รายงานการวิจัย

การพัฒนาโปรแกรมจัดเรียงชิ้นงานสำหรับงานแม่พิมพ์โลหะ

Software Development for Die Nesting

สุจริต สิงหพันธุ์

จตุพร ใจดำรงค์

สุรสิทธิ์ ระวังวงศ์

ว.
033
ส749
2549

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

พ.ศ. 2549

ได้รับทุนอุดหนุนจากงบผลประโยชน์ วิทยาเขตภาคใต้ ประจำปีงบประมาณ 2548

บทคัดย่อ

ในการออกแบบการจัดเรียงชิ้นงานแม่พิมพ์ ผู้ออกแบบจะต้องคำนึงถึงวิธีการจัดเรียงชิ้นงานเพื่อให้ประหยัดวัสดุให้มากที่สุด รูปทรงที่ซับซ้อนของชิ้นงานทำให้การออกแบบโดยใช้ความชำนาญของคนต้องใช้เวลานาน และบางครั้งอาจไม่ใช่คำตอบที่ดีที่สุดเนื่องจากไม่มีเวลาเพียงพอที่จะหาคำตอบได้ อาจจะทำให้ไม่ได้ตามแผนที่คาดการณ์ไว้

การทำวิจัยในครั้งนี้ เป็นการออกแบบอัลกอริธึมและสร้างฟังก์ชันเพื่อช่วยในการจัดรียงชิ้นงานแม่พิมพ์แบบแถวเดียวป้อนหนึ่งทางและแบบแถวเดียวป้อนสองทาง โดยอาศัยการทำงานในโปรแกรมออโต้แคทและใช้ออโต้ลิสในการสร้างฟังก์ชัน ด้วยการให้โปรแกรมอ่านข้อมูลของเส้นเข้ามาในหน่วยความจำแล้วนำมาคำนวณให้หมุนรูปทรงไปครั้งละ 1 องศา ในขณะที่คำนวณค่าพื้นที่แบลงค์เก็บไว้จนครบ 1 รอบ ค่ามุมที่ให้พื้นที่แบลงค์น้อยสุดคือตำแหน่งที่ให้ประสิทธิภาพสูงสุด

จากการทดลองเปรียบเทียบกับการใช้ความชำนาญของคนในการจัดเรียง สรุปได้ว่า การใช้โปรแกรมช่วยในการจัดเรียงชิ้นงานสามารถให้คำตอบที่ถูกต้องได้มากกว่าและใช้เวลาในการทำงานน้อยกว่า การนำฟังก์ชันนี้ไปใช้งานจะช่วยให้สามารถตัดสินใจเลือกรูปแบบในการวางชิ้นงานได้ดีขึ้น

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

กิตติกรรมประกาศ

ในการจัดทำวิจัยในครั้งนี้ได้รับทุนอุดหนุนจากงบผลประโยชน์ วิทยาเขตภาคใต้ ประจำปีงบประมาณ พ.ศ.2548 จากสถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคใต้

สารบัญ

เรื่อง	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญภาพ	III
สารบัญตาราง	V
บทที่ 1 บทนำ	
ความสำคัญและที่มาของปัญหา	1-1
วัตถุประสงค์	1-2
ขอบเขตของโครงการวิจัย	1-2
ข้อสมมตติฐาน	1-2
ประโยชน์ของผลการวิจัย	1-3
หน่วยงานที่นำผลการวิจัยไปใช้ประโยชน์	1-3
บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง	
การจัดเรียงชิ้นงานแบบแถวเดียวป้อนหนึ่งทาง	2-1
การจัดเรียงชิ้นงานแบบแถวเดียวป้อนสองทาง	2-2
การจัดเรียงชิ้นงานแบบหลายแถว	2-2
ประสิทธิภาพของการจัดเรียงชิ้นงาน	2-3
บทที่ 3 วิธีดำเนินการศึกษา	
การจัดทำข้อกำหนดของการจัดเรียงชิ้นงาน	3-2
การออกแบบขั้นตอนการทำงานของโปรแกรม	3-3

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการทดลองและสรุป	
ผลการเปรียบเทียบการจัดเรียงชิ้นงานแม่พิมพ์	4-2
สรุป	4-22
ข้อเสนอแนะทั่วไป	4-22
ข้อเสนอแนะในการวิจัยครั้งต่อไป	4-22
บรรณานุกรม	

สารบัญภาพ

ภาพที่		หน้า
2-1	วางชิ้นงานแบบแถวเดียวป้อนผ่านหนึ่งครั้ง	2-1
2-2	วางชิ้นงานแบบแถวเดียวป้อนผ่านสองครั้ง	2-2
2-3	การวางชิ้นงานที่มุม 90 องศา	2-2
2-4	ตัวอย่างของการหา Blank Area กรณีการป้อนผ่านหนึ่งครั้ง	2-3
2-5	ตัวอย่างของการหา Blank Area กรณีการป้อนผ่านสองครั้ง	2-3
2-6	การเขียนเส้นรอบรูปชิ้นงานบนกระดาษโปร่งแสง และนำไปเลื่อนหาตำแหน่งที่เหมาะสม	2-4
2-7	กำหนดขนาดแผ่น Strip ทิศทางการป้อน ระยะป้อน และคำนวณหาประสิทธิภาพการจัดเรียงชิ้นงาน	2-4
2-8	การใช้สี่เหลี่ยมล้อมรอบชิ้นงานที่ประกบกันในการเลื่อนประกบกันที่ระดับต่าง ๆ	2-6
2-9	การจัดเรียงชิ้นงานที่ได้จากการประกบชิ้นงาน	2-6
2-10	ตัวอย่างเส้นรอบรูปชิ้นงานทีต้องการหาจุดศูนย์ถ่วง	2-7
2-11	ตัวอย่างการหาจุดศูนย์ถ่วง โดยการคำนวณโมเมนต์ของเส้นรอบรูป	2-8
3-1	ระยะเผื่อขอบทางด้านข้าง	3-2
3-2	ระยะเผื่อขอบระหว่างชิ้นงาน	3-2
3-3	มุมหมุนของชิ้นงาน	3-3
3-4	จุดศูนย์ถ่วงของเส้นที่ได้	3-4
3-5	ข้อมูลของเส้นทั้งสองด้าน	3-5
3-6	การหา Blank Area ที่ตำแหน่งการวางชิ้นงานในมุมต่าง	3-5
3-7	การแสดงผลที่ Text Screen	3-6
3-8	ตัวอย่างการจัดเรียงชิ้นงานที่ได้	3-6
3-9	คำนวณได้จุดศูนย์ถ่วงของเส้น	3-8
3-10	การอ่านพิกัดของเส้นทีละด้าน	3-8
3-11	Blank Area ที่ตำแหน่งการวางชิ้นงานในตำแหน่งเริ่มต้น	3-9

สารบัญภาพ

ภาพที่		หน้า
3-12	Blank Area ที่ตำแหน่งการวางชิ้นงานในมุมที่หมุนไป	3-9
3-13	การแสดงผลที่ Text Screen	3-9
3-14	ตัวอย่างการจัดเรียงชิ้นงานแบบป้อนสองทางที่ได้จากการคำนวณ	3-10
4-1	ผลการจัดเรียงชิ้นงานที่ 1	4-2
4-2	ผลการจัดเรียงชิ้นงานที่ 2	4-3
4-3	ผลการจัดเรียงชิ้นงานที่ 3	4-4
4-4	ผลการจัดเรียงชิ้นงานที่ 4	4-5
4-5	ผลการจัดเรียงชิ้นงานที่ 5	4-6
4-6	ผลการจัดเรียงชิ้นงานที่ 6	4-7
4-7	ผลการจัดเรียงชิ้นงานที่ 7	4-8
4-8	ผลการจัดเรียงชิ้นงานที่ 8	4-9
4-9	ผลการจัดเรียงชิ้นงานที่ 9	4-10
4-10	ผลการจัดเรียงชิ้นงานที่ 10	4-11
4-11	ผลการจัดเรียงชิ้นงานที่ 11	4-12
4-12	ผลการจัดเรียงชิ้นงานที่ 12	4-13
4-13	ผลการจัดเรียงชิ้นงานที่ 13	4-14
4-14	ผลการจัดเรียงชิ้นงานที่ 14	4-15
4-15	ผลการจัดเรียงชิ้นงานที่ 15	4-16
4-16	ผลการจัดเรียงชิ้นงานที่ 16	4-17
4-17	ผลการจัดเรียงชิ้นงานที่ 17	4-18
4-18	ผลการจัดเรียงชิ้นงานที่ 18	4-19
4-19	ผลการจัดเรียงชิ้นงานที่ 19	4-20
4-20	ผลการจัดเรียงชิ้นงานที่ 20	4-21

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

สารบัญตาราง

ตารางที่		หน้า
1	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 1	4-2
2	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 2	4-3
3	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 3	4-4
4	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 4	4-5
5	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 5	4-6
6	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 6	4-7
7	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 7	4-8
8	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 8	4-9
9	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 9	4-10
10	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 10	4-11
11	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 11	4-12
12	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 12	4-13
13	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 13	4-14
14	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 14	4-15
15	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 15	4-16
16	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 16	4-17
17	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 17	4-18
18	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 18	4-19
19	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 19	4-20
20	ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 20	4-21

บทที่ 1
บทนำ

ความสำคัญและที่มาของปัญหา

ในปัจจุบันเทคโนโลยีทางด้านวิชาการในการออกแบบแม่พิมพ์ได้เข้ามามีบทบาท ในการเพิ่มผลผลิตทางด้านอุตสาหกรรม โดยเฉพาะอุตสาหกรรมการผลิตและขึ้นรูปชิ้นส่วนต่าง ๆ ไม่ว่าจะเป็นอุตสาหกรรมทางด้านการผลิตยวดยานพาหนะ หรืออุตสาหกรรมด้านอื่น ๆ ก็ตาม แม่พิมพ์ขึ้นรูปชิ้นส่วนมีความจำเป็นอย่างมากต่อการผลิตชิ้นส่วน เพิ่มเพิ่มปริมาณและคุณภาพของผลผลิตให้สูงขึ้น และดีกว่าเดิม

การผลิตชิ้นงานโดยผ่านกระบวนการแม่พิมพ์ตัดขึ้นรูปจะมีค่าใช้จ่ายในด้านของวัตถุดิบ ซึ่งในที่นี้ก็คือ แผ่นวัตถุดิบที่ใช้สำหรับจัดเรียงชิ้นงานนั่นเอง อันเนื่องมาจากการออกแบบแม่พิมพ์ในขั้นตอนของการจัดเรียงชิ้นงานลงบนแผ่นวัตถุดิบ เพราะชิ้นงานมีความหลากหลายทางด้านรูปร่างและขนาด ซึ่งมีผลทำให้การจัดเรียงชิ้นงานมีความยากลำบากทำให้สิ้นเปลืองวัตถุดิบมาก หรือเหลือเศษจากกการจัดเรียงมากนั่นเอง และวิธีการหนึ่งในการลดค่าใช้จ่ายในด้านนี้ที่สามารถทำได้คือการปรับปรุง หรือเปลี่ยนแปลงวิธีการจัดเรียงชิ้นงานในรูปแบบต่าง ๆ ที่ทำให้ได้จำนวนชิ้นงานในการจัดเรียงต่อหน่วยวัตถุดิบมีจำนวนมากขึ้น หรือเหลือเศษที่เกิดจากการจัดเรียงชิ้นงานน้อยที่สุด หรืออาจจะกล่าวได้ว่ามีประสิทธิภาพในการจัดเรียงชิ้นงานสูงสุด นั่นเอง

การเรียนการสอนรายวิชาที่เกี่ยวข้องกับการออกแบบแม่พิมพ์โลหะนั้นผู้สอนต้องอธิบายถึงกระบวนการในการออกแบบแม่พิมพ์ ซึ่งขั้นตอนแรก คือการจัดเรียงชิ้นงานลงบนแผ่นวัตถุดิบ และในขั้นตอนถัดไปจะเป็นการแออกแบบส่วนประกอบอื่นๆ ของชุดแม่พิมพ์ต่อไป และในขั้นตอนแรกนี้เองที่นักศึกษาจะต้องทำการศึกษาในหลักการจัดเรียงชิ้นงาน และสมควรที่จะได้ทำการฝึกในภาคปฏิบัติในการทดลองจัดเรียงชิ้นงาน โดยการใช้โปรแกรมคอมพิวเตอร์ เพื่อช่วยให้นักศึกษาเข้าใจในหลักการจัดเรียงชิ้นงานที่ได้เรียนมาอย่างดียิ่งขึ้น และ ช่วยให้นักศึกษาเข้าใจ และเห็นถึงความสำคัญของการนำคอมพิวเตอร์เข้ามาช่วยในการทำงาน

การจะจัดเรียงชิ้นงานให้ประหยัดแผ่นวัสดุได้นั้นผู้ออกแบบจะต้องนำแบบของชิ้นงานมาวิเคราะห์และทดลองวางชิ้นงานในมุมต่าง ๆ ทั้งแบบการป้อนหนึ่งทางและแบบการป้อนสองทาง

และเปรียบเทียบค่าเปอร์เซ็นของพื้นที่ใช้งานเพื่อเลือกใช้งานในค่าที่เหมาะสมที่สุดหรือให้ประหยัดวัสดุได้มากที่สุด ซึ่งในกรณีที่รูปทรงของชิ้นงานไม่ซับซ้อนมากนักผู้ออกแบบก็สามารถกำหนดมุมของการวางชิ้นงานได้ไม่ยากนัก แต่ในกรณีที่ชิ้นงานมีรูปทรงที่ซับซ้อนการหามุมที่เหมาะสมด้วยการเขียนแบบและทดลองด้วยมือจะทำได้ยาก ในปัจจุบันการใช้โปรแกรมคอมพิวเตอร์ช่วยในการเขียนแบบถือได้ว่าเป็นเรื่องจำเป็น ถ้าต้องการหามุมการจัดเรียงชิ้นงานเพื่อให้ประหยัดวัสดุโดยการใช้โปรแกรมช่วยในการเขียนแบบทั่ว ๆ ไป ก็อาจจะทำได้โดยการทดลองวางชิ้นงานในตำแหน่งต่าง ๆ แล้วคำนวณหาค่าประสิทธิภาพในตำแหน่งที่วางเพื่อนำค่าตำแหน่งที่ให้ประสิทธิภาพสูงสุดมาใช้งาน แต่วิธีการดังกล่าวก็ต้องใช้เวลาพอสมควร ทั้งนี้ก็ขึ้นอยู่กับรูปทรงของชิ้นงานเป็นสำคัญ แต่ถ้าต้องการหาซื้อโปรแกรมที่ใช้ในการจัดเรียงชิ้นงานมาใช้ก็มีจำหน่ายอยู่ทั่วไป ซึ่งส่วนใหญ่ก็ต้องนำเข้าจากต่างประเทศ

คณะผู้วิจัยเห็นว่ามีอีกแนวทางหนึ่งในการพึ่งพาตนเองก็คือการพัฒนาฟังก์ชันขึ้นมาใช้งานด้วยโปรแกรมช่วยในการเขียนแบบที่มีใช้กันอยู่ทั่วไป ในปัจจุบันโปรแกรมออโตแคด (AuotCAD) เป็นโปรแกรมหนึ่งที่ผู้ใช้งานสามารถเข้าถึงข้อมูลของเส้นต่าง ๆ และให้ผู้ใช้งานสามารถสร้างฟังก์ชันขึ้นมาทำงานร่วมกับฟังก์ชันที่มีอยู่เดิมได้ ด้วยภาษาออโตลิส (AutoLISP) จึงได้จัดทำโครงการนี้ขึ้นเพื่อจะเป็นประโยชน์ในการพัฒนาการเรียนการสอนทางด้านการจัดเรียงชิ้นงานแม่พิมพ์ ต่อไป

<u>วัตถุประสงค์</u>

1. การลดเวลาในการจัดเรียงชิ้นงานสำหรับงานแม่พิมพ์โลหะ
2. เพิ่มประสิทธิภาพการจัดเรียงชิ้นงานในด้านพื้นที่ที่เหลือของแผ่นวัตถุดิบ

<u>ขอบเขตของโครงการวิจัย</u>

1. จัดเรียงชิ้นงานสำหรับงานแม่พิมพ์โลหะในแบบหนึ่ง หรือสองทิศทาง
2. รูปแบบของชิ้นงานเป็นแบบไม่มีรูภายใน
3. โปรแกรมพัฒนาโดยใช้ภาษาออโต้ลิส ซึ่งพัฒนาบนโปรแกรมออโต้แคด

<u>ข้อสมมุติฐาน</u>

การจัดเรียงชิ้นงานที่ต้องการปั๊มตัดโดยกระบวนการทางแม่พิมพ์ ในขั้นตอนของการจัดเรียงชิ้นงานลงบนแผ่นวัตถุดิบเพื่อให้ได้จำนวนชิ้นงานที่มากที่สุด หรือเหลือเศษที่เกิดจากการจัดเรียงให้น้อยที่สุดนั้น ซึ่งแนวทางในการทำงาน ในขั้นแรกจะเป็นการเขียนรูปร่างชิ้นงานลงบนโปรแกรมออโต้แคช และสั่งทำงานโปรแกรมที่พัฒนาขึ้น ต่อจากนั้นจะกำหนดให้ผู้ใช้โปรแกรมเป็นผู้เลือกรูปชิ้นงานที่ต้องการนำมาทำการจัดเรียง และกำหนดค่าองค์ประกอบต่าง ๆ ของการจัดเรียง เช่น ขนาด

ของแผ่นวัตถุดิบ ระยะห่างของชิ้นงานที่นำมาทำการจัดเรียง ระยะห่างของชิ้นงานกับขอบของวัตถุดิบ และค่าอื่น ๆ ในขั้นตอนต่อไปโปรแกรมจะเริ่มทำงานโดยนำชิ้นงานสองชิ้นมาประกอบเข้าด้วยกัน ณ มุมหมุนหลาย ๆ มุม โดยโปรแกรมจะเลือกรูปแบบที่ดีที่สุดของการประกบกันของชิ้นงานที่มุมหมุนใด ๆ ซึ่งทำ ให้การจัดเรียงชิ้นงานลงบนแผ่นวัตถุดิบทั้งหมด มีจำนวนชิ้นงาน มากที่สุด หรือมีเศษที่เกิดจากการจัดเรียงให้น้อย ที่สุด

ประโยชน์ของผลการวิจัย

1. ได้โปรแกรมการจัดเรียงชิ้นงานสำหรับงานแม่พิมพ์โลหะ
2. ลดเวลาในการจัดเรียงชิ้นงานสำหรับงานแม่พิมพ์โลหะ
3. เพิ่มประสิทธิภาพในการจัดเรียงชิ้นงานในด้านพื้นที่ที่เหลือของแผ่นวัตถุดิบ
4. ใช้ประกอบการเรียนการสอนในรายวิชาแม่พิมพ์โลหะ สำหรับแผนกวิชาช่างกลโรงงาน
5. ได้แนวทางในการจัดเรียงชิ้นงาน ซึ่งสามารถนำไปประยุกต์ใช้ในอุตสาหกรรมด้านอื่น ๆ ได้

หน่วยงานที่นำผลการวิจัยไปใช้ประโยชน์

สถาบันการศึกษาที่มีการเรียนการสอนในด้านแม่พิมพ์โลหะ สามารถนำไปใช้สำหรับการประกอบการเรียนการสอน ในรายวิชาแม่พิมพ์โลหะเพื่อแสดงให้นักศึกษาเห็นถึงการจัดเรียงชิ้นงานอย่างมีประสิทธิภาพ เทคนิคในการจัดเรียงชิ้นงาน และการประยุกต์ใช้คอมพิวเตอร์ช่วยในการจัดเรียงชิ้นงาน

อุตสาหกรรมการผลิตด้านแม่พิมพ์โลหะ และด้านอุตสาหกรรม ตัดเย็บพื้นรองเท้าสามารถนำไปใช้ในการจัดเรียงชิ้นงานเพิ่มประสิทธิภาพในการใช้แผ่นวัตถุดิบอย่างมีประสิทธิภาพ

