

DPUC

ภาคผนวก

ภาคผนวก ก

ปัญหาตัวอย่างและผลการทดสอบของปัญหา

ภาคผนวก ก

ปัญหาตัวอย่างและผลการทดสอบของปัญหา

การทดสอบวิธีการแก้ปัญหาคำหนดปริมาณการสั่งซื้อหรือผลิตของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดทางสมรรถภาพร่วมกันและอยู่ภายใต้ข้อกำหนดที่ตั้งขึ้นด้วยวิธีการที่นำเสนอ นั้นจะใช้ปัญหาตัวอย่างซึ่งจำแนกออกเป็น 4 กลุ่มคือ กลุ่มปัญหาขนาดเล็ก กลุ่มปัญหขนาดกลาง กลุ่มปัญหาขนาดใหญ่และกลุ่มปัญหาตัวอย่างสำหรับทดสอบผลของสัดส่วนระหว่างค่าสั่งซื้อหรือผลิตต่อค่าเก็บรักษา ในการทดสอบประสิทธิภาพของวิธีการที่ได้นำเสนอ ผลการทดสอบของปัญหาในกลุ่มต่างๆ ได้ถูกรวบรวมไว้ภายใน 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

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

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 ส่วนวิธี การคำนวณแบบ 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

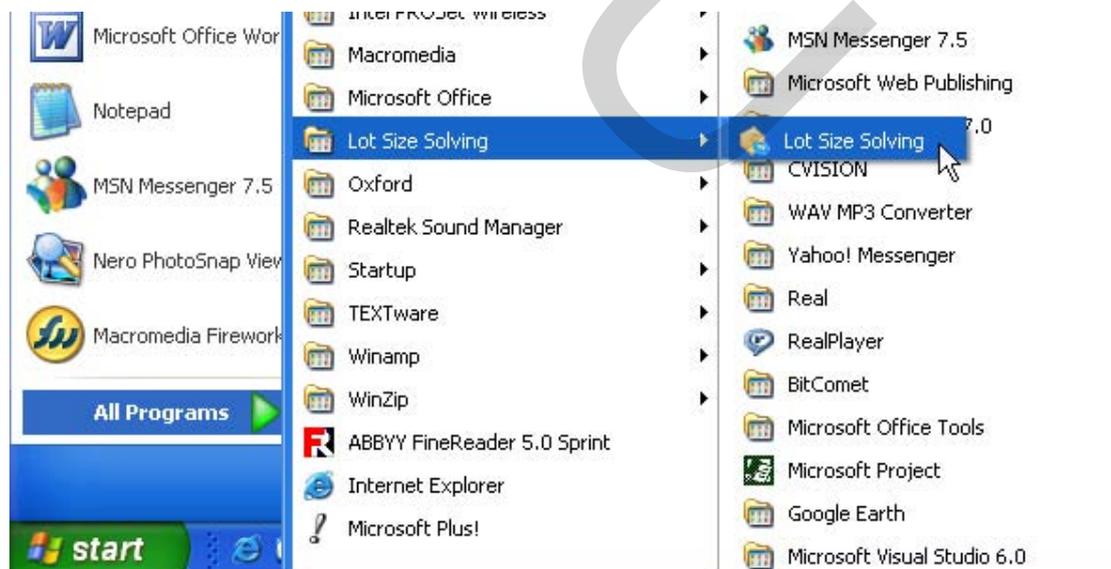
ผนวก ข

คู่มือการใช้งานโปรแกรมและโค้ดของโปรแกรม

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

ข.1 การเรียกใช้งานโปรแกรม Lot Size Solving

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



ภาพที่ ข.1 การเรียกใช้งานโปรแกรม

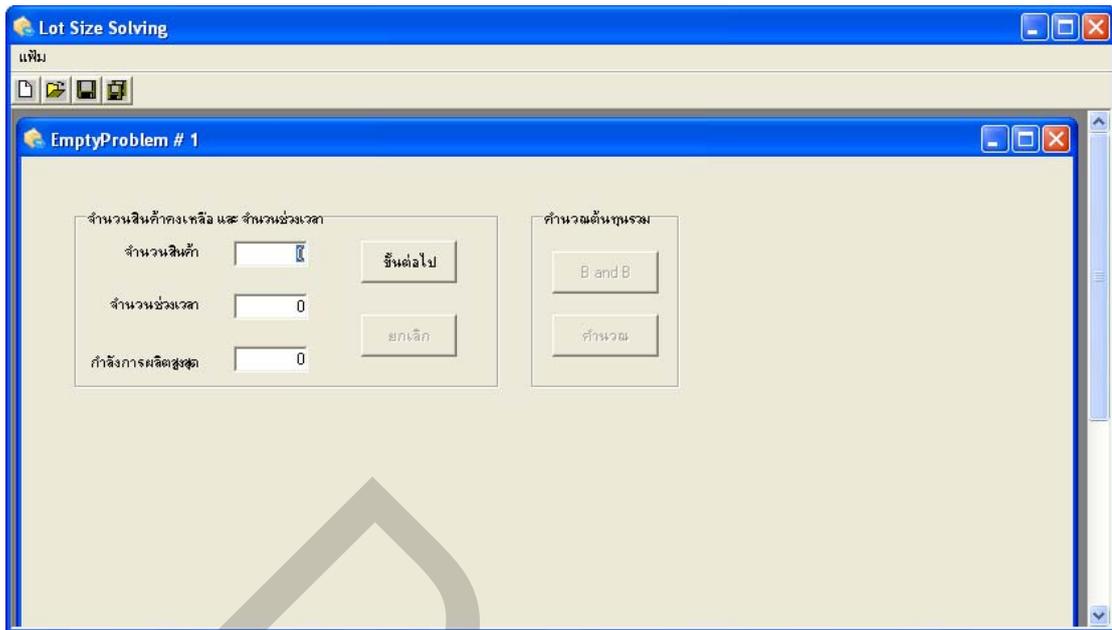
หลังจากคลิกที่ **Start > All Programs > Lot Size Solving > Lot Size Solving** จะปรากฏ หน้าต่างของโปรแกรม Lot Size Solving ดังภาพที่ ข.2



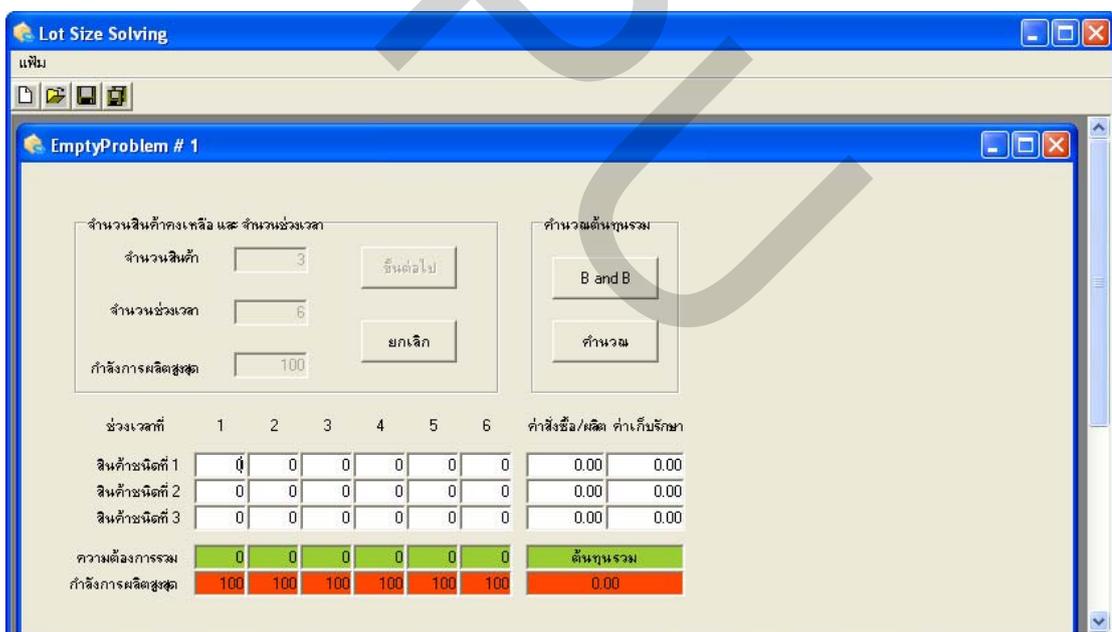
ภาพที่ ข.2 หน้าต่างโปรแกรม Lot Size Solving

