

เกม Sudoku

Sudoku game

นายวสันต์ ภู่วโรดม รหัส 46362083

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ.....2.5.11ค. 2553/.....
เลขทะเบียน.....15004-226.....
เลขเรียกหนังสือ.....ปร.....
มหาวิทยาลัยนเรศวร 4358ก
549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2549




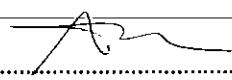
ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	เกม Sudoku
ผู้ดำเนินโครงการ	นายวสันต์ กุวัชรอม รหัส 46362083
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของ
การศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะกรรมการสอบโครงการวิศวกรรม


.....ประธานกรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)


.....กรรมการ
(ดร.พนมขวัญ ริยะมงคล)


.....กรรมการ
(ดร.สมยศ เกียรติวนิชวิไล)

หัวข้อโครงการ	เกม Sudoku		
ผู้ดำเนินโครงการ	นายวสันต์	ภู่วโรดม	รหัส 46362083
อาจารย์ที่ปรึกษา	ดร.สุรเดช	จิตประไพกุลศาล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2549		

บทคัดย่อ

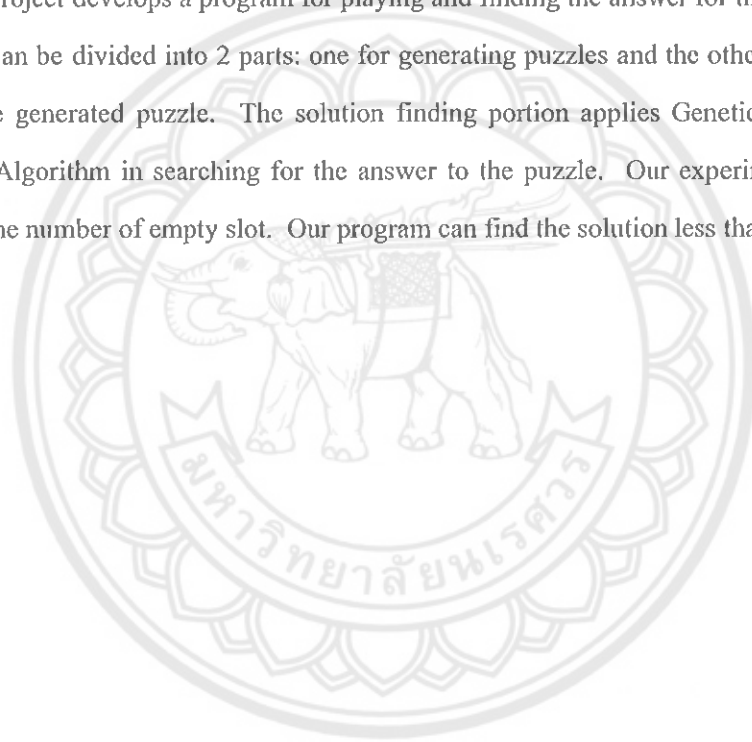
โครงการนี้ได้พัฒนาโปรแกรมประเภทปัญญาประดิษฐ์ (Artificial Intelligence) เพื่อเล่นเกม Sudoku โดยโปรแกรมที่พัฒนานั้นแยกได้เป็น 2 ส่วนคือ ส่วนของการตั้งปริศนา และส่วนของการหาคำตอบ ซึ่งส่วนของการหาคำตอบนั้นจะใช้หลักการของ Genetic Algorithm และ Backtracking Algorithm ในการหาคำตอบของเกม จากการทดลองพบว่าโปรแกรมสามารถหาคำตอบของเกมได้จากทุกจำนวนช่องว่าง



Project title	Sudoku game		
Name	Mr.Wasan	Poovarodom	ID. 46362083
Project advisor	Dr.Suradet	Jitprapaikulsam	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2006		

Abstract

This project develops a program for playing and finding the answer for the Sudoku game. Our program can be divided into 2 parts: one for generating puzzles and the other for finding the answer for the generated puzzle. The solution finding portion applies Genetic Algorithm and Backtracking Algorithm in searching for the answer to the puzzle. Our experiments show that regardless of the number of empty slot. Our program can find the solution less than 5 seconds.



กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์สำเร็จได้ด้วยดี ก็เนื่องจากความอนุเคราะห์จากท่าน
อาจารย์ที่ปรึกษา ดร.สุวเดช จิตประไพกุลศาสตร์ ที่กรุณาให้คำปรึกษา แนะนำวิธีการในการทำงาน
ตลอดถึงการตรวจสอบการทำงาน พร้อมทั้งเสนอแนะทางการแก้ไขตลอดระยะเวลาการทำโครงการ
สุดท้ายต้องขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆ ทุกคนที่ยังไม่ได้เอ่ยนามที่คอยให้คำแนะนำ
ในด้านต่างๆ และสนับสนุนในการทำโครงการครั้งนี้



สารบัญ

หน้า

บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของ โครงการงาน	1
1.2 วัตถุประสงค์โครงการ	1
1.3 ขอบข่ายการทำงาน	1
1.4 ขั้นตอนดำเนินงาน	2
1.5 แผนการดำเนินงาน	2
1.6 ผลที่คาดว่าจะได้รับ	3
1.7 งบประมาณ	3
บทที่ 2 หลักการและทฤษฎี	4
2.1 ประวัติความเป็นมาของเกม Sudoku	4
2.2 ขั้นตอนวิธีเชิงพันธุกรรมหรือเจเนติกอัลกอริทึม (Genetic Algorithms: GAs)	5
2.3 Backtracking Algorithm	8
บทที่ 3 วิธีการดำเนินการ	10
3.1 Start	11
3.2 Choose next step	12
3.3 Generate the solution	12
3.4 Evaluate the solution	14
3.5 Check solution	14
3.6 Store bad move	15
3.7 Backtracking	15
3.8 Termination Criterion	15
3.9 End	15

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง.....	16
บทที่ 5 สรุปผล	20
5.1 สรุปผลการทดลอง.....	20
5.2 ปัญหาและอุปสรรค	21
5.3 ข้อเสนอแนะ	21
บรรณานุกรม	22
ประวัติผู้เขียน โครงการงาน.....	23



สารบัญตาราง

หน้า

ตารางที่ 4.1 แสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบของเกม Sudoku	16
ตารางที่ 4.1 (ต่อ) แสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบของเกม Sudoku.....	17
ตารางที่ 4.1 (ต่อ) แสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบของเกม Sudoku.....	18



สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงตัวอย่างของตาราง ซูโดะกุ ขนาด 9×9	4
รูปที่ 2.2 แสดงขั้นตอนการทำงานของ Genetic Algorithm	6
รูปที่ 2.3 แสดงตัวอย่างของการสุ่ม Population จากข้อมูลทั้งหมด	6
รูปที่ 2.4 แสดงตัวอย่างของกระบวนการ Crossover	7
รูปที่ 2.5 แสดงตัวอย่างของกระบวนการ Mutation.....	8
รูปที่ 2.6 แสดงตัวอย่างการทำงานของ Backtracking Algorithm.....	9
รูปที่ 3.1 แสดงแผนผังแสดงขั้นตอนการทำงานของโปรแกรม	10
รูปที่ 3.2 แสดงตัวอย่างของ โจทย์ที่ต้องการให้ Bot หาคำตอบ	11
รูปที่ 3.3 แสดงตัวอย่างของเซตของคำตอบ	12
รูปที่ 3.4 แสดงตัวอย่างการเลือกของลงคำตอบ Bot.....	12
รูปที่ 4.1 แสดงกราฟของเวลาที่ใช้ในการหาคำตอบของโปรแกรมต่อจำนวน Empty slot	18
รูปที่ 4.2 แสดงกราฟของช่วงเวลาประมาณที่โปรแกรมใช้หาคำตอบ.....	19

