# รายงานการวิจัยฉบับสมบูรณ์

# การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

ผู้วิจัย

รศ.วนิดา รัตนมณี

งานวิจัยนี้ได้รับทุนอุดหนุนการวิจัย จากเงินรายได้
มหาวิทยาลัยสงขลานครินทร์ ประจำปีงบประมาณ 2554

# คำนำ

งานวิจัย “การประยุกต์ใช้เจเนติกอัลกอริทึมสำหรับจัดตารางสอนนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์” มีวัตถุประสงค์เพื่อสร้างโปรแกรมคอมพิวเตอร์สำหรับช่วยจัดตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ โดยขั้นตอนในการทำวิจัยจะเริ่มต้นจากการสำรวจข้อมูล การวิเคราะห์ข้อมูลและวิเคราะห์ปัญหาเพื่อหาสาเหตุ จากนั้นนำข้อมูลที่ได้มาทำการวิเคราะห์เพื่อออกแบบและสร้างโปรแกรมคอมพิวเตอร์เพื่อช่วยในการจัดตารางสอน โดยใช้ข้อมูลตัวอย่างคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งจากการวิเคราะห์ข้อมูลพบปัญหา คือข้อมูลที่นำมาใช้ในการจัดตารางสอนนั้นมีเป็นจำนวนมาก มีความซับซ้อนของข้อมูล และข้อมูลยังคงมีความผิดพลาด อีกทั้งในการจัดตารางสอนแต่ละครั้งของทะเบียนนั้นเวลาในการจัดตารางสอนเป็นเวลานานมาก เนื่องจากวิธีการจัดตารางสอนแบบเก่าเป็นการจัดโดยเจ้าหน้าที่ ซึ่งต้องมีความชำนาญมาก นอกจากนี้ยังพบอีกว่าตารางสอนที่ได้จากวิธีการจัดตารางสอนแบบเก่ายังคงมีความผิดพลาดอยู่ คือมีการซ้ำซ้อนของเวลาสอนของอาจารย์ มีการซ้ำซ้อนของเวลาการเรียนของกลุ่มนักศึกษา และมีการซ้ำซ้อนของเวลาการใช้ห้องเรียนอยู่ จึงเกิดแนวคิดที่จะออกแบบและสร้างโปรแกรมคอมพิวเตอร์เพื่อช่วยในการจัดตารางสอน เพื่อให้ได้ตารางสอนที่ถูกต้อง ไม่มีการซ้ำซ้อนเกิดขึ้น และลดเวลาที่ใช้ในกระบวนการการจัดตารางสอน

ผู้วิจัยหวังเป็นอย่างยิ่งว่าเอกสารงานวิจัยชิ้นนี้จะเป็นประโยชน์ต่อมหาวิทยาลัยและผู้ที่สนใจทั่วไป หากมีข้อผิดพลาดประการใด ผู้วิจัยขอน้อมรับด้วยความเต็มใจและพร้อมที่จะแก้ไขปรับปรุงในโอกาสต่อไป

วนิดา รัตนมณี

# บทคัดย่อ

การจัดตารางสอนเป็นกระบวนที่ค่อนข้างยุ่งยากซับซ้อน มีข้อมูลและเงื่อนไขที่ต้องพิจารณาเป็นจำนวนมาก การจัดตารางสอนในปัจจุบันจัดโดยเจ้าหน้าที่ซึ่งต้องมีความรู้ประสบการณ์และความชำนาญ ดังนั้นในการจัดตารางสอนแต่ละครั้งต้องใช้เวลานานและเกิดความผิดพลาดอยู่เสมอ ในงานวิจัยนี้ได้นำเอาวิธีการของเจเนติกอัลกอริทึมไปใช้ในการพัฒนาโปรแกรมช่วยในการจัดตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ มีวัตถุประสงค์เพื่อให้ได้ตารางสอนที่มีความถูกต้อง ขัดแย้งกับเงื่อนไขน้อยที่สุด (minimize fitness value) และลดเวลาที่ใช้การจัดตารางสอน โปรแกรมที่พัฒนาขึ้นประกอบด้วยสามส่วนหลัก คือ ส่วนการป้อนข้อมูลเข้า ส่วนการประมวลผล ซึ่งประมวลผลตามวิธีการของเจเนติกอัลกอริทึม และส่วนแสดงผลลัพธ์ ซึ่งแสดงผลลัพธ์เป็นตารางสอนอาจารย์ ตารางเรียนตามกลุ่มนักศึกษา ตารางการใช้ห้องเรียนและภาระงานของอาจารย์ ในการประมวลผลด้วยวิธีการของเจเนติกอัลกอริทึมนั้นจะทำการทดลองเพื่อหาค่าพารามิเตอร์เจเนติกอัลกอริทึมที่เหมาะสมเพื่อกำหนดให้เป็นค่าเริ่มต้นของโปรแกรม สำหรับข้อมูลที่นำมาทดสอบประกอบด้วยรายวิชา 425 รายวิชา อาจารย์ 152 คน กลุ่มนักศึกษา 83 กลุ่มและห้องเรียน 59 ห้อง โดยมีพารามิเตอร์ที่เหมาะสมคือจำนวนคำตอบเบื้องต้น 2,500 โครโมโซม จำนวนประชากรเท่ากับ 100 โครโมโซมและจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน เปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรมที่พัฒนาขึ้นกับตารางสอนที่จัดด้วยวิธีการแบบเดิมที่กองทะเบียนและประมวลผลจัดมา พบว่าตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นสามารถลดการขัดแย้งกับเงื่อนไขบังคับได้ 100 เปอร์เซ็นต์ แต่มีการเพิ่มขึ้นของข้อขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์ 16 เปอร์เซ็นต์ อย่างไรก็ตามการจัดตารางสอนให้ความสำคัญกับเงื่อนไขบังคับเป็นหลัก การจัดตารางสอนต้องไม่ขัดแย้งกับเงื่อนไขบังคับ ดังนั้นตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นสามารถนำไปใช้งานได้จริง อีกทั้งโปรแกรมที่พัฒนาขึ้นยังมีความยืดหยุ่นของโปรแกรมให้สามารถจัดตารางสอนแบบ 90 นาที/คาบ และสามารถกำหนดเวลาว่างของอาจารย์ได้ โดยโปรแกรมสามารถประมวลผลได้สูงสุด 8,991 คาบเรียน

# ABSTRACT

Timetabling procedures are very complex and difficult because they have a number of data and constraints to be considered and determined. In the present, timetabling requires people who have knowledge, experience and expertise. It uses a plenty of time, and always has mistaken at the same time. In this report, genetic algorithms has been selected to create a program which can help to generate a timetable for students in the faculty of Engineering, Prince of Songkla University. The purposes of this program are providing the accurate timetable, having minimize conflicts with the conditions which has minimize fitness value, and reducing the time. The developed program consists of three main parts. The first step is the part of input data. Secondly, the part of data processing which follows the theory of genetic algorithms. Lastly, the part of result which will report the timetable of lecturers, students, schedule of classes, and obligation of lecturers. The processing by the genetic algorithms will start with testing to find out the suitable parameter of genetic algorithms in order to set the initial data of program. For data used in test, there are 425 subjects, 152 lectures, 83 student groups, and 59 classrooms. Moreover, the optimal values of GA parameters were 2,500 initializes, 100 population size and 500 generations. The results from the program developed was compared with the timetable which it was generated by traditional methods. The timetable from the program developed can reduce 100% of confliction and hard constraints. In contrast, it has increased 16% of confliction and soft constraints. However, the most important of timetabling is not conflict with the hard constraint. Therefore, the timetable from the program developed can be used. Moreover, this program has flexibility to generate the timetable which uses 90 minutes per period. Also, it can identify the available time of lecturers that the program can serve up to 8,991 periods.

# กิตติกรรมประกาศ

งานวิจัย “การประยุกต์ใช้เจเนติกอัลกอริทึมสำหรับจัดตารางสอนนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์” ได้สำเร็จลุล่วงไปได้ด้วยดีเพราะได้รับความอนุเคราะห์ให้ความสนับสนุนจากผู้มีพระคุณหลายท่าน ที่ได้ให้แนวคิด คำแนะนำ และข้อเสนอแนะต่าง ๆ ในการดำเนินงานมาโดยตลอด เพื่อให้ผลงานชิ้นนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยจึงขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณ คุณสุธน ไชยสุวรรณ ผู้อำนวยการกองทะเบียนและประมวลผล ผู้ให้ความรู้เกี่ยวกับกระบวนการจัดตารางสอน และให้การสนับสนุนข้อมูลอย่างเต็มที่ในการนำมาพัฒนาโปรแกรมจนบรรลุวัตถุประสงค์ในการทำงานวิจัย

ขอขอบคุณ คุณชุกรี แดสา สำหรับคำแนะนำและแนวคิดในการพัฒนาโปรแกรมจัดตารางสอน จนบรรลุวัตถุประสงค์ในการทำงานวิจัย

สุดท้ายนี้ขอขอบคุณมหาวิทยาลัยสงขลานครินทร์ ที่ได้ให้การสนับสนุนทุนวิจัยจากงบประมาณเงินรายได้ของ มหาวิทยาลัยสงขลานครินทร์ ประจำปีงบประมาณ 2554

วนิดา รัตนมณี

# สารบัญ


| | หน้า |
|---|---|
| สารบัญ | (6) |
| รายการตาราง | (8) |
| รายการภาพประกอบ | (12) |
| บทที่ | |
| 1 บทนำ | 1 |
| 1.1 ความสำคัญและที่มาของปัญหา | 1 |
| 1.2 งานวิจัยที่เกี่ยวข้อง | 12 |
| 1.3 วัตถุประสงค์ของการวิจัย | 16 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย | 16 |
| 1.5 ขอบเขตการวิจัย | 16 |
| 2 ทฤษฎีที่เกี่ยวข้องกับการวิจัย | 17 |
| 2.1 ปัญหาการจัดตารางสอน (timetabling problem) | 17 |
| 2.2 คำจำกัดความของปัญหา (problem) ลำดับขั้นการแก้ปัญหา (algorithm) และความยากง่ายของลำดับการแก้ปัญหา (complexity) | 20 |
| 2.3 ปัญหาแบบ P, NP, NP-Completeness, NP-Hard, Strongly NP-Hard | 21 |
| 2.4 การหาจุดเหมาะสมที่สุด (optimization) | 22 |
| 2.5 ฮิวริสติก (heuristic) และเมตาฮิวริสติก (metaheuristic) | 23 |
| 2.6 เจเนติกอัลกอริทึม (Genetic Algorithms: GA) | 25 |
| 3 การพัฒนาโปรแกรมจัดตารางสอนโดยประยุกต์ใช้วิธีการเจเนติกอัลกอริทึม | 41 |
| 3.1 การวิเคราะห์ปัญหาระบบการจัดตารางสอน | 41 |
| 3.2 โครงสร้างของโปรแกรมจัดตารางสอน | 47 |
| 3.3 การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอน | 65 |
| 3.4 ลักษณะของโปรแกรมจัดตารางสอนที่พัฒนาขึ้น | 88 |
| 4 ผลลัพธ์โปรแกรมจัดตารางสอน | 108 |
| 4.1 การทดสอบความถูกต้อง | 109 |

# สารบัญ (ต่อ)


| | หน้า |
|---|---|
| 4.2 ผลลัพธ์จากการจัดตารางสอนของคณะวิศวกรรมศาสตร์ และการเปรียบเทียบผลลัพธ์ | 117 |
| 4.3 การวิเคราะห์ความไวเพื่อหาค่าของพารามิเตอร์เจเนติกอัลกอริทึม | 123 |
| 4.4 การวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ | 139 |
| 5 บทสรุปและข้อเสนอแนะ | 147 |
| 5.1 สรุปผลการดำเนินงานวิจัย | 147 |
| 5.2 ข้อเสนอแนะในการดำเนินงานวิจัย | 149 |
| บรรณานุกรม | 151 |
| ภาคผนวก | 155 |
| ภาคผนวก ก ตัวอย่างแบบสอบถามความสำคัญของเงื่อนไขหรือข้อจำกัดที่ใช้ในการจัดตารางสอน | 156 |

# รายการตาราง


| ตาราง | | หน้า |
|---|---|---|
| 1.1 | ลักษณะปัญหาของข้อมูลแต่ละตัวที่ใช้ในการจัดตารางสอน | 4 |
| 1.2 | รายละเอียดของข้อมูลที่ใช้ในการจัดตารางสอนที่ต้องทำการศึกษา | 8 |
| 2.1 | คำศัพท์และความหมายทางพันธุศาสตร์เปรียบเทียบกับเจเนติกอัลกอริทึม | 28 |
| 3.1 | ลำดับความสำคัญของเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์จากแบบสอบถาม | 45 |
| 3.2 | ลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์ | 46 |
| 3.3 | การกำหนดค่าน้ำหนักของเงื่อนไขในการจัดตารางสอน | 46 |
| 3.4 | ตัวแปรและข้อมูลตัวแปรที่ใช้ในการคำนวณคะแนนค่าความเหมาะสม | 50 |
| 3.5 | รายละเอียดรายวิชาที่เปิดสอนในภาคเรียนที่พิจารณา | 68 |
| 3.6 | รายละเอียดของรายวิชาที่จัดตารางสอน | 74 |
| 3.7 | ตัวอย่างการคำนวณค่าความน่าจะเป็นในการถูกคัดเลือก และค่าความน่าจะเป็นในการถูกคัดเลือกสะสม | 77 |
| 3.8 | ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมชุดใหม่ | 78 |
| 3.9 | ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมชุดใหม่ด้วยวิธี Tournament | 79 |
| 3.10 | ข้อดี-ข้อเสียของวิธีการคัดเลือกแบบวงล้อรู้เล็ทกับวิธีการคัดเลือกแบบ Tournament | 79 |
| 3.11 | ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมเพื่อทำการครอสโอเวอร์ | 82 |
| 3.12 | ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมเพื่อทำการมิวเตชัน | 85 |
| 3.13 | ข้อมูลป้อนเข้าสำหรับฐานข้อมูล | 90 |
| 4.1 | รายละเอียดตัวอย่างรายวิชาบรรยายที่ใช้ในการทดสอบความถูกต้อง | 110 |
| 4.2 | รายละเอียดตัวอย่างรายวิชาปฏิบัติที่ใช้ในการทดสอบความถูกต้อง | 111 |
| 4.3 | รายละเอียดตัวอย่างรายวิชานอกคณะที่ใช้ในการทดสอบความถูกต้อง | 111 |
| 4.4 | รายละเอียดตัวอย่างอาจารย์ผู้สอนที่ใช้ในการทดสอบความถูกต้อง | 112 |
| 4.5 | รายละเอียดตัวอย่างกลุ่มนักศึกษาที่ใช้ในการทดสอบความถูกต้อง | 112 |
| 4.6 | รายละเอียดตัวอย่างห้องเรียนที่ใช้ในการทดสอบความถูกต้อง | 113 |
| 4.7 | ตัวอย่างตำแหน่งของยีน | 114 |
| 4.8 | จำนวนครั้งที่ผิดเงื่อนไขของตารางสอนตัวอย่าง | 116 |

## รายการตาราง (ต่อ)


| ตาราง | | หน้า |
|---|---|---|
| 4.9 | ข้อมูลที่ใช้ในจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2553 | 117 |
| 4.10 | จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2553 (50 นาทีต่อคาบ) | 118 |
| 4.11 | จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2553 (90 นาทีต่อคาบ) | 119 |
| 4.12 | รายละเอียดการเปรียบเทียบจำนวนการขัดแย้งเงื่อนไขของตารางสอน ภาคการศึกษา 1/2553 | 121 |
| 4.13 | ข้อมูลที่ใช้ในจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2555 | 122 |
| 4.14 | จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2555 (50 นาทีต่อคาบ) | 122 |
| 4.15 | จำนวนยีนต่อ 1 โครโมโซมในแต่ละช่วงจำนวนยีน | 124 |
| 4.16 | รายละเอียดข้อมูลที่ใช้ในการจัดตารางสอนในแต่ละช่วงจำนวนยีน | 125 |
| 4.17 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีนน้อยกว่า 500 ยีน) | 126 |
| 4.18 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนน้อยกว่า 500 ยีน | 127 |
| 4.19 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 500-1,000 ยีน) | 127 |
| 4.20 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 500-1,000 ยีน | 128 |
| 4.21 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 1,000-2,000 ยีน) | 129 |

## รายการตาราง (ต่อ)


| ตาราง | | หน้า |
|---|---|---|
| 4.22 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 1,000-2,000 ยีน | 129 |
| 4.23 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 2,000-3,000 ยีน | 130 |
| 4.24 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 2,000-3,000 ยีน | 131 |
| 4.25 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 3,000-4,000 ยีน) | 132 |
| 4.26 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 3,000-4,000 ยีน | 132 |
| 4.27 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 4,000-5,000 ยีน) | 133 |
| 4.28 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 4,000-5,000 ยีน | 133 |
| 4.29 | ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (ยีนมากกว่า 5,000 ยีน) | 134 |
| 4.30 | ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนมากกว่า 5,000 ยีน | 135 |
| 4.31 | ค่าพารามิเตอร์ที่เหมาะสมสำหรับจำนวนยีนในแต่ละช่วงจำนวนยีน | 136 |
| 4.32 | ผลการวิเคราะห์จำนวนคำตอบที่เป็นไปได้เบื้องต้น | 137 |
| 4.33 | ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง | 139 |
| 4.34 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 1 | 140 |

## รายการตาราง (ต่อ)


| ตาราง | | หน้า |
|---|---|---|
| 4.35 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 2 | 141 |
| 4.36 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 3 | 142 |
| 4.32 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 4 | 143 |
| 4.38 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 5 | 144 |
| 4.39 | จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 6 | 145 |
| 4.40 | ผลการวิเคราะห์ความไวของค่าน้ำหนักเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง | 145 |

# รายการภาพประกอบ


| ภาพประกอบ | | หน้า |
|---|---|---|
| 1.1 | วิธีการรวบรวมข้อมูลที่ใช้ในการจัดตารางสอน | 3 |
| 1.2 | วิธีการจัดตารางสอน | 5 |
| 1.3 | แบบฟอร์มการจัดตารางสอน | 6 |
| 1.4 | แบบฟอร์มการจัดห้องเรียน | 7 |
| 1.5 | ตัวอย่างตารางเรียนของกลุ่มนักศึกษาวิศวกรรมอุตสาหการ ชั้นปีที่ 3 (3IE) | 9 |
| 1.6 | ตัวอย่างตารางการใช้ห้องเรียน IEF309 | 9 |
| 2.1 | ตัวอย่างตารางสอนอย่างง่าย | 18 |
| 2.2 | ตัวอย่างลักษณะทางพันธุกรรมของเมล็ดถั่ว | 27 |
| 2.3 | ขั้นตอนการทำงานของวิธีการของเจเนติกอัลกอริทึม | 29 |
| 2.4 | ตัวอย่างการเข้ารหัสแบบเลขฐานสอง | 31 |
| 2.5 | ตัวอย่างการเข้ารหัสแบบตัวเลข | 31 |
| 2.6 | ตัวอย่างการเข้ารหัสแบบเปลี่ยนลำดับ | 31 |
| 2.7 | การสุ่มหากลุ่มประชากรเบื้องต้นจำนวน 4 โครโมโซม | 33 |
| 2.8 | สัดส่วนของค่าความเหมาะสมของโครโมโซม | 34 |
| 2.9 | การครอสโอเวอร์แบบ One-Point Crossover | 37 |
| 2.10 | การกลายพันธุ์แบบกลับบิต | 38 |
| 2.11 | การกลายพันธุ์แบบผกผัน | 38 |
| 2.12 | การกลายพันธุ์แบบแทรก | 38 |
| 2.13 | การกลายพันธุ์แบบเคลื่อนตำแหน่ง | 39 |
| 2.14 | การกลายพันธุ์แบบสลับตำแหน่ง | 39 |
| 3.1 | รูปแบบของโครงสร้างโปรแกรมที่ออกแบบไว้ | 47 |
| 3.2 | โครงสร้างกระบวนการทำงานของโปรแกรมการจัดตารางสอน | 48 |
| 3.3 | ความสัมพันธ์ของตารางข้อมูลในฐานข้อมูล | 49 |
| 3.4 | ขั้นตอนการคำนวณจำนวนคาบสอนที่ซ้ำซ้อนกันของอาจารย์ | 52 |
| 3.5 | ขั้นตอนการคำนวณจำนวนคาบเรียนที่ซ้ำซ้อนกันของกลุ่มนักศึกษา | 53 |
| 3.6 | ขั้นตอนการคำนวณจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น. | 55 |

## รายการภาพประกอบ (ต่อ)


| ภาพประกอบ | | หน้า |
|---|---|---|
| 3.7 | ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่อาจารย์สอนติดต่อกันเกิน 4 คาบ | 56 |
| 3.8 | ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนติดต่อกันเกิน 4 คาบ | 58 |
| 3.9 | ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป | 60 |
| 3.10 | ขั้นตอนการคำนวณค่าจำนวนครั้งในแต่ละวันที่มีเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง | 62 |
| 3.11 | ขั้นตอนการคำนวณจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกัน | 64 |
| 3.12 | ตัวอย่างโครโมโซม | 65 |
| 3.13 | ตัวอย่างวัน-เวลาที่ใช้ในการจัดตารางสอน | 66 |
| 3.14 | ตัวอย่างแบบจำลองโครโมโซม | 67 |
| 3.15 | ตัวอย่างโครโมโซมที่สร้างจากข้อมูลตัวอย่าง | 68 |
| 3.16 | ขั้นตอนการประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึม | 71 |
| 3.17 | การสร้างโครโมโซม | 72 |
| 3.18 | ตารางสอนที่ได้จากการเข้ารหัสโครโมโซม | 75 |
| 3.19 | ตัวอย่างการสร้างวงล้อรูเล็ทจากข้อมูลตัวอย่างในตาราง 3.7 | 77 |
| 3.20 | ขั้นตอนการครอสโอเวอร์ | 81 |
| 3.21 | ตัวอย่างวิธีการครอสโอเวอร์แบบ Order Crossover (OX) | 83 |
| 3.22 | ขั้นตอนการคัดเลือกโครโมโซมเพื่อทำการมิวเตชัน | 85 |
| 3.23 | ตัวอย่างวิธีการมิวเตชันแบบ Two-point Swapping Mutation | 86 |
| 3.24 | ลักษณะโปรแกรมจัดตารางสอนที่พัฒนาขึ้น | 88 |
| 3.25 | หน้าจอ Login สำหรับจัดการฐานข้อมูล | 89 |
| 3.26 | เมนูจัดการฐานข้อมูล | 91 |
| 3.27 | หน้าต่างสำหรับป้อนข้อมูลรายวิชาในคณะ | 92 |
| 3.28 | หน้าต่างแสดงการเพิ่มข้อมูลรายวิชาในคณะ | 93 |

## รายการภาพประกอบ (ต่อ)


| ภาพประกอบ | | หน้า |
|---|---|---|
| 3.29 | หน้าต่างสำหรับป้อนข้อมูลรายวิชานอกคณะ | 94 |
| 3.30 | หน้าต่างแสดงการเพิ่มข้อมูลรายวิชานอกคณะ | 94 |
| 3.31 | หน้าต่างสำหรับป้อนข้อมูลอาจารย์ | 95 |
| 3.32 | หน้าต่างแสดงการเพิ่มข้อมูลอาจารย์ | 96 |
| 3.33 | หน้าต่างสำหรับป้อนข้อมูลกลุ่มนักศึกษา | 96 |
| 3.34 | หน้าต่างแสดงการเพิ่มข้อมูลกลุ่มนักศึกษา | 97 |
| 3.35 | หน้าต่างสำหรับป้อนข้อมูลห้องเรียน | 98 |
| 3.36 | หน้าต่างแสดงการเพิ่มข้อมูลห้องเรียน | 98 |
| 3.37 | หน้าต่างการกำหนดภาคการศึกษาและเวลาเรียนต่อคาบ | 100 |
| 3.38 | หน้าต่างการกำหนดช่วงเวลาของอาจารย์ที่ไม่ต้องการสอน | 101 |
| 3.39 | หน้าต่างการกำหนดค่าน้ำหนักของเงื่อนไขต่างๆ | 101 |
| 3.40 | หน้าต่างที่ปรากฏเมื่อกดปุ่ม “Run” | 102 |
| 3.41 | หน้าต่างการกำหนดการแสดงผล | 102 |
| 3.42 | หน้าต่างที่ปรากฏเมื่อกดปุ่ม “ทั้งหมด” | 103 |
| 3.43 | หน้าต่างแสดงตัวอย่างตารางสอนอาจารย์ | 103 |
| 3.44 | ตัวอย่างตารางสอนของอาจารย์ | 104 |
| 3.45 | หน้าต่างแสดงตัวอย่างตารางเรียนของกลุ่มนักศึกษา | 104 |
| 3.46 | ตัวอย่างตารางเรียนของกลุ่มนักศึกษา | 105 |
| 3.47 | หน้าต่างแสดงตัวอย่างตารางการใช้ห้องเรียน | 105 |
| 3.48 | ตัวอย่างตารางการใช้ห้องเรียน | 105 |
| 3.49 | ตัวอย่างภาระงานสอนของอาจารย์ | 106 |
| 3.50 | หน้าต่างเมนูเจเนติกอัลกอริทึม | 107 |
| 4.1 | ตัวอย่างส่วนหนึ่งของโครโมโซมที่ใช้ในการทดสอบความถูกต้อง | 114 |
| 4.2 | ตารางการใช้ห้องเรียน A200 | 114 |
| 4.3 | ตัวอย่างตารางสอนของอาจารย์ | 115 |
| 4.4 | ตัวอย่างตารางสอนของนักศึกษา | 115 |
| 4.5 | การเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้น | 136 |

## รายการภาพประกอบ (ต่อ)


| ภาพประกอบ | | หน้า |
|---|---|---|
| 4.6 | การลดลงของค่าความเหมาะสมเมื่อจำนวนคำตอบที่เป็นไปได้เบื้องต้นเพิ่มขึ้น | 138 |

# บทที่ 1

# บทนำ

## 1.1 ความสำคัญและที่มาของปัญหา

การจัดตารางสอนเป็นปัจจัยที่มีความสำคัญอย่างหนึ่งในการจัดการเรียนการสอน ให้สามารถดำเนินไปด้วยความราบรื่น และส่งผลต่อประสิทธิภาพการเรียนของนักศึกษา เป็นกระบวนการนำรายวิชาที่เปิดสอนมากำหนดในตารางเวลาให้มีความสัมพันธ์กับทรัพยากรที่มีอยู่ เช่น อาจารย์ ห้องเรียน อุปกรณ์การเรียนการสอน เป็นต้น

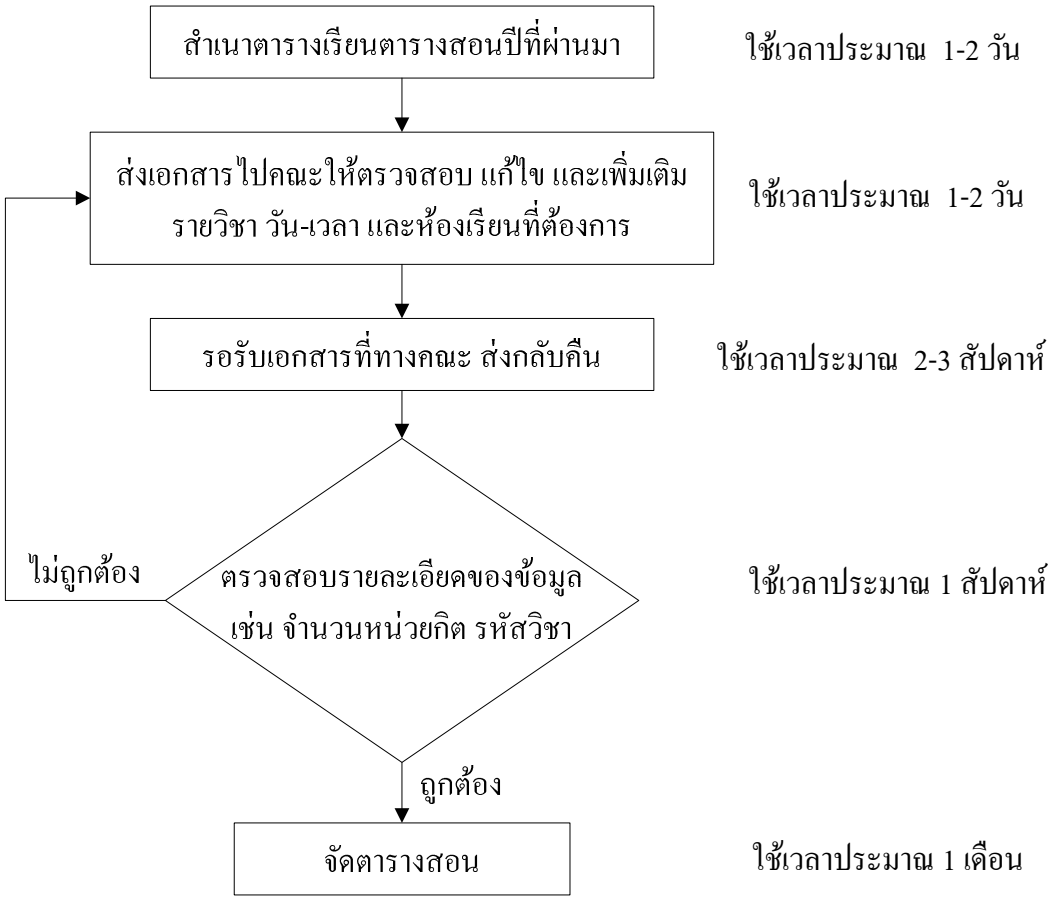
ปัจจุบันคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ได้มีการขยายตัวทางด้านการศึกษาเพิ่มมากขึ้น มีนโยบายที่จะขยายการศึกษาในสาขาวิชาต่างๆ อีกทั้งนักศึกษาได้มีจำนวนเพิ่มขึ้นมาก จากเหตุผลดังกล่าวจึงได้มีการเปิดการเรียนการสอน ทั้งหมด 28 สาขาวิชา แยกเป็นระดับปริญญาตรี 12 สาขาวิชา ระดับปริญญาโท 11 สาขาวิชา ปริญญาโทหลักสูตรพิเศษ 2 สาขาวิชา และระดับปริญญาเอก 8 สาขาวิชา ดังนั้นการจัดการเรียนการสอนแต่ละภาคการศึกษาจะมีประมาณ 350 รายวิชา 700 ตอน จำนวนนักศึกษาประมาณ 2800 คน มีอาจารย์สอนประมาณ 150 คน และมีห้องเรียนประมาณ 41 ห้องเรียน (ข้อมูลปี 2552)

กองทะเบียนและประมวลผลได้มีการปรับปรุงและพัฒนาวิธีการจัดตารางสอนจากเดิมบุคลากรที่รับผิดชอบเป็นผู้ดำเนินการทั้งหมด ทั้งการกรอกข้อมูล จัดเก็บข้อมูล และการจัดตารางสอน ต่อมากองทะเบียนและประมวลผลได้นำโปรแกรม Oracle เข้ามาช่วยในการจัดตารางสอน เพื่อลดขั้นตอนในการปฏิบัติงาน แต่โปรแกรมที่นำเข้ามานี้เป็นเพียงโปรแกรมที่เข้ามาช่วยในการจัดการฐานข้อมูลเท่านั้น ทำหน้าที่ในการจัดเก็บข้อมูลที่เกี่ยวข้องกับการจัดตารางสอน ซึ่งช่วยลดเวลาในกระบวนการกรอกข้อมูลและสำเนาตารางสอนของภาคการศึกษาที่ผ่านมาเท่านั้น ส่วนในขั้นตอนการจัดตารางสอนนั้นยังไม่มีโปรแกรมใดเข้ามาช่วยในกระบวนการนี้ ยังคงทำการกำหนดวัน-เวลา ห้องเรียน ให้กับนักศึกษาและอาจารย์ผู้สอนโดยเจ้าหน้าที่ ดังนั้นเจ้าหน้าที่ที่รับผิดชอบในการจัดตารางสอนจึงต้องมีความรู้ ประสบการณ์ และความชำนาญในการดำเนินงานที่ต้องดำเนินการด้วยความรวดเร็วและถูกต้อง

การจัดตารางสอนในปัจจุบันมีปัจจัยที่กองทะเบียนและประมวลผลต้องการในการจัดตารางสอนดังนี้ (หัวถัดไปที่กล่าวถึงขอใช้ คณะฯ แทน คณะวิศวกรรมศาสตร์ และ ภาควิชาฯ แทนภาควิชาที่สังกัดในคณะวิศวกรรมศาสตร์ ตลอดทั้งเล่มวิทยานิพนธ์)

1. **แผนการศึกษา** ของทุกหลักสูตร จะกำหนดแผนการศึกษาทุกสาขาวิชาเป็นภาคการศึกษาและชั้นปีตลอดหลักสูตร การศึกษาแผนการศึกษาเพื่อนำข้อมูลรายวิชาที่กำหนดในแผนการศึกษามาเป็นแนวทางการเปิดสอนในแต่ละภาคการศึกษา และเป็นแนวทางการจัดตารางสอน
2. **รายวิชาที่เปิดสอนในแต่ละภาคการศึกษา** คือ รายวิชาที่คณะ ภาควิชาฯ แจ้งเปิดสอนตามแผนการศึกษาที่กำหนดในหลักสูตรโดยแยกข้อมูลเป็นภาคการศึกษา ประกอบด้วย รหัสวิชา ชื่อรายวิชา และจำนวนหน่วยกิต
3. **จำนวนตอนที่เปิดสอนแต่ละรายวิชา** ทุกรายวิชาที่เปิดสอนจะมีจำนวนตอนที่เปิดสอนต่างกัน และคณะจะต้องแจ้งจำนวนตอนของรายวิชาที่เปิดสอนให้สอดคล้องกับการจัดตารางสอนของทุกหลักสูตร ลักษณะเนื้อหาวิชา กลุ่มผู้เรียน จำนวนนักศึกษา เครื่องมืออุปกรณ์การเรียนการสอน และความจุของห้องเรียน
4. **กลุ่มนักศึกษาแต่ละรายวิชา** คือ กลุ่มนักศึกษาที่คณะระบุให้เรียนในแต่ละรายวิชา การแบ่งกลุ่มนักศึกษาจะแบ่งตามภาควิชาและชั้นปี ซึ่งในแต่ละภาควิชาอาจจะแบ่งนักศึกษาออกเป็นกลุ่มย่อย หากเป็นกลุ่มนักศึกษาที่ใหญ่ โดยมีกลุ่มนักศึกษาทั้งหมด 83 กลุ่ม ยกตัวอย่างกลุ่มของนักศึกษา เช่น 2IE หมายความว่าเป็นนักศึกษาภาควิศวกรรมอุตสาหการ ชั้นปีที่ 2
5. **วันและเวลาเรียน** เพื่อความสะดวกในการจัดตารางสอนจะกำหนดวันเป็นอักษรย่อ และกำหนดเวลาเรียนหนึ่งวันมี 8 คาบๆ ละ 50 นาที เพื่อให้นักศึกษาและอาจารย์ผู้สอนมีเวลาประมาณ 10 นาทีในการเคลื่อนย้ายห้องเรียนเพื่อเรียนรายวิชาใหม่
6. **ห้องเรียน** คือ ห้องที่ใช้ในการจัดการเรียนการสอน ซึ่งกองทะเบียนและประมวลผลได้รวบรวมรายละเอียดข้อมูลห้องทุกห้องที่ใช้ในการจัดการเรียนการสอน เช่น ชื่อห้องเรียน ประเภทของห้องและความจุของห้องเรียน แล้วนำมาบันทึกเป็นฐานข้อมูล
7. **อาจารย์ผู้สอน** มี 4 ประเภท คือ อาจารย์ผู้สอนประจำ บุคลากรที่ปฏิบัติหน้าที่ช่วยสอน กลุ่มผู้สอนและอาจารย์พิเศษ
8. **จำนวนรับ** เป็นการกำหนดจำนวนรับนักศึกษาในกลุ่มนักศึกษาที่ต้องลงทะเบียนเรียนในแต่ละรายวิชา หากรายวิชาใด คณะฯ/ภาควิชาฯหรืออาจารย์ผู้สอนต้องการให้กำหนดจำนวนรับนักศึกษาให้แจ้งข้อมูลพร้อมกับการเปิดสอนทุกครั้ง

การได้มาซึ่งข้อมูลในการจัดตารางสอนสามารถแสดงได้ในภาพประกอบ 1.1 ทางกองทะเบียนและประมวลผลจะทำการสำเนาตารางสอนตามแผนการเรียนของปีที่ผ่านมา เพื่อที่จะใช้สำหรับการจัดตารางสอนในปีที่กำลังพิจารณา ส่งเอกสารให้คณะฯ แล้วคณะฯ จะส่งต่อเอกสารให้ภาควิชาฯ เพื่อทำการตรวจสอบ แก้ไข เพิ่มเติมรายวิชา วัน-เวลา และห้องเรียนตามต้องการ เมื่อแล้วเสร็จ ภาควิชาฯจะส่งคืนให้คณะ จากนั้นคณะจะรวบรวมส่งคืนให้กองทะเบียนและประมวลผลตามเวลาที่กำหนดในเอกสาร เมื่อกองทะเบียนและประมวลผลได้รับข้อมูลแล้วก็จะทำการตรวจสอบรายละเอียดของข้อมูลให้ถูกต้อง เช่น ตรวจสอบรายละเอียดของวิชาที่เปิดสอนให้ถูกต้องตามแผนการศึกษาของคณะ สาขาวิชา จำนวนหน่วยกิต และชั้นปี หากไม่ถูกต้องก็ดำเนินการติดต่อไปยังคณะ/ภาควิชา เพื่อทำการแก้ไข แต่หากถูกต้องแล้วก็จะนำข้อมูลไปใช้ในการจัดตารางสอนต่อไป ในส่วนของการเก็บรวบรวมข้อมูลเพื่อนำมาใช้ในการจัดตารางสอนจะดำเนินการเป็นระยะเวลาประมาณ 1 เดือน ตามแผนจะมีการดำเนินการในช่วงเดือน พฤศจิกายน – ธันวาคม



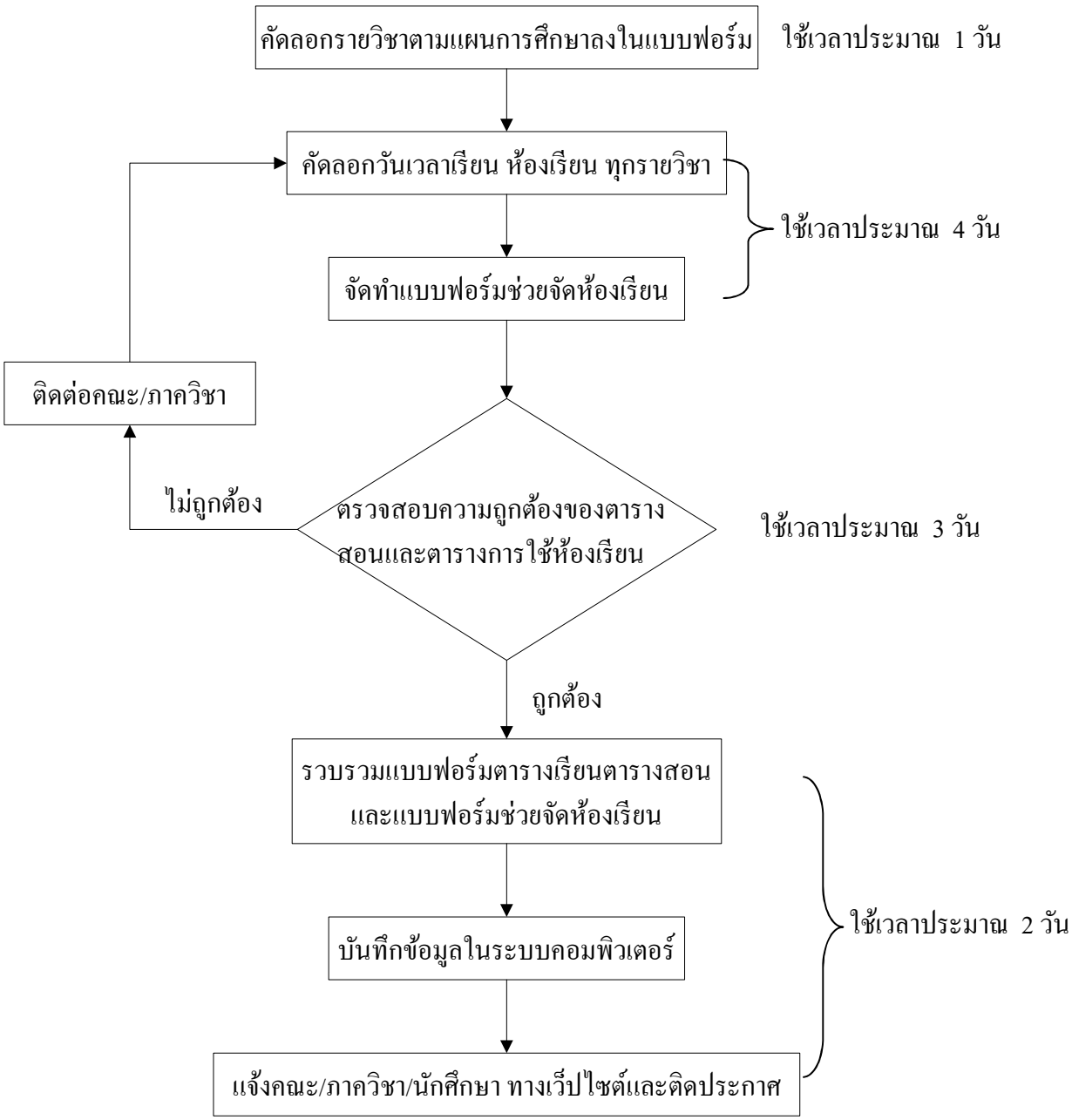

ภาพประกอบ 1.1 วิธีการรวบรวมข้อมูลที่ใช้ในการจัดตารางสอน

ถึงแม้ข้อมูลสำหรับการจัดตารางสอนที่กองทะเบียนและประมวลผลได้รับจะผ่านการตรวจสอบจากภาควิชาฯ มาแล้ว แต่ข้อมูลบางตัวก็ยังคงมีปัญหาอยู่ ปัญหาของข้อมูลแต่ละตัวอาจจะเกิดขึ้นจากความผิดพลาดของการตรวจสอบข้อมูลหรือเกิดขึ้นจากข้อจำกัดของทรัพยากรที่ใช้ในการจัดตารางสอนที่มีอยู่ จากตาราง 1.1 จะแสดงให้เห็นถึงลักษณะปัญหาของข้อมูลแต่ละตัวจะส่งผลกระทบต่อความยากง่าย ความซับซ้อนในการจัดตารางสอน เนื่องจากเจ้าหน้าที่ที่จัดตารางสอนจะต้องพิจารณาข้อมูลที่นำมาใช้ในการจัดตารางสอนที่มีจำนวนมากแล้ว แล้วยังจะต้องพิจารณาถึงลักษณะปัญหาของข้อมูลแต่ละชนิด ต้องจัดให้ตารางที่ได้นั้นมีข้อมูลที่ไม่ขัดแย้งกัน ให้ได้ตารางสอนที่เป็นไปได้สามารถนำไปใช้งานได้จริง

ตาราง 1.1 ลักษณะปัญหาของข้อมูลแต่ละตัวที่ใช้ในการจัดตารางสอน

| **ข้อมูล** | **ลักษณะปัญหาของข้อมูล** |
|---|---|
| รายวิชา | จำนวนรายวิชาเปิดสอนไม่เพียงพอกับจำนวนนักศึกษาที่ต้องเรียนตามแผนการเรียน โดยเฉพาะวิชาภาษา รายวิชาทางสังคมศาสตร์ และรายวิชาพลศึกษา |
| ห้องเรียน | - จำนวนห้องเรียนบรรยายที่ใช้จัดการเรียนการสอนมีจำนวนจำกัด<br>- ห้องที่มีขนาดใหญ่ที่มีความจุมากมีน้อยไม่สอดคล้องกับหลักสูตร สาขาวิชา และจำนวนนักศึกษาที่เพิ่มขึ้น<br>- สื่อทัศนูปกรณ์ในห้องเรียนไม่ทันสมัย มาตรฐานของสื่อการสอนแต่ละห้องไม่เหมือนกัน จึงทำให้อาจารย์ผู้สอนต้องการเลือกห้องเรียน<br>- จำนวนห้องปฏิบัติการและเครื่องมือในการปฏิบัติการไม่เพียงพอกับจำนวนนักศึกษา |
| อาจารย์ผู้สอน | จำนวนอาจารย์ผู้สอนไม่เพียงพอ |
| อื่นๆ | - มหาวิทยาลัยมีการปรับปรุง แก้ไขกฎ ระเบียบ ข้อบังคับ มติ ประกาศ และเงื่อนไข ที่มีผลกระทบกับการจัดการเรียนการสอน<br>- ปัจจุบันเปิดหลักสูตรหลากหลาย มีการจัดการเรียนการสอนต่างกัน เช่น บางหลักสูตรจัดการเรียนการสอนวันเสาร์อาทิตย์ หรือบางหลักสูตรเปิดภาคเรียนไม่ตรงกับหลักสูตรปกติ |

หลังจากกองทะเบียนและประมวลผลได้รับข้อมูลแล้ว ก็เริ่มเข้าสู่กระบวนการจัดตารางสอนของเจ้าหน้าที่ดังภาพประกอบ 1.2




ภาพประกอบ 1.2 วิธีการจัดตารางสอน

ซึ่งมีกระบวนการดังนี้

1. เจ้าหน้าที่เริ่มต้นจากการคัดลอกรายวิชา (รหัสวิชา) ที่เปิดสอนตามแผนการศึกษาของนักศึกษาแต่ละภาควิชาฯ แต่ละชั้นปี ลงในเอกสารแบบฟอร์มการจัดตารางสอน ดังภาพประกอบ 1.3 เพื่อช่วยในการตรวจสอบการจัดตารางสอนของนักศึกษาที่ลงทะเบียนเรียนตามแผนการศึกษาให้มีวัน-เวลาเรียนที่เหมาะสมไม่ซ้ำซ้อนกัน

2. คัดลอกวัน-เวลาเรียน ห้องเรียน ทุกรายวิชาที่เปิดสอน ยกเว้นรายวิชาที่ไม่ต้องกำหนดวันเวลาเรียน เช่น รายวิชาวิทยานิพนธ์ รายวิชาโครงงาน รายวิชาฝึกภาคสนาม รายวิชาฝึกงาน และรายวิชาที่อาจารย์ผู้สอนต้องการตกลงวันเวลาเรียนกับนักศึกษาเอง เป็นต้น
3. จัดทำแบบฟอร์มการจัดห้องเรียน ดังภาพประกอบ 1.4 โดยให้บันทึกชื่อห้องเรียนทุกห้องที่ได้รับผิดชอบลงแบบฟอร์ม บันทึกข้อมูลวันเวลาการใช้ห้องเรียนทุกรายวิชาที่เปิดสอนและใช้ห้องเรียนลงในแบบฟอร์มช่วยจัดห้องเรียน
4. ตรวจสอบข้อมูลตารางสอนและห้องเรียน หากรายวิชาใดที่มีเวลาเรียนหรือมีการใช้ห้องเรียนซ้ำซ้อนกัน ให้แก้ไขโดยประสานงานกับคณะและภาควิชาที่เกี่ยวข้อง เพื่อร่วมกันจัดวัน-เวลาเรียน และห้องเรียนใหม่ที่เหมาะสม
5. รวบรวมแบบฟอร์มตารางสอนของนักศึกษาแต่ละภาควิชา แต่ละชั้นปี และแบบฟอร์มช่วยจัดห้องเรียน เพื่อความสะดวกในการใช้ข้อมูล หรือแก้ไขข้อมูลภายหลัง
6. บันทึกข้อมูลในระบบคอมพิวเตอร์
7. แจ้งคณะ/ภาควิชา/นักศึกษาทางเว็ปไซต์และติดประกาศ

ภาคการศึกษา..... สำหรับ.....

| วัน | 8-9 | 9-10 | 10-11 | 11-12 | 12-13 | 13-14 | 14-15 | 15-16 | 16-17 | 17-18 | 18-19 | 19-20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| จ | | | | | | | | | | | | |
| อ | | | | | | | | | | | | |
| พ | | | | | | | | | | | | |
| พฤ | | | | | | | | | | | | |
| ศ | | | | | | | | | | | | |
| ส | | | | | | | | | | | | |
| อา | | | | | | | | | | | | |

| วิชาเรียน | สอบกลางภาค | สอบไล่ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| วิชาเรียน | สอบกลางภาค | สอบไล่ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| วิชาเรียน | สอบกลางภาค | สอบไล่ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

ภาพประกอบ 1.3 แบบฟอร์มการจัดตารางสอน

ห้อง..... จุ.....

| วัน | 8-9 | 9-10 | 10-11 | 11-12 | 12-13 | 13-14 | 14-15 | 15-16 | 16-17 |
|---|---|---|---|---|---|---|---|---|---|
| จ | | | | | | | | | |
| อ | | | | | | | | | |
| พ | | | | | | | | | |
| พฤ | | | | | | | | | |
| ศ | | | | | | | | | |
| ส | | | | | | | | | |
| อา | | | | | | | | | |

ภาพประกอบ 1. 4 แบบฟอร์มการจัดห้องเรียน

ในการจัดตารางสอนเจ้าหน้าที่จะต้องคำนึงถึงหลักการจัดตารางสอนอย่างเคร่งครัดเพื่อให้ได้ตารางสอนที่เหมาะสมและไม่เกิดข้อผิดพลาดขึ้น ซึ่งหลักการจัดตารางสอนนั้นมีหลักการดังนี้

1. จัดวัน-เวลาเรียนของนักศึกษาให้สอดคล้องกับวัน-เวลาสอนของอาจารย์ผู้สอน
2. จัดตารางสอนให้จำนวนนักศึกษาที่ลงทะเบียนเรียนเหมาะสมกับความจุของห้องเรียน
3. หากคณะหรือภาควิชามีความจำเป็นต้องปรับวันเวลาเรียน ด้วยเหตุที่ไม่ใช่เหตุผลส่วนตัว ก็จะปรับวันเวลาเรียนของรายวิชาเลือกโดยไม่ให้กระทบรายวิชาบังคับ
4. จัดวันเวลาเรียนรายวิชาบังคับให้คงเดิม ยกเว้นรายวิชามีเหตุผลที่จำเป็น เช่น มีการเปลี่ยนแปลงอาจารย์ผู้สอน เปลี่ยนแปลงกลุ่มผู้เรียน หรือไม่มีห้องเรียน
5. จัดตารางเรียนให้เหมาะสมกับลักษณะเนื้อหาวิชา เวลาเรียน อาจารย์ผู้สอนและห้องเรียน เช่น ในรายวิชาที่มีการคำนวณควรอยู่ในเวลาเช้า และห้องเรียนควรเป็นห้องที่มีกระดานเพื่อสะดวกต่อการสอน
6. กำหนดตารางเรียนรายวิชาบังคับหลัก และจัดรายวิชาที่ต้องเรียนรวมหลายกลุ่มสาขาวิชาเอก ก่อนโดยไม่ให้วันเวลาซ้ำซ้อนกัน
7. ไม่จัดตารางเรียนวันเสาร์และวันอาทิตย์ยกเว้นรายวิชาที่อาจารย์ผู้สอนต้องการหรือเป็นการกำหนดโดยหลักสูตร

8. วันเวลาเรียนรายวิชาบังคับในภาคการศึกษาเดียวกัน ของแต่ละปีการศึกษาพยายามคงเดิม เปลี่ยนแปลงเฉพาะรายวิชาเลือก

นอกจากหลักการจัดตารางสอนดังกล่าวที่เจ้าหน้าที่ต้องพิจารณาตลอดขณะทำการจัดตารางสอนแล้วเจ้าหน้าที่ยังต้องศึกษารายละเอียดของข้อมูลแต่ละประเภทที่ใช้ในการจัดตารางสอนก่อนจัดตารางสอน ซึ่งรายละเอียดของข้อมูลที่ต้องทำการศึกษาสามารถสรุปได้ตามตาราง 1.2

ตาราง 1.2 รายละเอียดของข้อมูลที่ใช้ในการจัดตารางสอนที่ต้องทำการศึกษา

| ข้อมูล | ลักษณะการพิจารณา |
|---|---|
| รายวิชาที่เปิดสอน | - ศึกษาจำนวนชั่วโมงที่เรียนต่อสัปดาห์ ประเภทของรายวิชาว่าเป็นรายวิชาบรรยาย รายวิชาปฏิบัติการ หรือรายวิชาที่มีทั้งเนื้อหาบรรยายและปฏิบัติการให้จัดชั่วโมงบรรยายก่อนปฏิบัติ<br>- ศึกษาลักษณะเนื้อหาวิชาก่อน โดยจัดช่วงเวลาเรียนให้เหมาะสม อาจเป็นช่วงเช้าหรือบ่าย |
| นักศึกษาที่เข้าเรียน | ศึกษาข้อมูลจำนวนนักศึกษาที่ลงทะเบียนเรียนในแต่ละรายวิชา |
| อาจารย์ผู้สอน | ศึกษาหน้าที่ที่รับผิดชอบและภาระงานอื่นๆ ของอาจารย์ผู้สอน เช่น งานด้านบริหาร |
| ห้องเรียน | - ศึกษาระยะทางระหว่างห้องเรียนให้นักศึกษาสามารถเข้าเรียนในห้องใกล้กัน เพื่อไม่ให้เสียเวลาในการเดินทาง เช่น จห้องในคณะเดียวกันสำหรับจัดวิชาบังคับ<br>- ศึกษาห้องเรียนใดเป็นของของคณะหรือภาควิชาใด เพื่อให้จัดรายวิชาที่สังกัดภาควิชาให้กับห้องเรียนของภาควิชานั้นก่อน<br>- ศึกษาข้อมูลรายละเอียดห้องเรียน เช่น ความจุ โสตทัศนูปกรณ์ ลักษณะห้องเรียน เป็นข้อมูลประกอบการพิจารณาการจัดห้องเรียน |
| วันและเวลาเรียน | - ศึกษาวันเวลาเรียนที่ใช้ทำการจัดตารางสอน เวลาที่ใช้การเรียนการสอนแต่ละคาบ เช่น รายวิชาบรรยายในจัดตารางสอนครั้งละ 1 คาบในช่วงเช้า รายวิชา 3 หน่วยกิตจัดให้เรียนวันจันทร์ วันพุธและวันศุกร์ ครั้งละ 1 คาบๆ ละ 50 นาที รายวิชาปฏิบัติการจัดให้เรียนช่วงบ่ายวันจันทร์ถึงวันศุกร์ ครั้งละ 3 ถึง 4 คาบ หรือช่วงเช้าวันอังคารและวันพฤหัสบดี ครั้งละ 3 คาบ |

ผลลัพธ์ที่ได้จากการจัดการเรียนการสอนของกองทะเบียนและประมวลผลในปัจจุบันมี 2 รูปแบบคือตารางเรียนตามกลุ่มนักศึกษาและตารางการใช้ห้องเรียน ดังภาพประกอบ 1.5 และ 1.6 ตามลำดับ

**ตารางเรียน 2/2552 ของกลุ่มนักศึกษา : 3IE**

| | 08.00 08.50 | 09.00 09.50 | 10.00 10.50 | 11.00 11.50 | 12.00 12.50 | 13.00 13.50 | 14.00 14.50 | 15.00 15.50 | 16.00 16.50 | 17.00 17.50 | 18.00 18.50 | 19.00 19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mon | 225-351(01) A200 วนิดา | 226-341(01) S102 กลางเดือน ธเนศ | 226-431(01) S104 วนิดา | 225-354(01) IEF309 นิกร | | | | | | | | |
| Tue | | 225-352(01) S103 เจริญ | | | | 226-435(01) IELAB สมชาย | | | | | | |
| Wed | 225-351(01) A200 วนิดา | 226-341(01) S101 กลางเดือน ธเนศ | 226-431(01) A401 วนิดา | 225-354(01) IEF308 นิกร | | 226-435(02) IELAB สมชาย | | | | | | |
| Thu | | 225-352(01) S101 เจริญ | | 226-431(01) S104 วนิดา | | 226-435(03) IELAB สมชาย | | | | | | |
| Fri | | 226-341(01) S101 กลางเดือน ธเนศ | 225-351(01) A202 วนิดา | | | 226-435(04) IELAB สมชาย | | | | | | |
| Sat | | | | | | | | | | | | |
| Sun | | | | | | | | | | | | |

รายวิชาที่ไม่ได้กำหนดเวลาเรียน : 225-353 225-353 225-353

ภาพประกอบ 1.5 ตัวอย่างตารางเรียนตามกลุ่มนักศึกษาวิศวกรรมอุตสาหการ ชั้นปีที่ 3 (3IE)

**ตารางการใช้ห้องเรียน 2/2552 : IEF309(00421)**

| | 08.00 08.50 | 09.00 09.50 | 10.00 10.50 | 11.00 11.50 | 12.00 12.50 | 13.00 13.50 | 14.00 14.50 | 15.00 15.50 | 16.00 16.50 | 17.00 17.50 | 18.00 18.50 | 19.00 19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mon | | 226-205(01) 2IE พิจิตร | | 225-354(01) 3IE นิกร | | | | 225-349(01) 3MfE กลางเดือน คำรณ | | | | |
| Tue | | 225-510(01) GISE สงวน | | | | | | | | | | |
| Wed | | 226-205(01) 2IE พิจิตร | 225-349(01) 3MfE กลางเดือน คำรณ | 226-317(01) 3MfE ประภาศ | | | | | | | | |
| Thu | | | | 226-317(01) 3MfE ประภาศ | | | | | | | | |
| Fri | | | 225-349(01) 3MfE กลางเดือน คำรณ | | | | | | | | | |
| Sat | | | | | | | | | | | | |
| Sun | | | | | | | | | | | | |

รายวิชาที่ไม่ได้กำหนดเวลาเรียน :

ภาพประกอบ 1.6 ตัวอย่างตารางการใช้ห้องเรียน IEF309

จากขั้นตอนการดำเนินการจัดตารางสอนจะเห็นได้ว่า ในการจัดแต่ละครั้งมีข้อมูลที่ใช้จัดตารางสอนเป็นจำนวนมาก ซึ่งประกอบไปด้วย รายวิชาที่เปิดสอน วันเวลาเรียน ห้องเรียน อาจารย์ผู้สอนและนักศึกษา นอกจากนั้นมีองค์ประกอบอื่นๆ เช่น หลักสูตร แผนการเรียน กฎระเบียบ ข้อบังคับ มติ ประกาศ ฯ ที่เกี่ยวกับการจัดตารางสอน หรือการกำหนดกิจกรรมต่างๆ ของมหาวิทยาลัย คณะ หรือภาควิชา เช่น สัปดาห์มอ.วิชาการ สัปดาห์วิทยาศาสตร์ เป็นต้น จากการที่มีข้อมูลที่เกี่ยวข้องจำนวนมากการจัดตารางสอนจึงเป็นกระบวนการที่ค่อนข้างยุ่งยากซับซ้อน ต้องสร้างความสัมพันธ์ของข้อมูลทั้งหมดไม่ให้เกิดความซ้ำซ้อน และเกิดความถูกต้องตามเงื่อนไขในการจัดตารางสอนมากที่สุด แต่ในปัจจุบันยังไม่มีโปรแกรมที่เข้ามาช่วยในขั้นตอนการจัดตารางสอนยังเป็นเพียงการจัดด้วยมือ เริ่มต้นจากกรอกแบบฟอร์มที่เป็นเพียงเอกสารที่มีเพียงตารางวันและเวลา เจ้าหน้าที่จะต้องกำหนดรายวิชา กลุ่มผู้เรียน อาจารย์ผู้สอน และห้องเรียนลงไปในตารางนั้น ทำให้ต้องใช้การจดจำ ประสบการณ์ และทักษะส่วนตัวในการจัด ซึ่งอาจจะเกิดข้อผิดพลาดได้ง่าย เนื่องจากไม่มีเครื่องมือใดที่ช่วยในการตรวจสอบข้อมูล ยังต้องอาศัยการตรวจสอบด้วยบุคลากรเพียงเท่านั้น จึงเป็นสาเหตุหนึ่งที่ทำให้เกิดข้อผิดพลาดจากการจัดตารางสอน โดยพบข้อผิดพลาดในการจัดตารางสอนของภาคการศึกษา 1/2553 ดังนี้

1. การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษาถึง 158 ครั้ง
2. การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอนถึง 129 ครั้ง
3. การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียนถึง 125 ครั้ง

ข้อผิดพลาดเหล่านี้จะทำให้ตารางสอนที่ได้ไม่สามารถใช้ได้จริง นอกจากนี้ยังพบข้อผิดพลาดอย่างอื่นที่ทำให้ตารางสอนที่ได้เป็นตารางสอนที่ยังไม่ค่อยเหมาะสม เช่น มีการจัดการเรียนการสอนในเวลาเที่ยงถึง 74 ครั้ง มีการจัดตารางสอนให้อาจารย์มีสอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบถึง 55 ครั้ง และจัดตารางเรียนให้นักศึกษามีเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบถึง 25 ครั้ง และมีการจัดตารางสอนในห้องต่างภาควิชาถึง 43 ครั้งเป็นต้น

จากข้อมูลดังกล่าวแสดงว่าปัญหาการจัดตารางสอนจัดเป็นปัญหาประเภท NP-Hard หมายถึงปัญหาที่มีขนาดข้อมูลจำนวนมากๆ ต้องใช้เวลาในการหาคำตอบยาวนานและเวลาในการหาคำตอบจะเพิ่มมากขึ้นแบบเอ็กซ์โปเนนเชียลเมื่อขนาดของปัญหาเพิ่มขึ้น แนวทางการแก้ไขปัญหาประเภท NP-Hard นี้สามารถแก้ไขได้ด้วยวิธีการแบบฮิวริสติก (heuristics) เนื่องจากใช้เวลาในการคำนวณน้อย อีกทั้งคำตอบที่ได้จากวิธีการแบบฮิวริสติก ก็สามารถยอมรับได้ในการนำไปใช้งานจริง การใช้วิธีการค้นหาคำตอบแบบฮิวริสติกหลายๆ วิธีได้ถูกนำมาประยุกต์ใช้ในการแก้ไขปัญหาต่างๆ อย่างแพร่หลาย ซึ่งเป็นที่รู้จักกัน เช่น วิธีการค้นหาแบบทาบู (Tabu Search) วิธีการ

ค้นหาคำตอบเชิงพันธุกรรมหรือวิธีการของเจเนติกอัลกอริทึม (Genetic Algorithm) วิธีการค้นหาแบบปีนเขา (Hill Climbing) วิธีการค้นหาดีสุดก่อน (Best-first Search) วิธีการค้นหาแบบกรีดดี (Greedy Algorithm) และวิธีการจำลองแอนเนลลิ่ง (Simulated Annealing) เป็นต้น ได้ถูกพัฒนาขึ้นและทดสอบกับการแก้ไขปัญหาต่างๆ พบว่าวิธีการแบบ ฮิวริสติกใช้งานได้ดีทั้งคุณภาพคำตอบและเวลาที่ได้จากวิธีการแบบฮิวริสติก สามารถปรับปรุงคุณภาพของคำตอบให้สูงขึ้นโดยการกำหนดระยะเวลาในการหาคำตอบให้มากขึ้นได้ในการประมวลผลเมื่อเทียบกับวิธีการหาคำตอบที่เหมาะสมที่สุด

วิธีการแก้ไขปัญหาการจัดตารางสอนสามารถทำได้หลายวิธี แต่ละวิธีการมีข้อดี-ข้อเสีย ต่างกัน เช่น วิธีการค้นหาแบบทาบูไม่ได้อาศัยการสุ่มหรือการเลือกจากความน่าจะเป็น แต่เป็นวิธีการค้นหาแบบดีเทอร์มินิสติก (deterministic) ซึ่งค้นหาคำตอบจากบริเวณข้างเคียงคำตอบที่ดีที่สุดในขณะนั้น โดยจะมีข้อห้ามไม่ให้ค้นหาซ้ำกับเซ็ตของคำตอบเดิมที่มีอยู่ หรือที่เรียกว่ารายการต้องห้าม (tabu list) ซึ่งวิธีการห้ามดังกล่าวจะเป็นการห้ามเพื่อมิให้ต้องไปหาผลเฉลยเดิม หรือหลงในวัฏจักร (cyclic) ซึ่งจะส่งผลให้ไม่สามารถหาคำตอบที่ดีขึ้นได้ วิธีการค้นหาแบบปีนเขาสามารถนำวิธีการฮิวริสติกที่ดีมาช่วยในกระบวนการค้นหาเพื่อให้ได้คำตอบอย่างรวดเร็วและมีประสิทธิภาพ แต่ในกรณีที่วิธีการฮิวริสติกไม่ดี อัลกอริทึมนี้ก็อาจหลงเส้นทางได้ และอาจไม่พบคำตอบแม้ว่าปริภูมิที่กำลังค้นหามีคำตอบอยู่ด้วยก็ตาม สาเหตุการหลงเส้นทางประการหนึ่งมาจากการเลือกสถานะลูก ซึ่งอัลกอริทึมจะไม่ได้พิจารณาสถานะลูกทุกตัวโดยเมื่อพบสถานะลูกตัวใดตัวหนึ่งที่ดีขึ้นก็จะเลือกเส้นทางนั้นทันที วิธีการค้นหาดีสุดก่อน เป็นกระบวนการค้นหาข้อมูลที่ได้นำเอาข้อดีของทั้งวิธีการค้นหาแบบลึกก่อน (Depth first search) และวิธีการค้นหาแบบกว้างก่อน (Breadth first search) มารวมกันเป็นวิธีการเดียว แต่วิธีการนี้จะใช้หน่วยความจำมาก เพราะต้องเก็บสถานะไว้ทุกตัวเพื่อหาเส้นทางจากสถานะเริ่มต้นไปหาคำตอบ วิธีการของเจเนติกอัลกอริทึม โดยใช้วิธีการทางพันธุ์ศาสตร์เข้ามาในกระบวนการค้นหาคำตอบของปัญหา เจเนติกอัลกอริทึมสามารถช่วยแก้ปัญหาที่มีขนาดค่อนข้างใหญ่และซับซ้อนได้ เนื่องจากมีคุณสมบัติของการเลียนแบบการถ่ายทอดลักษณะทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้พิจารณาในการหาคำตอบของประชากรรุ่นถัดไป เป็นกระบวนการที่จะทำให้ได้คำตอบที่ดีที่สุดจากกลุ่มคำตอบที่สุ่มมา นอกจากคำตอบที่ได้จะเป็นคำตอบที่เหมาะสมที่สุดแล้ว เจเนติกอัลกอริทึมยังเป็นวิธีการที่ใหม่กว่าวิธีการแบบอื่นดังที่กล่าวมา

สำหรับงานวิจัย “การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์” นำเอาวิธีการของเจเนติกอัลกอริทึมไปใช้ในการสร้างโปรแกรมช่วยในการจัดตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์ เพื่อให้ตารางสอนมีความเหมาะสม ถูกต้องและลดเวลาที่ใช้การจัดตารางสอน โดยผลลัพธ์ (output) ที่ได้จากโปรแกรมคือตารางสอนอาจารย์ ตารางเรียนตามกลุ่มนักศึกษา และตารางการใช้ห้องเรียน

## 1.2 งานวิจัยที่เกี่ยวข้อง

ปัญหาการจัดตารางสอนเป็นปัญหาที่เกิดขึ้นในสถาบันการศึกษาในทุกระดับตั้งแต่ระดับเตรียมอุดมศึกษาจนถึงระดับอุดมศึกษาซึ่งในการจัดตารางสอนแต่ละระดับจะมีข้อจำกัดและเงื่อนไขแตกต่างกัน โดยทั่วไปการจัดตารางสอนจะเป็นการจัดตารางเวลาที่ความเหมาะสมของหลักสูตร นักเรียนนักศึกษา อาจารย์ ห้องเรียน และคาบเรียน โดยผู้ที่มีหน้าที่จัดตารางสอนต้องพิจารณาตารางเวลาทั้งหมดของทุกหลักสูตรที่เปิดสอน คาบเรียน และห้องเรียน รวมถึงข้อจำกัดต่างๆ ในการจัดตารางสอนการจัดตารางสอน ดังนั้นผู้จัดตารางสอนจึงควรเป็นผู้มีความรู้ความสามารถมีเทคนิคและความชำนาญเป็นพิเศษ ด้วยการจัดตารางสอนมีข้อจำกัดหลายประการการจัดตารางสอนด้วยกำลังคนจะใช้เวลานานมาก ใช้จำนวนคนหลายคน และต้องตรวจสอบซ้ำๆ หลายครั้ง [1] แต่อย่างไรก็ตามตารางสอนก็ยังคงเกิดข้อผิดพลาดต่างๆ อยู่มากมาย เนื่องจากในปัจจุบันมีจำนวนผู้เรียนเพิ่มมากขึ้น จำนวนห้องเรียนเพิ่มมากขึ้น จำนวนรายวิชาที่เป็นสาระพื้นฐานและรายวิชาใหม่ๆ เป็นสาระเพิ่มเติมที่มีความเข้มข้นอย่างหลากหลายให้ผู้เรียนได้เลือกเรียนตามความถนัด ความสนใจ ความต้องการ ซึ่งต้องจัดให้ผู้เรียนได้เรียนพร้อมกันหลายกลุ่ม

ในการจัดการกับปัญหาการจัดตารางสอนได้มีผู้ประยุกต์ใช้ทฤษฎีทางวิศวกรรมคอมพิวเตอร์มาช่วยในการแก้ปัญหา โดยนักวิจัยได้มีการนำวิธีฮิวริสติกต่างๆ เข้ามาช่วยแก้ปัญหาการจัดตารางสอน เช่น วิธีการจำลองแอลเนลลิ่ง โดย D. Abramson [2] ได้นำวิธีการนี้เข้ามาแก้ไขปัญหาการจัดตารางสอนในโรงเรียนระดับมัธยม โดยมีเป้าหมายเพื่อให้ได้ตารางที่มีค่าฟังก์ชันต้นทุนน้อยที่สุด และได้ทดสอบกับขั้นตอนแบบอนุกรม (serial) พบว่าค่าเฉลี่ยเริ่มต้นของฟังก์ชันต้นทุนจะสูงตามลำดับความซับซ้อนของปัญหา ค่าฟังก์ชันต้นทุนที่น้อยที่สุดได้มาจากการกำหนดอัตราการลดค่าตัวแปรอย่างช้าๆ และพบว่าใช้เวลาในการประมวลนานมาก จึงได้เสนอแนะขั้นตอนวิธีการแบบขนาน (parallel) เพื่อช่วยลดเวลาในการประมวลผล วิธีการค้นหาแบบทาบูโดย D. Costa [3] ได้นำวิธีการนี้มาใช้ในการแก้ไขปัญหาการจัดตารางสอนในโรงเรียนมัธยมศึกษา พบว่าได้ผลดีใกล้เคียงกับการจัดตารางสอนด้วยมือ แต่ใช้เวลาในการทำงานน้อยกว่า แต่ก็ไม่สามารถจัดตารางสอนให้ได้ผลตามข้อบังคับได้ทั้งหมด ดังนั้น M. J. F. Souza [4] จึงได้ปรับปรุงการจัดตารางสอนด้วยวิธีการค้นหาแบบทาบูโดยในขั้นตอนของการสร้างคำตอบเบื้องต้นได้ใช้ greedy

procedure จากนั้นจะปรับปรุงคุณภาพคำตอบด้วยวิธีการค้นหาแบบทาบูโดยต้องการให้ได้ตารางสอนที่มีค่าฟังก์ชันเป้าหมายน้อยที่สุด แต่อย่างไรก็ตาม M. J. F. Souza ก็ได้ทดลองนำ Intraclasses - Interclasses procedure เข้ามาช่วยปรับปรุงหลังจากที่วิธีการค้นหาแบบทาบูให้ตารางที่ดีแล้ว พบว่าการใช้วิธีการค้นหาแบบทาบูร่วมกับ Intraclasses - Interclasses procedure ให้ตารางสอนที่มีคุณภาพดีขึ้นและรวดเร็วยิ่งขึ้น วิธีการค้นหาแบบระบบอาณาจักรมด (Ant Colony System) Socha [5] ได้ใช้ MAX-MIN Ant System ซึ่งเป็นการปรับปรุงจากวิธีการค้นหาแบบระบบอาณาจักรมดที่ J. Knowles and M. Sampels[6] ได้ใช้ในการแก้ไขปัญหาการจัดตารางสอน โดยเสนอ construction graph และรูปแบบ Pheromone ที่เหมาะสม ดังนั้น Burke, E. K. et al. [7] จึงได้นำวิธี Graph Based Hyper Heuristic มาแก้ไขปัญหาตารางสอนและตารางสอบ พบว่าวิธีนี้สามารถแก้ปัญหาได้เป็นอย่างดี โดยได้เสนอให้ปรับปรุงวิธีการนี้เพื่อให้สามารถแก้ปัญหาขนาดใหญ่ ที่มีความซับซ้อนต่อไป ดังนั้นวิธีการของเจเนติกอัลกอริทึมจึงถูกนำเข้ามาใช้ ซึ่งเป็นวิธีเหมาะสมกับปัญหาที่มีขนาดใหญ่ มีความยุ่งยากซับซ้อน [8] เนื่องจากเป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติซึ่งคิดค้นโดย John Holland เมื่อปี ค.ศ. 1970 มีคุณสมบัติในการเลียนแบบการถ่ายทอดลักษณะทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้พิจารณาในการหาคำตอบของประชากรรุ่นถัดมาเพื่อให้ได้คำตอบที่เหมาะสมที่สุด [9] อีกทั้งวิธีนี้ยังเป็นวิธีที่มีประสิทธิภาพและได้รับความนิยมสำหรับแก้ปัญหาต่างๆ ได้อย่างหลากหลายในปัจจุบัน เช่น ปัญหาการเดินรถทิศทางเดียว [10] ปัญหาการจัดสมดุลสายการประกอบแบบผลิตภัณฑ์ผสม [11] การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดลำดับการผลิตเฟอร์นิเจอร์ไม้ยางพารา [12] การประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบของปัญหาการออกแบบผังโรงงานเฟอร์นิเจอร์ไม้ยางพารา [13] เป็นต้น

