

การประยุกต์ใช้คอมพิวเตอร์แบบคู่ขนานในการค้นหาทาง

Application of Parallel Computer for Tabu Search

คำนำ

ปัจจุบันปัญหาการจัดตารางการผลิต (Production Scheduling) จัดเป็นปัญหาทางด้านอุตสาหกรรมที่เกี่ยวข้องกับการจัดเตรียมตารางการทำงานให้กับทรัพยากร ได้แก่ คน เครื่องจักร อุปกรณ์ และเวลาที่ใช้ในการปฏิบัติงานให้เหมาะสม เนื่องจากตารางการผลิตจะเป็นตัวกำหนดว่าการส่งงานช้าหรือไม่ ส่งผลไปยังค่าใช้จ่ายที่จะต้องเสียเมื่อเกิดการล่าช้าของงานความต้องการที่จะใช้เวลาในการผลิตให้สั้นที่สุดเพื่อสร้างความพึงพอใจให้กับลูกค้า ดังนั้นสำหรับระบบการผลิตแบบต่อเนื่อง (Flow Shop) ที่พบมากในโรงงานอุตสาหกรรมที่มีกระบวนการผลิตหลายขั้นตอนซึ่งต่อเนื่องกันไปตลอด ปัญหาการจัดตารางการผลิตจึงเป็นสิ่งสำคัญที่จะส่งผลกระทบต่อประสิทธิภาพและเวลาของระบบการผลิต การแก้ปัญหาดังกล่าวโดยทั่วไปมักใช้วิธีการหาคำตอบที่เหมาะสมที่สุด (Optimal Solution) นั้นต้องใช้เวลาในการประมวลผลสูงมาก หรืออาจจะเป็นไปได้ที่ไม่สามารถจะประมวลผลหาคำตอบที่เหมาะสมที่สุดได้ เนื่องจากปัญหาการจัดตารางการผลิตจัดเป็นปัญหาที่ไม่เป็นโพลิโนเมียล (NP-Problem) ขนาดใหญ่ ซึ่งยากที่จะหาคำตอบที่เหมาะสมที่สุด วิธีการหาคำตอบแบบเมตา-ฮิวริสติก (Meta – Heuristics) เป็นแนวทางหนึ่งที่น่าสนใจมาใช้ในการแก้ปัญหาดังกล่าวที่ไม่เป็นโพลิโนเมียล เพราะใช้เวลาในการคำนวณน้อยกว่ามาก อีกทั้งคำตอบที่ได้จากวิธีการหาคำตอบแบบเมตา-ฮิวริสติก ก็สามารถยอมรับได้ในการใช้งานจริงเหมาะสมในการนำมาแก้ปัญหาด้านอุตสาหกรรม

ดังนั้นวิธีการหาคำตอบแบบเมตา-ฮิวริสติกหลายวิธีได้ถูกนำมาประยุกต์ใช้กับปัญหาต่างๆ อย่างแพร่หลาย ซึ่งเป็นที่รู้จักกันดี ได้แก่ วิธีการค้นหาทาง (Tabu Search) วิธีการหาคำตอบเชิงพันธุกรรม (Genetic Algorithm) และวิธีการจำลองแอนนีลิ่ง (Simulated Annealing) ได้ถูกพัฒนาขึ้น และทดสอบกับปัญหาทางด้านอุตสาหกรรม พบว่าวิธีฮิวริสติกทั้งสามนี้ใช้งานได้ดีทั้งในด้านคุณภาพคำตอบและเวลาในการประมวลผลเมื่อเทียบกับวิธีการหาคำตอบที่เหมาะสมที่สุด วิธีการค้นหาทาง (Tabu Search) ซึ่งเป็นวิธีการหาคำตอบวิธีแบบเมตา-ฮิวริสติกที่ได้รับความนิยมมาใช้ในการแก้ปัญหาด้านอุตสาหกรรม เพราะคุณภาพของคำตอบที่ได้จากวิธีการนี้สามารถ

นำมาใช้ได้จริงในงานอุตสาหกรรมในเวลาที่รวดเร็ว อีกทั้งยังสามารถปรับปรุงคุณภาพของคำตอบให้สูงขึ้นโดยการกำหนดระยะเวลาในการหาคำตอบให้มากขึ้นได้ โดยงานวิจัยฉบับนี้ได้เสนอวิธีการปรับปรุงวิธีการค้นหาทาง (Tabu Search) ในการแก้ปัญหาการจัดตารางการผลิตแบบต่อเนื่อง (Flow Shop Scheduling) โดยการนำระบบคอมพิวเตอร์แบบคู่ขนาน (Parallel Computer) เพื่อปรับปรุงคุณภาพและเวลาในการค้นหาคำตอบ และเปรียบเทียบให้เห็นว่าคุณภาพและเวลาในการค้นหาคำตอบ โดยใช้คอมพิวเตอร์เครื่องเดียว กับคุณภาพและเวลาในการค้นหาคำตอบที่ใช้ระบบคอมพิวเตอร์แบบคู่ขนาน นั้นดีขึ้นเล็กน้อยเพียงใด

วัตถุประสงค์ของการวิจัย

1. เพื่อพัฒนาวิธีการค้นหาทางแบบขนาน (Parallel Tabu Search) ในการจัดตารางการผลิตแบบต่อเนื่อง (Flow Shop Scheduling)
- 2 . เปรียบเทียบคุณภาพและเวลาในการหาคำตอบของวิธีการค้นหาทางแบบขนาน (Parallel Tabu Search) กับวิธีการค้นหาทาง (Tabu Search) ที่ใช้คอมพิวเตอร์เครื่องเดียว
3. เปรียบเทียบคุณภาพคำตอบของวิธีการค้นหาทางแบบขนาน (Parallel Tabu Search) กับเมตา-ฮิวริสติก (Meta – Heuristics) วิธีอื่น

ขอบเขตของงานวิจัย

ศึกษาวิธีการจัดตารางการผลิตแบบต่อเนื่อง (Flow Shop Scheduling) ซึ่งเวลาในการผลิตรวมเกิดจากเวลาที่ใช้ในการผลิตจริงบวกกับเวลาที่ขึ้นงานรอคอยเครื่องจักรโดยไม่คำนึงถึงเวลาในการติดตั้ง (Setup Time) โดยกำหนดให้งานทุกงานต้องผ่านเครื่องจักรทุกเครื่อง โดยมีลำดับการผลิตบนแต่ละเครื่องเหมือนกัน

ประโยชน์ที่คาดว่าจะได้รับ

เพื่อปรับปรุงคุณภาพและระยะเวลาในการหาคำตอบ ที่ได้จาวิธีการค้นหาทาง (Tabu search) โดยใช้ระบบคอมพิวเตอร์แบบคู่ขนาน (Parallel Computer) มาใช้ในการจัดการการผลิตแบบต่อเนื่องโดยทำให้เวลาที่ใช้ในการผลิตรวม (Makespan) น้อยที่สุด ซึ่งทำให้คุณภาพของคำตอบที่ได้ดีขึ้น เพราะมีหน่วยประมวลผลหลายหน่วยช่วยในการคำนวณ

การตรวจเอกสาร

การหาคำตอบที่ดีที่สุดโดยทั่วไปจะเป็นเทคนิคในการแก้ปัญหาจากความสัมพันธ์ของตัวแปรต่างๆ เทคนิคที่สามารถเป็นที่เข้าใจได้ง่ายและเป็นที่ยอมรับ เช่น วิธีซิมเพล็กซ์ (Simplex Method) สำหรับปัญหาโปรแกรมเชิงเส้น (Linear Programming) โดยระบบสมการสามารถเขียนแทนด้วยสมการทางคณิตศาสตร์ซึ่งประกอบไปด้วยสมการเป้าหมาย (Objective Function) สมการข้อจำกัด (Constraints) ความสัมพันธ์ของตัวแปรต่างๆ ในสมการที่มีรูปแบบเป็นเส้นตรง และตัวแปรทุกตัวต้องมีค่ามากกว่าหรือเท่ากับศูนย์ แต่ข้อเสียที่พบ (South *et al.*, 1993) คือ ไม่สามารถแก้ปัญหาที่มีความซับซ้อนหรือปัญหาที่มีขนาดใหญ่ได้ ทำให้ไม่เหมาะสมกับปัญหาในการผลิตที่เป็นปัญหามีขนาดใหญ่

วิธีการค้นหาทาง (Tabu Search) ซึ่งเป็นวิธีการหาคำตอบแบบเมตา-ฮิวริสติกที่ได้รับความนิยมมาใช้ในการแก้ปัญหาทางด้านอุตสาหกรรม ที่มีขนาดใหญ่ เพราะคุณภาพของคำตอบที่ได้จากวิธีการนี้สามารถนำมาใช้ได้จริงในงานอุตสาหกรรมในเวลาที่รวดเร็ว อีกทั้งยังสามารถปรับปรุงคุณภาพของคำตอบให้สูงขึ้นโดยการกำหนดระยะเวลาในการหาคำตอบให้มากขึ้นได้

วิธีการค้นหาทาง

วิธีการค้นหาทาง (Tabu Search) เป็นวิธีการหาคำตอบแบบเมตา-ฮิวริสติก (Meta – Heuristics) ที่ได้รับการยอมรับว่าสามารถที่จะหลีกเลี่ยงการให้คำตอบสุดท้ายเฉพาะที่ (Local Optimum) และยังสามารถดำเนินการค้นหาต่อไปเรื่อยๆ จนกระทั่งได้คำตอบใกล้เคียงกับคำตอบที่ดีที่สุดทั่วไป (Global Optimum) เพราะคำว่า Tabu หรือ Taboo นั้นหมายถึงข้อห้าม หรือการห้าม ดังนั้น วิธีการค้นหาทาง จึงหมายถึงวิธีการค้นหาแบบมีข้อห้าม ในวิธีการห้ามดังกล่าวจะเป็นการห้ามเพื่อที่จะช่วยให้ไม่ต้องไปหาผลเฉลยเดิม หรือหลงในวัฏจักร (cyclic) ซึ่งจะส่งผลให้ไม่สามารถหาคำตอบที่ดีขึ้นได้ (Glover, 1990)

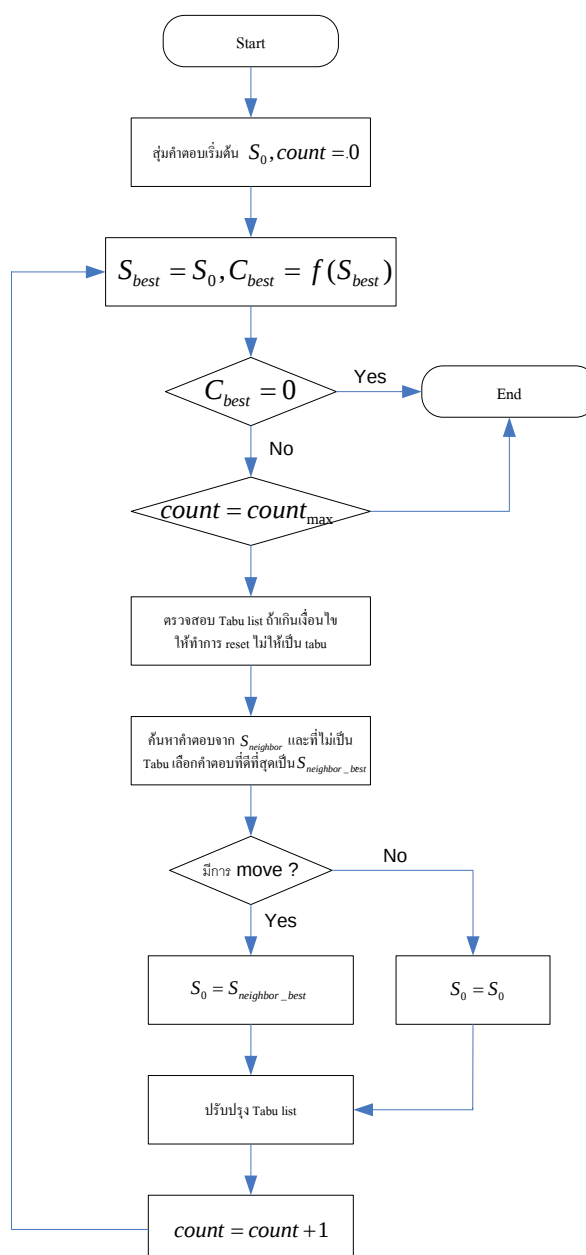
วิธีการค้นหาทางประกอบไปด้วยรูปแบบการค้นหาคำตอบ 2 รูปแบบที่สำคัญคือ การค้นหาคำตอบโดยใช้หน่วยความจำระยะสั้น (Short Term Memory) และการค้นหาคำตอบโดยใช้หน่วยความจำระยะยาว (Long Term Memory) ซึ่งการค้นหาคำตอบโดยใช้หน่วยความจำระยะสั้น ถือเป็นหน่วยความจำตามเวลา (Recency Base Memory) หมายถึงการค้นหาที่จดจำอดีตหรือ

ประสบการณ์ในการค้นหาที่ผ่านมาเพียงระยะสั้น ในทางตรงข้ามการค้นหาคำตอบโดยใช้หน่วยความจำระยะยาวซึ่งถือเป็นหน่วยความจำตามความถี่ (Frequency Base Memory) หมายถึงการค้นหาคำตอบที่ต้องจดจำอดีตหรือประสบการณ์ที่ผ่านมาเพื่อช่วยให้การค้นหาคำตอบเป็นไปอย่างมีประสิทธิภาพมากขึ้น (สุภชัย และ วณิดา, 2546) โดยมีองค์ประกอบหลักๆ คือ

1. ฟังก์ชันเป้าหมาย (Objective Function) เพื่อใช้สำหรับประกอบการตัดสินใจในการค้นหาคำตอบว่าจะค้นหาคำตอบไปในทิศทางใด ฟังก์ชันเป้าหมายจะใช้เพื่อเป็นตัวประเมินการค้นหาคำตอบว่าคำตอบที่ค้นหาอยู่ในแต่ละขั้นตอนนั้นมีประสิทธิภาพเพียงใด
2. ข้อจำกัดการเป็นทาบู (Tabu Restriction) นั้นเป็นตัวควบคุมการย้อนกลับ (Reverse) หรือการวนรอบ (Cyclic) อยู่ในขอบเขตการค้นหาเดิมๆ หรือคำตอบเดิมที่มีอยู่ การกำหนดรูปแบบการเป็นทาบูนั่นขึ้นอยู่กับแต่ละปัญหา ไม่จำเป็นต้องเหมือนกัน เช่นอาจถูกกำหนดไว้ว่าห้ามกลับมายังตำแหน่งเดิมเป็นต้น
3. รายการทาบู (Tabu List) ถือเป็นข้อมูลตามเวลาจริงหรือข้อมูลความคงอยู่ (Recency) เป็นตัวกำหนดจำนวนครั้ง ที่ต้องการห้ามไม่ให้เกิดการก้าวกลับมาที่เดิม เพราะการกำหนดรูปแบบการเป็นทาบูว่า ห้ามไม่ให้เกิดการกลับมาตำแหน่งเดิมนั้นตลอดไป อาจที่จะพลาดโอกาสการหาคำตอบที่ดีก็ได้ การกำหนดรายการทาบูจึงเป็นการกำหนดว่าจะห้ามเงื่อนไขของรูปแบบการเป็นทาบูอยู่นานเท่าไร การกำหนดจำนวนการเป็นทาบู (Tabu Size) ไม่มีกฎเกณฑ์ตายตัวว่าจะเป็นเท่าไร ซึ่งการกำหนดสูงเกินไปก็จะทำให้พลาดโอกาสที่จะได้คำตอบที่ดี ในขณะที่เดียวกันถ้ากำหนดน้อยเกินไปอาจจะทำให้ได้คำตอบที่ย้อนกลับหรือการวนรอบในขอบเขตเดิมๆ ได้
4. เกณฑ์ความปรารถนา (Aspiration Criteria) เป็นรูปแบบการควบคุมที่ช่วยในการค้นหาคำตอบที่ดีเป็นไปด้วยดี และรวดเร็วมากขึ้น โดยกำหนดเป็นเงื่อนไขสำหรับการค้นหาที่ถูกเป็นทาบูนั่นสามารถให้คำตอบที่เหมาะสมที่สุดเท่าที่เคยได้ตั้งแต่เริ่มการค้นหาจนถึงปัจจุบัน จะสามารถยกเลิกการห้ามสำหรับการค้นหานั่นได้ เกณฑ์ความปรารถนา ก็ไม่มีกฎเกณฑ์ที่ตายตัวขึ้นอยู่กับลักษณะของปัญหาหลัก

ขั้นตอนของวิธีการค้นหา

Bland and Dawson (1991) ได้สรุปแนวคิดและขั้นตอนวิธีการค้นหา (Tabu Search) ซึ่งเป็นขั้นตอนที่ถูกลำมาประยุกต์ใช้ในการหาคำตอบที่เหมาะสม ได้อย่างมีประสิทธิภาพโดยหลักการพื้นฐานที่ทำให้ วิธีการค้นหา แตกต่างจากการค้นหาคำตอบแบบอื่นคือมีการเพิ่ม Tabu list และ Aspiration criteria ทำให้การหาคำตอบเสร็จสิ้นได้เร็วขึ้น สามารถแสดงขั้นตอนการดำเนินการได้ดังรูป ที่ 1



ภาพที่ 1 โครงสร้างการดำเนินการของ วิธีการค้นหา (Tabu Search)

คำตอบที่ได้จากวิธีการค้นหาทาง Tabu Search นั้นคำตอบที่ได้อาจจะไม่ใช่คำตอบที่ดีที่สุด แต่เป็นคำตอบที่ยอมรับได้ และสามารถหาคำตอบที่ดีขึ้นโดยการเพิ่มรอบในการค้นหา

ข้อดีและข้อแตกต่างของวิธีการค้นหาทาง Tabu Search เมื่อเทียบกับวิธีอื่น (สมศักดิ์, 2547)

1. วิธีการค้นหาทาง Tabu Search ได้จากแนวคิดที่ว่าการค้นหาแบบฉลาดจะต้องพิจารณาถึงหน่วยความจำปรับตัว (Adaptive Memory) สำหรับการค้นหาอย่างมีประสิทธิภาพแต่ วิธหาคำตอบเชิงพันธุกรรม (Genetic Algorithm) และวิธีการจำลองแอนนีลิ่ง (Simulated Annealing) ใช้การสุ่ม (Random) ในการค้นหา
2. วิธีการค้นหาทาง Tabu Search ใช้การสำรวจแบบตอบสนอง (Responsive Exploration) ได้แนวคิดจากการที่ว่าในบางครั้งเส้นทางที่เลวให้ข้อมูลมากกว่าเส้นทางที่ดี เพื่อหาเส้นทางใหม่ที่ดีขึ้น
3. วิธีการค้นหาทาง Tabu Search ใช้หลักการสร้างความละเอียด (Intensification) และความหลากหลาย (Diversification) โดยเพิ่มความละเอียดในการค้นหาบริเวณใกล้เคียงกับคำตอบที่ดีที่เคยพบและค้นหาเพิ่มเติมในบริเวณที่แตกต่างจากคำตอบที่ดีที่เคยพบ
4. วิธีการค้นหาทาง Tabu Search สามารถที่จะหลีกเลี่ยงการให้คำตอบสุดท้ายที่คำตอบสุดท้ายเฉพาะที่ (Local Optimal) และยังสามารถดำเนินการค้นหาคำตอบต่อไปเรื่อยๆ จนกระทั่งได้คำตอบที่ค่อนข้างใกล้เคียงกับคำตอบที่เป็นคำตอบที่ดีที่สุดทั่วไป (Global Optimal)
5. วิธีการค้นหาทาง Tabu Search ใช้สำหรับการประยุกต์ใช้สำหรับหาคำตอบที่ดีที่สุดเชิงผสมผสาน (Combination Optimization) ได้อย่างมีประสิทธิภาพ

คอมพิวเตอร์แบบคู่ขนาน

ประเภทของคอมพิวเตอร์แบบคู่ขนาน

การทำระบบคอมพิวเตอร์แบบคู่ขนานกับโปรแกรมโดยทั่วไป สามารถแบ่งออกได้เป็น 2 รูปแบบ (Kumar *et al.*, 1994) คือ การทำระบบคู่ขนานกับส่วนของข้อมูลที่ต้องการตรวจสอบ หรือ

กับส่วนของการทำงานในขั้นตอนวิธีของโปรแกรม วิธีที่จะนำมาใช้งานขึ้นอยู่กับรูปแบบของการใช้งานโปรแกรม (Dietz, 1998)

1. MPMD (Multiple-Program, Multiple-Data) โปรแกรมในการประมวลผลจะแยกปัญหาให้หน่วยประมวลผลแต่ละหน่วยคำนวณอย่างอิสระต่อกัน และส่งคำตอบสุดท้ายกลับมารวบรวมที่หน่วยประมวลผลหลัก เพื่อประมวลผลขั้นสุดท้ายต่อไป การประมวลผลประเภทนี้เหมาะแก่วิธีการคำนวณแบบหาคำตอบเฉพาะที่ (Local Search)

2. SPMD (Single-Program, Multiple-Data) โปรแกรมในการประมวลผลจะแยกปัญหาให้หน่วยประมวลผลแต่ละหน่วยคำนวณและส่งคำตอบกลับมาประมวลที่หน่วยประมวลผลกลางไปมาอย่างต่อเนื่องตลอด การประมวลผลประเภทนี้เหมาะแก่วิธีการคำนวณที่เกี่ยวข้องกับเมตริกซ์

รูปแบบการติดต่อข้อมูลในการประมวลผล

1. แบบไม่ใช้อุปกรณ์ต่อพ่วง ซึ่งจะเหมาะสมกับการคำนวณแบบ MPMD แต่ผู้ใช้จะต้องเขียนโปรแกรมแยกสำหรับแต่ละหน่วยประมวลผล และจะต้องรวบรวมผลการทดลองในแต่ละหน่วยประมวลผลด้วยตัวผู้ใช้เอง วิธีนี้มีข้อดีตรงที่ไม่เกิดความล่าช้าในการประมวลผลของการส่งผ่านข้อมูลระหว่างอุปกรณ์ต่อพ่วง มีข้อเสียตรงที่ใช้ได้แต่การคำนวณแบบ MPMD และอาจเกิดความผิดพลาดในการรวบรวมผลการทดลองของผู้ใช้ได้

2. แบบใช้อุปกรณ์ต่อพ่วง ซึ่งจะเหมาะสมกับการคำนวณทั้งแบบ MPMD และ SPMD วิธีนี้มีข้อดีตรงที่สามารถส่งงานและรวบรวมผลได้จากหน่วยประมวลผลหลัก แต่มีข้อเสียตรงที่เกิดความล่าช้าในการประมวลผลของการส่งผ่านข้อมูลระหว่างอุปกรณ์ต่อพ่วง

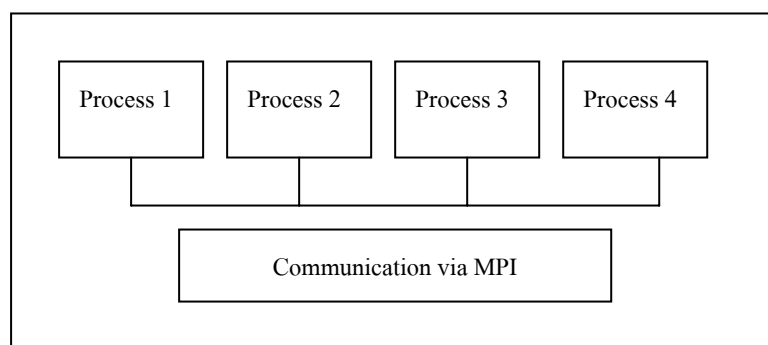
MPI

เป็นการประมวลผลแบบคู่ขนาน หลากๆกระบวนการจะสามารถดำเนินงานต่างหากกัน และกระบวนการเหล่านั้น มีการติดต่อกันผ่านทาง MPI (Message Passing Interface) ดังภาพที่ 2 โดยที่ผู้ใช้ไม่จำเป็นต้องรู้ถึงส่วนประกอบโครงสร้างภายใน จำเป็นต้องรู้แค่ว่าจะใช้ประโยชน์จากมันได้

อย่างไร ทุกๆอย่างใน MPI สามารถสรุปอย่างสั้นๆง่ายๆได้ว่า “ส่งข้อความ-รับข้อความ” (Message Sent-Message Receive)

ในรูปแบบของแนวคิดซึ่งเป็นแนวคิดที่มีเหตุผลมาก รูปแบบที่พบในการประมวลผลแบบคู่ขนาน มีหลายส่วนคล้ายกันกับรูปแบบการทำงานในสังคม เช่น การจัดการธุรกิจ ซึ่งมีแนวคิดของคนมากมายในที่ทำงานเดียวกัน ทุกคนทำงานร่วมกันเพื่อให้องค์กรไปถึงจุดมุ่งหมายเดียวกัน แนวคิดของ MPI (Message Passing Interface) นั่นก็คือแนวความคิดที่จะแบ่งแยกงานออกเป็นส่วนย่อยๆ จากจุดนี้พบว่า การประมวลผลแบบคู่ขนานนั้นลำดับขั้นตอนเริ่มต้นต้องคิดก่อนว่า ปัญหาอะไรที่ต้องการจะแก้ จากนั้นเข้าถึงแหล่งที่มีอยู่ของปัญหา จากนั้นพยายามแยกปัญหาที่มีอยู่ ออกเป็นส่วนย่อยที่อยู่ในรูปที่สามารถจัดการได้ ดังภาพที่ 2 โดยมีรูปแบบหลักๆ 2 ส่วนด้วยกันคือ

1. การที่จะทำความเข้าใจว่า การทำงานของระบบคอมพิวเตอร์แบบคู่ และ โปรแกรมแบบคู่ขนานนั้นมีหลักการในการทำงานอย่างไร
2. จะสามารถนำปัญหานั้นแบ่งแยกให้เกิดประสิทธิภาพได้อย่างไร และจะจัดการแก้ไขปัญหานั้นอย่างไร



ภาพที่ 2 กระบวนการการทำงานร่วมกันของ MPI

เริ่มต้นเกี่ยวกับการคำนวณแบบคู่ขนานโดยทำความเข้าใจจากระดับสูง ความแตกต่างระหว่าง Serial Code กับ Parallel Code ก็คือ Serial Code นั้นจะเป็นการทำงานเป็นแบบรวมศูนย์ ก็คือขั้นตอนในการแก้ปัญหาทุกอย่างจะถูกดำเนินการโดยผ่านตัวประมวลผลเพียงชุดเดียว ซึ่งจะแตกต่างกับ Parallel Code ที่เป็นการทำงานแบบคู่ขนาน ก็คือขั้นตอนต่างๆในการแก้ปัญหานั้นจะถูกแบ่งออกเป็นส่วนย่อยๆ และถูกดำเนินการประมวลผลไปพร้อมๆกัน และแต่ละหน่วย

ประมวลผลจะติดต่อสื่อสารกันเพื่อที่จะทำงานร่วมกัน ทำให้การแก้ปัญหาเป็นไปได้อย่างรวดเร็วและมีประสิทธิภาพมากกว่า Serial Code แต่ Parallel Code นั้นก็มีปัญหาอยู่ว่าจะแบ่งแยกปัญหาที่มีออกเป็นส่วนๆอย่างไรถึงจะทำให้เกิดประสิทธิภาพสูงสุด สิ่งที่สำคัญที่ต้องดู 2 สิ่งก็คือ

1. งานอะไรที่ทำให้การแก้ปัญหานั้นบรรลุเป้าหมาย
2. เมื่อไหร่ที่การติดต่อสื่อสารระหว่างกระบวนการมีความจำเป็น

