

การใช้การปรับอัตราค่าก่อสร้างทรัพยากรและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงในการปรับความยากง่ายของ
เกมวางแผนการรบแบบตบสนองทันกาล



นายปิยพจน์ เกษมภักดีพงษ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2553

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

UTILIZING RESOURCE PRODUCTION ADJUSTMENT AND TOP CULLING ENHANCEMENT FOR
DIFFICULTY BALANCING IN REAL-TIME STRATEGY GAMES

Mr. Piyapoj Kasempakdeepong



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2010
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การใช้การปรับอัตราการสร้างทรัพยากรและการปรับปรุง
วิธีการตัดค่าน้ำหนักที่สูงในการปรับความยากง่ายของ
เกมวางแผนการรบแบบตบสนองทันกาล

โดย

นายปิยพจน์ เกษมภักดีพงษ์


สาขาวิชา

วิศวกรรมคอมพิวเตอร์


อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

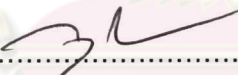
ผู้ช่วยศาสตราจารย์ ดร.วิษณุ โคตรจรัส

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต


..... คณะบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)


..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.วิษณุ โคตรจรัส)


..... กรรมการ
(อาจารย์ ดร.นัทธี นิภานันท์)


..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.สุรพงษ์ เลิศสิทธิชัย)

ปิยพจน์ เกษมภักดีพงษ์ : การใช้การปรับอัตราการผลิตทรัพยากรและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงในการปรับความยากง่ายของเกมวางแผนการรบแบบตอบสนองทันทีกาล. (UTILIZING RESOURCE PRODUCTION ADJUSTMENT AND TOP CULLING ENHANCEMENT FOR DIFFICULTY BALANCING IN REAL-TIME STRATEGY GAMES) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.วิษณุ โคตรจรัส, 149 หน้า.

งานวิจัยเกี่ยวกับเกมวางแผนการรบแบบตอบสนองทันทีกาลในปัจจุบันส่วนใหญ่มุ่งเน้นไปที่การทำให้ปัญญาประดิษฐ์เล่นได้เก่งขึ้นจนสามารถที่จะเอาชนะผู้เล่นได้ แต่ผู้เล่นแต่ละคนมีความสามารถที่แตกต่างกัน การพัฒนาปัญญาประดิษฐ์ที่มีความสามารถคงที่ อาจทำให้ผู้เล่นที่เก่งกว่าปัญญาประดิษฐ์เบื่อหน่ายเพราะเกมง่ายเกินไป ในขณะที่ผู้เล่นที่อ่อนกว่าปัญญาประดิษฐ์อาจรู้สึกว่ายากเกินไป ซึ่งอาจทำให้ผู้เล่นเลิกเล่นเกม นั้น ดังนั้นการทำให้ปัญญาประดิษฐ์สามารถปรับตัวให้มีความสามารถทัดเทียมกับผู้เล่นจึงมีความสำคัญ วิทยานิพนธ์นี้นำเสนอปัญญาประดิษฐ์ที่ใช้อัตราการผลิตยูนิตในการปรับความยากให้ทัดเทียมกับผู้เล่น โดยได้ใช้เกมสปริงซึ่งเป็นเกมแบบประเภทเปิดเผยแพร่โค้ดเป็นตัวอย่างทดสอบ และได้ใช้ฟังก์ชันวัดคะแนนที่ถูกสร้างขึ้นร่วมกับเทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการตัดค่าน้ำหนักที่สูงมาปรับปรุงให้สามารถปรับความยากให้เข้ากับผู้เล่นขณะที่ผู้เล่นกำลังเล่นเกมอยู่ได้ ผลการทดลองสรุปได้ว่าอัตราการผลิตยูนิตสามารถนำมาใช้ในการปรับความยากได้ แต่ยังคงควรมีการปรับปรุงเพื่อให้ไดนามิกสคริปต์สามารถใช้อัตราการผลิตยูนิตในการปรับความยากได้ดีขึ้น วิทยานิพนธ์นี้จึงเสนอวิธีการปรับปรุงไดนามิกสคริปต์ให้เหมาะสมกับการใช้อัตราการผลิตยูนิตเป็นกฎย่อยในกฎพื้นฐานอีก 4 วิธีการ ผลการทดลองสรุปได้ว่าการปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูนั้นมีประสิทธิภาพดีที่สุดเมื่อเทียบกับไดนามิกสคริปต์แบบดั้งเดิมและวิธีอื่นที่นำเสนอ ทั้งในแง่ของเวลาเฉลี่ยที่สูงขึ้น และผลการ ชนะ/แพ้ ทำเกมที่ใกล้เคียงกัน

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2553

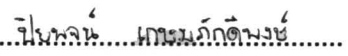
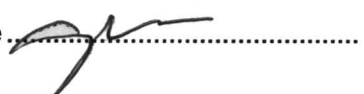
ลายมือชื่อนิสิต ปิยพจน์ เกษมภักดีพงษ์
ลายมือชื่อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....

5070351121 : MAJOR COMPUTER ENGINEERING

KEYWORDS : PRODUCTION RATE / REAL-TIME STRATEGY GAMES / ARTIFICIAL INTELLIGENCE

PIYAPOJ KASEMPAKDEEPPONG : UTILIZING RESOURCE PRODUCTION ADJUSTMENT AND TOP CULLING ENHANCEMENT FOR DIFFICULTY BALANCING IN REAL-TIME STRATEGY GAMES. ADVISOR : ASST. PROF. VISHNU KOTRAJARAS, Ph.D., 136 pp.

Most researches in the real-time strategy games focus on Artificial Intelligence that beats a human opposition. However, each human player has different skill. An Artificial Intelligence with certain skill level causes skillful players to become bored of the game because it is too easy for them. At the same time, new players may find the game too difficult. Consequently, players may stop playing the game altogether. This thesis presents a novel Artificial Intelligence that uses unit production to adjust appropriate game difficulty for players. This thesis uses the open-source real-time strategy engine, Spring, as its testing environment. Score functions are generated automatically to obtain fitness function for dynamic scripting and top culling. Our adjustment, unlike other researches, applies within a playing session instead of between sessions. The result shows that a gaming session can be adjusted to suit players' abilities without any need to tune the behavioral script of the game. This thesis also proposes 4 techniques for improving dynamic scripting and top culling so that rules with unit production can be used to their fullest potential to produce even matches. The results shows that the method of changing the number of selectable rules in a script according to the opponent's performance produces longer average game time and closer to 50:50 win/lose ratio compared to using the original dynamic script and the other techniques.

Department..... Computer Engineering..... Student's Signature..... .....
 Field of Study..... Computer Engineering..... Advisor's Signature..... .....
 Academic Year..... 2010.....

กิตติกรรมประกาศ

ตลอดระยะเวลาในการจัดทำวิทยานิพนธ์ฉบับนี้ ได้มีอุปสรรคต่าง ๆ เกิดขึ้น นานัปการ อันเป็นบทเรียนที่ทรงคุณค่ายิ่งแก่ผู้จัดทำ เพื่อที่จะได้ฝึกฝน เรียนรู้ และแก้ไขปัญหามาตลอดจนได้เพิ่มพูนทักษะต่าง ๆ ที่จำเป็นสำหรับการวิจัย ซึ่งทั้งหมดนี้ล้วนเป็นปัจจัยที่ช่วยส่งเสริมและผลักดันศักยภาพให้แก่ผู้จัดทำเป็นอย่างมาก อย่างไรก็ตาม วิทยานิพนธ์ฉบับนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดแรงสนับสนุนจากบุคคลหลายฝ่าย ซึ่งข้าพเจ้าซาบซึ้งในความกรุณาเหล่านี้เป็นอย่างล้นพ้น และใคร่ขอใช้เนื้อหาในกิตติกรรมประกาศของวิทยานิพนธ์ฉบับนี้ เป็นสื่อกลางในการแสดงความขอบพระคุณอย่างสุดซึ้งจากผู้จัดทำ

ประการแรก ขอบพระคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ผู้ช่วยศาสตราจารย์ ดร.วิษณุ โคตรจรัส ผู้ซึ่งอบรม สั่งสอน ชี้แนะ และแก้ไขศิษย์คนนี้ด้วยดีเสมอมา อันเป็นปัจจัยหลักที่ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ฉบับนี้ ที่ให้ข้อคิดและข้อเสนอแนะต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการพัฒนาคุณภาพของวิทยานิพนธ์ฉบับนี้ ซึ่งคณะกรรมการสอบวิทยานิพนธ์นั้น ประกอบไปด้วย ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม ผู้ช่วยศาสตราจารย์ ดร.สุรพงษ์ เลิศสิทธิชัย และอาจารย์ ดร.นัทที นิภานันท์

ขอบคุณเพื่อน ๆ ในห้องปฏิบัติการทุกคนที่ช่วยให้ชีวิตในการทำวิจัยมีสีสันและมีความหมายมากยิ่งขึ้น รวมทั้งช่วยเสนอแนวคิดต่าง ๆ ในการแก้ไขปัญหา

สุดท้ายที่ขาดเสียมิได้ ขอบพระคุณครอบครัวที่น่ารักของผู้จัดทำทุก ๆ คน ที่เป็นกำลังใจ และให้การสนับสนุนทุกสิ่งทุกอย่างอย่างด้วยดีเสมอมา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฎ
สารบัญภาพ	ฏ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 ประโยชน์ที่ได้รับ	3
1.5 วิธีดำเนินการวิจัย	3
1.6 ผลงานตีพิมพ์จากงานวิจัย	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.1.1 เกมวางแผนการรบแบบตอบสนองทันที (Real-time Strategy Games)	5
2.1.2 สปริง (Spring)	6
2.1.3 สภาพแวดล้อมข้อมูลสมบูรณ์ (Perfect Information Environment) และสภาพแวดล้อมข้อมูลไม่สมบูรณ์ (Imperfect Information Environment)	7
2.1.4 ปัญญาประดิษฐ์ (Artificial Intelligence)	8
2.1.5 ฟังก์ชันวัดคะแนน (Score Function) หรือฟังก์ชันประเมินคุณค่า (Evaluation Function)	8
2.1.6 ฟังก์ชันวัดความเหมาะสม (Fitness Function)	9
2.1.7 การเคลื่อนลงตามความชัน (Gradient Descent)	9
2.1.8 Temporal Difference Learning	10
2.2 งานวิจัยที่เกี่ยวข้อง	10
2.2.1 ไดนามิกสคริปต์ (Dynamic Script) และวิธีการปรับความยากให้เข้ากับผู้เล่นแต่ละคน	11
2.2.2 ฟังก์ชันวัดคะแนนสำหรับเกมสปริง	18

บทที่ 3	วิธีการสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นโดยใช้การปรับการ ผลิตยูนิตในเกมวางแผนการรบแบบตอบสนองทันกาล	24
3.1	การใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงใน การสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นด้วยการควบคุม การผลิตยูนิต	24
3.1.1	การสร้างโมดูลเพื่อควบคุมการผลิตยูนิตในปัญญาประดิษฐ์ E323	24
3.1.2	การสร้างโมดูลสำหรับเก็บข้อมูลของเกม	25
3.1.3	การหาค่าน้ำหนักของยูนิต (w_u) โดยใช้การเรียนรู้แบบ Temporal Difference จากแผนที่ทดสอบ	25
3.1.4	การหาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟส ของเกม (w_p) จากแผนที่ทดสอบ	28
3.1.5	การนำฟังก์ชันวัตถุประสงค์แนบมาปรับใช้ในการทดลอง	30
3.1.6	การนำเทคนิคการเรียนรู้แบบไดนามิกสคริปต์มาใช้ในการเรียนรู้	30
3.1.7	ปรับระดับความยากของเกมให้เข้ากับผู้เล่นโดยใช้วิธีการตัดค่าน้ำหนักที่ สูง (Top Culling)	32
3.1.8	การเก็บค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานในแต่ละคู่ปัญญาประดิษฐ์ ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบ	33
3.2	การปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการ ใช้การปรับอัตราการสร้างทรัพยากรในการปรับความยากง่ายของเกมวางแผน แผนการรบแบบตอบสนองทันกาล	34
3.2.1	การเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุกๆ เวลาหนึ่ง	34
3.2.2	การปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรู	35
3.2.3	วิธีการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ ซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ แนบ ณ เวลาก่อนหน้า	35
บทที่ 4	การทดลองและวิเคราะห์ผล	37
4.1	การวัดประสิทธิภาพของปัญญาประดิษฐ์ที่สร้างขึ้น	37
4.1.1	เวลาเฉลี่ย	37
4.1.2	ผลการ ชนะ/แพ้ ทำเกม	37
4.2	การเตรียมการทดลอง	37
4.2.1	ปัญญาประดิษฐ์ทดสอบ	37
4.2.2	แผนที่ทดสอบ	38
4.2.3	ปัญญาประดิษฐ์ที่สร้างขึ้น	40
4.2.4	การเก็บผลการทดลอง	41

4.3 ผลการทดลอง.....	41
4.4 ทดสอบว่าการผลิตยูนิโต้ใช้ในการปรับความยากให้รู้สึกกับระดับของผู้เล่นได้หรือไม่เพียงใด.....	44
4.4.1 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - E323	44
4.4.2 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - AAI.....	45
4.4.3 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - RAI.....	46
4.5 ทดสอบประสิทธิภาพของการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรในการปรับความยากง่ายของเกมวางแผนการรบแบบตบสนองทันกาล	48
4.5.1 ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลานั้นกับเวลา ก่อนหน้า (Change w_{inc} , w_{dec}).....	48
4.5.2 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก 7 นาที และวิธีการตัดค่าน้ำหนักที่สูง (7-minute increment)	50
4.5.3 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules).....	52
4.5.4 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลานั้นกับเวลา ก่อนหน้า (Mix)	53
4.6 วิเคราะห์ฟังก์ชันวัตถุประสงค์กับผลการทดลอง.....	56
4.7 วิเคราะห์ผลของจุดเกิดกับการทดลองในแต่ละแผนที่.....	66
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ	69
5.1 สรุปผลการวิจัย	69
5.2 ข้อเสนอแนะ	70
รายการอ้างอิง	72
ภาคผนวก	74
ภาคผนวก ก	75
ภาคผนวก ข	80
ภาคผนวก ค	90

ภาคผนวก ง	93
ประวัติผู้เขียนวิทยานิพนธ์	149



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 3.1 จำนวนเกมที่ใช้การเรียนรู้เพื่อหาค่าน้ำหนักของยูนิต (w_u) และค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) ในแต่ละแผนทดสอบ.....	27
---	----



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1.1 World Cyber Games 2008 [1].....	1
รูปที่ 2.1 DUNE 2 [7].....	6
รูปที่ 2.2 Spring [8].....	7
รูปที่ 2.3 สภาพแวดล้อมข้อมูลสมบูรณ์และไม่สมบูรณ์ [9].....	8
รูปที่ 2.4 การทำงานของไดนามิกสคริปต์ [11].....	12
รูปที่ 2.5 การสร้างชุดสคริปต์ [12].....	13
รูปที่ 2.6 การปรับค่าน้ำหนัก [12].....	15
รูปที่ 2.7 การลงโทษค่าฟิตเนสที่สูง [12].....	16
รูปที่ 2.8 Weight Clipping และ Top Culling [13].....	17
รูปที่ 2.9: ค่าฟังก์ชันฟิตเนส 0.942 ถูกเปลี่ยนเป็นความน่าจะเป็นในการทำนาย 0.72 [9].....	20
รูปที่ 2.10 จำนวนเกมที่เก็บข้อมูลจากการทดลองของ Bakkes [14].....	22
รูปที่ 3.1 การใช้การเรียนรู้แบบ Temporal Difference หาค่าน้ำหนักของยูนิท (w_{ij}).....	26
รูปที่ 3.2 การหาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม.....	29
รูปที่ 3.3 กราฟของค่าน้ำหนักที่เปลี่ยนแปลง [13].....	32
รูปที่ 3.4 การทำงานของไดนามิกสคริปต์และการปรับความยากเข้ากับผู้เล่นโดยวิธีการตัดค่าน้ำหนักที่สูง.....	33
รูปที่ 4.1 แผนที่ 1 (SmallDivide).....	39
รูปที่ 4.2 แผนที่ 2 (Barren).....	39
รูปที่ 4.3 แผนที่ 3 (Tomb Stone Desert V2).....	40
รูปที่ 4.4 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide).....	41
รูปที่ 4.5 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren).....	41
รูปที่ 4.6 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2).....	42
รูปที่ 4.7 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide).....	42
รูปที่ 4.8 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren).....	42
รูปที่ 4.9 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	42
รูปที่ 4.10 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide).....	43
รูปที่ 4.11 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren).....	43

รูปที่ 4.12 ผลการทดลองคู่อัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	43
รูปที่ 4.13 ผลการทดลองคู่อัญญาประดิษฐ์ทดสอบ - คู่อัญญาประดิษฐ์ทดสอบ ในแผนที่ 1 (SmallDivide).....	43
รูปที่ 4.14 ผลการทดลองคู่อัญญาประดิษฐ์ทดสอบ - คู่อัญญาประดิษฐ์ทดสอบ ในแผนที่ 2 (Barren).....	44
รูปที่ 4.15 ผลการทดลองคู่อัญญาประดิษฐ์ทดสอบ - คู่อัญญาประดิษฐ์ทดสอบ ในแผนที่ 3 (Tomb Stone Desert V2).....	44
รูปที่ 4.16 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ E323 - E323 ในแผนที่ 1 (SmallDivide).....	57
รูปที่ 4.17 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide).....	57
รูปที่ 4.18 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ AAI - AAI ในแผนที่ 1 (SmallDivide).....	58
รูปที่ 4.19 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide).....	58
รูปที่ 4.20 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ RAI - RAI ในแผนที่ 1 (SmallDivide).....	59
รูปที่ 4.21 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide).....	59
รูปที่ 4.22 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ E323 - E323 ในแผนที่ 2 (Barren).....	60
รูปที่ 4.23 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren).....	60
รูปที่ 4.24 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ AAI - AAI ในแผนที่ 2 (Barren).....	61
รูปที่ 4.25 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren).....	61
รูปที่ 4.26 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ RAI - RAI ในแผนที่ 2 (Barren).....	62
รูปที่ 4.27 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren).....	62
รูปที่ 4.28 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2).....	63
รูปที่ 4.29 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2).....	63
รูปที่ 4.30 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	64
รูปที่ 4.31 ตัวอย่างกราฟฟังก์ชันวัตถุประสงค์คะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	64

รูปที่ 4.32 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	65
รูปที่ 4.33 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	65
รูปที่ 4.34 ทรัพยากรในแผนที่ 1 (SmallDivide).....	66
รูปที่ 4.35 ทรัพยากรในแผนที่ 2 (Barren).....	67
รูปที่ 4.36 ทรัพยากรในแผนที่ 3 (Tomb Stone Desert V2).....	68
รูปที่ ก.1 ผลงานการทำงานของไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง.....	76
รูปที่ ก.2 ผลงานการทำงานของการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule).....	77
รูปที่ ก.3 ผลงานการทำงานของการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัดคะแนนซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัดคะแนน ณ เวลาก่อนหน้า (Change w_{inc} , w_{dec}).....	77
รูปที่ ก.4 ผลงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุกๆ ช่วงเวลาหนึ่ง และวิธีการตัดค่าน้ำหนักที่สูงในการทดลองจะเลือกเพิ่มกฎในชุดสคริปต์ที่ละกฎทุกๆ 7 นาที (7-minute increment) ..	78
รูปที่ ก.5 ผลงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules).....	78
รูปที่ ก.6 ผลงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัดคะแนนซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัดคะแนน ณ เวลาก่อนหน้า (Mix).....	79
รูปที่ ข.1 กฎย่อยในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น	83
รูปที่ ข.2 กฎย่อยในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) และค่าน้ำหนักเริ่มต้น	84
รูปที่ ข.3 กฎย่อยในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น.....	85
รูปที่ ข.4 กฎย่อยในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น	86
รูปที่ ข.5 กฎย่อยในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 2 (Barren) และค่าน้ำหนักเริ่มต้น	86

รูปที่ ข.6 กฎย่อยในกฎพื้นฐานของคู่มือปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น.....	86
รูปที่ ข.7 กฎย่อยในกฎพื้นฐานของคู่มือปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น	87
รูปที่ ข.8 กฎย่อยในกฎพื้นฐานของคู่มือปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 2 (Barren) และค่าน้ำหนักเริ่มต้น	88
รูปที่ ข.9 กฎย่อยในกฎพื้นฐานของคู่มือปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น.....	89
รูปที่ ค.1 Commander	90
รูปที่ ค.2 การเก็บเกี่ยวทรัพยากรในเกม	91
รูปที่ ค.3 การผลิตสิ่งก่อสร้างระดับถัดไป.....	92
รูปที่ ค.4 การต่อสู้กันระหว่างสองฝ่าย [8].....	92
รูปที่ ง.1 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule.....	93
รูปที่ ง.2 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}	94
รูปที่ ง.3 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	94
รูปที่ ง.4 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	95
รูปที่ ง.5 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	95
รูปที่ ง.6 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule	96
รูปที่ ง.7 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}	96
รูปที่ ง.8 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment	97
รูปที่ ง.9 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	97
รูปที่ ง.10 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Mix.....	98
รูปที่ ง.11 อีสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule.....	98

รูปที่ ง.12	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}	99
รูปที่ ง.13	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	99
รูปที่ ง.14	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules.....	100
รูปที่ ง.15	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix.....	100
รูปที่ ง.16	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule.....	101
รูปที่ ง.17	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}	101
รูปที่ ง.18	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	102
รูปที่ ง.19	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	102
รูปที่ ง.20	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	103
รูปที่ ง.21	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule.....	103
รูปที่ ง.22	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}	104
รูปที่ ง.23	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment.....	104
รูปที่ ง.24	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	105
รูปที่ ง.25	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Mix.....	105
รูปที่ ง.26	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule.....	106
รูปที่ ง.27	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}	106
รูปที่ ง.28	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	107

รูปที่ ง.29	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules.....	107
รูปที่ ง.30	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix.....	108
รูปที่ ง.31	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule.....	108
รูปที่ ง.32	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}	109
รูปที่ ง.33	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	109
รูปที่ ง.34	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	110
รูปที่ ง.35	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	110
รูปที่ ง.36	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule.....	111
รูปที่ ง.37	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}	111
รูปที่ ง.38	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment.....	112
รูปที่ ง.39	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	112
รูปที่ ง.40	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Mix.....	113
รูปที่ ง.41	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule.....	113
รูปที่ ง.42	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}	114
รูปที่ ง.43	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	114
รูปที่ ง.44	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules.....	115
รูปที่ ง.45	ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix.....	115

รูปที่ ง.46	ฮิสโทแกรมของ E323 - E323 ในแผนที่ 1 (SmallDivide).....	116
รูปที่ ง.47	ฮิสโทแกรมของ E323 - AAI ในแผนที่ 1 (SmallDivide)	116
รูปที่ ง.48	ฮิสโทแกรมของ E323 - RAI ในแผนที่ 1 (SmallDivide)	117
รูปที่ ง.49	ฮิสโทแกรมของ AAI - AAI ในแผนที่ 1 (SmallDivide).....	117
รูปที่ ง.50	ฮิสโทแกรมของ AAI - RAI ในแผนที่ 1 (SmallDivide).....	118
รูปที่ ง.51	ฮิสโทแกรมของ RAI - RAI ในแผนที่ 1 (SmallDivide).....	118
รูปที่ ง.52	ฮิสโทแกรมของ RAI - AAI ในแผนที่ 1 (SmallDivide).....	119
รูปที่ ง.53	ฮิสโทแกรมของ E323 - E323 ในแผนที่ 2 (Barren).....	119
รูปที่ ง.54	ฮิสโทแกรมของ E323 - AAI ในแผนที่ 2 (Barren).....	120
รูปที่ ง.55	ฮิสโทแกรมของ E323 - RAI ในแผนที่ 2 (Barren)	120
รูปที่ ง.56	ฮิสโทแกรมของ AAI - AAI ในแผนที่ 2 (Barren)	121
รูปที่ ง.57	ฮิสโทแกรมของ AAI - RAI ในแผนที่ 2 (Barren)	121
รูปที่ ง.58	ฮิสโทแกรมของ RAI - RAI ในแผนที่ 2 (Barren).....	122
รูปที่ ง.59	ฮิสโทแกรมของ RAI - AAI ในแผนที่ 2 (Barren).....	122
รูปที่ ง.60	ฮิสโทแกรมของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2)	123
รูปที่ ง.61	ฮิสโทแกรมของ E323 - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	123
รูปที่ ง.62	ฮิสโทแกรมของ E323 - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	124
รูปที่ ง.63	ฮิสโทแกรมของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)	124
รูปที่ ง.64	ฮิสโทแกรมของ AAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)	125
รูปที่ ง.65	ฮิสโทแกรมของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)	125
รูปที่ ง.66	ฮิสโทแกรมของ RAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)	126
รูปที่ ง.67	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule	127
รูปที่ ง.68	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} w_{dec}	127
รูปที่ ง.69	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	127
รูปที่ ง.70	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	128
รูปที่ ง.71	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	128
รูปที่ ง.72	กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule	128

รูปที่ ง.73 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} w_{dec}	129
รูปที่ ง.74 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment	129
รูปที่ ง.75 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	129
รูปที่ ง.76 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Mix.....	130
รูปที่ ง.77 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule	130
รูปที่ ง.78 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}	130
รูปที่ ง.79 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	131
รูปที่ ง.80 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules	131
รูปที่ ง.81 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix	131
รูปที่ ง.82 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule.....	132
รูปที่ ง.83 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} w_{dec}	132
รูปที่ ง.84 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	132
รูปที่ ง.85 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	133
รูปที่ ง.86 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	133
รูปที่ ง.87 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule	133
รูปที่ ง.88 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} w_{dec}	134
รูปที่ ง.89 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment	134

รูปที่ ง.90 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	134
รูปที่ ง.91 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Mix.....	135
รูปที่ ง.92 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule	135
รูปที่ ง.93 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} w_{dec}	135
รูปที่ ง.94 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	136
รูปที่ ง.95 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules	136
รูปที่ ง.96 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix	136
รูปที่ ง.97 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule.....	137
รูปที่ ง.98 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} w_{dec}	137
รูปที่ ง.99 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment.....	137
รูปที่ ง.100 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules.....	138
รูปที่ ง.101 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix.....	138
รูปที่ ง.102 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule	138
รูปที่ ง.103 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} w_{dec}	139
รูปที่ ง.104 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment	139
รูปที่ ง.105 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules.....	139
รูปที่ ง.106 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Mix.....	140

รูปที่ ง.107 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule	140
รูปที่ ง.108 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}	140
รูปที่ ง.109 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment.....	141
รูปที่ ง.110 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules	141
รูปที่ ง.111 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix	141
รูปที่ ง.112 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - E323 ในแผนที่ 1 (SmallDivide)	142
รูปที่ ง.113 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - AAI ในแผนที่ 1 (SmallDivide).....	142
รูปที่ ง.114 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - RAI ในแผนที่ 1 (SmallDivide)	142
รูปที่ ง.115 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - AAI ในแผนที่ 1 (SmallDivide)	143
รูปที่ ง.116 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - RAI ในแผนที่ 1 (SmallDivide)	143
รูปที่ ง.117 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - RAI ในแผนที่ 1 (SmallDivide).....	143
รูปที่ ง.118 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - AAI ในแผนที่ 1 (SmallDivide)	144
รูปที่ ง.119 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - E323 ในแผนที่ 2 (Barren)	144
รูปที่ ง.120 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - AAI ในแผนที่ 2 (Barren).....	144
รูปที่ ง.121 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - RAI ในแผนที่ 2 (Barren).....	145
รูปที่ ง.122 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - AAI ในแผนที่ 2 (Barren).....	145
รูปที่ ง.123 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - RAI ในแผนที่ 2 (Barren)	145
รูปที่ ง.124 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - RAI ในแผนที่ 2 (Barren).....	146
รูปที่ ง.125 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - AAI ในแผนที่ 2 (Barren)	146
รูปที่ ง.126 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2).....	146
รูปที่ ง.127 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	147
รูปที่ ง.128 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	147
รูปที่ ง.129 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	147
รูปที่ ง.130 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	148

รูปที่ ง.131 กราฟผล ชนะ/แพ้ ทำเกมของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2).....	148
รูปที่ ง.132 กราฟผล ชนะ/แพ้ ทำเกมของ RAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2).....	148



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันเกมวางแผนการรบแบบตอบสนองทันที (Real-time Strategy Games) นั้นเป็นที่นิยมอย่างมากในตลาดเกม มีการจัดแข่งขันในการแข่งขันเกมชั้นนำต่างๆ มากมาย เช่น World Cyber Games 2008 ที่ผ่านมาที่เมืองโคโลญ์ ประเทศเยอรมนี ได้ใช้เกม StarCraft: Brood War, WarCraft III: The Frozen Throne, Command & Conquer 3: Kane Wrath, Age of Empires III: The Asian Dynasties ดังรูปที่ 1 ในการแข่งขันจากจำนวนเกมคอมพิวเตอร์ที่ใช้ในการแข่งขันทั้งหมด 10 เกม [1] แสดงให้เห็นถึงความนิยมและการเติบโตอย่างรวดเร็วของวงการเกมวางแผนการรบแบบตอบสนองทันที



รูปที่ 1.1 World Cyber Games 2008 [1]

ระบบปัญญาประดิษฐ์ เป็นส่วนประกอบที่สำคัญมากในเกมวางแผนการรบแบบตอบสนองทันที เกมวางแผนการรบแบบตอบสนองทันทีนั้นไม่สามารถเน้นที่กราฟิกอย่างเดียวได้จะต้องมีระบบปัญญาประดิษฐ์ที่ดีด้วย [2] เนื่องจากเป็นส่วนประกอบที่ทำให้เกมมี

ความจริง ทำท่าย และสนุกมากยิ่งขึ้น ซึ่งปัญญาประดิษฐ์ในเกมวางแผนการรบแบบตอบสนองทันกาลนั้น มีการพัฒนาอย่างต่อเนื่อง Marc Ponsen และคณะ [3,4] ได้สร้างปัญญาประดิษฐ์สำหรับเกมวางแผนการรบแบบตอบสนองทันกาลโดยใช้เทคนิคการปรับตัวแบบไดนามิกสคริปต์ ในเกมประเภทเปิดเผยซอร์สโค้ดชื่อ Wargus โดยแบ่งเกมออกเป็นสถานะต่างๆ ตามชุดของสิ่งก่อสร้างในเกม แต่ละสถานะของเกมมีกฎพื้นฐานของตัวเอง และได้ปรับปรุงกฎพื้นฐานของไดนามิกสคริปต์ให้ดีขึ้นโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)

งานวิจัยเกี่ยวกับปัญญาประดิษฐ์ในเกมวางแผนการรบแบบตอบสนองทันกาลในปัจจุบันนั้นเป็นการพัฒนาให้ปัญญาประดิษฐ์เอาชนะผู้เล่นได้ แต่ผู้เล่นเกมแต่ละคนมีความสามารถและพื้นฐานการเล่นแตกต่างกัน การออกแบบให้ปัญญาประดิษฐ์สามารถเอาชนะผู้เล่น อาจทำให้ผู้เล่นที่อ่อนกว่าปัญญาประดิษฐ์เกิดความรู้สึกกังวลว่ากำลังเล่นเกมที่ยากเกินความสามารถ และผู้เล่นที่เก่งกว่าปัญญาประดิษฐ์จะเกิดความรู้สึกเบื่อที่เกมง่ายจนเกินไปสำหรับเขา [5, 6]

งานวิจัยชิ้นนี้จึงนำเสนอการสร้างปัญญาประดิษฐ์สำหรับเกมวางแผนการรบแบบตอบสนองทันกาลที่มีความสามารถสู้กับผู้เล่นโดยนำเทคนิคการปรับตัวแบบไดนามิกสคริปต์มาปรับปรุงให้สามารถปรับตัวให้เข้ากับผู้เล่นขณะที่ผู้เล่นกำลังเล่นเกมอยู่ โดยใช้วิธีการตัดค่าน้ำหนักที่สูง และใช้การผลิตรายการมาเป็นกฎพื้นฐาน นอกจากนี้ยังมีการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้กฎพื้นฐานที่เป็นการผลิตรายการด้วยเทคนิคเสริมต่างๆ

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อทดสอบว่าไดนามิกสคริปต์สามารถนำมาใช้ในการปรับความยากสำหรับเกมวางแผนการรบแบบตอบสนองทันกาลได้ดีหรือไม่เพียงใด
2. เพื่อนำอัตราการผลิตรายการมาใช้ในไดนามิกสคริปต์สำหรับการสร้างปัญญาประดิษฐ์สำหรับเกมวางแผนการรบแบบตอบสนองทันกาลที่มีความสามารถสู้กับผู้เล่น
3. การปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้กฎพื้นฐานที่เป็นการผลิตรายการให้สามารถปรับระดับความยากให้สู้กับผู้เล่นได้ดียิ่งขึ้น

1.3 ขอบเขตของการวิจัย

1. การปรับระดับความยากให้สุสึกับผู้เล่นนั้นจะเรียนรู้และปรับความยากเฉพาะในเกมที่มีผู้เล่น 2 ฝ่าย คือ ฝ่ายปัญญาประดิษฐ์กับฝ่ายศัตรู และปรับความยากในเกมครั้งที่เล่นเท่านั้น
2. ค่าน้ำหนักของยูนิต (w_u) และค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) จะเก็บข้อมูลจากการให้ปัญญาประดิษฐ์พื้นฐานของเกมสู้กันในพื้นที่ของเกมส์ปริงในชุดแผนที่ทดสอบจำนวน 3 แผนที่ คือ SmallDivide, Barren และ Tomb Stone Desert V2
3. ทำการทดสอบในพื้นที่ของเกมส์ปริงในชุดแผนที่ทดสอบจำนวน 3 แผนที่ คือ SmallDivide, Barren และ Tomb Stone Desert V2
4. การเปรียบเทียบว่าปัญญาประดิษฐ์ใดจะสุสึดีกว่ากันจะเปรียบเทียบโดยใช้เวลาเฉลี่ยในการเล่น และผลการ ชนะ/แพ้ ท้ายเกม

1.4 ประโยชน์ที่ได้รับ

1. ได้ตัวอย่างปัญญาประดิษฐ์ในเกมวางแผนการรบแบบตอบสนองทันกาลที่มีความสุสึกับผู้เล่น โดยสามารถนำแนวทางการปรับอัตราการผลิต ไปใช้กับเกมวางแผนการรบแบบตอบสนองทันกาลอื่นได้ ส่วนฟังก์ชันสำหรับวัดความสุสึนั้นแต่ละเกมจะต้องมีเป็นของตัวเอง แต่สามารถหาฟังก์ชันได้ด้วยหลักการที่คล้ายกับในงานวิทยานิพนธ์นี้
2. เกิดองค์ความรู้ใหม่ในสาขาปัญญาประดิษฐ์

1.5 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีพื้นฐาน และงานวิจัยที่เกี่ยวข้อง
2. ศึกษาเกมทดสอบและปัญญาประดิษฐ์พื้นฐานของเกมทดสอบ
3. สร้างโมดูลเพื่อควบคุมการผลิตยูนิตในปัญญาประดิษฐ์ E323
4. สร้างโมดูลสำหรับเก็บข้อมูลของเกม
5. หาค่าน้ำหนักของยูนิต (w_u) โดยใช้การเรียนรู้แบบ Temporal Difference จากแผนที่ทดสอบ
6. หาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) จากแผนที่ทดสอบ
7. สร้างปัญญาประดิษฐ์โดยใช้ไดนามิกสคริปต์
8. ปรับระดับความยากของเกมให้เข้ากับผู้เล่นโดยใช้วิธีการตัดค่าน้ำหนักที่สูง (Top Culling)
9. เก็บค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานในแต่ละปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบ

10. ทดสอบขั้นแรกว่าการผลิตยูนิตใช้ในการปรับความยากให้สูสีกับระดับของผู้เล่นได้หรือไม่เพียงใด
11. ปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากร
12. ทดสอบขั้นที่สองว่าการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรสามารถปรับระดับความยากให้สูสีกับผู้เล่นได้ดีกว่าขั้นตอนแรกหรือไม่เพียงใด
13. วิเคราะห์ผลการทดลอง
14. สรุปผลการวิจัยและเรียบเรียงวิทยานิพนธ์

1.6 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการ ดังนี้

- “Difficulty Balancing In Real-Time Strategy Gaming Session Using Resource Production Adjustment” โดย ปิยพจน์ เกษมภักดีพงษ์ และวิชญ์ โศตรจรัส ในงานประชุมวิชาการ “GAMEON-ASIA’2011” ซึ่งจัดขึ้น ณ ประเทศสิงคโปร์ ระหว่างวันที่ 1 มีนาคม ถึง 3 เมษายน 2554

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับบทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะมีการนำเสนอทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับปัญญาประดิษฐ์ รูปแบบของเกววางแผนการรบแบบตอบสนองทันที และหลักการที่จะใช้ในการสร้างปัญญาประดิษฐ์ที่มีความสามารถสู้กับผู้เล่น ซึ่งทั้งหมดนี้เป็นพื้นฐานในการวิจัยและพัฒนาระบบปัญญาประดิษฐ์โดยทั่วไป ตามด้วยงานวิจัยที่เกี่ยวข้องกับปัญญาประดิษฐ์ที่สามารถปรับความยากให้เข้ากับผู้เล่น และฟังก์ชันวัตถุประสงค์ที่ใช้วัดผลจากการทำงานของปัญญาประดิษฐ์

2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับหัวข้อทฤษฎีที่เกี่ยวข้องนี้ จะเริ่มต้นนำเสนอจาก ความเป็นมาของเกววางแผนการรบแบบตอบสนองทันที ความเป็นมาของเกวสปริง ลักษณะของสภาพแวดล้อม ข้อมูลสมบูรณ์และสภาพแวดล้อมข้อมูลไม่สมบูรณ์ ความหมายของปัญญาประดิษฐ์ ความหมายของฟังก์ชันวัตถุประสงค์ ความหมายของฟังก์ชันวัดความเหมาะสม หลักการของการเคลื่อนไหวตามความชัน และสุดท้ายจะเป็นส่วนของ Temporal Difference Learning

2.1.1 เกมวางแผนการรบแบบตอบสนองทันที (Real-time Strategy Games)

เกววางแผนการรบแบบตอบสนองทันทีเป็นเกมประเภทหนึ่งที่แยกออกมาจากเกมสงครามกลยุทธ์ (Strategy) โดยเกววางแผนการรบแบบตอบสนองทันทีเกมแรกคือ DUNE 2 ในปี 1992 โดย Brett Sperry จากบริษัท Westwood Studio ในครั้งแรก Sperry ได้ตั้งชื่อว่า Real-time Strategy Games เนื่องจากต้องการให้ผู้เล่นเข้าใจว่าเกม DUNE 2 นั้นมีความซับซ้อนน้อยกว่าเกมสงครามกลยุทธ์ (Strategy Games) เกมวางแผนการรบแบบตอบสนองทันที พบมากในเครื่องคอมพิวเตอร์เนื่องจากคีย์บอร์ดและเมาส์นั้นมีความเหมาะสมต่อการควบคุมเกม และมักจะเล่นร่วมกันได้หลายคนผ่านทางอินเทอร์เน็ตหรือผ่านระบบแลนอีกด้วย เนื้อเรื่องในเกมมีได้หลากหลายรูปแบบ แล้วแต่เกมนั้นๆ จะกำหนด ตั้งแต่เวทมนตร์ คาถา พ่อมด กองทหารยุคกลาง ไปจนถึงสงครามระหว่างดวงดาว ในเกววางแผนการรบแบบตอบสนองทันทีผู้เล่นทุกฝ่ายจะต้องแข่งกับเวลา เนื่องจากไม่มีการหยุดพักระหว่างรบ เวลาในเกมจะดำเนินไปตลอด รูปแบบการเล่นจะเน้นที่การสร้างและควบคุมกองทัพซึ่งประกอบไปด้วยยูนิตต่างๆ เข้าทำการสู้รบกัน รวมไปถึงการเก็บเกี่ยว บริหารทรัพยากร และการวางแผนกลยุทธ์ โดยมีเป้าหมายเพื่อทำลายฝ่ายตรงข้าม เกมวางแผนการรบแบบตอบสนองทันทีเป็นที่นิยมในหมู่นักเล่นเกมทั่วไป ตัวอย่างของเกมประเภทนี้ที่มีชื่อเสียง เช่น Command & Conquer, StarCraft และ Warcraft เป็นต้น รูปที่ 2.1 แสดงตัวอย่างหน้าจอของเกม DUNE 2



