การออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึม

ATM NETWORK DESIGN USING GENETIC ALGORITHM

ธนาวุฒิ อรชุน
THANAWUT ORACHUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2545

ISBN 974-643-752-3

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึม

ATM NETWORK DESIGN USING GENETIC ALGORITHM

ธนาวุฒิ อรชุน

THANAWUT ORACHUN

เลขหมู่.....
เลขทะเบียน 43259
วัน, เดือน, ปี 8 ส.ค. 2545

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2545

ISBN 974-648-752-3

COPYRIGHT 2002

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ATM NETWORK DESIGN USING GENETIC ALGORITHM

THANAWUT ORACHUN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

ISBN 974-648-752-3

หัวข้อวิทยานิพนธ์	การออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึม
นักศึกษา	นายธนาวุฒิ อรชุน
รหัสประจำตัว	42061063
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.สุวิพล สิทธิชีวภาค

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการในการออกแบบโครงข่าย ATM โดยการประยุกต์ใช้จีเนติกอัลกอริทึมซึ่งเป็นกระบวนการในการหารูปแบบคำตอบที่ดีที่สุดโดยมีวัตถุประสงค์เพื่อออกแบบโครงข่ายที่มีดีเลย์เฉลี่ยของเซลล์ในโครงข่ายและค่าใช้จ่ายในระบบสายสัญญาณที่เหมาะสมที่สุด โดยนำโครงสร้างแบบทรีมาใช้ในการออกแบบ ในกระบวนการของจีเนติกอัลกอริทึมนั้นรูปแบบที่เป็นไปได้ในการเชื่อมต่อระหว่างโหนดของโครงข่ายแบบทรีจะถูกแทนด้วยโครโมโซมซึ่งได้มาจากการเข้ารหัสโดยใช้ Prüfer Number โดยโครโมโซมเหล่านี้จะถูกนำเข้าสู่กระบวนการทางจีเนติกอัลกอริทึม จนกระทั่งได้รูปแบบการเชื่อมต่อของโครงข่ายแบบทรีที่มีความเหมาะสมที่สุด

Thesis Title	ATM Network Design using Genetic Algorithm
Student	Mr.Thanawut Orachun
Student ID.	42061063
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2002
Thesis Advisor	Assoc. Prof. Dr. Suvepon Sittichivapak

ABSTRACT

This thesis presents an ATM network design using Genetic Algorithm. The objective of this thesis is to design the ATM network with the optimum cost and average cell delay in network. We use the spanning tree topology to design the ATM network. For Genetic Algorithm, a possible interconnection of nodes in the ATM network is represented by chromosome. The chromosomes are encoded by Prüfer Number and passed the operation of Genetic Algorithm until the optimum interconnection of nodes in network is received.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์และความช่วยเหลือให้คำแนะนำ ให้คำปรึกษาในหลายๆด้านจาก รองศาสตราจารย์ ดร.สุวิพล สิทธิชีวภาค อาจารย์ที่ปรึกษา ซึ่งท่านได้ชี้แนวทางในการทำวิจัยจนสำเร็จลงด้วยดี กระผมขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูง

ขอขอบพระคุณ บิดา มารดา ญาติพี่น้อง และ ทุกๆคน ที่เป็นกำลังใจและให้ความช่วยเหลือด้วยดีมาโดยตลอด

ธนาวุฒิ อรชุน

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญรูป.....VII

บทที่1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์ของการศึกษา.....1

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....2

1.4 ขอบเขตการวิจัยและขั้นตอนการวิจัย.....2

1.5 โครงสร้างของวิทยานิพนธ์.....3

บทที่2 หลักการพื้นฐานสำหรับการวิเคราะห์ระบบโครงข่ายข้อมูล.....4

2.1 การสื่อสารข้อมูลแบบ ATM เบื้องต้น.....4

2.1.1 องค์ประกอบของโครงข่าย ATM.....4

2.1.2 การแบ่งชั้นสื่อสารใน ATM 5

2.2 แบบจำลองคิวของระบบโครงข่าย.....7

2.2.1 การส่งข้อมูลในโครงข่าย.....7

2.2.2แบบจำลองคิวประเภทต่างๆ.....10

2.3 การวิเคราะห์ประสิทธิภาพของระบบโครงข่าย 13

2.3.1 การวิเคราะห์ค่าดีเลย์เฉลี่ยของโครงข่าย.....13

2.3.2 การวิเคราะห์ค่าใช้จ่ายของระบบสายสัญญาณของโครงข่าย.....15

บทที่3 หลักการพื้นฐานเกี่ยวกับจีเนติกอัลกอริทึม.....16

3.1 พันธุศาสตร์ทางชีววิทยา.....16

3.2 จีเนติกอัลกอริทึมเบื้องต้น.....20

3.2.1 ฟังก์ชันเป้าหมายกับฟังก์ชันความเหมาะสม.....22

สารบัญ(ต่อ)

หน้า

3.2.2 รูปแบบโครโมโซม.....22

3.2.3 วัฏจักรจีเนติกอัลกอริทึม.....23

3.2.4 พันธุศาสตร์ทางชีววิทยา กับ จีเนติกอัลกอริทึม.....25

3.3 จีเนติกอัลกอริทึมแบบง่าย.....27

3.4 การประยุกต์จีเนติกอัลกอริทึมแบบง่าย.....37

บทที่4 การนำจีเนติกอัลกอริทึมมาใช้ในการออกแบบโครงข่าย ATM.....46

4.1 โมเดลของระบบ.....46

4.2 การใช้จีเนติกในการออกแบบโครงข่าย ATM.....46

4.2.1 การกำหนดรูปแบบโครโมโซม.....46

4.2.2 การวัดค่าความเหมาะสม.....48

4.2.3 การคัดเลือก.....48

4.2.4 การครอสโอเวอร์.....48

4.2.5 การมิวเตชัน.....49

4.2.6 การรีโพรดักชัน.....49

4.3 ผลการทดลอง.....51

บทที่5 สรุปผลการวิจัยและข้อเสนอแนะ.....57

เอกสารอ้างอิง.....59

ภาคผนวก.....60

ภาคผนวก ก. ซอร์ดโค๊ดโปรแกรม.....61

ภาคผนวก ข. ผลงานวิจัยที่ได้รับการตีพิมพ์.....78

ประวัติผู้เขียน.....79

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงคำศัพท์ทางพันธุศาสตร์กับทางจีเนติกอัลกอริทึม.....26

3.2 การกำหนด SGA พารามิเตอร์เปรียบเทียบผลการทำงานในการหาค่าสูงสุด ของฟังก์ชัน $y = x^n$ ที่ n=10.....42

4.1 อัตราเฉลี่ยในการส่งข้อมูลระหว่างโหนด(Mbps).....51

4.2 ระยะห่างระหว่างโหนด(Km).....51

สารบัญรูป

รูปที่ หน้า

2.1 แสดงมาตรฐานการเชื่อมต่อ UNI และ NNI ภายในโครงข่าย ATM.....5

2.2 โครงสร้างชั้นสื่อสารของ ATM.....5

2.3 โครงสร้างของเซลล์ ATM.....6

2.4 แสดงโครงข่ายที่มีหนึ่งบัฟเฟอร์และหนึ่งช่องสัญญาณการให้บริการ.....8

2.5 แสดงกราฟของ μCT และ N เทียบกับ ρ 11

2.6 แสดงกราฟของ μCW และ N_q เทียบกับ ρ สำหรับแบบจำลองคิว M/M/1.....12

2.7 กราฟของ μCT และ N เทียบกับ ρ สำหรับแบบจำลองคิว M/M/1 เปรียบเทียบกับ M/D/1.....13

3.1 แสดงโครโมโซมทางพันธุศาสตร์.....17

3.2 แสดงการแบ่งตัวของเซลล์ทางพันธุศาสตร์.....18

3.3 แสดงการครอสโอเวอร์ทางพันธุศาสตร์.....19

3.4 แสดงการมิวเตชันทางพันธุศาสตร์.....20

3.5 แสดงหลักการเบื้องต้นของ GA.....21

3.6 แสดงวัฏจักรจีเนติกอัลกอริทึม.....24

3.7 แสดงรายละเอียดทางพันธุศาสตร์ กับ จีเนติกอัลกอริทึม.....26

3.8 แสดงไดอะแกรมการทำงานของจีเนติกอัลกอริทึมแบบง่าย.....27

3.9 ครอสโอเวอร์แบบ 1 จุด.....32

3.10 ไบนารีมิวเตชัน.....33

3.11 รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10.....35

3.12 แสดงกราฟค่าความเหมาะสมสูงสุดของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10.....37

3.13 รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี.....38

3.14 ครอสโอเวอร์แบบ 2 จุด.....39

3.15 อินเวอร์ชัน.....40

3.16 ผลการทำงานโดยสรุปของ GA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10.....43

4.1 แสดงความสัมพันธ์ของโครงสร้างแบบทรี กับ Prüfer Number.....47

4.2 การทำครอสโอเวอร์.....49

4.3 การทำมิวเตชัน.....49

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.4 แผนผังการทำงานของกระบวนการทางจีเนติกอัลกอริทึม.....50

4.5 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมเมื่อ Fitness Value = 500 หน่วย.....52

4.6 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 500 หน่วย 53

4.7 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมเมื่อ Fitness Value = 400 หน่วย.....53

4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 400 หน่วย 54

4.9 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมเมื่อ Fitness Value = 300 หน่วย.....55

4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 300 หน่วย 55

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการออกแบบรูปแบบการเชื่อมต่อของโครงข่ายนั้นมีความสำคัญเป็นอย่างยิ่งไม่ว่าจะเป็นโครงข่ายแบบใดก็ตาม ซึ่งในปัจจุบันมีโครงข่ายต่างๆมากมายสำหรับโครงข่าย ATM นั้นก็เป็นโครงข่ายแบบหนึ่งที่มีประสิทธิภาพในการส่งข้อมูลสูงโดยสามารถให้บริการข้อมูลข่าวสารได้ทุกประเภทเช่น เสียงพูด ข้อมูล หรือภาพเคลื่อนไหว เป็นต้นโดยทำการส่งข้อมูลเป็นแพ็กเกจที่มีขนาดคงที่ที่เรียกว่าเซลล์ อีกทั้งมีความเร็วสูงในการส่งข้อมูล ดังนั้นถ้าสามารถออกแบบโครงข่ายให้มีความเหมาะสมยิ่งเป็นการเพิ่มประสิทธิภาพในการส่งข้อมูลของโครงข่าย ATM ให้มีประสิทธิภาพสูงยิ่งขึ้น จะเห็นได้ว่าการออกแบบรูปแบบของโครงข่ายนั้นมีความสำคัญเป็นอย่างมากโดยทั่วไปในการออกแบบโครงข่ายนั้นเพื่อที่จะออกแบบโครงข่ายให้มีประสิทธิภาพตามที่ต้องการจะต้องพิจารณาค่าต่างๆไม่ว่าจะเป็น ค่าดีเลย์เฉลี่ยในการส่งข้อมูลของโครงข่าย ค่าใช้จ่ายของโครงข่าย เป็นต้น ดังนั้นการออกแบบโครงข่ายที่ดีนั้นจะต้องทำให้สูญเสียค่าใช้จ่ายของโครงข่ายน้อยที่สุดเท่าที่จะทำได้โดยที่ประสิทธิภาพในการส่งข้อมูลของโครงข่ายนั้นก็มีประสิทธิภาพสูงเป็นไปตามที่ต้องการ อีกทั้งยังต้องการความรวดเร็วในการออกแบบ ซึ่งถ้าหากใช้คนในการทำการออกแบบโครงข่ายก็จะทำให้เสียเวลาในการออกแบบและอาจจะทำให้เกิดความผิดพลาดได้ง่าย ซึ่งในปัจจุบันคอมพิวเตอร์เข้ามามีบทบาทในงานต่างๆของมนุษย์มากมายไม่ว่าจะเป็นในส่วนของงานจัดเก็บข้อมูล งานทางด้านการประมวลผลข้อมูล และอื่นๆอีกมากมาย ดังนั้นสามารถที่จะนำคอมพิวเตอร์มาประยุกต์ใช้ในการออกแบบโครงข่าย ซึ่งจะทำให้มีความสะดวกและรวดเร็วในการออกแบบ

1.2 วัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาหลักการและขั้นตอนการทำงานของกระบวนการทางจีเนติกอัลกอริทึมและการประยุกต์ใช้งาน

2. เพื่อศึกษาแนวทางการนำกระบวนการทางจีเนติกอัลกอริทึมมาประยุกต์ใช้ในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรี โดยใช้คอมพิวเตอร์ช่วยในการออกแบบ

3. เพื่อศึกษากระบวนการในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีและค่าพารามิเตอร์ต่างๆที่มีผลต่อประสิทธิภาพของโครงข่าย

4. เพื่อศึกษาปัญหาที่เกิดขึ้นและแนวทางในการแก้ปัญหาเพื่อจะนำกระบวนการจีเนติกอัลกอริทึมมาประยุกต์ใช้งานให้มีประสิทธิภาพยิ่งขึ้น

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทมากขึ้นเรื่อยๆ ดังนั้นจึงได้นำคอมพิวเตอร์มาประยุกต์ใช้กับงานต่างๆ และได้พยายามพัฒนาคอมพิวเตอร์ให้มีความฉลาดยิ่งขึ้นโดยพยายามให้เครื่องคอมพิวเตอร์หรือโปรแกรมที่ใช้มีการเรียนรู้จดจำและสามารถตัดสินใจเองได้ ซึ่งจีเนติกอัลกอริทึมก็เป็นทฤษฎีหนึ่งที่นำไปประยุกต์ใช้งานในการวิเคราะห์ข้อมูลเพื่อเพิ่มประสิทธิภาพให้กับคอมพิวเตอร์หรือโปรแกรม ซึ่งจีเนติกอัลกอริทึมเป็นทฤษฎีที่เป็นการนำกระบวนการวิวัฒนาการของสิ่งมีชีวิตมาประยุกต์ใช้ในการหารูปแบบคำตอบที่ใกล้เคียงหรือที่ดีที่สุดของปัญหาจากจำนวนคำตอบที่เป็นไปได้ทั้งหมดของการแก้ปัญหานั้นๆ โดยแต่ละคำตอบจะเรียกว่าโครโมโซม จีเนติกอัลกอริทึมเป็นวิธีการหาคำตอบโดยพิจารณาและดำเนินการจากกลุ่มคำตอบของปัญหาที่ถูกสร้างขึ้นโดยการเข้ารหัสหรือแปลงค่าตัวแปรพารามิเตอร์ต่างๆของปัญหาให้อยู่ในรูปโครงสร้างของโครโมโซมตามที่กำหนดและทำการคัดเลือกโครโมโซมคำตอบที่เหมาะสมสำหรับสร้างวิวัฒนาการของคำตอบให้ดีขึ้นตามกระบวนการทางพันธุศาสตร์ไม่ว่าจะเป็นการแลกเปลี่ยนค่าพารามิเตอร์ต่างๆหรือเปลี่ยนแปลงบางส่วนในโครโมโซมของโครโมโซมที่ถูกคัดเลือกซึ่งจะทำให้คำตอบนั้นถูกปรับปรุงให้ดีขึ้นเรื่อยๆจนกระทั่งได้คำตอบที่ดีที่สุด ดังนั้นในการใช้จีเนติกมาประยุกต์ใช้ในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีเพื่อที่จะหารูปแบบโครงข่ายที่เหมาะสมหรือดีที่สุดโดยวิเคราะห์ค่าความเหมาะสมตามที่ได้กำหนดไว้ ซึ่งจะทำให้สามารถหารูปแบบของโครงข่ายที่เหมาะสมที่สุดตามที่ต้องการได้

1.4 ขอบเขตและขั้นตอนการวิจัย

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์ที่จะศึกษาวิจัยแนวทางในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีโดยประยุกต์ใช้กระบวนการทางจีเนติกอัลกอริทึม โดยกระบวนการจีเนติกอัลกอริทึมนั้นจะช่วยเพิ่มประสิทธิภาพและความรวดเร็วในการออกแบบด้วยคอมพิวเตอร์หรือโปรแกรม โดยในการศึกษาวิจัยจะต้องทำการศึกษาการออกแบบโครงข่าย ATM ที่มีโครงสร้างเป็นแบบทรีและพารามิเตอร์ต่างๆที่ใช้ในการพิจารณาประสิทธิภาพของโครงข่ายไม่ว่าจะเป็นค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายและค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายและทำการศึกษาทฤษฎีเกี่ยวกับจีเนติกอัลกอริทึมว่ามีลักษณะเป็นแบบใด มีกระบวนการอย่างไรบ้างและการประยุกต์ใช้งานต่างๆรวมไปถึงข้อดีข้อเสีย จากนั้นนำกระบวนการทางจีเนติกนี้มาประยุกต์ใช้สำหรับการออกแบบโครงข่าย ATM ที่มีโครงสร้างเป็นแบบทรี โดยทำการเขียนโปรแกรมคอมพิวเตอร์สำหรับใช้ในการออกแบบโครงข่ายและทำการทดลองออกแบบโครงข่ายเพื่อที่จะให้คอมพิวเตอร์สามารถออกแบบโครงข่ายที่มีความเหมาะสมที่สุดตามที่ต้องการได้

1.5 โครงสร้างของวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้เสนอกระบวนการในการออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึมโดยเนื้อหาแบ่งออกเป็น

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการศึกษา ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย ขอบเขตการวิจัยและขั้นตอนการศึกษา

บทที่ 2 หลักการพื้นฐานสำหรับการวิเคราะห์ระบบโครงข่าย กล่าวถึง การสื่อสารข้อมูลแบบ ATM เบื้องต้น ทฤษฎีทั่วไปที่ใช้ในการวิเคราะห์โครงข่าย แบบจำลองคิวของระบบโครงข่าย การวิเคราะห์ประสิทธิภาพของโครงข่าย

บทที่ 3 ความรู้พื้นฐานเกี่ยวกับจีเนติกอัลกอริทึม กล่าวถึงทฤษฎีพื้นฐานต่างๆเกี่ยวกับจีเนติกอัลกอริทึม กระบวนการ และการประยุกต์ใช้จีเนติกอัลกอริทึม

บทที่ 4 การนำจีเนติกอัลกอริทึมมาใช้ในการออกแบบโครงข่าย ATM กล่าวถึง โมเดลของระบบ การใช้จีเนติกอัลกอริทึมในการออกแบบโครงข่าย ATM และการทดสอบการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรี

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ กล่าวถึงสรุปผลการวิจัยการออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึมและข้อเสนอแนะต่างๆ

บทที่ 2
หลักการพื้นฐานสำหรับการวิเคราะห์ระบบโครงข่าย

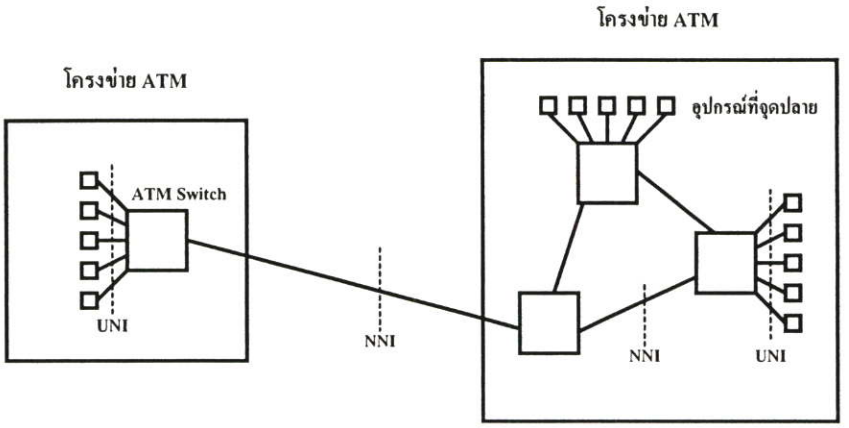
2.1 การสื่อสารข้อมูลแบบ ATM เบื้องต้น

Asynchronous Transfer Mode (ATM) เป็นเทคโนโลยีการส่งข้อมูลที่ได้รับการออกแบบมาเพื่อให้สามารถรองรับการสื่อสารข้อมูลชนิดต่างๆ เช่น เสียง (voice) ภาพ (image) วิดีโอ (video) ข้อมูลประเภทดาต้า(data) และ มัลติมีเดีย (multimedia) ภายในโครงข่ายเดียวกันได้ หลักการของ ATM อาศัยการแบ่งข้อมูลที่จะส่งออกเป็นเซลล์(cell)เล็กๆขนาด 53 ไบต์ โดยมีลักษณะการส่งคล้ายกับ packet switched networks การที่กำหนดให้เซลล์มีขนาดเล็กเพื่อที่จะให้การส่งข้อมูลมีดีเลย์(delay)น้อย ซึ่งหมายความว่าโครงข่ายสามารถใช้รองรับการส่งข้อมูลประเภท real time เช่น สัญญาณเสียงได้ และการกำหนดให้เซลล์มีขนาดคงที่ก็เพื่อที่จะทำให้การออกแบบสวิตช์ง่ายขึ้น ซึ่งช่วยให้การสร้างระบบโครงข่ายสามารถใช้ฮาร์ดแวร์(hardware) เป็นองค์ประกอบหลักได้ ปัจจัยเหล่านี้ทำให้โครงข่าย ATM มีขีดความสามารถในการรับส่งข้อมูลความเร็วสูงมากๆได้โดยที่ยังสามารถควบคุมขนาดของดีเลย์ให้อยู่ในระดับที่เหมาะสม

2.1.1 องค์ประกอบของโครงข่าย ATM

โครงข่าย ATM ประกอบด้วย 2 ส่วนหลักคือ ส่วนของสวิตช์(switch) และ อุปกรณ์ที่จุดปลาย(endpoint) สวิตช์นั้นทำหน้าที่เชื่อมต่ออุปกรณ์ที่จุดปลายหลายๆชุดเข้าด้วยกัน ซึ่งสำหรับในระบบโครงข่ายขนาดเล็กๆแล้ว อาจใช้สวิตช์เพียงตัวเดียวในการเชื่อมต่ออุปกรณ์ที่จุดปลายทั้งหมดได้ แต่ถ้าหากเป็นโครงข่ายที่มีขนาดใหญ่ขึ้นจำเป็นจะต้องมีการใช้สวิตช์หลายๆตัวมากขึ้น โดยที่สวิตช์เหล่านี้ก็จะต่อเชื่อมถึงกันเป็นโครงสร้างรูปแบบๆหนึ่ง ส่วนตัวอย่างของอุปกรณ์ที่จุดปลายที่ต่อเชื่อมอยู่กับสวิตช์ ATM ก็มีอยู่มากมายหลายชนิด เช่น เครื่องคอมพิวเตอร์ประเภทต่างๆ PCs workstations และ supercomputer กล้องวิดีโอ โมเด็ม เป็นต้น

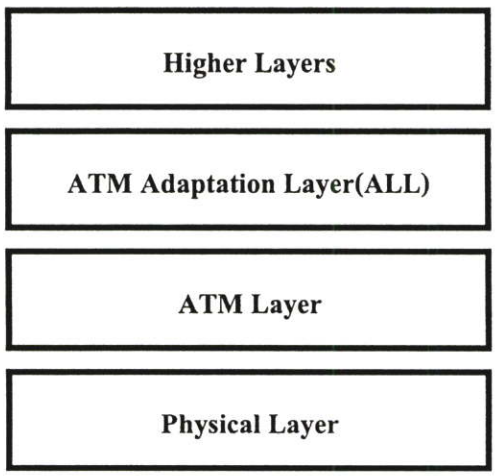
มาตรฐานการเชื่อมต่อของอุปกรณ์สื่อสารในระบบ ATM สามารถแบ่งออกได้เป็น 2 กลุ่ม คือ user-network interface (UNI) และ network-node interface (NNI) พิจารณารูปที่ 2.1 เป็นตัวอย่างแสดงรูปแบบการเชื่อมต่อแบบ UNI และ NNI ภายในโครงข่าย ATM โดย UNI เป็นส่วนของการเชื่อมต่อระหว่างอุปกรณ์จุดปลายกับสวิตช์ ATM หรือก็คือจุดต่อระหว่างผู้ใช้กับโครงข่าย ATM นั่นเอง ส่วน NNI ก็คือการเชื่อมต่อระหว่างโครงข่าย ATM 2 โครงข่าย หรือ ระหว่างอุปกรณ์สวิตช์ ATM 2 ชุด จุดเชื่อมทั้งสองแบบมีหน้าที่แตกต่างกัน ส่วนของ UNI มีหน้าที่ติดต่อกับผู้ใช้บริการในช่วงการขอเริ่มการติดต่อสื่อสาร ช่วงที่มีการรับส่งข้อมูลและสิ้นสุดการติดต่อสื่อสาร ส่วนจุดเชื่อมต่อ NNI ทำหน้าที่รับส่งข้อมูลทั้งในส่วนของผู้ใช้และสัญญาณควบคุมระหว่างสวิตช์ที่ติดกันหรือระหว่างโครงข่าย



รูปที่ 2.1 แสดงมาตรฐานการเชื่อมต่อ UNI และ NNI ภายในโครงข่าย ATM

2.1.2 การแบ่งชั้นสื่อสารใน ATM

ATM จะมีรูปแบบมาตรฐานหรือการแบ่งชั้นสื่อสารเป็นของตนเองดังแสดงให้เห็นในรูปที่ 2.2 โดยรูปแบบดังกล่าวจะมีชั้นหลักๆ อยู่ 4 ชั้นด้วยกันคือ ชั้นสื่อสารกายภาพ (physical layer) ชั้นสื่อสาร ATM (ATM layer) ชั้นปรับเปลี่ยนการใช้งาน (AAL: ATM adaptation layer) และชั้น Higher layer

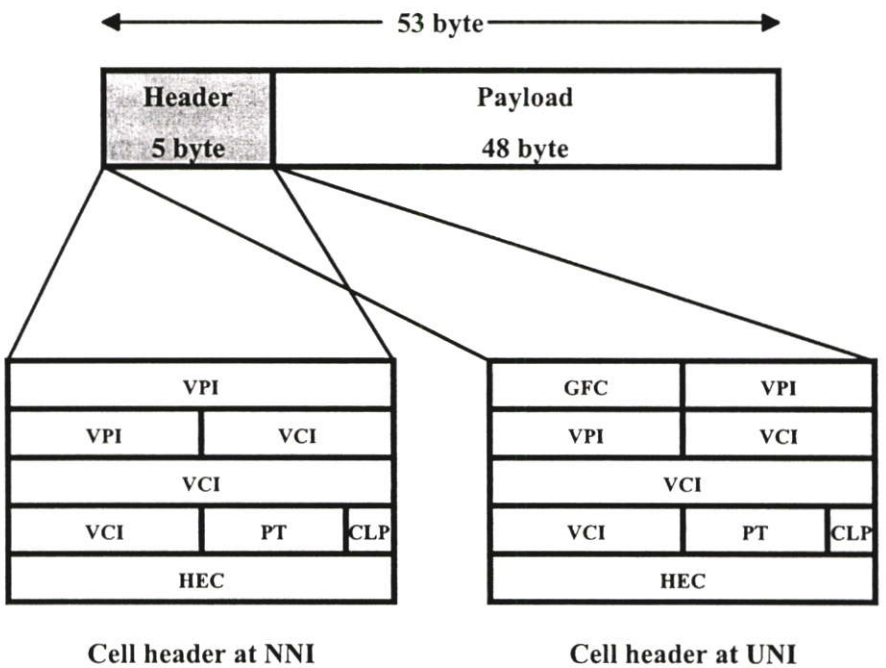


รูปที่ 2.2 โครงสร้างชั้นสื่อสารของ ATM

ชั้นสื่อสารกายภาพ (Physical Layer) เป็นชั้นที่กำหนดรายละเอียดในการติดต่อกับสายสื่อสารและอุปกรณ์รับส่งข้อมูล โดยในชั้นนี้จะมีชั้นสื่อสารย่อยอยู่ 2 ส่วนคือ PMD (Physical Medium Dependent) ในชั้นย่อยนี้จะเป็นส่วนที่เกี่ยวข้องกับการกำหนดชนิดของสื่อกลางที่ใช้ในการส่งข้อมูล เช่น อาจจะเป็น UTP cable, Coaxial cable , Fiber optic cable หรือ Wireless เป็นต้น และในแต่ละชนิดของสื่อกลางที่ใช้นั้น ชั้นย่อย PMD ก็ยังระบุถึงรายละเอียดของวิธีการส่งบิทข้อมูลที่เหมาะสมกับสื่อกลางชนิดนั้นๆด้วย เช่น bit transfer และ bit alignment นอกจากนี้ยังจะ

