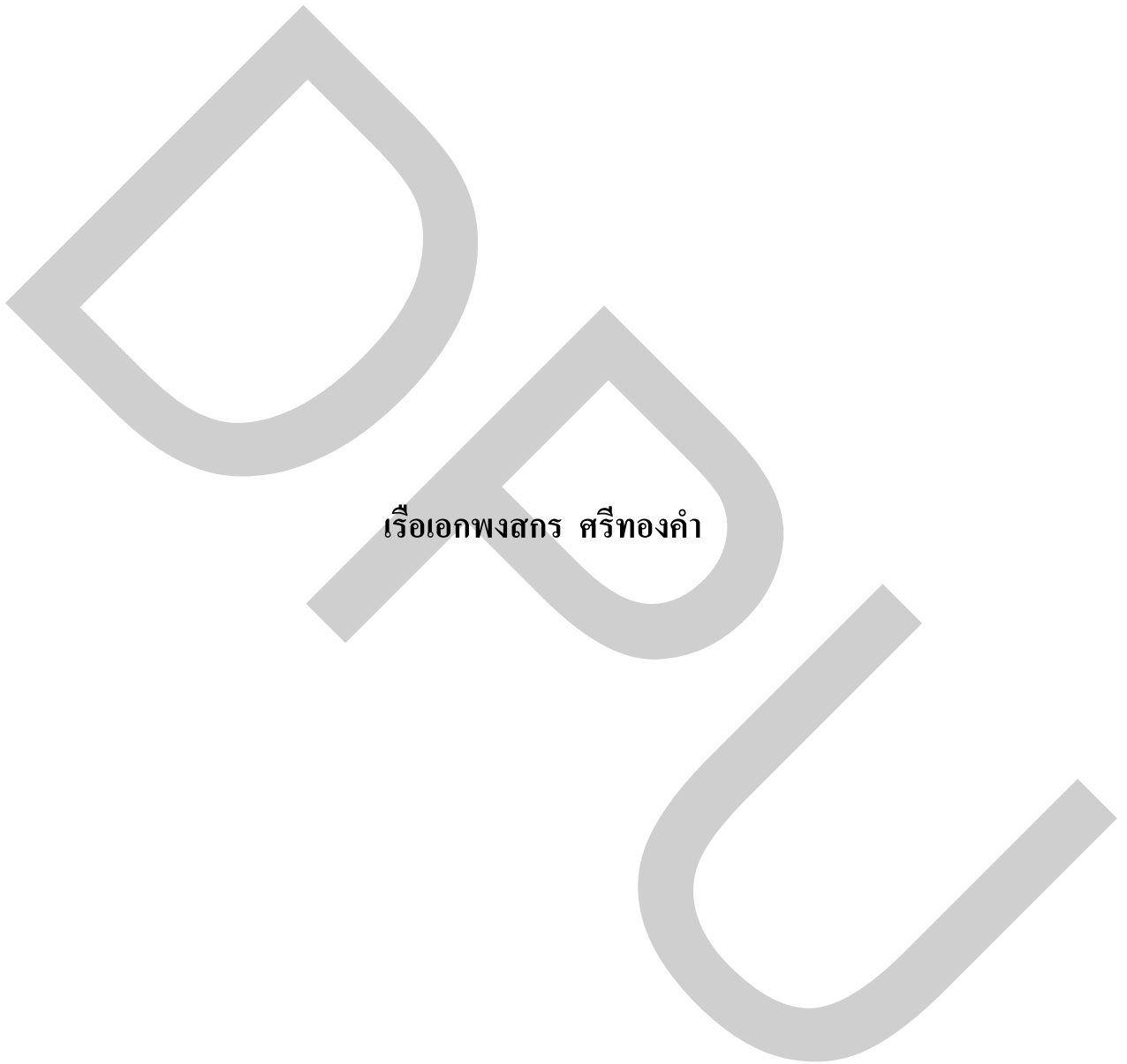


วิธีการหาคำตอบที่เหมาะสมโดยประมาณสำหรับปัญหาปริมาณการสั่งซื้อหรือผลิต
แบบประหยัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกัน



เรือเอกพงสกร ศรีทองคำ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการจัดการทางวิศวกรรม บัณฑิตวิทยาลัย มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ.2549

ISBN 974-9954-82-5

A Heuristic Approach for the Capacitated Multi-item Lot-sizing Problem



Lieutenant Pongsakorn Srithongkam

A Thesis Submitted in Partial Fulfillment of Requirements

for the Degree of Master of Science

Department of Engineering Management

Graduate School, Dhurakij Pundit University

2006

ISBN 974-9954-82-5

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้ก็เนื่องจากความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร.ไพฑูรย์ ฮ้อยิ่ง และ ผู้ช่วยศาสตราจารย์ ดร.ศุภรัชชัย วรรณันท์ ที่กรุณาได้รับเป็นอาจารย์ที่ปรึกษา และอาจารย์ที่ปรึกษาร่วมตามลำดับ ซึ่งให้การดูแล ให้คำแนะนำ และข้อคิดเห็นต่างๆที่มีประโยชน์ด้วยความเมตตาตลอดระยะเวลาของการดำเนินการวิจัย ผู้วิจัยขอกราบขอบพระคุณอาจารย์เป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณ นางสาว สุภาภรณ์ ธีวณิชธรรม และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ได้ให้ความช่วยเหลือและอำนวยความสะดวกในการติดต่อประสานงาน

ขอขอบคุณ พี่ๆน้องๆและเพื่อนๆสาขาการจัดการทางวิศวกรรมรุ่นที่ 1 ที่ให้ความช่วยเหลือในการทำงานวิจัยและเป็นกำลังใจแก่ผู้วิจัยด้วยความจริงใจเสมอมา

ขอกราบขอบพระคุณ บิดา มารดาและพี่ๆ เป็นอย่างสูงที่ให้การสนับสนุน ดูแลเอาใจใส่ผู้วิจัยด้วยความรัก ขอขอบคุณนางสาวธรรมรุจา อุดม ที่ให้ความช่วยเหลือด้วยความเต็มใจ สุดท้ายขอขอบคุณทุกความช่วยเหลือตลอดจนกำลังใจที่มีให้กับผู้วิจัยจนทำให้การวิจัยสำเร็จลุล่วงไปด้วยดี

เรือเอกพงสกร ศรีทองคำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....ฉ	ฉ
บทคัดย่อภาษาอังกฤษ.....ง	ง
กิตติกรรมประกาศ.....จ	จ
สารบัญตาราง.....ฉ	ฉ
สารบัญภาพ.....ญ	ญ
บทที่	
1. บทนำ.....1	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....1	1
1.2 วัตถุประสงค์ของงานวิจัย.....4	4
1.3 ขอบเขตการวิจัย.....4	4
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....4	4
1.5 ขั้นตอนในการวิจัย.....5	5
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....6	6
2.1 แนวคิดเกี่ยวกับสินค้าคงคลัง.....6	6
2.2 รูปแบบปัญหาและการจัดการสินค้าคงคลัง.....8	8
2.3 วิธีการแก้ปัญหาของตัวแบบสินค้าคงคลัง.....11	11
2.4 ทฤษฎีการแก้ปัญหาด้วยวิธีค้นหาคำตอบแบบฮิวริสติก.....12	12
2.5 งานวิจัยที่เกี่ยวข้อง.....15	15
2.6 สรุปปัญหา.....18	18
3. การพัฒนาวิธีหาคำตอบที่เหมาะสมโดยประมาณ.....19	19
3.1 แบบจำลองทางคณิตศาสตร์ของปัญหา.....19	19
3.2 แนวทางการแก้ไขปัญหา.....20	20
3.3 หลักการหาคำตอบที่เหมาะสมโดยประมาณที่นำเสนอ.....21	21
3.4 ขั้นตอนการหาคำตอบที่เหมาะสมโดยประมาณที่นำเสนอ.....22	22
3.5 ตัวอย่างการคำนวณ.....26	26
3.6 เครื่องมือที่ใช้ในงานวิจัย.....28	28
3.7 ตัวอย่างปัญหาสำหรับทดสอบวิธีการแก้ปัญหาที่นำเสนอ.....30	30

สารบัญ (ต่อ)

	หน้า
3.8 การวิเคราะห์ผลการทดลอง.....	30
3.9 ผลการทดลองเบื้องต้น.....	31
3.10 สรุปผลการทดลองเบื้องต้น.....	32
4. ผลของการศึกษา.....	33
4.1 รูปแบบของกลุ่มปัญหาตัวอย่าง.....	33
4.2 กลุ่มปัญหาตัวอย่างขนาดเล็ก.....	33
4.3 กลุ่มปัญหาตัวอย่างขนาดกลาง.....	34
4.4 กลุ่มปัญหาตัวอย่างขนาดใหญ่.....	35
4.5 ผลการศึกษา.....	36
4.6 การวิเคราะห์ผลการศึกษา.....	37
4.7 สรุปการวิเคราะห์ผลการศึกษา.....	48
5. สรุปผลการศึกษา.....	49
5.1 สรุปผลการศึกษา.....	49
5.2 ประโยชน์ที่ได้รับจากงานวิจัย.....	50
5.3 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต.....	50
บรรณานุกรม.....	51
ภาคผนวก.....	54
ภาคผนวก ก ปัญหาตัวอย่างและผลการทดสอบของปัญหา.....	55
ก.1 ปัญหาตัวอย่าง.....	56
ก.2 ผลการทดลองของกลุ่มปัญหาขนาดเล็ก.....	60
ก.3 ผลการทดลองของกลุ่มปัญหาขนาดกลาง.....	63
ก.4 ผลการทดลองของกลุ่มปัญหาขนาดใหญ่.....	67
ก.5 ผลการทดลองของกลุ่มปัญหาสำหรับทดสอบผลของ สัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา.....	77

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ข คู่มือการใช้งานโปรแกรมและโค้ดของโปรแกรม.....	81
ข.1 การเรียกใช้งาน โปรแกรม Lot Size Solving	82
ข.2 การสร้างหน้าต่างปัญหาใหม่และการคำนวณหาคำตอบ.....	83
ข.3 การบันทึกและบันทึกเป็นของโปรแกรม.....	88
ข.4 การออกจากปัญหาปัจจุบันและการเปิดปัญหาที่บันทึก ไว้ภายในเครื่องคอมพิวเตอร์.....	91
ข.5 การออกจากโปรแกรม Lot Size Solving.....	91
ข.6 โค้ดของโปรแกรม Lot Size Solving.....	91

สารบัญตาราง

ตารางที่	หน้า
3.1 ผลการทดลองปัญหาขนาดสินค้า 3 ชนิด 6 ช่วงเวลา.....	31
3.2 ผลการทดลองปัญหาขนาดสินค้า 5 ชนิด 10 ช่วงเวลา.....	31
4.1 ผลการทดสอบจำแนกตามจำนวนสินค้าและช่วงเวลา.....	36
4.2 ผลการทดสอบปัญหาเพื่อวัดผลของสัดส่วนระหว่าง ค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา.....	44

สารบัญภาพ

ภาพที่	หน้า
2.1 กราฟแสดงความสัมพันธ์ของต้นทุนทั้ง 4 ประเภท.....	10
3.1 ขั้นตอนการคำนวณหลัก.....	23
3.2 ขั้นตอนการคำนวณของโปรแกรมย่อยที่ 1.....	24
3.3 ขั้นตอนการคำนวณของโปรแกรมย่อยที่ 2.....	25
3.4 ข้อมูลปัญหาตัวอย่าง.....	26
3.5 ผลลัพธ์ของการคำนวณขั้นที่ 2.....	26
3.6 ผลลัพธ์ของการคำนวณขั้นที่ 3.....	27
3.7 ผลลัพธ์ของการคำนวณขั้นที่ 4.....	27
3.8 ผลลัพธ์ของการคำนวณขั้นที่ 5.....	28
3.9 โปรแกรมที่พัฒนาขึ้นเพื่อใช้เป็นเครื่องมือในการวิจัย.....	29
3.10 โปรแกรมสำเร็จรูปที่ใช้สำหรับเปรียบเทียบผลการทดลอง.....	29
4.1 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้า กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนช่วงเวลาที่เท่ากับ 5 ช่วงเวลา.....	37
4.2 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้า กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนช่วงเวลาที่เท่ากับ 10 ช่วงเวลา.....	38
4.3 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้า กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนช่วงเวลาที่เท่ากับ 15 ช่วงเวลา.....	38
4.4 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 3 ชนิด.....	39
4.5 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 5 ชนิด.....	40
4.6 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 10 ชนิด.....	40
4.7 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 15 ชนิด.....	41
4.8 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 20 ชนิด.....	41

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.9 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลา กับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 25 ชนิด.....	42
4.10 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ย ในการหาคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อ ผลรวมของกำลังการผลิตเมื่อจำนวนสินค้าเท่ากับ 15 ชนิด 5 ช่วงเวลา.....	43
4.11 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ย ในการหาคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อ ผลรวมของกำลังการผลิตเมื่อจำนวนสินค้าเท่ากับ 20 ชนิด 5 ช่วงเวลา.....	43
4.12 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ย ในการหาคำตอบกับสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา เมื่อจำนวนสินค้าเท่ากับ 5 ชนิด 5 ช่วงเวลา.....	45
4.13 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่าง ของคำตอบกับจำนวนสินค้าเมื่อช่วงเวลาเท่ากับ 5 ช่วงเวลา.....	46
4.14 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่าง ของคำตอบกับจำนวนช่วงเวลาเมื่อจำนวนสินค้าเท่ากับ 3 ชนิด.....	46
4.15 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่าง ของคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิต ต่อผลรวมของกำลังการผลิต.....	47
4.16 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่าง ของคำตอบกับสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา.....	48

หัวข้อวิทยานิพนธ์	วิธีการหาคำตอบที่เหมาะสมโดยประมาณสำหรับปัญหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกัน
ชื่อผู้เขียน	เรือเอกพงสกร ศรีทองคำ
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ไพฑูรย์ ฮ้อยยิ่ง
อาจารย์ที่ปรึกษาร่วม	ผู้ช่วยศาสตราจารย์ ดร.ศุภรัชชัย วรรัตน์
สาขาวิชา	การจัดการทางวิศวกรรม
ปีการศึกษา	2549

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการแก้ปัญห ปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันแบบหาคำตอบที่เหมาะสมโดยประมาณ และกำหนดให้ทราบปริมาณความต้องการของสินค้าที่แน่นอน โดยมีวัตถุประสงค์ของการวิจัยคือหาปริมาณในการสั่งซื้อหรือผลิตที่เหมาะสมเพื่อลดค่าใช้จ่ายที่เกิดขึ้นจากการสั่งซื้อหรือผลิตและค่าเก็บรักษาสินค้าให้ได้มากที่สุดในขณะที่ใช้เวลาในการคำนวณหาคำตอบน้อยที่สุด

การแก้ปัญห ปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันจะใช้เวลาในการคำนวณสูงหากต้องการหาคำตอบที่ดีที่สุดด้วยโปรแกรมสำเร็จรูปโดยทั่วไป วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการย้ายหรือแบ่งย้ายปริมาณความต้องการของสินค้าเพื่อรวมปริมาณความต้องการของสินค้าเข้าด้วยกันทำให้สามารถลดจำนวนครั้งในการสั่งซื้อหรือผลิตลงโดยพัฒนาโปรแกรมสำเร็จรูป (Visual Basic) ขึ้นมาช่วยในการคำนวณ ผลการทดลองวิธีการที่นำเสนอใช้เวลาเฉลี่ยในการคำนวณหาคำตอบเท่ากับ 5 มิลลิวินาที และมีเปอร์เซ็นต์ความคลาดเคลื่อนจากคำตอบที่ดีที่สุดในกลุ่มปัญหาขนาดเล็กเท่ากับ 1.81 ในกลุ่มปัญหาขนาดกลางเท่ากับ 3.07 ในกลุ่มปัญหาขนาดใหญ่เท่ากับ 1.80 จากการทดลองพบว่าปัจจัยที่มีผลต่อเวลาในการคำนวณหาคำตอบที่ชัดเจนที่สุดได้แก่จำนวนสินค้าและจำนวนช่วงเวลา ซึ่งจะมีความสัมพันธ์แบบแปรผันตรงต่อกัน สำหรับด้านคุณภาพของคำตอบนั้นปัจจัยที่มีผลชัดเจนที่สุดคือสัดส่วนระหว่างค่าการสั่งซื้อหรือผลิตต่อค่าเก็บรักษาซึ่งจะมีความสัมพันธ์แบบแปรผันตรงกับคุณภาพของคำตอบ

Thesis Title	A Heuristic Approach for the Capacitated Multi-item Lot-sizing Problem
Author	Lieutenant Pongsakorn Srithongkam
Thesis Advisor	Assistant Professor Dr.Paitoon Howying
Co-Thesis Advisor	Assistant Professor Dr.Suparatchai Vorarat
Department	Engineering Management
Academic Year	2006

ABSTRACT

This thesis provides a heuristic approach for the capacitated multi-item lot-sizing problem, which assume that the demand is deterministic. The purpose of this thesis is to find an appropriate quantity for procurement and manufacture , which will minimize cost of order and storage, at minimum time of computation.

The optimal value for the capacitated multi-item lot-sizing problem using standard program requires much time for computation. To reduce the operation time of computation, this thesis will provide lot elimination or partial lot elimination method for transferring and grouping demand data for each item and sum all demand in group by using program develop from Visual Basic Program. The result from program developed yield 5 millisecond for computation with percentage deviation equal 1.81 for small group data, 3.07 for medium group data and 1.80 for large group data. We also found that the number of item and the number of period is proportional to the computational time and the ratio between setup cost and holding cost is proportional to the heuristic deviated from the optimal values.

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การดำเนินงานขององค์กรต่างๆไม่ว่าภาครัฐหรือเอกชนผู้บริหารองค์กรจะต้องเข้าไปเกี่ยวข้องกับการบริหารและจัดการในด้านต่างๆมากมายหนึ่งในนั้นได้แก่ การจัดการสินค้าคงคลังซึ่งผู้บริหารจะต้องให้ความสำคัญและดูแลเอาใจใส่อย่างใกล้ชิดเพราะสินค้าคงคลังเป็นสินทรัพย์ที่มีมูลค่าสูงที่สุดในกลุ่มสินทรัพย์หมุนเวียนขององค์กร การจัดการสินค้าคงคลังที่ดีจะช่วยให้องค์กรสามารถดำเนินงานได้อย่างมีประสิทธิภาพและบรรลุตามวัตถุประสงค์ขององค์กรนั้นๆ ดังนั้นผู้บริหารจะต้องศึกษาและหารูปแบบการจัดการสินค้าคงคลังที่เหมาะสมเพื่อนำมาใช้ในการบริหารหรือจัดการสินค้าคงคลังขององค์กรให้เกิดประสิทธิภาพสูงสุด

สินค้าคงคลัง หมายถึงสินค้าหรือวัสดุที่องค์กรต่างๆทั้งภาครัฐและเอกชนจัดเก็บรักษาไว้เพื่อรอการจำหน่ายหรือใช้ประโยชน์ในอนาคต ซึ่งมีความหมายครอบคลุมตั้งแต่สินค้าหรือวัสดุขนาดเล็ก เช่น ปากกา ดินสอ สมุด กระดาษ ไปจนถึงสินค้าหรือวัสดุขนาดใหญ่ประเภท รถยนต์ หรือเครื่องจักรกลต่างๆ ประเภทของสินค้าคงคลังที่องค์กรต่างๆจำเป็นต้องจัดเก็บรักษาไว้สำหรับใช้งานหรือจัดจำหน่ายจะแตกต่างกันไปตามลักษณะของการดำเนินงานขององค์กรนั้นๆ เช่น กองทัพจะมี ลูกปืน ปืนขนาดต่างๆ ลูกกระเบิดมือหรืออาวุธปล่อยนำวิถี เป็นสินค้าคงคลังที่จัดเก็บรักษาไว้ใช้งานในภารกิจการป้องกันประเทศ ในขณะที่ภาคเอกชน เช่น โรงงานผลิตรถยนต์จะมี เหล็กแผ่น เครื่องยนต์ ยางรถยนต์และชิ้นส่วนต่างๆเป็นสินค้าคงคลังสำหรับผลิตรถยนต์ออกจำหน่ายเป็นต้น

ประโยชน์ของสินค้าคงคลังที่องค์กรต่างๆจัดเก็บรักษาไว้นั้นมีมากมาย สินค้าคงคลังแต่ละประเภทก็มีประโยชน์แตกต่างกันไป เช่น ในโรงงานหนึ่งๆ สินค้าคงคลังประเภทวัตถุดิบหรือชิ้นส่วนต่างๆช่วยป้องกันการขาดแคลนวัตถุดิบหรือชิ้นส่วน นอกจากนี้ยังสามารถใช้สำหรับเก็งกำไรในกรณีที่คาดว่าวัตถุดิบหรือชิ้นส่วนอาจขึ้นราคาในอนาคต สินค้าคงคลังประเภทงานระหว่างทำช่วยให้การผลิตสามารถดำเนินการได้อย่างต่อเนื่องโดยไม่ต้องรองานจากหน่วยผลิตก่อนหน้า สินค้าคงคลังประเภทสินค้าสำเร็จรูปจะช่วยป้องกันความผิดพลาดจากการพยากรณ์ยอดขาย สินค้าคงคลังประเภทสินค้าสำเร็จรูปนั้นจำเป็นต้องจัดเก็บรักษาให้สามารถตอบสนองต่อความต้องการของลูกค้าได้อย่างเพียงพอและทันเวลาเพราะหากไม่สามารถตอบสนองต่อความ

ต้องการสินค้าของลูกค้าได้อย่างเพียงพอและทันเวลาแล้วอาจทำให้สูญเสียลูกค้าที่ดีไปและทำให้องค์กรเสียผลประโยชน์ได้

สินค้าคงคลังที่องค์กรต่างๆจัดเก็บรักษาไว้นั้นเชื่อว่าจะมีแต่ประโยชน์อย่างเดียว ข้อเสียของสินค้าคงคลังก็มีเช่นกัน ได้แก่ ค่าใช้จ่ายต่างๆที่เกิดขึ้นมาจากการจัดหาและจัดเก็บรักษาสินค้าคงคลังเอาไว้ ตัวอย่างของค่าใช้จ่ายต่างๆที่เกิดขึ้นมาได้แก่ ค่าใช้จ่ายในการจัดซื้อ ราคาของตัวสินค้าคงคลังเองและค่าใช้จ่ายในการจัดเก็บรักษาเป็นต้น ดังนั้นองค์กรต่างๆจะต้องทำความเข้าใจและบริหารจัดการสินค้าคงคลังให้ดีเพื่อที่จะทำให้องค์กรได้รับประโยชน์จากสินค้าคงคลังมากกว่าโทษจากสินค้าคงคลัง

การจัดการสินค้าคงคลังที่ดีมีส่วนช่วยให้องค์กรสามารถดำเนินงานได้อย่างมีประสิทธิภาพและบรรลุวัตถุประสงค์หรือเป้าหมายขององค์กร แต่ในขณะเดียวกันการจัดการสินค้าคงคลังที่ไม่มีประสิทธิภาพเพียงพออาจเป็นสาเหตุให้องค์กรต้องเลิกกิจการไปเลยก็ได้ ด้วยเหตุนี้ผู้บริหารขององค์กรต่างๆจึงควรให้ความสนใจกับการจัดการสินค้าคงคลังอย่างใกล้ชิด โดยทั่วไปแล้ววัตถุประสงค์หรือจุดมุ่งหมายหลักของการจัดการสินค้าคงคลังคือ ทำอย่างไรจะสามารถลดค่าใช้จ่ายต่างๆในการจัดการสินค้าคงคลังให้มีค่าน้อยที่สุดเท่าที่จะทำได้

ในปัจจุบันองค์กรต่างๆจะต้องทำให้ผู้รับบริการหรือลูกค้าเกิดความพึงพอใจสูงสุดจากเหตุผลดังกล่าว องค์กรจะต้องเสียค่าใช้จ่ายสูงเพื่อจัดเก็บรักษาสินค้าคงคลังจำนวนมากไว้สำหรับตอบสนองต่อความต้องการสินค้าของผู้รับบริการหรือลูกค้าได้อย่างเพียงพอและทันเวลา ซึ่งเป็นสิ่งที่มีมักจะขัดแย้งกับจุดมุ่งหมายของการจัดการสินค้าคงคลัง เพราะถ้าองค์กรต้องการให้ต้นทุนในการจัดการสินค้าคงคลังมีค่าใช้จ่ายต่ำแล้ว ก็จะต้องมีสินค้าคงคลังในปริมาณน้อย แต่การกระทำดังกล่าวก็เสี่ยงต่อการที่สินค้าจะหมดและไม่สามารถส่งมอบสินค้าให้ลูกค้าได้ตามกำหนดเวลา ทำให้ลูกค้าเกิดความไม่พึงพอใจ ดังนั้นการจัดการสินค้าคงคลังจึงไม่ใช่การพยายามทำให้สินค้าคงคลังมีจำนวนน้อยที่สุด หากแต่จะต้องหาปริมาณของสินค้าคงคลังที่เหมาะสมที่ทำให้ต้นทุนรวมหรือค่าใช้จ่ายต่างๆมีค่าน้อยที่สุดในขณะที่องค์กรมีกำไรสูงสุด ฉะนั้นการตัดสินใจขั้นพื้นฐานเกี่ยวกับการแก้ปัญหาสินค้าคงคลังจึงมีอยู่ด้วยกัน 2 ประการคือ จะสั่งซื้อหรือผลิตสินค้าคงคลังครั้งละเท่าไร และจะซื้อหรือสั่งผลิตสินค้าคงคลังจำนวนนี้เมื่อใด ดังนั้นจึงจำเป็นที่ผู้บริหารองค์กรจะต้องมีเทคนิคและวิธีการแก้ไขปัญหาที่เหมาะสมในการหาความสมดุลระหว่างคำตอบของคำถามทั้งสองนี้ เพื่อช่วยให้การบริหารจัดการสินค้าคงคลังเกิดประสิทธิภาพและประสิทธิผลสูงสุดต่อองค์กร การแก้ปัญหาการบริหารจัดการเกี่ยวกับสินค้าคงคลังนั้นได้มีผู้พยายามศึกษา คิดค้น และนำเสนอวิธีการแก้ปัญหาดังกล่าวเรื่อยมาตั้งแต่ในอดีตจนถึงปัจจุบัน ทำให้มีวิธีการแก้ปัญหามากมายและแตกต่างกันไปเช่น วิธีการแก้ปัญหาคือเป็นที่รู้จักกันอย่างแพร่หลายคือ การหา

ปริมาณการสั่งซื้อหรือผลิตแบบประหยัดที่สุด (economic order quantity , EOQ) ที่นำเสนอโดย Harris ในปี 1913 ซึ่งเป็นโมเดลที่เหมาะสมสำหรับการแก้ปัญหาการสั่งซื้อหรือผลิตแบบประหยัดที่สุดของสินค้าคงคลังเมื่อความต้องการสินค้าคงคลังที่ทุกช่วงเวลา ซึ่งไม่สอดคล้องกับความต้องการสินค้าคงคลังที่เกิดขึ้นจริงเท่าใดนัก เนื่องจากในความเป็นจริงความต้องการของสินค้าคงคลังมักเปลี่ยนแปลงไปตามช่วงเวลาต่างๆ Wagner และWhitin (1958 : 89-96) ได้พัฒนาการแก้ปัญหาแบบพลวัต (dynamic programming) ขึ้นมาและสามารถนำมาแก้ปัญหาการสั่งซื้อหรือผลิตแบบประหยัดที่สุดเมื่อความต้องการสินค้าผันแปรตามช่วงเวลาได้ การคำนวณจะหาวิธีทั้งหมดที่จะสามารถเกิดขึ้นได้แล้วเลือกวิธีที่ทำให้ต้นทุนต่ำที่สุด คำตอบที่ได้จากวิธีของ Wagner และ Whitin จึงเป็นคำตอบที่ดีที่สุด (optimization) แต่วิธีดังกล่าวใช้เวลาในการคำนวณหาคำตอบนานมาก Silver และ Meal (1969) จึงได้นำเสนอวิธีการหาคำตอบโดยที่เหมาะสมโดยประมาณ (heuristic) ขึ้น โดยอาศัยหลักการของค่าใช้จ่ายต่ำที่สุดต่อช่วงเวลามาคำนวณหาว่าสมควรที่จะสั่งซื้อหรือผลิตให้เพียงพอกับความต้องการสินค้าในช่วงเวลาถัดไปด้วยหรือไม่ซึ่งใช้เวลาในการคำนวณน้อยกว่าวิธีของ Wagner และ Whitin แต่คำตอบที่ได้เป็นคำตอบที่เหมาะสมโดยประมาณเท่านั้นไม่อาจรับประกันได้ว่าเป็นคำตอบที่ดีที่สุด

ในความเป็นจริงแล้วนอกจากความต้องการสินค้าคงคลังจะผันแปรตามช่วงเวลาแล้วยังมีความต้องการสินค้าคงคลังมากกว่าหนึ่งชนิดด้วย ทำให้การแก้ปัญหาสินค้าคงคลังเป็นแบบหลายชนิดและหลายช่วงเวลา นอกจากนี้ยังมีข้อจำกัดทางด้านการใช้ทรัพยากรร่วมกันอีกด้วย ทำให้การแก้ปัญหามีความซับซ้อนเพิ่มมากขึ้น ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกัน (capacitated multi-item lot-sizing problem , CMLSP) เป็นปัญหาที่พิจารณาการสั่งซื้อหรือผลิตสินค้าคงคลัง N ชนิด ภายใน T ช่วงเวลา (ภัทรธรรณแสงฤดี, 2547 : 2) เพื่อให้สามารถตอบสนองความต้องการสินค้าคงคลังที่เกิดขึ้น โดยมีค่าใช้จ่ายต่ำที่สุดภายใต้เงื่อนไขของข้อจำกัดที่กำหนด ซึ่งจากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่าไม่มีผู้คิดค้นและเสนอวิธีการแก้ปัญหา CMLSP โดยมีวิธีการแก้ปัญหาทั้งโดยการหาคำตอบที่ดีที่สุดและการหาคำตอบที่เหมาะสมโดยประมาณ

สำหรับการแก้ปัญหาโดยการหาคำตอบที่ดีที่สุดนั้นมีขั้นตอนการคำนวณที่ยุ่งยากและซับซ้อนทำให้เสียเวลาในการคำนวณมาก ส่วนวิธีการหาคำตอบที่เหมาะสมโดยประมาณเป็นวิธีการที่มีความเหมาะสมทั้งทางด้านคุณภาพของคำตอบและเวลาที่ใช้ในการหาคำตอบ ปัจจุบันจึงมีผู้สนใจศึกษาค้นคว้าเพื่อให้ได้มาซึ่งวิธีการแก้ปัญหาโดยการหาคำตอบที่เหมาะสมโดยประมาณซึ่งมีอยู่หลากหลายวิธีด้วยกันได้แก่ วิธี lot elimination, meta-heuristic, relaxation heuristic และ decomposition heuristic เป็นต้น ซึ่งแต่ละวิธียังมีข้อจำกัดในเรื่องจำนวนชนิดของสินค้าคงคลัง

และจำนวนช่วงเวลาที่จะมีขนาดไม่มากนัก รวมทั้งระยะเวลาที่ใช้ในการหาคำตอบยังค่อนข้างสูงอยู่ ด้วยเหตุผลดังกล่าวข้างต้น ผู้วิจัยจึงมีความสนใจที่จะศึกษาค้นคว้าหาวิธีการหาคำตอบที่เหมาะสมโดยประมาณสำหรับปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกัน ซึ่งเป็นการพิจารณากำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลัง N ชนิด ภายใน T ช่วงเวลา เพื่อตอบสนองต่อความต้องการสินค้าที่ผันแปรตามช่วงเวลาโดยมีค่าใช้จ่ายในการสั่งซื้อหรือผลิตและเก็บรักษาต่ำที่สุด และอยู่ภายใต้ข้อกำหนดของข้อจำกัดที่ตั้งขึ้น ซึ่งใช้เวลาในการคำนวณน้อยกว่าวิธีแก้ปัญหาโดยการหาคำตอบที่ดีที่สุด ซึ่งเป็นปัญหาที่น่าสนใจที่จะพัฒนาหาแนวทางในการหาคำตอบที่เหมาะสมสำหรับการแก้ไขปัญหา CMLSP ให้ได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดในขณะที่ใช้เวลาในการคำนวณหาคำตอบน้อยกว่า

1.2 วัตถุประสงค์ของงานวิจัย

พัฒนาวิธีการหาคำตอบที่เหมาะสมโดยประมาณสำหรับปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันและอยู่ภายใต้ข้อกำหนดของข้อจำกัดที่ตั้งขึ้น เพื่อตอบสนองต่อความต้องการสินค้าที่เกิดขึ้น โดยจะต้องใช้เวลาในการคำนวณหาคำตอบน้อยและได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุด

1.3 ขอบเขตการวิจัย

1.3.1 ข้อสมมติฐานของงานวิจัยคือ กำหนดให้สินค้าคงคลังทุกชนิดมีความสำคัญเท่ากัน จำนวนสูงสุดของจำนวนสินค้าคงคลังและช่วงเวลาในงานวิจัยนี้คือ 25 ชนิด 15 ช่วงเวลา ตามลำดับ และข้อมูลต่างๆที่ใช้ในการคำนวณทั้งหมดนั้นทราบล่วงหน้า (deterministic)

1.3.2 ดัชนีชี้วัดประสิทธิภาพของวิธีการที่นำเสนอคือ เวลาที่ใช้ในการคำนวณหาคำตอบและคุณภาพของคำตอบเมื่อเทียบกับคำตอบที่ดีที่สุด

1.3.3 เขียนโปรแกรมคอมพิวเตอร์เพื่อแก้ปัญหา CMLSP ตามวิธีการที่ได้นำเสนอ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถหาวิธีการแก้ปัญหา CMLSP ได้

1.4.2 เป็นแนวทางให้กับผู้ที่สนใจนำไปประยุกต์ใช้หรือพัฒนาไปยังปัญหาที่มีความซับซ้อนหรือมีข้อจำกัดที่แตกต่างออกไปจากงานวิจัยนี้และนำไปประยุกต์ใช้ในการทำงานจริงได้

1.5 ขั้นตอนในการวิจัย

1.5.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.5.2 กำหนดปัญหา วัตถุประสงค์ ขอบเขตการศึกษา และข้อสมมติต่างๆ

1.5.3 พัฒนารูปแบบในการหาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมของปัญหา CMLSP

1.5.4 ออกแบบและสร้างโปรแกรมสำหรับแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมของปัญหา CMLSP

1.5.5 ทดสอบวิธีการที่นำเสนอและโปรแกรมที่พัฒนา แก้ไข และปรับปรุงข้อผิดพลาด

1.5.6 สรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้เป็นการศึกษาเกี่ยวกับวิธีการหาคำตอบที่เหมาะสมสำหรับปัญหาปริมาณการตั้งชื่อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกัน เนื้อหาในบทนี้จะกล่าวถึงแนวคิดและทฤษฎีของสินค้าคงคลังและงานวิจัยที่เกี่ยวข้องดังนี้

- 2.1 แนวคิดเกี่ยวกับสินค้าคงคลัง
- 2.2 รูปแบบปัญหาและแนวคิดเกี่ยวกับการจัดการสินค้าคงคลัง
- 2.3 วิธีการแก้ปัญหาของตัวแบบสินค้าคงคลัง
- 2.4 ทฤษฎีการแก้ปัญหาด้วยวิธีค้นหาคำตอบแบบฮิวริสติก
- 2.5 งานวิจัยที่เกี่ยวข้อง
- 2.6 สรุปปัญหา

2.1 แนวคิดเกี่ยวกับสินค้าคงคลัง

สินค้าคงคลังหมายถึง สินค้าหรือวัสดุที่เก็บรักษาไว้เพื่อการใช้งานหรือจำหน่ายในอนาคต (พิชิต สุขเจริญพงษ์, 2547 : 210) ซึ่งในความเห็นของพิภพ สถิตถาวรณ (2545 : 230-230) นั้นจะต้องจัดหาสินค้ามาให้เพียงพอกับความต้องการในช่วงเวลาที่เหมาะสมไม่เช่นนั้นแล้วอาจก่อให้เกิดความเสียหายต่อองค์กรได้ องค์กรต่าง ๆ ไม่ว่าจะเป็นองค์กรเพื่อการผลิตหรือการให้บริการอาจมีสินค้าคงคลังจำนวนมากนับร้อยชนิดตั้งแต่วัสดุชิ้นเล็ก ๆ เช่น ดินสอ ปากกา กระดาษ หมึกเย็บกระดาษ ไปจนถึงวัสดุขนาดใหญ่ เช่น เครื่องจักร รถยนต์ และอุปกรณ์เพื่อการก่อสร้าง โดยทั่วไปสินค้าคงคลังที่องค์กรแต่ละแห่งเก็บไว้ขึ้นอยู่กับลักษณะของกิจการที่องค์กรนั้นทำอยู่ ในองค์กรที่ผลิตผลิตภัณฑ์จะเก็บสินค้าคงคลังในรูปของวัตถุดิบเพื่อใช้ในการผลิต ชิ้นส่วนของเครื่องจักรเครื่องมือเพื่อใช้ทดแทนในกรณีที่เกิดการเสียหาย ผลิตภัณฑ์สำเร็จรูปซึ่งอยู่ในระหว่างการผลิตและสินค้าสำเร็จรูปที่ผลิตเสร็จเรียบร้อยแล้วรอการจำหน่าย ห้างสรรพสินค้าจะต้องเก็บสินค้าคงคลังในรูปของสินค้าสำเร็จรูปเพื่อรอการจำหน่าย ส่วนโรงพยาบาลต้องเก็บสินค้าคงคลังในรูปของยารักษาโรค , เครื่องมือทางการแพทย์และอุปกรณ์อื่น ๆ ที่มีความจำเป็นสำหรับการดูแลและรักษาพยาบาลคนไข้ภายในโรงพยาบาลให้ทันกับความต้องการใช้งานซึ่งบางครั้งไม่สามารถคาดการณ์ล่วงหน้าได้ว่าจะต้องใช้เมื่อใดและเป็นจำนวนเท่าไรได้เป็นต้น

