หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

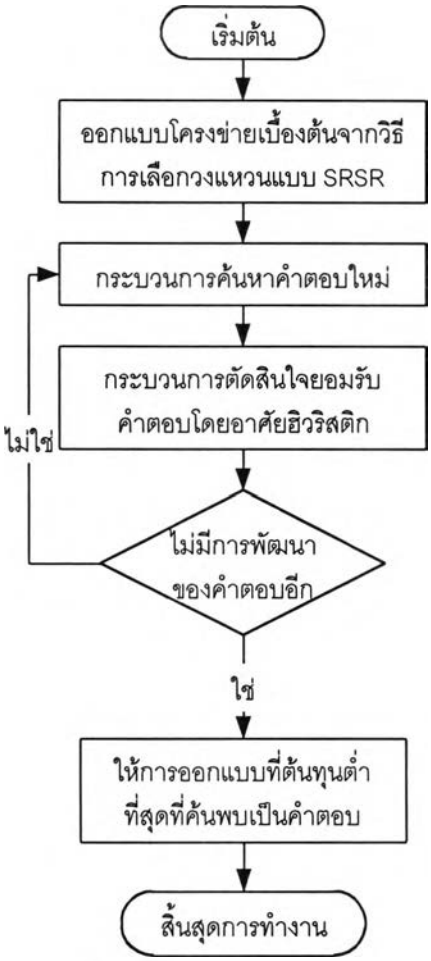
เทคนิคการปรับปรุงการออกแบบให้ดีขึ้นด้วยฮิวริสติกอัลกอริทึม

3.1 กล่าวนำ

จากบทที่แล้ว เราสามารถออกแบบโครงข่ายเพื่อรองรับทราฟฟิกทั้งหมดที่ต้องการได้ด้วยวิธีการเลือกวงแหวนเบื้องต้นแบบ SRSR แต่เนื่องจากเป็นการออกแบบที่ได้มาจากการเลือกวงแหวนให้แต่ละคู่โนดโดยอิสระต่อกัน วงแหวนที่ถูกเลือกจึงอาจมีจำนวนมาก โดยแต่ละวงรองรับทราฟฟิกของคู่โนดได้น้อย ทำให้มีการใช้ทรัพยากรไม่เต็มที่ เพื่อลดความไม่มีประสิทธิภาพนี้ จึงต้องปรับปรุงการออกแบบที่ได้จากขั้นตอนแรกไปสู่การออกแบบที่มีประสิทธิภาพมากขึ้น ในบทนี้จึงนำเสนอวิธีการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึม จากนั้นในบทที่ 4 จะเป็นการวิเคราะห์ค่าพารามิเตอร์ต่าง ๆ ที่ใช้ในฮิวริสติกอัลกอริทึมที่ได้นำเสนอ

3.2 ขั้นตอนการปรับปรุงการออกแบบ

การปรับปรุงคำตอบหรือการออกแบบแบ่งเป็น 2 ขั้นตอนสำคัญ คือ กระบวนการค้นหาคำตอบใหม่ และกระบวนการตัดสินใจเพื่อยอมรับคำตอบ กระบวนการค้นหาคำตอบใหม่ คือ การปรับเปลี่ยนการออกแบบโครงข่ายจากเดิมไปสู่การออกแบบที่อาจจะให้ต้นทุนดีขึ้น คำตอบใหม่นี้อาจได้รับการยอมรับหรือไม่ก็ได้ โดยการพิจารณาจะทำในกระบวนการตัดสินใจเพื่อยอมรับคำตอบ กระบวนการค้นหาคำตอบใหม่และกระบวนการตัดสินใจเพื่อยอมรับคำตอบจะถูกทำซ้ำหลาย ๆ รอบ (Iterative) จนกว่าจะได้คำตอบเป็นที่น่าพอใจหรือไม่เห็นการพัฒนาของคำตอบอีกภายในระยะเวลาที่กำหนดไว้ ขั้นตอนการปรับปรุงการออกแบบให้ดีขึ้นด้วยฮิวริสติกอัลกอริทึมมีผังงานดังรูปที่ 3.1