รวมไปถึงวิธีที่ใช้สำหรับการทำ line coding , opto-electronic conversion , modulation และ demodulation จะเห็นได้ว่าชั้น PMD เป็นส่วนที่เกี่ยวข้องกับโดยตรงกับสื่อกลางที่ใช้โดยจะกำหนดรายระเอียดของการรับส่งบิทข้อมูลที่แตกต่างกันสำหรับสื่อกลางแต่ละชนิด ส่วนชั้นย่อย TC (Transmission Convergence) จะไม่สนใจกับลักษณะการส่งหรือชนิดของสื่อกลาง แต่จะทำหน้าที่จัดการกับเซลล์ที่มาจาก ATM layer ให้สามารถส่งผ่านสื่อกลางนั้นๆได้ดังนั้นจะเห็นว่าชั้นย่อยนี้เองที่ทำให้การทำงานของ ATM layer ไม่ขึ้นอยู่กับชนิดของสื่อกลางที่ใช้เลย

ชั้นสื่อสาร ATM (ATM Layer) จะกำหนดรายละเอียดของเซลล์และการนำส่งเซลล์ กำหนดรูปแบบของเซลล์และความหมายของข้อมูลส่วนหัว รวมทั้งการติดต่อเพื่อเริ่มต้นการสื่อสารและการสิ้นสุดการสื่อสาร การควมคุมความคับคั่งของข้อมูล (congestion control) ก็เป็นส่วนหนึ่งของโปรโตคอลในชั้นนี้ โดยโครงสร้างของเซลล์ ATM ประกอบด้วย 2 ส่วนคือ ส่วนของ(Header) มีความยาวเท่ากับ 5 byte และส่วนของข้อมูล(Payload) มีความยาว 48 byte ดังแสดงในรูปที่ 2.3 โดยในส่วนหัวของเซลล์นั้นจะแตกต่างกัน 2 แบบทั้งนี้ขึ้นอยู่กับลักษณะและจุดหรือบริเวณการใช้งานว่าเป็นกรณีของ UNI หรือ NNI โดยแต่ละส่วนของส่วนหัวของเซลล์นั้นมีดังต่อไปนี้



รูปที่ 2.3 โครงสร้างของเซลล์ ATM

- Generic Flow Control (GFC) ใช้ดำเนินการควบคุมการไหลของข้อมูลที่อุปกรณ์ของผู้ใช้ไปยังโครงข่าย โดย GFC จะไม่สามารถควบคุมการไหลของข้อมูลในทิศทางอื่นๆ ยกตัวอย่างเช่น จากโครงข่ายไปยังผู้ใช้ ดังนั้นส่วนของ GFC จะไม่ได้ใช้ในโครงข่าย NNI จึงนำส่วนนี้เป็นของ VPI เพื่อเพิ่มความสามารถในของการเลือกเส้นทาง

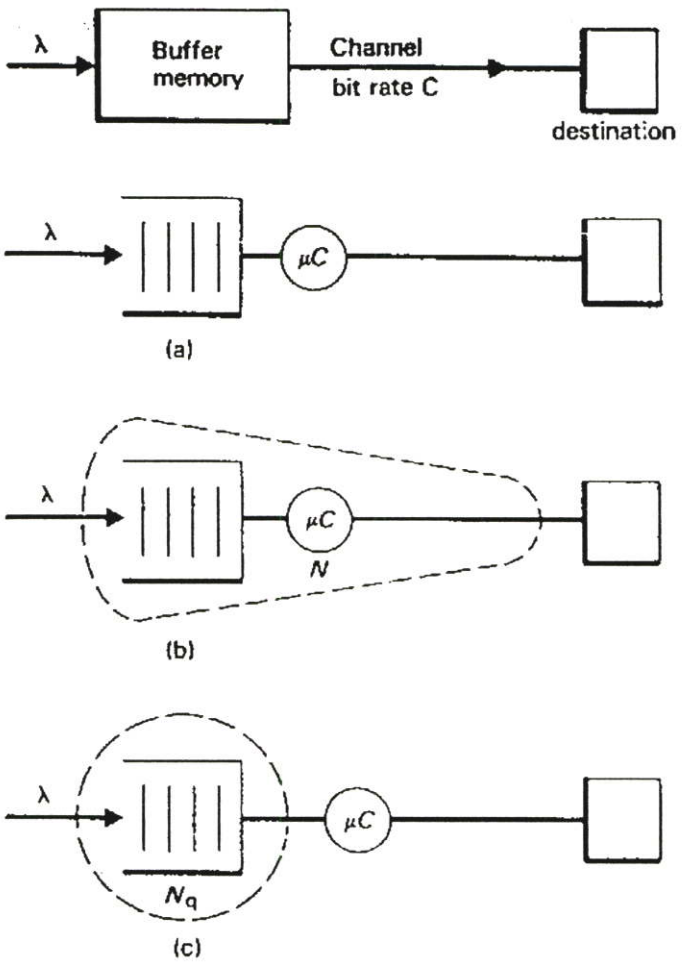
- Virtual Channel Identifier (VCI) เป็นค่าที่กำหนดใช้สำหรับการเชื่อมต่อสื่อสารของผู้ใช้โดยแต่ละการเชื่อมต่อก็จะได้ค่า VCI ที่แตกต่างกันไป
- Virtual Path Identifier (VPI) เป็นค่าที่ใช้งานในลักษณะที่คล้ายกันกับ VCI หากแต่ VPI ค่าๆหนึ่งใช้ในการระบุถึงกลุ่มของ VCI จำนวนหนึ่ง
- Payload Type (PT) เป็นตัวบ่งบอกว่าเซลล์นี้เป็นเซลล์ประเภทใดคือเป็นเซลล์ข้อมูลของผู้ใช้หรือเป็นเซลล์ที่บรรจุสัญญาณควบคุมเพื่อใช้ในการจัดการภายในโครงข่าย
- Cell Loss Priority (CLP) มีความยาวเพียงหนึ่งบิท ถ้าบิทนี้ถูกเซตเป็น 1 เซลล์เหล่านี้จะถือว่าเป็นเซลล์ที่มีความสำคัญไม่มากนัก ส่วนของโครงข่ายสามารถที่จะทิ้งเซลล์เหล่านี้หากเกิดความคับคั่งขึ้นในระบบ แต่ถ้าค่าบิทนี้ถูกเซตเป็น 0 เซลล์เหล่านี้ถือว่ามีความสำคัญและโครงข่ายจะต้องพยายามส่งผ่านเซลล์เหล่านี้ให้สำเร็จ
- Header Error Control (HEC) มีหน้าที่ในการตรวจสอบความผิดพลาดของบิทเฉพาะในส่วนหัวของเซลล์เกิดขึ้นหรือไม่

ชั้นปรับเปลี่ยนการใช้งาน (AAL: ATM adaptation layer) แบ่งออกเป็นสองชั้นสื่อสารย่อย คือ ชั้น SAR (Segmentation And Reassembly) และ ชั้น CS (Convergence Sublayer) โดยชั้น SAR เป็นชั้นบนทำหน้าที่ในการแบ่งแพ็กเกจข้อมูลออกเป็นเซลล์ในทางด้านผู้ส่งส่วนในทางด้านผู้รับก็จะนำเซลล์ทั้งหมดมาประกอบกลับเป็นแพ็กเกจอย่างเดิม ส่วนชั้น CS มีหน้าที่ช่วยในการแยกแยะประเภทของการให้บริการตามลักษณะที่โปรแกรมประยุกต์ต้องการ เช่น บริการจัดการแฟ้มข้อมูลต้องการความถูกต้องของข้อมูลร้อยเปอร์เซ็นต์เต็ม ในขณะที่บริการภาพเคลื่อนไหวนั้นสนใจความต่อเนื่องของข้อมูลมากกว่าการแก้ไขข้อผิดพลาดที่เกิดขึ้นเล็กน้อยในระหว่างการสื่อสาร

2.2 แบบจำลองคิวของระบบโครงข่าย

2.2.1 การส่งข้อมูลในโครงข่าย

พิจารณาถึงคุณสมบัติของโครงข่ายที่มีการส่งและการรับข้อมูลที่ประกอบด้วยอุปกรณ์จัดเก็บหรือบัพเฟอร์หนึ่งตัวและหนึ่งช่องสัญญาณการส่งออกด้วยอัตราการส่ง C บิทต่อวินาที ระบบดังกล่าวแสดงไว้ในรูปที่ 2.4



รูปที่ 2.4 แสดงโครงข่ายที่มีหนึ่งบัพเฟอร์และหนึ่งช่องสัญญาณการให้บริการ

(a) แสดงรูปแบบการเชื่อมต่อ

(b) แสดงระบบซึ่งพิจารณาถึงหน่วยบัพเฟอร์และช่องสัญญาณ

(c) แสดงระบบซึ่งพิจารณาถึงหน่วยบัพเฟอร์

กำหนดให้อัตราเฉลี่ยการรับข้อมูล(average input rate) มีค่าเป็น λ ข้อมูลต่อวินาทีและความยาวเฉลี่ยของข้อมูล(average length of the message) มีค่าเป็น $1/\mu$ บิทต่อข้อมูล

จากรูป 2.4-a จะพิจารณาว่าสำหรับข้อมูลใดๆจะถูกพิจารณาว่ายังอยู่ในโครงข่ายจนกระทั่งบิทสุดท้ายของข้อมูลนั้นถูกส่งออกจากโครงข่ายเรียบร้อยแล้ว ดังนั้นค่าดีเลย์ที่น้อยที่สุดที่เป็นไปได้ในการส่งข้อมูลหนึ่งข้อมูลคือเวลาที่ใช้ในการส่งข้อมูลนั้นไปยังช่องสัญญาณจนเสร็จสิ้น เนื่องจากค่าความยาวเฉลี่ยของข้อมูลมีค่า$1/\mu$ บิท และอัตราการส่งข้อมูลในช่องสัญญาณ(channel bit rate) มีค่าC บิทต่อวินาที จะได้ว่าค่าเฉลี่ยของเวลาที่ใช้ในการส่ง(average transmission หรือ processing time) มีค่า $1/\mu C$ วินาที

ในระบบ steady state นั้นอัตราเฉลี่ยการมาถึงของข้อมูลและอัตราเฉลี่ยการส่งข้อมูลมีค่าเท่ากันคือ λ ข้อมูลต่อวินาที เมื่อมีข้อมูลในช่องสัญญาณข้อมูลจะถูกส่งออกที่อัตราเฉลี่ย μC ข้อมูลต่อวินาที ผลต่างของ λ และ μC คือการแสดงให้เห็นถึงสถานะของช่องสัญญาณหรือความหนาแน่นของช่องสัญญาณ (Traffic Intensity) ρ โดยความสัมพันธ์ระหว่าง λ และ μC จะได้ดังนี้

$$\lambda = \rho\mu C \tag{2.1}$$

หรือ

$$\rho = \lambda / \mu C \tag{2.2}$$

จากสมการจะเห็นว่า ค่าความหนาแน่นของช่องสัญญาณจะเท่ากับค่าเฉลี่ยอัตราการมาถึงหารด้วยค่าอัตราเฉลี่ยการดำเนินการของช่องสัญญาณ ในกรณีที่ระบบโครงข่ายไม่เสถียร ค่าอัตราเฉลี่ยการมาถึงจะต้องมีค่าน้อยกว่าหรือเท่ากับ μC ซึ่งสอดคล้องกับสมการ 2.2 และค่า ρ ควรจะมีค่าน้อยกว่าหรือเท่ากับ 1

กำหนดให้ค่าเฉลี่ยของจำนวนข้อมูลที่ถูกเก็บไว้ในระบบโครงข่ายทั้งในบัพเฟอร์และช่องสัญญาณมีค่า N โดยกำหนดให้ N_q คือค่าเฉลี่ยจำนวนของข้อมูลที่ถูกเก็บไว้ในบัพเฟอร์ และเวลาเฉลี่ยในการให้บริการของบัพเฟอร์มีค่า W เมื่อค่าเวลาเฉลี่ยในการให้บริการของระบบมีค่า T

จาก Little's Law เมื่อพิจารณาทั้งระบบโครงข่าย

$$N = \lambda T \tag{2.3}$$

พิจารณาระบบในรูปที่ 2.4-b และ 2.4-c นำ Little's Law มาประยุกต์ใช้ในส่วนของบัพเฟอร์จะได้ว่า

$$N_q = \lambda W \tag{2.4}$$

ค่าดีเลย์เฉลี่ย T และ W มีความสัมพันธ์กันโดยค่าดีเลย์เฉลี่ยของทั้งบัพเฟอร์และช่องสัญญาณคือผลรวมระหว่างค่าดีเลย์เฉลี่ยในบัพเฟอร์และค่าเวลาเฉลี่ยในการส่งข้อมูล $(1/\mu C)$ ดังนั้น จะได้ว่า

$$T = W + 1/\mu C \tag{2.5}$$

พิจารณาความสัมพันธ์ระหว่าง N และ N_q โดยนำ λ มาคูณทั้งสองข้างของสมการที่ 2.5 และนำสมการที่ 2.3 และ สมการที่ 2.4 มาแทนค่า จะได้ว่า

$$N = N_q + \rho \tag{2.6}$$

จากสมการที่ 2.6 แสดงให้เห็นว่า ค่าเฉลี่ยจำนวนของข้อมูลที่ถูกเก็บไว้ในโครงข่ายมีค่าเท่ากับค่าเฉลี่ยจำนวนข้อมูลที่ถูกเก็บไว้ในบัพเฟอร์บวกกับค่า ρ ซึ่งค่า ρ นี้จะแสดงค่าเฉลี่ยจำนวนข้อมูลที่มีอยู่ในช่องสัญญาณ

2.2.2 แบบจำลองคิวประเภทต่างๆ

ในหัวข้อนี้จะกล่าวถึงระบบโครงข่ายอย่างง่ายซึ่งประกอบด้วยหนึ่งบัพเฟอร์และหนึ่งช่องสัญญาณดังแสดงในรูปที่ 2.4 ขั้นตอนการมาถึง(Arrival process) ถูกออกแบบให้เป็นกระบวนการแบบปัวซ์ซอง(Poisson process) สำหรับขั้นตอนการให้บริการจะมีอยู่สองชนิดคือพิจารณาความสัมพันธ์ระหว่างความหนาแน่นของช่องสัญญาณและจำนวนเฉลี่ยของข้อมูลในโครงข่าย หรือความสัมพันธ์ระหว่างความหนาแน่นของช่องสัญญาณและค่าดีเลย์เฉลี่ยของโครงข่าย

ในทฤษฎีระบบคิวนั้นมีการกำหนดสัญลักษณ์ขึ้นเพื่อแสดงถึงรูปแบบการมาถึง (Arrival process) รูปแบบของเวลาในการให้บริการ (message processing time) และจำนวนช่องสัญญาณที่ให้บริการ(number of output channel) ตัวอย่างเช่น ตัวอย่างเช่น A/B/C โดย A คือรูปแบบการมาถึงของข้อมูล B คือ รูปแบบของเวลาในการให้บริการ และ C คือจำนวนของช่องสัญญาณที่ให้บริการ ซึ่งในหัวข้อนี้จะยกตัวอย่างของแบบจำลองคิวแบบ M/M/1 และ M/D/1

แบบจำลองคิว M/M/1

แบบจำลองคิวแบบ M/M/1 มีรูปแบบการมาถึงของข้อมูลและรูปแบบของเวลาในการให้บริการข้อมูลเป็นแบบปัวซ์ซองซึ่งมีการกระจายแบบเอ็กโปรเน็ลเชียลและมีหนึ่งช่องสัญญาณการให้บริการ สามารถพิจารณาค่าต่างๆได้ดังนี้

$$N = \frac{\rho}{1-\rho} \tag{2.7}$$

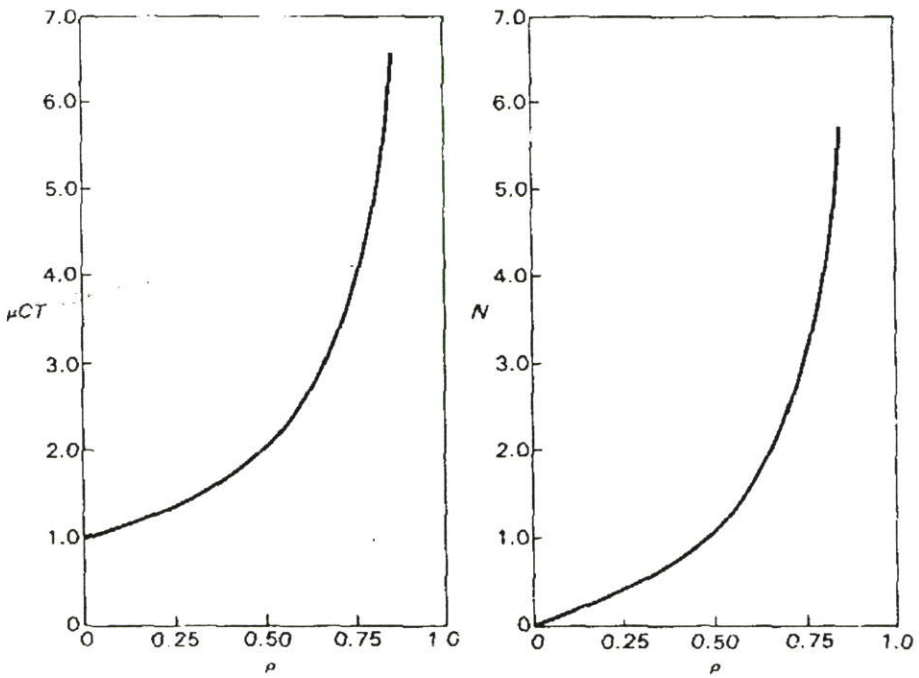
ค่าดีเลย์เฉลี่ยสำหรับข้อมูลใดๆในแบบจำลองคิว M/M/1 จะเกี่ยวข้องกับ N โดยใช้ Little's Law จะได้ว่า

$$\lambda T = N = \frac{\rho}{1-\rho}$$

ดังนั้นค่าดีเลย์เฉลี่ยของระบบคือ

$$T = \frac{1}{\lambda}\left(\frac{\rho}{1-\rho}\right) = \frac{1}{\mu C(1-\rho)} \tag{2.8}$$

โดยกราฟของ μCT และ N เทียบกับ ρ แสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 แสดงกราฟของ μCT และ N เทียบกับ ρ

พิจารณาค่าเฉลี่ยจำนวนของข้อมูลที่ถูกเก็บไว้ในบัพเฟอร์(N_q) และ เวลาเฉลี่ยในการให้บริการของบัพเฟอร์(W) สามารถหาได้จากสมการที่ 2.6 และ 2.5 ตามลำดับดังนี้

$$N_q = N - \rho$$

และ

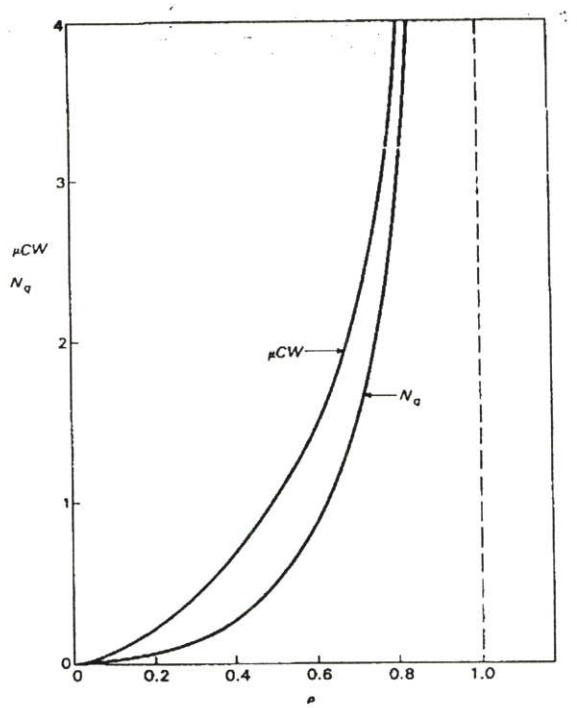
$$W = T - 1/\mu C$$

จากสมการที่ 2.7 สำหรับค่า N และ จากสมการที่ 2.8 สำหรับค่า T จะสามารถแสดง N_q และ W ในรูปของ ρ และ μC ได้ดังนี้

$$N_q = \frac{\rho^2}{1-\rho} \tag{2.9}$$

$$W = \frac{\rho}{\mu C(1-\rho)} \tag{2.10}$$

ค่าของ N_q และ μCW สามารถเขียนกราฟได้ดังรูปที่ 2.6



รูปที่ 2.6 แสดงกราฟของ μCW และ N_q เทียบกับ ρ สำหรับแบบจำลองคิว M/M/1

แบบจำลองคิว M/D/1

แบบจำลองคิว M/D/1 แตกต่างจาก M/M/1 ในส่วนรูปแบบของเวลาในการให้บริการข้อมูล มีค่าคงที่เท่ากับ $1/\mu C$ โดยจะได้ว่า

$$N = \rho + \frac{\rho^2}{2(1-\rho)} = \frac{\rho(2-\rho)}{2(1-\rho)} \tag{2.11}$$

และ

$$T = \frac{1}{\mu C} + \frac{\rho}{2\mu C(1-\rho)} = \frac{2-\rho}{2\mu C(1-\rho)} \tag{2.12}$$

ดังนั้นสามารถหาค่า N_q และ W ได้ดังนี้

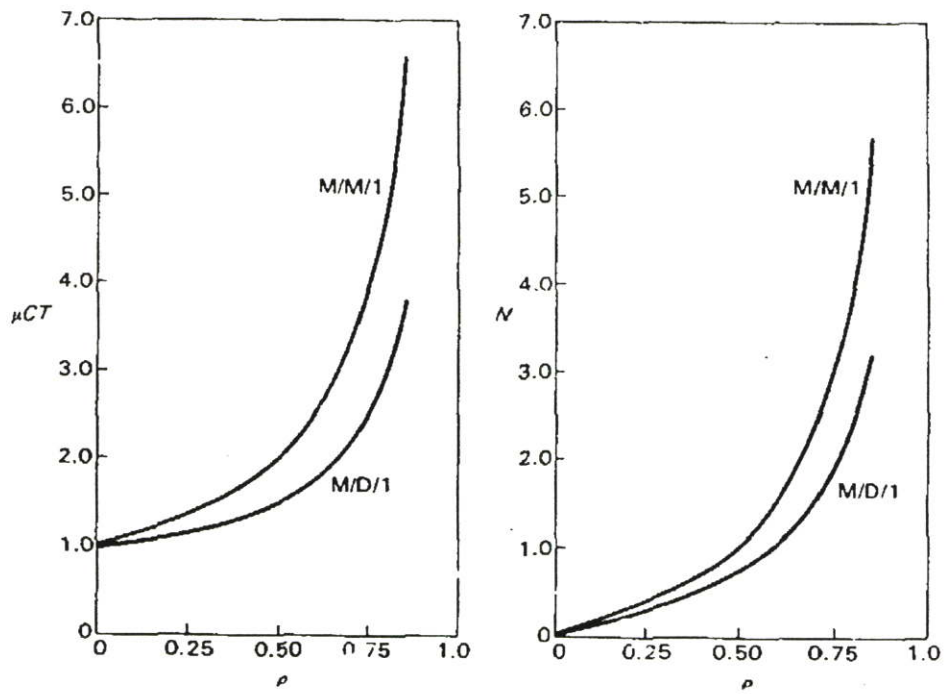
$$N_q = \frac{\rho^2}{2(1-\rho)} \tag{2.13}$$

และ

$$W = \frac{\rho}{2\mu C(1-\rho)} \tag{2.14}$$

จะสังเกตได้ว่าค่าเฉลี่ยจำนวนของข้อมูลที่ถูกเก็บไว้ในบัฟเฟอร์(N_q) และค่าของเวลาเฉลี่ยในการให้บริการของบัฟเฟอร์(W) สำหรับ M/D/1 จะมีค่าเป็นครึ่งหนึ่งของ M/M/1

โดยกราฟของ μCT และ N เทียบกับ ρ สำหรับแบบจำลองคิว M/M/1 เปรียบเทียบกับ M/D/1 สามารถแสดงได้ดังรูปที่ 2.7



รูปที่ 2.7 กราฟของ μCT และ N เทียบกับ ρ สำหรับแบบจำลองคิว M/M/1 เปรียบเทียบกับ M/D/1

2.3 การวิเคราะห์ประสิทธิภาพของระบบโครงข่าย

2.3.1 การวิเคราะห์ค่าดีเลย์เฉลี่ยของโครงข่าย

พิจารณาโครงข่ายข้อมูลซึ่งจะมีสวิตช์หรือโหนดเชื่อมต่อกันด้วยสายสัญญาณเพื่อใช้ในการส่งแพ็กเกจข้อมูล โดยค่าดีเลย์เฉลี่ยของแพ็กเกจในโครงข่ายมีค่า

$$T = \sum_{j=1}^{N}\sum_{k=1}^{N}\frac{\gamma_{jk}}{\gamma}Z_{jk} \tag{2.15}$$

เมื่อ γ_{jk} คือ อัตราเฉลี่ยของแพ็กเกจจากโหนด j ไปยังโหนด k

Z_{jk} คือ ค่าดีเลย์เฉลี่ยของแพ็กเกจจากโหนด j ไปยังโหนด k

โดยค่าอัตราเฉลี่ยของแพ็กเกจทั้งหมดที่เข้าไปในโครงข่ายมีค่า

$$\gamma = \sum_{j=1}^{N}\sum_{k=1}^{N}\gamma_{jk} \tag{2.16}$$

เมื่อพิจารณาที่ช่องสัญญาณ i ที่ทำการเชื่อมต่อระหว่างโหนด โดยมีความจุของช่องสัญญาณ (C_i)และมีเส้นทางการเชื่อมต่อของโหนด j และ k (π_{jk}) จะได้อัตราเฉลี่ยของแพ็กเกจในช่องสัญญาณ i มีค่า

$$\lambda_i = \sum_{\substack{j=1\\ j,k:C_i\in\pi_{jk}}}^{N}\sum_{k=1}^{N}\gamma_{jk} \tag{2.17}$$

และ

$$Z_{jk} = \sum_{i:C_i\in\pi_{jk}} T_i \tag{2.18}$$

ดังนั้นจากสมการ (2.15) จะได้ว่า

$$T = \sum_{i=1}^{M}\frac{\lambda_i}{\gamma}T_i \tag{2.19}$$

โดย M คือ จำนวนช่องสัญญาณ

T_i คือ ค่าดีเลย์เฉลี่ยที่ช่องสัญญาณ i

สำหรับการพิจารณาการส่งข้อมูลแบบ ATM ที่มีการส่งข้อมูลเป็นแพ็กเกจที่มีขนาดคงที่ นั้นในการวิจัยนี้จะพิจารณาแบบจำลอง M/D/1 โดยจะได้ค่าดีเลย์เฉลี่ยที่ช่องสัญญาณ i ดังนี้

$$T_i = \frac{2\mu C_i - \lambda_i}{2\mu C_i(\mu C_i - \lambda_i)} \tag{2.20}$$

แทนสมการ (2.20) ลงในสมการ (2.19) จะได้ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายมีค่า

$$T = \frac{1}{\gamma}\sum_{i=1}^{M}\frac{2\mu C_i\lambda_i - \lambda_i^{\,2}}{2\mu C_i(\mu C_i - \lambda_i)} \tag{2.21}$$

โดยค่าพารามิเตอร์ต่างๆในสมการที่ 2.21 มีค่าดังนี้

C_i คือ ความจุของช่องสัญญาณที่ i

λ_i คือ อัตราเฉลี่ยของเซลล์ในช่องสัญญาณที่ i

$1/\mu$ คือ ความยาวของเซลล์มีขนาด 424 บิท(53 ไบต์)

M คือ จำนวนช่องสัญญาณ

2.3.2 การวิเคราะห์ค่าใช้จ่ายในระบบสายสัญญาณของโครงข่าย

ในส่วนของค่าใช้จ่ายในระบบสายสัญญาณของโครงข่าย(Cost) มีค่าดังสมการที่2.22

$$D = \sum_{i=1}^{M} d_i(C_i) \tag{2.22}$$

เมื่อ $d_i(C_i)$ คือ ค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายความจุ C_i ของช่องสัญญาณที่ i

ค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายนั้นอาจมาจากการติดตั้งหรือเช่าสายสัญญาณที่ใช้ในการส่งข้อมูล จะเห็นได้ว่าค่าใช้จ่ายในระบบสายสัญญาณนั้นจะมีค่าขึ้นอยู่กับระยะทางของสายสัญญาณที่ใช้ในการเชื่อมต่อภายในโครงข่ายทั้งหมด ดังนั้นถ้ายิ่งใช้ระยะทางมากเท่าใดก็ยิ่งจะทำให้ค่าใช้จ่ายมีค่าสูงมากขึ้นเท่านั้น และค่าอีกค่าที่มีผลกับค่าใช้จ่ายในระบบสายสัญญาณคือค่าของความจุของช่องสัญญาณที่ใช้ในการเชื่อมต่อภายในโครงข่ายนั้นๆ เช่น ถ้าต้องการเช่าสายสัญญาณที่มีความจุในการส่งข้อมูลมากก็จะทำให้เสียค่าใช้จ่ายสูงกว่าสายสัญญาณที่มีความจุน้อย เป็นต้น ดังนั้นในการหาค่าความเหมาะสมของสายสัญญาณที่ใช้ในการส่งข้อมูลตามที่ต้องการ ถ้าสามารถหาและทำการออกแบบรูปแบบการเชื่อมต่อของสายสัญญาณที่ดีจะทำให้สูญเสียค่าใช้จ่ายในระบบสายสัญญาณน้อยในขณะที่การส่งข้อมูลมีประสิทธิภาพตามความต้องการ