รูปที่ 2.1 DUNE 2 [7]

2.1.2 สปริง (Spring)

สปริง [8] คือ เกมวางแผนการรบแบบตบสนองทันกาลที่สร้างขึ้นโดย กลุ่มผู้พัฒนาชาวสวีเดน และกลุ่มผู้พัฒนาอิสระ (Spring community) โดยเกิดจากแนวคิดที่จะนำเกม Total Annihilation มาทำใหม่ให้อยู่ในรูปแบบสามมิติเต็มรูปแบบ (ในเกมเดิมจะบังคับมุมมองกล้องของเกมไว้) เป็นเกมประเภทเปิดเผยซอร์สโค้ด (Open Source) ให้เหล่านักพัฒนาได้สร้างตัวเสริม (Mod) ของเกม ปัญญาประดิษฐ์ รวมถึงร่วมกันพัฒนาเกมได้ เกมสปริงนี้ผู้เล่นจะได้รับความคุมยูนิตหลากหลายชนิด บริหารการใช้ทรัพยากร วางแผนกลยุทธ์ สร้างสิ่งก่อสร้างในระดับที่สร้างยากหรือง่ายต่าง ๆ กันเพื่อให้สิ่งก่อสร้างช่วยผลิตยูนิตออกไปสู้รบกับฝ่ายตรงข้าม โดยมีเป้าหมายคือจะต้องทำลายยูนิตคอมมานเดอร์ (Commander) ของฝ่ายตรงข้ามให้ได้ จึงจะเป็นผู้ชนะในเกม รูปที่ 2.2 แสดงตัวอย่างหน้าจอของเกมสปริง รายละเอียดวิธีการเล่นนั้นอยู่ในภาคผนวก ค

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



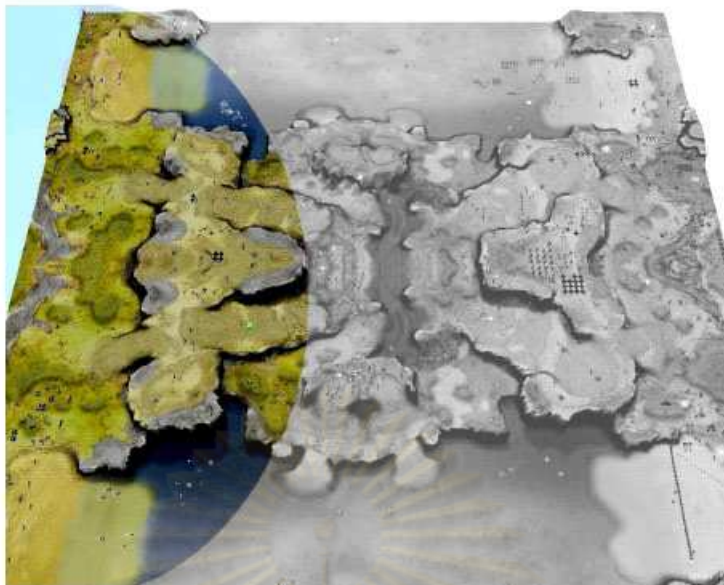
รูปที่ 2.2 Spring [8]

2.1.3 สภาพแวดล้อมข้อมูลสมบูรณ์ (Perfect Information Environment) และ สภาพแวดล้อมข้อมูลไม่สมบูรณ์ (Imperfect Information Environment)

สภาพแวดล้อมข้อมูลสมบูรณ์ หมายถึง การที่ปัญญาประดิษฐ์ของเกมสามารถเข้าถึงข้อมูลของเกมได้ทั้งหมดโดยไม่ขึ้นกับระยะการมองเห็นของยูนิต ซึ่งจะตรงข้ามกับสภาพแวดล้อมข้อมูลไม่สมบูรณ์ ที่ปัญญาประดิษฐ์ของเกมถูกจำกัดความสามารถในการเข้าถึงข้อมูลของเกมได้บางส่วนตามระยะการมองเห็นของยูนิต (Line of Sight)

จากรูปที่ 2.3 ในสภาพสภาพแวดล้อมข้อมูลไม่สมบูรณ์ ปัญญาประดิษฐ์จะมองเห็นเข้าถึงข้อมูลเฉพาะส่วนที่แทนด้วยการแรเงาไว้เท่านั้น แต่ในสภาพแวดล้อมข้อมูลสมบูรณ์จะเข้าถึงข้อมูลได้ทั้งส่วนที่แทนด้วยการแรเงาและไม่แรเงา

ศูนย์วิจัยเทคโนโลยีการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.3 สภาพแวดล้อมข้อมูลสมบูรณ์และไม่สมบูรณ์ [9]

2.1.4 ปัญญาประดิษฐ์ (Artificial Intelligence)

สำหรับความหมายของปัญญาประดิษฐ์ในเกมนั้น หมายถึง ฟังก์ชันที่ทำหน้าที่ควบคุมการทำงานของตัวละครหรือสภาพแวดล้อมภายในเกม โดยไม่คำนึงถึงวิธีการทำงานหรือวิธีการสร้าง ซึ่งจะเห็นว่าความหมายแตกต่างไปจากปัญญาประดิษฐ์ในวงการวิชาการ ปัญญาประดิษฐ์ในเกมไม่จำเป็นจะต้องเป็นระบบที่มีความสามารถในการเรียนรู้อย่างชาญฉลาด ประสาทเทียม (Neural Network) หรือขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) แต่อาจเป็นระบบที่ตอบโต้ตามที่ถูกกำหนดมาโดยไม่จำเป็นต้องมีการเรียนรู้แต่อย่างใด เพียงแต่ทำให้ผู้เล่นเกิดความพึงพอใจจากการเล่นหรือเกิดความรู้สึกท้าทายเท่านั้น

2.1.5 ฟังก์ชันวัดคะแนน (Score Function) หรือฟังก์ชันประเมินคุณค่า (Evaluation Function)

หมายถึงฟังก์ชันที่ใช้วัดผลจากการทำงานของปัญญาประดิษฐ์ ประโยชน์ของฟังก์ชันวัดคะแนน มีสองอย่างดังนี้

- วัดผลโดยรวม คือ ขณะนั้นปัญญาประดิษฐ์หรือผู้เล่นเก่งกว่ากันมากน้อยเพียงใด
- วัดผลการทำงานของชุดคำสั่งที่ปัญญาประดิษฐ์สร้างขึ้นในครั้งนั้นว่ามีประสิทธิภาพเพียงใด เพื่อนำไปเป็นข้อมูลป้อนกลับให้ปัญญาประดิษฐ์สามารถตัดสินใจเลือกชุดคำสั่งที่มีประสิทธิภาพมากที่สุดต่อไป

เนื่องจากความซับซ้อนของเกมในปัจจุบัน การออกแบบฟังก์ชันวัตถุประสงค์ที่เหมาะสมกับเกมนั้นเป็นงานที่ยาก เพราะถ้าออกแบบแล้วฟังก์ชันวัตถุประสงค์ไม่ถูกต้องหรือไม่เหมาะสมจะทำให้ขั้นตอนการเรียนรู้ของปัญญาประดิษฐ์ได้รับผลกระทบไปด้วย

2.1.6 ฟังก์ชันวัดความเหมาะสม (Fitness Function)

เป็นฟังก์ชันที่ใช้วัดผลการทำงานของชุดคำสั่งที่ปัญญาประดิษฐ์สร้างขึ้นในครั้งนั้นว่ามีประสิทธิภาพเพียงใด เพื่อนำไปเป็นข้อมูลป้อนกลับให้ปัญญาประดิษฐ์สามารถตัดสินใจเลือกชุดคำสั่งที่มีประสิทธิภาพมากที่สุด คล้ายกับฟังก์ชันวัตถุประสงค์ แตกต่างกันว่า ฟังก์ชันวัดความเหมาะสมอาจประกอบด้วยฟังก์ชันวัตถุประสงค์หลายตัว และโดยทั่วไปนิยมกำหนดขอบเขตของผลลัพธ์ให้มีค่าตั้งแต่ 0 ถึง 1

2.1.7 การเคลื่อนลงตามความชัน (Gradient Descent)

เป็นขั้นตอนวิธีการหาค่าต่ำสุดสัมพัทธ์ โดยการเคลื่อนลงตามความชัน มีวิธีการดังที่จะอธิบายต่อไปนี้

สมมติฟังก์ชัน $y = f(x)$ มีค่าต่ำสุดสัมพัทธ์อยู่ที่จุด x_{min} เราสามารถหาค่าต่ำสุดสัมพัทธ์ของฟังก์ชันได้โดยอาศัยขั้นตอนดังต่อไปนี้

1. สุ่มเลือกค่า x ขึ้นมา ให้เป็น x_1
2. หาค่า $f'(x_1)$
 - ถ้าได้ค่าเป็นลบแสดงว่า $x_{min} > x_1$
 - ถ้าได้ค่าเป็นบวกแสดงว่า $x_{min} < x_1$
3. ทำการปรับค่า x_1 ตามสมการที่ (2.1)

$$x_1 \rightarrow x_1 - \eta f'(x_1) \quad (2.1)$$

โดย η คือค่าน้ำหนักในการปรับค่าแต่ละครั้ง

ถ้าฟังก์ชัน y สามารถหาค่าต่ำสุดสัมพัทธ์ได้ สมการที่ (2.1) จะปรับค่าจนได้ค่าต่ำสุดสัมพัทธ์ของฟังก์ชัน ($f'(x_1) = 0$)

2.1.8 Temporal Difference Learning

การเรียนรู้แบบ Temporal Difference [10] เป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) ชนิดหนึ่งสำหรับเรียนรู้ที่จะทำนายผลลัพธ์ ซึ่งการเรียนรู้แบบมีผู้สอนปกตินี้จะใช้ผลต่างระหว่างค่าที่ทำนายได้และผลลัพธ์ แต่การเรียนรู้แบบ Temporal Difference จะใช้ผลต่างของ Temporally successive prediction แทน โดยค่าน้ำหนักที่เปลี่ยนแปลงไปจะเป็นไปดังสมการที่ (2.2)

$$\Delta W_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^t \lambda^{t-k} \nabla_w P_k \quad (2.2)$$

โดย

ΔW_t คือ ค่าน้ำหนักที่เปลี่ยนแปลงไปที่เวลา t

P คือ เซตของ Temporally successive prediction

α คือ อัตราการเรียนรู้ (Learning rate)

λ คือ ตัวแปรควบคุมน้ำหนักของ $\nabla_w P_k$ ที่เกิดขึ้น k ชั้นก่อนหน้า

$\nabla_w P_k$ คือ เกรเดียนของ P_k เทียบกับ w

ค่า α จะเป็นตัวกำหนดการปรับค่าน้ำหนักในแต่ละครั้งว่าปรับค่าน้ำหนักมากหรือน้อย

พจน์ λ^{t-k} จะเป็นตัวให้น้ำหนักของ $\nabla_w P_k$ โดยจะให้น้ำหนักของ $\nabla_w P_k$ ที่อยู่ใกล้กับ t หน้านั้นมากกว่า $\nabla_w P_k$ ที่อยู่ไกลกว่า t หน้านั้น

2.2 งานวิจัยที่เกี่ยวข้อง

ในส่วนงานวิจัยที่เกี่ยวข้องนี้จะแบ่งเป็น 2 ส่วน เริ่มจาก ไดนามิกสคริปต์และวิธีการปรับความยากให้เข้ากับผู้เล่น ซึ่งจะอธิบายหลักการและวิธีการทำงานของไดนามิกสคริปต์ว่าสามารถปรับตัวให้เอาชนะผู้เล่นได้อย่างไร ตามด้วยเทคนิคเสริมของไดนามิกสคริปต์ที่ทำหน้าที่ปรับความยากให้เข้ากับผู้เล่นทั้งสามแบบ คือ การลงโทษค่าฟิตเนสที่สูง การตัดค่าน้ำหนัก และการตัดค่าน้ำหนักที่สูง อีกส่วนจะเกี่ยวกับฟังก์ชันวัตถุประสงค์และผลจากการทำงานของปัญญาประดิษฐ์ในเกมสปริง

2.2.1 ไดนามิกสคริปต์ (Dynamic Script) และวิธีการปรับความยากให้เข้ากับผู้เล่นแต่ละคน

Pieter Spronck [11] ได้เสนอเทคนิคทางปัญญาประดิษฐ์ของเกมรูปแบบหนึ่ง ที่เรียกว่าไดนามิกสคริปต์ (Dynamic Script) ซึ่งได้แรงบันดาลใจมาจากวิธีการเรียนรู้แบบรีนฟอร์ซเมนต์ (Reinforcement Learning) โดย Spronck นำเสนอว่าคุณภาพของเกมนั้นขึ้นอยู่กับคุณค่าทางด้านความบันเทิงของเกม เกมโดยปกตินั้นจะมีลักษณะของปัญญาประดิษฐ์ที่อาจจะไม่เป็นที่พอใจของผู้เล่น ซึ่งปัญญาประดิษฐ์ในเกมมักจะฉลาดเพียงพอที่จะสู้กับผู้เล่นได้ ผู้เล่นจึงต้องการเล่นกับผู้เล่นที่เป็นคนจริงๆมากกว่า การปรับปรุงปัญญาประดิษฐ์ให้มีความฉลาด และสามารถสู้กับผู้เล่นได้เก่งขึ้นจึงเป็นสิ่งสำคัญเพื่อใช้ในกรณีที่ไม่สามารถหาผู้เล่นที่เป็นคนจริงๆ ได้ ซึ่งจะทำให้เกมมีความท้าทายมากขึ้นตามไปด้วย โดย Spronck ศึกษาเกมสวมบทบาท (Computer Role Playing Games(CRPGs)) ซึ่งเกมประเภทนี้ในแต่ละครั้งของการเล่นนั้นมีความเป็นไปได้ในการทำสิ่งต่างๆ ได้หลากหลาย ทำให้ปกติแล้วผู้พัฒนาเกมส่วนใหญ่เลือกที่จะใช้สคริปต์(Script) เป็นตัวกำหนดคำสั่งปัญญาประดิษฐ์ของเกมแทน เนื่องจากสามารถที่จะควบคุมได้ง่ายกว่า และได้ผลที่มีความแน่นอนมากกว่า ไม่เกิดปัญหาการแสดงออกที่มีลักษณะแปลกๆ ที่ไม่เหมาะสม โดยลักษณะของสคริปต์คำสั่งนี้มักจะเป็นรูปแบบตายตัว มีความยาวและความซับซ้อนสูง ซึ่งทำให้เกิดปัญหาคือ ปัญหาของความซับซ้อน (problem of complexity) และปัญหาของการปรับตัว (problem of adaptability) ซึ่งมีรายละเอียดดังนี้

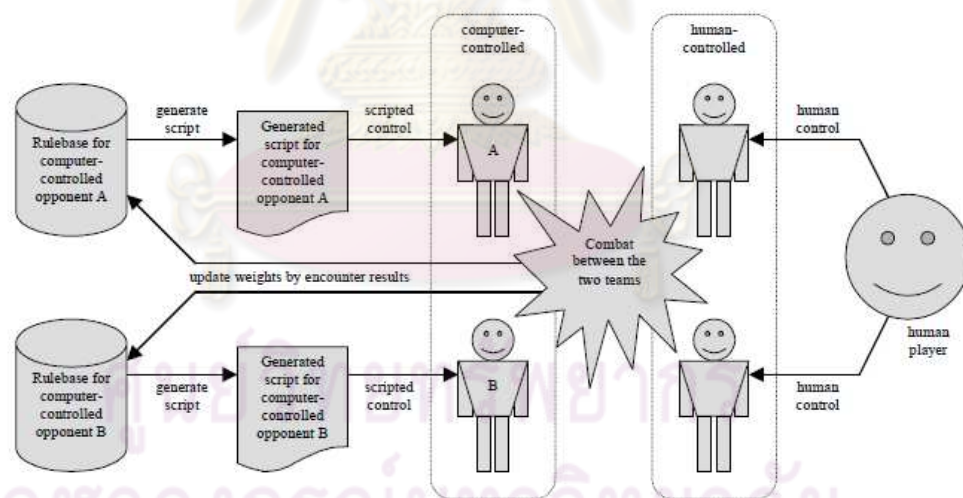
- ปัญหาของความซับซ้อน เนื่องจากการกระทำมีความซับซ้อนมากจึงทำให้มักจะมีจุดอ่อนอยู่ ทำให้ถูกผู้เล่นจับจุดและเอาชนะได้โดยง่าย
- ปัญหาของการปรับตัว เนื่องจากการกระทำที่กำหนดมีรูปแบบตายตัว ทำให้อาจไม่เหมาะกับผู้เล่นที่มีระดับความสามารถต่างๆ กันไปผู้เล่นอาจพบว่าเกมยากเกินไปหรือง่ายเกินไป

ปัญหาเหล่านี้ทำให้คุณค่าทางด้านความบันเทิงของตัวเกมลดลง การแก้ปัญหาเหล่านี้ทำได้โดยการนำเอา เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เข้ามาช่วย ซึ่งทำได้สองแบบ คือ การเรียนรู้แบบออฟไลน์ (Offline Learning) ซึ่งเป็นการแก้ไขปัญหาก่อนที่เกมจะออกวางจำหน่ายโดยทำการทดลองปรับความสมดุลของเกม และการเรียนรู้แบบออนไลน์ (Online Learning) ซึ่งจะเป็นการแก้ปัญหาหลังจากเกมได้วางจำหน่ายไปแล้ว โดยระบบจะทำการเรียนรู้ระหว่างที่เกมกำลังถูกเล่นอยู่ งานของ Spronck นำเสนอไปทางการเรียนรู้แบบออนไลน์เพื่อให้ปัญญาประดิษฐ์มีความสามารถปรับตัวให้เก่งขึ้นเพื่อเอาชนะผู้เล่นได้ในเวลาอันสั้น โดยการเรียนรู้แบบออนไลน์สามารถที่จะช่วยให้ระบบปัญญาประดิษฐ์สามารถซ่อมแซมจุดอ่อนในสคริปต์ของตนเอง และสามารถปรับตัวตามลักษณะการเล่นของผู้เล่นได้ ซึ่งเป็นการแก้ปัญหาของความซับซ้อนและปัญหาของการปรับตัวที่เกิดจากการใช้สคริปต์ในแง่ของการทำ

ให้ปัญญาประดิษฐ์ฉลาดขึ้น โดย Spronck เสนอว่า ระบบเรียนรู้ที่จะนำมาใช้นั้นจะต้องมีคุณสมบัติ คือ

- ความรวดเร็ว (Fast) เนื่องจากการเรียนรู้แบบออนไลน์เกิดขึ้นระหว่างการเล่นเกม ระบบการเรียนรู้จึงจำเป็นต้องมีความรวดเร็วในการเรียนรู้เพื่อไม่ให้ส่งผลกระทบต่อการเล่น และการทำงานของเกม
- ประสิทธิภาพ (Effective) เพื่อให้ผู้เล่นได้รับความบันเทิง สคริปต์ที่มีการปรับตัวจะต้องให้ความท้าทายได้อย่างน้อยเท่ากับสคริปต์ที่ถูกออกแบบมาตามปกติ
- ทนทาน (Robust) ระบบการเรียนรู้จะต้องสามารถครอบคลุมถึงความไม่แน่นอนที่มีอยู่ในระบบของตัวเกมปกติได้
- ประสิทธิภาพ (Efficient) ในเกมนั้นผู้เล่นจะได้พบกับคู่ต่อสู้เป็นจำนวนจำกัด ดังนั้นระบบการเรียนรู้ต้องสามารถเรียนรู้ได้จากการทดสอบเป็นจำนวนครั้งที่ไม่มาก

Spronck ได้นำเสนอระบบการเรียนรู้แบบออนไลน์ที่เรียกว่า ไดนามิกสคริปต์ ซึ่งมีคุณสมบัติสี่ข้อตามที่กล่าวข้างต้น โดยมุ่งเน้นให้ใช้กับเกมที่มีอยู่ในท้องตลาดจริงๆ ได้



รูปที่ 2.4 การทำงานของไดนามิกสคริปต์ [11]

จากรูปที่ 2.4 ในการทำงานของไดนามิกสคริปต์นั้น เริ่มแรกผู้สร้างเกมจะต้องออกแบบกฎพื้นฐาน (Rulebase) ขึ้นมาก่อน ในกฎพื้นฐานนั้นจะประกอบด้วยกฎ (Rule) โดยกฎนั้นอยู่ในรูปแบบของสคริปต์ที่ผู้สร้างเกมใช้ความรู้และประสบการณ์ที่มีอยู่ (Domain Knowledge) กำหนดขึ้นมา จากนั้นในสถานการณ์ต่างๆ ระบบจะทำการเลือกกฎจากกฎพื้นฐานไปสร้างเป็นชุดสคริปต์ โดยเลือกตามค่าน้ำหนัก (Weight) โดยกฎที่มีค่าน้ำหนักมากจะมีโอกาส

ถูกเลือกไปใช้งานมาก เมื่อเล่นกับผู้เล่นไปจนรู้แพ้วชนะในหนึ่งรอบการเล่นแล้ว ระบบจะทำการประเมินผลการเล่นและปรับค่าน้ำหนักของกฎแต่ละข้อตามผลการเล่นที่ประเมินได้ โดยกฎที่ทำให้ได้ผลดีจะได้รางวัลโดยการปรับค่าน้ำหนักขึ้น กฎที่ทำให้ได้ผลไม่ดีจะถูกลงโทษโดยการปรับค่าน้ำหนักลง ในการเล่นครั้งต่อไปศัตรูที่สร้างมาจากกฎเหล่านี้ก็จะมีความสามารถมากขึ้น และปรับตัวตามการเล่นของผู้เล่น โดยในงานของ Spronck นี้ได้นำเทคนิคนี้ไปทดสอบกับเกมสองเกม คือ เกมที่ทดลองสร้างเลียนแบบเกม BALDUR'S GATE กับเกมที่อยู่ในตลาดจริงๆ คือ เกม NEVERWINTER NIGHTS ผลการทดลองได้แสดงให้เห็นว่าเทคนิคไดนามิกสคริปต์สามารถนำไปใช้งานได้จริง อีกทั้งยังเป็นการปรับปรุงการแสดงออกทำให้ตัวละครหรือสภาพแวดล้อมในเกมมีการแสดงออกที่หลากหลายเหมาะสมกับสถานการณ์ ในขณะที่ยังคงความสามารถในการแสดงออกเฉพาะสิ่งที่ถูกกำหนดไว้ก่อนแล้วได้ ทำให้ไม่เกิดปัญหาเกมมีการแสดงออกผิดปกติหรือไม่เหมาะสม

Algorithm 1 : Script Generation

```

1:  ClearScript()
2:  sumweights = 0
3:  for i=0 to rulecount-1 do
4:      sumweights = sumweights + rule[i].weight
5:  end for
6:  {Repeated roulette wheel selection}
7:  for i=0 to scriptsize-1 do
8:      try = 0; lineadded = false
9:      while try < maxtries and not lineadded do
10:         j=0; sum=0; selected=-1
11:         fraction = random(sumweights)
12:         while selected < 0 do
13:             sum = sum + rule[j].weight
14:             if sum > fraction then
15:                 selected = j
16:             else
17:                 j = j+1
18:             end if
19:         end while
20:         lineadded = InsertInScript(rule[selected].line)
21:         try = try+1
22:     end while
23: end for
24: Finish Script()

```

กฎพื้นฐาน (Rulebase) เป็นอาเรย์ที่ประกอบด้วยกฎ (rule) ทุกๆ rule จะประกอบไปด้วยคุณลักษณะ 3 ข้อคือ

1. weight ใช้เก็บค่าน้ำหนักของกฎนั้น
2. line ใช้เก็บการกระทำของกฎนั้น ซึ่งจะถูกเพิ่มเข้าไปในชุดสคริปต์ เมื่อกฎนั้นถูกเลือก
3. activated ใช้เก็บค่าตรรกะที่แสดงว่ากฎนั้นถูกเลือก

รูปที่ 2.5 แสดงวิธีการสร้างชุดสคริปต์ ฟังก์ชัน InsertInScript ทำหน้าที่เพิ่ม line เข้าไปในชุดสคริปต์ ถ้า line นั้นอยู่ในชุดสคริปต์อยู่แล้วฟังก์ชันจะไม่ทำอะไรและคืนค่า 'false' กลับมา แต่ถ้า line ถูกเพิ่มเข้าไปได้สำเร็จฟังก์ชันจะคืนค่า 'true' กลับมา อัลกอริทึมนี้มีเป้าหมายที่จะได้ชุดสคริปต์ที่มีจำนวนกฎจำนวน scriptsize กฎ แต่ชุดสคริปต์จริงๆ อาจจะได้น้อยกว่านั้นถ้าค่าการพยายามเพิ่มบรรทัดนั้นเกินกว่าค่า maxtry ฟังก์ชัน Finish Script ทำหน้าที่เพิ่มบรรทัดปิดเพื่อให้แน่ใจว่าชุดสคริปต์นั้นจะมีคำสั่งที่จะถูกเรียกใช้งานได้เสมอ

รูปที่ 2.6 แสดงวิธีการปรับค่าน้ำหนัก ฟังก์ชัน CalculateAdjustment ทำหน้าที่คำนวณค่ารางวัล (Reward) หรือค่าโทษ (Penalty) ซึ่งแต่ละกฎที่ถูก activate จะได้รับค่าพารามิเตอร์ Fitness ใช้วัดผลของชุดสคริปต์ที่เลือกไปใช้ ฟังก์ชัน DistributeRemainder ทำหน้าที่กระจายผลต่างระหว่างค่าน้ำหนักขณะนั้นกับค่าน้ำหนักเริ่มต้นไปทั่วค่าน้ำหนักอื่นๆ จะเห็นว่าค่า w_{max} (maxweight) จะเป็นตัวกำหนดน้ำหนักสูงสุดที่เป็นไปได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 2 : Weight Adjustment

```

1:  active = 0
2:  for i=0 to rulecount-1 do
3:      If rule[i].activated then
4:          active = active+1
5:      end if
6:  end for
7:  if active<=0 or active>=rulecount then
8:      return (no updates are needed.)
9:  end if
10: nonactive = rulecount - active
11: adjustment = CalculateAdjustment(Fitness)
12: compensation = -active * adjustment/nonactive
13: remainder = 0
14: Credit assignment
15: for i=0 to rulecount-1 do
16:     if rule[i].activated then
17:         rule[i].weight = rule[i].weight + adjustment
18:     else
19:         rule[i].weight = rule[i].weight + compensation
20:     end if
21:     if rule[i].weight < minweight then
22:         remainder = remainder + (rule[i].weight - minweight)
23:         rule[i].weight = minweight
24:     else if rule[i].weight > maxweight then
25:         remainder = remainder + (rule[i].weight - maxweight)
26:         rule[i].weight = maxweight
27:     end if
28: end for
29: DistributeRemainder();

```

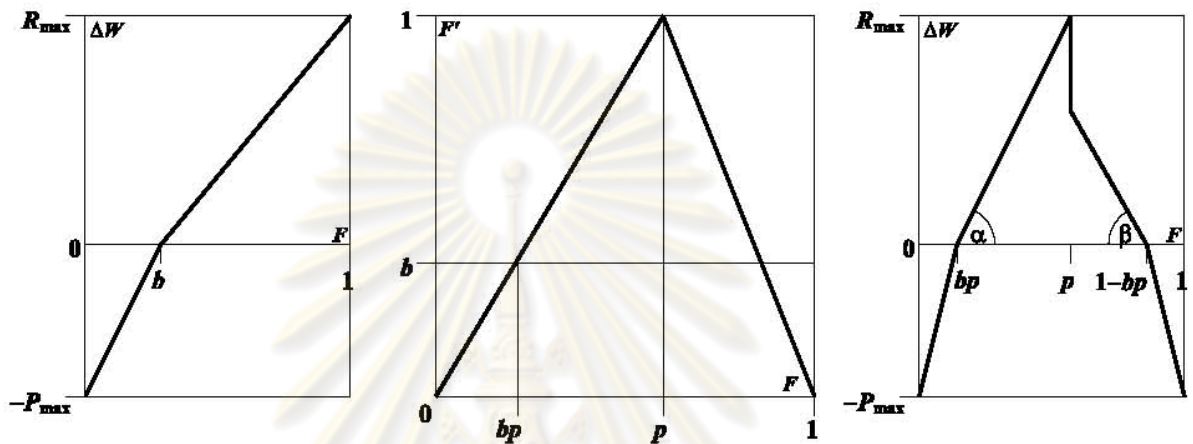
รูปที่ 2.6 การปรับค่าน้ำหนัก [12]

ในงานวิจัยต่อเนื่องชิ้นถัดมา [13] Spronck ได้นำเสนอเทคนิคการปรับความยากของเกมสวมบทบาท 3 วิธี คือ

2.2.1.1 การลงโทษค่าฟิตเนสที่สูง (High-Fitness Penalizing)

ปกติการปรับค่าน้ำหนักของไดนามิกสคริปต์ กฎที่นำไปใช้แล้วมีประสิทธิภาพดี (ค่าที่ได้จากฟังก์ชันวัดความเหมาะสมสูง) จะได้รับค่ารางวัล (Reward) สูง ดังรูปที่ 2.7 (ซ้าย) ซึ่งจะทำให้กฎนั้นได้รับการปรับค่าน้ำหนักมากด้วย เทคนิคการลงโทษค่าฟิตเนสที่สูงมีหลักการ

คือ จะเปลี่ยนค่าที่ได้จากฟังก์ชันวัดความเหมาะสมโดยให้ค่าที่ได้จากฟังก์ชันวัดความเหมาะสมระดับกลางแทนพฤติกรรมของปัญหาประติษฐ์ที่ดี ส่วนค่าที่ได้จากฟังก์ชันวัดความเหมาะสมระดับสูงแทนพฤติกรรมของปัญหาประติษฐ์ที่ไม่ดีแทน ดังรูปที่ 2.7 (กลาง) ซึ่งเมื่อเปลี่ยนเป็นการปรับค่าน้ำหนักแล้วจะทำให้ค่าที่ได้จากฟังก์ชันวัดความเหมาะสมระดับกลางได้รับการปรับค่าน้ำหนักมาก แต่ค่าที่ได้จากฟังก์ชันวัดความเหมาะสมระดับสูงจะได้รับการปรับค่าน้ำหนักน้อยแทน ดังรูปที่ 2.7 (ขวา)



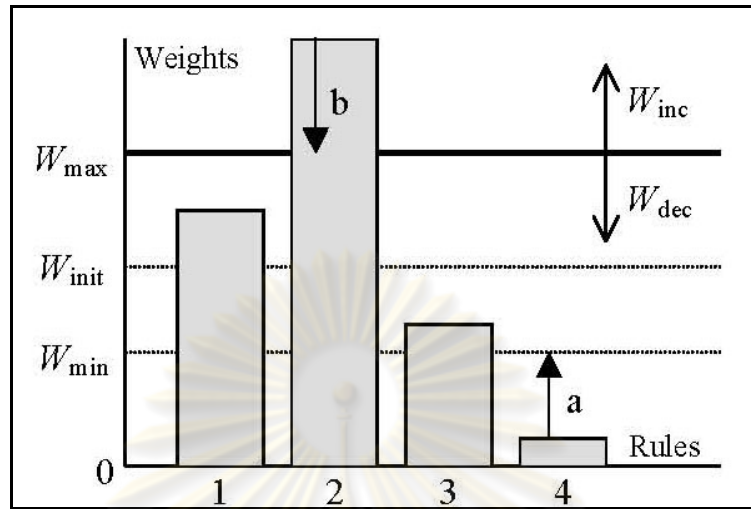
รูปที่ 2.7 การลดโทษค่าฟิตเนสที่สูง [12]

2.2.1.2 การตัดค่าน้ำหนัก (Weight Clipping)

ค่าน้ำหนักสูงสุด (w_{max}) เป็นค่าที่กำหนดระดับการเติบโตของค่าน้ำหนัก ค่าน้ำหนักสูงสุดที่สูงจะทำให้น้ำหนักสามารถเติบโตได้มาก ดังนั้นเมื่อกฎที่มีประสิทธิภาพดีถูกเลือกบ่อยครั้ง จะทำให้ไดนามิกสคริปต์ปรับตัวจนถึงระดับที่เก่งที่สุด แต่ถ้าเราจำกัดค่าน้ำหนักสูงสุดให้ต่ำ จะทำให้ความแตกต่างของค่าน้ำหนักของกฎในกฎพื้นฐานน้อยลงเพราะค่าน้ำหนักที่เกินจากค่าน้ำหนักสูงสุดจะถูกกระจายไปให้กฎอื่นๆ โอกาสในการเลือกกฎที่มีประสิทธิภาพดีจะถูกกระจายไปให้กฎที่มีประสิทธิภาพต่ำกว่าด้วย เทคนิคการตัดค่าน้ำหนักมีหลักการคือจะปรับค่าน้ำหนักสูงสุดโดยมีเป้าหมายคือให้ค่าน้ำหนักสูงสุดต่ำลงเมื่อปัญหาประติษฐ์ชนะบ่อยครั้ง และปรับให้ค่าน้ำหนักสูงสุดสูงขึ้นเมื่อปัญหาประติษฐ์แพ้บ่อยครั้ง โดยมีวิธีการทำงานคือเมื่อจบการปะทะครั้งนั้นถ้าปัญหาประติษฐ์ชนะจะปรับลดค่าน้ำหนักสูงสุดลง $w_{dec}\%$ ของค่าน้ำหนักสูงสุด แต่จะต้องไม่ต่ำกว่าค่าน้ำหนักเริ่มต้น (w_{init}) แต่ถ้าปัญหาประติษฐ์แพ้จะปรับเพิ่มค่าน้ำหนักสูงสุดขึ้น $w_{inc}\%$ ของค่าน้ำหนักสูงสุด

จากรูปที่ 2.8 ถ้าใช้การตัดค่าน้ำหนัก ก่อนการปรับค่าน้ำหนัก ค่าน้ำหนักสูงสุดจะถูกปรับ $w_{dec}\%$ หรือ $w_{inc}\%$ มาแล้วขึ้นอยู่กับผลปะทะ หลังการปรับค่าน้ำหนัก ค่าน้ำหนักของ

กฎ 4 ต่ำกว่าค่าน้ำหนักต่ำสุด (w_{min}) จึงถูกปรับขึ้นมาเป็นค่าน้ำหนักต่ำสุดแทน ส่วนค่าน้ำหนักของกฎ 2 สูงกว่าค่าน้ำหนักสูงสุด (w_{max}) จึงถูกปรับลงมาเป็นค่าน้ำหนักสูงสุดแทน



รูปที่ 2.8 Weight Clipping และ Top Culling [13]

2.2.1.3 การตัดค่าน้ำหนักที่สูง (Top Culling)

ใช้วิธีการปรับค่าน้ำหนักสูงสุดเช่นเดียวกับการตัดค่าน้ำหนัก แต่จะแตกต่างกันที่วิธีการตัดค่าน้ำหนักที่สูงนั้นจะอนุญาตให้ค่าน้ำหนักสามารถเติบโตเลยค่าน้ำหนักสูงสุดได้ โดยกำหนดเพิ่มเติมว่าเมื่อปัญหาประติมากรรมชะลอครั้ง ค่าน้ำหนักสูงสุดจะลดลง กฎที่มีค่าน้ำหนักมากกว่าค่าน้ำหนักสูงสุดจะไม่สามารถเลือกมาสร้างเป็นกฎพื้นฐานได้ ปัญหาประติมากรรมจะสามารถเลือกกฎมาสร้างเป็นกฎพื้นฐานได้เฉพาะกฎที่อ่อนแอ ในทางตรงกันข้ามเมื่อปัญหาประติมากรรมแพ้อยครั้ง ค่าน้ำหนักสูงสุดจะเพิ่มขึ้น กฎที่มีน้ำหนักมากกว่าค่าน้ำหนักสูงสุดในครั้งก่อนอาจจะกลับมาเลือกได้อีกครั้ง ถ้าน้ำหนักไม่เลยค่าน้ำหนักสูงสุด ปัญหาประติมากรรมจะสามารถเลือกกฎที่ประสิทธิภาพดีได้อีกครั้งหนึ่ง

จากรูปที่ 2.8 ถ้าใช้การตัดค่าน้ำหนักที่สูง ค่าน้ำหนักสูงสุดจะถูกปรับ $w_{dec}\%$ หรือ $w_{inc}\%$ มาแล้วขึ้นอยู่กับผลปะทะเหมือนการตัดค่าน้ำหนัก การตัดค่าน้ำหนักที่สูงจะไม่เปลี่ยนแปลงค่าน้ำหนักของกฎ 2 (ไม่ทำการกระทำ b) แต่กฎ 2 จะไม่สามารถเลือกมาสร้างเป็นชุดสคริปต์ได้เพราะว่ามีค่าน้ำหนักเกินค่าน้ำหนักสูงสุด

ซึ่ง Spronck ได้ทดสอบโดยการให้ปัญหาประติมากรรมกับผู้เล่น (แทนด้วยสคริปต์ที่ไม่มีการปรับตัว) โดยให้ต่อสู้กัน 150 ครั้ง 50 ครั้งแรกเป็นการให้ไดนามิกสคริปต์ทำการเรียนรู้ก่อน 100 ครั้งหลังถึงเก็บผลการทดลองโดยการวัดจำนวนที่ปัญหาประติมากรรมชนะผู้เล่นใกล้เคียง 50 ครั้ง (Spronck ให้อยู่ในช่วง 45-55 ครั้ง) ถือว่าการปรับความยากวิธีนั้นมี

ประสิทธิภาพดี ซึ่งจากผลการทดลองพบว่าวิธีการตัดค่าน้ำหนักที่สูงนั้นให้ผลการทดลองที่ดีที่สุด รองลงมาคือ วิธีการตัดค่าน้ำหนัก

วิทยานิพนธ์ฉบับนี้จะนำเอาเทคนิคไดนามิกสคริปต์และเทคนิคการปรับความยากแบบการตัดค่าน้ำหนักที่สูงมาปรับปรุงเพื่อสร้างปัญญาประดิษฐ์ที่สู้กับผู้เล่นในเกมวางแผนการรบแบบตอบสนองทันที

2.2.2 ฟังก์ชันวัดคะแนนสำหรับเกมสปริง

Sander Bakkes [9, 14, 15, 16] ได้สร้างฟังก์ชันวัดคะแนนในเกมสปริงขึ้นโดยมีเป้าหมาย คือ

1. สามารถทำนายผลการเล่นได้ว่าฝ่ายใดจะเป็นฝ่ายแพ้หรือชนะ
2. สามารถวัดได้ว่าปัญญาประดิษฐ์หรือผู้เล่นเก่งกว่ากันมากน้อยเพียงใด
3. สามารถนำผลการวัดไปเป็นข้อมูลป้อนกลับให้ปัญญาประดิษฐ์สามารถตัดสินใจเลือกชุดคำสั่งที่มีประสิทธิภาพมากที่สุด

โดยได้กำหนด ฟังก์ชันวัดคะแนน ดังสมการที่ (2.3)

$$v(p) = w_p v_1 + (1 - w_p) v_2 \quad (2.3)$$

โดย

$v(p)$ คือ ผลลัพธ์ของฟังก์ชันวัดคะแนน

v_1 คือ Material Strength

v_2 คือ Position Safety

w_p คือ ตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของ

เกม

$p \in N$ คือ ตัวแปรแทนเฟสของเกม

เทอม v_1 (Material Strength) แทน ฟังก์ชันวัดคะแนนส่วนที่วัดคะแนนส่วนยูนิท ดังสมการที่ (2.4)

$$v_1 = \sum_u w_u \left(C_{u_0} - \frac{C_{u_1}}{R} \right) \quad (2.4)$$

โดย

w_u คือ น้ำหนักของยูนิทชนิด u

C_{u_0} คือ จำนวนยูนิทชนิด u ของฝ่ายปัญญาประดิษฐ์

C_{u_1} คือ จำนวนยูนิทชนิด u ของฝ่ายตรงข้าม

$R \in [0,1]$ คือ เศษส่วนของสภาพแวดล้อมที่ฝ่ายปัญญาประดิษฐ์มองเห็น

เทอม v_2 (Position Safety) แทน ฟังก์ชันวัดคะแนนส่วนที่วัดคะแนนส่วนความปลอดภัยของ Commander ดังสมการที่ (2.5)

$$v_2 = \sum_{r \in D} w_r \left(\frac{O_{r_1}}{R_{r_1}} - \frac{O_{r_0}}{R_{r_0}} \right) \quad (2.5)$$

