

การค้นหาแบบตาบ Tabu Search

การค้นหาแบบตาบ หรือ tabu search (TS) เป็นขั้นตอนวิธีการคิดที่ถูกนำมาประยุกต์ใช้ในการแก้ปัญหาในระบบที่เรียกว่าปัญหาการหาค่าเหมาะที่สุดเชิงผสมผสาน (combinatorial optimization) ได้อย่างมีประสิทธิภาพ Glover [Glover, 1989][Glover, 1990] เป็นผู้ริเริ่มเสนอแนวคิดของ TS ไว้ตั้งแต่ปี พ.ศ. 2520 และหลังจากนั้น TS ก็ได้กลายมาเป็นที่ยอมรับกันอย่างกว้างขวาง ปัจจุบัน TS ได้เป็นที่ยอมรับโดยทั่วไปว่ามีความสามารถในการหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (local optimum) และยังสามารถทำการค้นหาคำตอบต่อไปจนกระทั่งให้คำตอบที่ใกล้เคียงกับคำตอบที่เหมาะสมที่สุดแบบวงกว้าง (near global optimum) ตัว TS เองยังสามารถประยุกต์ใช้งานกับหลายระบบได้อย่างไม่ยุ่งยากมากอันเนื่องมาจากหลักการและกลไกการทำงานของ TS ที่ไม่ซับซ้อนนั่นเองซึ่งทำให้ขั้นตอนการประยุกต์ใช้งานของ TS ค่อนข้างที่จะสะดวกกว่าเมื่อเปรียบเทียบกับวิธีการอื่นๆ ยกตัวอย่างเช่นวิธีการอบอ่อนจำลอง (simulated annealing) จีโนมิกส์อัลกอริทึม (genetic algorithm) เครือข่ายประสาทเทียม (artificial neural network) ฯลฯ ดังนั้นแล้วการเขียนโปรแกรมใช้งาน TS จึงอาจใช้เพียง 2-3 บรรทัดเท่านั้น

ข้อดีหลายอย่างของ TS ทำให้เป็นที่นิยมอย่างแพร่หลายและกลายเป็นเครื่องมือที่มีประสิทธิภาพสูง ในการแก้ปัญหาการหาค่าเหมาะที่สุดในงานหลายๆด้าน ไม่ว่าจะเป็นทางด้านวิทยาศาสตร์ประยุกต์ ทางธุรกิจหรือทางวิศวกรรม ตัวอย่างการนำ TS ไปประยุกต์ใช้งานเช่น [Glover and Laguna, 1980]

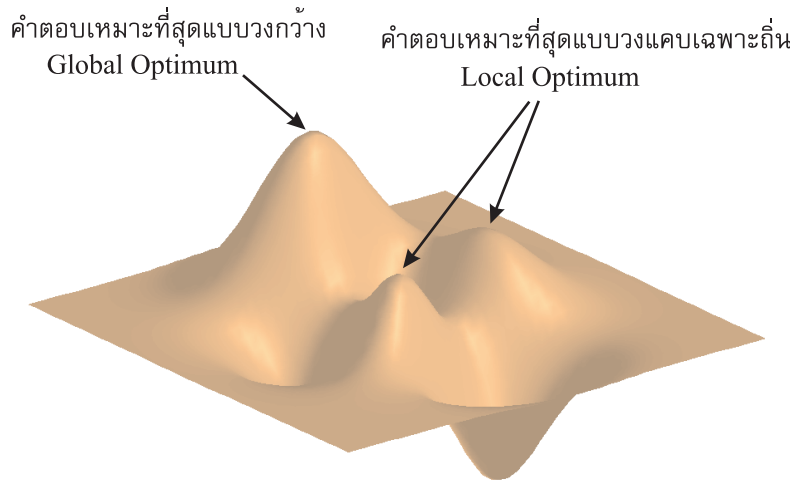
- การจัดกำหนดการ (scheduling)

- การจัดกำหนดการของห้องเรียน (classroom scheduling)
- การจัดกำหนดการของเครื่องจักร (machine scheduling)
- การจัดกำหนดการสายงานของโรงงาน (flow shop scheduling)
- การจัดกำหนดการงานของโรงงาน (job shop scheduling)
- การจัดกำหนดการสายงานเวลาของอุตสาหกรรม (flow-time cell manufacturing)

- การออกแบบ (design)

- การออกแบบใช้คอมพิวเตอร์ช่วยหรือแคด (computer-aided design หรือ CAD)
- การออกแบบเครือข่ายการจราจร (transport network design)

- ปัญหาการตัดชิ้นงานที่ไม่สม่ำเสมอ (irregular cutting problems)
- การวางแผนการใช้พื้นที่ในงานสถาปัตยกรรม (architectural space planning)
- ตรรกศาสตร์และปัญญาประดิษฐ์ (logic & artificial intelligence)
 - การจัดกลุ่ม (clustering)
 - การจดจำและจำแนกรูปแบบ (pattern recognition & classification)
 - การบูรณาการข้อมูล (data integrity)
 - การออกแบบและฝึกสอนเครือข่ายประสาทเทียม (neural network - design & training)
- เทคโนโลยี (technology)
 - การส่งจ่ายกำลังไฟฟ้า (electrical power distribution)
 - การออกแบบโครงสร้างทางวิศวกรรม (engineering structural design)
 - การออกแบบสถานีอวกาศ (space station construction)
 - การจัดวางตำแหน่งวงจร (circuit cell placement)
- โทรคมนาคม (telecommunication)
 - การจัดเส้นทางโทร (call routing)
 - การอัดแบนด์วิดท์ (bandwidth packing)
 - การจัดวางตำแหน่งฮับ (hub facility location)
 - การกำหนดเส้นทาง (path assignment)
- การผลิต สินค้าคงคลังและการลงทุน (production, inventory & investment)
 - การผลิตแบบยืดหยุ่น (flexible manufacturing)
 - การผลิตแบบให้ทันเวลาพอดี (just-in-time production)
 - การเลือกชิ้นส่วน (part selection)
 - การวางแผนสินค้าคงคลังแบบหลายรายการ (multi-item inventory planning)
 - การคิดการลดราคาแบบขายส่ง (volume discount acquisition)
- การจัดเส้นทาง (routing)
 - การจัดเส้นทางพาหนะ (vehicle routing)
 - ปัญหาการเดินทางของเซลส์แมน (traveling salesman problem)
 - ปัญหาการเดินทางของผู้ซื้อ (Traveling purchaser problem)
- การหาค่าเหมาะที่สุดของกราฟ (graph optimization)
 - การแบ่งส่วนกราฟ (graph partitioning)
 - การใส่สีกราฟ (graph coloring)
 - การแบ่งกลุ่ม (clique partitioning)
 - ปัญหาการแบ่งกลุ่มแบบใหญ่สุด (maximum clique problems)



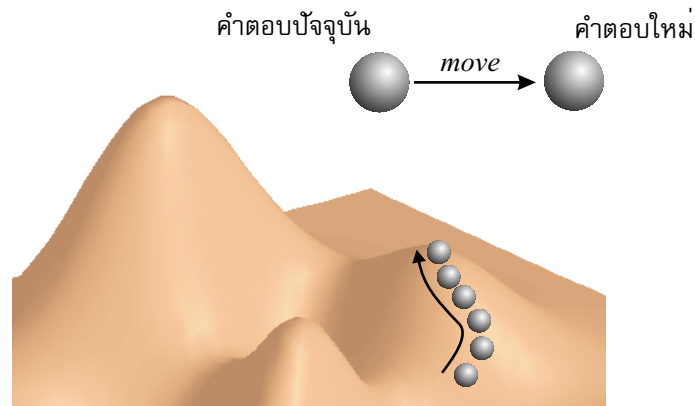
รูปที่ 1.1: ตัวอย่างคำตอบที่เหมาะสมแบบวงแคบเฉพาะถิ่นและแบบวงกว้าง

1.1 แนวคิดพื้นฐานของ TS Tabu Search Concept

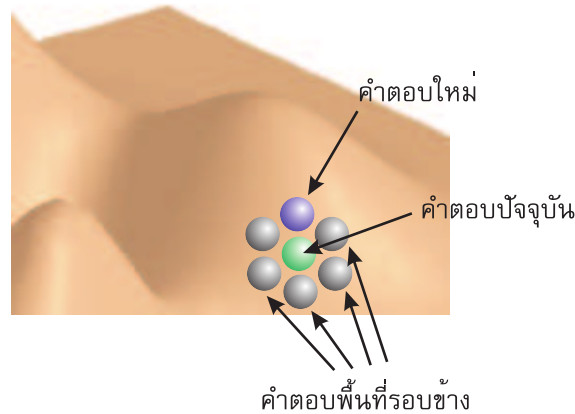
คำว่า ‘*tabu*’ มีความหมายตามพจนานุกรมทั่วไปว่า ‘ต้องห้าม’ ในโครงสร้างของ TS จึงมีองค์ประกอบที่มีสถานะต้องห้ามหรือองค์ประกอบที่ไม่สามารถใช้งานได้อยู่ด้วย อย่างไรก็ตามสถานะขององค์ประกอบดังกล่าวไม่จำเป็นจะต้องคงสภาพเช่นนั้นตลอดไป แต่จะมีการเปลี่ยนแปลงไปตามเวลาและสถานะต่างๆ ภายในระบบ ขั้นตอนการทำงานของ TS ใช้หลักแนวคิดดังกล่าวในการพิจารณาเส้นทางที่คาดว่าจะนำไปสู่คำตอบที่เหมาะสมที่สุด โดยที่การค้นหาค่าจะไม่หยุดอยู่คำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ที่ซึ่งหมายความว่าเส้นทางรอบๆ ในจุดนั้นๆ ไม่สามารถให้คำตอบที่ดีกว่าคำตอบในปัจจุบันอีกแล้ว (ดูตัวอย่างจากฟังก์ชัน $peaks = 3(1 - x)^2 e^{-(x^2 + (y+1)^2)} - 10(\frac{x}{5} - x^3 - y^5) e^{-(x^2 + y^2)} + \frac{1}{3} e^{-((x+1)^2 + y^2)}$ ใน MATLAB® ในรูปที่ 1.1)

TS ใช้หลักแนวคิดที่ทำให้การค้นหาคำตอบเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ เส้นทางต่างๆ ที่อยู่รอบๆ คำตอบในขณะนั้นสามารถให้ข้อมูลที่เป็นประโยชน์ในการตัดสินใจว่าจะเลือกคำตอบใหม่ในเส้นทางใด ซึ่งแตกต่างไปจากการเลือกค้นหาคำตอบด้วยวิธีสุ่ม (random search) ดังนั้นการเลือกเส้นทางค้นหาคำตอบ ซึ่งในบางครั้งไม่ได้ให้คำตอบที่ดีกว่าคำตอบปัจจุบันอาจจะเป็นกลยุทธ์ที่สามารถนำไปสู่การค้นหาคำตอบที่ดีกว่าต่อไปได้ (จุดประสงค์หลักในที่นี้ก็คือการหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นนั่นเอง)

พิจารณาองค์ประกอบพื้นฐานของการค้นหาคำตอบทั่วไปในรูปที่ 1.2 กำหนดให้พื้นผิวในรูปเป็นผลที่ได้จากการประเมิน (objective value) ของฟังก์ชันวัตถุประสงค์ (objective function) จุดหมายในที่นี้คือการหาค่าสูงสุดจากพื้นผิวของฟังก์ชันดังกล่าว (maximization) การคำนวณหาคำตอบใหม่จากจุดของคำตอบปัจจุบันใดๆ จะใช้ปฏิบัติการที่เรียกว่าการ ‘move’ (move operator) หรือ ‘การเดิน’ ซึ่งจะทำให้คำตอบหรือสภาวะปัจจุบันของคำตอบเปลี่ยนแปลงไปตามการเดิน สำหรับการค้นหาคำตอบต่างๆ ไปที่เรียกว่าการค้นหาคำตอบเฉพาะที่ (local search หรือ LS) หรือการค้นหาคำตอบพื้นที่รอบข้าง (neighborhood search หรือ NS) นั้น จะทำการเลือกคำตอบใหม่ที่ดีกว่าคำตอบปัจจุบันจากคำตอบที่มีอยู่รอบๆ โดยอาศัยการเดินเพื่อทำการประเมินค่าของคำตอบรอบๆ ข้างเหล่านั้น แล้วเลือกคำตอบที่ดีที่สุดขึ้นมาเป็นคำตอบใหม่ต่อไป ดังแสดงในรูปที่ 1.3 จากรูปจะเห็นว่า คำตอบใหม่มีค่าการประเมินที่สูงที่สุดในบรรดาคำตอบเพื่อนบ้านทั้งหมด การค้นหาคำตอบด้วยวิธีนี้บางครั้งจะเรียกว่าวิธีการไต่เขา (hill climbing) กำหนดให้ขบวนการค้นหาคำตอบเป็นการหาค่าที่เหมาะสมที่สุด (ในกรณีตัวอย่างนี้ให้เป็นการหาค่าเหมาะสมที่มากที่สุดหรือการทำ maximization) ของฟังก์ชันวัตถุประสงค์ $f(s)$ และ $s \in S$ โดยที่ $f(s)$ อาจจะเป็นฟังก์ชันเชิงเส้นหรือไม่เป็นเชิงเส้นใดๆ s เป็นคำตอบในปัจจุบันและ S เป็นเซตคำตอบที่เป็นไปได้ของระบบ



รูปที่ 1.2: ปฏิบัติการ 'move' หรือการเดิน เพื่อหาคำตอบใหม่จากคำตอบปัจจุบัน



รูปที่ 1.3: การค้นหาคำตอบเฉพาะที่ คำตอบใหม่ที่ได้ได้จากคำตอบพื้นที่รอบข้างที่ดีที่สุด

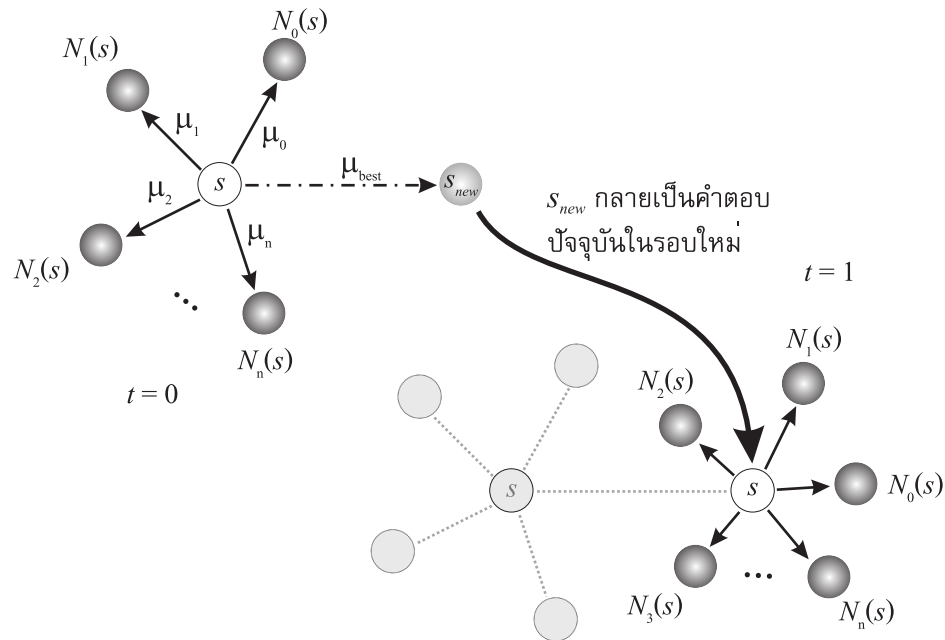
(นั่นคือ S เป็นพื้นที่ค้นหาหรือ search space) สำหรับ s แต่ละค่าซึ่งเป็นคำตอบในตอนนั้นจะมีคำตอบรอบข้างหรือ neighborhood คือ $N \subset S$ การกำหนดคำตอบรอบข้างนั้นจะขึ้นอยู่กับลักษณะของปัญหาและการเดิน

ถ้ากำหนดให้ M คือเซตของปฏิบัติการการเดินทั้งหมดในระบบ μ คือปฏิบัติการการเดิน (หรือฟังก์ชัน) เราสามารถเขียนความสัมพันธ์ระหว่างคำตอบรอบข้างกับปฏิบัติการการเดินได้ดังนี้

$$N_i(s) = \{\mu_i(s) \in N, \mu_i \in M\} \quad (1.1)$$

โดยที่ $i = 1, \dots, n$ ปฏิบัติการการเดินนั้นจะขึ้นอยู่กับโจทย์ปัญหาของระบบ ดังนั้นจึงสามารถมีปฏิบัติการการเดินที่แตกต่างกันออกไปแล้วแต่ระบบ ในกรณีนี้การเดินทำให้สถานะของระบบเปลี่ยนแปลงไปในทางที่ดีขึ้นนั่นคือมีคำตอบที่เข้าใกล้ค่าเหมาะที่สุด เราสามารถนิยามคำตอบที่เหมาะสมที่สุดได้ดังนี้

- ถ้า s เป็นคำตอบที่เหมาะสมที่สุดแบบวงกว้าง (global optimum) แล้ว จะได้ว่า
 - $f(s) \geq f(y)$ สำหรับ $\forall y \in S$ (กรณีหาค่ามากที่สุดหรือ maximization)
 - $f(s) \leq f(y)$ สำหรับ $\forall y \in S$ (กรณีหาค่าน้อยที่สุดหรือ minimization)
- ถ้า s เป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (local optimum) แล้ว จะได้ว่า
 - $f(s) \geq f(y)$ สำหรับ $\forall y \in N(x)$ (กรณีหาค่ามากที่สุดหรือ maximization)



รูปที่ 1.4: การค้นหาคำตอบพื้นที่รอบข้าง จากคำตอบปัจจุบัน s ด้วยปฏิบัติการการเดิน μ_i ไปยังคำตอบพื้นที่รอบข้าง N_i ($i = 0 \dots n$) กล่าวคือ $N_i(s)$ คือคำตอบใหม่ที่ได้จากการใช้ปฏิบัติการการเดิน μ_i กับคำตอบปัจจุบัน s เส้นประในรูปแสดงคำตอบพื้นที่รอบข้างที่ให้ค่าวัตถุประสงค์ดีที่สุดเป็น s_{new} และเรียกการเดินจาก s ไปยัง s_{new} นี้เป็น μ_{best} คำตอบใหม่ s_{new} นี้จะถูกใช้เป็น s ในรอบการเดินถัดไป ($t = 1$)

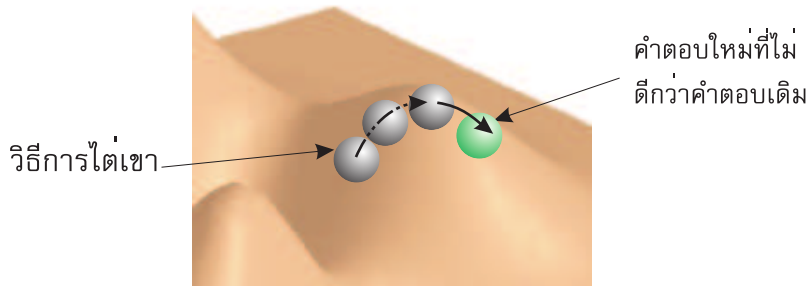
- $f(s) \leq f(y)$ สำหรับ $\forall y \in N(x)$ (กรณีหาค่าน้อยที่สุดหรือ minimization)

ขั้นตอนทั่วไปของการค้นหาคำตอบพื้นที่รอบข้างสามารถสรุปได้ดังนี้ (ดูรูปที่ 1.4)

▷ การค้นหาคำตอบพื้นที่รอบข้าง - Neighborhood Search

1. เลือก $s_0 \in S$ เป็นคำตอบเริ่มต้นของระบบ s_0 อาจจะมาจากการสุ่ม (random) กำหนดให้คำตอบปัจจุบันที่มีอยู่นี้เป็นคำตอบที่เหมาะสมที่สุด กล่าวคือ $s = s_0$
2. คำนวณหา $s_{new} \in N(s)$ โดยที่ $f(s_{new}) > f(s)$
3. ถ้าไม่สามารถหา s_{new} ในข้อ 2. ได้ แสดงว่า s เป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ให้ยุติการค้นหาได้
4. ให้ $s = s_{new}$ แล้วเริ่มข้อ 2. ใหม่

ข้อด้อยที่เห็นได้ชัดของวิธีการค้นหาคำตอบแบบนี้ก็คือ ไม่สามารถหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ เมื่อคำตอบที่ได้เป็นคำตอบที่ดีที่สุดในคำตอบรอบข้างแล้ว จะไม่มีการเดินใดๆ ที่ทำให้ได้คำตอบใหม่ที่ดีกว่าคำตอบปัจจุบันนี้อีกแล้ว อย่างไรก็ตามโครงสร้างการค้นหาคำตอบแบบนี้ก็ยังถูกใช้ในวิธีการค้นหาคำตอบแบบอื่นๆ ที่มีประสิทธิภาพที่ดีกว่า ยกตัวอย่างเช่นการค้นหาคำตอบที่เหมาะสมที่สุดโดยใช้อัลกอริทึมลงชันสุด (steepest descent) ที่ออกแบบให้ทิศทางการค้นหาคำตอบเป็นไปอย่างเหมาะสมที่สุดในเทอมของเกรเดียนต์ (gradient) ถึงแม้วิธีดังกล่าวจะไม่สามารถแก้ปัญหาของคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ก็ตาม สำหรับ



รูปที่ 1.5: การหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น เมื่อระบบทำการค้นหาคำตอบด้วยวิธีการไต่เขาไปตามเส้นประ ที่ซึ่งเลือกคำตอบใหม่ที่ต้องดีกว่าคำตอบเดิมเสมอ (เปรียบกับการไต่ขึ้นยอดเขา จะต้องมุ่งไปในทิศทางที่สูงขึ้นเท่านั้น) คำตอบที่ดีที่สุดที่ได้บนยอดเขาจะเป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น การหลีกเลี่ยงคำตอบดังกล่าวจำเป็นจะต้องเลือกเส้นทางค้นหาคำตอบใหม่ซึ่งไม่ดีกว่าคำตอบเดิม เมื่อเทียบกับคำตอบพื้นที่รอบข้าง

TS แล้ว คุณสมบัติในการหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นเป็นจุดเด่นที่ทำให้ TS กลายเป็นเครื่องมือในการค้นหาคำตอบที่มีประสิทธิภาพสูง

1.2 องค์ประกอบพื้นฐานของการค้นหาคำตอบแบบตาบ Fundamental TS Structures

TS เพิ่มเดิมขั้นตอนและเงื่อนไขในการเดินที่นอกเหนือไปจากการค้นหาคำตอบพื้นที่รอบข้าง โดยมีจุดประสงค์หลักคือ

- หลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (local optimum avoidance)
คำตอบที่ดีที่สุดในกลุ่มคำตอบพื้นที่รอบข้างจะมีค่าการประเมินที่สูงที่สุด ถ้าคำตอบใหม่มีค่าการประเมินที่ดีกว่าคำตอบปัจจุบัน การค้นหาคำตอบจะทำการเดินไปยังทิศทางนั้น และเมื่อใดที่ค่าการประเมินของคำตอบพื้นที่รอบข้างไม่ได้ดีกว่าคำตอบปัจจุบัน การค้นหาคำตอบที่ดีกว่าจะสิ้นสุดลง ซึ่งในกรณีนี้เป็นขั้นตอนที่เกิดขึ้นในการค้นหาคำตอบพื้นที่รอบข้าง คำตอบที่ได้ในขณะนั้นจะเป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นในทางตรงข้าม TS ยอมให้มีการเดินไปยังคำตอบใหม่มีค่าการประเมินที่ **ไม่** ดีกว่าคำตอบปัจจุบัน จุดประสงค์สำคัญก็เพื่อให้สามารถหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้นั่นเอง (ดูรูปที่ 1.5)
- หลีกเลี่ยงเส้นทางการค้นหาคำตอบที่ทำให้เกิดการวนรอบอยู่กับที่ (cycle avoidance)
เส้นทางบางเส้นทางสามารถนำไปสู่การวนรอบอยู่กับที่ ทำให้ไม่สามารถหลุดออกไปจากสถานะปัจจุบันได้ (ดังนั้นไม่สามารถเปลี่ยนเส้นทางที่นำไปสู่คำตอบที่ดีกว่าได้) ตัวอย่างของเส้นทางดังกล่าวได้แก่การเดินย้อนกลับ (inverse move) ซึ่งในบางโอกาสอาจจะทำให้การค้นหาคำตอบเกิดการเดินไปกลับโดยไม่มีที่สิ้นสุดได้ ดังนั้นใน TS จึงมีการตั้งค่าสถานะของการเดินที่ **เพิ่งถูกใช้** ให้เป็นสถานะต้องห้าม (tabu) เพื่อไม่ให้ใช้การเดินนี้อีกภายในระยะเวลาที่กำหนด (นั่นหมายความว่าถ้าเวลาผ่านไปภายในระยะเวลาที่กำหนดไว้แล้วสถานะของการเดินนั้นจะถูกยกเลิกการเป็นสถานะต้องห้าม)

สถานะต้องห้ามหรือสถานะตาบ - เซตของการเดินใดๆ ที่ถูกตั้งสถานะเป็นสถานะต้องห้ามจะไม่อนุญาตให้ถูกใช้ในการค้นหาคำตอบได้

ด้วยความสามารถใหม่ที่กล่าวมาข้างต้น จะเห็นได้ว่า TS ได้มีการใช้ข้อมูลของการค้นหาคำตอบในอดีตมาช่วยตัดสินใจว่าควรจะไปทิศทางใด องค์ประกอบใหม่ซึ่งเป็นส่วนสำคัญในโครงสร้างของ TS ที่ทำให้การค้นหาคำตอบมีประสิทธิภาพสูงขึ้นนี้ได้แก่

1. เงื่อนไขของความคงอยู่ล่าสุด (recency condition)

การใช้เงื่อนไขของความคงอยู่นี้เป็นการติดตามการค้นหาคำตอบในช่วงเวลาที่ผ่านมา เมื่อคำตอบหนึ่งถูกค้นพบแล้ว การเดินที่นำไปสู่คำตอบนั้นจะถูกตั้งเป็นสถานะต้องห้าม คำตอบที่ถูกค้นพบด้วยการเดินนี้จะถือเป็นคำตอบล่าสุดและจะไม่ถูกค้นอีกภายในระยะเวลาหนึ่ง (เนื่องจากการเดินที่นำไปสู่คำตอบนี้ถูกห้ามใช้อีกในขณะที่ยังมีสถานะต้องห้ามอยู่) หลังจากระยะเวลาที่กำหนดผ่านไป สถานะการเดินนั้นก็จะถูกตั้งค่ากลับสู่สถานะปกติ ดังนั้นแล้วภายหลังการเดินไปยังคำตอบหนึ่งๆ TS จะบังคับให้การค้นหาคำตอบทำการเดินไปยังคำตอบใหม่ โดยที่คำตอบเดิมจะไม่ถูกค้นอีก กลไกนี้ทำให้ TS สามารถหลุดออกจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น และทำการค้นหาคำตอบที่ดีขึ้นไปเรื่อยๆ ได้ (ถึงแม้ว่าในบางครั้ง คำตอบใหม่จะไม่ดีกว่าคำตอบที่มีอยู่ก็ตาม)

2. เงื่อนไขของความซ้ำซาก (frequency condition)

ในลักษณะเดียวกันเราสามารถบันทึกจำนวนครั้งที่การเดินหนึ่งๆ ถูกเรียกใช้ได้ TS ถือว่าถ้ารูปแบบการเดินใดถูกเรียกใช้เป็นจำนวนมากครั้งเกินไป (เกินจำนวนที่ตั้งเอาไว้) การเดินนั้นควรจะถูกต้องห้ามหรือถูกตั้งเป็นสถานะต้องห้าม เพื่อหลีกเลี่ยงเส้นทางการค้นหาคำตอบที่ทำให้เกิดการวนรอบอยู่กับที่ ทำให้สามารถหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้

เงื่อนไขทั้งสองจะถูกใช้ร่วมกันเสมอ เนื่องจากเงื่อนไขเพียงอย่างเดียวอย่างหนึ่งไม่เพียงพอ เราสามารถกล่าวได้ว่าเงื่อนไขทั้งสองเป็นส่วนเติมเต็ม หรือ complimentary ซึ่งกันและกัน กล่าวคือการเดินใดถูกตั้งค่าให้มีสถานะต้องห้ามด้วยเงื่อนไขของความซ้ำซาก และได้คงสถานะต้องห้ามนานเกินระยะเวลาที่กำหนดไว้ การเดินนั้นจะสามารถถูกตั้งค่ากลับสู่สถานะปกติได้ด้วยเงื่อนไขของความคงอยู่ล่าสุด

TS ยังมีอีก 2 องค์ประกอบอื่นๆอันเป็นกลไกสำคัญที่ทำให้การค้นหาคำตอบมีประสิทธิภาพเพิ่มขึ้นได้แก่ กลไกการเน้น (intensification) และกลไกการแปรเปลี่ยน (diversification) กลยุทธ์ทั้งสองอย่างมีผลต่อการเลือกเส้นทางในการค้นหาคำตอบ

- **กลไกการเน้น** คือการค้นหาคำตอบที่เน้นไปยังกลุ่มคำตอบที่ได้ค้นพบแล้วว่าเป็นคำตอบที่ดี โดยใช้ข้อมูลที่ได้นั้นที่จากการค้นหาคำตอบที่ผ่านมาในอดีต กลยุทธ์นี้ทำให้ TS กลับไปค้นหาคำตอบในย่านที่เคยเจอคำตอบที่ดี และทำการค้นหาในย่านนั้นอย่างละเอียดขึ้น ส่วนประกอบของคำตอบที่น่าจะเป็นประโยชน์หรือเส้นทางที่นำไปสู่คำตอบนั้น จะถูกใช้เป็นข้อมูลโดยกลไกการเน้นในการค้นหาคำตอบใหม่ได้
- **กลไกการแปรเปลี่ยน** ในทางตรงกันข้ามเป็นกลยุทธ์ที่ส่งเสริมให้ TS ไปทำการสำรวจย่านที่ยังไม่เคยถูกสำรวจมาก่อน ซึ่งอาจจะทำให้ได้คำตอบที่มีความแตกต่างไปจากกลุ่มคำตอบที่ได้ถูกสำรวจมาก่อนที่จะทำกลยุทธ์นี้ ในบางครั้งการเลือกเส้นทางอื่นที่ยังไม่เคยสำรวจและแตกต่างไปจากแนวทางของเส้นทางเดิมอาจจะทำให้มีโอกาสเจอคำตอบที่ดีกว่าได้เช่นกัน
- **เกณฑ์ความทะเยอทะยาน** นอกไปจากองค์ประกอบต่างๆที่ได้กล่าวมาแล้วข้างต้น TS ยังมีองค์ประกอบอื่นที่เป็นส่วนสำคัญในการออกแบบการค้นหาแบบตามนั้นก็คือ เกณฑ์ความทะเยอทะยาน (aspiration criteria) ซึ่งเป็นเงื่อนไขที่สามารถทำให้เกิดการเดินไปในทิศทางที่ต้องการ ถึงแม้ว่าการเดินนั้นจะมีสถานะต้องห้าม โดยที่การเดินดังกล่าวจะถูกอนุญาตก็ต่อเมื่อคำตอบที่ได้ดีกว่าทุกคำตอบที่เคยค้นพบมา (TS จะต้องทำการเก็บบันทึกผลการค้นหาคำตอบที่ดีที่สุดเอาไว้เพื่อใช้ในการตรวจสอบเงื่อนไข)

1.3 รายละเอียดโครงสร้างของ TS

สิ่งสำคัญที่ทำให้ TS แตกต่างไปจากการค้นหาคำตอบวิธีอื่นก็คือรายการต้องห้าม (tabu list) ซึ่งใช้เป็นตัวข้อมูลสำหรับเก็บบันทึกข้อมูลต่างๆในการค้นหาคำตอบ ไม่ว่าจะเป็นสถานะต้องห้ามของแต่ละการเดินหรือคุณสมบัติอื่นๆของ TS เช่น เงื่อนไขความซ้ำซากหรือเงื่อนไขความคงอยู่ล่าสุด ดังนั้นการออกแบบรายการต้องห้ามจึงมีความสำคัญอย่างยิ่ง เพราะคุณสมบัติของรายการต้องห้ามเช่น ขนาด หรือช่วงเวลาในการคงความเป็นสถานะต้องห้าม ฯลฯ จะมีผลโดยตรงต่อประสิทธิภาพในการค้นหาคำตอบของ TS

กำหนดให้เซตของการเดิน M ประกอบไปด้วย

1. เซตของการเดินที่ต้องห้าม T (เป็นตาบู่) และ
2. เซตของการเดินที่อนุญาต A (ไม่เป็นตาบู่)

ดังนั้น ณ เวลา t ใดๆ เราจะได้ว่า

$$M^{(t)} = A^{(t)} \cup T^{(t)} \quad (1.2)$$

และ

$$A^{(t)} \cap T^{(t)} = \{\phi\} \quad (1.3)$$

(สัญลักษณ์ด้วย $x^{(t)}$ หมายถึงให้พิจารณา x ณ เวลา t) จากความสัมพันธ์ (1.2) จะได้ว่าเซตการเดินทั้งหมดที่เป็นไปได้ประกอบไปด้วยเซตการเดินต้องห้ามและเซตการเดินที่อนุญาต และเราสามารถบอกจากความสัมพันธ์ (1.3) ได้ว่าไม่มีการเดินใดที่มีสถานะทั้งสองอย่างได้ในเวลาเดียวกัน การค้นหาคำตอบเริ่มต้นจาก $s^{(0)}$ วิธีที่ง่ายที่สุดในการเลือกคำตอบเริ่มต้นนี้ก็คือการสุ่ม การเดินทุกอย่างจะได้รับอนุญาตที่เวลาเริ่มต้นนี้ นั่นคือ $A^{(0)} = M$ และนั่นหมายความว่ายังไม่มีมีการเดินใดๆที่ต้องห้ามหรือ $T^{(0)} = \{\phi\}$ ที่เวลา t ใดๆ จะสามารถคำนวณหาคำตอบใหม่จากคำตอบปัจจุบัน $s^{(t)}$ ได้ด้วยการเดิน μ ซึ่งมาจากเซตของ $A^{(t)}$ (เซตการเดินที่ไม่เป็นตาบู่)

$$s^{(t+1)} = \mu(s^{(t)}) \quad (1.4)$$

โดยที่ $s^{(t+1)}$ เป็นคำตอบใหม่ที่ได้ในรอบการเดินถัดไปหรือที่เวลา $t+1$ และ μ เป็นฟังก์ชันหรือปฏิบัติการการเดินที่ทำให้ค่าประเมินของ $s^{(t)}$ มีค่าที่ดีที่สุดเปรียบเทียบกับคำตอบอื่นๆ ที่ไม่เป็นตาบู่ที่เวลา t โดยสามารถแสดงด้วยความสัมพันธ์ดังนี้

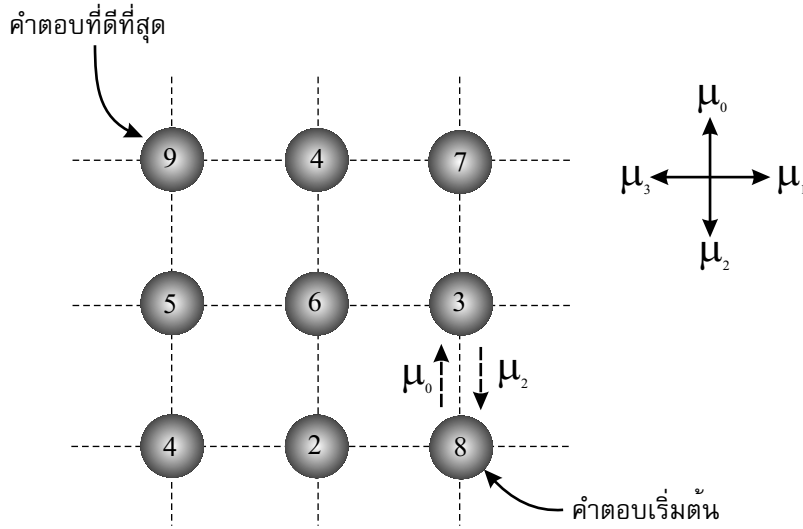
$$\bar{\mu} = \{\mu^* \in A^{(t)} : f(\mu^*(s^{(t)})) \geq f(\mu(s^{(t)})), \forall \mu \in A^{(t)}\} \quad (1.5)$$

ในความสัมพันธ์ข้างต้น เครื่องหมาย \geq จะกลายเป็น \leq เมื่อเป็นปัญหาของการหาค่าต่ำที่สุด ในกรณีที่มีการเดินมากกว่าหนึ่งที่ทำให้ค่าประเมินที่เท่ากัน TS จะทำการสุ่มเลือกใช้เดินตัวใดตัวหนึ่งเพียงตัวเดียวเท่านั้น

มาถึงจุดนี้แล้ว โครงสร้างของ TS ยังไม่มีความแตกต่างใดๆ เปรียบเทียบกับการค้นหาแบบพื้นที่รอบข้าง (neighborhood search) ดังที่ได้กล่าวมาแล้วในข้างต้น TS จะมีความแตกต่างกับวิธีการค้นหาวิธีอื่นตรงที่ มีการกำหนดให้การเดินมีสถานะเป็นตาบู่หรือสถานะต้องห้าม เราจะพิจารณาตัวอย่างกฎเกณฑ์ในการกำหนดสถานะดังกล่าวที่เป็นโครงสร้างหลักของ TS ดังต่อไปนี้

1.3.1 การกำหนดสถานะของการเดินย้อนกลับ Inverse Move

การกำหนดสถานะของการเดินย้อนกลับ ให้เป็นสถานะต้องห้ามมีความจำเป็นที่สามารถทำให้ TS สามารถหลุดพ้นจากคำตอบที่เหมาะสมแบบวงแคบเฉพาะถิ่นได้ ในบางโอกาสการกำหนดสถานะของการเดินนี้ทำให้ TS มีความสามารถในการแก้ปัญหาของการวนรอบอยู่กับที่ของคำตอบได้อีกด้วย



รูปที่ 1.6: การวนรอบอยู่กับที่กับการเดินย้อนกลับ

พิจารณาตัวอย่างต่อไปนี ถ้ากำหนดให้คำตอบปัจจุบันคือ $s^{(t)}$ เป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นแล้ว คำตอบอื่นๆรอบๆ $s^{(t)}$ ที่ได้จากการเดินจะต้องมีค่าการประเมินที่น้อยกว่า $s^{(t)}$ นั่นคือ

$$f(s^{(t+1)}) = f(\mu^{(t)}(s^{(t)})) < f(s^{(t)}) \quad (1.6)$$

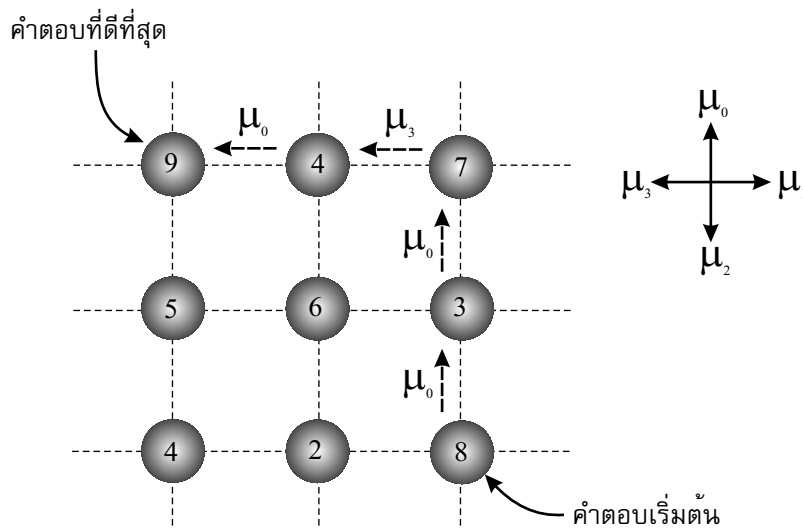
เมื่อพิจารณาการเดินถัดไปจากคำตอบปัจจุบัน $s^{(t+1)}$ ซึ่งจะต้องทำการเดินไปยังคำตอบที่ดีกว่า เราจะได้ชัดเจนว่ามีโอกาสที่จะเกิดการเดินกลับไปยัง $s^{(t)}$ เนื่องจาก $s^{(t)}$ เป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นในย่านนี้ การเดินนี้จะเป็นการเดินย้อนกลับ $\mu^{(t+1)} = \mu^{(t)-1}$ ดังนั้นจะได้ว่า

$$s^{(t+2)} = \mu^{(t+1)}(s^{(t+1)}) = \mu^{(t)-1}(\mu^{(t)}(s^{(t)})) = s^{(t)} \quad (1.7)$$

พิจารณาตัวอย่างในรูปที่ 1.6 ถ้ากำหนดให้การเดินมีทั้งหมด 4 ทิศทางโดยเทียบกับตัวเองแล้ว เริ่มจากคำตอบที่มีค่าการประเมินเท่ากับ 8 ($s^{(0)} = 8$) ซึ่งเป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (เนื่องจากทั้ง 2 และ 3 มีค่าน้อยกว่า 8)

- ที่เวลา $t = 1$ การค้นหาคำตอบจะทำการเดินด้วย $\mu_0^{(0)}$ ซึ่งจะได้คำตอบใหม่ที่ 3 (เพราะ 3 มีค่าการประเมินดีกว่า 2) ดังนั้น $s^{(1)} = \mu_0^{(0)}(s^{(0)}) = 3$
- ที่เวลา $t = 2$ การค้นหาคำตอบจะต้องพยายามทำการเดินไปยังคำตอบที่ดีกว่า ในขณะที่คำตอบที่ได้จาก μ_0 และ μ_3 ไม่สามารถให้คำตอบที่ดีกว่า 8 ได้ ดังนั้นการค้นหาคำตอบจะเลือกการเดิน μ_2 จะได้ว่า $s^{(2)} = \mu_2^{(1)}(s^{(1)}) = 8$ เราสามารถสังเกตได้ว่า μ_2 นั้นเป็นการเดินย้อนกลับซึ่งกันและกันกับ μ_0 หรือสามารถเขียนความสัมพันธ์ได้เป็น $\mu_2 = \mu_0^{-1}$ และ $\mu_0 = \mu_2^{-1}$

การเดินและการเดินย้อนกลับนี้ทำให้การค้นหาคำตอบ เกิดการวนรอบติดอยู่ที่คำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น การวนรอบนี้สามารถถูกแก้ไขได้ ถ้าการเดินย้อนกลับถูกกำหนดให้มีสถานะต้องห้ามภายในระยะเวลาหนึ่ง (ซึ่งควรจะเพียงพอที่จะทำให้การค้นหาคำตอบนั้นอยู่ห่างไปจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ไม่เช่นนั้นแล้วโอกาสที่การเดินจะนำไปสู่คำตอบเดิมก็ยังคงมีอยู่) เมื่อเวลาผ่านไปตามที่กำหนดก็จะทำการยกเลิกสถานะต้องห้ามของการเดินนั้น ช่วงเวลาของสถานะต้องห้ามนั้นต้องไม่นานเกินไปจนทำให้การเดินทั้งหมดที่มีอยู่กลายเป็นสถานะต้องห้าม ซึ่งทำให้ขบวนการค้นหาคำตอบหยุดชะงักไป พิจารณาตัวอย่างในรูปที่ 1.7 ซึ่งเป็นตัวอย่างเดียวกันกับในรูปที่ 1.6 ในครั้งนี้จะมีการกำหนดให้การเดินย้อนกลับมีสถานะต้องห้ามและคงอยู่ภายในระยะเวลา 1 รอบของการเดิน

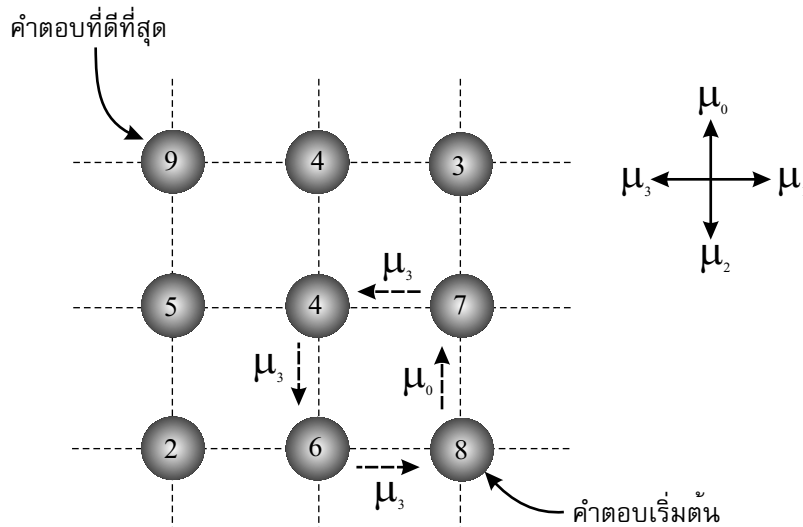


รูปที่ 1.7: การหลุดพ้นคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นด้วยการกำหนดสถานะตาบู่