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันการเสริมสร้างสติปัญญานั้นไม่จำเป็นต้องได้จากการเรียนหนังสือในห้องเรียนเพียงอย่างเดียวเท่านั้น เกมในปัจจุบันก็สามารถช่วยเสริมสร้างสติปัญญาได้เช่นกัน ซึ่งเกมในปัจจุบันนั้นนอกจากจะช่วยให้เกิดความสนุกสนานและผ่อนคลายแล้วยังช่วยเสริมสร้างความคิดและสติปัญญาอีกด้วย และเกมที่โครงการนี้เลือกมาศึกษาคือ เกม Sudoku

Sudoku (เขียนอีกอย่างหนึ่งว่า Su Doku และอ่านออกเสียง “ซู - โด - กู”) คือ เกมปริศนาตารางตัวเลข ที่มีกฎกติกาอยู่ด้วยกันหลายรูปแบบ ซึ่งในโครงการนี้จะเลือกมา 1 แบบ คือ ผู้เล่นจะต้องเติมตัวเลขลงในช่องว่างของตาราง 9×9 ช่อง โดยจะต้องใช้ตัวเลข 1 - 9 ไม่ให้ซ้ำกัน และใช้ตัวเลขแต่ละตัวได้เพียงครั้งเดียวเท่านั้น ทั้งในทุกแถวของแนวนตั้ง แนวนอน และตารางย่อยขนาด 3×3 ผู้เล่นจะต้องใช้หลักเหตุและผลหรือตรรกะ (logic) ในการไขปริศนานี้ โดยดูจากข้อมูลตัวเลขที่ให้มีมา

สำหรับความยากง่ายของเกมนั้นขึ้นอยู่กับปัจจัยต่างๆ เช่น การจัดวางตัวเลขของโจทย์ และจำนวนตัวเลขที่โจทย์บอกใบ้มาให้ แต่ก็ไม่ได้หมายความว่าเมื่อบอกใบ้ตัวเลขมาหลายตัวแล้ว โจทย์นั้นๆ จะง่ายกว่า โจทย์ที่บอกใบ้ตัวเลขน้อยกว่า ทั้งนี้เพราะมีการจัดวางตัวเลขไว้ให้แก่โจทย์ได้ยากขึ้นนั่นเอง

โครงการนี้ได้สร้างเกม Sudoku โดยจะแสดงผลเป็น Graphic เพื่อให้มีความสวยงามและง่ายต่อการเล่นและนอกจากจะทำในส่วนของตัวเกมแล้วยังมีส่วนของ AI (Artificial Intelligence) ที่จะใช้หลักการคิดทางตรรกะในการหาคำตอบในโจทย์ต่างๆ

1.2 วัตถุประสงค์โครงการ

1. สามารถพัฒนาเกม Sudoku
2. สามารถพัฒนา Bot ของเกม Sudoku

1.3 ขอบข่ายการทำงาน

1. ศึกษากฎกติกาของเกม Sudoku
2. ศึกษาวิธีการเล่นเกม Sudoku
3. สร้างและพัฒนาเกม Sudoku

1.6 ผลที่คาดว่าจะได้รับ

1. เข้าใจรูปแบบและวิธีการเล่นของเกม Sudoku
2. สามารถสร้างและพัฒนาเกม Sudoku
3. สามารถนำความรู้ในเรื่องของ AI มาพัฒนา Bot ของเกม Sudoku

1.7 งบประมาณ

1. ค่าวัสดุสำนักงาน	เป็นเงิน	500	บาท
2. ค่าถ่ายเอกสาร	เป็นเงิน	500	บาท
รวมเป็นเงินทั้งสิ้น		1,000	บาท (หนึ่งพันบาทถ้วน)



บทที่ 2

หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงประวัติความเป็นมาและกติกาของเกม Sudoku รวมไปถึงถึงทฤษฎีที่นำมาใช้ในการหาคำตอบของเกม Sudoku ซึ่งประกอบไปด้วย Genetic Algorithm ซึ่งเป็นวิธีการหา Solution โดยใช้หลักการทางชีววิทยาและ Backtracking Algorithm ซึ่งเป็นอีก Algorithm ที่นิยมนำมาใช้ในการหา Solution

2.1 ประวัติความเป็นมาของเกม Sudoku

ซูโดกุ (Sudoku) เกมปริศนาตัวเลข ที่ผู้เล่นต้องเลือกใส่ หมายเลขตั้งแต่ เลข 1 ถึงเลข 9 โดยมีเงื่อนไขว่าในแต่ละแถวและแต่ละหลักตัวเลขต้องไม่ซ้ำกัน ตารางซูโดกุจะมี 9×9 ช่อง ซึ่งประกอบจากตารางย่อย 9 ตาราง ในลักษณะ 3×3 แบ่งแยกกันโดยเส้นหนา และในแต่ละตารางย่อยจะมีตัวเลข 1 ถึง 9 เช่นเดียวกัน เมื่อเริ่มเกมจะมีตัวเลขบางส่วนให้มาเป็นคำใบ้ และผู้เล่นจะต้องใส่ทุกช่องที่เหลือให้ครบ โดยตามเงื่อนไขว่าแต่ละตัวเลขในแต่ละแถวและหลักจะใช้ได้ครั้งเดียว รวมถึงในแต่ละขอบเขตตารางย่อย การเล่นเกมนี้จำเป็นต้องใช้ความสามารถในด้าน ตรรกะ และ ความอดทนรวมถึงสมาธิ เกมนี้เริ่มต้นเป็นครั้งแรกในสหรัฐอเมริกาในปี พ.ศ. 2522 ในชื่อ นัมเบอร์เพลส (Number Place) แต่เป็นที่นิยมและโด่งดังในประเทศญี่ปุ่น ภายใต้ชื่อ ซูโดกุ ตั้งแต่ปี พ.ศ. 2529 และเป็นที่นิยมทั่วโลกอีกครั้งในปี พ.ศ. 2548

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

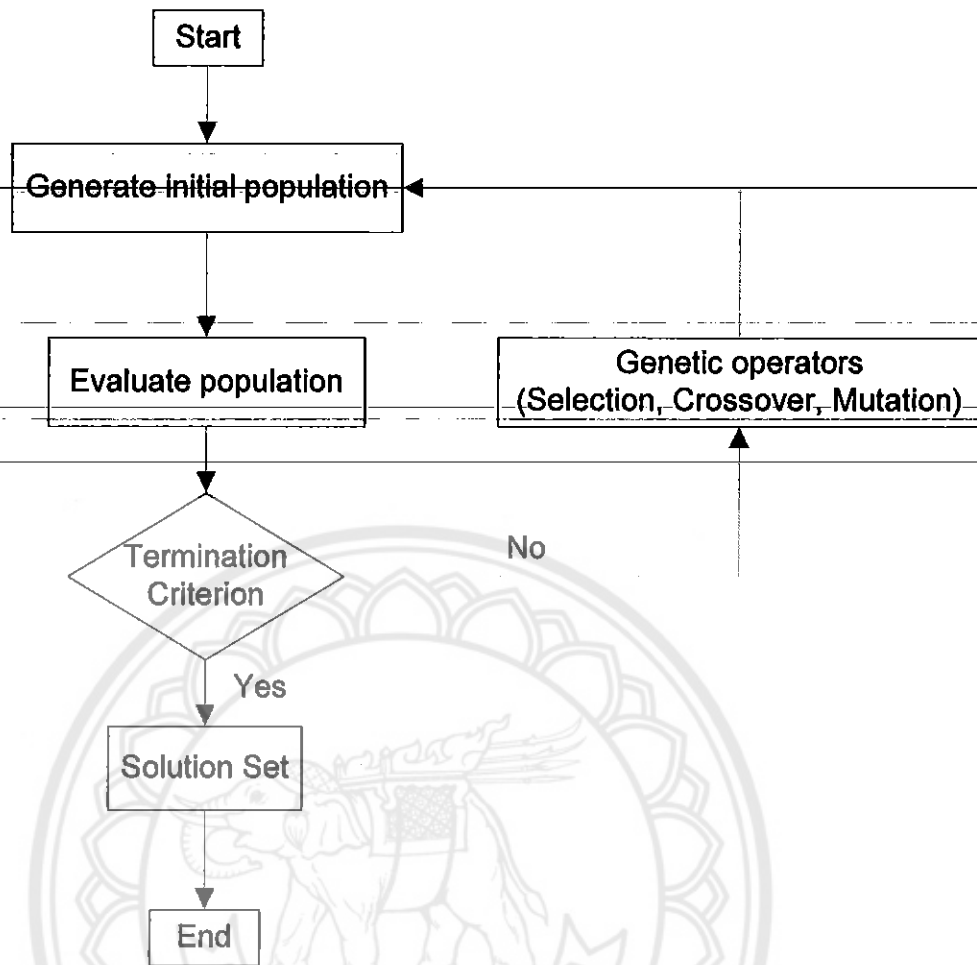
รูปที่ 2.1 แสดงตัวอย่างของตาราง ซูโดกุ ขนาด 9×9