โดย

w_r คือ น้ำหนักของรัศมี r มีค่า 0.75 เมื่อรัศมีเท่ากับ 500, 0.20 เมื่อรัศมีเท่ากับ 1000 และ 0.05 เมื่อรัศมีเท่ากับ 2000

O_{r_1} คือ จำนวนของยูนิทของฝ่ายปัญญาประดิษฐ์ที่ฝ่ายตรงข้ามมองเห็นภายในรัศมี r รอบ Commander ฝ่ายตรงข้าม

$R_{r_1} \in [0,1]$ คือ เศษส่วนของสภาพแวดล้อมภายในรัศมี r ที่ฝ่ายตรงข้ามมองเห็น

O_{r_0} คือ จำนวนของยูนิทของฝ่ายตรงข้ามที่ฝ่ายปัญญาประดิษฐ์มองเห็นภายในรัศมี r รอบ Commander ฝ่ายปัญญาประดิษฐ์

$R_{r_0} \in [0,1]$ คือ เศษส่วนของสภาพแวดล้อมภายในรัศมี r ที่ฝ่ายปัญญาประดิษฐ์มองเห็น

โดยถ้าผลของฟังก์ชันวัดคะแนนเป็นบวก จะหมายถึงฝ่ายปัญญาประดิษฐ์ชนะ ฝ่ายผู้เล่น แต่ถ้าผลของฟังก์ชันวัดคะแนนเป็นลบ จะหมายถึงฝ่ายผู้เล่นชนะฝ่ายปัญญาประดิษฐ์

สำหรับค่าของน้ำหนักของแต่ละยูนิตนั้น Bakkes ได้นำการเรียนรู้แบบ Temporal Difference ซึ่งใช้ในการเรียนรู้ค่าของตัวหมากในเกมหมากรุกโดย Beal & Smith [10] มาประยุกต์ใช้ในการหาน้ำหนักของยูนิต โดยใช้สมการที่ (2.2)

เมื่อ

ΔW_t คือ ค่าน้ำหนักของยูนิตที่เปลี่ยนแปลงไปที่เวลา t

P คือ เซตของ Temporally successive prediction

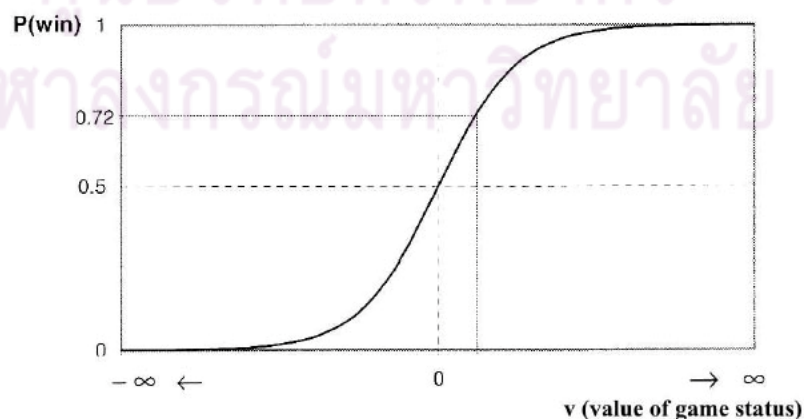
α คือ อัตราการเรียนรู้ (Learning rate) กำหนดให้เป็น 0.1 ตามการทดลองของ Beal & Smith

λ คือ ตัวแปรควบคุมน้ำหนักของ $\nabla_w P_k$ ที่เกิดขึ้น k ชั้นก่อนหน้า กำหนดให้เป็น 0.95 ตามการทดลองของ Beal & Smith

$\nabla_w P_k$ คือ เกรเดียนของ P_k เทียบกับ w

การนำการเรียนรู้แบบ Temporal Difference มาใช้นั้น ต้องประกอบด้วยเซตของ Temporally successive prediction ในงานของ Bakkes จึงได้ให้ความน่าจะเป็นในการทำนาย (prediction probability) ของเกมสถานะ v_t ($P(v_t)$) ดังสมการที่ (2.6)

$$P(v_t) = \frac{1}{1 + e^{-v_t}} \quad (2.6)$$



รูปที่ 2.9: ค่าฟังก์ชันวัดคะแนน 0.942 ถูกเปลี่ยนเป็นความน่าจะเป็นในการทำนาย 0.72 [9]

ซึ่งคือซิกมอยด์ฟังก์ชันที่ทำหน้าที่เปลี่ยนค่าฟังก์ชันวัดคะแนน ณ สถานะนั้น เป็นความน่าจะเป็นในการทำนาย ดังรูปที่ 2.9

การเรียนรู้ค่าน้ำหนักของยูนิตจะทำในสภาพแวดล้อมข้อมูลสมบูรณซึ่งค่า R คือ 1 จะได้สมการดังสมการที่ (2.7)

$$v = \sum_u w_u (c_{u_0} - c_{u_1}) \quad (2.7)$$

จากสมการที่ (2.6) เมื่อหาอนุพันธ์ของซิกมอยด์ฟังก์ชันจะได้ดังสมการที่ (2.8)

$$\frac{dP}{dv} = P(1-P) \quad (2.8)$$

ดังนั้นเมื่อหาส่วน ∇_{wP_k} ในสมการที่ (2.2) จะได้ดังสมการที่ (2.9)

$$\nabla_{wP_k} = (c_{u_0} - c_{u_1}) P_k (1 - P_k) \quad (2.9)$$

Bakkes ได้เก็บค่าน้ำหนักของยูนิตและพบว่าค่าน้ำหนักของยูนิตนั้นเป็นค่าที่บ่งบอกความสำคัญของยูนิตนั้นเฉพาะแผนที่ ซึ่งถ้าต้องการใช้ค่าน้ำหนักของยูนิตในแผนที่ใดก็ต้องทำการเรียนรู้ค่าน้ำหนักของยูนิตในแผนที่นั้นก่อน

Bakkes ได้แบ่งการทดลองออกเป็น 5 เฟสเนื่องจากความสำคัญของค่า v_1 และ v_2 ในแต่ละช่วงของเกมไม่เท่ากัน ดังนี้ คือ

- เฟส 1 ตรวจสอบสิ่งก่อสร้างระดับ 1
- เฟส 2 ตรวจสอบยูนิตระดับ 1 ที่ใช้เวลาในการสร้างมากกว่าหรือเท่ากับ 2500
- เฟส 3 ตรวจสอบสิ่งก่อสร้างระดับ 2
- เฟส 4 ตรวจสอบยูนิตระดับ 2 ที่ใช้เวลาในการสร้างมากกว่าหรือเท่ากับ 15,000
- เฟส 5 ตรวจสอบยูนิตหรือสิ่งก่อสร้างระดับ 3

โดยค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) Bakkes ได้หาค่าโดยใช้หลักการเคลื่อนลงตามความชัน (Gradient Descent) และทดสอบได้ผลว่า ถ้าปรับให้ตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกมเหมาะสมกับเฟสนั้นๆ ฟังก์ชันวัดคะแนนของเขาจะสามารถทำนายผลลัพธ์ได้ถูกต้องมากขึ้น

FRIENDLY TEAM	ENEMY TEAM	#GAMES IN TRAINING SET (Collected with perfect information)	#GAMES IN TEST SET (Collected with perfect information)	#GAMES IN TEST SET (Collected with imperfect information)
AAI	AAI (self-play)	500	200	200
AAI	TSI	100	200	200
AAI	CSAI	100	200	200
AAI	RAI	-	200	200

รูปที่ 2.10 จำนวนเกมที่เก็บข้อมูลจากการทดลองของ Bakkes [14]

สำหรับการทดสอบ Bakkes ได้ทำการเก็บข้อมูลในเกมเพื่อนำมาคิดค่าน้ำหนักของยูนิตจากเกมจำนวน 700 เกม คือ AAI สู้กับ AAI, AAI สู้กับ TSI, AAI สู้กับ CSAI ส่วน AAI สู้กับ RAI นั้นไม่ได้เก็บข้อมูล เนื่องจากต้องการทดสอบความครอบคลุมว่าค่าน้ำหนักของยูนิตนั้นสามารถนำไปเป็นส่วนประกอบของฟังก์ชันวัดคะแนนเพื่อวัดคะแนนในคู่ปัญญาประดิษฐ์คู่อื่นได้ด้วย ในการทดสอบประสิทธิภาพของฟังก์ชันวัดคะแนน Bakkes ได้แบ่งเป็นการทดสอบในสภาพแวดล้อมข้อมูลสมบูรณ์ และ สภาพแวดล้อมข้อมูลไม่สมบูรณ์ โดยมีจำนวนเกมที่เก็บผลการทดสอบในแต่ละสภาพแวดล้อมจำนวนสภาพแวดล้อมละ 800 เกม ดังรูปที่ 2.10

Bakkes ได้แบ่งการวัดผลประสิทธิภาพของฟังก์ชันวัดคะแนนเป็น 2 วิธี คือ

1. วัดผลว่าฟังก์ชันวัดคะแนนสามารถทำนายผลการเล่นครั้งนั้นได้ กำหนด ค่าความแม่นยำในการทำนายผลลัพธ์ท้ายสุด (Final Prediction Accuracy) เพื่อวัด โดยหมายถึง เปอร์เซ็นต์ของเกม que ฟังก์ชันวัดคะแนนสามารถทำนายผลลัพธ์ในเวลาใกล้จะจบเกม ซึ่งค่านี้ยิ่งมากแสดงว่าฟังก์ชันวัดคะแนนมีประสิทธิภาพดี
2. วัดผลว่าฟังก์ชันวัดคะแนนสามารถทำนายผลลัพธ์ได้ทั่วถึงทั้งเกม กำหนดค่าการทำนายความสัมพันธ์ (Relative Prediction) เพื่อวัด ค่าการทำนายความสัมพันธ์ยิ่งน้อยแสดงว่าฟังก์ชันวัดคะแนนสามารถทำนายผลลัพธ์ในช่วงนั้นได้ดี โดยแบ่งเป็น
 - Weak Relative Prediction หมายถึง เปอร์เซ็นต์ของเวลาในเกมที่ผลลัพธ์อย่างน้อย 50% ของเกมทดสอบสามารถทำนายได้ถูกต้อง (แทนช่วงต้นเกม)
 - Normal Relative Prediction หมายถึง เปอร์เซ็นต์ของเวลาในเกมที่ผลลัพธ์อย่างน้อย 60% ของเกมทดสอบสามารถทำนายได้ถูกต้อง (แทนช่วงกลางเกม)
 - Strong Relative Prediction หมายถึง เปอร์เซ็นต์ของเวลาในเกมที่ผลลัพธ์อย่างน้อย 70% ของเกมทดสอบสามารถทำนายได้ถูกต้อง (แทนช่วงท้ายเกม)

Bakkes ได้ทดสอบและสรุปผลว่า ฟังก์ชันวัดคะแนนของเขาสามารถทำนายผลลัพธ์ทำเกมและทำนายผลลัพธ์ในแต่ละช่วงของเกมทั้งในสภาพแวดล้อมข้อมูลสมบูรณ์ และสภาพแวดล้อมข้อมูลไม่สมบูรณ์ได้ดี

วิทยานิพนธ์ฉบับนี้จะนำเอาฟังก์ชันวัดคะแนนของ Bakkes มาใช้ในการวัดระดับของปัญญาประดิษฐ์ โดยวัดเฉพาะในสภาพแวดล้อมข้อมูลสมบูรณ์



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

วิธีการสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นโดยใช้การปรับ การผลิตยูนิตในเกมวางแผนการรบแบบตอบสนองทันที

การสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นในเกมวางแผนการรบแบบตอบสนองทันทีในวิทยานิพนธ์นี้จะนำเทคนิคการปรับตัวแบบไดนามิกสคริปต์มาปรับปรุงให้สามารถปรับตัวให้เข้ากับผู้เล่นขณะที่ผู้เล่นกำลังเล่นเกมอยู่ โดยใช้วิธีการตัดค่าน้ำหนักที่สูง และใช้การผลิตยูนิตมาเป็นกฎพื้นฐาน

สำหรับบทที่ 3 นี้ จะแบ่งเป็นสองส่วน ส่วนแรกนำเสนอการใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงในการสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่น ด้วยการควบคุมการผลิตยูนิต อีกส่วนหนึ่งจะเป็นการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้กฎพื้นฐานที่เป็นการผลิตยูนิตด้วยเทคนิคเสริมต่างๆ

3.1 การใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงในการสร้างปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นด้วยการควบคุมการผลิตยูนิต

3.1.1 การสร้างโมดูลเพื่อควบคุมการผลิตยูนิตในปัญญาประดิษฐ์ E323

การตัดสินใจของปัญญาประดิษฐ์ในเกมวางแผนการรบแบบตอบสนองทันทีนั้นแบ่งระดับการตัดสินใจออกเป็น 3 ระดับ [4] คือ

- ระดับปฏิบัติการ (Operation Level) คือ ระดับการตัดสินใจระดับล่างที่ใช้ควบคุมยูนิตตัวเดียว เช่น สั่งยูนิตให้เคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่งของแผนที่ สั่งยูนิตให้โจมตียูนิตหรือสิ่งก่อสร้างฝ่ายตรงข้าม
- ระดับยุทธวิธี (Tactical Level) คือ ระดับการตัดสินใจระดับกลางที่มีชุดคำสั่งลำดับไว้ให้ปฏิบัติตามเพื่อบรรลุจุดหมายใดจุดหมายหนึ่ง เช่น กลุ่มของยูนิตถูกสั่งให้เข้าโจมตีฐานของฝ่ายตรงข้ามตามขั้นตอนในชุดคำสั่ง หรือ กลุ่มของยูนิตถูกสั่งให้ทำการป้องกันฐานของฝ่ายตนเองตามขั้นตอนในชุดคำสั่ง
- ระดับกลยุทธ์ (Strategic Level) คือ ระดับการตัดสินใจระดับบนที่สร้างแผนการของการเล่นเกมนั้นขึ้น เช่น การสร้างสิ่งก่อสร้าง การสร้างยูนิต การวิจัยเพื่อพัฒนาเทคโนโลยีระดับถัดไป การวางแผนการรบ การตัดสินใจว่าขณะนั้นควร

จะทำการโจมตีหรือตั้งรับ การตัดสินใจว่าการไปโจมตีครั้งนั้นๆ ควรจะใช้กองกำลังระดับไหนประกอบด้วยยูนิตชนิดใดบ้าง

ในขั้นตอนการสร้างโมดูลเพื่อควบคุมอัตราการผลิตนี้ ทางผู้เขียนให้ปัญญาประดิษฐ์ E323 เป็นตัวควบคุมการตัดสินใจในระดับปฏิบัติการ ระดับยุทธวิธี และการตัดสินใจบางส่วนในระดับกลยุทธ์ โดยได้สร้างโมดูลเพื่อควบคุมปัญญาประดิษฐ์ E323 เฉพาะส่วนการสร้างยูนิตว่าจะผลิตยูนิตใด ด้วยจำนวนเท่าไร

3.1.2 การสร้างโมดูลสำหรับเก็บข้อมูลของเกม

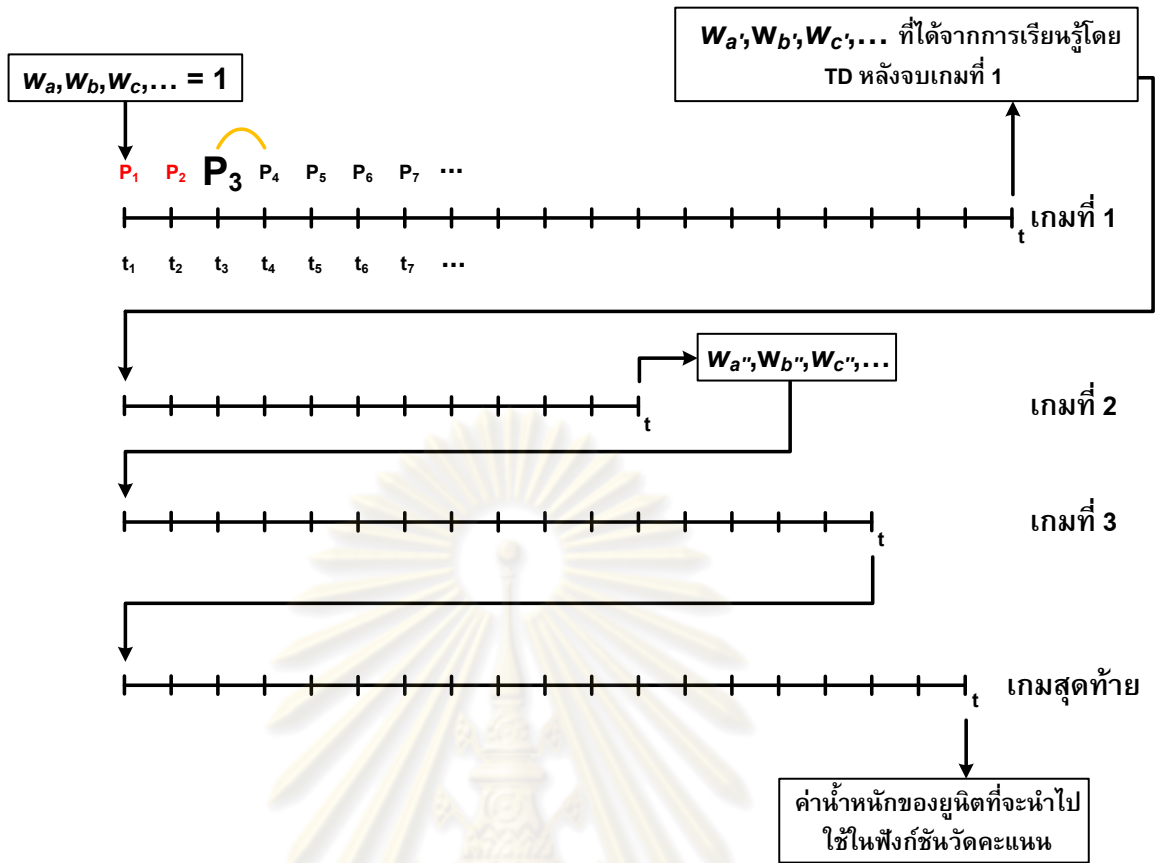
ข้อมูลที่ปัญญาประดิษฐ์ต้องใช้งาน มีดังต่อไปนี้

- จำนวนยูนิตแต่ละชนิดของฝ่ายปัญญาประดิษฐ์และฝ่ายศัตรู
- จำนวนยูนิตศัตรูในรัศมี 2000 รอบ Commander ฝ่ายปัญญาประดิษฐ์
- จำนวนยูนิตศัตรูในรัศมี 1000 รอบ Commander ฝ่ายปัญญาประดิษฐ์
- จำนวนยูนิตศัตรูในรัศมี 500 รอบ Commander ฝ่ายปัญญาประดิษฐ์
- จำนวนยูนิตของปัญญาประดิษฐ์ในรัศมี 2000 รอบ Commander ฝ่ายศัตรู
- จำนวนยูนิตของปัญญาประดิษฐ์ในรัศมี 1000 รอบ Commander ฝ่ายศัตรู
- จำนวนยูนิตของปัญญาประดิษฐ์ในรัศมี 500 รอบ Commander ฝ่ายศัตรู
- เวลาที่เกิดการเปลี่ยนเฟส

โดยการเก็บข้อมูลของเกมจะเก็บในทุกๆ เวลา 3 วินาที (90 game cycles) ซึ่งตรงกับการทำงานของปัญญาประดิษฐ์ E323

3.1.3 การหาค่าน้ำหนักของยูนิต (w_u) โดยใช้การเรียนรู้แบบ Temporal Difference จากแผนที่ทดสอบ

ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1 การใช้การเรียนรู้แบบ Temporal Difference หาค่าน้ำหนักของยูนิต (w_i)

w_a, w_b, w_c, \dots แทนค่าน้ำหนักของยูนิตชนิด a, ค่าน้ำหนักของยูนิตชนิด b, ค่าน้ำหนักของยูนิตชนิด c, ... โดยกำหนดให้ค่าน้ำหนักเริ่มต้นก่อนการเรียนรู้มีค่าเท่ากับ 1

จากสมการที่ (2.2) เมื่อแทน $\nabla w P_k$ ด้วยสมการที่ (2.9) จะได้

$$\Delta w_i = \alpha (P_{t+1} - P_t) \sum_{k=1}^t \lambda^{t-k} (C_{u_0(t=k)} - C_{u_1(t=k)}) P_k (1 - P_k) \quad (3.1)$$

ให้ $C_{u_0(t=k)} - C_{u_1(t=k)}$ แทนด้วย C_k จะได้สมการที่ (3.2)

$$\Delta w_i = \alpha (P_{t+1} - P_t) \sum_{k=1}^t \lambda^{t-k} C_k P_k (1 - P_k) \quad (3.2)$$

รูปที่ 3.1 แสดงการใช้การเรียนรู้แบบ Temporal Difference ในการหาค่าน้ำหนักของยูนิตแต่ละชนิด เช่น ถ้าต้องการหาค่าน้ำหนักของยูนิตชนิด a (w_a) ที่เปลี่ยนแปลงที่เวลา 3 (t_3) ในเกมที่ 1 จะหาได้จากผลต่างของ Temporally successive prediction ที่เวลา 4 (P_4) กับ Temporally successive prediction ที่เวลา 3 (P_3) คูณด้วยอัตราการเรียนรู้ (α) คูณ

ด้วยผลรวมของเกรเดียนของ P_k เทียบกับ w ที่มีการให้ความสำคัญของครั้งที่อยู่ใกล้มากกว่าครั้งที่อยู่ไกลด้วย λ โดยเมื่อแทนค่า $t=3$ ในสมการที่ (3.2) จะได้สมการที่ (3.3)

$$\Delta W_{a(t=3)} = \alpha(P_4 - P_3) \{(\lambda^2 C_1 P_1 (1 - P_1)) + (\lambda^1 C_2 P_2 (1 - P_2)) + (\lambda^0 C_3 P_3 (1 - P_3))\} \quad (3.3)$$

เมื่อได้ค่านำหนักของยูนิตแต่ละชนิดหลังจบเกมที่ 1 จะใช้ค่านำหนักของยูนิตแต่ละชนิดนั้นเป็นค่านำหนักเริ่มต้นของยูนิตแต่ละชนิดในเกมต่อไป ทำเช่นนี้จนกระทั่งครบทุกเกม จะได้ค่านำหนักของยูนิตแต่ละชนิดที่เมื่อเอาไปใช้ในฟังก์ชันวัดคะแนนที่เวลาใดแล้วฟังก์ชันวัดคะแนนสามารถทำนายผลลัพธ์ได้ใกล้เคียงกับฟังก์ชันวัดคะแนนที่เวลาท้ายเกม

แต่ละคู่ปัญญาประดิษฐ์และแต่ละแผนที่จะมีค่านำหนักของยูนิต (w_u) ต่างกัน วิทยานิพนธ์นี้จะเก็บข้อมูลที่ใช้ในการหาค่านำหนักของยูนิต (w_u) โดยใช้จำนวนเกมที่เก็บข้อมูลดังตารางที่ 3.1

ตารางที่ 3.1 จำนวนเกมที่ใช้การเรียนรู้เพื่อหาค่านำหนักของยูนิต (w_u) และค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) ในแต่ละแผนที่ทดสอบ

คู่ปัญญาประดิษฐ์ที่สร้างขึ้น-คู่ปัญญาประดิษฐ์ทดสอบ / แผนที่	แผนที่ 1 (Small Divide)	แผนที่ 2 (Barren)	แผนที่ 3 (Tombstone Desert V2)
ปัญญาประดิษฐ์ที่สร้างขึ้น vs E323	E323 vs E323 ในแผนที่ 1 จำนวน 700 เกม	E323 vs E323 ในแผนที่ 2 จำนวน 700 เกม	E323 vs E323 ในแผนที่ 3 จำนวน 700 เกม
ปัญญาประดิษฐ์ที่สร้างขึ้น vs AAI	E323 vs AAI ในแผนที่ 1 จำนวน 700 เกม	E323 vs AAI ในแผนที่ 2 จำนวน 700 เกม	E323 vs AAI ในแผนที่ 3 จำนวน 700 เกม
ปัญญาประดิษฐ์ที่สร้างขึ้น vs RAI	E323 vs RAI ในแผนที่ 1 จำนวน 700 เกม	E323 vs RAI ในแผนที่ 2 จำนวน 700 เกม	E323 vs RAI ในแผนที่ 3 จำนวน 700 เกม

3.1.4 การหาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) จากแผนที่ทดสอบ

ในวิทยานิพนธ์นี้จะแบ่งเฟสของเกมตามการทดลองของ Bakkes เนื่องจากในผลการทดลองของ Bakkes นั้นแสดงให้เห็นว่าประสิทธิภาพของฟังก์ชันวัตถุประสงค์นั้นขึ้นอยู่กับเฟสของเกม โดยแบ่งเป็น 5 เฟส คือ

- เฟส 1 ตรวจสอบสิ่งก่อสร้างระดับ 1
- เฟส 2 ตรวจสอบยูนิตรระดับ 1 ที่ใช้เวลาในการสร้างมากกว่าหรือเท่ากับ 2500
- เฟส 3 ตรวจสอบสิ่งก่อสร้างระดับ 2
- เฟส 4 ตรวจสอบยูนิตรระดับ 2 ที่ใช้เวลาในการสร้างมากกว่าหรือเท่ากับ 15,000
- เฟส 5 ตรวจสอบยูนิตรหรือสิ่งก่อสร้างระดับ 3

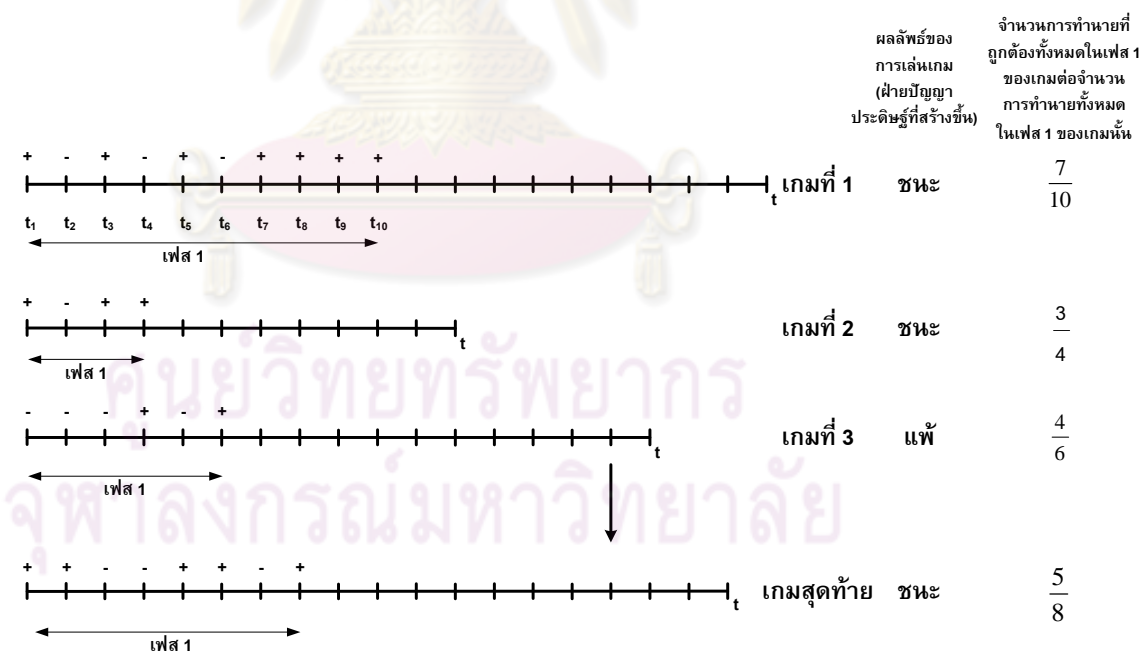
ซึ่งในแต่ละเฟสจะมีค่า w_p ของตัวเอง ซึ่งค่า w_p นี้ใช้เป็นส่วนประกอบของฟังก์ชันวัตถุประสงค์ สามารถหาค่า w_p ของแต่ละเฟสได้โดย

1. แทนค่า w_p เริ่มตั้งแต่ 0 โดยค่าจะเพิ่มได้สูงสุดถึง 1
2. หาค่าฟังก์ชันวัตถุประสงค์ในแต่ละช่วงเวลาที่เก็บข้อมูลของเกม โดยใช้ w_p จากข้อ 1
3. หาค่าจำนวนการทำนายที่ถูกต้องทุกช่วงเวลาในแต่ละเฟสของแต่ละเกมต่อจำนวนการทำนายทุกช่วงเวลาในแต่ละเฟสของเกมนั้น โดยเปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ในแต่ละช่วงเวลาที่เก็บข้อมูลกับผลลัพธ์ของการเล่นเกมนั้น
 - ถ้าฟังก์ชันวัตถุประสงค์นั้นมีค่าเป็นบวกและผลลัพธ์ของการเล่นเกมนั้นปัญญาประดิษฐ์ที่สร้างขึ้นชนะฝ่ายศัตรู นับว่า เป็นการทำนายที่ถูกต้อง
 - ถ้าฟังก์ชันวัตถุประสงค์นั้นมีค่าเป็นลบและผลลัพธ์ของการเล่นเกมนั้นปัญญาประดิษฐ์ที่สร้างขึ้นแพ้ฝ่ายศัตรู นับว่า เป็นการทำนายที่ถูกต้อง
4. รวมค่าจำนวนการทำนายที่ถูกต้องทั้งหมดในแต่ละเฟสของเกมต่อจำนวนการทำนายทั้งหมดในแต่ละเฟสของเกมนั้น ของทุกเกมทดสอบเข้าด้วยกัน
5. เพิ่ม w_p ขึ้นทีละ 0.01 แล้ว ทำซ้ำข้อ 2-4 จนค่า w_p มีค่า 1
6. เลือกค่า w_p ที่จะใช้โดยเปรียบเทียบค่าผลรวมของค่าจำนวนการทำนายที่ถูกต้องทั้งหมดในแต่ละเฟสของเกมต่อจำนวนการทำนายทั้งหมดในแต่ละเฟสในข้อ 4 โดยเลือกค่า w_p ที่ทำให้ค่าผลรวมของค่าจำนวนการทำนายที่ถูกต้องทั้งหมดในแต่ละเฟสของเกมต่อจำนวนการทำนายทั้งหมดในแต่ละเฟสมีค่าสูงสุด

7. ทำขั้นตอน 1-6 จนครบทั้ง 5 เฟส

แต่ละคู่ปัญญาประดิษฐ์และแต่ละแผนที่จะมีค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) ต่างกัน วิทยานิพนธ์นี้จะเก็บข้อมูลที่ใช้ในการหาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) โดยใช้จำนวนเกมที่เก็บข้อมูล ดังตารางที่ 3.1

จากรูปที่ 3.2 แสดงการหาผลรวมของค่าจำนวนการทำนายที่ถูกต้องทั้งหมดในเฟส 1 ของเกมต่อจำนวนการทำนายทั้งหมดในเฟส 1 ของเกมนั้น โดยหลังแทนค่า w_p แล้วได้ฟังก์ชันวัตถุประสงค์ในแต่ละช่วงเวลาที่เก็บข้อมูลของเกม ให้เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ในแต่ละช่วงเวลาที่เก็บข้อมูลกับผลลัพธ์ของการเล่นเกมจากเกมที่ 1 ผลลัพธ์ของการเล่นเกมฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นขณะ ดังนั้นจะนับฟังก์ชันวัตถุประสงค์ในแต่ละช่วงเวลาที่เก็บข้อมูลของเกมที่มีค่าเป็นบวกว่าเป็นการทำนายที่ถูกต้อง ตัวอย่างเช่น เฟส 1 ของเกมที่ 1 มีการทำนายที่ถูกต้องทั้งหมด 7 การทำนาย จากการทำนายทั้งหมด 10 การทำนาย แล้วจึงรวมค่าจำนวนการทำนายที่ถูกต้องทั้งหมดในเฟส 1 ต่อจำนวนการทำนายทั้งหมดในเฟส 1 ของทุกเกมเข้าด้วยกัน $\left(\frac{7}{10} + \frac{3}{4} + \frac{4}{6} + \dots + \frac{5}{8}\right)$ เพื่อใช้เปรียบเทียบว่าจะเลือก w_p ตัวไหนมาใช้ต่อไป



รูปที่ 3.2 การหาค่าตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) จากแผนที่ทดสอบ

3.1.5 การนำฟังก์ชันวัดคะแนนมาปรับใช้ในการทดลอง

ในวิทยานิพนธ์นี้จะแทนค่าเศษส่วนของสภาพแวดล้อมเป็น 1 เนื่องจากจะทดสอบการปรับความยากของเกมวางแผนการรบแบบตอบสนองทันกาลในสภาพแวดล้อมข้อมูลสมบูรณ์ จะได้ฟังก์ชันวัดคะแนน ดังสมการที่ (3.4)

$$v(p) = w_p \sum_u w_u (C_{u_0} - C_{u_1}) + (1 - w_p) \sum_{r \in D} w_r (O_{r_1} - O_{r_0}) \quad (3.4)$$

3.1.6 การนำเทคนิคการเรียนรู้แบบไดนามิกสคริปต์มาใช้ในการเรียนรู้

3.1.6.1 ออกแบบกฎย่อยในกฎพื้นฐานของเกม

ออกแบบกฎย่อยในกฎพื้นฐานของเกม โดยการใช้การผลิตยูนิตเป็นกฎย่อยในกฎพื้นฐาน ซึ่งได้แสดงรายละเอียดของยูนิตที่ใช้สร้างกฎย่อย รายละเอียดของกฎย่อยในกฎพื้นฐานของแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-คู่ปัญญาประดิษฐ์ทดสอบ และค่าน้ำหนักเริ่มต้นในภาคผนวก ข

3.1.6.2 ออกแบบฟังก์ชันวัดความเหมาะสม (Fitness Function)

เนื่องจากฟังก์ชันวัดความเหมาะสมเป็นฟังก์ชันที่ใช้วัดผลการทำงานของกฎที่ไดนามิกสคริปต์เลือกมาใช้ในครั้งนั้นว่ามีประสิทธิภาพเพียงใด จึงได้ออกแบบฟังก์ชันวัดความเหมาะสมโดยใช้ฟังก์ชันวัดคะแนนซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ ณ เวลานั้นเป็นส่วนประกอบ โดยให้ฟังก์ชันวัดความเหมาะสมที่เวลาใดเวลาหนึ่ง คือ ผลต่างของฟังก์ชันวัดคะแนนที่เวลาปัจจุบันกับฟังก์ชันวัดคะแนนที่เวลาก่อนหน้า แล้วนำมาใส่ฟังก์ชันซิกมอยด์เพื่อปรับเรนจ์ของข้อมูลให้อยู่ระหว่าง 0 ถึง 1 เพื่อให้เหมาะสมกับการนำมาคำนวณฟังก์ชันปรับค่าน้ำหนักต่อไป ดังสมการที่ (3.5)

$$F_t = \frac{1}{1 + e^{-(V_t - V_{t-1})}} \quad (3.5)$$

โดย

F_t คือ ฟังก์ชันวัดความเหมาะสมที่เวลา t

V_t คือ ฟังก์ชันวัดคะแนนที่เวลา t

V_{t-1} คือ ฟังก์ชันวัดคะแนนที่เวลา $t-1$

t คือ เวลาที่เทคนิคการเรียนรู้แบบไดนามิกสคริปต์ทำงาน โดยกำหนดให้เวลา t มีค่าเท่ากับ 30 วินาที

3.1.6.3 ออกแบบฟังก์ชันปรับค่าน้ำหนัก (Weight Update Functions)

ฟังก์ชันปรับค่าน้ำหนักจะทำหน้าที่เปลี่ยนค่าที่ได้จากฟังก์ชันวัดความเหมาะสม ให้เป็นการเปลี่ยนแปลงของค่าน้ำหนัก เพื่อที่จะเพิ่มน้ำหนักให้กฎที่มีประสิทธิภาพดี และ ลดค่าน้ำหนักให้กฎที่มีประสิทธิภาพไม่ดี ในวิทยานิพนธ์นี้จะใช้ ฟังก์ชันปรับค่าน้ำหนักตามที่ Spronck ใช้ในเกมแบบสวมทบาท [13] ดังสมการที่ (3.6)

$$\Delta W = \begin{cases} -P_{max} \cdot \frac{b-F}{b} & \{F < b\} \\ R_{max} \cdot \frac{F-b}{1-b} & \{F \geq b\} \end{cases} \quad (3.6)$$

โดย

ΔW คือ ค่าน้ำหนักที่เปลี่ยนแปลง

P_{max} คือ ค่าโทษสูงสุด (Maximum Penalty)

R_{max} คือ ค่ารางวัลสูงสุด (Maximum Reward)

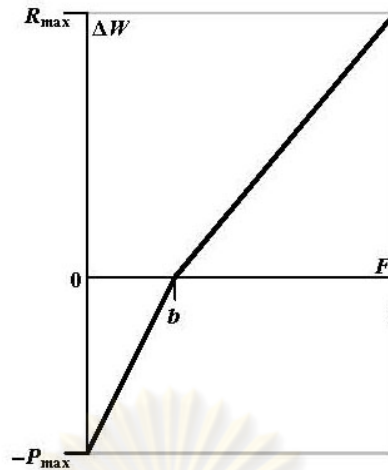
F คือ ฟังก์ชันวัดความเหมาะสม (Fitness Function) มีค่าอยู่ตั้งแต่ 0 ถึง

1

b คือ ค่ากลางระหว่างความเหมาะสมที่ได้รับรางวัลและความเหมาะสมที่
 โดนลงโทษ (break-even value)

เมื่อนำสมการ มาสร้างเป็นกราฟจะได้ดังรูปที่ 3.3

ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.3 กราฟของค่าน้ำหนักที่เปลี่ยนแปลง [13]

จะเห็นว่า

- ถ้าฟังก์ชันวัดความเหมาะสมให้ค่าเป็นบวก แสดงว่ากฎที่ไดนามิกสคริปต์เลือกมาใช้ในครั้งนั้นมีประสิทธิภาพดี เพราะทำให้ฟังก์ชันวัดคะแนนที่เวลา t มีค่ามากกว่าฟังก์ชันวัดคะแนนที่เวลา $t-1$ ฟังก์ชันปรับค่าน้ำหนักจะเพิ่มค่าน้ำหนักให้กับกฎที่ไดนามิกสคริปต์เลือกมาใช้ในครั้งนั้น ทำให้กฎเหล่านั้นมีความน่าจะเป็นในการถูกเลือกมาใช้ในเวลา que ไดนามิกสคริปต์ทำการประมวลผลถัดไป ($t+1$) เพิ่มขึ้น
- ถ้าฟังก์ชันวัดความเหมาะสมให้ค่าเป็นลบ แสดงว่ากฎที่ไดนามิกสคริปต์เลือกมาใช้ในครั้งนั้นมีประสิทธิภาพไม่ดี เพราะทำให้ฟังก์ชันวัดคะแนนที่เวลา t มีค่าน้อยกว่าฟังก์ชันวัดคะแนนที่เวลา $t-1$ ฟังก์ชันปรับค่าน้ำหนักจะลดค่าน้ำหนักให้กับกฎที่ไดนามิกสคริปต์เลือกมาใช้ในครั้งนั้น ทำให้กฎเหล่านั้นมีความน่าจะเป็นในการถูกเลือกมาใช้ในเวลา que ไดนามิกสคริปต์ทำการประมวลผลถัดไป ($t+1$) ลดลง

3.1.7 ปรับระดับความยากของเกมให้เข้ากับผู้เล่นโดยใช้วิธีการตัดค่าน้ำหนักที่สูง (Top Culling)