ยกตัวอย่าง เช่น ถ้าในห้องเรียนมีหนังสืออยู่เล่มเดียว จุดมุ่งหมายคือทุกคนต้องอ่านหนังสือกลยุทธ์ที่เป็นไปได้ คือแบ่งทุกคนในชั้นเรียนให้อ่านตามลำดับคิว นี่คือการบวนการลำดับขั้น ซึ่งเป็นแบบ Serial Code ซึ่งคนที่ไม่มีหนังสือก็ต้องคอยตามลำดับคิว เป็นการรออย่างว่างเปล่า อย่างไรก็ตามถ้าเป็น Parallel Code การแก้ปัญหาคือดีที่สุดที่ทุกคนได้อ่านหนังสือ ก็คือการที่ทุกคนมีหนังสือได้อ่าน (Karniadakis, 2003)

MPI Programming

จุดเริ่มต้นในการเขียนโปรแกรมของ MPI จะต้องมีการเรียกตัวเริ่มต้นกระบวนการของการเขียนโปรแกรมแบบ MPI ก็คือ

```
# include "mpi.h"
```

หมายถึง การประกาศเรียกใช้คำสั่งต่างๆที่อยู่ในไฟล์ mpi.h ที่ประกอบไปด้วยคำสั่งต่างๆที่เกี่ยวข้องกับการทำงานของระบบประมวลผลแบบคู่ขนาน

MPI จะประกอบด้วยหลายๆแผนการเพื่อเป็นตัวบ่งชี้สำหรับโปรแกรม โดยที่ตัวบ่งชี้ของ MPI นั้นเกือบทั้งหมดจะอยู่ในรูปของอักขระตัวใหญ่ และมักจะขึ้นต้นด้วย "MPI_" เช่น MPI_CHAR อักขระตัวแรกที่อยู่ใน MPI แต่ถ้าเป็นฟังก์ชันของ MPI มักจะขึ้นต้น "MPI_" เหมือนกัน แต่มักจะตามด้วยตัวย่อของฟังก์ชันนั้นที่เป็นอักขระตัวเล็ก เช่น MPI_Init เป็นต้น

ก่อนที่จะเรียกฟังก์ชันในการทำงานของ MPI อื่นๆ ต้องมีเรียก MPI_Init ก่อนและจะเรียกเพียงครั้งเดียว ตัวแปรที่ถูกกำหนดจะชี้ไปที่ตัวแปรของฟังก์ชัน argc และ argv การประกาศ

MPI_Init นี้ทำให้โปรแกรมที่เขียนขึ้นสามารถใช้ MPI Library ได้ หลังจากทำการเขียนโปรแกรมในส่วนของ MPI สิ้นสุดลง ก็จะต้องเรียกฟังก์ชัน MPI_Finalize เพื่อเป็นการบอกถึงการสิ้นสุดการดำเนินการทำงานของ MPI ซึ่งรูปแบบของการเขียนโปรแกรม MPI จะเป็นดังต่อไปนี้

```
# include "mpi.h"

main(int argv, char* argv[])
{
    /* No MPI function called before this*/

    MPI_Init(&argc, &argv);

    MPI_Finalize();

    /* No MPI function called after this*/
} /* main*/
```

แต่ถ้าโปรแกรมที่เขียนขึ้นนั้นมี MPI เป็นเพียงส่วนประกอบ ไม่ได้เป็นโปรแกรมที่เป็น MPI ทั้งโปรแกรม ก็ไม่มีความจำเป็นที่จะต้องเรียก MPI_Init ตั้งแต่ขั้นแรกของการเขียนเพียงแต่ว่า จะต้องมีการเรียกก่อนส่วนอื่นของ MPI Library ในทางเดียวกัน MPI_Finalize ก็ไม่จำเป็นต้องเรียกในขั้นตอนสุดท้ายของการเขียนโปรแกรม แต่ต้องเรียกเมื่อมีการสิ้นสุดของการโปรแกรมในส่วนของ MPI

ตัวอย่างโปรแกรม Hello World ที่เขียนแบบ MPI

```
# include <iostream.h>
#include <mpi.h>
int main(int argc, char** argv)
{
    MPI_Init(&argc, &argv);
    cout << "Hello World" << endl;
    MPI_Finalize();
}
```

ตัววัดประสิทธิภาพสำหรับระบบขนาน

ระบบขนานเป็นการผสมระหว่างขั้นตอนวิธีและสถาปัตยกรรมแบบขนานที่ใช้ในการพัฒนา ซึ่งมีตัววัดประสิทธิภาพหลายแบบที่ใช้กันทั่วไปมีดังนี้ (หทัยชนก, 2546)

1. เวลาที่ใช้ในการทำงานของโปรแกรม (Run Time)

เวลาที่ใช้ในการทำงานของโปรแกรมตามลำดับ (T_s) หมายถึง เวลาที่นับตั้งแต่เริ่มจนกระทั่งจบการทำงานบนเครื่องคอมพิวเตอร์ทุกขั้นตอน มีหน่วยเป็นวินาที

เวลาที่ใช้ในการทำงานของโปรแกรมแบบขนาน (T_p) หมายถึง เวลาที่นับตั้งแต่ชั่วขณะเวลาที่การคำนวณเริ่มขึ้นจนกระทั่งถึงชั่วขณะเวลาที่หน่วยประมวลผลสุดท้ายทำงานเสร็จ มีหน่วยเป็นวินาที

2. จำนวนเท่าความเร็ว (Speed up)

จำนวนเท่าความเร็วเป็นตัววัดประสิทธิภาพของการทำงานแบบขนานที่ได้จากการเปรียบเทียบกับการทำงานแบบตามลำดับ จำนวนเท่าความเร็วเป็นอัตราส่วนของเวลาที่ใช้ในการแก้ปัญหาบนหนึ่งหน่วยประมวลผลเทียบกับเวลาที่ใช้แก้ปัญหาเดียวกันที่ทำงานบนเครื่องคอมพิวเตอร์แบบขนานที่ใช้ p หน่วยประมวลผล

ขั้นตอนวิธีแบบตามลำดับที่มีไม่เหมาะที่จะปรับให้ทำงานแบบขนานได้ทุกแบบ คอมพิวเตอร์ที่ทำงานตามลำดับจะใช้ขั้นตอนวิธีแบบตามลำดับในการแก้ปัญหาซึ่งจะใช้เวลาน้อยสุดอยู่ค่าหนึ่ง การพิจารณาตัดสินว่าประสิทธิภาพของขั้นตอนวิธีแบบขนานที่ใช้งานควรที่จะเปรียบเทียบกับเวลาที่ใช้โดยขั้นตอนวิธีแบบตามลำดับที่เร็วที่สุดเป็นตัวเปรียบเทียบ แต่บางครั้งเราไม่รู้ว่ขั้นตอนวิธีแบบตามลำดับที่เร็วที่สุดคือแบบใด หรือเวลาที่ใช้ในการแก้ปัญหอาจใช้เวลา นานมากจนไม่สามารถวัดได้ในทางปฏิบัติ ฉะนั้นการวัดประสิทธิภาพของขั้นตอนวิธีแบบขนานจะเปรียบเทียบกับขั้นตอนวิธีแบบตามลำดับที่ดีที่สุดที่มีใช้ในการแก้ปัญหานั้นแทน โดยทั่วไปจะให้จำนวนเท่าความเร็วเป็นอัตราส่วนระหว่างเวลาในการทำงานตามลำดับของขั้นตอนวิธีที่ดีที่สุดเทียบกับเวลาที่ใช้โดยขั้นตอนวิธีแบบขนานที่ทำงานบนด้วย p หน่วยประมวลผลดัง

สมการที่ 1 โดยที่แต่ละหน่วยประมวลผลเป็นแบบเดียวกันกับที่ใช้โดยขั้นตอนวิธีแบบตามลำดับ Speedup แทนด้วยสัญลักษณ์ S ในทางทฤษฎี Speed up มีค่าไม่เกินจำนวนหน่วยประมวลผล p

$$S = T_s/T_p \quad (1)$$

3. ประสิทธิภาพ (Efficiency)

ในทางอุดมคติระบบขนานที่ใช้ p หน่วยประมวลผลจะมี Speedup เท่ากับ p แต่ทางปฏิบัติแล้วไม่สามารถใช้เวลาหน่วยประมวลผลถึง 100 เปอร์เซ็นต์ในการคำนวณขั้นตอนวิธีได้ อาจสูญเสียไปเนื่องจากใช้ในการติดต่อกันระหว่างหลายๆ หน่วยประมวลผล ประสิทธิภาพเป็นตัววัดสัดส่วนของเวลาที่ใช้งานหน่วยประมวลผลอย่าง ซึ่งมีค่าเป็นอัตราส่วนของ Speedup เทียบกับจำนวนหน่วยประมวลผลดังสมการที่ 2 ประสิทธิภาพแทนด้วยสัญลักษณ์ E ในทางอุดมคติระบบขนานจะมี S เท่ากับ p และประสิทธิภาพเท่ากับ 1 แต่ทางปฏิบัติ S น้อยกว่า p และประสิทธิภาพมีค่าระหว่าง 0 ถึง 1 ขึ้นกับระดับความมีประสิทธิภาพในการใช้งานหน่วยประมวลผลอย่างคุ้มค่ามากที่สุด โดยที่ประสิทธิภาพไม่ควรต่ำกว่า 0.6 จึงเหมาะสมในการทำงานแบบขนาน

$$E = S/p \quad (2)$$

4. ต้นทุน (Cost)

ค่าต้นทุนคิดจากเวลาที่ใช้ในการทำงาน โปรแกรมและจำนวนหน่วยประมวลผลที่ใช้ ซึ่งได้มาจากเวลารวมของแต่ละหน่วยประมวลผลใช้ในการแก้ปัญหา ประสิทธิภาพสามารถประเมินได้จากอัตราส่วนของเวลาที่ใช้ในการแก้ปัญหของขั้นตอนวิธีที่เร็วที่สุดกับต้นทุนของการแก้ปัญหาเดียวกันด้วยจำนวน p หน่วยประมวลผล

ต้นทุนของการแก้ปัญหามูลหน่วยประมวลผล 1 หน่วย คือ หาได้จากเวลาที่ใช้ในการแก้ปัญหของขั้นตอนวิธีที่เร็วที่สุด ระบบขนานจะประหยัดต้นทุนมากที่สุดเมื่อต้นทุนในการแก้ปัญหาเป็นค่าแปรผันตรงกับเวลาที่ใช้ในการแก้ปัญหของขั้นตอนวิธีที่เร็วที่สุดบนหนึ่งหน่วยประมวลผล ค่าต้นทุนที่ดีที่สุดที่เป็นได้มีค่าเท่ากับหนึ่ง และบางครั้งต้นทุนอาจอ้างอิงโดยงานที่ได้รับจากทุกหน่วยประมวลผลรวมกันทั้งหมด

การจัดตารางการผลิต

การจัดตารางการผลิตเป็นการจัดเตรียมตารางเวลาการทำงานให้ให้กับทรัพยากรที่เกี่ยวข้อง ได้แก่ คน เครื่องจักร อุปกรณ์ และเวลาที่ใช้ในการปฏิบัติงานให้เกิดประโยชน์สูงสุดเนื่องจากในสภาพความเป็นจริงมีการสั่งงานเข้ามาในโรงงานตลอดเวลาบางงานมีขั้นตอนที่ไม่ซับซ้อนสามารถกระทำการผลิตแบบง่ายๆ บางงานมีขั้นตอนการผลิตที่ซับซ้อนมาก นอกจากนั้นงานแต่ละงานอาจมีความสำคัญของงานแตกต่างกัน สิ่งต่างๆ เหล่านี้อาจมีผลต่อการพิจารณาการจัดตารางการผลิต ซึ่งกระบวนการในการจัดตารางการผลิตมีความยุ่งยากซับซ้อน เพราะแต่ละงานอาจใช้เวลาในแต่ละเครื่องจักรไม่เท่ากัน ทำให้เกิดปัญหาเกี่ยวกับประสิทธิภาพการทำงานของเครื่องจักร เนื่องจากเครื่องจักรบางเครื่องอาจต้องทำงานตลอดเวลา โดยที่บางเครื่องเกิดการว่างงาน นอกจากนั้นยังส่งผลกระทบต่อประสิทธิภาพในการทำงานด้วยเนื่องจากตารางการผลิตจะเป็นตัวกำหนดว่าการส่งงานช้าหรือไม่ ส่งผลไปยังค่าใช้จ่ายที่จะต้องเสียเมื่อเกิดการล่าช้าของงาน และยังมีผลต่อความพึงพอใจของลูกค้าด้วย (จันทร์เพ็ญ, 2539)

การจัดลำดับงานส่วนใหญ่อาศัยหลักเกณฑ์ต่างๆ ที่นิยมนำดังนี้

1. รับก่อนทำก่อน (First Come –First Serve) กล่าวคืองานที่เข้ามาที่หน่วยงานหรือเครื่องจักร จะเข้าแถวคอยรับบริการตามลำดับก่อนหลังของการมาถึงที่หน่วยงาน
2. ทำงานที่ใช้เวลาน้อยที่สุดก่อน (Shortest Processing Time) คืองานใดที่ใช้เวลาในการทำงานน้อยที่สุดจะได้รับการจัดเข้าเป็นอันดับแรก งานที่ใช้เวลาน้อยถัดไปก็จะเป็นอันดับที่ 2, 3 และ 4 จนกระทั่งถึงอันดับที่ k เมื่อ k คือจำนวนงานทั้งหมดที่คอย
3. การทำงานที่ใช้เวลานานที่สุดก่อน (Longest Processing Time) งานที่ใช้เวลาในการทำงานมากที่สุดจะได้รับการจัดเข้าเครื่องจักรก่อน
4. ทำงานที่จะถึงวันกำหนดส่งเร็วที่สุดก่อน (Earliest Due Date)
5. ทำงานขึ้นที่มีเวลาเหลือสำหรับการทำงานน้อยที่สุดก่อน (Minimum Slack Time)

กรณีชิ้นงานนั้นจะต้องผ่านหลายหน่วยงาน จะใช้วิธีการหาค่าเฉลี่ยของค่าเวลาเพียงพอ (Slack) ที่เกิดขึ้นบนแต่ละหน่วยงาน สำหรับค่าเวลาเพียงพอ (Slack) ของงานหาได้จากการเอาเวลาที่ต้องใช้ทั้งหมดบนหน่วยผลิตที่ต้องผ่านลบออกจากเวลาถึงกำหนดส่งมอบงาน หาค่าด้วยจำนวนหน่วยงานที่งานนั้นต้องผ่าน

6. เข้าที่หลังทำก่อน (Last Come-First Serve) งานที่เข้ามาในหน่วยงานหลังสุดจะได้รับการจัดเข้าเครื่องจักรก่อนงานอื่น

สำหรับการเลือกหลักเกณฑ์ต่าง ๆ ในการจัดลำดับงาน สามารถพิจารณาตามประเภทของกระบวนการผลิตซึ่งพอจะสรุปได้ดังนี้

1. การผลิตแบบต่อเนื่อง (Flow Shop Production) คือกระบวนการผลิตที่ผลิต สินค้าเป็นจำนวนมากมีลักษณะเหมือนกันมีกระบวนการผลิตไหลต่อเนื่องกันไปตลอด และมีสินค้าเพียงไม่กี่ชนิด มีขั้นตอนการผลิตที่แน่นอน

2. การผลิตแบบตามสั่ง (Job Shop Production) คือกระบวนการผลิตที่ผลิตงานตามที่ได้รับคำสั่งซึ่งอาจจะประกอบไปด้วยงานหลายชนิด แต่ละชนิดจะมีปริมาณการผลิตที่ไม่เท่ากันและไม่แน่นอน งานแต่ละชนิดที่เข้ามาจะมีขั้นตอนการทำงานที่แตกต่างกันไปทำให้ทำงานลำบากและซับซ้อนมากขึ้น

จุดประสงค์ในการจัดตารางการผลิต ซึ่งมีอยู่หลายประการด้วยกันแต่สามารถแจกแจงออกเป็น 3 ข้อคือ

1. เพื่อพิจารณาการใช้ทรัพยากรการผลิตให้มีประสิทธิภาพสูงสุด นั่นก็จะต้องจัดการอย่างไรจึงจะทำให้เกิดการรอคอยหรือสูญเสียน้อยที่สุด โดยปกติแล้วการใช้ทรัพยากรจะแปรผันโดยตรงกับเวลาที่ต้องการในการทำงาน เวลาดังกล่าวนี้ก็คือ เวลาในการทำงานรวม (Makespan or Max flow time) ของตารางการผลิตดังนั้นการปรับปรุงตารางการผลิตของงาน เพื่อต้องการลดเวลาดังกล่าวลงจำเป็นจะต้องพิจารณาการใช้ประโยชน์ของทรัพยากรการผลิตให้เกิดประโยชน์สูงสุด โดยลดช่วงเวลา (Idle time) หรือเวลารอคอยที่ไม่จำเป็น

2. เพื่อลดการรอคอยในกระบวนการผลิต คือ การลดจำนวนงานที่ต้องรอคอยโดยเฉลี่ยลงในขณะที่เครื่องจักรหรือคนยังต้องทำงานอยู่กับงานอื่นๆ

3. เพื่อลดความล่าช้าของงานลง ในกรณีทำงานไม่เสร็จตามกำหนดเวลาส่งมอบ แนวทางการลดความล่าช้าอาจจะทำได้โดยลดความล่าช้าสูงสุด (Max tardiness) การลดจำนวนงานที่ล่าช้าลง (Number of tardy tasks) หรือ การลดความล่าช้าเฉลี่ย (Mean tardiness)

เนื่องจากความต้องการลดต้นทุนการผลิตและต้องการใช้เวลาในการผลิตให้สั้นที่สุดเพื่อสร้างความพึงพอใจให้กับลูกค้า ดังนั้นสำหรับระบบการผลิตแบบต่อเนื่อง(Flow Shop) ซึ่งมีกระบวนการผลิตหลายขั้นตอนซึ่งต่อเนื่องกันไปตลอด การจัดลำดับงานจึงเป็นสิ่งสำคัญที่จะส่งผลกระทบต่อเวลาที่ใช้ทั้งหมดในการผลิต และด้วยจุดประสงค์ที่ต้องการจะใช้เวลาในการผลิตในระบบการผลิตแบบต่อเนื่อง(Flow Shop) ให้น้อยที่สุดนี้ จึงมีผู้สนใจศึกษาและพัฒนาวิธีการจัดลำดับงานในระบบการผลิตแบบ โฟลว์ช็อป (Flow Shop) กันอย่างต่อเนื่อง โดยแนวทางในการพัฒนาจะมุ่งไปที่การหาคำตอบที่ดีที่สุดและการหาคำตอบโดยใช้เวลาที่สั้นที่สุด ถ้าพิจารณาให้ลึกซึ้งแล้วจะพบว่าการลงทุนเพื่อที่จะจัดลำดับการทำงาน (Scheduling) ที่ดี เป็นการลงทุนที่คุ้มค่า เนื่องจากเกิค่าใช้จ่ายในการจัดลำดับการทำงาน (Scheduling) เพียงเล็กน้อย แต่การจัดลำดับการทำงาน (Scheduling) ที่ดีจะส่งผลให้สามารถลดต้นทุนการผลิตลดเวลาในการผลิต และสร้างความพึงพอใจให้กับลูกค้าได้เป็นอย่างดี

สำหรับการจัดลำดับงานแบบต่อเนื่อง(Flow Shop) เราสามารถเริ่มศึกษาจากทฤษฎีพื้นฐานที่ใช้ในการจัดลำดับงาน โดยทำการศึกษาโฟลว์ช็อปซึ่งมี 2 เครื่องจักร (Two-Machine Flow Shop) ซึ่งสามารถใช้กฎของจอห์นสัน (Johnson's Rule) จัดลำดับงานโดยใช้เวลาจัดที่รวดเร็วและได้คำตอบที่ดีที่สุด (Optimal) สำหรับปัญหาโฟลว์ช็อปซึ่งมี 2 เครื่องจักร (Two-Machine Flow Shop)

ในสถานการณ์ที่เวลาที่ใช้ในการทำงาน (Processing Time) ไม่แน่นอนนั้น Kouvelis et al. (2000) ได้ทำการศึกษาวิธีการดังกล่าว โดยทำการสร้างวิธีการในการปรับปรุง (Procedure Improve) ขึ้นมาในการแก้ปัญหา และ Cheng and Wang (1999) ได้หาวิธีจัดลำดับงานโดยมีจุดมุ่งหมายเพื่อหาเวลาที่น้อยที่สุด สำหรับ โฟลว์ช็อปซึ่งมี 2 เครื่องจักร (Two-Machine Flow Shop) โดยมีวัตถุประสงค์เพื่อนำปัญหาซึ่งเป็น NP-Complete มาทำ การพิจารณาทีละส่วนโดยมองเป็นปัญหาแบบโพลิโนเมียล (Polynomial) โดยการคิดแยกส่วนนี้ทำให้เราสามารถมองข้อจำกัดของปัญหา

แต่ละอย่างได้ชัดเจน นอกจากนี้ Elmaghraby and Thoney (1999) ได้นำกฎของจอห์นสัน (Johnson's Rule) มาเป็นคำตอบเริ่มต้นของการหาคำตอบเฉพาะที่ (local search) ที่เกิดขึ้นมาซึ่งเป็นวิธีการหาคำตอบ ที่มีแนวทางคล้ายวิธีการหาคำตอบแบบเจเนติก (Genetic Algorithm) กล่าวคือ มีรุ่นลูกซึ่งสืบเนื่องมาจากรุ่นพ่อแม่ โดย Elmaghraby ได้สร้างวิธีการหาคำตอบดังกล่าว เพื่อนำมาใช้กับการจัดลำดับงานแบบสโตคาสติกโฟลว์ช็อปซึ่งมี 2 เครื่องจักร (Two-Machine Stochastic Flowshop) ซึ่งประกอบด้วยขั้นตอนการทำงานสองขั้นตอน โดยมีเวลาที่ใช้ในการทำงาน (Processing Time) ไม่แน่นอน

ในส่วนของฮิวริสติก (Heuristic) นั้นวิธีพันธุกรรมอัลกอริทึม (Genetic Algorithm) เป็นวิธีที่ได้รับความสนใจในการนำมาประยุกต์ใช้สำหรับจัดลำดับงานในการผลิตแบบต่อเนื่อง (Flow Shop) อย่างมาก โดย Jain and Elmaghraby (1997) และ Yamada and Reeves (1997) ได้นำ วิธีการหาคำตอบแบบเจเนติก (Genetic Algorithm) มาประยุกต์ใช้ในการจัดลำดับงานในการผลิตแบบต่อเนื่อง (Flow Shop) โดยมีจุดมุ่งหมายเพื่อหาค่าที่ต่ำที่สุดของเวลาที่ใช้ในการผลิตและค่าเฉลี่ยที่งานอยู่ในระบบการผลิต (Minimizing the Makespan and Mean Flow Time) รวมถึงได้ทำการศึกษาค่าพารามิเตอร์ ของวิธีการหาคำตอบแบบเจเนติก (Genetic Algorithm)

วิธีหนึ่งซึ่งเป็นวิธีคลาสสิกซึ่งนิยมนำมาใช้ในเรื่องของค้นหาคำตอบเฉพาะที่ (local search) ทั้งหลายก็คือวิธีการจำลองแอนนีลิ่ง (Simulated Annealing) โดยในด้านการจัดลำดับงาน (Scheduling) นั้น McMullen and Frazier (2000) ได้ทำการศึกษา สิมูเลเทด แอนนีลิ่ง เพื่อนำมาใช้ในการจัดลำดับงานโดยทำการศึกษาค่าพารามิเตอร์ (Parameter) ที่มีผลต่อคำตอบที่ได้ ซึ่งค่าพารามิเตอร์ ที่ทำการศึกษาได้แก่อัตราการเย็นตัว (Cooling Rate) การเกิดซ้ำในแต่ละอุณหภูมิ (Iterations for Each Temperature Level) และ เปอร์เซนต์ค่าเริ่มต้น (% Initial)

การค้นหาคำตอบเฉพาะที่ (local search) อีกวิธีหนึ่งที่นิยมใช้กันก็คือ วิธีการค้นหาทาบู (Tabu Search) เป็นเทคนิคในการจัดตารางการผลิต ที่ Al-Turki *et al.*, (2001) นำไปใช้จัดตารางการผลิตแบบเครื่องจักรเดียว (Single machine scheduling) โดยนำไปเทียบกับวิธีการจำลองแอนนีลิ่ง (Simulated Annealing) ผลที่ได้ปรากฏว่าคำตอบที่ได้จากวิธีการค้นหาทาบู นั้นดีกว่าวิธีการจำลองแอนนีลิ่ง (Simulated Annealing) โดยมีค่าต่างจากคำตอบที่ดีที่สุด (Optimal) อยู่ที่ 0.001%

ในการแก้ปัญหาการจ่ายโหลดอย่างประหยัดนั้น Khamssawang (2003) ใช้วิธีการค้นหาทาง ซึ่งเป็นวิธีการค้นหาค่าที่เหมาะสมที่สุดที่กำลังได้รับความนิยมในปัจจุบันเพราะเป็นวิธีที่สามารถหลีกเลี่ยงคำตอบที่เหมาะสมที่สุดเฉพาะถิ่น โดยอาศัยการจดจำช่วงสั้นๆ ของคำตอบที่ผ่านมาแล้ว และใช้ประโยชน์จากการจดจำคำตอบเหล่านี้ไปสู่การค้นหาคำตอบที่เหมาะสมที่สุดแบบทั่วๆ ไป อย่างมีประสิทธิภาพ ในงานวิจัยนี้ทำการทดสอบการแก้ปัญหาการจ่ายโหลดอย่างประหยัดใน 4 กรณีศึกษา ทำการเปรียบเทียบผลลัพธ์ของคำตอบกับวิธีการต่างๆ เช่น วิธีหาคำตอบเชิงพันธุกรรม วิธีฮอปฟิลด์ นิวรอลเน็ตเวิร์ค วิธีการจำลองแอนนีลิ่ง และวิธีเทากันของแลมปีดา ในกรณีศึกษาทั้งหมด ทำการเปรียบเทียบผลลัพธ์ของคำตอบ และจากผลการดำเนินงานพบว่าวิธีการแก้ปัญหาการจ่ายโหลดอย่างประหยัดโดยใช้วิธีการค้นหาทาง พบว่าวิธีการค้นหาทางให้คำตอบที่เหมาะสมที่สุดดีกว่าทั้ง 5 วิธีดังกล่าวและเวลาที่ใช้ในการประมวลผลสัมฤทธิ์สมผล

จากวิธีการจัดลำดับงานแบบต่อเนื่อง(Flow Shop) ข้างต้น จะพบว่าแต่ละวิธีต่างมีจุดมุ่งหมายที่จะค้นหาใกล้เคียงค่าที่ดีที่สุด (Optimum) ให้มากที่สุด โดยพยายามใช้เวลาน้อยที่สุด โดยแต่ละวิธีการ (Algorithm) ก็จะมีข้อดีข้อเสียแตกต่างกันไป ซึ่งโดยทั่วไปแล้ว วิธีการที่มีจุดมุ่งหมายเน้นหนักในด้านการค้นหาใกล้เคียงค่าที่ดีที่สุด (Optimum) ก็จะมีขั้นตอนในการค้นหาที่ละเอียดซับซ้อน ซึ่งส่งผลให้เวลาที่ใช้ในการหาคำตอบค่อนข้างมากจึงไม่เหมาะกับปัญหาที่มีขนาดใหญ่ ส่วนวิธีการทางฮิวริสติก ที่เน้นหนักในด้านความรวดเร็วในการค้นหาคำตอบ ส่วนทางด้านด้านคุณภาพคำตอบก็อยู่ในค่าที่ยอมรับได้ อีกทั้งการเพิ่มเวลาในการค้นหาคำตอบนั้น ก็ส่งผลให้คำตอบที่ได้มีคุณภาพที่สูงขึ้นด้วย วิธีการจึงเหมาะกับปัญหาที่มีขนาดใหญ่

