจุดเริ่มต้นของกระบวนการค้นหาผลเฉลยใหม่เพื่อเปรียบเทียบในรอบถัดไปเพื่อให้เกิดการขยายขอบเขตของผลเฉลย

3.1.1 หลักการทั่วไปของอัลกอริทึมฮิวริสติกแบบ LS

อัลกอริทึมเริ่มจากกำหนดผลเฉลยที่เป็นไปได้เริ่มต้น $S_0 \in \Omega$ เมื่อ Ω คือปริภูมิการค้นหา (search space) และใช้ฟังก์ชัน *improve* ในการค้นหาผลเฉลยที่ดีกว่าผลเฉลยข้างเคียงของ S_0 ถ้าพบผลเฉลยที่ดีกว่าหรือ $S \in N(S_0)$ แล้ว จะทำการค้นหาผลเฉลยข้างเคียงหรือ $N(S)$ ของผลเฉลยใหม่ถัดไปจนกว่าจะพบผลเฉลยที่เหมาะที่สุดเฉพาะที่ (Local optimum) หรือเงื่อนไขเป็นเท็จ ซึ่งเงื่อนไขของฟังก์ชัน *improve* มีดังนี้

$$improve(S) = \begin{cases} any\ T \in N(S) & Cost(T) < Cost(S) \\ nil & otherwise \end{cases}$$

โดยสามารถอธิบายอัลกอริทึมค้นหาผลเฉลยแบบเฉพาะที่ได้ดังแสดงในรูปที่ 3.1

```
Algorithm LocalSearch (S0)
Begin
   S = S0;
   Repeat
      S* = S;
      S = improve(S*);
   Until S = nil;
   Return (S*)
End /* of LocalSearch */
```

รูปที่ 3.1 รหัสเทียมอธิบายอัลกอริทึมฮิวริสติกแบบ LS [13]

จากรูปที่ 3.1 สามารถอธิบายได้ว่า มีการกำหนดผลเฉลยเริ่มต้นเพื่อนำมาตรวจสอบกับฟังก์ชัน *improve* ถ้าเงื่อนไขเป็นจริงจะปรับผลเฉลยนั้นให้เป็นผลเฉลยใหม่แล้วนำไปใช้เปรียบเทียบกับรอบการวนซ้ำถัดไปจนกว่าเงื่อนไขจะเป็นเท็จหรือผลเฉลยปัจจุบันไม่พัฒนาขึ้นจากผลเฉลยในอดีตจนกว่าจะพบเกณฑ์การหยุดทำงาน (stop criterion) จึงจะหยุดทำงาน

3.2 อัลกอริทึมฮิวริสติกแบบ SA

หลังจาก Kirkpatrick ได้พัฒนาแนวความคิดซึ่งมีจุดเริ่มต้นมาจาก Metropolis loop โดยอาศัยความสัมพันธ์กันระหว่างการหลอมละลายโลหะกับการหาค่าที่เหมาะที่สุดเชิงการจัด (combinatorial optimization problem) เมื่อปี ค.ศ.1983 ในเวลาใกล้เคียงกันนั้น Cerny ได้เสนอแนวความคิดที่คล้ายคลึงกันขึ้นมาในปี ค.ศ.1985 จากการสังเกตการเทียบเคียงกันได้ระหว่างผล

เฉลยในการแก้ปัญหาการหาค่าที่เหมาะที่สุดกับสถานะทางกายภาพของโลหะ และการเทียบเคียงกันได้ระหว่างต้นทุนค่าใช้จ่าย (Cost) ของผลเฉลยจากการแก้ปัญหาการหาค่าที่เหมาะที่สุดเชิงการจัดและพลังงานอิสระในโลหะหลอมละลาย ผลลัพธ์ที่ได้มานั้นพบว่ามีลักษณะคล้ายคลึงกันจึงเป็นที่มาของคำว่า Simulated annealing

3.2.1 หลักการทั่วไปของอัลกอริทึมฮิวริสติกแบบ SA

อัลกอริทึมฮิวริสติกแบบ SA [13,14] จัดว่าเป็นอัลกอริทึมหนึ่งซึ่งได้รับความนิยมเป็นอย่างมากในการแก้ปัญหาการหาค่าที่เหมาะที่สุดเชิงการจัด ซึ่งเป็นปัญหาชนิด NP–hard โดยคำว่า annealing เทียบได้กับการเผาโลหะที่อุณหภูมิสูง ๆ จนโลหะเกิดการหลอมละลาย จากนั้นจึงค่อย ๆ ลดอุณหภูมิลงอย่างช้า ๆ เพื่อให้ได้ผลึกโลหะที่อยู่ในสภาวะที่เหมาะสมที่สุด อัลกอริทึมฮิวริสติกแบบ SA ถือว่าเป็นเทคนิคการค้นหาผลเฉลยแบบเฉพาะที่อย่างหนึ่ง ซึ่งมีกระบวนการทำงานแบบวนซ้ำเพื่อค้นหาผลเฉลยในปริภูมิผลเฉลยไปเรื่อยๆ จนกว่าจะได้ผลลัพธ์ที่พอใจ โดยเริ่มจากผลเฉลยเริ่มต้น อัลกอริทึมนี้เป็นฮิวริสติกแบบปรับตัวได้แบบ Non-deterministic

กระบวนการหลอมละลายโลหะนั้นจะอาศัยพื้นฐานเทคนิคการจำลองอัลกอริทึม Monte Carlo โดยกำหนดให้สถานะปัจจุบันหรือ S_i ของโลหะเทียบได้กับพลังงาน E_i และสถานะถัดไปของโลหะ (subsequent state) หรือ S_j เทียบได้กับพลังงาน E_j แล้วนำมาก่อกำเนิดกระบวนการสั่นเชิงกล โดยการเปลี่ยนจากสถานะปัจจุบันเป็นสถานะถัดไปให้มีความผิดเพี้ยนน้อยที่สุด ซึ่งสถานะใหม่ที่ถูกสร้างขึ้นมาจะถูกเลือกมาแบบสุ่ม ถ้าหากค่าพลังงานของสถานะใหม่มีค่าน้อยกว่าสถานะปัจจุบันหรือ $\Delta E = E_j - E_i \leq 0$ จึงจะยอมรับผลเฉลยเพื่อไปใช้ทดแทนผลเฉลยในอดีต แล้วสถานะปัจจุบันจะกลายเป็นสถานะใหม่ ถ้าหากผลต่างพลังงานมีค่ามากกว่า 0 ก็จะยอมรับผลเฉลยด้วยความน่าจะเป็น

$$\Pr(accept) = \exp\left(-\frac{\Delta E}{k_B T}\right) \tag{3.1}$$

โดย k_B คือค่าคงที่ Boltzmann และ T คืออุณหภูมิปัจจุบันของการหลอมละลาย โดยเงื่อนไขการยอมรับผลเฉลยข้างต้นนี้จะกระทำซ้ำ ๆ กันหลายวงรอบ