รูปที่ 3.1 ผังงานของฮิวริสติกอัลกอริทึมการปรับปรุงการออกแบบโครงข่าย

3.3 กระบวนการค้นหาคำตอบใหม่

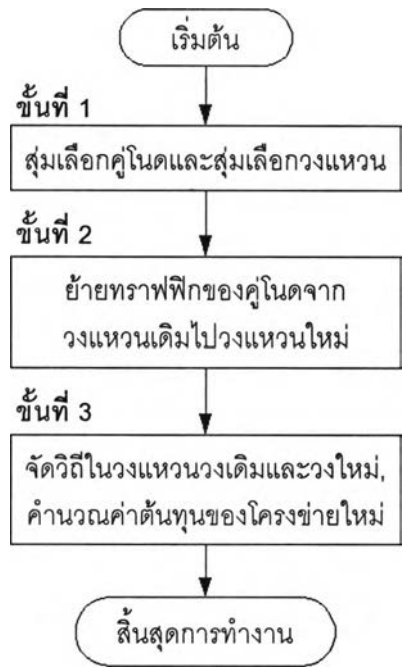
ในหัวข้อนี้จะเสนออัลกอริทึมที่ใช้ในกระบวนการค้นหาคำตอบใหม่ โดยจะอาศัยแนวทางจากการออกแบบโครงข่ายเบื้องต้นที่สรุปในส่วนวิเคราะห์ของบทที่แล้ว คือ ให้วงแหวนแต่ละวงสามารถรองรับทราฟฟิกของคู่โนดได้หลาย ๆ คู่ จำนวนวงแหวนที่เลือกควรมีจำนวนลดลงและวงแหวนควรมีขนาดเล็กเพื่อจะได้ต้นทุนต่ำ การปรับปรุงการออกแบบจึงกระทำโดยพยายามย้ายทราฟฟิกของคู่โนดบางคู่ให้มารวมกันในวงแหวนให้ได้มากขึ้น การรวมทราฟฟิกนี้มีวัตถุประสงค์เพื่อกำจัดเส้นใยแสงหรือวงแหวนที่มีประสิทธิภาพการใช้ทรัพยากรต่ำออกไป กระบวนการค้นหาคำตอบสามารถเขียนเป็นผังงานได้ดังรูปที่ 3.2 แต่ละขั้นตอนมีรายละเอียดดังนี้

ขั้นที่ 1 สุ่มเลือกคู่โนดมา 1 คู่ที่มีทราฟฟิกไม่เป็น 0, สุ่มเลือกวงแหวนมา 1 วงที่สามารถรองรับทราฟฟิกระหว่างคู่โนดนั้นได้และไม่ใช่วงแหวนเดิมที่รองรับทราฟฟิกของคู่โนดนั้นอยู่ และวงแหวนนั้นจะต้องอยู่ในชุดของวงแหวนที่ถูกเลือกไว้จากการเลือกวงแหวนแบบ

โดยวงแหวนนั้นจะต้องอยู่ในชุดของวงแหวนที่ถูกเลือกไว้จากการเลือกวงแหวนแบบ SRSR อีกด้วย ทั้งนี้การจำกัดชุดของวงแหวนมีวัตถุประสงค์เพื่อให้วงแหวนที่เลือกทั้งหมดมีจำนวนลดลง

ขั้นที่ 2 ย้ายทราฟฟิกทั้งหมดของคู่โนดจากวงแหวนเดิมไปยังวงแหวนที่เลือก

ขั้นที่ 3 จัดวิถีให้ทราฟฟิกทุกค่าในวงแหวนที่เดิมและวงแหวนใหม่, คำนวณหาค่าต้นทุนใหม่ของโครงข่าย



รูปที่ 3.2 ผังงานของกระบวนการค้นหาคำตอบใหม่

3.4 กระบวนการตัดสินใจยอมรับคำตอบ

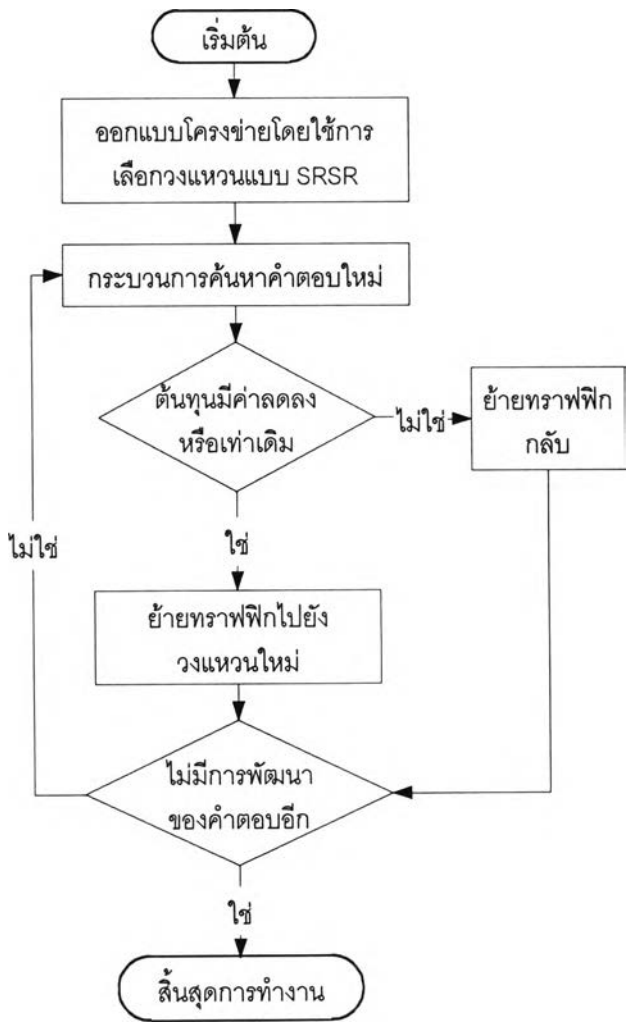
กระบวนการตัดสินใจเพื่อยอมรับคำตอบในวิทยานิพนธ์นี้จะใช้ฮิวริสติก 3 แบบ ได้แก่