อุปกรณ์และวิธีการ

อุปกรณ์

1. คอมพิวเตอร์ฮาร์ดแวร์ (Computer Hardware)

1. คอมพิวเตอร์ส่วนบุคคล (Personal Computer)

2. เครื่องคอมพิวเตอร์ที่ใช้งานวิจัยเป็นคอมพิวเตอร์ที่ติดตั้งให้สามารถทำงานแบบคู่ขนาน (Parallel Computer) โดยใช้คอมพิวเตอร์ความเร็ว Pentium III 1GHz Ram 256 MB ของห้องคอมพิวเตอร์ ภาควิชาวิศวกรรมอุตสาหกรรม มหาวิทยาลัยเกษตรศาสตร์

2. คอมพิวเตอร์ซอฟต์แวร์ (Computer Software)

ในการประมวลผลแบบขนานในที่นี้จะเลือกใช้ซอฟต์แวร์ที่ช่วยในการจัดการ คือ MPICH 1.2.5 ซอฟต์แวร์ที่ใช้เขียนแอปพลิเคชัน คือ Visual C++ 6.0 ส่วนระบบปฏิบัติการที่ใช้สามารถเลือกได้ 2 ระบบ คือ

1. Windows XP โดยแต่ละเครื่องในเครือข่ายจะต้องเปิดการ Sharing และ Log in โดยใช้ Username และ Password (ห้าม Blank Password) เดียวกัน แอปพลิเคชันที่จะใช้งานจะต้องคัดลอกลงทุกเครื่องในไคลเอนต์เดียวกัน ในการประมวลผลจะส่งงานและรับข้อมูลจากหน่วยประมวลผลหลักเพียงหน่วยเดียวโดยระบบปฏิบัติการที่ใช้จะต้องยกเลิก Firewall เพราะ MPICH 1.2.5 ไม่สนับสนุนการทำงานร่วมกับ Firewall

2. Windows Server2003 แอปพลิเคชันที่จะใช้งานคัดลอกลงในไคลเอนต์ที่ใช้ร่วมกัน ในการประมวลผลจะส่งงานและรับข้อมูลจากหน่วยประมวลผลหลักเพียงหน่วยเดียว

วิธีการ

วิธีการดำเนินงานในวิทยานิพนธ์ฉบับนี้ นำเสนอแนวทางการประยุกต์ใช้คอมพิวเตอร์แบบคู่ขนานในการค้นหาทาง ในการจัดตารางการผลิตแบบต่อเนื่อง โดยวิธีการดังกล่าวมีขั้นตอนดังนี้

1. ศึกษาวิธีค้นหาทาง (Tabu Search)

2. เขียนโปรแกรมการจัดตารางการผลิตแบบโพลีชีป โดยใช้วิธีค้นหาทางในการหาคำตอบ

3. ทดสอบโปรแกรมการจัดตารางการผลิตแบบต่อเนื่องโดยใช้วิธีค้นหาทาง เพื่อศึกษาพฤติกรรมและความสัมพันธ์ของข้อมูลที่ส่งระหว่างกัน จากนั้นพิจารณาจากข้อมูลที่เก็บเพื่อตัดสินใจว่าส่วนใดของโปรแกรมที่ควรทำระบบขนาน

4. พัฒนาการทำงานของโปรแกรมการจัดตารางการผลิต โดยใช้ระบบคอมพิวเตอร์แบบคู่ขนาน

5. ทดสอบและประเมินผลโปรแกรมการจัดตารางการผลิต โดยใช้ระบบคอมพิวเตอร์แบบคู่ขนาน

โดยขั้นตอนจะกล่าวละเอียดดังต่อไปนี้

1. ศึกษาวิธีค้นหาทาง (Tabu Search)

คำว่า Tabu หรือ Taboo นั้นหมายถึงข้อห้าม หรือ การห้าม ดังนั้นวิธีการค้นหาคำตอบแบบทาง จึงหมายถึงวิธีการค้นหาที่มีข้อห้าม หรือการห้ามการค้นหาคำตอบในบางขอบเขตในวิธีการห้ามดังกล่าวนั้นจะเป็นการห้ามเพื่อที่จะช่วยให้ไม่ต้องไปค้นหาคำตอบเดิม หรือเกิดการวนรอบ (cyclic) การค้นหาคำตอบซึ่งอยู่ในขอบเขตของการค้นหาเดิม ซึ่งจะส่งผลให้ไม่สามารถหาคำตอบที่ดีขึ้นได้

1.1 ขั้นตอนการหาคำตอบของวิธีค้นหาทาบู

วิธีค้นหาทาบูประกอบไปด้วยรูปแบบการค้นหาคำตอบ 2 รูปแบบที่สำคัญคือ การค้นหาคำตอบโดยใช้หน่วยความจำระยะสั้น (Short Term Memory) และการค้นหาคำตอบโดยใช้หน่วยความจำระยะยาว (Long Term Memory) ซึ่งการค้นหาคำตอบโดยใช้หน่วยความจำระยะสั้นถือเป็นหน่วยความจำตามเวลา (Recency Base Memory) หมายถึงการค้นหาที่จดจำอดีตหรือประสบการณ์การค้นหาที่ผ่านมาเพียงระยะสั้นในทางตรงข้ามการค้นหาคำตอบโดยใช้หน่วยความจำระยะยาวซึ่งถือเป็นหน่วยความจำตามความถี่ (Frequency Base Memory) หมายถึงการค้นหาคำตอบที่ต้องจดจำอดีตหรือประสบการณ์ที่ผ่านมาเพื่อช่วยให้การค้นหาคำตอบเป็นไปอย่างมีประสิทธิภาพมากขึ้น โดยมีขั้นตอนการทำงานดังนี้

1. กำหนดค่าพารามิเตอร์ คือการเลือกใช้เกณฑ์ความปรารถนา (Aspiration Criteria) และขนาดการเป็นทาบู (Tabu List Size)

2. สร้างคำตอบเริ่มต้น โดยใช้การสุ่ม (Random)

3. กำหนดให้คำตอบปัจจุบันที่มีอยู่เป็นคำตอบที่ดีที่สุด

4. สร้างเซตของคำตอบข้างเคียง (Neighborhood) โดยจำนวนเซตคำตอบข้างเคียงเท่ากับ

$$\left[\frac{N \times (N-1)}{2} \right] \text{ เซ็ต}$$

ลำดับงานเริ่มต้น: 1, 2, 3, 4, 5

สร้างเซตของคำตอบข้างเคียง(Neighborhood) ได้แก่ {1, 2}, {1, 3}, {1, 4}, {1, 5}, {2, 3}, {2, 4}, {2, 5}, {3, 4}, {3, 5}, {4, 5}

ลำดับงานที่ 1: 2, 1, 3, 4, 5

ลำดับงานที่ 2: 3, 2, 1, 4, 5

ลำดับงานที่ 3: 4, 2, 3, 1, 5

ลำดับงานที่ 4: 5, 2, 3, 4, 1

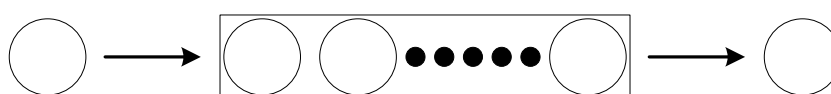
ลำดับงานที่ 5: 1, 3, 2, 4, 5

: : : : : : :

ลำดับงานสุดท้าย: 1, 2, 3, 5, 4

คำนวณค่าตอบของเซตคำตอบข้างเคียงทั้งหมดด้วยสมการเป้าหมาย (Objective Function)

5. ตรวจสอบรายการทาบ (Tabu List) เป็นรายการที่คอยบันทึกข้อมูลของกระบวนการค้นหาในอดีต ถ้าคำตอบข้างเคียงถูกเป็นทาบเกินระยะเวลาที่กำหนดให้ยกเลิกการเป็นทาบของคำตอบข้างเคียงนั้น



ภาพที่ 3 ลักษณะของรายการทาบ (Tabu List)

6. ดำเนินการค้นหาคำตอบจากกลุ่มคำตอบข้างเคียง โดยคำตอบข้างเคียงนั้นจะต้องไม่ถูกเป็นทาบและเป็นคำตอบที่ดีกว่าหรือเท่ากับคำตอบที่ดีที่สุด

7. แต่ถ้าคำตอบข้างเคียงนั้นถูกเป็นทาบแต่สามารถให้ คำตอบที่ดีที่ไม่เคยพบมาก่อนก็สามารถเลือกเป็นคำตอบได้

8. ถ้าคำตอบข้างเคียงสามารถให้คำตอบที่ดีกว่าหรือเท่ากับคำตอบที่ดีที่สุดได้ ให้คำตอบข้างเคียงนั้นเป็นทาบและนำมาแทนคำตอบที่ดีที่สุดและให้คำตอบข้างเคียงนี้เป็นคำตอบปัจจุบัน

9. แต่ถ้าคำตอบข้างเคียงสามารถไม่สามารถให้คำตอบที่ดีกว่าได้ ให้คำตอบข้างเคียงนั้นเป็นทาบและนำมาแทนคำตอบปัจจุบัน แต่ไม่ต้องปรับปรุงคำตอบที่ดีที่สุด

10. เริ่มต้นทำซ้ำตั้งแต่ข้อ 4 ลงมาเรื่อยๆ จนกระทั่งได้คำตอบที่ต้องการหรือครบจำนวนรอบ (Iteration) ที่กำหนด

1.2 วิธีการทำงานของวิธีการค้นหาทาง กรณี ปัญหาการจัดตารางการผลิตแบบต่อเนื่อง 5 job 5 machine

ตารางที่ 1 ข้อมูลเวลาที่ใช้ในแต่ละเครื่องจักร ของปัญหาการจัดตารางการผลิตแบบต่อเนื่อง ขนาด 5×5

งาน	M1	M2	M3	M4	M5
1	19	22	3	7	11
2	10	8	17	16	14
3	31	2	7	5	21
4	20	16	6	14	3
5	3	14	12	11	17

จากข้อมูลปัญหาการจัดตารางการผลิตแบบต่อเนื่องดังตารางที่ 1 กำหนดให้พารามิเตอร์ ขนาดการเป็นทาบ (Tabu Size) เท่ากับ 3 และจำนวนรอบการประมวลผลเท่ากับ 10 กำหนดให้ ลำดับงานเริ่มต้นจากการสุ่มดังนี้คือ $\{1, 5, 3, 2, 4\}$ โดยมีเวลาการทำงานรวมเท่ากับ 133 และเซต ลำดับงานข้างเคียงจะมี 10 ลักษณะ คือ $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}$

โดยที่ $\{1, 2\}$ คือ การสลับระหว่างลำดับงานที่ 1 กับลำดับงานที่ 2
 $\{1, 3\}$ คือ การสลับระหว่างลำดับงานที่ 1 กับลำดับงานที่ 3
 $\{1, 4\}$ คือ การสลับระหว่างลำดับงานที่ 1 กับลำดับงานที่ 4
 $\{1, 5\}$ คือ การสลับระหว่างลำดับงานที่ 1 กับลำดับงานที่ 5
 $\{2, 3\}$ คือ การสลับระหว่างลำดับงานที่ 2 กับลำดับงานที่ 3
 $\{2, 4\}$ คือ การสลับระหว่างลำดับงานที่ 2 กับลำดับงานที่ 4
 $\{2, 5\}$ คือ การสลับระหว่างลำดับงานที่ 2 กับลำดับงานที่ 5
 $\{3, 4\}$ คือ การสลับระหว่างลำดับงานที่ 3 กับลำดับงานที่ 4
 $\{3, 5\}$ คือ การสลับระหว่างลำดับงานที่ 3 กับลำดับงานที่ 5
 $\{4, 5\}$ คือ การสลับระหว่างลำดับงานที่ 4 กับลำดับงานที่ 5

โดยวิธีการทำงานของวิธีการค้นหาทาง กับ ปัญหาการจัดตารางการผลิตแบบต่อเนื่องจะแสดงรายละเอียดการค้นหาในแต่ละรอบดังตารางที่ 2 ถึงตารางที่ 11

ตารางที่ 2 วิธีการทำงานรอบประมวลผลที่ 1 ของวิธีการค้นหาทางของปัญหาการจัดตารางการผลิตแบบต่อเนื่องขนาด 5×5

Iteration 1						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	1	5	3	2	4	133		
ลำดับงานเริ่มต้น	1	5	3	2	4	133		
ลำดับงานที่ 1 {1,2}	2	5	3	1	4	124		
ลำดับงานที่ 2 {1,3}	3	5	1	2	4	133		
ลำดับงานที่ 3 {1,4}	4	5	3	2	1	136		
ลำดับงานที่ 4 {1,5}	5	1	3	2	4	122		T
ลำดับงานที่ 5 {2,3}	1	5	2	3	4	138		
ลำดับงานที่ 6 {2,4}	1	5	3	4	2	144		
ลำดับงานที่ 7 {2,5}	1	2	3	5	4	137		
ลำดับงานที่ 8 {3,4}	1	5	4	2	3	147		
ลำดับงานที่ 9 {3,5}	1	3	5	2	4	129		
ลำดับงานที่ 10 {4,5}	1	4	3	2	5	152		

รอบการประมวลผลที่ 1 (Iteration 1) สร้างคำตอบเริ่มต้นโดยใช้การสุ่ม (Random) กำหนดให้คำตอบเริ่มต้นที่มีอยู่เป็นคำตอบที่ดีที่สุด เท่ากับ 133 จากนั้นดำเนินการสร้างเซตของคำตอบข้างเคียงได้ลำดับงาน 10 รูปแบบ และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด (Minimize Makespan) เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 4 เท่ากับ 122 แล้วปรับปรุงรายการหาในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบู (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบูของลำดับงานที่ 4 ตามข้อจำกัดการเป็นทาบู (Tabu Restriction)

ตารางที่ 3 วิธีการทำงานรอบประมวลผลที่ 2 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่องขนาด 5×5

Iteration 2						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	5	1	3	2	4	122		
ลำดับงานเริ่มต้น	5	1	3	2	4	122		
ลำดับงานที่ 1 {1,2}	5	2	3	1	4	124		
ลำดับงานที่ 2 {1,3}	5	3	1	2	4	133		
ลำดับงานที่ 3 {1,4}	5	4	3	2	1	130		
ลำดับงานที่ 4 {1,5}	1	5	3	2	4	133	1	T
ลำดับงานที่ 5 {2,3}	5	1	2	3	4	123		
ลำดับงานที่ 6 {2,4}	5	1	3	4	2	144		
ลำดับงานที่ 7 {2,5}	2	1	3	5	4	122		T
ลำดับงานที่ 8 {3,4}	5	1	4	2	3	136		
ลำดับงานที่ 9 {3,5}	3	1	5	2	4	148		
ลำดับงานที่ 10 {4,5}	4	1	3	2	5	152		

รอบการประมวลผลที่ 2 (Iteration 2) คำตอบข้างเคียงที่ถูกเลือกดีกว่าคำตอบที่ดีที่สุด ทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบที่ดีที่สุด เท่ากับ 122 แล้วกำหนดให้คำตอบที่ดีที่สุดเป็นคำตอบเริ่มต้นเท่ากับ 122 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 7 เท่ากับ 122 แล้วปรับปรุงรายการทามูในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทามู (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทามูของลำดับงานที่ 7 และทำการเพิ่มจำนวนครั้งในการเป็นทามูของลำดับงานที่ 4 ตามข้อจำกัดการเป็นทามู (Tabu Restriction)

ตารางที่ 4 วิธีการทำงานรอบประมวลผลที่ 3 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่องขนาด 5×5

Iteration 3						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	5	1	3	2	4	122		
ลำดับงานเริ่มต้น	2	1	3	5	4	122		
ลำดับงานที่ 1 {1,2}	1	2	3	5	4	137		
ลำดับงานที่ 2 {1,3}	2	3	1	5	4	139		
ลำดับงานที่ 3 {1,4}	2	4	3	5	1	129		T
ลำดับงานที่ 4 {1,5}	2	5	3	1	4	124	2	T
ลำดับงานที่ 5 {2,3}	3	1	2	5	4	147		
ลำดับงานที่ 6 {2,4}	4	1	3	5	2	146		
ลำดับงานที่ 7 {2,5}	5	1	3	2	4	122	1	T
ลำดับงานที่ 8 {3,4}	2	1	4	5	3	142		
ลำดับงานที่ 9 {3,5}	2	1	5	3	4	129		
ลำดับงานที่ 10 {4,5}	2	1	3	4	5	150		

รอบการประมวลผลที่ 3 (Iteration 3) คำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 122 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 3 เท่ากับ 129 แล้วปรับปรุงรายการทาบูนในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบูน (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบูนของลำดับงานที่ 3 และทำการเพิ่มจำนวนครั้งในการเป็นทาบูนของลำดับงานที่ 4 และลำดับงานที่ 7 ตามข้อจำกัดการเป็นทาบูน (Tabu Restriction)

ตารางที่ 5 วิธีการทำงานรอบประมวลผลที่ 4 ของวิธีการค้นหาทางของปัญหาการจัดตารางการผลิตแบบต่อเนื่องขนาด 5×5

Iteration 4						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	5	1	3	2	4	122		
ลำดับงานเริ่มต้น	2	4	3	5	1	129		
ลำดับงานที่ 1 {1,2}	1	4	3	5	2	146		
ลำดับงานที่ 2 {1,3}	2	4	1	5	3	146		
ลำดับงานที่ 3 {1,4}	2	1	3	5	4	122	1	T
ลำดับงานที่ 4 {1,5}	2	4	3	1	5	156	3	T
ลำดับงานที่ 5 {2,3}	3	4	2	5	1	150		
ลำดับงานที่ 6 {2,4}	4	2	3	5	1	140		
ลำดับงานที่ 7 {2,5}	5	4	3	2	1	130	2	T
ลำดับงานที่ 8 {3,4}	2	3	4	5	1	142		
ลำดับงานที่ 9 {3,5}	2	4	5	3	1	132		
ลำดับงานที่ 10 {4,5}	2	5	3	4	1	126		T

รอบการประมวลผลที่ 4 (Iteration 4) คำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 129 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 10 เท่ากับ 126 แล้วปรับปรุงรายการทาบูนในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบูน (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบูนของลำดับงานที่ 10 และทำการเพิ่มจำนวนครั้งในการเป็นทาบูนของลำดับงานที่ 3 ลำดับงานที่ 4 และลำดับงานที่ 7 ตามข้อจำกัดการเป็นทาบูน (Tabu Restriction)

ตารางที่ 6 วิธีการทำงานรอบประมวลผลที่ 5 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตโฟลชีอขนาด 5×5

Iteration 5						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	5	1	3	2	4	122		
ลำดับงานเริ่มต้น	2	5	3	4	1	126		
ลำดับงานที่ 1 {1,2}	1	5	3	4	2	144		
ลำดับงานที่ 2 {1,3}	2	5	1	4	3	118		T
ลำดับงานที่ 3 {1,4}	2	5	3	1	4	124	2	T
ลำดับงานที่ 4 {1,5}	2	1	3	4	5	150		
ลำดับงานที่ 5 {2,3}	3	5	2	4	1	126		
ลำดับงานที่ 6 {2,4}	4	5	3	2	1	136		
ลำดับงานที่ 7 {2,5}	5	2	3	4	1	126	3	T
ลำดับงานที่ 8 {3,4}	2	5	4	3	1	126		
ลำดับงานที่ 9 {3,5}	2	3	5	4	1	126		
ลำดับงานที่ 10 {4,5}	2	4	3	5	1	129	1	T

รอบการประมวลผลที่ 5 (Iteration 5) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 4 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้น และคำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 126 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 2 เท่ากับ 118 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 2 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 3 ลำดับงานที่ 7 และลำดับงานที่ 10 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction)

ตารางที่ 7 วิธีการทำงานรอบประมวลผลที่ 6 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่อง ขนาด 5×5

Iteration 6						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	2	5	1	4	3	118		
ลำดับงานเริ่มต้น	2	5	1	4	3	118		
ลำดับงานที่ 1 {1,2}	1	5	2	4	3	140		
ลำดับงานที่ 2 {1,3}	2	5	3	4	1	126	1	T
ลำดับงานที่ 3 {1,4}	2	5	4	1	3	118	3	T
ลำดับงานที่ 4 {1,5}	2	1	5	4	3	129		
ลำดับงานที่ 5 {2,3}	3	5	1	4	2	146		
ลำดับงานที่ 6 {2,4}	4	5	1	2	3	148		
ลำดับงานที่ 7 {2,5}	5	2	1	4	3	118		T
ลำดับงานที่ 8 {3,4}	2	5	1	3	4	122		
ลำดับงานที่ 9 {3,5}	2	3	1	4	5	152		
ลำดับงานที่ 10 {4,5}	2	4	1	5	3	146	2	T

รอบการประมวลผลที่ 6 (Iteration 6) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 7 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้นและคำตอบข้างเคียงที่ถูกเลือกดีกว่าคำตอบที่ดีที่สุด ทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบที่ดีที่สุด เท่ากับ 118 แล้วกำหนดให้คำตอบที่ดีที่สุดเป็นคำตอบเริ่มต้น เท่ากับ 118 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วย ฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 7 เท่ากับ 118 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 7 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 2 ลำดับงานที่ 3 และลำดับงานที่ 10 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction)

ตารางที่ 8 วิธีการทำงานรอบประมวลผลที่ 7 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่องขนาด 5×5

Iteration 7						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	2	5	1	4	3	118		
ลำดับงานเริ่มต้น	5	2	1	4	3	118		
ลำดับงานที่ 1 {1,2}	5	1	2	4	3	125		
ลำดับงานที่ 2 {1,3}	5	2	3	4	1	126	2	T
ลำดับงานที่ 3 {1,4}	5	2	4	1	3	118		T
ลำดับงานที่ 4 {1,5}	1	2	5	4	3	137		
ลำดับงานที่ 5 {2,3}	5	3	1	4	2	146		
ลำดับงานที่ 6 {2,4}	5	4	1	2	3	140		
ลำดับงานที่ 7 {2,5}	2	5	1	4	3	118	1	T
ลำดับงานที่ 8 {3,4}	5	2	1	3	4	122		
ลำดับงานที่ 9 {3,5}	3	2	1	4	5	152		
ลำดับงานที่ 10 {4,5}	4	2	1	5	3	146	3	T

รอบการประมวลผลที่ 7 (Iteration 7) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 3 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้นและคำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 118 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 3 เท่ากับ 118 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 3 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 2 ลำดับงานที่ 7 และลำดับงานที่ 10 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction)

ตารางที่ 9 วิธีการทำงานรอบประมวลผลที่ 8 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่องขนาด 5×5

Iteration 8						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	2	5	1	4	3	118		
ลำดับงานเริ่มต้น	5	2	4	1	3	118		
ลำดับงานที่ 1 {1,2}	5	1	4	2	3	136		
ลำดับงานที่ 2 {1,3}	5	2	4	3	1	126	3	T
ลำดับงานที่ 3 {1,4}	5	2	1	4	3	118	1	T
ลำดับงานที่ 4 {1,5}	1	2	4	5	3	145		
ลำดับงานที่ 5 {2,3}	5	3	4	1	2	150		
ลำดับงานที่ 6 {2,4}	5	4	2	1	3	126		
ลำดับงานที่ 7 {2,5}	2	5	4	1	3	118	2	T
ลำดับงานที่ 8 {3,4}	5	2	3	1	4	124		T
ลำดับงานที่ 9 {3,5}	3	2	4	1	5	156		
ลำดับงานที่ 10 {4,5}	4	2	5	1	3	140		

การประมวลผลที่ 8 (Iteration 8) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 10 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้นและคำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 118 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 8 เท่ากับ 124 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 8 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 2 ลำดับงานที่ 3 และลำดับงานที่ 7 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction)

ตารางที่ 10 วิธีการทำงานรอบประมวลผลที่ 9 ของวิธีการค้นหาทางของปัญหาการจัดตารางการ
ผลิตแบบต่อเนื่องขนาด 5×5

Iteration 9						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	2	5	1	4	3	118		
ลำดับงานเริ่มต้น	5	2	3	1	4	124		
ลำดับงานที่ 1 {1,2}	5	1	3	2	4	122		T
ลำดับงานที่ 2 {1,3}	5	2	1	3	4	122		
ลำดับงานที่ 3 {1,4}	5	2	3	4	1	126	2	T
ลำดับงานที่ 4 {1,5}	1	2	3	5	4	137		
ลำดับงานที่ 5 {2,3}	5	3	2	1	4	124		
ลำดับงานที่ 6 {2,4}	5	4	3	1	2	150		
ลำดับงานที่ 7 {2,5}	2	5	3	1	4	124	3	T
ลำดับงานที่ 8 {3,4}	5	2	4	1	3	118	1	T
ลำดับงานที่ 9 {3,5}	3	2	5	1	4	127		
ลำดับงานที่ 10 {4,5}	4	2	3	1	5	156		

รอบการประมวลผลที่ 9 (Iteration 9) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 2 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้นและคำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 124 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือ เวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 1 เท่ากับ 122 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 1 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 3 ลำดับงานที่ 7 และลำดับงานที่ 8 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction)

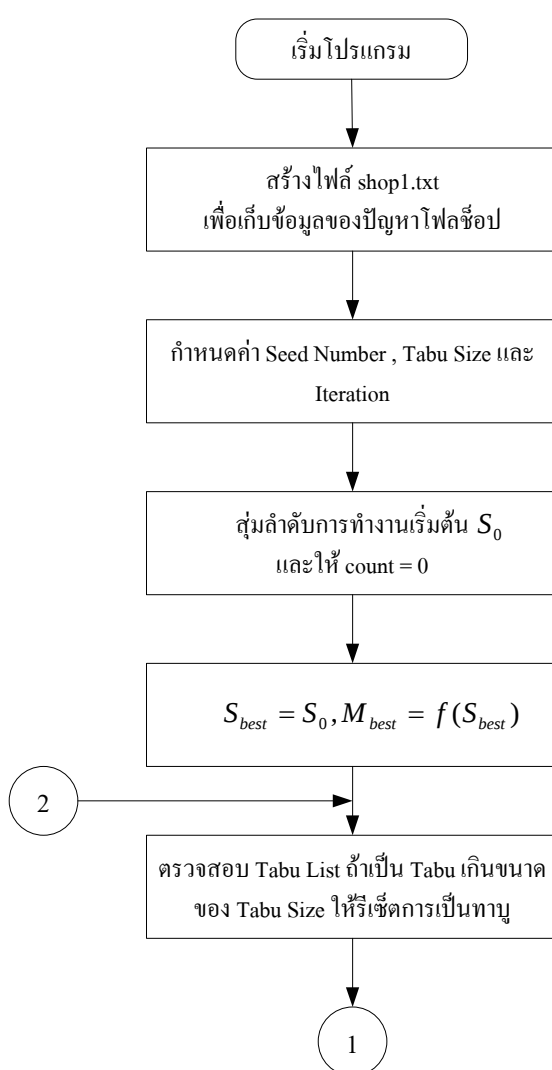
ตารางที่ 11 วิธีการทำงานรอบประมวลผลที่ 10 ของวิธีการค้นหาทางของปัญหาการจัดตารางการผลิตแบบต่อเนื่องขนาด 5×5