พิภพ สถิตาภรณ์ (2545 : 230) จำแนกประเภทของสินค้าคงคลังที่เก็บไว้ในองค์กรหรือหน่วยงานใด ๆ ออกเป็น 4 ประเภทดังนี้

1. วัตถุดิบ (raw material) หมายถึงสิ่งของรายการต่างๆ ที่สั่งซื้อมาเพื่อนำมาทำการผลิตให้เป็นสิ่งของที่ต้องการซึ่งอาจจะเป็นสินค้าสำเร็จรูปหรือชิ้นส่วนประกอบ เช่น แร่ เหล็ก แผ่นไม้ ผ้า พลาสติก ยางดิบ เป็นต้น
2. ชิ้นส่วนประกอบ (component) หมายถึง สิ่งต่าง ๆ ที่อาจซื้อหรือผลิตขึ้นเองเพื่อนำมาเป็นชิ้นส่วนประกอบของสินค้าสำเร็จรูปหรือเพื่อเป็นอะไหล่ซ่อมแซมของเก่า เช่น ลูกสูบ นี้อด เกียร์ ยางรถยนต์ อะไหล่รถยนต์หรือชิ้นส่วนวิทยุ เป็นต้น
3. วัสดุสิ้นเปลือง (suppliers) หมายถึง สิ่งที่ใช้หมดไปในการผลิตแต่มิได้กลายเป็นส่วนหนึ่งของสินค้าสำเร็จรูป ส่วนของวัสดุสิ้นเปลืองจะเป็นส่วนช่วยให้การผลิตดำเนินไปได้ อย่างราบรื่น เช่น น้ำมันหล่อลื่น จาระบี กระดาษทราย สบู่ล้างมือ เป็นต้น
4. งานระหว่างทำ (work in process) หมายถึง ผลิตภัณฑ์ที่ยังไม่เป็นสินค้าสำเร็จรูป ยังค้างอยู่ในระหว่างขั้นตอนการผลิตเพื่อรอคอยการผลิตขั้นต่อไปเพื่อให้เป็นสินค้าสำเร็จรูป
5. สินค้าสำเร็จรูป (finished goods) หมายถึง สินค้าต่างๆ ที่ทำเสร็จแล้ว พร้อมทั้งจะส่งออกขายได้ตลอดเวลา เช่น เสื้อผ้าสำเร็จรูป รถยนต์ พัดลม ตู้เย็น เป็นต้น

ความจำเป็นและความสำคัญของสินค้าคงคลัง พิภพ สถิตาภรณ์ (2545 : 231-232) พบว่าสินค้าคงคลังในองค์กรทุกแห่งช่วยเพิ่มความยืดหยุ่นให้กับการดำเนินงานโดยเฉพาะในโรงงานอุตสาหกรรมสินค้าคงคลังเป็นส่วนที่มีความสำคัญมากซึ่งพอจะสรุปได้ดังนี้

1. สินค้าคงคลังที่เป็นสินค้าสำเร็จรูป จะช่วยป้องกันความผิดพลาดอันเกิดจากความต้องการสินค้าที่มีมากกว่าที่พยากรณ์ไว้ การผิดพลาดจะไม่ได้รับการตอบสนองถ้ากิจการไม่มีสินค้าคงคลังที่เป็นสินค้าสำเร็จรูปเก็บไว้ ที่ให้ธุรกิจต้องขาดกำไรที่ควรจะได้ไป และอาจทำให้ความเชื่อถือของลูกค้าที่มีต่อธุรกิจลดลง และในกรณีที่รุนแรงก็อาจทำให้ลูกค้าหันไปซื้อสินค้าจากคู่แข่งได้ แต่ถ้าเรามีสินค้าคงคลังเก็บไว้จำนวนหนึ่งก็จะทำให้ความเสียหายดังกล่าวบรรเทาลงได้ ช่วยให้การผลิตสามารถดำเนินไปได้อย่างสม่ำเสมอ ไม่ต้องเปลี่ยนแปลงไปตามฤดูกาลเช่นเดียวกับความต้องการสินค้า โรงงานไม่จำเป็นต้องหยุดงานหรือเปลี่ยนแปลงจำนวนคนงานบ่อย ๆ ซึ่งจะช่วยให้ประหยัดต้นทุนในการผลิตและการดำเนินงาน ทั้งยังช่วยให้มีการใช้ประโยชน์จากเครื่องจักร เครื่องมือและอุปกรณ์การผลิต อาคาร และกำลังคนได้อย่างมีประสิทธิภาพเช่น ผลิตสินค้าคงคลังเก็บไว้ในช่วงระหว่างมีเวลาว่างเพื่อจำหน่ายในช่วงที่มีความต้องการสินค้าสูง โดยไม่ต้องเร่งการผลิตสินค้าหรือทำการผลิตนอกเวลา

2. สินค้าคงคลังงานระหว่างทำ ช่วยให้การผลิตในแต่ละหน่วยการผลิตสามารถดำเนินไปได้อย่างต่อเนื่องไม่จำเป็นต้องพึ่งพิงกันมากนักตัวอย่างเช่น การผลิตจากหน่วยผลิตที่หนึ่งแล้วส่งต่อไปยังหน่วยผลิตที่สอง หากการทำงานในหน่วยที่หนึ่งต้องหยุดชะงักลงก็ทำให้งานหน่วยผลิตที่สองต้องหยุดชะงักไปด้วย ถ้าเราให้หน่วยงานแรกทำงานเกินไว้ส่วนหนึ่งซึ่งเรียกว่า buffer stock จะช่วยทำให้งานในหน่วยผลิตที่สองดำเนินต่อไปได้ถึงแม้ว่าหน่วยผลิตแรกจะหยุดชะงักไปชั่วคราว นอกจากนี้ยังช่วยให้การผลิตสามารถดำเนินไปได้อย่างต่อเนื่องสม่ำเสมอถึงแม้ว่าการทำงานในแต่ละหน่วยผลิตจะมีความเร็วไม่เท่ากัน

3. สินค้าคงคลังที่เป็นวัตถุดิบหรือชิ้นส่วนที่ซื้อจากที่อื่น ช่วยป้องกันการขาดแคลนวัตถุดิบหรือชิ้นส่วนอันเนื่องมาจากการล่าช้าด้วยเหตุผลหลายอย่างเช่น การเปลี่ยนแปลงกำหนดเวลาในการขนส่งของผู้ขาย ผู้ขายขาดแคลนสินค้าไม่สามารถจัดส่งมาให้ได้หรือเกิดการนัดหยุดงานที่โรงงานของผู้ขายหรือเกิดอุทกภัยเป็นต้น ด้วยเหตุนี้จึงต้องมีวัตถุดิบคงเหลือไว้ให้เพียงพอ นอกจากนี้ยังช่วยลดค่าใช้จ่ายในการสั่งซื้อหรือผลิต เพราะการสั่งการสั่งซื้อหรือผลิตครั้งละจำนวนมาก ๆ ราคาต่อหน่วยมักจะลดลงตัวอย่างเช่น ถ้าเราซื้อวัตถุดิบมาเพื่อใช้ในการผลิตเป็นระยะเวลา 30 วัน จะประหยัดกว่าการซื้อวัตถุดิบมาเพื่อการผลิตหนึ่งวัน นอกจากนี้การมีวัตถุดิบคงเหลือเก็บไว้ยังช่วยป้องกันการขาดทุนที่อาจจะเกิดขึ้นอันเนื่องมาจากวัตถุดิบราคาสูงขึ้นก็ได้

นอกจากข้อดีของการมีสินค้าคงคลังดังกล่าวมาแล้วนั้นย่อมมีข้อเสียในด้านค่าใช้จ่ายต่าง ๆ เช่น ค่าเช่าพื้นที่สำหรับเก็บของเหล่านั้น ค่าใช้จ่ายคนคอยดูแลรักษาและทำบัญชีควบคุมปริมาณ และที่สำคัญที่สุดคือเงินทุนที่จะต้องมาจอมอยู่กับของเหล่านั้นโดยไม่สามารถใช้ประโยชน์ได้ทันที ฉะนั้นปริมาณการจัดเก็บสินค้าคงคลังที่เหมาะสมจึงขึ้นอยู่กับการเปรียบเทียบระหว่างข้อดีและข้อเสียของการมีสินค้าคงคลังเพื่อก่อให้เกิดประโยชน์สูงสุด

2.2 รูปแบบปัญหาและการจัดการสินค้าคงคลัง

ถ้าองค์กรต้องการให้ต้นทุนในการเก็บสินค้าคงคลังต่ำก็ต้องมีสินค้าคงคลังในปริมาณน้อย การกระทำเช่นนี้เสี่ยงต่อการที่สินค้าจะหมดและไม่สามารถส่งมอบสินค้าให้ลูกค้าตามกำหนดได้ ทำให้ลูกค้าเกิดความไม่พึงพอใจ ดังนั้น พิชิต สุขเจริญพงษ์(2547 : 212)สรุปว่าการตัดสินใจขั้นพื้นฐานเกี่ยวกับการแก้ปัญหาสินค้าคงคลังจึงมีอยู่ด้วยกัน 2 ประการคือ จะสั่งซื้อหรือผลิตสินค้าคงคลังครั้งละเท่าไร และจะซื้อหรือสั่งผลิตสินค้าคงคลังจำนวนนี้เมื่อใด ในการแก้ไขปัญหาทั้งสองประการนี้มักจะเข้าไปในทางตรงข้ามเพราะหากต้องการทำให้ต้นทุนในการสั่งซื้อ

หรือผลิตต่ำปริมาณในการสั่งซื้อหรือผลิตแต่ละครั้งต้องมาก แต่หากต้องการให้มีต้นทุนในการจัดให้มีสินค้าคงคลังต่ำปริมาณที่สั่งในแต่ละครั้งต้องน้อย ดังนั้นจึงจำเป็นต้องทราบเกี่ยวกับค่าใช้จ่ายต่างๆที่เกี่ยวข้องกับการจัดการสินค้าคงคลัง วิธีการหาปริมาณของผลิตภัณฑ์ที่จะสั่งซื้อหรือผลิตแต่ละครั้ง (lot size) ที่จะทำให้ค่าใช้จ่ายน้อยที่สุด

ต้นทุนในการจัดการสินค้าคงคลัง (Inventory management cost) การจัดการสินค้าคงคลังเป็นขบวนการหนึ่งในระบบการจัดการการผลิตหรือบริการ ดังนั้นนโยบายในการจัดการสินค้าคงคลังจะต้องสอดคล้องกับแผนการผลิตรวม (Aggregate production planning) ขององค์กร และในทำนองเดียวแผนการผลิตรวมก็ต้องสอดคล้องกับความต้องการของลูกค้าเช่นเดียวกัน ในการจัดการสินค้าคงคลังไม่ว่าจะเป็นสินค้าคงคลังประเภทใด ๆ ใน 5 ประเภทหลัก เช่น วัตถุดิบ ชิ้นส่วนประกอบและสินค้าสำเร็จรูปนั้นจำเป็นต้องมีต้นทุนด้วยกันทั้งสิ้น ซึ่งพิชิต สุขเจริญพงษ์ (2547 : 212) ได้จัดกลุ่มและจำแนกออกเป็น 4 ประเภทหลักดังนี้

1. ต้นทุนสินค้าคงคลัง (item cost) คือ ราคาหรือมูลค่าของสินค้าคงคลังเอง หากราคาต่อหน่วยของสินค้าคงคลังคงที่ตลอดไม่ว่าจะสั่งซื้อหรือผลิตด้วยจำนวนเท่าใด ต้นทุนสินค้าคงคลังจะไม่ถูกนำมาพิจารณาในการตัดสินใจกำหนดปริมาณการสั่งซื้อหรือผลิต แต่ถ้าราคาต่อหน่วยของสินค้าคงคลังผันแปรตามปริมาณการสั่งซื้อหรือผลิตแล้ว จะต้องนำต้นทุนสินค้าคงคลังมาพิจารณาประกอบในการตัดสินใจกำหนดปริมาณการสั่งซื้อหรือผลิต สำหรับต้นทุนสินค้าคงคลังนี้จะคำนวณออกมาเป็นตัวเลขต่อปี

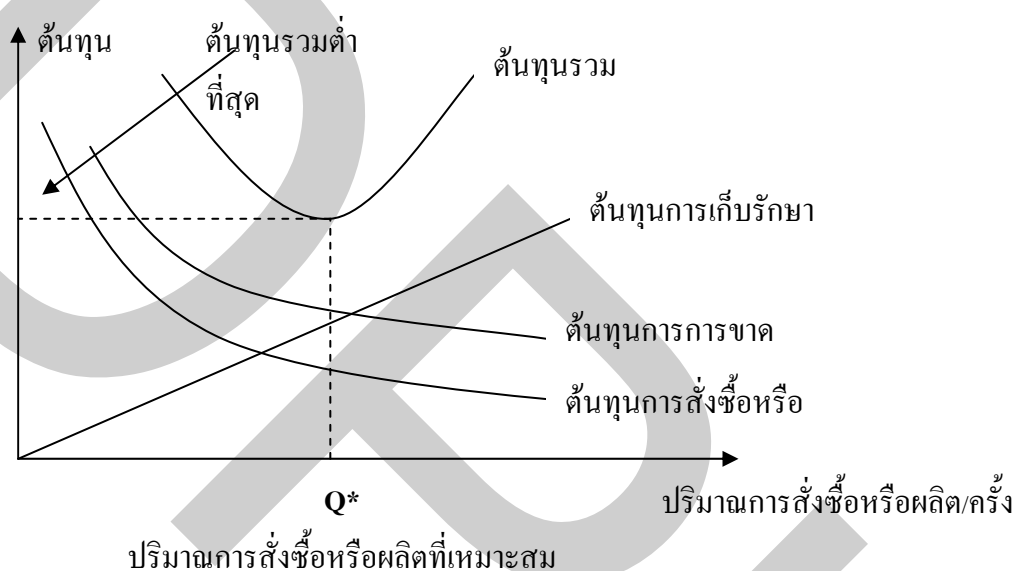
2. ต้นทุนการสั่งซื้อหรือผลิต (procurement of ordering cost) คือ ต้นทุนที่เกิดขึ้นเมื่อมีการสั่งซื้อหรือผลิตสินค้า เช่น ค่าโทรศัพท์ โทรเลข ค่าพิมพ์ใบสั่งของ ค่าขนส่งสินค้า ค่าจดหมาย และค่าไปรษณีย์ เป็นต้น ต้นทุนการสั่งซื้อหรือผลิตจะแปรผันตามจำนวนการสั่งซื้อหรือผลิต และผกผันกับปริมาณการเก็บของสินค้าคงคลัง

3. ต้นทุนการเก็บรักษาสินค้าคงคลัง (carrying cost) คือ ต้นทุนหรือค่าใช้จ่ายในการเก็บรักษาสินค้าคงคลังให้พร้อมใช้งาน ต้นทุนส่วนนี้จะประกอบไปด้วย ค่าดอกเบี้ยที่ต้องจ่ายไปเพื่อซื้อสินค้ามาเก็บไว้สำหรับใช้งาน หรือการขาดโอกาสที่จะได้ดอกเบี้ยถ้านำเงินไปฝากธนาคาร ค่าสถานที่สำหรับจัดเก็บสินค้าคงคลัง เช่น ค่าเช่า ค่าน้ำ ค่าไฟ ค่าจ้างคนดูแลรักษา ค่าขนส่งเคลื่อนย้าย ค่าประกันภัย ค่าภาษี ตลอดจนค่าเสื่อมหรือความเสียหายของสินค้าคงคลัง

4. ต้นทุนการขาดแคลน (shortage cost) คือ ค่าใช้จ่ายที่เกิดขึ้นเมื่อขาดแคลนสินค้าคงคลัง ต้นทุนส่วนนี้ประกอบด้วย การขาดโอกาสในการทำกำไรเพราะไม่มีสินค้าจำหน่าย ค่าปรับเนื่องผิดนัดการส่งมอบสินค้า ต้นทุนส่วนนี้จะแปรผกผันกับปริมาณของสินค้าคงคลัง

การหาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสม จากต้นทุนในการจัดการสินค้าคงคลังทั้ง 4 ประเภทที่กล่าวมาแล้วข้างต้นนั้น สามารถเขียนสมการแสดงต้นทุนรวมได้ดังนี้

$$\text{ต้นทุนรวม} = \text{ต้นทุนสินค้าคงคลัง} + \text{ต้นทุนการสั่งซื้อหรือผลิต} + \text{ต้นทุนการเก็บรักษาสินค้าคงคลัง} + \text{ต้นทุนการขาดแคลน}$$



ภาพที่ 2.1 กราฟแสดงความสัมพันธ์ของต้นทุนทั้ง 4 ประเภท

จากกราฟในภาพที่ 2.1 จุด Q^* คือปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมและมีต้นทุนต่ำที่สุด ซึ่งก็คือจุดต่ำสุดของกราฟของต้นทุนรวม และสามารถหาปริมาณของปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมได้โดยทำการหาอนุพันธ์ของสมการต้นทุนรวมแล้วเทียบกับปริมาณความต้องการของสินค้าแล้วหามีค่าเท่ากับศูนย์ แล้วแทนค่าตัวแปรต่าง ๆ ที่ทราบค่าจะได้ปริมาณการสั่งซื้อหรือผลิตที่เหมาะสม ด้วยหลักการนี้เองทำให้สามารถสร้างตัวแบบสินค้าคงคลังซึ่งเป็นสมการคณิตศาสตร์ในรูปแบบต่าง ๆ ที่สอดคล้องกับปัญหาในความเป็นจริงและหาคำตอบที่เหมาะสมได้

ตัวแบบสินค้าคงคลัง (inventory model) คือ สมการคณิตศาสตร์ที่ใช้สำหรับวิเคราะห์หาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมที่สุดหรือประหยัดที่สุด และช่วงเวลาที่เหมาะสมในการสั่งซื้อหรือผลิต พิชิต สุขเจริญพงษ์ (2547 : 214) จำแนกตัวแบบสินค้าคงคลังออกเป็น 2 ประเภทด้วยกันคือ

1. ตัวแบบดีเทอร์มิเนติก (deterministic model) คือตัวแบบที่ใช้หาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมที่สุด เมื่อตัวแปรต่าง ๆ เช่น ปริมาณความต้องการสินค้า เวลาการส่งมอบสินค้า และต้นทุนต่าง ๆ มีค่าคงที่แน่นอน

2. ตัวแบบสโตคาสติก (stochastic model) คือตัวแบบที่ใช้หาปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมที่สุด เมื่อตัวแปรต่าง ๆ มีค่าไม่แน่นอนและมีความน่าจะเป็นทางสถิติเข้ามาเกี่ยวข้อง

ในงานวิจัยนี้จะกล่าวถึงเฉพาะตัวแบบดีเทอร์มิเนติกเท่านั้น ซึ่งก็เป็นตัวแบบเป็นที่รู้จักกันโดยทั่วไปเช่น ตัวแบบการสั่งซื้อที่ประหยัดขั้นพื้นฐาน (EOQ), ตัวแบบสินค้าคงคลังเมื่อการส่งสินค้าเข้ามาไม่พร้อมกัน, ตัวแบบสินค้าคงคลังเมื่อมีส่วนลดเป็นต้น

วิธีการสั่งสินค้าคงคลัง การสั่งสินค้าคงคลังเพื่อนำมาเก็บรักษาไว้เพื่อใช้งานหรือรอจำหน่ายที่นิยมใช้กันโดยทั่วไปซึ่งจำแนกโดย พิชิต สุขเจริญพงษ์ (2547 : 256-258) มีอยู่ด้วยกัน 2 แบบดังนี้

1. สั่งซื้อหรือผลิตสินค้าเมื่อปริมาณของสินค้าคงคลังถึงจุดสั่งซื้อหรือผลิต การสั่งซื้อหรือผลิตด้วยวิธีการนี้จะซื้อสินค้าจำนวน Q^* (จำนวนสั่งซื้อหรือผลิตที่เหมาะสมคำนวณจาก EOQ) เมื่อปริมาณสินค้าคงคลังถึงจุดสั่งซื้อหรือผลิต (reorder point) สำหรับช่วงเวลาในการสั่งซื้อหรือผลิตแต่ละครั้งห่างไม่เท่ากัน

2. สั่งซื้อหรือผลิตสินค้าด้วยช่วงระยะเวลาที่กำหนดแน่นอน การสั่งซื้อหรือผลิตสินค้าด้วยวิธีนี้คือกำหนดระยะเวลาห่างของการซื้อแต่ละครั้งแน่นอนตายตัว เช่น ซื้อสินค้าทุก ๆ 3 เดือน ซึ่งในบางทีก็เป็นวิธีที่เหมาะสมในทางปฏิบัติเพราะผู้ผลิตบางรายผลิตสินค้าเป็นช่วง ๆ การซื้อด้วยวิธีนี้จำนวนการสั่งซื้อหรือผลิตจะไม่คงที่ตายตัวขึ้นอยู่กับจำนวนสินค้าคงคลังที่มีอยู่และช่วงเวลาในการซื้อในครั้งถัดไป ข้อดีของวิธีการนี้คือลดภาระในการตรวจสอบสินค้าคงคลัง

2.3 วิธีการแก้ปัญหาของตัวแบบสินค้าคงคลัง

การแก้ปัญหาของตัวแบบสินค้าคงคลังในรูปแบบต่าง ๆ จะมีความซับซ้อนและใช้เวลาในการหาคำตอบสูงเมื่อนขนาดของปัญหาใหญ่ขึ้นเช่น การพิจารณาสินค้าหลายชนิด, การพิจารณาสินค้าชนิดเดียวหลายช่วงเวลาและการพิจารณาสินค้าหลายชนิดหลายช่วงเวลาเป็นต้น สำหรับรูปแบบปัญหาในงานวิจัยครั้งนี้คือการพิจารณาสินค้าหลายชนิดหลายช่วงเวลาซึ่งจัดว่าเป็นปัญหาแบบ discrete optimization model และสามารถเขียนให้อยู่ในรูปของ Integer linear programming

ได้ซึ่งหากขนาดของปัญหามีขนาดใหญ่จะทำให้ใช้เวลาในการคำนวณสูงทำให้มีแนวทางการแก้ปัญหาในลักษณะนี้ 2 แนวทางด้วยกันคือ

1. การหาคำตอบที่ดีที่สุด วิธีการนี้จะให้คำตอบที่เป็นคำตอบที่ดีที่สุดหรืออาจกล่าวได้ว่าไม่สามารถหาคำตอบที่ดีกว่านี้ได้อีกแล้ววิธีการหาคำตอบที่ดีที่สุดที่รู้จักกันโดยทั่วไปเช่น Branch and bound และ Simplex เป็นต้น ข้อเสียของการหาคำตอบแบบนี้คือใช้เวลาการคำนวณหาคำตอบสูงมากเมื่อปัญหามีขนาดใหญ่

2. การหาคำตอบที่เหมาะสมโดยประมาณ คือเป็นวิธีการหาคำตอบโดยอาศัยข้อมูลในการเปลี่ยนจากสถานะปัจจุบันไปเป็นสถานะเป้าหมายเพื่อให้ได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดให้มากที่สุด ในขณะที่ใช้เวลาในการคำนวณหาคำตอบน้อยมาก

2.4 ทฤษฎีการแก้ปัญหาด้วยวิธีค้นหาคำตอบแบบฮิวริสติก

ภัทรธอร์ แสงฤดี (2547 : 12-16) ทำการศึกษาวิธีการของการค้นหาคำตอบแบบฮิวริสติก นั้นมีความหลากหลายและมีความเหมาะสมกับลักษณะปัญหาที่แตกต่างกันและจำแนกได้ดังนี้

2.4.1 Relaxation heuristic เป็นวิธีการแก้ปัญหาย่างง่ายของปัญหาแบบ Exact optimization อาศัยการลดความยุ่งยากของการแก้ปัญหาบางส่วน เช่น การ relax ค่า optimum โดยการปัดตัวเลข (rounding) เพื่อให้สามารถหาคำตอบได้ง่ายและรวดเร็วขึ้น lagrangean relaxation ทำการแก้ปัญหาโดยแบ่งปัญหาวางออกเป็นส่วน ๆ และหาคำตอบสำหรับแต่ละส่วนโดยอาศัยชุดคำตอบจากปัญหาย่อยเพื่อปรับปรุงค่าที่ดีที่สุด ซึ่งวิธี Relaxation heuristic จะไม่เหมาะกับปัญหาที่มีความซับซ้อนในการสร้างโมเดลหรือปัญหาที่เป็น integer-infeasible

2.4.2 Decomposition heuristic เป็นการแบ่งปัญหาวางออกเป็นส่วน ๆ และแก้ปัญหาแต่ละส่วนแยกจากกัน ซึ่งมีวิธีการแก้ปัญหามากมายแบบ ได้แก่ iteration ซึ่งจะหาคำตอบสำหรับปัญหาย่อยโดยคงค่าของตัวแปรอื่น ๆ และปรับปรุงค่าของตัวแปรตัดสินสำหรับชุดคำตอบย่อย ๆ นั้น column generation เป็นวิธีที่เหมาะสมเมื่อปัญหามีแง่มุมที่แสดงในรูปแบบทางคณิตศาสตร์ได้ยาก

2.4.3 Constructive search แก้ปัญหาโดยอาศัย null solution และทำการแก้ปัญหาแบบครั้งต่อครั้ง โดยอาศัยข้อมูลจากการแก้ปัญหาครั้งก่อน เพื่อให้ได้คำตอบของปัญหา (full solution) โดยการหาคำตอบของวิธี Constructive search มีลักษณะที่คำนึงถึงผลประโยชน์เฉพาะหน้าในการตัดสินใจแต่ละครั้งจะทำตาม greedy way ซึ่งจะเลือกคำตอบที่ดีที่สุดจากคำตอบปัจจุบันที่มีอยู่ ซึ่งมีข้อสังเกตหนึ่งของวิธีการนี้คือ คำตอบที่ได้จะขึ้นอยู่กับลำดับของการแก้ปัญหา ดังนั้นหากการแก้ปัญหาล่วงหน้าให้คำตอบที่ไม่ดีก็จะส่งผล (degrade) ต่อคำตอบในครั้งถัดไป

2.4.4 Improving search การหาคำตอบจะเริ่มตั้งแต่คำตอบเบื้องต้นที่มีจำนวนตัวแปรอยู่ครบทุกตัว (full solution) และจะทำการปรับปรุงคำตอบโดยการปรับค่าตัวแปรตามแนวทางการปรับค่าของตัวแปร (move set or neighborhood) ซึ่งทำให้คำตอบของปัญหานี้ขึ้นอยู่กับทางเลือกคำตอบเบื้องต้น โดยข้อเสียของวิธีการนี้คือ หากปัญหามีขนาดของ neighborhood ใหญ่เกินไป จะทำให้การปรับปรุงค่าในแต่ละครั้งไม่มีประสิทธิภาพแต่หาก neighborhood มีขนาดเล็กเกินไปจะทำให้มีโอกาสที่จะไม่ครอบคลุมคำตอบที่ดีที่สุดทั้งหมด นอกจากนี้สำหรับการปรับปรุงค่าในแต่ละครั้งควรจะมีความเป็นไปได้ของคำตอบ (feasibility) และหากจำกัดแนวทางการปรับปรุงค่าเพื่อให้คำตอบเป็นไปได้อย่างทุกครั้งที่ทำให้มีความยุ่งยากในการนำไปใช้มากขึ้น ดังนั้นแนวทางในการปรับปรุงค่าจึงยอมให้เกิดการปรับปรุงคำตอบที่เป็นไปไม่ได้ (infeasibility) แต่จะทำการหักค่าของคำตอบในฟังก์ชันวัตถุประสงค์ลง

2.4.5 Local search หรือ Hill climbing เป็นวิธีการที่ประยุกต์มาจาก Improving search โดยเริ่มต้นจาก initial feasibility solution จากนั้นทำการปรับปรุงค่าโดยพิจารณาตาม neighborhood หากคำตอบที่ได้มีค่าที่ดีขึ้นก็จะทำ iteration ซ้ำๆ ต่อไป แต่หากคำตอบที่ได้ไม่ดีขึ้นจะหยุดการทำซ้ำ โดยได้คำตอบแบบ local optimum ซึ่งวิธีการหนึ่งในการแก้ปัญหาคำตอบที่วนอยู่ใน local optimum คือ วิธีการ multistart ซึ่งจะเลือกจุดสำหรับใช้เป็น initial feasibility solution หลาย ๆ จุด ซึ่งแต่ละจุดจะทำให้ได้ค่า local optimum ที่แตกต่างกันโดยจะใช้ค่า local optimum ที่ดีที่สุด

2.4.6 Tabu search เป็นการขยายแนวคิดในการหาคำตอบจากวิธีการแบบ Local search โดยยอมให้สามารถย้ายค่าไปในตำแหน่งที่ไม่ปรับปรุงคำตอบให้ดีขึ้นได้ (nonimproving move) แต่ด้วยวิธีการนี้อาจนำไปสู่การปรับปรุงค่าโดยการย้ายอย่างไม่สิ้นสุด (infinite cycling) และการปรับปรุงค่าครั้งต่อไปจะทำให้คำตอบย้ายกลับไปสู่จุดเดิม ดังนั้น Tabu search จึงมีกลไกเพื่อป้องกันการเกิดปัญหาที่จะเกิดขึ้น โดยการห้ามปรับปรุงค่าบางแนวทาง (Tabu move) รวมถึงการปรับปรุงค่าที่จะทำให้คำตอบย้ายเข้าสู่ตำแหน่งก่อนหน้า เป็นการป้องกันการเกิดการวนรอบ (Short term cycling) โดยใช้ Tabu list บันทึกการย้ายตำแหน่งและเก็บคำตอบไว้เสมอ ดังนั้นทุก ๆ Iteration จะเกิดการปรับปรุงค่าคำตอบอย่าง Tabu แม้ว่าคำตอบจะไม่ดีขึ้น แต่ยังคงเก็บคำตอบจากทุก ๆ Iteration ไว้และเลือกค่าที่ดีที่สุดจากจำนวนการปรับปรุงค่าที่ตั้งไว้เป็น Heuristic optimum พารามิเตอร์ที่สำคัญของวิธีการนี้คือ การกำหนดขนาดของ Tabu list (Tabu List Size) ซึ่งค่าที่เหมาะสมขึ้นอยู่กับปัญหา ถ้าขนาดของ Tabu list มากเกินไปจะทำให้คำตอบที่ได้มีค่าไม่ดี แต่ถ้าขนาดของ Tabu list น้อยเกินไปจะทำให้คำตอบยังคงวนอยู่ในค่า local optimum

2.4.7 Simulated annealing (SA) อัลกอริทึมของ Simulated annealing คือ การยอมรับการเคลื่อนที่ที่ไม่ได้ปรับปรุงค่าคำตอบให้ดีขึ้นด้วยการทดสอบความน่าจะเป็นจากการสุ่มเพื่อหลีกเลี่ยงการวนรอบเริ่มต้นจาก feasibility solution กระบวนการเคลื่อนที่ของแต่ละ Iteration เริ่มต้นจากค่าสุ่ม ซึ่งได้มาจากการเคลื่อนที่ครั้งก่อนหน้า ซึ่งโดยทั่วไปไม่ได้พิจารณาเพียงค่าฟังก์ชันวัตถุประสงค์ (Objective function) แต่จะคำนวณเป็นค่าฟังก์ชันวัตถุประสงค์สุทธิ (Net objective function improvement ; Δ_{obj}) ถ้าคำตอบที่ได้ไม่ดีขึ้น ($\Delta_{obj} > 0$) จะยอมรับคำตอบนั้นทันที หรือในกรณีที่คำตอบมีค่าไม่ดีขึ้น ($\Delta_{obj} < 0$) จะยอมรับคำตอบด้วยความน่าจะเป็น ดังนี้

$$\text{Probability of acceptance} = e^{-\frac{\Delta_{obj}}{q}}$$

ค่าที่ดีที่สุดจะถูกเก็บไว้เป็นค่า heuristic optimum ซึ่งค่าการลดค่าคำตอบ ($\Delta_{obj} > 0$) ในความน่าจะเป็นแบบเอ็กซ์โปเนนเชียลจะแสดงถึงปริมาณที่ลดลงของค่าฟังก์ชันวัตถุประสงค์เทียบกับคำตอบจากการปรับปรุงค่า ดังนั้นการปรับปรุงค่าครั้งใด ๆ ที่ให้ค่าคำตอบน้อยย่อมถูกยอมรับด้วยโอกาสน้อยกว่าการปรับปรุงที่มีค่ามากกว่าเสมอ พารามิเตอร์ q เรียกว่า ค่าอุณหภูมิ (Temperature controlling) ของการค้นหาคำตอบเป็นตัวกำหนดความถี่ของการปรับปรุงค่า ถ้า q มีค่ามากจะทำให้สมการมีค่าเข้าใกล้ศูนย์ หมายความว่าโอกาสที่จะยอมรับความคลาดเคลื่อนที่เท่ากับ 1 เสมอแม้จะไม่มีปรับปรุงค่าคำตอบ และถ้า q มีค่าน้อยโอกาสที่จะยอมรับการเคลื่อนที่ที่ไม่มีการปรับปรุงค่าคำตอบก็จะลดลง

การนำ Simulated annealing ไปใช้ในการหาคำตอบจะเริ่มต้นจาก q ที่มีค่ามากก่อนแล้วค่อย ๆ ลดค่าลงจนเป็นศูนย์ โดยแนวคิดของวิธีการนี้คือ ให้การหาคำตอบสามารถขยายการค้นหาคำตอบ (Explorer) ได้อย่างทั่วถึงในขั้นแรกและจะจำกัดพื้นที่สำหรับการค้นหาลงมาให้เหลือเฉพาะพื้นที่ที่มีศักยภาพในภายหลัง (Exploit) และโดยปกติแล้วจำนวนครั้งของการปรับปรุงค่าโดยวิธีการนี้จะสูงกว่าวิธี Tabu search หรือ Hill climbing และมีการปฏิเสธการปรับปรุงค่าน้อยกว่า รวมถึงใช้ neighborhood จำนวนน้อยกว่าในการพิจารณา เมื่อเทียบกับวิธีแบบ Local search ที่มีทดสอบกับ neighborhood เกือบทุกค่า จากการศึกษาพบว่าจำนวนครั้งของการปรับปรุงค่าอย่างสุ่มที่มากพอจะทำให้ Simulated annealing หาคำตอบ heuristic optimum ที่ดีได้

2.4.8 Genetic algorithms (GAs) เป็นวิธีการแก้ปัญหาที่พัฒนามาจากวิธีแบบ Evolutionary algorithm ซึ่งใช้แก้ปัญหาประเภท Stochastic optimization โดยประยุกต์หลักการคัดเลือกโดยธรรมชาติและหลักการพันธุศาสตร์ในการหาคำตอบ เริ่มต้นโดยการสุ่มชุดของ Initial solution

ขึ้นมาเป็นประชากร (Population) และสำหรับแต่ละ Initial solution ซึ่งเรียกว่าโครโมโซม (Chromosome) ประกอบไปด้วยคำตอบของปัญหานั้น ๆ โดยการปรับปรุงชุดคำตอบ (Population solution) ตลอดจนการค้นหาคำตอบ โดยประชากรบางส่วนจะถูกเลือกโดยคำนึงถึงค่าความเหมาะสมของคำตอบ (Fitness value) ซึ่งส่งผลทำให้โอกาสในการถูกเลือกของประชากรแต่ละตัวไม่เท่ากัน จากนั้นกลุ่มของประชากรที่ได้รับการคัดเลือกจะแลกเปลี่ยนชุดตัวแปรของคำตอบย่อยในตำแหน่งเดียวกันเพื่อทำการปรับปรุงค่าคำตอบ เรียกขั้นตอนการปรับปรุงนี้ว่า Crossover จากนั้นจะทำการเลือกกลุ่มของประชากรสำหรับรุ่นถัดไป (Offspring) โดยทำซ้ำตามขั้นตอนเดิมและในบางครั้งก็จะทำการปรับปรุงค่าของคำตอบด้วยวิธีการ Mutation โดยการเปลี่ยนค่าของตัวแปรอย่างสุ่มภายในประชากรตัวใด ๆ โดยที่ความน่าจะเป็นในการเลือกกลุ่มประชากรใด ๆ คำนวณจากสมการนี้

$$\text{Probability} = \text{Individual solution value} / \text{Population total}$$

ในทางทฤษฎีจะเห็นว่า Genetic algorithms เป็นวิธีการหาคำตอบที่ไม่ขึ้นกับ neighborhood แต่ทั้งนี้การปรับปรุงคำตอบด้วยวิธีการ Crossover ไม่สามารถประกันได้ว่าคำตอบที่ได้มานั้นจะเป็นคำตอบที่เป็นไปได้ (Feasible solution) และวิธีการจัดตำแหน่งของตัวแปรคำตอบสำหรับประชากรก็มีผลอย่างมากต่อกระบวนการ Crossover ขึ้นอยู่กับว่าจุดตัดจะอยู่ในตำแหน่งใด นอกจากนี้วิธีการกำหนดความน่าจะเป็นสำหรับการถูกเลือกเป็นประชากรก็เหมาะสมเฉพาะกับปัญหาแบบหาค่ามากที่สุดเท่านั้น