3.2.2 หลักการทำงานของอัลกอริทึมฮิวริสติกแบบ SA

Kirkpatrick และ Cerny ให้ข้อสรุปไว้ว่า ผลเฉลยในการแก้ปัญหาการหาค่าที่เหมาะที่สุดเชิงการจัด เปรียบได้กับสถานะทางกายภาพ และต้นทุนของผลเฉลยเปรียบได้กับสถานะของพลังงาน โดยทำการเปรียบเทียบการหาค่าที่เหมาะที่สุดกับกระบวนการหลอมละลายโลหะไว้ว่า

การหาค่าที่เหมาะที่สุดของทั้งหมด (global optimum) มีส่วนคล้ายกับโครงสร้างผลึกโลหะแบบสมบูรณ์ (สถานะพลังงานต่ำที่สุดของโลหะ) และโครงสร้างผลึกโลหะที่ไม่สมบูรณ์ (Imperfection) เปรียบเสมือนกับการหาค่าเหมาะที่สุดเฉพาะที่

การให้ความร้อนเพื่อหลอมละลายโลหะนั้นต้องอาศัยอุณหภูมิค่อนข้างสูงเพื่อให้โลหะเกิดการเปลี่ยนสภานะ จึงต้องมีการควบคุมอุณหภูมิให้มีความเหมาะสมมากพอ สำหรับพารามิเตอร์ที่ใช้เป็นเกณฑ์ในการควบคุมอุณหภูมินี้เรียกว่า control parameter ซึ่งอัลกอริทึมฮิวริสติกแบบ SA จะอาศัยตามหลักการของอัลกอริทึม Metropolis คือ จะค่อย ๆ ปรับลดค่าพารามิเตอร์ดังกล่าวลงในแต่ละวงรอบการวนซ้ำ เพื่อให้ได้ค่าที่เหมาะสมกับรอบที่ค้นหาคำตอบ

สมมุติฐานของอัลกอริทึมฮิวริสติกแบบ SA นี้มีลักษณะเหมือนกับอัลกอริทึมฮิวริสติกแบบ LS คือ สมมุติให้มีโครงสร้างผลเฉลยข้างเคียง (Neighbourhood structure) และกลไกการทำงานปกติ เพื่อให้เกิดการเลือกเฟ้นหาผลเฉลยที่เหมาะที่สุด โดยจะขออธิบายบทนิยามที่สำคัญสำหรับกระบวนการตัดสินใจยอมรับผลเฉลยไว้ดังนี้

บทนิยาม 3.2.1 ให้ (S, f) แทนตัวอย่างของปัญหาการหาค่าเหมาะที่สุดเชิงการจัดโดยที่ S คือ ปริภูมิผลเฉลย (solution space) ใช้แทนเซตจำกัดของผลเฉลยที่เป็นไปได้ทั้งหมดและ f คือฟังก์ชันต้นทุนโดยมีนิยามว่า

$$f : S \to \mathbb{R} \tag{3.2}$$

i และ j เป็นผลเฉลยในอดีตและผลเฉลยปัจจุบันของต้นทุน $f(i)$ และ $f(j)$ ตามลำดับ โดยที่เกณฑ์การยอมรับค่าจะเป็นไปตามความน่าจะเป็นในการยอมรับค่า

$$\Pr(accept\ j) = \begin{cases} 1 & \text{if } f(j) \le f(i) \\ \exp\left(-\dfrac{f(j)-f(i)}{c}\right) & \text{if } f(j) > f(i) \end{cases} \tag{3.3}$$

เมื่อ $c \in \mathbb{R}^+$ แทนพารามิเตอร์ควบคุมอุณหภูมิ

บทนิยาม 3.2.2 การเปลี่ยนสถานะ (transition) คือผลลัพธ์ที่เกิดจากกระบวนการเปลี่ยนแปลงผลเฉลยปัจจุบันหนึ่งไปเป็นอีกผลเฉลยหนึ่ง ซึ่งกระบวนการที่ทำให้เกิดการเปลี่ยนสถานะดังกล่าวประกอบด้วยกันสองส่วน คือ ผลลัพธ์จากกลไกของการทำงานตามปกติเอง และผลลัพธ์จากการพิจารณาตัดสินเกณฑ์การยอมรับค่า

หอสมุดกลาง ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึมฮิวริสติกแบบ SA อธิบายได้ดังรูปที่ 3.2 โดยกำหนดค่า c เริ่มต้นให้มีค่ามาก ๆ เปรียบเสมือนกับกำหนดอุณหภูมิให้สูงเพื่อให้ยอมรับผลเฉลยที่ด้อยลงได้สูงในรอบแรก ๆ แล้วจึงลดค่า c ลงเพื่อให้ยอมรับค่าผลเฉลยที่ด้อยได้น้อยลงกว่าเดิม จนกระทั่งค่า c เข้าใกล้ศูนย์ผลเฉลยที่ด้อยก็จะไม่มีอีกเลยซึ่งมีลักษณะคล้ายกับการค้นหาผลเฉลยแบบเฉพาะที่ และค่าความน่าจะเป็นของการยอมรับค่านั้นจะเปรียบเทียบค่าระหว่าง $\exp(-(f(j)-f(i))/c)$ กับค่าที่ได้จากการสุ่มซึ่งมีการกระจายตัวแบบยูนิฟอร์มในช่วง [0,1) เพื่อใช้ในการตัดสินใจว่าจะยอมรับคำตอบของผลเฉลยในแต่ละรอบหรือไม่

```
procedure Simulated Annealing( );

begin
  Initialize (i_start, c_0, L);
  k := 0;
  i := i_start;
  repeat
    for l := 1 to L do
    begin
     Generate (j from S_i);
     if f(j) < f(i) then i:=j
     else
       if exp(-(f(j)-f(i))/c_k) > random[0,1) then
         i := j;
       end
     end;
     k := k+1;
     c_k := α · c_k;
  until stopcriterion
end;
```

รูปที่ 3.2 รหัสเทียมอธิบายอัลกอริทึมฮิวริสติกแบบ SA [14]