จากงานวิจัยที่ผ่านมาวิธีการของเจเนติกอัลกอริทึมได้ถูกเริ่มต้นนำมาแก้ไขปัญหาการจัดตารางสอบ โดยสร้างคำตอบเบื้องต้นให้เป็นคำตอบที่เป็นไปได้ทุกคำตอบและเสนอการครอสโอเวอร์และมิวเตชัน เพื่อให้ตารางสอบที่ได้ยังเป็นคำตอบที่เป็นไปได้ แต่ไม่ทำให้ค่าความเหมาะสมลดลงและยังไม่ต้องใช้กระบวนการซ่อมแซม [14] และยังได้มีการออกแบบและประยุกต์ระบบคอมพิวเตอร์ เพื่อใช้ในการจัดตารางสอบและตารางสอน ซึ่งจะเน้นทำให้การจัดตารางมีความยืดหยุ่น [15] แต่อย่างไรก็ตามการจัดตารางสอบก็ยังคงมีปัญหาและความยากในการแก้ไขปัญหาที่แตกต่างกับการจัดตารางสอน เนื่องจากการจัดตารางสอบขึ้นเป็นครั้งคราว แต่การจัดตารางสอนจะต้องจัดเป็นรายสัปดาห์ จึงทำให้มีสิ่งที่ต้องพิจารณามากกว่าการจัดตารางสอบ จึงได้มีการนำวิธีการของเจเนติกอัลกอริทึมในใช้ในการแก้ไขปัญหาการจัดตารางสอน โดยการสร้างคำตอบยังคงอาศัยวิธีการสร้างคำตอบเบื้องต้นให้เป็นไปได้ทั้งหมด และยังคงมีการครอสโอเวอร์

และมิวเตชัน แต่เนื่องจากปัญหาที่ต่างกันจึงทำให้ตารางสอนที่ผ่านกระบวนการเหล่านี้กลายเป็นตารางสอนที่เป็นไปไม่ได้ ไม่สามารถนำไปใช้งานได้จริง อีกทั้งยังไม่มีกระบวนการในการซ่อมแซมคำตอบจึงทำให้ใช้เวลานานมากในการหาคำตอบที่เหมาะสม ดังนั้น W. Erben and J. Keppler [16] จึงได้เสนอแนะว่าควรมีการทดลองโดยขั้นตอนวิธีแบบขนาน (parallel algorithm) [17] และปรับปรุงวิธีการของเจเนติกอัลกอริทึม ตลอดจนควรศึกษาการใช้ local search หรือการเก็บคำตอบที่ดี (elitism) และได้มีการคิดค้นกระบวนการซ่อมแซมคำตอบเพื่อแก้ไขตารางสอนที่เป็นไปไม่ได้ให้กลับมาเป็นตารางที่เป็นไปได้ แต่อย่างไรก็ตามก็ยังคงเป็นการแก้ไขปัญหาการจัดตารางสอนที่มีขนาดเล็ก เป็นการจัดตารางสอนในระดับโรงเรียนมัธยมเท่านั้น ซึ่งมีจำนวนอาจารย์ผู้สอนไม่มากนัก กลุ่มนักเรียนมีน้อย และมีห้องประจำสำหรับนักเรียนแต่ละกลุ่ม [18-20] จึงได้มีการนำเจเนติกอัลกอริทึมมาประยุกต์ในการจัดตารางสอนในมหาวิทยาลัย โดยที่การปรับปรุงกระบวนครอสโอเวอร์และมิวเตชัน ซึ่งสามารถช่วยลดเวลาที่ใช้ในการจัดตารางสอน โดยข้อมูลที่ใช้ในการทดสอบประกอบด้วย กลุ่มผู้เรียน 10 กลุ่ม วิชาเรียน 90 วิชา อาจารย์ 26 คน ซึ่งถึงแม้ว่าจะเป็นการจัดตารางสอนในระดับมหาวิทยาลัย แต่ยังคงใช้ข้อมูลตัวอย่างขนาดเล็ก

นอกจากนี้ยังมีงานวิจัยที่ศึกษาเปรียบเทียบผลการทำงานของวิธีการต่างๆ เช่น N. A. Zahra [21] ได้ทำการศึกษาเปรียบเทียบผลการทำงานของวิธีการจำลองแอลเนลลิ่ง วิธีการค้นหาแบบทาบู วิธีการค้นหาแบบระบบอาณาจักรมด และวิธีการของเจเนติกอัลกอริทึมในการแก้ปัญหาโดยคำตอบเบื้องต้นของวิธีการวิธีการจำลองแอลเนลลิ่ง วิธีการค้นหาแบบทาบูและ และวิธีการของเจเนติกอัลกอริทึมจะสร้างจากการสุ่มทั้งหมด แต่วิธีการค้นหาแบบระบบอาณาจักรมดจะมีเกณฑ์ในการสร้างคำตอบ ดังนั้นวิธีการค้นหาแบบระบบอาณาจักรมดจึงมีคำตอบเบื้องต้นดีที่สุด เมื่อประมวลผลจนเสร็จ พบว่าหากพิจารณาตามความสามารถในการหาคำตอบวิธีการค้นหาแบบระบบอาณาจักรมดให้คำตอบที่ดีที่สุด รองมาคือวิธีการของเจเนติกอัลกอริทึม วิธีการค้นหาแบบทาบูและวิธีการจำลองแอลเนลลิ่งแต่หากพิจารณาตามความสามารถในการลดค่าความฟังก์ชันต้นทุนโดยเปรียบเทียบในช่วงเวลาเดียวกันพบว่าวิธีการค้นหาแบบทาบูดีที่สุด และ P. Pongcharoen [22] ได้พัฒนาโปรแกรมจัดตารางสอนโดยใช้วิธี Stochastic optimization ซึ่งเป็นวิธีการที่นำวิธีการของเจเนติกอัลกอริทึมรวมกับวิธีการจำลองแอลเนลลิ่ง และได้เปรียบเทียบประสิทธิภาพการจัดตารางสอนโดยใช้วิธีการของเจเนติกอัลกอริทึมรวมกับวิธีการจำลองแอลเนลลิ่ง พบว่าขั้นตอนวิธีทั้งสองแบบสามารถแก้ไขปัญหาการจัดตารางสอนได้เป็นอย่างดี แต่จากการประเมินค่าตารางสอนที่จัดได้จากวิธีการจำลองแอลเนลลิ่ง จะให้ผลลัพธ์ที่ดีกว่าจากวิธีการของเจเนติกอัลกอริทึมเพียงเล็กน้อย ในขณะที่วิธีการของเจเนติกอัลกอริทึมจะประมวลผลได้เร็วกว่าวิธีการจำลองแอลเนลลิ่ง

จากการสำรวจงานวิจัยที่เกี่ยวข้องแสดงให้เห็นว่า วิธีการของเจเนติกอัลกอริทึมเป็นวิธีการหนึ่งที่น่าสนใจ เนื่องจากเป็นวิธีการฮิวริสติกแบบหนึ่งซึ่งพัฒนามาจากวิธีการอื่นๆ โดยมีลักษณะการค้นหาคำตอบอย่างมีเหตุผลโดยเลียนแบบการถ่ายทอดทางพันธุกรรมตามธรรมชาติ ซึ่งได้รับการยอมรับถึงประสิทธิภาพในการแก้ปัญหาการจัดตารางสอน การประยุกต์ใช้เจเนติกอัลกอริทึมสำหรับการสร้างโปรแกรมเพื่อใช้สำหรับการจัดตารางสอน ส่วนใหญ่ผู้วิจัยจะทดลองเก็บตัวอย่างข้อมูลที่น้อย ตารางสอนที่มีขนาดเล็ก จำนวนโครโมโซมไม่มากนัก จำนวนรุ่นน้อย จึงทำให้ได้ตารางสอนที่ดี ใช้เวลาในการจัดตารางสอนน้อย ผลลัพธ์ที่ได้ก็เพียงแค่ตารางสอนของกลุ่มผู้เรียน และตารางการใช้ห้องเรียนเท่านั้น โปรแกรมที่สร้างขึ้นไม่ได้นำไปใช้ในการปฏิบัติงานจริงของเจ้าหน้าที่จัดตารางสอน อีกทั้งจะต้องใส่ข้อมูลเป็นจำนวนมาก ซึ่งบางข้อมูลอาจจะไม่ต้องใส่ทุกครั้งที่มีการเปลี่ยนแปลง และยังไม่มีการนำเจเนติกอัลกอริทึมใช้ในกรณีที่จัดตารางสอนรายวิชาปฏิบัติ โดยใช้ทรัพยากรร่วมกับการจัดตารางสอนรายวิชาบรรยาย ซึ่งเงื่อนไขและข้อจำกัดของการจัดตารางให้รายวิชาทั้ง 2 ประเภทนั้นต่างกัน

ดังนั้นงานวิจัยนี้จึงทำการศึกษาการประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งถือเป็นปัญหาที่มีขนาดใหญ่ และมีเงื่อนไขในการผลิตที่ซับซ้อนโดยการพิจารณาตารางสอนที่ดีที่สุดเป็นตารางสอนที่ละเมิดเงื่อนไขน้อยที่สุด มีรายวิชาที่ต้องพิจารณาในการจัดตารางสอน 2 ประเภทคือ รายวิชาบรรยายและปฏิบัติ ประกอบกับเป็นปัญหาที่มีขนาดใหญ่ มีการเปิดการเรียนการสอนทั้งหมด 28 สาขาวิชา ระดับปริญญาตรี 12 สาขาวิชา ระดับปริญญาโท 11 สาขาวิชา ปริญญาโทหลักสูตรพิเศษ 2 สาขาวิชา และระดับปริญญาเอก 8 สาขาวิชา จัดการเรียนการสอนภาคการศึกษาละประมาณ 350 รายวิชา 700 ตอน จำนวนนักศึกษาประมาณ 2800 คน มีอาจารย์สอนประมาณ 150 คน และมีห้องเรียนประมาณ 41 ห้องเรียน อีกทั้งเพื่อแก้ไขข้อผิดพลาดของตารางสอนที่จัดโดยเจ้าหน้าที่ เช่น เวลาของอาจารย์ซ้ำซ้อนกัน เวลาของนักศึกษาซ้ำซ้อนกัน และเวลาการใช้ห้องเรียนซ้ำซ้อนกัน โดยมีสมมุติฐานสาเหตุของข้อผิดพลาดดังนี้

1. เวลาของอาจารย์ซ้ำซ้อนกันเกิดจากในบางรายวิชามีอาจารย์สอนร่วมกันหลายคน เช่น วิชาปฏิบัติ
2. เวลาของนักศึกษาซ้ำซ้อนกันเกิดจากการกำหนดกลุ่มนักศึกษาที่ซ้ำซ้อนกัน เช่น กลุ่ม 2EE แบ่งออกเป็น 2EE(A) และ 2EE(B)
3. มีการกำหนดภาระงานสอนของอาจารย์มากจนเกินไปโดยเฉพาะในรายวิชาปฏิบัติที่แบ่งเป็นจำนวนตอนย่อยในการลงปฏิบัติการของนักศึกษาในแต่ละสัปดาห์

จากสมมุติฐานดังกล่าวข้างต้นนั้นสามารถแก้ไขได้โดยการตรวจสอบข้อมูลที่ใช้ในการจัดตารางสอนก่อนโดยละเอียดเพื่อให้ตารางสอนที่ได้จากการพัฒนาโปรแกรมเป็นตารางสอนที่ดี สามารถนำไปใช้งานได้จริง

**1.3 วัตถุประสงค์ของการวิจัย**

การดำเนินงานวิจัยครั้งนี้มีวัตถุประสงค์ คือ เพื่อสร้างโปรแกรมคอมพิวเตอร์สำหรับจัดตารางสอนที่เหมาะสมของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

**1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย**

หากงานวิจัยครั้งนี้เสร็จสมบูรณ์คาดว่าจะได้รับประโยชน์ คือ

1. ทำให้การดำเนินการจัดตารางสอนเป็นไปด้วยความสะดวก และมีประสิทธิภาพ

**1.5 ขอบเขตการวิจัย**

ออกแบบและสร้างโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนทุกหลักสูตรที่เปิดสอนของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ โดยพิจารณาตามกลุ่มนักศึกษาที่ลงทะเบียนเรียนตามแผนการเรียนเป็นหลัก ส่วนวิชานอกคณะฯ จะพิจารณาเฉพาะวัน-เวลาและกลุ่มผู้เรียนตามที่กองทะเบียนและประมวลผลได้จัดไว้

# บทที่ 2

# ทฤษฎีที่เกี่ยวข้องกับการวิจัย

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับปัญหาการจัดตารางสอน ซึ่งจะประกอบด้วยหัวข้อย่อยเรื่อง ความหมายและความสำคัญของปัญหาการจัดตารางสอน การกำหนดขอบเขตของปัญหาการจัดตารางสอน การกำหนดเงื่อนไขของปัญหาการจัดตารางสอนและวิธีการที่นำมาใช้แก้ปัญหาการจัดตารางสอน คำจำกัดความของปัญหา (problem) ลำดับขั้นการแก้ปัญหา (algorithm) และความยากง่ายของลำดับการแก้ปัญหา (complexity) ปัญหาแบบ P, NP, NP-Completeness, NP-Hard, Strongly NP-Hard การหาจุดเหมาะสมที่สุด เจเนติกอัลกอริทึมซึ่งประกอบด้วยเรื่อง พันธุศาสตร์กับเจเนติกอัลกอริทึมและวิธีการของเจเนติกอัลกอริทึม

## 2.1 ปัญหาการจัดตารางสอน (timetabling problem)

### 2.1.1 ความหมายและความสำคัญของปัญหาการจัดตารางสอน

การจัดตารางสอน คือ การกำหนดวัน เวลา สถานที่ วิชา อาจารย์ผู้สอน และกลุ่มนักศึกษา ที่จะทำการเรียนการสอนลงในช่วงเวลาที่เหมาะสม ภายใต้เงื่อนไขต่างๆ ให้เป็นแนวทางที่น่าพอใจหรือใกล้เคียงกับวัตถุประสงค์ที่ต้องการมากที่สุด [23] โดยจะกำหนดไว้เป็นรายสัปดาห์ เงื่อนไขสำคัญที่ใช้ในการจัดตารางสอน คือ ต้องไม่มีการซ้ำซ้อนของอาจารย์ผู้สอน กลุ่มนักศึกษา และห้องเรียน ในช่วงเวลาเดียวกัน ภาพประกอบ 2.1 แสดงตัวอย่างตารางสอนอย่างง่าย จากตัวอย่างแสดงให้เห็นว่าเป็นการจัดตารางสอนที่มีการซ้ำซ้อนกันของอาจารย์ผู้สอน คือ อาจารย์ผู้สอน 1 มีการสอนในคาบที่ 1 วันจันทร์ซ้ำซ้อนกัน โดยจัดให้สอนกลุ่มนักศึกษา 1 มีเรียนในห้องเรียน 5 และจัดให้สอนกลุ่มนักศึกษา 3 มีเรียนในห้องเรียน 3 ซึ่งตัวอย่างของตารางสอนนี้ไม่สามารถนำไปใช้งานได้จริง

ก่อนที่จะสามารถดำเนินการแก้ไขปัญหาการจัดตารางสอนได้นั้น สิ่งที่จำเป็นจะต้องรู้อย่างชัดเจนนั้นก็คือ ขอบเขตของปัญหา ซึ่งการกำหนดขอบเขตของปัญหาการจัด

ตารางสอนของแต่ละสถานศึกษาจะมีความแตกต่างกันไป เนื่องจากแต่ละสถานศึกษามีระบบการจัดการเรียนการสอนที่ไม่เหมือนกัน ดังจะกล่าวในหัวข้อต่อไป

| | คาบที่ 1 | คาบที่ 2 | คาบที่ 3 | คาบที่ 4 |
|---|---|---|---|---|
| วันจันทร์ | อาจารย์ 1, กลุ่มนักศึกษา 1, ห้องเรียน 5 | | | |
| วันอังคาร | | | | |
| วันพุธ | | | | |
| วันพฤหัสบดี | | | | |
| วันศุกร์ | | | | |

| | คาบที่ 1 | คาบที่ 2 | คาบที่ 3 | คาบที่ 4 |
|---|---|---|---|---|
| วันจันทร์ | อาจารย์ 1, กลุ่มนักศึกษา 3, ห้องเรียน 3 | | | |
| วันอังคาร | อาจารย์ 2, กลุ่มนักศึกษา 1, ห้องเรียน 4 | | | |
| วันพุธ | | | | |
| วันพฤหัสบดี | | | | |
| วันศุกร์ | | | | |

ภาพประกอบ 2.1 ตัวอย่างตารางสอนอย่างง่าย

### 2.1.2 การกำหนดขอบเขตของปัญหาการจัดตารางสอน

ระบบการศึกษาของแต่ละสถานศึกษาจะมีระบบการศึกษาที่ไม่เหมือนกัน ทำให้มีโครงสร้างตารางสอนที่แตกต่างกันออกไป [24] ด้วยเหตุนี้จึงส่งผลให้การกำหนดขอบเขตของปัญหาของการจัดตารางสอนของแต่ละสถานศึกษาแตกต่างกันออกไปตามระบบการศึกษา ตามการจัดการเรียนการสอน เช่น การกำหนดจำนวนวัน จำนวนคาบเรียนต่อวัน เป็นต้น ในการจัดตารางสอนของแต่ละสถานศึกษานอกจากจะมีข้อมูลที่กำหนดไว้ตามระบบการศึกษา แล้วยังมีข้อมูลที่เปลี่ยนแปลงอยู่เสมอในแต่ละภาคการศึกษา เช่น ข้อมูลกลุ่มนักศึกษา ข้อมูลอาจารย์ผู้สอน ข้อมูลห้องเรียน ข้อมูลรายวิชาที่ใช้ในการจัดตารางสอน เป็นต้น [25] จึงทำให้การกำหนดขอบเขตของปัญหามีความแตกต่างกัน ซึ่งในงานวิจัยต่างๆ ได้มีการกำหนดขอบเขตของปัญหาในลักษณะที่แตกต่างกันออกไป เช่น กำหนดขอบเขตของปัญหาการจัดตารางสอนตามประเภทของสถานศึกษา คือ มหาวิทยาลัยและโรงเรียนมัธยมศึกษา [26] นอกจากนี้ยังมีการกำหนดขอบเขตของปัญหาตาม

ประเภทของตารางคือ กำหนดเป็นตารางสอนหรือตารางสอบ อย่างไรก็ตามแม้ว่าจะมีการกำหนดขอบเขตของปัญหาที่แตกต่างกันไปในแต่ละประเภท แต่เมื่อกำหนดขอบเขตของงานแล้ว ก่อนที่จะดำเนินการจัดตารางสอนได้นั้นจะต้องมีการกำหนดเงื่อนไขที่จะใช้สำหรับพิจารณาในการจัดตารางสอน ซึ่งจะกล่าวในหัวข้อถัดไป

### 2.1.3 การกำหนดเงื่อนไขของปัญหาการจัดตารางสอนและประเมินค่าตาราง

สาเหตุที่ทำให้การแก้ไขปัญหาการจัดตารางสอนยุ่งยากซับซ้อน เนื่องจากการจัดตารางสอนมีเงื่อนไขที่ต้องพิจารณาหลายเงื่อนไข [7] โดยเงื่อนไขนี้ได้ถูกจำแนกเป็น 2 ประเภทคือ เงื่อนไขบังคับ (Hard Constraints : HC ) และเงื่อนไขเพื่อความสมบูรณ์ (Soft Constraints : SC) [27]

เงื่อนไขบังคับพื้นฐานที่ต้องพิจารณาในการจัดตารางสอน โดยเป็นเงื่อนไขที่ต้องไม่เกิดขึ้นได้ในการจัดตารางสอน ตารางสอนที่ไม่มีการละเมิดเงื่อนไขบังคับเท่านั้นที่จะเรียกว่าเป็นคำตอบที่เป็นไปได้ เป็นตารางสอนที่เป็นไปได้ (feasible timetable) [28] หากคำตอบหรือตารางสอนใดที่เกิดการละเมิดเงื่อนไขบังคับเหล่านี้ก็จะเป็นคำตอบที่ไม่สามารถนำไปใช้ได้จริง หรือตารางสอนที่เป็นไปไม่ได้ (infeasible timetable)

เงื่อนไขเพื่อความสมบูรณ์เป็นเงื่อนไขที่ยอมให้เกิดขึ้นได้แต่ต้องเกิดขึ้นน้อยที่สุด ซึ่งจะช่วยปรับปรุงตารางสอนที่เป็นไปได้ หรือตารางสอนที่ผ่านเงื่อนไขบังคับให้เป็นตารางสอนที่น่าพึงใจของผู้ที่เกี่ยวข้องกับตารางสอนนั้น เช่น อาจารย์ผู้สอน หรือกลุ่มนักศึกษา

จากการกำหนดเงื่อนไขทั้งสองประเภทนั้น ตารางสอนที่สามารถนำไปใช้ได้จริงคือตารางสอนที่ไม่ละเมิดเงื่อนไขบังคับใดๆ เลย แต่ไม่จำเป็นต้องผ่านเงื่อนไขเพื่อความสมบูรณ์ทุกข้อ อาจจะมีการละเมิดเงื่อนไขเพื่อความสมบูรณ์บางข้อ ดังนั้นในงานวิจัยส่วนใหญ่มักจะคำนวณค่าความเหมาะสมจากฟังก์ชันค่าความเหมาะสม (fitness function) [29] จากการละเมิดเงื่อนไขเพื่อความสมบูรณ์ไปใช้ในการประเมินค่าตารางสอนที่ได้ว่าดีเพียงใด [30] ตารางสอนที่ดีจะต้องมีการละเมิดเงื่อนไขเพื่อความสมบูรณ์น้อยที่สุด โดยจะมีการกำหนดค่าน้ำหนัก (weight) ให้เงื่อนไขเพื่อความสมบูรณ์แต่ละข้อแตกต่างกันออกไปตามความสำคัญมากน้อย ทั้งนี้การกำหนดค่าน้ำหนักให้กับเงื่อนไขนั้นจะขึ้นอยู่กับระบบการศึกษาของแต่ละสถานศึกษา และความพึงพอใจของอาจารย์ผู้สอนและกลุ่มนักศึกษาด้วย

### 2.1.4 วิธีการที่นำมาใช้แก้ปัญหาการจัดตารางสอน

กระบวนการจัดตารางสอนเป็นกระบวนการที่ก่อให้เกิดปัญหาการจัดตารางสอนเป็นประจำในทุกภาคการศึกษา สถานศึกษาแต่ละแห่งจึงได้มีความพยายามให้ในการคิดค้นวิธีการแก้ไขปัญหาหลายวิธีการเป็นต้นว่า สถานศึกษาบางแห่งอาจจะใช้วิธีการจัดตารางสอนด้วยมือ (manual) โดยผู้จัดตารางสอนจะต้องเป็นบุคคลที่มีความเชี่ยวชาญ หรือมีประสบการณ์ในการจัดตารางเป็นอย่างมาก บางแห่งได้มีการนำโปรแกรมคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูล บางแห่งได้ใช้โปรแกรมประยุกต์ เช่น CELCAT, PC-UNIT 3.0 มาใช้จัดตารางสอน หรือโปรแกรมอื่นๆ ที่พัฒนาขึ้นมา แต่อย่างไรก็ตามโปรแกรมประยุกต์ส่วนใหญ่ที่ถูกพัฒนาขึ้นมานั้นจะสามารถใช้ได้เฉพาะสถานศึกษาแต่ละแห่งเท่านั้น ยังไม่มีโปรแกรมประยุกต์ใดที่สามารถจัดตารางสอนได้ตามข้อกำหนดของแต่ละสถานศึกษาได้ทั้งหมด ดังนั้นวิธีการจัดตารางสอนด้วยมือจึงเป็นวิธีที่แพร่หลายอยู่มาก ถึงแม้ว่าจะเป็นวิธีที่ใช้เวลาในการจัดตารางสอนมาก

## 2.2 คำจำกัดความของปัญหา (problem) ลำดับขั้นการแก้ปัญหา (algorithm) และความยากง่ายของลำดับการแก้ปัญหา (complexity)

ปัญหา (problem) หมายถึง สถานการณ์ คน หรือ สิ่งที่ต้องการการเอาใจใส่ การจัดการ หรือการแก้ปัญหานั้น โดยที่บุคคลไม่สามารถนำความรู้ หรือประสบการณ์เดิมมาใช้แก้ปัญหาในสถานการณ์ต่างๆ ได้ หรืออีกนัยหนึ่งสามารถกล่าวได้ว่า ปัญหาคือคำถามที่ต้องการคำตอบ ซึ่งปัญหาเกิดขึ้นได้ทุกที่ทุกเวลา และเมื่อมีปัญหาย่อมมีการแก้ปัญหาและการตัดสินใจ

ปัญหาการตัดสินใจ (decision problem) คือ ปัญหาที่ต้องทำการตัดสินใจ เป็นปัญหาที่สามารถใช้ทฤษฎีการคำนวณได้และมีความซับซ้อนในการคำนวณ ปัญหาการตัดสินใจจะประกอบไปด้วย

1. ตัวแปรและค่าคงที่ต่างๆ (variables and parameters)
2. คุณสมบัติของคำตอบที่ต้องการจากปัญหานั้น (properties of the answer)

ปัญหาทั่วไปในงานทางด้านการวิจัยและดำเนินงานนั้นมีมากมาย เช่น ปัญหาการวางแผนการผลิต ปัญหาการวางแผนโลจิสติกส์ ปัญหารการขนส่ง เป็นต้น ซึ่งสามารถนำปัญหาที่เป็นปัญหามาตรฐานมาใช้ในการพัฒนาวิธีการหาคำตอบใหม่ๆ ให้มีประสิทธิภาพสูงขึ้น ในขณะที่เวลาในการหาคำตอบลดลงเรื่อยๆ การวัดประสิทธิภาพของวิธีการหาคำตอบนั้นสามารถวัดได้จากคุณภาพของคำตอบและเวลาที่ใช้ในการหาคำตอบ

ลำดับขั้นการแก้ปัญหา (algorithm) หมายถึง ขั้นตอนวิธีการหาคำตอบที่ออกแบบขึ้นมาเพื่อใช้ในการแก้ปัญหาใดปัญหาหนึ่ง เช่น การเข้าชมภาพยนตร์ ปัญหา คือ จะทำอย่างไรเพื่อที่จะสามารถเข้าชมภาพยนตร์ในโรงภาพยนตร์ได้ ลำดับขั้นการแก้ปัญหาคือ

1. เริ่มจากการซื้อตั๋วเข้าชมภาพยนตร์ที่เคาวน์เตอร์ขายตั๋ว และเลือกที่นั่ง
2. นำตั๋วชมภาพยนตร์ดังกล่าวไปให้พนักงานตรวจตั๋วหน้าโรงภาพยนตร์
3. เข้าไปนั่งชมภาพยนตร์ในตำแหน่งของเก้าอี้ที่เลือกไว้

หลักการที่นำมาใช้ในการออกแบบลำดับขั้นการแก้ปัญหาหรืออัลกอริทึมอาจจะเป็นหลักการใดๆ ก็ได้ ทั้งนี้ขึ้นอยู่กับผู้ออกแบบที่จะสามารถคิดและออกแบบได้ตามประสบการณ์และความเหมาะสมที่จะสามารถนำมาใช้แก้ปัญหานั้นๆ ลำดับขั้นการแก้ปัญหาหรืออัลกอริทึมนั้นจะมีความยากง่ายแตกต่างกันไป ตามผู้ออกแบบวิธีการแต่ละคน ซึ่งความยากง่ายของลำดับขั้นการแก้ปัญหานี้เรียกว่า ความซับซ้อนของวิธีการ (complexity) ซึ่งความซับซ้อนในทางการวิจัยและดำเนินงานจะเป็นเวลาที่ใช้ในการคำนวณตามลำดับขั้นการแก้ปัญหานั้นๆ

## 2.3 ปัญหาแบบ P, NP, NP-Completeness, NP-Hard, Strongly NP-Hard

สำหรับความยากง่ายของลำดับขั้นการแก้ปัญหานั้นจะพิจารณาจากเวลาที่ใช้ในการแก้ปัญหานั้น ปัญหาบางปัญหาเมื่อขนาดของปัญหามีขนาดเล็ก ก็จะสามารถแก้ไขปัญหาได้โดยใช้เวลาไม่มากนัก แต่เมื่อขนาดของปัญหาเพิ่มมากขึ้น ก็อาจจะใช้เวลาในการแก้ไขนานมากขึ้นหรืออาจจะถึงขั้นไม่สามารถแก้ไขปัญหานั้นได้เลยด้วยคอมพิวเตอร์ PC ธรรมดา เช่น ปัญหาการจัดลำดับการผลิตหากมีงานที่ต้องจัดลำดับให้เพียง 10 งานจะใช้เวลาในการแก้ปัญหาเพียงสั้นๆ แต่เมื่อเพิ่มขนาดของปัญหาเป็น 100, 500 งาน หรือ 1,000 งาน อาจจะใช้เวลาในการคำนวณนานขึ้นหรืออาจจะไม่สามารถแก้ปัญหานั้นได้เลย

ลำดับการแก้ปัญหาที่มีความซับซ้อน หรือความยาก จะแบ่งเป็น 2 แบบ คือ แบบโพลิโนเมียล (polynomial) และแบบน็อนโพลิโนเมียลอัลกอริทึม (non-polynomial algorithm) ซึ่งทั้งสองแบบนี้ใช้ในการแก้ปัญหา เมื่อจำนวนหรือขนาดของปัญหาเพิ่มมากขึ้น ส่งผลให้ระยะเวลาในการคำนวณเพิ่มขึ้นด้วย แต่มีความแตกต่างกันที่ระยะเวลาในการคำนวณ ถ้าสมมุติให้ขนาดของปัญหาตัวอย่างเป็น n แบบโพลิโนเมียลจะใช้ระยะเวลาในการคำนวณเป็น $n^2$ แต่แบบน็อนโพลิโนเมียลจะมีระยะเวลาในการคำนวณเป็น $2^n$ เช่น ลำดับการแก้ปัญหาเพื่อหาลำดับการผลิตสินค้า 10 ชนิด หากวิธีการเป็นโพลิโนเมียลจะใช้เวลาในการคำนวณเป็น $10^2$ หรือ 100 แต่หากลำดับขั้นการ

คำนวณนั้นเป็นแบบน็อนโพลิโนเมียลจะใช้เวลาในการคำนวณเป็น $2^{10}$ ซึ่งมีค่าถึง 1024 จะเห็นได้ว่ามีความซับซ้อนทางเวลามากกว่าแบบโพลิโนเมียลถึง 10 เท่า

NP Problem หรือปัญหาเอ็นพี คือปัญหาใดๆ ที่ยังไม่มีวิธีการแก้ปัญหาที่มีความซับซ้อนทางเวลาเป็น Polynomial Time ในบรรดาปัญหาเอ็นพี สามารถแบ่งเป็นกลุ่มย่อยเป็นกลุ่มใหญ่ๆ ได้ 3 กลุ่ม คือ เอ็นพีคอมพลีต (NP-Complete), NP-Hard และสตรองลีเอ็นพีฮาร์ด (Strongly NP-Hard) หากสามารถคาดการณ์คำตอบของปัญหาไว้ แล้วต้องการตรวจสอบคำตอบนั้นว่าเป็นคำตอบที่ดีที่สุดหรือไม่ วิธีการคาดคะเนคำตอบนั้นเรียกว่า ปัญหาการตัดสินใจ หากมีวิธีในการตรวจสอบคำตอบ โดยวิธีการตรวจสอบนี้มีความซับซ้อนทางเวลาเป็นโพลิโนเมียล และวิธีการนั้นสามารถใช้ตรวจสอบปัญหา NP-Complete อื่นๆ ได้ในระดับ Time Complexity เดียวกัน ปัญหานั้นคือปัญหา NP-Complete หากเป็นปัญหาที่ไม่มีวิธีการตรวจสอบคำตอบที่เป็นโพลิโนเมียล แต่สามารถระบุขอบเขตของคำตอบที่เป็นไปได้ได้ ปัญหานั้นคือปัญหา NP-Hard แต่หากปัญหาใดไม่มีวิธีการตรวจสอบคำตอบ ถึงแม้จะมีขอบเขตของคำตอบ ปัญหาจะถูกให้เป็นปัญหาประเภท Strongly NP-Hard นั้นเอง

## 2.4 การหาจุดเหมาะสมที่สุด (optimization)

การหาจุดเหมาะสมที่สุด หรือการหาจุดที่ดีที่สุด จะเกี่ยวข้องกับงานต่างๆ โดยส่วนมากจะเป็นทางด้านวิศวกรรมศาสตร์ วิทยาศาสตร์ และการจัดการ เช่น การหาน้ำหนักต่ำสุดและความแข็งแกร่งสูงสุดในการออกแบบเครื่องบิน การหาเส้นทางที่สั้นที่สุดในการออกแบบโครงข่ายถนนเดินรถทางเดียว [10] การหาจุดที่เหมาะสมที่สุดสำหรับการวางแผนและการจัดการเวลา เป็นต้น วิธีการทางคณิตศาสตร์จึงถูกนำมาประยุกต์ใช้อย่างมากมายเพื่อหาจุดที่ดีที่สุด การหาจุดที่ดีที่สุดของวิธีการทางคณิตศาสตร์ ( mathematical programming) คือการหาค่า x ที่ทำให้ f(x) มีค่าสูงสุดหรือต่ำสุด

การหาจุดที่ดีที่สุด (global optimum) แบ่งออกเป็น 2 ลักษณะคือ การหาจุดสูงสุดเรียกว่า global maximum และการหาจุดต่ำสุด เรียกว่า global minimum ของฟังก์ชัน จุดสูงสุดหรือจุดต่ำสุดของฟังก์ชันที่ต้องการหาอาจจะมีด้วยกันหลายจุด จุดที่คล้ายกับจุดสูงสุดแต่ไม่ใช่จุดสูงสุดของฟังก์ชันเรียกว่า local maximum จุดที่คล้ายกับจุดต่ำสุดแต่ไม่ใช่จุดต่ำสุดของฟังก์ชันเรียกว่า local minimum ในการแยกจุดที่ดีที่สุดจริงๆ (global optimum) ออกจากจุดที่คล้ายกับจุดที่ดีที่สุด (local optimum) นั้นเป็นปัญหาที่ยากมาก ซึ่งวิธีการที่จะนำมาใช้ในการแก้ไขปัญหานี้มีด้วยกัน

หลายวิธี เช่น การหาค่าเหมาะสมแบบกลุ่มอนุภาค (particle swarm optimization) ระบบอาณาจักรมด (ant colony system) เจเนติกอัลกอริทึม (genetic algorithm) เป็นต้น

## 2.5 ฮิวริสติก (heuristic) และเมตาฮิวริสติก (metaheuristic)

เมื่อปัญหามีขนาดใหญ่ขึ้น จำนวนของคำตอบที่เป็นไปได้ก็จะมีเพิ่มขึ้นตามไปด้วย หากจะคำนวณหาคำตอบที่ดีที่สุดก็จะต้องใช้เวลามาก

ลำดับขั้นการแก้ปัญหา (algorithm) ที่ต่างกันก็จะใช้เวลาไม่เท่ากัน โดยการคำนวณหาคำตอบที่ดีที่สุดจะแบ่งออกเป็น 2 กลุ่มใหญ่ คือ วิธีการหาคำตอบที่ดีที่สุด (exact method) เช่น Simplex, Branch-and-Bound, Branch-and-Price, Column Generation เป็นต้น ซึ่งวิธีการเหล่านี้แพร่หลายในงานด้านการวิจัยและดำเนินงาน และวิธีที่สองคือ วิธีการหรือลำดับขั้นการแก้ปัญหาแบบฮิวริสติก ซึ่งวิธีนี้อาจจะไม่ได้คำตอบที่ดีที่สุด แต่ใช้เวลาในการคำนวณน้อยกว่าแบบวิธีแรกค่อนข้างมาก แต่ถึงแม้ว่าคำตอบที่ได้นั้นอาจจะไม่ใช่คำตอบที่ดีที่สุด แต่คุณภาพของคำตอบที่หาได้ก็ดีเพียงพอ และสามารถนำไปใช้งานได้จริง ซึ่งเป็นข้อดีของการใช้วิธีการฮิวริสติกในการแก้ไขปัญหาแทนการหาคำตอบที่ดีที่สุด

วิธีการฮิวริสติกเป็นวิธีการแบบอิสระที่สามารถสร้างกรรมวิธีหรือขั้นตอนใดๆ ก็ได้ ที่ออกแบบมาแล้วต้องใช้งานได้จริง การออกแบบวิธีการทางฮิวริสติกจะต้องออกแบบให้ใช้งานได้ง่าย และวิธีการหนึ่งที่น่าสนใจก็คือ เมตาฮิวริสติก ซึ่งหมายถึง ชุดลำดับขั้นการแก้ปัญหาแบบฮิวริสติกชนิดหนึ่งที่สามารถนำไปแก้ปัญหาได้หลากหลายปัญหาโดยใช้หลักการเดียวกัน หลักการทางเมตาฮิวริสติกได้รับความนิยมเนื่องจากคำตอบที่ได้มีคุณภาพดี แก้ปัญหาได้อย่างรวดเร็ว และสามารถนำไปใช้งานได้ง่าย

หลักการเบื้องต้นของเมตาฮิวริสติกมีดังนี้

1. เมตาฮิวริสติกเป็นวิธีการค้นหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้ (feasible region)
2. จุดประสงค์ของเมตาฮิวริสติกคือการหาคำตอบที่ดีที่สุดหรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดในระยะเวลาอันสั้น
3. วิธีการทางเมตาฮิวริสติกมีทั้งวิธีการที่ง่ายไม่ซับซ้อน เช่น การปรับปรุงคำตอบเฉพาะที่ (local search) และแบบที่ซับซ้อนมากกว่า เช่น วิธีระบบมด (ant system) วิธีการเชิงพันธุกรรม (genetic algorithm) วิธีการค้นหาต้องห้าม

(Tabu Search) และวิธีการเลียนแบบการอบอ่อน (simulated annealing) เป็นต้น

4. เมตาฮิวริสติกเป็นขั้นตอนการประมาณคำตอบ
5. เมตาฮิวริสติกอาจจะเกิดจากการรวบรวมจากหลากหลายเทคนิค เพื่อค้นหาคำตอบที่ดีที่สุดภายในพื้นที่คำตอบที่เป็นไปได้
6. เมตาฮิวริสติกมีระเบียบขั้นตอนมาตรฐานที่แน่นอน ถึงแม้ว่าในการนำวิธีการเมตาฮิวริสติกไปประยุกต์ใช้ในปัญหาที่แตกต่างกันจะมีรายละเอียดขั้นตอนย่อยแตกต่างกันไป แต่อย่างไรก็จะต้องดำเนินการตามขั้นตอนหลักของวิธีการเมตาฮิวริสติกดั้งเดิม
7. เมตาฮิวริสติกต้องสามารถประยุกต์ใช้ได้กับปัญหาที่หลากหลาย
8. เมตาฮิวริสติกอาจจะมีลักษณะเป็นคำบรรยายโดยย่อก็ได้ ซึ่งไม่จำเป็นต้องมีหลักการทางคณิตศาสตร์ก็ได้
9. เมตาฮิวริสติกในปัจจุบันนั้นจะใช้ความจำชั่วคราวมากขึ้นในการจำคำตอบเดิมเพื่อค้นหาคำตอบที่ไม่ซ้ำเดิมหรือแตกต่างไปจากเดิม เช่น วิธีการค้นหาต้องห้าม วิธีระบบมด

ปัจจุบันมีวิธีการเมตาฮิวริสติกอยู่มากมายให้เลือกตามความชอบและถนัด แต่ละวิธีการจะมีจุดดีและจุดด้อยที่แตกต่างกัน บางวิธีให้ผลลัพธ์ที่ดีแต่ใช้เวลาในการคำนวณนาน บางวิธีใช้เวลาน้อยแต่ให้ผลลัพธ์ด้อยกว่าวิธีการอื่นๆ วิธีการเมตาฮิวริสติกมีหลากหลายวิธีซึ่งมีวิธีการแบ่งดังนี้

1. แบบที่เกิดจากแรงบันดาลใจจากธรรมชาติ เช่น วิธีระบบมด (ant colony system) วิธีการเชิงพันธุกรรม (genetic algorithm) วิธีการลอกแบบ (memetic algorithm) วิธีการเลียนแบบการอบอ่อน (simulated annealing) และวิธีการที่ไม่ได้แรงบันดาลใจจากธรรมชาติ เช่น วิธีการค้นหาต้องห้าม (tabu Search) การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search) และวิธีการค้นหาจากคำตอบใกล้เคียงแบบมีเงื่อนไข (Variable Neighborthood Search) เป็นต้น
2. แบบใช้ประชากรหรือแบบไม่ใช้ประชากร (Population or Non Population Based Heuristic) วิธีการแบบใช้ประชากร ในหนึ่งรอบของการคำนวณจะมีมากกว่าหนึ่งคำตอบ เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม และวิธีการลอกแบบ ส่วนแบบไม่มีประชากร ในหนึ่งรอบการคำนวณจะมีเพียงคำตอบเดียว

เช่น วิธีการเลียนแบบการอบอ่อน วิธีการค้นหาต้องห้าม การค้นหาเฉพาะที่แบบวนรอบซ้ำ

3. แบบสมการเป้าหมายคงที่หรือไม่คงที่ (Dynamic or Static Objective Function) ซึ่งสามารถเปลี่ยนสมการเป้าหมายได้ในหนึ่งรอบการคำนวณ เพื่อให้ได้คำตอบใหม่ๆ เช่น การค้นหาคำตอบเฉพาะที่แบบมีการชี้นำ ส่วนกรณีที่ไม่มีการเปลี่ยนแปลงสมการเป้าหมาย เช่น วิธีระบบมด วิธีเชิงพันธุกรรม และวิธีการลอกแบบ เป็นต้น
4. แบบเปลี่ยนวิธีการหาคำตอบปัจจุบันคงที่ เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม วิธีการเลียนแบบการอบอ่อน ส่วนกรณีที่มีการเปลี่ยนวิธีการหาคำตอบที่ใกล้เคียง ได้แก่ วิธีการค้นหาจากคำตอบใกล้เคียงแบบมีเงื่อนไข
5. แบบมีและไม่มีหน่วยความจำ การใช้หน่วยความจำ จะเก็บคำตอบเพื่อเป็นข้อมูลในการหาคำตอบรอบถัดไป เช่น วิธีระบบมด วิธีการค้นหาต้องห้าม ส่วนวิธีที่ไม่ใช้หน่วยความจำ เช่น วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

แต่ในปัจจุบันได้มีการผสมผสานกันของการแบ่งประเภท ดังนั้นจึงไม่ได้มีการแบ่งที่แน่นอนมากนักสำหรับการออกแบบวิธีการเมตาฮิวริสติก

## 2.6 เจเนติกอัลกอริทึม (Genetic Algorithms: GA)

เจเนติกอัลกอริทึม ถูกคิดค้นและพัฒนามาจากกระบวนการทางพันธุกรรมของสิ่งมีชีวิตจากทฤษฎีวิวัฒนาการหรือทฤษฎีการอยู่รอดของสิ่งมีชีวิตของ Charles Darwin โดยในปีคริสต์ศักราช 1975 John Holland นักวิทยาศาสตร์สาขาวิทยาการคอมพิวเตอร์ และเพื่อนร่วมงานพร้อมด้วยนักศึกษาของมหาวิทยาลัยมิชิแกน ประเทศสหรัฐอเมริกา ได้ศึกษาเกี่ยวกับวิวัฒนาการทางธรรมชาติของสิ่งมีชีวิต ซึ่งเป็นกระบวนการทางชีววิทยาที่ประกอบด้วยการคัดเลือกทางธรรมชาติ (natural selection) นั่นคือ สิ่งมีชีวิตที่แข็งแรงกว่าย่อมมีโอกาสรอดกว่าสิ่งมีชีวิตที่อ่อนแอ หมายความว่า โครโมโซมที่มีลักษณะดีสามารถอยู่รอดได้มากกว่าโครโมโซมที่มีลักษณะด้อย โครโมโซมที่สามารถอยู่รอดจะทำการถ่ายทอดลักษณะที่ดีไปยังรุ่นถัดไปได้มากกว่า โดยผ่านกระบวนการต่างๆ เช่น การรวม (mating) การผลิตซ้ำ (reproduction) การคัดเลือก (selection) การสลับสายพันธุ์ และการกลายพันธุ์ ซึ่งพันธุกรรมที่ได้อาจจะเหมือนเดิม หรือเปลี่ยนแปลงไปเป็นรูปแบบที่แตกต่างจากเดิม เพื่อให้ได้พันธุกรรมที่แข็งแรงที่สุดที่สามารถอยู่รอดในทุกสภาวะ ดังนั้น

John Holland จึงได้นำแนวคิดดังกล่าวมาปรับใช้กับการแก้ไขปัญหาด้วยคอมพิวเตอร์เพื่อหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุด เรียกว่า เจเนติกอัลกอริทึม เพื่อปรับปรุงการหาคำตอบที่ดีขึ้น ซึ่งเป็นวิธีการแก้ปัญหาแบบหนึ่งที่ใช้ในการค้นหาเพื่อให้ได้จุดที่เหมาะสมที่สุด ที่เรียกว่า Optimum Points ทั้งนี้อาจจะเป็นจุดต่ำสุด (minimum point) หรือจุดสูงสุด (maximum point) ลักษณะการทำงานของเจเนติกอัลกอริทึมจะอยู่ในรูปแบบของการค้นหาคำตอบแบบขั้นตอนเชิงฮิวริสติก ลักษณะการค้นหาคำตอบจะเป็นแบบคู่ขนาน (Parallel Search) โดยคำตอบที่ได้จากการค้นหาคำตอบในรุ่นหนึ่งจะผ่านการปรับปรุง เพื่อนำไปสู่การค้นหาคำตอบที่ดีขึ้นในรุ่นถัดไป

วิธีการของเจเนติกอัลกอริทึมเป็นวิธีการที่มีความแตกต่างจากเทคนิคการหาคำตอบที่เหมาะสมที่สุดแบบมาตรฐานอื่นๆ นั้นคือเจเนติกอัลกอริทึมจะทำการค้นหาคำตอบในพื้นที่การค้นหาของตัวแปรตัดสินใจ (decision variable) ของปัญหา โดยที่ตัวแปรตัดสินใจจะถูกเข้ารหัสเป็นสายรหัส เจเนติกอัลกอริทึมจะทำการหาคำตอบจากหลายๆ จุดพื้นที่ที่เป็นไปได้ในการหาคำตอบ ทำให้โอกาสการค้นหาคำตอบที่เป็นค่าเหมาะสมที่สุดเฉพาะที่ (local optimum) ลดน้อยลง เจเนติกอัลกอริทึมจะใช้ข้อมูลที่เป็นผลที่เกิดจากค่าจุดประสงค์ (objective value) ในการกำหนดทิศทางในการหาคำตอบในพื้นที่การค้นหา ซึ่งเทคนิคอื่นจะใช้ข้อมูลที่ได้มาจากอนุพันธ์ (derivative) ของฟังก์ชันจุดประสงค์ (objective function) ในการกำหนดทิศทางในการหาคำตอบ เจเนติกอัลกอริทึมจะใช้กฎการส่งผ่านเชิงความน่าจะเป็น (probabilistic transmission rule) ในกระบวนการหาคำตอบ ซึ่งเทคนิคอื่นจะใช้กฎการส่งผ่านเชิงกำหนด (deterministic transmission rule) ในกระบวนการหาคำตอบ

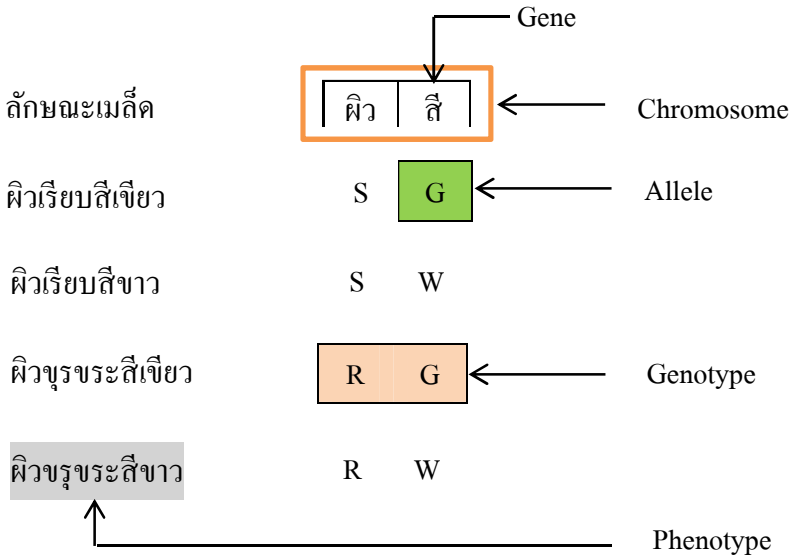
### 2.6.1 พันธุศาสตร์กับเจเนติก อัลกอริทึม

โครงสร้างของสิ่งมีชีวิตโดยธรรมชาติจะมีโครงสร้างและลักษณะที่แตกต่างกันไปตามลักษณะของโครโมโซม (chromosome) และสภาพแวดล้อมที่อยู่อาศัย ซึ่งจะมีส่งผลต่อความสามารถในการอยู่รอดของสิ่งมีชีวิตนั้นๆ การสืบพันธุ์ของสิ่งมีชีวิตจะถ่ายทอดลักษณะที่ดีไปยังรุ่นถัดไปเพื่อให้มีคุณสมบัติที่เหมาะสมในการมีชีวิตอยู่ ดังนั้นสิ่งมีชีวิตในแต่ละรุ่นก็จะมีวิวัฒนาการทางพันธุกรรมที่ดีขึ้น สามารถมีชีวิตอยู่รอดได้โดยมีลักษณะทางพันธุกรรมที่มีความเหมาะสมกับสภาพแวดล้อมมากที่สุด[26]

หน่วยที่ใช้เก็บลักษณะต่างๆ ภายในโครโมโซมของสิ่งมีชีวิตเรียกว่ายีน (gene) ซึ่งค่าที่แสดงลักษณะต่างๆ ของยีน เรียกว่า อัลลีล (allele) ลักษณะแต่ละชุดแบบของยีนเรียกว่าจีโนไทป์ (genotype) ซึ่งจะแสดงลักษณะภายนอกที่ปรากฏเรียกว่า ฟีโนไทป์ (phenotype) ตัวอย่าง

ลักษณะทางพันธุกรรมของเมล็ดถั่วดังภาพประกอบ 2.1 โครโมโซมของเมล็ดถั่วประกอบด้วยยีน 2 ชนิดที่เก็บลักษณะของผิวและสีของเมล็ดถั่วไว้ ค่าของยีนที่เก็บลักษณะผิวไว้จะเก็บลักษณะผิว 2 ลักษณะ คือ ผิวเรียบ (S) และผิวขรุขระ (R) และค่าของยีนที่เก็บลักษณะสีไว้ก็จะเก็บ 2 ลักษณะเช่นกัน คือ สีเขียว (G) และสีขาว (W) ซึ่งจากตัวอย่างจะเห็นลักษณะที่แสดงออกมาได้ 4 ลักษณะ คือ ผิวเรียบสีเขียว, ผิวเรียบสีขาว, ผิวขุรขระสีเขียว และผิวขรุขระสีขาว

หลักการทำงานของเจเนติกอัลกอริทึม คำตอบของปัญหาจะถูกกำหนดโดยใช้สายรหัสเพื่อแทนคุณลักษณะของตัวแปรตัดสินใจของปัญหาค่าเหมาะสมที่สุด โดยสายรหัสของคุณลักษณะเหล่านี้จะถูกเรียกว่าโครโมโซม โครโมโซมหนึ่งๆ จะประกอบด้วยกลุ่มของรหัสเรียกว่ายีน โดยที่ยีนหนึ่งๆ จะมีตำแหน่งอยู่บนโครโมโซมที่แน่นอน โดยที่ตำแหน่งของยีนนั้นจะเรียกว่าโลคัส และตามปกติยีนหนึ่งยีนจะมีค่าหรือสถานะที่เป็นไปได้หลายค่า แต่ละค่าที่เป็นไปได้นี้เรียกว่าอัลลีล ซึ่งเป็นตัวกำหนดลักษณะของตัวแปรตัดสินใจของคำตอบ ลักษณะของยีนที่ปรากฏอยู่ในโครโมโซมจะเรียกว่าจีโนไทป์ หลังจากถอดรหัสแล้วค่าที่ได้จะเป็นตัวแปรของการตัดสินใจเรียกว่าฟีโนไทป์




ภาพประกอบ 2.2 ตัวอย่างลักษณะทางพันธุกรรมของเมล็ดถั่ว

เนื่องจากเจเนติกอัลกอริทึมมีรากฐานมาจากทฤษฎีการวิวัฒนาการ จึงมีศัพท์เฉพาะทางเจเนติกอัลกอริทึมเกี่ยวกับด้านชีววิทยา ตามตาราง 2.1

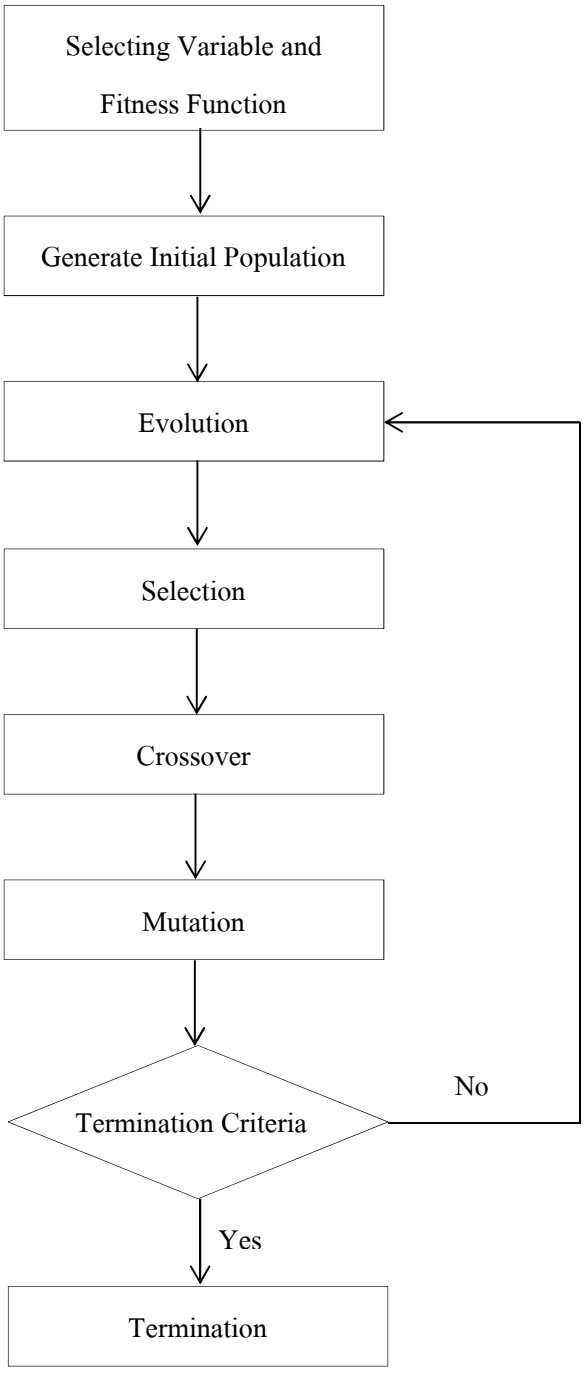
ตาราง 2.1 คำศัพท์และความหมายทางพันธุศาสตร์เปรียบเทียบกับเจเนติกอัลกอริทึม

| พันธุศาสตร์ | เจเนติกอัลกอริทึม | ความหมาย |
|---|---|---|
| โครโมโซม (chromosome) | สตริง (string) | สายรหัส |
| ยีน (gene) | บิท (character, bit) | สายรหัสหรือตัวอักษร |
| อัลลีล (allele) | ค่าของบิท (character value, bit value) | ค่าที่เป็นไปได้ในแต่ละตำแหน่งของสายรหัส |
| โลคัส (locus) | ตำแหน่ง (string position) | ตำแหน่งของรหัสบนสายรหัส |
| จีโนไทป์ (genotype) | โครงสร้าง (structure) | ลักษณะเฉพาะบนสายรหัส |
| ฟีโนไทป์ (phenotype) | โครงสร้างคำตอบ (a decode structure) | ตัวแปรของการตัดสินใจหลังการถอดรหัส |

การแก้ปัญหาด้วยวิธีการของเจเนติกอัลกอริทึม คำตอบที่สามารถเป็นไปได้ทั้งหมดของปัญหาจะถูกนำมาแปลงเป็นให้อยู่ในรูปแบบของโครโมโซม เพื่อนำโครโมโซมไปใช้ในกระบวนการเจเนติกอัลกอริทึม โดยจะใช้ฟังก์ชันความเหมาะสม (fitness function) เป็นตัวกำหนดค่าความเหมาะสมให้แต่ละโครโมโซม และพิจารณาโครโมโซมเหล่านั้นว่าโครโมโซมใดเป็นโครโมโซมที่ดีควรนำมาสืบสายพันธุ์ต่อไปหรือโครโมโซมใดเป็นโครโมโซมที่ด้อยไม่ควรนำมาสืบสายพันธุ์ ในขั้นตอนการหาคำตอบเบื้องต้นที่เป็นไปได้หรือโครโมโซมในแต่ละรุ่น (generations) จะใช้การสุ่มโครโมโซมขึ้นมาตามจำนวนประชากรเบื้องต้นที่กำหนดไว้ (population size) จึงทำให้คำตอบที่หาได้จะมีทั้งค่าสูงสุดหรือต่ำสุด ซึ่งเป็นการหาคำตอบที่เป็นไปได้อย่างสมบูรณ์ และเหมาะสมที่สุด เจเนติกอัลกอริทึม เป็นวิธีการหาคำตอบที่ช่วยในการหาคำตอบของปัญหาที่มีขนาดใหญ่และซับซ้อน มีตัวแปรและเงื่อนไขที่จะต้องพิจารณาเป็นจำนวนมาก เจเนติกอัลกอริทึมจะช่วยให้ได้คำตอบของปัญหาที่ใกล้เคียงคำตอบที่ดีที่สุด แต่อย่างไรก็ไม่สามารถที่จะประกันได้ว่าคำตอบที่หาได้นั้นเป็นคำตอบที่ดีที่สุดหรือไม่ เนื่องจากเจเนติกอัลกอริทึมมีคุณสมบัติการเลียนแบบการถ่ายทอดลักษณะทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่ดีที่สุดจากประชากรรุ่นก่อนมาใช้พิจารณาในการหาคำตอบของประชากรรุ่นถัดมา โดยผ่านตัวดำเนินการทางธรรมชาติ คือ การคัดเลือก (selection) การครอสโอเวอร์ (crossover) และการมิวเตชัน (mutation) โดยอาศัยการสุ่มเข้ามาใช้ในการหาคำตอบซึ่งจะช่วยให้มีความหลากหลาย (diversity) ในการหาคำตอบทุกบริเวณของคำตอบของปัญหา

### 2.6.2 วิธีการของเจเนติกอัลกอริทึม

ขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบทั่วไปสามารถแสดงได้ดังภาพประกอบ 2.3




ภาพประกอบ 2.3 ขั้นตอนการทำงานของวิธีการของเจเนติกอัลกอริทึม

ขั้นตอนเริ่มจากการกำหนดค่าต่างๆ ให้เรียบร้อยก่อน คือ การกำหนดฟังก์ชันจุดประสงค์ และการกำหนดรูปแบบโครโมโซม เมื่อกำหนดค่าต่างๆ เสร็จแล้วจะเข้าสู่กระบวนการทำงาน คือ การเข้ารหัสโครโมโซม การหาค่าความแข็งแรง การคัดเลือก การสลับสายพันธุ์ และการกลายพันธุ์

1. การกำหนดฟังก์ชันจุดประสงค์ (objective function) และการกำหนดรูปแบบโครโมโซม (chromosome encoding) [1] เมื่อกำหนดค่าต่างๆ เสร็จและจึงนำเข้าสู่กระบวนการทำงาน ซึ่งจะเริ่มโดยกล่าวถึงการกำหนดฟังก์ชันจุดประสงค์ก่อน

1.1 การกำหนดฟังก์ชันจุดประสงค์ (objective function) เป็นการกำหนดฟังก์ชันที่เกี่ยวกับเงื่อนไขต่าง ๆ ที่ต้องการขึ้นมา เพื่อใช้ในการหาค่าความแข็งแรงโดยสามารถแบ่งฟังก์ชันจุดประสงค์ได้เป็น 2 ประเภทคือ

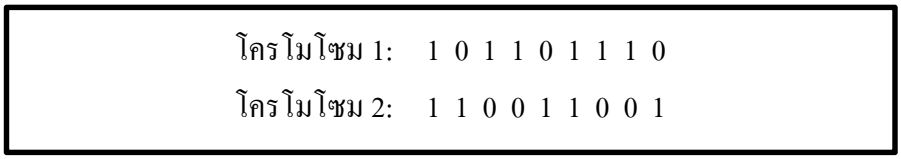
1.1.1 ฟังก์ชันแบบจุดประสงค์เดียว (single objective function) เป็นการกำหนดฟังก์ชันขึ้นมาหนึ่งฟังก์ชัน ที่ต้องการเพียงคำตอบเดียว ฟังก์ชันจุดประสงค์ประเภทนี้จะเหมาะสำหรับการแก้ปัญหาที่ไม่ค่อยยุ่งยากซับซ้อน และไม่ค่อยขัดแย้งกัน

1.1.2 ฟังก์ชันแบบหลายจุดประสงค์ (multi objectives function) เป็นการกำหนดฟังก์ชันขึ้นมาหลายๆ ฟังก์ชัน เพื่อที่จะต้องการหาคำตอบหลายๆ คำตอบ ซึ่งแต่ละคำตอบของแต่ละฟังก์ชันจะเป็นคู่แข่งกัน (candidate) โดยฟังก์ชันประเภทนี้จะเหมาะกับปัญหาที่มีความยุ่งยากซับซ้อนมาก และขัดแย้งกันมาก

ส่วนต่อไปจะกล่าวถึงการกำหนดรูปแบบโครโมโซมซึ่งถือว่าเป็นขั้นตอนที่สำคัญที่สุดสำหรับการแก้ปัญหาด้วยวิธีการของเจเนติกอัลกอริทึม เนื่องจากโครโมโซมเปรียบเสมือนคำตอบของปัญหาที่จะถูกนำไปปรับปรุง พัฒนาคำตอบให้ดีขึ้นต่อไป

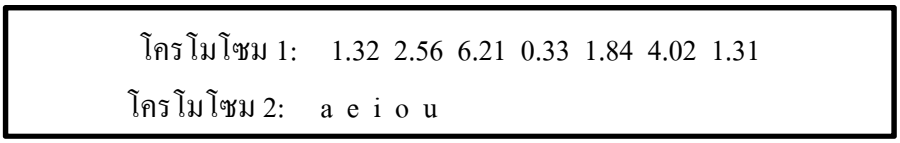
1.2 การกำหนดรูปแบบโครโมโซม (chromosome encoding) คือ การถอดรหัสหรือการคิด ออกแบบให้ได้มาซึ่งโครโมโซม ซึ่งเป็นสิ่งแรกที่จะเริ่มทำในการแก้ปัญหาโดยใช้ขั้นตอนวิธีการของเจเนติกอัลกอริทึม การถอดรหัสนั้นจะขึ้นอยู่กับลักษณะของปัญหาที่กำลังสนใจและต้องการจะแก้ไข ในปัจจุบันนั้นรูปแบบของโครโมโซมที่ออกแบบไว้สำหรับแก้ไขปัญหานั้นมีด้วยกันหลากหลาย ซึ่งมีความแตกต่างกันออกไปตามลักษณะของแต่ละปัญหานั้นๆ เช่น

1.2.1 การเข้ารหัสแบบเลขฐานสอง (binary encoding) เป็นรูปแบบโครโมโซมแบบแรกที่เริ่มมีการนำเข้ามาใช้แก้ปัญหาโดยขั้นตอนวิธีการของเจเนติกอัลกอริทึม ลักษณะของการเข้ารหัสแบบเลขฐานสอง คือ ทุกตำแหน่งของยีนในโครโมโซมจะมีค่าเป็น 0 หรือ 1 เท่านั้น ภาพประกอบ 2.4 แสดงตัวอย่างโครโมโซมที่มีการเข้ารหัสแบบเลขฐานสอง



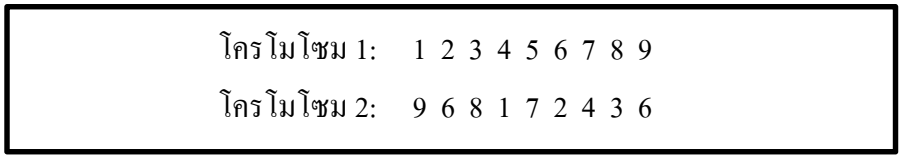

ภาพประกอบ 2.4 ตัวอย่างการเข้ารหัสแบบเลขฐานสอง

1.2.2 การเข้ารหัสแบบตัวเลข (value encoding) การเข้ารหัสแบบนี้โครโมโซมจะมีค่าที่สามารถบ่งบอกและเชื่อมโยงไปยังปัญหาได้ เช่น ตัวอักษร จำนวนจริง คำสั่ง หรืออื่นๆ ซึ่งรูปแบบโครโมโซมแบบนี้นิยมใช้ได้กับปัญหาที่ค่อนข้างซับซ้อนภาพประกอบ 2.5 แสดงตัวอย่างโครโมโซมที่มีการเข้ารหัสแบบตัวเลข




ภาพประกอบ 2.5 ตัวอย่างการเข้ารหัสแบบตัวเลข

1.2.3 การเข้ารหัสแบบเปลี่ยนลำดับ (permutation encoding) โครโมโซมรูปแบบนี้จะใช้ในการแก้ปัญหาการจัดลำดับ ทุกตำแหน่งของยีนของโครโมโซมจะเป็นค่าของจำนวนนับที่แทนตำแหน่งในลำดับ ซึ่งแต่ละโครโมโซมจะมีค่ายีนที่เหมือนกันทุกยีนในโครโมโซม แต่จะแตกต่างกันตรงลำดับตำแหน่งของค่ายีนแต่ละค่า ดังแสดงในภาพประกอบ 2.6 ซึ่งโครโมโซมทั้ง 2 นั้นมีค่ายีนแต่ละยีนเป็น 1-9 แต่ต่างกันที่การเรียงลำดับ 1-9 ของแต่ละโครโมโซม




ภาพประกอบ 2.6 ตัวอย่างการเข้ารหัสแบบเปลี่ยนลำดับ

เมื่อสามารถกำหนดรูปแบบโครโมโซมให้เหมาะสมกับลักษณะของปัญหาที่ต้องการแก้ไข ซึ่งรูปแบบของโครโมโซมแต่ละปัญหาจะแตกต่างกันไป แต่วิธีดำเนินการค้นหาคำตอบที่เหลือจะเหมือนกัน ดังนั้นกระบวนการต่อไปที่ควรทำหลังจากกำหนดรูปแบบโครโมโซมได้แล้ว คือการแปลงตัวแปรตัดสินใจของปัญหาให้เป็นแบบจำลองโครโมโซมได้ ปัญหานั้นก็สามารถหาคำตอบที่เหมาะสมได้ด้วยเทคนิคเจเนติกอัลกอริทึม ในการจำลองโครโมโซมนั้นมีขั้นตอนมีดังนี้ [33]

(1) การกำหนดความยาวของโครโมโซมตามตัวแปรที่ต้องการจะทราบคำตอบ เช่น โรงงานแห่งหนึ่งต้องการทราบจำนวนสินค้าที่ต้องการผลิตให้ได้กำไรเกิดขึ้นสูงสุด ซึ่งมีสินค้าที่ผลิตภายในโรงงาน 20 ชนิด แสดงว่าปัญหานี้มีตัวแปรตัดสินใจที่ต้องการทราบคำตอบ 20 ตัว ดังนั้นโครโมโซมที่เป็นคำตอบของปัญหานี้จึงมี 20 ตำแหน่ง เป็นต้น

(2) การกำหนดค่ารหัสแต่ละตำแหน่ง หรือกำหนดค่ายีนจากปัญหาในขั้นที่ 1 แสดงว่ายีนแต่ละตำแหน่งทั้ง 20 ตำแหน่ง สามารถกำหนดค่าให้เป็นแบบตัวเลขได้ เนื่องจากคำตอบของปัญหา คือจำนวนสินค้าที่ต้องการผลิต เป็นต้น

(3) การกำหนดค่าตำแหน่งแต่ละยีน ซึ่งแต่ละตำแหน่งยีนสามารถกำหนดแทนตัวแปรในปัญหาได้ เช่น ตำแหน่งที่ 1 แทนจำนวนสินค้าประเภทที่ 1 ตำแหน่งที่ 2 แทนจำนวนสินค้าประเภทที่ 2 เป็นต้น

(4) การกำหนดค่าของตัวแปรในยีน เพื่อบ่งบอกว่าค่าตัวแปรหรือคำตอบของตัวแปรตัดสินใจ ยกตัวอย่างเช่น หากว่าค่ารหัสที่ได้ในขั้นที่ 2 ณ ตำแหน่งโครโมโซมที่ 1 เป็นเลข 3 แสดงว่าค่าของตัวแปรที่ตำแหน่งที่ 1 หรือ จำนวนสินค้าประเภทที่ 1 ที่ต้องการผลิตมีค่า 3,000 ชิ้น เป็นต้น แต่ละตำแหน่งบนโครโมโซมค่ารหัสที่นำมาแปลงไม่จำเป็นต้องมีการปฏิบัติการทางคณิตศาสตร์ที่เหมือนกันก็ได้

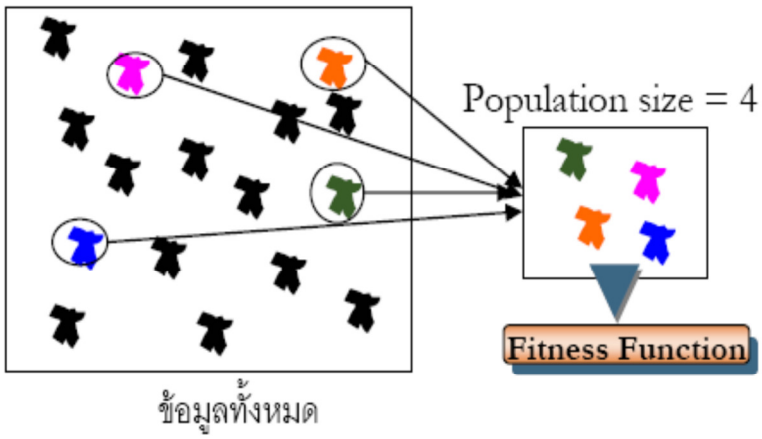
(5) การกำหนดชุดของตัวแปรตัดสินใจในบางปัญหา คำตอบที่ต้องการอาจจะเป็นไปได้หลายลักษณะ เช่น ในขั้นที่ 1 ค่าตัวแปรแต่ละตัวมีค่าเป็นค่าตัวเลข ซึ่งสามารถเปลี่ยนค่าตัวแปรเป็นตำแหน่งของแผนกต่างๆ ซึ่งการกำหนดรหัสตัวแปรอาจจะแตกต่างจากตัวแปรชุดที่ 1 ในการกำหนดชุดของตัวแปรตัดสินใจจะขึ้นอยู่กับลักษณะของปัญหาที่สนใจมากกว่าว่าต้องการหาคำตอบของอะไรบ้าง

