ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ระบบคำนวณหาคำตอบของเกมซูโดกุโดยใช้อัลกอริทึ่ม จีโอเมทริกซ์พาร์ทิเคิล สวอร์มออพติไมซ์เซชัน

GEOMETRIC PARTICLE SWARM OPTIMIZATION FOR SUDOKU PUZZLE

H004860

โดย

อรพิณ โพธิกูล

ORAPIN PHOTHIGOON

อาจารย์ที่ปรึกษา

รศ.ดร. วรพจน์ กรีสุระเดช

QA76.
.03355
2550

เลขหมู่.....
เลขทะเบียน 04860
วัน,เดือน,ปี 9 ต.ค. 2551

b.11978600
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GEOMETRIC PARTICLE SWARM OPTIMIZATION FOR SUDOKU PUZZLE

ORAPIN PHOTHIGOON

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ระบบคำนวณหาคำตอบของเกมซูโดกุ โดยใช้อัลกอริทึ่มจีโอเมทริกซ์พาร์ทิเคิลสวอร์มออพติไมซ์เซชัน
นักศึกษา	นางสาวอรพิณ โพธิกูล
รหัสนักศึกษา	48066839
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	รศ.ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

เอกสารฉบับนี้เป็นการนำเสนอการประยุกต์ใช้งานอัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO) ซึ่งเป็นอัลกอริทึ่มที่พัฒนามาจาก อัลกอริทึ่ม Particle Swarm Optimization (PSO) หรือที่รู้จักกันในชื่อของ การหาค่าความเหมาะสมแบบกลุ่มอนุภาค ซึ่งเป็นเทคนิคที่ใช้ในการหาค่าจุดสูงสุดและต่ำสุดของฟังก์ชั่นชนิด Continuous Nonlinear Function โดยอัลกอริทึ่มดังกล่าวได้สร้างจากการเลียนแบบพฤติกรรมทางสังคมในการอพยพของฝูงนกเพื่อการหาอาหาร และอัลกอริทึ่มดังกล่าวได้ถูกนำมาประยุกต์ใช้งานในด้านการหาค่าที่เหมาะสมที่สุด (Optimization) ในงานด้านต่างๆ ซึ่งในเอกสารฉบับนี้ได้นำ Geometric Particle Swarm Optimization มาประยุกต์ใช้งานกับการหาคำตอบของเกมซูโดกุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Proposal of Geometric Particle Swarm Optimization for Sudoku Puzzle
Student	Ms. Orapin Phothigoon
Student ID.	48066839
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Assoc. Prof. Dr. Worapoj Kreesuradej

ABSTRACT

This paper presents an application using the Geometric Particle Swarm Optimization algorithm, which is developed from the Particle Swarm Optimization algorithm. The algorithm is the technique used to search for maximum and minimum value of a Continuous Non-Linear Function. It creates a model imitating a social behavior of a swarm immigrating for food. The algorithm has widely used to search for an optimization for many applications. In this paper, it has been applied to find solutions for the Sudoku Puzzle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำโครงงานพัฒนาระบบฉบับนี้สำเร็จได้อย่างดี เนื่องด้วยคำแนะนำ และคำปรึกษา จาก รศ. ดร. วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ IS.20.2 ในภาควิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้า ที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ทำให้ข้าพเจ้าสามารถพัฒนาโครงงานฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์อันพึงมาจากการพัฒนาโครงงานฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

อรพิณ โพธิกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ..... ..III

สารบัญ.....IV

สารบัญตาราง..... VI

สารบัญรูป..... VII

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... ..1

1.2 แรงจูงใจในการพัฒนาโครงงาน..... 2

1.3 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....2

1.4 ขอบเขตการวิจัย..... 2

1.5 เครื่องมือที่ใช้ในการพัฒนาโครงงาน.....3

1.6 การจัดแบ่งบทตอนรายงานโครงการวิจัย..... 3

บทที่ 2 ทฤษฎีพื้นฐาน..... 5

2.1 ทฤษฎีพื้นฐานของการหาค่าเหมาะสมแบบกลุ่มอนุภาค..... ...5

2.1.1 แนวคิดในการสร้างแบบจำลอง..... ...6

2.1.2 โครงสร้างของแบบจำลอง..... 7

2.1.3 อัลกอริทึมของการหาค่าเหมาะสมแบบกลุ่มอนุภาค..... 10

2.2 SELF ORGANIZING SWARM 11

2.2.1 SELF ORGANIZING MAP..... 11

2.2.2 SELF ORGANIZING SWARM 12

2.3 Geometric Particle Swarm Optimization 15

2.3.1 Introduction..... 15

2.3.2 การออกแบบ GPSO สำหรับ Sudoku Puzzle.....16

2.3.3 การทำงานของ GPSO อัลกอริทึม..... 17

2.3.4 การทดลอง..... ..18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4 Sudoku Puzzle 18

2.4.1 ประวัติ และความรู้พื้นฐานของ Sudoku.....18

2.4.2 เงื่อนไขในการเล่น Sudoku Puzzle.....19

2.4.3 วิธีการหาคำตอบของ Sudoku Puzzle.....20

บทที่ 3 การวิเคราะห์ระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม GPSO.....22

3.1 วิเคราะห์ความต้องการของระบบ.....22

3.2 การวิเคราะห์ความต้องการของระบบ(System Requirement):.....25

3.3 การวิเคราะห์การจัดเก็บข้อมูล.....25

3.4 การวิเคราะห์การทำงานของระบบ.....28

3.4.1 Domain Class Diagram ของระบบ.....28

3.4.2 Use Case Diagram ของระบบ.....29

3.4.3 Use Case Description.....30

3.4.4 UML Sequence Diagram 38

บทที่ 4 การออกแบบระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม GPSO.....45

4.1 การออกแบบสถาปัตยกรรมของระบบ.....45

4.2 การออกแบบข้อมูลของระบบ..... 46

4.3 การออกแบบ User Interface.....47

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....58

บรรณานุกรม.....59

ประวัติผู้เขียน.....60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงการจัดเก็บข้อมูลของระบบ.....26
3.2 รายละเอียดยูสเคส Start New Game.....30
3.3 รายละเอียดยูสเคส Select Game Level 31
3.4 รายละเอียดยูสเคส Save Game 31
3.5 รายละเอียดยูสเคส View Game Solution 32
3.6 รายละเอียดยูสเคส Using General Method 33
3.7 รายละเอียดยูสเคส Using GPSO 34
3.8 รายละเอียดยูสเคส Load Game 35
3.9 รายละเอียดยูสเคส View Score List 35
3.10 รายละเอียดยูสเคส Add New Problem 36
3.11 รายละเอียดยูสเคส Set Game Level 37
3.12 รายละเอียดยูสเคส View Report 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 แสดงตัวอย่างการหาจุดสูงสุดและจุดต่ำสุด 5

2.2 แสดงการหาแหล่งอาหารของฝูงนก.....7

2.3 แสดงถึงการ Share ข้อมูลกันระหว่างโหนดของ PSO 9

2.4 แสดง PSO Algorithm แบบพื้นฐาน.....10

2.5 แสดงการเรียนรู้แบบ Supervised Learning.....11

2.6 แสดงการเรียนรู้แบบ Unsupervised Learning..... 11

2.7 แสดงโครงสร้างของ Self – Organizing Map ของระบบโครงข่ายประสาทเทียม.....12

2.8 แสดง Pseudo-Code แสดงการประยุกต์ใช้ PSO กับ SOM 13

2.9 แสดง Self – Organizing Swarm..... 13

2.10 แสดงอัลกอริทึ่ม GPSO 19

2.11 แสดงแสดง Flowchart ของอัลกอริทึ่ม GPSO 20

2.12 แสดงตัวอย่าง Sudoku Puzzle..... 19

2.13 แสดงตัวอย่างโจทย์ของ Sudoku Puzzle..... 20

2.14 แสดงตัวอย่างการหาคำตอบของ Sudoku Puzzle.....21

3.1 แสดงการใส่เลขในแถวแนวนอนของ Sudoku Puzzle..... 22

3.2 แสดงการใส่เลขในแถวแนวตั้งของ Sudoku Puzzle.....23

3.3 แสดงการใส่เลขในตารางขนาด 3x3 ของ Sudoku Puzzle.....23

3.4 แสดงตัวอย่าง Sudoku Puzzle..... 24

3.5 แสดง ER Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle.....27

3.6 Domain Class Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle.....28

3.7 Use Case Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle..... 29

3.8 Sequence Diagram ของการเริ่มต้นเกมส์ใหม่..... 38

3.9 Sequence Diagram ของการบันทึกเกมส์.....39

3.10 Sequence Diagram ของการ Load เกมส์.....40

3.11 Sequence Diagram ของการดูสถิติเวลาในการเล่มเกมส์ที่ดีที่สุด.....41

3.12 Sequence Diagram ของการดูคำตอบของ Sudoku 42

3.13 Sequence Diagram ของการเพิ่มโจทย์ของ Sudoku เข้าไปในระบบ.....43

3.14 Sequence Diagram ของการดูรายงานการหาคำตอบของ Sudoku ด้วยวิธีต่างๆ..... 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.1 แสดงการออกแบบสถาปัตยกรรมของระบบ.....45

4.2 แสดงการออกแบบข้อมูลของระบบ.....46

4.3 แสดงการออกแบบหน้าจอหลักของระบบ.....47

4.4 แสดงการออกแบบหน้าจอและองค์ประกอบหลักของระบบ.....48

4.5 แสดงการออกแบบหน้าจอในส่วนของ File Menu 50

4.6 แสดงการออกแบบหน้าจอในส่วนของในส่วนของการเริ่มต้นเกมส์ใหม่.....51

4.7 แสดงการออกแบบหน้าจอในส่วนของการ Load เกมส์.....52

4.8 แสดงการออกแบบหน้าจอในส่วนของการเลือกไฟล์ที่ต้องการ Load.....52

4.9 แสดงการออกแบบหน้าจอในส่วนของการบันทึกข้อมูล.....53

4.10 แสดงการออกแบบหน้าจอในส่วนของ การใส่ชื่อไฟล์ที่ต้องการบันทึก.....53

4.11 แสดงการออกแบบหน้าจอในส่วนของการพิมพ์โจทย์ของ Sudoku.....54

4.12 แสดงการออกแบบหน้าจอในส่วนของ การเลือก Printer เพื่อพิมพ์โจทย์ของ Sudoku.....54

4.13 แสดงการออกแบบหน้าจอในส่วนของ Action Menu 55

4.14 แสดงการออกแบบหน้าจอในส่วนของ Admin Console Menu.....56

4.15 แสดงการออกแบบหน้าจอในส่วนของ ของการ Login ก่อนใช้งานในส่วนAdmin.....56

4.16 แสดงการออกแบบหน้าจอในส่วนของ Help Menu 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การพัฒนาเทคโนโลยีสารสนเทศในปัจจุบันมีแนวโน้มที่จะพัฒนาคอมพิวเตอร์ให้มีความสามารถใกล้เคียงกับมนุษย์ เช่น การเข้าใจภาษาสื่อสารของมนุษย์ ระบบโครงข่ายประสาทเทียม ระบบเสมือนจริง เป็นต้น โดยพยายามนำไปใช้ให้เกิดประโยชน์ในด้านต่างๆมากขึ้น และพยายามพัฒนาระบบดังกล่าวให้เกิดข้อผิดพลาดในการทำงานให้ลดน้อยลง

ระบบปัญญาประดิษฐ์ (Artificial Intelligence System) เป็นการพัฒนาระบบคอมพิวเตอร์ให้มีความสามารถในการตอบสนองกับความต้องการของมนุษย์ได้ ให้มีพฤติกรรมเลียนแบบมนุษย์ มีความเข้าใจภาษามนุษย์ รับรู้ได้และตอบสนองด้วยการแสดงออกทางพฤติกรรมและภาษามนุษย์ได้ รวมถึงความสามารถในการจดจำ และการวิเคราะห์ แก้ไข ปัญหาต่างๆได้

ปัญญาประดิษฐ์ ประกอบด้วยสาขาวิชาต่างๆ ได้แก่

1. ภาษาธรรมชาติ (Natural Language)
2. โครงข่ายประสาทเทียม (Artificial Neural Network)
3. ระบบผู้เชี่ยวชาญ (Expert System)
4. ศาสตร์ด้านหุ่นยนต์ (Robotics)

ภาษาธรรมชาติ (Natural Language) เป็นการประยุกต์ใช้ภาษาต่างๆเช่นภาษาไทยบนระบบคอมพิวเตอร์ โดยนำวิทยาการด้านคอมพิวเตอร์และเทคโนโลยีด้านการประมวลผลภาษาธรรมชาติมาพัฒนาโปรแกรม

โครงข่ายประสาทเทียม (Artificial Neural Network) เป็นการสร้างคอมพิวเตอร์โดยอาศัยการจำลองเอาวิธีการทำงานของสมองมนุษย์ หรือทำให้คอมพิวเตอร์รู้จักคิดและจดจำในแนวเดียวกับโครงข่ายประสาทของมนุษย์ เพื่อช่วยให้คอมพิวเตอร์ฟังภาษามนุษย์ได้เข้าใจ อ่านออก และรู้จำได้ ซึ่งอาจเรียกได้ว่าเป็น สมองกล

ระบบผู้เชี่ยวชาญ (Expert System) เป็นการพัฒนาระบบคอมพิวเตอร์โดยมีเป้าหมายให้คอมพิวเตอร์คิดเองได้ และมีความฉลาดแบบมนุษย์ที่เป็นผู้เชี่ยนชาญในด้านต่างๆ เช่น Doctor Expert System ซึ่งเป็นซอฟต์แวร์คอมพิวเตอร์ที่สามารถวิเคราะห์การเจ็บป่วยได้เหมือนกับการพบแพทย์ เพียงแค่ใส่อาการของการเจ็บป่วยเข้าไป ระบบจะสามารถวิเคราะห์ออกมาได้ว่าเป็นอาการป่วยของโรคอะไร เป็นต้น

ศาสตร์ด้านหุ่นยนต์ (Robotics) เป็นการนำคอมพิวเตอร์มาพัฒนาให้สามารถทำงานร่วมกับเครื่องจักรกลได้ โดยสามารถควบคุมการทำงานของเครื่องจักรกลผ่านทางโปรแกรมคำสั่งต่างๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสำคัญดังกล่าวจึงทำให้เกิดแนวคิดในการพัฒนาโครงงานระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization โดยการพัฒนาโครงงานจะใช้เทคนิคด้านปัญญาประดิษฐ์ในด้านต่างๆตามที่กล่าวมาแล้วมาช่วยในการพัฒนาโครงงาน

1.2 แรงจูงใจในการพัฒนาโครงการ

แรงจูงใจในการทำโครงการคือ เพื่อเป็นการศึกษาถึงกฎเกณฑ์และหลักการใหม่ๆ ในการปรับปรุงและพัฒนาประสิทธิภาพของการสร้างระบบคอมพิวเตอร์ที่มีความสามารถในการเรียนรู้ด้วยตัวเอง (Machine Learning) และสามารถคำนวณเพื่อหาคำตอบและแก้ปัญหาได้อย่างชาญฉลาดเหมือนกันความคิดของมนุษย์ และพัฒนาให้ระบบดังกล่าวทำงานได้อย่างมีประสิทธิภาพดียิ่งขึ้น

โดยผู้จัดทำมีความสนใจที่จะนำเอาหลักการของ Geometric Particle Swarm Optimization (GPSO) ซึ่งเป็นหลักการที่มาจากการประยุกต์ทฤษฎีการหาค่าเหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) นำมาใช้ในการพัฒนาระบบคำนวณเพื่อหาคำตอบของ Sudoku Puzzle ซึ่งผู้จัดทำหวังว่าจะสามารถนำเอาหลักการของ GPSO มาใช้ในการหาคำตอบที่ถูกต้องของ Sudoku Puzzle ได้อย่างมีประสิทธิภาพ

1.3 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์หลักของโครงการคือ การพัฒนาระบบคอมพิวเตอร์ที่มีความสามารถในการเรียนรู้ด้วยตัวเอง (Machine Learning) และสามารถคิดเพื่อหาคำตอบและแก้ไขปัญหาได้อย่างชาญฉลาดเหมือนกับความคิดของมนุษย์ โดยในการพัฒนาโครงงานจะใช้อัลกอริทึม Geometric Particle Swarm Optimization (GPSO) มาใช้ในการคำนวณหาคำตอบของ Sudoku Puzzle ขนาด 9x 9 ซึ่งรายละเอียดของวัตถุประสงค์ในการพัฒนาระบบ ประกอบด้วย

1. เพื่อทำการศึกษาการสร้างระบบคอมพิวเตอร์ที่มีความสามารถในการเรียนรู้ด้วยตัวเอง (Machine Learning)
2. เพื่อทำการศึกษาทฤษฎีเรื่อง Geometric Particle Swarm Optimization (GPSO) และนำหลักการของ GPSO มาใช้ในการพัฒนาระบบคำนวณเพื่อหาคำตอบของ Sudoku Puzzle
3. เพื่อศึกษารายละเอียดของการสร้างโจทย์ และกระบวนการในการหาคำตอบของ Sudoku Puzzle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการวิจัย

ในการทำโครงงานฉบับนี้จะนำเสนอการพัฒนาระบบเกี่ยวกับการใช้ใช้อัลกอริทึม Geometric Particle Swarm Optimization (GPSO) มาใช้ในการคำนวณหาคำตอบของ Sudoku Puzzle ขนาด 9x 9 โดยมีจุดมุ่งหมายในการพัฒนาระบบคอมพิวเตอร์ให้มีความสามารถในการเรียนรู้ด้วยตัวเอง (Machine Learning) ได้ และสามารถคิดเพื่อหาคำตอบและแก้ไขปัญหาได้อย่างชาญฉลาดเหมือนกับความคิดของมนุษย์

1.5 เครื่องมือที่ใช้ในการพัฒนาโครงงาน

ระบบที่พัฒนาขึ้นจะอยู่ในรูปแบบของ Application Program โดยใช้เครื่องมือต่างๆในการพัฒนาระบบดังต่อไปนี้

1. ใช้ระบบปฏิบัติการ Windows XP ในการพัฒนาโครงงาน
2. ใช้โปรแกรม Microsoft Visual Studio .Net เป็น Tools ช่วยในการพัฒนาระบบ
3. โปรแกรมภาษาที่ใช้ในการพัฒนาโครงงานใช้ภาษา C#
4. การเก็บข้อมูลที่ใช้ในการพัฒนาระบบจะจัดเก็บในรูปแบบของ Text File

1.6 การจัดแบ่งบทตอนรายงานโครงการวิจัย

รูปแบบของการแบ่งบทตอนของรางงาน ประกอบด้วย

บทที่1 บทนำ

รายละเอียดในบทนี้ จะกล่าวถึงความเป็นมาของโครงงานวิจัย สาเหตุและแรงจูงใจในการพัฒนาโครงงานวิจัย ปัญหาและความสำคัญของปัญหา วัตถุประสงค์ของโครงงานวิจัย และขอบเขตในการพัฒนาโครงงาน

บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง

รายละเอียดในบทนี้ จะกล่าวถึง ทฤษฎีที่ก่อให้เกิดแนวคิดในการทำโครงการวิจัย ซึ่งประกอบด้วย ทฤษฎีพื้นฐานของการหาค่าความเหมาะสมแบบกลุ่มอนุภาค Particle Swarm Optimization (PSO) ทฤษฎีของ Geometric Particle Swarm Optimization (GPSO) และทฤษฎีของ Suduku Puzzle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การวิเคราะห์ระบบ

รายละเอียดในบทนี้ จะกล่าวถึง การศึกษาความต้องการของระบบ ได้แก่ Functional Requirement และ System Requirement รวมถึงการวิเคราะห์ข้อมูลของระบบ ซึ่งจะใช้ ER Diagram ในการวิเคราะห์ข้อมูลของระบบ และใช้ Use Case Diagram Use Case Description และ Sequence Diagram ในการวิเคราะห์การทำงานของระบบ

บทที่ 4 การออกแบบระบบ

การวิเคราะห์และออกแบบระบบซึ่งจะประกอบด้วย การออกแบบสถาปัตยกรรมของระบบ การออกแบบข้อมูลของระบบ โดยใช้ Design Class Diagram และการออกแบบหน้าจอของระบบ

บทที่ 5 บทสรุป

รายละเอียดในบทนี้ จะกล่าวถึง บทสรุปของการทำโครงการ พร้อมกับข้อเสนอแนะในการพัฒนาโครงการ และแนวทางในการพัฒนาโครงการต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

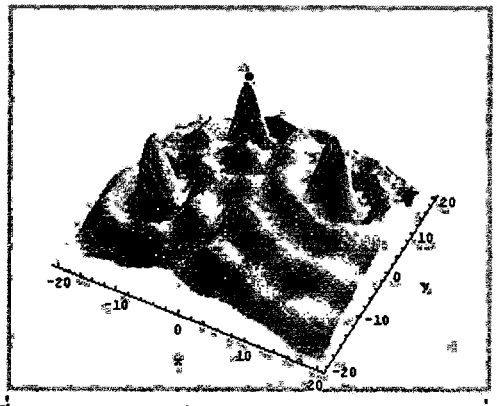
บทที่ 2

ทฤษฎี Geometric Particle Swarm Optimization

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการพัฒนาระบบคำนวณเพื่อหาคำตอบของ Sudoku Puzzle ซึ่งเนื้อหาในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานของการหาค่าเหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) และ Geometric Particle Swarm Optimization (GPSO) ซึ่งเป็นอัลกอริทึมที่ใช้ในการไขปัญหาดังกล่าว รวมถึงกฎข้อบังคับต่างๆของ Sudoku Puzzle