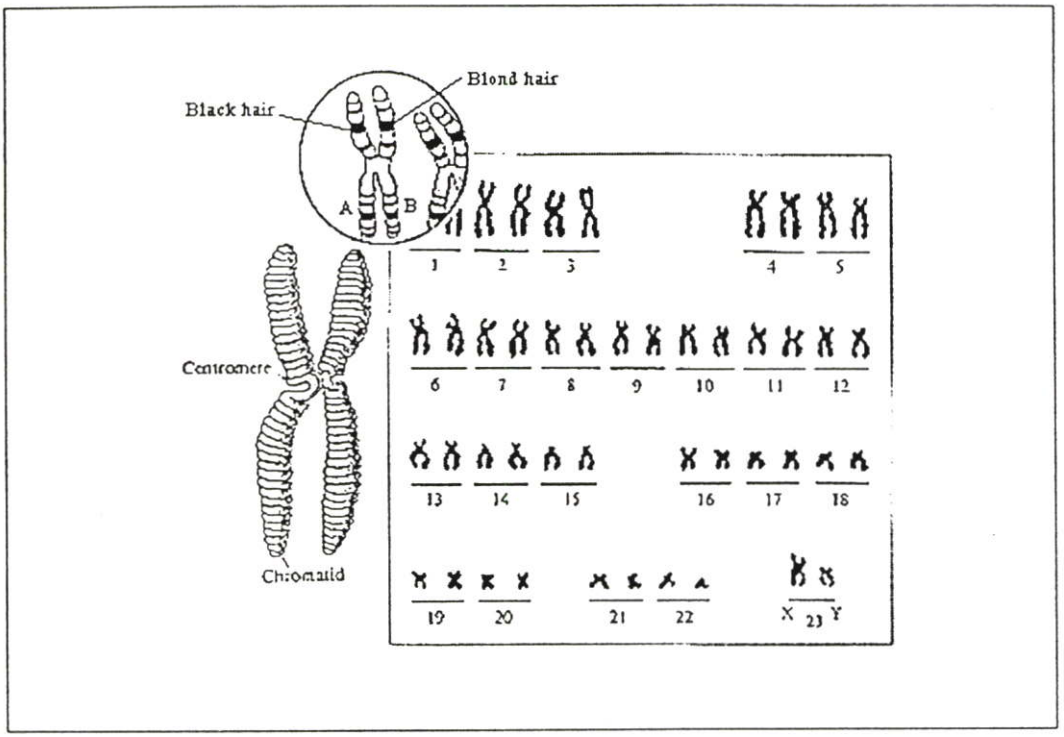
บทที่ 3
ความรู้พื้นฐานเกี่ยวกับจีเนติกอัลกอริทึม

ในปัจจุบันนี้ปัญหาที่ต้องการคำตอบที่สุด (Optimal Solution) ทางวิทยาศาสตร์ วิศวกรรมศาสตร์ คอมพิวเตอร์ หรือการทำงานต่างๆที่เกิดขึ้นมากมายนั้นสามารถหาคำตอบได้หลายวิธีซึ่งแตกต่างกันไปตามชนิดของปัญหา ความคิด เทคนิค วิธีการวิเคราะห์ปัญหานั้นๆ และความแพร่หลายในการพัฒนาศักยภาพของคอมพิวเตอร์ให้รู้จักเรียนรู้เพื่อช่วยหาคำตอบหรือช่วยตัดสินคำตอบในขั้นต้น มาช่วยในการศึกษาวิจัย เช่น นิวรอลเน็ตเวิร์ค(Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) เป็นต้น จีเนติกอัลกอริทึมเป็นอีกวิธีหนึ่งที่จำลองรูปแบบวิธีการทางชีววิทยา ในการให้กำเนิดประชากรรุ่นใหม่หรือขยายเผ่าพันธ์ในรุ่นลูก รุ่นหลานต่อไป ซึ่งอาศัยพื้นฐานความคิดของวิวัฒนาการทางธรรมชาติถ่ายทอดลักษณะต่างๆทางพันธุกรรม โดยปฏิบัติตามกระบวนการทางพันธุศาสตร์ เพื่อจะใช้ในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุดของปัญหาโดยคอมพิวเตอร์

3.1 พันธุศาสตร์ทางชีววิทยา

เมนเดล(Mendel) บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่ายีนส์(Genes) หน่วยเก็บลักษณะทางกรรมพันธ์เป็นตัวกำหนดลักษณะภายนอก ซึ่งยีนส์หลายๆยีนส์จะเรียงตัวกันอยู่บนเส้นโครโมโซม (Chromosome) อีกทีหนึ่งในเซลล์ของสิ่งมีชีวิตโดยจะอยู่กันเป็นคู่ๆ แต่จะแตกต่างกันที่ค่าลักษณะต่างๆ ในแต่ละยีนส์เรียกว่าแอลลี (Allele) ซึ่งแบบต่างๆของยีนส์ที่มีแอลลีต่างกันในแต่ละตำแหน่งยีนส์เดียวกันเรียกว่า ยีโนไทป์ (Genotype) สำหรับลักษณะภายนอกที่ปรากฏออกมาให้เห็นเรียกว่าฟีโนไทป์ (Phenotype) เช่น ในคนจะมีโครโมโซม 23 คู่ 46 โครโมโซม ซึ่งแต่ละโครโมโซมจะประกอบด้วยยีนส์ต่างๆกันราว 1250 ยีนส์ ตัวอย่างคู่โครโมโซมที่ 1 ในคนดังรูปที่ 3.1 ซึ่งประกอบด้วยยีนส์ลักษณะสีผม สีผิว สีตา และอื่นๆ อีกประมาณ 1247 ลักษณะ และโครโมโซม 1A มีแอลลีของยีนส์ลักษณะสีผมคือ ผมสีดำ ส่วนโครโมโซม 1B มีแอลลีของยีนส์ลักษณะสีผมคือ ผมสีบลอนด์

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

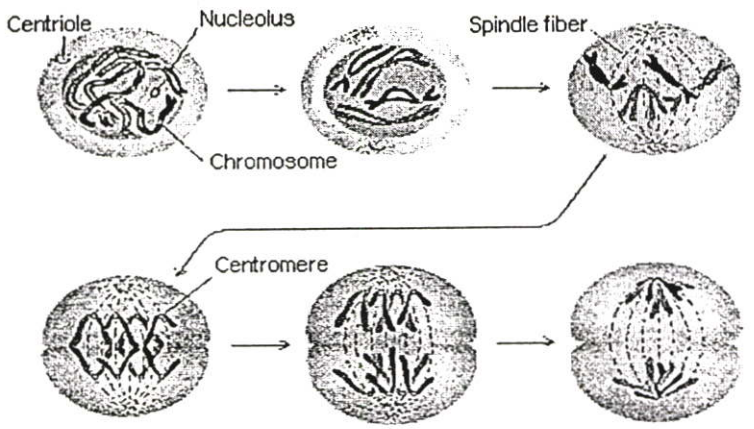


รูปที่3.1 แสดงโครโมโซมทางพันธุศาสตร์

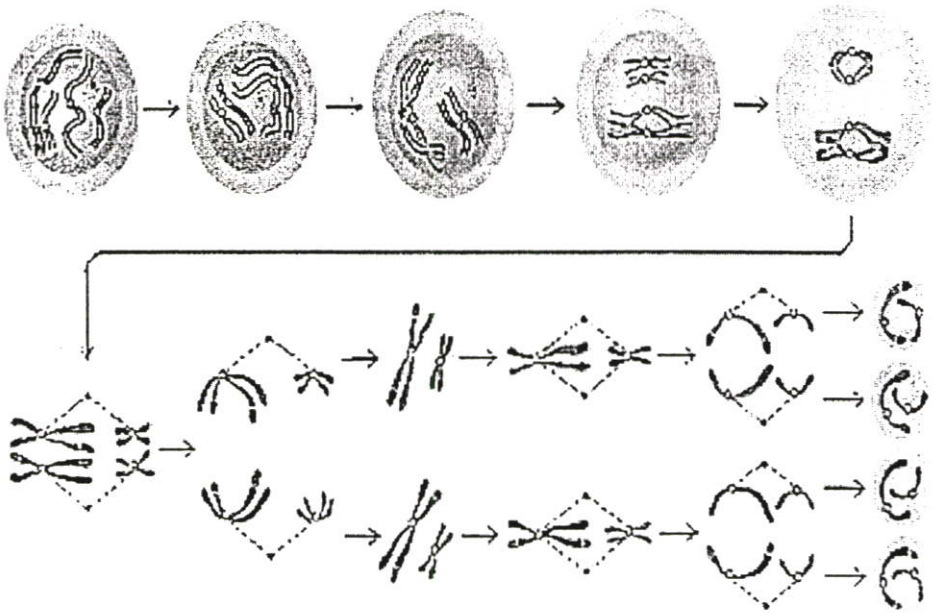
การถ่ายทอดลักษณะทางพันธุกรรมเป็นการถ่ายทอดลักษณะต่างๆของสิ่งมีชีวิตที่เกิดขึ้นเมื่อมีการแบ่งตัวของเซลล์สิ่งมีชีวิต ซึ่งมี 2 แบบ คือ

(1) การเพิ่มจำนวนเซลล์ เป็นการแบ่งตัวแบบไมโตสิส (Mitosis) โดยโครโมโซมแต่ละตัวจะขยายตัวเพิ่มจำนวนตัวเองขึ้นเป็นสอง และเยื่อหุ้มนิวเคลียส (Nucleous) จะสลายลงเพื่อดึงแยกโครโมโซมที่เพิ่ม จำนวนขึ้นออกจากโครโมโซมเดิมเป็นสองด้าน แล้วเยื่อหุ้มนิวเคลียสจะถูกสร้างขึ้นใหม่กลายเป็นเซลล์ใหม่ 2 เซลล์ ที่มีโครโมโซมเหมือนเดิม ดังรูปที่ 3.2-a

(2) การแบ่งตัวของเซลล์สืบพันธ์ เป็นการแบ่งตัวแบบไมโอสิส (Meiosis) โดยโครโมโซมจากเซลล์พ่อ 1 โครโมโซมและโครโมโซมจากเซลล์แม่ 1 โครโมโซม จะเริ่มจับคู่กันที่โครโมโซมชนิดเดียวกัน ในขณะเดียวกันโครโมโซมแต่ละตัวทั้งที่มาจากพ่อและมาจากแม่ ต่างก็จำลองแบบของตนเพิ่มขึ้นมาอีกแต่ละโครโมโซม ทำให้ได้จำนวนโครโมโซมทั้งหมดเพิ่มขึ้นเป็นสองเท่า และดำเนินการทางพันธุกรรมจนถึงระยะแบ่งตัว โครโมโซมพ่อพร้อมกับแบบจำลองและโครโมโซมแม่พร้อมกับแบบจำลองที่ได้จะแยกคู่ไปรวมกันเป็น 2 นิวเคลียส กลายเป็นเซลล์ใหม่ 2 เซลล์ ซึ่งจะแบ่งตัวต่อทันที โดยแต่ละโครโมโซมพ่อแยกตัวออกจากแบบจำลอง และโครโมโซมแม่ก็แยกตัวออกจากแบบจำลองรวมกันเป็นเซลล์ใหม่ 4 เซลล์ ดังรูปที่3.2-b



3.2-a การเพิ่มจำนวนเซลล์

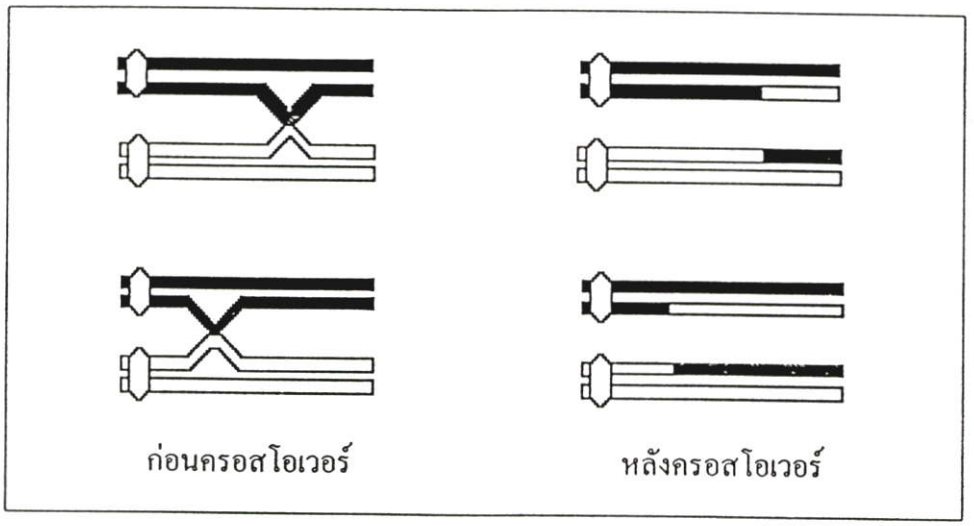


3.2-b การแบ่งตัวของเซลล์สืบพันธ์

รูปที่ 3.2 แสดงการแบ่งตัวของเซลล์ทางพันธุศาสตร์

วิธีการทางพันธุศาสตร์ในระหว่างที่เกิดการแบ่งตัวของไมโอสิสนั้น โครโมโซมจะมีโอกาสแลกเปลี่ยนส่วนบางส่วนซึ่งกันและกัน อันเป็นสาเหตุที่ทำให้เกิดปรากฏการณ์ที่เรียกว่าครอสโอเวอร์ (Crossover) ของลักษณะต่างๆขึ้น ซึ่งการครอสโอเวอร์นั้นเกิดขึ้นขณะที่มีการจำลองแบบเพิ่มขึ้น และเกิดขั้นระหว่างโครโมโซมพ่อกับโครโมโซมแม่ ไม่ใช่เกิดขึ้นระหว่างโครโมโซมพ่อกับแบบจำลอง ซึ่งการครอสโอเวอร์จะทำให้เกิดการแลกเปลี่ยนลักษณะของยีนส์ต่างๆ ของคู่

โคมโซมพ่อกับแม่ โดยเนื่องจากยนส์แต่ละยีนส์เรียงตัวกันบนเส้นโครโมโซมนั้นไม่ได้อยู่กันอย่างหนาแน่น แต่มีระยะห่างกันอย่างไม่สม่ำเสมอ เพราะคุณสมบัติของยีนส์นั้นเป็นโมเลกุลของสารโปรตีนประกอบตัวกันทางเคมี ช่องว่างระหว่างยีนส์นี่เองจะเป็นตำแหน่งที่แตกต่งออกมาได้เวลาครอสโอเวอร์ และแลกเปลี่ยนยีนส์ของโครโมโซมโดยส่วนที่อยธ่หลังรอยแตก โดยจะถูกย้ายไปอยู่อีกโครโมโซมหนึ่งทั้งหมด นอกจากนี้ยังจะสามารถแตกอีกกี่แห่งก็ได้ ซึ่งผลนั้นขึ้นอยู่กับความสามารถที่จะเชื่อมกันมากน้อยเพียงไรของช่องว่างระหว่างยีนส์ดังรูปที่ 3.3

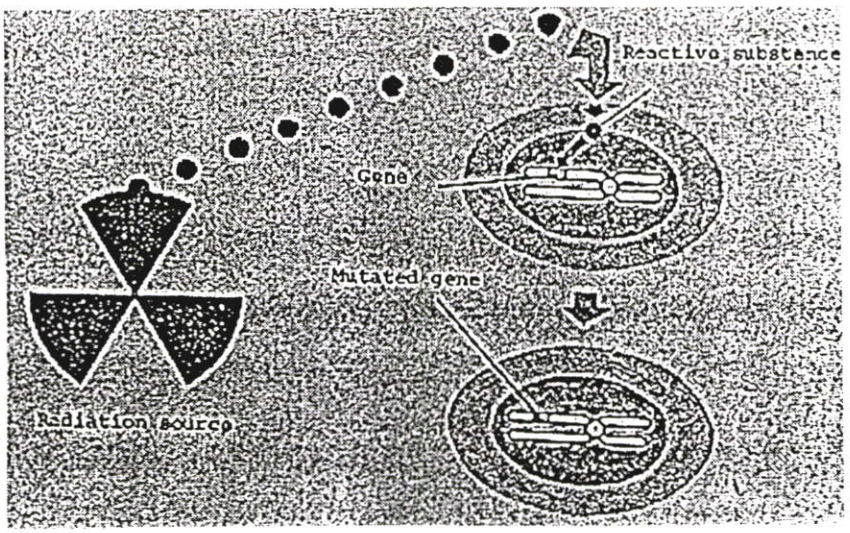


รูปที่ 3.3 แสดงการครอสโอเวอร์ทางพันธุศาสตร์

ประโยชน์ที่เกิดจากการครอสโอเวอร์ที่เราจะเห็น ได้คือทำให้มีโอกาสที่จะได้ลักษณะต่างๆกันมาอยู่รวมกันได้หลายแบบมากขึ้น ทำให้สิ่งมีชีวิตรุ่นลูกที่เกิดขึ้นมามีความหลากหลายมากขึ้น และอาจทำให้มีโอกาสเกิดสิ่งมีชีวิตที่มีลักษณะต่างๆ ที่ดีพอเหมาะรวมอยู่ด้วยกันได้อย่างพอดี เหมาะสมกับสิ่งแวดล้อม ถ้าการเกิดเซลล์ใหม่โดยถ่ายทอดโครโมโซมไม่มีการครอสโอเวอร์แล้ว โครโมโซมใดเคยมียีนส์ลักษณะใดก็จะมีลักษณะนั้นอยู่เรื่อยๆ รุ่นลูกก็จะมีลักษณะเช่นเดียวกัน โอกาสทีสิ่งมีชีวิตนั้นจะเจริญหรือปรับตัวให้ดีขึ้นย่อมมีได้ยากกว่าการเปลี่ยนลักษณะยีนส์ใหม่หลายๆแบบมากขึ้น

ลักษณะต่างๆ ของสิ่งมีชีวิตจะสามารถอยู่รอดได้โดยการคัดเลือกทางธรรมชาติคือ คัดเลือกโครโมโซมที่มีลักษณะที่ทำให้สิ่งมีชีวิตแข็งแรงเพียงพอ หรือเหมาะสมต่อสภาพแวดล้อม ซึ่งจะสามารถอยู่รอดและถ่ายทอดไปยังลูกหลาน ดังนั้นการคัดเลือกของธรรมชาติเพื่อถ่ายทอดลักษณะทางพันธุกรรม เป็นเพียงส่วนประกอบของการเปลี่ยนแปลงของสิ่งมีชีวิตเท่านั้น มิวเตชัน (Mutation) หรือการผ่าเหล่า คือการเปลี่ยนแปลงลักษณะของยีนส์ไปจากเดิมที่ควรเป็นไปตามการถ่ายทอด ซึ่งเป็นต้นเหตุของการเกิดลักษณะที่แปลกๆออกไปมากมายสำหรับสิ่งมีชีวิตหนึ่งๆ ซึ่งเท่ากับเป็นการให้โอกาสแก่ธรรมชาติในการที่จะเลือกลักษณะแปลกๆมากขึ้น เนื่องจากขบวนการวิวัฒนาการโดยธรรมชาติเองนั้นช้ามาก เพราะกว่าที่ธรรมชาติจะปรับสภาพแวดล้อมให้สิ่งมีชีวิต

ค่อยๆปรับตัวเองให้เหมาะสมนั้นมีโอกาสน้อยมาก การผ่าเหล่านั้นทุกลักษณะในแต่ละยีนส์ย่อมมีโอกาสที่จะเกิดความเปลี่ยนแปลงไปจากเดิมได้พอๆกัน และถ้าเหมาะสมกับสภาพแวดล้อมขณะนั้นก็จะคงอยู่ต่อไป แต่ถ้าการเปลี่ยนแปลงเป็นลักษณะใดเกิดผิดจังหวะคือไม่เหมาะสมกับสภาพขณะนั้นๆ ก็จะไม่ถูกคัดเลือกและหายไป ซึ่งข้อสรุปนี้ได้จากการทดลองโดยการนำเอายีนส์ของแบคทีเรียมาผสมกัน และจัดสภาพแวดล้อมที่ไม่อำนวนต่อการผสมกันแล้ว ยีนส์จะปรับตัวเองเพื่อเร่งขบวนการผสมพันธุ์จนได้ผลดีในรุ่นหลัง ตัวอย่างของการผ่าเหล่าในอดีตคือการกำเนิดของปลาทองนั้น มีต้นกำเนิดมาจากการกลายพันธุ์หรือผ่าเหล่าของปลาชิว และในปัจจุบันลักษณะใหม่ที่เกิดจากการผ่าเหล่าก็ยังคงมีให้เห็นอยู่ เช่น ความสามารถของเชื้อโครแบคทีเรียในการต้านทานต่อยาฆ่าเชื้อ หรือเซลล์ผิดปกติอันเกิดจากกัมมันตภาพรังสีต่างๆ ซึ่งมีผลต่อสารภายในเซลล์ทำให้ลักษณะของยีนส์ในเซลล์เปลี่ยนไป ซึ่งแสดงดังรูปที่ 3.4 เป็นต้น



รูปที่ 3.4 แสดงการมิวเตชันทางพันธุศาสตร์

3.2 จีเนติกอัลกอริทึมเบื้องต้น

ปี ค.ศ. 1975 John Hollan เริ่มสนใจศึกษาในทฤษฎีวิวัฒนาการทางธรรมชาติ(Natural Evolution) ในการกำเนิดประชากร (Population) สิ่งมีชีวิตในรุ่นต่อๆไปโดยกระบวนการธรรมชาติทางชีววิทยาประกอบด้วยการคัดเลือกทางธรรมชาติ(Natural Selection) คือสิ่งมีชีวิตใดแข็งแรงกว่าย่อมมีโอกาสอยู่รอดได้มากกว่านั้นหมายถึงการมีโครโมโซมซึ่งประกอบด้วยยีนส์ต่างๆที่มีลักษณะที่ดีนั้นจะมีโอกาสอยู่รอดได้มากกว่า โครโมโซมที่สามารถอยู่รอดได้ก็จะถูกถ่ายทอดยีนส์ที่มีลักษณะที่ดีเหล่านั้นไปยังลูกหลายได้มากกว่าเช่นกัน และกระบวนการทางพันธุศาสตร์(Genetic Operation) คือการกำเนิดโครโมโซมใหม่โดยการผสมพันธุ์เพื่อถ่ายทอดยีนส์จากครอสโอเวอร์หรือกลายพันธุ์จากมิวเตชัน

จากความเชื่อในวิวัฒนาการทางธรรมชาติ ที่แสดงถึงคุณลักษณะที่เป็นอยู่ของสิ่งมีชีวิตโดยการถ่ายทอดลักษณะต่างๆบนโครโมโซมนั้นมีคุณสมบัติทั่วไปที่ยอมรับกันคือ

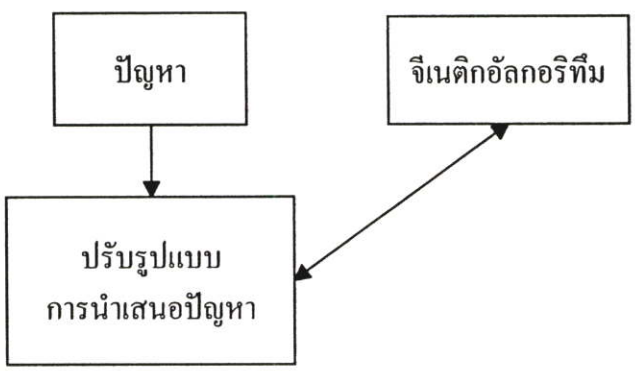
(1) วิวัฒนาการเป็นผลที่เกิดขึ้นเนื่องจากความเปลี่ยนแปลงบนโครโมโซม ที่เป็นอยู่ซึ่งแสดงลักษณะของสิ่งมีชีวิตนั้นๆ

(2) ธรรมชาติทางการคัดเลือกมีความสัมพันธกับโครโมโซมที่แสดงถึงประสิทธิภาพของโครงสร้างที่ดี ที่จะคัดเลือกเพื่อถ่ายทอดส่วนของโครงสร้างที่ดี

(3) การถ่ายทอดในขณะที่เกิดวิวัฒนาการนั้นโครโมโซมพ่อ-แม่มีการแลกเปลี่ยนส่วนโครงสร้างกันเพื่อสร้างโครโมโซมลูก และเหตุผลที่ทำให้เกิดโครโมโซมลูกที่แตกต่างออกไปคือขบวนการผ่าเหล่า

(4) วิวัฒนาการทางธรรมชาติมิได้เป็นสิ่งที่เกิดจากความจดจำ แต่เป็นกระบวนการที่เกิดจากโครงสร้างต่างๆในโครโมโซมที่เหมาะสมกับสภาพแวดล้อมที่เกิดขึ้นในขณะนั้น

Holland คิดว่าแนวความคิดจากคุณสมบัติเหล่านี้ น่าจะนำมาปรับใช้กับคอมพิวเตอร์ให้ช่วยแก้ปัญหาที่ยุ่งยากกต่างๆในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุด เขาจึงได้ทำการวิจัยโดยจำลองแบบเพื่อทดลองกับปัญหาแบบต่างๆโดยมีจุดมุ่งหมายที่จะศึกษาระบบปรับปรุงการประมวลผลเอง(self adaptive process) และเพื่อสร้างโปรแกรมระบบผู้เชี่ยวชาญ(artificial system software) เพื่อแก้ปัญหา โดยอาศัยแนวความคิดของระบบทางธรรมชาติ และค้นพบวิธีการใหม่ซึ่งเรียกว่าจีเนติกอัลกอริทึม (Genetic Algorithms :GA)



รูปที่ 3.5 แสดงหลักการเบื้องต้นของ GA

รูปที่3.5 แสดงหลักการเบื้องต้นในการใช้ GA แก้ปัญหา โดยจะต้องมีการปรับปรุงรูปแบบปัญหาในการนำเสนอ GA ในลักษณะที่เหมาะสม เพราะ GA เป็นวิธีการค้นหาคำตอบโดยอาศัยวิธีการเลียนแบบการคัดเลือกทางธรรมชาติ และธรรมชาติทางพันธุกรรมโดยการรวมกันหรือสลับเปลี่ยนตัวแปรต่างๆ อันเป็นองค์ประกอบโครงสร้างของปัญหาที่ให้คำตอบที่ต้องการซึ่งอาศัยหลักการสุ่ม เพื่อปรับปรุงความสามารถในการค้นหาคำตอบที่ดีขึ้น การค้นหาคำตอบจากรุ่นหนึ่งไปรุ่น

ถัดไปตามวิวัฒนาการทางธรรมชาตินั้น คำตอบในรุ่นใหม่เกิดขึ้นจากการสร้างความสัมพันธ์ของโครงสร้างต่างๆที่ประกอบด้วยค่าตัวแปรที่เหมาะสมดีในรุ่นก่อน ดังนั้นจึงทกให้ได้คำตอบที่ดีขึ้น จะเห็นได้ว่าวิธีการพื้นฐานของ GA เป็นแบบการสุ่ม แต่มีหลักการและประสิทธิภาพจากการคาดเดาคำตอบใหม่จากสถิติคำตอบเดิมที่ดี ซึ่งแตกต่างจากวิธีการทั่วไปคือ

(1) GA ค้นหาคำตอบภายใต้โครงสร้างของปัญหาอันเกิดจากการกำหนดรหัส (coding) รูปแบบโครงสร้างจากกลุ่มตัวแปรต่างๆของปัญหานั้น ไม่ใช่ค้นหาคำตอบจากค่าของกลุ่มตัวแปรนั้น

(2) GA ค้นหาคำตอบโดยพิจารณาจากประชากรคำตอบ หรือกลุ่มคำตอบ ไม่ใช่พิจารณาจากคำตอบใดคำตอบหนึ่ง

(3) GA ค้นหาคำตอบจากผลลัพธ์ของกลุ่มค่าตัวแปรที่เป็นฟังก์ชันเป้าหมายของปัญหา

(4) GA ค้นหาคำตอบโดยอาศัยการถ่วงน้ำหนักความเหมาะสมของแต่ละคำตอบจากกลุ่มคำตอบนั้นๆ

