

การกระจายภาระงานด้วยหลักโครงสร้างการรวมกลุ่มที่ดีที่สุด

Task Allocation with Optimal Coalition Structure Concept

พาณิษฐ์ สุกโคต (Panich Sudkhot)* ฉัตรตระกูล สมบัติธีระ (Chattrakul Sombattheera)*

และ สวนิต วัฒนศักดิ์กุล (Savanid Vatanasakdakul)**

บทคัดย่อ

การจัดสรรภาระงานเข้าสู่กระบวนการทำงาน นั้นนับว่าเป็นปัญหาที่มีความสำคัญต่อภาคการผลิต ภาคการขนส่ง ฯลฯ มีการพัฒนาเทคนิคการแก้ปัญหาอย่างต่อเนื่องเพื่อหาวิธีการที่จะทำให้เกิดประสิทธิภาพมากที่สุดแก่ระบบ ในบทความนี้ผู้วิจัยได้ทำศึกษาและพัฒนาอัลกอริทึมโดยใช้หลักการ Optimal Coalition Structure (OCS) เพื่อใช้ในการแก้ไขปัญหการจัดลำดับภาระงาน วิธีการนี้ใช้การหาค่าระยะเวลาในการทำงานของกลุ่มภาระงานที่มีผลรวมของเวลาในการทำงานน้อยที่สุด ซึ่งก็คือการจัดกลุ่มภาระงานซึ่งก่อให้เกิดกระบวนการทำงานที่มีประสิทธิภาพที่สุด

คำสำคัญ: โครงสร้างการรวมกลุ่มที่ดีที่สุด การเพิ่มประสิทธิภาพของกระบวนการ การจัดสรรภาระงาน

Abstract

Task allocation in business processes is an important problem in manufacturing, logistics, etc. There have been researches in developing techniques which can suggest the most efficient solution to the systems. In this research, we have developed an algorithm for task allocation based on optimal coalition structure technique in order to solve the task allocation problem. The technique searches for the allocations which have minimal execution time which is the most efficient business process.

Keyword: Optimal Coalition Structure, Optimization Process, Task Allocation.

*ภาควิชาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

**Department of Accounting & Corporate Governance, Faculty of Business and Economics, Macquarie University

1. บทนำ

ปัจจุบันนับว่าปัญหาและอุปสรรคในการทำงานสำนักงานหรือปัญหาด้านการผลิตต่างๆ นั้นมาจากหลายสาเหตุ แต่ปัจจัยที่สำคัญคือปัญหาด้านทรัพยากรมนุษย์และด้านทรัพยากรระบบงาน ผู้บริหารจะต้องศึกษาวิเคราะห์แต่ละปัญหาเพื่อหาทางปรับปรุงให้เกิดคุณภาพงานที่ดีมีประสิทธิภาพและประสิทธิผลเพื่อบรรลุเป้าหมายตามวัตถุประสงค์ขององค์กร

ซึ่งจากปัญหาและอุปสรรคของการจัดการกระบวนการทำงานและกระบวนการจัดสรรงานนั้นไม่ว่าจะเป็นกระบวนการผลิตของภาคโรงงานอุตสาหกรรม หรือแม้กระทั่งกระบวนการขนส่งของภาคการขนส่ง จะมีกระบวนการทำงานซึ่งในแต่ละองค์กรอาจมีความแตกต่างกันในรูปแบบต่างๆ แต่จะมีสิ่งเหมือนกันคือการไหลของงาน เช่น ในโรงงานอุตสาหกรรมมีกระบวนการวางแผนการผลิตและกระบวนการผลิต ซึ่งในบางอุตสาหกรรมอาจจะยังไม่มีมีการจัดการกระบวนการที่ดีส่งผลให้กระบวนการผลิตภายในล่าช้าและผลิตไม่ได้ตามเป้าหมายที่กำหนดไว้อาจส่งผลกระทบต่อในหลายๆ ด้าน ไม่ว่าจะเป็นต่อโรงงานและประเทศอาจสูญเสียรายได้ในส่วนนี้ไป ดังนั้นโรงงานอุตสาหกรรมจึงจำเป็นต้องมีการจัดการการวางแผนการผลิตและกระบวนการผลิตที่มีประสิทธิภาพเพื่อเพิ่มขีดความสามารถในการผลิต

ดังนั้นในงานวิจัยนี้จึงออกแบบและพัฒนาอัลกอริทึมที่ใช้ในการจัดสรรภาระงานโดยใช้หลักโครงสร้างการรวมกลุ่มที่ดีที่สุด Optimal Coalition Structure (OCS) และการประยุกต์ใช้กระบวนการทำงานต่างๆ มาวิเคราะห์ เพื่อจำลองการไหลของภาระงาน ซึ่งจะสามารถช่วยเพิ่มประสิทธิภาพและขีดความสามารถในการผลิต การขนส่ง ฯลฯ เก็บบันทึกข้อมูล

ทางการผลิตและติดตามกระบวนการทำงานของภาระงานในแต่ละแผนก ซึ่งจะช่วยให้ในเรื่องการวางแผนการวางตำแหน่งของภาระงานที่เหมาะสมและยังสามารถประยุกต์ใช้กับการจำลองการไหลของภาระงานต่างๆ ได้

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 เอเจนต์

เอเจนต์ (Agent) หรือ ซอฟต์แวร์เอเจนต์ (Software agent) ได้ถูกใช้ในด้านเทคโนโลยีมากมายและยังได้รับการใช้อย่างแพร่หลาย เช่น ใน Artificial Intelligence (AI), Databases, Operating systems และ Computer networks แม้ว่าเอเจนต์จะไม่มีคามหมายเดียวใน Genesereth and Ketchpel, 1994 [1], Wooldridge and Jennings, 1995 [2] Russell and Norvig, 2003 [3] แต่านิยามของทุกคนยอมรับว่าเอเจนต์เป็นส่วนประกอบหลักของซอฟต์แวร์พิเศษที่มีความเป็นอิสระมีอินเทอร์เน็ตการทำงานร่วมกันกับระบบ หรือพฤติกรรมที่สามารถจำลองระบบที่ซับซ้อน

ในความหมายทางด้านคอมพิวเตอร์และปัญญาประดิษฐ์ (Computer AI) เอเจนต์ (Agent) คือ ระบบคอมพิวเตอร์ที่มีความสามารถกระทำในบางสิ่งบางอย่างได้อย่างอิสระได้ด้วยตัวเอง ซึ่งเอเจนต์สามารถคิดวางแผนในการกระทำของตัวเองเพื่อให้บรรลุเป้าหมายตามที่ได้รับมอบหมายมา [5-7]

ซอฟต์แวร์เอเจนต์ (Software Agent) คือ ซอฟต์แวร์ซึ่งทำหน้าที่อย่างต่อเนื่องและเป็นอิสระจากกันในสภาพแวดล้อมที่อาศัยอยู่ และใน [8, 9] มีความพยายามที่จะพัฒนาแพลตฟอร์มที่มีการทำงานแบบอัตโนมัติ

