

รายงานวิจัยฉบับสมบูรณ์

โครงการ การการเพิ่มผลผลิตการใช้ไม้ MDF ในการผลิตเฟอร์นิเจอร์
**Medium Fiber Board Productivity Improvement
in Furniture Production**

โดย ดร.เตือนใจ สมบูรณ์วิวัฒน์ และคณะ

27 มีนาคม 2547

รายงานวิจัยฉบับสมบูรณ์

โครงการ การเพิ่มผลผลิตการใช้ไม้ MDF ในการผลิตเฟอร์นิเจอร์ Medium Fiber Board Productivity Improvement in Furniture Production

คณะผู้วิจัย

สังกัด

1. ดร.เตือนใจ สมบูรณ์วัฒน์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
2. นาย สุโรจน์ ภัคดี มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย (สกว.)

Executive Summary

Project Title: Medium Fiber Board Productivity Improvement in Furniture Production

Medium fiber board (MDF) is one of the main materials used in furniture production, and significantly affects overall production costs. Thus, MDF productivity improvement is necessary. The MDF utilization and waste at P.P. Parawood Co., Ltd. is studied. The 2-dimensional MDF cut for different sets of product parts is the case study. The company has about 200 product items and each product item is composed of 4-5 parts. Currently, the company designs cutting patterns for each product item manually. Examination of the existing production reveals that wastes of MDF are about 30 % in the first step of the pattern cutting process. Consequently, the 2-dimensional cutting patterns are improved such that MDF wastes are minimized. The purposes of the project are to develop two-dimensional board cutting patterns for each product produced, that minimize total area usage of the MDF material, and to design rectangular cutting patterns for two-dimensional cut parts using the computer program.

The methodology of the research work is to (1) study and analyze MDF requirements, utilization, cutting constraints, waste, and MDF productivity improvement, (2) construct cutting models and select algorithms to solve the model, (3) design rectangular cutting patterns for two-dimensional cut parts using the computer, (4) test, analyze and compare results in terms of quantity and costs. The study presents the complicated and large-scale problem, then the multi-phase methodology for solving the problem is developed. The phases consist of possible pattern development, cutting pattern selection and the optimization pattern decisions. To solve the model in the multi-phase, the computer program is developed and implemented to design cutting patterns, select maximized utilization patterns, and to decide the optimal patterns. The program is designed and constructed in three parts. These are (1) the form for the input data, (2) the database linkage to the data in the main program, and (3) the results and report presentation.

The program is applied and tested using many work pieces required for the furniture production in the company. The program is developed for ease of use, time reduction in the cutting pattern calculation process, the waste minimization for the patterns, the graphical pattern presentation results for the cutting process, and then waste and cost calculations are addressed. The testing results show that the waste ranges from 10 to 15 %, depending on the size constraints and

required number of combined workpieces. This waste reduction increases MDF productivity and reduces furniture production costs.

บทคัดย่อ

โครงการวิจัยฉบับนี้เป็นการวิจัยและศึกษาการประยุกต์ใช้คอมพิวเตอร์ เพื่อนำมาช่วยในการแก้ปัญหาการตัดวัสดุแบบสองมิติ ซึ่งมีลักษณะรูปแบบการตัดแบบกิโลตันตั้งตรง โดยลักษณะปัญหาที่ทำการศึกษาจะเป็นการพัฒนาวิธีในการออกแบบรูปแบบการตัดแบ่งชิ้นงานไม้บอร์ดสี่เหลี่ยมมุมฉากขนาดมาตรฐานขนาดเดียว ออกเป็นรูปสี่เหลี่ยมมุมฉากย่อยขนาดต่างๆตามต้องการ โดยมีจุดมุ่งหมายให้เกิดการสูญเสียไม้บอร์ดให้น้อยที่สุด การดำเนินงานการวิจัยจะทำการศึกษารายละเอียดของปัจจัยต่างๆที่มีผลกระทบกับการออกแบบ ซึ่งได้พัฒนาวิธีการแก้ปัญหาขึ้นโดยการผสมผสานวิธีการทางคณิตศาสตร์ และ วิธีการทางฮิวริสติก ซึ่งขั้นตอนในการแก้ปัญหาที่มีลักษณะเป็น Muti-Phase มีแบ่งปัญหาออกเป็น 3 ขั้นตอนหลักคือ

- การหารูปแบบที่เป็นไปได้ทั้งหมด
- การคัดเลือกรูปแบบที่ให้ประโยชน์สูงสุดในแต่ละรอบการทำงานมาประมวลผล
- การหาปริมาณรูปแบบที่เหมาะสมกับเงื่อนไขการผลิต

การประมวลผลนั้นจะอาศัยการสร้างโปรแกรมทดสอบขึ้นเพื่อแก้ปัญหาในแต่ละขั้นตอนซึ่งจะมีวิธีการประยุกต์ใช้ที่แตกต่างกันออกไป (1) สำหรับการหารูปแบบในการตัดแบบกิโลตันตั้งตรงที่เป็นไปได้ทั้งหมดนั้นอาศัยวิธีการทางฮิวริสติก โดยการสร้างเงื่อนไขในการวางชิ้นงานซึ่งมีรูปแบบลงล่าง-ชิดซ้าย โดยปริมาณสถานะที่เกิดขึ้นนั้นเป็นสถานะที่สามารถนำไปใช้ได้จริง โดยจะแปรผันตามจำนวนรูปร่างของชิ้นงานที่นำมาจัดเรียง ขนาดชิ้นงานทดสอบ และอยู่ภายใต้ข้อจำกัดด้านหน่วยความจำของคอมพิวเตอร์และเวลาในการประมวลผล (2) การคัดเลือกรูปแบบที่ให้ประโยชน์สูงสุดในแต่ละรอบการทำงานมาประมวลผลเนื่องจากมีข้อจำกัดในการด้านตัวแปรการคำนวณของโปรแกรมร่วมที่ใช้ทำให้ต้องมีการคัดเลือกรูปแบบที่ดีมาทำการประมวลผล (3) ประมวลผลเพื่อหาปริมาณรูปแบบที่เหมาะสมกับเงื่อนไขการผลิต จากนั้นศึกษาประสิทธิภาพโดยรวมของโปรแกรม เปรียบเทียบผลลัพธ์จากการใช้โปรแกรมกับผลลัพธ์จากการปฏิบัติงานจริง รวมถึงความสามารถในการเปลี่ยนแปลงข้อมูลการผลิตบางข้อมูล หรือ เปลี่ยนแปลงเป้าหมายการผลิตใหม่

จากการศึกษาวิจัยพบว่า โปรแกรมที่ได้สร้างขึ้นเพื่อช่วยในการจัดวางชิ้นงานบนแผ่นไม้บอร์ด ทำให้การหารูปแบบการจัดวางมีประสิทธิภาพมากขึ้น อีกทั้งช่วยเพิ่มความสะดวกและรวดเร็ว หากมีการเปลี่ยนแปลงการผลิต โดยโปรแกรมนี้อาจสามารถทำงานได้อย่างต่อเนื่องมีการแสดงผลออกมาในรูปแบบข้อมูลในเชิงปริมาณ ต้นทุนที่เกิดขึ้น และรูปกราฟฟิก

ABSTRACT

The purpose of this research project is to create a computer program to design two-dimensional stock cutting patterns. This problem is referred to as an orthogonal guillotine pattern cutting problem. In this study, the objective is to design cutting patterns, from a standard size to various smaller demand sizes, with the goal of minimizing trim loss. The Algorithms and techniques have been developed and applied to obtain a solution based on user specifications. Furthermore, the results are analyzed and examined to make efficiency comparisons.

This program was tested on a number of actual work pieces, and the results were compared with actual designs from process engineers. In addition, one of the outstanding features of this program is that users can investigate alternative results when some factors or production targets change.

กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณหน่วยงานและบุคคลที่มีส่วนช่วยในการสนับสนุนงานวิจัยครั้งนี้คือ

1. สำนักงานกองทุนสนับสนุนการวิจัย(สกว.)และบริษัท P.P. PARAWOOD จำกัด
2. ที่ได้ให้ทุนสนับสนุนในงานวิจัยครั้งนี้
3. รองศาสตราจารย์ ดร. ลัดดาวัลย์ รัชมิทัต ที่ช่วยอำนวยความสะดวกและประสานงานให้กับคณะผู้วิจัยกับสำนักงานกองทุนสนับสนุนการวิจัย(สกว.) และบริษัท P.P. PARAWOOD จำกัด
4. คุณพงษ์พนิช วงษ์ดีไทย คุณทองด้อย ผ่านมรดา คุณธีรพงษ์ เอี้ยวอุษะจินดา และ คุณสาโรช ดวงสิทธิตานนท์ ตลอดจนบุคคลากรในโรงงานทุกท่านที่ช่วยประสานงานและอำนวยความสะดวกในการศึกษาระบบงานและข้อมูลที่เกี่ยวข้องกับงานวิจัยครั้งนี้เป็นอย่างดี
5. ภาควิชาวิศวกรรมอุตสาหกรรม บัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
6. คณะกรรมการทุกท่านที่ช่วยในการตรวจสอบเอกสาร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
รายการตาราง	ช
รายการรูปประกอบ	ซ
รายการสัญลักษณ์	ญ
ประมวลคำศัพท์และคำย่อ	ฉ
 บทที่	
1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ขั้นตอนการดำเนินงานวิจัย	2
1.5 ประโยชน์ที่ได้รับจากงานวิจัย	2
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 งานวิจัยที่เกี่ยวข้อง	4
2.2 ทฤษฎี	8
3 วิธีการดำเนินงานวิจัย	12
3.1 ศึกษาสภาพปัจจุบันของบริษัทที่เข้าร่วมโครงการ	12
3.2 การวิเคราะห์ข้อมูลและการแก้ปัญหา	12
3.3 การดำเนินการแก้ปัญหาการจัดวางชิ้นงาน	15
3.4 การหารูปแบบการจัดเรียงชิ้นงานเริ่มต้นทั้งหมดที่เป็นไปได้	16
3.5 การคัดเลือกรูปแบบการจัดเรียงชิ้นงานเบื้องต้นก่อนการประมวลผล	24
3.6 การประมวลผลเพื่อหารูปแบบที่เหมาะสมกับการผลิต	25

สารบัญ(ต่อ)

	หน้า
3.7 การออกแบบโครงสร้างการใช้งานโปรแกรม	26
3.7.1 การออกแบบหน้าจอการทำงานของโปรแกรม	27
3.7.2 การออกแบบระบบฐานข้อมูล	27
4 ผลการดำเนินงานวิจัย	31
4.1 ผลการศึกษาสภาพปัจจุบันของบริษัทที่เข้าร่วมโครงการ	31
4.1.1 สภาพปัญหาในส่วนงานผลิตที่เกี่ยวข้อง	34
4.2 ผลการวิเคราะห์และเก็บข้อมูล	35
4.3 ตัวแบบทางคณิตศาสตร์	35
4.3.1 การหารูปแบบการตัดที่เป็นไปได้	35
4.3.2 การหาจำนวนแผ่นไม้ที่น้อยที่สุดที่ต้องเบิกใช้งาน	37
4.4 ผลประยุคต์การใช้งานตัวแบบทางคณิตศาสตร์ด้วยคอมพิวเตอร์	38
4.4.1 ลำดับขั้นตอนในการดำเนินงานในการแก้ไขปัญา	38
4.4.2 ลำดับขั้นตอนในการประมวลผลของโปรแกรม	40
4.5 ผลทดสอบการทำงานของโปรแกรม	47
4.5.1 รูปแบบในการตัดแบบกิโยตินตั้งตรงตัวอย่างที่ 1	47
4.5.2 รูปแบบในการตัดแบบกิโยตินตั้งตรงตัวอย่างที่ 2	50
4.5.3 รูปแบบในการตัดแบบกิโยตินตั้งตรงตัวอย่างที่ 3	52
4.5.4 รูปแบบในการตัดแบบกิโยตินตั้งตรงตัวอย่างที่ 4	55
4.6 ผลการเปรียบเทียบประสิทธิภาพในด้านต่างๆของโปรแกรมทดสอบ	59
5 สรุปผลการดำเนินการและข้อเสนอแนะในการดำเนินการวิจัย	60
5.1 บทสรุป	60
5.2 ข้อเสนอแนะ	61
เอกสารอ้างอิง	62

สารบัญ(ต่อ)

	หน้า
ภาคผนวก	
ก แสดงการแผนภาพการไหลของการควบคุมการตัดไม้บอร์ด	65
ข ตัวอย่างรูปแบบการจัดวางชิ้นงานของหน่วยวางแผนการผลิต	69
ค แสดงตัวอย่างการวิเคราะห์เพื่อหาต้นทุนที่เกิดจากการวางแผนงานตัดในหน่วยการผลิต	72
ง แสดงภาษาคอมพิวเตอร์ของโปรแกรม VBA	78
จ แสดงภาษาคอมพิวเตอร์ของโปรแกรม VB	84

รายการตาราง

ตาราง	หน้า
3.1 แสดงตัวอย่างชิ้นส่วนที่ต้องการผลิต	13
3.2 แสดงรายการชิ้นส่วนที่ต้องการผลิต	14
3.3 แสดงการเปรียบเทียบผลการทดลองกับผลที่ได้จากหน่วยการผลิต	30
4.1 แสดงต้นทุนวัตถุดิบที่เกิดขึ้นตลอดระยะเวลา 3 เดือน	35
4.2 แสดงข้อมูลที่ต้องการทดสอบของตัวอย่างที่ 1	47
4.3 แสดงข้อมูลผลจากการทดสอบด้วยโปรแกรม	47
4.4 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ	49
4.5 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 1 กับผลที่ได้จากหน่วยการผลิต	49
4.6 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ HB ความหนา 4 มม.	50
4.7 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 2 ด้วยโปรแกรม	50
4.8 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ	51
4.9 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 2 กับผลที่ได้จากหน่วยการผลิต	51
4.10 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ MDF ความหนา 18 มม.	52
4.11 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 3 ด้วยโปรแกรม	53
4.12 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ	54
4.13 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 3 กับผลที่ได้จากหน่วยการผลิต	54
4.14 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ MDF ความหนา 2.6 มม.	55
4.15 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 4 ด้วยโปรแกรม	56
4.16 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ	57
4.17 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 4 กับผลที่ได้จากหน่วยการผลิต	58
4.18 แสดงเปอร์เซ็นต์ความแตกต่างในด้านประสิทธิภาพในด้านต่างๆ	59
ก.1 แสดงพื้นที่ต่างๆที่ได้จากการวางแผนการตัด	73
ก.2 แสดงการคำนวณที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 7	75
ก.3 แสดงการคำนวณที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 8	76
ก.4 แสดงการคำนวณที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 9	77

รายการรูปประกอบ

รูป	หน้า
2.1 แสดงรูปแบบ Orthogonal ของการจัดวางวัสดุ	4
2.2 แสดงรูปแบบ Non-Orthogonal ของการจัดวางวัสดุ	5
2.3 แสดงแบบจำลองปรัชญาของกำหนดการเชิงพลวัต (Dynamic Programming)	8
2.4 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านขวา	9
2.5 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านขวา-บน	9
2.6 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านบน	10
2.7 แผนการไหลของโปรแกรมในขั้นตอนเริ่มต้นของรูปแบบการจัดวางชิ้นงานแบบกิโลตินตั้งตรง	11
3.1 แสดงรูปการจัดวางชิ้นงานที่เกิดขึ้นจากชิ้นส่วนต่างๆที่ต้องการผลิต	13
3.2 แผนผังการไหลของขั้นตอนที่ 1 ของการบรรจุสี่เหลี่ยมขนาดเล็กลงในสี่เหลี่ยมขนาดใหญ่	19
3.3 แผนผังการไหลขั้นตอนที่ 1 ของการบรรจุสี่เหลี่ยมขึ้นต่อไปตามลำดับ	20
3.4 แผนผังการขั้นตอนที่ 1 ของการบรรจุสี่เหลี่ยมขึ้นต่อไป	21
3.5 แผนผังการไหลขั้นตอนที่ 1 ของการบรรจุสี่เหลี่ยมขึ้นต่อไป	22
3.6 แผนผังการไหลขั้นตอนที่ 2 การบรรจุกรณีที่บรรจุลงในช่องว่างด้านขวา	23
3.7 แสดงขั้นตอนการจัดเรียงข้อมูลเพื่อคัดเลือก	24
3.8 แสดงภาพรวมของการไหลของข้อมูลผ่านโปรแกรมทางคอมพิวเตอร์	26
3.9 แสดงรูปคอนโทรลต่างๆในโปรแกรม Visual Basic 6 ที่ใช้ในการออกแบบหน้าจอ	27
3.10 แสดงการออกแบบตาราง BLOCK ในฐานข้อมูล TEMP.MDB	28
3.11 แสดงการออกแบบตาราง BLOCKPOSITION ในฐานข้อมูล TEMP.MDB	28
3.12 แสดงการออกแบบตาราง BLOCKPOSITION ในฐานข้อมูล TEMP.MDB	29
3.13 แสดงการออกแบบตาราง PATTERN ในฐานข้อมูล TEMP.MDB	29
3.14 แสดงการออกแบบตาราง SPACES ในฐานข้อมูล TEMP.MDB	30
4.1 แสดงแผนผังเครื่องจักรในแผนงานตัด/เพชร	33
4.2 การใช้ไม้บอร์ดในขั้นตอนการตัด	34
4.3 แสดงแผนภาพการไหลของโปรแกรม	39
4.4 แสดงการตั้งค่าเริ่มต้นการทำงาน	40
4.5 แสดงการกรอกรายละเอียดของชิ้นงาน	41
4.6 แสดงผลการประมวลผลเพื่อหารูปแบบที่เป็นไปได้ทั้งหมด	42

รายการรูปประกอบ(ต่อ)

รูป		หน้า
4.7	แสดงตารางฐานข้อมูลในการเก็บข้อมูลต่างๆในการประมวลผล	43
4.8	แสดงตาราง ALLPATTERN ในโปรแกรมเก็บฐานข้อมูล ACCESS	44
4.9	แสดงข้อมูลในตาราง ALLPATTERN	44
4.10	แสดงการประมวลผลใน EXCEL WORKSHEET โดยผ่าน SOLVER	44
4.11	แสดงตาราง PATTERN ในโปรแกรมเก็บฐานข้อมูล ACCESS	45
4.12	แสดงข้อมูลในตาราง PATTERN	45
4.13	แสดงการแสดงผลกราฟฟิคของโปรแกรม	46
4.14	แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 1	48
4.15	แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 2	51
4.16	แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 3	53
4.17	แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 4	57
ก.1	FLOW CHART แสดงการควบคุมการตัดไม้บอร์ค	66
ก.2	FLOW CHART แสดงการควบคุมการตัดไม้บอร์ค	67
ก.3	FLOW CHART แสดงการควบคุมการเพรส	68
ข.1	แสดงพื้นที่ในแต่ละส่วนของรูปแบบการจัดวางตัวอย่างที่ 1	70
ข.2	แสดงพื้นที่ในแต่ละส่วนของรูปแบบการจัดวางตัวอย่างที่ 2	71

รายการสัญลักษณ์

C_{ij}^-	=	จำนวนแถวของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j
C_{ij}^+	=	จำนวนสคมภ์ของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j
C_{ij}	=	จำนวนชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j
W_{ij}	=	ความกว้างของชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j
L_{ij}	=	ความยาวของชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j
x_j	=	จำนวนแพทเทินที่ j
D_i	=	จำนวนชิ้นงานที่ i ที่ต้องการผลิต
A_i	=	พื้นที่ชิ้นงานที่ i

ประมวลคำศัพท์และคำย่อ

Two-dimensional cutting stock	ปัญหาการตัดแบ่งวัสดุแบบ 2 มิติ
Orthogonal cutting	รูปแบบการตัดแบ่งแบบตั้งฉาก
Non-Orthogonal cutting	รูปแบบการตัดแบ่งแบบไม่ตั้งฉาก
Guillotine	รูปแบบการตัดที่ตัดได้ตลอดแนวการตัด
Non-Guillotine	รูปแบบการตัดที่ตัดไม่ได้ตามแนวการตัด

บทที่1 บทนำ

1.1 ความเป็นมาและความสำคัญของเรื่อง

อุตสาหกรรมเฟอร์นิเจอร์ไม้เป็นอุตสาหกรรมหนึ่งที่มีบทบาทสำคัญในการพัฒนาเศรษฐกิจ ของประเทศมูลค่าการส่งออกเฟอร์นิเจอร์ไม้ในช่วงสามปีที่ผ่านมา มีดังนี้ ปี2542 มีมูลค่า 14,017.89 ล้านบาท ปี2543 มีมูลค่า 18,555.12 ล้านบาท และในปี 2544 มีมูลค่า 14,580.99 ล้านบาท(กรมศุลกากร) ในปัจจุบันสถานะการแข่งขันในอุตสาหกรรมเฟอร์นิเจอร์ไม้มีมากขึ้นและ ลูกค้ามีความต้องการสินค้าหลากหลายชนิด เพื่อตอบสนองความต้องการของลูกค้า ผู้ผลิตต้องเผชิญกับปัญหาการจัดการผลิตที่มีความซับซ้อนมากขึ้น รวมทั้งปัญหาด้านต้นทุนการผลิตซึ่ง เป็นส่วนสำคัญต่อความสำเร็จของบริษัท จากการสำรวจข้อมูลของสถาบันเพิ่มผลผลิตแห่งชาติ พบว่าอุตสาหกรรมเฟอร์นิเจอร์ไม้มีต้นทุนการผลิตทั้งหมดร้อยละ 73 และต้นทุนการขายและบริหาร ร้อยละ 27 โดยต้นทุนการผลิตทั้งหมดแบ่งเป็นต้นทุนวัตถุดิบมีมูลค่าร้อยละ 39 วัสดุสิ้นเปลือง ร้อยละ 6 พลังงาน ร้อยละ 4 ค่าใช้จ่ายในการซ่อมแซมเครื่องจักรประจำปี และค่าเสื่อมราคาโรงงาน เครื่องจักรและอุปกรณ์ร้อยละ 3 ค่าจ้างเหมาจ่ายหน่วยงานภายนอกผลิตร้อยละ 1 และอื่นๆ ร้อยละ 7 เมื่อพิจารณาโครงสร้างของต้นทุนแล้ว จะเห็นว่าต้นทุนวัตถุดิบมีสัดส่วนที่สูง ดังนั้นการวิจัยเพื่อเพิ่มผลผลิตของวัตถุดิบจึงมีผลมากต่อการเพิ่มผลผลิตโดยรวมของการผลิตเฟอร์นิเจอร์

1.2 วัตถุประสงค์ของโครงการวิจัย

โครงการวิจัยนี้มีวัตถุประสงค์เพื่อหารูปแบบการจัดวางชิ้นงานบนแผ่นไม้บอร์ดขนาดมาตรฐาน โดยนำคอมพิวเตอร์มาเป็นเครื่องมือช่วย เพื่อให้ได้ผลลัพธ์ที่มีรูปแบบการจัดวางชิ้นงานที่เหมาะสมกับเงื่อนไขของสถานะการผลิตงานเฟอร์นิเจอร์ไม้มากที่สุดและใช้เวลาการออกแบบลดลง

1.3 ขอบเขตของโครงการวิจัย

1. ศึกษารวบรวมข้อมูลเงื่อนไขต่างๆที่ใช้ในออกแบบการจัดวางชิ้นงานบนแผ่นไม้ MDF (Medium Density Fiberboard) ขนาดมาตรฐาน
2. พัฒนารูปแบบการจัดวางชิ้นงานบนแผ่นไม้ MDF ซึ่งมีรูปแบบการจัดวางแบบ 2 มิติ ตามส่วนประกอบของผลิตภัณฑ์ ที่ต้องการผลิต เพื่อให้มีการใช้วัตถุดิบไม้ MDF โดยรวมน้อยที่สุด
3. การกำหนดรูปแบบการตัดไม้รูปสี่เหลี่ยมแบบ 2 มิติ โดยใช้คอมพิวเตอร์ช่วยในการคำนวณ

ข้อมูลเบื้องต้นเกี่ยวกับสถานประกอบการที่ร่วมโครงการ

บริษัท	พี.พี.พาราวูด จำกัด
ที่ตั้ง	111/1 หมู่ 1 ตำบล บ้านช้าง อำเภอนันทนิคม จังหวัดชลบุรี 20140
ฝ่าย	วางแผนการผลิต (Production Planning Department)
สินค้าที่ผลิต	โรงงานผลิตเฟอร์นิเจอร์เครื่องเรือนจากไม้ยางพารา เพื่อการส่งออกและจำหน่ายในประเทศ

1.4 ขั้นตอนการดำเนินงานวิจัย

1. การศึกษาปัญหาในงานที่ผลิตและส่วนที่เกี่ยวข้อง เก็บข้อมูลเพื่อนำมาวิเคราะห์
2. สํารวจงานวิจัยและสร้างตัวแบบพร้อมทั้ง เสนอแนวทางในการปรับปรุงเบื้องต้น
3. กำหนดตัวแบบการตัดไม้ด้วยคอมพิวเตอร์ และทดสอบผลการทดลอง
4. ปรับปรุงพัฒนาตัวแบบและ โปรแกรมการจัดวางชิ้นงานไม้บอร์ด
5. วิเคราะห์และเปรียบเทียบผลในเชิงปริมาณและต้นทุนที่เกิดขึ้น
6. สรุปและทำรายงานผลการศึกษาวิจัย

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. ช่วยเพิ่มประสิทธิภาพการทำงานในการออกแบบการจัดวางชิ้นงาน โดยสามารถช่วยลดระยะเวลาและลดความผิดพลาดที่เกิดจากการค้นหาลักษณะการจัดวางชิ้นงานลงบนแผ่นไม้บอร์ดที่เหมาะสม
2. ช่วยให้ผู้ใช้งานสามารถตัดสินใจเลือกรูปแบบการจัดวางที่เหมาะสมกับสถานะการผลิตได้ถูกต้องรวดเร็วขึ้น เนื่องจากในสถานะจริงจะมีปัจจัยต่างๆและเงื่อนไขมากซึ่งโปรแกรมที่เขียนจะมีความยืดหยุ่นให้ผู้ใช้งานเป็นผู้ตัดสินใจในการเลือกรูปแบบในการวางชิ้นงาน
3. เป็นแนวทางให้ผู้พัฒนาโปรแกรมนี้ต่อไปได้มีความเข้าใจถึงวิธีการ รูปแบบข้อกำหนดต่างๆ ของรูปแบบปัญหา 2 มิติ (Two-dimension cutting stock)

โครงสร้างของโครงการวิจัย

โครงการวิจัยฉบับนี้ได้แบ่งเนื้อหาออกเป็นบทต่างๆ จำนวน 6 บท ดังต่อไปนี้

บทที่ 1 ความสำคัญในการดำเนินการวิจัย โดยใช้แนวทางหาคำตอบที่ดีที่สุด (Optimization) เพื่อช่วยในการแก้ไขปัญหาที่เกิดขึ้นในภาคอุตสาหกรรม ตลอดจนวัตถุประสงค์ วิธีการดำเนินงาน และประโยชน์ที่ได้รับจากโครงการวิจัย

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ตัวแบบทางคณิตศาสตร์ การกำหนดพารามิเตอร์เพื่อใช้ในการแก้ปัญหาทางด้าน Two-dimension cutting stock ที่นำมาใช้ในงานวิจัยนี้

บทที่ 3 วิธีการดำเนินงานวิจัยซึ่งจะกล่าวถึงการศึกษาสภาพปัจจุบันของโรงงานในส่วนของการผลิตที่เกี่ยวข้อง ตลอดจนการนำเอาตัวแบบทางคณิตศาสตร์ที่จำลองแบบไปประยุกต์ใช้กับโปรแกรมทางคอมพิวเตอร์

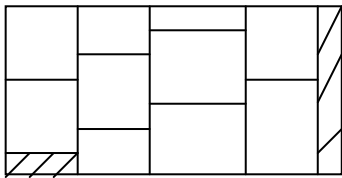
บทที่ 4 ผลการดำเนินงานวิจัย ทดสอบผลการทดลอง พร้อมทั้งปรับปรุงพัฒนาตัวแบบทางคณิตศาสตร์และโปรแกรมการจัดวางชิ้นงานไม้บอร์ด วิเคราะห์และเปรียบเทียบผลในเชิงปริมาณ และต้นทุนที่เกิดขึ้น

บทที่ 5 สรุปผลการดำเนินการและข้อเสนอแนะในการดำเนินการวิจัย

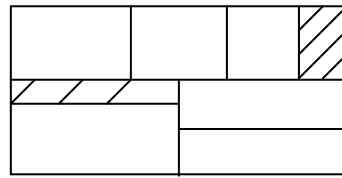
บทที่2 งานวิจัยที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

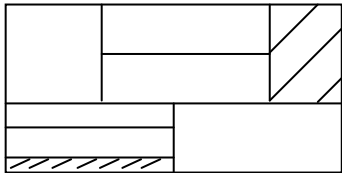
ปัญหาการตัดแบ่งวัสดุแบบ 2 มิติ (Two-Dimensional Cut ,TDC) เป็นปัญหาที่เกิดขึ้นในหลายอุตสาหกรรม เช่น แก้ว เหล็ก เสื้อผ้า กระดาษ และ ไม้ กว่า 60 ปีที่ได้มีการศึกษาปัญหาดังกล่าวโดยมีจุดประสงค์ที่จะมีกระบวนการแก้ปัญหาอย่างไรที่ทำให้เกิดการสูญเสียจากการตัดน้อยที่สุด ลักษณะของรูปแบบการจัดวางชิ้นงานการตัดนั้นจะมีอยู่ 2 ลักษณะคือ Orthogonal และ Non-Orthogonal ดังรูป(2.1) ซึ่งในปัญหาการตัดในลักษณะ Orthogonal ที่พบในอุตสาหกรรมแก้ว เหล็ก และไม้ ซึ่งรูปแบบการตัดแบ่งออกได้อีกคือ Guillotine และ Non-Guillotine ดังรูป(2.2) การแก้ปัญหา TDC พัฒนามาจากปัญหาการตัดวัสดุแบบมิติเดียว(One-dimensional Cutting Stock Problems) 1960 โดย Gilmore และ Gomory (1963)



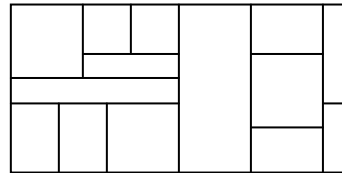
1 การตัดแบบกิโยติน 2 ชั้นตอนไม่เต็ม



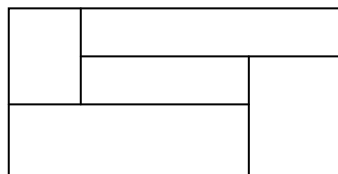
2 การตัดแบบกิโยติน 2 ชั้นตอนเต็ม



3 การตัดแบบกิโยติน 3 ชั้นตอน

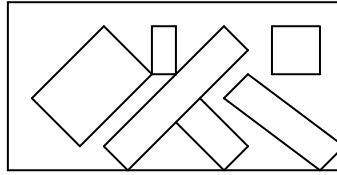


4 การตัดแบบกิโยตินแบบทั่วไป



5 การตัดแบบนอน-กิโยติน

รูปที่ 2.1 แสดงรูปแบบ Orthogonal ของการจัดวางวัสดุ



รูปที่ 2.2 แสดงรูปแบบ Non-Orthogonal ของการจัดวางวัสดุ

Dyson และ Gregory (1974) นำวิธีการของ Gilmore และ Gomory (1965) ไปประยุกต์ใช้กับอุตสาหกรรมกระจก โดยผสมผสานระหว่างทฤษฎีกำหนดการเชิงเส้น (Linear Programming) กับ Heuristic Scheduling ซึ่งสามารถลดของเสียจากการตัดไม้ได้เป็นอย่างมาก โดยสร้างปัญหาการตัดกระจกสี่เหลี่ยมมุมฉากขนาดใหญ่ออกเป็นกระจกสี่เหลี่ยมมุมฉากขนาดย่อยๆ ให้เหมาะสมกับกระบวนการผลิตและปริมาณความต้องการสินค้าในแต่ละขนาด กรณีศึกษาเป็นอุตสาหกรรมกระจกที่มีกระจกขนาดใหญ่ 2-3 ขนาด และมีขนาดที่ลูกค้าต้องการถึง 100-5000 ขนาด ซึ่งพิจารณาการสูญเสียเนื่องจากการตัดเป็นสำคัญ ด้วยวิธีกำหนดการเชิงเส้นและ Heuristic Scheduling ซึ่งมีความซับซ้อนมากกว่าปัญหาการตัดวัสดุโดยทั่วไป เนื่องจากเป็นปัญหาของรูปแบบในการตัด (The Pattern Allocation Problem) ที่จะต้องสอดคล้องกับการสูญเสียเนื่องจากการตัด และแผนลำดับในการตัดด้วย ต่อมา Dyson และ Chambers (1976) จึงทำการศึกษาต่อในลักษณะของปัญหาในอุตสาหกรรมกระจกเช่นเดิม โดยเพิ่มเติมในการเลือกขนาดของการตัดที่ต้องการ ด้วยวิธี Integer Programming ผสมกับวิธี Heuristic ซึ่งสามารถลดการสูญเสียของวัสดุได้มากกว่าเดิม

M.Adamowicz และ A.Albano ได้ทำการวิจัยเรื่อง “ A solution to the rectangular cutting stock problem” (1976) โดยเสนอ Two-stage Algorithms ขั้นแรกระบุถึงกลุ่มของรูปทรงชิ้นงาน ขั้นที่สองคือ การจัดวางชิ้นงานที่มีรูปร่างแบบรูปทรงสี่เหลี่ยมผืนผ้า(Rectangular Shape) ซึ่งจัดวางโดยใช้วิธี Muti-stage Dynamic Programming ประสิทธิภาพที่ได้รับจากการจัดวางด้วยมือ

Golden (1976) ทำการสรุปและรวบรวมวิธีการในปัญหาการตัดวัสดุ (Cutting Stock Problem) ในอดีตที่ผ่านมา โดยมุ่งพิจารณาที่วิธีการและเครื่องมือที่นำมาใช้เพื่อแก้ปัญหา ซึ่งมีรายละเอียดเกี่ยวกับตัวอย่าง Column Generation Zero-One Approach Combinatorial Bin Packing Heuristics และ Sub-Gradient Optimization รวมทั้งแสดงผลที่ได้ทดลองใช้วิธีดังกล่าวด้วยโปรแกรมคอมพิวเตอร์