2.1 ทฤษฎีพื้นฐานของการหาค่าเหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization)

ในปัจจุบันการแก้ไขปัญหาต่างๆที่พบในการปฏิบัติงานไม่ว่าจะเป็นในงานทางวิชาการ หรือในโรงงานอุตสาหกรรม พบว่าการดำเนินงานหรือกิจกรรมต่างๆ นั้นมีวัตถุประสงค์เพื่อการหาค่าที่เหมาะสมที่สุด (Optimization) เป็นวัตถุประสงค์หลัก ตัวอย่างเช่น ในโรงงานผลิตไฟฟ้าต้องการหาจุดที่เครื่องกำเนิดไฟฟ้าให้กำเนิดพลังงานไฟฟ้าได้สูงที่สุด และในขณะเดียวกันก็ต้องการให้เครื่องผลิตไฟฟ้านั้นสูญเสียพลังงานน้อยที่สุด เป็นต้น โดยส่วนมากการแก้ไขปัญหาดังกล่าวจะใช้การวิเคราะห์ทางคณิตศาสตร์เป็นหลัก แต่ในการแก้ไขปัญหาบางอย่างการใช้การวิเคราะห์ทางคณิตศาสตร์อย่างเดียวอาจไม่เพียงพอ จึงเกิดวิธีการใหม่ขึ้นมาสำหรับแก้ไขปัญหาดังกล่าว ตัวอย่างเช่น Genetic Algorithm เป็นวิธีการหนึ่งซึ่งมีพื้นฐานของการหาค่าที่เหมาะสมโดยอาศัยพฤติกรรมการวิวัฒนาการทางพันธุกรรมในการถ่ายทอดลักษณะต่างๆไปยังลูกหลาน หรือการหาค่าความเหมาะสมแบบกลุมอนุภาค (Particle Swarm Optimization) เป็นอีกวิธีการหนึ่งที่ใช้สำหรับการหาค่าที่เหมาะสมแบบกลุ่มอนุภาคโดยเลียนแบบพฤติกรรมทางสังคมในการอพยพของฝูงนกเพื่อการหาอาหารเป็นต้น



รูปที่ 2.1 แสดงตัวอย่างการหาจุดสูงสุดและจุดต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่าความเหมาะสมแบบกลุมอนุภาค (Particle Swarm Optimization) หรือเรียกสั้นๆว่า PSO ถูกคิดค้นโดย James Kennedy และ Russell Eberhart ในปี ค.ศ. 1995 โดยมีการใช้วิธีนี้ในการหาค่าที่เหมาะสมแบบกลุ่มกับข้อมูลชนิด Continuous Nonlinear Function ซึ่งผู้คิดค้นได้แรงจูงใจมาจากพฤติกรรมทางสังคมในการอพยพของฝูงนก และได้ทดลองโครงสร้างผ่านทางการทำแบบจำลองของพฤติกรรมดังกล่าว แต่อย่างไรก็ตามการหาค่าที่เหมาะสมที่สุดแบบสุ่ม (Stochastic) ยังคงต้องอาศัยพื้นฐานของความน่าจะเป็น

2.1.1 แนวคิดในการสร้างแบบจำลองพฤติกรรมทางสังคมการอพยพของฝูงนกเพื่อการหาแหล่งอาหาร

มีนักวิชาการส่วนมากที่สร้างแบบจำลองต่างๆด้วยคอมพิวเตอร์เพื่อให้สอดคล้องกับพฤติกรรมทางสังคมในการอพยพไปตามสถานที่ต่างๆเพื่อวัตถุประสงค์ในการหาแหล่งอาหารของฝูงนก เช่น Reynolds และ Grenander ได้มีการนำเสนอ Simulation เกี่ยวกับการอพยพของฝูงนกเพื่อค้นหาแหล่งอาหาร ภายใต้กฎข้อบังคับบางอย่าง เช่นการเปลี่ยนทิศทางการบินของฝูงนกที่ประกอบไปด้วยสมาชิกจำนวนมากที่พร้อมใจกันเปลี่ยนทิศทางไปพร้อมกันอย่างฉับพลัน หรือ การกระจายตัวของฝูงนกและมีการกลับมารวมกลุ่มกันใหม่ เป็นต้น ซึ่งนักวิทยาศาสตร์ทั้งสองคนได้สนใจเกี่ยวกับพฤติกรรมของฝูงนกที่ไม่สามารถคาดการได้ล่วงหน้าซึ่งคล้ายกับพฤติกรรมของสัตว์ทั่วๆไปนั่นเอง โดยอาศัยข้อมูลจากสมาคมนักชีววิทยาที่ได้มีการเผยแพร่ทฤษฎีเกี่ยวกับการหาอาหารของฝูงปลา โดยมีการอ้างถึงข้อมูลที่ว่า สมาชิกในฝูงปลาจะได้ประโยชน์จากประสบการณ์การหาอาหารของสมาชิกตัวอื่นในฝูง ซึ่งแสดงว่าภายในฝูงปลานั้นมีการใช้ข้อมูลร่วมกันและจากการค้นพบดังกล่าวทำให้นำมาสู่การตัดสินใจในการค้นคว้าทฤษฎีการหาอาหารของฝูงนก

ผู้คิดค้นวิธีนี้ได้สังเกตพฤติกรรมทางสังคมของฝูงนก พบว่าประชากรของฝูงนกจะอยู่กันเป็นกลุ่มรอบๆที่พักผ่อน เช่น ตามต้นไม้ สายไฟฟ้า และสายโทรศัพท์ เป็นต้น แต่เมื่อฝูงนกดังกล่าวออกไปค้นหาแหล่งอาหาร ถ้าบริเวรนั้นมีแหล่งอาหารเพียงที่เดียวฝูงนกดังกล่าวจะรู้ได้อย่างไรว่าแหล่งอาหารนั้นอยู่ที่ไหน วิธีการคือ นกทั้งฝูงจะทำการบินเพื่อการค้นหาแหล่งอาหาร เมื่อผ่านการค้นหาในแต่ละรอบถึงจะรู้ว่ามีสมาชิกในฝูง (Agent) ตัวใดที่อยู่ใกล้แหล่งอาหารมากที่สุด หลังจากนั้นสมาชิกในฝูงนกตัวอื่นๆก็จะทำการบินตามนกตัวดังกล่าวไปยังแหล่งอาหารได้ ที่เป็นเช่นนี้ได้เพราะสมาชิกในฝูงนกนั้นมีการใช้ข้อมูลร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่2.2 แสดงการหาแหล่งอาหารของฝูงนก

ผู้คิดค้นได้อ้างว่าการสร้างแบบจำลองดังกล่าวนั้นสามารถ Implement ได้โดยไม่ยาก โดย Operation ของ Particle Swarm Optimization นั้นใช้การคำนวณพื้นฐานทางคณิตศาสตร์โดยอ้างอิงหลักการความน่าจะเป็น และในการประมวลผลดังกล่าวก็ใช้ทรัพยากรทางด้านคอมพิวเตอร์ (เช่น หน่วยประมวลผล และ หน่วยความจำเป็นต้น) ไม่มากนักโดยในตอนเริ่มต้นผู้คิดค้นได้ใช้ Algorithm ของวิธีการนี้เพื่อการ Train ค่า Weight สำหรับงานวิจัยเรื่อง Neural Network

2.1.2 โครงสร้างของแบบจำลองพฤติกรรมทางสังคมการอพยพของฝูงนกเพื่อการหาแหล่งอาหาร

หลักการของ Particle Swarm Optimization นั้นใช้พื้นฐานจากการหาค่าเหมาะสมที่สุดแบบสุ่ม (Stochastic) โดยมีแรงจูงใจมาจากพฤติกรรมทางสังคมของฝูงนกเพื่อการหาแหล่งอาหาร

ในการหาค่าเหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) นั้นคำตอบที่เป็นไปได้จะถูกกำกับด้วยค่าที่เหมาะสมค่าหนึ่งซึ่งเรียกว่า Fitness Value ส่วนบริเวณที่ฝูงนกใช้ในการหาแหล่งอาหารแต่ละรอบเรียกว่า Particle โดย Particle Swarm Optimization Algorithm จะประกอบด้วยตัวแปรต่างๆดังต่อไปนี้

- บริเวณในการค้นหารแหล่งอาหารของนกแต่ละตัวเรียกว่า Particle
- ค่าความเหมาะสมที่ใช้กำกับในการหาคำตอบเรียกว่า Fitness Value
- ค่าความเร็วในการเคลื่อนที่ของนกแต่และตัวเรียกว่า Velocity
- ตำแหน่งของนกแต่ละตัวเรียกว่า Position ซึ่งประกอบไปด้วยตำแหน่งในแกน x และตำแหน่งในแกน y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตอนเริ่มต้นการค้นหาแหล่งอาหารของฝูงนกนั้นมีการตั้งสมมติฐานว่าแหล่งอาหารมีเพียงแหล่งเดียวแล้วสมาชิกของนกในฝูงนั้นไม่มีสมาชิกตัวใดเลยที่รู้ตำแหน่งของแหล่งอาหารในตอนเริ่มต้น เมื่อผ่านการค้นหาแหล่งอาหารในแต่ละรอบ (Particle) แล้วจึงมีสมาชิกตัวใดตัวหนึ่งของฝูงนกที่รู้ตำแหน่งของแหล่งอาหาร โดยสมาชิกตัวอื่นมีแนวโน้มที่จะติดตาม Particle สมาชิกของฝูงนกที่มีข้อมูลที่ดีเพื่อไปยังแหล่งอาหารดังกล่าว

ขั้นตอนของการทำ PSO

ขั้นตอนที่ 1 กำหนดค่าเริ่มต้นของ Particle

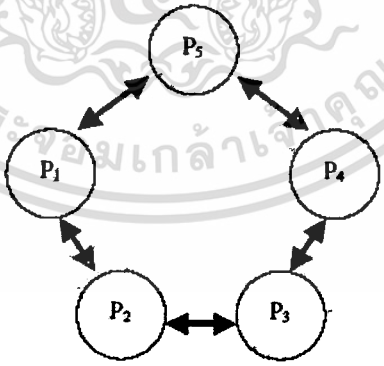
ขั้นตอนที่ 2 ทำการคำนวณหาค่าที่เหมาะสม (Fitness Value) ในแต่และ particle โดยค่าที่เหมาะสมดังกล่าวนี้จะมีผลต่อความเร็วในการลู่เข้าสู่คำตอบ (เป้าหมาย) โดยจะมีการคำนวณค่าเหมาะสมจากค่าที่ดีของ particle ก่อนหน้านี้

ขั้นตอนที่ 3 นำค่า Fitness Value ไปเปรียบเทียบกับค่า pbest ถ้า Fitness Value ดีกว่าค่า pbest เดิม กำหนดค่า Fitness Value ให้เป็น pbest ค่าใหม่

ขั้นตอนที่ 4 ทำการเลือก particle ที่มีที่มีค่า Fitness Value ที่ดีที่สุดนั่นคือค่า gbest

ขั้นตอนที่ 5 ทำการเปรียบเทียบค่า pbest กับค่า gbest ถ้าค่า pbest ดีกว่าค่า gbest กำหนดให้ค่า pbest เป็นค่า gbest ค่าใหม่

ขั้นตอนที่ 6 นำค่าตัวแปรต่างๆ ไปคำนวณหาค่าความเร็วในการลู่เข้าสู่เป้าหมาย (Velocity)



รูปที่2.3 แสดงถึงการ Share ข้อมูลกันระหว่างโหนดของ PSO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v_i(t+1) = w * v_i(t) + (c_1 * r_1 * (p_{best} - x_i(t))) + (c_2 * r_2 * (g_{best} - x_i(t))) \quad (2.1)$$

$$W = wmax - ((wmax - wmin) / itermax) * iter \quad (2.2)$$

จากสมการที่ 1 pbest เป็นตำแห่งที่ดีที่สุดใน Particle นั้นๆ ส่วนค่า gbest เป็นตำแหน่งที่ดีที่สุดของ Particle ทั้งหมด ค่า c1 และ c2 เป็น Weight ที่ใช้ในการคำนวณหา pbest ส่วนค่าความเร็ว (Velocity) ในสมาการแทนด้วย xi โดย i คือค่า location ในปัจจุบัน ส่วนค่า r1 และ r2 เป็นค่า random ซึ่งมีค่าอยู่ระหว่าง 0 ถึง 1 และค่า w เป็นค่าสัมประสิทธิ์ Momentum

จากสมการที่ 2 เป็นสมาการที่ใช้ในการหาค่า w เป็นค่าสัมประสิทธิ์ Momentum และค่า itermax และ iter เป็นจำนวนของการวนรอบ ส่วนค่า wmax และ wmin เป็นค่าสูงสุดและต่ำสุดของ w ส่วนการ Update ค่าความเร็ว xi ในครั้งแรกจะใช้สมการที่ 3 และค่าความเร็วของ pbest จะใช้สมการที่ 4 และสมการที่ 5 ในการ Update

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.3)$$

$$y_i(t+1) = y_i(t) \text{ if, } f(x_i(t)) \leq f(y_i(t)) \quad (2.4)$$

$$y_i(t+1) = x_i(t) \text{ if, } f(x_i(t)) > f(y_i(t)) \quad (2.5)$$

หลังจากที่ตำแหน่งของ Particle ทั้งหมดมีการ Update แล้ว จะมีการตรวจสอบเพื่อหาค่า gbest จากสมการที่ 6

$$y \in (y_0,\ldots,y_{n-1}) \mid f(y) = \max(f(y_0),\ldots,f(y_{n-1})) \quad (2.6)$$

ในแต่ละรอบของการทำงาน จะมีการเก็บค่าสถิติเอาไว้ เพื่อใช้ในการหา Gbest (Global position)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 อัลกอริทึมของการหาค่าเหมาะสมแบบกลุ่มอนุภาค

รูปที่ 2-4 แสดงตัวอย่างการทำงานของ PSO Algorithm แบบพื้นฐาน ซึ่งแสดงการทำงานด้วย Pseudo Code

```
Initialize parameters
Initialize population
Evaluate
Do
    {
        Find pbest
        Find gbest
        Update velocity
        Update position
        Evaluate
    }
While (Termination)
```

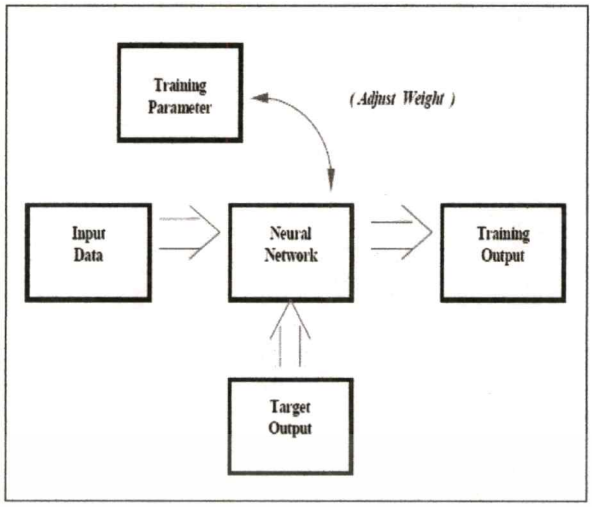
รูปที่2.4 แสดง PSO Algorithm แบบพื้นฐาน

2.2 SELF- ORGANIZING SWARM

2.2.1 SELF- ORGANIZING MAP

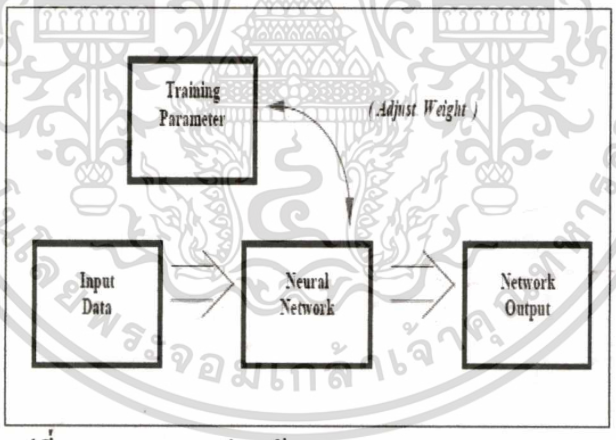
การเรียนรู้ในระบบ Neural Network มี 2 ลักษณะคือ Supervised Learning และ Unsupervised Learning Supervised Learning การเรียนรู้แบบมีการสอนเป็นการเรียนแบบที่มีการตรวจผลลัพธ์ที่ได้ว่ามีความคลาดเคลื่อนไปจากผลลัพธ์ที่ต้องการมากน้อยเพียงใด เพื่อใช้ในการปรับชุดข้อมูลที่ใช้ Training ให้แก่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.5 แสดงการเรียนรู้แบบ Supervised Learning

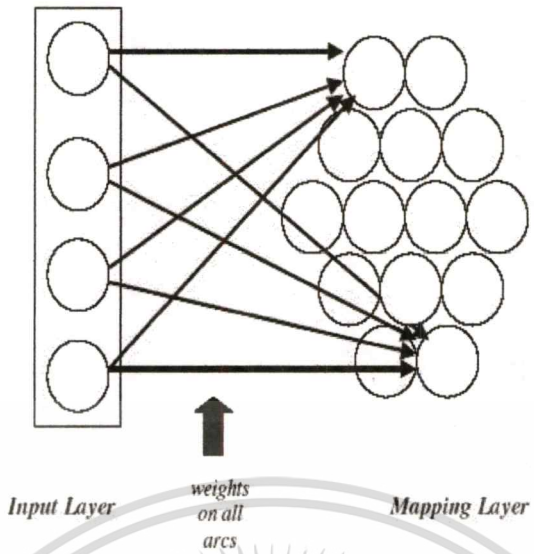
Unsupervised Learning การเรียนรู้แบบไม่มีผู้แนะนำ พร้อมทั้งไม่มีผลลัพธ์ที่ต้องการไว้สำหรับตรวจว่าผลลัพธ์ที่ได้นั้นถูกหรือผิด ระบบจะมีการจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล ผลลัพธ์ที่ได้ ระบบจะสามารถจัดหมวดหมู่ของข้อมูลได้



รูปที่ 2.6 แสดงการเรียนรู้แบบ Unsupervised Learning

Self – Organizing Map (SOM) เป็นวิธีการที่ใช้ในการจัดกลุ่มข้อมูลในระบบโครงข่ายประสาทเทียม (Neural Network) แบบการเรียนรู้ด้วยตัวเอง โดยไม่ต้องมีผู้แนะนำ (Unsupervised Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.7 แสดงโครงสร้างของ Self – Organizing Map ของระบบโครงข่ายประสาทเทียม

ข้อมูลที่ประมวลผลในระบบจะถูกส่งไปในทิศทางเดียวจาก Input nodes ส่งต่อมาเรื่อย ๆ จนถึง output nodes โดยไม่มีการย้อนกลับของข้อมูล หรือแม้แต่ Nodes ใน layer เดียวกันก็ไม่มีการเชื่อมต่อกัน วัตถุประสงค์ของ Self – Organizing Map คือการจัดกลุ่มข้อมูล Input ที่เหมือนกัน ให้อยู่บน Node เดียวกันใน Mapping Layer ในกรณีที่ Input Space มีความใกล้เคียงกันก็ทำให้ผลลัพธ์ที่ได้ใน Mapping Space ใกล้เคียงกันด้วยเช่นกัน

2.2.2 SELF- ORGANIZING SWARM

Self – Organizing Swarm เป็นการประยุกต์ใช้งาน PSO อัลกอริทึมกับ Self – Organizing Map ในระบบโครงข่ายประสาทเทียม โดยนำ PSO อัลกอริทึมมาใช้ในการปรับค่า Weight ของระบบ โดย Self – Organizing Swarm จะนำ PSO อัลกอริทึมมาใช้ในการปรับค่า weight ของแต่ละ Connection ให้เหมาะสมเพื่อทำการจัดกลุ่มข้อมูล Input โดยกลุ่มข้อมูลที่มีลักษณะเดียวกัน Self – Organizing Swarm (SoSwarm) จะมีลักษณะคล้ายคลึงกันกับการนำ Self – Organizing Map (SOM) มาใช้ในการทำ Mapping ซึ่งเป็นการ Mapping ในลักษณะแบบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการทำงานแทนที่จะมีการปรับค่า Vector ใน Map Space ตามค่า Input เพียงอย่างเดียวแต่ในกรณีนี้จะมีการใช้ PSO ในส่วนของฟังก์ชันการ Update สำหรับ Training ให้กับ SOM และในแต่ละรอบของการทำงานจะมีการนำข้อมูลของเพื่อนบ้านโหนดที่อยู่ใกล้เคียงมาใช้ในการคำนวณ ซึ่งทำให้เกิดการเรียนรู้และการปรับตัว โดยมีการคำนึงถึงค่าของตัวเอง ค่าของเพื่อนบ้าน และค่าของกลุ่มโดยรวม ทำให้การคำนวณในแต่ละรอบ จะทำให้ได้ค่าที่สัมพันธ์กับกลุ่ม

```
0   initialize particles in mapping layer randomly

1   for( max number of iterations )
2      for( each input training vector in turn )

3          set lbest to the value of the input vector

4          set pbest of each particle to be its
           current position

5          find particle with closest match to lbest

6          denote this particle as the firing particle

7          update firing particle's velocity
           and position vectors

8          update velocity and location vectors
           of neighbors of firing particle

9       endfor
10  endfor

11  assign class to each particle using training data

12  calculate classification accuracy using test data
```