เป็นการนำวิธีการตัดค่าน้ำหนักที่สูงแบบเดิมของ Spronck มาปรับให้สามารถปรับความยากในขณะที่เกมกำลังดำเนินอยู่ได้ โดยวิธีการตัดค่าน้ำหนักที่สูงจะทำการปรับระดับของค่าน้ำหนักสูงสุด (w_{max}) แบบอัตราคงที่ (w_{dec} , w_{inc}) ซึ่งวิธีการตัดค่าน้ำหนักที่สูงแบบเดิมของ Spronck จะใช้ผลการปะทะในครั้งก่อนมาตัดสินใจว่าจะปรับค่าน้ำหนักสูงสุดขึ้นหรือลง แต่เราจะ

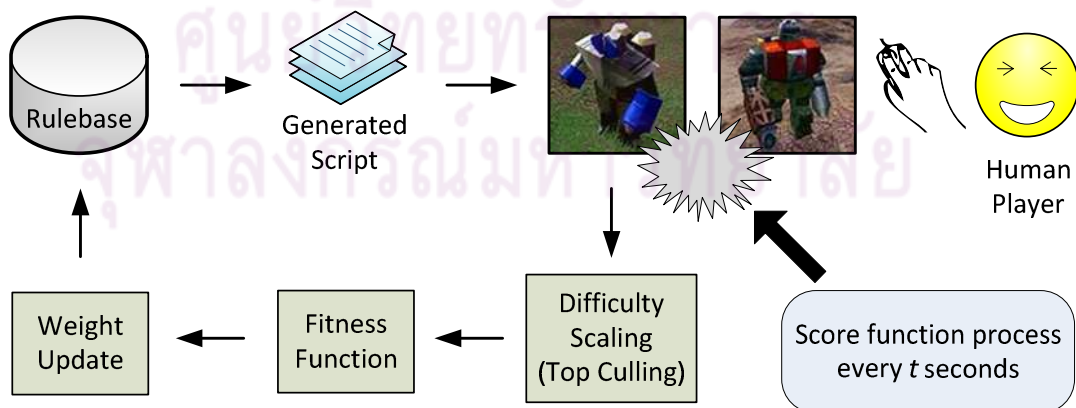
ใช้ฟังก์ชันวัตถุประสงค์แนะนำเป็นตัวตัดสินแทนเพื่อให้สามารถปรับระดับความยากในขณะเกมกำลังดำเนินอยู่ได้ โดย

- ถ้าฟังก์ชันคแนะนำให้ผลเป็น บวก จะปรับค่าน้ำหนักสูงสุดลง w_{dec} % ของค่าน้ำหนักสูงสุด
- ถ้าฟังก์ชันคแนะนำให้ผลเป็น ลบ จะปรับค่าน้ำหนักสูงสุดขึ้น w_{inc} % ของค่าน้ำหนักสูงสุด

3.1.8 การเก็บค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานในแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบ

มีจุดประสงค์เพื่อให้ค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานเป็นค่าที่เหมาะสมสำหรับทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสามารถสู้กับปัญญาประดิษฐ์ทดสอบได้อย่างสูสีโดยไม่ต้องเสียเวลาในการเรียนรู้ สามารถทำได้โดยการให้ปัญญาประดิษฐ์ที่สร้างขึ้นสู้กับปัญญาประดิษฐ์ทดสอบทั้งสามตัวที่แทนผู้เล่นแต่ละคน โดยเริ่มจากการให้ทุกกฎในกฎพื้นฐานของแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบมีค่าน้ำหนักเท่ากัน ปล่อยให้เกมดำเนินไปจนจบ นำค่าน้ำหนักของกฎในกฎพื้นฐานหลังจากจบเกมของแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบ ไปเป็นค่าตั้งต้นของกฎในกฎพื้นฐานของแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบคู่หนึ่งเมื่อเริ่มต้นเกมถัดไป ในวิทยานิพนธ์นี้จะใช้เกมสำหรับปรับค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานในแต่ละคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์ทดสอบ คู่ละประมาณ 50 เกม

ขั้นตอนการทำงานของปัญญาประดิษฐ์ที่สร้างขึ้นแสดงดังรูปที่ 3.4 โดยที่จะจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ โดยฝังงานได้แสดงไว้ในภาคผนวก ก



รูปที่ 3.4 การทำงานของไดนามิกสคริปต์และการปรับความยากเข้ากับผู้เล่นโดยวิธีการตัดค่าน้ำหนักที่สูง

3.2 การปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรในการปรับความยากของเกมวางแผนการรบแบบตอบสนองทันที

ไดนามิกสคริปต์แบบเดิมนั้น ได้จำกัดจำนวนกฎของชุดสคริปต์ไว้ ซึ่งบางครั้งพบว่าเกิดปัญหาขึ้นเมื่อมีการใช้กฎที่เป็นการผลิตยูนิต เช่น ถ้าจำกัดจำนวนกฎของชุดสคริปต์ไว้จำนวนน้อยเกินไป อาจทำให้ช่วงหลังของเกม ปัญญาประดิษฐ์ที่สร้างขึ้นไม่สามารถสู้กับศัตรูได้ ตามปกติแล้วเมื่อปัญญาประดิษฐ์ที่สร้างขึ้นพบว่าฝ่ายศัตรูเก่งกว่า ปัญญาประดิษฐ์ที่สร้างขึ้นจะใช้วิธีการตัดค่าน้ำหนักที่สูงในการปรับค่า w_{max} ให้สูงขึ้น เพื่อให้กฎที่มีประสิทธิภาพดีสามารถถูกเลือกมาใช้ได้ แต่บางเกมแม้ว่าปัญญาประดิษฐ์ที่สร้างขึ้นจะเลือกกฎที่มีประสิทธิภาพดีมาใช้แล้ว ก็ยังไม่สามารถสู้กับศัตรูที่มีจำนวนมากกว่าได้อยู่ดี ในขณะเดียวกัน ถ้าจำกัดจำนวนกฎของชุดสคริปต์ไว้จำนวนมากเกินไป อาจทำให้ช่วงต้นของเกม ปัญญาประดิษฐ์ที่สร้างขึ้นเอาชนะศัตรูได้อย่างรวดเร็ว ตามปกติเมื่อปัญญาประดิษฐ์ที่สร้างขึ้นพบว่าฝ่ายศัตรูอ่อนกว่า ปัญญาประดิษฐ์ที่สร้างขึ้นจะใช้วิธีการตัดค่าน้ำหนักที่สูงในการปรับค่า w_{max} ให้ต่ำลง เพื่อให้กฎที่มีประสิทธิภาพดีไม่สามารถถูกเลือกมาใช้ได้ แต่บางเกมแม้ว่าปัญญาประดิษฐ์ที่สร้างขึ้นจะเลือกกฎที่มีประสิทธิภาพไม่ดีมาใช้แล้ว ในชุดสคริปต์ยังประกอบด้วยกฎที่มีประสิทธิภาพไม่ดีจำนวนมาก ซึ่งเมื่อรวมกันแล้วทำให้ผลผลิตยูนิตออกมาจำนวนมากจนศัตรูไม่สามารถต้านทานได้

จากปัญหาดังกล่าวนี้ ทางผู้วิจัยได้ทดลองปรับค่าน้ำหนักของกฎพื้นฐานเพิ่มเติม พบว่าปัญญาประดิษฐ์ที่สร้างขึ้นยังไม่สามารถปรับตัวได้ดีขึ้นกว่าเดิม เนื่องจากขั้นตอนการเลือกกฎมาใช้เป็นชุดสคริปต์นั้น มีขั้นตอนการสุ่มค่าน้ำหนักเพื่อใช้เลือกกฎ ซึ่งประสิทธิภาพของกฎจะแปรผันตรงกับโอกาสในการถูกเลือกมาใช้ แต่เนื่องจากเป็นการสุ่มเลือกค่าน้ำหนัก กฎที่ถูกเลือกมาใช้ในครั้งนั้นอาจไม่เหมาะสมเสมอไป

ดังนั้นผู้วิจัยจึงเสนอวิธีการปรับปรุงไดนามิกสคริปต์โดยการปรับจำนวนของกฎในชุดสคริปต์ให้เหมาะสม โดยเสนอวิธีการเพิ่มจำนวนกฎในชุดสคริปต์ทีละกฎทุก ๆ ช่วงเวลาหนึ่ง และการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรู

3.2.1 การเพิ่มจำนวนกฎในชุดสคริปต์ทีละกฎทุก ๆ เวลาหนึ่ง

เมื่อพิจารณาจากวิธีการเล่นเกมวางแผนการรบแบบตอบสนองทันที ในช่วงแรกของเกมจะเป็นช่วงของการสร้างโรงงานสร้างทรัพยากร โรงงานสร้างยูนิตโจมตีขั้นแรกที่สามารถผลิตยูนิตโจมตีขั้นพื้นฐาน และจำนวนของโรงงานสร้างยูนิตโจมตีขั้นแรกมีจำนวนน้อย เมื่อเวลาผ่านไปช่วงกลางเกมและท้ายเกม แต่ละฝ่ายจะมียูนิตโจมตีที่เก่งขึ้น และสามารถสร้างได้ทีละหลายตัวเนื่องจากมีโรงงานสร้างยูนิตโจมตีจำนวนมากขึ้น ทางผู้วิจัยจึงเสนอวิธีการเพิ่ม

จำนวนกฎในชุดสคริปต์ที่ละกฎทุก ๆ เวลาหนึ่ง ซึ่งทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสามารถปรับตัวเข้ากับจำนวนยูนิตของศัตรูที่เพิ่มขึ้นตามเวลาได้ โดยฝั่งงานของการเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก ๆ เวลาหนึ่ง (ในวิทยานิพนธ์นี้เลือกเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก ๆ 7 นาที) ได้แสดงไว้ในภาคผนวก ก

3.2.2 การปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรู

สืบเนื่องจากวิธีการเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก ๆ เวลาหนึ่งนั้น ในบางครั้งหลังจากการปะทะกันของสองทีม ปัญญาประดิษฐ์ที่สร้างขึ้นสามารถทำลายโรงงานสร้างยูนิตของศัตรูได้ ทำให้อัตราการผลิตยูนิตของฝ่ายศัตรูลดลง แต่วิธีการเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก ๆ เวลาหนึ่ง ยังเพิ่มกฎเข้าไปในชุดสคริปต์อยู่ ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นไม่สามารถใช้เพียงแค่วิธีตัดค่าน้ำหนักที่สูง ในการปรับลดความยากลงให้เหมาะสมกับฝ่ายศัตรูได้ ทางผู้วิจัยจึงเสนอวิธีการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรู เพื่อให้จำนวนกฎในชุดสคริปต์เหมาะสมกับระดับของศัตรูขณะนั้นด้วย โดยมีวิธีการ คือ

- ปรับจำนวนกฎในชุดสคริปต์ขึ้นที่ละกฎ เมื่อฟังก์ชันวัดคะแนนขณะนั้นมีค่าเป็นลบ ซึ่งแสดงถึงขณะนั้นปัญญาประดิษฐ์ที่สร้างขึ้นอ่อนกว่าฝ่ายศัตรู ดังนั้นการปรับจำนวนกฎในชุดสคริปต์ขึ้นที่ละกฎจะช่วยให้ฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นสามารถเลือกกฎเพื่อมาสร้างเป็นชุดสคริปต์ได้จำนวนมากขึ้น ซึ่งส่งผลให้จำนวนการผลิตยูนิตของฝ่ายปัญญาประดิษฐ์สูงขึ้น
- ปรับจำนวนกฎในชุดสคริปต์ลงที่ละกฎ เมื่อฟังก์ชันวัดคะแนนขณะนั้นมีค่าเป็นบวก ซึ่งแสดงถึงขณะนั้นปัญญาประดิษฐ์ที่สร้างขึ้นเก่งกว่าฝ่ายศัตรู ดังนั้นการปรับจำนวนกฎในชุดสคริปต์ลงที่ละกฎจะช่วยให้ฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นสามารถเลือกกฎเพื่อมาสร้างเป็นชุดสคริปต์ได้จำนวนน้อยลง ซึ่งส่งผลให้จำนวนการผลิตยูนิตของฝ่ายปัญญาประดิษฐ์น้อยลง

โดยฝั่งงานของการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูได้แสดงไว้ใน

ภาคผนวก ก

3.2.3 วิธีการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัดคะแนนซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัดคะแนน ณ เวลาก่อนหน้า

อีกปัจจัยที่มีผลต่อการปรับระดับของปัญญาประดิษฐ์ด้วยไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง คือ ค่าน้ำหนักสูงสุด (w_{max}) ซึ่งเป็นค่าที่บ่งบอกถึงระดับสูงสุดของน้ำหนักของกฎที่สามารถเลือกมาใช้ในชุดสคริปต์ได้ เสมือนเป็นค่าซึ่งบ่งบอกถึงระดับของ

ปัญญาประดิษฐ์นั่นเอง ปัญหาหนึ่งของวิธีการตัดค่าน้ำหนักที่สูงแบบเดิมนั้นคือ ได้กำหนดค่า w_{max} ไว้ให้สามารถปรับค่าได้ w_{dec} เเปอร์เซ็นต์ และ w_{inc} เเปอร์เซ็นต์เมื่อเทียบกับค่า w_{max} ซึ่งค่า w_{dec} , w_{inc} นั้นคงที่ ซึ่งการปรับค่าแบบนี้บางครั้งถ้าจำนวนยูนิตในเกมเกิดการเปลี่ยนแปลงจำนวนมากในครั้งเดียว จะต้องใช้วิธีการตัดค่าน้ำหนักที่สูงหลายครั้ง จึงจะสามารถปรับระดับของปัญญาประดิษฐ์ที่สร้างขึ้นให้เหมาะสมกับศัตรูได้ ผู้วิจัยจึงเสนอการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า เพื่อดูแนวโน้มของการเปลี่ยนแปลงของค่าฟังก์ชันวัตถุประสงค์เพื่อให้สามารถปรับค่า w_{max} ให้เหมาะสมโดยใช้จำนวนการปรับน้อยครั้ง โดยมีวิธีการปรับคือ

ณ เวลาครั้งแรกที่ไดนามิกสคริปต์ทำงาน

- ถ้าฟังก์ชันวัตถุประสงค์เป็นบวก จะปรับค่า w_{max} ลง w_{dec} เเปอร์เซ็นต์ของ w_{max}
- ถ้าฟังก์ชันวัตถุประสงค์เป็นลบ จะปรับค่า w_{max} ขึ้น w_{inc} เเปอร์เซ็นต์ของ w_{max}

ณ เวลาใดๆ ที่ไดนามิกสคริปต์ทำงาน

- ถ้าฟังก์ชันวัตถุประสงค์เป็นบวก และ ถ้าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้าเป็นบวก ให้เพิ่มค่า w_{dec} อีก 10 เเปอร์เซ็นต์ของ w_{dec} ณ เวลาก่อนหน้า แล้วจึงปรับค่า w_{max}
- ถ้าฟังก์ชันวัตถุประสงค์เป็นบวก และ ถ้าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้าเป็นลบ ให้ปรับค่า w_{dec} เป็นค่า w_{dec} เริ่มต้น แล้วจึงปรับค่า w_{max}
- ถ้าฟังก์ชันวัตถุประสงค์เป็นลบ และ ถ้าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้าเป็นลบ ให้เพิ่มค่า w_{inc} อีก 10 เเปอร์เซ็นต์ของ w_{inc} ณ เวลาก่อนหน้า แล้วจึงปรับค่า w_{max}
- ถ้าฟังก์ชันวัตถุประสงค์เป็นบวก และ ถ้าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้าเป็นลบ ให้ปรับค่า w_{inc} เป็นค่า w_{inc} เริ่มต้น แล้วจึงปรับค่า w_{max}

โดยฝั่งงานของวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ ณ เวลานั้นกับเวลาก่อนหน้าได้แสดงไว้ในภาคผนวก ก

บทที่ 4

การทดลองและวิเคราะห์ผล

สำหรับวิทยานิพนธ์นี้ เพื่อที่จะสามารถประเมินคุณภาพและประสิทธิภาพของวิธีที่นำเสนอได้อย่างครบถ้วน จะต้องแบ่งการทดลองออกเป็นสองส่วนหลัก ๆ ด้วยกัน คือ

1. ทดสอบว่าการผลิตยูนิทใช้ในการปรับความยากให้สัมพันธ์กับระดับของผู้เล่นได้หรือไม่เพียงใด
- และ 2. ทดสอบประสิทธิภาพของการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรในการปรับความยากง่ายของเกมวางแผนการรบแบบตอบสนองทันกาล

4.1 การวัดประสิทธิภาพของปัญญาประดิษฐ์ที่สร้างขึ้น

วิทยานิพนธ์นี้แบ่งการวัดประสิทธิภาพออกเป็น 2 แบบ คือ

4.1.1 เวลาเฉลี่ย

ค่าเฉลี่ยของเวลาที่ใช้ในการเล่นเกมนับตั้งแต่ต้นเกมจนจบเกมของปัญญาประดิษฐ์แต่ละคู่ทดสอบจะถูกวัด ในวิทยานิพนธ์นี้จะเปรียบเทียบเวลาเฉลี่ยของเกมที่ทำให้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักคู่ ซึ่งเปรียบเสมือนผู้เล่นที่มีความสามารถเท่าเทียมกันสักคู่ กับเวลาเฉลี่ยของเกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสักคู่กับปัญญาประดิษฐ์ทดสอบตัวนั้น ถ้าเวลาเฉลี่ยของเกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสักคู่กับปัญญาประดิษฐ์ทดสอบตัวนั้นสูงกว่า แสดงว่าปัญญาประดิษฐ์ที่สร้างขึ้นนั้นสามารถเล่นกับปัญญาประดิษฐ์ทดสอบตัวนั้น ได้ดีกว่า เพราะสามารถรักษาสภาพไม่ชนะและไม่แพ้ได้นาน

4.1.2 ผลการ ชนะ/แพ้ ทำเกม

ผล ชนะ/แพ้ รวมทั้งหมดทุกเกมของแต่ละคู่ทดสอบจะถูกบันทึกไว้ ซึ่งถ้าผลการ ชนะ/แพ้ ทำเกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสักคู่กับปัญญาประดิษฐ์ทดสอบตัวนั้น มีค่าเข้าใกล้ 50 เปอร์เซ็นต์ จะถือว่า ปัญญาประดิษฐ์ที่สร้างขึ้นนั้นสามารถเล่นได้ดีกับปัญญาประดิษฐ์ทดสอบตัวนั้น เพราะจำนวนครั้งในการชนะ/แพ้ออกมาเท่าๆกัน

4.2 การเตรียมการทดลอง

4.2.1 ปัญญาประดิษฐ์ทดสอบ

เป็นปัญญาประดิษฐ์ที่ใช้คู่กับปัญญาประดิษฐ์ที่นำเสนอในวิทยานิพนธ์นี้ ซึ่งวิทยานิพนธ์นี้ใช้ปัญญาประดิษฐ์ทดสอบ จำนวน 3 ตัว คือ

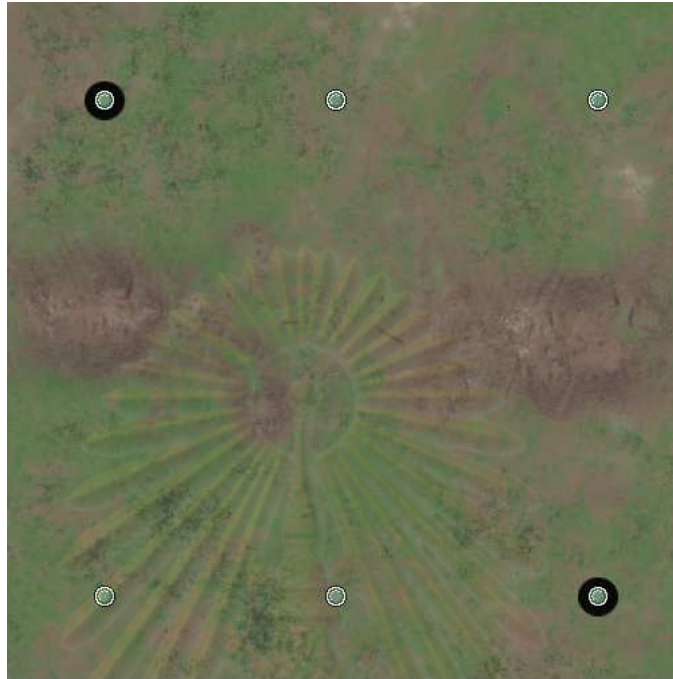
- E323 เป็นปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้นโดย F. Huizinga และ slogic มีความสามารถในการควบคุมทั้งยูนิตแต่ละตัวและกลุ่มของยูนิตได้ดีมาก สามารถควบคุมยูนิตเข้าโจมตีและตั้งรับได้อย่างมีประสิทธิภาพ สามารถเก็บทรัพยากรได้อย่างรวดเร็ว บริหารการใช้ทรัพยากรได้อย่างมีประสิทธิภาพ และมีรูปแบบการเล่นในการบุกโจมตีอย่างรวดเร็วตั้งแต่ต้นเกม (Rush)
- AAI เป็นปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้นโดย Alexander Seizinger สามารถปรับแต่งค่าที่ใช้ควบคุมปัญญาประดิษฐ์ได้ มีความสามารถในการควบคุมกลุ่มของยูนิตได้ดี สามารถบริหารการใช้ทรัพยากรได้ดี และสามารถปรับพฤติกรรมตามศัตรูได้ (ในวิทยานิพนธ์นี้จะปิดความสามารถในการปรับตัวตามศัตรู)
- RAI เป็นปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้นโดย Reth มีความสามารถในการควบคุมทั้งยูนิตแต่ละตัวและกลุ่มของยูนิตได้ดี สร้างสิ่งก่อสร้างรวดเร็ว และมีความสามารถในการตั้งฐานรอง

4.2.2 แผนที่ทดสอบ

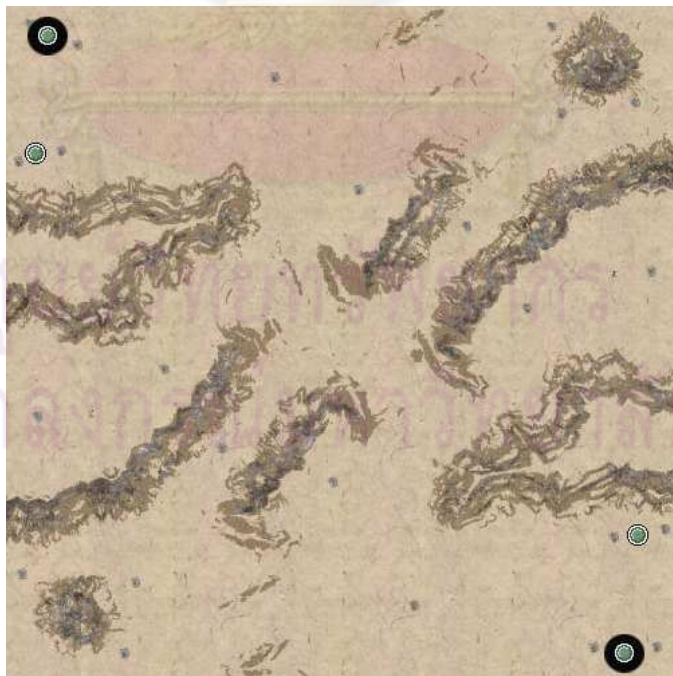
ในวิทยานิพนธ์นี้ใช้แผนที่ทดสอบ จำนวน 3 แผนที่ เนื่องจากต้องการทดสอบว่าปัญญาประดิษฐ์ที่สร้างขึ้นสามารถปรับความยากให้เข้ากับผู้เล่นได้แม้ว่าภูมิภาคในแต่ละแผนที่จะต่างกัน โดยแต่ละแผนที่มีลักษณะภูมิภาคแตกต่างกัน คือ

- แผนที่ 1 (SmallDivide) เป็นแผนที่ขนาด 8x8 รองรับจำนวนผู้เล่นสูงสุด 6 คน ลักษณะของแผนที่จะมีเนินเขาตรงกลางแผนที่ดังรูปที่ 4.1 โดยที่ยูนิตสามารถเดินข้ามไปได้แต่ใช้เวลาในการเดินมากกว่าในการเดินบนพื้นธรรมดา กำหนดให้จุดเกิดของทั้งสองฝั่งอยู่ที่มุมซ้ายบนและมุมขวาล่าง
- แผนที่ 2 (Barren) เป็นแผนที่ขนาด 8x8 รองรับจำนวนผู้เล่นสูงสุด 4 คน ลักษณะของแผนที่จะมีภูเขาหลายลูกกั้นระหว่างสองฝ่ายดังรูปที่ 4.2 โดยที่ยูนิตสามารถเดินข้ามไปได้แต่ใช้เวลาในการเดินมากกว่าในการเดินบนพื้นธรรมดา กำหนดให้จุดเกิดของทั้งสองฝั่งอยู่ที่มุมซ้ายบนและมุมขวาล่าง
- แผนที่ 3 (Tomb Stone Desert V2) เป็นแผนที่ขนาด 8x8 รองรับจำนวนผู้เล่นสูงสุด 10 คน ลักษณะของแผนที่จะมีภูเขากระจายอยู่ทั่วแผนที่ดังรูปที่ 4.3 โดยที่ยูนิตสามารถเดินข้ามไปได้แต่ใช้เวลาในการเดินมากกว่าในการเดินบนพื้นธรรมดา มีความแตกต่างจากแผนที่ 2 ตรงที่ยูนิตสามารถเดินอ้อมภูเขาได้สอง

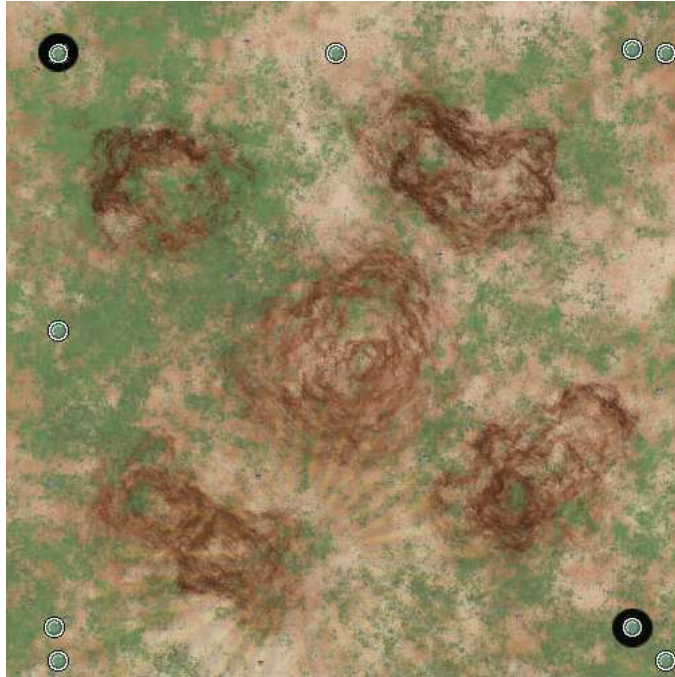
ทางเพื่อที่จะไปโจมตีอีกฝ่าย กำหนดให้จุดเกิดของทั้งสองฝั่งอยู่ที่มุมซ้ายบน และมุมขวาล่าง



รูปที่ 4.1 แผนที่ 1 (SmallDivide)



รูปที่ 4.2 แผนที่ 2 (Barren)



รูปที่ 4.3 แผนที่ 3 (Tomb Stone Desert V2)

4.2.3 ปัญญาประดิษฐ์ที่สร้างขึ้น

ปัญญาประดิษฐ์ที่สร้างขึ้นจะมี 5 แบบ คือ

- ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) ซึ่งขั้นตอนการทำงานได้แสดงไว้ในบทที่ 3 หัวข้อ 3.1.6 - 3.1.7
- ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์แทนซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า (w_{inc} , w_{dec}) ซึ่งขั้นตอนการทำงานได้แสดงไว้ในบทที่ 3 หัวข้อ 3.2.3
- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ทีละกฎทุกๆ ช่วงเวลาหนึ่ง และวิธีการตัดค่าน้ำหนักที่สูง ในการทดลองจะเลือกเพิ่มกฎในชุดสคริปต์ทีละกฎทุกๆ 7 นาที (7-minute increment) ซึ่งขั้นตอนการทำงานได้แสดงไว้ในบทที่ 3 หัวข้อ 3.2.1
- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) ซึ่งขั้นตอนการทำงานได้แสดงไว้ในบทที่ 3 หัวข้อ 3.2.2

- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดย การตรวจสอบค่าฟังก์ชันวัตถุประสงค์ซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า (Mix) วิธีนี้เป็นการผสมวิธี (Change w_{inc} , w_{dec}) กับ (Change number of rules) เข้าด้วยกัน

4.2.4 การเก็บผลการทดลอง

การทดลองแต่ละคู่ทดสอบ จะทดสอบเป็นจำนวน 200 เกม และนำมาวิเคราะห์ ผลหาค่าต่างๆ ดังนี้

- เวลาเฉลี่ยของทั้ง 200 เกม (avg)
- เกมที่ใช้เวลาน้อยสุด (min)
- เกมที่ใช้เวลามากสุด (max)
- เวลาเฉลี่ยของเกมทีชนะ (avg win)
- เวลาเฉลี่ยของเกมทีแพ้ (avg lose)
- จำนวนเกมที่ ชนะ/แพ้ (games)
- เปอร์เซนต์ของจำนวนเกมที่ ชนะ/แพ้ (%)

4.3 ผลการทดลอง

Methods	P.E.		Time (s)				Win/Lose	
	avg	min	max	avg win	avg lose	games	%	
Fix 1 rule	1868.64	183	6780	1305.47	2133.66	64/136	32/68	
Change w_{inc} , w_{dec}	1958.03	207	10818	1125.97	3609.72	133/67	66.5/33.5	
7-minute increment	1974.47	216	10452	1283.85	3077.65	123/7	61.1/38.5	
Change number of rules	2923.28	309	13464	1427.69	4074.74	87/113	43.5/56.5	
Mix	2382.06	234	12102	1245.94	3139.48	80/120	40/60	

รูปที่ 4.4 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide)

Methods	P.E.		Time (s)				Win/Lose	
	avg	min	max	avg win	avg lose	games	%	
Fix 1 rule	2493.89	243	8676	1616.27	2926.14	66/134	33/67	
Change w_{inc} , w_{dec}	1215.00	291	7914	856.08	3727.44	175/25	87.5/12.5	
7-minute increment	2627.60	261	9705	1790.27	3993.75	124/76	62/38	
Change number of rules	3573.78	381	18333	1767.51	4855.15	83/117	41.5/58.5	
Mix	3365.00	357	14004	1691.09	4390.94	76/124	38/62	

รูปที่ 4.5 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		1545.32	195	5484	1452.14	1556.83	22/178	11/89
Change w_{inc}, w_{dec}		1709.81	204	5550	1128.04	2105.80	81/119	40.5/59.5
7-minute increment		1887.65	201	6126	1368.18	2137.76	65/135	32.5/67.5
Change number of rules		2114.52	252	8292	1209.82	2933.06	95/105	47.5/52.5
Mix		1919.49	210	6630	1336.67	2187.50	63/137	31.5/68.5

รูปที่ 4.6 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		2349.41	354	8613	2598.00	2255.11	55/145	27.5/72.5
Change w_{inc}, w_{dec}		2593.04	456	9996	2291.62	3205.00	134/66	67/33
7-minute increment		2546.76	630	10143	2324.53	3008.31	135/65	67.5/32.5
Change number of rules		2867.46	579	14046	2601.23	3266.81	120/80	60/40
Mix		2690.48	573	11043	2495.24	3086.86	134/66	67/33

รูปที่ 4.7 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		3832.50	774	12612	3939.70	3463.27	155/45	77.5/22.5
Change w_{inc}, w_{dec}		3817.85	816	18249	4239.65	2482.13	152/48	76/24
7-minute increment		3746.79	882	18606	3649.98	5410.09	189/11	94.5/5.5
Change number of rules		4750.25	885	34890	5578.04	3675.07	113/87	56.5/43.5
Mix		3964.50	840	25626	4081.74	3430.42	164/36	82/18

รูปที่ 4.8 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		4720.32	477	33366	4462.95	4822.91	57/143	28.5/71.5
Change w_{inc}, w_{dec}		5537.07	699	41583	4671.95	6101.90	79/121	39.5/60.5
7-minute increment		5493.74	537	34551	4195.88	7361.38	118/82	59/41
Change number of rules		6672.87	765	49302	5308.51	8242.61	107/93	53.5/46.5
Mix		5467.19	663	40398	4220.79	6426.80	87/113	43.5/56.5

รูปที่ 4.9 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		1168.23	447	2562	1175.77	1163.41	78/122	39/61
Change w_{inc}, w_{dec}		1261.38	405	2625	1258.71	1265.07	116/84	58/42
7-minute increment		1271.09	471	2892	1318.83	1202.38	118/82	59/41
Change number of rules		1377.35	507	3588	1385.07	1370.07	97/103	48.5/51.5
Mix		1220.19	492	3177	1283.72	1171.27	87/113	43.5/56.5

รูปที่ 4.10 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		1505.72	435	3879	1599.13	1477.02	47/153	23.5/76.5
Change w_{inc}, w_{dec}		1693.85	459	4278	1950.30	1522.88	80/120	40/60
7-minute increment		1646.82	441	3987	1853.85	1323.00	122/78	61/39
Change number of rules		1844.84	666	4641	2045.19	1659.89	96/104	48/52
Mix		1475.67	513	3885	1780.17	1264.07	82/118	41/59

รูปที่ 4.11 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren)

Methods	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
Fix 1 rule		1254.56	489	3291	1519.71	1168.51	49/151	24.5/75.5
Change w_{inc}, w_{dec}		1462.46	516	3813	1676.79	1339.25	73/127	36.5/63.5
7-minute increment		1420.17	471	3678	1790.20	1216.51	71/129	35.5/64.5
Change number of rules		1605.80	579	3972	1676.23	1548.16	90/110	45/55
Mix		1272.50	534	3612	1637.40	1096.80	65/135	32.5/67.5

รูปที่ 4.12 ผลการทดลองคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2)

Types	P.E.	Time (s)				Win/Lose		
		avg	min	max	avg win	avg lose	games	%
E323 vs E323		741.47	147	4248	737.65	744.85	94/106	47/53
E323 vs AAI		718.76	435	1380	719.68	534.00	199/1	99.5/0.5
E323 vs RAI		653.55	375	1491	653.55	-	200/0	100/0
AAI vs AAI		1149.08	297	3534	1281.00	1084.10	66/134	33/67
AAI vs RAI		877.71	357	1587	751.71	898.22	28/172	14/86
RAI vs RAI		752.64	342	1263	738.08	771.17	112/88	56/44
RAI vs AAI		1064.81	351	2124	1146.49	964.97	110/90	55/45

รูปที่ 4.13 ผลการทดลองคู่ปัญญาประดิษฐ์ทดสอบ - คู่ปัญญาประดิษฐ์ทดสอบ ในแผนที่ 1 (SmallDivide)

Types	P.E.	Time (s)					Win/Lose	
		avg	min	max	avg win	avg lose	games	%
E323 vs E323		980.52	180	6603	988.72	976.92	61/139	30.5/69.5
E323 vs AAI		715.50	339	1488	715.50	-	200/0	100/0
E323 vs RAI		710.60	414	1896	710.60	-	200/0	100/0
AAI vs AAI		2028.24	315	4962	1935.91	2073.72	66/134	33/67
AAI vs RAI		1255.76	321	3426	1030.85	1436.08	89/111	44.5/55.5
RAI vs RAI		1188.83	492	3582	1287.04	1116.23	85/115	42.5/57.5
RAI vs AAI		1237.82	369	2793	1437.81	1109.95	78/122	39/61

รูปที่ 4.14 ผลการทดลองคู่ปัญญาประดิษฐ์ทดสอบ - คู่ปัญญาประดิษฐ์ทดสอบ ในแผนที่ 2 (Barren)

Types	P.E.	Time (s)					Win/Lose	
		avg	min	max	avg win	avg lose	games	%
E323 vs E323		1318.97	174	7290	1156.37	1491.62	103/97	51.5/48.5
E323 vs AAI		675.93	441	1278	675.93	-	200/0	100/0
E323 vs RAI		633.65	441	1386	633.65	-	200/0	100/0
AAI vs AAI		1529.00	288	3015	1374.15	1627.99	78/122	39/61
AAI vs RAI		1183.74	279	2652	1193.96	1159.90	140/60	70/30
RAI vs RAI		1051.97	510	2400	1061.94	1041.58	102/98	51/49
RAI vs AAI		1137.33	348	2949	1115.37	1146.08	57/143	28.5/71.5

รูปที่ 4.15 ผลการทดลองคู่ปัญญาประดิษฐ์ทดสอบ - คู่ปัญญาประดิษฐ์ทดสอบ ในแผนที่ 3 (Tomb Stone Desert V2)

4.4 ทดสอบว่าการผลิตยูนิทใช้ในการปรับความยากให้สูสีกับระดับของผู้เล่นได้หรือไม่เพียงใด

ในการทดสอบว่าการผลิตยูนิทใช้ในการปรับความยากให้สูสีกับระดับของผู้เล่นได้หรือไม่เพียงใด จะใช้ปัญญาประดิษฐ์ที่สร้างขึ้นที่ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์ และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) โดยจะเปรียบเทียบเวลาเฉลี่ยและผลการ ชนะ/แพ้ ทำเกมของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - ปัญญาประดิษฐ์ทดสอบ กับ ปัญญาประดิษฐ์ทดสอบ - ปัญญาประดิษฐ์ทดสอบ ซึ่งคู่ปัญญาประดิษฐ์ทดสอบ - ปัญญาประดิษฐ์ทดสอบนั้น เปรียบเสมือนผู้เล่นที่มีความสามารถเท่าเทียมกัน

4.4.1 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - E323

จากรูปที่ 4.13 E323 - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 741.47 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 47/53 เปอร์เซนต์ และจากรูปที่ 4.15 E323 - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1318.97 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 51.5/48.5 เปอร์เซนต์ ซึ่งสัดส่วนการแพ้ชนะแสดงถึงการ เล่นที่สูสีกันของทั้งสองฝ่ายซึ่งเป็นปัญญาประดิษฐ์ชนิดเดียวกัน แต่จากรูป 4.14 E323 - E323

เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 980.52 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 30.5/69.5 เปอร์เซ็นต์ ซึ่งสัดส่วนการแพ้ชนะที่ไม่เท่ากัน แม้จะใช้ปัญญาประดิษฐ์ ตัวเดียวกันสู้กันในแผนที่ 2 นี้ อาจมีสาเหตุมาจากสถานที่เริ่มต้นเกมของสองฝ่ายซึ่งแตกต่างกัน แหล่งทรัพยากรรอบๆ ที่ไม่เหมือนกัน ทำให้ปัญญาประดิษฐ์ทดสอบซึ่งเป็นปัญญาประดิษฐ์ที่มี ขั้นตอนการทำงานที่แน่นอน (Fixed Script) ไม่สามารถสู้กันได้อย่างสูสี

พิจารณารูปที่ 4.4 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 เมื่อ ปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1868.64 ซึ่งเป็น 2.52 เท่าของ เวลาเฉลี่ยของเมื่อ E323 - E323 สู้กันในแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 32/68 เปอร์เซ็นต์ ตามด้วย จากรูปที่ 4.5 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 เมื่อ ปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 2493.89 ซึ่งเป็น 2.54 เท่าของ เวลาเฉลี่ยของเมื่อ E323 - E323 สู้กันในแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 33/67 เปอร์เซ็นต์ ตามด้วย จากรูปที่ 4.6 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 เมื่อ ปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1545.32 ซึ่งเป็น 1.17 เท่าของ เวลาเฉลี่ยของเมื่อ E323 - E323 สู้กันในแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 11/89 เปอร์เซ็นต์