บทที่ 2
งานวิจัยและทฤษฎีที่เกี่ยวข้อง

ชาญชัยและคณะ [1] ได้กล่าวเกี่ยวกับความสำคัญเกี่ยวกับ Lay Out Scrap Strip ว่าเป็นงานอันดับแรกในการออกแบบแม่พิมพ์ ซึ่งจะมีผลโดยตรงต่อผลกำไรหรือความสูญเปล่าของวัสดุในงานปั๊มทุกชนิด ในบางงานอาจจะเหมาะกับวิธีการป้อนผ่านครั้งเดียว แต่ในบางงานอาจจะต้องใช้การป้อนผ่านสองครั้งจึงจะทำให้ประหยัดวัสดุได้มากขึ้น แต่การป้อนผ่านสองครั้งจะทำให้ค่าแรงงานเพิ่มขึ้นประมาณ 10-15 % ถ้าสามารถ Lay Out Scrap Strip ได้ประสิทธิภาพสูงมากขึ้นก็จะยิ่งไปชดเชยค่าใช้จ่ายส่วนอื่นที่เพิ่มขึ้นได้

การจัดตำแหน่งของชิ้นงาน (Blank) บนแผ่นสตริป (Strip) แบ่งได้ 3 ลักษณะ คือ การจัดเรียงชิ้นงานแบบแถวเดียวป้อนหนึ่งทาง (Single Row One Pass) การจัดเรียงชิ้นงานแบบแถวเดียวป้อนสองทาง (Single Row Two Pass) และการจัดเรียงชิ้นงานแบบหลายแถว (Multiple Row)

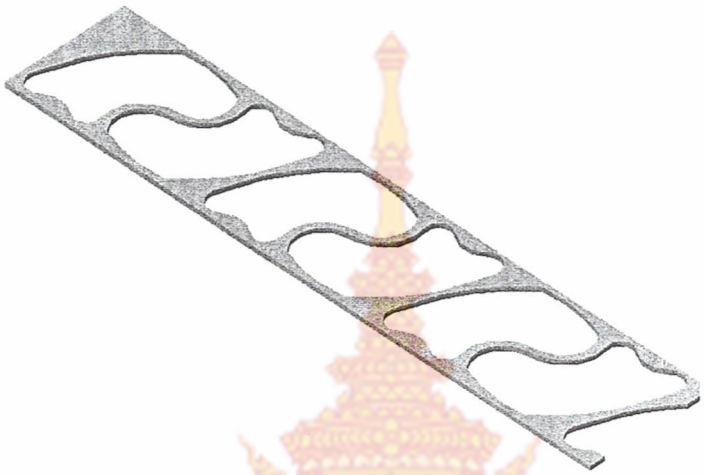
<u>การจัดเรียงชิ้นงานแบบแถวเดียวป้อนหนึ่งทาง</u>

เป็นการจัดเรียงชิ้นงาน โดยให้ด้านต่อด้านของชิ้นงานเรียงกันไป ในชิ้นงานที่มีรูปทรงซับซ้อนมักจะทำให้สิ้นเปลืองวัสดุ นิยมใช้กับงานที่เป็นรูปทรงอย่างง่าย เช่น รูปสี่เหลี่ยมผืนผ้า รูปตัวแอล (L) เป็นต้น

ภาพที่ 2-1 วางชิ้นงานแบบแถวเดียวป้อนผ่านหนึ่งครั้ง

การจัดเรียงชิ้นงานแบบแถวเดียวป้อนสองทาง

เป็นการจัดเรียงชิ้นงาน โดยให้ชิ้นงานถัดไปวางกลับด้านกับชิ้นเดิมโดยหมุนไป 180 องศา ในการผลิตจริงจะทำแม่พิมพ์เพียงชุดเดียว แต่ใช้วิธีการป้อนสลับข้างกัน มักจะเลือกใช้วิธีนี้เพื่อต้องการประหยัดวัสดุ

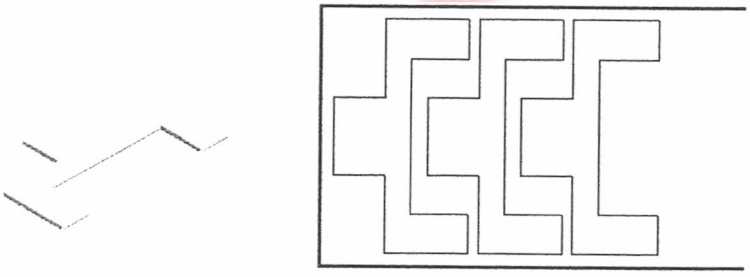


ภาพที่ 2- 2 วางชิ้นงานแบบแถวเดียวป้อนผ่านสองครั้ง

การจัดเรียงชิ้นงานแบบหลายแถว

เป็นการจัดเรียงชิ้นงาน โดยให้ชิ้นงานเรียงสลับกันเป็นมากกว่า 1 แถวขึ้นไป มักใช้กับการผลิตชิ้นงานจำนวนมาก เพื่อประหยัดเวลาการทำงาน และประหยัดวัสดุ

สมพงษ์ [2] เขียนวารสารเกี่ยวกับการออกแบบโพรเกรสซีฟดาย แสดงมาตรฐานการเลย์เอาท์แบบต่าง ๆ และแสดงให้เห็นว่าสำหรับชิ้นงานบางลักษณะการวางตำแหน่ง 90 องศา จะให้ประสิทธิภาพสูงสุด ดังในภาพที่ 2-3



ภาพที่ 2-3 การวางชิ้นงานที่มุม 90 องศา

ประสิทธิภาพของการจัดเรียงชิ้นงาน

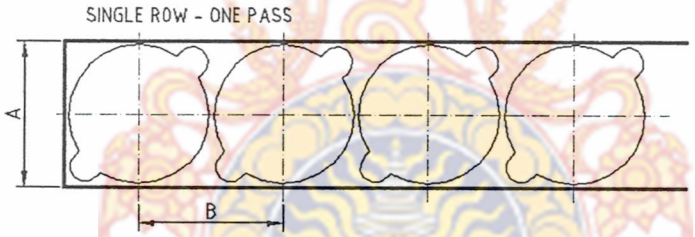
ประสิทธิภาพของการจัดเรียงชิ้นงาน (η) คือการคิดค่าของ พื้นที่ที่ได้เป็นชิ้นงานจริง ต่อพื้นที่ที่ต้องใช้ในการป้อนงาน 1 ครั้ง พื้นที่ที่ต้องใช้ในการป้อนงาน 1 ครั้งนี้เรียกว่า พื้นที่แบลงค์ (Blank Area) ฉะนั้นเราจึงมีสูตรในการคำนวณค่าต่าง ๆ ดังนี้

η = พื้นที่ของชิ้นงานที่ได้ / พื้นที่แบลงค์ ..... (2.1)

สำหรับในการวิจัยครั้งนี้ จะพิจารณาเฉพาะในการจัดเรียงชิ้นงานแบบแถวเดียว เพราะฉะนั้นในการหาพื้นที่แบลงค์จึงแบ่งพิจารณา ได้เป็น 2 กรณี คือ

กรณีที่ 1 การจัดเรียงชิ้นงานแบบแถวเดียวป้อนหนึ่งทาง ดังในภาพที่ 2-4

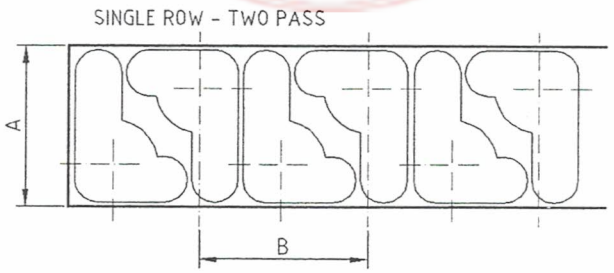
พื้นที่แบลงค์ = A x B ..... (2.2)



ภาพที่ 2-4 ตัวอย่างของการหาพื้นที่แบลงค์ กรณีการจัดเรียงชิ้นงานแบบแถวเดียวป้อนหนึ่งทาง

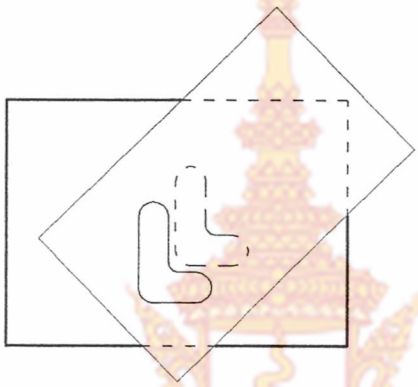
กรณีที่ 2 การจัดเรียงชิ้นงานแบบแถวเดียวป้อนสองทาง ดังในภาพที่ 2-5

พื้นที่แบลงค์ = A x B / 2 .....(2.3)

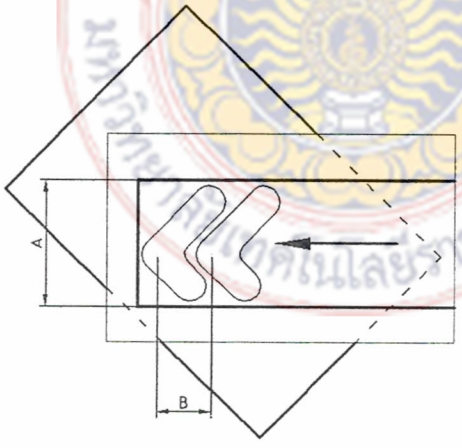


ภาพที่ 2-5 ตัวอย่างของการหาพื้นที่แบลงค์ กรณีการจัดเรียงชิ้นงานแบบแถวเดียวป้อนสองทาง

การหาวิธีการจัดเรียงชิ้นงาน เพื่อให้ได้ตำแหน่งที่ดีที่สุดนั้น ในกรณีของการหาโดยใช้คน ชาญชัยได้อธิบายวิธีการหาตำแหน่งการวางชิ้นงานโดยให้ใช้วิธีการเขียนเส้นรอบรูปชิ้นงานลงบนกระดาษโปร่งแสง แล้วจึงค่อย ๆ เลื่อนหาตำแหน่งที่เหมาะสมที่สุด ดังในภาพที่ 2-6 เมื่อได้ตำแหน่งที่เหมาะสม แล้วจึงไปกำหนดเป็นขนาดของแผ่น Strip ทิศทางการป้อน และระยะของการป้อน และนำไปสู่การคำนวณหาประสิทธิภาพการจัดเรียงชิ้นงานต่อไป ดังในภาพที่ 2-7



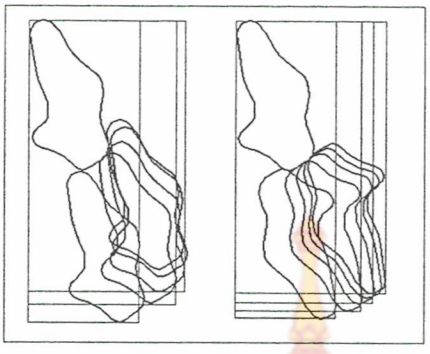
ภาพที่ 2-6 การเขียนเส้นรอบรูปชิ้นงานบนกระดาษโปร่งแสง และนำไปเลื่อนหาตำแหน่งที่เหมาะสม



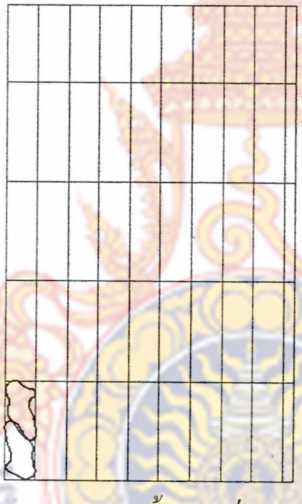
ภาพที่ 2-7 กำหนดขนาดแผ่น Strip ทิศทางการป้อน ระยะป้อน และคำนวณหาประสิทธิภาพการจัดเรียงชิ้นงาน

ในการแก้ปัญหาของการจัดเรียงชิ้นงานลงบนวัสดุแผ่นบางให้ประหยัดวัสดุ โดยไม่เกิดการซ้อนทับกัน ได้มีผู้คิดค้นวิธีการหาคำตอบและเสนอแนะไว้หลายวิธี ในกรณีที่เป็นชิ้นงานรูปทรงทั่วไป จะไม่สามารถหาตำแหน่งที่ดีที่สุดได้ในเวลาจำกัด การหาตำแหน่งที่ใกล้เคียงตำแหน่งที่ดีที่สุดจึงเป็นวิธีหนึ่งในการแก้ปัญหา Jacob [8] ได้ใช้วิธีการแทนชิ้นส่วนที่มีรูปร่างต่าง ๆ ซึ่งเรียกว่ารูปทรง โพลีกอน (Polygon) ด้วยรูปสี่เหลี่ยม และนำมาวางบนวัสดุให้อยู่ทางซ้ายล่างของแผ่นวัสดุ และใช้วิธีแบบเจนแนติกอัลกอริธึมเพื่อจัดลำดับของโพลีกอนที่ดีที่สุด สามารถจัดเรียงได้รวดเร็ว แต่ประสิทธิภาพของการใช้วัสดุยังมีค่าต่ำเกินไป Oliveira [9] ได้อธิบายวิธีการจัดเรียงโพลีกอนโดยใช้อัลกอริธึมแบบโนฟิตโพลีกอน (no-fit Polygon) ร่วมกับวิธีกรีดี (Greedy Method) และเรียกวิธีนี้ว่า ทูโพส (TOPOS) โดยการวางโพลีกอนลงในแผ่นวัสดุทีละชิ้นโดยเลือกตำแหน่งที่วางแล้วให้คำตอบที่ดีที่สุดจนครบทุกชิ้น ทดสอบวิธีการจัดเรียงที่แตกต่างกัน 126 แบบ และเลือกวิธีการจัดเรียงที่ดีที่สุด Daniel and Milenkovic [10] ได้ทำการวิจัยและสร้างโปรแกรมจัดเรียงชิ้นงานลงในแผ่นวัสดุโดยใช้หลักการฮิวสิสติค (Heuristic) ผลการวิจัยได้ถูกนำมาเปรียบเทียบกับโปรแกรมการจัดเรียงชิ้นงานที่มีใช้อยู่ก่อนแล้วและเปรียบเทียบกับการจัดเรียงด้วยความชำนาญของคน ผลปรากฏว่าโปรแกรมที่สร้างขึ้นบางทีก็จัดเรียงได้ประสิทธิภาพน้อยกว่าการใช้ความชำนาญของคนหรือโปรแกรมที่ใช้อยู่ก่อนแล้ว

ผดุงศิลป์ [3] ได้วิจัยและสร้างโปรแกรมเพื่อจัดเรียงชิ้นงานรูปทรงเดี่ยวลงบนแผ่นวัสดุโดยใช้หลักการ โกลบอลเซอช (Global Search) คือพิจารณาจากชิ้นงานที่หมุนไปทีละองศาจนครบ 360 องศา ในแต่ละองศาที่หมุนไปจะสร้างกรอบสี่เหลี่ยมครอบภาพชิ้นงาน และนำค่ามุมองศาที่ให้พื้นที่สี่เหลี่ยมน้อยสุดใช้เป็นมุมเริ่มต้นการวางชิ้นงาน จากนั้นทำการคัดลอกรูปทรงชิ้นงานมาหาตำแหน่งเพื่อประกอบกันด้วยค่ามุมองศาเดิมและค่ามุมองศาที่หมุนไป 180 องศา เพื่อหาประสิทธิภาพการจัดเรียงที่มากสุด โดยแบ่งการทำงานเป็น 2 อัลกอริธึม(Algorithm) โดยอัลกอริธึมที่ 1 ให้หาพื้นที่สี่เหลี่ยมขนาดเล็กสุด อัลกอริธึมที่ 2 หาจำนวนมากสุดของรูปสี่เหลี่ยมที่ครอบรูปทรงชิ้นงานแล้วนำไปบรรจุลงในแผ่นวัสดุ ดังในภาพที่ 2-8



ภาพที่ 2-8 การใช้สี่เหลี่ยมล้อมรอบชิ้นงานที่ประกบกันในการเลื่อนประกบกันที่ระดับต่าง ๆ



ภาพที่ 2-9 การจัดเรียงชิ้นงานที่ได้จากการประกบชิ้นงาน

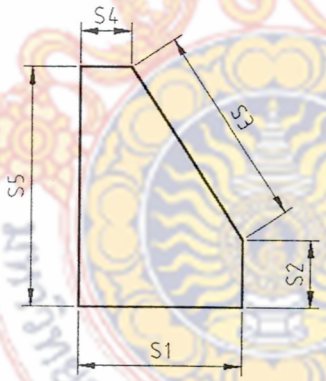
เมื่อนำโปรแกรมไปใช้งานเปรียบเทียบกับโปรแกรมที่มีการใช้งานอยู่ก่อนแล้ว และเปรียบเทียบกับการจัดเรียงโดยใช้ความชำนาญของคน ผลปรากฏว่า โปรแกรมที่สร้างขึ้นบางครั้งก็ให้ประสิทธิภาพที่มากกว่า บางครั้งให้ประสิทธิภาพเท่ากัน และบางครั้งให้ประสิทธิภาพน้อยกว่า

การหาตำแหน่งการวางชิ้นงานโดยการหมุนชิ้นงาน ไปครั้งละเท่ากับค่าความละเอียดของการหมุนเพื่อที่จะหาตำแหน่งการวางที่ดีที่สุดนั้น เป็นวิธีการที่ใช้กันโดยทั่วไป ส่วนการประมวลผลที่จะหาว่ามุมใดที่ให้ประสิทธิภาพมากสุดนั้น ในแต่ละอัลกอริธึมจะมีวิธีการหาที่แตกต่างกันออกไป

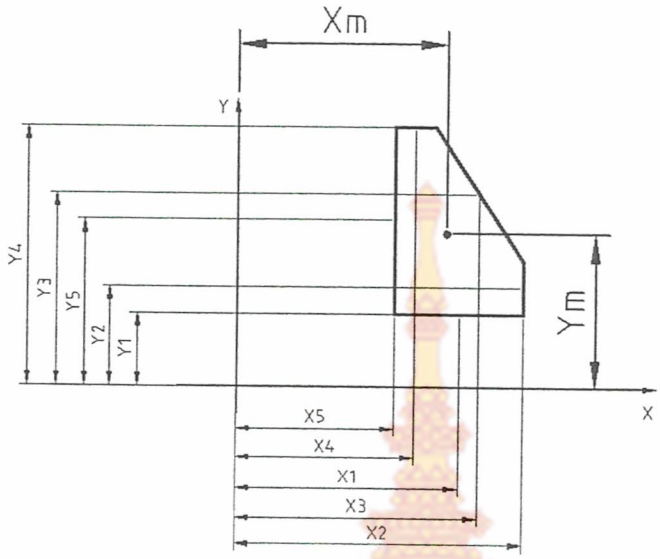
ในส่วนของการหมุนชิ้นงานนั้น จุดที่เป็นศูนย์กลางของการหมุนหรือที่เรียกว่า Base point นั้น ควรจะเป็นจุดที่เมื่อหมุนชิ้นงานไปแล้วไม่ทำให้ตำแหน่งของเส้นแตกต่างไปจากเดิมมากนัก จุดที่ให้คุณสมบัตินี้จุดหนึ่งก็คือ จุดศูนย์ถ่วง (Centroid) ของรูปทรง

บรรเลง [4] ได้แสดงตำแหน่งจุดศูนย์ถ่วงของเส้นตรงและส่วนโค้งของวงกลม (Arc)ซึ่งเป็นเส้นพื้นฐานของเส้นรอบรูปชิ้นงานทุกประเภท ชาญ และคณะ [5] ได้อธิบายวิธี การหาตำแหน่งจุดศูนย์ถ่วงของรูปปิด โดยการพิจารณาแบ่งเป็นรูปทรงเรขาคณิตอย่างง่าย หรือใช้การพิจารณาเส้นรอบรูปชิ้นงานทุกเส้น แล้วใช้หลักการโมเมนต์ในการคำนวณหาตำแหน่งจุดศูนย์ถ่วงทีละแกน ทำให้สามารถคำนวณหาตำแหน่งจุดศูนย์ถ่วงของเส้นรอบรูปชิ้นงาน 2 มิติ ได้ทุกประเภท และจะได้นำมาประยุกต์ใช้ในการคำนวณของฟังก์ชันที่สร้างขึ้นในการวิจัยในครั้งนี้ต่อไป

จากภาพที่ 2-10 คือ ตัวอย่างเส้นรอบรูปชิ้นงาน ซึ่งจะใช้เป็นตัวอย่างในการหาจุดศูนย์ถ่วงของเส้นทั้งหมด



