บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงวรรณกรรม ซึ่งมีรายละเอียดที่เกี่ยวข้องดังต่อไปนี้

1 **ปัญหาการจัดตารางการผลิต**

1.1 ความหมายและความสำคัญของปัญหาการจัดตารางการผลิต

1.2 การกำหนดขอบเขตของปัญหาการจัดตารางการผลิต

1.3 การกำหนดข้อบังคับของการจัดตารางการผลิต

1.4 การประเมินค่าการจัดตารางการผลิต

1.5 วิธีการที่นำมาใช้แก้ปัญหาการจัดตารางการผลิต

2. **วิธีการหาค่าคำตอบที่ดีที่สุด** (Optimization algorithms)

2.1 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ (Conventional optimization algorithms: COAs)

2.2 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักของการประมาณ (Approximation optimization algorithms: AOAs)

3. **การออกแบบการทดลองและการวิเคราะห์ข้อมูลเชิงสถิติ**

3.1 การออกแบบการทดลอง (Design of the experiment)

3.2 การวิเคราะห์ข้อมูลเชิงสถิติ (Statistical analysis of the data)

1. ปัญหาการจัดตารางการผลิต

1.1 ความหมายและความสำคัญของปัญหาการจัดตารางการผลิต

การจัดตาราง (Scheduling) หมายถึง การจัดสรรทรัพยากร (Resource) ที่มีอยู่อย่างจำกัดให้กับงาน (Task) จำนวนหนึ่งภายใต้ระยะเวลาที่กำหนดให้ เพื่อที่จะทำให้องค์กรสามารถบรรลุถึงเป้าหมาย (Goal) หรือ วัตถุประสงค์ (Objective) สูงสุดที่องค์กรกำหนดเอาไว้ที่เวลานั้นได้ (ปารเมศ ชุติมา, 2546)

การจัดลำดับงานการผลิต (Production sequencing) หมายถึง การจัดเครื่องจักรที่จะนำไปผลิตสินค้าและบริการ รวมทั้งจัดสรรกำลังการผลิตเพื่อให้กระบวนการการผลิตเป็นไป

อย่างต่อเนื่อง ทั้งนี้เพื่อให้เกิดผลผลิตจำนวนมากที่สุด และสิ้นเปลืองน้อยที่สุด (วิชัย แหวนเพชร, 2543)

การจัดตาราง (Scheduling) เป็นกระบวนการตัดสินใจอย่างหนึ่งที่มีความสำคัญ เนื่องจากเป็นผลลัพธ์ของกระบวนการตัดสินใจ ซึ่งเป็นกำหนดการ (Schedule) ของกิจกรรมต่าง ๆ ที่ต้องปฏิบัติ เพื่อเพิ่มประสิทธิภาพในการทำงานให้สูงขึ้น เช่น การผลิต การขนส่ง การจัดซื้อ ซึ่งองค์กรจำนวนมากได้นำเอาทฤษฎีการจัดตารางมาประยุกต์ใช้จัดการกับกิจกรรมต่าง ๆ แบ่งประเภทของปัญหาการจัดตารางตามคุณลักษณะหลัก 7 ประการดังนี้

1. การจัดตารางที่ขึ้นกับลักษณะคำสั่งซื้อจากลูกค้า แบ่งตามระบบการผลิต ออกเป็น 2 แบบคือ

1.1 ระบบเปิด (Open shop) คำสั่งซื้อทั้งหมดจะถูกกำหนด จากการสั่งซื้อโดยตรงจากลูกค้า ไม่มีการเก็บสต็อกสินค้า

1.2 ระบบปิด (Closed shop) การผลิตจะเกิดขึ้น เพื่อผลิตสินค้าป้อนสู่คลังสินค้า ซึ่งจะมีปัญหาของการจัดลำดับงานเข้ามาเกี่ยวข้อง

2. การจัดตารางที่ขึ้นจากความซับซ้อนของกระบวนการผลิตมาเป็นตัวกำหนดตารางการผลิต โดยเกี่ยวข้องกับจำนวนของขั้นตอนในการปฏิบัติงานของแต่ละงาน เป็น 4 ประเภท คือ

2.1 ขั้นงานเดียว มีหนึ่งเครื่องจักร

2.2 ขั้นงานเดียว เป็นระบบเครื่องจักรขนาน

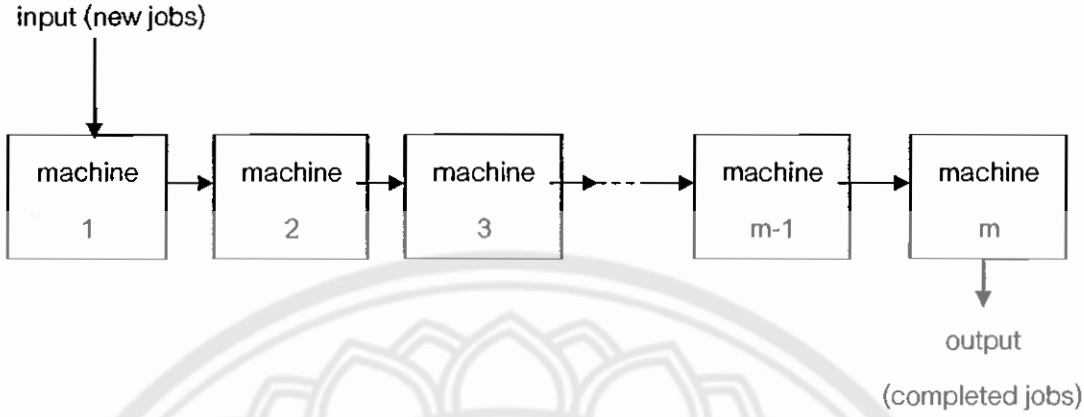
2.3 ขั้นหลายงาน ที่มีระบบการผลิตแบบไหลเลื่อน

2.4 ขั้นหลายงาน ที่มีระบบการผลิตแบบตามงาน

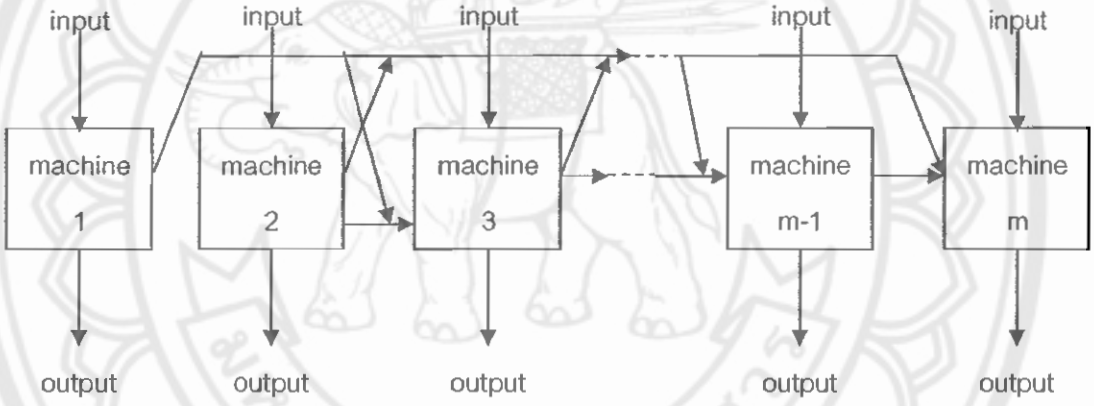
ประเภทการจัดตารางการผลิตสามารถแบ่งตามลักษณะการผลิต ดังนี้

1. การจัดตารางการผลิตของการผลิตแบบไหลเลื่อน (Flow shop scheduling)

ลักษณะการผลิตแบบการไหลของสายงาน ประกอบด้วยเครื่องจักรหรือสถานีงานหลายสถานีงานที่ทำงานต่อเนื่องกันโดยลำดับขั้นตอนการทำงานของทุกงานเหมือนกัน ปัญหาการจัดตารางการผลิตแบบการไหลของสายงานประกอบด้วยเครื่องจักรที่ต่างกัน m เครื่อง และงานแต่ละงานประกอบด้วยจำนวนขั้นตอนการทำงาน m ขั้นตอน (Operation) โดยในแต่ละขั้นตอนการทำงานใช้เครื่องจักรที่ต่างกัน (Baker, 1974)



ภาพ 1 แสดงการผลิตแบบ Pure flow shop (ปารเมศ ชุติมา, 2546)

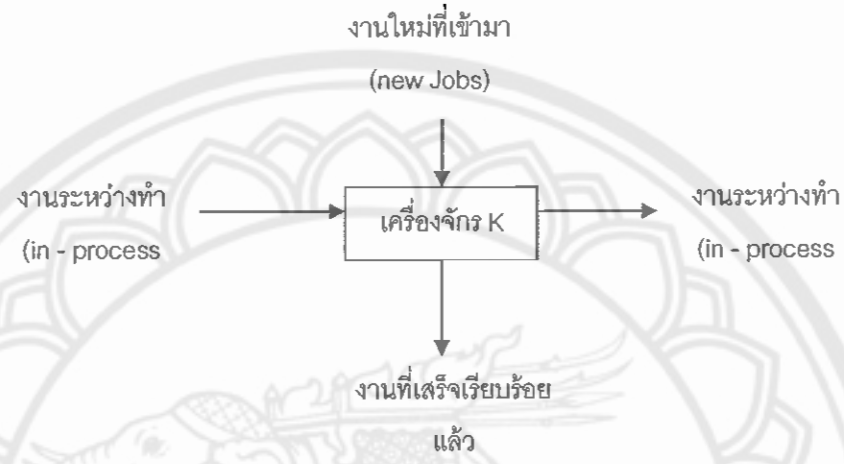


ภาพ 2 แสดงการผลิตแบบ General flow shop (ปารเมศ ชุติมา, 2546)

2. การจัดตารางการผลิตของการผลิตแบบสั่งผลิตเป็นงาน ๆ (Job shop scheduling)

ปัญหาการจัดตารางการผลิตแบบสั่งผลิตเป็นงาน ๆ มีลักษณะแตกต่างจากปัญหาการจัดตารางการผลิตแบบการไหลของสายงานคือ เส้นทางการไหลของงานมีความแตกต่างกันไปตามชนิดของงาน ปัญหาการจัดตารางการผลิตแบบสั่งผลิตเป็นงาน ๆ ประกอบไปด้วยเครื่องจักรจำนวนหนึ่งและงานหลาย ๆ ประเภท โดยงานแต่ละงานประกอบไปด้วยขั้นตอนการทำงานหลาย ๆ ขั้นตอน ซึ่งมีลำดับก่อน และลำดับหลังในการผลิตที่แน่นอน (Baker, 1974) การจัดตารางการผลิตแบบสั่งเป็นงาน ๆ ประกอบด้วยชุดเครื่องจักรที่แตกต่างกัน เช่น เครื่องกลึง เครื่องเจาะ เป็นต้น โดยชิ้นงานแต่ละประเภทมีเส้นทางการไหลหรือขั้นตอนการทำงานที่แน่นอน

แต่มีความแตกต่างไปตามประเภทของงาน ปัญหาการจัดตารางการผลิตแบบสั่งผลิตเป็นงาน ๆ เป็นปัญหาในลักษณะ NP-hard และเป็นปัญหาที่จัดการได้ยากมากที่สุดปัญหาหนึ่ง (Lawler, et al. 1993)



ภาพ 3 แสดงการผลิตแบบ General job shop (ปารเมศ ชุติมา, 2546)

3. การจัดตารางโดยใช้เกณฑ์ในการประเมินผลมากำหนดวิธีการจัดตารางเป็นแบบต่าง ๆ เป็น 2 ประเภทคือ

3.1 แบ่งตามค่าใช้จ่ายที่เกิดขึ้น เช่น ค่าใช้จ่ายในการปรับตั้งเครื่องจักร ค่าใช้จ่ายด้านการเก็บพัสดุคงคลัง

3.2 แบ่งตามสมรรถนะของตาราง เช่น ค่าเฉลี่ยของเวลาไหลของงาน ค่าเฉลี่ยของเวลาล่าช้าของงาน

4. การจัดตารางซึ่งอาศัยธรรมชาติของงาน มาเป็นตัวกำหนดตารางงาน โดยพิจารณาจากรายละเอียดของงานที่เกี่ยวข้อง แบ่งออกเป็น 2 แบบ คือ

4.1 ปัญหาเชิงกำหนด (Deterministic problem)

4.2 ปัญหาเชิงเฟ้นสุ่ม (Stochastic problem)

5. การจัดตารางงานตามสภาพแวดล้อม เป็นการพิจารณาสภาพแวดล้อมต่าง ๆ มาช่วยจัดลำดับในการจัดตารางการทำงาน แบ่งออกเป็น 2 แบบ คือ

5.1 ปัญหาเชิงสถิต (Static problem)

5.2 ปัญหาเชิงพลวัต (Dynamic problem)

6. แบ่งตามทิศทางของการจัดตาราง เป็นการพิจารณาการจัดลำดับของการจัดตารางจากซ้ายไปขวา แบ่งออกเป็น 2 แบบ คือ

6.1 จัดแบบไปข้างหน้า

6.2 จัดแบบถอยหลัง

7. แบ่งตามข้อจำกัดในการใช้เครื่องจักร เป็นการพิจารณาข้อจำกัดในการใช้เครื่องจักร ที่มาเป็นเงื่อนไขของการจัดตาราง แบ่งออกเป็น 2 แบบ คือ

7.1 จัดแบบจำกัดจำนวนเครื่องจักร

7.2 จัดแบบไม่จำกัดจำนวนเครื่องจักร

เทคนิคที่ใช้ในการจัดตารางการผลิต ในงานวิจัยนี้

แบ่งออกเป็น 2 เทคนิค คือ

1. การจัดตารางแบบไปข้างหน้า (Forward scheduling)

เริ่มจัดตารางเมื่อทราบถึงความต้องการต่างๆ เกี่ยวกับงาน การวางงานสู่ตำแหน่งต่างๆ บนตารางจะมีทิศทางจากซ้ายไปขวาเมื่อดูจากแผนภูมิแกนต์ เทคนิคการจัดตารางแบบนี้ถูกนำไปใช้อย่างแพร่หลายในหลายหน่วยงาน เช่น โรงพยาบาล คลินิก ภัตตาคาร หรือโรงกลึง เป็นต้น งานที่หน่วยงานประเภทนี้รับเข้ามาทำจะเป็นประเภทที่ทำตามคำสั่งซื้อของลูกค้า ซึ่งลูกค้าส่วนมากอยากจะได้สินค้าหรือต้องการรับบริการอย่างเร็วที่สุดเท่าที่หน่วยงานนั้น จะทำได้ หมายความว่า มีการกำหนดเวลาส่งมอบให้กับแต่ละงานนั่นเอง ถึงแม้ว่าตารางที่สร้างขึ้นจากเทคนิคการจัดตารางแบบไปข้างหน้านี้จะไม่ละเมิดข้อจำกัดด้านการผลิต แต่ก็อาจจะทำให้เกิดการส่งมอบงานล่าช้าได้ และนอกจากนั้นแล้วยังอาจจะทำให้เกิดชิ้นงานระหว่างทำขึ้นเป็นจำนวนมากในระบบอีกด้วย

2. การจัดแบบถอยหลัง (Backward scheduling)

เริ่มต้นจากเวลาส่งมอบ และจัดตารางให้กับการดำเนินงานสุดท้ายก่อนเป็นอันดับแรก ขั้นตอนการดำเนินงานอื่น ๆ จะถูกจัดตารางทีละขั้นตอนตามลำดับที่ย้อนกลับ **การวางงานลงสู่ตำแหน่งต่าง ๆ บนตารางจะมีทิศทางจากขวาไปซ้ายเมื่อดูจากแผนภูมิแกนต์ เมื่อลบผลรวมของเวลานำอันเกิดจากแต่ละการดำเนินงานที่ประกอบกันขึ้นเป็นงานหนึ่งงานออกจะทำให้ทราบถึงการเริ่มต้นของงาน ๆ นั้นได้ ถึงแม้ว่าตารางที่สร้างขึ้นจากเทคนิคการจัดตารางแบบถอยหลังนี้จะไม่ทำให้เกิดงานล่าช้าขึ้นก็ตาม แต่อาจจะไม่สามารถหาตารางที่เป็นจริงก็ได้ เนื่องจากตารางดังกล่าวมีการละเมิดข้อจำกัดด้านกำลังการผลิตขึ้น (เมื่อมีงานเริ่มต้นที่เวลา $t < 0$)**

การจัดตารางแบบถอยหลังนี้ใช้มากในทั้งโรงงาน และในงานที่เกี่ยวกับบริการ เช่น การจัดเลี้ยง รับรอง การจัดตารางงานผ่าตัด เป็นต้น (ปารเมศ ชุติมา, 2546.หน้า 18)

ซึ่งเทคนิคการจัดตารางทั้ง 2 แบบมีความแตกต่างกันคือ การจัดตารางแบบ ไปข้างหน้า (Forward scheduling) จะเริ่มต้นจากซ้ายไปขวา ณ เวลาเริ่มต้น หรือเริ่มดำเนินการ จัดตารางทันที เริ่มทำงานทันทีเมื่อเครื่องจักรดำเนินการไปจนผลิตเสร็จ โดยไม่พิจารณาวันที่ส่งมอบงานให้กับลูกค้า ทำให้ในบางครั้งเกิดกรณีที่งานผลิตเสร็จก่อนกำหนดเร็วมาก ทำให้เกิดค่าใช้จ่ายในการเก็บรักษาช่วงก่อนการส่งมอบ ในบางครั้งเกิดกรณีที่งานผลิตเสร็จล่าช้ากำหนดมาก ทำให้ไม่สามารถส่งมอบสินค้าให้ลูกค้าได้ จนเกิดค่าปรับที่เกิดจากงานเสร็จล่าช้าเกิดขึ้น ทั้งสองกรณีก่อให้เกิดค่าใช้จ่ายทั้งสิ้น ดังนั้นจึงต้องพยายามควบคุมโดยการนำเทคนิคการจัดตารางแบบถอยหลัง (Backward scheduling) ซึ่งเป็นการจัดตารางการผลิตโดยพิจารณากำหนดการ ส่งมอบมากำหนดลำดับของการจัดตารางการผลิต เพื่อจะได้ผลิตงานให้เสร็จพอดี ๆ กับวันส่งมอบงานมากที่สุด เพื่อให้เกิดค่าปรับที่เกิดจากงานเสร็จก่อน หรือเสร็จช้ากว่ากำหนด ให้ต่ำที่สุดเท่าที่จะทำได้ ทั้งนี้ต้องอยู่บนพื้นฐานของการมีทรัพยากรที่สามารถทำได้ ในช่วงเวลา ที่กำหนดด้วย ซึ่งในทางปฏิบัติแล้ว เราต้องใช้ทั้งการจัดตารางแบบไปข้างหน้า และแบบถอยหลังร่วมกัน เพื่อให้เกิดดุลยภาพที่เหมาะสมระหว่างตารางการผลิตที่เป็นจริงได้