2.2 ระบบมัลติเอเจนต์

ใน [6], [7] และ [10-12] ระบบมัลติเอเจนต์ คือ ระบบที่ประกอบด้วยเอเจนต์จำนวนหนึ่ง ที่มีการปฏิสัมพันธ์กัน และสามารถทำงานตามคำสั่งที่ได้รับมอบหมาย เพื่อให้บรรลุเป้าหมายหรือตอบโจทย์ของผู้ใช้งาน ลักษณะของเอเจนต์มีอยู่สามอย่าง 1. Cooperate คือ การทำงานร่วมกัน 2. Coordinate คือ การทำงานประสานกัน 3. Negotiate คือ การต่อรองกัน

2.3 ทฤษฎีเกม

จากการมีปฏิสัมพันธ์กันระหว่างเอเจนต์ข้างต้นจำเป็นต้องรวมเอาทฤษฎีเกม (Game theory) เพื่อเป็นการศึกษาและทำความเข้าใจใน Decision-makers ซึ่งการตัดสินใจจะมี

ผลกระทบต่อกัน ใครจะตัดสินใจอย่างไร แต่ละผลที่จะเกิดขึ้นเป็นอย่างไร ที่มาของทฤษฎีเกมเกิดจากนักคณิตศาสตร์จอห์น ฟอน นอยมันน์ (John von Neumann) ทฤษฎีเกมสามารถแบ่งออกได้เป็นสองอย่าง คือ Non-cooperative game คือ เกมที่ไม่มีความร่วมมือ กติกากำหนดให้ผู้เล่นไม่สามารถพูดคุยเจรจาต่อรองหรือติดต่อกันได้ Cooperative game คือ เกมที่ผู้เล่นมีความร่วมมือกัน ผู้เล่นสามารถที่จะพูดคุยแลกเปลี่ยนการตัดสินใจกันได้ หรือการเจรจาตกลงกันได้ [4]

2.4 โครงสร้างการรวมกลุ่มที่ดีที่สุด

Optimal Coalition Structure [5] คือ โครงสร้างการรวมกลุ่มที่ดีที่สุดของเอเจนต์ (Agents) กำหนดให้ $N = \{1, 2, 3, 4\}$ เป็นเซตของเอเจนต์ กลุ่มของเอเจนต์ประกอบด้วยสับเซตที่ไม่เป็น NULL เกิดจากสมการ $2^n - 1$ ได้แก่ $S = \{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 1\}, \{3, 1\}, \{3, 2\}, \{4, 1\}, \{4, 2\}, \{4, 3\}, \{3, 2, 1\}, \{4, 2, 1\}, \{4, 3, 1\}, \{4, 3, 2\}, \{4, 3, 2, 1\}\}$ มูลค่าของเอเจนต์ในแต่ละกลุ่มจะถูกกำหนดโดย Characteristic Function เช่น $V(S_1) = 10, V(S_2) = 5, V(S_3) = 1$ เป็นต้น และการหาค่าโครงสร้างการรวมกลุ่มที่ดีที่สุดโดยทั่วไปจะสามารถเกิดขึ้นได้หลายแบบ MAX MIX AVG ในงานวิจัยนี้จะใช้ค่าการรวมกลุ่มที่น้อยที่สุดหาได้จากสมการด้านล่าง

$$V(CS) = \sum_{S \in CS} V(S) \quad (1)$$

$$CS^* = \argmin_{CS \in LV(CS)} V(CS)$$

ค่าของกลุ่มที่ดีที่สุดหาได้จากผลรวมของกลุ่มที่มีค่าน้อยที่สุดตามอกรีเมนต์ที่กำหนดขึ้นมาหรืออาจจะเป็นค่ามากที่สุดขึ้นอยู่กับความต้องการและการนำไปใช้

2.5 งานวิจัยที่เกี่ยวข้อง

Sen et. al. [14] ได้เสนออัลกอริทึมในการค้นหาโดยอยู่บนพื้นฐานของเจนเนติกอัลกอริทึม (Order-Based Genetic Algorithm (OBGA)) และยังสามารถประยุกต์เอาหลักโครงสร้างการรวมกลุ่มที่ดีที่สุดที่อยู่ในลักษณะของฟังก์ชันเกมส์ (Optimal Coalition Structure in Characteristic Function Games (CFGs)) [16, 17] มาใช้ในการค้นหาและได้จำลองภาระงานขึ้นมาโดยจำลองให้เห็นถึงการจัดสรรภาระงานโดยใช้หลักทฤษฎีคิว

Sombattheera et. al. [15] ได้เสนออัลกอริทึมที่ตั้งอยู่บนทฤษฎี α -core โดยการค้นหาโครงสร้างการรวมกลุ่มภายใน Linear Production Domain โดยเอเจนต์แต่ละเอเจนต์มี

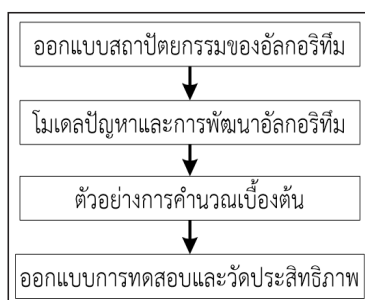
ทรัพยากรที่จะนำมาผลิตสินค้าเพื่อจำหน่าย เอเจนต์จะรวมกันเพื่อเพิ่มผลผลิตและยอดขาย แต่จะมีต้นทุนในการรวมกลุ่มกัน ซึ่งจะทำให้ผลกำไรของกลุ่มและส่วนแบ่งของเอเจนต์ลดลง โดยแต่ละเอเจนต์จะคัดเลือกเอเจนต์ที่เมื่อรวมกลุ่มแล้วมีแนวโน้มจะช่วยเพิ่มกำไรให้กลุ่ม

Michalak et. al. [16] ได้เสนอโครงสร้างการรวมกลุ่มที่ดีที่สุดและการแบ่งโครงสร้างกลุ่มด้วยวิธีหลักพาร์ติชันฟังก์ชันเกมส (Partition Function Game) โดยการแบ่งกลุ่มนั้นเป็นในลักษณะการเกิดกลุ่ม และได้ทดสอบ 4 ชั้นใน Super-additive+, Super-additive-, Sub-additive+, Sub-additive- เพื่อหาโครงสร้างการรวมกลุ่มที่ดีที่สุดที่เกิดในแต่ละชั้น

Sandholm et. al. [17] ได้เสนอถึงการให้กำเนิดโครงสร้างการรวมกลุ่มที่ดีที่สุด โดยได้นั้นในเรื่องของการค้นหาโครงสร้างของกลุ่มที่ดีที่สุดในกลุ่มที่มีขนาดใหญ่ ได้ค้นหาโครงสร้างที่ดีที่สุดได้อย่างรวดเร็ว โดยได้วางการทำงานไว้ 3 ขั้นตอน ลำดับที่ 1 ทำการหาขนาดของโครงสร้างของกลุ่มที่เกิดขึ้น ลำดับที่ 2 ทำการหาค่าของโครงสร้างกลุ่มที่ดีที่สุดในแต่ละโครงสร้าง และลำดับที่ 3 หาค่าของกลุ่มเพื่อแก้ปัญหาในการให้กำเนิดกลุ่มในหมู่เอเจนต์ ได้บอกถึงการค้นหาโครงสร้างกลุ่มกรณีนี้ที่แย่ที่สุดในการค้นหาทุกขอบบนและขอบล่างของการค้นหา