ภาพที่ 2-10 ตัวอย่างเส้นรอบรูปชิ้นงานที่ต้องการหาจุดศูนย์ถ่วง



ภาพที่ 2-11 ตัวอย่างการหาจุดศูนย์ถ่วง โดยการคำนวณโมเมนต์ของเส้นรอบรูป

วิธีการหาจุดศูนย์ถ่วงของเส้นรอบรูป

ถ้าให้ Xm คือระยะของจุดศูนย์ถ่วงไปยังจุดกำเนิดทางแกน X

Ym คือระยะของจุดศูนย์ถ่วงไปยังจุดกำเนิดทางแกน Y

$$\therefore \quad Xm = \frac{X_1S_1 + X_2S_2 + X_3S_3 + X_4S_4 + X_5S_5}{S_1 + S_2 + S_3 + S_4 + S_5} \qquad \text{.....(2.4)}$$

$$\therefore \quad Ym = \frac{Y_1S_1 + Y_2S_2 + Y_3S_3 + Y_4S_4 + Y_5S_5}{S_1 + S_2 + S_3 + S_4 + S_5} \qquad \text{.....(2.5)}$$

กอบเกียรติ [6] โอมูระ [7] ได้อธิบายเกี่ยวกับการใช้ภาษาออโต้ลิสซึ่งมาพร้อมกับโปรแกรมออโต้แคด ว่าผู้ใช้งานสามารถนำมาพัฒนาสร้างฟังก์ชันช่วยในการคำนวณเกี่ยวภาพกราฟฟิคต่าง ๆ แล้วนำไปทำงานร่วมกับส่วนต่าง ๆ ของออโต้แคดที่มีอยู่เดิมได้ ซึ่งผู้วิจัยได้ใช้ภาษาออโต้ลิส นี้ ในการทดลองอัลกอริธึมที่ออกแบบขึ้น

บทที่ 3
วิธีดำเนินการวิจัย

การวิจัยครั้งนี้ มีวัตถุประสงค์เพื่อพัฒนาคำสั่งช่วยในการคำนวณหามุมที่เหมาะสมของการวางชิ้นงานในการออกแบบแม่พิมพ์โลหะ โดยได้เลือกใช้ภาษาออโตลิสซึ่งพัฒนาบนออโต้แคท ซึ่งมีฟังก์ชันเดิมเพียงพอที่จะใช้ในการดำเนินการวิจัยในครั้งนี้ได้

คณะผู้วิจัยได้ออกแบบอัลกอริธึม และได้ทดลองพัฒนาฟังก์ชันเพื่อช่วยในการจัดเรียงชิ้นงานแบบแถวเดียวทั้งแบบการป้อนหนึ่งทางและแบบการป้อนสองทางเพื่อเป็นตัวเลือกให้ใช้งานได้ โดยได้กำหนดวิธีการดำเนินงานตามหัวข้อต่อไปนี้

3.1 การจัดทำข้อกำหนดของการจัดเรียงชิ้นงาน

3.1.1. ระยะเผื่อขอบ

3.1.2. ระยะเผื่อขอบระหว่างชิ้นงาน

3.1.3. มุมหมุนของชิ้นงาน

3.2 การออกแบบขั้นตอนการทำงานของโปรแกรม

3.2.1. การรับข้อมูลของโปรแกรม

3.2.2. การหาค่ามุมที่ให้ประสิทธิภาพสูงสุดในกรณีของการป้อนหนึ่งทาง

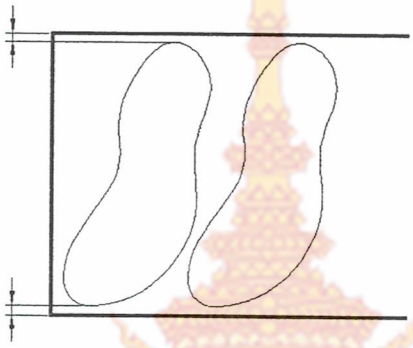
3.2.3. การหาค่ามุมที่ให้ประสิทธิภาพสูงสุดในกรณีของการป้อนสองทาง

3.2.4. การแสดงผล และตัวอย่างการจัดเรียงชิ้นงาน

3.1 การจัดทำข้อกำหนดของการจัดเรียงชิ้นงาน

3.1.1. ระยะเผื่อขอบ

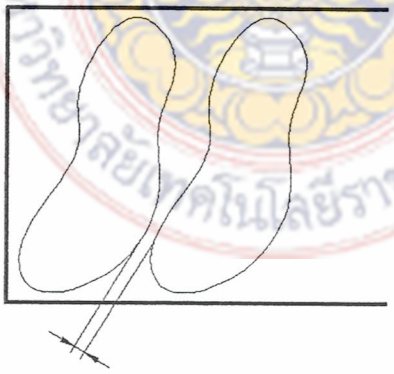
ในงานแม่พิมพ์โลหะแผ่นนั้น จะต้องมี ระยะเผื่อขอบจากขอบของชิ้นงานไปยังขอบของแผ่นสตริปที่จะนำมาปั๊มตัดชิ้นงาน ซึ่งผู้ออกแบบแม่พิมพ์จะต้องหาค่าที่เหมาะสำหรับแต่ละงานเอง ในการวิจัยนี้ผู้ใช้งานสามารถกำหนดค่าระยะเผื่อขอบได้ ดังในภาพที่ 3-1



ภาพที่ 3-1 ระยะเผื่อขอบทางด้านข้าง

3.1.2. ระยะเผื่อขอบระหว่างชิ้นงาน

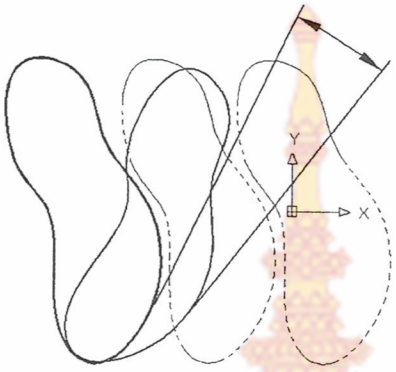
การจัดเรียงชิ้นงานลงบนแผ่นสตริป จำเป็นจะต้องมีระยะเผื่อขอบระหว่างชิ้นงาน ซึ่งฟังก์ชันที่ได้จากการวิจัยนี้ ให้ผู้ใช้งานสามารถกำหนดค่าระยะเผื่อขอบระหว่างชิ้นงานได้ ดังในภาพที่ 3-2



ภาพที่ 3-2 ระยะเผื่อขอบระหว่างชิ้นงาน

3.1.3. มุมหมุนของชิ้นงาน

การหาตำแหน่งการวางชิ้นงานเพื่อให้ประหยัดวัสดุมากสุดนั้น โปรแกรมจะทำการค้นหาโดยการหมุนชิ้นงานจากตำแหน่งตั้งต้นของชิ้นงานที่ผู้ใช้เขียนแบบลงไป เมื่อได้ค่าที่ทำให้ประหยัดวัสดุมากสุดแล้วจะบอกเป็นค่ามุมองศาวัดจากชิ้นงานเดิมดังในภาพที่ 3-3



ภาพที่ 3-3 มุมหมุนของชิ้นงาน

3.2 <u>การออกแบบขั้นตอนการทำงานของโปรแกรม</u>

3.2.1. การรับข้อมูลของโปรแกรม

ขั้นตอนในการรับข้อมูลของโปรแกรมเป็นดังนี้

1. ตรวจสอบคุณสมบัติของชิ้นงาน
2. เลือกชนิดของการป้อนหนึ่งทางหรือสองทาง
3. ระยะเผื่อขอบระหว่างชิ้นงาน
4. กำหนดค่าความละเอียดของการหมุนชิ้นงาน

3.2.2. การหาค่ามุมที่ให้ประสิทธิภาพสูงสุดในกรณีของการป้อนหนึ่งทาง

มีลำดับขั้นตอนในการทำงานดังนี้

1. อ่านข้อมูลของเส้นต่าง ๆ เข้ามาในหน่วยความจำ
2. คำนวณหาตำแหน่งจุดศูนย์ถ่วงของรูปทรง
3. หาตำแหน่งสูงสุดและต่ำสุดของรูปทรง
4. อ่านข้อมูลของเส้นจากตำแหน่งสูงสุดลงมาทั้งสองด้านเพื่อค้นหาค่าความกว้างมากสุดของรูปทรง
5. นำค่าที่ได้ไปคำนวณค่าประสิทธิภาพของการใช้พื้นที่

6. เปลี่ยนมุมวางชิ้นงานโดยหมุนไปเท่ากับค่าความละเอียดที่กำหนด แล้วเริ่มทำในขั้นตอนที่ 3 จนถึง ขั้นตอนที่ 5
7. เปลี่ยนมุมวางชิ้นงานจนกระทั่งครบ 360 องศา
8. นำค่ามุมที่ให้ประสิทธิภาพการจัดเรียงชิ้นงานมากสุด สุด แสดงผลทางหน้าจอเท็กซ์สกรีน (Text Screen) พร้อมทั้งแสดงตัวอย่างการวางชิ้นงาน

ซึ่งสามารถอธิบายในรายละเอียดได้ดังนี้ คือ โปรแกรมจะเริ่มทำงานจากการอ่านข้อมูลของเส้นต่าง ๆ ที่ประกอบกันเป็นรูปทรงชิ้นงานเข้ามาเก็บไว้ในหน่วยความจำ ข้อมูลที่รับเข้ามาจะอยู่ในรูปของโพลีไลน์ (Polyline) ซึ่งจะนำไปถอดออกมาเป็นข้อมูลในรูปของเส้น ซึ่งจะแบ่งเป็น 2 ชนิด คือ ข้อมูลของเส้นตรง (Line) และข้อมูลของส่วนโค้ง (Arc) กรณีที่เป็นเส้นตรงก็จะหาค่าพิกัดของจุดเริ่มต้นและจุดสิ้นสุดของเส้น กรณีที่เป็นส่วนโค้งก็จะต้องหาค่าของจุดศูนย์กลาง ค่ารัศมี ค่ามุมเริ่มต้นของส่วนโค้ง ค่ามุมสิ้นสุดของส่วนโค้ง พิกัดของจุดเริ่มต้นส่วนโค้ง และพิกัดของสิ้นสุดส่วนโค้ง จากนั้นก็จะนำค่าทั้งหมดไป คำนวณหาจุดศูนย์ถ่วงของรูปทรง เพื่อจะใช้เป็นจุดศูนย์กลางในการหมุนเส้นไปยังตำแหน่งต่าง ๆ การหาตำแหน่งจุดศูนย์ถ่วงของรูปปิดนั้น มีวิธีการหาได้หลายวิธี แต่ในที่นี้จะใช้วิธีการหาโมเมนต์ของเส้น ทั้งทางแกน X และแกน Z

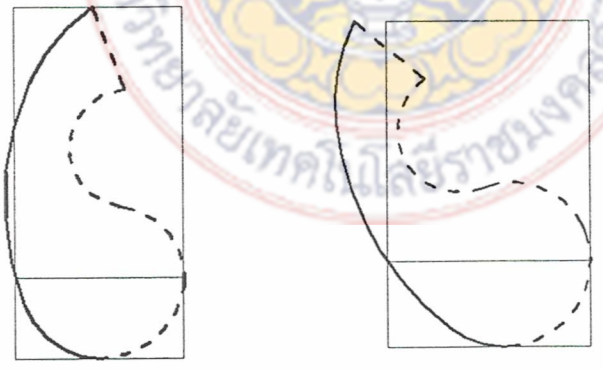
ภาพที่ 3-4 จุดศูนย์ถ่วงของเส้นที่ได้

การอ่านข้อมูลของเส้นทั้งหมดจะทำให้ทราบตำแหน่งสูงสุด หรือค่าพิกัดมากสุดทางแกน Y และตำแหน่งต่ำสุดหรือค่าพัดน้อยสุดทางแกน Y ในขณะนั้นได้ พิกัดของจุดสูงสุดและต่ำสุดที่ได้ จะถูกนำมาเป็นจุดแบ่งเพื่อแบ่งเส้นออกเป็น 2 กลุ่ม คือกลุ่มอยู่ทางซ้ายมือและกลุ่มที่อยู่ทางขวามือ

ภาพที่ 3-5 ข้อมูลของเส้นทั้งสองด้าน

การทำงานในขั้นตอนต่อมาก็คือ การอ่านข้อมูลของเส้นทั้งซ้ายและขวาพร้อม ๆ กัน เพื่อหาระยะที่กว้างที่สุดทางแกน X ของรูปทรงในตำแหน่งที่วางอยู่ในขณะนั้น การทำงานในขั้นตอนนี้จำเป็นจะต้องอาศัยฟังก์ชันย่อยช่วยในการทำงานได้แก่ ฟังก์ชันหาระยะกว้างมากสุดในกรณีเส้นตรงกับเส้นตรง ฟังก์ชันหาระยะกว้างมากสุดในกรณีเส้นตรงกับส่วนโค้ง และฟังก์ชันหาระยะกว้างมากสุดในกรณีส่วนโค้งกับส่วนโค้ง

การทราบค่าสูงสุด ต่ำสุด และค่าความกว้างมากสุดของชิ้นงาน จะทำให้คำนวณหาค่าพื้นที่แบลงค์ ได้ เมื่อนำไปคำนวณร่วมกับค่า พื้นที่ของชิ้นงานซึ่งได้จากการอ่านค่าด้วยฟังก์ชันของออดตแคดที่มีอยู่แล้ว ก็จะทำให้หาประสิทธิภาพของการใช้พื้นที่ในขณะตำแหน่งการวางชิ้นงานในขณะนั้น ได้



ภาพที่ 3-6 การหาพื้นที่แบลงค์ ที่ตำแหน่งการวางชิ้นงานในมุมต่าง ๆ

การทำงานต่อไปก็คือการคำนวณเส้นให้หมุนไปเท่ากับค่าความละเอียดของการหมุนที่ต้องการ แล้วก็เริ่มหาตำแหน่งจุดสูงสุด ต่ำสุด และค่าความกว้างของชิ้นงานในขณะนั้น ทำให้สามารถหาค่าพื้นที่แบลงค์และหาประสิทธิภาพได้

โปรแกรมจะวนลูปหาประสิทธิภาพที่ตำแหน่งมุมต่าง ๆ จนครบ 180 องศา และจำค่าองศาที่ทำให้ได้ประสิทธิภาพสูงสุด ก็จะเป็นคำตอบของการจัดเรียงชิ้นงานรูปทรงนั้น

```
**********************************************

 This is the data for 1 way strip layout.

    The best angle = 138.0 degree.
 Distance for feed = 103.612 mm.
       Strip width = 185.925 mm.
     The Blank Area = 19264.1.<1 workpiece>
       The Efficiency = 74.1 %

**********************************************
```

ภาพที่ 3-7 การแสดงผลที่เท็กซ์สกรีน

ในการคำนวณหาประสิทธิภาพที่แท้จริง ก็จะนำค่าระยะเผื่อขอบมาบวกเพิ่มทั้งค่าระยะเผื่อขอบสตริปและค่าระยะเผื่อขอบระยะห่างระหว่างชิ้นงาน

ในขั้นตอนสุดท้ายก็จะแสดงภาพตัวอย่างของการวางชิ้นงาน โดยการคัดลอกชิ้นงานไปวางถัดไปจากชิ้นงานเดิม ห่างออกไปเท่ากับความกว้างของพื้นที่แบลงค์บวกกับ ระยะเผื่อขอบเพื่อแสดงให้เห็นเป็นตัวอย่างของการจัดเรียงชิ้นงานที่ให้ค่าประสิทธิภาพมากสุด

ภาพที่ 3-8 ตัวอย่างการจัดเรียงชิ้นงานที่ได้

3.2.3. การหาค่ามุมที่ให้ประสิทธิภาพสูงสุดในกรณีของการป้อนสองทาง มีลำดับขั้นตอนในการทำงานดังนี้

1. อ่านข้อมูลของเส้นต่าง ๆ เข้ามาในหน่วยความจำ
2. คำนวณหาตำแหน่งจุดศูนย์ถ่วงของรูปทรง
3. หาตำแหน่งสูงสุดและต่ำสุดของรูปทรง
4. อ่านข้อมูลของเส้นทางด้านขวาและคำนวณให้หมุนไป 180 องศา และนำข้อมูลเก็บไว้ในหน่วยความจำ
5. อ่านข้อมูลของเส้นทางด้านซ้ายและคำนวณให้หมุนไป 180 องศา และนำข้อมูลเก็บไว้ในหน่วยความจำ
6. อ่านข้อมูลของเส้นทางด้านขวาเดิมและ ข้อมูลทางด้านขวาที่หมุนไปแล้ว 180 องศา จากตำแหน่งสูงสุดลงมาทั้งสองด้านเพื่อค้นหาค่าความกว้างมากสุด
7. อ่านข้อมูลของเส้นทางด้านซ้ายเดิมและ ข้อมูลทางด้านซ้ายที่หมุนไปแล้ว 180 องศา จากตำแหน่งสูงสุดลงมาทั้งสองด้านเพื่อค้นหาค่าความกว้างมากสุด
8. นำค่ามากสุดที่ได้จากข้อ 6 และ 7 มารวมกับค่าระยะเผื่อระหว่างชิ้นงาน ไปคำนวณค่าประสิทธิภาพของการใช้พื้นที่
9. เปลี่ยนมุมวางชิ้นงานโดยหมุนไปเท่ากับค่าความละเอียดที่กำหนด แล้วเริ่มทำในขั้นตอนที่ 3 จนถึง ขั้นตอนที่ 8
10. เปลี่ยนมุมวางชิ้นงานจนกระทั่งครบ 360 องศา
11. นำค่ามุมที่ให้ประสิทธิภาพการจัดเรียงชิ้นงานมากสุด แสดงผลทางหน้าจอเท็กซ์สกรีนพร้อมทั้งแสดงตัวอย่างการวางชิ้นงาน

ซึ่งสามารถอธิบายในรายละเอียดได้ดังนี้ โปรแกรมจะเริ่มทำงานจากการอ่านข้อมูลของเส้นต่าง ๆ ที่ประกอบกันเป็นรูปทรงชิ้นงานเข้ามาเก็บไว้ในหน่วยความจำ ทำนองเดียวกับการป้อนทางเดียวและนำไปถอดออกมาเป็นข้อมูลในรูปของข้อมูลเส้นตรงและข้อมูลของส่วนโค้ง จากนั้นนำค่าทั้งหมดไปคำนวณหาจุดศูนย์ถ่วงของรูปทรง เพื่อจะใช้เป็นจุดศูนย์กลางในการหมุนเส้นด้วยวิธีการโมเมนต์ของเส้น ทั้งทางแกน X และแกน Z เมื่อเสร็จในขั้นตอนนี้จะได้พิกัดของจุดศูนย์ถ่วงของรูปทรง เช่นเดียวกับกรณีป้อนหนึ่งทาง

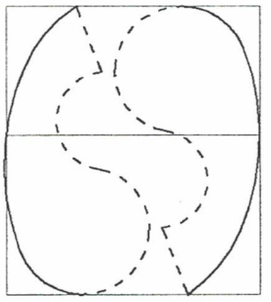
ภาพที่ 3-9 คำนวณได้จุดศูนย์ถ่วงของเส้น

การอ่านข้อมูลของเส้นทั้งหมดจะทำให้ทราบตำแหน่งสูงสุด หรือค่าพิกัดมากสุดทางแกน Y และตำแหน่งต่ำสุดหรือค่าพัดน้อยสุดทางแกน Y ในขณะนั้นได้ พิกัดของจุดสูงสุดและต่ำสุดที่ได้ จะถูกนำมาเป็นจุดแบ่งเพื่อแบ่งเส้นออกเป็น 2 กลุ่ม คือกลุ่มอยู่ทางซ้ายมือและกลุ่มที่อยู่ทางขวามือ

การทำงานในขั้นตอนต่อมาคือการนำข้อมูลเฉพาะกลุ่มที่อยู่ทางด้านขวามาคำนวณให้หมุนไป 180 องศา และอ่านข้อมูลชุดใหม่นี้พร้อม ๆ กับกลุ่มข้อมูลที่เป็นชุดที่อยู่ด้านขวาเดิม เพื่อหาระยะที่กว้างสุดทางแกน X ระยะกว้างสุดที่ได้นี้เมื่อบวกกับระยะเผื่อขอบระหว่างชิ้นงานจะได้ตำแหน่งการวางชิ้นงานชิ้นแรกของการป้อนครั้งที่สอง สำหรับในกรณีนี้ก็คือ ชิ้นงานเดิมที่หมุนไป 180 วงศานั่นเอง