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

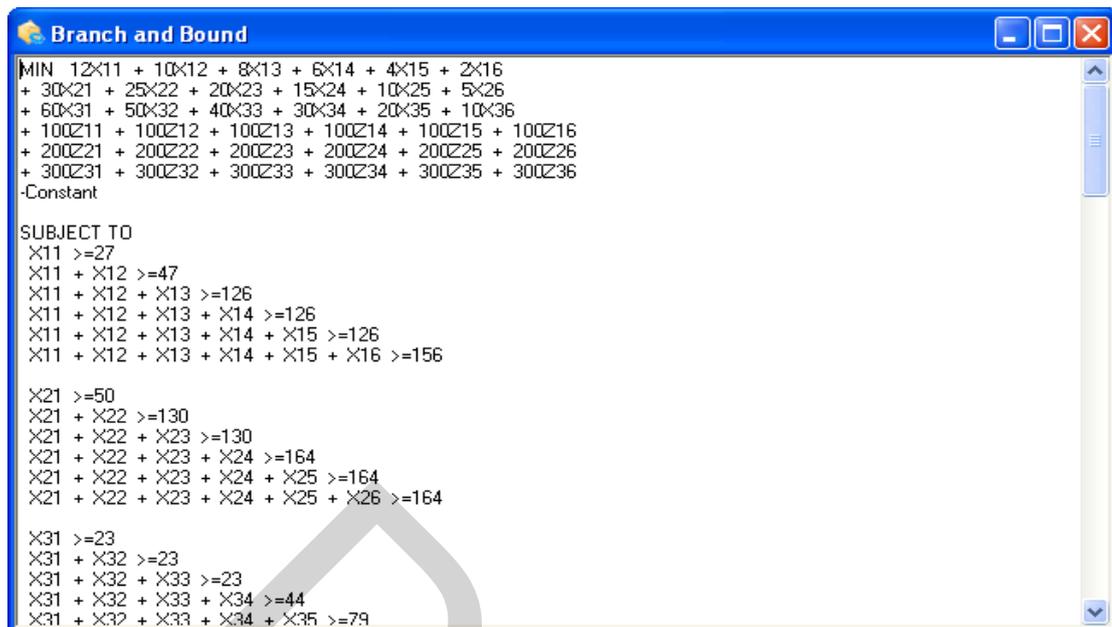
ให้ทำการกรอกค่าต่างๆของปัญหาเช่น ความต้องการของสินค้าแต่ละชนิดในช่วงเวลาต่างๆ ค่าสั่งซื้อหรือผลิตและค่าเก็บรักษาของสินค้าแต่ละชนิดตามที่ต้องการดังภาพที่ ข.5

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	ต้นทุนรวม	
ค่าสั่งการผลิตสูงสุด	100	100	100	100	100	100	3,300.00	

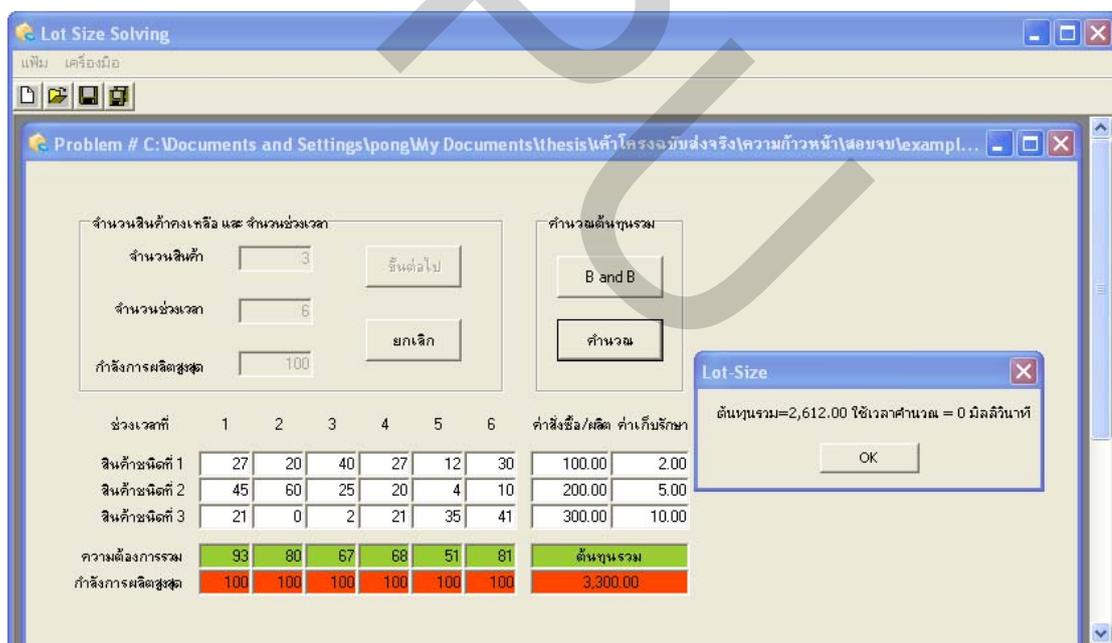
ภาพที่ ข.5 หน้าต่างปัญหาเปล่าที่กรอกข้อมูลสำหรับแก้ปัญหาเรียบร้อยแล้ว