3. วิธีดำเนินการวิจัย

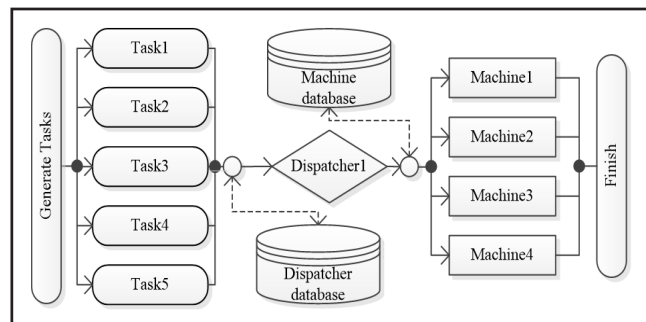
เนื้อหาในส่วนที่ 3 กล่าวถึงวิธีการดำเนินการวิจัยของการออกแบบและพัฒนาอัลกอริทึมสำหรับจัดสรรภาระงานเข้าสู่กระบวนการ ได้นำเสนอแนวคิดออกแบบและพัฒนาเพื่อบรรลุวัตถุประสงค์ของงานวิจัย โดยได้อาศัยแนวคิดและวิธีการวิจัยตามทฤษฎีและงานวิจัยที่เกี่ยวข้องในส่วนที่ 2 ซึ่งแบ่งวิธีดำเนินการวิจัยออกเป็น 4 ขั้นตอนคือ ออกแบบสถาปัตยกรรมของอัลกอริทึม โมเดลปัญหาและการพัฒนาอัลกอริทึม ตัวอย่างการคำนวณเบื้องต้น และออกแบบการทดสอบและวัดประสิทธิภาพ



ภาพที่ 1 วิธีดำเนินการวิจัย

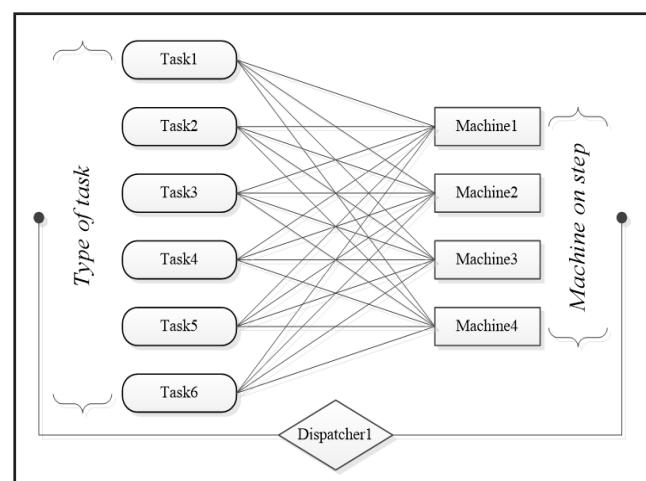
3.1 ออกแบบสถาปัตยกรรมของอัลกอริทึม

อธิบายถึงกระบวนการในการนำภาระงานเข้าสู่กระบวนการเริ่มต้นจากการให้กำเนิดภาระงาน (Generate Tasks) ภาระงานที่ถูกสร้างขึ้นมาจะมีคุณสมบัติที่แตกต่างกันโดยยึดตามฟังก์ชันของกระบวนการ ภาระงานจะถูกนำเข้าสู่ตัวกระจายภาระงาน (Dispatcher) ตัวกระจายจะทำหน้าที่ตัดสินใจในการนำภาระงานเข้าสู่กระบวนการทำงาน (Process) และภาระงานที่ผ่านกระบวนการแล้วจะไหลเข้าสู่กระบวนการต่อไปในลักษณะเดียวกันจนกว่าจะเสร็จสิ้นกระบวนการของภาระงานนั้นๆ



ภาพที่ 2 ภาพแสดงกระบวนการทำงาน

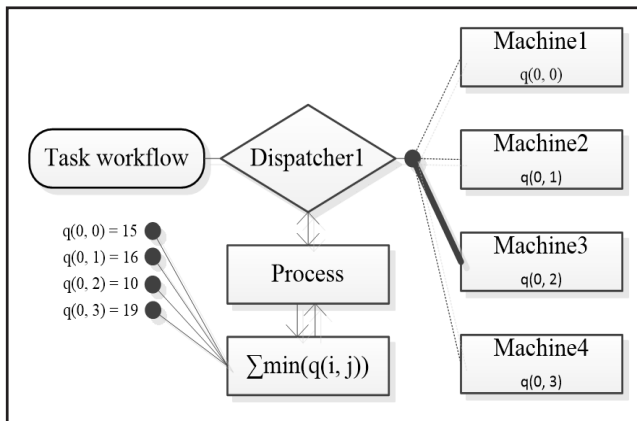
จากภาพที่ 2 เป็นการออกแบบโครงสร้างของกระบวนการทำงาน ผู้วิจัยได้ทำการให้กำเนิดชนิดของภาระงานขึ้นมา โดยข้อมูลจะมีความสัมพันธ์โดยตรงกับฟังก์ชันเวลา ซึ่งภาระงานแต่ละชนิดนั้นมีการกำหนดเวลาให้ทำงานในเครื่องจักรแต่ละเครื่องไม่เท่ากัน เมื่อภาระงานไหลเข้าสู่ตัวกระจายภาระงานจะมีการบันทึกข้อมูลลงฐานข้อมูลและสามารถดูได้ว่าภาระงานชนิดใดเข้าสู่ตัวกระจาย จากนั้นเมื่อภาระงานเข้าสู่ตัวกระจายภาระงานแล้ว ตัวกระจายจะทำการตัดสินใจว่าจะนำภาระงานที่เข้ามาใหม่กระจายสู่เครื่องจักรใดและเมื่อนำภาระงานเข้าสู่เครื่องจักรจะมีการเก็บบันทึกข้อมูลด้วยเช่นกัน



ภาพที่ 3 ความสัมพันธ์ของภาระงานต่อเครื่องจักร

หลังจากนั้นเครื่องจักรจะทำงานกับงานชนิดนั้นๆ โดยมี การประมวลผลตามฟังก์ชันของเวลาที่กำหนดไว้จนเสร็จสิ้น กระบวนการทำงานของภาระงานนั้นๆ

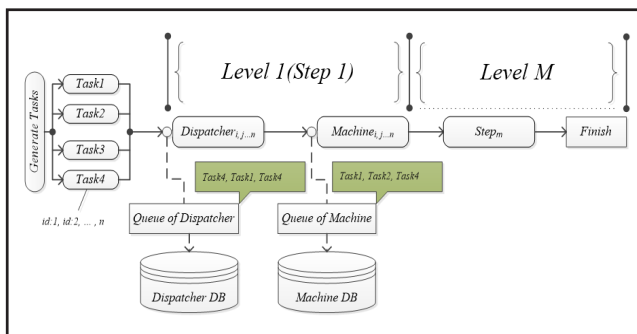
จากภาพที่ 3 แสดงให้เห็นถึงจำนวนการกระจายที่สามารถเกิดขึ้นได้โดยผ่านตัวกระจายภาระงาน ซึ่งแต่ละ ชนิดนั้นจะกระจายได้เป็นจำนวน $M!$ (ตามจำนวนเครื่อง) งานแต่ละชนิดจะสามารถกระจายเข้าสู่เครื่องจักรได้หลายๆ วิธีโดยไม่กำหนดตำแหน่งที่แน่นอน โดยงาน 1 ชิ้นสามารถกระจายไปสู่เครื่องจักรได้ 4 วิธี