- *Local Search*
- *Simulated Annealing*
- *Tabu Search*

สำหรับฮิวริสติกอีกแบบที่มีการใช้ในการแก้ปัญหาการออกแบบโครงข่าย คือ Genetic Algorithm แต่เนื่องจากได้มีการนำไปใช้ในการออกแบบโครงข่าย WDM บนพื้นฐานของโครงสร้าง

3.5 การปรับปรุงการออกแบบโดยใช้ฮิวริสติกอัลกอริทึมแบบ Local Search

ในหัวข้อนี้จะกล่าวถึงฮิวริสติกอัลกอริทึมที่ใช้การตัดสินใจยอมรับคำตอบแบบแรก คือ ฮิวริสติกอัลกอริทึมแบบ Local Search มีหลักการพิจารณาคำตอบง่าย ๆ คือ หากกระบวนการค้นหาคำตอบสามารถหาคำตอบใหม่ที่ดีขึ้น คือ มีต้นทุนลดลงได้ จะยอมรับคำตอบที่ดีขึ้นนั้นเสมอ และจะถือเอาเป็นจุดเริ่มต้นของกระบวนการค้นหาคำตอบใหม่ในรอบถัดไป แต่หากคำตอบด้อยลงจะไม่ยอมรับและจะถือเอาคำตอบเดิมเป็นจุดเริ่มต้นในการค้นหาในรอบถัดไป ส่วนในกรณีที่ต้นทุนมีค่าเท่าเดิม จะยอมรับคำตอบใหม่และถือเอาเป็นจุดเริ่มต้นของกระบวนการค้นหาคำตอบใหม่ในรอบถัดไปเพื่อให้เกิดการขยายขอบเขตของคำตอบ ฮิวริสติกอัลกอริทึมการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึมแบบ Local Search สามารถแสดงเป็นผังงานได้ดังรูปที่ 3.3



รูปที่ 3.3 ผังงานของการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึมแบบ Local Search

3.6 การปรับปรุงการออกแบบโดยใช้ฮิวริสติกอัลกอริทึมแบบ Simulated Annealing

ในหัวข้อนี้จะกล่าวถึงการนำ Simulated Annealing ซึ่งเป็นฮิวริสติกอีกชนิดหนึ่งที่จะใช้ในการปรับปรุงการออกแบบ Simulated Annealing ถูกพัฒนาดัดแปลงโดย Kirkpatrick [14]-[15] จำลองมาจากกระบวนการการให้ความร้อนแก่โลหะเพื่อให้อ่อนตัว เป็นวิธีค้นหาคำตอบได้ดีในปัญหาที่ซับซ้อนและแก้ปัญหาที่การค้นหามักจะติดอยู่ใน Local Optimum คือ การที่คำตอบที่ค้นพบจำกัดอยู่ในวงแคบ ซึ่งเกิดจากการที่กระบวนการค้นหาคำตอบยอมรับแต่คำตอบที่ดีขึ้นเท่านั้น ทำให้โอกาสที่จะค้นพบคำตอบแบบอื่น ๆ น้อย คำตอบสุดท้ายที่ได้จึงเป็นคำตอบที่ต่ำที่สุดเฉพาะในกลุ่มของคำตอบเพียงกลุ่มเดียวเท่านั้น

