การใช้จีเนติกอัลกอริทึม ช่วยในการตัดสินใจ

Decision Support Tools using Genetic Algorithm

H001678

โดย

นายธรรมจักร จักรกริชกูล

รหัส 41067113

อาจารย์ที่ปรึกษา

ผศ.ดร. เอื้อน ปิ่นเงิน

วัน เดือน ปี 25 ธ.ค. 2549
เลขทะเบียน 01678
เลขเรียกหนังสือ วพ. ส328ก 2543
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การใช้จีเนติกอัลกอริทึม ช่วยในการตัดสินใจ
นักศึกษา	นายธรรมจักร จักรกริชกูล
อาจารย์ที่ปรึกษา	ผศ. ดร. เอื้อน ปิ่นเงิน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

จีเนติกอัลกอริทึม เป็นรูปแบบการนำเสนอโดยอาศัยแนวความคิดทางด้าน ชีววิทยา กล่าวคือ กระบวนการการคัดเลือกทางธรรมชาติ และกระบวนการทางพันธุศาสาตร์ สำหรับประยุกต์ใช้สร้างวิวัฒนาการหาคำตอบในการแก้ปัญหาต่าง ๆ ดังนั้นโครงการพัฒนาระบบงานนี้ จึงนำเสนอการสร้าง จีเนติกอัลกอริทึม ในรูปแบบของ คอมโพเนนต์ เพื่อให้ผู้พัฒนาโปรแกรมสามารถเรียกใช้งาน ได้โดยการใส่ค่า พารามิเตอร์ ต่าง ๆ โดยเลือกใช้โปรแกรมออบเจ็คปาสคาล หรือ เดลไฟล์ 4 เพื่อเป็นตัวพัฒนาโปรแกรมดังกล่าว ซึ่งจะช่วยให้การเขียนโปรแกรม ที่เรียกใช้จีเนติกอัลกอริทึม มีความกระชับ, ขนาดเล็ก ง่ายต่อการนำไปใช้งานใน แอพพลิเคชัน อื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Decision Support Tools using Genetic Algorithm
Student	Mr. Thamajak Jakkrichkul
Advisor	Asst. Prof. Dr. Ouen Pin-ngern
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

Genetic Algorithm is a computer methodology that borrows the working concept in biology field. Natural selection and genetic processes are the major methods used for developing solutions to solve various kind of problems. This system development project is focusing on using the genetic algorithm to solve problem. Such components can be used by specifying parameters through Object Pascal or Delphi 4 development environment. As a result, programs that need the genetic algorithm ability can conveniently utilize this component to produce a compact and easy to use application.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญตาราง	V
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการพัฒนาระบบงาน	2
1.3 แนวคิดที่ใช้ในโครงการพัฒนาระบบงาน	2
1.4 ขอบเขตโครงการพัฒนาระบบงาน	2
บทที่ 2 ความรู้พื้นฐานเกี่ยวกับจีเนติกอัลกอริทึม	4
2.1 พันธุศาสตร์ทางชีววิทยา	4
2.2 จีเนติกอัลกอริทึม	6
2.3 ฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม	8
2.4 รูปแบบโครโมโซม	8
2.5 การทำงานของจีเนติกอัลกอริทึม	8
บทที่ 3 จีเนติกอัลกอริทึมกับปัญหาประเภทต่าง ๆ	21
3.1 ปัญหาการทายคำ	21
3.2 ปัญหาทาว์นเวอร์ออฟฮานอย	23
3.3 ปัญหาการหาเส้นทางที่ระยะทางสั้นที่สุด (TSP)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 4 Delphi และ คอมโพเนนต์จีเนติกอัลกอริทึม	29
4.1 การพัฒนาโปรแกรมด้วย Delphi	29
4.2 ส่วนประกอบของ Delphi	32
4.3 ขั้นตอนการติดตั้ง คอมโพเนนต์จีเนติก	36
4.4 การเรียกใช้งาน คอมโพเนนต์จีเนติกอัลกอริทึม	39
4.5 โปรแกรม Demo ทดสอบการทำงานของคอมโพเนนต์จีเนติกอัลกอริทึม	39
4.6 สรุปผลการทำงาน	42
เอกสารอ้างอิง	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงคำศัพท์ที่ใช้ในทางพันธุศาสตร์กับจีเนติกอัลกอริทึม	10
3.1 การกำหนดค่าบิทของโครโมโซมปัญหาทายคำ	21
3.2 การกำหนดค่าบิทของโครโมโซม THP	25
3.3 การกำหนดค่าบิทของโครโมโซม TSP	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงลักษณะทางพันธุศาสตร์ของโครโมโซมของมนุษย์	4
2.2 แสดงก่อนและหลังการครอสโอเวอร์ของโครโมโซม	5
2.3 แสดงการเกิดมิวเตชันของโครโมโซม	6
2.4 แสดงหลักการเบื้องต้นของ จีเนติกอัลกอริทึม	7
2.5 การทำงานของจีเนติก อัลกอริทึม	9
2.6 แสดงวิธีรีโพรดักชันแบบรักษาค่าที่ดี 1 โครโมโซม	18
2.7 แสดงวิธีรีโพรดักชันแบบรักษาค่าที่ดี 2 โครโมโซม	18
2.8 การครอสโอเวอร์ แบบ 2 จุด	19
2.9 การอินเวอร์ชัน	20
3.1 การครอสโอเวอร์แบบหลายจุด	23
3.2 ภาพปัญหาทาว์นเวอร์ออฟฮานอย	24
3.3 รีเดอร์ซีฟอัลกอริทึมของทาว์นเวอร์ออฟฮานอย	24
3.4 การเคลื่อนย้ายวงแหวนของโครโมโซม THP(1323)	25
3.5 ภาพปัญหาการหาเส้นทางที่สั้นที่สุด	27
4.1 ไอคอนโปรแกรม Delphi 4	32
4.2 Windows ต่าง ๆ ของ โปรแกรม Delphi	33
4.3 วินโดว์หลัก	33
4.4 เมนูบาร์	33
4.5 สปีดบาร์	34
4.6 คอมโพเนนต์พาเล็ต	34
4.7 หน้าต่าง Object Inspector	35
4.8 หน้าต่าง Code Editor	36
4.9 ไอคอน แพกเก็ตสำหรับติดตั้งคอมโพเนนต์	36
4.10 ไดอะล็อกบ็อกซ์สำหรับติดตั้งคอมโพเนนต์	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11 การเซ็ทค่าไดเร็คทอรีที่ทำการจัดเก็บไฟล์	38
4.12 คอมโพเนนต์ Genetic ปรากฏใน คอมโพเนนต์พาเล็ต	38
4.13 Properties ของ คอมโพเนนต์จีเนติกอัลกอริทึม	39
4.14 ไอคอนโปรแกรมจีเนติกอัลกอริทึม	39
4.15 โปรแกรมจีเนติกอัลกอริทึมในรูปแบบของ Text Mode	40
4.16 โปรแกรมจีเนติกอัลกอริทึมในรูปแบบของ Graphic Mode	40
4.17 โปรแกรมจีเนติกอัลกอริทึมบนบราวเซอร์ สร้างโดยใช้ ActiveX Form ใน Delphi	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมาของปัญหา

การพัฒนาซอฟต์แวร์ได้เปลี่ยนแปลงตามสภาพการพัฒนา เริ่มจากการคิดค้นหาภาษาคอมพิวเตอร์เพื่อช่วยในการสั่งการ การพัฒนาซอฟต์แวร์ที่ใช้งานจึงขึ้นอยู่กับการเขียนโปรแกรมในยุคต้น ภาษาที่ใช้สั่งการเป็นภาษาเชิงลำดับ งานประยุกต์จึงใช้ภาษาเชิงลำดับสั่งงาน เช่น การพัฒนาซอฟต์แวร์ด้วยภาษาฟอร์แทรน เบสิก โคบอล ต่อมาเมื่อพบว่างานที่สั่งให้คอมพิวเตอร์ทำมีความซับซ้อนมากขึ้น การสั่งงานทำให้วิธีการเขียนโปรแกรมขึ้นอยู่กับตัวบุคคล เพราะลำดับความคิดที่ถ่ายออกมาเป็นรูปโปรแกรมนั้นยากที่จะทำความเข้าใจได้ โปรแกรมที่พัฒนาจึงขึ้นกับตัวบุคคล ไม่สามารถให้อีกบุคคลหนึ่งดำเนินการตรวจสอบหรือทำความเข้าใจเกี่ยวกับโปรแกรมได้ง่าย ด้วยความคิดนี้จึงต้องทำซอฟต์แวร์ให้เป็นโครงสร้าง มีการนิยามภาษาคอมพิวเตอร์แบบกระบวนความ (procedure) เน้นการพัฒนาซอฟต์แวร์แบบโครงสร้าง จึงมีการใช้โปรแกรมแบบโครงสร้าง การพัฒนาภาษาคอมพิวเตอร์หรือเครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์จึงเน้นแบบโครงสร้าง ใช้ภาษาปาสคาล ซี หรือภาษาต่าง ๆ ที่พัฒนามาในรูปแบบกระบวนความ เพื่อให้มีการพัฒนาซอฟต์แวร์แบบท็อปดาวน์ มากขึ้น

อย่างไรก็ตาม การพัฒนาซอฟต์แวร์แบบกระบวนความที่เป็นโครงสร้างก็ยังเป็นแนวทางของการพัฒนาซอฟต์แวร์แบบนามธรรม ใช้จินตนาการ ดังนั้นการสร้างจินตนาการในงานที่ซับซ้อนยังเป็นเรื่องยาก ซอฟต์แวร์ตามแนวจินตนาการของบุคคลหนึ่งจึงยากที่จะนำมาใช้กับอีกบุคคลหนึ่ง ซอฟต์แวร์ที่พัฒนาแล้วยังยากที่จะนำมาใช้งานใหม่ ทั่วโลกจึงมีซอฟต์แวร์ที่เขียนกันขึ้นมามากมาย ยากที่จะนำกลับมาใช้ใหม่ได้ จึงเกิดแนวคิด ด้านความสัมพันธ์เชิงวัตถุขึ้น เพราะเป็นสิ่งที่เข้าใจง่าย และสามารถสร้างตามความเข้าใจ เห็นภาพชัดเจน ทั้งนี้เพราะในโลกแห่งความเป็นจริง การแทนโมเดลไม่สามารถที่จะสื่อความหมายทุกสิ่งทุกอย่างได้หมด แต่ขอบเขตของการสร้างโมเดลเพื่ออธิบายความหมายในเชิงสร้างซอฟต์แวร์มีบางสิ่งบางอย่างที่อาจแตกต่างออกไปบ้างโดยเน้นให้สิ่งต่าง ๆ ที่เป็นวัตถุและตัวที่ทำให้เกิดเหตุการณ์ได้ เป็นออบเจ็กต์ในความหมายของสิ่งที่จะใช้ทางซอฟต์แวร์ โดยให้ออบเจ็กต์นั้นมีข้อมูลอยู่ภายใน และยังให้คุณลักษณะของออบเจ็กต์ไปยังออบเจ็กต์อื่น ๆ ได้ ออบเจ็กต์จะอยู่ในกลุ่มที่เรียกว่า คลาส และให้คุณสมบัติสำหรับคลาสสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ่ายทอดกันได้ แล้วด้วยคุณสมบัติที่สามารถถ่ายทอดกันได้ ทำให้การพัฒนาโปรแกรมเชิงวัตถุได้รับความนิยมอย่างรวดเร็ว โดยมีการสร้างเป็น คอมโพเนนต์ (Component) เพื่อเรียกใช้งานในแต่ละแบบแตกต่างกันไปตามลักษณะงานที่พัฒนาขึ้น

1.2 วัตถุประสงค์ของโครงการพัฒนาระบบงาน

ในงานวิจัยนี้มุ่งเน้นไปที่การนำจีเนติกอัลกอริทึม มาทำเป็นเครื่องช่วยในการ ตัดสินใจของผู้พัฒนาโปรแกรม (Decision Support Tools) ซึ่งนอกจากจะทำให้การพัฒนาโปรแกรมเป็นไปอย่างรวดเร็ว และถูกต้องตามหลักของการนำ จีเนติกอัลกอริทึมไปใช้งาน คอมโพเนนต์จีเนติก จึงเป็นอีกทางเลือกหนึ่งที่นำวิธีการวิวัฒนาการทางธรรมชาตินำมาประยุกต์ใช้กับปัญหาต่าง ๆ โดยจะช่วยให้ผู้พัฒนาโปรแกรมสามารถทำงานได้อย่างรวดเร็ว และลดขั้นการทำงานต่าง ๆ ลง

1.3 แนวคิดที่ใช้ในโครงการพัฒนาระบบงาน

จีเนติกอัลกอริทึมเป็นทฤษฎีที่จำลองกระบวนการทางธรรมชาติ คือการคัดเลือกทางธรรมชาติ และอาศัยพื้นฐานความคิดทางพันธุกรรมในการถ่ายทอดลักษณะต่าง ๆ จากพ่อ-แม่ ไปยังลูกหลานที่สามารถนำมาพัฒนาใช้หาคำตอบที่ใกล้เคียง หรือดีที่สุดของปัญหา จีเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบ โดยพิจารณาจากกลุ่มคำตอบของปัญหาที่ถูกสร้างขึ้น และใช้วิธีการเข้ารหัส (Coding) คือการแปลงค่าตัวแปร หรือพารามิเตอร์ต่าง ๆ ของปัญหา ให้อยู่ในรูปโครงสร้างของโครโมโซมตามที่กำหนด เพื่อคัดเลือกโครโมโซมคำตอบที่เหมาะสมสำหรับสร้างวิวัฒนาการคำตอบให้ดีขึ้น ตามกระบวนการทางพันธุศาสตร์ โดยการแลกเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ระหว่างโครโมโซมตามที่กำหนด เพื่อคัดเลือกโครโมโซมคำตอบที่เหมาะสมสำหรับสร้างวิวัฒนาการคำตอบให้ดีขึ้น ตามกระบวนการทางพันธุศาสตร์ โดยการแลกเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ระหว่าโครโมโซมที่ถูกคัดเลือก จะทำให้คำตอบของปัญหาถูกปรับปรุงให้ดีขึ้น ในทำนองเดียวกัน การใช้จีเนติกอัลกอริทึมเพื่อช่วยในการตัดสินใจ ทำได้โดยการ ให้ผู้ใช้งาน ใส่ค่าตัวแปรต่าง ๆ ให้อยู่ในรูปโครงสร้าง โครโมโซมที่กำหนด เพื่อคัดเลือกโครโมโซมคำตอบที่เหมาะสมสำหรับสร้างวิวัฒนาการคำตอบให้ดีขึ้น ตามกระบวนการทางพันธุศาสตร์ โดยการแลกเปลี่ยนค่าพารามิเตอร์ต่างๆ ระหว่างโครโมโซมที่ถูกคัดเลือก จะทำให้คำตอบของปัญหาถูกปรับปรุงให้ดีขึ้น

1.4 ขอบเขตของโครงการพัฒนาระบบงาน

โครงการพัฒนาระบบงานนี้นำเสนอการประยุกต์ใช้จีเนติกอัลกอริทึม มาเพื่อในการช่วยตัดสินใจ โดยผู้ใช้งานต้องทำการใส่ค่าพารามิเตอร์ต่าง ๆ ตามที่โปรแกรมกำหนด เพื่อให้ได้ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมา การพัฒนาโปรแกรมจะทำในรูปแบบของ คอมโพเนนต์ เพื่อใช้งานร่วมกับโปรแกรมเดลไฟล์ 4 โดยมีวัตถุประสงค์เพื่อให้สามารถนำกลับมาใช้ได้ใหม่ และ ช่วยลดขั้นตอน และเวลาในการสร้างแอพพลิเคชันลง

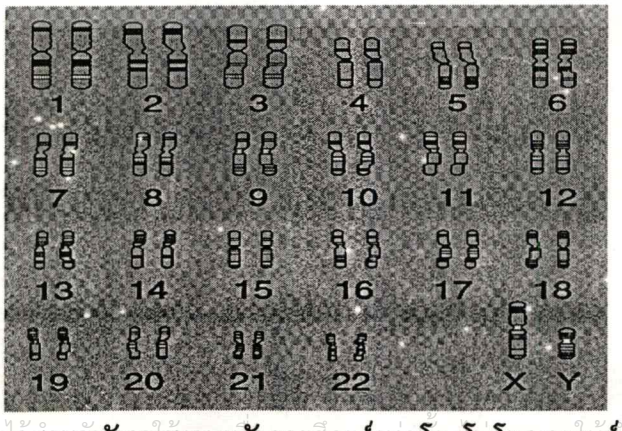
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้พื้นฐานเกี่ยวกับจีเนติกอัลกอริทึม