รูปภาพจาก <http://upload.wikimedia.org/wikipedia/commons/thumb/f/ff/Sudoku-by-L2G-20050714.svg/250px-Sudoku-by-L2G-20050714.svg.png>

2.2 ขั้นตอนวิธีเชิงพันธุกรรมหรือเจเนติกอัลกอริทึม (Genetic Algorithms: GAs)

ขั้นตอนวิธีเชิงพันธุกรรม หรือ Genetic Algorithm เป็นเทคนิคสำหรับค้นหาผลเฉลย (Solutions) หรือคำตอบโดยประมาณของปัญหา โดยอาศัยหลักการจากทฤษฎีวิวัฒนาการจากชีววิทยา และการคัดเลือกตามธรรมชาติ (Natural selection) นั่นคือ สิ่งมีชีวิตที่เหมาะสมที่สุดจึงจะอยู่รอด

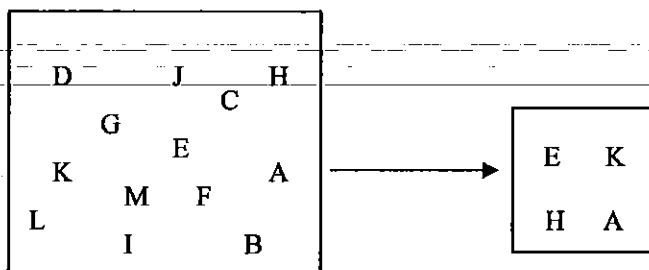
กระบวนการคัดเลือกได้เปลี่ยนแปลงสิ่งมีชีวิตให้เหมาะสมยิ่งขึ้น ด้วยตัวปฏิบัติการทางพันธุกรรม (Genetic operator) เช่น การสืบพันธุ์ (Inheritance หรือ Reproduction), การกลายพันธุ์ (Mutation), การแลกเปลี่ยนยีน (Recombination) ขั้นตอนวิธีเชิงพันธุกรรมเป็นการจำลองทางคอมพิวเตอร์ เพื่อแก้ปัญหาหาค่าที่เหมาะสมที่สุด (Optimal solution) โดยการแทนคำตอบที่มีอยู่ในลักษณะ โครโมโซม (Chromosomes) แล้วปรับปรุงคำตอบแต่ละชุด (เรียกว่า Individual) ด้วยวิธีการต่าง ๆ ซึ่งเกี่ยวข้องกับการวิวัฒนาการ (Evolutionary operation) การเปลี่ยนแปลงยีนแบบสุ่ม ด้วยตัวปฏิบัติการทางพันธุกรรม (Evolutionary operator) เพื่อให้ได้คำตอบที่ดีขึ้น โดยทั่วไปจะแทนคำตอบด้วยเลขฐานสอง (สายอักขระของเลข 0 และ 1) การวิวัฒนาการ (Evolution) เพื่อหาคำตอบที่เหมาะสมที่สุด (The fitness solution) จะเริ่มจากประชากรที่ได้จากการสุ่มทั้งหมดและจะทำเป็นรุ่น ๆ ในแต่ละรุ่นคำตอบหลายชุดจะถูกสุ่มเลือกขึ้นมาเปลี่ยนแปลง ซึ่งอาจทำให้เกิดการกลายพันธุ์หรือสืบเปลี่ยนยีนระหว่างกัน จนได้ประชากรรุ่นใหม่ ที่มีค่าความเหมาะสม (Fitness) มากขึ้น การวิวัฒนาการนี้จะทำไปเรื่อย ๆ จนกระทั่งพบคำตอบที่มีค่าความเหมาะสมตามต้องการ



รูปที่ 2.2 แสดงขั้นตอนการทำงานของ Genetic Algorithm

2.2.1 Generate Initial population

เป็นการกำหนดค่าเริ่มต้นให้กับ Chromosome แต่ละชุด โดยจะสุ่มเลือกค่าของ Population จากข้อมูลทั้งหมดมาเก็บไว้ใน Chromosome



ข้อมูลทั้งหมด

ขนาดของ Population = 4

รูปที่ 2.3 แสดงตัวอย่างของการสุ่ม Population จากข้อมูลทั้งหมด

2.2.2 Evaluate population

เป็นการคำนวณหาค่า Fitness value ของ Chromosome แต่ละตัวเพื่อหาว่า Chromosome ชุดใดที่มีความเหมาะสมที่จะนำมาใช้มากที่สุด ซึ่งหลักการคำนวณหาค่า Fitness value ก็จะมีหลายวิธีแตกต่างกันไป

2.2.3 Genetic operators

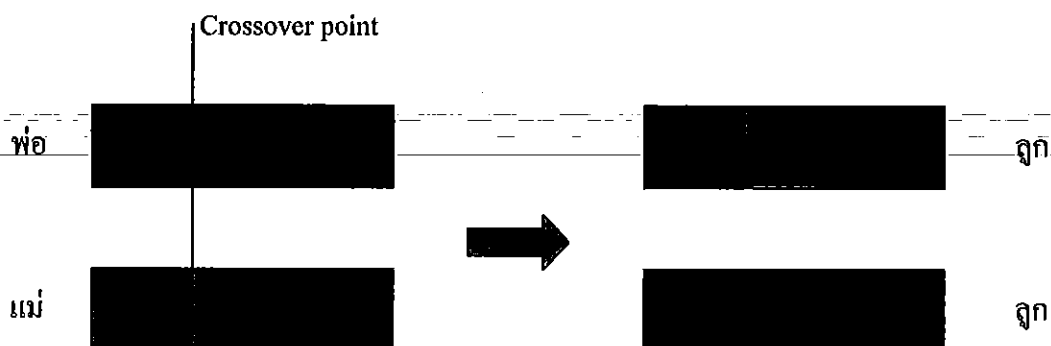
ใน Genetic Algorithm จะมีวิธีการที่ใช้ในการเปลี่ยนแปลงรูปแบบของ Chromosome เพื่อที่จะหา Chromosome ที่มีความเหมาะสมมากที่สุดอยู่ด้วยกัน 3 วิธี คือ

1. Selection

เป็นการเลือก Chromosome ที่ดีที่สุด (โดยจะดูจากค่าของ Fitness value เป็นเกณฑ์ในการคัดเลือก) ซึ่งจะนำ Chromosome ที่เลือกมานี้ไปทำในขั้นตอน Crossover ต่อไป

2. Crossover

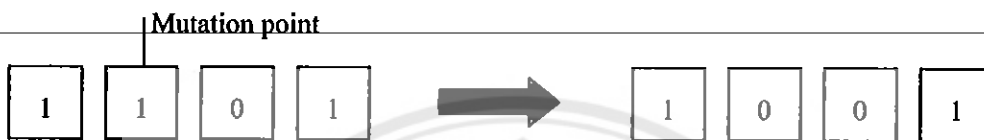
กระบวนการ Crossover นั้นถือเป็นส่วนสำคัญของ Genetic algorithm เลขก็ว่าได้ เนื่องจากในกระบวนการ Crossover นั้นจะมีการเปลี่ยนแปลงของ Chromosome มากที่สุด โดยขั้นตอนของการ Crossover นั้นจะเลือก Chromosome ขึ้นมา 2 ชุด สมมติให้เป็นพ่อและแม่ ดังนั้น Chromosome ลูกนั้นจะเกิดจากการผสมกันระหว่าง Chromosome พ่อและแม่โดยที่ Crossover point เป็นตัวแบ่งว่า Chromosome ลูกจะได้ Genes ใดของ Chromosome พ่อและแม่ไปบ้าง โดย Crossover point นั้นจะหาได้จากการสุ่มค่าตำแหน่งของ Genes จากจำนวนของ Genes ทั้งหมดของ Chromosome ชุดนั้น