2.5 งานวิจัยที่เกี่ยวข้อง

Silver E.A., Pyke D.F. and Peterson R (1998 : 754) พบว่าความพยายามในการใช้ความรู้ทางคณิตศาสตร์เพื่อจัดการกับปัญหาสินค้าคงคลังเริ่มขึ้นในปี 1913 โดย Harris ซึ่งได้นำเสนอสมการที่ใช้สำหรับคำนวณหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัด ที่นิยมเรียกกันว่า EOQ และกลายเป็นตัวแบบสินค้าคงคลังขั้นพื้นฐานสำหรับเริ่มต้นการศึกษาการจัดการสินค้าคงคลังมาจนถึงปัจจุบัน แต่สมการ EOQ ตั้งอยู่บนสมมติฐานที่ว่าความต้องการสินค้ามีค่าคงที่ตลอดเวลาซึ่งไม่สอดคล้องกับความเป็นจริงในบางปัญหา ดังนั้นในปี 1958 Wagner and Whitin (1958 : 89-96) นำเสนอวิธีการแก้ปัญหาสินค้าคงคลังเมื่อความต้องการสินค้าเปลี่ยนแปลงไปตามช่วงเวลา (dynamic lot-size models) วิธีการนี้นิยมเรียกว่า Wagner and Whitin algorithm ซึ่งเป็นการ

แก้ปัญหาแบบ Dynamic Programming วิธีการนี้เริ่มจากคำนวณทุกทางเลือกที่สามารถเป็นไปได้ และเลือกทางเลือกที่ทำให้ต้นทุนต่ำสุด วิธีการนี้จะเสียเวลาในการหาคำตอบเมื่อนำไปแก้ปัญหาซึ่งมีความซับซ้อนมาก ๆ

เนื่องจากวิธีการในการหาคำตอบที่ดีที่สุดเสียเวลาในการคำนวณมากเมื่อปัญหาสินค้าคงคลังมีความซับซ้อนมาก ๆ นั้น ทำให้เกิดแนวคิดในการหาคำตอบที่เหมาะสมโดยประมาณขึ้น Silver and Meal (1969) จึงนำเสนอวิธีการคำนวณเพื่อให้ได้ค่าที่เหมาะสมโดยประมาณ ซึ่งคำตอบที่ได้ใกล้เคียงกับคำตอบที่ดีที่สุดและในบางครั้งได้คำตอบที่ดีที่สุด โดยใช้หลักการค่าเฉลี่ยของต้นทุนค่าสั่งซื้อหรือผลิตบวกด้วยค่าเก็บรักษาต่อช่วงเวลา วิธีการนี้ใช้เวลาในการหาคำตอบได้เร็วกว่าวิธีต่าง ๆ ในอดีต

สำหรับปัญหาของสินค้าคงคลังที่ซับซ้อนหรือใกล้เคียงกับสภาพความเป็นจริงในปัจจุบันมีอยู่หลายรูปแบบด้วยกัน Seerongk Prichanont (2002 : 48-69) ได้จำแนกปัญหาที่เกี่ยวข้องกับการสั่งซื้อหรือผลิตแบบประหยัดที่มีความซับซ้อนและหลากหลายในปัจจุบันออกตามลักษณะของปัญหาได้ดังนี้

1. แบ่งตามจำนวนของผลิตภัณฑ์ (Number of product) จะแบ่งออกได้เป็น 2 ระบบ คือ มีระบบที่มีผลิตภัณฑ์เดียว (single-item systems) และระบบที่มีหลายผลิตภัณฑ์ (Multi-item systems)
2. แบ่งตามโครงสร้างของการผลิต (product structure) จะแบ่งออกได้เป็น 2 ระบบ คือ มีระดับการผลิตเดียว และมีหลายระดับการผลิต
3. แบ่งตามข้อจำกัดเชิงสมรรถภาพ (capacity limitation) จะแบ่งออกได้เป็น 2 ระบบ คือ ไม่มีข้อจำกัดเชิงสมรรถภาพ และมีข้อจำกัดเชิงสมรรถภาพ
4. แบ่งตามวัตถุประสงค์ของ (objective function) สำหรับวัตถุประสงค์หลักของปัญหาสินค้าคงคลัง คือ ต้องการให้ต้นทุนทางด้านการผลิตต่ำที่สุด แต่บางปัญหาอาจมีวัตถุประสงค์อื่นต่างไป เช่น การทยอยส่งสินค้า เป็นต้น
5. แบ่งตามคุณสมบัติของตัวแปรตัดสินใจ (property of decision variable) เช่น บางปัญหาจะบังคับให้ตัวแปรตัดสินใจบางตัวเป็นจำนวนเต็ม เป็นต้น
6. แบ่งตามวิธีการหาคำตอบ (solution method) ซึ่งวิธีการในการคำตอบของปัญหาแบบเดียวกันนักวิจัยแต่ละคนอาจใช้วิธีการในการคำตอบที่แตกต่างกัน

ในงานวิจัยนี้สนใจปัญหาการหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้า N ชนิด และ T ช่วงเวลา ซึ่งอยู่ภายใต้ข้อจำกัดเดียวกัน ซึ่งมีนักวิจัยในอดีตให้ความสนใจและ

นำเสนองานวิจัยไว้มากมาย เช่น ในปี 1958 Manne (1958 : 15-135) เสนอวิธีการแก้ปัญหาโดยอาศัยหลักการของโปรแกรมเชิงเส้น

ในปี 1981 Dixon and Silver (1981 : 23-39) แบ่งปัญหาการหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้า N ชนิด และ T ช่วงเวลา ออกเป็น 3 กลุ่ม คือ (1) ปัญหาขนาดเล็ก (สินค้า 2 = ชนิด 10 ช่วงเวลา) (2) ปัญหาขนาดใหญ่ (สินค้า 10 และ 25 ชนิด 5 ช่วงเวลาถึง 15 ช่วงเวลา) และ (3) ปัญหาในความเป็นจริง จากนั้นใช้วิธีการรวมปริมาณการสั่งซื้อหรือผลิตเพื่อลดต้นทุนในการสั่งซื้อหรือผลิต โดยให้มีต้นทุนรวมต่ำที่สุด วิธีการนี้สามารถแก้ปัญหาขนาดใหญ่และปัญหาในความเป็นจริงได้ดี แต่กลับแก้ปัญหาขนาดเล็กไม่ค่อยนักเมื่อต้นทุนการสั่งซื้อหรือผลิตของสินค้าแต่ละชนิดมีความแตกต่างกันมาก ๆ และในปีเดียวกัน Dogramaci (1981 : 294-303) นำเสนอวิธีการ forward pass algorithm ซึ่งสามารถแก้ปัญหาความแตกต่างของต้นทุนการสั่งซื้อหรือผลิตของสินค้าแต่ละชนิดได้ดี

ในปี 1982 Karni and Roll (1982 : 249-256) ใช้วิธี Wagner and Whitin algorithm เพื่อหาขีดจำกัดล่าง (lower bound) ของต้นทุนที่ต่ำที่สุดโดยละเอียดเชิงสมรรถภาพ จากนั้นใช้เทคนิคการรวมปริมาณการสั่งซื้อหรือผลิตของสินค้าเพื่อให้อยู่ภายใต้ข้อจำกัดเชิงสมรรถภาพ วิธีการของ Karni and Roll สามารถหาคำตอบได้ภายในเวลา 180 วินาที จำนวน 28 การทดลอง จากการทดลอง 100 ครั้ง และได้คำตอบที่ดีที่สุด 17 การทดลอง และคำตอบที่ได้จากการทดลอง 28 การทดลองมีค่าความคลาดเคลื่อน 1 % โดยเฉลี่ยจากคำตอบที่ดีที่สุด

ในปี 1993 Millar and Yang (1993 : 409-420) นำเสนอวิธีการแก้ปัญหาโดยแปลงปัญหาการหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดไปเป็นปัญหาการขนส่ง (transportation problem) ของสินค้า N ชนิดแบบเป็นอิสระต่อกัน และแก้ปัญหาดังกล่าวด้วยวิธี Subgradient optimization พบว่าสามารถหา Primal Solution และ Dual Solution ที่ดีได้

จากงานวิจัยต่างประเทศที่กล่าวมาแล้วข้างต้นยังพบปัญหาในเรื่องของการใช้เวลาในการคำนวณค่อนข้างมาก เนื่องจากคำนวณทางคณิตศาสตร์หลายขั้นตอนในการคำนวณหาคำตอบ ดังนั้นในปี 2547 ภัทรธร แสงฤดี (2547 : 25-33) ได้นำเสนอวิธีการแก้ปัญหาโดยประยุกต์วิธีการของ Karni and Roll แต่ไม่ได้หาขีดจำกัดล่างของต้นทุนที่ต่ำที่สุด และใช้หลักการง่าย ๆ ในการรวมปริมาณการสั่งซื้อหรือผลิตเพื่อลดเวลาในการคำนวณ พบว่าวิธีการที่นำเสนอสามารถแก้ปัญหาขนาดเล็กได้ดี สำหรับงานวิจัยที่พัฒนาตัวแบบสินค้าคงคลังขึ้นมาและพัฒนาโปรแกรมตามตัวแบบสินค้าคงคลังที่พัฒนาขึ้นมาพบได้ในงานวิจัยของ ไพฑูรย์ อ้อยยิ่ง (2549 : 145-149) ซึ่งทำให้การตัดสินใจเกี่ยวกับปัญหาของสินค้าคงคลังทำได้สะดวกและรวดเร็วยิ่งขึ้น นอกจากนี้ยังทำให้ผู้ที่ไม่

มีความรู้ในด้านการจัดการสินค้าคงคลังเลยสามารถใช้โปรแกรมเพื่อการตัดสินใจเกี่ยวกับปัญหาของสินค้าคงคลังได้ถูกต้องและรวดเร็ว

2.6 สรุปปัญหา

จากการศึกษางานวิจัยของต่างประเทศพบว่าปัญหาหลักในการหาคำตอบของปัญหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังหลายชนิดทั้งแบบวิธีหาคำตอบที่ดีที่สุดและวิธีหาคำตอบที่เหมาะสมโดยประมาณ คือ ใช้เวลาในการคำนวณหาคำตอบสูงเมื่อปัญหาที่มีความซับซ้อน เพราะเนื่องจากใช้วิธีการคำนวณทางคณิตศาสตร์หลายขั้นตอนในการคำนวณหาคำตอบ ส่วนงานวิจัยภายในประเทศพยายามใช้หลักการง่าย ๆ เพื่อลดเวลาในการคำนวณหาคำตอบซึ่งได้ผลเป็นที่น่าพอใจ แต่ยังมีจุดที่น่าจะสามารถปรับปรุงให้สามารถคำนวณหาคำตอบได้ใกล้เคียงกับคำตอบที่ดีที่สุดได้มากขึ้น ดังนั้นในงานวิจัยนี้จึงสนใจศึกษาปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกัน ซึ่งเป็นปัญหาที่พิจารณาการสั่งซื้อหรือผลิตของสินค้าคงคลัง N ชนิด ภายใน T ช่วงเวลา และมีข้อจำกัดทางสมรรถภาพ คือ กำลังการผลิต โดยในงานวิจัยนี้จะหาวิธีการในการแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกัน ให้ได้คำตอบที่เหมาะสมโดยประมาณและใกล้เคียงกับคำตอบที่ดีที่สุดในขณะที่ใช้เวลาในการคำนวณหาคำตอบน้อยกว่า

บทที่ 3

การพัฒนาวิธีหาคำตอบที่เหมาะสมโดยประมาณ

การพัฒนาวิธีหาคำตอบที่เหมาะสมโดยประมาณของปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกัน โดยมีเป้าหมายของการพัฒนาวิธีหาคำตอบเพื่อให้ได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดและใช้เวลาในการคำนวณหาคำตอบที่น้อยกว่า

3.1 แบบจำลองทางคณิตศาสตร์ของปัญหา

สำหรับปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันของสินค้าคงคลัง N ชนิดภายในช่วงเวลา T โดยมีเป้าหมายเพื่อลดค่าใช้จ่ายต่าง ๆ ให้มีค่าต่ำที่สุดนั้นสามารถเขียนเป็นแบบจำลองทางคณิตศาสตร์ได้ดังนี้

$$\text{Minimize Total Cost} = \sum_{i=1}^N \sum_{t=1}^T z_{it} o_{it} + \sum_{i=1}^N \sum_{t=1}^T e_{it} h_i \quad (1)$$

Subject to

$$e_{it} = e_{i(t-1)} + x_{it} - d_{it} \quad \forall i, t \quad (2)$$

$$z_{it} = \begin{cases} 0 & \text{for } x_{it} = 0, \forall i, t \\ 1 & \text{for } x_{it} > 0, \forall i, t \end{cases} \quad (3)$$

$$x_{it} \leq P_t z_{it} \quad \forall i, t \quad (4)$$

$$\sum_i x_{it} \leq P_t \quad \forall i, t \quad (5)$$

$$x_{it}, e_{it} \geq 0 \text{ และ } \in I^+ \quad \forall i, t \quad (6)$$

เมื่อพารามิเตอร์คือ

i = ชนิดของสินค้าที่จะต้องทำการสั่งซื้อ $i=1,2,3\dots N$

t = ช่วงเวลาในการสั่งซื้อ $t=1,2,3\dots T$

P_t = ข้อจำกัดทางสมรรถภาพในช่วงเวลา t (มีค่าแน่นอน)

o_i = ต้นทุนการสั่งซื้อหรือผลิตต่อครั้งของสินค้าชนิด i

h_i = ต้นทุนต่อหน่วยต่อช่วงเวลาของการเก็บสินค้าชนิด i

d_{it} = ปริมาณความต้องการของสินค้าชนิด i ในช่วงเวลา t

และนอนพารามิเตอร์คือ

x_{it} = ปริมาณการสั่งซื้อหรือผลิตของสินค้าชนิด i ในช่วงเวลา t

e_{it} = ปริมาณสินค้าที่เหลือชนิด i ในช่วงเวลา t

z_{it} = การสั่งซื้อหรือผลิตสินค้าชนิด i ในช่วงเวลา t

จากแบบจำลองทางคณิตศาสตร์ของปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันข้างต้นจะเห็นได้ว่าสมการวัตถุประสงค์ของการแก้ปัญหาคือ ทำอย่างไรให้ผลบวกระหว่างผลรวมของค่าใช้จ่ายในการสั่งซื้อหรือผลิตและผลรวมของค่าใช้จ่ายในการเก็บรักษาของสินค้าคงคลัง N ชนิดภายในช่วงเวลา T มีค่าต่ำที่สุด ทั้งนี้การกระทำใดๆก็ตามเพื่อลดต้นทุนค่าใช้จ่ายที่เกิดขึ้นนั้นจะต้องอยู่ภายใต้ข้อจำกัดต่างๆดังนี้

- ข้อจำกัดที่ 1 : สมการที่ 2 ปริมาณสินค้าที่เหลือมีค่าเท่ากับผลรวมของปริมาณที่สั่งซื้อหรือผลิตในแต่ละช่วงเวลาลบกับผลรวมของความต้องการของสินค้าในแต่ละช่วงเวลา

- ข้อจำกัดที่ 2 : สมการที่ 3 จะไม่มีการสั่งซื้อหรือผลิตเมื่อ $Z=0$

- ข้อจำกัดที่ 3 : สมการที่ 4 ความต้องการของสินค้าแต่ละชนิดในแต่ละช่วงเวลาจะต้องมีค่าน้อยกว่าหรือเท่ากับข้อจำกัดเมื่อมีการสั่งซื้อหรือผลิต

- ข้อจำกัดที่ 4 : สมการที่ 5 ในแต่ละช่วงเวลาไม่สามารถสั่งซื้อหรือผลิตเกินจากข้อจำกัดที่ตั้งขึ้น

- ข้อจำกัดที่ 5 : สมการที่ 6 ปริมาณการสั่งซื้อหรือผลิตและปริมาณสินค้าที่เหลือต้องมากกว่าศูนย์

ในแบบจำลองทางคณิตศาสตร์ของปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันดังที่กล่าวมาแล้วมีค่าใช้จ่ายที่ถูกนำมาพิจารณาเพื่อลดต้นทุนรวมอยู่สองชนิดด้วยกันคือ ค่าใช้จ่ายในการสั่งซื้อหรือผลิตและค่าใช้จ่ายในการเก็บรักษาโดยปกติแล้วค่าใช้จ่ายในการสั่งซื้อหรือผลิตจะมีค่ามากกว่าค่าเก็บรักษาหลายเท่าตัว ดังนั้นแนวทางที่เป็นไปได้สำหรับการลดต้นทุนรวมคือการลดค่าใช้จ่ายการสั่งซื้อหรือผลิตสินค้าโดยการลดจำนวนครั้งในการสั่งซื้อหรือผลิตสินค้าลงให้ได้มากที่สุด

3.2 แนวทางการแก้ไขปัญหา

การแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันโดยวิธีการหาคำตอบที่ดีที่สุดนั้นใช้เวลาในการคำนวณสูงจึงได้เกิดแนวคิดในการหาคำตอบที่เหมาะสมโดยประมาณขึ้นหลากหลายวิธี และวิธีการแก้ปัญหาแบบที่หาคำตอบที่

เหมาะสมโดยประมาณที่มีประสิทธิภาพในเรื่องการลดเวลาในการคำนวณหาคำตอบที่น่าสนใจคือวิธีการรวมปริมาณการสั่งซื้อหรือผลิตเพื่อลดต้นทุนในการสั่งซื้อหรือผลิต ซึ่ง Dixon and Silver (1981 : 23-39) และภัทรธร แสงฤดี(2547 : 92) ได้พิสูจน์แล้วว่าสามารถลดเวลาในการคำนวณหาคำตอบได้ดี

วิธีการรวมปริมาณการสั่งซื้อหรือผลิตสามารถลดต้นทุนรวมได้ก็จริงแต่ก็เป็นต้นเหตุให้เกิดค่าใช้จ่ายในการเก็บรักษาเช่นกัน ดังนั้นการรวมปริมาณการสั่งซื้อหรือผลิตจะทำให้ต่อเมื่อ ค่าใช้จ่ายในการสั่งซื้อหรือผลิตที่ลดลงอันเนื่องมาจากการรวมปริมาณการสั่งซื้อหรือผลิตจะต้องมีค่ามากกว่าค่าใช้จ่ายในการเก็บรักษาที่เกิดขึ้นจากการรวมปริมาณการสั่งซื้อหรือผลิตนั่นเองตัวอย่างเช่น สินค้าชนิดหนึ่งมีค่าการสั่งซื้อหรือผลิตเท่ากับ 200 บาท มีค่าการเก็บรักษาเท่ากับ 5 บาท มีปริมาณการสั่งซื้อหรือผลิตในวันที่ 1 และวันที่ 2 เท่ากันเท่ากับ 50 หน่วย กำลังการผลิตสูงสุดต่อวันเท่ากับ 100 หน่วย หากต้องการลดต้นทุนรวมด้วยวิธีการรวมปริมาณการสั่งซื้อหรือผลิตในวันที่ 1 และวันที่ 2 เข้าด้วยกัน ซึ่งจะสามารถลดค่าการสั่งซื้อหรือผลิตได้จำนวน 200 บาทแต่ในกรณีนี้จะไม่มีการย้ายปริมาณการสั่งซื้อหรือผลิตของวันที่ 2 ไปรวมกับปริมาณการสั่งซื้อหรือผลิตของวันที่ 1 ถึงแม้จะสามารถย้ายปริมาณการสั่งซื้อหรือผลิตในวันที่ 2 ไปรวมกับปริมาณการสั่งซื้อหรือผลิตในวันที่ 1 ได้เพราะกำลังการผลิตรวมในวันที่ 1 ไม่เกินกำลังการผลิตสูงสุดก็ตามเหตุก็เพราะค่าการเก็บรักษาที่เกิดขึ้นจำนวน 250 บาทมีค่ามากกว่าค่าสั่งซื้อหรือผลิตที่ลดได้จำนวน 200 บาทนั่นเอง การย้ายจะเกิดขึ้นก็ต่อเมื่อค่าการเก็บรักษาที่เกิดขึ้นน้อยกว่า 200 บาทเท่านั้น

วิธีการรวมปริมาณการสั่งซื้อหรือผลิตมีผู้ศึกษาวิจัยไว้หลายวิธีด้วยกัน เช่น มีการหาค่า Lower bound ก่อนแล้วจึงปรับปรุงคำตอบให้อยู่ภายใต้ข้อจำกัด บางวิธีใช้เงื่อนไขที่ไม่ซับซ้อนในการรวมปริมาณการสั่งซื้อหรือผลิต สำหรับวิธีการรวมปริมาณการสั่งซื้อหรือผลิตที่น่าสนใจนั้นจะใช้เงื่อนไขในการรวมปริมาณการสั่งซื้อหรือผลิตที่ไม่ซับซ้อนเช่นกัน แต่มีความแตกต่างจากวิธีอื่นๆที่ผ่านมาก็คือจะมีการพิจารณาแบ่งย้ายปริมาณการสั่งซื้อหรือผลิตในกรณีที่ไม่สามารถรวมปริมาณการสั่งซื้อหรือผลิตได้ทั้งหมดในคราวเดียว ซึ่งจะเพิ่มโอกาสในการลดต้นทุนรวมทำให้ได้คำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดมากยิ่งขึ้น

3.3 หลักการหาคำตอบที่เหมาะสมโดยประมาณที่น่าสนใจ

วิธีการหาคำตอบที่เหมาะสมโดยประมาณที่น่าสนใจมีหลักการทำงานที่สำคัญคือ จะพิจารณาย้ายความต้องการสินค้าไปรวมกับความต้องการสินค้าของช่วงเวลาก่อนหน้าเพื่อให้ได้ต้นทุนรวมที่ลดลง และในกรณีที่ไม่สามารถย้ายความต้องการสินค้าทั้งหมดไปยังช่วงเวลาก่อน

หน้าได้ให้พิจารณาแบ่งย้ายความต้องการสินค้าไปยังช่วงเวลาก่อนหน้าที่ติดกันจนเต็มข้อจำกัดเชิงสมรรถภาพแล้วจึงย้ายความต้องการสินค้าที่เหลือไปยังช่วงเวลาก่อนหน้าอื่นๆต่อไป แต่ทั้งนี้ต้องทำให้ต้นทุนรวมลดลงเท่านั้นตัวอย่างเช่น สินค้าชนิดหนึ่งมีค่าการสั่งซื้อหรือผลิตเท่ากับ 500 บาท มีค่าการเก็บรักษาเท่ากับ 5 บาท มีปริมาณการสั่งซื้อหรือผลิตในวันที่ 1 เท่ากับ 95 หน่วย วันที่ 2 เท่ากับ 80 หน่วย วันที่ 3 เท่ากับ 25 หน่วย กำลังการผลิตสูงสุดต่อวันเท่ากับ 100 หน่วย จะเห็นได้ว่าไม่สามารถลดต้นทุนรวมด้วยวิธีการย้ายปริมาณการสั่งซื้อหรือผลิตทั้งหมดไปยังช่วงเวลาใดๆได้ เนื่องจากเกินกำลังการผลิตสูงสุด แต่จะสามารถแบ่งย้ายปริมาณการสั่งซื้อหรือผลิตในวันที่ 3 จำนวน 20 ไปรวมกับปริมาณการสั่งซื้อหรือผลิตในวันที่ 2 มีค่าเก็บรักษาเกิดขึ้นจำนวน 100 บาท และสามารถแบ่งย้ายปริมาณการสั่งซื้อหรือผลิตในวันที่ 3 จำนวน 5 ไปรวมกับปริมาณการสั่งซื้อหรือผลิตในวันที่ 1 มีค่าเก็บรักษาเกิดขึ้นจำนวน 50 บาท ดังนั้นค่าเก็บรักษาที่เกิดขึ้นรวมเท่ากับ 150 บาท ซึ่งน้อยกว่าค่าการสั่งซื้อหรือผลิตที่ลดลงได้จำนวน 500 บาท ซึ่งการแบ่งย้ายสินค้าข้างต้นทำให้สามารถลดต้นทุนรวมลงได้จำนวน 350 บาท

3.4 ขั้นตอนการหาคำตอบที่เหมาะสมโดยประมาณที่นำเสนอ

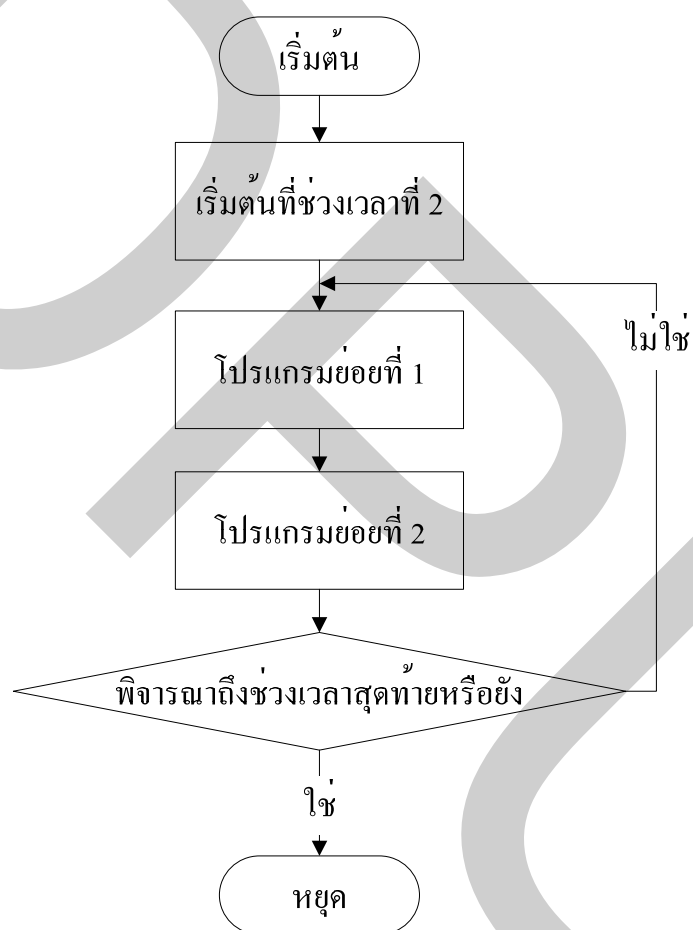
3.4.1 การหาคำตอบที่เหมาะสมโดยประมาณจะพิจารณาที่ละช่วงเวลาโดยเริ่มจากช่วงเวลา 2 ก่อน ให้พิจารณาความต้องการสินค้าของสินค้าแต่ละชนิดตลอดช่วงเวลาที่กำลังพิจารณา โดยจะย้ายความต้องการสินค้าที่มากกว่าศูนย์และไม่ทำให้ผลรวมของข้อจำกัดเชิงสมรรถภาพของช่วงเวลาก่อนหน้าที่กำลังพิจารณาอยู่เกินข้อจำกัด ไปรวมกับความต้องการสินค้าชนิดเดียวกันของช่วงเวลาก่อนหน้าที่กำลังพิจารณาอยู่ ทั้งนี้การย้ายปริมาณความต้องการสินค้าดังกล่าวจะกระทำได้ก็ต่อเมื่อค่าใช้จ่ายในการสั่งซื้อหรือผลิตที่ลดลงเนื่องจากการรวมปริมาณการสั่งซื้อหรือผลิตมีค่ามากกว่าค่าใช้จ่ายในการเก็บรักษาที่เกิดขึ้น กรณีที่สามารถย้ายความต้องการสินค้าได้มากกว่าหนึ่งชนิดในช่วงเวลาที่กำลังพิจารณาอยู่ให้ย้ายความต้องการสินค้าที่สามารถลดต้นทุนรวมได้มากที่สุดก่อนตามลำดับ ทำการพิจารณาเช่นนี้ไปเรื่อยๆจนกระทั่งถึงช่วงเวลา 1

3.4.2 หากพิจารณาตามขั้นตอนในข้อ 3.4.1 แล้วไม่สามารถย้ายความต้องการสินค้าใดๆในช่วงเวลาที่กำลังพิจารณาอยู่นั้นไปยังช่วงเวลาก่อนหน้าได้แล้ว ให้พิจารณาแบ่งย้ายความต้องการสินค้าของช่วงเวลาที่กำลังพิจารณาอยู่นั้นไปรวมกับความต้องการสินค้าของช่วงเวลาก่อนหน้า โดยแบ่งย้ายความต้องการสินค้าในช่วงเวลาที่กำลังพิจารณาไปยังช่วงเวลาก่อนหน้าที่อยู่ติดกันให้เต็มข้อจำกัดก่อน แล้วพิจารณาย้ายความต้องการสินค้าที่เหลือไปรวมกับความต้องการสินค้าของช่วงเวลาก่อนหน้าถัดไปโดยพิจารณาถัดไปเรื่อยๆจนกระทั่งถึงเวลาที่ 1 ทั้งนี้การย้ายปริมาณความต้องการสินค้านี้ดังกล่าวจะกระทำได้ก็ต่อเมื่อค่าใช้จ่ายในการสั่งซื้อหรือผลิตที่ลดลงเนื่องจาก

การรวมปริมาณการสั่งซื้อหรือผลิตที่มีการแบ่งย้ายแล้วมีค่ามากกว่าค่าใช้จ่ายในการเก็บรักษาที่เกิดขึ้น

3.4.3 ให้กระทำซ้ำขั้นตอนที่ 3.4.1 และ 3.4.2 จนกระทั่งครบทุกช่วงเวลา

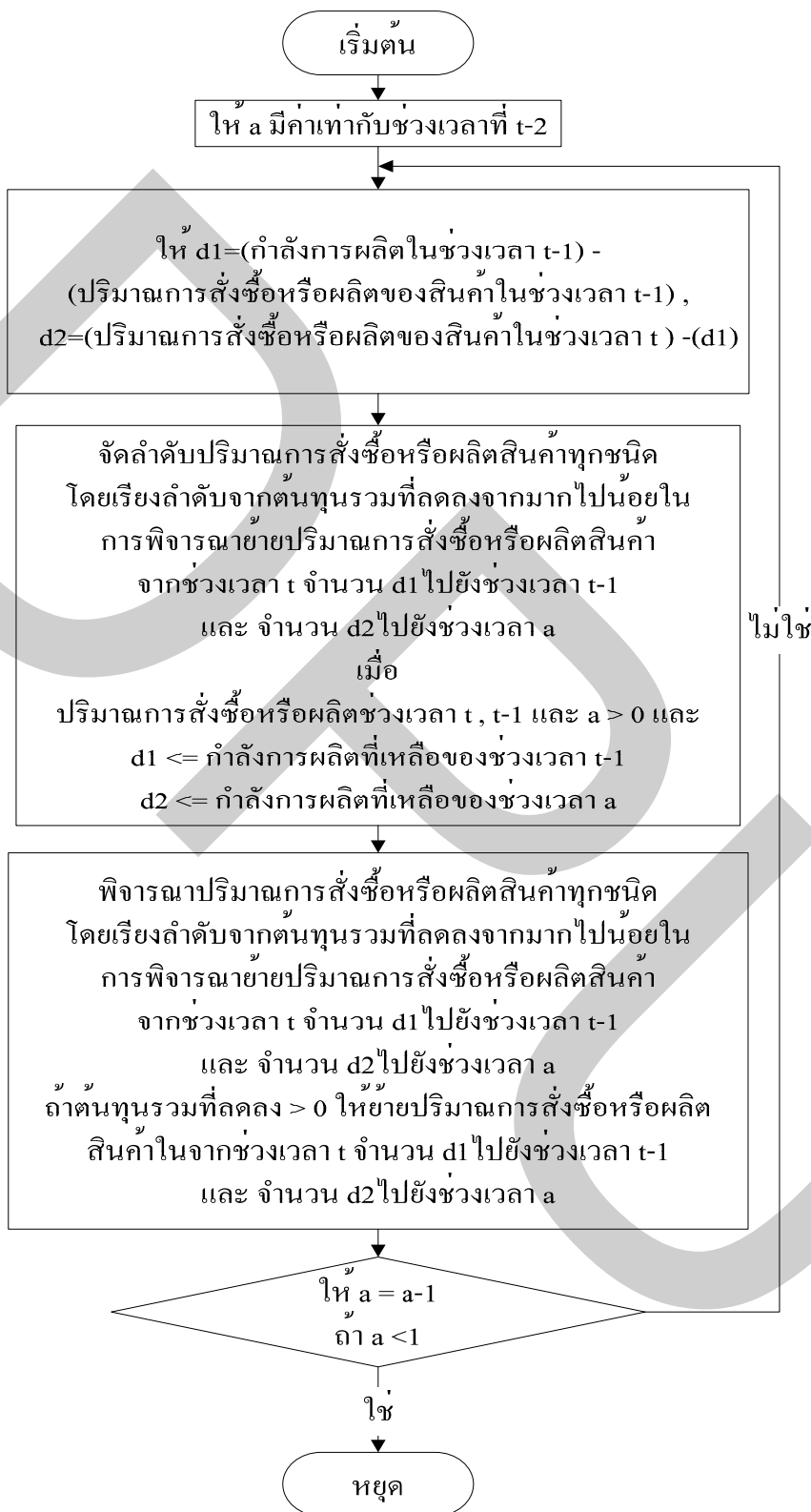
ขั้นตอนการคำนวณหรือหาคำตอบที่อธิบายมาทั้งหมดสามารถเขียนเป็น แผนผังการย้ายปริมาณการสั่งซื้อหรือผลิตสินค้าได้ดังภาพที่ 3.1-ภาพที่ 3.3



ภาพที่ 3.1 ขั้นตอนการคำนวณหลัก



ภาพที่ 3.2 ขั้นตอนการคำนวณของโปรแกรมย่อยที่ 1



ภาพที่ 3.3 ขั้นตอนการคำนวณของโปรแกรมย่อยที่ 2

3.5 ตัวอย่างการคำนวณ

เพื่อความเข้าใจในวิธีการแก้ปัญหาตามที่ได้นำเสนอดีขึ้นไปนี้จะยกตัวอย่างการคำนวณหาคำตอบโดยใช้ข้อมูลตามภาพที่ 3.4

สินค้าช่วงเวลา	1	2	3	4	5	6	ค่าสั่งซื้อหรือผลิต	ค่าเก็บรักษาต่อหน่วย
สินค้าชนิดที่ 1	27	20	40	27	12	30	100	2
สินค้าชนิดที่ 2	45	60	25	20	4	10	200	5
สินค้าชนิดที่ 3	21	0	2	21	35	41	300	10
ความต้องการรวม	93	80	67	68	51	81	ต้นทุนรวม	3300
กำลังผลิต	100	100	100	100	100	100		

ภาพที่ 3.4 ข้อมูลปัญหาตัวอย่าง

ขั้นที่ 1 พิจารณาช่วงเวลา 2 ไม่สามารถย้ายความต้องการสินค้าใดๆไปยังช่วงเวลาก่อนหน้าได้ เนื่องจากจะทำให้เกินกำลังการผลิต

ขั้นที่ 2 พิจารณาช่วงเวลา 3 สามารถย้ายความต้องการสินค้าชนิดที่ 3 ไปยังช่วงเวลา 1 ได้ สามารถย้ายความต้องการสินค้าชนิดที่ 2 ไปยังช่วงเวลา 2 ได้จำนวน 20 หน่วย และย้ายไปยังช่วงเวลา 1 ได้ 5 หน่วย

สินค้าช่วงเวลา	1	2	3	4	5	6	ค่าสั่งซื้อหรือผลิต	ค่าเก็บรักษาต่อหน่วย
สินค้าชนิดที่ 1	27	20	40	27	12	30	100	2
สินค้าชนิดที่ 2	50	80	0	20	4	10	200	5
สินค้าชนิดที่ 3	23	0	0	21	35	41	300	10
ความต้องการรวม	100	100	40	68	51	81	ต้นทุนรวม	2990
กำลังผลิต	100	100	100	100	100	100		

ภาพที่ 3.5 ผลลัพธ์ของการคำนวณขั้นที่ 2

ขั้นที่ 3 พิจารณาช่วงเวลาที 4 สามารถย้ายความต้องการสินค้าชนิดที่ 1 ไปยังช่วงเวลาที 3 ได้

สินค้าช่วงเวลา	1	2	3	4	5	6	คำสั่งซื้อ หรือผลิต	ค่าเก็บรักษา ต่อหน่วย
สินค้าชนิดที่ 1	27	20	67	0	12	30	100	2
สินค้าชนิดที่ 2	50	80	0	20	4	10	200	5
สินค้าชนิดที่ 3	23	0	0	21	35	41	300	10
ความต้องการรวม	100	100	67	41	51	81	ต้นทุนรวม	2944
กำลังผลิต	100	100	100	100	100	100		

ภาพที่ 3.6 ผลลัพธ์ของการคำนวณขั้นที่ 3

ขั้นที่ 4 พิจารณาช่วงเวลาที 5 สามารถย้ายความต้องการสินค้าชนิดที่ 2 ไปยังช่วงเวลาที 4 ได้ สามารถย้ายความต้องการสินค้าชนิดที่ 1 ไปยังช่วงเวลาที 3 ได้

สินค้าช่วงเวลา	1	2	3	4	5	6	คำสั่งซื้อ หรือผลิต	ค่าเก็บรักษา ต่อหน่วย
สินค้าชนิดที่ 1	27	20	79	0	0	30	100	2
สินค้าชนิดที่ 2	50	80	0	24	0	10	200	5
สินค้าชนิดที่ 3	23	0	0	21	35	41	300	10
ความต้องการรวม	100	100	79	45	35	81	ต้นทุนรวม	2712
กำลังผลิต	100	100	100	100	100	100		

ภาพที่ 3.7 ผลลัพธ์ของการคำนวณขั้นที่ 4

ขั้นที่ 5 พิจารณาช่วงเวลาที 6 สามารถย้ายความต้องการสินค้าชนิดที่ 2 ไปยังช่วงเวลาที 4 ได้