ปัญหาที่ต้องการคำตอบที่ดีที่สุด (Optimal Solution) ในทางวิทยาศาสตร์ สามารถหาคำตอบได้หลายวิธีซึ่งแตกต่างกันไปตามชนิดของปัญหา โดยมีการนำทฤษฎีที่เกี่ยวข้องกับการเลียนแบบทางธรรมชาติมาช่วยในการวิจัย เช่น การประมวลผลภาษาธรรมชาติ (Natural Language Processing) ระบบผู้เชี่ยวชาญ (Expert System) ฟัซซีลอจิก (Fuzzy Logic) นิวรอลเน็ตเวิร์ค (Neural Network) เป็นต้น จีเนติกอัลกอริทึม (Genetic Algorithm) เป็นอีกทฤษฎีที่จำลองการทำงานทางชีววิทยา ในการให้กำเนิดประชากรรุ่นใหม่ ซึ่งอาศัยพื้นฐานของการวิวัฒนาการทางพันธุกรรมในการถ่ายทอดลักษณะต่าง ๆ ไปยังรุ่นลูกหลาน โดยปฏิบัติตามหลักการทางพันธุศาสตร์ นำมาประยุกต์ใช้ในการแก้ปัญหา เพื่อหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุด

2.1 พันธุศาสตร์ทางชีววิทยา

ยีนส์ (Genes) เป็นหน่วยเก็บลักษณะทางกรรมพันธุ์ ซึ่งค้นพบโดยนักวิทยาศาสตร์ที่ชื่อ เมนเดล (Mendel) และเป็นตัวกำหนดรูปร่างลักษณะภายนอกของสิ่งมีชีวิต ซึ่งยีนส์จะเรียงตัวอยู่บนโครโมโซม (Chromosome) อีกที ในเซลของสิ่งมีชีวิตโครโมโซมจะจับกันอยู่เป็นคู่ ๆ แต่จะแตกต่างกันที่ค่าลักษณะต่าง ๆ ในแต่ละยีนส์เรียกว่า แอลลี (Allele) ซึ่งแบบต่าง ๆ ของยีนส์ที่มีแอลลีต่างกันในแต่ละตำแหน่งเรียกว่า ยีโนไทป์ (Genoype) สำหรับลักษณะภายนอกที่ปรากฏออกมาให้เห็นเรียกว่า ฟีโนไทป์ (Phenotype) ตัวอย่างคู่โครโมโซมของมนุษย์ ดังแสดงในรูปท่ 2.1 ซึ่งประกอบด้วยยีนส์ที่มีลักษณะ ต่าง ๆ กัน



2.1 แสดงลักษณะทางพันธุศาสตร์ของโครโมโซมมนุษย์

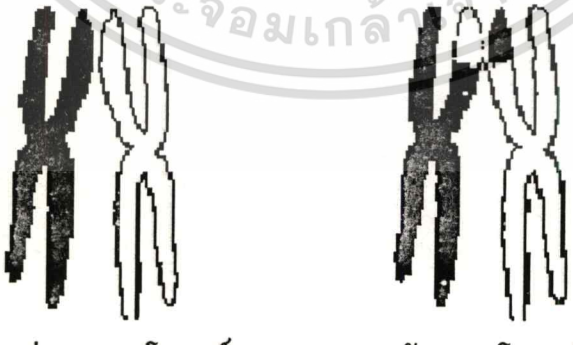
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถ่ายทอดลักษณะทางพันธุกรรมเป็นการถ่ายทอดลักษณะต่าง ๆ ของสิ่งมีชีวิตที่เกิดขึ้นเมื่อมีการแบ่งตัวของเซล ซึ่งมี 2 แบบคือ

1. การแบ่งตัวแบบไมโตสิส (Mitosis) เป็นการเพิ่มจำนวนเซล โดยโครโมโซมแต่ละตัวจะเพิ่มจำนวนตัวเองเป็นสอง และเยื่อหุ้มนิวเครียสจะสลายลงเพื่อแยกโครโมโซมที่เพิ่มจำนวนขึ้นออกจากโครโมโซมเดิมแล้วเยื่อหุ้มนิวเครียสจะถูกสร้างขึ้นใหม่เป็น 2 เซล

2. การแบ่งตัวแบบไมโอสิส (Meiosis) เป็นการแบ่งตัวของเซลสืบพันธ์ โดยโครโมโซมจากเซลพ่อและแม่อย่างละ 1 โครโมโซม จับคู่กันและต่างก็จำลองแบบของตนเพิ่มมาอีก ทำให้ได้โครโมโซมทั้งหมดเพิ่มขึ้นเป็น 2 เท่า โครโมโซมพ่อและแม่พร้อมทั้งแบบจำลองจะแยกคู่ไปรวมกันเป็น 2 นิวเครียส กลายเป็นเซลใหม่ 2 เซล ซึ่งแบ่งตัวต่อไป โดยโครโมโซมพ่อและแม่แยกตัวออกจากแบบจำลอง ทำให้ได้เซลใหม่ 4 เซล

วิธีการแบบไมโอสิสโครโมโซมจะมีโอกาสแลกเปลี่ยนบางส่วนซึ่งกันและกัน เรียกว่าครอสโอเวอร์ (Crossover) การครอสโอเวอร์จะเกิดขึ้นระหว่างโครโมโซมพ่อกับโครโมโซมแม่เนื่องจากยีนส์ แต่ละยีนส์ที่เรียงตัวกันบนโครโมโซมไม่ได้อยู่กันอย่างหนาแน่น และมีระยะห่างไม่สม่ำเสมอ ช่องว่างระหว่างยีนส์นี้เองที่สามารถแตกออกมาได้ขณะที่มีการครอสโอเวอร์ และมีการแลกเปลี่ยนยีนส์ของโครโมโซม โดยส่วนที่อยู่หลังรอยแตกทั้งหมดจะถูกย้ายไปอยู่อีกโครโมโซม นอกจากนี้ยังสามารถแตกอีกกี่แห่งก็ได้ ซึ่งขึ้นอยู่กับความสามารถที่จะเชื่อมกันได้มากน้อยเพียงไร ดังแสดงในรูปที่ 2.2

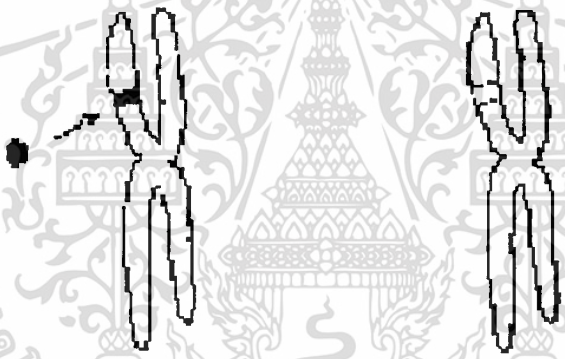


ก่อนครอสโอเวอร์ หลังครอสโอเวอร์

รูปที่ 2.2 แสดงก่อนและหลังการครอสโอเวอร์ของโครโมโซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ที่เกิดจากการครอสโอเวอร์คือ ได้ลักษณะต่าง ๆ มาอยู่รวมกัน ทำให้สิ่งมีชีวิตรุ่นลูกหลานมีความหลากหลาย และ อาจทำให้เกิดโอกาสเกิดสิ่งมีชีวิต ที่มีลักษณะที่ดีรวมกันอยู่ได้อย่าง พอเหมาะ ถ้าเซลเกิดใหม่ โดยไม่มีการครอสโอเวอร์แล้ว โครโมโซมใด ที่เคยมียีนส์ลักษณะใดก็จะมีลักษณะนั้นอยู่เรื่อย ๆ โอกาสที่สิ่งมีชีวิตนั้น จะปรับตัวให้ดีขึ้นย่อมมีได้ยากกว่า นอกจากนี้ยังมี การผ่าเหล่า (Mutation) คือการเปลี่ยนแปลงของยีนส์ที่มีลักษณะแปลก ๆ มากขึ้น เนื่องจากขบวนการวิวัฒนาการทางธรรมชาตินั้นช้ามาก เพราะกว่าที่ธรรมชาติจะรับสภาพแวดล้อมให้สิ่งมีชีวิตค่อย ๆ ปรับตัวเองให้เหมาะสมนั้นมีโอกาสน้อยมาก การผ่าเหล่านั้นทุกลักษณะในแต่ละยีนส์ย่อมมีโอกาสที่จะเกิดการเปลี่ยนแปลงไปจากเดิมได้พอ ๆ กัน และถ้าเหมาะสมกับสภาพแวดล้อมในขณะนั้นก็จะคงอยู่ต่อไป แต่ถ้าการเปลี่ยนแปลงใดเกิดผิดจังหวะ คือไม่เหมาะสมกับสภาพแวดล้อมขณะนั้น ๆ ก็จะไม่ถูกคัดเลือกและหายไปในที่สุด ดังรูปที่ 2.3



ก่อน มิวเตชัน หลัง มิวเตชัน

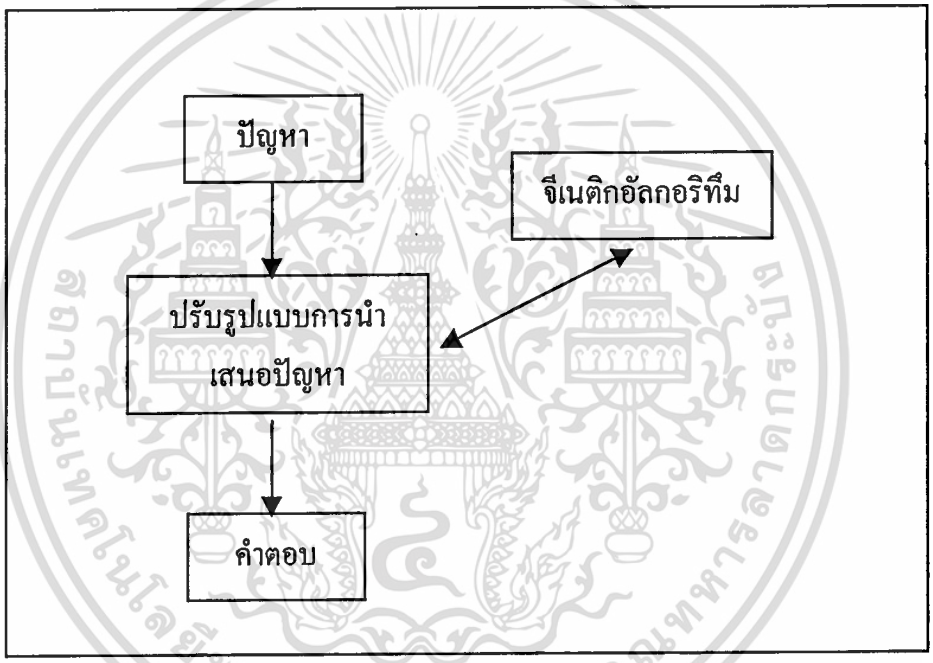
รูปที่ 2.3 แสดงการเกิดมิวเตชันของโครโมโซม

2.2 จีเนติกอัลกอริทึม

ปี ค.ศ. 1975 John Holland ได้ศึกษาทฤษฎีที่เกี่ยวข้องกับวิวัฒนาการทางธรรมชาติ (Natural Evolution) ในการให้กำเนิดประชากรสิ่งมีชีวิตในรุ่นต่อ ๆ ไป โดยกระบวนการทางชีววิทยาประกอบด้วยการคัดเลือกทางธรรมชาติ (Natural Selection) คือ สิ่งมีชีวิตที่แข็งแรงกว่า ย่อมมีโอกาสอยู่รอดได้มากกว่าสิ่งมีชีวิตที่อ่อนแอ โครโมโซมที่สามารถอยู่รอดได้ก็จะถูกถ่ายทอดยีนส์ที่มีลักษณะที่ดีเหล่านั้นไปยังลูกหลานได้มากกว่าเช่นกัน และกระบวนการทางพันธุศาสตร์ (Genetic Operation) คือการกำเนิดโครโมโซมใหม่เพื่อถ่ายทอดยีนส์จากการครอสโอเวอร์ หรือกลายพันธุ์จากมิวเตชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแนวคิดดังกล่าว Holland จึงได้นำมาปรับใช้กับคอมพิวเตอร์เพื่อช่วยแก้ปัญหาในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุด โดยมีจุดมุ่งหมายที่จะศึกษาระบบปรับปรุงการประมวลผลเอง (Self Adaptive Process) และสร้างระบบผู้เชี่ยวชาญ (Expert System) โดยอาศัยแนวคิดของระบบการคัดเลือกทางธรรมชาติเรียกว่า จีเนติกอัลกอริทึม (Genetic Algorithms : GA) เพื่อปรับปรุงการหาคำตอบที่ดีขึ้น หลักการเบื้องต้นในการใช้จีเนติกอัลกอริทึมแก้ปัญหาคือ ต้องปรับปรุงรูปแบบปัญหาให้เหมาะสมกับการนำเสนอของจีเนติกอัลกอริทึม ดังรูปที่ 2.4



รูปที่ 2.4 แสดงหลักการเบื้องต้นของจีเนติกอัลกอริทึม

จีเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบ โดยการเลียนแบบการคัดเลือกทางธรรมชาติและธรรมชาติทางพันธุกรรมซึ่งอาศัยหลักการสุ่ม เพื่อปรับปรุงความสามารถในการค้นหาคำตอบที่ดีขึ้น โดยมีวิธีการคือ

1. ค้นหาคำตอบภายใต้โครงสร้างของปัญหา อันเกิดจากการคำตอบรหัส (Coding) รูปแบบแบบโครงสร้างจากกลุ่มตัวแปรต่าง ๆ ของปัญหานั้น ไม่ใช่ค้นหาคำตอบจากค่าของกลุ่มตัวแปรนั้น

2. ค้นหาคำตอบโดยพิจารณาจากประชากรคำตอบ หรือกลุ่มคำตอบ ไม่ใช่จากค่าของกลุ่มตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ค้นหาคำตอบจากผลลัพธ์ของกลุ่มค่าตัวแปรที่เป็นฟังก์ชันเป้าหมายของปัญหา

4. ค้นหาคำตอบโดยอาศัยการถ่วงน้ำหนัก ความเหมาะสม ของแต่ละคำตอบจากกลุ่มคำตอบนั้น ๆ

2.3 ฟังก์ชันเป้าหมายกับฟังก์ชันความเหมาะสม

การหาคำตอบที่ดีที่สุดของปัญหาของจีเนติกอัลกอริทึม มีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา วิธีการของจีเนติกอัลกอริทึมจะไม่พิจารณาจากขั้นตอนของการแก้ไขปัญหา แต่จะพิจารณาโดยตัดสินค่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือเป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่ จากฟังก์ชันเป้าหมาย (Objective Function : f) เนื่องจากแต่ละปัญหาสามารถกำหนดฟังก์ชันเป้าหมาย ซึ่งแสดงความสัมพันธ์ของแต่ละตัวแปร พารามิเตอร์ เงื่อนไข หรือ ข้อกำหนดต่าง ๆ สำหรับฟังก์ชันความเหมาะสม (Fitness Function : F) เป็นฟังก์ชันที่ใช้เป็นตัวกำหนดค่าความเหมาะสมของแต่ละโครโมโซมว่ามีโอกาสจะถูกคัดเลือกมากน้อยเพียงใด โดยทั่วไปมักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสม หรืออาจใช้ฟังก์ชันเป้าหมายที่ถูกปรับให้เหมาะสมกับการนำเสนอจีเนติกอัลกอริทึม เป็นฟังก์ชันความเหมาะสมก็ได้

2.4 รูปแบบโครโมโซม

สำหรับ จีเนติกอัลกอริทึม ยีนส์ ซึ่งเป็นตัวแสดงค่าคำตอบ ของปัญหา ที่เปลี่ยนไปตามการประยุกต์ใช้งาน ซึ่งโดยทั่วไปยีนส์หมายถึงตัวแปร พารามิเตอร์ เงื่อนไขหรือข้อกำหนดต่าง ๆ ลำดับของยีนส์บนโครโมโซมจะอยู่ในรูปแบบสตริง (string) ประกอบด้วยบิท (bit) ซึ่งลักษณะที่เป็นไปได้เรียกว่า ค่าของบิท (bit value) รูปแบบค่าบิทที่จัดเรียงบนโครโมโซมคือ ยีโนไทป์ (genotype) ที่แสดงถึงค่าของตัวแปร พารามิเตอร์ต่าง ๆ ที่เป็นไปได้ชุดหนึ่ง หรือฟีโนไทป์ (phenotype) นั่นเอง

2.5 การทำงานของจีเนติกอัลกอริทึม

เมื่อกำหนดรูปแบบโครโมโซมและฟังก์ชันความเหมาะสมได้แล้วจีเนติกอัลกอริทึม จะสร้างวิวัฒนาการกลุ่มคำตอบในรุ่นต่อ ๆ ไปโดยมีการทำงาน ดังรูปที่ 2.5 ซึ่งมี 4 ขั้นตอน คือ

