

## การหาผลเฉลยโดยวิธี Tabu search

ศุภชัย ปทุมนากุล<sup>1</sup> และ วนิดา รัตนมณี<sup>2</sup>

### Abstract

Pathumnakul, S.<sup>1</sup> and Ratanamane, W.<sup>2</sup>

Finding solution by Tabu search

Songklanakar J. Sci. Technol., 2003, 25(6) : 799-805

In this paper, the method of finding the solution by Tabu search is addressed. Tabu search is one of the local search heuristics, which has been applied extensively in operations research works. In this paper, the basic concepts of the Tabu search heuristic and some examples of applying this search technique in operations research problems are presented.

**Key words :** Tabu search, heuristic

<sup>1</sup>Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002 Thailand. <sup>2</sup>Department of Industrial Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla 90110 Thailand.

<sup>1</sup>Ph.D.(Industrial Engineering), ผู้ช่วยศาสตราจารย์, ภาควิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น อำเภอเมือง จังหวัดขอนแก่น 40002 <sup>2</sup>M.S.(Industrial Engineering), ผู้ช่วยศาสตราจารย์, ภาควิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อำเภอหาดใหญ่ จังหวัดสงขลา 90110

Corresponding e-mail: kwanida@ratree.psu.ac.th

รับต้นฉบับ 22 พฤศจิกายน 2545      รับลงพิมพ์ 22 กรกฎาคม 2546

## บทคัดย่อ

ศุภชัย ปทุมนากุล และ วนิดา รัตนมณี  
การหาผลเฉลยโดยวิธี Tabu search

ว. สงขลานครินทร์ วทท. 2546 25(6) : 799-805

บทความนี้จะนำเสนอการหาผลเฉลย โดยวิธี Tabu search ที่ได้ถูกนำมาใช้อย่างแพร่หลายในการหาผลเฉลยทางการวิจัยดำเนินงาน (Operations research) โดยถือว่าเป็นวิธีการค้นหาคำตอบแบบวิธีฮิวริสติก (Heuristic) แบบหนึ่ง ซึ่งในบทความนี้จะกล่าวถึงหลักการพื้นฐานของวิธี Tabu search และตัวอย่างการประยุกต์ใช้งานในปัญหาต่างๆ ในด้านการวิจัยดำเนินงาน

ในการหาผลเฉลยในปัญหาต่างๆ ที่มีรูปแบบปัญหาที่ไม่เป็นโพลีโนเมียล (NP-Problem) เช่น ปัญหาการกำหนดเส้นทางการขนส่งของรถบรรทุก (vehicle routing problem) หรือปัญหาการจัดตารางการผลิต (production scheduling) นั้นเป็นการยากที่จะหาผลเฉลยที่เหมาะสมที่สุด (optimal solution) โดยเฉพาะอย่างยิ่งเมื่อปัญหานั้นมีขนาดใหญ่ การหาผลเฉลยที่เหมาะสมที่สุดอาจใช้เวลาการคำนวณหาผลเฉลยนานมาก หรือก็เป็นไปไม่ได้เลยที่จะหาผลเฉลยที่เหมาะสมที่สุดได้ ซึ่งวิธีการหาผลเฉลยที่เหมาะสมที่สุด หรือ exact algorithm เช่น วิธี linear programming, integer programming, branch and bound หรือ dynamic programming นั้นมักจะใช้ในการคำนวณผลเฉลยที่เหมาะสมที่สุดในปัญหาขนาดใหญ่ไม่ได้ แต่ในความเป็นจริงปัญหาต่างๆ ในอุตสาหกรรมจัดอยู่ในประเภทปัญหาขนาดใหญ่ เช่น การจัดเส้นทางการขนส่งไปยังลูกค้า 30-40 ราย ต่อหนึ่งเที่ยวของการขนส่ง หรือการจัดตารางการผลิตของงาน 40 งานเข้าสู่เครื่องจักร การที่จะหาผลเฉลยที่เหมาะสมที่สุดจาก วิธี exact algorithm นั้นแทบที่จะเป็นไปไม่ได้เลย ด้วยเหตุนี้การหาผลเฉลยต่อปัญหาต่างๆ นั้นสามารถที่จะกระทำได้โดยการใช้วิธีการทางฮิวริสติก (heuristic) วิธีการทางฮิวริสติกจะถูกพัฒนาขึ้นตามลักษณะของปัญหาที่ต้องการหาผลเฉลย วิธีการทางฮิวริสติกนี้จะไม่รับประกันว่าผลเฉลยที่ได้จะเป็นผลเฉลยที่เหมาะสมที่สุด แต่จะเป็นผลเฉลยที่สามารถยอมรับได้ แต่เวลาที่ใช้ในการคำนวณหาผลเฉลยนั้นจะน้อยกว่าวิธี exact algorithm มาก และสามารถที่จะคำนวณหาผลเฉลยของปัญหาขนาดใหญ่ได้ ปัญหาต่างๆ ในอุตสาหกรรมนั้นจึงดูจะเหมาะสมในการหา