สินค้าช่วงเวลา	1	2	3	4	5	6	คำสั่งซื้อ หรือผลิต	ค่าเก็บรักษา ต่อหน่วย
สินค้าชนิดที่ 1	27	20	79	0	0	30	100	2
สินค้าชนิดที่ 2	50	80	0	34	0	0	200	5
สินค้าชนิดที่ 3	23	0	0	21	35	41	300	10
ความต้องการรวม	100	100	79	55	35	71	ต้นทุนรวม	2612
กำลังผลิต	100	100	100	100	100	100		

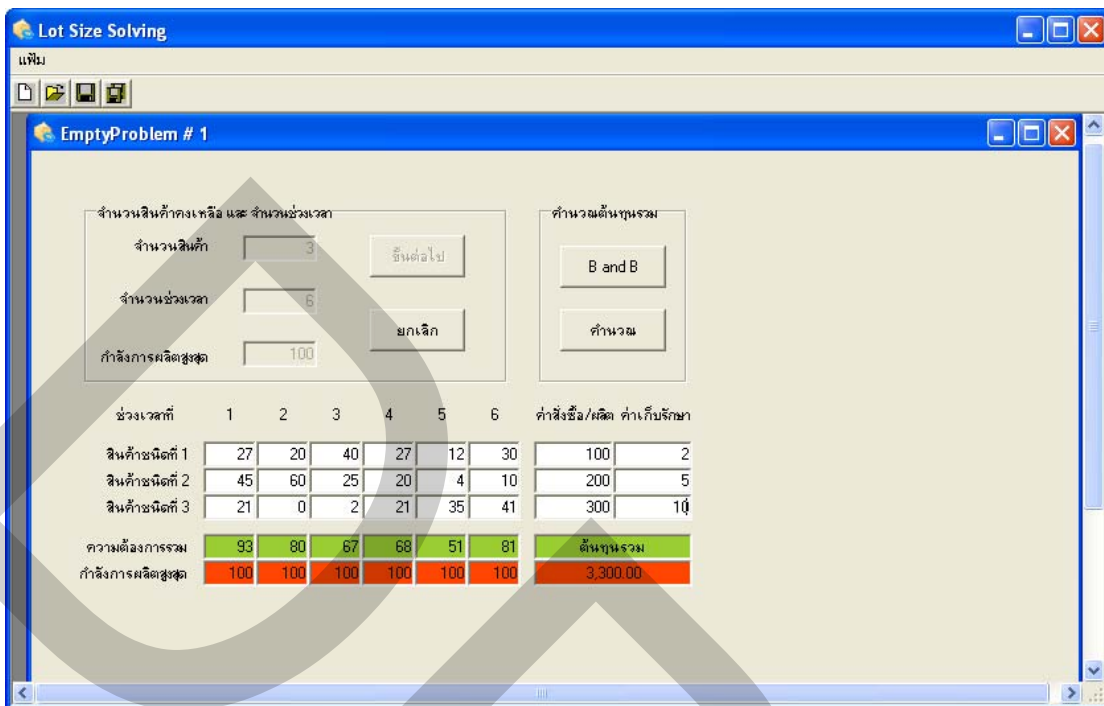
ภาพที่ 3.8 ผลลัพธ์ของการคำนวณขั้นที่ 5

จากตัวอย่างการคำนวณจะเห็นได้ว่าวิธีการที่นำเสนอสามารถลดต้นทุนรวมของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกันได้ และอยู่ภายใต้ข้อจำกัดที่ตั้งไว้

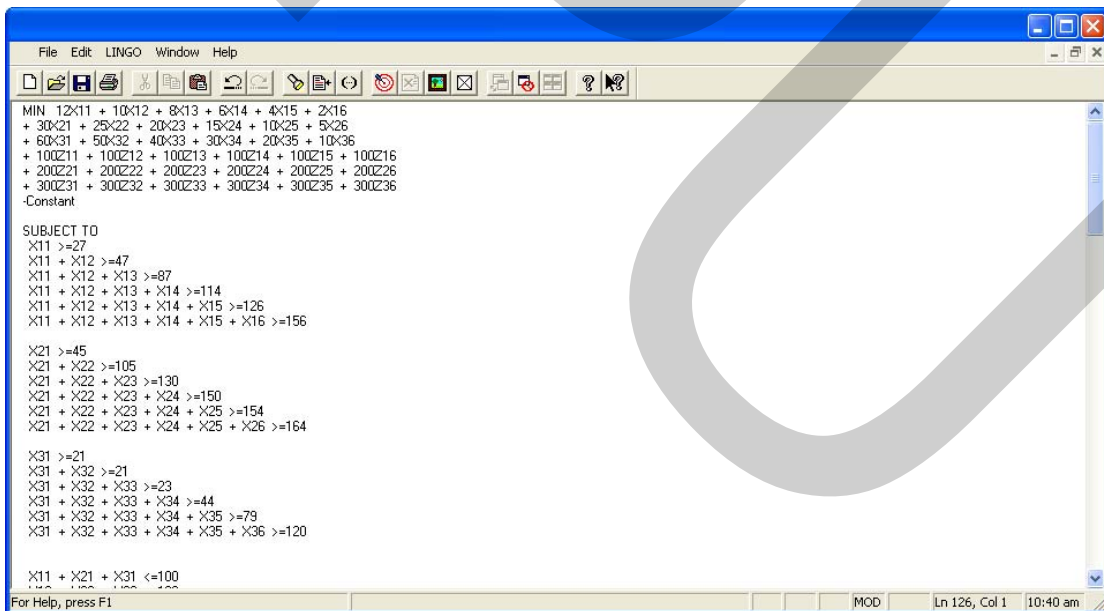
3.6 เครื่องมือที่ใช้ในงานวิจัย

สำหรับเครื่องมือที่ใช้ทดสอบวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกัน ที่ได้เสนอนั้น เป็นโปรแกรมที่พัฒนาขึ้นด้วยภาษา Microsoft Visual Basic และ Microsoft Visual C++ สาเหตุที่ต้องใช้ถึงสองภาษาก็เนื่องมาจาก Microsoft Visual Basic สามารถเขียนเป็นโปรแกรมสำหรับติดต่อกับผู้ใช้งาน (User Interface) ได้สะดวก จึงใช้ภาษา Microsoft Visual Basic สร้างโปรแกรมในส่วนที่ต้องใช้สำหรับติดต่อกับผู้ใช้งาน ส่วนภาษา Microsoft Visual C++ ถูกนำมาใช้พัฒนาโปรแกรมส่วนที่เป็นขั้นตอนการแก้ปัญหา (Algorithm) เนื่องจากภาษา Microsoft Visual C++ เป็นที่ยอมรับกันโดยทั่วไปว่าสามารถทำงานได้รวดเร็วเนื่องจากงานวิจัยครั้งนี้จำเป็นต้องใช้เวลาในการคำนวณให้น้อยที่สุด

ส่วนเครื่องมือที่ใช้สำหรับเปรียบเทียบผลการวิจัยวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกันที่ได้เสนอนั้น จะใช้โปรแกรมสำเร็จรูป ซึ่งใช้วิธีการคำนวณหาคำตอบด้วยวิธี Branch and Bound โปรแกรมนี้เป็นโปรแกรมที่ใช้สำหรับหาคำตอบที่ดีที่สุดได้ และเป็นที่ยอมรับกันโดยทั่วไป



ภาพที่ 3.9 โปรแกรมที่พัฒนาขึ้นเพื่อใช้เป็นเครื่องมือในการวิจัย



ภาพที่ 3.10 โปรแกรมสำเร็จรูปที่ใช้สำหรับเปรียบเทียบผลการทดลอง

คอมพิวเตอร์ที่ใช้สำหรับทดสอบวิธีการที่นำเสนอในงานวิจัยและวิธีการคำนวณแบบ Branch and Bound นั้นเป็นคอมพิวเตอร์ Notebook รุ่น Intel[®] Pentium[®] M 725 processor 1.6 GHz 400 MHz FSB 2MB L2 cache RAM 512 MB ติดตั้ง Windows XP

3.7 ตัวอย่างปัญหาสำหรับทดสอบวิธีการแก้ปัญหาที่นำเสนอ

สำหรับปัญหาที่ใช้สำหรับทดสอบวิธีการแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันที่นำเสนอจะแบ่งออกเป็น 3 กลุ่มด้วยกัน คือ 1) กลุ่มปัญหาขนาดเล็ก 2) กลุ่มปัญหาขนาดกลาง 3) กลุ่มปัญหาขนาดใหญ่ โดยข้อมูลต่างๆ ของปัญหาเช่น ความต้องการของสินค้าแต่ละชนิดในช่วงเวลาต่างๆ ค่าสั่งซื้อหรือผลิตและค่าเก็บรักษา ของปัญหาในแต่ละกลุ่มจะได้อมาจากการสุ่มข้อมูล

3.8 การวิเคราะห์ผลการทดลอง

การวิเคราะห์ประสิทธิภาพผลการทดลองหรือคำตอบของวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกันที่นำเสนอ จะพิจารณาเปรียบเทียบกับวิธีการหาคำตอบที่ดีที่สุดด้วยวิธี Branch and Bound ใน 2 ด้านคือ 1) ด้านเวลาที่ใช้ในการคำนวณหาคำตอบ 2) ด้านคุณภาพของคำตอบ

3.8.1 ด้านเวลาที่ใช้ในการคำนวณหาคำตอบ สำหรับการวิเคราะห์ในด้านเวลาที่ใช้ในการหาคำตอบนั้นจะใช้เวลาที่ใช้ในการหาคำตอบจากโปรแกรมที่คำนวณตามวิธีการที่นำเสนอมาเปรียบเทียบกับเวลาที่ใช้ในการหาคำตอบด้วยการคำนวณด้วยวิธี Branch and Bound จากโปรแกรมสำเร็จ ซึ่งเวลาที่ได้จากโปรแกรมทั้งสองนั้นใช้วิธีการจับเวลาด้วยคำสั่งคอมพิวเตอร์ ภายในแต่ละโปรแกรมเองทำให้ได้เวลาที่มีความเที่ยงตรงและถูกต้องสูง

3.8.2 ด้านคุณภาพของคำตอบ สำหรับการวิเคราะห์คุณภาพของคำตอบจะใช้คำตอบที่คำนวณด้วยวิธีการที่นำเสนอเปรียบเทียบกับคำตอบที่คำนวณด้วยวิธี Branch and Bound จากโปรแกรมสำเร็จรูป โดยจะคำนวณเปอร์เซ็นต์ความแตกต่าง(% Difference)ของคำตอบที่คำนวณด้วยวิธีการที่นำเสนอเทียบกับวิธีการคำนวณแบบ Branch and Bound จากสูตร

$$\% \text{ Difference} = \left(\frac{\text{Heuristic Solution} - \text{Optimal Solution}}{\text{Optimal Solution}} \right) \times 100\%$$

3.9 ผลการทดลองเบื้องต้น

การทดสอบวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพรวมกันด้วยวิธีการที่นำเสนอในเบื้องต้น จะใช้ตัวอย่างปัญหาขนาดสินค้า 3 ชนิด 6 ช่วงเวลาและปัญหาขนาดสินค้า 5 ชนิด 10 ช่วงเวลา

ตารางที่ 3.1 ผลการทดลองปัญหาขนาดสินค้า 3 ชนิด 6 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	11,115.00	11,115.00	0.010	0.100	0.00
2	3,010.00	2,852.00	0.001	0.490	5.54
3	3,092.00	3,020.00	0.005	1.000	2.38
4	3,269.00	3,269.00	0.005	0.470	0.00
5	8,261.00	8,205.00	0.001	0.290	0.68
เฉลี่ย			0.004	0.470	1.72

ตารางที่ 3.2 ผลการทดลองปัญหาขนาดสินค้า 5 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	12,321.00	12,107.00	0.001	6.000	1.77
2	12,683.00	12,478.00	0.005	12.000	1.64
3	20,122.00	20,092.00	0.005	6.000	0.15
4	10,575.00	10,117.00	0.005	8.000	4.53
5	14,139.00	14,121.00	0.005	218.000	0.13
เฉลี่ย			0.004	50.000	1.64

3.10 สรุปผลการทดลองเบื้องต้น

สำหรับจำนวนสินค้าและช่วงเวลาที่ใช้ในการทดลองเป็นจำนวนที่สมมุติขึ้นมาและจากผลการทดลองเบื้องต้นจะเห็นได้ว่าวิธีการที่นำเสนอสามารถแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิด โดยมีข้อจำกัดทางสมรรถภาพร่วมกันได้โดยอยู่ภายใต้ข้อกำหนดที่กำหนดขึ้น สำหรับเวลาที่ใช้ในการหาคำตอบนั้นวิธีการที่นำเสนอใช้เวลาในการคำนวณหาคำตอบน้อยกว่าในทุกปัญหาและผลลัพธ์ที่ได้มีเปอร์เซ็นต์ความแตกต่างเฉลี่ยเมื่อเทียบกับคำตอบที่ดีที่สุดน้อยกว่า 1.68%

บทที่ 4

ผลการศึกษา

จากผลการทดลองในเบื้องต้นจะเห็นได้ว่าวิธีการที่นำเสนอสามารถแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันได้และอยู่ภายใต้ข้อกำหนดที่ต้งขึ้นและสามารถแก้ปัญหาเรื่องเวลาในการคำนวณหาคำตอบของวิธีการหาคำตอบที่ดีที่สุดซึ่งใช้เวลาในการคำนวณสูงได้ สำหรับในบทนี้จะทำการทดลองและวิเคราะห์ผลการทดลองกับตัวอย่างปัญหาจำนวน 3 กลุ่มด้วยกันคือ

- 1 กลุ่มปัญหาขนาดเล็ก
- 2 กลุ่มปัญหขนาดกลาง
- 3 กลุ่มปัญหขนาดใหญ่

ผลการทดลองของปัญหาทั้ง 3 กลุ่มจะถูกนำมาวิเคราะห์หาปัจจัยต่างๆที่มีผลกับประสิทธิภาพของคำตอบทั้งในด้านของเวลาและคุณภาพของคำตอบ

4.1 รูปแบบของกลุ่มปัญหาตัวอย่าง

กลุ่มของปัญหาตัวอย่างที่ใช้สำหรับทดสอบวิธีการที่นำเสนอในงานวิจัยครั้งนี้มีทั้งหมด 3 กลุ่มด้วยกันคือ

- 4.1.1 กลุ่มปัญหาขนาดเล็ก
- 4.1.2 กลุ่มปัญหขนาดกลาง
- 4.1.3 กลุ่มปัญหขนาดใหญ่

สำหรับจำนวนสินค้าและจำนวนช่วงเวลาเป็นตัวเลขที่สมมุติขึ้นมาเพื่อทดสอบวิธีการที่นำเสนอโดยที่ข้อมูลต่างๆของปัญหาตัวอย่างเช่น ปริมาณการสั่งซื้อหรือผลิตสินค้า,กำลังการผลิต, ค่าสั่งซื้อหรือผลิตและค่าเก็บรักษาจะไดจากการสุ่มและมีรูปแบบการสุ่มของปัญหาในแต่ละกลุ่มตามความเหมาะสมของปัญหาในกลุ่มนั้น ๆ

4.2 กลุ่มปัญหาตัวอย่างขนาดเล็ก

สำหรับตัวอย่างปัญหขนาดเล็กจะประกอบด้วยตัวอย่างปัญหาจำนวน 4 กลุ่มย่อย โดยแต่ละกลุ่มย่อยมีจำนวน 10 ปัญหาดังนี้

4.4.10 สินค้า 25 ชนิด 15 ช่วงเวลา รูปแบบของข้อมูลที่ใช้ในการทดสอบของปัญหานี้คือ กำลังการผลิตสูงสุดเท่ากับ 2000 อัตราส่วนระหว่างคำสั่งซื้อหรือผลิตและค่าเก็บรักษาเท่ากับ 100 ปริมาณการสั่งซื้อหรือผลิตของสินค้าเกิดจากการสุ่มข้อมูล

4.5 ผลการการศึกษา

จากการทดสอบวิธีการแก้ปัญหาคำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันตามวิธีการที่นำเสนอกับกลุ่มปัญหขนาดเล็กรวมถึงกลุ่มปัญหขนาดกลางและกลุ่มปัญหขนาดใหญ่นั้น สามารถแสดงผลของการทดสอบของปัญหาจำแนกตามจำนวนสินค้าและจำนวนช่วงเวลาได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบจำแนกตามจำนวนสินค้าและช่วงเวลา

จำนวน สินค้า	ช่วงเวลา		
	5	10	15
	เปอร์เซ็นต์ความ แตกต่างของ คำตอบ	เปอร์เซ็นต์ความ แตกต่างของ คำตอบ	เปอร์เซ็นต์ความ แตกต่างของ คำตอบ
3	1.88	1.32	2.95
5	1.61	5.54	*
10	2.44	*	*
15	2.19	*	*
20	1.59	*	*
25	1.80	*	*

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณสูง

จากผลการทดสอบพบว่าวิธีการที่นำเสนอสามารถแก้ปัญหาดังกล่าวได้ทุกปัญหา และใช้เวลาในการคำนวณน้อยกว่าวิธีการคำนวณแบบ Branch and Bound มากในทุกๆปัญหา สำหรับในกลุ่มปัญหขนาดใหญ่วิธีการคำนวณแบบ Branch and Bound นั้นไม่สามารถคำนวณหาคำตอบได้เป็นผลสำเร็จ เนื่องจากใช้เวลาในการคำนวณหาคำตอบเฉลี่ยมากกว่า 48 ชั่วโมงขึ้นไปยังไม่

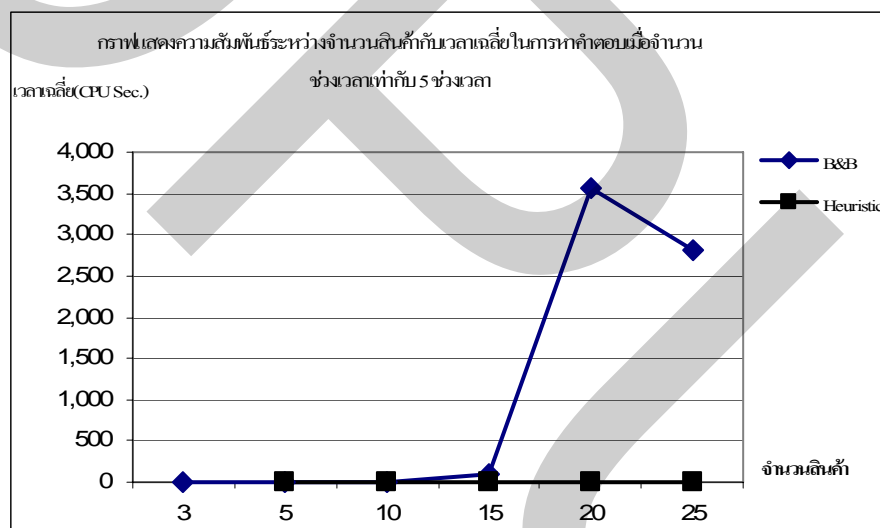
สามารถหาคำตอบที่ดีที่สุดได้ ทำให้เครื่องคอมพิวเตอร์ที่ใช้ในการวิจัยครั้งนี้ไม่สามารถคำนวณหาคำตอบที่ดีที่สุดของปัญหาขนาดใหญ่บางปัญหาได้เป็นผลสำเร็จ

4.6 การวิเคราะห์ผลการศึกษา

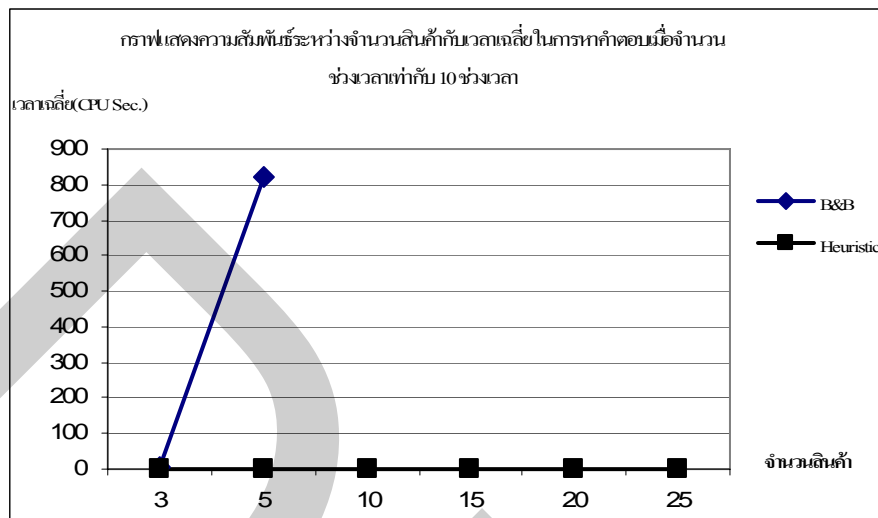
การวิเคราะห์ผลการทดลองในการวิจัยครั้งนี้จะพิจารณาใน 2 แนวทางด้วยกันคือ ด้านเวลาที่ใช้ในการคำนวณหาคำตอบและคุณภาพของคำตอบที่ได้

4.6.1 การวิเคราะห์เวลาที่ใช้ในการคำนวณหาคำตอบ

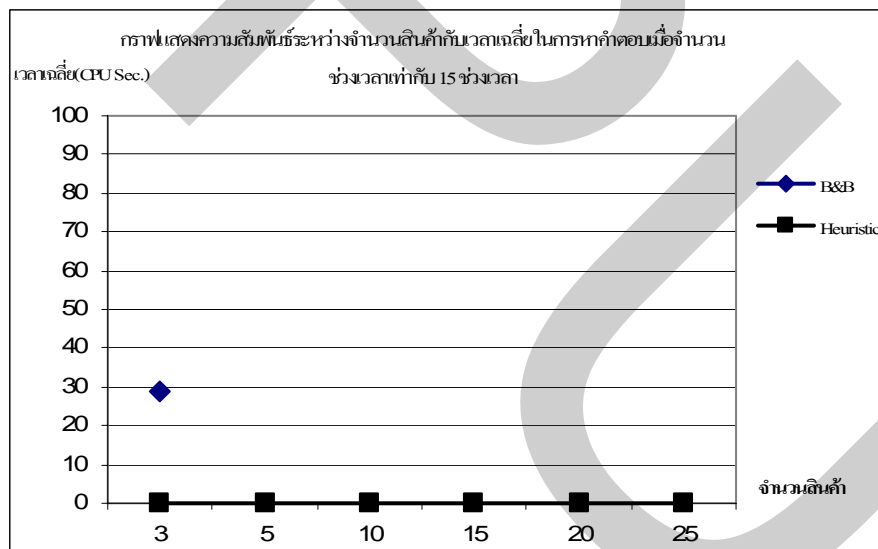
4.6.1.1 พิจารณาเวลาที่ใช้ในการคำนวณหาคำตอบของปัญหาการกำหนดปริมาณการตั้งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อจำนวนสินค้ามีการเปลี่ยนแปลง



ภาพที่ 4.1 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้ากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนช่วงเวลาเท่ากับ 5 ช่วงเวลา



ภาพที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้ากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวน
ช่วงเวลาเท่ากับ 10 ช่วงเวลา



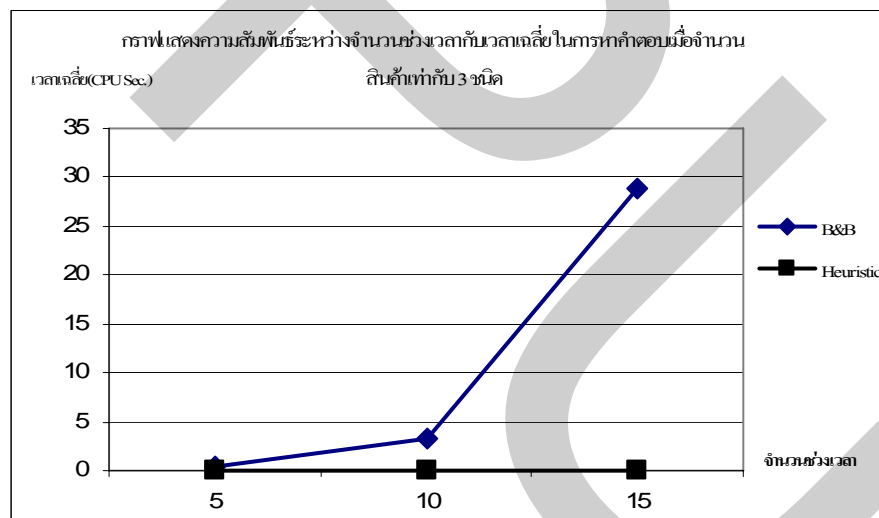
ภาพที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสินค้ากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวน
ช่วงเวลาเท่ากับ 15 ช่วงเวลา

จากภาพที่ 4.1 ถึงภาพที่ 4.3 จะเห็นได้ว่าจำนวนสินค้ามีผลต่อเวลาเฉลี่ยที่ใช้ในการ
คำนวณ หาคำตอบของวิธีการคำนวณแบบ Branch and Bound เพราะเมื่อจำนวนสินค้าเพิ่มขึ้นจะทำ
ให้ปัญหามีตัวแปรและความซับซ้อนมากขึ้น ส่งผลให้ต้องใช้เวลาในการคำนวณหาคำตอบมากขึ้น

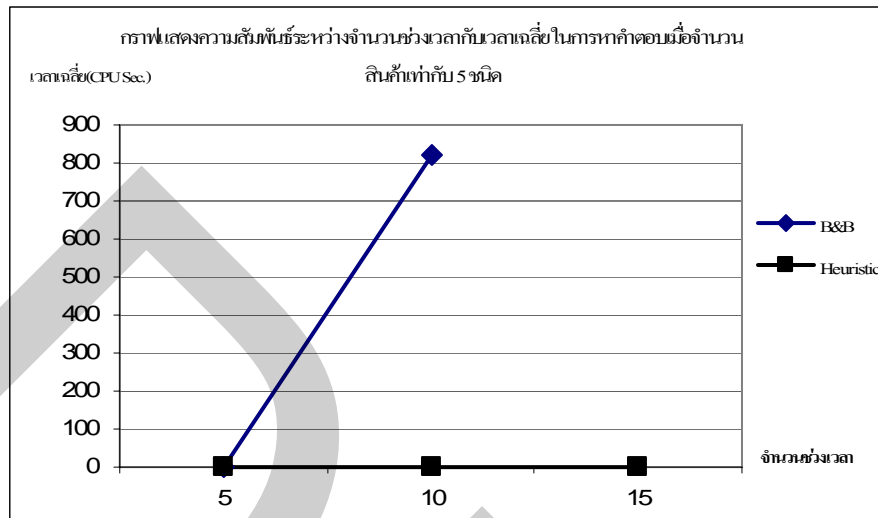
ด้วย แต่จากกราฟในภาพที่ 4.1 จะเห็นว่าในช่วงเวลาที่ 25 ใช้เวลาเฉลี่ยในการคำนวณหาคำตอบน้อยกว่าช่วงเวลา 20 สาเหตุก็เนื่องมาจากเวลาเฉลี่ยของช่วงเวลา 25 ไม่ได้นำเวลาของปัญหาที่ไม่สามารถหาคำตอบที่ดีที่สุดได้มาบวกรวมด้วย จึงทำให้ช่วงเวลา 25 ใช้เวลาเฉลี่ยในการคำนวณหาคำตอบน้อยกว่าช่วงเวลา 20

ส่วนเวลาเฉลี่ยที่ใช้ในการคำนวณหาคำตอบของวิธีการที่นำเสนอไม่สามารถเห็นได้ชัดเจนจากกราฟในภาพที่ 4.1 ถึงภาพที่ 4.3 เนื่องจากวิธีการที่นำเสนอใช้เวลาในการคำนวณคำตอบใกล้เคียงกันมากคือน้อยกว่า 10 มิลลิวินาทีซึ่งถือว่าน้อยมากทำให้เปรียบเทียบกันได้ยาก แต่ถ้าหากพิจารณาจากวิธีการหาคำตอบของวิธีการที่นำเสนอประกอบก็จะเห็นได้ว่าเมื่อจำนวนสินค้าเพิ่มขึ้นเวลาที่ใช้ในการคำนวณหาคำตอบก็จะเพิ่มขึ้นด้วยเช่นกัน

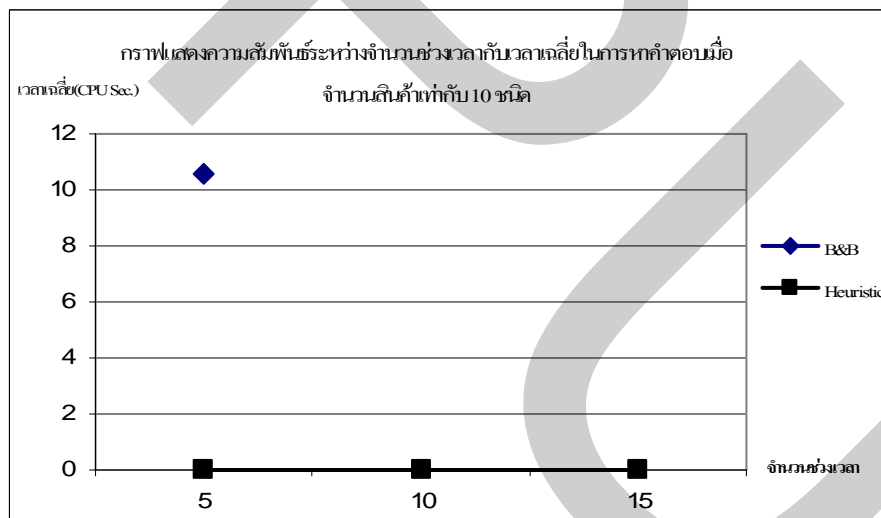
4.6.1.2 พิจารณาเวลาที่ใช้ในการคำนวณหาคำตอบของปัญหาการกำหนดปริมาณการผลิตหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อจำนวนช่วงเวลามีการเปลี่ยนแปลง



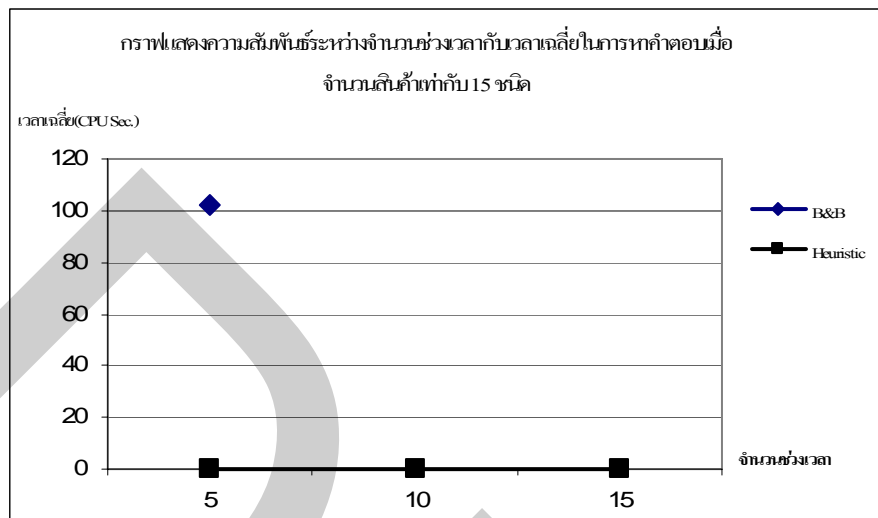
ภาพที่ 4.4 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 3 ชนิด



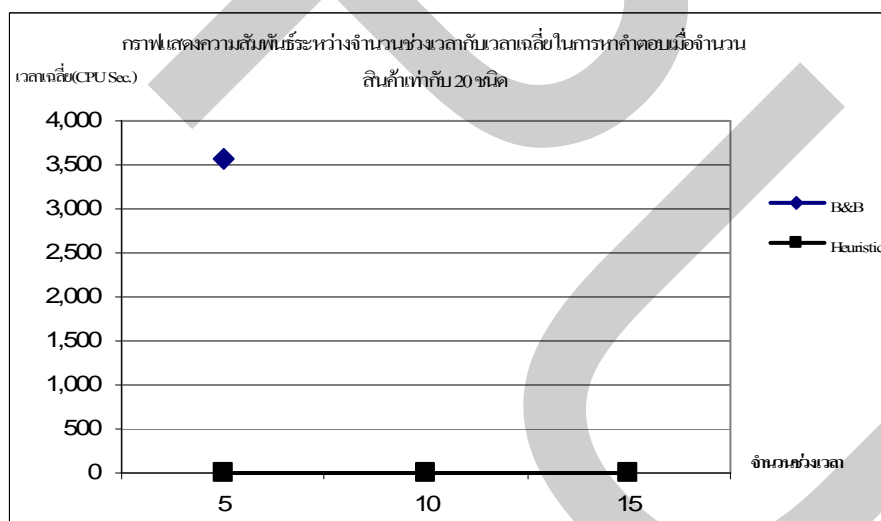
ภาพที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 5 ชนิด



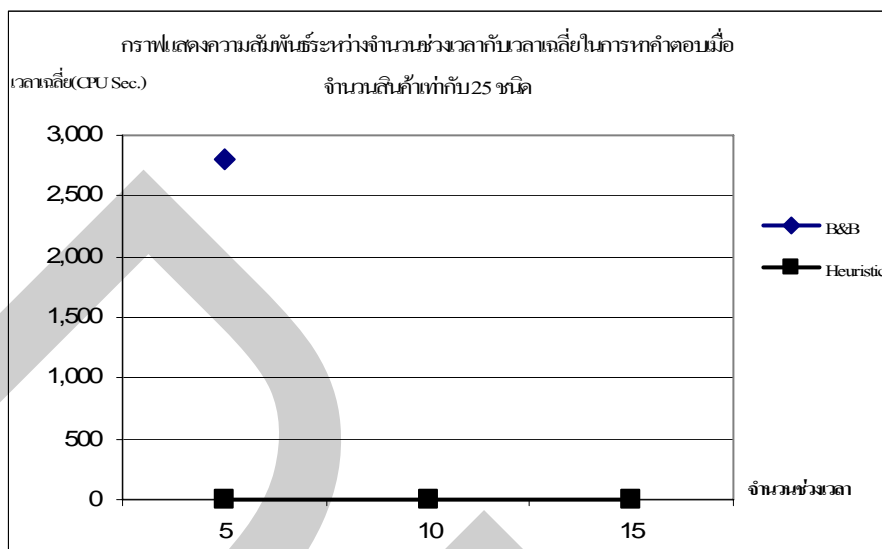
ภาพที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 10 ชนิด



ภาพที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 15 ชนิด



ภาพที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 20 ชนิด

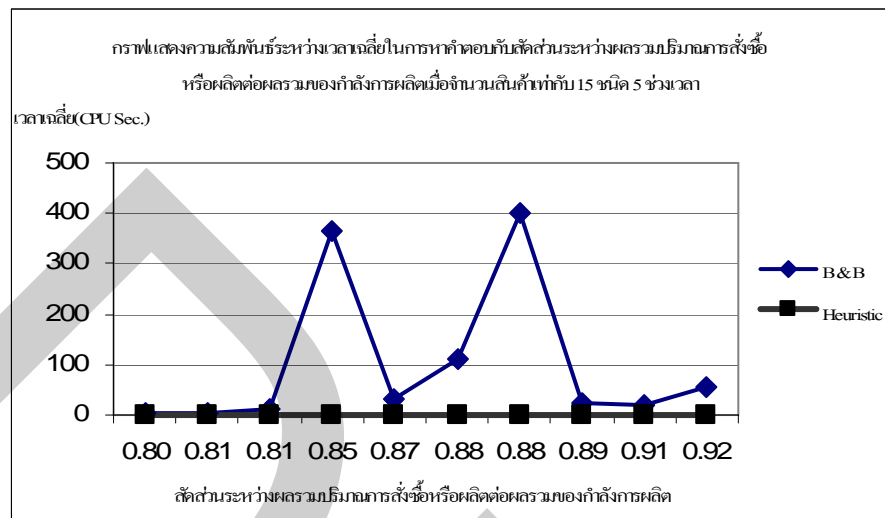


ภาพที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเวลาเฉลี่ยในการหาคำตอบเมื่อจำนวนสินค้าเท่ากับ 25 ชนิด

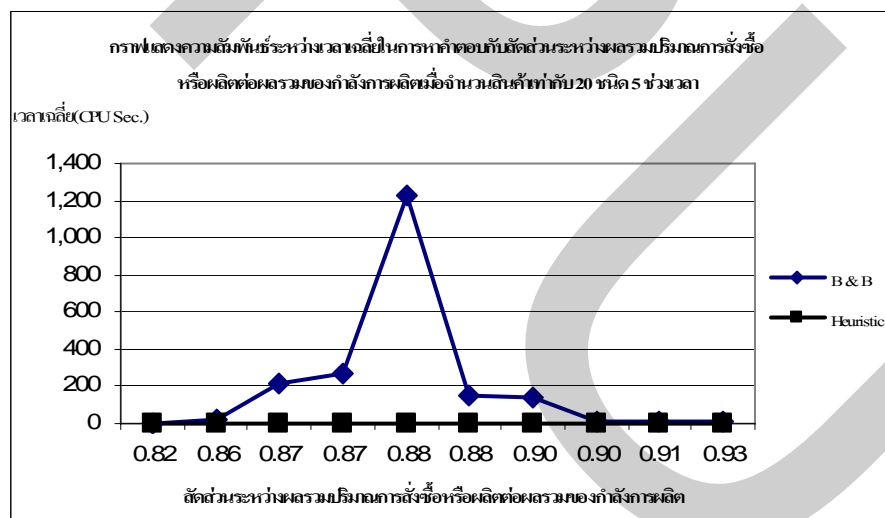
จากภาพที่ 4.4 ถึงภาพที่ 4.9 จะเห็นว่าจำนวนช่วงเวลามีผลต่อเวลาที่ใช้ในการคำนวณหาคำตอบของวิธีการคำนวณแบบ Branch and Bound เช่นเดียวกับจำนวนสินค้าเพราะเมื่อจำนวนช่วงเวลาเพิ่มขึ้นเวลาเฉลี่ยที่ใช้ในการคำนวณคำตอบก็เพิ่มขึ้นด้วยเช่นกัน ส่วนวิธีการที่นำเสนอใช้เวลาในการคำนวณหาคำตอบใกล้เคียงกันมากและเวลาที่ใช้น้อยมากทำให้เปรียบเทียบได้ยาก แต่ถ้าพิจารณาจากวิธีการหาคำตอบของวิธีการที่นำเสนอประกอบก็จะเห็นได้ว่าเมื่อจำนวนช่วงเวลาเพิ่มขึ้นเวลาที่ใช้ในการคำนวณหาคำตอบก็จะเพิ่มขึ้นด้วยเช่นกัน