(6) ค่าที่ได้จากรหัสตัวแปรทั้งหมด เป็นการหาคำตอบของตัวแปรทั้งหมดว่ามีคำตอบเป็นอะไรบ้าง เช่นตัวแปรทั้ง 20 ตัว ทั้งหมดมีค่าเป็นค่าใดบ้าง หรือตำแหน่งของแต่ละแผนกอยู่ที่ใดบ้าง เป็นต้น

เมื่อกำหนดฟังก์ชันจุดประสงค์และรูปแบบโครโมโซมให้กับปัญหาที่กำลังสนใจได้เรียบร้อยแล้ว ต่อมาก็จะต้องมีการสร้างประชากรเบื้องต้นตามจำนวนที่กำหนด โดยการสร้างโครโมโซมแต่ละโครโมโซมในกลุ่มประชากรเบื้องต้น (หรือประชากรรุ่นแรก) จะเป็นการสร้างทางเลือกที่สามารถเป็นไปได้ ซึ่งจะสร้างตามรูปแบบของโครโมโซมที่กำหนดไว้

2. การสร้างกลุ่มประชากรเบื้องต้น (initial population) เป็นการสร้างกลุ่มคำตอบที่เป็นไปได้กลุ่มแรกตามจำนวนที่กำหนดไว้ (population size) โดยการสุ่ม (random) โดยที่ยังไม่มีการสนใจค่าความเหมาะสมของแต่ละโครโมโซม ดังแสดงตัวอย่างในภาพประกอบ 2.7 ซึ่งเป็นการ

กระทำอันดับแรกก่อนที่จะนำกลุ่มประชากรเบื้องต้นดังกล่าวเข้าสู่กระบวนการของวิธีการของเจเนติกอัลกอริทึม




ภาพประกอบ 2.7 การสุ่มหากลุ่มประชากรเบื้องต้นจำนวน 4 โครโมโซม

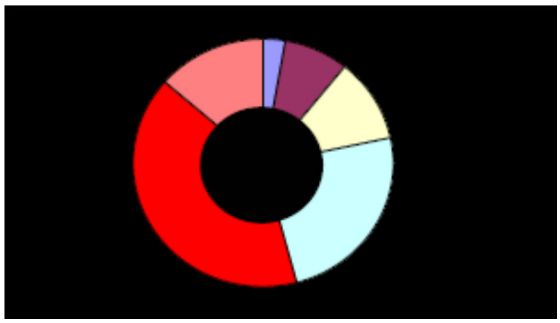
3. ค่าความเหมาะสม (fitness function) คือตัวที่ใช้ประเมินว่าแต่ละเส้นทางเลือก (solution) นั้น มีความเหมาะสม หรือสามารถใช้แก้ปัญหาได้ดีหรือไม่ โครโมโซมทุกตัวจะต้องถูกนำมาคำนวณหาค่าความเหมาะสม ที่จะบอกถึงความเหมาะสมว่าสมควรนำไปสืบสายพันธุ์ต่อหรือไม่ โดยใช้สมการหาค่าความเหมาะที่สอดคล้องกับปัญหา

4. การดำเนินการทางพันธุกรรม (genetic operator) การดำเนินการทางพันธุกรรมเป็นหัวใจสำคัญของขั้นตอนวิธีการของเจเนติกอัลกอริทึม ซึ่งมีกระบวนการพื้นฐานที่สำคัญด้วยกัน 3 ส่วน ดังนี้

4.1 การคัดเลือก เป็นการคัดเลือกโครโมโซมเพื่อที่จะนำมาเป็น พ่อแม่ (parent) ในการสืบสายพันธุ์ ซึ่งจะต้องคัดเลือกโครโมโซมที่น่าพอใจ เป็นโครโมโซมที่ดี เพื่อให้เกิดการอยู่รอดของสิ่งมีชีวิตตามทฤษฎีของชาร์ลีส ดาร์วิน (Charles Darwin) ด้วยเหตุนี้จึงทำให้เกิดรูปแบบในการเลือกโครโมโซมอย่างมากมาย เพื่อให้ได้โครโมโซมที่น่าพอใจที่สุดเพื่อนำไปสืบสายพันธุ์ ดังนี้

(1) การคัดเลือกแบบวงล้อรูเล็ต (roulette wheel) มีหลักการคัดเลือกที่เกิดจากการเลียนแบบการเล่นรูเล็ต หลักการทำงานคือ เริ่มต้นจากการกำหนดความกว้างของช่องแต่ละช่องของวงล้อรูเล็ตจากค่าความเหมาะสมของโครโมโซมแต่ละตัว ซึ่งโครโมโซมแต่ละตัวจะมีค่าความเหมาะสมไม่เท่ากันอยู่แล้ว จึงส่งผลทำให้เป็นวงล้อรูเลตมีความลำเอียง (bias roulette wheel) นั้นคือถ้ามีโครโมโซมของกลุ่มประชากรตัวใดที่มีค่าความเหมาะสมสูงจะมีโอกาสในการถูกเลือกซ้ำหลายครั้ง ทำให้กลุ่มประชากรรุ่นถัดไปมีโครโมโซมที่มีค่าความเหมาะสมสูงหลายๆ ตัว แต่ก็มิได้

หมายความว่าโครโมโซมที่มีค่าความเหมาะสมน้อยจะไม่ถูกเลือกไปเลย เพียงแต่มีโอกาสในการถูกเลือกน้อยเท่านั้น ดังแสดงพื้นที่ที่แทนโอกาสที่จะถูกเลือกของแต่ละโครโมโซมในภาพประกอบ 2.8 จากนั้นทำการกำหนดตัวชี้ตำแหน่งตายตัว (fixed point) และทำการหมุนวงล้อรูเล็ต เมื่อวงล้อหยุดหมุนจะเลือกสมาชิกของกลุ่มประชากรที่มีตัวชี้ตำแหน่งชี้อยู่ ทำเช่นนี้ซ้ำจนได้สมาชิกของกลุ่มประชากรครบตามจำนวนในหนึ่งรุ่น



ภาพประกอบ 2.8 สัดส่วนของค่าความเหมาะสมของโครโมโซม

(2) การเลือกสุ่มตัวอย่างแบบเฟ้นสุ่มสากล (stochastic universal sampling selection) มีหลักการคัดเลือกเหมือนกับการคัดเลือกแบบวงล้อรูเล็ต ต่างกันที่หลังจากกำหนดจุดชี้ตำแหน่งโดยการสุ่มในครั้งแรกแล้ว ทำการเลือกสมาชิกของกลุ่มประชากรที่มีตัวชี้ตำแหน่งชี้อยู่เป็นตัวแรก ต่อจากนั้นทำการเลื่อนตัวชี้ตำแหน่งจากจุดเดิมทีละขั้น โดยที่แต่ละขั้นนั้นจะเท่ากับ 360 องศา ต่อจำนวนสมาชิกของกลุ่มประชากร แล้วทำการเลือกสมาชิกของกลุ่มประชากรที่มีตัวชี้ตำแหน่งชี้อยู่จนครบตามจำนวนสมาชิกของกลุ่มประชากรในหนึ่งรุ่น ดังนั้นการคัดเลือกพันธุ์แบบการสุ่มเลือกตัวอย่างแบบเฟ้นสุ่มสากลนี้สามารถลดความลำเอียงในการคัดเลือกได้ เนื่องจากโอกาสที่สมาชิกของกลุ่มประชากรตัวใดจะถูกเลือกซ้ำหลายๆ ครั้ง จะเกิดขึ้นต่อเมื่อสมาชิกของกลุ่มประชากรตัวนั้นๆ มีค่าความแข็งแรงสูงมากๆ

(3) การคัดเลือกแบบช่วง (ranking) คือ เลือกประชากรที่มีค่าความเหมาะสมที่ดีที่สุด โดยที่ไม่สนใจประชากรตัวอื่นเลย

(4) การคัดเลือกแบบ elitist เป็นแนวคิดที่ป้องกันการหายของเส้นทางที่ดีที่สุด หมายความว่า มีการคัดลอกโครโมโซมที่ดีที่สุดไว้ก่อน ส่วนประชากรส่วนที่เหลือจะที่จะต้องคัดเลือกจะใช้วิธีการเลือกแบบอื่นๆ

4.2 การสลับสายพันธุ์ (crossover) จะนำสมาชิกของประชากรที่ผ่านการคัดเลือกมาแล้วสองตัวกำหนดให้เป็นสมาชิกรุ่นพ่อกับสมาชิกรุ่นแม่ (parent individual) จากนั้นทำการแลกเปลี่ยนยีนระหว่างสมาชิกรุ่นพ่อกับสมาชิกรุ่นแม่ จะทำให้เกิดเป็นสมาชิกรุ่นลูก (offspring

individuals) สองตัว สมาชิกรุ่นนี้จะถูกนำไปเป็นสมาชิกของประชากรรุ่นถัดไป การสุ่มเลือกสมาชิกรุ่นพ่อกับสมาชิกรุ่นแม่มาทำการสลับสายพันธุ์จะถูกกำหนดโดยความน่าจะเป็นในการสลับสายพันธุ์ (Probability of Crossover : Pc) อัตราการครอสโอเวอร์นั้นเป็นพารามิเตอร์ที่สำคัญสำหรับการหาคำตอบของเจเนติกอัลกอริทึมซึ่งก็คืออัตราส่วนของจำนวนโครโมโซมลูกที่ถูกสร้างขึ้นในแต่ละรุ่นต่อขนาดของประชากร วิธีการสลับสายพันธุ์มีอยู่ด้วยกันหลายวิธี [1, 11] เช่น

One-Point Crossover การครอสโอเวอร์แบบจุดเดียว โดยกระบวนการจะเริ่มจากการสุ่มตำแหน่งที่จะครอสโอเวอร์ ($X_p$) ขึ้นมา 1 ตำแหน่ง ส่วนหัวหรือค่าในตำแหน่งแรกจนถึง $X_p$-1 ของโครโมโซมลูกแต่ละตัวจะได้มาจากค่าในตำแหน่งเดียวกันของโครโมโซมพ่อแม่ตัวหนึ่ง ส่วนหางของโครโมโซมลูก (ตำแหน่งที่ $X_p$+1) ขึ้นไปจะได้มาจากโครโมโซมพ่อแม่อีกตัวที่ถูกตัดงานที่ซ้ำกับงานที่อยู่ในส่วนหัวของโครโมโซมลูกตัวนั้นออกแล้ว

*N*-point crossover การกำหนดจุดและเปลี่ยนยีนระหว่างคู่โครโมโซมพ่อแม่เพื่อสร้างโครโมโซมลูกนั้น จะเกิดขึ้นที่ข้างใดข้างหนึ่งของตำแหน่งการแลกเปลี่ยนยีน และจะเกิดเฉพาะในช่วงของสองตำแหน่งของการแลกเปลี่ยนยีนที่อยู่บนโครโมโซม ซึ่งตำแหน่งการแลกเปลี่ยนยีนจะเกิดจากเลขสุ่มในช่วง [1, (ความยาวของโครโมโซม – 1)] โดยที่ความยาวของโครโมโซมนับจากจำนวนบิตที่ใช้ในการนำเสนอโครโมโซมและ n จะเป็นตัวกำหนดจำนวนของตำแหน่งของการแลกเปลี่ยน โดยที่ n ต้องมากกว่า 1

Uniform Crossover ในการสลับสายพันธุ์ด้วยวิธีนี้ ยีนในแต่ละโลคัสของสมาชิกรุ่นลูกตัวหนึ่งจะได้จากการสุ่มเลือกคู่ยีนในตำแหน่งที่ตรงกันของสมาชิกรุ่นพ่อแม่ทั้งสอง โดยกำหนดให้ความน่าจะเป็นในการเลือกยีนแต่ละตัวจากสมาชิกรุ่นพ่อแม่มีค่าเท่ากัน เมื่อยีนหนึ่งถูกเลือกให้สมาชิกรุ่นลูกตัวหนึ่งแล้ว ยีนที่เหลือก็จะถูกส่งผ่านไปให้สมาชิกรุ่นลูกอีกตัว การสลับสายพันธุ์แบบเอกรูปจะทำการและเปลี่ยนยีนเป็นอิสระต่อตำแหน่งของยีนนั้น ทำให้โอกาสที่แต่ละโลคัสจะถูกแลกเปลี่ยนยีนมีโอกาสเท่าๆ กัน ซึ่งเป็นการเพิ่มความสามารถในการสำรวจพื้นที่การค้นหา ซึ่งวิธีนี้เหมาะที่จะนำไปใช้กับกลุ่มประชากรที่มีขนาดเล็ก หรือใช้ในขณะที่สมาชิกในกลุ่มประชากรมีลักษณะคล้ายคลึงกันจนนำไปสู่การได้คำตอบที่ดีเฉพาะที่ การสลับสายพันธุ์แบบนี้จึงถือได้ว่าเป็นเครื่องมือหนึ่งที่ช่วยเพิ่มความแตกต่างในหมู่ประชากร

Partial-Mapped Crossover วิธีนี้จะเริ่มจากการสุ่มตำแหน่งที่จะครอสโอเวอร์ขึ้นมา 2 ตำแหน่ง ยีนที่อยู่ในช่วงของตำแหน่งทั้งสองจะเรียกว่าเป็นยีนย่อย โครโมโซมลูกจะได้จากโครโมโซมพ่อและแม่ที่ถูกแลกเปลี่ยนยีนย่อยกัน หลังการแลกเปลี่ยนถ้างานในโครโมโซมย่อยซ้ำกับงานตัวอื่น ให้แทนงานตัวอื่นที่ซ้ำด้วยงานที่อยู่ในตำแหน่งตรงกันของโครโมโซมอีกตัว

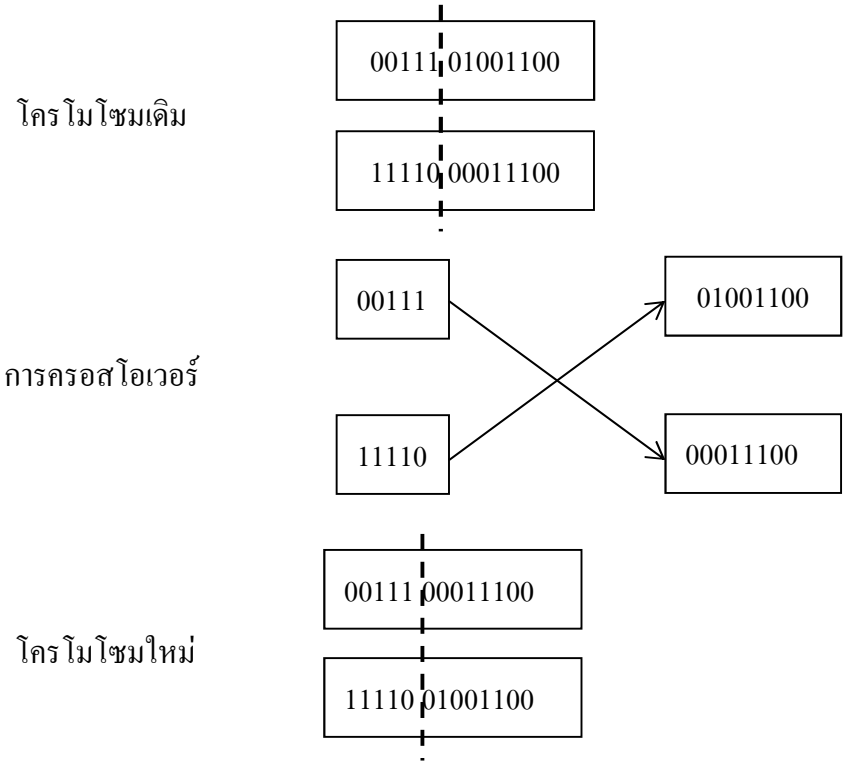
Order Crossover การครอสโอเวอร์แบบลำดับจะที่ต้องเลือกโครโมโซมย่อยอย่างสุ่มมาจากโครโมโซมพ่อแล้วคัดลอกลงไปยังโครโมโซมลูกเบื้องต้นในตำแหน่งเดียวกัน จากนั้นลบงานที่ปรากฏอยู่ในโครโมโซมลูกเบื้องต้นแล้วออกจากโครโมโซมแม่ จากนั้นนำงานที่เหลือในโครโมโซมแม่มาใส่ในตำแหน่งที่ยังว่างของโครโมโซมลูกเบื้องต้นตัวนั้นตามลำดับจากซ้ายไปขวา

Cycle Crossover วิธีนี้เริ่มต้นโดยพิจารณาว่าตำแหน่งเริ่มต้นของโครโมโซมพ่อมีค่าเท่าไร หากในตำแหน่งเดียวกันนี้ของโครโมโซมแม่มีค่าไม่เท่ากัน ให้คงค่าในตำแหน่งนี้ของโครโมโซมพ่อแม่ไว้ จากนั้นหาว่าตำแหน่งใดของโครโมโซมพ่อมีค่าเท่ากับค่าในโครโซมแม่ในตำแหน่งที่ผ่านมา แล้วพิจารณาว่าในตำแหน่งดังกล่าวของโครโมโซมแม่มีค่าเท่ากับค่าในตำแหน่งเริ่มต้นของโครโมโซมพ่อหรือไม่ หากมีค่าไม่เท่ากันให้คงค่าของโครโมโซมพ่อแม่ในตำแหน่งดังกล่าวไว้ และทำซ้ำขั้นตอนดังกล่าวจนพบตำแหน่งในโครโมโซมแม่ที่มีค่าเท่ากับค่าในตำแหน่งเริ่มต้นของโครโมโซมพ่อ จากนั้นให้ทำการสลับที่ระหว่างค่าในตำแหน่งที่ไม่ผ่านขั้นตอนขั้นต้นของโครโมโซมพ่อแม่

Position-Base Crossover เริ่มต้นจากการเลือกตำแหน่งครอสโอเวอร์จากโครโมโซมพ่ออย่างสุ่มแล้วนำค่าในตำแหน่งที่เลือกของโครโมโซมพ่อไปใส่ในตำแหน่งเดียวกันในโครโมโซมลูก ตัดค่าที่อยู่ตรงตำแหน่งครอสโอเวอร์ที่เลือกของโครโมโซมพ่อออกจากโครโมโซมแม่ นำค่าที่เหลืออยู่ในโครโมโซมแม่มาใส่ในโครโมโซมลูก

เทคนิคของการครอสโอเวอร์ส่วนใหญ่ขึ้นอยู่กับการใช้รูปแบบโครโมโซมแบบต่างๆ เช่น การครอสโอเวอร์ของโครโมโซมที่มีการเข้ารหัสแบบเลขฐานสองจะนิยมใช้วิธีการครอสโอเวอร์ที่มีลักษณะการสุ่มตำแหน่งการครอสโอเวอร์เพียงหนึ่งตำแหน่งหรือสองตำแหน่ง การครอสโอเวอร์ของโครโมโซมที่มีการเข้ารหัสแบบเปลี่ยนลำดับจะนิยมใช้วิธีการคัดลอกข้อมูลจากตำแหน่งแรกถึงตำแหน่งครอสโอเวอร์ของพ่อแม่ตัวแรก และอ่านข้อมูลจากพ่อแม่ตัวที่สองถ้าข้อมูลบางตัวที่ไม่เหมือนกับพ่อแม่ตัวแรกก็จะเพิ่มเข้าไปโดยมีรายละเอียดของวิธีการครอสโอเวอร์เป็นดังที่ได้กล่าวไปแล้ว ซึ่งการเลือกใช้วิธีการครอสโอเวอร์แบบต่างๆ นั้นขึ้นอยู่กับการนำไปประยุกต์ใช้ให้เหมาะสมกับปัญหา ตัวอย่างการครอสโอเวอร์แบบ One-Point Crossover เป็นดังภาพประกอบ 2.9 [35] การครอสโอเวอร์แบบ One-Point Crossover ของโครโมโซมพ่อและแม่โดยเลือกตำแหน่งสำหรับทำการครอสโอเวอร์เพียงตำแหน่งเดียวคือตำแหน่งที่ 5 ผลที่ได้จากการครอสโอเวอร์ด้วยวิธีนี้ คือจะได้โครโมโซมลูกหรือโครโมโซมใหม่ 2 ตัว โดยโครโมโซมใหม่ที่ 1 จะเกิดจากยีนในตำแหน่งก่อนหน้าของตำแหน่งที่ถูกเลือกในโครโมโซมพ่อ กับยีนในหลังตำแหน่งของตำแหน่งที่ถูกเลือกในโครโมโซมแม่ และโครโมโซมใหม่ตัวที่ 2 จะเกิดจากยีนในตำแหน่งก่อนหน้า

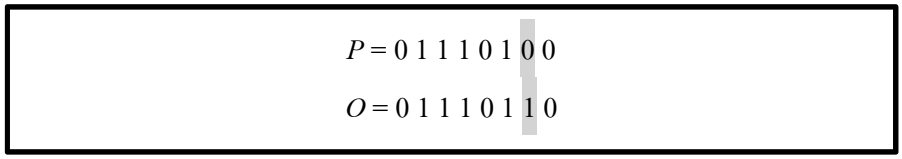
ของตำแหน่งที่ถูกเลือกในโครโมโซมแม่ กับยีนในหลังตำแหน่งของตำแหน่งที่ถูกเลือกในโครโมโซมพ่อ




ภาพประกอบ 2.9 การครอสโอเวอร์แบบ One-Point Crossover

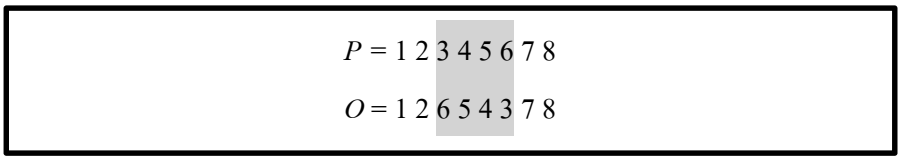
4.3 การกลายพันธุ์ (mutation) เป็นการแลกเปลี่ยนยีนภายในสมาชิกแต่ละตัวเท่านั้นจะไม่เกิดขึ้นกับสมาชิกทั้งหมดเป็นขั้นตอนที่อาจช่วยให้โครโมโซมมีค่าความเหมาะสมดีขึ้นหลังจากครอสโอเวอร์โดยการกลับค่าบางส่วนของโครโมโซมเป็นค่าใหม่ในตำแหน่งที่สุ่มได้ ตามอัตราความน่าจะเป็นในการมิวเตชัน (Probability of Mutation: Pm) ที่กำหนด อัตราการมิวเตชันหมายถึง เปอร์เซ็นต์ของจำนวนยีนทั้งหมดในประชากรที่จะเกิดการมิวเตชันขึ้น คำตอบที่ได้จากการกลายพันธุ์จะมีเพียงหนึ่งคำตอบที่มีลักษณะบางส่วนแตกต่างไปจากลักษณะต้นแบบ วิธีการมิวเตชันมีหลายวิธี เช่น

การกลายพันธุ์แบบกลับบิต (bit-flipped mutation) เป็นเทคนิคการกลายพันธุ์ที่ใช้กับโครโมโซมแบบเลขฐานสอง สามารถทำได้โดยการกลับค่าบิตเป็นค่าตรงกันข้ามจากค่าเดิม (complement) คือจาก 0 เป็น 1 หรือ 1 เป็น 0 จากตำแหน่งที่สุ่มได้ตามค่าความน่าจะเป็นของการกลายพันธุ์ ดังภาพประกอบ 2.10 กำหนดให้ *P* คือสมาชิกรุ่นพ่อแม่ และ *O* คือสมาชิกรุ่นลูก



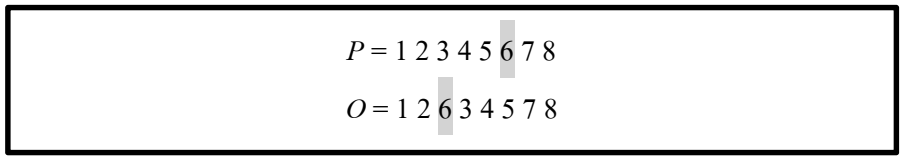

ภาพประกอบ 2.10 การกลายพันธุ์แบบกลับบิต

การกลายพันธุ์แบบผกผัน (inversion mutation) เป็นการสลับตำแหน่งแบบหลังไปหน้า โดยสุ่มเลือกสมาชิกมาหนึ่งตัว สุ่มเลือกช่วงที่จะทำการกลายพันธุ์และทำการสลับตำแหน่งภายในช่วงการตัด ดังภาพประกอบ 2.11 กำหนดให้ *P* คือสมาชิกรุ่นพ่อแม่ และ *O* คือสมาชิกรุ่นลูก



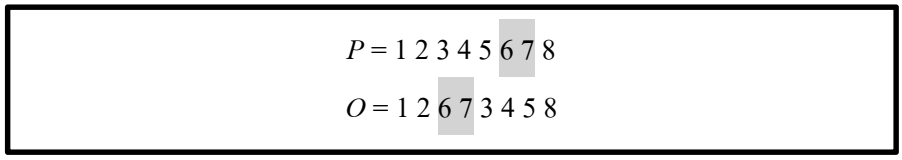

ภาพประกอบ 2.11 การกลายพันธุ์แบบผกผัน

การกลายพันธุ์แบบแทรก (insertion mutation) เป็นการเปลี่ยนตำแหน่งโดยการแทรกตำแหน่ง ซึ่งจะสุ่มเลือกสมาชิกมาหนึ่งตัว สุ่มเลือกช่วงที่ต้องการจะแทรกสุ่มเลือกยีนที่จะนำมาแทรก แล้วนำยีนที่สุ่มได้มาแทรกตรงที่สุ่มเลือกช่วงที่ต้องการแทรก ดังภาพประกอบ 2.12 กำหนดให้ *P* คือสมาชิกรุ่นพ่อแม่ และ *O* คือสมาชิกรุ่นลูก



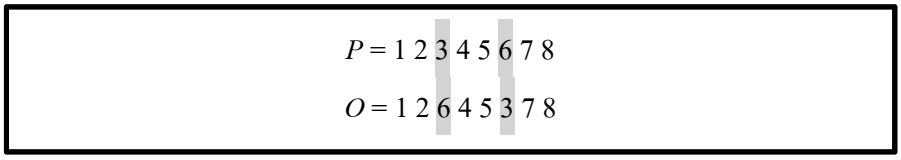

ภาพประกอบ 2.12 การกลายพันธุ์แบบแทรก

การกลายพันธุ์แบบเคลื่อนตำแหน่ง (displacement mutation) เป็นการเปลี่ยนตำแหน่งโดยแทรกเป็นช่วงของตำแหน่ง ซึ่งจะสุ่มเลือกสมาชิกมาหนึ่งตัว สุ่มเลือกตำแหน่งที่ต้องการจะแทรก สุ่มเลือกช่วงตำแหน่งยีนที่จะนำมาแทรก แล้วนำช่วงที่สุ่มได้ มาแทรกหน้าตำแหน่งที่สุ่มไว้ ดังภาพประกอบ 2.13 กำหนดให้ *P* คือสมาชิกรุ่นพ่อแม่ และ *O* คือสมาชิกรุ่นลูก




ภาพประกอบ 2.13 การกลายพันธุ์แบบเคลื่อนตำแหน่ง

การกลายพันธุ์แบบสลับตำแหน่ง (reciprocal exchange mutation) เป็นการเปลี่ยนตำแหน่งยีนสองยีน ซึ่งจะสุ่มเลือกสมาชิกมาหนึ่งตัว สุ่มเลือกตำแหน่งที่ต้องการจะสลับตำแหน่งยีนสองตำแหน่ง แล้วสลับตำแหน่งระหว่างยีนทั้งสองตัวที่สุ่มได้ ดังภาพประกอบ 2.14 กำหนดให้ *P* คือสมาชิกรุ่นพ่อแม่ และ *O* คือสมาชิกรุ่นลูก




ภาพประกอบ 2.14 การกลายพันธุ์แบบสลับตำแหน่ง

5. การกำหนดพารามิเตอร์ ปฏิบัติการและกลไกการทำงานของ GA

จากรายละเอียดลำดับขั้นตอนในการทำงานของ GA ที่เสนอในข้างต้น แสดงให้เห็นว่าพารามิเตอร์ ปฏิบัติการและกลไกในการทำงานของ GA มีหลากหลายประเภท ซึ่งโดยปกติแล้วการทำงานของ GA จะต้องมีการกำหนดค่าพารามิเตอร์ต่างๆ ได้แก่

ขนาดของประชากร ถ้ากำหนดจำนวนน้อยการทำงานของโปรแกรมจะทำได้เร็ว แต่คำตอบที่ได้อาจจะไม่ใกล้เคียงกับคำตอบที่แท้จริง ในทางกลับกันถ้ากำหนดจำนวนมาก โปรแกรมก็จะทำงานช้าแต่คำตอบที่ได้มีโอกาสที่จะเข้าใกล้คำตอบที่แท้จริงสูง

จำนวนรุ่น การกำหนดจำนวนรุ่นว่าจะทำให้กลไกการทำงานของโปรแกรมไปเรื่อยๆ จนได้จำนวนประชากรเท่ากับที่ต้องการ ความมากน้อยของจำนวนรุ่นมีผลเหมือนกับค่าจำนวนประชากรในแต่ละรุ่น เมื่อโปรแกรมทำงานได้จำนวนรุ่นที่ต้องการแล้ว โปรแกรมก็เลือกคำตอบที่ดีที่สุดจากทั้งหมดมา แล้วก็จะหยุดการทำงาน

ความน่าจะเป็นในการครอสโอเวอร์หรือความน่าจะเป็นของการสลับสายพันธุ์ จะมีค่าที่อยู่ในช่วง 0 – 100 จากการทดลองของนักวิทยาศาสตร์หลายท่านได้พบว่าความน่าจะเป็นของการ สลับสายพันธุ์ ส่วนใหญ่อยู่ที่ 60% - 95% ถ้าหากไม่มีการสลับสายพันธุ์ ( 0%) ผลที่ได้คือการทำสำเนาที่ถูกต้องจาก parent แต่ถ้ามีการสลับสายพันธุ์เกิดขึ้นบ่อยจะทำให้เกิดผลลัพธ์ที่

หลากหลาย บางปัญหาอาจจะแก้ปัญหาได้ดีในความน่าจะเป็นอยู่ที่ 85% - 95% ตัวอย่างการเกิดการสลับสายพันธุ์ กำหนดให้ความน่าจะเป็นของการเกิดสลับสายพันธุ์ เป็น 85% ค่าที่สุ่มอยู่ในช่วง 0-100 นั่นหมายความว่าหากสุ่มตัวเลข ได้ค่าที่ <= 85 ก็จะเกิดการสลับสายพันธุ์นอกจากนั้นจะไม่เกิดการสลับสายพันธุ์จากตัวอย่างสุ่มค่าได้ 35 จึงเกิดการสลับสายพันธุ์

ความน่าจะเป็นในการมิวเตชันหรือความน่าจะเป็นของการกลายพันธุ์ จะมีค่าที่อยู่ในช่วง 0 – 100 จากการทดลองของนักวิทยาศาสตร์หลายท่าน ได้พบว่าความน่าจะเป็นของการกลายพันธุ์ส่วนใหญ่อยู่ที่ 0% - 1% ต่อ 1 ตำแหน่งของโครโมโซมถ้าไม่มีการกลายพันธุ์ หมายความว่าผลที่ได้เกิดจากการสลับสายพันธุ์เพียงอย่างเดียว แต่ถ้าหากมีการกลายพันธุ์ 100% จะทำให้โครโมโซมมีการเปลี่ยนแปลงทั้งหมด ซึ่งการเกิดกลายพันธุ์จะไม่เกิดบ่อยมากนักเพราะจะทำให้วิธีการแบบขั้นตอนวิธีเชิงพันธุกรรมเปลี่ยนไปเป็น random search ตัวอย่างการเกิดกลายพันธุ์โดยกำหนดให้ความน่าจะเป็นของการเกิดกลายพันธุ์เป็น 1% ค่าที่สุมอยู่ในช่วง 0-100 นั่นหมายความว่า ณ ตำแหน่งนั้นหากสุ่มตัวเลขได้ค่าที่ <= 1 ก็จะเกิดการกลายพันธุ์จากตัวอย่างจะเห็นว่า ณ ตำแหน่งที่ 2 ของโครโมโซมมีการสุ่มที่มีค่าเป็นไปตามเงื่อนไขจึงเกิดกลายพันธุ์ ส่วน ณ ตำแหน่งอื่นๆ ของโครโมโซมได้ค่าสุ่มที่ไม่เป็นไปตามเงื่อนไขจึงไม่เกิดกลายพันธุ์ขึ้น

สรุปขั้นตอนการทำงานของวิธีการของเจเนติกอัลกอริทึม คือการกำหนดฟังก์ชันจุดประสงค์รวมทั้งรูปแบบโครโมโซมเสียก่อน จากนั้นจึงเริ่มสร้างประชากรเบื้องต้นตามรูปแบบโครโมโซมที่ได้กำหนดไว้ เมื่อได้ประชากรเริ่มต้นแล้วก็ทำการวัดค่าความเหมาะสมของแต่ละโครโมโซมโดยใช้ฟังก์ชันค่าความเหมาะสมเพื่อคัดเลือกเข้าสู่กระบวนการรีโปรดักชันโดยการคัดเลือกเอาเฉพาะโครโมโซมที่มีค่าความเหมาะสมที่เป็นที่น่าพอใจชุดหนึ่งเก็บไว้ โครโมโซมที่คัดเลือกไว้นั้นจะถูกนำมาทำการครอสโอเวอร์และมิวเตชันได้เป็นโครโมโซมชุดใหม่ และนำเอาโครโมโซมชุดใหม่นี้มาวัดค่าความเหมาะสมเพื่อทำการคัดเลือกและดำเนินการต่อไปจนสิ้นสุดตามเงื่อนไขที่ได้กำหนดไว้ก็จะได้โครโมโซมที่มีค่าความเหมาะสมเป็นที่น่าพอใจ หรือได้คำตอบของปัญหา

# บทที่ 3

# การพัฒนาโปรแกรมจัดตารางสอนโดยประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึม

การพัฒนาโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนโดยประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึม เริ่มต้นดำเนินการวิจัยจากการสำรวจข้อมูลเบื้องต้นเกี่ยวกับการจัดตารางสอนในมหาวิทยาลัยซึ่งได้แก่ข้อมูลเกี่ยวกับกระบวนการจัดตารางสอนและข้อมูลเกี่ยวกับการจัดตารางสอนเพื่อศึกษารูปแบบของปัญหาการจัดตารางสอน และนำข้อมูลดังกล่าวมาใช้ในการออกแบบและพัฒนาโปรแกรม ซึ่งจะนำเอาวิธีการของเจเนติกอัลกอริทึมเข้ามาประยุกต์ใช้ในส่วนของการประมวลผลของโปรแกรมเพื่อหาคำตอบที่เหมาะสมให้กับปัญหาโดยใช้การพิจารณาการขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อยที่สุด (fitness value minimization) หรือเรียกว่า ค่าความเหมาะสม เป็นตัวชี้วัดตารางสอนที่ดีที่สุด จากนั้นจึงดำเนินการออกแบบและพัฒนาโปรแกรมจากข้อมูลและวิธีการที่ได้วางแผนด้วยโปรแกรม Microsoft Visual Studio 2010 (visual C#) และเชื่อมต่อกับระบบฐานข้อมูลด้วยโปรแกรม Microsoft Access 2007 โดยเนื้อหาที่จะกล่าวถึงในบทนี้ประกอบด้วย การวิเคราะห์ปัญหาของระบบการจัดตารางสอน โครงสร้างโปรแกรมการจัดตารางสอน การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอน และลักษณะโปรแกรมจัดตารางสอนที่พัฒนาขึ้น

## 3.1 การวิเคราะห์ปัญหาระบบการจัดตารางสอน

ในทุกภาคการศึกษามีกระบวนการหนึ่งที่ต้องจัดทำกันอยู่ทุกภาคการศึกษา นั้นคือกระบวนการจัดตารางสอน จากการศึกษากระบวนการจัดตารางสอนพบว่าเป็นกระบวนการที่ค่อนข้างยุ่งยากซับซ้อน เนื่องจากการจัดตารางสอนแต่ละครั้งมีสิ่งต่างๆ ที่ต้องพิจารณามากมาย เช่น ข้อมูลสำหรับการจัดตารางสอน เงื่อนไขในการพิจารณาเพื่อให้ตารางสอนที่จัดออกมาเป็นตารางสอนที่พึงพอใจ โดยความพึงพอใจนั้นจะขึ้นอยู่กับแต่ละสถานศึกษา เช่น เวลาสอนของอาจารย์ไม่ซ้ำซ้อนกัน เวลาเรียนของนักศึกษาไม่ซ้ำซ้อน เวลาการใช้ห้องเรียนไม่ซ้ำซ้อนกัน อาจารย์สอนติดต่อกันไม่เกิน 4 คาบ หรือนักศึกษาเรียนติดต่อกันไม่เกิน 4 คาบ เป็นต้น ซึ่งแต่ละ

สถานศึกษาจะมีข้อมูลสำหรับการจัดตารางสอนที่ต่างกัน เช่น จำนวนหลักสูตรที่เปิดสอน จำนวนอาจารย์ผู้สอน จำนวนนักศึกษา จำนวนห้องเรียน เป็นต้น และแต่ละสถานศึกษาก็จะมีลักษณะการเรียนการสอนที่แตกต่างกันออกไป เช่น บางสถานศึกษามีห้องเรียนประจำสำหรับนักเรียนแต่ละกลุ่ม บางสถานศึกษาใช้ห้องเรียนร่วมกัน บางสถานศึกษาเน้นการเรียนในเชิงปฏิบัติ บางสถานศึกษาเน้นการเรียนด้วยตนเอง เป็นต้น ด้วยเหตุผลดังกล่าวข้างต้นแต่ละสถานศึกษาจึงมีวิธีการในการจัดตารางสอนที่ต่างกันออกไป มีเงื่อนไขที่ต้องพิจารณาต่างกัน ดังนั้นจึงต้องมีการกำหนดเงื่อนไขในการจัดตารางสอนที่เหมาะสมในการจัดตารางสอนสำหรับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งจะกล่าวถึงในหัวข้อถัดไป

### 3.1.1 เงื่อนไขการจัดตารางสอน

การจัดตารางสอนเป็นการจัดตารางนัดหมายให้ลงตัวระหว่างนักศึกษา อาจารย์ และห้องเรียนตามแต่ละรายวิชา โดยการจัดตารางสอนนั้นเป็นการจัดตารางตามหลักสูตรของนักศึกษาที่เรียนตามแผนการเรียน ที่ต้องลงทะเบียนเรียนในแต่ละภาคการศึกษา โดยจะกำหนดจำนวนกลุ่มผู้เรียนในรายวิชานั้นๆ ตามหลักสูตรของแต่ละสาขาวิชา นอกจากนี้การจัดตารางสอนนั้นจะต้องจัดตามเงื่อนไขที่กำหนดไว้ เช่น เวลาการสอนของอาจารย์ผู้สอนต้องไม่ซ้ำซ้อนกัน เวลาเรียนของกลุ่มนักศึกษาต้องไม่ซ้ำซ้อนกัน เวลาการใช้ห้องเรียนต้องไม่ซ้ำซ้อนกัน เป็นต้น เพื่อให้ได้ตารางสอนที่เหมาะสม อย่างไรก็ตามจากการศึกษาปัญหาการจัดตารางสอนนั้น พบว่าปัญหาต่างๆ เกิดขึ้นเนื่องจากการไม่สามารถควบคุมเงื่อนไขอันเป็นความต้องการของนักศึกษา อาจารย์ วิชาเรียน และห้องเรียนได้อย่างสมบูรณ์ จึงต้องมีการปรับปรุงตารางสอนอยู่เสมอ ซึ่งจากปัญหาที่พบ เช่น มีการซ้ำซ้อนกันของเวลาอาจารย์ผู้สอน มีการซ้ำซ้อนกันของเวลาเรียนของกลุ่มนักศึกษา มีการซ้ำซ้อนกันของเวลาการใช้ห้องเรียนเพื่อการเรียนการสอน และมีการเรียนการสอนในเวลาเที่ยง เป็นต้น ดังนั้นควรมีการนำเงื่อนไขต่างๆ มาช่วยควบคุมการจัดตารางสอนอย่างเหมาะสมเพื่อให้ได้ตารางสอนที่ตรงตามความต้องการมากที่สุด ในงานวิจัยนี้ได้นำเสนอเงื่อนไขที่ใช้ในการจัดตารางสอน 2 ประเภทคือ เงื่อนไขหรือข้อจำกัดบังคับ (hard constraints) คือเงื่อนไขที่ต้องไม่เกิดขึ้นในการจัดตารางสอน และเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ (soft constraints) คือเงื่อนไขที่ยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ต้องเกิดขึ้นน้อยที่สุดจึงจะให้ตารางสอนที่ตรงกับความต้องการมากที่สุด รายละเอียดเงื่อนไขทั้ง 2 ชนิดสามารถแสดงได้ดังนี้

**เงื่อนไขหรือข้อจำกัดบังคับ**

1. ในวันและเวลาเดียวกันอาจารย์ผู้สอนหนึ่งคนสามารถสอนได้หนึ่งกลุ่มเท่านั้น

2. ในวันและเวลาเดียวกันนักศึกษาหนึ่งกลุ่มสามารถเรียนได้หนึ่งรายวิชาเท่านั้น
3. ในวันและเวลาเดียวกันสามารถกำหนดการเรียนการสอนให้กับห้องเรียน 1 ห้องเรียนได้หนึ่งรายวิชาเท่านั้น
4. กำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องให้กับห้องปฏิบัติการ

**เงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์**

1. คาบที่ 2 ของรายวิชาเดียวกันไม่ควรจัดให้อยู่ในวันเดียวกันโดยเว้นคาบว่าง เช่น วิชาสถิติวิศวกรรม 1 จัดในวันจันทร์ เวลา 08.00-08.50 น. ไม่ควรที่จะจัดในวันจันทร์ เวลา 13.00-13.50 น. อีกครั้ง เป็นต้น หากต้องการให้มีการเรียนการสอนในรายวิชานี้ 2 คาบใน 1 วัน ควรจัดให้เป็น 2 คาบติดต่อ เช่น จัดในวันจันทร์ เวลา 08.00-09.50 น. เป็นต้น
2. ควรจัดการสอนรายวิชาประจำภาควิชาให้กับห้องเรียนในภาควิชานั้นก่อน เช่น วิชาสถิติวิศวกรรมเป็นรายวิชาประจำภาควิชาวิศวกรรมอุตสาหการ ก็ควรจัดให้กับห้องเรียนในภาควิชาวิศวกรรมอุตสาหการก่อน
3. ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป เช่นหากในวันนั้นนักศึกษามีเรียนในเวลา 08.00-08.50 น. แล้วไม่ควรมีคาบว่างแล้วมีเรียนต่อในเวลา 11.00-11.50 น. เป็นต้น
4. ในแต่ละวันควรมีการเว้นคาบว่างในเวลา 12.00-12.50 น. เพื่อให้นักศึกษาพักทานอาหารกลางวัน
5. ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดกันเกิน 4 คาบ
6. ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ

นอกจากนี้ยังมีข้อกำหนดพื้นฐานของแต่ละภาควิชาฯ ที่ใช้ในการจัดตารางสอน เช่น อาจารย์ประจำภาควิชาวิศวกรรมอุตสาหการทั้งหมดจะต้องว่างพร้อมกันทุกวันพฤหัสบดี เวลา 13.30-16.30 น. เพื่อประชุมภาควิชา เป็นต้น หรือเงื่อนไขของอาจารย์ทีมผู้บริหารที่ต้องการกำหนดเวลาว่างเพื่อประชุมทีมผู้บริหาร ซึ่งเป็นเงื่อนไขต่างๆ ดังที่กล่าวมาข้างต้นล้วนแล้วแต่เป็นเงื่อนไขที่มีผลต่อการจัดตารางสอนที่เป็นที่น่าพึงพอใจ จึงจำเป็นที่การจัดตารางสอนต้องสามารถกำหนดเวลาว่างของอาจารย์แต่ละคนได้ด้วย เนื่องจากอาจารย์แต่ละคนนั้นมีภาระหน้าที่ในงานด้านต่างๆ ที่แตกต่างกันออกไป

### 3.1.2 การกำหนดฟังก์ชันความเหมาะสม (fitness function)

ฟังก์ชันความเหมาะสม เป็นฟังก์ชันที่ใช้วัดค่าความเหมาะสมของโครโมโซมตารางสอนเพื่อค้นหาโครโมโซมตารางสอนที่มีค่าความเหมาะสมที่ดีที่สุดเป็นคำตอบของปัญหา โดยฟังก์ชันความเหมาะสมเป็นฟังก์ชันสร้างจากเงื่อนไขที่ใช้ในการจัดตารางสอนทั้ง 2 ประเภทที่ได้กำหนดไว้ จะมีนำการกำหนดค่าน้ำหนักของเงื่อนไขแต่ละข้อเข้ามาช่วยในการจำแนกโครโมโซมที่ดี โดยการกำหนดค่าน้ำหนักของเงื่อนไขบังคับสูงๆ ให้มีความแตกต่างกับค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์มากๆ จะทำให้โครโมโซมที่ไม่ผ่านเงื่อนไขบังคับมีค่าความเหมาะสมสูงมาก ซึ่งบ่งบอกว่าเป็นโครโมโซมที่ไม่ดี และไม่ถูกเลือกไปเป็นคำตอบของปัญหา

การวัดค่าโครโมโซมตารางสอนมีเกณฑ์การวัดค่าความเหมาะสมจากฟังก์ชันความเหมาะสมโดยโครโมโซมที่มีค่าความเหมาะสมน้อยที่สุดจะเป็นโครโมโซมที่ดีที่สุด ซึ่งฟังก์ชันค่าความเหมาะสมเป็นไปดังสมการที่ 3.1

$$\text{fitness value} = \sum_{i=1}^{N}(n_i * w_i) \tag{3.1}$$

n = จำนวนครั้งที่เกิดการละเมิดเงื่อนไข i

N = จำนวนเงื่อนไขทั้งหมด (รวมทั้ง 2 ประเภท)

W = ค่าน้ำหนักของเงื่อนไข i (ให้ค่าตามลำดับความสำคัญของเงื่อนไขแต่ละข้อโดยกำหนดค่าน้ำหนักโดยผู้ทำวิจัย)

การให้ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์จะให้ตามลำดับความสำคัญของเงื่อนไขแต่ละข้อเป็นหลัก ในงานวิจัยนี้ได้ดำเนินการหาลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจากการทำแบบสอบถามขึ้นมา ตัวอย่างแบบสอบถามตามภาคผนวก ก เป็นแบบสอบถามที่ให้ผู้ตอบแบบสอบถามเรียงลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์ในการจัดตารางสอนจากมากไปน้อย และได้สอบถามเกี่ยวกับข้อเสนอแนะที่ต้องการใช้พิจารณาสำหรับการจัดตารางสอน โดยสอบถามอาจารย์คณะวิศวกรรมศาสตร์จำนวน 125 คน จากอาจารย์ทั้งทั้งหมด 150 คน ซึ่งคิดเป็นร้อยละ 80 จากอาจารย์ทั้งหมด จากแบบสอบถามสามารถสรุปลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์ได้ตามตาราง 3.1 จากตารางสามารถอธิบายได้ว่าผู้ตอบแบบสอบถามให้ความสำคัญของเงื่อนไขข้อที่ 6 มาเป็นลำดับที่ 1 ตามด้วยข้อที่ 5, 4, 2, 3 และ 1 ตามลำดับ ซึ่งค่าเหล่านี้ได้จากการหาค่าเฉลี่ยของลำดับความสำคัญ เงื่อนไขข้อใดที่มีค่าเฉลี่ยน้อย

ที่สุดก็จะมีความสำคัญมากที่สุด เนื่องจากกำหนดให้ 1 เป็นค่าลำดับความสำคัญที่มีค่ามาก ถึง 6 เป็นลำดับความสำคัญที่มีค่าน้อย โดยแบบสอบถามนี้จะนำมาเป็นเพียงแนวทางในการให้น้ำหนักของเงื่อนไขแต่ละข้อในการจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์เท่านั้น ดังนั้นหากจะนำลำดับความสำคัญของเงื่อนไขแต่ละข้อนี้ไปใช้ในการจัดตารางสอนสำหรับคณะ หรือมหาวิทยาลัยอื่นๆ อาจจะต้องมีการเปลี่ยนแปลงลำดับความสำคัญในการกำหนดค่าน้ำหนักของเงื่อนไขในการจัดตารางสอนแต่ละข้อให้เหมาะสมยิ่งขึ้นได้ เมื่อได้แนวโน้มของลำดับความสำคัญจากการทำแบบสอบถามแล้ว จะนำค่าลำดับความสำคัญเหล่านี้ไปใช้เป็นปัจจัยในการพิจารณากำหนดค่าน้ำหนักของเงื่อนไขแต่ละข้อ ซึ่งค่าน้ำหนักขึ้นอยู่กับความต้องการของผู้จัดตารางสอนว่าจะให้ค่าน้ำหนักเท่าไรก็ได้ โดยเงื่อนไขใดที่มีลำดับความสำคัญมากและไม่ต้องการให้เกิดการขัดแย้งของเงื่อนไขในตารางสอนที่จัดทำขึ้น ก็จะให้ค่าน้ำหนักข้อนั้นมากเป็นพิเศษก็ได้

ตาราง 3.1 ลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์จากแบบสอบถาม

| **เงื่อนไขเพื่อความสมบรูณ์** | **ลำดับความสำคัญ** |
|---|---|
| 1. ช่วงเวลาที่ 2 ของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 6 |
| 2. รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 4 |
| 3. ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบเพื่อเรียนในคาบถัดไป | 5 |
| 4. ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 3 |
| 5. ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดกันเกิน 4 คาบ | 2 |
| 6. ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 1 |

แต่อย่างไรก็ตามแบบสอบถามนี้ได้ทำการสอบถามเฉพาะอาจารย์ผู้สอนเพียงอย่างเดียว ไม่ได้ทำการสอบถามนักศึกษา ดังนั้นผู้ทำวิจัยจึงเห็นว่าเงื่อนไขบางข้อที่ควรมีลำดับความสำคัญเท่ากัน เช่น ผู้สอนไม่ควรมีการสอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบขึ้นไปกับนักศึกษาไม่ควรมีการเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบขึ้นไป แต่ผลจากแบบสอบถามให้ลำดับความสำคัญของอาจารย์เป็นลำดับ 1 นักศึกษาลำดับ 2 ดังนั้นจึงได้มีปรับเปลี่ยนการให้ลำดับความสำคัญของเงื่อนไขแต่ละข้อใหม่เพื่อให้ตารางสอนเป็นที่น่าพึงพอใจทั้งอาจารย์และนักศึกษาตามตาราง 3.2

ตาราง 3.2 ลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์

| เงื่อนไขเพื่อความสมบรูณ์ | ลำดับความสำคัญ |
|---|---|
| 1. ช่วงเวลาที่ 2 ของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 4 |
| 2. รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 3 |
| 3. ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 4 |
| 4. ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 2 |
| 5. ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดกันเกิน 4 คาบ | 1 |
| 6. ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 1 |

การกำหนดค่าน้ำหนักเงื่อนไขที่เป็นเงื่อนไขบังคับนั้นให้มีค่าน้ำหนักมาก ให้มีความแตกต่างกับค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ ทั้งนี้เพื่อต้องการให้มีการแยกเงื่อนไขบังคับออกจากเงื่อนไขเพื่อความสมบรูณ์อย่างชัดเจน อย่างเช่น กำหนดค่าน้ำหนักของเงื่อนไขบังคับเท่ากับ 1000 และกำหนดค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เพียงหลักสิบ โดยเรียงลำดับคะแนนมากน้อยตามลำดับความสำคัญของเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อ ทั้งนี้เนื่องจากกลไกทางเจเนติกอัลกอริทึมจะพยายามกำจัดคำตอบที่ด้อยไปเองตามธรรมชาติ ดังนั้นหากคำตอบใดผิดเงื่อนไขบังคับก็ทำให้มีค่าความเหมาะสมมาก ซึ่งไม่ได้คำตอบที่ดี ก็จะไม่ถูกเลือกต่อไป การกำหนดค่าน้ำหนักของเงื่อนไขในการจัดตารางสอนทั้ง 2 ประเภทที่ใช้ในการคำนวณค่าความเหมาะสมของงานวิจัยนี้เป็นไปตามตาราง 3.3

ตาราง 3.3 การกำหนดค่าน้ำหนักของเงื่อนไขในการจัดตารางสอน

| เงื่อนไข | ค่าน้ำหนัก |
|---|---|
| **เงื่อนไขบังคับ** | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 1,000 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 1,000 |
| การซ้ำซ้อนของวัน-เวลาการใช้ห้องเรียน | 1,000 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องจัดให้กับห้องปฏิบัติการ | 1,000 |

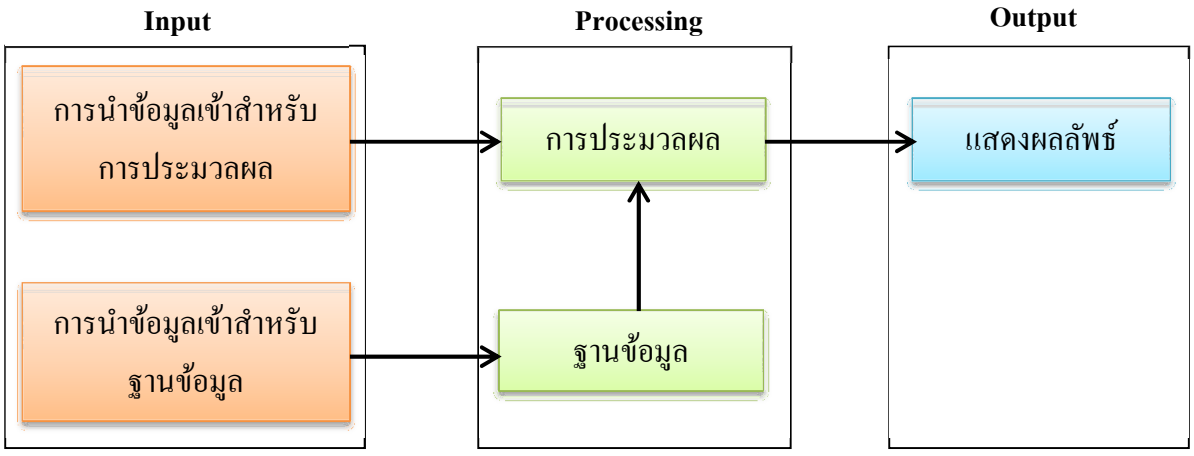
ตาราง 3.3 การกำหนดค่าน้ำหนักของเงื่อนไขในการจัดตารางสอน (ต่อ)

| เงื่อนไข | ค่าน้ำหนัก |
|---|---|
| **เงื่อนไขเพื่อความสมบูรณ์** | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 10 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 20 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 10 |
| ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 30 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 40 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 40 |

นอกจากนี้ยังมีข้อเสนอแนะที่น่าสนใจ เช่น หากเป็นคาบเรียนติดกันไม่ควรจัดห้องเรียนให้ไกลกันมาก ควรจะจัดให้อยู่ในอาคารเดียวกัน และควรจัดคาบเรียนเป็นคาบละ 90 นาที เนื่องในอนาคตอาจจะมีนโยบายของมหาวิทยาลัยออกมาให้เปลี่ยนระยะเวลาของคาบเรียนจากเดิมคาบ 50 นาที เปลี่ยนเป็นคาบละ 90 นาที จึงได้มีการนำข้อเสนอแนะมาพัฒนาโปรแกรมเพื่อให้โปรแกรมมีความยืดหยุ่น สามารถรองรับนโยบายของมหาวิทยาลัยข้อนี้ต่อไปในอนาคตได้

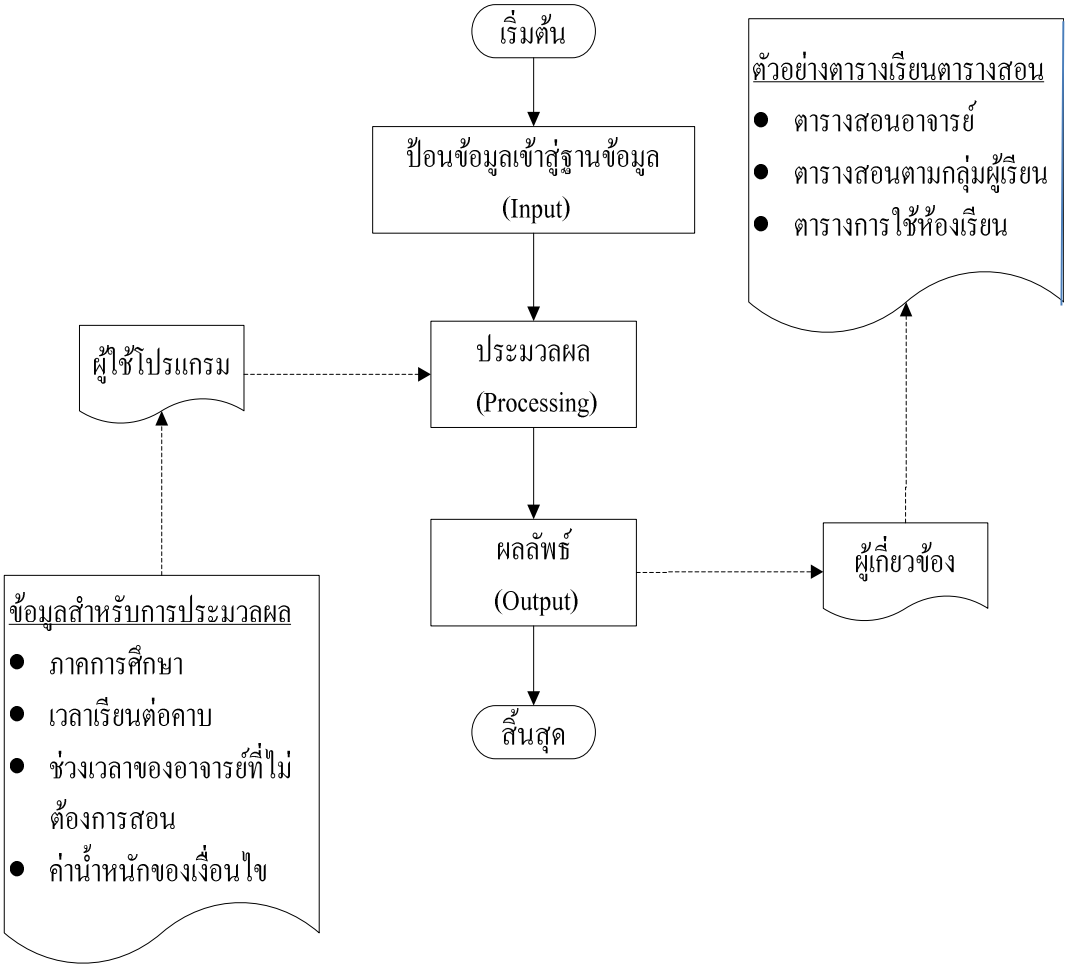
## 3.2 โครงสร้างโปรแกรมจัดตารางสอน

โปรแกรมการจัดตารางสอนมีโครงสร้างดังภาพประกอบ 3.1 ซึ่งประกอบด้วย 3 ส่วนหลัก คือ ส่วนการป้อนข้อมูลเข้า ส่วนการประมวลผล และส่วนแสดงผลลัพธ์




ภาพประกอบ 3.1 รูปแบบของโครงสร้างโปรแกรมที่ออกแบบไว้

กระบวนการทำงานโปรแกรมจัดตารางสอน เริ่มต้นโดยผู้ใช้งานโปรแกรมจะทำการป้อนข้อมูลเข้าสู่ตัวโปรแกรมในส่วนของการประมวลผล เพื่อทำการประมวลผลตามวิธีการของเจเนติกอัลกอริทึม จากนั้นแสดงผลในรูปแบบตารางการใช้ห้องเรียน ตารางสอนอาจารย์ และตารางเรียนตามกลุ่มนักศึกษา ที่มีข้อขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์น้อยที่สุดดังแสดงกระบวนการทำงานของโปรแกรมในภาพประกอบ 3.2




ภาพประกอบ 3.2 โครงสร้างกระบวนการทำงานของโปรแกรมการจัดตารางสอน

สามารถอธิบายรายละเอียดแต่ละส่วนของโปรแกรมได้ดังนี้

1. ส่วนการป้อนข้อมูลเข้า (input)

ข้อมูลที่จำเป็นต้องป้อนเข้าสู่การประมวลผลสำหรับโปรแกรมการจัดตารางสอน 2 ส่วนคือ ข้อมูลสำหรับการประมวลผล และข้อมูลเข้าสู่ฐานข้อมูล

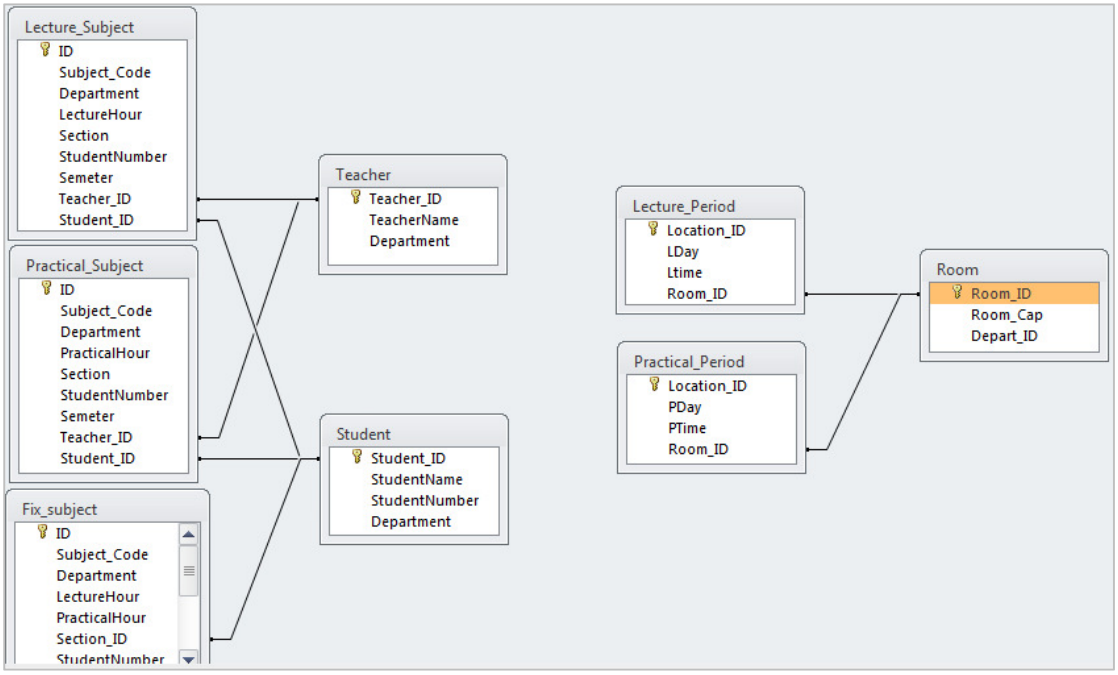
2. ส่วนการประมวลผล (processing)

ในส่วนการประมวลผลของโปรแกรมจะประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึมที่เหมาะสมกับปัญหาการจัดตารางสอน โดยพิจารณาการมีข้อขัดแย้งกับเงื่อนไขบังคับและเงื่อนไขเพื่อความสมบูรณ์น้อยที่สุดเป็นตัวชี้วัดตารางสอนที่เหมาะสมที่สุด

3. ส่วนการแสดงผลลัพธ์ (output)

ผลลัพธ์ที่ได้จากโปรแกรมการจัดตารางสอนคือ ตารางการใช้ห้องเรียน ตารางสอนอาจารย์ และตารางเรียนตามกลุ่มนักศึกษา

ฐานข้อมูลที่ได้ออกแบบไว้ประกอบด้วยตารางข้อมูลที่เกี่ยวข้องกับการจัดตารางสอนทั้งสิ้น 8 ตาราง คือ ตารางข้อมูลรายวิชาบรรยาย ตารางข้อมูลรายวิชาปฏิบัติ ตารางรายวิชานอกคณะ ตารางข้อมูลอาจารย์ ตารางข้อมูลกลุ่มนักศึกษา ตารางข้อมูลคาบเรียนรายวิชาบรรยาย ตารางข้อมูลคาบเรียนรายวิชาปฏิบัติ และตารางข้อมูลห้องเรียน และสามารถแสดงความสัมพันธ์ของข้อมูลในฐานข้อมูลสามารถแสดงได้ดังภาพประกอบ 3.3




ภาพประกอบ 3.3 ความสัมพันธ์ของตารางข้อมูลในฐานข้อมูล

การทำงานของโปรแกรมส่วนการคำนวณค่าความเหมาะสมของแต่ละโครโมโซมจากเงื่อนไขที่กำหนดขึ้นตามหัวข้อ 3.1.1 เพื่อใช้ในการจัดตารางสอนครั้งนี้ มีเงื่อนไขทั้งหมด 10 ข้อ โดยแบ่งเป็นเงื่อนไขบังคับ 4 ข้อ และเงื่อนไขเพื่อความสมบูรณ์ 6 ข้อ จากการออกแบบโครโมโซมทำให้สามารถลดขั้นตอนในการตรวจสอบเงื่อนไขบังคับข้อที่ 3. คือ ในวันและเวลา

เดียวกันกำหนดการเรียนให้ห้องเรียนได้หนึ่งรายวิชาเท่านั้น โดยตำแหน่งยืนแต่ละตำแหน่งจะมีวัน-เวลา และห้องเรียนที่ใช้ในการจัดตารางสอนไม่ซ้ำซ้อน ทำให้ไม่สามารถกำหนดรายวิชาลงในตำแหน่งยืนเดียวกันได้ จึงไม่เกิดการซ้ำซ้อนของตารางการใช้ห้องเรียน และจากการออกแบบโครโมโซมที่แบ่งเป็นโครโมโซมย่อยรายวิชาบรรยายและโครโมโซมย่อยรายวิชาปฏิบัติประกอบกับออกแบบฐานข้อมูลแยกตารางของรายวิชาออกเป็นตารางรายวิชาบรรยายและตารางรายวิชาปฏิบัติทำให้สามารถลดขั้นตอนในการตรวจสอบเงื่อนไขบังคับข้อที่ 4. กำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา โดยในช่วงโครโมโซมย่อยรายวิชาบรรยายจะสุ่มรายวิชาจากตารางรายวิชาบรรยาย ส่วนโครโมโซมย่อยรายวิชาปฏิบัติก็จะสุ่มรายวิชาจากตารางรายวิชาปฏิบัติ

เมื่อกระบวนการสร้างโครโมโซมเสร็จสิ้นเรียบร้อยแล้ว กระบวนการต่อไปคือกระบวนการตรวจสอบเงื่อนไขการจัดตารางสอนของแต่ละโครโมโซมที่สร้างขึ้น เพื่อความเข้าใจในการอธิบายกระบวนการตรวจสอบเงื่อนไขการจัดตารางสอนมากยิ่งขึ้น จึงขอสรุปตัวแปรต่างๆ และข้อมูลที่ถูกจัดเก็บในตัวแปรเหล่านั้นตามตาราง 3.4 เพื่อนำไปใช้ในการประมวลผลของโปรแกรมในส่วนของการตรวจสอบเงื่อนไขการจัดตารางสอน

ตาราง 3.4 ตัวแปรและข้อมูลตัวแปรที่ใช้ในการคำนวณคะแนนค่าความเหมาะสม

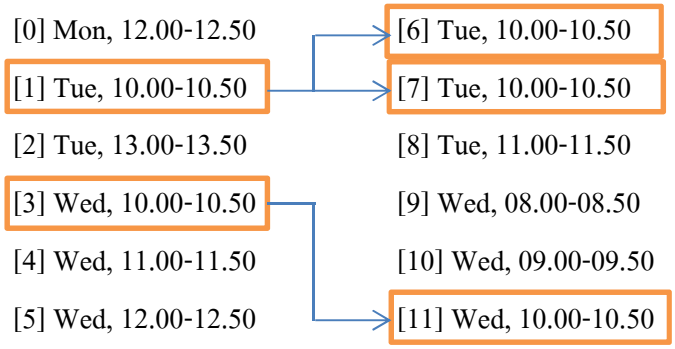
| **ตัวแปร** | **ข้อมูลที่ถูกจัดเก็บในตัวแปร** |
|---|---|
| List_Teacher | - ชื่อ-นามสกุล ของอาจารย์<br>- วัน-เวลา ที่อาจารย์คนนั้นมีสอน |
| List_Student | - กลุ่มนักศึกษา<br>- วัน-เวลา ที่กลุ่มนักศึกษากลุ่มนั้นมีเรียน |
| List_Subject | - รหัสวิชา<br>- วัน-เวลา ที่รายวิชาถูกจัดให้มีการเรียนการสอน |
| List_Room | ชื่อห้องเรียน |

จากตาราง 3.4 สามารถนำข้อมูลของตัวแปรเหล่านั้นไปคำนวณจำนวนครั้งของการละเมิดเงื่อนไขแต่ละข้อ เพื่อนำไปหาค่าความเหมาะสมของแต่ละโครโมโซมต่อไป ขั้นตอนการประมวลผลของโปรแกรมในการตรวจสอบเงื่อนไขแต่ละข้อสามารถอธิบายได้ดังนี้

(1) การคำนวณหาจำนวนครั้งที่เวลาสอนของอาจารย์ซ้ำซ้อนกัน

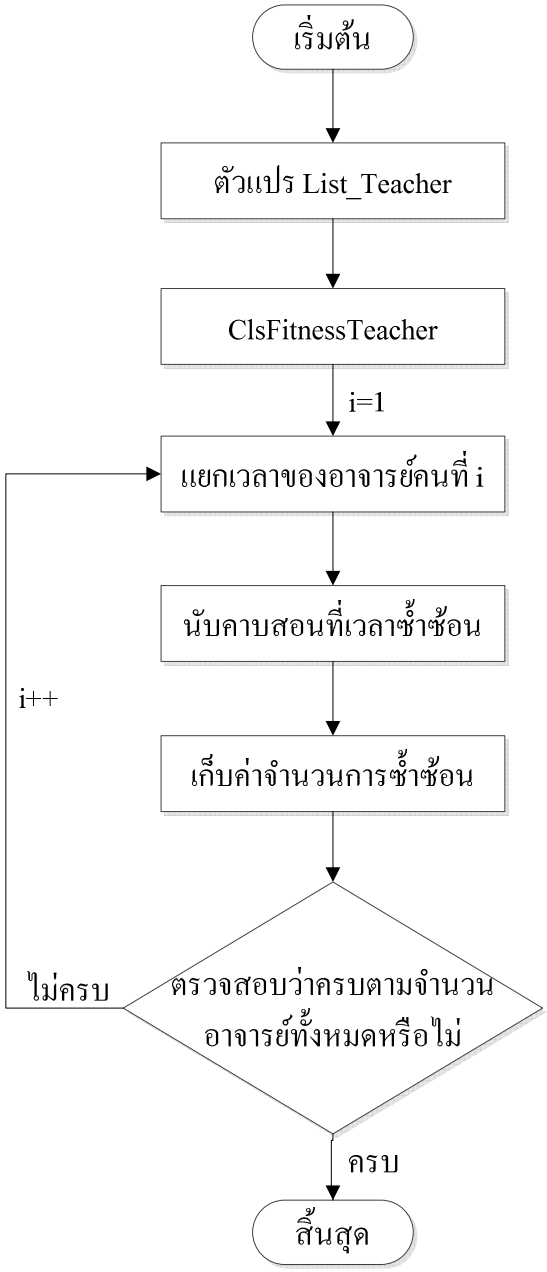
เงื่อนไขข้อนี้เป็นเงื่อนไขบังคับที่ไม่สามารถเกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง หากตารางสอนใดที่ขัดแย้งกับเงื่อนไขนี้แล้ว จะถือว่าเป็นตารางสอนที่เป็นไปไม่ได้ ไม่สามารถนำไปใช้งานได้จริง การทำงานของโปรแกรมแสดงดังภาพประกอบ 3.4 มีขั้นตอนดังนี้

1. โปรแกรมจะสร้างตัวแปรชื่อ List_Teacher แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessTeacher ซึ่งหน้าที่ในการตรวจสอบการซ้ำซ้อนของเวลาสอนของอาจารย์
2. จากนั้นทำการแยกเวลาสอนของอาจารย์ผู้สอนแต่ละคนเพื่อทำการนับจำนวนครั้งที่เวลาสอนของอาจารย์ซ้ำซ้อนกันในแต่ละโครโมโซม โดยขอยกตัวอย่างอาจารย์ขึ้นมา 1 คน เพื่ออธิบายกระบวนการนับเวลาที่ซ้ำซ้อนของอาจารย์ เช่น TeacherA มีชั่วโมงสอน 12 ชม ก็จะทำการแยกเวลาดังนี้




จากตัวอย่างจะพบว่ามีคาบสอนที่เวลาของอาจารย์ซ้ำซ้อนเท่ากับ 3 ครั้ง คือ ในวัน อังคาร 10.00-10.50 น. ซ้ำซ้อนกัน 2 ครั้ง และในวันพุธ 10.00-10.50 น. ซ้ำซ้อนกัน 1 ครั้ง เป็นต้น

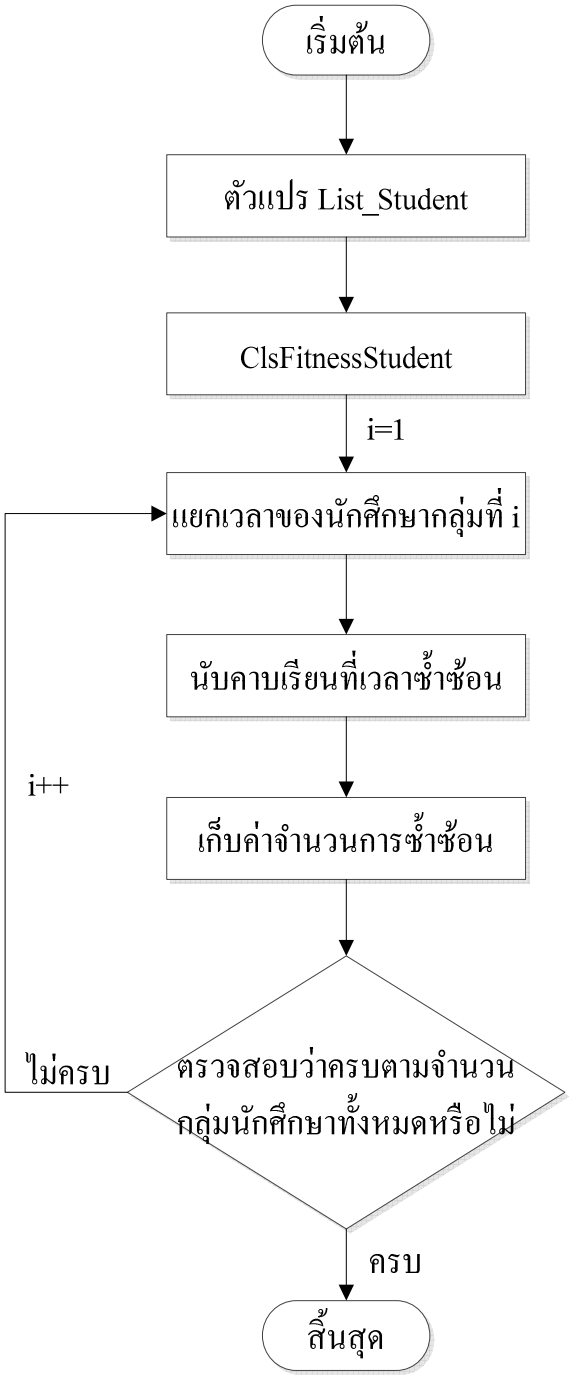
3. เก็บค่าจำนวนครั้งที่คาบสอนของอาจารย์ที่ซ้ำซ้อนกัน
4. ตรวจสอบว่านับจำนวนคาบสอนที่ซ้ำซ้อนครบตามจำนวนอาจารย์ทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของอาจารย์คนต่อไป และนับจำนวนคาบสอนที่เวลาซ้ำซ้อนของอาจารย์คนดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งที่เกิดการซ้ำซ้อนของอาจารย์ทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป




ภาพประกอบ 3.4 ขั้นตอนการคำนวณจำนวนคาบสอนที่ซ้ำซ้อนกันของอาจารย์

(2) การคำนวณหาจำนวนครั้งที่เวลาเรียนของกลุ่มนักศึกษาซ้ำซ้อนกัน

เงื่อนไขข้อนี้เป็นเงื่อนไขบังคับที่ไม่สามารถเกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง หากตารางสอนใดที่ขัดแย้งกับเงื่อนไขนี้แล้ว จะถือว่าเป็นตารางสอนที่เป็นไปไม่ได้ ไม่สามารถนำไปใช้งานได้จริง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.5




ภาพประกอบ 3.5 ขั้นตอนการคำนวณจำนวนคาบเรียนที่ซ้ำซ้อนกันของกลุ่มนักศึกษา

สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งที่คาบเรียนซ้ำซ้อนกันของกลุ่มนักศึกษา มีขั้นตอนการทำงานดังต่อไปนี้

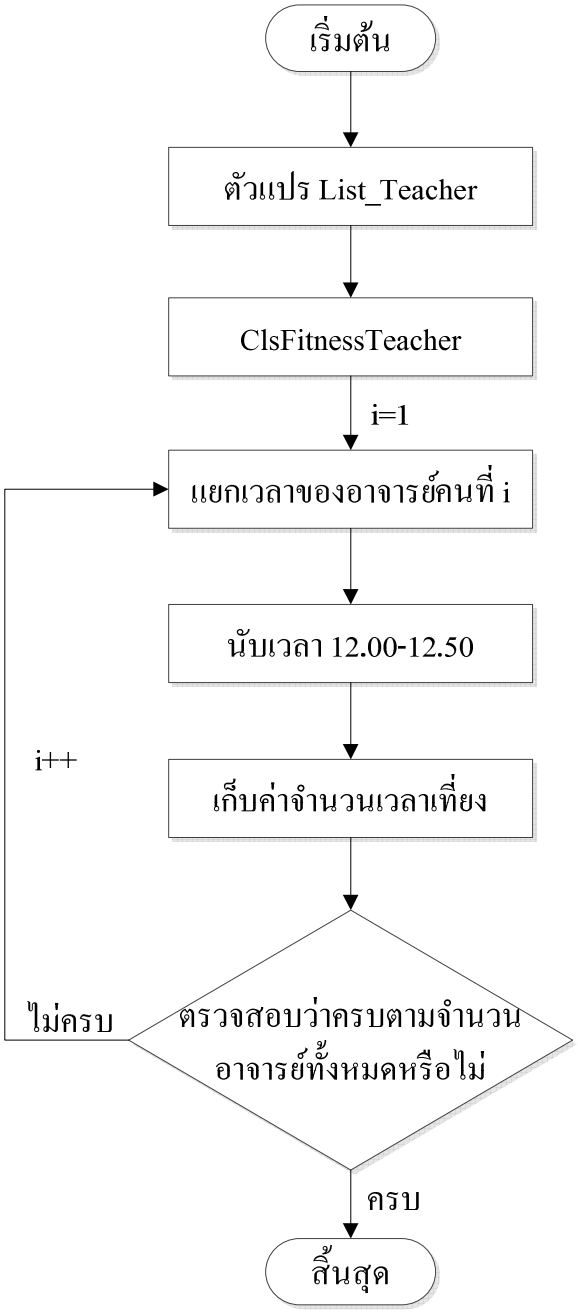
1. โปรแกรมจะสร้างตัวแปรชื่อ List_Student แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessStudent ซึ่งทำหน้าที่ในการตรวจสอบการซ้ำซ้อนของเวลาเรียนของกลุ่มนักศึกษา

2. จากนั้นทำการแยกเวลาเรียนของนักศึกษาแต่ละกลุ่มเพื่อทำการนับจำนวนครั้งที่เวลาเรียนของกลุ่มนักศึกษาซ้ำซ้อนกันในแต่ละโครโมโซม ซึ่งกระบวนการนับจำนวนการซ้ำซ้อนของเวลาเรียนของนักศึกษามีกระบวนการที่เหมือนกับการนับการซ้ำซ้อนของเวลาสอนของอาจารย์
3. เก็บค่าจำนวนครั้งที่มีคาบเรียนที่เวลาของนักศึกษาซ้ำซ้อนกันเกิดขึ้น
4. ตรวจสอบว่านับจำนวนคาบเรียนที่เวลาของนักศึกษาซ้ำซ้อนครบตามจำนวนกลุ่มนักศึกษาทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของนักศึกษากลุ่มต่อไป และนับจำนวนคาบสอนที่เวลาซ้ำซ้อนของนักศึกษากลุ่มดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งที่เกิดการซ้ำซ้อนของนักศึกษาทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป

(3) การคำนวณจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น.