ภาพที่ 4 กระบวนการหาค่าเวลาที่น้อยที่สุดของตัวกระจาย

จากภาพที่ 4 แสดงถึงการหาค่า OCS ของโครงสร้างกลุ่ม โดยการหาค่าระยะเวลาในการทำงานของกลุ่มภาระงานที่มี ผลรวมของเวลาในการทำงานน้อยที่สุด โดยใช้หลักการ OCS จากภาพจะเห็นว่าเมื่อภาระงานถูกนำเข้าสู่ตัวกระจายตัว จะ ทำหน้าที่หาคิวที่มีเวลาทำงานน้อยที่สุดและทำการตัดสินใจ และภาพจะเห็นว่าตัวกระจายตัดสินใจกระจายภาระงานเข้าสู่เครื่องจักรที่ 3 $q(0, 2)$

3.2 โมเดลปัญหาและการพัฒนาอัลกอริทึม



ภาพที่ 5 โครงสร้างการไหลของภาระงาน

จากภาพที่ 5 โครงสร้างการไหลของภาระงาน จะจำลอง การไหลของภาระงานจากเริ่มต้นจนจบกระบวนการเริ่มต้น การสร้างข้อมูลภาระงานชนิดต่างๆ งานแต่ละชนิดจะ ประกอบด้วยชิ้นงาน (ID ของภาระงาน) จากนั้นงานจะถูก นำเข้าสู่ตัวกระจายภาระงานเพื่อคำนวณหาค่าเวลาของคิว ในเครื่องจักรที่น้อยที่สุดในตัวกระจายภาระงานจะมีฐาน ข้อมูลเพื่อเก็บบันทึกค่าเวลา ชนิดงาน ชิ้นงาน เป็นต้น เมื่อ ได้ค่าของเวลาของคิวที่ใช้เวลาในการประมวลผลน้อยที่สุด แล้วภาระงานจะถูกนำเข้าสู่คิวของเครื่องจักรนั้นเช่นเดียวกัน เครื่องจักรที่ประจำอยู่ในแต่ละแผนกจะมีฐานข้อมูลเพื่อเก็บ บันทึกค่าต่างๆ และเมื่อภาระงานทำงานในเครื่องจักร จนเสร็จเรียบร้อยแล้วงานจะถูกนำเข้าสู่สถานะรอจ่ายงาน และรอเข้าสู่แผนกต่อไป

ตารางที่ 1 ตารางอธิบายตัวแปร

ตัวแปร	คำอธิบาย
$ T $	เซตชนิดของภาระงานที่จะนำเข้ามาทำงาน โดยที่ $t_{1,...,o}; 1 \leq k \leq o$
D	เซตของตัวกระจายภาระงานเข้าสู่เครื่องจักร ต่างๆ ซึ่งมีประจำอยู่แต่ละแผนก
M	เซตของขั้นตอน (Step) คือ จำนวนแผนกทั้งหมด โดยที่ $1, ..., m; i$
N	เซตของเครื่องจักรทั้งหมดที่อยู่ในแผนกนั้นๆ โดยที่ $1, ..., n; j$
$f(M, T)$	ฟังก์ชันที่กำหนดให้บอกเวลาเฉลี่ยของเครื่องจักร สามารถทำงานกับงานชนิดต่างๆ
$ D $	จำนวนของตัวกระจายที่อยู่ในแต่ละแผนก
$ M $	จำนวนของแผนก
$ N $	จำนวนของเครื่องจักรที่ประจำอยู่ในแต่ละแผนก
P	ภาระงานที่รอเข้าทำงานในเครื่องจักร
$ a_{i,j} $	เอเจนต์ที่ทำหน้าที่ควบคุมเครื่องจักรรับส่งภาระงาน
$ q_{i,j} $	คิวของงานที่อยู่ในเครื่องจักร i, j
$\tau(q_{i,j})$	ฟังก์ชันหาค่าเวลาจากคิวของเครื่องจักร
CS	โครงสร้างการรวมกลุ่ม โดยแบ่งตามชนิดของงาน
$V(CS)$	ค่าของโครงสร้างกลุ่ม
CS^*	โครงสร้างการรวมกลุ่มที่ดีที่สุด
$ar\ gmin$	กำหนดค่าผลรวมที่ได้จาก CS^* คือค่าน้อยที่สุด

จากตารางที่ 1 เขียนในรูปแบบเซตดังต่อไปนี้
 $T = \{ t_1, t_2, \dots, t_o \}$ แทนเซตชนิดของงานที่เข้ามาทำงานในกระบวนการต่างๆ ที่อยู่ในระบบ $D = \{ d_1, d_2, \dots, d_m \}$ แทนเซตของตัวกระจายซึ่งตัวกระจายมีหน้าที่นำงานที่ถูกนำเข้ามาในตัวกระจายต่างๆ ที่อยู่ในแต่ละแผนกมีหน้าที่ตัดสินใจว่าจะนำงานเข้าไปที่คิวใด $M = \{ m_1, m_2, \dots, m_i \}$ แทนเซตของแผนกและ $N = \{ n_1, n_2, \dots, n_j \}$ แทนเซตของเครื่องจักรที่ประจำอยู่แต่ละแผนก ซึ่งในภาพรวมของระบบงานนั้นเขียนได้ในรูปแบบดังต่อไปนี้

$$\begin{matrix} a_{1,1} + & a_{1,2} + & \dots & a_{1,n} \\ a_{2,1} + & a_{2,2} + & \dots & a_{2,n} \\ & \vdots & & \vdots \\ a_{m,1} & a_{m,2} + & \dots & a_{m,n} \end{matrix} \quad (1)$$

โดยที่

$$a_{i,j} \begin{cases} 1 \leq i \leq m \\ 1 \leq j \leq n \end{cases}$$

จากสมการที่ 1 โมเดลเอเจนต์ข้างต้นนั้นเขียนในรูปแบบตัวแปล $a_{i,j}$ จะแทนเอเจนต์ในแผนกต่างๆ เช่น $a_{1,1}$ นั่นก็คือเอเจนต์ที่ 1 ของแผนกที่ 1, $a_{2,2}$ นั่นก็คือเอเจนต์ที่ 2 ของแผนกที่ 2 เป็นต้น และโมเดลของเอเจนต์พบว่าในแผนกต่างๆ นั้นจะมีคนที่ทำหน้าที่นำภาระงานเข้ามาทำในเครื่องจักรเป็นคนคอยควบคุมการทำงานของเครื่องจักรและรับภาระงานเข้า หรือส่งออกภาระงานต่างๆ ซึ่งถ้าเอาไปใช้ในการทำงานจริงจะขึ้นตามโครงสร้างการไหลของงานในโรงงานอุตสาหกรรม ฯลฯ

$$\begin{matrix} a_{1,1} + & a_{1,2} + & \dots & a_{1,n} \\ a_{2,1} + & a_{2,2} + & \dots & a_{2,n} \\ & \vdots & & \vdots \\ a_{m,1} & a_{m,2} + & \dots & a_{m,n} \end{matrix}$$