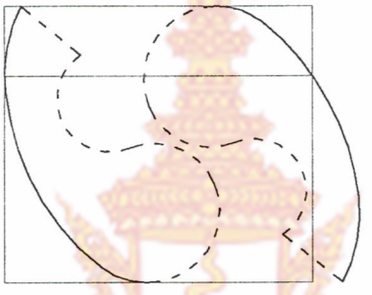
ภาพที่ 3-10 การอ่านพิกัดของเส้นทีละด้าน

ขั้นตอนต่อไปก็คือให้อ่านข้อมูลในกลุ่มซ้ายและคำนวณให้เส้นต่าง ๆ หมุนไป 180 องศาและเก็บไว้ในหน่วยความจำ ดังนั้นเมื่อให้โปรแกรมเริ่มอ่านข้อมูลของเส้นใหม่อีกครั้งหนึ่ง โดยให้อ่านข้อมูลของเส้นในกลุ่มซ้ายเดิมพร้อมกับข้อมูลในกลุ่มซ้ายที่หมุนไปแล้ว 180 องศา เพื่อหาระยะกว้างที่สุดทางแกน X ของรูปทรงในตำแหน่งที่วางชิ้นงานอยู่ในขณะนั้น



ภาพที่ 3-11 พื้นที่แบลงค์ ที่ตำแหน่งการวางชิ้นงานในตำแหน่งเริ่มต้น

จากค่าความกว้างและความสูงมากสุดทำให้สามารถะหาค่าพื้นที่แบลงค์และประสิทธิภาพของการวางงานในตำแหน่งนั้นได้



ภาพที่ 3-12 พื้นที่แบลงค์ที่ตำแหน่งการวางชิ้นงานในมุมที่หมุนไป

เมื่อทำการวนลูปเพื่อหาประสิทธิภาพที่ตำแหน่งของการวางชิ้นงานในมุมต่าง ๆ จนครบ 180 องศา และจำค่าองศาที่ทำให้ได้ประสิทธิภาพสูงสุดไว้ ก็จะสามารถทราบคำตอบของการวางชิ้นงานแบบแถวเดียวป้อนสองทางได้

```
*********************************************

 This is the data for 2 way strip layout.

    The best angle = 28.0 degree.
 Distance for feed = 190.308 mm.
       Strip width = 203.46 mm.
     The Blank Area = 38720.2.<2 workpiece>
       The Efficiency = 73.7324 %

*********************************************
```

ภาพที่ 3-13 การแสดงผลที่เท็กซ์สกรีน

การทำงานในขั้นตอนสุดท้ายก็คือ จะแสดงภาพตัวอย่างของการวางชิ้นงานโดยการคัดลอกชิ้นงานและหมุนไป 180 องศา นำไปวางถัดไปจากชิ้นงานเดิม ห่างออกไปเท่ากับความกว้างของมากสุดทางแกน X ของชิ้นงานในขณะนั้น บวกกับ ระยะเผื่อขอบแล้วทำการคัดลอกอีกครั้งโดยต้นแบบใหม่ก็คือ ชิ้นงานต้นแบบเดิมและชิ้นงานที่หมุนไปแล้ว 180 องศา นำไปวางเรียงกันห่างออกไปเท่ากับ ระยะป้อน บวกกับระยะเผื่อช่องว่างระหว่างชิ้นงาน ก็จะเป็นตัวอย่างของการจัดเรียงชิ้นงานที่ให้ค่าประสิทธิภาพมากสุดในการวางแบบแถวเดียวป้อนสองทาง

ภาพที่ 3-14 ตัวอย่างการจัดเรียงชิ้นงานแบบป้อนสองทางที่ได้จากการคำนวณ

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

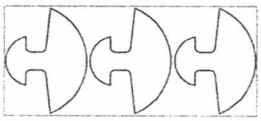
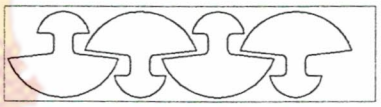
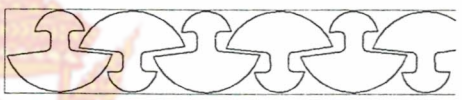
บทที่ 4
ผลการทดลอง และสรุป

ในบทนี้จะเป็นการแสดงผลของการวิจัยที่เป็นการเปรียบเทียบผลของการทดลองจัดเรียงชิ้นงานลงบนแผ่นวัสดุ ในด้านประสิทธิภาพของการใช้พื้นที่ของแผ่นวัสดุ ในการเลเอาท์แผ่นสตริป และในด้านเวลาของการทำงาน กรณีศึกษาเหล่านี้เป็นการใช้แบบชิ้นงานที่พบเห็นได้จากตำราการออกแบบแม่พิมพ์ต่าง ๆ จำนวน 2 0 ชิ้นงาน โดยใช้นักศึกษาที่ผ่านการเรียนวิชาการออกแบบแม่พิมพ์มาแล้ว และมีความสามารถในการใช้โปรแกรมช่วยในการเขียนแบบพอสมควร ในการทดลองจัดเรียงชิ้นงานด้วยคนให้สามารถจัดเรียงโดยใช้เครื่องมือเขียนแบบได้ทุกชนิดรวมทั้งการใช้โปรแกรมช่วยในการออกแบบเขียนแบบที่ถนัด แต่ไม่มีการใช้โปรแกรมสำเร็จรูปที่ช่วยในการจัดเรียงชิ้นงานแม่พิมพ์ เนื่องจากยังไม่มีใช้อยู่ในการเรียนการสอนจริง

ในการทดลอง ได้จัดให้นักศึกษาออกแบบวิธีการจัดเรียงชิ้นงานที่เห็นว่าประหยัดวัสดุมากสุด ในชิ้นงานแต่ละชิ้นจะออกแบบเป็นสองลักษณะ คือ การป้อนทางเดียวและการป้อนสองทาง โดยจับเวลาในการทำงานและวัดประสิทธิภาพในการใช้วัสดุ นำผลที่ได้มาเปรียบเทียบกับผลการจัดเรียงด้วยฟังก์ชันที่สร้างขึ้น ทั้งวิธีการป้อนแบบทางเดียวและแบบป้อนสองทาง ซึ่งในการคิดประสิทธิภาพในการใช้วัสดุนั้นจะแบ่งเป็น 2 กรณี คือ ในกรณีการป้อนหนึ่งทางจะคิดต่อชิ้นงาน 1 ชิ้น ส่วนในกรณีป้อนสองทางจะคิดต่อชิ้นงาน 2 ชิ้น โดยไม่คิดค่าระยะเผื่อระหว่างชิ้นงาน

ผลการเปรียบเทียบการจัดเรียงชิ้นงานแม่พิมพ์

กรณีศึกษาที่ 1 : ชิ้นงานที่ 1

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-1 ผลการจัดเรียงชิ้นงานที่ 1

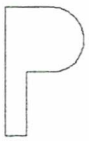
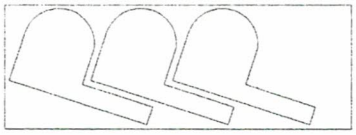
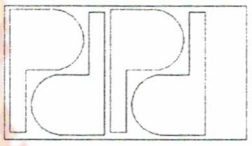
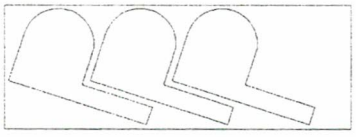
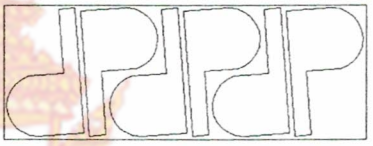
ตารางที่ 1 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 1

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	90	15127	49.29	10
คน : ป้อนแบบสองทาง	0	2299.7	71.1	20
โปรแกรม : ป้อนแบบหนึ่งทาง	25	1253.12	67.22	1
โปรแกรม : ป้อนแบบสองทาง	1	2165.4	77.8	1

สรุปผล

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 2 : ชิ้นงานที่ 2

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-2 ผลการจัดเรียงชิ้นงานที่ 2

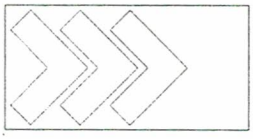
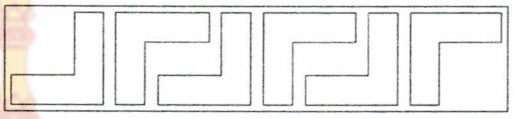
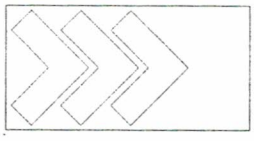
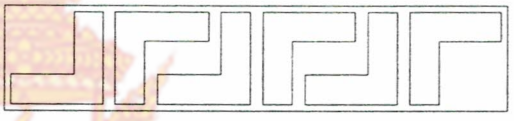
ตารางที่ 2 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 2

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	72	23317	82	10
คน : ป้อนแบบสองทาง	0	43211	85	20
โปรแกรม : ป้อนแบบหนึ่งทาง	72	23317	82.	1
โปรแกรม : ป้อนแบบสองทาง	4	41889	92	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 3 : ชิ้นงานที่ 3

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-3 ผลการจัดเรียงชิ้นงานที่

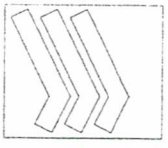
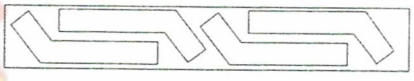
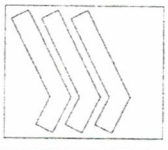
ตารางที่ 3 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 3

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	45	2000	80	5
คน : ป้อนแบบสองทาง	0	3500	91	10
โปรแกรม : ป้อนแบบหนึ่งทาง	45	2000	80	1
โปรแกรม : ป้อนแบบสองทาง	0	3500	91	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมและคนให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 4 : ชิ้นงานที่ 4

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง

ภาพที่ 4-4 ผลการจัดเรียงชิ้นงานที่ 4

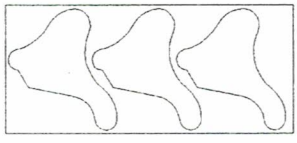
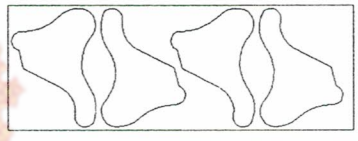
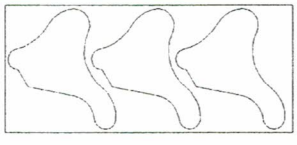
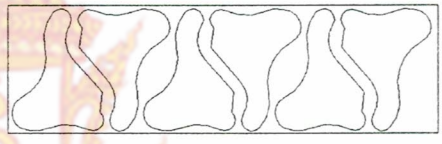
ตารางที่ 4 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 4

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	116	1054	91	15
คน : ป้อนแบบสองทาง	0	2261	85	15
โปรแกรม : ป้อนแบบหนึ่งทาง	116	1054	91	1
โปรแกรม : ป้อนแบบสองทาง	0	2261	85	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมและคนให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 5 : ชิ้นงานที่ 5

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-5 ผลการจัดเรียงชิ้นงานที่ 5

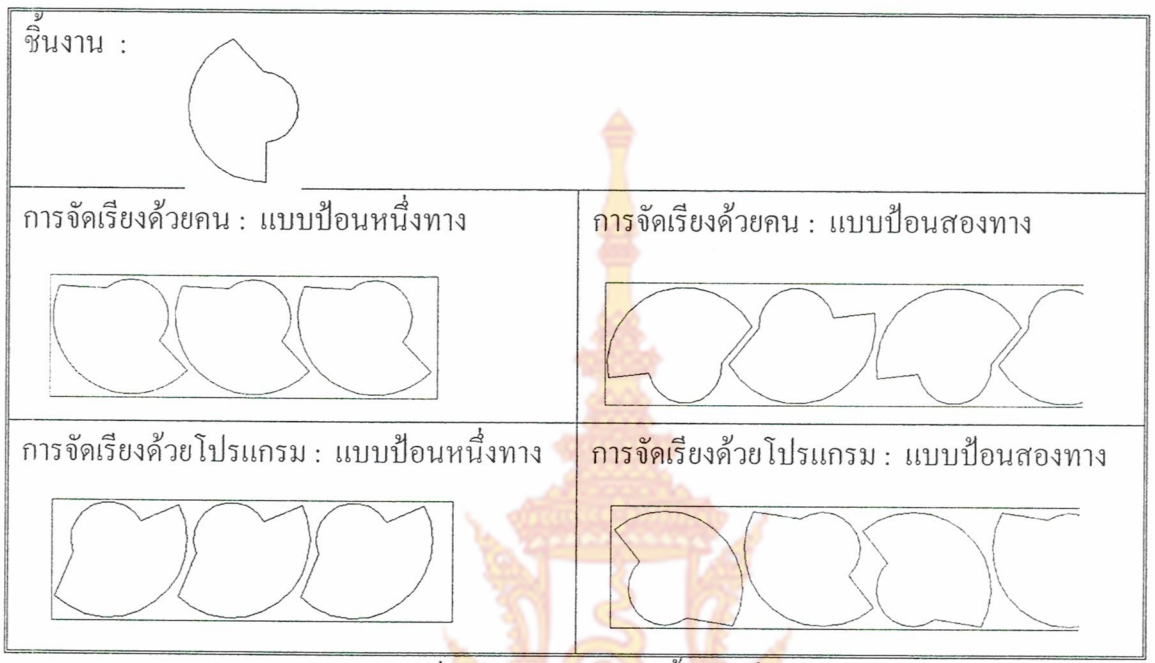
ตารางที่ 5 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 5

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	0	2743	64	40
คน : ป้อนแบบสองทาง		5772	61	60
โปรแกรม : ป้อนแบบหนึ่งทาง	16	2743	64	1
โปรแกรม : ป้อนแบบสองทาง	158	4339	81	2

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบหนึ่งทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 6 : ชิ้นงานที่ 6



ภาพที่ 4-6 ผลการจัดเรียงชิ้นงานที่ 6

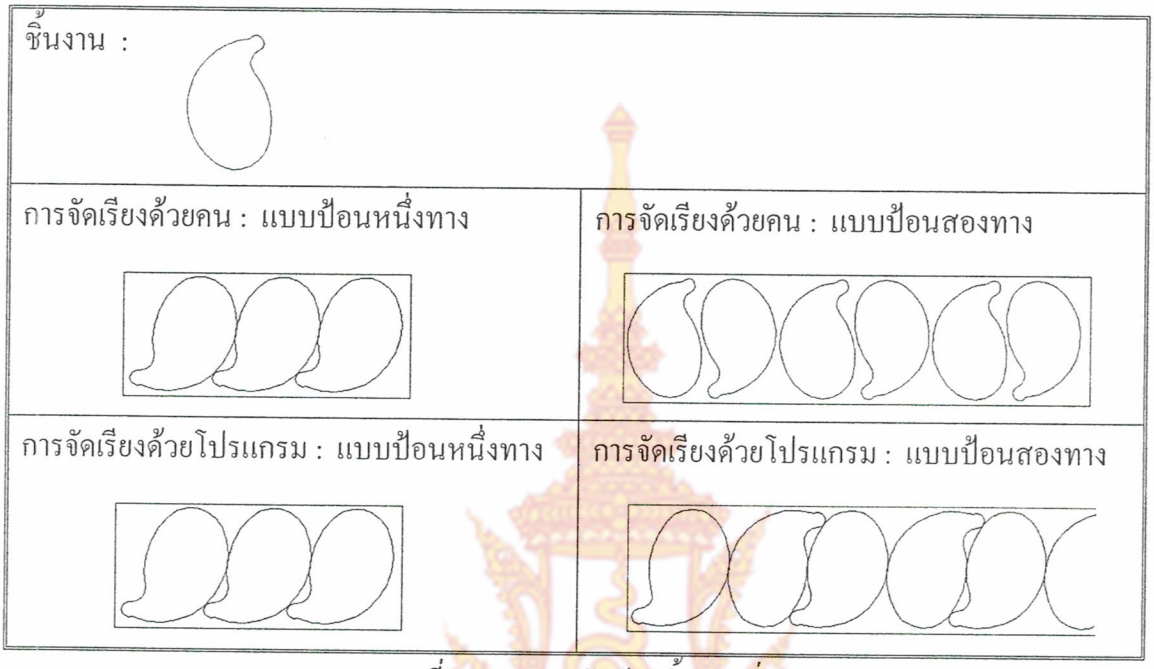
ตารางที่ 6 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 6

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง		4210	73	20
คน : ป้อนแบบสองทาง		7983	77	40
โปรแกรม : ป้อนแบบหนึ่งทาง	23	3612	85	1
โปรแกรม : ป้อนแบบสองทาง	122	7389	83	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 7 : ชิ้นงานที่ 7



ภาพที่ 4-7 ผลการจัดเรียงชิ้นงานที่ 7

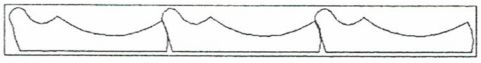
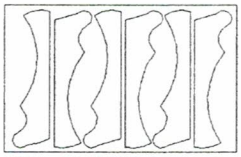
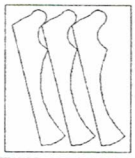
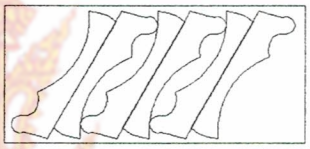
ตารางที่ 7 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 7

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	151	2660	83.4	10
คน : ป้อนแบบสองทาง	0	5761	77	15
โปรแกรม : ป้อนแบบหนึ่งทาง	151	2660	83.4	2
โปรแกรม : ป้อนแบบสองทาง	157	5406	82.11	2

<u>สรุปผล</u>

การจัดเรียงแบบป้อนหนึ่งทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรมและคนในกรณีการป้อนแบบหนึ่งทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 8 : ชิ้นงานที่ 8

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-8 ผลการจัดเรียงชิ้นงานที่ 8

ตารางที่ 8 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 8

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	90	1225	60	10
คน : ป้อนแบบสองทาง	0	2213	65.47	20
โปรแกรม : ป้อนแบบหนึ่งทาง	14	1107	65.4	1
โปรแกรม : ป้อนแบบสองทาง	150	1988	72.86	1

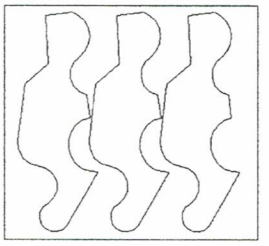
<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

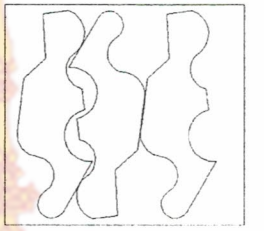
กรณีศึกษาที่ 9 : ชิ้นงานที่ 9

ชิ้นงาน :

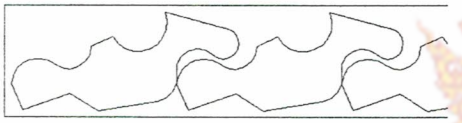
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง



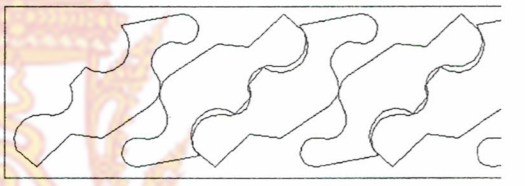
การจัดเรียงด้วยคน : แบบป้อนสองทาง



การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง



การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง



ภาพที่ 4-9 ผลการจัดเรียงชิ้นงานที่ 9

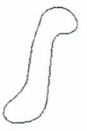
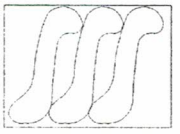
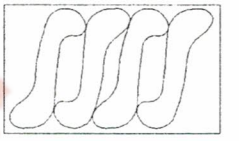
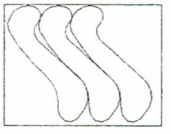
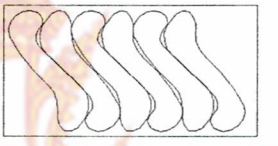
ตารางที่ 9 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 9

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	90	3097	61	40
คน : ป้อนแบบสองทาง	90	5866	64.8	60
โปรแกรม : ป้อนแบบหนึ่งทาง	16	3056	62.2	2
โปรแกรม : ป้อนแบบสองทาง	41	5568	68	2

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 10 : ชิ้นงานที่ 10

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-10 ผลการจัดเรียงชิ้นงานที่ 10