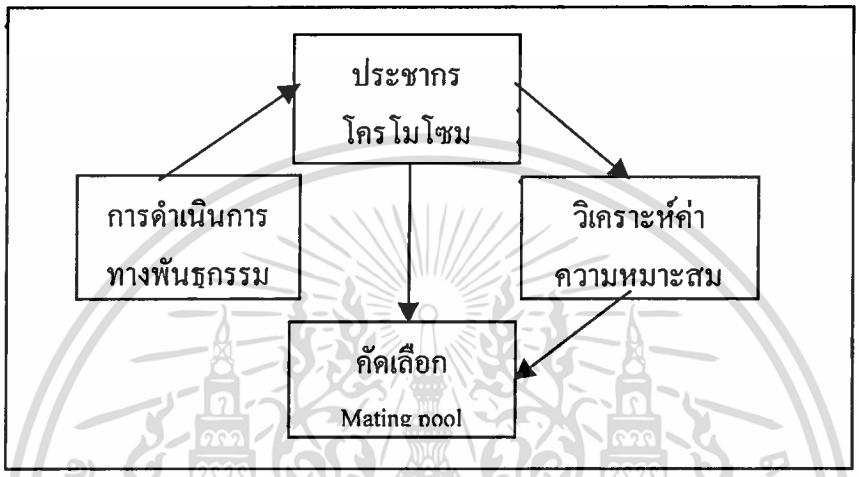
1. สร้างประชากรโครโมโซมรุ่นเก่า ตามรูปแบบที่กำหนดไว้ โดยประชากรต้นกำเนิด (Initial Population) เกิดจากการสร้างชุดโครโมโซมโดยสุ่มค่าแต่ละบิท

2. วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซมโดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่าง ๆ ของแต่ละบิท และคำนวณค่าความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้าง mating pool คือชุดโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ ที่สามารถอยู่รอดเป็นต้นแบบ โดยพิจารณาถ่วงน้ำหนักจากค่าความเหมาะสมของแต่ละโครโมโซม

4. ดำเนินการทางพันธุศาสตร์ โดยสุ่มจับคู่โครโมโซมต้นแบบใน mating pool เพื่อสร้างประชากรรุ่นใหม่ ซึ่งตัวดำเนินการทางพันธุศาสตร์ประกอบด้วยครอสโอเวอร์หรือมิวเตชัน



รูปที่ 2.5 การทำงานของ จีเนติกอัลกอริทึม

การค้นหาคำตอบของจีเนติกอัลกอริทึมจะประมวลผลซ้ำ ๆ จนกว่าจะได้รับคำตอบที่พอใจตามเงื่อนไขที่ได้ตั้งไว้ หรือในระยะเวลาตามจำนวนรุ่นที่ต้องการ ซึ่งแสดงอัลกอริทึมการทำงานของจีเนติกอัลกอริทึมดังนี้

Simple Genetic Algorithm

Begin

Initial population;

Evaluate population;

While termination criterion not reached

Begin

Select solution for next population;

Crossover;

Mutation;

Evaluate population;

End;

End;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับจีเนติกอัลกอริทึม ตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของสตริง ซึ่งมักเรียกว่าโครโมโซม ประกอบด้วยอักขระหรือบิท แต่ละตำแหน่งของโครโมโซมประกอบด้วยค่าของอักขระหรือค่าของบิท ที่แสดงถึงโครงสร้างของแต่ละโครโมโซมที่มีค่าตัวแปรหรือพารามิเตอร์ของปัญหาที่แตกต่างกัน และเป็นตัวกำหนดค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของแต่ละปัญหา ซึ่งในตารางที่ 2.1 อธิบายคำศัพท์เปรียบเทียบที่ใช้ทางพันธุศาสตร์กับจีเนติกอัลกอริทึม

Natural Genetic	Genetic Algorithm
Chromosome	String
Gene	Character Bit
Allele	Character value, Bit value
Locus	String position
Genotype	Structure
Phenotype	Decode structure

ตารางที่ 2.1 แสดงคำศัพท์ที่ใช้ทางพันธุศาสตร์ กับ จีเนติกอัลกอริทึม

จีเนติกอัลกอริทึมในยุคเริ่มแรกของ Holland นั้น คือ จีเนติกอัลกอริทึมแบบง่าย (Simple Genetic Algorithm : SGA) ซึ่งมีขั้นตอนพื้นฐานไม่มากนัก แบ่งออกเป็น 2 ส่วนคือการเตรียมการและการทำงาน

สำหรับในส่วนการเตรียมการนี้เป็นขั้นตอนการปรับปรุงรูปแบบของปัญหาให้เหมาะสมสำหรับการนำเสนอเพื่อใช้สำหรับแก้ปัญหานั้น ๆ ประกอบด้วย

1. กำหนดฟังก์ชันความเหมาะสม เพื่อความสะดวกและง่ายต่อการเข้าใจ จะกำหนดตัวอย่างการหาคำตอบของปัญหาการหาค่าสูงสุดของฟังก์ชัน $y = 2x^2$ ที่ x มีค่าระหว่างจำนวนเต็ม I

ตัวอย่าง ฟังก์ชันเป้าหมายคือ $f = 2x^2$

กำหนดฟังก์ชันความเหมาะสม คือ $F = 2x^2$

ซึ่งคำตอบที่ดีที่สุดคือ ค่า x ที่มีค่าเหมาะสมสูงสุด MAX(F)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กำหนดรูปแบบโครโมโซม โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปไบนารีโครโมโซม และมีความยาว (Chromosome Length : Ichrom) ตามที่กำหนด เช่น

$$B_1 B_2 B_3 \ldots B_{ichrom} \qquad \text{ซึ่ง } B_i \in 1\,[0,1]$$

ตัวอย่าง วิธีแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี 6 บิท (Ichrom = 6) ดังนั้น โครโมโซมของปัญหา จะมีค่าอยู่ในช่วง 000000 ถึง 111111 ซึ่งเมื่อถอดรหัสแล้วจะมีค่าอยู่ช่วง 0 ถึง 63

สำหรับในส่วนการทำงานจะประกอบด้วย

1.ประชากรรุ่นเก่า (Old Population) เป็นชุดโครโมโซมที่ถูกคัดเลือกไปเป็นโครโมโซมต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) ในวิวัฒนาการรุ่น (generation : gen) ต่อไป โดยประชากรเริ่มต้นที่ gen = 0 และถูกสร้างขึ้นโดยการสุ่มตามจำนวนโครโมโซม ในแต่ละรุ่น (Populaiton Size : popsize) ที่กำหนด

ตัวอย่าง

ลำดับ	โครโมโซม
1	101110
2	111001
3	101000
4	110011

ชุดโครโมโซมเริ่มต้นนี้เป็นชุดโครโมโซมที่กำหนดให้ในแต่ละรุ่น จะประกอบด้วย 4 โครโมโซม ซึ่งแต่ละโครโมโซม เกิดจากการสุ่มค่าไบนารีจำนวน 6 ครั้ง

2.การวิเคราะห์ค่าความเหมาะสมเป็นขั้นตอน การถอดรหัสจากโครงสร้างโครโมโซมที่กำหนดไว้ เพื่อคำนวณค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของปัญหา ในที่นี้ฟังก์ชันความเหมาะสม คือ $F = 2x^2$

ตัวอย่าง

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม (F)
1	101110	46	4232
2	111001	57	6498
3	101000	40	3200
4	110011	51	5202

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การคัดเลือก เป็นขั้นตอนการคัดเลือกโครโมโซมต้นแบบเพื่อสร้าง mating pool โดยโครโมโซมรุ่นเก่าเป็นโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ เพื่อใช้ในการสร้างโครโมโซมรุ่นลูก-หลานต่อไป โครโมโซม ที่มีค่าความเหมาะสมที่ดีจะถูกกำหนดน้ำหนักค่าความน่าจะเป็นที่จะถูกเลือก การกำหนดค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้ง (Probability of Selected Value : pselect) ของแต่ละโครโมโซมโดยกำหนดจากค่าความเหมาะสมเทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่ 2.1

$$\text{pselect}_i = F_i / \sum F \tag{2.1}$$

ซึ่งสามารถคำนวณค่าคาดหวังที่จะสุ่มได้ (Expected Value : E) ของแต่ละโครโมโซมในแต่ละรุ่นดังสมการที่ 2.2

$$E_i = \text{pselect}_i * \text{popsize} = F_i / F \tag{2.2}$$

สำหรับวิธีการสุ่มโครโมโซมต้นแบบ เป็นการจำลองการหมุนวงล้อถ่วงน้ำหนัก (Rollet Wheel : RW) ซึ่งกำหนดขนาดแต่ละช่องของวงล้อตามความน่าจะเป็นที่สุ่มได้ในแต่ละครั้งของโครโมโซม ซึ่งมีวิธีการดังนี้

1. หาค่า ความเหมาะสมของแต่ละโครโมโซม
2. หาค่าความน่าจะเป็นที่สุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม
3. หาความถี่สะสม (q) ของค่าความน่าจะเป็นของแต่ละโครโมโซมดังสมการที่ 2.3

$$q_i = \sum_{j=1}^{i} pselect_j \tag{2.3}$$

4. สร้างเลขสุ่มจำนวนจริง (r) ที่มีค่าอยู่ในช่วง [0.0, 1.0]
5. เลือกโครโมโซมลำดับที่ r ซึ่งมีค่าอยู่ระหว่าง q_{i-1} และ q_I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม (F)	ค่าความน่าจะเป็น pselect	จำนวนที่คาดหวัง E_i	จำนวนที่สุ่มได้จาก RW
1	101110	46	4232	0.221	0.885	1
2	101101	57	6498	0.340	1.359	2
3	101000	40	3200	0.167	0.669	0
4	110011	51	5202	0.272	1.088	1
	รวม		19132	1.000	4.000	
	ค่าเฉลี่ย		4783	0.250	1.000	
	ค่าสูงสุด		6498	0.340	1.359	

ตัวอย่างการกำหนดค่าความน่าจะเป็น โดยกำหนดค่าความเหมาะสมเทียบกับผลรวมของค่าความเหมาะสมทั้งหมด จะเห็นได้ว่าในการคัดเลือกโครโมโซมต้นแบบจาก 4 โครโมโซมนี้โอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 1, 2, 3, 4 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.221, 0.340, 0.167 และ 0.272 ตามลำดับ และจำนวนโครโมโซมต้นแบบที่สุ่มได้จากการหมุนวงล้อมีดังนี้

ลำดับโครโมโซม	1	2	3	4
ค่าความเหมาะสม (F)	4232	6498	3200	5202
ค่าความน่าจะเป็นที่สุ่มได้แต่ละครั้ง ($pselect_i$)	0.221	0.340	0.167	0.272
ความถี่สะสมค่าความน่าจะเป็น (q_i)	0.221	0.561	0.728	1.000
เลขสุ่มจากการหมุนวงล้อแต่ละครั้ง (r)	0.333	0.844	0.456	0.128
ลำดับโครโมโซมที่ถูกเลือก ($q_{i-1} \leq r \geq q_i$)	2	4	2	1

ซึ่งจำนวนที่สุ่มได้เป็นโครโมโซมต้นแบบใน mating pool ของแต่ละโครโมโซมเป็น 1, 2, 0 และ 1 ตามลำดับ จะเห็นว่าโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมสูงที่สุดจะมีโอกาสถูกเลือกมากที่สุด

4. การดำเนินการทางพันธุศาสตร์ เป็นขั้นตอนการจำลองแบบทางพันธุกรรม ซึ่งมีตัวดำเนินการพันธุศาสตร์คือ การครอสโอเวอร์ และ การมิวเตชัน โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การครอสโอเวอร์ เป็นตัวดำเนินการในการแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่ ตามอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover : P_c) เพื่อสร้างชุดโครโมโซมรุ่นใหม่ มีขั้นตอนการทำงานคือ

ขั้นตอนที่ 1 จับคู่โครโมโซม พ่อ-แม่ ใน matting pool ที่สร้างไว้จากการคัดเลือก

ขั้นตอนที่ 2 สร้างเลขสุ่มจำนวนจริง (r) ที่มีค่าอยู่ในช่วง 0.0, 1.0] โดยถ้า $r \leq P_c$ แล้ว โครโมโซมพ่อ-แม่ นั้นจึงมีการครอสโอเวอร์

ขั้นตอนที่ 3 ครอสโอเวอร์ โดยแลกเปลี่ยนส่วนคู่ของโครโมโซมพ่อ-แม่นั้น ซึ่งการครอสโอเวอร์ ของจีเนติกอัลกอริทึมแบบง่ายเป็นแบบ 1 จุด (One-point Crossover) ดังนี้

คู่โครโมโซมพ่อ-แม่ $d_1\ d_2\ d_3 \ldots d_{pos}\ d_{pos+1} \ldots d_{Ichrom}$

$m_1\ m_2\ m_3 \ldots m_{pos}\ m_{pos+1} \ldots m_{Ichrom}$

จำนวนการครอสโอเวอร์ในแต่ละรุ่นขึ้นอยู่กับค่า P_c ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรในแต่ละรุ่น popsize เท่ากับ 40 โครโมโซม และกำหนดให้ $P_c = 0.6$ แล้วจำนวนการ ครอสโอเวอร์ในแต่ละรุ่นเท่ากับ P_c * (popsize / 2) = 0.6 * (40/2) = 12 ครั้ง (การครอสโอเวอร์ 1 ครั้งเกิดจาก 2 โครโมโซม)

ตัวอย่าง กำหนด $P_c = 0.5$ โครโมโซม พ่อ-แม่ ใน mating pool จากการครอสโอเวอร์ดังนี้

ลำดับที่คัดเลือก	Mating pool	สุ่มจับคู่พ่อ-แม่	เลขสุ่ม (r)	ก่อนครอสโอเวอร์	ตำแหน่ง pos	หลังครอสโอเวอร์	x	F	ลำดับโครโมโซมถูก
2	111001	1,2	0.321	101|110	3	101|001	41	3362	1
4	110011		≤ 0.5	111|001		111|110	62	7688	2
2	111001	2,4	0.654	ไม่ครอสโอเวอร์		111001	57	6498	3
1	101110		≤ 0.5			110011	51	5202	4
	รวม							22750	
	ค่าเฉลี่ย							56880	
	ค่าสูงสุด							7688	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการจับคู่ใน mating pool ได้โครโมโซมลำดับที่ 1 คู่กับโครโมโซมลำดับที่ 2 และลำดับที่ 1 คู่กับลำดับที่ 4 แต่เฉพาะโครโมโซม คู่แรกที่มีการครอสโอเวอร์ เพราะค่า r $\leq$ 0.5 โดยตำแหน่งที่มีการครอสโอเวอร์คือ pos = 3 หลังจากการครอสโอเวอร์มีค่าความเหมาะสมดีกว่าโครโมโวมพ่อ-แม่ทั้งหมด คือ 7688 ซึ่งแสดงให้เห็นว่าการจำลองกระบวนการครอสโอเวอร์ตามธรรมชาติช่วยสร้างคำตอบที่ดีขึ้นได้

การมิวเตชัน เป็นการดำเนินการผ่าเหล่าตัวหนึ่งที่ให้โครโมโซม มีค่าความเหมาะสมดีขึ้นหลังการครอสโอเวอร์โดยกลับค่าบิทที่สุ่มได้ตามอัตราความน่าจะเป็นของการมิวเตชัน (Probability of Mutation : P_m) ที่กำหนด เช่น

ก่อนการมิวเตชัน $C_1, C_2, C_3, \ldots\ldots\ldots\ldots, C_{pos}, \ldots\ldots\ldots\ldots, C_{lchrom}$

สำหรับ การมิวเตชันของ SGA เป็นแบบไบนารีมิวเตชัน คือการกลับค่าคอมพลีเมนต์จาก 0 เป็น 1 ดังนั้นจะได้โครโมโซมใหม่คือ

หลังการมิวเตชัน $C_1, C_2, C_3, \ldots\ldots\ldots\ldots, C'_{pos}, \ldots\ldots\ldots\ldots, C_{lchrom}$

จำนวนการมิวเตชัน ในแต่ละรุ่นขึ้นอยู่กับค่า P_m ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรในแต่ละรุ่น popsize เท่ากับ 40 โครโมโซม แต่ละโครโมโซมประกอบด้วย 6 บิท และ กำหนดให้ $P_m = 0.02$ แล้ว จำนวนการมิวเตชันเท่ากับ P_m * popsize * lchrom = 0.02 * 40 * 6 = 5 บิท