จากปัญหาที่มีขนาดต่างแตกกัน เมื่อปัญหามีขนาดใหญ่ขึ้น เวลาที่ต้องใช้ในการหาคำตอบที่เหมาะสมที่สุด สำหรับขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนาม จะเพิ่มขึ้นน้อยกว่าประเภทที่มีความซับซ้อนด้านเวลาเชิงเลขยกกำลังอย่างมาก และถ้าความเร็วของคอมพิวเตอร์สูงขึ้น ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงเลขยกกำลังจะได้รับผลดีในเชิงบวก เมื่อปัญหาใหญ่ขึ้น ในขณะที่ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนามจะได้รับผลดี ในลักษณะทวีคูณ ดังนั้นความก้าวหน้าในด้านเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ จะมีประโยชน์ ต่อกระบวนการหาคำตอบที่ใช้ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนามมากกว่าเชิงเลข ยกกำลัง เราพบว่าขั้นตอนวิธีเสริมสร้างเป็นขั้นตอนที่มีความซับซ้อนด้านเวลาเชิงพหุนาม ในขณะที่การแตกกิ่งและจำกัดเขต (Branch and Bound) และ ไดนามิกโปรแกรมมิ่ง (Dynamic programming) เป็นขั้นตอนที่มีความซับซ้อนด้านเวลาเชิงเลขยกกำลังที่มีความซับซ้อนด้านเวลาเชิงพหุนาม เรียกว่า ปัญหาประเภทเอ็นพี (NP-Class) NP มาจาก Nondeterministic Polynamial ซึ่งปัญหาประเภทนี้จะมีความซับซ้อนด้านเวลาเชิงยกกำลัง ซึ่งมีแนวทางที่เกี่ยวกับปัญหาการหาคำตอบที่ดีที่สุดเชิงการจัด (Combinatorial optimization) ส่วนมากจะอยู่ในกลุ่มของปัญหาประเภทเอ็นพี ซึ่งเป็นปัญหาที่ยากในการหาคำตอบ เช่นเดียวกับปัญหาการจัดตาราง ซึ่งเป็นที่

ยอมรับอย่างเป็นทางการว่าการแก้ปัญหาเหล่านี้ เพื่อให้ได้คำตอบที่ดีที่สุดนั้น จะต้องอาศัยขั้นตอนวิธีเชิงพหุนามเท่านั้น ซึ่งถูกจัดเป็นปัญหาเอ็นพี แบบยาก จะต้องใช้เวลาในการหาคำตอบนานมาก อาจเป็นปี ๆ ในการหาคำตอบที่ดีที่สุดได้

ดังนั้นจึงนำแนวทางฮิวริสติก (Heuristic) ถึงแม้ว่าแนวทางนี้จะไม่รับประกัน ว่าสุดท้ายแล้วเราจะได้คำตอบที่ดีสุดก็ตาม ซึ่งวิธีการหาคำตอบแบบฮิวริสติก (Heuristic) เป็นวิธีการที่สามารถหาคำตอบที่เหมาะสมที่สุดได้อย่างรวดเร็ว ถึงแม้ค่าของคำตอบที่ได้จากระบบซึ่งอาจไม่ใช้คำตอบที่ดีที่สุด แต่ก็เป็นค่าที่ยอมรับได้

การเลือกใช้เทคนิคใดนั้นขึ้นอยู่กับความซับซ้อนของปัญหา ธรรมชาติของแบบจำลอง และฟังก์ชันวัตถุประสงค์ และปัจจัยที่เกี่ยวข้องอื่น ๆ รวมทั้งความรู้ ประสบการณ์ในการเลือกใช้เทคนิคใดที่เหมาะสมมาใช้ในการจัดตารางการผลิต เพื่อให้ได้ตารางการผลิตที่มีฟังก์ชันวัตถุประสงค์ที่ดีที่สุด ใช้เวลาในการหาค่าคำตอบที่เหมาะสมที่สุด

1.2 การกำหนดขอบเขตของปัญหาการจัดตารางการผลิต

ก่อนจะเข้าถึงข้อมูลของการจัดตารางการผลิตในงานวิจัยนี้นั้น จำเป็นอย่างยิ่งที่จะต้องทำความเข้าใจลักษณะของปัญหาก่อน ว่าปัญหาที่นำมาทดสอบนี้มีความแตกต่างจากวิธีการจัดตารางแบบทั่ว ๆ ไป ทั้งในเรื่องของขนาดที่แตกต่างกันของปัญหาถึง 4 แบบ และมีลำดับของงานที่ยุ่งยากมาก เนื่องจากเป็นปัญหาที่มีลำดับขั้นตอนการผลิตต้องเป็นไปตามโครงสร้างของแต่ละผลิตภัณฑ์ ที่กำหนดเป็นเงื่อนไขในการจัดตารางการผลิต ทำให้การจัดตารางการผลิตมีความยาก และซับซ้อนมากขึ้น ดังนั้นจะเห็นได้ว่าข้อมูลของโรงงานอุตสาหกรรมที่ใช้ในการจัดการผลิต Pongcharoen (2001) ที่นำมาใช้เป็นปัญหาในการทดสอบ PSO จะมีข้อมูลของปัญหา 4 ขนาด ดังรายละเอียดแสดงในตารางที่ 1 โดยมีรายละเอียด ดังแสดงในตารางต่อไปนี้

ตาราง 1 แสดงรายละเอียดของปัญหาทั้ง 4 ขนาดที่ใช้ในการจัดตารางการผลิตด้วย PSO

ขนาดของปัญหา	หมายเลขผลิตภัณฑ์	รายละเอียดของปัญหาที่ใช้ในการจัดตารางการผลิต				
		จำนวนผลิตภัณฑ์	สาย การผลิต	จำนวนชิ้นงาน/จำนวนผลิตภัณฑ์	จำนวนเครื่องจักร	ระดับของโครงสร้าง
เล็ก	245 & 451	2	6	25/9	8	4
กลาง	229 & 451	2	8	57/10	7	4
ใหญ่	4 & 228	2	12	118/17	17	4
ใหญ่พิเศษ	227	1	46	229/39	25	6

จากตาราง 1 มีปัญหา 4 ขนาดคือ เล็ก กลาง ใหญ่ และใหญ่พิเศษ (ดูรายละเอียดเพิ่มเติมที่ภาคผนวก ก) ซึ่งนำมาใช้ทดสอบวิธีการจัดตารางการผลิตโดยใช้พาร์ทิเคิลสวอมออฟติไมเซชั่น (PSO) เมื่อพิจารณารายละเอียดปัญหาการผลิตที่จะนำมาใช้ในการจัดตารางแต่ละขนาดทั้ง 4 ขนาดแล้วผู้วิจัยพบว่าเป็นปัญหาที่ซับซ้อน น่าสนใจมาก ประกอบกับศึกษางานวิจัยที่ใช้วิธีการค้นหาแบบ Stochastic search ในการหาค่าเหมาะสมของค่าคำตอบ ในหลายงานวิจัย จนกระทั่งพบงานวิจัยของแอลเบลทากิ (Elbeltagi, Hegazy &Grierson. 2005) ที่ได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมซอง Evolutionary algorithms (EA) ซึ่งเป็นวิธีการค้นหาแบบ Stochastic search ผสมผสานกับ Natural biological evolution หรือ Social behavior มาพัฒนา เพื่อหาค่าคำตอบที่ใกล้เคียงของค่าคำตอบที่เหมาะสมที่สุด ซึ่งเป็นการนำสมการทางคณิตศาสตร์ 2 สมการ ซึ่งเป็นงานวิจัยที่ได้นำอัลกอริทึมทั้ง 5 แบบคือ Genetic Algorithms (GA), Memetic Algorithms (MA), Particles Swarm Optimization (PSO), Ant-Colony Systems (ACS), Shuffled Frog Leaping (SFL) มาทำการทดสอบโดยให้แต่ละอัลกอริทึมคำนวณค่าคำตอบ ที่เหมาะสมที่สุดแล้วนำมาเปรียบเทียบหาอัตราการหาค่าคำตอบที่เหมาะสมสูงสุด (Success rate) พบว่า PSO มีการพบค่าคำตอบที่เหมาะสมที่สุดสูงถึง 60 เปอร์เซ็นต์ (%) ในขณะที่ MA, ACO ได้ 20 เปอร์เซ็นต์ (%) และ GA, SFL มีการพบค่าคำตอบเพียงเล็กน้อยที่ 0 เปอร์เซ็นต์ (%)

ดังจะเห็นได้ว่า PSO เป็นอัลกอริทึมที่มีอัตราการหาค่าคำตอบ (Success rate) ที่เหมาะสมสูงสุด จากที่กล่าวมาแล้ว ทำให้ผู้วิจัยสนใจศึกษา PSO และนำมาประยุกต์ใช้เพื่อแก้ปัญหาการจัดตารางการผลิตในงานวิจัยนี้

1.3 การกำหนดข้อสมมติฐานของการจัดตารางการผลิต

ในการจัดตารางการผลิตนั้นเป็นปัญหาที่มีความเกี่ยวข้องกับปัจจัยต่าง ๆ เป็นจำนวนมาก เช่นลักษณะและข้อจำกัดของกระบวนการทำให้ระบบการผลิตอาจมีลักษณะเฉพาะตัว และข้อจำกัดที่ไม่เหมือนกับระบบอื่น เราสามารถแบ่งลักษณะและข้อจำกัดที่บ่งบอกถึงความแตกต่างของระบบการผลิต (ปารเมศ ชุติมา, 2546.)

ดังนั้นการศึกษาการจัดตารางการผลิตได้นั้น จะต้องมีการกำหนดข้อตกลงต่าง ๆ ที่จะนำมาใช้สำหรับพิจารณาการจัดตารางการผลิตดังต่อไปนี้ สมมติฐานของปัญหาการจัดตาราง (Assumption of scheduling problems) (Pongcharoen, 2001) สรุปได้ดังนี้

1. อัตราค่าปรับต่อวันเป็นค่าคงที่และไม่เท่ากัน สำหรับกรณีที่ผลิตภัณฑ์เสร็จเร็วหรือช้ากว่ากำหนด
2. เมื่อการทำงานก่อนหน้าสิ้นสุดลง การทำงานในลำดับต่อไปก็จะถูกปฏิบัติงานในทันที (Conway, Maxwell & Miller. 1967, Baker, 1974)
3. ณ ช่วงเวลาใด ๆ เครื่องจักร (Machine) แต่ละเครื่องสามารถทำงานได้ทีละงานเท่านั้น สมมติฐานนี้ประยุกต์ใช้ตามความสามารถในการผลิตที่จำกัด (Finite capacity) (Jain & Meeran, 1999)
4. ณ ช่วงเวลาใด ๆ งานแต่ละงานจะถูกปฏิบัติงานบนเครื่องจักรได้ทีละเครื่อง (Blazewicz et al., 1996)
5. ไม่มีการขัดจังหวะระหว่างการทำงาน (King & Spakis, 1980)
6. ไม่มีการแก้งาน (Ramasesh, 1990)
7. ไม่คำนึงถึงเวลาติดตั้ง (Setup) และเคลื่อนย้าย (Transfer) หรือถ้าคำนึงก็จะคิดเท่า ๆ กัน (Efstathiou, Calinescu & Bermejo. 1996)
8. ไม่มีงานประกอบชิ้นส่วน (No assembly operations) (Ramashes, 1990)

ลักษณะปัญหาของการจัดตาราง (Classification of scheduling problems) (Pongcharoen, 2001) ปัญหาการจัดตารางได้มีการศึกษากันมาอย่างแพร่หลายมามากกว่า 40 ปี ซึ่งงานวิจัยเกี่ยวกับการจัดตารางการผลิตก็เป็นงานวิจัยหนึ่งในอีกหลายงานวิจัยของปัญหาการจัดตารางที่มีการศึกษากันอย่างมาก Conway, Maxwell & Miller (1967) ได้นิยาม สัญลักษณ์ 4 ค่าที่

ใช้สำหรับแทนลักษณะของปัญหาการจัดตารางแต่ละปัญหา โดยใช้ A/B/C/D นอกจากนี้ยังมีบางงานวิจัย ใช้ n/m/F/C_{max} (Ogbu & Smith, 1990) คำอธิบายแต่ละค่ามีดังนี้

A คือ จำนวนทั้งหมดของงานที่ได้รับ คือเป็นทั้งจำนวนเฉพาะ (Specific number) หรือ n ที่ใช้แทนจำนวนที่กำหนดเอง (Arbitrary number)

B คือ ลักษณะเฉพาะของจำนวนเครื่องจักรในโรงงาน (Shop) ถ้าไม่เจาะจงแต่จำกัด จำนวนของเครื่องจักรแทนด้วย m ถ้าไม่เช่นนั้น กำหนดเครื่องจักรเป็นจำนวนเฉพาะ

C คือ อธิบายรูปแบบการไหลของงาน (Flow pattern) ในโรงงาน ได้แก่ F (Flow shop), R (Randomly routed job shop) หรือ G (General) ค่าเหล่านี้ถูกยกเว้นในกรณีของเครื่องจักรเดี่ยว (Single machine)

D คือ ใช้แทนตัววัดการทำงาน (Performance measure) ใช้สำหรับการประเมินค่าตาราง มี 3 ตัววัดที่ใช้กัน ได้แก่ Complete time (C) คือ เวลาเสร็จงาน, Total flow time (F) คือ เวลาไหลของงานเป็นระยะเวลาทั้งหมดที่งานใช้เวลาอยู่ในระบบ, Lateness (L) คือ เวลาสายของงาน (งานเสร็จหลังกำหนดเวลา)

1.4 การประเมินค่าการจัดตารางการผลิต

ทฤษฎีการจัดตารางเกี่ยวข้องกับการสร้างและพัฒนาแบบจำลองทางคณิตศาสตร์และหาเทคนิคที่เหมาะสมในการแก้ปัญหาการจัดตาราง โดยการนำปัญหาซึ่งเป็นโครงสร้างของปัญหาการจัดตารางการผลิต มาปรับเปลี่ยนเป็นสมการทางคณิตศาสตร์ที่เหมาะสม ซึ่งกระบวนการนี้เกี่ยวข้องกับการปรับเปลี่ยนลักษณะของเป้าหมาย ความต้องการ ให้สอดคล้องกับทรัพยากรในด้านต่าง ๆ ที่มีอยู่อย่างจำกัด เพื่อช่วยให้การตัดสินใจ เป็นไปตามฟังก์ชัน **วัตถุประสงค์** (Objective function) **และอยู่บนเงื่อนไข หรือข้อจำกัด** (Constraint) **ต่าง ๆ ดังรายละเอียด ในฟังก์ชันวัตถุประสงค์ ของการจัดตารางในงานวิจัยนี้นั้น ค่าใช้จ่าย** (Penalty cost) **ที่เกิดขึ้นในระบบการผลิตทั้งหมด ที่ได้รับผลกระทบโดยตรงต่อการตัดสินใจในการจัดตารางการผลิต ซึ่งประกอบด้วย**

สมการวัตถุประสงค์

$$\text{Total penalty cost} = \sum PE(E_{jk} + E_k) + \sum PT(T_k) \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

กำหนดให้ i = จำนวนของงาน (Operations) $(i = 1, \ldots, o)$

j = จำนวนของชิ้นส่วน (Components) $(j = 1, \ldots, c)$

k = จำนวนของผลิตภัณฑ์ (Products) $(p = 1, \ldots, p)$

m = จำนวนของเครื่องจักร (Machines) $(m = 1, \ldots, M)$

C_{max} = เวลาปิดงานของการผลิต (Makespan)

C_k = เวลาเสร็จงานของผลิตภัณฑ์ที่ k (Product k^{th})

D_k = กำหนดส่งมอบ (Due date) ของผลิตภัณฑ์ที่ k (Product k^{th})

C_{jk} = เวลาเสร็จงานของงานสุดท้ายของชิ้นงาน j (Component j^{th})

D_{jk} = กำหนดส่งมอบของงานสุดท้ายของชิ้นงาน j

E_k = เวลาเสร็จงานก่อนกำหนด (Earliness) ของผลิตภัณฑ์ที่ k (Product k^{th})

E_{jk} = เวลาเสร็จงานก่อนกำหนดของผลิตภัณฑ์ที่ j (Product j^{th})

T_k = เวลาเสร็จงานล่าช้ากว่ากำหนด (Tardiness) ของผลิตภัณฑ์ที่ k (Product k^{th})

SU_{ijkm} = เวลาติดตั้ง (Set-up time) ของชิ้นส่วนที่ c (Operation c^{th}) ของงานที่ j (Component j^{th}) ของผลิตภัณฑ์ที่ k (Product k^{th}) บนเครื่องจักร m (Machine m^{th})

S_{ijkm} = เวลาเริ่มทำงาน (Start time) ของชิ้นส่วนที่ c ของงานที่ j ของผลิตภัณฑ์ที่ k บนเครื่องจักรที่ m

P_{ijkm} = เวลาของกระบวนการทำงาน (Processing time) ของชิ้นส่วนที่ c ของงานที่ j ของผลิตภัณฑ์ที่ k บนเครื่องจักรที่ m

F_{ijkm} = เวลาสิ้นสุด (Finish time) ของชิ้นส่วนที่ c ของงานที่ j ของผลิตภัณฑ์ที่ k บนเครื่องจักรที่ m

TR_{ijkm} = เวลาเปลี่ยนเครื่อง (Transfer time) ของชิ้นส่วนที่ c ของงานที่ j ของผลิตภัณฑ์ที่ k บนเครื่องจักรที่ m

PE = ค่าปรับ (Penalty) ของการทำงานเสร็จก่อนกำหนด

PT = ค่าปรับของการทำงานเสร็จล่าช้ากว่ากำหนด (Tardiness)

A(x) = เซตของชิ้นส่วนย่อย (Child component) ของผลิตภัณฑ์ย่อย (Part) x

Sh = กะการทำงาน (Shift pattern) — 8, กรณี 3 กะต่อวัน; 16, กรณี 2 กะต่อวัน; 24, กรณี 1 กะต่อวัน

สมการเงื่อนไข

$$F_{ijkm} = S_{ijkm} + P_{ijkm} + TR_{ijkm} + SU_{ijkm} \quad \forall i, j, k, m \quad (2)$$

$$C_{jk} = F_{ojkL} \quad \forall m \quad (3)$$

$$E_{jk} = D_{jk} - \text{round}(C_{jk}/sh) \quad \forall j, k \quad (4)$$

$$E_{k} = D_{k} - \text{round}(C_{k}/sh) \quad \forall k \quad (5)$$

$$L_{k} = \text{round}(C_{k}/sh) - D_{k} \quad \forall k \quad (6)$$

$$C_{k} \leq C_{max} \quad \forall k \quad (7)$$

$$SU_{ijkm}, TR_{ijkm}, S_{ijkm}, P_{ijkm}, F_{ijkm} \geq 0 \quad \forall i, j, k, m \quad (8)$$

$$S_{xjkm} \geq F_{yjkm} \quad \forall j, k, m, i = n+1 \quad (9)$$

$$S_{ixkm} = \max F_{iykm} \quad \forall i, j, k, m, y \in A(x) \quad (10)$$

$$C_{k}, E_{k}, T_{k} > 0 \quad \forall k \quad (11)$$

$$C_{jk}, E_{jk} > 0 \quad \forall j, k \quad (12)$$

สมการที่ (1) เป็นสมการเป้าหมายของงานวิจัย เพื่อหาผลรวมของค่าปรับจากการทำงานเสร็จก่อนกำหนดของชิ้นส่วนและผลิตภัณฑ์สุดท้ายกับค่าปรับที่เกิดจากทำงานเสร็จล่าช้ากว่ากำหนดของผลิตภัณฑ์สุดท้ายที่น้อยที่สุด สำหรับค่าปรับที่ดีที่สุดคือ ศูนย์ (ไม่มีค่าปรับ)