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.6 มีขั้นตอนการทำงานดังนี้

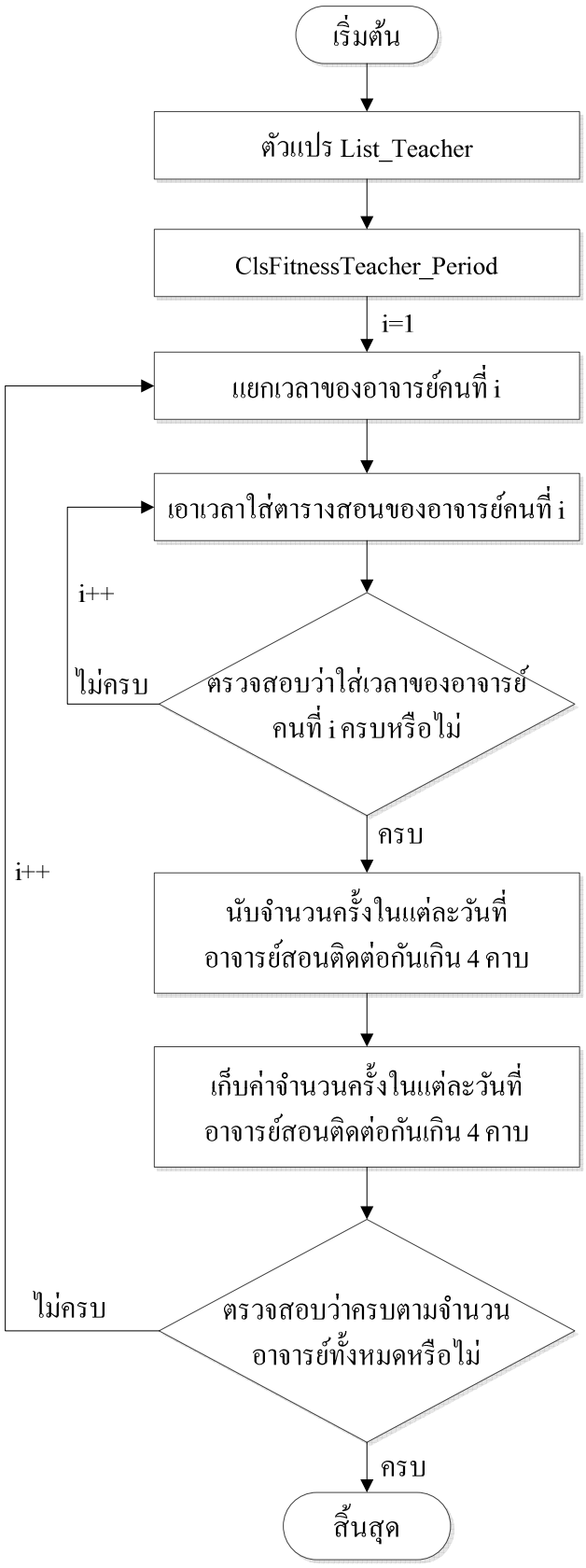
1. โปรแกรมจะสร้างตัวแปรชื่อ List_Teacher แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessTeacher ซึ่งหน้าที่ในการตรวจสอบการสอนของอาจารย์ในเวลา 12.00-12.50 น.
2. จากนั้นทำการแยกแวลาของอาจารย์ผู้สอนแต่ละคนเพื่อทำการนับจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น. ในแต่ละโครโมโซม
3. เก็บค่าจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น. เกิดขึ้น
4. ตรวจสอบว่านับจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น.ครบตามจำนวนอาจารย์ทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของอาจารย์คนต่อไป และนับจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น. ของอาจารย์คนดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น.ของอาจารย์ทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป




ภาพประกอบ 3.6 ขั้นตอนการคำนวณจำนวนครั้งที่มีการเรียนการสอนในเวลา 12.00-12.50 น.

(4) การคำนวณจำนวนครั้งในแต่ละวันที่อาจารย์สอนติดต่อกันเกิน 4 คาบ

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.7




ภาพประกอบ 3.7 ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่อาจารย์สอนติดต่อกันเกิน 4 คาบ

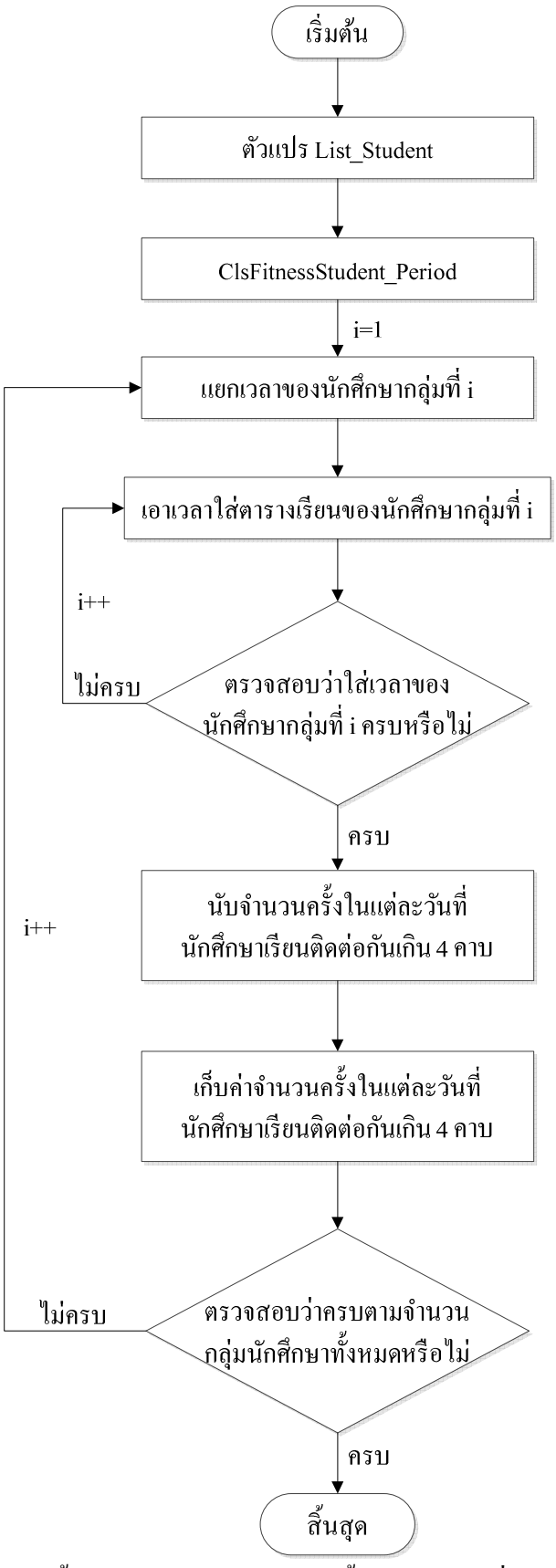
สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ มีขั้นตอนการทำงานดังต่อไปนี้

1. โปรแกรมจะสร้างตัวแปรชื่อ List_Teacher แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessTeacher_Period ซึ่งหน้าที่ในการตรวจสอบการสอนของอาจารย์ในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
2. จากนั้นทำการแยกเวลาของอาจารย์ผู้สอนแต่ละคน
3. เอาเวลาของอาจารย์แต่ละคน ใส่ในตารางสอนของอาจารย์คนนั้น
4. ตรวจสอบว่าเอาเวลาของอาจารย์ใส่ในตารางสอนของอาจารย์แต่ละคนครบแล้วหรือไม่ หากยังไม่ครบก็ไปเอาเวลามาใส่จนครบตามจำนวนคาบสอนของแต่ละคนจนครบ
5. หากครบแล้วก็ทำการนับจำนวนครั้งในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
6. เก็บค่าจำนวนครั้งในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
7. ตรวจสอบว่านับจำนวนครั้งในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ ครบตามจำนวนอาจารย์ทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของอาจารย์คนต่อไป และนับจำนวนครั้งในแต่ละวันอาจารย์สอนติดกันไม่เกิน 4 คาบ ของอาจารย์คนดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งในแต่ละวันที่อาจารย์สอนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ ของอาจารย์ทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป

(5) การคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนติดต่อกันเกิน 4 คาบ

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.8 สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ มีขั้นตอนการทำงานดังต่อไปนี้

1. โปรแกรมจะสร้างตัวแปรชื่อ List_Student แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessStudent_Period ซึ่งหน้าที่ในการตรวจสอบการเรียนของนักศึกษาในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
2. จากนั้นทำการแยกเวลาของนักศึกษาแต่ละกลุ่ม



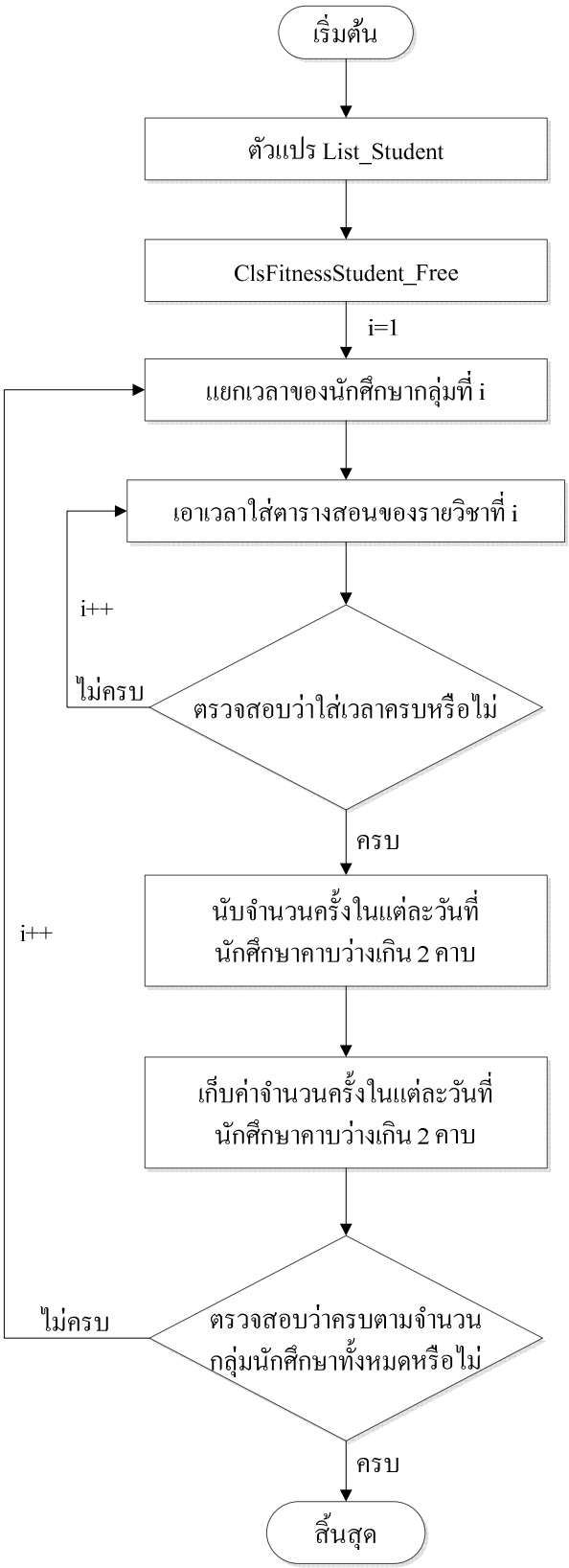

ภาพประกอบ 3.8 ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนติดต่อกันเกิน 4 คาบ

3. เอาเวลาของนักศึกษาแต่ละกลุ่ม ใส่ในตารางสอนของนักศึกษากลุ่มนั้น
4. ตรวจสอบว่าเอาเวลาของนักศึกษาใส่ในตารางสอนของนักศึกษาแต่ละกลุ่มครบแล้วหรือไม่ หากยังไม่ครบก็ไปเอาเวลามาใส่จนครบตามจำนวนคาบเรียนของแต่ละกลุ่มจนครบ
5. หากครบแล้วก็ทำการนับจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
6. เก็บค่าจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ
7. ตรวจสอบว่านับจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ ครบตามจำนวนกลุ่มนักศึกษาทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของนักศึกษากลุ่มต่อไป และนับจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบ ของนักศึกษากลุ่มดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งในแต่ละวันที่นักศึกษาเรียนรายวิชาบรรยายติดต่อกันเกิน 4 คาบของนักศึกษาทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป

(6) การคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.9 สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป มีขั้นตอนการทำงานดังต่อไปนี้

1. โปรแกรมจะสร้างตัวแปรชื่อ List_Student แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessStudent_Free ซึ่งหน้าที่ในการตรวจสอบการเรียนของนักศึกษาในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป
2. จากนั้นทำการแยกเวลาของนักศึกษาแต่ละกลุ่ม
3. เอาเวลาของนักศึกษาแต่ละกลุ่ม ใส่ในตารางสอนของนักศึกษากลุ่มนั้น
4. ตรวจสอบว่าเอาเวลาของนักศึกษาใส่ในตารางสอนของนักศึกษาแต่ละกลุ่มครบแล้วหรือไม่ หากยังไม่ครบก็ไปเอาเวลามาใส่จนครบตามจำนวนคาบเรียนของแต่ละกลุ่มจนครบ




ภาพประกอบ 3.9 ขั้นตอนการคำนวณจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป

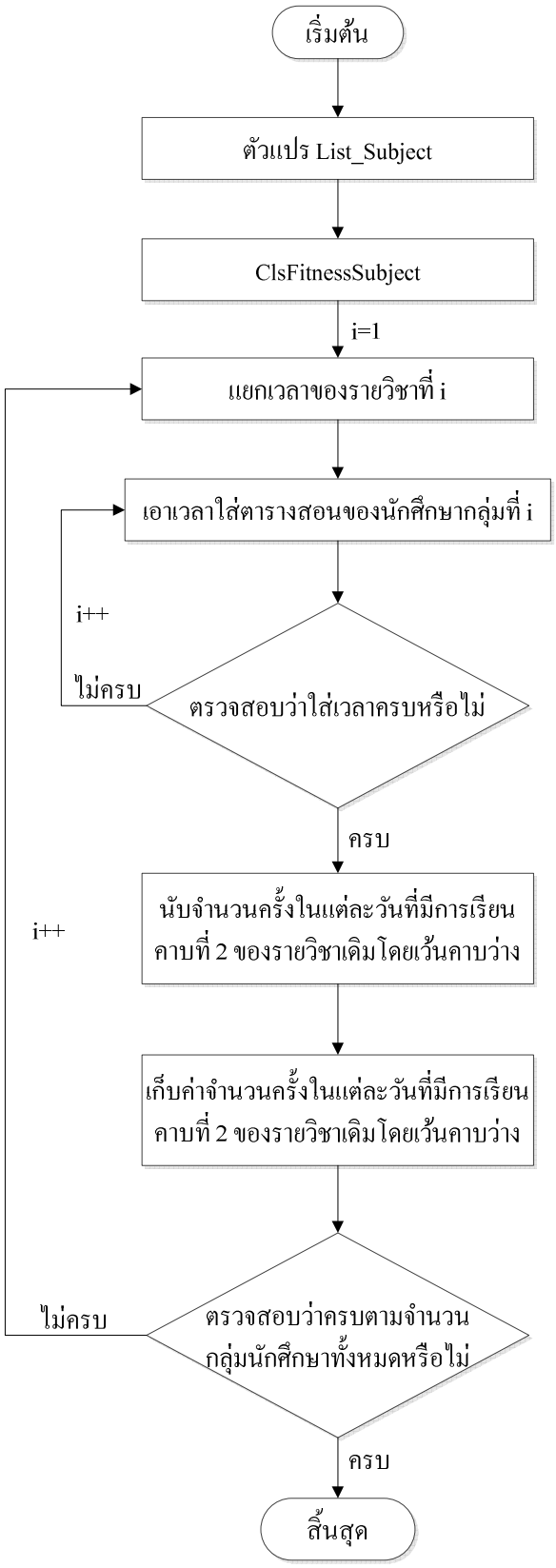
5. หากครบแล้วก็ทำการนับจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป

6. เก็บค่าจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป
7. ตรวจสอบว่านับจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป ครบตามจำนวนนักศึกษาทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของนักศึกษากลุ่มต่อไป และนับจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไป ของนักศึกษากลุ่มดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งในแต่ละวันที่นักศึกษามีคาบว่างเกิน 2 คาบเพื่อเรียนคาบถัดไปของนักศึกษาทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป

(7) การคำนวณจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.10 สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง มีขั้นตอนการทำงานดังต่อไปนี้

1. โปรแกรมจะสร้างตัวแปร List_Subject แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessSuject ซึ่งหน้าที่ในการตรวจสอบการเรียนการสอนของรายวิชาในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง
2. จากนั้นทำการแยกเวลาของรายวิชาแต่ละรายวิชา
3. เอาเวลาของรายวิชาแต่ละรายวิชา ใส่ในตารางสอนของรายวิชานั้น
4. ตรวจสอบว่าเอาเวลาของรายวิชาใส่ในตารางสอนของรายวิชาแต่ละรายวิชาครบแล้วหรือไม่ หากยังไม่ครบก็ไปเอาเวลามาใส่จนครบตามจำนวนคาบเรียนของแต่ละรายวิชาจนครบ หากครบแล้วก็ทำการนับจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง
5. เก็บค่าจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง




ภาพประกอบ 3.10 ขั้นตอนการคำนวณค่าจำนวนครั้งในแต่ละวันที่มีเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง

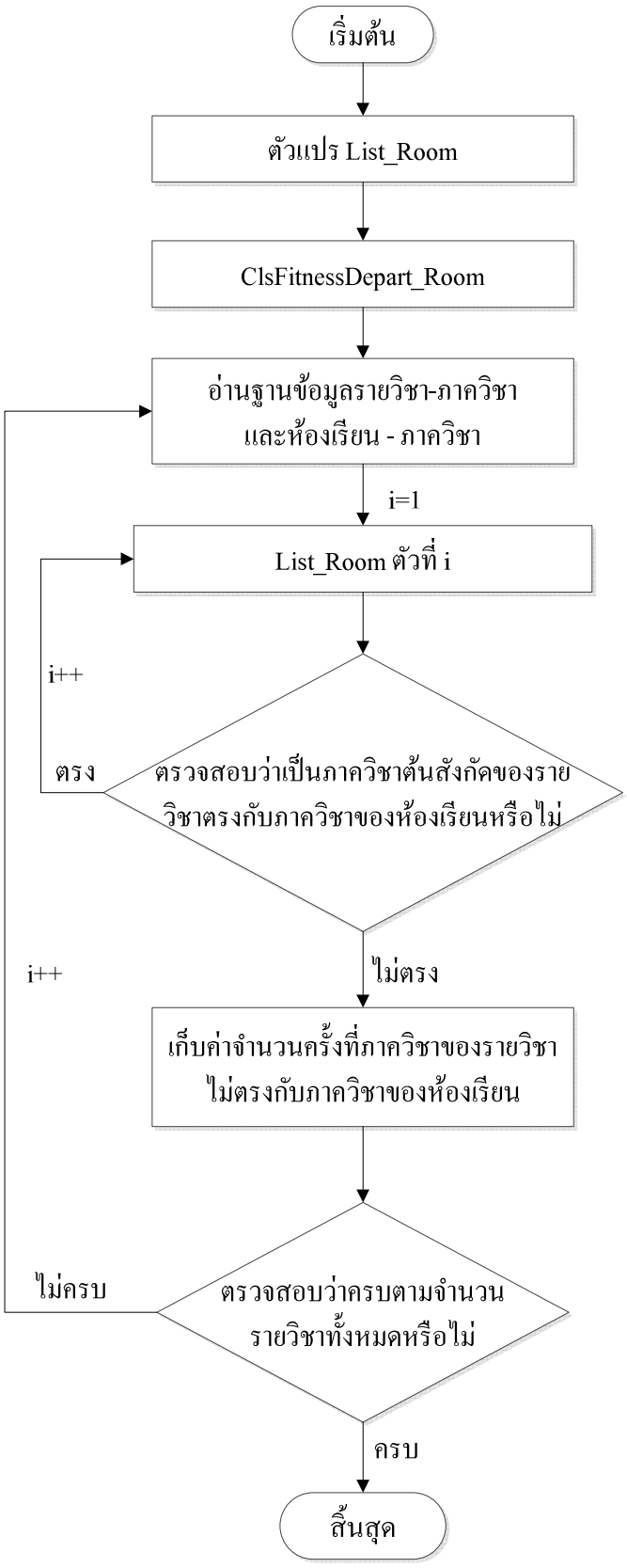
6. ตรวจสอบว่านับจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่าง ครบตามจำนวนรายวิชาทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการแยกเวลาของรายวิชาต่อไป และนับจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่างของรายวิชาดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งในแต่ละวันที่มีการเรียนคาบที่ 2 ของรายวิชาเดิมโดยเว้นคาบว่างของรายวิชาทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป

(8) การคำนวณจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกัน

เงื่อนไขข้อนี้เป็นเงื่อนไขเพื่อความสมบูรณ์ที่สามารถยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ละครั้ง ซึ่งการทำงานของโปรแกรมแสดงดังภาพประกอบ 3.11

สำหรับขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกัน มีขั้นตอนการทำงานดังต่อไปนี้

1. โปรแกรมจะสร้างตัวแปรชื่อ List_Room แล้วส่งค่าตัวแปรนี้ไปยังฟังก์ชันชื่อ ClsFitnessDepart_Room ซึ่งหน้าที่ในการตรวจสอบภาควิชาของรายวิชากับห้องเรียนตรงกันหรือไม่
2. จากนั้นทำการอ่านฐานข้อมูลรายวิชา-ภาควิชา และห้องเรียน-ภาควิชา
3. ตรวจสอบว่าภาควิชาต้นสังกัดของรายวิชาตรงกับภาควิชาของห้องเรียนที่จัดให้หรือไม่ หากตรงกันก็ตรวจสอบรายวิชาถัดไป หากไม่ตรงกันก็เก็บค่าจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกัน
4. ตรวจสอบว่านับจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกันครบตามจำนวนรายวิชาทั้งหมดหรือไม่ หากยังไม่ครบก็ทำการอ่านฐานข้อมูลของรายวิชาต่อไป และนับจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกันของรายวิชาดังกล่าวต่อไป หากครบแล้วก็เก็บค่าจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกันของรายวิชาทั้งหมดเพื่อนำไปคิดค่าความเหมาะสมต่อไป




ภาพประกอบ 3.11 ขั้นตอนการคำนวณจำนวนครั้งที่จัดรายวิชาให้ห้องเรียนที่ภาควิชาไม่ตรงกัน

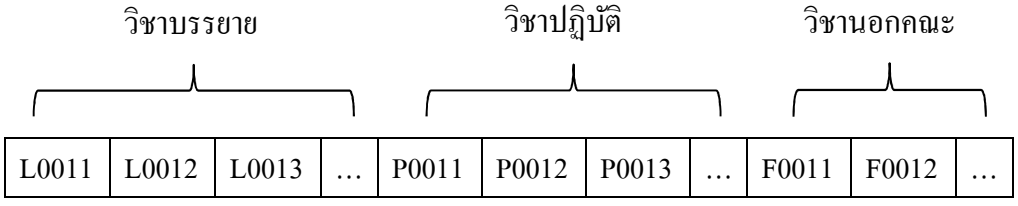
เมื่อทำการตรวจสอบเงื่อนไขในการจัดตารางสอนจนครบทุกเงื่อนไขแล้วผลลัพธ์ที่ได้จะออกมาในรูปแบบของจำนวนครั้งที่เกิดการขัดแย้งของเงื่อนไขในแต่ละข้อ ของแต่ละโครโมโซม จากนั้นนำค่าเหล่านี้ไปคำนวณค่าความเหมาะสมโดยหาผลรวมของการการนำจำนวนครั้งที่เกิดการขัดแย้งของเงื่อนไขในแต่ละข้อคูณกับค่าน้ำหนักของเงื่อนไขข้อนั้นๆ จะได้เป็นค่าความเหมาะสมของโครโมโซม ซึ่งโครโมโซมใดที่มีค่าความเหมาะสมน้อยที่สุดก็จะถือว่าเป็นโครโมโซมที่ดีที่สุด นั้นหมายความว่าโครโมโซมที่จะเป็นโครโมโซมที่ดีจะต้องเป็นโครโมโซมที่ขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อยที่สุด

### 3.3 การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอน

การประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึมเข้ามาช่วยในการแก้ไขปัญหาการจัดตารางสอน เริ่มต้นด้วยการจำลองโครโมโซมนั้นคือ การแปลงคำตอบของปัญหาให้อยู่ในรูปของรหัสโครโมโซม ซึ่งคำตอบของปัญหาการจัดตารางสอนที่กำลังสนใจก็คือ ตารางสอนทั้งหมดของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ดังนั้นรูปแบบการจำลองโครโมโซมที่ออกแบบจะต้องสามารถแสดงตารางสอนในรูปโครโมโซมได้อย่างถูกต้อง ดังนั้นรูปแบบโครโมโซมที่นำมาใช้จึงเป็นรูปแบบการเรียงสับเปลี่ยนรายวิชาทุกรายวิชาที่ต้องการจัดการเรียนการสอนซึ่งอยู่ในรูปของรหัสตัวอักษรที่บ่งบอกประเภทวิชาและตัวเลขที่บ่งบอกรายวิชา โดยตำแหน่งยีนแต่ละตำแหน่งบนโครโมโซมจะบ่งบอกถึงวัน-เวลา และห้องเรียน ที่จะทำการจัดรายวิชาลงไป การแปลงรหัสคำตอบของปัญหาการจัดตารางสอนให้อยู่ในรูปโครโมโซม มีขั้นตอนดังต่อไปนี้

1. โครโมโซม 1 โครโมโซมแทนคำตอบของปัญหา 1 คำตอบนั้นคือ โครโมโซม 1 โครโมโซมจะแทนตารางสอนที่เป็นไปได้ของคณะวิศวกรรมศาสตร์ 1 ตาราง

2. ใน 1 โครโมโซมประกอบด้วยโครโมโซมย่อย (sub-chromosome) 3 โครโมโซมย่อย คือ โครโมโซมย่อยรายวิชาบรรยาย โครโมโซมย่อยรายวิชาปฏิบัติ และโครโมโซมย่อยรายวิชานอกคณะ ซึ่งมีการแบ่งช่วงของโครโมโซมย่อยดังภาพประกอบ 3.12




ภาพประกอบ 3.12 ตัวอย่างโครโมโซม

3. ในโครโมโซมประกอบด้วยยีนเรียงกันอยู่ ซึ่งจำนวนยีนจะเท่ากับจำนวนห้องเรียนคูณกับจำนวนวันที่ใช้ในการจัดการเรียนการสอนคูณกับจำนวนชั่วโมงที่สอนในแต่ละวัน ดังสมการที่ 3.2 ซึ่งข้อมูลที่ใช้ในการจัดตารางสอนของคณะฯ ประกอบด้วยห้องที่เรียนทั้งหมด 59 ห้อง จัดการเรียนการสอนทั้งหมด 5 วัน วันจันทร์-วันศุกร์ ในแต่ละวันมีคาบเรียนทั้งหมด 12 คาบ ตั้งแต่ 8.00-19.50 น. จะมีความยาวโครโมโซมตารางสอนของคณะฯ ทั้งหมดคือ 3,540 ยีน

$$\text{ความยาวโครโมโซม} = \text{จำนวนห้องเรียน} \times \text{จำนวนวัน} \times \text{จำนวนคาบเรียน} \qquad (3.2)$$

4. ตำแหน่งของยีน จะบ่งบอกถึงวัน-เวลาและห้องเรียนที่ใช้ในการจัดตารางสอน ในส่วนของการอธิบายตำแหน่งของยีนจะขอยกตัวอย่าง ดังภาพประกอบ 3.13

การจำลองโครโมโซมจะเริ่มจากการสร้างความสัมพันธ์ระหว่างวัน (แนวตั้ง) ที่ทำการจัดตารางสอน 5 วัน วันจันทร์-วันศุกร์ และเวลา (แนวนอน) ที่จะทำการจัดตารางสอน ในแต่ละวันมีคาบเรียนทั้งหมด 8 คาบตั้งแต่ 8.00-15.50 น. คาบละ 50 นาที แล้วให้ลำดับเป็นเลขจำนวนนับ สามารถให้ความหมายลำดับได้ดังนี้ เช่น ลำดับ 1 หมายถึง วันจันทร์ เวลา 08.00-08.50 น. เป็นต้น

| | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 |
|---|---|---|---|---|---|---|---|---|
| จันทร์ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| อังคาร | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| พุธ | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| พฤหัสบดี | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| ศุกร์ | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |

ภาพประกอบ 3.13 ตัวอย่างวัน-เวลาที่ใช้ในการจัดตารางสอน

จากนั้นจะนำลำดับที่สร้างจากวัน-เวลาที่ใช้ในการจัดตารางสอน (แนวนอน) มาสร้างความสัมพันธ์กับห้องเรียนที่ใช้ในการจัดตารางสอน (แนวตั้ง) ยกตัวอย่างมา 3 ห้องเรียน ดังภาพประกอบ 3.14 จะได้แบบจำลองโครโมโซมที่มีความยาว 120 ยีน สามารถให้ความหมายยีนแต่ละตำแหน่งได้ดังนี้ เช่น ยีนตำแหน่งที่ 1 หมายถึง วันจันทร์ เวลา 08.00-08.50 น. ห้อง A200 เป็นต้น

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | … | 40 |
|---|---|---|---|---|---|---|---|---|---|
| A200 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | … | 40 |
| IELAB | 41 | 42 | 43 | 44 | 45 | 46 | 47 | … | 80 |
| **** | 81 | 82 | 83 | 84 | 85 | 86 | 87 | … | 120 |

ภาพประกอบ 3.14 ตัวอย่างแบบจำลองโครโมโซม

5. ค่าของยีน ในแต่ละยีนจะแสดงด้วยตัวอักษร L, P, F หรือ E แต่ละตัวอักษรมีความหมายดังนี้

- L หมายถึง วิชาบรรยาย
- P หมายถึง วิชาปฏิบัติ
- F หมายถึง วิชานอกคณะ
- E หมายถึง ไม่มีการจัดการเรียนการสอน

หลังตัวอักษรดังกล่าวจะตามด้วยตัวเลข 4 ตำแหน่ง โดยที่ 3 ตำแหน่งแรกแทนรายวิชา และตำแหน่งสุดท้ายแทนชั่วโมงที่ทำการเรียนการสอนของแต่ละรายวิชา เช่น 0011 แทนรายวิชาที่ 1 ชั่วโมงที่ 1, 0012 แทนรายวิชาที่ 1 ชั่วโมงที่ 2 และ 0013 แทนรายวิชาที่ 1 ชั่วโมงที่ 3 เป็นต้น และเพื่อความเข้าใจมากยิ่งขึ้นในขั้นตอนการแปลงรหัสโครโมโซมจึงขอยกตัวอย่างจากข้อมูลรายวิชาที่ต้องจัดตารางสอนตามตาราง 3.5 รายวิชาที่เปิดสอนในภาคการศึกษาที่พิจารณาประกอบด้วยวิชา 3 วิชา คือ วิชาบรรยาย 1 จำนวน 2 ตอน ชั่วโมงการสอนต่อตอน 3 ชั่วโมง, วิชาปฏิบัติ 1 จำนวน 1 ตอน ชั่วโมงการสอนต่อตอน 3 ชั่วโมง และวิชานอกคณะ 1 จำนวน 1 ตอน ชั่วโมงการสอนต่อตอน 3 ชั่วโมง โดยวิชาที่เป็นวิชาเดียวกันแต่มีจำนวนตอนที่เปิดสอนมากกว่า 1 ตอนให้ถือว่าเป็นคนละวิชา และจากข้อมูลดังกล่าวสามารถแปลงข้อมูลเป็นรหัสยีนได้ตามตาราง 3.5 และสามารถสร้างโครโมโซมที่เป็นคำตอบของปัญหาตัวอย่างได้ดังภาพประกอบ 3.15

จากโครโมโซมตัวอย่างสามารถอธิบายได้ว่า โครโมโซมนี้ประกอบด้วยยีนทั้ง 120 ยีน แต่ละยีนบ่งบอกถึงการจัดรายวิชาให้กับวัน-เวลา และห้องเรียน ของแต่ละรายวิชาโดยไม่ซ้ำกัน โดยยีนจะทำการจัดตารางสอนตั้งแต่ยีนลำดับแรกจนถึงลำดับที่ 120 นั้นคือ รหัสยีน L0011 หรือวิชาบรรยาย 1 ชั่วโมงที่ 1 ถูกจัดในวันจันทร์ เวลา 08.00-08.50 น. ห้อง A200, รหัสยีน E000i หมายถึงไม่มีการจัดการเรียนการสอนของรายวิชาใดในช่วงวัน-เวลา และห้องเรียนนั้น, รหัสยีน L0021 หรือวิชาบรรยาย 2 ชั่วโมงที่ 1 ถูกจัดในวันจันทร์ เวลา 10.00-10.50 น. ห้อง A200 ทำการจัดรายวิชาตามลำดับยีนเช่นนี้ไปจนถึงยีนลำดับสุดท้าย จะได้ตารางสอนของรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษาที่ต้องการจัดตารางสอน

ตาราง 3.5 รายละเอียดรายวิชาที่เปิดสอนในภาคเรียนที่พิจารณา

| รหัสวิชา | ชื่อวิชา | หน่วยกิต | ตอนที่เปิดสอน | ชั่วโมงสอนที่ | กลุ่มนักศึกษา | อาจารย์ผู้สอน | รหัสยีน |
|---|---|---|---|---|---|---|---|
| 111-AAA | บรรยาย 1 | 3 | 1 | 1 | Stu1 | T1 | L0011 |
| | | | 1 | 2 | Stu1 | T1 | L0012 |
| | | | 1 | 3 | Stu1 | T1 | L0013 |
| 111-AAA | บรรยาย 1 | | 2 | 1 | Stu2 | T1 | L0021 |
| | | | 2 | 2 | Stu2 | T1 | L0022 |
| | | | 2 | 3 | Stu2 | T1 | L0023 |
| 222-AAA | ปฏิบัติ 1 | 1 | 1 | 1 | Stu1 | T2 | P0011 |
| | | | 1 | 2 | Stu1 | T2 | P0012 |
| | | | 1 | 3 | Stu1 | T2 | P0013 |
| 333-AAA | นอกคณะ 1 | 3 | 1 | 1 | Stu1,2 | T3 | F0011 |
| | | | 1 | 2 | Stu1,2 | T3 | F0012 |
| | | | 1 | 3 | Stu1,2 | T3 | F0013 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| L0011 | E0001 | L0021 | L0022 | E0002 | L0023 | L0013 | E0003 | L0012 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | … | 40 |
| E0004 | P0011 | P0012 | P0013 | E0005 | E0006 | E0007 | E0008 | E0009 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | … | 80 |
| F0011 | E0010 | F0012 | E0011 | E0012 | E0013 | E0014 | E0015 | F0013 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | … | 120 |

ภาพประกอบ 3.15 ตัวอย่างโครโมโซมที่สร้างจากข้อมูลตัวอย่าง

เมื่อแปลงคำตอบของปัญหาให้อยู่ในรูปโครโมโซมได้แล้วขั้นตอนต่อไปคือการสร้างคำตอบเบื้องต้น สำหรับงานวิจัยนี้มีแนวคิดในการสร้างคำตอบเบื้องต้นของปัญหาด้วยกัน 2 แนวคิด แต่ละแนวคิดมีข้อดีข้อเสียดังนี้

(1) แนวคิดที่ 1 ตรวจสอบความเป็นไปได้ของคำตอบเบื้องต้นก่อนที่เข้าสู่กระบวนการเจเนติกอัลกอริทึม หมายความว่าคำตอบที่ได้ต้องไม่มีเวลาสอนของอาจารย์ซ้ำซ้อนกัน เวลาเรียนของกลุ่มนักศึกษาไม่ซ้ำซ้อนกัน เวลาการใช้ห้องเรียนไม่ซ้ำซ้อนกัน แนวคิดนี้มีข้อดีคือได้คำตอบที่เป็นไปได้ทันที ข้อเสียคือ เมื่อคำตอบเหล่านี้เข้าสู่กระบวนการเจเนติกอัลกอริทึมก็อาจกลายเป็นคำตอบที่เป็นไปไม่ได้อีก ทำให้ต้องเสียเวลาในการปรับปรุงคำตอบให้เป็นไปได้ อีกทั้งแนวคิดยังใช้เวลาในการประมวลผลนานเนื่องจากต้องพยายามสร้างคำตอบให้เป็นไปได้ตั้งแต่กระบวนการเริ่มต้น

(2) แนวคิดที่ 2ไม่ต้องตรวจสอบความเป็นไปได้ของคำตอบเบื้องต้น แต่กำหนดค่าน้ำหนักของเงื่อนไขบังคับให้สูง ด้วยสมมุติฐานที่ว่าคำตอบที่เป็นไปไม่ได้จะไม่ถูกคัดเลือกมาเป็นคำตอบของปัญหาด้วยกระบวนการเจเนติกอัลกอริทึม ข้อดีของแนวคิดนี้คือ ใช้เวลาในการประมวลผลน้อย แต่ข้อเสียคือ อาจจะไม่มีโอกาสพบคำตอบที่เป็นไปได้ หรือต้องอาศัยการพัฒนาตำตอบหลายเจนเนอเรชันจึงจะพบคำตอบที่เป็นไปได้

งานวิจัยนี้ได้เลือกแนวคิดที่ 2 มาเป็นวิธีการในการสร้างคำตอบเบื้องต้น โดยสร้างคำตอบเบื้องต้นตามจำนวนประชากรเบื้องต้นที่กำหนด เช่น กำหนดจำนวนประชากรเบื้องต้นเท่ากับ 20 ก็จะทำการสุ่มโครโมโซมขึ้นมา 20 โครโมโซม ซึ่งจะอธิบายอย่างละเอียดในหัวข้อ 3.3.1 จากนั้นเข้าสู่ขั้นตอนการรีโปรดักชันซึ่งเป็นขั้นตอนการคัดเลือกโครโมโซมที่มีความเหมาะสมเพื่อเข้าสู่ขั้นตอนต่อไปของวิธีการของเจเนติกอัลกอริทึม การรีโปรดักชันประกอบด้วยขั้นตอนย่อย คือขั้นตอนการถอดรหัสคำตอบ การคำนวณค่าความเหมาะสม และการคัดเลือก จะอธิบายรายละเอียดในหัวข้อ 3.3.2 โครโมโซมที่ผ่านการคัดเลือกจะถูกนำมาจับคู่เพื่อทำการครอสโอเวอร์โดยการแลกเปลี่ยนยีนบางส่วนในโครโมโซมพ่อแม่ เพื่อให้ได้โครโมโซมลูกเบื้องต้น จะอธิบายรายละเอียดในหัวข้อ 3.3.3 และเข้าสู่ขั้นตอนการมิวเตชันที่มีรายละเอียดในหัวข้อ 3.3.4 ต่อไป เมื่อได้โครโมโซมชุดใหม่ที่ผ่านขั้นตอนการครอสโอเวอร์และมิวเตชันมาแล้ว ก็จะคำนวณค่าความเหมาะสมอีกครั้งและบันทึกค่าความเหมาะสมนั้นไว้ เป็นอันเสร็จสิ้นขั้นตอนเจเนติกอัลกอริทึมในรุ่นที่หนึ่ง จากนั้นโปรแกรมจะทำการประมวลผลซ้ำตามขั้นตอนเจเนติกอัลกอริทึมดังที่ได้กล่าวไปแล้วจนครบตามจำนวนรุ่นที่กำหนดไว้ ซึ่งในแต่ละรุ่นก็จะมีการบันทึกค่าความ

เหมาะสมที่ดีที่สุดเอาไว้ ทำให้คำตอบที่ได้เป็นคำตอบที่ดีขึ้นเรื่อยๆ และสุดท้ายจะได้คำตอบที่เหมาะสมที่สุดสำหรับปัญหาซึ่งจะแสดงผลลัพธ์เป็นตารางสอนที่ขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อยที่สุด ขั้นตอนการประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึมแสดงได้ดังภาพประกอบ 3.16 ซึ่งมีรายละเอียดของแต่ละขั้นตอนสำหรับปัญหาการจัดตารางสอน ดังนี้

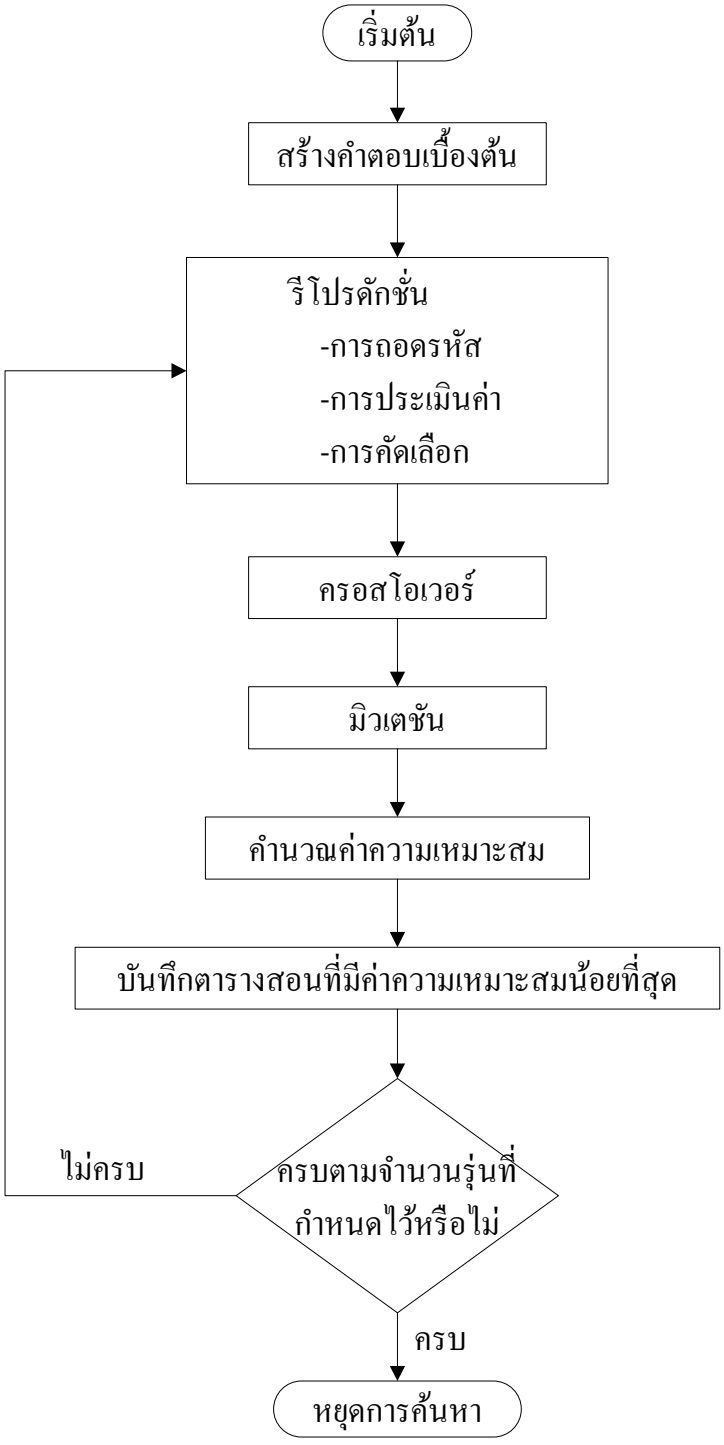
### 3.3.1 การสร้างคำตอบเบื้องต้น (initialization)

เมื่อได้รูปแบบโครโมโซมที่เหมาะสมสำหรับปัญหาแล้ว ก็จะทำการสร้างคำตอบเบื้องต้นหรือการสร้างโครโมโซมดังภาพประกอบ 3.17 ด้วยการสุ่มประชากรเบื้องต้นขึ้นมาจำนวนหนึ่งเพื่อนำไปใช้ในกระบวนการของเจเนติกอัลกอริทึม โครโมโซม 1 ตัว คือประชากร 1 ตัว จำนวนประชากรเบื้องต้นที่ต้องการสร้างนั้นเป็นพารามิเตอร์ตัวหนึ่งที่จะต้องมีการกำหนดให้เหมาะสมกับปัญหา ขั้นตอนการสร้างโครโมโซมจะเริ่มจากอ่านข้อมูลรายวิชา วัน-เวลา และห้องเรียนจากฐานข้อมูล แล้วคำนวณความยาวของโครโมโซมจากสมการความยาวโครโมโซมที่กล่าวมาข้างต้น เพื่อสร้างแบบจำลองโครโมโซม จากนั้นทำการสุ่มตำแหน่งยีน และแทนรายวิชาลงในตำแหน่งยีนที่สุ่มได้ ทำเช่นนี้จนครบทุกรายวิชาที่ต้องการจัดตารางสอน จะได้โครโมโซม 1 โครโมโซม และทำจนครบตามจำนวนประชากรเบื้องต้นที่กำหนดไว้

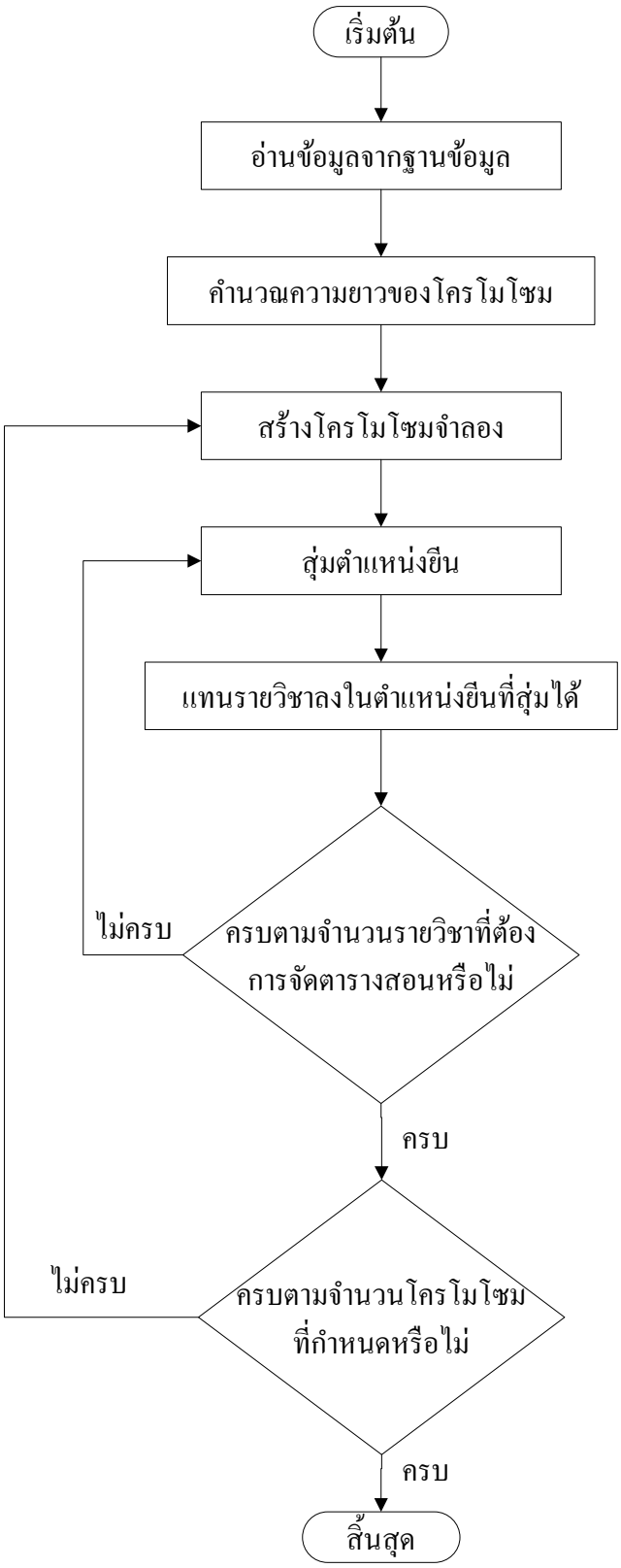
สมมติ กำหนดให้จำนวนประชากรเบื้องต้นเท่ากับ 5 ก็จะต้องทำการสุ่มโครโมโซมขึ้นมาจำนวน 5 โครโมโซม ตัวอย่างโครโมโซมที่สุ่มมาได้เป็นดังนี้

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C1 : | L0011 | E0001 | L0021 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| C2 : | L0031 | L0041 | L0022 | E0001 | E0002 | … | P0011 | P0012 | P0013 |
| C3 : | L0032 | L0033 | E0001 | L0012 | L0011 | … | P0021 | P0022 | P0023 |
| C4 : | L0013 | L0023 | E0001 | L0021 | L0011 | … | P0041 | P0042 | P0043 |
| C5 : | E0001 | L0041 | L0013 | E0002 | E0003 | … | P0031 | P0032 | P0033 |

เมื่อได้โครโมโซมตามจำนวนประชากรเบื้องต้นที่กำหนดแล้วก็จะนำโครโมโซมเหล่านี้เข้าสู่ขั้นตอนต่อไปของวิธีการของเจเนติกอัลกอริทึม คือขั้นตอนการรีโปรดักชัน




ภาพประกอบ 3.16 ขั้นตอนการประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึม




ภาพประกอบ 3.17 การสร้างโครโมโซม

### 3.3.2 การรีโปรดักชัน (reproduction)

การรีโปรดักชันเป็นขั้นตอนการสร้างโครโมโซมตัวใหม่ขึ้นมาโดยใช้กลุ่มคำตอบพื้นฐานตามโครโมโซมที่เกิดขึ้นในกลุ่มประชากรเบื้องต้น ประกอบด้วย 3 ขั้นตอนย่อยคือ การถอดรหัสคำตอบ การคำนวณค่าความเหมาะสม และการคัดเลือก สำหรับปัญหาการจัดตารางสอนมีรายละเอียดของแต่ละขั้นตอนย่อยดังนี้

(1) การถอดรหัสคำตอบ (decoding)

การถอดรหัสคำตอบ คือการแปลงโครโมโซมให้เป็นตารางสอนสำหรับคณะวิศวกรรมศาสตร์สำหรับ 1 ภาคเรียนโดยการนำห้องเรียน วัน และเวลาไปจัดให้กับรายวิชาที่เปิดสอนตามลำดับเช่น จากตัวอย่างโครโมโซม 1 ที่สุ่มมาได้ คือ

L0011 E0001 L0021 L0022 E0002 … P0011 P0012 P0013… F0011 F0012 F0013

ยีน L0011 คือห้อง A200 วันจันทร์ 08.00-08.50 น. ถูกจัดให้กับวิชา 111-AAA ตอนที่ 1 ชั่วโมงที่ 1 รหัสยีน E0001 หมายถึงไม่มีการจัดการเรียนการสอนของรายวิชาใดในช่วงวัน-เวลา และห้องเรียนนั้น, รหัสยีน L0021 หรือวิชาบรรยาย 2 ชั่วโมงที่ 1 ถูกจัดในวันจันทร์ เวลา 10.00-10.50 น. ห้อง A200 ทำการจัดรายวิชาตามลำดับยีนเช่นนี้ไปจนถึงยีนลำดับสุดท้าย ไปเรื่อยๆ จนถึงยีนสุดท้าย คือยีน F0013 รายวิชา 333-AAA ตอนที่ 1 ชั่วโมงที่ 3 ซึ่งเป็นวิชานอกคณะจะเอาเวลาจากที่กองทะเบียนและประมวลผลจัดมาใส่นั้นคือ วันศุกร์ เวลา 09.00-09.50 น. และจะเป็นห้อง *** (ซึ่งไม่พิจารณาร่วมกับห้องอื่นในการจัดตารางสอน) เป็นต้น ดังแสดงรายละเอียดตามตาราง 3.6 และสามารถแปลงเป็นตารางสอนได้ดังภาพประกอบ 3.18

(2) การคำนวณค่าความเหมาะสม (fitness)

การคำนวณค่าความเหมาะสมของปัญหาการจัดตารางสอนนี้ ก็คือการคำนวณค่าความเหมาะสมที่ได้จากการจัดตารางสอนที่มีการขัดแย้งกับเงื่อนไขในการจัดตารางสอนนั่นเอง ซึ่งฟังก์ชันวัตถุประสงค์ของงานวิจัยนี้เป็นการจัดตารางสอนเพื่อให้ขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อยที่สุด

(3) การคัดเลือก (selection)

เมื่อคำนวณค่าความเหมาะสมของแต่ละโครโมโซมจนครบตามจำนวนประชากรเบื้องต้นที่กำหนดแล้ว ขั้นตอนสุดท้ายของการรีโปรดักชันก็คือการคัดเลือก ซึ่งวิธีการคัดเลือกนี้จะพิจารณาจากค่าความเหมาะสมของแต่ละโครโมโซม โดยโครโมโซมที่มีค่าความเหมาะสมมากจะมีโอกาสถูกคัดเลือกได้มากกว่าโครโมโซมที่มีค่าความเหมาะสมน้อย ในงานวิจัยนี้โครโมโซมที่มี

โอกาสถูกเลือกมากคือโครโมโซมที่มีค่าความเหมาะสมน้อยนั่นคือมีการขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อย การคิดค่าความเหมาะสมจึงคิดเป็นส่วนกลับของค่าความเหมาะสม

ตาราง 3.6 รายละเอียดของรายวิชาที่จัดตารางสอน

| **รหัสวิชา** | **ชื่อวิชา** | **ตอนที่เปิดสอน** | **ชั่วโมงที่สอน** | **กลุ่มนักศึกษา** | **อาจารย์ผู้สอน** | **ห้องเรียน** | **วัน-เวลา** |
|---|---|---|---|---|---|---|---|
| 111-AAA | บรรยาย 1 | 1 | 1 | Stu1 | T1 | A200 | จันทร์ 08.00-08.50 |
| | | 1 | 2 | Stu1 | T1 | A200 | พุธ 08.00-08.50 |
| | | 1 | 3 | Stu1 | T1 | A200 | ศุกร์ 08.00-08.50 |
| 111-AAA | บรรยาย 1 | 2 | 1 | Stu2 | T1 | A200 | จันทร์ 10.00-10.50 |
| | | 2 | 2 | Stu2 | T1 | A200 | จันทร์ 11.00-11.50 |
| | | 2 | 3 | Stu2 | T1 | A200 | พุธ 15.00-15.50 |
| 222-AAA | ปฏิบัติ 1 | 1 | 1 | Stu1 | T2 | IELAB | จันทร์ 13.00-13.50 |
| | | 1 | 2 | Stu1 | T2 | IELAB | จันทร์ 14.00-14.50 |
| | | 1 | 3 | Stu1 | T2 | IELAB | จันทร์ 15.00-15.50 |
| 333-AAA | นอกคณะ 1 | 1 | 1 | Stu1,2 | T3 | *** | จันทร์ 09.00-09.50 |
| | | 1 | 2 | Stu1,2 | T3 | *** | พุธ 09.00-09.50 |
| | | 1 | 3 | Stu1,2 | T3 | *** | ศุกร์ 09.00-09.50 |

| | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 |
|---|---|---|---|---|---|---|---|---|
| จันทร์ | 111-AAA(1)<br>A200<br>T1, Stu1 | 333-AAA(1)<br>***<br>T3, Stu1,2 | 111-AAA(2)<br>A200<br>T1, Stu2 | 111-AAA(2)<br>A200<br>T1, Stu2 | | 222-AAA(1)<br>IELAB<br>T2, Stu1 | | |
| อังคาร | | | | | | | | |
| พุธ | 111-AAA(1)<br>A200<br>T1, Stu1 | 333-AAA(1)<br>***<br>T3, Stu1,2 | | | | | | 111-AAA(2)<br>A200<br>T1, Stu2 |
| พฤหัสบดี | | | | | | | | |
| ศุกร์ | 111-AAA(1)<br>A200<br>T1, Stu1 | 333-AAA(1)<br>***<br>T3, Stu1,2 | | | | | | |

ภาพประกอบ 3.18 ตารางสอนที่ได้จากการเข้ารหัสโครโมโซม

วิธีการคัดเลือกที่ใช้คือวิธีการคัดเลือกแบบวงล้อรูเล็ท (roulette wheel) เปรียบเทียบกับวิธีการคัดลือกแบบ Tournament ซึ่งขออธิบายวิธีการคัดเลือกแบบวงล้อรูเล็ทก่อนดังนี้

(3.1) วิธีการคัดเลือกแบบวงล้อรูเล็ท

วิธีการนี้เริ่มจากการหาค่าความเหมาะสมรวม (sum fitness) ของโครโมโซมทั้งหมดจากผลรวมของส่วนกลับค่า fitness value ของโครโมโซมแต่ละตัว ค่าความเหมาะสมรวมสามารถคำนวณได้จากสมการที่ 3.3

$$\text{sum fitness} = \sum_{i=1}^{\text{popsize}} \left(\frac{1}{\text{fitness}_i}\right) \qquad (3.3)$$

โดยที่

$\text{fitness}_i$ คือ ค่าความเหมาะสมของโครโมโซมตัวที่ i

i คือ ลำดับที่ของโครโมโซม (= 1, 2, 3, …, popsize)

popsize คือ จำนวนประชากรเบื้องต้นที่กำหนด

จากนั้นทำการหาค่าความน่าจะเป็นในการถูกคัดเลือก (selection of probability) ของโครโมโซมแต่ละตัวคำนวณได้จากส่วนกลับของค่าความเหมาะสมของโครโมโซมแต่ละ

ตัวหารด้วยผลรวมส่วนกลับของค่าความเหมาะสมของโครโมโซมทั้งหมด เนื่องจากค่าความเหมาะสมคือมีการขัดแย้งกับเงื่อนไขในการจัดตารางสอนน้อย ดังนั้นยิ่งค่าความเหมาะสมมีค่าต่ำก็จะยิ่งทำให้ค่าความน่าจะเป็นในการถูกคัดเลือกมีค่ามาก ซึ่งคำนวณได้จากสมการที่ 3.4

$$P_i = \frac{(\frac{1}{\text{fitness}_i})}{sum\ \text{fitness}} \tag{3.4}$$

โดยที่

$P_i$ คือ ค่าความน่าจะเป็นในการถูกคัดเลือกของโครโมโซมที่ i

i คือ ลำดับที่ของโครโมโซม (i = 1, 2, 3, …, popsize)

จากนั้นหาความน่าจะเป็นในการถูกคัดเลือกสะสม (cumulative of probability) ของโครโมโซมแต่ละตัว ซึ่งคำนวณได้จากสมการที่ 3.5

$$\text{Cum}_i = \sum_{i=1}^{popsize} P_i \tag{3.5}$$

โดยที่

$\text{Cum}_i$ คือ ค่าความน่าจะเป็นในการถูกคัดเลือกสะสมของโครโมโซม i

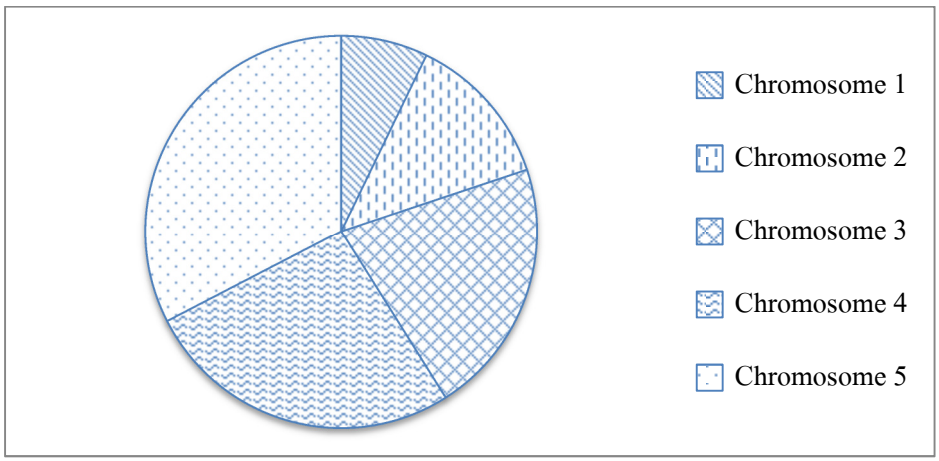
i คือ ลำดับที่ของโครโมโซม (i = 1, 2, 3, …, popsize)

popsize คือ จำนวนประชากรเบื้องต้นที่กำหนด

จากนั้นจึงนำค่าความน่าจะเป็นในการถูกคัดเลือกของโครโมโซมแต่ละตัวไปสร้างวงล้อรูเล็ทโดยโครโมโซมทั้งหมดที่อาศัยอยู่ในขนาดพื้นที่ของวงล้อ คือสัดส่วนของค่าความเหมาะสมของทุกโครโมโซม ค่าความเหมาะสมที่มากที่สุดคือพื้นที่ที่มากที่สุดในส่วนของวงล้อ เมื่อมีการหมุนวงล้อโครโมโซมที่มีค่าความเหมาะสมมากจะมีโอกาสถูกเลือกได้มาก ตัวอย่างการคำนวณค่าความน่าจะเป็นในการถูกคัดเลือก และค่าความน่าจะเป็นในการถูกคัดเลือกสะสมของแต่ละโครโมโซม เป็นดังตาราง 3.7 ซึ่งแสดงตัวอย่างค่าที่ได้จากการคำนวณหาค่าความน่าจะเป็นในการถูกคัดเลือกจากสมการที่ 3.4 และค่าที่ได้จากการคำนวณค่าความน่าจะเป็นในการถูกคัดเลือกสะสมจากสมการที่ 3.5 จากนั้นนำข้อมูลตัวอย่างจากตารางดังกล่าวไปสร้างวงล้อรูเล็ทได้ดังภาพประกอบ 3.19 ซึ่งแสดงสัดส่วนของค่าความเหมาะสมของโครโมโซมแต่ละตัวโดยโครโมโซมที่มีค่าความเหมาะสมมากจะมีสัดส่วนพื้นที่ในวงกลมมากตามไปด้วย

ตาราง 3.7 ตัวอย่างการคำนวณค่าความน่าจะเป็นในการถูกคัดเลือก และค่าความน่าจะเป็นในการถูกคัดเลือกสะสม

| โครโมโซม | ค่าความเหมาะสม | 1/ค่าความเหมาะสม | ความน่าจะเป็น | ความน่าจะเป็นสะสม |
|---|---|---|---|---|
| 1 | 1824 | 0.00055 | 0.22 | 0.22 |
| 2 | 2238 | 0.00044 | 0.17 | 0.39 |
| 3 | 1538 | 0.00065 | 0.26 | 0.65 |
| 4 | 2429 | 0.00041 | 0.16 | 0.81 |
| 5 | 2026 | 0.00049 | 0.19 | **1.00** |
| **รวม** | **10055** | **0.00254** | **1.00** | |




ภาพประกอบ 3.19 ตัวอย่างการสร้างวงล้อรูเล็ทจากข้อมูลตัวอย่างในตาราง 3.7

ประชากรเบื้องต้น โครโมโซมที่มีความเหมาะสมมาก จะมีโอกาสถูกเลือกมาก ทำให้โครโมโซมชุดใหม่ที่ได้มีค่าความเหมาะสมโดยเฉลี่ยต่ำกว่าโครโมโซมชุดเดิม ขั้นตอนการคัดเลือกโครโมโซมชุดใหม่นี้จะใช้วิธีการสุ่มตัวเลข (0-1) ขึ้นมาตามจำนวนประชากรเบื้องต้นที่กำหนด จากนั้นนำตัวเลขสุ่มแต่ละตัวไปเทียบกับค่าความน่าจะเป็นในการถูกคัดเลือกสะสมในวงล้อรูเล็ท และตรวจสอบค่าตัวเลขสุ่มว่าตกอยู่ในพื้นที่ของโครโมโซมตัวใด โครโมโซมตัวนั้นก็จะถูกเลือกมาเป็นโครโมโซมชุดใหม่ ตัวอย่างข้อมูลตัวเลขสุ่มเพื่อใช้ในการคัดเลือกโครโมโซมเป็นดัง