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

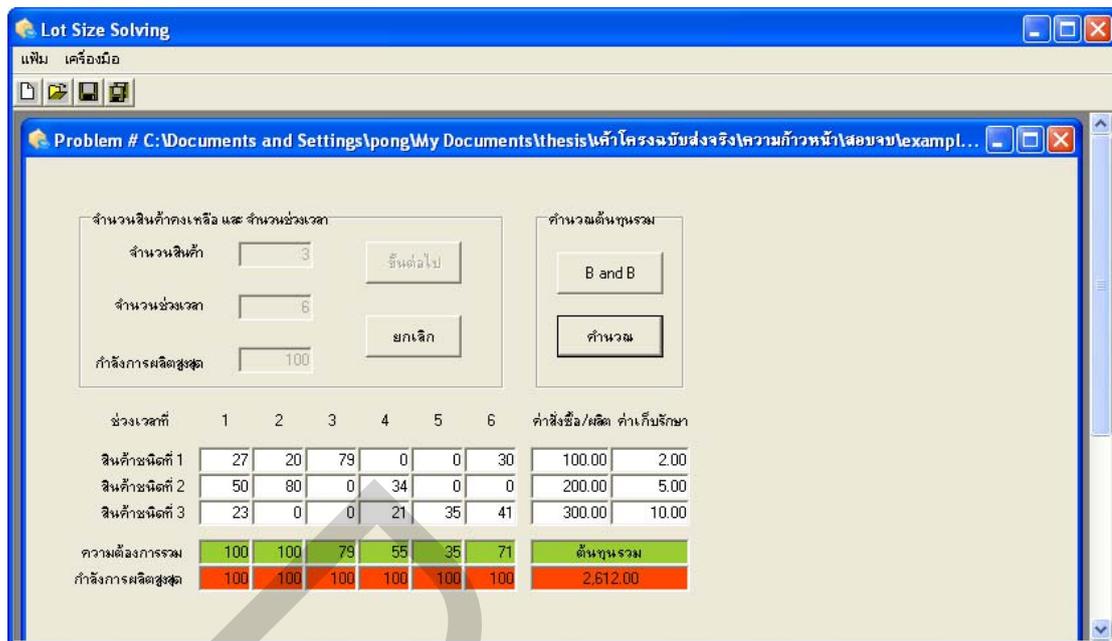
เมื่อมาถึงขั้นตอนนี้สามารถที่จะคำนวณหาคำตอบที่เหมาะสมโดยประมาณได้โดยคลิกที่ปุ่ม **คำนวณ** โปรแกรมจะทำการคำนวณหาคำตอบและเมื่อ โปรแกรมคำนวณเสร็จเรียบร้อยแล้วจะปรากฏหน้าต่างแสดงต้นทุนรวมและเวลาที่ใช้ในการคำนวณดังภาพที่ ข.7 จากภาพที่ ข.7 ให้คลิกที่ปุ่ม **OK** โปรแกรมจะทำการแสดงผลปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมโดยประมาณของสินค้าในแต่ละช่วงเวลาและต้นทุนรวมตามปริมา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