- ที่เวลา $t = 0$ ทุกการเดินจะมีสถานะอนุญาต นั่นคือ $A^{(0)} = \{\mu_0, \mu_1, \mu_2, \mu_3\}$ และ $T^{(0)} = \{\phi\}$
- ที่เวลา $t = 1$ จะเกิดการเดินด้วย $\mu_0^{(0)}$ ทำให้การค้นหาคำตอบหลุดพ้นจากคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ในขณะนั้นการเดินย้อนกลับของ μ_0 ซึ่งก็คือ μ_2 จะมีสถานะต้องห้าม ดังนั้นจะได้ว่า $T^{(1)} = \{\mu_2\}$ และ $A^{(1)} = \{\mu_0, \mu_1, \mu_3\}$
- ที่เวลา $t = 2$ เนื่องจาก μ_2 มีสถานะต้องห้าม จึงเป็นการกั้นไม่ให้การค้นหาคำตอบย้อนกลับไปที่ตำแหน่งเริ่มต้นอีก ดังนั้นการเดินที่สามารถใช้ได้จาก A ที่มีค่าการประเมินที่ดีที่สุดคือ μ_0 ซึ่งให้ค่าการประเมินเท่ากับ 7 จะเห็นได้ว่าขณะนี้การค้นหาคำตอบสามารถหลุดออกจากคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้แล้ว และขณะนี้คำตอบเป็นคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นซึ่งมีค่าไม่ดีกว่าคำตอบเดิม (น้อยกว่า 8) อย่างไรก็ตาม กลไกของการใช้สถานะต้องห้ามนี้ ทำให้การค้นหาคำตอบสามารถหลุดออกจากคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ เมื่อมีการเดินใหม่เกิดขึ้น μ_2 จะถูกรีเซตค่า แต่เนื่องจากการเดินยังคงเป็น μ_0 ดังนั้น การเดินย้อนกลับของ μ_0 ซึ่งก็คือ μ_2 ก็จะถูกตั้งค่าสถานะต้องห้ามอีกครั้ง ทั้ง $A^{(2)}$ และ $T^{(2)}$ ยังคงมีค่าเดียวกับที่เวลา $t = 1$
- ที่เวลา $t = 3$ มีการเดินด้วย μ_3 ไปยังคำตอบ 4 ดังนั้นการเดินย้อนกลับของ μ_3 ซึ่งก็คือ μ_1 จะมีสถานะต้องห้าม ดังนั้นจะได้ว่า $A^{(3)} = \{\mu_0, \mu_2, \mu_3\}$ และ $T^{(3)} = \{\mu_1\}$ จะเห็นได้ว่าขณะนี้ μ_2 ได้พ้นสถานะของการถูกต้องห้ามเรียบร้อยแล้ว
- ที่เวลา $t = 4$ การค้นหาคำตอบได้เจอคำตอบที่ดีที่สุด

โดยปกติแล้วขบวนการการค้นหาคำตอบควรจะสิ้นสุดลง ณ เวลานั้น อย่างไรก็ตามเราก็คงยังไม่สามารถบอกได้ว่า คำตอบที่มีในปัจจุบันนี้เป็นค่าที่ดีที่สุดแล้วหรือยัง ดังนั้น TS ก็จะทำการค้นหาคำตอบใหม่ไปเรื่อยๆ โดยมีการบันทึกคำตอบที่ดีที่สุดในอดีตเอาไว้ เมื่อคำตอบใหม่ที่ได้ตรงตามเงื่อนไขที่ต้องการหรือจำนวนรอบในการคำนวณถึงค่าที่ตั้งเอาไว้ การค้นหาคำตอบก็จะสิ้นสุดลงได้

ระยะเวลาที่กำหนดให้การเดินคงสถานะต้องห้ามไว้คือขนาดของรายการต้องห้าม (tabu list size) ซึ่งเป็นพารามิเตอร์ที่สำคัญอย่างหนึ่งของ TS ตัวอย่างข้างต้นมีการกำหนดสถานะต้องห้ามเท่ากับหนึ่งรอบการคำนวณ ซึ่งหมายความว่าในการเดินถัดไป สถานะต้องห้ามใดๆที่มีอยู่ก่อนหน้านี้ก็จะถูกยกเลิกไป เราจะได้เห็นว่าระยะเวลาดังกล่าวจะต้องสัมพันธ์กับหลายๆองค์ประกอบด้วย เช่น สำหรับตัวอย่างข้างต้นถ้ากำหนดให้ขนาดของรายการต้องห้ามมีค่ามากกว่าหรือเท่ากับ 4 แล้ว เมื่อเวลาผ่านไประยะหนึ่ง ทุกการเดินอาจจะไม่มีสถานะต้องห้ามก็เป็นได้



รูปที่ 1.8: การวนรอบอยู่กับที่

(เพราะในตัวอย่างนี้มีการเดินอยู่เพียง 4 ทิศทาง) อย่างไรก็ตามขนาดของรายการต้องห้ามเองยังมีผลโดยตรงต่อการหลุดพ้นจากการวนรอบอยู่กับที่ ซึ่งสามารถพิจารณาจากตัวอย่างได้ในรูปที่ 1.8 จากตัวอย่างข้างต้นจะเห็นได้ว่า ถึงแม้จะมีการกำหนดให้การเดินย้อนกลับมีสถานะต้องห้าม แต่การค้นหาคำตอบได้เกิดการวนรอบด้วยการเดิน μ_3 ไปเรื่อยๆ (การเดินครั้งแรกใช้การเดิน μ_0) ทำให้ไม่สามารถหลุดออกจากลูปดังกล่าวได้ สำหรับปัญหาในกรณีนี้ การใช้คุณสมบัติอีกอย่างหนึ่งของ TS ที่เรียกว่าเงื่อนไขของความซ้ำซากหรือ "frequency" สามารถใช้แก้ไขปัญหาลักษณะนี้ได้

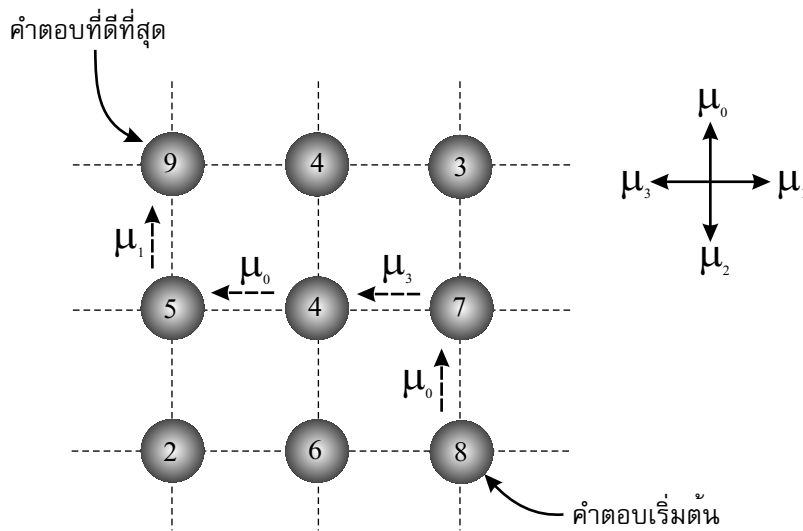
1.3.2 การกำหนดสถานะของการเดินที่มีความซ้ำซาก

ในบางกรณีการค้นหาคำตอบเกิดการวนรอบอยู่กับที่ ที่ซึ่งโดยลำพังของการกำหนดสถานะต้องห้ามของการเดินย้อนกลับนั้น ไม่เพียงพอที่จะทำให้หลุดพ้นออกจากคำตอบที่เหมาะสมแบบวงแคบเฉพาะถิ่นได้ การนำเงื่อนไขของความซ้ำซากเข้ามาใช้ในการกำหนดสถานะต้องห้ามของการเดิน จึงสามารถช่วยเพิ่มประสิทธิภาพในการค้นหาคำตอบได้

ในการใช้เงื่อนไขของความซ้ำซากเข้ามาช่วยในการค้นหาคำตอบ จะพิจารณาว่าชุดการเดินใดที่ถูกใช้บ่อยจนเกินไป สถานะนั้นๆ จะถูกกำหนดสถานะให้เป็นต้องห้ามภายในระยะเวลาหนึ่ง จำนวนครั้งในการเรียกใช้ชุดการเดินก่อนที่จะถูกห้ามจะขึ้นอยู่กับแต่ละปัญหา เช่นเดียวกันกับระยะเวลาในการคงความเป็นสถานะต้องห้ามของการเดิน ความหมายทั่วไปของการใช้เงื่อนไขของความซ้ำซากก็คือ “เมื่อเดินตามเส้นทางรูปแบบหนึ่งไปซักระยะแล้ว ควรจะต้องมีการเปลี่ยนไปใช้เส้นทางรูปแบบอื่นบ้าง อาจจะทำให้พบคำตอบที่ดีกว่าในแนวทางที่แตกต่างออกไปได้” ยกตัวอย่างการใช้เงื่อนไขของความซ้ำซากจากตัวอย่างข้างต้นได้ดังนี้

กำหนดให้จำนวนครั้งของการเดินซ้ำก่อนที่จะถูกต้องห้ามมีค่าเท่ากับ 1 รอบ ดังนั้นการเดินใดๆ จะไม่สามารถถูกเรียกใช้ติดต่อกันมากกว่า 1 ครั้งได้ พิจารณาในรูปที่ 1.9

- ที่เวลา $t = 0$ การเดิน μ_0 ให้คำตอบที่ดีที่สุดคือ 7 การเดินย้อนกลับ μ_2 จะมีสถานะต้องห้าม ในขณะเดียวกันจำนวนครั้งในการเดิน μ_0 มีค่าเท่ากับเงื่อนไขของความซ้ำซาก การเดิน μ_0 จึงมีสถานะต้องห้ามเช่นกัน ดังนั้นขณะ $\mathbf{A}^{(0)} = \{\mu_1, \mu_3\}$ และ $\mathbf{T}^{(0)} = \{\mu_0, \mu_2\}$
- ที่เวลา $t = 1$ การเดิน μ_3 ซึ่งได้มีสถานะอนุญาตถูกเรียกใช้ ดังนั้นการเดิน μ_3 และการเดินย้อนกลับของ μ_3 ซึ่งก็คือ μ_1 จึงมีสถานะต้องห้าม ในขณะที่สถานะต้องห้ามของการเดิน μ_0 และ μ_2 ถูกยกเลิกไป สถานะ



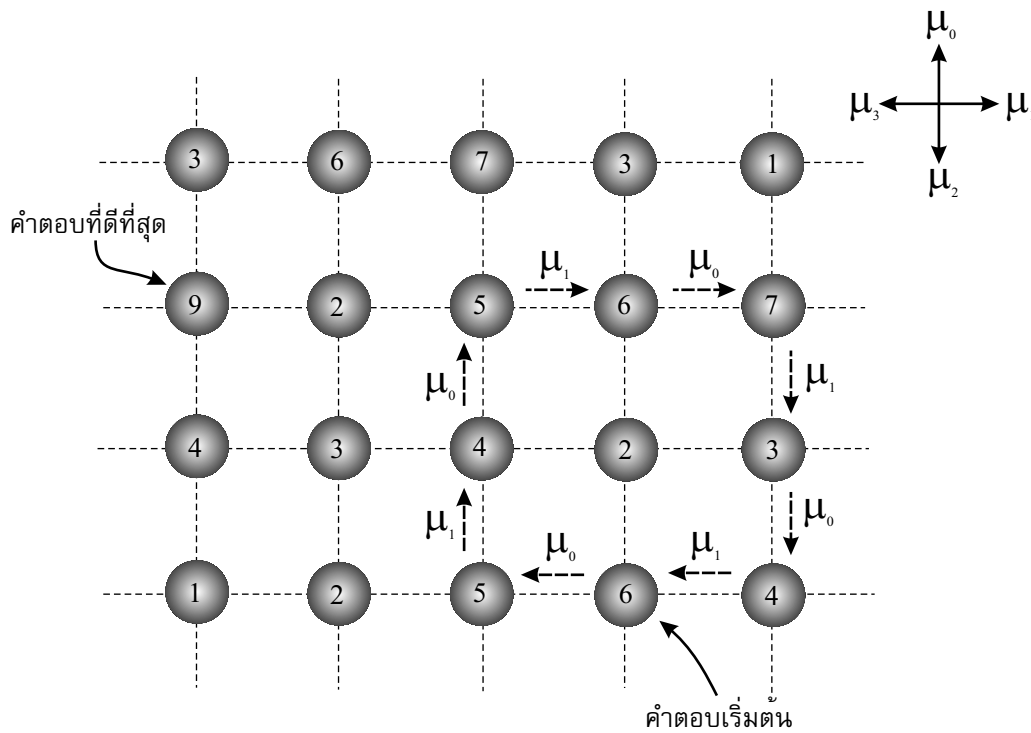
รูปที่ 1.9: เงื่อนไขความซ้ำซากกับการกำหนดสถานะต้องห้าม

โดยรวมของระบบขณะนี้คือ $A^1 = \{\mu_0, \mu_1\}$ และ $T^{(1)} = \{\mu_1, \mu_3\}$

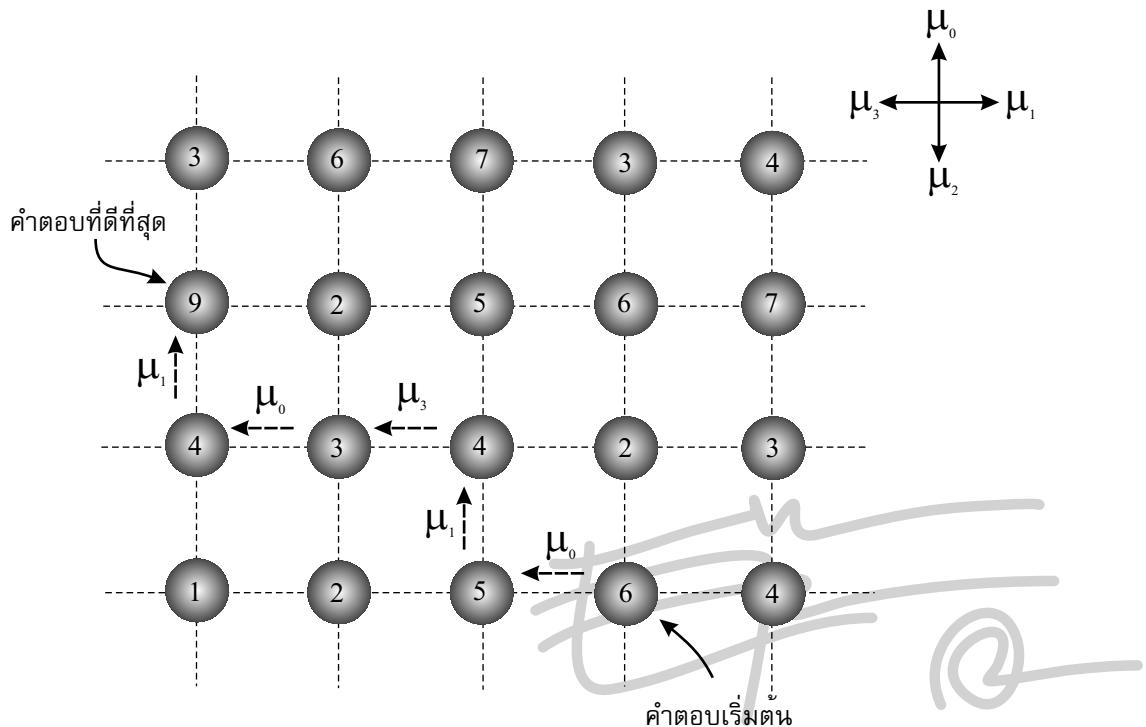
- ที่เวลา $t = 2$ ถึงแม้ว่าคำตอบที่มีค่าเท่ากับ 6 จะเป็นคำตอบที่ดีกว่าคำตอบอื่นๆก็ตาม แต่เนื่องจาก μ_3 มีสถานะต้องห้าม ดังนั้นการเดินที่สามารถเรียกใช้ได้จึงเป็น μ_0 เท่านั้น คำตอบที่ถูกเลือกจึงเป็นคำตอบที่มีค่าเท่ากับ 5 สถานะโดยรวมของการเดินคือ $A^{(2)} = \{\mu_1, \mu_3\}$ และ $T^{(2)} = \{\mu_0, \mu_3\}$
- ที่เวลา $t = 3$ เนื่องจากคำตอบที่ดีกว่าได้มาจากการเดิน μ_1 ซึ่งมีสถานะอนุญาต ดังนั้นการค้นหาคำตอบจึงเจอคำตอบที่ดีที่สุด

จากตัวอย่างดังกล่าว จะเห็นได้ว่าการใช้เงื่อนไขของความซ้ำซากช่วยแก้ปัญหาของการวนรอบอยู่กับที่ได้เป็นอย่างดี อย่างไรก็ตาม มีปัญหาย่อยมากมายที่ค่าของเงื่อนไขความซ้ำซากเพียง 1 รอบไม่สามารถแก้ไขได้ พิจารณารูปที่ 1.10 เป็นตัวอย่าง จากตัวอย่างข้างต้น ถึงแม้ว่าจะมีการใช้เงื่อนไขของความซ้ำซาก แต่ปัญหาของการวนรอบอยู่กับที่ก็ยังคงมีอยู่ โดยที่ขนาดของวงรอบมีขนาด 2 ก้าวของการเดิน ปัญหาที่เกิดขึ้นนี้ไม่ใช่ว่าการใช้เงื่อนไขของความซ้ำซากไม่ได้ผล แต่เนื่องจากช่วงเวลาของความคงอยู่ของสถานะต้องห้ามที่สั้นเกินไป สังเกตรูปแบบของการเดินที่เกิดขึ้นคือ $\mu_0 \rightarrow \mu_1 \rightarrow \mu_0 \rightarrow \mu_1 \rightarrow \dots$ พิจารณาถ่วงลดใช้ความคงอยู่ของสถานะต้องห้ามที่มีขนาดเท่ากับ 2 ก้าวของการเดินดังรูปที่ 1.11

- ที่เวลา $t = 0$ การเดิน μ_0 ถูกเรียกใช้ ดังนั้น μ_0 (และ μ_2) จะมีสถานะต้องห้าม
- ที่เวลา $t = 1$ เนื่องจากการเดิน μ_0 ถูกห้ามใช้ ดังนั้นจึงเลือกเดินด้วย μ_1 (แม้ μ_0 จะไม่ถูกห้าม แต่คำตอบ 4 ก็ยังเป็นคำตอบที่ดีกว่า 2) ในขณะนี้สถานะของ μ_0 ก็ยังคงต้องห้ามเนื่องจาก μ_0 จะต้องรอการเดิน 2 รอบเพื่อจะให้พ้นจากสถานะต้องห้าม ในขณะเดียวกัน μ_1 ซึ่งเป็นการเดินล่าสุดก็จะมีสถานะต้องห้ามเช่นกัน คำตอบในขณะนี้คือ 5
- ที่เวลา $t = 2$ เนื่องจากเหลือเพียงการเดิน μ_3 ดังนั้นจึงต้องเดินไปยังคำตอบ 3 การเดิน μ_3 มีการเปลี่ยนสถานะเป็นต้องห้าม ในขณะที่สถานะต้องห้ามของ μ_0 ถูกยกเลิกอันเนื่องมาจากค่าความคงอยู่ของสถานะต้องห้ามได้หมดลง (หลังจาก 2 รอบ)
- ที่เวลา $t = 3$ การเดิน μ_0 ถูกเลือกใช้อีกครั้งและถูกเซตสถานะเป็นต้องห้าม ในขณะที่การเดิน μ_1 ถูกยกเลิกสถานะต้องห้าม



รูปที่ 1.10: การวนรอบอยู่กับที่ที่มีขนาดของการวนรอบใหญ่กว่า 1 ก้าวของการเดิน



รูปที่ 1.11: แก้ปัญหาการวนรอบอยู่กับที่โดยใช้ค่าความคงอยู่ของความเป็นตามูที่นานขึ้น

จากตัวอย่างข้างต้น จะเห็นได้ว่าปัญหาของการวนรอบอยู่กับที่ได้ถูกแก้ไขด้วยค่าความคงอยู่ของสถานะต้องห้ามที่มากขึ้น เราจะสังเกตเห็นว่า ในแต่ละช่วงเวลา ในรายการตามหรือรายการต้องห้าม (tabu list) จะมีอยู่อย่างน้อย 2 การเดินเสมอที่มีสถานะต้องห้ามอยู่ นั่นคือขนาดของรายการต้องห้ามหรือ tabu list size จะมีความสัมพันธ์กับค่าความคงอยู่ของสถานะต้องห้ามนั่นเอง สังเกตรูปแบบการเดินที่เกิดขึ้นในกรณีนี้คือ $\mu_0 \rightarrow \mu_1 \rightarrow \mu_3 \rightarrow \mu_0 \rightarrow \mu_1 \rightarrow \dots$ ซึ่งแตกต่างไปจากกรณีในช่วงเวลาของความคงอยู่ของสถานะต้องห้ามที่มีค่าเท่ากับ 1 ก้าว จะเห็นได้ว่าในกรณีนี้ไม่มีการเดินใดที่เกิดขึ้นติดต่อกันภายในช่วงเวลา 2 ก้าว

โดยปกติแล้วช่วงเวลาของความคงอยู่ของสถานะต้องห้ามควรจะใหญ่เพียงพอที่จะทำให้หลุดพ้นจากการวนรอบอยู่กับที่ แต่ก็ควรจะไม่ใช่ใหญ่เกินไปจนทำให้เกิดการจำกัดของการเดินได้ ยกตัวอย่างเช่น ถ้าการเดินมีทั้งหมด 4 วิธีดังตัวอย่างและค่าความคงอยู่ของสถานะต้องห้ามมีค่ามากกว่าหรือเท่ากับ 4 ในช่วงเวลาหนึ่งจะจะไม่มีการเดินใดเลยที่ไม่มีสถานะต้องห้าม ขบวนการค้นหาคำตอบก็จะหยุดชะงักลงได้

การออกแบบขนาดของรายการต้องห้ามให้มีความเหมาะสมนั้นขึ้นอยู่กับชนิดและองค์ประกอบของปัญหาแบบต่างๆ การกำหนดขนาดของรายการต้องห้ามไว้ค่าหนึ่ง อาจจะเพียงพอในบางสถานการณ์ที่ปัญหาของการวนรอบอยู่กับที่มีขนาดที่ใหญ่ขึ้น (ซึ่งเป็นไปได้ยากที่จะสามารถคำนวณหรือคาดคะเนล่วงหน้าว่าจะมีขนาดเท่าใด) ดังนั้นจึงได้มีการปรับปรุง TS ให้มีประสิทธิภาพมากขึ้น อันนำไปสู่ TS ชนิดต่างๆซึ่งมีอยู่หลากหลายแบบ Reactive TS หรือ RTS [Battiti and Tecchiolli, 1994] เป็นตัวอย่างหนึ่งที่มีโครงสร้างของขนาดรายการต้องห้าม ที่เปลี่ยนแปลงไปตามสภาวะและสถานการณ์หลายๆอย่างของระบบ