Christofides และ C. Whitlock ได้ทำการวิจัยเรื่อง "An algorithms for two-dimensional cutting problems" (1977) โดยเสนอ A Tree Search Algorithm สำหรับใช้กับการตัดชิ้นงานรูปร่างแบบรูปทรงสี่เหลี่ยมผืนผ้า (Rectangular Shape) โดยอัลกอริทึมนี้จะเป็นตัวควบคุมจำนวนชิ้นงานแบบลงค์ แต่ละรูปร่างที่วางได้มากที่สุด แต่สามารถนำไปใช้กับการตัดแบบ Guillotine-type Cut ได้เพียงอย่างเดียว โดยอัลกอริทึมนี้มีข้อเสียคือ เมื่อใช้จัดวางชิ้นงานขนาดใหญ่จะใช้เวลาในการประมวลผลผ่านคอมพิวเตอร์นาน

Cani (1978) ทำการศึกษาการเปรียบเทียบรูปแบบในการตัดวัสดุสี่เหลี่ยม 2 มิติใน 2 รูปแบบ คือ รูปแบบในการตัดในแนวตรง (Orthogonal Cutting Pattern) ดังภาพที่ 2.1 (1-5) และรูปแบบในการตัดแนวเอียง (Non-Orthogonal Cutting Pattern) ดังภาพที่ 2.2 ซึ่งอธิบายด้วยเรขาคณิตและพิสูจน์ด้วยตรีโกณมิติ ซึ่งสามารถแสดงได้ว่ารูปแบบในการตัดในแนวเอียงนั้นสามารถลดการใช้พื้นที่วัสดุโดยรวมลงได้ หรืออีกนัยหนึ่งสามารถลดการสูญเสียวัสดุลงได้

Haessler (1980) ทำการปรับปรุงวิธีการแก้ปัญหาเดิมของ Gilmore และ Gomory (1965) โดยการเปลี่ยนจุดเริ่มต้นของการประมวลผล ด้วยการกำหนดรูปแบบของการตัดก่อนล่วงหน้า เพื่อที่จะลดขนาดของปัญหาลง แต่ก็ยังใช้เวลาในการประมวลผลด้วยคอมพิวเตอร์ที่มีมากกว่าวิธีการ Knapsack Problem ซึ่งต่อมา Haessler ได้ร่วมกับ Sweeney (1991) ทำการรวบรวมการสร้างปัญหาพื้นฐานในการตัดวัสดุ (Cutting Stock Problems) และวิธีการในการแก้ปัญหาการตัดวัสดุมิติเดียวและ 2 มิติด้วยวิธีกำหนดการเชิงเส้น วิธี Heuristic และการผสมผสานทั้ง 2 วิธีเข้าด้วยกัน จนถึงปี 1991 พบว่าในการแก้ปัญหาการตัดวัสดุมิติเดียวนั้น จะใช้วิธีกำหนดการเชิงเส้น วิธี Sequential Heuristic (SHP) และผสมทั้ง 2 วิธี สำหรับการแก้ปัญหาการตัดวัสดุ 2 มิตินั้น จะเป็นการตัดแบบกิโยติน (Guillotine Cutting) โดยใช้วิธี Dynamic Programming

Dyckhoff (1981) พัฒนาวิธีการแก้ปัญหาการตัดวัสดุของ Gilmore และ Gomory (1965) โดยใช้วิธีการในทฤษฎี Dynamic Programming ในการพิจารณาหารูปแบบการตัดวัสดุมิติเดียว และยังได้นำเสนอให้นำวิธีการนี้ไปใช้กับการตัดวัสดุหลายมิติต่อไป ในเวลาต่อมา Dyckhoff (1990) ทำการรวบรวมปัญหาการตัดและการบรรจุวัสดุ (Cutting and Packing Problems) ในอดีตที่ผ่านมาจนถึงปี 1990 เพื่อที่จะสร้างรูปแบบมาตรฐานของปัญหาลักษณะนี้ ที่สามารถจำแนกกลุ่มได้ชัดเจนขึ้น โดยใช้สัญลักษณ์ 4 ลำดับอักษร (X/X/X/X) ที่บ่งบอกถึง รูปแบบขนาดของมิติ การกำหนดชนิดลักษณะของวัสดุใหญ่และลักษณะของวัสดุเล็ก

S.S Israni และ J.L Sander ได้ทำการวิจัยเรื่อง “Performance testing of rectangular part-nesting heuristic”(1985) โดยพัฒนาวิธีการตรวจสอบเป้าหมายของอัลกอริทึม สำหรับการจัดวางชิ้นงานแบบไม่จำกัดรูปทรง (Irregular) โดยใช้วิธี Fractional Factorial Design และออกแบบตัวสุ่มพิเศษสำหรับชิ้นงานแบบไม่จำกัดรูปทรง (Irregular) สำหรับใช้ทดสอบความไวของอัลกอริทึมที่มีตัวแปรหลายตัวเช่น ขนาดของชิ้นงาน ลักษณะของชิ้นงานและจำนวนของชิ้นงานในรายการชิ้นส่วนประกอบ (Bill of Material)

C.H Dagli และ Y.Tatoglu ได้ทำการวิจัยเรื่อง “An approach to two dimensional cutting stock problem”(1987) ได้เสนอ Two-stage Hierarchy ที่เกี่ยวข้องกับปัญหาหระยะเพื่อของชิ้นงานป้อนขึ้นรูปที่มีรูปร่างแบบไม่จำกัดรูปทรง (Irregular) ใน Multi-plate 2 มิติ โดยขั้นแรกจะเกี่ยวข้องกับการสร้างระยะเพื่อตั้งต้นของแบบชิ้นงาน รวมทั้งสร้างโปรแกรมทางคณิตศาสตร์ ขั้นที่สองจะเกี่ยวข้องกับการสร้างรายละเอียดของระยะเพื่อ 2 มิติ ตลอดจนการสร้างอัลกอริทึมทางฮิวริสติก

Y.K.D.V. Prasad ได้ทำการวิจัยเรื่อง “ A set of heuristic algorithm for optimal nesting of the two-dimensional irregularly shaped sheet-metal blank”(1995) ซึ่งเกี่ยวข้องกับการพัฒนา Intelligent Nesting Algorithms สำหรับจัดวางชิ้นงานป้อนขึ้นรูปที่มีขนาดใหญ่เล็กไม่เท่ากัน สำหรับอุตสาหกรรมโลหะแผ่น โดยสร้างวิธีการจัดวางชิ้นงานที่มีลักษณะแตกต่างกัน 2 ชิ้นงานบนแผ่นวัสดุเดียวกัน MPSR (Multiple Product Single Row) ชื่อว่า Sliding Technique มีผลให้หน่วยความจำที่ใช้ในการเก็บข้อมูลน้อยลง และให้ค่าสัดส่วนการใช้ประโยชน์ (Utilization Ratio) ดีขึ้น แต่ประสิทธิภาพของระบบยังต้องปรับปรุงเพราะเวลาที่คอมพิวเตอร์ใช้ประมวลผลยังใช้เวลานานเกินไป

Van-Dat Cung,Mhand Hifi,Bertrand Le Cun (2000)ได้ทำการวิจัยเรื่อง “Constrained Two-dimensional cutting stock problems a best-first branch-and-bound algorithm” โดยการพัฒนาปรับปรุงอัลกอริทึมของ Hafi ในปี(1977) เสียใหม่เพื่อช่วยในการแก้ปัญหา Constrained Two-Dimensional Cutting โดยอาศัย Generic Branch and Bound

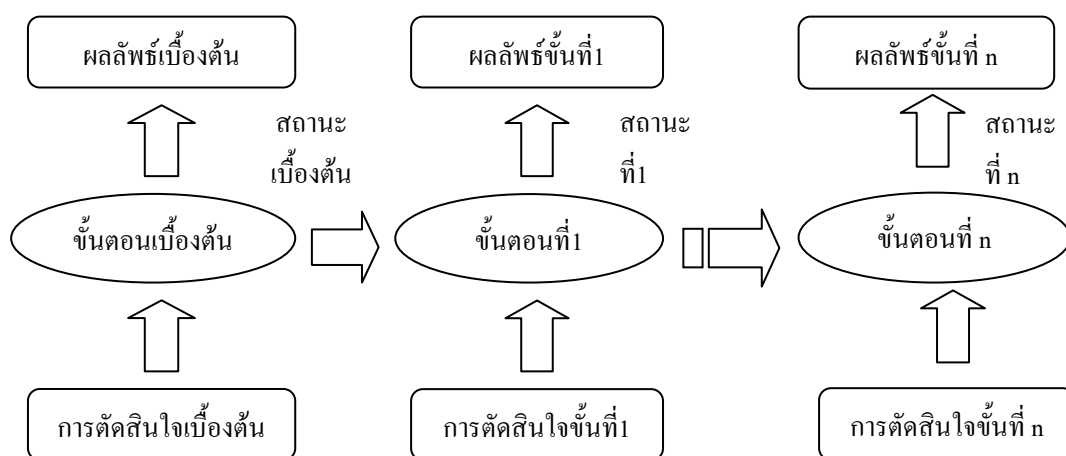
Ramon Alvarez-Valdes,Antonio Parajon และ Parajon,JosZ Manuel Tamarit ได้ทำการวิจัยเรื่อง “A Tabu Search Alogorithm for Large- Scale Guillotine(un)Constrained Two-Dimentional Cutting Problems” (2000) โดยการพัฒนาจากฮิวริสติก อัลกอริทึม เพื่อใช้แก้ปัญหา TDC (Two-Dimentional Cutting) นอกจากนั้นยังได้พัฒนา GRASP(Greedy Adaptive Search Procedure) อัลกอริทึม เพื่อช่วยในการแก้ปัญหามาตรฐานใหญ่

T.W.leung,C.H. Yung,Mavin D.Troutt (2001) ได้ทำการวิจัยเรื่อง “ Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem” โดยการปรับปรุง Genetic อัลกอริทึม และการ Simulate ปัญหาบน-กิโลตัน และการเปรียบเทียบประสิทธิภาพและประสิทธิผลของ อัลกอริทึม ทั้ง 2

สำหรับปัญหาที่มีขนาดใหญ่จะใช้ Heuristic algorithm ในการแก้ปัญหาและมีการพัฒนาอย่างต่อเนื่อง (Morabito et al used depth-first search and hill climbing strategies,producing algorithm สำหรับปัญหาที่มีลักษณะ Unconstrained case, DH.Fayard and Zissimopoulos ได้ออกแบบอัลกอริทึมเพื่อแก้ปัญหา Knapsack โดยใช้ dynamic programming (algorithm KD) หลังจากนั้น Hafi ได้เสนอการผสมผสานอัลกอริทึม DK/KD เข้าด้วยกันเพื่อให้แก้ลักษณะปัญหาที่มีขนาดใหญ่ได้ดีขึ้น เพื่อให้แก้ปัญหาได้ซับซ้อนและรวดเร็วขึ้น Raamon Alvarez-Vaides, Antonio Parajoh, Jose Manuel Tamarit ได้เสนอการผสมผสานการใช้ GRASP(Greedy Adaptive Search Procedure) อัลกอริทึมและTabu search ช่วยในการแก้ปัญหา TDC

ในงานวิจัยนี้ได้เลือกการหารูปแบบการจัดวางชิ้นงานที่เป็นไปได้ซึ่งเป็นการจัดวางชิ้นงานรูปสี่เหลี่ยมขนาดเล็กลงไปในวัสดุรูปสี่เหลี่ยมขนาดใหญ่ด้วยวิธีการของการกำหนดการเชิงพลวัต (Dynamic- Programming) โดยการกำหนดสถานะ การกำหนดทางเลือก การกำหนดพื้นที่ที่เหลือ และการตัดสินใจในผลลัพธ์ที่ให้มูลค่าโดยรวมสูงสุด

ปรัชญาของการกำหนดการเชิงพลวัต (Dynamic Programming) นั้น Bellman (1957) ซึ่งได้กล่าวไว้ว่าผลลัพธ์ที่ดีที่สุดนั้นมีผลมาจาก สถานะและการตัดสินใจในเบื้องต้น (Initial State and Decision) ซึ่งจะได้ผลลัพธ์ของการตัดสินใจก่อนหน้านี้ดังกล่าว แล้วจะส่งผลไปยังสถานะและการตัดสินใจในขั้นตอนถัดไป ดังนั้นจึงสามารถจำลองแบบปรัชญาดังกล่าวได้ดังรูปที่ 2.3

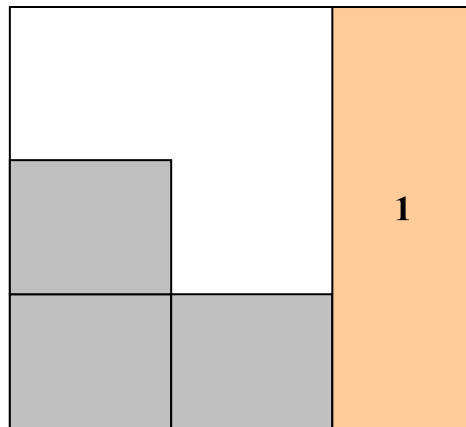


รูปที่ 2.3 แสดงแบบจำลองปรัชญาของการกำหนดการเชิงพลวัต (Dynamic Programming)

การหารูปแบบการจัดวางชิ้นงานการตัดที่เป็นไปได้จะอาศัยวิธีการแบบลงล่าง-ชิดซ้าย (BL-Bottom Left) โดยการกำหนดสถานะ การกำหนดทางเลือก การกำหนดพื้นที่ที่เหลือและการตัดสินใจในผลลัพธ์ที่ให้มูลค่าโดยรวมสูงสุดดังนี้

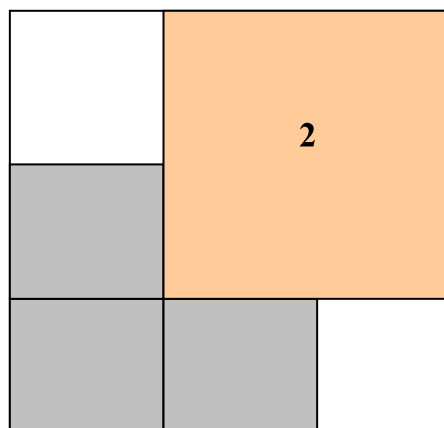
1. กำหนดสถานะ (State) เป็นพื้นที่ที่ต้องการตัดของทุกคู่ลำดับ X, Y
2. กำหนดทางเลือก (Alternative) เป็นจำนวนสี่เหลี่ยมขนาดเล็กที่ตัดได้
3. กำหนดพื้นที่สี่เหลี่ยมที่เหลือ (Available Areas) แบ่งออกเป็น 3 ส่วน ดังแสดงในรูปที่ 2.4 2.5 และ 2.6

1. พื้นที่สี่เหลี่ยมที่เหลือด้านขวา (Available Right Area)



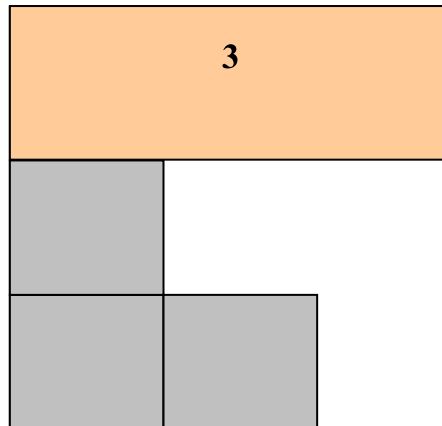
รูปที่ 2.4 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านขวา

2. พื้นที่สี่เหลี่ยมที่เหลือด้านขวา-บน (Available Top-Right Area)



รูปที่ 2.5 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านขวา-บน

3. พื้นที่สี่เหลี่ยมที่เหลือด้านขวา-บน (Available Top Area)



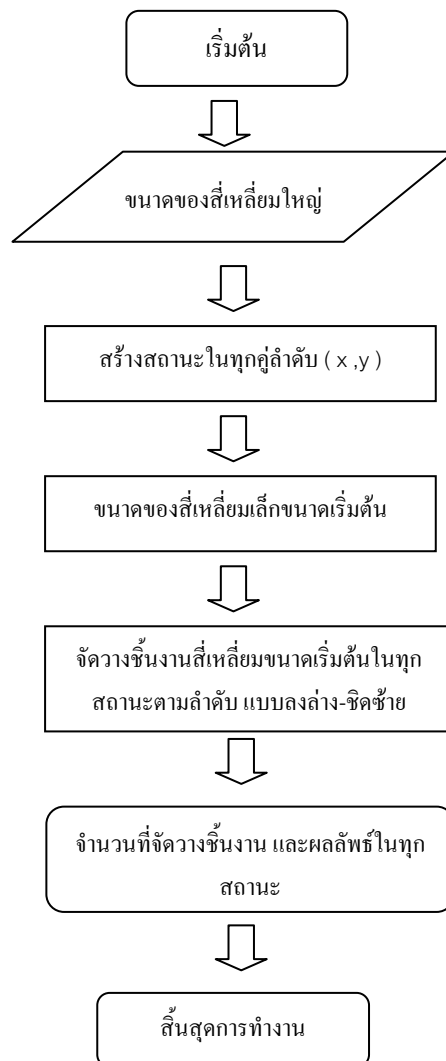
รูปที่ 2.6 แสดงพื้นที่สี่เหลี่ยมที่เหลือด้านบน

4. ผลลัพธ์ (Return) เป็นผลรวมมูลค่าของสี่เหลี่ยมขนาดเล็กที่สามารถตัดได้ในแต่ละทางเลือกรวมกับมูลค่าของสี่เหลี่ยมขนาดเล็กที่สามารถตัดได้พื้นที่ที่เหลือทั้ง 3 ส่วนดังกล่าว ซึ่งเป็นผลลัพธ์ที่มีมูลค่าสูงที่สุด จากการตัดสินใจในขั้นตอนก่อนหน้านี้
5. การตัดสินใจ (Decision) เลือกผลลัพธ์ที่สูงที่สุดในแต่ละสถานะ

เมื่อได้สถานะที่เป็นไปได้ตามรูปแบบดังกล่าวทั้งหมดแล้วนั้น ก็จะนำสถานะนั้นๆ มาเป็นลำดับในการจัดวางชิ้นงานลงในพื้นที่สี่เหลี่ยมขนาดใหญ่ ด้วยรูปแบบการวางชิ้นงานแบบล่าง-ซ้าย (BL- Bottom Left) ดังกล่าวก็จะทำให้ได้สถานะ (State) ที่แท้จริงที่สามารถจัดวางชิ้นงานลงในรูปสี่เหลี่ยมขนาดใหญ่ได้ เมื่อถึงขั้นตอนนี้ก็จะได้สถานะของรูปแบบการจัดวางชิ้นงานที่เป็นไปได้ทั้งหมดด้วยวิธีการดังกล่าว เพื่อที่จะนำไปใช้ในลำดับต่อไป

การกำหนดสถานะในแต่ละขั้นตอน

ในขั้นตอนเริ่มต้น (Initial State) นั้น จำนวนของสถานะจะขึ้นอยู่กับขนาดและความละเอียดของหน่วยของขนาดของพื้นที่สี่เหลี่ยมขนาดใหญ่ รวมถึงลักษณะและวิธีการจัดเรียงชิ้นงาน ในขั้นตอนเริ่มต้นนี้จะมีเพียงสถานะ (State) ทางเลือก (Alternative) และผลลัพธ์ (Return) ของการตัดพื้นที่สี่เหลี่ยมขนาดเล็กขนาดที่ 1 เท่านั้น ซึ่งไม่จำเป็นต้องมีการตัดสินใจ (Decision) เนื่องจากผลลัพธ์ในทุกสถานะมีเพียงผลลัพธ์เดียวที่มีมูลค่าสูงสุดเท่านั้น ซึ่งสามารถแสดงด้วยแผนผังการไหลของโปรแกรมได้ดังรูปที่ 2.7



รูปที่ 2.7 แผนผังการไหลของโปรแกรมในขั้นตอนเริ่มต้นของรูปแบบการจัดวางชิ้นงานแบบกิโลตินตั้งตรง

บทที่3 วิธีการดำเนินงานวิจัย

เนื่องจากงานวิจัยนี้ได้พัฒนารูปแบบการจัดวางชิ้นงานบนแผ่นไม้บอร์ดขนาดมาตรฐาน แนวทางการวิจัยจึงเน้นหนักไปในเรื่องวิธีการแก้ปัญหาโดยอาศัยวิธีการทางอภิปรัชญาและการใช้แบบจำลองทางคณิตศาสตร์ พร้อมทั้งสร้างโปรแกรมสำเร็จรูป เพื่อสามารถนำไปใช้งานได้ ดังนั้นลักษณะวิธีการดำเนินงานวิจัยจึงเป็นลักษณะการคิดค้นตัวแบบทางคณิตศาสตร์และการดำเนินงานที่แบ่งออกเป็นขั้นตอนต่างๆ โดยมีการ ทดสอบ สรุปผล คัดเลือกจนกระทั่งได้รูปแบบการจัดวางชิ้นงานที่เหมาะสม โดยมีขั้นตอนการดำเนินงานซึ่งมีรายละเอียดดังต่อไปนี้

1. ศึกษาสภาพปัจจุบันของบริษัทที่เข้าร่วมโครงการ
2. การวิเคราะห์ข้อมูลที่ได้รวบรวมจากหน่วยการผลิตที่เกี่ยวข้อง
3. การสร้างตัวแบบทางคณิตศาสตร์
4. ประยุกต์การใช้งานตัวแบบทางคณิตศาสตร์ด้วยคอมพิวเตอร์
5. ทดสอบการทำงานของโปรแกรม

3.1 ศึกษาสภาพปัจจุบันของบริษัทที่ทำการวิจัย

การศึกษาศภาพกระบวนการวางแผนการผลิต เพื่อให้ทราบถึงขั้นตอนการดำเนินงานโดยเริ่มต้นตั้งแต่การรับคำสั่งซื้อจากลูกค้าจนถึงการออกไปจำหน่ายให้กับพนักงานประจำเครื่องตัดในหน่วยของงานตัดและงานเพรส โดยมีวิธีการดำเนินงาน ดังนี้

1. ศึกษาแผนผังการควบคุมการตัดไม้บอร์ด
2. ทำความเข้าใจเกี่ยวกับขั้นตอนการดำเนินงานในการควบคุมการตัดไม้บอร์ด
3. ศึกษาวิธีการจัดการและการวางแผนงานตัดที่ใช้อยู่ในปัจจุบัน
4. รวบรวมข้อมูลที่มีความจำเป็นต่อการพัฒนาโปรแกรมคอมพิวเตอร์

3.2 การวิเคราะห์ข้อมูลและการแก้ปัญหา

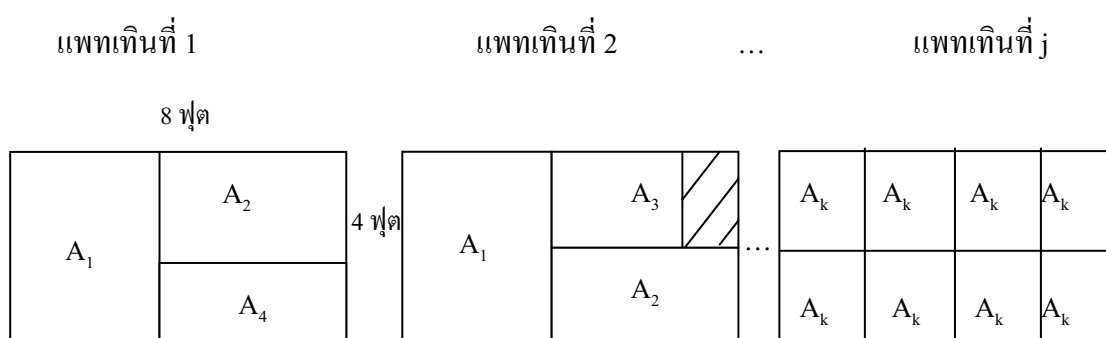
ลักษณะปัญหาที่นำมาศึกษา

เมื่อโรงงานรับคำสั่งซื้อจากลูกค้า ฝ่ายวางแผนการผลิตจะทำการแยกรายการชิ้นส่วนประกอบของผลิตภัณฑ์แล้วนำไปวางแผนการผลิต โดยที่ชิ้นส่วนแต่ละรายการมีขนาดและปริมาณที่แตกต่างกัน ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงตัวอย่างชิ้นส่วนที่ต้องการผลิต

ลำดับ	ความกว้าง	ความยาว	พื้นที่	ปริมาณที่ต้องการ
1	W_1	L_1	A_1	D_1
2	W_2	L_2	A_2	D_2
3	W_3	L_3	A_3	D_3
·	·	·	·	·
k	W_k	L_k	A_k	D_k
·	·	·	·	·
K	W_K	L_K	A_K	D_K

ลักษณะของปัญหาต้องการผลิตชิ้นส่วนที่ใช้วัสดุที่ทำจากไม้บอร์ด โดยที่ไม้บอร์ดขนาดมาตรฐานคือ 1220 มม. X 2440 มม. หรือ 4 ฟุต X 8 ฟุต โดยชิ้นงานแต่ละรายการมีขนาดความกว้างคือ W_k ความยาวคือ L_k และมีปริมาณความต้องการ คือ D_k ดังนั้นการที่จะผลิตรายการชิ้นส่วนดังกล่าวต้องใช้ไม้บอร์ดขนาดมาตรฐานจำนวนเท่าไร และ รูปแบบการจัดวางชิ้นงานเป็นอย่างไร แต่ละรูปแบบมีปริมาณเท่าไร ภายใต้เงื่อนไข คือ ใช้ปริมาณไม้บอร์ดขนาดมาตรฐานน้อยที่สุด เหลือเศษทิ้งน้อยที่สุด และเพียงพอต่อปริมาณความต้องการของแต่ละรายการ ซึ่งรูปแบบการจัดวางที่เป็นไปได้จะขึ้นอยู่กับจำนวนรายการและขนาดของชิ้นส่วนดังแสดงในรูปที่ 3.1



รูปที่ 3.1 แสดงรูปแบบการจัดวางชิ้นงานที่เกิดขึ้นจากชิ้นส่วนต่างๆที่ต้องการผลิต

สภาพปัจจุบันปัญหาที่เกิดขึ้นเป็นปัญหามหาศาลที่มีความซับซ้อนมาก ดังแสดงตัวอย่างในตารางที่ 3.2 ทำให้วิธีการที่จะค้นหารูปแบบการจัดวางชิ้นงานที่เหมาะสมกับปริมาณความต้องการผลิตเป็นไปอย่างลำบาก ซึ่งผลจากการออกแบบในแต่ละครั้งนั้นจะส่งผลกระทบต่อต้นทุนด้วย

ตารางที่3.2 แสดงรายการชิ้นส่วนที่ต้องการผลิต

ลำดับ	รหัสสินค้า	ชื่อชิ้นงาน	วัสดุ	ความหนา	ความกว้าง	ความยาว	จำนวนที่ต้องการ
1	102-229B	แผงข้าง	PB	18	337	670	104
2	102-229B	พื้นกันลื่นชัก	PB	18	380	800	104
3	102-229B	พื้นล่าง	PB	18	380	800	52
4	102-147B	แผงข้าง	PB	18	555	1455	104
5	102-147B	พื้นบน	PB	18	545	1190	52
6	102-147B	พื้นกลาง	PB	18	520	1190	52
7	102-147B	พื้นล่าง	PB	18	470	1190	52
8	102-147B	ชั้นกลาง	PB	18	470	1170	52
9	102-147B	ชั้นปรับระดับ#1	PB	18	510	1170	52
10	102-147B	ชั้นปรับระดับ#2	PB	18	510	1170	52
11	102-147B	ก๊วยบนตัวหลัง	PB	18	1210	1230	6
12	102-032B	แผงข้างตัวล่าง	PB	18	510	580	104
13	102-032B	ชั้นกลาง-ล่าง	PB	18	450	1200	104
14	102-032B	ชั้นบน	PB	18	490	1200	52
15	102-032B	ท็อป	PB	18	540	1200	52
16	102-042B	แผงข้าง	PB	18	400	900	104
17	102-042B	ชั้นกลาง	PB	18	400	1460	208
18	102-146B	แผงข้าง	PB	18	350	600	104
19	102-146B	พื้นล่าง	PB	18	600	1280	52
20	102-230B	แผงข้าง	PB	18	410	630	104
21	102-230B	ชั้นล่างกลาง	PB	18	380	990	104
22	102-230B	ชั้นกลาง	PB	18	400	990	52
23	102-230B	ท็อป	PB	18	440	1005	104
24	102-219B	แผงข้าง	PB	18	420	630	104
25	102-219B	ชั้นล่างกลาง	PB	18	380	990	104
26	102-219B	ชั้นกลาง	PB	18	400	990	52

3.3 การดำเนินการแก้ปัญหาการจัดวางชิ้นงาน

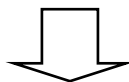
การนำเอาข้อมูลที่ได้จากการวิเคราะห์หามาพัฒนาตัวแบบทางทางคณิตศาสตร์ใหม่ เพื่อช่วยให้การวางแผนการจัดการมีประสิทธิภาพมากยิ่งขึ้น ซึ่งจะมีการผสมผสานกันระหว่างวิธีการทางฮิวริสติกและสมการทางคณิตศาสตร์ ซึ่งต้องอาศัยการแก้ปัญหาโดยแบ่งออกเป็นส่วนๆ(Multi -Phase) ด้วยกัน

การประมวลผลการหารูปแบบการจัดวางที่เหมาะสม

วิธีการแก้ปัญหาการจัดวางชิ้นงานที่ได้จำลองขึ้น โดยการประยุกต์ขั้นตอนต่างๆเข้าด้วยกัน โดยมีวิธีการดำเนินงานแบ่งออกเป็นขั้นตอนต่างๆ (Multi -Phase) ดังต่อไปนี้

ขั้นตอนที่ 1

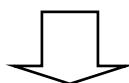
การหารูปแบบการจัดเรียงเริ่มต้นทั้งหมดที่เป็นไปได้
ได้จะใช้วิธีการทางฮิวริสติก คือ Bottom-Left



สิ่งที่ได้คือรูปแบบการจัดเรียงที่เป็นไปได้ทั้งหมด
ภายใต้เงื่อนไขคือจัดเรียงจนกระทั่งไม่สามารถ
ที่จะจัดเรียงต่อไปได้

ขั้นตอนที่ 2

การคัดเลือกรูปแบบการจัดเรียงเบื้องต้นก่อนนำไป
ประมวลผล โดยอาศัยวิธีการคัดเลือกจากพื้นที่การใช้
ประโยชน์สูงสุดที่เกิดขึ้นในแต่ละรอบของการ
ประมวลผลจากขั้นตอนที่ 1



สิ่งที่ได้คือรูปแบบการจัดเรียงจำนวนหนึ่ง ภายใต้
เงื่อนไขคือ การคัดเลือกจากการใช้ประโยชน์
สูงสุดที่เกิดขึ้นในแต่ละรอบ

ขั้นตอนที่ 3

การหารูปแบบการจัดวางชิ้นงานที่เหมาะสมกับ
ปริมาณความต้องการที่ต้องการผลิตโดยอาศัย
แบบจำลองคณิตศาสตร์

3.4 การหารูปแบบการจัดเรียงชิ้นงานเริ่มต้นทั้งหมดที่เป็นไปได้

การหารูปแบบในการตัดแบบกิโยตินตั้งตรงจะอาศัยแบบจำลองทางคณิตศาสตร์ผสมผสานวิธีการบรรจุแบบ (BL-Bottom Left algorithm) โดยมีเงื่อนไขคือต้องมีการจัดเรียงแบบกิโยติน (Guillotine Pattern) ดังนี้

สมการทางคณิตศาสตร์

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^+ W_{ij} \leq W_{ij} \quad \dots\dots\dots 1$$

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^- L_{ij} \leq L_{ij} \quad \dots\dots\dots 2$$

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^+ C_{ij}^- W_{ij} L_{ij} \leq W_{ij} L_{ij} \quad \dots\dots\dots 3$$

กำหนดให้

- C_{ij}^- คือจำนวนแถวของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j
- C_{ij}^+ คือจำนวนสดมภ์ของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j
- C_{ij} คือจำนวนชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j
- W_i คือ ความกว้างของ Pattern ที่ j
- L_i คือ ความยาวของ Pattern ที่ i

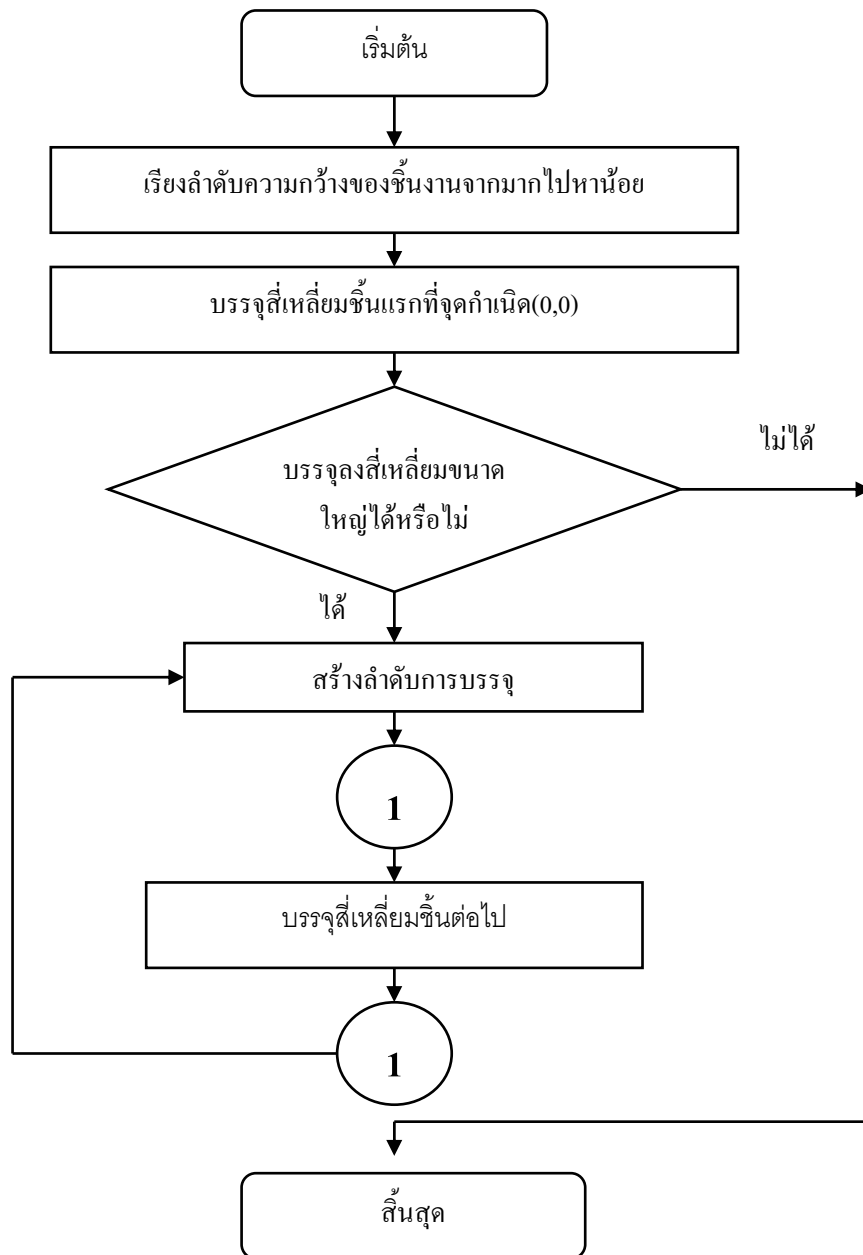
จากนั้นทำการสร้างโปรแกรมคอมพิวเตอร์จากโปรแกรม Visual basic 6 และ Excel Solver เพื่อหา รูปแบบการจัดวางชิ้นงานสี่เหลี่ยมมุมฉากขนาดย่อยๆลงในวัสดุสี่เหลี่ยมมุมฉากขนาดใหญ่ที่เป็นไปได้ (Feasible Patterns) มีขั้นตอนดังนี้