โดยที่

$$a_{i,j} \begin{cases} 1 \leq i \leq m \\ 1 \leq j \leq n \end{cases}$$

จากสมการที่ 2 โมเดลคิวของภาระงาน ในเครื่องจักรในคิว $q_{i,j}$ นั้นจะมีการงานที่รอการเข้าทำงานในเครื่องจักรอยู่ และใน $q_{i,j}$ จะมีการงานที่อยู่ในคิวต่างๆ ซึ่งกำหนดไว้เป็น p แสดงส่วนประกอบดังสมการที่ 3

$$|q_{i,j}| = |p_{k,id}|; \in m_{i,j} \quad (3)$$

จากสมการที่ 3 โมเดลคิวของเครื่องจักร จะเห็นว่า $q_{i,j}$ จะประกอบด้วยเซตของ P และ id ดังตัวสมการที่ 4

$$|q_{2,1}| = |(p_{i,2}), (p_{i,3}), \dots, (p_{i,x})|; \in m_{2,1} \quad (4)$$

จากสมการที่ 4 ตัวอย่างโมเดลคิวของเครื่องจักร แสดงภาระงานที่อยู่ใน $q_{2,1}$ ซึ่งจะประกอบด้วยภาระงานต่างๆ มีชนิดและไอดีเป็นตัวกำหนดระยะเวลาเฉลี่ยในการเข้าทำงานในเครื่องจักร โดยจะสัมพันธ์กับฟังก์ชัน $J(M,T)$ ซึ่งเป็นฟังก์ชันที่ทำหน้าที่บอกเวลาเฉลี่ยของเครื่องจักรสามารถทำงานกับงานชนิดต่างๆ

$$d_1 + d_2 + \dots d_m \quad (5)$$

จากสมการที่ 5 โมเดลตัวกระจายภาระงาน จะประจำอยู่ในแต่ละแผนกซึ่งแต่ละแผนกนั้นจะมีจำนวน 1 เครื่องเท่านั้น ซึ่งมีหน้าที่รับภาระงานเข้ามาและตรวจสอบหาค่าผลรวมของเวลาในคิวต่างๆ ของเครื่องจักรที่อยู่ในแผนกนั้นๆ เพื่อนำภาระงานที่เข้ามาใหม่กระจายไปสู่คิวที่มีค่าผลรวมเวลาในการทำงานน้อยที่สุดตามค่า $CS^*argmin$

ในภาพรวมของโมเดลต่างๆ ข้างต้นจะมีค่าของเวลาที่ใช้ในการทำงานร่วมกันกับเครื่องจักรที่อยู่ในแต่ละแผนกซึ่งเมื่อมีงานที่ถูกนำเข้ามาสู่ตัวกระจายแล้ว ตัวกระจายจะทำการตัดสินใจว่าควรนำเอาภาระงานเข้าไปที่เครื่องจักรไหน ดังนั้นจึงจำเป็นต้องมีฟังก์ชันที่ทำการคำนวณค่าเวลาของคิวที่อยู่ในแต่ละแผนก สามารถเขียนในรูปแบบของการรวมกลุ่ม ซึ่งจะเอาค่าของเวลาที่ใช้ในการทำงานร่วมกับเครื่องจักรโดยกำหนดให้ $V(\tau(q_{i,j}))$ ซึ่งเป็นค่าของเวลาที่อยู่ในคิว $q_{i,j}$ จะได้ค่าตามที่กำหนดไว้ตั้งแต่ต้นคือ $CS^*argmin$ คือค่าของกลุ่มที่ใช้เวลาในการทำงานน้อยที่สุด ดังนั้นภาระงานที่ถูกนำเข้ามานั้นตัวกระจายจะนำภาระงานนั้นส่งเข้าสู่คิวของเครื่องจักรที่ใช้เวลาในการทำงานน้อยที่สุด สามารถคำนวณได้จากสมการ 6 และ 7 สมการคำนวณหาเวลาของโครงสร้างภาระงานที่ดีที่สุด

$$\tau(q_{i,j}) = \sum_{p_{k,id} \in q_{i,j}} p_{k,id} \quad (6)$$

$$CS = q_{i,j} \quad (7)$$

$$V_{c,s} = \tau(q_{i,j}) = \sum_{p_{k,id} \in q_{i,j}} p_{k,id}$$

$$CS^* = argmin \tau(q_{i,j})$$



3.3 ตัวอย่างการคำนวณเบื้องต้น

จากหัวข้อ 3.2 โมเดลปัญหาและพัฒนาอัลกอริทึมข้างต้น ได้ทำการออกแบบโมเดลต่างๆ เพื่อที่จะนำมาแก้ไขปัญหาที่กล่าวในส่วนของบทนำ เพื่อให้เห็นภาพการออกแบบโมเดลปัญหาที่ชัดเจนผู้วิจัยได้ทำการแสดงการคำนวณหาค่า $CS^*argmin$ เบื้องต้นไว้ในส่วนนี้

ตารางที่ 2 กำหนดค่าตัวแปรเบื้องต้น

ตัวแปร	กำหนดค่า	คำอธิบาย
$ T $	4	กำหนดให้ชนิดของงานมี 4 ชนิด
D	2	กำหนดให้ตัวกระจายมี 2 เครื่อง
M	2	กำหนดให้แผนกมี 2 แผนก
N	4	กำหนดให้เครื่องจักรที่อยู่ในแต่ละแผนกมี 4 เครื่อง

ตารางที่ 3 ความสามารถของเครื่องจักรที่สามารถทำงานได้กับภาระงานแผนกที่ 1

$t_{1,...,4}$	$q_{0,0}$	$q_{0,1}$	$q_{0,2}$	$q_{0,3}$
t_1	10	15	5	10
t_2	5	20	10	10
t_3	15	1	15	5
t_4	1	5	30	10

ตารางที่ 4 ความสามารถของเครื่องจักรที่สามารถทำงานได้กับภาระงานแผนกที่ 2

$t_{1,...,4}$	$q_{1,0}$	$q_{1,1}$	$q_{1,2}$	$q_{1,3}$
t_1	6	10	5	15
t_2	12	20	15	10
t_3	15	10	20	30
t_4	20	5	5	5

จากตารางที่ 2 ตารางที่ 3 และตารางที่ 4 เป็นการกำหนดค่าเบื้องต้นให้กับระบบ ตารางที่ 2 นั้นกำหนดค่าขึ้นมาโดยให้ชนิดของงานมี 4 ชนิด ตัวกระจายภาระงานมี 2 เครื่อง ส่วนนี้จะสัมพันธ์กับจำนวนแผนก จำนวนแผนกมี 2 แผนก และจำนวนเครื่องจักรที่ประจำอยู่ในแต่ละแผนก มีอยู่แผนกละ 4 เครื่อง และใน ตารางที่ 3 และตารางที่ 4 คือ $J(M,T)$ ซึ่งมีหน้าที่บอกคุณสมบัติของเครื่องจักรว่ามีความเร็วใน

การทำงานร่วมกับงานชนิดต่างๆ โดยใช้เวลาเฉลี่ยนานเท่าใด ซึ่งผู้วิจัยได้กำหนดขึ้นมาเพื่อเป็นตัวอย่างในการคำนวณเบื้องต้นซึ่งในความเป็นจริงแล้วค่าเหล่านี้จะได้จากการคำนวณค่าความลึกหรือของเครื่องจักรนั้นๆ

ตารางที่ 5 จำลองภาระงานที่อยู่ในคิวของเครื่องจักร (ภาระงานใน $q_{i,j}$ ในแผนกที่ 1)