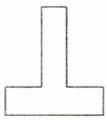
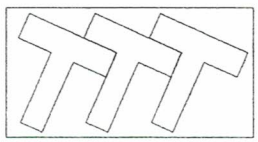
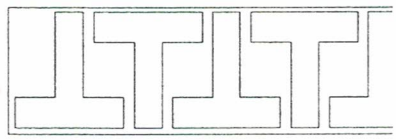
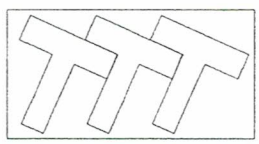
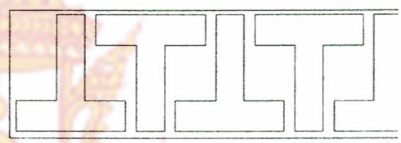
ตารางที่ 10 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 10

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง		2189	67.1	30
คน : ป้อนแบบสองทาง		4340	67.75	40
โปรแกรม : ป้อนแบบหนึ่งทาง	53	1704	86.26	2
โปรแกรม : ป้อนแบบสองทาง	51	3316	88.65	2

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 11 : ชิ้นงานที่ 11

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-11 ผลการจัดเรียงชิ้นงานที่ 11

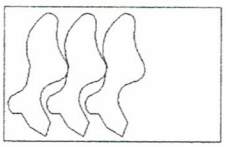
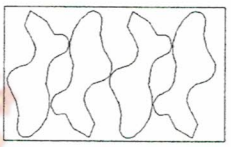
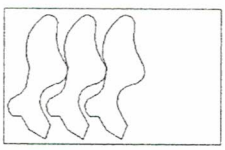
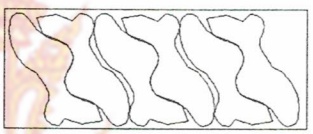
ตารางที่ 11 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 11

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	156	3501	56.76	15
คน : ป้อนแบบสองทาง	0	5549	71.63	10
โปรแกรม : ป้อนแบบหนึ่งทาง	156	3501	56.76	1
โปรแกรม : ป้อนแบบสองทาง	0	5549	71.63	1

<u>สรุปผล</u>

การจัดเรียงแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรม และคนให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 12 : ชิ้นงานที่ 12

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-12 ผลการจัดเรียงชิ้นงานที่ 12

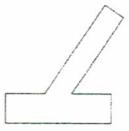
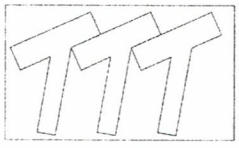
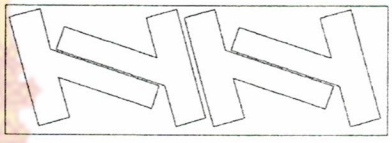
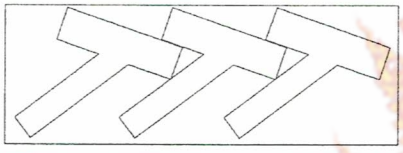
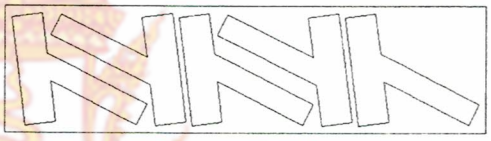
ตารางที่ 12 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 12

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	0	1167	71.4	15
คน : ป้อนแบบสองทาง		2959	60.2	20
โปรแกรม : ป้อนแบบหนึ่งทาง	0	1167	71.4	1
โปรแกรม : ป้อนแบบสองทาง	42	1911	87.22	2

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบหนึ่งทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 13 : ชิ้นงานที่ 13

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-13 ผลการจัดเรียงชิ้นงานที่ 13

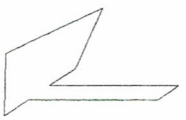
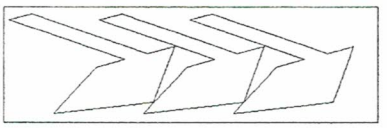
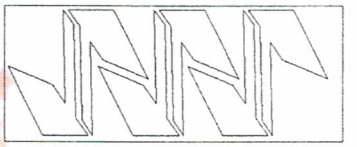
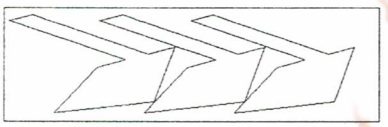
ตารางที่ 13 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 13

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	180	3963	49.7	6
คน : ป้อนแบบสองทาง		6184	63.72	15
โปรแกรม : ป้อนแบบหนึ่งทาง	161	3828	51.45	1
โปรแกรม : ป้อนแบบสองทาง	97	5936	66.37	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 14 : ชิ้นงานที่ 14

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง

ภาพที่ 4-14 ผลการจัดเรียงชิ้นงานที่ 14

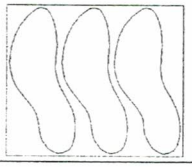
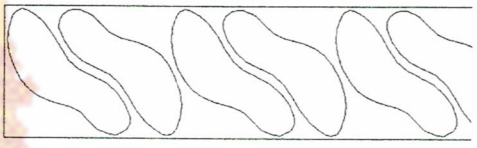
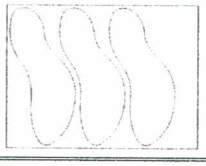
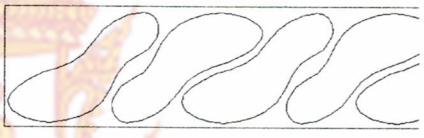
ตารางที่ 14 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 14

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	161	4114	63.2	30
คน : ป้อนแบบสองทาง	90	7993	65.03	20
โปรแกรม : ป้อนแบบหนึ่งทาง	161	4114	63.2	1
โปรแกรม : ป้อนแบบสองทาง	100	8477	61.33	1

<u>สรุปผล</u>

การจัดเรียงด้วย คนแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรมและคนในกรณีการป้อนแบบหนึ่งทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 15 : ชิ้นงานที่ 15

ชิ้นงาน :	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-15 ผลการจัดเรียงชิ้นงานที่ 15

ตารางที่ 15 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 15

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง		8371	81	40
คน : ป้อนแบบสองทาง	220	18412	73.68	60
โปรแกรม : ป้อนแบบหนึ่งทาง	106	8168	83	2
โปรแกรม : ป้อนแบบสองทาง	36	15864	85.5	3

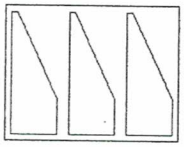
<u>สรุปผล</u>

การจัดเรียงด้วยโปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

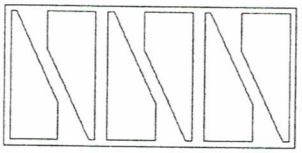
กรณีศึกษาที่ 16 : ชิ้นงานที่ 16

ชิ้นงาน :

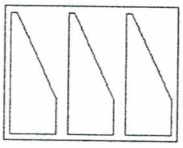
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง



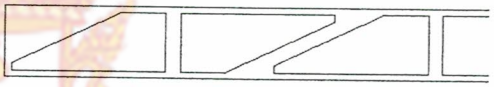
การจัดเรียงด้วยคน : แบบป้อนสองทาง



การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง



การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง



ภาพที่ 4-16 ผลการจัดเรียงชิ้นงานที่ 16

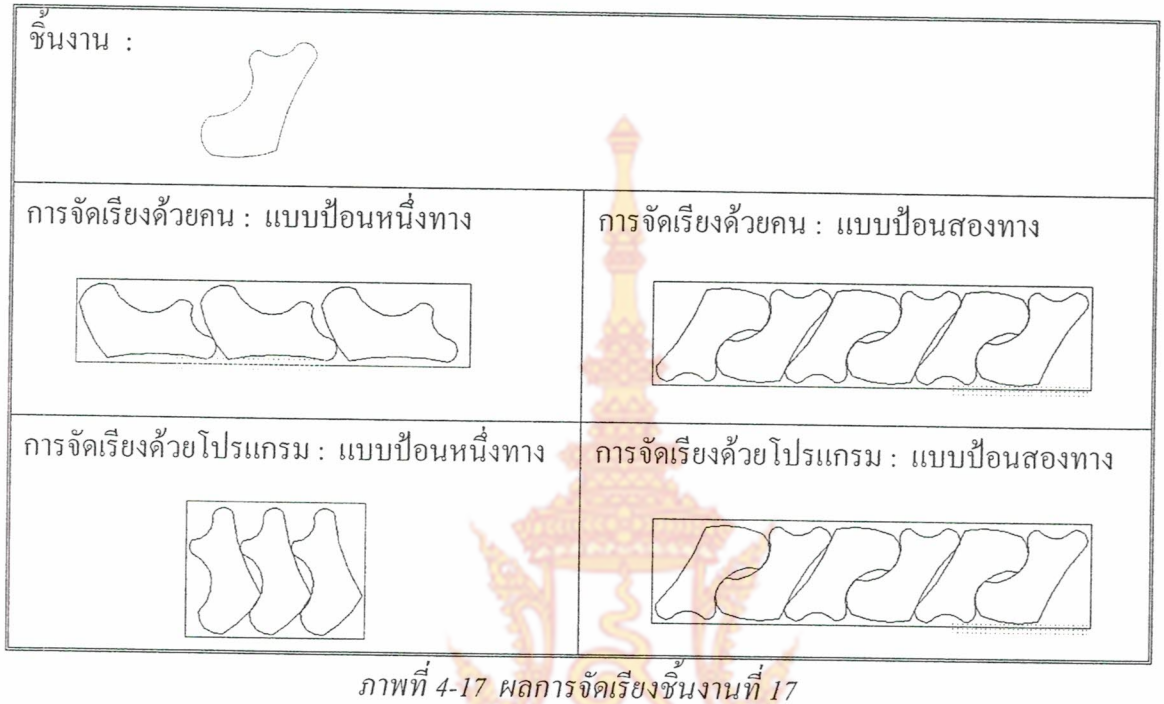
ตารางที่ 16 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 16

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	0	6080	74.75	7
คน : ป้อนแบบสองทาง	0	9793	92.8	10
โปรแกรม : ป้อนแบบหนึ่งทาง	0	6080	74.75	1
โปรแกรม : ป้อนแบบสองทาง	90	9469	96	1

<u>สรุปผล</u>

การจัดเรียงด้วย โปรแกรมแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบหนึ่งทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 17 : ชิ้นงานที่ 17



ภาพที่ 4-17 ผลการจัดเรียงชิ้นงานที่ 17

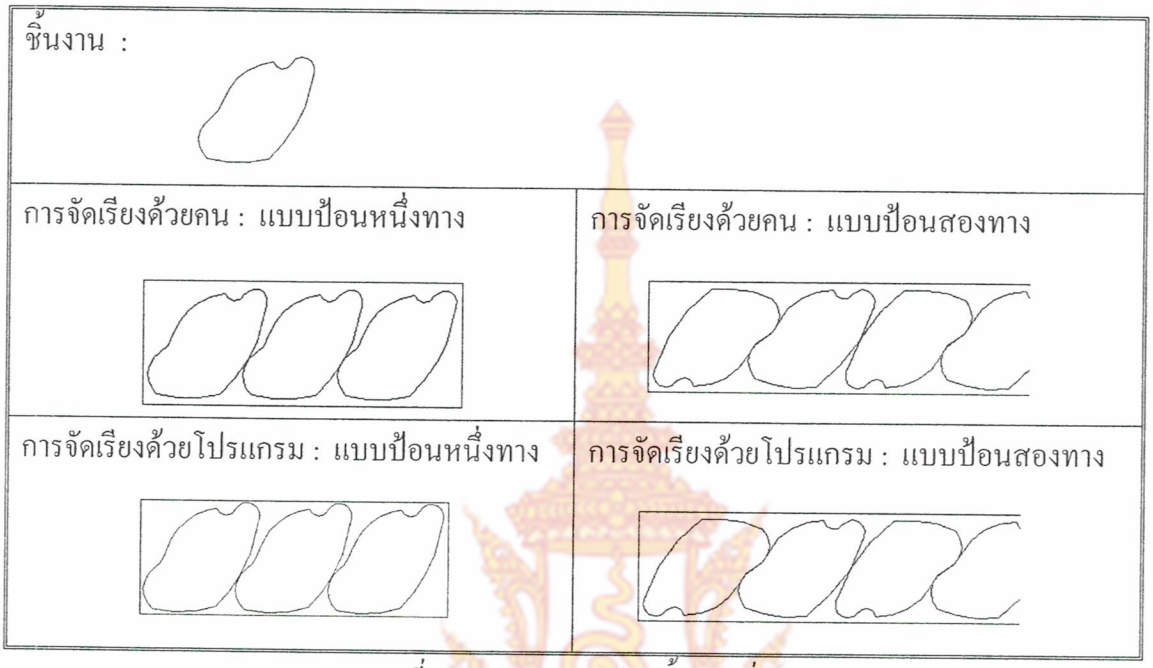
ตารางที่ 17 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 17

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง		3524	68.86	7
คน : ป้อนแบบสองทาง	170	5562	87.27	30
โปรแกรม : ป้อนแบบหนึ่งทาง	42	3262	74.39	1
โปรแกรม : ป้อนแบบสองทาง	170	5562	87.27	2

<u>สรุปผล</u>

การจัดเรียงแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบสองทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 18 : ชิ้นงานที่ 18



ภาพที่ 4-18 ผลการจัดเรียงชิ้นงานที่ 18

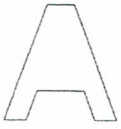
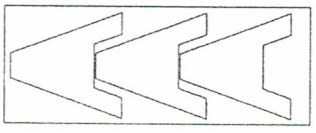
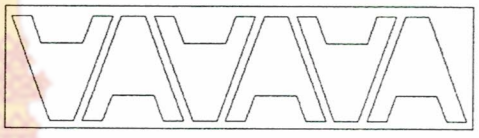
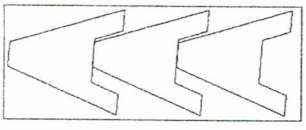
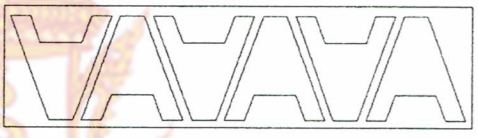
ตารางที่ 18 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 18

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	0	4100	77	6
คน : ป้อนแบบสองทาง	119	7467	84.6	20
โปรแกรม : ป้อนแบบหนึ่งทาง	2	3916	80.67	2
โปรแกรม : ป้อนแบบสองทาง	119	7467	84.6	2

<u>สรุปผล</u>

การจัดเรียงแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรม และคนในกรณีการป้อนแบบสองทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วย โปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสอง ทาง

กรณีศึกษาที่ 19 : ชิ้นงานที่ 19

ชิ้นงาน : 	
การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยคน : แบบป้อนสองทาง 
การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง 	การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง 

ภาพที่ 4-19 ผลการจัดเรียงชิ้นงานที่ 19

ตารางที่ 19 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 19

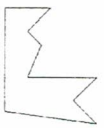
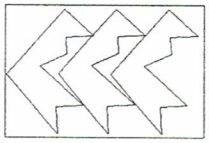
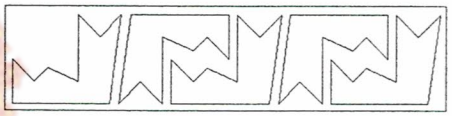
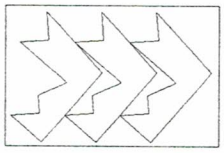
การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง	90	1041	63.4	5
คน : ป้อนแบบสองทาง	180	1663	79.3	5
โปรแกรม : ป้อนแบบหนึ่งทาง	86	1007	65.52	1
โปรแกรม : ป้อนแบบสองทาง	180	1663	79.3	1

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

สรุปผล

การจัดเรียงแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบสองทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

กรณีศึกษาที่ 20 : ชิ้นงานที่ 20

<table>
<tr><td colspan="2">ชิ้นงาน :
  </td></tr>
<tr><td>การจัดเรียงด้วยคน : แบบป้อนหนึ่งทาง
  </td><td>การจัดเรียงด้วยคน : แบบป้อนสองทาง
  </td></tr>
<tr><td>การจัดเรียงด้วยโปรแกรม : แบบป้อนหนึ่งทาง
  </td><td>การจัดเรียงด้วยโปรแกรม : แบบป้อนสองทาง
 </td></tr>
</table>

ภาพที่ 4-20 ผลการจัดเรียงชิ้นงานที่ 20

ตารางที่ 20 ผลการเปรียบเทียบการจัดเรียงชิ้นงานที่ 20

การจัดเรียงโดย	มุมวางชิ้นงาน (องศา)	Blank Area (mm^2)	ประสิทธิภาพ (%)	เวลา (นาที)
คน : ป้อนแบบหนึ่งทาง		1753	57.33	20
คน : ป้อนแบบสองทาง	90	2588	77.66	2
โปรแกรม : ป้อนแบบหนึ่งทาง	139	1695	59.29	1
โปรแกรม : ป้อนแบบสองทาง	90	2588	77.66	2

<u>สรุปผล</u>

การจัดเรียงแบบป้อนสองทางให้ประสิทธิภาพดีที่สุด และการจัดเรียงด้วย โปรแกรมและคนในกรณีการป้อนแบบสองทางให้ประสิทธิภาพไม่แตกต่างกัน แต่การจัดเรียงด้วยโปรแกรม ใช้เวลาน้อยกว่าการทำงานด้วยคนทั้งแบบการป้อนแบบหนึ่งทางและการป้อนแบบสองทาง

สรุป

ในกรณีที่ชิ้นงานมีรูปทรงที่ไม่ซับซ้อนมากนัก การจัดเรียงด้วยคนก็สามารถให้ค่าประสิทธิภาพสูงได้ แต่ในกรณีที่ชิ้นงานมีรูปทรงที่ซับซ้อนการจัดเรียงด้วยฟังก์ชันที่สร้างขึ้น จะช่วยเพิ่มประสิทธิภาพในการจัดเรียงวัสดุ และประหยัดเวลาในการทำงาน ได้มาก และจากการทดลองจะเห็นได้ว่าประสิทธิภาพสูงสุดในการจัดเรียงวัสดุไม่ขึ้นกับแบบของการป้อน ในบางกรณีการป้อนแบบสองให้ประสิทธิภาพได้มากกว่า แต่ในบางกรณีการป้อนแบบหนึ่งทางก็ให้ประสิทธิภาพได้มากกว่า ซึ่งถ้าหากต้องการหาคำตอบว่าวิธีการป้อนแบบใดจะให้ประสิทธิภาพสูงสุด การหาคำตอบด้วยคนอาจจะต้องใช้เวลามาก แต่การทำงานด้วยโปรแกรมจะสามารถให้คำตอบได้ในเวลาไม่มากนัก และค่าตัวเลขที่ได้จะเชื่อถือได้มากกว่าซึ่งในกรณีของการทำงานด้วยคนจะหาค่ามุมที่แน่นอนได้ยาก ซึ่งจะส่งผลถึงค่าที่ต้องคำนวณตามมาเช่น ระยะการป้อนชิ้นงาน และค่าประสิทธิภาพ เป็นต้น ซึ่งอาจจะทำให้ผลที่ได้ไม่เป็นไปตามแผนที่คาดการณ์ไว้ก็ได้

อย่างไรก็ตาม ในการทำงานด้วยฟังก์ชันที่สร้างขึ้นก็จะมีข้อจำกัด คือไม่สามารถหาคำตอบที่ถูกต้องได้ในกรณีที่ชิ้นงานมีรูปทรงเป็นส่วนเว้าเข้าไปมากและทำให้สามารถวางชิ้นงานสอดเข้าไปภายในอีกชิ้นได้ เช่นในกรณีของชิ้นงานที่ 14 การพิจาณาด้วยคนจะทำให้สามารถให้ผลคำตอบที่ดีกว่าได้

ข้อเสนอแนะทั่วไป

จากผลการทดลองข้างต้น สถานศึกษาที่มีการเรียนการสอนวิชาออกแบบแม่พิมพ์สามารถนำผลที่ได้ไปใช้ประโยชน์ ได้ ซึ่งจะช่วยให้นักศึกษาสามารถตรวจสอบผลการออกแบบด้วยตนเองได้

ข้อเสนอแนะในการวิจัยครั้งต่อไป

นอกจากการพัฒนาในส่วนที่เป็นข้อจำกัดของฟังก์ชันตามที่กล่าวแล้ว ผู้ที่มีความสนใจสามารถนำฟังก์ชันที่ได้ไปพัฒนาต่อไปในการเรียงชิ้นงาน ในงานอื่น ๆ ซึ่ง อัลกอริธึมที่มีอยู่ก่อนแล้ว โดยทั่วไปมักใช้วิธีการลากเส้นในแนวต่าง ๆ ซึ่งทำให้ต้องใช้เวลาในการทำงานนาน แต่ถ้าใช้วิธีการเก็บข้อมูลของเส้นต่าง ๆ ที่สำคัญ แล้วนำมาคำนวณหาค่ามากสุดเฉพาะในเงื่อนไขที่ต้องการจะช่วยให้ลดเวลาในการทำงานของโปรแกรมลงได้มาก

บรรณานุกรม

(1) ชาญชัย ทรัพยากร ประสิทธิ์ สวัสดิสรรพ์ และ วิรุฬ ประเสริฐสรนันท์ . 2539. การออกแบบแม่พิมพ์ . พิมพ์ครั้งที่ 11 . สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) . กรุงเทพ .