ดังนั้นจะเห็นได้ว่าทั้งจำนวนสินค้าและจำนวนช่วงเวลามีผลต่อเวลาที่ใช้ในการคำนวณหาคำตอบในลักษณะที่แปรผันตรงต่อกัน

4.6.1.3 พิจารณาเวลาที่ใช้ในการคำนวณหาคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อตัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิตมีการเปลี่ยนแปลง



ภาพที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ยในการหาคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิตเมื่อจำนวนสินค้าเท่ากับ 15 ชนิด 5 ช่วงเวลา



ภาพที่ 4.11 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ยในการหาคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิตเมื่อจำนวนสินค้าเท่ากับ 20 ชนิด 5 ช่วงเวลา

จากภาพที่ 4.10 และภาพที่ 4.11 จะเห็นได้ว่าอัตราส่วนระหว่างผลรวมของปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของข้อจำกัดเชิงสมรรถภาพไม่มีแนวโน้มที่ชัดเจนต่อเวลาที่ใช้ในการ

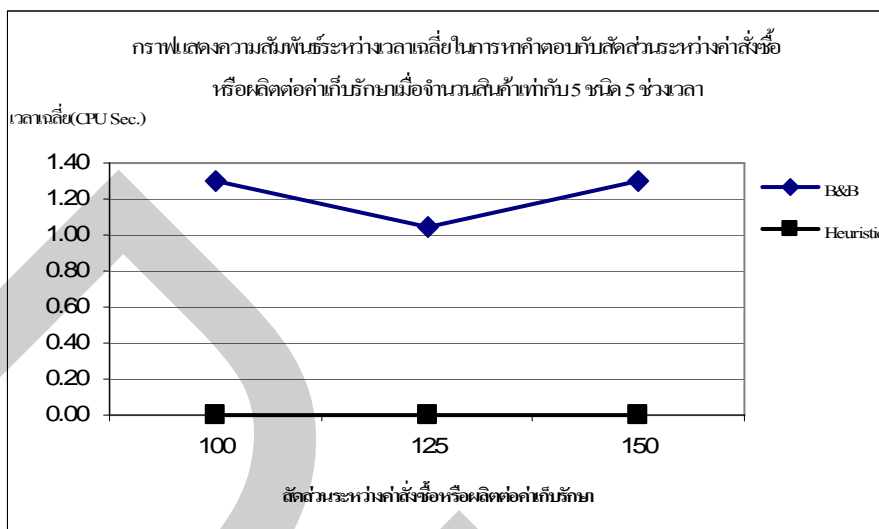
คำนวณหาคำตอบของวิธีการคำนวณแบบ Branch and Bound ส่วนวิธีการที่นำเสนอใช้เวลาในการคำนวณหาคำตอบใกล้เคียงกันมากคือน้อยกว่า 10 มิลลิวินาทีซึ่งถือว่าน้อยมากทำให้ไม่สามารถวิเคราะห์ในประเด็นนี้ได้

4.6.1.4 พิจารณาเวลาที่ใช้ในการคำนวณหาคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าการเก็บรักษามีการเปลี่ยนแปลง เนื่องจากปัญหาตัวอย่างทั้งหมดที่ใช้ในการทดลองมีสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าการเก็บรักษาเป็น 100 เท่า เท่ากันทั้งหมดทำให้ไม่สามารถวิเคราะห์ในประเด็นนี้ได้ ดังนั้นจึงได้สร้างปัญหาตัวอย่างขึ้นมาอีกกลุ่มหนึ่งเพื่อทดสอบวิธีการที่นำเสนอในกรณีนี้โดยเฉพาะอีกจำนวน 30 ตัวอย่างปัญหาและมีสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าการเก็บรักษาเป็น 100 , 125 และ 150 เท่าโดยทั้งหมดมีจำนวนสินค้า 5 ชนิด 5 ช่วงเวลาทั้งหมดและได้ผลการทดลองดังนี้

ตารางที่ 4.2 ผลการทดสอบปัญหาเพื่อวัดผลของสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา

จำนวน สินค้า	สัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าการเก็บรักษา					
	100		125		150	
และ จำนวน	เปอร์เซ็นต์ความ แตกต่างของ		เปอร์เซ็นต์ความ แตกต่างของ		เปอร์เซ็นต์ความ แตกต่างของ	
	คำตอบ	เวลา	คำตอบ	เวลา	คำตอบ	เวลา
5 x 5	4.00	129,900.00	4.47	103,900.00	5.19	129,900.00

จากผลการทดลองทั้งหมด 30 ปัญหาตัวอย่าง สามารถนำผลการทดลองทั้งหมดมาสร้างกราฟดังภาพที่ 4.12



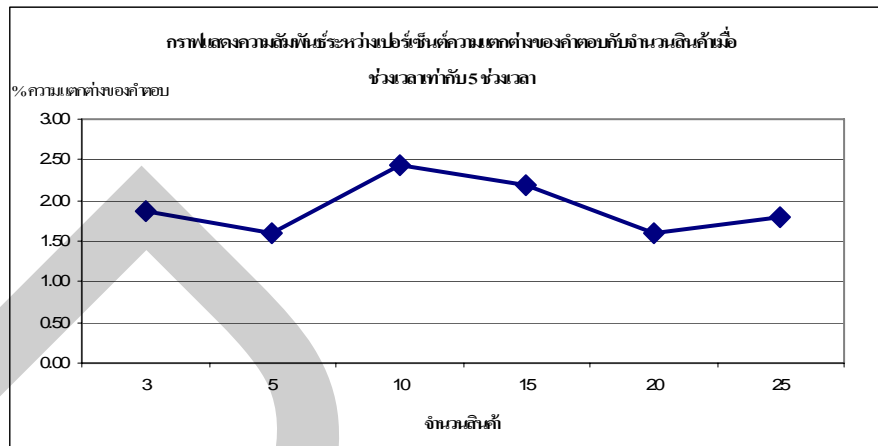
ภาพที่ 4.12 กราฟแสดงความสัมพันธ์ระหว่างเวลาเฉลี่ยในการหาคำตอบกับสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเมื่อจำนวนสินค้าเท่ากับ 5 ชนิด 5 ช่วงเวลา

จากภาพที่ 4.12 จะเห็นว่าความสัมพันธ์ระหว่างเวลาเฉลี่ยในการหาคำตอบกับสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษาของวิธีการคำนวณแบบ Branch and Bound ไม่มีแนวโน้มที่ชัดเจน ส่วนวิธีการที่นำเสนอใช้เวลาในการคำนวณหาคำตอบใกล้เคียงกันมากคือน้อยกว่า 10 มิลลิวินาทีซึ่งถือว่าน้อยมากทำให้ไม่สามารถวิเคราะห์ในประเด็นนี้ได้

4.6.2 การวิเคราะห์คุณภาพของคำตอบ

4.6.2.1 พิจารณาคุณภาพคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อจำนวนสินค้ามีการเปลี่ยนแปลง

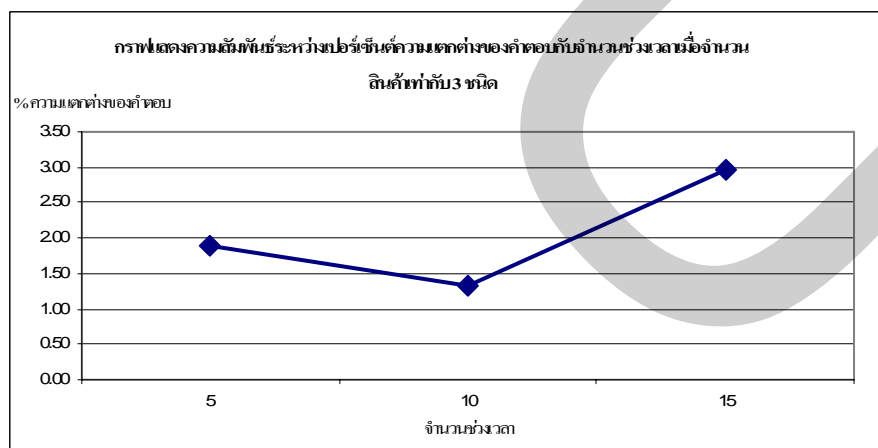
สำหรับการวิเคราะห์คุณภาพของคำตอบที่ได้จากวิธีการที่นำเสนอ จะใช้เปอร์เซ็นต์ของความคลาดเคลื่อนจากคำตอบที่ดีที่สุดที่ได้จากการคำนวณด้วยวิธี Branch and Bound มาวิเคราะห์ แต่เนื่องจากในกลุ่มปัญหาขนาดใหญ่ใช้เวลาในการคำนวณคำตอบสูงมากจนไม่สามารถทำการทดลองเพื่อหาคำตอบที่ดีที่สุดได้เป็นส่วนใหญ่ ทำให้การวิเคราะห์ผลการทดลองในแง่ของคุณภาพของคำตอบกระทำได้เฉพาะปัญหาขนาดเล็กและขนาดกลางเท่านั้น



ภาพที่ 4.13 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับจำนวนสินค้า
เมื่อช่วงเวลาเท่ากับ 5 ช่วงเวลา

จากภาพที่ 4.13 แสดงว่าความสัมพันธ์ระหว่างจำนวนสินค้ากับเปอร์เซ็นต์ความ
แตกต่างของคำตอบไม่มีแนวโน้มที่ชัดเจน

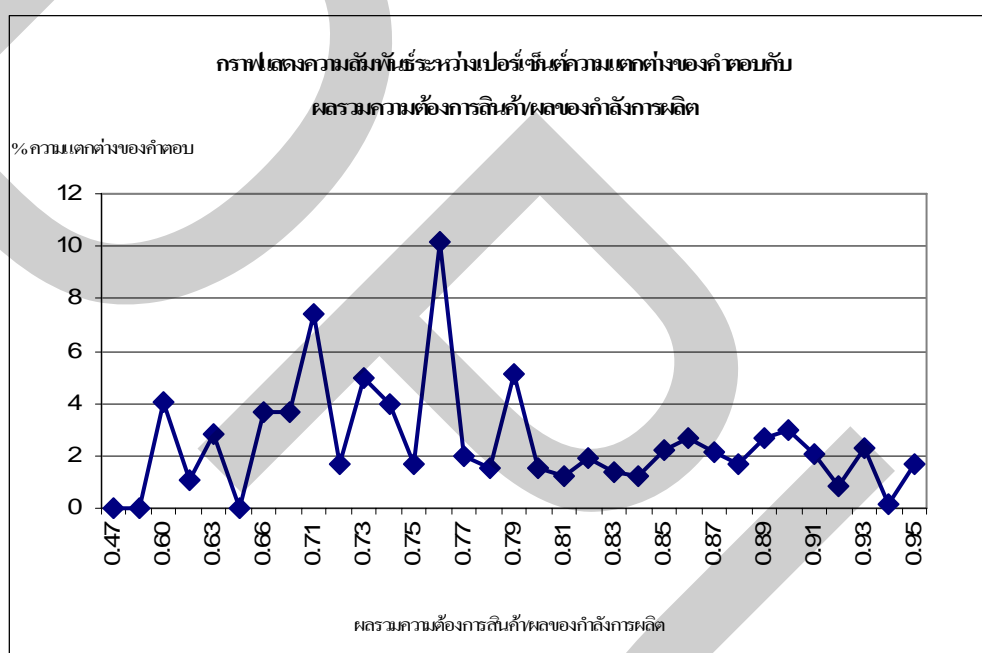
4.6.2.2 พิจารณาคุณภาพคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของ
สินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดี
ที่สุดเมื่อช่วงเวลามีการเปลี่ยนแปลง



ภาพที่ 4.14 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับจำนวน
ช่วงเวลาเมื่อจำนวนสินค้าเท่ากับ 3 ชนิด

จากภาพที่ 4.14 แสดงว่าความสัมพันธ์ระหว่างจำนวนช่วงเวลากับเปอร์เซ็นต์ความแตกต่างของคำตอบไม่มีแนวโน้มที่ชัดเจน

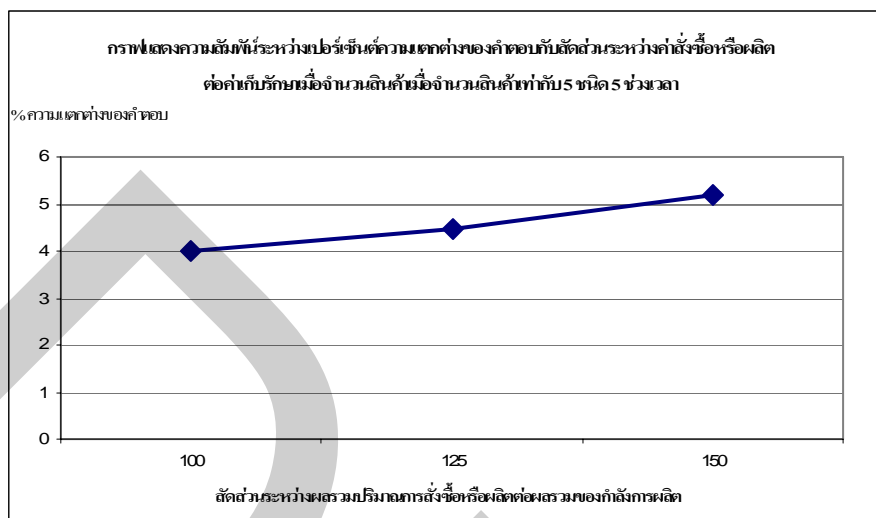
4.6.2.3 พิจารณาคูณภาพคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิตมีการเปลี่ยนแปลง



ภาพที่ 4.15 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิต

จากภาพที่ 4.15 จะเห็นได้ว่าความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับสัดส่วนระหว่างผลรวมปริมาณการสั่งซื้อหรือผลิตต่อผลรวมของกำลังการผลิตไม่มีแนวโน้มที่ชัดเจน

4.6.2.4 พิจารณาคูณภาพคำตอบของปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันทั้งวิธีการที่นำเสนอและวิธีหาคำตอบที่ดีที่สุดเมื่อสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษามีการเปลี่ยนแปลง จากผลการทดสอบของกลุ่มปัญหาเพื่อวัดผลของสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษาสามารถนำข้อมูลมาสร้างกราฟได้ดังภาพที่ 4.16



ภาพที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษา

จากภาพที่ 4.16 จะเห็นได้ว่าความสัมพันธ์ระหว่างเปอร์เซ็นต์ความแตกต่างของคำตอบกับสัดส่วนค่าการสั่งซื้อหรือผลิตต่อค่าเก็บรักษามีความสัมพันธ์แบบแปรผันตรงต่อกัน คือถ้าสัดส่วนค่าการสั่งซื้อหรือผลิตต่อค่าเก็บรักษามีค่ามากขึ้นเปอร์เซ็นต์ความแตกต่างของคำตอบก็จะมากขึ้นด้วย

4.7 สรุปการวิเคราะห์ผลการศึกษา

จากผลการวิเคราะห์ทั้งหมดพบว่าปัจจัยที่มีผลต่อเวลาที่ใช้ในการคำนวณเด่นชัดคือจำนวนสินค้าและจำนวนช่วงเวลา ส่วนในด้านคุณภาพของคำตอบปัจจัยที่มีผลต่อคุณภาพของคำตอบที่เด่นชัดที่สุดคือสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษา และจากผลการทดลองทั้งหมดจะเห็นได้ว่าวิธีการที่นำเสนอสามารถแก้ปัญหาเรื่องการใช้เวลาในการคำนวณหาคำตอบสูงของวิธีการแก้ปัญหาคำตอบที่ดีที่สุดได้เป็นอย่างดี

บทที่ 5

สรุปผลการศึกษา

จากการทดลองและการวิเคราะห์ผลการทดลองของวิธีการแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกัน ตามวิธีที่นำเสนอโดยเปรียบเทียบกับวิธีการหาคำตอบที่ดีที่สุดด้วยวิธีการคำนวณแบบ Branch and Bound สามารถสรุปผลการศึกษาได้ดังนี้

5.1 สรุปผลการศึกษา

จากผลการทดลองของปัญหาปริมาณการสั่งซื้อหรือผลิตสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดทางสมรรถภาพร่วมกันและอยู่ภายใต้ข้อกำหนดที่ตั้งขึ้นด้วยตัวอย่างปัญหา 3 กลุ่มจำนวน 180 ตัวอย่าง วิธีการหาคำตอบที่ดีที่สุดที่ใช้วิธีการคำนวณแบบ Branch and Bound สามารถหาคำตอบของปัญหาได้จำนวน 85 ปัญหาและไม่สามารถหาคำตอบของปัญหาได้จำนวน 95 ปัญหา ส่วนวิธีที่นำเสนอสามารถแก้ปัญหาตัวอย่างได้ทั้งหมด 180 ตัวอย่างและใช้เวลาเฉลี่ยในการคำนวณหาคำตอบเท่ากับ 5 มิลลิวินาทีเท่านั้น ซึ่งน้อยกว่าวิธีการหาคำตอบที่ดีที่สุดในทุกๆปัญหานั้นวิธีที่นำเสนอจึงสามารถแก้ปัญหาในด้านเวลาในการคำนวณหาคำตอบได้ผลเป็นอย่างมาก ส่วนในด้านของคุณภาพของคำตอบวิธีที่นำเสนอยังให้ผลของคำตอบที่ไม่ดีนัก โดยมีเปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยจากคำตอบที่ดีที่สุดในการหาคำตอบได้ทั้ง 2 วิธีเท่ากับ 2.23 และมีเปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยจากคำตอบที่ดีในกลุ่มปัญหาตัวอย่างขนาดเล็กเท่ากับ 1.81 ในกลุ่มปัญหาขนาดกลางเท่ากับ 3.07 และกลุ่มปัญหาขนาดใหญ่เท่ากับ 1.80

ในด้านเวลาที่ใช้ในการคำนวณตอบ วิธีที่นำเสนอสามารถลดเวลาที่ใช้ในการคำนวณหาคำตอบได้ผลเป็นที่น่าพอใจ โดยในวิธีที่นำเสนอปัจจัยที่มีผลต่อเวลาในการคำนวณหาคำตอบนั้นอาจไม่สามารถพิจารณาจากผลการทดลองได้เนื่องจากวิธีที่นำเสนอใช้เวลาในการคำนวณหาคำตอบน้อยมากจนไม่สามารถวัดได้ชัดเจน แต่ถ้าหากพิจารณาจากวิธีการในการหาคำตอบของวิธีที่นำเสนอแล้วจะเห็นได้ว่าจำนวนสินค้าและจำนวนช่วงเวลามีมากขึ้นก็ต้องการเวลาเพื่อหาคำตอบมากขึ้นด้วยดังนั้นจึงสรุปได้ว่า เวลาที่ใช้ในการคำนวณหาคำตอบแปรผัน

ตรงกับจำนวนสินค้าและจำนวนช่วงเวลาซึ่งก็สอดคล้องกับวิธีการหาคำตอบที่ดีที่สุดเช่นกัน ส่วนปัจจัยอื่น ไม่มีแนวโน้มที่ชัดเจน

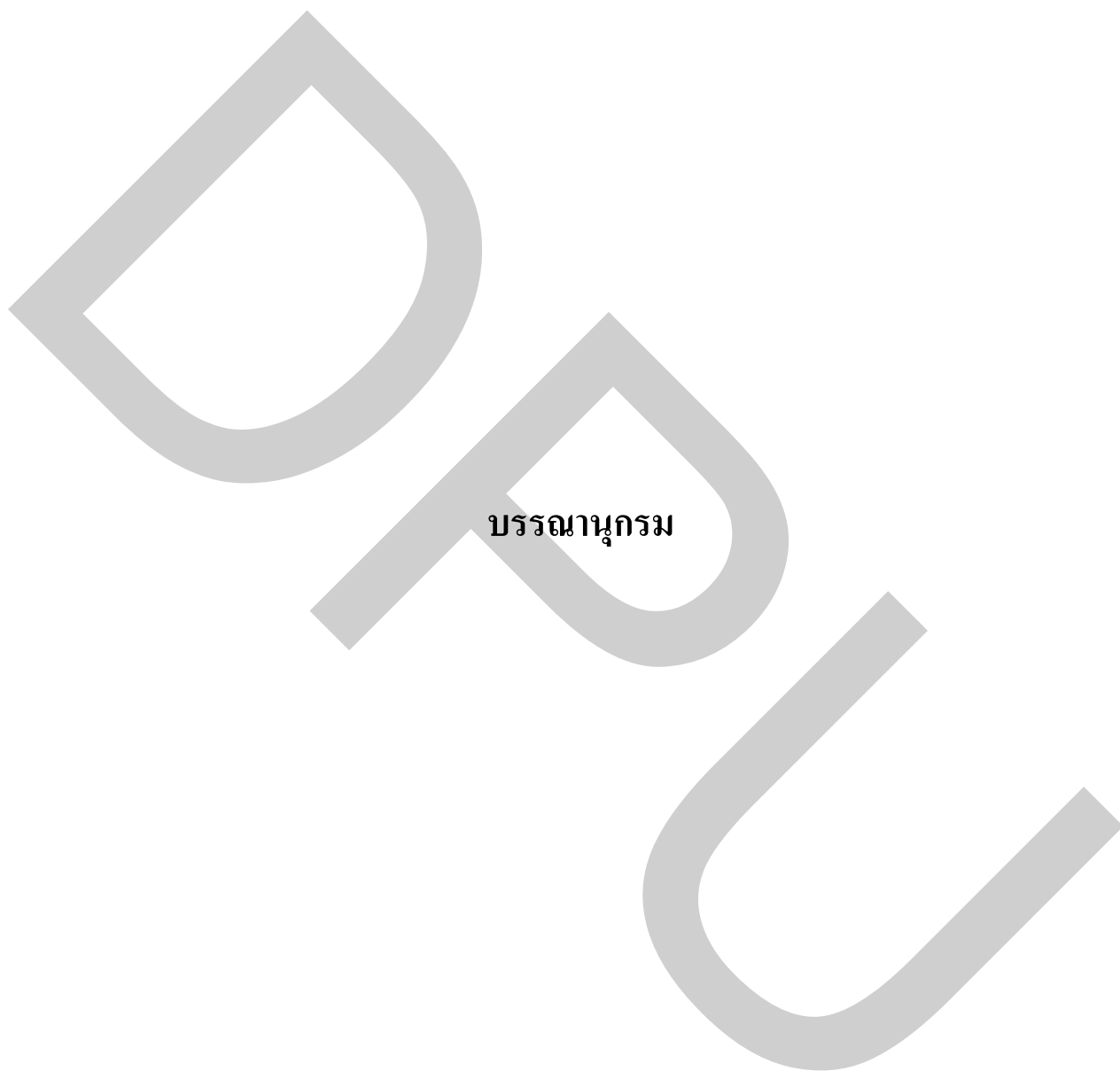
ในด้านของคุณภาพของคำตอบ วิธีการที่นำเสนอให้ผลการทดลองที่ดีในระดับหนึ่ง และจากผลการทดลองปัจจัยที่มีผลต่อคุณภาพของคำตอบคือ สัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษา ส่วนปัจจัยอื่นๆ ไม่มีแนวโน้มที่ชัดเจน

5.2 ประโยชน์ที่ได้รับจากงานวิจัย

จากผลการทดลองจะเห็นได้ว่าวิธีการที่นำเสนอประสบความสำเร็จในการลดเวลาในการคำนวณหาคำตอบซึ่งเป็นข้อเสียของวิธีการหาคำตอบที่ดีที่สุดได้เป็นอย่างดี ส่วนเรื่องคุณภาพของคำตอบซึ่งมีความคลาดเคลื่อนเฉลี่ยรวมจากคำตอบที่ดีที่สุดไม่เกิน 2.23 เท่านั้นก็น่าจะเป็นจำนวนความคลาดเคลื่อนที่สามารถยอมรับได้เมื่อเปรียบเทียบเวลาที่ต้องเสียไปเพื่อรอคำตอบที่ดีที่สุด และในกรณีที่ปัญหามีขนาดใหญ่ขึ้นจากการทดลองใช้เวลาในการคำนวณหาคำตอบสูงมากในบางปัญหาอาจถึงขั้นที่ไม่สามารถคำนวณหาคำตอบด้วยเครื่องคอมพิวเตอร์ส่วนบุคคลได้ นอกจากนี้โปรแกรมที่ได้พัฒนาขึ้นยังสามารถใช้งานได้ง่ายกว่าโปรแกรมสำเร็จรูปที่ใช้สำหรับคำนวณหาคำตอบที่ดีที่สุดอีกด้วย

5.3 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

การวิจัยเพื่อพัฒนาวิธีการแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันและอยู่ภายใต้ข้อกำหนดที่ตั้งขึ้นในอนาคตนั้น ควรหาวิธีการในการปรับปรุงคุณภาพของคำตอบให้ดีขึ้น โดยอาจจะยอมให้สามารถย้ายปริมาณความต้องการสินค้าไปยังช่วงเวลาถัดไปได้ด้วยแทนที่จะพิจารณาย้ายความต้องการสินค้าไปยังช่วงเวลาก่อนหน้าอย่างเดียวนั่น



บรรณานุกรม

บรรณานุกรม

ภาษาไทย

หนังสือ

พิชิต สุขเจริญพงษ์. (2547). **การจัดการวิศวกรรมการผลิต**. กรุงเทพฯ: ซีเอ็ดดูเคชั่น.

พิภพ ลลิตาภรณ์. (2543). **การบริหารของคลังระบบ MRP และ ROP**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

_____. (2545). **ระบบการวางแผนและควบคุมการผลิต (ฉบับปรับปรุง)**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

บทความ

ไพฑูรย์ ฮ้อยิ่ง. (2549, มกราคม). “การหาปริมาณการสั่งซื้อวัสดุชิ้นส่วน กรณีโปรโมชันลดราคาหรือปรับขึ้นราคา.” **วารสารเทคนิค**, 23, 257. หน้า 145-149.

วิทยานิพนธ์

ภัทร์อร แสงฤดี. (2547). **วิธีหาคำตอบที่เหมาะสมสำหรับปัญหาการกำหนดปริมาณการสั่งซื้อที่มีสินค้าคงคลังหลายชนิดโดยมีข้อจำกัดเชิงสมรรถภาพร่วมกัน**. วิทยานิพนธ์ปริญญาโท สาขาวิชาวิศวกรรมอุตสาหการ. กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย.

ภาษาต่างประเทศ

BOOKS

Radin,L. Radin. (1998). **Optimization In Operations Research**. New Jersey: Printice-Hall.

Silver,E.A. , Pyke, D.F.and Peterson, R. (1998). **Inventory Management and Production Planning and Scheduling** (3 rd ed.). New York: John Wiley & Sons.

ARTICLES

- Dixon, P.S. and Silver, E.A. (1981). "heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem." **Journal of operations management**, **2**. pp.23-39.
- Dogramaci, A. , J.C. and Adam N.R. (1981). "The dynamic lot-sizing problem for multiple items under limited capacity." **AIIE Transactions**, **13**. pp.294-303.
- Kami, R. and Roll, Y. (1982). "A heuristic algorithm for the multi-item lot-sizing problem with capacity constraints." **IIE Transactions**, **14**. pp.249-256.
- Millar , H.H. and Yang, M. (1993). "An application of lagrangean decomposition to capacitated multi-item lot sizing problem." **Computers Operations Research**, **20**. pp.409-420.
- Wagner, H. M. and T. Whitin. (1958). "Dynamic Version of the Economic Lot Size Model." **Management Science**, **5**. pp.89-96.

DISSERTATION

- Prichanont, S. (2002). **Multi-item multi-attribute outsourcing in manufacturing supply networks**. Doctor of Philosophy. Industrial Engineering: University of Wisconsin-Madison.

ด

พ

ภาคผนวก

ช



ภาคผนวก ก

ปัญหาตัวอย่างและผลการทดสอบของปัญหา

ภาคผนวก ก

ปัญหาตัวอย่างและผลการทดสอบของปัญหา

การทดสอบวิธีการแก้ปัญหาคำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันและอยู่ภายใต้ข้อกำหนดที่ตั้งขึ้นด้วยวิธีการที่นำเสนอ นั้นจะใช้ปัญหาตัวอย่างซึ่งจำแนกออกเป็น 4 กลุ่มคือ กลุ่มปัญหาขนาดเล็ก กลุ่มปัญหขนาดกลาง กลุ่มปัญหาขนาดใหญ่และกลุ่มปัญหาตัวอย่างสำหรับทดสอบผลของสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา ในการทดสอบประสิทธิภาพของวิธีการที่ได้นำเสนอ ผลการทดสอบของปัญหาในกลุ่มต่างๆได้ถูกรวบรวมไว้ภายใน ภาคผนวก ก นี้

ก.1 ปัญหาตัวอย่าง

ก.1.1 ในการเปรียบเทียบผลของการทดลองนั้นจะต้องทำการแปลงข้อมูลต่างๆของปัญหาให้อยู่ในรูปของสมการคณิตศาสตร์เพื่อนำไปหาค่าที่ดีที่สุดด้วย โปรแกรมสำเร็จรูปและนำคำตอบที่ได้และเวลาที่ใช้ในการคำนวณมาเปรียบเทียบกับคำตอบและเวลาที่ใช้คำนวณตามวิธีการที่นำเสนอ ซึ่งสามารถแสดงรูปแบบปัญหาและสมการคณิตศาสตร์ได้ดังนี้

สินค้าช่วงเวลา	1	2	3	4	5	ค่าสั่งซื้อหรือผลิต	ค่าเก็บรักษาต่อหน่วย
สินค้าชนิดที่ 1	25	30	7	2	32	12	0.12
สินค้าชนิดที่ 2	27	32	16	55	28	432	4.32
สินค้าชนิดที่ 3	17	12	40	10	27	172	1.72
ความต้องการรวม	69	74	63	67	87	ต้นทุนรวม	3080
กำลังผลิต	100	100	100	100	100		

ภาพที่ ก.1 ข้อมูลตัวอย่างของปัญหาขนาดสินค้า 3 ชนิด 5 ช่วงเวลา

จากข้อมูลต่างๆตามภาพที่ ก.1ของปัญหาขนาดสินค้า 3 ชนิด 6 ช่วงเวลา สามารถแปลงเป็นสมการคณิตศาสตร์ได้ดังนี้

MIN $0.6X_{11} + 0.48X_{12} + 0.36X_{13} + 0.24X_{14} + 0.12X_{15} + 21.6X_{21} + 17.28X_{22}$
 $+ 12.96X_{23} + 8.64X_{24} + 4.32X_{25} + 8.6X_{31} + 6.88X_{32} + 5.16X_{33} + 3.44X_{34}$
 $+ 1.72X_{35} + 12Z_{11} + 12Z_{12} + 12Z_{13} + 12Z_{14} + 12Z_{15} + 432Z_{21} + 432Z_{22}$
 $+ 432Z_{23} + 432Z_{24} + 432Z_{25} + 172Z_{31} + 172Z_{32} + 172Z_{33} + 172Z_{34} + 172Z_{35}$
 -Constant

SUBJECT TO

$$X_{11} \geq 25$$

$$X_{11} + X_{12} \geq 55$$

$$X_{11} + X_{12} + X_{13} \geq 62$$

$$X_{11} + X_{12} + X_{13} + X_{14} \geq 64$$

$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} \geq 96$$

$$X_{21} \geq 27$$

$$X_{21} + X_{22} \geq 59$$

$$X_{21} + X_{22} + X_{23} \geq 75$$

$$X_{21} + X_{22} + X_{23} + X_{24} \geq 130$$

$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} \geq 158$$

$$X_{31} \geq 17$$

$$X_{31} + X_{32} \geq 29$$

$$X_{31} + X_{32} + X_{33} \geq 69$$

$$X_{31} + X_{32} + X_{33} + X_{34} \geq 79$$

$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} \geq 106$$

$$X_{11} + X_{21} + X_{31} \leq 100$$

$$X_{12} + X_{22} + X_{32} \leq 100$$

$$X_{13} + X_{23} + X_{33} \leq 100$$

$$X_{14} + X_{24} + X_{34} \leq 100$$

$$X_{15} + X_{25} + X_{35} \leq 100$$

$$X_{11} \geq 0$$

X12 ≥ 0

X13 ≥ 0

X14 ≥ 0

X15 ≥ 0

X21 ≥ 0

X22 ≥ 0

X23 ≥ 0

X24 ≥ 0

X25 ≥ 0

X31 ≥ 0

X32 ≥ 0

X33 ≥ 0

X34 ≥ 0

X35 ≥ 0

X11 - 100Z11 ≤ 0

X12 - 100Z12 ≤ 0

X13 - 100Z13 ≤ 0

X14 - 100Z14 ≤ 0

X15 - 100Z15 ≤ 0

X21 - 100Z21 ≤ 0

X22 - 100Z22 ≤ 0

X23 - 100Z23 ≤ 0

X24 - 100Z24 ≤ 0

X25 - 100Z25 ≤ 0

X31 - 100Z31 ≤ 0

X32 - 100Z32 ≤ 0

X33 - 100Z33 ≤ 0

X34 - 100Z34 ≤ 0

X35 - 100Z35 ≤ 0

Constant=2491.92

END

GIN Z11

GIN Z12

GIN Z13

GIN Z14

GIN Z15

GIN Z21

GIN Z22

GIN Z23

GIN Z24

GIN Z25

GIN Z31

GIN Z32

GIN Z33

GIN Z34

GIN Z35

GIN X11

GIN X12

GIN X13

GIN X14

GIN X15

GIN X21

GIN X22

GIN X23

GIN X24

GIN X25

GIN X31

GIN X32

GIN X33

GIN X34

GIN X35

ก.2 ผลการทดลองของกลุ่มปัญหาขนาดเล็ก

ก.2.1 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 5 ช่วงเวลา จำนวนปัญหาทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.1

ตารางที่ ก.1 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	2,631.36	2,574.16	0.010	0.702	2.22
2	3,470.60	3,339.20	0.000	0.320	3.94
3	1,998.12	1,998.12	0.005	0.160	0.00
4	1,167.90	1,135.72	0.000	0.280	2.83
5	3,135.83	3,135.83	0.000	0.050	0.00
6	1,342.40	1,342.40	0.000	0.200	0.00
7	4,413.23	4,190.46	0.000	0.230	5.32
8	2,136.17	2,136.17	0.005	0.180	0.00
9	1,799.70	1,797.14	0.000	1.000	0.14
10	4,322.68	4,141.68	0.000	0.230	4.37
เฉลี่ย			0.002	0.335	1.88

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหาและมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 1.88

ก.2.2 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 10 ช่วงเวลา จำนวนปัญหาทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.2

ตารางที่ ก.2 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	6,135.99	6,036.75	0.005	4.000	1.64
2	5,607.82	5,607.82	0.000	6.000	0.00
3	5,916.33	5,874.17	0.000	1.000	0.72
4	7,277.76	7,204.52	0.000	9.000	1.02
5	4,028.10	4,011.22	0.000	1.000	0.42
6	889.78	858.42	0.010	2.000	3.65
7	5,183.68	4,999.81	0.000	2.000	3.68
8	1,681.25	1,673.94	0.000	1.000	0.44
9	2,523.84	2,518.92	0.005	2.000	0.20
10	2,761.44	2,723.27	0.000	4.000	1.40
		เฉลี่ย	0.002	3.200	1.32

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหาและมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 1.32

ก.2.3 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 5 ช่วงเวลา จำนวนปัญหาทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.3

ตารางที่ ก.3 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	4,456.42	4423.03	0.000	1.000	0.75
2	2,564.72	2,497.78	0.000	2.000	2.68
3	3,474.96	3,474.96	0.005	1.000	0.00
4	2,373.53	2,373.53	0.010	1.000	0.00
5	2,201.22	2,201.22	0.000	1.000	0.00
6	4,135.52	3,956.83	0.000	1.000	4.52
7	3,162.45	3,162.24	0.010	1.000	0.01
8	3,121.86	3,089.04	0.000	1.000	1.06
9	3,243.11	3,165.09	0.000	2.000	2.47
10	3,724.87	3,559.03	0.000	2.000	4.66
		เฉลี่ย	0.003	1.300	1.61

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหาและมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 1.61

ก.2.4 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 10 ชนิด 5 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.4

ตารางที่ ก.4 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 10 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	9,715.45	9,680.89	0.005	4.000	0.36
2	9,339.04	9,239.21	0.000	11.000	1.08
3	8,553.00	8,310.60	0.010	4.000	2.92
4	10,679.47	10,288.33	0.000	31.000	3.80
5	9,522.51	8,958.20	0.000	3.000	6.30
6	9,721.64	9,277.32	0.000	13.000	4.79
7	9,034.50	8,931.93	0.000	1.000	1.15
8	10,873.77	10,872.85	0.000	10.000	0.01
9	8,903.76	8,701.18	0.000	23.000	2.33
10	10,505.39	10,335.22	0.000	6.000	1.65
เฉลี่ย			0.002	10.600	2.44

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหาและมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 2.44