สมการที่ (2) เวลาสิ้นสุดของงานเป็นผลรวมของเวลาเริ่มงาน เวลาที่ใช้ในกระบวนการทำงาน เวลาเปลี่ยนเครื่อง และเวลาติดตั้งเครื่องจักร ของทุก ๆ งาน, ชิ้นส่วน, ผลิตภัณฑ์ และเครื่องจักร

สมการที่ (3) เวลาเสร็จงานสุดท้ายของงาน j คือเวลาสิ้นสุดของงานของทุกเครื่องจักร m

สมการที่ (4) เป็นการคำนวณค่าของเวลาที่ทำงานเสร็จก่อนกำหนดของชิ้นส่วน (Part)

สมการที่ (5) เป็นการคำนวณค่าของเวลาที่ทำงานเสร็จก่อนกำหนดของผลิตภัณฑ์ (Product)

สมการที่ (6) เป็นการคำนวณเวลาของการทำงานเสร็จล่าช้ากว่ากำหนดของผลิตภัณฑ์

สมการที่ (7) กำหนดให้เวลาเสร็จงานของผลิตภัณฑ์ มีค่าน้อยกว่าหรือเท่ากับเวลาปิดงานของระบบการผลิต

สมการที่ (8) กำหนดให้ค่าของเวลาติดตั้ง เวลาเปลี่ยนเครื่อง เวลาเริ่มงาน เวลาที่ใช้ในกระบวนการทำงาน และเวลาสิ้นสุดการทำงาน ของทุกงาน ชิ้นส่วน ผลิตภัณฑ์ และเครื่องจักร มีค่ามากกว่าหรือเท่ากับศูนย์

สมการที่ (9) เป็นเวลาเริ่มการทำงานของงานลำดับต่อมามีค่ามากกว่าหรือเท่ากับเวลาเสร็จงานของงานลำดับก่อนหน้า ของทุกชิ้นส่วน ผลิตภัณฑ์ และเครื่องจักร ซึ่งเป็นการพิจารณาลำดับงานก่อนหลัง (Operation precedence) กำหนดให้ทำงานในลำดับต่อมาได้หลังจากที่งานก่อนหน้าทำเสร็จเรียบร้อยแล้ว

สมการที่ (10) เวลาเริ่มงานของชิ้นส่วนต่อมา จะเท่ากับเวลาสิ้นสุดของงานย่อยทั้งหมดของงานก่อนหน้า ของทุกงาน ชิ้นส่วน ผลิตภัณฑ์ และเครื่องจักร เป็นการพิจารณาลำดับก่อนหลังของชิ้นส่วนและผลิตภัณฑ์ในการประกอบ (Part precedence)

สมการที่ (11) กำหนดให้เวลาเสร็จงานของผลิตภัณฑ์ เวลาทำงานเสร็จก่อนกำหนดของผลิตภัณฑ์ และเวลาทำงานเสร็จล่าช้ากว่ากำหนดของผลิตภัณฑ์ มีค่ามากกว่าศูนย์ ของทุกผลิตภัณฑ์

สมการที่ (12) กำหนดให้เวลาเสร็จงานของชิ้นส่วน เวลาทำงานเสร็จก่อนกำหนดของชิ้นส่วนมีค่ามากกว่าศูนย์ ของทุกชิ้นส่วนและผลิตภัณฑ์

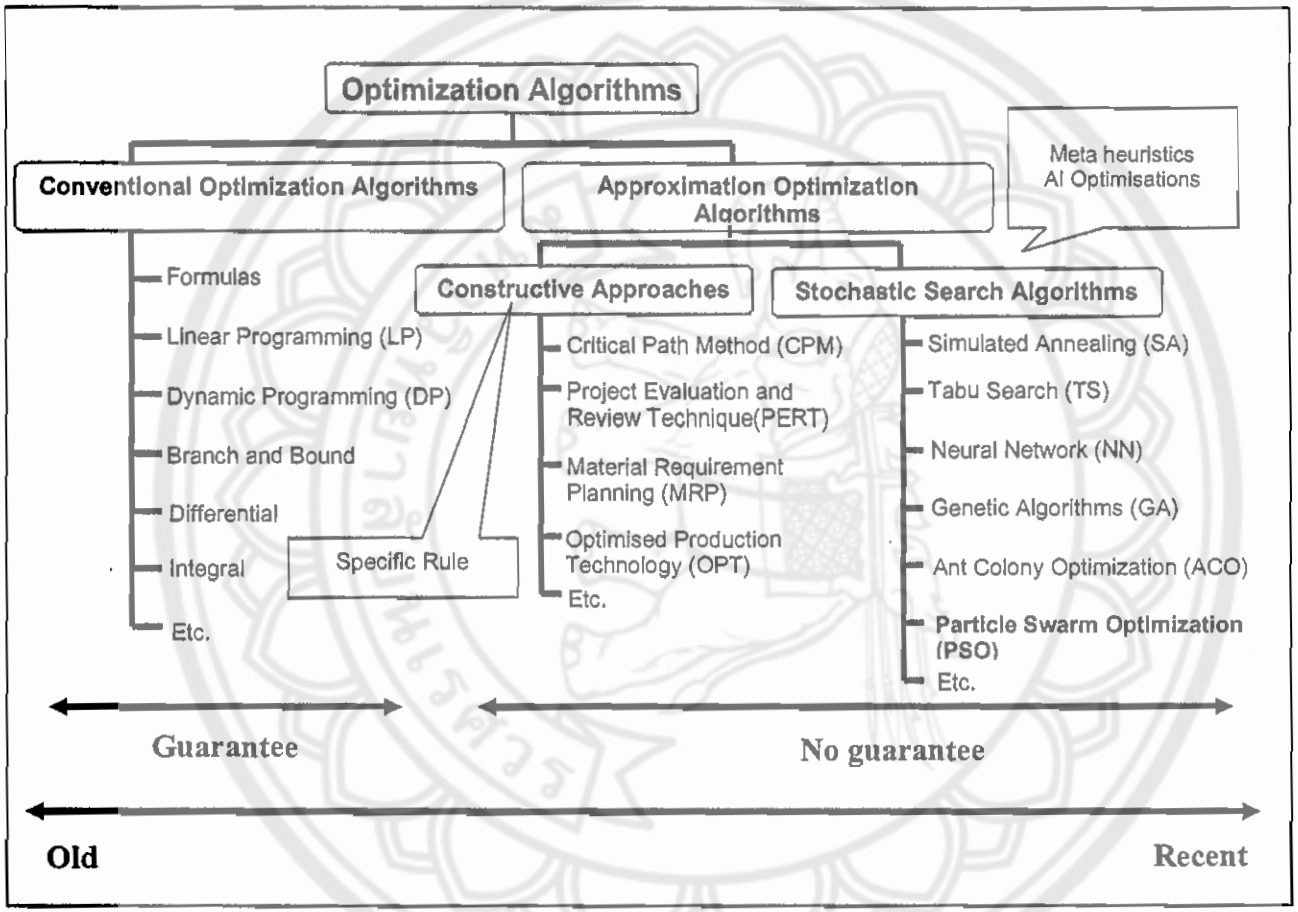
1.5 วิธีการที่นำมาใช้แก้ปัญหาการจัดตารางการผลิต

การจัดตารางการผลิตในโรงงานผลิตแต่ละแห่งนั้น มีความแตกต่างกันออกไปในหลายแนวทาง และมีวิธีการแก้ปัญหาหลากหลายรูปแบบ เช่น ใช้วิธีการจัดเรียงเครื่องจักรที่แตกต่างกันขึ้นอยู่กับระบบการทำงานและปัจจัยแวดล้อมต่าง ๆ ขององค์กร (ปารเมศ ชุติมา, 2546.) ทฤษฎีการจัดตารางได้เสนอแนะเทคนิคจำนวนมาก ที่มีประโยชน์ต่อการแก้ปัญหาการจัดตาราง เช่น เทคนิคการหาค่าที่เหมาะสมที่สุดเชิงการจัด (Combinationatorial optimization) คอมพิวเตอร์ซิมมิวเลชัน (Simulation) การวิเคราะห์โครงข่าย (Network) และฮิวริสติก (Heuristic) เป็นต้น

2. วิธีการหาค่าคำตอบที่เหมาะสมที่สุด (Optimization algorithms)

ขั้นตอนวิธีการ หรืออัลกอริทึม (Algorithms) เป็นเครื่องมือสำคัญในการแสดงขั้นตอน การทำงานของระบบงาน เพื่อให้การเขียนโปรแกรมเป็นไปอย่างรวดเร็วและง่ายยิ่งขึ้น โดยจะแสดงเป็นภาพหรือข้อความ เพื่ออธิบายลำดับขั้นตอนการทำงานของโปรแกรม การเลือกใช้เครื่องมือตัวใดตัวหนึ่งนั้นขึ้นอยู่กับความถนัด หรือความชอบของโปรแกรมเมอร์ เช่น Flow chart ซูโดโค้ด (Pseudo code) เป็นลักษณะข้อความที่เขียนขึ้นโดยผสมผสานระหว่างคำภาษาอังกฤษกับภาษาคอมพิวเตอร์ ซึ่งมีเป้าหมาย เพื่อใช้บรรยายลำดับขั้นตอนการทำงานและไม่มีการใช้รูปหรือสัญลักษณ์แต่อย่างใด ซูโดโค้ด จะมีส่วนที่คล้ายกับการเขียนโปรแกรมจริงมาก ดังนั้นจึงเป็นรูปแบบการอธิบายอัลกอริทึมที่สามารถดูแล้วแปลงเป็นภาษาโปรแกรมมิ่งได้สะดวกและเป็นที่นิยมใช้กันมากในปัจจุบัน แต่ในปัญหาเดียวกันเราสามารถคิดอัลกอริทึมเพื่อมาแก้ปัญหาได้หลายแบบ ซึ่งในแต่ละแบบเครื่องคอมพิวเตอร์ก็จะใช้หน่วยความจำ และเวลาในการประมวลผลไม่เท่ากัน ดังนั้น การจะเปรียบเทียบว่าโปรแกรมคอมพิวเตอร์ใครเก่งกว่าใครกันนั้น จึงใช้การเปรียบเทียบประสิทธิภาพของอัลกอริทึมด้วย

เอกสารประกอบการสอนวิชาการจัดการโครงงาน (Pongcharoen, 2004) จากภาพ 4 แสดงประเภทการหาค่าคำตอบที่เหมาะสมที่สุด (Optimization algorithms) ซึ่งแบ่งออกเป็น 2 ประเภท วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ (Conventional optimization algorithms: COAs) เป็นการอาศัยหลักทางคณิตศาสตร์ ซึ่งมีสูตรการคำนวณที่แน่นอน สามารถหาคำตอบของปัญหาได้คงที่แน่นอนเสมอ ดังนั้นคำตอบที่ได้จึงเป็นคำตอบที่คงที่เสมอ ไม่ว่าจะคำนวณสักกี่ครั้งก็ตาม และวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณ (Approximation optimization algorithms: AOAs) เป็นวิธีการหาค่าคำตอบที่เหมาะสมที่สุดด้วยหลักของการสุ่มค่า ซึ่งสามารถแบ่งออกเป็น 2 แบบคือ Constructive approaches เป็นการอาศัยกฎเกณฑ์ต่าง ๆ มาเป็นเงื่อนไข เพื่อกำหนดค่าของคำตอบที่เหมาะสมที่สุด และ Stochastic search algorithms **เป็นการทำงานแบบวนซ้ำ เพื่อค้นหาค่าคำตอบที่เหมาะสมที่สุด ซึ่งวิธีการหาค่าคำตอบทั้งสองแบบของวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณ (AOAs) ไม่ได้รับรองคำตอบที่ได้ว่าจะคำนวณออกมาแล้วต้องได้ค่าคำตอบเดิมทุกครั้ง เนื่องจากอาศัยหลักการสุ่มค่า เพื่อค้นหาค่าคำตอบที่ได้ในแต่ละการวนซ้ำ**



ภาพ 4 แสดงประเภทการหาค่าคำตอบที่เหมาะสมที่สุด (Optimization algorithms)

แล้วเลือกค่าคำตอบที่ดีที่สุดเมื่อพบค่าคำตอบที่ต้องการหรือจำนวนวนรอบสูงสุดตามที่กำหนดไว้แล้ว จึงหยุดการทำงาน แม้ว่าวิธีการนี้ไม่รับรองค่าคำตอบที่ได้ แต่ในเวลาที่จำกัด และต้องการค่าคำตอบที่ใกล้เคียงค่าคำตอบที่ดีที่สุด ที่สามารถยอมรับค่าคำตอบที่เกิดขึ้นได้ ดังมีรายละเอียดต่าง ๆ ที่จะกล่าวถึงดังต่อไปนี้

วิธีการหาค่าคำตอบที่ดีที่สุด (Optimization algorithms) จำแนกได้เป็น 2 ประเภทคือ

2.1 **วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์** (Conventional optimization algorithms: COAs)

วิธีการในกลุ่มนี้ถูกพัฒนาขึ้นมาในสมัยสงครามโลกครั้งที่ 2 โดยมีจุดประสงค์ในการแก้ปัญหาทางด้านการทหารที่มีความซับซ้อน (Pongcharoen, 2001) หลังจากนั้นวิธีการเหล่านี้ได้ถูกนำไปใช้แก้ปัญหาต่าง ๆ อย่างแพร่หลาย เช่น ปัญหาด้านการจัดการตาราง (Scheduling problem) เมื่อศึกษาจากงานวิจัยหลาย ๆ งานทำให้ทราบถึงวิธีการที่นำมาใช้การหาค่าคำตอบนั้นมีอยู่หลายวิธียกตัวอย่าง เช่น วิธีโปรแกรมเชิงเส้น (Linear Programming), วิธีการพลวัต (Dynamic Programming) และวิธีบรานช์แอนด์บาวด์ (Branch-and-Bound algorithms) เนื่องจากวิธีการในกลุ่มของ COAs นี้เหมาะกับปัญหาขนาดเล็ก (Nagar, Haddock & Heragu. 1995; Pongcharoen, 2001) และมีกฎเกณฑ์ในการหาค่าของคำตอบที่ตายตัวเกินไป (Enumerative search) ทำให้เกิดความยุ่งยากในการคำนวณหาค่าคำตอบ และใช้เวลาในการคำนวณที่นานมากสำหรับปัญหาที่มีขนาดใหญ่ ที่มีความซับซ้อนมาก ซึ่งต้องใช้วิธีการอื่นที่เหมาะกว่า และจะกล่าวถึงในหัวข้อต่อไป ตัวอย่างของนักวิจัยหลายท่านที่นำวิธีการในกลุ่ม COAs นี้ใช้ในการแก้ปัญหา

2.2 **วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณ** (Approximation optimization algorithms: AOAs) (Nagar, Haddock & Heragu, 1995; Blazewicz. Domschke & Pesch, 1996a; Blazewicz et al., 1996b)

วิธีการในกลุ่มนี้จะมีรูปแบบการค้นหาแบบสุ่ม (Stochastic search) มีความเหมาะสม และทำได้ดีเมื่อนำไปประยุกต์ใช้กับปัญหาขนาดใหญ่ที่มีความซับซ้อนสูงอย่างปัญหาในตระกูลของการหาค่าที่ดีที่สุดเชิงการจัด (Combinatorial optimization problems) (Nagar Haddock & Heragu. 1995, Pongcharoen, 2001) เมื่อพิจารณาปัญหาในการหาค่าที่ดีที่สุดโดยอาศัยการจัดเรียง (Sequencing optimization problems) คือแต่ละคำตอบจะแตกต่างกันเมื่อลำดับของทรัพยากรแตกต่างกัน เช่น งาน (Job) ในปัญหา Scheduling เมื่อปัญหามีขนาดใหญ่มากขึ้น การจะหาและนำค่าคำตอบที่เป็นไปได้ทั้งหมดมาเปรียบเทียบกันแล้วเลือกค่าคำตอบที่ดี

ที่สุดนั้นเป็นเรื่องยากมากด้วย ด้วยเหตุนี้จึงทำให้มีวิธีการหาค่าคำตอบที่ดีที่สุดโดยการประมาณเกิดขึ้น (Pongcharoen, 2001) ได้แก่ วิธีการเมทต้าฮิวริสติก (Metaheuristics) ซึ่งเป็นสาขาหนึ่งของ AOAs ที่ประสบความสำเร็จอย่างมากในการจัดการกับปัญหาที่มีความซับซ้อนสูง ๆ โดยวิธีการ ในกลุ่ม เมทต้าฮิวริสติกนี้ จะมีรูปแบบของการวนซ้ำ (Iterative) เป็นการนำลักษณะเด่นที่เหมือนกัน แต่จะแตกต่างกันตรงกลไกที่ถูกนำมาใช้พิจารณาในการค้นหาและสำรวจกลุ่มของค่าคำตอบ ที่เป็นไปได้ให้มีประสิทธิภาพ เพื่อให้ได้มาซึ่งค่าคำตอบที่ใกล้เคียงความเป็นจริงมากที่สุด (Near optimum solution) (Osman & Laporte, 1996) ดังนั้นวิธีการที่จะกล่าวถึงต่อไปนี้ คือ วิธีการ ต่าง ๆ ที่อยู่ในกลุ่มของเมทต้าฮิวริสติก (Metaheuristics) ซึ่งได้แก่ ซิมูเลทเท็ดแอนนิลลิง (Simulated annealing: SA), ทาร์บูเสิร์จ (Tabu Search: TS), นิวรอลเน็ทเวิร์ค (Neural Network: NN), จีเนติกอัลกอริทึม (Genetic Algorithms: GA), แอนท์คอลโลนีออฟติไมเซชัน (Ant Colony Optimization: ACO), และ พาร์ทิเคิลสวอมออฟติไมเซชั่น (Particle Swarm Optimization: PSO) อธิบายรายละเอียดของวิธีการดังกล่าวดังต่อไปนี้

ซิมูเลทเท็ดแอนนิลลิง (Simulated Annealing: SA)