(2) สมพงษ์ ธีระคานนท์ . 2541 . "การออกแบบโพรเกรสซีฟดาย" . วารสารแมพิมพ์ 10 (เมย.-มิย. 2541) . หน้า 35-40

(3) ผดุงศิลป์ พิทักษ์ . 2541 . การสร้างโปรแกรมในการวางรูปแบบชิ้นงานแผ่นบางแบบ 2 มิติ . วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต . สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ . ภาควิชาวิศวกรรมการผลิต คณะวิศวกรรมศาสตร์ .

(4) บรรเลง ศรนิล .2524 . ตารางงานโลหะ . สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ . กรุงเทพ .

(5) ชาญ ถนัดงาน เชาวลิต แสนสุข ประสาทศิลป์ อ่อนอรรถ และ คณะ . 2533 . คู่มือการออกแบบและสร้างแม่พิมพ์ขนาดเล็ก . สถาบันพัฒนาอุตสาหกรรมเครื่องจักรกลและโลหะการ . กรุงเทพ

(6) กอบเกียรติ สระอุบล . 2537 . Advance AutoCAD Release 12 . โรงพิมพ์ เฮช. เอ็น. กรุ๊ป จำกัด . กรุงเทพ.

(7) จอร์จ โอมูระ. อนิรุต ลิ่วพาทอง แปล . 2536 .พื้นฐานการโปรแกรม AutoLISP . โรงพิมพ์ เฮช. เอ็น. กรุ๊ป จำกัด . กรุงเทพ.

(8) Jacobs S. 1994 . "On genetic algorithms for the packing of polygon" . European Journal of Operational Research 88 . pp 195-181

(9) Oliveira, J.F., Gomes , A.M., Ferreira, S. 2000 "TOPOS A new constructive algorithm for nesting ploblems", OR Spektrum 22. pp 263-284

(10) Karen Daniel, and Victor J. Milenkovic . 1996 . "Column – Base Strip Packing using Orderd and Compliant Containment" . Coputer Science 27 . pp 33-38