ภาพที่ ข.6 หน้าต่างแสดงสมการคณิตศาสตร์

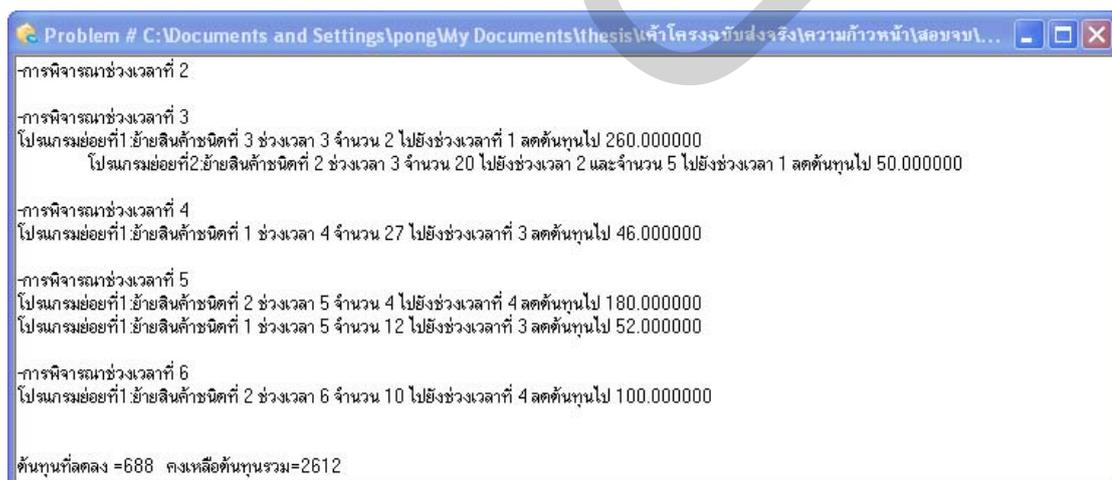


ภาพที่ ข.7 หน้าต่างแสดงต้นทุนรวมและเวลาที่ใช้ในการคำนวณ



ภาพที่ ข.8 ปริมาณการสั่งซื้อหรือผลิตที่เหมาะสมโดยประมาณของสินค้าในแต่ละช่วงเวลาและต้นทุนรวม

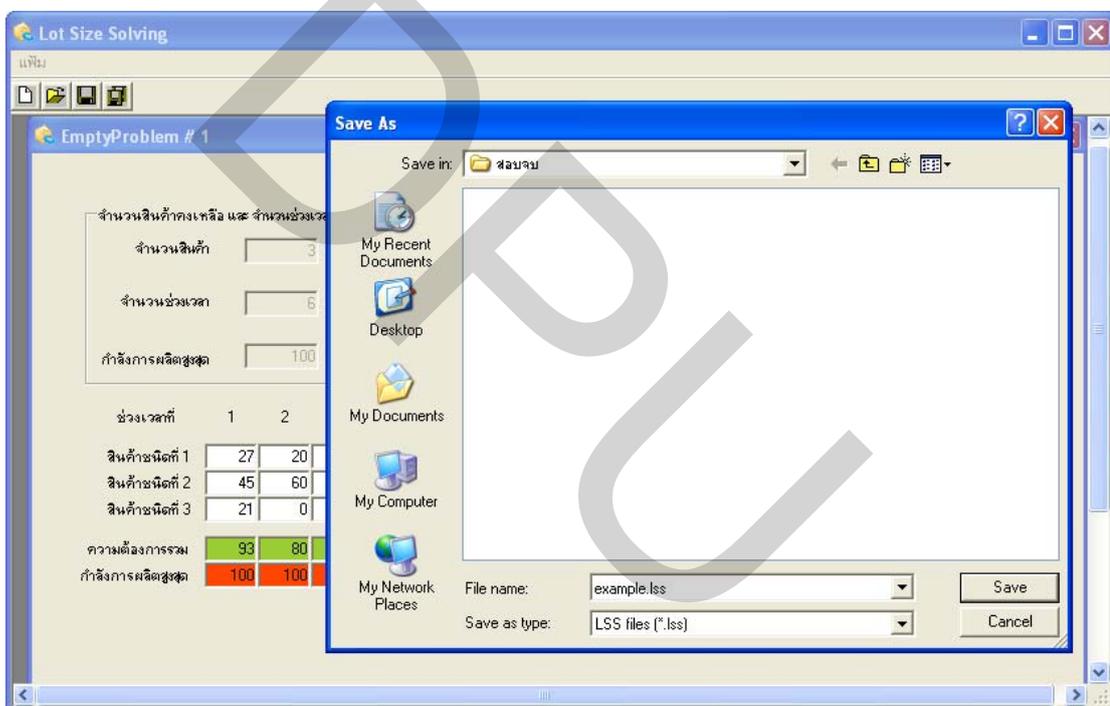
และโปรแกรมจะสร้างหน้าต่างรายงานว่าทำการย้ายหรือแบ่งย้ายความต้องการของสินค้าชนิดใด ช่วงเวลาใด เป็นจำนวนกี่หน่วยและสามารถลดต้นทุนได้เท่าไรดังภาพ ข.9