ตาราง 3.8 ซึ่งจะใช้ตัวเลขสุ่มนี้เปรียบเทียบกับค่าความน่าจะเป็นในการคัดเลือกสะสมว่าตกอยู่ในช่วงของโครโมโซมตัวใดโครโมโซมนั้นก็จะถูกเลือกมา ทำการสุ่มตัวเลขเพื่อคัดเลือกโครโมโซมชุดใหม่จนครบตามจำนวนประชากรเบื้องต้นที่กำหนด

ตาราง 3.8 ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมชุดใหม่

| ตัวเลขสุ่ม | ค่าอยู่ในช่วงของความน่าจะเป็นสะสม | โครโมโซมที่ถูกเลือก |
|---|---|---|
| 0.45 | 0.39-0.65 | 3 |
| 0.19 | 0-0.22 | 1 |
| 0.69 | 0.65-0.81 | 4 |
| 0.77 | 0.65-0.81 | 4 |
| 0.32 | 0.22-0.39 | 2 |

จากตาราง 3.8 โครโมโซมชุดใหม่ที่ถูกเลือกด้วยวิธีการคัดเลือกแบบวงล้อรูเล็ทได้แก่โครโมโซม 3, 1, 4, 4 และ 2 ดังนี้

C3 : L0032 L0033 E0001 L0012 L0011 … P0021 P0022 P0023

C1 : L0011 E0001 L0021 L0022 E0002 … P0011 P0012 P0013

C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043

C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043

C2 : L0031 L0041 L0022 E0001 E0002 … P0011 P0012 P0013

(3.2) วิธีการคัดเลือกแบบ Tournament

ส่วนวิธีการคัดเลือกแบบ Tournament ที่ใช้ในการคัดเลือกโครโมโซมเพื่อเข้าสู่ขั้นตอนของวิธีการของเจเนติกอัลกอริทึม วิธีนี้ขั้นตอนเริ่มต้นเหมือนกับวิธีการคัดเลือกแบบวงล้อรูเล็ทนั้นคือกระทำตามขั้นตอนของวิธีการคัดเลือกแบบวงล้อรูเล็ทจนกระทั่งได้วงล้อรูเล็ท ดังภาพประกอบ 3.19 ขั้นตอนของทั้ง 2 วิธีนี้จะต่างกันตรงที่วิธีการคัดเลือกแบบ Tournament จะทำการสุ่มตัวเลขมา 2 ครั้งแล้วคัดเลือกมา 2 โครโมโซม แล้วทำการเปรียบเทียบค่าความเหมาะสมระหว่าง 2 โครโมโซมนี้เพื่อให้ได้มา 1 โครโมโซมที่มีค่าความเหมาะสมที่ดีกว่าอีกโครโมโซมหนึ่ง ทำเช่นนี้จนกระทั่งได้โครโมโซมครบตามจำนวนประชากรที่กำหนด ตามตาราง 3.9

ตาราง 3.9 ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมชุดใหม่ด้วยวิธี Tournament

| **ครั้งที่ 1** | | | **ครั้งที่ 2** | | | **แข่งขัน** |
|---|---|---|---|---|---|---|
| **ตัวเลขสุ่ม** | **ช่วงความน่าจะเป็นสะสม** | **โครโมโซมที่ถูกเลือก** | **ตัวเลขสุ่ม** | **ช่วงความน่าจะเป็นสะสม** | **โครโมโซมที่ถูกเลือก** | **โครโมโซมที่ถูกเลือก** |
| 0.45 | 0.39-0.65 | 3 | 0.16 | 0-0.22 | 1 | 3 |
| 0.19 | 0-0.22 | 1 | 0.52 | 0.39-0.65 | 3 | 3 |
| 0.69 | 0.65-0.81 | 4 | 0.70 | 0.65-0.81 | 4 | 4 |
| 0.77 | 0.65-0.81 | 4 | 0.38 | 0.39-0.65 | 2 | 4 |
| 0.32 | 0.22-0.39 | 2 | 0.54 | 0.39-0.65 | 3 | 3 |

จากตาราง 3.9 โครโมโซมชุดใหม่ที่ถูกเลือกด้วยวิธีการคัดเลือกแบบ Tournament ได้แก่โครโมโซม 3, 3, 4, 4 และ 3 ดังนี้

C3 : L0032 L0033 E0001 L0012 L0011 … P0021 P0022 P0023

C3 : L0032 L0033 E0001 L0012 L0011 … P0021 P0022 P0023

C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043

C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043

C3 : L0032 L0033 E0001 L0012 L0011 … P0021 P0022 P0023

จากการเปรียบเทียบวิธีการคัดเลือกทั้ง 2 วิธีสามารถสรุปข้อดีข้อเสียของแต่ละวิธีได้ตามตาราง 3.10 ดังนี้

ตาราง 3.10 ข้อดี-ข้อเสียของวิธีการคัดเลือกแบบวงล้อรู้เล็ทกับวิธีการคัดเลือกแบบ Tournament

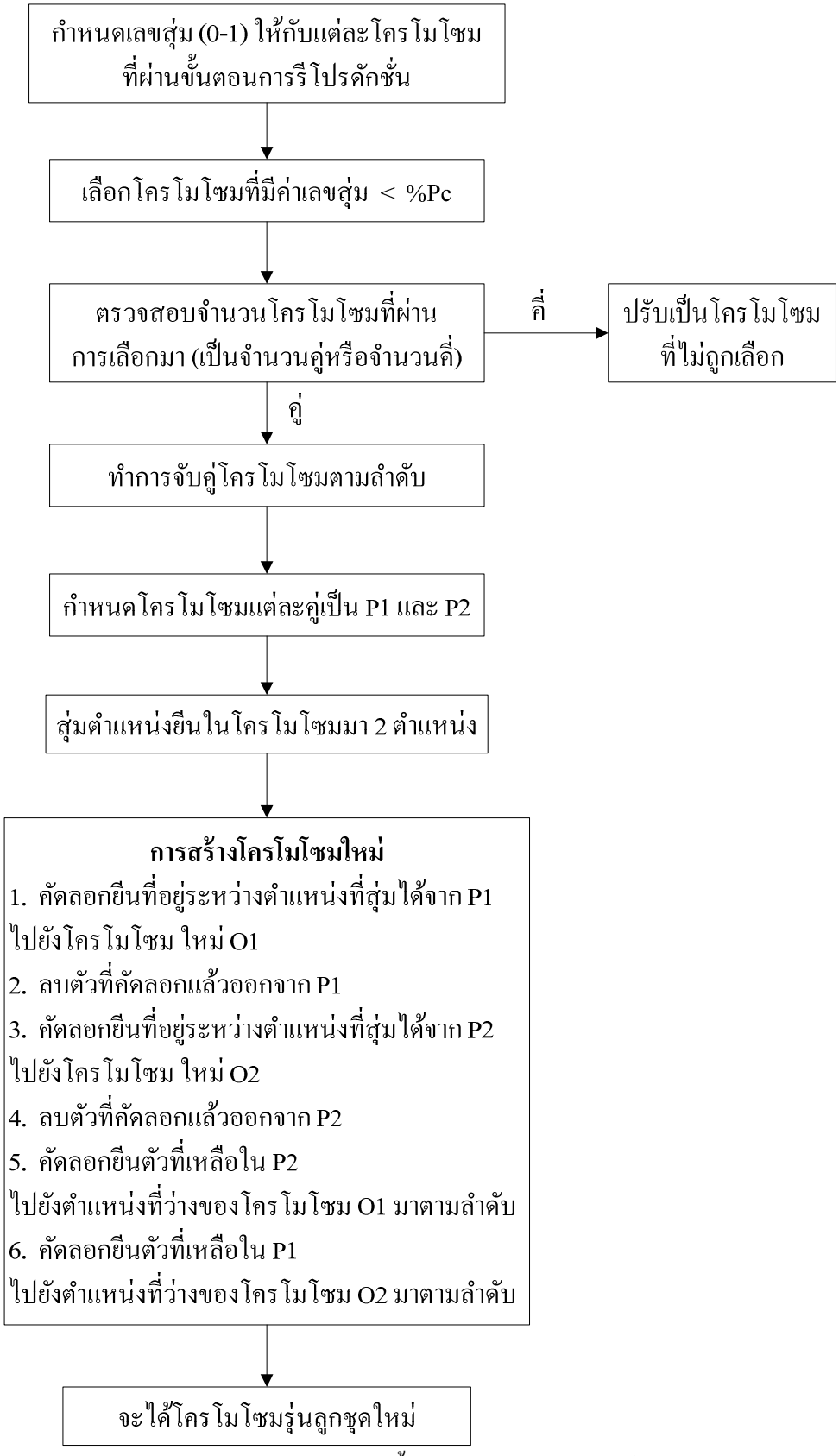
| **วิธีการคัดเลือก** | **ข้อดี** | **ข้อเสีย** |
|---|---|---|
| วงล้อรูเล็ท | - กระบวนง่าย ไม่ยุ่งยากซับซ้อน<br>- ได้โครโมโซมที่มีความหลากหลาย | - ใช้เวลาในการประมวลผลนาน เนื่องจากโครโมโซมมีความหลากหลาย |
| Tournament | - สามารถลดเวลาในการประมวลผลได้ | - ความหลากหลายของโครโมโซมที่ถูกคัดเลือกน้อย |

ดังนั้นงานวิจัยนี้จึงได้เลือกใช้วิธีการคัดเลือกแบบวงล้อรูเล็ทเนื่องจากเป็นกระบวนที่ง่าย ไม่ยุ่งยากซับซ้อน และได้โครโมโซมที่มีความหลากหลายกว่าวิธีการคัดเลือกแบบ Tournament ซึ่งส่งผลให้ได้คำตอบที่ดีไปด้วย ถึงแม้จะเสียเวลาในการประมวลผลไปบ้าง จากโครโมโซมชุดใหม่ที่ถูกเลือกมานี้จะมีโครโมโซมเพียงบางส่วนที่จะถูกเลือกมากกว่า 1 ครั้ง เพื่อเข้าสู่ขั้นตอนต่อไปของวิธีการของเจเนติกอัลกอริทึมคือ ขั้นตอนการครอสโอเวอร์

### 3.3.3 การครอสโอเวอร์ (crossover)

การครอสโอเวอร์เป็นขั้นตอนที่ทำหลังจากขั้นตอนการรีโปรดักชันโดยการแลกเปลี่ยนบางส่วนของโครโมโซมพ่อแม่ (parent) ซึ่งจากโครโมโซมชุดใหม่ที่ได้จากการคัดเลือกนั้นจะมีเพียงโครโมโซมบางส่วนเท่านั้นที่จะถูกคัดเลือกมาตามอัตราความน่าจะเป็นในการครอสโอเวอร์ ($P_c$) เพื่อสร้างโครโมโซมรุ่นใหม่หรือโครโมโซมรุ่นลูก (offspring) ชุดใหม่ขึ้นมาโดยใช้วิธีการครอสโอเวอร์แบบลำดับ (Order Crossover : OX) เปรียบเทียบกับวิธีการครอสโอเวอร์แบบค่าน้ำหนัก (Weight Mapping Crossover : WMX) กำหนดให้ค่าความน่าจะเป็นในการครอสโอเวอร์เริ่มต้นเท่ากับ 0.8 ขั้นตอนแสดงการคัดเลือกโครโมโซมเพื่อทำการครอสโอเวอร์เป็นดังภาพประกอบ 3.20

ขั้นตอนการครอสโอเวอร์เริ่มต้นจากการกำหนดเลขสุ่ม (0-1) ให้กับแต่ละโครโมโซมที่ผ่านขั้นตอนการรีโปรดักชัน ซึ่งได้แก่โครโมโซม C3, C1, C4, C4 และ C2 มีค่าตัวเลขสุ่มคือ 0.89, 0.41, 0.22, 0.64 และ 0.35 ตามลำดับดังตัวอย่างในตาราง 3.11 จากนั้นก็พิจารณาเลขสุ่มของโครโมโซมแต่ละตัวและเลือกโครโมโซมที่มีค่าเลขสุ่มน้อยกว่าค่าความน่าจะเป็นในการครอสโอเวอร์ที่กำหนดไว้ คือ 0.8 เช่น โครโมโซม C1, C4, C4 และ C2 มีค่าเลขสุ่ม 0.41, 0.22, 0.64 และ 0.35 ตามลำดับซึ่งเป็นค่าที่น้อยกว่าค่า 0.8 ดังนั้นโครโมโซม C1, C4, C4 และ C2 จึงถูกเลือกมาเพื่อทำการครอสโอเวอร์ ส่วนโครโมโซมที่ไม่ถูกเลือกเพื่อทำการครอสโอเวอร์มีเพียงตัวเดียวคือโครโมโซม C3 เนื่องจากมีค่าเลขสุ่ม 0.89 ซึ่งมากกว่า 0.8 จากนั้นทำการจับคู่โครโมโซมที่ถูกเลือกมาทีละคู่ตามลำดับแล้วกำหนดเป็นโครโมโซมพ่อแม่ (P1,P2)




ภาพประกอบ 3.20 ขั้นตอนการครอสโอเวอร์

ตาราง 3.11 ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมเพื่อทำการครอสโอเวอร์

| **โครโมโซมชุดใหม่** | **ตัวเลขสุ่ม** | **เลือกโครโมโซมที่ตัวเลขสุ่ม < %Pc (0.8)** |
|---|---|---|
| C3 | 0.89 | ☒ |
| C1 | 0.41 | ☑ |
| C4 | 0.22 | ☑ |
| C4 | 0.64 | ☑ |
| C2 | 0.35 | ☑ |

หมายเหตุ สัญลักษณ์ ☒ ไม่ถูกเลือกเพื่อทำการครอสโอเวอร์
สัญลักษณ์ ☑ ถูกเลือกเพื่อทำการครอสโอเวอร์

จากตาราง 3.11 โครโมโซมที่ถูกเลือกเพื่อทำการครอสโอเวอร์ได้แก่ โครโมโซม C1, C4, C4 และ C2 ซึ่งจะนำโครโมโซมเหล่านี้มาจับคู่เพื่อทำการครอสโอเวอร์ตามลำดับ ดังนั้น โครโมโซมคู่แรกได้แก่ โครโมโซม C1 และโครโมโซม C4 คู่ที่สองได้แก่ โครโมโซม C4 และโครโมโซม C2 จากนั้นกำหนดเป็นโครโมโซมพ่อแม่ คือ P1 และ P2 ในแต่ละคู่โครโมโซม โดยโครโมโซมลำดับแรกกำหนดให้เป็น P1 และโครโมโซมลำดับ 2 กำหนดให้เป็น P2 การกำหนดจะทำในลักษณะนี้ของทุกคู่โครโมโซม เพื่อทำการครอสโอเวอร์ ดังตัวอย่างต่อไปนี้

C1 : L0011 E0001 L0021 L0022 E0002 … P0011 P0012 P0013 ┐ P1
C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043 ┘ P2

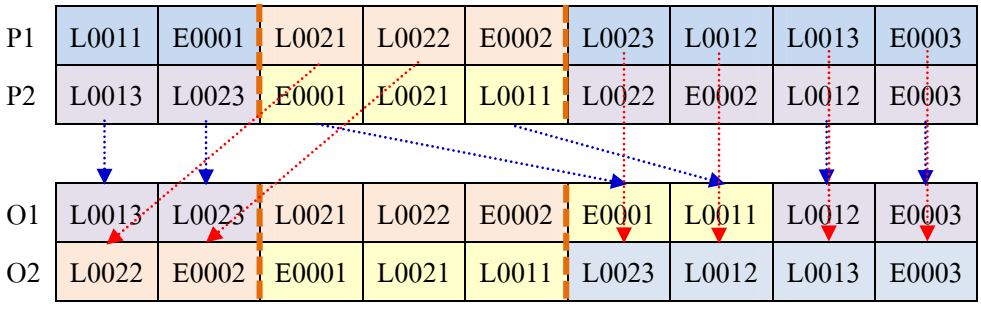
C4 : L0013 L0023 E0001 L0021 L0011 … P0041 P0042 P0043 ┐ P1
C2 : L0031 L0041 L0022 E0001 E0002 … P0011 P0012 P0013 ┘ P2

นำโครโมโซมแต่ละคู่มาทำการครอสโอเวอร์ดังแสดงตัวอย่างในภาพประกอบ 3.21 ซึ่งแสดงวิธีการครอสโอเวอร์แบบ OX โดยสามารถอธิบายขั้นตอนการครอสโอเวอร์ได้ดังนี้

1. การสุ่มตำแหน่งยีนภายในโครโมโซม P1 และ P2 มาสองตำแหน่ง จากภาพประกอบ 3.20 ตำแหน่งยีนที่สุ่มได้ คือ ตำแหน่งที่ 2 และ 5
2. คัดลอกยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม P1 คือ ยีน L0021 L0022 E0000 ไปยังยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม O1

3. ลบค่ายีน L0021 L0022 E0000 ออกจากโครโมโซม P2 ซึ่งอยู่ในตำแหน่งที่ 4, 6 และ 7
4. นำยีนที่เหลือในโครโมโซม P2 คือยีน L0013 L0023 E0001 L0011 L0012 E0003 มาใส่ในตำแหน่งที่ยังว่างของโครโมโซม O1 ตามลำดับจากซ้ายไปขวา จะได้โครโมโซม O1 ซึ่งผ่านกระบวนการครอสโอเวอร์แล้ว
5. คัดลอกยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม P2 คือ ยีน E0001 L0021 L0011 ไปยังยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม O2
6. ลบค่ายีน E0001 L0021 L0011 ออกจากโครโมโซม P1 ซึ่งอยู่ในตำแหน่งที่ 1, 2 และ 3
7. นำยีนที่เหลือในโครโมโซม P1 คือยีน L0022 E0002 L0023 L0012 L0013 E0003 มาใส่ในตำแหน่งที่ยังว่างของโครโมโซม O2 ตามลำดับจากซ้ายไปขวา จะได้โครโมโซม O2 ซึ่งผ่านกระบวนการครอสโอเวอร์แล้วเช่นเดียวกับโครโมโซม O1

โดยที่การครอสโอเวอร์จะแยกทำในส่วนของแต่ละประเภทวิชาคือวิชาบรรยายจะทำการครอสโอเวอร์กันเฉพาะในส่วนของวิชาบรรยาย วิชาปฏิบัติก็จะทำการครอสโอเวอร์เฉพาะในส่วนของวิชาปฏิบัติเช่นเดียวกัน




ภาพประกอบ 3.21 ตัวอย่างวิธีการครอสโอเวอร์แบบ Order Crossover (OX)

เมื่อทำการครอสโอเวอร์จนครบทุกคู่ก็จะได้โครโมโซมตัวใหม่ดังนี้

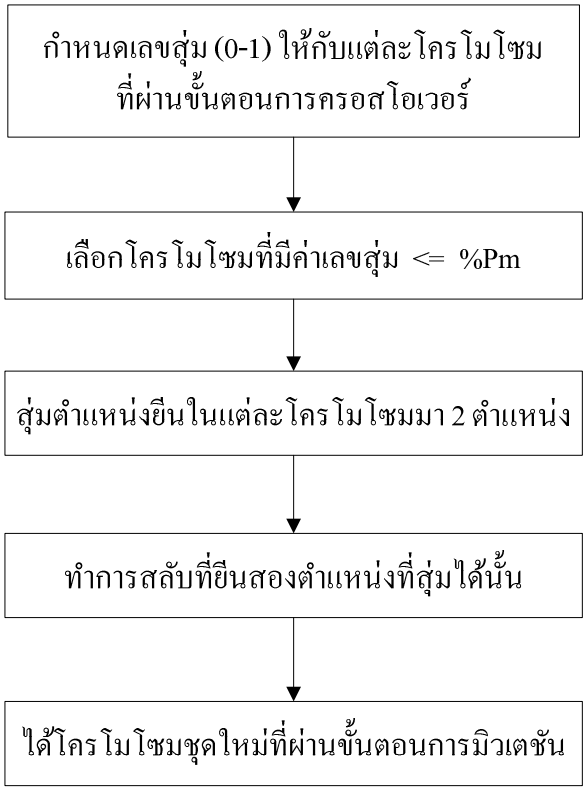
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| O1 : | L0013 | L0023 | L0021 | L0022 | E0002 | E0001 | L0011 | L0012 | E0003 |
| O2 : | L0022 | E0002 | E0001 | L0021 | L0011 | L0023 | L0012 | L0013 | E0003 |
| O3 : | L0021 | L0011 | L0022 | E0001 | E0003 | L0012 | L0023 | E0002 | L0013 |
| O4 : | E0002 | L0021 | L0011 | L0013 | L0023 | E0001 | L0012 | E0003 | L0022 |

มาถึงส่วนนี้จะได้โครโมโซมที่ผ่านการครอสโอเวอร์แล้ว 4 โครโมโซมคือ O1, O2, O3 และ O4 และมีอีก 1 โครโมโซมไม่ได้ถูกคัดเลือกสำหรับทำการครอสโอเวอร์ คือ C3

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C3 : | L0032 | L0033 | E0001 | L0012 | L0011 | … | P0021 | P0022 | P0023 |
| O1 : | L0013 | L0023 | L0021 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| O2 : | L0022 | E0002 | E0001 | L0021 | L0011 | … | P0021 | P0022 | P0023 |
| O3 : | L0021 | L0011 | L0022 | E0001 | E0003 | … | P0041 | P0042 | P0043 |
| O4 : | E0002 | L0021 | L0011 | L0013 | L0023 | … | P0031 | P0032 | P0033 |

### 3.3.4 การมิวเตชัน (mutation)

การมิวเตชันเป็นขั้นตอนที่อาจช่วยให้โครโมโซมมีค่าความเหมาะสมดีขึ้นหลังจากการครอสโอเวอร์ การมิวเตชันทำได้โดยการสลับตำแหน่งของค่าภายในโครโมโซมตัวเดียว จะมีโครโมโซมเพียงบางส่วนเท่านั้นที่จะถูกนำมาทำการมิวเตชันซึ่งขึ้นอยู่กับค่าความน่าจะเป็นในการมิวเตชัน (%Pm) กำหนดให้ค่าความน่าจะเป็นในการมิวเตชันเริ่มต้นเท่ากับ 0.3 ทำการมิวเตชันโดยใช้วิธีการมิวเตชันแบบ Two-point Swapping Mutation เนื่องจากเป็นวิธีการมิวเตชันที่เหมาะสำหรับปัญหาที่มีรูปแบบโครโมโซมเป็นแบบการจัดตารางเวลา ขั้นตอนการคัดเลือกโครโมโซมเพื่อทำการมิวเตชันเป็นดังภาพประกอบ 3.22 ซึ่งเริ่มต้นจากกำหนดเลขสุ่ม (0-1) ให้กับแต่ละโครโมโซมที่ผ่านขั้นตอนการครอสโอเวอร์ ซึ่งได้แก่โครโมโซม O1, O2, O3 และ O4 มีค่าเลขสุ่ม 0.07, 0.34, 0.62 และ 0.91 ตามลำดับดังแสดงตามตาราง 3.9 โดยโครโมโซมที่มีค่าเลขสุ่มน้อยกว่าค่าความน่าจะเป็นในการมิวเตชันที่กำหนดไว้ คือ 0.3 จะถูกเลือกไปทำขั้นตอน มิวเตชันต่อไป ซึ่งจากตัวอย่างพบว่ามีเพียงโครโมโซม O1 เท่านั้นที่มีค่าเลขสุ่ม น้อยกว่า 0.3 คือมีค่า 0.07 ดังนั้นโครโมโซม O1 จึงถูกเลือกมาเพื่อทำการมิวเตชันต่อไป ส่วนโครโมโซมที่ไม่ถูกเลือกเพื่อทำการมิวเตชันได้แก่โครโมโซม O2, O3 และ O4 เนื่องจากมีค่าเลขสุ่ม 0.34, 0.62 และ 0.91 ตามลำดับซึ่งเป็นค่าที่มากกว่า 0.3 จากนั้นทำการมิวเตชันโครโมโซมที่ถูกเลือกมา คือโครโมโซม O1 โดยการสุ่มตำแหน่งยีนย่อยภายในโครโมโซม O1 มา 2 ตำแหน่งแล้วทำการสลับที่ค่าในตำแหน่งยีนที่สุ่มมาได้นั้น จะได้โครโมโซมตัวใหม่ที่ผ่านขั้นตอนการมิวเตชัน ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมเพื่อทำการมิวเตชันแสดงได้ดังตาราง 3.12




ภาพประกอบ 3.22 ขั้นตอนการคัดเลือกโครโมโซมเพื่อทำการมิวเตชัน

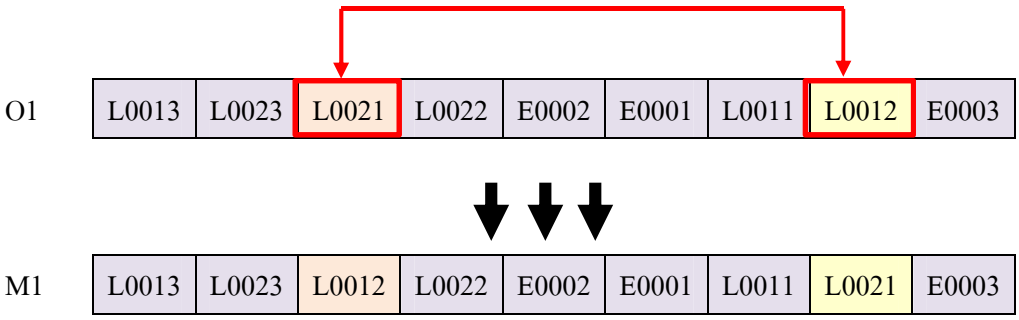
ตาราง 3.12 ตัวอย่างข้อมูลที่ใช้ในการคัดเลือกโครโมโซมเพื่อทำการมิวเตชัน

| **โครโมโซมลูกเบื้องต้น** | **ตัวเลขสุ่ม** | **เลือกโครโมโซมที่ตัวเลขสุ่ม < %Pm (0.3)** |
|---|---|---|
| O1 | 0.07 | ☑ |
| O2 | 0.34 | ☒ |
| O3 | 0.62 | ☒ |
| O4 | 0.91 | ☒ |

หมายเหตุ สัญลักษณ์ ☒ ไม่ถูกเลือกเพื่อทำการมิวเตชัน

สัญลักษณ์ ☑ ถูกเลือกเพื่อทำการมิวเตชัน

จากตาราง 3.12 พบว่าโครโมโซมลูกเบื้องต้น O1 ตัวแรกเท่านั้นที่มีโอกาสทำการมิวเตชันเนื่องจากมีค่าตัวเลขสุ่มต่ำกว่าค่าความน่าจะเป็นในการมิวเตชันที่กำหนด คือมีค่าเลขสุ่มเท่ากับ 0.07 ซึ่งต่ำกว่าค่า 0.3 ดังนั้นโครโมโซม O1 จึงถูกเลือกมาเพื่อทำการมิวเตชันโดยใช้วิธีการมิวเตชันแบบ Two-point Swapping Mutation ดังภาพประกอบ 3.23




ภาพประกอบ 3.23 ตัวอย่างวิธีการมิวเตชันแบบ Two-point Swapping Mutation

จากภาพประกอบ 3.23 แสดงวิธีการมิวเตชันโดยสุ่มตำแหน่งยีนภายในโครโมโซม O1 มา 2 ตำแหน่ง และทำการสลับค่าของยีน 2 ตำแหน่งที่สุ่มมานั้น เช่น ตำแหน่งยีนที่สุ่มมา คือตำแหน่งที่ 3 และ 8 นั่นคือยีน L0021 และ L0012 ทำการสลับที่ยีนสองตำแหน่งนี้ จะได้โครโมโซมตัวใหม่ที่ผ่านการมิวเตชันคือ โครโมโซม M1 ที่มียีนในตำแหน่งที่ 3 เปลี่ยนจากยีน L0021 เป็นยีน L0012 และยีนในตำแหน่งที่ 7 เปลี่ยนจากยีน L0012 เป็นยีน L0021 โครโมโซม O1 ที่ผ่านขั้นตอนการมิวเตชันแล้ว เป็นดังนี้

M1: L0013 L0023 L0012 L0022 E0002 E0001 L0011 L0021 E0003

จากโครโมโซมที่ผ่านการครอสโอเวอร์มาแล้ว 4 โครโมโซม คือโครโมโซม O1, O2, O3 และ O4 พบว่ามี 1 โครโมโซมที่ถูกเลือกเพื่อทำการมิวเตชัน คือโครโมโซม O1 เมื่อผ่านการมิวเตชันแล้วกลายเป็นโครโมโซม M1 ดังนั้นโครโมโซมที่ผ่านการครอสโอเวอร์และการมิวเตชันแล้วเป็นดังนี้

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C3 : | L0032 | L0033 | E0001 | L0012 | L0011 | … | P0021 | P0022 | P0023 |
| M1: | L0013 | L0023 | L0012 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| O2 : | L0022 | E0002 | E0001 | L0021 | L0011 | … | P0021 | P0022 | P0023 |
| O3 : | L0021 | L0011 | L0022 | E0001 | E0003 | … | P0041 | P0042 | P0043 |
| O4 : | E0002 | L0021 | L0011 | L0013 | L0023 | … | P0031 | P0032 | P0033 |

จากชุดโครโมโซมเบื้องต้น 5 โครโมโซม คือโครโมโซม C3, C1, C4, C4 และ C2 พบว่ามี 4 โครโมโซมที่ผ่านการครอสโอเวอร์ คือโครโมโซม O1, O2, O3 และ O4 และจาก

โครโมโซมที่ผ่านการครอสโอเวอร์แล้วนี้มีเพียง 1 โครโมโซมที่ผ่านการมิวเตชัน คือโครโมโซม M1 ดังนั้นโครโมโซมชุดใหม่ที่ได้นี้ คือโครโมโซม C3, M1, O2, O3 และ O4h ซึ่งจะเป็นโครโมโซมรุ่นแรกที่จะเข้าสู่กระบวนการเจเนติกอัลกอริทึมต่อไป

เปรียบเทียบรูปแบบโครโมโซมชุดเดิมที่ยังไม่ผ่านการครอสโอเวอร์ โครโมโซมชุดที่ผ่านการครอสโอเวอร์แล้ว และโครโมโซมชุดที่ผ่านการมิวเตชันแล้ว ได้ดังนี้

โครโมโซมชุดเดิม :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C3 : | L0032 | L0033 | E0001 | L0012 | L0011 | … | P0021 | P0022 | P0023 |
| C1 : | L0011 | E0001 | L0021 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| C4 : | L0013 | L0023 | E0001 | L0021 | L0011 | … | P0041 | P0042 | P0043 |
| C4 : | L0013 | L0023 | E0001 | L0021 | L0011 | … | P0041 | P0042 | P0043 |
| C2 : | L0031 | L0041 | L0022 | E0001 | E0002 | … | P0011 | P0012 | P0013 |

โครโมโซมชุดที่ผ่านการครอสโอเวอร์ :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| O1 : | L0013 | L0023 | L0021 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| O2 : | L0022 | E0002 | E0001 | L0021 | L0011 | … | P0021 | P0022 | P0023 |
| O3 : | L0021 | L0011 | L0022 | E0001 | E0003 | … | P0041 | P0042 | P0043 |
| O4 : | E0002 | L0021 | L0011 | L0013 | L0023 | … | P0031 | P0032 | P0033 |

โครโมโซมชุดที่ผ่านการมิวเตชัน

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| M1: | L0013 | L0023 | L0012 | L0022 | E0002 | … | P0011 | P0012 | P0013 |

ดังนั้นชุดโครโมโซมรุ่นแรกที่ผ่านขั้นตอนการครอสโอเวอร์และมิวเตชันแล้ว เป็นดังนี้

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C3 : | L0032 | L0033 | E0001 | L0012 | L0011 | … | P0021 | P0022 | P0023 |
| M1: | L0013 | L0023 | L0012 | L0022 | E0002 | … | P0011 | P0012 | P0013 |
| O2 : | L0022 | E0002 | E0001 | L0021 | L0011 | … | P0021 | P0022 | P0023 |
| O3 : | L0021 | L0011 | L0022 | E0001 | E0003 | … | P0041 | P0042 | P0043 |
| O4 : | E0002 | L0021 | L0011 | L0013 | L0023 | … | P0031 | P0032 | P0033 |

นำชุดโครโมโซมรุ่นแรกนี้เข้าสู่กระบวนการเจเนติกอัลกอริทึมสำหรับสร้างชุดโครโมโซมรุ่นถัดไปเพื่อค้นหาโครโมโซมที่เหมาะสมที่สุดตามวิธีการของเจเนติกอัลกอริทึม ทำไปเรื่อยๆ จนครบตามจำนวนรุ่นที่กำหนดแล้วจึงหยุดการค้นหา

### 3.3.5 การหยุดการค้นหา

กลไกการทำงานของวิธีการของเจเนติกอัลกอริทึมจะเริ่มต้นด้วยการสุ่มจำนวนประชากรของคำตอบเริ่มต้นขึ้นมา ซึ่งการกำหนดจำนวนประชากรในแต่ละรุ่นจะมีผลในการหาคำตอบที่น่าพึงพอใจที่สุด จากนั้นต้องกำหนดจำนวนรุ่นว่าจะให้กลไกทำงานไปเรื่อยๆ จนได้จำนวนประชากรเท่ากับที่ต้องการหรือจนกว่าจะได้คำตอบที่ดีที่สุดหรือคำตอบที่ยอมรับได้สำหรับปัญหานั้นๆ ความมากน้อยของจำนวนรุ่นมีผลเหมือนกับค่าจำนวนประชากรในแต่ละรุ่น เมื่อโปรแกรมดำเนินการได้ตามจำนวนรุ่นที่ต้องการแล้วก็จะค้นหาคำตอบที่ดีที่สุดจากที่มีการสำรวจมาทั้งหมด และหยุดการค้นหาซึ่งเป็นอันเสร็จสิ้นขั้นตอนเจเนติกอัลกอริทึม

## 3.4 ลักษณะโปรแกรมจัดตารางสอนที่พัฒนาขึ้น

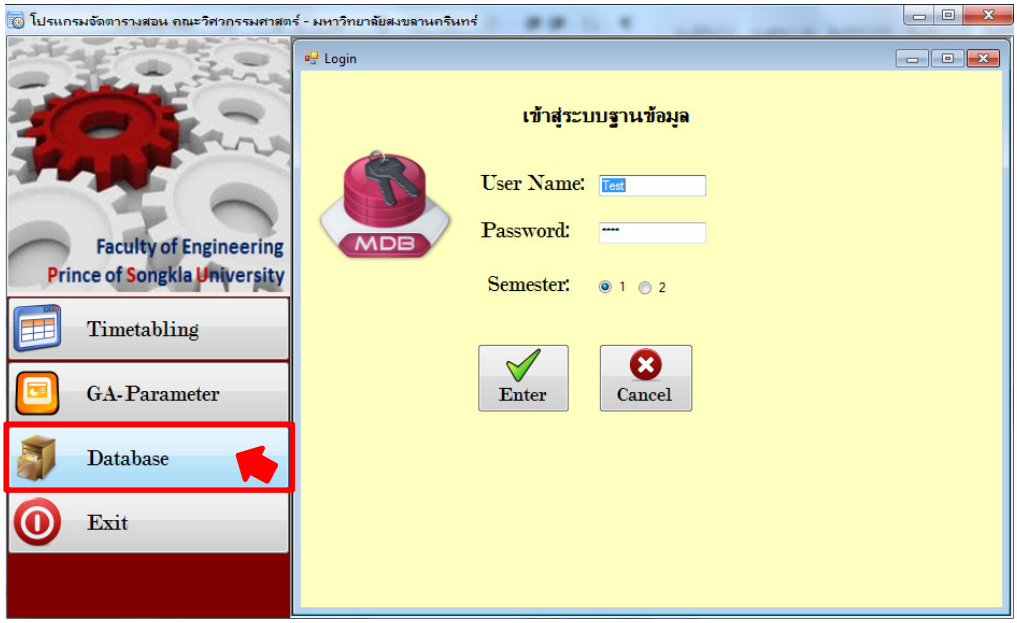
ลักษณะโปรแกรมจัดตารางสอนที่พัฒนาขึ้น โดยหน้าต่างแรกของโปรแกรมประกอบด้วยเมนูหลัก 3 เมนู เพื่อเข้าสู่ส่วนต่าง ๆ ของโปรแกรม ดังภาพประกอบ 3.24

ภาพประกอบ 3.24 ลักษณะโปรแกรมตารางสอนที่พัฒนาขึ้น

สำหรับ 3 เมนูหลักนั้นคือ เมนูจัดตารางสอน เมนูเจเนติกอัลกอริทึม และเมนูจัดการฐานข้อมูล โดยเมนูจัดตารางสอนจะเป็นเมนูหลักที่จะใช้ในการป้อนข้อมูลเพื่อประมวลผลของโปรแกรม โดยจะต้องมีการป้อนข้อมูลรายละเอียดต่างๆ ของข้อมูลที่เกี่ยวข้องกับการจัดตารางสอนลงในฐานข้อมูลก่อนด้วยเมนูจัดการฐานข้อมูล ส่วนเมนูเจเนติกอัลกอริทึมจะเป็นการกำหนดค่าพารามิเตอร์ทางเจเนติกอัลกอริทึมที่เหมาะสมในการประมวลผลโปรแกรม ซึ่งโปรแกรมได้กำหนดค่าเริ่มต้นไว้แล้ว กระบวนการทำงานของเมนูต่างๆ ข้างต้นสามารถอธิบายได้ในหัวข้อถัดไปดังนี้

### 1.4.1 เมนูจัดการฐานข้อมูล

เมื่อผู้ใช้เลือกเมนูจัดการฐานข้อมูล จะปรากฏหน้าต่าง Login ดังภาพประกอบ 3.25 ขึ้นมา เพื่อให้เจ้าหน้าที่จัดตารางสอนเท่านั้นที่สามารถจัดการข้อมูลส่วนนี้ได้ สามารถเพิ่มข้อมูล ลบข้อมูล หรือแก้ไขข้อมูลในฐานข้อมูลได้ เหตุผลที่จำเป็นต้องจำกัดผู้ที่สามารถจัดการฐานข้อมูลของโปรแกรมจัดตารางสอนเนื่องจากข้อมูลที่ใช้ในการจัดตารางสอนมีจำนวนมาก ข้อมูลแต่ละข้อมูลมีรายละเอียดมาก และหากมีการกรอกข้อมูลผิดพลาดจะส่งผลกระทบต่อการประมวลผลของโปรแกรม ทำให้โปรแกรมไม่สามารถประมวลผลได้ หรือให้คำตอบที่ผิดพลาดตามไปด้วย




ภาพประกอบ 3.25 หน้าต่าง Login สำหรับจัดการฐานข้อมูล

เมนูจัดการฐานข้อมูลเป็นส่วนของโปรแกรมที่ทำหน้าที่กำหนดข้อมูลที่จำเป็นทั้งหมดในการจัดตารางสอนเพื่อเก็บไว้ในฐานข้อมูลสำหรับเรียกไปใช้ในการประมวลผลของโปรแกรม ข้อมูลป้อนเข้าสำหรับเมนูจัดการฐานข้อมูลนี้ประกอบด้วยข้อมูลรายวิชา อาจารย์ กลุ่มนักศึกษา และห้องเรียน ดังแสดงตามตาราง 3.13

ตาราง 3.13 ข้อมูลป้อนเข้าสำหรับฐานข้อมูล

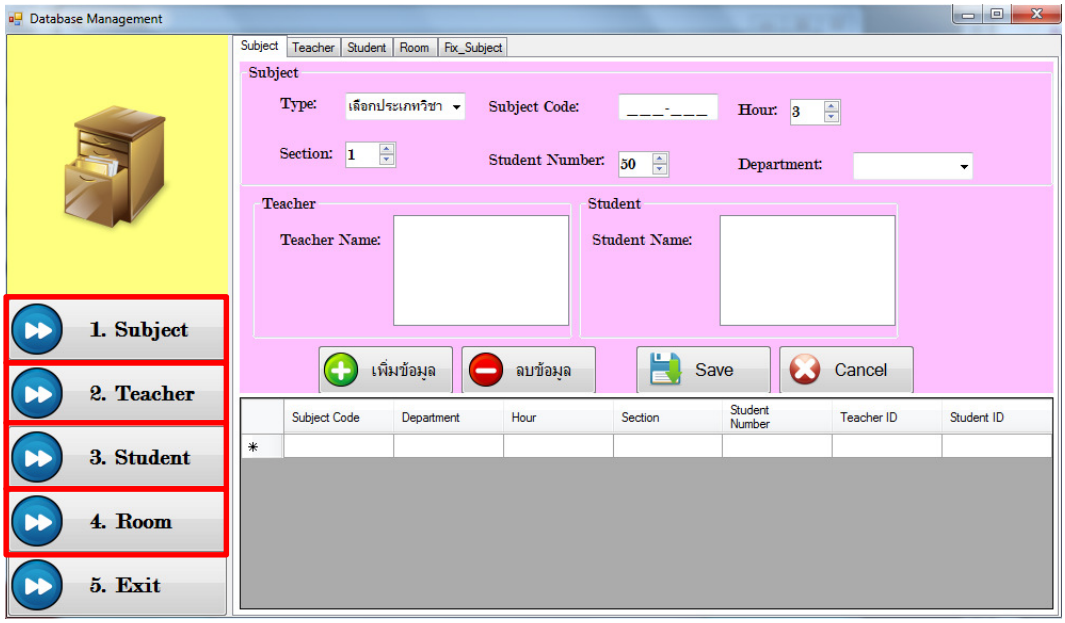
| **ข้อมูลป้อนเข้า** | **รายละเอียดข้อมูลป้อนเข้า** |
|---|---|
| ข้อมูลรายวิชา<br>- วิชาในคณะ | รหัสวิชา ประเภทวิชา จำนวนหน่วยกิต ตอนที่เปิดสอน ภาควิชา รหัสอาจารย์ผู้สอน รหัสกลุ่มนักศึกษาที่เรียน จำนวนนักศึกษาที่ลงทะเบียนเรียน ภาคการศึกษา |
| - วิชานอกคณะ | รหัสวิชา จำนวนหน่วยกิต ตอนที่เปิดสอน รหัสกลุ่มนักศึกษาที่เรียน และวัน-เวลาเรียนที่กำหนดไว้ |
| ข้อมูลอาจารย์ | รหัสอาจารย์ ชื่อ-นามสกุลของอาจารย์ ภาควิชาฯ |
| ข้อมูลกลุ่มนักศึกษา | รหัสกลุ่มนักศึกษา ชื่อกลุ่มนักศึกษา จำนวนนักศึกษา ภาควิชาฯ |
| ข้อมูลห้องเรียน | ชื่อห้องเรียน ประเภทของห้องเรียน ความจุของห้องเรียน ภาควิชาฯ |

***ขั้นตอนการใช้งานเมนูจัดการฐานข้อมูล***

(1) ป้อนข้อมูล User Name และ Password

(2) เลือกภาคการศึกษาที่ต้องการเข้าไปจัดการฐานข้อมูล

(3) กดปุ่ม “Enter” เพื่อเข้าสู่เมนูจัดการฐานข้อมูล ดังภาพประกอบ 3.26




ภาพประกอบ 3.26 เมนูจัดการฐานข้อมูล

จากภาพประกอบ 3.26 แสดงหน้าต่างแรกเมื่อเข้าสู่เมนูจัดการฐานข้อมูล จะมีเมนูย่อย 4 เมนู ซึ่งประกอบด้วยหน้าต่างสำหรับการป้อนข้อมูลต่างๆ ที่จะใช้ในการจัดตารางสอน ได้แก่ ข้อมูลรายวิชา ข้อมูลอาจารย์ ข้อมูลกลุ่มนักศึกษา และข้อมูลห้องเรียน ดังนี้

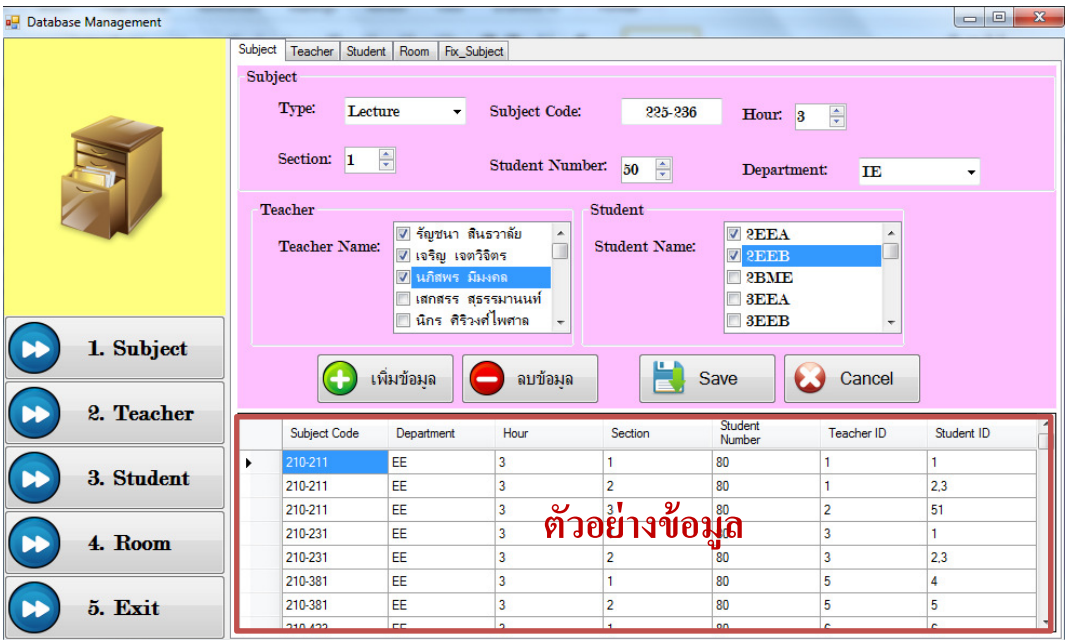
*1 หน้าต่างข้อมูลรายวิชา*

หน้าต่างนี้เป็นหน้าต่างสำหรับให้ผู้ใช้ทำการป้อนข้อมูลเกี่ยวกับรายวิชาที่ใช้ในการจัดตารางสอน ประกอบด้วยรายวิชา 2 ประเภทคือ 1). วิชาในคณะฯ ซึ่งแบ่งเป็นวิชาบรรยาย และวิชาปฏิบัติ 2). วิชานอกคณะฯ ซึ่งรายวิชานี้จะสนใจเฉพาะกลุ่มนักศึกษาที่เรียนและวัน-เวลาที่กองทะเบียนและประมวลผลกำหนดมาแล้ว สำหรับหน้าต่างรับข้อมูลรายวิชาในคณะฯ สามารถแสดงได้ดังภาพประกอบ 3.27 และสามารถอธิบายขั้นตอนการป้อนข้อมูลรายวิชาในคณะได้ดังนี้

(1) เลือกประเภทวิชา เมื่อเลือกประเภทวิชาเรียบร้อย โปรแกรมจะแสดงข้อมูลที่มีอยู่ในฐานข้อมูลโดยจะแสดงข้อมูลตามประเภทวิชา และภาคการศึกษาที่เลือก ยกตัวอย่างเช่น เลือกวิชาบรรยาย และภาคการศึกษาที่ 1 โปรแกรมก็จะแสดงข้อมูลวิชาบรรยายทั้งหมดในภาคการศึกษาที่ 1 ในฐานข้อมูลดังภาพประกอบ 3.27

(2) ป้อนข้อมูลรหัสวิชา โดยข้อมูลที่ป้อนสามารถป้อนได้เฉพาะตัวเลขเท่านั้นโดยรหัสวิชาเป็น _ _ _ - _ _ _ คือตัวเลข 3 หลักแรกบ่งบอกภาควิชาฯ

ต้นสังกัดของรายวิชา และตัวเลข 3 หลักสุดท้ายบ่งบอกรายวิชา ดังภาพประกอบ 3.27




ภาพประกอบ 3.27 หน้าต่างสำหรับป้อนข้อมูลรายวิชาในคณะ

(3) ป้อนข้อมูลชั่วโมงการเรียนการสอน โดยกำหนดค่าเริ่มต้นไว้ที่ 3 เนื่องจากรายวิชาส่วนใหญ่มีการเรียนการสอน 3 คาบต่อสัปดาห์

(4) ป้อนข้อมูลตอนที่เปิดสอน

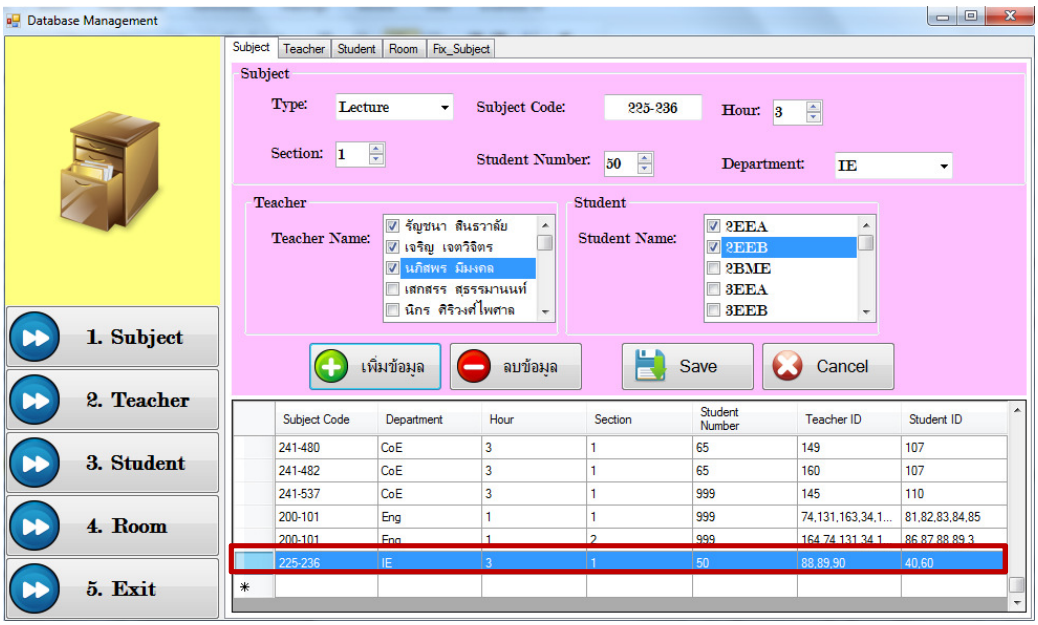
(5) ป้อนข้อมูลจำนวนนักศึกษาที่เปิดรับให้นักศึกษาสามารถลงทะเบียนเรียนรายวิชานั้น

(6) ป้อนภาควิชาต้นสังกัดของรายวิชานั้น

(7) เลือกอาจารย์ผู้สอนรายวิชานั้น โดยอาจารย์ที่สอนรายวิชานั้นจะเป็นอาจารย์ประจำภาควิชาเดียวกับรายวิชา และสามารถเลือกอาจารย์ได้มากกว่า 1 คน เนื่องจากบางรายวิชามีอาจารย์ผู้สอนมากกว่า 1 คน

(8) เลือกกลุ่มนักศึกษาที่มีเรียนรายวิชานั้นตามหลักสูตร โดยสามารถเลือกกลุ่มนักศึกษาได้มากกว่า 1 กลุ่ม เนื่องจากบางรายวิชามีกลุ่มนักศึกษาเรียนร่วมกันมากกว่า 1 กลุ่ม

(9) กดปุ่ม “เพิ่มข้อมูล” เพื่อเพิ่มข้อมูลรายวิชาในคณะ ดังภาพประกอบ 3.28




ภาพประกอบ 3.28 หน้าต่างแสดงการเพิ่มข้อมูลรายวิชาในคณะ

(10) กดปุ่ม “ลบข้อมูล” หากต้องการลบข้อมูลรายวิชาในคณะ

(11) กดปุ่ม “Save” เพื่อบันทึกข้อมูลลงฐานข้อมูล

(12) กดปุ่ม “Cancel” หากไม่ต้องการบันทึกข้อมูลลงฐานข้อมูล

สำหรับหน้าต่างของวิชานอกคณะฯ นั้นมีข้อมูลที่ต้องป้อนต่างไปจากข้อมูลของรายวิชาในคณะ รายวิชานี้จะจัดแตกต่างไปจากรายวิชาในคณะด้วย โดยรายวิชานี้จะพิจารณาเฉพาะกลุ่มนักศึกษาและเวลาเรียนของรายวิชานี้เท่านั้น ซึ่งจะนำข้อมูลที่กองทะเบียนและประมวลผลได้จัดไว้แล้วมาใช้ในการพิจาณา ซึ่งส่วนใหญ่จะเป็นรายวิชาของนักศึกษาชั้นปีที่ 1 ดังภาพประกอบ 3.29 ซึ่งมีขั้นตอนการป้อนข้อมูลรายวิชานอกคณะดังนี้

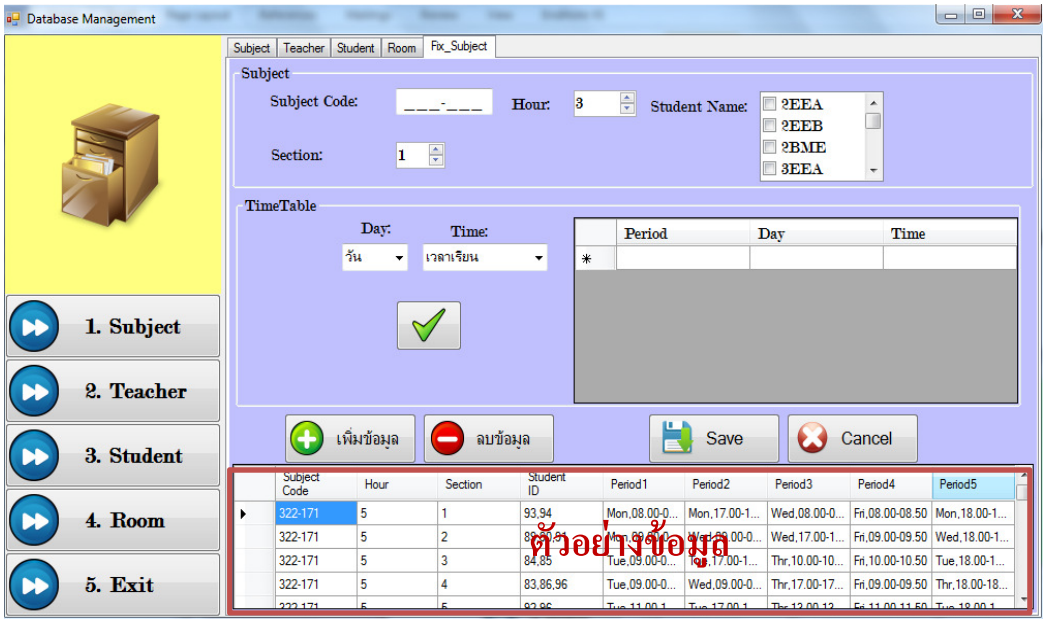
(1) ป้อนข้อมูลรหัสวิชา

(2) ป้อนข้อมูลชั่วโมงการเรียนการสอน

(3) ป้อนข้อมูลตอนที่เปิดสอน

(4) ป้อนข้อมูลกลุ่มนักศึกษา

(5) เลือกวัน-เวลาที่กองทะเบียนและประมวลผลกำหนดไว้ โดยจะต้องป้อนให้ครบตามจำนวนข้อมูลชั่วโมงการเรียนการสอน



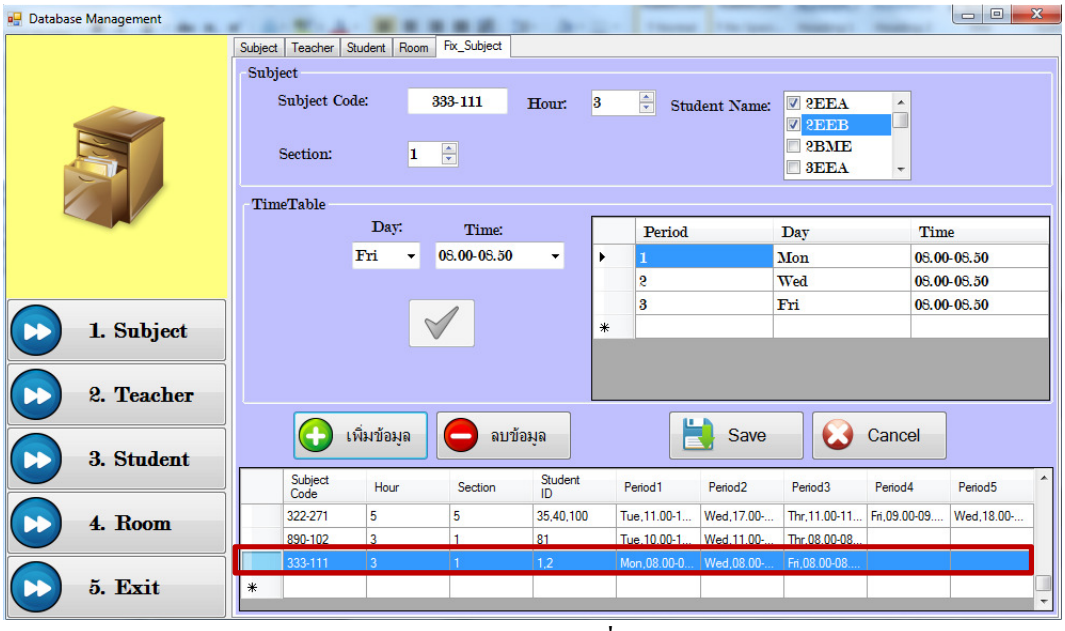

ภาพประกอบ 3.29 หน้าต่างสำหรับป้อนข้อมูลรายวิชานอกคณะ

(6) กดปุ่ม “เพิ่มข้อมูล” เพื่อเพิ่มข้อมูลรายวิชานอกคณะ ดังภาพประกอบ 3.30

(7) กดปุ่ม “ลบข้อมูล” หากต้องการลบข้อมูลรายวิชานอกคณะ

(8) กดปุ่ม “Save” เพื่อบันทึกข้อมูลลงฐานข้อมูล

(9) กดปุ่ม “Cancel” หากไม่ต้องการบันทึกข้อมูลลงฐานข้อมูล




ภาพประกอบ 3.30 หน้าต่างแสดงการเพิ่มข้อมูลรายวิชานอกคณะ

2 *หน้าต่างข้อมูลอาจารย์*

หน้าต่างนี้เป็นหน้าต่างสำหรับป้อนข้อมูลของอาจารย์ที่มีการสอนในรายวิชาที่เปิดสอนของภาคการศึกษาที่ต้องการทำการจัดตารางสอน ดังภาพประกอบ 3.31 ขั้นตอนการป้อนข้อมูลอาจารย์มีดังนี้

(1) ป้อนข้อมูลชื่อ-นามสกุล ของอาจารย์

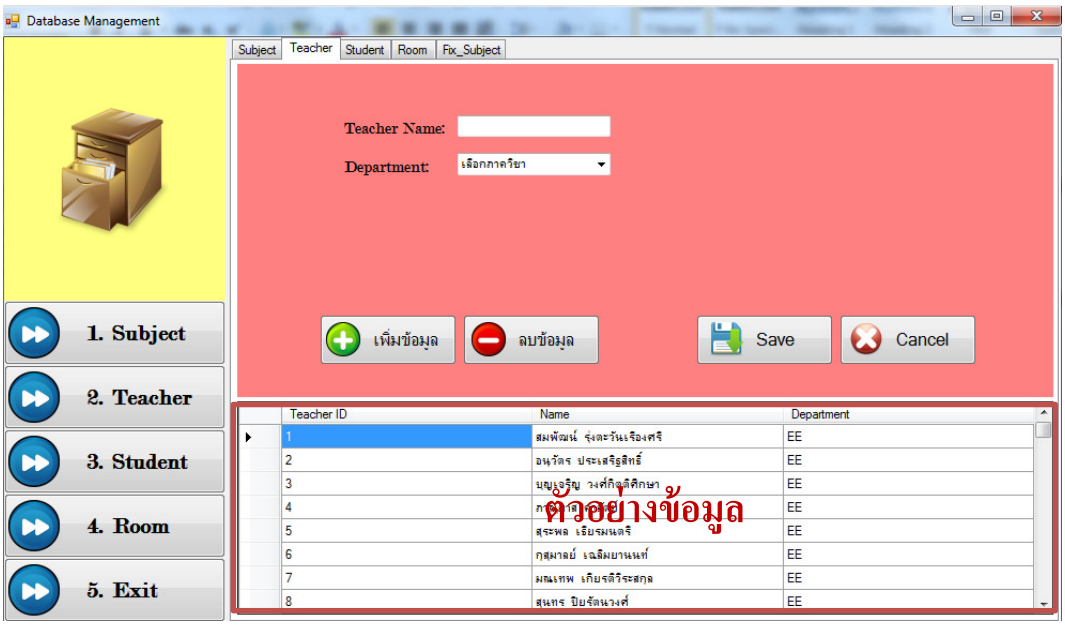
(2) ป้อนข้อมูลภาควิชาต้นสังกัดของอาจารย์

(3) กดปุ่ม “เพิ่มข้อมูล” เพื่อเพิ่มข้อมูลของอาจารย์ ดังภาพประกอบ 3.32

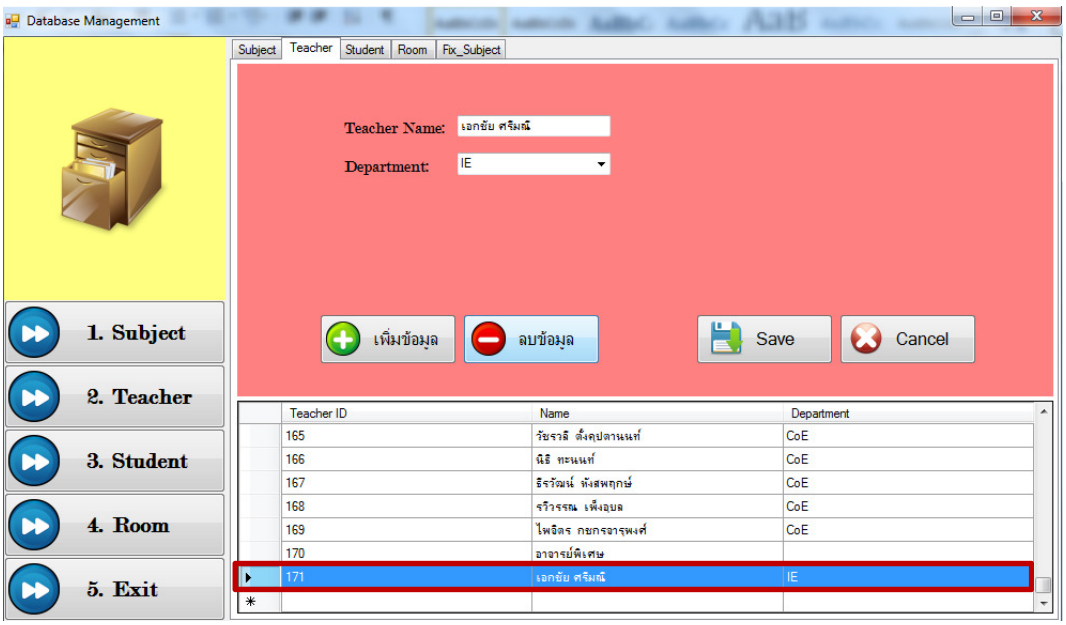
(4) กดปุ่ม “ลบข้อมูล” หากต้องการลบข้อมูลของอาจารย์

(5) กดปุ่ม “Save” เพื่อบันทึกข้อมูลลงฐานข้อมูล

(6) กดปุ่ม “Cancel” หากไม่ต้องการบันทึกข้อมูลลงฐานข้อมูล




ภาพประกอบ 3.31 หน้าต่างสำหรับป้อนข้อมูลอาจารย์



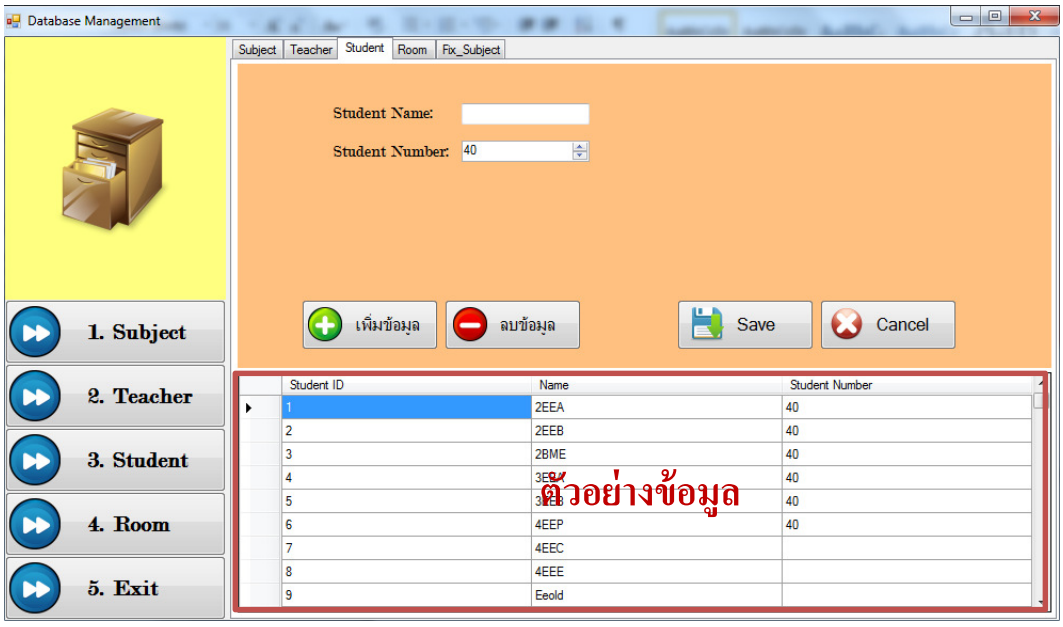

ภาพประกอบ 3.32 หน้าต่างแสดงการเพิ่มข้อมูลอาจารย์

*3 หน้าต่างข้อมูลกลุ่มนักศึกษา*

หน้าต่างนี้เป็นหน้าต่างสำหรับป้อนข้อมูลของกลุ่มนักศึกษา ซึ่งเป็นนักศึกษาแต่ละภาควิชา แต่ละชั้นปี ดังภาพประกอบ 3.33 ขั้นตอนการป้อนข้อมูลกลุ่มนักศึกษามีดังนี้

(1) ป้อนข้อมูลชื่อกลุ่มนักศึกษา

(2) ป้อนจำนวนนักศึกษาของกลุ่มนักศึกษา



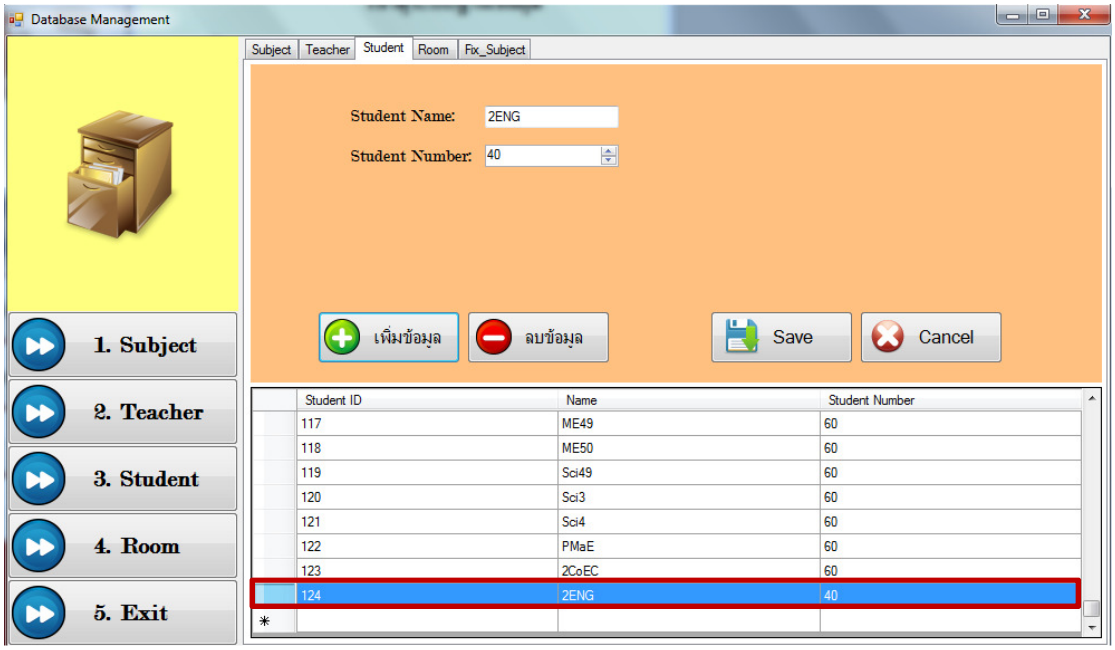

ภาพประกอบ 3.33 หน้าต่างสำหรับป้อนข้อมูลกลุ่มนักศึกษา

(3) กดปุ่ม “เพิ่มข้อมูล” เพื่อเพิ่มข้อมูลกลุ่มนักศึกษา ดังภาพประกอบ 3.34

(4) กดปุ่ม “ลบข้อมูล” หากต้องการลบข้อมูลกลุ่มนักศึกษา

(5) กดปุ่ม “Save” เพื่อบันทึกข้อมูลลงฐานข้อมูล

(6) กดปุ่ม “Cancel” หากไม่ต้องการบันทึกข้อมูลลงฐานข้อมูล




ภาพประกอบ 3.34 หน้าต่างแสดงการเพิ่มข้อมูลกลุ่มนักศึกษา

*4 หน้าต่างข้อมูลห้องเรียน*

หน้าต่างนี้เป็นหน้าต่างสำหรับป้อนข้อมูลของห้องเรียนที่ใช้ในการจัดตารางสอน ดังภาพประกอบ 3.35 ขั้นตอนการป้อนข้อมูลห้องเรียนมีดังนี้