ภาคผนวก

โปรแกรมช่วยในการจัดเรียงชิ้นงานแม่พิมพ์

```
(defun IsWhat (tmp1)
  (setq enType (cdr (assoc '0 (cdr (entget tmp1))))
        LAYER_ (cdr (assoc '8 (cdr (entget tmp1))))) ;defun
(defun IsLne (tmp1)
  (setq tmp2 (cdr (entget tmp1)))
  (setq pStt (cdr (assoc '10 tmp2)) pEnd (cdr (assoc '11 tmp2)))
  ) ;defun
(defun IsArc (tmp1)
  (setq tmp2 (cdr (entget tmp1)))
  (setq pCen (cdr (assoc '10 tmp2))
        EndAng (cdr (assoc '51 tmp2))
        radi_ (cdr (assoc '40 tmp2))
        St_Ang (cdr (assoc '50 tmp2))
        pStt (polar pCen St_Ang radi_)
        pEnd (polar pCen EndAng radi_))) ;defun
(defun M&RotPt (tmp5)
  (setq tmp4 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) angRot) (distance c_tdPt tmp5))) )
(defun Ro&MPt (tmp5)
  (setq tmp3 (+ yMax yMin))
  (setq tmp4 (polar '(0.0 0.0) (+ (+ (angle c_tdPt tmp5) angRot) pi) (distance c_tdPt tmp5)))
  (setq tmp6 (polar tmp4 (* pi 0.5) tmp3)) )
(defun Ro&M_2 (tmp5)
  (setq tmp3 (+ yMax yMin))
  (setq tmp4 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) pi) (distance c_tdPt tmp5)))
  (setq tmp6 (polar tmp4 (* pi 0.5) tmp3)) )
(defun Ro&M_3 (tmp5 tmp7)
  (setq tmp3 (+ yMax yMin))
  (setq tmp4 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) pi) (distance c_tdPt tmp5)))
  (setq tmp6 (polar tmp4 (* pi 0.5) tmp3))
  (setq tmp8 (polar tmp6 0.0 tmp7)) )
(defun inLYoN (tmp1 tmp2 tmp3)
  (setq tmp4 (abs (- (cadr tmp2) (cadr tmp1))))
  (if (equal tmp4 0 preci_)
    (setq disNow 0.0)
    (if (equal tmp4 (+ (abs (- (cadr tmp3) (cadr tmp1))) (abs (- (cadr tmp2) (cadr tmp3))))
        preci_)
      (progn
        (setq disNow (abs (- (+ (/ (* (- (cadr tmp3) (cadr tmp1)) (- (car tmp2) (car tmp1))) (-
        (cadr tmp2) (cadr tmp1))) (car tmp1)) (car tmp3))))
      ) ;progn
      (setq disNow 0.0) ) ; if
    ) ;(if (equal tmp4 0 preci_)
  (if (> disNow disMem) (setq disMem disNow pMem0 tmp3)) )
(defun inL_N2 (tmp1 tmp2 tmp3)
  (setq tmp4 (abs (- (cadr tmp2) (cadr tmp1))))
  (if (equal tmp4 0 preci_)
    (if (equal (cadr tmp3) (cadr tmp1) preci_)
      (setq tmp5 (/ (+ (car tmp1) (car tmp2)) 2)
            inLne 1)
      (setq inLne 0))
    (if (equal tmp4 (+ (abs (- (cadr tmp3) (cadr tmp1))) (abs (- (cadr tmp2) (cadr tmp3))))
        preci_)
```

```
      (setq tmp5 (+ (/ (* (- (cadr tmp3) (cadr tmp1)) (- (car tmp2) (car tmp1))) (- (cadr tmp2)
        (cadr tmp1))) (car tmp1))
            inLne 1)
      (setq inLne 0) ) ; if
    ) ;(if (equal tmp4 0 preci_)
  )
 (defun inAYoN (tmp1 tmp2 tmp3 tmp4)
  (if (= tmp4 nil)
    (setq InArc 0)
      (progn
        (setq tmp5 (- (angle tmp1 tmp3) (angle tmp1 tmp2)) )
        (if (<= tmp5 0) (setq tmp5 (+ tmp5 (* 2 pi))))
        (setq tmp6 (- (angle tmp1 tmp4) (angle tmp1 tmp2)) )
        (if (< tmp6 0) (setq tmp6 (+ tmp6 (* 2 pi))))
        (if (> tmp6 tmp5) (setq InArc 0) (setq InArc 1))
        ) ;progn
    ) ;if
  );;defun
(defun XinCir (tmp1 tmp2 tmp3)
  (setq tmp4 (car tmp1) tmp5 (cadr tmp1))
  (setq tmp6 (- (* tmp2 tmp2) (* (- tmp3 tmp5)(- tmp3 tmp5))))
  (if (> tmp6 0)
    (progn
      (setq tmp7 (- tmp4 (sqrt tmp6)))
      (setq tmp8 (+ tmp4 (sqrt tmp6)))
      (setq PinCir 1)
      )
    (setq PinCir 0))
)

(defun BA2 ()
  (command "UNDO" "m")
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "EXPLODE" "all" "")
  (princ "\n Select an object in the right side of all, please.")
  (setq e2 (ssget))
  (setq e1 (ssname e2 0))
  (IsWhat e1)
  (cond ((= enType "LINE")
         (IsLne e1)
         (if (> (cadr pStt) (cadr pEnd)) (setq direct 2) (setq direct 1)))
        ((= enType "ARC")
         (IsArc e1)
         (setq pS_L1 (polar pStt (angle pStt pEnd) (/ (distance pStt pEnd) 2)))
         (setq pS_L2 (polar pCen (angle pCen pS_L1) radi_))
         (inAYoN pCen pStt pEnd pS_L2)
         (if (= InArc 0) (setq pS_L2 (polar pCen (+ (angle pCen pS_L1) pi) radi_)))
         (if (> (car pS_L1) (car pS_L2)) (setq direct 2) (setq direct 1)))
        ) ;;;cond
  (command "PEDIT" e1 "y" "j" "all" "" "")
  (setq e1 (entlast)) (command "list" e1 "")
  (setq areaS (getvar "AREA")) (command "EXPLODE" e1 "")
  (setq c_tdPt nil)
  (graphscr)
  (setq e1 (entnext))
```

```
  (setq sumXbar 0 sumYbar 0 sumELn 0)
  (setq preci_ 0.00001)
  (while e1
   (setq en_Type (cdr (assoc '0 (entget e1))))
   (cond ((= en_Type "LINE")
          (IsLne e1)
          (setq X_bar (/ (+ (car pStt) (car pEnd)) 2)
           Y_bar (/ (+ (cadr pStt) (cadr pEnd)) 2))
          (setq E_Ln (distance pStt pEnd)))
         ((= en_Type "ARC")
          (IsArc e1)
          (setq cd_Lng (distance pStt pEnd))
          (if (> St_Ang EndAng)
            (setq a_Ang (- (* 2 pi) (- St_Ang EndAng)))
            (setq a_Ang ( - EndAng St_Ang))  );;;if
          (setq E_Ln (* radi_ a_Ang))
          (setq a_Ang2 (/ a_Ang 2))
          (setq X_ba1 (/ (* radi_ (sin a_Ang2)) a_Ang2)
           cd_Ag (- (angle pStt pEnd) (* pi 0.5)))
          (setq X_bar (+ (car pCen)(* radi_ (cos cd_Ag)))
           Y_bar (+ (cadr pCen)(* radi_ (sin cd_Ag))) )
         (t nil)  ) ;cond
    (setq sumXbar (+ sumXbar (* X_bar E_Ln))
          sumYbar (+ sumYbar (* Y_bar E_Ln)) )
    (setq sumELn (+ sumELn E_Ln))
    (setq e1 (entnext e1)) ) ;while
   (setq X_Bar (/ sumXbar sumELn)
         Y_Bar (/ sumYbar sumELn))
   (setq c_tdPt (list X_Bar Y_Bar))
   (setvar "OSMODE" osVar)
   ) ;defun

(defun c:2w ()
  (ba2)
  (setvar "CMDECHO" 0)
  (setq preci_ 0.00001)
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "move" "all" "" c_tdPt "0,0")
  (setq c_tdPt (list 0.0 0.0))
  (command "zoom" "e")
  (command "zoom" "0.3XP")
  (setq det_Ag 0.5)
  (setq  angDeg 0)
  (setq a_Min nil a_Max 0.0 mxDiYoN 0 disNow 0.0)
  (princ "\n")
  (princ "  Plaese Wait. ")
  (princ "\n")
  (while (<= angDeg 180)
    (setq angRot (* (/ pi 180) angDeg))
    (setq disMem 0.0 yMax 0.0 yMin 0.0)
    (setq m 0)
    (setq e1 (entnext))
    ;;;STEP1
    (while e1
      (if e1
```

```
    (progn
      (IsWhat e1) (setq enTy_1 enType)
      (cond ((= enTy_1 "LINE") (IsLne e1) (M&RotPt pStt) (setq pS_L1 tmp4)
                               (M&RotPt pEnd) (setq pE_L1 tmp4)
                               (if (> (cadr pS_L1) (cadr pE_L1)) (setq tmp1 pS_L1 pS_L1 pE_L1
    pE_L1 tmp1))
                               (if (> (cadr pE_L1) yMax) (setq yMax (cadr pE_L1) pYmax pE_L1
    eMax e1))
                               (if (< (cadr pS_L1) yMin) (setq yMin (cadr pS_L1) pYmin pS_L1
    eMin e1)))
        ((= enTy_1 "ARC") (IsArc e1) (setq ra_A1 radi_)
                          (M&RotPt pCen) (setq pC_A1 tmp4)
                          (M&RotPt pStt) (setq pS_A1 tmp4)
                          (M&RotPt pEnd) (setq pE_A1 tmp4)
                          (setq pChk (polar pC_A1 (* 0.5 pi) ra_A1))
                          (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                          (if (= InArc 1)
                            (if (> (cadr pChk) yMax) (setq yMax (cadr pChk) pYmax pChk
    eMax e1))
                            (progn
                                (if (> (cadr pS_A1) yMax) (setq yMax (cadr pS_A1) pYmax
    pS_A1 eMax e1))
                                (if (> (cadr pE_A1) yMax) (setq yMax (cadr pE_A1) pYmax
    pE_A1 eMax e1))))
                          (setq pChk (polar pC_A1 (* 1.5 pi) ra_A1))
                          (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                          (if (= InArc 1)
                            (if (< (cadr pChk) yMin)(setq yMin (cadr pChk) pYmin pChk eMin
    e1))
                            (progn
                                (if (< (cadr pS_A1) yMin) (setq yMin (cadr pS_A1) pYmin
    pS_A1 eMin e1))
                                (if (< (cadr pE_A1) yMin) (setq yMin (cadr pE_A1) pYmin
    pE_A1 eMin e1))))
        )) ;;;cond
      (setq e1 (entnext e1))
      )  ;; porgn
    ) ;;; if e1
  )
;;;STEP2
(setq LstSuj nil e1 (entnext))
(while e1
  (if e1
    (progn
      (IsWhat e1) (setq enTy_1 enType)
      (cond ((= enTy_1 "LINE")
        (IsLne e1) (M&RotPt pStt) (setq pS_L1 tmp4)
        (M&RotPt pEnd) (setq pE_L1 tmp4)
        (if (or (and (equal pS_L1 pYmax preci_) (equal pE_L1 pYmin preci_))
                (and (equal pS_L1 pYmin preci_) (equal pE_L1 pYmax preci_)))
          (progn
            (setq pChk (list (/ (+ (car pS_L1) (car pE_L1)) 2) (/ (+ (cadr pS_L1) (cadr pE_L1))
    2)) )
            (setq tmp1 (append (list '(0)) (list pS_L1) (list pChk)))
            (setq tmp2 (append (list '(0)) (list pChk) (list pE_L1)))
```

```
    (setq tmp3 nil)
    )
   (progn
    (setq tmp1 (append (list '(0)) (list pS_L1) (list pE_L1)))
    (setq tmp2 nil tmp3 nil)
    )
   )
 )
((= enTy_1 "ARC")
 (IsArc e1) (setq ra_A1 radi_)
 (M&RotPt pCen) (setq pC_A1 tmp4)
 (M&RotPt pStt) (setq pS_A1 tmp4)
 (M&RotPt pEnd) (setq pE_A1 tmp4)
 (setq pChk (polar pC_A1 (* 0.5 pi) ra_A1))
 (if (equal pChk pYmax preci_)
    (progn
     (inAYoN pC_A1 pS_A1 pE_A1 pChk)
     (if (= InArc 1)
         (if (equal pS_A1 pChk preci_) (setq pCut1 nil)
           (if (equal pE_A1 pChk preci_) (setq pCut1 nil)
             (setq pCut1 pChk)) )
         (setq pCut1 nil) )
     )
    (setq pCut1 nil)
    ) ; if
 (setq pChk (polar pC_A1 (* 1.5 pi) ra_A1))
 (if (equal pChk pYmin preci_)
  (progn
   (inAYoN pC_A1 pS_A1 pE_A1 pChk)
   (if (= InArc 1)
     (if (equal pS_A1 pChk preci_) (setq pCut2 nil)
         (if (equal pE_A1 pChk preci_) (setq pCut2 nil)
           (setq pCut2 pChk)) )
     (setq pCut2 nil) )
   )
  (setq pCut2 nil)
  ) ;if
(if pCut1
  (if pCut2
   (if (and (> (angle pC_A1 pS_A1) (* 0.5 pi)) (< (angle pC_A1 pS_A1) (* 1.5 pi)) )
     (if (= direct 1)
         (progn
           (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut2)))
           (setq tmp2 (append (list pC_A1) (list pCut2) (list pCut1)))
           (setq tmp3 (append (list pC_A1) (list pCut1) (list pE_A1))))
         (progn
           (setq tmp1 (append (list pC_A1) (list pCut1) (list pE_A1)))
           (setq tmp2 (append (list pC_A1) (list pCut2) (list pCut1)))
           (setq tmp3 (append (list pC_A1) (list pS_A1) (list pCut2)))) ) ;if
    (if (= direct 1)
         (progn
           (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut1)))
           (setq tmp2 (append (list pC_A1) (list pCut1) (list pCut2)))
           (setq tmp3 (append (list pC_A1) (list pCut2) (list pE_A1))))
         (progn
```

```
                (setq tmp1 (append (list pC_A1) (list pCut2) (list pE_A1)))
                (setq tmp2 (append (list pC_A1) (list pCut1) (list pCut2)))
                (setq tmp3 (append (list pC_A1) (list pS_A1) (list pCut1)))) ) ;if
            )
           (if (= direct 1)
            (progn
                (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut1)))
                (setq tmp2 (append (list pC_A1) (list pCut1) (list pE_A1)))
                (setq tmp3 nil))
            (progn
                (setq tmp1 (append (list pC_A1) (list pCut1) (list pE_A1)))
                (setq tmp2 (append (list pC_A1) (list pS_A1) (list pCut1)))
                (setq tmp3 nil))
            )
           ) ;if pCut2
          (if pCut2
           (if (= direct 1)
            (progn
                (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut2)))
                (setq tmp2 (append (list pC_A1) (list pCut2) (list pE_A1)))
                (setq tmp3 nil))
            (progn
                (setq tmp1 (append (list pC_A1) (list pCut2) (list pE_A1)))
                (setq tmp2 (append (list pC_A1) (list pS_A1) (list pCut2)))
                (setq tmp3 nil))
            )
           (progn
            (setq tmp1 (append (list pC_A1) (list pS_A1) (list pE_A1)))
            (setq tmp2 nil) (setq tmp3 nil))
           ) ;if pCut2
          ) ; if pCut1
         )
        ) ;cond
       (cond ((and (= tmp3 nil) (= tmp2 nil))
        (if LstSuj (setq LstSuj (append LstSuj (list tmp1))) (setq LstSuj (list tmp1))) )
       ((= tmp3 nil)
        (if LstSuj (setq LstSuj (append LstSuj (list tmp1) (list tmp2))) (setq LstSuj (append (list
    tmp1) (list tmp2)))) )
        (t (if LstSuj (setq LstSuj (append LstSuj (list tmp1) (list tmp2) (list tmp3))) (setq LstSuj
    (append (list tmp1) (list tmp2) (list tmp3))) )  )
        )
       (setq e1 (entnext e1))
       )  ;; porgn
     ) ;;; if e1
   )
(setq n 0)
(setq nMax (length LstSuj))
(setq LstTmp nil)
(setq step 0 swit 0 flag nil)
(while (< n nMax)
  (setq tmp0 (nth n LstSuj))
  (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
  (if (or (and (equal tmp2 pYmax preci_) (equal tmp3 pYmin preci_))
          (and (equal tmp2 pYmin preci_) (equal tmp3 pYmax preci_)) )
    (progn
```

```
    (setq pC_A1 tmp1 pS_A1 tmp2 pE_A1 tmp3)
    (setq ra_A1 (distance tmp1 tmp2))
    (setq pChk (polar pS_A1 (angle pS_A1 pE_A1) (/ (distance pS_A1 pE_A1) 2)))
    (setq pM_A1 (polar pC_A1 (angle pC_A1 pChk) ra_A1))
    (if (= direct 1)
      (progn
        (setq tmLst1 (append (list pC_A1) (list pS_A1) (list pM_A1)))
        (setq tmLst2 (append (list pC_A1) (list pM_A1) (list pE_A1))) )
      (progn
        (setq tmLst1 (append (list pC_A1) (list pM_A1) (list pE_A1)))
        (setq tmLst2 (append (list pC_A1) (list pS_A1) (list pM_A1))) )
      )
    )
   (setq tmLst1 tmp0 tmLst2 nil)
   )
  (if LstTmp
    (if (= tmLst2 nil) (setq LstTmp (append LstTmp (list tmLst1))) (setq LstTmp (append
    LstTmp (list tmLst1) (list tmLst2))))
    (if (= tmLst2 nil) (setq LstTmp (list tmLst1)) (setq LstTmp (append (list tmLst1) (list
    tmLst2)))) )
  (setq n (+ n 1))
  ) ;while
(setq lstMem LstTmp)
(setq LstSuj LstTmp)
(setq n 0)
(setq nMax (length LstSuj))
(setq LstTmp nil LstLft nil LstRgh nil LstL_V nil LstR_V nil)
(setq step 0 swit 0 flag nil)
(while (< n nMax)
  (setq tmp0 (nth n LstSuj))
  (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
  (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_)
          (equal tmp2 pYmin preci_) (equal tmp3 pYmin preci_))
    (progn
      (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_)) (setq MxOrMn 1) (setq
    MxOrMn 0) )
      (setq tmp4 (nth (+ n 1) LstSuj))
      (if tmp4
        (progn
          (setq tmp5 (cadr tmp4) tmp6 (caddr tmp4))
          (if (or (equal tmp5 pYmax preci_) (equal tmp6 pYmax preci_)
                  (equal tmp5 pYmin preci_) (equal tmp6 pYmin preci_))
       (if (or (equal tmp5 pYmax preci_) (equal tmp6 pYmax preci_))
          (if (= MxOrMn 1)
            (if (= swit 2) (setq swit 1) (setq swit 2))
            (setq swit 1))
          (if (= MxOrMn 0)
            (if (= swit 2) (setq swit 1) (setq swit 2))
            (setq swit 1))
          )
       (setq swit 1)
       )
          ) ;progn
        (if (= swit 2) (setq swit 1) (setq swit 2))
        )
```

```
      (if (= swit 1)
       (progn
         (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_))
       ;true
       (if (= direct 1)
         (progn
           (if (= step 3) (setq flag 31))
           (setq step 1))
         (progn
           (if (= step 4) (setq flag 42))
           (setq step 2))
         ) ;if
       (if (= direct 1)
         (progn
           (if (= step 1) (setq flag 13))
           (setq step 3))
         (progn
           (if (= step 2) (setq flag 24))
           (setq step 4))
         ) ;if
       ) ; (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_))
          ) ;progn
        ) ; (if (= swit 1)
      ) ;progn
    ) ;if
  (cond ((= step 0) (if LstTmp (setq LstTmp (append LstTmp (list tmp0))) (setq LstTmp (list
    tmp0))) )
       ((= step 1) (if LstLft (setq LstLft (append LstLft (list tmp0))) (setq LstLft (list tmp0))) )
       ((= step 2) (if LstRgh (setq LstRgh (append LstRgh (list tmp0))) (setq LstRgh (list
    tmp0))) )
       ((= step 3) (if LstR_V (setq LstR_V (append LstR_V (list tmp0))) (setq LstR_V (list
    tmp0))) )
       ((= step 4) (if LstL_V (setq LstL_V (append LstL_V (list tmp0))) (setq LstL_V (list
    tmp0))) ) )
  (setq n (+ n 1))
  ) ;while
(cond ((= flag 31)(if (and LstLft LstTmp) (setq LstLft (append LstLft LstTmp))))
      ((= flag 24)(if (and LstL_V LstTmp) (setq LstL_V (append LstL_V LstTmp))))
      ((= flag 13)(if (and LstR_V LstTmp) (setq LstR_V (append LstR_V LstTmp))))
      ((= flag 42)(if (and LstRgh LstTmp) (setq LstRgh (append LstRgh LstTmp)))) )
(if (and LstR_V (= LstRgh nil)) (setq LstRgh (reverse LstR_V)))
(if (and LstL_V (= LstLft nil)) (setq LstLft (reverse LstL_V)))
(if (and LstRgh (= LstR_V nil)) (setq LstR_V (reverse LstRgh)))
(if (and LstLft (= LstL_V nil)) (setq LstL_V (reverse LstLft)))
(setq n 0)
(setq nMax (length LstRgh))
(setq disMem 0.0)
(while (< n nMax)
  (setq tmp0 (nth n LstRgh))
  (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
  (if (= (cadr tmp1) nil) (setq enTy_1 "LINE") (setq enTy_1 "ARC") )
  (cond ((= enTy_1 "LINE") (if (> (cadr tmp2) (cadr tmp3)) (setq pS_L1 tmp3 pE_L1 tmp2)
    (setq pS_L1 tmp2 pE_L1 tmp3)) )
       ((= enTy_1 "ARC") (setq ra_A1 (distance tmp1 tmp2)) (setq pC_A1 tmp1) (setq pS_A1
    tmp2) (setq pE_A1 tmp3))
```

```
    ) ;cond
(setq n2 0)
(while(< n2 nMax)
  (setq tmp0 (nth n2 LstR_V))
  (setq tmp1 (car tmp0))
  (if (= (cadr tmp1) nil) (setq enTy_2 "LINE") (setq enTy_2 "ARC") )
  (cond ((= enTy_2 "LINE")
        (Ro&M_2 (cadr tmp0)) (setq tmp1 tmp6)
        (Ro&M_2 (caddr tmp0)) (setq tmp2 tmp6)
        (if (> (cadr tmp1) (cadr tmp2)) (setq pS_L2 tmp2 pE_L2 tmp1) (setq pS_L2 tmp1
  pE_L2 tmp2)))
       ((= enTy_2 "ARC")
        (Ro&M_2 (car tmp0)) (setq pC_A2 tmp6)
        (Ro&M_2 (cadr tmp0)) (setq pS_A2 tmp6)
        (Ro&M_2 (caddr tmp0)) (setq pE_A2 tmp6)
        (setq ra_A2 (distance pC_A2 pS_A2)) )  ) ;cond
  (cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE"))
        (if (or (equal (cadr pE_L1) (cadr pS_L2) preci_) (equal (cadr pE_L2) (cadr pS_L1)
  preci_))
     (setq xxx 1)
     (if (or (and (>= (cadr pE_L1) (cadr pS_L2)) (<= (cadr pS_L1) (cadr pE_L2)))
              (and (>= (cadr pE_L2) (cadr pS_L1)) (<= (cadr pS_L2) (cadr pE_L1))))
       (progn
         (inLYoN pS_L1 pE_L1 pS_L2)
         (inLYoN pS_L1 pE_L1 pE_L2)
         (inLYoN pS_L2 pE_L2 pS_L1)
         (inLYoN pS_L2 pE_L2 pE_L1) ) ;progn
       )
     )
        )
       ((or (and (= enTy_1 "LINE") (= enTy_2 "ARC"))
       (and (= enTy_1 "ARC") (= enTy_2 "LINE")))
        (if (= enTy_1 "ARC") (setq pS_L1 pS_L2 pE_L1 pE_L2
                                   pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1))
        (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
        (setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
        (XinCir pC_A2 ra_A2 (cadr pS_L1))
        (if (= PinCir 1)
     (progn
       (setq pChk (list tmp7 (cadr pS_L1)))
       (inAYoN pC_A2 pS_A2 pE_A2 pChk)
       (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
       (setq pChk (list tmp8 (cadr pS_L1)))
       (inAYoN pC_A2 pS_A2 pE_A2 pChk)
       (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
       ))
        (XinCir pC_A2 ra_A2 (cadr pE_L1))
        (if (= PinCir 1)
     (progn
       (setq pChk (list tmp7 (cadr pE_L1)))
       (inAYoN pC_A2 pS_A2 pE_A2 pChk)
```

```
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  (setq pChk (list tmp8 (cadr pE_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  ))
   (inLYoN pS_L1 pE_L1 pS_A2)
   (inLYoN pS_L1 pE_L1 pE_A2)
   )
  ((and (= enTy_1 "ARC") (= enTy_2 "ARC"))
   (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
   (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
   (if (<= C_CinY (+ ra_A1 ra_A2))
(progn
  (setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
  (if (> (cadr pC_A1) (cadr pC_A2))
    (setq Y_Bar (- (cadr pC_A1) tmp2))
    (setq Y_Bar (+ (cadr pC_A1) tmp2)))
  (XinCir pC_A1 ra_A1 Y_Bar)
  (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
  (XinCir pC_A2 ra_A2 Y_Bar)
  (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
  (if (= InArc 1)
    (progn
      (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
      (if (= InArc 1)
          (progn
            (setq disNow (abs (- (car pCut1)(car pCut4))))
            (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
            (setq mxDiYoN 1)
            ))
      ))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
  (if (= InArc 1)
    (progn
      (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
      (if (= InArc 1)
          (progn
            (setq disNow (abs (- (car pCut2)(car pCut3))))
            (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
            (setq mxDiYoN 1)
            ))
      ))
  ) ;;; progn
)  ;;; if
   (if (= mxDiYoN 0)
(progn
  (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
  (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
  (setq tmp1 (abs (- ra_A2 ra_A1)))
  (if (<= C_CinY tmp1)
    (progn
      (setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2)(cadr pC_A1)))) tmp1))
      (if (<= ra_A1 ra_A2)
```

```
          (if(< (cadr pC_A1)(cadr pC_A2))(setq Y_Bar(-(cadr pC_A1)tmp2))(setq
Y_Bar(+(cadr pC_A1)tmp2)))
          (if(< (cadr pC_A2)(cadr pC_A1))(setq Y_Bar(-(cadr pC_A1)tmp2))(setq
Y_Bar(+(cadr pC_A1)tmp2)))
          )
       (XinCir pC_A1 ra_A1 Y_Bar)
       (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
       (XinCir pC_A2 ra_A2 Y_Bar)
       (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
       (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
       (if (= InArc 1)
           (progn
            (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
            (if (= InArc 1)
              (progn
               (setq disNow (abs (- (car pCut1)(car pCut3))))
               (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
               (setq mxDiYoN 1)
               ))
            ))
       (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
       (if (= InArc 1)
           (progn
            (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
            (if (= InArc 1)
              (progn
               (setq disNow (abs (- (car pCut2)(car pCut4))))
               (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
               (setq mxDiYoN 1)
               ))
            ))
      ))
    ))
     (if (= mxDiYoN 0)
   (progn
    (XinCir pC_A2 ra_A2 (cadr pS_A1))
    (if (= PinCir 1)
     (progn
      (setq pChk (list tmp7 (cadr pS_A1)))
      (inAYoN pC_A2 pS_A2 pE_A2 pChk)
      (if (= InArc 1)
          (progn
           (setq disNow (abs (- tmp7 (car pS_A1))))
           (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
           ))
      (setq pChk (list tmp8 (cadr pS_A1)))
      (inAYoN pC_A2 pS_A2 pE_A2 pChk)
      (if (= InArc 1)
          (progn
           (setq disNow (abs (- tmp8 (car pS_A1))))
           (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
           ))
      ))
    (XinCir pC_A2 ra_A2 (cadr pE_A1))
    (if (= PinCir 1)
```

```
    (progn
     (setq pChk (list tmp7 (cadr pE_A1)))
     (inAYoN pC_A2 pS_A2 pE_A2 pChk)
     (if (= InArc 1)
         (progn
          (setq disNow (abs (- tmp7 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
     (setq pChk (list tmp8 (cadr pE_A1)))
     (inAYoN pC_A2 pS_A2 pE_A2 pChk)
     (if (= InArc 1)
         (progn
          (setq disNow (abs (- tmp8 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    ))
  (XinCir pC_A1 ra_A1 (cadr pS_A2))
  (if (= PinCir 1)
   (progn
    (setq pChk (list tmp7 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
         (setq disNow (abs (- tmp7 (car pS_A2))))
         (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
         ))
    (setq pChk (list tmp8 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
         (setq disNow (abs (- tmp8 (car pS_A2))))
         (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
         ))
    ))
  (XinCir pC_A1 ra_A1 (cadr pE_A2))
  (if (= PinCir 1)
   (progn
    (setq pChk (list tmp7 (cadr pE_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
         (setq disNow (abs (- tmp7 (car pE_A2))))
         (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
         ))
    (setq pChk (list tmp8 (cadr pE_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
         (setq disNow (abs (- tmp8 (car pE_A2))))
         (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
         ))
    ))
  ) ;progn
 (setq mxDiYoN 0)
 ) ; (if (= mxDiYoN 0)
```

```
        )
       )
    (setq n2 (+ n2 1))
    ) ;while
  (setq n (+ n 1))
  )
(setq d_toShf disMem)
(setq n 0)
(setq nMax (length LstLft))
(setq disMem 0.0)
(while (< n nMax)
  (setq tmp0 (nth n LstLft))
  (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
  (if (= (cadr tmp1) nil) (setq enTy_1 "LINE") (setq enTy_1 "ARC") )
  (cond ((= enTy_1 "LINE") (if (> (cadr tmp2) (cadr tmp3)) (setq pS_L1 tmp3 pE_L1 tmp2)
    (setq pS_L1 tmp2 pE_L1 tmp3)) )
       ((= enTy_1 "ARC") (setq ra_A1 (distance tmp1 tmp2)) (setq pC_A1 tmp1) (setq pS_A1
    tmp2) (setq pE_A1 tmp3))
       ) ;cond
  (setq n2 0)
  (while(< n2 nMax)
    (setq tmp0 (nth n2 LstL_V))
    (setq tmp1 (car tmp0))
    (if (= (cadr tmp1) nil) (setq enTy_2 "LINE") (setq enTy_2 "ARC") )
    (cond ((= enTy_2 "LINE")
          (Ro&M_3 (cadr tmp0) d_toShf) (setq tmp1 tmp8)
          (Ro&M_3 (caddr tmp0) d_toShf) (setq tmp2 tmp8)
          (if (> (cadr tmp1) (cadr tmp2)) (setq pS_L2 tmp2 pE_L2 tmp1) (setq pS_L2 tmp1
    pE_L2 tmp2)))
         ((= enTy_2 "ARC")
          (Ro&M_3 (car tmp0) d_toShf) (setq pC_A2 tmp8)
          (Ro&M_3 (cadr tmp0) d_toShf) (setq pS_A2 tmp8)
          (Ro&M_3 (caddr tmp0) d_toShf) (setq pE_A2 tmp8)
          (setq ra_A2 (distance pC_A2 pS_A2)) )  ) ;cond
    (cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE"))
          (if (or (equal (cadr pE_L1) (cadr pS_L2) preci_) (equal (cadr pE_L2) (cadr pS_L1)
    preci_))
       (setq xxx 1)
       (if (or (and (>= (cadr pE_L1) (cadr pS_L2)) (<= (cadr pS_L1) (cadr pE_L2)))
               (and (>= (cadr pE_L2) (cadr pS_L1)) (<= (cadr pS_L2) (cadr pE_L1))))
         (progn
          (inLYoN pS_L1 pE_L1 pS_L2)
          (inLYoN pS_L1 pE_L1 pE_L2)
          (inLYoN pS_L2 pE_L2 pS_L1)