ส่วนค่าของพารามิเตอร์ควบคุมอุณหภูมิจะปรับให้ลดลงเรื่อย ๆ ตามสมการ (3.4)

$$c_{k+1} = \alpha \cdot c_k, \quad k = 1, 2, \ldots, \tag{3.4}$$

โดยที่ α เป็นค่าคงที่ที่มีค่าน้อยกว่าแต่ใกล้เคียงกับ 1 (นิยมใช้ระหว่าง 0.8 ถึง 0.99) [13,14] ส่วนพารามิเตอร์ L เป็นจำนวนรอบการทำงานที่ใช้ในการเปลี่ยนสถานะ ซึ่งเป็นไปตามเงื่อนไขการยอมรับคำตอบที่ด้อยลงของอัลกอริทึม Metropolis

3.3 อัลกอริทึมฮิวริสติกแบบ TS

อัลกอริทึมฮิวริสติกแบบ TS [13-16] เป็นกระบวนการค้นหาผลเฉลยแบบเฉพาะที่อย่างหนึ่งที่จะให้ค่าคำตอบผลเฉลยที่ใกล้เคียงกับผลเฉลยที่เหมาะสมที่สุดสำหรับกรณีที่ปัญหาเป็นแบบ NP-hard โดยใช้เทคนิคการค้นหาที่ใช้โครงสร้างของหน่วยความจำ (memory) เข้ามาเป็นส่วนประกอบ ดังนั้น อัลกอริทึมฮิวริสติกแบบ TS จึงถูกเรียกว่าเป็นอัลกอริทึมแบบ meta-heuristics ซึ่งนำเสนอโดย Glover [13,15,16] เมื่อปี ค.ศ.1986 ความหมายของคำว่าตาบู (Tabu) คือ ข้อห้าม หรือ สิ่งต้องห้าม พูดอีกนัยหนึ่งก็คือ การหลีกเลี่ยงการเกิดการเลือกผลเฉลยใหม่ที่ซ้ำซ้อน (cyclic) กับผลเฉลยในอดีตที่ไม่ก่อให้เกิดการพัฒนาขึ้นของคำตอบ ซึ่งเป็นที่มาของการนำหน่วยความจำเข้ามาเป็นส่วนประกอบของอัลกอริทึม

3.3.1 หลักการทั่วไปของอัลกอริทึมฮิวริสติกแบบ TS

ในการเพิ่มความหลากหลายของชุดคำตอบของอัลกอริทึมฮิวริสติกแบบ TS จะป้องกันไม่ให้ผลเฉลยใหม่ซ้ำกับคำตอบของผลเฉลยในอดีต โดยในแต่ละรอบของกระบวนการค้นหาคำตอบใหม่ จะมีการพิจาณาการย้าย (Move) ผลเฉลยที่มีโอกาสทำให้คำตอบของผลเฉลยใหม่ที่จะค้นหาซ้ำซ้อนกับคำตอบของผลเฉลยในอดีต การย้ายเหล่านี้จะถูกห้ามและถูกบันทึกลงในหน่วยความจำที่เรียกว่า Tabu list อย่างไรก็ตาม การย้ายที่อยู่ใน list นี้จะถูกห้ามอยู่เพียงระยะเวลาหนึ่งเท่านั้น หลังจากนั้นจะถูกลบออกจาก list และอนุญาตให้กระทำได้ดังเดิม ดังนั้นการกำหนดความยาวของ Tabu list คือ การกำหนดระยะเวลาที่การย้ายแต่ละแบบจะถูกห้ามนั่นเอง

Tabu restrictions เป็นข้อจำกัดของ tabu list โดยจะเคลื่อนย้ายตาบู (Tabu move) ได้ก็ต่อเมื่อเงื่อนไข Tabu restrictions เป็นจริง Tabu search ประกอบไปด้วยรูปแบบการค้นหาคำตอบ 2 รูปแบบที่สำคัญคือ การค้นหาผลเฉลยโดยการใช้ความจำระยะสั้น (Short-term memory search) และการค้นหาผลเฉลยโดยการใช้ความจำระยะยาว (Long-term memory search) ซึ่งการค้นหาผลเฉลยโดยการใช้ความจำระยะสั้นนั้นเป็นหลักการสำคัญและเป็นหลักการเบื้องต้นของวิธี Tabu search การใช้หน่วยความจำระยะสั้นหมายถึง การค้นหาที่จดจำอดีตหรือประสบการณ์การค้นหาที่ผ่านมาเพียงระยะสั้น ในทางตรงข้ามกับการใช้ความจำระยะยาวนั้นจะต้องจดจำอดีตหรือประสบการณ์ที่ผ่านมาตลอดเพื่อช่วยให้การค้นหาคำตอบเป็นไปอย่างมีประสิทธิภาพมากขึ้น โดยวิทยานิพนธ์นี้จะศึกษาเฉพาะการนำ Short-term memory มาประยุกต์เท่านั้น

3.3.2 หลักการทำงานของอัลกอริทึมฮิวริสติกแบบ TS

```
Algotithm Tabu_Search( );
Ω    : เซตของผลเฉลยที่เป็นไปได้ทั้งหมด
S    : ผลเฉลยปัจจุบัน
S*   : ผลเฉลยที่ดีที่สุด
Cost : ฟังก์ชันวัตถุประสงค์
N(S) : ผลเฉลยข้างเคียง (S ∈ Ω)
V*   : ผลเฉลยข้างเคียงที่นำมาสุ่มคัดเลือก
T    : Tabu list
AL   : เงื่อนไขการยอมรับคำตอบ

begin
Start with an initial feasible solution s ∈ Ω
Initialize tabu list and aspiration level ;
for fixed number of iterations do
  Generate neighbor solutions V* ⊂ N(S) ;
  Find best S* ∈ V* ;
  if move S to S* is not in T then
    Accept move and update best solution ;
    Update tabu list and aspiration level ;
    Increment iteration number ;
  else
    if Cost(S*) < AL then
      Accept move and update best solution ;
      Update tabu list and aspiration level ;
      Increment iteration number ;
    endif
  endif
endfor
end.
```

รูปที่ 3.3 รหัสเทียมอธิบายอัลกอริทึม TS [13]