สำหรับตัวอย่างนี้ที่มีจำนวนการเดินเพียง 4 แบบนั้น ถือว่ามีจำนวนน้อยเกินไป ทำให้เกิดความไม่สมจริงในแง่ของการใช้งานจริงๆ อย่างไรก็ตามตัวอย่างดังกล่าวก็สามารถแสดงให้เห็นภาพโดยรวมในการทำงานของ TS ได้เป็นอย่างดี

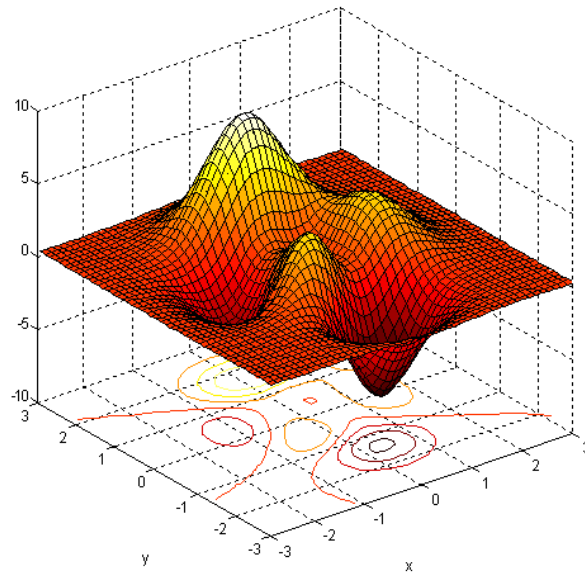
■ ตัวอย่างที่ 1.1 การออกแบบโครงสร้างของ TS ในการค้นหาคำตอบที่เหมาะสมที่สุดแบบวงกว้าง

พิจารณาตัวอย่างการค้นหาคำตอบที่ดีที่สุดแบบวงกว้างที่เป็นค่าสูงสุดของฟังก์ชัน peaks (ดูรูปที่ 1.12)

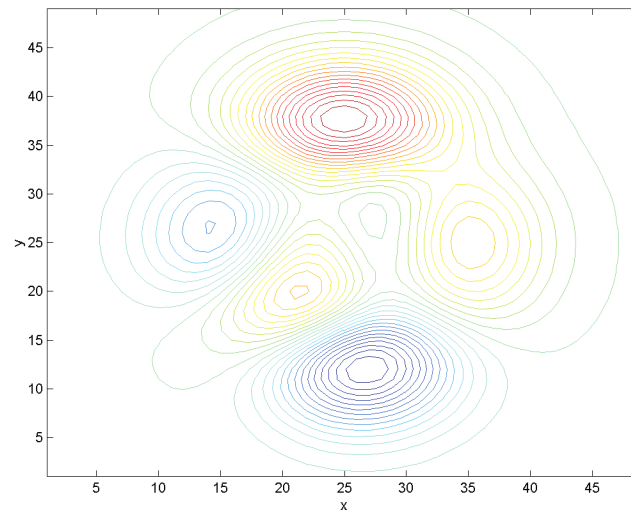
$$\text{peaks} = 3(1-x)^2 e^{-(x^2+(y+1)^2)} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-(x^2+y^2)} + \frac{1}{3} e^{-((x+1)^2+y^2)} \quad (1.8)$$

ฟังก์ชัน peaks มีความเหมาะสมในการแสดงประสิทธิภาพของ TS เนื่องจากมีทั้งคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (local optimum) และแบบวงกว้าง (global optimum) อย่างไรก็ตามมีพื้นผิวของฟังก์ชันที่เป็นมาตรฐานในการใช้ทดสอบอัลกอริทึมในการค้นหา สาเหตุที่เลือกใช้ฟังก์ชันนี้ก็เพราะเป็นฟังก์ชันที่มีพื้นผิวที่ไม่ซับซ้อนทำให้การแสดงผลได้เข้าใจง่าย การค้นหาคำตอบในปัญหานี้มีขอบเขตการค้นหาคือตำแหน่ง (x, y) ทั้งระนาบพื้นผิวของฟังก์ชัน ดังนั้นคำตอบที่เป็นเป้าหมายคือตำแหน่ง (x, y) ที่ให้ค่าสูงสุดของฟังก์ชัน สำหรับการประเมินค่าของคำตอบที่ถูกค้นหา ฟังก์ชัน $z = f(x, y) = \text{peaks}(x, y)$ ข้างต้นจะถูกใช้เป็นฟังก์ชันวัตถุประสงค์สำหรับการประเมินค่าของคำตอบที่ตำแหน่ง (x, y) นั้นๆ ในกรณีการค้นหาคำตอบที่เป็นค่าที่มากที่สุด คำตอบที่มีค่าการประเมินจากฟังก์ชันวัตถุประสงค์ที่สูงกว่าจะถือเป็นคำตอบที่ดีกว่า

สำหรับปัญหานี้การเดินคือการเปลี่ยนแปลงตำแหน่ง (x, y) ซึ่งเป็นคำตอบปัจจุบันไปยังตำแหน่งต่างๆ รอบๆ ตำแหน่ง (x, y) นั้น ในกรณีที่มีตัวแปรอยู่ 2 ตัวคือ x และ y เราจะเลือกใช้การเดินในพื้นที่รอบข้างแบบ 8 ทิศ (8-neighborhood) ซึ่งเป็นที่เพียงพอสำหรับปัญหานี้ ดังนั้นเราสามารถนิยามการเดินได้ทั้งหมด 8 แบบดังแสดงในรูปที่ 1.14 ถ้ากำหนดให้ s เป็นคำตอบที่กำลังถูกประเมินค่า โดยที่ $s = (x, y)$ ถ้าให้ M เป็นเซตของการเดินที่เป็นไปได้ทั้งหมดแล้ว จะได้ว่า $M = \{\mu_0, \dots, \mu_8\}$ และเราสามารถนิยามการเดินที่เปลี่ยนตำแหน่งของคำตอบจาก s



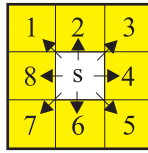
รูปที่ 1.12: ฟังก์ชัน peaks สำหรับทดสอบหาค่าสูงสุดของพื้นผิว จาก MATLAB® Toolbox



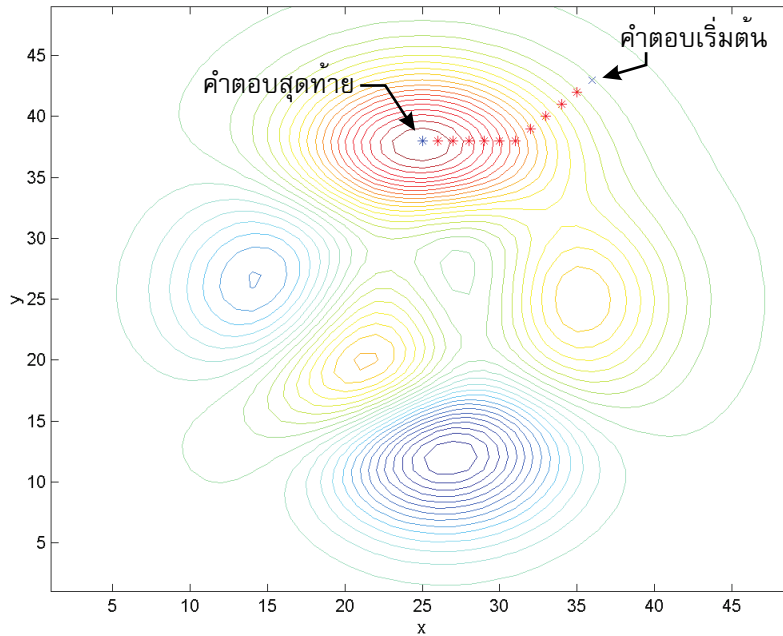
รูปที่ 1.13: เส้นแสดงพื้นที่สูงต่ำของฟังก์ชัน peaks ในรูปที่ 1.12

ไปรอบๆ ภายในพื้นที่เพื่อนบ้านได้ดังนี้

$$\begin{aligned}
 s_0 &= \mu_0(s) = s + (0, 0) \\
 s_1 &= \mu_1(s) = s + (-1, -1) \\
 s_2 &= \mu_2(s) = s + (0, -1) \\
 s_3 &= \mu_3(s) = s + (1, -1) \\
 s_4 &= \mu_4(s) = s + (1, 0) \\
 s_5 &= \mu_5(s) = s + (1, 1) \\
 s_6 &= \mu_6(s) = s + (0, 1) \\
 s_7 &= \mu_7(s) = s + (-1, 1) \\
 s_8 &= \mu_8(s) = s + (-1, 0)
 \end{aligned} \tag{1.9}$$



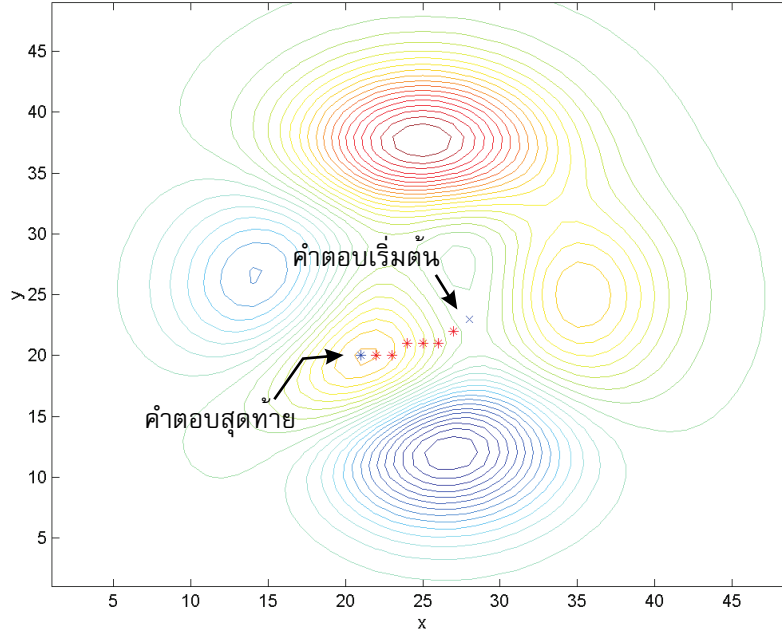
รูปที่ 1.14: การเดินในพื้นที่เพื่อนบ้าน 8 ที่รอบคำตอบปัจจุบัน s



รูปที่ 1.15: การค้นหาค่าตอบแบบ NS ที่ได้คำตอบที่เหมาะสมที่สุดแบบวงกว้าง

สังเกตว่า μ_0 ไม่ทำให้เกิดการเปลี่ยนแปลงของคำตอบ (อยู่กับที่) ดังนั้นปัญหานี้ก็คือการค้นหาคำตอบที่เป็นตำแหน่งต่าง ๆ บนพื้นผิวของฟังก์ชัน peaks เพื่อค้นหาคำตอบสูงสุดของฟังก์ชัน ตัวอย่างการใช้การค้นหาคำตอบที่เหมาะสมที่สุดโดยการค้นหาคำตอบพื้นที่รอบข้าง (neighborhood search หรือ NS) แสดงในรูปที่ 1.15 และ 1.16 (เครื่องหมาย \times และ $*$ (เข็ม) แสดงคำตอบเริ่มต้นและคำตอบสุดท้ายตามลำดับ เครื่องหมาย $*$ (อ่อน) แสดงคำตอบที่ได้ระหว่างการค้นหาคำตอบ ในรูปแสดงเส้นโครงร่างหรือ contour ของพื้นผิวฟังก์ชัน peaks ในระนาบ 2 มิติ) จากการค้นหาคำตอบแบบ NS จะเห็นได้ว่าคำตอบสุดท้ายจะขึ้นอยู่กับข้อกำหนดคำตอบเริ่มต้น ในรูปที่ 1.15 คำตอบเริ่มต้นที่ตำแหน่งที่ใกล้เคียงกับคำตอบที่เหมาะสมที่สุดแบบวงกว้าง ดังนั้นการเดินค้นหาคำตอบจึงมีโอกาในการเจอคำตอบที่เหมาะสมที่สุดแบบวงกว้าง ในขณะที่คำตอบในรูปที่ 1.16 เป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ซึ่งจะเห็นได้ชัดเจนว่าเป็นเพราะคำตอบเริ่มต้น อยู่ใกล้กับคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นมากกว่าคำตอบที่เหมาะสมที่สุดแบบวงกว้าง เนื่องมาจากการค้นหาคำตอบแบบ NS ไม่มีกลไกในการหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น ดังนั้นการค้นหาคำตอบจะสิ้นสุดที่คำตอบเฉพาะที่นั้น เราจะทำการเพิ่มกลไกของ TS เข้าไปใน NS เพื่อให้สามารถแก้ไขปัญหาดังกล่าวได้ดังรายละเอียดต่อไปนี้

- **รายการต้องห้าม (tabu list)** - มีหน้าที่บันทึกข้อมูลการเดินในแต่ละรอบของการค้นหาคำตอบ ขนาดของรายการต้องห้ามนั้นโดยปกติแล้วจะขึ้นอยู่กับหลายๆองค์ประกอบ สำหรับตัวอย่างนี้มีการเดินที่แตกต่างกันอยู่ 8 แบบ (ไม่รวมการเดินอยู่กับที่) ดังนั้นขนาดของรายการต้องห้ามควรจะมีขนาดไม่เกิน 6 (หรือ $L - 2$)



รูปที่ 1.16: การค้นหาคำตอบแบบ NS ที่ได้คำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น

เมื่อจำนวนรูปแบบในการเดินทั้งหมดคือ L) สาเหตุที่จำเป็นจะต้องออกแบบให้รายการต้องห้ามมีขนาดเล็กกว่าจำนวนรูปแบบในการเดินทั้งหมดที่ใช้ได้ เนื่องจากจะทำให้การค้นหาคำตอบไม่หยุดชะงักเมื่อเกิดกรณีที่ทุกการเดินมีสถานะต้องห้าม (อย่างน้อยยังมีเหลืออีกการเดินอีก 2 ทิศทางที่ยังไม่ถูกห้ามเดิน)

- การเดินย้อนกลับ (inverse move) - เราสามารถนิยามการเดินย้อนกลับได้ดังนี้

$$\begin{aligned}
 \mu_1^{-1}(s) &= s + (1, 1) = s_5 \\
 \mu_2^{-1}(s) &= s + (0, 1) = s_6 \\
 \mu_3^{-1}(s) &= s + (-1, 1) = s_7 \\
 \mu_4^{-1}(s) &= s + (-1, 0) = s_8 \\
 \mu_5^{-1}(s) &= s + (-1, -1) = s_1 \\
 \mu_6^{-1}(s) &= s + (0, -1) = s_2 \\
 \mu_7^{-1}(s) &= s + (1, -1) = s_3 \\
 \mu_8^{-1}(s) &= s + (1, 0) = s_4
 \end{aligned} \tag{1.10}$$

เราสามารถกล่าวเป็นคำพูดทั่วไปสำหรับการเดินย้อนกลับและการกำหนดค่าให้เป็นสถานะต้องห้ามได้คือ “เมื่อเกิดการเดินขึ้นในทิศทางใดทิศทางหนึ่งแล้ว ไม่ควรจะต้องเดินย้อนกลับไปในทิศทางเดิมอีก เพราะดูเหมือนว่าคำตอบของทางเดินที่ผ่านมาจะไม่ใช้คำตอบที่ต้องการ” ดังนั้นทุกครั้งที่มีการเดินใดๆ เกิดขึ้น การเดินย้อนกลับนั้นจะถูกตั้งสถานะเป็นต้องห้ามในการเดินครั้งถัดไป สังเกตว่าการเดินและการเดินย้อนกลับในกรณีนี้มีความสัมพันธ์กันคือ $\mu_n^{-1} = -\mu_n$ ส่วนระยะเวลาของความคงอยู่ของสถานะต้องห้ามจะแตกต่างกันไปตามการลักษณะของปัญหาต่างๆ นั่นคือ

- ระยะเวลาความคงอยู่ของสถานะต้องห้าม (recency) - เป็นคุณสมบัติที่สำคัญอย่างหนึ่งที่มีผลต่อทิศทางการหาคำตอบของระบบ เมื่อการเดินใดๆ ถูกกำหนดให้มีสถานะต้องห้ามแล้ว ภายในระยะเวลาที่กำหนดไว้ค่าหนึ่ง สถานะต้องห้ามของการเดินนั้นจะถูกยกเลิกไป ดังที่เคยกล่าวไปแล้วข้างต้น ในบางเงื่อนไข (ซึ่งขึ้น

อยู่กับการออกแบบระบบที่แตกต่างกัน) ระยะเวลาความคงอยู่ของสถานะต้องห้ามนี้ จะมีความสัมพันธ์โดยตรงกับขนาดของรายการต้องห้าม ดังนั้นระยะเวลาความคงอยู่ของสถานะต้องห้ามนี้ ควรจะนานเพียงพอที่จะทำให้การค้นหาคำตอบหลุดพ้นจากการวนรอบอยู่กับที่ได้ สำหรับตัวอย่างนี้การใช้เงื่อนไขความคงอยู่ของสถานะต้องห้าม จะถูกใช้คู่กับเงื่อนไขความซ้ำซาก ซึ่งโดยปกติแล้วเงื่อนไขทั้งสองจะถูกใช้ร่วมกันเสมอ เพราะเงื่อนไขอย่างใดอย่างหนึ่งอาจจะไม่เพียงพอให้ TS สามารถแก้ไขปัญหาได้

- **จำนวนครั้งสำหรับเงื่อนไขของความซ้ำซาก (frequency)** - เงื่อนไขนี้จะกำหนดจำนวนครั้งสูงสุดของการเดินรูปแบบหนึ่งๆ ไม่ให้ถูกเรียกใช้บ่อยเกินไปอันจะก่อให้เกิดการวนรอบอยู่กับที่ได้ เมื่อการเดินนั้นๆ ถูกเรียกใช้จนเกิดจำนวนครั้งที่กำหนดไว้ การเดินนั้นก็จะมิสสถานะต้องห้ามอยู่ จนกระทั่งระยะเวลาความคงอยู่ของสถานะต้องห้ามนั้นหมดไป

ในการออกแบบ TS นั้นจะต้องมีเงื่อนไขอย่างน้อย 2 อย่างที่คอยทำหน้าที่ตรงข้ามกันและสัมพันธ์ซึ่งกันและกัน ยกตัวอย่างเช่นเมื่อมีการออกแบบเงื่อนไขที่ทำให้การเดินมีสถานะต้องห้ามแล้ว (การเดินย้อนกลับและเงื่อนไขของความซ้ำซาก) ก็จำเป็นจะต้องออกแบบเงื่อนไขที่สามารถยกเลิกสถานะต้องห้ามของการเดินนั้นได้ (เงื่อนไขความคงอยู่ของสถานะต้องห้าม) โครงร่างโปรแกรมการทำงานโดยรวมของ TS มีดังนี้

1. กำหนดค่าคงที่ของระบบ

- จำนวนรอบสูงสุดที่จะทำการค้นหาคำตอบ $= t_{max}$
- ระยะเวลาความคงอยู่ของสถานะต้องห้าม $= t_{tabu}$
- จำนวนครั้งสูงสุดของการเดินที่ซ้ำซาก $= f_{max}$

2. กำหนดค่าเริ่มต้นตัวแปรของระบบ

- เซตการเดินทั้งหมด $M = \{\mu_0, \dots, \mu_n\}$
- เซตการเดินที่มีสถานะต้องห้าม $T = \{\phi\}$
- เซตของเวลาล่าสุดของการเดินที่มีสถานะต้องห้าม $\Lambda = \{\phi\}$
- เซตของจำนวนครั้งการเดินที่ซ้ำซาก $\Gamma = \{\phi\}$
- เซตการเดินที่อนุญาต $A = M$
- ตัวแปรนับรอบ $t = 0$

3. เลือกคำตอบเริ่มต้นของระบบ (สุ่มภายในย่านของคำตอบที่เป็นไปได้ของระบบ) $s = s_0$

เริ่มต้นการวนรอบ

4. เลือกการเดินที่ดีที่สุดที่ไม่มีสถานะต้องห้ามโดยตัดสินจากค่าประเมินที่ดีที่สุดจากพื้นที่เพื่อนบ้านของ s_0

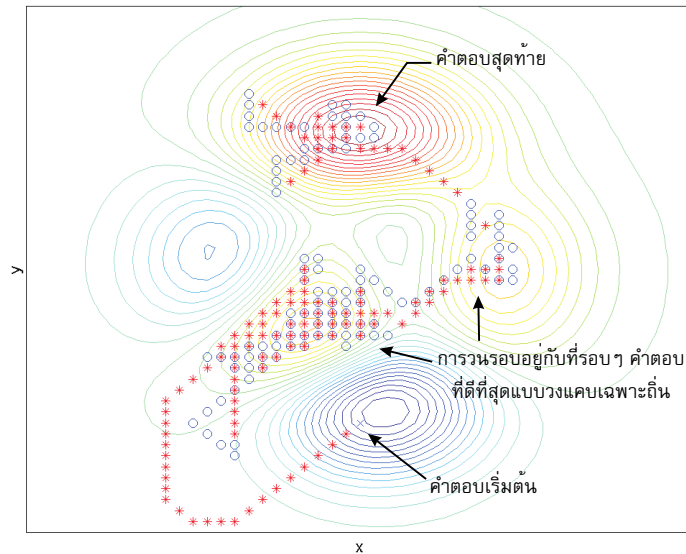
- $s_i = \{\mu_i(s)\}$ โดยที่ $\mu_i \in A$ และ $i = 1, \dots, n$
- s_{best} คือคำตอบที่มีค่าการประเมินของ $f(s_i)$ ที่ดีที่สุดและ $s_{best} = \mu_{best}(s)$

5. ตรวจสอบใน T ว่ามีการเดินใดที่จะพ้นสถานะต้องห้าม

- $T = T - \{\mu\}$ ถ้า $\Lambda(\mu) - t > t_{tabu}$ สำหรับ $\forall \mu \in T$

6. ตรวจสอบความซ้ำซากของการเดินปัจจุบัน

- ถ้า $\Gamma(\mu_{best}) + 1 > f_{max}$ แล้ว $T = T + \{\mu_{best}\}$ และบันทึกเวลา t ของ μ_{best} ใน Λ



รูปที่ 1.17: ผลการค้นหาคำตอบด้วย TS สังเกตการวนรอบอยู่กับที่รอบๆ คำตอบที่ดีที่สุดแบบวงแคบเฉพาะถิ่น เกิดขึ้นบริเวณที่เป็นยอดเขาของพื้นผิว แต่ไม่ใช่ยอดเขาที่สูงที่สุด