วิธีการซิมูเลทเท็ดแอนนิลลิง (Simulated Annealing: SA) อาศัยหลักการทางอุณหพลศาสตร์ของกระบวนการอบเหนียว (Annealing) ซึ่งเป็นวิธีการที่จำลองแบบของการลดอุณหภูมิลงอย่างช้า ๆ ระหว่างการหลอมโลหะ เพื่อให้โลหะนั้นอ่อนตัวแล้วค่อย ๆ ทำให้เย็นลง เพื่อให้ได้โลหะที่เหนียวและอยู่ในสภาวะที่เหมาะสมที่สุด โดย ซิมูเลทเท็ดแอนนิลลิง (Simulated Annealing: SA) ถูกพัฒนาขึ้นมาจากแนวคิดของ (Kirkpatrick Gelatt & Vecchi., 1983) ซึ่งซิมูเลทเท็ดแอนนิลลิง (Simulated Annealing: SA) จะอาศัยรูปแบบการค้นหาแบบสุ่ม (Stochastic search) จากการสุ่มค่าของกลุ่มคำตอบเพื่อหาค่าคำตอบที่ใกล้เคียง (Neighborhood space) และมีค่าอุณหภูมิ (Temperature: t) เป็นตัวแปรควบคุม (Control parameter) ซึ่งค่าคำตอบที่เหมาะสมที่สุด (Fitness function) ที่กำกับผลเฉลยเปรียบเสมือนพลังงานของสถานะ และทำงานแบบวนซ้ำเปรียบเสมือนกับการค่อย ๆ ลดอุณหภูมิลงเรื่อย ๆ สำหรับขั้นตอนในการทำงานนั้นเริ่มต้นจากการสุ่มสถานะเริ่มต้น (Initial state) ซึ่งประกอบด้วยตำแหน่งของค่าคำตอบเริ่มต้นในกลุ่มของค่าคำตอบที่เป็นไปได้ทั้งหมด (Solution space) และเวลาเริ่มต้น (Initial temperature: t_0) จากนั้น ก็เข้าสู่ขั้นตอนการสุ่มหาตำแหน่งของค่าคำตอบตัวถัดไปโดยการพิจารณาจากกลุ่มค่าคำตอบ ที่ข้างเคียงของตำแหน่งเริ่มต้น ถ้าค่าคำตอบที่ได้เป็นค่าคำตอบที่ดีกว่าเดิม ซึ่งพิจารณาได้จากกลุ่มค่าคำตอบที่ได้จากฟังก์ชันเป้าหมาย (Objective function) ก็ให้ใช้ตำแหน่งของค่าคำตอบนั้น เป็นจุดเริ่มต้นในการหาตำแหน่งอื่นต่อไป แต่ถ้าค่าคำตอบที่ได้ ไม่ดีกว่าเดิมก็อาจจะยอมรับ ค่าคำตอบใหม่ที่แย่กว่าค่าคำตอบเดิมที่ได้ จากพฤติกรรมของกระบวนการฟิสิกส์

ระหว่างการอบเหนียวการอบเหนียว หมายความว่า ในระยะแรกของการวนซ้ำค่าอุณหภูมิเริ่มต้นจะมีค่าสูงทำให้โอกาสในการยอมรับ (Probability of accepting) ค่าคำตอบที่แย่กว่านั้นมีค่าสูงตามไปด้วย รูปแบบการค้นหาก็จะหยาบ ๆ เพื่อให้ขอบเขตในการค้นหานั้นสามารถแผ่ขยายออกไปได้ในวงกว้างไม่ยึดติดอยู่กับกลุ่มของค่าคำตอบเพียงกลุ่มใดกลุ่มหนึ่ง แต่เมื่อค่าอุณหภูมิลดลงเรื่อย ๆ จะทำให้การยอมรับค่าคำตอบที่แย่กว่านั้นเป็นไปได้ยากขึ้น อาจกล่าวได้ว่าการที่อุณหภูมิต่ำ ๆ ค่าคำตอบที่แย่กว่าจะไม่สามารถถูกยอมรับได้เลย ซึ่งค่าคำตอบที่จะถูกยอมรับได้จะต้องดีกว่าค่าคำตอบเดิมเท่านั้น ดังนั้นการค้นหารูปแบบในช่วงอุณหภูมิต่ำ ๆ นี้ จะมีขอบเขตของการค้นหาที่แคบลง แต่จะมีการค้นหามากขึ้นกว่าเดิม (Osman & Laporte, 1996, Pongcharoen, 2001) และการวนซ้ำจะสิ้นสุดเมื่ออุณหภูมิเย็นลงถึงช่วงที่กำหนด หรือสภาพที่ไม่ได้ค่าคำตอบที่ดีกว่าหรือเกินเวลาที่ตั้งเกณฑ์ไว้

```
Pseuducode for a SA procedure
Select an initial state i ∈ S
Select an initial temperature T > 0;
Set temperature change counter t = 0;
Repeat
        Set repetition Counter n = 0;
        Repeat
                Generate state j,a neighbour of i;
                Calculate δ = f(j) – f(i);
                If δ < 0 Then i;= j;
                Else if random (0,1) < exp(-δ/T) Then i := j;
                n := n + 1
        Unit n = N(t);
        t := t + 1
        T := T(t);
Until Stopping Criterion true;
```

ภาพ 5 แสดง Pseuducode ของ SA (Naji - Azimi, 2005 P.709)

ทาร์บูเสิร์จ (Tabu Search: TS)

วิธีการค้นหาแบบทาร์บูเสิร์จ (Tabu Search: TS) เป็นการค้นหาค่าคำตอบที่อาศัยหลักการวนซ้ำ (Iterative search) หลาย ๆ ครั้งจึงจะได้มาซึ่งค่าคำตอบที่เหมาะสมที่สุด เช่นเดียวกับ SA และ GA ซึ่ง TS ถูกพัฒนาขึ้นโดย Glover (1989) กล่าวว่า ทาร์บูเสิร์จ (Tabu Search: TS) จะอาศัยหลักการหาค่าคำตอบข้างเคียง (Neighbor search) โดยการทำงานจะเริ่มจากการสุ่มสถานะเริ่มต้นซึ่งคล้ายกับการทำงานของซิมูเลทเท็ดแอนนิลลิง (Simulated Annealing: SA) จากนั้นก็จะค้นหาค่าคำตอบข้างเคียงที่ดีที่สุดของค่าคำตอบ ในขณะนั้น เมื่อได้ค่าคำตอบใหม่ที่ดีกว่าแล้วก็จะถือว่าเป็น 1 รอบการทำงาน (Iteration) และให้วนซ้ำเช่นนี้ไปเรื่อย ๆ จนจบการทำงาน โดยที่แต่ละรอบนั้นก็จะใช้ค่าคำตอบที่ได้จากรอบที่แล้วมาเป็นค่าคำตอบเริ่มต้นในการทำงานต่อไปแต่ลักษณะเฉพาะของ ทาร์บูเสิร์จ (Tabu Search: TS) ที่ต่างจากวิธีการอื่นคือ จะพิจารณาเฉพาะค่าคำตอบถัดไปที่ไม่ถูกเลือกมาเท่านั้น โดยจะทำการค้นหาในบริเวณพื้นที่ของค่าคำตอบที่ไม่เคยไปมาก่อนถึงแม้ว่าจะได้ค่าคำตอบที่แย่กว่าก็ตาม ดังนั้นจึงมีการเก็บประวัติค่าของคำตอบไว้เป็นรายการต้องห้ามที่เรียกว่า ทาร์บูลิสต์ (Tabu list) ที่ถูกสร้างขึ้นเพื่อป้องกันมิให้เกิดการค้นหาที่ซ้ำซ้อน (Cycling) โดยในลิสต์จะเก็บค่าคำตอบที่เคยถูกนำไปเป็นค่าคำตอบเริ่มต้นในการหาค่าคำตอบที่ข้างเคียง และได้ถูกทดแทนด้วยค่าคำตอบ ที่ดีกว่าไปแล้ว Tabu จึงกำหนดให้ค่าคำตอบเริ่มต้นดังกล่าวนี้เป็นค่าคำตอบต้องห้าม (Forbidden solution) และจะถูกนำไปเก็บเอาไว้ใน Tabu list นั่นหมายความว่า ภายใน m รอบของการทำงาน นับตั้งแต่รอบการทำงานที่ค่าคำตอบต้องห้ามนี้ถูกนำไปใช้เป็นค่าคำตอบเริ่มต้น ซึ่งจะไม่ยอม ให้ค่าคำตอบต้องห้ามนี้ถูกนำไปใช้เป็นค่าคำตอบเริ่มต้นอีก

```
Pseuducode for a TS procedure
Initialization
        s := initial solution in X;
        nbiter := 0;
          (* current iteration *)
        bestiter := 0;
          (* iteration when the best solution has been found *)
        bestsol := s;
          (* best solution *)
        T := 0;
          Initialize the aspiration function A;
While (f(s) > f') and (nbiter – bestiter < nbmax) do
        nbiter := nbiter + 1;
        Generate a set V* of solutions s_i in N(s) which are either
        not tabu or such that A(f(s)) >= f(s_i);
        Choose a solution s* minimizing f over V*;
        Update the aspiration function A and the tabu list T;
        If f(s*) < f(bestsol) then
        bestsol := s*; bestiter := nbiter;
          s := s*;
End while
```

ภาพ 6 แสดง Pseuducode ของ TS (Naji - Azimi, 2005)

นิวรอลเน็ทเวิร์ค (Neural Network: NN)

นิวรอลเนท์เวิร์ค (NN) จะมีโครงสร้างการทำงานที่เลียนแบบพฤติกรรมการทำงานของสมองมนุษย์ โดยจะเก็บความรู้ที่ได้จากกระบวนการเรียนรู้ (Learning process) เอาไว้เพื่อใช้เป็นประสบการณ์ในการทำงานครั้งต่อไป และ Osman & Laporte (1996) ได้กล่าวไว้ว่า นิวรอลเน็ทเวิร์ค (NN) เป็นเครื่องมือที่มีประสิทธิภาพมากในแขนงหนึ่งของวิทยาศาสตร์

ป TS
155.6
ร112ก
2550
21 พ.ย. 2550
1.3979827
สำนักหอสมุด

และวิศวกรรมศาสตร์ ถูกนำมาประยุกต์ใช้กับปัญหาการค่าที่เหมาะสมที่สุด (Optimization problem) โดย นิวรอลเนท์เวิร์ค (NN) มักจะถูกนำไปใช้ในการพยากรณ์ (Predict) การจัดกลุ่ม (Classifly) หรือใช้ในการจดจำรูปแบบต่าง ๆ (Recognise pattern) ได้เป็นอย่างดี เช่น ใช้แก้ปัญหาการเดินทางของเซลส์แมน (Traveling Salesman Problem: TSP) Hopfield Network ได้นำ NN ไปแก้ปัญหาโดยได้เสนอ NN แบบใหม่ที่เรียกว่า Hopfield network ซึ่งสามารถหาค่าคำตอบที่ดีที่สุด สำหรับ TSP ได้เป็นครั้งแรก หรือนำไปใช้แก้ปัญหาการจัดตาราง (Scheduling problem) ได้นำเอา นิวรอลเนท์เวิร์ค (NN) รูปแบบต่าง ๆ มาประยุกต์ใช้ในการจัดตารางสอน (Class/Teacher Timetabling Problem: CTTP) โดยเริ่มต้น จากการแสดงลักษณะของปัญหา Hard constraints และ Soft constraints สำหรับการหาค่าในสมการหรือ Energy function ที่จะเอาไปใช้ใน NN แบ่งออกเป็น 2 แบบ คือ แบบ Potts mean-field บนพื้นฐานของ Continuous potts neurons แบบที่สองจะเป็นรูปแบบ Discrete neural network ผลจากการทดลองพบว่าแบบนี้มีประสิทธิภาพที่ดีกว่าทั้งในด้านเวลา และค่าคำตอบเหมาะสมที่ได้จาก Discrete neural network สามารถนิวรอลเนท์เวิร์ค (NN) เช่น Rafiq, Bugmann & Easterbrook. (2001) นำไปช่วยในการออกแบบของวิศวกร เนื่องจากคอมพิวเตอร์คือส่วนหนึ่งของกิจกรรม Engineers design ในทุก ๆ วันนี้ และ วิศวกรได้ใช้ประโยชน์จากโปรแกรมหลาย ๆ ตัว ในการช่วยงานออกแบบของพวกเขา ถึงแม้ว่าคอมพิวเตอร์จะถูกใช้ในหลาย ๆ งานทางวิศวกรรม ปัจจุบันความสำคัญหลักของโปรแกรมต่าง ๆ มุ่งเน้นไปที่การ Defined rules ที่ดี กิจกรรมที่เกี่ยวข้องกับ Conceptual state ของการออกแบบซึ่งไม่สามารถที่จะทำได้โดยการใช้คอมพิวเตอร์

Artificial Neural Networks (ANN) ปัจจุบันมีการใช้กันอย่างกว้างขวางมาแล้วก่อนหน้านี้ การใช้ ANN มีคุณสมบัติอันหนึ่งที่แตกต่างจากที่เคยมีมาแล้วว่าสามารถเรียนรู้ได้จากประสบการณ์และตัวอย่าง จากนั้นมันจะมีการปรับเปลี่ยนสถานการณ์เอง วิศวกรมักจะต้องขบคิดเกี่ยวกับ Incomplete และ Noisy data ซึ่งสิ่งเหล่านี้ NN สามารถทำได้ทั้งหมด ซึ่งสิ่งนี้เป็น Conceptual stage

นำเสนอแนวทางการออกแบบ ANN สำหรับ Engineering applications ด้วย NN 3 ประเภทคือ Mulit-Layer Perceptron: MLP, Radial Basis network:RBF และ Normalized RBF: NRBF เป็นวิธีการใหม่สำหรับเลือกทำการ Normalization ของ Training data

อีกตัวอย่างหนึ่งของการประยุกต์ใช้ Heuristic method บนพื้นฐานของ Artificial neural networks โดย Fernandez & Gomez. (2005) ได้นำมาใช้ในการกำหนดขอบเขตความสัมพันธ์ที่มีประสิทธิภาพสำหรับ การเลือกปัญหาของ Portfolio เราพิจารณาใช้วิธีการ

มาตรฐานของ Markowitz meanvariance ซึ่งรวมถึง Cardinality และ Bounding constraints เงื่อนไขเป็นการลงทุนในทรัพย์สินที่แตกต่างกัน และจำนวนทุนที่จำกัดในแต่ละทรัพย์สิน นำเสนอประสบการณ์บางส่วนซึ่งเป็นผลมาจาก NN heuristic และเปรียบเทียบผลที่ได้กับอีก 3 Heuristic methods การเลือกปัญหา Portfolio เป็นตัวอย่างที่ได้จากชุดของ Quadratic programming problems เมื่อเลือกใช้วิธีการของ Markowitz ยังรวมถึง Cardinality และ Bounding constraints

Portfolio selection problem จะเป็นการผสมผสาน Quadratic และ Integer programming problem ด้วย โดยพิจารณาจาก Model ที่มีอยู่ในตอนนี้จะเห็นว่าไม่มี Algorithm ใดๆ เลยที่จะเป็นแนวทาง แก้ไขปัญหา Portfolio selection problem ได้อย่างมีประสิทธิภาพ การใช้ Heuristic algorithms ในกรณีนี้มีความจำเป็นอย่างหลีกเลี่ยงไม่ได้ จึงนำ Neural Network (NN) แบบ Hopfield Network, มาใช้เพื่อแก้ไขปัญหาและประยุกต์ใช้กับ Portfolio selection problem และเปรียบเทียบผลที่ได้รับกับผลที่ได้จากวิธีการเดิม (Previous heuristic algorithm)

จีเนติกอัลกอริทึม (Genetic Algorithms: GA)

จีเนติคอัลกอริทึม จะมีวิธีการในการหาค่าคำตอบ โดยอาศัยรูปแบบกลไกการคัดสรรพันธุกรรมการเลือกจากธรรมชาติ ซึ่งพันธุกรรมที่ดี (ค่าคำตอบที่เหมาะสมที่สุด) จะสามารถปรับตัวเอง จากสิ่งแวดล้อมที่มีการเปลี่ยนแปลงอยู่ตลอดเวลาให้ตัวเองสามารถอยู่รอดได้ และสืบทอดลักษณะที่ดีกล่าวไปยังรุ่นต่อ ๆ ไปได้ (Goldberg, 1989)

เริ่มต้นจากการสุ่มค่าคำตอบขึ้นมาจำนวนหนึ่ง กำหนดให้เป็น โครโมโซม (Chromosome) โดยทุกโครโมโซมจะมีหน่วยเล็ก ๆ มาเรียงตัวกันที่เราเรียกว่ายีน (Gene) ของกลุ่มประชากร (Population) ซึ่งจะมีกระบวนการทางพันธุกรรม (Genetic operation) มี 2 ขั้นตอน คือ การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) เมื่อผ่านกระบวนการทางพันธุกรรมแล้ว โครโมโซมทั้งหมดจะถูกประเมิน (Evaluate) เพื่อคำนวณหาค่าความเหมาะสม (Fitness value) ที่จะกลายเป็นโอกาสในการอยู่รอดของแต่ละโครโมโซม (Probability of selection) และเข้าสู่กระบวนการคัดสรร (Selection operation) ซึ่งวิธีการคัดสรรที่ได้รับความนิยมที่สุดวิธีหนึ่งก็คือ การคัดสรรด้วย วงล้อเสี่ยงทาย (Roulette wheel) (Goldberg, 1989) หากการเสี่ยงทายตกอยู่ในช่องใดของวงล้อเสี่ยงทาย โครโมโซมที่ช่องนั้นอ้างถึงก็จะถูกเลือกเพื่อให้เป็นโครโมโซมในรุ่น (Generation) ต่อไป ซึ่งความกว้างของวงล้อเสี่ยงทายในแต่ละช่องจะแปรผันไปตามโอกาสในการอยู่รอดของโครโมโซม ที่สามารถอยู่รอดได้มากจะมีขอบเขตของวงล้อที่กว้างมากตามไปด้วย ทำให้โอกาสที่จะถูกคัดสรรก็มีมากขึ้นตามไปด้วย

มีตัวอย่างงานวิจัยที่มีการประยุกต์ใช้จีเนติกอัลกอริทึม (Genetic Algorithms: GA) เช่น Pongcharoen et al. (2001) ได้ศึกษาและทดสอบการสลับสายพันธุ์ และกลายพันธุ์กับปัญหา

การจัดตาราง (Scheduling problem) โดยได้รวมตัวแปรอื่น ๆ ที่มีผลต่อการทำงานของจีเนติกอัลกอริทึม (Genetic Algorithms: GA) เข้ามาพิจารณาด้วย อันได้แก่ ขนาดของประชากร (Population size), จำนวนรุ่นของโครโมโซม (Generation), เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการสลับสายพันธุ์ (%C), เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการกลายพันธุ์ (%M) และวิเคราะห์ผลการทดลอง โดยใช้หลักการทางสถิติซึ่งสรุปผลว่า การสลับสายพันธุ์ และการกลายพันธุ์ที่ดีที่สุดเป็นการใช้ วิธีการของ Enhanced edge recombination crossover (EERX) และ Two operation adjacent swap (2OAS)

วีณา พรหมเทศ (2548) นำจีเนติคอัลกอริทึมมาประยุกต์ใช้ในการจัดตารางสอน โดยเพิ่มกระบวนการซ่อมแซม เข้าไปเป็นส่วนหนึ่งในขั้นตอนการทำงานของจีเนติกอัลกอริทึม (Genetic Algorithms: GA) สำหรับใช้ดัดแปลง Infeasible timetable ให้เป็น Feasible timetable พบว่าสามารถแก้ปัญหาการจัดตารางสอนและนำไปใช้งานได้จริง

```
Pseuducode for a GA procedure
Begin;
    Generate random population of P solutions (Chromosomes);
        For each individual i ∈ P: calculate fitness (i);
        For i = 1 to number of generations;
            Randomly select an operation (crossover or mutation);
            If crossover;
                Select two parents at random i_a and i_b;
                Generate on offspring i_c = crossover (i_a and i_b);
            Else if mutation;
                Select one chromosome i at random;
                Generate on offspring i_c = mutation (i);
            End if;
            Calculate the fitness of the offspring i_c;
            If i_c is better than the worst chromosome then
            replace the worst chromosome by i_c;
        Next i;
    Check if termination = true;
End;
```