รูปที่ 2.8 แสดง Pseudo-Code แสดงการประยุกต์ใช้ PSO กับ SOM

พิจารณาถึง Firing Particle (คือ Particle ที่มีค่า Match กับค่า Input มากที่สุด)

$$Firing\ particle = \underset{i}{argmin} \left\{ \sqrt{\sum_{j=1}^{d} (V_j - P_{ij})^2} \right\} \tag{2.7}$$

V: Correspond to the input vector

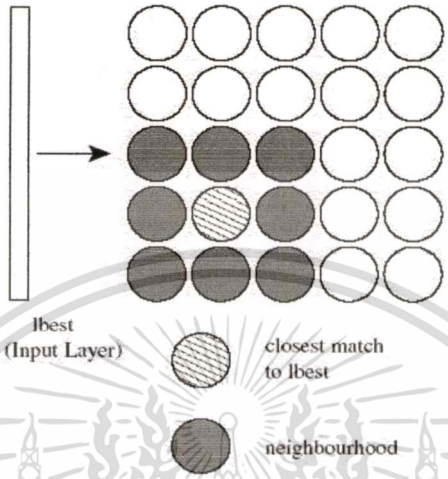
Pi : ตำแหน่งของ Particle ที่ iใดๆ

i : Particle

รูปที่ 2-9 แสดงถึงการประยุกต์ใช้งาน SoSwarm ในลักษณะการ Mapping ใน 2 มิติ ในแต่ละรอบการทำงานจะต้องมีการกำหนด neighborhood สำหรับ Input ก่อนถึงสามารถคำนวณหา Firing Particle ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Firing Particle เป็น Particle ที่ Match กับ Input มากที่สุดในที่นี้แทนค่าด้วย lbest จะมีการปรับค่า Position ของ Vector ในแต่ละ Particle โดยใช่สมการที่ 1 ถึงสมการที่ 5 โดยจะต้องมีการกำหนด Node ที่อยู่รอบๆตัวก่อน จากนั้นถึงจะมีการปรับปรุง Position โดยใช้สมการขอ PSO เช่นกัน



รูปที่2.9 แสดง Self – Organizing Swarm

เริ่มต้นโดยการ train ค่าให้กับแต่ละโหนดใน Mapping Layer โดยในส่วนของการแบ่งกลุ่มข้อมูลที่เป็น Input ไปยังโหนดบน Mapping Layer จะใช้วิธีการหา Distance ระหว่างข้อมูล Input และ Node บน Mapping Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Geometric Particle Swarm Optimization (GPSO)

2.3.1 Introduction

การนำ Particle Swarm Optimization (PSO) อัลกอริทึมแบบดั้งเดิมไปดัดแปลง ให้มีการประยุกต์ใช้งานกับการ Search ในลักษณะ Continuous Search Space โดยต่อมาได้มีการนำ PSO ไปพัฒนาต่อกันอย่างแพร่หลาย โดยการพัฒนาต่อนั้นสามารถทำได้ 2 วิธีด้วยกันคือ

1. ดัดแปลงแล้วนำเสนอในรูปแบบใหม่ที่ต่างจากของเดิม
2. ใช้คณิตศาสตร์เป็นตัวช่วยในการพัฒนาอัลกอริทึมซึ่งอัลกอริทึมดังกล่าวสามารถใช้ในการอธิบายกฎต่างๆได้

ในเอกสารฉบับนี้จะมีการแสดง PSO อัลกอริทึมให้อยู่ในรูปแบบที่เป็น Formal โดยใช้ทฤษฎีทางคณิตศาสตร์ทำให้ PSO อยู่ในรูปแบบที่ต่างจาก Search อัลกอริทึมได้ โดยใช้ Geometric Crossover และ Geometric Mutation

ในการหาคำตอบของ Sudoku Puzzle จะต้องใช้หลักการของการแทนที่ (Placement Algorithm) และเมื่อทำการใส่เลขที่ต้องการแล้วจะต้องมีการตรวจสอบว่าเลขที่ใส่เข้าไปนั้นถูกต้องตามกฎของ Sudoku หรือไม่ซึ่งรายละเอียดของ Sudoku Puzzle จะกล่าวอยู่ในหัวข้อที่ 2.4

การที่จะทำให้คอมพิวเตอร์สามารถหาคำตอบของ Sudoku นั้น วิธีที่ง่ายที่สุดคือการใช้วิธีการ Brute Force ในการหาคำตอบ โดยการใช้วิธีนี้จะมีผลต่อประสิทธิภาพของโปรแกรมเพราะวิธีนี้จะทำให้ประสิทธิภาพการทำงานของโปรแกรมต่ำ นอกจากนั้นยังมีการใช้ Graph Coloring Problem มาช่วยในการหาคำตอบของ Sudoku ด้วย โดยจะมีการกำหนดให้สีของกราฟประกอบไปด้วยสีจำนวน 9 สี

จึงได้มีการใช้ Geometric Framework ในการออกแบบและพัฒนา Evolutionary อัลกอริทึม เพื่อใช้ในการ Solve ปัญหาของ Sudoku Puzzle ซึ่งต่อมาได้ถูกเรียกว่า Geometric Particle Swarm Optimization (GPSO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การออกแบบ GPSO สำหรับ Sudoku Puzzle

ในการออกแบบ GPSO สำหรับ Sudoku Puzzle นั้นต้องมีการคำนึงถึง ข้อบังคับของ Sudoku ประกอบด้วย

1. ต้องมีการกำหนดค่าและตำแหน่ง ของโจทย์ให้มีค่าคงที่ในตำแหน่งนั้นเสมอ
2. เปลี่ยนแปลงค่าใน แถว หลัก และ ตารางย่อย ขนาด 3x3 ค่าที่สามารถเปลี่ยนแปลงได้นั้นจะต้องเป็นค่าที่นอกเหนือจากโจทย์เท่านั้น
3. ในการหาคำตอบของ Sudoku จะทำการแทนค่าที่ละ Row โดย

จากข้อบังคับดังที่ได้กล่าวมา ในการคำนวณเพื่อหาคำตอบของ Sudoku เราจะเลือกข้อบังคับเกี่ยวกับการกำหนดค่าและตำแหน่งที่คงที่มาใช้เพื่อกำหนดตำแหน่งและค่าของโจทย์ ซึ่งทำให้การหาคำตอบของ Sudoku จะขึ้นอยู่กับการเปลี่ยนแปลงค่าของข้อบังคับทั้ง 3 ข้อที่เหลือคือ จะเติมตัวเลขในแต่และแถว แต่ละหลัก และแต่ละ Box ของตาราง โดยจะเติมตัวเลขเฉพาะช่องที่สามารถเติมตัวเลขได้เท่านั้น (ในที่นี้หมายถึงช่องที่นอกเหนือจากโจทย์)

2.3.3 การทำงานของ GPSO Algorithm

Geometric PSO algorithm

for all particle i **do**
  initialise position x_i at random in the search space
end for
while stop criteria not met **do**
  for all particle i **do**
    set personal best $\hat{x}_i$ as best position found so far by the particle
    set global best $\hat{g}$ as best position found so far by the whole swarm
  end for
  for all particle i **do**
    update position using a randomized convex combination

$$x_i = CX((x_i, \omega), (\hat{g}, \phi_1), (\hat{x}_i, \phi_2))$$

    mutate x_i
  end for
end while

รูปที่2.10 แสดงการทำงาน GPSO Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

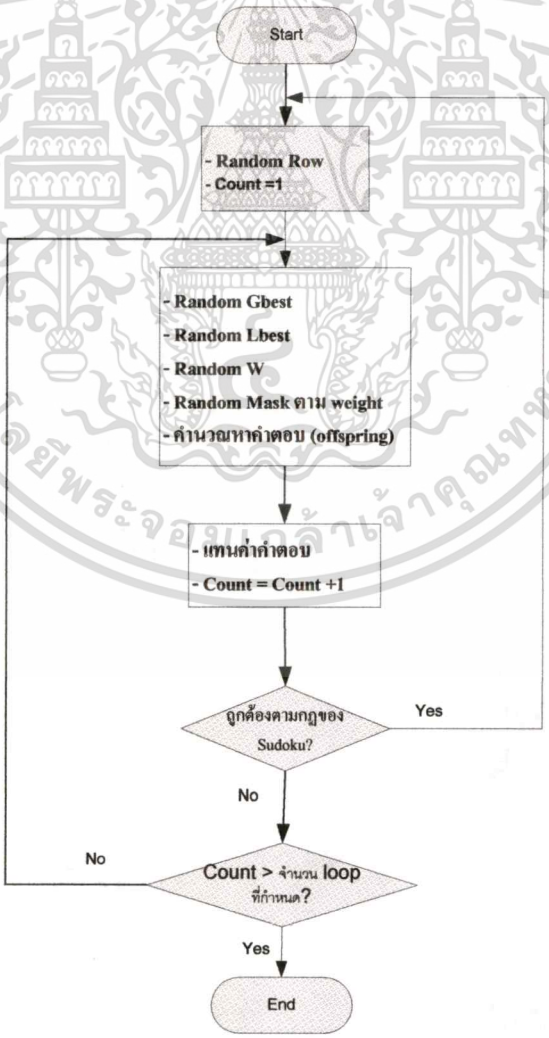
ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

จากรูปที่ 2.10 แสดงการทำงานของ GPSO อัลกอริทึ่ม โดยการทำงานจะเริ่มจกการกำหนดค่า Local Best และ Global Best และค่า Random ต่างๆที่ใช้ในการคำนวณหา Convex Combination ตามสมการที่ 2.8

$$x_i = CX((x_i, \omega), (\hat{g}, \phi_1), (\hat{x}_i, \phi_2)) \tag{2.8}$$

กำหนดให้

- *Xi* แทนค่า Row ที่ Random ขึ้นมา
- g แทนค่า Global Best (Gbest)
- x แทนค่า Local Best (Lbest)
- *w* , Ø1 , Ø2 แทนค่า weight



รูปที่2.11 Flowchart แสดงการทำงานของ GPSO อัลกอริทึ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
04860

2.3.4 การทดลอง

ในการทดลองต้องมีการกำหนดค่าต่างที่ใช้ในการทดลองซึ่งประกอบด้วย

1. Encoding เนื่องจาก Sudoku เป็นตาราง 9x9 ทำให้มีจำนวนสมาชิกของตารางทั้งหมด 81 ตัว ดังนั้นในการเก็บข้อมูลของตารางจะเก็บข้อมูลเป็น Array ตัวเลขจำนวนเต็ม (Integer) ขนาด 81 ตัว และ Range ของตัวเลขในแต่ละแถว คือ ตัวเลข 1 ถึงเลข 9 [1..9]
2. Initialization ในตอนเริ่มต้นจะใช้วิธีการสุ่มตัวเลข1-9 ในแต่ละ Row โดยใช้สูตรการคำนวณในสมการที่ 2.8 แล้วเลือกเฉพาะ ค่าที่ Match กับโจทย์ที่ได้กำหนดไว้แล้วนำมาตรวจสอบว่าขัดกับข้อบังคับของ Sudoku หรือไม่
3. ขนาดของจำนวนประชากร (Population) ขนาดของจำนวนประชากรที่ใช้ในการทดลองใช้จำนวน 5000 เซต และมีจำนวน 2500 เซตที่สามารถเป็น Candidate ของ Solution ได้

2.4 Sudoku Puzzle

2.4.1 ประวัติ และความรู้พื้นฐานของ Sudoku

ความหมายของ Sudoku (ซูโดกุ) ในภาษาญี่ปุ่น เป็นคำย่อจากคำว่า ซูจิวะโดะกุชินนิคางิรุ (Suuji wa dokushin ni kagiru) มีความหมายว่า "ตัวเลขต้องมีเพียงเลขเดียว" ชื่อของซูโดกุ มีการเรียกชื่อแตกต่างกันในแต่ละภาษา ตั้งแต่ ซูโดกุ ซูโดกู หรือ ซูโดคู

Sudoku (ซูโดกุ) คือ เกมปริศนาตัวเลข ที่ผู้เล่นต้องเลือกใส่ หมายเลขตั้งแต่ เลข 1 ถึงเลข 9 โดยมีเงื่อนไขว่าในแต่แถวและแต่ละหลักตัวเลขต้องไม่ซ้ำกัน ตารางซูโดกุจะมี 9x9 ช่อง ซึ่งประกอบจากตารางย่อย 9 ตาราง ในลักษณะ 3x3 แบ่งแยกกันโดยเส้นหนา และในแต่ละตารางย่อยจะมีตัวเลข 1 ถึง 9 เช่นเดียวกัน เมื่อเริ่มเกมจะมีตัวเลขบางส่วนให้มาเป็นคำใบ้ และผู้เล่นจะต้องใส่ทุกช่องที่เหลือให้ครบ โดยตามเงื่อนไขว่าแต่ละตัวเลขในแต่ละแถวและหลักจะใช้ได้ครั้งเดียว รวมถึงในแต่ละขอบเขตตารางย่อย

ไอเดียของ Sudoku มาจากสิ่งที่เรียกว่า Latin Squares หรือตารางที่ประกอบด้วย N x N ช่อง (ตามแนวนอน จำนวน N ช่อง และแนวตั้งจำนวน N ช่อง เช่น 9 x 9 ช่อง รวมเป็น 81 ช่อง) โดยมีหมายเลขเรียงไปตั้งแต่ช่อง 1, 1 (แถวที่ 1 กับหลักที่ 1) ช่องที่ 1, 2 (แถวที่ 1 กับหลัก2) นักคณิตศาสตร์ชื่อ Leonard Euler เป็นผู้คิดค้นเกมนี้ขึ้นในปี ค.ศ.1783 ซึ่งบางครั้งเรียกว่า Graeco-Latin Squares

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากปี 1783 เป็นต้นมาแนวคิดนี้ได้กลายเป็นเกมปริศนาที่เรียกว่า Number Place ในนิตยาสาร Dell Pencil Puzzles and Word Games ในตอนปลายทศวรรษ 1970 ผู้สร้างเกมปริศนานี้จริง ๆ แล้วคือ Howard Garnets (ชาวอเมริกันที่มาเติบโตในญี่ปุ่น) ซึ่งต่อมาเขาได้พัฒนาเป็นเกมที่ชื่อว่า Sudoku ในปี 1979 แต่ก็ไม่เป็นที่นิยมในอเมริกา จนกระทั่งเมื่อเขาย้ายอยู่ที่ญี่ปุ่นเกมนี้จึงเริ่มดังขึ้นมา ในปี 1984 บริษัท Nikoli ในประเทศญี่ปุ่น ได้ตีพิมพ์นิตยสารเกมปริศนาต่าง ๆ เช่น ปริศนาต่างๆได้นำ Number Place มาให้คนญี่ปุ่นเล่น โดยตั้งชื่อว่า Suuji wa dokushin ni kagiru ซึ่งเป็นที่นิยมอย่างมาก ในปี 1986 ได้มีการปรับปรุงเกม Number Place และเปลี่ยนชื่อใหม่ให้จำง่าย ๆ ว่า Sudoku ในประเทศญี่ปุ่นแต่ละเดือนบรรดานิตยสารที่ตีพิมพ์เกม Sudoku มียอดขายเพิ่มขึ้นอย่างมาก

ในช่วงเวลาที่ Sudoku เป็นที่นิยมในญี่ปุ่นนั้นคนยุโรปและคนอเมริกันแทบจะไม่ให้ความสำคัญกับ Sudoku เลย จนกระทั่งในปี 1997 Wayne Gould อดีตผู้พิพากษาชาวนิวซีแลนด์ได้เข้ามาซื้อหนังสือที่ร้านขายหนังสือญี่ปุ่นเพื่อฆ่าเวลา ถึงแม้จะอ่านภาษาญี่ปุ่นไม่ออกแต่เมื่อพบหนังสือ Sudoku ก็ซื้อมาเล่นโดยอาศัยการเดาทิศทางเอาเอง จากนั้น Wayne Gould ก็คิดหาเงินจาก Sudoku ด้วยการเขียนโปรแกรมคอมพิวเตอร์ผลิตเมล์ Sudoku อย่างไม่รู้จบออกขาย จากนั้นไม่นานคนยุโรปและคนอเมริกันก็นิยมเล่น Sudoku กันอย่างแพร่หลาย

2.4.2 เงื่อนไขในการเล่น Sudoku Puzzle

Sudoku Puzzle เป็นตารางที่มีขนาด 9x9 และแบ่งย่อยเป็นขนาด 3x3 ภายในอีกที ภายในตารางจะประกอบไปด้วยตัวเลข 1-9 โดยมีเงื่อนไขว่าในแต่ละแถวและแต่ละคอลัมน์นั้นตัวเลขจะต้องไม่ซ้ำกัน และในตารางย่อยขนาด 3x3 ก็จะต้องประกอบด้วยตัวเลข 1-9 เช่นกัน

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

รูปที่2.12 แสดงตัวอย่าง Sudoku Puzzle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขของ Sudoku Puzzle มี 3 ข้อ ดังนี้

1. ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน
2. ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน
3. ทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน

2.4.3 วิธีการหาคำตอบของ Sudoku Puzzle

ในส่วนนี้จะเป็นการแนะนำวิธีการคิดหาคำตอบของ Sudoku ซึ่งจะมีหลายระดับด้วยกัน ตั้งแต่ระดับง่ายไปยาก โดยทั่วไปแล้วการคิดหาคำตอบของ Sudoku นั้นจะเป็นการตัดตัวเลือกให้เหลือตัวเลือกน้อยที่สุด จนกระทั่งเหลือเพียงตัวเลือกเดียว จึงรู้คำตอบ

วิธีการหาคำตอบของ Sudoku Puzzle

1. ในแต่ละแถว(row)/หลัก(column) ในตาราง สามารถใช้ตัวเลขที่มีอยู่ได้เพียงครั้งเดียวเท่านั้น ห้ามซ้ำกัน หรืออธิบายง่ายๆว่า ภายในแต่ละแถว/หลัก ห้ามมีตัวเลขซ้ำกันนั่นเอง

2. ในแต่ละกลุ่ม(box ขนาด 3x3) ในตาราง สามารถใช้ตัวเลขทั้งหมดได้เพียงครั้งเดียว ห้ามซ้ำกันเช่นเดียวกับข้อ 1

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 2.13 แสดงตัวอย่างโจทย์ของ Sudoku Puzzle

จากรูปให้สังเกตที่ box ขนาด 3x3 ตรงมุมขวาบน จากข้อกำหนดข้อที่ 2 คือแต่ละกล่องจะต้องมีตัวเลขครบทุกตัวคือมีตัวเลขตั้งแต่ 1 ถึง 9 และกล่องที่ที่อยู่ตรงขวาบนก็จะต้องมีตัวเลขครบทุกตัวเช่นกัน จากโจทย์ที่ให้มาจะมีเลข 6 ดังนั้นตัวเลขที่สามารถเติมในกล่องนี้ได้คือ 1 2 3 4 5 7 และ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป กล่องมุมขวาบน พิจารณาตัวเลขที่สามารถเติมลงไปในกล่องดังกล่าวได้โดยอ้างอิงจากข้อกำหนดข้อที่ 1 คือในแต่ละแถว/หลัก ห้ามใส่ตัวเลขซ้ำกัน จากรูปที่2.12 จะเห็นว่า แถวที่ที่ 1 และ 2 มีตัวเลข 5 อยู่แล้ว ดังนั้นช่องว่างในแถวดังกล่าว (แถวที่ 1 และแถวที่5) จึงไม่สามารถลงเลข 5 ได้อีก ทำให้ในกล่องมุมขวาบน เหลือตำแหน่งที่จะลงเลข 5 ได้เพียงตำแหน่งเดียวคือตำแหน่ง 3,7 (ตำแหน่งที่ แถวที่ 3 หลักที่ 7)

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

รูปที่2.14 แสดงตัวอย่างการหาคำตอบของ Sudoku Puzzle

จากวิธีการด้านบนจะเป็นวิธีการหาคำตอบของ Sudoku Puzzle แบบง่าย โดยที่ผู้เล่นจะต้องใช้วิธีดังกล่าวหาคำตอบจนกว่าจะได้ตัวเลขครบทุกช่องในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์ระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization

ในบทที่ 3 นี้จะกล่าวถึงการวิเคราะห์ความต้องการ ของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization และการวิเคราะห์การจัดเก็บข้อมูล เพื่อใช้ในการสร้างโจทย์ของ Sudoku Puzzle รวมถึงการวิเคราะห์โปรเซสในการหาคำตอบของ Sudoku Puzzle ด้วยวิธีการแบบทั่วไปและรวมถึงการใช้อัลกอริทึม Geometric Particle Swarm Optimization ด้วย

3.1 วิเคราะห์ความต้องการของระบบ

Sudoku Puzzle คือ เกมปริศนาตัวเลข ที่ผู้เล่นต้องเลือกใส่ หมายเลขตั้งแต่ เลข 1 ถึงเลข 9 ให้ตรงตามเงื่อนไข โดยมีเงื่อนไขดังต่อไปนี้