ภาพที่ ข.9 หน้าต่างรายงานแสดงผลการคำนวณ

ข.3 การบันทึกและบันทึกเป็นของโปรแกรม

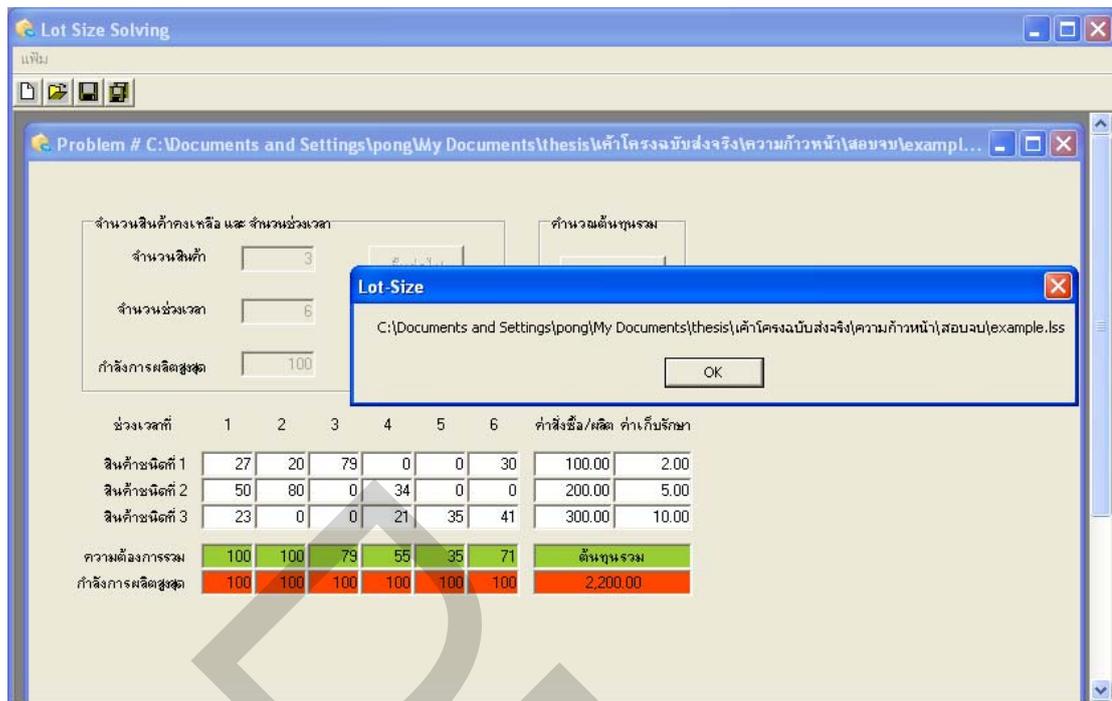
โปรแกรมสามารถเก็บบันทึกปัญหาการกำหนดปริมาณการสั่งซื้อหรือผลิตแบบประหัดของสินค้าคงคลังหลายชนิดที่มีข้อจำกัดเชิงสมรรถภาพร่วมกันที่ได้กรอกข้อมูลต่างๆไว้เรียบร้อยแล้วได้โดยคลิกที่ **แฟ้ม > บันทึก** หรือคลิกที่ไอคอน  จะปรากฏหน้าต่างดังภาพที่ ข.10 ซึ่งสามารถที่จะเลือกสถานที่จัดเก็บไฟล์ได้จากช่อง Save in และสามารถกรอกชื่อไฟล์ที่ต้องการได้ในช่อง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