ตัวอย่าง กำหนด $P_m = 0.1$ การดำเนินการมิวเตชันโครโมโซมลูกที่ได้จากการครอสโอเวอร์ดังนี้

ลำดับ	ก่อนมิวเตชัน	เลขสุ่ม	หลังมิวเตชัน	x	F
1	101001	0.896, 0.254, 0.753, 0.062, 0.351, 0.684	101101	45	4050
2	111110	0.984, 0.421, 0.564, 0.241, 0.958, 0.547	111110	62	7688
3	111001	0.552, 0.637, 0.258, 0.491, 0.746, 0.029	111000	56	6272
4	110011	0.951, 0.874, 0.464, 0.829, 0.648, 0.214	110011	51	5202
	รวม				23212
	ค่าเฉลี่ย				5803
	ค่าสูงสุด				7688

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการสุ่มตำแหน่งที่จะมีการมิวเตชัน โดยสร้างเลขสุ่ม r ของแต่ละบิทโครโมโซมแล้ว ตำแหน่งที่ 3 ของโครโมโซมที่ 1 และตำแหน่งที่ 4 ของโครโมโซมที่ 3 เป็นตำแหน่งที่ค่า $r \leq 0.1$ ตามอัตราการมิวเตชัน ทำให้ค่าความเหมาะสมจาก 3362 และ 6498 เป็น 4050 และ 6272 ตามลำดับ จะเห็นได้ว่าการมิวเตชันเป็นตัวดำเนินการที่ทำให้เกิดค่าความเหมาะสมสูงขึ้นหรือต่ำลงก็ได้ แต่อย่างไรก็ตามค่าเฉลี่ยของความเหมาะสมดีขึ้นจาก 5688 เป็น 5803 แสดงถึงการหาคำตอบของจีเนติกอัลกอริทึมแบบง่ายดีขึ้น

5.ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากการวิวัฒนาการต่าง ๆ ทั้งหมด ซึ่งประชากรรุ่นใหม่ที่เกิดขึ้นจะถูกถ่ายทอด กลายเป็นประชากรรุ่นเก่า สำหรับวิวัฒนาการในรุ่นต่อไป ซึ่งเรียกวิวัฒนาการแบบนี้ว่า การรีโพรดักชัน (Reproduction) กระบวนการต่าง ๆ จะถูกปฏิบัติซ้ำ ๆ จนกระทั่งถึงรุ่นที่มากที่สุด (Max generation : maxgen) ที่ต้องการ

จะเห็นได้ว่าจีเนติกอัลกอริทึมเป็นวิวัฒนาการทางธรรมชาติที่นำมาประยุกต์ใช้กับปัญหาต่าง ๆ ซึ่งมีการทำงานเบื้องต้นเป็นจีเนติกอัลกอริทึมแบบง่าย โดยมีโครโมโซมเป็นแบบไบนารี และมีตัวดำเนินการทาง พันธุศาสตร์ คือ การครอสโอเวอร์ และการมิวเตชัน จะเห็นว่าค่าความเหมาะสมของ จีเนติกอัลกอริทึมแบบง่าย มีค่าสูงขึ้นและลดลงได้ด้วยวิธีการสุ่ม เพื่อช่วยให้จีเนติกอัลกอริทึมสามารถหาคำตอบที่ดีขึ้นโดยปรับปรุงการทำงานของจีเนติกอัลกอริทึม ดังนี้

1. รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี เนื่องจากในการค้นหาคำตอบของจีเนติกอัลกอริทึมแบบง่าย นั้นมีโอกาสที่จะสูญเสียโครโมโซมในรุ่นเก่าที่มีค่าความเหมาะสมที่ดีไปได้ ซึ่งจะทำให้คำตอบในรุ่นถัดไปมีค่าความเหมาะสมดีมากขึ้น หรือน้อยลงได้ ดังนั้นหากปรับปรุงจีเนติกอัลกอริทึมแบบง่ายให้ควบคุมการหาคำตอบโดยรักษาโครโมโซมที่ดีไว้แล้วจะช่วยให้วิวัฒนาการในการหาคำตอบในรุ่นถัดไปดีขึ้นเรื่อย ๆ โดยมีวิธีการดังนี้

- กำหนดจำนวนโครโมโซมที่ดีที่สุด (# best) ของรุ่นเก่าที่ต้องการรักษาเป็น 1, 2, 4, ...
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 1 ให้สร้างชุดโครโมโซมรุ่นใหม่ที่มีค่าความเหมาะสมน้อยที่สุด
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 2, 4, ... ให้คัดลอกโครโมโซมที่ดีที่สุดในรุ่นเก่าตามจำนวนที่กำหนดมาเป็นโครโมโซมรุ่นใหม่ แล้วจึงสร้างโครโมโซมรุ่นใหม่ส่วนที่เหลือต่อไปดังรูปที่ 2.6 แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครโมโซม และรูปที่ 2.7 แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 2 โครโมซม

2. การครอสโอเวอร์แบบ 2 จุด การแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่นั้น บางครั้งหากแลกเปลี่ยนเพียงบางช่วง ของโครโมโซมแล้ว จะสร้างโครโมโซมที่ดีกว่า เช่น การหาค่าสูงสุดของฟังก์ชัน $y = 2x^2$ ของคู่โครโมโซมพ่อ-แม่ 101110 และ 111001 ซึ่งมีค่าความเหมาะสมเป็น 4232 และ 6498 หากแลกเปลี่ยนค่าบิทตำแหน่งที่ 4 และ 5 เท่านั้น จะทำให้เกิดโครโมโซมลูกคือ 101000 และ 111111 มีค่าความเหมาะสมเป็น 3200 และ 7938 ซึ่งโครโมโซม 111111 เป็นโครโมโซมที่ให้คำตอบที่ดีที่สุดที่ต้องการ ดังนั้นการพัฒนาตัวดำเนินการครอสโอเวอร์เป็นแบบ 2 จุด (Two-point Crossover) จะทำให้จีเนติกอัลกอริทึมแบบง่าย ค้นหาคำตอบที่ดีขึ้นได้ ดังรูปที่ 2.8 มีวิธีการดังนี้

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะครอสโอเวอร์ตามลำดับ ซึ่ง pos_1 มีค่าอยู่ระหว่าง [1, lchrom-1] และ pos_2 มีค่าอยู่ระหว่าง [1, lchrom] โดยที่ pos_1 น้อยกว่า pos_2
- แลกเปลี่ยนค่าในแต่ละบิทของคู่โครโมโซมพ่อ-แม่ ตั้งแต่ตำแหน่งที่ pos_1+1 ถึง pos_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้				โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1. สร้างโครโมโซมรุ่นใหม่ทั้งหมด				
				ครอสโอเวอร์	มิวเตชัน	ค่าความเหมาะสม		
1	101110	4232	1	101110 pos 1 101001 pos	101001	3362		
2	111001	6498	2	111001 3.5 111110	110100	5408	110100	5408
3	101000	3200	0	110011 2 111001 3	110001	4802	110001	4802
4	110011	5202	1	111001 110011	11001	5202	110011	5202
				2. คัดลองจาก โครโมโซมรุ่นเก่าลำดับที่ 2			111001	6498

รูปที่ 2.6 การรีโพรดักชันรักษาค่าความเหมาะสมที่ดี 1 โครโมโซม

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้				โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1. สร้างโครโมโซมรุ่นเก่าลำดับที่ 2 และ 4			111001	6498
							110011	5202
1	101110	4232	1	2. สร้างโครโมโซมใหม่ทั้งหมด				
2	111001	6498	2	ครอสโอเวอร์	มิวเตชัน	ค่าความเหมาะสม		
3	101000	3200	0	101110 pos 3 101001 pos	101001	3362	110001	3362
4	110011	5202	1	111001 111110 3.5	110100	5408	110011	5408

รูปที่ 2.7 การรีโพรดักชันรักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

คู่โครโมโซมพ่อ-แม่	$d_1\ d_2\ d_3 \ldots d_{pos1}\quad d_{pos1+1} \ldots d_{pos2}\quad d_{pos2+1} \ldots d_{lchrom}$
	$m_1\ m_2\ m_3 \ldots m_{pos1}\quad m_{pos1+1} \ldots m_{pos2}\quad m_{pos2+1} \ldots m_{lchrom}$
คู่โครโมโซมลูก	$d_1\ d_2\ d_3 \ldots d_{pos1}\quad m_{pos1+1} \ldots m_{pos2}\quad d_{pos2+1} \ldots d_{lchrom}$
	$m_1\ m_2\ m_3 \ldots m_{pos1}\quad d_{pos1+1} \ldots d_{pos2}\quad m_{pos2+1} \ldots m_{lchrom}$

(pos_1, pos_2 คือตำแหน่งที่สุ่มได้ซึ่ง $pos_1 \in [1, lchrom-1]$, $pos_2 \in [1, lchrom]$ และ $pos_1 \quad pos_2$)

รูปที่ 2.8 การคอรสโอเวอร์แบบ 2 จุด

3. ไบนารีมิวเตชันแบบกำหนดค่าบิท เนื่องจากการหาคำตอบของจีเนติกอัลกอริทึมแบบง่ายนั้น กระบวนการไบนารีมิวเตชัน อาจทำให้โครโมโซมที่เปลี่ยนแปลงไปให้คำตอบที่ดีลดลงและทำให้สูญเสียโครโมโซมที่ดีไป เช่น โครโมโซม 111110 มีค่าความเหมาะสมเป็น 7688 หากสุ่มได้บิทตำแหน่งที่ 1 เกิดมิวเตชันแล้ว โครโมโซมที่เกิดขึ้นจากการมิวเตชันคือ *0*11110 ทำให้มีค่าความเหมาะสมลดลงเป็น 1800 แต่ในบางครั้งข้อดีหรือจุดเด่นของปัญหาจะสามารถนำมาปรับให้เข้ากับการค้นหาคำตอบที่ดีขึ้นได้ สำหรับในการหาค่าสูงสุดของฟังก์ชัน $y = 2x^2$ นี้ค่าบิทของโครโมโซมที่เป็น 1 จะทำให้ค่าความเหมาะสมสูงขึ้นเสมอ ดังนั้นหากปรับปรุงไบนารีมิวเตชันให้เป็นแบบกำหนดค่าแน่นนอนให้กับบิทที่เกิดมิวเตชัน โดยกำหนดให้บิทที่เกิดมิวเตชันมีค่าบิทเป็น 1 เสมอจะช่วยปรับแนวทางการค้นหาคำตอบของจีเนติกอัลกอริทึมแบบง่ายดีขึ้น เช่น หากประยุกต์ไบนารีมิวเตชันที่กำหนดค่าบิทให้เป็น 1 เสมอกับโครโมโซม 111110 ในตำแหน่งที่ 1 แล้วโครโมโซมที่เกิดขึ้นจะเหมือนเดิม และยังเป็นการรักษาโครโมโซมที่มีค่าความเหมาะสมที่ดีไว้ด้วย

4. อินเวอร์ชัน (Inversion) เป็นตัวดำเนินการที่ประยุกต์เพิ่มเติมในจีเนติกอัลกอริทึมแบบง่าย โดยจำลองแบบลักษณะของการอินเวอร์ชันในทางพันธุศาสตร์ ที่เป็นลักษณะของการกลับหัวกลับหางส่วนของยีนส์ภายในโครโมโซมที่อาจช่วยให้เกิดโครโมโซมที่ดีขึ้นได้ โดยการกลับส่วนค่าบิทภายในช่วงตำแหน่งของโครโมโซมที่สุ่มได้ตามอัตราความน่าจะเป็นของการอินเวอร์ชัน แต่ละโครโมโซม (Probability of Inversion : P_i) ที่กำหนด ดังรูปที่ 2.9 มีขั้นตอนดังนี้

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะอินเวอร์ชันตามลำดับ ซึ่ง pos_1 และ pos_2 มีค่าอยู่ระหว่าง [1, lchrom] โดยที่ pos_1 น้อยกว่า pos_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กลับค่าบิทในช่วงตำแหน่งที่ pos_1 ถึง pos_2 ของโครโมโซม โดยสลับค่าบิท pos_1 กับ pos_2 , pos_1+1 และ pos_2-1 , pos_1+2 และ pos_2-2 ,...

ก่อนอินเวอร์ชัน	$c_1\ c_2\ c_3 \cdots c_{pos1}$	$c_{pos1+1} \cdots c_{pos2-1}$	$c_{pos2} \cdots c_{lchrom}$
หลังอินเวอร์ชัน	$c_1\ c_2\ c_3 \cdots c_{pos2}$	$c_{pos2-1} \cdots c_{pos1+1}$	$c_{pos1} \cdots c_{lchrom}$

(pos_1 , pos_2 คือตำแหน่งที่สุ่มได้ซึ่ง pos_1 , $pos_2 \in [1, lchrom]$ และ pos_1 pos_2)

รูปที่ 2.9 การอินเวอร์ชัน

เช่น สุ่มโครโมโซมที่จะอินเวอร์ชันคือ 101010 มีค่าความเหมาะสมเป็น 3528 โดยสุ่มตำแหน่ง pos_1 มีค่าเท่ากับ 2 และ pos_2 เท่ากับ 5 แล้ว จะเห็นว่าการอินเวอร์ชันทำให้เกิดโครโมโซม 1*10100* ซึ่งมีค่าความเหมาะสมดีขึ้นเป็น 5408 เป็นต้น สำหรับจำนวนการอินเวอร์ชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า P_i ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 40 โครโมโซม และกำหนดให้ $P_i = 0.1$ แล้ว จำนวนการอินเวอร์ชันในแต่ละรุ่นเท่ากับ P_I * popsize = 0.1 * 0 = 4 ครั้ง

จะเห็นว่าจีเนติกอัลกอริทึม เป็นการเลียนแบบการวิวัฒนาการทางธรรมชาติที่สามารถนำมาประยุกต์ใช้กับคอมพิวเตอร์ เพื่อช่วยแก้ไขปัญหาในการหาคำตอบต่าง ๆ ซึ่งมีพื้นฐานการทำงานเบื้องต้นเป็นจีเนติกอัลกอริทึมแบบง่าย โดยมีรูปแบบโครโมโซมเป็นไบนารี และตัวดำเนินการทางพันธุศาสตร์ที่สำคัญคือ การครอสโอเวอร์และการมิวเตชันที่ไม่ซับซ้อน แต่สามารถปรับปรุงให้เข้ากับปัญหา เพื่อช่วยให้จีเนติกอัลกอริทึมมีการค้นหาคำตอบที่ดีขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

จีเนติกอัลกอริทึมกับปัญหาประเภทต่าง ๆ

จีเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบที่เลียนแบบกระบวนการทางธรรมชาติ ดังนั้นจะต้องมีการปรับรูปแบบการนำเสนอปัญหาและขั้นตอน ๆ ที่แตกต่างกัน ขึ้นอยู่กับลักษณะของแต่ละปัญหา ตัวอย่างการประยุกต์ใช้จีเนติกอัลกอริทึมแก้ปัญหาประเภทต่าง ๆ ที่จะกล่าวถึงคือปัญหาการทายคำ ปัญหาทาวเวอร์ออฟฮานอย ปัญหาการหาเส้นทางที่สั้นที่สุด เพื่อเป็นแนวทางในการประยุกต์ใช้แก้ปัญหาอื่น ๆ ต่อไป

3.1 ปัญหาการทายคำ

ปัญหาการทายคำ (Word Guessing Problem) เป็นการเล่นเกมส์ทายคำตัวอักษร เช่น การทายคำศัพท์ที่มีเงื่อนไขเป็นจำนวนครั้งในการทาย และตรวจสอบตัวอักษรทุกตัวในคำนั้นจะต้องถูกต้องตามคำศัพท์ที่ต้องการ

3.1.1 โครโมโซมของปัญหาทายคำ

ลักษณะของโครโมโซมของปัญหาทายคำประกอบด้วยบิทที่เป็นตัวอักษร (Charactor) ภายในคำที่ต้องการคำตอบ ดังนั้นรูปแบบโครโมโซมของปัญหาทายคำจึงเป็นการกำหนดรหัสค่าบิทของตัวอักษรแทนด้วยตัวเลข ดังตารางที่ 3.1 คือ อักษร a ถึง z มีค่าอยู่ในช่วง 1 ถึง 26 และตัวอักษร A ถึง Z ที่อยู่ในช่วง 27 ถึง 52 และจำนวนบิททั้งหมดของโครโมโซมจะเท่ากับจำนวนตัวอักษรทั้งหมด ($Charactor_n$) ของคำที่ต้องการหาคำตอบ

ค่าบิท	ตัวอักษร	ค่าบิท	ตัวอักษร
1	a	27	A
2	b	28	B
:	:	:	:
26	z	52	Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 การกำหนดค่าบิทของโครโมโซมปัญหาทายคำ

