Mastermind

1st นางสาว จิราพร วังคำหาญ
65070501008
หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

2nd นางสาว ชนม์นิภา เทียมพันธุ์
65070501010
หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

3rd นาย ธนพล เหนือโท
65070501024
หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

4th นาย ธเนศ จอมพูล
65070501025
หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

5th นาย สิรวิชญ์ อาสานอก
65070501056
หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

I. บทนำ

เกมถอดรหัส Mastermind เป็นเกมที่ให้ผู้เล่นถอดรหัสเพื่อตรวจสอบว่า สามารถถอดรหัสได้เร็วเพียงใด โดยเกมนี้จะใช้ผู้เล่นสองคนคือ ผู้เล่นที่สร้างรหัส กับผู้เล่นที่ถอดรหัส และมีการแบ่งระดับความยากเป็น การถอดรหัสจากตัวเลข สีหมุด หรือสัญลักษณ์อื่น ๆ 4 ตำแหน่ง 6 สี กับ 5 ตำแหน่ง 8 สี วิธีการเล่น คือ เกมจะให้ผู้เล่นที่สร้างรหัสสร้างรหัสลับขึ้นมา จากนั้นผู้เล่นที่ถอดรหัสจะทำการสุ่มเลือกสีมา และให้ผู้เล่นที่สร้างรหัสจะตอบกลับด้วยการบอกจำนวนสี และตำแหน่งที่ถูกต้อง กับจำนวนสีที่ถูกต้องแต่ตำแหน่งไม่ถูกต้อง

ซึ่งคณะผู้จัดทำได้เล็งเห็นว่า เกมถอดรหัสสามารถประยุกต์ให้เข้ากับการเขียนโปรแกรมได้ โดยการสร้างชุดข้อมูลตัวเลข n ตัว $(1, 2, \ldots, n)$ และตำแหน่ง r ตำแแหน่ง $(1, 2, \ldots, r)$ ทั้งหมดจำนวน $C_{n,r}$ ข้อมูล จากนั้นให้โปรแกรมสุ่มรหัสขึ้นมา 1 ชุด และตั้งชุดข้อมูลการคาดเดาอันแรก 1 ชุด แล้วใช้ Genertic Algorithm ในการสร้างชุดข้อมูลการคาดเดาในรอบถัดไปที่มีความใกล้เคียง กับรหัสลับมากที่สุด โดยในแต่ละรอบของการคาดเดาจะผ่านการตรวจสอบความถูกต้องด้วยค่า X_i และ Y_i ซึ่งค่า X_i คือ จำนวนตัวเลขและตำแหน่งที่ถูกต้องของชุดข้อมูลการคาดเดาในรอบที่ i กับ ค่า Y_i คือ จำนวนตัวเลขที่ถูกต้องแต่ตำแหน่งไม่ถูกต้องของชุดข้อมูลการคาดเดาในรอบที่ i

ซึ่งทางคณะผู้จัดทำมีวัตถุประสงค์ในการสร้างเกมถอดรหัส Mastermind จากการเขียนโปรแกรม 3 แบบ แบบที่ 1 เขียนโปรแกรมภาษา C ผ่านเว็บไซต์ Replit โดยการเลือกใช้ Genertic Algorithm แบบที่ 2 เขียนโปรแกรม Wolfram ผ่านโปรแกรม Wolfram Mathematica โดยการเลือกใช้ Genertic Algorithm แบบที่ 3 เขียนโปรแกรมภาษา C ผ่านเว็บไซต์ Replit โดยการเลือกใช้ Brute force algorithm เพื่อใช้หาจำนวนการคาดเดาเฉลี่ยที่จำเป็นในการค้นหารหัสลับทั้งสองอัลกอริทึมโดยจะเฉลี่ยจากจำนวนเกมที่เล่นทั้งหมด 3 เกม เพื่อเปรียบเทียบอัลกอริทึมทั้งสองว่าวิธีการใดมีประสิทธิภาพในการคาดเดาคำตอบได้ดีที่สุด โดยจะมีการนำเสนอข้อมูลในรูปแบบตาราง

II. เอกสารและงานที่เกี่ยวข้อง

A. Genetic Algorithm

เป็นวิธีการแก้ปัญหาการปรับแต่งค่าแบบมีเงื่อนไขและไม่มีเงื่อนไข โดยอาศัยหลักการคัดเลือกตามธรรมชาติ ซึ่งเป็นกระบวนการวิวัฒนาการทางชีววิทยา Genetic Algorithm จะทำการปรับเปลี่ยนประชากรของชุดคำตอบแต่ละชุดซ้ำ ๆ ในแต่ละขั้นตอน โดยจะเลือกชุดคำตอบจากประชากรปัจจุบันให้เป็น parent เพื่อใช้ในการสร้างลูกหลานสำหรับรุ่นต่อไป ในทุกรุ่นประชากรจะพัฒนาไปสู่คำตอบที่เหมาะสม