รูปที่ 3.3 อธิบายอัลกอริทึมการสร้าง TS อย่างง่าย โดยกระบวนการเริ่มจากการกำหนดผลเฉลยที่เป็นไปได้เริ่มต้น S เป็นผลเฉลยปัจจุบัน ในปริภูมิการค้นหา Ω นิยามผลเฉลยข้างเคียงของแต่ละ S เป็น $N(S)$ การสุ่มตัวอย่างหนึ่งจากผลเฉลยข้างเคียงแทนด้วย $\mathbf{V}^* \subset N(S)$ และโอกาสที่เป็นไปได้อย่างหนึ่ง คือการใช้ผลเฉลยข้างเคียงทั้งหมดที่แทนด้วย $\mathbf{V}^* = N(S)$ แต่โดยทั่วไปแล้ววิธีนี้ไม่สามารถทำได้จริงเนื่องจากต้องสิ้นเปลืองเวลาที่ใช้ในการคำนวณมาก จึงต้องสุ่มบริเวณพื้นที่ข้างเคียงที่มีขนาดเล็ก ($\mathbf{V}^* \subset N(S)$) ที่เรียกว่าผลเฉลยทดลองสุ่ม (trial) ($|\mathbf{V}^*| = n << |N(S)|$) โดยจะสุ่มเลือกจนกระทั่งได้ผลเฉลยที่ใกล้เคียงกับค่าที่เหมาะที่สุด กล่าวคือ เลือก $S^* \in \mathbf{V}^*$ เพื่อใช้ในการพิจารณาผลเฉลยถัดไป การย้าย S ไป S^* นั้นจะย้ายได้ก็ต่อเมื่อ S^* ด้อยกว่า S หรือ $Cost(S^*) > Cost(S)$ นั่นเอง

ในการเลือกการย้ายที่เหมาะที่สุดใน $\mathbf{V}^*$ นั้นอยู่บนสมมุติฐานที่ว่าการย้ายที่ดีควรจะมีค่าใกล้เคียงกับผลเฉลยที่ให้ค่าคำตอบใกล้เคียงกับผลเฉลยที่เหมาะมากที่สุดจึงต้องพิจารณาผลเฉลยทดลอง $S^* \in \mathbf{V}^*$ ด้วย เพราะผลเฉลยทดลองที่เหมาะที่สุดนั้นอาจทำให้ผลเฉลยปัจจุบันดีหรือไม่ดีขึ้นก็ได้ ลักษณะเช่นนี้เป็นการหลีกเลี่ยงการเกิดค่าที่เหมาะที่สุดเฉพาะที่ได้

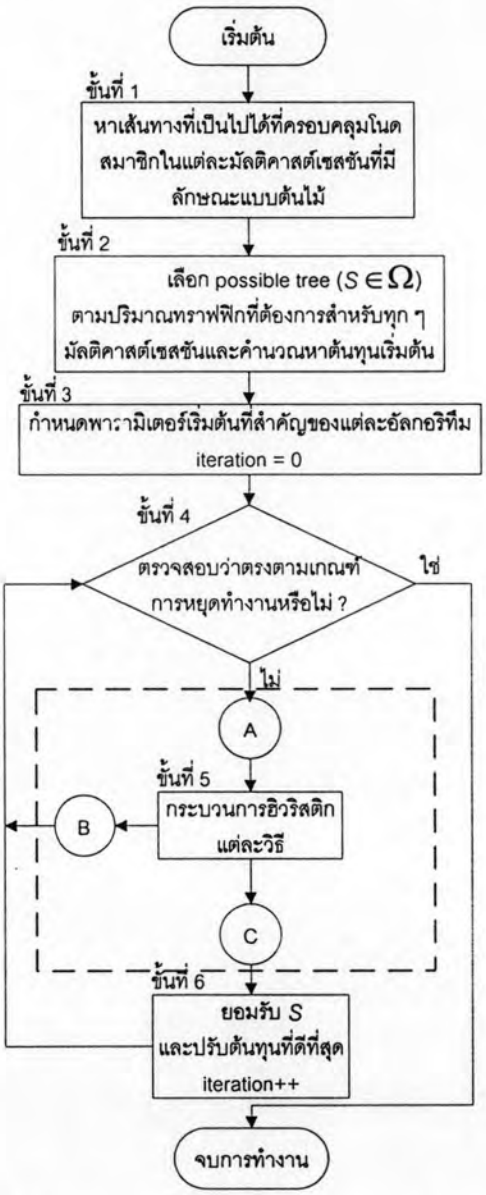
Aspiration criterion เป็นกลไกอย่างหนึ่งที่นำมาพิจารณาใช้หากเกิดสถานะ tabu ขึ้นภายใน tabu list หากการย้ายนั้นดีพอหรือผ่านเกณฑ์การยอมรับผลเฉลยก็จะเกิดการยอมรับคำตอบ

พิจารณารูปที่ 3.3 อีกครั้ง เริ่มแรกจะกำหนดผลเฉลยปัจจุบันเริ่มต้นให้เป็นผลเฉลยที่เหมาะที่สุดในการสำเนาผลเฉลยปัจจุบันย่อมส่งผลต่อการย้ายของเซตผลเฉลยใหม่ ฉะนั้นผลเฉลยที่เหมาะสมที่สุดจะถูกเลือกขึ้นมา ถ้าการย้ายนั้นไม่ใช่ตาบูแล้ว ผลเฉลยดังกล่าวจะกลายเป็นผลเฉลยปัจจุบัน แต่ถ้าการย้ายนั้นเป็นตาบูก็ต้องทำการตรวจสอบเกณฑ์การยอมรับค่า ถ้าเงื่อนไขดังกล่าวเป็นจริง จะเคลื่อนย้ายค่าคำตอบหรือคุณสมบัติบางอย่างของมันเก็บไว้ใน tabu list (ขนาดความยาวของ tabu list ที่นิยมใช้อยู่ในช่วงระหว่าง 5 ถึง 12 [13,15]) มิฉะนั้นมันอาจจะเลือกค่าที่ให้ผลเฉลยซ้ำเดิมทำให้สิ้นเปลืองเวลา ถ้าผลเฉลยถัดไปดีกว่าผลเฉลยที่เหมาะสมที่สุดในอดีตแล้วต้องทำการปรับผลเฉลยที่เหมาะสมที่สุดใหม่ เมื่อใดก็ตามที่มีการยอมรับการย้ายจะต้องทำการเพิ่มค่ารอบการปรับตัว กระบวนการจะทำต่อเนื่องเช่นนี้ไปเรื่อย ๆ จนตรงตามเกณฑ์การหยุด อันประกอบไปด้วย

- จะหยุดทำก็ต่อเมื่อมีการปรับจำนวนรอบจนครบตามที่กำหนดไว้
- จะหยุดทำหลังจากปรับค่าไปจนกระทั่งค่าของฟังก์ชันวัตถุประสงค์ (objective function) ไม่มีการพัฒนาคำตอบที่ดีกว่าเดิม

3.4 การประยุกต์ใช้อัลกอริทึมฮิวริสติกสำหรับการออกแบบการจัดสรรเส้นทางและกำหนดความยาวคลื่น