Iteration 10						Makespan	Tabu Restriction	Tabu Status
ลำดับงานที่ดีที่สุด	2	5	1	4	3	118		
ลำดับงานเริ่มต้น	5	1	3	2	4	122		
ลำดับงานที่ 1 {1,2}	5	2	3	1	4	124	1	T
ลำดับงานที่ 2 {1,3}	5	3	1	2	4	133		
ลำดับงานที่ 3 {1,4}	5	4	3	2	1	130	3	T
ลำดับงานที่ 4 {1,5}	1	5	3	2	4	133		
ลำดับงานที่ 5 {2,3}	5	1	2	3	4	123		T
ลำดับงานที่ 6 {2,4}	5	1	3	4	2	144		
ลำดับงานที่ 7 {2,5}	2	1	3	5	4	122		
ลำดับงานที่ 8 {3,4}	5	1	4	2	3	136	2	T
ลำดับงานที่ 9 {3,5}	3	1	5	2	4	148		
ลำดับงานที่ 10 {4,5}	4	1	3	2	5	152		

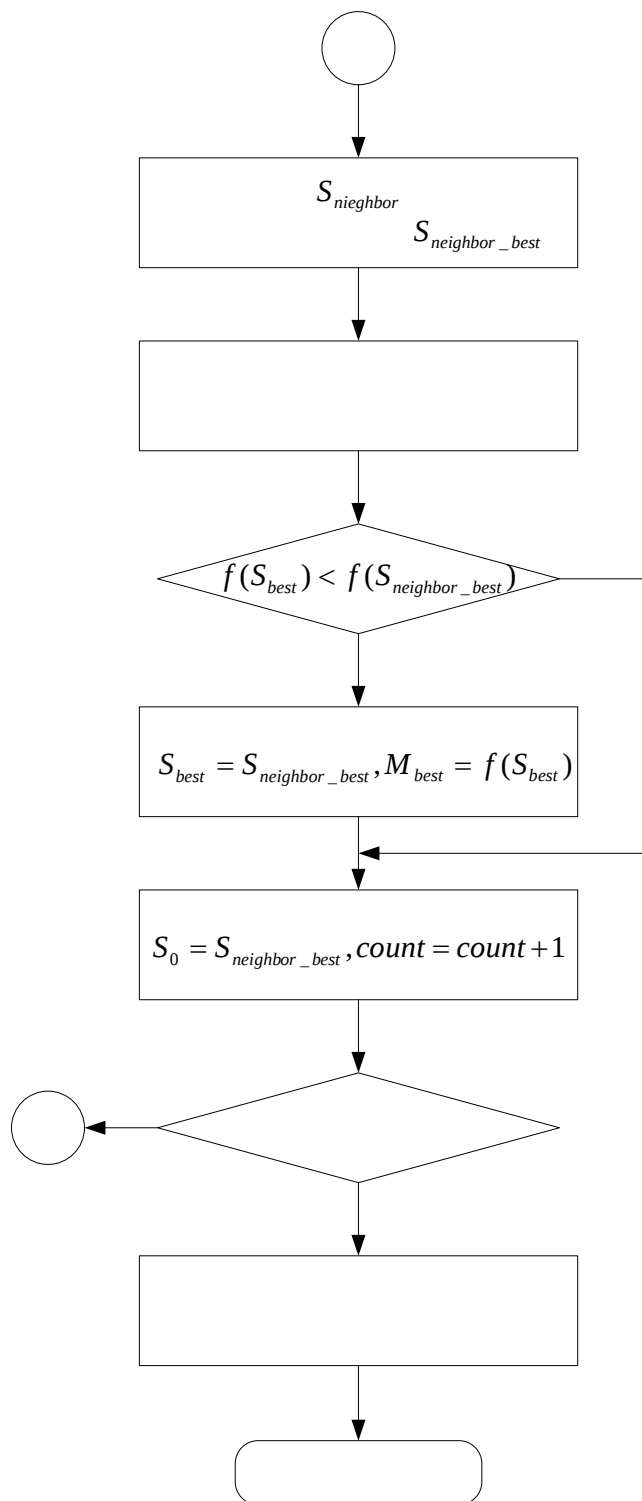
รอบการประมวลผลที่ 10 (Iteration 10) ตรวจสอบรายการทาบ (Tabu List) พบว่าลำดับงานที่ 7 มีสถานะเป็นทาบเท่ากับขนาดการเป็นทาบ (Tabu Size) ให้ยกเลิกสถานะการเป็นทาบของคำตอบข้างเคียงนั้นและคำตอบข้างเคียงที่ถูกเลือกไม่ได้ดีกว่าคำตอบที่ดีที่สุด จึงไม่จำเป็นต้องปรับปรุงคำตอบที่ดีที่สุด แล้วทำการย้ายตำแหน่งคำตอบข้างเคียงที่ถูกเลือกไปเป็นคำตอบเริ่มต้นเท่ากับ 122 จากนั้นสร้างเซตของคำตอบข้างเคียง และคำนวณคำตอบของลำดับงานทั้งหมดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) โดยมีสมการเป้าหมาย (Objective Function) คือเวลาในการทำงานรวมน้อยที่สุด เลือกคำตอบข้างเคียงที่ดีที่สุดคือ ลำดับงานที่ 5 เท่ากับ 123 แล้วปรับปรุงรายการทาบในตำแหน่งของคำตอบที่ถูกเลือกโดยบันทึกสถานะการเป็นทาบ (Tabu Status) ด้วยสัญลักษณ์ T แล้วเริ่มบันทึกจำนวนครั้งในการเป็นทาบของลำดับงานที่ 5 และทำการเพิ่มจำนวนครั้งในการเป็นทาบของลำดับงานที่ 1 ลำดับงานที่ 3 และลำดับงานที่ 8 ตามข้อจำกัดการเป็นทาบ (Tabu Restriction) และตรวจสอบการทำงานพบว่าครบจำนวนรอบการทำงานที่ตั้งไว้เท่ากับ 10 จึงหยุดกระบวนการทำงาน ลำดับงานที่ดีที่สุดจากการค้นหาคำตอบครั้งนี้คือ { 2, 5, 1, 4, 3} โดยใช้เวลาในการทำงานรวม (Makespan) เท่ากับ 118

2. เขียนโปรแกรมการจัดตารางการผลิตแบบโฟล์ช้อป โดยใช้วิธีการหาคำตอบแบบทามู

นำหลักการและแนวคิดของวิธีการหาคำตอบแบบทามูมาใช้ในการจัดตารางการผลิตแบบโฟล์ช้อป โดยมีวัตถุประสงค์ของโปรแกรมคือ สามารถหาลำดับงานที่ดีที่สุดที่ให้ เวลาในการทำงานรวมน้อยที่สุด (Minimize Makespan) มาพัฒนาเป็นโปรแกรมในการหาคำตอบ โดยมีขั้นตอนการทำงานของโปรแกรกดังนี้ และแสดงในภาพที่ 4 และภาพที่ 5



ภาพที่ 4 ขั้นตอนการทำงานของโปรแกรมจัดตารางการผลิตแบบโฟล์ช้อป โดยใช้วิธีการหาคำตอบแบบทามู



ภาพที่ 5 ขั้นตอนการทำงานของโปรแกรมจัดการตารางการผลิตแบบโพลีชีป โดยใช้วิธีการหาคำตอบแบบทวน

1. สร้างไฟล์ shop1.txt โดยในไฟล์จะมีข้อมูลของปัญหาโพลีซ้อป ประกอบไปด้วย
จำนวนงาน N งาน ที่ถูกผลิตบนเครื่องจักร M เครื่อง และเวลาของแต่ละงาน บนแต่ละเครื่องจักร
เพื่อนำไปใช้เป็นข้อมูลของโปรแกรม
2. กำหนดค่าต่างๆที่เป็นเงื่อนไขในการทำงานของโปรแกรม คือ ค่าเริ่มต้น(Seed Number)
ขนาดของทาบู (Tabu Size) และจำนวนรอบที่ใช้ในการหาคำตอบ (Iteration)
3. ทำการสร้างลำดับการทำงานเริ่มต้น S_0 โดยการสุ่ม แล้วเริ่มในการค้นหาคำตอบของ
โปรแกรม (count = 0)
4. ให้ลำดับการทำงานเริ่มต้น S_0 เป็นลำดับการทำงานที่ดีที่สุด S_{best} แล้วคำนวณคำตอบ
ของลำดับงานที่ดีที่สุดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan : M) แล้วกำหนดให้
เป็น เวลาในการทำงานรวมน้อยที่สุด (M_{best})
5. ตรวจสอบลำดับงานที่สถานะเป็นทาบู ว่าครบกำหนดการเป็นทาบูตามขนาดของ Tabu Size ถ้า
ครบแล้วให้ยกเลิกสถานะของลำดับงานนี้ไม่ให้เป็นทาบู
6. สร้างเซตคำตอบของลำดับงานข้างเคียง ($S_{neighbor}$) ที่ไม่เป็นทาบูแล้วทำการหาเซตของ
ลำดับงานข้างเคียงที่ดีที่สุด ($S_{neighbor_best}$)
7. ทำการปรับปรุงรายการทาบู (Tabu List) โดยทำการเพิ่มจำนวนครั้งการเป็นทาบู ของ
ลำดับงานที่เป็นทาบูอยู่แล้ว และกำหนดให้สถานะลำดับงานข้างเคียงที่ดีที่สุดเป็นทาบู
8. เปรียบเทียบระหว่างเวลาในการทำงานรวมของลำดับงานข้างเคียงที่ดีที่สุด
 $f(S_{neighbor_best})$ กับเวลาในการทำงานรวมที่ดีที่สุด (M_{best}) ถ้าคำตอบข้างเคียงที่ได้แย่กว่าคำตอบ
ที่ดีที่สุดให้ข้ามไปข้อ 10
9. ถ้าคำตอบข้างเคียงดีกว่าคำตอบที่ดีที่สุด ให้แทนลำดับงานที่ดีที่สุด ด้วยลำดับงาน
ข้างเคียงที่ดีที่สุด แล้วคำนวณหาเวลาการทำงานรวม (M_{best})

10. กำหนดให้ลำดับงานข้างเคียงที่ดีที่สุด เป็นลำดับงานเริ่มต้น ($S_0 = S_{neighbor}$) เพื่อใช้เป็นคำตอบเริ่มต้นในการหาคำตอบรอบต่อไป แล้วทำการเพิ่มค่า count ซึ่งหมายถึงจบกระบวนการ 1 รอบการทำงาน

11. ตรวจสอบว่าครบตามจำนวนรอบที่กำหนด(Iteration) ไว้หรือยัง ถ้ายังไม่ครบตามที่ตั้งไว้ให้ย้อนกลับไปทำที่ข้อ 5

12. ถ้าครบตามจำนวนรอบที่กำหนดไว้แล้วให้สร้างไฟล์ “outsolution.txt” เพื่อเก็บผลของคำตอบที่ได้จากการประมวลผล

3. ทดสอบและเก็บข้อมูลการทำงานของโปรแกรม

เป็นการศึกษาพฤติกรรมและเก็บข้อมูลการทำงานของโปรแกรมในส่วนของเวลาที่ใช้ในการทำงานเพื่อพิจารณาศึกษาพฤติกรรมและเก็บข้อมูลการทำงานของโปรแกรม ในส่วนของเวลาที่ใช้ทำงาน และความสัมพันธ์ของข้อมูลที่ส่งระหว่างกัน จากนั้นพิจารณาจากข้อมูลที่เก็บเพื่อตัดสินใจว่าส่วนใดของโปรแกรมที่ควรทำระบบขนาน

โดยเมื่อนำโปรแกรมมาทดสอบการทำงานกับข้อมูลชนิดเดียวกัน แต่พิจารณาจำนวนรอบการทำงานที่ต่างกันจะได้ผลดังตารางที่ 12 และเมื่อนำโปรแกรมมาทดสอบการทำงานกับข้อมูลที่มีขนาดของปัญหาแตกต่างกัน แต่จำนวนรอบในการทำงานเท่ากันจะได้ผลดังตารางที่ 13

ตารางที่ 12 เวลาประมวลผลและคำตอบ ของปัญหาโฟลชีอปที่จำนวนรอบในการทำงานต่างกัน

ขนาด ของปัญหา	จำนวนรอบ การทำงาน	เวลาประมวลผล (วินาที)	Makespan C_{max}
50 x 10	1000	40.2	3093
	2000	79.6	3093
	4000	158.5	3085
	6000	238.7	3085
	8000	320.5	3071
	10000	400.7	3071

จากตารางที่ 12 พบว่าเมื่อพิจารณาจำนวนรอบการทำงาน(Iteration) ในกรณีขนาดของปัญหาเท่ากัน พบว่าเมื่อจำนวนรอบมากขึ้นย่อมส่งผลต่อเวลาที่ใช้ในการประมวลผลเพิ่มขึ้น และเมื่อพิจารณาเวลาในการทำงานรวม (Makespan) พบว่าคุณภาพของคำตอบก็ดีขึ้นด้วย แสดงให้เห็นว่าในการนำโปรแกรมไปใช้ในการจัดตารางการผลิตนั้น เพื่อที่จะให้ได้คุณภาพของคำตอบที่ดี ต้องกำหนดจำนวนรอบการทำงานมากที่สุด แต่เมื่อจำนวนรอบมากขึ้นก็ย่อมส่งผลต่อเวลาที่ใช้ในการประมวลผลเพิ่มขึ้นตามไปด้วย

ตารางที่ 13 เวลาประมวลผลและคำตอบ ของปัญหาการจัดตารางการผลิตแบบต่อเนื่องที่มีขนาดของปัญหาต่างกัน

ขนาด ของปัญหา	จำนวนรอบ การทำงาน	เวลาประมวลผล (วินาที)	Makespan $C_{\max}$
50 x 5	20000	364	2752
50 x 10	20000	738	3108
50 x 20	20000	1496	3899
100 x 5	20000	2911	5527
100 x 10	20000	6000	5825
100 x 20	20000	13680	6352
200 x 10	20000	56400	10983
200 x 20	20000	105480	11438

จากตารางที่ 13 พบว่าเมื่อพิจารณา ถึงขนาดของปัญหา ในกรณีที่จำนวนรอบการทำงานเท่ากัน แสดงให้เห็นว่าเมื่อขนาดของปัญหาใหญ่ขึ้น เวลาที่ใช้ในการประมวลผลก็เพิ่มขึ้น เนื่องมาจากวิธีการหาคำตอบแบบทามู ในส่วนของจำนวนเซตของลำดับงานข้างเคียงหาได้จาก $\left[\frac{N \times (N - 1)}{2} \right]$ โดยที่ N คือ จำนวนงาน และเมื่อนำมาใช้ในการพิจารณาปัญหาขนาด 50 job จะพบว่ามีเซตของลำดับงานข้างเคียง เท่ากับ 2448 เซต และเมื่อนำมาเปรียบเทียบกับปัญหาขนาด 200 job ที่มีเซตของลำดับงานข้างเคียง เท่ากับ 19900 เซต ส่งผลให้เวลาที่ใช้ในการประมวลผลเมื่อนำมาเปรียบเทียบกับกันนั้นต่างกันมาก

4. การปรับปรุงโปรแกรมการจัดตารางการผลิตแบบโพลีซ็อป โดยใช้วิธีการหาคำตอบแบบทามูให้ทำงานแบบขนาน

จากการพิจารณา ผลการทดสอบโปรแกรมจัดตารางการผลิตแบบต่อเนื่องที่ใช้วิธีหาคำตอบแบบทามูใน ตารางที่ 12 และ ตารางที่ 13 ถึงความสัมพันธ์ระหว่างเวลาที่ใช้ในการประมวลผล ขนาดของปัญหา จำนวนรอบการทำงาน และคุณภาพของคำตอบ สรุปได้ว่า

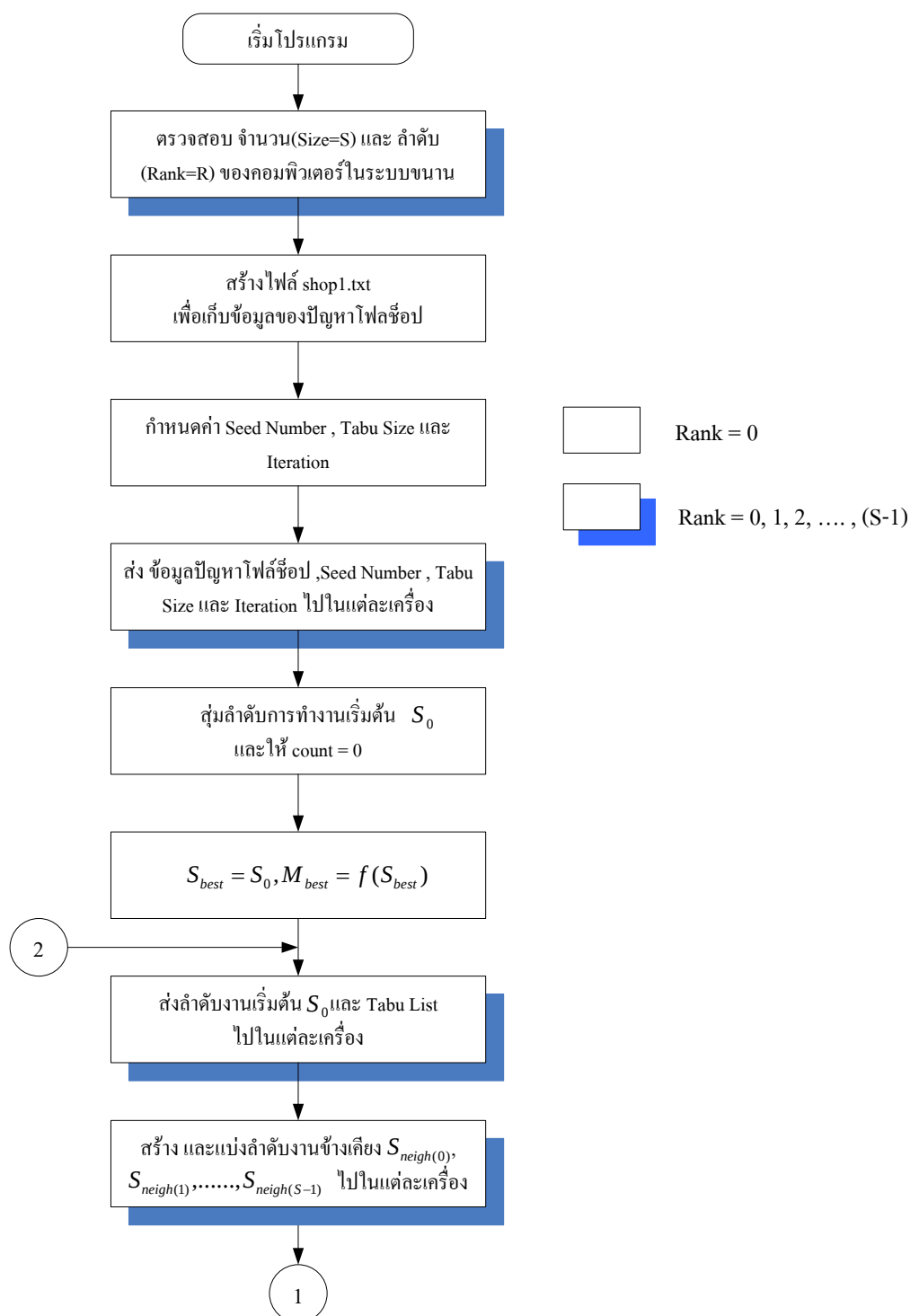
1. เมื่อขนาดของปัญหาใหญ่ขึ้น ทำให้เวลาที่ใช้ในการประมวลผลเพิ่มขึ้น
2. เมื่อต้องการพัฒนาคุณภาพของคำตอบให้ดีขึ้น ก็ต้องเพิ่มจำนวนรอบการทำงานให้มากขึ้น ทำให้เวลาที่ใช้ในการประมวลผลเพิ่มมากขึ้นตาม

เมื่อได้ข้อสรุปความสัมพันธ์ทั้งหมดที่ได้จากการวิเคราะห์ข้อมูลเวลาในการทำงานแล้ว ขั้นตอนต่อไป คือ การออกแบบโปรแกรมการจัดตารางการผลิตแบบโพลีซ็อป โดยใช้วิธีการหาคำตอบแบบทามู ให้สามารถทำงานในระบบขนานเพื่อเพิ่มความเร็วในการประมวลผลกับปัญหาการจัดตารางการผลิตขนาดใหญ่ เมื่อเทียบกับการประมวลผลโดยใช้หน่วยประมวลผลเพียง 1 หน่วย

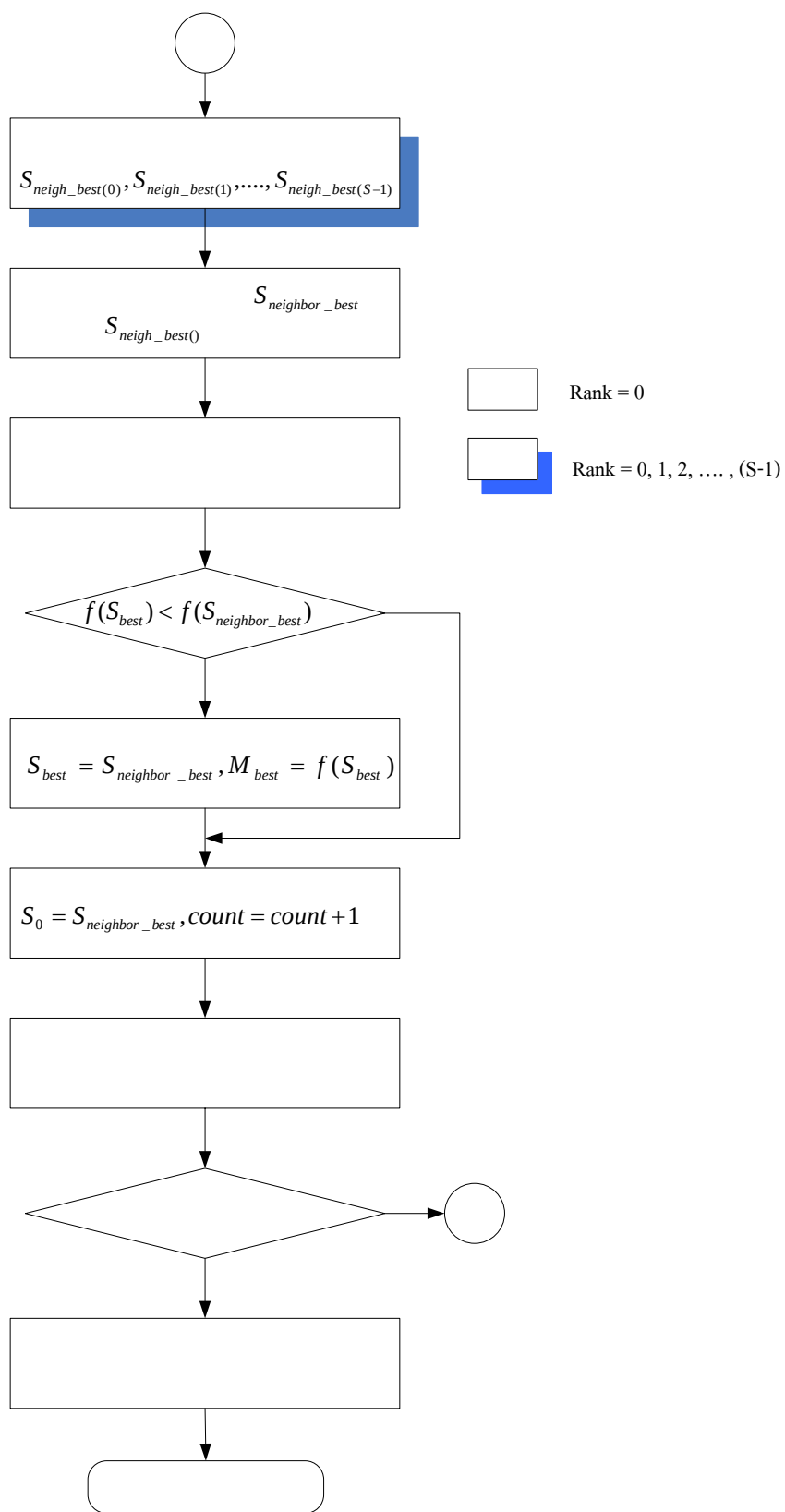
การออกแบบโปรแกรมการจัดตารางการผลิตแบบขนาน นั้นจะอาศัยหลักการ ของ MPI (Message Passing Interface) มาใช้ในการออกแบบโปรแกรม จากการศึกษาพบว่าหลักการของโปรแกรมขนาน คือการแบ่งงานออกเป็นส่วนย่อยๆ และถูกดำเนินการประมวลผลไปพร้อมๆกัน และแต่ละหน่วยประมวลผลจะติดต่อสื่อสารกันเพื่อที่จะทำงานร่วมกัน ทำให้การแก้ปัญหาานั้นเป็นไปได้อย่างรวดเร็วและมีประสิทธิภาพมากกว่าโปรแกรมการจัดตารางการผลิตแบบเดิม

เมื่อพิจารณาขั้นตอนการทำงานของโปรแกรมการจัดตารางการผลิตแบบเดิม และผลที่ได้จากการทดสอบ ตารางที่ 12 และตารางที่ 13 พบว่าขั้นตอนของโปรแกรมที่เหมาะสมจะนำมาแบ่งเป็นส่วนย่อยๆ ของโปรแกรมแบบขนานก็คือในส่วนของเซตของลำดับงานข้างเคียง ซึ่งเป็นส่วนที่ใช้เวลามากที่สุดในโปรแกรม แล้วถ้าขนาดของงานมากขึ้นส่วนของเซตลำดับงานข้างเคียงก็จะมากขึ้นด้วยส่งผลให้เวลาในการประมวลผลสูงเมื่อนำไปใช้กับงานที่มีขนาดใหญ่ จึงเป็นส่วนที่เหมาะสมจะนำมาแบ่งย่อยเป็นการทำงานแบบขนานเพื่อลดเวลาการประมวลผลของโปรแกรม

โดยเมื่อนำหลักการของ MPI มาประยุกต์ใช้ในการออกแบบโปรแกรมการจัดตารางการผลิตแบบขนานนั้น จะมีขั้นตอนของโปรแกรมดังแสดงในภาพที่ 6 และภาพที่ 7



ภาพที่ 6 ขั้นตอนการทำงานของโปรแกรมจัดตารางการผลิตแบบขนาน



ภาพที่ 7 ขั้นตอนการทำงานของโปรแกรมจัดการการผลิตแบบขนาน

ขั้นตอนการทำงานของโปรแกรมจัดตารางการผลิตแบบขนาน ในการทำงานนั้นส่วนหนึ่งก็จะใช้คอมพิวเตอร์ในระบบขนานทุกตัวช่วยกันประมวลผล เพื่อจะช่วยลดเวลาการในการทำงาน อีกส่วนก็ใช้คอมพิวเตอร์เครื่องหลักในการประมวลผล โดยจะทำหน้าที่ส่งค่าเริ่มต้นต่างๆ จากที่ผู้กำหนด ไปยังคอมพิวเตอร์ทุกตัวในระบบ โดยจะอธิบายรายละเอียดจาก ภาพที่ 6 และภาพที่ 7 เป็นขั้นตอนดังต่อไปนี้