B. Population Size

กลุ่มตัวเลือกคำตอบในแต่ละรุ่น สามารถเรียกอีกอย่างว่า ชุดโครโมโซม โดยต้องรักษาความหลากหลายของประชากร และขนาดประชากรไม่ควรมีขนาดใหญ่เกินไปเพราะจะทำให้ Genetic Algorithm ทำงานช้าลง ในขณะเดียวกัน ขนาดที่เล็กเกินไปอาจไม่เพียงพอสำหรับการผสมพันธุ์ที่ดี ดังนั้นจึงจำเป็นต้องกำหนดขนาดประชากรที่เหมาะสมโดยการลองผิดลองถูก

C. Parent Selection

เป็นกระบวนการคัดเลือก parent ที่ผสมพันธุ์และรวมตัวกันใหม่เพื่อสร้างลูกหลานให้กับรุ่นต่อไป การคัดเลือก parent มีความสำคัญอย่างยิ่งต่ออัตราการบรรจบกันของกระบวนการ Genetic Algorithm เนื่องจาก parent ที่ดีจะผลักดันไปสู่คำตอบที่เหมาะสม

D. Crossover Rate

เป็นโอกาสที่โครโมโซมสองตัวจะสลับตำแหน่งในโครโมโซม หรือ โอกาสในการเกิด Crossover โดยค่าที่ดี คือ ประมาณ 0.7

E. Mutation Rate

เป็นโอกาสที่ค่าใดค่าหนึ่งของบิตภายในโครโมโซมหนึ่งจะถูกสลับกับค่าของบิทภายในอีกโครโมโซมหนึ่ง หรือโอกาสที่เกิด Mutation ซึ่งโดยปกติจะเป็นค่าที่ต่ำมากสำหรับยีนที่เข้ารหัสแบบไบนารี เช่น 0.001

F. Termination Criteria

เกณฑ์ที่จะกำหนดว่าเมื่อใดควรจะหยุดกระบวนการสร้างแบบจำลองแทน เกณฑ์การยุติจะมีความสำคัญต่อการรักษาสมดุลระหว่างความแม่นยำและประสิทธิภาพ

G. Crossover and Mutation Operators

- Crossover
  เป็นตัวดำเนินการที่ใช้ในการเปลี่ยนแปลงการโปรแกรมของโครโมโซมหรือโครโมโซมจากรุ่นหนึ่งไปยังอีกรุ่นหนึ่ง มีการเลือกสายสองสายจากแหล่งผสมพันธุ์โดยการสุ่มโครโมโซมมาทำการสลับตำแหน่งของโครโซมหนึ่งไปยังอีกตัวหนึ่งในตำแหน่งใดตำแหน่งหนึ่ง เพื่อผลิตลูกหลานที่ดีกว่า วิธีที่เลือกจะขึ้นอยู่กับวิธีการเข้ารหัส

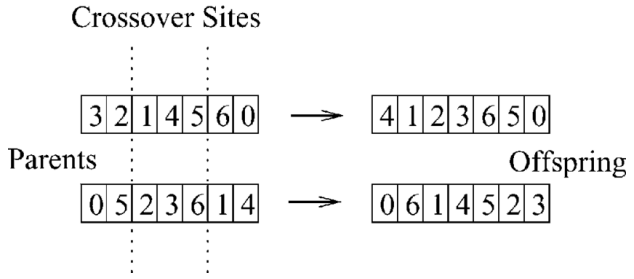

Fig. 1. Partially Matched Crossover (PMX)

- Mutation
  เป็นตัวดำเนินการที่จะใช้เพื่อรักษาความหลากหลายทางพันธุกรรมของโครโมโซมของประชากร โดยจะใช้ในการเปลี่ยนแปลงแบบสุ่มกับยีนหนึ่งรายการขึ้นไปเพื่อสร้างลูกหลานใหม่