ผลเฉลยโดยวิธีการทางฮิวริสติกมากกว่าการใช้วิธี exact algorithm ในการหาผลเฉลย

การพัฒนาการหาผลเฉลยโดยวิธีการทางฮิวริสติกนั้นมีหลายแบบขึ้นอยู่กับลักษณะปัญหา ซึ่งมีทั้งการประยุกต์วิธี exact algorithm เข้าช่วย หรือเป็นวิธีที่ไม่เกี่ยวข้องใดๆ เลยกับวิธี exact algorithm เช่น วิธี Clarke & Wright algorithm (Clarke and Wright, 1964) ที่ใช้ในการแก้ปัญหาการกำหนดเส้นทางการเดินทาง (vehicle routing problem) ซึ่งใช้การหาระยะทางที่ประหยัดได้ในการรวมเส้นทาง 2 เส้นทางเข้าด้วยกันเป็นตัวกำหนดตำแหน่งที่รถจะเดินทางต่อไป

ในปัจจุบันมีวิธีการทางฮิวริสติกหลายๆ วิธีที่ได้ถูกนำมาประยุกต์ใช้กับปัญหาต่างๆ อย่างแพร่หลาย เช่น วิธี simulated annealing, genetic algorithm และ Tabu search ซึ่งทั้ง 3 วิธีนี้เป็นวิธีการค้นหาคำตอบหรือผลเฉลยของปัญหาที่คิดว่าเป็นผลเฉลยที่เหมาะสม แต่ไม่สามารถรับประกันว่าผลเฉลยที่ได้นั้นเป็นผลเฉลยที่เหมาะสมที่สุด ดังนั้นวิธีการค้นหาผลเฉลยดังกล่าวข้างต้น จึงถือว่าเป็นการค้นหาคำตอบจากคำตอบที่เป็นไปได้เฉพาะที่ (local search)

ในบทความนี้จะกล่าวถึงหลักการเบื้องต้นของการใช้วิธี Tabu search ในการหาผลเฉลยที่เหมาะสม โดยใช้หลักการของวิธี Tabu search ซึ่งหลักการนี้ได้รับการตีพิมพ์เผยแพร่ในเอกสารทางวิชาการหลายฉบับ เช่น ในงานของ Glover (Glover, 1990 และ Glover, 1993) โดยจะกล่าวถึงหลักการต่างๆ ไป และการนำมาประยุกต์ใช้ในการหาผลเฉลยของปัญหาต่างๆ

### หลักการเบื้องต้นของวิธี Tabu search

คำว่า Tabu หรือ Taboo นั้นหมายถึงข้อห้าม หรือ การห้าม ดังนั้น Tabu search จึงหมายถึงวิธีการค้นหาที่มีข้อห้าม หรือการห้ามการค้นหาคำตอบในบางขอบเขต ในวิธีการห้ามดังกล่าวนั้นจะเป็นการห้ามเพื่อที่จะช่วยให้ไม่ต้องไปค้นหาผลเฉลยเดิม หรือหลงในวัฏจักร (cyclic) การค้นหาคำตอบซึ่งอยู่ในขอบเขตของการค้นหาเดิม ซึ่งจะส่งผลให้ไม่สามารถหาคำตอบที่ดีขึ้นได้ เพื่อช่วยต่อความเข้าใจ ให้พิจารณาการเดินทางของนักสำรวจ สมมติว่ามีนักสำรวจคนหนึ่งต้องการสำรวจเทือกเขาซึ่งมีหุบเขาอยู่มากมาย โดยจะสำรวจหาหุบเขาที่ลึกที่สุดว่าเป็นหุบเขาใด และลึกประมาณกี่เมตร โดยการสำรวจนั้นจะทำการเดินทางและนับจำนวนก้าวที่เดิน ซึ่งเทือกเขานี้เป็นไปตาม Figure 1 จากรูปจะเห็นได้ว่าจุดที่ลึกที่สุดคือจุด c ซึ่งถือว่าเป็นผลเฉลยที่เหมาะสมที่สุด (optimal solution) ของปัญหานี้ หรืออาจกล่าวอีกลักษณะหนึ่งได้ว่าจุด c เป็นจุดที่ต่ำที่สุดของเทือกเขาทั้งหมดหรือที่เรียกว่า global minimum ในขณะที่จุด a, b และ d เป็นจุดที่ต่ำที่สุดในบริเวณเทือกเขาช่วงต่างๆ หรือที่เรียกว่า local minimum ถ้าหากนักสำรวจผู้นั้นตั้งกฎเกณฑ์การเดินทางว่าจะเดินไปยังพื้นที่ต่ำลงไปเรื่อยๆ เท่านั้น นักสำรวจนั้นก็จะได้เดินไปยังจุด a แล้วก็จะหยุด เพราะคิดว่าจุด a นั้นเป็นหุบเขาที่ต่ำที่สุดของเทือกเขานี้ ซึ่งถือว่าการสำรวจหรือการค้นหาผลเฉลยนี้ติดอยู่ที่ local minimum หรือในกรณีที่นักสำรวจที่ยอมเดินขึ้นเขาต่ออีกระยะหนึ่ง เช่น เดินไป 1 ก้าว (สมมติว่าอยู่ที่จุด z) แต่นักสำรวจยังถือหลักที่ว่าเดินไปยังที่ต่ำกว่าเสมอ นักสำรวจก็จะเดินย้อนกลับมายังจุด a อีกครั้ง