หัวข้อนี้อธิบายการนำอัลกอริทึมฮิวริสติกแต่ละวิธีมาประยุกต์ใช้ออกแบบการจัดสรรเส้นทางและกำหนดความยาวคลื่นเมื่อโครงข่ายทำงานอยู่ในสภาวะปกติสำหรับโครงข่าย WDM ทั้งกรณีที่มีและไม่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น โดยรูปที่ 3.4 แสดงผังงานอัลกอริทึมฮิวริสติกโดยรวมที่ใช้ออกแบบการจัดสรรเส้นทางและกำหนดความยาวคลื่น



รูปที่ 3.4 ผังงานอัลกอริทึมฮิวริสติกสำหรับกระบวนการจัดสรรเส้นทางบนโครงข่าย WDM

พิจารณารูปที่ 3.4 สามารถอธิบายการทำงานของผังงานอัลกอริทึมฮิวริสติกได้ดังนี้

- <u>กรณีที่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น</u>

ขั้นที่ 1 หาปริภูมิการค้นหา (Search space) ที่มีลักษณะเส้นทางเป็นโครงสร้างต้นไม้โดยมีการใช้ข่ายเชื่อมโยงน้อยสุดที่เป็นไปได้และครอบคลุมทุกโนดสมาชิกในแต่ละมัลติคาสต์เซสชัน เพื่อใช้สำหรับสุ่มหาคำตอบผลเฉลย

ขั้นที่ 2 กำหนดเส้นทางเริ่มต้นให้กับทุก ๆ ช่องสัญญาณที่แต่ละมัลติคาสต์เซสชันต้องการ และคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงเริ่มต้นให้เป็นค่าที่เหมาะที่สุดสำหรับทดสอบจำนวนความต้องการใช้เส้นใยนำแสงในรอบถัดไป

ขั้นที่ 3 กำหนดพารามิเตอร์เริ่มต้นที่อัลกอริทึมฮิวริสติกแต่ละแบบต้องการและกำหนดรอบการวนซ้ำเริ่มต้นเท่ากับ 0

ขั้นที่ 4 ตรวจสอบว่าพบเงื่อนไขตามเกณฑ์การหยุดการทำงานหรือไม่ ถ้าใช่ ให้สิ้นสุดการทำงาน ถ้าไม่ใช่ให้ทำจุดต่อ A ซึ่งเป็นอินพุทให้กับขั้นที่ 5 ต่อไป

ขั้นที่ 5 ทำกระบวนการตัดสินใจยอมรับผลเฉลยฮิวริสติกแต่ละวิธี โดยจะให้เอาท์พุทไปที่จุดต่อ B หรือจุดต่อ C ถ้าเอาท์พุทปรากฏที่จุดต่อ B ให้ทำขั้นที่ 4 อีกครั้ง และถ้าเอาท์พุทปรากฏที่จุดต่อ C ให้ทำขั้นที่ 6 ต่อไป โดยจะอธิบายกระบวนการตัดสินใจนี้ในหัวข้อที่ 3.6

ขั้นที่ 6 ยอมรับผลเฉลยถัดไปแล้วปรับค่าจำนวนความต้องการใช้เส้นใยนำแสงที่เหมาะที่สุดใหม่ และปรับรอบการทำงานขึ้นอีก 1 รอบ จากนั้นให้ทำขั้นที่ 4

- <u>กรณีที่ไม่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น</u>

ขั้นที่ 1 หาปริภูมิการค้นหาที่มีลักษณะเส้นทางเป็นโครงสร้างต้นไม้โดยมีการใช้ข่ายเชื่อมโยงน้อยสุดที่เป็นไปได้และครอบคลุมทุกโนดสมาชิกในแต่ละมัลติคาสต์เซสชัน เพื่อใช้สำหรับสุ่มหาคำตอบผลเฉลย

ขั้นที่ 2 กำหนดเส้นทางเริ่มต้นให้กับทุก ๆ ช่องสัญญาณที่แต่ละมัลติคาสต์เซสชันต้องการโดยกำหนดความยาวคลื่นที่เหมาะสมให้กับแต่ละเส้นทางที่เลือก (1 เส้นทางต่อ 1 ความยาวคลื่น) และคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงเริ่มต้นให้เป็นค่าที่เหมาะที่สุดสำหรับทดสอบจำนวนความต้องการใช้เส้นใยนำแสงในรอบถัดไป

ขั้นที่ 3 กำหนดพารามิเตอร์เริ่มต้นที่อัลกอริทึมฮิวริสติกแต่ละแบบต้องการและกำหนดรอบการวนซ้ำเริ่มต้นเท่ากับ 0

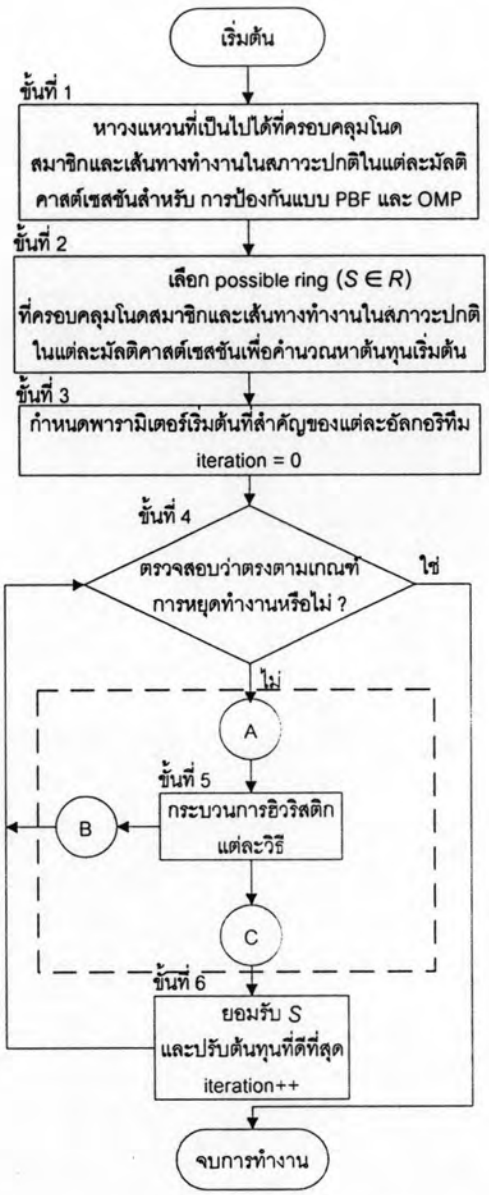
ขั้นที่ 4 ตรวจสอบว่าพบเงื่อนไขตามเกณฑ์การหยุดการทำงานหรือไม่ ถ้าใช่ ให้สิ้นสุดการทำงาน ถ้าไม่ใช่ให้ทำจุดต่อ A ซึ่งเป็นอินพุทให้กับขั้นที่ 5 ต่อไป