1. ตรวจสอบจำนวน(Size=S) ของคอมพิวเตอร์ เพื่อให้ทราบจำนวนคอมพิวเตอร์ที่ต่ออยู่ในระบบแบบขนาน และตรวจสอบลำดับ (Rank=R) ของคอมพิวเตอร์ เพื่อใช้ระบุถึงลำดับของคอมพิวเตอร์แต่ละตัวที่อยู่ในระบบขนาน โดยลำดับของคอมพิวเตอร์ในระบบขนานจะเริ่มตั้งแต่ $R = 0$ ไปจนถึงลำดับสุดท้ายคือ $R = S-1$ โดยเราจะแบ่งเป็น คอมพิวเตอร์หลัก($R=0$) และคอมพิวเตอร์ทุกตัวในระบบขนาน ($R = 0, 1, 2, \dots, S-1$)
2. สร้างไฟล์ shop1.txt โดยในไฟล์จะมีข้อมูลของปัญหาโฟลชีป ประกอบไปด้วยจำนวนงาน N งาน ที่ถูกผลิตบนเครื่องจักร M เครื่อง และเวลาของแต่ละงาน บนแต่ละเครื่องจักร โดยข้อมูลนี้จะถูกโหลดเก็บไว้ในคอมพิวเตอร์หลัก($R=0$)
3. กำหนดค่าต่างๆที่เป็นเงื่อนไขในการทำงานของโปรแกรม คือ ค่าเริ่มต้น (Seed Number) ขนาดของทาบู (Tabu Size) และจำนวนรอบที่ใช้ในการหาคำตอบ (Iteration) โดยจะเก็บไว้ในคอมพิวเตอร์หลัก($R=0$)
4. ส่งข้อมูลต่างๆของปัญหาโฟลชีป ค่าเริ่มต้น (Seed Number) ขนาดของทาบู (Tabu Size) และจำนวนรอบที่ใช้ในการหาคำตอบ (Iteration) ซึ่งเป็นข้อมูลพื้นฐานที่นำไปใช้ประมวลผลไปยังคอมพิวเตอร์ทุกตัวในระบบขนาน ($R = 0, 1, 2, \dots, S-1$)
5. ทำการสร้างลำดับการทำงานเริ่มต้น S_0 โดยการสุ่ม แล้วเริ่มในการค้นหาคำตอบของโปรแกรม (count = 0) โดยใช้คอมพิวเตอร์หลัก($R=0$)
6. ให้ลำดับการทำงานเริ่มต้น S_0 เป็นลำดับการทำงานที่ดีที่สุด S_{best} แล้วคำนวณคำตอบของลำดับงานที่ดีที่สุดด้วยฟังก์ชันการหาเวลาในการทำงานรวม (Makespan) แล้วกำหนดให้เป็น เวลาในการทำงานรวมน้อยที่สุด (M_{best}) โดยจะเก็บค่าไว้ในคอมพิวเตอร์หลัก($R=0$)

7. ส่งข้อมูลลำดับการทำงานเริ่มต้น S_0 และลำดับงานที่มีสถานะเป็นทาบู (Tabu List) ไปยังคอมพิวเตอร์ทุกตัวในระบบขนาน ($R = 0, 1, 2, \dots, S-1$)

8. ทำการแบ่งเซตคำตอบของลำดับงานข้างเคียงที่มีทั้งหมดออกเป็นส่วนย่อยๆ ไปยังคอมพิวเตอร์ทุกตัวในระบบขนาน ($S_{neigh(0)}, S_{neigh(1)}, \dots, S_{neigh(S-1)}$)

9. ทำการพิจารณาเซตของลำดับงานข้างเคียงของคอมพิวเตอร์แต่ละเครื่อง ที่ไม่เป็นทาบู ที่ให้คำตอบที่ดีที่สุดของคอมพิวเตอร์ทุกตัวในระบบขนาน ($S_{neigh_best(0)}, S_{neigh_best(1)}, \dots, S_{neigh_best(S-1)}$)

10. พิจารณาลำดับงานข้างเคียงที่ดีที่สุดของคอมพิวเตอร์ทุกตัวในระบบขนาน เพื่อหาลำดับงานข้างเคียงที่ดีที่สุดของระบบ ($S_{neighbor_best}$) เก็บไว้ในคอมพิวเตอร์หลัก ($R=0$)

11. ทำการปรับปรุงรายการทาบู (Tabu List) โดยทำการเพิ่มจำนวนครั้งการเป็นทาบู ของลำดับงานที่เป็นทาบูอยู่แล้ว และกำหนดให้สถานะลำดับงานข้างเคียงที่ดีที่สุดของระบบเป็นทาบู

12. เปรียบเทียบระหว่างเวลาในการทำงานรวมของลำดับงานข้างเคียงที่ดีที่สุดของระบบ $f(S_{neighbor_best})$ กับเวลาในการทำงานรวมที่ดีที่สุด (M_{best}) ถ้าเวลารวมจากคำตอบข้างเคียงที่ได้ แย่กว่าเวลาในการทำงานรวมที่ดีที่สุด ให้ข้ามไปข้อ 14

13. ถ้าเวลารวมจากคำตอบข้างเคียงที่ได้ดีกว่าเวลาในการทำงานรวมที่ดีที่สุดให้แทนลำดับงานที่ดีที่สุด (S_{best}) ด้วยลำดับงานข้างเคียงที่ดีที่สุด ($S_{neighbor_best}$) แล้วคำนวณหาเวลาการทำงานรวม (M_{best})

14. กำหนดให้ลำดับงานข้างเคียงที่ดีที่สุด ($S_{neighbor_best}$) เป็นลำดับงานเริ่มต้น (S_0) เพื่อใช้เป็นคำตอบเริ่มต้นในการหาคำตอบรอบต่อไป แล้วทำการเพิ่มค่า count ซึ่งหมายถึงจบกระบวนการ 1 รอบการทำงาน

15. ตรวจสอบลำดับงานที่มีสถานะเป็นทาบู ว่าครบกำหนดการเป็นทาบูตามขนาดของ Tabu Size ถ้าครบแล้วให้ยกเลิกสถานะของลำดับงานนี้ไม่ให้เป็นทาบู

17. ตรวจสอบว่าครบตามจำนวนรอบที่กำหนด(Iteration) ไว้หรือยัง ถ้ายังไม่ครบตามที่ตั้งไว้ให้ย้อนกลับไปทำที่ข้อ 7

18. ถ้าครบตามจำนวนรอบที่กำหนดไว้แล้วให้สร้างไฟล์ “outsolution.txt” เพื่อเก็บผลของคำตอบที่ได้จากการประมวลผล

5. ทดสอบและประเมินผลโปรแกรมจัดตารางการผลิต โดยใช้ระบบคอมพิวเตอร์แบบคู่ขนาน

ในขั้นตอนต่อไปเป็นทดสอบและประเมินผลการทำงานของโปรแกรมจัดตารางการผลิตที่ใช้วิธีค้นหาทาบแบบขนาน และพิจารณาผลการทดสอบในส่วนของผลและวิจารณ์ผลการทดลองการคำนวณหาจำนวนเท่าความเร็วและประสิทธิภาพของโปรแกรม ที่ปรับให้ทำงานแบบขนานดังแสดงในสมการที่ 3 - 6

$$T_{seq} = T_1 \quad (3)$$

$$\text{Testing } n \text{ Processor} \Rightarrow \{T_1, T_2, \dots, T_n\} \quad (4)$$

$$\text{Speedup} = T_{seq} / \{T_1, T_2, \dots, T_n\} \quad (5)$$

$$E = \text{Speedup} / n \quad (6)$$

โดยที่

T_{seq} คือ เวลาในการทำงานของโปรแกรมโดยใช้หนึ่งหน่วยประมวลผล

Testing n Processor คือ เวลาในการทำงานของโปรแกรมโดยใช้ n หน่วยประมวลผล

Speedup คือ จำนวนเท่าความเร็วของโปรแกรมเมื่อทำงานแบบขนาน

E คือ ประสิทธิภาพของโปรแกรมเมื่อทำงานแบบขนาน โดยใช้ n หน่วยประมวลผล

ผลและวิจารณ์

จากหลักการของโปรแกรมการจัดตารางการผลิตแบบโฟลว์ช็อป โดยใช้วิธีค้นหาทาบแบบขนาน ได้นำมาใช้ในการจัดลำดับงานโดยใช้โปรแกรมภาษา Visual C++ ที่พัฒนาขึ้นเพื่อนำมาทดสอบและค้นหาคำตอบโดยปัญหาที่นำมาทดสอบใช้ตัวอย่างปัญหาของโฟลว์ช็อป ของ Taillard (1993) ซึ่งประกอบไปด้วยปัญหาโฟลว์ช็อปขนาดต่างๆ ดังนี้

1. 50×5
2. 50×10
3. 50×20
4. 100×5
5. 100×10
6. 100×20
7. 200×10
8. 200×20

โดยในการทดสอบโปรแกรมการจัดตารางการผลิตแบบโฟลว์ช็อป ในการหาคำตอบด้วยวิธีค้นหาทาบแบบขนาน นั้นจะเปรียบเทียบเวลาในการทำงานของโปรแกรมกับปัญหาโฟลว์ช็อปแต่ละขนาด โดยใช้หน่วยประมวลผล 1 หน่วยประมวลผล เทียบกับแบบขนาน 2, 3 และ 4 หน่วยประมวลผล และวิเคราะห์ผลด้วยวิธีทางสถิติ โดยใช้การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่ม ด้วยการทดสอบแบบ T ที่ระดับนัยสำคัญ 0.05 โดยมีสมมติฐานดังนี้

$$H_0 : \mu_n = \mu_{n+1}$$

$$H_1 : \mu_n > \mu_{n+1}$$

โดย $n = 1, 2, 3$

μ_n หมายถึง ค่าเฉลี่ยของเวลาที่ใช้ในการประมวลผล เมื่อใช้คอมพิวเตอร์ในการประมวลผล n หน่วยประมวลผล

μ_{n+1} หมายถึง ค่าเฉลี่ยของเวลาที่ใช้ในการประมวลผล เมื่อใช้คอมพิวเตอร์ในการประมวลผลเพิ่มขึ้น 1 หน่วยประมวลผล

จำนวนงาน (Job)	=	50	งาน
จำนวนเครื่องจักร (Machine)	=	5	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 14 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 50×5

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
1	2752	370.66	242.35	224.63	217.934
2	2854	380.51	243.65	221.65	214.36
3	2652	375.11	239.66	226.716	218.35
4	2776	364.554	245.31	227.36	214.325
5	2886	369.51	238.222	228.35	216.63

ตารางที่ 15 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 50×5

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	43.54	0.000
2P และ 3P	8	9.16	0.000
3P และ 4P	8	6.43	0.000

จำนวนงาน (Job)	=	50	งาน
จำนวนเครื่องจักร (Machine)	=	10	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 16 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 50×10

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
6	3108	745.326	430.25	340.25	321.783
7	2927	750.26	425.66	341.23	325.498
8	2923	741.23	439.62	349.232	318.25
9	3071	738.672	430.21	339.25	320.21
10	3095	739.65	431.35	350.21	319.33

ตารางที่ 17 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 50×10

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	99.93	0.000
2P และ 3P	8	26.75	0.000
3P และ 4P	8	8.64	0.000

จำนวนงาน (Job)	=	50	งาน
จำนวนเครื่องจักร (Machine)	=	20	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 18 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 50×20

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
11	3899	1502.36	800.14	611.25	541.128
12	3735	1506.35	809.32	606.31	552.313
13	3696	1498.53	798.26	605.18	539.78
14	3760	1496.99	802.35	600.94	539.621
15	3661	1499.65	804.908	603.11	540.12

ตารางที่ 19 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 50×20

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	274.88	0.000
2P และ 3P	8	76.11	0.000
3P และ 4P	8	20.94	0.000

จำนวนงาน (Job)	=	100	งาน
จำนวนเครื่องจักร (Machine)	=	5	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	15	

ตารางที่ 20 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 100×5

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
16	5227	2910.63	1502.36	1129.63	1003.61
17	5268	2907.61	1509.63	1128.61	1057.33
18	5179	2915.09	1506.32	1124.28	998.63
19	5035	2918.01	1510.609	1120.36	1009.07
20	5255	2914.656	1507.12	1121.78	999.013

ตารางที่ 21 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 100×5

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	603.89	0.000
2P และ 3P	8	164.05	0.000
3P และ 4P	8	9.89	0.000

จำนวนงาน (Job)	=	100	งาน
จำนวนเครื่องจักร (Machine)	=	10	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 22 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 100×10

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
21	5825	6001.12	3025.63	2150.36	1638.96
22	5403	6025.11	3081.321	2148.63	1642.5
23	5745	6050.78	3019.62	2157.34	1640.66
24	5830	6031.22	3055.66	2179.36	1638.4
25	5518	6019.65	3044.63	2180.66	1637.08

ตารางที่ 23 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 100×10

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	217.58	0.000
2P และ 3P	8	67.37	0.000
3P และ 4P	8	74.25	0.000

จำนวนงาน (Job)	=	100	งาน
จำนวนเครื่องจักร (Machine)	=	20	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 24 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 100×20

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
26	6352	13800.6	7231.54	4733.62	3615.36
27	6343	13790	7230.62	4780.63	3471.08
28	6363	13706.4	7289.41	4747.36	3500.15
29	6422	13722	7291.33	4769.66	3491.51
30	6442	13769	7271.94	4721.2	3487.16

ตารางที่ 25 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่มโดยขนาดของปัญหาเท่ากับ 100×20

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	282.97	0.000
2P และ 3P	8	144.5	0.000
3P และ 4P	8	43.81	0.000

จำนวนงาน (Job)	=	200	งาน
จำนวนเครื่องจักร (Machine)	=	10	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 26 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 200×10

ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
31	10983	56521	28142	18620	13200.2
32	10542	56814	27129.4	18635	13722.8
33	11120	56410.7	27432.1	18664.9	13701.3
34	10844	56440.4	27991.4	18599.4	13753
35	10577	56394.3	27565.9	18609.4	13443.3

ตารางที่ 27 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่ม โดยขนาดของปัญหาเท่ากับ 200×10

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	143.91	0.000
2P และ 3P	8	48.71	0.000
3P และ 4P	8	47.27	0.000

จำนวนงาน (Job)	=	200	งาน
จำนวนเครื่องจักร (Machine)	=	20	เครื่องจักร
จำนวนครั้งในการค้นหา (Iteration)	=	20,000	ครั้ง
ค่าเริ่มต้น (Seed)	=	1111	
ขนาดของทาบู (Tabu Size)	=	10	

ตารางที่ 28 ค่า Makespan และเวลาที่ใช้ในการประมวลผลแบบขนาน โดยขนาดของปัญหาเท่ากับ 200×20

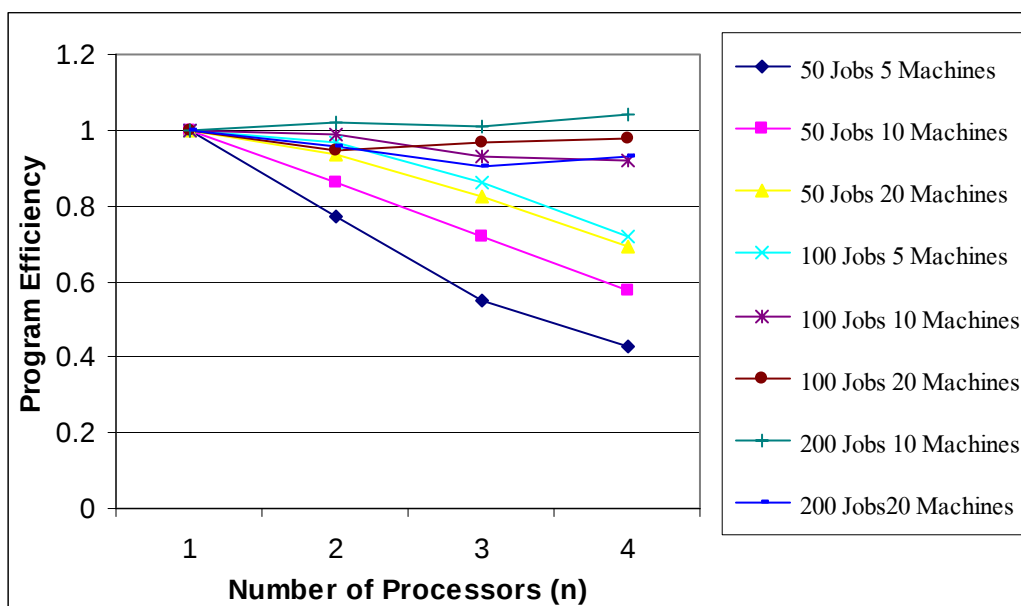
ตัวอย่างที่	Makespan	เวลาประมวลผล (วินาที)			
		1 Process	2 Process	3 Process	4 Process
36	11438	109500	55932	40028.3	28952.5
37	11616	105635	55412	38831.3	28706
38	11658	105721	55531.6	39495.1	28135
39	11481	105528	55697.8	39338.6	28609.3
40	11512	105423.6	55873.2	38400.9	28679.1

ตารางที่ 29 การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่ม โดยขนาดของปัญหาเท่ากับ 200×20

จำนวนหน่วย ประมวลผล	องศาเสรี (d.f.)	ค่า T - Value	ค่า P - Value
1P และ 2P	8	63.95	0.000
2P และ 3P	8	86.18	0.000
3P และ 4P	8	69.7	0.000

ตารางที่ 30 จำนวนเท่าของความเร็วจึงประสิทธิภาพของโปรแกรมจัดการการผลิตโดยวิธี
ค้นหาแบบขนานกับปัญหาโฟลชี้อปขนาดต่างๆกัน

Problem	2 Process		3 Process		4 Process	
	Speedup	E	Speedup	E	Speedup	E
50 x 5	1.539	0.769	1.649	0.550	1.720	0.430
50 x 10	1.723	0.861	2.160	0.720	2.315	0.579
50 x 20	1.869	0.935	2.479	0.826	2.766	0.692
100 x 5	1.933	0.966	2.590	0.863	2.876	0.719
100 x 10	1.979	0.989	2.786	0.929	3.675	0.919
100 x 20	1.894	0.947	2.896	0.965	3.917	0.979
200 x 10	2.044	1.022	3.034	1.011	4.168	1.042
200 x 20	1.910	0.955	2.712	0.904	3.717	0.929



ภาพที่ 16 ประสิทธิภาพในการประมวลผลแบบขนานของโปรแกรม กับปัญหาโฟลชี้อปขนาดต่างๆ

จากผลการทดสอบโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยใช้วิธีการค้นหาหาที่ทำงานแบบขนานซึ่งทดสอบกับปัญหามิติต่างกัน ด้วยวิธีทางสถิติ เมื่อพิจารณาตารางที่ 15 17 19 21 23 25 27 และ 29 เมื่อพิจารณาค่า P – Value พบว่าจากการทดสอบทั้งหมดมีค่าน้อยกว่าระดับนัยสำคัญ 0.05 แสดงว่าปฏิเสธสมมติฐานหลัก ดังนั้นจึงสรุปว่าการเพิ่มจำนวนหน่วยประมวลผลในการทำงานสามารถช่วยลดเวลาในการประมวลผลได้

และเมื่อพิจารณาประสิทธิภาพของโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยใช้วิธีการค้นหาหาที่ทำงานแบบขนานกับขนาดของปัญหาที่ต่างกันสามารถสรุปได้ดังนี้

1. ในการทดสอบกับปัญหามิติขนาด 50x5 ได้ผลดังตารางที่ 14 และประสิทธิภาพของโปรแกรมดังตารางที่ 30 และภาพที่ 16 พบว่าเมื่อเพิ่มหน่วยประมวลผล 2 หน่วยประมวลผลนั้นสามารถลดเวลาในการทำงานลง และประสิทธิภาพในการทำงานเท่ากับ 0.769 ซึ่งเหมาะสมในการทำงานแบบขนาน และเมื่อเพิ่มหน่วยประมวลผล 3 และ 4 หน่วยประมวลผลนั้น สามารถลดเวลาในการประมวลผลลง แต่ประสิทธิภาพในการทำงานเท่ากับ 0.55 และ 0.43 ทำให้ไม่เหมาะสมในการทำงานแบบขนาน เนื่องจากประสิทธิภาพในการทำงานแบบขนานไม่ควรต่ำกว่า 0.6

2. ในการทดสอบกับปัญหามิติขนาด 50x10 ได้ผลดังตารางที่ 16 และประสิทธิภาพของโปรแกรมดังตารางที่ 30 และภาพที่ 16 พบว่าเมื่อเพิ่มหน่วยประมวลผล 2 และ 3 หน่วยประมวลผลนั้นสามารถลดเวลาในการทำงานลง และประสิทธิภาพในการทำงานเท่ากับ 0.861 และ 0.72 ซึ่งเหมาะสมในการทำงานแบบขนาน และเมื่อเพิ่มหน่วยประมวลผล 4 หน่วยประมวลผลนั้น สามารถลดเวลาในการประมวลผลลง แต่ประสิทธิภาพในการทำงานเท่ากับ 0.579 ทำให้ไม่เหมาะสมในการทำงานแบบขนาน เนื่องจากประสิทธิภาพในการทำงานแบบขนานไม่ควรต่ำกว่า 0.6

3. ในการทดสอบกับปัญหามิติขนาด 50x20 100x5 100x10 100x20 200x10 และ 200x20 ได้ผลดังตารางที่ 18 20 22 24 26 28 ประสิทธิภาพของโปรแกรมดังตารางที่ 30 และภาพที่ 16 พบว่าเมื่อเพิ่มหน่วยประมวลผล 2 3 และ 4 หน่วยประมวลผลนั้นสามารถลดเวลาในการทำงานลง และประสิทธิภาพในการทำงานสูงกว่า 0.6 ซึ่งเหมาะสมในการทำงานแบบขนาน โดยเฉพาะปัญหาที่มีขนาดของปัญหาค้างตั้งแต่ 100×10 ขึ้นไปจะเห็นว่าประสิทธิภาพของโปรแกรมมีค่าสูงกว่า 0.9 ซึ่งถือว่าเหมาะสมที่สุดในการทำงานแบบขนาน

ในส่วนคุณภาพของคำตอบที่ได้จากโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีทาบูกานูแบบขนานนั้น มาใช้ในการเปรียบเทียบกับคำตอบที่ได้จากวิธีการจำลองแอนนีลิ่งของ Ogbu and Smith (1990) และวิธีค้นหาทาบูกานู ที่ได้รับการปรับปรุงโดย Ben and Fawzan (1998) กับค่า Best known ซึ่งมีค่าเท่ากับค่า LB ของ Taillard (1993) ดังตารางที่ 31

ตารางที่ 31 แสดงค่า Makespan ($C_{\max}$) และค่า PRD ของโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีทาบูกานู ที่ทำงานแบบขนาน เปรียบเทียบกับ วิธี OS-SA และวิธี BF-Tabu

ตัวอย่าง	Best known	Parallel Tabu		OR-SA		BF-Tabu	
	$C_{\max}$	$C_{\max}$	PRD (%)	$C_{\max}$	PRD (%)	$C_{\max}$	PRD (%)
50×5	2712	2745	1.217	2791	2.913	2724	0.442
50×10	2907	3025	4.059	3399	16.925	3037	4.472
50×20	3480	3899	12.040	4438	27.529	3886	11.667
100×5	5437	5527	1.655	5674	4.359	5493	1.030
100×10	5729	5825	1.676	6367	11.136	5776	0.820
100×20	5851	6352	8.563	6721	14.869	6345	8.443
200×10	10816	10983	1.544	11924	10.244	10894	0.721
200×20	10979	11348	3.361	12798	16.568	11398	3.816
Minimum			1.127		2.913		0.422
Maximum			12.04		27.529		11.667
Average			4.264		13.068		3.927

ผลจากตารางที่ 31 เป็นคำตอบของโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีค้นหาทาบูกานูแบบขนาน (Parallel Tabu) กับคำตอบของวิธี OR-SA และ วิธี BF-Tabu และพิจารณาค่าเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ระหว่างคำตอบที่ได้ของทั้ง 3 วิธี กับค่า Best known ของ Taillard (1993) โดยที่ $PRD = \frac{100(C_{\max} - LB)}{LB}$

เมื่อวิเคราะห์คุณภาพของคำตอบที่ได้จากวิธีค้นหาทาบูกานูแบบขนาน เปรียบเทียบกับวิธีการจำลองแอนนีลิ่งของ Ogbu and Smith (1990) ด้วยวิธีทางสถิติโดยใช้การทดสอบนัยสำคัญของ

ความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่ม ด้วยการทดสอบแบบ T ที่ระดับนัยสำคัญ 0.05 โดยมีสมมุติฐานดังนี้

$$H_0 : \mu_{PT} = \mu_{SA}$$

$$H_1 : \mu_{PT} < \mu_{SA}$$

โดย μ_{PT} หมายถึง ค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีค้นหาทาบแบบขนาน

μ_{SA} หมายถึง ค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีการจำลองแอนนีลิงของ Ogbu and Smith (1990)

จากการทดสอบด้วยโปรแกรมทางคณิตศาสตร์ได้ผลดังนี้คือ ค่า P – Value เท่ากับ 0.007 และเมื่อพิจารณาค่า P – Value พบว่ามีค่าน้อยกว่าระดับนัยสำคัญที่ 0.05 แสดงว่าปฏิเสธสมมุติฐานหลัก ดังนั้นจึงสรุปได้ว่าค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีค้นหาทาบแบบขนานนั้นน้อยกว่าค่าเฉลี่ยของคำตอบที่ได้จากวิธีการจำลองแอนนีลิง ซึ่งหมายถึงคุณภาพของคำตอบที่ได้จากวิธีค้นหาทาบแบบขนานดีกว่าวิธีการจำลองแอนนีลิง

เมื่อวิเคราะห์คำตอบที่ได้จากวิธีค้นหาทาบแบบขนาน เปรียบเทียบกับวิธีค้นหาทาบของ Ben and Fawzan (1998) ด้วยวิธีทางสถิติโดยใช้การทดสอบนัยสำคัญของความแตกต่างระหว่างค่าเฉลี่ยของสองกลุ่ม ด้วยการทดสอบแบบ T ที่ระดับนัยสำคัญ 0.05 โดยมีสมมุติฐานดังนี้

$$H_0 : \mu_{PT} = \mu_T$$

$$H_1 : \mu_{PT} < \mu_T$$

โดย μ_{PT} หมายถึง ค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีค้นหาทาบแบบขนาน

μ_T หมายถึง ค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีค้นหาทาบของ Ben and Fawzan (1990)

จากการทดสอบด้วยโปรแกรมทางด้านสถิติได้ผลดังนี้คือ ค่า P – Value เท่ากับ 0.565 และเมื่อพิจารณาค่า P – Value พบว่ามีมากกว่าระดับนัยสำคัญที่ 0.05 แสดงว่ายอมรับสมมติฐานหลัก ดังนั้นจึงสรุปได้ว่าค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) ที่ได้จากวิธีค้นหาทาบแบบขนาน กับวิธีค้นหาทาบของ Ben and Fawzan (1998) มีค่าไม่แตกต่างกัน ซึ่งหมายถึงคุณภาพของคำตอบที่ได้จากวิธีค้นหาทาบแบบขนานไม่แตกต่างกับคุณภาพของคำตอบที่ได้จากวิธีค้นหาทาบของ Ben and Fawzan (1998)

สรุป

งานวิจัยเรื่องนี้เป็นปรับปรุงวิธีการค้นหาทฤษฎี โดยใช้หลักการของคอมพิวเตอร์แบบคู่ขนานในการจัดการการผลิตแบบต่อเนื่อง เมื่อพิจารณาผลการทดสอบในส่วนของเวลาที่ใช้ในการหาคำตอบของวิธีค้นหาทฤษฎีแบบขนาน การทดสอบทางสถิติ และประสิทธิภาพของการค้นหาทฤษฎีแบบขนาน สามารถสรุปได้ดังนี้

1. การค้นหาทฤษฎีแบบขนานสามารถช่วยลดเวลาในการประมวลผลลงได้
2. การค้นหาทฤษฎีแบบขนานเหมาะสมที่จะนำมาใช้กับปัญหาที่มีขนาดใหญ่

ในส่วนคุณภาพของคำตอบที่ได้จากวิธีค้นหาทฤษฎีแบบขนาน เมื่อพิจารณาค่าเปอร์เซ็นต์ความแตกต่าง (Percentage Relative Difference) มีค่าต่ำกว่า 5% ซึ่งถือว่าเป็นคำตอบที่ดีมีค่าใกล้เคียงกับค่า best-known ของคำตอบ และเมื่อนำไปทดสอบกับเมตา-ฮิวริสติก (Meta – Heuristics) วิธีอื่นที่ใช้ปัญหาเดียวกันในการทดสอบ โดยใช้วิธีทางสถิติพบว่าคุณภาพของคำตอบที่ได้จากวิธีค้นหาทฤษฎีแบบขนานดีกว่าคำตอบที่ได้จากวิธีการจำลองแอนนีลิงของ Ogbu and Smith (1990) และเมื่อเปรียบเทียบกับวิธีค้นหาทฤษฎีที่ปรับปรุงโดย Ben and Fawzan (1998) พบว่าคุณภาพของคำตอบที่ได้ไม่แตกต่างกัน

และจากผลการวิจัยพบว่าการนำเอาระบบคอมพิวเตอร์แบบคู่ขนานสามารถค้นหาคำตอบที่ดีกว่าได้ เพราะว่าประสิทธิภาพของการค้นหาคำตอบจะขึ้นอยู่กับปัจจัยจำนวนรอบในการค้นหาด้วยเช่นกัน ซึ่งหมายความว่าเราสามารถนำเวลาที่ลดลงจากการใช้หน่วยประมวลผลแบบคู่ขนานไปเพิ่มจำนวนรอบในการค้นหา และทำให้มีโอกาสมากขึ้นที่จะพบคำตอบที่ดีกว่าเดิม

ข้อเสนอแนะ

ในงานวิจัยครั้งนี้ได้นำระบบคอมพิวเตอร์แบบกลุ่มงานมาปรับปรุงวิธีการค้นหาทามู ซึ่งยังมีเมตา-ฮิวริสติกวิธีอื่นที่สามารถนำเอาหลักการของระบบคอมพิวเตอร์แบบกลุ่มงานไปประยุกต์ใช้ เช่น วิธีการค้นหาทามู ของ Ben and Fawzan (1998) ที่ปรับปรุงวิธีการค้นหาทามูแบบเดิมเพื่อให้คุณภาพของคำตอบสูงขึ้น ถ้าศึกษาขั้นตอนการทำงานและนำเอาระบบคอมพิวเตอร์แบบกลุ่มงานไปประยุกต์ใช้ก็จะสามารถช่วยลดเวลาในการหาคำตอบได้ และนำเวลาที่ลดลงไปเพิ่มจำนวนรอบในการค้นหาเพื่อเพิ่มโอกาสในการหาคำตอบที่ดีกว่าเดิมได้

เอกสารและสิ่งอ้างอิง

จันทร์เพ็ญ อักกะวินศ 2539. เทคนิคฮิวริสติกเชิงรวมสำหรับการจัดตารางงานแบบโฟลว์ช็อป
วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์

ศุภชัย ปทุมนากุล และ วนิดา รัตนมณี 2546. การหาผลเฉลยโดยวิธี Tabu search ว. สงขลา
นกรินทร์ วทท. 25 (6) : 799-805

สมศักดิ์ สันทะ 2547. การศึกษาเปรียบเทียบวิธีค้นหาแบบตามูและ วิธีเชิงพันธุกรรมสำหรับ
ปัญหาพื้นผิวดอสนอง วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยธรรมศาสตร์

หทัยชนก แสงศิลป์มณี 2546. แนวทางการปรับ BLASTN ให้ทำงานแบบขนาน วิทยานิพนธ์
ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์

Al-Turki, U., Fedjki, C and Andijani, A. 2001 Tabu search for a class of single-machine
scheduling problems. **Compyter & Operations Research** 28: 1223-1230.