และถ้าหากยอมเดินขึ้นเขาอีกหนึ่งก้าวก็จะกลับไปยังจุด z และกลับมายังจุด a อีก ซึ่งในที่สุดแล้วนักสำรวจผู้นั้นก็จะยอมรับว่าจุด a เป็นจุดต่ำสุดของเทือกเขา แสดงว่านักสำรวจนั้นหลงวนเวียน (trapping) อยู่ในขอบเขตของผลเฉลยเดิมเท่านั้น

จากลักษณะดังกล่าวของปัญหานี้ การที่นักสำรวจจะสามารถหาผลเฉลย หรือจุดต่ำสุดของเทือกเขาที่แท้จริงนั้น นักสำรวจจะต้องเดินต่อไปแม้จะเป็นตำแหน่งที่สูงขึ้นไปเรื่อยๆ รวมทั้งยังตั้งกฎเกณฑ์ว่าจะไม่เดินกลับที่เดิมแม้จะเป็นตำแหน่งที่ต่ำกว่าก็ตาม นักเดินเขาจึงจะมีโอกาสสำรวจพบตำแหน่งหุบเขา b และ c ในที่สุด การห้ามเดินกลับมายังจุดเดิมแม้จะเป็นตำแหน่งที่ต่ำกว่านั้นเป็นการป้องกันการหลงวนเวียนอยู่ในผลเฉลยเดิมๆ ของปัญหา และการห้ามดังกล่าวเป็นหลักการหนึ่งของวิธี Tabu search คือการห้ามให้มีการค้นหาผลเฉลยจากขอบเขตหรือพื้นที่เดิมๆ ที่เพิ่งจะมีการค้นหามา เพื่อป้องกันการหลงวนเวียน แต่การห้ามดังกล่าวไม่ได้ห้ามตลอดไป แต่จะห้ามไว้เพียงระยะหนึ่งเท่านั้น ซึ่งจะกล่าวถึงโดยละเอียดในช่วงต่อไป ส่วนในด้านการเดินทางต่อไปเรื่อยๆ นั้น ถ้าไม่มีข้อกำหนดใดๆ แก่การเดินทางของนักสำรวจ หมายถึงนักสำรวจจะทำการเดินทางต่อไปเรื่อยๆ เพื่อสำรวจความลึกของหุบเขา ซึ่งเท่ากับค้นหาผลเฉลยที่เหมาะสมที่สุด ถ้าหากมีหุบเขาทั้งหมด 10,000 หุบเขาในเทือกเขานี้ นักสำรวจผู้นั้นอาจจะใช้เวลาทั้งชีวิตในการหาคำตอบ ซึ่งสามารถเปรียบเทียบได้กับเวลาในการคำนวณของการหาผลเฉลยที่เหมาะสมที่สุด โดยวิธีทาง exact algorithm ในปัญหาที่มีขนาดใหญ่ดังที่กล่าวข้างต้น ดังนั้นเพื่อไม่เป็นการที่ต้องเดินทางสำรวจที่ยาวนานเกินไป นัก

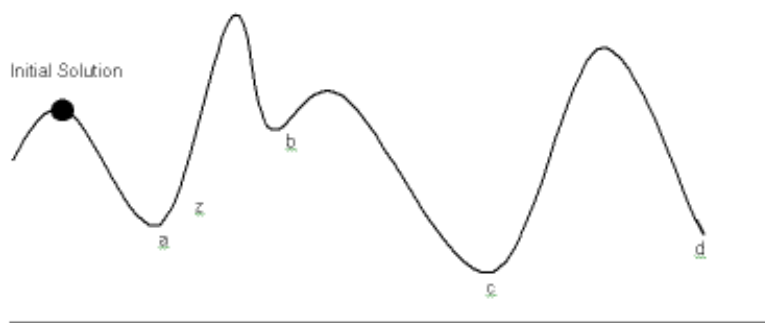


Figure 1 An example of searching the lowest point.