รูปที่ 2.4 แสดงตัวอย่างของกระบวนการ Crossover

3. Mutation

กระบวนการ Mutation จะเกิดหลังจากกระบวนการ Crossover นั้นเสร็จสิ้นแล้ว โดยจะนำ Chromosome ลูกที่เกิดจากการผสมระหว่าง Chromosome พ่อและแม่นั้นมาทำการ Mutation ซึ่งจะทำให้ Chromosome ลูกนั้นมีลักษณะใหม่ๆเกิดขึ้นมาโดยจุดที่ต้องการทำการ Mutation นั้นหาได้จากการสุ่มค่าตำแหน่งของ Genes จากจำนวนของ Genes ทั้งหมดของ Chromosome ชุดนั้น

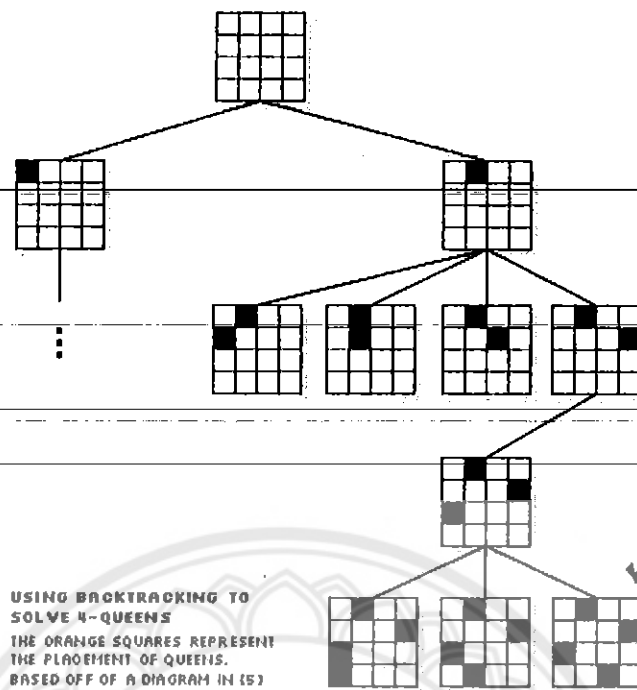


รูปที่ 2.5 แสดงตัวอย่างของกระบวนการ Mutation

2.3 Backtracking Algorithm

Backtracking Algorithm เป็นอีก Algorithm หนึ่งที่เป็นที่นิยมในการใช้หา Solution ต่างๆ เนื่องจากเป็น Algorithm ที่เข้าใจง่ายแล้วมีประสิทธิภาพ โดยหลักการทำงานของ Algorithm นี้คือ จะสุ่มเลือกคำตอบจากเซตของคำตอบทั้งหมด โดยจะสุ่มไปเรื่อยๆจนกว่าจะพบคำตอบแต่ถ้าขณะที่สุ่มอยู่นั้นเกิดสุ่มเจอคำตอบที่ผิดก็จะกลับมาทำ Step ก่อนหน้านั้นและถ้ายังหาคำตอบไม่ได้ก็จะถอยกลับมาทำยัง Step ก่อนหน้านั้นถัดไป ซึ่งวิธีการของ Backtracking Algorithm นั้นคล้ายกับการลองผิดลองถูกคือ ถ้าคำตอบที่เลือกมานั้นถูกต้องก็จะสุ่มหาคำตอบตัวถัดไปต่อไปแต่ถ้าคำตอบที่สุ่มมานั้นผิดก็จะกลับไปเลือกคำตอบก่อนหน้าใหม่

Backtracking Algorithm ที่โครงการนี้นำมาใช้จะเป็นแบบ Depth-first-search ซึ่งหลักการในการหาคำตอบของวิธีนี้จะสุ่มเลือกคำตอบที่เป็นไปได้ทั้งหมดไปเรื่อยๆและเมื่อหาคำตอบต่อไม่ได้แล้วก็จะย้อนกลับมายัง Step ก่อนหน้านั้นและหาคำตอบต่อในตัวเลือกอื่นซึ่งจะไม่เลือกคำตอบที่ผิดตัวนั้นอีก โดยท้ายสุดแล้วถ้าไม่มีคำตอบใดที่สามารถเลือกได้แล้วก็จะถือว่าหาคำตอบไม่ได้



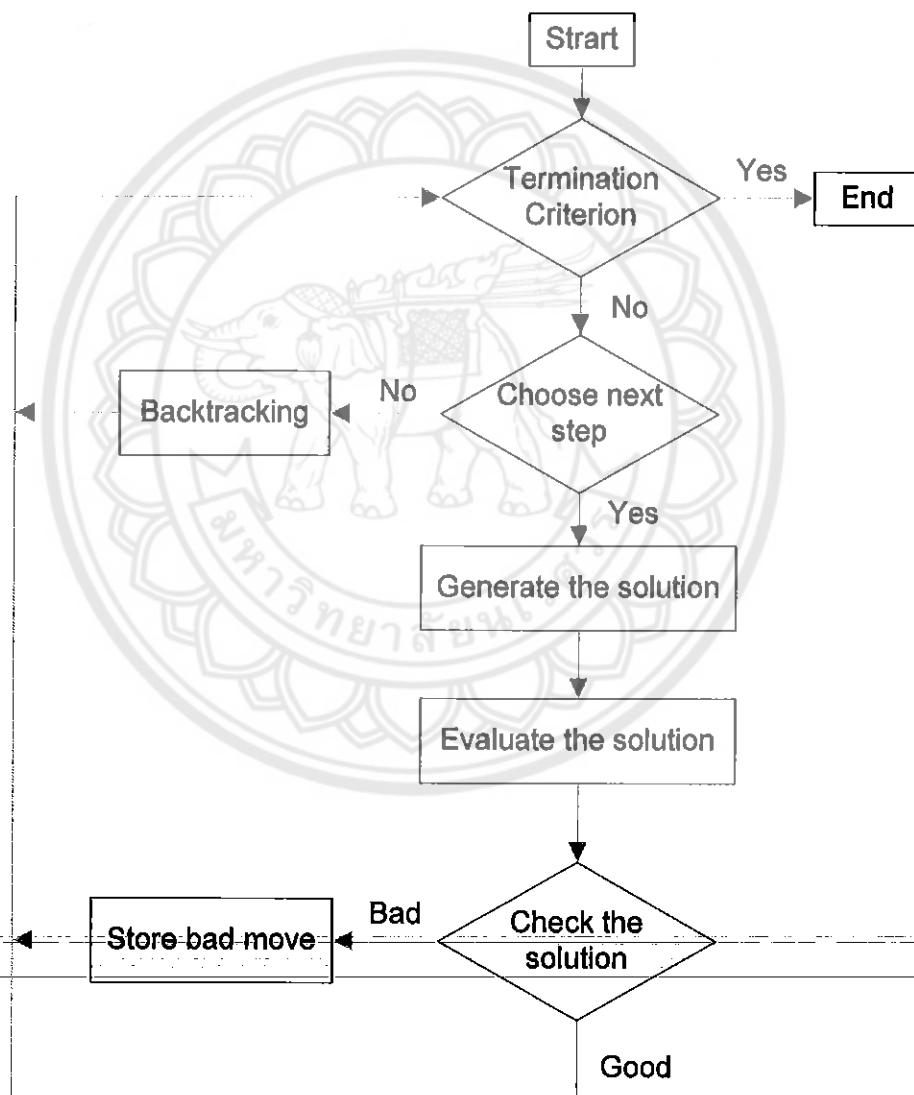
รูปที่ 2.6 แสดงตัวอย่างของ Backtracking Algorithm ที่ใช้หาคำตอบในเกม 4-Queens Problem