7. การเดินย้อนกลับมีสถานะต้องห้าม

- $\mathbf{T} = \mathbf{T} + \{\mu_{best}^{-1}\}$
- บันทึกเวลา t ของ μ_{best}^{-1} ใน Λ

8. บันทึกค่าที่เปลี่ยนแปลงใหม่ในเซตการเดินที่อนุญาต

- $\mathbf{A} = \mathbf{M} - \mathbf{T}$

9. $s = s_{best}$

10. $t = t + 1$

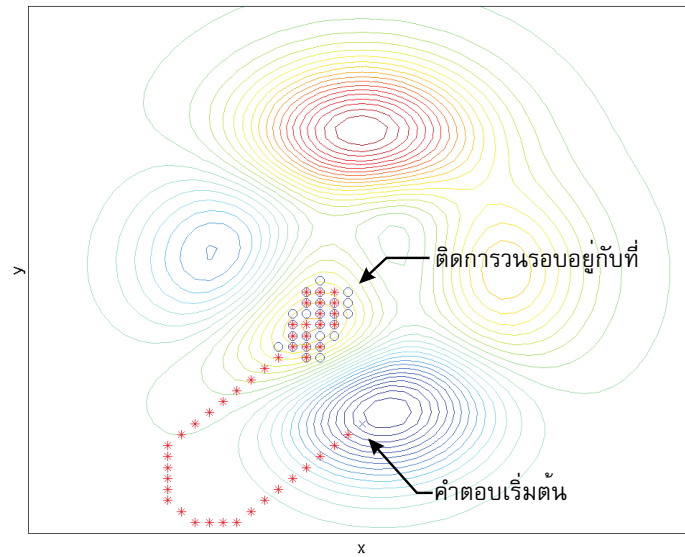
11. ถ้า $t > t_{max}$ หรือ $f(s_{best})$ ให้ค่าประเมินตามเป้าหมายแล้ว

- ให้ยุติการการค้นหา
- รายงานผลการค้นหาคำตอบ
- มิฉะนั้นแล้ว ให้กลับไปเริ่มต้นที่ 4. อีกครั้ง

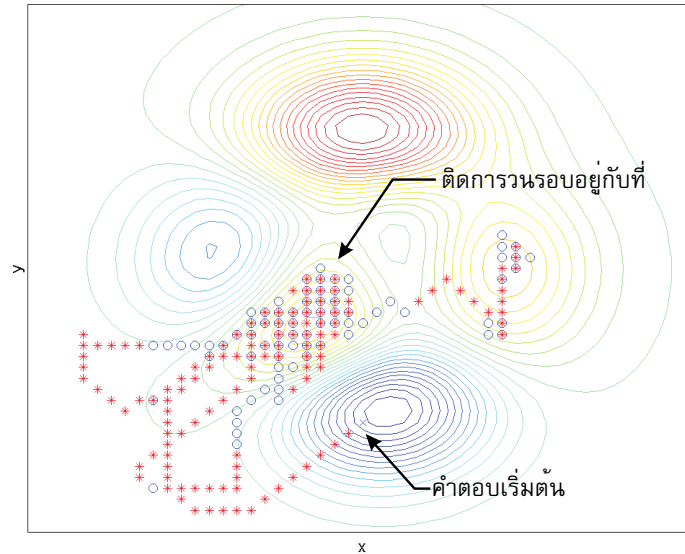
ผลการค้นหาคำตอบด้วย TS มีแสดงในรูปที่ 1.17 โดยมีรายละเอียดพารามิเตอร์ของ TS ดังนี้

- $t_{max} = 550$
- $t_{tabu} = 24$
- $f_{max} = 8$

สัญลักษณ์ * (อ่อน) แสดงการเดินธรรมดา ในขณะที่สัญลักษณ์ o (เข้ม) แสดงการเดินที่มีค่าการประเมินที่ไม่ดีกว่าคำตอบปัจจุบัน ผลการค้นหาคำตอบจะเห็นได้ว่า TS สามารถหลุดพ้นจากคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ และสุดท้ายก็ได้พบเจอคำตอบที่ดีที่สุด (โดยการผ่านคำตอบเหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นทุกคำตอบ)



รูปที่ 1.18: TS ที่มี $t_{tabu} = 12$



รูปที่ 1.19: TS ที่มี $f_{max} = 7$

พิจารณากำหนดค่าพารามิเตอร์ต่างๆของระบบอันได้แก่ t_{tabu} และ f_{max} นั้นได้มาจากการทดลองสุ่ม ในรูปที่ 1.18 แสดงผลการค้นหาคำตอบสำหรับ $t_{tabu} = 24$ ในขณะที่ผลการทดลองในรูปที่ 1.19 มีค่า f_{max} เท่ากับ 7 ซึ่งสามารถเห็นได้ชัดเจนว่าผลการค้นหาคำตอบไม่ได้เจอคำตอบที่เหมาะสมที่สุดในวงกว้าง การปรับแต่งพารามิเตอร์ต่างๆเพื่อให้ TS ทำงานอย่างมีประสิทธิภาพสูงสุดจึงเป็นสิ่งที่จำเป็น ปัญหาที่เกิดขึ้นก็คืออะไรคือพารามิเตอร์ที่เหมาะสมที่สุดสำหรับระบบหนึ่งๆ??? ผลการทดลองในรูปที่ 1.18 และ 1.19 แสดงให้เห็นอย่างชัดเจนว่าถ้า t_{tabu}

มีค่าน้อยลงไป TS อาจจะไม่สามารถหลุดออกจากการวนรอบอยู่กับที่ได้ (ซึ่งในระหว่างขบวนการค้นหาคำตอบ เราไม่สามารถบอกได้ว่าขนาดของการวนรอบอยู่กับที่ที่เกิดขึ้นจะมีขนาดเท่าไร ทำให้ไม่สามารถเลือกค่า t_{tabu} ให้สัมพันธ์กันได้) ในขณะที่ถ้า t_{max} มีค่าลดลงไปก็จะมีผลต่อความสามารถในการหลุดออกจากการวนรอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ ■

การค้นหาคำตอบที่เหมาะสมที่สุดแบบตามูช่วยแก้ปัญหาของคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นและการวนรอบอยู่กับที่ พารามิเตอร์ที่สำคัญต่างๆในโครงสร้างของ TS มีผลต่อพฤติกรรมในการค้นหาคำตอบ โดยทฤษฎีแล้วยังไม่มีการคำนวณที่แน่นอนว่าพารามิเตอร์ต่างๆ ควรจะมีค่าเป็นเท่าไรที่จะทำให้การค้นหาคำตอบของ TS เป็นไปอย่างมีประสิทธิภาพสูงสุด โดยปกติแล้วการปรับค่าพารามิเตอร์ต่างๆ จะได้จากการสุ่มทดลองและสังเกตผล อย่างไรก็ตามการทดลองที่ได้พิสูจน์ให้เห็นแล้วว่า TS สามารถหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้อย่างมีประสิทธิภาพ ผู้อ่านที่สนใจในรายละเอียดเพิ่มเติมเกี่ยวกับการค้นหาแบบตามูสามารถศึกษาได้จากหนังสือ "Tabu Search" ใน [Glover and Laguna, 1997] ซึ่งเป็นหนังสือเล่มแรกที่มีเนื้อหาเชิงลึกของการค้นหาแบบตามูและแต่งโดยผู้คิดค้นอัลกอริทึมเอง จึงเหมาะแก่การเป็นหนังสืออ้างอิงได้เป็นอย่างดี

1.4 การค้นหาแบบตามูชนิดปรับตัวเองได้ Adaptive Tabu Search

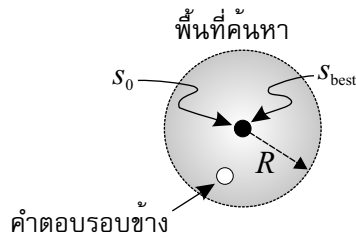
การค้นหาแบบตามูได้ถูกนำไปทำการปรับปรุงรายละเอียดมากมายหลายวิธี เพื่อเพิ่มประสิทธิภาพในการค้นหาคำตอบ เช่นการค้นหาตามูแบบชนิดโต้ตอบได้ (reactive tabu search) [Battiti and Tecchiolli, 1994] หรือการค้นหาแบบตามูชนิดปรับตัวเองได้ (adaptive tabu search) [Areerak and Sujitjorn, 2002] ฯลฯ ในหัวข้อนี้จะทำการศึกษาถึงรายละเอียดของตามูชนิดปรับตัวเองได้ รวมไปถึงตัวอย่างการนำไปประยุกต์ใช้งานแบบต่างๆ

1.4.1 หลักพื้นฐานการประยุกต์ใช้การค้นหาแบบตามู

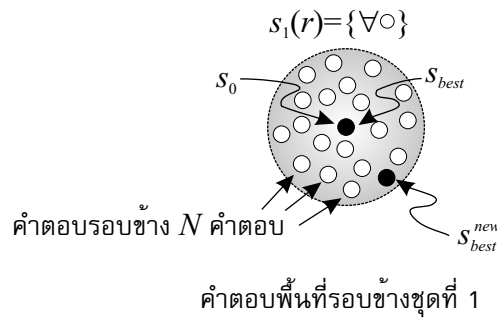
กลไกในการนำเอาการค้นหาแบบตามูไปใช้งานจริงสามารถสรุปได้ดังนี้

▷ อัลกอริทึมการค้นหาแบบตามูพื้นฐาน

1. กำหนดให้ t เป็นจำนวนรอบของการค้นหาและ t_{max} เป็นจำนวนรอบสูงสุดที่จะทำการค้นหา
2. ทำการสุ่มคำตอบเริ่มต้น s_0 ภายในพื้นที่ค้นหารอบข้างในรัศมี R ขณะนี้ s_0 มีสถานะเป็นคำตอบที่ดีที่สุดแบบวงแคบเฉพาะถิ่น กำหนดให้ค่า $s_0 = s_{best}$ (ดูรูปที่ 1.20)
3. ทำการสุ่มเลือกคำตอบใหม่ N คำตอบโดยทำการเดินรอบๆ คำตอบ s_0 ภายในพื้นที่ค้นหา กำหนดให้ $s_1(r)$ เป็นเซตของคำตอบที่ประกอบไปด้วยคำตอบใหม่ทั้งหมด N คำตอบ (ดูรูปที่ 1.21)
4. ทำการประเมินค่าของคำตอบที่อยู่ภายใน $s_1(r)$ ด้วยฟังก์ชันวัตถุประสงค์ แล้วเลือกคำตอบที่ดีที่สุด กำหนดให้คำตอบนั้นเป็น s_{best}^{new} (ดูรูปที่ 1.21)
5. ถ้า s_{best}^{new} ดีกว่า s_{best} แล้ว กำหนดให้ $s_{best} = s_{best}^{new}$ (ดูรูปที่ 1.22) ถ้าไม่มีคำตอบใดที่ดีกว่า s_{best} ให้ไปยังขั้นตอนที่ 6
6. กำหนดค่า $s_0 = s_{best}$ (ดูรูปที่ 1.23)



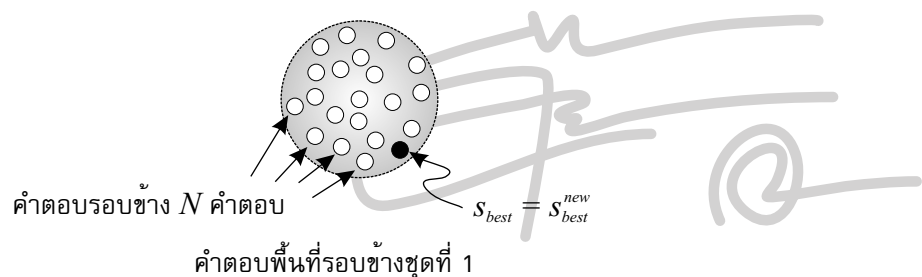
รูปที่ 1.20: สุ่มคำตอบ s_0 ในพื้นที่ค้นหา



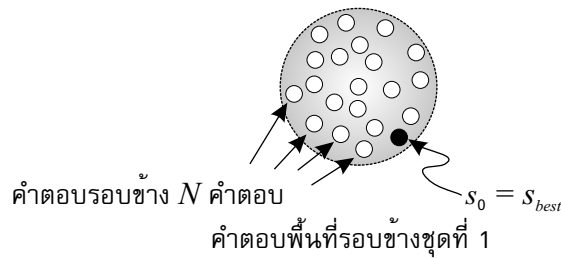
รูปที่ 1.21: คำตอบพื้นที่รอบข้าง s_0

7. ถ้า s_{best} ไม่อยู่ใน $NeighborList$ ให้ทำการบันทึก s_{best}^{new} ลงใน $NeighborList$
8. ถ้า $t > t_{max}$ ให้ยุติการค้นหา คำตอบที่ได้จะเป็นคำตอบที่ดีที่สุดตั้งแต่ทำการค้นหามา
9. ถ้า $t < t_{max}$ ให้กลับไปเริ่มที่ข้อ 2 ใหม่และทำการค้นหาจนกระทั่งถึงเป้าหมายที่ตั้งไว้ (ดูรูปที่ 1.24) ◀

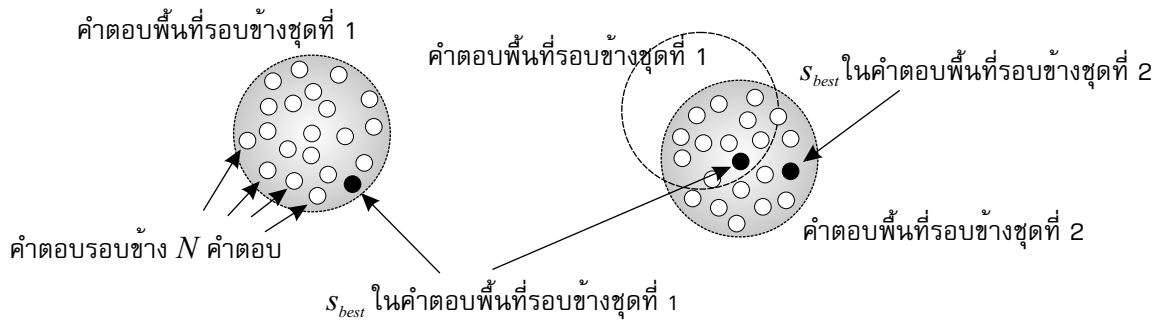
อัลกอริทึมการค้นหาลูปพื้นฐานหรือ TS ข้างต้น เป็นการประยุกต์ใช้งานจริง ดังนั้นในแง่ของการโปรแกรมแล้ว TS ถือว่ามีโครงสร้างของอัลกอริทึมที่ไม่มีความซับซ้อนแต่อย่างใด ความแตกต่างในการนำเอา TS ไปประยุกต์ใช้กับงานด้านต่างๆ จึงมีเพียงฟังก์ชันวัตถุประสงค์เท่านั้น กลไกเดียวกันของ TS สามารถนำไปใช้กับปัญหาต่างๆ ได้ โดยสิ่งที่ต้องพิจารณาปรับแก้ก็คือพารามิเตอร์ของ TS เช่นขนาดของรายการต้องห้าม ค่าระยะเวลาการคงอยู่ของสถานะต้องห้าม เป็นต้น ดังที่ได้กล่าวมาแล้วว่าสำหรับปัญหาที่แตกต่างกัน ค่าพารามิเตอร์ของ TS นี้จะมีค่าที่เหมาะสมกับปัญหาแตกต่างกันด้วย ในระหว่างการค้นหาคำตอบ TS จะทำการเลือกคำตอบด้วยกลไกของตามพื้นฐาน พร้อมทั้งประเมินสถานะของคำตอบผ่านค่าวัตถุประสงค์จากฟังก์ชันวัตถุประสงค์ รูปที่ 1.25 แสดงแผนผังทั่วไปในการนำเอา TS มาประยุกต์ใช้ จากแผนผังดังกล่าว กลไกการทำงานของ TS จึงเปรียบเสมือน



รูปที่ 1.22: กำหนดค่าใหม่ให้กับคำตอบพื้นที่รอบข้างที่ดีที่สุด



รูปที่ 1.23: กำหนดค่าใหม่ให้กับ s_0



รูปที่ 1.24: การค้นหาคำตอบในรอบต่อไป

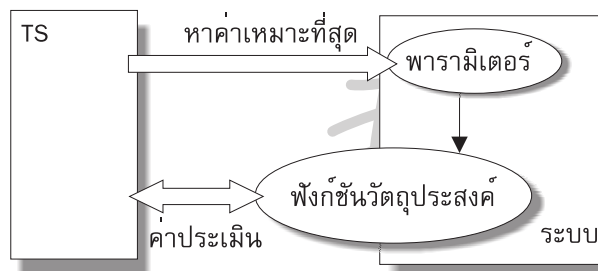
กับกระบวนการหาค่าเหมาะที่สุด (optimization) ของพารามิเตอร์ของระบบนั่นเอง

1.4.2 การค้นหาแบบตามูชนิดปรับตัวได้

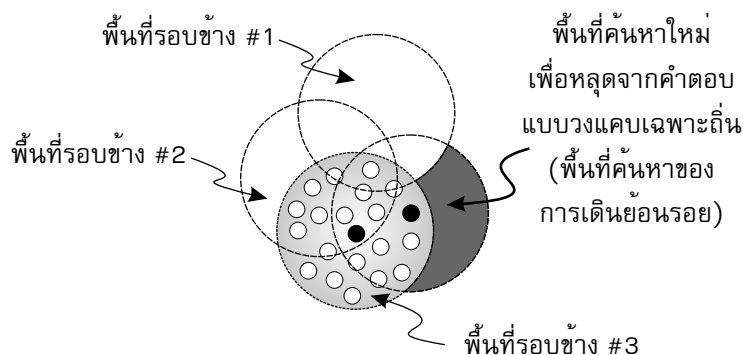
ในหลายๆ กรณี การค้นหาแบบตามูพื้นฐานยังมีประสิทธิภาพที่ไม่เพียงพอต่อการประยุกต์ใช้งาน จึงมีผู้นำเอาโครงสร้างของการค้นหาแบบตามูไปทำการปรับปรุง ซึ่งในที่นี้เป็นการปรับปรุงที่เรียกว่าการค้นหาแบบตามูชนิดปรับตัวได้ (Adaptive Tabu Search หรือ ATS) ซึ่งเป็นอัลกอริทึมที่ได้พัฒนาขึ้นโดยกลุ่มวิจัย สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี (ดู [Puangdownreong, Areerak, Srikaew, Sujitjorn, and Totarong, 2002]) การปรับปรุงดังกล่าวได้ทำการเพิ่ม 2 กลไกเข้าไปในการค้นหาแบบตามูธรรมดา กลไกดังกล่าวคือการเดินย้อนรอยและการปรับรัศมีการค้นหา ดังรายละเอียดในหัวข้อต่อไปนี้

- การเดินย้อนรอย (Back Tracking)

การเดินย้อนรอยเป็นขั้นตอนที่อนุญาตให้ระบบค้นหาทำการกลับไปค้นหาพื้นที่คำตอบเก่าที่เคยถูกค้นหามาแล้ว คำตอบที่ดีกว่าจะถูกเลือกจากพื้นที่คำตอบที่ถูกย้อนรอยนี้และคำตอบปัจจุบันที่มีอยู่ รูปที่ 1.26 แสดง



รูปที่ 1.25: โครงสร้างการประยุกต์ใช้งาน TS



รูปที่ 1.26: การเดินย้อนรอยใน ATS

รายละเอียดของขั้นตอนการย้อนรอย โดยจากการกำหนดคำตอบใหม่ให้เป็นจุดเริ่มต้นในการค้นหา พื้นที่ในการค้นหาใหม่ก็จะเกิดขึ้นด้วย นั่นหมายความว่า การค้นหาใหม่มีโอกาสที่จะหลุดออกจากคำตอบที่เป็นแบบวงแคบเฉพาะถิ่น คำตอบใหม่ที่ถูกใช้เป็นจุดเริ่มต้นนี้ **ไม่จำเป็น** จะต้องเป็นคำตอบที่ดีที่สุดในพื้นที่ปัจจุบัน การเดินย้อนรอยจึงเป็นกลไกหนึ่ง ที่สามารถเพิ่มประสิทธิภาพในการหลุดจากคำตอบที่เป็นแบบวงแคบเฉพาะถิ่นได้เป็นอย่างดี

• การปรับรัศมีการค้นหา (Adaptive Radius)

ขบวนการการปรับรัศมีการค้นหาจะทำการลดรัศมีการค้นหาในระหว่างการค้นหาลง ซึ่งการลดจะดำเนินไปเรื่อยๆ จนกระทั่งการค้นหาเข้าใกล้คำตอบที่ดีที่สุดแบบวงกว้าง โดยปกติแล้วรัศมีการค้นหาคำตอบในแต่ละรอบของการค้นหานั้นถือเป็นองค์ประกอบสำคัญอย่างหนึ่ง รัศมีการค้นหาที่กว้างจะให้ผลของการค้นหาที่หยาบ ซึ่งอาจจะมีผลให้การค้นหาพลาดคำตอบที่ต้องการได้ ในทางตรงกันข้าม ถ้ารัศมีการค้นหาคำตอบมีขนาดเล็ก การค้นหาคำตอบจะต้องใช้เวลาที่มากขึ้น ยิ่งไปกว่านั้น รัศมีการค้นหาคำตอบที่เล็กมากๆ อาจจะทำให้ระบบไม่สามารถครอบคลุมพื้นที่ของคำตอบที่ต้องการค้นหาได้อย่างมีประสิทธิภาพ ดังนั้นการปรับรัศมีการค้นหาให้เหมาะสมกับสถานการณ์การค้นหาคำตอบในขณะหนึ่งๆ จึงสามารถเพิ่มประสิทธิภาพของการค้นหาได้ ใน ATS จึงได้พัฒนาการปรับรัศมีการค้นหาให้เหมาะสมในระหว่างการค้นหาคำตอบ โดยการใช้ค่าการประเมินเป็นองค์ประกอบในการพิจารณาปรับค่ารัศมีการค้นหา กล่าวคือเมื่อคำตอบปัจจุบันให้ค่าการประเมินที่ดีขึ้น รัศมีการค้นหาจะลดลงตามความสัมพันธ์ดังนี้