เงื่อนไขของ Sudoku Puzzle มี 3 ข้อ ดังนี้

1. ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน

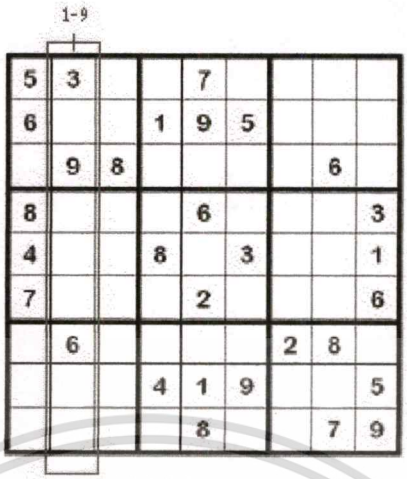
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

1-9

รูปที่3.1 แสดงการใส่เลขในแถวแนวนอนของ Sudoku Puzzle

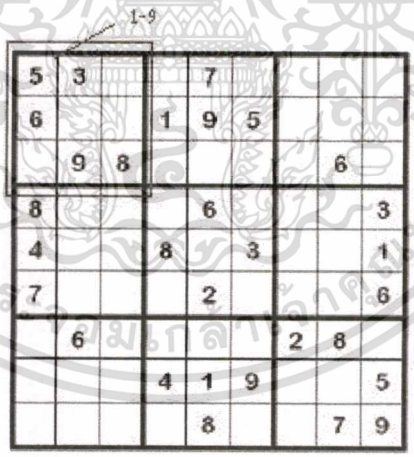
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน



รูปที่3.2 แสดงการใส่เลขในแถวแนวตั้งของ Sudoku Puzzle

3. ทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน



รูปที่3.3 แสดงการใส่เลขในตารางขนาด 3x3 ของ Sudoku Puzzle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 Functional Requirement ของระบบมีดังต่อไปนี้

1. Sudoku Puzzle ของระบบมีลักษณะเป็นตารางขนาด 9x9 ดังรูป

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

รูปที่3.4 แสดงตัวอย่าง Sudoku Puzzle

2. ในการเริ่มต้นเล่นเกมใหม่ผู้เล่นสามารถเลือกระดับความยากของเกมได้ โดยระบบจะมีการแบ่งระดับความยากของเกมเป็น 3 ระดับด้วยกันดังนี้
    1. Easy :ความยากระดับน้อยที่สุด
    2. Medium:ความยากระดับปานกลาง
    3. Hard :ความยากระดับมากที่สุด

3. เมื่อเริ่มต้นเล่นเกมแล้ว ในส่วนของโจทย์ที่ระบบแสดงขึ้นมานั้น ผู้เล่นจะไม่สามารถแก้ไขตัวเลขในส่วนของโจทย์ได้
4. ในส่วนที่เป็นช่องว่างของตารางนั้น ผู้เล่นจะสามารถใส่ข้อมูลลงไปได้เฉพาะข้อมูลที่เป็นตัวเลข 1 ถึง 9 เท่านั้น
5. ในกรณีที่ผู้เล่นยังไม่แน่ใจในคำตอบของช่องต่างๆ ผู้เล่นสามารถใส่ Comment ไว้ก่อนเพื่อความสะดวกในการพิจารณาคำตอบได้
6. ในกรณีที่ผู้เล่นยังเล่นเกมไม่จบ ผู้เล่นสามารถบันทึก (Save) หรือดึงข้อมูล (Load) ที่บันทึกไว้ขึ้นมาเล่นในครั้งต่อไปได้
7. เมื่อผู้เล่นสามารถหาคำตอบของเกมได้ครบถ้วนและถูกต้อง ระบบจะต้องมีการบันทึกเวลาในการเล่น และ ชื่อของผู้เล่นได้
8. ผู้เล่นสามารถดูสถิติของเวลาที่ดีที่สุดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ในกรณีที่ผู้เล่นไม่สามารถหาคำตอบได้ ผู้เล่นสามารถให้ระบบแสดงคำตอบได้ โดยในการหาคำตอบของระบบจะใช้วิธีการหาคำตอบ 2 วิธีด้วยกันคือ
    1. ใช้วิธีแบบทั่วไปในการหาคำตอบ ในกรณีนี้ระบบจะใช้วิธี Search ในการหาคำตอบ
    2. ใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO) ในการหาคำตอบ
10. นอกจากผู้แล่นทั่วไปแล้วระบบต้องมีส่วนที่ให้ผู้ที่เป็น Admin เข้าไปทำการเพิ่มโจทย์ของ Sudoku Puzzle พร้อมทั้งสามารถกำหนดระดับความยากของโจทย์ที่เพิ่มเข้าไปได้
11. ก่อนเข้าส่วนของ Admin ต้องมีการถาม Password ของ Admin ก่อน
12. Admin สามารถดู Report ได้ ซึ่งใน Report จะมีรายละเอียดของการเปรียบเทียบการหาคำตอบของ Sudoku โดยใช้วิธีแบบทั่วไป กับใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO) ในการหาคำตอบ

3.2 การวิเคราะห์ความต้องการของระบบ(System Requirement)

สถาปัตยกรรมของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization มีรายละเอียดดังนี้

1. ระบบได้ถูกพัฒนาในรูปแบบของ Application Program
2. ทำงานอยู่บนเครื่อง Personal computer ในลักษณะ Standalone
3. ลักษณะของการเก็บข้อมูลที่จำเป็นของระบบ มีการจัดเก็บข้อมูลในลักษณะ Text File เพื่อความสะดวกในการติดตั้งระบบ

ความต้องการของระบบ (System Requirement)

1. CPU :Pentium III 8oo MHz
2. Memory :128 MB
3. Hard disk :200 MB
4. Operating System :Windows XP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

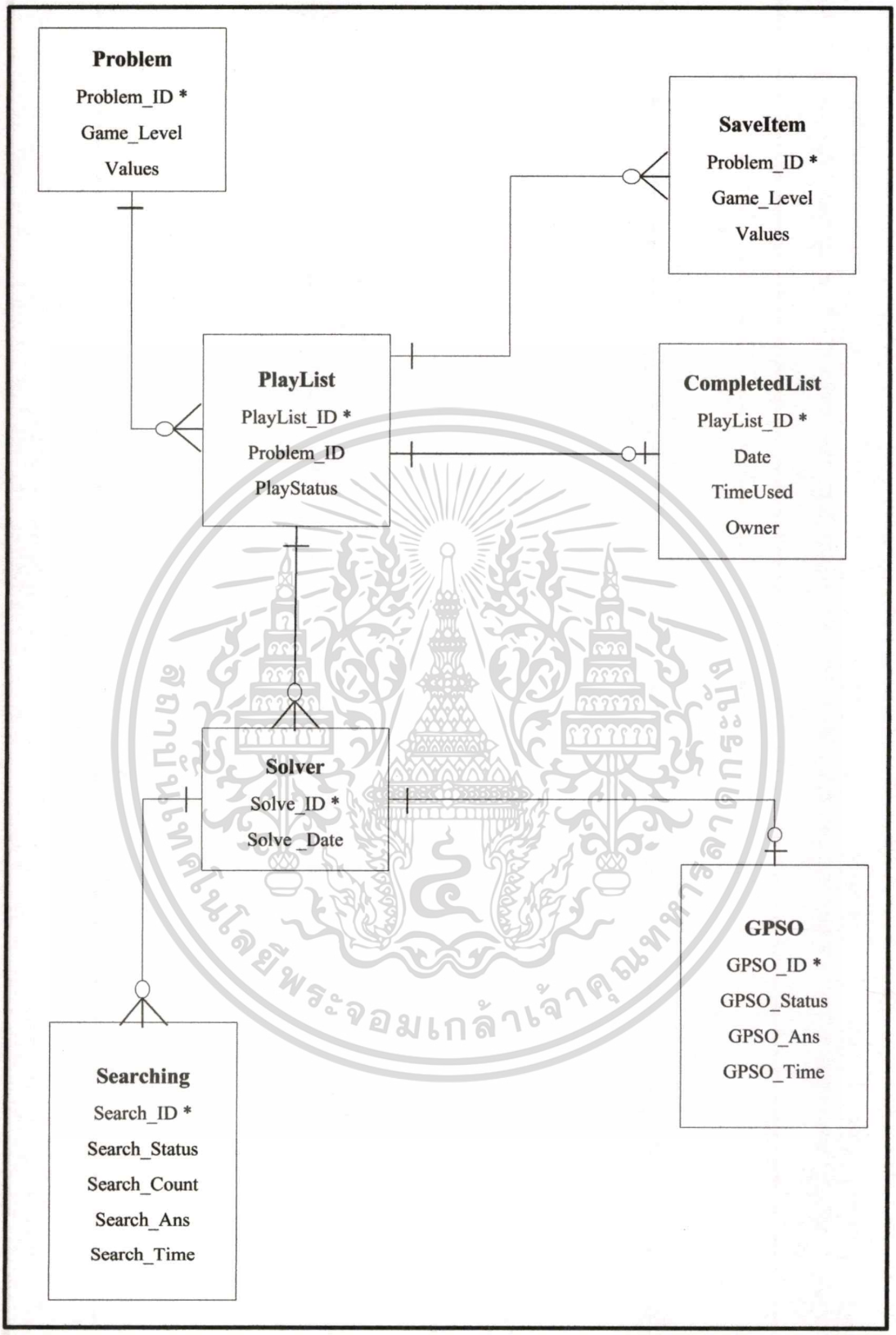
3.3 การวิเคราะห์การจัดเก็บข้อมูล

การวิเคราะห์การจัดเก็บข้อมูลของระบบ จะมีการจัดเก็บข้อมูลให้อยู่ในรูปของ Text File เพื่อความสะดวกในการติดตั้ง โดยจะมีการจัดเก็บข้อมูลในส่วนของโจทย์ของ Sudoku ในระดับความยากต่างๆ และการเก็บสถิติเวลาที่ดีที่สุดในการเล่น และข้อมูลอื่นๆที่จำเป็นต่อระบบ ซึ่งสามารถดูรายละเอียดได้จากตารางที่ 3.1

ตารางที่ 3.1 แสดงการจัดเก็บข้อมูลของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้ อัลกอริทึม Geometric Particle Swarm Optimization

Entity	Description
Problem	เก็บข้อมูลที่เป็นโจทย์ของ Sudoku
Play List	เก็บข้อมูลการเล่นของ User
Completed List	เก็บข้อมูลที่ผู้เล่น สามารถเล่นเกมได้จนเสร็จสมบูรณ์และถูกต้อง
Save Item	เก็บข้อมูล สถานะของเกม ที่ผู้เล่นยังเล่นไม่เสร็จ และต้องการบันทึกไว้เล่นต่อ
Solver	เก็บข้อมูลในกรณีที่ให้คอมพิวเตอร์หาคำตอบของ Sudoku
Searching	เก็บข้อมูลในกรณีที่ให้คอมพิวเตอร์หาคำตอบของ Sudoku โดยใช้วิธีการ Searching
GPSO	เก็บข้อมูลในกรณีที่ให้คอมพิวเตอร์หาคำตอบของ Sudoku โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

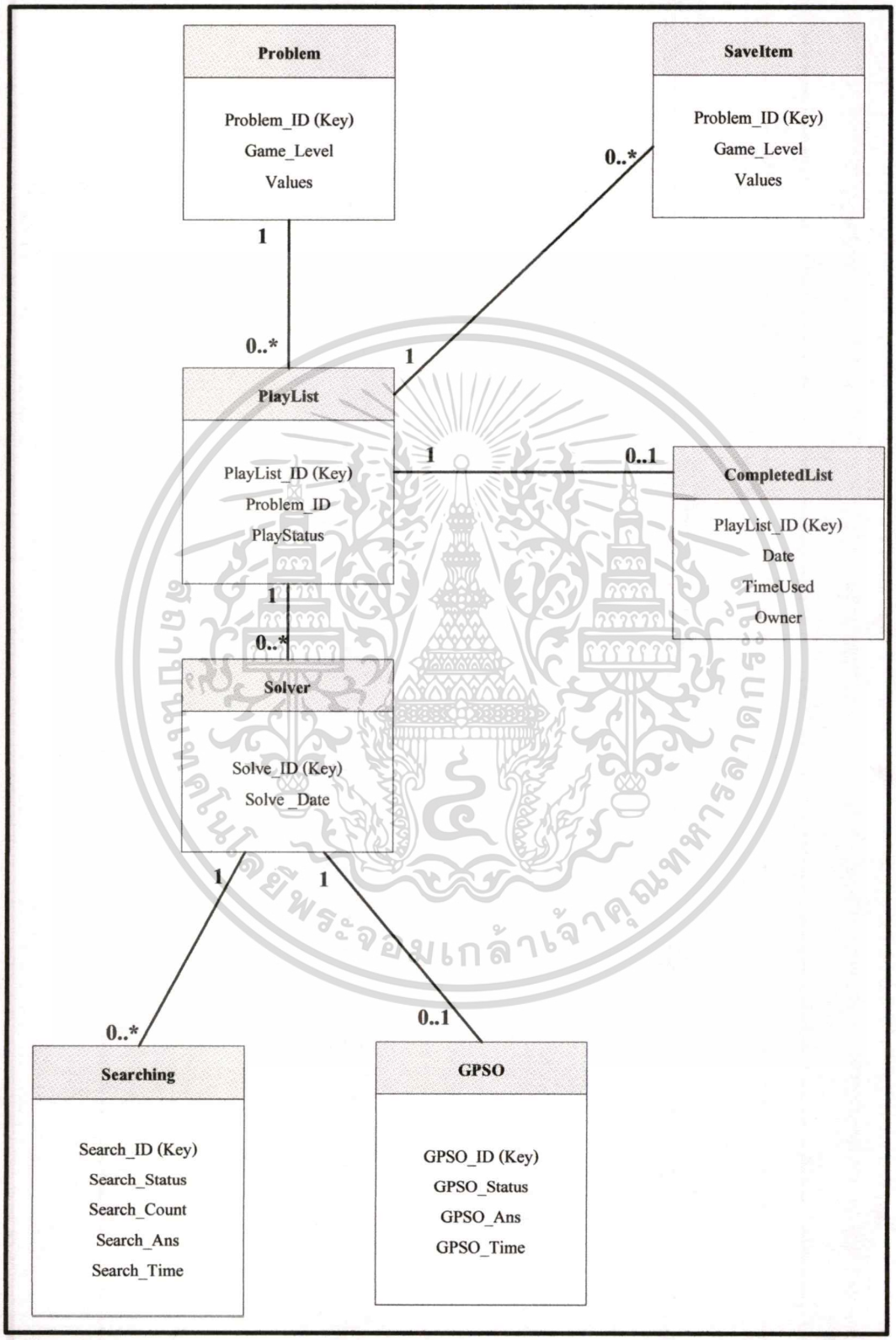


รูปที่3.5 แสดง ER Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การวิเคราะห์การทำงานของระบบ

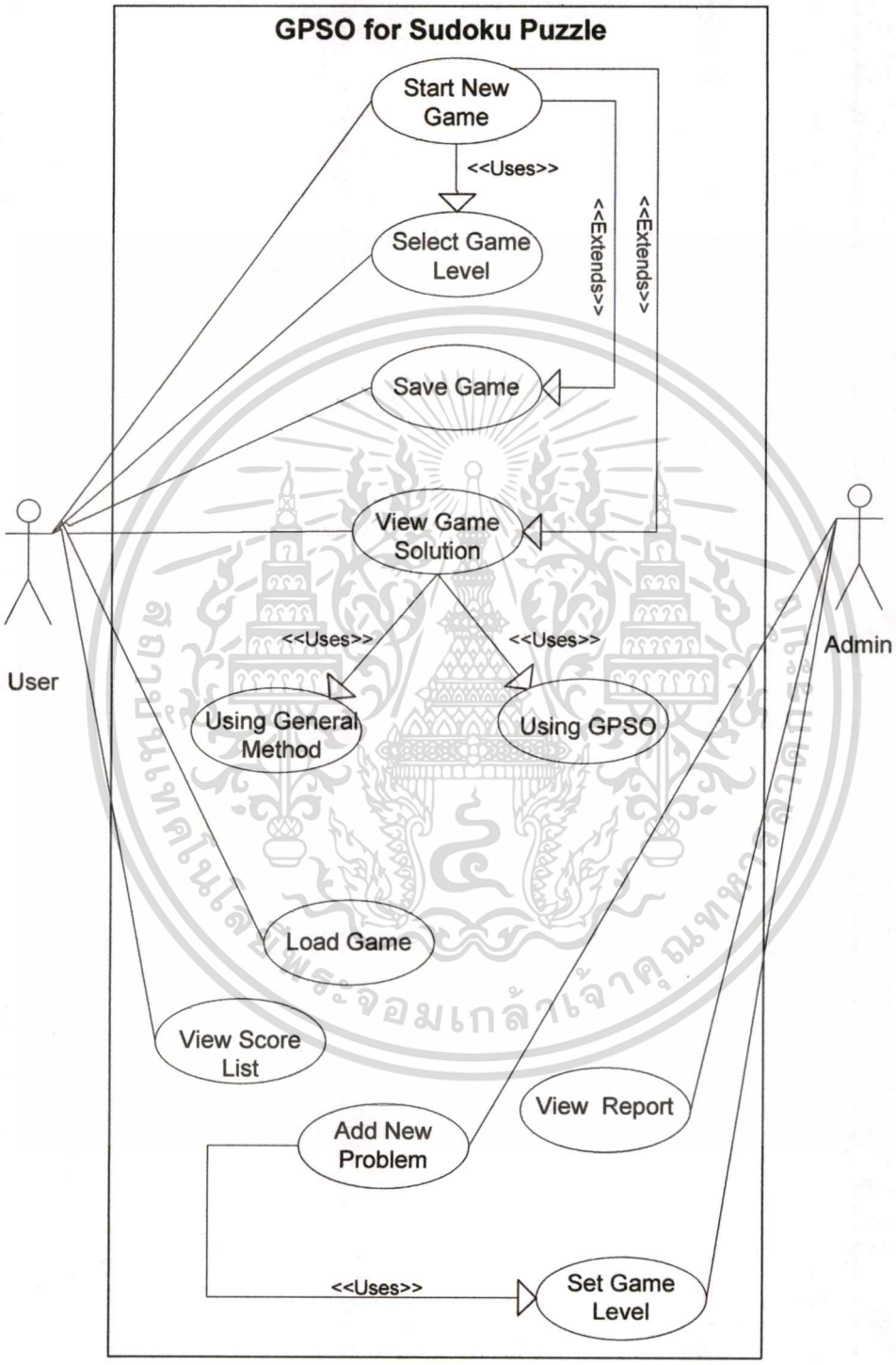
3.4.1 Domain Class Diagram ของระบบ



รูปที่ 3.6 Domain Class Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 UML Use Case Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization



รูปที่ 3.7 Use Case Diagram ของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึม Geometric Particle Swarm Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 Use Case Description

ตารางที่ 3.2 รายละเอียดยูสเคส Start New Game

ชื่อยูสเคส	Start New Game
คำอธิบายยูสเคส	ผู้เล่นเริ่มเล่นเกมใหม่
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Select Game Level
เงื่อนไขเริ่มต้น	ก่อนที่จะทำการเริ่มต้นเกมใหม่ผู้เล่นจะต้องมีการเลือกระดับความยากของเกมเสียก่อน
การทำงาน	1. เมื่อผู้เล่นทำการกดปุ่มเพื่อเริ่มต้นเกมใหม่ระบบจะทำการสุ่มหาโจทย์ของ Sudoku ตามระดับความยากที่ผู้เล่นเลือกไว้ 2. ระบบทำการ Display โจทย์ในตารางขนาด 9x9 3. ระบบเริ่มจับเวลาที่ใช้ในการเล่นของผู้เล่นดังกล่าว
เงื่อนไขการทำงาน	1. โจทย์ของ Sudoku ที่จะ Display นั้นจะต้องมีระดับความยากตามที่ผู้เล่นได้เลือกไว้ 2. ในกรณีที่ผู้เล่นไม่ได้ทำการเลือกระดับความยากของเกม ค่า Default จะถูกกำหนดไว้ที่ระดับความยากที่น้อยที่สุด คือ Easy
เงื่อนไขเมื่อสำเร็จ	ในตารางขนาด 9x9 มีการ Display โจทย์ของ Sudoku ซึ่งไม่สามมารถแก้ไขเปลี่ยนแปลงตัวเลขได้ และ ในช่องว่างที่มีไว้สำหรับใส่ข้อมูลนั้น ผู้เล่นสามารถใส่ข้อมูลที่เป็นตัวเลขลงไปได้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 รายละเอียดยูสเคส Select Game Level

ชื่อยูสเคส	Select Game Level
คำอธิบายยูสเคส	ผู้เล่นเลือกระดับความยากของเกม
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Start New Game
เงื่อนไขเริ่มต้น	1. เมื่อผู้เล่นต้องการเริ่มต้นเกมใหม่จะต้องมีการเลือกระดับความยากของเกมเสียก่อน
การทำงาน	1. เมื่อผู้เล่นทำการเลือกระดับความยากของ Sudoku แล้วกดปุ่มเพื่อเริ่มต้นเกมใหม่ระบบจะทำการสุ่มหาโจทย์ของ Sudoku ตามระดับความยากที่ผู้เล่นเลือกไว้ แล้วระบบจะทำการ Display โจทย์ในตารางขนาด 9x9
เงื่อนไขการทำงาน	1. ในกรณีที่ผู้เล่นไม่ได้ทำการเลือกระดับความยากของเกม ค่า Default จะถูกกำหนดไว้ที่ระดับความยากที่น้อยที่สุด คือ Easy
เงื่อนไขเมื่อสำเร็จ	ระบบจะทำการสุ่มหาโจทย์ของ Sudoku ตามระดับความยากที่ผู้เล่นเลือกไว้ได้อย่างถูกต้อง