ก.3 ผลการทดลองของกลุ่มปัญหาขนาดกลาง

ก.3.1 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.5

ตารางที่ ก.5 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 3 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	6,564.18	6,477.32	0.000	15.000	1.34
2	3,699.05	3,540.51	0.005	52.000	4.48
3	11,431.64	10,834.17	0.000	39.000	5.51
4	6,392.85	6,283.07	0.000	5.000	1.75
5	4,557.17	4,480.95	0.010	25.000	1.70
6	13,277.44	12,478.72	0.000	64.000	6.40
7	5,840.37	5,611.29	0.000	12.000	4.08
8	8,282.94	8,184.72	0.000	50.000	1.20
9	5,426.85	5,409.14	0.005	5.000	0.33
10	7,990.20	7,779.42	0.000	21.000	2.71
		เฉลี่ย	0.002	28.800	2.95

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 2.95

ก.3.2 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 10 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.6

ตารางที่ ก.6 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	5,559.31	5,492.49	0.000	1430.000	1.22
2	4,390.89	4,221.27	0.005	27.000	4.02
3	3,865.94	3,806.52	0.010	21.000	1.56
4	5,084.66	4,947.49	0.005	37.000	2.77
5	7,382.20	6,203.35	0.000	5820.000	19.00
6	4,087.86	3,806.90	0.000	23.000	7.38
7	5,804.79	5,452.95	0.005	18.000	6.45
8	4,811.91	4,470.25	0.000	22.000	7.64
9	6,855.00	6,753.00	0.000	167.000	1.51
10	7,716.00	7,434.00	0.000	661.000	3.79
		เฉลี่ย	0.003	822.600	5.54

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 5.54

ก.3.3 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 5 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.7

ตารางที่ ก.7 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	14,004.80	13,721.88	0.005	4.000	2.06
2	17,849.18	17,339.24	0.000	366.000	2.94
3	16,214.17	15,738.31	0.000	399.000	3.02
4	13,589.67	13,447.18	0.000	19.000	1.06
5	14,698.71	14,100.56	0.010	22.000	4.24
6	15,506.60	15,335.26	0.000	5.000	1.12
7	16,565.32	16,148.42	0.000	32.000	2.58
8	20,045.92	19,680.94	0.005	110.000	1.85
9	16,000.34	15,782.09	0.000	11.000	1.38
10	15,734.51	15,475.91	0.000	54.000	1.67
		เฉลี่ย	0.002	102.200	2.19

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหา โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 2.19

ก.3.4 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 5 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.8

ตารางที่ ก.8 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	20,503.55	20,284.32	0.000	137.000	1.08
2	19,084.67	18,834.43	0.000	14.000	1.33
3	19,159.48	18,627.34	0.000	1,223.000	2.86
4	18,970.95	18,830.99	0.000	271.000	0.74
5	20,427.42	20,247.61	0.005	13.000	0.89
6	17,657.72	17,423.43	0.000	214.000	1.34
7	16,190.30	16,021.20	0.010	19.000	1.06
8	17,050.80	16,641.79	0.000	33,692.590	2.46
9	20,369.41	19,905.58	0.000	13.000	2.33
10	16,437.36	16,146.58	0.005	154.000	1.80
		เฉลี่ย	0.002	3575.059	1.59

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหา โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 1.59

ก.4 ผลการทดลองของกลุ่มปัญหาขนาดใหญ่

ก.4.1 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.9

ตารางที่ ก.9 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 5 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	8,307.53	*	0.010	*	*
2	8,153.85	*	0.005	*	*
3	9,531.72	*	0.000	*	*
4	9,016.10	*	0.005	*	*
5	9,750.23	*	0.000	*	*
6	8,146.34	*	0.005	*	*
7	14,066.09	*	0.000	*	*
8	7,124.15	*	0.000	*	*
9	9,146.82	*	0.005	*	*
10	7,281.86	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.2 ตัวอย่างปัญหาขนาดสินค้า 10 ชนิด 10 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.10

ตารางที่ ก.10 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 10 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	17,069.87	*	0.000	*	*
2	19,242.70	*	0.000	*	*
3	15,845.74	*	0.005	*	*
4	13,912.22	*	0.005	*	*
5	14,343.88	*	0.005	*	*
6	15,097.27	*	0.000	*	*
7	21,903.79	*	0.000	*	*
8	20,170.79	*	0.000	*	*
9	21,112.85	*	0.010	*	*
10	15,991.95	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.3 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 10 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.11

ตารางที่ ก.11 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 10 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	22,530.17	*	0.010	*	*
2	23,618.33	*	0.000	*	*
3	31,109.72	*	0.000	*	*
4	22,060.53	*	0.000	*	*
5	23,354.75	*	0.010	*	*
6	17,035.96	*	0.000	*	*
7	15,394.77	*	0.000	*	*
8	32,564.21	*	0.005	*	*
9	24,186.69	*	0.005	*	*
10	31,028.76	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.4 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 10 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.12

ตารางที่ ก.12 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	29,635.48	*	0.005	*	*
2	31,922.03	*	0.000	*	*
3	22,595.11	*	0.005	*	*
4	26,054.21	*	0.005	*	*
5	23,663.19	*	0.015	*	*
6	26,312.77	*	0.000	*	*
7	28,371.78	*	0.000	*	*
8	25,166.56	*	0.000	*	*
9	21,575.34	*	0.000	*	*
10	28,583.22	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.5 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.13

ตารางที่ ก.13 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 15 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	37,163.52	*	0.000	*	*
2	41,120.80	*	0.000	*	*
3	54,152.49	*	0.000	*	*
4	41,188.95	*	0.010	*	*
5	36,381.87	*	0.000	*	*
6	32,447.97	*	0.005	*	*
7	36,957.60	*	0.000	*	*
8	42,964.41	*	0.010	*	*
9	35,081.62	*	0.000	*	*
10	38,665.23	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.6 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 10 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.14

ตารางที่ ก.14 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	37,678.39	*	0.005	*	*
2	37,538.88	*	0.000	*	*
3	35,785.98	*	0.005	*	*
4	48,122.79	*	0.000	*	*
5	44,871.82	*	0.010	*	*
6	35,494.20	*	0.000	*	*
7	35,451.59	*	0.000	*	*
8	44,687.12	*	0.005	*	*
9	32,399.50	*	0.000	*	*
10	35,288.67	*	0.000	*	*
		เฉลี่ย	0.003	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.7 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.15

ตารางที่ ก.15 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 20 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	67,330.90	*	0.000	*	*
2	55,899.35	*	0.005	*	*
3	47,218.57	*	0.000	*	*
4	61,293.81	*	0.010	*	*
5	57,841.84	*	0.000	*	*
6	55,487.53	*	0.010	*	*
7	46,228.91	*	0.010	*	*
8	45,055.43	*	0.000	*	*
9	53,070.30	*	0.000	*	*
10	60,141.08	*	0.000	*	*
		เฉลี่ย	0.004	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.8 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 5 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.16

ตารางที่ ก.16 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	22,750.78	22,502.42	0.000	135.000	1.10
2	22,658.14	21,975.48	0.005	16,383.000	3.11
3	23,513.70	*	0.000	*	*
4	26,133.87	*	0.005	*	*
5	25,402.72	25,372.28	0.000	231.000	0.12
6	27,977.89	*	0.000	*	*
7	26,007.08	*	0.000	*	*
8	25,605.96	24,781.22	0.010	60.000	3.33
9	22,960.43	*	0.000	*	*
10	25,590.71	25,246.53	0.005	11,238.000	1.36
		เฉลี่ย	0.003	2,804.700	1.80

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound สามารถหาคำตอบได้จำนวน 5 ปัญหาเท่านั้น และไม่สามารถคำนวณหาคำตอบได้จำนวน 5 ปัญหาเนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้ สำหรับปัญหาที่สามารถคำนวณหาคำตอบได้ทั้ง 2 วิธีมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 1.80

ก.4.9 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 10 ช่วงเวลา จำนวนทั้งหมด 10 ตัวอย่าง สามารถแสดงผลการทดลองได้ดังตารางที่ ก.17

ตารางที่ ก.17 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 10 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	50,980.49	*	0.010	*	*
2	39,699.28	*	0.000	*	*
3	52,339.64	*	0.010	*	*
4	44,016.21	*	0.000	*	*
5	45,498.01	*	0.005	*	*
6	40,042.03	*	0.000	*	*
7	47,853.02	*	0.005	*	*
8	47,448.56	*	0.000	*	*
9	43,949.64	*	0.010	*	*
10	41,921.97	*	0.005	*	*
		เฉลี่ย	0.005	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.4.10 ผลการทดสอบของปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 15 ช่วงเวลา จำนวนทั้งหมด 10 ปัญหา สามารถแสดงผลการทดลองได้ดังตารางที่ ก.18

ตารางที่ ก.18 ผลการทดลองปัญหาตัวอย่างขนาดสินค้า 25 ชนิด 15 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	71,030.86	*	0.020	*	*
2	80,474.84	*	0.010	*	*
3	68,812.69	*	0.010	*	*
4	76,703.39	*	0.010	*	*
5	77,558.20	*	0.005	*	*
6	74,277.88	*	0.005	*	*
7	68,096.07	*	0.000	*	*
8	78,360.01	*	0.000	*	*
9	63,465.52	*	0.010	*	*
10	61,539.93	*	0.005	*	*
		เฉลี่ย	0.008	-	-

หมายเหตุ : * ไม่สามารถหาคำตอบได้เนื่องจากใช้เวลาคำนวณสูง

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้ทุกปัญหา ส่วนวิธีการคำนวณแบบ Branch and Bound ไม่สามารถคำนวณหาคำตอบได้เนื่องจากใช้เวลาในการคำนวณเฉลี่ยสูงมากกว่า 48 ชั่วโมงยังไม่สามารถคำนวณหาคำตอบได้

ก.5 ผลการทดลองของกลุ่มปัญหาเพื่อวัดผลของสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษา

ก.5.1 ผลการทดลองเมื่อสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเท่ากับ 100 เท่า สามารถแสดงผลการทดลองได้ดังตารางที่ ก.19

ตารางที่ ก.19 ผลการทดลองเมื่อสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเท่ากับ 100
เท่า ขนาดสินค้า 5 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	5,617.00	5,455.00	0.000	2.000	2.97
2	4,762.00	4,682.00	0.000	1.000	1.71
3	3,204.00	3,204.00	0.005	1.000	0.00
4	4,189.00	3,753.00	0.000	1.000	11.62
5	3,837.00	3,837.00	0.000	1.000	0.00
6	3,867.00	3,556.00	0.010	1.000	8.75
7	4,093.00	3,929.00	0.010	1.000	4.17
8	3,755.00	3,755.00	0.000	1.000	0.00
9	4,670.00	4,216.00	0.000	1.000	10.77
10	4,631.00	4,631.00	0.000	3.000	0.00
เฉลี่ย			0.003	1.300	4.00

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหา โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 4.00

ก.5.2 ผลการทดลองเมื่อสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเท่ากับ 125 เท่า สามารถแสดงผลการทดลองได้ดังตารางที่ ก.20

ตารางที่ ก.20 ผลการทดลองเมื่อสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเท่ากับ 125
เท่า ขนาดสินค้า 5 ชนิด 5 ช่วงเวลา

ลำดับ ที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	6,942.00	6,730.00	0.000	1.000	3.15
2	5,837.00	5,688.00	0.000	2.000	2.62
3	3,904.00	3,904.00	0.005	1.000	0.00
4	5,139.00	4,525.00	0.005	1.000	13.57
5	4,587.00	4,587.00	0.000	1.000	0.00
6	4,532.00	4,156.00	0.010	1.000	9.05
7	4,993.00	4,779.00	0.010	0.400	4.48
8	4,530.00	4,481.00	0.000	1.000	1.09
9	5,695.00	5,141.00	0.000	1.000	10.78
10	5,656.00	5,656.00	0.000	1.000	0.00
เฉลี่ย			0.003	1.040	4.47

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหา โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 4.47

ก.5.3 ผลการทดลองเมื่อสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษาเท่ากับ 150 เท่า สามารถแสดงผลการทดลองได้ดังตารางที่ ก.21

ตารางที่ ก.21 ผลการทดลองเมื่อสัดส่วนระหว่างคำสั่งซื้อหรือผลิตต่อค่าการเก็บรักษาเท่ากับ 150
เท่าขนาดสินค้า 5 ชนิด 5 ช่วงเวลา

ลำดับที่	Heuristic	Optimization	เวลา (CPU Sec.)		เปอร์เซ็นต์ ความแตกต่าง ของคำตอบ
			Heuristic	B&B	
1	8,267.00	8,005.00	0.005	2.000	3.27
2	6,912.00	6,688.00	0.000	1.000	3.35
3	4,604.00	4,492.00	0.005	1.000	2.49
4	6,089.00	5,275.00	0.000	1.000	15.43
5	5,337.00	5,337.00	0.000	1.000	0.00
6	5,232.00	4,756.00	0.005	1.000	10.01
7	5,893.00	5,629.00	0.010	1.000	4.69
8	5,305.00	5,206.00	0.005	1.000	1.90
9	6,720.00	6,066.00	0.000	1.000	10.78
10	6,681.00	6,681.00	0.005	3.000	0.00
		เฉลี่ย	0.004	1.300	5.19

จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถหาคำตอบได้และใช้เวลาในการคำนวณหาคำตอบน้อยกว่าวิธี Branch and Bound ในทุกปัญหา โดยมีเปอร์เซ็นต์ความแตกต่างของคำตอบเฉลี่ยเท่ากับ 5.19



ภาคผนวก ข

คู่มือการใช้งานโปรแกรมและโค้ดของโปรแกรม

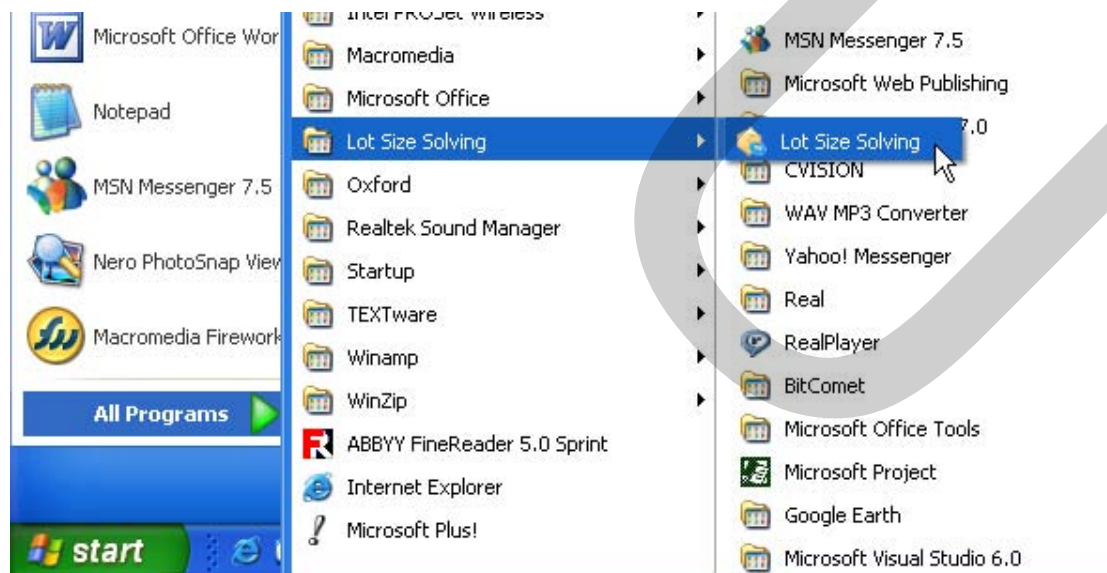
ผนวก ข

คู่มือการใช้งานโปรแกรมและโค้ดของโปรแกรม

โปรแกรมที่พัฒนาขึ้นมาเพื่อใช้ในการทดสอบวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตแบบประหยัคของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันตามวิธีการที่นำเสนอในงานวิจัยครั้งนี้ชื่อว่า โปรแกรม Lot Size Solving ซึ่งถูกพัฒนาขึ้นด้วยภาษา Visual Basic และ Visual C++ และถูกออกแบบให้ใช้งานได้ง่ายและสะดวก ซึ่งเป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows การใช้งาน โปรแกรมก็คล้ายๆกับโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows ทั่วไป

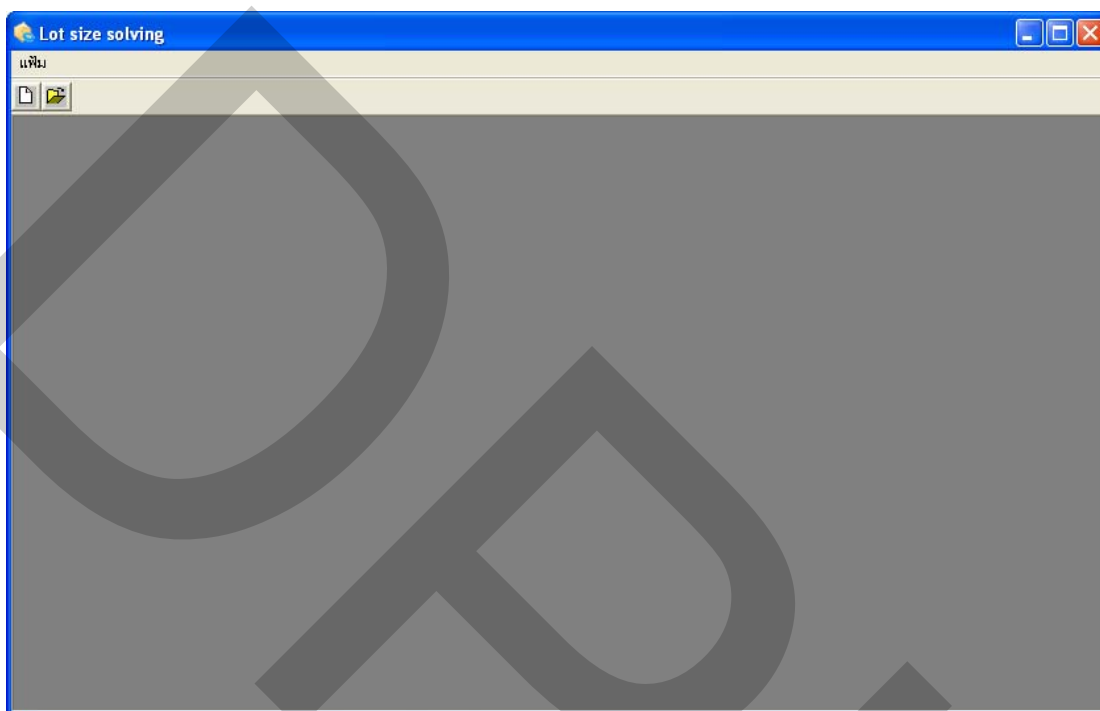
ข.1 การเรียกใช้งานโปรแกรม Lot Size Solving

การแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัคของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันด้วยวิธีการที่นำเสนอสามารถทำได้ด้วยการเรียกใช้งาน โปรแกรม Lot Size Solving โดยให้คลิกที่ **Start > All Programs > Lot Size Solving > Lot Size Solving** ดังภาพที่ ข.1




ภาพที่ ข.1 การเรียกใช้งานโปรแกรม

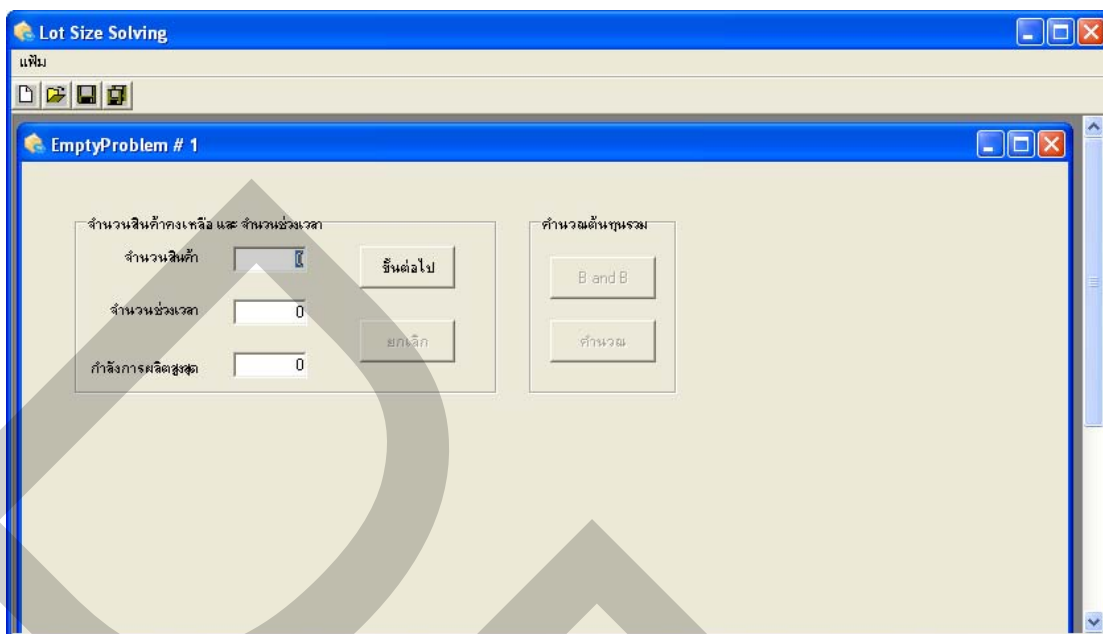
หลังจากคลิกที่ **Start > All Programs > Lot Size Solving > Lot Size Solving** จะปรากฏ หน้าต่างของโปรแกรม Lot Size Solving ดังภาพที่ ข.2



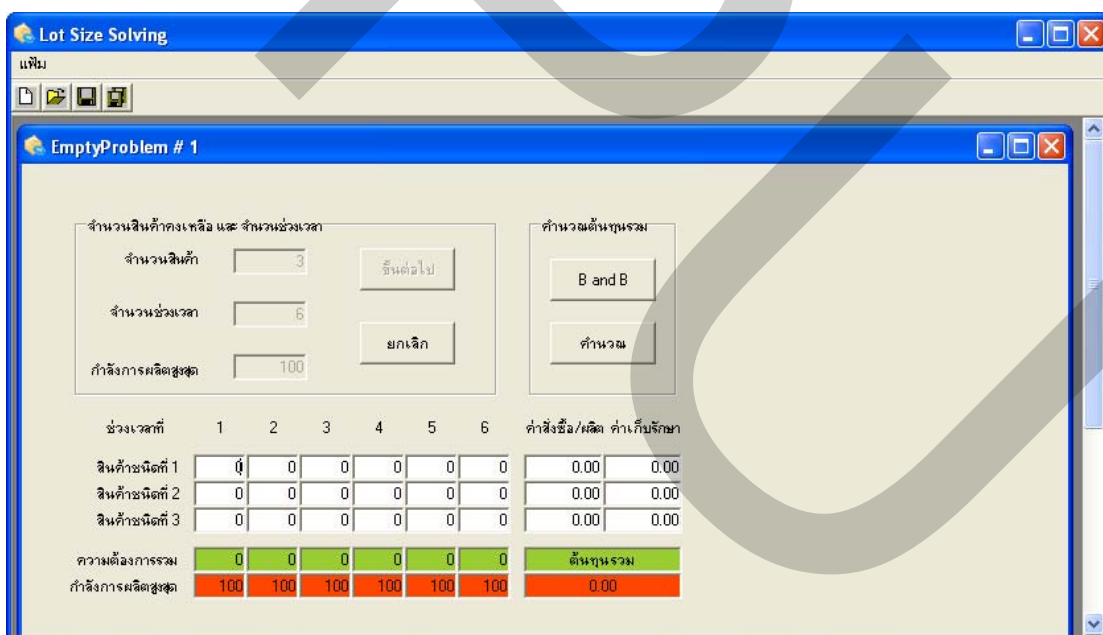
ภาพที่ ข.2 หน้าต่างโปรแกรม Lot Size Solving

ข.2 การสร้างหน้าต่างปัญหาใหม่และการคำนวณหาคำตอบ

การสร้างหน้าต่างปัญหาใหม่เพื่อแก้ปัญหาปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันด้วยวิธีการที่นำเสนอสามารถทำได้โดยคลิกที่ **แฟ้ม > สร้าง** หรือคลิกที่ไอคอน  จะปรากฏหน้าต่างปัญหาเปล่าดังภาพที่ ข.3 ในหน้าต่างปัญหาเปล่าที่ปรากฏขึ้นมานั้นจะมีช่องสำหรับระบุ จำนวนสินค้า จำนวนช่วงเวลาและกำลังการผลิตสูงสุดของปัญหาที่ต้องการคำนวณหาปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังที่เหมาะสมโดยประมาณ จะขอยกตัวอย่างโดยให้กรอกจำนวนสินค้าเท่ากับ 3 ชนิด จำนวนช่วงเวลาเท่ากับ 6 ช่วงเวลา และกำลังการผลิตสูงสุดเท่ากับ 100 หน่วย เมื่อกำหนดค่าต่างๆเรียบร้อยแล้วให้คลิกที่ปุ่ม **ขั้นต่อไป** จะปรากฏหน้าต่างสำหรับแก้ปัญหาของสินค้า 3 ชนิด 6 ช่วงเวลา ซึ่งจะมีช่องสำหรับกรอกข้อมูลความต้องการของสินค้า 3 ชนิด 6 ช่วงเวลา ค่าสั่งซื้อหรือผลิตและค่าเก็บรักษาของสินค้าแต่ละชนิดในลักษณะของเมตริกซ์ขนาด 3x8 ปรากฏขึ้นมาดังภาพที่ ข.4



ภาพที่ ข.3 หน้าต่างปัญหาเปล่า



ภาพที่ ข.4 หน้าต่างปัญหาเปล่าสำหรับแก้ปัญหขนาดสินค้า 3 ชนิด 6 ช่วงเวลา กำลังการผลิตสูงสุดเท่ากับ 100 หน่วย

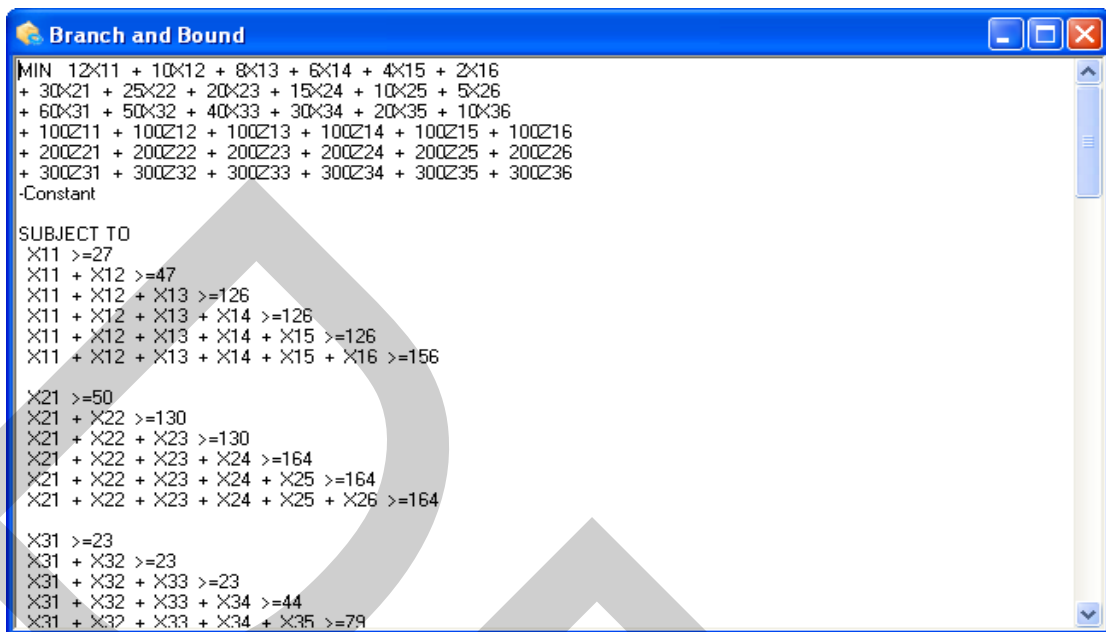
ให้ทำการกรอกค่าต่างๆของปัญหาเช่น ความต้องการของสินค้าแต่ละชนิดในช่วงเวลาต่างๆ ค่าสั่งซื้อหรือผลิตและค่าเก็บรักษาของสินค้าแต่ละชนิดตามที่ต้องการดังภาพที่ ข.5

ช่วงเวลา	1	2	3	4	5	6	ค่าสั่งซื้อ/ผลิต	ค่าเก็บรักษา
สินค้าชนิดที่ 1	27	20	40	27	12	30	100	2
สินค้าชนิดที่ 2	45	60	25	20	4	10	200	5
สินค้าชนิดที่ 3	21	0	2	21	35	41	300	10
ความต้องการรวม	93	80	67	68	51	81	คำนวณรวม	
ค่าสั่งการผลิตสูงสุด	100	100	100	100	100	100	3,300.00	

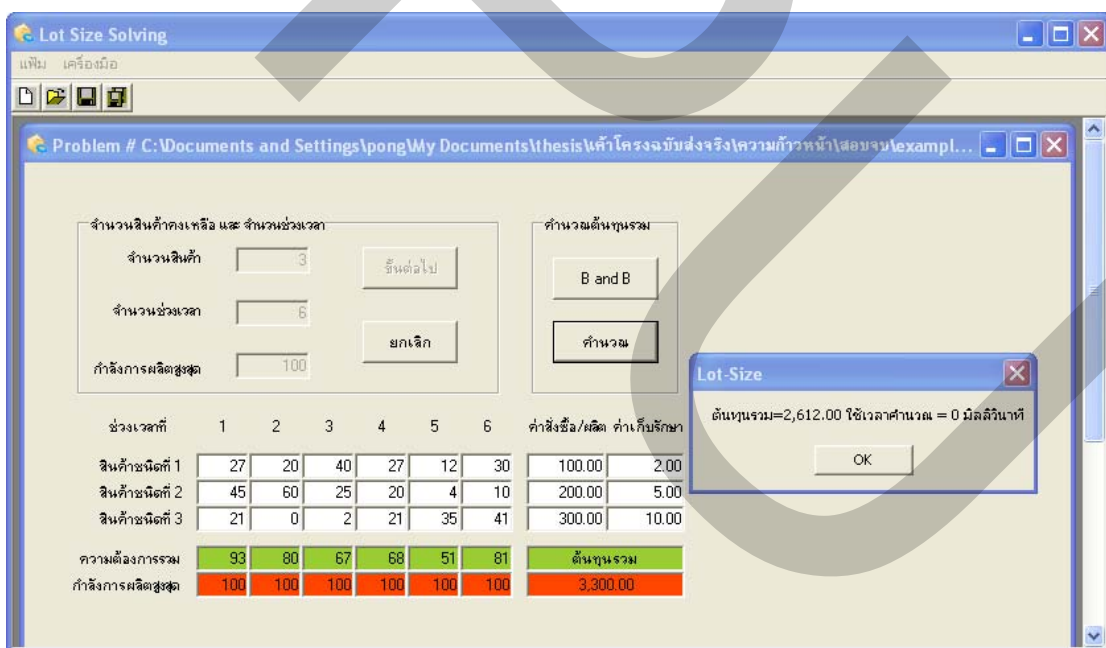
ภาพที่ ข.5 หน้าต่างปัญหาเปล่าที่กรอกข้อมูลสำหรับแก้ปัญหาเรียบร้อยแล้ว

จากภาพที่ ข.5 หากต้องการแก้ไขขนาดสินค้าและช่วงเวลาสามารถทำได้โดยคลิกที่ปุ่มยกเลิก จากนั้นก็แก้ไขขนาดสินค้าและช่วงเวลาตามที่ต้องการได้ สำหรับในงานวิจัยครั้งนี้ได้พัฒนาให้โปรแกรมสามารถแปลงรูปแบบของปัญหาให้อยู่ในรูปของปัญหาคณิตศาสตร์ที่สามารถนำไปแก้ปัญหาด้วยโปรแกรมสำเร็จรูปที่ใช้สำหรับหาค่าที่ดีที่สุดได้ทันทีที่โดยคลิกที่ปุ่ม **B & B** ซึ่งจะปรากฏหน้าต่างใหม่ที่มีสมการคณิตศาสตร์ให้สามารถคัดลอกไปวางในโปรแกรมสำเร็จรูปที่ใช้สำหรับหาค่าที่ดีที่สุดได้ทันที เพื่อนำคำตอบที่ดีที่สุดมาเปรียบเทียบกับคำตอบที่ได้จากโปรแกรม Lot Size Solving ดังภาพที่ ข.6

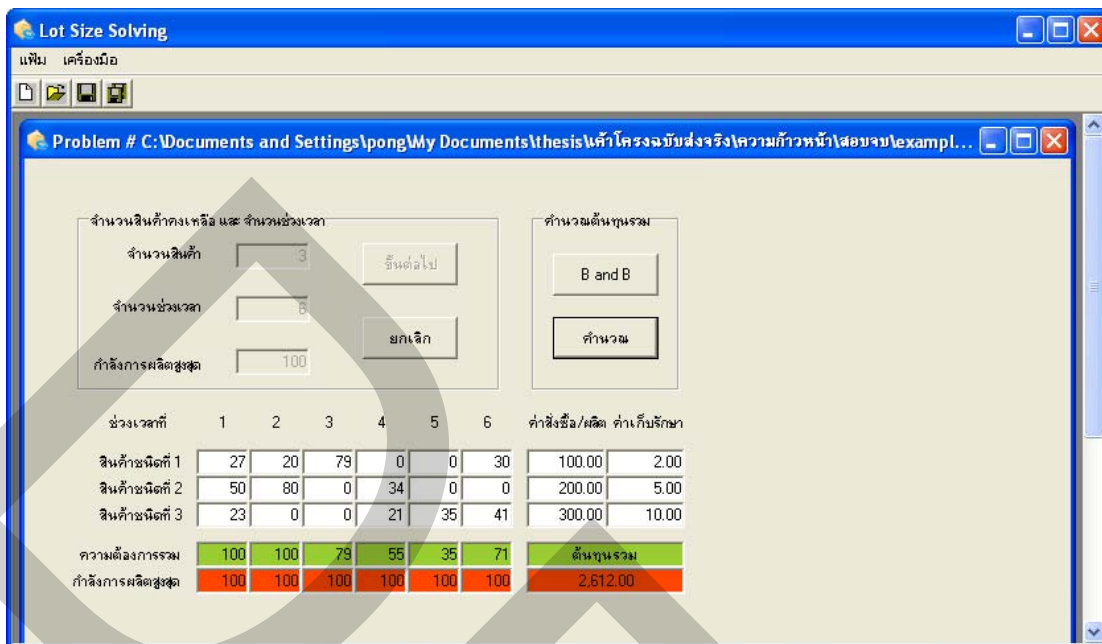
เมื่อมาถึงขั้นตอนนี้สามารถที่จะคำนวณหาคำตอบที่เหมาะสมโดยประมาณได้โดยคลิกที่ปุ่ม **คำนวณ** โปรแกรมจะทำการคำนวณหาคำตอบและเมื่อ โปรแกรมคำนวณเสร็จเรียบร้อยแล้วจะปรากฏหน้าต่างแสดงต้นทุนรวมและเวลาที่ใช้ในการคำนวณดังภาพที่ ข.7 จากภาพที่ ข.7 ให้คลิกที่ปุ่ม **OK** โปรแกรมจะทำการแสดงผลปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมโดยประมาณของสินค้าในแต่ละช่วงเวลาและต้นทุนรวมตามปริมาณการสั่งซื้อหรือผลิตใหม่ที่ได้จากโปรแกรมดังภาพที่ ข.8



ภาพที่ ข.6 หน้าต่างแสดงสมการคณิตศาสตร์

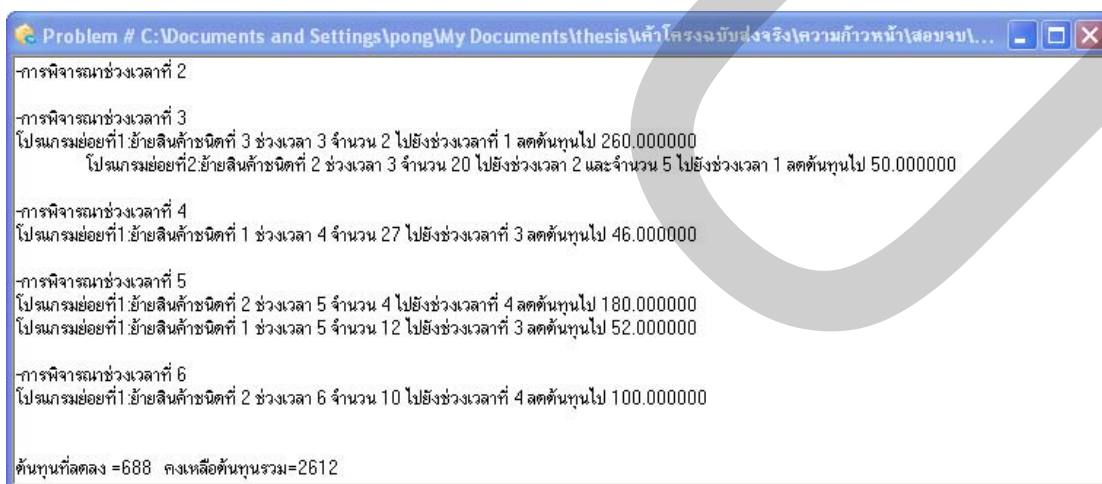


ภาพที่ ข.7 หน้าต่างแสดงต้นทุนรวมและเวลาที่ใช้ในการคำนวณ




ภาพที่ ข.8 ปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมโดยประมาณของสินค้าในแต่ละช่วงเวลาและต้นทุนรวม