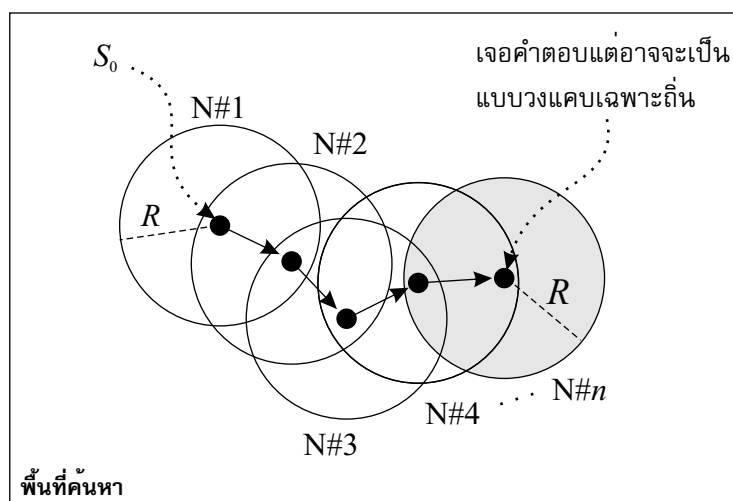
$$radius_{new} = \frac{radius_{old}}{DF}$$

โดยที่ $DF > 1$ เป็นตัวประกอบการลดของรัศมี (decreasing factor) ค่าที่เหมาะสมของ DF จะแตกต่างกัน และขึ้นอยู่กับปัญหาแต่ละอย่าง

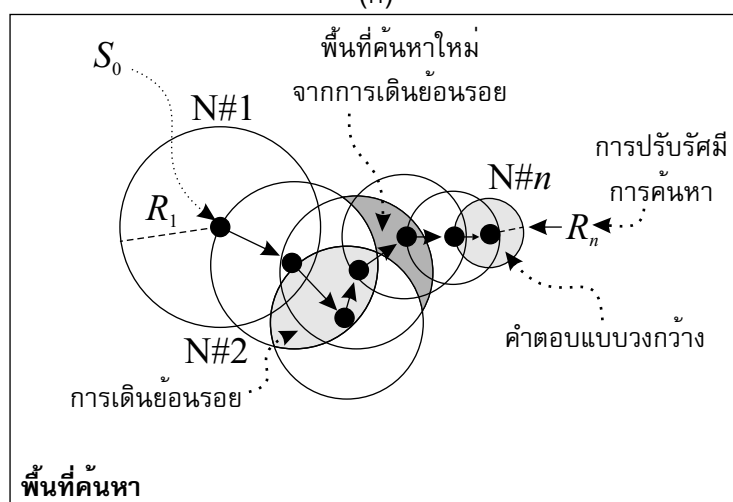
โดยปกติการค้นหาแบบตามชนิดปรับตัวได้จะเพิ่มกลไกการเดินย้อนรอยเข้าไปในขั้นตอนที่ 6 และ 7 ของการค้นหาแบบตามชนิดธรรมดา ในขณะที่กลไกการปรับรัศมีการค้นหาจะใช้ในช่วงท้ายสุดของขั้นตอนที่ 7 รูปที่ 1.27-(ก) และ 1.27-(ข) แสดงรายละเอียดความแตกต่างระหว่างการค้นหาแบบตามชนิดธรรมดาและชนิดปรับตัวได้

1.5 การประยุกต์ใช้งาน ATS: การออกแบบตัวควบคุม PID ที่เหมาะสมที่สุด

ที่มา ขอขอบพระคุณ อ.กองพันธ์ อารีรักษ์ อ.มัธยดิน ดาเต๊ะและรศ.ดร.สรารุณี สุจิตจร จากสาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี สำหรับรายละเอียดในโจทย์ปัญหานี้



(ก)



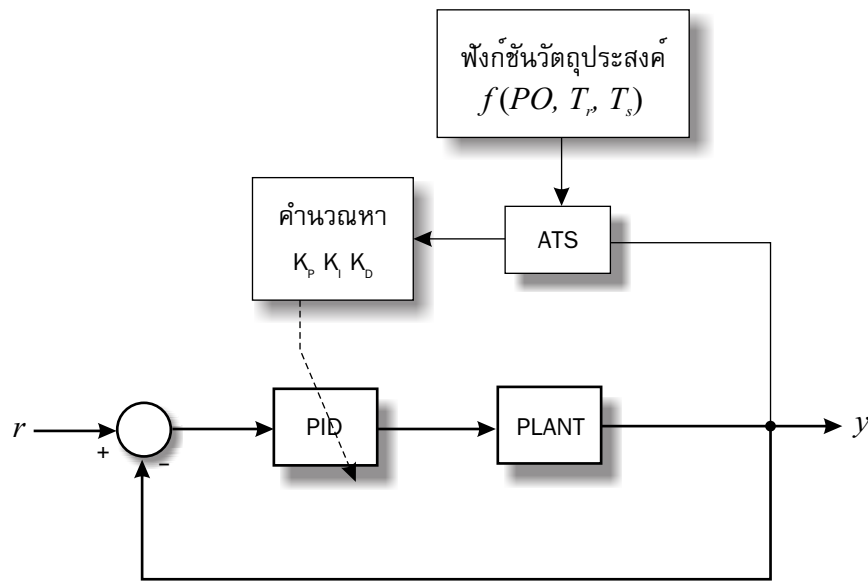
(၁)

รูปที่ 1.27: (ก) การค้นหาแบบตามชนิดธรรมดา (ข) การค้นหาแบบตามชนิดปรับตัวได้

ปัจจุบันได้มีการนำเอาตัวควบคุมพีไอดี (หรือ PID Controller) มาใช้ในระบบควบคุมกันอย่างแพร่หลาย โดยวิธีการออกแบบตัวควบคุมพีไอดีนั้นสามารถทำได้หลากหลายวิธี แต่ละวิธีก็มีจุดเด่นจุดด้อยแตกต่างกันออกไป ในหัวข้อนี้จะทำการออกแบบตัวควบคุมพีไอดี โดยนำเอาการค้นหาแบบตามขูชนิดปรับตัวได้ มาใช้ในการค้นหาคำตอบของตัวควบคุมพีไอดีได้แก่ K_P K_I และ K_D ดังรายละเอียดต่อไปนี้

1.5.1 โครงสร้างของระบบ

ในการประยุกต์ใช้การค้นหาคำตอบมาแก้ปัญหานั้น โครงสร้างของระบบจะแตกต่างกันออกไป ในที่นี้เราจะทำการออกแบบตัวควบคุมพีไอดี ดังนั้นองค์ประกอบหลักๆ จะเป็นส่วนที่เกี่ยวข้องกับระบบควบคุม แผนผังการทำงานของระบบมีแสดงในรูปที่ 1.28 แผนผังดังกล่าวแสดงโครงสร้างของการนำเอาการค้นหาคำตอบแบบตาม มาช่วยในการค้นหาพารามิเตอร์ของตัวควบคุมพีไอดีคือ K_P K_I และ K_D โดยมีเป้าหมายเหมือนกับระบบควบคุมทั่วๆ ไป คือช่วงเวลาขึ้น (rise time หรือ T_R) ช่วงเวลาเข้าที่ (settling time หรือ T_S) และเปอร์เซ็นต์ค่าพุ่งเกิน (percent overshoot หรือ PO) ซึ่งค่าทั้งสามจะถูกใช้เป็นพารามิเตอร์ในฟังก์ชันวัตถุประสงค์ของการค้นหา โดยฟังก์ชัน



รูปที่ 1.28: แผนผังระบบควบคุมป้อนกลับ

วัตถุประสงค์จะทำการประเมินค่าของตัวควบคุมจากพารามิเตอร์ที่กำลังค้นหา K_P K_I และ K_D โดยมีเป้าหมายให้ระบบมีการตอบสนองของสัญญาณที่ดีที่สุด กล่าวคือให้ค่า T_R T_S และ PO ที่น้อยที่สุดนั่นเอง เนื่องจากเป้าหมายในฟังก์ชันวัตถุประสงค์มีอยู่ถึง 3 ค่า จึงได้มีการออกแบบให้มีการปรับฟังก์ชันวัตถุประสงค์ โดยแทนที่จะทำการประเมินค่าของระบบแยกเป็น 3 เป้าหมาย เราสามารถทำการรวมเป็นเพียงฟังก์ชันเดียว ซึ่งเรียกว่าเป็นฟังก์ชันถ่วงน้ำหนัก (weight function) ดังนี้

$$w(T_R, T_S, PO) = \sigma T_R + \alpha T_S + \gamma PO$$

โดยที่

$$\sigma + \alpha + \gamma = 1$$

และ

- $\sigma \geq 0$ คือสัมประสิทธิ์สำหรับกำหนดนัยสำคัญของค่า T_R
- $\alpha \geq 0$ คือสัมประสิทธิ์สำหรับกำหนดนัยสำคัญของค่า T_S
- $\gamma \geq 0$ คือสัมประสิทธิ์สำหรับกำหนดนัยสำคัญของค่า PO

ค่า σ α และ γ ข้างต้นเป็นส่วนที่จะต้องออกแบบเอง เพราะว่าเป็นค่าที่ขึ้นอยู่กับว่าต้องการเน้นความสำคัญของระบบที่เป้าหมายใด ค่าการประเมินที่น้อยกว่าหมายถึงระบบที่ดีกว่า ส่วนการนำเอาค่า T_R T_S และ PO มาทำการพิจารณาในฟังก์ชันการประเมินร่วมกัน เราควรต้องทำให้พารามิเตอร์ทั้งสามตัวนั้นเป็นบรรทัดฐานเดียวกันก่อน เพราะค่าเชิงตัวเลขของแต่ละพารามิเตอร์มีนัยสำคัญที่แตกต่างกัน เช่น PO อาจจะมีอยู่ในรูปค่าเปอร์เซ็นต์ ในขณะที่ค่า T_R และ T_S จะอยู่ในหน่วยเวลา ส่วนฟังก์ชันถ่ายโอนในโดเมน s ของตัวควบคุมแบบ PID นี้คือ

$$C(s) = K_P + K_D s + \frac{K_I}{s}$$

ในหัวข้อต่อไปจะได้นำเสนอรายละเอียดของหลักการทำงานของตัวควบคุมข้างต้น รวมไปถึงผลการทำงานของระบบ

1.5.2 หลักการทำงาน

หลักการทำงานของระบบโดยรวมอธิบายได้จากรูปที่ 1.28

▷ อัลกอริทึมการค้นหามิติเตอร์สำหรับตัวควบคุมแบบ PID ด้วย ATS

1. อินพุตอ้างอิง r เป็นสัญญาณแบบขั้นบันไดหนึ่งหน่วย (unit step) ป้อนเข้าสู่ระบบ
2. สัญญาณ u เป็นค่าความผิดพลาดระหว่างอินพุตและเอาต์พุตของระบบป้อนเข้าสู่ตัวควบคุมพีไอดี
3. กำหนดค่าต่ำสุดและสูงสุดของ K_P K_I และ K_D (ไม่มีประโยชน์ที่จะทำการค้นหาในช่วงที่ไม่สามารถใช้งานได้จริง) ดังนั้น $K_P \in [0, K_{P_{MAX}}]$ $K_I \in [0, K_{I_{MAX}}]$ และ $K_D \in [0, K_{D_{MAX}}]$ ค่า $K_{P_{MAX}}$ $K_{I_{MAX}}$ และ $K_{D_{MAX}}$ เป็นค่าที่ผู้ใช้ออกแบบค่าเอง
4. กำหนดค่าเริ่มต้น (initial value) ให้กับ K_P K_I และ K_D
5. กำหนดพารามิเตอร์ของระบบการค้นหา ซึ่งได้แก่จำนวนรอบในการค้นหาคำตอบและค่าการประเมินที่เป็นเป้าหมาย
6. คำนวณหาค่าเอาต์พุต y จากตัวควบคุมโดยใช้ค่า K_P K_I และ K_D ที่มีอยู่
7. คำนวณหาค่า T_R T_S และ PO เพื่อนำไปคำนวณหาค่าการประเมินจากฟังก์ชันวัตถุประสงค์ w (หรือฟังก์ชันถ่วงน้ำหนัก)
8. ค่าการประเมินจากฟังก์ชันวัตถุประสงค์ w จะถูกใช้เป็นอินพุตเข้าสู่ระบบการค้นหาคำตอบด้วย ATS โดยการค้นหาจะเป็นไปในทิศทางที่ให้ค่าการประเมินน้อยที่สุดดังนี้

$$\text{ค่าการประเมิน} = \min[w(T_R, T_S, PO)]$$

9. นำคำตอบที่ได้จากระบบ ATS ซึ่งได้แก่ K_P K_I และ K_D ไปทำการประเมิน โดยใช้ค่า K_P K_I และ K_D ใหม่ที่ได้
10. ทำขั้นตอนต่างๆ ซ้ำไปเรื่อยๆ จนค่าการประเมินจากฟังก์ชันวัตถุประสงค์เป็นไปตามเป้าหมายที่ต้องการ ◁

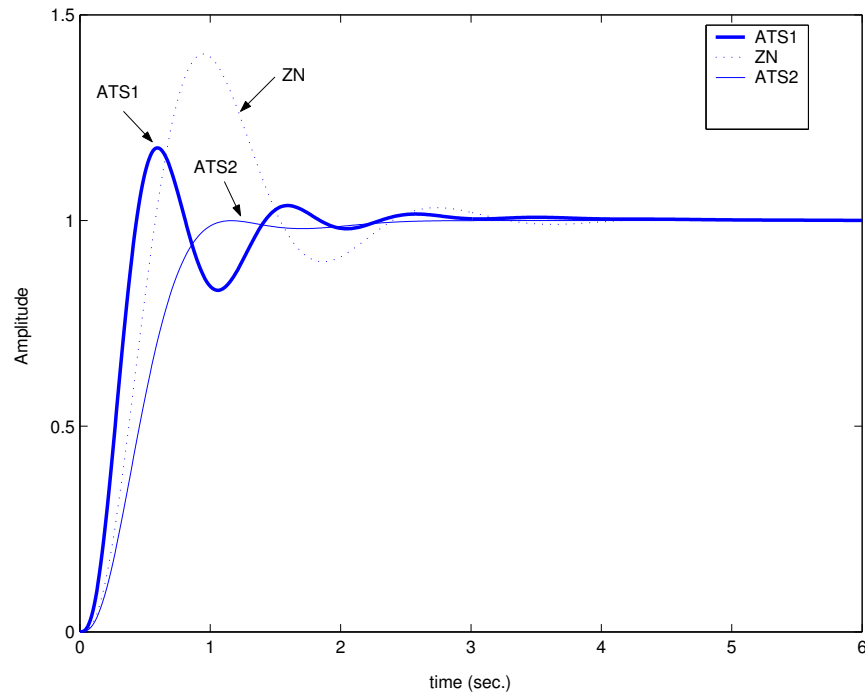
ตัวอย่างผลการค้นหาที่ได้จาก ATS เปรียบเทียบกับวิธีการคำนวณแบบดั้งเดิม (วิธีซีกเลอร์นิโคล) แสดงในรูปที่ 1.29 โดยมีฟังก์ชันถ่วงน้ำหนักต่อไปนี้

$$P_1(s) = \frac{1}{(1+s)(1+\frac{s}{6})^3}$$

และใช้ฟังก์ชันวัตถุประสงค์ดังนี้

$$w_1(T_R, T_S, PO) = 0.34T_R + 0.33T_S + 0.33PO$$

สังเกตว่าฟังก์ชันวัตถุประสงค์ให้ความสำคัญกับทุกเป้าหมายเท่าๆ กัน) ผลลัพธ์ที่ได้แสดงถึงคำตอบจาก ATS ที่ดีกว่าวิธีดั้งเดิมอย่างเห็นได้ชัด สิ่งสำคัญในที่นี้ก็คือเราไม่จำเป็นต้องใช้กรรมวิธีเชิงคณิตศาสตร์หรือพีชคณิต ดังที่ใช้ในวิธีซีกเลอร์นิโคล เพื่อแก้ปัญหาพารามิเตอร์ของตัวควบคุม PID เทคนิควิธีการค้นหาคำตอบด้วยปัญญาเชิงคำนวณนี้ช่วยลดเวลาการทำงาน โดยเฉพาะกับปัญหาที่มีความยุ่งยากซับซ้อน ดังจะได้นำเสนอในหัวข้อต่อไป



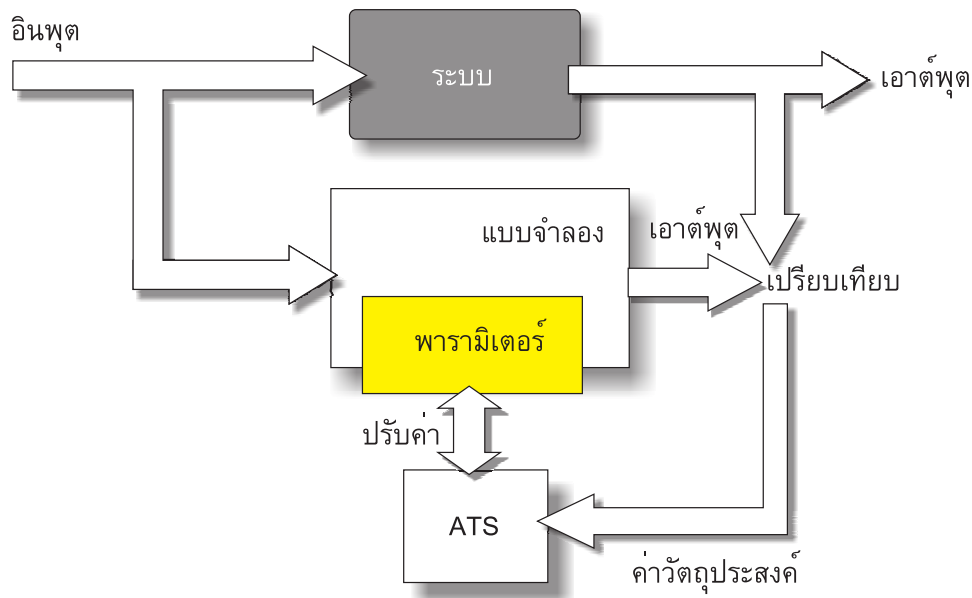
รูปที่ 1.29: ตัวอย่างผลตอบสนองที่ได้ ($K_P = 2.7240$ $K_I = 2.0266$ และ $K_D = 0.7129$)

1.6 การประยุกต์ใช้งาน ATS: การระบุเอกลักษณ์เพื่อหาแบบจำลองของระบบ

การระบุเอกลักษณ์หรือ system identification เป็นกระบวนการหลักอย่างหนึ่งที่ใช้ในการหาแบบจำลองของระบบทั้งในระบบที่เป็นเชิงเส้นและไม่เป็นเชิงเส้น โดยทั่วไปแล้ววิธีการวิเคราะห์แบบถดถอย (regression analysis) สามารถนำมาประยุกต์ใช้กับระบบได้หลากหลาย วิธีดั้งเดิมต่างๆ เช่น ARX ARMAX หรือ Box-Jenkin ฯลฯ ยังได้รับความนิยมในการใช้งาน อย่างไรก็ตาม มีระบบอีกมากมายที่ไม่สามารถประยุกต์ใช้แบบจำลองดังกล่าวได้ ปัญหาเชิงคำนวณจึงได้รับความนิยม ในการนำมาค้นหาพารามิเตอร์ของแบบจำลองได้อย่างเหมาะสมที่สุด ในหัวข้อนี้จะได้นำเสนอตัวอย่างการนำเอา ATS มาประยุกต์ใช้งานการระบุเอกลักษณ์เพื่อหาแบบจำลองของระบบแบบต่างๆ เนื้อหาที่นำเสนอจะเรียบเรียงมาจากผลงานวิจัยของผู้เขียนและทีมงานวิจัยของสาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี ผู้อ่านที่สนใจสามารถศึกษารายละเอียดเพิ่มเติมจาก [Sujitjorn, Srikaew, Pongdownreong, Attakitmongkol, and Totarong, 2003] และ [Puangdownreong, Areerak, Srikaew, Sujitjorn, and Totarong, 2002] รวมถึงเอกสารอื่นๆ ที่เกี่ยวข้องได้

1.6.1 การหาพารามิเตอร์ในแบบจำลองด้วย ATS

รูปที่ 1.30 แสดงโครงสร้างทั่วไปในการหาพารามิเตอร์ของแบบจำลอง พิจารณาระบบที่มีแบบจำลองที่ยังไม่ทราบพารามิเตอร์ การระบุเอกลักษณ์คือการหาค่าพารามิเตอร์ต่างๆ ที่ทำให้แบบจำลองตอบสนองต่ออินพุตของระบบได้เหมือนกับเอาต์พุตจริง ดังนั้นกลไกการค้นหาค่าพารามิเตอร์ดังกล่าว จะอาศัยการเปรียบเทียบผลลัพธ์ของแบบจำลองกับเอาต์พุตจริงของระบบ แล้วทำการปรับค่าพารามิเตอร์ไปในทิศทางที่ทำให้ค่าความแตกต่างระหว่างเอาต์พุตทั้งสองลดลง หรือไปเป็นตามเงื่อนไขที่กำหนด ในที่นี้ โดยกลไกการค้นหาค่าพารามิเตอร์จะใช้ ATS เป็นหลัก ประสิทธิภาพในการค้นหาค่าตอบของ ATS ทำให้สามารถนำไปประยุกต์ใช้งานกับแบบจำลอง ทั้งที่เป็นแบบเชิงเส้นและไม่เป็นเชิง



รูปที่ 1.30: แผนผังการค้นหาพารามิเตอร์ของแบบจำลองระบบด้วยการระบุเอกลักษณ์

เส้นใต้ ดังแสดงด้วยตัวอย่างแบบจำลองต่างๆ ในหัวข้อต่อไปนี้

1.6.2 แบบจำลองการหกลมละลายของพอลิเมอร์แบบ Die-Swell

แบบจำลองการหกลมละลายของพอลิเมอร์แบบ die-swell สามารถอธิบายได้ด้วยความสัมพันธ์ต่อไปนี้

$$B = (a + bS_R^x)^y \quad (1.11)$$

โดยที่ B คืออัตราส่วน die-swell
 S_R คือความเครียดเฉือนในการคืบสภาพ
 a, b, x และ y คือพารามิเตอร์ของแบบจำลอง

ตารางที่ 1.1 แสดงข้อมูลจริงจากการหกลมละลายของพอลิเมอร์ ข้อมูลอินพุต/เอาต์พุต (S_R, B) ดังกล่าวถูกนำไปใช้เปรียบเทียบกับคู่อินพุต/เอาต์พุตที่ได้จากแบบจำลอง ตัวอย่างผลการค้นหาพารามิเตอร์ของระบบด้วย ATS มีดังนี้

$$\begin{aligned} a &= 2.6811 \\ b &= 0.0793 \\ x &= 2.7946 \\ y &= 0.5775 \end{aligned}$$