ใน Simulated Annealing การแก้ปัญหา Local Optimum จะทำโดยยอมรับคำตอบที่ยืดหยุ่นกว่าใน Local Search คือ จะไม่ยอมรับแต่คำตอบที่ดีขึ้นเท่านั้น แต่จะมีการยอมรับคำตอบที่ด้อยลงด้วยความน่าจะเป็นค่าหนึ่ง (P) ทำให้โอกาสที่จะค้นพบคำตอบที่ดีซึ่งอาจจะอยู่ในบริเวณอื่นเป็นไปได้มากขึ้น ค่าความน่าจะเป็นที่จะยอมรับคำตอบที่ด้อยลงดังกล่าวมีค่าไม่เท่ากันในแต่ละรอบของการค้นหาคำตอบ ในรอบแรก ๆ จะมีค่าสูงก่อน เมื่อกระบวนการค้นหาคำตอบดำเนินต่อไประยะหนึ่งจึงค่อย ๆ ถูกลดค่าลงจนเป็น 0 ดังสมการที่ (3.1) เมื่อ $Cost_{i+1}$ คือ ค่าต้นทุนของคำตอบใหม่ที่ค้นหาได้, $Cost_i$ คือ ค่าต้นทุนของคำตอบปัจจุบัน, t_i คือ ค่าอุณหภูมิในแต่ละรอบ ค่าความน่าจะเป็น P นี้มีค่าเท่ากับเอกซ์โพเนนเชียลเชิงลบของค่าความแตกต่างระหว่างต้นทุนของคำตอบเดิมกับคำตอบใหม่ในแต่ละรอบหารด้วยค่าอุณหภูมิ

ค่าอุณหภูมินี้เป็นค่าพารามิเตอร์ควบคุม (Control Parameter) ทำหน้าที่ควบคุมความน่าจะเป็นในการยอมรับคำตอบที่ด้อยลง และทำหน้าที่ปรับค่าความน่าจะเป็นให้เหมาะสมกับรอบที่ค้นหาคำตอบ ดังนั้นจึงกำหนดให้มีค่าไม่คงตัว

$$P = exp\left(-\frac{\left|Cost_{i+1} - Cost_i\right|}{t_i}\right) \tag{3.1}$$

จากสมการที่ (3.1) จะเห็นว่าหากอุณหภูมิมีค่าสูงเมื่อเทียบกับค่าความแตกต่างของต้นทุนค่าความน่าจะเป็นที่ยอมรับคำตอบที่ด้อยลงจะมีค่าสูง ในการค้นหาคำตอบด้วย Simulated Annealing จะกำหนดให้อุณหภูมิเริ่มต้นที่ใช้ในรอบแรก ๆ ของการค้นหาคำตอบมีค่าสูงมาก ๆ เมื่อเทียบกับค่าต้นทุนของโครงข่ายเริ่มต้นก่อนเพื่อให้เกิดการยอมรับคำตอบที่ด้อยลงได้ง่าย เมื่อกระบวนการค้นหาคำตอบดำเนินไป ค่าอุณหภูมินี้จะถูกลดลงจนเข้าหาค่า 0 ทำให้การยอมรับ

คำตอบที่ด้อยลงเกิดยากขึ้น จนกระทั่งเมื่ออุณหภูมิมีค่าลดลงเข้าใกล้ 0 ความน่าจะเป็นที่จะยอมรับคำตอบที่ด้อยลงจะเข้าหา 0 ด้วย กระบวนการค้นหาคำตอบหลังจากที่อุณหภูมิมีค่าใกล้ 0 มากจะมีลักษณะคล้ายกับการค้นหาคำตอบที่ใช้ Local Search เนื่องจากจะไม่มีการยอมรับคำตอบที่ด้อยลงอีก ดังนั้นในกระบวนการ Simulated Annealing เงื่อนไขในการลดค่าอุณหภูมิจึงมีความสำคัญและต้องกำหนดให้เหมาะสมกับปัญหา โดยจะกำหนดให้ค่าอุณหภูมิมีค่าคงตัวจนกระทั่งเมื่อค้นพบคำตอบที่ดีขึ้นจำนวน r ครั้ง จึงจะลดค่าอุณหภูมิตามสมการที่ (3.2) อัตราส่วนในการลดค่าอุณหภูมิ (*factor*) จะกำหนดให้เป็นค่าคงตัวระหว่าง 0.50 ถึง 0.99

ทุก r ครั้งที่ค้นพบคำตอบที่ดีขึ้น : $$t_{i+1} = factor \times t_i \qquad (3.2)$$

จากที่กล่าวมาทั้งหมด ค่าตัวแปรต่าง ๆ ทั้งหมดที่ต้องกำหนดในอัลกอริทึมให้เหมาะสมกับปัญหา ได้แก่ ค่าอุณหภูมิเริ่มต้น (t_0), เงื่อนไขในการลดค่าอุณหภูมิ คือ ค่า r ที่จะให้อุณหภูมิมีค่าคงตัวก่อนจะลดค่าลง และค่า *factor* ในการลดค่าอุณหภูมิ ผังงานของอัลกอริทึมการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึมแบบ Simulated Annealing แสดงดังรูปที่ 3.4 แต่ละขั้นตอนมีรายละเอียดดังนี้