(1) ป้อนข้อมูลชื่อห้องเรียน

(2) ป้อนข้อมูลความจุของห้องเรียน

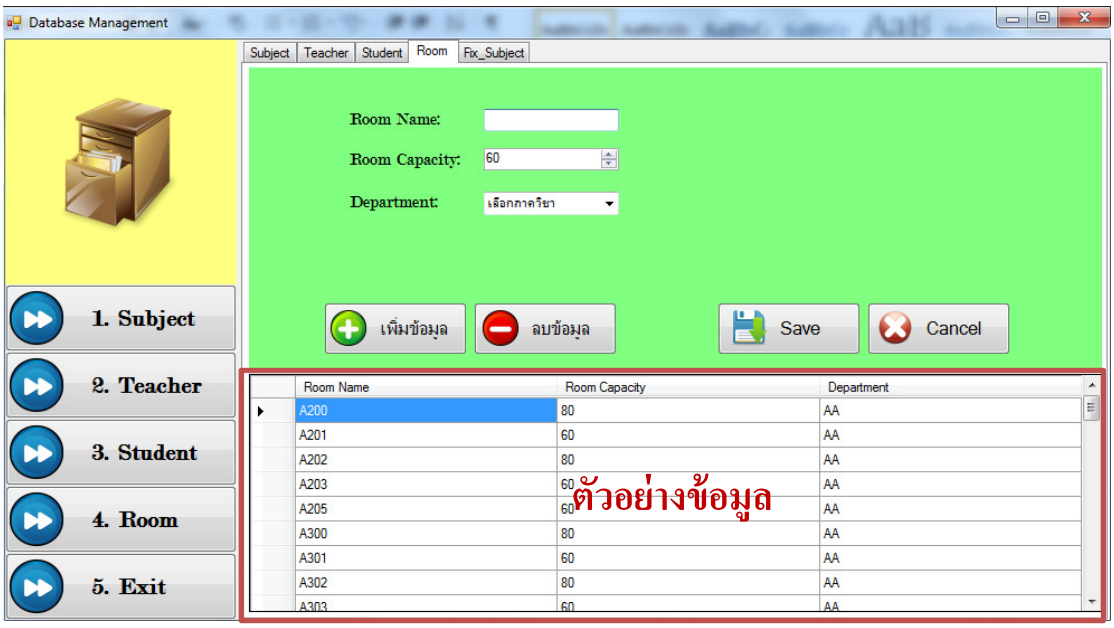
(3) ป้อนข้อมูลภาควิชาฯ ของห้องเรียน

(4) กดปุ่ม “เพิ่มข้อมูล” เพื่อเพิ่มข้อมูลห้องเรียน ดังภาพประกอบ 3.36

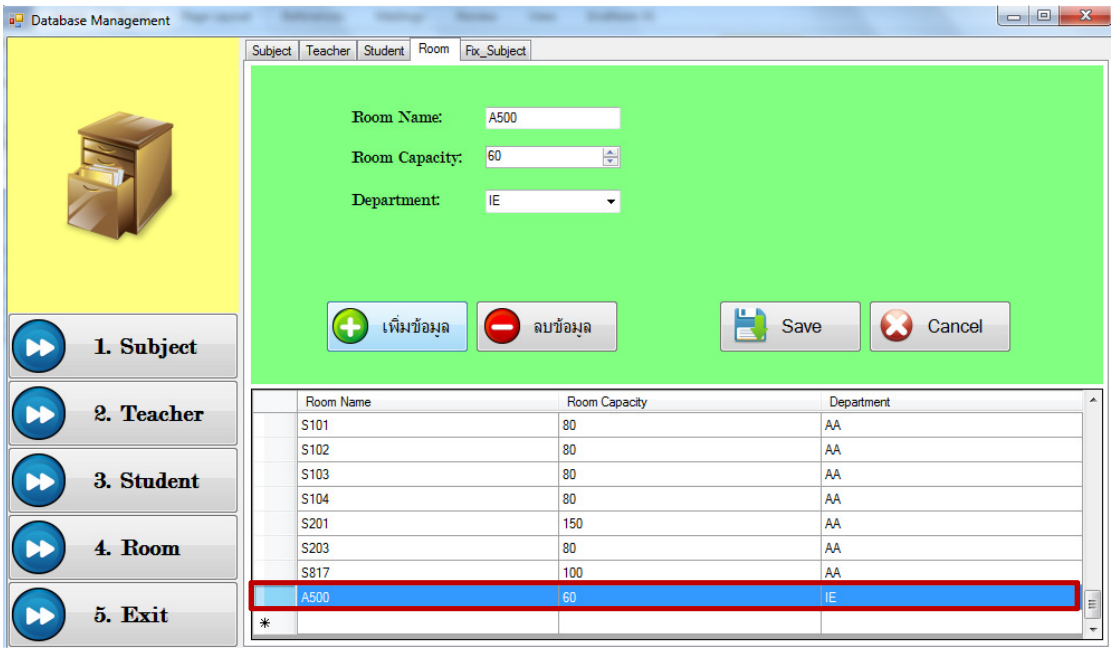
(5) กดปุ่ม “ลบข้อมูล” หากต้องการลบข้อมูลห้องเรียน

(6) กดปุ่ม “Save” เพื่อบันทึกข้อมูลลงฐานข้อมูล

(7) กดปุ่ม “Cancel” หากไม่ต้องการบันทึกข้อมูลลงฐานข้อมูล




ภาพประกอบ 3.35 หน้าต่างสำหรับป้อนข้อมูลห้องเรียน




ภาพประกอบ 3.36 หน้าต่างแสดงการเพิ่มข้อมูลห้องเรียน

จากกระบวนการป้อนข้อมูลเข้าสู่ฐานข้อมูลโดยนำข้อมูลจากกองทะเบียนและประมวลผลมาป้อนเข้าสู่ฐานข้อมูลโดยตรงของผู้ทำวิจัย พบว่าเกิดข้อผิดพลาดในการประมวลผลของโปรแกรม โปรแกรมไม่สามารถหาคำตอบที่เป็นไปได้ของปัญหาให้ได้ นั้นคือตารางสอนที่ได้จากโปรแกรมยังคงมีเวลาสอนของอาจารย์ที่ซ้ำซ้อนกันและมีเวลาเรียนของกลุ่มนักศึกษาที่ซ้ำซ้อน

กัน จึงได้ทำการวิเคราะห์หาสาเหตุที่ทำให้โปรแกรมไม่สามารถหาคำตอบที่เป็นไปได้ โดยพิจารณาไปที่อาจารย์และกลุ่มนักศึกษาที่เกิดการซ้ำซ้อนกันมากๆ โดยพบว่าสาเหตุหลักที่ทำให้เกิดการซ้ำซ้อนกันของเวลาอาจารย์และกลุ่มนักศึกษานั้นคือ มีอาจารย์บางคนมีภาระงานสอนมากกว่า 30 คาบต่อสัปดาห์ มีกลุ่มนักศึกษาบางกลุ่มมีเรียนมาก 30 คาบต่าสัปดาห์ จากสาเหตุดังกล่าวเหล่านี้เมื่อตรวจสอบถึงข้อมูลที่ได้รับจากกองทะเบียนและประมวลผลพบว่าข้อมูลที่กองทะเบียนและประมวลผลใช้ในการจัดตารางสอนนั้นมีความผิดพลาดอยู่ คือมีการกำหนดกลุ่มนักศึกษาและอาจารย์ผู้สอนในบางรายวิชาซ้ำซ้อนกัน เช่น รายวิชาปฏิบัติมีบางวิชาได้มีการแบ่งนักศึกษาเป็นกลุ่มย่อยในการลงปฏิบัติการ หรือบางรายวิชามีผู้เรียนมากกว่า 1 กลุ่ม บางรายวิชามีผู้สอนมากกว่า 1 คน ซึ่งเมื่อตรวจสอบข้อมูลนั้นมีความซับซ้อนกันอยู่ เช่น มีกลุ่มนักศึกษา 2EE, 2EEA และ 2EEB ซึ่งเมื่อตรวจสอบแล้วพบว่า 2EEA และ 2EEB เป็นกลุ่มย่อยของ 2EE จึงทำให้เกิดการซ้ำซ้อนของกลุ่มนักศึกษาขึ้น เป็นต้น ดังนั้นผู้ทำวิจัยจึงได้ดำเนินการแก้ไขข้อมูลใหม่ โดยเริ่มจากการพิจารณาหลักสูตรการเรียนของกลุ่มนักศึกษาทุกกลุ่ม เพื่อกำหนดรายวิชาที่จะเปิดสอนในภาคการศึกษาที่ต้องการจัดตารางสอน เมื่อทราบรายวิชาทั้งหมดที่จะต้องจัดตารางสอนแล้วก็กำหนดอาจารย์และกลุ่มนักศึกษาสำหรับรายวิชาแต่ละรายวิชา โดยการกำหนดอาจารย์ที่สอนแต่ละรายวิชาจะพิจารณาตามรายวิชาที่อาจารย์แต่ละคนรับผิดชอบ และกำหนดกลุ่มนักศึกษาแต่ละรายวิชาโดยพิจารณาตามนักศึกษาที่ลงทะเบียนตามแผนการเรียนเป็นหลักหลังจากนั้นจึงทำการป้อนข้อมูลเข้าสู่ฐานข้อมูลใหม่อีกครั้ง ซึ่งสามารถสรุปแนวทางในการป้อนข้อมูลเข้าสู่ฐานข้อมูลได้ดังนี้

(1) สำหรับกลุ่มนักศึกษาที่มีเรียนรายวิชาปฏิบัติที่มีการแบ่งกลุ่มนักศึกษาออกเป็นกลุ่มย่อยเพื่อลงปฏิบัติการในคนละวันนั้นให้สร้างเป็นกลุ่มนักศึกษาย่อยแทนกลุ่มนักศึกษาทั้งหมด เช่น กลุ่มนักศึกษา 2EE ได้มีเรียนรายวิชาปฏิบัติที่แบ่งเป็นกลุ่มย่อย 2 กลุ่มเพื่อลงปฏิบัติการใน 2 วัน ก็จะทำการแบ่งกลุ่มนักศึกษาออกเป็น 2EEA และ 2EEB เพื่อป้อนข้อมูลลงสู่รายวิชาสำหรับนักศึกษาแต่ละกลุ่ม และเมื่อเป็นวิชาที่เรียนด้วยกันเป็น 2EE ทั้งหมดก็จะป้อนข้อมูลว่ารายวิชามีกลุ่มผู้เรียน 2 กลุ่มคือ 2EEA และ 2EEB ซึ่งเมื่อ 2 กลุ่มนักศึกษานี้ร่วมกันก็จะเท่ากับกลุ่มนักศึกษา 2EE เท่าเดิม

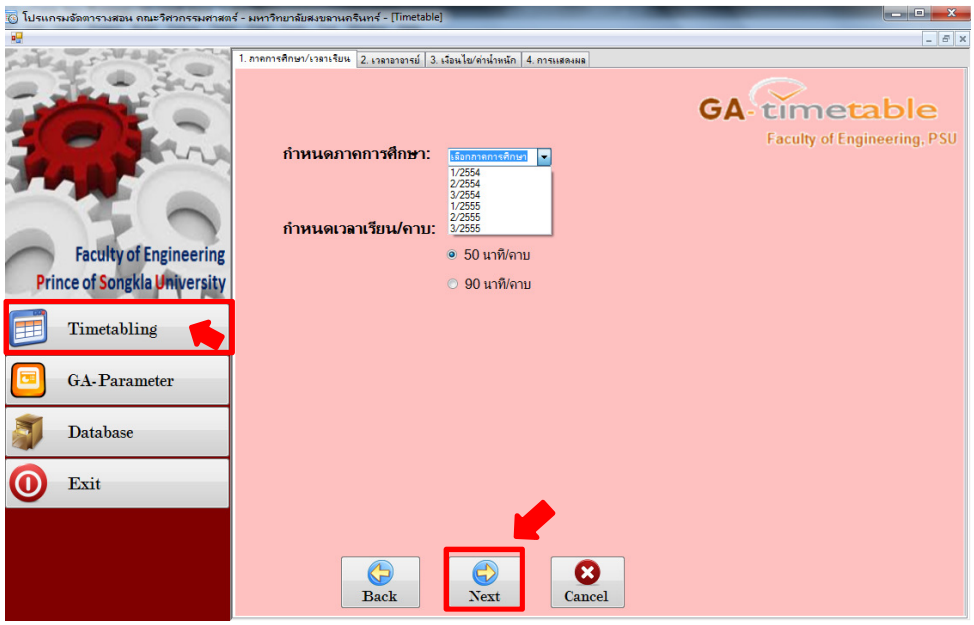
(2) สำหรับรายวิชาที่มีอาจารย์สอนร่วมกันหลายคนนั้นก็ทำการตรวจสอบก่อนว่ารายวิชานี้ใช่รายวิชาที่มีอาจารย์สอนร่วมกันหลายคนหรือไม่ เนื่องจากบางรายวิชาได้เปิดสอนสำหรับหลายกลุ่มนักศึกษาซึ่งมีการเปิดสอนหลายตอนและอาจารย์แต่ละคนก็รับผิดชอบสอนคนละตอนสำหรับนักศึกษาแต่ละกลุ่ม แต่กองทะเบียนและประมวลผลได้รับข้อมูลที่ผิดพลาดจึงทำให้เข้าใจว่า

อาจารย์ทุกคนสอนทุกกลุ่มนักศึกษา จึงทำให้ภาระการสอนของอาจารย์เพิ่มขึ้น ทำให้ไม่สามารถจัดตารางสอนสำหรับอาจารย์ได้เนื่องจากไม่มีเวลาว่างพอให้จัดตารางสอนได้ แต่หากเป็นวิชาที่สอนร่วมกันหลายกลุ่มจริงก็ป้อนข้อมูลอาจารย์ให้ครบทุกคนตามความเป็นจริง

### 3.4.2 เมนูจัดตารางสอน

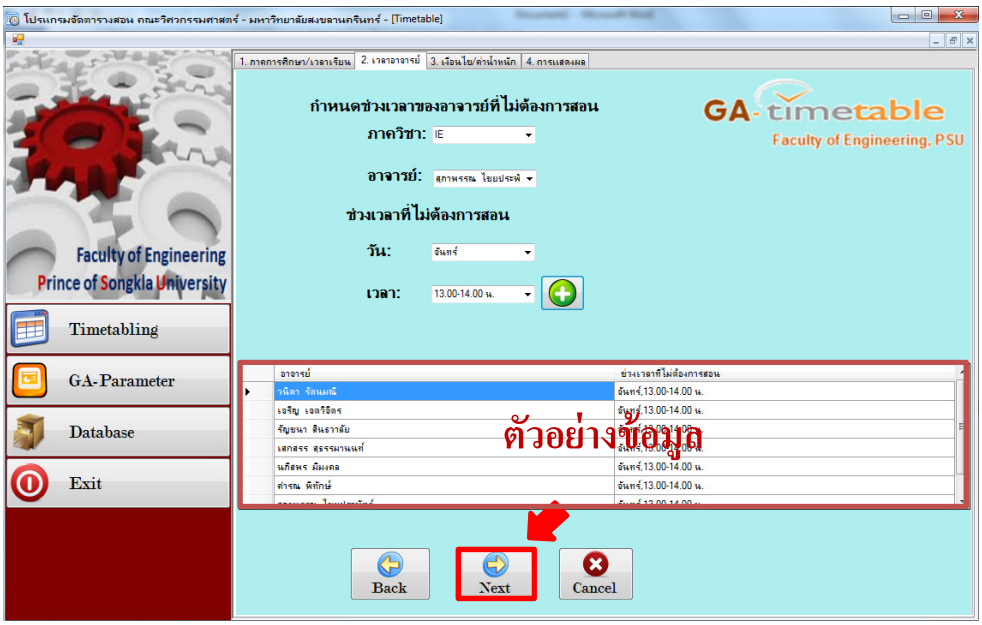
เมื่อผู้ใช้เลือกเมนูนี้จะปรากฏหน้าต่างสำหรับการจัดตารางสอนดังภาพประกอบ 3.37 ซึ่งเป็นหน้าต่างสำหรับรับข้อมูลที่จำเป็นสำหรับการจัดตารางสอนที่จะถูกกำหนดโดยผู้ใช้หรือผู้จัดตารางสอน ข้อมูลเหล่านี้เป็นข้อมูลที่สามารถเปลี่ยนแปลงได้ทุกครั้งที่จะทำการจัดตารางสอน โดยโปรแกรมจะรับข้อมูลส่วนนี้เพื่อไปดึงข้อมูลจากฐานข้อมูลและนำค่าที่ได้รับมาทำการประมวลผลในการหาคำตอบ สำหรับขั้นตอนการใช้งานเมนูจัดตารางสอนมีดังนี้

(1) กำหนดภาคการศึกษาและเวลาเรียนต่อคาบที่ต้องการทำการจัดตารางสอน และกดปุ่ม “Next” ดังภาพประกอบ 3.37




ภาพประกอบ 3.37 หน้าต่างการกำหนดภาคการศึกษาและเวลาเรียนต่อคาบ

(2) กำหนดช่วงเวลาของอาจารย์ที่ไม่ต้องการสอน โดยเลือกภาควิชาต้นสังกัดของอาจารย์ วันและเวลาที่อาจารย์ไม่ต้องการสอน และกดปุ่ม “Next” ดังภาพประกอบ 3.38

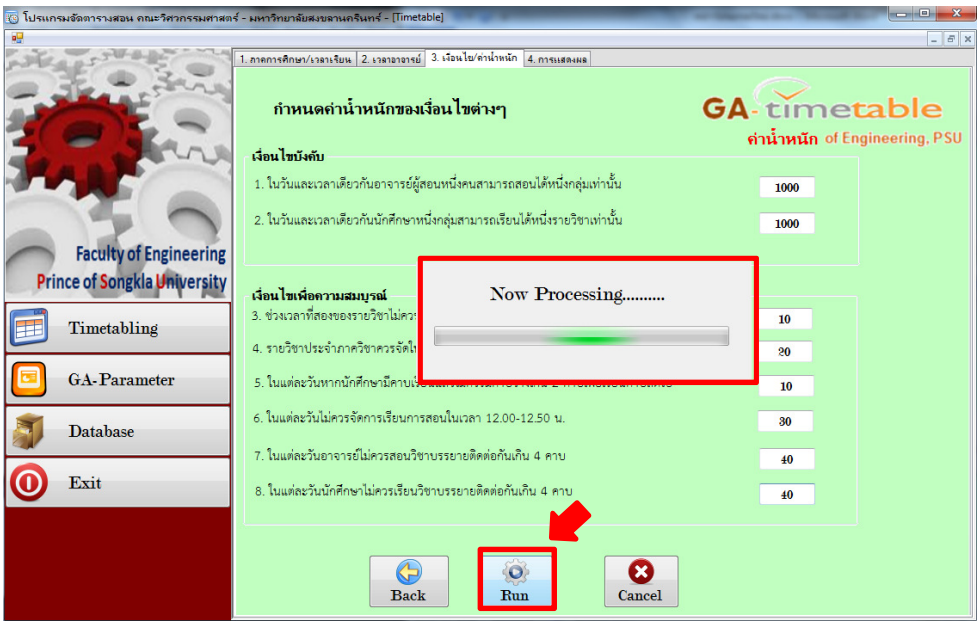



ภาพประกอบ 3.38 หน้าต่างการกำหนดช่วงเวลาของอาจารย์ที่ไม่ต้องการสอน

(3) กำหนดค่าน้ำหนักของเงื่อนไขต่างๆ ในการจัดตารางสอน ดังภาพประกอบ 3.39

ภาพประกอบ 3.39 หน้าต่างการกำหนดค่าน้ำหนักของเงื่อนไขต่างๆ

(4) กดปุ่ม “Run” เพื่อประมวลผลโปรแกรมดังภาพประกอบ 3.40




ภาพประกอบ 3.40 หน้าต่างที่ปรากฏเมื่อกดปุ่ม “Run”

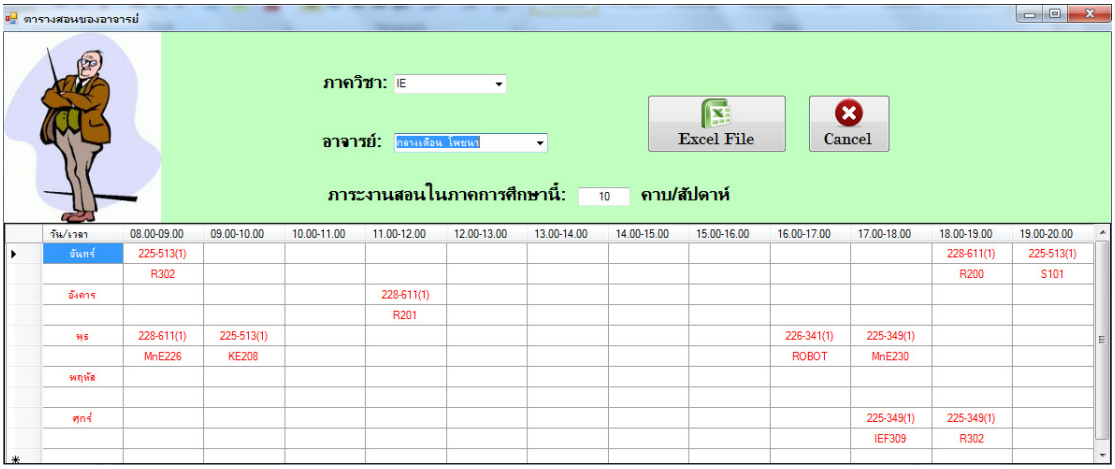
เมื่อโปรแกรมประมวลผลเสร็จแล้ว จะปรากฏหน้าต่างการกำหนดการแสดงผล ขึ้นมาดังภาพประกอบ 3.41 สามารถเรียกดูผลลัพธ์ตารางสอนทั้งหมดที่ได้

ภาพประกอบ 3.41 หน้าต่างการกำหนดการแสดงผล

(5) กดปุ่ม “ทั้งหมด” เพื่อสร้างตารางสอนที่ได้จากการประมวลผลของโปรแกรมในรูปแบบไฟล์ Excel ดังภาพประกอบ 3.42

ภาพประกอบ 3.42 หน้าต่างที่ปรากฏเมื่อกดปุ่ม “ทั้งหมด”

(6) กดปุ่ม “Teacher” แล้วเลือกภาควิชา และอาจารย์ เพื่อดูตารางสอนของอาจารย์ดังกล่าวที่ได้จากการประมวลผลโปรแกรม ดังภาพประกอบ 3.43 และสามารถเรียกดูไฟล์ Excel ของตารางสอนอาจารย์ได้ โดยกดปุ่ม “Excel File” จะแสดงตารางสอนของอาจารย์ ดังภาพประกอบ 3.44

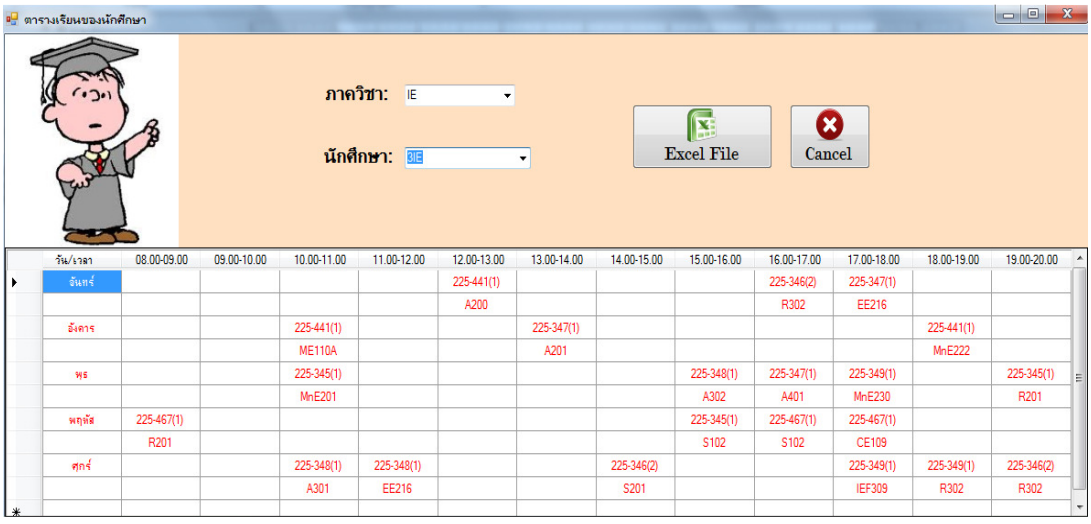



ภาพประกอบ 3.43 หน้าต่างแสดงตัวอย่างตารางสอนอาจารย์

| กลางเดือน โพธนา | 08.00-09.00 | 09.00-10.00 | 10.00-11.00 | 11.00-12.00 | 12.00-13.00 | 13.00-14.00 | 14.00-15.00 | 15.00-16.00 | 16.00-17.00 | 17.00-18.00 | 18.00-19.00 | 19.00-20.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mon. | 225-513(1) | | | | | | | | | | 228-611(1) | 225-513(1) |
| | R302 | | | | | | | | | | R200 | S101 |
| Tue. | | | | 228-611(1) | | | | | | | | |
| | | | | R201 | | | | | | | | |
| Wed. | 228-611(1) | 225-513(1) | | | | | | | 226-341(1) | 225-349(1) | | |
| | MnE226 | KE208 | | | | | | | ROBOT | MnE230 | | |
| Thu. | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| Fri. | | | | | | | | | | 225-349(1) | 225-349(1) | |
| | | | | | | | | | | IEF309 | R302 | |

ภาพประกอบ 3.44 ตัวอย่างตารางสอนของอาจารย์

(7) กดปุ่ม “Student” แล้วเลือกภาควิชา และกลุ่มนักศึกษา เพื่อดูตารางเรียนของกลุ่มนักศึกษาดังกล่าวที่ได้จากการประมวลผลโปรแกรม ดังภาพประกอบ 3.45 และสามารถเรียกดูไฟล์ Excel ของตารางสอนตามกลุ่มนักศึกษาได้ โดยกดปุ่ม “Excel File” จะแสดงตารางเรียนของกลุ่มนักศึกษา ดังภาพประกอบ 3.46

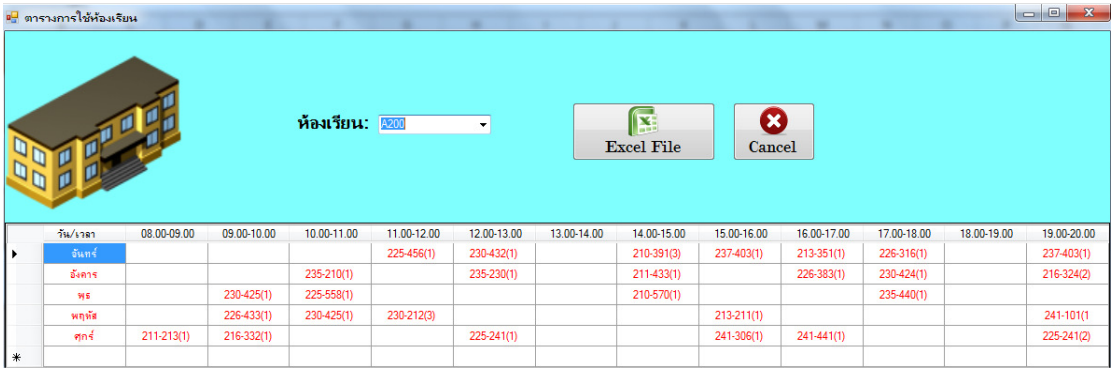



ภาพประกอบ 3.45 หน้าต่างแสดงตัวอย่างตารางเรียนของกลุ่มนักศึกษา

| 3IE | 08.00-09.00 | 09.00-10.00 | 10.00-11.00 | 11.00-12.00 | 12.00-13.00 | 13.00-14.00 | 14.00-15.00 | 15.00-16.00 | 16.00-17.00 | 17.00-18.00 | 18.00-19.00 | 19.00-20.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mon. | | | | | 225-441(1) | | | | 225-346(2) | 225-347(1) | | |
| | | | | | A200 | | | | R302 | EE216 | | |
| Tue. | | | 225-441(1) | | | 225-347(1) | | | | | 225-441(1) | |
| | | | ME110A | | | A201 | | | | | MnE222 | |
| Wed. | | | 225-345(1) | | | | | 225-348(1) | 225-347(1) | 225-349(1) | | 225-345(1) |
| | | | MnE201 | | | | | A302 | A401 | MnE230 | | R201 |
| Thu. | 225-467(1) | | | | | | | 225-345(1) | 225-467(1) | 225-467(1) | | |
| | R201 | | | | | | | S102 | S102 | CE109 | | |
| Fri. | | | 225-348(1) | 225-348(1) | | | 225-346(2) | | | 225-349(1) | 225-349(1) | 225-346(2) |
| | | | A301 | EE216 | | | S201 | | | IEF309 | R302 | R302 |

ภาพประกอบ 3.46 ตัวอย่างตารางเรียนของกลุ่มนักศึกษา

(8) กดปุ่ม “Room” แล้วเลือกห้องเรียน เพื่อดูตัวอย่างตารางการใช้ห้องเรียน ดังกล่าวที่ได้จากการประมวลผลโปรแกรม ดังภาพประกอบ 3.47 และสามารถเรียกดูไฟล์ Excel ของตารางการใช้ห้องเรียนได้ โดยกดปุ่ม “Excel File” จะแสดงตารางการใช้เรียน ดังภาพประกอบ 3.48



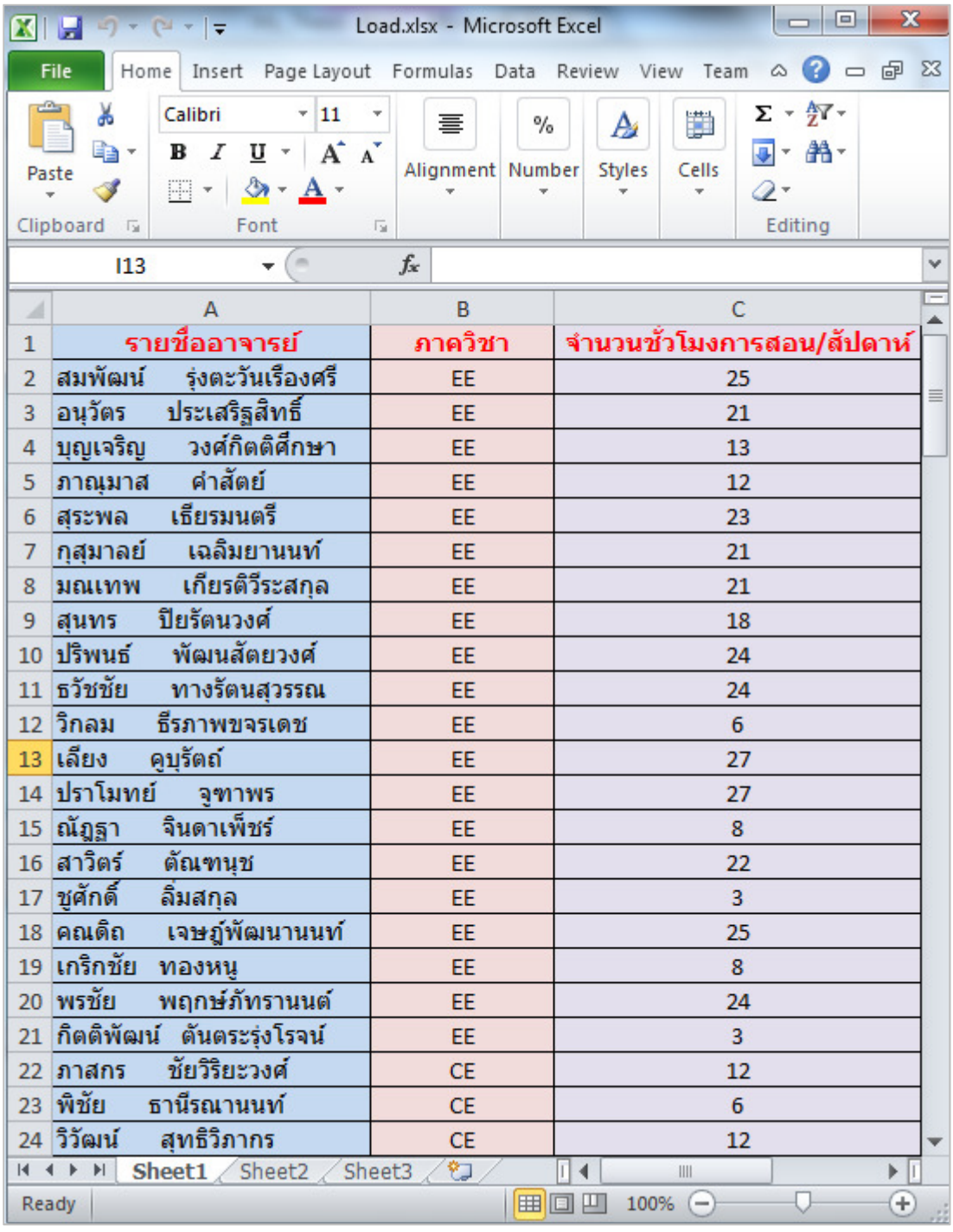

| วัน/เวลา | 08.00-09.00 | 09.00-10.00 | 10.00-11.00 | 11.00-12.00 | 12.00-13.00 | 13.00-14.00 | 14.00-15.00 | 15.00-16.00 | 16.00-17.00 | 17.00-18.00 | 18.00-19.00 | 19.00-20.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| จันทร์ | | | | 225-456(1) | 230-432(1) | | 210-391(3) | 237-403(1) | 213-351(1) | 226-316(1) | | 237-403(1) |
| อังคาร | | | 235-210(1) | | 235-230(1) | | 211-433(1) | | 226-383(1) | 230-424(1) | | 216-324(2) |
| พธ | | 230-425(1) | 225-558(1) | | | | 210-570(1) | | | 235-440(1) | | |
| พฤหัส | | 226-433(1) | 230-425(1) | 230-212(3) | | | | 213-211(1) | | | | 241-101(1 |
| ศุกร์ | 211-213(1) | 216-332(1) | | | 225-241(1) | | | 241-306(1) | 241-441(1) | | | 225-241(2) |

ภาพประกอบ 3.47 หน้าต่างแสดงตัวอย่างตารางการใช้ห้องเรียน

| A200 | 08.00-09.00 | 09.00-10.00 | 10.00-11.00 | 11.00-12.00 | 12.00-13.00 | 13.00-14.00 | 14.00-15.00 | 15.00-16.00 | 16.00-17.00 | 17.00-18.00 | 18.00-19.00 | 19.00-20.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mon. | | | | 225-456(1) | 230-432(1) | | 210-391(3) | 237-403(1) | 213-351(1) | 226-316(1) | | 237-403(1) |
| | | | | | | | | | | | | |
| Tue. | | | 235-210(1) | | 235-230(1) | | 211-433(1) | | 226-383(1) | 230-424(1) | | 216-324(2) |
| | | | | | | | | | | | | |
| Wed. | | 230-425(1) | 225-558(1) | | | | 210-570(1) | | | 235-440(1) | | |
| | | | | | | | | | | | | |
| Thu. | | 226-433(1) | 230-425(1) | 230-212(3) | | | | 213-211(1) | | | | 241-101(1 |
| | | | | | | | | | | | | |
| Fri. | 211-213(1) | 216-332(1) | | | 225-241(1) | | | 241-306(1) | 241-441(1) | | | 225-241(2) |
| | | | | | | | | | | | | |

ภาพประกอบ 3.48 ตัวอย่างตารางการใช้ห้องเรียน

(9) กดปุ่ม “ภาระงานสอนของอาจารย์” จะสามารถเรียกดูไฟล์ Excel ภาระงานสอนของอาจารย์ทุกคนในภาคการศึกษาที่ทำการจัดตารางสอน ดังภาพประกอบ 3.49

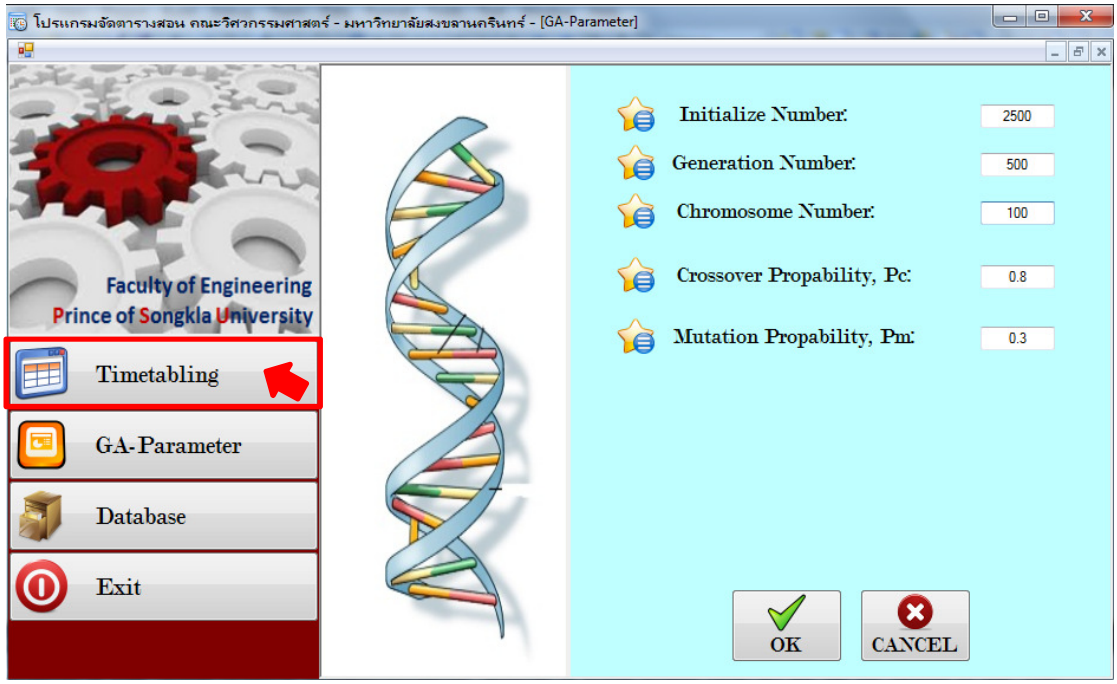



| | A | B | C |
|---|---|---|---|
| 1 | รายชื่ออาจารย์ | ภาควิชา | จำนวนชั่วโมงการสอน/สัปดาห์ |
| 2 | สมพัฒน์ รุ่งตะวันเรืองศรี | EE | 25 |
| 3 | อนุวัตร ประเสริฐสิทธิ์ | EE | 21 |
| 4 | บุญเจริญ วงศ์กิตติศึกษา | EE | 13 |
| 5 | ภาณุมาส ค่าสัตย์ | EE | 12 |
| 6 | สุระพล เธียรมนตรี | EE | 23 |
| 7 | กุสุมาลย์ เฉลิมยานนท์ | EE | 21 |
| 8 | มณเทพ เกียรติวีระสกุล | EE | 21 |
| 9 | สุนทร ปิยรัตนวงศ์ | EE | 18 |
| 10 | ปริพนธ์ พัฒนสัตยวงศ์ | EE | 24 |
| 11 | ธวัชชัย ทางรัตนสุวรรณ | EE | 24 |
| 12 | วิกลม ธีรภาพขจรเดช | EE | 6 |
| 13 | เลียง คูบุรัตถ์ | EE | 27 |
| 14 | ปราโมทย์ จูฑาพร | EE | 27 |
| 15 | ณัฏฐา จินดาเพ็ชร์ | EE | 8 |
| 16 | สาวิตร์ ตัณฑนุช | EE | 22 |
| 17 | ชูศักดิ์ ลิ่มสกุล | EE | 3 |
| 18 | คณดิถ เจษฎ์พัฒนานนท์ | EE | 25 |
| 19 | เกริกชัย ทองหนู | EE | 8 |
| 20 | พรชัย พฤกษ์ภัทรานนต์ | EE | 24 |
| 21 | กิตติพัฒน์ ตันตระรุ่งโรจน์ | EE | 3 |
| 22 | ภาสกร ชัยวิริยะวงศ์ | CE | 12 |
| 23 | พิชัย ธานีรณานนท์ | CE | 6 |
| 24 | วิวัฒน์ สุทธิวิภากร | CE | 12 |

ภาพประกอบ 3.49 ตัวอย่างภาระงานสอนของอาจารย์

### 1.4.3 เมนูเจเนติกอัลกอริทึม

เมื่อผู้ใช้เลือกเมนูนี้จะปรากฏหน้าต่างสำหรับกำหนดค่าเจเนติกอัลกอริทึมซึ่งเป็นหน้าจอสำหรับการกำหนดค่าพารามิเตอร์ให้กับโปรแกรม โดยสามารถป้อนข้อมูลพารามิเตอร์ที่เหมาะสมสำหรับการประมวลผลโปรแกรมเพื่อการค้นหาคำตอบที่ดีที่สุด พารามิเตอร์เจเนติกอัลกอริทึมที่กำหนดให้ป้อนข้อมูลได้ คือ Initialize Number, Generation Number, Chromosome Number, Pc และ Pm ซึ่งผู้วิจัยจะกำหนดค่าพารามิเตอร์ที่เหมาะสมไว้เป็นค่าเริ่มต้นของโปรแกรม ดังภาพประกอบ 3.50




ภาพประกอบ 3.50 หน้าต่างเมนูเจเนติกอัลกอริทึม

จากการพัฒนาโปรแกรมการจัดตารางสอนโดยประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึมในการประมวลผลเพื่อหาคำตอบที่ดีที่สุดดังที่ได้กล่าวมาแล้วนั้น ผู้วิจัยได้ทำการทดสอบโปรแกรมที่พัฒนาขึ้นกับข้อมูลตัวอย่างเพื่อประเมินผลลัพธ์โปรแกรม และทำการเปรียบเทียบผลลัพธ์ที่ได้กับตารางสอนเดิมที่กองทะเบียนและประมวลผลจัดไว้ ด้วยเครื่องคอมพิวเตอร์ประมวลผลบนระบบปฏิบัติการวินโดวส์ (Windows) ความเร็วหน่วยประมวลผล (CPU) Intel Core i5, 2.3 GHz โดยรายละเอียดของผลลัพธ์โปรแกรมจะแสดงไว้ในบทที่ 4

# บทที่ 4

# ผลลัพธ์โปรแกรมจัดตารางสอน

เมื่อได้ทำการออกแบบและพัฒนาโปรแกรมจัดตารางสอนโดยนำวิธีการของเจเนติกอัลกอริทึมเข้ามาประยุกต์ใช้แล้ว ผู้วิจัยได้ดำเนินการสร้างโปรแกรมต้นแบบขึ้นมาโดยอาศัยหลักการและข้อมูลที่ได้กล่าวไว้ในบทที่ 2 และ 3 สำหรับเนื้อหาที่จะกล่าวถึงในบทนี้ คือผลลัพธ์โปรแกรมจัดตารางสอนที่พัฒนาขึ้น ผลลัพธ์ที่ได้จะแบ่งเป็น 3 ส่วน คือส่วนการทดสอบความถูกต้องของโปรแกรม ส่วนการวิเคราะห์ความไวเพื่อหาค่าของพารามิเตอร์เจเนติกอัลกอริทึม และส่วนการเปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรมกับตารางสอนที่จัดโดยเจ้าหน้าที่จัดตารางสอนของกองทะเบียนและประมวลผล โดยผู้ทำวิจัยเป็นผู้ทดสอบ สำหรับการทดสอบโปรแกรมที่พัฒนาขึ้นด้วยเครื่องคอมพิวเตอร์ประมวลผลบนระบบปฏิบัติการวินโดวส์ ความเร็วหน่วยประมวลผล 2.3 GHz

ในการทดสอบเพื่อประเมินผลโปรแกรมและการเปรียบเทียบผลลัพธ์นั้น ก่อนที่จะประมวลผลโปรแกรมผู้วิจัยได้กำหนดข้อมูลที่จำเป็นทั้งหมดในการจัดตารางสอนไว้ในฐานข้อมูลโดยตรงสำหรับเรียกใช้ในการประมวลผลของโปรแกรม โดยข้อมูลที่ป้อนเข้าสู่ฐานข้อมูลประกอบด้วยข้อมูลต่างๆ ดังนี้

(1) ข้อมูลรายวิชาในคณะ โดยป้อนข้อมูลประเภทวิชา (บรรยายหรือปฏิบัติ), รหัสวิชา, ภาควิชา, ชั่วโมงการเรียนการสอน, ตอนที่เปิดสอน, จำนวนนักศึกษาที่ลงทะเบียนเรียนของแต่ละรายวิชา, ภาคการศึกษาที่จะเปิดสอนแต่ละรายวิชา, รหัสอาจารย์ผู้สอน และรหัสกลุ่มนักศึกษาที่เรียนรายวิชานั้นๆ ตามแผนการศึกษา

(2) ข้อมูลรายวิชานอกคณะ โดยป้อนข้อมูลรหัสวิชา, ชั่วโมงการเรียนการสอน, ตอนที่เปิดสอน, รหัสกลุ่มนักศึกษาที่เรียนรายวิชานั้นๆ ตามแผนการศึกษาและข้อมูลของวัน-เวลาที่กองทะเบียนและประมวลผลกำหนดไว้

(3) ข้อมูลอาจารย์ทั้งหมดของคณะวิศวกรรมศาสตร์ โดยป้อนข้อมูลชื่อ-นามสกุลของอาจารย์ และภาควิชาต้นสังกัดของอาจารย์แต่ละคน

(4) ข้อมูลกลุ่มนักศึกษาทั้งหมดของคณะวิศวกรรมศาสตร์ โดยป้อนข้อมูลชื่อกลุ่มนักศึกษา และจำนวนนักศึกษาของกลุ่มนักศึกษา

(5) ข้อมูลห้องเรียนทั้งหมดของคณะวิศวกรรมศาสตร์ โดยป้อนข้อมูลชื่อห้องเรียน, ความจุของห้องเรียน และภาควิชาของห้องเรียน

เนื่องจากงานวิจัยนี้มีวัตถุประสงค์หลักคือ เพื่อสร้างโปรแกรมคอมพิวเตอร์สำหรับจัดตารางสอนที่เหมาะสมสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยประยุกต์วิธีการของเจเนติกอัลกอริทึมเข้ามาใช้ในการหาคำตอบของปัญหา เพื่อให้ได้ตารางสอนที่ผิดเงื่อนไขในการจัดตารางสอนน้อยที่สุด ซึ่งถือว่าเป็นคำตอบหรือตารางสอนที่ดีที่สุด ดังนั้นการทดสอบเพื่อประเมินผลลัพธ์โปรแกรมจึงเป็นการทดสอบโปรแกรมเพื่อหาตารางสอนที่ผิดเงื่อนไขการจัดตารางสอนน้อยที่สุดที่โปรแกรมประมวลผลมาได้ การทดสอบเพื่อประเมินผลลัพธ์โปรแกรมแบ่งเป็น 2 ประเภท คือ การทดสอบความถูกต้องของโปรแกรม (validation) และการทดสอบเพื่อวิเคราะห์ความไวเพื่อหาค่าพารามิเตอร์เจเนติกอัลกอริทึม (sensibility) นอกจากนี้ก็จะทำการเปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรมกับตารางสอนที่จัดด้วยวิธีการแบบเดิม ซึ่งจะกล่าวในหัวข้อถัดไป

### 4.1 การทดสอบความถูกต้องของโปรแกรม

การทดสอบความถูกต้องของโปรแกรมเป็นการทดสอบโปรแกรมที่พัฒนาขึ้นว่ามีเงื่อนไขและข้อมูลตรงตามสิ่งที่ต้องการหรือไม่ ในการทดสอบความถูกต้องของโปรแกรมที่พัฒนาขึ้นนี้จะทำการทดสอบด้วยมือกับตัวอย่างข้อมูลที่มีขนาดเล็กเพื่อความง่ายและสะดวกต่อการประมวลผลและการตรวจสอบความถูกต้อง โดยจะทำการทดสอบกับตัวอย่างตารางสอน 1 ตาราง โดยข้อมูลที่ใช้สำหรับการทดสอบความถูกต้องของโปรแกรมมีดังนี้

(1) รายวิชา 43 รายวิชา แบ่งเป็นรายวิชาในคณะประเภทบรรยาย รายวิชาในคณะประเภทปฏิบัติ และรายวิชานอกคณะรายละเอียดตามตาราง 4.1, 4.2 และ 4.3 ตามลำดับ

(2) อาจารย์ผู้สอน 15 คน รายละเอียดตามตาราง 4.4

(3) กลุ่มนักศึกษา 10 กลุ่ม รายละเอียดตามตาราง 4.5

(4) ห้องเรียน 6 ห้อง รายละเอียดตามตาราง 4.6

โดยทำการจัดตารางสอนในวันจันทร์-วันศุกร์ เวลา 08.00-19.50 น. คาบละ 50 นาที มีระยะเวลาเผื่อระหว่างเปลี่ยนคาบเรียน 10 นาที เพื่อใช้ในการเดินทางกรณีที่ห้องเรียนอยู่ไกล

ตาราง 4.1 รายละเอียดตัวอย่างรายวิชาบรรยายที่ใช้ในการทดสอบความถูกต้อง

| ID | Subject | Department | LectureHour | Section | Student Number | Teacher_ID | Student_ID |
|---|---|---|---|---|---|---|---|
| 1 | L1 | IE | 3 | 1 | 60 | 1 | 1,5 |
| 2 | L2 | IE | 3 | 1 | 60 | 2 | 8 |
| 3 | L3 | IE | 3 | 1 | 60 | 3 | 1 |
| 4 | L4 | IE | 3 | 1 | 60 | 4 | 1,5 |
| 5 | L5 | IE | 3 | 1 | 60 | 5 | 1,5 |
| 6 | L6 | IE | 3 | 1 | 60 | 6 | 3 |
| 7 | L7 | IE | 3 | 1 | 60 | 7 | 5 |
| 8 | L8 | IE | 3 | 1 | 60 | 8 | 9 |
| 9 | L9 | IE | 3 | 1 | 60 | 9 | 9 |
| 10 | L10 | IE | 3 | 1 | 60 | 7 | 9,7 |
| 11 | L11 | IE | 3 | 1 | 60 | 8,9 | 9,4 |
| 12 | L12 | IE | 3 | 1 | 60 | 9 | 9,4 |
| 13 | L13 | IE | 3 | 1 | 60 | 1 | 10,4 |
| 14 | L14 | IE | 3 | 1 | 60 | 2 | 10,4,7 |
| 15 | L15 | IE | 3 | 1 | 60 | 3 | 10 |
| 16 | L16 | IE | 3 | 1 | 60 | 4 | 8,7 |
| 17 | L17 | IE | 3 | 1 | 60 | 8 | 8,7 |
| 18 | L18 | IE | 3 | 1 | 60 | 9 | 8 |
| 19 | L19 | IE | 3 | 1 | 60 | 8 | 2 |
| 20 | L20 | IE | 3 | 1 | 60 | 9 | 2,6 |
| 21 | L21 | ChE | 3 | 1 | 60 | 10,14,15 | 2,6,3 |
| 22 | L22 | ChE | 3 | 1 | 60 | 11 | 2,6,3 |
| 23 | L23 | ChE | 3 | 1 | 60 | 12 | 2,6 |
| 24 | L24 | ChE | 2 | 1 | 60 | 13 | 1 |
| 25 | L25 | ChE | 2 | 1 | 60 | 14 | 9 |
| 26 | L26 | ChE | 2 | 1 | 60 | 15 | 8 |
| 27 | L27 | ChE | 2 | 1 | 60 | 10 | 2,3 |

ตาราง 4.1 รายละเอียดตัวอย่างรายวิชาบรรยายที่ใช้ในการทดสอบความถูกต้อง (ต่อ)

| ID | Subject | Department | LectureHour | Section | Student Number | Teacher_ID | Student_ID |
|---|---|---|---|---|---|---|---|
| 28 | L28 | ChE | 2 | 1 | 60 | 11 | 3 |
| 29 | L29 | ChE | 1 | 1 | 60 | 14 | 3,4,7 |
| 30 | L30 | ChE | 4 | 1 | 60 | 15 | 10,6 |

ตาราง 4.2 รายละเอียดตัวอย่างรายวิชาปฏิบัติที่ใช้ในการทดสอบความถูกต้อง

| ID | Subject | Department | PracticalHour | Section | Student Number | Teacher_ID | Student_ID |
|---|---|---|---|---|---|---|---|
| 1 | P1 | IE | 3 | 1 | 60 | 1,2,6,7 | 1,5 |
| 2 | P2 | IE | 3 | 1 | 60 | 2,4, | 5 |
| 3 | P3 | IE | 3 | 1 | 60 | 3,4,6 | 2 |
| 4 | P4 | IE | 3 | 1 | 60 | 3,4,6 | 3 |
| 5 | P5 | IE | 3 | 1 | 60 | 5,1,7 | 6 |
| 6 | P6 | IE | 3 | 1 | 60 | 5 | 7 |
| 7 | P7 | ChE | 3 | 1 | 60 | 10,12,13 | 8 |
| 8 | P8 | ChE | 3 | 1 | 60 | 10,11 | 9 |
| 9 | P9 | ChE | 2 | 1 | 60 | 11 | 10 |

ตาราง 4.3 รายละเอียดตัวอย่างรายวิชานอกคณะที่ใช้ในการทดสอบความถูกต้อง

| ID | Subject | Lecture Hour | Sec | Student ID | Period1 | Period2 | Period3 |
|---|---|---|---|---|---|---|---|
| 1 | 322-101 | 3 | 1 | 1,5 | Mon,08.00-08.50 | Wed,08.00-08.50 | Fri,08.00-08.50 |
| 2 | 322-101 | 3 | 2 | 8 | Mon,09.00-09.50 | Wed,09.00-09.50 | Fri,09.00-09.50 |
| 3 | 890-101 | 3 | 1 | 1,5 | Mon,09.00-09.50 | Wed,09.00-09.50 | Fri,09.00-09.50 |
| 4 | 890-101 | 3 | 2 | 8 | Mon,10.00-10.50 | Wed,10.00-10.50 | Fri,10.00-10.50 |

ตาราง 4.4 รายละเอียดตัวอย่างอาจารย์ผู้สอนที่ใช้ในการทดสอบความถูกต้อง

| Teacher_ID | TeacherName | Department |
|---|---|---|
| 1 | TIE1 | IE |
| 2 | TIE2 | IE |
| 3 | TIE3 | IE |
| 4 | TIE4 | IE |
| 5 | TIE5 | IE |
| 6 | TIE6 | IE |
| 7 | TIE7 | IE |
| 8 | TIE8 | IE |
| 9 | TIE9 | IE |
| 10 | TChE1 | ChE |
| 11 | TChE2 | ChE |
| 12 | TChE3 | ChE |
| 13 | TChE4 | ChE |
| 14 | TChE5 | ChE |
| 15 | TChE6 | ChE |

ตาราง 4.5 รายละเอียดตัวอย่างกลุ่มนักศึกษาที่ใช้ในการทดสอบความถูกต้อง

| Student_ID | StudentName | StudentNumber | Department |
|---|---|---|---|
| 1 | 2IE | 60 | IE |
| 2 | 3IE | 60 | IE |
| 3 | 4IE | 60 | IE |
| 4 | IEOLD | 60 | IE |
| 5 | 2MfE | 60 | IE |
| 6 | 3MfE | 60 | IE |
| 7 | 4MfE | 60 | IE |
| 8 | 2ChE | 60 | ChE |
| 9 | 3ChE | 60 | ChE |
| 10 | 4ChE | 60 | ChE |

ตาราง 4.6 รายละเอียดตัวอย่างห้องเรียนที่ใช้ในการทดสอบความถูกต้อง

| Room_ID | Room_Cap | Department_ID |
|---|---|---|
| A200 | 60 | AA |
| A400 | 60 | AA |
| KE208 | 60 | ChE |
| IEF309 | 60 | IE |
| IELAB | 60 | IE |
| ChELAB | 60 | ChE |

ขั้นตอนในการทดสอบความถูกต้องของโปรแกรมจะเริ่มต้นโดยโปรแกรมจะทำการดึงข้อมูลตัวอย่างข้างต้นจากฐานข้อมูล และประมวลผลคำตอบที่ดีที่สุดออกมา โดยจะทำการทวนสอบตั้งแต่กระบวนการแปลงโครโมโซมที่ได้ให้อยู่ในรูปของตาราง ซึ่งจะขอยกตัวอย่างส่วนหนึ่งของโครโมโซมมาอธิบายดังภาพประกอบ 4.1 คือยีนลำดับที่ 1-60 ของโครโมโซม จากทั้งหมด 360 ยีน ตัวเลข 1-60 ด้านล่างรหัสยีน คือ ตัวเลขที่บ่งบอกถึงตำแหน่งยีนที่สร้างขึ้นเพื่อง่ายต่อความเข้าใจ ซึ่งจะมีความสัมพันธ์กับตัวเลข 1-60 ในตาราง 4.7 ตัวเลขของตำแหน่งยีน สามารถบ่งบอกได้ถึงวัน-เวลา และห้องเรียนได้ โดยตำแหน่งที่ยกตัวอย่างมานั้นเป็นตำแหน่งของห้องเรียน A200 เมื่อเข้าใจถึงตำแหน่งยีนแล้ว ก็จะทำการอ่านค่ารหัสยีนเพื่อแปลงใส่ตารางของห้องเรียน A200 โดยอันดับแรกจะอ่านที่ตัวอักษรก่อน ถ้าเป็น ตัว E หมายถึงไม่มีการเรียนการสอนคาบเรียนนั้น หากเป็นอักษร L หมายถึงวิชาบรรยาย อักษร P หมายถึงวิชาปฏิบัติ และอักษร F หมายถึงวิชานอกคณะฯ ซึ่งจะใช้ตัวอักษร L และ P เป็นตัวตรวจสอบว่าประเภทวิชาเหมาะสมกับประเภทห้องหรือไม่ ตัวเลข 3 หลักแรกจะบอกรหัสวิชา และหลักสุดท้ายจะบอกชั่วโมงของรายวิชานั้น จากโครโมโซมตัวอย่างเป็น E ตั้งแต่ตำแหน่งที่ 1-9 หมายความว่าไม่มีการเรียนการสอนในห้องเรียน A200 ตั้งแต่ 08.00-16.50 น. และในตำแหน่งที่ 10 เป็นรหัส L0021 หมายความว่า รายวิชา L002 คาบเรียนที่ 1 ถูกจัดให้มีการเรียนการสอนในห้องเรียน A200 เวลา 17.00-17.50 น.จึงนำรหัสรายวิชาไปใส่ในตารางการใช้ห้องเรียนนั้นดังภาพประกอบ 4.2 ทำการอ่านและแปลงรหัสยีนจนครบทั้งโครโมโซมก็จะได้ตารางการใช้เรียนจนครบทุกห้องเรียนที่มีการเรียนการสอน

จากภาพประกอบ 4.2 สามารถแปลงออกมาเป็นตารางสอนของอาจารย์และตารางเรียนตามกลุ่มนักศึกษาได้ เช่น จากรายวิชา L2 ที่ถูกจัดให้มีการเรียนการสอนห้อง A200 วันจันทร์ เวลา 17.00-18.00 น. จะได้ว่า อาจารย์ TIE 2 สอนวิชา L2 ห้อง A200 วันจันทร์ เวลา 17.00-17.50 น. และนักศึกษากลุ่ม 2ChE เรียนวิชา L2 ห้อง A200 วันจันทร์ เวลา 17.00-17.50 น. โดยการดึงข้อมูล

จากฐานข้อมูล ทำการสร้างตารางสอนของอาจารย์และตารางสอนของนักศึกษาจนครบทุกรายวิชา จะได้ตารางสอนอาจารย์ ดังภาพประกอบ 4.3 และได้ตารางเรียนตามกลุ่มนักศึกษา ดังภาพประกอบ 4.4 เป็นต้น แล้วทำการตรวจสอบตารางสอนด้วยสายตาคนจัดตารางสอนที่ได้ว่าผิดเงื่อนไขในการจัดตารางสอนหรือไม่ แล้วเปรียบกับการตรวจสอบของโปรแกรมว่าตรงกันหรือไม่ ซึ่งได้ผลตามตาราง 4.8

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E0001 | E0002 | E0003 | E0004 | E0005 | E0006 | E0007 | E0008 | E0009 | L0021 | E0010 | E0011 | E0012 | E0013 | L0191 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| E0014 | E0015 | E0016 | E0017 | E0018 | L0152 | L0161 | E0019 | L0162 | E0020 | L0132 | E0021 | E0022 | L0051 | L0221 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| L0073 | E0023 | E0024 | E0025 | E0026 | L0032 | E0027 | L0101 | E0028 | L0281 | L0163 | E0029 | L0193 | E0030 | L0052 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| L0043 | L0303 | L0093 | E0031 | E0032 | E0033 | E0034 | L0013 | L0181 | L0143 | L0142 | E0035 | L0102 | E0036 | L0171 |
| 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

ภาพประกอบ 4.1 ตัวอย่างส่วนหนึ่งของโครโมโซมที่ใช้ในการทดสอบความถูกต้อง

ตาราง 4.7 ตัวอย่างตำแหน่งของยีน

| | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 | 16.00-16.50 | 17.00-17.50 | 18.00-18.50 | 19.00-19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| จันทร์ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| อังคาร | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| พุธ | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| พฤหัสบดี | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| ศุกร์ | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

| | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 | 16.00-16.50 | 17.00-17.50 | 18.00-18.50 | 19.00-19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| จันทร์ | | | | | | | | | | L2 | | |
| อังคาร | | | L19 | | | | | | L15 | L16 | | |
| พุธ | | L13 | | | L5 | L22 | L7 | | | | | L3 |
| พฤหัสบดี | | L10 | | L28 | L16 | | L19 | | L5 | L4 | L30 | L9 |
| ศุกร์ | | | | | L1 | L18 | L14 | L14 | | L10 | | L17 |

ภาพประกอบ 4.2 ตารางการใช้ห้องเรียน A200

| TIE2 | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 | 16.00-16.50 | 17.00-17.50 | 18.00-18.50 | 19.00-19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Mon.** | | | | P1(1) | P1(1) | P1(1) | | | | L2(1) | | L2(1) |
| | | | | IELAB | IELAB | IELAB | | | | A200 | | IEF309 |
| **Tue.** | | | | P2(1) | P2(1) | P2(1) | | | | | | |
| | | | | IELAB | IELAB | IELAB | | | | | | |
| **Wed.** | L14(1) | | | | | | | | | | | |
| | A400 | | | | | | | | | | | |
| **Thu.** | | | | | | | | | | | L2(1) | |
| | | | | | | | | | | | IEF309 | |
| **Fri.** | | | | | | | L14(1) | L14(1) | | | | |
| | | | | | | | A200 | A200 | | | | |

ภาพประกอบ 4.3 ตัวอย่างตารางสอนอาจารย์

| 2ChE | 08.00-08.50 | 09.00-09.50 | 10.00-10.50 | 11.00-11.50 | 12.00-12.50 | 13.00-13.50 | 14.00-14.50 | 15.00-15.50 | 16.00-16.50 | 17.00-17.50 | 18.00-18.50 | 19.00-19.50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Mon.** | | 322-101(2) | 890-101(2) | | | | | | | L2(1) | | L2(1) |
| | | XXX | XXX | | | | | | | A200 | | IEF309 |
| **Tue.** | | | P7(1) | P7(1) | P7(1) | | L17(1) | | | L16(1) | | L16(1) |
| | | | ChELAB | ChELAB | ChELAB | | IEF309 | | | A200 | | A200 |
| **Wed.** | | 322-101(2) | 890-101(2) | | | | | | | | L26(1) | L18(1) |
| | | XXX | XXX | | | | | | | | A400 | A400 |
| **Thu.** | | | | | L16(1) | | L26(1) | | | L17(1) | L2(1) | |
| | | | | | A200 | | A400 | | | IEF309 | IEF309 | |
| **Fri.** | | 322-101(2) | 890-101(2) | | | L18(1) | | | | | L18(1) | L17(1) |
| | | XXX | XXX | | | A200 | | | | | IEF309 | A200 |

ภาพประกอบ 4.4 ตัวอย่างตารางเรียนตามกลุ่มนักศึกษา

ตาราง 4.8 จำนวนการขัดแย้งเงื่อนไขของตารางสอนตัวอย่าง

| **เงื่อนไข** | **จำนวนที่ผิดเงื่อนไข (ครั้ง)** |
|---|---|
| เงื่อนไขบังคับ | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 0 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 0 |
| การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน | 0 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องให้กับห้องปฏิบัติการ | 0 |
| เงื่อนไขเพื่อความสมบูรณ์ | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 16 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 6 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบเพื่อเรียนในคาบถัดไป | 22 |
| ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 18 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 0 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 3 |

จากการทดสอบความถูกต้องของโปรแกรมทั้งในกระบวนการสร้างโครโมโซม แปลงโครโมโซมเป็นตารางสอน และการตรวจสอบเงื่อนไขในการจัดตารางสอน โดยเปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรมกับผลลัพธ์จากการจัดด้วยวิธีการเดิมโดยเจ้าหน้าที่ ให้ผลลัพธ์ที่เหมือนกัน ดังนั้นจึงสรุปว่าโปรแกรมสามารถทำงานได้ถูกต้องตามที่ต้องการ

## 4.2 ผลลัพธ์จากการจัดตารางสอนของคณะวิศวกรรมศาสตร์ และการเปรียบเทียบผลลัพธ์

ผลลัพธ์ที่ได้จากโปรแกรมจัดตารางสอนโดยประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึม แสดงผลออกมาในรูปแบบของตารางสอนอาจารย์ ตารางเรียนตามกลุ่มนักศึกษา และตารางการใช้ห้องเรียน ซึ่งข้อมูลที่ใช้ในการประมวลผลของโปรแกรมเป็นข้อมูลที่ใช้ในการจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2553 สามารถสรุปได้ตามตาราง 4.9

ตาราง 4.9 ข้อมูลที่ใช้ในจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2553

| ข้อมูล | จำนวน |
|---|---|
| รายวิชา | 425 รายวิชา |
| - รายวิชาบรรยาย | 294 รายวิชา |
| - รายวิชาปฏิบัติ | 91 รายวิชา |
| - รายวิชานอกคณะ | 40 รายวิชา |
| อาจารย์ | 152 คน |
| กลุ่มนักศึกษา | 83 กลุ่ม |
| ห้องเรียน | 59 ห้อง |

จากตาราง 4.9 ได้นำข้อมูลดังกล่าวมาจัดตารางสอนโดยให้โปรแกรมประมวลผลหาคำตอบที่ดีที่สุดสำหรับตารางสอนที่มีระยะเวลาการเรียนการสอนคาบละ 50 นาที ในแต่ละวันมีคาบเรียนทั้งหมด 12 คาบ ตั้งแต่เวลา 08.00-19.50 น. ซึ่งจัดทำการเรียนการสอนในวันจันทร์-ศุกร์ ดังนั้นใน 1 สัปดาห์สามารถจัดการเรียนการสอนได้ 60 คาบต่อสัปดาห์ โดยมีห้องเรียนที่ใช้ในการจัดการเรียนการสอน 59 ห้องนั้นคือสามารถจัดทำการเรียนการสอนได้ถึง 3,540 คาบ จากการประมวลผลของโปรแกรมจนกระทั่งได้ผลลัพธ์ที่ดีที่สุดที่โปรแกรมประมวลผลออกได้ พบว่าตารางสอนที่ได้ไม่ขัดแย้งกับเงื่อนไขบังคับ แต่ยังคงมีการขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์อยู่ และสามารถสรุปจำนวนการขัดแย้งเงื่อนไขแต่ละข้อของตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นได้ตามตาราง 4.10 จากตารางดังกล่าวจะเห็นได้ว่าผลลัพธ์ที่ได้นี้ไม่มีการขัดแย้งกับเงื่อนไขบังคับใดๆ เลย นั้นหมายความว่าตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นเป็นตารางสอนที่เป็นไปได้

สามารถนำไปใช้จัดการเรียนการสอนได้จริง เนื่องจากไม่มีเวลาใดๆ ที่ซ้ำซ้อนกันเกิดขึ้นในตารางสอน

ตาราง 4.10 จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2553 (50 นาทีต่อคาบ)

| เงื่อนไข | จำนวนการขัดแย้งเงื่อนไข (ครั้ง) |
|---|---|
| **เงื่อนไขบังคับ** | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 0 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 0 |
| การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน | 0 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องจัดให้กับห้องปฏิบัติการ | 0 |
| **เงื่อนไขบังคับเพื่อความสมบูรณ์** | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 131 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 112 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 115 |
| ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 95 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 13 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 10 |

นอกจากนี้ยังได้ทำการประมวลผลโปรแกรมในการหาคำตอบที่ดีที่สุดสำหรับตารางสอนที่มีระยะการเรียนการสอนคาบละ 90 นาที เพิ่มเติมเนื่องจากในอนาคตอาจจะมีนโยบายของทางมหาวิทยาลัยที่จะเปลี่ยนแปลงลักษณะการเรียนการสอน จากเดิมที่มีการเรียนการสอน 50 นาทีต่อคาบเป็น 90 นาทีต่อคาบ เพื่อศึกษาลักษณะผลลัพธ์ที่ได้จากโปรแกรมจัดตารางสอนที่พัฒนาขึ้น ว่าหากมีระยะเวลาการเรียนการสอนที่เปลี่ยนไปโปรแกรมสามารถประมวลผลให้คำตอบเป็นอย่างไร โดยสิ่งที่ต้องเปลี่ยนแปลงในการประมวลผลของโปรแกรมคือฐานข้อมูลและลักษณะโครงสร้างกระบวนการทำงานของโปรแกรมในบางส่วน และการจัดตารางสอนแบบระยะเวลาการเรียนการสอนเป็น 90 นาทีต่อคาบ จะทำให้จำนวนคาบในแต่ละวันเปลี่ยนไปจาก 12 คาบต่อวัน

เป็น 8 คาบต่อวัน ตั้งแต่เวลา 08.00-19.50 น. ซึ่งจัดทำการเรียนการสอนในวันจันทร์-ศุกร์ ดังนั้นใน 1 สัปดาห์สามารถจัดการเรียนการสอนได้ 40 คาบต่อสัปดาห์ โดยมีห้องเรียนที่ใช้ในการจัดการเรียนการสอน 59 ห้องนั้นคือสามารถจัดทำการเรียนการสอนได้ถึง 2,360 คาบ ซึ่งจะเห็นได้ว่ามีคาบเรียนที่สามารถจัดตารางสอนได้มีจำนวนน้อยลง แต่อย่างไรก็ตามจำนวนครั้งในการจัดตารางสอนแต่ละรายวิชาก็น้อยลงตามไปด้วย เช่น จากเดิมที่ระยะเวลาการเรียนการสอนเป็น 50 นาทีต่อคาบ ในรายวิชา 3 หน่วยกิตจะต้องทำการจัดตารางสอนให้รายวิชานี้ 3 คาบ แต่เมื่อเปลี่ยนเป็น 90 คาบต่อนาที รายวิชา 3 หน่วยกิตก็จัดถูกจัดตารางสอนแค่ 2 คาบเท่านั้น จากการประมวลผลของโปรแกรมพบว่าตารางสอนที่ได้ไม่ขัดแย้งกับเงื่อนไขบังคับ แต่ยังคงมีการขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์อยู่ และสามารถสรุปจำนวนการขัดแย้งเงื่อนไขแต่ละข้อของตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นได้ตามตาราง 4.11

ตาราง 4.11 จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2553 (90 นาทีต่อคาบ)

| **เงื่อนไข** | **จำนวนการขัดแย้งเงื่อนไข (ครั้ง)** |
|---|---|
| **เงื่อนไขบังคับ** | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 0 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 0 |
| การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน | 0 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องจัดให้กับห้องปฏิบัติการ | 0 |
| **เงื่อนไขบังคับเพื่อความสมบูรณ์** | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 22 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 117 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 22 |
| ในแต่ละไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 91 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 6 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 2 |

จากตาราง 4.11 จะเห็นได้ว่าผลลัพธ์ที่ได้นี้ไม่มีการขัดแย้งกับเงื่อนไขบังคับใดๆ เลยเช่นเดียวกับการประมวลผลจัดตารางสอนแบบที่มีระยะเวลาการเรียนการสอนคาบละ 50 นาที นั้นหมายความว่าตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นเป็นตารางสอนที่เป็นไปได้ สามารถนำไปใช้จัดการเรียนการสอนได้จริง เนื่องจากไม่มีเวลาใดๆ ที่ซ้ำซ้อนกันเกิดขึ้นในตารางสอน

การเปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรมที่พัฒนาขึ้นกับตารางสอนที่ได้จากการจัดด้วยวิธีการแบบเดิมนั้น ทำการเปรียบเทียบโดยตารางสอนที่จัดด้วยวิธีการแบบเดิม ตารางสอนที่จัดโดยโปรแกรมจัดตารางสอนที่พัฒนาขึ้นทั้งแบบที่ระยะเวลาในการเรียนการสอนคาบละ 50 นาที และคาบละ 90 นาที ให้ผลการเปรียบเทียบตามตาราง 4.12 โดยวิธีการเปรียบเทียบจะทำการแปลงตารางสอนที่จัดด้วยวิธีการแบบเดิมมาแปลงให้อยู่ในรูปของโครโมโซม แล้วให้โปรแกรมทำการตรวจสอบเงื่อนไขในการจัดตารางสอน ทั้งนี้เพื่อต้องการให้ตัวตรวจสอบเป็นระบบเดียวกัน ผลการเปรียบเทียบในกรณีที่ระยะเวลาในการเรียนการสอนคาบละ 50 นาที พบว่าตารางสอนที่จัดด้วยโปรแกรมที่พัฒนาขึ้นสามารถให้ตารางสอนที่เป็นไปได้ นั้นคือไม่ขัดแย้งกับเงื่อนไขบังคับ หรือไม่มีการซ้ำซ้อนของเวลาใดๆ เกิดขึ้น ในขณะที่ตารางสอนที่จัดด้วยวิธีการแบบเดิมยังมีการขัดแย้งกับเงื่อนไขบังคับ หรือมีการซ้ำซ้อนของเวลาเกิดขึ้นจำนวนหลายครั้ง ในทางปฏิบัติแล้วตารางสอนที่จัดด้วยวิธีการแบบเดิมไม่สามารถนำไปใช้งานได้จริง แต่ตารางสอนที่จัดด้วยโปรแกรมสามารถนำไปใช้งานได้จริง ถึงแม้ว่ายังมีการละเมิดเงื่อนไขเพื่อความสมบูรณ์ไปบ้าง ทั้งก็เพื่อหลีกเลี่ยงไม่ให้เกิดการซ้ำซ้อนของเวลาขึ้น