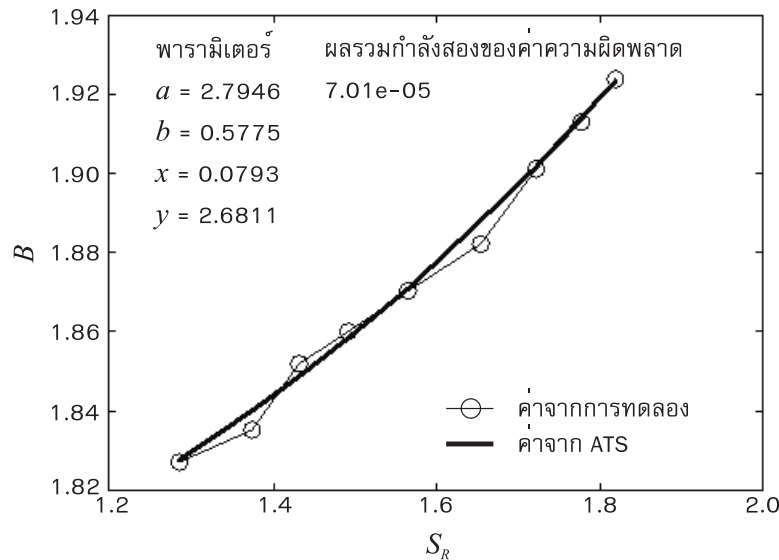
ผลลัพธ์ที่ได้จากแบบจำลองเทียบกับข้อมูลจริงแสดงในรูปที่ 1.31 จะเห็นว่าพารามิเตอร์ที่ค้นหาได้ด้วย ATS ให้เอาต์พุตของแบบจำลองที่สอดคล้องกับข้อมูลจริง

1.6.3 แบบจำลองระบบ 2 มวลที่มีเรโซแนนซ์เชิงบิดและการอิ่มตัวไม่เป็นเชิงเส้น Torsional Resonance & Nonlinear Saturation Model

รูปที่ 1.32 แสดงระบบเชิงหมุน 2 มวลที่ปรากฏเรโซแนนซ์เชิงบิดขึ้น เรโซแนนซ์เชิงบิดดังกล่าวจะส่งผลกระทบต่อประสิทธิภาพความเร็วในรูปของการสั่นด้วยปริมาณน้อยๆ แบบจำลองทางคณิตศาสตร์ที่มีการลดรูปอันดับของฟังก์ชัน

ตารางที่ 1.1: ข้อมูลจริงจากการหลอมละลายของพอลิเมอร์

S_R	1.287	1.374	1.432	1.492	1.565	1.655	1.722	1.779	1.819
B	1.827	1.835	1.852	1.860	1.870	1.882	1.901	1.913	1.924



รูปที่ 1.31: ผลตอบสนองของแบบจำลองกับข้อมูลจริงของการหลอมละลายของพอลิเมอร์

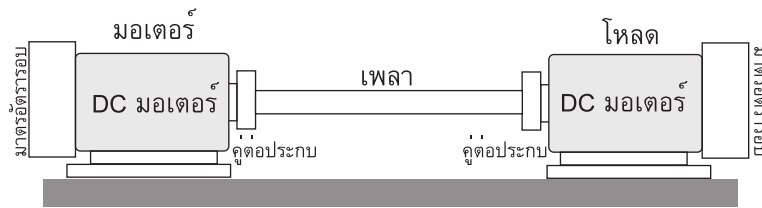
ชั้นถ่ายโอนแสดงได้ดังนี้

$$G(s) = \frac{K}{a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (1.12)$$

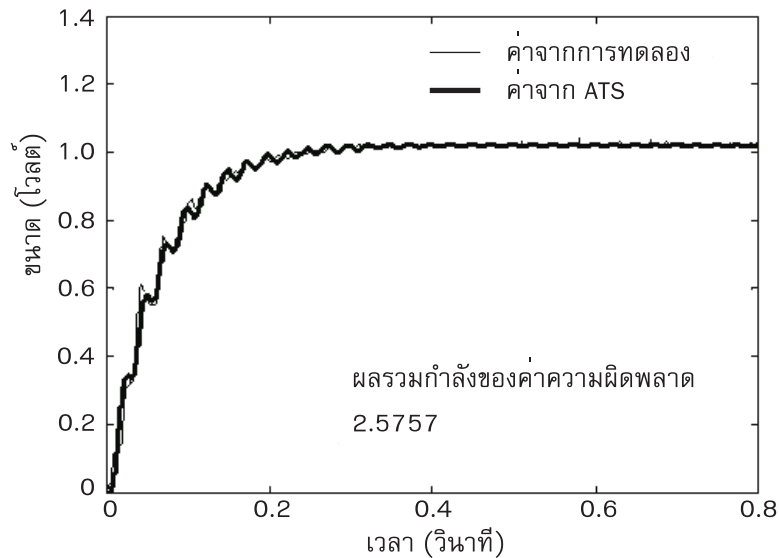
เราสามารถพิจารณาให้ระบบเป็นเชิงเส้นด้วยแบบจำลองอันดับ 5 ข้างต้น โดยมีสัมประสิทธิ์ของแบบจำลองคือสัมประสิทธิ์ของแบบจำลองจากการค้นหาด้วย ATS สรุปได้ดังแสดงในตารางที่ 1.2 ดังนี้ รูปที่ 1.33 แสดงผลตอบสนองของแบบจำลองเชิงเส้นของระบบเชิงหมุน 2 มวลเทียบกับผลตอบสนองจริงของระบบ จะเห็นได้ชัดเจนว่าผลตอบสนองของแบบจำลองสอดคล้องกับผลตอบสนองจริงของระบบ แบบจำลองในรูปของฟังก์ชันถ่ายโอนข้างต้นสามารถใช้งานได้ในช่วงการทำงานช่วงหนึ่งเท่านั้น ในการที่จะเพิ่มประสิทธิภาพเชิงพลวัตของแบบจำลองดังกล่าวได้รวมเอาผลป้อนกลับและไปข้างหน้ามารวมไว้ด้วย ถ้าเราต้องการที่จะขยายช่วงการทำงานให้เพิ่มขึ้นตามการใช้งานจริง ผลที่ได้จะทำให้แบบจำลองดังกล่าวเกิดความไม่เป็นเชิงเส้นขึ้น ความไม่เป็นเชิงเส้นดังกล่าวเกิดขึ้นจากกลไกป้องกันวงจรรขยายตัวขับ (drive amplifier) และตัวชดเชย (compensator) ระบบที่เปลี่ยนไปนี้สามารถกล่าวได้ว่ามีคุณลักษณะของการอิ่มตัวแบบไม่เป็นเชิงเส้น (nonlinear saturation) เกิดขึ้น ซึ่งพารามิเตอร์ต่างๆ มีแสดงในรูปที่ 1.34 โดยที่ m คือค่าความชัน x_u จุดเริ่มต้นการอิ่มตัวด้านบวกและ x_l คือจุดเริ่มต้นการอิ่มตัวด้านลบ คุณลักษณะการอิ่มตัวดังกล่าวมีลักษณะความไม่เป็นเชิงเส้นอย่างชัดเจน จึงเป็นการยากมากที่จะระบุเอกลักษณ์ด้วยวิธีการดั้งเดิม อย่างไรก็ตาม การประยุกต์ใช้ ATS สามารถที่จะค้นหาพารามิเตอร์ของคุณลักษณะการ

ตารางที่ 1.2: พารามิเตอร์ของระบบเรโซแนนซ์เชิงบิดที่ค้นหาได้ด้วย ATS

K	a_0	a_1	a_2	a_3	a_4	a_5
8.60×10^{11}	8.42×10^{11}	5.38×10^{10}	1.24×10^8	9.44×10^5	1590	1

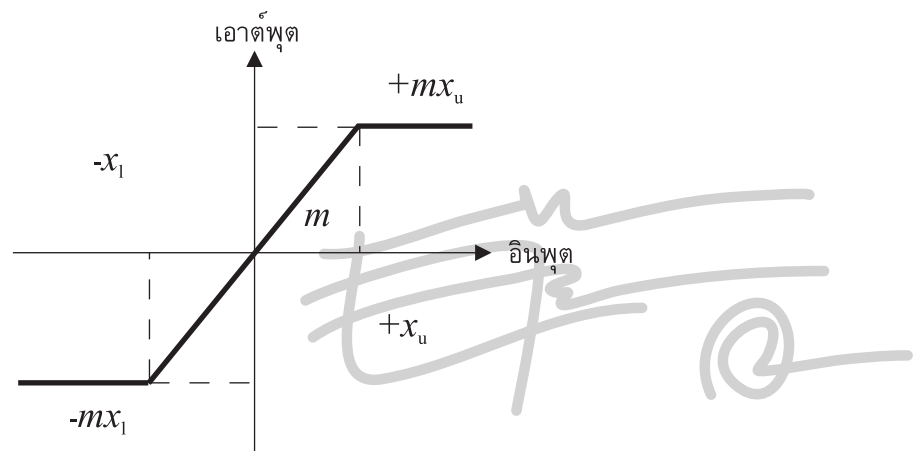


รูปที่ 1.32: ระบบสองมวลที่เกิดเรโซแนนซ์เชิงบิด

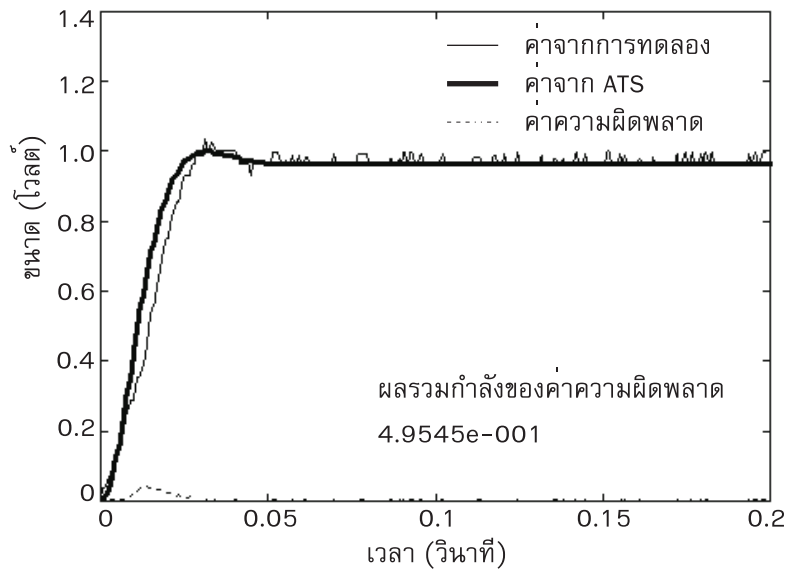


รูปที่ 1.33: ผลตอบสนองของแบบจำลองเชิงเส้นของระบบเชิงหมุน 2 มวลเทียบกับผลตอบสนองจริง แกน y ของกราฟคือขนาดของความเร็ววัดพร้อมกับโหลด

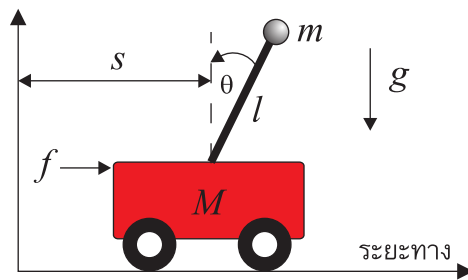
อิมพัลส์ดังกล่าวได้เป็นอย่างดีที่น่าพอใจ ผลลัพธ์ที่ได้คือ $m = 1.1447$ $x_u = 2.7590$ และ $x_l = 1.9159$ รูปที่ 1.35 แสดงผลตอบสนองของแบบจำลองที่ได้ ซึ่งสอดคล้องกับผลตอบสนองจริงของระบบเป็นอย่างดี ในที่นี้ ATS ให้ผลการค้นหารามิเตอร์ทั้งแบบจำลองระบบที่เป็นเชิงเส้นและไม่เป็นเชิงเส้นได้อย่างมีประสิทธิภาพ



รูปที่ 1.34: คุณลักษณะการอิมพัลส์



รูปที่ 1.35: ผลตอบสนองของแบบจำลองระบบ 2 มวลที่มีโรโซแนซ์เชิงบิดแบบไม่เป็นเชิงเส้นเทียบกับข้อมูลจริงของระบบ



รูปที่ 1.36: รถเพนดูลัมผกผัน

1.6.4 แบบจำลองรถเพนดูลัมผกผัน Inverted Pendulum Model

พิจารณาการหาแบบจำลองของระบบรถเพนดูลัมผกผัน (inverted pendulum cart system) ดังแสดงในรูปที่ 1.36 ถ้าพิจารณามุมการสั่น θ ที่น้อยๆ เราสามารถจำลองระบบดังกล่าวให้เป็นแบบเชิงเส้นได้ด้วยฟังก์ชันถ่ายโอนอันดับ 10 ดังต่อไปนี้

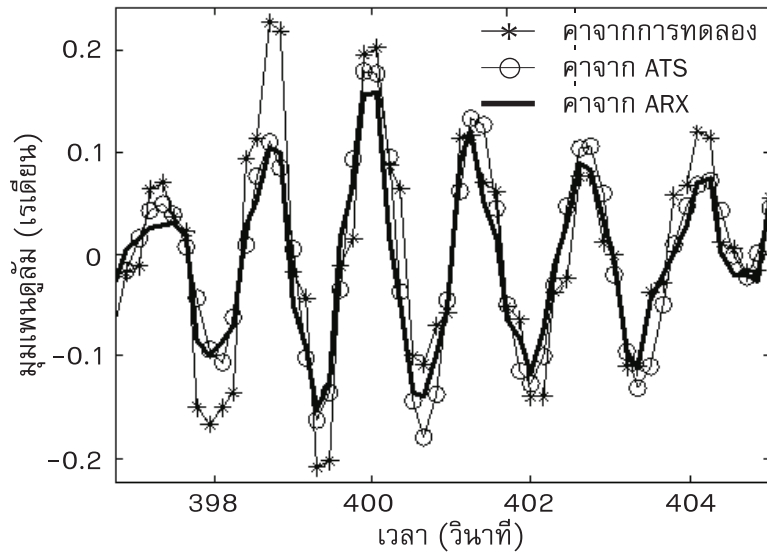
$$G(s) = \frac{N(s)}{D(s)} \quad (1.13)$$

$$N(s) = z_{10}s^{10} + z_9s^9 + z_8s^8 + z_7s^7 + z_6s^6 + z_5s^5 + z_4s^4 + z_3s^3 + z_2s^2 + z_1s + z_0 \quad (1.14)$$

$$D(s) = p_{10}s^{10} + p_9s^9 + p_8s^8 + p_7s^7 + p_6s^6 + p_5s^5 + p_4s^4 + p_3s^3 + p_2s^2 + p_1s + p_0 \quad (1.15)$$

ผลการค้นหาพารามิเตอร์ของแบบจำลองอันดับ 10 ข้างต้นด้วย ATS แสดงในสมการต่อไปนี้

$$N(s) = -0.004152s^{10} + 35.06s^9 - 3719s^8 + 3.14 \times 10^4s^7 - 2.53 \times 10^6s^6 + 1.48 \times 10^6s^5 - 2.90 \times 10^8s^4 + 1.17 \times 10^9s^3$$



รูปที่ 1.37: มุมการแกว่งของเฟนดูลัม (ข้อมูลจริงเทียบกับแบบจำลองอันดับ 10 จาก ARX และ ATS)

$$+4.90 \times 10^8 s^2 + 1.61 \times 10^{10} s - 2.99 \times 10^{10} \quad (1.16)$$

$$D(s) = 0.0851s^{10} + 152.5s^9 + 12145s^8 + 971670s^7 + 1.69 \times 10^7 s^6 + 3.22 \times 10^8 s^5 + 3.61 \times 10^9 s^4 + 1.31 \times 10^{10} s^3 + 6.26 \times 10^{10} s^2 + 9.16 \times 10^{10} s + 1.52 \times 10^{11} \quad (1.17)$$

รูปที่ 1.37 แสดงข้อมูลจริงจากการวัดมุมการแกว่งของเฟนดูลัม เทียบกับผลตอบสนองของแบบจำลองที่ได้จากการค้นหาด้วย ATS ในสมการที่ 1.16 และ 1.17 พร้อมกับผลตอบสนองที่ได้จากวิธีการ ARX ค่าผลรวมกำลังสองของความผิดพลาดมีค่าเท่ากับ 7.7966 และ 6.2859 สำหรับแบบจำลองจากวิธี ARX และ ATS ตามลำดับ ในรูปที่ 1.38 แสดงผลการค้นหาพารามิเตอร์ของแบบจำลองรถเฟนดูลัมผกผันที่มีอันดับเท่ากับ 5 (เนื่องจากแบบจำลองที่มีอันดับสูงเท่ากับ 10 จะไม่สะดวกในการใช้งานจริง) พารามิเตอร์ของแบบจำลองอันดับ 5 ที่ ATS ค้นหาได้มีดังนี้

$$N(s) = 0.00257s^5 + 23.36s^4 - 22.86s^3 - 45.27s^2 + 1.25 \times 10^4 s + 1053 \quad (1.18)$$

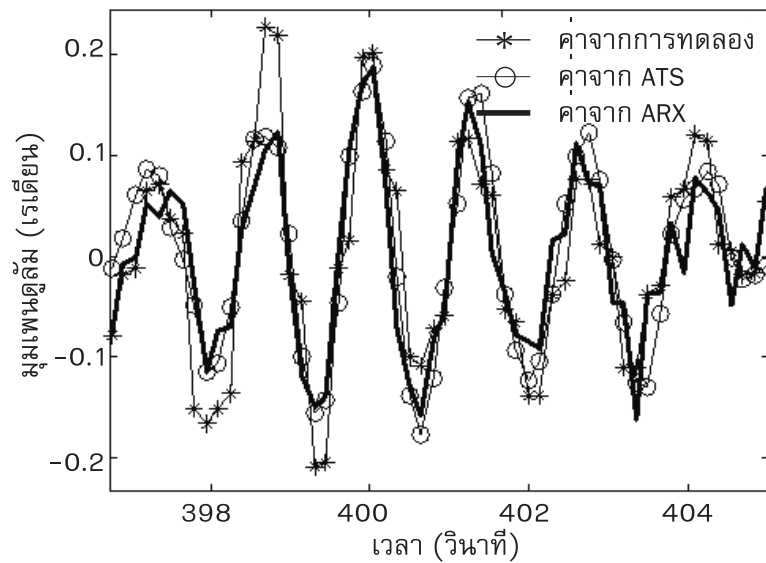
$$D(s) = 5.91s^5 + 35.06s^4 + 2579s^3 + 1.05 \times 10^4 s^2 + 3.78 \times 10^4 s + 8.53 \times 10^4 \quad (1.19)$$

ผลที่ได้จากแบบจำลองทั้งอันดับ 5 และ 10 แสดงความสอดคล้องกันกับข้อมูลที่วัดได้จริงจากระบบ เป็นการยืนยันประสิทธิภาพการค้นหาคำตอบของ ATS ได้เป็นอย่างดี

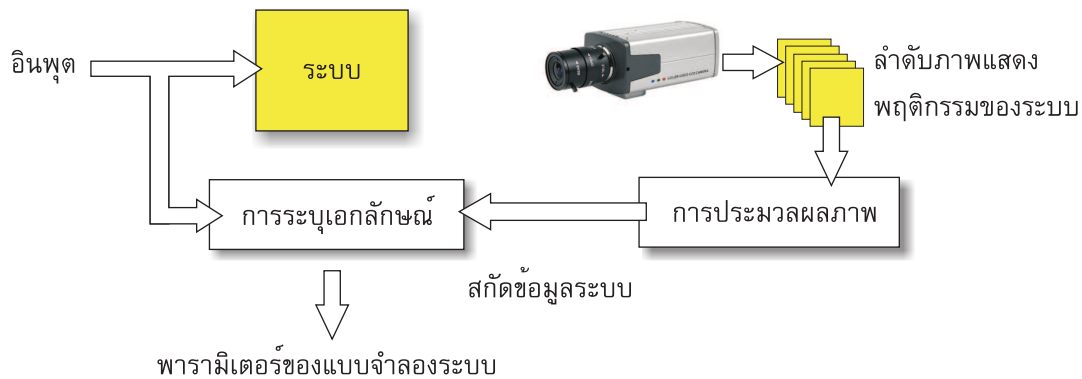
1.6.5 การระบุเอกลักษณ์ด้วยเทคนิคการประมวลผลภาพ

System Identification using Image Processing Technique

ในงานระบุเอกลักษณ์ด้วยวิธีดั้งเดิมจำเป็นจะต้องมีการติดตั้งตัวตรวจจับเข้ากับระบบ เพื่ออ่านข้อมูลสถานะของระบบมาใช้ในการระบุวิธีการระบุเอกลักษณ์ เช่นค่าผลตอบสนองของเอาต์พุตของระบบเป็นต้น ในบางโอกาส การติดตั้งตัวตรวจจับดังกล่าวอาจมีความยุ่งยาก งานใน [Sujitjorn, Srikaew, Pongdownreong, Attakitmongkol, and Totarong, 2003] ได้นำเสนอการเอาตัวตรวจจับภาพดิจิทัล (กล้องวิดีโอ) มาอ่านและสกัดข้อมูลออกจากระบบเพื่อใช้ในการระบุเอกลักษณ์ของระบบ ข้อดีของการใช้กล้องก็คือเราไม่จำเป็นต้องติดตั้งอุปกรณ์ใดๆ เข้ากับตัวระบบโดยตรง กล้องวิดีโอสามารถจับภาพของระบบที่มีการเปลี่ยนแปลงหรือเคลื่อนไหวที่มองเห็นได้ การ



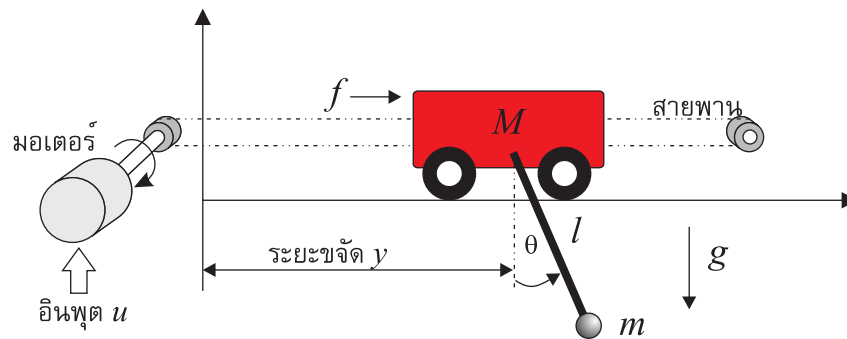
รูปที่ 1.38: มุมการแกว่งของเพนดูลัม (ข้อมูลจริงเทียบกับแบบจำลองอันดับ 5 จาก ARX และ ATS)



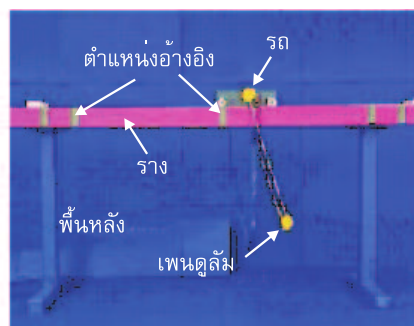
รูปที่ 1.39: การระบุเอกลักษณ์ด้วยการประมวลผลภาพ