3.2.1 ฟังก์ชันเป้าหมายกับฟังก์ชันความเหมาะสม

การหาคำตอบที่ดีที่สุดของปัญหาของ GA มีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา วิธีการของ GA จะไม่พิจารณาจากขั้นตอนของการแก้ปัญหาแต่จะพิจารณาโดยตัดสินว่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือเป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่ จากฟังก์ชันเป้าหมาย (Object Function : f) เนื่องจากแต่ละปัญหาจะสามารถกำหนดฟังก์ชันเป้าหมายซึ่งเป็นฟังก์ชันที่แสดงความสัมพนธ์ของแต่ละตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆ ของปัญหานั้นๆ ที่ระบุคำตอบใดคำตอบหนึ่งที่สามารถเป็นไปได้ ณ.ค่าพารามิเตอร์ เงื่อนไข หรือข้อกำหนดชุดดังกล่าว สำหรับฟังก์ชันความเหมาะสม (Fitness Function: F) เป็นฟังก์ชันที่กำหนดค่าความเหมาะสม(fitness) ของแต่ละโครโมโซมเปรียบเสมือนค่าความสามารถในการอยู่รอดของแต่ละโครโมโซมและเป็นฟังก์ชันที่กำหนดโอกาส หรือสัดส่วนที่แต่ละโครโมโซมเหมาะสมจะถูกคัดเลือกมากน้อยเพียงใด นั่นคือฟังก์ชันความเหมาะสมจะเป็นฟังก์ชันที่แสดงถึงค่าคำตอบที่เกิดขึ้นจากชุดตัวแปรของปัญหาของโครโมโซมนั้นดีเพียงใด โดยทั่วไปแล้วเรามักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสมหรืออาจจะใช้ฟังก์ชันเป้าหมายที่ถูกปรับให้เหมาะสมกับการนำเสนอ GA เป็นฟังก์ชันความเหมาะสมก็ได้

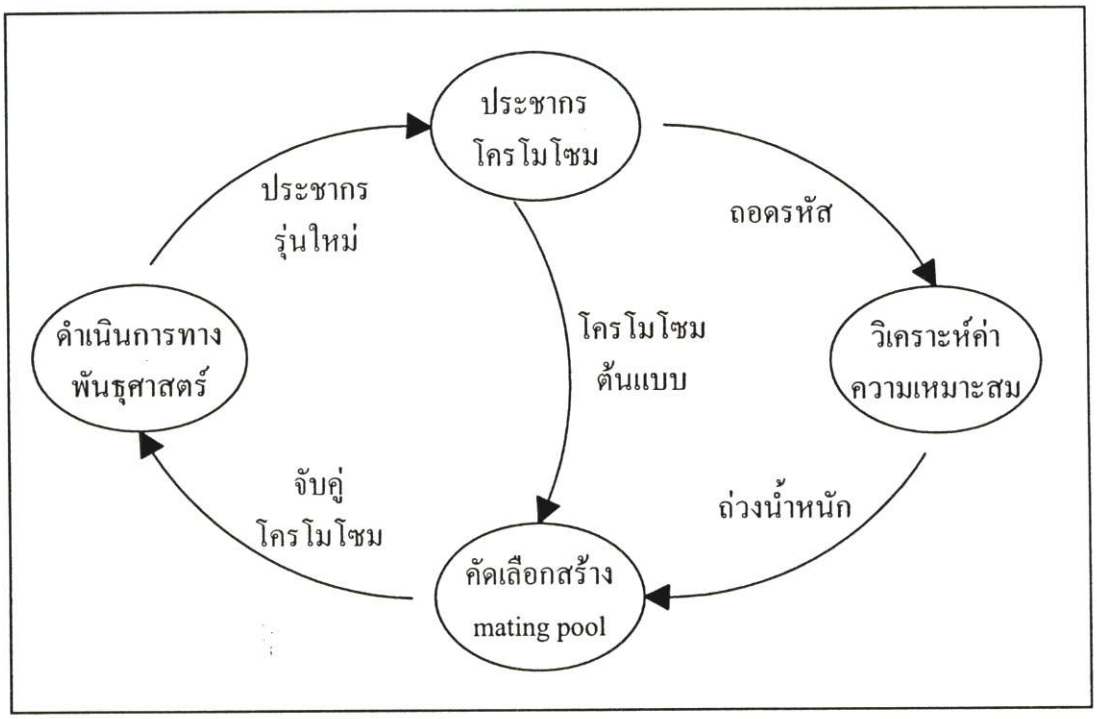
3.2.2 รูปแบบโครโมโซม

เราทราบกันแล้วว่าวิวัฒนาการทางธรรมชาติชีววิทยานั้นเป็นความเปลี่ยนแปลงต่างๆของสิ่งมีชีวิตเกิดขึ้นในโครโมโซม ดังนั้นจุดเริ่มต้นของการจำลองแบบทางธรรมชาติของ GA เพื่อใช้แก้ปัญหาจึงเริ่มจากการมองปัญหาเทียบเท่ากับโครโมโซมชนิดหนึ่ง ประกอบด้วยยีนส์ลักษณะต่างๆซึ่งหมายถึงลำดับข้อมูลต่างๆที่จะแปลความหมายแล้วให้ค่าคำตอบของปัญหาค่าหนึ่ง การมอง

ภาพยีนส์ของ GA ให้ถือเสมือนยีนส์ทางพันธ์กรรมที่แสดงความหมายหรือเป็นตัวแทนคำตอบในคำตอบหนึ่งหรือลักษณะใดลักษณะหนึ่งทางพันธุ์ ในทางพันธุศาสตร์นั้นยีนส์เป็นตัวแสดงลักษณะที่อยู่รอดในสภาพแวดล้อมขณะนั้น สำหรับ GA นั้นยีนส์เป็นตัวแสดงค่าคำตอบของปัญหาที่แปรผันไปตามการประยุกต์ใช้งาน ซึ่งโดยทั่วไปยีนส์หมายถึงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆที่เป็นองค์ประกอบของปัญหา ดังนั้นการกำหนดรูปแบบโครโมโซมของแต่ละปัญหาโดยการแปลงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆให้อยู่ในรูปลำดับของยีนส์บนโครโมโซมหรือเรียกว่าสตริง(string) อันประกอบด้วยบิท(bit) หรือเรียกว่าอักขระ(character) ซึ่งลักษณะต่างๆที่เป็นไปได้ของแต่ละยีนส์คือค่าของบิท(bit Value) หรือค่าตัวแปรพารามิเตอร์ต่างๆที่เป็นไปได้ และรูปแบบของบิทที่จัดเรียงบนโครโมโซมคือ ยีโนไทป์(genotype) ที่จะแสดงถึงค่าของตัวแปรพารามิเตอร์ต่างๆที่เป็นไปได้ชุดหนึ่งหรือฟีโนไทป์(phenotype) นั่นเอง การกำหนดรูปแบบโครโมโซมของปัญหาให้เป็นตามแบบธรรมชาติ โดยกำหนดรหัสในรูปแบบตัวเลขหรือตัวอักษรในช่วงที่จำกัดตามค่าตัวแปรหรือพารามิเตอร์ และประกอบรวมกันในจำนวนยีนส์หรือความยาวของโครโมโซมที่คงที่ เช่น หากต้องการหาค่าสูงสุดของฟังก์ชันทางคณิตศาสตร์ $y = x^2$ ที่ x เป็นจำนวนเต็มอยู่ในช่วง [0,31] แล้ววิธีการของ GA ในการแก้ปัญหาโดยกำหนดรูปแบบโครโมโซมจากการกำหนดรหัสตัวแปร x เป็นตัวเลขไบนารี 0 หรือ 1 จำนวน 5 ตำแหน่ง ซึ่ง x จะมีค่าตั้งแต่ 00000 ถึง 11111 เป็นค่า 0 ถึง 31 ตามต้องการ เป็นต้น

3.2.3 วัฏจักรจีเนติกอัลกอริทึม

เมื่อกำหนดรูปแบบโครโมโซมและฟังก์ชันความเหมาะสมของปัญหาแล้ว GA จะสามารถประมวลผลหาคำตอบของปัญหาได้ โดยสร้างวิวัฒนาการกลุ่มคำตอบในรุ่นต่อไปตามวัฏจักรการทำงานของ GA (Genetic Algorithm Cycle) ดังรูปที่ 3.6 ซึ่งมี 4 ขั้นตอน คือ



รูปที่ 3.6 แสดงวัฏจักรจีเนติกอัลกอริทึม

(1) สร้างประชากรโครโมโซมรุ่นเก่าตามรูปแบบโครโมโซมที่กำหนดไว้ โดยประชากรต้นกำเนิด(Initial Population) เกิดจากการสร้างชุดโครโมโซมต้นกำเนิด จากการสุ่มสร้างค่าแต่ละบิทของแต่ละโครโมโซม

(2) วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซม โดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่างๆ ของแต่ละบิทในโครโมโซมและคำนวณค่าความเหมาะสมจากฟังก์ชันความเหมาะสมที่กำหนดไว้

(3) สร้าง mating pool คือชุดโครโมโซมต้นแบบหรือชุดโครโมโซมพ่อ-แม่ ที่สามารถอยู่รอดเป็นต้นแบบ ซึ่งอาศัยการจำลองการคัดเลือกทางธรรมชาติ โดยพิจารณาถ่วงน้ำหนักจากค่าความเหมาะสมของแต่ละโครโมโซม หากโครโมโซมมีค่าความเหมาะสมมากก็จะมีโอกาสถูกคัดเลือกเป็นต้นแบบมาก

(4) ดำเนินการทางพันธุศาสตร์โดยสุ่มจับคู่โครโมโซมต้นแบบใน mating pool เพื่อสร้างประชากรโครโมโซมรุ่นใหม่ ซึ่งตัวดำเนินการทางพันธุศาสตร์ประกอบด้วย ครอสโอเวอร์ โดยการแลกเปลี่ยนค่าบิทบางส่วนของโครโมโซมซึ่งกันและกัน หรือมิวเตชัน โดยสุ่มเปลี่ยนค่าบิทบางบิทของแต่ละโครโมโซมเป็นต้น

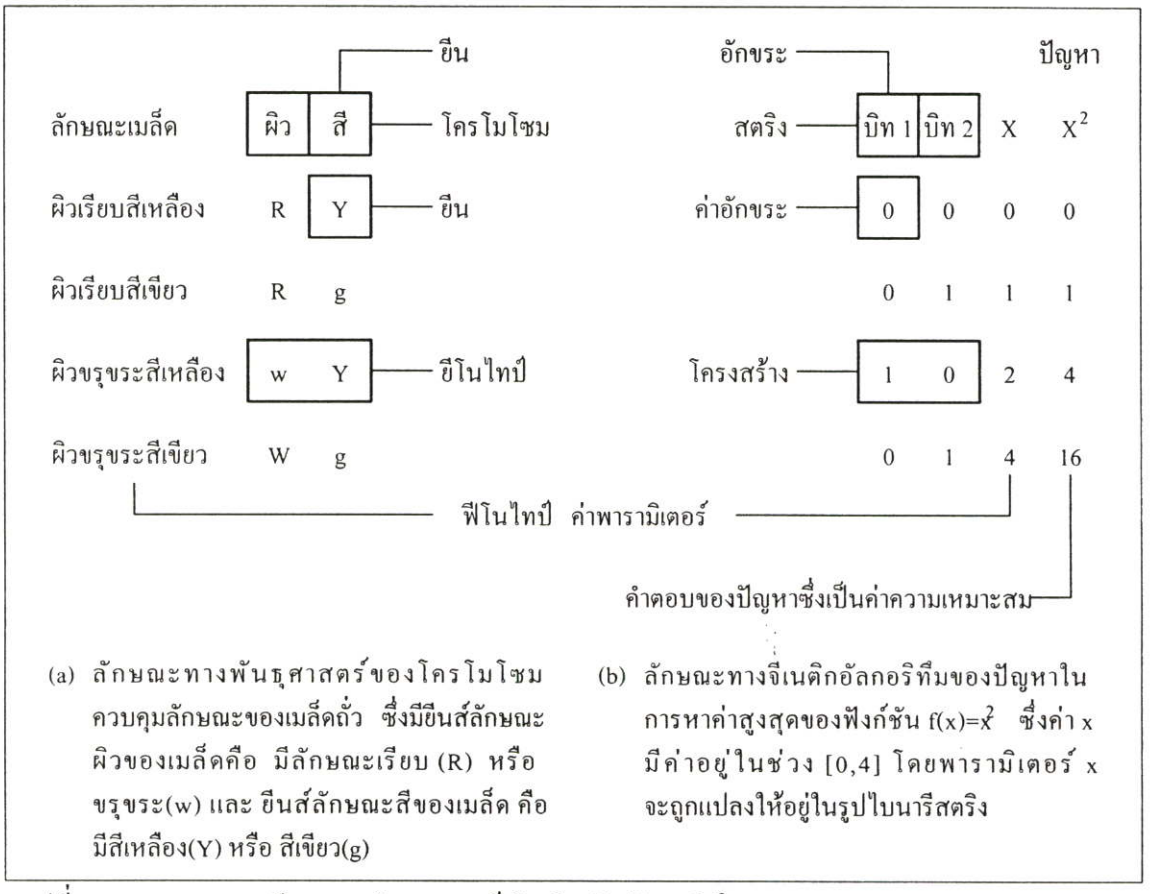
การค้นหาคำตอบของ GA จะประมาณผลซ้ำตามวัฏจักร GA จนกว่าจะได้รับคำตอบที่พอใจตามกฎเกณฑ์ที่ตั้งไว้ หรือในระยะเวลาตามจำนวนรุ่นที่ดำเนินการที่ต้องการ ซึ่งแสดงอัลกอริทึมการทำงานของ GA ดังนี้

อัลกอริทึม GA

```
t := 0;
// สร้างประชากรโครโมโซมต้นกำเนิดโดยการสุ่ม
Initpopulation P(t);
// วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซมประชากรต้นกำเนิด
Evaluate P(t);
// ตรวจสอบเงื่อนไขความพอใน (เช่น เวลา, ค่าความเหมาะสม, เป็นต้น)
while not terminate
begin
        t := t+1;
        // คัดเลือกโครโมโซมต้นแบบจากประชากรรุ่นก่อน
        P'(t) := Selectparents P(t-1);
        // แลกเปลี่ยนส่วนยีนส์ภายในโครโมโซมต้นแบบ
        Recombine P'(t);
        // มิวเตชันโครโมโซมต้นแบบ
        Mutate P'(t);
        // วิเคราะห์ค่าความเหมาะสมของประชากรรุ่นใหม่
        Evaluate P'(t);
        // ประชากรรุ่นใหม่กลายเป็นประชากรรุ่นเก่าต่อไป
        P(t) := P'(t);
end;
END.
```

3.2.4 พันธุศาสตร์ทางชีววิทยา กับ จีเนติกอัลกอริทึม

เพื่อเปรียบเทียบลักษณะโครงสร้างทางพันธุศาสตร์ กับ จีเนติกอัลกอริทึม แล้วเรากล่าวโดยสรุปได้คือ ในทางพันธุศาสตร์ แต่ละโครโมโซมประกอบด้วยหน่วยเก็บลักษณะหรือยีนส์ ซึ่งเก็บค่าแสดงลักษณะ หรือ แอลลี และแต่ละแบบของชุดยีนส์เรียกว่ายีโนไทป์ ซึ่งแสดงลักษณะภายนอกที่ปรากฏเรียกว่า ฟีโนไทป์ ดังรูปที่ 3.7-a



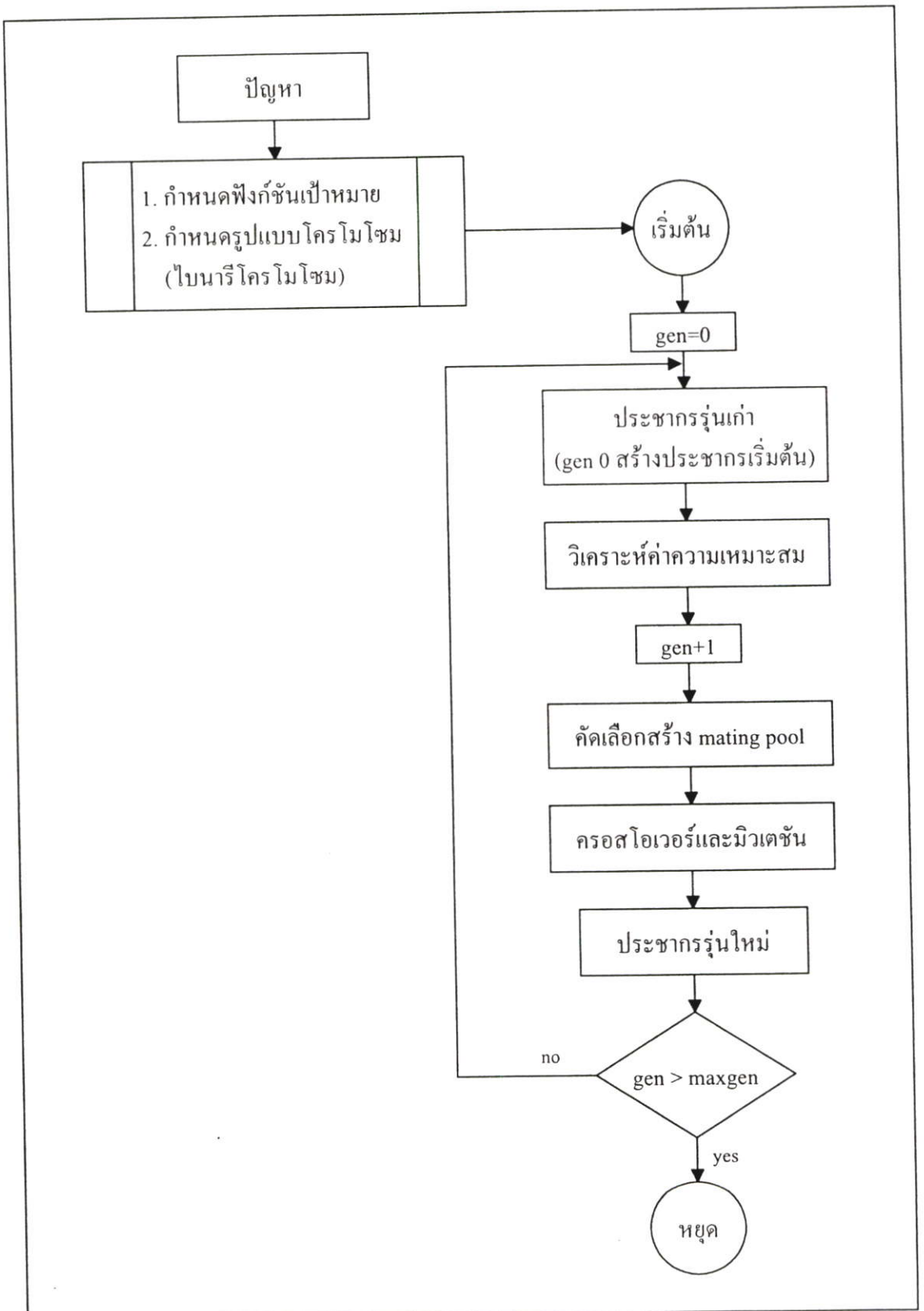
รูปที่ 3.7 แสดงรายละเอียดทางพันธุศาสตร์ กับ จีเนติกอัลกอริทึม

สำหรับในทางจีเนติกอัลกอริทึม ตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของสตริง ซึ่งมักเรียกว่าโครโมโซม ประกอบด้วยอักขระหรือบิท แต่ละตำแหน่งของโครโมโซมจะเก็บค่าอักขระหรือค่าของบิท ที่แสดงถึงโครงสร้างของแต่ละโครโมโซมที่มีค่าตัวแปรหรือพารามิเตอร์ของปัญหาแตกต่างกัน และเป็นตัวกำหนดค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของแต่ละปัญหา ดังรูปที่ 3.7-b ซึ่งสรุปความหมายเปรียบเทียบคำศัพท์ ที่ใช้ทางพันธุศาสตร์กับทางจีเนติกอัลกอริทึม ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงคำศัพท์ทางพันธุศาสตร์กับทางจีเนติกอัลกอริทึม

Natural Genetic	Genetic Algorithm
chromosome	string
gene	character, bit
allele	character value, bit value
locus	string position
genotype	structure
phenotype	a decode structure

3.3 จีเนติกอัลกอริทึมแบบง่าย



รูปที่ 3.8 แสดงไดอะแกรมการทำงานของจีเนติกอัลกอริทึมแบบง่าย

GA ในยุคเริ่มแรกของ Holland นั้นคือ จีเนติกอัลกอริทึมแบบง่าย (Simple Genetic Algorithm : SGA) ซึ่งมีขั้นตอนพื้นฐานที่มีกระบวนการไม่มากนักและง่ายในการศึกษาความเข้าใน แต่ละขั้นตอบการทำงานของ GA เพื่อแก้ปัญหาในการหาคำตอบ แสดงดังไดอะแกรมในรูปที่ 3.8 แบ่งออกเป็น 2 ส่วนคือ ขั้นเตรียมการและขั้นตอนการทำงาน

สำหรับในส่วนของขั้นเตรียมการนั้นเป็นส่วนของการปรับรูปแบบของปัญหาให้เหมาะสมสำหรับการนำเสนอ GA เพื่อใช้ในการแก้ปัญหานั้นๆ ประกอบด้วย

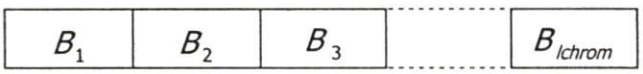
1. กำหนดฟังก์ชันความเหมาะสม เพื่อความสะดวกและง่ายต่อความเข้าใจในขั้นตอนการทำงานต่างๆ จะกำหนดตัวอย่างปัญหาสำหรับอธิบายรายละเอียดการหาคำตอบของ SGA คือ ปัญหาการหาค่าสูงสุดของฟังก์ชัน $y = x^2$ ที่ x มีค่าระหว่างจำนวนเต็ม I [0,31] ดังนั้น

ตัวอย่าง : ฟังก์ชันเป้าหมาย คือ $f = x^2$

และกำหนดให้ฟังก์ชันความเหมาะสมคือ $F = x^2$

ซึ่งคำตอบที่ดีที่สุด คือ ค่า x ที่มีค่าความเหมาะสมสูงสุด ($MAX(F)$)

2. กำหนดรูปแบบโครโมโซม รูปแบบโครโมโซมของ SGA นั้นเป็นแบบไบนารี โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของไบนารีโครโมโซม คือประกอบด้วยบิทที่มีค่าเป็น 0 หรือ 1 ซึ่งเป็นค่าในเลขฐานสองและมีความยาว (Chromosome Length : $lchrom$) ตามแต่จะกำหนด ซึ่งแสดงด้วยสัญลักษณ์ได้ดังนี้



ซึ่ง $B_i \in I$ [0,1]

ตัวอย่าง : วิธีการเข้ารหัสแบบไบนารีโดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี บิท 5 บิท($lchrom$ =5) ดังนั้นโครโมโซมของปัญหาจะมีค่าอยู่ในช่วง 00000 ถึง 11111 ซึ่งเมื่อถอดรหัสแล้วจะทำให้ x มีค่าอยู่ในช่วง 0 ถึง 31 ตามที่ต้องการ

ในส่วนของรายระเอียดขั้นตอนการทำงานของ SGA จะเป็นขั้นตอนพื้นฐานเบื้องต้นแบบง่ายประกอบด้วย

1. ประชากรรุ่นเก่า (Old Population) เป็นชุดโครโมโซมที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) ในวิวัฒนาการรุ่น (generation : gen) ต่อไป โดยประชากรเริ่มต้นที่ gen=0 จะถูกสร้างขึ้นโดยการสุ่มตามจำนวนโครโมโซมในแต่ละรุ่น (Population Size: popsize) ที่กำหนด

ตัวอย่าง:

ลำดับ	โครโมโซม
1	0 1 1 1 0
2	1 1 0 0 1
3	0 1 0 0 0
4	1 0 0 1 1

ชุดโครโมโซมในรุ่นเริ่มต้นนี้เป็นชุดโครโมโซมที่กำหนดให้ในแต่ละรุ่นประกอบด้วย 4 โครโมโซม ซึ่งแต่ละโครโมโซมเกิดจากการสุ่มค่าไบนารี 0 หรือ 1 จำนวน 5 ครั้ง

2. วิเคราะห์ค่าความเหมาะสม เป็นขั้นตอนของการถอดรหัสจากรูปแบบโครโมโซมที่กำหนดไว้ เพื่อคำนวณค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของปัญหาของแต่ละโครโมโซม ในที่นี้ฟังก์ชันเป้าหมายหรือฟังก์ชันความเหมาะสม คือ $F = x^2$ ดังนั้นการวิเคราะห์ค่าความเหมาะสมของ SGA โดยถอดรหัสเลขฐาน 2 ของแต่ละโครโมโซมเป็นค่าของตัวแปร x และคำนวณค่าความเหมาะสมคือค่า x^2

ตัวอย่าง:

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม
1	0 1 1 1 0	14	196
2	1 1 0 0 1	25	625
3	0 1 0 0 0	8	64
4	1 0 0 1 1	19	361

ชุดโครโมโซมในรุ่นเริ่มต้นมีค่าความเหมาะสมเป็น 196 , 625 , 64 และ 361 ตามลำดับ

3. การคัดเลือก เป็นขั้นตอนที่จำลองแบบการคัดเลือกทางธรรมชาติเพื่อสร้าง mating pool โดยคัดเลือกชุดโครโมโซมรุ่นเก่าให้เป็นโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ เพื่อใช้สร้างโครโมโซมลูกรุ่นต่อไป สำหรับการคัดเลือกของ SGA เป็นแบบอ้างอิงค่าความเหมาะสม(Fitness-base Selection) โดยพิจารณาค่าความเหมาะสมเป็นตัวตัดสินว่าโครโมโซมใดในรุ่นเก่ามีโอกาสถูกเลือกเป็นโครโมโซมพ่อ-แม่มากน้อยเพียงใด โครโมโซมที่มีค่าความเหมาะสมที่ดีจะถูกเลือกต่อการสุ่มเลือกแต่ละครั้งสูง การกำหนดค่าความน่าจะเป็นที่จะถูกเลือกต่อการสุ่มเลือกแต่ละครั้ง (Probability of Selected Value : $pselect$) ของแต่ละโครโมโซมโดยกำหนดตจากค่าความเหมาะสมเทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่3.1

$$pselect_i = \frac{F_i}{\sum F} \tag{3.1}$$

ซึ่งสามารถคำนวณค่าที่คาดหวังว่าจะสุ่มได้ (Expected Value : E) ของแต่ละโครโมโซมในแต่ละรุ่นดังสมการที่3.2

$$E_i = pselect_i * popsize = \frac{F_i}{\overline{F}} \quad (3.2)$$

สำหรับวิธีการสุ่มโครโมโซมต้นแบบของ SGA เป็นแบบจำลองการหมุนวงล้อถ่วงน้ำหนัก (Roullet Wheel : RW) ซึ่งกำหนดขนาดแต่ละช่องของวงล้อนั้นตามค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม ซึ่งมีวิธีการดังนี้

(1) หาค่าความเหมาะสมของแต่ละโครโมโซม

(2) หาค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม

(3) หาความถี่สะสม(q) ของค่าความน่าจะเป็นของแต่ละโครโมโซมดังสมการที่3.3

$$q_i = \sum_{j=1}^{i} pselect_j \quad (3.3)$$

(4) สร้างเลขสุ่มจำนวนจริง(r) มีค่าอยู่ในช่วง [0.0,1.0]

(5) เลือกโครโมโซมลำดับที่ r ซึ่ง r มีค่าอยู่ระหว่าง q_{i-1} และ q_i

ตัวอย่าง :

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม (F)	ค่าความน่าจะเป็น ($pselect_i$)	จำนวนที่คาดหวัง (E_i)	จำนวนที่สุ่มได้จาก RW
1	0 1 1 1 0	14	196	0.157	0.628	1
2	1 1 0 0 1	25	625	0.502	2.008	2
3	0 1 0 0 0	8	64	0.051	0.204	0
4	1 0 0 1 1	19	361	0.290	1.160	1
	รวม		1246	1.000	4.000	
	ค่าเฉลี่ย		312	0.250	1.000	
	ค่าสูงสุด		625	0.502	2.008	

ตัวอย่างการกำหนดค่าความน่าจะเป็น โดยกำหนดจากค่าความเหมาะสมเทียบกับผลรวมของค่าความเหมาะสมทั้งหมด จะเห็นได้ว่าในการคัดเลือกโครโมโซมต้นแบบจาก 4 โครโมโซมนี้ โอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 1 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.157 และโอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 2,3,4 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.502, 0.051

และ 0.290 ตามลำดับ และจำนวนโครโมโซมต้นแบบที่สุ่มได้โดยจำลองการหมุนวงล้อดังนี้

ลำดับโครโมโซม	1	2	3	4
ค่าความเหมาะสม(F)	196	625	64	361
ค่าความน่าจะเป็นที่สุ่มได้แต่ละครั้ง($pselect_i$)	0.157	0.502	0.051	0.290
ความถี่สะสมค่าความน่าจะเป็น(q_i)	0.157	0.659	0.710	1.000
สร้างเลขสุ่มในการหมุนวงล้อแต่ละครั้ง(r)	0.333	0.844	0.456	0.128
ลำดับโครโมโซมที่ถูกเลือก($q_{i-1} \le r \le q_i$)	2	4	2	1