Ben-Daya and Al-Fawzan. 1998. A tabu search approach for the flow shop scheduling problem.

European Journal of Operations Research 109: 88-95.

Bland , J.A. and G.P.Dawson. 1991. Tabu search and design optimization. **Computer-aided**

Design 23 (3): 195-201

Cheng, T.C. and G. Wang. 1999. Scheduling the fabrication and assembly of components

in a two-machine flowshop. **IIE Transactions** 31 (2): 135-143.

Elmaghraby, S.E. and K.A. Thoney. 1999. The two-machine stochastic flowshop problem

with arbitrary processing time distributions. **IIE Transactions** 31(5): 467-477.

Glover, F. 1990. Tabu search: A tutorial. **Interface** 20 (4): 74-79.

Glover, F. 1993. Tabu search – Part II **ORSA Journal on Computer** 2 (1): 4-32

Jain, A.K. and H.A. Elmaraghy. 1997. Single process plan scheduling with genetic algorithms.

Production Planning & Control 8 (4): 363-376.

Karniadakis, G. 2003. **Parallel scientific computing in C++ and MPI**. Cambridge University

Press, New York.

Khamsawang, S., Boonseng, C. and Pothiya, S. 2002. Solving the economic dispatch problem

with tabu search algorithm. **2002 IEEE International Conference** 1: 274-278.

Kouvelis, P., R.L. Daniels and G. Vairaktarakis. 2000. Robust scheduling of a two- machine

flow shop with uncertain processing times. **IIE Transactions** 32 (5): 421-432.

Kumar, V., A. Grama, A. Gupta and G. Karypis. 1994. **Introduction to parallel computing:**

Design and Analysis of Algorithms. The Benjamin/Cummings Publishing Company, Inc., California.

- Mcmullen, P.R. and G.V. Frazier. 2000. A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. **IIIE Transactions** 32 (8): 679-686.
- Ogbu F.A. and Smith D.K. 1990. The application of the simulated annealing algorithm to the solution of the $n/m/C_{\max}$ flow shop problem. **Computers and Operation Research** 17: 243-253.
- South M.C., Wetherill B., and Tham M.T. 1993. Hitch-hikers guide to genetic algorithms. **Journal of Applied Statistics** 20: 153-175.
- Taillard, E. 1993. Benchmarks for basic scheduling problems. **European Journal of Operation Research** 64: 278-285.
- Yamada T. and C. Reeves. 1997. Permutation flowshop scheduling by genetic local search. **Proc. of the 2nd IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems (GALESIA '97)**: 232-238.

แสดงตัวอย่างการจัดลำดับงานกับปัญหามิติต่างๆ

สำหรับทุกตัวอย่างใช้ข้อมูลจากปัญหาของโพลชีอปอง Taillard (1993) โดยในแต่ละตัวอย่างจะประกอบไปด้วยข้อมูล 2 ส่วน คือ

1. N และ M โดยการแทนจำนวนงาน N งาน ที่ถูกผลิตบนเครื่องจักร M เครื่อง
2. เวลาของงานแต่ละงานที่ใช้ในแต่ละเครื่องจักร

ปัญหา 50×5

50	5				
1	75 26 48 26 77	23	89 29 23 21 46	45	84 65 77 97 85
2	87 37 4 67 94	24	30 29 1 81 18	46	58 12 5 64 46
3	13 25 92 4 9	25	68 94 55 28 93	47	93 58 14 73 42
4	11 95 92 14 57	26	21 20 91 70 58	48	76 45 47 28 18
5	41 49 72 93 29	27	78 42 67 89 85	49	50 49 80 4 36
6	43 12 45 54 79	28	46 80 91 54 58	50	30 15 45 87 2
7	93 59 5 21 55	29	99 94 4 96 97		
8	69 17 98 20 73	30	10 35 60 62 10		
9	80 46 93 6 65	31	17 8 38 46 79		

10	13 20 17 18 86	32	23 41 25 60 93
11	24 52 79 75 25	33	83 65 90 19 2
12	72 44 11 25 39	34	47 4 93 97 87
13	38 92 16 16 76	35	86 71 13 13 17
14	81 75 89 77 24	36	18 30 65 7 18
15	83 95 81 28 38	37	67 14 25 44 10
16	88 33 92 24 5	38	46 32 34 7 50
17	26 10 45 15 91	39	4 50 47 73 8
18	6 45 61 77 29	40	14 30 98 15 26
19	89 2 39 36 22	41	4 27 91 66 14
20	67 62 28 16 27	42	20 98 11 70 21
21	70 62 94 32 39	43	88 39 46 97 15
22	30 82 87 46 31	44	50 84 50 33 10

ปัญหา 50 × 10

50	10		
1	46 61 3 51 37 79 83 22 27 24	26	73 71 38 12 7 66 48 69 51 23
2	52 87 1 24 16 93 87 29 92 47	27	19 66 25 62 66 11 4 26 2 34
3	79 51 58 21 42 68 38 99 75 39	28	69 94 24 43 54 35 37 24 81 87
4	45 25 85 57 47 75 38 25 94 66	29	12 7 90 49 86 52 82 55 12 59
5	97 73 33 69 94 37 86 98 18 41	30	73 15 7 54 49 8 57 98 40 2
6	10 93 71 51 14 44 67 55 41 46	31	85 11 11 87 3 40 61 86 59 38
7	44 28 58 50 94 34 23 80 37 24	32	23 99 49 29 48 62 6 30 32 84
8	24 90 56 51 34 39 19 82 58 23	33	53 37 2 2 44 25 97 92 16 62
9	85 94 64 21 72 76 97 33 56 68	34	16 50 76 18 93 24 5 94 87 10
10	75 59 43 19 36 62 78 68 20 50	35	88 56 17 75 37 30 27 66 78 11
11	66 64 48 63 88 74 66 47 2 93	36	8 69 32 39 82 1 95 47 41 93
12	49 2 69 91 51 28 67 74 39 22	37	26 22 39 77 31 73 46 3 43 57
13	95 16 96 11 41 78 7 26 91 64	38	42 56 9 69 59 27 92 41 94 81
14	61 35 35 6 71 43 23 61 81 81	39	58 67 83 15 78 16 46 41 1 10

15	19 53 82 31 94 98 67 95 33 94	40	63 63 69 78 33 91 52 47 93 40
16	47 40 53 63 99 83 8 55 14 97	41	7 96 67 68 36 33 8 89 22 62
17	84 81 64 36 11 91 77 11 88 54	42	2 74 28 37 3 11 11 28 93 49
18	13 26 11 39 97 27 71 42 22 82	43	44 4 88 22 58 99 7 39 62 90
19	11 85 61 57 44 6 85 72 36 11	44	38 42 23 41 10 2 54 80 53 34
20	19 4 36 47 77 82 29 14 65 91	45	24 40 91 92 98 60 72 47 30 11
21	98 4 53 56 69 60 49 8 79 23	46	76 30 71 67 6 90 57 57 34 81
22	2 10 87 65 91 44 3 98 23 32	47	85 93 3 24 44 36 85 74 27 51
23	85 63 88 59 38 43 94 90 66 26	48	61 36 26 87 62 62 22 38 30 21
24	44 96 10 4 25 76 76 36 5 22	49	32 25 41 91 24 15 87 59 54 39
25	7 55 32 10 87 99 95 75 15 12	50	90 87 96 31 94 3 65 5 77 27

ปัญหา 50 × 20

50	20
1	52 63 82 16 63 94 79 22 80 96 53 54 71 27 95 3 92 80 61 74
2	95 99 21 26 55 34 21 6 13 3 19 67 90 93 53 80 62 38 86 38
3	42 69 79 46 59 89 93 24 64 50 99 25 59 49 54 78 11 55 16 4
4	75 70 95 66 35 62 32 55 77 57 62 77 82 63 22 32 83 34 42 31
5	44 53 46 76 21 47 8 48 17 66 88 38 22 65 84 53 87 85 14 62
6	57 21 23 31 59 66 45 57 78 84 93 98 88 34 54 43 66 44 92 39
7	89 10 40 36 78 76 37 78 82 98 34 96 35 10 2 85 98 47 67 97
8	53 31 95 8 25 15 78 5 4 55 72 20 49 56 80 19 42 66 77 57
9	84 80 87 37 30 18 26 50 72 70 42 15 78 51 84 48 23 19 46 9
10	62 18 37 21 38 54 98 83 93 32 65 36 69 97 66 49 45 66 41 54
11	91 5 24 3 78 24 17 70 68 31 39 65 76 52 25 66 52 61 78 13
12	14 18 24 76 79 55 25 21 25 64 79 97 2 46 16 22 6 60 3 47
13	95 17 65 67 58 96 21 71 67 11 9 27 14 16 79 37 3 98 72 6
14	89 71 62 5 44 10 28 58 80 9 26 25 3 50 90 51 64 82 95 70

15 4 90 19 47 38 12 68 36 43 32 72 61 22 96 51 82 55 79 53 19
 16 95 93 67 72 76 96 24 50 93 58 29 24 26 85 29 59 97 71 59 97
 17 2 14 66 66 70 53 62 31 21 98 36 97 44 61 29 88 83 28 34 41
 18 97 49 6 56 72 92 89 86 33 95 48 61 1 76 90 77 42 74 66 1
 19 68 52 65 95 85 77 60 29 14 25 57 75 4 30 83 19 81 27 42 57
 20 20 7 59 49 8 6 64 30 30 4 95 92 16 90 83 32 92 33 63 60
 21 33 78 2 47 10 91 38 93 59 45 93 73 55 42 19 52 68 13 27 62
 22 51 57 67 26 84 14 90 49 83 60 79 21 43 88 95 9 46 9 92 14
 23 98 41 82 81 42 41 87 83 85 87 88 29 87 37 87 96 56 12 8 90
 24 8 75 90 56 67 30 1 89 85 31 77 3 35 43 12 23 88 51 65 76
 25 85 98 30 76 20 85 99 44 70 1 94 96 76 88 34 64 50 16 34 12

 26 86 93 63 66 24 17 34 38 35 96 39 51 98 91 23 22 13 49 6 89
 27 73 33 5 36 75 23 9 62 2 22 74 26 78 14 44 37 23 83 42 37
 28 4 75 93 53 23 60 22 45 76 95 46 44 81 63 30 3 13 48 39 35
 29 40 68 53 26 33 76 74 22 46 73 17 56 48 65 82 52 49 13 2 91
 30 98 33 85 52 60 39 14 85 72 77 30 31 25 74 83 44 18 78 7 69
 31 12 60 81 29 20 85 14 39 69 30 62 64 81 71 42 11 50 96 85 55
 32 59 82 73 36 75 10 84 98 46 88 77 38 27 8 56 21 94 77 32 48
 33 44 24 34 68 83 65 75 56 3 14 43 44 84 39 89 85 71 68 14 56
 34 46 99 74 21 26 15 37 68 57 22 98 46 59 95 38 6 64 88 74 84
 35 2 4 13 71 92 55 32 84 71 93 48 66 98 82 96 40 31 77 59 22
 36 41 97 78 61 29 41 29 77 77 48 14 31 14 17 10 68 21 76 95 51
 37 28 24 35 71 39 28 32 67 33 10 45 48 32 38 3 30 2 73 48 43
 38 83 50 20 69 14 93 89 53 49 7 25 27 95 69 53 35 63 92 37 50
 39 28 55 16 28 74 88 12 46 59 14 98 82 30 17 97 58 58 72 59 62
 40 21 91 48 86 66 27 47 24 82 91 30 51 13 24 11 31 36 87 4 61
 41 80 46 12 27 86 77 19 52 59 5 90 90 68 66 65 11 64 66 42 10
 42 71 58 11 41 10 81 97 96 70 43 92 63 19 75 47 11 52 98 93 87

43	4 17 80 86 27 19 7 2 76 30 35 85 57 52 76 6 8 40 32 99
44	60 47 9 55 8 76 12 88 10 79 13 36 65 59 22 59 94 31 30 40
45	34 82 24 17 7 55 43 33 65 39 75 69 13 4 17 64 51 75 16 91
46	55 6 76 62 97 67 89 27 19 34 55 67 63 73 14 65 36 45 95 64
47	53 15 32 96 84 65 14 49 77 77 80 81 26 56 11 23 82 98 58 62
48	96 91 35 59 56 8 33 78 86 81 67 18 96 19 69 80 30 90 12 53
49	37 74 66 53 61 18 56 82 21 11 3 81 53 39 91 75 17 4 95 33
50	37 42 48 93 9 56 57 65 75 10 93 72 94 51 53 63 21 23 21 16

ปัญหา 100 × 5

100	5		
1	73 34 8 62 10	26	69 52 31 18 80
2	84 46 37 86 60	27	4 40 92 89 84
3	57 97 38 46 34	28	21 76 84 49 18
4	52 88 30 3 96	29	25 92 25 20 68
5	66 52 20 22 79	30	74 22 20 76 6
6	67 49 68 33 62	31	11 46 65 90 98
7	33 88 4 90 6	32	85 19 36 64 47
8	62 15 78 94 15	33	50 18 67 89 6
9	65 55 41 98 94	34	21 50 96 16 55
10	7 6 2 9 39	35	93 31 35 3 73
11	6 77 62 84 85	36	98 41 59 12 70
12	31 49 38 69 17	37	91 51 76 45 56
13	42 46 95 26 55	38	26 73 66 67 46
14	82 14 82 22 59	39	4 22 83 97 50

15	48 35 96 49 48	40	28 21 37 80 90
16	45 40 56 90 11	41	61 79 33 97 89
17	79 1 61 81 63	42	8 93 2 92 88
18	86 70 34 12 98	43	55 98 95 56 50
19	10 46 1 54 33	44	68 89 20 50 99
20	47 28 9 53 49	45	30 44 84 95 41
21	67 73 57 18 41	46	26 64 43 25 36
22	86 24 80 56 82	47	81 93 4 52 16
23	64 44 58 43 12	48	95 66 44 21 84
24	38 55 7 50 61	49	49 14 74 47 98
25	13 94 56 82 94	50	6 10 92 4 44

51	82 36 55 67 42	76	44 37 71 54 35
52	71 36 98 62 43	77	8 90 52 26 3
53	61 69 8 63 62	78	55 51 96 32 56
54	84 79 30 25 49	79	55 84 89 10 1
55	29 90 18 3 22	80	27 16 79 91 17
56	44 51 28 55 16	81	69 39 91 90 70
57	83 81 54 87 14	82	79 16 57 31 45
58	12 16 68 84 55	83	97 37 2 7 87
59	54 28 42 28 45	84	39 19 33 6 37
60	19 26 28 73 20	85	54 15 42 77 70
61	52 97 12 8 35	86	75 68 1 91 84
62	59 33 57 38 89	87	88 7 40 42 69
63	74 77 81 80 7	88	87 58 76 5 7
64	42 68 19 53 64	89	97 47 85 46 5
65	60 28 16 70 36	90	4 27 99 54 97
66	43 46 88 9 5	91	79 82 65 56 47
67	80 16 75 87 63	92	59 72 98 1 12

68	34 96 11 33 8	93	7 1 78 24 54
69	74 59 57 11 76	94	89 7 61 62 84
70	46 56 99 65 95	95	61 22 83 47 49
71	40 4 4 92 40	96	42 40 30 63 89
72	27 52 51 34 51	97	79 59 14 39 86
73	79 89 26 64 98	98	85 16 62 63 37
74	1 84 6 96 77	99	66 7 41 95 52
75	98 42 16 67 92	100	99 72 91 45 85

ปัญหา 100 × 10

100	10		
1	52 82 63 79 80 53 71 95 92 61	26	86 63 24 34 35 39 98 23 13 6
2	95 21 55 21 13 19 90 53 62 86	27	73 5 75 9 2 74 78 44 23 42
3	42 79 59 93 64 99 59 54 11 16	28	4 93 23 22 76 46 81 30 13 39
4	75 95 35 32 77 62 82 22 83 42	29	40 53 33 74 46 17 48 82 49 2
5	44 46 21 8 17 88 22 84 87 14	30	98 85 60 14 72 30 25 83 18 7
6	57 23 59 45 78 93 88 54 66 92	31	12 81 20 14 69 62 81 42 50 85
7	89 40 78 37 82 34 35 2 98 67	32	59 73 75 84 46 77 27 56 94 32
8	53 95 25 78 4 72 49 80 42 77	33	44 34 83 75 3 43 84 89 71 14
9	84 87 30 26 72 42 78 84 23 46	34	46 74 26 37 57 98 59 38 64 74
10	62 37 38 98 93 65 69 66 45 41	35	2 13 92 32 71 48 98 96 31 59
11	91 24 78 17 68 39 76 25 52 78	36	41 78 29 29 77 14 14 10 21 95
12	14 24 79 25 25 79 2 16 6 3	37	28 35 39 32 33 45 32 3 2 48
13	95 65 58 21 67 9 14 79 3 72	38	83 20 14 89 49 25 95 53 63 37
14	89 62 44 28 80 26 3 90 64 95	39	28 16 74 12 59 98 30 97 58 59

15	4 19 38 68 43 72 22 51 55 53	40	21 48 66 47 82 30 13 11 36 4
16	95 67 76 24 93 29 26 29 97 59	41	80 12 86 19 59 90 68 65 64 42
17	2 66 70 62 21 36 44 29 83 34	42	71 11 10 97 70 92 19 47 52 93
18	97 6 72 89 33 48 1 90 42 66	43	4 80 27 7 76 35 57 76 8 32
19	68 65 85 60 14 57 4 83 81 42	44	60 9 8 12 10 13 65 22 94 30
20	20 59 8 64 30 95 16 83 92 63	45	34 24 7 43 65 75 13 17 51 16
21	33 2 10 38 59 93 55 19 68 27	46	55 76 97 89 19 55 63 14 36 95
22	51 67 84 90 83 79 43 95 46 92	47	53 32 84 14 77 80 26 11 82 58
23	98 82 42 87 85 88 87 87 56 8	48	96 35 56 33 86 67 96 69 30 12
24	8 90 67 1 85 77 35 12 88 65	49	37 66 61 56 21 3 53 91 17 95
25	85 30 20 99 70 94 76 34 50 34	50	37 48 9 57 75 93 94 53 21 21
51	63 16 94 22 96 54 27 3 80 74	76	93 66 17 38 96 51 91 22 49 89
52	99 26 34 6 3 67 93 80 38 38	77	33 36 23 62 22 26 14 37 83 37
53	69 46 89 24 50 25 49 78 55 4	78	75 53 60 45 95 44 63 3 48 35
54	70 66 62 55 57 77 63 32 34 31	79	68 26 76 22 73 56 65 52 13 91
55	53 76 47 48 66 38 65 53 85 62	80	33 52 39 85 77 31 74 44 78 69
56	21 31 66 57 84 98 34 43 44 39	81	60 29 85 39 30 64 71 11 96 55
57	10 36 76 78 98 96 10 85 47 97	82	82 36 10 98 88 38 8 21 77 48
58	31 8 15 5 55 20 56 19 66 57	83	24 68 65 56 14 44 39 85 68 56
59	80 37 18 50 70 15 51 48 19 9	84	99 21 15 68 22 46 95 6 88 84
60	18 21 54 83 32 36 97 49 66 54	85	4 71 55 84 93 66 82 40 77 22
61	5 3 24 70 31 65 52 66 61 13	86	97 61 41 77 48 31 17 68 76 51
62	18 76 55 21 64 97 46 22 60 47	87	24 71 28 67 10 48 38 30 73 43
63	17 67 96 71 11 27 16 37 98 6	88	50 69 93 53 7 27 69 35 92 50
64	71 5 10 58 9 25 50 51 82 70	89	55 28 88 46 14 82 17 58 72 62
65	90 47 12 36 32 61 96 82 79 19	90	91 86 27 24 91 51 24 31 87 61
66	93 72 96 50 58 24 85 59 71 97	91	46 27 77 52 5 90 66 11 66 10
67	14 66 53 31 98 97 61 88 28 41	92	58 41 81 96 43 63 75 11 98 87

68	49 56 92 86 95 61 76 77 74 1	93	17 86 19 2 30 85 52 6 40 99
69	52 95 77 29 25 75 30 19 27 57	94	47 55 76 88 79 36 59 59 31 40
70	7 49 6 30 4 92 90 32 33 60	95	82 17 55 33 39 69 4 64 75 91
71	78 47 91 93 45 73 42 52 13 62	96	6 62 67 27 34 67 73 65 45 64
72	57 26 14 49 60 21 88 9 9 14	97	15 96 65 49 77 81 56 23 98 62
73	41 81 41 83 87 29 37 96 12 90	98	91 59 8 78 81 18 19 80 90 53
74	75 56 30 89 31 3 43 23 51 76	99	74 53 18 82 11 81 39 75 4 33
75	98 76 85 44 1 96 88 64 16 12	100	42 93 56 65 10 72 51 63 23 16

ปัญหา 100 × 20

100	20
1	12 27 24 42 5 27 51 48 42 31 48 51 8 1 1 28 4 44 6 57
2	72 97 57 16 42 69 9 44 18 49 92 84 37 77 59 81 40 6 49 88
3	29 53 68 44 47 38 22 73 31 78 43 28 33 92 53 96 5 18 84 88
4	16 16 58 75 47 66 15 95 18 23 35 63 4 57 22 99 37 51 72 43
5	22 36 76 32 47 21 84 72 55 37 7 86 8 46 10 79 80 77 39 25
6	61 28 70 66 78 65 38 12 82 34 51 99 38 75 48 97 46 47 51 81
7	69 50 72 31 94 3 71 30 64 43 85 27 97 2 94 15 14 27 77 3
8	71 41 56 6 18 7 11 22 36 7 66 24 75 14 46 6 81 17 59 3
9	86 42 56 6 14 10 45 25 48 69 87 36 76 14 57 77 16 42 97 78
10	16 86 55 52 97 11 48 92 74 97 71 83 93 79 55 98 10 84 5 97
11	54 2 89 7 54 41 29 5 17 15 16 24 69 64 66 5 79 3 83 74
12	53 45 23 92 73 64 31 44 60 38 40 54 10 8 69 51 35 69 51 71
13	64 86 42 45 94 9 14 90 18 80 22 34 25 73 7 51 41 3 9 50
14	96 43 26 70 17 15 56 35 88 78 12 49 40 15 13 60 83 6 9 70

15 79 92 87 72 97 71 59 24 37 89 2 43 88 71 64 17 13 94 59 23
 16 68 73 17 48 7 86 35 64 29 95 90 5 90 6 83 98 93 58 30 43
 17 98 99 38 86 94 2 7 82 19 1 63 70 57 54 39 46 55 65 86 51
 18 87 84 24 63 94 26 20 7 51 18 52 56 39 59 46 87 48 36 49 97
 19 40 83 82 61 68 57 48 34 20 11 15 71 29 49 59 44 35 28 46 57
 20 74 61 55 92 4 78 88 91 7 11 6 12 97 38 36 18 75 68 13 77
 21 22 7 47 73 60 84 54 59 43 92 18 30 51 21 69 71 66 73 45 90
 22 23 8 7 49 70 20 30 5 77 99 87 77 97 60 47 93 82 81 51 58
 23 47 62 99 53 68 21 81 47 95 81 66 55 5 8 83 97 3 12 68 75
 24 55 3 89 20 22 3 2 96 92 79 7 92 75 35 25 44 79 89 82 4
 25 21 21 20 97 17 39 1 7 27 53 43 78 91 81 12 18 78 53 6 12