1. เริ่มต้นด้วยโปรแกรมหลัก โดยทำการเรียงลำดับชิ้นงานที่มีความกว้างมากที่สุดไปยังชิ้นงานที่มีความกว้างน้อยที่สุด
2. ทำการบรรจุชิ้นงานแรกที่จุดกำเนิด (0,0) ลงไปในสี่เหลี่ยมใหญ่มาตรฐาน แล้วทำการตรวจสอบว่าสามารถที่จะบรรจุสี่เหลี่ยมขนาดใหญ่ได้หรือไม่ ถ้าไม่สามารถบรรจุได้ก็สิ้นสุดโปรแกรม
3. ถ้าสามารถบรรจุได้ก็ทำการสร้างลำดับการบรรจุจากนั้นก็ทำการบรรจุสี่เหลี่ยมชิ้นต่อไป ซึ่งเป็นขั้นตอนการบรรจุที่ 1
4. เมื่อเสร็จสิ้นขั้นตอนการบรรจุก็จะกลับไปยังการสร้างลำดับการบรรจุและทำการบรรจุสี่เหลี่ยมชิ้นต่อไปจนกระทั่งสิ้นสุดการบรรจุ
5. เมื่อตรวจสอบได้ว่าไม่สามารถบรรจุลงสี่เหลี่ยมขนาดใหญ่ได้แล้ว ซึ่งเป็นการสร้างลำดับการบรรจุในแต่ละขนาด ดังแผนผังการไหลดังรูปที่ 3.2 3.3 3.4 และ 3.5

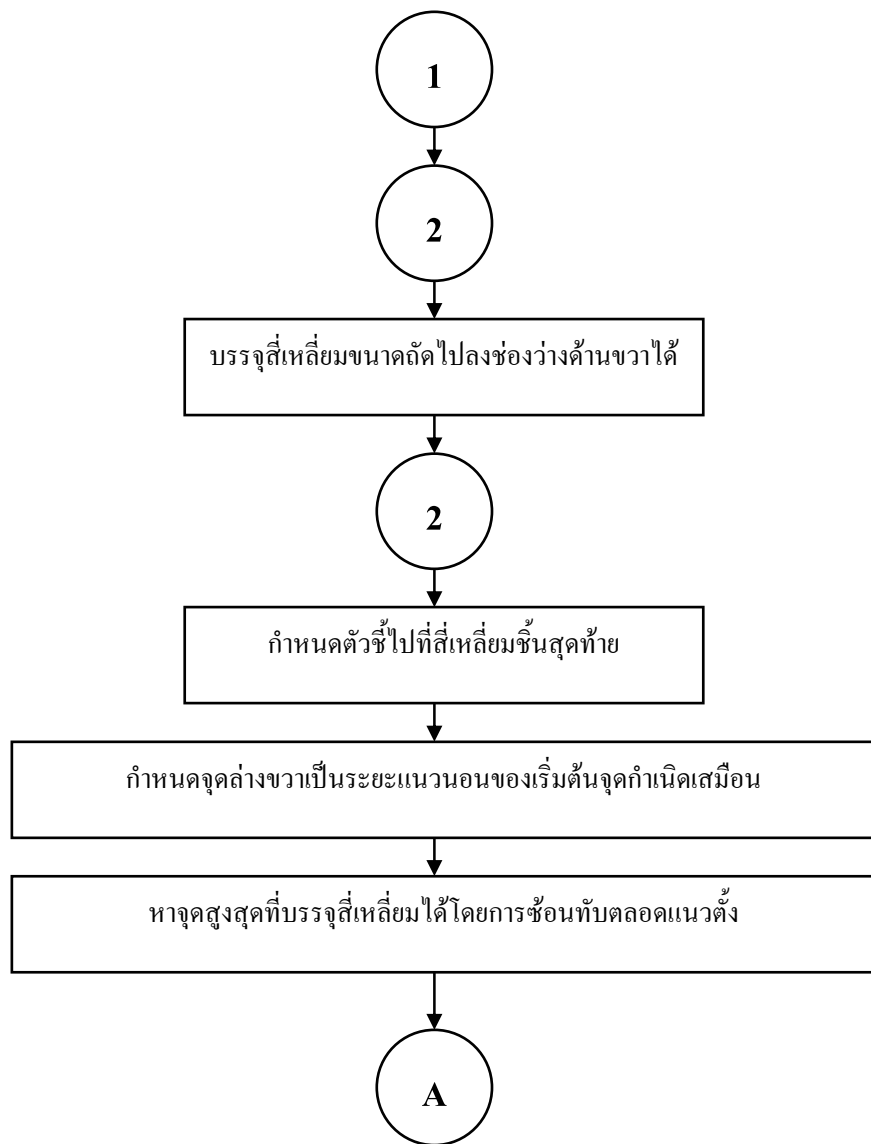
ขั้นตอนการบรรจุที่ 1 เป็นการบรรจุสี่เหลี่ยมชิ้นต่อไปนั้น เริ่มต้นด้วยการเข้าสู่ขั้นตอนการบรรจุที่ 2 ก่อน ซึ่งเป็นการบรรจุสี่เหลี่ยมขนาดเล็กชิ้นต่อไปในกรณีที่สามารถบรรจุลงในช่องว่างด้านขวาได้ จากนั้นทำการกำหนดตัวชี้ไปที่สี่เหลี่ยมที่บรรจุเป็นชิ้นสุดท้าย กำหนดจุดล่างขวาที่ระยะในแนวนอนเป็นจุดกำเนิดเสมือน (Virtual Origin) เริ่มต้น จากนั้นทำการหาจุดสูงสุดที่จะบรรจุสี่เหลี่ยมชิ้นต่อไปได้ ด้วยการหาจุดซ้อนทับกัน (Intersection) ตลอดแนวตั้ง ก็จะพบจุดที่จะทำการบรรจุในระยะแนวตั้ง แล้วจึงทำการตรวจสอบว่ามีระยะที่เหลืตามแนวนอนนั้นมีเพียงพอหรือไม่ ถ้ามีเพียงพอก็จะไปสู่กระบวนการบรรจุที่จุดกำเนิดเสมือนนั้นได้เลย แต่ถ้าเหลือไม่เพียงพอก็ต้องทำการบรรจุทางด้านบนต่อไป แล้วทำการตรวจสอบว่ามีระยะที่เหลืตามแนวตั้งมีเพียงพอหรือไม่ ถ้าเหลือไม่เพียงพอก็สิ้นสุดโปรแกรมการบรรจุ เนื่องจากไม่สามารถที่จะบรรจุได้แล้ว แต่ถ้ามีระยะเหลือเพียงพอตามแนวตั้ง จะต้องทำการเลือกว่าในแต่ละจุดของชิ้นงานก่อนหน้านั้น จุดใดเป็นจุดที่ไว้ระยะของแกนตั้งที่ต่ำที่สุด เนื่องจากการบรรจุแบบลงล่าง-ชิดซ้ายนั่นเอง ต่อจากนั้นทำการตรวจสอบว่าการบรรจุนั้นชิดขอบซ้ายหรือสี่เหลี่ยมด้านซ้ายแล้วหรือไม่ ถ้าชิดแล้วก็ทำการบรรจุได้เลย แต่ถ้ายังไม่ชิดก็ต้องหาจุดที่ต่ำที่สุดโดยการซ้อนทับตลอดแนวตั้งเพื่อหาจุดชิดขอบล่าง แล้วต้องวนกลับไปตรวจสอบการชิดขอบซ้ายดังแผนผังการไหลรูปที่ 3.6 จนกระทั่งสามารถบรรจุได้ เมื่อบรรจุได้แล้วก็ทำการตรวจสอบว่าสี่เหลี่ยมชิ้นที่บรรจุลงไปในนั้นเป็นชิ้นสุดท้ายด้านขวามือหรือไม่ เพื่อประโยชน์ในขั้นตอนการบรรจุที่ 2 กรณีที่สามารถบรรจุลงในช่องว่างด้านขวาได้ ถ้าไม่เป็นชิ้น

สุดท้ายด้านขวามือ ก็สิ้นสุดขั้นตอนการบรรจุที่1 แต่ถ้าเป็นสิ้นสุดท้ายด้านขวามือก็ทำการเก็บข้อมูล
สี่เหลี่ยมขึ้นด้านขวามือไว้ ก่อนการสิ้นสุดกระบวนการในขั้นตอนการบรรจุที่1

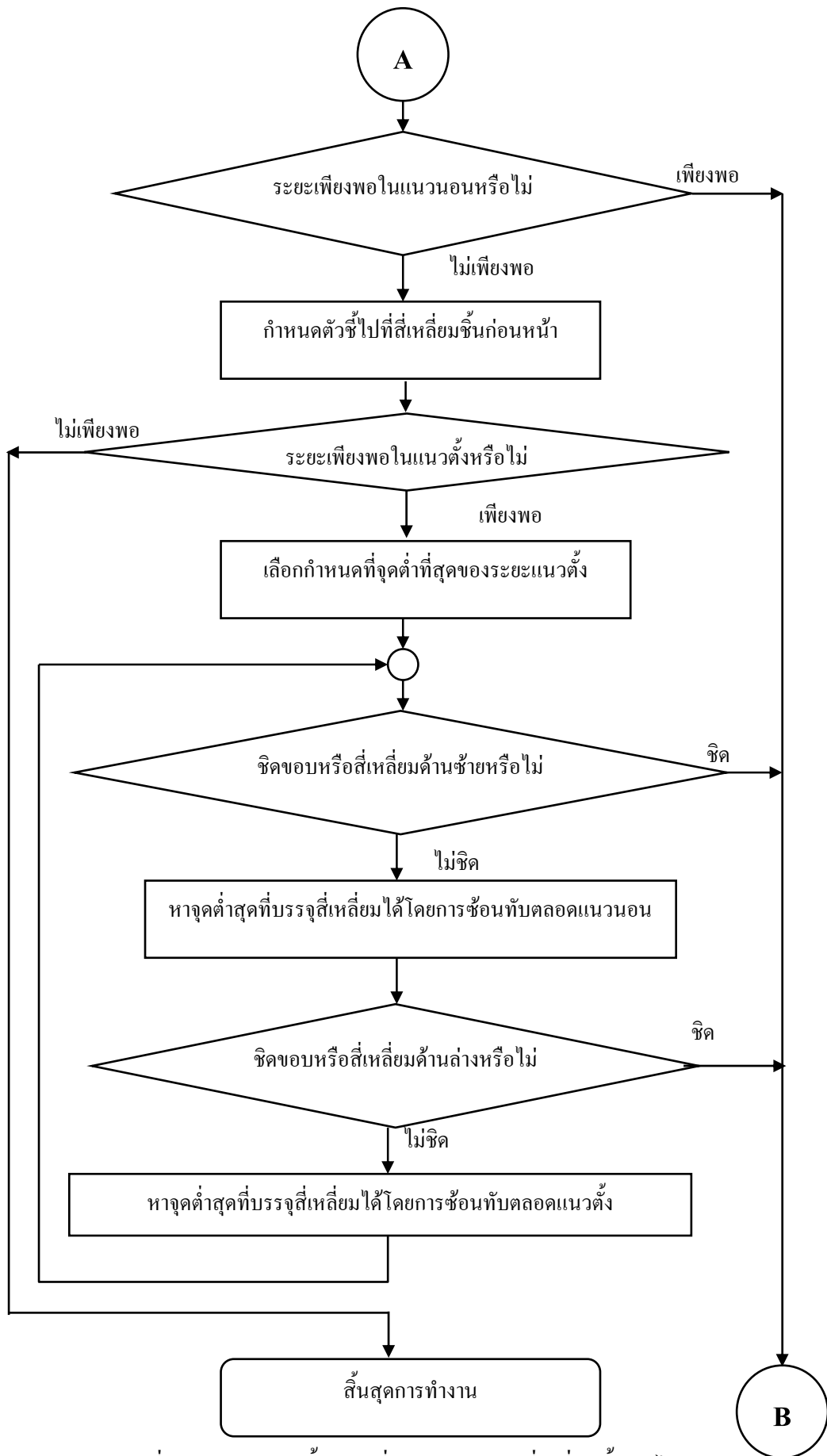
ขั้นตอนการบรรจุที่2 เป็นการบรรจุสี่เหลี่ยมกรณีที่สามารถบรรจุลงในช่องว่างด้านขวาได้ เริ่มต้น
ด้วยการตรวจสอบว่าสี่เหลี่ยมที่กำหนดตัวชี้ไปนั้นเป็นสิ้นสุดท้ายด้านขวาหรือไม่ โดยตรวจสอบใน
ลำดับค่าสี่เหลี่ยมขึ้นด้านขวาที่เก็บเอาไว้ นั่นเอง ถ้าไม่เป็นสิ้นสุดท้ายด้านขวาก็สิ้นสุดโปรแกรมย่อย
นี้ แต่ถ้าเป็นสิ้นสุดท้ายด้านขวาก็กำหนดตัวชี้ไปที่ขึ้นด้านขวาสิ้นสุดท้ายแล้วทำการตรวจสอบว่า
สี่เหลี่ยมที่จะบรรจุนั้นลงได้หรือไม่ ถ้าลงไม่ได้ก็สิ้นสุดโปรแกรมย่อยนี้เช่นกัน แต่ถ้าลงได้ก็ทำการ
หาจุดสูงสุดที่สามารถบรรจุสี่เหลี่ยมขึ้นนี้ได้ โดยการหาการซ้อนทับตลอดแนวตั้ง เมื่อพบจุดที่จะ
ทำการบรรจุก็ต้องทำการตรวจสอบว่า เมื่อบรรจุแล้วสี่เหลี่ยมขึ้นนั้นชิดกับสี่เหลี่ยมด้านซ้ายมือ
หรือไม่ ถ้าไม่ชิดก็ทำการหาจุดต่ำสุดตามแนวนอนที่จะทำการบรรจุสี่เหลี่ยมขึ้นนั้น ด้วยการหาการ
ซ้อนทับตลอดแนวนอน ก็จะพบจุดที่ชิดกับสี่เหลี่ยมด้านซ้ายหรือขอบด้านซ้ายก็ทำการบรรจุได้เลย
เช่นเดียวกันกรณีที่ไม่วัดชิดขอบซ้ายจะต้องทำการตรวจสอบว่าการบรรจุชิดขอบล่าง หรือสี่เหลี่ยม
ด้านล่างตามหลักการบรรจุแบบลงล่าง-ชิดซ้ายแล้วหรือไม่ ดังนั้นถ้าชิดขอบล่างแล้วก็ทำการการ
บรรจุได้เลย แต่ถ้าไม่ชิดจะต้องวนกลับไปหาจุดต่ำสุดที่สามารถบรรจุสี่เหลี่ยมขึ้นนี้ได้ โดยการหา
การซ้อนทับตลอดแนวตั้ง ดังแผนผังการไหลขั้นตอนการบรรจุที่2 จนกระทั่งสิ้นสุดกระบวนการ
ของขั้นตอนการบรรจุที่2นี้ ซึ่งแสดงได้ด้วยแผนผังการไหล ดังรูปที่ 3.6



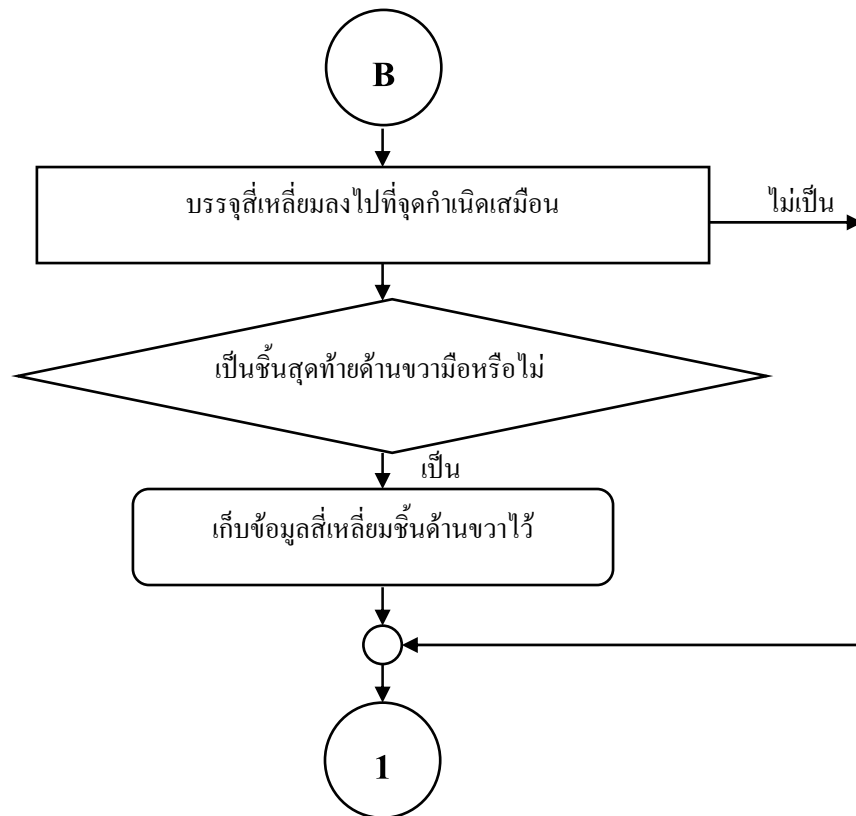
รูปที่3.2 แผนผังการไหลของขั้นตอนที่1ของการบรรจุสี่เหลี่ยมขนาดเล็กลงในสี่เหลี่ยมขนาดใหญ่



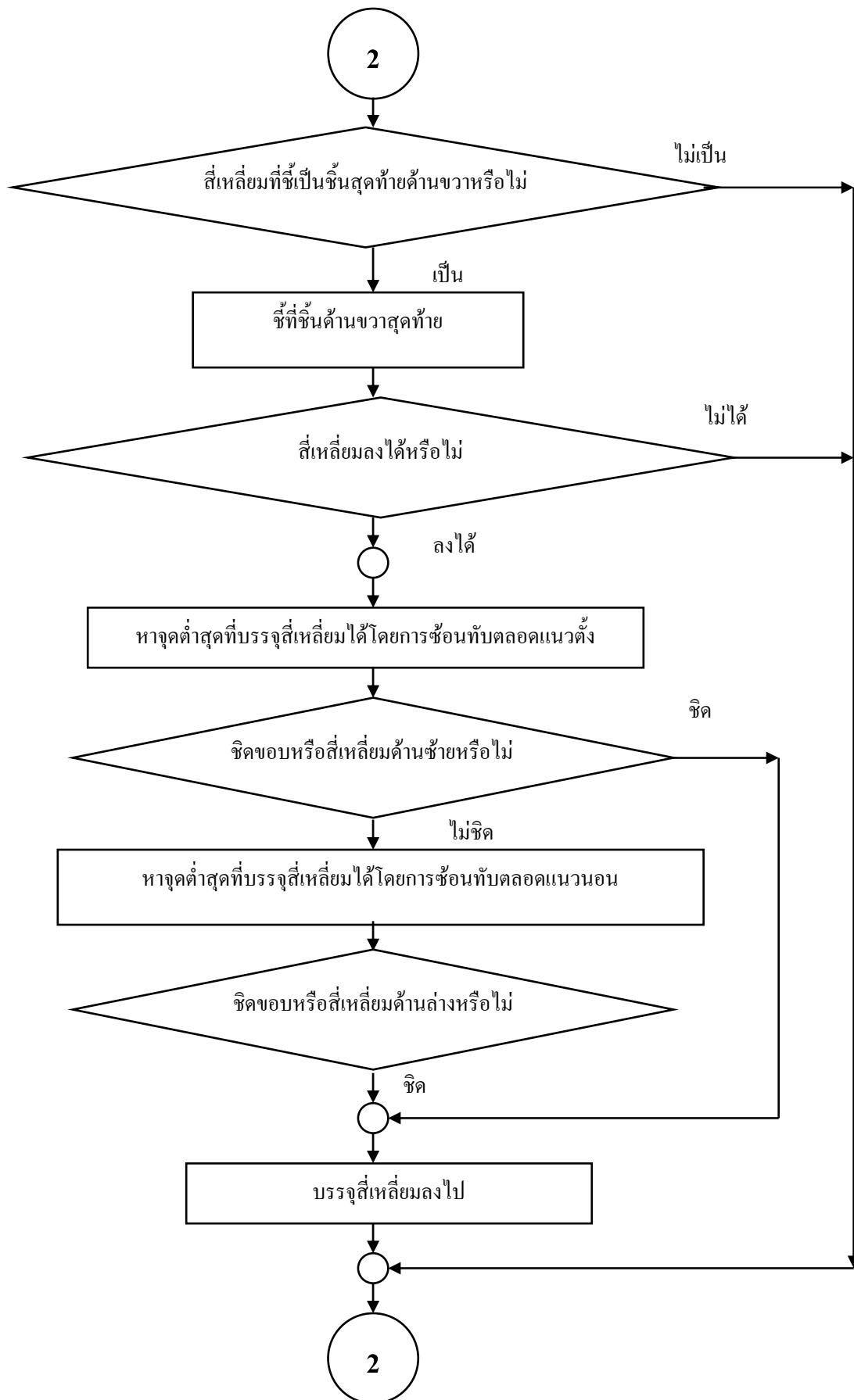
รูปที่ 3.3 แผนผังการไหลขั้นตอนที่1ของการบรรจุ ผู้ให้วัคซีนขึ้นไปตามลำดับ



รูปที่ 3.4 แผนผังการขั้นตอนที่1ของการบรรจุสี่เหลี่ยมขึ้นไป



รูปที่ 3.5 แผนผังการไหลขั้นตอนที่1ของการบรรจุสี่เหลี่ยมขึ้นไป

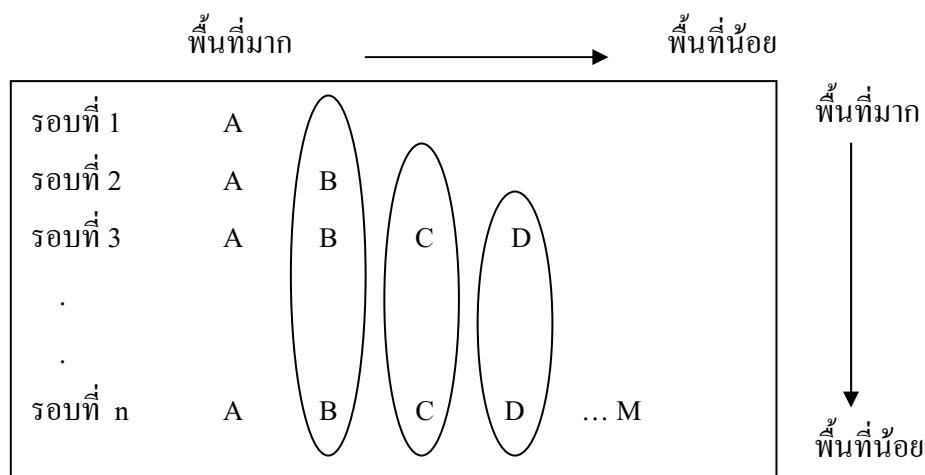


รูปที่ 3.6 แผนผังการไหลขั้นตอนที่ 2 การบรรจุกรณีที่บรรจุลงในช่องว่างด้านขวา

3.5 การคัดเลือกรูปแบบการจัดเรียงเบื้องต้นก่อนการประมวลผล

เนื่องจากมีข้อจำกัดในด้านตัวแปรในโปรแกรมร่วมที่ใช้ในการประมวลผล ดังนั้นจะต้องมีการคัดเลือกรูปแบบการจัดวางชิ้นงานเบื้องต้น ก่อนที่จะนำไปคำนวณในช่วงต่อไป โดยมีเงื่อนไขและหลักเกณฑ์ดังต่อไปนี้

1. เรียงลำดับรูปแบบการจัดเรียงที่มีพื้นที่การใช้ประโยชน์มากที่สุดในแต่ละรอบของการประมวลผล
2. เรียงลำดับรูปแบบการจัดเรียงที่มีพื้นที่การใช้ประโยชน์มากที่สุด โดยเรียงจากรอบการประมวลรอบแรกไปยังรอบสุดท้ายแล้ววนกลับมาเริ่มต้นใหม่ ดังรูป 3.7



รูปที่ 3.7 แสดงขั้นตอนการจัดเรียงข้อมูลเพื่อคัดเลือก

หลังจากได้ผ่านกระบวนการจัดเรียงแล้วจะนำเอาข้อมูลรูปแบบได้ประมวลผลต่อไป โดยคัดเลือก 180 ลำดับแรก เพื่อไปประมวลผลขั้นต้นในขั้นตอนต่อไป

3.6 การประมวลผลเพื่อหารูปแบบที่เหมาะสมกับการผลิต

การหาปริมาณการหาค่าปริมาณของรูปแบบการจัดวางชิ้นงานที่ใช้จริงโดยจะกำหนดรูปแบบการจัดวางชิ้นงานเริ่มแรกโดยการนำเอาสถานะที่ได้จากการคัดเลือกรูปแบบการจัดวางในขั้นตอนก่อนหน้านี้มาประมวลผลเริ่มต้น โดยแบบจำลองคณิตศาสตร์ดังนี้

สมการทางคณิตศาสตร์

$$\text{สมการเป้าหมาย} \quad \text{Maximize} \quad \sum_{i=1}^n \sum_{j=1}^m A_i C_{ij} X_j$$

$$\text{ข้อจำกัด} \quad \sum_{j=1}^m C_{ij} x_j \geq D_i$$

$$x_j \geq 0 \text{ และ จำนวนเต็ม ; } i = 1, 2, 3, \dots, n ; j = 1, 2, 3, \dots, m$$

กำหนดให้

x_j คือจำนวนแพทเทินที่ j

C_{ij} คือจำนวนชิ้นงานที่ i ในแพทเทินที่ j

D_i คือจำนวนชิ้นงานที่ i ที่ต้องการผลิต

A_i คือพื้นที่ชิ้นงานที่ i

การประมวลผลจะอาศัยวิธีการ Branch & Bound หาปริมาณรูปแบบการจัดวางชิ้นงานที่ใช้ในแต่ละรูปแบบ ซึ่งปริมาณรูปแบบการจัดวางชิ้นงานที่ใช้นั้นค่าที่ได้จะเป็นขอบเขตบน และเกิดปริมาณที่เหลือ (Retain) เกิดขึ้นซึ่งต้องมีการปรับปรุงโดยการวนรอบหารูปแบบที่เหมาะสมเพิ่มเติมต่อไป

การหารูปแบบที่เป็นไปได้โดยมีมูลค่าโดยรวมสูงสุด โดยใช้คอมพิวเตอร์ในการคำนวณ ซึ่งขั้นตอนนี้จะอาศัยการแก้ไขปัญหาคำนวณด้วยโปรแกรมเสริม Excel Solver ในการคำนวณด้วยวิธีการ Branch & Bound โดยทางผู้วิจัยได้เขียนโปรแกรม VBA และเชื่อมโยงกับโปรแกรม Visual Basic โดยโครงสร้างในการคำนวณมีดังนี้

1. สร้างตารางคำนวณพร้อมรูปแบบการตัดเริ่มแรก (Initial Solution) ที่ได้จากการคำนวณผ่านโปรแกรม Visual Basic 6 แล้วส่งค่ากลับสู่ Excel Worksheet
2. การเขียนโปรแกรม VBA เพื่อควบคุมการทำงานของ Solver ให้มีการคำนวณแบบอัตโนมัติ โดยมีการกำหนดค่า Target cells, Changing cells, สมการขีดจำกัดแล้วทำการ Run Solver เมื่อได้ผลลัพธ์จากการ Run Solver จะได้คำตอบที่สามารถนำไปใช้งานได้

3.7 การออกแบบโครงสร้างการใช้งานของโปรแกรม

การออกแบบโครงสร้างการใช้งานของโปรแกรมนั้นต้องแสดงให้เห็นภาพรวมของการทำงานดังแสดงให้เห็นภาพรวมการไหลของข้อมูล (Information Flow) ผ่านโปรแกรมทางคอมพิวเตอร์ แสดงดังรูปที่ 3.8 เพื่อเป็นตัวกำหนดรูปร่างหน้าจอการทำงานหลักๆของโปรแกรม



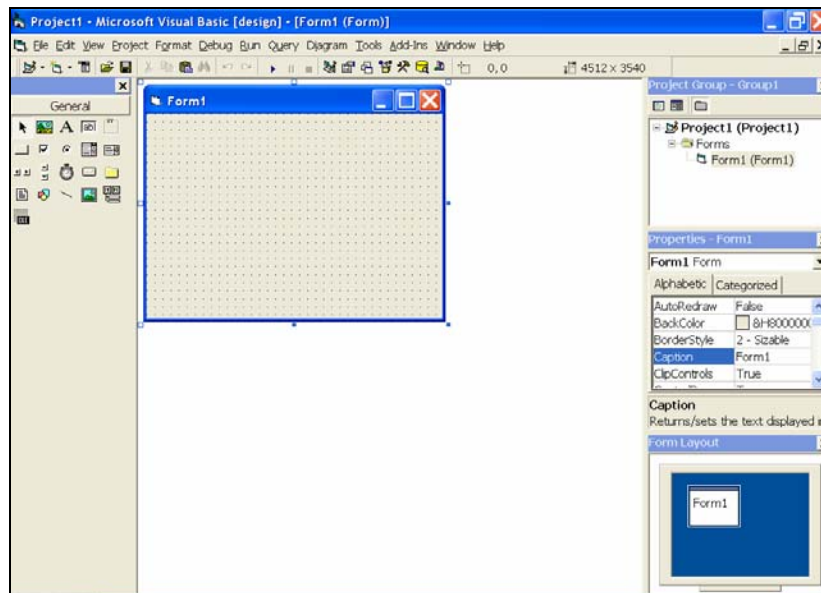
รูปที่ 3.8 แสดงภาพรวมของการไหลของข้อมูลผ่านโปรแกรมทางคอมพิวเตอร์

การออกแบบหน้าจอของโปรแกรมที่ใช้งานต้องให้สอดคล้องกับการประมวลผลการใช้งานและข้อมูลที่มีความจำเป็นที่ต้องใช้ในการผลิต ซึ่งแบ่งออกเป็นส่วนต่างๆดังต่อไปนี้

1. การออกแบบการฟอร์มที่ใช้ในการกรอกข้อมูลเบื้องต้น การแสดงผล และรายงานต่างๆ
2. การออกแบบระบบฐานข้อมูลและการเชื่อมโยงฐานข้อมูลกับโปรแกรมร่วม

3.7.1 การออกแบบการฟอร์มที่ใช้ในการกรอกข้อมูลเบื้องต้น

วิธีการเขียนโปรแกรมหน้าจอการทำงานนั้นอาศัยโปรแกรม Visual Basic 6 เป็นโปรแกรมหลักในการเขียน โดยใช้คอนโทรลต่างๆในโปรแกรม Visual Basic 6 ในการออกแบบหน้าจอการทำงาน ดังแสดงในรูปที่ 3.9



รูปที่3.9 แสดงรูปคอนโทรลต่างๆในโปรแกรม Visual Basic 6 ที่ใช้ในการออกแบบหน้าจอ

3.7.2 การออกแบบระบบฐานข้อมูล

ในงานวิจัยนี้ระบบฐานข้อมูลที่ใช้คือ โปรแกรม Microsoft Access 97 เป็นระบบฐานข้อมูลชื่อ Temp.mdb โดยการสร้างตารางเก็บข้อมูลขึ้นดังนี้

1. ตาราง BLOCK จะเก็บข้อมูลเบื้องต้นของชิ้นงานที่ต้องการนำมาคำนวณ เช่น ชื่อชิ้นงาน ความกว้าง ความยาว จำนวนที่ต้องการ เป็นต้น ดังแสดงในรูปที่ 3.10
2. ตาราง BLOCKPOSITION จะเก็บค่าตำแหน่งของชิ้นงานแต่ละชิ้นที่ได้จากการประมวลผลผ่านโปรแกรม VB แล้วนำไปแสดงผลเป็นกราฟฟิคต่อไป ดังแสดงในรูปที่ 3.11
3. ตาราง ALLPATTERN จะเก็บค่าปริมาณชิ้นงานแต่ละขนาดในแต่ละรูปแบบที่ได้จากการประมวลผลของโปรแกรม เพื่อนำไปใช้ในการคัดเลือกหารูปแบบการจัดวางที่เหมาะสมกับการผลิตต่อไป ดังแสดงในรูปที่ 3.12
4. ตาราง PATTERN จะเก็บข้อมูลการประมวลผลการคัดเลือกหารูปแบบการจัดวางที่เหมาะสมกับการผลิตโดยผ่านโปรแกรม EXCEL SOLVER ที่ควบคุมการทำงานด้วยโปรแกรม VBA แล้วส่งค่ากลับมาเก็บไว้ในตาราง ดังแสดงในรูปที่ 3.13

5. ตาราง SPACES จะเก็บข้อมูลของพื้นที่ในส่วนที่เหลือ เพื่อใช้ในการสร้างรูปแบบการจัดวางเริ่มต้นที่เป็นไปได้ทั้งหมด ดังแสดงในรูปที่ 3.14

The screenshot shows the Microsoft Access interface for the 'BLOCK : ตาราง' (BLOCK : Table) table. The table structure is as follows:

ชื่อเขตข้อมูล	ชนิดข้อมูล	ค่าอธิบาย
BLOCKNAME	Text	
BWIDTH	Number	
BLENGTH	Number	
NEED	Number	
Pro_No	Text	
Con_No	Text	
P/O_No	Text	
Od_No	Text	
Mat_Type	Text	
BThick	Number	
BLOCKORDER	Number	
AREA	Number	
WIDTHFORM	Number	
LENGTHFORM	Number	
WIDTHRFORM	Number	
LENGTHRFORM	Number	

The Field Properties window is open, showing the following settings:

- หัวข้อ: ค้นหา
- ขนาดเขตข้อมูล: Long Integer
- รูปแบบ: Auto
- จุดทศนิยม: 0
- รูปแบบการป้อนข้อมูล: 0
- ป้ายคำอธิบาย: No
- ค่าเริ่มต้น: Yes (Duplicates OK)

At the bottom, the status bar shows: มุมมองออกแบบ F6 = สลับหน้าต่าง F1 = วิสัย

รูปที่ 3.10 แสดงการออกแบบตาราง BLOCK ในฐานข้อมูล TEMP.MDB

The screenshot shows the Microsoft Access interface for the 'BLOCKPOSITION : ตาราง' (BLOCKPOSITION : Table) table. The table structure is as follows:

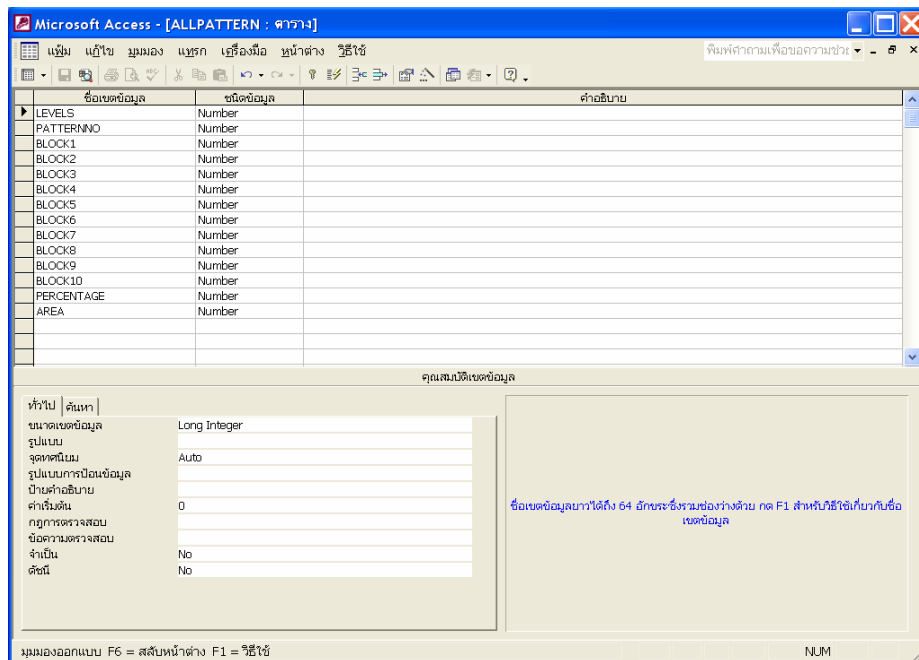
ชื่อเขตข้อมูล	ชนิดข้อมูล	ค่าอธิบาย
PATTERNNO	Number	
BLOCKNO	Number	
BLOCKORDER	Number	
XBEGIN	Number	
XEND	Number	
YBEGIN	Number	
YEND	Number	
TOPSPACE	Number	
RIGHTSPACE	Number	
SPACENO	Number	
LEVELS	Number	

The Field Properties window is open, showing the following settings:

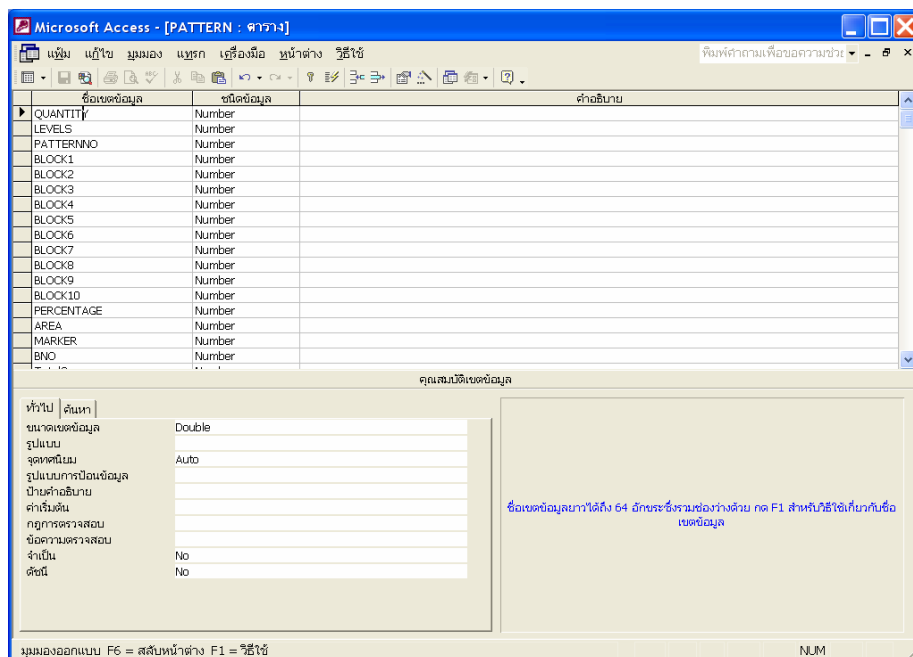
- หัวข้อ: ค้นหา
- ขนาดเขตข้อมูล: Long Integer
- รูปแบบ: Auto
- จุดทศนิยม: 0
- รูปแบบการป้อนข้อมูล: 0
- ป้ายคำอธิบาย: No
- ค่าเริ่มต้น: No

At the bottom, the status bar shows: มุมมองออกแบบ F6 = สลับหน้าต่าง F1 = วิสัย

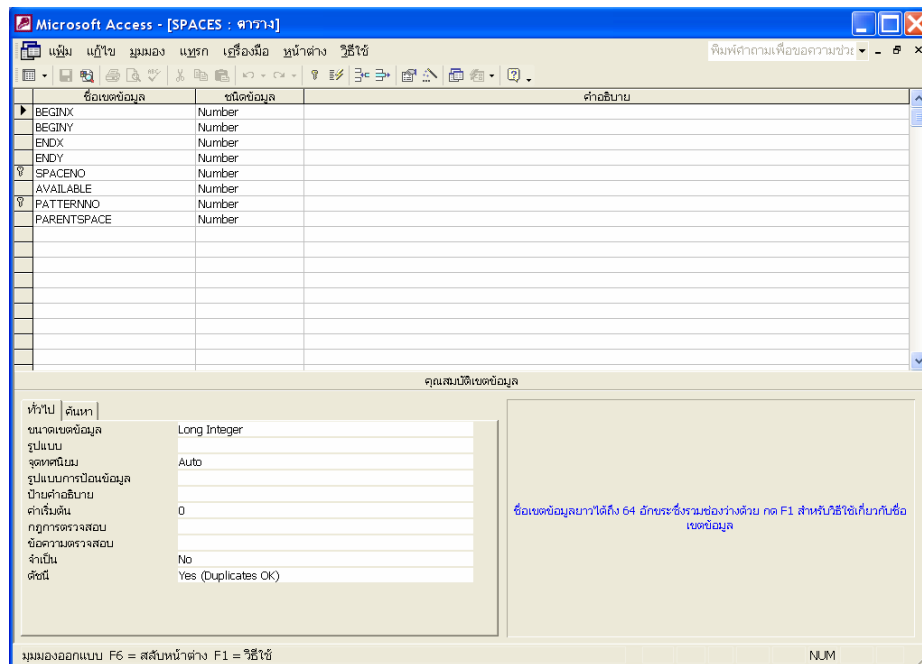
รูปที่ 3.11 แสดงการออกแบบตาราง BLOCKPOSITION ในฐานข้อมูล TEMP.MDB



รูปที่ 3.12 แสดงการออกแบบตาราง BLOCKPOSITION ในฐานข้อมูล TEMP.MDB



รูปที่ 3.13 แสดงการออกแบบตาราง PATTERN ในฐานข้อมูล TEMP.MDB



รูปที่ 3.14 แสดงการออกแบบตาราง SPACES ในฐานข้อมูล TEMP.MDB

3.8 การทดสอบผลการทำงานของโปรแกรม

การทดสอบผลการทำงานของโปรแกรมการจัดวางชิ้นงานมีขั้นตอนดังนี้

1. นำข้อมูลจากฝ่ายวางแผนการผลิตมาทดสอบ
2. เปรียบเทียบกับผลการทดลองที่ได้จากการประมวลผลผ่านโปรแกรมกับผลที่เกิดจากการวางแผนงานปัจจุบันที่ใช้อยู่
3. วิเคราะห์ต้นทุนที่เกิดขึ้น

การเปรียบเทียบผลการทดลองจะให้ความสนใจถึงปริมาณวัตถุดิบที่ต้องใช้งาน ปริมาณเศษเหลือ เวลาที่ใช้ในการออกแบบ ตลอดจนต้นทุนวัตถุดิบที่เกิดขึ้น โดยแสดงตารางเปรียบเทียบในส่วนต่างๆดังที่แสดงในตารางที่ 3.3

ตารางที่ 3.3 แสดงการเปรียบเทียบผลการทดลองกับผลที่ได้จากหน่วยการผลิต

แหล่งที่มาข้อมูล	ปริมาณไม้ที่ต้องเบิก มาใช้งาน	เปอร์เซ็นต์การ สูญเสีย	เวลาที่ใช้ใน การออกแบบ	ต้นทุน วัตถุดิบ
หน่วยการผลิต				
โปรแกรมคอมพิวเตอร์				

บทที่ 4 ผลการดำเนินงาน

ในบทนี้จะกล่าวถึงผลการดำเนินงานวิจัยการพัฒนารูปแบบการจัดวางชิ้นงานลงบนแผ่นไม้บอร์ด ที่ได้จากการศึกษาและทำการทดสอบผลการทดลอง ซึ่งเป็นไปตามโครงสร้างการดำเนินงานที่ได้กล่าวรายละเอียดไว้ในบทที่ 3 มาทำการทดสอบกับข้อมูลจริง ซึ่งได้รับการอนุเคราะห์จากบริษัท พี.พี.พาราวูด จำกัด โดยมีเป้าหมายเพื่อทดสอบหาประสิทธิภาพการทำงานของโปรแกรม ทำการเปรียบเทียบผลลัพธ์ที่ได้กับรูปแบบการจัดวางชิ้นงานจริง ในเชิงปริมาณ ต้นทุน และการบริหารการจัดการที่เกิดขึ้น ซึ่งผลการดำเนินงานการศึกษาแบ่งออกเป็นหัวข้อดังต่อไปนี้

1. ผลศึกษาสภาพปัจจุบันของบริษัทที่เข้าร่วมโครงการ
2. ผลการวิเคราะห์ข้อมูลที่ได้รวบรวมจากหน่วยการผลิตที่เกี่ยวข้อง
3. ผลการสร้างตัวแบบทางคณิตศาสตร์
4. ผลประยุกต์การใช้งานตัวแบบทางคณิตศาสตร์ด้วยคอมพิวเตอร์
5. ผลทดสอบการทำงานของโปรแกรม
6. ผลการเปรียบเทียบประสิทธิภาพในด้านต่างๆของโปรแกรมทดสอบ

4.1 ผลการศึกษาสภาพปัจจุบันของบริษัทที่เข้าร่วมโครงการ

บริษัท พี.พี. พาราวูด จำกัด เป็นโรงงานผลิตเฟอร์นิเจอร์เครื่องเรือนจากไม้ยางพาราเพื่อการส่งออกและจำหน่ายในประเทศ โรงงานตั้งอยู่ที่อำเภอพนัสนิคม จังหวัดชลบุรี ผลิตภัณฑ์ที่ผลิตได้แก่ ตู้ เตียง และเก้าอี้ จำนวนการผลิตต่อเดือนมีดังนี้ ตู้ 2000-2500 เตียง 800-1000 โต๊ะ 2500 และเก้าอี้ 12000-15000 หน่วย การผลิตเป็นลักษณะตามแบบที่ลูกค้าสั่ง ซึ่งสินค้ามีความหลากหลาย วัตถุดิบของโรงงานได้แก่ ไม้ยางพารา บอร์ด และ วนิเยร์ ซึ่งเป็นไม้นำเข้า โดยประมาณมูลค่าไม้ยางพารา 20 ล้านบาท ไม้บอร์ด 10 ล้านบาท และวนิเยร์ 2.7 ล้านบาทต่อเดือน ปัจจุบันโรงงานมีต้นทุนการผลิตสูง จากความสูญเสียที่เกิดขึ้นในกระบวนการการใช้ไม้ MDF และ PB ในการผลิต จากขั้นตอนการตัดชิ้นไม้เพื่อนำไปผลิต และต้นทุนการผลิตรวมยังขึ้นอยู่กับปริมาณไม้ที่ใช้ผลิต

4.1.1 สภาพปัญหาในส่วนงานผลิตที่เกี่ยวข้อง

เมื่อโรงงานรับใบสั่งซื้อสินค้าจากลูกค้าในแต่ละเดือน ซึ่งผลิตภัณฑ์ส่วนใหญ่ได้แก่ ตู้ เตียง และ โต๊ะ ที่ใช้ไม้บอร์ดในการผลิต ฝ่ายวางแผนจะแยกแบบรายการสินค้าในแต่ละเดือนแล้วส่งให้แผนกงานตัดและงานเพรส คำนวณหารูปแบบการจัดวางชิ้นงานการตัดและปริมาณไม้บอร์ดที่ต้องใช้

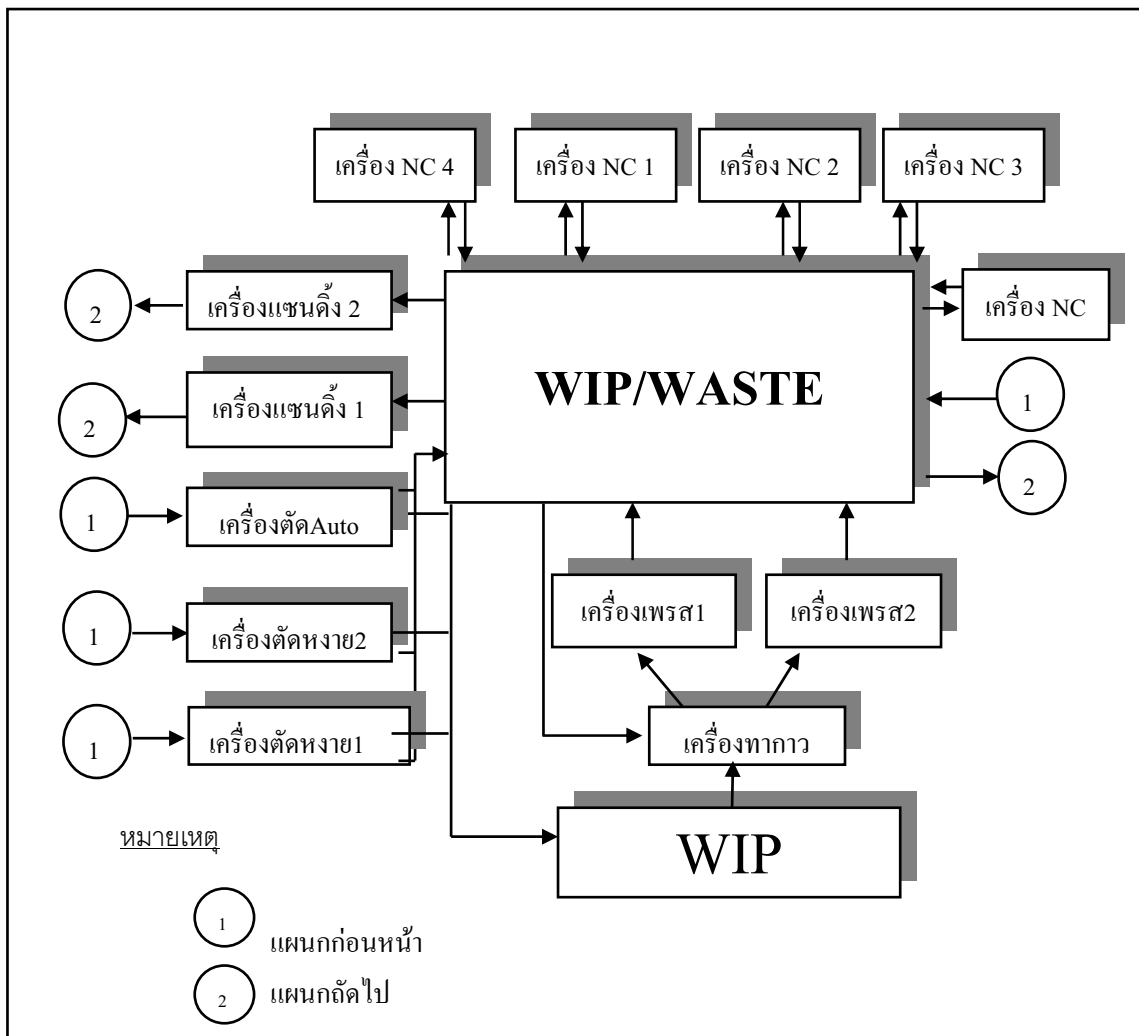
โดยที่มีขนาดมาตรฐาน $4' \times 8'$ ซึ่งจะแบ่งตามชนิดความหนาของไม้ดังนี้

- MDF(Medium density fiber) หนา 12,15,16,18,2.6,25,3,4,6,9 มม.
- MDF(Medium density fiber) + ไม้เบญจพรรณขาว 2 หน้า (หนา 0.5 มม.) หนา 2.6 มม.
- MDF(Medium density fiber) + ไม้เบอร์ซ่า 2 หน้า(หนา 0.5 มม.) หนา 2.6 มม.
- MDF(Medium density fiber) + ไม้ยางพาราขาว(หนา 0.5 มม.) หนา 2.6 มม.
- MDF(Medium density fiber) + ไม้ยาง หนา 4 มม.
- PB(particle board) หนา 12 15 18 22 25 มม.

การคำนวณจะอาศัยชนิดและความหนาของไม้เป็นหลัก การคำนวณส่วนใหญ่จะคิดเฉพาะรายการสินค้าเพียงรายการเดียว โดยมีเงื่อนไขดังนี้ ให้เหลือเศษไม้ให้น้อยที่สุด มีการเผื่อขนาดจากขนาดจริงรอบตัว 10 มม. การตัดต้องตัดตลอดแนว ไม้ที่ยังใช้ประโยชน์ได้ต้องนำกลับมาเปลาะแล้วใช้ได้อีกโดยที่ไม้บอร์ดจะนำไปเปลาะได้ต้องกว้างไม่น้อยกว่า 2 นิ้วและต้องหนาไม่ต่ำกว่า 18 มม. จากการคำนวณพบว่ารายการที่คำนวณบางรายการมีจำนวนไม้ครบตามจำนวนต้องการเพราะกลัวจะเหลือเศษมาก ซึ่งจะสะสมไว้รอการคำนวณ ภายหลังมีการรวบรวมจำนวนไม้ที่ไม่ครบจำนวนทั้งหมดของรายการในแต่ละเดือน นอกจากนั้นแล้วการคำนวณบางครั้งอาจนำเอารายการสินค้าที่ยังไม่มีการผลิตในเดือนนั้นมารวมด้วยเนื่องจากชนิดและขนาดไม้ในรายการอื่นไม่สามารถมารวมกันได้ ทำให้เสียพื้นที่จัดเก็บเป็นจำนวนมาก รายการไม้ที่ตัดเสร็จจะมีในลังงานแบบอยู่ได้แก่ สีชมพู ขาว เหลือง ฟา และเขียว ขึ้นอยู่กับผลิตภัณฑ์ เศษไม้ที่เหลือที่ใช้ประโยชน์ได้จะไม่มีใบแบบอาจจะมิชอบลักเขียนบอกขนาดไว้ เมื่อเวลาผ่านไปไม่นานฝุ่นควันจะปกคลุมชอล์กเหล่านั้นทำให้การติดตามรายการไม้ยากไม่รู้ขนาดส่วนไม้จากรายการที่ไม่ได้สั่งผลิต แม้ว่าจะมีใบชี้บ่งบอกลักษณะผลิตภัณฑ์แต่จะเห็นได้ชัดว่ามีเหลือเก็บอยู่มากเสียพื้นที่และปะปนอยู่กับเศษไม้ที่เหลือทำให้การตรวจเช็คติดตามรายการไม้ค่อนข้างยาก นอกจากนั้นแล้วพบว่าพนักงานตัดยังมีความผิดพลาด บ่อยครั้งเกิดความผิดพลาดจากการป้อนระยะ การตั้งระยะมีดตัด วางไม้ผิดด้าน ไม่เป็นไปตามรูปแบบการจัดวางชิ้นงานที่วางไว้ให้ ไม้ที่สั่งมาครบจำนวนไม่สามารถที่จะสั่งมาเพิ่มอีกได้จึงต้องประยุกต์แผนการตัดเสียใหม่ทำให้มีรายการขาดมากขึ้น นอกจากนั้นจะมีไม้บางรายการเหลือเก็บอยู่มากและโอกาสการใช้งานน้อยซึ่งได้แก่ MDF หนา 9 12 25 มม. บางครั้งในการคำนวณหากรายการสินค้าที่ต้องใช้ไม้ MDF หนา 6 มม. ต้องมีการประยุกต์โดยการนำไม้ MDF หนา 9 มม. มาการลดขนาดไม้โดยการปิดกระดาดทราย หน้าเดียว ออก 3 มม. เพื่อให้ได้ไม้หนา 6 มม. ในงานลดขนาดไม้โดยการปิดกระดาดทรายยังไม่เก็บข้อมูลการทำงานไว้เมื่อพนักงานคนใหม่มารับงานต่อไม่ทราบว่ามีไม้ได้ถูกลดขนาดไม้โดยการปิดกระดาดทรายไปแล้วเท่าไร

จากการสำรวจสภาพการทำงานในแผนกงานตัด/งานเพรส ซึ่งมีแผนผังการจัดวางเครื่องจักรดัง
แสดงใน รูปที่ 4.1 พบว่ามีการไหลตัวของวัตถุดิบไปยังกระบวนการต่าง ๆ ดังนี้

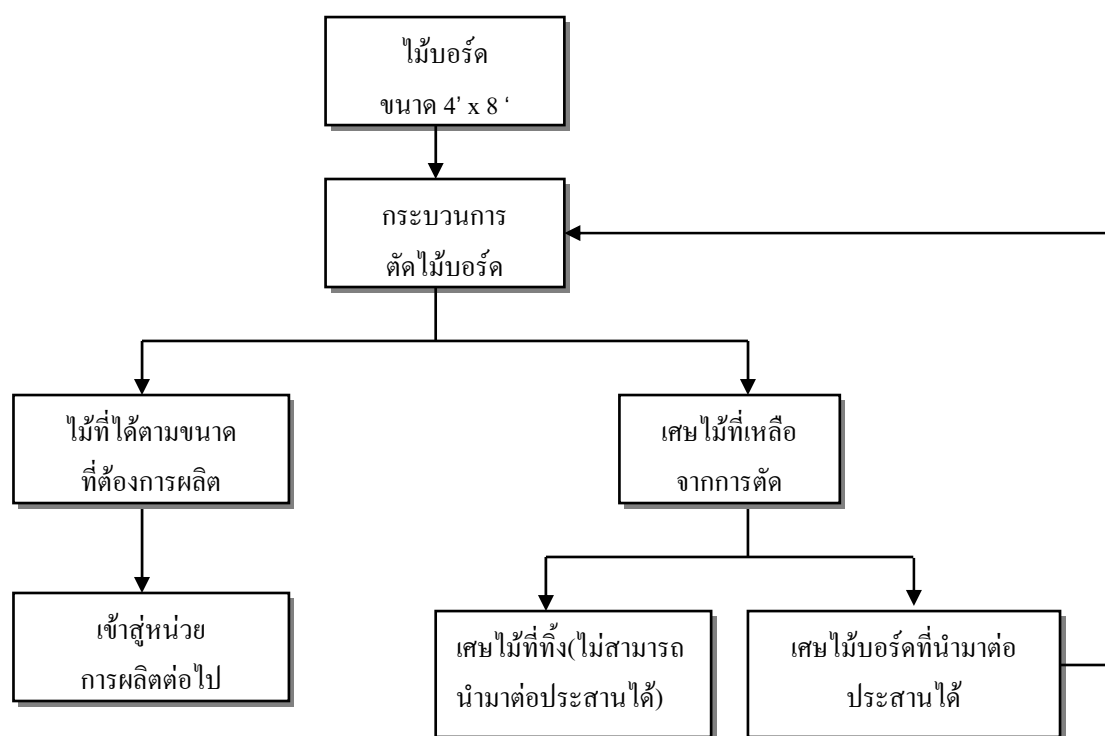
1. คลังเก็บวัตถุดิบ→เครื่องตัด→พท.เก็บ WIP→เครื่องทากาว→เครื่องเพรส→
พท.เก็บ WIP/WASTE→เครื่องลดขนาดไม้โดยการปัดกระดาษทราย→แผนกถัดไป
2. เครื่อง NC →พท.เก็บ WIP/WASTE→แผนกถัดไป



รูปที่4.1 แสดงแผนผังเครื่องจักรในแผนกงานตัด/เพรส

4.2 ผลการวิเคราะห์และเก็บข้อมูล

ผลวิเคราะห์กระบวนการตัดแต่ง รูปที่ 4.2 พบว่าปริมาณบอร์ดที่ใช้ได้ประมาณ 70 % ส่วนไม้บอร์ดที่เป็นเศษที่เหลือ ประมาณเศษที่ใช้ได้ 25-30 % จะถูกนำไปใช้กับผลิตภัณฑ์ตัวอื่น และเมื่อมีเศษเหลืออีกจะนำไปผ่านกระบวนการอัดประสานครั้งแรก เพื่อนำกลับมาใช้อีกและถ้ามีเศษเหลือที่ต่อได้จะมีการอัดประสานอีกครั้ง ทำให้มีเศษเหลือครั้งหลังสุดประมาณ 5-10 %



รูปที่ 4.2 การใช้ไม้บอร์ดในขั้นตอนการตัด

จากสภาพที่เกิดขึ้นความสูญเสียจากปริมาณเศษสูงเกิดในระหว่างงานผลิตเป็นผลให้เสียพื้นที่ในการเก็บ เงินทุนจมในวัตถุดิบที่เหลือ เกิดปริมาณงานมากในขั้นตอนการอัดประสาน ดังนั้นปัญหาความสูญเสียที่เกิดขึ้นดังกล่าวนี้สามารถลดลงได้โดยการกำหนดแบบการตัดที่เหมาะสมให้มีผลรวมของเศษไม้น้อยที่สุด หรือ มีผลรวมของการใช้ไม้บอร์ดน้อยที่สุด

เพื่อศึกษาขั้นตอนของกระบวนการในเบื้องต้นจึงทำการเก็บข้อมูลกระบวนการควบคุมการตัดไม้บอร์ดและการเพรส (ดังแสดงใน รูปที่ ก.1 ก.2 และ ก.3 ภาคผนวก ก) โดยเริ่มจากการรับแผนการผลิตจากผู้จัดการฝ่ายผลิต จนกระทั่งสิ้นสุดกระบวนการ แล้วทำการศึกษารูปแบบการจัดวางชิ้นงานการตัดไม้บอร์ด (ดังแสดงในตัวอย่าง รูปที่ ข.1 และ ข.2 ภาคผนวก ข) และศึกษาต้นทุนวัตถุดิบที่เกิดขึ้นจากการวางรูปแบบการจัดวางชิ้นงาน (ดังแสดงในตารางที่ ค.1 ค.2 ค.3 และ ค.4 ภาคผนวก ค) ซึ่งสามารถที่จะสรุปผลการศึกษาด้านทุนที่เกิดขึ้นดังแสดงในตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 แสดงต้นทุนวัตถุดิบที่เกิดขึ้นตลอดระยะเวลา 3 เดือน

เดือน/ต้นทุน	ต้นทุนที่เกิดจากเศษไม้ที่ใช้ไม่ได้	ต้นทุนที่เกิดจากเศษไม้ที่เหลือ
เดือนที่ 7	183,089.00 บาท	85,377.50 บาท
เดือนที่ 8	75,179.00 บาท	93,987.50 บาท
เดือนที่ 9	169,978.50 บาท	74,798.00 บาท
ค่าใช้จ่ายรวม	<u>428,246.50</u> บาท	<u>254,163.00</u> บาท

4.3 ตัวแบบทางคณิตศาสตร์

ตัวแบบทางคณิตศาสตร์ที่พัฒนาขึ้นนั้นแบ่งออกเป็นส่วนๆดังต่อไปนี้

- การหารูปแบบการตัดที่เป็นไปได้ (Feasible Patterns)
- การหาจำนวนแผ่นไม้ที่น้อยที่สุดที่ต้องเบิกใช้งาน

4.3.1 การหารูปแบบการตัดที่เป็นไปได้ (Feasible Patterns)

การหารูปแบบการตัดที่เป็นไปได้ (Feasible Patterns) มีรูปแบบสมการทางคณิตศาสตร์ที่เกี่ยวข้อง รวมถึงการผสมผสานวิธีการบรรจุแบบ (BL-Bottom Left algorithm) ดังนี้

สมการทางคณิตศาสตร์

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^+ W_{ij} \leq W_{ij} \quad \dots\dots\dots 1$$

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^- L_{ij} \leq L_{ij} \quad \dots\dots\dots 2$$

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij}^+ C_{ij}^- W_{ij} L_{ij} \leq W_{ij} L_{ij} \quad \dots\dots\dots 3$$

กำหนดให้

C_{ij}^- คือจำนวนแถวของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j

C_{ij}^+ คือจำนวนสดมภ์ของการบรรจุชิ้นงานที่ i ในแพทเทินที่ j

C_{ij} คือจำนวนชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j

W_{ij} คือ ความกว้างของชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j

L_{ij} คือ ความยาวของชิ้นงานที่ i ในรูปแบบการจัดวางชิ้นงานที่ j

สมการทางคณิตศาสตร์ข้างต้นเป็นแบบจำลองที่ได้สร้างขึ้นเพื่อต้องการหารูปแบบที่เป็นไปได้ทั้งหมด ในการจัดเรียงชิ้นงานขนาดต่างๆลงในสี่เหลี่ยมขนาดมาตรฐาน โดยแบบจำลองที่สร้างขึ้นต้องอาศัยวิธีการทางฮิวริสติกแบบอัลกอริทึมลงล่างซ้าย (BL-Bottom Left algorithm) ในการที่จะกำหนดทิศทางการจัดเรียงชิ้นงานด้วย สำหรับสมการที่ได้จำลองขึ้นสามารถที่จะอธิบายได้ดังนี้

สมการที่ 1

แบบจำลองที่แสดงถึงการจัดเรียงชิ้นงานขนาดต่างๆลงในสี่เหลี่ยมขนาดมาตรฐาน โดยที่ผลรวมของความกว้างชิ้นงานต้องน้อยกว่าหรือเท่ากับความกว้างของสี่เหลี่ยมขนาดมาตรฐาน กล่าวโดยสรุปคือต้องการเรียงชิ้นงานให้ได้มากที่สุดจากมุมล่างซ้ายขึ้นไปด้านบนตามความกว้างของไม้บอร์ด

สมการที่ 2

แบบจำลองที่แสดงถึงการจัดเรียงชิ้นงานขนาดต่างๆลงในสี่เหลี่ยมขนาดมาตรฐาน โดยที่ผลรวมของความยาวชิ้นงานต้องน้อยกว่าหรือเท่ากับความยาวของสี่เหลี่ยมขนาดมาตรฐาน กล่าวโดยสรุปคือต้องการเรียงชิ้นงานให้ได้มากที่สุดจากมุมล่างซ้ายไปทางด้านขวามือตามความยาวของไม้บอร์ด

สมการที่ 3

แบบจำลองที่แสดงถึงผลรวมของพื้นที่ในการจัดเรียงชิ้นงานขนาดต่างๆลงในสี่เหลี่ยมขนาดมาตรฐาน โดยที่ผลรวมของพื้นที่ชิ้นงานที่บรรจุต้องน้อยกว่าหรือเท่ากับพื้นที่ของสี่เหลี่ยมขนาดมาตรฐาน กล่าวโดยสรุปคือ พื้นที่ชิ้นงานทั้งหมดที่เกิดจากการจัดเรียงต้องไม่มากกว่าพื้นที่ของไม้บอร์ด

4.3.2 การหาจำนวนแผ่นไม้ที่น้อยที่สุดที่ต้องเบิกใช้งาน

การหาจำนวนแผ่นไม้ที่น้อยที่สุดที่ต้องเบิกใช้งาน โดยอาศัยรูปแบบการตัดที่เป็นไปได้ (Feasible-Patterns) ทั้งหมดที่เกิดจากการคำนวณหาในช่วงที่ 1 ซึ่งมีรูปแบบสมการทางคณิตศาสตร์ที่เกี่ยวข้องดังนี้

สมการทางคณิตศาสตร์

$$\begin{aligned} \text{สมการเป้าหมาย} \quad & \text{Maximize} \quad \sum_{i=1}^n \sum_{j=1}^m A_i C_{ij} X_j \\ \text{ข้อจำกัด} \quad & \sum_{j=1}^m C_{ij} x_j \geq D_i \\ & \geq 0 \text{ และ จำนวนเต็ม ; } i = 1, 2, 3, \dots, n ; j = 1, 2, 3, \dots, m \end{aligned}$$

กำหนดให้

- x_j คือจำนวนแพทเทินที่ j
- C_{ij} คือจำนวนชิ้นงานที่ i ในแพทเทินที่ j
- D_i คือจำนวนชิ้นงานที่ i ที่ต้องการผลิต
- A_i คือพื้นที่ชิ้นงานที่ i

สมการทางคณิตศาสตร์ข้างต้นเป็นจำลองที่ได้สร้างขึ้นนี้จะเป็นการสร้างสมการเป้าหมายและเงื่อนไขในการจัดเรียงชิ้นงานซึ่งอธิบายได้ดังนี้

สมการเป้าหมาย

แบบจำลองที่แสดงถึงผลรวมของพื้นที่ ที่เกิดจากรูปแบบการจัดเรียงชิ้นงานแต่ละรูปแบบที่ได้คัดเลือกมาจะต้องน้อยที่สุด กล่าวโดยสรุป วัตถุประสงค์คือต้องการรูปแบบการจัดเรียงที่ทำให้เกิดการประหยัดไม้บอร์ดให้มากที่สุด