ภาพที่ ข.10 หน้าต่างสำหรับบันทึกปัญหา

หลังจากบันทึกข้อมูลเรียบร้อยแล้วหากมีการแก้ไขข้อมูลความต้องการสินค้า ค่าการสั่งซื้อหรือผลิตหรือค่าเก็บรักษาและหากต้องการบันทึกสิ่งที่ได้แก้ไขไปสามารถทำได้ 2 แบบคือ

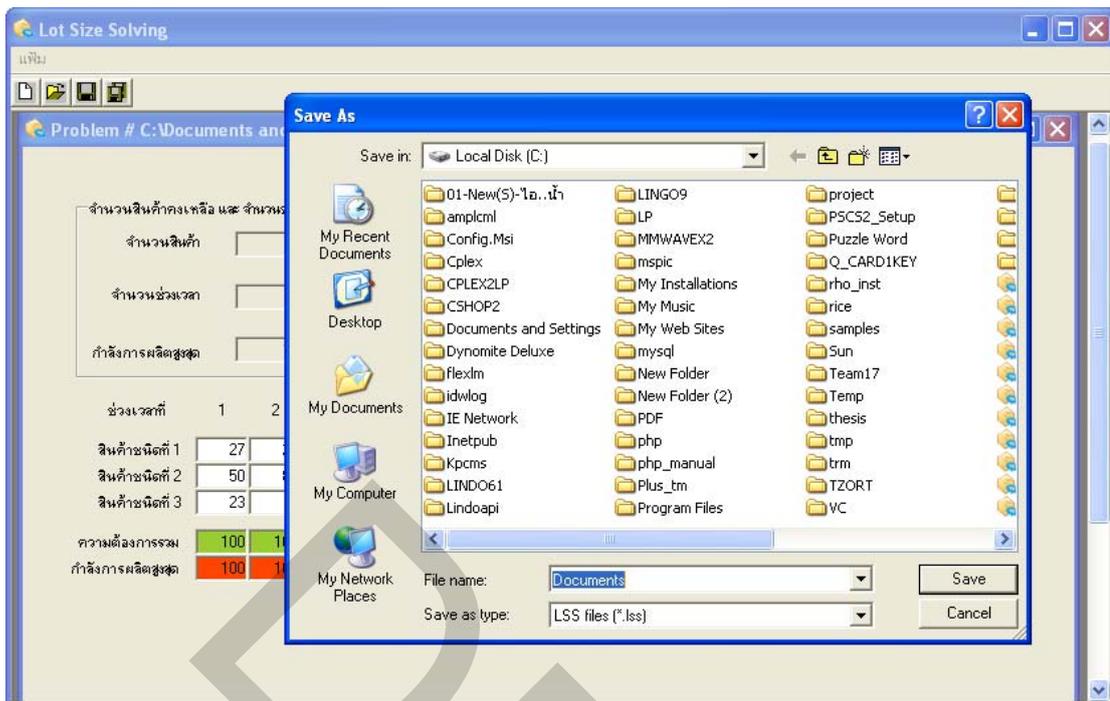
ข.3.1 การบันทึกไฟล์ในชื่อและสถานที่จัดเก็บเดิม การบันทึกในลักษณะนี้เป็นการบันทึกทับไฟล์เดิมนั้นเองสามารถทำได้โดยคลิกที่ **แฟ้ม > บันทึก** หรือคลิกที่ไอคอน  โปรแกรมจะบันทึกข้อมูลใหม่ทันทีและจะมีหน้าต่างแจ้งว่าได้บันทึกไฟล์นั้นไว้ที่ตำแหน่งใดในเครื่องคอมพิวเตอร์ดังภาพที่ ข.11 หลังจากนั้นให้คลิกที่ **OK**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