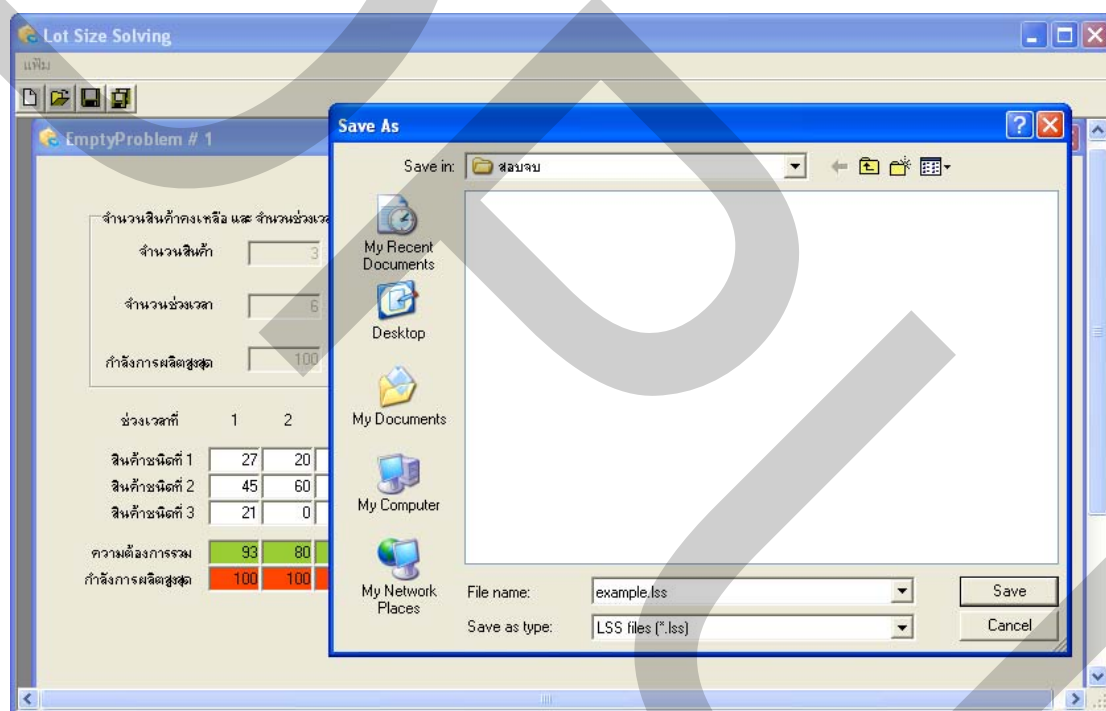
และโปรแกรมจะสร้างหน้าต่างรายงานว่าทำการย้ายหรือแบ่งย้ายความต้องการของสินค้าชนิดใด ช่วงเวลาใด เป็นจำนวนกี่หน่วยและสามารถลดต้นทุนได้เท่าไรดังภาพ ข.9



ภาพที่ ข.9 หน้าต่างรายงานแสดงผลการคำนวณ


ข.3 การบันทึกและบันทึกเป็นของโปรแกรม

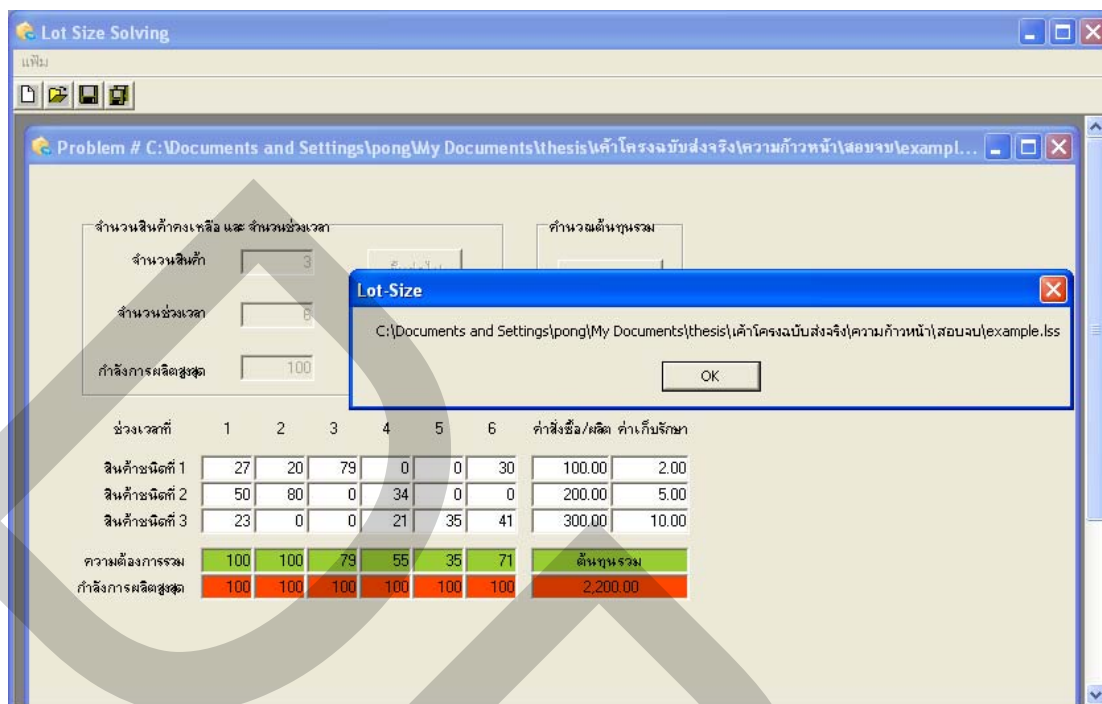
โปรแกรมสามารถเก็บบันทึกปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตแบบประหัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันที่ได้กรอกข้อมูลต่างๆไว้เรียบร้อยแล้วได้โดยคลิกที่ **แฟ้ม > บันทึก** หรือคลิกที่ไอคอน  จะปรากฏหน้าต่างดังภาพที่ ข.10 ซึ่งสามารถที่จะเลือกสถานที่จัดเก็บไฟล์ได้จากช่อง Save in และสามารถกรอกชื่อไฟล์ที่ต้องการได้ในช่อง File name โดยนามสกุลของโปรแกรม Lot Size Solving จะเป็น .LSS เมื่อกรอกชื่อไฟล์และเลือกสถานที่จัดเก็บไฟล์เรียบร้อยแล้วก็คลิกที่ปุ่ม **Save** โปรแกรมจะทำการบันทึกปัญหาไว้ในสถานที่ที่เลือกไว้ในช่อง Save in หากต้องการยกเลิกการบันทึกให้คลิกที่ปุ่ม **Cancel**




ภาพที่ ข.10 หน้าต่างสำหรับบันทึกปัญหา

หลังจากบันทึกข้อมูลเรียบร้อยแล้วหากมีการแก้ไขข้อมูลความต้องการสินค้า ค่าการสั่งซื้อหรือผลิตหรือค่าเก็บรักษาและหากต้องการบันทึกสิ่งที่ได้แก้ไขไปสามารถทำได้ 2 แบบคือ

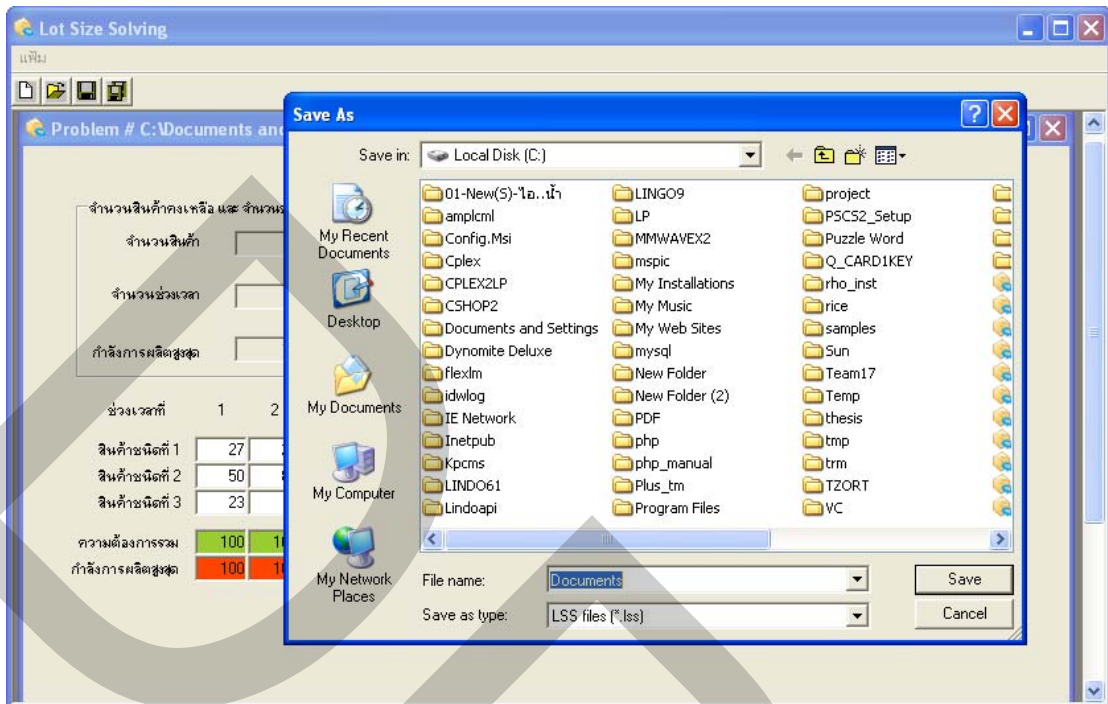
ข.3.1 การบันทึกไฟล์ในชื่อและสถานที่จัดเก็บเดิม การบันทึกในลักษณะนี้เป็นการบันทึกทับไฟล์เดิมนั้นเองสามารถทำได้โดยคลิกที่ **แฟ้ม > บันทึก** หรือคลิกที่ไอคอน  โปรแกรมจะบันทึกข้อมูลใหม่ทันทีและจะมีหน้าต่างแจ้งว่าได้บันทึกไฟล์นั้นไว้ที่ตำแหน่งใดในเครื่องคอมพิวเตอร์ดังภาพที่ ข.11 หลังจากนั้นให้คลิกที่ **OK** เพื่อปิดหน้าต่างการแจ้งเตือนดังกล่าวทิ้งไป



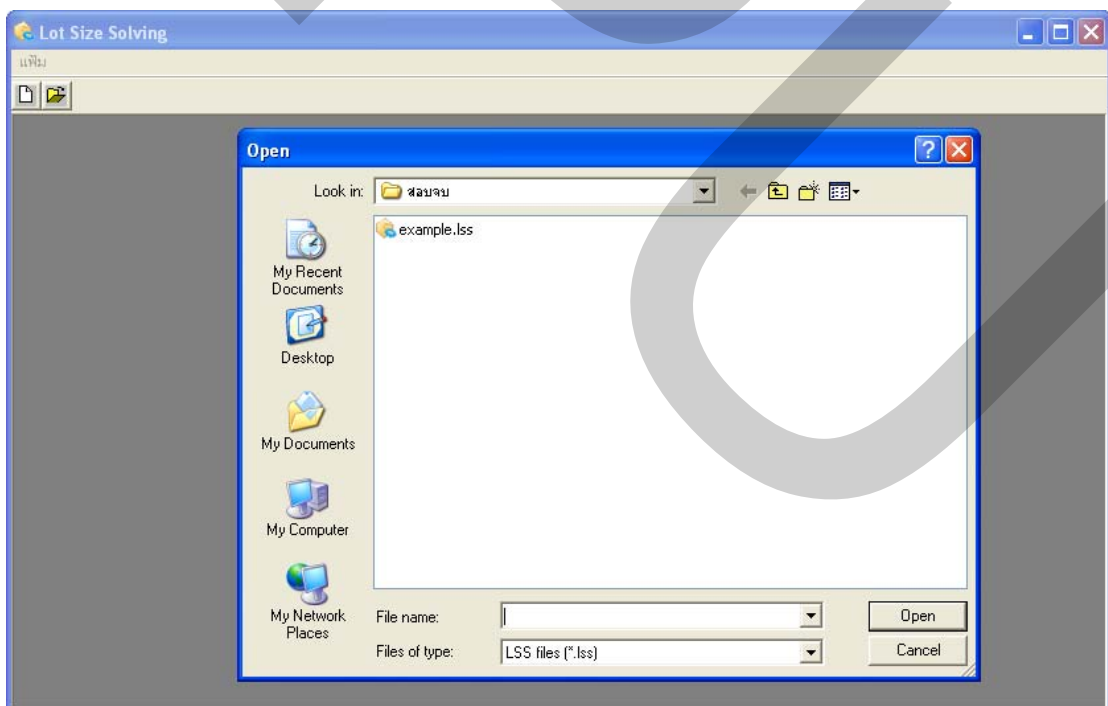
ภาพที่ ข.11 หน้าต่างแสดงการบันทึกปัญหา

ข.3.2 การบันทึกไฟล์เป็นชื่อใหม่ในสถานที่จัดเก็บเดิมหรือสถานที่จัดเก็บใหม่ การบันทึกในลักษณะนี้เป็นการบันทึกไฟล์ในชื่อใหม่ในสถานที่จัดเก็บเดิมหรือสถานที่จัดเก็บใหม่ก็ได้ สามารถทำได้คลิกที่ **แฟ้ม > บันทึกเป็น** หรือคลิกที่ไอคอน  จะปรากฏหน้าต่างดังภาพที่ ข.12 ซึ่งสามารถเลือกสถานที่จัดเก็บไฟล์ได้ใหม่จากช่อง Save in และสามารถเปลี่ยนแปลงชื่อไฟล์ได้ใหม่ในช่อง File name เมื่อเปลี่ยนแปลงชื่อไฟล์และเลือกสถานที่จัดเก็บไฟล์ใหม่เรียบร้อยแล้วก็คลิกที่ปุ่ม **Save** โปรแกรมจะทำการบันทึกปัญหาไว้ในชื่อใหม่ในสถานที่จัดเก็บเดิมหรือสถานที่จัดเก็บใหม่ตามทีเลือกไว้ในช่อง Save in ทั้งนี้ หากต้องการยกเลิกการบันทึกเป็นให้คลิกที่ปุ่ม **Cancel**

ข้อควรระวัง เมื่อทำการคำนวณต้นทุนรวมเรียบร้อยแล้วห้ามคลิกเพื่อทำการบันทึกไฟล์เพราะโปรแกรมจะทำการบันทึกผลของการคำนวณทับไฟล์ปัญหาโจทย์ทันที ทำให้ไม่สามารถทราบได้ว่าปัญหาโจทย์เดิมเป็นอย่างไร





ภาพที่ ข.12 หน้าต่างแสดงการบันทึกเป็นไฟล์ชื่อใหม่




ภาพที่ ข.13 หน้าต่างแสดงการเปิดไฟล์

ข.4 การออกจากปัญหาปัจจุบันและการเปิดปัญหาที่บันทึกไว้ภายในเครื่องคอมพิวเตอร์

เมื่อทำการคำนวณต้นทุนรวมเรียบร้อยแล้วและต้องการปิดจอปัญหาปัจจุบันสามารถทำได้โดยคลิกที่ **แฟ้ม > ปิด** หรือคลิกที่ไอคอน  ในกรอบหน้าต่างด้านในของโปรแกรม โปรแกรมจะทำการปิดหน้าต่างปัญหาปัจจุบันทันที และเมื่อปิดปัญหาเรียบร้อยแล้วต้องการเปิดปัญหาที่บันทึกไว้ในเครื่องสามารถทำได้โดยคลิกที่ **แฟ้ม > เปิด** หรือคลิกที่ไอคอน  จะปรากฏหน้าต่างสำหรับเลือกสถานที่เก็บไฟล์และไฟล์ดังกล่าวที่ ค.13 สามารถเลือกสถานที่จัดเก็บไฟล์เพื่อหาไฟล์ที่ต้องการเปิดได้จากช่อง Look in เมื่อเลือกสถานที่จัดเก็บไฟล์ได้แล้วก็ให้คลิกเลือกที่ชื่อไฟล์ที่ต้องการเปิด หลังจากนั้นจะปรากฏชื่อไฟล์ที่ได้คลิกเลือกในช่อง File name ต่อไปให้คลิกที่ปุ่ม **Open** เพื่อเปิดปัญหาที่ได้เลือกไว้ เท่านั้นโปรแกรมก็จะเปิดไฟล์ที่ต้องการให้ทันที

ข.5 การออกจากโปรแกรม Lot Size Solving

หลังจากที่จบการใช้งานโปรแกรม Lot Size Solving แล้วผู้ใช้งานสามารถออกจากโปรแกรมได้โดยคลิกที่ **แฟ้ม > จบการทำงาน** หรือคลิกที่ไอคอน  ที่กรอบหน้าต่างด้านนอกสุด โปรแกรมจะปิดตัวเองลงทันที

โปรแกรม Lot Size Solving สามารถใช้งานได้ง่ายและไม่มีอะไรยุ่งยากมากนัก หากคุ้นเคยกับการใช้งานโปรแกรมบนระบบปฏิบัติการ Windows อยู่แล้วก็ไม่จำเป็นต้องอ่านคู่มือนี้เลยก็ได้ แต่ถึงอย่างไรก็ตามสำหรับผู้ใช้งานที่ไม่คุ้นเคยกับการใช้งานโปรแกรมบนระบบปฏิบัติการ Windows มาก่อนก็สามารถใช้คู่มือนี้เป็นแนวทางในการใช้งานโปรแกรมได้ไม่ยากนัก

ข.6 โค้ดของโปรแกรม Lot Size Solving

โปรแกรม Lot Size Solving ถูกพัฒนาขึ้นจากภาษา Visual Basic และ Visual C++ โดยที่โปรแกรมในส่วนที่ใช้สำหรับติดต่อกับผู้ใช้งานถูกพัฒนาขึ้นจากโปรแกรม Visual Basic ส่วนโปรแกรมที่ใช้สำหรับคำนวณหาคำตอบที่เหมาะสมโดยประมาณของวิธีการแก้ปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตแบบประหยัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันตามวิธีการที่นำเสนอถูกพัฒนาด้วย Visual C++ เนื่องจาก Visual C++ เป็นที่ยอมรับกันโดยทั่วไปว่าสามารถทำงานได้เร็ว ดังนั้นโปรแกรม Lot Size Solving จึงประกอบไปด้วยสองส่วนด้วยกันคือ

ข.6.1 โปรแกรมสำหรับติดต่อกับผู้ใช้งานซึ่งพัฒนาจาก Visual Basic ประกอบไปด้วย 4 ฟอรัมได้แก่

ข.6.1.1 ฟอรัมหลักของโปรแกรม Lot Size Solving ฟอรัมหลักนี้มีหน้าที่ควบคุมการทำงานต่างๆของโปรแกรมเช่น การสร้างฟอรัมลูกเพื่อใช้ในการป้อนข้อมูลปัญหา การสั่งบันทึกปัญหา เป็นต้นมีโค้ดโปรแกรมดังนี้

```
Imports System.IO
Imports Microsoft.VisualBasic
Public Class Form1
    Inherits System.Windows.Forms.Form
    Public Shared gFormCounter = 0
    #Region " Windows Form Designer generated code "
    Public Sub New()
        MyBase.New()
        'This call is required by the Windows Form Designer.
        InitializeComponent()
        'Add any initialization after the InitializeComponent() call
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
```

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem4 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem5 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem6 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem7 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem8 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem9 As System.Windows.Forms.MenuItem

Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu

Friend WithEvents MenuItem3 As System.Windows.Forms.MenuItem

Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem

Friend WithEvents ToolBar1 As System.Windows.Forms.ToolBar

Friend WithEvents MenuImages As System.Windows.Forms.ImageList

Friend WithEvents ToolBarButton1 As System.Windows.Forms.ToolBarButton

Friend WithEvents ToolBarButton2 As System.Windows.Forms.ToolBarButton

Friend WithEvents ToolBarButton3 As System.Windows.Forms.ToolBarButton

Friend WithEvents ToolBarButton4 As System.Windows.Forms.ToolBarButton

<System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()

 Me.components = New System.ComponentModel.Container

 Dim resources As System.Resources.ResourceManager = New

System.Resources.ResourceManager(GetType(Form1))

 Me.MainMenu1 = New System.Windows.Forms.MainMenu

 Me.MenuItem1 = New System.Windows.Forms.MenuItem

 Me.MenuItem2 = New System.Windows.Forms.MenuItem

 Me.MenuItem7 = New System.Windows.Forms.MenuItem

 Me.MenuItem5 = New System.Windows.Forms.MenuItem

 Me.MenuItem9 = New System.Windows.Forms.MenuItem


```

Me.MenuItem3 = New System.Windows.Forms.MenuItem
Me.MenuItem4 = New System.Windows.Forms.MenuItem
Me.MenuItem8 = New System.Windows.Forms.MenuItem
Me.MenuItem6 = New System.Windows.Forms.MenuItem
Me.ToolBar1 = New System.Windows.Forms.ToolBar
Me.ToolBarButton1 = New System.Windows.Forms.ToolBarButton
Me.ToolBarButton2 = New System.Windows.Forms.ToolBarButton
Me.ToolBarButton3 = New System.Windows.Forms.ToolBarButton
Me.ToolBarButton4 = New System.Windows.Forms.ToolBarButton
Me.MenuImages = New System.Windows.Forms.ImageList(Me.components)
Me.SuspendLayout()
'
'MainMenu1
'
Me.MainMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem1})
'
'MenuItem1
'
Me.MenuItem1.Index = 0
Me.MenuItem1.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem2, Me.MenuItem7, Me.MenuItem5, Me.MenuItem9, Me.MenuItem3,
Me.MenuItem4, Me.MenuItem8, Me.MenuItem6})
Me.MenuItem1.Text = "แ&พีจ"
'
'MenuItem2
'
Me.MenuItem2.Index = 0
Me.MenuItem2.ShowShortcut = False
Me.MenuItem2.Text = "&ศั๓๓..."

```

```
' MenuItem7
,

Me.MenuItem7.Index = 1
Me.MenuItem7.Text = "&ปิด..."
,

' MenuItem5
,

Me.MenuItem5.Enabled = False
Me.MenuItem5.Index = 2
Me.MenuItem5.Text = "ปิด"
,

' MenuItem9
,

Me.MenuItem9.Index = 3
Me.MenuItem9.Text = "-"
,

' MenuItem3
,

Me.MenuItem3.Enabled = False
Me.MenuItem3.Index = 4
Me.MenuItem3.Text = "&บันทึก"
,

' MenuItem4
,

Me.MenuItem4.Enabled = False
Me.MenuItem4.Index = 5
Me.MenuItem4.Text = "บันทึกเป็น..."
,

' MenuItem8
,
```

```
Me.MenuItem8.Index = 6
Me.MenuItem8.Text = "-"
,
'MenuItem6
,
Me.MenuItem6.Index = 7
Me.MenuItem6.Text = "&จบการทำงาน"
,
'ToolBar1
,
Me.ToolBar1.Buttons.AddRange(New System.Windows.Forms.ToolBarButton()
{Me.ToolBarButton1, Me.ToolBarButton2, Me.ToolBarButton3, Me.ToolBarButton4})
Me.ToolBar1.ButtonSize = New System.Drawing.Size(22, 24)
Me.ToolBar1.DropDownArrows = True
Me.ToolBar1.ImageList = Me.MenuImages
Me.ToolBar1.Location = New System.Drawing.Point(0, 0)
Me.ToolBar1.Name = "ToolBar1"
Me.ToolBar1.ShowToolTips = True
Me.ToolBar1.Size = New System.Drawing.Size(830, 30)
Me.ToolBar1.TabIndex = 1
Me.ToolBar1.TabStop = True
Me.ToolBar1.Wrappable = False
,
'ToolBarButton1
,
Me.ToolBarButton1.ImageIndex = 0
Me.ToolBarButton1.ToolTipText = "สร้าง"
,
'ToolBarButton2
,
```

```

Me.ToolBarButton2.ImageIndex = 1
Me.ToolBarButton2.ToolTipText = "เปิด"
,
'ToolBarButton3
,
Me.ToolBarButton3.ImageIndex = 2
Me.ToolBarButton3.ToolTipText = "บันทึก"
,
'ToolBarButton4
,
Me.ToolBarButton4.ImageIndex = 3
Me.ToolBarButton4.ToolTipText = "บันทึกเป็น"
,
'MenuImages
,
Me.MenuImages.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit
Me.MenuImages.ImageSize = New System.Drawing.Size(16, 16)
Me.MenuImages.ImageStream = CType(resources.GetObject("MenuImages.ImageStream"),
System.Windows.Forms.ImageListStreamer)
Me.MenuImages.TransparentColor = System.Drawing.Color.Transparent
,
'Form1
,
Me.AllowDrop = True
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(830, 479)
Me.Controls.Add(Me.ToolBar1)
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
Me.ImeMode = System.Windows.Forms.ImeMode.AlphaFull
Me.IsMdiContainer = True

```

```
Me.Menu = Me.MainMenu1
Me.Name = "Form1"
Me.Text = "Lot Size Solving"
Me.ResumeLayout(False)
End Sub

#End Region

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    gFormCounter = 0
    If (Lot_Size.Form1.gFormCounter <= 0) Then
        MenuItem3.Enabled = False
        MenuItem4.Enabled = False
        MenuItem5.Enabled = False
        Me.ToolBarButton3.Visible = False
        Me.ToolBarButton4.Visible = False
    End If
End Sub

Private Sub MenuItemNew_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem2.Click
    gFormCounter += 1
    MenuItem3.Enabled = True
    MenuItem4.Enabled = True
    MenuItem5.Enabled = True
    Me.ToolBarButton3.Visible = True
    Me.ToolBarButton4.Visible = True
    Dim ChildForm As New LotChildForm
    ChildForm.MdiParent = Me
    ChildForm.Text = "EmptyProblem # " & gFormCounter.ToString
```

```
ChildForm.Show()
End Sub

Private Sub MenuItemClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem5.Click
    Lot_Size.LotChildForm.ActiveForm.ActiveMdiChild.Close()
End Sub

Private Sub MenuItemExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem6.Click
    Me.Close()
End Sub

Private Sub MenuItemOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem7.Click
    gFormCounter += 1
    MenuItem3.Enabled = True
    MenuItem4.Enabled = True
    MenuItem5.Enabled = True

    Dim myFile As String
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim nRowGlobal As Integer
    Dim nColGlobal As Integer
    Dim maxCapGlobal As Integer

    Dim openFileDialog1 As New OpenFileDialog
    openFileDialog1.InitialDirectory = "c:\Program Files\dpu\lotsizing\samples"
    openFileDialog1.Filter = "LSS files (*.lss)|*.lss"
```

```
openFileDialog1.FilterIndex = 2
openFileDialog1.RestoreDirectory = True

If openFileDialog1.ShowDialog() = DialogResult.OK Then
    'myStream = openFileDialog1.OpenFile()
    Dim sb As New FileStream(openFileDialog1.FileName, FileMode.Open)
    Dim sw As New StreamReader(sb)
    Dim ReadCost As Integer
    Dim ChildForm As New LotChildForm
    ChildForm.MdiParent = Me
    myFile = openFileDialog1.FileName
    'MsgBox("Read Vale" & myFile)
    ChildForm.Text = "Problem # " & myFile
    ChildForm.Show()
    Dim activeMdiChild As LotChildForm
    activeMdiChild = Me.ActiveMdiChild

    If Not (sw Is Nothing) Then
        ' Insert code to read the stream here.
        nRowGlobal = sw.ReadLine
        activeMdiChild.Text1.Text = nRowGlobal
        nColGlobal = sw.ReadLine
        activeMdiChild.Text2.Text = nColGlobal
        maxCapGlobal = sw.ReadLine
        activeMdiChild.Text3.Text = maxCapGlobal
        ChildForm.Button1.PerformClick()
        k = 1
        For i = 0 To nRowGlobal - 1
            For j = 0 To nColGlobal - 1
                Dim loadNum As Integer
```

```
        loadNum = sw.ReadLine
        activeMdiChild.Demand(k).Text = loadNum
        k = k + 1
    Next
Next
k = 0
For i = 0 To nRowGlobal - 1
    Dim loadNum As Single
    loadNum = sw.ReadLine
    activeMdiChild.ProductCost(k).Text = Format(loadNum, "##,##0.00")
    k = k + 1
Next
k = 0
For i = 0 To nRowGlobal - 1
    Dim loadNum As Single
    loadNum = sw.ReadLine
    activeMdiChild.HoldingCost(k).Text = Format(loadNum, "##,##0.00")
    k = k + 1
Next
ReadCost = sw.ReadLine()
activeMdiChild.ToTalCost.Text = Format(ReadCost, "##,##0.00")
sw.Close()
End If
Me.ToolBarButton3.Visible = True
Me.ToolBarButton4.Visible = True
activeMdiChild.cal_Total_Cost()
Else
    gFormCounter -= 1
    If (gFormCounter = 0) Then
        MenuItem3.Enabled = False
```



```
MenuItem4.Enabled = False
MenuItem5.Enabled = False
End If
End If
End Sub

Private Sub MenuItemSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem3.Click
    Dim myFile As String
    Dim subString As String
    Dim i As Integer
    Dim j As Integer
    Dim activeMdiChild As LotChildForm
    activeMdiChild = Me.ActiveMdiChild
    myFile = activeMdiChild.Text.ToString
    Dim myArray() As String = Split(myFile)
    myFile = myFile.Remove(0, 9)
    If (myArray(0) = "EmptyProblem") Then
        Me.MenuItem4.PerformClick()
    Else
        Dim nRowGlobal As Integer
        Dim nColGlobal As Integer
        Dim maxCapGlobal As Integer
        nRowGlobal = activeMdiChild.Text1.Text
        nColGlobal = activeMdiChild.Text2.Text
        maxCapGlobal = activeMdiChild.Text3.Text
        MsgBox(myFile)
        Dim sb As New FileStream(myFile, FileMode.OpenOrCreate, FileAccess.Write)
        Dim sw As New StreamWriter(sb)
        If Not (sw Is Nothing) Then
```

```

sw.WriteLine(nRowGlobal)
sw.WriteLine(nColGlobal)
sw.WriteLine(maxCapGlobal)
For i = 0 To nRowGlobal - 1
    For j = 0 To nColGlobal - 1
        Dim nValue As Integer
        nValue = activeMdiChild.DemandArr(i, j)
        sw.WriteLine(nValue)
    Next
Next
For i = 0 To nRowGlobal - 1
    Dim pValue As Double
    pValue = activeMdiChild.ProductCostArr(i)
    sw.WriteLine(pValue)
Next
For i = 0 To nRowGlobal - 1
    Dim pValue As Double
    pValue = activeMdiChild.HoldingCostArr(i)
    sw.WriteLine(pValue)
Next
Dim totalValue As Integer
totalValue = activeMdiChild.ToTalCost.Text
sw.WriteLine(totalValue)
sw.Close()
End If
End If
End Sub

Public Sub Menu_Enable(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Me.MenuItem3.Enabled = True

```

```
Me.MenuItem4.Enabled = True
Me.MenuItem5.Enabled = True
End Sub

Public Property FindMenuState() As Boolean
    Get
        FindMenuState = MenuItem3.Visible
    End Get
    Set(ByVal value As Boolean)
        MenuItem3.Visible = value
    End Set
End Property

Private Sub MenuItemSaveAs_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem4.Click
    Dim myFile As String
    Dim i As Integer
    Dim j As Integer
    Dim nRowGlobal As Integer
    Dim nColGlobal As Integer
    Dim maxCapGlobal As Integer
    Dim activeMdiChild As LotChildForm
    activeMdiChild = Me.ActiveMdiChild
    myFile = activeMdiChild.Text.ToString
    Dim myArray() As String = Split(myFile)
    Dim saveFileDialog1 As New SaveFileDialog
    nRowGlobal = activeMdiChild.Text1.Text
    nColGlobal = activeMdiChild.Text2.Text
    maxCapGlobal = activeMdiChild.Text3.Text
    saveFileDialog1.Filter = "LSS files (*.lss)*.lss"
```

```
saveFileDialog1.FilterIndex = 2
saveFileDialog1.RestoreDirectory = True
saveFileDialog1.FileName = myArray(2)
If saveFileDialog1.ShowDialog() = DialogResult.OK Then
    Dim sb As New FileStream(saveFileDialog1.FileName, FileMode.OpenOrCreate)
    Dim sw As New StreamWriter(sb)
    If Not (sw Is Nothing) Then
        sw.WriteLine(nRowGlobal)
        sw.WriteLine(nColGlobal)
        sw.WriteLine(maxCapGlobal)
        For i = 0 To nRowGlobal - 1
            For j = 0 To nColGlobal - 1
                Dim nValue As Integer
                nValue = activeMdiChild.DemandArr(i, j)
                sw.WriteLine(nValue)
            Next
        Next
        Next
        For i = 0 To nRowGlobal - 1
            Dim pValue As Double
            pValue = activeMdiChild.ProductCostArr(i)
            sw.WriteLine(pValue)
        Next
        For i = 0 To nRowGlobal - 1
            Dim pValue As Double
            pValue = activeMdiChild.HoldingCostArr(i)
            sw.WriteLine(pValue)
        Next
        Dim totalValue As Integer
        totalValue = activeMdiChild.ToTalCost.Text
        sw.WriteLine(totalValue)
```

```
        sw.Close()
    End If
    activeMdiChild.Text = "Problem # " & saveFileDialog1.FileName()
End If
End Sub

Private Sub MenuItem10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
End Sub

Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MenuItem1.Click
End Sub

Private Sub ToolBar1_ButtonClick_1(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar1.ButtonClick
    Select Case ToolBar1.Buttons.IndexOf(e.Button)      ' Evaluate the Button property
        Case 0
            Me.MenuItem2.PerformClick()
            ' Insert code to open the file.
        Case 1
            Me.MenuItem7.PerformClick()
            ' Insert code to save the file.
        Case 2
            Me.MenuItem3.PerformClick()
            ' Insert code to print the file.
        Case 3
            Me.MenuItem4.PerformClick()
            ' Insert code to print the file.
    End Select
End Select
```

```

End Sub

Private Sub MenuItem10_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    Dim activeMdiChild As LotChildForm
    activeMdiChild = Me.ActiveMdiChild
    activeMdiChild.Text3.Enabled = True
    activeMdiChild.Button2.Enabled = True

End Sub

End Class

```

ข.6.1.2 ฟอรัมลูกของโปรแกรม Lot Size Solving มีหน้าที่สำหรับรับข้อมูลต่างๆเช่น ความต้องการของสินค้าแต่ละชนิดตามช่วงเวลาต่างๆ ค่าตั้งซื้อหรือผลิตและค่าเก็บรักษาเป็นต้นมีโค้ดโปรแกรมดังนี้

```

Public Class LotChildForm
    Inherits System.Windows.Forms.Form
    #Region " Windows Form Designer generated code "
    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()
        'Add any initialization after the InitializeComponent() call
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
    End Sub

```

```
End If
MyBase.Dispose(disposing)
End Sub

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer
'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
Friend WithEvents Button2 As System.Windows.Forms.Button
Friend WithEvents Label2 As System.Windows.Forms.Label
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Button1 As System.Windows.Forms.Button
Friend WithEvents Text2 As System.Windows.Forms.TextBox
Friend WithEvents Text1 As System.Windows.Forms.TextBox
Friend WithEvents Label3 As System.Windows.Forms.Label
Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
Friend WithEvents Button3 As System.Windows.Forms.Button
Friend WithEvents Text3 As System.Windows.Forms.TextBox
Friend WithEvents Button4 As System.Windows.Forms.Button
<System.Diagnostics.DebuggerStepThrough> Private Sub InitializeComponent()
    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(LotChildForm))
    Me.GroupBox1 = New System.Windows.Forms.GroupBox
    Me.Text3 = New System.Windows.Forms.TextBox
    Me.Label3 = New System.Windows.Forms.Label
    Me.Button2 = New System.Windows.Forms.Button
    Me.Label2 = New System.Windows.Forms.Label
    Me.Label1 = New System.Windows.Forms.Label
```

```

Me.Button1 = New System.Windows.Forms.Button
Me.Text2 = New System.Windows.Forms.TextBox
Me.Text1 = New System.Windows.Forms.TextBox
Me.GroupBox2 = New System.Windows.Forms.GroupBox
Me.Button3 = New System.Windows.Forms.Button
Me.Button4 = New System.Windows.Forms.Button
Me.GroupBox1.SuspendLayout()
Me.GroupBox2.SuspendLayout()
Me.SuspendLayout()
,
'GroupBox1
,
Me.GroupBox1.Controls.Add(Me.Text3)
Me.GroupBox1.Controls.Add(Me.Label3)
Me.GroupBox1.Controls.Add(Me.Button2)
Me.GroupBox1.Controls.Add(Me.Label2)
Me.GroupBox1.Controls.Add(Me.Label1)
Me.GroupBox1.Controls.Add(Me.Button1)
Me.GroupBox1.Controls.Add(Me.Text2)
Me.GroupBox1.Controls.Add(Me.Text1)
Me.GroupBox1.Location = New System.Drawing.Point(40, 40)
Me.GroupBox1.Name = "GroupBox1"
Me.GroupBox1.Size = New System.Drawing.Size(320, 136)
Me.GroupBox1.TabIndex = 4
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "จำนวนสินค้าคงเหลือ และ จำนวนช่วงเวลา"
,
'Text3
,
Me.Text3.Location = New System.Drawing.Point(120, 104)