สำรวจอาจสามารถตั้งกฎเกณฑ์การสำรวจให้กับตัวเองไว้ เช่น การเดินทางนั้นจะเดินขึ้นไม่เกิน 100 ก้าว ถ้าใน 100 ก้าวยังไม่พบหนทางลงเลย จะทำการหยุดเดินและหุบเขาที่พบว่าต่ำที่สุดจนถึงเวลาหยุดเดินจะเป็นหุบเขาที่ต่ำที่สุด หรือผลเฉลยที่ได้ การกำหนดเกณฑ์การหยุดการค้นหา (stopping criteria) นั้นในวิธีการหาผลเฉลยแบบฮิวริสติกนั้นมีได้หลายวิธีซึ่งจะได้กล่าวถึงในตอนต่อไป การกำหนดเกณฑ์การหยุดการค้นหาดังกล่าวนั้นส่งผลให้ผลเฉลยที่ได้ไม่จำเป็นที่จะต้องเป็นผลเฉลยที่เหมาะสมที่สุดได้ แต่จะเป็นผลเฉลยที่พอจะยอมรับได้ หรือถ้าโชคดีก็จะเป็นผลเฉลยที่เหมาะสมที่สุด และใช้เวลาในการค้นหาผลเฉลยน้อย หรือไม่มากเท่ากับวิธี exact algorithm ในหัวข้อต่อไปจะได้กล่าวถึงองค์ประกอบเบื้องต้นของวิธี Tabu search และตัวอย่างง่าย ๆ ในการนำวิธี Tabu search ไปใช้

#### องค์ประกอบหลักของวิธี Tabu search

ในหัวข้อนี้จะกล่าวถึงองค์ประกอบหลักในวิธี Tabu search โดยจะเปรียบเทียบกับตัวอย่างของนักสำรวจในการเดินเพื่อสำรวจหุบเขาซึ่งได้ยกตัวอย่างในหัวข้อที่ผ่านมา หรือตัวอย่างอื่นที่สามารถอธิบายให้เข้าใจได้มากกว่า ตัวอย่างนักสำรวจในบางองค์ประกอบ

Tabu search ประกอบไปด้วยด้วยรูปแบบการค้นหาคำตอบ 2 รูปแบบที่สำคัญคือ การค้นหาผลเฉลยโดยการใช้ความจำระยะสั้น (short-term memory search) และการค้นหาผลเฉลยโดยการใช้ความจำระยะยาว (long-term memory search) ซึ่งการค้นหาผลเฉลยโดยการใช้ความจำระยะสั้นนั้นเป็นหลักการสำคัญและเป็นหลักการเบื้องต้นของวิธี Tabu search การใช้ความจำระยะสั้นหมายถึงการค้นหาที่จดจำอดีตหรือประสบการณ์การค้นหาที่ผ่านมาเพียงระยะสั้น เช่น ในกรณีนักสำรวจเพื่อการค้นหาที่มีประสิทธิภาพอาจจะจดจำตำแหน่งที่เคยสำรวจมาแล้วเพียง 100 ก้าวที่ผ่านมาเท่านั้น พอเดินไปที่ก้าวที่ 101 ตำแหน่งการก้าวของก้าวที่ก้าวที่ 1 ก็จะไม่จดจำเป็นประสบการณ์อีก ในทางตรงกันข้ามกับการใช้ความจำระยะยาวนั้นจะต้องจดจำอดีตหรือประสบการณ์ที่ผ่านมาตลอดเพื่อช่วยให้การค้นหาคำตอบเป็นไปอย่างมีประสิทธิภาพมากขึ้น เช่น ตัวอย่างของนักสำรวจ นักสำรวจจะต้องจดจำตำแหน่งหรือประสบการณ์การเดินทางที่ผ่านมาให้มากที่สุด

ในบทความนี้จะกล่าวถึงเฉพาะองค์ประกอบในการค้นหาผลเฉลยโดยการใช้ความจำระยะสั้น ซึ่งมีองค์ประกอบหลักๆ คือ cost function, Tabu list type, Tabu list size, aspiration criteria, move และ stopping criteria ซึ่งสามารถที่จะอธิบายในรายละเอียดแต่ละองค์ประกอบได้ดังนี้

**Cost function** เป็นฟังก์ชันของต้นทุนเพื่อใช้ประกอบการตัดสินใจในการค้นหาคำตอบว่าจะค้นหาคำตอบหรือผลเฉลยไปในทิศทางใด หรือขอบเขตใดๆ หรืออาจจะกล่าวได้ว่าเป็นตัวชี้วัดการค้นหาผลเฉลย ถ้าเปรียบเทียบกับตัวอย่างนักสำรวจที่เดินสำรวจหาหุบเขาที่ลึกที่สุดนั้น Cost function อาจจะหมายถึงความลึกที่ได้เพิ่มขึ้น หรือลดลงจากการก้าวเดินก้าวนี้ ถ้าก้าวที่นักสำรวจกำลังที่จะก้าวต่อไปทำให้ได้ความลึกเพิ่มขึ้น เพราะก้าวลงไปในที่ที่ต่ำกว่าที่ที่ยืนอยู่ในปัจจุบัน จะได้ต้นทุน (cost) ที่ลดลงในทำนองเดียวกัน ถ้าการก้าวเท้าก้าวถัดไปของนักสำรวจเป็นการเดินขึ้นไปยังที่ที่สูงกว่าที่ที่ยืนอยู่ในปัจจุบัน จะได้ต้นทุนที่สูงขึ้น ดังนั้น cost function จึงเป็นตัวประเมินการค้นหาผลเฉลยว่าผลเฉลยที่ค้นหาอยู่ในแต่ละขั้นตอนนั้นมีประสิทธิภาพเพียงใด