ซึ่งจำนวนที่สุ่มได้เป็นโครโมโซมต้นแบบใน mating pool ของแต่ละโครโมโซมเป็น 1,2,0 และ 1 ตามลำดับ จะเห็นได้ว่าโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมสูงที่สุดจะมีโอกาสถูกเลือกในจำนวนที่มากที่สุด ส่วนโครโมโซมลำดับที่ 3 มีค่าความเหมาะสมต่ำมากจึงมีโอกาสที่จะไม่ถูกเลือกเลย

4. ดำเนินการทางพันธุศาสตร์ เป็นขั้นตอนที่จำลองแบบธรรมชาติทางพันธุกรรม ซึ่งตัวดำเนินการทางพันธุศาสตร์ของ SGA คือ ครอสโอเวอร์และมิวเตชัน ซึ่งมีรายละเอียดดังนี้

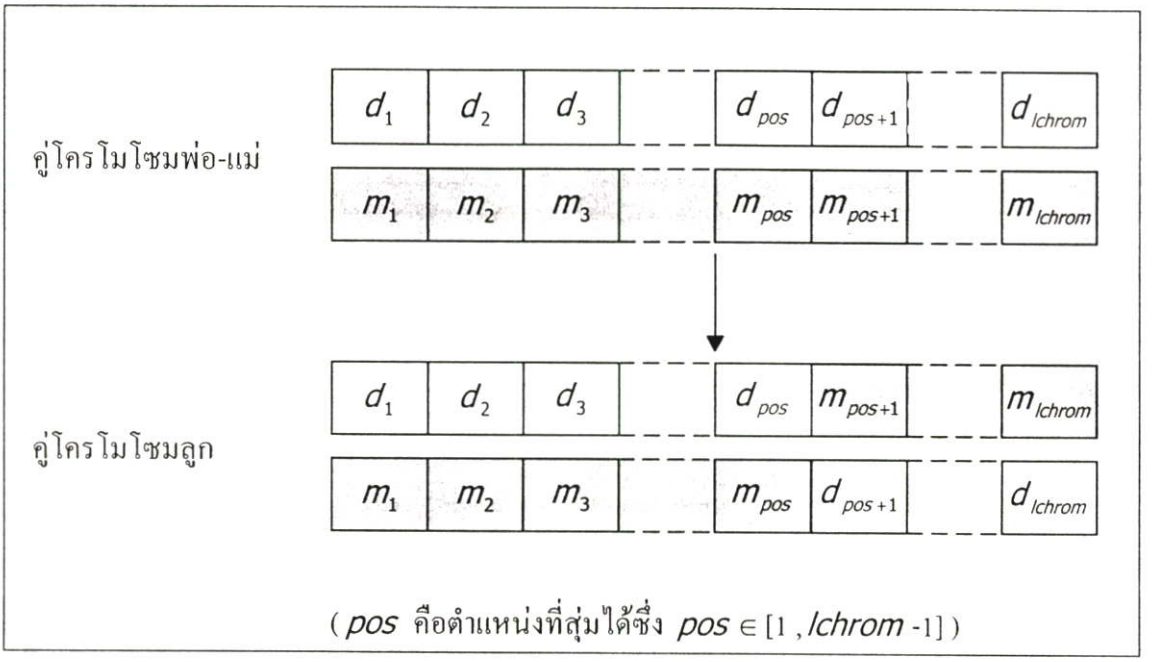
4.1 ครอสโอเวอร์ เป็นตัวดำเนินการในการแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่ตามการกำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossove : P_c) เพื่อสร้างชุดโครโมโซมรุ่นใหม่หรือโครโมโซมลูก มีขั้นตอนการทำงานคือ

ขั้นตอนแรก : สุ่มจับคู่โครโมโซมพ่อ-แม่ใน mating pool ที่สร้างไว้จากการเลือก

ขั้นตอนที่สอง : สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0,1.0] โดยถ้า $r \le P_c$ แล้วโครโมโซมพ่อ-แม่นั้นจึงมีการครอสโอเวอร์

ขั้นตอนที่สาม : ครอสโอเวอร์โดยแลกเปลี่ยนส่วนของคู่โครโมโซมพ่อ-แม่นั้นซึ่งการครอสโอเวอร์ของ SGA เป็นการครอสโอเวอร์แบบ 1 จุด (One-point Crossover) แสดงดังรูปที่ 3.9 ดังนี้

- สุ่มเลือกตำแหน่ง pos เป็นตำแหน่งที่จะครอสโอเวอร์ ซึ่ง pos มีค่าในช่วง [1, $lchrom$ -1]
- แลกเปลี่ยนค่าในแต่ละบิทของคู่โครโมโซมพ่อ-แม่ตั้งแต่ตำแหน่งที่ pos +1 ถึง lchrom ซึ่งจะทำให้เกิดโครโมโซมลูกใหม่ 2 โครโมโซม



รูปที่ 3.9 ครอสโอเวอร์แบบ 1 จุด

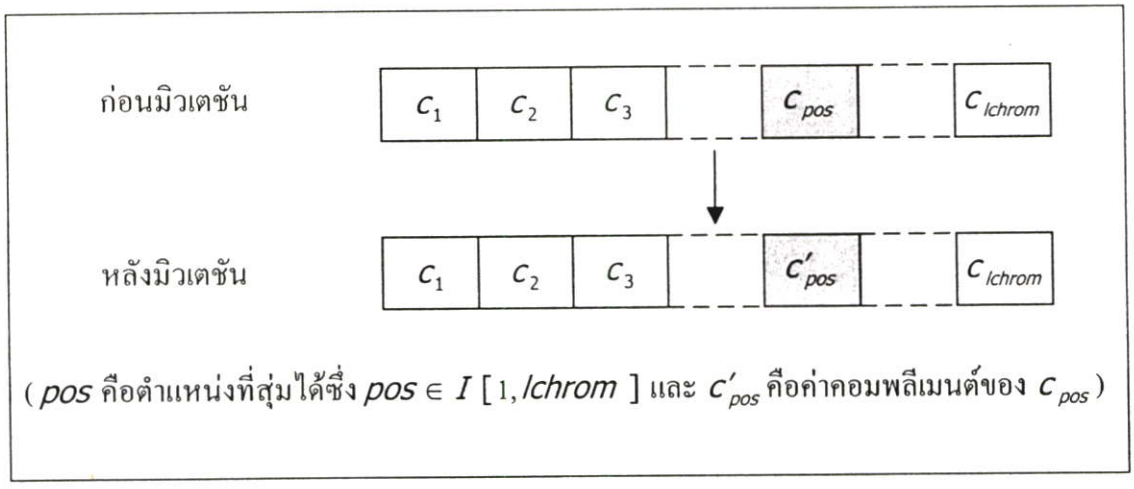
จำนวนการครอสโอเวอร์ในแต่ละรุ่นดำเนินการขึ้นอยู่กับการกำหนดค่า P_c ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม และกำหนดให้ P_c = 0.6 แล้วจำนวนการครอสโอเวอร์ในแต่ละรุ่นเท่ากับ P_c * (popsize/2)=0.6*(30/2)=9 ครั้ง (การครอสโอเวอร์ 1 ครั้งเกิดจากโครโมโซม 2 โครโมโซม)

ตัวอย่าง : กำหนด P_c =0.5 โครโมโซมพ่อ-แม่ใน mating pool จากการคัดเลือกครอสโอเวอร์ดังนี้

ลำดับที่คัดเลือก	mating pool	สุ่มจับคู่พ่อแม่	เลขสุ่ม (r)	ก่อนครอสโอเวอร์	สุ่มตำแหน่ง (pos)	หลังครอสโอเวอร์	x	ค่าความเหมาะสม (F)	ลำดับโครโมโซมลูก
2	1 1 0 0 1	1,2	0.321	0 1 1 1 1 0	2	0 1 0 0 1	9	81	1
4	1 0 0 1 1		≤0.5	1 1 0 0 1		1 1 1 1 0	30	**900**	2
2	1 1 0 0 1	2,4	0.654	ไม่ครอสโอเวอร์		1 1 0 0 1	25	625	3
1	0 1 1 1 0		> 0.5			1 0 0 1 1	19	361	4
	รวม							1967	
	ค่าเฉลี่ย							492	
	ค่าสูงสุด							900	

จากการสุ่มจับคู่โครโมโซมพ่อ-แม่ใน mating pool ได้โครโมโซมลำดับที่1 คู่ลำดับที่2 และลำดับที่2 คู่กับลำดับที่4 แต่เฉพาะโครโมโซมคู่แรกจะเกิดการครอสโอเวอร์ เนื่องจากเลขสุ่ม $r \leq 0.5$ ตามอัตราการครอสโอเวอร์ที่กำหนด โดยตำแหน่งในการครอสโอเวอร์ที่สุ่มได้คือ pos =2 จะเห็นได้ว่าโครโมโซมพ่อ-แม่ทั้งหมดในรุ่นก่อนเป็น 900 ซึ่งแสดงให้เห็นถึงการจำลองแบบกระบวนการครอสโอเวอร์ตามธรรมชาติทางพันธุศาสตร์ของ SGA ช่วยสร้างคำตอบที่ดีขึ้น

4.2 มิวเตชัน เป็นตัวดำเนินการผ่าเหล่าตัวหนึ่งที่อาจช่วยให้โครโมโซม มีค่าความเหมาะสมดีขึ้นหลังจากครอสโอเวอร์ โดยกลับค่าของบิทเป็นค่าใหม่ในตำแหน่งบิทที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิท (Probability of Mutation : P_m) ที่กำหนด สำหรับการมิวเตชันของ SGA นั้นเป็นแบบไบนารีมิวเตชัน (Binary Mutation) โดยกลับค่าบิทเป็นค่าคอมพลีเมนต์คือจาก 0 เป็น 1 หรือจาก 1 เป็น 0 ดังรูปที่3.10



รูปที่ 3.10 ไบนารีมิวเตชัน

จำนวนการมิวเตชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า P_m ซึ่งแตกต่างกันในแต่ละปัญหา เช่นถ้าจำนวนประชาการแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม ซึ่งแต่ละโครโมโซมประกอบด้วย 5 บิท และกำหนดให้ P_m = 0.02 แล้ว จำนวนการมิวเตชันในแต่ละรุ่นเท่ากับ P_m *popsize*lchrom = 0.02*30*5 = 3 บิท

ตัวอย่าง : กำหนด $P_m = 0.1$ ดำเนินการมิวเตชันโครโมโซมลูกที่ได้จากการครอสโอเวอร์ดังนี้

ลำดับ	ก่อนมิวเตชัน	เลขสุ่ม (r)	หลังมิวเตชัน	x	ค่าความเหมาะสม (F)
1	0 1 0 0 1	0.581 0.346 **0.062** 0.785 0.401	0 1 **1** 0 1	13	169
2	1 1 1 1 0	0.829 0.534 0.947 0.308 0.277	1 1 1 1 0	30	900
3	1 1 0 0 1	0.398 0.646 0.494 0.765 **0.029**	0 1 0 0 **0**	24	576
4	1 0 0 1 1	0.175 0.335 0.837 0.577 0.308	1 1 1 1 0	19	361
	รวม				2006
	ค่าเฉลี่ย				**502**
	ค่าสูงสุด				900

จากการสุ่มตำแหน่งที่จะมิวเตชันโดยสร้างเลขสุ่ม r ของแต่ละตำแหน่งบิทในแต่ละโครโมโซมแล้ว ตำแหน่งบิทที่ 3 ของโครโมโซมลำดับที่ 1 และตำแหน่งบิทที่ 4 ของโครโมโซมลำดับที่ 3 เป็นตำแหน่งที่ $r \leq 0.1$ ตามอัตราการมิวเตชันที่กำหนดจึงเกิดมิวเตชัน ทำให้โครโมโซมที่มีค่าความเหมาะสมจาก 81 และ 625 เป็น 169 และ 576 ตามลำดับ จะเห็นได้ว่ามิวเตชันเป็นตัวดำเนินการที่อาจทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือลดลงได้ แต่อย่างไรก็ตามค่าเฉลี่ยของค่าความเหมาะสมดีขึ้นจาก 492 เป็น 502 แสดงถึงการหาคำตอบของ SGA โดยส่วนมากดีขึ้น และความสำคัญของการหาคำตอบของ GA นั้นเป็นความต้องการได้คำตอบโดยพิจารณาจากคำตอบที่ดีขึ้นเกิดขึ้น ซึ่งจะมีโอกาสอยู่รอดเพื่อถ่ายทอดส่วนที่ดีในรุ่นต่อไป

5. ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากขั้นตอบของวิวัฒนาการต่างๆทั้งหมด ซึ่งประชากรรุ่นใหม่ทั้งหมดที่เกิดขึ้น จะถูกถ่ายทอดกลายเป็นประชากรรุ่นเก่าสำหรับวิวัฒนาการในรุ่นถัดไป ซึ่งเรียกว่าวิวัฒนาการแบบนี้ว่า การถ่ายทอดแบบทั่วไปหรือรีโพรดักชันแบบทั่วไป (General Reproduction) กระบวนการต่างๆจะถูกปฏิบัติซ้ำๆจนกระทั่งถึงรุ่นที่มากที่สุด(max generation : maxgen) ที่ต้องการ

การทดลองเพื่อทดสอบการทำงานของ SGA จะทดสอบโดยใช้ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ x มีค่าระหว่าง $[1,2^l\text{-}1]$ ซึ่ง l คือจำนวนบิทของโครโมโซม ด้วยวิธีการของ SGA กำหนดให้ฟังก์ชันเป้าหมาย คือ $f = x^n$ เพื่อความสะดวกในการวิเคราะห์การทำงานของ SGA จึงปรับให้ค่าความเหมาะสมอยู่ในช่วง [0,1] ซึ่งคำตอบที่ดีที่สุดจะมีค่าความเหมาะสมเป็น 1 ดังนั้น

ฟังก์ชันความเหมาะสมคือ $F = \left(\frac{x}{2^l - 1} \right)^n$

SGA Parameters

Population size (popsize)	= 30
Chromosome length (l)	= 30
Maximum # of generation (maxgen)	= 80
Crossover probability (Pc)	= 8.0000000000E-01
Mutation probability (Pm)	= 3.3300000000E-02

Initial Generation Statistics

Initial population max fitness	= 9.4811791207E-01
Initial population average fitness	= 1.0300239870E-01
Initial population minimum fitness	= 5.3711099438E-21
Initial population sum fitness	= 3.0900719609E+00

Population Report

#	Generation 0 string	x	fitness	# parents	pos 1	pos 2	Generation 1 string	x	fitness
1)	111111101010001111000110101100	1.068E+09	0.9481	1) (1, 2)	5	30	111001101010001111000110101100	9.674E+08	0.3523
2)	111101110011111100110101101111	1.037E+09	0.7062	2) (1, 2)	5	30	111101111011111100110101100101	1.039E+09	0.7206
3)	111011000011011100101110001111	9.908E+08	0.4474	3) (3, 3)	16	30	111101000011011110101110001111	1.024E+09	0.6243
4)	111000111100010110000110110010	9.553E+08	0.3109	4) (3, 3)	16	30	111011000011011100101110001111	9.908E+08	0.4474
5)	110111111011000110101100100001	9.382E+08	0.2595	5) (1, 4)	18	30	111111101010001111000110101100	1.068E+09	0.9481
6)	110101110011111111101111010111011	9.028E+08	0.1766	6) (1, 4)	18	30	101000111100010110000110110010	6.869E+08	0.0115
7)	110011101000000000110111000101 0	8.661E+08	0.1166	7) (3, 6)	13	30	111011000011011100101110000111	9.908E+08	0.4474
8)	101111101100010111111000101000111	8.002E+08	0.0528	8) (3, 6)	13	30	110100110011111110111101011011	8.860E+08	0.1464
9)	101100000010001111110111000110	7.388E+08	0.0238	9) (6, 4)	11	30	110101110011111110111100011011	9.028E+08	0.1766
10)	101100000000001100111010011111	7.383E+08	0.0236	10) (6, 4)	11	30	111000111100010110000110110010	9.553E+08	0.3109
11)	101001010001010111110101101000	6.924E+08	0.0124	11) (2, 5)	1	30	111101110011111101110101101111	1.037E+09	0.7062
12)	100111101011111001111001110110	6.658E+08	0.0084	12) (2, 5)	1	30	110111110011000110101100100001	9.361E+08	0.2538
13)	100001010111001101111111111001	5.597E+08	0.0015	13) (3, 7)	12	30	111010001011011100101110101111	9.761E+08	0.3853
14)	011111101001000100110100101001	5.309E+08	0.0009	14) (3, 7)	12	30	110011101010000000110111000101 0	8.667E+08	0.1173
15)	011111000000001011111001001001 0	5.202E+08	0.0007	15) (6, 1)	6	30	110101110011111111011101011011	9.028E+08	0.1766
16)	011101010011100101000001001100	4.917E+08	0.0004	16) (6, 1)	6	30	111111101010001111000110101100	1.068E+09	0.9481
17)	011011100111010100101011110010	4.633E+08	0.0002	17) (1, 1)	15	30	111111111010001111000110101100	1.072E+09	*0.9860*
18)	010010110000011110000101001 0000	3.148E+08	0.0000	18) (1, 1)	15	30	111111101010001111000110101101	1.068E+09	0.9481
19)	001111001001000101110001110000	2.540E+08	0.0000	19) (1, 2)	5	30	111111101010001111000110101100	1.068E+09	0.9481
20)	001101110101001001011100010111	2.320E+08	0.0000	20) (1, 2)	5	30	111101110011101100110101111111	1.037E+09	0.7057
21)	001100110011010000110111111011	2.148E+08	0.0000	21) (3, 6)	1	30	111011000011011100101110001111	9.908E+08	0.4474
22)	001100110010111111100110001111	2.147E+08	0.0000	22) (3, 6)	1	30	110101110011111110111101011111	9.028E+08	0.1766
23)	001100010110001010001010100100	2.071E+08	0.0000	23) (1, 3)	18	30	111110101010001111000110101100	1.051E+09	0.8093
24)	001011100001100001101101110000	1.933E+08	0.0000	24) (1, 3)	18	30	111011000011011100101110011111	9.908E+08	0.4474
25)	000111110110001111101001100111	1.317E+08	0.0000	25) (1, 4)	3	30	111111101100001111000101101100	1.069E+09	0.9528
26)	000100101101001000000111011011	7.894E+07	0.0000	26) (1, 4)	3	30	111000111100010110000010110010	9.553E+08	0.3109
27)	000011010100111011000000000011	5.582E+07	0.0000	27) (1, 5)	1	30	111111101010001111000110101100	1.068E+09	0.9481
28)	000001100001110011011100000010	2.564E+07	0.0000	28) (1, 5)	1	30	110111111111000110101100100001	9.393E+08	0.2624
29)	000001011011011001110111100010	2.396E+07	0.0000	29) (6, 2)	21	30	110101110011111110111101011011	9.028E+08	0.1766
30)	000000100110011111011101101111	1.009E+07	0.0000	30) (6, 2)	21	30	111101110011111100110101100111	1.037E+09	0.7062

Note: Generation 1 & Accumulated Statistics: max=0.9860, min=0.0115, avg=0.5199, sum=15.5984, nmutation=32, ncross= 12

(a) วิวัฒนาการรุ่นที่ 1

รูปที่ 3.11 รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10

Population Report

#	string	Generation 5 x	fitness	# parents	pos 1	2	string	Generation 6 x	fitness
1)	11111111101000111100011010 1100	1.072E+09	0.9860	1) (4,20)	1	30	11100111101000111100011010 1100	9.716E+08	0.3679
2)	11111111101000111100011010 1100	1.072E+09	0.9860	2) (4,20)	1	30	01111100100001111100001010 1010	5.223E+08	0.0007
3)	11111111101000111100011010 1100	1.072E+09	0.9860	3) (18, 3)	27	30	11111010001000011100011010 1100	1.049E+09	0.7930
4)	11111111101000111100011010 1100	1.072E+09	0.9860	4) (18, 3)	27	30	11111111101000111100011010 1100	1.072E+09	0.9860
5)	11111111100000111100100101101	1.072E+09	0.9812	5) (16, 6)	11	30	11111100001101101010111000011 0	1.058E+08	0.8616
6)	11111111100000111100011010 1101	1.072E+09	0.9812	6) (16, 6)	11	30	01111111100000111100011010 1100	5.348E+08	0.0009
7)	11111111100000111100011000 1101	1.072E+09	0.9812	7) (21, 5)	20	30	11111100001000110100010010 1000	1.058E+09	0.8590
8)	11111111010010111101011010 1100	1.071E+09	0.9728	8) (21, 5)	20	30	11111111100000111111010010 0101	1.072E+09	0.9812
9)	11111110110000111100010010 0101	1.069E+09	0.9528	9) (26,16)	7	30	11101100001101111010111000011 0	9.908E+08	0.4474
10)	11111110101000101100011010 0100	1.068E+09	0.9480	10) (26,16)	7	30	01111100001001101010111000011 0	5.207E+08	0.0007
11)	11111110100001011100011010 1101	1.068E+09	0.9446	11) (26, 4)	5	30	11101100001101111010111100011 0	9.908E+08	0.4474
12)	11111101100000111100011110 1100	1.063E+09	0.9071	12) (26, 4)	5	30	11111111101000111100011010 1100	1.072E+09	0.9860
13)	11111100101000111100110011 1000	1.060E+09	0.8762	13) (1, 2)	14	30	11111111101000111100011010 1100	1.072E+09	0.9860
14)	11111100101000111100110010 1000	1.060E+09	0.8762	14) (1, 2)	14	30	11111111111000111100111010 1100	1.073E+09	*0.9957*
15)	11111100101000111001110010 1000	1.060E+09	0.8762	15) (17,12)	1	30	11111010101100011100011010 1100	1.051E+09	0.8110
16)	11111100000110110101011000 0110	1.058E+09	0.8616	16) (17,12)	1	30	11111101100000111100011110 1100	1.063E+09	0.9071
17)	11111010101100011100011010 1100	1.051E+09	0.8110	17) (15,21)	27	30	11111100101000111001100101 1000	1.060E+09	0.8762
18)	11111010001000011100011010 1100	1.049E+09	0.7930	18) (15,21)	27	30	11110100001000111100110010 1000	1.024E+09	0.6223
19)	11111001001110100011010 1011111	1.045E+09	0.7648	19) (1,14)	6	30	11111111101000111100011010 0100	1.072E+09	0.9860
20)	11110100100001111100001010 1000	1.026E+09	0.6323	20) (1,14)	6	30	11110100101000111100110010 1000	1.026E+09	0.6351
21)	11110100001000111100110010 1000	1.024E+09	0.6223	21) (21,29)	7	30	11110100001000111100110010 1010	1.024E+09	0.6223
22)	11110000100010111100000010 1100	1.009E+09	0.5365	22) (21,29)	7	30	11100100011101110010111000 1011	9.583E+08	0.3205
23)	11101110110100111100010110 0100	1.002E+09	0.4994	23) (21,13)	22	30	11110100001010111100110010 1000	1.024E+09	0.6231
24)	11101110100000111100010010 1101	1.000E+09	0.4929	24) (21,13)	22	30	11011100101010111100110011 1000	9.256E+08	0.2265
25)	11101110110000011111000111001 100	9.962E+08	0.4726	25) (13, 6)	22	30	11111100101000111100110011 1000	1.060E+09	0.8762
26)	11101100001101111010111000 0110	9.908E+08	0.4474	26) (13, 6)	22	30	11111111100000111100111010 1101	1.072E+09	0.9812
27)	11101100001101110010111000 0110	9.908E+08	0.4474	27) (19,11)	5	30	11111001001111100011010 1011111	1.045E+09	0.7653
28)	11101011111001011001011010 0010	9.894E+08	0.4414	28) (19,11)	5	30	11111101000101111000110101 1101	1.068E+09	0.9446
29)	11100100011101110010111000 1011	9.583E+08	0.3205	29) (19, 3)	1	30	11111001001110100011010 1011111	1.045E+09	0.7648
30)	01100110111010011110011011 01100	4.313E+08	0.0001	30) (19, 3)	1	30	11111011101000111100011010 1000	1.055E+09	0.8422

Note: Generation 6 & Accumulated Statistics: max=0.9957, min=0.0007, avg=0.6839, sum=20.5181, nmutation=185, ncross= 74

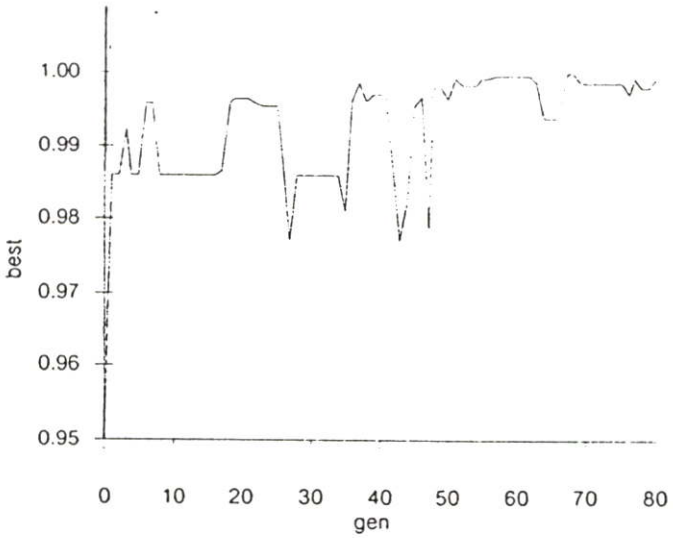
(b) วิวัฒนาการรุ่นที่ 6

รูปที่3.11(ต่อ)

ผลการทดสอบการทำงานของ SGA ในการหาคำตอบ เมื่อกำหนดให้ n=10 และกำหนดค่า SGA พารามิเตอร์คือ l = 30, popsize = 30, maxgen = 30 , P_c = 0.8, P_m = 0.0333 จะเห็นว่าในรุ่นเริ่มต้น (gen = 0) ซึ่งประชากรโครโมโซมเกิดจากการสุ่มนั้นคำตอบที่ดีที่สุดคือ 0.9481 ดังรูปที่ 3.11-a และผลการหาคำตอบของ SGA ในรุ่นที่1 ซึ่งแสดงรายละเอียดการจับคู่โครโมโซมพ่อ-แม่ในคู่ลำดับของ parents และตำแหน่งในการครอสโอเวอร์แลกเปลี่ยนค่าบิทตั้งแต่ตำแหน่ง pos_1 ที่สุ่มได้ถึง pos_2 = 30 ซึ่งหาก pos_1 มีค่าเป็น 1 และ pos_2 มีค่าเป็น 30 นั่นหมายถึงคู่โครโมโซมพ่อ-แม่คู่นั้นไม่มีการครอสโอเวอร์ จะเห็นว่าการครอสโอเวอร์ทำให้โครโมโซมมีความเปลี่ยนแปลงรูปแบบของบิทให้คล้ายกันมากขึ้นในแบบของโครโมโซมที่ดีซึ่งมีค่าความเหมาะสมสูง และแสดงผล

จำนวนการครอสโอเวอร์และจำนวนบิทที่เกิดมิวเตชันทั้งหมดคือ ncross และ nmutation ตามลำดับ รวมทั้งสรุปผลค่าความเหมาะสมสูงสุด ต่ำสุด และ ค่าเฉลี่ย คือ max, min และ avg ตามลำดับ ซึ่งในรุ่นที่1 นี้ SGA สามารถหาคำตอบที่ดีที่สุดคือ max เท่ากับ 0.9860 รูปที่ 3.11-b แสดงผลวิวัฒนาการของ SGA ในรุ่นที่6 จะเห็นได้ว่า รูปแบบโครโมโซมที่ดีเกิดมากยิ่งขึ้นและ SGA สามารถหาคำตอบได้ดีขึ้นเป็น 0.9957

3.4 การประยุกต์จีเนติกอัลกอริทึมแบบง่าย



รูปที่ 3.12 แสดงกราฟค่าความเหมาะสมสูงสุดของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10

วิเคราะห์การทำงานของ SGA ซึ่งเป็น GA ในยุคแรกๆ แล้วจะเห็นว่าวิธีการของ GA เป็นการหาคำตอบแบบสุ่ม ซึ่งเป็นวิธีการที่ไม่มีการบันทึกหรือจดจำคำตอบที่ดีที่สุดของรุ่นก่อน จึงทำให้การหาคำตอบของ SGA ได้คำตอบที่ดีมากขึ้นหรือน้อยลงได้ ดังรูปที่ 3.12 แสดงกราฟคำตอบค่าความเหมาะสมสูงสุดแต่ละรุ่นภายใน 30 รุ่นในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10 จะเห็นได้ว่าคำตอบค่าความเหมาะสมสูงสุดของ SGA มีค่าสูงขึ้นและลดลงด้วยวิธีการทำงานที่อาศัยการสุ่ม ดังนั้นหากสามารถพัฒนาวิธีการค้นหาคำตอบของ GA ให้ดีขึ้นแล้วก็จะเป็นการปรับปรุงสมรรถนะของ GA ยิ่งขึ้น สำหรับหาค่าสูงสุดของฟังก์ชัน $y = x^n$ เราปรับปรุงการค้นหาคำตอบของ GA โดย