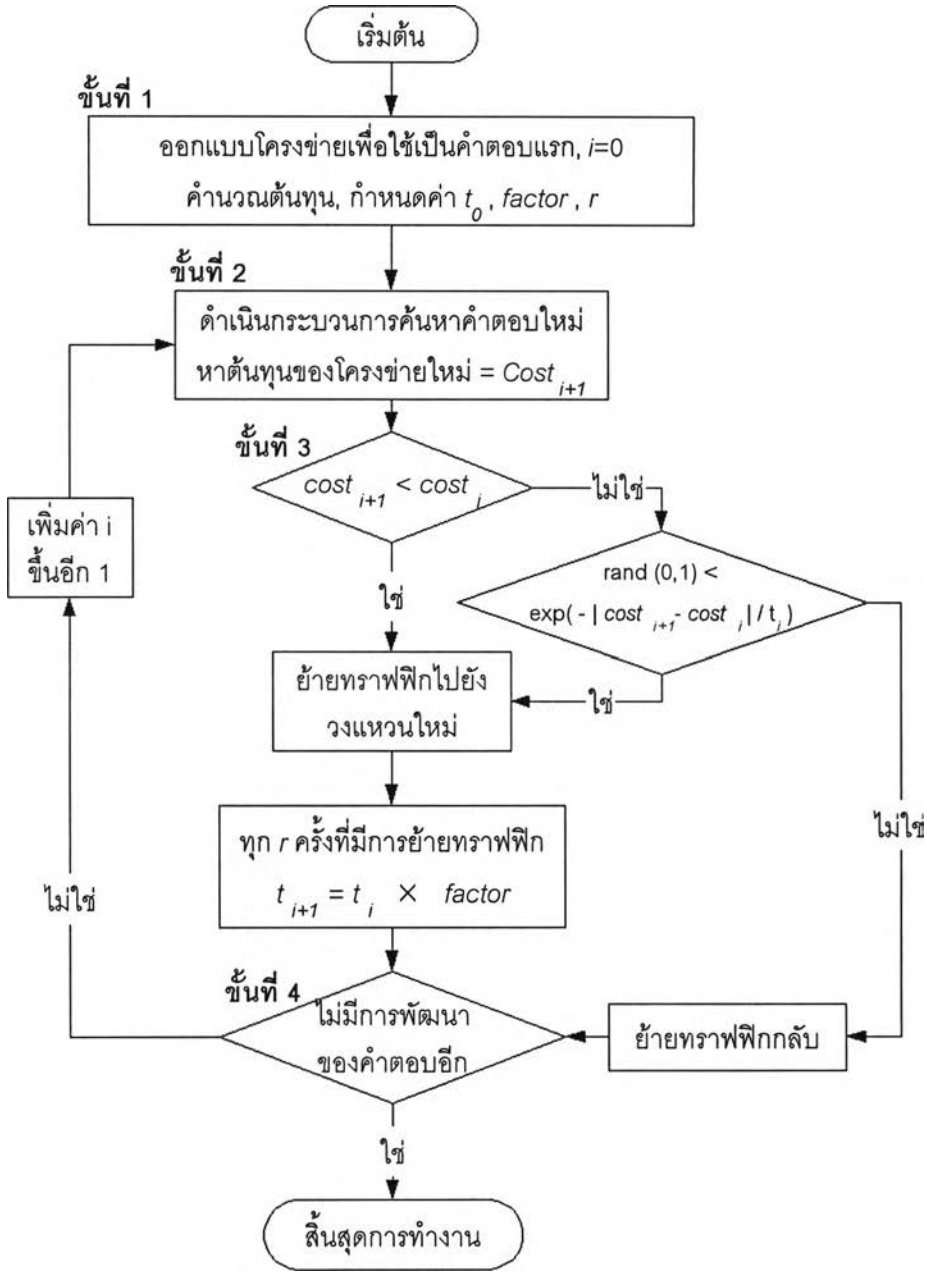
ขั้นที่ 1 หาคำตอบเบื้องต้นของการออกแบบจากวิธีการเลือกวงแหวนแบบ SRSR, คำนวณค่าต้นทุนของโครงข่ายแรก, กำหนดค่าพารามิเตอร์ต่าง ๆ ที่ต้องใช้ในอัลกอริทึม