จับภาพดังกล่าวสามารถติดตั้งกล้องห่างจากระบบได้ ตัวอย่างระบบในหัวข้อนี้จะเป็นการระบุเอกลักษณ์ระบบรถเพนดูลัมผกผันด้วยเทคนิคการประมวลผลภาพ แผนผังการทำงานทั่วไปของระบบแสดงในรูปที่ 1.39

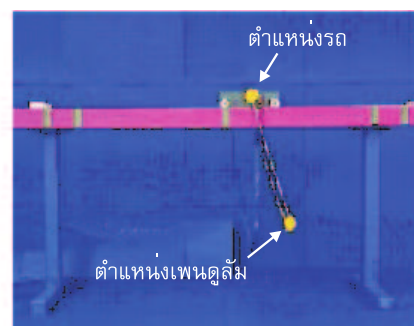
พิจารณาการระบุเอกลักษณ์ของระบบรถเพนดูลัมผกผันอีกครั้ง ในกรณีนี้การติดตั้งระบบมีความแตกต่างไปจากระบบในหัวข้อก่อนหน้านี้ รูปที่ 1.40 แสดงแผนผังระบบรถเพนดูลัมผกผัน ระบบเป็นชุดทดลองระบบรถเพนดูลัม โดยการขับเคลื่อนรถเพนดูลัมจะใช้มอเตอร์ในการขับเคลื่อนสายพานที่ยึดติดอยู่กับตัวรถ ดังนั้นอินพุตที่ใช้ควบคุมรถคือคำสั่งในการควบคุมมอเตอร์นั่นเอง เทคนิคการประมวลผลภาพในที่นี้ทำการสกัดข้อมูล "ตำแหน่ง" ของรถเพนดูลัมที่อยู่ในระนาบของภาพจากลำดับภาพ ความเร็วของการจับภาพเข้าสู่ระบบและการประมวลผลจะต้องเพียงพอที่จะทำให้ผลการคำนวณที่ได้ไม่มีการหน่วง (delay) ไปจากพฤติกรรมในเวลาจริงของระบบ ตัวอย่างการใช้เทคนิคตัดแยกองค์ประกอบต่างๆ ในระบบรถเพนดูลัมผกผันด้วยสี (color segmentation) แสดงในรูปที่ 1.41 โดยรูปที่ 1.41-(ก) แสดงการระบุสีขององค์ประกอบต่างๆ ในระบบ รูปที่ 1.41-(ข) แสดงภาพที่บันทึกได้และรูปที่ 1.41-(ค) แสดงผลการตัดแยกองค์ประกอบด้วยสีของตำแหน่งรถและลูกตุ้มเพนดูลัม และรูปที่ 1.41-(ง) แสดงแผนผังข้อมูล "ตำแหน่ง" ของระบบที่จะใช้ในการระบุเอกลักษณ์ต่อไป ในที่นี้ จะได้ทำการพิจารณาแบบจำลองของรถเพนดูลัมทั้งแบบเชิงเส้นและไม่เป็นเชิงเส้น ดังรายละเอียดต่อไปนี้



รูปที่ 1.40: ระบบรถเพนดูลัมผกผัน



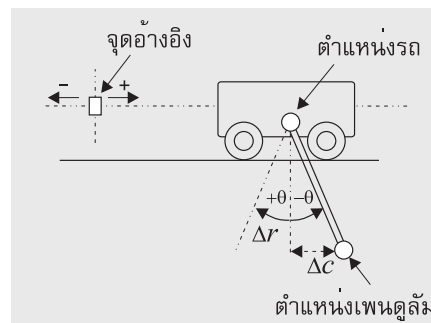
(ก)



(ข)



(ค)

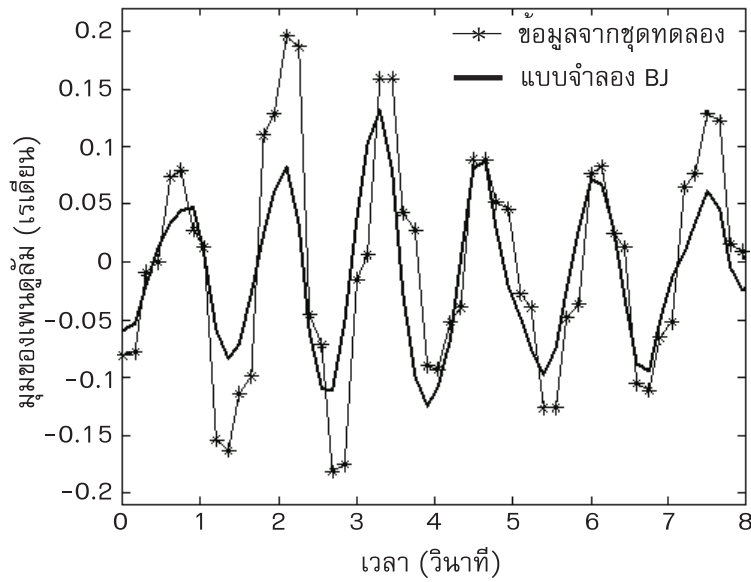


(ง)

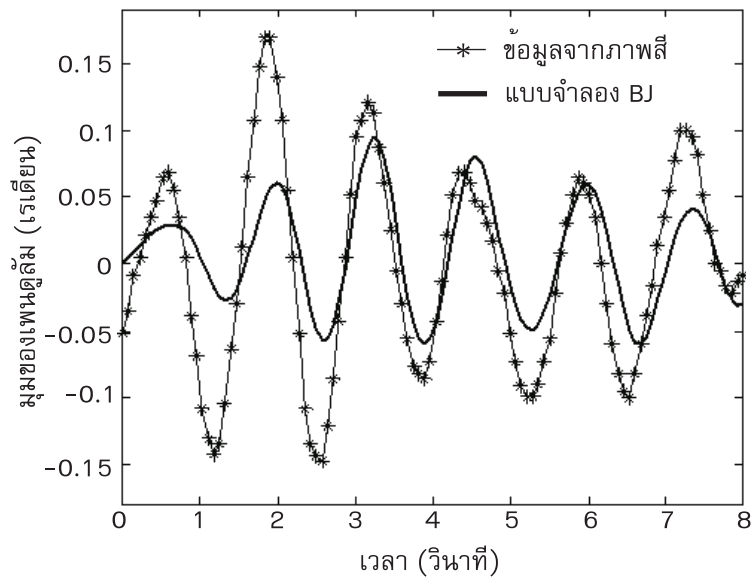
รูปที่ 1.41: โครงแบบการสกัดข้อมูลตำแหน่งรถและเพนดูลัมจากภาพสี

แบบจำลองเชิงเส้นของรถเพนดูลัม

แบบจำลองของรถเพนดูลัมข้างต้นสามารถถูกพิจารณาให้เป็นระบบเชิงเส้นได้ ถ้ามุมในการแกว่งของเพนดูลัมมีค่าไม่มาก (ประมาณ ± 0.2 เรเดียน) ในกรณีนี้แบบจำลองแบบ BJ (Box-Jenkins) ถูกใช้แทนแบบจำลองของระบบด้วยอันดับเท่ากับ 5 ซึ่งเพียงพอต่อการจำลองระบบที่เป็นเชิงเส้นนี้ รูปที่ 1.42 แสดงผลการจำลองระบบรถเพนดูลัม โดยในรูปที่ 1.42-(ก) ได้จากการใช้แบบจำลองแบบ BJ ด้วยข้อมูลจากชุดการทดลองรถเพนดูลัม (ใช้ตัวตรวจจับที่มากับชุดรถเพนดูลัม) ส่วนรูปที่ 1.42-(ข) ได้จากการใช้แบบจำลองแบบ BJ ด้วยข้อมูลจากการสกัดจากภาพสี ผลที่ได้แสดงให้เห็นในเบื้องต้นว่าการใช้ข้อมูลที่สกัดจากภาพสีมีความสอดคล้องกับระบบจริง ทำให้สามารถนำมาใช้ในการระบุเอกลักษณ์ได้ในระดับหนึ่ง ที่ซึ่งอาจจำเป็นต้องมีการชดเชยความผิดพลาดที่เกิดขึ้นในแบบจำลองเพิ่มเติม



(ก) แบบจำลองเชิงเส้นด้วยข้อมูลจากชุดทดลอง



(ข) แบบจำลองเชิงเส้นด้วยข้อมูลที่สกัดจากภาพสี

รูปที่ 1.42: ผลการจำลองระบบเชิงเส้นด้วยแบบจำลองแบบ BJ

แบบจำลองไม่เป็นเชิงเส้นของรถเพนดูลัม

ในกรณีที่เพนดูลัมแกว่งด้วยมุมที่มากขึ้น ระบบจะไม่เป็นเชิงเส้นอีกต่อไปและสามารถอธิบายได้ด้วยความสัมพันธ์ต่อไปนี้

$$ml\ddot{\theta} + m \cos \theta \ddot{y} + mg \sin \theta = 0 \quad (1.20)$$

$$(M + m)\ddot{y} + ml \cos \theta \ddot{\theta} - ml\dot{\theta}^2 \sin \theta = f \quad (1.21)$$

โดยที่ f คือแรงที่ขับเคลื่อนสายพานของตัวรถและควบคุมด้วยมอเตอร์ ดังนั้นเราสามารถพิจารณาได้ว่าเป็นการระบุเอกลักษณ์ของแรง f ซึ่งในที่นี้จะพิจารณาฟังก์ชันของ f เป็นพหุนามอันดับที่ 7 ดังสมการต่อไปนี้

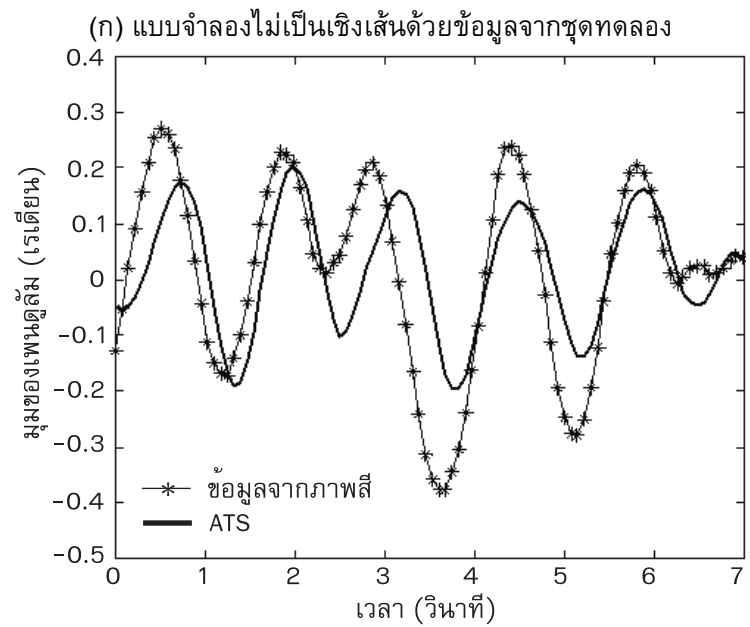
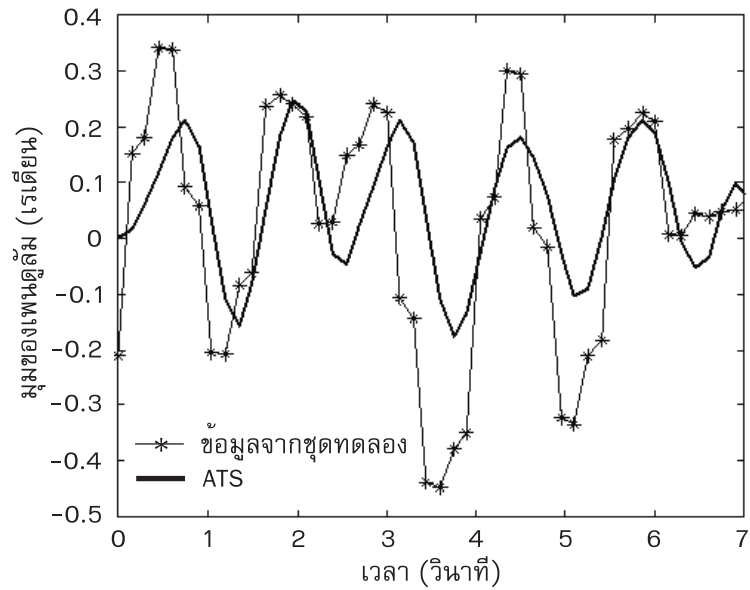
$$f = Au^7 + Bu^6 + Cu^5 + Du^4 + Eu^3 + Fu^2 + Gu + H \quad (1.22)$$

นั่นคือพารามิเตอร์ที่ต้องทำการระบุเอกลักษณ์คือ A, B, \dots, H ของฟังก์ชันแรง f เราจะเห็นได้ว่าวิธีการระบุเอกลักษณ์แบบดั้งเดิมนั้นไม่สามารถที่จะแก้ปัญหาดังกล่าวได้ ATS จึงถูกนำมาประยุกต์ใช้ในการหาพารามิเตอร์ของระบบ ผลการจำลองสถานการณ์ของแบบจำลองที่ได้แสดงในรูปที่ 1.43 โดยในรูปที่ 1.43-(ก) ใช้พารามิเตอร์ที่ค้นหาด้วย ATS โดยใช้ข้อมูลจากชุดการทดลองรถเพนดูลัม และรูปที่ 1.42-(ข) ใช้พารามิเตอร์ที่ค้นหาด้วย ATS โดยใช้ข้อมูลจากการสกัดจากภาพสี การประยุกต์ใช้ ATS ในการระบุเอกลักษณ์ เพื่อหาพารามิเตอร์ของระบบนั้นมีความยุ่งยากน้อยกว่าวิธีการระบุเอกลักษณ์แบบดั้งเดิม โดยเฉพาะเมื่อระบบมีความไม่เป็นเชิงเส้น สิ่งที่จะต้องพิจารณาอีกคือแบบจำลองคร่าวๆ ของระบบ พร้อมทั้งข้อมูลอินพุต/เอาต์พุต (ผลตอบสนองหรือพฤติกรรม) ของระบบ ทั้งนี้การคำนวณหาผลเฉลยรูปแบบปิดจึงไม่จำเป็น การใช้ ATS จึงเป็นที่ยอมรับในการนำมาประยุกต์ใช้ในงานระบุเอกลักษณ์กันอย่างกว้างขวาง

1.7 สรุป

TS เป็นการค้นหาคำตอบที่เหมาะสมที่สุดที่มีจุดเด่นหลากหลายประการ ความสามารถในการหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบ จนนำไปสู่คำตอบที่เหมาะสมที่สุดแบบวงกว้าง โครงสร้างการทำงานที่ง่ายแต่มีประสิทธิภาพ รวมไปถึงรูปแบบอื่นๆ ของ TS ที่มีจุดเด่นแตกต่างออกไปเช่น RTS หรือ ATS ทำให้ TS ได้รับความสนใจและเป็นที่ยอมรับในการนำไปประยุกต์ใช้งานด้านต่างๆ อย่างมากมาย TS ในปัจจุบันงานวิจัยการขยายอัลกอริทึมรูปแบบใหม่ๆ ของ TS ก็ยังได้รับความสนใจจากนักวิจัยอีกมากมาย การเพิ่มประสิทธิภาพในการค้นหาคำตอบของ TS จึงยังคงก้าวไปอย่างไม่ถูก ‘ต้องห้าม’





รูปที่ 1.43: ผลการจำลองระบบแบบไม่เชิงเส้นด้วยแบบจำลองจาก ATS

โจทย์คำถาม

- 1.1. จงออกแบบเขียนโปรแกรมการค้นหาค่าเหมาะที่สุดแบบตาบูซันดิคปรับตัวได้สำหรับค้นหาพารามิเตอร์ของตัวควบคุมพีไอดี K_P K_I และ K_D ของระบบที่มีฟังก์ชันถ่ายโอนดังนี้

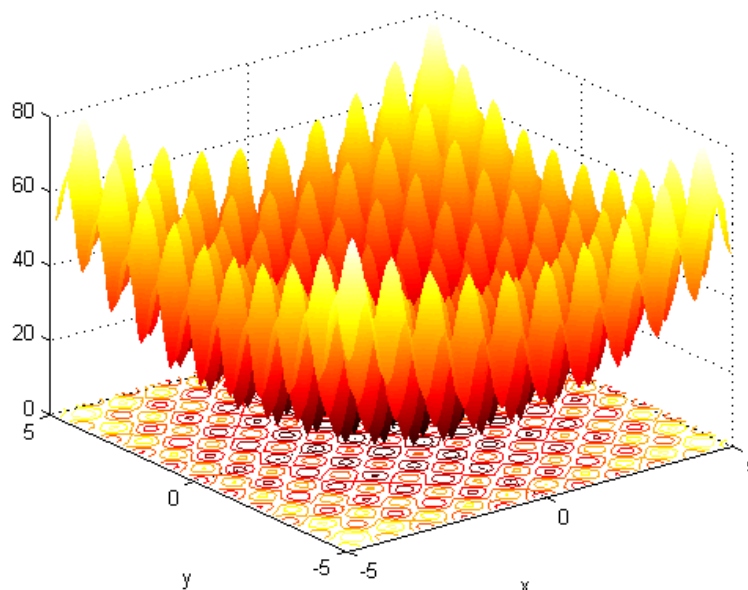
$$P_1(s) = \frac{1}{(1+s)(1+\frac{s}{6})^3}$$

ให้ใช้ฟังก์ชันวัตถุประสงค์ดังนี้

$$w_1(T_R, T_S, PO) = \alpha T_R + \beta T_S + \gamma PO$$

โดยให้แสดงผลการทดลองที่ได้และนำไปเปรียบเทียบกับวิธีการคำนวณหาพารามิเตอร์ของตัวควบคุมพีไอดีแบบดั้งเดิมคือวิธีซีกเลอร์นิโคลพร้อมทั้งทดลองปรับค่าสัมประสิทธิ์ในการกำหนดค่านัยสำคัญของ α β และ γ เพื่อให้ได้ผลตอบสนองของระบบที่ดีขึ้น แสดงผลเปรียบเทียบกับผลที่ได้จากตัวอย่างในหัวข้อก่อนหน้านี้ อธิบายรายละเอียดการออกแบบ วิเคราะห์และสรุปผลการออกแบบ (สังเกตว่าค่าของพารามิเตอร์แต่ละชุดในฟังก์ชันวัตถุประสงค์มีหน่วยที่ไม่เหมือนกัน การออกแบบจึงควรต้องระวังในเรื่องการปรับให้เป็นบรรทัดฐานเดียวกัน)

- 1.2. พิจารณาฟังก์ชันของ Rastrigin $f(x, y) = 20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)$ ในรูปที่ 1.44 ที่ซึ่งประกอบไปด้วยจุดต่ำสุดทั้งแบบวงแคบเฉพาะถิ่น (local minima) และแบบวงกว้าง (global minima) จงออกแบบใช้ TS ในการหาค่าต่ำสุดแบบวงกว้างของฟังก์ชันดังกล่าว พร้อมทั้งหาพารามิเตอร์ที่เหมาะสมของ TS อธิบายผลลัพธ์ที่ได้



รูปที่ 1.44: ฟังก์ชัน Rastrigin

- 1.3. ATS สามารถนำไปใช้ในกระบวนการฝึกสอนเครือข่ายประสาทเทียมได้ จงอธิบายการนำเอา ATS ไปใช้ในการหาโครงสร้างที่เหมาะสมที่สุดของเครือข่ายประสาทเทียม และทำการจำลองการทำงานโดยพิจารณาใช้ ATS ในการค้นหาค่าน้ำหนักประสาของเครือข่ายเพอร์เซพตรอน (perceptron) ขนาด 2 นิวรอนสำหรับปัญหา XOR (ดูรายละเอียดได้ในภาคเครือข่ายประสาทเทียม)

1.4. จากการค้นหาค่าเหมาะที่สุดของฟังก์ชัน peaks ให้ออกแบบประยุกต์ใช้ ATS ในการค้นหาคำตอบแทน TS พร้อมทั้งเปรียบเทียบผลลัพธ์ที่ได้

1.5. พิจารณาแบบจำลองสำหรับท่ออากาศแบบร้อน (hot-air tube) ซึ่งมีแบบจำลองอันดับ 1 ที่มีตัวหน่วง ในรูปต่อไปนี้

$$G(s) = \frac{Ke^{-\tau_d s}}{\tau s + 1}$$

โดยที่ K คืออัตราขยายของระบบ τ คือค่าคงที่เวลา (วินาที) และ τ_d คือเวลาหน่วง (วินาที) แบบจำลองข้างต้นคือการประมาณของ Pade จากข้อมูลในชุดการทดลอง พารามิเตอร์ของระบบถูกค้นหาค้นหาด้วย AT-S ได้ผลลัพธ์ดังนี้

$$K = 0.8645$$

$$\tau = 0.3337 \text{ วินาที}$$

$$\tau_d = 0.2042 \text{ วินาที}$$

- ให้ใช้ค่าพารามิเตอร์ดังกล่าว ทำการจำลองระบบเพื่อหาค่าเอาต์พุตผลตอบสนองของระบบ
- จากเอาต์พุตที่ได้ ให้ทำการใส่สัญญาณรบกวนรูปแบบต่างๆ แล้วนำเอาผลลัพธ์ที่ได้มาทำการค้นหาแบบจำลองของระบบด้วย ATS อีกครั้ง พร้อมทั้งอภิปรายประสิทธิภาพการทำงานของ ATS ต่อสัญญาณรบกวน



- K-N. Areerak and S. Sujitjorn. Performance comparison between genetic algorithm and tabu search methods. *Suranaree J. Sci. Technol.*, 9:61--68, 2002.
- R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126--140, 1994.
- F. Glover. Tabu search - part i. *ORSA Journal on Computing*, 1(3):190--206, 1989.
- F. Glover. Tabu search - part ii. *ORSA Journal on Computing*, 2(1):4 -- 32, 1990.
- F. Glover and M Laguna. *Handbook of Applied Optimization*, chapter Tabu Search. Oxford Academic Press, 1980.
- F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.
- D. Puangdownreong, K-N. Areerak, A. Srikaew, S. Sujitjorn, and P. Totarong. System identification via adaptive tabu search. In *Proceedings IEEE International Conference on Industrial Technology (ICIT02)*, pages 915--920, 2002.
- S. Sujitjorn, A. Srikaew, P. Pongdownreong, K. Attakitmongcol, and P. Totarong. Model identification using image processing technique. In *International Conference on Computer, Communication and Control Technologies*, pages 530--535, Florida, USA, August 2003.



A handwritten signature in grey ink, consisting of stylized, overlapping horizontal and vertical strokes. To the right of the signature is a circular symbol with a dot in the center, resembling an '@' symbol.