พบว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 นั้น มากกว่าเวลาเฉลี่ย E323 - E323 ในทุกแผนที่ทดสอบ แต่ผลการ ชนะ/แพ้ ท้ายเกม นั้น ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 ยังทำได้ด้อยกว่า E323 - E323 ค่อนข้างมาก ยกเว้นในแผนที่ 2 ซึ่งทำได้ดีกว่าเล็กน้อย ซึ่งแสดงถึงการผลิตยูนิตมีความสามารถในการปรับ ระดับความยากให้สูสีกับระดับผู้เล่นได้ในช่วงระหว่างการเล่นเพราะเวลาในการเล่นโดยเฉลี่ยมี ค่าสูงขึ้น แต่จากผลการ ชนะ/แพ้ ท้ายเกม นั้น สังเกตได้ว่าฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นมี จำนวนเกมที่ชนะน้อยกว่า E323 อย่างเห็นได้ชัด ซึ่งเนื่องมาจากการที่ไดนามิกสคริปต์จำกัด จำนวนกฎของชุดสคริปต์ไว้ 1 กฎ ทำให้เมื่อเกมดำเนินไปถึงช่วงกลางเกมและท้ายเกม ปัญญาประดิษฐ์ทดสอบจะสามารถสร้างยูนิตโจมตีที่เก่งขึ้น และสามารถสร้างได้ที่หลายตัว เนื่องจากมีโรงงานสร้างยูนิตโจมตีจำนวนมากขึ้น ทำให้แม้ว่าปัญญาประดิษฐ์ที่สร้างขึ้นจะ พยายามทำให้เกมสูสีด้วยการเลือกกฎที่มีประสิทธิภาพดีมาใช้แล้วก็ยังไม่สามารถสู้กับศัตรูที่มี จำนวนมากกว่าได้อยู่ดี ดังนั้น การปรับการผลิตยูนิต (ซึ่งในการทดลองนี้ยังจำกัดจำนวนกฎที่ เลือกใช้ได้ และยังคงใช้วิธีการตัดค่าน้ำหนักแบบปกติตามงานวิจัยของ Spronck) จึงสามารถ ปรับความยากให้สูสีระหว่างการเล่นได้ในช่วงเวลาจำกัดเท่านั้น

4.4.2 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - AAI

จากรูปที่ 4.13 AAI - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1149.08 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 33/67 เปอร์เซ็นต์ จากรูปที่ 4.14

AAI - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 2028.24 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 33/67 เปอร์เซนต์ และจากรูปที่ 4.15 AAI - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1529 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 39/61 เปอร์เซนต์ ซึ่งสรุปได้เหมือนกรณีแรกคือแม้เราจะใช้ปัญญาประดิษฐ์ตัวเดียวกันคู่กันผล ชนะ/แพ้ ทำเกม อาจไม่ใกล้เคียงกัน ซึ่งอาจมีผลมาจากสาเหตุดังที่กล่าวไว้ในหัวข้อ 4.4.1

พิจารณาจากรูปที่ 4.7 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2349.41 ซึ่งเป็น 2.04 เท่าของเวลาเฉลี่ยของเมื่อ AAI - AAI สู้กันในเกมที่ 1 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 27.5/72.5 เปอร์เซนต์ ตามด้วย จากรูปที่ 4.8 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3832.5 ซึ่งเป็น 1.89 เท่าของเวลาเฉลี่ยของเมื่อ AAI - AAI สู้กันในเกมที่ 2 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 77.5/22.5 เปอร์เซนต์ ตามด้วย จากรูปที่ 4.9 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 4720.32 ซึ่งเป็น 3.09 เท่าของเวลาเฉลี่ยของเมื่อ AAI - AAI สู้กันในเกมที่ 3 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 28.5/71.5 เปอร์เซนต์

พบว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI นั้นมากกว่าเวลาเฉลี่ย AAI - AAI ในทุกเกมที่ทดสอบ แต่ผลการ ชนะ / แพ้ ทำเกมนั้นปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI ยังทำได้ด้อยกว่า AAI - AAI ซึ่งแสดงถึงการผลิตยูนิตมีความสามารถในการปรับระดับความยากให้สู้กับระดับผู้เล่นได้ในระหว่างการเล่นจริงเพราะเวลาเฉลี่ยมีค่าสูงขึ้น แต่จากผล ชนะ/แพ้ ทำเกมนั้น สังเกตได้ว่าในเกมที่ 1, 3 ฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นมีจำนวนเกมที่ชนะน้อยกว่า AAI คล้ายกับผลที่ได้ในหัวข้อ 4.4.1 นั้นคือการผลิตยูนิตใช้ปรับให้เกมสูสีได้ในช่วงเวลาจำกัด แต่ในเกมที่ 2 ฝ่ายปัญญาประดิษฐ์ที่สร้างขึ้นมีจำนวนเกมที่ชนะมากกว่า AAI ซึ่งเป็นผลมาจากการที่ไดนามิกสคริปต์จำกัดจำนวนกฎของชุดสคริปต์ไว้ 1 กฎ ถ้าจำกัดจำนวนเกมที่ใช้เรียนรู้ค่าน้ำหนักของกฎย่อยในกฎพื้นฐานของการทดลองให้ใกล้เคียงกัน (ประมาณ 50 เกม) ในบางครั้งค่าน้ำหนักของกฎย่อยในกฎพื้นฐานหลังจากการเรียนรู้นั้นอาจจะยังไม่เหมาะสม ทำให้เมื่อนำมาใช้ในไดนามิกสคริปต์แบบจำกัดจำนวนกฎของชุดสคริปต์ไว้ 1 กฎแล้ว แม้ว่าปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) จะพยายามปรับความยากให้สูสีกับ AAI แล้ว ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) ก็ยังชนะ AAI ได้อย่างรวดเร็วอยู่ดี

4.4.3 กรณีปัญญาประดิษฐ์ที่สร้างขึ้น - RAI

จากรูปที่ 4.13 RAI - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 752.64 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 56/44 เปอร์เซนต์ จากรูปที่ 4.14 RAI - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1188.83 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 42.5/57.5 เปอร์เซนต์ และจากรูปที่ 4.15 RAI - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1051.97 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 51/49 เปอร์เซนต์ ซึ่งสรุปได้เหมือนกรณีแรกคือแม้เราจะใช้ปัญญาประดิษฐ์ตัวเดียวกันสู้กันผล ชนะ/แพ้ ทำายเกม อาจไม่ใกล้เคียงกัน ซึ่งอาจมีผลมาจากสาเหตุดังที่กล่าวไว้ในหัวข้อ 4.4.1

จากรูปที่ 4.10 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1168.23 ซึ่งเป็น 1.55 เท่าของ เวลาเฉลี่ยของเมื่อ RAI - RAI สู้กันในแผนที่ 1 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 39/61 เปอร์เซนต์ ตามด้วย จากรูปที่ 4.11 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1505.72 ซึ่งเป็น 1.27 เท่าของ เวลาเฉลี่ยของเมื่อ RAI - RAI สู้กันในแผนที่ 2 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 23.5/76.5 เปอร์เซนต์ ตามด้วย จากรูปที่ 4.12 ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1254.56 ซึ่งเป็น 1.19 เท่าของ เวลาเฉลี่ยของเมื่อ RAI - RAI สู้กันในแผนที่ 3 และ ผลการ ชนะ/แพ้ ทำายเกม มีค่า 24.5/75.5 เปอร์เซนต์

ซึ่งเมื่อพิจารณาแล้วจะพบว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI นั้นมากกว่าเวลาเฉลี่ย RAI - RAI ในทุกแผนที่ทดสอบ แต่ผลการ ชนะ / แพ้ ทำายเกม นั้นปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI ยังทำได้ด้อยกว่า RAI - RAI ค่อนข้างมาก ซึ่งแสดงว่าการผลิตยูนิตมีความสามารถในการปรับระดับความยากให้สูสีกับระดับผู้เล่นในระหว่างการเล่นได้เพราะเวลาเฉลี่ยมีค่าสูงขึ้น แต่จากผล ชนะ/แพ้ ทำายเกม นั้น คล้ายกับผลที่ได้ในหัวข้อ 4.4.1 นั่นคือการผลิตยูนิตใช้ปรับให้เกมสูสีได้ในช่วงเวลาจำกัดเท่านั้น

ต่อมาเมื่อพิจารณาจากรูปที่ 4.13 จะสังเกตได้ว่าในแผนที่ 1 นั้น E323 เป็นปัญญาประดิษฐ์ที่มีระดับความเก่งสูงที่สุด รองลงมาคือ RAI ตามด้วย AAI จากรูปที่ 4.14 จะสังเกตได้ว่าในแผนที่ 2 นั้น E323 เป็นปัญญาประดิษฐ์ที่มีระดับความเก่งสูงที่สุด ส่วน AAI และ RAI มีระดับความเก่งพอๆ กัน เนื่องจากเมื่อเราสลบจุดเกิดของ AAI กับ RAI แล้วทำให้ผลการ ชนะ/แพ้ ทำายเกม สลับกัน สุดท้ายพิจารณาจากรูปที่ 4.15 จะสังเกตได้ว่าในแผนที่ 1 นั้น E323 เป็นปัญญาประดิษฐ์ที่มีระดับความเก่งสูงที่สุด รองลงมาคือ AAI ตามด้วย RAI

พบว่าปัญญาประดิษฐ์ที่สร้างขึ้น สามารถสู้ได้กับทั้ง E323, AAI และ RAI โดยใช้การผลิตยูนิตเพื่อปรับให้เข้ากับปัญญาประดิษฐ์ทดสอบแต่ละตัวโดยไม่ต้องปรับพฤติกรรม

ของปัญญาประดิษฐ์ ใช้เพียงการปรับเปลี่ยนค่าน้ำหนักของยูนิตและค่าน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานตามปัญญาประดิษฐ์ทดสอบแต่ละตัวในแต่ละแผนที่เท่านั้น ซึ่งแสดงว่าการผลิตยูนิตใช้ในการปรับความยากให้สอดคล้องกับระดับของผู้เล่น (แทนด้วยปัญญาประดิษฐ์ทดสอบแต่ละตัว) ได้ดีระดับหนึ่ง

4.5 ทดสอบประสิทธิภาพของการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรในการปรับความยากง่ายของเกมวางแผนการรบแบบตอบสองทีนกาล

ในการทดสอบประสิทธิภาพของการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรจะเปรียบเทียบเวลาเฉลี่ยและผลการ ชนะ/แพ้ ทำเกมของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น - ปัญญาประดิษฐ์ทดสอบแต่ละแบบกับปัญญาประดิษฐ์ที่สร้างขึ้นที่ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงที่จำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) โดยถ้าเวลาเฉลี่ยสูงกว่าและผลการ ชนะ/แพ้ ทำเกมใกล้เคียงกันมากกว่า แสดงว่าปัญญาประดิษฐ์ที่ปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรตัวนั้น สามารถเล่นได้สูสีกว่านั่นเอง

4.5.1 ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลานั้นกับเวลา ก่อนหน้า (Change w_{inc} , w_{dec})

วิธีนี้ ยังคงใช้การเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูง เพียงแต่ไม่ใช่ค่า w_{inc} และ w_{dec} ที่คงที่ แต่จะปรับค่าตามค่าฟังก์ชันวัตถุประสงค์ ณ เวลาปัจจุบันกับเวลาก่อนหน้า ดังที่ได้กล่าวไว้ในหัวข้อที่ 4.2.3 จากรูปที่ 4.4 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1958.03 ซึ่งเป็น 1.05 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 66.5/33.5 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 32/68 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) เล็กน้อย ต่อมาจากรูปที่ 4.5 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1215 ซึ่งเป็น 0.49 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 87.5/12.5 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) แย่กว่า 33/67 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.6 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1709.81 ซึ่งเป็น 1.11 เท่าของเวลาเฉลี่ยของเมื่อ

ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 40.5/59.5 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 11/89 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก

จากรูปที่ 4.7 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2593.04 ซึ่งเป็น 1.1 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 67/33 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 27.5/72.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ต่อมาจากรูปที่ 4.8 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3817.85 ซึ่งเป็น 0.996 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 76/24 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 77.5/22.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) เล็กน้อย และจากรูปที่ 4.9 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 5537.07 ซึ่งเป็น 1.17 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 39.5/60.5 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 28.5/71.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule)

จากรูปที่ 4.10 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1261.38 ซึ่งเป็น 1.08 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 58/42 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 39/61 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) เล็กน้อย ต่อมาจากรูปที่ 4.11 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1693.85 ซึ่งเป็น 1.12 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 40/60 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 23.5/76.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.12 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1462.46 ซึ่งเป็น 1.17 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 36.5/63.5 เปอร์เซ็นต์ (Change w_{inc} , w_{dec}) ดีกว่า 24.5/75.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule)

เมื่อพิจารณาแล้วจะเห็นว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - ปัญญาประดิษฐ์ทดสอบ ส่วนมากจะดีกว่าเล็กน้อยหรือใกล้เคียงกับเวลาเมื่อใช้ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - ปัญญาประดิษฐ์ทดสอบ ยกเว้นการทดลองปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - E323 ในแผนที่ 2 ที่มีค่าเวลาเฉลี่ยน้อยกว่า

ส่วน ผลการ ชนะ/แพ้ ท้ายเกม จะเข้าใกล้ 50 เปอร์เซ็นต์มากกว่าเล็กน้อยยกเว้นการทดลองดังกล่าวเช่นกัน ซึ่งจากการดูไฟล์บันทึกการเล่นของปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) - E323 ในแผนที่ 2 นี้ พบว่าฝ่าย E323 นั้นไม่ได้สร้างฐานบริเวณจุดเกิดเหมือนในแผนที่ 1 และแผนที่ 3 แต่จะเดินไปสร้างฐานบริเวณกลางแผนที่ ซึ่งทำให้แพ้ปัญญาประดิษฐ์ที่สร้างขึ้น (Change w_{inc} , w_{dec}) อย่างรวดเร็ว จากการทดลองทั้งหมดในหัวข้อนี้สรุปได้ว่าการปรับค่า w_{inc} และ w_{dec} ระหว่างการเล่น (Change w_{inc} , w_{dec}) สามารถช่วยให้ปัญญาประดิษฐ์ที่สร้างขึ้นนั้นปรับความยากให้รู้สึกกับระดับของผู้เล่นได้ดีกว่าการใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์ และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) เล็กน้อย

4.5.2 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก 7 นาที และวิธีการตัดค่าน้ำหนักที่สูง (7-minute increment)

จากรูปที่ 4.4 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1974.47 ซึ่งเป็น 1.06 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 61.1/38.5 เปอร์เซ็นต์ (7-minute increment) ดีกว่า 32/68 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ต่อมาจากรูปที่ 4.5 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 2627.6 ซึ่งเป็น 1.05 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 62/38 เปอร์เซ็นต์ (7-minute increment) ดีกว่า 33/67 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.6 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1887.65 ซึ่งเป็น 1.22 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 32.5/67.5 เปอร์เซ็นต์ (7-minute increment) ดีกว่า 11/89 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ค่อนข้างมาก

จากรูปที่ 4.7 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2546.76 ซึ่งเป็น 1.08 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 67.5/32.5 เปอร์เซ็นต์ (7-minute increment) ดีกว่า 27.5/72.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ต่อมาจากรูปที่ 4.8 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3746.79 ซึ่งเป็น 0.98 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 94.5/5.5 เปอร์เซ็นต์ (7-minute increment) แย่

กว่า 77.5/22.5 เปอร์เซนต์ของเมื่อใช้วิธี (Fix 1 rule) ค่อนข้างมาก และจากรูปที่ 4.9 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 5493.74 ซึ่งเป็น 1.16 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 59/41 เปอร์เซนต์ (7-minute increment) ดีกว่า 28.5/71.5 เปอร์เซนต์ของเมื่อใช้วิธี (Fix 1 rule) ค่อนข้างมาก

จากรูปที่ 4.10 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1271.09 ซึ่งเป็น 1.09 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 59/41 เปอร์เซนต์ (7-minute increment) ดีกว่า 39/61 เปอร์เซนต์ของเมื่อใช้วิธี (Fix 1 rule) เล็กน้อย ต่อมาจากรูปที่ 4.11 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1646.82 ซึ่งเป็น 1.09 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 61/39 เปอร์เซนต์ (7-minute increment) ดีกว่า 23.5/76.5 เปอร์เซนต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.12 ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1420.17 ซึ่งเป็น 1.13 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 35.5/64.5 เปอร์เซนต์ (7-minute increment) ดีกว่า 24.5/75.5 เปอร์เซนต์ของเมื่อใช้วิธี (Fix 1 rule)

เมื่อพิจารณาแล้วจะเห็นว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - ปัญญาประดิษฐ์ทดสอบ ส่วนมากจะดีกว่าเล็กน้อยหรือแย่กว่าเล็กน้อยเมื่อเทียบกับปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - ปัญญาประดิษฐ์ทดสอบ ส่วน ผลการ ชนะ/แพ้ ท้ายเกม จะเข้าใกล้ 50 เปอร์เซนต์มากกว่าเล็กน้อยยกเว้นการทดลอง ปัญญาประดิษฐ์ที่สร้างขึ้น (7-minute increment) - AAI ในแผนที่ 2 ซึ่งอาจมีผลมาจาก เมื่อเกมดำเนินไปถึงช่วงหลังของเกม จำนวนกฎในชุดสคริปต์จะมีจำนวนเพิ่มขึ้น สืบเนื่องจากเวลาเฉลี่ยของเกมชนะคือ 3649.98 ซึ่งเมื่อคำนวณออกมาเป็นจำนวนกฎในชุดสคริปต์แล้ว จะพบว่ามีจำนวน 9 กฎ ทำให้ถึงแม้วิธีการตัดค่าน้ำหนักที่สูงจะป้องกันไม่ให้กฎที่มีประสิทธิภาพดีถูกเลือกแล้ว แต่ AAI ก็ยังไม่สามารถต้านทานการบุกของยูนิทที่สร้างกฎที่มีประสิทธิภาพไม่ดีจำนวนหลายกฎ สรุปได้ว่าการใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก 7 นาที และวิธีการตัดค่าน้ำหนักที่สูง (7-minute increment) นั้นสามารถช่วยให้ปัญญาประดิษฐ์ที่สร้างขึ้นนั้นสามารถปรับความยากให้สูสีกับระดับของผู้เล่นได้ดีกว่าการใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) เล็กน้อย แต่ก็ยังมีประเด็นที่ต้องปรับปรุงเรื่องจำนวนกฎ ที่เมื่อเกมดำเนินไปนานจะทำให้

จำนวนกฎสูง ทำให้วิธีการตัดค่าน้ำหนักที่สูงไม่สามารถปรับระดับความยากได้อย่างมีประสิทธิภาพ

เมื่อเปรียบเทียบการใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก 7 นาที และวิธีการตัดค่าน้ำหนักที่สูง (7-minute increment) กับ การใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์แทนซึ่งแสดงถึงระดับของปัญหาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า ($Change\ w_{inc}, w_{dec}$) จะพบว่าไม่มีประสิทธิภาพที่ใกล้เคียงกันแต่ทั้งสองแบบยังไม่มีประสิทธิภาพเท่าที่ควร

4.5.3 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules)

จากรูปที่ 4.4 ปัญหาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 เมื่อปัญหาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2923.28 ซึ่งเป็น 1.56 เท่าของเวลาเฉลี่ยของเมื่อปัญหาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 1 และ ผลการชนะ/แพ้ ทำเกม มีค่า 43.5/56.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 32/68 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ต่อมาจากรูปที่ 4.5 ปัญหาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 เมื่อปัญหาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3573.78 ซึ่งเป็น 1.43 เท่าของเวลาเฉลี่ยของเมื่อปัญหาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 41.5/58.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 33/67 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.6 ปัญหาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 เมื่อปัญหาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 2114.52 ซึ่งเป็น 1.37 เท่าของเวลาเฉลี่ยของเมื่อปัญหาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - E323 สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 47.5/52.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 11/89 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก

จากรูปที่ 4.7 ปัญหาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI เมื่อปัญหาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2867.46 ซึ่งเป็น 1.22 เท่าของเวลาเฉลี่ยของเมื่อปัญหาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 60/40 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 27.5/72.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก ต่อมาจากรูปที่ 4.8 ปัญหาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI เมื่อปัญหาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 4750.25 ซึ่งเป็น 1.24 เท่าของเวลาเฉลี่ยของเมื่อปัญหาประดิษฐ์ที่สร้างขึ้น (Fix 1

rule) - AAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 56.5/43.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 77.5/22.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก และจากรูปที่ 4.9 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 6672.87 ซึ่งเป็น 1.41 เท่าของ เวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - AAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 53.5/46.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 28.5/71.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก

จากรูปที่ 4.10 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1377.35 ซึ่งเป็น 1.18 เท่าของ เวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 1 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 48.5/51.5 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 39/61 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ต่อมาจากรูปที่ 4.11 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในแผนที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1844.84 ซึ่งเป็น 1.23 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 48/52 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 23.5/76.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก และจากรูปที่ 4.12 ปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันใน แผนที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1605.8 ซึ่งเป็น 1.28 เท่าของเวลาเฉลี่ยของเมื่อ ปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - RAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มี ค่า 45/55 เปอร์เซ็นต์ (Change number of rules) ดีกว่า 24.5/75.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) ก่อนข้างมาก

เมื่อพิจารณาแล้วจะเห็นว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - ปัญญาประดิษฐ์ทดสอบ ดีกว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Fix 1 rule) - ปัญญาประดิษฐ์ทดสอบมาก ส่วน ผลการ ชนะ/แพ้ ท้ายเกม จะเข้าใกล้ 50 เปอร์เซ็นต์ มากกว่าและดีกว่าทุกแบบของปัญญาประดิษฐ์ที่สร้างขึ้น ซึ่งมีผลมาจากจำนวนกฎในชุด สคริปต์จะเปลี่ยนแปลงตามระดับของศัตรูซึ่งสามารถแก้ไขข้อด้อยของการปรับปรุงเทคนิคการ เรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุก 7 นาที และวิธีการตัด ค่าน้ำหนักที่สูง (7-minute increment) ได้

4.5.4 ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับ จำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่า น้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของ

ปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ขณะนั้นกับเวลา ก่อนหน้า (Mix)

วิธีนี้จะรวมเอาการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ก่อนหน้า เข้ากับ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรู เพื่อดูผลว่าจะปรับระดับความยากได้ดีขึ้นกว่าใช้การปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูอย่างเดียวหรือไม่

จากรูปที่ 4.4 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2382.06 ซึ่งเป็น 0.81 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 สู้กันบนแผนที่ 1 และ ผลการชนะ/แพ้ ท้ายเกม มีค่า 40/60 เปอร์เซนต์ (Mix) แยกว่า 43.5/56.5 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules) เล็กน้อย ต่อมาจากรูปที่ 4.5 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3365 ซึ่งเป็น 0.94 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 38/62 เปอร์เซนต์ (Mix) แยกว่า 41.5/58.5 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules) เล็กน้อย และจากรูปที่ 4.6 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - E323 เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1919.49 ซึ่งเป็น 0.91 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - E323 สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 31.5/68.5 เปอร์เซนต์ (Mix) แยกว่า 47.5/52.5 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules)

จากรูปที่ 4.7 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 2690.48 ซึ่งเป็น 0.94 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI สู้กันบนแผนที่ 1 และ ผลการชนะ/แพ้ ท้ายเกม มีค่า 67/33 เปอร์เซนต์ (Mix) แยกว่า 60/40 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules) ต่อมาจากรูปที่ 4.8 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 3964.5 ซึ่งเป็น 0.83 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI สู้กันบนแผนที่ 2 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 82/18 เปอร์เซนต์ (Mix) แยกว่า 56.5/43.5 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules) ค่อนข้างมาก และจากรูปที่ 4.9 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - AAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในพื้นที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 5467.19 ซึ่งเป็น 0.82 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - AAI สู้กันบนแผนที่ 3 และ ผลการ ชนะ/แพ้ ท้ายเกม มีค่า 43.5/56.5 เปอร์เซนต์ (Mix) แยกว่า 53.5/46.5 เปอร์เซนต์ของเมื่อใช้วิธี (Change number of rules) เล็กน้อย

จากรูปที่ 4.10 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 1 จะเห็นว่า เวลาเฉลี่ยมีค่า 1220.19 ซึ่งเป็น 0.89 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI สู้กันบนเกมที่ 1 และ ผลการชนะ/แพ้ ทำเกม มีค่า 43.5/56.5 เปอร์เซ็นต์ (Mix) แยกว่า 48.5/51.5 เปอร์เซ็นต์ของเมื่อใช้วิธี (Change number of rules) ต่อมาจากรูปที่ 4.11 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 2 จะเห็นว่า เวลาเฉลี่ยมีค่า 1475.67 ซึ่งเป็น 0.80 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI สู้กันบนเกมที่ 2 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 41/59 เปอร์เซ็นต์ (Mix) แยกว่า 48/52 เปอร์เซ็นต์ของเมื่อใช้วิธี (Fix 1 rule) และจากรูปที่ 4.12 ปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - RAI เมื่อปัญญาประดิษฐ์คู่นี้สู้กันในเกมที่ 3 จะเห็นว่า เวลาเฉลี่ยมีค่า 1272.5 ซึ่งเป็น 0.79 เท่าของเวลาเฉลี่ยของเมื่อปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - RAI สู้กันบนเกมที่ 3 และ ผลการ ชนะ/แพ้ ทำเกม มีค่า 32.5/67.5 เปอร์เซ็นต์ (Mix) แยกว่า 45/55 เปอร์เซ็นต์ของเมื่อใช้วิธี (Change number of rules) ค่อนข้างมาก

เมื่อพิจารณาแล้วจะเห็นว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Mix) - ปัญญาประดิษฐ์ทดสอบ แยกว่าเวลาเฉลี่ยของปัญญาประดิษฐ์ที่สร้างขึ้น (Change number of rules) - ปัญญาประดิษฐ์ทดสอบ และ ผลการ ชนะ/แพ้ ทำเกม ยังห่างจาก 50 เปอร์เซ็นต์มากกว่า ซึ่งมีผลมาจากบางครั้งปัญญาประดิษฐ์ที่สร้างขึ้นดำเนินการปรับระดับจำนวนกฎในชุดสคริปต์แล้ว ยังปรับค่า w_{max} เข้าซ้อนจากวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ซึ่งแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์แผน ๓ เวลาก่อนหน้าอีก ทำให้การปรับความยากบางครั้งเกินหรือต่ำกว่าระดับของศัตรูไป

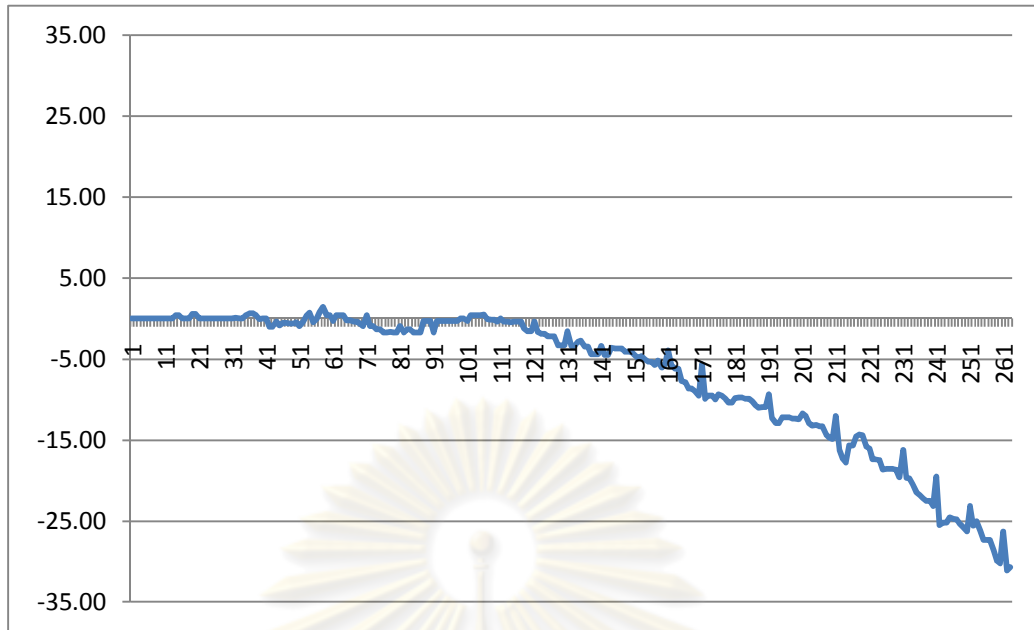
จากผลการทดลองทั้งหมดแสดงให้เห็นว่าการปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) นั้นเป็นวิธีที่สามารถปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การปรับอัตราการสร้างทรัพยากรได้ดีกว่าการใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) ซึ่งเป็นวิธีแบบดั้งเดิมของ Spronck ซึ่งไม่เหมาะสมกับการใช้กฎเมื่อกฎพื้นฐานเป็นการผลิตยูนิต

4.6 วิเคราะห์ฟังก์ชันวัตถุประสงค์และแผนกับผลการทดลอง

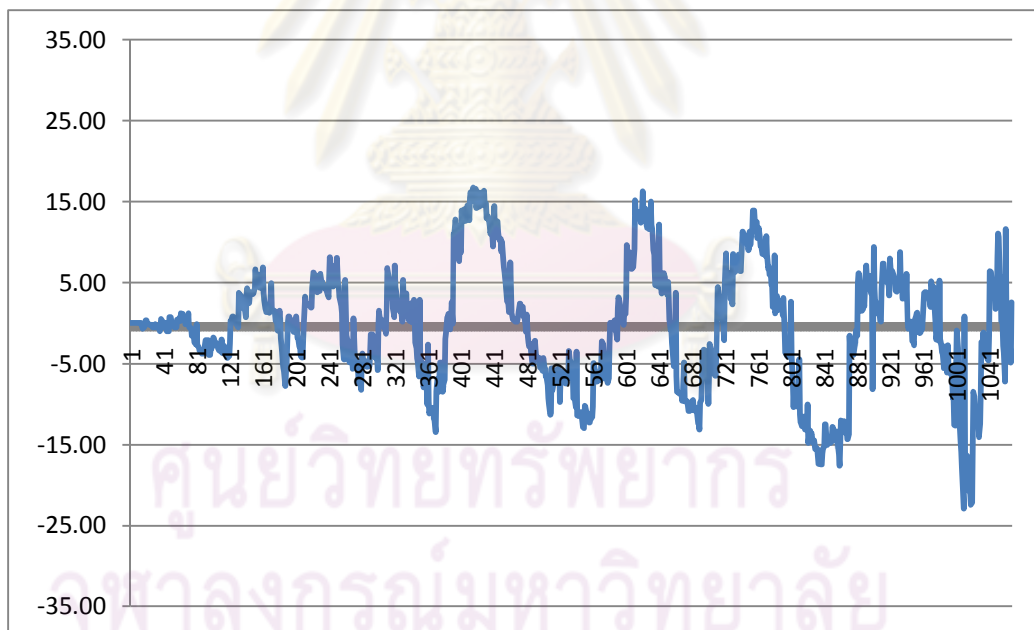
ในส่วนนี้จะเป็นการเป็นการเปรียบเทียบฟังก์ชันวัตถุประสงค์และแผนกับเวลาที่เก็บข้อมูลจากเกมตัวอย่างระหว่างเกมที่ทำให้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักในในแต่ละแผนที่กับเกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสู้กับปัญญาประดิษฐ์ทดสอบตัวนั้นตาม รูปที่ 4.16 - รูปที่ 4.33 โดยแกนนอนแสดงจุดที่เก็บข้อมูลของเกมซึ่งเก็บทุก ๆ 3 วินาที แกนตั้งแสดงฟังก์ชันวัตถุประสงค์และปัญญาประดิษฐ์ที่สร้างขึ้นจะใช้ปัญญาประดิษฐ์แบบใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) เป็นตัวแทนในการเปรียบเทียบกับปัญญาประดิษฐ์ทดสอบ

เมื่อเปรียบเทียบกราฟของฟังก์ชันวัตถุประสงค์และเวลาที่เก็บข้อมูลจากเกมตัวอย่างระหว่างเกมที่ทำให้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักในในแต่ละแผนที่กับเกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสู้กับปัญญาประดิษฐ์ทดสอบตัวนั้นจะเห็นว่าเกมที่ทำให้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักในเองนั้น ฟังก์ชันวัตถุประสงค์จะมีลักษณะที่เข้าใกล้ศูนย์ในช่วงแรกแต่ต่อมาฟังก์ชันวัตถุประสงค์จะมีค่าเพิ่มขึ้นหรือลดลงขึ้นกับว่าฝ่ายใดเป็นฝ่ายที่ชนะในเกมนั้น แสดงถึงว่าเมื่อเราใช้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักในนั้นเมื่อถึงเวลาหนึ่งปัญญาประดิษฐ์ทดสอบตัวหนึ่งจะเก่งมากกว่าปัญญาประดิษฐ์ทดสอบอีกตัวแล้วชนะไปในที่สุด แต่เกมที่ทำให้ปัญญาประดิษฐ์ที่สร้างขึ้นสู้กับปัญญาประดิษฐ์ทดสอบตัวนั้นจะเห็นว่ากราฟนั้นมีการปรับให้ค่าเข้าใกล้ศูนย์แสดงถึงเมื่อปัญญาประดิษฐ์ที่สร้างขึ้นเก่งกว่าปัญญาประดิษฐ์ทดสอบ (ฟังก์ชันวัตถุประสงค์มีค่ามากกว่าศูนย์) ปัญญาประดิษฐ์ที่สร้างขึ้นจะพยายามปรับตัวโดยลดการผลิตรายการลงเพื่อให้ฟังก์ชันวัตถุประสงค์มีค่าลดลง แต่ถ้าปัญญาประดิษฐ์ที่สร้างขึ้นอ่อนกว่าปัญญาประดิษฐ์ทดสอบ (ฟังก์ชันวัตถุประสงค์มีค่าน้อยกว่าศูนย์) ปัญญาประดิษฐ์ที่สร้างขึ้นจะพยายามปรับตัวโดยเพิ่มการผลิตรายการขึ้นเพื่อให้ฟังก์ชันวัตถุประสงค์มีค่าเพิ่มขึ้น เมื่อพิจารณาประกอบกันแล้วจะเห็นว่าปัญญาประดิษฐ์ที่สร้างขึ้นมีการปรับตัวให้เข้ากับฝ่ายตรงข้ามตลอดทั้งเกมแตกต่างกับปัญญาประดิษฐ์ปกติที่เมื่อถึงจุดหนึ่งฝ่ายใดฝ่ายหนึ่งจะมีความเก่งมากกว่าอีกฝ่ายไปตลอดเวลาที่เหลือของเกม

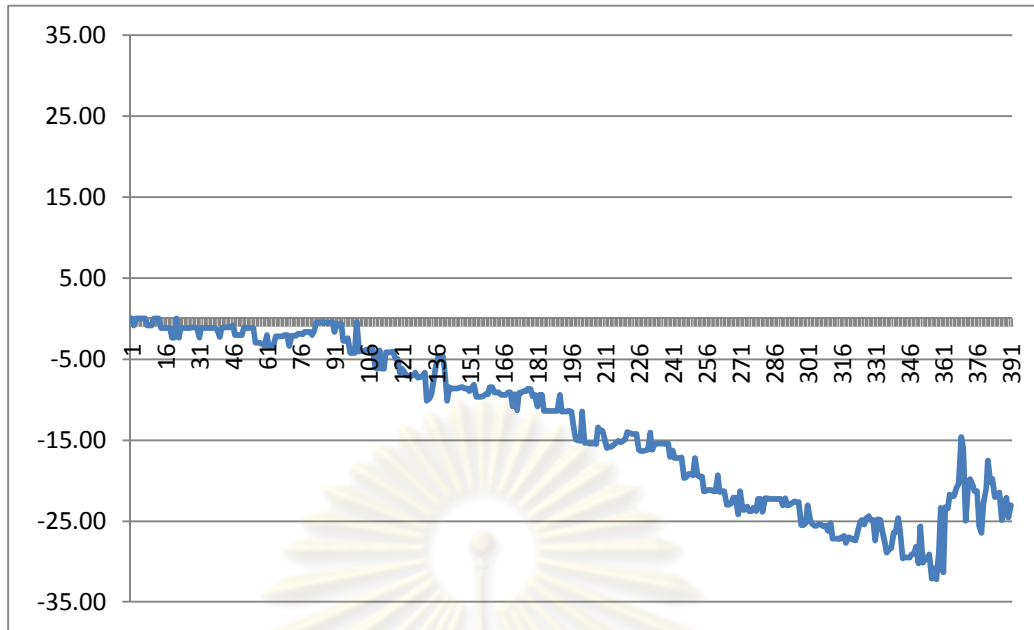
ศูนย์วิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



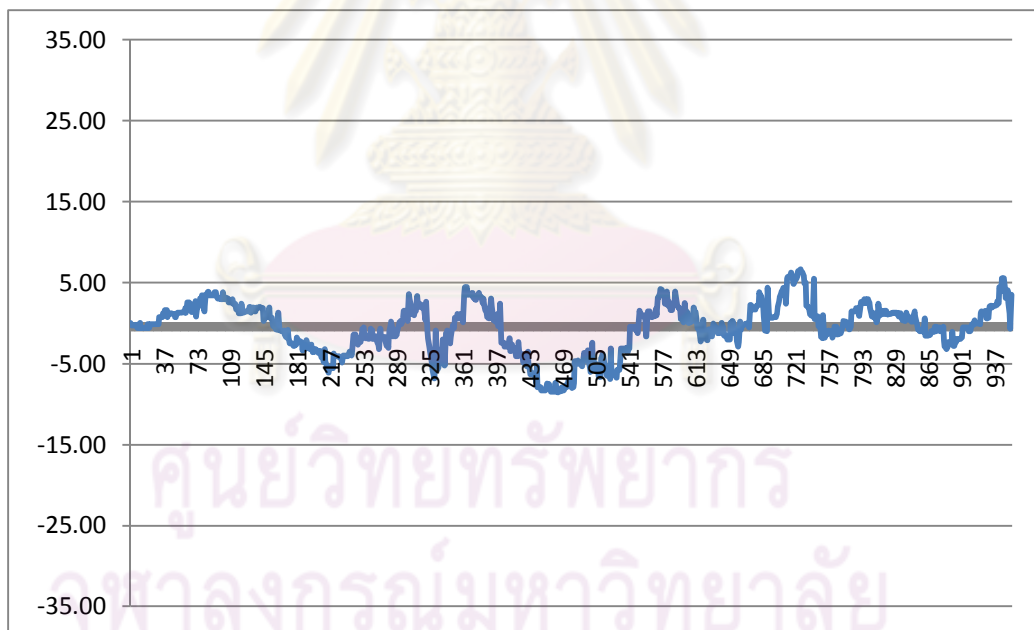
รูปที่ 4.16 ตัวอย่างกราฟฟังก์ชันวัตคະແນของ E323 - E323 ในແຜນທີ່ 1 (SmallDivide)



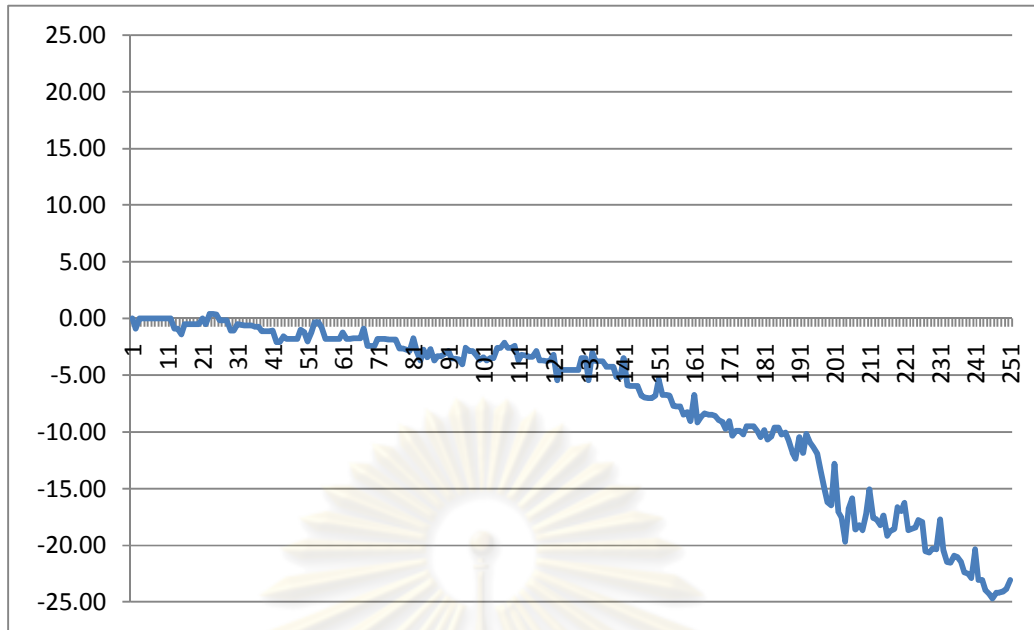
รูปที่ 4.17 ตัวอย่างกราฟฟังก์ชันวัตคະແນของປັດຊະຮູ້ທີ່ສ້າງຂຶ້ນ - E323 ในແຜນທີ່ 1 (SmallDivide)