$q_{0,0}$	$q_{0,1}$	$q_{0,2}$	$q_{0,3}$
$t(1), t(2),$ $t(1), t(4),$ $t(1), t(1),$ $t(3), t(2)$	$t(3), t(1),$ $t(2), t(3),$ $t(2), t(1)$	$t(1), t(2),$ $t(3), t(1),$ $t(1)$	$t(4), t(4),$ $t(4), t(4),$ $t(4), t(1),$ $t(3)$

ตารางที่ 6 จำลองภาระงานที่อยู่ในคิวของเครื่องจักร (ภาระงานใน $q_{i,j}$ ในแผนกที่ 2)

$q_{0,0}$	$q_{0,1}$	$q_{0,2}$	$q_{0,3}$
$t(1), t(2),$ $t(1), t(1),$ $t(4)$	$t(3), t(1),$ $t(1), t(1),$ $t(2), t(4),$ $t(1), t(2),$ $t(1), t(4)$	$t(1), t(1),$ $t(1), t(4),$ $t(1), t(4),$ $t(1), t(4),$ $t(4)$	$t(1), t(4),$ $t(1), t(3),$ $t(3), t(2)$

จากตารางที่ 5 และ ตารางที่ 6 จะแสดงในเรื่องของการจำลองภาระงานที่อยู่ในคิวของเครื่องจักรที่ประจำอยู่ในแต่ละแผนก ซึ่งในแต่ละคิวของเครื่องจักรนั้นจะมีงานต่างๆ ที่ถูกส่งมาจากตัวกระจายภาระงานและภาระงานนั้นมีเวลาในการทำงานร่วมกับเครื่องจักรตั้งนั้นเมื่อมีงานที่ถูกส่งเข้ามาใหม่ตัวกระจายจะทำหน้าที่หาค่าผลรวมเวลาของภาระงานที่อยู่ในคิวต่างๆ และเมื่อได้ค่าผลรวมเสร็จเรียบร้อยแล้วตัวกระจายจะส่งงานใหม่เข้าไปยังคิวของเครื่องจักรที่ใช้เวลาในการทำงานน้อยที่สุด ดังแสดงในสมการที่ 3 ตัวอย่างการหาค่าโครงสร้างของคิวที่ใช้เวลาในการประมวลผลน้อยที่สุด

$$CS^*m_0 = argmin \tau(q_{i,j}) = \sum_{p_{k,id} \in q_{i,j}} p_{t_k,id} = \begin{cases} q_{0,0} = 66 \\ q_{0,1} = 57 \\ q_{0,2} = 90 \\ q_{0,3} = 65 \end{cases} \quad (8)$$

$$CS^*m_1 = argmin \tau(q_{i,j}) = \sum_{p_{k,id} \in q_{i,j}} p_{t_k,id} = \begin{cases} q_{0,0} = 66 \\ q_{0,1} = 57 \\ q_{0,2} = 90 \\ q_{0,3} = 65 \end{cases}$$

จากสมการ 8 แสดงการคำนวณเวลารวมในการทำงานในคิวเครื่องจักรต่างๆ ในแผนก m_0 และ m_1 ซึ่งจากการคำนวณนั้นจะเอาค่ากลุ่มที่ใช้เวลาในการทำงานน้อยที่สุดจากการคำนวณข้างต้นนั้น $m_0 = q_{0,1} = 57$ และ $m_1 = q_{1,2} = 45$ ในแผนกที่ 1 นั้นคิวของเครื่องจักรที่ใช้เวลาน้อยที่สุดคือ $q_{0,1}$ มีค่าเท่ากับ 57 และในแผนกที่ 2 คิวของเครื่องจักรที่ใช้เวลาน้อยที่สุดคือ $q_{1,2}$ มีค่าเท่ากับ 45 ดังนั้นภาระงานที่เข้ามาใหม่จะถูกส่งเข้าไปยังกลุ่มคิวที่มีผลรวมเวลาในการทำงานน้อยที่สุด