ซึ่งลักษณะโครโมโซมของปัญหาทายคำคือ

$Charactor_1$	$Charactor_2$	$Charactor_3$	$Charactor_4$	$Charactor_{n-5}$
H	O	U	S	E

แสดงในรูปสัญลักษณ์ได้ดังนี้

$B_1B_2B_3...B_n$ ซึ่ง $B_I \in I[1,52]$

3.1.2 ฟังก์ชันเป้าหมายของปัญหาทายคำ

ฟังก์ชันเป้าหมายของปัญหาทายคำเป็นฟังก์ชันตรวจความถูกต้องของแต่ละตัวอักษรของคำที่ต้องการ โดยไม่คำนึงถึงตัวพิมพ์เล็ก (lowercase) หรือตัวพิมพ์ใหญ่ (uppercase) ดังสมการที่ 3.1

$$f = \sum_{i=1}^{n} correct(charactor_i) \qquad (3.1)$$

โดยที่ n = จำนวนตัวอักษรทั้งหมด

correct(x) = 1 ถ้า uppercase(x) หรือ lowercase(x) ถูกต้อง

0 ถ้า uppercase(x) หรือ lowercase(x) ถูกต้อง

ดังนั้น คำตอบที่ดีที่สุดของปัญหาทายคำคือ คำที่มีอักษรถูกต้องทั้งหมด MAX(f)

3.1.3 ตัวดำเนินการทางพันธุศาสตร์สำหรับปัญหาทายคำ

การครอสโอเวอร์แบบหลายจุด (Multiple-point Crossover) โดยกำหนดจำนวนจุดในการแลกเปลี่ยนส่วน (Crossover point : CP) เช่น ถ้า CP =2 นั่นคือ การครอสโอเวอร์แบบ 2 จุด มี ขั้นตอนดังนี้

1. สุ่มตำแหน่ง pos_i ซึ่ง i = 1, 2, ..., CP คือ ตำแหน่งที่จะครอสโอเวอร์ ซึ่ง pos_i มีค่าอยู่ในช่วง [1,lchrom-1]
2. แลกเปลี่ยนค่าแต่ละบิทของคู่โครโมโซมพ่อ-แม่ ในแต่ละช่วงของตำแหน่งเหล่านั้น ซึ่งจะทำให้เกิดโครโมโซมลูก 2 โครโมโซม ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่โครโมโซมพ่อ-แม่

$d_1\ d_2 \ldots\ d_{pos1}\ \ d_{pos1+1} \ldots\ d_{pos2}\ d_{pos2+1} \ldots\ d_{poscp}\ \ d_{poscp+1} \ldots\ d_{lchrom}$

$m_1\ m_2 \ldots\ m_{pos1}\ \ m_{pos1+1} \ldots\ m_{pos2}\ m_{pos2+1} \ldots\ m_{poscp}\ \ m_{poscp+1} \ldots\ m_{lchrom}$

คู่โครโมโซมลูก

$d_1\ d_2 \ldots\ d_{pos1}\ m_{pos1+1} \ldots\ m_{pos2}\ d_{pos2+1} \ldots\ d_{poscp}\ \ m_{poscp+1} \ldots\ m_{lchrom}$

$m_1\ m_2 \ldots\ m_{pos1}\ d_{pos1+1} \ldots\ d_{pos2}\ m_{pos2+1} \ldots\ m_{poscp}\ \ d_{poscp+1} \ldots\ d_{lchrom}$

pos_i คือตำแหน่งที่สุ่มได้ ซึ่ง $pos_i \in [1, lchrom-1]$ และ pos_i ไม่เท่ากับ pos_j ถ้า i ไม่เท่ากับ j)

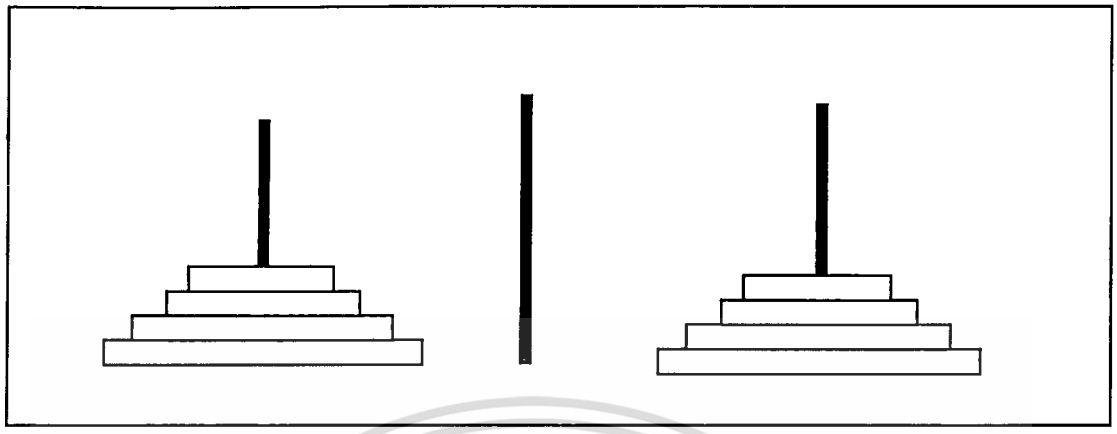
รูปที่ 3.1 การครอสโอเวอร์แบบหลายจุด

มิวเตชันแบบตัวเลข (Integer Mutation) โดยการกลับค่าบิทเดิมเป็นค่าใหม่ที่สามารถเป็นไปได้ของตำแหน่งบิทนั้น เช่น ถ้าบิทเดิมที่จะทำการมิวเตชัน เป็น 1 ให้กลับค่าเป็น 2 หรือ 3 โดยวิธีการสุ่ม

3.2 ปัญหาทาว์นเวอร์ออฟฮานอย

ปัญหาทาว์นเวอร์ออฟฮานอย (Tower of Hanoi Problem : THP) เป็นการเล่นเกมส์ชนิดหนึ่ง ซึ่งประกอบด้วนแกนหลัก 3 แกนบนกระดาน และกลุ่มของวงแหวนกลมขนาดต่างกัน โดยวงแหวนทั้งหมดถูกวางอยู่ในแกนหลักเริ่มต้นจากขนาดเล็กถึงขนาดใหญ่ โดยลำดับจากบนลงล่าง วิธีการเล่นเกมส์โดยการเคลื่อนย้าย (step) วงแหวนทั้งหมดไปยังแกนเป้าหมายที่ต้องการ ซึ่งการเรียงลำดับของวงแหวนคงเหมือนเดิมดังรูปที่ 3.2 โดยเงื่อนไขในการเคลื่อนย้ายแต่ละครั้งคือ เคลื่อนย้ายครั้งละ 1 วงแหวน และวงแหวนที่มีขนาดใหญ่กว่าห้ามวางบนวงแหวนที่มีขนาดเล็กกว่า การเล่นที่ดีที่สุดคือจำนวนครั้งในการเคลื่อนย้ายวงแหวนตามเงื่อนไขน้อยที่สุด ซึ่งสำหรับวงแหวนจำนวน r วงแหวน จำนวนการเคลื่อนย้ายวงแหวนตามเงื่อนไขน้อยที่สุด ซึ่งสำหรับวงแหวนจำนวน r วงแหวน จำนวนการเคลื่อนย้ายที่น้อยที่สุดเท่ากับ $2^r - 1$ ครั้ง วิธีการแก้ปัญหา THP โดยวิธีรีเคอร์ซีฟ มีอัลกอริทึม ดังรูปที่ 3.3 จีเนติกอัลกอริทึมเป็นอีกวิธีการหนึ่งที่นำเสนอเพื่อใช้ทดสอบในการแก้ปัญหานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ภาพปัญหาทาว์นเวอร์ออฟฮานอย

```
Tower(n, source, target)
Begin
    If n = 1 then
        Move(n,source,taget)
    Else
        Tower(n-1,source,spare)
        Move(n,source,taget)
        Tower(n-1,spare,source)
End;
```

รูปที่ 3.3 รีเคอร์ซีฟอัลกอริทึมของทาว์นเวอร์ออฟฮานอย

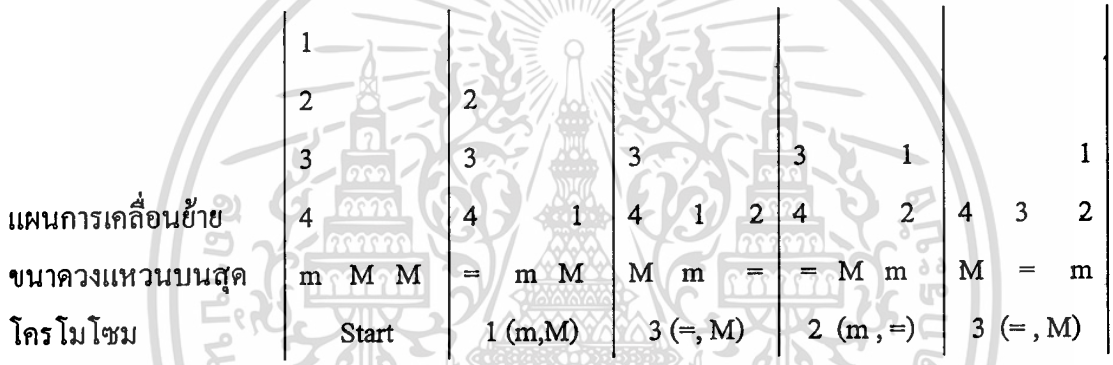
3.2.1 โครโมโซมของ THP

ลักษณะโครโมโซมของ THP ประกอบด้วยบิทที่แสดงถึงวิธีการเคลื่อนย้ายวงแหวนแต่ละครั้งดังแสดงในตารางที่ 3.2 โดยพิจารณาจากขนาดของวงแหวนบนสุดของแต่ละแกนคือ ขนาดใหญ่ (max : M) ขนาดกลาง (median : =) และขนาดเล็ก (min : m) หากแกนใดไม่มีวงแหวนเลยให้ถือเป็นแกนที่มีวงแหวนขนาดใหญ่ที่สุด ซึ่งแสดงตัวอย่างการเคลื่อนย้ายวงแหวนจำนวน 4 วงแหวนของโครโมโซม 1323 โดยใช้สัญลักษณ์ตัวเลข 1 ถึง 4 แทนขนาด ของวงแหวน ดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าบิท	(From, To)	ความหมาย
1	(m, M)	ย้ายวงแหวนขนาดเล็กไปบนวงแหวนขนาดใหญ่
2	(m, =)	ย้ายวงแหวนขนาดเล็กไปบนวงแหวนขนาดกลาง
3	(=., M)	ย้ายวงแหวนขนาดกลางไปบนวงแหวนขนาดใหญ่

ตารางที่ 3.2 การกำหนดค่าบิทของโครโมโซม THP



รูป 3.4 การเคลื่อนย้ายวงแหวนของโครโมโซม THP (1323)

ดังนั้นจำนวนบิทของโครโมโซม ของ THP เท่ากับ 2' – 1 และลักษณะโครโมโซมของ THP คือ

Step1	Step2	Step3		Step 2' - 1
3	2	3	...	1

ซึ่งแสดงในรูปของสัญลักษณ์ได้ดังนี้

$B_1B_2\,B_3\,....\,B_{2'-1}$ เมื่อ $B_1 \in I\,[1,3]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ฟังก์ชันเป้าหมายของ THP

ฟังก์ชันเป้าหมายของ THP เป็นฟังก์ชันวัดค่าของแผนการโดยพิจารณาผลลัพธ์จำนวนวงแหวนและขนาดของวงแหวนที่อยู่ในแกนเป้าหมาย รวมถึงพิจารณารูปแบบหลังการเคลื่อนย้ายแต่ละครั้งไม่ควรจะซ้ำกัน ดังสมการที่ 3.2 เป็นฟังก์ชนเป้าหมายซึ่งเจาะจงแกนเป้าหมายที่ต้องการคือแกน p

$$f = 1 + r_p + \dim(p) + \sum_{s=1}^{2^{r-1}} notred(\sum_{i=1}^{s} dup(patt[s], patt[i])) \qquad (3.2)$$

โดยที่ p = แกนเป้าหมายที่ต้องการ

r = จำนวนวงแหวนทั้งหมด

r_p = จำนวนวงแหวนที่อยู่ในแกน p

$\dim(x) = \sum_{i=1}^{r} i * d_i$ ซึ่ง d_i = ขนาดของวงแหวนลำดับที่ I ในแกน x

s = ลำดับที่เคลื่อนย้าย

Patt(x) = รูปแบบหลังการเคลื่อนย้ายลำดับที่ x

dup(Patt[x1], Patt[x2]) = 1 ถ้า Patt[x1] = Patt[x2]

0 ถ้า Patt[x1] ≠ Patt[x2]

ดังนั้นคำตอบที่ดีที่สุดของ THP คือ แผนการเคลื่อนย้ายวงแหวนที่มีค่าฟังก์ชันเป้าหมายสูงสุด MAX (f)

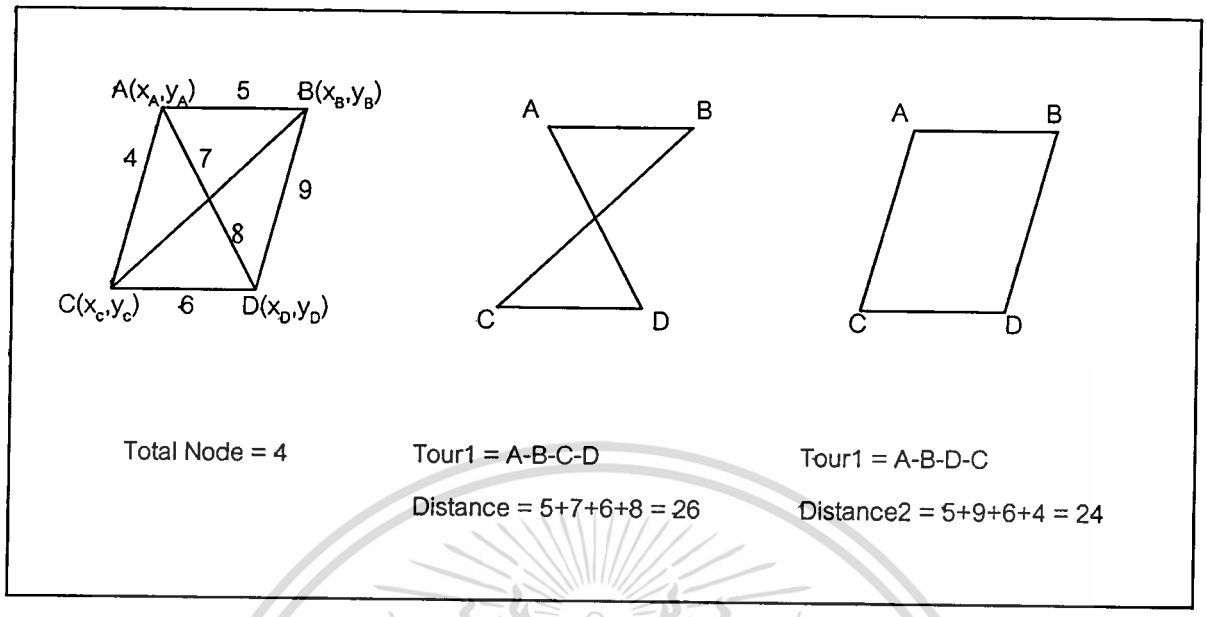
3.2.3 ตัวดำเนินการทางพันธุศาสตร์สำหรับ THP

ตัวดำเนินการทางพันธุศาสตร์สำหรับ THP ประกอบด้วย การครอสโอเวอร์แบบหลายจุด การมิวเตชันแบบตัวเลข และการอินเวอร์ชัน ตามที่กล่าวมาแล้ว

3.3 ปัญหาการหาเส้นทางที่ระยะทางสั้นที่สุด (TSP)

ปัญหาเส้นทางที่สั้นที่สุด (Traveling Saleman Problem : TSP) เป็นปัญหาในการหาเส้นทางการเดินทาง (tour) ที่ผ่านจุด (node) ทุกจุดและผ่านจุดละหนึ่งครั้ง โดยที่ระยะทางรวมของการเดินทาง (distance) ที่สั้นที่สุด ตัวอย่างแสดงดังรูปที่ 3.6 ซึ่ง TSP เป็นปัญหาพื้นฐานของการจัดตาราง (Schedule) เบื้องต้น ที่สามารถประยุกต์ใช้จีเนติกอัลกอริทึมหาคำตอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ภาพปัญหาการหาเส้นทางที่สั้นที่สุด

3.3.1 โครโมโซมของ TSP

ลักษณะโครโมโซมของ TSP คือเส้นทางของการเดินทางที่เกิดจากการจัดลำดับจุดต่าง ๆ ในการเดินทาง คือ