**Tabu list type** หรือ Tabu type หรือ Tabu restriction นั้นเป็นรูปแบบการควบคุมการย้อนกลับ (reverse) หรือการหลงในวัฏจักร (cyclic) อยู่ในขอบเขตการค้นหาเดิมๆ หรือผลเฉลยเดิมที่มีอยู่ การกำหนด Tabu list type นั้นขึ้นอยู่กับแต่ละปัญหา ไม่จำเป็นต้องเหมือนกัน ในตัวอย่างของเราเพื่อป้องกันการย้อนกลับของการค้นหาคำตอบ Tabu list type อาจถูกกำหนดไว้ว่า ไม่ให้มีการเดินกลับมายังตำแหน่งเดิมเป็นต้น

**Tabu list size** เป็นตัวกำหนดจำนวนครั้ง หรือจำนวนที่ทำซ้ำ (iteration) ที่ต้องการห้ามไม่ให้มีการก้าวกลับมาที่เดิม เพราะการที่เรากำหนดว่า Tabu type ว่าห้ามไม่ให้มีการเดินกลับมาตำแหน่งเดิมนั้นตลอดไป เราอาจที่จะพลาดโอกาสการหาผลเฉลยที่ดีก็ได้ สำหรับหัวข้อนี้ ตัวอย่างนักสำรวจอาจจะเป็นที่เข้าใจได้ยาก จะขอยกตัวอย่างการจัดงานเข้าสู่เครื่องจักร 1 เครื่อง สมมติว่ามีงาน 5 งาน คือ งาน J1, J2, J3, J4 และ J5 และการจัดลำดับงานที่เหมาะสมที่สุดที่จะได้ต้นทุนที่ประหยัดที่สุดคือจัดงาน J1 ก่อนแล้วตามด้วย J3, J4, J2 แล้ว J5 ตามลำดับ

แต่ถ้าเราเริ่มต้นค้นหาคำตอบโดยจัดงานให้ J1 เข้าสู่เครื่องจักรก่อนตามด้วย J2, J3, J4 และ J5 ตามลำดับ (เหมือนกับตำแหน่งเริ่มต้นของนักสำรวจ) จะได้ต้นทุนการผลิตด้วยสาเหตุใดก็แล้วแต่เท่ากับ 500 บาท ถ้าทำการค้นหาผลเฉลยใหม่ด้วยการสลับงานที่เข้า เช่น สลับ J1 และ J2 ผลเฉลยที่ได้คือ J2, J1, J3, J4 และ J5 ทำให้ได้ต้นทุนการผลิต 520 บาท เพื่อป้องกันการย้อนกลับหรือหลงวนเวียนของการค้นหาผลเฉลยอยู่ในขอบเขตเดิมๆ เราจึงกำหนด Tabu list type ว่าจะไม่ทำการสลับงานที่สลับไปแล้ว ดังนั้นเราจะไม่สลับ J1 และ J2 อีก สมมติต่อไปอีกว่าการค้นหาคำตอบในอีกสองครั้งถัดมาเราสลับงาน J1 และ J3 แล้วตามด้วยการสลับ J4 และ J1 ผลเฉลยทั้งสองครั้งจะเป็น J2, J3, J1, J4, J5 และ J2, J3, J4, J1, J5 ตามลำดับ ถ้าเราห้ามการสลับงาน J1 และ J2 ตลอดไป เราจะพลาดโอกาส การได้คำตอบ J1, J3, J4, J2, J5 ซึ่งเป็นคำตอบที่เหมาะสมที่สุดตลอดไป หรืออาจต้องใช้เวลานานในการค้นหาผลเฉลยนี้ยาวนานมากขึ้น

ดังนั้นการกำหนด Tabu list size จึงเป็นการกำหนดว่าจะห้ามเงื่อนไขที่เป็น Tabu list type อยู่นานเท่าไร (จำนวนที่ซ้ำ) เช่น กำหนดไว้ที่ 2 ถ้าจำนวนการซ้ำผ่านไปแล้ว 2 ครั้ง ก็สามารถสลับ J1 กับ J2 ได้อีก ซึ่งจะทำให้เรามีโอกาสที่จะได้ผลเฉลยที่ดีที่สุด การกำหนด Tabu list size นี้ไม่มีกฎเกณฑ์ตายตัวว่าจะเป็นเท่าไร ซึ่งการกำหนดสูงเกินไปจะทำให้พลาดโอกาสที่จะได้คำตอบที่ดีที่สุด ในขณะที่กำหนดน้อยเกินไปอาจจะทำให้ได้คำตอบที่ย้อนกลับหรือหลงวนเวียนในขอบเขตผลเฉลยเดิมๆ ได้