ขั้นที่ 5 ทำกระบวนการตัดสินใจยอมรับผลเฉลยฮิวริสติกแต่ละวิธี โดยจะให้เอาท์พุทไปที่จุดต่อ B หรือจุดต่อ C ถ้าเอาท์พุทปรากฏที่จุดต่อ B ให้ทำขั้นที่ 4 อีกครั้ง และถ้าเอาท์พุทปรากฏที่จุดต่อ C ให้ทำขั้นที่ 6 ต่อไป โดยจะอธิบายกระบวนการตัดสินใจนี้ในหัวข้อที่ 3.6

ขั้นที่ 6 ยอมรับผลเฉลยถัดไปแล้วปรับค่าจำนวนความต้องการใช้เส้นใยนำแสงที่เหมาะที่สุดใหม่ และปรับรอบการทำงานขึ้นอีก 1 รอบ จากนั้นให้ทำขั้นที่ 4

3.5 การประยุกต์ใช้อัลกอริทึมฮิวริสติกสำหรับการออกแบบการป้องกันโครงข่ายเมื่อหนึ่งข่ายเชื่อมโยงได้รับความเสียหาย

หัวข้อนี้อธิบายการนำอัลกอริทึมฮิวริสติกแต่ละวิธีมาประยุกต์ใช้ออกแบบการป้องกันโครงข่ายเมื่อหนึ่งข่ายเชื่อมโยงได้รับความเสียหายสำหรับโครงข่าย WDM ทั้งกรณีที่มีและไม่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น โดยผังงานในรูปที่ 3.5 จะใช้อธิบายการทำงานอัลกอริทึมฮิวริสติกของการป้องกันโครงข่ายเมื่อหนึ่งข่ายเชื่อมโยงได้รับความเสียหาย จากบทที่ 2 จะเห็นว่าหลักการทำงานของกลยุทธ์การป้องกันโครงข่ายแบบ PBF และ OMP จะอาศัยแนวคิดการสร้างวงแหวนหลายวงเพื่อจัดสรรเป็นเส้นทางสำรองเหมือนกันทั้งสองวิธี ดังนั้นจึงต้องทำการหาวงแหวนที่เป็นไปได้ที่ครอบคลุมโนดสมาชิกและเส้นทางทำงานที่จัดสรรไว้แล้วในสภาวะปกติมาก่อนเพื่อนำไปใช้สุ่มเลือกเส้นทางวงแหวนสำหรับอัลกอริทึมฮิวริสติกแต่ละวิธี



รูปที่ 3.5 ผังงานอัลกอริทึมฮิวริสติกสำหรับกระบวนการป้องกันโครงข่ายเมื่อหนึ่งข่ายเชื่อมโยงได้รับความเสียหายบนโครงข่าย WDM

กลยุทธ์การป้องกันโครงข่ายแบบ PBF และ OMP มีหลักการเลือกวงแหวนเหมือนกัน ดังนั้นจึงพิจารณารูปที่ 3.5 ร่วมกัน โดยอธิบายการทำงานของผังงานอัลกอริทึมฮิวริสติกได้ดังนี้

- <u>กรณีที่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น</u>

ขั้นที่ 1 หาปริภูมิการค้นหาที่เป็นไปได้ที่มีลักษณะเป็นวงแหวนโดยมีการใช้ข่ายเชื่อมโยงน้อยสุดที่เป็นไปได้และครอบคลุมทุกโนดสมาชิกและเส้นทางทำงานในสภาวะปกติของแต่ละมัลติคาสต์เซสชัน เพื่อใช้สำหรับสุ่มหาคำตอบผลเฉลย

ขั้นที่ 2 กำหนดเส้นทางวงแหวนเริ่มต้นให้กับทุก ๆ ช่องสัญญาณให้เพียงพอที่แต่ละมัลติคาสต์เซสชันต้องการ และคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงเริ่มต้นให้เป็นค่าที่เหมาะที่สุดเพื่อใช้สำหรับทดสอบจำนวนความต้องการใช้เส้นใยนำแสงในรอบถัดไป

ขั้นที่ 3 กำหนดพารามิเตอร์เริ่มต้นที่อัลกอริทึมฮิวริสติกแต่ละแบบต้องการและกำหนดรอบการวนซ้ำเริ่มต้นเท่ากับ 0

ขั้นที่ 4 ตรวจสอบว่าพบเงื่อนไขตามเกณฑ์การหยุดการทำงานหรือไม่ ถ้าใช่ ให้สิ้นสุดการทำงาน ถ้าไม่ใช่ให้ทำจุดต่อ A ซึ่งเป็นอินพุทให้กับขั้นที่ 5 ต่อไป

ขั้นที่ 5 ทำกระบวนการตัดสินใจยอมรับผลเฉลยฮิวริสติกแต่ละวิธี โดยจะให้เอาท์พุทไปที่จุดต่อ B หรือจุดต่อ C ถ้าเอาท์พุทปรากฏที่จุดต่อ B ให้ทำขั้นที่ 4 อีกครั้ง และถ้าเอาท์พุทปรากฏที่จุดต่อ C ให้ทำขั้นที่ 6 ต่อไป โดยจะอธิบายกระบวนการตัดสินใจนี้ในหัวข้อที่ 3.6

ขั้นที่ 6 ยอมรับผลเฉลยถัดไปแล้วปรับค่าจำนวนความต้องการใช้เส้นใยนำแสงที่เหมาะที่สุดชุดใหม่ และปรับรอบการทำงานขึ้นอีก 1 รอบ จากนั้นให้ทำขั้นที่ 4

- <u>กรณีที่ไม่มีการติดตั้งอุปกรณ์แปลงผันความยาวคลื่น</u>

ขั้นที่ 1 หาปริภูมิการค้นหาที่เป็นไปได้ที่มีลักษณะเป็นวงแหวนโดยมีการใช้ข่ายเชื่อมโยงน้อยสุดที่เป็นไปได้และครอบคลุมทุกโนดสมาชิกและเส้นทางทำงานในสภาวะปกติของแต่ละมัลติคาสต์เซสชัน เพื่อใช้สำหรับสุ่มหาคำตอบผลเฉลย

ขั้นที่ 2 กำหนดเส้นทางวงแหวนเริ่มต้นให้กับทุก ๆ ช่องสัญญาณให้เพียงพอที่แต่ละมัลติคาสต์เซสชันต้องการโดยคำนึงถึงความยาวคลื่นที่ถูกกำหนดในสภาวะการทำงานปกติด้วย และคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงเริ่มต้นให้เป็นค่าที่เหมาะที่สุดสำหรับทดสอบจำนวนความต้องการใช้เส้นใยนำแสงในรอบถัดไป