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้					โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด					
				ครอสโอเวอร์		มิวเตชัน	ค่าความเหมาะสม		
				pos		*pos*			
1	0 1 1 1 0	196	1	0 1 1 1 0	2	0 1 0 0 1	0 1 0 0 1	81	
2	1 1 0 0 1	625	2	1 1 0 0 1		1 1 1 1 0	2,4 1 *0* 1 *0* 0	400	1 0 1 0 0	400
3	0 1 0 0 0	64	0	1 0 0 1 1	1	1 1 0 0 1	2 1 *0* 0 0 1	289	1 0 0 0 1	289
4	1 0 0 1 1	361	1	1 1 0 0 1		1 0 0 1 1	1 0 0 1 1	361	1 0 0 1 1	361
				2.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่2					**1 1 0 0 1**	**625**

(a) รักษาค่าความเหมาะสมที่ดี 1 โครโมโซม

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้					โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่ 2 และ 4				**1 1 0 0 1**	**625**
								1 0 0 1 1	**361**
				2.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด					
1	0 1 1 1 0	196	1	ครอสโอเวอร์		มิวเตชัน	ค่าความเหมาะสม		
2	1 1 0 0 1	625	2	*pos*		*pos*			
3	0 1 0 0 0	64	0	0 1 1 1 0	2	0 1 0 0 1	0 1 0 0 1	81	0 1 0 0 1	81
4	1 0 0 1 1	361	1	1 1 0 0 1		1 1 1 1 0	2,4 1 *0* 1 *0* 0	400	1 0 1 0 0	400
				2.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่2						

(b) รักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

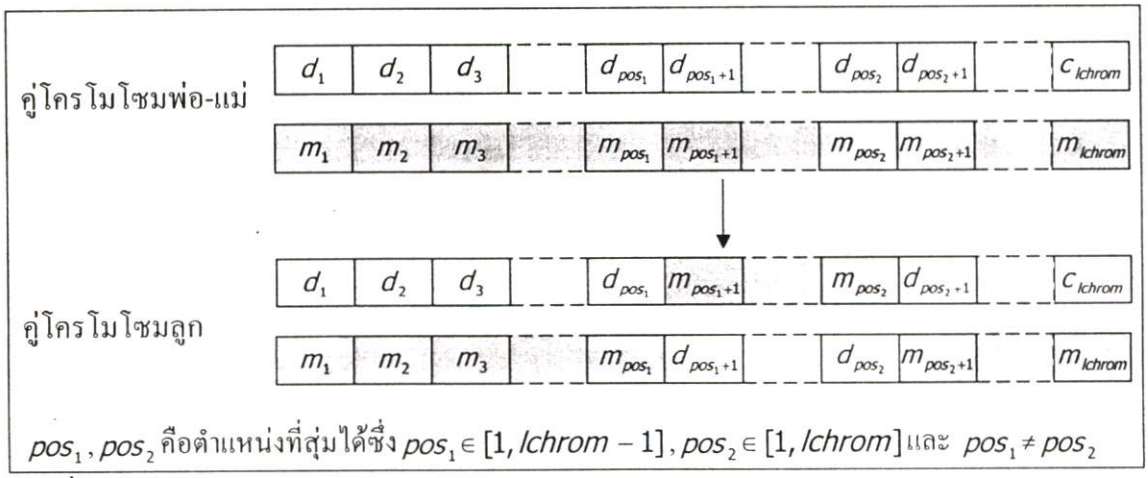
รูปที่ 3.13 รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี

1. โพรดักชันแบบรักษาค่าความเหมาะสมที่ดี เนื่องจากในการค้นหาคำตอบของ SGA นั้น มีโอกาสที่จะสูญเสียโครโมโซมในรุ่นเก่า ที่มีค่าความเหมาะสมที่ดีไปได้ ซึ่งจะทำให้คำตอบในรุ่นถัดไปนั้นดีมากขึ้นหรือน้อยลง ดังนั้นหากปรับปรุง SGA ให้ควบคุมการค้นหาคำตอบ โดยรักษาโครโมโซมที่ดีไว้แล้ว จะช่วยให้วิวัฒนาการคำตอบในรุ่นถัดไปดีขึ้นเรื่อยๆ โดยมีวิธีการดังนี้

- กำหนดจำนวนโครโมโซมที่ดีที่สุด(# best) ของรุ่นเก่าที่ต้องการรักษา เป็น 1,2,4,...
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 1 ให้สร้างชุดโครโมโซมรุ่นใหม่ทั้งหมดแล้วจึงคัดลอก (copy) โครโมโซมที่ดีที่สุดของรุ่นเก่ามาแทนที่โครโมโซมรุ่นใหม่ที่มีค่าความเหมาะสมน้อยที่สุด
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 2,4,... ให้คัดลอกโครโมโซมที่ดีที่สุดในรุ่นเก่าตามจำนวนที่กำหนดมาเป็นโครโมโซมรุ่นใหม่แล้วจึงสร้างโครโมโซมรุ่นใหม่ส่วนที่เหลือต่อไป

ดังรูปที่ 3.13-a แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 1 โครโมโซม และรูปที่ 3.13-b แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

2. ครอสโอเวอร์แบบ 2 จุด การแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่นั้นบางครั้งหากแลกเปลี่ยนค่าบิทเพียงบางช่วงของโครโมโซมแล้วจะสร้างโครโมโซมที่ดีกว่า เช่น การหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ของคู่โครโมโซมพ่อ-แม่ 01110 และ 11001 ซึ่งมีค่าความเหมาะสมเป็น 196 และ 625 หากแลกเปลี่ยนค่าบิทตำแหน่งที่ 3 และ 4 เท่านั้นจะทำให้เกิดโครโมโซมลูกคือ 01000 และ 11111 มีค่าความเหมาะสมเป็น 64 และ 961 ซึ่งโครโมโซม 11111 เป็นโครโมโซมที่ให้คำตอบที่ดีที่สุดที่ต้องการ ดังนั้นการประยุกต์ SGA โดยพัฒนาตัวดำเนินการครอสโอเวอร์ให้เป็นแบบ 2 จุด (Two-point Crossover) จะทำให้ SGA ค้นหาคำตอบที่ดีขึ้นได้ดังรูปที่ 3.14 มีวิธีการดังนี้

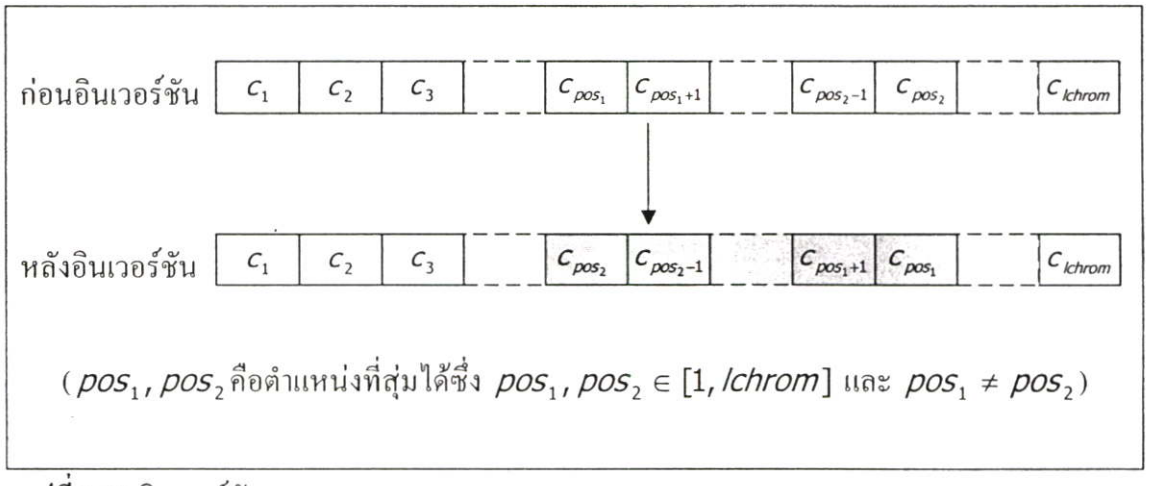


รูปที่ 3.14 ครอสโอเวอร์แบบ 2 จุด

- สุ่มเลือกตำแหน่ง pos_1 และ pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะครอสโอเวอร์ตามลำดับ ซึ่ง pos_1 มีค่าในช่วง $[1, lchrom - 1]$ และ pos_2 มีค่าในช่วง$[1, lchrom]$ โดยที่ pos_1 มีค่าน้อยกว่า pos_2
- แลกเปลี่ยนค่าในแต่ละบิทของคู่โครโมโซมพ่อ-แม่ตั้งแต่ตำแหน่งที่ $pos_1 + 1$ ถึง pos_2

3. ไบนารีมิวเตชันแบบกำหนดค่าบิท เนื่องจากการหาคำตอบของ SGA นั้น กระบวนการไบนารีมิวเตชันอาจทำให้โครโมโซมที่เปลี่ยนแปลงไปให้คำตอบที่ดีลดลงและทำให้สูญเสียโครโมโซมที่ดีไป เช่น โครโมโซม 11110 มีค่าความเหมาะสมเป็น 900 หากสุ่มได้บิทตำแหน่งที่ 1 เกิดมิวเตชันแล้ว โครโมโซมที่เกิดขึ้นจากการมิวเตชันคือ 01110 ทำให้มีค่าความเหมาะสมลดลงเป็น 196 แต่ในบางครั้งข้อดีหรือจุดเด่นของปัญหาจะสามารถนำมาปรับให้เข้ากับการค้นหาคำตอบที่ดีขึ้นได้ สำหรับในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ นี้ค่าบิทของโครโมโซมที่เป็น 1 จะทำให้ค่าความเหมาะสมสูงขึ้นเสมอ ดังนั้นหากปรับปรุงไบนารีมิวเตชันให้เป็นแบบกำหนดค่าแน่นอนให้กับบิทที่เกิดมิเตชัน โดยกำหนดให้บิทที่เกิดมิวเตชันมีค่าบิทเป็น 1 เสมอจะช่วยปรับแนวทางการค้นหาคำตอบของ SGA ดีขึ้น เช่น หากประยุกต์ไบนารีมิวเตชันที่กำหนดค่าให้เป็น 1 เสมอกับโครโมโซม 11110 ในตำแหน่งที่ 1 แล้วโครโมโซมที่เกิดขึ้นจะเหมือนเดิม และยังเป็นการรักษาโครโมโซมที่มีค่าความเหมาะสมที่ดีไว้ด้วย

4. อินเวอร์ชัน (Inversion) เป็นตัวดำเนินการที่ประยุกต์เพิ่มเติมใน SGA โดยจำลองแบบลักษณะของการอินเวอร์ชันในทางพันธุศาสตร์ที่เป็นลักษณะของการกลับหัวกลับหางส่วนของยีนส์ภายในโครโมโซม ที่อาจช่วยให้เกิดโครโมโซมที่ดีขึ้นได้ โดยการกลับส่วนค่าบิทภายในช่วงตำแหน่งของโครโมโซมที่สุ่มได้ตามอัตราค่าความน่าจะเป็นของการอินเวอร์ชันแต่ละโครโมโซม (Probabilityu of Inversion : P_i) ที่กำหนด ดังรูปที่ 3.15 มีขั้นตอนดังนี้



รูปที่ 3.15 อินเวอร์ชัน

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะอินเวอร์ชันตามลำดับ ซึ่ง pos_1 และ pos_2 มีค่าอยู่ในช่วง $[1, lchrom]$ โดยที่ pos_1 มีค่าน้อยกว่า pos_2
- กลับค่าบิทในช่วงของตำแหน่งที่ pos_1 ถึง pos_2 ของโครโมโซมโดยสลับค่าบิท pos_1 กับ pos_2 , pos_1 +1 กับ pos_2 -1 , pos_1 +2 กับ pos_2 -2 ,...

เช่น สุ่มโครโมโซมที่จะอินเวอร์ชันคือ 01010 มีค่าความเหมาะสมเป็น 100 โดยสุ่มตำแหน่ง pos_1 =1 และ pos_2 = 4 แล้วจะเห็นว่าการอินเวอร์ชันทำให้เกิดโครโมโซม 10100 ซึ่งมีค่าความเหมาะสมดีขึ้นเป็น 400 เป็นต้น สำหรับจำนวนการอินเวอร์ชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า P_i ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม และกำหนดให้ P_i = 0.1 แล้วจำนวนการอินเวอร์ชันในแต่ละรุ่นเท่ากับ P_i *popsize = 0.1*30 = 3 ครั้ง

ผลการทำงานโดยสรุปของการประยุกต์ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n =10 ภายใน 30 รุ่น ซึ่งกำหนดค่า SGA พารามิเตอร์ต่างๆ เปรียบเทียบดังตารางที่ 3.2 แสดงดังรูปที่ 3.16-a ถึง 3.16-g โดยเปรียบเทียบการหาคำตอบที่มีกำหนดให้มีการมิวเตชันในรูปที่ 3.16-b นั้นดีกว่าการหาคำตอบเพียงครอสโอเวอร์เท่านั้นในรูปที่ 3.16-a และจะเห็นได้ว่า SGA สามารถหาคำตอบได้ดีขึ้นเมื่อมีการปรับปรุงวิธีการต่างๆ คือ หากใช้รีโพรดักชันแบบรักษาค่าที่ดีจำนวน 4 โครโมโซม และครอสโอเวอร์แบบ 2 ตำแหน่งในรูปที่ 3.16-e แล้ว ผลการทำงานของ SGA สามารถหาคำตอบที่ดีที่สุดเท่ากับ 1.0 ได้ ซึ่งดีกว่าในรูปที่3.16-d เมื่อรีโพรดักชันแบบรักษาค่าที่ดีจำนวน 1 โครโมโซมที่สามารถหาคำตอบได้เป็น 0.9999 และดีกว่าการรีโพรดักแบบทั่วไปในรูปที่ 3.16-a , 3.16-b ,3.16-c และ 3.16-d ที่ยังไม่สามารถหาคำตอบทีดีที่สุดได้ภายในการดำเนินการ 30 รุ่น สำหรับในรูปที่ 3.16-f นั้นเมื่อเพิ่มตัวดำเนินการอินเวอร์ชันแล้ว ผลการดำเนินงานนั้นสามารถคำตอบที่ดีที่สุดเท่ากับ 1.0 ได้เร็วขึ้นในรุ่นที่ 20 และเมื่อเพิ่มเทคนิคการกำหนดค่าบิทของการมิวเตชันให้เป็น 1 เสมอแล้ว ก็ยิ่งทำให้ SGA ทำงานได้ดียิ่งขึ้น โดยสามารถหาคำตอบที่ดีที่สุดได้เมื่อดำเนินงานผ่านไปเพียง 9 รุ่นเท่านั้น ดังในรูปที่ 3.16-g

ตารางที่3.2 การกำหนด SGA พารามิเตอร์เปรียบเทียบผลการทำงาน

ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10

รูปที่	P_c	P_m	P_i	จำนวนการรักษาโครโมโซมรุ่นเก่า (# best)	จำนวนตำแหน่งในการครอสโอเวอร์ (pt. cross)	การกำหนดค่าบิทมิวเตชัน (fix value)
3.16-a	0.8000	0.0000	0.0000	0	1	0->1และ1->0
3.16-b	0.8000	0.0333	0.0000	0	1	0->1และ1->0
3.16-c	0.8000	0.0333	0.0000	0	2	0->1และ1->0
3.16-d	0.8000	0.0333	0.0000	4	2	0->1และ1->0
3.16-e	0.8000	0.0333	0.0000	1	2	0->1และ1->0
3.16-f	0.8000	0.0333	0.3000	4	2	0->1และ1->0
3.16-g	0.8000	0.0333	0.3000	4	2	0 ,1 ->1

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0000 pt.crs = 1
Max.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	*0.9481*	0.1030	0.0000
1	0.9481	0.5498	0.1166
2	0.9481	0.7032	0.1766
3	0.9481	0.7352	0.3109
4	0.9481	0.8152	0.4474
5	0.9481	0.8819	0.4474
6	0.9481	0.9153	0.4474
7	0.9481	0.9320	0.7062
8	0.9481	0.9401	0.7062
9	0.9481	0.9401	0.7062
10	0.9481	0.9481	0.9481
11	0.9481	0.9481	0.9481
12	0.9481	0.9481	0.9481
13	0.9481	0.9481	0.9481
14	0.9481	0.9481	0.9481
15	0.9481	0.9481	0.9481
16	0.9481	0.9481	0.9481
17	0.9481	0.9481	0.9481
18	0.9481	0.9481	0.9481
19	0.9481	0.9481	0.9481
20	0.9481	0.9481	0.9481
21	0.9481	0.9481	0.9481
22	0.9481	0.9481	0.9481
23	0.9481	0.9481	0.9481
24	0.9481	0.9481	0.9481
25	0.9481	0.9481	0.9481
26	0.9481	0.9481	0.9481
27	0.9481	0.9481	0.9481
28	0.9481	0.9481	0.9481
29	0.9481	0.9481	0.9481
30	0.9481	0.9481	0.9481

(a)

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0333 pt.crs = 1
Max.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9860	0.5199	0.0115
2	0.9860	0.5905	0.0150
3	0.9921	0.5721	0.1563
4	0.9860	0.6123	0.0009
5	0.9860	0.7462	0.0001
6	0.9957	0.6839	0.0007
7	0.9957	0.6596	0.0007
8	0.9860	0.5824	0.0000
9	0.9859	0.6774	0.0553
10	0.9859	0.6368	0.0000
11	0.9859	0.7170	0.0007
12	0.9859	0.8187	0.3933
13	0.9859	0.7007	0.0006
14	0.9859	0.6491	0.0007
15	0.9860	0.8273	0.0455
16	0.9860	0.8119	0.0009
17	0.9865	0.8549	0.0205
18	0.9957	0.8328	0.0005
19	*0.9963*	0.8734	0.2589
20	0.9963	0.8912	0.2354
21	0.9963	0.7875	0.0089
22	0.9957	0.7705	0.0002
23	0.9954	0.8397	0.0009
24	0.9954	0.7015	0.0002
25	0.9953	0.7313	0.0009
26	0.9883	0.7610	0.0000
27	0.9771	0.7299	0.0000
28	0.9860	0.7715	0.0001
29	0.9859	0.8475	0.2015
30	0.9859	0.7080	0.0005

(b)

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5559	0.0005
2	0.9957	0.5741	0.0009
3	0.9969	0.7093	0.0358
4	0.9960	0.7531	0.2475
5	0.9971	0.7064	0.0008
6	0.9971	0.8505	0.0330
7	0.9770	0.7278	0.0001
8	0.9911	0.7542	0.0002
9	0.9911	0.7146	0.0007
10	0.9952	0.7890	0.0009
11	0.9960	0.7329	0.0000
12	0.9956	0.8780	0.3621
13	0.9956	0.7700	0.0000
14	0.9956	0.7285	0.0501
15	0.9958	0.7667	0.0009
16	0.9958	0.7655	0.0003
17	0.9958	0.7429	0.0006
18	0.9957	0.7776	0.0007
19	*0.9982*	0.7790	0.0009
20	0.9982	0.7279	0.0009
21	0.9982	0.7525	0.0432
22	0.9982	0.7452	0.0009
23	0.9982	0.7096	0.0002
24	0.9976	0.8281	0.2434
25	0.9861	0.6068	0.0000
26	0.9923	0.8072	0.0009
27	0.9887	0.7854	0.0006
28	0.9958	0.7924	0.0005
29	0.9959	0.8595	0.4295
30	0.9959	0.7752	0.0008

(c)

Pop.size = 30 Pc = 0.8000 # best = 1
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5875	0.0238
2	0.9957	0.6060	0.0194
3	0.9963	0.7498	0.0524
4	0.9969	0.7991	0.2620
5	0.9969	0.7512	0.0512
6	0.9969	0.8960	0.4962
7	0.9969	0.7753	0.0005
8	0.9980	0.8075	0.0450
9	0.9980	0.7623	0.0050
10	0.9981	0.8387	0.0308
11	0.9981	0.7796	0.0005
12	0.9981	0.9174	0.4760
13	0.9981	0.8194	0.2426
14	0.9987	0.7726	0.0554
15	0.9987	0.8230	0.0219
16	0.9987	0.8139	0.2426
17	0.9987	0.7908	0.0009
18	0.9987	0.8296	0.2605
19	0.9987	0.8628	0.2441
20	0.9987	0.7987	0.0367
21	0.9987	0.8101	0.0526
22	0.9987	0.8145	0.0425
23	0.9988	0.7884	0.0010
24	0.9988	0.9105	0.4667
25	0.9988	0.6898	0.0005
26	0.9988	0.8623	0.2399
27	0.9988	0.8495	0.0476
28	0.9988	0.8530	0.0007
29	*0.9999*	0.8977	0.6664
30	0.9999	0.8486	0.0009

(d)

รูปที่ 3.16 ผลการทำงานโดยสรุปของ GA ในการหาค่าสูงสุดของฟังก์ชัน $y=x^n$ ที่ $n=10$

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5857	0.0005
2	0.9957	0.6655	0.0009
3	0.9963	0.7556	0.0524
4	0.9963	0.7705	0.0524
5	0.9963	0.8069	0.0524
6	0.9975	0.7939	0.0009
7	0.9975	0.8738	0.0454
8	0.9975	0.8026	0.0001
9	0.9987	0.8697	0.2170
10	0.9987	0.7446	0.0002
11	0.9997	0.8911	0.0347
12	0.9997	0.8241	0.0005
13	0.9997	0.8020	0.0000
14	0.9998	0.9362	0.2589
15	0.9998	0.8406	0.0000
16	0.9998	0.7955	0.0546
17	0.9998	0.8405	0.0010
18	0.9998	0.9286	0.2590
19	0.9998	0.7849	0.0007
20	0.9998	0.8373	0.0007
21	0.9998	0.8608	0.0008
22	0.9998	0.8767	0.0010
23	0.9998	0.7984	0.0010
24	0.9998	0.8458	0.0456
25	0.9998	0.8151	0.0010
26	0.9998	0.8314	0.0002
27	0.9998	0.8100	0.0527
28	0.9998	0.9182	0.0475
29	*1.0000*	0.6918	0.0000
30	1.0000	0.8837	0.0010

(e)

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.2000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9648	0.5539	0.0001
2	0.9648	0.6905	0.0004
3	0.9803	0.7530	0.0009
4	0.9803	0.7691	0.0224
5	0.9900	0.7140	0.0004
6	0.9900	0.7182	0.0042
7	0.9900	0.7613	0.0350
8	0.9997	0.7830	0.0009
9	0.9997	0.7269	0.0009
10	0.9998	0.8061	0.0352
11	0.9998	0.7000	0.0181
12	0.9998	0.7294	0.0455
13	0.9998	0.8265	0.0563
14	0.9998	0.8679	0.0009
15	0.9998	0.8829	0.0236
16	0.9998	0.8733	0.0006
17	0.9998	0.7911	0.0010
18	0.9998	0.8801	0.0009
19	0.9998	0.8065	0.0009
20	*1.0000*	0.8335	0.0009
21	1.0000	0.8084	0.0006
22	1.0000	0.8212	0.0009
23	1.0000	0.8201	0.0008
24	1.0000	0.8541	0.1786
25	1.0000	0.8065	0.0202
26	1.0000	0.8479	0.0559
27	1.0000	0.8045	0.0008
28	1.0000	0.7436	0.0003
29	1.0000	0.7853	0.0028
30	1.0000	0.8771	0.0008

(f)

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.2000 fix val. = 1

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9481	0.5959	0.0156
2	0.9860	0.8297	0.0862
3	0.9860	0.8738	0.1108
4	0.9860	0.9068	0.6612
5	0.9951	0.9356	0.7049
6	0.9951	0.9670	0.7062
7	0.9956	0.9723	0.7062
8	0.9997	0.9875	0.9481
9	*1.0000*	0.9818	0.8139
10	1.0000	0.9484	0.0427
11	1.0000	0.9837	0.9521
12	1.0000	0.9572	0.5236
13	1.0000	0.9582	0.5236
14	1.0000	0.9702	0.5496
15	1.0000	0.9574	0.5496
16	1.0000	0.9374	0.5496
17	1.0000	0.9078	0.5245
18	1.0000	0.9413	0.3657
19	1.0000	0.9774	0.4798
20	1.0000	0.9799	0.4798
21	1.0000	0.9884	0.6714
22	1.0000	0.9595	0.5245
23	1.0000	0.9621	0.5030
24	1.0000	0.9678	0.6714
25	1.0000	0.9794	0.8543
26	1.0000	0.9754	0.8543
27	1.0000	0.9801	0.8543
28	1.0000	0.9796	0.8543
29	1.0000	0.9817	0.8543
30	1.0000	0.9625	0.0456

(g)

รูปที่ 3.16(ต่อ)

สรุป

จีเนติกอัลกอริทึมเป็นทฤษฎีที่เลียนแบบกระบวนการวิวัฒนาการทางธรรมชาติ ที่นำมาประยุกต์ใช้กับคอมพิวเตอร์เพื่อช่วยแก้ปัญหาในการหาคำตอบต่างๆ ซึ่งมีพื้นฐานการทำงานเบื้องต้นเป็นจีเนติกอัลกอริทึมแบบง่าย โดยมีรูปแบบโครโมโซมเป็นแบบไบนารี และตัวดำเนินการทางพันธุศาสตร์คือ การครอสโอเวอร์และมิวเตชันที่ไม่ซับซ้อน แต่สามารถปรับปรุงให้เข้ากับปัญหาเพื่อช่วยให้ จีเนติกอัลกอริทึมมีแนวทางในการค้นหาคำตอบที่ดีขึ้น ทั้งนี้สามารถที่จะประยุกต์ใช้จีเนติกอัลกอริทึมในการแก้ปัญหารูปแบบต่างๆได้ โดยจะต้องมีวิธีการปรับปรุงรูปแบบโครโมโซมหรือตัวดำเนินการทางพันธุศาสตร์เพื่อให้เหมาะสมกับแต่ละปัญหา

บทที่ 4
การนำจีเนติกอัลกอริทึมมาใช้ในการออกแบบโครงข่าย ATM

4.1 โมเดลของระบบ

ในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีที่มีสวิตช์หรือโหนดเชื่อมต่อกันด้วยสายสัญญาณเพื่อใช้ในการส่งเซลล์ข้อมูลนั้น ปัจจัยหลักที่จะใช้พิจารณาว่าการออกแบบนั้นมีความเหมาะสมหรือไม่คือ ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่าย (Average Cell Delay) และ ค่าใช้จ่ายในระบบสายสัญญาณของโครงข่าย ในวิทยานิพนธ์นี้จะคำนวณหาค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายได้โดยใช้แบบจำลอง M/D/1 สำหรับในการออกแบบโครงข่ายแบบทรี รูปแบบการเชื่อมต่อระหว่างโหนดต่างๆในโครงข่ายเราได้นำกระบวนการทางจีเนติกอัลกอริทึม(Genetic Algorithm) มาใช้ในการออกแบบโดยกระบวนการทาง จีเนติกนี้เป็นกระบวนการในการหาคำตอบที่ดีที่สุด (Optimization) ของปัญหาต่างๆจากจำนวนคำตอบที่เป็นไปได้ทั้งหมดของการแก้ปัญหานั้น ซึ่งในวิทยานิพนธ์นี้การออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรี เราได้พิจารณาค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายและค่าใช้จ่ายในการติดตั้งหรือเช่าสายสัญญาณเป็นปัจจัยหลัก โดยทำการเข้ารหัสรูปแบบต่างๆของโครงข่ายโดยใช้ Prüfer Number จากนั้นจึงแทนด้วยโครโมโซม ซึ่งโครโมโซมเหล่านั้นจะถูกนำเข้าสู่กระบวนการจีเนติกต่างๆอันได้แก่ การคัดเลือก การครอสโอเวอร์ การมิวเตชัน และการรีโพรดักชัน จนกระทั่งได้โครโมโซมที่มีความเหมาะสมที่สุดจากนั้นทำการถอดรหัสโครโมโซมที่ได้ก็จะได้รูปแบบการเชื่อมต่อของโครงข่ายที่เหมาะสมที่สุด

4.2 การใช้จีเนติกในการออกแบบโครงข่าย ATM

ในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรี จะมีกระบวนการต่างๆดังนี้

4.2.1 กำหนดรูปแบบโครโมโซม (Chromosome)