ตารางที่ 3.4 รายละเอียดยูสเคส Save Game

ชื่อยูสเคส	Save Game
คำอธิบายยูสเคส	ผู้เล่นทำการบันทึกสถานะของเกมเอาไว้เล่นต่อในครั้งต่อไป
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Start New Game
เงื่อนไขเริ่มต้น	1. เมื่อผู้เล่นเริ่มต้นเล่นเกมแล้ว 2. ผู้เล่นยังเลนเกมนั้นไม่จบและต้องการบันทึกไว้เล่นต่อในครั้งถัดไป
การทำงาน	1. เมื่อผู้เล่นกดปุ่มเพื่อต้องการ Save เกมระบบจะแสดงหน้าจอให้ผู้เล่นทำการใส่ชื่อไฟล์ 2. ระบบจะทำการบันทึกข้อมูลของผู้เล่นลงในไฟล์ตามที่ได้ตั้งชื่อไว้
เงื่อนไขการทำงาน	1. ต้องมีการกำหนดชื่อไฟล์ที่ต้องการจะบันทึกข้อมูลด้วยทุกครั้ง
เงื่อนไขเมื่อสำเร็จ	มีข้อความแจ้งเตือนผู้เล่นว่าทำการบันทึกข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 รายละเอียดยูสเคส View Game Solution

ชื่อยูสเคส	View Game Solution
คำอธิบายยูสเคส	ผู้เล่นต้องการดูคำตอบของเกม
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Using General Method, Using GPSO
เงื่อนไขเริ่มต้น	1. เมื่อผู้เล่นต้องการดูคำตอบของเกมโดยที่โจทย์ดังกล่าวไม่จำเป็นต้องมีอยู่ในระบบก็ได้ 2. โจทย์ที่จะให้ระบบหาคำตอบให้ต้องไม่ขัดต่อข้อบังคับของ Sudoku Puzzle ทั้ง 3 ข้อคือ • ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน
การทำงาน	1. ระบบทำการหาคำตอบของ Sudoku โดยมีวิธีที่ใช้ในการหาคำตอบด้วยกัน 2 วิธีดังนี้ • ใช้อัลกอริทึม GPSO • ใช้วิธีการ Searching 2. เมื่อเสร็จสิ้นการหาคำตอบจะมีการแสดงผลลัพธ์ว่าสามารถหาคำตอบได้หรือไม่
เงื่อนไขการทำงาน	1. ในการหาคำตอบของ Sudoku ด้วยวิธีการทั้ง 2 คือ ใช้อัลกอริทึ่ม GPSO และ ใช้การ Searching นั้นไม่จำเป็นว่าจะมีแค่เพียงคำตอบเดียวเพราะคำตอบของ Sudoku นั้นขึ้นอยู่กับโจทย์ที่ใช้ว่ามีได้หลายคำตอบหรือไม่ 2. ในการใช้อัลกอริทึ่ม GPSO ในการหาคำตอบของ Sudoku นั้นอาจจะหาคำตอบของ Sudoku ไม่ได้ในบางโจทย์
เงื่อนไขเมื่อสำเร็จ	ระบบทำการแสดงคำตอบของ Sudoku ที่หาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 รายละเอียดยูสเคส Using General Method

ชื่อยูสเคส	Using General Method
คำอธิบายยูสเคส	หาคำตอบของ Sudoku Puzzle โดยใช้วิธีการ Searching
แอกเตอร์	-
ยูสเคสที่เกี่ยวข้อง	View Game Solution
เงื่อนไขเริ่มต้น	1. ต้องมีการกำหนดโจทย์ของ Sudoku มาให้ก่อน 2. โจทย์ที่จะให้ระบบหาคำตอบให้ต้องไม่ขัดต่อข้อบังคับของ Sudoku Puzzle ทั้ง 3 ข้อคือ • ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน
การทำงาน	1. ระบบจะใช้วิธีการแทนค่าทุกค่าที่ไม่ขัดต่อข้อบังคับทั้งสามข้อของ Sudoku จนกว่าจะได้ค่าทุกๆค่าในตาราง 2. เมื่อได้คำตอบแล้วจะไม่แสดงคำตอบทันที ระบบจะหาคำตอบทุกๆ คำตอบที่เป็นไปได้
เงื่อนไขการทำงาน	1. ในการหาคำตอบนั้นไม่จำเป็นว่าคำตอบที่ได้จะมีแค่เพียงคำตอบเดียว ขึ้นอยู่กับการกำหนดโจทย์
เงื่อนไขเมื่อสำเร็จ	ระบบทำการแสดงคำตอบของ Sudoku ที่หาได้ พร้อมทั้งแสดงจำนวนคำตอบที่เป็นไปได้ของโจทย์นั้นๆ ว่ามีทั้งหมดจำนวนกี่คำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 รายละเอียดยูสเคส Using GPSO

ชื่อยูสเคส	Using GPSO
คำอธิบายยูสเคส	หาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO)
แอกเตอร์	-
ยูสเคสที่เกี่ยวข้อง	View Game Solution
เงื่อนไขเริ่มต้น	1. ต้องมีการกำหนดโจทย์ของ Sudoku มาให้ก่อน 2. โจทย์ที่จะให้ระบบหาคำตอบให้ต้องไม่ขัดต่อข้อบังคับของ Sudoku Puzzle ทั้ง 3 ข้อคือ • ทุกแถวในแนวนอน ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกแถวในแนวตั้ง ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน • ทุกตารางย่อย 3 x 3 ต้องมีตัวเลข 1 ถึงเลข 9 และต้องไม่ซ้ำกัน
การทำงาน	1. ระบบจะใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO) คำตอบของ Sudoku
เงื่อนไขการทำงาน	1. ในการหาคำตอบด้วยวิธีการนี้อาจจะไม่สามารถหาคำตอบของ Sudoku ได้ในบางโจทย์ 2. จะมีการกำหนดเวลาไว้ระยะเวลาหนึ่ง เพื่อป้องกันในกรณีที่เกิดการหาคำตอบไม่เสร็จเพื่อไม่ให้ระบบทำงานแบบไม่รู้จบ 3. มีการแสดงผลลัพธ์ว่าระบบสามารถหาคำตอบของ Sudoku ได้สำเร็จหรือไม่
เงื่อนไขเมื่อสำเร็จ	ระบบทำการแสดงคำตอบของ Sudoku

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 รายละเอียดยูสเคส Load Game

ชื่อยูสเคส	Load Game
คำอธิบายยูสเคส	ผู้เล่น Load เกมที่เคยบันทึกไว้เพื่อมาเล่นต่อ
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	Save Game
เงื่อนไขเริ่มต้น	ต้องมีการ Save เกมไว้ก่อน
การทำงาน	1. ผู้เล่นจะต้องเลือกชื่อไฟล์ที่ต้องการ Load จาก List ที่ระบบแสดงขึ้นมา
เงื่อนไขการทำงาน	1. ถ้าผู้เล่นสามารถหาคำตอบของเกมได้หมดแล้วข้อมูล Save ดังกล่าวจะถูกลบออกจากที่เก็บข้อมูล
เงื่อนไขเมื่อสำเร็จ	ระบบจะ Load ข้อมูลที่ได้บันทึกไว้ลงในตาราง Sudoku ขนาด 9x9

ตารางที่ 3.9 รายละเอียดยูสเคส View Score List

ชื่อยูสเคส	View Score List
คำอธิบายยูสเคส	ผู้เล่นดูสถิติเวลาในการเล่นที่ดีที่สุด
แอกเตอร์	User
ยูสเคสที่เกี่ยวข้อง	-
เงื่อนไขเริ่มต้น	1. ระบบจะสามารถแสดงข้อมูลในส่วนนี้ได้จะต้องมีการบันทึกข้อมูลการเล่นมาก่อนหน้านี้
การทำงาน	1. เมื่อผู้เล่นสามารถหาคำตอบของ Sudoku ได้อย่างครบถ้วนและถูกต้อง ระบบจะมีการให้กรอกข้อมูลชื่อผู้เล่นเพื่อบันทึกข้อมูลเก็บไว้ และเมื่อผู้เล่นต้องการดูสถิติเวลาในการเล่นที่ดีที่สุด ระบบก็จะดึงข้อมูลดังกล่าวขึ้นมาแสดงผล
เงื่อนไขการทำงาน	1. ข้อมูลที่แสดงจะประกอบด้วยชื่อผู้เล่นและเวลาที่ใช้ โดยจะมีการแบ่งคะแนนตามระดับความยากง่าย จะไม่แสดงข้อมูลรวมกันทั้งหมด
เงื่อนไขเมื่อสำเร็จ	มีชื่อผู้ดูแลระบบคนใหม่ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 รายละเอียดยูสเคส Add New Problem

ชื่อยูสเคส	Add New Problem
คำอธิบายยูสเคส	Admin ทำการเพิ่มเกมใหม่เข้าไปในระบบ
แอกเตอร์	Admin
ยูสเคสที่เกี่ยวข้อง	Set Game Level
เงื่อนไขเริ่มต้น	1. ต้องมีการใส่ Password ของ Admin ก่อนที่จะเข้าไปเพิ่มโจทย์ของ Sudoku ได้
การทำงาน	1. เมื่อใส่ Password ของ Admin เข้ามาแล้วระบบจะแสดงตารางขนาด 9x9 ให้ใส่ตัวเลขของโจทย์ใหม่ที่ต้องการเพิ่มเข้าไปในระบบ 2. เมื่อใส่ข้อมูลเรียบร้อยแล้วจะต้องมีการกำหนดระดับความยากของเกม\ 3. ระบบทำการตรวจสอบว่าโจทย์ที่ใส่เข้ามานั้นถูกต้องตามข้อบังคับของ Sudoku ทั้ง 3 ข้อหรือไม่ ถ้าถูกต้องระบบจะทำการบันทึกข้อมูลดังกล่าว
เงื่อนไขการทำงาน	1. โจทย์ที่จะเพิ่มเข้าไปในระบบนั้นเป็นตัวเลข 1-9 ได้เท่านั้น 2. ในการเพิ่มโจทย์นั้นจะต้องมีการกำหนดระดับความยากของเกมทุกครั้ง
เงื่อนไขเมื่อสำเร็จ	มีการบันทึกโจทย์ใหม่เข้าไปในระบบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 รายละเอียดยูสเคส Set Game Level

ชื่อยูสเคส	Set Game Level
คำอธิบายยูสเคส	Admin ทำการกำหนดระดับความยากของเกมหลังจากที่ทำการเพิ่มเกมใหม่เข้าไปในระบบ
แอกเตอร์	Admin
ยูสเคสที่เกี่ยวข้อง	Add New Problem
เงื่อนไขเริ่มต้น	1. มีการกำหนดโจทย์ของ Sudoku ให้เรียบร้อยก่อน
การทำงาน	1. เมื่อทำการกำหนดระดับความยากของเกมแล้วทำการบันทึก เข้าในระบบจะมีการเก็บข้อมูลโจทย์แยกตามระดับความยาก
เงื่อนไขการทำงาน	1. ระดับความยากของโจทย์ที่สามารถกำหนดได้นั้นจะประกอบด้วย 3 ระดับดังนี้ ● Easy ● Medium ● Hard
เงื่อนไขเมื่อสำเร็จ	มีการบันทึกโจทย์ของเกมใหม่ตามระดับความยากที่กำหนด

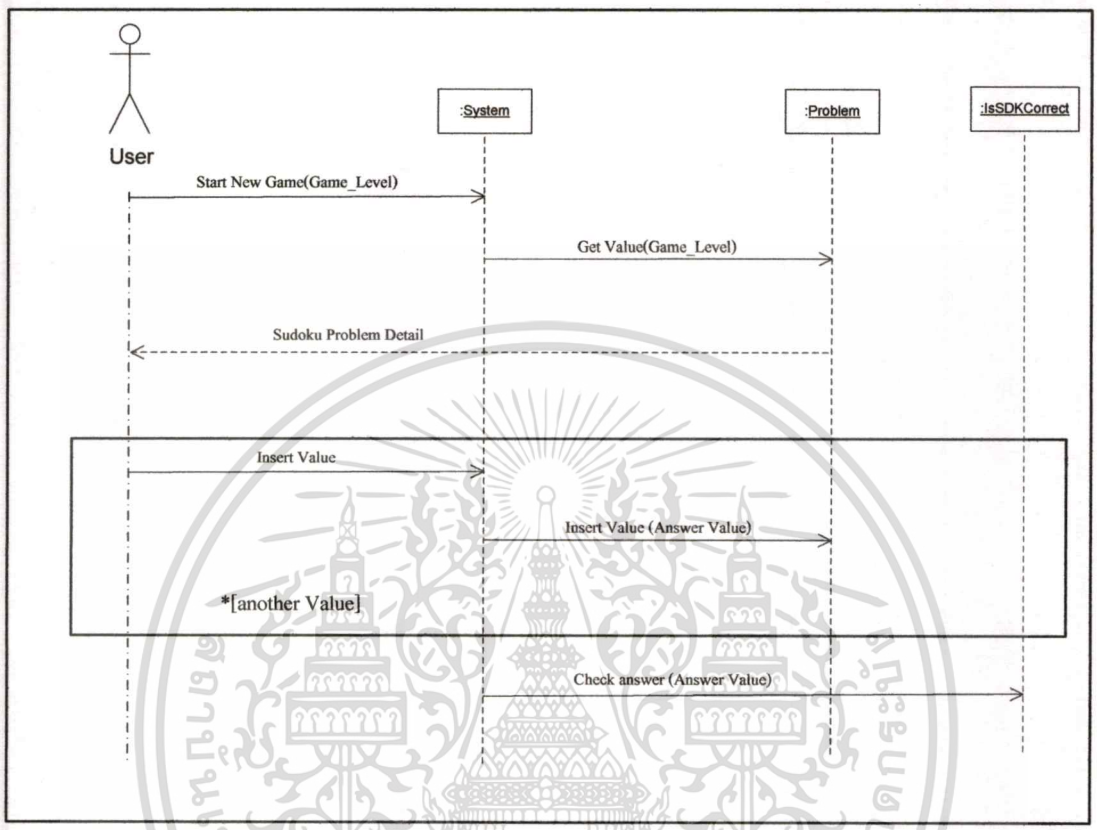
ตารางที่ 3.12 รายละเอียดยูสเคส View Report

ชื่อยูสเคส	View Report
คำอธิบายยูสเคส	Admin ดู Report เปรียบเทียบการหาคำตอบของ Sudoku โดยใช้วิธีแบบทั่วไปกับใช้อัลกอริทึม Geometric Particle Swarm Optimization (GPSO) ในการหาคำตอบ
แอกเตอร์	Admin
ยูสเคสที่เกี่ยวข้อง	-
เงื่อนไขเริ่มต้น	1. ต้องต้องมีการใส่ Password ของ Admin ก่อนที่จะเข้าดู Report ได้ 2. และต้องมีการบนทึกข้อมูลในส่วนของการให้ระบบหาคำตอบของ Sudoku ไว้แล้ว
การทำงาน	1. ผู้ดูแลระบบระดับสิทธิ์สูงสุดป้อนชื่อผู้ใช้งานและรหัสผ่านเพื่อเข้าระบบ 2. ผู้ดูแลระบบทำการเพิ่มรายชื่อผู้ดูแลระบบคนอื่นๆ และกำหนดสิทธิ์ให้กับ
เงื่อนไขการทำงาน	1. ระบบจะทำการดึงข้อมูลในส่วนของการหาคำตอบของ Sudoku ทั้งสองวิธีมาทำรายงายสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขเมื่อสำเร็จ	ระบบแสดงผลรายงานสรุปการหาคำตอบของ Sudoku

3.4.4 UML Sequence Diagram



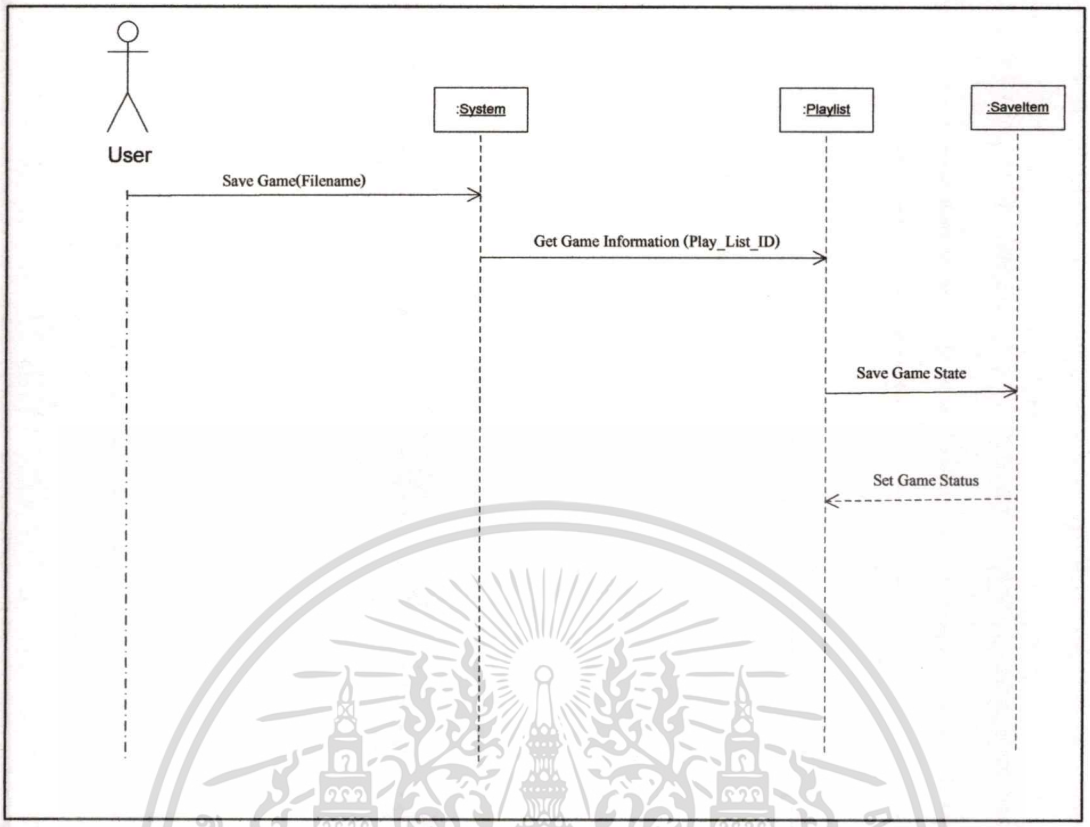
รูปที่ 3.8 Sequence Diagram ของการเริ่มต้นเกมใหม่

จากรูป Sequence Diagram ของการเริ่มต้นเกมใหม่ เป็นการแสดงขั้นตอนการทำงานในส่วนของการเริ่มต้นเกมใหม่ (Start New Game) โดยในขั้นตอนนี้จะเริ่มต้นจาก User ทำการเลือกระดับความยากของเกม ในระบบนี้จะประกอบด้วย 3 ระดับคือ

- Easy :ระดับความยากที่น้อยที่สุด
- Medium:ระดับความยากปนกลาง
- Hard :ระดับความยากสูงสุด

เมื่อทำการเลือกระดับความยากแล้วระบบจะใช้การสุ่มโจทย์ของ Sudoku จากที่เก็บข้อมูลตามระดับความยากที่ User ได้กำหนดไว้ก่อนทำการเริ่มต้นเล่นเกม หลังจากที่ User เล่นเกมเสร็จแล้ว ระบบจะทำการตรวจสอบว่าคำตอบทั้งหมดนั้นถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

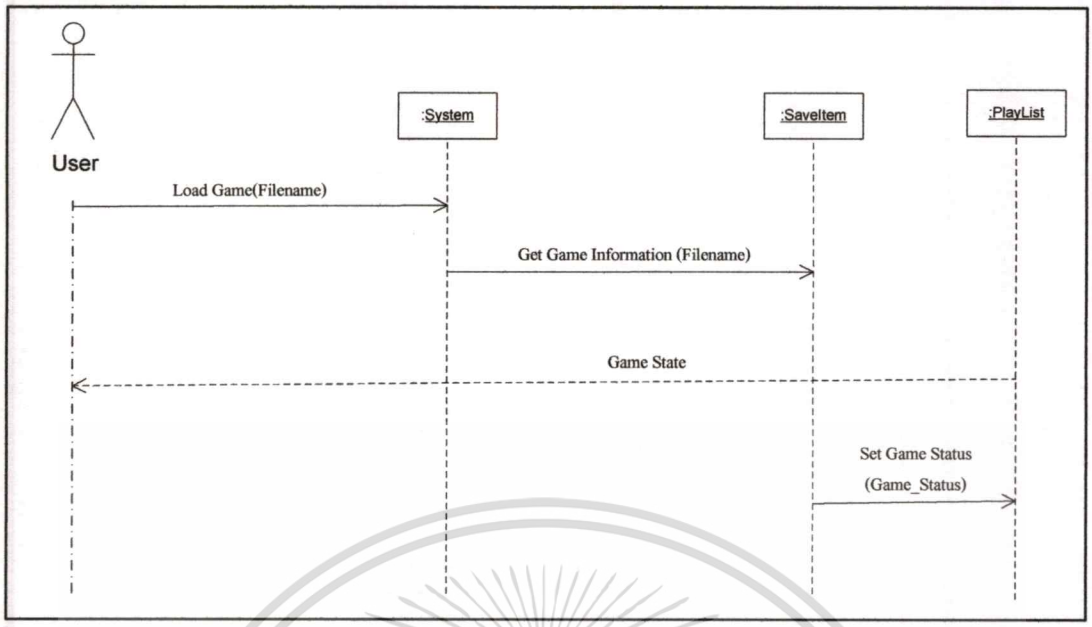


รูปที่ 3.9 Sequence Diagram ของการบันทึกเกม

จากรูป Sequence Diagram ของการบันทึกเกม เป็นการแสดงขั้นตอนการทำงานในส่วนของบันทึกเกม (Save Game) ในกรณีที่ User ทำการเล่นเกมแต่ยังเล่นไม่จบ และต้องการที่จะทำการบันทึกไว้เล่นต่อในครั้งต่อไป ในการบันทึกเกมไว้เล่นต่อนั้นจำเป็นจะต้องมีการกำหนดชื่อไฟล์ (File _Name) ที่ต้องการจะบันทึกไว้เพื่อใช้ในการอ้างอิง เมื่อ User ต้องการดึงข้อมูลที่บันทึกไว้ออกมาเล่นต่อ โดยหลังจากที่มีการกำหนดชื่อไฟล์เรียบร้อยแล้วระบบจะมีการกำหนดสถานะของเกม (Play_Status) เป็น S ซึ่งหมายถึง Save สถานะของเกมจะประกอบด้วย