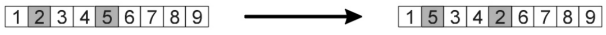

Fig. 2. Random Resetting Mutation

H. Elitism

เป็นกลยุทธ์ในอัลกอริทึมวิวัฒนาการ ที่คัดเลือกชุดคำตอบที่ดีที่สุด จากแต่ละรุ่น แล้วนำไปสู่รุ่นต่อไปโดยไม่ต้องมีการเปลี่ยนแปลงใด ๆ กลยุทธ์นี้ช่วยเร่งความเร็วในการบรรจบของอัลกอริทึม แต่ต้องใช้ร่วมกับกลยุทธ์อื่น ๆ เพื่อป้องกันการสูญพันธุ์

I. Brute force algorithm

เป็นอัลกอริธึมประเภทพื้นฐานและง่ายที่สุด อัลกอริทึมแบบ Brute Force เป็นแนวทางที่ตรงไปตรงมาในการแก้ไขปัญหา นั่นคือแนวทางแรกที่เรานึกถึงเมื่อมองเห็นปัญหา ซึ่งในทางเทคนิคแล้ว มันก็เหมือนกับการวนซ้ำทุกความเป็นไปได้เพื่อแก้ไขปัญหานั้น

J. Combination

ในทางคณิตศาสตร์เป็นวิธีการเลือกสิ่งของจำนวนหนึ่งมาจากสิ่งของที่มีอยู่ทั้งหมด โดยไม่คำนึงถึงลำดับของการจัดหมู่สิ่งของ k สิ่ง จากสิ่งของทั้งหมด n สิ่ง มีวิธีการจัดทั้งหมด

$$ {}_nC_r = \binom{n}{r} = \frac{n!}{r!(n-r)!} $$

Fig. 3. สมการของ Combination

จำนวนความเป็นไปได้ทั้งหมดของรหัสที่เป็นไปได้ใน Mastermind จะมีค่าเท่ากับจำนวนของสีทั้งหมดยกกำลังด้วยความยาวของรหัส โดยในกรณีที่มี 6 สีและรหัสมีความยาว 4 สี จำนวนความเป็นไปได้ทั้งหมดคือ 6^4 = 1,296 กรณี

III. วิธีดำเนินการ

ในการทดลองสร้างเกม Mastermind แบบ 6 สี 4 ตำแหน่ง และแบบ 8 สี 5 ตำแหน่งโดยใช้ Genetic Algorithm 2 แบบ และ Brute force algorithm 1 แบบ โดยแบ่งวิธีการดังนี้

- Genetic Algorithm (C programming)
  เลือกใช้ Population Size 40 และ Parent Selection ใช้ generation ปัจจุบัน, ใช้วิธีการ PMX Crossing over มี Crossing over rate 100% ใช้วิธีการ Random resetting mutation มี mutation rate 12% และ Elitism 0%
- Genetic Algorithm (Wolfram Mathematica)
  เลือกใช้ Population Size 10 และ Parent Selection ใช้ generation ปัจจุบัน 8 และ generation ก่อนหน้า 2 ใช้วิธีการ Uniform Crossoverover มี Crossing over Random rate 0 – 100 % ใช้วิธีการ Random resetting mutation มี Random mutation rate 0 – 10 % และ Elitism 0%
- Brute force algorithm (C programming)
  เลือกใช้ Brute Force ในการค้นหาโค้ดลับที่ถูกต้อง โดยจะสุ่มเลือกมาทดสอบจนกว่าจะค้นพบโค้ดที่ถูกต้อง

ผู้จัดทำจะทำการนำอัลกอริทึมทั้งสามรูปแบบมาทำการคาดเดารหัสลับ โดยเล่นเกมทั้งหมด 3 เกมเพื่อหาจำนวนการคาดเดาในแต่ละเกมและในแต่ละอัลกอริทึม จากนั้นทำการหาค่าเฉลี่ยจำนวนครั้งในการคาดเดาของอัลกอริทึมทั้งสามแบบและบันทึกผลลงตาราง

IV. ผลการทดลอง

จากการทดลองได้นำวิธีการ Genetic Algorithm แบบ C programming และ Wolfram Mathematica เพื่อนำมาทำการเปรียบเทียบ parameter ของวิธีการ Genetic Algorithm ทั้งสองแบบแล้วนำเสนอข้อมูลแบบตาราง ดังตางรางที่ 1

TABLE I
ตารางการปรับแต่ง Genertic Algorithm ทั้งสองแบบ

parameter	Genetic Algorithm (C programming)	Genetic Algorithm (Wolfram Mathematica)
Population size	40	10
Termination criteria	More than 0 element in Eligible Set	Maxgen <= 10 Eligible Set <= 10
Parent selection method	current generation	current generation = 8 previous generation = 2
Crossover method	PMX	Uniform Crossover
Crossover rate	100 %	Random 0-100 %
Mutation method	Random resetting	Random resetting
Mutation rate	12 %	Random 0-10%
Elitism	0 %	0 %