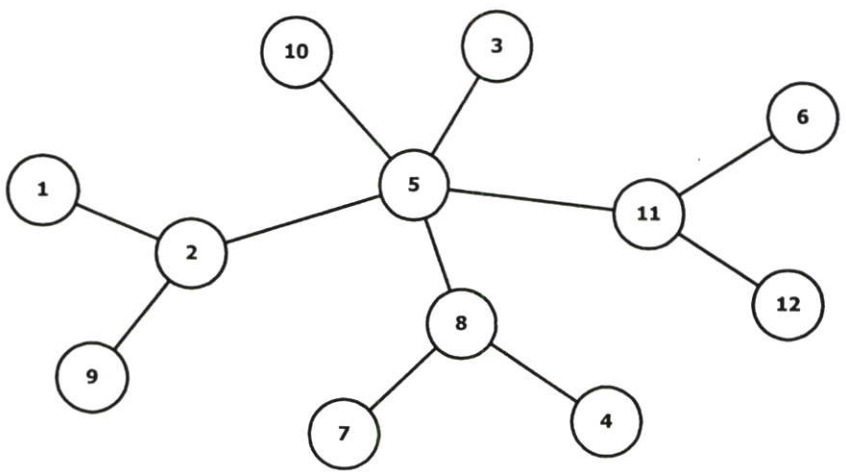
ในการกำหนดรูปแบบโครโมโซมนั้นสำหรับในวิทยานิพนธ์นี้จะใช้ Prüfer Number ในการเข้ารหัสและถอดรหัสรูปแบบต่างของโครงข่ายที่เป็นแบบทรี ซึ่งลักษณะของโครโมโซมจะประกอบด้วยบิทที่เป็นตัวเลขตั้งแต่ 1 ถึงจำนวนโหนดทั้งหมดที่ต้องการออกแบบและจำนวนบิททั้งหมดของโครโมโซมจะน้อยกว่าจำนวนโหนดอยู่สองจำนวน ซึ่งจากรูป 4.1 เป็นตัวอย่างที่แสดงความสัมพันธ์ของโครงสร้างแบบทรี กับ Prüfer Number โดยในการเข้ารหัสและถอดรหัสเมื่อพิจารณาโครงข่ายที่มีจำนวนโหนดเท่ากับ N มีวิธีการดังนี้

1) การเข้ารหัส Prüfer Number

- ให้โหนด j เป็นโหนดที่มีหมายเลขน้อยที่สุดที่เป็นจุดปลายของทรี
- มีโหนด k ซึ่งเป็นโหนดที่เชื่อมต่อกับโหนด j เราก็จะได้ว่า k เป็นหมายเลขในรหัสของ Prüfer Number ซึ่งจะเรียงจากซ้ายไปขวา
- ย้ายโหนด j ออกจากการเชื่อมต่อระหว่าง j กับ k
- ทำซ้ำขั้นตอนแรกใหม่จนกระทั่งจำนวนของ Prüfer Number มีค่า N-2

2) การถอดรหัส Prüfer Number

- ให้ P เป็น Prüfer Number และ $\overline{P}$ เป็นเซตของหมายเลขโหนดที่ไม่ได้อยู่ใน P
- ให้ j เป็นหมายเลขน้อยสุดใน $\overline{P}$ และ k เป็นหมายเลขที่อยู่ซ้ายสุดใน P แล้วจะได้การเชื่อมต่อของโหนด j กับ k จากนั้นย้าย j ออกจาก $\overline{P}$ และย้าย k ออกจาก P แล้วถ้าหมายเลข k ไม่มีอยู่ใน P ก็ทำการเพิ่ม k ลงใน $\overline{P}$
- ทำซ้ำขั้นตอนแรกจนกระทั่งไม่มีหมายเลขใน P
- ถ้าไม่มีหมายเลขใน P จะเหลือหมายเลขใน $\overline{P}$ อยู่สองหมายเลข จากนั้นจึงทำการเชื่อมต่อโหนดทั้งสองหมายเลขเข้าด้วยกัน ก็จะได้โครงสร้างทรีที่มีจำนวนการเชื่อมต่อทั้งหมด N-1



รูปที่4.1 แสดงความสัมพันธ์ของโครงสร้างแบบทรี กับ Prüfer Number

เมื่อเรากำหนดรูปแบบโครโมโซมได้แล้วก็จะทำการสร้างประชากรเริ่มต้น(Initial Population) หรือจำนวนชุดโครโมโซมเริ่มต้นเพื่อนำเสนอรูปแบบโครงข่ายที่เป็นไปได้โดยทำการกำหนด Prüfer Number ซึ่งแต่ละโครโมโซมจะมีความยาวเท่ากับ N-2 และรหัสบิทมีค่าตั้งแต่ 1 ถึง N

4.2.2 การวัดค่าความเหมาะสม (Fitness Value)

ในวิทยานิพนธ์นี้เราจะใช้ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายและค่าใช้จ่ายของสายสัญญาณในโครงข่ายเป็นปัจจัยในการวัดค่าความเหมาะสมโดยโครโมโซมที่มีค่าความเหมาะสมที่สุดจะต้องมีค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายน้อยกว่าค่าดีเลย์เฉลี่ยสูงสุดที่เรากำหนดและมีค่าใช้จ่ายน้อยที่สุดซึ่งแสดงได้ดังนี้

$$\text{Fitness Value} = \text{Minimize}\{D\}$$

$$\text{เงื่อนไข} \qquad T < T_{\max}$$

โดย

$$T = \frac{1}{\gamma}\sum_{i=1}^{M}\frac{2\mu C_i\lambda_i - \lambda_i^{\,2}}{2\mu C_i(\mu C_i - \lambda_i)}$$

$$D = \sum_{i=1}^{M} d_i(C_i)$$

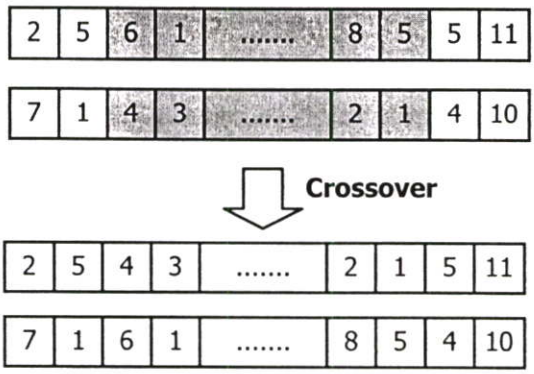
ซึ่งค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่าย (T) และ ค่าใช้จ่ายในระบบสายสัญญาณของโครงข่าย(D) ได้กล่าวโดยละเอียดมาแล้วในบทที่ 2

4.2.3 การคัดเลือก (Selection)

เป็นการคัดเลือกโครโมโซมให้เป็นโครโมโซมต้นแบบเพื่อใช้ในการสร้างโครโมโซมในรุ่นถัดไป สำหรับการคัดเลือกนั้นจะพิจารณาค่าความเหมาะสมเป็นหลักโดยโครโมโซมที่มีความเหมาะสมที่ดีจะมีโอกาสถูกเลือกในแต่ละครั้งสูงเพื่อเป็นโครโมโซมต้นแบบในการสร้างโครโมโซมในรุ่นถัดไป ซึ่งจะเป็นการถ่ายทอดส่วนที่ดีให้ในรุ่นต่อไป

4.2.4 การครอสโอเวอร์ (Crossover)

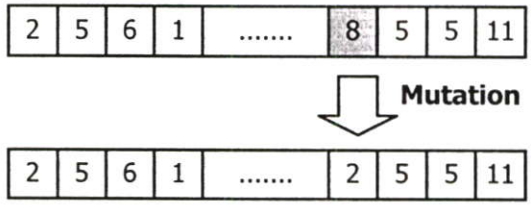
เป็นการแลกเปลี่ยนส่วนของโครโมโซมระหว่างสองโครโมโซมเพื่อสร้างชุดโครโมโซมใหม่ที่แตกต่างไปจากโครโมโซมเดิม ซึ่งจะนำโครโมโซมที่ถูกคัดเลือกให้เป็นโครโมโซมพ่อ-แม่มาทำการครอสโอเวอร์กัน จากรูปเป็นการทำครอสโอเวอร์แบบสองจุด



รูปที่4.2 การทำครอสโอเวอร์

4.2.5 การมิวเตชัน (Mutation)

เป็นการเปลี่ยนบางส่วนในโครโมโซมทำให้โครโมโซมเดิมนั้นกลายเป็นโครโมโซมใหม่ ซึ่งอาจทำให้โครโมโซมใหม่ที่ได้นั้นมีค่าความเหมาะสมสูงขึ้น



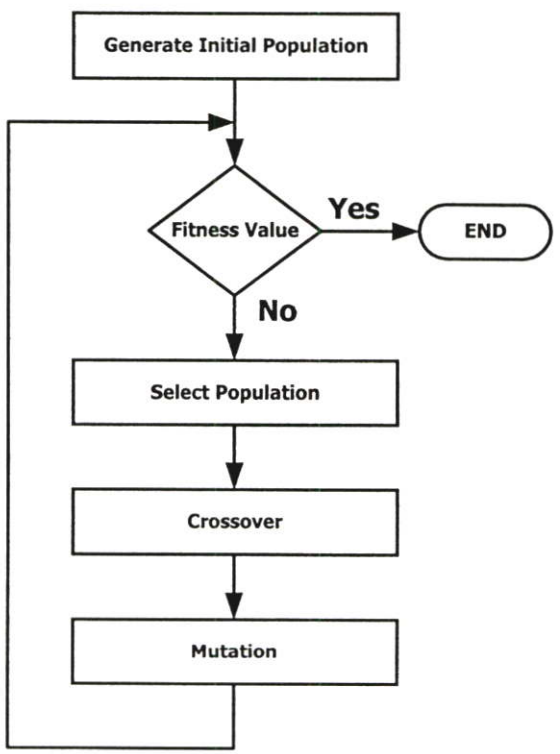
รูปที่4.3 การทำมิวเตชัน

4.2.6 การรีโพรดักชัน (Reproduction)

โครโมโซมรุ่นใหม่จะถูกถ่ายทอดกลายเป็นโครโมโซมรุ่นเก่าสำหรับวิวัฒนาการในรุ่นถัดไปโดยใช้การรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดีไว้จะช่วยให้วิวัฒนาการของคำตอบในรุ่นถัดไปดีขึ้นเรื่อยๆ ในการวิจัยนี้จะใช้การรีโพรดักชันแบบรักษาค่าที่ดี 1 โครโมโซม โดยการนำโครโมโซมที่มีความเหมาะสมมากที่สุดในรุ่นที่แล้วมาแทนโครโมโซมที่มีความเหมาะสมน้อยที่สุดของโครโมโซมรุ่นใหม่

ในการออกแบบโครงข่ายนั้นจะทำตามกระบวนการต่างๆตามที่ได้กล่าวมาแล้วจนกระทั่งได้โครโมโซมที่มีความเหมาะสมที่สุด ซึ่งกระบวนการต่างๆจะเป็นดังรูปที่4.4 จะเห็นได้ว่าเมื่อทำการสร้างประชากรเริ่มต้นแล้วก็จะเป็นการหาค่าความเหมาะสมโดยถ้าในประชากรรุ่นนั้นมีโครโมโซมที่มีความเหมาะสมตามเงื่อนไขของการพิจารณาการวัดค่าความเหมาะสม ก็จะเป็นการจบการทำงานและนำโครโมโซมนั้นมาทำการถอดรหัส Prüfer Number ซึ่งก็จะได้รูปแบบการเชื่อม

ต่อของโครงที่เป็นแบบทรีที่มีความเหมาะสมที่สุดตามที่ต้องการออกมา แต่ถ้าในประชากรเริ่มต้นนั้นยังไม่มีโครโมโซมที่มีความเหมาะสมตามที่ต้องการก็จะทำกระบวนการต่อไปซึ่งก็คือในส่วนของการคัดเลือกดังที่ได้กล่าวมาแล้วว่าจะทำการคัดเลือกโครโมโซมในรุ่นนั้นๆที่มีความเหมาะสมสูงให้มีโอกาสในการถูกคัดเลือกเป็นโครโมโซมพ่อ-แม่สูง จากนั้นก็จะเป็นการนำโครโมโซมที่ถูกคัดเลือกมาทำการครอสโอเวอร์และมิวเตชันตามลำดับเพื่อเป็นการสร้างโครโมโซมรุ่นใหม่ที่จะช่วยในการสร้างวิวัฒนาการของคำตอบให้ดีขึ้นหรือมีความเหมาะสมมากขึ้น เมื่อได้โครโมโซมรุ่นใหม่มาจะทำการวัดค่าความเหมาะสมของโครโมโซมรุ่นใหม่นั้นและทำการรีโพรดักชันซึ่งในการวิจัยนี้จะใช้การรีโพรดักชันแบบรักษาค่าที่ดี 1 โครโมโซม โดยการนำโครโมโซมที่มีความเหมาะสมมากที่สุดในรุ่นที่แล้วมาแทนโครโมโซมที่มีความเหมาะสมน้อยที่สุดของโครโมโซมรุ่นใหม่ เมื่อได้โครโมโซมรุ่นใหม่ก็จะกลับมาวัดค่าความเหมาะสมอีกครั้งหนึ่งและถ้าได้โครโมโซมที่มีความเหมาะสมที่สุดตามที่ต้องการก็จะจบการทำงาน แต่ถ้ายังไม่มีโครโมโซมที่มีความเหมาะสมที่สุด กระบวนการต่างๆที่ได้กล่าวมาก็จะถูกทำซ้ำๆจนกระทั่งได้โครโมโซมที่มีความเหมาะสมที่สุดหรือจนกระทั่งถึงจำนวนรุ่นที่มากที่สุดที่ต้องการ ก็จะเป็นการจบการทำงานของโปรแกรม และนำโครโมโซมที่มีความเหมาะสมที่สุดมาถอดรหัสก็จะได้รูปแบบการเชื่อมต่อของโครงข่ายที่มีความเหมาะสุดที่สุดออกมา



รูปที่ 4.4 แผนผังการทำงานของกระบวนการทางจีเนติกอัลกอริทึม

4.3 ผลการทดลอง

ในหัวข้อนี้จะแสดงการทดลองการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีโดยใช้จีเนติกอัลกอริทึม โดยกำหนดให้ต้องการออกแบบโครงข่ายที่มีจำนวนโหนดเท่ากับ 8โหนด โดยมีอัตราเฉลี่ยในการส่งข้อมูลระหว่างโหนดดังตารางที่1 ค่าระยะห่างระหว่างโหนดดังตารางที่2 และ อัตราค่าใช้จ่ายของความจุช่องสัญญาณ 150 Mbps มีค่า 1 หน่วยต่อกิโลเมตร และ 300 Mbps มีค่า 2 หน่วยต่อกิโลเมตร

ตารางที่4.1 อัตราเฉลี่ยในการส่งข้อมูลระหว่างโหนด(Mbps)

Node	1	2	3	4	5	6	7	8
1	0	18	20	5	10	5	20	10
2	18	0	10	5	8	3	10	5
3	20	10	0	10	20	10	10	20
4	5	5	10	0	5	10	3	15
5	10	8	20	5	0	5	5	10
6	5	3	10	10	5	0	3	20
7	20	10	10	3	5	3	0	5
8	10	5	20	15	10	20	5	0

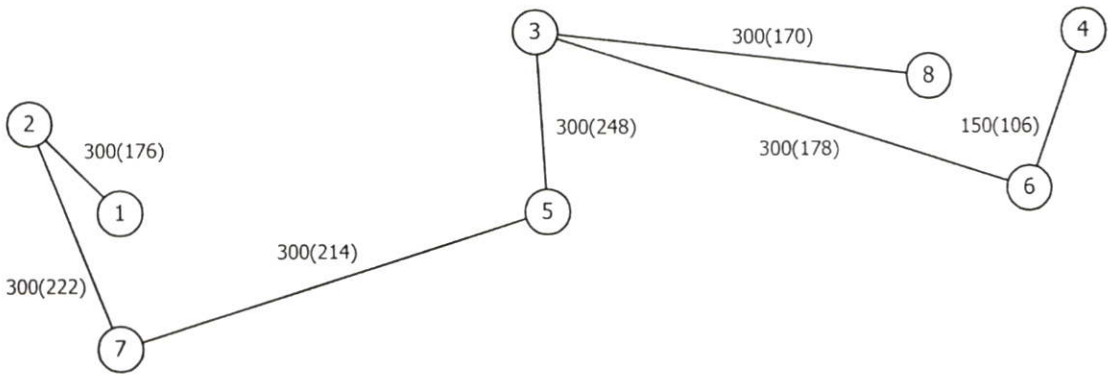
ตารางที่4.2 ระยะห่างระหว่างโหนด(Km)

Node	1	2	3	4	5	6	7	8
1	0	11	60	134	57	124	15	110
2	11	0	70	146	71	137	27	123
3	60	70	0	72	20	66	70	50
4	134	146	72	0	74	18	140	18
5	57	71	20	74	0	62	61	50
6	124	137	66	18	62	0	126	16
7	15	27	70	140	61	126	0	116
8	110	123	50	18	50	16	116	0

ทำการออกแบบโครงข่าย ATM ที่มีจำนวนโหนดทั้งหมด 8 โหนดโดยใช้กระบวนการจีเนติกอัลกอริทึมในการหารูปแบบโครงข่ายที่มีความเหมาะสมที่สุด โดยกำหนดให้ค่าจำนวนประชากร (Popsize) ในแต่ละรุ่นเท่ากับ 10 , ค่าจำนวนรุ่นที่มากที่สุด(Maxgen) มีค่าเท่ากับ 200 ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายสูงสุดที่ยอมรับได้ (T_{max}) มีค่าเท่ากับ 0.01 ms โดยให้หาค่าความเหมาะสม (Fitness Value) ตามค่าต่างๆดังนี้

ค่าความเหมาะสม (Fitness Value) = 500 หน่วย

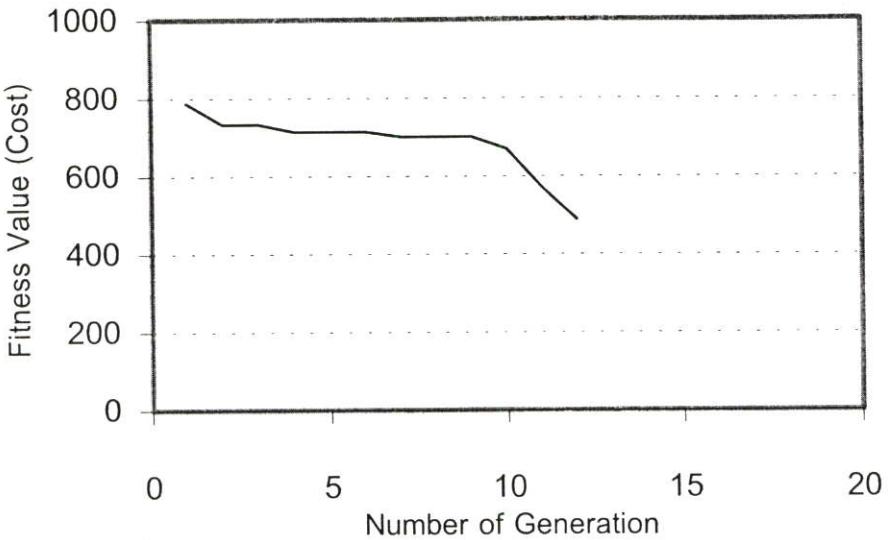
ทำการประมวลผลจนกระทั่งได้โครโมโซมที่มีความเหมาะสมที่สุดตามที่ต้องการ คือ 276353 มีค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายเท่ากับ 0.008 ms และมีค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายมีค่าเท่ากับ 488 หน่วย โดยมีรูปแบบการเชื่อมต่อของโครงข่ายเป็นดังรูปที่ 4.5



รูปที่ 4.5 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมเมื่อ Fitness Value = 500 หน่วย

จากรูปที่ 4.5 จะเห็นได้ว่าการเชื่อมต่อกันดังรูปนี้ได้มาจากการถอดรหัส Prüfer Number ของโครโมโซม 276353 และในส่วนของค่าความจุของช่องสัญญาณและค่าอัตราเฉลี่ยในการส่งข้อมูลของแต่ละช่องสัญญาณจะแสดงดังตัวเลขที่อยู่ระหว่างการเชื่อมต่อของโหนดต่างๆ

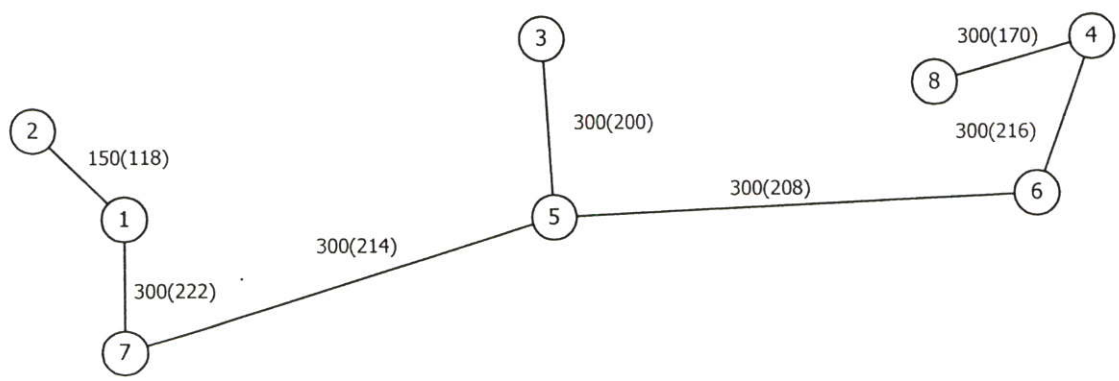
สำหรับกรณีนี้ในกระบวนการจีเนติกอัลกอริทึมโครโมโซมที่มีความเหมาะสมที่สุดตามที่ต้องการพบว่าได้ในประชากร(Generation) รุ่นที่ 12 ซึ่งแนวโน้มของการออกแบบโดยใช้จีเนติกอัลกอริทึมเป็นดังรูปที่ 4.6 ซึ่งแสดงค่าความเหมาะสมของประชากรในแต่ละรุ่นจะเห็นได้ว่ากราฟของกระบวนการ จีเนติกอัลกอริทึมในแต่ละรุ่นจะมีค่าความเหมาะสมดีขึ้นเรื่อยๆจนกระทั่งถึงรุ่นที่ 12 โปรแกรมก็จะหยุดการทำงานเพราะว่าสามารถหาโครโมโซมที่มีค่าความเหมาะสมที่สุดตามที่ต้องการได้



รูปที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 500 หน่วย

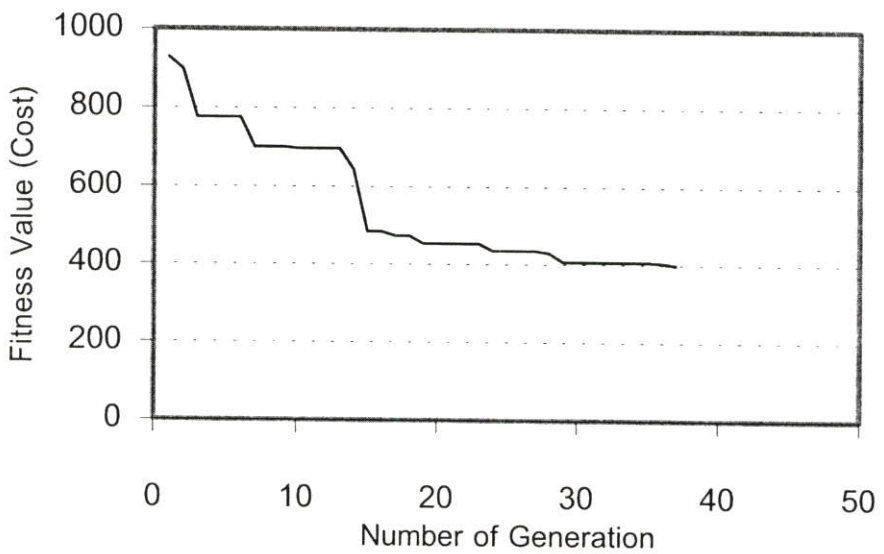
ค่าความเหมาะสม (Fitness Value) = 400 หน่วย

ทำการประมวลผลจนกระทั่งได้โครโมโซมที่มีความเหมาะสมที่สุดตามที่ต้องการ คือ 175564 มีค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายเท่ากับ 0.0083 ms และมีค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายมีค่าเท่ากับ 399 หน่วย โดยมีรูปแบบการเชื่อมต่อของโครงข่ายเป็นดังรูปที่ 4.7



รูปที่ 4.7 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมสำหรับ Fitness Value = 400 หน่วย

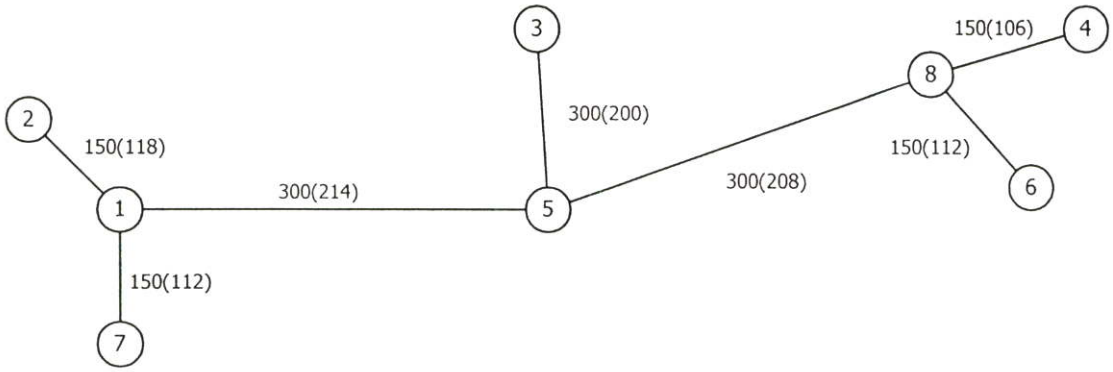
สำหรับการออกแบบในกรณีนี้ พบว่าสามารถหาโครโมโซมที่มีความเหมาะสมที่สุดตามที่ต้องการสามารถหาได้ในประชากร (Generation) รุ่นที่ 37 ซึ่งแนวโน้มของการออกแบบโดยใช้จีเนติกอัลกอริทึมเป็นดังรูปที่ 4.8



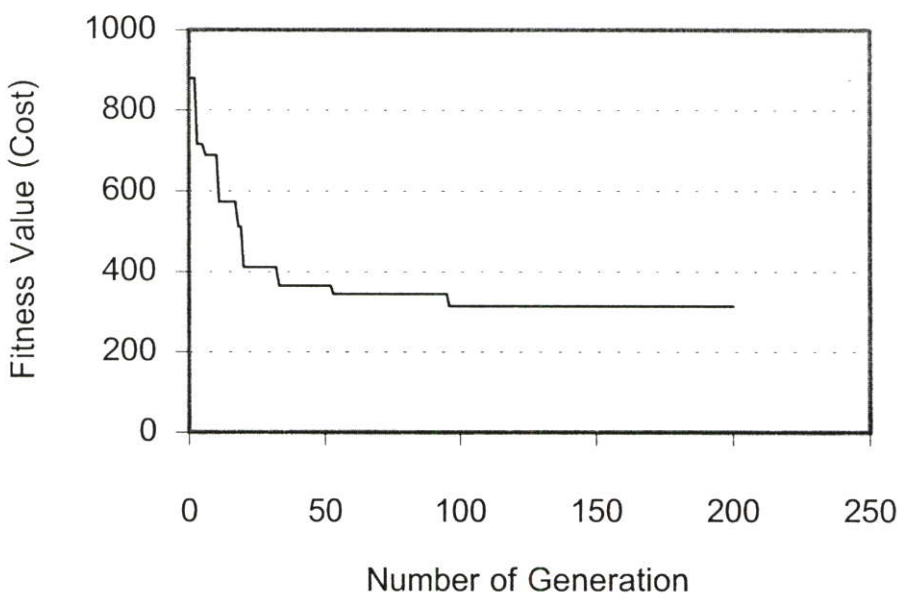
รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 400 หน่วย

ค่าความเหมาะสม (Fitness Value) = 300 หน่วย

สำหรับการออกแบบในกรณีนี้ พบว่าไม่สามารถหาโครโมโซมที่มีความเหมาะสมตามที่ต้องการได้จนกระทั่งถึงค่าจำนวนรุ่นที่มากที่สุด(Maxgen) ซึ่งกำหนดไว้คือรุ่นที่ 200 โปรแกรมจะหยุดทำงานและนำโครโมโซมที่มีความเหมาะสมที่สุดเท่าที่หาได้มาเป็นคำตอบ ซึ่งโครโมโซมที่ได้คือ 158815 มีค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายเท่ากับ 0.009 ms และมีค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายมีค่าเท่ากับ 314 หน่วย โดยมีรูปแบบการเชื่อมต่อของโครงข่ายเป็นดังรูปที่ 4.9 และแนวโน้มของการออกแบบโดยใช้จีเนติกอัลกอริทึมเป็นดังรูปที่ 4.10



รูปที่ 4.9 โครงข่ายที่ออกแบบโดยใช้จีเนติกอัลกอริทึมสำหรับ Fitness Value = 300 หน่วย



รูปที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า Fitness Value กับจำนวน Generation เมื่อ Fitness Value = 300 หน่วย