สมการเงื่อนไข

แบบจำลองที่แสดงถึงข้อจำกัดของผลรวมของจำนวนชิ้นงานที่ในแต่ละรูปแบบของการจัดเรียงที่คัดเลือกมาต้องมากกว่าหรือเท่ากับปริมาณความต้องการที่ต้องการผลิตกล่าวโดยสรุปคือชิ้นงานที่ได้ต้องเพียงพอต่อการผลิต

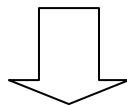
4.4 ผลประยุกต์การใช้งานตัวแบบทางคณิตศาสตร์ด้วยคอมพิวเตอร์

4.4.1 ลำดับขั้นตอนในการดำเนินงานในการแก้ไขปัญหา

การประยุกต์ตัวแบบทางคณิตศาสตร์ด้วยคอมพิวเตอร์มีลำดับขั้นตอนการแก้ไขปัญหาดังนี้

ขั้นตอนที่ 1

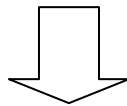
จัดเรียงชิ้นงานสี่เหลี่ยมจัตุรัสด้วยวิธีการ Bottom Left



ทำให้ได้รูปแบบการจัดวางชิ้นงาน
ทุกรูปแบบที่สามารถที่จัดเรียงได้

ขั้นตอนที่ 2

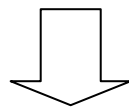
เรียงลำดับการใช้ประโยชน์ของพื้นที่โดยรวมสูงสุดในแต่ละ
ระดับ โดยเรียงลำดับจากพื้นที่ใช้ประโยชน์จากมากไปหาน้อย



ทำให้ได้รูปแบบการจัดวางชิ้นงานที่
ใช้ประโยชน์ของพื้นที่โดยรวม
สูงสุด แต่ละระดับ

ขั้นตอนที่ 3

คัดเลือกรูปแบบการจัดเรียงเพียง 180 ลำดับแรกที่ได้คัดเลือก
ทำการประมวลผลเริ่มต้น เนื่องจากข้อจำกัดด้านตัวแปร



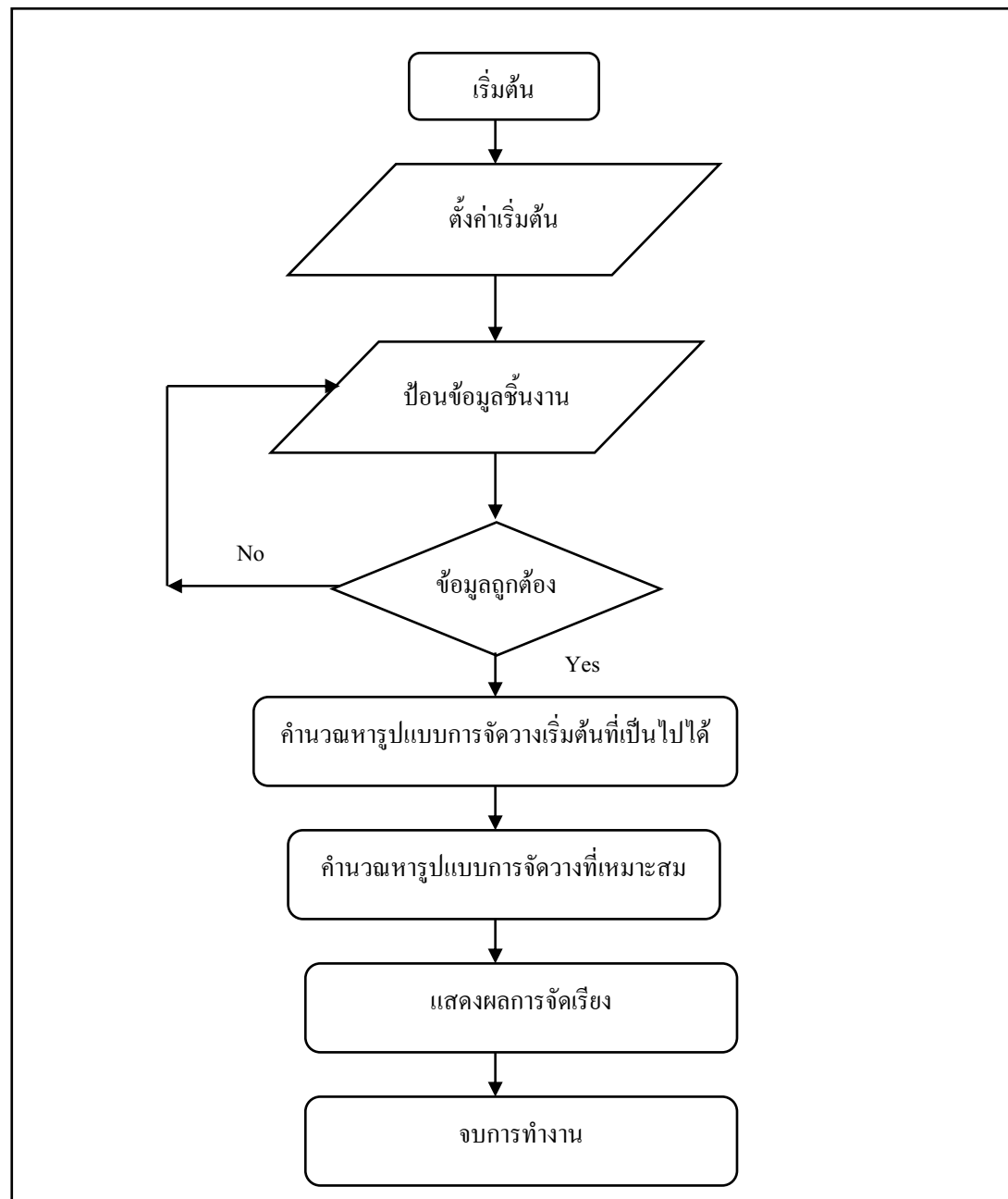
ทำให้ได้รูปแบบการจัดวางชิ้นงานที่
ใช้ประโยชน์ของพื้นที่โดยรวม
สูงสุด 180 ลำดับแรก เพื่อการ
ประมวลผล

ขั้นตอนที่ 4

ประมวลผลหารูปแบบการจัดวางชิ้นงานที่เหมาะสมให้ได้
จำนวนที่ต้องการ โดยมีเงื่อนไขคือผลรวมการใช้ไม้บอร์ดน้อย
ที่สุด

การประมวลผลด้วยโปรแกรมคอมพิวเตอร์

การประยุกต์ตัวแบบทางคณิตศาสตร์มาใช้นั้นจะอาศัยคอมพิวเตอร์ช่วยในการประมวลผล ซึ่งโปรแกรมที่ได้จัดทำขึ้นนั้นจะเป็นการผสมผสานและการเชื่อมโยงฐานข้อมูลของโปรแกรมรวมต่างๆ เพื่อให้เกิดความสะดวกในการใช้งานและข้อมูลที่ต้องกับการใช้งาน โดยรายละเอียดการทำงานจะแสดงในแผนภาพการไหล ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 แสดงแผนภาพการไหลของโปรแกรม

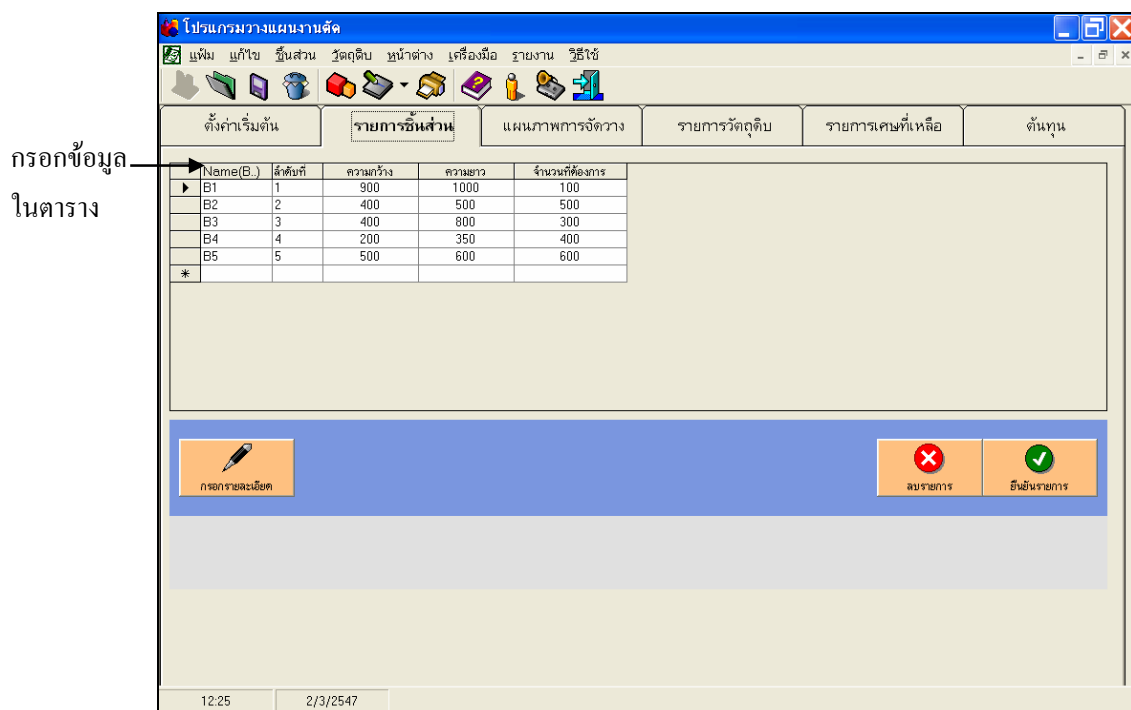
4.4.2 ลำดับขั้นตอนในการประมวลผลของโปรแกรม

1. การตั้งค่าเริ่มต้นการทำงานโดยผ่านโปรแกรม Visual Basic เพื่อเป็นการกำหนดข้อมูลเบื้องต้นในการประมวลผล เช่น ระบบหน่วยวัด การตั้งค่าเพื่อระยะใบเลื่อย เงื่อนไขในการกรอกรายละเอียดของข้อมูล รวมถึงความละเอียดในการประมวลผล ซึ่งการกำหนดค่าเริ่มต้นนี้ทางผู้วิจัยได้ตั้งค่าเริ่มต้นในโปรแกรมไว้ให้ โดยตั้งค่าเริ่มต้นที่จำเป็นเบื้องต้นในการประมวลผล หากผู้ใช้ต้องการปรับเปลี่ยนก็สามารถที่จะกระทำได้ โดยการคลิกเพื่อเลือกการตั้งค่าที่ต้องการดังแสดงในรูปที่ 4.4 ที่แสดงถึงหน้าจอการทำงานในส่วนของการตั้งค่าเริ่มต้นให้กับโปรแกรม

เลือกข้อมูล
ในตาราง

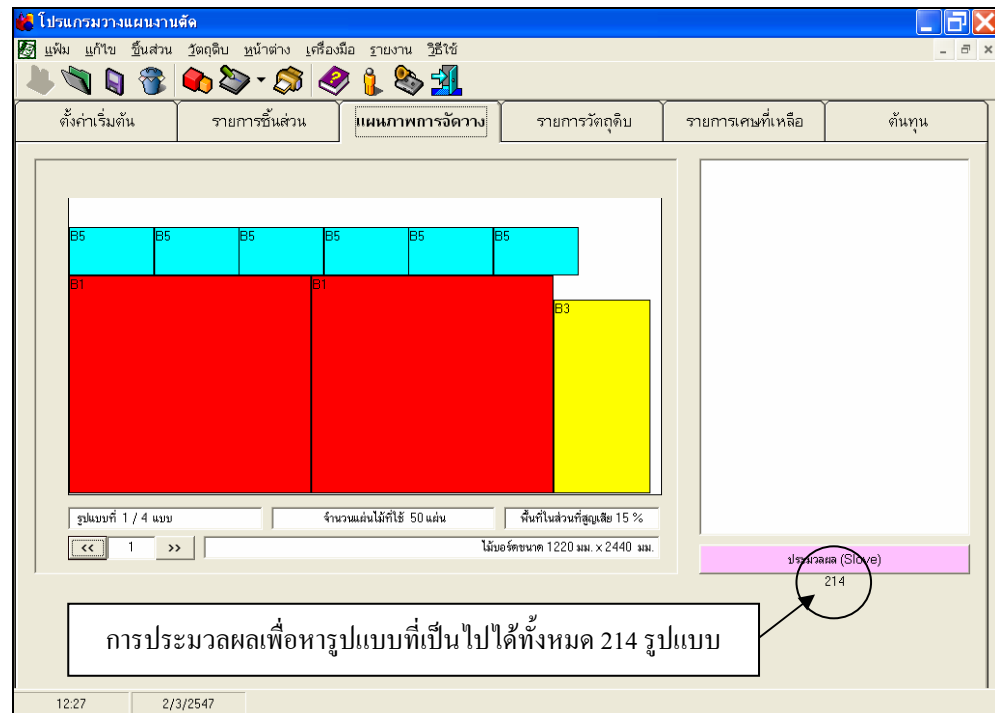
รูปที่ 4.4 แสดงการตั้งค่าเริ่มต้นการทำงาน

2. กรอกข้อมูลที่ต้องการผ่านการทำงานในส่วนของโปรแกรม Visual Basic โดยข้อมูลหลักๆ ที่มีความสำคัญคือขนาดความกว้าง ความยาว และปริมาณความต้องการ ซึ่งเกิดจากการตั้งค่าเริ่มต้นการใช้งานในขั้นตอนที่ 1 ที่ต้องการเลือกการกรอกข้อมูลบางส่วนที่ไม่ต้องการรายละเอียดของข้อมูลมากนัก เพื่อให้เกิดความรวดเร็วในการประมวลผล โดยอาจมีจุดประสงค์เพื่อการคำนวณเพื่อหาปริมาณวัตถุดิบคร่าวๆ เพื่อการประมาณราคา และการตรวจสอบวัตถุดิบคงคลัง เป็นต้น ดังแสดงในรูป ที่ 4.5 หากผู้ใช้ต้องการกรอกรายละเอียดมากขึ้นสามารถกระทำได้โดยการคลิกเลือกการกรอกข้อมูลทั้งหมดที่ต้องการได้ในรายการของการตั้งค่าเริ่มต้นในขั้นตอนที่ 1



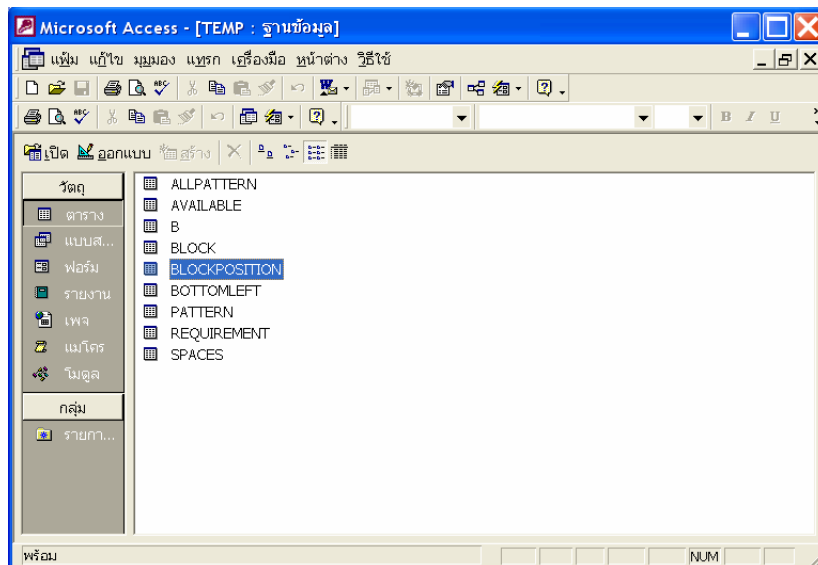
รูปที่ 4.5 แสดงการกรอกรายละเอียดของชิ้นงาน

3. โปรแกรม Visual Basic จะทำการประมวลผลเพื่อหารูปแบบการจัดวางชิ้นงานที่เป็นไปได้ทั้งหมด โดยอาศัยการประยุกต์แบบจำลองคณิตศาสตร์ที่ได้สร้างขึ้นผสมผสานกับการจัดเรียงแบบฮิวริสติก (BL-Bottom Left algorithm) ดังแสดงในรูปที่ 4.6 ซึ่งจะแสดงการประมวลผลในขั้นตอนการหารูปแบบการจัดเรียงที่เป็นไปได้ทั้งหมด 214 รูปแบบที่เกิดจากชิ้นงานทดสอบ เพื่อที่จะนำไปคัดเลือกเพื่อหารูปแบบการจัดเรียงและปริมาณที่เหมาะสมต่อไป



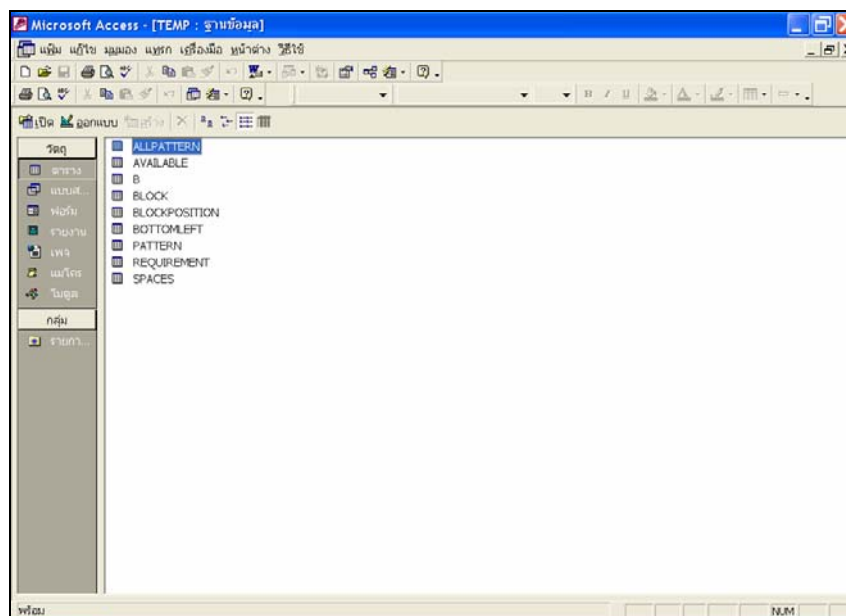
รูปที่ 4.6 แสดงผลการประมวลผลเพื่อหารูปแบบที่เป็นไปได้ทั้งหมด

4. จากนั้นจะส่งข้อมูลที่ได้จากการประมวลผลไปเก็บเป็นฐานข้อมูลในโปรแกรม ACCESS ดังแสดงในรูปที่ 4.7



รูปที่4.7 แสดงตารางฐานข้อมูลในการเก็บข้อมูลต่างๆในการประมวลผล

5. ข้อมูลจากที่เก็บใน ACCESS ในตาราง ALLPATTERN จะถูกส่งไปยัง Excel Solver เพื่อทำการประมวลผล ดังแสดงในรูปที่ 4.8 และ 4.9



รูป 4.8 แสดงตาราง ALLPATTERN ในโปรแกรมเก็บฐานข้อมูล ACCESS

LEVELS	PATTERNNO	BLOCK1	BLOCK2	BLOCK3	BLOCK4	BLOCK5	BLOCK6
0	1	2	0	1	0	6	0
1	2	1	4	0	0	8	0
1	3	0	8	0	0	6	0
2	4	1	3	1	0	6	0
2	5	1	2	1	1	6	0
2	6	1	1	2	1	5	0
2	7	1	0	4	1	4	0
2	8	0	7	0	1	6	0
2	9	0	6	1	0	8	0
2	10	0	5	2	0	7	0
2	11	0	4	3	0	6	0
2	12	0	3	3	1	6	0
2	13	0	2	4	1	6	0
2	14	0	1	6	1	4	0
2	15	0	0	9	0	0	0
3	16	2	0	0	1	8	0
3	17	1	3	0	1	7	0
3	18	1	2	0	2	9	0
3	19	1	1	1	2	8	0
3	20	1	1	0	3	9	0
3	21	1	4	0	0	8	0
3	22	1	4	0	0	8	0
3	23	1	4	0	0	8	0
3	24	1	4	0	0	8	0
3	25	0	6	0	2	7	0
3	26	0	5	1	2	5	0

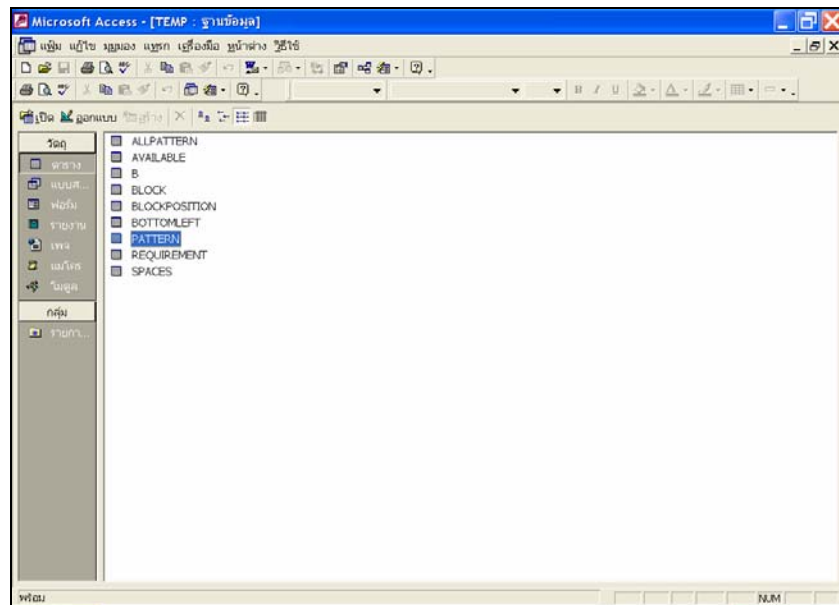
รูป 4.9 แสดงข้อมูลในตาราง ALLPATTERN

- โปรแกรม Excel Solver ทำการประมวลผลเพื่อหารูปแบบการจัดวางชิ้นงานที่เหมาะสม โดยที่สอดคล้องกับปริมาณความต้องการจากรูปแบบการจัดวางชิ้นงานทั้งหมดดังแสดงในรูปที่ 4.10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	QUANTITY	LEVELS	PATTERNNO	BLOCK1	BLOCK2	BLOCK3	BLOCK4	BLOCK5	BLOCK6	BLOCK7	BLOCK8	BLOCK9	BLOCK10	PERCENTAGE	AREA	MARK
2	28	2	15	0	0	9	0	0	0	0	0	0	0	97	2540000	1
3	0	4	156	0	0	6	2	8	0	0	0	0	0	97	2520000	0
4	29	3	52	0	0	0	14	1	0	0	0	0	0	96	2560000	1
5	63	1	3	0	8	0	0	6	0	0	0	0	0	95	2660000	1
6	50	0	1	2	0	1	0	6	0	0	0	0	0	85	2540000	1
7	0	4	154	0	0	7	0	9	0	0	0	0	0	96	2470000	0
8	0	3	46	0	0	6	4	2	0	0	0	0	0	96	2490000	0
9	0	2	8	0	7	0	1	6	0	0	0	0	0	89	2440000	0
10	0	1	2	1	4	0	0	8	0	0	0	0	0	91	2820000	0
11	0	4	155	0	0	6	3	5	0	0	0	0	0	96	2660000	0
12	0	3	45	0	0	7	3	0	0	0	0	0	0	95	2530000	0
13	0	2	14	0	1	6	1	4	0	0	0	0	0	91	2390000	0
14	0	4	153	0	0	7	1	6	0	0	0	0	0	96	2730000	0
15	0	3	44	0	0	8	1	1	0	0	0	0	0	95	2480000	0
16	0	2	9	0	6	1	0	8	0	0	0	0	0	90	2660000	0
17	0	4	173	0	0	3	6	10	0	0	0	0	0	96	2490000	0
18	0	3	49	0	0	3	9	1	0	0	0	0	0	95	2430000	0

รูปที่ 4.10 แสดงการประมวลผลใน EXCEL WORKSHEET โดยผ่าน SOLVER

7. ส่งข้อมูลกลับไปยังฐานข้อมูลในตาราง Pattern ของโปรแกรม ACCESS เพื่อนำไปแสดงผลออกมาในรูปของกราฟฟิคต่อไป ดังแสดงในรูปที่ 4.11 และ 4.12

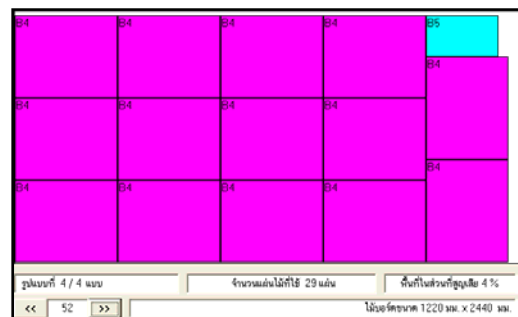
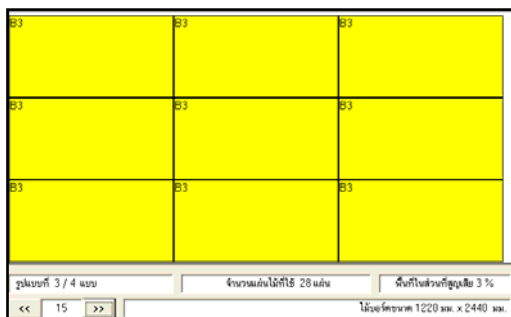
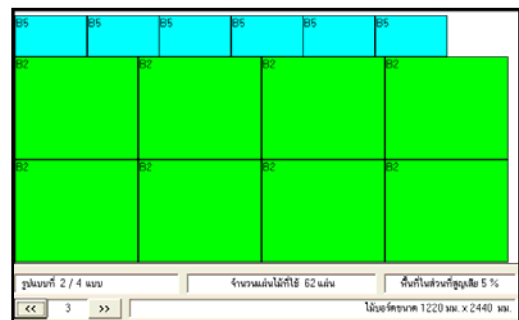
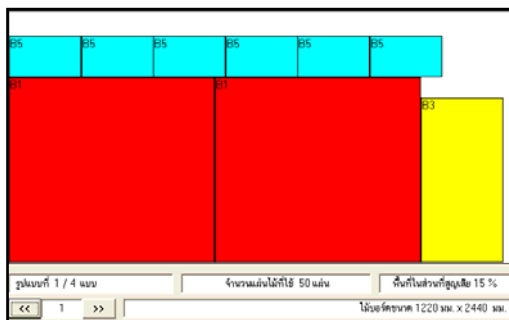


รูปที่ 4.11 แสดงตาราง PATTERN ในโปรแกรมเก็บฐานข้อมูล ACCESS

BLOCK2	BLOCK3	BLOCK4	BLOCK5	BLOCK6	BLOCK7	BLOCK8	BLOCK9	BLOCK10	PERCENTAGE	AREA	MARKER
0	9	0	0	0	0	0	0	0	96.748185972	2540000	1
0	6	2	8	0	0	0	0	0	96.748185972	2520000	0
0	0	14	1	0	0	0	0	0	98.412254770	2580000	1
8	0	0	6	0	0	0	0	0	94.732598764	2660000	1
0	1	0	6	0	0	0	0	0	85.326525128	2540000	1
0	7	0	9	0	0	0	0	0	96.412254770	2470000	0
0	6	4	2	0	0	0	0	0	96.076323569	2490000	0
7	0	1	6	0	0	0	0	0	91.373286751	2440000	0
4	0	0	8	0	0	0	0	0	89.357899543	2820000	0
0	6	3	5	0	0	0	0	0	98.412254770	2680000	0
0	7	3	0	0	0	0	0	0	95.404461166	2530000	0
1	6	1	4	0	0	0	0	0	90.701424348	2390000	0
0	7	1	6	0	0	0	0	0	96.076323569	2730000	0
0	8	1	1	0	0	0	0	0	95.068529965	2480000	0
6	1	0	8	0	0	0	0	0	90.029561946	2660000	0
0	3	6	10	0	0	0	0	0	96.076323569	2490000	0
0	3	9	1	0	0	0	0	0	95.068529965	2430000	0
0	4	1	4	0	0	0	0	0	89.357899543	2720000	0
0	7	2	3	0	0	0	0	0	95.740392368	2440000	0
0	5	5	3	0	0	0	0	0	94.398667562	2680000	0
5	2	0	7	0	0	0	0	0	88.349905939	2680000	0
0	3	7	7	0	0	0	0	0	95.740392368	2780000	0
0	2	10	2	0	0	0	0	0	93.388873959	2680000	0
4	3	0	6	0	0	0	0	0	86.670249933	2630000	0
0	8	0	4	0	0	0	0	0	95.404461166	2540000	0
0	4	7	1	0	0	0	0	0	92.381080355	2680000	0

รูป 4.12 แสดงข้อมูลในตาราง PATTERN

8. โปรแกรมแสดงผลออกมาในรูปกราฟโดยจะเลือกรูปแบบที่เหมาะสมภายใต้เงื่อนไข
ประหยัดและเพียงพอต่อความต้องการที่จะใช้งานดังรูปที่ 4.13



รูปที่ 4.13 แสดงการแสดงผลกราฟฟิกของโปรแกรม

4.5 ผลทดสอบการทำงานของโปรแกรม

4.5.1 รูปแบบในการตัดแบบกิโลดินตั้งตรงตัวอย่างที่ 1

ตัวอย่างชิ้นงานทดสอบที่ใช้วัสดุคือ MDF 2.6 มีขนาดและปริมาณที่ต้องการดังตารางที่ 4.2

ตารางที่ 4.2 แสดงข้อมูลที่ต้องการทดสอบของตัวอย่างที่ 1

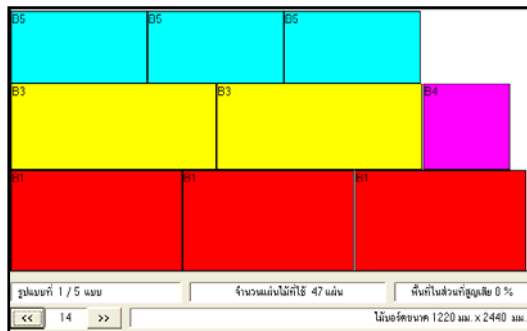
ลำดับ	ชื่อ	ความกว้าง	ความยาว	พื้นที่	ปริมาณที่ต้องการ
1	B1	470	805	378350	204
2	B2	460	640	294400	204
3	B3	404	965	389860	204
4	B4	404	470	189880	102
5	B5	340	640	217600	554

ผลจากการทดสอบโปรแกรม Cutting Stock ในการคำนวณหามูลค่าโดยรวมสูงสุดสรุปได้ตามตารางที่ 4.3 ซึ่งจะแสดงข้อมูลของปริมาณชิ้นส่วนที่ได้จากการคำนวณและปริมาณชิ้นงานที่เกิน

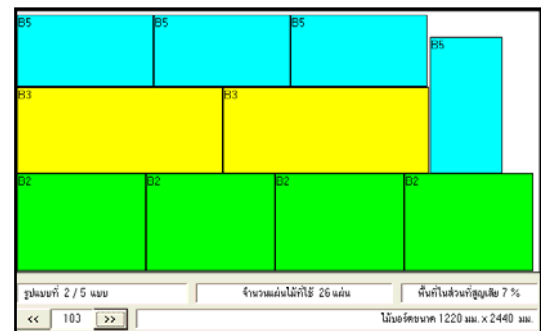
ตารางที่ 4.3 แสดงข้อมูลผลจากการทดสอบด้วยโปรแกรม

ลำดับ	ชื่อ	ความกว้าง	ความยาว	ปริมาณที่ต้องการ	ปริมาณที่คำนวณได้	ปริมาณที่เกิน
1	B1	470	805	204	205	1
2	B2	460	640	204	204	-
3	B3	404	965	204	205	1
4	B4	404	470	100	101	1
5	B5	340	640	552	553	1

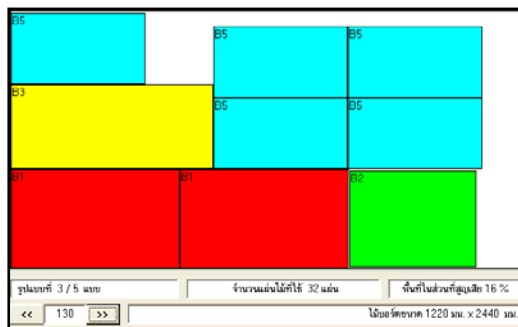
การแสดงผลของการจัดวางชิ้นงานที่คำนวณได้จากโปรแกรม Cutting Stock ทั้งหมด 5 รูปแบบดัง
แสดงในรูปที่ 4.14



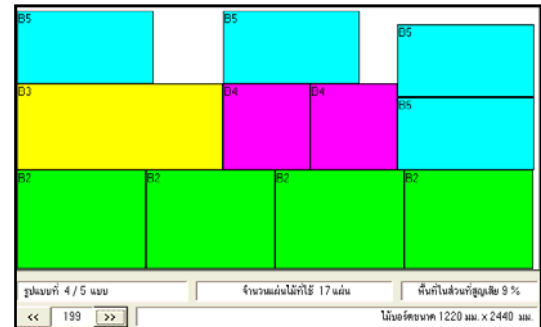
รูปแบบการจัดเรียงรูปแบบที่ 1



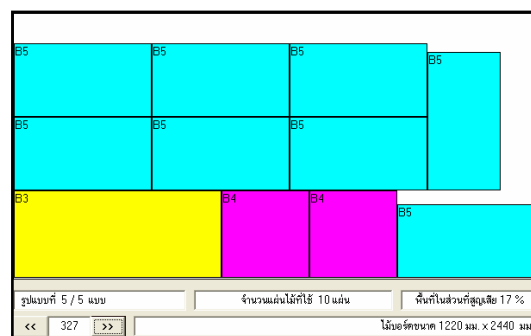
รูปแบบการจัดเรียงรูปแบบที่ 2



รูปแบบการจัดเรียงรูปแบบที่ 3



รูปแบบการจัดเรียงรูปแบบที่ 4



รูปแบบการจัดเรียงรูปแบบที่ 5

รูปที่ 4.14 แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 1

ผลสรุปของรูปแบบการจัดเรียงแต่ละรูปแบบที่ได้ดังแสดงในตารางที่ 4.4 โดยจะแสดงถึงปริมาณวัตถุดิบที่ต้องใช้ และ % พื้นที่สูญเสียในแต่ละรูปแบบการจัดเรียง

ตารางที่ 4.4 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ

รูปแบบ	จำนวนไม้บอร์ดที่ต้องใช้ตัด (แผ่น)	% พื้นที่สูญเสีย
1	47	8 %
2	26	7 %
3	32	16 %
4	17	9 %
5	10	17 %
สรุป	132	10.55%

ผลสรุปของรูปแบบการเปรียบเทียบผลการทดลองตัวอย่างที่ 1 กับผลที่ได้จากหน่วยการผลิตดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 1 กับผลที่ได้จากหน่วยการผลิต

แหล่งที่มาข้อมูล	ปริมาณไม้ที่ต้องเบิกมาใช้งาน	เปอร์เซ็นต์การสูญเสีย	เวลาที่ใช้ในการออกแบบ	ต้นทุนวัตถุดิบ
หน่วยการผลิต	144	12.3%	1-2 ชม.	64,800 บาท
โปรแกรมคอมพิวเตอร์	132	10.553%	1-5 นาที	59,400 บาท

ที่มาของการคำนวณ

ปริมาณไม้ที่ต้องเบิกมาใช้งานได้จากการประมวลผลของโปรแกรมโดยอาศัยการประยุกต์แบบจำลองที่ได้สร้างขึ้นดังที่ได้กล่าวมาแล้วข้างต้น

เปอร์เซ็นต์การสูญเสียได้จาก

$$\begin{aligned}
 &= (\text{ผลรวมของปริมาณพื้นที่สูญเสียของรูปแบบที่คัดเลือก}) \times 100 / (\text{ปริมาณของพื้นที่ของไม้บอร์ดที่ต้องเบิก}) \\
 &= ((47 \times 0.08) + (26 \times 0.08) + (32 \times 0.08) + (17 \times 0.08) + (10 \times 0.08)) \times 100 / 132 \\
 &= (13.93) \times 100 / 132 \\
 &= 10.553\%
 \end{aligned}$$

เวลาที่ใช้ในการออกแบบได้จาก เวลาในการประมวลผลของโปรแกรม

ต้นทุนวัตถุดิบได้จาก

= ปริมาณของไม้บอร์ดที่ต้องเบิก x ราคาต่อหน่วยวัตถุดิบ

= 132 x 450

= 59,400 บาท

4.5.2 รูปแบบในการตัดแบบกิโอติงตรงตัวอย่างที่ 2

ตัวอย่างชิ้นงานทดสอบที่ใช้วัตถุดิบคือ HB 4 มีขนาดและปริมาณที่ต้องการดังตารางที่ 4.6

ตารางที่ 4.6 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ HB ความหนา 4 มม.