ภาพ 7 แสดง Pseuducode ของ GA (Elbertagi Hegazy & Grierson., 2005)

แอนท์คอลโลนีออฟติไมเซชัน (Ant Colony Optimization: ACO)

ACO เป็นขั้นตอนวิธีที่ได้แนวความคิดมาจากพฤติกรรมการหาอาหารของมด พัฒนาโดย Dorigo Maniezzo & Colorni. (1996) อธิบายไว้ว่า เป็นการค้นหาเส้นทางที่สั้นที่สุดของมดระหว่างอาหารที่พบกับรังของมัน และเพื่อบอกให้มดตัวอื่น ๆ ทราบถึงเส้นทางดังกล่าวที่มดตัวหนึ่งค้นพบ มดแต่ละตัวจะทิ้งหลักฐาน เรียก Pheromone ไว้ตามเส้นทางที่ตนพบอาหาร เมื่อเส้นทางใด มีจำนวนของมดเดินมากขึ้น ก็จะมีความหนาแน่นของ Pheromone มากขึ้น ทำให้หนาแน่นกว่าเส้นทางอื่น ๆ และเมื่อมดตัดสินใจในการเลือกทางเดินครั้งต่อไป มดแต่ละตัวก็จะเลือกเส้นทางดังกล่าว ซึ่งพบว่าเป็นเส้นทางที่สั้นที่เหมาะสมที่สุด

```
Pseudocode for an ACO procedure
Begin;
    Initialize the pheromone trails and parameters;
    Generate population of m solutions(ants);
    For each individual ant k ∈ m calculate fitness (k);
    For each ant determine its best position;
    Determine the best global ant;
    Update the pheromone trail;
  Check if termination = true;
End ;
```

ภาพ 8 แสดง Pseuducode ของ ACO (Elbertagi, Hegazy & Grierson., 2005

พาร์ทิเคิลสวอมออฟติไมเซชั่น (Particle Swarm Optimization: PSO)

Particle swarm optimization เป็นแขนงหนึ่งของ Artificial life และ Social psychology พัฒนาโดย Eberhart & Kennedy (1995) ซึ่งสามารถนำมาใช้แก้ปัญหาทางด้านวิทยาการคอมพิวเตอร์ และวิศวกรรม ตลอดจนปัญหาที่ซับซ้อนอื่น ๆ ได้เป็นอย่างดี ซึ่งเป็นการนำวิธีการของ Evolutionary computation ประกอบด้วยกลุ่มของประชากร ซึ่งจะเรียกว่า Particles โดยแต่ละ Particles จะบินไปทั่วทั้งหมดของบริเวณพื้นที่ของปัญหา ในการกำหนด ค่าเบื้องต้นให้กับ Particles ด้วยการสุ่มค่าตัวแปร (Position and velocity) ให้ทุก Particles ซึ่งจะมีการวนซ้ำ โดยในแต่ละการวนซ้ำนั้น แต่ละ Particles จะต้องรักษาระดับของ

ความเร็วที่ใช้ในการบิน (Velocity) ซึ่งจะเก็บบันทึกไว้เป็นสถิติของแต่ละตัว เพื่อทำการปรับปรุงและพัฒนาตัวเอง ในการหาตำแหน่ง (Position) ของค่าคำตอบที่เหมาะสมที่สุด (Fitness solution)

```
Pseudocode for a PSO procedure
For each particle
    Initialize particle
End For
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pBest) in history
Set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
    Calculate  particle velocity according equation (a)
    Updete  particle position according equation (b)
End
    While maximum iterations or minimum error criteria is not attained
```

ภาพ 9 แสดง Pseuducode ของ PSO (Kennedy, Eberhart &Shi. 2001)

ทฤษฎีพื้นฐานเกี่ยวข้องกับพาร์ทิเคิลสวอมออฟติไมเซชั่น (Particle Swarm Optimization: PSO)

Particle Swarm Optimization เป็นแขนงหนึ่งของ Artificial life และ Social psychology (Shi and Eberhart, 1999, Eberhart & Kennedy, 1995) เป็นวิธีการแนวใหม่ โดยนำแนวคิดในการบินหาอาหารของนก (Bird flocking) การเรียนรู้ของปลา (Fish schooling) กระบวนการในการเรียนรู้ทำให้เกิดความฉลาดได้ ด้วยความเชื่อและความคิดที่เกิดจากการสังเกตความเปลี่ยนแปลงจากสิ่งแวดล้อมต่าง ๆ และทฤษฎีการเคลื่อนที่ (Swarming theory) กันเป็นฝูง ซึ่งที่พัฒนาขึ้นครั้งแรกโดย Dr.James Kennedy และ Dr.Russell Eberhart เป็นแนวคิดสำหรับ Optimization of nonlinear functions โดยใช้ Particles swarm methodology ทำการประเมินค่า

ความเหมาะสม สามารถนำมาใช้แก้ปัญหาทางด้านวิทยาการคอมพิวเตอร์ และวิศวกรรม ตลอดจนปัญหาที่เป็นระบบซับซ้อน ตัวอย่างของระบบซับซ้อน การอพยพของฝูงนก ระบบสมอง ระบบการจราจร และปัญหาอื่น ๆ ที่มีความยุ่งยากในการแก้ปัญหา จัดเป็นปัญหาแบบ NP-hard (Even, Itai & Shamir. 1976) ได้เป็นอย่างดี เมื่อปัญหามีขนาดใหญ่ขึ้น เวลาที่ต้องใช้ในการหาคำตอบที่ดีที่สุดสำหรับขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนาม จะเพิ่มขึ้นน้อยกว่าประเภทที่มีความซับซ้อนด้านเวลาเชิงเลขยกกำลังอย่างมาก และถ้าความเร็วของคอมพิวเตอร์สูงขึ้น ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงเลขยกกำลังจะได้รับผลดีในเชิงบวกเมื่อปัญหาใหญ่ขึ้น ในขณะที่ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนามจะได้รับผลดี ในลักษณะทวีคูณ ดังนั้น ความก้าวหน้าในด้านเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ จะมีประโยชน์ ต่อกระบวนการหาคำตอบที่ใช้ขั้นตอนวิธีที่มีความซับซ้อนด้านเวลาเชิงพหุนามมากกว่าเชิงเลขยกกำลัง ซึ่งเป็นการนำวิธีการของ Evolutionary computation ที่แตกต่างจากจีเนติกอัลกอริทึม (Genetic Algorithms: GA) ก็คือ PSO จะไม่มีการสร้างประชากรใหม่จากพ่อแม่เหมือนจีเนติกอัลกอริทึม (Genetic Algorithms: GA) แต่จะสุ่มค่าให้ประชากรในรอบแรก ซึ่งจะถูกปรุงปรับให้ฉลาดขึ้นจากประสบการณ์ที่ผ่านการเรียนรู้ของตนเอง และจากประชากรที่ดีที่สุด PSO จะประกอบด้วยกลุ่มของประชากรหรือสมาชิก (Population member)ในกลุ่ม หรือ ฝูง คือ รูปแบบของพฤติกรรมการแผ่กระจายของสัตว์ (Reynold, 1987) ซึ่งจะเรียกว่า Particles โดยแต่ละ Particles จะบินไปทั่วทั้งหมดของบริเวณพื้นที่ของปัญหา ในการกำหนดค่าเบื้องต้นให้กับ Particles จะทำโดยการสุ่มค่าตัวแปร (Position and velocity) ให้กับทุก Particles ซึ่งจะมีการวนซ้ำ โดยในแต่ละการทำซ้ำนั้น แต่ละ Particles จะต้องมีระดับของอัตราเร็วที่ใช้ในการบิน (Velocity) ซึ่งจะเก็บบันทึกไว้เป็นสถิติของแต่ละตัว เพื่อทำการปรับปรุง และพัฒนาตัวเองให้เกิดความฉลาดจากประสบการณ์ที่ดีที่สุดของทุก Particles ในการหาตำแหน่ง (Position) ของค่าคำตอบที่เหมาะสมที่สุด (Fitness solution) ที่แต่ละ Particles ซึ่งการบินเป็นฝูง Particles systems เป็นการเคลื่อนที่ของหมู่แมลงตัวเล็กๆ ที่จะบินแผ่กระจายออกไป เข้าหาไฟและควัน ซึ่งเป็นตัวอย่างที่เห็นได้ชัดเจนเป็นการเคลื่อนที่ ซึ่งเรียกว่า Particles Swarm (Reeves, 1983) จากคำกล่าวของ Millonas (1994) ที่ Santa fe institute ได้กล่าวถึง คุณสมบัติของการเคลื่อนที่ (Swarm) กันเป็นกลุ่ม อย่างชาญฉลาด ต้องประกอบด้วยคุณสมบัติพื้นฐานที่สำคัญ 5 ประการด้วยกันคือ

Proximity principle คือหลักการเคลื่อนที่ ให้สัมพันธ์กันในทุกช่วงเวลา มีการรักษาระยะห่างระหว่างกันในการบินเป็นกลุ่ม ซึ่งจะเป็นการป้องกันการชนกันของ แต่ละ Particles ที่เป็นสมาชิกในกลุ่ม

Quality principle **คือ หลักการใช้การปรับตัว หรือการตอบสนองต่อปัจจัยต่าง ๆ จากสิ่งแวดล้อมที่เปลี่ยนแปลงตลอดเวลาของแต่ละ** Particles **เพื่อให้สามารถดำเนินชีวิตอยู่ได้ในทุกสภาวะ**

Diverse response **คือ ความหลากหลายของค่าคำตอบที่เหมาะสม เป็นการไม่ยึดติดกับค่าคำตอบใดคำตอบหนึ่ง** ซึ่งค่าคำตอบที่ได้ จะสามารถปรับปรุงและเปลี่ยนแปลงไปตามปัจจัยที่เหมาะสม ณ เวลาหนึ่ง ๆ

Stability principle คือ หลักการมีเสถียรภาพทางด้านพฤติกรรมของแต่ละ Particles ซึ่งพฤติกรรมของสมาชิกในสังคมจะไม่เปลี่ยนแปลงไปตามสภาพแวดล้อมที่เปลี่ยนแปลงไป Principle of adaptability คือ มีหลักการปรับปรุง พัฒนาตัวเอง สามารถคำนวณค่าที่เหมาะสมหรือค่าคำตอบที่ใกล้เคียงที่เหมาะสมที่สุด ได้อย่างมีประสิทธิภาพ

จากหลักการทั้ง 5 ประการดังกล่าวของ Mark Millonas นำมาอธิบายพฤติกรรมของ Particle Swarm Optimization ได้ว่า (Millonas, 1994)

ในการหาอาหารของสัตว์ตัวเล็กๆ อย่างเช่น ผึ้ง ปลา นก ซึ่งต้องอาศัยการเคลื่อนที่กันไปเป็นกลุ่ม (Term) ที่สัมพันธ์กันของแต่ละ Particles หรือ Population member ซึ่งในทางคณิตศาสตร์ จะเปรียบเหมือนเป็นค่าของจุด ๆ หนึ่ง ซึ่งเรียกว่า Point ซึ่งจะมีค่าของ Velocity ที่สามารถควบคุมให้เหมาะสมแม้ว่าจะมีการหยุดพักของการเคลื่อนที่ (Swarm) ในบางขณะก็ตาม แนวทางในการพัฒนาของ Evolutionary computation มีการศึกษา 4 แนวทาง (Kennedy, Eberhart & Shi. 2001) ที่สำคัญได้แก่

Genetic Algorithms: GA

Evolutionary Programming: EP

Evolution Strategies: ES

Genetic Programming: GP (Genetic Programming บางครั้งอาจถูกจัดอยู่ในหมวด Genetic algorithms)

มีโครงสร้างพื้นฐานในการเรียนรู้สังคมของ PSO ประกอบด้วยคุณสมบัติ 3 ประการ **คือ**

1. **การประเมินสถานการณ์** (Evaluate)
2. **การเปรียบเทียบ** (Compare)
3. **การจำลองหรือเลียนแบบ** (Imitate)

ทฤษฎีที่ทำให้เกิดการเป็นอยู่แบบรวมกันเป็นสังคม (Social) ที่เกี่ยวข้อง Adaptive culture model และ Particles Swarms ซึ่งเป็นขั้นตอนของการเกิดเป็นสังคม จะเป็นการนำรูปแบบของการประยุกต์ใช้วัฒนธรรมในขั้นสูง มาพัฒนาจนเกิดเป็นจารีตประเพณีของสังคมนั้น ๆ ซึ่งได้มาจากการเรียนรู้จากประสบการณ์ที่ผ่านมา นำไปใช้ในการแก้ปัญหาในด้านต่างๆ จนเกิดเป็นวัฒนธรรม หรือประเพณีที่สืบทอดต่อ ๆ กันมารุ่นต่อ ๆ ไป ประกอบด้วย หลักการตามโครงสร้างพื้นฐานของการเรียนรู้เช่นเดียวกับการเรียนรู้ของ Particle Swarm Optimization ทั้ง 3 ประการ

การประเมินสถานการณ์ (Evaluate)

ปัจจัยต่าง ๆ จากสิ่งแวดล้อม มักจะมีแนวโน้มที่จะมากระตุ้นภายในระบบใดระบบหนึ่ง ทำให้เกิดการประเมินค่าของสถานการณ์ ในขณะนั้นว่าอยู่ในระดับใด ดีหรือเลว เกิดความพึงพอใจหรือไม่พอใจ ชอบใจหรือไม่ชอบใจ เกิดความคิดในเชิงบวกหรือในเชิงลบ เหล่านี้ ซึ่งเป็นลักษณะของพฤติกรรมที่ปรากฏในที่ต่าง ๆ ที่มีการรวมตัวกัน อยู่กันเป็นหมู่ จนเกิดเป็นสังคมที่ประกอบด้วยกลุ่มต่าง ๆ มากมาย แม้แต่ในสังคมของแบคทีเรีย ก็จะเกิดความโกลาหลวุ่นวายขึ้นได้ ถ้าสภาพแวดล้อมเกิดเป็นพิษขึ้นมา จะเกิดการเรียนรู้ถึงสิ่งที่จะเกิดขึ้นตามมาจากสภาพแวดล้อมที่เป็นพิษนั้นเป็นอย่างไร ก็จะมีการประเมินสถานการณ์ ณ เวลานั้นว่าเป็นอย่างไร ต้องทำอย่างไร เพื่อให้สามารถดำรงชีวิตอยู่ต่อไปได้ จากการเรียนรู้ (Learning) นี้ ทำให้สามารถอธิบายถึงเหตุและผลของการเปลี่ยนแปลงที่เกิดขึ้นในสังคม ตลอดจนทำการปรับปรุง จากผลของการประเมินค่าสถานการณ์ที่เกิดขึ้นได้ เพื่อความอยู่รอดของสังคมต่อไป

การเปรียบเทียบ (Compare)

Social comparison theory (Festinger,1954) อธิบายแนวคิดบางอย่างที่ประชากรใช้เป็นมาตรฐานวัดระดับความสามารถของตนเองว่าเมื่อเปรียบเทียบกับบุคคลอื่นที่อยู่ในสังคมเดียวกันว่าตนเองจะอยู่ในระดับใด ของมาตรฐานที่เป็นที่ยอมรับของสังคม เพื่อพัฒนาและปรับปรุงตนเองจนเป็นที่ยอมรับของสังคมได้ ซึ่งเดิมทฤษฎีในการเปรียบเทียบทางสังคมของ Festinger (1954) เป็นการศึกษาสังคมในเบื้องต้น เป็นเพียงการทดสอบอย่างง่ายๆ ซึ่งต่อมาจึงถูกพัฒนาขึ้นเป็น Social psychology theories จึงได้รับความสนใจและมีผู้ศึกษามากขึ้น

การจำลองหรือเลียนแบบ (Imitate)

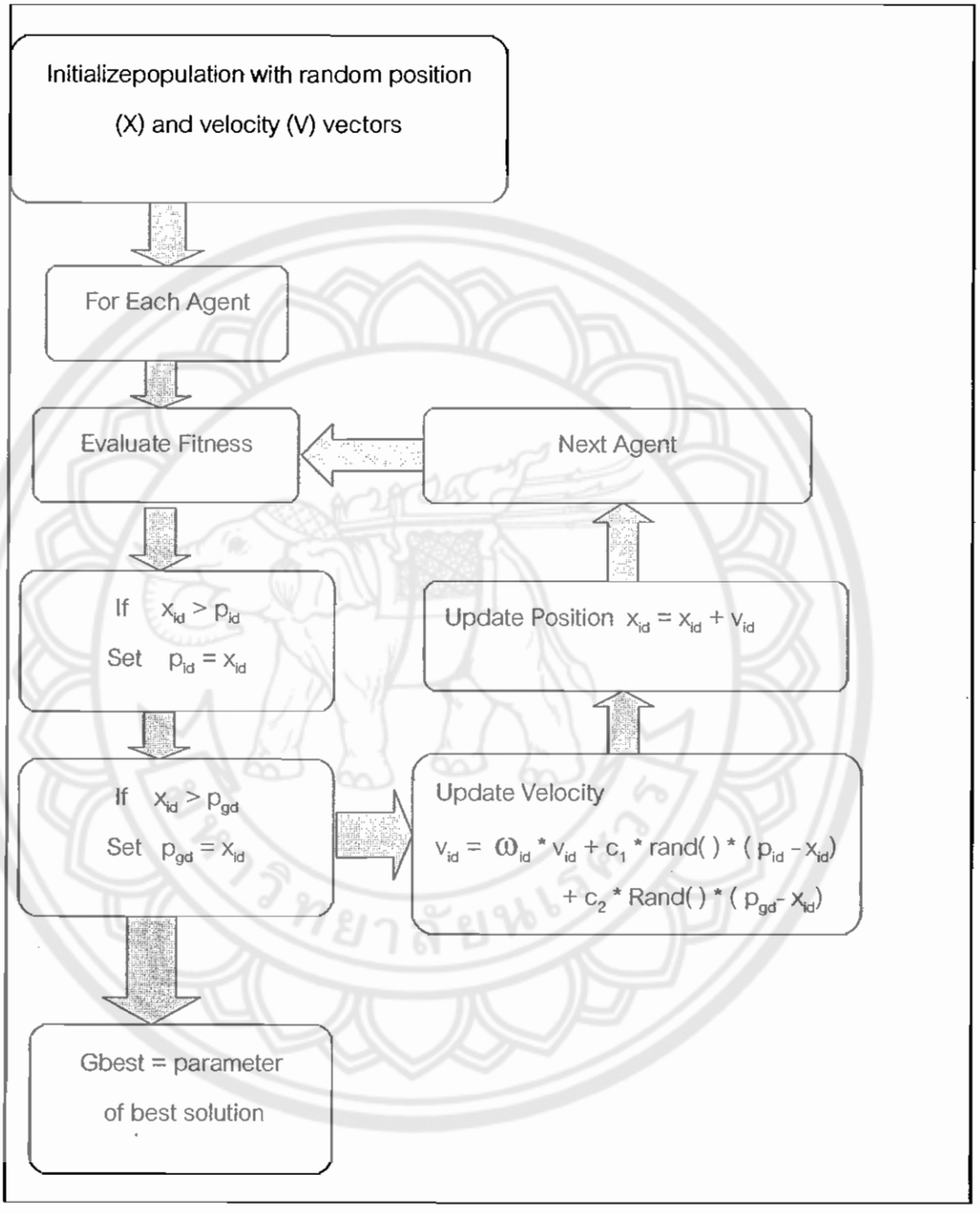
ถ้าคุณคิดที่จะเลียนแบบสิ่งต่าง ๆ หรือพฤติกรรมที่เกิดขึ้นในธรรมชาตินั้น คุณจะพบว่ามีสิ่งต่าง ๆ ที่คุณจะเลียนแบบได้มากมาย มีการเรียนรู้และศึกษาได้หลายแนวทาง การศึกษาพฤติกรรมการอาหารของมดที่จะค้นหาเส้นทางที่สั้นที่สุดระหว่างรังกับแหล่งที่พบอาหาร แล้วตัดสินเลือกทางเดินที่มี Pheromone หนาแน่นที่สุด นำมาประยุกต์ใช้การแก้ปัญหาการเดิน