ส่วนผลการเปรียบเทียบตารางสอนที่จัดด้วยโปรแกรมที่พัฒนาขึ้นกรณีตารางสอนที่มีระยะเวลาในการเรียนการสอนคาบละ 50 นาที กับตารางสอนที่มีระยะเวลาในการเรียนการสอนคาบละ 90 นาที พบว่าการจัดตารางสอนทั้ง 2 กรณีไม่ขัดแย้งกับเงื่อนไขบังคับเลย หมายความว่าโปรแกรมการจัดตารางสอนที่พัฒนาขึ้นสามารถให้คำตอบที่เป็นไปได้สำหรับการจัดตารางสอนทั้ง 2 กรณี และเมื่อพิจารณาถึงการขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์ของตารางสอนที่ได้จากโปรแกรมจัดตารางสอนที่พัฒนาขึ้น พบว่าการจัดตารางสอนแบบ 90 นาทีต่อคาบมีการขัดแย้งกับเงื่อนไขน้อยกว่าการจัดตารางสอนแบบ 50 นาที ถึง 216 ครั้ง ซึ่งถือว่าจัดตารางสอนแบบ 90 นาที เป็นตารางสอนที่เหมาะสมกว่าจัดตารางสอนแบบ 50 นาที แต่อย่างไรก็ตามในการจัดตารางสอนแบบ 90 นาทีต่อคาบยังมีข้อจำกัด คือ ไม่สามารถจัดรายวิชาที่ไม่ใช่รายวิชา 3 หน่วยกิตได้พอดีกับระยะเวลา เนื่องจากนิยามของหน่วยกิต คือ 1 หน่วยกิต เรียน 50 นาที

ดังนั้นสรุปว่าตารางสอนที่ได้จากโปรแกรมดีกว่าตารางสอนที่จัดด้วยวิธีการแบบเดิม เนื่องจากสามารถนำไปงานได้จริง ไม่มีการซ้ำซ้อนของเวลาเกิดขึ้นอีกทั้งยังเป็นตารางสอนที่น่าพอใจของอาจารย์และนักศึกษา

ตาราง 4.12 รายละเอียดการเปรียบเทียบจำนวนการขัดแย้งเงื่อนไขของตารางสอน ภาคการศึกษา 1/2553

| เงื่อนไข | จำนวนการขัดแย้งเงื่อนไข (ครั้ง) | | |
|---|---|---|---|
| | วิธีการเดิม | โปรแกรม ( 50 นาที) | โปรแกรม ( 90 นาที) |
| **เงื่อนไขบังคับ** | | | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 129 | 0 | 0 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 158 | 0 | 0 |
| การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน | 125 | 0 | 0 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องจัดให้กับห้องปฏิบัติการ | 0 | 0 | 0 |
| **เงื่อนไขบังคับเพื่อความสมบูรณ์** | | | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 60 | 131 | 22 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 43 | 112 | 117 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 18 | 115 | 22 |
| ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 74 | 95 | 91 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 55 | 13 | 6 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 25 | 10 | 2 |

นอกจากนี้ยังได้จัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2555 สามารถสรุปข้อมูลที่ใช้ในการจัดตารางสอนได้ตามตาราง 4.13

ตาราง 4.13 ข้อมูลที่ใช้ในจัดตารางสอนของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ภาคการศึกษา 1/2555

| ข้อมูล | จำนวน |
|---|---|
| รายวิชา<br>- รายวิชาบรรยาย<br>- รายวิชาปฏิบัติ<br>- รายวิชานอกคณะ | 445 รายวิชา<br>320 รายวิชา<br>85 รายวิชา<br>40 รายวิชา |
| อาจารย์ | 164 คน |
| กลุ่มนักศึกษา | 83 กลุ่ม |
| ห้องเรียน | 59 ห้อง |

จากตาราง 4.13 ข้อมูลที่ใช้ในการจัดตารางสอนของภาคการศึกษา 1/2555 มีจำนวนข้อมูลที่เพิ่มขึ้นจากข้อมูลที่ใช้ในการจัดตารางสอนของภาคการศึกษา 1/2553 คือมีจำนวนรายวิชาเพิ่มขึ้น 20 รายวิชา และจำนวนอาจารย์ผู้สอนเพิ่มขึ้น 12 คน ข้อมูลอื่นๆ ยังคงเดิม แล้วนำข้อมูลดังกล่าวมาจัดตารางสอนโดยให้โปรแกรมประมวลผลหาคำตอบที่ดีที่สุดสำหรับตารางสอนที่มีระยะเวลาการเรียนการสอนคาบละ 50 นาที จากการประมวลผลของโปรแกรม พบว่าตารางสอนที่ได้ไม่ขัดแย้งกับเงื่อนไขบังคับ แต่ยังคงมีการขัดแย้งกับเงื่อนไขเพื่อความสมบูรณ์อยู่ และสามารถสรุปจำนวนการขัดแย้งเงื่อนไขแต่ละข้อของตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นได้ตามตาราง 4.14

ตาราง 4.14 จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2555 (50 นาทีต่อคาบ)

| เงื่อนไข | จำนวนการขัดแย้งเงื่อนไข (ครั้ง) |
|---|---|
| **เงื่อนไขบังคับ** | |
| การซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน | 0 |
| การซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา | 0 |
| การซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน | 0 |
| การกำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องจัดให้กับห้องปฏิบัติการ | 0 |

ตาราง 4.14 จำนวนการขัดแย้งเงื่อนไขของตารางสอนภาคการศึกษา 1/2555 (50 นาทีต่อคาบ) (ต่อ)

| เงื่อนไข | จำนวนการขัดแย้งเงื่อนไข (ครั้ง) |
|---|---|
| **เงื่อนไขบังคับเพื่อความสมบูรณ์** | |
| ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 145 |
| รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 118 |
| ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 127 |
| ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 94 |
| ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 4 |
| ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 17 |

จากตาราง 4.14 จะเห็นได้ว่าผลลัพธ์ที่ได้นี้ไม่มีการขัดแย้งกับเงื่อนไขบังคับใดๆ เลย นั้นหมายความว่าตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นเป็นตารางสอนที่เป็นไปได้ สามารถนำไปใช้จัดการเรียนการสอนได้จริง เนื่องจากไม่มีเวลาใดๆ ที่ซ้ำซ้อนกันเกิดขึ้นในตารางสอน กองทะเบียนและประมวลผลสามารถนำผลจากการประมวลผลของโปรแกรมไปเป็นแนวทางในการจัดตารางสอนของภาคการศึกษา 1/2555 ได้

### 4.3 การวิเคราะห์ความไวเพื่อหาค่าของพารามิเตอร์เจเนติกอัลกอริทึม

การวิเคราะห์ความไว (sensitivity analysis) คือการศึกษาถึงการเปลี่ยนแปลงของคำตอบที่ดีที่สุด (optimal solution) เมื่อเกิดการเปลี่ยนแปลงส่วนใดส่วนหนึ่งของแบบจำลอง เช่น ค่าคงที่ ค่าพารามิเตอร์ การวิเคราะห์ศึกษาว่าปัจจัยใดมีผลต่อฟังก์ชันวัตถุประสงค์ของแบบจำลอง โดยจะกำหนดปัจจัยที่ต้องการทดสอบให้มีการเปลี่ยนแปลงค่าไป แต่ควบคุมปัจจัยอื่นๆ ให้คงที่ การวิเคราะห์ความไวสามารถบอกได้ว่าปัจจัยใดเป็นปัจจัยเสี่ยงที่ต้องระวังและพยายามควบคุมไม่ให้คลาดเคลื่อนมากจนเกินไป เพราะจะมีผลต่อฟังก์ชันวัตถุประสงค์ในเชิงลบ

ในการเลือกตัวแปรที่นำมาวิเคราะห์ความไวโดยทั่วไปมักพิจารณาตัวแปรที่มีความสำคัญ และผู้วิเคราะห์ความไวไม่มีความมั่นใจในความถูกต้องของข้อมูลที่ได้มา และต้องการประเมินว่าหากข้อมูลตัวเลขหรือข้อสมมุติที่ใช้มีความคลาดเคลื่อน จะทำให้ตัวเลขผลลัพธ์คำนวณ

ได้แตกต่างไปจากค่าเดิมมากน้อยเพียงใด ในกรณีของตัวเลขมักจะใช้ค่าสูงสุดหรือค่าต่ำสุดที่มีความเป็นไปได้มาใช้เป็นตัวแทนเพื่อการคำนวณในการวิเคราะห์ความไว

สำหรับงานวิจัยนี้ใช้การวิเคราะห์ความไวของผลการวิจัยจากการพัฒนาโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนโดยประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบ โดยการวิเคราะห์ความไวของพารามิเตอร์ของเจเนติกอัลกอริทึม เพื่อหาค่าพารามิเตอร์ที่เหมาะสมสำหรับขนาดข้อมูลที่จะจัดตารางสอนซึ่งจะส่งผลให้ได้คำตอบที่ดีที่สุดสำหรับข้อมูลนั้นๆ โดยจะวิเคราะห์การเปลี่ยนแปลงของคำตอบที่ดีที่สุดเมื่อค่าพารามิเตอร์ของเจเนติกอัลกอริทึมของโปรแกรมเปลี่ยนไป พารามิเตอร์ของเจเนติกอัลกอริทึมที่วิเคราะห์ได้แก่ จำนวนประชากร (population size) และจำนวนเจนเนอเรชัน (generation) โดยกำหนดขนาดของข้อมูลสำหรับประมวลผล ซึ่งขนาดของข้อมูลในที่นี้จะหมายถึงจำนวนยีนหรือจำนวนรายวิชา อาจารย์ผู้สอน กลุ่มนักศึกษาและห้องเรียนที่ใช้ประมวลผลโปรแกรมซึ่งกำหนดเป็น 7 ช่วง โดยแบ่งช่วงขนาดของข้อมูลเทียบกับขนาดของข้อมูลจริงเป็น 10 เปอร์เซ็นต์ 20 เปอร์เซ็นต์ 50 เปอร์เซ็นต์ 80 เปอร์เซ็นต์ 100 เปอร์เซ็นต์ 120 เปอร์เซ็นต์ และ 150 เปอร์เซ็นต์ของข้อมูลจริงตามตาราง 4.15

ตาราง 4.15 จำนวนยีนต่อ 1 โครโมโซมในแต่ละช่วงจำนวนยีน

| **ขนาดข้อมูล (เทียบกับขนาดข้อมูลจริง)** | **จำนวนยีนต่อ 1 โครโมโซม** |
|---|---|
| 10 เปอร์เซ็นต์ | 360 |
| 20 เปอร์เซ็นต์ | 720 |
| 50 เปอร์เซ็นต์ | 1,800 |
| 80 เปอร์เซ็นต์ | 2,880 |
| 100 เปอร์เซ็นต์ (ข้อมูลจริง) | 3,540 |
| 120 เปอร์เซ็นต์ | 4,320 |
| 150 เปอร์เซ็นต์ | 5,400 |

โดยที่ขนาดข้อมูลแต่ละช่วงนั้นสามารคำนวณความยาวของโครโมโซมหรือจำนวนยีนต่อ 1 โครโมโซมโดยเทียบอัตราส่วนของจำนวนยีนของข้อมูลจริงซึ่งมีจำนวนยีน 3,540 ยีนต่อโครโมโซม ยกตัวอย่างเช่น 10 เปอร์เซ็นต์ของขนาดข้อมูลจริงจะมีจำนวนยีน 354 ยีนต่อโครโมโซม แต่เนื่องจากจำนวนยีนสามารถคำนวณได้จากจำนวนวันที่จัดตารางสอนคูณกับจำนวนคาบต่อวันคูณกับจำนวนห้องเรียน ซึ่งจำนวนวันคือ จัด 5 วันต่อสัปดาห์ วันละ 12 คาบ ดังนั้นจึงจำเป็นต้องใช้ห้องเรียนจำนวน 6 ห้อง เพื่อให้ได้จำนวนยีนเท่ากับ 360 ยีนต่อโครโมโซม ซึ่งมีค่า

ใกล้เคียงกับจำนวนยีน 354 ต่อโครโมโซมซึ่งเป็น 10 เปอร์เซ็นต์ของขนาดข้อมูลจริง เป็นต้น ซึ่งในแต่ละช่วงขนาดของยีนก็จะใช้วิธีการคำนวณเช่นเดียวกัน ทำให้สามารถสรุปขนาดข้อมูลในแต่ละช่วงที่เทียบกับข้อมูลจริงเป็นช่วงจำนวนยีนและสามารถสรุปจำนวนของข้อมูลที่ใช้ในการจัดตารางสอนในแต่ละช่วงจำนวนยีนได้ตามตาราง 4.16

ตาราง 4.16 รายละเอียดข้อมูลที่ใช้ในการจัดตารางสอนในแต่ละช่วงจำนวนยีน

| **ช่วงจำนวนยีน** | **จำนวนยีนต่อโครโมโซม** | **จำนวนรายวิชา (รายวิชา)** | **จำนวนอาจารย์ (คน)** | **จำนวนกลุ่มนักศึกษา (กลุ่ม)** | **จำนวนห้องเรียน (ห้อง)** |
|---|---|---|---|---|---|
| น้อยกว่า 500 | 360 | 43 | 15 | 10 | 6 |
| 500-1,000 | 720 | 86 | 30 | 20 | 12 |
| 1,000-2,000 | 1,800 | 215 | 75 | 50 | 30 |
| 2,000-3,000 | 2,880 | 344 | 120 | 80 | 48 |
| 3,000-4,000 | 3,540 | 430 | 152 | 101 | 59 |
| 4,000-5,000 | 4,320 | 516 | 180 | 120 | 72 |
| มากกว่า 5,000 | 5,400 | 645 | 225 | 150 | 90 |

นอกจากนี้ในการวิเคราะห์ความไวจะทำการกำหนดค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8 และค่าความน่าจะเป็นในการมิวเตชันเท่ากับ 0.3 เป็นกรณีศึกษา เพื่อหาค่าจำนวนประชากรและจำนวนเจนเนอเรชันซึ่งเป็นค่าพารามิเตอร์เจเนติกอัลกอริทึม โดยค่าที่ใช้ในการวิเคราะห์ในแต่ละช่วงยีนนั้นกำหนดเป็นดังนี้

(1) จำนวนประชากรที่วิเคราะห์คือ 10, 20, 50 และ 100 โครโมโซม

(2) จำนวนเจนเนอเรชันที่วิเคราะห์คือ 100, 200, 300, 400, 500, 600, 700 และ 800 เจนเนอเรชัน

ผลจากการวิเคราะห์ความไวของพารามิเตอร์เจเนติกอัลกอริทึม จำนวนประชากรและจำนวนเจนเนอเรชัน ที่พิจารณาในแต่ละช่วงจำนวนยีนที่กำหนดเป็นดังนี้

### 4.3.1 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีนน้อยกว่า 500 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากร และจำนวนเจนเนอเรชัน เมื่อจำนวนยีนต่อโครโมโซมในการประมวลผลโปรแกรมน้อยกว่า 500 ยีนนั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.17

ตาราง 4.17 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีนน้อยกว่า 500 ยีน)

| **จำนวนประชากร (โครโมโซม)** | **เจนเนอเรชัน** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **เริ่มต้น** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** |
| **10** | 590 | 510 | 470 | 450 | 430 | 420 | 410 | 410 | 410 |
| **20** | 580 | 520 | 480 | 450 | 430 | 410 | 400 | 380 | 380 |
| **50** | 580 | 510 | 490 | 440 | 410 | 400 | 370 | 370 | 370 |
| **100** | 570 | 500 | 470 | 420 | 400 | 360 | 360 | 360 | 360 |

จากตาราง 4.17 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนน้อยกว่า 100 ยีน สามารถสรุปได้ตามตาราง 4.18 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากรและจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีนน้อยกว่า 500 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 360 คะแนน

ตาราง 4.18 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนน้อยกว่า 500 ยีน

| จำนวนประชากร | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|
| **10** | 600 | 410 | 30.50 | 33 |
| **20** | 700 | 380 | 34.48 | 72 |
| **50** | 600 | 370 | 36.20 | 165 |
| **100** | 500 | 360 | 36.82 | 252 |

### 4.3.2 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีน 500-1,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนในการประมวลผลโปรแกรม 500-1,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.19

ตาราง 4.19 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 500-1,000 ยีน)

| จำนวนประชากร (โครโมโซม) | เจนเนอเรชัน | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **10** | 2,610 | 2,100 | 1,960 | 1,870 | 1,860 | 1,810 | 1,730 | 1,670 | 1,670 |
| **20** | 2,590 | 2,060 | 1,920 | 1,840 | 1,800 | 1,670 | 1,590 | 1,590 | 1,590 |
| **50** | 2,540 | 2,020 | 1,910 | 1,820 | 1,750 | 1,640 | 1,520 | 1,520 | 1,520 |
| **100** | 2,480 | 2,000 | 1,830 | 1,670 | 1,580 | 1,460 | 1,460 | 1,460 | 1,460 |

จากตาราง 4.19 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีน 500-1,000 ยีน สามารถสรุปได้ตามตาราง 4.20 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีน 500-1,000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 1,460 คะแนน

ตาราง 4.20 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 500-1,000 ยีน

| จำนวนประชากร | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|
| **10** | 700 | 1,670 | 36.02 | 75 |
| **20** | 600 | 1,590 | 38.61 | 159 |
| **50** | 600 | 1,520 | 40.16 | 268 |
| **100** | 500 | 1,460 | 41.13 | 478 |

### 4.3.3 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีน 1,000-2,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชันเมื่อจำนวนยีนในการประมวลผลโปรแกรม 1,000-2,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.21

ตาราง 4.21 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 1,000-2,000 ยีน)

| จำนวนประชากร (โครโมโซม) | เจนเนอเรชัน | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **10** | 9,020 | 8,130 | 7,430 | 6,420 | 5,880 | 5,850 | 5,540 | 5,540 | 5,540 |
| **20** | 9,110 | 8,180 | 7,450 | 7,030 | 6,490 | 6,080 | 5,780 | 5,430 | 5,430 |
| **50** | 8,920 | 7,760 | 6,970 | 6,090 | 5,880 | 5,560 | 5,420 | 5,220 | 5,220 |
| **100** | 8,830 | 7,430 | 6,610 | 5,980 | 5,560 | 5,040 | 5,040 | 5,040 | 5,040 |

จากตาราง 4.21 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีน 1,000-2,000 ยีน สามารถสรุปได้ตามตาราง 4.22

ตาราง 4.22 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 1,000-2,000 ยีน

| จำนวนประชากร | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|
| **10** | 600 | 5,540 | 38.58 | 156 |
| **20** | 700 | 5,430 | 40.40 | 343 |
| **50** | 700 | 5,220 | 41.47 | 628 |
| **100** | 500 | 5,040 | 42.92 | 1,116 |

จากตาราง 4.22 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จาก

การวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีน 1,000-2,000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 5,040 คะแนน

### 4.3.4 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีน 2,000-3,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชันเมื่อจำนวนยีนในการประมวลผลโปรแกรม 2,000-3,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.23

ตาราง 4.23 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 2,000-3,000 ยีน)

| จำนวนประชากร (โครโมโซม) | เจนเนอเรชัน | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **10** | 13,120 | 11,320 | 10,520 | 9,110 | 8,140 | 8,140 | 8,140 | 8,140 | 8,140 |
| **20** | 13,220 | 11,540 | 10,320 | 9,250 | 8,120 | 8,120 | 8,120 | 8,120 | 8,120 |
| **50** | 12,900 | 11,210 | 10,560 | 9,180 | 8,270 | 7,980 | 7,550 | 7,550 | 7,550 |
| **100** | 12,900 | 11,670 | 10,780 | 9,060 | 8,140 | 7,200 | 7,200 | 7,200 | 7,200 |

จากตาราง 4.23 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีน 2,000-3,000 ยีน สามารถสรุปได้ตามตาราง 4.24 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลัก

ของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีน 2,000-3,000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 7,200 คะแนน

ตาราง 4.24 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 2,000-3,000 ยีน

| จำนวนประชากร | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|
| **10** | 400 | 8,140 | 37.96 | 239 |
| **20** | 400 | 8,120 | 38.58 | 501 |
| **50** | 600 | 7,550 | 41.47 | 1,013 |
| **100** | 500 | 7,200 | 44.19 | 1,902 |

### 4.3.5 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีน 3,000-4,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชันเมื่อจำนวนยีนในการประมวลผลโปรแกรม 3,000-4,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.25 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีน 3,000-4,000 ยีน สามารถสรุปได้ตามตาราง 4.26 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจน

เนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีน 3,000-4,000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 9,480 คะแนน

ตาราง 4.25 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 3,000-4,000 ยีน)

| **จำนวนประชากร (โครโมโซม)** | **เจนเนอเรชัน** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **เริ่มต้น** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** |
| **10** | 18,950 | 14,720 | 13,690 | 12,540 | 11,970 | 11,270 | 10,590 | 10,350 | 10,350 |
| **20** | 19,110 | 14,850 | 13,700 | 12,720 | 11,880 | 11,310 | 10,620 | 10,190 | 10,190 |
| **50** | 19,720 | 14,760 | 13,420 | 12,590 | 11,560 | 10,920 | 9,940 | 9,940 | 9,940 |
| **100** | 19,980 | 14,820 | 13,570 | 11,970 | 11,970 | 10,760 | 9,480 | 9,480 | 9,480 |

ตาราง 4.26 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 3,000-4,000 ยีน

| **จำนวนประชากร** | **เจนเนอเรชันที่พบคำตอบที่ดีที่สุด** | **ค่าความเหมาะสมที่ต่ำที่สุด** | **การลดลงของค่าความเหมาะสม (%)** | **เวลาในการประมวลผลโปรแกรม (วินาที)** |
|---|---|---|---|---|
| **10** | 700 | 10,350 | 45.38 | 422 |
| **20** | 700 | 10,190 | 46.68 | 786 |
| **50** | 600 | 9,940 | 49.60 | 1,409 |
| **100** | 500 | 9,480 | 52.55 | 2,889 |

### 4.3.6 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีน 4,000-5,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชันเมื่อจำนวนยีนในการประมวลผลโปรแกรม 4,000-5,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800

เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.27

ตาราง 4.27 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (จำนวนยีน 4,000-5,000 ยีน)

| **จำนวนประชากร (โครโมโซม)** | **เจนเนอเรชัน** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **เริ่มต้น** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** |
| **10** | 27,090 | 20,760 | 19,670 | 18,720 | 17,650 | 16,970 | 16,010 | 15,960 | 15,960 |
| **20** | 29,070 | 20,850 | 19,780 | 18,810 | 17,020 | 16,000 | 14,650 | 14,650 | 14,650 |
| **50** | 29,650 | 21,010 | 19,520 | 18,000 | 16,520 | 15,430 | 14,620 | 14,620 | 14,620 |
| **100** | 29,650 | 20,960 | 18,850 | 16,960 | 15,780 | 14,560 | 14,560 | 14,560 | 14,560 |

จากตาราง 4.27 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีน 4,000-5,000 ยีน สามารถสรุปได้ตามตาราง 4.28

ตาราง 4.28 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนเท่ากับ 4,000-5,000 ยีน

| **จำนวนประชากร** | **เจนเนอเรชันที่พบคำตอบที่ดีที่สุด** | **ค่าความเหมาะสมที่ต่ำที่สุด** | **การลดลงของค่าความเหมาะสม (%)** | **เวลาในการประมวลผลโปรแกรม (วินาที)** |
|---|---|---|---|---|
| **10** | 700 | 15,960 | 41.09 | 685 |
| **20** | 600 | 14,650 | 49.60 | 1,236 |
| **50** | 600 | 14,620 | 50.70 | 2,644 |
| **100** | 500 | 14,560 | 50.90 | 5,712 |

จากตาราง 4.28 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีน 4,000-5000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 14,560 คะแนน

### 4.3.7 ผลการวิเคราะห์ความไวของพารามิเตอร์เมื่อจำนวนยีนมากกว่า 5,000 ยีน

การวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชันเมื่อจำนวนยีนในการประมวลผลโปรแกรมมากกว่า 5,000 ยีน นั้นจะพิจารณาค่าความเหมาะสมที่น้อยที่สุดที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 เมื่อพิจารณาจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.29

ตาราง 4.29 ค่าความเหมาะสมที่น้อยที่สุด (คะแนน) ที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อจำนวนประชากรเท่ากับ 10, 20, 50 และ 100 โครโมโซม (ยีนมากกว่า 5,000 ยีน)

| **จำนวนประชากร (โครโมโซม)** | **เจนเนอเรชัน** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **เริ่มต้น** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** |
| **10** | 48,550 | 32,700 | 30,780 | 28,950 | 27,170 | 25,340 | 25,340 | 25,340 | 25,340 |
| **20** | 48,120 | 31,850 | 30,890 | 28,740 | 27,020 | 25,340 | 25,340 | 25,340 | 25,340 |
| **50** | 47,960 | 31,540 | 30,530 | 27,980 | 26,950 | 24,590 | 22,560 | 22,560 | 22,560 |
| **100** | 46,850 | 29,760 | 28,720 | 26,730 | 24,050 | 22,480 | 20,520 | 20,520 | 20,520 |

จากตาราง 4.29 เมื่อพิจารณาค่าความเหมาะสมที่น้อยที่สุดในแต่ละจำนวนประชากร พบว่าค่าความเหมาะสมมีแนวโน้มลดลงเมื่อจำนวนประชากรเพิ่มขึ้น และในแต่ละจำนวนประชากรค่าความเหมาะสมจำลดลงและคงที่เมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น

จากผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนมากกว่า 5,000 ยีน สามารถสรุปได้ดังตาราง 4.30 เมื่อจำนวนประชากรเพิ่มขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดลดลง เปอร์เซ็นต์การลดลงของค่าความเหมาะสมเมื่อเปรียบเทียบกับเจนเนอเรชันเริ่มต้นมีค่าเพิ่มขึ้น และใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นตามจำนวนประชากร สำหรับงานวิจัยนี้จะพิจารณาค่าความเหมาะสมที่น้อยที่สุดเป็นเป้าหมายหลักของการวิจัย ดังนั้นพารามิเตอร์ที่ได้จากการวิเคราะห์ความไวคือ จำนวนประชากร และจำนวนเจนเนอเรชันที่เหมาะสมสำหรับข้อมูลที่มีจำนวนยีนมากกว่า 5,000 ยีน คือ จำนวนประชากรเท่ากับ 100 โครโมโซม และจำนวนเจนเนอเรชันเท่ากับ 600 เจนเนอเรชัน ซึ่งมีค่าความเหมาะสมที่น้อยที่สุดเท่ากับ 20,520 คะแนน

ตาราง 4.30 ผลการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อจำนวนยีนมากกว่า 5,000 ยีน

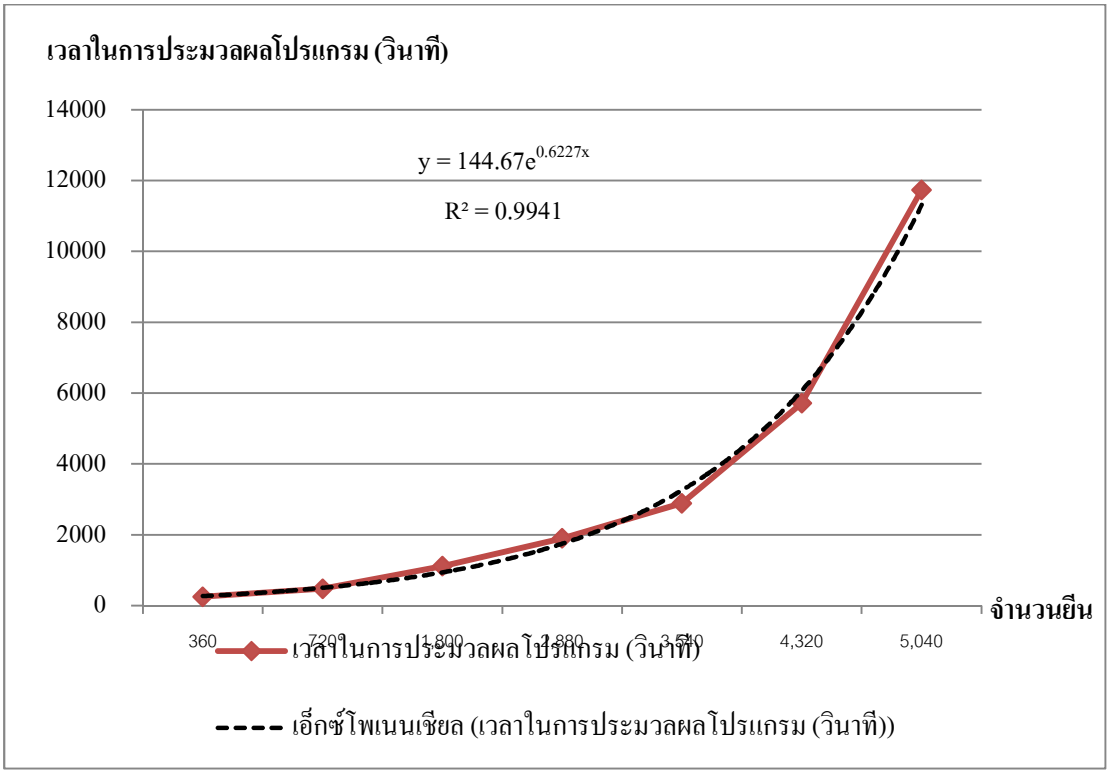
| จำนวนประชากร | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|
| **10** | 500 | 25,340 | 47.80 | 1,466 |
| **20** | 500 | 25,340 | 47.33 | 2,713 |
| **50** | 600 | 22,560 | 52.96 | 5,802 |
| **100** | 600 | 20,520 | 56.20 | 11,735 |

กล่าวโดยสรุปสำหรับการวิเคราะห์ความไวของพารามิเตอร์ จำนวนประชากรและจำนวนเจนเนอเรชัน เมื่อกำหนดจำนวนยีนแต่ละช่วงที่ต่างกัน พบว่าค่าพารามิเตอร์ที่เหมาะสมสำหรับจำนวนยีนในแต่ละช่วง แสดงได้ตามตาราง 4.31 สรุปได้ว่าจำนวนประชากรที่เหมาะสมสำหรับแต่ละช่วงจำนวนยีน คือ 100 โครโมโซมโดยจำนวนประชากรที่มากขึ้นจะทำให้ค่าความเหมาะสมที่น้อยที่สุดมีค่าลดลง จำนวนเจนเนอเรชันที่เหมาะสมสำหรับช่วงจำนวนยีนน้อยกว่า 500, 500-1,000, 1,000-2,000, 2,000-3,000, 3,000-4,000, 4,000-5,000 ยีน คือ 500 เจนเนอเรชัน และจำนวนยีนมากกว่า 5,000 ยีน คือ 600 เจนเนอเรชัน โดยจำนวนยีนจะมีผลต่อเวลาในการประมวลผลโปรแกรม คือจำนวนยีนที่มากขึ้นจะทำให้โปรแกรมใช้เวลาในการประมวลผลนานมากขึ้นตามไปด้วย ซึ่งอาจจะต้องมีการพัฒนาโปรแกรมนี้ต่อไปเพื่อให้โปรแกรมมีประสิทธิภาพในการทำงานมากที่สุด โดยกราฟแสดงการเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้น

(จำนวนยีนที่วิเคราะห์คือ 360, 720, 1,800, 2,880, 3,540, 4,320 และ 5,400) แสดงได้ดังภาพประกอบ 4.5

ตาราง 4.31 ค่าพารามิเตอร์ที่เหมาะสมสำหรับจำนวนยีนในแต่ละช่วงจำนวนยีน

| ช่วงจำนวนยีน | จำนวนประชากร (โครโมโซม) | จำนวนเจนเนอเรชัน | ค่าความเหมาะสมที่ต่ำที่สุด | การลดลงของค่าความเหมาะสม (%) | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|---|---|
| น้อยกว่า 500 | 100 | 500 | 360 | 36.28 | 252 |
| 500-1,000 | 100 | 500 | 1,460 | 41.24 | 478 |
| 1,000-2,000 | 100 | 500 | 5,040 | 42.89 | 1,116 |
| 2,000-3,000 | 100 | 500 | 7,200 | 44.21 | 1,902 |
| 3,000-4,000 | 100 | 500 | 9,480 | 52.55 | 2,889 |
| 4,000-5,000 | 100 | 500 | 14,560 | 50.90 | 5,712 |
| มากกว่า 5,000 | 100 | 600 | 20,520 | 57.11 | 11,735 |




ภาพประกอบ 4.5 การเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้น

จากภาพประกอบ 4.5 กราฟแสดงการเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้นซึ่งมีลักษณะการเพิ่มขึ้นของเวลาในการประมวลผลเป็นแบบเอ็กโปเนนเชียล แสดงเป็นสมการคือ $y = 144.67e^{0.6227x}$

นอกจากค่าพารามิเตอร์ต่างๆ ที่กล่าวมาข้างต้นแล้ว ยังมีพารามิเตอร์อีกหนึ่งตัวที่คิดว่าน่าจะมีผลต่อคำตอบที่ดีที่สุดนั้นคือ จำนวนประชากรเบื้องต้น (initialize) คือ จำนวนเริ่มต้นที่กำหนดให้โปรแกรมประมวลผลจำนวนโครโมโซมเพื่อที่จะคัดเลือกไปเป็นประชากรที่เข้าสู่กระบวนการเจเนติกอัลกอริทึมตามจำนวนประชากรที่กำหนด เนื่องจากสมมุติฐานที่ว่า หากเพิ่มพื้นที่ในการหาคำตอบให้มาก ค่าคำตอบที่ดีที่สุดของปัญหาก็จะดีขึ้นด้วยดังนั้นจึงได้ทำการทดสอบสมมุติฐาน โดยเปลี่ยนแปลงค่าจำนวนประชากรเบื้องต้นและกำหนดค่าพารามิเตอร์ตัวอื่นๆ ให้คงที่ คือ

- จำนวนประชากรเท่ากับ 100 โครโมโซม
- จำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน
- ค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8
- ค่าความน่าจะเป็นในมิวเตชันเท่ากับ 0.3

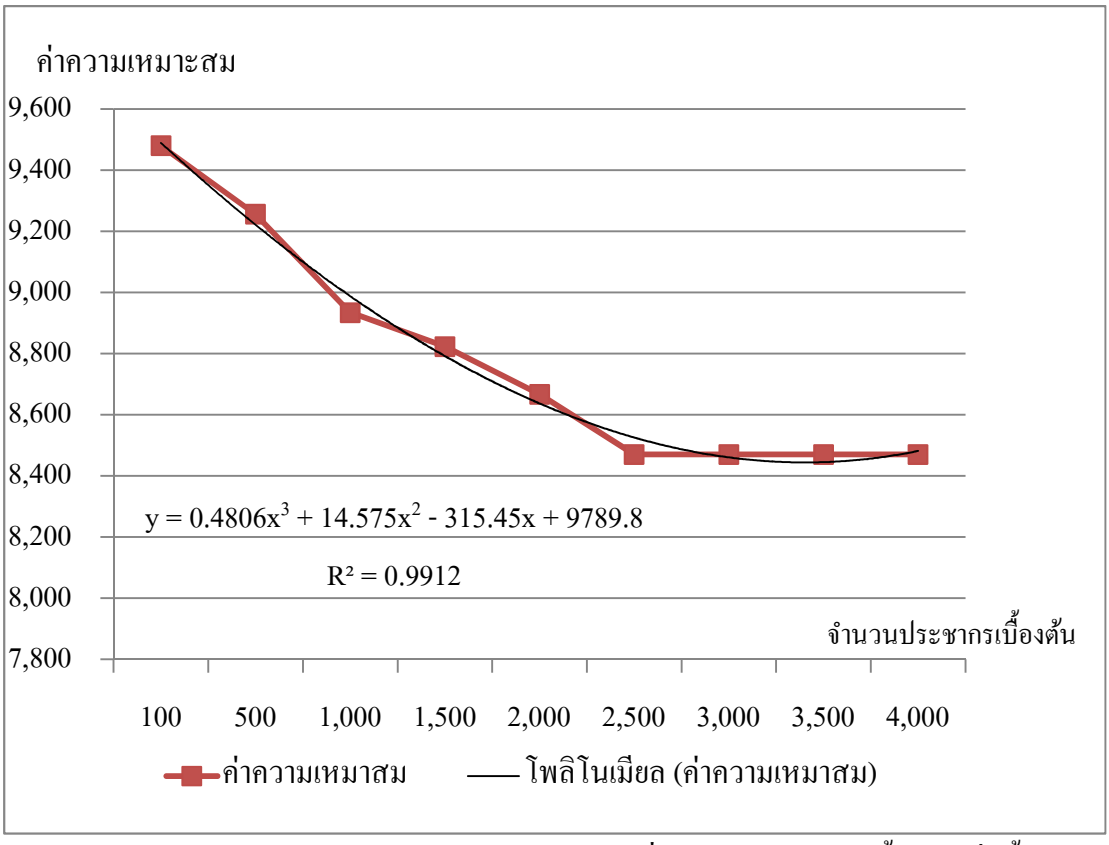
โดยใช้ข้อมูลจำนวนยีนเท่ากับ 3,540 เนื่องจากเป็นชุดข้อมูลจริงของคณะฯ ผลจากการวิเคราะห์จำนวนประชากรเบื้องต้นให้ผลตามตาราง 4.32

ตาราง 4.32 ผลการวิเคราะห์จำนวนประชากรเบื้องต้น

| **จำนวนประชากรเบื้องต้น (โครโมโซม)** | **ค่าความเหมาะสม** |
|---|---|
| 100 | 9,480 |
| 500 | 9,256 |
| 1,000 | 8,934 |
| 1,500 | 8,823 |
| 2,000 | 8,667 |
| 2,500 | 8,470 |
| 3,000 | 8,470 |
| 3,500 | 8,470 |
| 4,000 | 8,470 |

จากตาราง 4.32 พบว่าเมื่อจำนวนประชากรเบื้องต้นเพิ่มมากขึ้น ค่าความเหมาะสมก็จะยิ่งลดลง สรุปได้ว่าจำนวนประชากรเบื้องต้นที่เหมาะสม คือ 2,500 โครโมโซม โดยจำนวนประชากรเบื้องต้นจะมีผลต่อค่าความเหมาะสม คือเมื่อจำนวนประชากรเบื้องต้นมากขึ้นก็จะส่งผลให้ค่าความเหมาะสมที่ลดลงตามไปด้วย โดยกราฟแสดงการลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นที่มากขึ้น แสดงได้ดังภาพประกอบ 4.6

จากภาพประกอบ 4.6 กราฟแสดงการลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นเพิ่มขึ้น ซึ่งมีลักษณะการลดลงของค่าความเหมาะสม สามารถแสดงเป็นสมการ คือ $y = 0.4806x^3 + 14.575x^2 - 315.45x + 9789.8$




ภาพประกอบ 4.6 การลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นเพิ่มขึ้น

จากการวิเคราะห์ความไวหาค่าพารามิเตอร์ที่เหมาะสมที่ได้ทำการวิเคราะห์ดังที่กล่าวมาข้างต้นแล้ว ผู้วิจัยมีสมมุติฐานว่ายังมีค่าพารามิเตอร์อีกตัวหนึ่งที่ผลต่อค่าความเหมาะสมของตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นนั้นคือ ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ ที่จะกล่าวถึงในหัวข้อถัดไป

### 4.4 การวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์

ในการวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่เหมาะสมสำหรับโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนโดยประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบ โดยจะวิเคราะห์การเปลี่ยนแปลงของการหาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ โดยกำหนดช่วงของค่าน้ำหนักเพื่อความสมบูรณ์สำหรับประมวลผล ซึ่งค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ของแต่ละข้อนั้นจะถูกกำหนดตามลำดับความสำคัญของเงื่อนไขข้อนั้นๆ แบ่งการกำหนดค่าหนักของเงื่อนไขเพื่อความสมบูณ์เป็น 6 ช่วง ตามตาราง 4.33 และทำการทดสอบตามค่าพารามิเตอร์ที่เหมาะสมคือ จำนวนประชากรเบื้องต้น 2,500 โครโมโซม จำนวนประชากร 100 โครโมโซม กำหนดค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8 และค่าความน่าจะเป็นในการมิวเตชันเท่ากับ 0.3 และจำนวนเจนเนอเรชันที่วิเคราะห์คือ 100, 200, 300, 400, 500, 600, 700 และ 800 เจนเนอเรชัน

ตาราง 4.33 ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง

| ข้อ | เงื่อนไขเพื่อความสมบูรณ์ | ค่าน้ำหนัก | | | | | |
|---|---|---|---|---|---|---|---|
| | | ช่วงที่ 1 | ช่วงที่ 2 | ช่วงที่ 3 | ช่วงที่ 4 | ช่วงที่ 5 | ช่วงที่ 6 |
| 1 | ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 40 | 100 | 200 | 300 | 400 | 500 |
| 2 | ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ | 40 | 100 | 200 | 300 | 400 | 500 |
| 3 | ในแต่ละวันไม่ควรจัดการเรียนการสอนในเวลา 12.00-12.50 น. | 30 | 75 | 150 | 225 | 300 | 375 |
| 4 | รายวิชาประจำภาควิชาควรจัดให้อยู่ในห้องเรียนของภาควิชานั้นก่อน | 20 | 50 | 100 | 150 | 200 | 250 |
| 5 | ช่วงเวลาที่สองของรายวิชาไม่ควรอยู่ในวันเดียวกันโดยเว้นคาบว่าง | 10 | 25 | 50 | 75 | 100 | 125 |
| 6 | ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | 10 | 25 | 50 | 75 | 100 | 125 |

ผลจากการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่กำหนดเป็นดังนี้

### 4.4.1 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 1

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 1 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.34

ตาราง 4.34 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 1

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| ข้อที่ 1 | infeasible | 58 | 45 | 32 | 24 | 13 | 13 | 13 | 13 |
| ข้อที่ 2 | infeasible | 65 | 48 | 36 | 31 | 10 | 10 | 10 | 10 |
| ข้อที่ 3 | infeasible | 134 | 128 | 109 | 105 | 95 | 95 | 95 | 95 |
| ข้อที่ 4 | infeasible | 156 | 134 | 118 | 103 | 112 | 112 | 112 | 112 |
| ข้อที่ 5 | infeasible | 184 | 152 | 138 | 135 | 131 | 131 | 131 | 131 |
| ข้อที่ 6 | infeasible | 175 | 138 | 120 | 119 | 115 | 115 | 115 | 115 |

จากตาราง 4.34 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 1 เมื่อพิจารณาถึงคำตอบที่เป็นไปได้โปรแกรมสามารถหาคำตอบที่เป็นไปได้ ณ เจนเนอเรชันที่ 100 และพบว่าหลังจากที่หาคำตอบที่เป็นไปได้ได้แล้ว จำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจะลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น และลดลงคงที่ ณ เจนเนอเรชันที่ 500 ซึ่งถือว่าเป็นคำตอบที่ดีที่สุด

### 4.4.2 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 2

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 2 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.35 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 2 เมื่อพิจารณาถึงคำตอบที่เป็นไปได้โปรแกรมสามารถหาคำตอบที่เป็นไปได้ ณ เจนเนอเรชันที่ 100 และพบว่าหลังจากที่หาคำตอบที่เป็นไปได้ได้แล้ว จำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจะลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น และลดลงคงที่ ณ เจนเนอเรชันที่ 500 ซึ่งถือว่าเป็นคำตอบที่ดีที่สุด

ตาราง 4.35 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 2

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| ข้อที่ 1 | infeasible | 60 | 42 | 32 | 24 | 13 | 13 | 13 | 13 |
| ข้อที่ 2 | infeasible | 65 | 48 | 35 | 28 | 10 | 10 | 10 | 10 |
| ข้อที่ 3 | infeasible | 138 | 131 | 119 | 115 | 95 | 95 | 95 | 95 |
| ข้อที่ 4 | infeasible | 156 | 134 | 121 | 115 | 112 | 112 | 112 | 112 |
| ข้อที่ 5 | infeasible | 185 | 151 | 135 | 135 | 131 | 131 | 131 | 131 |
| ข้อที่ 6 | infeasible | 174 | 132 | 117 | 120 | 115 | 115 | 115 | 115 |

### 4.4.3 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 3

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 3 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิด

เงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.36

ตาราง 4.36 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 3

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **ข้อที่ 1** | infeasible | infeasible | infeasible | 55 | 42 | 38 | 22 | 13 | 13 |
| **ข้อที่ 2** | infeasible | infeasible | infeasible | 72 | 45 | 35 | 25 | 10 | 10 |
| **ข้อที่ 3** | infeasible | infeasible | infeasible | 135 | 138 | 122 | 115 | 95 | 95 |
| **ข้อที่ 4** | infeasible | infeasible | infeasible | 146 | 134 | 121 | 118 | 112 | 112 |
| **ข้อที่ 5** | infeasible | infeasible | infeasible | 178 | 148 | 132 | 128 | 131 | 131 |
| **ข้อที่ 6** | infeasible | infeasible | infeasible | 164 | 143 | 118 | 122 | 115 | 115 |

จากตาราง 4.36 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 3 เมื่อพิจารณาถึงคำตอบที่เป็นไปได้โปรแกรมสามารถหาคำตอบที่เป็นไปได้ ณ เจนเนอเรชันที่ 300 และพบว่าหลังจากที่หาคำตอบที่เป็นไปได้ได้แล้ว จำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจะลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น และลดลงคงที่ ณ เจนเนอเรชันที่ 700 ซึ่งถือว่าเป็นคำตอบที่ดีที่สุด

### 4.4.4 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 4

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 4 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์

เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.37

ตาราง 4.37 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 4

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **ข้อที่ 1** | infeasible | infeasible | infeasible | infeasible | 62 | 45 | 38 | 28 | 13 |
| **ข้อที่ 2** | infeasible | infeasible | infeasible | infeasible | 69 | 41 | 35 | 25 | 10 |
| **ข้อที่ 3** | infeasible | infeasible | infeasible | infeasible | 138 | 128 | 132 | 109 | 95 |
| **ข้อที่ 4** | infeasible | infeasible | infeasible | infeasible | 152 | 144 | 121 | 121 | 112 |
| **ข้อที่ 5** | infeasible | infeasible | infeasible | infeasible | 168 | 148 | 135 | 127 | 131 |
| **ข้อที่ 6** | infeasible | infeasible | infeasible | infeasible | 154 | 153 | 128 | 118 | 115 |

จากตาราง 4.37 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 4 เมื่อพิจารณาถึงคำตอบที่เป็นไปได้โปรแกรมสามารถหาคำตอบที่เป็นไปได้ ณ เจนเนอเรชันที่ 400 และพบว่าหลังจากที่หาคำตอบที่เป็นไปได้ได้แล้ว จำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจะลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น และลดลงคงที่ ณ เจนเนอเรชันที่ 800 ซึ่งถือว่าเป็นคำตอบที่ดีที่สุด

### 4.4.5 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 5

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 5 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.38

ตาราง 4.38 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 5

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| ข้อที่ 1 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 72 | 42 | 35 |
| ข้อที่ 2 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 69 | 48 | 39 |
| ข้อที่ 3 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 148 | 125 | 134 |
| ข้อที่ 4 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 152 | 142 | 125 |
| ข้อที่ 5 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 165 | 155 | 132 |
| ข้อที่ 6 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | 144 | 148 | 124 |

จากตาราง 4.38 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 5 เมื่อพิจารณาถึงคำตอบที่เป็นไปได้โปรแกรมสามารถหาคำตอบที่เป็นไปได้ ณ เจนเนอเรชันที่ 600 และพบว่าหลังจากที่หาคำตอบที่เป็นไปได้ได้แล้ว จำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อจะลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้น แต่ไม่สามารถหาคำตอบที่ดีที่สุดได้ในช่วงจำนวนเจนเนอเรชันที่กำหนดสำหรับการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในครั้งนี้

### 4.4.6 ผลการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในช่วงที่ 6

การวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 6 โดยพิจารณาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชันเริ่มต้นไปจนถึงเจนเนอเรชันที่ 800 ซึ่งผลการวิเคราะห์แสดงได้ตามตาราง 4.39 ผลการวิเคราะห์ความไวของของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 6 โดยโปรแกรมไม่สามารถหาคำตอบที่เป็นไปได้

และคำตอบที่ดีที่สุดได้ในช่วงจำนวนเจนเนอเรชันที่กำหนดสำหรับการวิเคราะห์ความไวของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในครั้งนี้

ตาราง 4.39 จำนวนครั้งของการละเมิดเงื่อนไขในแต่ละข้อที่เปลี่ยนไป ณ เจนเนอเรชันต่างๆ เมื่อค่าน้ำหนักของเงื่อนไขอยู่ในช่วงที่ 6

| เงื่อนไขเพื่อความสมบูรณ์ | จำนวนครั้งที่เกิดการละเมิดเงื่อนไข | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | เริ่มต้น | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| ข้อที่ 1 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |
| ข้อที่ 2 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |
| ข้อที่ 3 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |
| ข้อที่ 4 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |
| ข้อที่ 5 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |
| ข้อที่ 6 | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible | infeasible |

จากการวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ เมื่อกำหนดช่วงของค่าน้ำหนักเพื่อความสมบูรณ์แต่ละช่วงที่ต่างกัน พบว่าค่าพารามิเตอร์ที่เหมาะสมสำหรับค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง แสดงได้ตามตาราง 4.40

ตาราง 4.40 ผลการวิเคราะห์ความไวของค่าน้ำหนักเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง

| ช่วงค่าน้ำหนัก | เจนเนอเรชันที่พบคำตอบที่เป็นไปได้ | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด |
|---|---|---|
| ช่วงที่ 1 | 100 | 500 |
| ช่วงที่ 2 | 100 | 500 |
| ช่วงที่ 3 | 200 | 700 |
| ช่วงที่ 4 | 300 | 800 |
| ช่วงที่ 5 | 500 | N/A |
| ช่วงที่ 6 | N/A | N/A |

จากตาราง 4.40 ค่าน้ำหนักของเงื่อนไขช่วงที่ 1 และ 2 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 100 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 500 ค่าน้ำหนักของเงื่อนไขช่วงที่ 3 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 200 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 700 ค่าน้ำหนักของเงื่อนไขช่วงที่ 4 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 300 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 800 ค่าน้ำหนักของเงื่อนไขช่วงที่ 5 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 800 แต่ไม่พบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ทำการทดลอง และค่าน้ำหนักของเงื่อนไขช่วงที่ 6 ไม่พบคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดในเจนเนอเรชันที่ทำการทดลอง ทั้งนี้เหตุที่ไม่สามารถทำการทดลองในจำนวนเจนเนอเรชันที่มากกว่าที่กำหนดนั้น เนื่องจากความสามารถทางด้านคอมพิวเตอร์ที่ใช้ในการทดลอง

กล่าวโดยสรุปได้ว่า ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วงนั้นมีผลต่อจำนวนเจนเนอเรชันที่พบคำตอบที่เป็นไปได้และจำนวนเจนเนอเรชันที่พบคำตอบที่ดีที่สุด เนื่องจากในงานวิจัยนี้ได้กำหนดค่าน้ำหนักของเงื่อนไขบังคับเท่ากับ 1,000 ดังนั้นช่วงของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่มีค่าน้อย แตกต่างกับค่าน้ำหนักของเงื่อนไขบังคับมาก ส่งผลให้ช่วงนั้นๆ สามารถหาคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดได้อย่างรวดเร็ว

# บทที่ 5

# บทสรุปและข้อเสนอแนะ

การดำเนินงานวิจัยครั้งนี้มีวัตถุประสงค์เพื่อสร้างโปรแกรมคอมพิวเตอร์สำหรับจัดตารางสอนที่เหมาะสมสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยนำเอาวิธีการของเจเนติกอัลกอริทึมเข้ามาประยุกต์ใช้ในการหาคำตอบของปัญหา โดยพิจารณาคำตอบที่มีค่าความเหมาะสมน้อยที่สุดหรือตารางสอนที่ผิดเงื่อนไขในการจัดตารางสอนน้อยที่สุดเป็นตัวชี้วัดตารางสอนที่ดีที่สุด โปรแกรมประกอบด้วย 3 ส่วนหลัก คือ ส่วนการป้อนข้อมูลเข้า ส่วนประมวลผล และส่วนแสดงผลลัพธ์ ซึ่งมีหลักการทำงานคือ โปรแกรมจะรับข้อมูลป้อนเข้าจากผู้ใช้และดึงรายละเอียดของข้อมูลเหล่านั้นจากฐานข้อมูลมาทำการประมวลผลตามวิธีการของเจเนติกอัลกอริทึมซึ่งประกอบด้วยขั้นตอน การสร้างคำตอบเบื้องต้น การคัดเลือก การครอสโอเวอร์ การมิวเตชัน และการหยุดการค้นหา สุดท้ายแสดงผลลัพธ์เป็นตารางสอน ซึ่งสามารถแสดงออกมาเป็นตารางสอนอาจารย์ ตารางเรียนตามกลุ่มนักศึกษา และตารางการใช้ห้องเรียน ในรูปแบบของโปรแกรม Microsoft Excel ซึ่งผู้ใช้สามารถเรียกดูข้อมูลหรือนำข้อมูลไปใช้งานต่อได้อย่างสะดวก

## 5.1 สรุปผลการดำเนินงานวิจัย

การพัฒนาโปรแกรมคอมพิวเตอร์ช่วยในการตารางสอนโดยประยุกต์ใช้วิธีการของเจเนติกอัลกอริทึมจะทำให้ได้โปรแกรมจัดตารางสอนสำเร็จรูปซึ่งเป็นโปรแกรมต้นแบบที่สามารถนำผลลัพธ์ไปใช้งานได้จริง เนื่องจากผลลัพธ์ที่ได้ไม่เกิดการซ้ำซ้อนกันของเวลาการเรียนการสอนทั้งของอาจารย์และนักศึกษา อีกทั้งตารางการใช้ห้องเรียนก็มีเวลาการใช้ห้องที่ไม่ซ้ำซ้อนกัน และสามารถนำโปรแกรมนี้ไปพัฒนาต่อไปให้เหมาะสมกับแต่ละสถานศึกษาได้ เนื่องจากสามารถปรับเปลี่ยนลำดับความสำคัญของเงื่อนไขในการจัดตารางสอนให้เหมาะสมกับแต่ละสถานศึกษาได้ ผลจากการวิจัยพบว่าโปรแกรมสามารถค้นหาตารางสอนที่ผิดเงื่อนไขในการจัดตารางสอนน้อยที่สุด ซึ่งถือเป็นตารางสอนที่ดีที่สุด

ในการประมวลผลโปรแกรมนั้นการกำหนดค่าพารามิเตอร์เจเนติกอัลกอริทึมที่เหมาะสมกับขนาดของปัญหา (ในที่นี้คือขนาดจำนวนยีน) จะมีผลต่อการประมวลผลโปรแกรมในการค้นหาคำตอบที่เหมาะสมที่สุด โดยพารามิเตอร์เจเนติกอัลกอริทึมที่มีค่าเหมาะสมจะทำให้โปรแกรมประมวลผลได้คำตอบที่ดีที่สุดและใกล้เคียงกับค่าจริงที่สุดสำหรับปัญหานั้นๆ

จากการทดสอบหาค่าพารามิเตอร์ที่เหมาะสมสำหรับงานวิจัยนี้ ได้พิจารณาขนาดจำนวนยีนแบ่งเป็น 7 ช่วง คือ จำนวนยีนน้อยกว่า 500 ยีน จำนวนยีน 500-1,000 ยีน จำนวนยีน 1,000-2,000 ยีน จำนวนยีน 2,000-3,000 ยีน จำนวนยีน 3,000-4,000 ยีน จำนวนยีน 4,000-5,000 ยีน และจำนวนยีนมากกว่า 5,000 ยีน ทำการวิเคราะห์หาค่าพารามิเตอร์ที่เหมาะสมสำหรับขนาดจำนวนยีนต่างๆ ดังกล่าวโดยพารามิเตอร์ที่พิจารณาในงานวิจัยนี้ คือจำนวนประชากรเบื้องต้น และจำนวนเจนเนอเรชัน โดยการประมวลผลโปรแกรมที่พัฒนาขึ้นเพื่อหาค่าความเหมาะสมของตารางสอนที่น้อยที่สุด ณ เจนเนอเรชันต่างๆ ตั้งแต่เจนเนอเรชัน 100, 200, 300, 400, 500, 600, 700 ถึง 800 เมื่อพิจารณาจำนวนประชากรเบื้องต้นเท่ากับ 10 20 50 และ 100 โครโมโซม กำหนดค่าความน่าจะเป็นในการครอสโอเวอร์และค่าความน่าจะเป็นในการมิวเตชันคงที่เท่ากับ 0.8 และ 0.3 ตามลำดับ การประมวลผลโปรแกรมพบว่าค่าความเหมาะสมของตารางสอนที่น้อยที่สุดจะมีค่าลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้นและลดลงต่ำสุดคงที่ค่าหนึ่ง ณ เจนเนอเรชันใดๆ ซึ่งหมายถึงค่าที่เหมาะสมที่สุดสำหรับปัญหานั้นๆ ซึ่งจากผลการการวิเคราะห์สามารถกำหนดค่าพารามิเตอร์ที่เหมาะสมสำหรับงานวิจัยนี้ คือ จำนวนประชากรเบื้องต้น เท่ากับ 2,500 จำนวนประชากร เท่ากับ 100 จำนวนเจนเนอเรชัน เท่ากับ 500 ค่าความน่าจะเป็นในการครอสโอเวอร์ 0.8 ค่าความน่าจะเป็นในการมิวเตชัน 0.3

นอกจากนี้จากการประมวลผลของโปรแกรมยังพบว่า ขนาดจำนวนยีน มีผลต่อเวลาในการประมวลผลโปรแกรมโดยขนาดจำนวนยีนที่เพิ่มขึ้นจะทำให้ใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นแบบเอ็กโปเนนเชียล

ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์มีผลต่อจำนวนเจนเนอเรชันที่พบคำตอบที่เป็นไปได้และจำนวนเจนเนอเรชันที่พบคำตอบที่ดีที่สุด โดยค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่มีค่าน้อย แตกต่างกับค่าน้ำหนักของเงื่อนไขบังคับมาก ส่งผลให้ช่วงนั้นๆ สามารถหาคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดได้อย่างรวดเร็ว

เมื่อทำการเปรียบเทียบผลลัพธ์โปรแกรมหรือตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นกับตารางสอนที่จัดด้วยวิธีการแบบเดิม พบว่าโปรแกรมที่พัฒนาขึ้นให้ผลที่ดีกว่าการจัดตารางสอนด้วยวิธีการแบบเดิม โดยโปรแกรมที่พัฒนาขึ้นสามารถค้นหาตารางสอนที่ไม่ขัดแย้งกับเงื่อนไขบังคับ นั้นหมายความว่าตารางสอนที่ได้ไม่มีเวลาซ้ำซ้อนกัน สามารถนำไปงานได้งานจริง

โดยใช้เวลาในการประมวลผลประมาณ 50 นาที แต่เนื่องจากโปรแกรมที่พัฒนาขึ้นมานั้นใช้ข้อมูลของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ดังนั้นเงื่อนไขในการพิจารณาสำหรับการจัดตารางสอนจึงเป็นเงื่อนไขที่เหมาะสมสำหรับสถานศึกษานี้เท่านั้น อีกทั้งในการจัดตารางสอนแต่ละครั้ง จะไม่ทำการจัดรายวิชานอกคณะ จะทำการนำเวลาที่กองทะเบียนและประมวลผลได้เคยจัดไว้แล้วมากำหนดให้กับกลุ่มนักศึกษาที่มีเรียนรายวิชานอกคณะเท่านั้น ดังนั้นโปรแกรมที่พัฒนาขึ้นจึงเหมาะกับการจัดตารางสอนที่พิจารณาเฉพาะรายวิชาที่เรียนของคณะฯ ทำการเรียนการสอนภายในห้องเรียนของคณะฯ หรือภาควิชาฯ เท่านั้น

สำหรับการจัดตารางสอนที่มีระยะเวลาการเรียนการสอนคาบละ 90 นาที สามารถให้ตารางสอนที่เหมาะสมกว่าตารางสอนที่มีระยะเวลาการเรียนการสอนคาบละ 50 นาที แต่ยังคงไม่เหมาะสมกับลักษณะการเรียนการสอนของคณะฯ ในปัจจุบัน เนื่องจากรายวิชาในคณะฯ ประกอบด้วยรายวิชาที่มีหน่วยกิตแตกต่างกัน ซึ่งการจัดตารางสอนที่มีระยะเวลาการเรียนการสอนคาบละ 90 นาที เหมาะสมกับรายวิชาที่มี 3 หน่วยกิตเท่านั้น

โปรแกรมการจัดตารางสอนโดยประยุกต์ใช้เทคนิคเจเนติกอัลกอริทึมที่พัฒนาขึ้นนี้เป็นโปรแกรมต้นแบบในการนำเทคนิคเจเนติกอัลกอริทึมเข้ามาช่วยในการหาคำตอบซึ่งจากผลการวิจัยพบว่าเจเนติกอัลกอริทึมสามารถช่วยค้นหาตารางสอนที่เป็นไปได้ สามารถนำไปใช้งานได้จริง และสามารถประมวลผลรายวิชาได้สูงสุด 8,991 คาบเรียน ซึ่งเป็นผลที่น่าพอใจ สามารถนำไปประยุกต์ใช้กับกระบวนการจัดตารางสอนในระดับคณะฯ หรือภาควิชาฯได้ หรือสถานศึกษาที่มีปัจจัยและเงื่อนไขในการจัดตารางสอนเช่นเดียวกับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ เพียงแต่ปรับปรุงฐานข้อมูลให้ตรงกับความต้องการของโปรแกรม

## 5.2 ข้อเสนอแนะในการดำเนินงานวิจัย

จากงานวิจัยการประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สามารถสรุปเป็นข้อเสนอแนะแนวทางการปรับปรุงประสิทธิภาพการทำงานของโปรแกรมจัดตารางสอนที่พัฒนาขึ้นเพื่อใช้ประโยชน์ต่อไปได้ ดังต่อไปนี้

1. ปัญหาการจัดตารางสอนเป็นปัญหาที่มีเงื่อนไขและข้อกำหนดอย่างหลากหลาย ซึ่งสถาบันการศึกษาระดับอุดมศึกษาแต่ละสถาบันมีความต้องการในการจัดตารางสอนที่แตกต่างกัน ดังนั้นควรมีการควบคุมเงื่อนไขและปรับเปลี่ยนเงื่อนไขได้ตามความต้องการในแต่ละที่ได้ รวมถึงสามารถเพิ่มเงื่อนไขเพื่อความสมบูรณ์ของการจัดตารางสอน เช่น เงื่อนไขในการจัด

ตารางสอนที่พิจารณาถึงพื้นที่ในการเรียนคือในแต่ละรายวิชาควรใช้อาคารเรียนหรือห้องเรียนที่ไม่ห่างกันมากเกินไป เป็นต้น เนื่องจากขอบเขตของงานวิจัยนี้คือภายในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ จึงไม่ได้นำเงื่อนไขดังกล่าวมาพิจารณา

2. หาวิธีการเข้ารหัสโครโมโซมให้มีขนาดเล็ก หรือนำแนวคิดเรื่องการบีบอัดข้อมูลมาใช้ลดขนาดของโครโมโซมที่อยู่ในหน่วยความจำขนาดเล็ก เพื่อที่จะได้สามารถสร้างจำนวนโครโมโซมได้มากขึ้น และสามารถแก้ปัญหาที่มีขนาดใหญ่ได้ด้วยเครื่องคอมพิวเตอร์ส่วนบุคคลเพียงเครื่องเดียว

3. ปัจจุบันมีเทคนิคต่างๆ ในการหาคำตอบที่ดีที่สุดมีเพิ่มมากขึ้น การศึกษาและนำเอาเทคนิคใหม่ๆ มาปรับใช้กับเทคนิคเจเนติกอัลกอริทึมในการหาคำตอบจะทำให้ประสิทธิภาพในการหาคำตอบเพิ่มมากขึ้น

4. ควรพัฒนาโปรแกรมให้สามารถจัดตารางสอนในระดับปัญหาที่ใหญ่ขึ้น มีเงื่อนไขและปัจจัยที่ต้องพิจารณามากขึ้น เช่น การพัฒนาโปรแกรมการจัดตารางสอนให้สามารถจัดในระดับมหาวิทยาลัย เนื่องจากต้องจัดร่วมกันหลายคณะ

5. ควรพัฒนาโปรแกรมให้มีความยืดหยุ่นในการรับข้อมูลเข้าสู่ฐานข้อมูล เช่น พัฒนาโปรแกรมให้มีประสิทธิภาพในการเชื่อมโยงฐานข้อมูล สามารถป้อนข้อมูลออนไลน์ได้ หรือสามารถรับไฟล์ข้อมูลที่เป็นไฟล์ word หรือ excel เข้าสู่ฐานข้อมูลได้ทันที

6. ควรพัฒนาโปรแกรมให้สามารถสร้างตารางสอนที่มีระยะเวลาต่อคาบเรียนทั้ง 50 นาทีต่อคาบ และ 90 นาทีต่อคาบ เพื่อได้ตารางสอนที่มีความยืดหยุ่นมากยิ่งขึ้น

7. ควรพัฒนาโปรแกรมในขั้นตอนการสร้างคำตอบเบื้องต้นควรสร้างโดยอ้างอิงจากตารางสอนที่เป็นไปได้ของปีการศึกษาที่ผ่านมา เพื่อให้ได้ผลลัพธ์ที่น่าจะดีกว่าวิธีการเดิม และโปรแกรมสามารถประมวลผลหาคำตอบที่ดีที่สุดได้เร็วยิ่งขึ้น

8. ควรพัฒนาโปรแกรมให้สามารถกำหนดคาบเรียนได้ เช่น เมื่อรายวิชาใดสุ่มคาบเรียนได้ในวันจันทร์ เวลา 08.00-08.50 น. แล้ว คาบเรียนต่อไปควรกำหนดให้เรียนในวันพุธ เวลา 08.00-08.50 น. และวันศุกร์ เวลา 08.00-08.50 น. เป็นต้นเพื่อโปรแกรมสามารถประมวลผลหาคำตอบที่ดีที่สุดได้เร็วยิ่งขึ้น

**บรรณานุกรม**


[1] สิริลักษณ์ จุณณทัศน์. "จีเนติกอัลกอริทึมแบบหลายจุดประสงค์สำหรับแก้ปัญหาการจัดตารางสอนของโรงเรียน." วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2547.

[2] D. Abramson. "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms." *Management Science,* vol. 37, pp. 98-113. 1991.

[3] D. Costa. "A tabu search algorithm for computing an operational timetable." *European Journal of Operational Research,* vol. 76, pp. 98-110. 1994.

[4] M. J. F. Souza. "A GRASP-tabu search algorithm to solve a school timetabling problem," presented at the In 4th Metaheuristic international conference, Portugal:Porto, 2001.

[5] K. Socha., M. Sampels., and M. Manfrin. "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," in *Applications of Evolutionary Computing*. vol. 2611, pp. 334-345. 2003.

[6] J. Knowles. and M. Sampels. "A max-min ant system for the university course timetabling problem." *Lecture Notes in Computer Science,* vol. 2463, pp. 1-13. 2002.

[7] E. K. Burke., B. McCollum., A. Meisels., S. Petrovic., and R. Qu. "A graph-based hyper-heuristic for educational timetabling problems." *European Journal of Operational Research,* vol. 176, pp. 177-192. 2007.

[8] M. P. Carrasco. and M. V. Pato. "A multiobjective genetic algorithm for the class/teacher timetabling problem," in *Practice and Theory of Automated Timetabling Iii*. vol. 2079, pp. 3-17. 2001.

[9] ปริศนา แช่มสุขขี. "เอกสารสัมมนาคอมพิวเตอร์ Genetic Algorithm (GA)". ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา. 2547. (สำเนา)

[10] W. Rattanamanee. "Application of the genetic algorithm to design path direction for automated guided vehicle's movement network." *Songklanakarin Journal of Science Technology,* vol. 25, pp. 91-102. 2003.

[11] ปารเมศ ชุติมา และญาณี เฉยใจชื่น, "การจัดตารางการผลิต กรณีศึกษา: โรงงานประกอบโคมไฟฟ้าสำเร็จรูป," การประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ, คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย 2550.

[12] กุสุมา เรืองดิษฐ์, วนิดา รัตนมณี, สมชาย ชูโฉม, และสุริยา จิรสถิตสิน, "การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดลำดับการผลิตเฟอร์นิเจอร์ไม้ยางพารา," การประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น, 2552.

[13] อนุพนธ์ ยอดต่อ, วนิดา รัตนมณี, สมชาย ชูโฉม, และสุริยา จิรสถิตสิน, "การประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบของปัญหาการออกแบบผังโรงงานเฟอร์นิเจอร์ไม้ยางพารา," การประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น, 2552.

[14] E. K. Burke., David Elliman., and R. Weare. "A Genetic Algorithm Based University Timetabling System," in *In Proceedings of the 2nd East-West International Conference on Computer Technologies in Education*, Ukraine : Crime, pp. 35-40. 1994.

[15] M. Dimopoulou. and P. Miliotis. "Implementation of a university course and examination timetabling system," *European Journal of Operational Research,* vol. 130, pp. 202-213. 2001.

[16] W. Erben. and J. Keppler. "A genetic algorithm solving a weekly course-timetabling problem Practice and Theory of Automated Timetabling." vol. 1153, pp. 198-211. 1996.

[17] D. Abramson. and J. Abela. "A Parallel Genetic Algorithm for Solving the School Timetabling Problem." *In 15th Australian Computer Science Conference*, Hobart, pp. 1-11. 1992.

[18] R. A. Caldeira. "School Timetabling using Genetic Search." in *In Practice and theory of automated timetabling*, L.-I. IST, Ed., ed Toronto : University of Toronto, pp. 115-122. 1997.

[19] กิตติ ไพฑูรย์วัฒนกิจ และ กาญจนี วงศ์วิภาพร. "GA สำหรับจัดตารางสอนโรงเรียน." วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2541.

[20] Y.Zen Wang "Using genetic algorithm methods to solve course scheduling problems." *Expert Systems with Applications,* vol. 25, pp. 12-19. 2003.

[21] N. A. Zahra. "Hybrid heuristics for Examination Timetabling problem." *Applied Mathematics and Computation,* vol. 163, pp. 705-733. 2005.

[22] P. Pongcharoen., W. Promtet., P. Yenradee., and C. Hicks. "Stochastic optimisation timetabling tool for university course scheduling." *International Journal of Production Economics,* vol. 112, pp. 903-918. 2008.

[23] A. Wren. "The Practice and Theory of Automated Timetabling : Selectd Papers from the 1st International confrerance on the practise and theory of automated timetabling." Edinburgh: Napier University, 1996.

[24] วีณา พรหมเทศ. "การจัดตารางสอนอัตโนมัติแบบยืดหยุ่นโดยใช้ Genetic Algorithm." วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาวิทยาศาสตร์ประยุกต์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏเลย, เลย, 2550.

[25] ณัฐกฤตา ศักดิ์เรืองฤทธิ์, "ระบบจัดตารางเรียนตารางสอนโดยใช้ขั้นตอนวิธีพันธุกรรม," วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่, เชียงใหม่, 2550.

[26] ณวัฒน์ นันทะเสน, "การประยุกต์ใช้จีเนติกอัลกอริทึมในการจัดการกับปัญหาการจัดตารางเรียนระดับอุดมศึกษา," วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาสนเทศศาสตร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยหอการค้าไทย, กรุงเทพมหานคร, 2550.

[27] M. T. V. Sehraneh Ghaemi. and Ali Aghagolzadeh. "Using a Genetic Algorithm Optimizer Tool to Solve University Timetable Scheduling Problem." *IEEE,* 2007.

[28] A. Monfroglio. "Hybrid genetic algorithms for timetabling." *International Journal of Intelligent Systems,* vol. 11, pp. 477-523. 1996.

[29] A. Abbas. and E. P. K. Tsang. "Software engineering aspects of constraint-based timetabling - a case study." *Information and Software Technology,* vol. 46, pp. 359-372. 2004.

[30] P. F. Spyros Kazarlis. "Solving University Timetabling Problems Using Advanced Genetic Algorithms." in *In 5th International Conference on Technology and Automation*, 2005.

[31] ระพีพันธ์ ปิตาคะโส. "วิธีการเมตาฮิวริสติก *Metaheuristic* เพื่อแก้ไขปัญหาการวางแผนการผลิตและการจัดการโลจิสติกส์." สำนักพิมพ์ ส.ส.ท., 2554.

[32] สุธน ไชยสุวรรณ และวนิดา ตันติธรรมภูษิต. "คู่มือปฏิบัติงานการจัดตารางเรียน ตารางสอนและตารางสอบ." กองทะเบียนและประมวลผล กองบริการการศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, 2550.