สำหรับการทดลองในเงื่อนไขที่ค่าความเหมาะสม(Fitness Value) มีค่าต่างๆกันพบว่าแนวโน้มของการออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึมจะได้ค่าความเหมาะสมดีขึ้นเรื่อยๆจนกระทั่งได้โครโมโซมที่มีค่าความเหมาะสมตามที่ต้องการหรือถึงค่าจำนวนรุ่นสูงสุดตามที่ได้กำหนดไว้ โปรแกรมก็จะหยุดการทำงานและนำโครโมโซมที่มีความเหมาะสมที่สุดตามที่ได้มาเป็นคำตอบ จากนั้นนำโครโมโซมมาทำการถอดรหัส Prüfer Number ก็จะได้รูปแบบของโครงข่ายที่มีความเหมาะสมที่สุดออกมา

จากการทดลองจะเห็นได้ว่าสามารถที่จะออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีที่ใช้รหัส Prüfer Number แทนรูปแบบการเชื่อมต่อของโครงข่ายที่มีจำนวนโหนดตามที่ต้องการได้ อีกทั้งรูปแบบของโครงข่ายก็มีความเหมาะสมตามที่ได้กำหนดไว้ ซึ่งในการวิจัยนี้ได้ใช้ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายและค่าใช้จ่ายในระบบสายสัญญาณของโครงข่ายเป็นปัจจัยในการพิจารณาค่าความเหมาะสม โดยการออกแบบนั้นได้ใช้กระบวนการทางจีเนติกอัลกอริทึมซึ่งเป็นกระบวนการในการหารูปแบบคำตอบที่ดีที่สุดเพื่อช่วยให้การออกแบบนั้นมีประสิทธิภาพและมีความรวดเร็วในการออกแบบ

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในวิทยานิพนธ์ฉบับนี้เป็นการนำเสนอการออกแบบโครงข่าย ATM โดยใช้กระบวนการจีเนติกอัลกอริทึมมาประยุกต์ใช้ ซึ่งการออกแบบรูปแบบการเชื่อมต่อของโครงข่ายนั้นมีความสำคัญเป็นอย่างยิ่งไม่ว่าจะเป็นโครงข่ายแบบใดก็ตาม สำหรับโครงข่าย ATM นั้นเป็นโครงข่ายที่มีประสิทธิภาพและความเร็วในการส่งข้อมูลสูง โดยทำการส่งข้อมูลเป็นแพ็กเกจที่มีขนาดคงที่ที่เรียกว่าเซลล์ ดังนั้นถ้าสามารถออกแบบโครงข่ายให้มีความเหมาะสมยิ่งเป็นการเพิ่มประสิทธิภาพในการส่งข้อมูลของโครงข่าย ATM ให้มีประสิทธิภาพสูงยิ่งขึ้น โดยในวิทยานิพนธ์นี้ได้พิจารณาค่าใช้จ่ายในระบบสายสัญญาณและค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายเป็นปัจจัยหลักในการพิจารณาว่ารูปแบบของโครงข่ายว่ามีความเหมาะสมหรือไม่ โดยการออกแบบจะต้องทำให้เสียค่าใช้จ่ายของระบบสายสัญญาณน้อยที่สุดหรือตามที่ต้องการ โดยที่ค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายมีค่าไม่เกินค่าดีเลย์เฉลี่ยสูงสุดของเซลล์ในโครงที่ยอมรับได้ซึ่งจะทำให้ประสิทธิภาพในการส่งข้อมูลของโครงข่ายนั้นก็มีประสิทธิภาพสูงเป็นไปตามที่ต้องการ อีกทั้งยังต้องการความรวดเร็วในการออกแบบ ดังนั้นจึงนำคอมพิวเตอร์มาประยุกต์ใช้ในการออกแบบโครงข่าย ซึ่งจะทำให้มีความสะดวกและรวดเร็วในการออกแบบ

กระบวนการจีเนติกอัลกอริทึมเป็นทฤษฎีหนึ่งที่นำมาประยุกต์ใช้งานในการวิเคราะห์ข้อมูลเพื่อเพิ่มประสิทธิภาพให้กับคอมพิวเตอร์หรือโปรแกรม ซึ่งจีเนติกอัลกอริทึมเป็นทฤษฎีที่เป็นการนำกระบวนการวิวัฒนาการของสิ่งมีชีวิตมาประยุกต์ใช้ในการหารูปแบบคำตอบที่ใกล้เคียงหรือที่ดีที่สุดของปัญหาจากจำนวนคำตอบที่เป็นไปได้ทั้งหมดของการแก้ปัญหานั้นๆ โดยแต่ละคำตอบจะเรียกว่าโครโมโซม ซึ่งจีเนติกอัลกอริทึมเป็นวิธีการหาคำตอบโดยพิจารณาและดำเนินการจากกลุ่มคำตอบของปัญหาที่ถูกสร้างขึ้นโดยการเข้ารหัสหรือแปลงค่าตัวแปรพารามิเตอร์ต่างๆของปัญหาให้อยู่ในรูปโครงสร้างของโครโมโซมตามที่กำหนดและผ่านกระบวนการต่างๆไม่ว่าจะเป็นการวัดค่าความเหมาะสม การคัดเลือก การครอสโอเวอร์ การมิวเตชัน และการรีโพรดักชัน ซึ่งกระบวนการต่างๆที่กล่าวมาจะทำให้วิวัฒนาการของคำตอบมีค่าดีขึ้นเรื่อยๆจนกระทั่งได้คำตอบที่ดีที่สุด ดังนั้นในการใช้จีเนติกมาประยุกต์ใช้ในการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีเพื่อที่จะหารูปแบบโครงข่ายที่เหมาะสมหรือดีที่สุดโดยวิเคราะห์ค่าความเหมาะสมตามที่ได้กำหนดไว้ ซึ่งจะทำให้สามารถหารูปแบบของโครงข่ายที่เหมาะสมที่สุดตามที่ต้องการได้

สำหรับในการทดลองการออกแบบโครงข่าย ATM ที่มีโครงสร้างแบบทรีที่ใช้รหัส Prüfer Number เป็นรหัสของโครงข่ายแบบทรีโดยประยุกต์ใช้จีเนติกอัลกอริทึมช่วยในการออกแบบนั้นพบว่าสามารถที่จะออกแบบโครงข่ายได้ตามที่ต้องการซึ่งก็คือจะมีค่าใช้จ่ายของระบบสายสัญญาณ

ในโครงข่ายที่ต่ำตามที่ต้องการและค่าดีเลย์เฉลี่ยของเซลล์ในโครงข่ายมีค่าไม่เกินค่ามีค่าดีเลย์เฉลี่ยสูงสุดของเซลล์ในโครงข่ายที่ยอมรับได้ตามที่ได้กำหนดไว้ และมีความรวดเร็วในการออกแบบ แต่บางครั้งในการออกแบบโครงข่าย ก็อาจไม่สามารถหารูปแบบของโครงข่ายได้ตามที่ต้องการซึ่งก็ต้องให้คอมพิวเตอร์ทำการออกแบบใหม่อีกครั้ง แต่รูปแบบของโครงข่ายที่ได้นั้นก็มีความเหมาะสมและยอมรับได้ อย่างไรก็ตามสามารถนำโครงสร้างแบบทรีนี้ไปใช้กับโครงข่ายอื่นๆได้และยังสามารถใช้ปัจจัยอื่นๆได้อีกนอกจากที่พิจารณาในวิทยานิพนธ์นี้มาใช้เป็นปัจจัยเพิ่มเติมในการพิจารณาการออกแบบโครงข่าย หรืออาจจะนำกระบวนการจีเนติกอัลกอริทึมไปประยุกต์ใช้ในงานอื่นๆ โดยการที่จะประยุกต์ใช้จีเนติกอัลกอริทึมกับแต่ละปัญหานั้น จะต้องปรับรูปแบบของปัญหาให้อยู่ในรูปแบบของโครโมโซมที่เข้ากับกระบวนการต่างๆของจีเนติกอัลกอริทึม โดยลักษณะของโครโมโซมจะแตกต่างกันไปตามแต่ละปัญหา

เอกสารอ้างอิง

[1] L. Kleinrock **Queueing Systems** Vol.2 Wiley-Interscience 1976.

[2] J.L.Hammond , P.J.P. O'Reilly **Performance Analysis of Local Computer Networks** Addison–Wesley Publishing Company,Inc. 1986.

[3] D.E. Goldberg **Genetic Algorithms : in Search Optimization and Machine Learning** Addison- Wesley Publishing Company,Inc. 1989.

[4] J M Pitts , J A Schormans **Introduction to ATM Design and Performance** John Wiley & Sons 1996.

[5] กาญจนี วงศ์วิภาพร. " การจัดตารางสอนของโรงเรียนแบบอัตโนมัติโดยจีเนติก อัลกอริทึม. " วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541.

[6] M.Gen , K. Ida, and J.R. Kim "A Spanning Tree-based Genetic Algorithm for Bicriteria Topological Network Design " Proceedings of IEEE International Conference on Evolutionary Computation, 1998. pp.15-20.

[7] K.T. Ko , K.S. Tang , C.Y. Chan and K.F. Man " Packet switched communication network designs using GA " Genetic Algorithm in Engineering Systems : Innovations and Applications ,1997. GALESIA 97. Second International Conference On (Conf. Publ. No.446) 1997. pp. 398-403.

[8] M.Gen,G. Zhou , and M. Takayama "A Comparative Study of Tree Encodings on Spanning Tree Problems" Proceedings of IEEE International Conference on Evolutionary Computation 1998. pp. 33-38.

ภาคผนวก

ภาคผนวก ก.

ซอร์ดโค๊ดโปรแกรม

```
'----------Form1----------
Private Sub Command1_Click()
Node = Val(Text1.Text)
Ca(1) = Val(Text2(0).Text)
Ca(2) = Val(Text2(1).Text)
Ca(3) = Val(Text2(2).Text)
Ca(4) = Val(Text2(3).Text)
Ca(5) = Val(Text2(4).Text)
Co(1) = Val(Text2(5).Text)
Co(2) = Val(Text2(6).Text)
Co(3) = Val(Text2(7).Text)
Co(4) = Val(Text2(8).Text)
Co(5) = Val(Text2(9).Text)
Pc = Val(Text3.Text)
Pm = Val(Text4.Text)
GenMax = Val(Text5.Text)
Popsize = Val(Text6.Text)
Delaymax = Val(Text7.Text)/1000
Fitness = Val(Text8.Text)
Unload Form1
Form2.Show
End Sub

Private Sub Command2_Click()
End
End Sub
```

```
'----------Form2----------
Dim  i As Integer, j As Integer

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
i = 1
j = 1
Label3.Caption = i
Label4.Caption = j
Label8.Caption = i
Label9.Caption = j
End Sub

Private Sub Command1_Click()
If j <= Node Then
   a(i, j) = Val(Text1.Text)
   Dis(i, j) = Val(Text2.Text)
   j = j + 1
Else
   i = i + 1
   j = 1
   a(i, j) = Val(Text1.Text)
   Dis(i, j) = Val(Text2.Text)
   j = j + 1
End If
If i = Node Then
 If j - 1 = Node Then
 Form3.Show
```

```
GoTo e1
 End If
 End If
 Form2.Refresh
 Label3.Caption = i
 Label4.Caption = j
 Label8.Caption = i
 Label9.Caption = j
 Text1.Text = ""
 Text2.Text = ""
If j > Node Then
 Label3.Caption = i + 1
 Label4.Caption = "1"
 Label8.Caption = i + 1
 Label9.Caption = "1"
End If
e1:
End Sub
```

```
'----------Form3----------
Private Sub Command1_Click()
ChrLength = Node - 2
Link = Node - 1
Length = 424
u = 1 / Length
Generation = 0
List1.Clear
Nc = 5
'--------Genetic Algorithm---------
Chromosome
re:
ClearVar
Decode
FitnessValue
 If Generation < GenMax - 1 Then
  For pop = 1 To Popsize
   If (Fit(pop) < Fitness) And (D(pop) < Delaymax) Then
    GoTo e1:
   End If
  Next
    Generation = Generation + 1
    Selection
    Crossover
    Mutation
    GoTo re:
 End If
e1:
'----------Display----------
 Text5.Text = cost(pp)
 Text3.Text = Generation + 1
 Text1.Text = ""
```

```
For i = 1 To ChrLength
Text1.Text = Text1.Text + Str$(p(pp, i))
Next
Text4.Text = D(pp)
For i = 1 To Link
List1.AddItem "----------Link" + Str$(i) + "----------"
List1.AddItem "Node" + Str$(r(pp, i, 1)) + " -" + Str$(r(pp, i, 2))
List1.AddItem "Traffic =" + Str$(f(pp, i)) + " Mbps"
List1.AddItem "Capacity =" + Str$(c(pp, i)) + " Mbps"
List1.AddItem "Distance =" + Str$(Dis(r(pp, i, 1), r(pp, i, 2))) + " Km"
Next
List1.AddItem "-----------------------------"
For i = 1 To GenMax
Debug.Print "MinCost of Gen" + Str$(i) + Str$(kop(i))
List1.AddItem "Gen." + Str$(i) + " ->" + "Min.Cost =" + Str$(kop(i))
Next
End Sub

Private Sub Command2_Click()
End
End Sub
```

```
'----------Module1----------

Public  Node, Length, GenMax, Popsize, Fitness, FitValue
Public  ChrLength, Link, pop, pos, pos1, pos2, pp, Generation
Public  Pc, Pm, Sum, Delaymax, S, a(50, 50), f(50, 50), c(50, 50), u, Costmin, Capacity,
        Dis(50, 50), kop(200)
Public  D(50), Fit(50), Sum0, Sum1, ra, aa, DD(50), Dmin, cost(50), cost1(50), Distance(50),
        Co(10), Ca(10), Cunit(50, 50)
Public  t(50), w(50), L(50), Nc, ff(50), Ep(50)
Public  p(50, 50), q(50, 50), r(50, 50, 50), best(50, 50), par(50, 50)
Public  E As Integer, EE As Integer


Sub Chromosome()
Randomize
For pop = 1 To Popsize
 For i = 1 To ChrLength
 p(pop, i) = Int((Node * Rnd) + 1)
 Next
Next
End Sub


Sub Selection()
For pop = 1 To Popsize
   If D(pop) < Delaymax Then
      totalcost = totalcost + 1 / cost(pop)
   End If
Next
For pop = 1 To Popsize
   If D(pop) < Delaymax Then
      E = ((1 / cost(pop)) / totalcost) * Popsize
      For i = EE To EE + (E - 1)
        Ep(i + 1) = pop
      Next
```

```
      EE = EE + E
    End If
  Next
  pops = Popsize
  For pop = 1 To Popsize
    ran_cross = Int((pops * Rnd) + 1)
    For i = 1 To ChrLength
      par(pop, i) = p(Ep(ran_cross), i)
    Next
    pops = pops - 1
    For x = ran_cross To pops
      Ep(x) = Ep(x + 1)
    Next
  Next
  End Sub

  Sub Crossover()
   For pop = 1 To Popsize Step 2
    ra = Rnd
    If ra < Pc Then
    pos1 = Int((ChrLength - 1) * Rnd + 1)
    pos2 = Int((ChrLength - pos1) * Rnd + (pos1 + 1))
     For i = pos1 + 1 To pos2
     z = par(pop, i)
     par(pop, i) = par(pop + 1, i)
     par(pop + 1, i) = z
     Next
    End If
   Next
   For pop = 1 To Popsize
     For i = 1 To ChrLength
       p(pop, i) = par(pop, i)
```

```
  Next
Next
End Sub

Sub Mutation()
For pop = 1 To Popsize
  ra = Rnd
  If ra < Pm Then
  pos = Int(ChrLength * Rnd + 1)
  p(pop, pos) = Int((Node * Rnd) + 1)
  End If
Next
For i = 1 To ChrLength
p(pp, i) = best(pp, i)
Next
End Sub

Sub Decode()
For pop = 1 To Popsize
  x = 1
  Y = 1
  v = 0
  z = 0
g11:
For j = Y To ChrLength
  If p(pop, j) <> x Then
    q(pop, Y) = x
  Else
g44:
    x = x + 1
    GoTo g11:
  End If
```

```
Next
For i = 1 To Node
 If i <> Y Then
  If q(pop, i) = x Then
  GoTo g44:
  End If
 End If
Next
If Y > ChrLength Then
   q(pop, Y) = x
End If
If Y = Node Then
 GoTo g22:
Else
 Y = Y + 1
 z = 1
g33:
 For k = 1 To Node
   If q(pop, k) = z Then
     z = z + 1
     GoTo g33:
   End If
  Next
 x = z
 GoTo g11:
End If
g22:
'--------- Display r ------------
For i = 1 To Popsize
 For z = 1 To Link
 r(i, z, 1) = q(i, z)
 r(i, z, 2) = p(i, z)
```

```
If z > ChrLength Then
r(i, z, 1) = q(i, z)
r(i, z, 2) = q(i, z + 1)
End If
Next
Next
For i = 1 To Node
For j = 1 To Node
If i = j Then
GoTo g1:
End If
For x = 1 To Link
For h = x To Link
If (i = r(pop, h, 1) And j = r(pop, h, 2)) Or (i = r(pop, h, 2) And j = r(pop, h, 1)) Then
f(pop, h) = f(pop, h) + a(i, j)
GoTo g1:
End If
Next
If i = r(pop, x, 1) Or i = r(pop, x, 2) Then
If i = r(pop, x, 1) Then
w(S) = r(pop, x, 2)
Else
w(S) = r(pop, x, 1)
End If
If j = w(S) Then
f(pop, x) = f(pop, x) + a(i, j)
GoTo g1:
Else
t(v) = x
L(S) = x
GoTo g3:
End If
```

```
  Else
   GoTo g2:
  End If
g3:
For k = 1 To Link
 For E = 0 To v
 If k = t(E) Then
   GoTo g4:
  End If
 Next
 If w(S) = r(pop, k, 1) Or w(S) = r(pop, k, 2) Then
    If w(S) = r(pop, k, 1) Then
     S = S + 1
     w(S) = r(pop, k, 2)
    Else
     S = S + 1
     w(S) = r(pop, k, 1)
    End If
    If j = w(S) Then
     For z = 0 To S - 1
      f(pop, L(z)) = f(pop, L(z)) + a(i, j)
     Next
     f(pop, k) = f(pop, k) + a(i, j)
     S = 0
     For E = 0 To v
     t(E) = 0
     L(E) = 0
     Next
     v = 0
    GoTo g1:
    Else
     L(S) = k
```

```
    v = v + 1
    t(v) = k
    GoTo g3:
   End If
Else
   GoTo g4:
End If
g4:
 If k = Link Then
  If S = 0 Then
   t(v) = 0
   L(S) = 0
  Else
   S = S - 1
   GoTo g3:
  End If
 GoTo g2:
 End If
Next
g2:
 Next
g1:
 Next
Next
Next
End Sub

Sub FitnessValue()
'---------------Cost----------------------
For pop = 1 To Popsize
 For i = 1 To Link
  For j = 1 To Nc
```

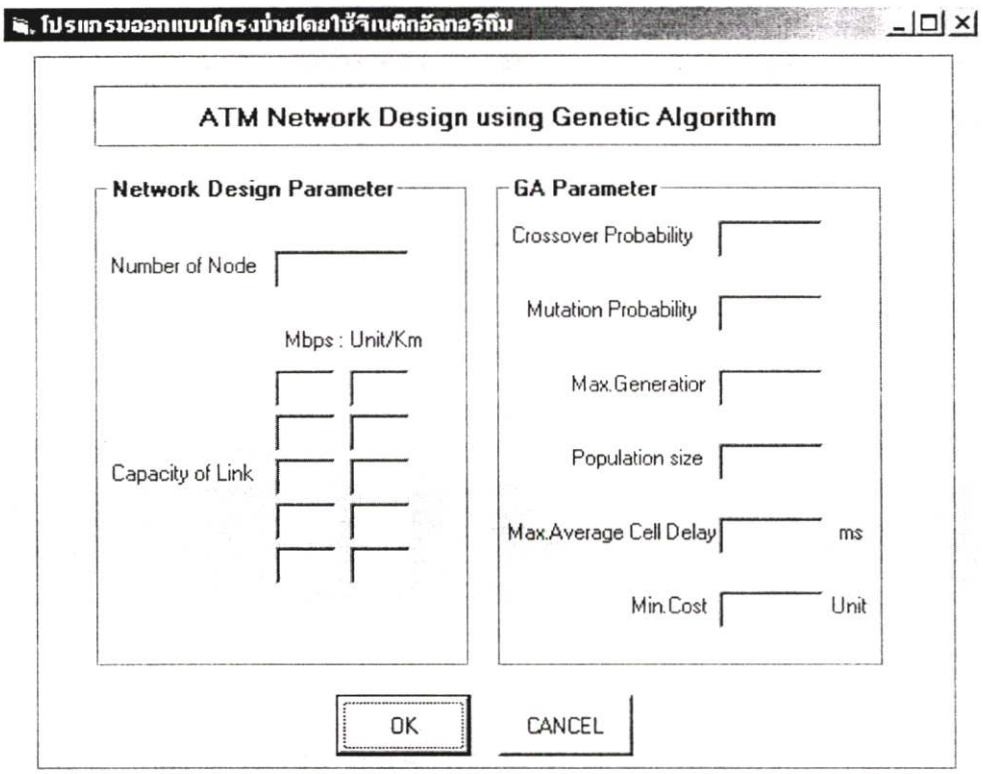
```
    If Ca(j) > f(pop, i) Then
      c(pop, i) = Ca(j)
      Cunit(pop, i) = Co(j)
      GoTo gg:
    End If
  Next
gg:
 Next
Next
For pop = 1 To Popsize
  For i = 1 To Link
    cost(pop) = cost(pop) + Cunit(pop, i) * Dis(r(pop, i, 1), r(pop, i, 2))
  Next
  Fit(pop) = cost(pop)
Next
'----Total average packet in network----
 For i = 1 To Node
  For j = 1 To Node
  aa = aa + a(i, j)
  Next
 Next
'---------Total average delay----------
 For pop = 1 To Popsize
 Sum1 = 0
 Sum0 = 0
 For i = 1 To Link
 Sum1 = ((2 * c(pop, i) * f(pop, i)) - (f(pop, i) * f(pop, i))) / ((2 * c(pop, i)) * (c(pop, i) - f(pop, i)))
 Sum0 = Sum0 + Sum1
 Next
 D(pop) = (Length / (aa * 1000000)) * Sum0
 Next
 '-------------------------------------
```

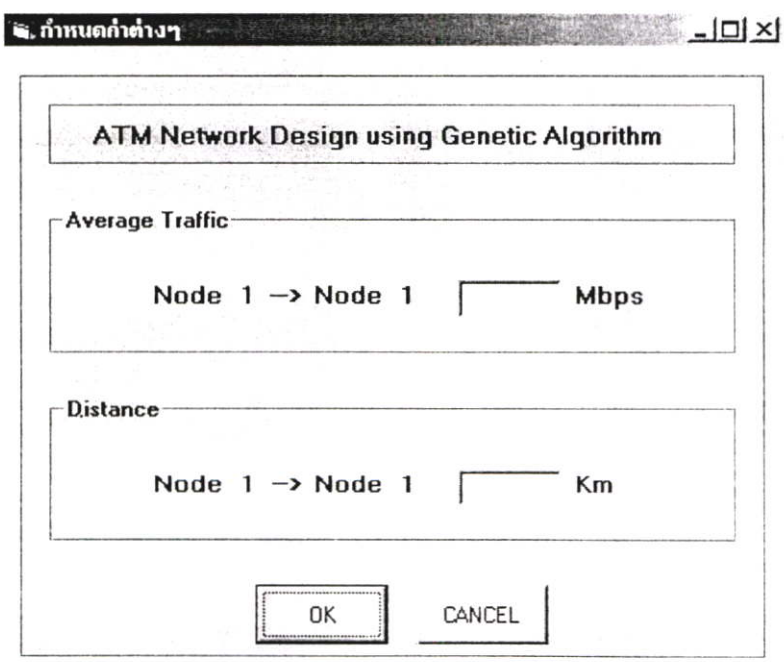
```
For pop = 1 To Popsize
 If D(pop) < Delaymax Then
  ff(yy) = pop
  yy = yy + 1
 End If
Next
'----Find Best Chromosome Min.Cost----
For pop = 1 To Popsize
   cost1(pop) = cost(pop)
Next
For i = 0 To yy - 1
If cost1(ff(i)) < cost1(ff(i + 1)) Then
  cost1(ff(i + 1)) = cost1(ff(i))
End If
Next
For i = 0 To yy - 1
 If cost1(ff(yy - 1)) - cost1(ff(i)) = 0 Then
   GoTo g1:
 End If
Next
g1:
If yy = 0 Then
MsgBox "New Run Again", 48, "warnning"
End
End If
Costmin = cost1(ff(yy - 1))
kop(Generation + 1) = Costmin
gen = Generation + 1
pp = ff(i)
For i = 1 To ChrLength
best(pp, i) = p(pp, i)
Next
```

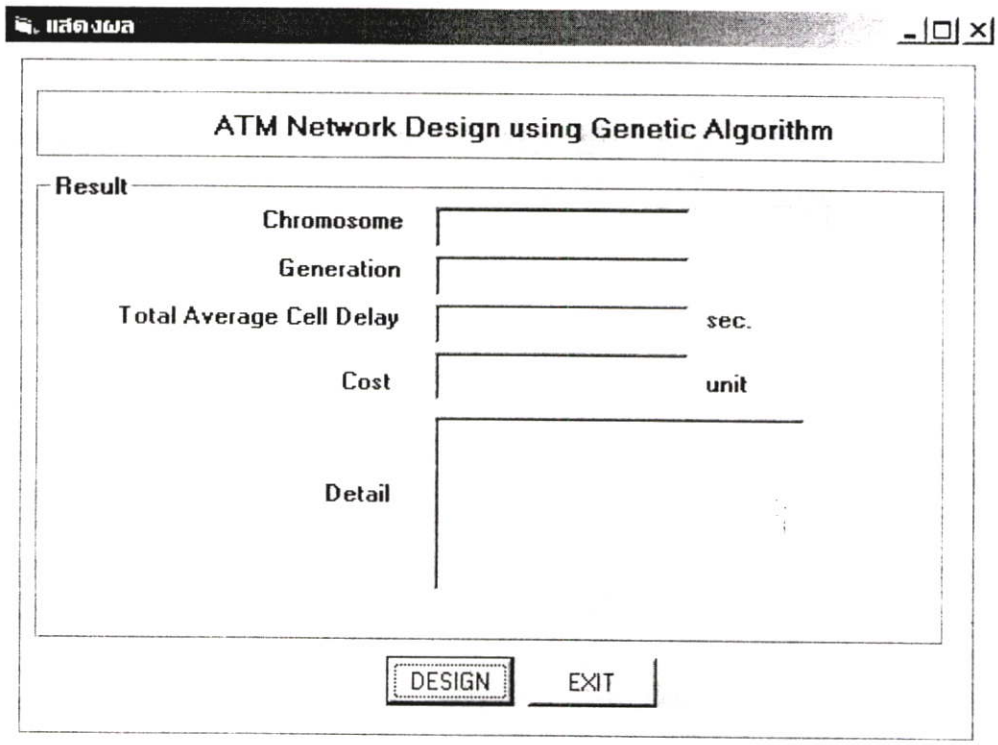
```
yy = 0
'----------Display----------
Debug.Print "--------------------------------------"
Debug.Print "generation=" + Str$(Generation + 1)
For pop = 1 To Popsize
Next
Debug.Print "Minimum Cost(Delay < MinDelay)=" + Str$(Costmin)
End Sub

Sub ClearVar()
For pop = 1 To Popsize
DD(pop) = 0
cost(pop) = 0
cost1(pop) = 0
ff(pop - 1) = 0
For i = 1 To Node
q(pop, i) = 0
Next
For i = 1 To ChrLength + 1
r(pop, i, 1) = 0
r(pop, i, 2) = 0
Next
For x = 1 To Link
f(pop, x) = 0
Next
Next
aa = 0
Sum = 0
Sum1 = 0
yy = 0
EE = 0
End Sub
```

โปรแกรมสำหรับออกแบบโครงข่าย ATM






แสดงผล
ATM Network Design using Genetic Algorithm
Result
Chromosome
Generation
Total Average Cell Delay
sec.
Cost
unit
Detail
DESIGN
EXIT

ภาคผนวก ข.
ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] ธนาวุฒิ อรชุน, สุวิพล สิทธิชีวภาค และ เกรียงไกร วงศ์โรจน์ภรณ์. "การออกแบบโครงข่าย ATM โดยใช้จีเนติกอัลกอริทึม." วิศวสารลาดกระบัง , ปีที่18 , ฉบับที่ 3 , หน้า 61-65 , กันยายน , 2544.

ประวัติผู้เขียน

นายธนาวุฒิ อรชุน เกิดเมื่อวันที่ 10 สิงหาคม 2519 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนโยธินบูรณะ และ วิศกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยศรีนครินทรวิโรฒ ปี พ.ศ. 2542