          (inLYoN pS_L2 pE_L2 pE_L1) ) ;progn
         )
       )
          )
         ((or (and (= enTy_1 "LINE") (= enTy_2 "ARC"))
         (and (= enTy_1 "ARC") (= enTy_2 "LINE")))
          (if (= enTy_1 "ARC") (setq pS_L1 pS_L2 pE_L1 pE_L2
                                     pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1))
          (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2))
          (inAYoN pC_A2 pS_A2 pE_A2 pChk)
          (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
```

```
   (setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
   (inAYoN pC_A2 pS_A2 pE_A2 pChk)
   (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
   (XinCir pC_A2 ra_A2 (cadr pS_L1))
   (if (= PinCir 1)
 (progn
  (setq pChk (list tmp7 (cadr pS_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  (setq pChk (list tmp8 (cadr pS_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  ))
   (XinCir pC_A2 ra_A2 (cadr pE_L1))
   (if (= PinCir 1)
 (progn
  (setq pChk (list tmp7 (cadr pE_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  (setq pChk (list tmp8 (cadr pE_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  ))
   (inLYoN pS_L1 pE_L1 pS_A2)
   (inLYoN pS_L1 pE_L1 pE_A2)
   )
  ((and (= enTy_1 "ARC") (= enTy_2 "ARC"))
   (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
   (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
   (if (<= C_CinY (+ ra_A1 ra_A2))
 (progn
  (setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
  (if (> (cadr pC_A1) (cadr pC_A2))
    (setq Y_Bar (- (cadr pC_A1) tmp2))
    (setq Y_Bar (+ (cadr pC_A1) tmp2)))
  (XinCir pC_A1 ra_A1 Y_Bar)
  (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
  (XinCir pC_A2 ra_A2 Y_Bar)
  (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
  (if (= InArc 1)
    (progn
      (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
      (if (= InArc 1)
          (progn
            (setq disNow (abs (- (car pCut1)(car pCut4))))
            (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
            (setq mxDiYoN 1)
            ))
     ))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
  (if (= InArc 1)
    (progn
      (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
      (if (= InArc 1)
```

```
            (progn
              (setq disNow (abs (- (car pCut2)(car pCut3))))
              (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
              (setq mxDiYoN 1)
              ))
          ))
      ) ;;; progn
    )  ;;; if
      (if (= mxDiYoN 0)
    (progn
      (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
      (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
      (setq tmp1 (abs (- ra_A2 ra_A1)))
      (if (<= C_CinY tmp1)
       (progn
        (setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2)(cadr pC_A1)))) tmp1))
        (if (<= ra_A1 ra_A2)
            (if(< (cadr pC_A1)(cadr pC_A2))(setq Y_Bar(-(cadr pC_A1)tmp2))(setq
Y_Bar(+(cadr pC_A1)tmp2)))
            (if(< (cadr pC_A2)(cadr pC_A1))(setq Y_Bar(-(cadr pC_A1)tmp2))(setq
Y_Bar(+(cadr pC_A1)tmp2)))
            )
        (XinCir pC_A1 ra_A1 Y_Bar)
        (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
        (XinCir pC_A2 ra_A2 Y_Bar)
        (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
        (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
        (if (= InArc 1)
            (progn
              (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
              (if (= InArc 1)
                (progn
                  (setq disNow (abs (- (car pCut1)(car pCut3))))
                  (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
                  (setq mxDiYoN 1)
                  ))
              ))
        (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
        (if (= InArc 1)
            (progn
              (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
              (if (= InArc 1)
                (progn
                  (setq disNow (abs (- (car pCut2)(car pCut4))))
                  (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
                  (setq mxDiYoN 1)
                  ))
              ))
        ))
      ))
       (if (= mxDiYoN 0)
     (progn
      (XinCir pC_A2 ra_A2 (cadr pS_A1))
      (if (= PinCir 1)
       (progn
```

```
    (setq pChk (list tmp7 (cadr pS_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pS_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pS_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp8 (car pS_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    ))
(XinCir pC_A2 ra_A2 (cadr pE_A1))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pE_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp8 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    ))
(XinCir pC_A1 ra_A1 (cadr pS_A2))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pS_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp8 (car pS_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    ))
(XinCir pC_A1 ra_A1 (cadr pE_A2))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_A2)))
```

```
            (inAYoN pC_A1 pS_A1 pE_A1 pChk)
            (if (= InArc 1)
                (progn
                  (setq disNow (abs (- tmp7 (car pE_A2))))
                  (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
                  ))
            (setq pChk (list tmp8 (cadr pE_A2)))
            (inAYoN pC_A1 pS_A1 pE_A1 pChk)
            (if (= InArc 1)
                (progn
                  (setq disNow (abs (- tmp8 (car pE_A2))))
                  (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
                  ))
            ))
          ) ;progn
        (setq mxDiYoN 0)
        ) ; (if (= mxDiYoN 0)
          ; ) ;;progn
          ; ) ;;if
          )
         )
      (setq n2 (+ n2 1))
      (if (> nMax 4)
        (progn
          (setq XX (rem n2 4))
          (cond ((= XX 0) (princ "\n ") (princ "\n ") (princ "\n Please wait.  / ") (princ "\n ")
      (princ "\n "))
             ((= XX 1) (princ "\n ") (princ "\n ") (princ "\n Please wait.  - ") (princ "\n ") (princ "\n
      "))
             ((= XX 2) (princ "\n ") (princ "\n ") (princ "\n Please wait.  \\\ ") (princ "\n ") (princ
      "\n "))
             ((= XX 3) (princ "\n ") (princ "\n ") (princ "\n Please wait.  | ") (princ "\n ") (princ "\n
      ")))
          )
        )
      ) ;while n2
    (setq n (+ n 1))
    (princ "\n")
    ) ;while n
  (setq d_Area disMem)
  (setq hgh (abs (- yMax yMin)))
  (setq areaN_ (* d_Area hgh))
  (if (or (< areaN_ a_Min)(= a_Min nil)) (setq a_Min areaN_ agM_Min angRot
                                          dsM_Min disMem pM_Min pMem0 yMaxMm yMax
      yMinMm yMin
                                          d_AreaMm d_Area d_ShfMm d_toShf))
  (setq angDeg (+ angDeg det_Ag))
  ) ; while  (<= angRot
  (princ "\n ") (princ "\n ")(princ "\n Please wait.  * ")(princ "\n ") (princ "\n ")
(setq angMeD (/ (* agM_Min 180) pi))
(setq bstAg1 angMeD)
(setvar "OSMODE" osVar)
(princ "\n") (princ "\n")
(princ "\n") (princ "  Press any key to continue,please.")
(princ "\n") (princ "\n")
```

```
 (read-char)
 (tes2w))

(defun tes2w ()
 (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
 (setq gap (getreal "\n GAP <0> = "))
 (if (= gap nil) (setq gap 0))
 (setq agM_MinD (/ (* agM_Min 180) pi))
 (setq dsMarry (* (- 1)(+ (* 2 gap) d_AreaMm)))
 (setq tmp3 (+ yMaxMm yMinMm))
 (setq tmp2 (polar '(0 0) (* pi 0.5) tmp3))
 (setq tmp4 (+ gap d_ShfMm))
 (setq tmp1 (polar tmp2 0 tmp4))
 (graphscr)
 (command "EXPLODE" "all" "")
 (command "PEDIT" "l" "y" "j" "all" "" "")
 (command "ROTATE" "all" "" "0,0" agM_MinD)
 (command "COPY" "p" "" "0,0" "0,0")
 (command "ROTATE" "p" "" "0,0" "180")
 (command "MOVE" "p" "" "0,0" tmp1)
 (command "ARRAY" "all" "" "R" "1" "3" dsMarry)
 (setvar "OSMODE" osVar)
 (princ "\n") (princ "\n")
 (princ "\n") (princ "  Press any key to see data.")
 (princ "\n")
 (read-char)
 (read-char)
 (setq tmp1 (* (/ (* areaS 2) a_Min) 100))
 (setq tmp2 (- yMaxMm yMinMm))
 (textscr)
 (princ "\n") (princ "*******************************************")
 (princ "\n") (princ "*******************************************")
 (princ "\n") (princ "                                           ")
 (princ "\n") (princ "   This is the data for 2 way strip layout.")
 (princ "\n") (princ "                                           ")
 (princ "\n ") (princ "     The best angle = ") (princ agM_MinD) (princ " degree.")
 (princ "\n ") (princ "  Distance for feed = ") (princ dsM_Min)  (princ " mm. <no gap>")
 (princ "\n ") (princ "        Strip width = ") (princ tmp2) (princ " mm. <no gap>")
 (princ "\n ") (princ "      The best Area = ") (princ a_Min) (princ ".<2 workpiece>")
 (princ "\n ") (princ "     The efficiency = ") (princ tmp1) (princ " % <no gap>")
 (princ "\n") (princ "                                           ")
 (princ "\n") (princ "*******************************************")
 (princ "\n") (princ "**    If the function is not correct or    **")
 (princ "\n") (princ "**    have some problem, please tell me.   **")
 (princ "\n") (princ "**                                         **")
 (princ "\n") (princ "**                    S.Singhapant     **")
 (princ "\n") (princ "**          Tel  074322901  073917164    **")
 (princ "\n") (princ "*******************************************")

 (princ "\n") (princ "\n")
 (princ "\n") (princ "  Press any key to continue,please.")
 (princ "\n") (princ "\n")
 (read-char)
 (graphscr))
```

```
(defun ba1 ()
  (command "UNDO" "m")
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "EXPLODE" "all" "")
  (setq e1 (entnext)) (command "PEDIT" e1 "y" "j" "all" "" "")
  (setq e1 (entlast)) (command "list" e1 "")
  (setq areaS (getvar "AREA")) (command "EXPLODE" e1 "")
  (setq c_tdPt nil)
  (graphscr)
  (setq e1 (entnext))
  (setq sumXbar 0 sumYbar 0 sumELn 0)
  (while e1
    (setq en_Type (cdr (assoc '0 (entget e1))))
    (cond ((= en_Type "LINE")
           (IsLne e1)
           (setq X_bar (/ (+ (car pStt) (car pEnd)) 2)
            Y_bar (/ (+ (cadr pStt) (cadr pEnd)) 2))
           (setq E_Ln (distance pStt pEnd)))
          ((= en_Type "ARC")
           (IsArc e1)
           (setq cd_Lng (distance pStt pEnd))
           (if (> St_Ang EndAng)
             (setq a_Ang (- (* 2 pi) (- St_Ang EndAng)))
             (setq a_Ang ( - EndAng St_Ang))  );;;if
           (setq E_Ln (* radi_ a_Ang))
           (setq a_Ang2 (/ a_Ang 2))
           (setq X_ba1 (/ (* radi_ (sin a_Ang2)) a_Ang2)
            cd_Ag (- (angle pStt pEnd) (* pi 0.5)))
           (setq X_bar (+ (car pCen)(* radi_ (cos cd_Ag)))
            Y_bar (+ (cadr pCen)(* radi_ (sin cd_Ag)))) )
          (t nil)  ) ;cond
    (setq sumXbar (+ sumXbar (* X_bar E_Ln))
          sumYbar (+ sumYbar (* Y_bar E_Ln)) )
    (setq sumELn (+ sumELn E_Ln))
    (setq e1 (entnext e1))  ) ;while
  (setq X_Bar (/ sumXbar sumELn)
        Y_Bar (/ sumYbar sumELn) )
  (setq c_tdPt (list X_Bar Y_Bar))
  (setvar "OSMODE" osVar))

(defun c:1w ()
  (ba1)
  (setq preci_ 0.00001)
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "move" "all" "" c_tdPt "0,0")
  (setq c_tdPt (list 0.0 0.0))
  (command "zoom" "e")
  (command "zoom" "0.3XP")
  (setq det_Ag 1.0)
  (setq  angDeg 0)
  (setq a_Min nil a_Max 0.0 mxDiYoN 0 disNow 0.0)
  (setq Lstang nil)
  (setq n 1)
  (while (<= angDeg 180)
    (setq angRot (* (/ pi 180) angDeg))
```

```
(setq disMem 0.0 yMax 0.0 yMin 0.0)
(setq e1 (entnext))
(while e1
 (if e1
   (progn
    (setq e2 e1)
    (IsWhat e1) (setq enTy_1 enType)
    (cond ((= enTy_1 "LINE") (progn
                          (IsLne e1)(M&RotPt pStt)(setq pS_L1 tmp4)
                                (M&RotPt pEnd)(setq pE_L1 tmp4)
                          (if (> (cadr pS_L1)(cadr pE_L1))(setq tmp1 pS_L1 pS_L1 pE_L1
  pE_L1 tmp1))
                          (if (> (cadr pE_L1) yMax)(setq yMax(cadr pE_L1)))
                          (if (< (cadr pS_L1) yMin)(setq yMin(cadr pS_L1)))))
     ((= enTy_1 "ARC") (progn (IsArc e1)(setq ra_A1 radi_)
                         (M&RotPt pCen)(setq pC_A1 tmp4)
                         (M&RotPt pStt)(setq pS_A1 tmp4)
                         (M&RotPt pEnd)(setq pE_A1 tmp4)
                         (setq pChk (polar pC_A1 (* 0.5 pi) ra_A1))
                         (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                         (if (= InArc 1)
                          (if (> (cadr pChk) yMax)(setq yMax(cadr pChk)))
                          (progn
                              (if (> (cadr pS_A1) yMax)(setq yMax(cadr pS_A1)))
                              (if (> (cadr pE_A1) yMax)(setq yMax(cadr pE_A1)))))
                         (setq pChk (polar pC_A1 (* 1.5 pi) ra_A1))
                         (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                         (if (= InArc 1)
                          (if (< (cadr pChk) yMin)(setq yMin(cadr pChk)))
                          (progn
                              (if (< (cadr pS_A1) yMin)(setq yMin(cadr pS_A1)))
                              (if (< (cadr pE_A1) yMin)(setq yMin(cadr pE_A1)))))
                        (setq pChk (polar pC_A1 (* 0.0 pi) ra_A1))
                        (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                        (if (= InArc 1)
                         (progn
                             (setq pChk (polar pC_A1 pi ra_A1))
                             (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                             (if (= InArc 1)
                              (progn
                                (setq disNow (* ra_A1 2))
                                (if (> disNow disMem) (setq disMem disNow pMem0
  pChk))))))))
     ((= enTy_1 "CIRCLE") (setq e2 (entlast) )) ) ;;;cond
    (while e2
     (setq e2 (entnext e2))
     (if e2
       (progn
     (IsWhat e2) (setq enTy_2 enType)
     (cond ((= enTy_2 "LINE")
           (progn
               (IsLne e2)(M&RotPt pStt)(setq pS_L2 tmp4)
               (M&RotPt pEnd)(setq pE_L2 tmp4)
               (if (> (cadr pS_L2) (cadr pE_L2))(setq tmp1 pS_L2 pS_L2 pE_L2 pE_L2 tmp1))
               (if (> (cadr pE_L2) yMax) (setq yMax (cadr pE_L2)))
```

```
        (if (< (cadr pS_L2) yMin) (setq yMin (cadr pS_L2)))
        ))
      ((= enTy_2 "ARC")
       (progn (IsArc e2)(setq ra_A2 radi_)
        (M&RotPt pCen)(setq pC_A2 tmp4)
        (M&RotPt pStt)(setq pS_A2 tmp4)
        (M&RotPt pEnd)(setq pE_A2 tmp4)
        (setq pChk (polar pC_A2 (* 0.5 pi) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)
         (if (> (cadr pChk) yMax)(setq yMax(cadr pChk)))
         (progn
          (if (> (cadr pS_A2) yMax)(setq yMax(cadr pS_A2)))
          (if (> (cadr pE_A2) yMax)(setq yMax(cadr pE_A2)))))
        (setq pChk (polar pC_A2 (* 1.5 pi) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)
         (if (< (cadr pChk) yMin)(setq yMin(cadr pChk)))
         (progn
          (if (< (cadr pS_A2) yMin)(setq yMin(cadr pS_A2)))
          (if (< (cadr pE_A2) yMin)(setq yMin(cadr pE_A2)))))
        (setq pChk (polar pC_A2 0 ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)
         (progn
          (setq pChk (polar pC_A2 pi ra_A2))
          (inAYoN pC_A2 pS_A2 pE_A2 pChk)
          (if (= InArc 1)
           (progn
            (setq disNow (* ra_A2 2))
            (if (> disNow disMem) (setq disMem disNow pMem0 pChk)))
          )))))
     ((= enTy_2 "CIRCLE") (setq XXX nil)) ) ;;; cond
(cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE"))
     (if (or (and (>= (cadr pE_L1) (cadr pS_L2))(<= (cadr pS_L1) (cadr pE_L2)))
             (and (>= (cadr pE_L2) (cadr pS_L1))(<= (cadr pS_L2) (cadr pE_L1))))
         (progn
          (inLYoN pS_L1 pE_L1 pS_L2)
          (inLYoN pS_L1 pE_L1 pE_L2)
          (inLYoN pS_L2 pE_L2 pS_L1)
          (inLYoN pS_L2 pE_L2 pE_L1) ) ;progn
         ) )
    ((or (and (= enTy_1 "LINE") (= enTy_2 "ARC"))
         (and (= enTy_1 "ARC") (= enTy_2 "LINE")))
     (progn
        (if (= enTy_1 "ARC")
         (setq pS_L1 pS_L2 pE_L1 pE_L2
               pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1))
        (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
        (setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
        (XinCir pC_A2 ra_A2 (cadr pS_L1))
```

```
      (if (= PinCir 1)
       (progn
         (setq pChk (list tmp7 (cadr pS_L1)))
         (inAYoN pC_A2 pS_A2 pE_A2 pChk)
         (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
         (setq pChk (list tmp8 (cadr pS_L1)))
         (inAYoN pC_A2 pS_A2 pE_A2 pChk)
         (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
         ))
      (XinCir pC_A2 ra_A2 (cadr pE_L1))
      (if (= PinCir 1)
       (progn
         (setq pChk (list tmp7 (cadr pE_L1)))
         (inAYoN pC_A2 pS_A2 pE_A2 pChk)
         (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
         (setq pChk (list tmp8 (cadr pE_L1)))
         (inAYoN pC_A2 pS_A2 pE_A2 pChk)
         (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
         ))
      (inLYoN pS_L1 pE_L1 pS_A2)
      (inLYoN pS_L1 pE_L1 pE_A2)
      )
 )
((and (= enTy_1 "ARC") (= enTy_2 "ARC"))
 (progn
      (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
      (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
      (if (<= C_CinY (+ ra_A1 ra_A2))
       (progn
         (setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
         (if (> (cadr pC_A1) (cadr pC_A2))
           (setq Y_Bar (- (cadr pC_A1) tmp2))
           (setq Y_Bar (+ (cadr pC_A1) tmp2)))
         (XinCir pC_A1 ra_A1 Y_Bar)
         (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
         (XinCir pC_A2 ra_A2 Y_Bar)
         (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
         (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
         (if (= InArc 1)
           (progn
               (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
               (if (= InArc 1)
                 (progn
                   (setq disNow (abs (- (car pCut1)(car pCut4))))
                   (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
                   (setq mxDiYoN 1)
                   ))
               ))
         (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
         (if (= InArc 1)
           (progn
               (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
               (if (= InArc 1)
                 (progn
                   (setq disNow (abs (- (car pCut2)(car pCut3))))
```

```
                    (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
                    (setq mxDiYoN 1)
                    ))
                ))
            ) ;;; progn
          ) ;;; if
        (if (= mxDiYoN 0)
          (progn
            (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
            (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
            (setq tmp1 (abs (- ra_A2 ra_A1)))
            (if (<= C_CinY tmp1)
              (progn
                (setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2)(cadr pC_A1)))) tmp1))
                (if (<= ra_A1 ra_A2)
                  (if(< (cadr pC_A1)(cadr pC_A2))(setq Y_Bar(-(cadr
pC_A1)tmp2))(setq Y_Bar(+(cadr pC_A1)tmp2)))
                  (if(< (cadr pC_A2)(cadr pC_A1))(setq Y_Bar(-(cadr
pC_A1)tmp2))(setq Y_Bar(+(cadr pC_A1)tmp2)))
                  )
                (XinCir pC_A1 ra_A1 Y_Bar)
                (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
                (XinCir pC_A2 ra_A2 Y_Bar)
                (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
                (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
                (if (= InArc 1)
                  (progn
                    (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
                    (if (= InArc 1)
                      (progn
                        (setq disNow (abs (- (car pCut1)(car pCut3))))
                        (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
                        (setq mxDiYoN 1)
                        ))
                    ))
                (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
                (if (= InArc 1)
                  (progn
                    (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
                    (if (= InArc 1)
                      (progn
                        (setq disNow (abs (- (car pCut2)(car pCut4))))
                        (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
                        (setq mxDiYoN 1)
                        ))
                    ))
                ))
            ))
          (if (= mxDiYoN 0)
            (progn
              (XinCir pC_A2 ra_A2 (cadr pS_A1))
              (if (= PinCir 1)
                (progn
                  (setq pChk (list tmp7 (cadr pS_A1)))
                  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
```

```
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp7 (car pS_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pS_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp8 (car pS_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
   ))
 (XinCir pC_A2 ra_A2 (cadr pE_A1))
 (if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp7 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pE_A1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp8 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
   ))
 (XinCir pC_A1 ra_A1 (cadr pS_A2))
 (if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp7 (car pS_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
    (setq pChk (list tmp8 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
          (setq disNow (abs (- tmp8 (car pS_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
          ))
   ))
 (XinCir pC_A1 ra_A1 (cadr pE_A2))
 (if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
```

```
                                (progn
                                    (setq disNow (abs (- tmp7 (car pE_A2))))
                                    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
                                    ))
                                (setq pChk (list tmp8 (cadr pE_A2)))
                                (inAYoN pC_A1 pS_A1 pE_A1 pChk)
                                (if (= InArc 1)
                                  (progn
                                    (setq disNow (abs (- tmp8 (car pE_A2))))
                                    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
                                    ))
                                ))
                            ) ;progn
                        (setq mxDiYoN 0)
                        ) ; (if (= mxDiYoN 0)
                    ) ;progn
                )
               ) ;cond
             (if (> n 50)
              (progn
               (setq XX (rem n 4))
               (cond ((= XX 0) (princ "\n ") (princ "\n ") (princ "\n Please wait.  / ") (princ "\n ")
        (princ "\n "))
                     ((= XX 1) (princ "\n ") (princ "\n ") (princ "\n Please wait.  - ") (princ "\n ")
        (princ "\n "))
                     ((= XX 2) (princ "\n ") (princ "\n ") (princ "\n Please wait.  \\\ ") (princ "\n ")
        (princ "\n "))
                     ((= XX 3) (princ "\n ") (princ "\n ") (princ "\n Please wait.  | ") (princ "\n ")
        (princ "\n ")))
              )
             )
              (setq n (+ n 1))
              )  ;;progn
                );; if e2
               ) ;;while e2
              (setq e1 (entnext e1))
              )  ;; porgn
            ) ;;; if e1
          ) ;;while e1
        (setq hgh (abs (- yMax yMin)))
        (setq areaN_ (* disMem hgh))
        (if (or (< areaN_ a_Min)(= a_Min nil)) (setq a_Min areaN_ agM_Min angRot
                                                     dsM_Min disMem pM_Min pMem0 yMaxMm yMax
           yMinMm yMin))
        (setq angDeg (+ angDeg det_Ag))
        ) ; while  (<= angRot 360)
      (setq angMeD (/ (* agM_Min 180) pi))
      (setq bstAg1 angMeD)
      (setvar "OSMODE" osVar)
      (princ "\n") (princ "\n")
      (princ "\n") (princ "  Press any key to continue,please.")
      (princ "\n") (princ "\n")
      (read-char)
      (tes) )
```

```
(defun tes ()
 (setq agM_MinD (/ (* agM_Min 180) pi))
 (setq gap (getreal "\n GAP <0> = "))
 (if (= gap nil) (setq gap 0))
 (setq dsMarry (* (- 1)(+ gap dsM_Min)))
 (setq tmp1 (* (/ areaS a_Min) 100))
 (setq tmp2 (- yMaxMm yMinMm))
 (graphscr)
 (command "ROTATE" "all" "" "0,0" agM_MinD)
 (command "ARRAY" "all" "" "R" "1" "3" dsMarry)
 (princ "\n") (princ "\n")
 (princ "\n") (princ "  Press any key to see data.")
 (princ "\n") (princ "\n")
 (read-char)
 (textscr)
 (princ "\n") (princ "*********************************************")
 (princ "\n") (princ "*********************************************")
 (princ "\n") (princ "                                             ")
 (princ "\n") (princ "   This is the data for 1 way strip layout.")
 (princ "\n") (princ "                                             ")
 (princ "\n ") (princ "     The best angle = ") (princ agM_MinD) (princ " degree.")
 (princ "\n ") (princ "  Distance for feed = ") (princ dsM_Min)  (princ " mm. <no gap>")
 (princ "\n ") (princ "        Strip width = ") (princ tmp2) (princ " mm. <no gap>")
 (princ "\n ") (princ "      The best Area = ") (princ a_Min) (princ ".<1 workpiece>")
 (princ "\n ") (princ "     The efficiency = ") (princ tmp1) (princ " % <no gap>")
 (princ "\n") (princ "                                             ")
 (princ "\n") (princ "*********************************************")
 (princ "\n") (princ "**    If the function is not correct or    **")
 (princ "\n") (princ "**    have some problem, please tell me.   **")
 (princ "\n") (princ "**                                         **")
 (princ "\n") (princ "*********************************************")
 (princ "\n") (princ "\n")
 (princ "\n") (princ "  Press any key to continue,please.")
 (princ "\n") (princ "\n")
 (read-char)
 (graphscr) )
```