# การหาค่าพารามิเตอร์ทางเจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

# Finding Parameter of Genetic Algorithm-based Timetable for Undergraduate Engineering Student, PSU

พรไพลิน อามีน [1*] และวนิดา รัตนมณี [2]

**บทคัดย่อ**

การจัดตารางสอนเป็นกระบวนที่ค่อนข้างยุ่งยากซับซ้อน มีข้อมูลและเงื่อนไขที่ต้องพิจารณาเป็นจำนวนมาก ดังนั้นในการจัดตารางสอนแต่ละครั้งต้องใช้เวลานานและเกิดความผิดพลาดอยู่เสมอ ในงานวิจัยนี้ได้นำเอาวิธีการของเจเนติกอัลกอริทึมไปใช้ในการพัฒนาโปรแกรมช่วยในการจัดตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ มีวัตถุประสงค์เพื่อให้ได้ตารางสอนที่มีความถูกต้อง ขัดแย้งกับเงื่อนไขน้อยที่สุด (minimize fitness value) และลดเวลาที่ใช้การจัดตารางสอน โดยวิธีการทางเจเนติกอัลกอริทึมที่นำมาประยุกต์ใช้มีขั้นตอนเริ่มจากการสร้างคำตอบที่เป็นไปได้ในรูปแบบโครโมโซม ซึ่งพิจารณาจากความสัมพันธ์ของตัวแปรต่างๆที่ใช้ในการจัดตารางสอน ได้แก่ กลุ่มผู้เรียน อาจารย์ผู้สอน วิชาเรียน ห้องเรียน ภายใต้เงื่อนไขต่างๆในการจัดตารางสอน แล้วประเมินค่าความเหมาะสมของโครโมโซม จากนั้นทำการสลับสายพันธุ์ (Crossover) และทำการกลายพันธุ์ (Mutation) จากผลการทดลองสำหรับข้อมูลที่ประกอบด้วยรายวิชา 425 รายวิชา อาจารย์ 152 คน กลุ่มนักศึกษา 83 กลุ่มและห้องเรียน 59 ห้อง และได้ทำการวิเคราะห์หาค่าพารามิเตอร์ที่เหมาะสมเพื่อกำหนดให้เป็นค่าเริ่มต้นของโปรแกรม ค่าพารามิเตอร์ที่เหมาะสมคือจำนวนคำตอบเบื้องต้นเท่ากับ 2,500 โครโมโซม จำนวนประชากรเท่ากับ 100 โครโมโซมและจำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8 ค่าความน่าจะเป็นในการมิวเตชันเท่ากับ 0.3 นอกจากนี้การกำหนดค่าน้ำหนักของเงื่อนไขบังคับและเงื่อนไขเพื่อความสมบูรณ์จะมีผลต่อจำนวนเจนเนอชันที่สามารถหาคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดอีกด้วย

**คำสำคัญ**: การจัดตารางสอน เจเนติกอัลกอริทึม ฮิวริสติก

**Abstract**

Timetabling procedures are very complex and difficult because they have a number of data and constraints to be considered and determined. It uses a plenty of time, and always has mistaken at the same time. In this report, genetic algorithms has been selected to create a program which can help to generate a timetable for students in the faculty of Engineering, Prince of Songkla University. The purposes of this program are providing the accurate timetable, having minimize conflicts with the conditions which has minimize fitness value, and reducing the time. The first step of genetic algorithm for timetabling is a creation of timetable chromosome patterns which are produced by considering a relation between related variables used in timetabling such as students, lecturers, subjects and classrooms. These chromosomes must be not conflict with the hard constraints in timetabling. Then these chromosomes are evaluated by fitness function, crossover and mutation for automatic adjustment to achieve the better timetable For data used in test, there are 425 subjects, 152 lectures, 83 student groups, and 59 classrooms. And analyzed to determine the appropriate parameters to the default values of the program. Moreover, the optimal values of GA parameters were 2,500 initializes, 100 population, 500 generations probability of crossover was 0.8 and probability of mutation was 0.3. In addition, the weight of hard and soft constraints has effect of the generation that is going to be feasible timetable and the best timetable.

**Keyword :** Course Timetabling, Genetic algorithm, Heuristics

[1*] ผู้นิพนธ์ประสานงาน โทรศัพท์ +66(81)6086776 E-mail: a.pornpailin@gmail.com

[2] รองศาสตราจารย์ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา

## 1. บทนำ

การจัดตารางสอนเป็นปัจจัยที่มีความสำคัญอย่างหนึ่งในการจัดการเรียนการสอนให้สามารถดำเนินไปด้วยความราบรื่น และส่งผลต่อประสิทธิภาพการเรียนของนักศึกษา [1] เป็นกระบวนการนำรายวิชาที่เปิดสอนมากำหนดในตารางเวลาให้มีความสัมพันธ์กับทรัพยากรที่มีอยู่ เช่น อาจารย์ ห้องเรียน อุปกรณ์การเรียนการสอน เป็นต้น [2] ซึ่งเป็นกระบวนการที่มีความยุ่งยากซับซ้อน เนื่องจากมีข้อมูลและเงื่อนไขต่างๆ ที่ต้องพิจารณามากมาย ทำให้การจัดตารางสอนในแต่ละภาคการศึกษาต้องใช้เวลามาก แต่การจัดตารางสอนในปัจจุบันยังเป็นการจัดตารางสอนโดยเจ้าหน้าที่เท่านั้น ไม่มีโปรแกรมคอมพิวเตอร์ใดเข้ามาช่วยในการจัดตารางสอน ทำให้ต้องใช้เวลานาน ต่อมาได้นำโปรแกรมคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูลของตารางสอน เพื่อลดขั้นตอนในการปฏิบัติงาน แต่ระบบโปรแกรมคอมพิวเตอร์ไม่สามารถช่วยในการปฏิบัติงานได้ทุกขั้นตอน โดยเฉพาะขั้นตอนในการตรวจสอบตารางสอน ดังนั้นตารางสอนที่ได้ยังคงมีข้อผิดพลาดอยู่ เช่น

1. มีการซ้ำซ้อนของวัน-เวลาเรียนของนักศึกษาที่ลงทะเบียนตามแผนการศึกษา
2. มีการซ้ำซ้อนของวัน-เวลาสอนของอาจารย์ผู้สอน
3. มีการซ้ำซ้อนของวัน-เวลาในการใช้ห้องเรียน

ข้อผิดพลาดเหล่านี้จะทำให้ตารางสอนที่ได้ไม่สามารถใช้ได้จริง อีกทั้งหากต้องการเปลี่ยนแปลงตารางสอนในแต่ละครั้งเป็นงานที่ทำได้ยุ่งยาก ไม่สามารถปรับเปลี่ยนตารางสอนได้อย่างทันดี

จากที่มาของปัญหาดังกล่าว งานวิจัยนี้จึงได้มีแนวความคิดที่จะมีการนำวิธีการทางเจเนติกอัลกอริทึมมาช่วยในการจัดตารางสอน เพื่อให้การจัดตารางสอนมีความเหมาะสม ถูกต้อง ลดเวลาที่ใช้การจัดตารางสอน เนื่องจากวิธีการทางเจเนติกอัลกอริทึมเป็นการนำกระบวนการทางพันธุ์ศาสตร์มาประยุกต์ใช้ในกระบวนการค้นหาคำตอบของปัญหา วิธีการทางเจเนติกอัลกอริทึมสามารถช่วยแก้ปัญหาที่มีขนาดค่อนข้างใหญ่และซับซ้อนได้ เนื่องจากมีคุณสมบัติของการเลียนแบบการถ่ายทอดลักษณะทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้พิจารณาในการหาคำตอบของประชากรรุ่นถัดไป เป็นกระบวนการที่จะทำให้ได้คำตอบที่ดีที่สุดจากกลุ่มคำตอบที่สุ่มมา คำตอบที่ได้จึงเป็นคำตอบที่ดีเหมาะสมสุดแล้ว อีกทั้งเจเนติกอัลกอริทึมยังเป็นวิธีการที่ใหม่ โดยจะนำเอาวิธีการทางเจเนติกอัลกอริทึมไปใช้ในการสร้างโปรแกรมจัดตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยมีผลลัพธ์ (Output) ที่คือ ตารางเรียนของกลุ่มนักศึกษา ตารางสอนตามอาจารย์ผู้สอน ตารางการใช้ห้องเรียน และทำการหาค่าพารามิเตอร์ทางเจเนติกอัลกอริทึมที่เหมาะสำหรับเป็นค่าเริ่มต้นในประมวลผลโปรแกรมการจัดตารางสอน

## 2. งานวิจัยที่เกี่ยวข้อง

ในการจัดการกับปัญหาการจัดตารางสอนได้มีผู้ประยุกต์ใช้ทฤษฎีทางวิศวกรรมคอมพิวเตอร์มาช่วยในการแก้ปัญหา โดยนักวิจัยได้มีการนำวิธีฮิวริสติกต่างๆ เข้ามาช่วยแก้ปัญหาการจัดตารางสอน เช่น วิธีการจำลองแอลเนลลิ่ง โดย D. Abramson [3] ได้นำวิธีการนี้เข้ามาแก้ไขปัญหาการจัดตารางสอนในโรงเรียนระดับมัธยม โดยมีเป้าหมายเพื่อให้ได้ตารางที่มีค่าฟังก์ชันต้นทุนน้อยที่สุด และได้ทดสอบกับขั้นตอนแบบอนุกรม (serial) พบว่าค่าเฉลี่ยเริ่มต้นของฟังก์ชันต้นทุนจะสูงตามลำดับความซับซ้อนของปัญหา ค่าฟังก์ชันต้นทุนที่น้อยที่สุดได้มาจากการกำหนดอัตราการลดค่าตัวแปรอย่างช้าๆ และพบว่าใช้เวลาในการประมวลนานมาก จึงได้เสนอแนะขั้นตอนวิธีการแบบขนาน (parallel) เพื่อช่วยลดเวลาในการประมวลผล วิธีการค้นหาแบบทาบูโดย D. Costa [4] ได้นำวิธีการนี้มาใช้ในการแก้ไขปัญหาการจัดตารางสอนในโรงเรียนมัธยมศึกษา พบว่าได้ผลดีใกล้เคียงกับการจัดตารางสอนด้วยมือ แต่ใช้เวลาในการทำงานน้อยกว่า แต่

ก็ไม่สามารถจัดตารางสอนให้ได้ผลตามข้อบังคับได้ทั้งหมด ดังนั้น M. J. F. Souza [5] จึงได้ปรับปรุงการจัดตารางสอนด้วยวิธีการค้นหาแบบทาบูโดยในขั้นตอนของการสร้างคำตอบเบื้องต้นได้ใช้ greedy procedure จากนั้นจะปรับปรุงคุณภาพคำตอบด้วยวิธีการค้นหาแบบทาบูโดยต้องการให้ได้ตารางสอนที่มีค่าฟังก์ชันเป้าหมายน้อยที่สุด แต่อย่างไรก็ตาม M. J. F. Souza ก็ได้ทดลองนำ Intraclasses - Interclasses procedure เข้ามาช่วยปรับปรุงหลังจากที่วิธีการค้นหาแบบทาบูให้ตารางที่ดีแล้ว พบว่าการใช้วิธีการค้นหาแบบทาบูร่วมกับ Intraclasses - Interclasses procedure ให้ตารางสอนที่มีคุณภาพดีขึ้นและรวดเร็วยิ่งขึ้น วิธีการค้นหาแบบระบบอาณาจักรมด (Ant Colony System) Socha [6] ได้ใช้ MAX-MIN Ant System ซึ่งเป็นการปรับปรุงจากวิธีการค้นหาแบบระบบอาณาจักรมดที่ J. Knowles and M. Sampels [7] ได้ใช้ในการแก้ไขปัญหาการจัดตารางสอน โดยเสนอ construction graph และรูปแบบ Pheromone ที่เหมาะสม ดังนั้น Burke, E. K. et al. [8] จึงได้นำวิธี Graph Based Hyper Heuristic มาแก้ไขปัญหาตารางสอนและตารางสอบ พบว่าวิธีนี้สามารถแก้ปัญหาได้เป็นอย่างดี โดยได้เสนอให้ปรับปรุงวิธีการนี้เพื่อให้สามารถแก้ปัญหาขนาดใหญ่ ที่มีความซับซ้อนต่อไป ดังนั้นวิธีการของเจเนติกอัลกอริทึมจึงถูกนำเข้ามาใช้ ซึ่งเป็นวิธีเหมาะสมกับปัญหาที่มีขนาดใหญ่มีความยุ่งยากซับซ้อน [9] เนื่องจากเป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติซึ่งคิดค้นโดย John Holland เมื่อปี ค.ศ. 1970 มีคุณสมบัติในการเลียนแบบการถ่ายทอดลักษณะทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้พิจารณาในการหาคำตอบของประชากรรุ่นถัดมาเพื่อให้ได้คำตอบที่เหมาะสมที่สุด [10]

จากงานวิจัยที่ผ่านมาวิธีการของเจเนติกอัลกอริทึมได้ถูกเริ่มต้นนำมาแก้ไขปัญหาการจัดตารางสอบ โดยสร้างคำตอบเบื้องต้นให้เป็นคำตอบที่เป็นไปได้ทุกคำตอบและเสนอการครอสโอเวอร์และมิวเตชันเพื่อให้ตารางสอบที่ได้ยังเป็นคำตอบที่เป็นไปได้ แต่ไม่ทำให้ค่าความเหมาะสมลดลงและยังไม่ต้องใช้กระบวนการซ่อมแซม [11] และยังได้มีการออกแบบและประยุกต์ระบบคอมพิวเตอร์ เพื่อใช้ในการจัดตารางสอบและตารางสอน ซึ่งจะเน้นทำให้การจัดตารางมีความยืดหยุ่น [12] แต่อย่างไรก็ตามการจัดตารางสอบก็ยังคงมีปัญหาและความยากในการแก้ไขปัญหาที่แตกต่างกับการจัดตารางสอน เนื่องจากการจัดตารางสอบขึ้นเป็นครั้งคราว แต่การจัดตารางสอนจะต้องจัดเป็นรายสัปดาห์ จึงทำให้มีสิ่งที่ต้องพิจารณามากกว่าการจัดตารางสอบ จึงได้มีการนำวิธีการของเจเนติกอัลกอริทึมในใช้ในการแก้ไขปัญหาการจัดตารางสอน โดยการสร้างคำตอบยังคงอาศัยวิธีการสร้างคำตอบเบื้องต้นให้เป็นไปได้ทั้งหมด และยังคงมีการครอสโอเวอร์และมิวเตชัน แต่เนื่องจากปัญหาที่ต่างกันจึงทำให้ตารางสอนที่ผ่านกระบวนการเหล่านี้กลายเป็นตารางสอนที่เป็นไปไม่ได้ไม่สามารถนำไปใช้งานได้จริง อีกทั้งยังไม่มีกระบวนการในการซ่อมแซมคำตอบจึงทำให้ใช้เวลานานมากในการหาคำตอบที่เหมาะสม ดังนั้น W. Erben and J. Keppler [13] จึงได้เสนอแนะว่าควรมีการทดลองโดยขั้นตอนวิธีแบบขนาน (parallel algorithm) [14] และปรับปรุงวิธีการของเจเนติกอัลกอริทึมตลอดจนควรศึกษาการใช้ local search หรือการเก็บคำตอบที่ดี (elitism) และได้มีการคิดค้นกระบวนการซ่อมแซมคำตอบเพื่อแก้ไขตารางสอนที่เป็นไปไม่ได้ให้กลับมาเป็นตารางที่เป็นไปได้ แต่อย่างไรก็ตามก็ยังคงเป็นการแก้ไขปัญหาการจัดตารางสอนที่มีขนาดเล็กเป็นการจัดตารางสอนในระดับโรงเรียนมัธยมเท่านั้น ซึ่งมีจำนวนอาจารย์ผู้สอนไม่มากนัก กลุ่มนักเรียนมีน้อยและมีห้องประจำสำหรับนักเรียนแต่ละกลุ่ม [15-17] จึงได้มีการนำเจเนติกอัลกอริทึมมาประยุกต์ในการจัดตารางสอนในมหาวิทยาลัย โดยที่การปรับปรุงกระบวนครอสโอเวอร์และมิวเตชัน ซึ่งสามารถช่วยลดเวลาที่ใช้ในการจัดตารางสอน โดยข้อมูลที่ใช้ในการทดสอบประกอบด้วย กลุ่มผู้เรียน 10 กลุ่ม วิชาเรียน 90 วิชา อาจารย์ 26 คน ซึ่งถึงแม้ว่าจะเป็นการจัดตารางสอนในระดับมหาวิทยาลัย แต่ยังคงใช้ข้อมูลตัวอย่างขนาดเล็ก

นอกจากนี้ยังมีงานวิจัยที่ศึกษาเปรียบเทียบผลการทำงานของวิธีการต่างๆ เช่น N. A. Zahra [18] ได้

ทำการศึกษาเปรียบเทียบผลการทำงานของวิธีการจำลองแอลเนลลิ่ง วิธีการค้นหาแบบทาบู วิธีการค้นหาแบบระบบอาณาจักรมด และวิธีการของเจเนติกอัลกอริทึมในการแก้ปัญหา โดยคำตอบเบื้องต้นของวิธีการวิธีการจำลองแอลเนลลิ่ง วิธีการค้นหาแบบทาบู และ และวิธีการของเจเนติกอัลกอริทึมจะสร้างจากการสุ่มทั้งหมด แต่วิธีการค้นหาแบบระบบอาณาจักรมดจะมีเกณฑ์ในการสร้างคำตอบ ดังนั้นวิธีการค้นหาแบบระบบอาณาจักรมดจึงมีคำตอบเบื้องต้นดีที่สุด เมื่อประมวลผลจนเสร็จ พบว่าหากพิจารณาตามความสามารถในการหาคำตอบวิธีการค้นหาแบบระบบอาณาจักรมดให้คำตอบที่ดีที่สุด รองมาคือวิธีการของเจเนติกอัลกอริทึม วิธีการค้นหาแบบทาบูและ วิธีการจำลองแอลเนลลิ่งแต่หากพิจารณาตามความสามารถในการลดค่าความฟังก์ชันต้นทุนโดยเปรียบเทียบในช่วงเวลาเดียวกัน พบว่าวิธีการค้นหาแบบทาบูดีที่สุด และ P. Pongcharoen [19] ได้พัฒนาโปรแกรมจัดตารางสอนโดยใช้วิธี Stochastic optimization ซึ่งเป็นวิธีการที่นำวิธีการของเจเนติกอัลกอริทึมรวมกับวิธีการจำลองแอลเนลลิ่ง และได้เปรียบเทียบประสิทธิภาพการจัดตารางสอนโดยใช้วิธีการของเจเนติกอัลกอริทึมรวมกับวิธีการจำลองแอลเนลลิ่ง พบว่าขั้นตอนวิธีทั้งสองแบบสามารถแก้ไขปัญหาการจัดตารางสอนได้เป็นอย่างดี แต่จากการประเมินค่าตารางสอนที่จัดได้จากวิธีการจำลองแอลเนลลิ่ง จะให้ผลลัพธ์ที่ดีกว่าจากวิธีการของเจเนติกอัลกอริทึมเพียงเล็กน้อย ในขณะที่วิธีการของเจเนติกอัลกอริทึมจะประมวลผลได้เร็วกว่าวิธีการจำลองแอลเนลลิ่ง

จากการสำรวจงานวิจัยที่เกี่ยวข้องแสดงให้เห็นว่า วิธีการของเจเนติกอัลกอริทึมเป็นวิธีการหนึ่งที่น่าสนใจ เนื่องจากเป็นวิธีการฮิวริสติกแบบหนึ่งซึ่งพัฒนามาจากวิธีการอื่นๆ โดยมีลักษณะการค้นหาคำตอบอย่างมีเหตุผลโดยเลียนแบบการถ่ายทอดทางพันธุกรรมตามธรรมชาติซึ่งได้รับการยอมรับถึงประสิทธิภาพในการแก้ปัญหาการจัดตารางสอน การประยุกต์ใช้เจเนติกอัลกอริทึมสำหรับการสร้างโปรแกรมเพื่อใช้สำหรับการจัดตารางสอน ส่วนใหญ่ผู้วิจัยจะทดลองเก็บตัวอย่างข้อมูลที่น้อย ตารางสอนที่มีขนาดเล็ก จำนวนโครโมโซมไม่มากนัก จำนวนรุ่นน้อย จึงทำให้ได้ตารางสอนที่ดี ใช้เวลาในการจัดตารางสอนน้อย ผลลัพธ์ที่ได้ก็เพียงแค่ตารางสอนของกลุ่มผู้เรียน และตารางการใช้ห้องเรียนเท่านั้น โปรแกรมที่สร้างขึ้นไม่ได้นำไปใช้ในการปฏิบัติงานจริงของเจ้าหน้าที่จัดตารางสอน อีกทั้งจะต้องใส่ข้อมูลเป็นจำนวนมาก ซึ่งบางข้อมูลอาจจะไม่ต้องใส่ทุกครั้งที่มีการเปลี่ยนแปลง และยังไม่มีการนำเจเนติกอัลกอริทึมใช้ในกรณีที่จัดตารางสอนรายวิชาปฏิบัติ โดยใช้ทรัพยากรร่วมกับการจัดตารางสอนรายวิชาบรรยาย ซึ่งเงื่อนไขและข้อจำกัดของการจัดตารางให้รายวิชาทั้ง 2 ประเภทนั้นต่างกัน อีกทั้งยังไม่เคยมีงานวิจัยใดที่ทำการวิเคราะห์หาค่าจำนวนประชากรเบื้องต้น (initialize) ซึ่งเป็นค่าพารามิเตอร์ของเจเนติกอัลกอริทึมตัวหนึ่งที่น่าจะมีผลต่อการคำตอบที่ดีที่สุดของปัญหา

ดังนั้นงานวิจัยนี้จึงทำการศึกษาการประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอนสำหรับนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งถือเป็นปัญหาที่มีขนาดใหญ่ และมีเงื่อนไขในการผลิตที่ซับซ้อนโดยการพิจารณาตารางสอนที่ดีที่สุดเป็นตารางสอนที่ละเมิดเงื่อนไขน้อยที่สุด มีรายวิชาที่ต้องพิจารณาในการจัดตารางสอน 2 ประเภทคือรายวิชาบรรยายและปฏิบัติ ประกอบกับเป็นปัญหาที่มีขนาดใหญ่ มีการเปิดการเรียนการสอน ทั้งหมด 28 สาขาวิชา ระดับปริญญาตรี 12 สาขาวิชา ระดับปริญญาโท 11 สาขาวิชา ปริญญาโทหลักสูตรพิเศษ 2 สาขาวิชา และระดับปริญญาเอก 8 สาขาวิชา จัดการเรียนการสอนภาคการศึกษาละประมาณ 350 รายวิชา 700 ตอน จำนวนนักศึกษาประมาณ 2800 คน มีอาจารย์สอนประมาณ 150 คน และมีห้องเรียนประมาณ 41 ห้องเรียน และทำการวิเคราะห์หาค่าพารามิเตอร์ของเจเนติกอัลกอริทึม ทั้งจำนวนประชากร จำนวนเจนเนอเรชันและจำนวนประชากรเบื้องต้น ซึ่งในส่วนของการหาค่าจำนวนประชากรเบื้องต้นนี้ไม่พบงานวิจัยใดที่เคยทำการทดสอบไว้ อีกทั้งเพื่อแก้ไขข้อผิดพลาดของตารางสอนที่จัดโดยเจ้าหน้าที่ เช่น เวลาของอาจารย์ซ้ำซ้อนกัน เวลา

ของนักศึกษาซ้ำซ้อนกัน และเวลาการใช้ห้องเรียนซ้ำซ้อนกัน โดยมีสมมุติฐานสาเหตุของข้อผิดพลาดดังนี้

1. เวลาของอาจารย์ซ้ำซ้อนกันเกิดจากในบางรายวิชามีอาจารย์สอนร่วมกันหลายคน เช่น วิชาปฏิบัติ
2. เวลาของนักศึกษาซ้ำซ้อนกันเกิดจากการกำหนดกลุ่มนักศึกษาที่ซ้ำซ้อนกัน เช่น กลุ่ม 2EE แบ่งออกเป็น 2EE(A) และ 2EE(B)
3. มีการกำหนดภาระงานสอนของอาจารย์มากจนเกินไปโดยเฉพาะในรายวิชาปฏิบัติที่แบ่งเป็นจำนวนตอนย่อยในการลงปฏิบัติการของนักศึกษาในแต่ละสัปดาห์

จากสมมุติฐานดังกล่าวข้างต้นนั้นสามารถแก้ไขได้โดยการตรวจสอบข้อมูลที่ใช้ในการจัดตารางสอนก่อนโดยละเอียดเพื่อให้ตารางสอนที่ได้จากการพัฒนาโปรแกรมเป็นตารางสอนที่ดี สามารถนำไปใช้งานได้จริง

## 3. การกำหนดเงื่อนไขในการจัดตารางสอน

ในงานวิจัยนี้ได้นำเสนอเงื่อนไขที่ใช้ในการจัดตารางสอน 2 ประเภทคือ เงื่อนไขหรือข้อจำกัดบังคับ (hard constraints) คือเงื่อนไขที่ต้องไม่เกิดขึ้นในการจัดตารางสอน และเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ (soft constraints) คือเงื่อนไขที่ยอมให้เกิดขึ้นได้ในการจัดตารางสอนแต่ต้องเกิดขึ้นน้อยที่สุดจึงจะให้ตารางสอนที่ตรงกับความต้องการมากที่สุด ตารางสอนที่ไม่ขัดแย้งกับเงื่อนไขบังคับเท่านั้นที่จะเป็นตารางสอนที่เป็นไปได้ เงื่อนไขทั้ง 2 ชนิดมีดังนี้

### เงื่อนไขหรือข้อจำกัดบังคับ

1. ในวันและเวลาเดียวกันอาจารย์ผู้สอนหนึ่งคนสามารถสอนได้หนึ่งกลุ่มเท่านั้น
2. ในวันและเวลาเดียวกันนักศึกษาหนึ่งกลุ่มสามารถเรียนได้หนึ่งรายวิชาเท่านั้น
3. ในวันและเวลาเดียวกันกำหนดการเรียนการสอนให้กับห้องเรียน 1 ห้องเรียนได้หนึ่งรายวิชาเท่านั้น
4. กำหนดห้องเรียนให้เหมาะสมกับประเภทรายวิชา เช่น วิชาปฏิบัติจะต้องให้กับห้องปฏิบัติการ

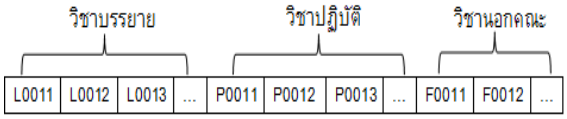
### เงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์

1. คาบที่ 2 ของรายวิชาเดียวกันไม่ควรจัดให้อยู่ในวันเดียวกันโดยเว้นคาบว่าง เช่น วิชาสถิติวิศวกรรม 1 จัดในวันจันทร์ เวลา 08.00-08.50 น. ไม่ควรที่จะจัดในวันจันทร์ เวลา 13.00-13.50 น. อีกครั้ง เป็นต้น หากต้องการให้มีการเรียนการสอนในรายวิชานี้ 2 คาบใน 1 วัน ควรจัดให้เป็น 2 คาบติดต่อ เช่น จัดในวันจันทร์ เวลา 08.00-09.50 น.เป็นต้น
2. ควรจัดการสอนรายวิชาประจำภาควิชาให้กับห้องเรียนในภาควิชานั้นก่อน เช่น วิชาสถิติวิศวกรรมเป็นรายวิชาประจำภาควิชาวิศวกรรมอุตสาหการ ก็ควรจัดให้กับห้องเรียนในภาควิชาวิศวกรรมอุตสาหการก่อน
3. ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป เช่นหากในวันนั้นนักศึกษามีเรียนในเวลา08.00-08.50 น. แล้วไม่ควรมีคาบว่างแล้วมีเรียนต่อในเวลา 11.00-11.50 น. เป็นต้น
4. ในแต่ละวันควรมีการเว้นคาบว่างในเวลา 12.00-12.50 น. เพื่อให้นักศึกษาพักทานอาหารกลางวัน
5. ในแต่ละวันอาจารย์ไม่ควรสอนวิชาบรรยายติดกันเกิน 4 คาบ
6. ในแต่ละวันนักศึกษาไม่ควรเรียนวิชาบรรยายติดต่อกันเกิน 4 คาบ

## 4. การประยุกต์ใช้เจเนติกอัลกอริทึมในการจัดตารางสอน

### 4.1 กำหนดรูปแบบโครโมโซม

การกำหนดรูปแบบโครโมโซม (Chromosome) คือการแปลงคำตอบของปัญหาให้อยู่ในรูปของรหัสโครโมโซม โดยโครโมโซม 1 โครโมโซมแทนคำตอบของปัญหา 1 คำตอบนั้นคือ โครโมโซม 1 โครโมโซมจะแทนตารางสอนที่เป็นไปได้ของคณะวิศวกรรมศาสตร์ 1 ตาราง ใน 1 โครโมโซมประกอบด้วยโครโมโซมย่อย (sub-chromosome) 3 โครโมโซมย่อย คือ โครโมโซมย่อยรายวิชาบรรยาย โครโมโซมย่อยรายวิชาปฏิบัติ และโครโมโซมย่อยรายวิชานอกคณะซึ่งมีการแบ่งช่วงของโครโมโซมย่อยดังแสดงในรูปที่ 1




รูปที่ 1 ตัวอย่างโครโมโซมฃ

ค่าของยีน (gene) ในแต่ละยีนจะประกอบด้วยตัวอักษรและตัวเลข ตัวอักษรแต่ละตัวมีความหมายดังนี้

-L หมายถึง วิชาบรรยาย

-P หมายถึง วิชาปฏิบัติ

-F หมายถึง วิชานอกคณะ

-E หมายถึง ไม่มีการจัดการเรียนการสอน

หลังตัวอักษรดังกล่าวจะตามด้วยตัวเลข 4 ตำแหน่ง โดยที่ 3 ตำแหน่งแรกแทนรายวิชา และตำแหน่งสุดท้ายแทนชั่วโมงที่ทำการเรียนการสอนของแต่ละรายวิชา เช่น 0011 แทนรายวิชาที่ 1 ชั่วโมงที่ 1, 0012 แทนรายวิชาที่ 1 ชั่วโมงที่ 2 และ 0013 แทนรายวิชาที่ 1 ชั่วโมงที่ 3 เป็นต้น

ตำแหน่งของยีน จะบ่งบอกถึงวัน-เวลาและห้องเรียนที่ใช้ในการจัดตารางสอน การจำลองโครโมโซมจะเริ่มจากการสร้างความสัมพันธ์ระหว่างวัน (แนวตั้ง) ที่ทำการจัดตารางสอน 5วัน วันจันทร์-วันศุกร์ และเวลา (แนวนอน) ที่จะทำการจัดตารางสอน ในแต่ละวันมีคาบเรียนทั้งหมด 8 คาบตั้งแต่ 08.00-15.50 น. คาบละ 50 นาที แล้วให้ลำดับเป็นเลขจำนวนนับ ดังแสดงในรูปที่ 2

| | 8.00 - 8.50 | 9.00 - 9.50 | 10.00 - 10.50 | 11.00 - 11.50 | 12.00 - 12.50 | 13.00 - 13.50 | 14.00 - 14.50 | 15.00 - 15.50 |
|---|---|---|---|---|---|---|---|---|
| จ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| อ | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| พ | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| พฤ | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| ศ | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |

รูปที่ 2 ตัวอย่างวัน-เวลาที่ใช้ในการจัดตารางสอน

จากนั้นจะนำลำดับที่สร้างจากวัน-เวลาที่ใช้ในการจัดตารางสอน(แนวนอน) มาสร้างความสัมพันธ์กับห้องเรียนที่ใช้ในการจัดตารางสอน (แนวตั้ง) ยกตัวอย่างมา 3 ห้องเรียน ดังแสดงในรูปที่ 3

| | คาบเรียน 1 | 2 | 3 | 4 | 5 | 6 | … | 40 |
|---|---|---|---|---|---|---|---|---|
| A200 | 001 | 002 | 003 | 004 | 005 | 006 | … | 040 |
| IEF309 | 041 | 042 | 043 | 044 | 045 | 046 | … | 080 |
| IELAB | 081 | 082 | 083 | 084 | 085 | 086 | … | 120 |

ห้องเรียน

รูปที่ 3 ตัวอย่างแบบจำลองโครโมโซม

จะได้โครโมโซมที่มีความยาว 120 ยีน สามารถให้ความหมายยีนแต่ละตำแหน่งได้ดังนี้ เช่น ยีนตำแหน่งที่ 1 หมายถึง วันจันทร์ เวลา 08.00-08.50 น. ห้อง A200 เป็นต้น สามารถหาความยาวโครโมโซมแต่ละโครโมโซมได้ดังสมการที่ 1

ความยาวโครโมโซม =

จำนวนห้องเรียน × จำนวนวัน × จำนวนคาบเรียน (1)

หมายเหตุ จำนวนคาบเรียนในแต่ละวันต้องเท่ากันเสมอ

### 4.2 การประเมินค่าความเหมาะสม

เมื่อได้จำนวนโครโมโซมตามจำนวนประชากรเบื้องต้นที่กำหนดแล้ว จะทำการประเมินค่าความเหมาะสมของแต่ละโครโมโซมโดยฟังก์ชันความเหมาะสม โดยฟังก์ชันความเหมาะสมเป็นฟังก์ชันสร้างจากเงื่อนไขที่ใช้ในการจัดตารางสอนทั้ง 2 ประเภท ที่ได้กำหนดไว้จะมีนำการกำหนดค่าน้ำหนักของเงื่อนไขแต่ละข้อเข้ามาช่วยในการจำแนกโครโมโซมที่ดี โดยการกำหนดค่าน้ำหนักของเงื่อนไขบังคับสูงๆ ให้มีความแตกต่างกับค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์มากๆ จะทำให้โครโมโซมที่ไม่ผ่านเงื่อนไขบังคับมีค่าความเหมาะสมสูงมาก ซึ่งบ่งบอกว่าเป็นโครโมโซมที่ไม่ดี และไม่ถูกเลือกไปเป็นคำตอบของปัญหา โดยค่าน้ำหนักของเงื่อนไขที่ใช้ในงานวิจัยนี้กำหนดโดยสำรวจความคิดเห็นจากอาจารย์ในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยให้อาจารย์กำหนดความสำคัญของเงื่อนไขแต่ละข้อ และให้ค่าน้ำหนักของเงื่อนไขตามลำดับความสำคัญแต่ละข้อ

การวัดค่าโครโมโซมตารางสอนมีเกณฑ์การวัดค่าความเหมาะสมจากฟังก์ชันความเหมาะสมโดยโครโมโซมที่มีค่าความเหมาะสมน้อยที่สุดจะเป็นโครโมโซมที่ดีที่สุด ซึ่งฟังก์ชันค่าความเหมาะสมเป็นไปดังสมการที่ 2

$$\text{fitness value} = min \sum_{i=1}^{N}(n_i * w_i) \qquad (2)$$

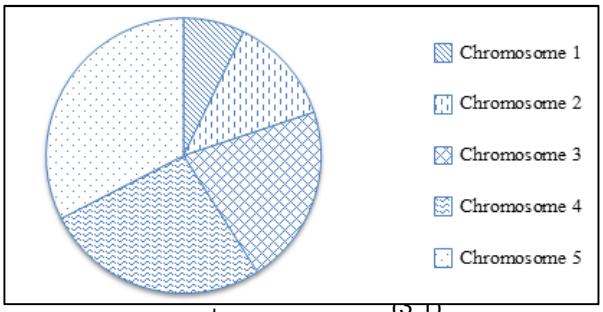
n = จำนวนครั้งที่เกิดการละเมิดเงื่อนไขข้อที่ i
N = จำนวนเงื่อนไขทั้งหมด (รวมทั้ง 2 ประเภท)
W = ค่าน้ำหนักของเงื่อนไขข้อที่ i

### 4.3 การคัดเลือกโครโมโซม

เมื่อคำนวณค่าความเหมาะสมของแต่ละโครโมโซมจนครบตามจำนวนประชากรเบื้องต้นที่กำหนดแล้ว ขั้นตอนต่อไปคือการคัดเลือก ในงานวิจัยนี้ได้เลือกใช้วิธีการคัดเลือกแบบวงล้อรูเร็ท ซึ่งวิธีการคัดเลือกนี้จะพิจารณาจากค่าความเหมาะสมของแต่ละโครโมโซม โดยโครโมโซมที่มีค่าความเหมาะสมมากจะมีโอกาสถูกคัดเลือกได้มากกว่าโครโมโซมที่มีค่าความเหมาะสมน้อย ซึ่งวิธีการนี้จะต้องทำการหาส่วนกลับของค่าความเหมาะสมของแต่ละโครโมโซม และหาค่าความน่าจะเป็นในการถูกคัดเลือกดังแสดงในตารางที่ 1 และสร้างวงล้อรูเร็ทดังแสดงในรูปที่ 4

ตารางที่ 1 ค่าความน่าจะเป็นในการถูกคัดเลือก

| โครโมโซม | ค่าความเหมาะสม | 1/ค่าความเหมาะสม | ความน่าจะเป็น | ความน่าจะเป็นสะสม |
|---|---|---|---|---|
| 1 | 1824 | 0.00055 | 0.22 | 0.22 |
| 2 | 2238 | 0.00044 | 0.17 | 0.39 |
| 3 | 1538 | 0.00065 | 0.26 | 0.65 |
| 4 | 2429 | 0.00041 | 0.16 | 0.81 |
| 5 | 2026 | 0.00049 | 0.19 | 1.00 |
| **รวม** | **10055** | **0.00254** | **1.00** | |




รูปที่ 4 การสร้างวงล้อรูเล็ท

### 4.4 การครอสโอเวอร์

การครอสโอเวอร์เป็นขั้นตอนที่ทำหลังจากขั้นตอนการคัดเลือกโครโมโซมโดยการแลกเปลี่ยนบางส่วนของโครโมโซมพ่อแม่ (parent) ซึ่งจากโครโมโซมชุดใหม่ที่ได้จากการคัดเลือกนั้นจะมีเพียงโครโมโซมบางส่วนเท่านั้นที่จะถูกคัดเลือกมาตามอัตราความน่าจะเป็นในการครอสโอเวอร์ ($P_c$) เพื่อสร้างโครโมโซมรุ่นใหม่หรือโครโมโซมรุ่นลูก (offspring) ชุดใหม่ขึ้นมาโดยใช้วิธีการครอสโอเวอร์แบบลำดับ (Order Crossover : OX) โดยนำโครโมโซมแต่ละคู่มาทำการครอสโอเวอร์ดังแสดงตัวอย่างในภาพประกอบ 3.21 ซึ่งแสดงวิธีการครอสโอเวอร์แบบ OX โดยสามารถอธิบายขั้นตอนการครอสโอเวอร์ได้ดังนี้

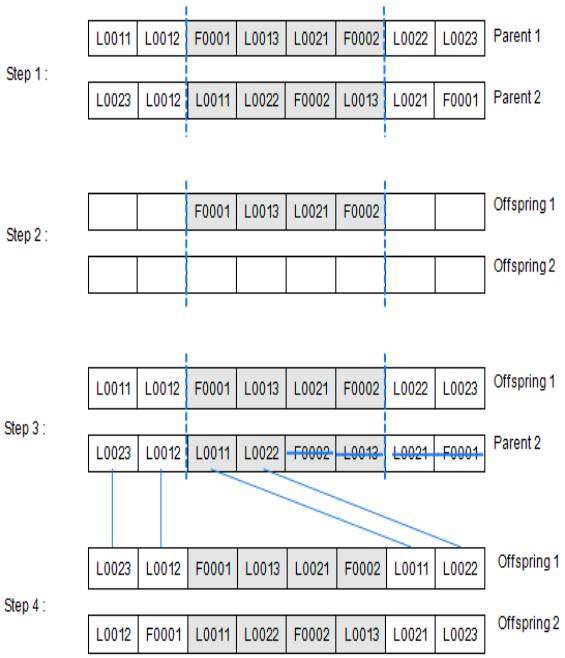
1. การสุ่มตำแหน่งยีนภายในโครโมโซม P1 และ P2 มาสองตำแหน่ง จากรูปที่ 5 ตำแหน่งยีนที่สุ่มได้ คือ ตำแหน่งที่ 2 และ 5
2. คัดลอกยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม P1 คือ ยีน F0001 L0013 L0021 F0002 ไปยังยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม O1
3. ลบค่ายีน F0001 L0013 L0021 F0002 ออกจากโครโมโซม P2 ซึ่งอยู่ในตำแหน่งที่ 4-8
4. นำยีนที่เหลือในโครโมโซม P2 คือยีน L0023 L0012 L0011 L0022 มาใส่ในตำแหน่งที่ยังว่างของโครโมโซม O1 ตามลำดับจากซ้ายไปขวา จะได้โครโมโซม O1 ซึ่งผ่านกระบวนการครอสโอเวอร์แล้ว
5. คัดลอกยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม P2 คือ ยีน L0011 L0022 F0002

L0013 ไปยังยีนที่อยู่ระหว่างตำแหน่งที่ 2 และ 5 ของโครโมโซม O2

6. ลบค่ายีน L0011 L0022 F0002 L0013 ออกจากโครโมโซม P1 ซึ่งอยู่ในตำแหน่งที่ 1, 2 และ 3

7. นำยีนที่เหลือในโครโมโซม P1 คือยีน L0012 F0001 L0021 F0002 มาใส่ในตำแหน่งที่ยังว่างของโครโมโซม O2 ตามลำดับจากซ้ายไปขวา จะได้โครโมโซม O2 ซึ่งผ่านกระบวนการครอสโอเวอร์แล้วเช่นเดียวกับโครโมโซม O1

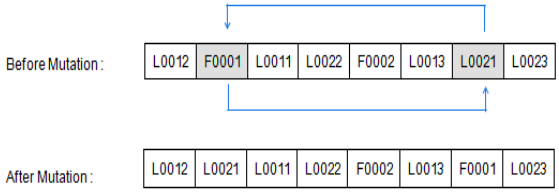
โดยที่การครอสโอเวอร์จะแยกทำในส่วนของแต่ละประเภทวิชาคือวิชาบรรยายจะทำการครอสโอเวอร์กันเฉพาะในส่วนของวิชาบรรยาย วิชาปฏิบัติก็จะทำการครอสโอเวอร์เฉพาะในส่วนของวิชาปฏิบัติเช่นเดียวกัน ซึ่งมีขั้นตอนดังแสดงในรูปที่ 5




รูปที่ 5 วิธีการครอสโอเวอร์แบบลำดับ

### 4.5 การมิวเตชัน

การมิวเตชันเป็นขั้นตอนที่อาจช่วยให้โครโมโซมมีค่าความเหมาะสมดีขึ้นหลังจากการครอสโอเวอร์ การมิวเตชันทำได้โดยการสลับตำแหน่งของค่าภายในโครโมโซมตัวเดียว จะมีโครโมโซมเพียงบางส่วนเท่านั้นที่จะถูกนำมาทำการมิวเตชันซึ่งขึ้นอยู่กับค่าความน่าจะเป็นในการมิวเตชัน (%Pm) โดยใช้วิธีการมิวเตชันแบบ Two-point Swapping Mutation เนื่องจากเป็นวิธีการมิวตชันที่เหมาะสำหรับปัญหาที่มีรูปแบบโครโมโซมเป็นแบบการจัดตารางเวลา ซึ่งมีขั้นตอนดังแสดงในรูปที่ 6




รูปที่ 6 วิธีการมิวเตชันแบบ Two-point Swapping Mutation

## 5. การวิเคราะห์ความไวเพื่อหาค่าของพารามิเตอร์เจเนติกอัลกอริทึม

การวิเคราะห์ความไว (sensitivity analysis) คือการศึกษาถึงการเปลี่ยนแปลงของคำตอบที่ดีที่สุด (optimal solution) เมื่อเกิดการเปลี่ยนแปลงส่วนใดส่วนหนึ่งของแบบจำลอง เช่น ค่าคงที่ ค่าพารามิเตอร์ การวิเคราะห์ศึกษาว่าปัจจัยใดมีผลต่อฟังก์ชันวัตถุประสงค์ของแบบจำลอง โดยจะกำหนดปัจจัยที่ต้องการทดสอบให้มีการเปลี่ยนแปลงค่าไปแต่ควบคุมปัจจัยอื่นๆ ให้คงที่ การวิเคราะห์ความไวสามารถบอกได้ว่าปัจจัยใดเป็นปัจจัยเสี่ยงที่ต้องระวังและพยายามควบคุมไม่ให้คลาดเคลื่อนมากจนเกินไปเพราะจะมีผลต่อฟังก์ชันวัตถุประสงค์ในเชิงลบ

สำหรับงานวิจัยนี้ใช้การวิเคราะห์ความไวของผลการวิจัยจากการพัฒนาโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนโดยประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบโดยการวิเคราะห์ความไวของพารามิเตอร์ของเจเนติกอัลกอริทึมเพื่อหาค่าพารามิเตอร์ที่เหมาะสมสำหรับขนาดข้อมูลที่จะจัดตารางสอนซึ่งจะส่งผลให้ได้คำตอบที่ดีที่สุดสำหรับข้อมูลนั้นๆ โดยจะวิเคราะห์การเปลี่ยนแปลงของคำตอบที่ดีที่สุดเมื่อค่าพารามิเตอร์ของเจเนติกอัลกอริทึมของโปรแกรมเปลี่ยนไป พารามิเตอร์ของเจเนติกอัลกอริทึมที่วิเคราะห์ได้แก่ จำนวนประชากร (population size) และจำนวนเจนเนอเรชัน (generation) โดยกำหนดขนาดของข้อมูลสำหรับประมวลผล ซึ่งขนาดของข้อมูลในที่นี้จะหมายถึงจำนวนยีนหรือจำนวนรายวิชา อาจารย์ผู้สอน กลุ่มนักศึกษา

และห้องเรียนที่ใช้ประมวลผลโปรแกรมซึ่งกำหนดเป็น 7 ช่วงโดยแบ่งช่วงขนาดของข้อมูลเทียบกับขนาดของข้อมูลจริงเป็น 10 เปอร์เซ็นต์ 20 เปอร์เซ็นต์ 50 เปอร์เซ็นต์ 80 เปอร์เซ็นต์ 100 เปอร์เซ็นต์120 เปอร์เซ็นต์และ 150 เปอร์เซ็นต์ของข้อมูลจริง โดยข้อมูลจริงประกอบด้วยห้องรียนจำนวน 59 ห้อง จัดทำการเรียนการสอน 5 วัน วันละ 12 คาบเรียน ดังนั้นจำนวนยีนต่อ 1 โครโมโซมเท่ากับ 3,540 ยีน ตามตารางที่ 2 และจากการทดลองได้ผลดังตารางที่ 3

ตารางที่ 2 ช่วงจำนวนยีนที่ทำการทดลอง

| ขนาดข้อมูล (เทียบกับขนาดข้อมูลจริง) | จำนวนยีนต่อ 1 โครโมโซม |
|---|---|
| 10 เปอร์เซ็นต์ | 360 |
| 20 เปอร์เซ็นต์ | 720 |
| 50 เปอร์เซ็นต์ | 1,800 |
| 80 เปอร์เซ็นต์ | 2,880 |
| 100 เปอร์เซ็นต์ (ข้อมูลจริง) | 3,540 |
| 120 เปอร์เซ็นต์ | 4,320 |
| 150 เปอร์เซ็นต์ | 5,400 |

ตารางที่ 3 ค่าพารามิเตอร์ที่เหมาะสมสำหรับจำนวนยีนในแต่ละช่วงจำนวนยีน

| ช่วงจำนวนยีน | จำนวนประชากร (โครโมโซม) | จำนวนเจนเนอเรชัน | เวลาในการประมวลผลโปรแกรม (วินาที) |
|---|---|---|---|
| 10 % | 100 | 500 | 252 |
| 20 % | 100 | 500 | 478 |
| 50 % | 100 | 500 | 1,116 |
| 80 % | 100 | 500 | 1,902 |
| 100 % | 100 | 500 | 2,889 |
| 120 % | 100 | 500 | 5,712 |
| 150 % | 100 | 600 | 11,735 |

สามารถแสดงการเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้นได้ดังแสดงในรูปที่ 7 ซึ่งมีลักษณะการเพิ่มขึ้นของเวลาในการประมวลผลเป็นแบบเอ็กโปเนนเชียล แสดงเป็นสมการคือ $y = 144.67e^{0.6227x}$

นอกจากค่าพารามิเตอร์ต่างๆ ที่กล่าวมาข้างต้นแล้ว ยังมีพารามิเตอร์อีกหนึ่งตัวที่คิดว่าน่าจะมีผลต่อคำตอบที่ดีที่สุดนั้นคือ จำนวนประชากรเบื้องต้น (initialize) คือ จำนวนเริ่มต้นที่กำหนดให้โปรแกรมประมวลผลจำนวนโครโมโซมเพื่อที่จะคัดเลือกไปเป็นประชากรที่เข้าสู่กระบวนการเจเนติกอัลกอริทึมตามจำนวนประชากรที่กำหนด เนื่องจากสมมุติฐานที่ว่าหากเพิ่มพื้นที่ในการหาคำตอบให้มาก ค่าคำตอบที่ดีที่สุดของปัญหาก็จะดีขึ้นด้วย ซึ่งไม่มีงานวิจัยใดที่เคยทดสอบสมมุติฐานนี้ ดังนั้นจึงได้ทำการทดสอบสมมุติฐาน โดยเปลี่ยนแปลงค่าจำนวนประชากรเบื้องต้นและกำหนดค่าพารามิเตอร์ตัวอื่นๆ ให้คงที่ คือ

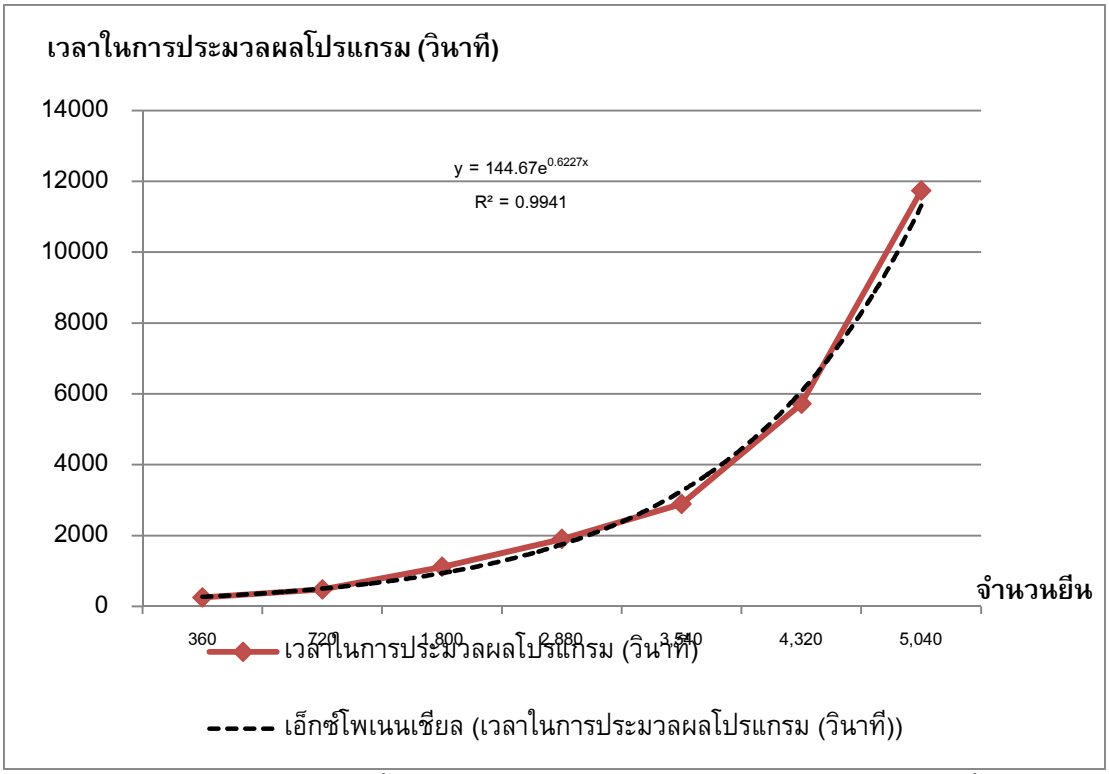
- จำนวนประชากรเท่ากับ 100 โครโมโซม
- จำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน
- ค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8
- ค่าความน่าจะเป็นในมิวเตชันเท่ากับ 0.3

ผลการวิเคราะห์จำนวนประชากรเบื้องต้นดังตารางที่ 4

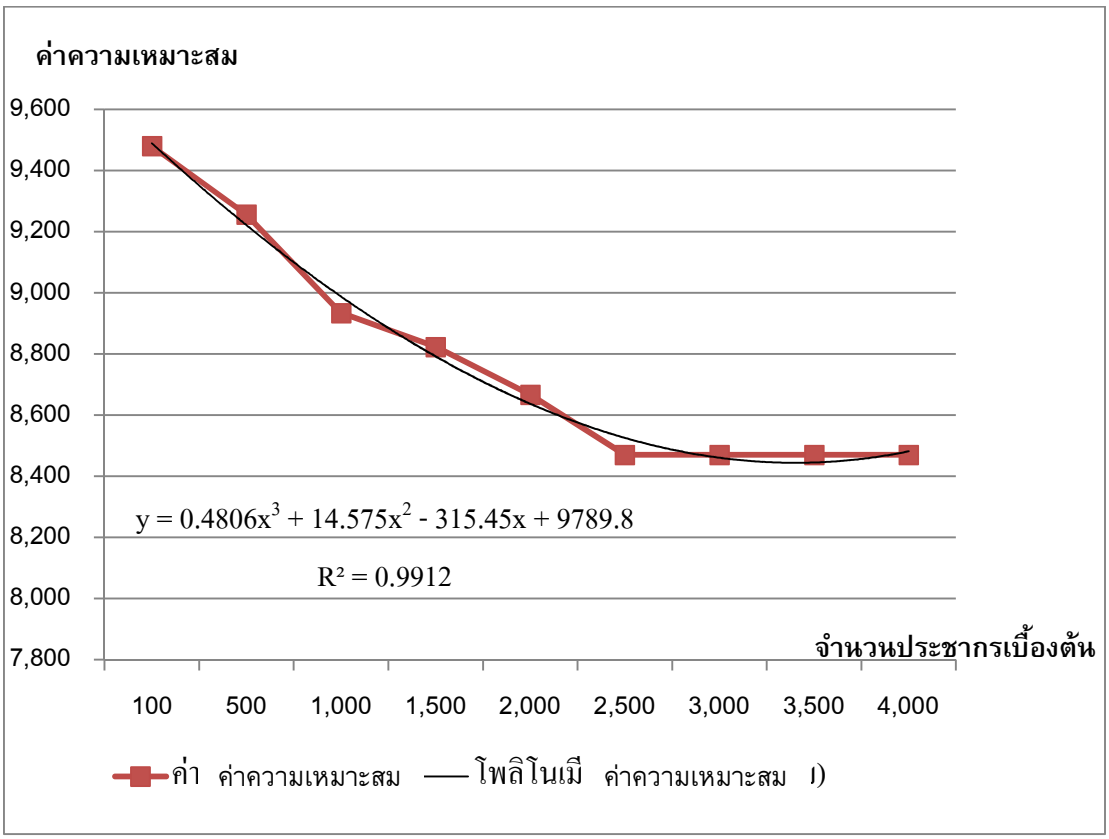
ตารางที่ 4 ผลการวิเคราะห์จำนวนประชากรเบื้องต้น

| จำนวนประชากรเบื้องต้น (โครโมโซม) | ค่าความเหมาะสม |
|---|---|
| 100 | 9,480 |
| 500 | 9,256 |
| 1,000 | 8,934 |
| 1,500 | 8,823 |
| 2,000 | 8,667 |
| 2,500 | 8,470 |
| 3,000 | 8,470 |
| 3,500 | 8,470 |
| 4,000 | 8,470 |

พบว่าเมื่อจำนวนประชากรเบื้องต้นเพิ่มมากขึ้นค่าความเหมาะสมก็จะยิ่งลดลง สรุปได้ว่าจำนวนประชากรเบื้องต้นที่เหมาะสม คือ 2,500 โครโมโซม




รูปที่ 7 การเพิ่มขึ้นของเวลาในการประมวลผลโปรแกรมเมื่อจำนวนยีนเพิ่มขึ้น




รูปที่ 8 การลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นเพิ่มขึ้น

จำนวนประชากรเบื้องต้นจะมีผลต่อค่าความเหมาะสม คือเมื่อจำนวนประชากรเบื้องต้นมากขึ้นก็จะส่งผลให้ค่าความเหมาะสมที่ลดลงตามไปด้วย โดยกราฟแสดงการลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นที่มากขึ้น แสดงได้ดังรูปที่ 8 ซึ่งกราฟแสดงการลดลงของค่าความเหมาะสมเมื่อจำนวนประชากรเบื้องต้นเพิ่มขึ้น ซึ่งลักษณะการลดลงของค่าความเหมาะสม นั้นสามารถแสดงได้ดังสมการคือ $y = 0.4806x^3 + 14.575x^2 - 315.45x + 9789.8$

จากการวิเคราะห์ความไวหาค่าพารามิเตอร์ที่เหมาะสมที่ได้ทำการวิเคราะห์ดังที่กล่าวมาข้างต้นแล้ว ผู้วิจัยมีสมมุติฐานว่ายังมีค่าพารามิเตอร์อีกตัวหนึ่งที่ผลต่อค่าความเหมาะสมของตารางสอนที่ได้จากโปรแกรมที่พัฒนาขึ้นนั้นคือ ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ ที่จะกล่าวถึงในหัวข้อถัดไป

## 6. การวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์

ในการวิเคราะห์ความไวเพื่อหาค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่เหมาะสมสำหรับโปรแกรมคอมพิวเตอร์ช่วยในการจัดตารางสอนโดยประยุกต์ใช้เจเนติกอัลกอริทึมในการหาคำตอบโดยจะวิเคราะห์การเปลี่ยนแปลงของการหาคำตอบที่เป็นไปได้และจำนวนครั้งของการละเมิดเงื่อนไขเพื่อความสมบูรณ์แต่ละข้อของคำตอบที่ดีที่สุดเมื่อค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์เปลี่ยนไป ณ เจนเนอเรชันต่างๆ โดยกำหนดช่วงของค่าน้ำหนักเพื่อความสมบูรณ์สำหรับประมวลผล ซึ่งค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ของแต่ละข้อนั้นจะถูกกำหนดตามลำดับความสำคัญของเงื่อนไขข้อนั้นๆ แบ่งการกำหนดค่าหนักของเงื่อนไขเพื่อความสมบูรณ์เป็น 6 ช่วงตามตารางที่ 5 และทำการทดสอบตามค่าพารามิเตอร์ที่เหมาะสมคือ จำนวนประชากรเบื้องต้น 2,500 โครโมโซมจำนวนประชากร 100 โครโมโซมกำหนดค่าความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.8 และค่าความน่าจะเป็นในการมิวเตชันเท่ากับ 0.3 และจำนวนเจนเนอเรชันที่วิเคราะห์คือ 100, 200, 300, 400, 500, 600, 700 และ 800 เจนเนอเรชัน จากการทดลองได้ผลการทดลองดังตามตารางที่ 6

ตารางที่ 5 ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง

| ข้อ | ค่าน้ำหนัก | | | | | |
|---|---|---|---|---|---|---|
| | ช่วงที่ 1 | ช่วงที่ 2 | ช่วงที่ 3 | ช่วงที่ 4 | ช่วงที่ 5 | ช่วงที่ 6 |
| 1 | 40 | 100 | 200 | 300 | 400 | 500 |
| 2 | 40 | 100 | 200 | 300 | 400 | 500 |
| 3 | 30 | 75 | 150 | 225 | 300 | 375 |
| 4 | 20 | 50 | 100 | 150 | 200 | 250 |
| 5 | 10 | 25 | 50 | 75 | 100 | 125 |
| 6 | 10 | 25 | 50 | 75 | 100 | 125 |

ตารางที่ 6 ผลการวิเคราะห์ความไวของค่าน้ำหนักเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วง

| ช่วงค่าน้ำหนัก | เจนเนอเรชันที่พบคำตอบที่เป็นไปได้ | เจนเนอเรชันที่พบคำตอบที่ดีที่สุด |
|---|---|---|
| **ช่วงที่ 1** | 100 | 500 |
| **ช่วงที่ 2** | 100 | 500 |
| **ช่วงที่ 3** | 200 | 700 |
| **ช่วงที่ 4** | 300 | 800 |
| **ช่วงที่ 5** | 500 | N/A |
| **ช่วงที่ 6** | N/A | N/A |

จากตารางที่ 6 ค่าน้ำหนักของเงื่อนไขช่วงที่ 1 และ 2 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 100 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 500 ค่าน้ำหนักของเงื่อนไขช่วงที่ 3 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 200 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 700 ค่าน้ำหนักของเงื่อนไขช่วงที่ 4 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 300 และพบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ 800 ค่าน้ำหนักของเงื่อนไขช่วงที่ 5 พบคำตอบที่เป็นไปได้ในเจนเนอเรชันที่ 800 แต่ไม่พบคำตอบที่ดีที่สุดในเจนเนอเรชันที่ทำการทดลอง และค่าน้ำหนักของเงื่อนไขช่วงที่ 6 ไม่พบคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดในเจนเนอเรชันที่ทำการทดลองทั้งนี้เหตุที่ไม่สามารถทำการทดลองในจำนวนเจนเนอเรชันที่มากกว่าที่กำหนดนั้น เนื่องจากความสามารถทางด้านคอมพิวเตอร์ที่ใช้ในการทดลอง

กล่าวโดยสรุปได้ว่า ค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ในแต่ละช่วงนั้นมีผลต่อจำนวนเจนเนอเรชันที่พบคำตอบที่เป็นไปได้และจำนวนเจนเนอเรชันที่พบคำตอบที่ดีที่สุด เนื่องจากในงานวิจัยนี้ได้กำหนดค่าน้ำหนักของเงื่อนไขบังคับเท่ากับ 1,000 ดังนั้นช่วงของค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่มีค่าน้อยแตกต่างกับค่าน้ำหนักของเงื่อนไขบังคับมาก ส่งผลให้ช่วงนั้นๆ สามารถหาคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดได้อย่างรวดเร็ว

## 7. ผลสรุป

การประมวลผลโปรแกรมพบว่าค่าความเหมาะสมของตารางสอนที่น้อยที่สุดจะมีค่าลดลงเมื่อจำนวนเจนเนอเรชันเพิ่มขึ้นและลดลงต่ำสุดคงที่ค่าหนึ่ง ณ เจนเนอเรชันใดๆ ซึ่งหมายถึงค่าที่เหมาะสมที่สุดสำหรับปัญหานั้นๆ ซึ่งจากผลการการวิเคราะห์สามารถกำหนดค่าพารามิเตอร์ที่เหมาะสมสำหรับงานวิจัยนี้คือ จำนวนประชากรเบื้องต้นเท่ากับ 2,500 โครโมโซม จำนวนประชากรเท่ากับ 100 โครโมโซม จำนวนเจนเนอเรชันเท่ากับ 500 เจนเนอเรชัน ค่าความน่าจะเป็นในการครอสโอเวอร์ 0.8 และค่าความน่าจะเป็นในการมิวเตชัน 0.3 และไม่พบคำตอบที่เป็นไปไม่ได้เนื่องจากมีการกำหนดค่าน้ำหนักของเงื่อนไขบังคับและเงื่อนไขเพื่อความสมบูรณ์ที่แตกต่างกันมาก ดังนั้นโครโมโซมใดที่ละเมิดเงื่อนไขบังคับก็จะมีค่าน้ำหนักมาก ทำให้ไม่ถูกคัดเลือกเข้าสู่เจนเนอเรชันถัดไป

นอกจากนี้จากการประมวลผลของโปรแกรมยังพบว่า ขนาดจำนวนยีน มีผลต่อเวลาในการประมวลผลโปรแกรมโดยขนาดจำนวนยีนที่เพิ่มขึ้นจะทำให้ใช้เวลาในการประมวลผลโปรแกรมเพิ่มขึ้นแบบเอ็กโปเนนเชียล

ส่วนค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์มีผลต่อจำนวนเจนเนอเรชันที่พบคำตอบที่เป็นไปได้และจำนวนเจนเนอเรชันที่พบคำตอบที่ดีที่สุด โดยค่าน้ำหนักของเงื่อนไขเพื่อความสมบูรณ์ที่มีค่าน้อยแตกต่างกับค่าน้ำหนักของเงื่อนไขบังคับมาก ส่งผลให้ช่วงนั้นๆ สามารถหาคำตอบที่เป็นไปได้และคำตอบที่ดีที่สุดได้อย่างรวดเร็ว


## กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนจากเงินรายได้มหาวิทยาลัยสงขลานครินทร์


## เอกสารอ้างอิง


[1] H. Kanoh and Y. Sakamoto, “Knowledge-based Genetic Algorithm for University Course Timetabling Problems”, *Japan International Journal of Knowledge-based and Intelligent Engineering Systems*., 2008, Vol. 12, pp 283-294.

[2] A. Wren, "Scheduling, timetables and roustering - A special relationship? In : Burke, E., Ross, P. (Eds), Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science,," *Springer, Berlin,* vol. 1153, 1996.

[3] D. Abramson. "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms." *Management Science,* vol. 37, pp. 98-113. 1991.

[4] D. Costa. "A tabu search algorithm for computing an operational timetable." *European Journal of Operational Research,* vol. 76, pp. 98-110. 1994.

[5] M. J. F. Souza. "A GRASP-tabu search algorithm to solve a school timetabling problem," presented at the In 4th Metaheuristic international conference, Portugal:Porto, 2001.

[6] K. Socha., M. Sampels., and M. Manfrin. "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," in *Applications of Evolutionary Computing*. vol. 2611, pp. 334-345. 2003.

[7] J. Knowles. and M. Sampels. "A max-min ant system for the university course timetabling problem." *Lecture Notes in Computer Science,* vol. 2463, pp. 1-13. 2002.

[ 8] E. K. Burke., B. McCollum., A. Meisels., S. Petrovic., and R. Qu. "A graph-based hyper-heuristic for educational timetabling problems." *European Journal of Operational Research,* vol. 176, pp. 177-192. 2007.

[9] M. P. Carrasco. and M. V. Pato. "A multiobjective genetic algorithm for the class/teacher timetabling problem," in *Practice and Theory of Automated Timetabling Iii*. vol. 2079, pp. 3-17. 2001.

[10] ปริศนา แช่มสุขขี. "เอกสารสัมมนาคอมพิวเตอร์ Genetic Algorithm (GA)". ภาควิชาวิทยาการ

คอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา. 2547. (สำเนา)

[11] E. K. Burke., David Elliman., and R. Weare. "A Genetic Algorithm Based University Timetabling System," in *In Proceedings of the 2nd East-West International Conference on Computer Technologies in Education*, Ukraine : Crime, pp. 35-40. 1994.

[12] M. Dimopoulou. and P. Miliotis. "Implementation of a university course and examination timetabling system," *European Journal of Operational Research,* vol. 130, pp. 202-213. 2001.

[13] W. Erben. and J. Keppler. "A genetic algorithm solving a weekly course-timetabling problem Practice and Theory of Automated Timetabling." vol. 1153, pp. 198-211. 1996.

[14] D. Abramson. and J. Abela. "A Parallel Genetic Algorithm for Solving the School Timetabling Problem." *In 15th Australian Computer Science Conference*, Hobart, pp. 1-11. 1992.

[15] R. A. Caldeira. "School Timetabling using Genetic Search." in *In Practice and theory of automated timetabling*, L.-I. IST, Ed., ed Toronto : University of Toronto, pp. 115-122. 1997.

[16] กิตติ ไพฑูรย์วัฒนกิจ และ กาญจนี วงศ์วิภาพร. "GA สำหรับจัดตารางสอนโรงเรียน." วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2541.

[17] Y.Zen Wang "Using genetic algorithm methods to solve course scheduling problems." *Expert Systems with Applications,* vol. 25, pp. 12-19. 2003.

[18] N. A. Zahra. "Hybrid heuristics for Examination Timetabling problem." *Applied Mathematics and Computation,* vol. 163, pp. 705-733. 2005.

[19] P. Pongcharoen., W. Promtet., P. Yenradee., and C. Hicks. "Stochastic optimisation timetabling tool for university course scheduling." *International Journal of Production Economics,* vol. 112, pp. 903-918. 2008.

# ภาคผนวก

# ภาคผนวก ก
# ตัวอย่างแบบสอบถามความสำคัญของเงื่อนไขหรือข้อจำกัดที่ใช้ในการจัดตารางสอน

**แบบสอบถามความสำคัญของเงื่อนไขหรือข้อจำกัดที่ใช้ในการจัดตารางสอน**

**คำชี้แจง** แบบสอบถามนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโท คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ จัดทำขึ้นเพื่อเป็นแนวทางในการปรับปรุงแก้ไขการจัดการตารางสอนของนักศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยทำการสำรวจเกี่ยวกับความสำคัญของเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ใช้ในการพิจารณาการจัดตารางสอนในปัจจุบัน

แบบสอบถามนี้ แบ่งเป็น 3 ตอน คือ

ตอนที่ 1 ข้อมูลเกี่ยวกับผู้ตอบแบบสอบถาม

ตอนที่ 2 ความคิดเห็นเกี่ยวกับลำดับความสำคัญของเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้การจัดตารางสอน

ตอนที่ 3 ความคิดเห็นเพิ่มเติมเกี่ยวกับเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้การจัดตารางสอน

โปรดตอบแบบสอบถามตามความเป็นจริงและตรงกับความคิดเห็นของท่านมากที่สุด พร้อมทั้งให้ข้อเสนอแนะเกี่ยวกับเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้ในการจัดตารางสอน

**ตอนที่ 1 ข้อมูลเกี่ยวกับผู้ตอบแบบสอบถาม**

**คำชี้แจง** โปรดเขียนเครื่องหมาย ✓ ลงในช่อง ❒ และกรอกข้อมูลเกี่ยวกับตัวท่านลงในช่องว่าง

1. ประจำภาควิชา

……**……**

2. ตำแหน่งทางวิชาการ ❒ 1) ศาสตราจารย์ ❒ 2) รองศาสตราจารย์
❒ 3) ผู้ช่วยศาสตราจารย์ ❒ 4) อาจารย์

3. ประสบการณ์ด้านการสอนในมหาวิทยาลัยสงขลานครินทร์……ปี

4. ปัจจุบันท่านสอนอยู่ในระดับใดบ้าง (ตอบได้มากกว่า 1 ข้อ)
❒ ปริญญาตรี ❒ ปริญญาโท ❒ ปริญญาเอก

**ตอนที่ 2 ความคิดเห็นเกี่ยวกับลำดับความสำคัญของเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้การจัดตารางสอน**

**คำชี้แจง** โปรดให้ลำดับความสำคัญโดยเรียงจากลำดับความสำคัญมากไปลำดับความสำคัญน้อย (1-6) ที่ตรงกับความคิดเห็นของท่านมากที่สุด

| เงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ | ลำดับความสำคัญ |
|---|---|
| ก. ช่วงเวลาที่ 2 ของวิชาไม่ควรมีอยู่ในวันเดียวกัน | |
| ข. ควรจัดการสอนรายวิชาของภาควิชาให้กับห้องเรียนในอยู่ในภาควิชานั้นก่อน | |
| ค. ในแต่ละวันเมื่อนักศึกษาที่มีคาบเรียนก่อนหน้าแล้วไม่ควรมีคาบว่างเกิน 2 คาบ เพื่อเรียนในคาบถัดไป | |
| ง. ในแต่ละวันควรมีการเว้นคาบว่างในเวลา 12.00-12.50 น. เพื่อให้นักศึกษาพักทานกลางวัน | |
| จ. ในแต่ละวันนักศึกษาไม่ควรเรียนติดต่อกันเกิน 4 คาบ | |
| ฉ. ในแต่ละวันอาจารย์ไม่ควรสอนติดกันเกิน 4 คาบ | |

**ตอนที่ 3 ความคิดเห็นเพิ่มเติมเกี่ยวกับเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้การจัดตารางสอน**

**คำชี้แจง** โปรดให้ความคิดเห็นเพิ่มเติมเกี่ยวกับเงื่อนไขหรือข้อจำกัดเพื่อความสมบูรณ์ที่ควรใช้การจัดตารางสอน เช่น ไม่ควรให้มีการสอนสำหรับอาจารย์ประจำภาควิชาวิศวกรรมอุตสาหการในช่วงเช้าของวันพฤหัสบดี เนื่องจากมีการประชุมภาควิชาในช่วงเวลานั้น

……

……

……

……

……

……

……

……

……

……

……