**Aspiration criteria** เป็นรูปแบบการควบคุมที่ช่วยในการค้นหาคำตอบที่ดีที่สุดเป็นไปได้ด้วยดี และรวดเร็วมากขึ้น โดยกำหนดเป็นเงื่อนไขสำหรับการค้นหาที่ยังเป็น Tabu หรือตกอยู่ในแดนต้องห้าม เช่น จากตัวอย่างการจัดงานที่ได้กล่าวถึงในหัวข้อ Tabu list size นั้น ถ้าเรากำหนด Tabu list size ไว้ที่ 3 นั้นหมายถึงว่า J1 และ J2 ห้ามสลับกันอีกภายใน 3 iterations หลังจากที่ทำกรสลับกันก่อนหน้านั้น ด้วยเหตุนี้เราจะไม่สามารถค้นหาผลเฉลย J1, J3, J4, J2, J5 ได้ในขั้นนี้ แต่เพื่อเป็นการช่วยในการค้นหา เราอาจจะตั้งเงื่อนไขไว้ว่าถ้าการสลับคู่คู่ที่ยังเป็น Tabu หรือถูกห้ามสลับอยู่นั้นทำให้ได้ผลเฉลยที่เหมาะสมที่สุด

เท่าที่เคยได้ตั้งแต่การเริ่มต้นหาจนถึงปัจจุบัน เราสามารถยกเลิกการเป็น Tabu ของการสลับคู่คู่นั้นได้ ลักษณะเช่นนี้เรียกว่า การกำหนด aspiration criteria การกำหนด aspiration criteria ก็ไม่มีกฎเกณฑ์ที่ตายตัว ขึ้นอยู่กับลักษณะของปัญหาเป็นหลัก

**Move** เป็นการเปลี่ยนจากตำแหน่งหรือคำตอบหนึ่งไปเป็นอีกคำตอบหนึ่ง หรือเป็นการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่ง ในตัวอย่างการเดินทางนั้นอาจหมายถึงการตัดสินใจเดินขึ้นไปอีกหนึ่งก้าวหลังจากใคร่ครวญดีแล้ว เป็นต้น

**Stopping criteria** เป็นเงื่อนไขที่ใช้สำหรับการหยุดค้นหาคำตอบ ดังที่กล่าวข้างต้นถ้าไม่มีการกำหนดเงื่อนไขการหยุดค้นหา การค้นหาผลเฉลยก็จะดำเนินไปไม่หยุด ซึ่งจะต้องใช้เวลาในการคำนวณหรือค้นหามากเกินไปในปัญหาที่มีขนาดใหญ่ การกำหนดเงื่อนไขการหยุดค้นหานั้นมีหลายวิธี และขึ้นอยู่กับลักษณะของปัญหาที่ต้องการผลเฉลยด้วย ในที่นี้จะยกตัวอย่าง 2 วิธี

1. วิธีการกำหนดจำนวนครั้งของการทำซ้ำของการค้นหา เช่น กำหนดว่าจะค้นหาผลเฉลยไม่เกิน 100 ครั้ง ในตัวอย่างนักสำรวจอาจหมายถึงการสำรวจไม่เกิน 100 ก้าว เมื่อเดินครบแล้วตำแหน่งที่ต่ำสุดของ 100 ก้าวที่เดินผ่านมาคือผลเฉลยที่ได้จากการค้นหา

2. วิธีการกำหนดจำนวนครั้งการทำซ้ำเพื่อไว้ เช่น กำหนดว่าเมื่อพบคำตอบที่เหมาะสมที่สุดเท่าที่ค้นหา มาจะค้นหาเพื่อต่อไปอีก 20 iterations ถ้าในจำนวน 20 iterations นี้ ไม่มีคำตอบที่ดีกว่าคำตอบที่เหมาะสมที่สุดเท่าที่ค้นหา มาจะหยุดการค้นหา และคำตอบที่เหมาะสมที่สุดเท่าที่สำรวจมาจะเป็นผลเฉลยของการค้นหา ในตัวอย่างนักสำรวจอาจหมายถึงนักสำรวจเดินสำรวจมาถึงตำแหน่งที่ต่ำสุดของเทือกเขาเท่าที่เดินสำรวจมา (ไม่ได้หมายถึงตำแหน่งที่ลึกที่สุดของเทือกเขาทั้งหมด) นักสำรวจจะเดินต่อไปอีก 20 ก้าว ถ้าไม่เจอตำแหน่งที่ลึกกว่าตำแหน่งต่ำสุดที่ค้นพบ จะหยุดการค้นหาและถือว่าจุดที่ลึกที่สุดที่ค้นพบล่าสุดเป็นผลเฉลยของปัญหา