- S หมายถึง เกมดังกล่าวผู้เล่นยังเล่นไม่จบแล้วได้ทำการบันทึกไว้เล่นต่อ
- C หมายถึงเกมดังกล่าวผู้เล่นได้ทำการเล่นจนเสร็จสมบูรณ์แล้ว
- L หมายถึง เกมดังกล่าวผู้เล่นยังเล่นไม่จบแล้วได้ทำการดึงข้อมูล (Load) ที่บันทึกไว้มาเล่นต่อ

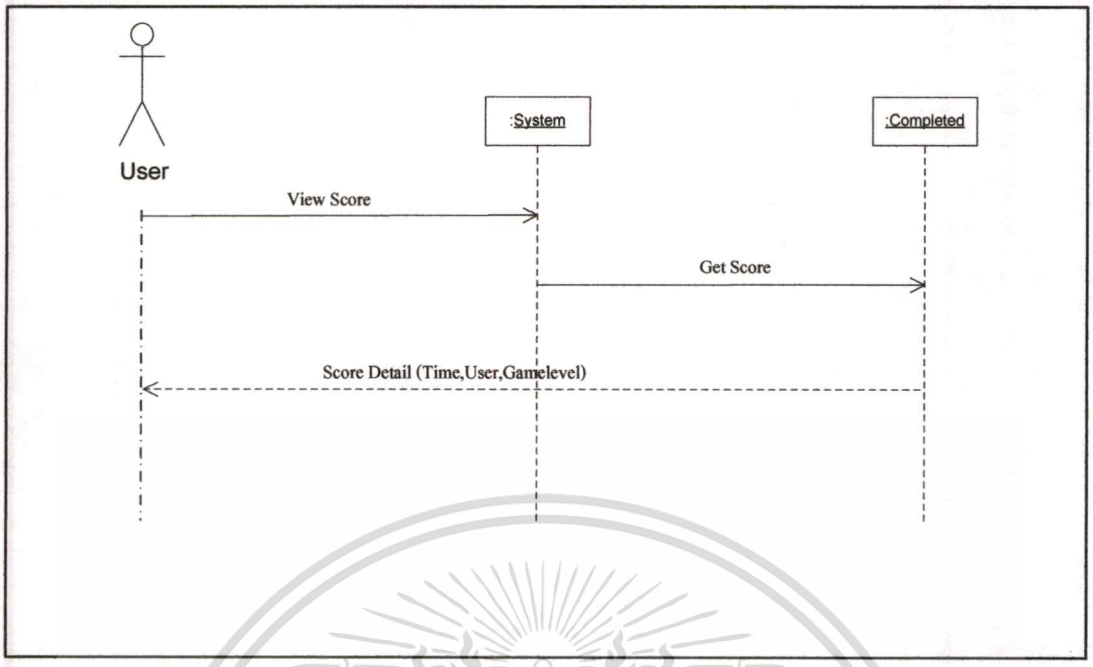
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Sequence Diagram ของการ Load เกม

จากรูป Sequence Diagram ของการดึงข้อมูลที่บันทึกไว้ (Load Game) ขั้นตอนการทำงานในส่วนนี้จะเริ่มต้นเมื่อ User ทำการกดที่ปุ่ม Load Game จากนั้นระบบจะแสดงหน้าจอในส่วนของ List รายชื่อไฟล์ที่ได้มีการบันทึกไว้ก่อนหน้านี้ให้ User ทำการเลือกไฟล์ที่ต้องการจะ Load มาเล่นต่อ จากนั้นระบบจะทำการค้นหาไฟล์ที่ User เลือกแล้วทำการแสดงผลกลับไปยัง User หลังจากนั้นระบบจะทำการ Update Status ของเกมใน Play_List ให้เป็นสถานะ L ซึ่งหมายถึง เกมดังกล่าวได้ถูกดึงข้อมูล (Load) ที่บันทึกไว้มาเล่นต่อ

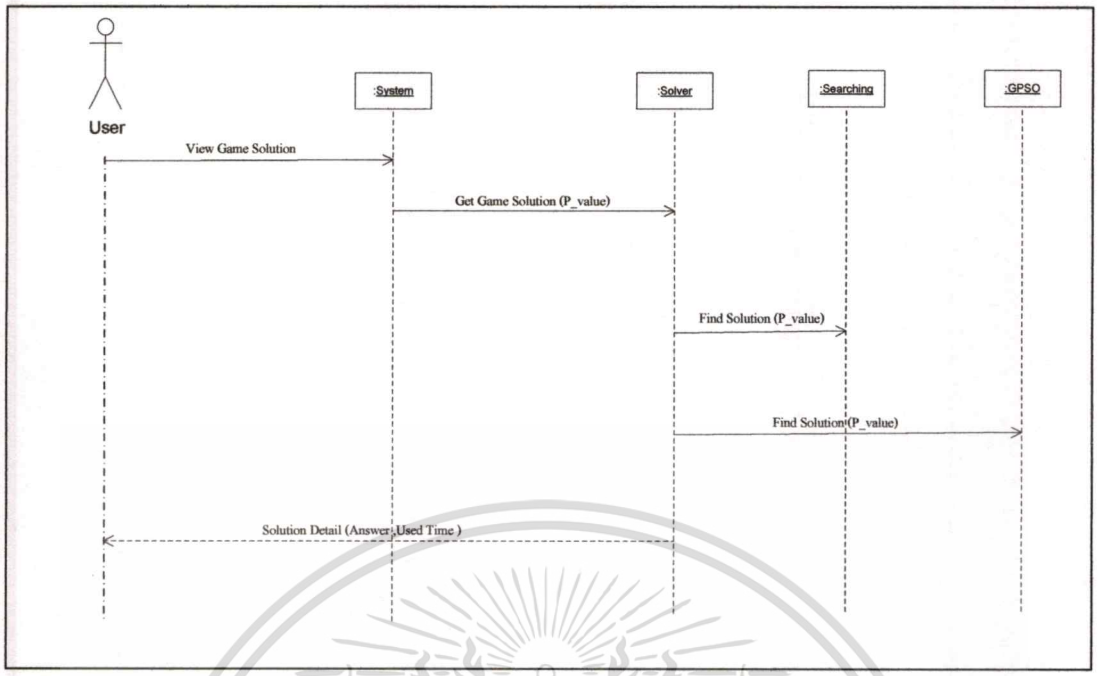
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 Sequence Diagram ของการดูสถิติเวลาในการเล่นเกมที่ดีที่สุด

จากรูป Sequence Diagram ของการดูสถิติเวลาในการเล่นเกมที่ดีที่สุด ขั้นตอนการทำงานในส่วนนี้จะเริ่มต้นเมื่อ User กดที่ปุ่ม View Score ระบบจะทำการดึงข้อมูลจาก Class ที่ชื่อ Completed ซึ่งเป็น Class ที่ใช้เก็บข้อมูลของผู้เล่นที่เล่นเกมจนเสร็จสมบูรณ์ และถูกต้องเท่านั้น โดยระบบจะใช้เวลาในการเล่นเป็นตัววัดคะแนน ผู้เล่นที่ใช้เวลาน้อยที่สุดถือว่าเป็นเวลาที่ดีที่สุด ระบบจะแบ่งเวลาที่ดีที่สุดตามระดับความยากของเกม โดนแจะแสดงผลเวลาที่ดีที่สุด 5 อันดับแรกเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



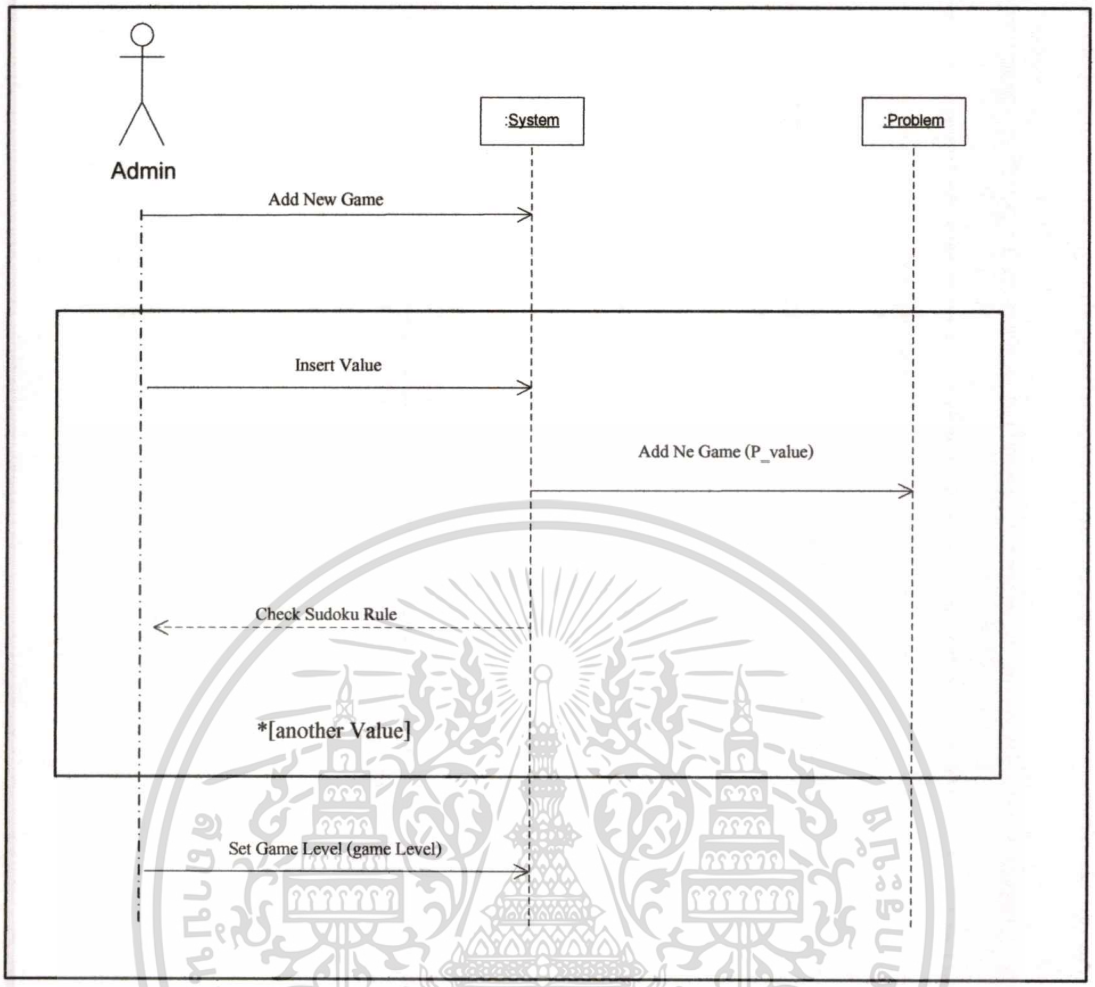
รูปที่ 3.12 Sequence Diagram ของการดูคำตอบของ Sudoku

จากรูป Sequence Diagram ของการดูดูคำตอบของ Sudoku ขั้นตอนการทำงานในส่วนนี้จะเกิดขึ้นเมื่อ User ไม่ต้องการจะเล่นเกมต่อแล้ว และต้องการที่จะดูคำตอบของเกม หรือ ผู้เล่นมีโจทย์อยู่แล้วและต้องการให้ระบบหาคำตอบให้ ในการหาคำตอบของระบบจะมีวิธีการหาคำตอบ 2 วิธีด้วยกันคือ

1. วิธีการแบบทั่วไปในที่นี้จะใช้วิธีการ Search เพื่อหาคำตอบที่เป็นไปได้ในทุกๆ กรณีที่สามารถเกิดขึ้นได้
2. ใช้อัลกอริทึม Geometric Particle Swarm Optimization ในการหาคำตอบ

ในการหาคำตอบของ Sudoku นั้นระบบอาจจะหาคำตอบของเกมได้มากกว่า 1 คำตอบก็ได้ ซึ่งขึ้นอยู่กับโจทย์

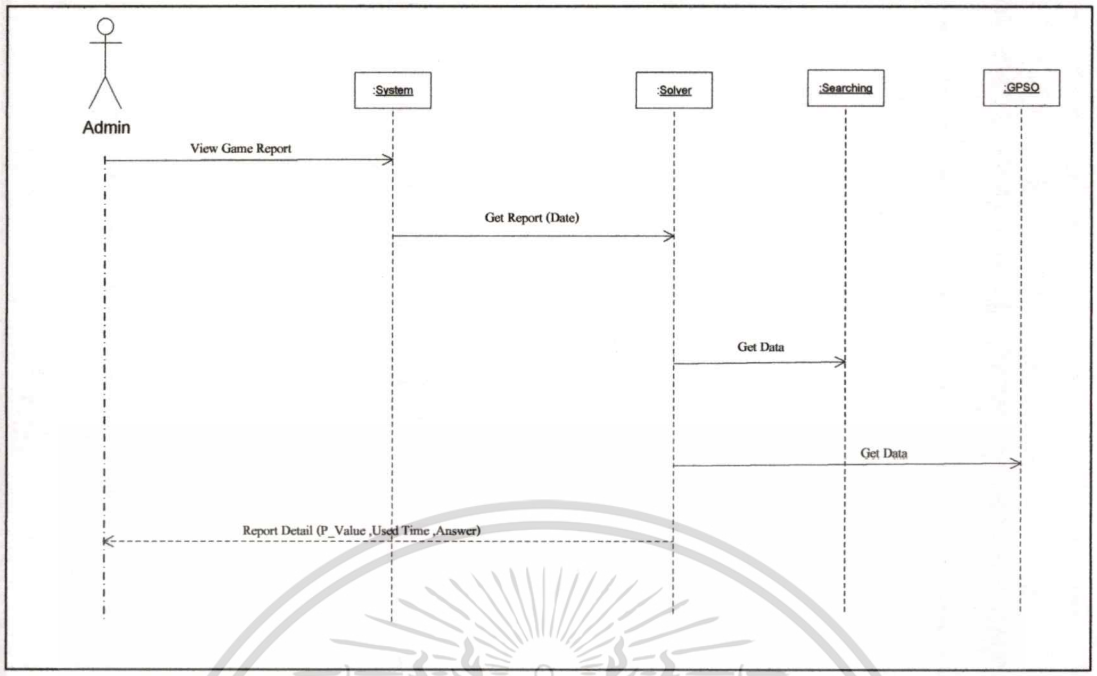
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Sequence Diagram ของการเพิ่มโจทย์ของ Sudoku เข้าไปในระบบ

จากรูป Sequence ของการเพิ่มโจทย์ของ Sudoku เข้าไปในระบบ ผู้ที่สามารถทำขั้นตอนนี้ได้จะต้องเป็น Admin ของระบบเท่านั้น โดยระบบจะมีการถาม Password ของ admin ก่อนที่จะเข้าไปทำงานในส่วนนี้ได้ เมื่อผ่านขั้นตอนดังกล่าวไปแล้ว Admin จะต้องใส่โจทย์ใหม่ผ่านทางหน้าจอที่ระบบแสดงขึ้นเมา เมื่อทำการใส่ข้องมูลเสร็จแล้ว ระบบจะมีการตรวจสอบว่าโจทย์ที่ใส่เข้าไปนั้นขัดต่อข้องบังคับ 3 ข้อของ Sudoku หรือไม่ ถ้าตรงตามข้อบังคับทั้ง 3 แล้ว Admin จะต้องมีการกำหนดระดับความยากของเกมที่เพิ่มเข้าไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 Sequence Diagram ของการดูรายงานการหาคำตอบของ Sudoku ด้วยวิธีต่างๆ

จากรูป Sequence Diagram ของการดูรายงานการหาคำตอบของ Sudoku ด้วยวิธีต่างๆ ผู้ที่สามารถทำขั้นตอนนี้ได้จะต้องเป็น Admin ของระบบเท่านั้น โดยระบบจะมีการถาม Password ของ admin ก่อนที่จะเข้าไปทำงานในส่วนนี้ได้ โดยระบบจะมีการดึงข้อมูลต่างๆที่จำเป็นในการออกรายงานจาก Class ต่างๆดังนี้ Solver Searching และ GPSO เมื่อดึงข้อมูลมาแล้ว ระบบจะทำการประมวลผลข้อมูล แล้วทำการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization

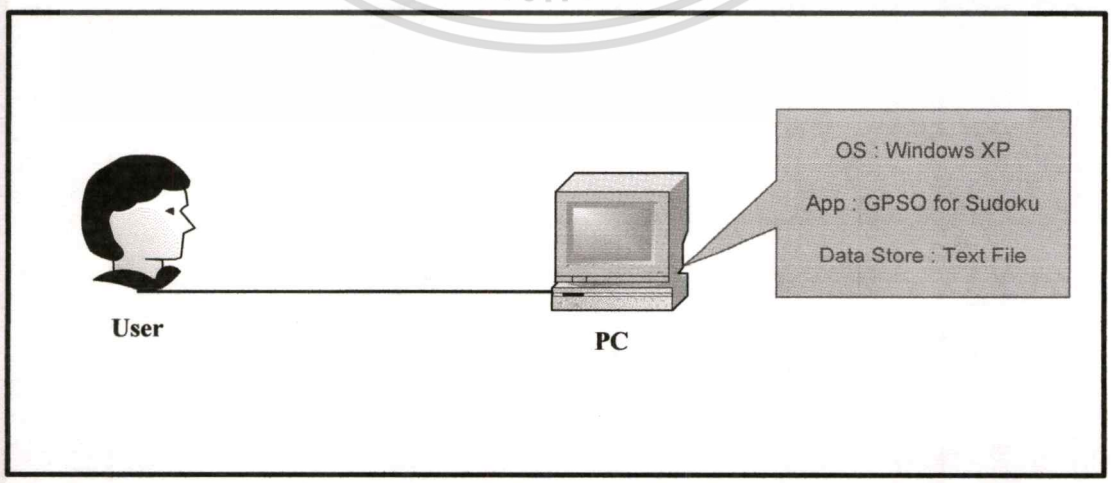
ในบทที่ 4 นี้จะกล่าวถึงการออกแบบระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization และออกแบบโปรเซสในการทำงานของระบบ ซึ่งประกอบด้วย

1. การออกแบบสถาปัตยกรรมของระบบ
2. การออกแบบข้อมูลของระบบ
3. การออกแบบ User Interface

4.1 การออกแบบสถาปัตยกรรมของระบบ

สถาปัตยกรรมของระบบคำนวณหาคำตอบของ Sudoku Puzzle โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization มีรายละเอียดดังนี้

1. ระบบได้ถูกพัฒนาในรูปแบบของ Application Program
2. ทำงานอยู่บนเครื่อง Personal computer ในลักษณะ Standalone
3. ลักษณะของการเก็บข้อมูลที่จำเป็นของระบบ มีการจัดเก็บข้อมูลในลักษณะ Text File เพื่อความสะดวกในการติดตั้งระบบ