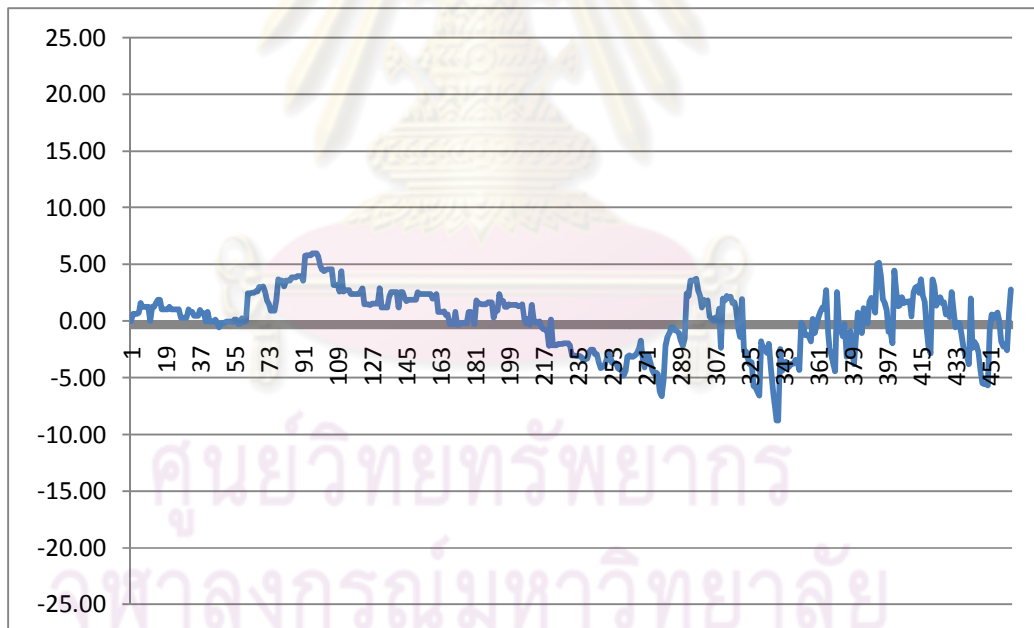
รูปที่ 4.18 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ AAI - AAI ในแผนที่ 1 (SmallDivide)



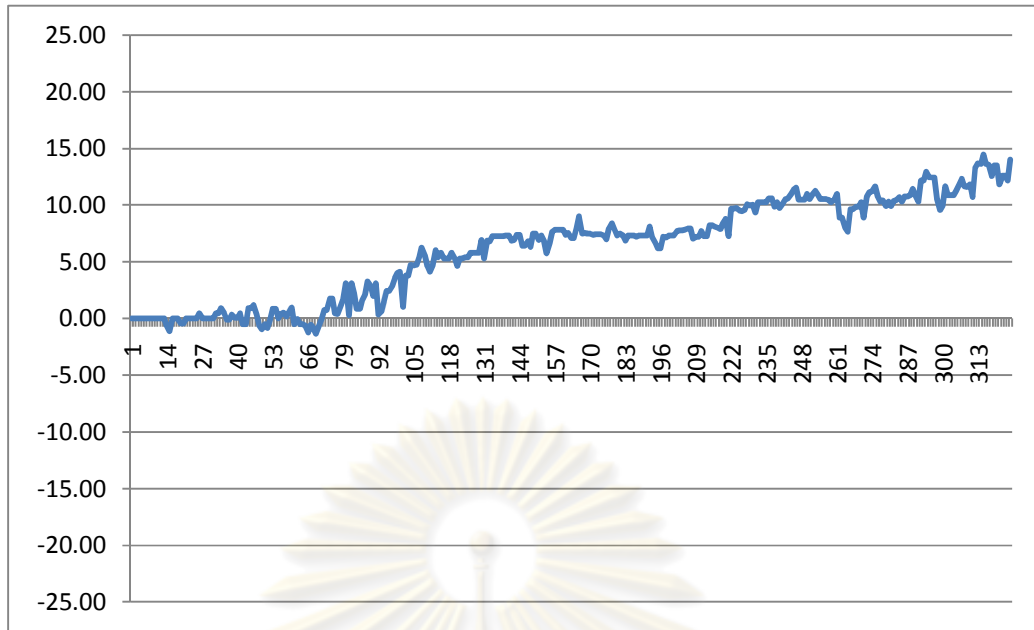
รูปที่ 4.19 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide)



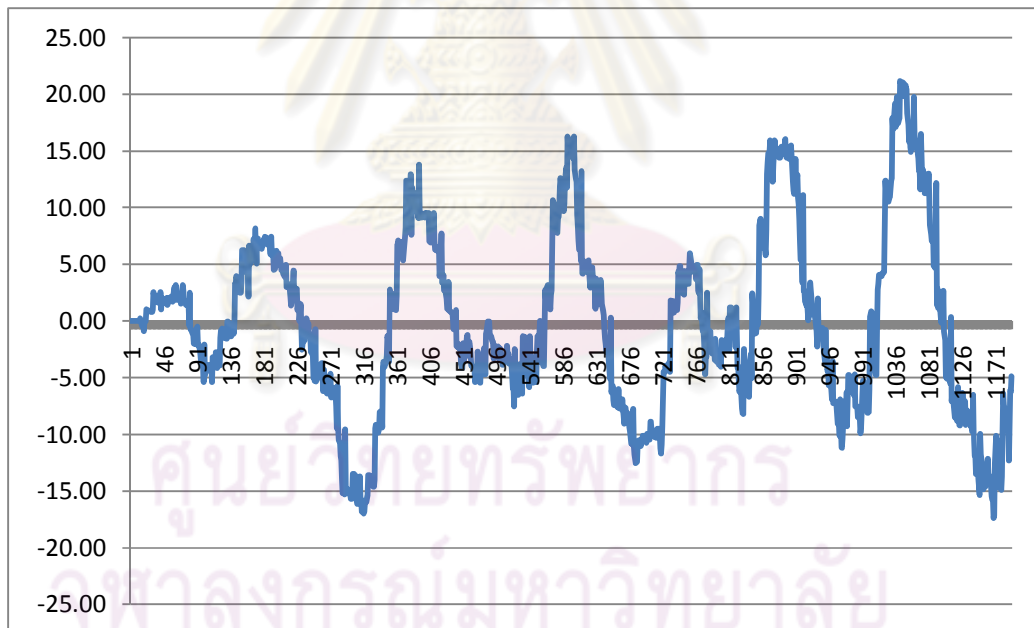
รูปที่ 4.20 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ RAI - RAI ในแผนที่ 1 (SmallDivide)



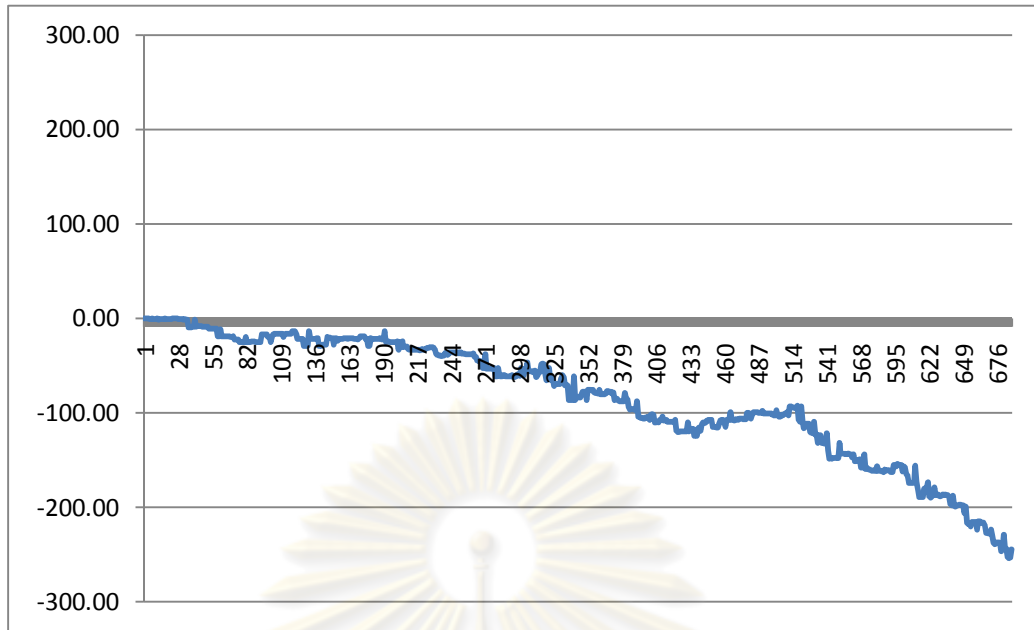
รูปที่ 4.21 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide)



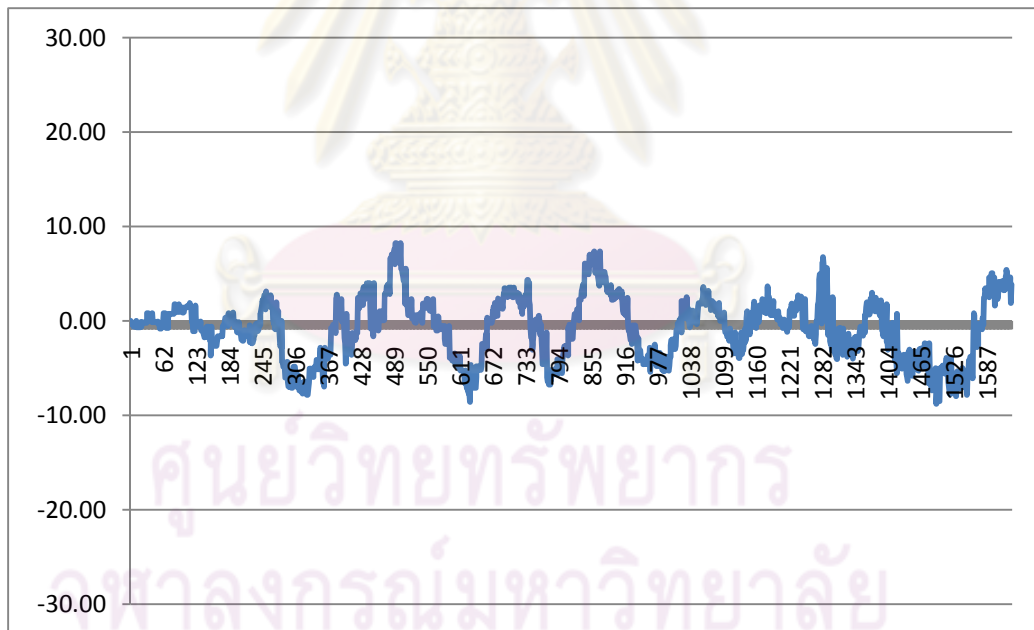
รูปที่ 4.22 ตัวอย่างกราฟฟังก์ชันวัตคະແນของ E323 - E323 ในແຜນທີ່ 2 (Barren)



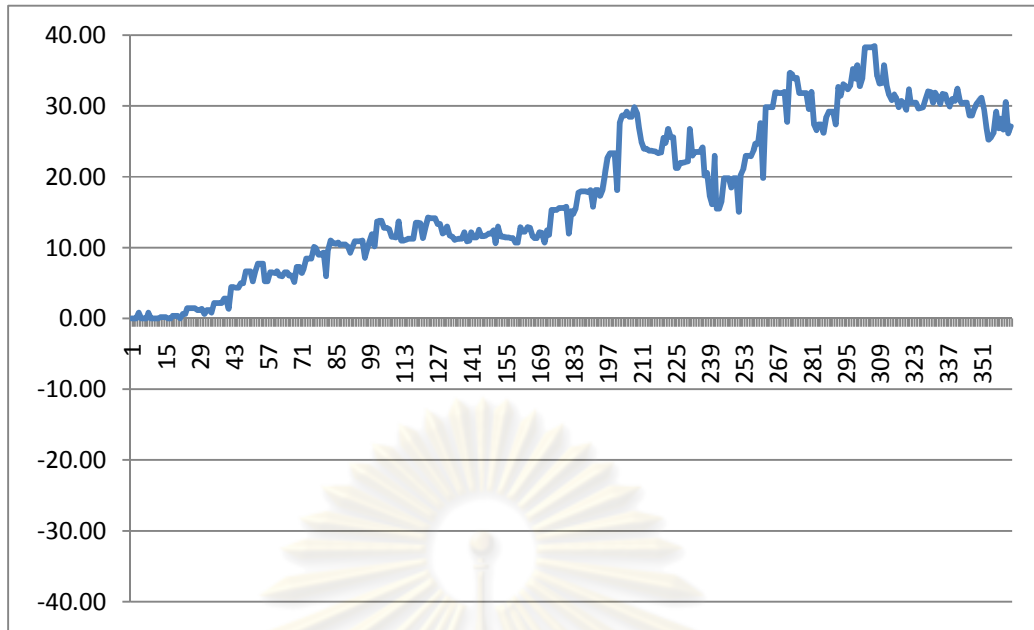
รูปที่ 4.23 ตัวอย่างกราฟฟังก์ชันวัตคະແນของປັດຊຸມປະດິທະນູທີ່ສ້າງຂຶ້ນ - E323 ໃນແຜນທີ່ 2 (Barren)



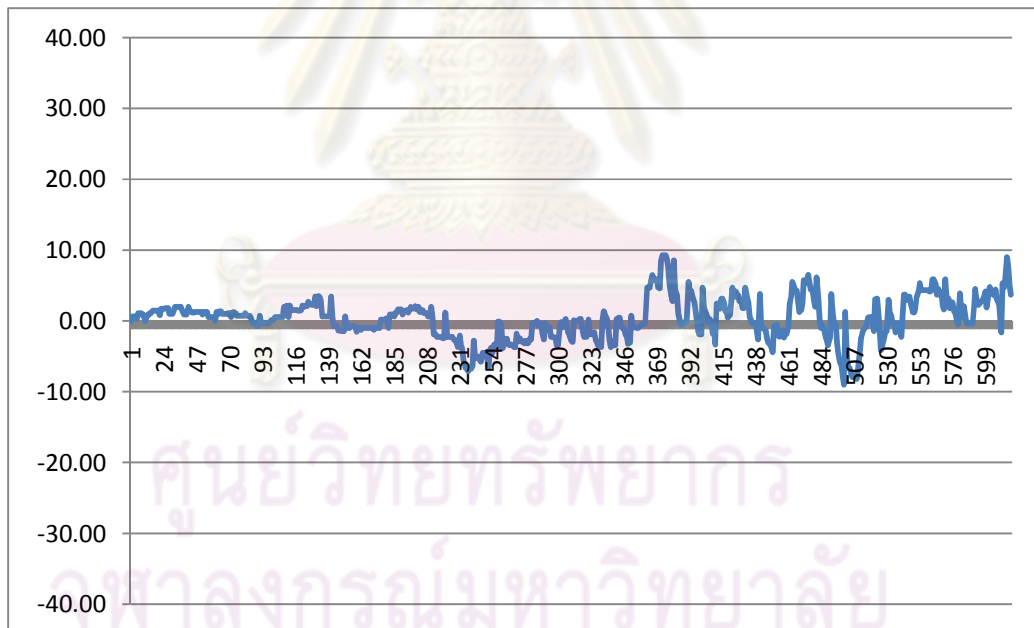
รูปที่ 4.24 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ AAI - AAI ในแผ่นที่ 2 (Barren)



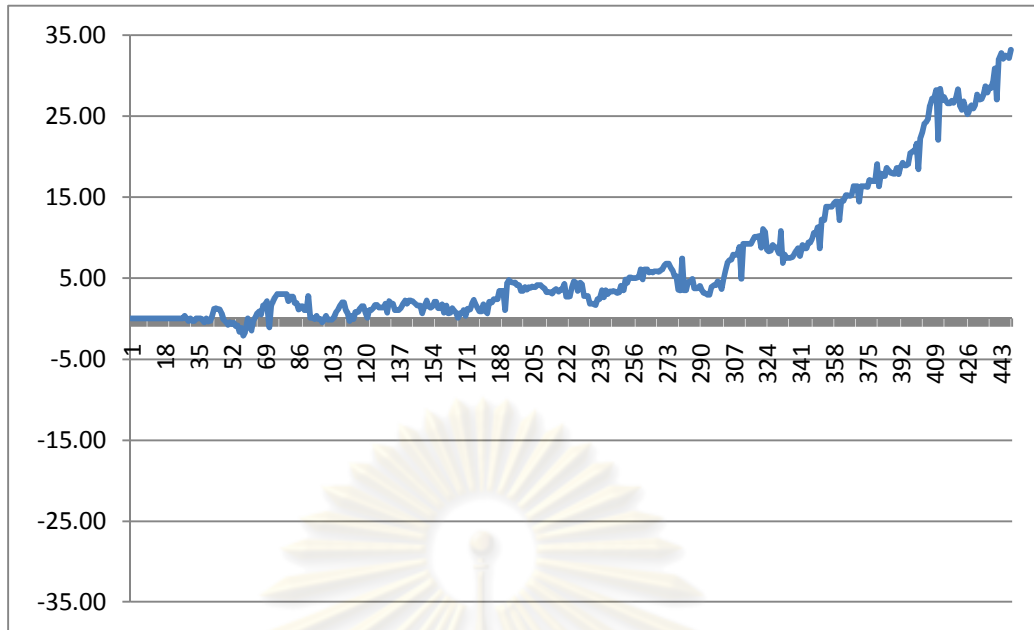
รูปที่ 4.25 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผ่นที่ 2 (Barren)



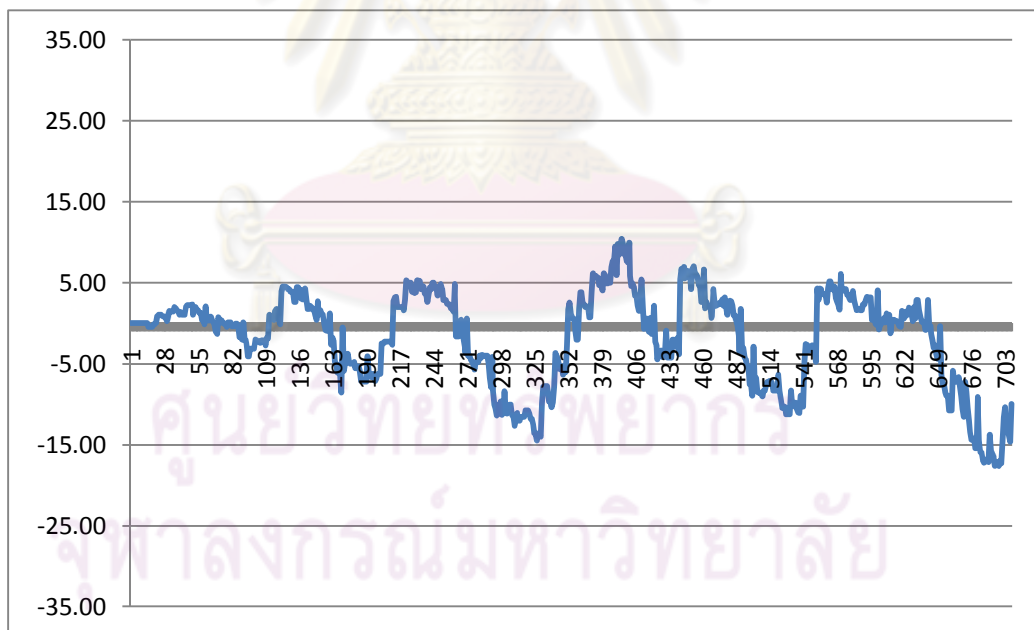
รูปที่ 4.26 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ RAI - RAI ในแผนกที่ 2 (Barren)



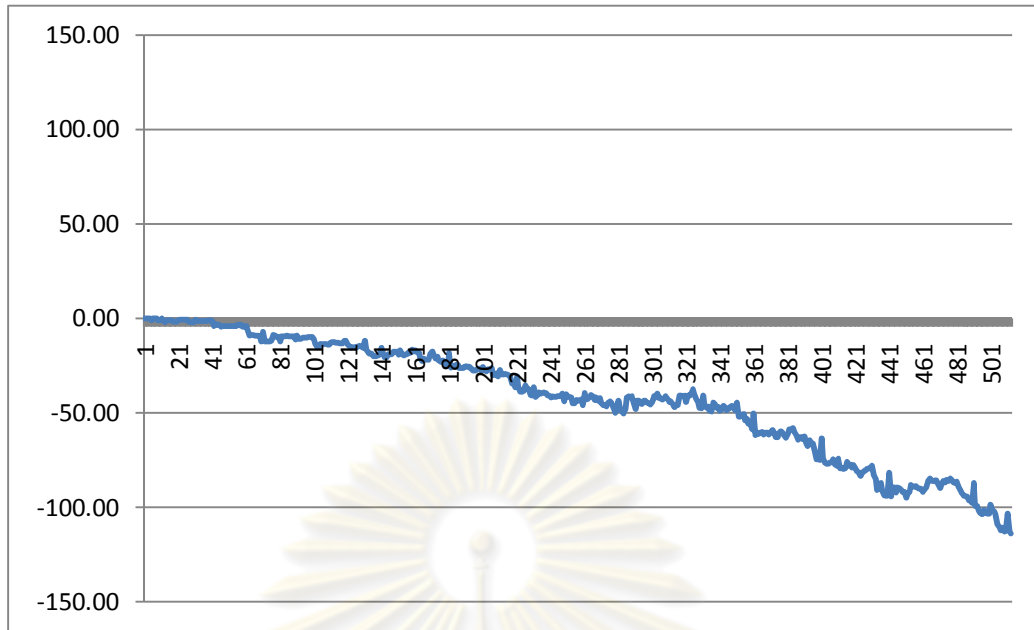
รูปที่ 4.27 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนกที่ 2 (Barren)



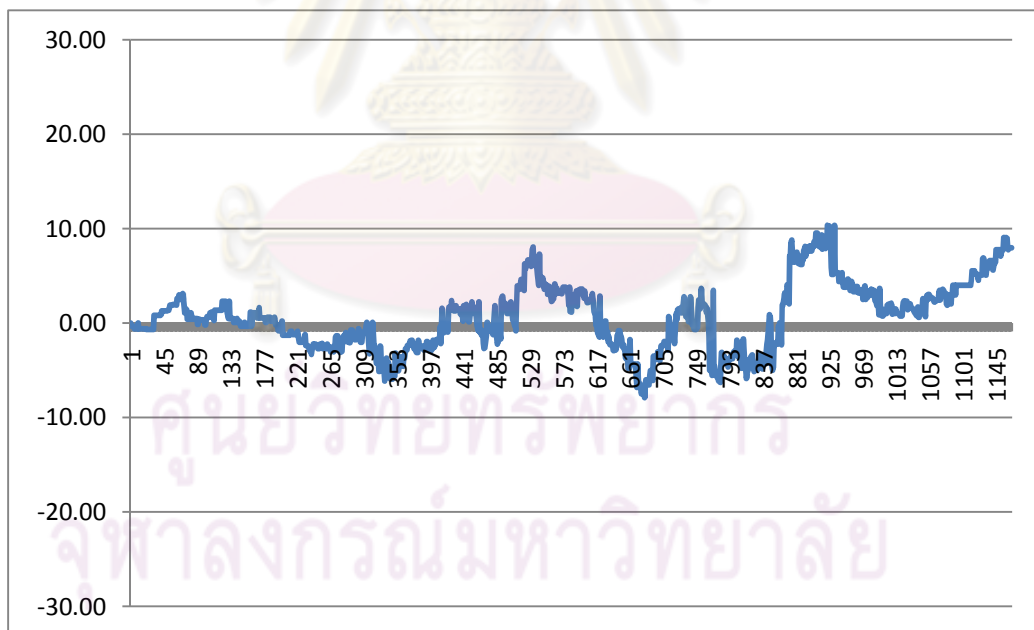
รูปที่ 4.28 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2)



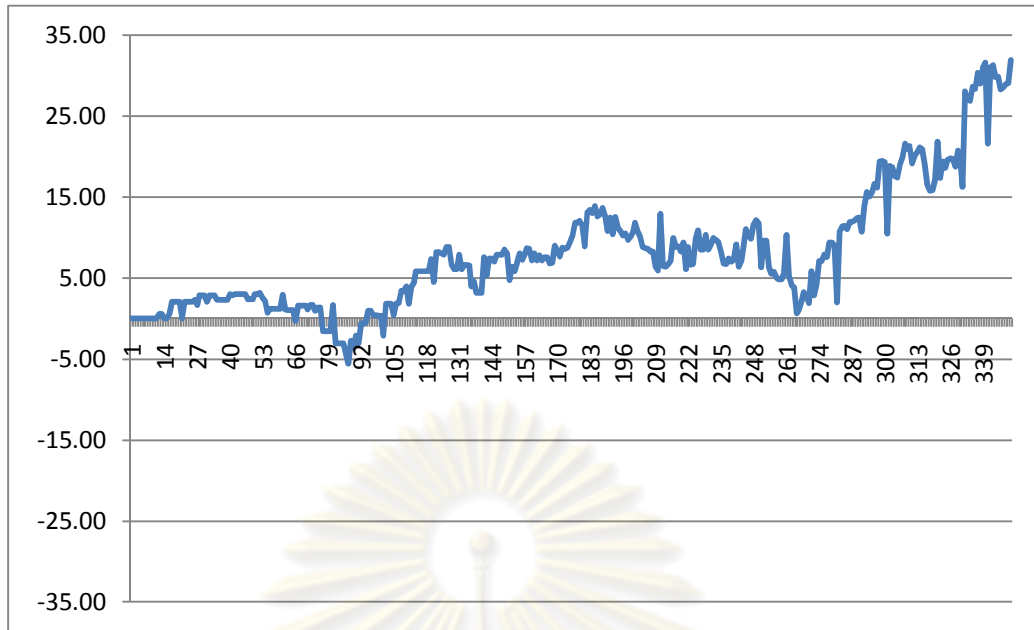
รูปที่ 4.29 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญหาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2)



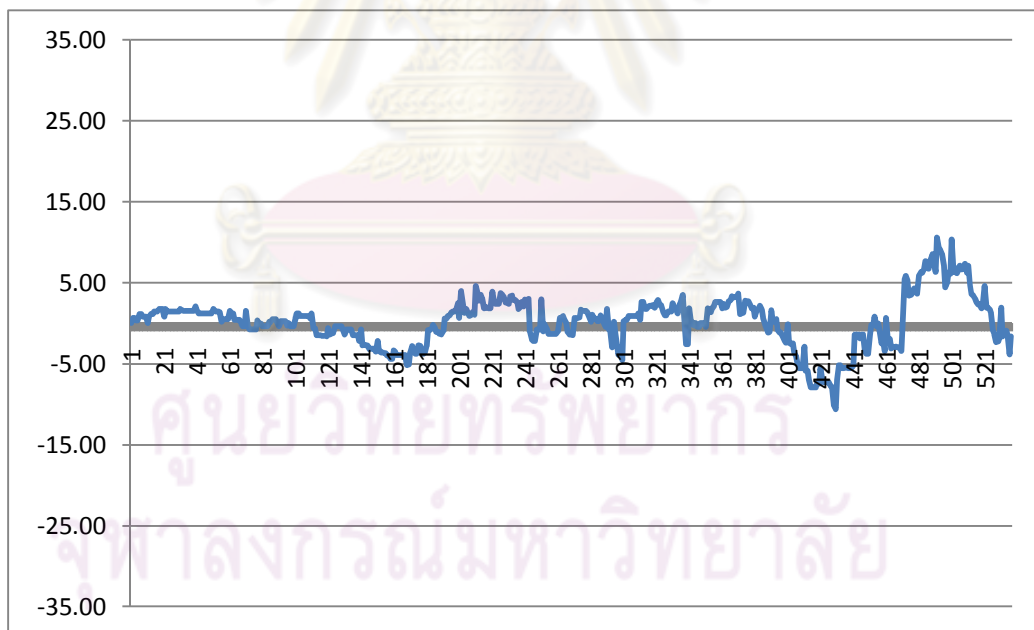
รูปที่ 4.30 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ 4.31 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2)



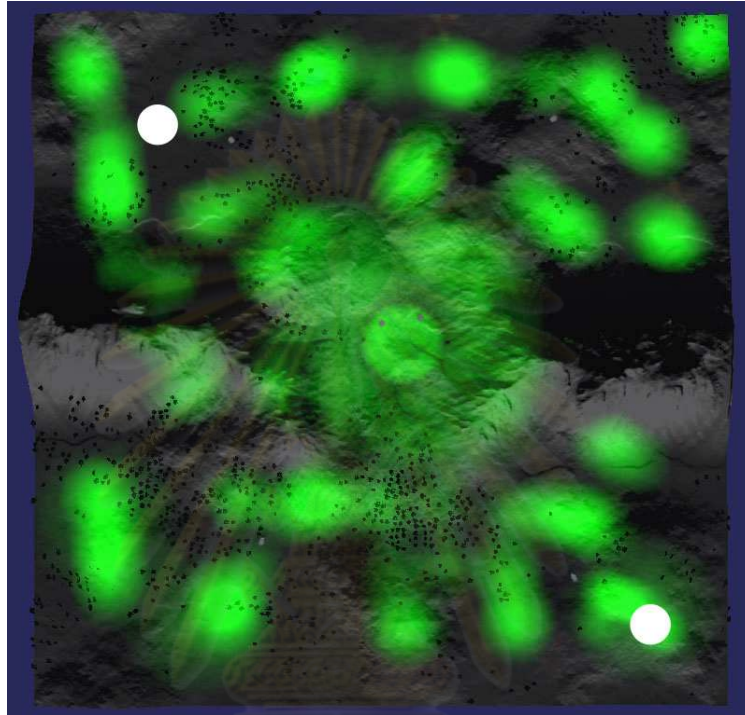
รูปที่ 4.32 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ 4.33 ตัวอย่างกราฟฟังก์ชันวัดคะแนนของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2)

4.7 วิเคราะห์ผลของจุดเกิดกับการทดลองในแต่ละแผนที่

เนื่องมาจากการทดลองเลือกใช้แผนที่แต่ละแบบที่มีลักษณะสมดุลงกันทั้งสองฝั่ง แต่เมื่อใช้ปัญญาประดิษฐ์ตัวเดียวกันสักันผลการ ชนะ/แพ้ ทำเกมในบางการทดลองกลับไม่เท่ากัน ในส่วนนี้จึงมาวิเคราะห์ถึงผลของจุดเกิดและลักษณะของทรัพยากรต่อผลการทดลอง

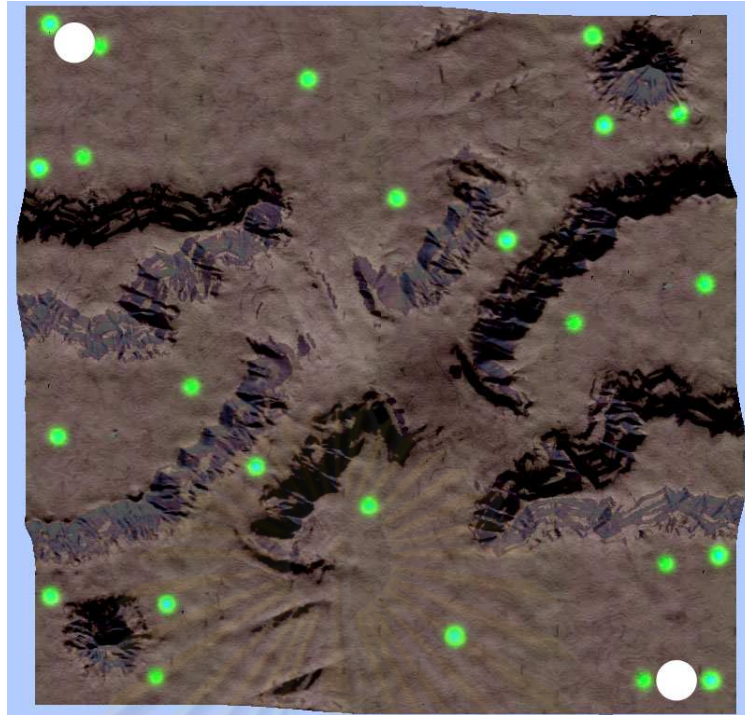


รูปที่ 4.34 ทรัพยากรในแผนที่ 1 (SmallDivide)

จากรูปที่ 4.34 สังเกตว่าในแผนที่ 1 (SmallDivide) ทรัพยากรค่อนข้างมีการกระจายตัว ในลักษณะการกระจายออกไปทั่วทั้งแผนที่ แต่การกระจายตัวไม่ได้มีความสมดุลงกันทั้งแผนที่ โดยแหล่งทรัพยากรหลักจะไปรวมอยู่ในช่วงกลางของแผนที่

กรณี E323 - E323 และกรณี AAI - AAI ปัญญาประดิษฐ์ตัวที่มีจุดเกิดบริเวณมุมขวาล่างจะได้เปรียบกว่าเพราะแหล่งทรัพยากรบริเวณมุมขวาล่างของแผนที่นั้นมีมากกว่าเล็กน้อยทำให้ถึงแม้ใช้ปัญญาประดิษฐ์ตัวเดียวกันสักันผลการ ชนะ/แพ้ ทำเกมก็ยังออกมาไม่เท่ากัน

กรณี RAI - RAI ผลที่ได้ใกล้เคียงกันมาก เนื่องจากว่า พฤติกรรมของ RAI นั้น จะเน้นการกระจายฐานออกไปสร้างฐานย่อย ตามจุดต่างๆ ของแผนที่อยู่ ไม่ได้เน้นที่บริเวณจุดเกิดเป็นหลัก และแผนที่นี้ก็มีลักษณะการกระจายของทรัพยากรค่อนข้างมาก จึงทำให้สามารถสู้ได้อย่างสูสีใกล้เคียงกัน

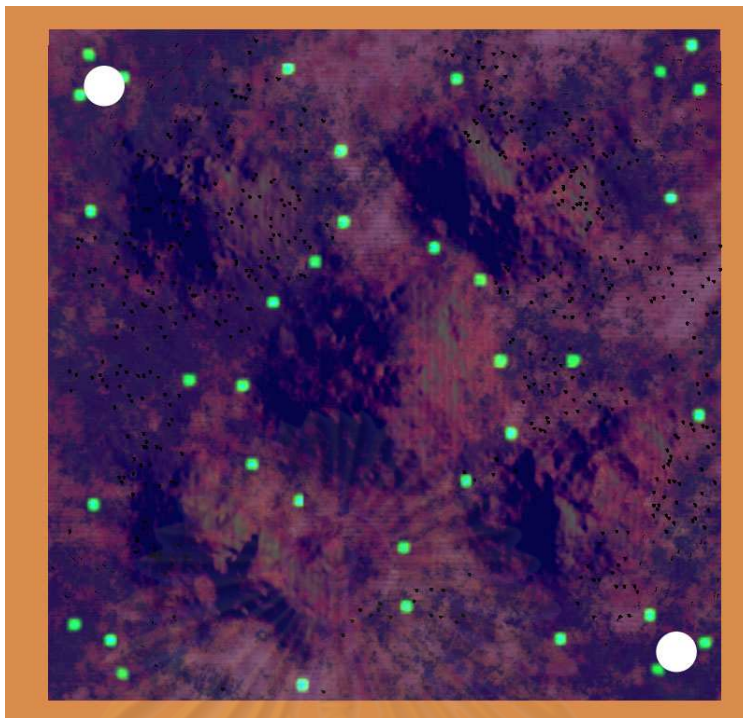


รูปที่ 4.35 ทรัพยากรในแผนที่ 2 (Barren)

จากรูปที่ 4.35 สังเกตว่าในแผนที่ 2 (Barren) ทรัพยากรจะมีลักษณะเป็นจุดกระจายตัวอยู่ตามตำแหน่งต่าง ๆ ในแผนที่ ซึ่งจำนวนแหล่งทรัพยากรนี้ก็มีไม่ค่อยมาก

จะเห็นว่าทั้งกรณี E323 - E323, AAI - AAI และ RAI - RAI แม้ใช้ปัญญาประดิษฐ์ทดสอบตัวเดียวกันสักัน ผล ชนะ/แพ้ ทำเกมก็ยังไม่เท่ากันเนื่องมาจากเหตุผลที่อาจเป็นไปได้คือปัญญาประดิษฐ์ทดสอบทั้งสามตัวเป็นปัญญาประดิษฐ์ที่มีขั้นตอนการทำงานที่แน่นอน (Fixed Script) ทำให้ถึงแม้ว่าแหล่งทรัพยากรจะมีลักษณะสมดุลงัน แต่สภาพภูมิประเทศรอบจุดเกิดที่แตกต่างกันและการวางตัวของทรัพยากรรอบจุดเกิดที่ต่างกันอาจทำให้ปัญญาประดิษฐ์ทดสอบนั้นมีพฤติกรรมแตกต่างกัน อีกเหตุผลหนึ่งคือเนื่องจากจำนวนเกมที่ใช้ในการทดลองแต่ละแบบมีจำนวน 200 เกม ซึ่งอาจน้อยเกินไป

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.36 ทรัพยากรในแผนที่ 3 (Tomb Stone Desert V2)

จากรูปที่ 4.36 สังเกตว่าในแผนที่ 3 (Tomb Stone Desert V2) ทรัพยากรมีการกระจายตัวสมดุลกัน และแบ่งออกเป็นจุดๆ คล้ายกับแผนที่ 2 แต่อาจมีจำนวนแหล่งทรัพยากรมากกว่าแผนที่ 2 เล็กน้อย

ในกรณี E323 - E323 และกรณี RAI - RAI จะเห็นว่าผล ชนะ/แพ้ ท้ายเกมนั้นใกล้เคียงกัน แต่ในกรณี AAI-AAI จะเห็นว่าผล ชนะ/แพ้ ท้ายเกมก็ยังไม่เท่ากันทั้งนี้อาจเนื่องมาจากเหตุผลเดียวกับในแผนที่ 2

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

งานวิจัยนี้ ได้นำเสนอปัญญาประดิษฐ์ที่มีความสามารถทัดเทียมกับผู้เล่นโดยนำเทคนิคการปรับตัวแบบไดนามิกสคริปต์มาปรับปรุงให้สามารถปรับตัวให้เข้ากับผู้เล่นขณะที่ผู้เล่นกำลังเล่นเกมอยู่ โดยใช้วิธีการตัดค่าน้ำหนักที่สูงในการปรับระดับความยาก และใช้การผลิตยูนิตมาเป็นกฎพื้นฐาน ได้มีการทำการทดลองและวิเคราะห์ผลไว้อย่างละเอียดดังที่นำเสนอในบทที่ 4 ซึ่งจะเห็นได้ว่าการผลิตยูนิตนั้นสามารถใช้ในการปรับระดับความยากให้สูสีกับระดับของผู้เล่นได้ และได้ทำการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การผลิตยูนิตมาเป็นกฎพื้นฐาน โดยผลจากการวิจัยทั้งหมดที่ได้นำเสนอไปนั้นสามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

1. ไดนามิกสคริปต์สามารถนำมาใช้ในเกมวางแผนการรบแบบตอบสนองทันกาล และการผลิตยูนิตนั้นสามารถใช้ในการปรับระดับความยากให้สูสีกับระดับของผู้เล่นได้ โดยจากผลการทดลองจะเห็นได้ว่าปัญญาประดิษฐ์ที่สร้างขึ้นที่ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์ และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) นั้นมีเวลาเฉลี่ยที่ดีกว่าคู่ ปัญญาประดิษฐ์ทดสอบ - ปัญญาประดิษฐ์ทดสอบ (แสดงถึงผู้เล่นคนเดียวกัน) ส่วน ผลการ ชนะ/แพ้ ทำเกม แม้ว่าบางการทดลองให้ผลแยกว่าคู่ ปัญญาประดิษฐ์ทดสอบ - ปัญญาประดิษฐ์ทดสอบ แต่ก็สามารถทำการปรับปรุงได้อีก

2. การใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) นั้นเป็นวิธีที่ดีที่สุดในการปรับปรุงไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูงให้เหมาะสมกับการใช้การผลิตยูนิตมาเป็นกฎพื้นฐานโดยจากผลการทดลองจะเห็นได้ว่า เวลาเฉลี่ยของการปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) นั้นสูงกว่า และ ผลการ ชนะ/แพ้ ทำเกม จะเข้าใกล้ 50 เปอร์เซ็นต์มาก ซึ่งเป็นอัตราส่วนที่ดีกว่าปัญญาประดิษฐ์ที่สร้างขึ้นทุกแบบอีกด้วย

3. ปัญญาประดิษฐ์ที่สร้างขึ้น สามารถสู้กับปัญญาประดิษฐ์ทดสอบซึ่งเสมือนเป็นผู้เล่นแต่ละคนซึ่งมีพฤติกรรมในการเล่นเกมที่แตกต่างกัน โดยใช้การผลิตยูนิตเพื่อปรับให้เข้ากับปัญญาประดิษฐ์ทดสอบแต่ละตัวโดยไม่ต้องปรับพฤติกรรมของปัญญาประดิษฐ์ ใช้เพียงแต่ปรับเปลี่ยนน้ำหนักของยูนิต (w_u), ตัวแปรที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละ

เฟสของเกม (w_p) และน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานตามปัญญาประดิษฐ์ทดสอบแต่ละตัว ในแต่ละแผนที่เท่านั้น ซึ่งทำให้สะดวกในการสร้างปัญญาประดิษฐ์ เพราะสร้างปัญญาประดิษฐ์ ครั้งเดียวแต่สามารถปรับใช้ได้กับผู้เล่นหลายประเภท

5.2 ข้อเสนอแนะ

จากผลการวิจัยจะเห็นว่าแม้ว่าผลการวิจัยจะแสดงให้เห็นว่าการผลิตยูนิตนั้น สามารถใช้ในการปรับระดับความยากให้สูสีกับระดับของผู้เล่นได้ แต่การที่ผู้เล่นเกมเล่นกับ ปัญญาประดิษฐ์ที่ปรับระดับความยากให้สูสีนั้น บางครั้งอาจเกิดความเบื่อหน่ายเนื่องจากใช้เวลา ในการเล่นนานเกินไป จึงควรจะใช้ปัญญาประดิษฐ์ที่ปรับระดับความยากให้สูสีในช่วงเวลาหนึ่ง เท่านั้น เช่น อาจให้ปัญญาประดิษฐ์ที่นำเสนอเล่นในช่วงห้านาทีแรก จากนั้นจึงเปลี่ยนให้เป็น ปัญญาประดิษฐ์อย่างอื่น หรือ อาจใช้ปัญญาประดิษฐ์ที่นำเสนอเล่นเกมในช่วงที่ปัญญาประดิษฐ์ ไกลจะแพ้ หรือไกลจะชนะ เพียงช่วงหนึ่งของเกมก็ได้ เพื่อให้ฝ่ายที่เสียเปรียบสามารถมีโอกาส แก่สถานการณ์ได้

ข้อเสนอแนะต่อมาก็คือนำหนักของยูนิต (w_u), ตัวแปรที่บ่งบอกความสำคัญของ เทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) และน้ำหนักเริ่มต้นของกฎในกฎพื้นฐานนั้นจำเป็นต้อง ใช้เกมที่ปัญญาประดิษฐ์สู้กับผู้เล่นในการเรียนรู้ ซึ่งในการทดลองนี้ใช้ปัญญาประดิษฐ์ ทดสอบแทนที่ผู้เล่นแต่ละคน ทำให้สามารถใช้เกมที่ใช้เพื่อการเรียนรู้จำนวนมากได้ แต่การเล่น กับผู้เล่นที่เป็นคนจริงนั้นไม่สามารถทำได้ ในการทดลองกับผู้เล่นที่เป็นคนจริงจึงเสนอให้ผู้เล่น ลองเล่นกับปัญญาประดิษฐ์ E323, AAI และ RAI ทีละตัว แล้วสอบถามผู้เล่นว่า ผู้เล่นคิดว่าเมื่อ เล่นกับปัญญาประดิษฐ์ตัวใดแล้วสูสีกับตัวเองมากที่สุด แล้วนำค่าน้ำหนักของยูนิต (w_u), ตัวแปร ที่บ่งบอกความสำคัญของเทอม v_1 และ v_2 ในแต่ละเฟสของเกม (w_p) และน้ำหนักเริ่มต้นของกฎ ในกฎพื้นฐาน ของปัญญาประดิษฐ์ที่สร้างขึ้นสำหรับใช้กับปัญญาประดิษฐ์ตัวที่ผู้เล่นคิดว่าสูสี มากที่สุดนั้นมา让玩家ลองเล่นจริง แล้ววัดผลโดยใช้แบบสอบถามว่าผู้เล่นคิดว่าเล่นกับ ปัญญาประดิษฐ์ที่สร้างขึ้นนั้นสูสีกว่าเล่นกับปัญญาประดิษฐ์ตัวที่ผู้เล่นคิดว่าสูสีในตอนแรก หรือไม่

ปัญหาอีกอย่างคือปัญหาของความสมจริง เนื่องจากยูนิตที่สร้างขึ้นยังไม่ผูกกับ สิ่งก่อสร้างที่ผลิตยูนิตชนิดนั้นๆ อาจเกิดปัญหาว่ามียูนิตปรากฏโดยยังไม่มีสิ่งก่อสร้างที่สามารถ สร้างยูนิตนั้นได้ วิธีแก้คือเข้าไปคุมอัตราการสร้างสิ่งก่อสร้างแทนยูนิต หรืออาจเข้าไปคุมทั้ง อัตราการสร้างสิ่งก่อสร้างและอัตราการผลิตยูนิต โดยลดอัตราการผลิตยูนิตลงถ้าสิ่งก่อสร้างที่ สามารถสร้างยูนิตชนิดนั้นถูกทำลาย เพื่อความสมจริง

ในงานวิจัยนี้กำหนดให้ทดลองกับเกมที่มีผู้เล่น 2 ฝ่ายเท่านั้น อันเนื่องมาจาก ข้อจำกัดของฟังก์ชันวัตถุประสงค์และแผนของ Bakkes ซึ่งขึ้นกับข้อมูลของผู้เล่น 2 ฝ่าย ในการปรับปรุงให้

ดีขึ้นนั้น อาจทำการหาฟังก์ชันวัดคะแนนใหม่ที่สามารถวัดระดับของผู้เล่นได้หลายฝ่าย เพื่อให้ปัญญาประดิษฐ์ที่สร้างขึ้นสามารถปรับความยากให้เข้ากับผู้เล่นที่เล่นพร้อมกันหลายคนได้

ในวิทยานิพนธ์นี้ได้ใช้ชุดของกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้นในการสู้กับปัญญาประดิษฐ์แต่ละตัวคนละชุดกัน เนื่องมาจากในขั้นตอนการเลือกกฎเพื่อมาสร้างเป็นกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้นนั้นนั้น ถ้าใช้สู้กับปัญญาประดิษฐ์ตัวที่มีความสามารถต่ำ เช่น AAI การที่ไดนามิกสคริปต์สุ่มเลือกกฎมาใช้ในชุดสคริปต์แล้วได้กฎย่อยในกฎพื้นฐานที่ผลิตรายการหลายตัวอาจทำให้ปัญญาประดิษฐ์ตัวที่มีความสามารถต่ำไม่สามารถสู้ได้ ดังนั้นถ้าเลือกกฎย่อยในกฎพื้นฐานที่ผลิตรายการหลายตัวมาใช้เป็นกฎในกฎพื้นฐานแล้ว แม้ขั้นตอนการเรียนรู้ค่าน้ำหนักของกฎในกฎพื้นฐานจะทำให้น้ำหนักของกฎย่อยในกฎพื้นฐานที่ผลิตรายการหลายตัวนั้นมีค่าต่ำมาก ๆ กฎนั้นก็ยังมีโอกาสถูกเลือกมาใช้ในชุดสคริปต์อยู่ดี เนื่องจากว่าในบางครั้งชุดสคริปต์ถูกลดค่าน้ำหนักแล้วนำไปกระจายค่าน้ำหนักให้กฎอื่นๆ ที่ไม่ถูกเลือก กฎที่มีน้ำหนักต่ำนั้นอาจจะมีค่าน้ำหนักสูงขึ้นมาและมีโอกาสในการถูกเลือกมาใช้ได้ ดังนั้นสำหรับการสู้กับปัญญาประดิษฐ์ที่มีความสามารถต่ำนั้น จำเป็นต้องเลือกกฎหลายข้อทิ้งไประหว่างกระบวนการหาน้ำหนักของยูนิท ถ้าจะไม่นำกฎเหล่านั้นออกไป อาจแก้ปัญหาความยุ่งเหยิงไปของปัญญาประดิษฐ์ที่สร้างขึ้นได้โดยใช้วิธีการลบกฎทิ้งในระหว่างการเล่น โดยให้ระบบทำการลบกฎที่มีน้ำหนักน้อยเกินค่าที่กำหนดออกจากเกมทันที

ในการที่จะนำวิธีการที่สร้างปัญญาประดิษฐ์ที่สามารถปรับความยากให้เข้ากับผู้เล่นในงานวิจัยนี้ไปใช้ในเกมวางแผนการรบแบบตอบสนองทันกาลเกมอื่น เราสามารถนำการปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) ไปใช้ได้เลย แต่ส่วนฟังก์ชันวัดคะแนนอาจจะใช้ฟังก์ชันวัดคะแนนที่ตัดส่วนที่วัดคะแนนส่วนความปลอดภัยของ Commander ออกไป การออกแบบกฎย่อยในกฎพื้นฐานให้ใช้ยูนิทที่เหมาะสมกับเกมวางแผนการรบแบบตอบสนองทันกาลเกมนั้นแทน และอาจใช้วิธีการนี้เพียงช่วงเวลาหนึ่งภายในเกมดังที่ได้กล่าวไว้แล้ว

จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] WCG, Inc. (2008). World Cyber Game [Online]. Available from <http://www.wcg.com/6th/2008/games/officialgames.asp> [2009, March 17]
- [2] Tozour, P. (2002). The Evolution of Game AI. AI Game Programming Wisdom, pp. 3-15. Charles River Media.
- [3] Ponsen, M., Muñoz-Avila, H., Spnck, P., and Aha, D. W. (2005). Automatically Acquiring Domain Knowledge For Adaptive Game AI Using Evolutionary Learning. AAAI, pp. 1535-1540.
- [4] Spronck, P., and Ponsen M. (2008). Automatic Generation of Strategies. AI Game Programming Wisdom 4, pp. 659-670. Charles River Media.
- [5] Adams, E., and Rollings, A. (2007). Fundamentals of Game Design. New Jersey: Person Education.
- [6] Koster, R. (2004). Theory of Fun for Game Design, Phoenix: Paraglyph Press.
- [7] Wikipedia. Dune 2 [Online]. Available from http://en.wikipedia.org/wiki/Dune_II [2009, March 17]
- [8] Wikipedia. TA Spring [Online]. Available from [http://en.wikipedia.org/wiki/Spring_\(computer_game\)](http://en.wikipedia.org/wiki/Spring_(computer_game)) [2009, March 17]
- [9] Bakkes, S., Kerbusch, P., Spronck, P., and Herik, J.V.D. (2007). Predicting Success in an Imperfect-Information Game. Proceedings of the Computer Games Workshop 2007, pp. 219-230. Universiteit Maastricht, The Netherlands.
- [10] Beal, D., and Smith, M. (1997). Learning piece values using temporal Differences. International Computer Chess Association (ICCA) Journal. 20. 3: pp. 147-151.
- [11] Spronck, P., Sprinkhuizen-Kuyper, I. G., and Postma, E. O. (2004). On-line Adaptation of Game Opponent AI with Dynamic Scripting. International Journal of Intelligent Games and Simulation, 3. 1: pp. 45-53. University of Wolverhampton: EUROSIS.
- [12] Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I., and Postma, E. (2006). Adaptive Game AI with Dynamic Scripting. Machine Learning, 63. 3: pp. 217-248.

- [13] Spronck, P., Sprinkhuizen-Kuyper, I. G., and Postma, E. O. (2004). Difficulty Scaling of Game AI. GAME-ON 2004: 5th International Conference on Intelligent Games and Simulation, pp. 33-37. Belgium: EUROSIS.
- [14] Bakkes, S., Spronck, P., and Herik, J. V. D. (2007). Phase-dependent Evaluation in RTS games. Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence, pp. 3-10. Universiteit Utrecht, The Netherlands.
- [15] Bakkes, S., Kerbusch, P., Spronck, P., and Herik, J. V. D. (2007). Automatically Evaluating the Status of an RTS game. Proceedings of the Annual Belgian-Dutch Machine Learning Conference, pp. 143-144. University of Amsterdam.
- [16] Bakkes, S., and Spronck, P. (2008). Automatically Generating a Score Function for Strategy Games. AI Game Programming Wisdom 4, pp. 647-658. Charles River Media.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



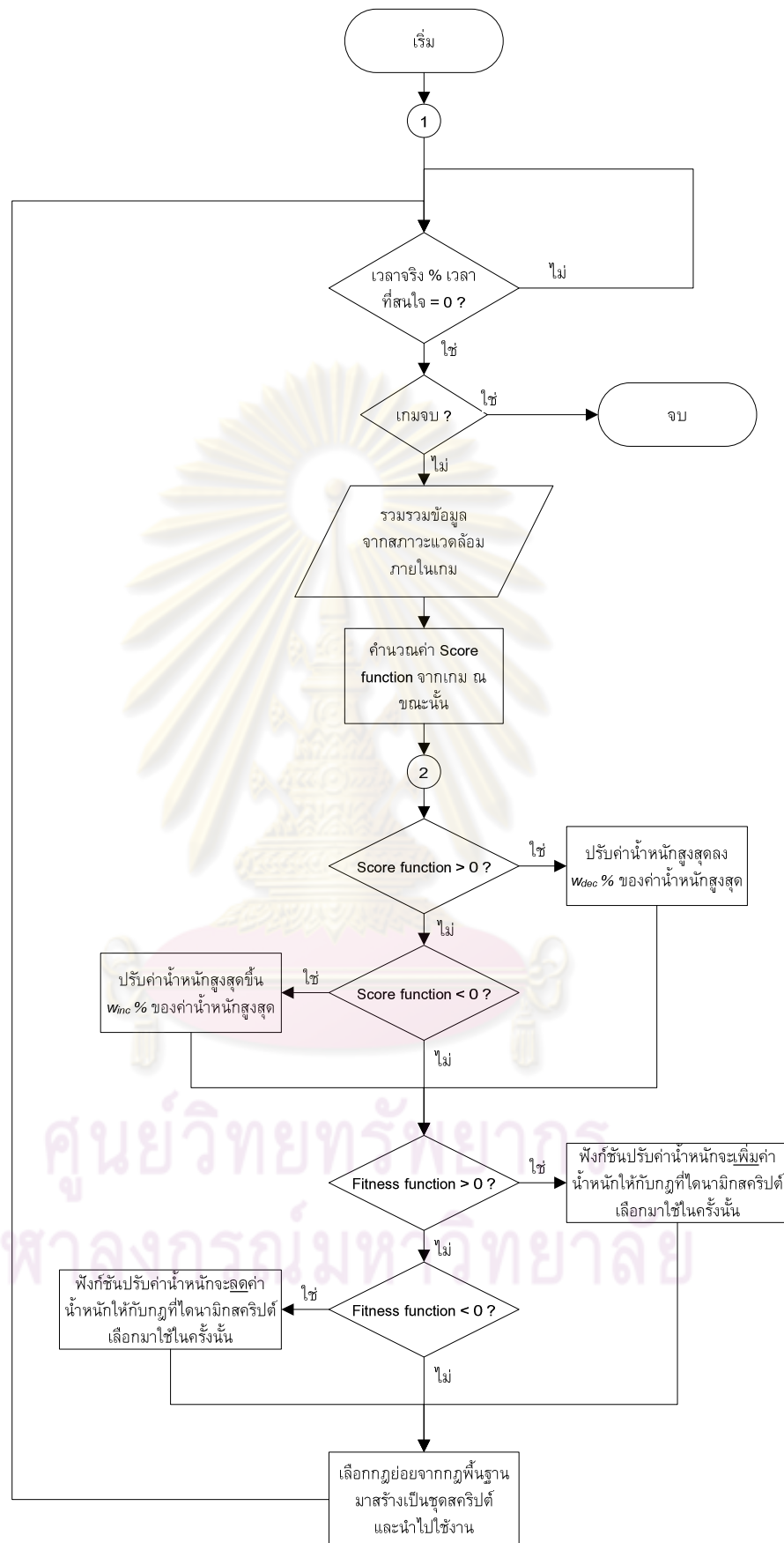
ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

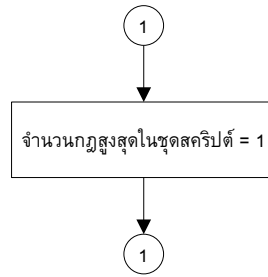
ภาคผนวก ก

ผังงานการทำงานของปัญญาประดิษฐ์ที่สร้างขึ้นจะมี 5 แบบ ดังนี้

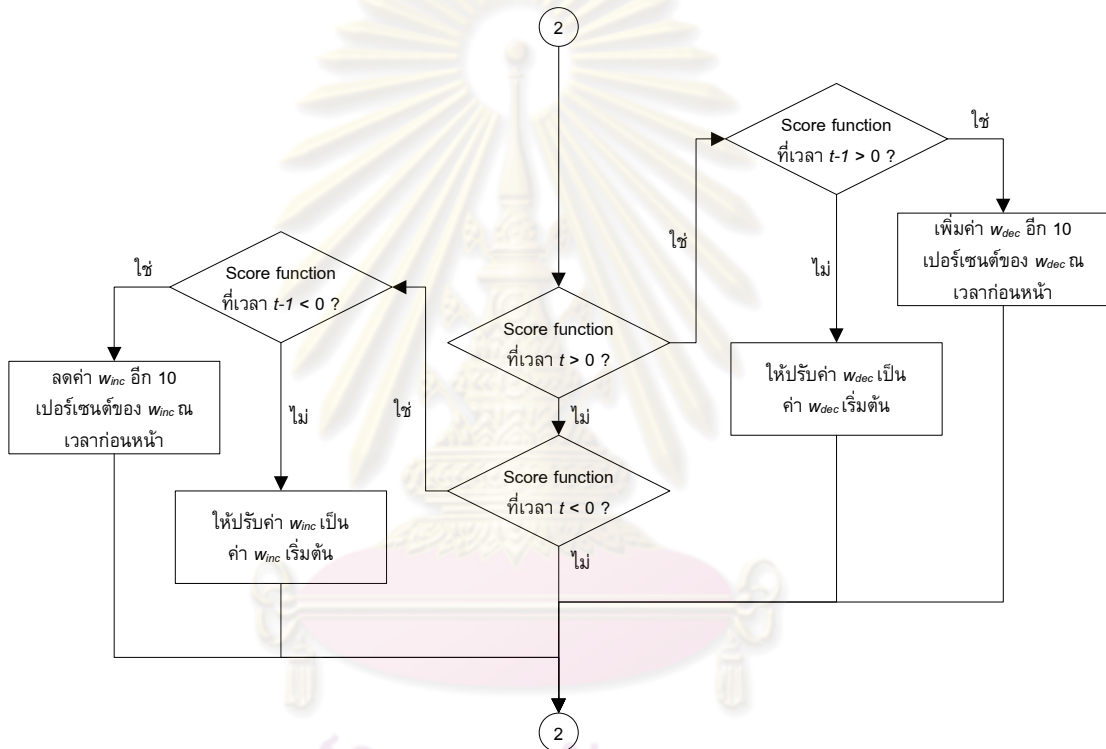
- ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule) แสดงด้วยรูปที่ ก.1 โดยแทนค่าจุดเชื่อม 1 ด้วยรูปที่ ก.2 และจุดเชื่อม 2 ไม่ต้องแทนค่า
- ใช้เทคนิคการเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า (Change w_{inc} , w_{dec}) แสดงด้วยรูปที่ ก.1 โดยจุดเชื่อม 1 ไม่ต้องแทนค่า และแทนค่าจุดเชื่อม 2 ด้วยรูปที่ ก.3
- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ทีละกฎทุกๆ ช่วงเวลาหนึ่ง และวิธีการตัดค่าน้ำหนักที่สูง ในการทดลองจะเลือกเพิ่มกฎในชุดสคริปต์ทีละกฎทุกๆ 7 นาที (7-minute increment) แสดงด้วยรูปที่ ก.1 โดยจุดเชื่อม 1 ไม่ต้องแทนค่า และแทนค่าจุดเชื่อม 2 ด้วยรูปที่ ก.4
- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules) แสดงด้วยรูปที่ ก.1 โดยจุดเชื่อม 1 ไม่ต้องแทนค่า และแทนค่าจุดเชื่อม 2 ด้วยรูปที่ ก.5
- ใช้การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์และแสดงถึงระดับของปัญญาประดิษฐ์ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า (Mix) วิธีนี้เป็นารผสมวิธี (Change w_{inc} , w_{dec}) กับ (Change number of rules) เข้าด้วยกัน แสดงด้วยรูปที่ ก.1 โดยจุดเชื่อม 1 ไม่ต้องแทนค่า และแทนค่าจุดเชื่อม 2 ด้วยรูปที่ ก.6



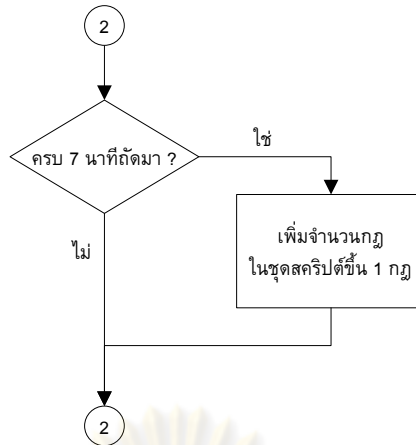
รูปที่ ก.1 ผังงานการทำงานของไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง



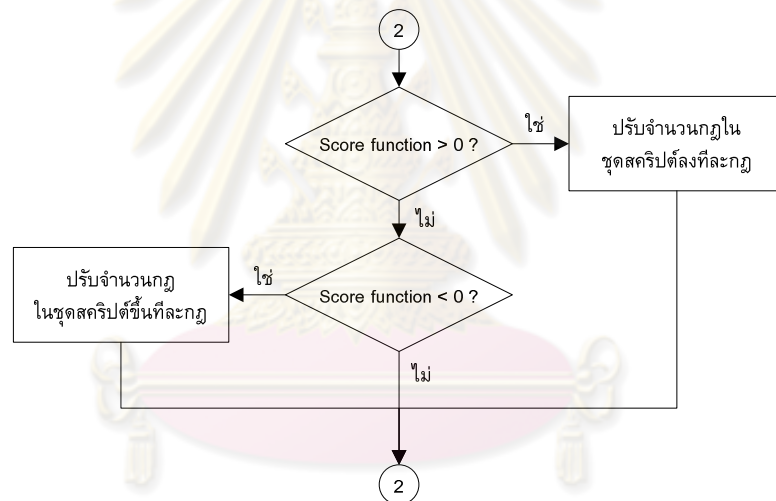
รูปที่ ก.2 ผังงานการทำงานของกรเรียนรู้แบบไดนามิกสคริปต์และวิธีการตัดค่าน้ำหนักที่สูง โดยจำกัดจำนวนกฎในชุดสคริปต์ไว้ 1 กฎ (Fix 1 rule)



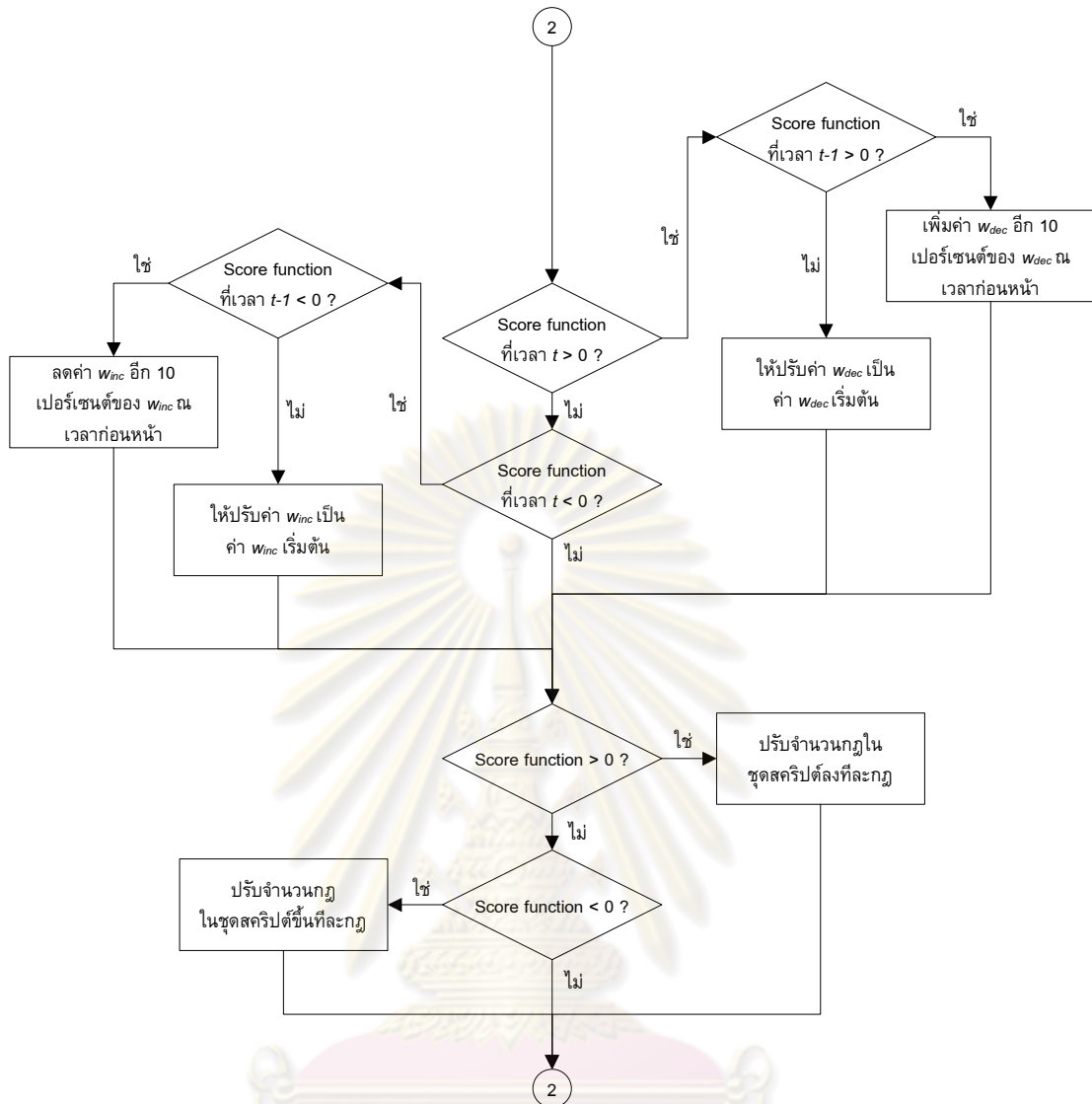
รูปที่ ก.3 ผังงานการทำงานของกรเรียนรู้แบบไดนามิกสคริปต์และการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัตถุประสงค์ที่แสดงถึงระดับของปัญหาประติษฐ์ ขณะนั้นกับค่าฟังก์ชันวัตถุประสงค์ ณ เวลาก่อนหน้า (Change w_{inc} , w_{dec})



รูปที่ ก.4 ผังงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบเพิ่มจำนวนกฎในชุดสคริปต์ที่ละกฎทุกๆ ช่วงเวลาหนึ่ง และวิธีการตัดค่าน้ำหนักที่สูง ในการทดลอง จะเลือกเพิ่มกฎในชุดสคริปต์ที่ละกฎทุกๆ 7 นาที (7-minute increment)



รูปที่ ก.5 ผังงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและวิธีการตัดค่าน้ำหนักที่สูง (Change number of rules)



รูปที่ ก.6 ผังงานการทำงานของ การปรับปรุงเทคนิคการเรียนรู้แบบไดนามิกสคริปต์แบบการปรับจำนวนกฎในชุดสคริปต์ตามระดับของศัตรูและการปรับปรุงวิธีการตัดค่าน้ำหนักที่สูงโดยการตรวจสอบค่าฟังก์ชันวัดคะแนนซึ่งแสดงถึงระดับของปัญหาประติบัติขณะนั้นกับค่าฟังก์ชันวัดคะแนน ณ เวลาก่อนหน้า (Mix)

จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

รายละเอียดของยูนิตที่ใช้สร้างกฏ รายละเอียดของกฏย่อยในกฎพื้นฐานของแต่ละ
 ละครูปปัญหาประดิษฐ์ที่สร้างขึ้น-คู่มือปัญหาประดิษฐ์ทดสอบ และคำนำหน้ากเริ่มต้น



ชื่อยูนิต Warrior

ระดับของยูนิต 1

ประเภทของยูนิต Medium Infantry Kbot



ชื่อยูนิต Janus

ระดับของยูนิต 1

ประเภทของยูนิต Twin Medium Rocket Launcher



ชื่อยูนิต Hammer

ระดับของยูนิต 1

ประเภทของยูนิต Light Plasma Kbot



ชื่อยูนิต Rocko

ระดับของยูนิต 1

ประเภทของยูนิต Rocket Kbot



ชื่อยูนิต Jeffy

ระดับของยูนิต 1

ประเภทของยูนิต Light Scout Vehicle



ชื่อยูนิต Jethro

ระดับของยูนิต 1

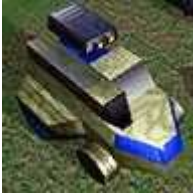
ประเภทของยูนิต Amphibious Anti-air Kbot



ชื่อยูนิต Stumpy

ระดับของยูนิต 1

ประเภทของยูนิต Medium Assault Tank



ชื่อยูนิต Samson

ระดับของยูนิต 1

ประเภทของยูนิต Missile Truck



ชื่อยูนิต Zeus

ระดับของยูนิต 2

ประเภทของยูนิต Assault Kbot



ชื่อยูนิต Pewee

ระดับของยูนิต 1

ประเภทของยูนิต Infantry Kbot



ชื่อยูนิต Flea

ระดับของยูนิต 1

ประเภทของยูนิต Fast Scout Kbot



ชื่อยูนิต Flash

ระดับของยูนิต 1

ประเภทของยูนิต Fast Assault Tank

กฎย่อยในกฎพื้นฐานของแต่ละคู่มือมาตรฐานประดิษฐ์ที่สร้างขึ้น-ปัญญาประดิษฐ์
ทดสอบ และคำนำหน้านักเริ่มต้น

- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.1
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.2
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.3
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 1 (SmallDivide) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.4
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น- AAI ในแผนที่ 2 (Barren) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.5
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น- AAI ในแผนที่ 3 (Tomb Stone Desert V2) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.6
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 1 (SmallDivide) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.7
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น- RAI ในแผนที่ 2 (Barren) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.8
- กฎย่อยในกฎพื้นฐานของปัญญาประดิษฐ์ที่สร้างขึ้น- RAI ในแผนที่ 3 (Tomb Stone Desert V2) และคำนำหน้านักเริ่มต้น แสดงดังรูปที่ ข.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.02911825
สร้างยูนิต Warrior จำนวน 1 ตัว	1.02954290
สร้างยูนิต Warrior จำนวน 2 ตัว	4.27519751
สร้างยูนิต Warrior จำนวน 3 ตัว	9.92253590
สร้างยูนิต Warrior จำนวน 4 ตัว	0.08711100
สร้างยูนิต Janus จำนวน 1 ตัว	1.02225441
สร้างยูนิต Janus จำนวน 2 ตัว	2.47750711
สร้างยูนิต Janus จำนวน 3 ตัว	4.52451229
สร้างยูนิต Janus จำนวน 4 ตัว	7.26437187
สร้างยูนิต Hammer จำนวน 1 ตัว	0.02954290
สร้างยูนิต Hammer จำนวน 2 ตัว	0.02954290
สร้างยูนิต Hammer จำนวน 3 ตัว	0.02954290
สร้างยูนิต Hammer จำนวน 4 ตัว	0.02954290
สร้างยูนิต Rocko จำนวน 1 ตัว	0.95967150
สร้างยูนิต Rocko จำนวน 2 ตัว	2.42954290
สร้างยูนิต Rocko จำนวน 3 ตัว	7.51041126
สร้างยูนิต Rocko จำนวน 4 ตัว	9.02954290
สร้างยูนิต Jeffy จำนวน 1 ตัว	0.02954290
สร้างยูนิต Jeffy จำนวน 2 ตัว	0.02674985
สร้างยูนิต Jeffy จำนวน 3 ตัว	0.02954290
สร้างยูนิต Jeffy จำนวน 4 ตัว	8.77848816
สร้างยูนิต Jethro จำนวน 1 ตัว	1.18654476
สร้างยูนิต Jethro จำนวน 2 ตัว	0.02954290
สร้างยูนิต Jethro จำนวน 3 ตัว	0.02954290
สร้างยูนิต Jethro จำนวน 4 ตัว	8.02954290
สร้างยูนิต Stumpy จำนวน 1 ตัว	0.02954290
สร้างยูนิต Stumpy จำนวน 2 ตัว	0.02954290
สร้างยูนิต Stumpy จำนวน 3 ตัว	2.86608791
สร้างยูนิต Stumpy จำนวน 4 ตัว	8.78169584
สร้างยูนิต Samson จำนวน 1 ตัว	1.02954290
สร้างยูนิต Samson จำนวน 2 ตัว	1.02954290
สร้างยูนิต Samson จำนวน 3 ตัว	4.02954290
สร้างยูนิต Samson จำนวน 4 ตัว	0.02954290
สร้างยูนิต Zeus จำนวน 1 ตัว	0.12954290

รูปที่ ข.1 กฎย่อยในกฎพื้นฐานของคู่มือปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น

กลยุทธ์	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.77483761
สร้างยูนิต Warrior จำนวน 1 ตัว	0.02377844
สร้างยูนิต Warrior จำนวน 2 ตัว	1.26971579
สร้างยูนิต Warrior จำนวน 3 ตัว	0.02377844
สร้างยูนิต Warrior จำนวน 4 ตัว	0.02377844
สร้างยูนิต Janus จำนวน 1 ตัว	1.35507584
สร้างยูนิต Janus จำนวน 2 ตัว	0.02377844
สร้างยูนิต Janus จำนวน 3 ตัว	2.47875941
สร้างยูนิต Janus จำนวน 4 ตัว	5.37264109
สร้างยูนิต Hammer จำนวน 1 ตัว	0.24459553
สร้างยูนิต Hammer จำนวน 2 ตัว	0.17849314
สร้างยูนิต Hammer จำนวน 3 ตัว	1.47456038
สร้างยูนิต Hammer จำนวน 4 ตัว	0.18824804
สร้างยูนิต Rocko จำนวน 1 ตัว	1.02298355
สร้างยูนิต Rocko จำนวน 2 ตัว	0.32591724
สร้างยูนิต Rocko จำนวน 3 ตัว	2.55527496
สร้างยูนิต Rocko จำนวน 4 ตัว	7.80768251
สร้างยูนิต Jeffy จำนวน 1 ตัว	0.17395258
สร้างยูนิต Jeffy จำนวน 2 ตัว	0.37949538
สร้างยูนิต Jeffy จำนวน 3 ตัว	1.22545397
สร้างยูนิต Jeffy จำนวน 4 ตัว	0.79757011
สร้างยูนิต Jethro จำนวน 1 ตัว	0.02377844
สร้างยูนิต Jethro จำนวน 2 ตัว	1.39380503
สร้างยูนิต Jethro จำนวน 3 ตัว	1.25748253
สร้างยูนิต Jethro จำนวน 4 ตัว	5.74354088
สร้างยูนิต Stumpy จำนวน 1 ตัว	0.02377844
สร้างยูนิต Stumpy จำนวน 2 ตัว	0.19628799
สร้างยูนิต Stumpy จำนวน 3 ตัว	0.04178886
สร้างยูนิต Stumpy จำนวน 4 ตัว	8.47349250
สร้างยูนิต Samson จำนวน 1 ตัว	0.02377844
สร้างยูนิต Samson จำนวน 2 ตัว	1.67827773
สร้างยูนิต Samson จำนวน 3 ตัว	0.02377844
สร้างยูนิต Samson จำนวน 4 ตัว	0.02377844
สร้างยูนิต Zeus จำนวน 1 ตัว	0.02377844

รูปที่ ข.2 กลยุทธ์ในกฎพื้นฐานของคู่ปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) และ
ค่าน้ำหนักเริ่มต้น

กฏย่อ	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.03954947
สร้างยูนิต Warrior จำนวน 1 ตัว	0.03954947
สร้างยูนิต Warrior จำนวน 2 ตัว	0.03954947
สร้างยูนิต Warrior จำนวน 3 ตัว	0.03954947
สร้างยูนิต Warrior จำนวน 4 ตัว	0.03954947
สร้างยูนิต Janus จำนวน 1 ตัว	0.03954947
สร้างยูนิต Janus จำนวน 2 ตัว	0.03954947
สร้างยูนิต Janus จำนวน 3 ตัว	0.03954947
สร้างยูนิต Janus จำนวน 4 ตัว	4.00841761
สร้างยูนิต Hammer จำนวน 1 ตัว	0.03954947
สร้างยูนิต Hammer จำนวน 2 ตัว	2.70067120
สร้างยูนิต Hammer จำนวน 3 ตัว	4.11648512
สร้างยูนิต Hammer จำนวน 4 ตัว	0.03954947
สร้างยูนิต Rocko จำนวน 1 ตัว	1.03954947
สร้างยูนิต Rocko จำนวน 2 ตัว	0.03954947
สร้างยูนิต Rocko จำนวน 3 ตัว	3.34684610
สร้างยูนิต Rocko จำนวน 4 ตัว	0.03954947
สร้างยูนิต Jeffy จำนวน 1 ตัว	0.03954947
สร้างยูนิต Jeffy จำนวน 2 ตัว	3.98661518
สร้างยูนิต Jeffy จำนวน 3 ตัว	0.03954947
สร้างยูนิต Jeffy จำนวน 4 ตัว	5.00786448
สร้างยูนิต Jethro จำนวน 1 ตัว	0.03954947
สร้างยูนิต Jethro จำนวน 2 ตัว	0.03954947
สร้างยูนิต Jethro จำนวน 3 ตัว	4.09727097
สร้างยูนิต Jethro จำนวน 4 ตัว	4.91798687
สร้างยูนิต Stumpy จำนวน 1 ตัว	0.86865830
สร้างยูนิต Stumpy จำนวน 2 ตัว	0.03954947
สร้างยูนิต Stumpy จำนวน 3 ตัว	0.03954947
สร้างยูนิต Stumpy จำนวน 4 ตัว	0.03954947
สร้างยูนิต Samson จำนวน 1 ตัว	0.03954947
สร้างยูนิต Samson จำนวน 2 ตัว	0.03954947
สร้างยูนิต Samson จำนวน 3 ตัว	0.03954947
สร้างยูนิต Samson จำนวน 4 ตัว	0.03954947
สร้างยูนิต Zeus จำนวน 1 ตัว	0.03954947

รูปที่ ข.3 กฏย่อในกฎพื้นฐานของคูปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.08289160
สร้างยูนิต Jethro จำนวน 1 ตัว	7.47576940
สร้างยูนิต Peewee จำนวน 1 ตัว	0.06482800
สร้างยูนิต Hammer จำนวน 1 ตัว	7.00828916
สร้างยูนิต Flea จำนวน 1 ตัว	3.00828916
สร้างยูนิต Rocko จำนวน 1 ตัว	0.00941249

รูปที่ ข.4 กฎย่อยในกฎพื้นฐานของกลุ่มปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.13043070
สร้างยูนิต Jethro จำนวน 1 ตัว	0.03180604
สร้างยูนิต Peewee จำนวน 1 ตัว	1.00200069
สร้างยูนิต Hammer จำนวน 1 ตัว	0.95408207
สร้างยูนิต Flea จำนวน 1 ตัว	3.03180604
สร้างยูนิต Rocko จำนวน 1 ตัว	0.97709376