จากการทดลองสร้างเกม Mastermind ในรูปแบบของ 6 สี 4 ตำแหน่ง ได้นำวิธีการ Genetic Algorithm 2 แบบ คือ แบบ C programming และ Wolfram Mathematica และวิธีการ Brute force algorithm จากนั้นจะทำการเก็บรวบรวมผลการทดลอง และนำผลการทดลองที่ได้ไปวิเคราะห์หาข้อมูล แล้วนำเสนอข้อมูลแบบตาราง ดังตารางที่ 2 และ 3

TABLE II
ตารางบันทึกผลจำนวนการคาดเดารหัสลับและค่าเฉลี่ยการคาดเดาของแต่ละ algorithm ของเกม mastermind แบบ 6 สี 4 ตำแหน่ง

จำนวนการคาดเดา ของ Algorithm	จำนวนเกมที่เล่น			ค่าเฉลี่ย
	เกมที่ 1	เกมที่ 2	เกมที่ 3	
Genetic Algorithm (C programming)	2	4	3	3.00
Genetic Algorithm (Wolfram Mathematica)	2	3	4	3.00
Brute force algorithm	513	1046	117	558.67

TABLE III
ตารางแสดงผลการเล่นของเกม mastermind แบบ 6 สี 4 ตำแหน่ง ของ Genetic Algorithm ทั้งสองแบบ

Guesses และ Solution ของแต่ละเกม		จำนวนการคาดเดาของ Algorithm	
		Genetic Algorithm (C programming)	Genetic Algorithm (Wolfram Mathematica)
เกมที่ 1	Guesses	1123, 1562	4432, 5256
	Solution	1562	5256
เกมที่ 2	Guesses	1123, 2344, 6331, 3331	2552, 4451, 4453
	Solution	3331	4453
เกมที่ 3	Guesses	1123, 1252, 3164, 3626	6323, 4224, 4521, 4525
	Solution	3626	4525

จากการทดลองสร้างเกม mastermind ในรูปแบบของ 8 สี 5 ตำแหน่ง ได้นำวิธีการ Genetic Algorithm 2 แบบ คือ แบบ C programming และ Wolfram Mathematica และวิธีการ Brute force algorithm จากนั้นจะทำการเก็บรวบรวมผลการทดลอง และนำผลการทดลองที่ได้ไปวิเคราะห์หาข้อมูล แล้วนำเสนอข้อมูลแบบตาราง ดังตารางที่ 4 และ 5

TABLE IV
ตารางบันทึกผลจำนวนการคาดเดารหัสลับและค่าเฉลี่ยการคาดเดาของแต่ละ algorithm ของเกม mastermind แบบ 8 สี 5 ตำแหน่ง

จำนวนการคาดเดา ของ Algorithm	จำนวนเกมที่เล่น			ค่าเฉลี่ย
	เกมที่ 1	เกมที่ 2	เกมที่ 3	
Genetic Algorithm (C programming)	5	5	4	4.67
Genetic Algorithm (Wolfram Mathematica)	5	4	4	4.33
Brute force algorithm	4784	10227	49013	21341.33

TABLE V
ตารางแสดงผลการเล่นของเกม mastermind แบบ 8 สี 5 ตำแหน่ง ของ Genetic Algorithm ทั้งสองแบบ

Guesses และSolution ของแต่ละเกม		จำนวนการคาดเดาของ Algorithm	
		Genetic Algorithm (C programming)	Genetic Algorithm (Wolfram Mathematica)
เกมที่ 1	Guesses	11234, 61323, 42364, 78213, 21816	65157, 73417, 33551, 82637, 86327
	Solution	21816	86327
เกมที่ 2	Guesses	11234, 47283, 58241, 23271, 73271	78275, 35836, 53388 43518
	Solution	73271	43518
เกมที่ 3	Guesses	11234, 37263, 74536, 45736	63616, 32486, 83264, 43862
	Solution	45736	43862

V. สรุปและอภิปรายผล

การทดลอง Mastermind โดยการใช้ Genetic Algorithm แบบ C programming และ Wolfram Mathematica เทียบกับ Brute Force Algorithm และการเปรียบเทียบในความมีประสิทธิภาพของ Guesses และ Solution ของ Genetic Algorithm ระหว่างแบบ C programming และแบบ Wolfram Mathematica สามารถสรุปได้ดังนี้