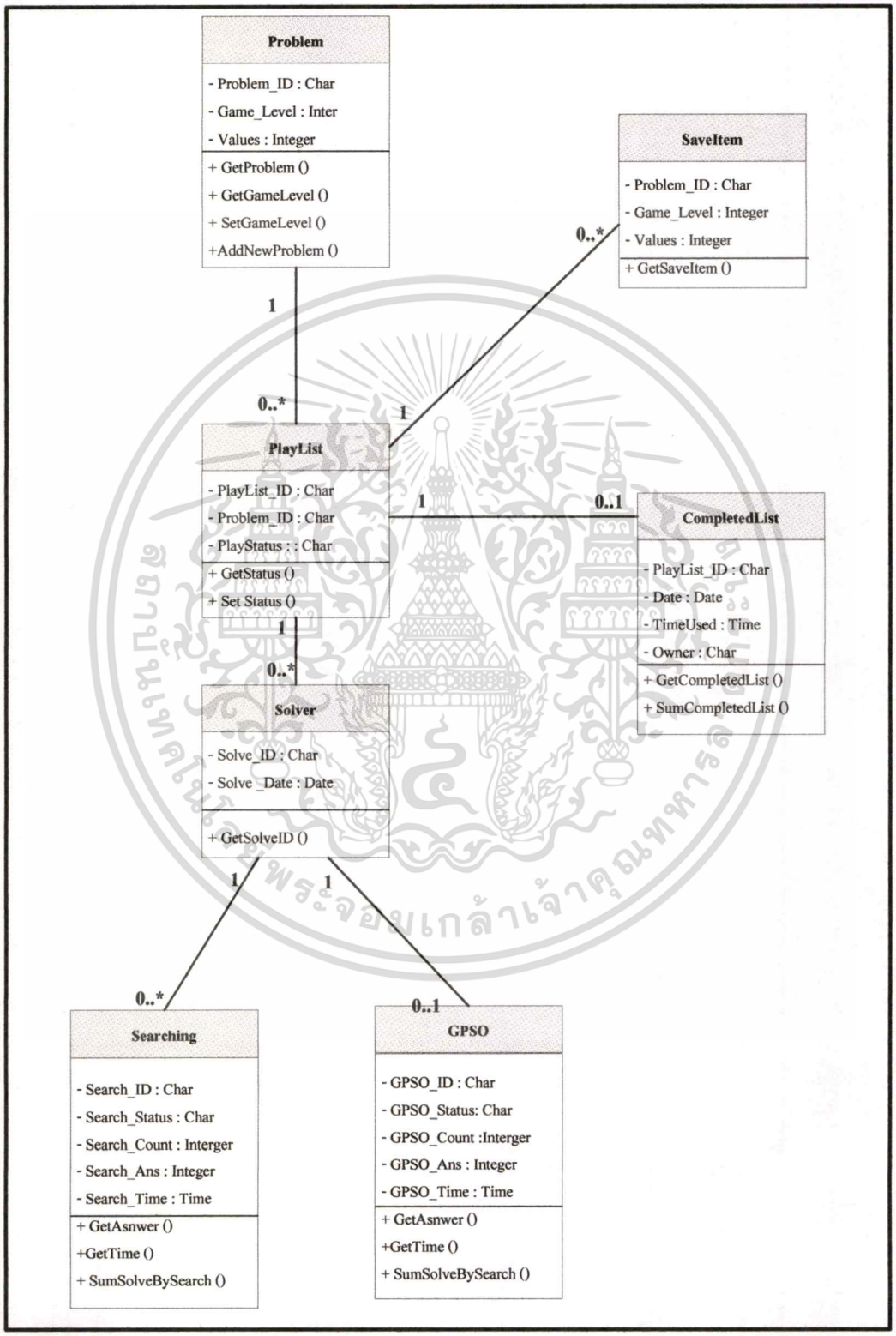


รูปที่4.1 แสดงการออกแบบสถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบข้อมูลของระบบ

ในส่วนนี้จะใช้ UML Design Class Diagram ในการอธิบายการออกแบบข้อมูลของระบบ ดังรูปที่ 4.2

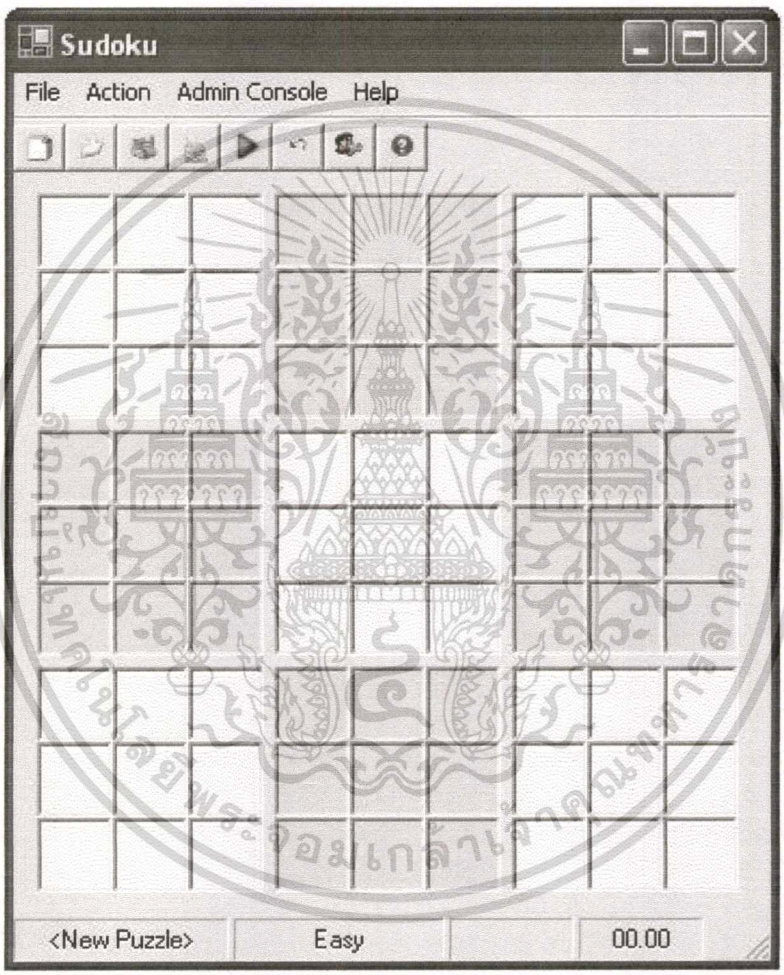


รูปที่4.2 แสดงการออกแบบข้อมูลของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบ User Interface

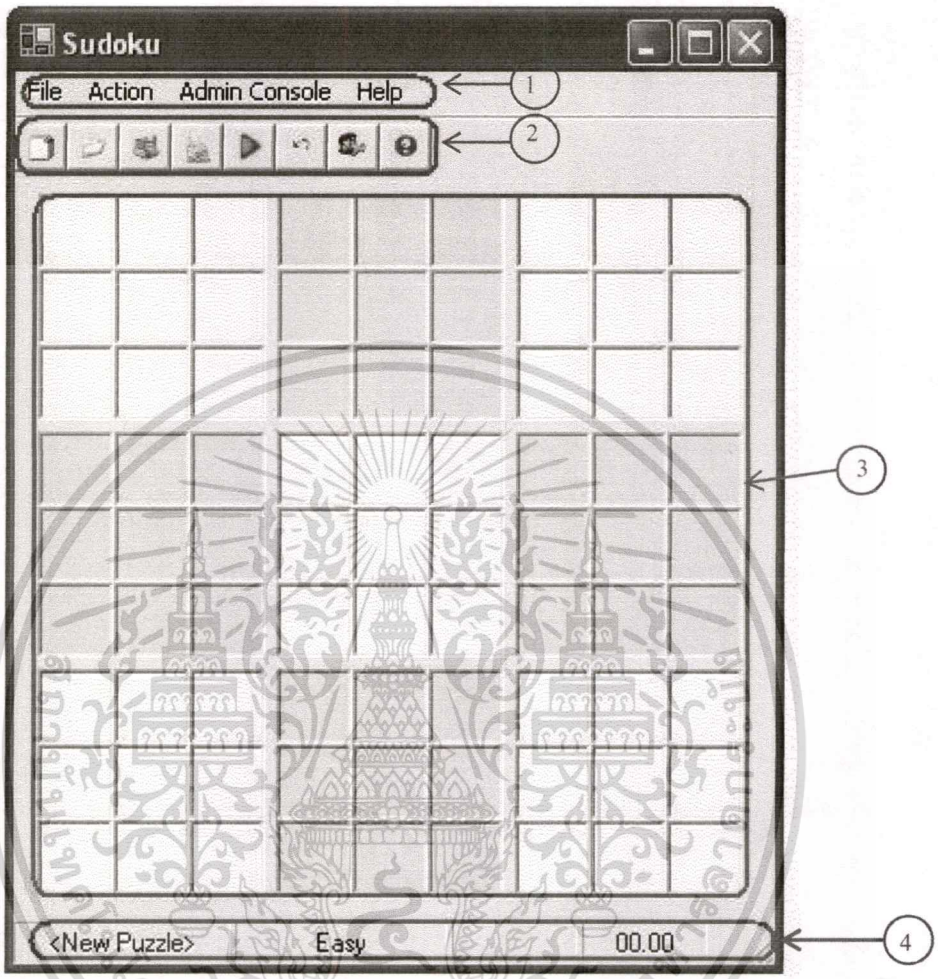
ในส่วนนี้จะกล่าวถึงการออกแบบหน้าจอ (User Interface) ของระบบ เนื่องจากระบบที่พัฒนาขึ้นมานั้นเป็นเกม จึงทำให้รูปแบบในการออกแบบหน้าจอไม่ค่อยแตกต่างจากระบบที่มีอยู่มากนัก แต่จะมีความแตกต่างในส่วนของอัลกอริทึ่มในการทำงานเสียมากกว่า การออกแบบหน้าจอหลักของระบบ แสดงดังรูปที่4.3



รูปที่4.3 แสดงการออกแบบหน้าจอหลักของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าในการออกแบบหน้าจอหลักของระบบนั้นจะมีการแบ่งส่วนการทำงานออกเป็น 4 ส่วน ดังรูปที่4.4



รูปที่4.4 แสดงการออกแบบหน้าจอและองค์ประกอบหลักของระบบ

จากรูปจะเห็นว่าองค์ประกอบหลักของหน้าจอประกอบด้วย 4 ส่วนด้วยกันคือ

1. หมายเลข 1 เป็นส่วนของ Main Menu ก็จะประกอบด้วย File Menu, Action Menu, Admin console Menu และ Help Menu ซึ่งในแต่และส่วนการทำงานก็จะประกอบไปด้วยองค์ประกอบย่อยๆของส่วนนั้นๆ ขึ้นอยู่กับฟังก์ชันการทำงานของส่วนต่างๆตามความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หมายเลข 2 เป็นส่วนของ Toolbar เป็นส่วนที่เพิ่มความสะดวกในการใช้งานระบบให้แก่ User โดยมีปุ่มที่ใช้กดเพื่อเป็นทางลัดในการใช้งานฟังก์ชันของระบบ โดยมากแล้วปุ่มใน Toolbar จะเป็นฟังก์ชันที่ใช้งานกันบ่อยๆ จากรูปจะเห็นว่า Toolbar ประกอบด้วย

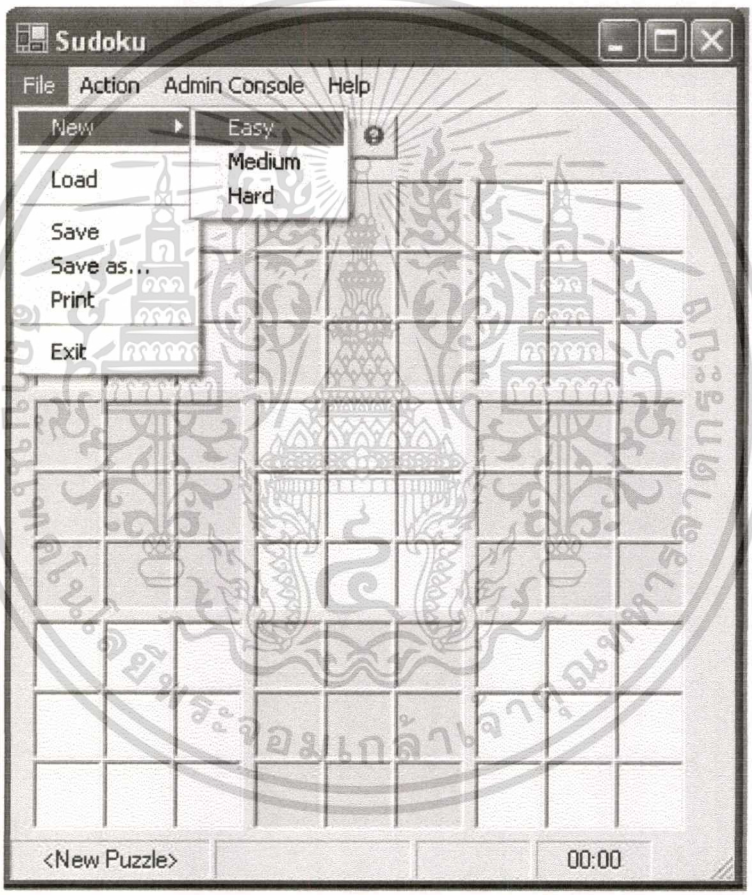
- เมื่อทำการกดที่ปุ่มนี้จะเป็นการเริ่มต้นเกมใหม่
- เมื่อทำการกดที่ปุ่มนี้จะเป็นการเปิดไฟล์ที่ได้ทำการบันทึกไว้เพื่อนำมาเล่นต่อ
- เมื่อทำการกดที่ปุ่มนี้ระบบจะทำการบันทึกข้อมูลของเกมที่ผู้เล่นกำลังเล่นอยู่ขณะนั้นเพื่อใช้ในการเล่นในครั้งต่อไป
- เป็นปุ่มที่ใช้ในการพิมพ์โจทย์ของ Sudoku
- เมื่อทำการกดที่ปุ่มนี้ระบบจะทำการทดสอบคำตอบของ User ว่าถูกต้องหรือไม่
- เป็นปุ่มที่ใช้ในการ Undo
- เป็นปุ่มที่ใช้ในการ Login เพื่อเข้าใช้งานในส่วนของ Admin Console
- เป็นปุ่มที่ใช้เรียก Help ของระบบ

3. หมายเลข 3 เป็นส่วนของพื้นที่สำหรับเล่นเกม
4. หมายเลข 4 เป็นส่วนของ Status bar ซึ่งจะประกอบด้วย
   - ส่วนที่บอกสถานะของเกม ตัวอย่างเช่น New Puzzle
   - ส่วนที่ใช้บอกระดับความยากของเกม ตัวอย่างเช่น Easy
   - และส่วนสุดท้ายเป็น Timer บอกเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนของ File Menu ก็จะประกอบด้วย 6 ส่วนด้วยกันดังรูปที่ 4.5

1. New เป็นส่วนของการเริ่มต้นเล่นเกมใหม่
2. Load เป็นส่วนของการดึงข้อมูลที่เคยบันทึกไว้ขึ้นมาเพื่อทำการเล่นต่อ
3. Save เป็นส่วนของการบันทึกข้อมูลขณะนั้นไว้
4. Save As เป็นการบันทึกข้อมูลคล้ายๆกับ Save
5. Print เป็นการพิมพ์โจทย์ Sudoku
6. Exit เป็นส่วนที่ใช้ในการออกจากระบบ



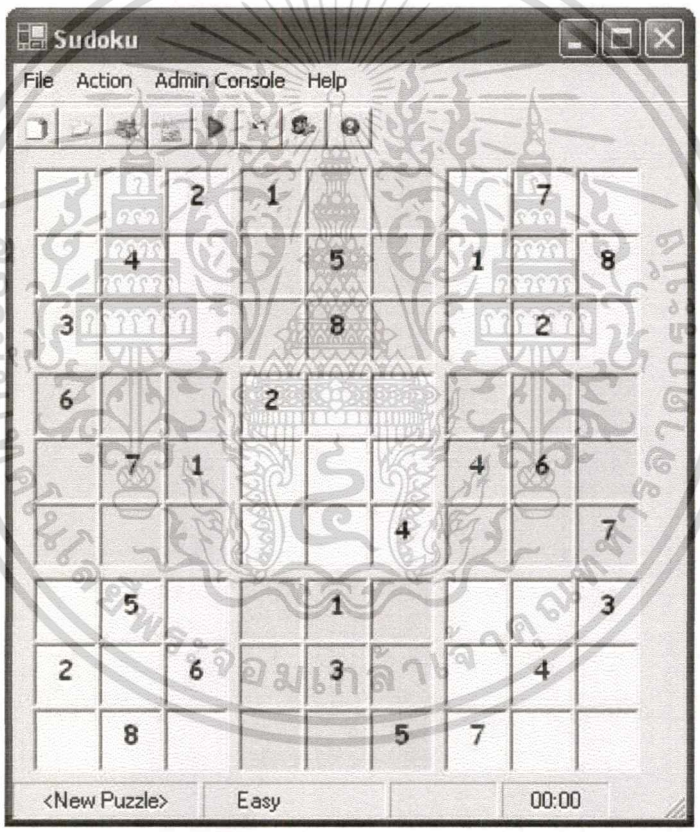
รูปที่4.5 แสดงการออกแบบหน้าจอในส่วนของ File Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าในส่วนของ New ซึ่งเป็นการเริ่มต้นเกมใหม่ ซึ่งจะต้องมีการกำหนดระดับความยากของเกมด้วย ซึ่งระดับความยากของเกมจะประกอบด้วย 3 ระดับคือ

- Easy ระดับความยากที่น้อยที่สุด
- Medium ระดับความยากระดับปานกลาง
- Hard รับความยากระดับสูงสุด

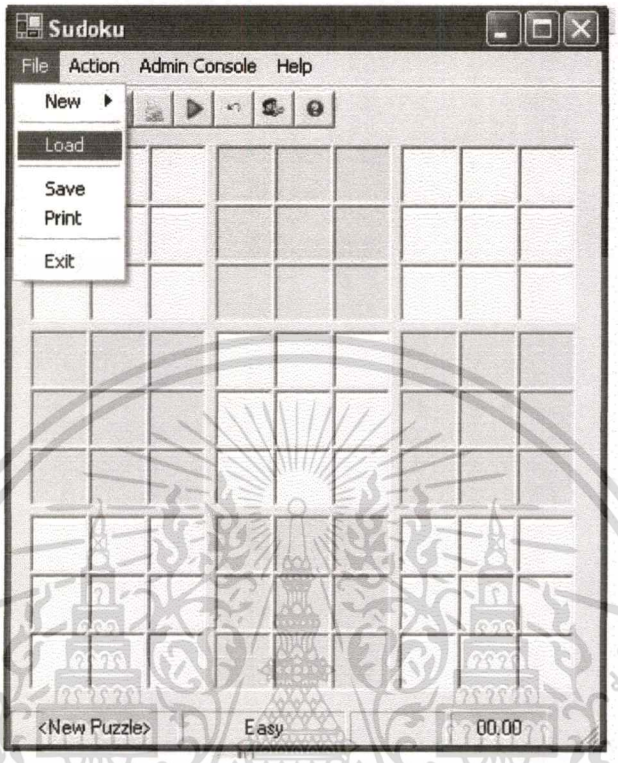
หลังจากที่ทำการเลือกระดับความยากของเกมแล้วระบบจะทำการสุ่มโจทย์แล้วทำการแสดงผลเพื่อให้ผู้เล่นเริ่มต้นเล่นเกมได้ ดังรูปที่ 4.6



รูปที่4.6 แสดงการออกแบบหน้าจอในส่วนของการเริ่มต้นเกมใหม่

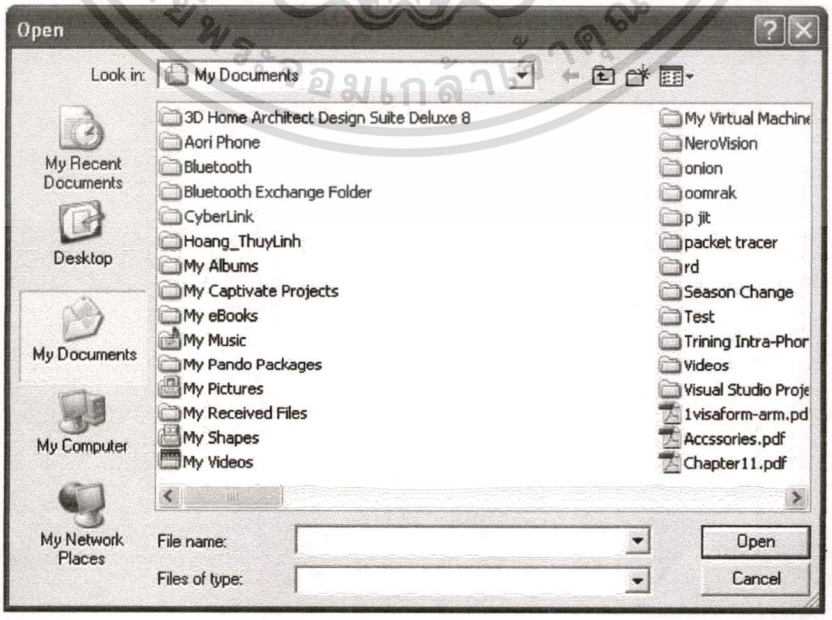
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบหน้าจอในส่วนของการ Load เกมเป็นการดึงข้อมูลที่เคยบันทึกไว้เพื่อทำการเล่นเกมต่อ แสดงดังรูปที่ 4.7



รูปที่4.7 แสดงการออกแบบหน้าจอในส่วนของการ Load เกม

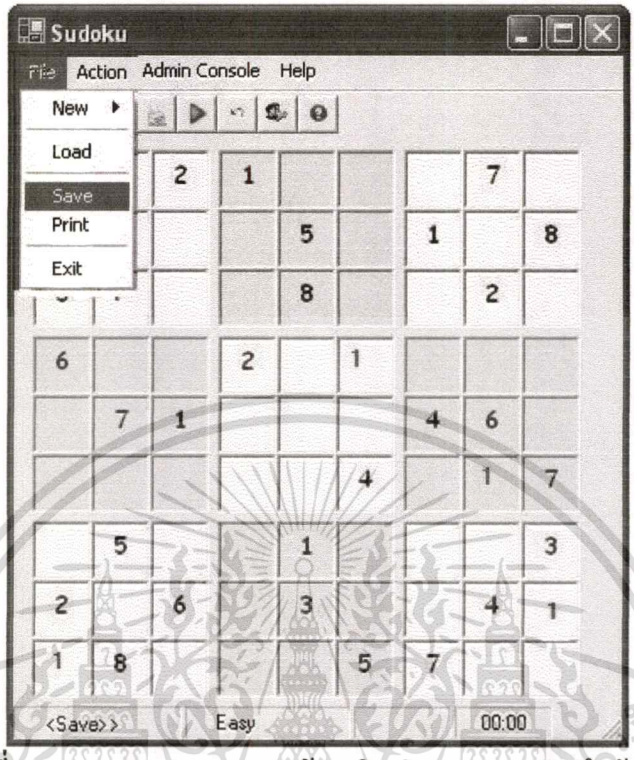
หลังจากที่กดที่ปุ่ม Load เกมแล้วระบบจะแสดงหน้าจอให้ทำการเลือกไฟล์ที่ต้องการโหลด ดังรูปที่ 4.8