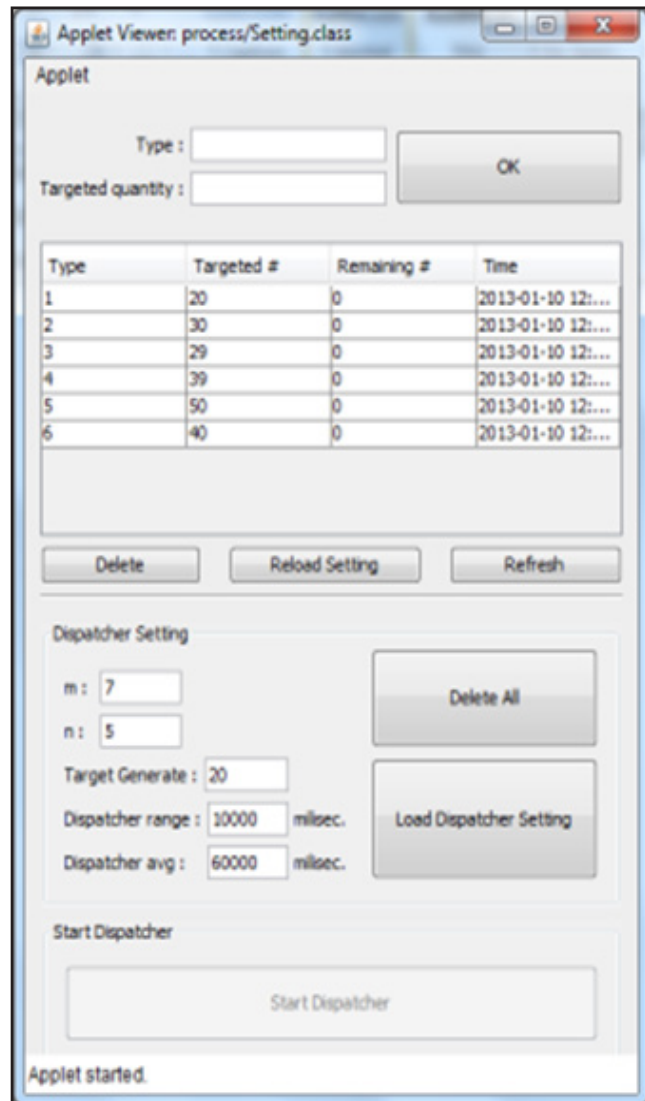
3.4 ออกแบบการทดสอบและวัดประสิทธิภาพ

อัลกอริทึมพัฒนาด้วยภาษาจาวา (Java) ฮาร์ดแวร์ Intel Core (R) i5 CPU 3.20 GHz 8 GB of RAM ใช้ในการพัฒนาซอฟต์แวร์และทดสอบระบบ ซอฟต์แวร์ Microsoft Windows 8 x64 OS, NetBeans IDE 7.1.2, Java JDK Version 7 Update 21 และ MySQL Server Version 5.0.51b ส่วนของการกำหนดฟังก์ชันตั้งค่าเวลาของตัวกระจายงานกำหนดค่าเป็น ($avgtime6000millisec \pm range1000millisec$) ครึ่งละ 20 ภาระงาน ฟังก์ชันเวลาการทำงานของเครื่องจักรกำหนด ($avgtime6000millisec \pm range5000millisec$) ฟังก์ชันเวลาของภาระงานที่มีผลโดยตรงกับเวลาทำงานของเครื่องจักร ($avgtime2000millisec \pm range1000millisec$) และการทดสอบได้ทำการกำหนดค่าของการตั้งค่าในการทดลองผู้วิจัยได้จำลองข้อมูลของแผนก (m) จำนวน 5 แผนกตัวอย่างเช่นงานชนิดที่ 1 งานทั้งหมด 20 งาน ผ่านการทำงานในเครื่องจักรเท่ากับ $20 \times 5 = 100$ งาน การทำงานของภาระงาน: ต่อขึ้นผ่านเครื่องจักรในแต่ละแผนกใช้เวลาโดยเฉลี่ย 4.04 และ 6.671 วินาที (ตัวอย่างชนิดงานที่ 1 ในตารางที่ 8) เครื่องจักร (n) ที่อยู่ในแต่ละแผนกจำนวน 7 เครื่อง จำนวนชนิดงาน 15 ชนิด สามารถดูได้จากตารางที่ 7 ออกแบบการทดสอบ และภาพที่ 6 ตัวอย่างตัวแบบจำลองการตั้งค่าโดยมีการให้กำเนิดภาระงานขึ้นมาจากนั้นนำไปเปรียบเทียบหาค่าความเร็วของอัลกอริทึมที่พัฒนาขึ้นเปรียบเทียบกับอัลกอริทึม Order-Based Genetic Algorithm (OBGA) [14] ในสภาพแวดล้อมเดียวกันซึ่งภาระงานกว่า 1193 จำนวนของกระบวนการกว่า 5965 กระบวนการ

การทดสอบแบ่งออกเป็น 2 ส่วน คือ หาค่าความเร็วเฉลี่ยของอัลกอริทึมที่พัฒนาขึ้นเปรียบเทียบกับอัลกอริทึม

ตารางที่ 7 ผลการคำนวณสนับสนุนสูงสุด 5 อันดับ

ตัวแปร	กำหนดค่า	คำอธิบาย
$ T $	15	กำหนดให้ชนิดของงาน
D	5	กำหนดให้ตัวกระจายมี 5 เครื่อง
M	5	กำหนดให้แผนกมี 5 แผนก
N		กำหนดให้เครื่องจักรที่อยู่ในแต่ละแผนกมี 7 เครื่อง



ภาพที่ 6 ตัวอย่างตัวแบบจำลองการตั้งค่า

Order-Based Genetic Algorithm (OBGA) [14] ของเวลาที่ใช้ในการประมวลผลกับภาระงานชนิดต่างๆ โดยคิดเป็นร้อยละ (ความเร็ว) ในสภาพแวดล้อมเดียวกัน หาค่าความเร็วเฉลี่ยของภาระงานแต่ละชนิดที่จัดสรรเข้าไปในแต่ละแผนก

4. ผลการวิจัยและการอภิปราย

ในส่วนนี้ผู้วิจัยจะกล่าวถึงผลการการวิจัยและการอภิปราย โดยมีรายละเอียดดังนี้

4.1 เกณฑ์การวัดประสิทธิภาพ

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องในเรื่อง โครงสร้างการรวมกลุ่มที่ดีที่สุดที่เขียนในส่วนที่ 2 นั้นผู้วิจัย มีแนวคิดที่จะนำเอาหลักโครงสร้างการรวมกลุ่มที่ดีที่สุดมา พัฒนาอัลกอริทึมในการจัดสรรภาระงานในกระบวนการ ต่างๆ ที่เกิดขึ้นเพื่อแก้ไขการกระจายภาระงานให้เหมาะสม ที่สุดในการที่จะนำเอาภาระงานเข้าไปทำงานในกระบวนการ เพื่อที่จะทำให้ใช้เวลาได้อย่างคุ้มค่าที่สุด ทำให้เกิด การจัดสรรภาระงานให้มีประสิทธิภาพและประสิทธิผลมาก ที่สุด และจากการศึกษาอัลกอริทึมที่ใช้อยู่ในการจัดสรร ภาระงานนั้นพบว่ายังมีผู้ที่พัฒนาอัลกอริทึมชนิดด้านนี้มี ปริมาณน้อย ดังนั้นผู้วิจัยได้ทำการยกเอาอัลกอริทึมด้าน การจัดสรรภาระงานมาหนึ่งอัลกอริทึมนั่นคือ Order-Based Genetic Algorithm (OBGA) [14] ซึ่งผู้วิจัยจะทำการวัด ประสิทธิภาพในด้านความเร็วเพื่อเปรียบเทียบกับอัลกอริทึม ของผู้วิจัย Task Allocation Coalition Structure (TACS) ที่ได้ พัฒนาขึ้นโดยมีหลักเกณฑ์การวัดประสิทธิภาพดังต่อไปนี้

- 1) สภาพแวดล้อมในการทดลอง
- 2) ข้อมูลที่ใช้ในการจำลองภาระงานเข้าสู่กระบวนการ

4.2 ค่าความเร็วเฉลี่ยของภาระงานแต่ละชนิดที่ จัดสรรเข้าในแต่ละแผนก OBGAเปรียบเทียบกับ TACS

จากตารางที่ 8 ผลการทดสอบของ TACS และ OBGA พบว่าค่าเฉลี่ยของภาระงานทั้ง 15 ชนิด TACS ใช้เวลาใน การทำงานในแต่ละแผนกต่างๆ ได้ดีกว่า OBGA โดยหาค่าเฉลี่ย รวมจากค่าของภาระงานทุกชนิด อัลกอริทึมที่พัฒนาขึ้น ทำงานได้ดีกว่าเฉลี่ย 1.476706667 วินาที

4.3 ค่าเฉลี่ยความเร็วของเครื่องจักรในแต่ละแผนก

เมื่อทำการรันทดสอบอัลกอริทึม TACS แล้วเครื่องจักรที่ ถูกจำลองขึ้นจะทำงานผสมหน้าที่กับเวลาที่กำหนดไว้ก่อน หน้านี้ ซึ่งสามารถดูผลการการทดสอบได้ในตารางที่ 9

จากตารางที่ 9 อธิบายถึงเครื่องจักรที่ประจำอยู่ในแต่ละ แผนกซึ่งบอกถึงค่าเฉลี่ยของภาระงานแต่ละชนิดที่ทำงาน กับเครื่องจักร เช่น แผนกที่ 1 (Step1) เครื่องจักร 0 มีค่าเฉลี่ย ทำงานกับภาระงาน 3.76 วินาทีต่อชิ้นงาน และบอกถึง

ค่าเฉลี่ยโดยรวมของแผนกเช่น แผนกที่ 1 (Step1) มีผลรวม เฉลี่ย 3.55 วินาทีต่อชิ้นงาน ซึ่งการหาสภาพแวดล้อมที่ เอื้ออำนวยต่อการกระจายภาระงานมากที่สุดสามารถดูได้ จาก ตารางที่ 10 ผลรวมเวลาเฉลี่ยของเครื่องจักรแต่ละ เครื่องในแต่ละแผนก

ตารางที่ 10 อธิบายถึงเวลาเฉลี่ยของเครื่องจักรที่ทำงาน อยู่ในแต่ละแผนก ซึ่งภาระงานแต่ละชิ้นจะไหลผ่านเครื่องจักร เหล่านี้และใช้เวลาเฉลี่ยต่อชิ้น เช่น แผนกที่ 1 เครื่องจักรที่ 0 เครื่องจักรตัวนี้ใช้เวลาในการประมวลผลต่อชิ้นเวลาเฉลี่ย 3.76 วินาที