รูปภาพจาก <http://4c.ucc.ie/web/outreach/backtracking.jpg>

บทที่ 3

วิธีการดำเนินการ

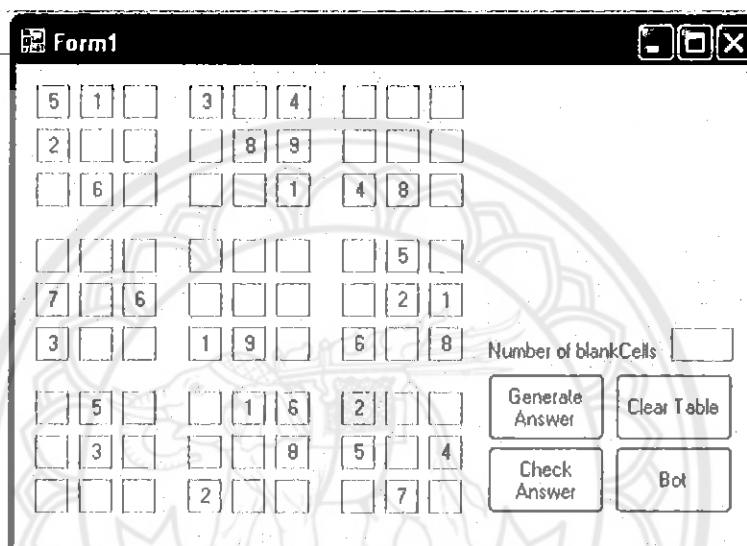
หลังจากที่ได้ศึกษาทฤษฎีของเกม Sudoku และ Algorithm ที่จะนำมาใช้ในการหาคำตอบของเกมแล้ว ในบทนี้จะกล่าวถึงขั้นตอนการทำงานของโปรแกรมที่ใช้ในการหาคำตอบของเกม Sudoku โดยจะนำความรู้ที่ได้จากการศึกษามาประยุกต์ใช้ในการพัฒนาโปรแกรม ซึ่งโปรแกรมที่พัฒนามานั้นมีแผนผังการทำงานดังนี้



รูปที่ 3.1 แสดงแผนผังแสดงขั้นตอนการทำงานของโปรแกรม

3.1 Start

เริ่มแรก Bot จะรับ โจทย์เข้ามาจากตัวโปรแกรมโดย โจทย์ที่รับมานั้นแต่ละช่องจะเป็นตัวเลขที่มีค่าตั้งแต่ 1-9 เท่านั้น จากนั้นก็จะนำตัวเลขที่รับมานั้นมาสร้างเซตของคำตอบของเกมซึ่งจะเก็บค่าของตัวเลขที่สามารถลงในแต่ละช่องที่สามารถลงได้ตามกฎกติกาของเกม Sudoku คือ ตัวเลขที่สามารถลงได้นั้นจะต้องเป็นตัวเลขที่มีค่า 1-9 เท่านั้น และในแถว, หลัก และตารางย่อยขนาด 3x3 ใดๆจะต้องไม่มีตัวเลขที่ซ้ำกัน



รูปที่ 3.2 แสดงตัวอย่างของ โจทย์ที่ต้องการให้ Bot หาคำตอบ

-	-	7, 8, 9	-	2, 6, 7	-	7, 9	6, 9	2, 6, 7, 9
-	4, 7	3, 4, 7	5, 6, 7	-	-	1, 3, 7	1, 3, 6	3, 5, 6, 7
9	-	3, 7, 9	5, 7	2, 5, 7	-	-	-	2, 3, 5, 7, 9
1, 4, 8, 9	2, 4, 8, 9	1, 2, 4, 8, 9	4, 6, 7, 8	2, 3, 4, 6, 7	2, 3, 7	3, 7, 9	-	3, 7, 9
-	4, 8, 9	-	4, 5, 8	3, 4, 5	-	-	-	-
-	9	-	8	5	3, 5	3, 9	-	-
-	2, 4	2, 4, 5	-	-	2, 5, 7	-	4	-
4, 8, 9	-	4, 7, 8, 9	4, 7, 9	-	-	-	3, 9	3, 9
1, 6, 9	-	1, 2, 7, 9	7, 9	7	-	-	1, 6, 9	-
1, 4, 6, 8, 9	4, 8, 9	1, 4, 8, 9	-	3, 4, 5	3, 5	1, 3, 8, 9	-	3, 6, 9

รูปที่ 3.3 แสดงตัวอย่างของเซตของคำตอบ

3.2 Choose next step

ในขั้นตอนนี้จะตรวจสอบว่ามีแถว, หลัก หรือตารางย่อยขนาด 3×3 ใดๆบ้างหรือไม่ที่เมื่อ Bot ลงตัวเลขลงไปแถว, หลัก หรือตารางย่อยขนาด 3×3 ใดๆนั้นแล้วทำให้ Bot สามารถหาคำตอบในช่องอื่นๆต่อได้อีกและไม่ไปกระทบกับคำตอบในช่องต่างๆ

3.3 Generate the solution

ในขั้นตอนนี้ Generate the solution นั้น จะเป็นการสุ่มหาคำตอบโดยจะสุ่มจากเซตของคำตอบของแต่ละช่อง โดยจะมีหลักในการสุ่มคือ จะสุ่มทีละแถว, หลัก หรือตารางย่อยขนาด 3×3 ที่มีจำนวนของคำตอบที่สามารถลงได้น้อยที่สุดก่อน เพราะจะมีโอกาสที่จะหาคำตอบได้ง่ายกว่าที่จะสุ่มลงตัวเลขทีละตัว

-	-	7, 8, 9	-	2, 6, 7	-	7, 9	6, 9	2, 6, 7, 9
-	4, 7	3, 4, 7	5, 6, 7	-	-	1, 3, 7	1, 3, 6	3, 5, 6, 7
9	-	3, 7, 9	5, 7	2, 5, 7	-	-	-	2, 3, 5, 7, 9
1, 4, 8, 9	2, 4, 8, 9	1, 2, 4, 8, 9	4, 6, 7, 8	2, 3, 4, 6, 7	2, 3, 7	3, 7, 9	-	3, 7, 9
-	4, 8, 9	-	4, 5, 8	3, 4, 5	3, 5	3, 9	-	-
-	2, 4	2, 4, 5	-	-	2, 5, 7	-	4	-
4, 8, 9	-	4, 7, 8, 9	4, 7, 9	-	-	-	3, 9	3, 9
1, 6, 9	-	1, 2, 7, 9	7, 9	7	-	-	1, 6, 9	-
1, 4, 6, 8, 9	4, 8, 9	1, 4, 8, 9	-	3, 4, 5	3, 5	1, 3, 8, 9	-	3, 6, 9

รูปที่ 3.4 แสดงตัวอย่างการเลือกของลงคำตอบ Bot

จากรูปจะเห็นว่า Bot นั้นจะเลือกลงในตารางย่อยขนาด 3×3 ที่ 5 ก่อนเนื่องจากเป็น ตารางย่อยที่มีจำนวนตัวเลขที่สามารถลงได้น้อยที่สุด

เมื่อได้ แถว, หลัก หรือตารางย่อยขนาด 3×3 ที่มีจำนวนของตัวเลขที่สามารถลงได้น้อยที่สุดแล้วก็จะทำการสุ่มหาคำตอบ โดยจะใช้วิธีการเก็บคำตอบที่สุ่มไว้ใน Chromosome โดย Chromosome ที่นำมาเก็บคำตอบที่สุ่มนั้นจะมีขนาดเท่ากับ 9 ตัว (ตามขนาดของตาราง 9×9) ซึ่งจะสุ่มหาคำตอบเฉพาะในช่องที่โหนดยังไม่มีการลงตัวเลขลงไป โดยการสุ่มหาคำตอบในแถว, หลัก