ขั้นที่ 2 ดำเนินกระบวนการค้นหาคำตอบใหม่

ขั้นที่ 3 หากมีการพัฒนาของคำตอบ คือ ต้นทุนของโครงข่ายลดลงหรือเท่าเดิม จะย้ายทราฟฟิกไปยังวงแหวนใหม่ และยอมรับคำตอบใหม่เป็นคำตอบปัจจุบัน แต่หากต้นทุนของโครงข่ายสูงขึ้นจะยอมรับเป็นคำตอบปัจจุบันได้ด้วยความน่าจะเป็น P ซึ่งคำนวณจากสมการที่ (3.1) หากพบว่ามีการยอมรับคำตอบครบ r ครั้งหลังจากการลดอุณหภูมิครั้งล่าสุด จะลดค่าอุณหภูมิตามสมการที่ (3.2)

ขั้นที่ 4 ทำซ้ำโดยเริ่มจากขั้นที่ 2 ไปจนกว่าจะไม่พบว่ามีการพัฒนาของคำตอบให้เห็นอีกแล้วภายในจำนวนรอบที่กำหนดจึงยุติกระบวนการค้นหาคำตอบ คำตอบสุดท้ายของการออกแบบ คือ คำตอบที่ให้ค่าต้นทุนต่ำที่สุดในชุดของคำตอบที่ค้นพบทั้งหมด

I19314616



รูปที่ 3.4 ผังงานของการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึม
แบบ Simulated Annealing

3.7 การปรับปรุงการออกแบบโดยใช้ฮิวริสติกอัลกอริทึมแบบ Tabu Search

ในหัวข้อนี้จะกล่าวถึงการนำ Tabu Search ซึ่งเป็นฮิวริสติกอีกชนิดหนึ่งที่จะใช้ในการปรับปรุงการออกแบบ Tabu Search [16] ใช้ในการแก้ปัญหาที่คำตอบในกระบวนการปรับปรุงการออกแบบโครงข่ายมักจะติดอยู่ใน Local Optimum ด้วยการเพิ่มความหลากหลายของชุด

คำตอบพร้อม ๆ ไปกับการยอมรับคำตอบใหม่ทุกกรณี ไม่ว่าการออกแบบจะให้ต้นทุนดีขึ้น, เท่าเดิม หรือด้อยลงก็ตาม

ในการเพิ่มความหลากหลายของชุดคำตอบของ Tabu Search จะป้องกันไม่ให้คำตอบใหม่ซ้ำกับคำตอบเดิมในรอบก่อน ๆ โดยในแต่ละรอบของกระบวนการค้นหาคำตอบใหม่ จะมีการพิจารณาการย้าย (Move) บางอย่างที่มีโอกาสทำให้คำตอบใหม่ที่จะค้นหาซ้ำกับคำตอบเดิมในรอบก่อน ๆ การย้ายเหล่านี้จะถูกห้ามและถูกบันทึกในตารางที่เรียกว่า Tabu List อย่างไรก็ตาม การย้ายที่อยู่ในตารางนี้จะถูกห้ามอยู่เพียงชั่วคราวเท่านั้น หลังจากนั้นจะถูกลบออกจากตารางและอนุญาตให้กระทำได้ดังเดิม ดังนั้นการกำหนดความยาวของตาราง Tabu List คือ การกำหนดระยะเวลาที่การย้ายแต่ละแบบจะถูกห้ามนั่นเอง

การนำเทคนิค Tabu Search มาใช้ในอัลกอริทึมการปรับปรุงการออกแบบขึ้นอยู่กับปัญหาแต่ละแบบ สำหรับปัญหาการออกแบบโครงข่ายบนพื้นฐานของโครงสร้างแบบวงแหวนหลายวงในวิทยานิพนธ์นี้ จะนำ Tabu Search มาประยุกต์ใช้ทั้งกับในกระบวนการค้นหาคำตอบใหม่และกระบวนการตัดสินใจเพื่อยอมรับคำตอบ แบ่งเป็น 2 วิธี คือ