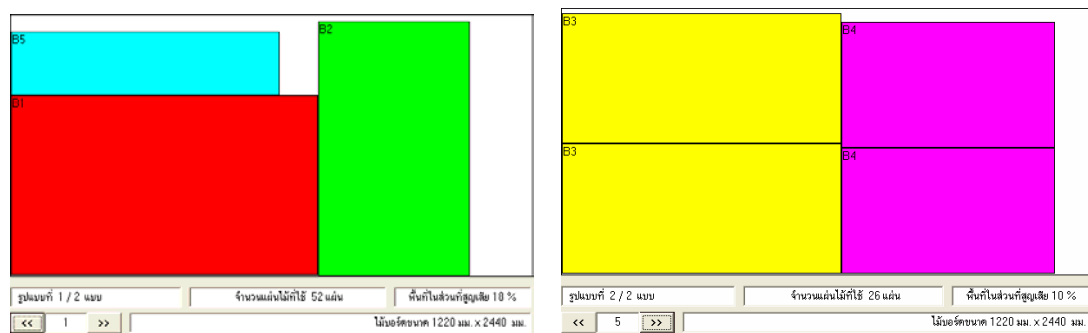
ลำดับ	ชื่อ	รหัสสินค้า	ชื่อชิ้นงาน	วัสดุ	ความ หนา	ความ กว้าง	ความ ยาว	จำนวนที่ ต้องการ
1	B1	102-042B	ปิดหลัง	HB	4	845	1438	52
2	B2	102-032B	ปิดหลัง	HB	4	710	1194	52
3	B3	102-147B	ปิดบน	HB	4	600	1287	52
4	B4	102-230B	ปิดหลัง	HB	4	581	985	52
5	B5	102-146B	ปิดหลัง	HB	4	300	1260	52

ผลจากการทดสอบโปรแกรม Cutting Stock ในการคำนวณหามูลค่าโดยรวมสูงสุดสรุปได้ตามตารางที่ 4. 7 ซึ่งจะแสดงข้อมูลของปริมาณชิ้นส่วนที่ได้จากการคำนวณและปริมาณชิ้นงานที่เกิน

ตารางที่ 4.7 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 2 ด้วยโปรแกรม

ลำดับ	ความกว้าง	ความยาว	ปริมาณที่ต้องการ	ปริมาณที่ คำนวณได้	ปริมาณที่ เกิน
1	845	1438	52	52	-
2	710	1194	52	52	-
3	600	1287	52	52	-
4	581	985	52	52	-
5	300	1260	52	52	-

การแสดงผลของการจัดวางชิ้นงานที่คำนวณได้จากโปรแกรม Cutting Stock ทั้งหมด 2 รูปแบบดัง
แสดงในรูปที่ 4.15



รูปแบบการจัดเรียงชิ้นงานแบบที่ 1

รูปแบบการจัดเรียงชิ้นงานแบบที่ 2

รูปที่ 4.15 แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 2

ผลสรุปของรูปแบบการจัดเรียงแต่ละรูปแบบที่ได้ดังแสดงในตารางที่ 4.8 โดยจะแสดงถึงปริมาณ
วัตถุดิบที่ต้องใช้ และ % พื้นที่สูญเสียในแต่ละรูปแบบการจัดเรียง

ตารางที่ 4.8 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ

รูปแบบ	จำนวนไม้บอร์ดที่ต้องใช้ตัด (แผ่น)	% พื้นที่สูญเสีย
1	52	18 %
2	26	10 %
สรุป	78	15.33%

ผลสรุปของรูปแบบการเปรียบเทียบผลการทดลองตัวอย่างที่ 1 กับผลที่ได้จากหน่วยการผลิตดัง
แสดงในตารางที่ 4.9

ตารางที่ 4.9 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 2 กับผลที่ได้จากหน่วยการผลิต

แหล่งที่มาข้อมูล	ปริมาณไม้ที่ต้องเบิก มาใช้งาน	เปอร์เซ็นต์การ สูญเสีย	เวลาที่ใช้ใน การออกแบบ	ต้นทุน วัตถุดิบ
หน่วยการผลิต	78	15.33%	1-2 ชม.	8,970 บาท
โปรแกรมคอมพิวเตอร์	78	15.33%	1-5 นาที	8,970 บาท

ที่มาของการคำนวณ

ปริมาณไม้ที่ต้องเบิกมาใช้งานได้จาก การประมวลผลของโปรแกรมโดยอาศัยการประยุกต์แบบจำลองที่ได้สร้างขึ้นดังที่ได้กล่าวมาแล้วข้างต้น

เปอร์เซ็นต์การสูญเสียได้จาก

$$\begin{aligned} &= (\text{ผลรวมของปริมาณพื้นที่สูญเสียของรูปแบบที่คัดเลือก}) \times 100 / (\text{ปริมาณของพื้นที่ของไม้บอร์ดที่ต้องเบิก}) \\ &= ((52 \times 0.18) + (26 \times 0.10)) \times 100 / 78 \\ &= (11.96) \times 100 / 78 \\ &= 15.33\% \end{aligned}$$

เวลาที่ใช้ในการออกแบบได้จาก เวลาในการประมวลผลของโปรแกรม

ต้นทุนวัตถุดิบได้จาก

$$\begin{aligned} &= \text{ปริมาณของไม้บอร์ดที่ต้องเบิก} \times \text{ราคาต่อหน่วยวัตถุดิบ} \\ &= 78 \times 115 \\ &= 8,970 \text{ บาท} \end{aligned}$$

4.5.3 รูปแบบในการตัดแบบกียอดินตั้งตรงตัวอย่างที่ 3

ตัวอย่างชิ้นงานทดสอบที่ใช้วัตถุดิบคือ MDF 18 มีขนาดและปริมาณที่ต้องการดังตารางที่ 4.10

ตารางที่ 4.10 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ MDF ความหนา 18 มม.

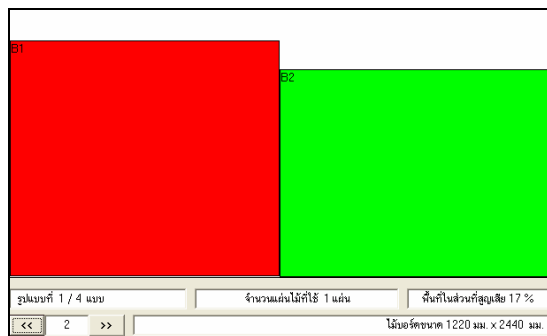
ลำดับ	ชื่อ	รหัสสินค้า	ชื่อชิ้นงาน	วัสดุ	ความหนา	ความกว้าง	ความยาว	จำนวนที่ต้องการ
1	B1	102-242B	เสริมขอบหน้าโต๊ะตัวยาว	MDF	18	1080	1220	6
2	B2	102-242B	เสริมขอบหน้าโต๊ะตัวยาว	MDF	18	950	1220	11
3	B3	102-242B	เสริมขอบหน้าโต๊ะตัวเสริม	MDF	18	600	1220	11

ผลจากการทดสอบโปรแกรม Cutting Stock ในการคำนวณหามูลค่าโดยรวมสูงสุดสรุปได้ตามตารางที่ 4. 11 ซึ่งจะแสดงข้อมูลของปริมาณชิ้นส่วนที่ได้จากการคำนวณและปริมาณชิ้นงานที่เกิน

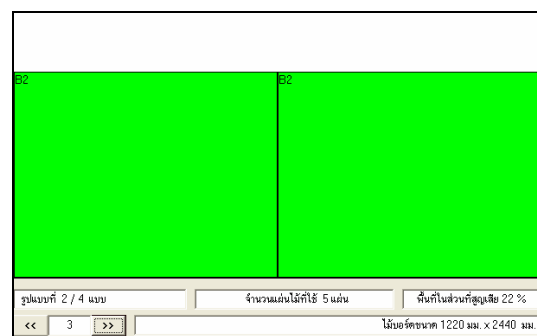
ตารางที่ 4.11 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 3 ด้วยโปรแกรม

ลำดับ	ความกว้าง	ความยาว	ปริมาณที่ต้องการ	ปริมาณที่คำนวณได้	ปริมาณที่เกิน
1	1080	1220	6	6	-
2	950	1220	11	12	1
3	600	1220	11	12	1

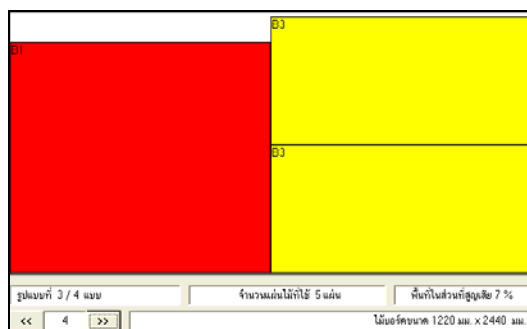
การแสดงผลของการจัดวางชิ้นงานที่คำนวณได้จากโปรแกรม Cutting Stock ทั้งหมด 5 รูปแบบดังแสดงในรูปที่ 4.16



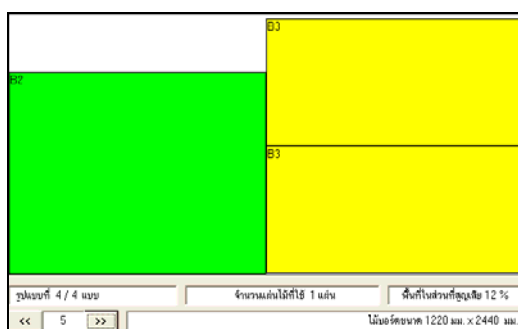
รูปแบบการจัดเรียงชิ้นงานแบบที่ 1



รูปแบบการจัดเรียงชิ้นงานแบบที่ 2



รูปแบบการจัดเรียงชิ้นงานแบบที่ 3



รูปแบบการจัดเรียงชิ้นงานแบบที่ 4

รูปที่ 4.16 แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 3

ผลสรุปของรูปแบบการจัดเรียงแต่ละรูปแบบที่ได้ดังแสดงในตารางที่ 4.12 โดยจะแสดงถึงปริมาณวัตถุดิบที่ต้องใช้ และ % พื้นที่สูญเสียในแต่ละรูปแบบการจัดเรียง

ตารางที่ 4.12 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ

รูปแบบ	จำนวนไม้บอร์ดที่ต้องใช้ตัด (แผ่น)	% พื้นที่สูญเสีย
1	1	17 %
2	5	22 %
3	5	7 %
4	1	12 %
สรุป	12	14.5%

ผลสรุปของรูปแบบการเปรียบเทียบผลการทดลองตัวอย่างที่ 3 กับผลที่ได้จากหน่วยการผลิตดังแสดงในตารางที่ 4.13

ตารางที่ 4.13 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 3 กับผลที่ได้จากหน่วยการผลิต

แหล่งที่มาข้อมูล	ปริมาณไม้ที่ต้องเบิกมาใช้งาน	เปอร์เซ็นต์การสูญเสีย	เวลาที่ใช้ในการออกแบบ	ต้นทุนวัตถุดิบ
หน่วยการผลิต	12	14.5%	1-2 ชม.	5,400 บาท
โปรแกรมคอมพิวเตอร์	12	14.5%	1-5 นาที	5,400 บาท

ที่มาของการคำนวณ

ปริมาณไม้ที่ต้องเบิกมาใช้งานได้จากการประมวลผลของโปรแกรมโดยอาศัยการประยุกต์แบบจำลองที่ได้สร้างขึ้นดังที่ได้กล่าวมาแล้วข้างต้น

เปอร์เซ็นต์การสูญเสียได้จาก

$$\begin{aligned}
 &= (\text{ผลรวมของปริมาณพื้นที่สูญเสียของรูปแบบที่คัดเลือก}) \times 100 / (\text{ปริมาณของพื้นที่ของไม้บอร์ดที่ต้องเบิก}) \\
 &= ((1 \times 0.17) + (5 \times 0.22) + (5 \times 0.07) + (1 \times 0.12)) \times 100 / 12 \\
 &= (1.74) \times 100 / 12 \\
 &= 14.5\%
 \end{aligned}$$

เวลาที่ใช้ในการออกแบบได้จาก เวลาในการประมวลผลของโปรแกรม

ต้นทุนวัตถุดิบได้จาก

= ปริมาณของไม้บอร์ดที่ต้องเบิก x ราคาต่อหน่วยวัตถุดิบ

= 12 x 450

= 5,400 บาท

4.5.4 รูปแบบในการตัดแบบกิโอตินตั้งตรงตัวอย่างที่ 4

ตัวอย่างการตัดไม้บอร์ดซึ่งพื้นที่(WXH) = 1220 x 1440 ตร.มม. วัตถุดิบคือ MDF 2.6 มีขนาดและปริมาณที่ต้องการดังตารางที่ 4.14

ตารางที่ 4.14 แสดงข้อมูลรายละเอียดของชิ้นงานที่ใช้วัตถุดิบ MDF ความหนา 2.6 มม.

ลำดับ	รหัสสินค้า	ชื่อชิ้นงาน	วัสดุ	ความ หนา	ความ กว้าง	ความ ยาว	จำนวนที่ ต้องการ
1	102-242B	เสริมหน้าโต๊ะ	MDF	2.6	1176	1440	168
2	102-272B	เสริมหน้าโต๊ะ	MDF	2.6	779	1538	220
3	102-272B	เสริมแผ่นรีฟ	MDF	2.6	630	1538	104
4	102-032B	ท็อป	MDF	2.6	630	1290	240
5	102-229B	เสริมแผ่นท็อปบน	MDF	2.6	505	906	1040
6	102-032B	พื้นลิ้นชักกลาง	MDF	2.6	430	1060	208
7	102-229B	พื้นลิ้นชัก	MDF	2.6	430	700	156
8	102-032B	พื้นลิ้นชักบนข้าง	MDF	2.6	390	490	1040
9	102-032B	พื้นลิ้นชักบนกลาง	MDF	2.6	300	490	204

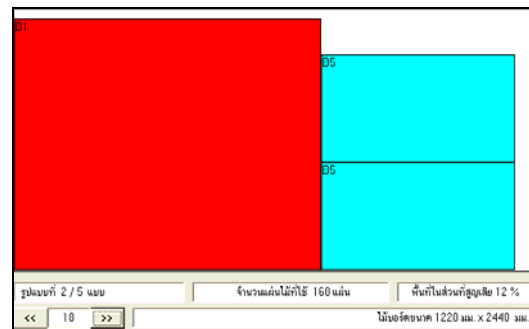
ตารางที่ 4.15 แสดงข้อมูลผลจากการทดสอบตัวอย่างที่ 4 ด้วยโปรแกรม

ลำดับ	ความกว้าง	ความยาว	ปริมาณที่ต้องการ	ปริมาณที่ คำนวณได้	ปริมาณที่ เกิน
1	1176	1440	168	168	-
2	779	1538	220	220	-
3	630	1538	104	104	-
4	630	1290	240	240	-
5	505	906	1040	1212	172
6	430	1060	208	220	12
7	430	700	156	156	-
8	390	490	1040	1056	16
9	300	490	204	220	16

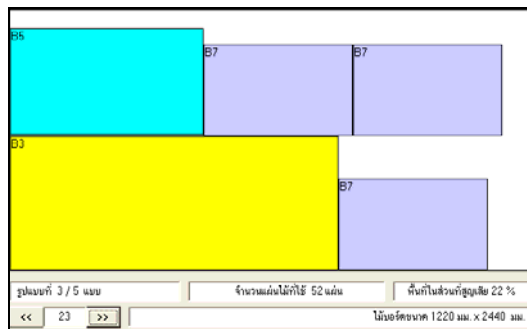
การแสดงผลของการจัดวางชิ้นงานที่คำนวณได้จากโปรแกรม Cutting Stock ทั้งหมด รูปแบบดัง
แสดงในรูปที่ 4.17



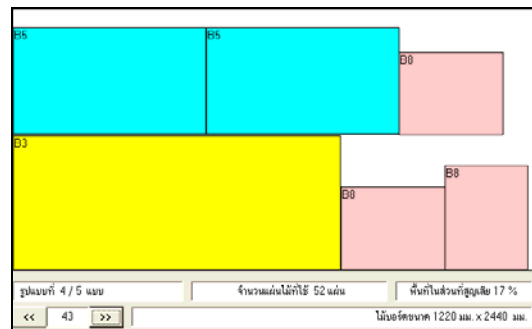
รูปแบบการจัดเรียงชิ้นงานแบบที่ 1



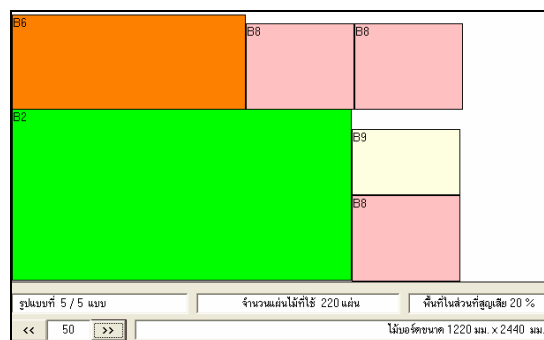
รูปแบบการจัดเรียงชิ้นงานแบบที่ 2



รูปแบบการจัดเรียงชิ้นงานแบบที่ 3



รูปแบบการจัดเรียงชิ้นงานแบบที่ 4



รูปแบบการจัดเรียงชิ้นงานแบบที่ 5

รูปที่ 4.17 แสดงการแสดงผลกราฟฟิคของตัวอย่างที่ 4

ผลสรุปของรูปแบบการจัดเรียงแต่ละรูปแบบที่ได้แสดงในตารางที่ 4.16 โดยจะแสดงถึงปริมาณวัสดุดิบที่ต้องใช้ และ % พื้นที่สูญเสียในแต่ละรูปแบบการจัดเรียง

ตารางที่ 4.16 แสดงผลสรุปของการคำนวณที่ได้ของรูปแบบการจัดเรียงแต่ละรูปแบบ

รูปแบบ	จำนวนไม้bordที่ต้องใช้ตัด (แผ่น)	% พื้นที่สูญเสีย
1	240	20%
2	168	12%
3	52	22%
4	52	17%
5	220	20%
รวม	732	18.09%

ผลสรุปของรูปแบบการเปรียบเทียบผลการทดลองตัวอย่างที่ 4 กับผลที่ได้จากหน่วยการผลิต
แสดงในตารางที่ 4.17

ตารางที่ 4.17 แสดงการเปรียบเทียบผลการทดลองตัวอย่างที่ 4 กับผลที่ได้จากหน่วยการผลิต

แหล่งที่มาข้อมูล	ปริมาณไม้ที่ต้องเบิก มาใช้งาน	เปอร์เซ็นต์การ สูญเสีย	เวลาที่ใช้ใน การออกแบบ	ต้นทุน วัตถุดิบ
หน่วยการผลิต	930	23%	1-4 ชม.	88,350 บาท
โปรแกรมคอมพิวเตอร์	732	18.09%	1-5 นาที	69,540 บาท

ที่มาของการคำนวณ

ปริมาณไม้ที่ต้องเบิกมาใช้งานได้จาก การประมวลผลของโปรแกรมโดยอาศัยการประยุกต์
แบบจำลองที่ได้สร้างขึ้นดังที่ได้กล่าวมาแล้วข้างต้น

เปอร์เซ็นต์การสูญเสียได้จาก

$$\begin{aligned}
 &= (\text{ผลรวมของปริมาณพื้นที่สูญเสียของรูปแบบที่คัดเลือก}) \times 100 / (\text{ปริมาณของพื้นที่ของ} \\
 &\quad \text{ไม้บอร์ดที่ต้องเบิก}) \\
 &= ((240 \times 0.2) + (168 \times 0.12) + (52 \times 0.22) + (52 \times 0.17) + (220 \times 0.2)) \times 100 / 732 \\
 &= (132.44) \times 100 / 732 \\
 &= 18.09\%
 \end{aligned}$$

เวลาที่ใช้ในการออกแบบได้จาก เวลาในการประมวลผลของโปรแกรม

ต้นทุนวัตถุดิบได้จาก

$$\begin{aligned}
 &= \text{ปริมาณของไม้บอร์ดที่ต้องเบิก} \times \text{ราคาต่อหน่วยวัตถุดิบ} \\
 &= 732 \times 95 \\
 &= 69,540 \text{ บาท}
 \end{aligned}$$

4.6 ผลการเปรียบเทียบประสิทธิภาพในด้านต่างๆของโปรแกรมทดสอบ

การเปรียบเทียบผลการทดลองที่ได้จากโปรแกรมและจากหน่วยการผลิตดังแสดงในตารางที่ 4.14 4.15 4.16 และ 4.17 โดยจะเปรียบเทียบผลในเชิงปริมาณและต้นทุนที่เกิดขึ้น ซึ่งจะแสดงผลที่แตกต่างดังแสดงในตารางที่ 4.18

ตารางที่ 4.18 แสดงเปอร์เซ็นต์ความแตกต่างในด้านประสิทธิภาพในด้านต่างๆ

	ความแตกต่างระหว่างโปรแกรมคอมพิวเตอร์กับหน่วยการผลิต			
	ตย.1	ตย.2	ตย.3	ตย.4
ปริมาณไม้ที่ต้องเบิกมาใช้งาน น้อยลง	12	-	-	198
เปอร์เซ็นต์การสูญเสียลดลง	1.747 %	-	-	4.91%
เวลาที่ใช้ในการออกแบบลดลง	~115 นาที	~115 นาที	~115 นาที	~115 นาที
ต้นทุนวัตถุดิบที่ลดได้	5,400 บาท	-	-	18,810 บาท

การหาความแตกต่างของประสิทธิภาพในด้านต่างๆสามารถคำนวณ ได้ดังนี้

ความแตกต่างของปริมาณไม้ที่ต้องเบิกมาใช้งาน

$$= | \text{ปริมาณไม้ที่เบิกมาใช้งานที่ได้จากโปรแกรม} - \text{ปริมาณไม้ที่เบิกมาใช้งานที่ได้จากหน่วยการผลิต} |$$

% การสูญเสีย

$$= | \% \text{ การสูญเสียที่เกิดขึ้นที่ได้โปรแกรม} - \% \text{ การสูญเสียที่เกิดขึ้นที่ได้จากหน่วยการผลิต} |$$

เวลาที่ใช้ในการออกแบบ

$$= | \text{เวลาที่ใช้ในการออกแบบที่ได้โปรแกรม} - \text{เวลาที่ใช้ในการออกแบบที่ได้จากหน่วยการผลิต} |$$

ต้นทุนวัตถุดิบออกแบบ

$$= | \text{ต้นทุนวัตถุดิบออกแบบที่ได้โปรแกรม} - \text{ต้นทุนวัตถุดิบการออกแบบที่ได้จากหน่วยการผลิต} |$$

บทที่ 5 สรุปและข้อเสนอแนะ

5.1 สรุปงานด้านโปรแกรมการตัดไม้บอร์ค

ปัญหาการตัดวัสดุไม้บอร์คนั้น เป็นปัญหาประเภทสองมิติซึ่งมีรูปแบบการตัดแบบกิโลตัน ซึ่งรูปแบบในการตัดที่เป็นไปได้ทั้งหมดมีจำนวนมากมาย ซึ่งจะขึ้นอยู่กับขนาดของปัญหา ทางผู้วิจัยได้พัฒนาวิธีการแก้ปัญหาขึ้นโดยการผสมผสานวิธีการทางคณิตศาสตร์ และ วิธีการทางฮิวริสติก ซึ่งขั้นตอนในการแก้ปัญหาที่มีลักษณะเป็น Muti-Phase มีแบ่งปัญหออกเป็น 3 ขั้นตอนหลักคือ

- การหารูปแบบที่เป็นไปได้ทั้งหมด
- การคัดเลือกรูปแบบที่ให้ประโยชน์สูงสุดในแต่ละรอบการทำงานมาประมวลผล
- การหาปริมาณรูปแบบที่เหมาะสมกับเงื่อนไขการผลิต

สำหรับการหาสถานะของรูปแบบในการตัดแบบกิโลตันดังตรงที่เป็นไปได้ทั้งหมด โดยการสร้างเงื่อนไขในการวางชิ้นงานซึ่งมีรูปแบบลงล่าง-ชิดซ้าย โดยปริมาณสถานะที่เกิดขึ้นนั้นเป็นสถานะที่สามารถนำไปใช้ได้จริง โดยจะแปรผันตามจำนวนรูปร่างของชิ้นงานที่นำมาจัดเรียง ขนาดชิ้นงาน ทดสอบ และอยู่ภายใต้ข้อจำกัดด้านหน่วยความจำของคอมพิวเตอร์และเวลาในการประมวลผล

การคัดเลือกรูปแบบที่ให้ประโยชน์สูงสุดในแต่ละรอบการทำงานมาประมวลผลเนื่องจากมีข้อจำกัดในการคำนวณตัวแปรการคำนวณของโปรแกรมรวมที่ใช้ทำให้ต้องมีการคัดเลือกรูปแบบที่ดีมาทำการประมวลผล โดยคัดเลือกมา 180 รูปแบบซึ่งในอนาคตหากความสามารถของเทคโนโลยีด้านคอมพิวเตอร์สูงขึ้นตัวแปรที่ใช้ในการประมวลผลจะมากขึ้นซึ่งจะส่งผลต่อประสิทธิภาพของโปรแกรมให้ดีขึ้นต่อไป

โปรแกรม Cutting Stock ที่ได้เขียนขึ้นมานั้นอาศัยการเชื่อมโยงกันระหว่าง ภาษา Visual Basic กับ Excel Solver ซึ่งมีข้อจำกัดในเรื่องของตัวแปรที่นำไปใช้ในการคำนวณ ดังนั้นทางผู้วิจัยจึงได้กำหนดเงื่อนไขจำกัดจำนวนชิ้นส่วนในการคำนวณเอาไว้ อย่างไรก็ตามโปรแกรมที่พัฒนาขึ้นนี้เป็นเพียงเครื่องมือที่ช่วยให้ผู้ใช้งานตัดสินใจ เพื่อให้เกิดความสะดวกและรวดเร็วขึ้น

5.2 ข้อเสนอแนะ

การวิเคราะห์เวลาในการประมวลผล

จากผลการประมวลผลการตัดพื้นที่สีเหลืองที่สามารถตัดได้ในแต่ละตัวอย่างนั้น เมื่อนำมาเปรียบเทียบกับผลที่เกิดจากการออกแบบของหน่วยผลิตพบว่ามีประสิทธิภาพในด้านต่างๆพบว่า

จุดเด่นของโปรแกรม

โปรแกรมที่ได้สร้างขึ้นสามารถที่จะช่วยลดเวลาในการออกแบบลง ใช้งานง่ายสะดวกมีการแสดงผลในรูปของรายงานพร้อมรูปภาพประกอบและรองรับต่อการเปลี่ยนแปลงเป้าหมายการผลิตได้เป็นอย่างดี

ข้อจำกัดของโปรแกรม

มีการเชื่อมโยงข้อมูลกับโปรแกรมร่วมหลายตัวทำให้ประสิทธิภาพในเรื่องความเสถียรภาพลดลง บางส่วน โปรแกรมที่ได้สร้างขึ้นยังไม่ได้รองรับกับรูปแบบการจัดวางชิ้นงานที่มีการประสานต่อไม้บอร์ด ซึ่งต้องได้รับการพัฒนาต่อไป

เอกสารอ้างอิง

1. Peerasak,P.(1972) **Loss reduction and control of cutting in the sheet metal forming industry**; Chul alongkorn University,pp.1- 196.
2. Kantarattanakul,W.(1996) **Calculation for sheet metal coil cutting in the steel pipe industry**; Chulalongkorn University,pp.1-124.
3. Sukto,S.(2000) **Maximizing the benefit of rectangular cutting patterns**; Kasetsart University,pp.1- 165.
4. Hafi, M. and Zissimopoulos ,V. (1996) Constrained two-dimensional cutting: an improvement of Chrisfofides and Whitlock's exact algorithm. **Journal of Operational Research** ,Vol. 94,pp.548-560.
5. Tangpagasit,T.(2001) **A comparison between centralized and decentralized one-dimensional cutting stock planning**; Kasetsart University,pp.1-127.
6. Cung,V. and Hafi ,M. and Le Cun ,V.(2000)Constrained two-dimensional stock problems a best-fist branch-and-bound algorithm; **International transactions in operational research**,Vol. 7,pp.185-210.
7. Leung,T.W. and Yung, C.H. and D.Troutt,M.(2001) **Application of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem**; computer & industrial engineering,Vol. 40,pp.201-214.
8. Fayard,D. and Zissimopoulos,V.(1995)An approximation algorithm for solving unconstrained two-dimensional Knapsack problems; **Euro. J. of Opl.Res.** ,Vol.84,pp.618-632.
- 9.Parada Daza ,V. and Gomes de Alvarenga ,A. and Jose de Diego(1995) Exact solutions for constrained two-dimensional cutting problems; **Euro. J. of Opl.Res.** ,Vol.84,pp.633-644.
10. Healy,P.and Creavin,M. and Kuusik,A. (1999)An optimal algorithm for rectangle placement; **Operations research letters** ,Vol.24,pp.73-80.
11. Riehme,J. and Scheithauer,G. and Terno,J.(1996) The solution of two-stage guillotine cutting stock problems having extremely varying order demands; **Euro. J. of Opl.Res.**,Vol.91,pp.543-552.
12. Yuen,BJ. And Richardson, K.V.(1995) Establishing the optimality of sequencing heuristics for cutting stock problems; Euro. J. of Opl.Res. 84,pp.590-598.
13. Hifi.,M.(1998) Exact algorithms for the guillotine strip cutting/ packing problem; Computers Ops Res. Vol.25, No.11, pp. 925-940.

14. Hifi,M.(1997) An improvement of viswanathan and bagchi's exact algorithm for constrained two-dimensional cutting stock; , **Computers Ops Res.** Vol.24, No.8, pp. 727-736.
15. J.Zak,E.(2000)Row and Column generation technique for a multistage cutting stock problem , **Computers Ops Res.**, Vol.29, pp. 1143-1156.
16. Klempour,R. and Kotowski,J. and Szlachcic,E.(1996) Interactive procedure in large-scale two-dimensional cutting stock problems, **J. of Computational and applied mathematics** ,Vol.66,pp.323-331.
17. Healy,P. and Creavin,M. and Kuusik,A.(1999)An optimal algorithm for rectangle placement ,**Operations research letters** ,Vol.24,pp.73-80.
18. Alvarez-valdes,R. and Parajon,A. and Tamarit,J.M.(2002) A tabu search algorithm for large-scale guillotine(un)constrained two-dimensional cutting problems; **computer&operations research**,pp.925-947.
19. Rode,M. and Rosenberg,O.(1987)**An analysis of heuristic trim loss algorithm,Engineering costs and Economics** ,Vol. 12,pp.71-78.
20. Baker,B.S.,E.G.Coffman and R.L. Rivest(1980) Orthogonal packing in two dimensions.Siam ,**J. Comput.**,Vol. 9 ,pp. 846-855.
21. Bartholdi,J.J III,J.H.V. Vate and J.Zhang.(1989)Expected performance of the shelf heuristic for 2 dimensional packing. **Operations Research Letters** ,Vol.8,pp.11-16.
- 22.Beasley,J.E.(1985) An exact two-dimension non-guillotine cutting tree search rocedure.**Operations Research**, Vol.33,pp. 49-64.
23. Bellman,R.(1957)**Dynamic Programming.Princeton University Press**,Princeton ,New Jersey,pp.1-342.
24. Bertsekas,D.P. (1987) Dynamic Programmong : **Deterministic and Stochastic Models. Pretic-Hall,Inc.**,Englewood Cliffs,New Jersey,pp.1-376
25. Berkey,J.O. and P.Y. Wang. (1987) Two-dimensional finite bin-packing algorithms., **J. Opl.Res.16**,pp. 215-221.
26. Cani,D.P. (1978) A note on the two-dimensional rectangular cutting stock problem. **J. of Opl.Res.Soc**,Vol. 29,pp.703-706.
27. Chambers,M.L. and R.G. Dyson.(1976) The cutting stock problem in the flat glass industry-selection of stock sizes, **J. of Opl. Res.**,Vol.27,pp.949-957.
28. Chumh,F.R.K.,M.R. Gray and D.S. Johnson. (1982) On packing two-dimensional bins Siam J, **Alg.Disc.**,Vol. 3,pp.66-76.