ทางของพนักงานขาย เพื่อหาคำตอบของระยะทางระหว่าง 2 เมืองที่มีค่าความเหมาะสมสั้นที่สุด (Dorigo & Blum, 2005; Dorigo & Gambardella, 1997)

PSO เป็นการจำลองพฤติกรรมการหาอาหารของฝูงนก โดยอาศัยความร่วมมือของสมาชิกในฝูง ทำการบินไปตลอดพื้นที่ของเป้าหมาย ด้วยการสุ่มเลือกพื้นที่จุดต่าง ๆ ของสมาชิกแต่และตัว ในการเริ่มต้นการค้นหาแหล่งอาหาร ซึ่งเปรียบเสมือนค่าคำตอบที่ต้องการในรอบของการทำงานรอบแรกนกแต่ละตัวจะยังไม่ทราบตำแหน่ง และระยะทางของอาหารว่าอยู่ที่ไหน หรือไกลเท่าใด ดังนั้นวิธีการเดียวที่สามารถทำได้คือการติดตามนกตัวอื่นที่ใกล้แแหล่งอาหารที่สุด PSO เป็นการเรียนรู้จากประสบการณ์ที่ผ่านมา ใช้ในการแก้ปัญหา Optimization problems เราจะเรียกนกแต่ละตัวใน Search space ในบริเวณพื้นที่ของค่าคำตอบ (D-dimensional space) เรียกว่า Particles ซึ่งจะแข่งขันกันหาค่าของคำตอบที่เหมาะสมที่สุด (Fitness values) หรือค่าคำตอบที่ใกล้เคียง (Neighbors) ของค่าที่เหมาะสมที่สุดของปัญหา ซึ่งมีระดับของค่าความเหมาะสม ที่สามารถวัดได้จากค่าคำตอบที่ได้จาก Fitness function ที่ใช้หรือเป็นปัญหาที่ใช้ในการทดสอบค่าคำตอบที่ต้องการหาค่าความเหมาะสมของปัญหาที่สนใจ Robinson, Sinton & Rahmat – Samii. (2002) ได้สนใจนำ GA และ PSO มาประยุกต์ใช้ในการคำนวณค่า finess function จากสมการในการคำนวณหาค่าความถี่ของการรับคลื่นเสียงของลำโพง ซึ่งจำลองปัญหามาเพื่อทดสอบหาค่าของปัจจัยที่เหมาะสมในการส่งเสียงไปได้ในระยะไกลที่สุด

โดยกำหนดค่าของ PSO ใช้ค่าสุ่ม rand () อยู่ในช่วง [0, 1] ค่าปัจจัยการเรียนรู้ตนเองและสังคมให้มีค่าเท่ากับ 2 ค่าถ่วงน้ำหนัก (ω) ให้เท่ากับ [0.4, 0.9] และค่าพาร์ทิเคิล เท่ากับ 10

ในแต่ละอัลกอริทึมกำหนด Population size เท่ากับ 10 ส่วน ปัจจัยของ GA กำหนดความน่าจะเป็นในการ Crossorver เท่ากับ 0.6 และกำหนดค่าความน่าจะเป็นในการ Mutation เท่ากับ 0.1 นำมาทดสอบใน 4 รูปแบบ ได้แก่ GA, GA – PSO, PSO และ PSO – GA พบว่าในรูปแบบ PSO – GA มีค่า Fitness สูงสุด



ภาพ 10 แสดงขั้นตอนการทำงานของ Particle Swarm Optimization (PSO)

(Robinson, Sinton & Rahmat – Samii. 2002)

จากภาพ 10 แสดงขั้นตอนการทำงานของ Particle Swarm Optimization (PSO) ซึ่งเป็นกระบวนการไหลของการทำงานของ PSO โดยแสดงทั้ง 6 ขั้นตอนดังที่กล่าวมาแล้วสามารถเขียนเป็น Flow chart แสดงลักษณะการทำงานที่ใช้กันโดยทั่วไป สามารถอธิบายลำดับขั้นตอนในการทำงานของ Particle Swarm Optimization (PSO) ดังนี้

ก่อนจะศึกษาต่อไป ควรเข้าใจก่อนว่าการ Initialize เป็นการสร้าง Particles ขึ้นมาตั้งแต่เริ่มต้นการทำงานของโปรแกรม เพื่อใช้ในการเก็บค่าของข้อมูลหรือ Particles ที่ใช้ในโปรแกรม ซึ่งจำนวน Particles ที่สร้างขึ้นมานั้นจะมีจำนวนเท่ากับจำนวนของข้อมูลทั้งหมด (N)

วิธีการเบื้องต้นในการหาค่าความเหมาะสมของ Continuous nonlinear functions ด้วยวิธีในการจำลองรูปแบบของสังคมของ Particle Swarm Optimization ซึ่งมีขั้นตอนดังนี้

1. กำหนดค่าเริ่มต้นให้แต่ละ Particles ด้วยการสุ่มค่า โดย

1.1 กำหนดค่าของ d (Dimension)คือ มิติของพื้นที่ค่าคำตอบของปัญหา มีค่าตั้งแต่ 1 ถึง n

1.2 กำหนดค่าของตำแหน่ง (Position) $x_{id} = (x_{i1}, x_{i2}, x_{i3},\ldots x_{in})$

1.3 กำหนดค่าของอัตราเร็ว (Velocity) ในการบินเป็นค่าของ $v_{id} = (v_{i1}, v_{i2}, v_{i3},\ldots v_{in})$

1.4 กำหนดค่าของตำแหน่งที่เหมาะสมที่สุดของตนเอง (Best previous position) $p_{id} = (p_{i1}, p_{i2}, p_{i3},\ldots p_{in})$

1.5 กำหนดค่า c_1 และc_2 โดยทั่วไปกำหนดให้มีค่าเท่ากับ 2 ซึ่งเป็นค่าคงที่แสดงระดับปัจจัยการเรียนรู้ตนเอง และสังคมของ Particles

1.6 การกำหนดค่า rand() และ Rand() เป็นการสุ่มค่าซึ่งอยู่ในช่วง [0,1] ให้กับฟังก์ชัน ในการ Update Velocity ตามสมการ 13 หน้า 36

1.7 กำหนดค่าถ่วงน้ำหนัก (Inertia weight : ω) เท่ากับ 0.9 เป็นค่าที่ใช้ถ่วงน้ำหนักของ v_{id} ในการหาค่าคำตอบของฟังก์ชัน Update velocity ตามสมการ 13 หน้า 36

2. ประเมินค่าความเหมาะสมของแต่ละ Particles ซึ่งจะถูกคำนวณหาค่าคำตอบ**ที่เหมาะสมที่สุด** (Fitness values) **จากสมการฟังก์ชันเป้าหมาย** (Fitness function)

3. **เปรียบเทียบค่าคำตอบที่เหมาะสมที่ได้** (Current values) **กับค่าคำตอบที่ดีที่สุดจากประสบการณ์ที่ผ่านมาของแต่ละ Particles ถ้าได้ค่าคำตอบที่เหมาะสมกว่า ค่า Pbest ในรอบการทำงานที่ผ่านมา ให้กำหนดค่า Current values ให้เป็นค่า Pbest ใหม่ ซึ่งเป็นค่าตำแหน่งที่มีค่าของคำตอบที่เหมาะสมที่สุดของแต่ละ Particles $p_{id} = (p_{i1}, p_{i2}, p_{i3},\ldots p_{in})$ เปรียบเทียบค่าคำตอบ**

ที่เหมาะสมที่ได้ (Current values) กับค่าคำตอบที่ดีที่สุดจากประสบการณ์ที่ผ่านมาของ Particles ทั้งหมด ถ้าได้ค่าคำตอบที่เหมาะสมกว่าค่า Gbest ในรอบการทำงานที่ผ่านมา ให้กำหนดค่า Current values ให้เป็นค่า p_{gd} เป็นค่า Gbest ใหม่ ซึ่งเป็นค่าตำแหน่งที่มีค่าของคำตอบที่เหมาะสมที่สุดของ Particles ทั้งหมด

4. ปรับปรุงค่าของอัตราเร็ว (Velocity) ในการบินในรอบการทำงานของแต่ละ Particles ด้วยสมการ (13) ดังนี้

$$v_{id} = \omega_{id} * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id}) \quad (13)$$

5. ปรับปรุงค่าของตำแหน่ง (Position) ในรอบการทำงานของแต่ละ Particles ด้วยสมการ (14) ดังนี้

$$x_{id} = x_{id} + v_{id} \quad (14)$$

จำนวนชิ้นงานทั้งหมดจะถูกปรับเป็นตัวเลขดังแสดงใน Total operation list โดยมีหมายเลข 0 แทน ผลิตภัณฑ์ {245:1 0} ซึ่งบอกว่าเป็นผลิตภัณฑ์ 245:1 ของ operation ที่ 0 หมายเลข 1 แทนผลิตภัณฑ์ 246:1 มี operation ที่ศูนย์ (0) เป็นเช่นนี้จนครบทั้ง 34 ชิ้นงานของปัญหาขนาดเล็ก ซึ่งจะทำเช่นเดียวกันกับปัญหาขนาดอื่น ๆ ด้วย

จากนั้นจะทำการสุ่มลำดับการทำงานให้กับผลิตภัณฑ์ประกอบทั้งหมด 34 ชิ้น โดยเริ่มจากลำดับ 0 – 33 ซึ่งจะได้ลำดับการทำงานที่ยังไม่สามารถปฏิบัติได้จริง ต้องนำไปผ่านขั้นตอนการ Repair part เพื่อตรวจสอบสายการผลิตและผ่านขั้นตอนการ Repair operation เพื่อตรวจสอบลำดับของ Operation ก่อนและหลังให้ถูกต้องก่อนจะดำเนินการผลิตตามตารางการผลิตได้จริง

จำนวนชิ้นงานทั้งหมดของปัญหาแต่ละขนาดเมื่อทำการ Repair part และทำการ Repair operation เรียบร้อยแล้ว จะถูกนำไปคำนวณวิธีการจัดตารางการผลิตเพื่อหาค่าปรับ (Penalty cost) ของตารางการผลิตที่จัดด้วยพาร์ทิเคิลสวอมออฟติไมเซชั่น

แสดงจำนวนลำดับทั้งหมดที่นำมาสุ่มลำดับการทำงานของปัญหาขนาดเล็ก (Small problem)

ลำดับการจัดตารางการผลิตใช้เป็นตำแหน่งในการสลับโดยบันทึกเป็นคู่ลำดับ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

X(t+1) **เป็นการจัดตารางการผลิต**ที่ถูกปรับปรุงขึ้นใหม่ด้วย PSO

32	28	30	27	25	24	23	22	21	17	20	16	5	18	15	12	14	11	10	8	33	31	19	4	26	3	1	2	29	13	9	0	6	7

สุ่ม X(t) **ซึ่งเป็นการสุ่มลำดับการ**จัดตารางการผลิต โดยผ่านการ Repair part และ Repair operation ถูกต้องแล้ว

20	28	21	29	13	12	24	8	1	14	26	16	2	31	9	10	15	7	22	32	33	30	19	3	23	17	0	25	27	11	6	18	4	5

เมื่อเปรียบเทียบตารางการผลิตของ X(t) และ X(t+1) จะได้รายการที่ต้องปรับปรุง ซึ่งเป็น V(t+1)

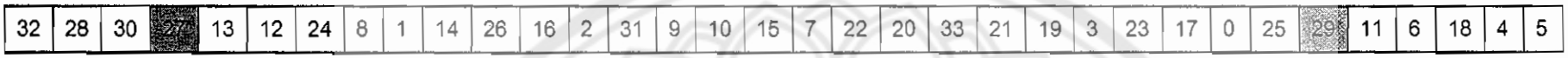
X(t) **ที่ต้องปรับปรุงครั้งที่ 1 ตำแหน่ง**ที่ (1, 20) จะได้ค่า 20 สลับกับค่า 32 ดังนี้

32	28	21	29	13	12	24	8	1	14	26	16	2	31	9	10	15	7	22	20	33	30	19	3	23	17	0	25	27	11	6	18	4	5

X(t) **ที่ต้องปรับปรุงครั้งที่ 2 ตำแหน่ง**ที่ (3, 22) จะได้ค่า 21 สลับกับค่า 30 ดังนี้

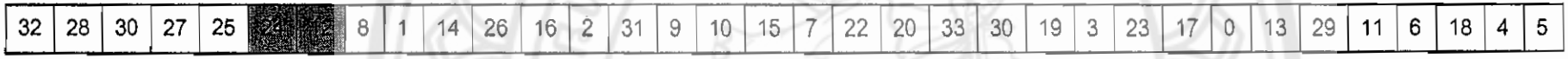
32	28	30	29	13	12	24	8	1	14	26	16	2	31	9	10	15	7	22	20	33	21	19	3	23	17	0	25	27	11	6	18	4	5

X(t) ที่ต้องปรับปรุงครั้งที่ 3 ตำแหน่งที่ (4, 29) จะได้ค่า 29 สลับกับค่า 27 ดังนี้



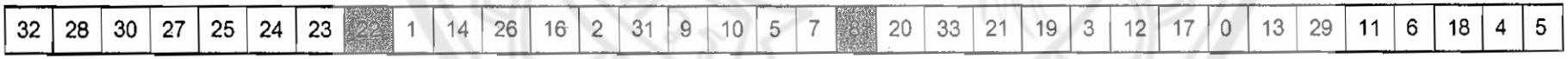
X(t) ที่ต้องปรับปรุงครั้งที่ 4 ตำแหน่งที่ (5, 28) จะได้ค่า 13 สลับกับค่า 25 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 5 ตำแหน่งที่ (6, 7) จะได้ค่า 12 สลับกับค่า 24 ดังนี้



X(t) ที่ต้องปรับปรุงครั้งที่ 6 ตำแหน่งที่ (7, 25) จะได้ค่า 12 สลับกับค่า 23 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 7 ตำแหน่งที่ (8, 19) จะได้ค่า 8 สลับกับค่า 22 ดังนี้



X(t) ที่ต้องปรับปรุงครั้งที่ 8 ตำแหน่งที่ (9, 22) จะได้ค่า 1 สลับกับค่า 21 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 9 ตำแหน่งที่ (11, 20) จะได้ค่า 26 สลับกับค่า 20 ดังนี้

32	28	30	27	25	24	23	22	21	17	20	16	2	31	9	10	15	7	8	26	33	1	19	3	12	14	0	13	29	11	6	18	4	5

X(t) ที่ต้องปรับปรุงครั้งที่ 10 ตำแหน่งที่ (13, 34) จะได้ค่า 2 สลับกับค่า 5 ดังนี้

32	28	30	27	25	24	23	22	21	17	20	16	5	31	9	10	15	7	8	26	33	1	19	3	12	14	0	13	29	11	6	18	4	2

X(t) ที่ต้องปรับปรุงครั้งที่ 11 ตำแหน่งที่ (14, 32) จะได้ค่า 31 สลับกับค่า 18 ดังนี้

32	28	30	27	25	24	23	22	21	17	20	16	5	18	9	10	15	7	8	26	33	1	19	3	12	14	0	13	29	11	6	31	4	2

X(t) ที่ต้องปรับปรุงครั้งที่ 12 ตำแหน่งที่ (15, 23) จะได้ค่า 19 สลับกับค่า 15 ดังนี้

32	28	30	27	25	24	23	22	21	17	20	16	5	18	15	10	9	7	8	26	33	1	19	3	12	14	0	13	29	11	6	31	4	2

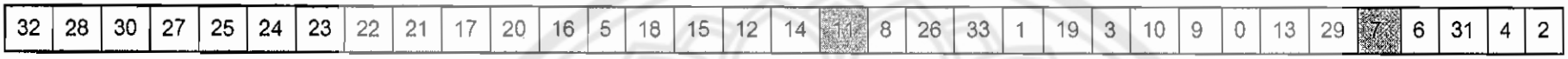
X(t) ที่ต้องปรับปรุงครั้งที่ 13 ตำแหน่งที่ (16, 25) จะได้ค่า 10 สลับกับค่า 12 ดังนี้

32	28	30	28	25	24	23	22	21	17	20	16	5	18	15	12	9	7	8	26	33	1	19	3	10	14	0	13	29	11	6	31	4	2

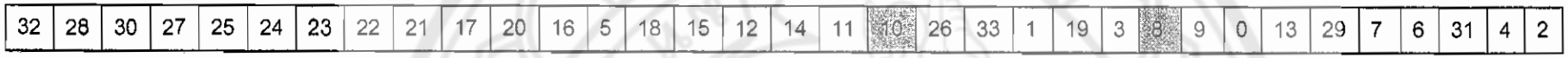
X(t) ที่ต้องปรับปรุงครั้งที่ 14 ตำแหน่งที่ (17, 26) จะได้ค่า 9 สลับกับค่า 14 ดังนี้

32	28	30	27	25	24	23	22	21	17	20	16	5	18	15	12	14	7	8	26	33	1	19	3	10	9	0	13	29	11	6	31	4	2

X(t) ที่ต้องปรับปรุงครั้งที่ 15 ตำแหน่งที่ (18, 30) จะได้ค่า 7 สลับกับค่า 11 ดังนี้



X(t) ที่ต้องปรับปรุงครั้งที่ 16 ตำแหน่งที่ (19, 25) จะได้ค่า 8 สลับกับค่า 10 ดังนี้



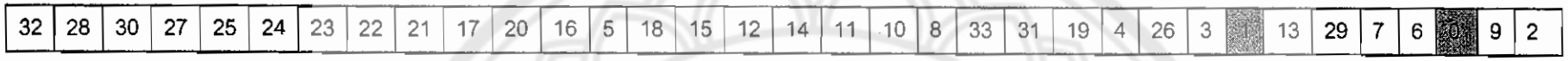
X(t) ที่ต้องปรับปรุงครั้งที่ 17 ตำแหน่งที่ (20, 25) จะได้ค่า 26 สลับกับค่า 8 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 18 ตำแหน่งที่ (22, 32) จะได้ค่า 1 สลับกับค่า 31 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 19 ตำแหน่งที่ (24, 33) จะได้ค่า 3 สลับกับค่า 4 ดังนี้