```



```
Me.Text3.Name = "Text3"
Me.Text3.Size = New System.Drawing.Size(56, 20)
Me.Text3.TabIndex = 2
Me.Text3.Text = "0"
Me.Text3.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
,
'Label3
,
Me.Label3.Location = New System.Drawing.Point(8, 104)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(88, 23)
Me.Label3.TabIndex = 6
Me.Label3.Text = "กำลังการผลิตสูงสุด"
Me.Label3.TextAlign = System.Drawing.ContentAlignment.MiddleRight
,
'Button2
,
Me.Button2.Enabled = False
Me.Button2.Location = New System.Drawing.Point(216, 80)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(72, 32)
Me.Button2.TabIndex = 4
Me.Button2.Text = "ยกเลิก"
,
'Label2
,
Me.Label2.Location = New System.Drawing.Point(24, 64)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(72, 16)
Me.Label2.TabIndex = 4
```

```
Me.Label2.Text = "จำนวนช่วงเวลา"
Me.Label2.TextAlign = System.Drawing.ContentAlignment.MiddleRight
,
'Label1
,
Me.Label1.Location = New System.Drawing.Point(24, 24)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(72, 16)
Me.Label1.TabIndex = 3
Me.Label1.Text = "จำนวนสินค้า"
Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleRight
,
'Button1
,
Me.Button1.Location = New System.Drawing.Point(216, 24)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(72, 32)
Me.Button1.TabIndex = 3
Me.Button1.Text = "จึ้นต่อไป"
,
'Text2
,
Me.Text2.Location = New System.Drawing.Point(120, 64)
Me.Text2.Name = "Text2"
Me.Text2.Size = New System.Drawing.Size(56, 20)
Me.Text2.TabIndex = 1
Me.Text2.Text = "0"
Me.Text2.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
,
'Text1
```

```
Me.Text1.Location = New System.Drawing.Point(120, 24)
Me.Text1.Name = "Text1"
Me.Text1.Size = New System.Drawing.Size(56, 20)
Me.Text1.TabIndex = 0
Me.Text1.Text = "0"
Me.Text1.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
'
'GroupBox2
'
Me.GroupBox2.Controls.Add(Me.Button3)
Me.GroupBox2.Controls.Add(Me.Button4)
Me.GroupBox2.Location = New System.Drawing.Point(384, 40)
Me.GroupBox2.Name = "GroupBox2"
Me.GroupBox2.Size = New System.Drawing.Size(112, 136)
Me.GroupBox2.TabIndex = 5
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "คำนวณต้นทุนรวม"
'
'Button3
'
Me.Button3.Enabled = False
Me.Button3.Location = New System.Drawing.Point(16, 80)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(80, 32)
Me.Button3.TabIndex = 0
Me.Button3.Text = "คำนวณ"
'
'Button4
'
Me.Button4.Enabled = False
```

```
Me.Button4.Location = New System.Drawing.Point(16, 32)
Me.Button4.Name = "Button4"
Me.Button4.Size = New System.Drawing.Size(80, 32)
Me.Button4.TabIndex = 6
Me.Button4.Text = "B and B"
'
'LotChildForm
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.AutoScroll = True
Me.ClientSize = New System.Drawing.Size(792, 566)
Me.Controls.Add(Me.GroupBox2)
Me.Controls.Add(Me.GroupBox1)
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
Me.Name = "LotChildForm"
Me.Text = "LotChildForm"
Me.GroupBox1.ResumeLayout(False)
Me.GroupBox2.ResumeLayout(False)
Me.ResumeLayout(False)
End Sub
#End Region
Public Demand() As System.Windows.Forms.TextBox
Public SumDemand() As System.Windows.Forms.TextBox
Dim Cap() As System.Windows.Forms.TextBox
Public ProductCost() As System.Windows.Forms.TextBox
Public HoldingCost() As System.Windows.Forms.TextBox
Dim LaCountTime() As System.Windows.Forms.Label
Dim LaProduct() As System.Windows.Forms.Label
Dim LaSumDemand As System.Windows.Forms.Label
Dim LaTime As System.Windows.Forms.Label
```

```
Dim LaMaxCap As System.Windows.Forms.Label
Dim LaProductCost As System.Windows.Forms.Label
Dim LaHoldingCost As System.Windows.Forms.Label
Public DemandArr(,) As Integer
Public DemandArrCon(,) As Integer
Public HoldingCostArr() As Double
Public ProductCostArr() As Double
Dim SumDemandArr() As Integer
Dim CapArr() As Integer
Dim ArrConstEnable As Integer
Dim ToTalCostText As System.Windows.Forms.TextBox
Public ToTalCost As System.Windows.Forms.TextBox
Dim CalButton As System.Windows.Forms.Button
Private Sub LotChildForm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ArrConstEnable = 0
End Sub
Private Sub LotChildForm_Close(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Closed
    Lot_Size.Form1.gFormCounter -= 1
    Dim ActiveParent As Form1 = Form1.ActiveForm
    If (Lot_Size.Form1.gFormCounter <= 0) Then
        ActiveParent.MenuItem3.Enabled = False
        ActiveParent.MenuItem4.Enabled = False
        ActiveParent.MenuItem5.Enabled = False
        ActiveParent.ToolBarButton3.Visible = False
        ActiveParent.ToolBarButton4.Visible = False
    End If
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button1.Click
```

```
Dim i As Short
```

```
Dim j As Short
```

```
Dim k As Single
```

```
Dim ArrNo As Short
```

```
' resize the array of
```

```
ArrNo = (Text1.Text * Text2.Text)
```

```
ReDim Demand(ArrNo)
```

```
ReDim SumDemand(Text2.Text)
```

```
ReDim Cap(Text2.Text)
```

```
ReDim ProductCost(Text1.Text)
```

```
ReDim ProductCostArr(Text1.Text)
```

```
ReDim SumDemandArr(Text2.Text)
```

```
ReDim HoldingCost(Text1.Text)
```

```
ReDim HoldingCostArr(Text1.Text)
```

```
ReDim CapArr(Text2.Text)
```

```
ReDim LaCountTime(Text2.Text)
```

```
ReDim LaProduct(Text1.Text)
```

```
ReDim DemandArr(Text1.Text - 1, Text2.Text - 1)
```

```
ReDim DemandArrCon(Text1.Text - 1, Text2.Text - 1)
```

```
If (Text1.Text = 0) Or (Text2.Text = 0) Then
```

```
MsgBox("กรุณาระบุจำนวนสินค้าและช่วงเวลา")
```

```
GoTo GetEndSub1
```

```
End If
```

```
'สร้างLabelค่าการผลิต
```

```
Me.LaProductCost = New System.Windows.Forms.Label
```

```
With LaProductCost
```

```

.Text = "ค่าตั้งชื่อ/ผลิต"
.Size = New Size(60, 20)
.Location = New Point(140 + (Text2.Text * 40), 190)
.TextAlign = ContentAlignment.MiddleLeft
End With
Me.Controls.Add(LaProductCost)
'สร้าง Label ค่าการเก็บรักษา
Me.LaHoldingCost = New System.Windows.Forms.Label
With LaHoldingCost
.Text = "ค่าเก็บรักษา"
.Size = New Size(100, 20)
.Location = New Point(160 + (Text2.Text * 40), 190)
.TextAlign = ContentAlignment.MiddleRight
End With
Me.Controls.Add(LaHoldingCost)
k = 0
For i = 1 To Text1.Text 'แถว

'สร้างLabelสินค้า
Me.LaProduct(i) = New System.Windows.Forms.Label
With LaProduct(i)
.Text = "สินค้าชนิดที่ " & i
.Size = New Size(100, 20)
.Location = New Point(20, 200 + (i * 20))
.TextAlign = ContentAlignment.MiddleRight
End With
Me.Controls.AddRange(LaProduct)

'สร้างอินพุตความต้องการในแต่ละช่วงเวลา-----
For j = 1 To Text2.Text

```

```

k = k + 1
Me.Demand(k) = New System.Windows.Forms.TextBox
With Demand(k)
    .Name = "TextBox1"
    .Text = 0
    .TextAlign = HorizontalAlignment.Right
    .Size = New Size(40, 20)
    .Location = New Point(90 + (j * 40), 200 + (i * 20))
    .Tag = k
    AddHandler .TextChanged, AddressOf Me.btnArrayChange
End With
Me.Controls.AddRange(Demand)
If j = Text2.Text Then

    'สร้าง TextBox ค่าการผลิต
    Me.ProductCost(i - 1) = New System.Windows.Forms.TextBox
    With ProductCost(i - 1)
        .Text = Format(0, "##,##0.00")
        .TextAlign = HorizontalAlignment.Right
        .Size = New Size(60, 20)
        .Location = New Point(140 + (Text2.Text * 40), 200 + (i * 20))
        .Tag = i - 1
        AddHandler .TextChanged, AddressOf Me.tbxProductionChanged
    End With
    Me.Controls.AddRange(ProductCost)

    'สร้าง TextBox ค่าการเก็บรักษา
    Me.HoldingCost(i - 1) = New System.Windows.Forms.TextBox
    With HoldingCost(i - 1)
        .Text = Format(0, "##,##0.00")

```



```

        .TextAlign = HorizontalAlignment.Right
        .Size = New Size(60, 20)
        .Location = New Point(200 + (Text2.Text * 40), 200 + (i * 20))
        .Tag = i - 1
        AddHandler .TextChanged, AddressOf Me.tbxHoldingChanged
    End With
    Me.Controls.AddRange(HoldingCost)
End If
Next
Next
For j = 1 To Text2.Text
    Me.SumDemand(j - 1) = New System.Windows.Forms.TextBox
    With SumDemand(j - 1)
        .Text = 0
        .BackColor = System.Drawing.Color.YellowGreen
        .TextAlign = HorizontalAlignment.Right
        .Size = New Size(40, 20)
        .ReadOnly = True
        .Location = New Point(90 + (j * 40), 210 + ((Text1.Text + 1) * 20))
        .Tag = j - 1
        AddHandler .TextChanged, AddressOf Me.tbxSumDemandChanged
    End With
    Me.Controls.AddRange(SumDemand)

```

'สร้าง Label ช่วงเวลา

```

Me.LaCountTime(j) = New System.Windows.Forms.Label
With LaCountTime(j)
    .Text = j
    .Size = New Size(40, 20)

```

```

        .Location = New Point(90 + (j * 40), 190)
        .TextAlign = ContentAlignment.MiddleCenter
    End With
    Me.Controls.AddRange(LaCountTime)

    'สร้าง TextBox ของ Max Capacity
    Me.Cap(j - 1) = New System.Windows.Forms.TextBox
    CapArr(j - 1) = Text3.Text
    With Cap(j - 1)
        .Text = Text3.Text
        .BackColor = System.Drawing.Color.OrangeRed
        .TextAlign = HorizontalAlignment.Right
        .ReadOnly = True
        .Size = New Size(40, 20)
        .Location = New Point(90 + (j * 40), 230 + ((Text1.Text + 1) * 20))
        .Tag = j - 1
    End With
    Me.Controls.AddRange(Cap)
Next

'สร้างLabelของความต้องการรวม
Me.LaSumDemand = New System.Windows.Forms.Label
With LaSumDemand
    .Text = "ความต้องการรวม"
    .Size = New Size(100, 20)
    .Location = New Point(20, 210 + ((Text1.Text + 1) * 20))
    .TextAlign = ContentAlignment.MiddleRight
End With
Me.Controls.Add(LaSumDemand)

```

'สร้างLabelบอกช่วงเวลา

```
Me.LaTime = New System.Windows.Forms.Label
```

```
With LaTime
```

```
.Text = "ช่วงเวลาที่"
```

```
.Size = New Size(100, 20)
```

```
.Location = New Point(10, 190)
```

```
.TextAlign = ContentAlignment.MiddleRight
```

```
End With
```

```
Me.Controls.Add(LaTime)
```

'สร้างLabelMax Cap

```
Me.LaMaxCap = New System.Windows.Forms.Label
```

```
With LaMaxCap
```

```
.Text = "กำลังการผลิตสูงสุด"
```

```
.Size = New Size(100, 20)
```

```
.Location = New Point(20, 230 + ((Text1.Text + 1) * 20))
```

```
.TextAlign = ContentAlignment.MiddleRight
```

```
End With
```

```
Me.Controls.Add(LaMaxCap)
```

'สร้าง input ต้นทุนรวม

```
Me.ToTalCostText = New System.Windows.Forms.TextBox
```

```
With ToTalCostText
```

```
.Text = "ต้นทุนรวม"
```

```
.BackColor = System.Drawing.Color.YellowGreen
```

```
.TextAlign = HorizontalAlignment.Center
```

```
.Size = New Size(120, 20)
```

```
.ReadOnly = True
```

```
.Location = New Point(140 + (Text2.Text * 40), 210 + ((Text1.Text + 1) * 20))
```

```
.Tag = 0
```

End With

Me.Controls.Add(ToTalCostText)

'สร้าง input ต้นทุนรวม

Me.ToTalCost = New System.Windows.Forms.TextBox

With ToTalCost

.Text = Format(0, "##,##0.00")

.BackColor = System.Drawing.Color.OrangeRed

.TextAlign = HorizontalAlignment.Center

.Size = New Size(120, 20)

.ReadOnly = True

.Location = New Point(140 + (Text2.Text * 40), 230 + ((Text1.Text + 1) * 20))

.Tag = 0

End With

Me.Controls.Add(ToTalCost)

Me.Button1.Enabled = False

Me.Text1.Enabled = False

Me.Text2.Enabled = False

Me.Text3.Enabled = False

Me.Button2.Enabled = True

Me.Button3.Enabled = True

Me.Button4.Enabled = True

For i = 0 To Text1.Text - 1

For j = 0 To Text2.Text - 1

DemandArr(i, j) = 0

DemandArrCon(i, j) = 0

Next

Next

GetEndSub1:

End Sub

```
Private Sub btnArrayChange(ByVal sender As System.Object, ByVal e As EventArgs)
```

```
    Dim i As Short
```

```
    Dim j As Short
```

```
    Dim k As Short
```

```
    Dim l As Integer
```

```
    Dim m As Integer
```

```
    If Not IsNumeric(sender.Text) And (sender.Text = "") Then MsgBox("ป้อนค่าเป็นเลข  
จำนวนเต็มเท่านั้น")
```

```
    If Not IsNumeric(sender.Text) Then sender.Text = 0
```

```
    If Not IsNumeric(sender.Text) Then GoTo GetEndNow
```

```
    k = sender.Tag
```

```
    For i = 0 To Text1.Text - 1
```

```
        For j = 0 To Text2.Text - 1
```

```
            k = k - 1
```

```
            If k = 0 Then
```

```
                DemandArr(i, j) = sender.Text
```

```
                If (ArrConstEnable = 0) Then DemandArrCon(i, j) = sender.Text
```

```
                GoTo GetIndex
```

```
            End If
```

```
        Next
```

```
    Next
```

```
GetIndex:
```

```
    SumDemand(j).Text = 0
```

```
    For i = 0 To Text1.Text - 1
```

```
        m = DemandArr(i, j)
```

```
        SumDemand(j).Text = m + SumDemand(j).Text
```

```
    Next
```

```
    Dim sdm As Integer
```

```

Dim cmax As Integer
sdm = SumDemand(j).Text
cmax = Cap(j).Text
If sender.Text = "" Then
    sender.Text = 0
End If
GetEndNow:
    ToTalCost.Text = Format(0, "##,##0.00")
    cal_Total_Cost()
End Sub

Public Declare Function cInputCost Lib "cCalCost.dll" Alias _
    "cInputCost" (ByRef DemandArr As Integer, ByRef DemandArrCon As Integer, ByVal count1
As Integer, ByVal count2 As Integer, ByRef ProductCostArr As Double, ByRef HoldingCostArr
As Double, ByRef SumDemandArr As Integer, ByRef CapArr As Integer) As Single

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As EventArgs) Handles
Button3.Click

    Dim lSum As Integer

    Dim result, searchNum As Integer

    Dim presult As Single

    Dim i As Short

    Dim j As Short

    Dim k As Integer

    Dim VBTotalCost As Single

    Dim nRow As Integer

    Dim nCol As Integer

    VBTotalCost = ToTalCost.Text

    ArrConstEnable = 1

    nRow = Text1.Text

    nCol = Text2.Text

```

```

For i = 0 To Text1.Text - 1
    For j = 0 To Text2.Text - 1
        lSum = DemandArrCon(i, j)
    Next
Next

Dim StartTime As Date = Now() 'เริ่มต้นจับเวลา
'เริ่มต้นเรียกโปรแกรมในหัวข้อ ข.6.2 มาคำนวณตามวิธีการที่นำเสนอ
presult = LotChildForm.cInputCost(DemandArr(0, 0), DemandArrCon(0, 0),
UBound(DemandArr, 1) + 1, UBound(DemandArr, 2) + 1, ProductCostArr(0),
HoldingCostArr(0), SumDemandArr(0), CapArr(0))

Dim RunLength As System.TimeSpan = Now.Subtract(StartTime) 'หาเวลาที่ใช้คำนวณ
Dim Millisecs As Integer = RunLength.Milliseconds
MsgBox("ต้นทุนรวม=" & Format(presult, "#,##0.00") & " ใช้เวลาคำนวณ = " & Millisecs &
" มิลลิวินาที")

k = 1
For i = 0 To Text1.Text - 1
    For j = 0 To Text2.Text - 1
        Me.Demand(k).Text = DemandArr(i, j)
        k = k + 1
    Next
Next

ToTalCost.Text = Format(presult, "#,##0.00")
ArrConstEnable = 0

'สร้าง Report Form
Dim activeReport As New ReportForm
activeReport.Text = Me.Text.ToString

```

```

activeReport.Show()

Dim richTextBoxReport As New RichTextBox
richTextBoxReport.Dock = DockStyle.Fill
FileOpen(1, "data.txt", OpenMode.Input) ' Open file.
Do While Not EOF(1) ' Loop until end of file.
    richTextBoxReport.Text += LineInput(1) ' Read line into variable.
    richTextBoxReport.Text += Chr(13)
Loop
richTextBoxReport.Text += Chr(13)
richTextBoxReport.Text += "ต้นทุนที่ลดลง =" & VBTotCost - presult & " คงเหลือต้นทุนรวม=" & presult
FileClose(1) ' Close file.
activeReport.Controls.Add(richTextBoxReport)
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Dim i As Short
    Dim j As Short
    Dim k As Short
    k = 0
    For i = 1 To Text1.Text
        Me.Controls.Remove(LaProduct(i))
        Me.Controls.Remove(ProductCost(i - 1))
        Me.Controls.Remove(HoldingCost(i - 1))
        For j = 1 To Text2.Text
            k = k + 1
            Me.Controls.Remove(Demand(k))
        If i = Text1.Text Then

```



```

        Me.Controls.Remove(SumDemand(j - 1))
        Me.Controls.Remove(Cap(j - 1))
        Me.Controls.Remove(LaCountTime(j))
    End If
Next
Next
Me.Controls.Remove(LaSumDemand)
Me.Controls.Remove(LaTime)
Me.Controls.Remove(LaMaxCap)
Me.Controls.Remove(LaProductCost)
Me.Controls.Remove(LaHoldingCost)
Me.Controls.Remove(ToTalCostText)
Me.Controls.Remove(ToTalCost)
Me.Controls.Remove(CalButton)
Me.Button1.Enabled = True
Me.Text1.Enabled = True
Me.Text2.Enabled = True
Me.Text3.Enabled = True
Me.Button2.Enabled = False
Me.Button3.Enabled = False
Me.Button4.Enabled = False
ArrConstEnable = 0
End Sub

```

```

Private Sub Text1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Text1.TextChanged
    If Not IsNumeric(Text1.Text) Then MsgBox("ป้อนค่าเป็นเลขจำนวนเต็มเท่านั้น")
    If Not IsNumeric(Text1.Text) Then sender.Text = 0
End Sub

```

```
Private Sub Text2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Text2.TextChanged
```

```
    If Not IsNumeric(Text2.Text) Then MsgBox("ป้อนค่าเป็นเลขจำนวนเต็มเท่านั้น")
```

```
    If Not IsNumeric(Text2.Text) Then sender.Text = 0
```

```
End Sub
```

```
Private Sub tbxHoldingChanged(ByVal sender As System.Object, ByVal e As EventArgs)
```

```
    HoldingCostArr(sender.Tag) = sender.Text
```

```
    cal_Total_Cost()
```

```
End Sub
```

```
Private Sub tbxProductionChanged(ByVal sender As System.Object, ByVal e As EventArgs)
```

```
    ProductCostArr(sender.Tag) = sender.Text
```

```
    cal_Total_Cost()
```

```
End Sub
```

```
Private Sub tbxSumDemandChanged(ByVal sender As System.Object, ByVal e As EventArgs)
```

```
    SumDemandArr(sender.Tag) = sender.Text
```

```
    cal_Total_Cost()
```

```
End Sub
```

```
Public Sub cal_Total_Cost()
```

```
    Dim i As Short
```

```
    Dim j As Short
```

```
    Dim vbTotalcost As Single
```

```
    vbTotalcost = 0
```

```
    For i = 0 To Text1.Text - 1
```

```
        For j = 0 To Text2.Text - 1
```

```
            If Not (DemandArr(i, j) = 0) Then
```

```
                vbTotalcost = vbTotalcost + ProductCost(i).Text
```

```

        End If
    Next
Next
ToTalCost.Text = Format(vbTotalcost, "#,##0.00")
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    Dim activeEquatin As New EquationForm
    activeEquatin.Show()
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim m As Double
    Dim n As Integer
    Dim p As Integer
    Dim W As Double
    Dim Constant As Double
    Dim richTextBox1 As New RichTextBox
    richTextBox1.Dock = DockStyle.Fill
    richTextBox1.Text = "MIN "

    'สร้างสมการเพื่อนำไปหาคำตอบที่ดีที่สุด *****
    Constant = 0
    p = Text2.Text
    For i = 1 To Text1.Text
        W = 0
        k = 0
        For j = 1 To Text2.Text
            richTextBox1.Text += " " & p * HoldingCostArr(i - 1) & "X" & i & j & " "
        
```

```

For n = 1 To Text2.Text - k
    m = DemandArr(i - 1, j - 1) * HoldingCostArr(i - 1)
    W = W + m
Next
k = k + 1
If (Not j = Text2.Text) Then
    richTextBox1.Text += "+"
End If
p = p - 1
Next
p = Text2.Text
richTextBox1.Text += Chr(13)
Constant = Constant + W
k = Text2.Text
richTextBox1.Text += "+"
Next
For i = 1 To Text1.Text
    For j = 1 To Text2.Text
        richTextBox1.Text += " " & ProductCostArr(i - 1) & "Z" & i & j & " "
        If (Not j = Text2.Text) Then
            richTextBox1.Text += "+"
        End If
    Next
    richTextBox1.Text += Chr(13)
    If (Not i = Text1.Text) Then
        richTextBox1.Text += "+"
    End If

    If (i = Text1.Text) Then
        richTextBox1.Text += "-"
    End If

```

End If

Next

richTextBox1.Text += "Constant"

richTextBox1.Text += Chr(13)

richTextBox1.Text += Chr(13)

richTextBox1.Text += "SUBJECT TO"

richTextBox1.Text += Chr(13)

For i = 1 To Text1.Text

p = 1

For j = 1 To Text2.Text

W = 0

For k = 1 To p

richTextBox1.Text += " " & "X" & i & k & " "

m = DemandArr(i - 1, k - 1)

W = W + m

If (Not k = p) Then

richTextBox1.Text += "+"

End If

If (k = p) Then

richTextBox1.Text += ">=" & W

End If

Next

p = p + 1

richTextBox1.Text += Chr(13)

Next

```

    richTextBox1.Text += Chr(13)
Next

richTextBox1.Text += Chr(13)

For j = 1 To Text2.Text
    For i = 1 To Text1.Text
        richTextBox1.Text += " " & "X" & i & j & " "
        If (Not i = Text1.Text) Then
            richTextBox1.Text += "+"
        End If
        If (i = Text1.Text) Then
            richTextBox1.Text += "<=" & CapArr(j - 1)
        End If
    Next
    richTextBox1.Text += Chr(13)
Next

richTextBox1.Text += Chr(13)

For i = 1 To Text1.Text
    For j = 1 To Text2.Text
        richTextBox1.Text += " " & "X" & i & j & " "
        richTextBox1.Text += ">=" & 0
        richTextBox1.Text += Chr(13)
    Next
    richTextBox1.Text += Chr(13)
Next

For i = 1 To Text1.Text
    For j = 1 To Text2.Text

```

```

richTextBox1.Text += " " & "X" & i & j & " - " & CapArr(j - 1) & "Z" & i & j
richTextBox1.Text += "<=" & 0
richTextBox1.Text += Chr(13)
Next
richTextBox1.Text += Chr(13)
Next
richTextBox1.Text += Chr(13)
richTextBox1.Text += "Constant=" & Constant
richTextBox1.Text += Chr(13)
richTextBox1.Text += "END"
richTextBox1.Text += Chr(13)

For i = 1 To Text1.Text
  For j = 1 To Text2.Text
    richTextBox1.Text += "GIN " & "Z" & i & j
    richTextBox1.Text += Chr(13)
  Next
  richTextBox1.Text += Chr(13)
Next

richTextBox1.Text += Chr(13)
For i = 1 To Text1.Text
  For j = 1 To Text2.Text
    richTextBox1.Text += "GIN " & "X" & i & j
    richTextBox1.Text += Chr(13)
  Next
  richTextBox1.Text += Chr(13)
Next

activeEquatin.Controls.Add(richTextBox1)

```

End Sub

End Class

ข.6.1.3 ฟอรัมสมการของโปรแกรม Lot Size Solving มีหน้าที่แสดงสมการคณิตศาสตร์ เพื่อนำไปหาค่าที่ดีที่สุดด้วยโปรแกรมสำเร็จรูปมีโค้ดโปรแกรมดังนี้

```
Public Class EquationForm
```

```
Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
Public Sub New()
```

```
MyBase.New()
```

```
'This call is required by the Windows Form Designer.
```

```
InitializeComponent()
```

```
'Add any initialization after the InitializeComponent() call
```

```
End Sub
```

```
'Form overrides dispose to clean up the component list.
```

```
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
If disposing Then
```

```
    If Not (components Is Nothing) Then
```

```
        components.Dispose()
```

```
    End If
```

```
End If
```

```
MyBase.Dispose(disposing)
```


End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()

Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(EquationForm))

,

'EquationForm

,

Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)

Me.ClientSize = New System.Drawing.Size(640, 366)

Me.Icon = CType(resources.GetObject("\$this.Icon"), System.Drawing.Icon)

Me.Name = "EquationForm"

Me.Text = "Branch and Bound"

End Sub

#End Region

End Class

ข.6.1.4 ฟอรัมรายงานของโปรแกรม Lot Size Solving มีหน้าที่แสดงรายงานเพื่อแจ้งให้
ผู้ใช้งานทราบว่าโปรแกรมได้ย้ายหรือแบ่งย้ายความต้องการสินค้าชนิดใด จำนวนเท่าใด ไปยัง
ช่วงเวลาใดบ้างและคำนวณด้วยโปรแกรมย่อยไหนมีโค้ดโปรแกรมดังนี้

#Region " Windows Form Designer generated code "

Public Sub New()

MyBase.New()

'This call is required by the Windows Form Designer.

InitializeComponent()

'Add any initialization after the InitializeComponent() call

End Sub

'Form overrides dispose to clean up the component list.

Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)

If disposing Then

If Not (components Is Nothing) Then

components.Dispose()

End If

End If

MyBase.Dispose(disposing)

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()

```

    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(ReportForm))
,
'ReportForm
,
Me.AutoScaleBaseSize = New System.Drawing.Size(6, 15)
Me.ClientSize = New System.Drawing.Size(752, 266)
Me.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(222, Byte))
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
Me.Name = "ReportForm"
Me.Text = "Lot Sizing Report"
End Sub
#End Region
End Class

```

ข.6.2 โปรแกรมคำนวณตามวิธีการที่นำเสนอซึ่งพัฒนาจาก Visual C++ มีโค้ดโปรแกรมดังนี้

```

// cCalCost.cpp : Defines the initialization routines for the DLL.
#include "stdafx.h"
#include "cCalCost.h"
#include "afxtempl.h"
#include <math.h>
#ifdef _DEBUG
#define new DEBUG_NEW
#endif
#include <algorithm>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

```

using namespace System;

BEGIN_MESSAGE_MAP(CcCalCostApp, CWinApp)
END_MESSAGE_MAP()

// CcCalCostApp construction
CcCalCostApp::CcCalCostApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

#####Calculate Total Cost Algorithm#####
float __declspec (dllexport) __stdcall cInputCost(
int *pDemandArr, // The array's pointer
int *pDemandArrCon,
int nRow, // The number of rows in the array
int nCol, // The number of columns in the array
double *ProductArr,
double *HoldingArr,
int *SumDemandArr,
int *CapArr)// The number to be searched
{
    char buffer1[200];
    char buffer2[200];
    char buffer3[1000];

    Int32 ProductFac[] = __gc new Int32[nRow];
    Int32 CanMove[] = __gc new Int32[nRow];
    Int32 MoveIndex[] = __gc new Int32[nRow];
    Int32 outDemand[] = __gc new Int32[nRow];
    char cHandle;

```

```

//*****Calculate Total Cost before move*****

float m=0;
float mm=0;
int value=0;
double totalCost=0;
for (int i=0;i<nRow;i++)
{
    for (int j=0;j<nCol;j++)
    {
        outDemand[i]=outDemand[i]+(pDemandArr[value]-
        pDemandArrCon[value]);
        totalCost=totalCost+(outDemand[i]*HoldingArr[i]);
        if(pDemandArr[value]!=0)totalCost=totalCost+ProductArr[i];
        value++;
    }
}

FILE *stream;
stream = fopen( "data.txt", "w" );

//*****Check! What demand that can be moved*****

int colCheck=2;
for (int j=1;j<nCol;j++)//วนลูปเท่าจำนวนช่วงเวลา
{
    if(stream!=NULL){
        fprintf( stream,"-การพิจารณาช่วงเวลาที %d\n",j+1);
    }

    int k,x;
    float HoldingCost;
    k=j;
    while(k>=0)

```

```

{
    if(j-k!=0){
        for (int i=0;i<nRow;i++)
        {
            HoldingCost=pDemandArr[j+(i*nCol)]*HoldingArr[i]*(j-k);
            CanMove[i]=ProductArr[i]-HoldingCost;
            MoveIndex[i]=i;
        }//for i
        //*****Sort Demand *****
        //โปรแกรมส่วนนี้ใช้สำหรับหาชนิดสินค้าที่ขายแล้วทำให้ได้ต้นทุนที่ต่ำที่สุดก่อน
        //*****
        for (int ii = 0; ii < (nRow-1); ii++)
        {
            int minIndex = ii;
            for (int jj = ii + 1; jj < nRow; jj++)
            {
                if (CanMove[jj] > CanMove[minIndex])
                {
                    minIndex = jj;
                }//if (CanMove[jj] > CanMove[minIndex])
            }//for (int jj = ii + 1; jj < nRow; jj++)
            if (minIndex > ii)
            {
                int temp;
                temp = CanMove[ii];
                CanMove[ii] = CanMove[minIndex];
                CanMove[minIndex] = temp;

                temp = MoveIndex[ii];

```

```

        MoveIndex[ii] = MoveIndex[minIndex];
        MoveIndex[minIndex] = temp;
    } //if (minIndex > ii)

} //for (int ii = 0; ii < (nRow-1); ii++)

//*****Move Demands that can be move*****
//โปรแกรมส่วนนี้ใช้สำหรับย้ายความต้องการที่เรียงลำดับแล้ว
//*****

for(int i = 0; i < nRow; i++ )
    {
        int doMove=MoveIndex[i];
        if(CanMove[i]>=0)
        {
            int MoveDemand=pDemandArr[j+(doMove*nCol)];
            HoldingCost=pDemandArr[j+(doMove*nCol)]*HoldingArr[doMove]*(j-k);
            if(SumDemandArr[k]+MoveDemand<=CapArr[k] &&
                pDemandArr[k+(doMove*nCol)]!=0 && pDemandArr[j+(doMove*nCol)]!=0 &&
                ProductArr[doMove]-HoldingCost>0)
                {
                    totalCost=totalCost-(ProductArr[doMove]-HoldingCost);
                    x=pDemandArr[j+(doMove*nCol)]+pDemandArr[k+(doMove*nCol)];
                    pDemandArr[k+(doMove*nCol)]=x;

                    if(stream!=NULL){
                        fprintf( stream,"โปรแกรมย่อยที่1:ย้ายสินค้าชนิดที่ %d ช่วงเวลา %d จำนวน %d ไปยังช่วงเวลา
                        %d ลดต้นทุนไป %f\n",doMove+1,
                        j+1,pDemandArr[j+(doMove*nCol)],k+1,ProductArr[doMove]-HoldingCost);
                    }

                    x=SumDemandArr[k]+pDemandArr[j+(doMove*nCol)];
                    SumDemandArr[k]=x;

```

```

x=SumDemandArr[j]-pDemandArr[j+(doMove*nCol)];
SumDemandArr[j]=x;
pDemandArr[j+(doMove*nCol)]=0;

```

```

}
} //if(CanMove[i]>=0)
} //for(int i = 0; i < nRow; i++ )
} //if(j-k!=0)
k--;
} //while(k>=0)

```

***** Sub Algorithm*****

//โปรแกรมส่วนนี้ใช้สำหรับหาว่าสามารถแบ่งย้ายความต้องการไปยังช่วงเวลาก่อนหน้าได้หรือไม่

```

if(j>1)
{
int a;
a=j;
while(a >0)
{
if(j-a !=0)
{
for(int i = 0; i < nRow; i++ )
{
int MoveDemand=pDemandArr[j+(i*nCol)];
int NearDemandCap=CapArr[j-1]-SumDemandArr[j-1];
int FarDemandCap=CapArr[a-1]-SumDemandArr[a-1];
int FarDemand=MoveDemand-NearDemandCap;
float SubNearHoldingCost=NearDemandCap*HoldingArr[i]*(j-a);

```



```

float SubFarHoldingCost=FarDemand*HoldingArr[i]*(j-a);
float SubTotalCost = ProductArr[i]- (SubNearHoldingCost+SubFarHoldingCost);
CanMove[i]=SubTotalCost;
MoveIndex[i]=i;
} //for(int i = 0; i < nRow; i++ )
*****Sub Sort*****
//โปรแกรมส่วนนี้ใช้สำหรับเรียงว่าสามารถแบ่งย้ายความต้องการของสินค้าชนิดใดไปยังช่วงเวลา
ก่อนหน้าแล้วทำให้ต้นทุนลดลงมากที่สุด
*****
for (int ii = 0; ii < (nRow-1); ii++)
{
    int minIndex = ii;
    for (int jj = ii + 1; jj < nRow; jj++)
    {
        if (CanMove[jj] > CanMove[minIndex])
        {
            minIndex = jj;
        } //if (CanMove[jj] > CanMove[minIndex])
    } //for (int jj = ii + 1; jj < nRow; jj++)
    if (minIndex > ii)
    {
        int temp;
        temp = CanMove[ii];
        CanMove[ii] = CanMove[minIndex];
        CanMove[minIndex] = temp;
        temp = MoveIndex[ii];
        MoveIndex[ii] = MoveIndex[minIndex];
        MoveIndex[minIndex] = temp;
    } //if (minIndex > ii)

```

```

} //for (int ii = 0; ii < (nRow-1); ii++)

//***** Sub Move Demands that can be move*****
//โปรแกรมส่วนนี้ใช้สำหรับย้ายความต้องการที่เรียงลำดับแล้ว
//*****

for(int i = 0; i < nRow; i++)
{
    if(CanMove[i]>=0)
    {
        int doMove=MoveIndex[i];
        int NextDemand=pDemandArr[j+1+(doMove*nCol)];
        int MoveDemand=pDemandArr[j+(doMove*nCol)];
        int NearDemandCap=CapArr[j-1]-SumDemandArr[j-1];
        int FarDemandCap=CapArr[a-1]-SumDemandArr[a-1];
        int FarDemand=MoveDemand-NearDemandCap;

if(MoveDemand<=NearDemandCap+FarDemandCap && MoveDemand!=0 &&
pDemandArr[(j-1)+(doMove*nCol)]!=0&&pDemandArr[(a-1)+(doMove*nCol)]!=0)
        {
            float SubNearHoldingCost=NearDemandCap*HoldingArr[doMove];
            float SubFarHoldingCost=FarDemand*(HoldingArr[doMove]*(j+1-a));
            float SubTotalCost = ProductArr[doMove]-(SubNearHoldingCost+SubFarHoldingCost);

if(NextDemand+SumDemandArr[j-1]>=CapArr[j-1]&& SubTotalCost>0)
            {
                if(cHandle==IDCANCEL)
                {
                    goto StopHere;
                } //if(cHandle==IDCANCEL)

pDemandArr[(j-1)+(doMove*nCol)]=pDemandArr[(j-1)+(doMove*nCol)]+NearDemandCap;

```

```

pDemandArr[(a-1)+(doMove*nCol)]=pDemandArr[(a-1)+(doMove*nCol)]+FarDemand;

if(stream!=NULL){
fprintf( stream,"\tAlgorithm2:ย้ายสินค้าชนิดที่ %d ช่วงเวลา %d จำนวน %d ไปยังช่วงเวลา %d
และจำนวน %d ไปยังช่วงเวลา %d ลดต้นทุนไป %f\n ",doMove+1,
j+1,NearDemandCap,j,FarDemand,a,SubTotalCost);
    }

    SumDemandArr[j-1]=SumDemandArr[j-1]+NearDemandCap;
    SumDemandArr[a-1]=SumDemandArr[a-1]+FarDemand;
    SumDemandArr[j]=SumDemandArr[j]-MoveDemand;
    pDemandArr[j+(doMove*nCol)]=0;
    totalCost=totalCost-SubTotalCost;
}
}
}
CanMove[i]=-1;
}
}
a--;
}
}

if(stream!=NULL){
fprintf( stream,"\n");
}
}

StopHere ;;

fclose( stream );

return totalCost;

free ( pDemandArr );

free ( pDemandArrCon );

free ( ProductArr );

```

```
free ( HoldingArr );
free ( SumDemandArr );
free ( CapArr );
}
// The one and only CcCalCostApp object
CcCalCostApp theApp;
// CcCalCostApp initialization
BOOL CcCalCostApp::InitInstance()
{
    CWinApp::InitInstance();

    return TRUE;
}
```