3.7.1 การปรับปรุงการออกแบบโดยใช้ฮิวริสติกอัลกอริทึมแบบ Tabu Search

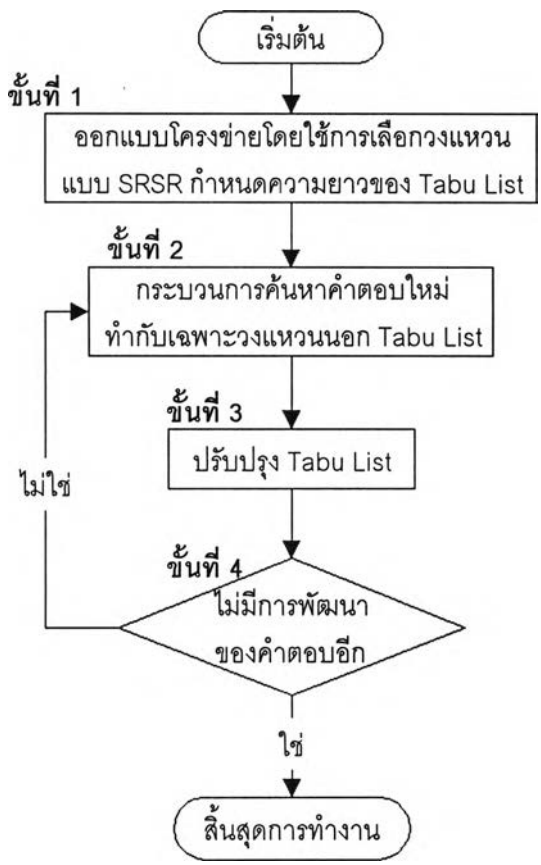
ในวิธีนี้จะนำหลักการของ Tabu Search มาใช้กับทั้งกระบวนการค้นหาคำตอบใหม่และกระบวนการตัดสินใจเพื่อยอมรับคำตอบ ในแต่ละรอบจะห้ามย้ายทราฟฟิกของคู่โนดกลับเข้าไปในวงแหวนที่เคยรองรับทราฟฟิกของคู่โนดนั้นอยู่ เพื่อไม่ให้คำตอบซ้ำกับในรอบก่อน ๆ ดังนั้นในตาราง Tabu List จะบันทึกหมายเลขวงแหวนที่เคยถูกเลือกเพื่อรองรับทราฟฟิกของแต่ละคู่โนด คำตอบหลังจากการย้ายทราฟฟิกจะได้รับการยอมรับทุกกรณี การปรับปรุงการออกแบบให้ดีขึ้นด้วยฮิวริสติกอัลกอริทึมแบบ Tabu Search มีผังงานดังรูปที่ 3.5 และมีรายละเอียดดังนี้

ขั้นที่ 1 หาคำตอบเบื้องต้นของการออกแบบจากวิธีการเลือกวงแหวนแบบ SRSR, คำนวณค่าต้นทุนของโครงข่ายแรก และกำหนดความยาวของ Tabu List ซึ่งต้องมีค่าไม่เกินจำนวนวงแหวนสูงสุดที่เป็นตัวเลือกในการย้ายทราฟฟิกคิดจากทุก ๆ คู่โนด

ขั้นที่ 2 ดำเนินกระบวนการค้นหาคำตอบใหม่ โดยมีข้อกำหนดเพิ่มเติมว่า วงแหวนที่เลือกในการย้ายทราฟฟิกจะต้องไม่อยู่ใน Tabu List

ขั้นที่ 3 ปรับปรุง (Update) ตาราง Tabu List โดยหากตารางมีสมาชิกเต็มแล้ว จะกำจัดวงแหวนที่คงอยู่ในตารางนานที่สุดออกไป วงแหวนเดิมซึ่งทราฟฟิกถูกย้ายออกไปในรอบนั้นจะถูกเพิ่มเข้ามาในตารางแทน

ขั้นที่ 4 ทำซ้ำโดยเริ่มจากขั้นที่ 2 ไปจนกว่าจะไม่พบว่ามีการพัฒนาของคำตอบให้เห็นอีกแล้วภายในจำนวนรอบที่กำหนด จึงยุติกระบวนการค้นหาคำตอบ คำตอบสุดท้ายของการออกแบบ คือ คำตอบที่ให้ค่าต้นทุนต่ำที่สุดในชุดของคำตอบที่ค้นพบทั้งหมด