การกำหนด stopping criteria อาจเป็นวิธีใดวิธีหนึ่งดังกล่าวข้างต้น หรือเป็นแบบผสมของทั้ง 2 วิธีก็ได้ การกำหนดจำนวนการทำซ้ำของทั้งสองวิธีนั้นไม่มีกฎที่แน่นอน ขึ้นอยู่กับลักษณะของปัญหา และความเข้มข้นของ

การค้นหา การกำหนดจำนวนการทำซ้ำมากๆ จะทำให้ได้คำตอบที่ดี แต่จะต้องใช้เวลาในการค้นหาคำตอบนาน ในทางตรงกันข้ามถ้าใช้จำนวนการทำซ้ำน้อยๆ จะทำให้ได้คำตอบเร็วแต่อาจจะได้คำตอบที่ไม่ดีเท่าที่ควร

### ตัวอย่างการประยุกต์ใช้วิธี Tabu search

ในหัวข้อนี้จะกล่าวถึงการประยุกต์ใช้ วิธี Tabu search การแก้ปัญหาการจัดตารางการผลิต (production scheduling) ซึ่งจะเป็นการจัดลำดับการผลิตของงาน 8 งาน บนเครื่องจักร 1 เครื่อง โดยที่งานทั้ง 8 งานมีเวลาในการผลิต (processing time) กำหนดส่งงาน (due-date) และค่าเก็บรักษาในกรณีที่ผลิตเสร็จก่อนกำหนดส่ง (บาท/หน่วย/เวลา) แตกต่างกัน รายละเอียดของตัวอย่างเป็นดัง Table 1

ในตัวอย่างนี้จะใช้วิธี Tabu search ในการหาผลเฉลยหรือลำดับของงานที่เข้าสู่เครื่องจักรที่ทำให้เกิดค่าเก็บรักษาโดยรวมต่ำสุด โดยการค้นหาผลเฉลยนั้นจะเริ่มที่ลำดับของงานที่จัดเรียงตามลำดับงานที่มีกำหนดส่งเร็วที่สุด เข้าสู่เครื่องจักรก่อน ซึ่งก็คือลำดับงาน 8, 7, 6, 5, 4, 3, 2 และ 1 ตามลำดับ แล้วพยายามจัดงานแต่ละงานให้ใกล้กับเวลากำหนดส่งให้มากที่สุดเท่าที่เป็นไปได้ การค้นหาคำตอบที่ดีขึ้นจะทำได้โดยการเลือกสลับคู่ของงานที่ให้ผลเฉลยที่เหมาะสมที่สุดในแต่ละครั้งของการทำซ้ำ โดยกำหนดกฎเกณฑ์สำหรับ Tabu search ดังต่อไปนี้

*Tabu list size* = 3

*Tabu list type*:

คู่ที่ถูกสลับจะสลับกันอีกไม่ได้จนกว่าจะครบ 3 iterations หรือเป็นไปตามเงื่อนไขที่กำหนดใน aspiration criteria.

*Aspiration criteria*:

คู่งานที่อยู่ในสถานะ Tabu หรือถูกห้ามไม่ให้สลับกัน ถ้าทำการทดลองสลับกันแล้วให้คำตอบต่ำสุดเท่าที่ทำการค้นหาคำตอบมา ให้ทำการสลับได้โดยไม่ต้องคำนึงถึงสถานะ Tabu ของงานคู่นั้น

*Stopping criteria*:

การค้นหาผลเฉลยจะหยุดเมื่อมีการค้นหาผ่านไป

ในที่นี้จะแสดงการทำการค้นหาคำตอบทั้งหมดเพียง 2 iterations จากลำดับของงาน {8→7→6→5→4→3→2→1} นั้นโดยการจัดงานให้ใกล้กับกำหนดส่งของงานแต่ละงานให้มากที่สุดจะได้ว่า งาน 1 เริ่มที่เวลา 77 แล้วเสร็จเวลา 80 งาน 2 เริ่มที่เวลา 73 แล้วเสร็จเวลา 77 งาน 3 เริ่มที่เวลา 64 แล้วเสร็จเวลา 73 งาน 4 เริ่มที่เวลา 54 แล้วเสร็จเวลา 64 งาน 5 เริ่มที่เวลา 44 แล้วเสร็จเวลา 54 งาน 6 เริ่มที่เวลา 39 แล้วเสร็จเวลา 44 งาน 7 เริ่มที่เวลา 32 แล้วเสร็จเวลา 39 และงาน 8 เริ่มที่เวลา 30 แล้วเสร็จเวลา 32 โดยการจัดตารางการผลิตแบบนี้จะมีค่าเก็บรักษาทั้งสิ้น 244 บาท