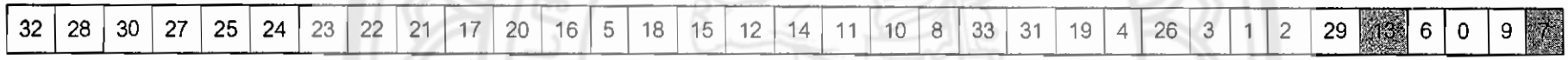
X(t) ที่ต้องปรับปรุงครั้งที่ 20 ตำแหน่งที่ (26, 33) จะได้ค่า 9 สลับกับค่า 3 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 21 ตำแหน่งที่ (27, 32) จะได้ค่า 0 สลับกับค่า 1 ดังนี้



X(t) ที่ต้องปรับปรุงครั้งที่ 22 ตำแหน่งที่ (28, 34) จะได้ค่า 2 สลับกับค่า 13 ดังนี้

X(t) ที่ต้องปรับปรุงครั้งที่ 23 ตำแหน่งที่ (31, 34) จะได้ค่า 7 สลับกับค่า 13 ดังนี้



X(t) ที่ต้องปรับปรุงครั้งที่ 24 ตำแหน่งที่ (31, 33) จะได้ค่า 6 สลับกับค่า 9 ดังนี้

ดังนั้น V(t+1) ซึ่งเป็นค่าอัตราเร็วของพาร์ทิเคิลที่จะใช้ในการปรับปรุงพาร์ทิเคิล ในรอบถัดไปจะถูกบันทึกเป็นลำดับของตำแหน่งที่ต้องการสลับ เป็นการแสดงความแตกต่างของการจัดตารางการผลิต มีค่าเท่ากับ {(1, 20), (3, 22), (4, 29), (5, 28), (6, 7), (7,25), (8,19), (9, 22), (11, 20), (13, 34), (14, 32), (15, 23), (16, 25), (17, 26), (18, 30), (19, 25), (20, 25), (22, 32), (24, 33), (26, 33), (27, 32), (28, 34), (30, 34), (31, 33)} ซึ่งเป็นความแตกต่างของการจัดตารางการผลิตแบบเดิม (X(t)) และการจัดตารางการผลิตแบบใหม่ (X(t+1))

เมื่อทำการสลับตารางการผลิตเดิม (X(t)) ตามคู่ลำดับที่กำหนดตำแหน่งที่ต้องการสลับไว้แล้ว จะได้ผลลัพธ์เป็นตารางการผลิต**ใหม่** (X(t+1))

X(t) **ที่ต้องปรบปรุงทั้งหมด** 24 ครั้ง ตามรายการ V(t+1) ซึ่งจะได้ตารางการผลิต (X(t+1))

32	28	30	27	25	24	23	22	21	17	20	16	5	18	15	12	14	11	10	8	33	31	19	4	26	3	1	2	29	13	9	0	6	7

X(t+1)

32	28	30	27	25	24	23	22	21	17	20	16	5	18	15	12	14	11	10	8	33	31	19	4	26	3	1	2	29	13	9	0	6	7

6. ทำการวนซ้ำจนกระทั่งพบค่าคำตอบที่เหมาะสมที่สุด หรือเมื่อมีจำนวนรอบของการทำซ้ำสูงสุดจึงจบการทำงานแล้วออกจาก Loop ไป

ปัจจัยที่จะนำมาทำการทดสอบเพื่อศึกษาการทำงานของ PSO จะต้องมีการกำหนดค่าปัจจัยต่าง ๆ ที่สำคัญได้แก่

6.1 จำนวนของพาร์ทิเคิล/จำนวนรอบ (Number of Particles /Iteration: N/I) จากงานวิจัยของ Shi & Eberhart (1999) เลือกที่จะทดสอบประสิทธิภาพของ PSO โดยเปรียบเทียบค่าคำตอบที่ได้จาก Non-linear function 4 แบบ โดยกำหนด Population size ขนาด 20, 40, 80 และ 160 กำหนด จำนวนรอบ 50 รอบ Eberhart & Shi (2000) และ Carlisle & Dozier (2001) ได้ใช้ปัญหา Non-linear function 5 แบบ โดยกำหนด Population size ขนาด 30 แต่กำหนดจำนวนรอบ 20 รอบ และ Clerc & Kennedy (2002) ใช้ปัญหา Non-linear function 8 แบบ โดยกำหนด Population size ขนาด 20 แต่กำหนดจำนวนรอบ 20 รอบ Zhang, Yu & Hu. (2005) ใช้ปัญหา Non-linear function 9 แบบ โดยกำหนด Population size ซึ่งมีค่า [20, 50] และพบว่าควรกำหนดเป็น 30 จะได้ค่าคำตอบที่มีประสิทธิภาพที่สุด Elbeltagi, Hegazy & Grierson. (2005) ได้ทำการเปรียบเทียบอัลกอริทึม 5 แบบ (Genetic Algorithms: GA, Memetic Algorithms: MA, Particle Swarm Optimization: PSO, Ant Colony Optimization: ACO & Shuffled Frog Leaping: SFL) ด้วยสมการ Non-linear, Non-separable โดยกำหนด Population size ขนาด 40 จะเห็นได้ว่าในแต่ละงานวิจัยที่กล่าวมาแล้ว มีการกำหนดค่าที่แตกต่างกันไปตามแต่ละปัญหา ดังนั้นผู้วิจัยจึงทำการรันทดสอบแล้วพบว่าการกำหนดค่าจำนวนของพาร์ทิเคิล/จำนวนรอบ (N/I) ให้มีจำนวนของการค้นหาทั้งหมด 400 ครั้ง (10/40, 20/20 และ40/10) ซึ่งเพียงพอต่อการหาค่าคำตอบที่เหมาะสมแล้ว

6.2 ค่าถ่วงน้ำหนัก (Inertia weight: ω) ที่กำหนดในแต่ละงานวิจัยดังมีรายละเอียดดังต่อไปนี้ Shi & Eberhart (1999) กำหนด ω เท่ากับ [0.4, 0.9] Eberhart & Shi (2000) และ Zhang, Yu & Hu. (2005) กำหนด ω เท่ากับ [0.01, 1] ซึ่งพบว่าต้องเลือกให้เหมาะสมกับปัญหาที่นำมาทดสอบ Elbeltagi, Hegazy & Grierson. (2005) กำหนด ω เป็นค่าที่อยู่ระหว่าง [0,1.4] ซึ่งจากงานวิจัยที่สรุปมา จะเห็นได้ว่าการกำหนดค่า ω สำหรับงานวิจัยต่างๆ จะกำหนดในช่วง [0, 1.2] ดังนั้นจะกำหนดค่า ω ที่ใช้ในการทดสอบจะกำหนดไว้เป็น 3 ระดับคือ 0.1, 0.65 และ 1.20 ตามลำดับ

6.3 ปัจจัยการเรียนรู้ตนเอง/สังคม (Cognition learning rate/social learning rate: c_1/c_2) เป็นค่าคงที่ 2 ค่าที่ได้จากการสุ่ม ซึ่งมีค่า [0, 1] Shi & Eberhart (1999)

และ Elbeltagi, Hegazy & Grierson. (2005) กำหนดให้ค่า c_1 และ c_2 มีค่าเท่ากับ 2 กำหนด Eberhart & Shi (2000) กำหนด c_1 และ c_2 เท่ากับ 1.49445 Carlisle & Dozier (2001) กำหนด c_1 และ c_2 เท่ากับ 2.05 Zhang, Yu & Hu. (2005) ได้ทดสอบแล้วแนะนำให้ค่า c_1/c_2 ควรเป็น 2.8/1.3 ซึ่งเมื่อนำค่า c_1บวก c_2 เท่ากับ 4.1

3. การออกแบบการทดลองและวิเคราะห์ข้อมูลทางสถิติ

ในการออกแบบการทดลองนั้นผู้วิจัยต้องการหาข้อสรุปที่มีความหมายจากข้อมูลที่มีอยู่ จากปัญหาจัดตารางการผลิตที่สนใจ ต้องการทำการทดลอง เพื่อให้เป็นไปตามเป้าหมายที่ผู้วิจัยกำหนดไว้ จะมีวิธีการทางสถิติเข้ามาช่วยในการวางแผนการทดลอง โดยใช้การออกแบบการทดลองเชิงสถิติ (Statistical design of experiment) ซึ่งเป็นกระบวนการสำหรับวางแผนการทดลอง เพื่อให้ได้ข้อมูลที่เหมาะสมที่จะสามารถใช้วิเคราะห์ผลการทดลองด้วยวิธีการทางสถิติ ซึ่งจะทำให้สามารถสรุปผลการทดลองได้อย่างถูกต้อง ตรงตามวัตถุประสงค์ของงานวิจัยนี้ได้นั้น สิ่งสำคัญที่ควรพิจารณาสำหรับการทดลองในการใช้แก้ปัญหาการจัดตารางมี 2 ประการคือ การออกแบบการทดลอง และการวิเคราะห์ข้อมูลทางสถิติ ซึ่งมีความเกี่ยวข้องสัมพันธ์กันเป็นอย่างมาก ทั้งนี้เพราะการเลือกใช้สถิติที่จะนำมาใช้วิเคราะห์ข้อมูลปัญหาที่สนใจได้อย่างเหมาะสม จะต้องพิจารณาให้เหมาะสมกับการทดลองที่ออกแบบไว้ (Montgomery, 1997) ซึ่งสามารถแบ่งออกเป็น 2 แบบดังนี้

1. การออกแบบการทดลองทางสถิติ (Statistical design of experiment)
2. การวิเคราะห์ข้อมูลทางสถิติ (The statistical analysis of the data)

3.1 การออกแบบการทดลองทางสถิติ

การออกแบบการทดลองทางสถิติ (Statistical design of experiment) หมายถึง กระบวนการในการวางแผนการทดลอง เพื่อให้ได้ข้อมูลที่เหมาะสม นำไปวิเคราะห์ โดยวิธีการทางสถิติ ซึ่งจะทำให้หาข้อสรุปที่สมเหตุผลได้ ดังนั้นการออกแบบการทดลองเชิงสถิติจึงจำเป็นต้องปฏิบัติ เพื่อให้ได้ข้อสรุปที่มีความหมายจากข้อมูลที่มีอยู่ โดยเลือกใช้หลักการของการเรพลิเคชัน (Replicaton) ซึ่งเป็นการทดลองซ้ำ ทำให้สามารถหาค่าประมาณของความผิดพลาดที่อาจจะเกิดขึ้นจากการทดลอง ในเบื้องต้นได้ ใช้วัดว่าข้อมูลที่ใช้ในการทดลองนั้น มีความแตกต่างกันในเชิงสถิติหรือไม่ เมื่อนำมาวิเคราะห์ค่าความแปรปรวนของข้อมูลแต่ละตัวแปร ที่ต้องการเปรียบเทียบ ค่าของการตอบสนองที่ได้ ในแต่ละระดับของตัวแปร เพื่อหาข้อสรุปทางสถิติต่อไป

การทดลอง (Experiment) คือ การทดสอบ (Test) หรือชุดของการทดสอบที่มีการเปลี่ยนแปลงกับตัวแปรนำเข้า (Input variable) ของกระบวนการหรือระบบ เพื่อสังเกต หาเหตุบ่งชี้ถึงการเปลี่ยนแปลงที่เกิดขึ้น กับผลตอบขาออก (Output response) ของระบบได้

ส่วนใหญ่ในการทำการทดลองมักจะเกี่ยวข้องกับตัวแปรต่าง ๆ หลายตัวที่มีผลกระทบกับระบบ ขึ้นอยู่กับวัตถุประสงค์ที่ผู้วิจัยต้องการจะศึกษา เพื่อหาค่าของคำตอบที่ต้องการทราบตามลักษณะของปัญหาแบบต่าง ๆ ที่ผู้วิจัยสนใจศึกษา ในการหาผลกระทบของปัจจัย และผลลัพธ์ที่ได้จากการตอบสนองของระบบ ซึ่งเรียกการวางแผนและดำเนินการทดลองว่า กลยุทธ์ของการทดลอง (Strategy of experimentation) (Montgomery, 1997)

มีกลยุทธ์หลายแบบที่ผู้วิจัยนำมาทดลองใช้ เพื่อวางแผน ออกแบบการทดลองและดำเนินการทดลองซึ่งจะต้องเลือกใช้กลยุทธ์ที่เหมาะสมกับงานวิจัยแตกต่างกันไป ดังนั้นผู้วิจัยจะกล่าวถึงกลยุทธ์ที่นำมาใช้ในการศึกษาแบ่งเป็นข้อย่อยต่างๆ ดังนี้

1. การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ (2^k Factorial designs)
2. การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ (Two-level fractional factorial designs)
3. การออกแบบการทดลองเชิงแฟกทอเรียลแบบสามระดับ (3^k Factorial designs)

1. การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ (2^k Factorial designs)

การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ หมายถึง การออกแบบในกรณีที่มีปัจจัย k ปัจจัย ซึ่งแต่ละปัจจัยประกอบด้วย 2 ระดับ แทนด้วยระดับสูง และระดับต่ำของปัจจัยหนึ่ง ๆ ในการทดลองทำซ้ำ 1 ครั้ง (Replicate) ที่สมบูรณ์สำหรับการออกแบบข้อมูลที่ประกอบด้วย $2 \times 2 \times 2 \times \ldots \times 2 = 2^k$ ข้อมูล การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ ถูกนำมาใช้อย่างแพร่หลาย เพื่อคัดกรองปัจจัยที่มีอยู่จำนวนมากให้เหลือน้อยลง ซึ่งเป็นประโยชน์อย่างมากต่อการทดลองในช่วงเริ่มแรกที่มีจำนวนปัจจัยมากที่ผู้วิจัยต้องการจะตรวจสอบ ซึ่งการออกแบบเช่นนี้จะทำให้จำนวนการทดลองน้อยที่สุดที่จะทำได้ เพื่อศึกษาถึงผลของปัจจัยทั้ง k ชนิดได้อย่างสมบูรณ์

2. **การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ** (Two-level fractional factorial designs)

การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ จัดว่าเป็นการออกแบบที่มีการใช้งานอย่างแพร่หลายในการออกแบบกระบวนการ เพื่อช่วยหาแนวทางในการปรับปรุงกระบวนการให้มีประสิทธิภาพเพิ่มมากขึ้น ดังนั้นเมื่อจำนวนปัจจัยในการออกแบบ 2^k เพิ่มขึ้น ทำให้ในการทดลองทำซ้ำ 1 ครั้ง (Replicate) มีความสมบูรณ์เพิ่มขึ้นมากเกินกว่าที่ทรัพยากรที่มีอยู่จะรองรับได้ ดังนั้นการออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ จึงถูกนำมาใช้ในการกรองปัญหาเพื่อหาปัจจัยที่มีผล (Screening experiment) กล่าวคือ ในการทดลองหนึ่ง ๆ อาจจะมีปัจจัยมากมายที่ผู้วิจัยเกิดความสนใจ สงสัยต้องการทราบค่าคำตอบ ทำให้ผู้วิจัยต้องใช้การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ ค้นหาตัวแปรหรือปัจจัยใดบ้างที่เป็นปัจจัยที่มีนัยสำคัญ โดยใช้การออกแบบเศษส่วนแบบ 2^{k-1} (The one-half fraction of the 2^k design) ซึ่งเป็นการออกแบบครึ่งหนึ่ง (1/2) ของการออกแบบ 2^k หรือหากการทดลองมีปัจจัยที่เพิ่มมากขึ้นอีกก็อาจจะใช้การออกแบบเศษส่วน แบบ 2^{k-2} (The one-quarter fraction of the 2^k design) ซึ่งเป็นการออกแบบครึ่งหนึ่ง (1/4) ของการทดลองแบบ 2^k ก็ได้

3. **การออกแบบการทดลองเชิงแฟกทอเรียลแบบสามระดับ** (3^k Factorial designs)

การออกแบบการทดลองเชิงแฟกทอเรียลแบบสามระดับ หมายถึง การพิจารณาปัจจัยทั้ง k ปัจจัย ซึ่งปัจจัยทุกตัวมี 3 ระดับ คือ ระดับต่ำ ระดับปานกลาง และระดับสูง อาจจะใช้สัญลักษณ์แทนเป็นตัวเลข 0, 1, 2 หรือ –1, 0 ,1 ก็ได้ ทำให้ในการทดลองทำซ้ำ 1 ครั้ง (Replicate) มีความสมบูรณ์สำหรับการออกแบบข้อมูลที่ ประกอบด้วย $3 \times 3 \times 3 \times ... \times 3 = 3^k$ ข้อมูล จะแทนตัวเลข k ตัว ซึ่งตัวเลขตัวแรกแทนระดับของปัจจัย A ตัวเลขตัวสองแทนระดับของปัจจัย B และตัวเลขที่ k แทนระดับของปัจจัย k (Montgomery, 2001)

จากที่ผู้วิจัยได้เก็บรวบรวมข้อมูล และศึกษาปัจจัยที่เกี่ยวข้อง ซึ่งได้มีการออกแบบการทดลอง เพื่อศึกษาปัจจัย 3 ปัจจัย ประกอบด้วย จำนวนของพาร์ทิเคิล /จำนวนรอบ (Number of particle /Iteration: N/I), ค่าถ่วงน้ำหนัก (ω) และปัจจัยในการเรียนรู้ตนเองและสังคมของพาร์ทิเคิล (c_1และc_2) ซึ่งในทุกปัจจัยมี 3 ระดับคือ ระดับต่ำ (0), ระดับปานกลาง (1) และระดับสูง (2) มีการทดลองทำซ้ำทั้งหมด 27 ครั้งต่อการเลือกสุ่ม (Random seed) 1 ครั้ง เมื่อได้ผลการทดลองโดยรันจนครบ 30 seed แล้วนำข้อมูลที่ได้รับจะต้องนำไปวิเคราะห์ทางสถิติ

ต่อไป เพื่ออธิบายผลการทดลองที่เกิดขึ้น ว่ามีความหมายอย่างไร ด้วยการวิเคราะห์ข้อมูลทางสถิติในหัวข้อต่อไป

3.2 การวิเคราะห์ข้อมูลทางสถิติ (The statistical analysis of the data)

สถิติเชิงพรรณา (Descriptive statistics) เป็นสถิติที่ใช้สรุปลักษณะของกลุ่มข้อมูล โดยที่ข้อมูลที่เก็บรวบรวมจากตัวอย่าง หรือประชากร สถิติเชิงพรรณา ได้แก่ จำนวน ร้อยละ ค่าเฉลี่ย ค่าเบี่ยงเบนมาตราฐาน ค่าพิสัย สถิติเชิงพรรณาที่ใช้ในงานวิจัยนี้เป็นสถิติเชิงพรรณาสำหรับข้อมูลเชิงปริมาณ ข้อมูลเชิงปริมาณ หมายถึง สเกลแบ่งช่วง และสเกลอัตราส่วน (Interval and ratio scale) ซึ่งเป็นข้อมูลที่ต้องคำนวณค่าสถิติ เพื่อสรุปลักษณะข้อมูล โดยจะต้องหาค่าสถิติที่แสดงค่ากลางและการกระจายข้อมูล

ค่ากลางสำหรับข้อมูลเชิงปริมาณ ประกอบด้วย

1. ค่าเฉลี่ย (Average) เป็นค่ากลางที่นิยมใช้กันมากที่สุด ใช้ได้กับข้อมูลสเกลแบบช่วงและสเกลแบบอัตราส่วน
2. ค่ามัธยฐาน (Median) เป็นค่าของตำแหน่งที่อยู่ตรงกลางของข้อมูล
3. ค่าฐานนิยม (Mode) ของข้อมูลชุดหนึ่งคือ ค่าของข้อมูลที่เกิดขึ้นบ่อยที่สุด