กรณีนี้ทั้ง 2 ตัวเลขมีจำนวนช่องที่สามารถลงได้เท่ากัน ซึ่งต้องสุ่มตัวเลขใดตัวเลขหนึ่งมาลงก่อน สมมติว่าสุ่มได้เลข 3 แต่ว่าเลข 3 นั้นก็มีอยู่ 2 ช่องดังนั้นจึงต้องสุ่มช่องที่ลงอีกครั้ง สมมติว่าสุ่มได้ช่องที่ 3

7	5	3	-	2	1	6	4	8
---	---	---	---	---	---	---	---	---

5. ตอนนี้ช่องที่ลงนั้นเหลือเพียงช่องเดียวและตัวเลขที่จะลงนั้นก็เหลือเพียงตัวเดียวเช่นกันก็คือช่องที่ 4 และเลข 9

7	5	3	9	2	1	6	4	8
---	---	---	---	---	---	---	---	---

คำตอบที่ได้ในแต่ละ Chromosome นั้นมีโอกาสที่จะไม่เหมือนกันเพราะคำตอบที่ได้มานั้นเกิดจากการสุ่มดังนั้นจึงต้องทำการสุ่มหาคำตอบเอาไว้หลายๆครั้งเพื่อที่จะมีโอกาสพบคำตอบที่ถูกต้อง

3.4 Evaluate the solution

เป็นการนำ Solution ที่ได้จากการสุ่มนั้นมาคำนวณหาค่า Fitness value เพื่อที่จะเปรียบเทียบว่า Solution ไหนที่มีความเหมาะสมที่จะนำไปใช้มากที่สุด

วิธีการหาค่าของ Fitness value นั้นจะทำได้โดยการนำ Chromosome ที่ต้องการหาค่า Fitness value นั้นมาทดลองลงในช่องนั้นๆ แล้วตรวจสอบดูว่าเมื่อลงตัวเลขไปแล้วมีผลกระทบที่ทำให้ไม่สามารถหาคำตอบต่อไปได้หรือไม่ เช่น เมื่อลงไปแล้วทำให้มีบางช่องที่ไม่สามารถลงตัวเลขได้เลย เป็นต้น ซึ่งถ้าลงตัวเลขไปแล้วไม่มีผลกระทบกับการหาคำตอบครั้งต่อไป Chromosome ชุดนั้นก็จะมีค่าของ Fitness value ที่ต่ำ แต่ถ้าลงไปแล้วทำให้ไม่สามารถหาคำตอบต่อไปได้ก็จะมีค่า Fitness value ที่สูงเช่นกัน

3.5 Check solution

ในขั้นตอนนี้จะเป็นการตรวจสอบว่า Chromosome ที่ Bot ได้สุ่มหาค่าออกมานั้นมี Chromosome ตัวไหนที่มีค่าของ Fitness value = 0 บ้างไหม

- ถ้ามีก็จะนำตัวเลขที่ Chromosome ตัวนั้นเก็บเอาไว้ไปลงในโจทย์ และทำการเก็บสถานะของตารางของเกมขณะนั้นว่าลงตัวเลขอะไรที่ตำแหน่งไหนไปแล้วบ้างเพื่อเอาไว้ใช้ใน

- ถ้ามีก็จะนำตัวเลขที่ Chromosome ตัวนั้นเก็บเอาไว้ไปลงในโจทย์ และทำการเก็บสถานะของตารางของเกมขณะนั้นว่าลงตัวเลขอะไรที่ตำแหน่งไหนไปแล้วบ้างเพื่อเอาไว้ใช้ใน Backtracking แต่ถ้าในกรณีที่มี Chromosome ที่มีค่าของ Fitness value = 0 มากกว่า 1 ตัว ทาง Bot ก็ จะสุ่มมาใช้เพียง 1 ตัวเท่านั้น

- ถ้าไม่มี Chromosome ตัวใดเลยที่มีค่าของ Fitness value = 0 แล้ว Bot ก็จะเก็บค่าเอาไว้ว่า การลงคำตอบในแถว, หลัก หรือตารางย่อยขนาด 3x3 ครั้งนี้ไม่มี Chromosome ตัวใดที่มีค่า Fitness value = 0 เลย (จะถือว่าการลงครั้งนี้เป็น Bad move คือ การลงที่ไม่ควรลงอีก)

3.6 Store bad move

จะเป็นการเก็บค่าของแถว, หลักหรือตารางย่อยขนาด 3x3 ใดๆที่เมื่อ Bot ลงตัวเลขในแถว, หลัก หรือตารางย่อยขนาด 3x3 นั้นๆแล้วทำให้ไม่สามารถหาคำตอบต่อไปของเกมได้ ดังนั้นจึงต้องทำการเก็บเป็นสถิติเอาไว้เพื่อที่จะไม่ให้ Bot นั้นกลับไปเลือกลงในช่องที่ลงไปแล้วทำให้หาคำตอบต่อไม่ได้อีก

3.7 Backtracking

ในขั้นตอน Backtracking นี้จะทำก็ต่อเมื่อ Bot นั้นไม่สามารถหาคำตอบต่อได้แล้วและตัวเลขที่ลงนั้นก็ยังไม่ครบทุกช่อง จึงอาจสันนิษฐานได้ว่าคำตอบที่ Bot ได้ตอบมาก่อนหน้านั้นนี้อาจจะผิดพลาดดังนั้นจึงทำขั้นตอน Backtracking เพื่อที่จะกลับไปแก้ไขในคำตอบที่ผิดพลาดใน Step การลงคำตอบก่อนหน้านี้

หลักการที่ใช้ในการพิจารณาว่าจะใช้ Backtracking เมื่อไรคือ Bot จะย้อนกลับไปทำใน Step ก่อนที่จะลงตัวเลขหน้านี้เพื่อที่จะเลือกลงตัวเลขอื่นและในช่องอื่นแทน และจะไม่กลับมาเลือก ลงใน Step ที่ทำให้หาคำตอบต่อไม่ได้อีก

3.8 Termination Criterion

Bot จะตรวจสอบว่าคำตอบที่ลงไปทั้งหมดขณะนี้ครบทุกช่องหรือยังและถูกต้องหรือไม่ ซึ่งถ้าคำตอบที่ลงไปทั้งหมดขณะนี้ครบทุกช่องและถูกต้องแล้วก็จะจบการทำงานของ Bot และจะส่งค่าของคำตอบนี้ออกทางหน้าจอของโปรแกรม

3.9 End

จบการทำงานของโปรแกรม

บทที่ 4

ผลการทดลอง

จากการทดลองโดยการลองให้โปรแกรมหาคำตอบของเกม Sudoku โดยกำหนดให้โจทย์มีช่องว่างให้ลง (Empty slot) ตั้งแต่ 0-81 แล้วจับเวลาว่าโปรแกรมใช้เวลาเฉลี่ยเท่าใดในการหาคำตอบบ้างซึ่งสรุปได้ดังตารางดังนี้

จำนวนของ Empty slot	ช่วงเวลารวมที่ใช้ในการหาคำตอบ (second)	เวลาเฉลี่ยที่ใช้ในการหาคำตอบ (second)
0	-	-
1	immediately	0
2	immediately	0
3	immediately	0
4	immediately	0
5	immediately	0
6	immediately	0
7	immediately	0
8	immediately	0
9	immediately	0
10	immediately	0
11	immediately	0
12	immediately	0
13	immediately	0
14	immediately	0
15	immediately	0
16	immediately	0
17	immediately	0
18	immediately	0
19	immediately	0
20	immediately	0
21	immediately	0
22	immediately	0
23	immediately	0
24	immediately	0
25	immediately	0
26	immediately	0
27	immediately	0
28	immediately	0
29	immediately	0
30	immediately	0
31	immediately	0
32	immediately	0
33	immediately	0