 26 90 50 65 58 59 64 78 53 70 47 27 79 80 96 28 59 48 14 99 2
 27 68 71 3 59 98 51 8 60 91 48 78 52 81 61 30 9 99 88 12 4
 28 14 95 45 29 55 83 8 77 74 32 80 19 60 89 49 45 91 78 8 28
 29 42 70 89 98 76 12 71 61 62 58 73 16 97 58 81 78 29 63 36 54
 30 41 60 96 61 3 16 60 14 80 33 86 21 14 32 56 67 87 81 24 39
 31 40 4 16 83 21 3 41 88 85 65 82 37 2 59 96 65 27 29 82 30
 32 63 58 42 8 10 1 39 97 8 86 33 85 55 89 89 50 2 15 73 71
 33 23 16 70 83 11 88 8 92 39 60 92 56 97 99 73 87 48 81 66 89
 34 75 30 69 73 44 80 96 65 85 47 8 29 60 7 52 29 33 84 69 33
 35 4 80 77 2 29 77 3 73 83 89 87 56 8 79 12 80 92 36 71 85
 36 31 71 46 56 43 2 79 19 76 40 64 23 30 38 98 14 15 8 70 2
 37 87 59 20 7 60 14 48 99 31 72 7 19 87 1 12 70 55 60 61 69
 38 25 75 71 15 14 60 23 92 12 54 26 52 42 67 40 10 24 24 42 20
 39 47 52 91 55 46 45 38 19 69 29 79 39 16 63 70 30 59 57 77 64
 40 70 25 8 3 3 39 85 66 26 57 49 4 56 42 35 19 41 87 10 94
 41 1 86 34 73 60 32 32 25 54 93 71 46 86 3 58 51 49 61 2 3
 42 49 48 72 13 72 97 69 79 91 65 72 59 4 29 95 4 88 57 6 41

43 59 53 40 47 90 98 56 36 61 39 69 22 30 23 41 23 67 85 24 37
 44 11 54 82 76 1 65 99 10 27 8 30 76 15 21 71 9 49 45 97 36
 45 93 98 60 15 42 86 51 23 88 87 28 34 24 25 90 62 81 69 15 74
 46 38 36 59 43 74 6 59 37 53 57 46 63 23 48 3 30 89 15 91 53
 47 5 9 59 44 54 14 47 76 47 5 75 32 57 33 68 20 43 7 32 49
 48 43 78 43 13 41 99 79 83 39 56 78 72 81 92 5 69 24 87 68 77
 49 82 80 72 16 15 93 67 58 14 80 34 92 77 2 18 88 70 58 17 13
 50 33 88 10 18 38 54 83 99 78 2 4 72 43 12 26 22 74 61 1 37

51 26 52 63 16 24 54 58 29 90 17 23 26 82 32 2 43 44 82 49 34
 52 89 30 67 18 91 92 49 35 90 84 69 46 97 38 21 63 23 73 25 31
 53 86 62 12 3 49 57 50 49 50 86 33 30 3 67 86 67 89 84 46 15
 54 10 8 41 11 23 39 19 89 64 65 11 49 46 93 8 22 96 63 37 32
 55 16 45 5 98 63 13 64 79 94 53 5 3 86 80 33 82 80 14 56 45
 56 72 77 25 9 88 70 5 82 79 5 54 81 62 75 88 41 44 73 50 50
 57 96 90 67 28 1 73 5 92 10 16 42 4 26 59 33 69 69 84 9 81
 58 47 52 95 44 30 16 52 45 18 58 12 46 56 51 65 81 7 9 62 46
 59 9 39 46 14 85 43 15 42 7 60 40 95 67 7 8 21 10 38 10 93
 60 56 95 37 39 22 9 26 21 26 78 30 20 87 65 89 54 15 67 89 13
 61 8 86 41 76 94 2 67 51 51 25 51 56 46 6 46 42 55 76 66 47
 62 69 15 2 92 39 76 11 30 23 71 94 57 66 59 76 53 27 18 98 50
 63 87 98 25 59 70 38 77 70 43 93 24 85 53 30 57 52 55 70 10 70
 64 84 59 99 95 85 66 74 2 95 99 28 31 34 67 51 26 2 69 70 21
 65 62 22 48 11 7 21 88 20 57 6 84 44 52 41 58 69 82 4 92 41
 66 52 43 24 55 65 32 83 82 30 62 69 1 81 32 62 80 42 58 36 12
 67 82 85 18 56 43 78 96 13 86 60 44 11 68 63 59 46 88 66 82 18

68 66 93 10 16 4 41 73 92 80 33 98 72 48 7 71 10 84 11 54 28
 69 67 6 23 56 60 52 59 91 55 57 36 6 4 8 12 81 38 1 34 22
 70 33 46 66 23 90 88 94 2 76 79 21 68 25 89 11 99 69 51 34 60
 71 3 51 82 84 73 76 15 80 12 7 9 62 76 51 82 6 31 38 5 69
 72 53 97 84 22 21 14 50 38 91 69 48 18 46 81 56 90 5 16 12 49
 73 95 79 14 32 25 86 20 54 95 82 30 21 13 9 57 79 3 81 3 38
 74 42 46 13 40 33 35 54 45 28 26 15 48 72 13 53 34 37 39 56 80
 75 59 71 88 47 48 81 16 19 12 41 21 32 72 27 45 56 14 56 44 52

76 77 18 7 92 25 32 33 65 18 52 43 42 82 74 96 75 35 83 41 45
 77 38 33 40 61 27 75 9 35 80 38 91 79 32 59 31 54 61 27 14 34
 78 12 13 20 51 37 38 74 66 32 21 49 74 25 98 39 94 43 29 25 3
 79 1 3 68 74 63 55 11 26 67 84 86 82 58 31 12 84 38 93 78 61
 80 66 32 22 85 28 96 21 18 2 74 36 3 17 98 13 55 48 70 25 96
 81 24 28 92 6 88 19 97 32 4 76 41 64 99 48 8 64 89 64 89 99
 82 17 48 53 13 27 35 15 51 30 16 10 90 54 74 49 14 86 22 83 49
 83 58 8 16 32 94 41 43 23 64 75 75 10 81 76 4 18 37 84 78 78
 84 71 42 10 8 20 77 2 18 69 75 56 72 73 75 33 6 20 25 23 34
 85 88 26 23 43 54 36 1 87 76 36 70 2 93 78 22 27 33 73 51 78
 86 63 26 88 30 17 24 74 74 92 79 73 49 73 11 30 20 1 81 7 79
 87 60 95 65 53 74 52 37 10 50 71 57 88 79 71 34 54 8 10 20 64
 88 86 44 11 24 35 60 61 76 35 82 51 53 54 54 19 30 61 46 30 55
 89 8 74 89 2 15 43 3 31 20 70 34 59 72 10 24 58 41 32 30 63
 90 8 82 78 11 53 30 75 31 23 22 62 88 37 96 56 35 40 38 30 24
 91 44 28 94 33 64 86 41 93 52 39 3 97 68 97 96 66 10 68 63 58
 92 99 4 32 27 38 6 86 61 25 8 34 98 22 75 35 85 10 40 82 25

93	80 42 64 35 79 8 51 26 82 22 27 94 94 41 43 78 75 33 40 93
94	76 81 42 40 45 31 65 39 18 79 29 95 37 30 8 15 9 79 39 73
95	94 56 74 40 67 80 42 83 2 90 25 93 15 73 5 29 81 64 37 45
96	75 38 87 96 93 82 50 11 29 55 97 95 31 84 65 31 40 36 44 93
97	63 50 63 69 16 2 48 57 70 77 30 6 46 22 64 45 13 62 2 37
98	57 12 94 23 59 59 88 81 7 3 28 1 33 8 20 41 10 82 27 64
99	76 43 87 32 47 47 52 40 36 20 48 15 82 98 54 84 53 44 71 92
100	34 51 72 13 95 21 46 97 96 61 99 66 75 68 38 51 28 99 45 59

ปัญหา 200 × 10

200	10		
1	17 30 28 15 52 5 53 55 81 47	26	75 64 95 20 92 74 34 79 35 4
2	37 26 82 67 82 44 43 65 44 51	27	83 28 50 68 41 68 68 66 25 11
3	22 2 88 57 17 2 43 38 46 44	28	2 13 88 3 38 30 77 19 21 60
4	71 29 33 64 66 98 56 76 3 27	29	60 7 19 17 26 69 31 41 44 23
5	79 98 69 23 62 27 48 45 93 76	30	62 13 50 55 2 17 10 71 40 87
6	37 67 28 9 57 10 83 99 14 4	31	83 93 25 59 90 34 57 60 20 50
7	69 92 92 1 58 25 34 84 86 76	32	73 93 88 18 79 95 21 72 83 18
8	53 51 66 2 91 27 93 67 23 49	33	55 56 1 31 17 3 56 62 76 23
9	59 53 92 43 42 41 95 40 59 5	34	3 72 6 84 61 24 36 17 26 24
10	64 82 30 73 69 59 48 93 47 25	35	71 71 45 61 81 51 26 25 50 82
11	11 86 67 94 48 89 63 70 76 23	36	34 28 29 55 67 55 11 59 98 16
12	67 47 53 96 46 36 93 45 45 67	37	40 95 2 46 18 60 6 95 6 96
13	27 36 93 78 18 62 91 75 64 55	38	13 96 54 77 1 61 86 47 83 37
14	64 23 45 14 94 29 94 91 10 40	39	62 55 71 59 36 15 2 10 51 29

15	59 34 7 32 83 17 32 21 4 58	40	56 58 81 98 31 38 89 94 46 19
16	31 66 70 23 48 37 81 72 29 26	41	28 66 22 23 85 83 28 39 99 26
17	43 43 58 83 81 35 69 15 69 36	42	53 3 57 18 92 62 62 53 65 56
18	42 71 58 29 57 47 29 73 46 83	43	71 77 16 17 6 84 63 50 46 33
19	56 31 17 75 93 37 44 43 68 65	44	5 80 88 70 78 44 52 99 24 53
20	88 36 41 55 67 52 2 62 89 91	45	73 2 5 73 83 80 12 63 47 6
21	71 41 56 91 19 3 66 40 48 77	46	95 82 52 38 39 48 13 59 24 95
22	86 6 34 44 65 87 25 43 68 62	47	23 86 23 13 74 45 43 30 32 63
23	6 28 62 46 49 4 60 96 13 6	48	49 16 21 77 77 54 99 67 66 84
24	54 43 81 59 45 50 69 39 36 57	49	17 29 21 74 85 71 11 18 48 13
25	92 18 44 11 22 96 44 23 85 87	50	81 24 58 79 88 10 70 12 83 6
51	88 89 96 51 25 96 82 9 3 19	76	3 80 40 40 55 29 3 78 47 77
52	62 96 37 48 33 23 14 41 74 16	77	45 30 73 25 62 43 37 18 73 27
53	35 99 46 16 36 8 39 67 97 80	78	55 24 91 94 21 17 15 59 1 29
54	78 15 78 93 32 56 83 52 39 37	79	92 58 7 73 71 78 50 11 19 83
55	90 49 87 29 82 35 4 39 80 20	80	67 43 1 44 80 28 75 67 92 22
56	36 78 91 28 98 72 86 90 27 62	81	13 6 71 98 5 47 80 36 31 10
57	46 48 9 6 3 53 93 25 70 81	82	30 95 72 94 48 78 3 24 29 42
58	98 64 96 82 46 13 87 13 65 52	83	27 43 8 61 87 77 9 59 68 64
59	75 3 93 40 68 72 82 81 1 14	84	19 99 68 65 22 92 28 24 31 80
60	36 63 98 9 11 38 78 3 26 67	85	51 22 49 44 5 42 44 39 92 73
61	74 14 31 88 30 1 32 8 46 19	86	86 99 7 75 71 86 2 1 48 91
62	86 95 78 51 75 69 82 66 8 86	87	20 11 47 4 69 6 71 80 87 38
63	67 88 20 48 42 85 24 67 74 1	88	95 99 90 17 75 19 89 58 8 60
64	87 9 75 1 93 2 88 51 47 53	89	79 60 59 1 72 28 24 76 90 53
65	68 61 18 4 36 28 45 54 19 21	90	48 42 39 12 36 18 73 59 57 38
66	1 24 34 77 97 18 48 85 85 33	91	83 33 49 78 54 4 28 90 40 19
67	87 38 48 48 1 10 67 10 62 68	92	91 68 28 51 24 24 98 92 25 14

68	87 50 49 6 63 70 14 95 33 28	93	78 78 22 55 55 57 17 3 49 35
69	7 47 69 1 86 67 34 59 96 25	94	41 26 10 53 5 82 27 51 26 10
70	34 42 88 32 54 95 76 64 33 69	95	61 77 96 96 40 7 90 1 35 44
71	1 30 30 79 33 72 1 90 5 71	96	90 1 2 92 56 4 4 55 90 97
72	91 97 34 40 71 64 51 28 23 60	97	57 13 88 82 8 94 13 50 99 17
73	7 49 22 14 1 19 15 48 43 69	98	4 67 90 20 19 9 83 15 30 50
74	6 79 81 20 18 60 67 57 63 99	99	21 87 69 22 7 57 43 31 5 22
75	46 19 50 86 75 31 57 85 10 53	100	68 31 44 13 12 82 63 99 52 19
101	87 49 74 40 44 98 68 8 12 70	126	70 88 88 86 87 94 11 8 43 60
102	62 90 16 43 66 54 74 44 11 88	127	62 39 44 10 20 17 63 86 55 54
103	69 6 90 23 39 2 12 91 72 31	128	81 56 74 57 84 45 74 62 89 52
104	79 76 7 30 41 71 69 83 32 7	129	61 24 18 88 28 68 6 41 4 71
105	69 94 14 16 60 94 38 75 81 81	130	65 91 95 59 59 60 23 66 1 9
106	35 76 47 76 11 69 74 66 21 58	131	65 86 2 73 65 93 94 94 97 99
107	25 79 48 30 31 51 98 98 81 2	132	33 80 22 43 56 30 93 54 65 41
108	68 66 36 59 65 80 86 77 64 89	133	62 58 7 25 2 5 26 9 38 30
109	14 51 62 64 16 79 72 8 45 1	134	19 54 82 37 36 31 49 75 87 23
110	9 14 1 72 67 85 76 6 4 80	135	57 57 52 70 41 30 30 43 58 60
111	74 89 63 61 19 56 53 26 10 52	136	82 42 17 97 7 71 50 9 47 1
112	61 87 86 13 69 79 94 47 90 15	137	78 32 59 29 23 29 47 10 53 67
113	58 1 49 81 56 46 45 55 11 25	138	94 5 42 33 25 82 89 79 51 55
114	81 65 48 25 96 71 30 39 88 11	139	65 43 51 99 81 20 43 10 40 64
115	52 53 3 90 78 85 39 91 52 97	140	14 80 79 34 6 13 32 97 80 21
116	93 19 51 67 36 52 60 78 96 90	141	9 86 93 96 67 94 45 39 20 16
117	44 21 80 21 56 42 69 49 92 62	142	29 47 65 65 5 47 47 30 24 94

118	57 98 86 70 12 66 12 23 44 69	143	70 95 27 90 89 57 68 74 77 11
119	89 90 97 96 37 82 83 92 54 34	144	90 86 25 52 71 95 13 52 37 90
120	81 45 4 99 1 91 21 68 28 66	145	86 40 60 95 86 8 86 90 13 48
121	69 77 58 64 50 48 25 88 85 54	146	12 31 72 3 48 46 97 12 29 85
122	98 54 79 68 43 70 50 9 51 84	147	40 61 29 64 50 80 2 61 28 34
123	78 86 28 61 58 21 47 38 21 62	148	12 12 55 21 54 32 16 34 18 7
124	70 2 19 36 13 82 43 60 4 71	149	64 64 58 16 21 7 27 88 22 79
125	3 3 93 57 55 12 58 2 42 70	150	32 17 22 70 83 34 38 27 75 95
151	18 52 58 67 27 33 53 68 24 50	176	41 14 75 49 27 86 27 22 47 28
152	11 56 61 13 38 33 37 14 79 78	177	58 58 96 94 36 84 99 43 22 8
153	40 35 89 96 79 18 94 88 99 39	178	62 89 28 63 94 15 11 60 9 8
154	14 60 67 57 71 29 78 74 35 79	179	45 44 12 96 65 99 91 20 68 48
155	22 75 14 7 53 19 84 71 45 69	180	67 16 99 43 89 8 78 32 4 7
156	49 10 10 30 2 74 49 14 95 27	181	64 20 63 73 23 87 97 67 67 55
157	95 50 40 96 9 58 25 16 46 42	182	6 69 19 19 93 61 23 1 92 51
158	82 66 11 98 49 52 59 46 15 15	183	77 67 26 19 14 42 48 42 49 89
159	7 53 99 72 79 10 36 47 7 51	184	94 80 8 81 57 25 20 95 23 61
160	19 58 7 31 39 70 33 54 24 44	185	24 81 72 82 93 35 99 84 37 82
161	95 12 20 11 28 78 15 91 45 71	186	38 82 55 69 60 64 39 37 16 39
162	39 66 78 36 39 59 36 89 20 32	187	73 27 67 13 82 74 53 30 39 89
163	71 81 6 10 70 32 47 59 35 66	188	59 5 1 85 16 40 3 12 47 47
164	34 51 71 20 72 20 83 39 61 49	189	61 44 23 32 69 89 65 87 94 24
165	89 75 48 26 70 12 24 52 97 8	190	14 38 35 34 57 4 81 13 71 79
166	63 71 39 67 38 98 71 91 69 7	191	59 53 6 57 8 43 47 60 36 86
167	85 46 58 13 48 18 7 21 78 44	192	83 16 89 69 52 12 25 92 72 60

168	60 88 7 60 57 12 6 90 13 1	193	3 48 93 63 98 28 36 52 67 67
169	91 58 17 50 11 57 55 58 53 15	194	46 16 88 49 67 68 44 51 78 10
170	49 73 63 28 38 73 43 74 16 6	195	16 31 86 56 19 46 29 91 13 61
171	46 27 73 32 37 91 82 30 94 51	196	56 79 37 37 56 23 85 76 68 60
172	10 26 43 49 56 46 40 63 59 91	197	40 75 53 54 38 3 26 32 36 68
173	19 38 14 54 64 64 78 19 64 33	198	7 14 45 60 75 70 4 81 69 72
174	50 48 66 46 64 92 64 18 72 77	199	79 67 32 97 69 81 9 57 28 9
175	65 8 59 15 68 45 96 88 13 69	200	55 70 69 44 14 14 59 71 69 47

ปัญหา 200 × 20

200	20
1	68 10 77 68 26 99 88 68 93 10 9 79 44 84 35 18 75 89 52 32
2	40 76 30 48 97 58 35 29 33 99 93 2 40 1 3 77 43 41 89 78
3	43 75 88 56 5 61 33 75 48 37 84 54 27 47 9 73 70 16 39 84
4	72 29 21 54 89 33 93 63 71 79 58 78 41 60 71 91 11 7 71 27
5	87 10 29 38 77 30 84 2 46 24 14 71 9 31 99 96 64 84 72 32
6	59 86 12 65 47 49 19 18 23 82 31 65 77 76 89 10 88 53 71 84
7	67 62 2 56 60 54 43 43 7 12 19 95 98 12 92 58 94 56 68 74
8	89 26 95 29 12 36 14 6 70 84 16 21 64 75 60 70 9 71 42 72
9	84 1 24 21 96 38 22 13 78 49 41 18 14 90 44 42 29 40 94 85
10	43 62 20 99 63 63 1 55 71 15 9 54 2 72 59 21 58 17 83 60
11	92 89 66 34 27 97 55 84 34 48 67 1 19 8 8 11 90 33 14 55
12	90 2 82 25 87 45 5 1 43 58 74 87 9 86 51 88 78 3 76 60
13	23 62 45 66 46 65 14 70 88 9 85 29 19 86 60 91 63 51 90 8
14	20 96 4 54 10 26 82 25 35 74 90 19 28 70 35 78 34 76 14 34

15 97 99 22 1 97 76 84 38 6 22 77 38 69 46 5 70 71 2 83 54
 16 57 91 81 52 59 23 10 98 59 43 40 44 90 52 81 53 49 63 3 47
 17 33 85 56 23 58 40 17 17 43 84 33 46 71 31 67 51 69 84 1 60
 18 7 42 85 50 73 91 39 23 98 18 42 25 83 36 70 96 97 55 12 8
 19 69 7 3 61 11 51 39 28 22 30 74 48 2 10 79 27 10 3 51 39
 20 56 4 82 36 31 41 75 80 13 86 16 66 16 19 7 47 43 82 46 42
 21 16 23 94 60 41 8 59 24 43 53 45 81 9 85 51 30 22 46 2 10
 22 41 74 72 1 23 42 51 29 64 87 62 42 60 49 1 37 16 58 92 62
 23 7 83 96 57 36 71 30 16 18 79 44 48 1 61 38 63 31 24 23 52
 24 74 41 55 38 54 20 71 13 25 83 68 34 44 10 91 61 22 57 69 65
 25 54 63 3 68 7 54 95 35 62 93 76 73 25 26 3 73 62 59 65 32

 26 67 99 63 45 31 19 17 31 53 7 76 32 86 38 88 80 14 86 30 61
 27 15 55 65 85 90 32 46 40 85 78 78 68 45 97 88 5 11 39 48 9
 28 85 37 20 13 83 65 61 40 31 31 29 60 68 71 6 82 46 55 70 25
 29 83 23 4 55 65 27 37 22 47 95 15 63 1 34 68 49 85 69 15 15
 30 42 1 95 98 61 20 93 59 23 32 30 49 56 74 26 11 59 36 62 56
 31 29 84 20 12 15 99 28 78 95 55 93 19 64 12 38 79 73 98 48 3
 32 64 46 57 97 10 81 57 10 74 46 16 54 7 77 93 78 22 82 19 13
 33 3 31 38 6 61 87 7 94 36 3 38 14 68 16 89 57 70 44 49 59
 34 9 93 46 94 55 4 53 80 29 93 27 40 96 55 85 47 52 70 5 95
 35 35 79 76 43 81 10 82 86 39 36 32 83 30 90 82 91 70 67 22 93
 36 24 32 10 86 87 2 22 18 43 77 31 66 62 61 99 92 45 74 45 49
 37 72 71 79 2 81 61 50 34 59 67 88 27 32 43 47 94 45 61 15 67
 38 70 60 7 84 50 10 93 69 24 29 79 89 25 42 60 33 37 38 29 17
 39 47 42 12 94 43 22 29 94 42 70 86 69 27 81 4 52 2 87 65 39
 40 78 24 60 28 67 59 9 9 88 34 61 34 97 1 41 12 90 35 72 91
 41 60 51 85 68 54 45 76 87 16 45 30 27 54 8 86 37 2 37 45 60
 42 11 98 73 45 79 8 20 24 23 16 55 3 28 51 12 47 40 98 53 90

43 60 64 18 43 26 27 6 44 33 99 8 97 65 16 19 39 70 26 8 94
 44 5 85 75 89 64 24 39 31 20 70 90 40 7 14 80 4 93 49 11 53
 45 24 31 77 86 83 53 49 38 2 48 70 57 71 25 37 50 44 12 21 25
 46 52 11 71 8 56 53 31 25 1 86 58 3 74 92 55 90 71 52 46 12
 47 39 95 83 32 8 84 20 29 47 31 35 45 8 98 59 58 83 49 95 63
 48 14 58 63 62 78 46 79 70 39 88 11 42 46 28 55 46 13 92 42 75
 49 65 57 52 11 27 17 46 3 84 69 66 1 76 97 26 14 3 43 66 63
 50 99 28 66 36 1 5 22 69 94 10 63 63 45 38 40 84 99 75 23 7

51 69 76 30 3 97 77 78 77 40 54 26 14 11 95 22 97 27 83 39 80
 52 96 40 71 23 35 96 19 26 74 12 92 29 92 78 72 49 98 3 86 16
 53 54 96 57 79 29 41 87 63 34 10 29 53 54 48 90 33 62 99 4 39
 54 37 54 83 99 22 3 85 91 7 74 98 5 31 66 39 48 17 87 95 37
 55 35 91 34 57 16 99 8 49 20 86 59 71 51 18 60 68 35 60 8 7
 56 19 90 31 4 87 99 2 5 80 56 18 44 48 19 14 36 71 92 41 1
 57 10 41 11 69 79 37 57 30 34 82 39 57 27 25 92 66 94 75 85 38
 58 88 72 50 69 82 2 99 79 50 40 1 77 55 79 47 6 58 53 66 48
 59 61 37 22 30 65 10 69 59 26 93 27 43 39 90 90 54 77 30 68 38
 60 95 38 7 99 62 40 55 73 28 10 4 59 71 26 43 16 29 59 37 39
 61 18 64 82 51 99 63 14 40 8 59 38 25 18 48 15 7 61 95 51 83
 62 49 98 34 66 22 29 24 54 30 11 42 52 48 81 8 35 41 85 46 27
 63 69 36 5 33 79 82 42 30 49 35 69 19 8 95 24 64 47 63 93 14
 64 86 20 62 13 37 14 64 18 4 84 4 34 35 37 93 18 2 21 59 46
 65 28 3 3 89 31 39 96 18 1 88 68 16 62 6 40 8 16 78 14 46
 66 6 41 68 73 30 11 40 62 31 40 30 38 52 86 68 61 41 72 25 59
 67 40 93 35 26 40 18 87 20 18 78 39 61 65 83 68 41 59 87 71 12

68 88 44 79 25 93 44 43 26 1 57 75 53 95 67 67 65 27 52 15 20
 69 94 3 10 88 22 37 96 52 43 21 87 36 55 74 41 71 18 39 89 7
 70 78 19 60 70 91 77 94 90 79 5 45 87 7 91 72 34 33 93 94 85
 71 43 66 67 72 8 40 75 47 98 68 83 3 37 99 72 35 1 22 61 74
 72 31 55 10 56 44 81 27 99 87 73 30 56 91 55 73 92 99 82 5 49
 73 54 66 87 1 95 28 97 18 72 13 55 1 91 79 44 70 3 21 45 48
 74 80 68 25 6 93 52 51 69 70 71 37 4 73 46 34 65 15 65 52 6
 75 11 1 24 97 83 64 75 40 26 35 31 4 56 10 24 76 45 81 9 73

76 27 5 4 57 38 21 33 21 23 13 74 98 22 52 19 71 82 59 70 57
 77 3 59 36 64 74 39 13 49 1 71 49 90 98 22 27 81 20 73 50 44
 78 32 64 44 40 24 97 32 18 68 76 18 23 96 39 31 32 11 74 44 41
 79 2 1 22 4 50 37 93 89 31 28 76 78 82 8 80 27 90 33 79 35
 80 49 5 8 71 75 40 53 86 62 13 46 64 84 13 4 95 65 53 64 68
 81 27 11 49 84 60 75 38 28 48 52 78 90 31 32 94 49 73 47 47 61
 82 81 46 97 50 15 9 16 53 17 58 99 18 44 4 3 51 71 69 4 1
 83 18 10 8 1 73 4 45 14 69 31 58 19 19 7 31 65 26 11 59 16
 84 83 56 9 30 68 69 79 91 13 53 84 67 60 47 38 17 97 31 45 37
 85 84 25 18 84 63 22 90 87 14 12 89 82 19 34 44 91 50 69 91 70
 86 49 95 34 33 79 54 48 18 42 24 74 16 56 1 87 23 43 67 35 26
 87 64 12 54 69 40 86 37 51 37 83 1 53 6 12 77 17 7 3 71 76
 88 20 48 42 19 2 41 72 73 21 71 86 18 45 93 2 19 78 21 63 87
 89 52 11 70 37 45 59 53 35 12 49 95 17 24 62 66 28 27 29 94 42
 90 42 45 59 25 94 56 71 38 37 78 58 25 10 11 20 70 50 28 86 26
 91 51 25 8 27 3 76 22 53 95 15 17 71 6 81 79 67 70 45 3 22
 92 66 62 29 71 16 47 63 41 42 24 18 12 44 26 89 40 65 2 77 2