**Iteration ที่ 1:** จากลำดับ {8→7→6→5→4→3→2→1} ทดลองสลับทุกคู่งานที่เป็นไปได้ ซึ่งมี (1,2) (1,3) (1,4) (1,5) (1,6) (1,7) (1,8) (2,3) (2,4) (2,5) (2,6) (2,7) (2,8) (3,4) (3,5) (3,6) (3,7) (3,8) (4,5) (4,6) (4,7) (4,8) (5,6) (5,7) (5,8) (6,7) (6,8) (7,8)

การสลับคู่งานที่ให้คำตอบที่เหมาะสมที่สุดในครั้งนี้คือการสลับคู่ระหว่างงาน 4 และ 5 ซึ่งจะทำให้ได้ลำดับงานคือ {8→7→6→4→5→3→2→1} ซึ่งจะมีค่าเก็บรักษาทั้งสิ้น 232 บาท ดังนั้นคู่งาน (4,5) จะสลับกันอีกไม่ได้จนกว่าจะครบ 3 iterations หรือจะเริ่มสลับกันได้อีกที่

**Table 1. Parameters of an example**

| Job (i)                                      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
|--|----|----|----|----|----|----|----|----|
| Processing time ( $t_i$ )<br>(unit time)     | 3  | 4  | 9  | 10 | 10 | 5  | 7  | 2  |
| Due-date ( $D_i$ )<br>(Unit time)            | 80 | 80 | 75 | 66 | 64 | 60 | 50 | 43 |
| Earliness cost ( $E_i$ )<br>(Bath/unit time) | 6  | 4  | 9  | 2  | 3  | 7  | 5  | 1  |

iteration ที่ 4

**Iteration ที่ 2:** จากลำดับ {8→7→6→4→5→3→2→1} ทดลองสลับทุกคู่งานที่เป็นไปได้ ซึ่งมี (1,2) (1,3) (1,4) (1,5) (1,6) (1,7) (1,8) (2,3) (2,4) (2,5) (2,6) (2,7) (2,8) (3,4) (3,5) (3,6) (3,7) (3,8) (4,5) (4,6) (4,7) (4,8) (5,6) (5,7) (5,8) (6,7) (6,8) (7,8) เลือกคู่งานที่สลับแล้วให้ค่าเก็บรักษารวมน้อยที่สุด (ค่าที่น้อยที่สุดใน iteration นี้ไม่จำเป็นที่จะต้องน้อยกว่า 232 บาท แต่ถ้าค่าน้อยที่สุดในการทดลองที่ iteration นี้คือ การสลับคู่ (4,5) จะไม่ถูกเลือกเพราะจะทำให้กลับไปลำดับงาน {8→7→6→5→4→3→2→1} อีก ยกเว้นในกรณีที่การสลับคู่งาน (4,5) จะให้ค่าเก็บรักษารวมน้อยกว่า 232 บาท (เป็นไปตามเงื่อนไขของ aspiration criteria) ซึ่งในตัวอย่างนี้ไม่ใช่ จึงตัดการสลับคู่ระหว่าง (4,5) ออกจากการพิจารณา ให้ทำการสลับคู่อื่นที่ให้ผลเฉลยที่เหมาะสมที่สุด แต่ไม่ใช่คู่ (4,5) เพื่อป้องกันการย้อนกลับของปัญหา ในขั้นตอนต่อไปก็ทำในลักษณะเดียวกันนี้จนกว่าจะครบ 30 iterations (stopping criteria) แล้วจึงเลือกลำดับการเข้างานที่เหมาะสมที่สุดที่ให้ค่าเก็บรักษาดำสุด

สรุป

ในบทความนี้ได้กล่าวถึงหลักการการค้นหาคำตอบแบบฮิวริสติกโดยทั่วไป เพื่อใช้ในการหาผลเฉลยที่ดี และรวดเร็ว โดยเฉพาะกับปัญหาที่มีขนาดใหญ่ที่ไม่สามารถจะหาคำตอบโดยวิธีที่หาผลเฉลยที่เหมาะสมที่สุด ได้หรือต้องใช้เวลาในการหาผลเฉลยนานมากเกินไปเกินกว่าที่จะรอได้ ในที่นี้ได้นำเสนอหลักการเบื้องต้นของวิธีฮิวริสติก แบบ Tabu search ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายในการหาผลเฉลยในงานวิจัยดำเนินงาน พร้อมทั้งยกตัวอย่างประกอบในการทำการค้นหาผลเฉลย

#### เอกสารอ้างอิง

- Clarke, G. and Wright, J.W. 1964. Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12(4), pp. 568-581.
- Glover, F. 1990, Tabu search: A tutorial, *Interface*, 20 (4), pp. 74-79.
- Glover, F. 1993. A user's guide to Tabu search, *Annual of Operations Research*, 41, pp. 3-28