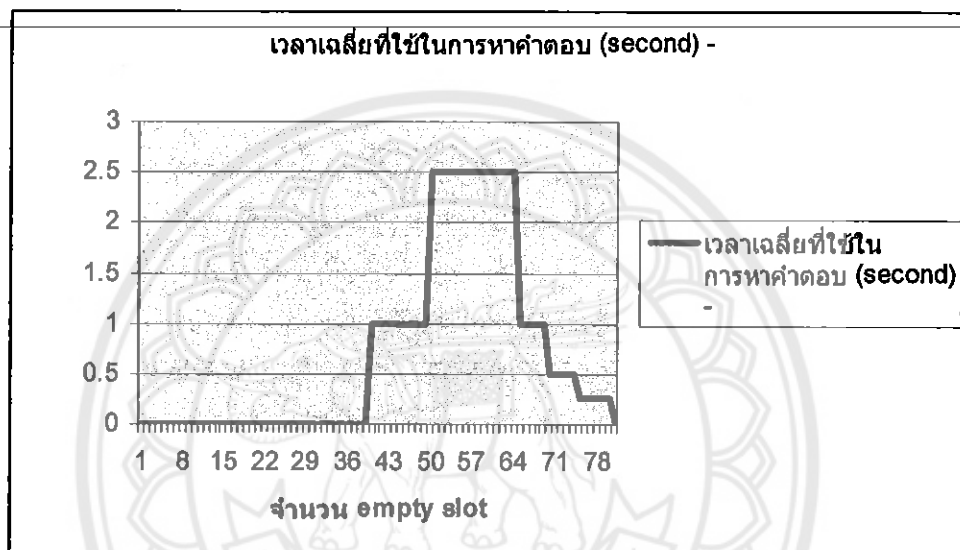
ตารางที่ 4.1 แสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบของเกม Sudoku

จำนวนช่อง Empty slot	ช่วงเวลาประมาณที่ใช้ในการหาคำตอบ (second)	เวลาเฉลี่ยที่ใช้ในการหาคำตอบ (second)
34	immediately	0
35	immediately	0
36	immediately	0
37	immediately	0
38	immediately	0
39	immediately	0
40	0-2	1
41	0-2	1
42	0-2	1
43	0-2	1
44	0-2	1
45	0-2	1
46	0-2	1
47	0-2	1
48	0-2	1
49	0-2	1
50	0-5	2.5
51	0-5	2.5
52	0-5	2.5
53	0-5	2.5
54	0-5	2.5
55	0-5	2.5
56	0-5	2.5
57	0-5	2.5
58	0-5	2.5
59	0-5	2.5
60	0-5	2.5
61	0-5	2.5
62	0-5	2.5
63	0-5	2.5
64	0-5	2.5
65	0-2	1
66	0-2	1
67	0-2	1
68	0-2	1
69	0-2	1
70	0-1	0.5
71	0-1	0.5
72	0-1	0.5
73	0-1	0.5
74	0-1	0.5
75	0-0.5	0.25
76	0-0.5	0.25
77	0-0.5	0.25
78	0-0.5	0.25

ตารางที่ 4.1 (ต่อ) แสดงเวลาเฉลี่ยที่ใช้โปรแกรมใช้ในการหาคำตอบของเกม Sudoku

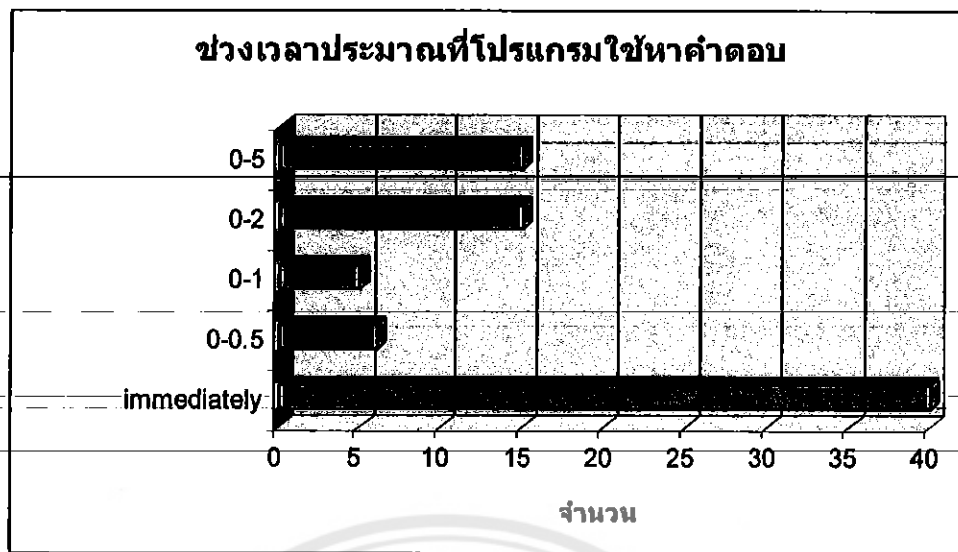
จำนวนของ Empty slot	ช่วงเวลาระมาณที่ใช้ในการหาคำตอบ (second)	เวลาเฉลี่ยที่ใช้ในการหาคำตอบ (second)
79	0-0.5	0.25
80	0-0.5	0.25
81	Immediately	0

ตารางที่ 4.1 (ต่อ) แสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบของเกม Sudoku



รูปที่ 4.1 แสดงกราฟของเวลาที่ใช้ในการหาคำตอบของ โปรแกรมต่อจำนวน Empty slot

จากกราฟในรูปที่ 4.1 จะแสดงเวลาเฉลี่ยที่โปรแกรมใช้ในการหาคำตอบโดยกำหนดให้ค่าของ Empty slot มีค่าตั้งแต่ 0-81 ซึ่งสรุปได้ว่า ในช่วงที่ Empty slot มีค่าตั้งแต่ 1-39 และ 70-81 นั้นโปรแกรมสามารถหาคำตอบออกมาได้ทันทีเนื่องจากช่องว่างที่ให้จะให้หาคำตอบนั้นเหลือน้อยจึงทำให้ค่าของตัวเลขที่สามารถลงได้ในช่องที่เหลือนั้นเหลือน้อยตามลงไปด้วย โปรแกรมจึงสามารถหาคำตอบได้อย่างรวดเร็ว และในช่วงที่ Empty slot มีค่าตั้งแต่ 40-69 จะเป็นช่วงที่โปรแกรมใช้เวลาในการหาคำตอบมากกว่าช่วงที่กล่าวในข้างต้นอยู่บ้าง เนื่องจากจำนวนของตัวเลขที่โจทย์กำหนดมาให้ในช่วงนี้จะมีประมาณครึ่งหนึ่งของจำนวนช่องทั้งหมดซึ่งจะทำให้ค่าของตัวเลขที่สามารถลงได้ในแต่ละช่องนั้นจะมีประมาณ 2-5 ค่า โดยจะทำให้การสุ่มหาคำตอบนั้นเป็นไปได้ยากและอาจจะต้องใช้เวลาในการหาคำตอบนานกว่าปกติ



รูปที่ 4.2 แสดงกราฟของช่วงเวลาประมาณที่โปรแกรมใช้หาคำตอบ

จากกราฟในรูปที่ 4.2 จะแสดงช่วงเวลาที่ใช้โปรแกรมใช้ในการหาคำตอบในทุกๆ Empty slot ซึ่งจะเห็นว่า โปรแกรมสามารถหาคำตอบได้ทันทีใน Empty slot 40 กรณีด้วยกัน คือ Empty slot ที่มีค่าตั้งแต่ 1-39 และ 81 ซึ่งโดยรวมแล้วถือว่าโปรแกรมสามารถหาคำตอบของเกมได้อย่างรวดเร็ว