Tour Permutation of ($node_1$, $node_2$, ..., $node_n$)

ดังนั้นโครโมโซมของ TSP จึงประกอบด้วยบิทที่แสดงถึงจุดต่าง ๆ ในการเดินทางซึ่งแต่ละบิทมีความสัมพันธ์เป็นแบบลำดับ (order) นั่นหมายถึงถ้ามีจำนวนจุดทั้งหมด n จุด (เช่น $node_1$= A , $node_2$= B,..) การกำหนดค่าบิทแสดงดังตารางที่ 3.3

ค่าบิท	จุด
1	$node_1$(A)
2	$node_2$(B)
3	$node_3$(C)
4	$node_4$(D)
:	:
n	$node_n$(N)

ตารางที่3.3 การกำหนดค่าบิทของโครโมโซม TSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น จำนวน บิทของโครโมโซม TSP เท่ากับ n และลักษณะของโครโมโซมของ TSP คือ

Order 1	Order 2	Order 3		Order n
$Node_2$	$Node_{10}$	$Node_4$	...	$Node_7$

ซึ่งแสดงในรูปของสัญลักษณ์ได้ดังนี้

$B_1 B_2 B_3 .... B_n$ ซึ่ง. $B_i \in I[1, n]$ และ $B_i \neq B_j$ ถ้า $i \neq j$

ตัวอย่างโครโมโซมของเส้นทางในรูปที่ 3.7 คือ $Tour_1$ = A-B-C-D จะมีโครโมโซมเป็น 1234 และ $Tour_2$ = A-B-D-C จะมีโครโมโซมเป็น 1243

3.3.2 ฟังก์ชัน เป้าหมายของ TSP

ฟังก์ชันเป้าหมายของ TSP เป็นฟังก์ชันหาค่าระยะทางรวมของเส้นทางการเดินทาง ดังสมการที่3.3

$$f = \sum_{i=1}^{n} d_{B_i, B_{i+1}} \tag{3.3}$$

โดยที่ n = จำนวน node ทั้งหมด

B_i = node ในลำดับที่ i ของเส้นทาง

$D_{Bi,Bi+1}$ = ระยะทางระหว่าง node ในลำดับที i และ i+1 (กำหนดให้ i+1 =1 ถ้า i = n) ซึ่ง

$X_{B1} - X_{.B1-1}$ คือคู่ลำดับของ node ในลำดับที่ i เท่ากับ

$$\sqrt{(x_{B_i} - x_{B_{i-1}})^2 + (y_{B_i} - y_{B_{i-1}})^2}$$

ดังนั้นคำตอบที่ดีที่สุดของ TSP คือ เส้นทางที่ระยะทางรวมสั้นที่สุดหรือเส้นทางที่มีค่าฟังก์ชันเป้าหมายต่ำสุด MIN(f)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Delphi และ คอมโพเนนต์จีเนติกอัลกอริทึม

4.1 การพัฒนาโปรแกรมด้วย Delphi

Delphi เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมบน Windows โดยใช้ภาษาปาสคาล (Pascal) เป็นหลักในการพัฒนาโปรแกรม ซึ่งเป็นโครงสร้างภาษาที่เขียนง่าย และถึงแม้ว่าผู้ใช้จะไม่มีความรู้เกี่ยวกับภาษาปาสคาลเลยก็ไม่ได้เป็นอุปสรรคต่อการพัฒนาโปรแกรมแต่อย่างใด เนื่องจาก Delphi มีเครื่องมือช่วยเหลือในการนำคำสั่งต่าง ๆ มาใช้งานได้อย่างสะดวกและรวดเร็ว

Delphi เป็นผลิตภัณฑ์ของบริษัท Borland (ต่อมาได้เปลี่ยนชื่อเป็น Inprise โดยยังคงใช้ชื่อผลิตภัณฑ์ว่า Borland Delphi เช่นเดิม) ซึ่งประสบความสำเร็จจากการพัฒนา Turbo Pascal ที่มีชื่อเสียงโด่งดังบน DOS จน กระทั่งมาเป็น Delphi

Delphi เป็นเครื่องมือสำหรับการพัฒนาโปรแกรม ที่มีสิ่งอำนวยความสะดวกไว้อย่างครบถ้วน โดยมีสภาพแวดล้อมในการทำงาน (Development Environment) ที่ช่วยให้สามารถทำทุกอย่างได้จากใน Delphi เอง มีเครื่องมือทุกชนิดที่จำเป็นสำหรับการสร้างแอพพลิเคชันบน Windows ทั้งในส่วนของการติดต่อกับผู้ใช้ การแสดงผลกราฟิก การติดต่อกับฐานข้อมูล การจัดการระบบ ตลอดจนการพัฒนาแอพพลิเคชันเพื่อทำงานบนอินเตอร์เน็ต และด้วยคุณสมบัติที่มีอยู่มากมายของ Delphi พอจะแยกเป็นข้อๆ ดังนี้

4.1.1 วิชวลโปรแกรมมิ่ง (Visual Programming)

การพัฒนาโปรแกรมแบบวิชวล คือการพัฒนาโดยเห็นผลที่จะเกิดขึ้นเมื่อรันโปรแกรมได้ตั้งแต่ในขณะที่กำลังสร้าง โดยการนำชิ้นส่วนต่าง ๆ ที่ต้องการ ได้แก่ ปุ่ม (Button), ข้อความ(Label), รูปภาพ(Image) ฯลฯ ซึ่งสิ่งเหล่านี้เรียกโดยรวม ๆ ว่า คอมโพเนนต์ (Component) นำมาวางบนวินโดว์ที่เรียกว่า ฟอร์ม (Form) ปรับขนาดและตำแหน่งรวมทั้งคุณสมบัติต่าง ๆ ของคอมโพเนนต์ และแม้แต่ฟอร์มเองให้ได้ผลตามที่ต้องการ โดยการเปลี่ยนคุณสมบัติเหล่านี้ จะมีผลตั้งแต่ในขณะที่กำลังออกแบบ และเมื่อรันโปรแกรมก็จะได้ผลลัพธ์เหมือนกับที่เห็นในขณะออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การโปรแกรมเชิงวัตถุ (Object Oriented Programming – OOP)

การโปรแกรมเชิงวัตถุเป็นการพัฒนาโปรแกรมโดยการสร้าง วัตถุ หรือ ออบเจ็ค (Object) ที่ต้องการในมุมมองของตัววัตถุเองว่าต้องการให้มีลักษณะเป็นอย่างไร และสามารถทำอะไรได้บ้าง แทนที่จะมองที่การสร้างรูทีน (Routine) หรือโพรซีเยอร์ (Procedure) เป็นหลักเช่นดังก่อนนี้ ประโยชน์ที่ได้ก็คือ เราสามารถสร้างวัตถุโดยเริ่มจากวัตถุที่ง่าย ๆ ไม่ซับซ้อนเป็นพื้นฐานขึ้นมาเสียก่อน จากนั้นจึงนำวัตถุเหล่านั้นมาตกแต่งปรับปรุงให้มีความสามารถมากขึ้น หรือทำงานได้หลากหลายขึ้น โดยนำสิ่งที่เหมือนกันหรือใช้ร่วมกันมาไว้ในวัตถุ ซึ่งจะเรียกว่า Base Object หรือ Base Class จากนั้นจึงแต่งเติม Base Object นี้ให้กลายเป็นออบเจ็คอื่น ๆ ตามที่ต้องการ เราสามารถกำหนดวัตถุที่ได้นี้กลับมาใช้ใหม่ได้เรื่อย ๆ ในแอพพลิเคชั่นต่าง ๆ และแต่งเติมต่อไปได้เรื่อย ๆ เช่นกัน

4.1.2.1 โครงสร้างของออบเจ็ค

ชนิดของออบเจ็ค	ออบเจ็คแต่ละออบเจ็คจะถือว่าเป็นคนละชนิด (Type) กัน เมื่อนำออบเจ็คไปสร้างต่อให้เป็นออบเจ็คใหม่ ก็จะต้องเป็นชนิดใหม่เสมอ หรือเรียกได้ว่าเป็นคนละคลาส (Class) กันนั่นเอง ชนิดของออกเจ็ค ได้แก่ ออกเจ็คชนิดปุ่ม และชนิดข้อความ เป็นต้น
คุณสมบัติ	หรือเรียกว่า "พร็อพเพอร์ตี้" (Property) หมายถึงคุณลักษณะของออบเจ็คแต่ละตัวที่สามารถกำหนดให้แตกต่างกันไปตามความต้องการที่ต่างกัน เช่น ขนาดและสีของปุ่ม หรือข้อความของออบเจ็คที่แสดงอยู่บนปุ่ม เป็นต้น
พฤติกรรม	หรือ เมธอด (Method) คือความสามารถในการทำงานของออบเจ็ค ตัวอย่างเช่นการแสดงปุ่ม (Show) หรือการซ่อนปุ่ม (Hide) เป็นต้น

4.1.2.2 หลักการของ OOP

Encapsulation	เป็นการซ่อนเร้นส่วนการทำงานภายในออบเจ็ค ที่ไม่เกี่ยวข้องกับภายนอกไว้ ไม่ให้เห็นและไม่ให้แก้ไขเปลี่ยนแปลงส่วนที่ซ่อนไว้นี้ ซึ่งเราจะนำออบเจ็คไปใช้หรือไปดัดแปลงได้เฉพาะส่วนที่ออบเจ็คนั้นยอมให้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Inheritance	เป็นการสืบทอดคุณสมบัติของออบเจ็ค เมื่อนำออบเจ็คใด ๆไปสร้างเป็นออบเจ็คใหม่ คุณสมบัติของออบเจ็คเดิมก็จะยังคงมีอยู่และสามารถเรียกใช้และทำงานได้อย่างครบถ้วน
Polymorphism	คือลักษณะของการทำงานที่แตกต่างกันของคุณสมบัติหรือพฤติกรรมเดียวกัน แต่เป็นของออบเจ็คคนละชนิดกัน ตัวอย่างเช่น เมธอด Save To File ของเมโมจะได้ text file ที่เก็บข้อความนั้น ในขณะที่ Save To File ของอิมเมจจะได้ไฟล์รูปภาพ ซึ่งแม้จะเป็นการบันทึกข้อมูลไว้เป็นไพล์เหมือนกัน แต่วิธีการบันทึกจะต่างกัน

4.1.3 คอมไพเลอร์อย่างแท้จริง

Delphi เป็นคอมไพเลอร์ที่ใช้แปลภาษาโปรแกรมเป็นภาษาเครื่อง ซึ่งหลังจากการคอมไพล์โปรแกรมเราจะได้ไฟล์ .EXE ซึ่งเก็บภาษาของเครื่องนั้น ๆ และสามารถทำงานได้เลยโดยไม่ต้องผ่านขั้นตอนการแปลภาษาในระหว่างการทำงานอีก ทำให้สามารถทำงานได้เร็วและไม่มีขีดจำกัด

4.1.4 คอมโพเนนต์ไลบรารี (Component Library)

คอมโพเนนต์เป็นส่วนประกอบย่อย ๆ ที่เราจะนำมาใช้ในการสร้างแอพพลิเคชัน ซึ่งใน Delphi มีคอมโพเนนต์ให้เลือกใช้อยู่เป็นจำนวนมาก โดยจะเก็บอยู่ใน "ห้องสมุด" หรือ คอมโพเนนต์ไลบรารี (component library) และจัดแยกเป็นกลุ่ม ๆ ตามประเภทของการใช้งาน

4.1.5 OCX และ ActiveX

OCX คือออบเจ็คที่ใช้ใน Visual Basic ของไมโครซอฟท์ ซึ่งมีลักษณะคล้ายกับคอมโพเนนต์ของ Delphi ส่วน ActiveX ก็เป็นออบเจ็คที่ถูกพัฒนาในรูปแบบเดียวกับ OCX แต่มีวัตถุประสงค์เพื่อใช้งานบนอินเตอร์เน็ต ซึ่งทั้ง OCX และ ActiveX ต่างก็เป็นออบเจ็คที่นำมาใช้กับ Delphi ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 Wizard และ Object Repository

Delphi มี Wizard ที่ใช้ในการสร้างฟอร์มและออบเจ็คที่ต้องใช้บ่อย ๆ หรือสร้างได้ยาก ไว้ให้ใช้เป็นจำนวนมาก ตัวอย่างเช่น Wizard ในการสร้างรายงานแบบต่าง ๆ หรือฟอร์ม เช่น About Box และ Dialog Box ซึ่งจะเรียกอีกอย่างหนึ่งว่า เทมเพลต (Template) ก็ได้ เหล่านี้ล้วนช่วยให้การสร้างแอพพลิเคชันเป็นไปอย่างรวดเร็วและมีความกลมกลืนมากขึ้น นอกจากนี้ยังสามารถเก็บฟอร์มหรือออบเจ็คที่คิดว่าต้องใช้บ่อย ๆ เพิ่มเข้าไปไว้ให้เรียกใช้ในแอพพลิเคชันอื่น ๆ ได้ โดยการบันทึกไว้ใน Object Repository จากนั้นเมื่อต้องการใช้งานก็เรียกใช้ได้เหมือนกับการใช้ Wizard ทั่ว ๆ ไป

4.1.7 การติดต่อกับฐานข้อมูล

การติดต่อกับฐานข้อมูลนั้น Delphi มีคอมโพเนนต์ที่สามารถเชื่อต่อ เพื่อจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลทุกประเภท ไม่ว่าจะเป็นการเพิ่ม ลบ แก้ไข หรือการเรียกดูข้อมูล โดยผู้ใช้ไม่ต้องเขียนชุดคำสั่งใด ๆ ในโปรแกรมเลยก็สามารถสร้างแอพพลิเคชันได้อย่างง่าย ๆ ที่ทำงานกับฐานข้อมูลขึ้นมาได้แล้ว

เราสามารถใช้ Delphi จัดการกับฐานข้อมูลที่เป็นแบบง่าย ๆ ซึ่งได้แก่ dBase หรือ Paradox และ MS Access ซึ่งพวกนี้จะเรียกว่าเป็น Local Database คือ Database ที่ทำงานในเครื่องนั้นเอง หรือจะเรียกอีกอย่างหนึ่งว่า เป็นแบบ File_oriented Database ก็ได้ เพราะเป็นการเก็บข้อมูลไว้ในโครงสร้างของไพล์ นอกจากนี้ยังสามารถใช้งานกับระบบฐานข้อมูลที่เป็น Database Server ต่าง ๆ เช่น SQL Server หรือ InterBase ซึ่งอาจจะทำงานอยู่ในเครื่องเดียวกันกับแอพพลิเคชันในกรณีของ Local InterBase หรืออาจจะทำงานอยู่บนเครื่อง Server เครื่องอื่น ซึ่งเรียกว่าเป็นการใช้ฐานข้อมูลในแบบ Client/Server

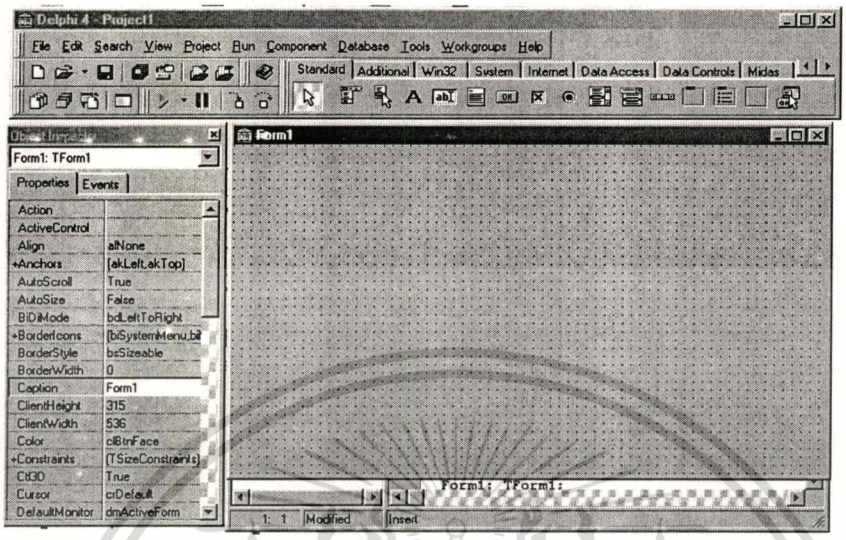
4.2 ส่วนประกอบของ Delphi

ขั้นตอนกาเรียกใช้โปรแกรม Delphi4 สามารถเลือกได้จาก Start > program > Borland Delphi4 และเลือกไอคอนรูปที่ 4.1

รูปที่ 4.1 ไอคอนโปรแกรม Delphi 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