93 95 12 6 55 29 53 25 29 3 6 70 24 57 31 90 81 96 53 37 81
 94 18 67 42 19 55 38 9 44 21 72 49 32 43 41 84 54 27 36 84 55
 95 37 17 33 43 3 6 38 8 75 48 21 44 1 5 40 52 66 5 50 12
 96 70 59 67 80 72 26 58 93 15 3 43 39 42 40 8 38 18 79 97 47
 97 22 30 7 93 87 27 42 47 5 49 26 40 20 11 65 6 96 76 21 26
 98 87 73 64 76 64 81 84 26 21 81 67 47 83 56 80 8 66 32 42 91
 99 67 34 7 52 33 17 54 61 51 15 66 53 57 67 50 31 82 87 6 37
 100 15 27 60 26 64 87 58 47 99 13 89 75 34 49 89 98 23 4 19 41

101 1 50 79 70 64 22 12 99 38 36 66 83 67 92 29 14 69 44 96 11
 102 37 87 26 36 32 32 99 57 55 94 63 35 64 8 98 17 28 8 51 82
 103 91 77 57 23 23 34 24 50 80 3 88 53 92 79 45 31 95 16 53 26
 104 59 88 98 72 94 37 85 48 36 35 45 1 35 5 66 37 38 93 95 57
 105 2 80 27 23 93 58 35 50 39 37 52 35 55 72 64 85 89 93 76 33
 106 44 75 66 28 69 7 13 2 99 48 89 67 99 77 86 84 19 64 95 21
 107 29 41 89 14 75 53 20 20 59 66 83 26 23 60 75 7 6 65 50 37
 108 37 84 90 48 65 70 54 80 35 31 80 13 52 21 64 54 52 1 52 67
 109 30 24 44 54 68 20 46 6 1 4 58 52 30 23 83 40 70 40 27 22
 110 20 27 91 86 93 88 23 75 96 6 52 31 6 51 12 95 4 74 97 33
 111 19 52 36 81 4 34 27 70 64 23 21 11 81 16 69 97 81 59 96 47
 112 79 73 66 12 91 34 90 16 84 48 34 37 91 73 7 44 1 21 88 3
 113 41 75 71 26 15 27 74 73 38 50 96 13 89 59 27 4 36 79 92 55
 114 79 23 32 76 83 40 96 4 79 41 69 73 45 20 20 46 30 9 33 19
 115 49 52 80 41 38 46 52 45 75 23 81 37 33 83 92 10 13 2 23 61
 116 21 97 25 89 34 38 55 92 12 57 65 56 55 9 28 63 90 70 38 38
 117 97 40 32 70 67 35 12 15 46 5 61 55 74 60 36 76 30 88 37 63

118 86 50 39 54 24 99 61 70 60 5 82 81 46 68 2 23 66 3 11 42
 119 15 95 38 6 78 65 63 41 33 23 75 61 50 14 86 95 28 95 6 20
 120 57 34 5 65 41 58 13 6 81 21 61 75 29 67 71 73 36 77 55 62
 121 76 2 97 7 87 17 56 42 13 58 22 34 2 86 34 78 38 76 40 45
 122 14 76 56 14 55 5 49 42 86 96 52 2 80 35 74 46 48 94 1 78
 123 57 26 65 88 83 41 50 21 5 9 10 29 12 58 86 40 97 90 69 57
 124 80 48 14 98 5 46 85 7 76 40 50 2 63 13 68 34 56 67 96 31
 125 93 16 58 57 34 14 42 43 12 2 30 60 96 55 94 92 92 27 76 33

126 31 20 35 19 16 10 52 64 46 55 84 17 88 57 78 42 93 18 46 71
 127 66 86 97 72 57 70 54 89 39 2 86 55 47 14 38 56 81 49 66 73
 128 30 62 24 41 66 44 47 77 91 9 16 3 21 79 77 4 24 3 8 87
 129 82 54 46 40 23 97 70 41 86 78 54 94 30 72 69 63 38 79 84 46
 130 11 50 91 62 13 79 36 93 49 37 54 70 44 27 56 93 39 12 62 21
 131 71 26 53 47 35 50 49 33 7 52 70 35 64 42 12 72 33 6 95 52
 132 94 4 29 65 8 57 81 36 56 78 37 71 51 59 64 63 7 72 30 95
 133 35 4 72 14 17 68 23 73 75 18 18 62 95 58 99 62 75 46 20 66
 134 71 59 43 99 20 80 64 66 73 85 96 92 57 36 41 30 43 7 90 24
 135 76 24 62 65 18 47 20 64 58 79 80 39 96 63 25 91 92 56 66 27
 136 76 14 27 14 66 27 97 43 88 6 80 47 62 1 58 96 44 33 76 63
 137 26 49 57 99 10 12 36 8 3 23 21 10 11 45 20 53 35 59 46 48
 138 31 35 14 65 48 24 36 75 40 47 58 72 8 10 86 19 2 86 38 43
 139 58 78 92 62 71 88 53 50 56 82 70 5 92 8 58 40 96 33 41 87
 140 31 77 69 30 64 3 88 11 43 66 71 40 1 66 13 14 28 28 83 44
 141 98 72 61 6 97 98 42 41 27 24 62 1 51 62 20 16 13 84 43 87
 142 26 10 8 80 59 29 93 50 93 14 4 15 32 95 11 66 67 59 37 58

143 92 87 78 52 25 23 94 15 30 9 1 67 96 88 33 54 61 54 93 24
 144 11 24 43 35 67 8 61 71 36 37 4 10 88 69 55 54 71 99 80 50
 145 72 26 53 25 5 55 92 76 10 1 69 31 15 38 5 64 20 30 68 33
 146 46 46 2 54 15 86 16 73 98 33 70 40 5 30 71 86 54 14 92 62
 147 20 40 72 44 33 98 4 23 65 94 77 7 56 60 62 31 53 65 94 56
 148 38 86 23 40 6 90 88 21 49 82 28 54 68 33 50 99 75 34 21 86
 149 57 40 85 99 27 46 75 80 59 60 47 86 10 87 93 10 58 6 68 58
 150 5 8 49 21 34 22 85 40 76 79 78 4 89 91 74 2 74 91 53 42

151 19 39 6 74 12 19 28 43 10 4 64 2 43 48 6 74 65 89 76 84
 152 65 11 34 10 28 24 7 40 24 69 58 89 4 46 39 29 21 80 40 35
 153 35 96 6 46 15 95 84 6 84 92 47 36 51 18 90 77 60 34 4 91
 154 89 74 51 99 36 75 63 4 11 51 17 33 84 31 5 14 72 35 89 49
 155 41 22 17 88 22 97 69 46 8 7 67 98 35 22 81 30 4 91 52 52
 156 15 22 66 91 82 95 94 30 33 36 84 70 53 2 28 14 85 44 54 86
 157 99 50 31 57 43 99 23 66 54 96 59 91 82 96 33 48 47 52 87 45
 158 19 6 58 85 97 78 2 16 90 40 51 64 41 56 78 1 80 73 86 62
 159 74 95 84 62 6 30 43 26 30 15 47 41 62 17 95 4 71 93 20 3
 160 63 83 70 46 28 6 26 25 51 53 61 19 78 49 82 71 26 98 15 50
 161 92 70 87 88 65 86 7 39 24 90 51 24 72 87 38 54 73 30 61 83
 162 29 1 99 11 45 59 44 69 4 19 97 84 32 35 41 30 36 20 25 57
 163 47 8 88 27 73 7 12 45 51 19 65 50 17 69 4 42 38 58 85 8
 164 80 94 60 35 63 61 51 39 16 97 7 51 9 66 94 43 13 34 58 88
 165 7 65 4 50 41 53 19 90 67 39 71 60 67 69 44 33 66 14 6 10
 166 43 70 2 68 12 54 5 50 21 37 75 3 74 30 10 32 82 46 40 88
 167 89 61 99 58 11 25 62 32 90 4 47 56 27 30 28 12 2 61 98 50

168 9 80 79 47 85 55 28 92 9 25 83 79 55 77 33 68 76 73 93 22
 169 99 82 49 44 82 8 96 87 18 63 1 8 76 98 66 8 29 87 78 49
 170 38 23 53 37 41 80 9 10 6 95 46 34 1 55 16 82 90 41 55 44
 171 76 51 16 76 87 52 70 43 72 12 54 93 41 78 45 27 36 9 75 6
 172 52 92 64 30 84 67 8 14 33 55 76 79 73 45 51 59 32 57 97 26
 173 48 25 72 98 25 17 94 18 54 89 54 72 89 84 89 20 84 82 52 34
 174 55 23 59 57 59 78 22 91 27 11 59 7 2 47 69 60 78 17 27 40
 175 17 40 62 7 89 57 74 12 46 38 37 90 45 58 23 76 94 19 32 26

176 63 69 66 15 86 92 94 28 52 47 47 2 99 24 9 44 45 32 81 49
 177 92 42 92 84 15 1 13 34 69 45 55 76 37 56 8 29 14 47 73 93
 178 77 13 28 39 17 42 9 72 34 27 76 1 74 50 43 68 72 89 29 77
 179 31 17 72 62 44 70 95 87 29 2 75 7 95 53 64 24 3 2 34 38
 180 44 27 50 1 71 34 78 36 92 98 23 57 78 12 15 46 18 94 35 15
 181 76 74 80 41 15 86 51 7 14 89 41 19 48 43 15 94 58 17 78 50
 182 18 7 48 59 10 78 26 48 62 24 28 2 95 73 26 79 87 98 29 15
 183 7 25 12 92 68 69 58 21 39 58 78 11 11 3 48 68 30 13 45 22
 184 40 73 32 51 58 87 53 56 27 84 58 84 28 32 9 49 73 84 25 88
 185 47 81 76 82 33 49 31 5 4 33 51 9 21 91 11 85 71 12 54 42
 186 73 84 33 49 14 34 74 88 96 24 40 88 8 25 56 87 5 66 74 63
 187 41 40 65 78 3 80 35 16 63 71 16 50 88 44 75 17 77 90 77 56
 188 70 28 18 87 89 38 59 4 15 25 21 52 29 55 75 99 60 86 85 61
 189 86 86 88 60 94 14 39 43 62 14 88 49 60 97 63 76 6 83 64 64
 190 8 17 60 55 19 93 53 49 63 37 59 10 2 10 17 81 61 75 2 63
 191 63 66 67 57 33 86 53 42 40 45 3 23 47 42 93 52 23 67 27 30
 192 77 33 54 70 55 35 43 91 72 30 39 30 15 73 38 24 29 33 81 4

193	91 20 61 11 55 18 18 64 30 27 73 75 1 74 22 57 2 40 17 21
194	24 84 72 62 37 11 37 99 30 43 34 43 64 16 22 8 67 56 31 95
195	19 23 50 53 26 19 8 60 91 36 45 88 55 47 17 36 69 21 37 20
196	63 73 42 95 22 29 69 9 94 88 20 66 32 68 24 67 38 43 20 30
197	72 25 30 97 75 56 18 11 88 44 66 23 49 9 93 57 37 1 69 86
198	17 87 24 9 4 2 68 10 89 20 35 34 20 92 17 66 80 92 59 95
199	46 2 27 85 82 58 63 88 99 29 96 1 33 4 48 5 51 4 39 82
200	66 15 6 17 30 2 48 41 63 48 24 82 68 48 27 69 34 44 13 10

ภาคผนวก ข

แสดงตัวอย่าง ผลของการจัดลำดับงานกับปัญหาขนาดต่างๆ

ปัญหา 50×5 เมื่อแก้ปัญหาคด้วยโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีหาบูที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 217.9 วินาที (4 Processors) ได้ Makespan เท่ากับ 2745 จากลำดับงาน

31 41 42 34 24 49 36 13 8 32 5 47 18 40 30 2 29 45 39 38 43 21 28 12 25
26 50 7 16 10 3 15 20 27 44 4 6 17 35 37 1 9 14 22 23 46 11 33 19 48

ปัญหา 50×10 เมื่อแก้ปัญหาคด้วยโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีหาบูที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 321.7 วินาที (4 Processors) ได้ Makespan เท่ากับ 3025 จากลำดับงาน

18 34 44 33 37 43 42 49 14 6 30 2 20 22 23 10 38 40 15 31 25 9 35 29 27
46 4 11 48 41 36 13 28 47 12 3 1 7 17 32 5 21 45 26 16 50 8 19 24 39

ปัญหา 50×20 เมื่อแก้ปัญหาคด้วยโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีทาบที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 514.1 วินาที (4 Processors) ได้ Makespan เท่ากับ 3899 จากลำดับงาน

35 43 31 37 34 39 47 14 17 15 5 8 7 10 33 28 45 27 29 6 24 49 1 42 46
16 48 19 40 30 2 20 44 36 41 23 12 9 22 26 11 21 13 32 4 18 38 25 50 3

ปัญหา 100×5 เมื่อแก้ปัญหาคด้วยโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีทาบที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 1003.6 วินาที (4 Processors) ได้ Makespan เท่ากับ 5527 จากลำดับงาน

27 74 26 4 71 62 45 73 15 46 22 94 31 60 85 7 41 50 51 36 8 91 70 81 77
14 21 25 28 40 32 13 20 86 39 24 95 34 48 68 44 17 69 72 99 18 6 10 53
61 37 80 76 59 87 56 75 33 12 97 90 84 3 16 43 96 65 42 93 98 38 52 55
82 58 83 100 49 63 11 19 92 78 9 30 54 57 47 1 64 5 67 2 66 79 29 35 89
23 88

ปัญหา 100×10 เมื่อแก้ปัญหาคด้วยโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีทาบที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 1638.9 วินาที (4 Processors) ได้ Makespan เท่ากับ 5825 จากลำดับงาน

70 58 61 87 39 55 89 4 21 60 6 67 24 1 92 5 80 35 83 31 85 10 96 64 51
46 56 22 84 57 17 13 88 43 20 68 25 90 38 44 14 2 8 9 27 18 15 62 29 97
75 49 76 78 34 63 79 33 54 93 73 7 50 77 74 23 66 42 45 98 41 52 28 82
81 47 100 86 36 11 19 32 26 69 59 48 94 3 16 95 72 99 40 53 37 65 71 91
30 12

ปัญหา 100×20 เมื่อแก้ปัญหาค้างโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีหาบูที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 6315.3 วินาที (4 Processors) ได้ Makespan เท่ากับ 6352 จากลำดับงาน

54 78 1 59 82 74 21 47 3 51 40 18 33 81 48 22 31 89 76 39 5 60 93 12 28
10 62 85 24 35 53 20 61 23 57 25 70 4 83 34 71 65 43 32 6 29 63 8 72 80
50 44 79 56 38 30 37 94 99 77 16 14 92 42 36 15 91 2 17 45 55 88 98 96
58 27 49 46 73 11 100 90 9 26 41 97 95 87 68 86 69 19 67 66 84 75 64 52
13 7

ปัญหา 200×10 เมื่อแก้ปัญหาค้างโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีหาบูที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 13200.2 วินาที (4 Processors) ได้ Makespan เท่ากับ 10983 จากลำดับงาน

66 23 87 74 132 124 131 4 190 195 44 191 193 89 134 146 123 164 45 11
153 90 148 188 177 46 22 197 181 94 31 60 85 175 182 10 7 41 71 141 155
48 156 159 96 36 172 70 8 103 154 81 77 166 47 151 21 107 25 105 28 184
1 120 170 2 32 13 35 130 86 144 142 180 121 24 139 194 189 95 34 68 116
14 152 161 115 129 17 110 128 69 111 119 82 99 18 163 6 20 136 53 102
135 50 117 150 196 37 80 76 169 59 5 109 118 122 198 145 56 75 143 33
186 127 97 12 104 84 173 61 3 16 185 51 65 165 42 187 192 93 171 54 149
98 108 160 183 38 140 200 147 199 52 55 15 101 58 100 83 112 49 125 106
63 62 92 19 137 174 78 9 126 30 57 114 73 64 178 29 67 43 113 40 162
176 179 91 133 79 138 26 39 88 167 27 158 72 157 168

ปัญหา 200×20 เมื่อแก้ปัญหาค้างโปรแกรมจัดการการผลิตแบบต่อเนื่อง โดยวิธีหาบูที่ทำงานแบบขนาน ได้ผลดังนี้

ประมวลผล 20000 รอบ ใช้เวลา 28952.5 วินาที (4 Processors) ได้ Makespan เท่ากับ
11438 จากลำดับงาน

79 76 57 182 83 158 77 89 94 150 151 167 33 198 155 13 63 188 133 110
101 172 105 54 7 22 31 5 126 144 146 178 4 187 141 195 123 85 148 88
147 106 50 36 81 48 51 70 108 104 154 149 177 166 175 66 67 107 25 26
139 72 168 120 64 136 91 170 98 35 86 181 142 39 121 2 40 156 189 95 34
90 111 132 152 44 71 135 164 75 128 69 53 119 52 113 18 163 6 196 24
116 102 28 61 117 130 10 37 80 115 169 59 8 138 118 122 131 145 56 41
143 161 186 60 97 12 14 127 173 157 162 129 17 3 65 47 42 87 96 125 9
190 194 197 103 191 74 32 140 55 199 62 174 200 82 112 58 153 171 45
192 93 23 73 11 124 27 78 99 137 46 43 92 193 1 16 84 19 29 179 30 49
134 176 184 38 183 185 180 109 21 160 15 20 159 100 68 114 165

ภาคผนวก ค

โปรแกรม

```

#include <stdio.h>
#include <fstream.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#include "mpi.h"

int n,m,inf=300000,v1,v2,*up,tot,aseed,**tabu,prob,bb[150],bc[150],cbb;
int besttot,bestseed,bestrank,*bestsol,rank,bestiter,balliter,bltabu;
double timebest;

int **q,ltabu,*memtb1,*memtb2,tc,**fin,**st,nump,h,l,tct,loop,loopchk,bestloop=0;
long iter;
void swapp(int&,int&);
void cal(int*);
void initial(int*);
void modify(int*);

```

```

void updatetabu();
void srand(unsigned);
int random(int _num);
clock_t t1, t2;
unsigned int seed;
int main(int argc, char *argv[])
{
    int i,*p,j,k;
    long maxiter;
    double start, end;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&num);
    MPI_Status status;
    if(rank==0)
        {cout<<"Enter number of iteration "<<endl;
        cin>>maxiter;
        cout<<"Enter number of tabusize "<<endl;
        cin>>ltabu;
        cout<<"Enter number of seed"<<endl;
        cin>>seed;
        ifstream yy("shop1.txt");
        yy>>prob;
        yy>>n;
        yy>>m;
        tabu=new int*[n+1];
        p=new int[n+1];
        up=new int[n+1];
        bestsol=new int[n+1];
        q=new int*[n+1];

```

```

fin=new int*[m+1];
st=new int*[m+1];
for(j=1;j<=m;j++)
{
    fin[j]=new int[n+2];
    st[j]=new int[n+2];
}
for(i=0;i<=n;i++)
{ tabu[i]=new int[n+1];
  q[i]=new int[m+1];
}
for(i=1;i<=n;i++)
{ for(j=1;j<=m;j++)
  { yy>>k;
    yy>>q[i][j];
  }
}
memb1=new int[ltabu+1];
memb2=new int[ltabu+1];
yy.close();
}

MPI_Barrier(MPI_COMM_WORLD);

MPI_Bcast(&maxiter, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&ltabu, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&seed, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&prob, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&m, 1, MPI_INT, 0, MPI_COMM_WORLD);

MPI_Barrier(MPI_COMM_WORLD);

if(rank!=0)

```

```

{
    tabu=new int*[n+1];
    p=new int[n+1];
    up=new int[n+1];
    bestsol=new int[n+1];
    q=new int*[n+1];
    fin=new int*[m+1];
    st=new int*[m+1];
    for(j=1;j<=m;j++)
    {
        fin[j]=new int[n+2];
        st[j]=new int[n+2];

    }
    for(i=0;i<=n;i++)
    {
        tabu[i]=new int[n+1];
        q[i]=new int[m+1];
    }
    memtb1=new int[ltabu+1];
    memtb2=new int[ltabu+1];
}

MPI_Barrier(MPI_COMM_WORLD);
for(i=1;i<=n;i++)
    MPI_Bcast(&(q[i][1]), m, MPI_INT, 0, MPI_COMM_WORLD);

cout<<"Com "<<rank<<"maxiter ="<<maxiter<<" tabu size = "<<ltabu<<"seed = "<<seed;
cout<<" (Prob Ta"<<prob<<","<<n<<" job "<<m<<" machines)"<<endl;
MPI_Barrier(MPI_COMM_WORLD);
cout<<"Com "<<rank<<" Load Data complete "<<endl;
for(i=1;i<=150;i++)
    {
        bb[i-1]=0;
    }

```

```

    bc[i-1]=0;
}

    cbb=0;

    t1=clock();

    start = MPI_Wtime();

    tot=inf;

srand(seed);

initial(p);

bb[cbb]=0;

bc[cbb]=tc;

cbb=cbb+1;

iter=0;

do{ iter++;

    loopchk=0;

    modify(p);

MPI_Barrier(MPI_COMM_WORLD);

MPI_Allreduce(&tct,&besttot,1,MPI_INT,MPI_MIN,MPI_COMM_WORLD);

if(tct!=besttot)

    loopchk=500000;

MPI_Barrier(MPI_COMM_WORLD);

MPI_Allreduce(&loopchk,&bestloop,1,MPI_INT,MPI_MIN,MPI_COMM_WORLD);

if (loopchk==bestloop)

    {

        MPI_Send(&h,1,MPI_INT,0,0,MPI_COMM_WORLD);

        MPI_Send(&l,1,MPI_INT,0,1,MPI_COMM_WORLD);

        MPI_Send(&v1,1,MPI_INT,0,2,MPI_COMM_WORLD);

        MPI_Send(&v2,1,MPI_INT,0,3,MPI_COMM_WORLD);

        MPI_Send(&tct,1,MPI_INT,0,4,MPI_COMM_WORLD);

    }

MPI_Barrier(MPI_COMM_WORLD);

```

```

if (rank==0)
{
    MPI_Recv(&h,1,MPI_INT,MPI_ANY_SOURCE,0,MPI_COMM_WORLD,&status);
    MPI_Recv(&l,1,MPI_INT,MPI_ANY_SOURCE,1,MPI_COMM_WORLD,&status);
    MPI_Recv(&v1,1,MPI_INT,MPI_ANY_SOURCE,2,MPI_COMM_WORLD,&status);
    MPI_Recv(&v2,1,MPI_INT,MPI_ANY_SOURCE,3,MPI_COMM_WORLD,&status);
    MPI_Recv(&tct,1,MPI_INT,MPI_ANY_SOURCE,4,MPI_COMM_WORLD,&status);
}

MPI_Barrier(MPI_COMM_WORLD);

    MPI_Bcast(&h, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&l, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&v1, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&v2, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&tct, 1, MPI_INT, 0, MPI_COMM_WORLD);

MPI_Barrier(MPI_COMM_WORLD);

    swapp(p[l],p[h+1]);

    cal(p);

if (tc<tot)
{tot=tc;
for(i=1;i<=n;i++)
up[i]=p[i];
aseed=seed;
balliter=iter;
}

    if (iter==1)

        cout<<"iter= "<<iter<<" ("<<tc<<") "<<" Makespan = "<<tot<<endl;

        if ((iter%100)==0)

            cout<<"iter= "<<iter<<" ("<<tc<<") "<<" Makespan = "<<tot<<endl;

if(cbb<=150)

if(bc[cbb-1]!=tot)

```

```

{
    bb[cbb]=iter;
    bc[cbb]=tot;
    cbb++;
}
updatetabu();
MPI_Barrier(MPI_COMM_WORLD);
}while(iter<maxiter);
end = MPI_Wtime();
t2=clock();
if (rank==0){
ofstream hut("outsolution.txt");
    hut<<"Problem ta "<<prob<<" "<<num<<" Processor "<<endl;
    for(i=1;i<=n;i++)
    { hut<<up[i]<<" ";
    if ((i%10)==0) hut<<endl;
    }
    hut<<"Iterations = "<<maxiter<<endl;
    hut<<"Makespan          = "<<tot<<" [iter= "<<balliter<<" ]"<<endl;
    hut<<"seed            ="<<aseed<<endl;
    hut<<"tabu list size ="<<ltabu<<endl;
    hut<<"The Overall time was: "<<((t2 - t1)/1000.00 )<<endl;
    hut<<"\n result of tabu search each iteration \n"<<endl;
    for(i=1;i<=150;i++)
    if(bc[i-1]!=0)
    {
        hut<<i<<"\t"<<bb[i-1]<<"\t"<<bc[i-1]<<endl;
    }
    hut.close();
}

```

```

    MPI_Finalize();

    return 0;
}

void swapp(int &x,int &y)
{
    int z;

    z=x;

    x=y; y=z;
}

void initial(int *p)
{
    int *v,maxv,*vv,sv,i,j;

    v=new int[n+1];
    vv=new int[n+1];

    for(i=1;i<=n;i++)

        v[i]=random(2*n);

    for(i=1;i<=n;i++)

        {maxv=-1;

        for(j=1;j<=n;j++)

            {if ((v[j]>maxv)&&(vv[j]!=1)) {sv=j;maxv=v[j];}}

        }

    p[i]=sv; vv[sv]=1;
}

cal(p);

if (tc<tot)

    {tot=tc;

    for(i=1;i<=n;i++)

        up[i]=p[i];

    aseed=seed;

    }

for(i=1;i<=ltabu;i++)

    {membt1[i]=0; membt2[i]=0;}

```

```

for(i=0;i<=n;i++)
for(j=0;j<=n;j++)
    tabu[i][j]=0;
}

void modify(int *p)
{ int i,b,tmax=inf,numk=rank,numj=0;

  loop=0;
  for(b=1;b<=n-2;b++)
  for(i=b;i<=n-1;i++)
  { loop++;
    if(numj==rank)
      if (tabu[p[b]][p[i+1]]==0)
      {swapp(p[b],p[i+1]);
      cal(p);
      if (tc<tmax)
      {h=i;
      l=b;
      tmax=tc;
      tct=tc;
      loopchk=loop;
      v1=p[b];
      v2=p[i+1];
      }
      swapp(p[i+1],p[b]);
      }

      numj++;
      if(numj==numk)numj=0;
    }
  }

void updatetabu()

```

```

{int i;

    tabu[v1][v2]=1; tabu[v2][v1]=1;

    tabu[memtb1[1]][memtb2[1]]=0; tabu[memtb2[1]][memtb1[1]]=0;

    for(i=1;i<=ltabu-1;i++)

        {memtb1[i]=memtb1[i+1];

          memtb2[i]=memtb2[i+1];

        }

    memtb1[ltabu]=v1;

    memtb2[ltabu]=v2;

}

void cal(int *p)

{ int i,j,t;

    for(j=1;j<=m;j++)

        fin[j][n+1]=0;

    for(t=n;t>=1;t--)

        {

            for(j=1;j<=m;j++)

                {

                    if (j>1)

                        {

                            if (fin[j][t+1] > fin[j-1][t])

                                st[j][t]=fin[j][t+1];

                            else st[j][t]=fin[j-1][t];

                        }else st[1][t]=fin[1][t+1];

                    fin[j][t]=st[j][t]+q[p[n-t+1]][j];

                }

            }

        tc=fin[m][1];

}

int random(int __num)

```

```
{ return(int)(((long)rand()*__num)/(RAND_MAX+1)); }
```