บทที่ 5

สรุปผล

Sudoku คือ เกมปริศนาตารางตัวเลขที่มีกฎกติกาอยู่ด้วยกันหลายรูปแบบ ซึ่งในโครงการนี้จะเลือกแบบที่ปัจจุบันกำลังได้รับความนิยมอยู่เป็นอย่างมากมาศึกษาและพัฒนาโปรแกรมที่ใช้เล่นเกม Sudoku ซึ่งโปรแกรมที่พัฒนานั้นแยกได้เป็น 2 ส่วนคือ ส่วนตั้งปริศนา และส่วนหาคำตอบ โดยกฎและกติกาของเกม Sudoku ที่นำมาใช้ในโครงการนี้คือ

1. ผู้เล่นจะต้องเติมตัวเลขลงในช่องว่างของตาราง 9×9 ช่อง
2. ตัวเลขที่ใช้เติมในแต่ละช่องนั้นจะต้องใช้ตัวเลข 1 – 9 เท่านั้น
3. ในแถว, หลักและตารางย่อยขนาด 3×3 ใดๆจะต้องไม่มีตัวเลขที่ซ้ำกัน

ทฤษฎีที่นำมาใช้ในการหาคำตอบของเกม Sudoku คือ Genetic Algorithm ซึ่งเป็น Algorithm ที่ใช้ในการหา Solution โดยใช้หลักการเปลี่ยนแปลงพันธุกรรมทางชีววิทยา และ Backtracking Algorithm ซึ่งเป็นอีก Algorithm หนึ่งที่ใช้ในการหา Solution เช่นกัน โดย Backtracking Algorithm ที่นำมาใช้ในโครงการนี้จะ เป็นแบบ Depth-first-search

ขั้นตอนการทำงานของโปรแกรมจะเริ่มจากรับโจทย์เข้ามาแล้วจากนั้นก็มาพิจารณาว่าในแต่ละช่องที่ยังไม่ได้ลงตัวเลขลงไปนั้นแต่ละช่องสามารถลงตัวเลขอะไรได้บ้าง โดยจะนำตัวเลขจากโจทย์นั้นมาช่วยในการตัดตัวเลขที่ไม่สามารถลงได้ออกตามกฎของเกม Sudoku จากนั้นโปรแกรมก็จะเริ่มหาคำตอบ โดยจะสุ่มหาครั้งละแถว, หลัก หรือตารางย่อยขนาด 3×3 ที่มีจำนวนของตัวเลขที่สามารถลงได้น้อยที่สุดซึ่งจะใช้หลักการของ Genetic Algorithm และ Backtracking Algorithm ช่วยในการหาคำตอบ

5.1 สรุปผลการทดลอง

จากการทดลองสรุปได้ว่าระดับความยากของเกม Sudoku นั้นขึ้นอยู่กับหลายปัจจัยด้วยกันเช่น จำนวนของตัวเลขที่โจทย์กำหนดมาให้และความสัมพันธ์ของตัวเลขที่กำหนดมาให้ เป็นต้น ซึ่งถ้าเป็นโจทย์ที่มีระดับความยากในระดับที่ง่ายถึงปานกลางนั้น Bot จะใช้เวลาในการหาคำตอบได้อย่างรวดเร็ว แต่ถ้าเป็นคำตอบในระดับที่ยากนั้น Bot อาจจะใช้เวลาในการหาคำตอบนานกว่าเล็กน้อย และคำตอบของโจทย์บางโจทย์นั้นสามารถที่จะมีคำตอบได้มากกว่าหนึ่งคำตอบ ขึ้นอยู่กับการลงตัวเลขและตำแหน่งการลงของตัวเลขนั้นๆด้วย

5.2 ปัญหาและอุปสรรค

1. การเลือกใช้ Algorithm ที่จะนำมาประยุกต์ให้ Bot สามารถหาคำตอบได้อย่างถูกต้องและรวดเร็ว เนื่องจากการหาคำตอบของเกม Sudoku นั้นสามารถหาได้หลายวิธีคั้งนั้นจึงต้องลองผิดลองถูกจนได้ Algorithm ที่เหมาะสม ซึ่งในตอนแรกได้ใช้การสุ่มหาคำตอบทีละแถวโดยจะพบปัญหาที่ว่าในแถวท้ายๆจะไม่สามารถลงตัวเลขได้เนื่องจากตัวเลขที่ลงในแถวแรกๆนั้นไปลงในตำแหน่งที่ไม่เหมาะสมทำให้หาคำตอบได้ยาก ต่อมาจึงได้ทดลองใช้ Genetic Algorithm ซึ่งปัญหาที่พบคือ Bot จะไม่สามารถใช้ขั้นตอนการ Crossover และ Mutation ของ Genetic Algorithm ได้ เนื่องจากการ Crossover และ Mutation นั้นจะทำให้คำตอบเปลี่ยนแปลงไปมากซึ่งจากตัวเลขที่ลงถูกช่องอยู่แล้วอาจจะถูกเปลี่ยนไปลงในช่องที่ผิด ถึงแม้ว่า Genetic Algorithm จะมีโอกาสที่จะหาคำตอบของเกมได้แต่ว่าเวลาที่ใช้ในการหาคำตอบก็นานพอสมควรเหมือนกันและไม่มีหลักประกันว่าจะหาคำตอบได้หรือไม่ด้วย จากการลองผิดลองถูกในหลายๆ Algorithm ทำให้ได้ Algorithm ที่ใช้ในการหาคำตอบที่มีประสิทธิภาพซึ่งเกิดจากการผสมผสานระหว่างจุดเด่นและข้อดีของแต่ละ Algorithm โดยประกอบไปด้วย การเก็บเซตคำตอบเอาไว้ใน Chromosome และการหาค่าของ Fitness value ของ Genetic Algorithm และการย้อนกลับมาทำในขั้นตอนก่อนหน้านั้นเมื่อไม่สามารถหาคำตอบได้ต่อของ Backtracking Algorithm ซึ่งจุดเด่นของแต่ละ Algorithm ก็จะมาถูกช่องโหว่ของแต่ละ Algorithm ทำให้การหาคำตอบของ Bot มีประสิทธิภาพมากขึ้น

2. การหาค่า Fitness value เนื่องจากการที่โปรแกรมจะสามารถหาคำตอบได้ถูกต้องและรวดเร็วได้นั้นส่วนหนึ่งมาจากการหาค่า Fitness value ซึ่งถ้ามีหลักการในการหาค่า Fitness value ที่ดีก็จะทำให้สามารถหาคำตอบได้ถูกต้องและรวดเร็วยิ่งขึ้น ซึ่งการที่จะได้หลักการในการหาค่า Fitness value ที่ดีนั้นต้องผ่านการลองผิดลองถูกมาเป็นจำนวนมาก

5.3 ข้อเสนอแนะ

แนวคิดที่ใช้ในการหาคำตอบของ Bot ในโครงงานนี้สามารถนำไปประยุกต์ใช้ในการหาคำตอบของโจทย์อื่นได้อีกเนื่องจากหลักการที่ Bot ใช้นั้นเกิดจากการสุ่มหาจากเซตของคำตอบและมีการใช้ Fitness value เข้ามาช่วยในการพิจารณาว่าคำตอบที่ได้จากการสุ่มตัวไหนที่มีความเหมาะสมกับโจทย์มากที่สุด ซึ่งถ้าเราปรับปรุงการหาค่า Fitness value ให้ดีขึ้นก็จะสามารถหาคำตอบได้เร็วและดีขึ้นมาก

บรรณานุกรม

- [1] Stuart J. Russell and Peter Norvig. "Artificial Intelligence A Modern Approach Second Edition". United States of America : Pearson Education
- [2] ไม่ปรากฏชื่อผู้แต่ง. "ซูโดะกุ." [Online]. Available : <http://th.wikipedia.org/wiki/Sudoku>



ประวัติผู้เขียนโครงการ



ชื่อ นายสันต์ ภู่วโรดม
 ภูมิลำเนา 20 ถนนไสฤาไท อำเภอเมือง จังหวัดพิษณุโลก 65000
 ประวัติการศึกษา

- โรงเรียนพิษณุโลกพิทยาคม จังหวัดพิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail: ek_haru@hotmail.com