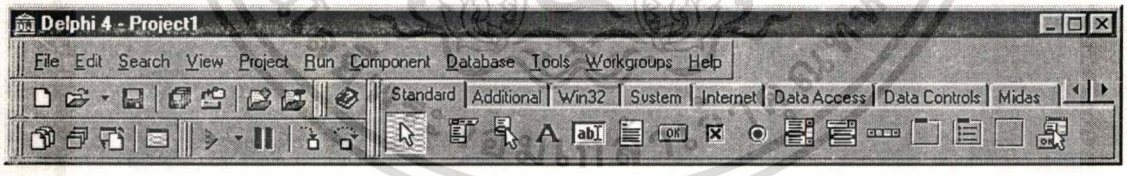
หน้าตาของ Delphi จะประกอบไปด้วย Window ต่าง ๆ ดังนี้



รูปที่ 4.2 Windows ต่าง ๆ ของโปรแกรม Delphi

4.2.1 วินโดว์หลัก (Main Windows)

วินโดว์หลักทำหน้าที่เป็นตัวกลางสำหรับควบคุมกระบวนการในการพัฒนาโปรแกรม ไม่ว่าจะเป็นการจัดการเกี่ยวกับไฟล์ภายในแอพพลิเคชัน การคอมไพล์ และการตรวจสอบข้อผิดพลาดเมื่อรันโปรแกรม Delphi โดยจะอยู่บริเวณด้านบน ภายในวินโดว์หลักจะแบ่งออกเป็น 3 ส่วนคือ



รูปที่ 4.3 วินโดว์หลัก

4.2.1.1 เมนู (Menubar)

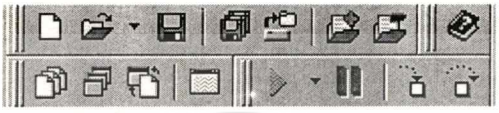
เป็นส่วนหลักที่ใช้ในการทำงานคำสั่งต่ง ๆ เกือบทั้งหมดสามารถเรียกใช้ผ่านเมนูได้โดยลักษณะดังรูป

รูปที่ 4.4 เมนูบาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 Speedbar

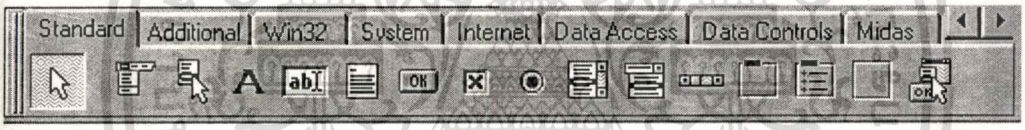
Speedbar ประกอบไปด้วยปุ่มของคำสั่งที่ใช้งานบ่อย ๆ สำหรับการทำงานของ Delphi สามารถใช้การ Click Mouse ที่ปุ่ม เพื่อเรียกคำสั่งเหล่านั้นมาทำงานโดยปกติ Speedbar จะอยู่ใต้เมนู และอยู่ทางด้านซ้ายของจอ โดยมีลักษณะดังรูป



รูปที่ 4.5 Speedbar

4.2.1.3 Component Palette

ภายใน Component Palette จะมีรูปเล็ก ๆ เรียกว่า ไอคอน ที่แสดงถึงคอมโพเนนต์ต่าง ๆ ใน Delphi โดยจะจัดแบ่งออกเป็นหลายกลุ่ม วางอยู่ในแท็ปต่าง ๆ ของ Component Palette



รูปที่ 4.6 Component Palette

4.2.2 หน้าต่าง Object Inspector

Object Inspector เป็นที่สำหรับกำหนดคุณสมบัติและโปรซีเดอร์ที่ควบคุม Event ของ Component ในฟอร์มด้านบนของ Object Inspector จะเป็นคอมโบบ๊อกที่แสดงชื่อของคอมโพเนนต์และชนิดของคอมโพเนนต์ ภายใน Object Inspector จะประกอบไปด้วยแท็ปสองแท็ป คือ

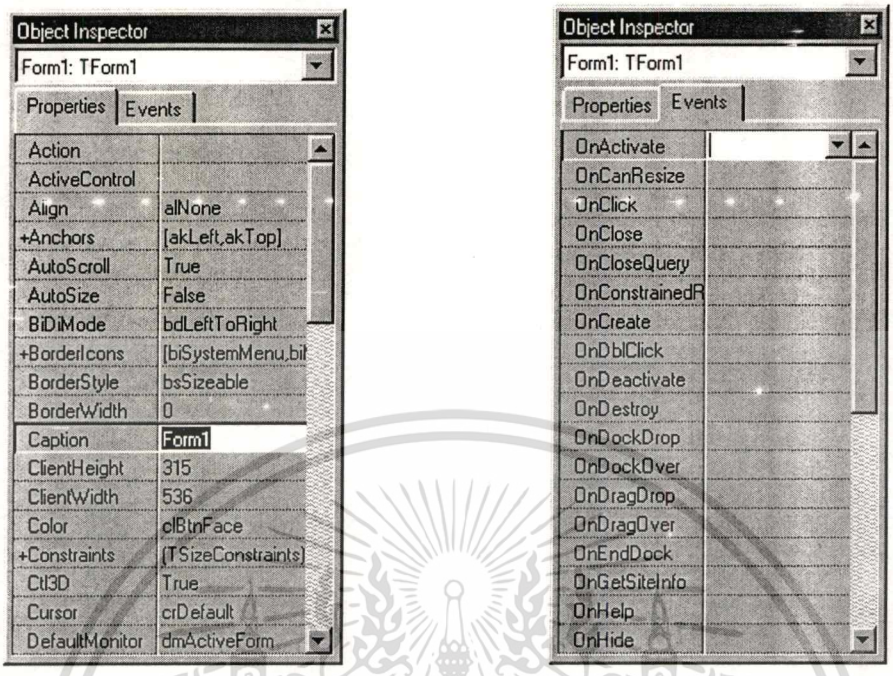
4.2.2.1 แท็ป Properties

จะแสดงคุณสมบัติ (Properties) ต่าง ๆ ของ Component ภายในแท็ป Properties จะประกอบไปด้วยสองคอลัมน์ โดยที่ทางด้านซ้ายจะเป็นชื่อของคุณสมบัติ และทางด้านขวาจะเป็นค่าของคุณสมบัตินั้น ๆ

4.2.2.2 แท็ป Events

ใช้กำหนดคำสั่งต่าง ๆ ที่จะกระทำก็ต่อเมื่อเกิดเหตุการณ์หนึ่งขึ้นกับคอมโพเนนต์ เช่นในการ คลิกเมาท์ หรือกดปุ่ม <Enter >

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 หน้าต่าง Object Inspector

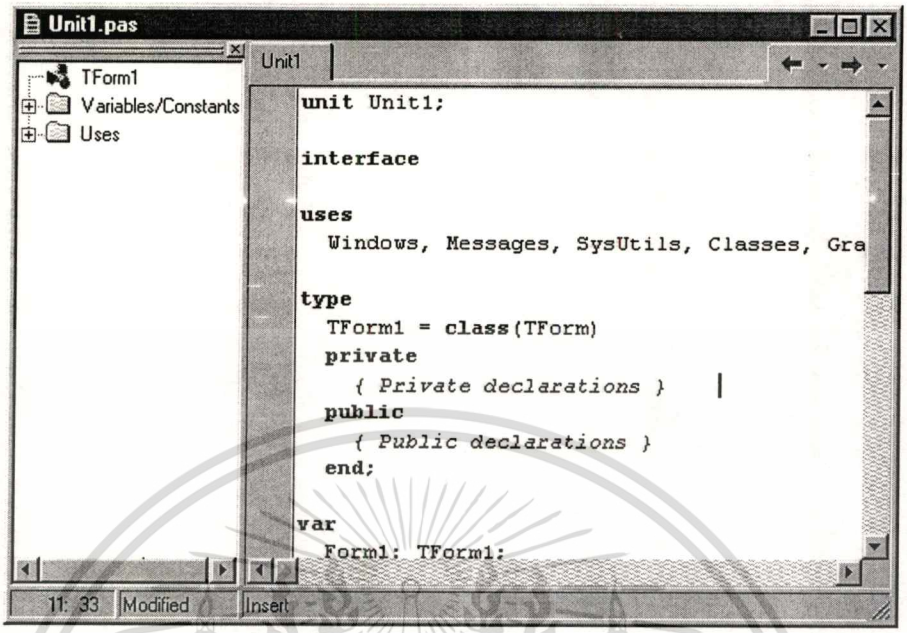
4.2.3 หน้าต่าง Form Designer

Form Designer ใช้สำหรับออกแบบส่วนที่ติดต่อกับผู้ใช้ ซึ่งสามารถนำคอมโพเนนต์ต่าง ๆ จาก Component Palette มาวางลงบน Form Designer ได้

4.2.4 หน้าต่าง Code Editor

Code Editor เป็น Text Editor สำหรับการเขียนโปรแกรมใน Delphi ซึ่งประกอบไปด้วยคุณสมบัติต่าง ๆ ที่อำนวยความสะดวกในการเขียนโปรแกรม Delphi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าต่าง Code Editor

Code Editor จะประกอบไปด้วย

1. จุดบอกตำแหน่งของเคอร์เซอร์ โดยจะบอกว่าตอนนี้เคอร์เซอร์อยู่ที่บรรทัดใด และคอลัมน์ใด

2. สถานะการแก้ไข สามารถเช็คได้ว่ามีการแก้ไขแล้วหรือยัง ในกรณีที่มีการแก้ไขเกิดขึ้นจะขึ้นคำว่า "Modified"

3. สถานะการพิมพ์ เป็นตัวบอกสถานะการพิมพ์ว่า เป็นการพิมพ์แทรก (Insert) หรือการพิมพ์ทับ (Overwrite)

4. Code Explorer มีลักษณะเป็น Tree เพื่อช่วยในการเข้าถึง Object ต่าง ๆ ภายใน Unit

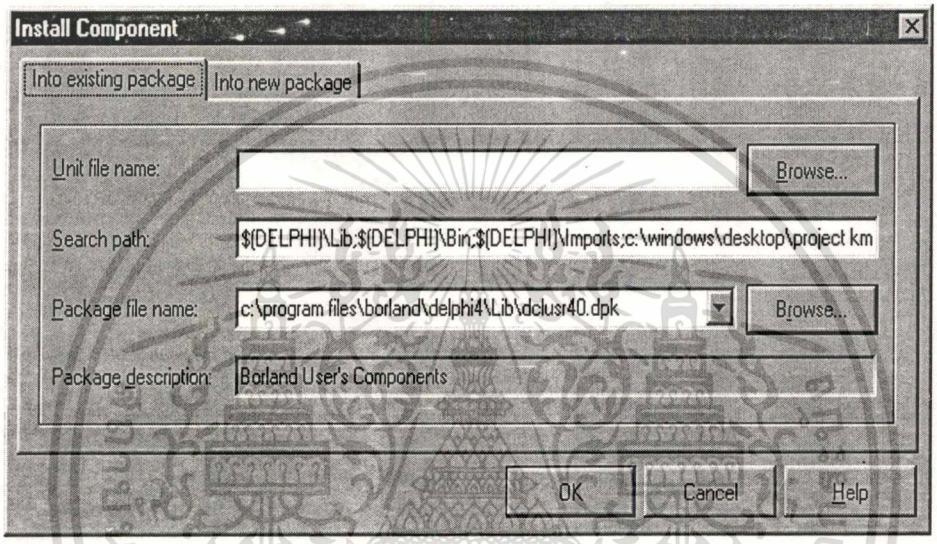
4.3 ขั้นตอนการติดตั้ง คอมโพเนนต์จีเนติก

การติดตั้งคอมโพเนนต์จีเนติก นั้นจะถูกเก็บอยู่ภายใต้ชุดแพ็กเก็ตดังรูปที่ 4.9

รูปที่ 4.9 ไอคอน แพกเก็ตสำหรับติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Delphi เวอร์ชันแรก ๆ นั้น การติดตั้งคอมโพเนนต์ลงบนคอมโพเนนต์พาเล็ต จะใช้วิธีคอมไพล์ลงในคอมโพเนนต์ไลบรารี แต่ใน Delphi ตั้งแต่เวอร์ชัน 3 เป็นต้นมา จะเก็บคอมโพเนนต์ไว้ในแพ็กเกจ (package) หลังจากนั้นจึงติดตั้งแพ็กเกจลงในคอมโพเนนต์พาเล็ต ในขณะที่สร้างคอมโพเนนต์โดยใช้ Component Expert นั้น สามารถติดตั้งคอมโพเนนต์ได้เลยโดยคลิกที่ปุ่ม install ซึ่งจะปรากฏไดอะล็อกบ็อกซ์สำหรับติดตั้งคอมโพเนนต์ดังรูปที่ 4.10

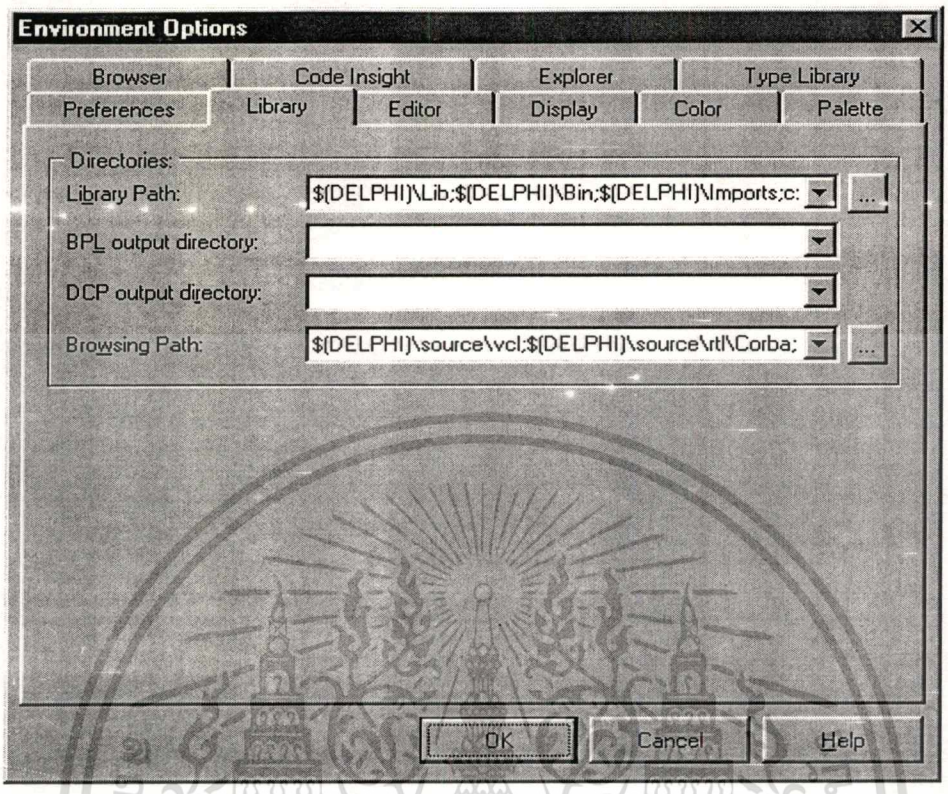


รูปที่ 4.10 ไดอะล็อกบ็อกซ์สำหรับติดตั้งคอมโพเนนต์

ภายในไดอะล็อกบ็อกซ์ Install Component จะแบ่งเป็น 2 แท็บคือ Into existing package สำหรับเก็บคอมโพเนนต์แพ็กเกจที่มีอยู่แล้ว และ Into new package สำหรับสร้างแพ็กเกจใหม่เพื่อใช้เก็บคอมโพเนนต์

การติดตั้งคอมโพเนนต์ Delphi จะสร้างไฟล์ .bpl ให้ ซึ่งก่อนที่จะติดตั้งคอมโพเนนต์เราจะต้องกำหนดไดเร็คทอรีที่จะเก็บไพล์นี้ก่อน ในกรณีที่ไม่ได้กำหนดก็จะเป็นการใช้ค่าดีฟอลต์ที่ Delphi กำหนดไว้ตั้งแต่ตอนติดตั้งโปรแกรม วิธีการแก้ไขจะต้องระบุในแท็บ Library ของไดอะล็อกบ็อกซ์ Environment Options ดังรูป 4.11

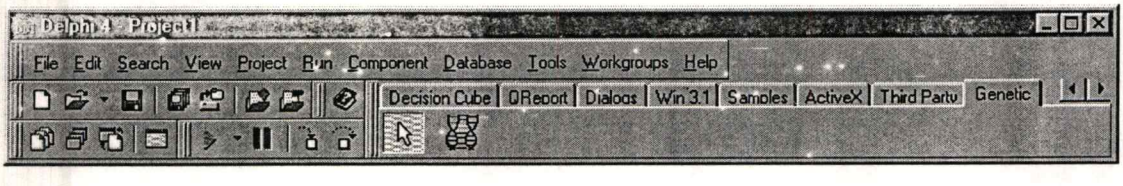
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การเซ็ทค่าไดเร็คทอรีที่ทำการจัดเก็บไฟล์