ขั้นที่ 3 กำหนดพารามิเตอร์เริ่มต้นที่อัลกอริทึมฮิวริสติกแต่ละแบบต้องการและกำหนดรอบการวนซ้ำเริ่มต้นเท่ากับ 0

ขั้นที่ 4 ตรวจสอบว่าพบเงื่อนไขตามเกณฑ์การหยุดการทำงานหรือไม่ ถ้าใช่ ให้สิ้นสุดการทำงาน ถ้าไม่ใช่ให้ทำจุดต่อ A ซึ่งเป็นอินพุทให้กับขั้นที่ 5 ต่อไป

ขั้นที่ 5 ทำกระบวนการตัดสินใจยอมรับผลเฉลยฮิวริสติกแต่ละวิธี โดยจะให้เอาท์พุทไปที่จุดต่อ B หรือจุดต่อ C ถ้าเอาท์พุทปรากฏที่จุดต่อ B ให้ทำขั้นที่ 4 อีกครั้ง และถ้าเอาท์พุทปรากฏที่จุดต่อ C

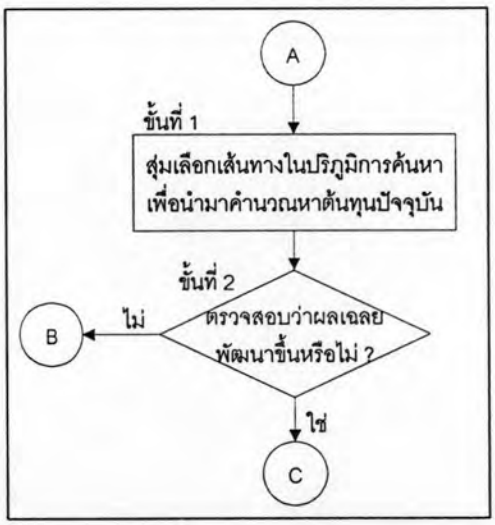
ให้ทำขั้นที่ 6 ต่อไป โดยจะอธิบายกระบวนการตัดสินใจนี้ในหัวข้อที่ 3.6

ขั้นที่ 6 ยอมรับผลเฉลยถัดไปแล้วปรับค่าจำนวนความต้องการใช้เส้นใยนำแสงที่เหมาะที่สุดชุดใหม่ และปรับรอบการทำงานขึ้นอีก 1 รอบ จากนั้นให้ทำขั้นที่ 4

3.6 กระบวนการตัดสินใจยอมรับคำตอบผลเฉลยของอัลกอริทึมฮิวริสติกแต่ละวิธี

จากขั้นที่ 5 ของผังงานรูปที่ 3.4 ซึ่งเป็นส่วนที่สำคัญสำหรับกระบวนการตรวจสอบผลเฉลยว่ามีการพัฒนาขึ้นกว่าเดิมหรือไม่ โดยจำแนกเป็นกระบวนการตัดสินใจยอมรับผลเฉลยสำหรับอัลกอริทึมฮิวริสติกแต่ละวิธีได้ดังนี้

3.6.1 การประยุกต์ใช้อัลกอริทึมฮิวริสติกแบบ LS



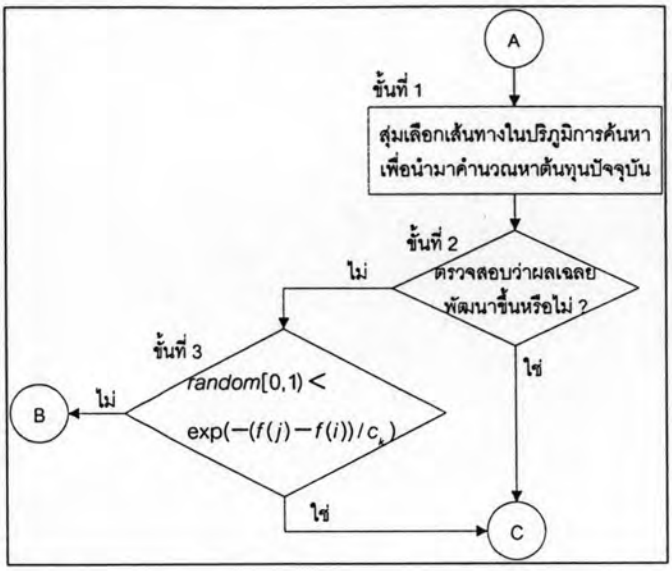
รูปที่ 3.6 ผังงานกระบวนการตัดสินใจยอมรับผลเฉลยโดยใช้อัลกอริทึมฮิวริสติกแบบ LS

พิจารณารูปที่ 3.6 สามารถอธิบายผังงานกระบวนการตัดสินใจยอมรับผลเฉลยของอัลกอริทึม LS ได้ดังนี้

ขั้นที่ 1 สุ่มเลือกเส้นทางในปริภูมิการค้นหาให้ครบตามปริมาณทราฟฟิกที่แต่ละมัลติคาสต์เซสชันต้องการของรอบการทำงานถัดไป จากนั้นนำมาคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงของผลเฉลย S เพื่อนำไปใช้ในกระบวนการยอมรับการตัดสินใจในขั้นที่ 2

ขั้นที่ 2 ตรวจสอบผลเฉลยจำนวนความต้องการใช้เส้นใยนำแสงรอบถัดไปว่าดีกว่าผลเฉลยปัจจุบันหรือไม่ ถ้าใช่ให้ทำจุดต่อ C ถ้าไม่ใช่ ให้ทำจุดต่อ B โดยย้ายทราฟฟิกกลับมาเป็นผลเฉลยที่เหมาะที่สุดชุดเดิม และเพิ่มรอบการทำงานอีก 1 รอบ