- Genetic Algorithm:
  - ทั้ง Genetic Algorithm แบบ C programming และ Wolfram Mathematica มีค่าเฉลี่ยจำนวนของการคาดเดาที่ใกล้เคียงกัน จากกรณี Mastermind 6 สี 4 ตำแหน่ง และ Mastermind 8 สี 5 ตำแหน่ง

  - Genetic Algorithm ทั้งสองรูปแบบมีประสิทธิภาพในการทำงานเมื่อถูกนำมาใช้ในการคาดการณ์ จากการทดสอบแสดงให้เห็นได้ว่าทั้งสองรูปแบบนั้นจะมีประสิทธิภาพในการคาดเดาที่ใกล้เคียงกัน
  - การเลือกใช้ Genetic Algorithm แบบไหนขึ้นอยู่กับความต้องการ และทรัพยากรที่มีอยู่ เช่น ขนาดของ Population และเวลาที่สามารถให้กับการทดลอง
- Brute Force Algorithm:
  - Brute Force Algorithm มีจำนวนการคาดเดาสูงมาก เนื่องจากต้องทดสอบทุกรายการที่เป็นไปได้
  - มีความช้ามาก และไม่เหมาะสมสำหรับปัญหาขนาดใหญ่ เนื่องจากต้องทดลองทุกรูปแบบที่เป็นไปได้
- การเปรียบเทียบความมีประสิทธิภาพของ Guesses และ Solution
  - Genetic Algorithm (C programming) จะกำหนด Initial Guess ไว้เป็นค่าเดิมตลอด ดังนั้นจะทำการเปรียบเทียบที่ Solution แทน ค่าที่ดีที่สุด คือ 1562 ของ 6 สี 4 ตำแหน่ง, 45736 ของ 8 สี 5 ตำแหน่ง และค่าที่แย่ที่สุด คือ 3331 ของ 6 สี 4 ตำแหน่ง, 21816, 73271 ของ 8 สี 5 ตำแหน่ง
  - Genetic Algorithm (Wolfram Mathematica) ทำการสุ่มค่า Initial Guess จึงทำการเปรียบเทียบที่ Initial Guess และ Solution ค่าที่ดีที่สุด คือ Initial Guess คือ 4432, Solution คือ 5256 ของ 6 สี 4 ตำแหน่ง, Initial Guess คือ 78275, 63616, Solution คือ 43518, 43862 ตามลำดับ ของ 8 สี 5 ตำแหน่ง, ค่าที่แย่ที่สุด คือ Initial Guess คือ 6323, Solution คือ 4525 ของ 6 สี 4 ตำแหน่ง, Initial Guess คือ 65157, Solution คือ 86327 ของ 8 สี 5 ตำแหน่ง

ดังนั้นจะสรุปได้ว่า Genetic Algorithm จึงเป็นทางเลือกที่ดีในการคาดเดา Mastermind เมื่อนำเปรียบเทียบกับ Brute Force Algorithm ที่มีความช้ามาก และการเลือกใช้ Genetic Algorithm แบบไหนจะขึ้นอยู่กับความต้องการของปัญหาและทรัพยากรที่มีอยู่ เนื่องจากทั้ง Genetic Algorithm ในรูปแบบ C programming และ ในรูปแบบ Wolfram Mathematica มีประสิทธิภาพใกล้เคียงกัน แต่อาจมีความแตกต่างในกรณีใดกรณีหนึ่ง เช่น ขนาดของ Population และอัลกอริทึมที่ใช้ในการ Crossover และ Mutation

References

[1] https://www.chessgoshop.com/category/85/เกมฝึกiq/เกมถอดรหัส-mastermind
[2] https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html
[3] https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm
[4] https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/
[5] https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)
[6] http://www.ai-junkie.com/ga/intro/gat2.html
[7] https://www.igi-global.com/dictionary/multi-objective-evolutionary-algorithms/9592
[8] https://web.itu.edu.tr/ etaner/courses/NIC/handouts/genetic_algorithms_handouts.pdf
[9] https://www.sciencedirect.com/topics/computer-science/termination-criterion
[10] B.R.Rajakumara, Aloysius George, "APOGA: An Adaptive Population Pool Size Based Genetic Algorithm", Published by Elsevier B.V, vol.4, 2013
[11] R. Santiago-Mozos, Sancho Salcedo-Sanz, Mario DePrado-Cumplido, Carlos Bousoño-Calzón, "A two-phase heuristic evolutionary algorithmfor personalizing course timetables: A casestudy in a Spanish university" , Published by researchGate , 2003