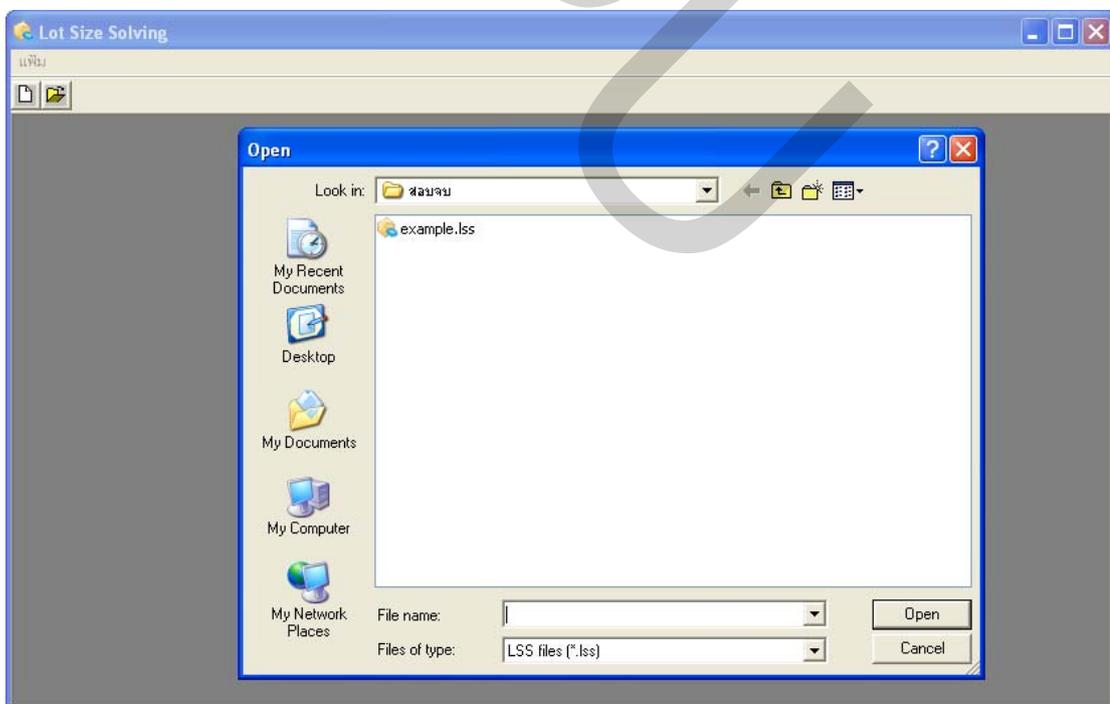
ภาพที่ ข.11 หน้าต่างแสดงการบันทึกปัญหา

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 อยู่แล้วก็อาจไม่จำเป็นต้องอ่านคู่มือนี้เลยก็ได้ แต่ถึงอย่างไรก็ตามสำหรับผู้ใช้งานที่ไม่คุ้นเคยกับการใช้งานโปรแกรมบนระบบปฏิบัติการ Windows มาก่อนก็สามารถใช้คู่มือนี้เป็นแนวทางในการใช้งานโปรแกรมได้ไม่ยากนัก

ข.6 โค้ดของโปรแกรม Lot Size Solving

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

ข.6.1 โปรแกรมสำหรับติดต่อกับผู้ใช้งานซึ่งพัฒนาจาก Visual Basic ประกอบไปด้วย 4 ฟอรัมได้แก่

ข.6.1.1 ฟอรั่มหลักของโปรแกรม Lot Size Solving ฟอรั่มหลักนี้มีหน้าที่ควบคุมการทำงานต่างๆของโปรแกรมเช่น การสร้างฟอรั่มลูกเพื่อใช้ในการป้อนข้อมูลปัญหา 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 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.

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 = "&ไฟล์"
'
'MenuItem2
'
Me.MenuItem2.Index = 0
Me.MenuItem2.ShowShortcut = False
Me.MenuItem2.Text = "&การตั้งค่า..."

```

```
' MenuItem7
,

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

'MenuItem5
,

Me.MenuItem5.Enabled = False
Me.MenuItem5.Index = 2
Me.MenuItem5.Text = "ปี&ด"
,

'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 = "บ้&นทึกเป็น..."
,

'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)
    End If
End If

```

```
        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 Sub
```

```

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 ค่าตั้งซื้อหรือผลิตและค่าเก็บรักษาเป็นต้นมี โค้ดโปรแกรมดังนี้

```

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 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.MidRight
,
'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)

'สร้างอินพุทความต้องการในแต่ละช่วงเวลา-----
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

'สร้าง Label ของความต้องการรวม
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=" & VBTotalCost - 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 j
    Next i
End Sub

```

```

        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
    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;  
}
```