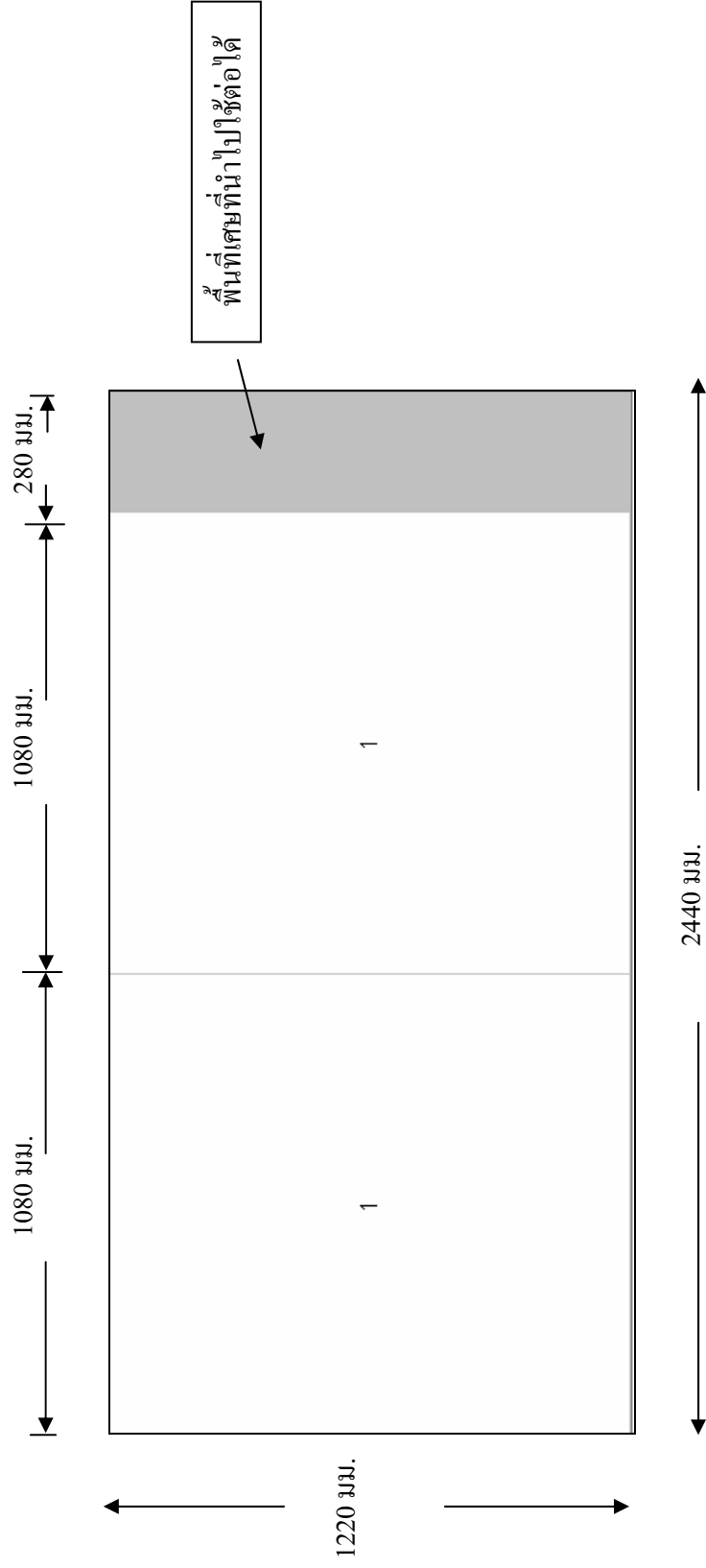
29. Coffman,E.G. and P.W. Shor.(1990) Average-Case analysis of cutting and packing in two dimensions. **Euro.J. of Opl. Res.**,Vol. 44 ,pp.134-144.
30. Dyckhoff,H. (1981) A new linear programming approach to the cutting stock problem. **Operations Research** ,Vol.29,pp.1092-1104.
31. Dyson ,R.G. and A.S. Gregory. (1974) The cutting stock problem in the flat glass industry. **J. of Opl.Res.Qly.**,Vol.25,pp. 41-53.
32. Gilmore,P.C. and R.E. Gomory. (1961) A linear programming approach to the cutting stock problem. **Operations Research**,Vol. 9,pp.849-859.
33. Golden,B.L. (1976) **Approaches to the cutting stock problem**. AIIE Transactions,Vol.8 ,pp. 265-274.
34. Hessler,R.W. (1980) A note on computation modifications to the Gilmore-Gomory cutting stock algorithm. **Operations Research**,Vol.28 ,pp.1001-1005.
35. Hessler,R.W. and P. Sweeney. (1991) Cutting stock problem and solution procedures. **Euro. J. of Opl.Res.**,Vol.54,pp.141-150
36. Hahn,S.G. (1968) On the optimal cutting of defective sheets. **Operations Research** ,Vol.16,pp.1100-1114.
37. Wang,P.Y.(1983) **Two algorithms for constrained two-dimensional cutting stock problem**.**Operations Research** ,Vol.31 ,pp.573-586.
38. Farley, A.A, (1992) Limiting the number of each piece in two-dimensional cutting stock patterns; **Mathl. Comput. Modelling** ,Vol.16, No.1, pp.75-87.
39. Faina,L.(1999) An application of simulated annealing to the cutting stock problem; **Euro. J. of Opl.Res.**,Vol.114,pp.42-556.
40. Christofides,N. and S. Eilon. (1971) **The loading problem**.**Management Sci**, Vol.17,pp.259-268.
41. Westerlund,T. and Isaksson,J. and Harjunkoski,I.(1998) Solving a two-dimensional trim-loss problem with MILP; **Euro. J. of Opl.Res.**,Vol 104,pp.574-581.

ภาคผนวก

ข

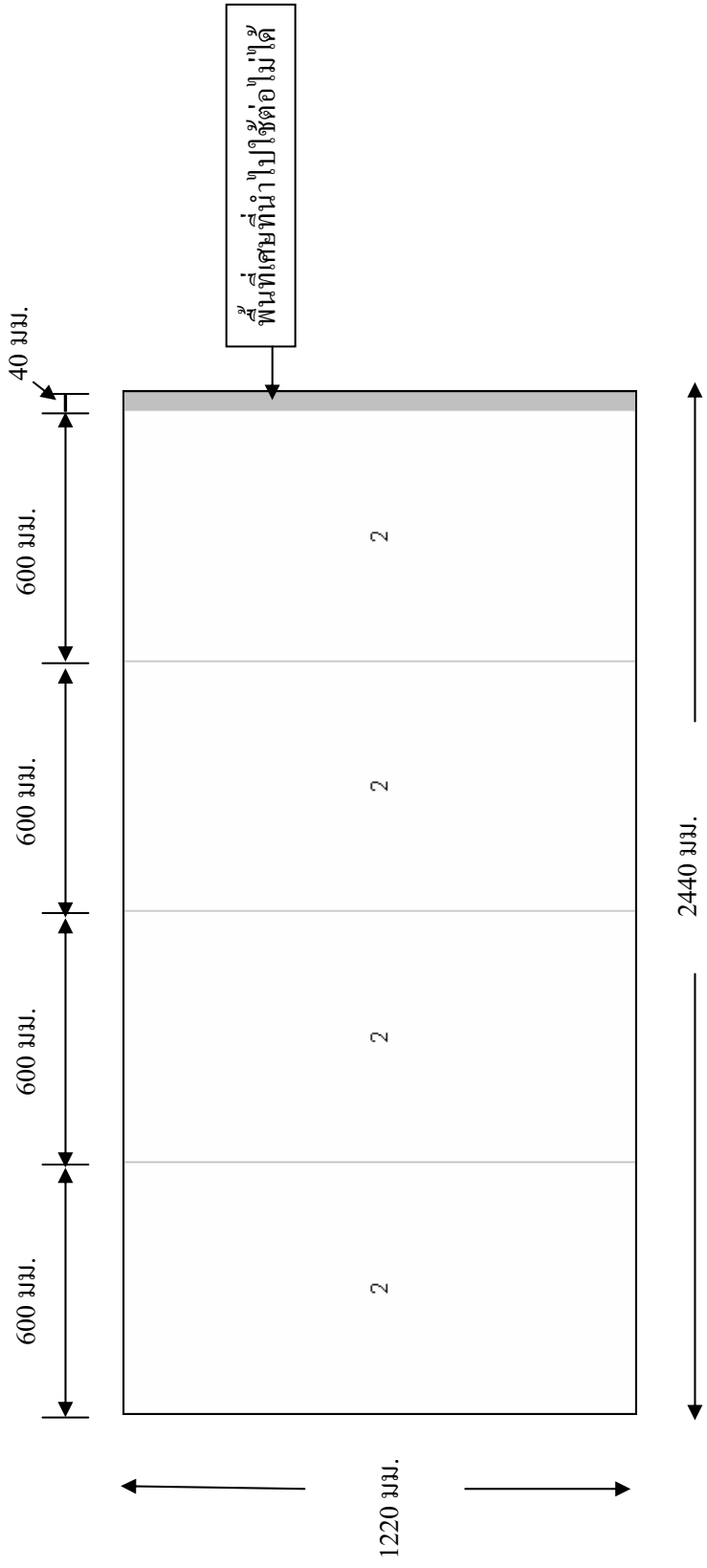
ตัวอย่างการรูปแบบการจัดวางชิ้นงานของหน่วยวางแผนการผลิต

ตัวอย่างที่ 1 ชิ้นงานหน้าโต๊ะ MDF 18 ขนาด 1080 มม. x 1220 มม.



รูปที่ ข.1 แสดงพื้นที่ในแต่ละส่วนของรูปแบบการจัดวางตัวอย่างที่ 1

ตัวอย่างที่ 2 ชิ้นงานเสริมขอบหน้าโต๊ะตัวเสริม MDF 18 ขนาด 600 มม. x 1220 มม.



รูปที่ จ.2 แสดงพื้นที่ในแต่ละส่วนของรูปแบบการจัดวางตัวอย่างที่ 2

ภาคผนวก

ก

แสดงตัวอย่างการวิเคราะห์เพื่อหาต้นทุนที่เกิดจากการวางแผนงานตัด
ในหน่วยการผลิต

ตัวอย่างการวิเคราะห์เพื่อหาต้นทุนที่เกิดจากการวางแผนงานตัดในหน่วยการผลิต

วิธีการคำนวณต้นทุนที่เกิดจากปริมาณเศษที่เกิดขึ้นในแต่ละรูปแบบการจัดวางที่เกิดขึ้นในแต่ละเดือนมีดังนี้

1. หาผลรวมของพื้นที่ไม้ที่เกิดขึ้นในแต่ละรูปแบบ โดยแยกประเภทวัตถุดิบ ความหนา พื้นที่เศษไม้ที่นำไปใช้ได้ พื้นที่เศษไม้ที่นำไปใช้ต่อไม่ได้ ตลอดจนชิ้นส่วนที่ขาด ดังตัวอย่างในตาราง ก.1

ตารางที่ ก.1 แสดงพื้นที่ต่างๆที่ได้จากการวางแผนการตัด

รหัสสินค้า	รายการ	จำนวน (แผ่น)	เศษไม้ที่ใช้ไม่ได้ (คิดเป็นพื้นที่ ตร.มม.)	เศษที่นำไปใช้ต่อได้ (คิดเป็นพื้นที่ ตร.มม.)	ชิ้นส่วนที่ขาด (ตร.มม.)
641-270	MDF6	18	8,791,200		
920-282	MDF6	4	1,464,000	1,281,000	
348-210	MDF6	9	5,671,800	781,000	
348-211	MDF6	9	3,733,200	686,000	
348-215	MDF6	10	4,586,500		948,700
348-216	MDF6	10	6,152,000		882,000
348-230/231	MDF6	90	17,283,960		
875-410	MDF6	5	488,000		707,600
348-301	MDF6	4	2,314,500		16,833,600
348-350B	MDF6	19	4,560,000		
875-301	MDF6	5	915,000		17,242,200
348-350H	MDF6	8	5,241,600		
500-004	MDF6	7	2,619,600	115,600	
รวม	MDF6	198	63,821,360	2,863,600	36,614,100

* หมายเหตุ

- ชิ้นส่วนที่ขาด : เกิดจากการวางแผนการจัดวางชิ้นงาน โดยที่ปริมาณการตัดที่ไม่เพียงพอกับการผลิต (โดยจะนำไปวางแผนใหม่ภายหลัง)
- เศษไม้ที่ใช้ไม่ได้ : เกิดจากรูปแบบการจัดวางที่ได้ออกแบบขึ้น โดยเศษที่เกิดขึ้นนั้นไม่สามารถที่จะนำไปใช้ประโยชน์ได้อีกเป็นเศษทิ้ง
- เศษไม้ที่นำไปใช้ได้ : เกิดจากรูปแบบการจัดวางที่ได้ออกแบบขึ้น โดยเศษที่เกิดขึ้นนั้นสามารถที่จะนำไปใช้ประโยชน์ได้อีกได้อีก

2. การคิดต้นทุนที่เกิดขึ้นจากเศษไม้

ต้นทุนปริมาณเศษที่นำมาใช้ได้

$$= (\text{พื้นที่เศษที่นำมาใช้ได้} / \text{พื้นที่ไม้มาตรฐาน } 4 \times 8 \text{ ฟุต}) \times \text{ราคาต่อหน่วย}$$

ต้นทุนปริมาณเศษที่นำมาใช้ไม่ได้

$$= (\text{พื้นที่เศษที่นำมาใช้ไม่ได้} / \text{พื้นที่ไม้มาตรฐาน } 4 \times 8 \text{ ฟุต}) \times \text{ราคาต่อหน่วย}$$

ต้นทุนปริมาณชิ้นส่วนที่ขาด

$$= (\text{พื้นที่ชิ้นส่วนที่ขาด} / \text{พื้นที่ไม้มาตรฐาน } 4 \times 8 \text{ ฟุต}) \times \text{ราคาต่อหน่วย}$$

เนื่องจากไม้บอร์ดสามารถนำไปต่อเพลาะได้ ดังนั้นการประมาณต้นทุนที่เกิดขึ้นจากเศษไม้ที่นำมาใช้ได้และต้นทุนที่เกิดขึ้นส่วนที่ขาดจึงนำมาหักลบกัน ซึ่งผลจากการหักลบกันจะแสดงให้เห็นถึงต้นทุนจริงที่เกิดขึ้น ดังแสดงในตารางที่ ก.2 ก.3 และ ก.4

ตารางที่ ค.2 แสดงการต้นทุนที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 7

เดือน	รายการวัสดุ	ปริมาณเศษไม้ที่ใช้ไม่ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณเศษที่นำไปใช้ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณที่ขาด (คิดเป็นพื้นที่ ตร.มม.)	ต้นทุนต่อหน่วย (บาท/ 1 แผ่น)	ต้นทุนที่เกิดจาก เศษที่ใช้ไม่ได้ (บาท)	ต้นทุนรวม (ปริมาณเศษที่ใช้ได้-ปริมาณที่ขาด)
7	MDF2.6	437775510	478784173	336912132	95	15,295.00	4,560.00
7	MDF6	173644490	258065925	42207910	165	14,355.00	12,045.00
7	MDF9	10782000	37576880	817600	230	2,990.00	2,990.00
7	MDF16	110199600	163222600	21547724	390	21,450.00	18,720.00
7	MDF18	115196720	318240280	202924200	450	48,150.00	17,550.00
7	PB12	3402000	-	991700	197	-	197.00
7	PB16	109054720	271018560	8497600	215	19,780.00	19,135.00
7	PB18	515289636	336935675	327587263	250	28,500.00	1,000.00
7	PB19	14139840	41860640	23616400	250	3,750.00	1,750.00
7	PB22	8090240	19801568	12179456	332	2,324.00	996.00
7	PB25	155398000	207273510	158926900	378.5	26,495.00	6,434.50
รวมรายจ่าย						183,089.00	85,377.50

ตารางที่ ค.3 แสดงการต้นทุนที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 8

เดือน	รายการวัสดุ	ปริมาณเศษไม้ที่ใช้ไม่ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณเศษที่นำไปใช้ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณที่ขาด (คิดเป็นพื้นที่ ตร.มม.)	ต้นทุนต่อหน่วย (บาท/ 1 แผ่น)	ต้นทุนที่เกิดจาก เศษที่ใช้ไม่ได้ (บาท)	ต้นทุนรวม (ปริมาณเศษที่ใช้ได้-ปริมาณที่ขาด)
8	MDF2.6	547970600	950642350	382580078	95	17,575.00	18,145.00
8	MDF4		36636600	-	115	0.00	1,495.00
8	MDF6	44736880	4502780	65350120	165	2,640.00	3,465.00
8	MDF9	11863700	62560960	69214860	230	920.00	690.00
8	MDF19	158438100	298787880	246913900	450	24,300.00	8,100.00
8	MDF26	6690480	67990600	37840672	700	2,100.00	7,700.00
8	PB16	107144350	177479300	39821130	215	7,740.00	10,105.00
8	PB18	82581356	140280400	554402246	250	7,000.00	35,000.00
8	PB19	55624800	60433000	113064700	250	4,750.00	4,500.00
8	PB22	57987800	91959280	62077190	332	6,640.00	3,652.00
8	PB25	9750300	28182000	20247500	378.5	1,514.00	1,135.50
					รวมรายจ่าย	75,179.00	93,987.50

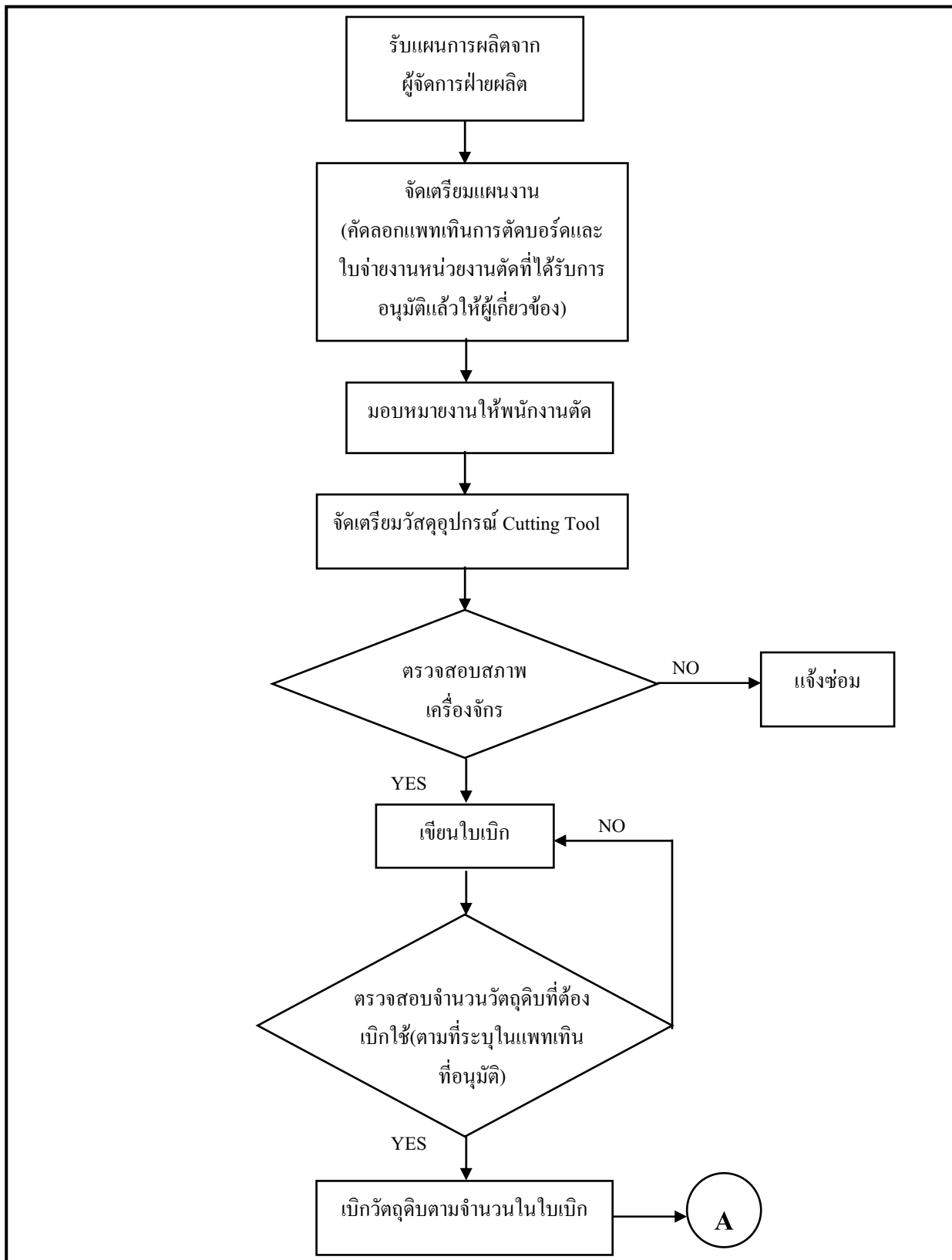
ตารางที่ ค.4 แสดงการต้นทุนที่เกิดจากรูปแบบการจัดวางของหน่วยการผลิตที่เกิดขึ้นในเดือนที่ 9

เดือน	รายการวัสดุ	ปริมาณเศษไม้ที่ใช้ไม่ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณเศษที่นำไปใช้ได้ (คิดเป็นพื้นที่ ตร.มม.)	ปริมาณที่ขาด (คิดเป็นพื้นที่ ตร.มม.)	ต้นทุนต่อหน่วย (บาท/ 1 แผ่น)	ต้นทุนที่เกิดจาก เศษที่ใช้ไม่ได้ (บาท)	ต้นทุนรวม (ปริมาณเศษที่ใช้ได้-ปริมาณที่ขาด)
9	MDF2.6	653760030	462323492	468446799	95	20,863.75	285.00
9	MDF6	63821360	2863600	36614100	165	3,537.53	1,980.00
9	MDF9	88280560	147827600	-	230	6,820.92	11,500.00
9	MDF12	6109600	1598200	1981280	300	615.72	300.00
9	MDF16	26308800	46556400	64576800	390	3,446.80	2,730.00
9	MDF18	370972320	190619510	233078225	450	56,079.53	6,750.00
9	MDF25	150000720	217062800	257534624	700	35,272.95	9,800.00
9	PB16	18942000	77852324	-	215	1,368.09	5,805.00
9	PB18	442468110	533317060	165385810	250	37,159.71	31,000.00
9	PB22	43159129	37342980	78058144	332	4,813.50	4,648.00
รวมรายจ่าย						169,978.50	74,798.00

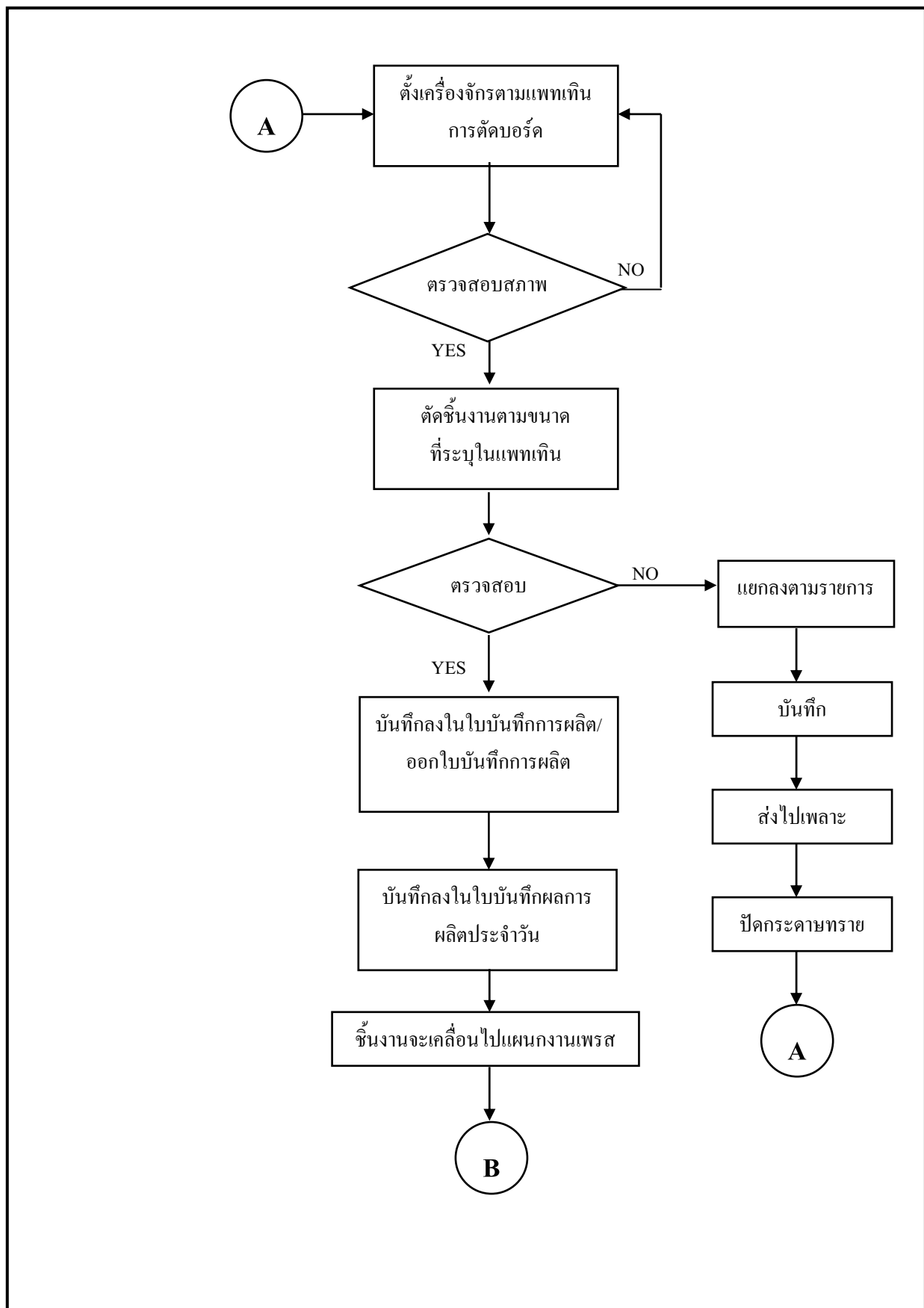
ภาคผนวก

ก

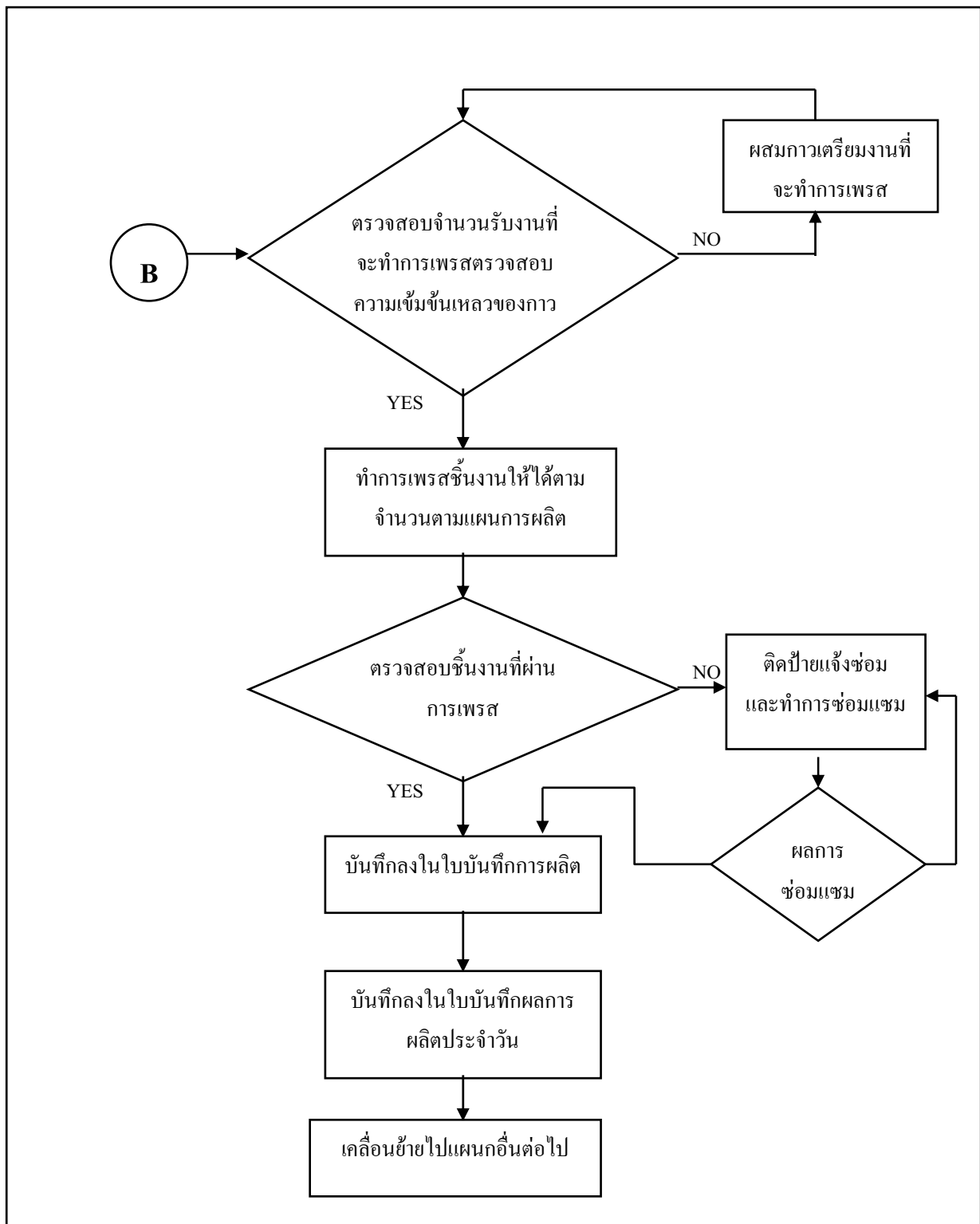
แสดงการแผนภาพการไหลของการควบคุมการตัดไม้บอร์ด



รูปที่ ก.1 FLOW CHART แสดงการควบคุมการตัดไม้บอร์ด



รูปที่ ก.2 FLOW CHART แสดงการควบคุมการตัดไม้บอร์ด



รูปที่ ก.3 FLOW CHART แสดงการควบคุมการเพรส

ภาคผนวก

ง

แสดงภาษาคอมพิวเตอร์ของโปรแกรม VBA

ภาษาคอมพิวเตอร์ของโปรแกรม VBA

โปรแกรมหลัก

Sub Run()

 Sheets("Sheet4").Select

 Cells.Select

 Selection.ClearContents

 Sheets("Sheet1").Select

 Cells.Select

 Selection.Copy

 Sheets("Sheet3").Select

 Cells.Select

 Range("A1").Activate

 ActiveSheet.Paste

 Rows("1:1").Select

 Selection.Insert Shift:=xlDown

 Selection.Insert Shift:=xlDown

 Selection.Insert Shift:=xlDown

 Selection.Insert Shift:=xlDown

 Columns("A:A").Select

 Selection.Insert Shift:=xlToRight

Macro2

Macro5

Macro6

Macro4

Macro1

solver

Macro8

Macro10

Macro11

End Sub

โปรแกรมย่อยที่1

Sub Macro1()

Range("Q4").Select

ActiveCell.FormulaR1C1 = "Index1"

Range("Q5").Select

ActiveCell.FormulaR1C1 = "=IF(RC[-1]*RC[-2]>0,RC[-1])"

Selection.AutoFill Destination:=Range("Q5:Q2000"), Type:=xlFillDefault

' Range("Q5:Q2000").Select

Range("P5:P2000").Select

Selection.Copy

Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _

False, Transpose:=False

Application.CutCopyMode = False

Range("B5:Q2000").Select

Range("Q5").Activate

ActiveWindow.ScrollRow = 18

ActiveWindow.ScrollRow = 1

Selection.Sort Key1:=Range("Q5"), Order1:=xlAscending, Key2:=Range("N5") _

, Order2:=xlDescending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _

False, Orientation:=xlTopToBottom

End Sub

โปรแกรมย่อยที่2

Sub Macro7()

Range("P5:P2000").Select

Selection.Copy

Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _

False, Transpose:=False

Application.CutCopyMode = False

End Sub

โปรแกรมย่อยที่3

Sub Macro8()

Sheets("Sheet4").Select

Cells.Select

Range("A1").Select

ActiveCell.FormulaR1C1 = "QUANTITY"

Range("B1").Select

ActiveCell.FormulaR1C1 = "LEVELS"

Range("C1").Select

ActiveCell.FormulaR1C1 = "PATTERNNO"

Range("D1").Select

ActiveCell.FormulaR1C1 = "BLOCK1"

Range("D1").Select

Selection.AutoFill Destination:=Range("D1:M1"), Type:=xlFillDefault

Range("D1:M1").Select

Range("N1").Select

ActiveCell.FormulaR1C1 = "PERCENTAGE"

Range("O1").Select

ActiveCell.FormulaR1C1 = "AREA"

Range("P1").Select

ActiveCell.FormulaR1C1 = "MARKER"

End Sub

โปรแกรมย่อยที่4

Sub solver()

Sheets("Sheet3").Select

SolverReset

SolverLPOptions MaxTime:=1000, Iterations:=1000, Precision:=0.000001, PivotTol:= _

0.000001, ReducedTol:=0.000001, StepThru:=False, Scaling:=False, AssumeNonneg:= _

True, BypassReports:=False

SolverAdd cellRef:="\$D\$3:\$M\$3", relation:=3, formulaText:="\$D\$2:\$M\$2"

SolverAdd cellRef:="\$A\$5:\$A\$200", relation:=4

```

SolverOk SetCell:="$B$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$A$5:$A$200", _
    Engine:=2, EngineDesc:="Standard Simplex LP"
SolverSolve userFinish:=True
End Sub

```

โปรแกรมย่อยที่ 5

```

Sub Macro10()
    Sheets("Sheet3").Select
    Range("R4").Select
    ActiveCell.FormulaR1C1 = "Marker"
    Range("R5").Select
    ActiveCell.FormulaR1C1 = "=IF(RC[-17]>0.99,1,0)"
    Selection.AutoFill Destination:=Range("R5:R2000"), Type:=xlFillDefault
    Range("R5:R2000").Select
    Selection.Copy
    Sheets("Sheet4").Select
    Range("P2").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    Sheets("Sheet3").Select
    Range("A5:O2000").Select
    Selection.Copy
    Sheets("Sheet4").Select
    Range("A2").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
End Sub

```

โปรแกรมย่อยที่7

Sub Macro4()