ในกรณีที่ไม่ได้ติดตั้งคอมโพเนนต์ไว้ตั้งแต่ตอนที่สร้างคอมโพเนนต์ สามารถติดตั้งในภายหลังได้โดยใช้คำสั่งในเมนู Component > Install Component ซึ่งจะปรากฏไดอะล็อกบ็อกซ์ Install Component เช่นกัน

เมื่อทำการติดตั้ง คอมโพเนนต์เรียบร้อยแล้ว จะปรากฏแท็ป Genetic ในคอมโพเนนต์พาเล็ตเพิ่มเติมขึ้นมาดังรูปที่ 4.12

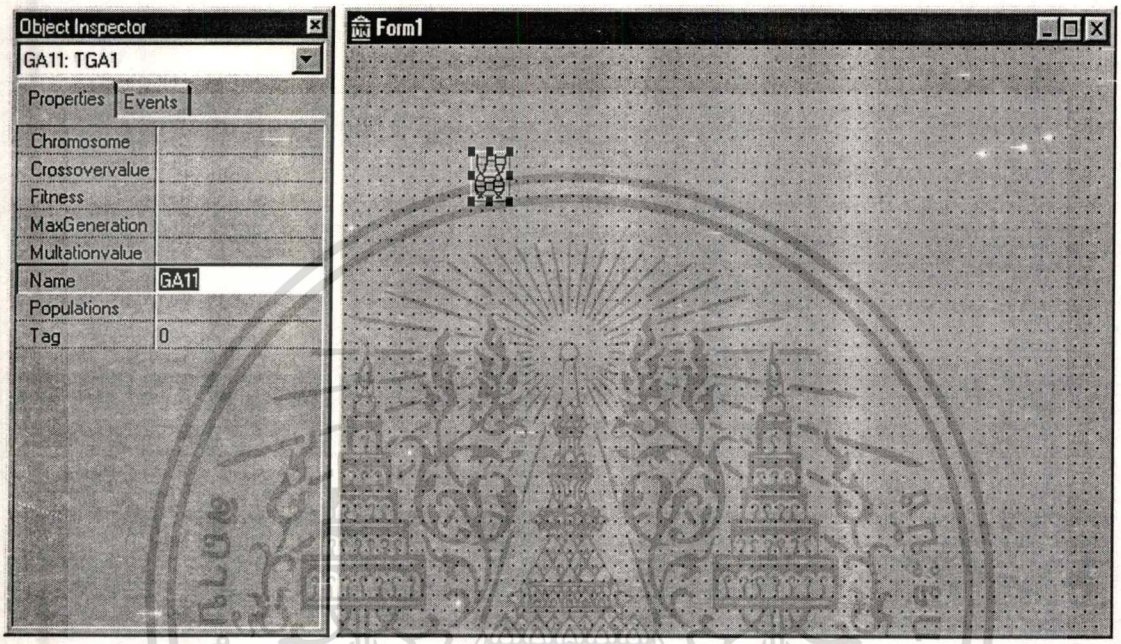


รูปที่ 4.12 คอมโพเนนต์ Genetic ปรากฏใน คอมโพเนนต์พาเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเรียกใช้งาน คอมโพเนนต์ จีเนติกอัลกอริทึม

การใช้งานคอมโพเนนต์ จีเนติกอัลกอริทึก จะเหมือนกับการใช้งาน คอมโพเนนต์อื่น ๆ ของ Delphi โดยสามารถ นำมาวางไว้บน ฟอร์ม และกำหนดค่า properties .ให้กับ คอมโพเนนต์



รูปที่ 4.13 Properties ของ คอมโพเนนต์จีเนติกอัลกอริทึม

4.5 โปรแกรม Demo ทดสอบการทำงานของ คอมโพเนนต์จีเนติกอัลกอริทึม

โปรแกรม Demo จะประกอบไปด้วย 3 โปรแกรมหลัก ซึ่งสร้างจากคอมโพเนนต์ จีเนติก อัลกอริทึก คือ โปรแกรมจีเนติกอัลกอริทึกสำหรับทำงานบน PC ,โปรแกรมจีเนติกอัลกอริทึก สำหรับแสดงผลผ่านบราวเซอร์ (Browser) และโปรแกรมจีเนติก สำหรับการแก้ปัญหา Traveling Saleman Problem (TSP) ซึ่งทั้ง 3 โปรแกรมจะแสดงให้เห็นถึงประโยชน์ของ การทำจีเนติกอัลกอริทึมในรูปแบบของคอมโพเนนต์ไปใช้งาน

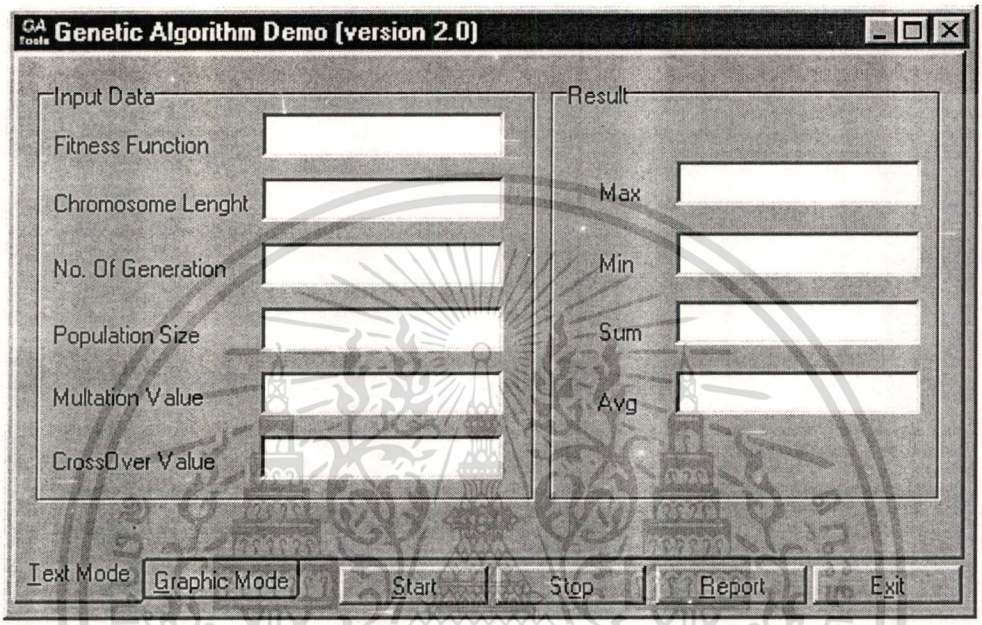
4.5.1 โปรแกรมจีเนติกอัลกอริทึกสำหรับทำงานบน PC

ขั้นตอนการทำงานจะเริ่มจากการเปิดโปรแกรมขึ้นมาใช้งาน โดยคลิกที่ไอคอนรูปที่ 4.14

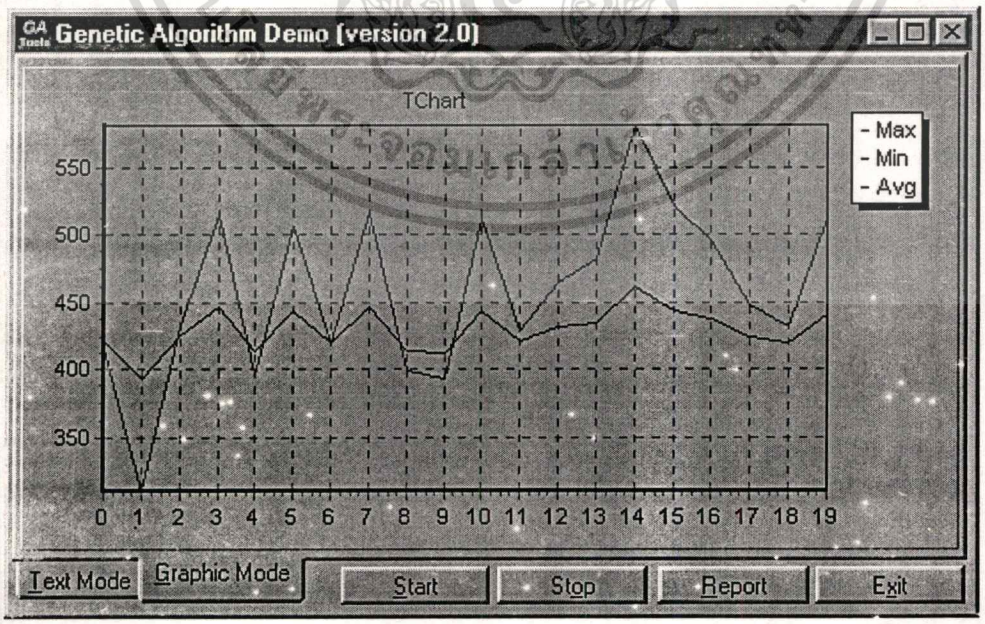
รูปที่ 4.14 ไอคอนโปรแกรมจีเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะใส่ค่าพารามิเตอร์ที่จำเป็นลงไปเพื่อให้ได้ผลลัพธ์ที่ต้องการ โดยโปรแกรมดังกล่าวจะแยกออกเป็น 2 ส่วน คือ ส่วนที่แสดงผลเป็น Text Mode และส่วนที่แสดงผลแบบ Graphic Mode



รูปที่ 4.15 โปรแกรมจีเนติกอัลกอริทึกในรูปแบบของ Text Mode

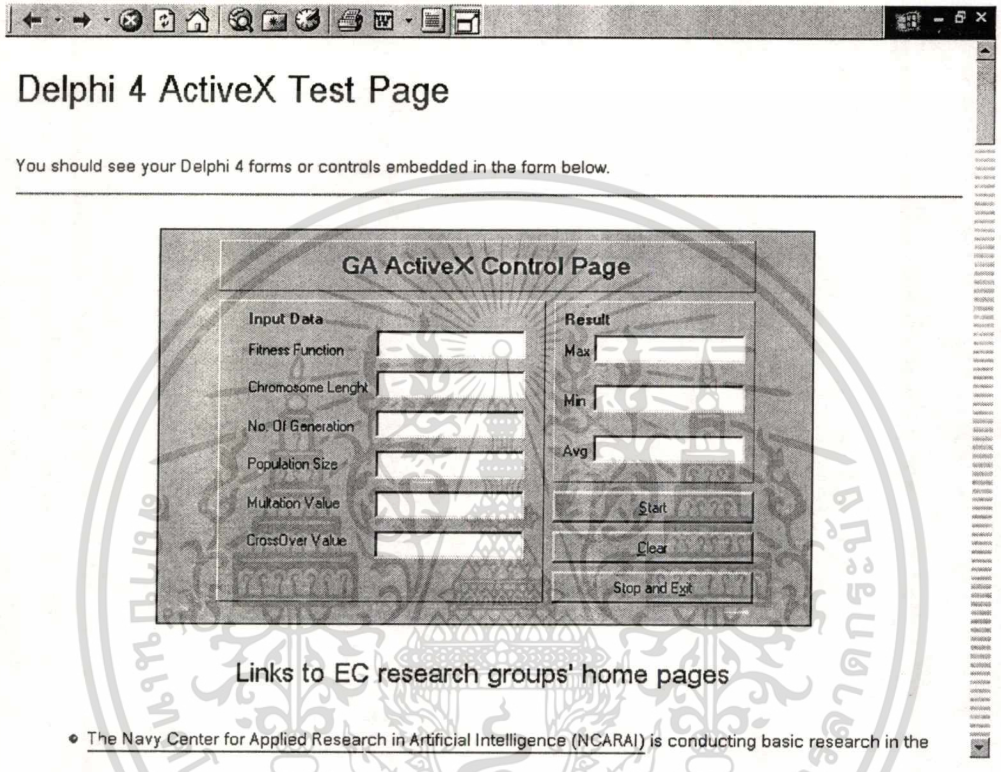


รูปที่ 4.16 โปรแกรมจีเนติกอัลกอริทึมในรูปแบบของ Graphic Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 โปรแกรมจีเนติกอัลกอริทึมสำหรับทำงานผ่านบราวเซอร์

โปรแกรมในส่วนที่ทำงานบนบราวเซอร์จะมีลักษณะคล้ายกับการทำงานบน PC รูปแบบการนำเสนอ จะอยู่ในรูปแบบของ การนำ Active X มาใช้งาน ซึ่งเป็นคุณสมบัติของ Delphi

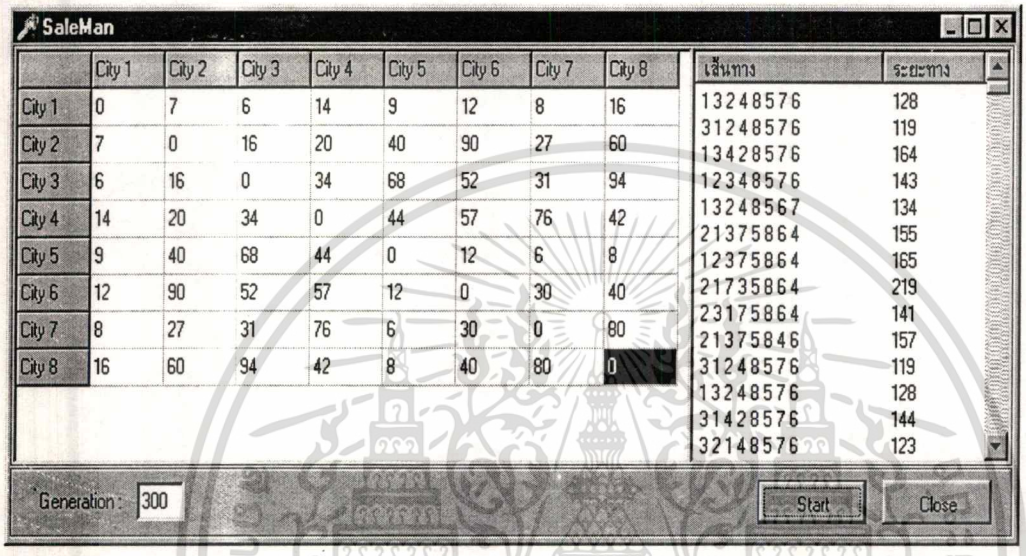


รูปที่ 4.17 โปรแกรมจีเนติกอัลกอริทึมบนบราวเซอร์ สร้างโดยใช้ ActiveX Form ใน Delphi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 โปรแกรมจีเนติกอัลกอริทึมสำหรับแก้ปัญหา Traveling Saleman Problem **(TSP)**

โปรแกรมสำหรับแก้ไขปัญหา Traveling Saleman Problem จะแสดงวิธีการแก้ไขปัญหาโดยจะเลือกเส้นทางที่สั้นที่สุดสำหรับการเดินทางไปยังเมือง ต่าง ๆ



4.6 สรุปผลการทำงาน

จากการเขียนโปรแกรม Delphi 4 ร่วมกับคอมโพเนนต์จีเนติกอัลกอริทึม ที่สร้างขึ้น สามารถลดขั้นตอนการเขียนโปรแกรมลงได้มากกว่า 50 เปอร์เซ็น ซึ่งนับว่าเป็นการประหยัดเวลาที่มีค่าในการพัฒนาโปรแกรมตามวัตถุประสงค์ของการพัฒนาคอมโพเนนต์ดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Davis, L. 1991. **Handbook of Genetic Algorithms,** New York,Van Nostrand Reinhold.

[2] Goldberg, D.E., 1989. **Genetic Algorithms in Search, Optimaization and Machine Learning,** Addison Wesley Publishing Company.

[3] Grefenstette, J.J., Baker, J.E., 1989. **How Genetic Algorithm Work : A Critical Look at Implicit Parallelism, in Proceedings of the Third International Conference on Genetic Algorithms,** Morgan Kaufmann Publishers, Los Altos, CA, pp. 20-26.

[4] Grefenstette, J.J., Gopal, R., and Gucht, D. V., 1985. **Genetic Algorithms for Traveling Salesman Problem, Proceedings of the First International Conference on Genetic Algorithms,** Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 160-169.

[5] Morrow, M., April 1991. **Genetic Algorithms,** Dr. Dobb's Journal, pp. 26-32.

[6] Koza, J. R., 1991. **Genetic Programming,** MIT Press, Cambridge, MA, pp. 94-97

[7] กาญจนี วงศ์วิภาพร. **การจัดตารางสอนของโรงเรียนแบบอัตโนมัติ โดยจีเนติกอัลกอริทึม.** วิทยานิพนธ์ วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า. บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541

[8] วีรศักดิ์ วัฒนายากร. **ระบบตรวจสอบลายเซ็นแบบออนไลน์โดยใช้จีเนติกอัลกอริทึม.** วิทยานิพนธ์เทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ. บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้