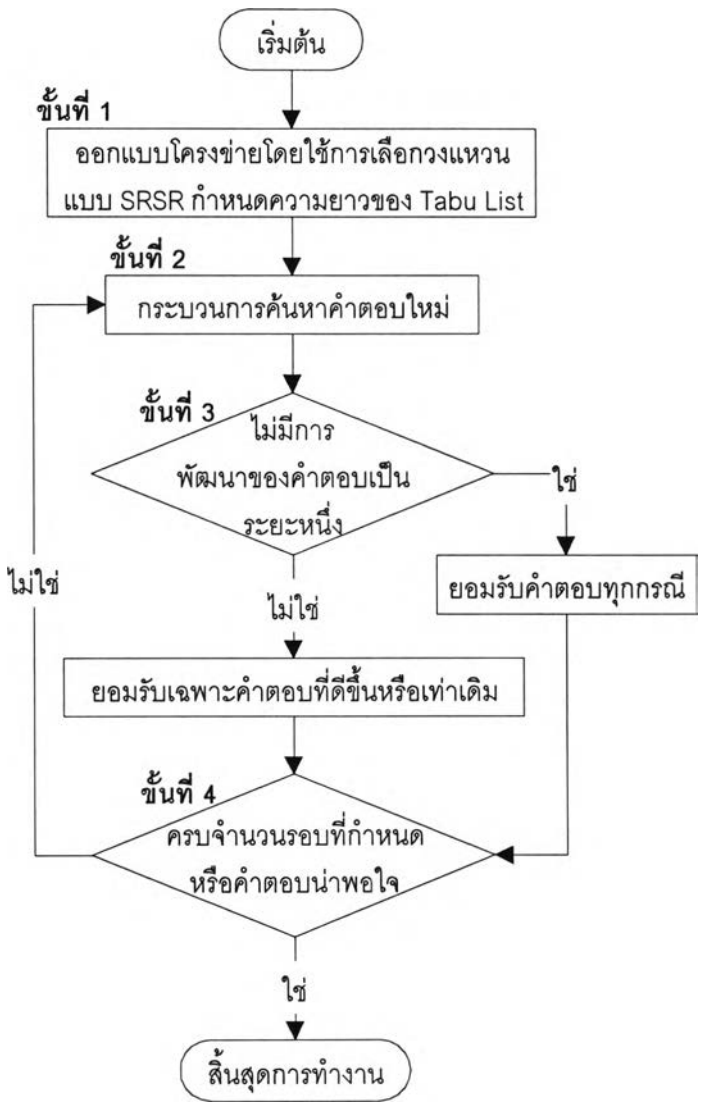


รูปที่ 3.5 ผังงานของการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึม
แบบ Tabu Search

3.7.2 การปรับปรุงการออกแบบโดยใช้ฮิวริสติกอัลกอริทึมแบบ Tabu Search ประยุกต์ร่วมกับแบบ Local Search

ในวิธีนี้จะนำหลักการของทั้ง Local Search และ Tabu Search มาประยุกต์ใช้ร่วมกัน ในช่วงแรกกระบวนการค้นหาคำตอบใหม่และกระบวนการตัดสินใจเพื่อยอมรับคำตอบจะใช้วิธีเดียวกับฮิวริสติกอัลกอริทึมแบบ Local Search คือ มีการยอมรับเฉพาะการออกแบบที่ให้

ค่าต้นทุนดีขึ้นหรือเท่าเดิมเท่านั้น แต่เมื่อไม่พบการพัฒนาของคำตอบเป็นระยะหนึ่งจะเปลี่ยนกระบวนการยอมรับคำตอบเป็นยอมรับคำตอบทุกกรณี เพื่อเปิดโอกาสให้ชุดของคำตอบมีการเปลี่ยนกลุ่มไปจากเดิม การยอมรับคำตอบในช่วงนี้จะทำแค่ระยะเวลาสั้น ๆ หลังจากนั้นจะเปลี่ยนกระบวนการยอมรับคำตอบเป็นแบบ Local Search ดังเดิม การปรับปรุงการออกแบบให้ดีขึ้นด้วยฮิวริสติกอัลกอริทึมแบบ Tabu Search ประยุกต์ร่วมกับแบบ Local Search มีผังงานดังรูปที่ 3.6



รูปที่ 3.6 ผังงานของการปรับปรุงการออกแบบด้วยฮิวริสติกอัลกอริทึม
แบบ Tabu Search ประยุกต์ร่วมกับแบบ Local Search

ขั้นที่ 1 หาคำตอบเบื้องต้นของการออกแบบจากวิธีการเลือกวงแหวนแบบ SRSR, คำนวณค่าต้นทุนของโครงข่ายแรก

ขั้นที่ 2 ดำเนินกระบวนการค้นหาคำตอบใหม่

ขั้นที่ 3 ถ้าคำตอบไม่มีการพัฒนาภายในระยะเวลาที่กำหนด ให้ยอมรับคำตอบทุกกรณีไม่ว่าต้นทุนจะดีขึ้นหรือด้อยลง แต่หากคำตอบยังมีการพัฒนาอยู่ ให้ยอมรับเฉพาะคำตอบที่ให้ต้นทุนดีขึ้นหรือเท่ากันเท่านั้น

ขั้นที่ 4 ทำซ้ำโดยเริ่มจากขั้นที่ 2 ไปจนกว่าจะครบจำนวนรอบที่กำหนดหรือได้คำตอบเป็นที่น่าพอใจ จึงยุติกระบวนการค้นหาคำตอบ คำตอบสุดท้ายของการออกแบบ คือ คำตอบที่ให้ค่าต้นทุนต่ำที่สุดในชุดของคำตอบที่ค้นพบทั้งหมด