Sheets("Sheet2").Select

Range("E1:E10").Select

Selection.Copy

Range("R1").Select

Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _

False, Transpose:=True

Application.CutCopyMode = False

Selection.Copy

Sheets("Sheet3").Select

Range("D2").Select

ActiveSheet.Paste

End Sub

โปรแกรมย่อยที่8

Sub Macro11()

Sheets("Sheet1").Select

Cells.Select

Selection.ClearContents

Sheets("Sheet2").Select

Cells.Select

Selection.ClearContents

Sheets("Sheet3").Select

Cells.Select

Selection.ClearContents

End Sub

ภาคผนวก

จ

แสดงภาษาคอมพิวเตอร์ของโปรแกรม VB

ภาษาคอมพิวเตอร์ของโปรแกรม

1. โปรแกรมหลัก

```
Dim iFormLoad As Integer
Dim AlarmTime
Dim conn As New ADODB.Connection
Dim rsItem As New ADODB.Recordset
Dim Tape As New clsTape
Dim vntChannel As Variant
Dim rs As New ADODB.Recordset
Dim AccessDB, SpaceDB As Database
Dim RecSet As DAO.Recordset
Dim SQLSTR3 As String
Dim LEVELS, BlockOrderNo, MAXBLOCK, MAXCELL, REMAINNO,
PARENTSPACENO As Integer
Dim BLOCKAREA(10), BLOCKWIDTH(10), BLOCKLENGTH(10),
BLOCKRWIDTH(10), BLOCKRLENGTH(10), PLATEXBEGIN(3),
PLATEYBEGIN(3), PLATEXEND(3), PLATEYEND(3) As Integer
Dim PATTERNNO, TOPSPACE, SPACENO, AVAILABLE, BLOCKORDER,
NOTAVAILABLE, VBOTTOM, VLEFT, PLATERUNNO As Integer
Dim RIGHTSPACE As Integer
Dim PatternPictureFileName As String
Private OptionV As Integer
Private OptionT As Integer
Private OptionD As Integer

Private Sub ClearBtn_Click()
    Dim COUNTI As Integer
    For COUNTI = 1 To BlockOrderNo
        cell(COUNTI).Visible = False
    Next COUNTI
End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Private Sub optov_Click(Index As Integer)
    Dim i As Byte
    OptionD = Index
    For i = 0 To 1
        If Index = i Then
            optov(i).Picture = opcut(1).Picture
        Else
            optov(i).Picture = opcut(0).Picture
        End If
    Next i
    If OptionD = 0 Then
        Text2.Text = "5"
        Text3.Text = "1"
        Text2.Enabled = True
    End If
End Sub
```

```

        Text3.Enabled = True
Text2.BackColor = &H8000000E
Text3.BackColor = &H8000000E
End If
If OptionD = 1 Then
    Text2.Text = ""
    Text3.Text = ""
    Text2.Enabled = False
    Text3.Enabled = False
    Text2.BackColor = &HC0C000
    Text3.BackColor = &HC0C000
End If
End Sub

Private Sub opVText_Click(Index As Integer)
Dim i As Byte
    OptionD = Index
    For i = 0 To 1
        If Index = i Then
            optov(i).Picture = opcut(1).Picture
        Else
            optov(i).Picture = opcut(0).Picture
        End If
    Next i
    If OptionD = 0 Then
        Text2.Text = "5"
        Text3.Text = "1"
        Text2.Enabled = True
        Text3.Enabled = True
        Text2.BackColor = &H8000000E
        Text3.BackColor = &H8000000E
    End If
    If OptionD = 1 Then
        Text2.Text = ""
        Text3.Text = ""
        Text2.Enabled = False
        Text3.Enabled = False
        Text2.BackColor = &HC0C000
        Text3.BackColor = &HC0C000
    End If
End Sub

Private Sub CreateBox_Click()
    Call UPDATEALLPATTERN
End Sub

```

```

Private Sub Form_Load()

```

Dim i As Integer

***** การตั้งค่าเริ่มต้น *****

```
OptionV = 4
OptionT = 0
OptionD = 1
Call optText_Click(OptionV)
Call optTextta_Click(OptionT)
Call opVText_Click(OptionD)
```

```
file_path = App.Path
If Right$(file_path, 1) <> "\" Then file_path = file_path & "\"
txtExcelFile.Text = file_path & "Books.xls"
txtAccessFile.Text = file_path & "Temp.mdb"
```

```
Data1.DatabaseName = App.Path + "\TEMP.mdb"
frmproject.WindowState = 2
iFormLoad = True
Set AccessDB = Workspaces(0).OpenDatabase(App.Path & "\TEMP.mdb")
Set SpaceDB = Workspaces(0).OpenDatabase(App.Path & "\SPACE.mdb")
MAXCELL = 0
```

End Sub

Private Sub SetCaptionTime()

```
Caption = Format(Time, "Medium Time")
```

End Sub

Private Sub Form_Unload(Cancel As Integer)

CAP.Visible = False

```
MDIForm1.Tool.Buttons(2).Enabled = True
MDIForm1.Tool.Buttons(3).Enabled = True
MDIForm1.Tool.Buttons(4).Enabled = False
MDIForm1.Tool.Buttons(5).Enabled = False
MDIForm1.Tool.Buttons(7).Enabled = False
MDIForm1.Tool.Buttons(8).Enabled = False
MDIForm1.Tool.Buttons(9).Enabled = False
MDIForm1.mnu_Save.Enabled = False
MDIForm1.mnu_SaveAs.Enabled = False
MDIForm1.mnu_Print.Enabled = False
MDIForm1.mnu_Print_Preview = False
MDIForm1.mnu_Print_Setup = False
```

End Sub

```

Private Sub optTextta_Click(Index As Integer)
Dim i As Byte
OptionT = Index
For i = 0 To 2
    If Index = i Then
        otable(i).Picture = optable(1).Picture
    Else
        otable(i).Picture = optable(0).Picture
    End If
Next i
    If OptionT = 0 Then
OpenTable.Visible = False
DBGrid1.Visible = False
DBGrid2.Visible = True
    End If
    If OptionT = 1 Then
OpenTable.Visible = False
    DBGrid1.Visible = True
DBGrid2.Visible = False
    End If
    If OptionT = 2 Then
OpenTable.Enabled = True
    OpenTable.Visible = True
    DBGrid1.Visible = True
DBGrid2.Visible = False

    End If
End Sub

```

```

Private Sub optUnit_Click(Index As Integer)
Dim i As Byte
OptionV = Index
For i = 0 To 5
    If Index = i Then
        optUnit(i).Picture = optArt(1).Picture
    Else
        optUnit(i).Picture = optArt(0).Picture
    End If
Next i
    If OptionV = 0 Then
Combo1.Enabled = True
Combo2.Enabled = False
    End If
    If OptionV = 1 Then
Combo1.Enabled = True
Combo2.Enabled = False
    End If
    If OptionV = 2 Then
Combo1.Enabled = False
Combo2.Enabled = True

```

```

End If
    If OptionV = 3 Then
        Combo1.Enabled = False
        Combo2.Enabled = True
    End If
    If OptionV = 4 Then
        Combo1.Enabled = False
        Combo2.Enabled = True
    End If
    If OptionV = 5 Then
        Combo1.Enabled = False
        Combo2.Enabled = False
    End If
End Sub

Private Sub optText_Click(Index As Integer)
    Dim i As Byte
    OptionV = Index
    For i = 0 To 5
        If Index = i Then
            optUnit(i).Picture = optArt(1).Picture
        Else
            optUnit(i).Picture = optArt(0).Picture
        End If
    Next i
    If OptionV = 0 Then
        Combo1.Enabled = True
        Combo2.Enabled = False
    End If
    If OptionV = 1 Then
        Combo1.Enabled = True
        Combo2.Enabled = False
    End If
    If OptionV = 2 Then
        Combo1.Enabled = False
        Combo2.Enabled = True
    End If
    If OptionV = 3 Then
        Combo1.Enabled = False
        Combo2.Enabled = True
    End If
    If OptionV = 4 Then
        Combo1.Enabled = False
        Combo2.Enabled = True
    End If
    If OptionV = 5 Then
        Combo1.Enabled = False
        Combo2.Enabled = False
    End If
End Sub

```

```

Private Sub otable_Click(Index As Integer)
Dim i As Byte
    OptionT = Index
    For i = 0 To 2
        If Index = i Then
            otable(i).Picture = optable(1).Picture
        Else
            otable(i).Picture = optable(0).Picture
        End If
    Next i
        If OptionT = 0 Then
OpenTable.Visible = False
DBGrid1.Visible = False
DBGrid2.Visible = True
        End If
        If OptionT = 1 Then
OpenTable.Visible = False
DBGrid1.Visible = True
DBGrid2.Visible = False
        End If
        If OptionT = 2 Then
OpenTable.Enabled = True
OpenTable.Visible = True
DBGrid1.Visible = True
DBGrid2.Visible = False
        End If
End Sub

Private Sub PatternNoBox_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Call CREATEBLOCKTOSHOWPATTERN
        Call HIDEPATTERN
        Call SHOWPATTERN(CStr(PatternNoBox.Text))
    End If
End Sub

Private Sub Solver_Click()
Dim conn As ADODB.Connection
Dim rs As ADODB.Recordset
Dim rs1 As ADODB.Recordset
Dim excel_app As Excel.Application
Dim excel_sheet, excel_sheet1 As Excel.Worksheet
Dim mExcelFile As String
Dim mAccessFile As String
Dim mWorkSheet As String
Dim mTableName As String
Dim mDataBase As Database

AccessDB.Execute ("DELETE FROM ALLPATTERN")

```

AccessDB.Execute ("DELETE FROM PATTERN")

CAP.NextPattern.Visible = True

CAP.PatternNoBox.Text = " 0 "

CAP.Text4.Text = " "

CAP.Text5.Text = " "

CAP.Text7.Text = " "

CAP.Text9.Text = " "

CAP.Text8.Text = "0"

CAP.Waste.Text = " "

CAP.Quantity.Text = ""

Call calculate

*****การส่งข้อมูลจาก Access ไปยัง Excel

Screen.MousePointer = vbHourglass

DoEvents

' Open the Access database.

Set conn = New ADODB.Connection

conn.ConnectionString = _

 "Provider=Microsoft.Jet.OLEDB.4.0;" & _

 "Data Source=" & txtAccessFile.Text

conn.Open

' Select the Access data.

Set rs = conn.Execute("AllPattern")

Set rs1 = conn.Execute("Block")

' Create the Excel application.

Set excel_app = CreateObject("Excel.Application")

' Uncomment this line to make Excel visible.

excel_app.Visible = False

' Open the Excel workbook.

excel_app.Workbooks.Open txtExcelFile.Text

If Val(excel_app.Application.Version) >= 8 Then

 Set excel_sheet = excel_app.Worksheets(1)

 Set excel_sheet1 = excel_app.Worksheets(2)

 excel_sheet.Activate

 excel_sheet1.Activate

 Else

End If

' Use the Recordset to fill the table.

```

excel_sheet.Cells.CopyFromRecordset rs
excel_sheet1.Cells.CopyFromRecordset rs1

' Save the workbook.
excel_app.Application.Run "Run"
excel_app.ActiveWorkbook.Save

' Shut down.
' excel_app.Quit

rs.Close
rs1.Close
conn.Close
excel_app.Quit
Screen.MousePointer = vbNormal

***** การส่งข้อมูลจาก Excel ไปยัง Access

Call delete_Table
Call Excel2Access
End Sub

Private Sub UPDATEPATTERN(PATTERNNO1, LEVEL, B1, B2, B3, B4, B5, B6,
B7, B8, B9, B10 As Integer)
AccessDB.Execute ("INSERT INTO ALLPATTERN
(PATTERNNO,LEVEL,BLOCK1,BLOCK2,BLOCK3,BLOCK4,BLOCK5,BLOCK6
,BLOCK7,BLOCK8,BLOCK9,BLOCK10,AREA)" & _
"SELECT " & _
CStr(PATTERNNO1) + " AS Ex1," & _
CStr(LEVEL) + " AS Ex2," & _
CStr(B1) + " AS Ex3," & _
CStr(B2) + " AS Ex4," & _
CStr(B3) + " AS Ex5," & _
CStr(B4) + " AS Ex6," & _
CStr(B5) + " AS Ex7," & _
CStr(B6) + " AS Ex8," & _
CStr(B7) + " AS Ex9," & _
CStr(B8) + " AS Ex10," & _
CStr(B9) + " AS Ex11," & _
CStr(B10) + " AS Ex12," & _
CStr((BLOCKAREA(1) * Abs(B1)) + (BLOCKAREA(2) * Abs(B2)) +
(BLOCKAREA(3) * Abs(B3)) + (BLOCKAREA(4) * Abs(B4)) + (BLOCKAREA(5)
* Abs(B5)) + (BLOCKAREA(6) * Abs(B6)) + (BLOCKAREA(7) * Abs(B7)) +
(BLOCKAREA(8) * Abs(B8)) + (BLOCKAREA(9) * Abs(B9)) +
(BLOCKAREA(10) * Abs(B10))) + " AS Ex13")
End Sub

```

```

Private Sub SPACEAVAILABLE(XBEGIN, YBEGIN, XEND, YEND,
PLATENUMBER As Integer)
    AccessDB.Execute ("INSERT INTO SPACES (BEGINX, BEGINY, ENDX,
ENDY, AVAILABLE, SPACENO,PATTERNNO,PARENTSPACE)" & _
    "SELECT " & _
    CStr(XBEGIN) + " AS Ex1," & _
    CStr(YBEGIN) + " AS Ex2," & _
    CStr(XEND) + " AS Ex3," & _
    CStr(YEND) + " AS Ex4," & _
    "1 AS Ex5," & _
    CStr(PLATENUMBER) + " AS Ex6," & _
    CStr(PATTERNNO) + " AS Ex7," & _
    CStr(PARENTSPACENO) + " AS Ex8")
End Sub

Private Sub SPACEDECISION(BNO, BFORM, XBEGIN1, YBEGIN1, XEND1,
YEND1, XBEGIN2, YBEGIN2, XEND2, YEND2 As Integer)
Dim CountI1, B_NO, B_FORM, ForecastHORIZONTAL1, ForecastVERTICAL1,
ForecastHORIZONTAL2, ForecastVERTICAL2 As Integer
    CountI1 = 0
    B_NO = BNO
    B_FORM = BFORM
    ForecastHORIZONTAL1 = 0
    ForecastVERTICAL1 = 0
    ForecastHORIZONTAL2 = 0
    ForecastVERTICAL2 = 0

    Do While ((ForecastHORIZONTAL1 * ForecastVERTICAL1) = 0 And
(ForecastHORIZONTAL2 * ForecastVERTICAL2) = 0) And (B_NO <=
MAXBLOCK)
        If B_FORM = 0 Then
            ForecastHORIZONTAL1 = Fix((XEND1 - XBEGIN1) /
BLOCKLENGTH(B_NO))
            ForecastVERTICAL1 = Fix((YEND1 - YBEGIN1) /
BLOCKWIDTH(B_NO))
            ForecastHORIZONTAL2 = Fix((XEND2 - XBEGIN2) /
BLOCKLENGTH(B_NO))
            ForecastVERTICAL2 = Fix((YEND2 - YBEGIN2) /
BLOCKWIDTH(B_NO))
        End If
        If (ForecastHORIZONTAL1 * ForecastVERTICAL1) <= 0 And
(ForecastHORIZONTAL2 * ForecastVERTICAL2) <= 0 And B_FORM = 0 Then
            B_FORM = 1
        End If
        If B_FORM = 1 Then
            ForecastHORIZONTAL1 = Fix((XEND1 - XBEGIN1) /
BLOCKRLENGTH(B_NO))
            ForecastVERTICAL1 = Fix((YEND1 - YBEGIN1) /
BLOCKRWIDTH(B_NO))

```

```

ForecastHORIZONTAL2 = Fix((XEND2 - XBEGIN2) /
BLOCKRLENGTH(B_NO))
ForecastVERTICAL2 = Fix((YEND2 - YBEGIN2) /
BLOCKRWIDTH(B_NO))
End If
If (ForecastHORIZONTAL1 * ForecastVERTICAL1) <= 0 And
(ForecastHORIZONTAL2 * ForecastVERTICAL2) <= 0 And B_FORM = 1 Then
    B_FORM = 0
    B_NO = B_NO + 1
End If
Loop

If (ForecastHORIZONTAL1 * ForecastVERTICAL1) > (ForecastHORIZONTAL2
* ForecastVERTICAL2) Then
    If ((XEND1 - XBEGIN1) * (YEND1 - YBEGIN1)) > 0 Then
        PLATERUNNO = PLATERUNNO + 1
        TOPSPACE = PLATERUNNO
        Call SPACEAVAILABLE(XBEGIN1, YBEGIN1, XEND1, YEND1,
PLATERUNNO) 'TopSpace
    Else
        TOPSPACE = 0
    End If
    If ((XEND2 - XBEGIN2) * (YEND2 - YBEGIN2)) > 0 Then
        PLATERUNNO = PLATERUNNO + 1
        RIGHTSPACE = PLATERUNNO
        Call SPACEAVAILABLE(XBEGIN2, YBEGIN2, XEND2, YBEGIN1,
PLATERUNNO) 'RightSpace
    Else
        RIGHTSPACE = 0
    End If
    ElseIf (ForecastHORIZONTAL1 * ForecastVERTICAL1) <
(ForecastHORIZONTAL2 * ForecastVERTICAL2) Then
        If ((XEND1 - XBEGIN1) * (YEND1 - YBEGIN1)) > 0 Then
            PLATERUNNO = PLATERUNNO + 1
            TOPSPACE = PLATERUNNO
            Call SPACEAVAILABLE(XBEGIN2, YBEGIN2, XEND2, YEND2,
PLATERUNNO) 'TopSpace
        Else
            TOPSPACE = 0
        End If
        If ((XEND2 - XBEGIN2) * (YEND2 - YBEGIN2)) > 0 Then
            PLATERUNNO = PLATERUNNO + 1
            RIGHTSPACE = PLATERUNNO
            Call SPACEAVAILABLE(XBEGIN1, YBEGIN1, XBEGIN2, YEND1,
PLATERUNNO) 'RightSpace
        Else
            RIGHTSPACE = 0
        End If
        ElseIf (ForecastHORIZONTAL1 * ForecastVERTICAL1) =
(ForecastHORIZONTAL2 * ForecastVERTICAL2) Then

```

```

        If ((XEND1 - XBEGIN1) * (YEND1 - YBEGIN1)) >= ((XEND2 - XBEGIN2) *
(YEND2 - YBEGIN2)) Then
            If ((XEND1 - XBEGIN1) * (YEND1 - YBEGIN1)) > 0 Then
                PLATERUNNO = PLATERUNNO + 1
                TOPSPACE = PLATERUNNO
                Call SPACEAVAILABLE(XBEGIN1, YBEGIN1, XEND1, YEND1,
PLATERUNNO) 'TopSpace
            Else
                TOPSPACE = 0
            End If
            If ((XEND2 - XBEGIN2) * (YEND2 - YBEGIN2)) > 0 Then
                PLATERUNNO = PLATERUNNO + 1
                RIGHTSPACE = PLATERUNNO
                Call SPACEAVAILABLE(XBEGIN2, YBEGIN2, XEND2, YBEGIN1,
PLATERUNNO) 'RightSpace
            Else
                RIGHTSPACE = 0
            End If
        Else
            If ((XEND1 - XBEGIN1) * (YEND1 - YBEGIN1)) > 0 Then
                PLATERUNNO = PLATERUNNO + 1
                TOPSPACE = PLATERUNNO
                Call SPACEAVAILABLE(XBEGIN2, YBEGIN2, XEND2, YEND2,
PLATERUNNO) 'TopSpace
            Else
                TOPSPACE = 0
            End If
            If ((XEND2 - XBEGIN2) * (YEND2 - YBEGIN2)) > 0 Then
                PLATERUNNO = PLATERUNNO + 1
                RIGHTSPACE = PLATERUNNO
                Call SPACEAVAILABLE(XBEGIN1, YBEGIN1, XBEGIN2, YEND1,
PLATERUNNO) 'RightSpace
            Else
                RIGHTSPACE = 0
            End If
        End If
    End If
End Sub

```

```

Private Sub UPDATEBLOCKPOSITION2(BLOCKNO, BLOCKFORM,
REMAINNO, XBEGIN, YBEGIN, COLUMNNO, ROWNO, TOPSPACENO1,
RIGHTSPACENO1 As Integer)
Dim i, j, XBGIN, YBGIN, XN, YN, COUNTI As Integer
COUNTI = 1
    For j = 1 To COLUMNNO
        For i = 1 To ROWNO
            BLOCKORDER = BLOCKORDER + 1
            If BLOCKFORM = 0 Then
                XBGIN = XBEGIN + (BLOCKLENGTH(BLOCKNO) * (i - 1))
                YBGIN = YBEGIN + (BLOCKWIDTH(BLOCKNO) * (j - 1))

```

```

        XN = XBEGIN + (BLOCKLENGTH(BLOCKNO) * i)
        YN = YBEGIN + (BLOCKWIDTH(BLOCKNO) * j)
    Else
        XBGIN = XBEGIN + (BLOCKRLENGTH(BLOCKNO) * (i - 1))
        YBGIN = YBEGIN + (BLOCKRWIDTH(BLOCKNO) * (j - 1))
        XN = XBEGIN + (BLOCKRLENGTH(BLOCKNO) * i)
        YN = YBEGIN + (BLOCKRWIDTH(BLOCKNO) * j)
    End If

    If COUNTI <= REMAINNO Then
        AccessDB.Execute ("INSERT INTO BLOCKPOSITION
(PATTERNNO,BLOCKNO,BLOCKORDER,XBEGIN,YBEGIN,XEND,YEND,TOP
SPACE,RIGHTSPACE,SPACENO,LEVELS) " & _
        "SELECT " & _
        CStr(PATTERNNO - 1) + " AS Ex1," & _
        CStr(BLOCKNO) + " AS Ex2," & _
        CStr(BLOCKORDER) + " AS Ex3," & _
        CStr(XBGIN) + " AS Ex4," & _
        CStr(YBGIN) + " AS Ex5," & _
        CStr(XN) + " AS Ex6," & _
        CStr(YN) + " AS Ex7," & _
        CStr(TOPSPACENO1) + " AS Ex8," & _
        CStr(RIGHTSPACENO1) + " AS Ex9," & _
        CStr(SPACENO) + " AS Ex10," & _
        CStr(LEVELS) + " AS Ex11")
    End If
    COUNTI = COUNTI + 1
Next i
Next j
End Sub

Private Function FINDBIGGESTBLOCK(STARTBNO, STARTBFORM,
WIDTHAREA1, LENGTHAREA1, WIDTHAREA2, LENGTHAREA2,
WIDTHAREA3, LENGTHAREA3 As Integer) As Integer
Dim BBNO, BBFORM, COUNTF As Integer
    BBNO = STARTBNO
    BBFORM = STARTBFORM
    FINDBIGGESTBLOCK = 0

    For COUNTF = BBNO To MAXBLOCK
        If BBFORM = 0 Then
            If (WIDTHAREA1 >= BLOCKWIDTH(COUNTF) And LENGTHAREA1
>= BLOCKLENGTH(COUNTF)) Or (WIDTHAREA2 >=
BLOCKWIDTH(COUNTF) And LENGTHAREA2 >=
BLOCKLENGTH(COUNTF)) Then
                If FINDBIGGESTBLOCK = 0 Then
                    FINDBIGGESTBLOCK = COUNTF
                End If
            End If
        End If
        BBFORM = 1
    End If

```

```

    If BBFROM = 1 Then
        If (WIDTHAREA1 >= BLOCKRWIDTH(COUNTF) And LENGTHAREA1
        >= BLOCKRLENGTH(COUNTF)) Or (WIDTHAREA2 >=
        BLOCKRWIDTH(COUNTF) And LENGTHAREA2 >=
        BLOCKRLENGTH(COUNTF)) Then
            If FINDBIGGESTBLOCK = 0 Then
                FINDBIGGESTBLOCK = 1000 + COUNTF
            End If
        End If
        BBFORM = 0
    End If
Next COUNTF

```

End Function

```

Private Sub SPACEDECISION2(BREMAIN, BNO, BFORM, REMAINNO,
VERTICALNO, HORIZONTALNO As Integer)
Dim CountI1, CountI2, CountI3, CountI4, CountI5, B_NO, B_FORM,
PossiblePlateXBEGIN(5, 3), PossiblePlateYBEGIN(5, 3), PossiblePlateXEND(5, 3),
PossiblePlateYEND(5, 3) As Integer
Dim LengthB, WidthB, ForecastHORIZONTAL1, ForecastVERTICAL1,
ForecastHORIZONTAL2, ForecastVERTICAL2 As Integer
    CountI1 = 0
    CountI2 = 0
    CountI3 = 0
    CountI4 = 0
    B_NO = BNO
    B_FORM = BFORM
    For CountI1 = 1 To 5
        For CountI2 = 1 To 3
            PossiblePlateXBEGIN(CountI1, CountI2) = 0
            PossiblePlateYBEGIN(CountI1, CountI2) = 0
            PossiblePlateXEND(CountI1, CountI2) = 0
            PossiblePlateYEND(CountI1, CountI2) = 0
        Next CountI2
    Next CountI1
    CountI1 = 0

    If B_FORM = 0 Then
        LengthB = BLOCKLENGTH(B_NO)
        WidthB = BLOCKWIDTH(B_NO)
    Else
        LengthB = BLOCKRLENGTH(B_NO)
        WidthB = BLOCKRWIDTH(B_NO)
    End If
    If REMAINNO < HORIZONTALNO Then
        PossiblePlateXBEGIN(1, 1) = PLATEXBEGIN(1)
        PossiblePlateYBEGIN(1, 1) = PLATEYBEGIN(1) + WidthB
        PossiblePlateXEND(1, 1) = PLATEXBEGIN(1) + (REMAINNO * LengthB)
    End If

```

PossiblePlateYEND(1, 1) = PLATEYEND(1)

PossiblePlateXBEGIN(1, 2) = PLATEXBEGIN(1) + (REMAINNO * LengthB)

PossiblePlateYBEGIN(1, 2) = PLATEYBEGIN(1)

PossiblePlateXEND(1, 2) = PLATEXEND(1)

PossiblePlateYEND(1, 2) = PLATEYEND(1)

'----- ÃÛ»á°;ÏÃµÑ'·Õèà»ç¹ä»ä´é á°·Õè2

PossiblePlateXBEGIN(2, 1) = PLATEXBEGIN(1) + (REMAINNO * LengthB)

PossiblePlateYBEGIN(2, 1) = PLATEYBEGIN(1)

PossiblePlateXEND(2, 1) = PLATEXEND(1)

PossiblePlateYEND(2, 1) = PLATEYBEGIN(1) + WidthB

PossiblePlateXBEGIN(2, 2) = PLATEXBEGIN(1)

PossiblePlateYBEGIN(2, 2) = PLATEYBEGIN(1) + WidthB

PossiblePlateXEND(2, 2) = PLATEXEND(1)

PossiblePlateYEND(2, 2) = PLATEYEND(1)

CountI3 = 1

CountI4 = 1

For CountI1 = 1 To 5

For CountI2 = 1 To 3

CountI5 = FINDBIGGESTBLOCK(B_NO, B_FORM,
(PossiblePlateYEND(CountI3, CountI4) - PossiblePlateYBEGIN(CountI3, CountI4)),
(PossiblePlateXEND(CountI3, CountI4) - PossiblePlateXBEGIN(CountI3, CountI4)),
(PossiblePlateYEND(CountI1, CountI2) - PossiblePlateYBEGIN(CountI1, CountI2)),
(PossiblePlateXEND(CountI1, CountI2) - PossiblePlateXBEGIN(CountI1,
CountI2)),0,0)

If CountI5 > 0 Then

If CountI5 < 1000 Then

LengthB = BLOCKLENGTH(CountI5)

WidthB = BLOCKWIDTH(CountI5)

Else

LengthB = BLOCKRLENGTH(CountI5 - 1000)

WidthB = BLOCKRWIDTH(CountI5 - 1000)

End If

End If

ForecastHORIZONTAL1 = Fix((PossiblePlateXEND(CountI3, CountI4) -
PossiblePlateXBEGIN(CountI3, CountI4)) / LengthB)

ForecastVERTICAL1 = Fix((PossiblePlateYEND(CountI3, CountI4) -
PossiblePlateYBEGIN(CountI3, CountI4)) / WidthB)

ForecastHORIZONTAL2 = Fix((PossiblePlateXEND(CountI1, CountI2) -
PossiblePlateXBEGIN(CountI1, CountI2)) / LengthB)

ForecastVERTICAL2 = Fix((PossiblePlateYEND(CountI1, CountI2) -
PossiblePlateYBEGIN(CountI1, CountI2)) / WidthB)

```

        If (ForecastHORIZONTAL2 * ForecastVERTICAL2) >=
(ForecastHORIZONTAL1 * ForecastVERTICAL1) Then
            If (ForecastHORIZONTAL2 * ForecastVERTICAL2) =
(ForecastHORIZONTAL1 * ForecastVERTICAL1) Then
                If ((PossiblePlateXEND(CountI1, CountI2) -
PossiblePlateXBEGIN(CountI1, CountI2)) * (PossiblePlateYEND(CountI1, CountI2) -
PossiblePlateYBEGIN(CountI1, CountI2))) > ((PossiblePlateXEND(CountI3, CountI4)
- PossiblePlateXBEGIN(CountI3, CountI4)) * (PossiblePlateYEND(CountI3, CountI4)
- PossiblePlateYBEGIN(CountI3, CountI4))) Then
                    CountI3 = CountI1
                    CountI4 = CountI2
                End If
            Else
                CountI3 = CountI1
                CountI4 = CountI2
            End If
        End If
    Next CountI2
Next CountI1
PLATERUNNO = PLATERUNNO + 1
Call SPACEAVAILABLE(PossiblePlateXBEGIN(CountI3, 1),
PossiblePlateYBEGIN(CountI3, 1), PossiblePlateXEND(CountI3, 1),
PossiblePlateYEND(CountI3, 1), PLATERUNNO)
PLATERUNNO = PLATERUNNO + 1
Call SPACEAVAILABLE(PossiblePlateXBEGIN(CountI3, 2),
PossiblePlateYBEGIN(CountI3, 2), PossiblePlateXEND(CountI3, 2),
PossiblePlateYEND(CountI3, 2), PLATERUNNO)

ElseIf (REMAINNO Mod HORIZONTALNO) = 0 Then
    PossiblePlateXBEGIN(1, 1) = PLATEXBEGIN(1)
    PossiblePlateYBEGIN(1, 1) = PLATEYBEGIN(1) + (Fix(REMAINNO /
HORIZONTALNO) * WidthB)
    PossiblePlateXEND(1, 1) = PLATEXBEGIN(1) + (HORIZONTALNO *
LengthB)
    PossiblePlateYEND(1, 1) = PLATEYEND(1)

    PossiblePlateXBEGIN(1, 2) = PLATEXBEGIN(1) + (HORIZONTALNO *
LengthB)
    PossiblePlateYBEGIN(1, 2) = PLATEYBEGIN(1)
    PossiblePlateXEND(1, 2) = PLATEXEND(1)
    PossiblePlateYEND(1, 2) = PLATEYEND(1)

    PossiblePlateXBEGIN(2, 1) = PLATEXBEGIN(1) + (HORIZONTALNO *
LengthB)
    PossiblePlateYBEGIN(2, 1) = PLATEYBEGIN(1)
    PossiblePlateXEND(2, 1) = PLATEXEND(1)
    PossiblePlateYEND(2, 1) = PLATEYBEGIN(1) + (Fix(REMAINNO /
HORIZONTALNO) * WidthB)

    PossiblePlateXBEGIN(2, 2) = PLATEXBEGIN(1)

```

```

PossiblePlateYBEGIN(2, 2) = PLATEYBEGIN(1) + (Fix(REMAINNO /
HORIZONTALNO) * WidthB)
PossiblePlateXEND(2, 2) = PLATEXEND(1)
PossiblePlateYEND(2, 2) = PLATEYEND(1)

CountI3 = 1
CountI4 = 1

For CountI1 = 1 To 5
    For CountI2 = 1 To 3

        CountI5 = FINDBIGGESTBLOCK(B_NO, B_FORM,
(PossiblePlateYEND(CountI3, CountI4) - PossiblePlateYBEGIN(CountI3, CountI4)),
(PossiblePlateXEND(CountI3, CountI4) - PossiblePlateXBEGIN(CountI3, CountI4)),
(PossiblePlateYEND(CountI1, CountI2) - PossiblePlateYBEGIN(CountI1, CountI2)),
(PossiblePlateXEND(CountI1, CountI2) - PossiblePlateXBEGIN(CountI1, CountI2)), 0,
0)

        If CountI5 > 0 Then
            If CountI5 < 1000 Then
                LengthB = BLOCKLENGTH(CountI5)
                WidthB = BLOCKWIDTH(CountI5)
            Else
                LengthB = BLOCKRLENGTH(CountI5 - 1000)
                WidthB = BLOCKRWIDTH(CountI5 - 1000)
            End If
        End If

        ForecastHORIZONTAL1 = Fix((PossiblePlateXEND(CountI3, CountI4) -
PossiblePlateXBEGIN(CountI3, CountI4)) / LengthB)
        ForecastVERTICAL1 = Fix((PossiblePlateYEND(CountI3, CountI4) -
PossiblePlateYBEGIN(CountI3, CountI4)) / WidthB)
        ForecastHORIZONTAL2 = Fix((PossiblePlateXEND(CountI1, CountI2) -
PossiblePlateXBEGIN(CountI1, CountI2)) / LengthB)
        ForecastVERTICAL2 = Fix((PossiblePlateYEND(CountI1, CountI2) -
PossiblePlateYBEGIN(CountI1, CountI2)) / WidthB)

        If (ForecastHORIZONTAL2 * ForecastVERTICAL2) >=
(ForecastHORIZONTAL1 * ForecastVERTICAL1) Then
            If (ForecastHORIZONTAL2 * ForecastVERTICAL2) =
(ForecastHORIZONTAL1 * ForecastVERTICAL1) Then
                If ((PossiblePlateXEND(CountI1, CountI2) -
PossiblePlateXBEGIN(CountI1, CountI2)) * (PossiblePlateYEND(CountI1, CountI2) -
PossiblePlateYBEGIN(CountI1, CountI2))) > ((PossiblePlateXEND(CountI3, CountI4) -
PossiblePlateXBEGIN(CountI3, CountI4)) * (PossiblePlateYEND(CountI3, CountI4) -
PossiblePlateYBEGIN(CountI3, CountI4))) Then
                    CountI3 = CountI1
                    CountI4 = CountI2
                End If
            Else
                CountI3 = CountI1
            End If
        End If
    End For
End For

```