รูปที่4.8 แสดงการออกแบบหน้าจอในส่วนของการเลือกไฟล์ที่ต้องการ Load

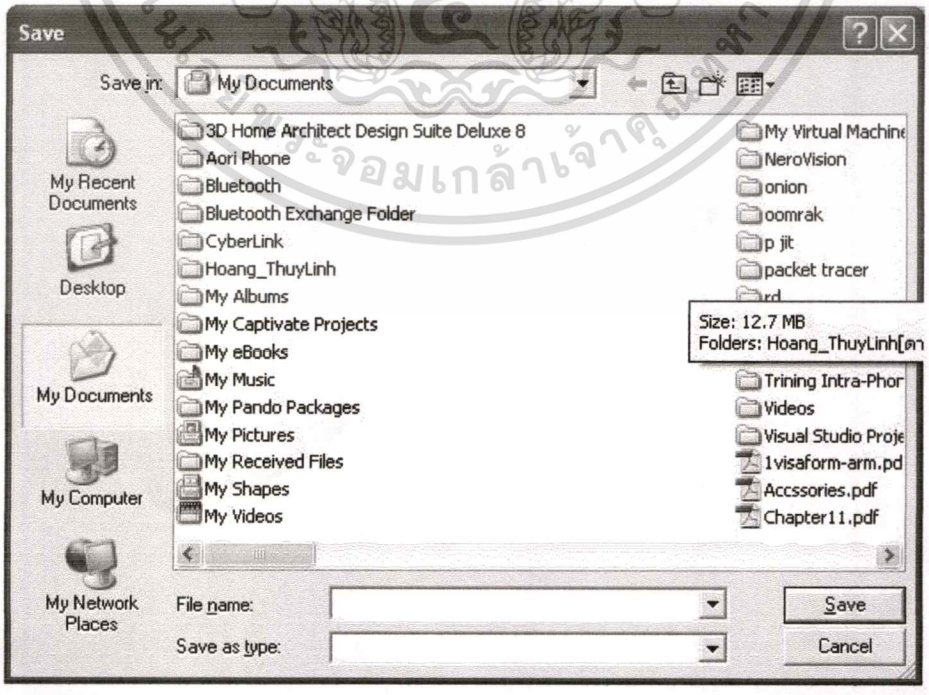
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เล่นทำการเล่นเกมไประหนึ่งแล้วต้องการทำการบันทึกไว้เล่นต่อ ต้องต้องเลือกที่ File Menu → Save ดังรูปที่ 4.9



รูปที่4.9 แสดงการออกแบบหน้าจอในส่วนของการบันทึกข้อมูล

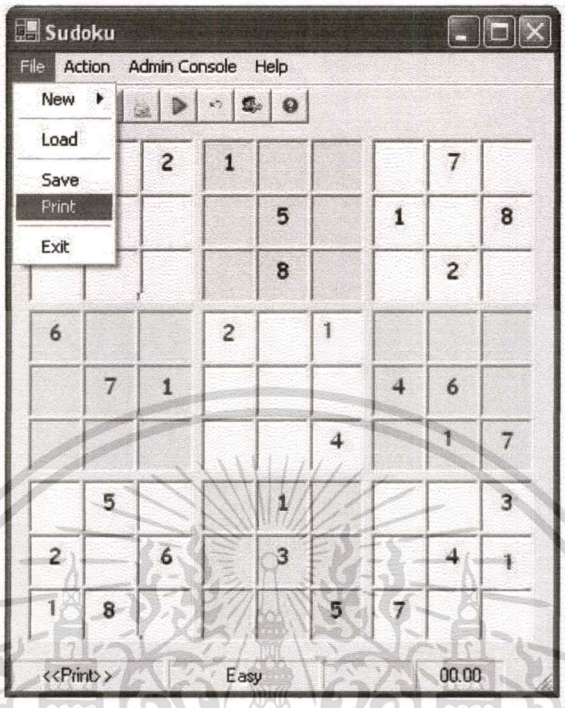
หลังจากนั้นระบบจะแสดงหน้าจอเพื่อให้ทำการใส่ชื่อไฟล์ที่ต้องการบันทึกข้อมูล ดังรูปที่ 4.10



รูปที่4.10 แสดงการออกแบบหน้าจอในส่วนของการใส่ชื่อไฟล์ที่ต้องการบันทึก

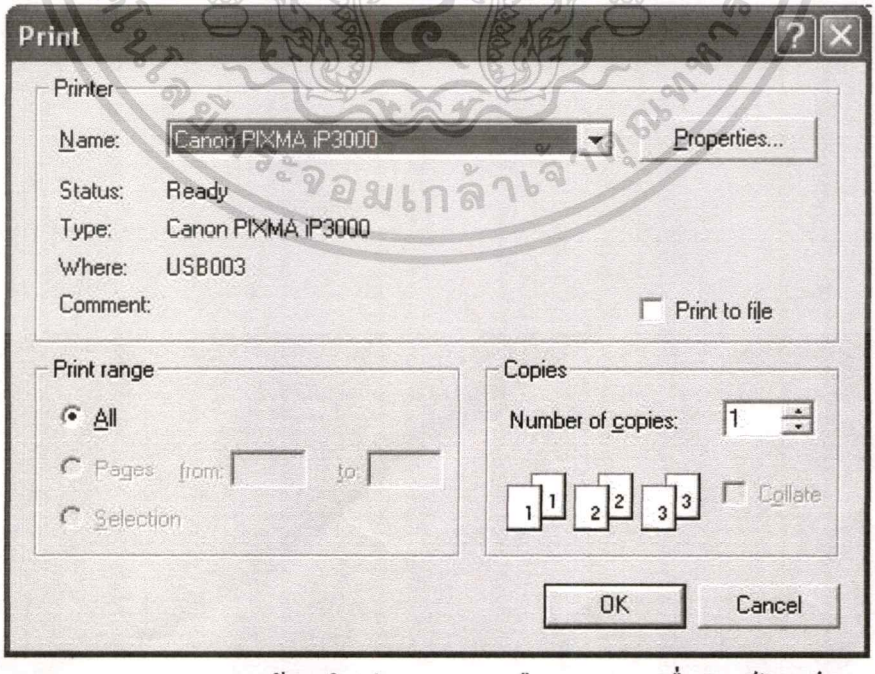
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เล่นทำการเล่นต้องการทำการพิมพ์โจทย์ของ Sudoku ต้องต้องเลือกที่ File Menu → Print ดังรูปที่ 4.11



รูปที่4.11 แสดงการออกแบบหน้าจอในส่วนของการพิมพ์โจทย์ของ Sudoku

หลังจากนั้นระบบจะแสดงหน้าจอเพื่อให้ทำการเลือก Printer ที่เราต้องการจะพิมพ์เอกสารออก ดังรูปที่ 4.12

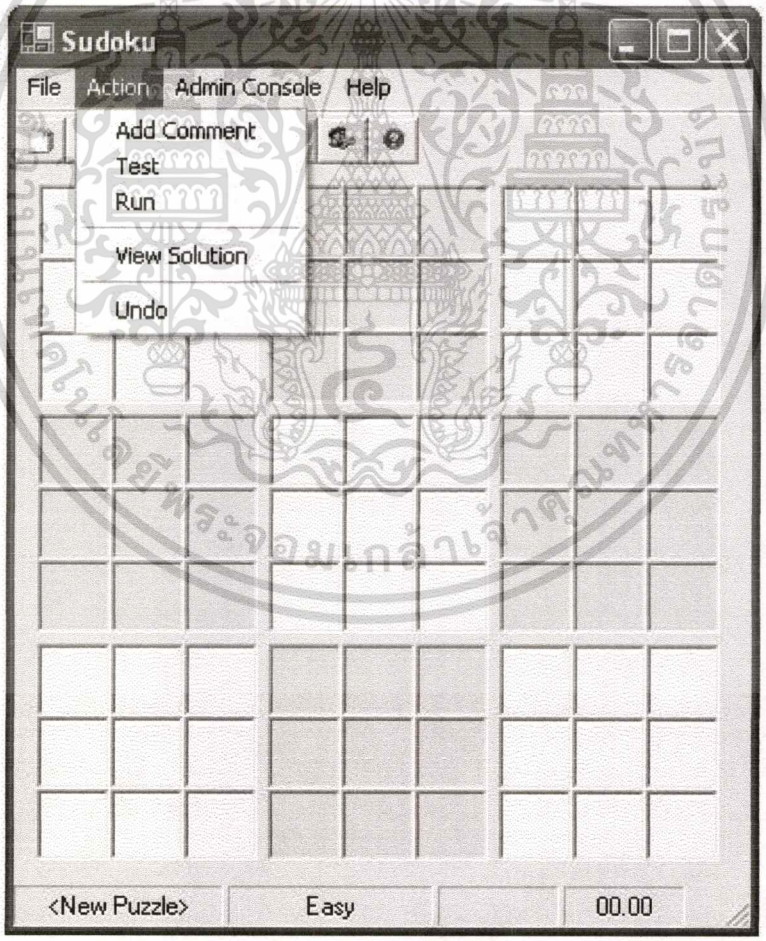


รูปที่4.12 แสดงการออกแบบหน้าจอในส่วนของการเลือก Printer เพื่อพิมพ์โจทย์ของ Sudoku

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนของ Action ก็จะประกอบด้วย 5 ส่วนด้วยกันดังรูปที่ 13

1. Add Comment เป็นส่วนของการเพิ่ม Comment ในเกมซึ่งจะเกิดขึ้นเมื่อผู้เล่นยังไม่แน่ในคำตอบของช่องนั้นๆ
2. Test เป็นส่วนของการทดสอบว่าคำตอบที่ใส่ไปนั้นถูกต้องตามกดของ Sudoku หรือไม่
3. Run เป็นส่วนของการตรวจสอบคำตอบว่าถูกต้องหรือไม่โดยเมนูนี้จะใช้ได้ก็ต่อเมื่อผู้เล่นใส่คำตอบของ Sudoku ครบถ้วนทุกๆช่องแล้ว
4. View Solution ในกรณีนี้เกิดขึ้นเมื่อผู้เล่นต้องการให้ระบบเป็นตัวหาคำตอบของ Sudoku ให้ ซึ่งระบบจะมีวิธีในการหาคำตอบ 2 วิธีการด้วยกันคือ ใช้หลักการใน Searching ในการหาคำตอบ และใช้อัลกอริทึม GPSO ในการหาคำตอบ
5. Undo เป็นส่วนที่ใช้ในการยกเลิกการรายการที่ทำไปก่อนหน้านี้

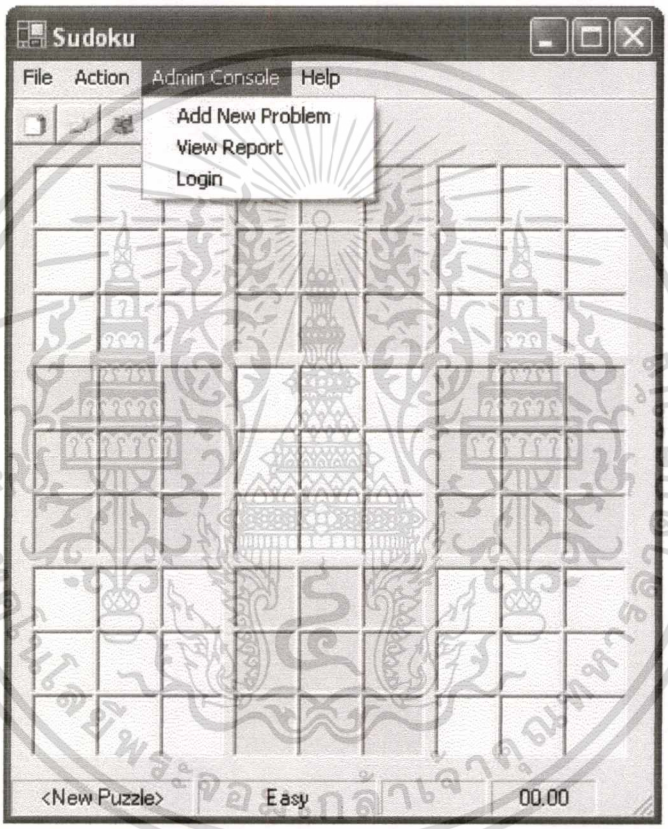


รูปที่4.13 แสดงการออกแบบหน้าจอในส่วนของ Action Menu

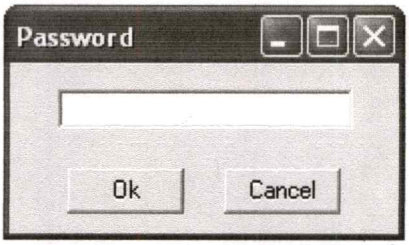
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนของ Admin Console จะประกอบด้วย 3 ส่วนด้วยกันดังรูปที่ 4.14

1. Login เป็นส่วนที่ใช้ในการ Login เข้าไปก่อนจะใช้งานในส่วนของ Add New Problem และส่วนของ View Report ซึ่งจะแสดงหน้าจอดังรูปที่ 4.15
2. Add New Problem เป็นส่วนของการเพิ่มโจทย์ใหม่ๆเข้าไปในระบบ
3. View Report เป็นส่วนของการดูรายงานของระบบที่ใช้ในการหาคำตอบของ Sudoku



รูปที่4.14 แสดงการออกแบบหน้าจอในส่วนของ Admin Console Menu

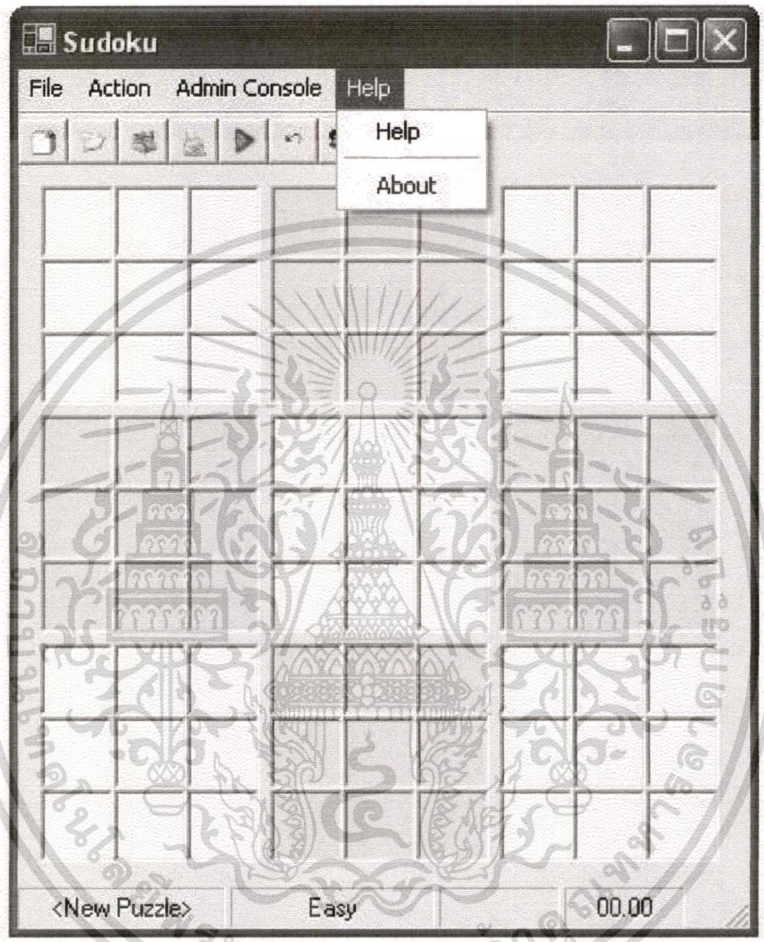


รูปที่4.15 แสดงการออกแบบหน้าจอในส่วนของการ Login เพื่อเข้าใช้งานส่วนของ Admin Console

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนของ Help จะประกอบด้วย 2 ส่วนด้วยกันดังรูปที่ 4.16

1. Help เป็นส่วนที่ใช้ในการดูวิธีการใช้งานระบบ
2. About เป็นส่วนของการดู Version ของระบบ



รูปที่4.17 แสดงการออกแบบหน้าจอในส่วนของ Help Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

ในปัจจุบันมีแนวโน้มด้านการพัฒนาระบบคอมพิวเตอร์ ได้เน้นเรื่องการที่จะพัฒนาคอมพิวเตอร์ให้มีความสามารถใกล้เคียงกับมนุษย์เป็นสำคัญ เช่น การเพิ่มคามสามารถให้ระบบคอมพิวเตอร์เข้าใจภาษาสื่อสารของมนุษย์ หรือระบบโครงข่ายประสาทเทียม รวมถึงระบบเสมือนจริง เป็นต้น โดยพยายามนำไปใช้ให้เกิดประโยชน์ในด้านต่างๆมากขึ้น และพยายามพัฒนาระบบดังกล่าวให้เกิดข้อผิดพลาดในการทำงานให้ลดน้อยลง

ระบบปัญญาประดิษฐ์ (Artificial Intelligence System) เป็นการพัฒนาระบบคอมพิวเตอร์ให้มีความสามารถในการตอบสนองกับความต้องการของมนุษย์ได้ ให้มีพฤติกรรมเลียนแบบมนุษย์ มีความเข้าใจภาษามนุษย์ รับรู้ได้และตอบสนองด้วยการแสดงออกทางพฤติกรรมและภาษามนุษย์ได้ รวมถึงความสามารถในการจดจำ และการวิเคราะห์ แก้ไข ปัญหาต่างๆได้

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการคำนวณเพื่อหาคำตอบของ Sudoku Puzzle ซึ่งมีส่วนหนึ่งของระบบเป็นส่วนที่ให้ระบบคอมพิวเตอร์นั้นคิดคำนวณ แก้ปัญหา โจทย์ของ Sudoku Puzzle ซึ่งเป็นการเพิ่มความสามารถให้ระบบคอมพิวเตอร์สามารถคิดเพื่อแก้ไขปัญหาที่ซับซ้อนได้ โดยเลียนแบบพฤติกรรมในการคิดแก้ไขปัญหาของมนุษย์ โดยใช้อัลกอริทึ่ม Geometric Particle Swarm Optimization (GPSO) ซึ่งพัฒนามาจากอัลกอริทึ่มการหาค่าเหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) และอีกวิธีการหนึ่งจะใช้วิธีการ Searching คือใช้การค้นหาคำตอบของระบบโดยการลองใส่ค่าทุกค่าที่เป็นไปได้แล้วตรวจสอบว่าค่าที่ใส่ไปนั้นขัดแย้งต่อข้อบังคับของ Sudoku หรือไม่ แล้วจะสนใจเฉพาะค่าที่ใส่ไปแล้วไม่ขัดแย้งต่อข้อบังคับของ Sudoku

จากการพัฒนาระบบทำให้สามารถสรุปได้ว่า วิธีการทั้งสองวิธีนั้นสามารถใช้ในการหาคำตอบของ Sudoku ได้ทั้งสองวิธี แต่การหาคำตอบของ Sudoku ด้วยอัลกอริทึ่ม GPSO นั้นให้ผลที่ค่อนข้างต่ำ 15% ถ้าเปรียบเทียบกับการใช้ Rule base ผสมกับ Search ซึ่งให้ผลลัพธ์ 100 % พร้อมทั้งเวลาที่ใช้ในการหาคำตอบที่น้อยมาก

เทคนิคที่ใช้ในการพัฒนาระบบที่เกี่ยวกับปัญญาประดิษฐ์นั้นมีหลากหลายวิธีการด้วยกัน การเลือกใช้อัลกอริทึ่มขึ้นอยู่กับระบบที่พัฒนา และความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

A. Moraglio, R. Poli. 2005. "Geometric Crossover for the permutation Representation." **Technical Report CSM.**

Alberto Moraglio, Julian Togelius. 2007. "Geometric Particle Swarm Optimization for the Sudoku Puzzle." **GECCO 2007 Workshop.**

Alberto Moraglio, Julian Togelius, Simon Lucas. 2006. "Product Geometric Crossover for Sudoku Puzzle." **IEEE.**

James Kennady, Russell Eberhart. 1997. "Particle Swarm Optimization." **IEEE.**

Michale O' Neill, Anthony Brabazon. 2006. "Particle Swarm Optimization for Unsupervised Learning." **IEEE.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวอรพิณ โพธิกูล
วัน เดือน ปีเกิด	31 ตุลาคม พ.ศ.2524
ประวัติการศึกษา	ปริญญาตรี วิทยาศาสตร์บัณฑิต
	สาขา วิทยาการคอมพิวเตอร์
	คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล
ตำแหน่งงานปัจจุบัน	System Engineer
สถานที่ทำงาน	บริษัท ISONET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้