จากผลการทดสอบในตารางที่ 10 พบว่าเครื่อง M2 มีผล รวมของเวลาเฉลี่ยอยู่ที่ 21.4 วินาที และเครื่อง M5 มีผลรวม ของเวลาเฉลี่ยอยู่ที่ 18.74 วินาที พิจารณาระหว่าง M2 และ M5 พบว่า M5 ใช้เวลาน้อยที่สุดในการประมวลผลภาระงาน ดังนั้น M5 เป็นเครื่องจักรที่มีสภาพแวดล้อมที่เอื้ออำนวยต่อ การกระจายภาระงานมากที่สุด และ M2 เป็นเครื่องจักรที่มี สภาพแวดล้อมเอื้ออำนวยต่อการกระจายภาระงานน้อยที่สุด ทั้งนี้ ในตารางที่ 10 เป็นการหาโซลูชันที่ดีที่สุดไปจนถึง โซลูชันที่แย่ที่สุดของสภาพแวดล้อมในการกระจายภาระงาน เข้าสู่เครื่องจักรต่างๆ

ตารางที่ 8 ผลการทดสอบการจัดสรรภาระงานเข้าสู่แผนก ต่างๆ คิดเป็นค่าความเร็วเฉลี่ย

Type	Task	Sum Task	OBGA(sec)	TACS(sec)
1	20	100	6.671	4.04
2	25	125	6.58	4.048
3	15	75	5.212	3.9733
4	35	175	4.91	3.88
5	61	305	5.124	3.918
6	55	275	5.3156	3.9273
7	150	750	5.152	3.8973
8	200	1000	4.941	3.979
9	31	155	5.5321	4.2065
10	99	495	5.388	3.8586
11	56	280	5.313	3.8571
12	99	495	5.173	3.8828
13	99	480	5.04	3.8729
14	50	250	5.424	3.868
15	201	1005	5.521	3.9373
AVG			5.41978	3.943073333



ตารางที่ 9 ค่าเฉลี่ยที่เครื่องจักรทำงานกับภาระงาน

แผนก	เครื่องจักร	ค่าเฉลี่ย (Sec)
Step 1	0	3.76
	1	3.59
	2	3.31
	3	3.30
	4	3.48
	5	3.78
	6	3.59
AVG		3.55
แผนก	เครื่องจักร	ค่าเฉลี่ย (Sec)
Step 2	0	4.59
	1	4.75
	2	4.41
	3	4.50
	4	4.57
	5	4.70
	6	4.49
AVG		4.57
แผนก	เครื่องจักร	ค่าเฉลี่ย (Sec)
Step 3	0	3.06
	1	3.05
	2	3.09
	3	3.32
	4	2.95
	5	2.96
	6	2.79
AVG		3.03
แผนก	เครื่องจักร	ค่าเฉลี่ย (Sec)
Step 4	0	4.17
	1	3.87
	2	4.16
	3	4.00
	4	3.99
	5	4.27
	6	3.74
AVG		4.03
แผนก	เครื่องจักร	ค่าเฉลี่ย (Sec)
Step 5	0	3.21
	1	3.50
	2	6.43
	3	5.94
	4	5.77
	5	3.03
	6	5.21
AVG		4.73

ตารางที่ 10 ผลรวมเวลาเฉลี่ยของเครื่องจักรแต่ละเครื่องในแต่ละแผนก

Step	M0	M1	M2	M3	M4	M5	M6
1	3.76	3.59	3.31	3.30	3.48	3.78	3.59
2	4.59	4.75	4.41	4.50	4.57	4.70	4.49
3	3.06	3.05	3.09	3.32	2.95	2.96	2.79
4	4.17	3.87	4.16	4.00	3.99	4.27	3.74
5	3.21	3.50	6.43	5.94	5.77	3.03	5.21
Sum	18.79	18.76	21.4	21.06	20.76	18.74	19.82
* Step = แผนก, M = เครื่องจักร;							

5. สรุปผล

จากการที่ผู้วิจัยได้ทำการศึกษาและพัฒนาอัลกอริทึม TACS โดยยึดเอาหลัก OCS มาใช้ในการแก้ไขปัญหาการจัดสรรภาระงานเข้าสู่กระบวนการทำงานโดยเปรียบเทียบกับ OBGA ผลการทดลองในแง่ของเวลาพบว่า TACS ทำงานได้ดีกว่า OBGA และยังได้มีการหาโซลูชันที่ดีที่สุดจนถึงโซลูชันแย่ที่สุดที่เกิดขึ้นในการกระจายภาระงาน เพื่อที่จะทำให้ใช้เวลาได้อย่างคุ้มค่าที่สุด ทำให้เกิดการจัดสรรภาระงานได้อย่างมีประสิทธิภาพและประสิทธิผลต่อการดำเนินการกระบวนการ

6. เอกสารอ้างอิง

- [1] R. Michael Genesereth and P. Steven Ketchpel. "Software Agents." *Communications of the ACM*. Vol. 37, pp. 48-53, 1994.
- [2] M. Wooldridge and N. R. Jennings. *Towards a Theory of Cooperative Problem Solving*. Distributed Software Agents and Applications: Springer Berlin Heidelberg , pp. 40-53, 1994.
- [3] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*: Pearson, 1998.
- [4] D. Philip Straffin. *Game Theory and Strategy*. The Mathematical Association of America 1993.
- [5] Mark D'Inverno, Michael Luck. *Understanding Agent Systems*. 1 ed: Springer-Verlag, 2001.
- [6] Michael Wooldridge. *An Introduction to MultiAgent Systems*. 2 ed: Wiley 2009.



- [7] Gerhard Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 2000.
- [8] M. Jeffrey Bradshaw. *Software Agents*: AAAI Press, 1997.
- [9] Fabio Luigi Bellifemine and Giovanni Caire. *Dominic Greenwood. Developing Multi-Agent Systems with JADE*. 1 ed: Wiley 2007.
- [10] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Chapter 2 Intelligent Agents: Cambridge University Press, 2008.
- [11] Mark d’Inverno and M. Luck. *Understanding Agent Systems: Springer-Verlag*, 2001.
- [12] H. Rafael Bordini, Jomi Fred Hübner and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. 1 ed: Wiley-Interscience, 2007.
- [13] Jv. Neumann, O. Morgenstern, HW. Kuhn and A. Rubinstein. *Theory of Games and Economic Behavior*. Princeton University Press; 60 Anv edition, 2007.
- [14] S. Sen and P. S. Dutta. *Searching for optimal coalition structures*. pp. 287-292, 2000.
- [15] C. Sombattheera and A. Ghose. “A pruning-based algorithm for computing optimal coalition structures in linear production domains.” *In Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence*, Québec City, Québec, Canada, pp. 13-24, 2006.
- [16] T. Michalak, A. Dowell, P. McBurney and et al. “Optimal Coalition Structure Generation In Partition Function Games.” *In Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pp. 388-392, 2008.
- [17] T. Sandholm, K. Larson, M. Andersson and et al. “Coalition structure generation with worst case guarantees.” *Artif. Intell.*, Vol. 111, No. 1-2, pp. 209-238, 1999.