3.6.2 การประยุกต์ใช้อัลกอริทึมฮิวริสติกแบบ SA



รูปที่ 3.7 ผังงานกระบวนการตัดสินใจยอมรับผลเฉลยโดยใช้อัลกอริทึมฮิวริสติกแบบ SA

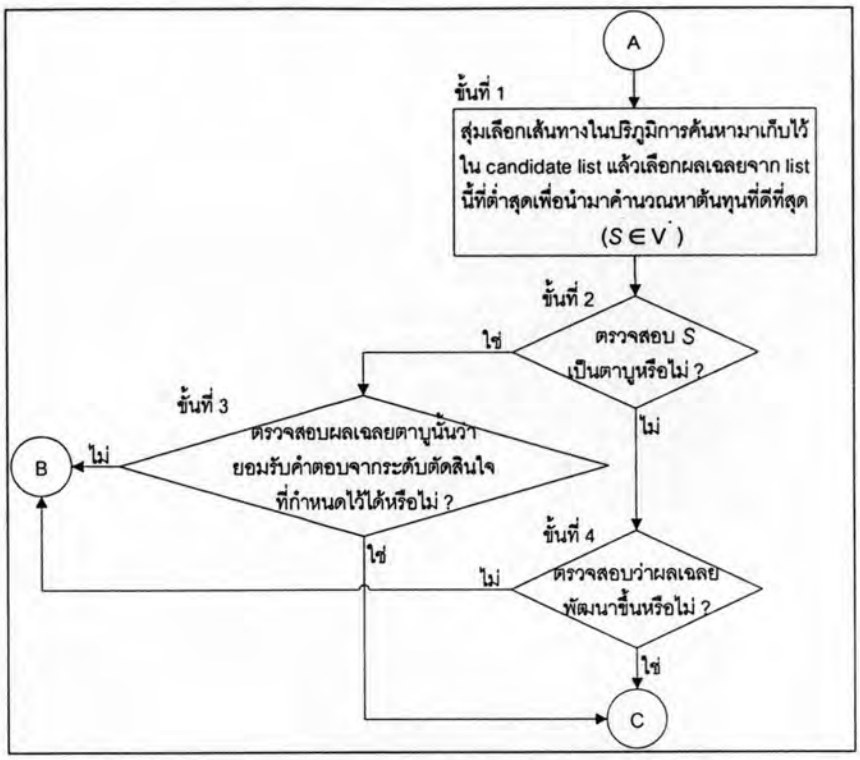
พิจารณารูปที่ 3.7 สามารถอธิบายผังงานกระบวนการตัดสินใจยอมรับผลเฉลยของอัลกอริทึม SA ได้ดังนี้

ขั้นที่ 1 สุ่มเลือกเส้นทางในปริภูมิการค้นหาให้ครบตามปริมาณทราฟฟิกที่แต่ละมัลติคาสต์เซสชันต้องการของรอบการทำงานถัดไป จากนั้นนำมาคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงของผลเฉลย S เพื่อนำไปใช้ในกระบวนการยอมรับการตัดสินใจในขั้นที่ 2

ขั้นที่ 2 ตรวจสอบผลเฉลยจำนวนความต้องการใช้เส้นใยนำแสงรอบถัดไปว่าดีกว่าผลเฉลยปัจจุบันหรือไม่ ถ้าใช่ให้ทำจุดต่อ C และปรับพารามิเตอร์ควบคุมอุณหภูมิเป็น $c_{k+1} = \alpha \cdot c_k$ เมื่อ k คือรอบการวนซ้ำในสภาวะปัจจุบัน (ตามที่กล่าวมาแล้วในหัวข้อที่ 3.2) ถ้าไม่ใช่ ให้ทำขั้นที่ 3

ขั้นที่ 3 ตรวจสอบผลเฉลยเมื่อจำนวนความต้องการใช้เส้นใยนำแสงรอบถัดไปมากกว่าจำนวนความต้องการใช้เส้นใยนำแสงปัจจุบันโดยการสุ่มค่าแบบยูนิฟอร์ม [0,1) แล้วเปรียบเทียบกับความน่าจะเป็นในการยอมรับค่า ถ้าเงื่อนไขเป็นจริงให้ทำจุดต่อ C หากเงื่อนไขเป็นเท็จให้ทำจุดต่อ B โดยย้ายทราฟฟิกกลับมาเป็นผลเฉลยที่เหมาะที่สุดชุดเดิม และเพิ่มรอบการทำงานอีก 1 รอบ

3.6.3 การประยุกต์ใช้อัลกอริทึมฮิวริสติกแบบ TS



รูปที่ 3.8 ผังงานกระบวนการตัดสินใจยอมรับผลเฉลยโดยใช้อัลกอริทึมฮิวริสติกแบบ TS

พิจารณารูปที่ 3.8 สามารถอธิบายผังงานกระบวนการตัดสินใจยอมรับผลเฉลยของอัลกอริทึม TS ได้ดังนี้

ขั้นที่ 1 สุ่มเลือกเส้นทางในปริภูมิการค้นหาให้ครบตามปริมาณทราฟฟิกที่แต่ละมัลติคาสต์เซสชันต้องการของรอบการทำงานถัดไปมาเก็บไว้ใน candidate list $(\mathbf{V}^*)$ เพื่อหาค่าที่ต่ำสุดใน list นี้ จากนั้นนำมาคำนวณหาจำนวนความต้องการใช้เส้นใยนำแสงของผลเฉลย S เพื่อนำไปใช้ในกระบวนการยอมรับการตัดสินใจในขั้นที่ 2

ขั้นที่ 2 ตรวจสอบ S ว่ามีการเลือกเส้นทางซ้ำซ้อนกับเส้นทางในอดีตหรือไม่ถ้าซ้ำจะมีสถานะเป็นตาบู แล้วทำขั้นที่ 3 หากเส้นทางที่ตรวจสอบไม่ซ้ำซ้อนกับเส้นทางเดิมในอดีตจะอัพเดทเส้นทางเก็บไว้ใน tabu list เพื่อใช้ตรวจสอบสถานะตาบู ในรอบการทำงานถัดไป แล้วจึงทำขั้นที่ 4 ต่อไป

ขั้นที่ 3 ตรวจสอบผลเฉลยจำนวนความต้องการใช้เส้นใยนำแสงที่มีสถานะ tabu ว่าสามารถยอมรับระดับการยอมรับ (aspiration level) ผลเฉลยที่กำหนดไว้หรือไม่ ถ้ายอมรับให้ทำจุดต่อ C หากไม่ยอมรับให้ทำจุดต่อ B โดยย้ายทราฟฟิกกลับมาเป็นผลเฉลยที่เหมาะที่สุดชุดเดิม และเพิ่มรอบการทำงานอีก 1 รอบ

ขั้นที่ 4 ตรวจสอบผลเฉลยจำนวนความต้องการใช้เส้นใยนำแสงรอบถัดไปว่าดีกว่าผลเฉลยปัจจุบันหรือไม่ ถ้าใช่ให้ทำจุดต่อ C ถ้าไม่ใช่ ให้ทำจุดต่อ B โดยย้ายทราฟฟิกกลับมาเป็นผลเฉลยที่เหมาะที่สุดชุดเดิม และเพิ่มรอบการทำงานอีก 1 รอบ