ค่าการกระจายของข้อมูลเชิงปริมาณ นอกจากจะต้องสรุปลักษณะของข้อมูลด้วยค่ากลาง ซึ่งแสดงถึงตำแหน่งของข้อมูลชุดนั้นแล้ว ควรจะต้องสรุปค่าการกระจายของข้อมูลด้วย เพื่อแสดงค่าความแตกต่างของข้อมูล ถ้าข้อมูลมีความแตกต่างกันมาก ก็จะมีการกระจายมากด้วย การวัดค่าการกระจายทางสถิติที่ใช้ทั่วไปคือ

4. ค่าพิสัย (Range) เป็นการหาค่าผลต่างของค่าสูงสุด และค่าต่ำสุด
5. ค่าแปรแปรวน (Variance) เป็นค่าที่ใช้วัดการกระจายของข้อมูลที่นิยมกันมาก พิจารณาจากค่าเฉลี่ยของความแตกต่างระหว่างข้อมูลแต่ละค่ากับค่าเฉลี่ยยกกำลังสอง ถ้าค่าแปรปรวนมีค่ามาก แสดงว่าข้อมูลนั้นมีการกระจายมากด้วย
6. ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation: SD) เป็นค่าฐานที่สองของค่าแปรปรวน

การวิเคราะห์ข้อมูลทางสถิติเป็นขั้นตอนที่ทำได้ง่าย โดยใช้โปรแกรมประยุกต์ทางด้านสถิติ มาช่วยวิเคราะห์ทางสถิติได้ เช่น Minitab 13.20 หรือ SPSS 12 และควรตรวจสอบส่วนตกค้าง (Residual) เพื่อความถูกต้องของแบบจำลองที่ได้ออกแบบการทดลองมาแล้วนั้น ได้แก่ การตรวจสอบสมมุติฐานของความปกติ (Normal probability plot of the residuals) ใช้ตรวจสอบความผิดพลาดของการแจกแจงว่าเป็นปกติหรือไม่ ถ้าหากการแจกแจงของความผิดพลาดเป็นปกติ

รูปกราฟที่พล็อตขึ้นจะมีลักษณะเป็นเส้นตรง การพล็อตส่วนตกค้างตามลำดับเวลา (Residuals versus the order of the data) เป็นการพล็อตส่วนตกค้างตามลำดับเวลาของการเก็บข้อมูล ใช้ตรวจสอบความเกี่ยวพันระหว่างส่วนตกค้าง หากข้อมูลที่ได้จากการรัน มีแนวโน้มเป็นค่าบวกหรือค่าลบ จะบ่งชี้ว่าสมมุติฐานนั้นถูกละเมิดแล้ว การพล็อตส่วนตกค้างกับค่าที่ถูกฟิต (Residuals versus the fitted values) เป็นการพล็อตความสัมพันธ์ของค่าตัวแปรต่าง ๆ รวมถึงค่าของผลตอบ (Predicted response) ด้วย ซึ่งถ้าหากแบบจำลองถูกต้องและมีสมมุติฐานที่เหมาะสมแล้ว รูปกราฟที่ได้จากการพล็อตนี้ไม่ควรมีรูปแบบหรือโครงสร้างใด ๆ ทั้งสิ้น

หลังจากที่ได้เริ่มทำการทดลอง นำข้อมูลมาวิเคราะห์ทางสถิติ และตรวจสอบความถูกต้องของแบบจำลอง จนกระทั่งได้ค่าตัวแปรและสมมุติฐานที่เหมาะสมแล้ว จึงนำไปหาข้อสรุปของปัญหาที่ได้ทำการศึกษาต่อไป สำหรับในงานวิจัยนี้จะกล่าวถึงเฉพาะการวิเคราะห์ข้อมูลทางสถิติที่ใช้ในการศึกษาครั้งนี้ ดังต่อไปนี้

การวิเคราะห์ความแปรปรวน (Analysis of variance: ANOVA)

ในการวิเคราะห์ความแปรปรวน เป็นการวิเคราะห์อัตราส่วนระหว่างความแปรปรวนระหว่างกลุ่ม (Between-group variance) และความแปรปรวนภายในกลุ่ม (Within-group variance) ความแปรปรวนระหว่างกลุ่มเป็นค่าที่เกิดจากความแตกต่างของค่าเฉลี่ยระหว่างกลุ่มต่างๆ ถ้าค่าเฉลี่ยระหว่างกลุ่มต่างๆแตกต่างกันมาก ค่าความแปรปรวนระหว่างกลุ่มก็จะมากตามไปด้วย สำหรับความแปรปรวนภายในกลุ่มเป็นค่าที่แสดงให้เห็นว่า คะแนนแต่ละตัวที่รวบรวมมานั้นภายในแต่ละกลุ่มมีการกระจายมากหรือน้อย ค่าที่คำนวณได้เรียกว่า ความคลาดเคลื่อน (ชูศรี วงศ์รัตนะ, 2534)

การวิเคราะห์ความแปรปรวนโดยทั่วไปจะใช้เพื่อวิเคราะห์ผลจากการทดลองเชิงแฟกทอเรียล ตัวอย่างเช่น การทดลองเชิงแฟกทอเรียลในกรณีที่มีปัจจัยที่จะทำการศึกษา 2 ปัจจัย คือ A และ B โดยปัจจัย A ประกอบด้วย a ระดับ และปัจจัย B ประกอบด้วย b ระดับ ซึ่งทั้งหมดนี้ถูกจัดให้อยู่ในรูปของการออกแบบเชิงแฟกทอเรียลนั่นคือ ในแต่ละเรพลิเคตของการทดลองจะประกอบด้วยการทดลองร่วมปัจจัยทั้งหมด ab การทดลอง โดยปกติจะมีจำนวนเรพลิเคตทั้งหมด n ครั้ง รูปแบบทั่วไปของการออกแบบเชิงแฟกทอเรียล 2 ปัจจัยและมีการทำซ้ำทั้งหมด n ครั้ง สามารถแสดงได้ดังตาราง 2 เมื่อกำหนดให้ y_{ijk} คือ ผลตอบสนองที่เกิดจากระดับที่ i ของปัจจัย A (เมื่อ i = 1, 2, ...,a) และระดับที่ j ของปัจจัย B (เมื่อ j = 1, 2, ..., b) สำหรับเรพลิเคตที่ k (เมื่อ k = 1, 2, ..., n) (Montgomery, 1997)

ตาราง 2 แสดงรูปแบบข้อมูลการทดลองเชิงแฟกทอเรียลเมื่อมี 2 ปัจจัย

		Factor B			
		1	2	...	b
Factor A	1	$y_{111}, y_{112}, \cdots, y_{11n}$	$y_{121}, y_{122}, \cdots, y_{12n}$	...	$y_{1b1}, y_{1b2}, \cdots, y_{1bn}$
	2	$y_{211}, y_{212}, \cdots, y_{21n}$	$y_{221}, y_{222}, \cdots, y_{22n}$	...	$y_{2b1}, y_{2b2}, \cdots, y_{2bn}$
	⋮	⋮	⋮	⋮	⋮
	A	$y_{a11}, y_{a12}, \cdots, y_{a1n}$	$y_{a21}, y_{a22}, \cdots, y_{a2n}$	...	$y_{ab1}, y_{ab2}, \cdots, y_{abn}$

ข้อมูลจากตาราง 2 อาจจะเขียนในรูปของแบบจำลองสถิติเชิงเส้น (Linear statistical model) ได้ดังนี้คือ

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \quad \text{..... (15)}$$

เมื่อ $i = 1, 2, \ldots, a$

$j = 1, 2, \ldots, b$

$k = 1, 2, \ldots, n$

โดยที่ y_{ijk} หมายถึง ผลตอบสนองที่สังเกตได้เมื่อปัจจัย A อยู่ที่ระดับ i และปัจจัย B อยู่ที่ระดับ j สำหรับเรพลิเคตที่ k,

μ หมายถึง ค่าเฉลี่ยทั้งหมด,

τ_i หมายถึง ผลที่เกิดจากระดับที่ i ของแถว (Row) ของปัจจัย A,

β_j หมายถึง ผลที่เกิดจากระดับที่ j ของคอลัมน์ (Column) ของปัจจัย B,

$(\tau\beta)_{ij}$ หมายถึง ผลที่เกิดจากปฏิสัมพันธ์ระหว่าง τ_i กับ β_j,

ε_{ijk} หมายถึง องค์ประกอบของความผิดพลาดแบบสุ่ม

เนื่องจากในการทดลองนี้มีจำนวนเรพลิเคต n ครั้ง ดังนั้นจำนวนข้อมูลที่ได้จากการสังเกตทั้งหมดจึงเท่ากับ abn จำนวน (Montgomery, 1997)

การตรวจสอบความถูกต้องของสมมติฐานตามแบบจำลองสมการ (15) ความคลาดเคลื่อน หรือองค์ประกอบของความผิดพลาดแบบสุ่ม (ε_{ijk}) จะต้องมีการแจกแจงแบบปกติ และเป็นอิสระด้วยค่าเฉลี่ยเท่ากับ 0 และค่าความแปรปรวน σ^2 มีค่าคงตัวแต่ไม่ทราบค่า ถ้าสมมติฐาน

เหล่านี้เป็นจริงกระบวนการวิเคราะห์ความแปรปรวนนี้ก็จะเป็นการทดสอบสมมติฐานเกี่ยวกับการไม่มีความแตกต่างในค่าเฉลี่ยของระดับที่ถูกต้อง ในทางปฏิบัติการจะเชื่อในผลที่ได้จากการวิเคราะห์ความแปรปรวนนั้นจะต้องตรวจสอบความถูกต้องของสมมติฐานว่าเป็นจริงก่อน ซึ่งการตรวจสอบสมมติฐานขั้นต้นและความถูกต้องของแบบจำลองทำได้โดยการวิเคราะห์ส่วนตกค้าง (Residual analysis) ซึ่งจะมีการตรวจสอบสมมติฐานของความเป็นปกติโดยการทำกราฟความน่าจะเป็นแบบปกติของส่วนตกค้าง (Normal probability plot of the residuals) ถ้าหากการแจกแจงของความผิดพลาดเป็นแบบปกติรูปกราฟที่ได้จะเป็นเส้นตรง, ทำการตรวจสอบสมมติฐานค่าความแปรปรวน σ^2 มีค่าคงตัวโดยการทำกราฟส่วนตกค้างกับค่าที่ถูกฟิต (Residuals versus the fitted values) ถ้าหากแบบจำลองถูกต้องและสมมติฐานมีความเหมาะสมแล้วกราฟที่ได้ไม่ควรจะมีรูปแบบหรือรูปร่างเฉพาะแต่อย่างใดทั้งสิ้น และสุดท้ายคือทำการตรวจสอบสมมติฐานของความเป็นอิสระโดยการทำกราฟส่วนตกค้างตามลำดับเวลาของการเก็บข้อมูล (Residuals versus the order of the data) ถ้าส่วนตกค้างมีแนวโน้มที่จะเพิ่มขึ้นหรือลดลงตามลำดับเวลาของการเก็บข้อมูล หรือกราฟที่ได้มีการกระจายที่ปลายข้างหนึ่งมากกว่าที่ปลายอีกข้างหนึ่ง แสดงว่าสมมติฐานของความเป็นอิสระถูกละเมิด จากสมการ (15) ถ้าสมมติว่าแบบจำลองตามสมการ (15) เป็นแบบจำลองที่เหมาะสม และพจน์ของความผิดพลาด ε_{ijk} มีการกระจายแบบปกติและเป็นอิสระ โดยมีค่าความแปรปรวนคงตัวเท่ากับ σ^2 วิธีการทดสอบจะทำโดยอาศัยตารางการวิเคราะห์ความแปรปรวน (ตาราง 3) ซึ่งโดยทั่วไปตาราง ANOVA จะประกอบด้วย แหล่งความแปรปรวน (Source of variation), ผลรวมกำลังสอง (Sum of square: SS), องศาแห่งความอิสระ (Degrees of freedom: DF), ค่าเฉลี่ยกำลังสอง (Mean squares: MS) และ ค่า F (F value)

ตาราง 3 ตารางการวิเคราะห์ความแปรปรวนของการทดลองเชิงแฟกทอเรียล 2 ตัวแปร แบบ Fixed effects model

Source of Variation	Sum of Square	DF	Mean Square	F
ปัจจัย A	$SS_A = \frac{1}{bn}\sum_{i=1}^{a} y_{i..}^2 - \frac{y_{...}^2}{abn}$	a-1	$MS_A = \frac{SS_A}{a-1}$	$\frac{MS_A}{MS_E}$
ปัจจัย B	$SS_B = \frac{1}{an}\sum_{j=1}^{b} y_{j.}^2 - \frac{y_{...}^2}{abn}$	b-1	$MS_B = \frac{SS_B}{b-1}$	$\frac{MS_B}{MS_E}$
AB	$SS_{AB} = \frac{1}{n}\sum_{i=1}^{a}\sum_{j=1}^{b} y_{ij}^2 - \frac{y_{...}^2}{abn} - SS_A - SS_B$	(a-1)(b-1)	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_E}$

ตาราง 3 ตารางการวิเคราะห์ความแปรปรวนของการทดลองเชิงแฟกทอเรียล 2 ตัวแปร แบบ Fixed effects model

Source of Variation	Sum of Square	DF	Mean Square	F
ความคลาดเคลื่อน	$SS_E = SS_T - SS_A - SS_B - SS_{AB}$	ab(n-1)	$MS_E = \frac{SS_E}{ab(n-1)}$	
ผลรวม	$SS_T = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}^2 - \frac{y_{...}^2}{abn}$	abn-1		

4 **การพัฒนา และเลือกใช้ภาษา Tcl/Tk**

เนื่องจากภาษา Tcl มีลักษณะพิเศษโดยแสดงผลแบบกราฟฟิกได้ดี มีความรวดเร็วในการพัฒนาแอพพิเคชั่น ซึ่งมีการประมวลผลรวดเร็วกว่าแอพพิเคชั่นที่สร้างด้วยภาษาอื่น ๆ ตลอดจน Tcl/Tk มีเครื่องมือช่วยในการสร้างแอพพิเคชั่นในลักษณะของ GUI ได้ง่าย เช่น TK canvas จะเป็นคำสั่งง่าย ๆ ในการสร้างหน้าจอกราฟฟิก มีลักษณะเป็น Hypertext สามารถออกแบบและพัฒนาร่วมกับภาษาอื่น ๆ เช่น C++ ได้ง่าย Tcl/TK สามารถรันบนระบบปฏิบัติการ Window Mac และ Unix อื่น ๆ เป็นภาษาระดับสูงที่คุณสามารถจะเขียน Code และทำงานที่สามารถตอบสนองใน Platform ที่แตกต่างกันได้ อีกทั้งยังเป็นภาษาที่ใช้งานง่ายมาก ในการเขียนโปรแกรมทดสอบโดยคุณสามารถที่จะเรียนรู้ Tcl และวิธีการสร้างแอพพิเคชั่น เพียงคุณใช้เวลาศึกษาเรียนรู้ 2-3 ชั่วโมงสำหรับผู้ที่มีพื้นฐานด้านการโปรแกรมมาบ้างแล้ว หรือ 2-3 วัน คุณก็สามารถที่จะสร้างโปรแกรมในการทดลองของคุณได้ง่ายหรืออาจสร้างสคริปสั้น ๆ ที่สามารถใช้งานได้ Tcl/Tk ได้มีการพัฒนาอย่างต่อเนื่อง เป็นที่การยอมรับและใช้มานานแล้ว Tcl /Tk สามารถนำไปประยุกต์ใช้ร่วมกับซอฟแวร์อื่น ๆ ได้อย่างง่าย ด้วยการเขียนสคิปไว้ในแอพพิเคชั่น เช่น เดียวกับ C, C++ และ Java มีลักษณะเด่นคือ สามารถใช้ภาษา Tcl นำมาควบคุม HW และใช้ควบคุมการเชื่อมต่อเน็ตเวิร์ค การพัฒนา Tcl เป็นโปรแกรม Dynamic languages Tcl/Tk เป็นภาษา ที่มีแนวคิดสร้างขึ้นมา เพื่อที่จะใช้ทดสอบซอฟท์แวร์ และฮาร์ดแวร์ ซึ่งมีประสิทธิภาพดี มากเป็นภาษาที่มีวัตถุประสงค์ที่ให้ผู้ใช้เกิดความสะดวกสบายสามารถเชื่อมต่อ อุปกรณ์ภายในต่าง ๆ ได้ดี รวมทั้งมีฟังก์ชั่นที่ใช้ในการทดสอบให้เลือกมากมาย สามารถตรวจสอบการประมวลผล แสดงรายงานข้อผิดพลาดที่เกิดขึ้น ให้ผู้ใช้ได้ทราบ และดำเนินการแก้ไขได้ถูกต้อง ในการพัฒนา Tcl เพื่อที่จะสร้างโปรแกรมสามารถบันทึกเป็นไฟล์ *.tcl ซึ่งสามารถเรียกกลับมาใช้ใหม่ได้ ตลอดจนสามารถใช้ร่วมกับภาษาระดับต่ำได้ดี

Tcl/Tk ยังมีตัวอย่างการเขียนโปรแกรมในแต่ละเหตุการณ์ที่สมบูรณ์และเป็น

ประโยชน์ในการเรียนรู้และเขียนโปรแกรมให้พัฒนาไปอย่างรวดเร็ว สามารถเขียน Code โปรแกรมบน Network ได้ง่ายด้วยการเขียนโค๊ดเพียงไม่กี่บรรทัด Tcl/Tk มีหลายเหตุผลดึงดูดใจให้มีผู้นิยมมาใช้ Tcl กันมากขึ้นเนื่องจากมีโปรแกรมเมอร์กลุ่มใหญ่ที่ช่วยแก้ปัญหาสำหรับผู้ใช้ที่ต้องการพัฒนาแอพพิเคชั่นด้วย นอกจากนี้ Tcl/Tk ฟรี ซึ่งผู้ใช้ไม่ต้องเสียค่าใช้จ่ายใด ๆ เนื่องจาก Tcl/Tk เป็น Open source สามารถแก้ไข ซอสโค๊ดต่าง ๆ ได้อย่างอิสระตามความต้องการของผู้ใช้ ทั้งยังสามารถใช้ติดต่อได้ทุก DataBase ที่เป็นมาตรฐาน อาทิเช่น MySQL, Oracle, Sybase, Postgress, Berkeley DB, และ DB อื่น ๆ ที่ติดต่อผ่าน ODBC ได้อีกด้วย จากเหตุผลต่าง ๆ ดังกล่าวมาแล้วข้างต้น ทำให้ผู้วิจัยสนใจ เลือกใช้ ภาษา Tcl/Tk และนำมาใช้พัฒนาแอพพิเคชั่นที่ใช้เป็นเครื่องมือทดสอบ สำหรับปัญหาการจัดตารางการผลิตในงานวิทยานิพนธ์ฉบับนี้ได้เป็นอย่างดี สามารถดาวน์โหลดโปรแกรมได้ที่ http://www.tcl.tk/software/tcltk/