รูปที่ ข.5 กฎย่อยในกฎพื้นฐานของกลุ่มปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 2 (Barren) และค่าน้ำหนักเริ่มต้น

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.02280381
สร้างยูนิต Jethro จำนวน 1 ตัว	0.18289160
สร้างยูนิต Peewee จำนวน 1 ตัว	0.02280381
สร้างยูนิต Hammer จำนวน 1 ตัว	7.14020300
สร้างยูนิต Flea จำนวน 1 ตัว	1.06482800
สร้างยูนิต Rocko จำนวน 1 ตัว	0.02280381

รูปที่ ข.6 กฎย่อยในกฎพื้นฐานของกลุ่มปัญญาประดิษฐ์ที่สร้างขึ้น-AAI ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.09703416
สร้างยูนิต Flea จำนวน 1 ตัว	0.09703416
สร้างยูนิต Flea จำนวน 2 ตัว	4.48219341
สร้างยูนิต Flea จำนวน 3 ตัว	0.01282721
สร้างยูนิต Rocko จำนวน 1 ตัว	0.09703416
สร้างยูนิต Rocko จำนวน 2 ตัว	2.39222980
สร้างยูนิต Rocko จำนวน 3 ตัว	0.00489225
สร้างยูนิต Peewee จำนวน 1 ตัว	1.39703416
สร้างยูนิต Peewee จำนวน 2 ตัว	3.19703416
สร้างยูนิต Peewee จำนวน 3 ตัว	7.02575712
สร้างยูนิต Jeffy จำนวน 1 ตัว	0.49703416
สร้างยูนิต Jeffy จำนวน 2 ตัว	1.85244875
สร้างยูนิต Jeffy จำนวน 3 ตัว	0.09703416
สร้างยูนิต Flash จำนวน 1 ตัว	3.09703416
สร้างยูนิต Flash จำนวน 2 ตัว	4.29703416
สร้างยูนิต Flash จำนวน 3 ตัว	0.09703416

รูปที่ ข.7 กฎย่อยในกฎพื้นฐานของคูปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 1 (SmallDivide) และค่าน้ำหนักเริ่มต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	1.36968791
สร้างยูนิต Flea จำนวน 1 ตัว	0.01532245
สร้างยูนิต Flea จำนวน 2 ตัว	2.32524228
สร้างยูนิต Flea จำนวน 3 ตัว	0.01532245
สร้างยูนิต Rocko จำนวน 1 ตัว	1.95425487
สร้างยูนิต Rocko จำนวน 2 ตัว	0.84413195
สร้างยูนิต Rocko จำนวน 3 ตัว	2.49138117
สร้างยูนิต Peewee จำนวน 1 ตัว	2.08736682
สร้างยูนิต Peewee จำนวน 2 ตัว	0.01532245
สร้างยูนิต Peewee จำนวน 3 ตัว	4.24106526
สร้างยูนิต Jeffy จำนวน 1 ตัว	0.01532245
สร้างยูนิต Jeffy จำนวน 2 ตัว	0.01532245
สร้างยูนิต Jeffy จำนวน 3 ตัว	0.01532245
สร้างยูนิต Flash จำนวน 1 ตัว	1.68642187
สร้างยูนิต Flash จำนวน 2 ตัว	0.01532245
สร้างยูนิต Flash จำนวน 3 ตัว	0.01532245

รูปที่ ข.8 กฎย่อยในกฎพื้นฐานของคูปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 2 (Barren) และ
ค่าน้ำหนักเริ่มต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

กฎย่อย	ค่าน้ำหนักเริ่มต้น
ไม่สร้างยูนิต	0.05016138
สร้างยูนิต Flea จำนวน 1 ตัว	0.00126434
สร้างยูนิต Flea จำนวน 2 ตัว	0.00126434
สร้างยูนิต Flea จำนวน 3 ตัว	2.70148468
สร้างยูนิต Rocko จำนวน 1 ตัว	0.00126434
สร้างยูนิต Rocko จำนวน 2 ตัว	0.00126434
สร้างยูนิต Rocko จำนวน 3 ตัว	5.88999009
สร้างยูนิต Peewee จำนวน 1 ตัว	1.77027345
สร้างยูนิต Peewee จำนวน 2 ตัว	0.00126434
สร้างยูนิต Peewee จำนวน 3 ตัว	1.70433283
สร้างยูนิต Jeffy จำนวน 1 ตัว	1.39615655
สร้างยูนิต Jeffy จำนวน 2 ตัว	3.32094049
สร้างยูนิต Jeffy จำนวน 3 ตัว	0.00126434
สร้างยูนิต Flash จำนวน 1 ตัว	0.00126434
สร้างยูนิต Flash จำนวน 2 ตัว	2.78496456
สร้างยูนิต Flash จำนวน 3 ตัว	1.47317600

รูปที่ ข.9 กฎย่อยในกฎพื้นฐานของคูปัญญาประดิษฐ์ที่สร้างขึ้น-RAI ในแผนที่ 3 (Tomb Stone Desert V2) และค่าน้ำหนักเริ่มต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ในภาคผนวก ค นี้จะอธิบายวิธีการเล่นเกมสปริง โดยในการเล่นเกมสปริงผู้เล่นแต่ละคนจะเริ่มจากยูนิต Commander ดังรูปที่ ค.1 ยูนิต Commander จะสามารถสร้างสิ่งก่อสร้างพื้นฐานของเกมได้

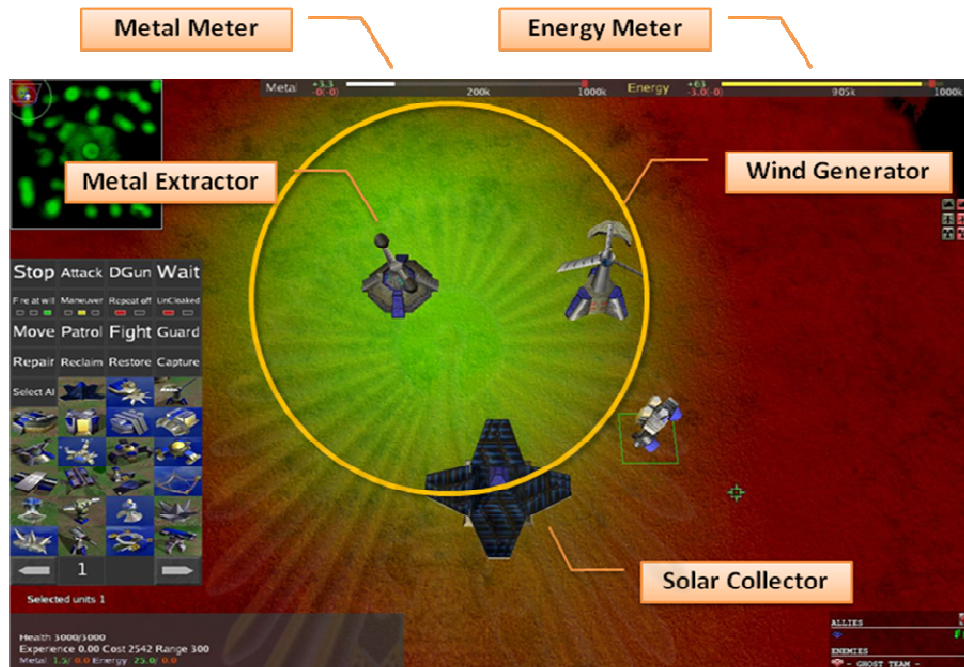


รูปที่ ค.1 Commander

ซึ่งการสร้างสิ่งก่อสร้างและยูนิตต่าง ๆ นั้น จะต้องใช้ทรัพยากรในการสร้าง ทรัพยากรในเกมสปริงแบ่งออกเป็น 2 ประเภท คือ

1. พลังงาน (Energy) สิ่งก่อสร้างที่สามารถผลิตพลังงานได้ มี 4 ประเภท คือ Wind Generator, Solar Collector, Geothermal Power Plant, Fusion Reactor แต่ละชนิดจะผลิตพลังงานได้ไม่เท่ากัน โดยสิ่งก่อสร้างที่สามารถผลิตพลังงานได้มากจะสามารถสร้างได้ต่อเมื่อสร้างสิ่งก่อสร้างที่จำเป็นในการสร้างสิ่งก่อสร้างนั้นๆ ก่อน
2. โลหะ (Metal) สิ่งก่อสร้างที่สามารถผลิตโลหะได้ คือ Metal Extractor โดยเราจะต้องสร้าง Metal Extractor ในบริเวณที่มีโลหะในเกม หรือ เราสามารถเปลี่ยนพลังงานเป็นโลหะได้โดยใช้ Metal Maker

จากรูปที่ ค.2 Wind Generator และ Solar Collector ทำหน้าที่ผลิตพลังงาน ส่วน Metal Extractor ทำหน้าที่ผลิตโลหะ โดยเราจะสร้างในบริเวณที่มีโลหะในเกม (ภายในวงกลม)



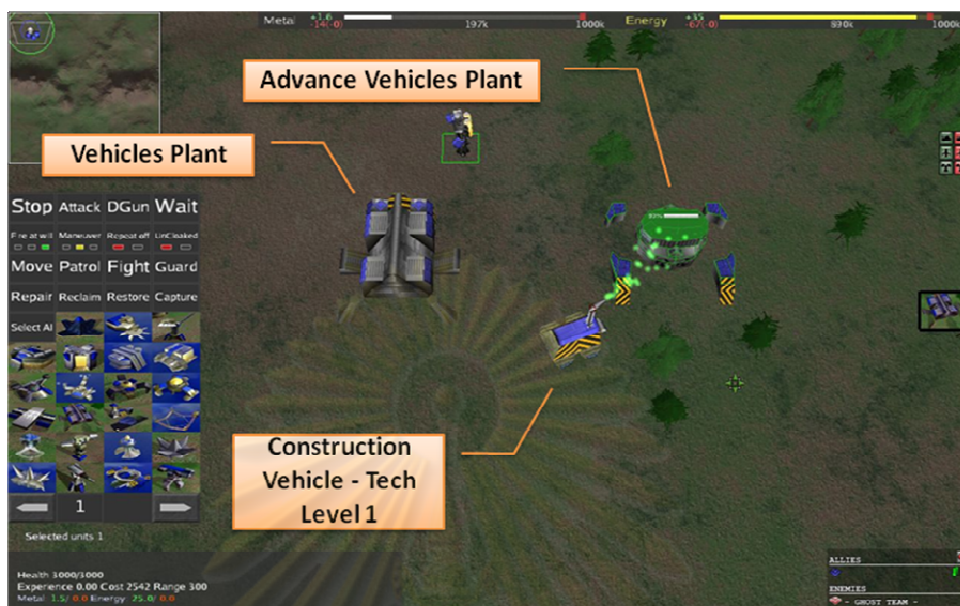
รูปที่ ค.2 การเก็บเกี่ยวทรัพยากรในเกม

ยูนิตที่ใช้ในการรบแบ่งเป็น 4 ประเภท คือ

1. K-bots คือ ยูนิตหุ่นยนต์ สามารถเดินผ่านพื้นที่ ที่มีความชันต่างกันได้ แบ่งออกเป็น 3 ระดับ
2. Vehicles คือ พาหนะ มีความเร็วและความแข็งแกร่งมากกว่ายูนิตหุ่นยนต์ แบ่งออกเป็น 3 ระดับ
3. Aircraft คือ เครื่องบิน มีความแข็งแกร่งแต่ต้องใช้ทรัพยากรในการสร้างสูง แบ่งออกเป็น 2 ระดับ
4. Ship คือ เรือ เคลื่อนที่ได้ในน้ำเท่านั้น มีอำนาจการยิงสูงแต่เคลื่อนที่ได้ช้า แบ่งออกเป็น 2 ระดับ
5. Hovercraft คือ ยานสะเทินน้ำสะเทินบก เคลื่อนที่ได้ทั้งในน้ำและบนบกแต่มีอำนาจการยิงต่ำ แบ่งออกเป็น 1 ระดับ

การสร้างยูนิตในการรบระดับถัดไป จะไม่ใช้วิธีการสร้างโดยตรง แต่จะทำการสร้างโดยผ่านยูนิตสำหรับสร้างสิ่งก่อสร้างเสียก่อน เช่น ถ้าต้องการสร้างโรงงานผลิต Vehicles ระดับ 2 จะต้องมี Vehicles Plant สำหรับผลิต Vehicle ระดับ 1 แล้วสั่งให้ Vehicles Plant สร้าง

Construction Vehicle - Tech Level 1 เพื่อสร้าง Advance Vehicles Plant สำหรับผลิต Vehicle ระดับ 2 อีกที ดังรูปที่ ค.3



รูปที่ ค.3 การผลิตสิ่งก่อสร้างระดับถัดไป

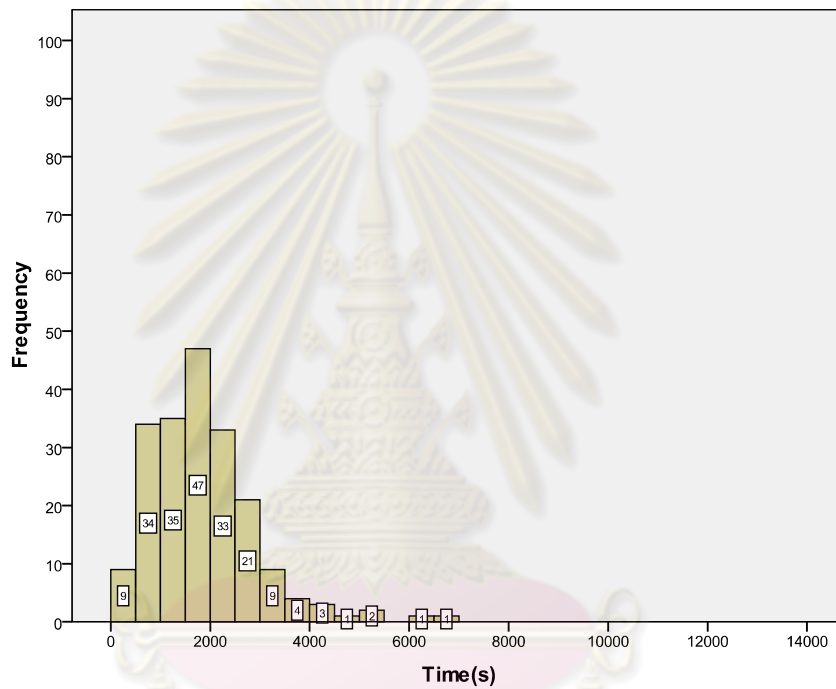
ในการเล่นเกมนั้นมีเป้าหมายคือทำลายยูนิต Commander ของฝ่ายตรงข้าม เมื่อผู้เล่นสร้างยูนิตที่ใช้ในการรบแล้ว ผู้เล่นจะต้องวางแผนการนำยูนิตนั้นไปใช้ในการทำลายยูนิต Commander ของฝ่ายตรงข้าม ดังรูปที่ ค.4 รวมทั้งบริหารการใช้ทรัพยากรไปด้วย เมื่อทำลายยูนิตคอมมานเดอร์ฝ่ายตรงข้ามได้แล้วจึงจะชนะในเกมนั้น



รูปที่ ค.4 การต่อสู้กันระหว่างสองฝ่าย [8]

ภาคผนวก ง

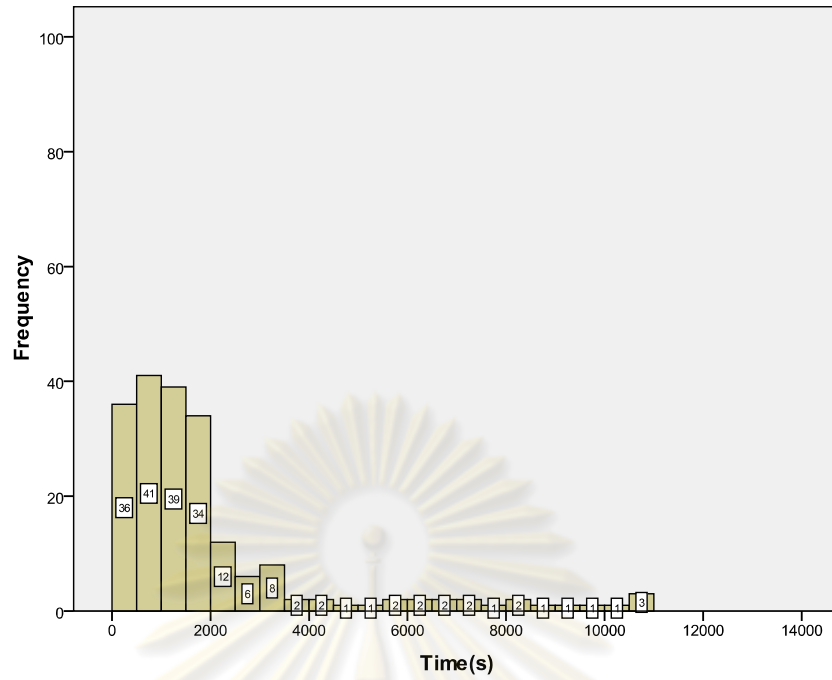
- รูปที่ ง.1 - รูปที่ ง.66 ฮิสโทแกรมของปัญญาประดิษฐ์แต่ละคู่ โดยแกนนอนแสดงเวลา (วินาที) และแกนตั้งแสดงจำนวนเกมที่อยู่ในอินเทอร์เน็ตภาคชั้นนั้น
- รูปที่ ง.67 - รูปที่ ง.132 กราฟแสดงผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์แต่ละคู่ กับ เวลา (วินาที) ของเกมนั้น โดย 0 หมายถึงปัญญาประดิษฐ์ฝ่ายแรกเป็นผู้ชนะ และ 1 หมายถึงปัญญาประดิษฐ์ฝ่ายหลังเป็นผู้ชนะ



รูปที่ ง.1 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 1 (SmallDivide) โดยวิธี

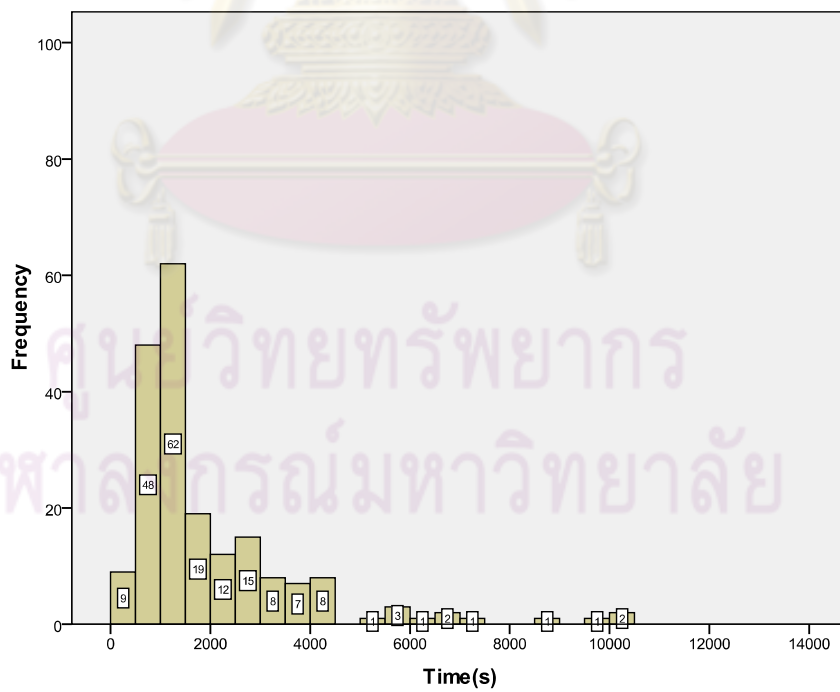
Fix 1 rule

ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

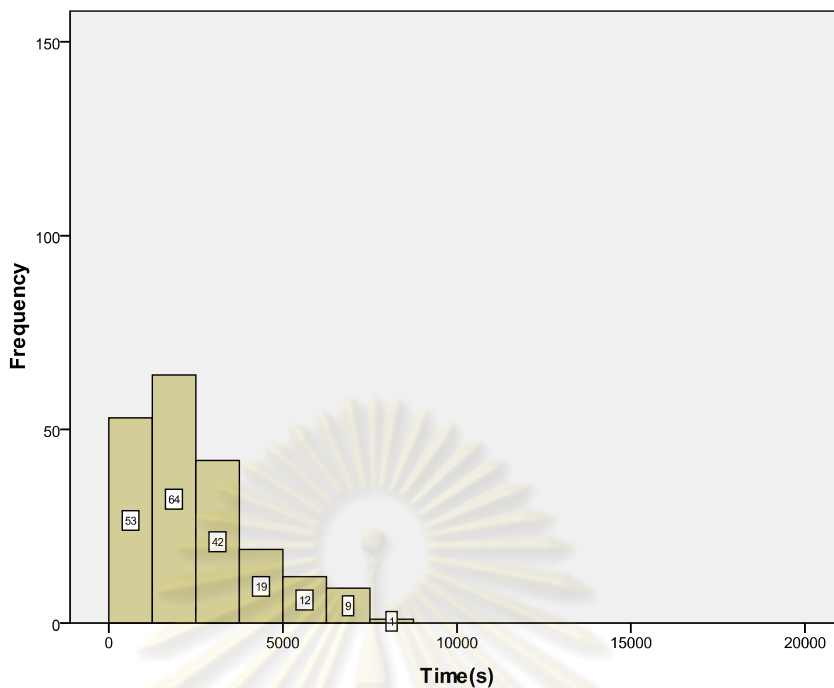


รูปที่ ง.2 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น-E323 ในแผ่นที่ 1 (SmallDivide) โดยวิธี

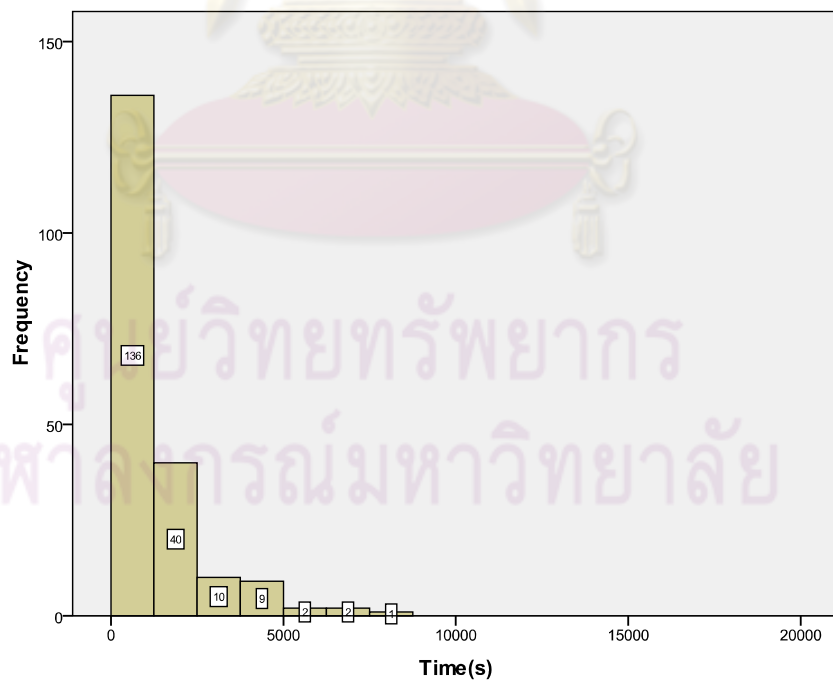
Change w_{inc} , w_{dec}



รูปที่ ง.3 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น-E323 ในแผ่นที่ 1 (SmallDivide) โดยวิธี 7-minute increment

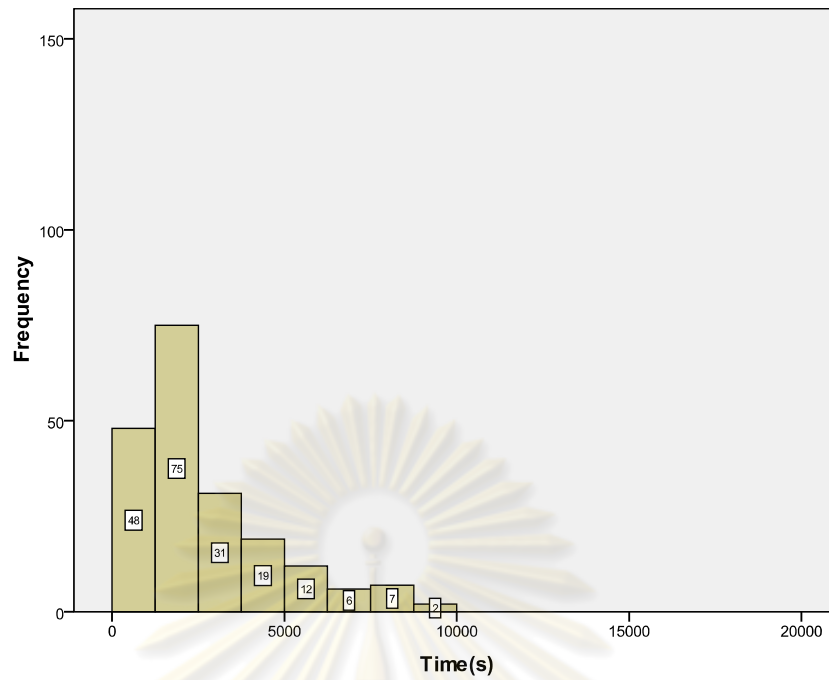


รูปที่ ง.6 ฮิสโทแกรมของปัญหาประติมากรรมที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule

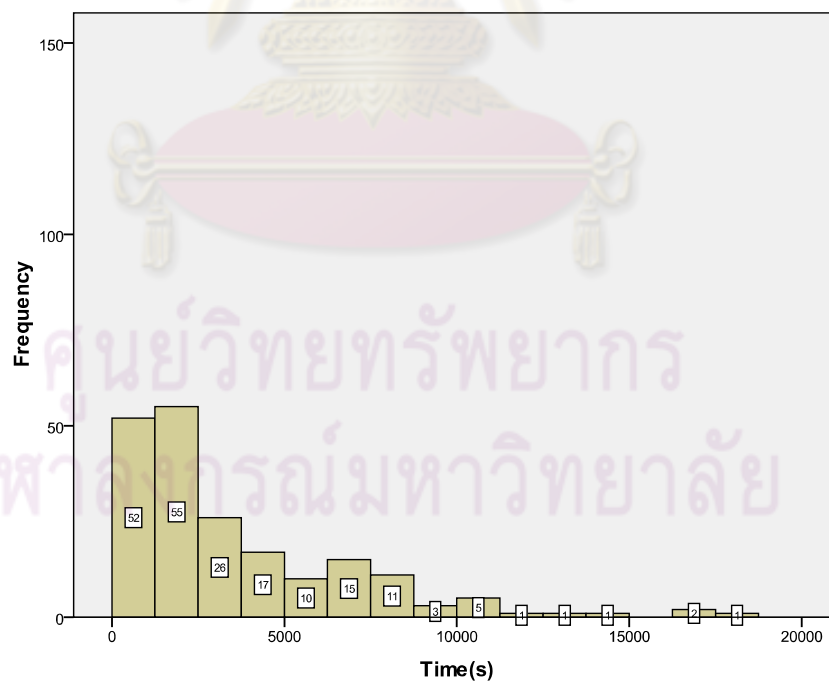


รูปที่ ง.7 ฮิสโทแกรมของปัญหาประติมากรรมที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Change

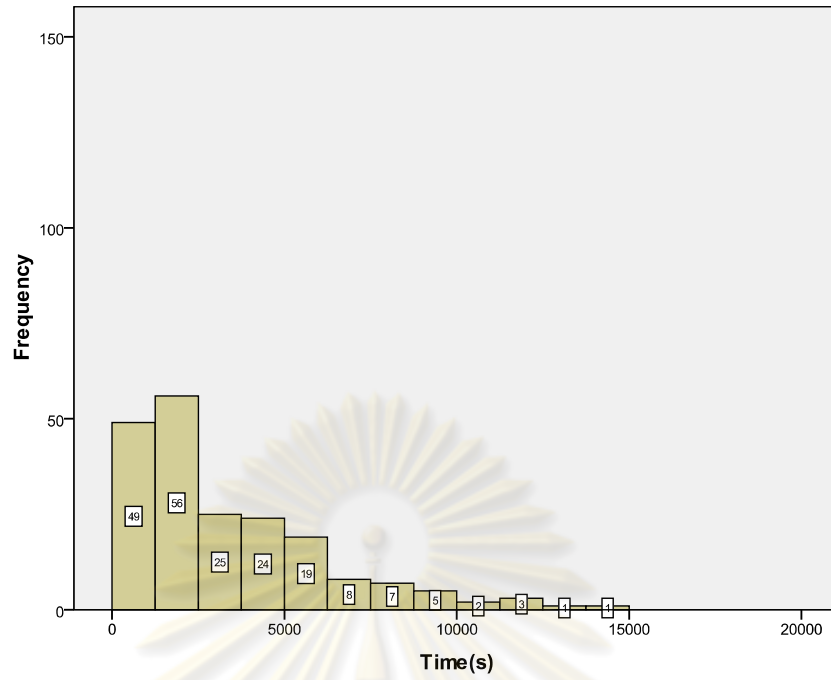
$$W_{inc}, W_{dec}$$



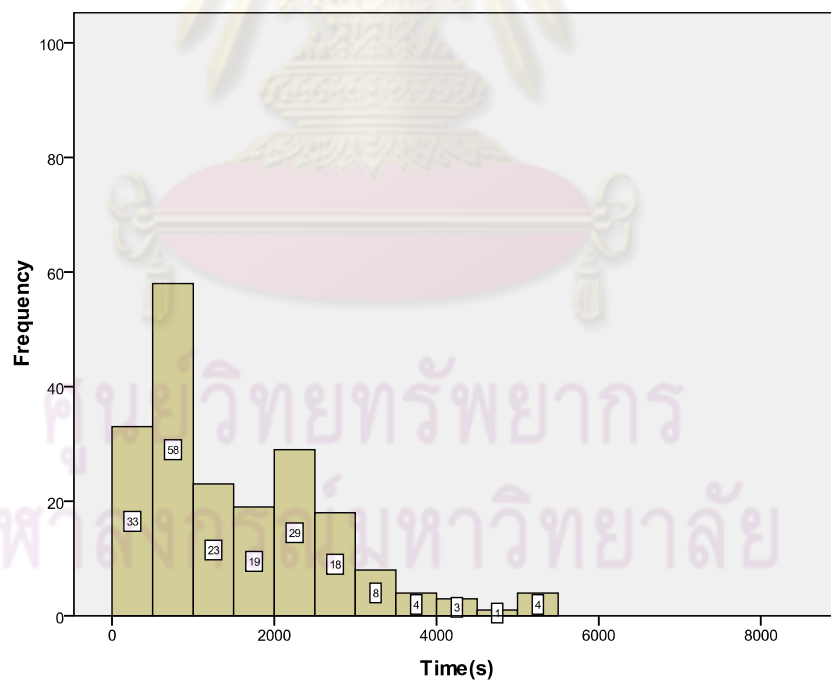
รูปที่ ง.8 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment



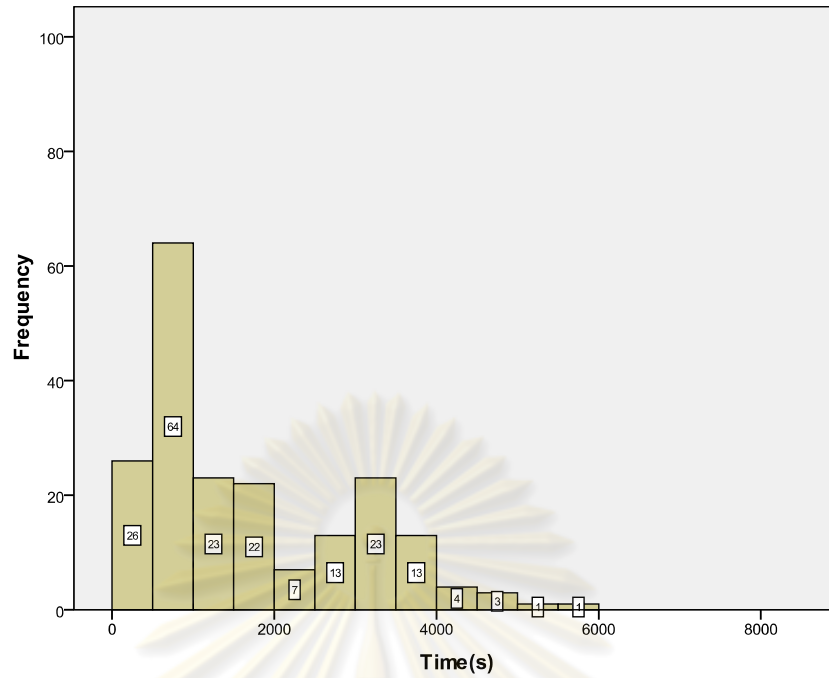
รูปที่ ง.9 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Change number of rules



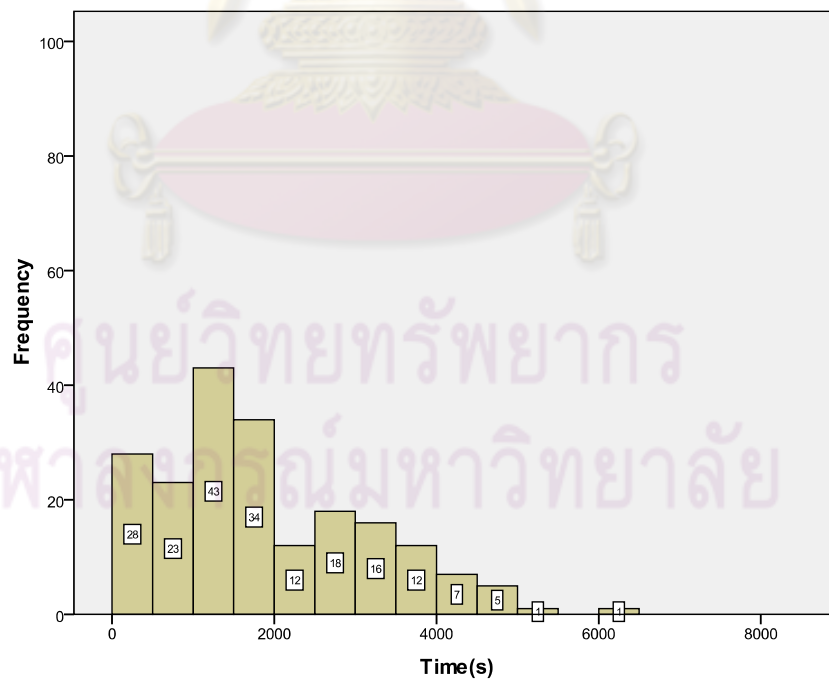
รูปที่ ง.10 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 2 (Barren) โดยวิธี Mix



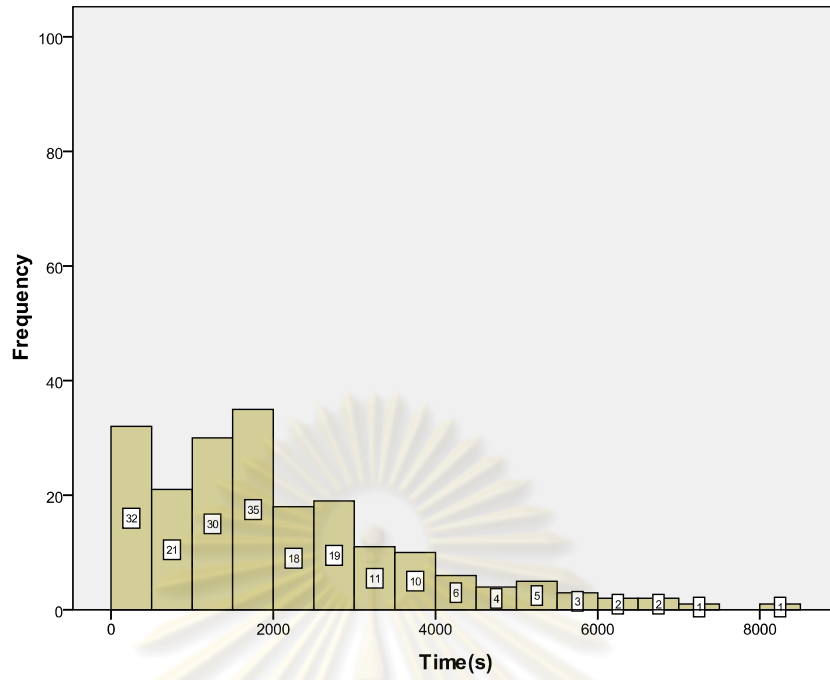
รูปที่ ง.11 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule



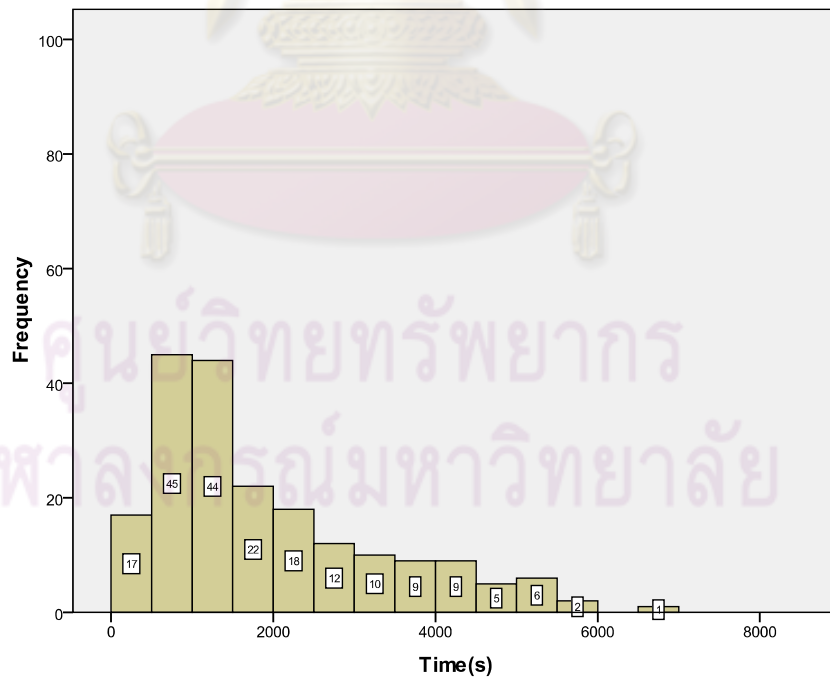
รูปที่ ง.12 ฮิสโทแกรมของปัญหาประติษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}



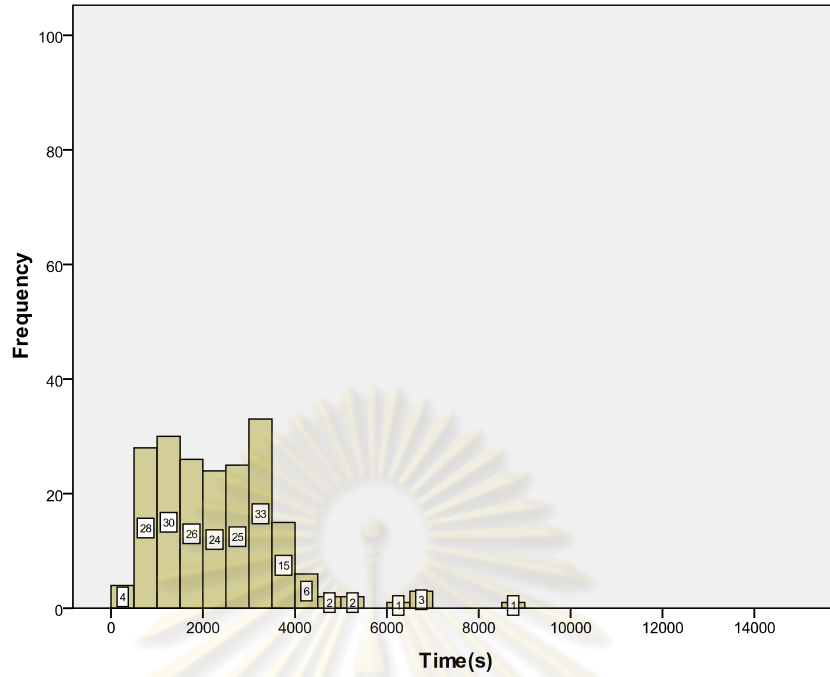
รูปที่ ง.13 ฮิสโทแกรมของปัญหาประติษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment



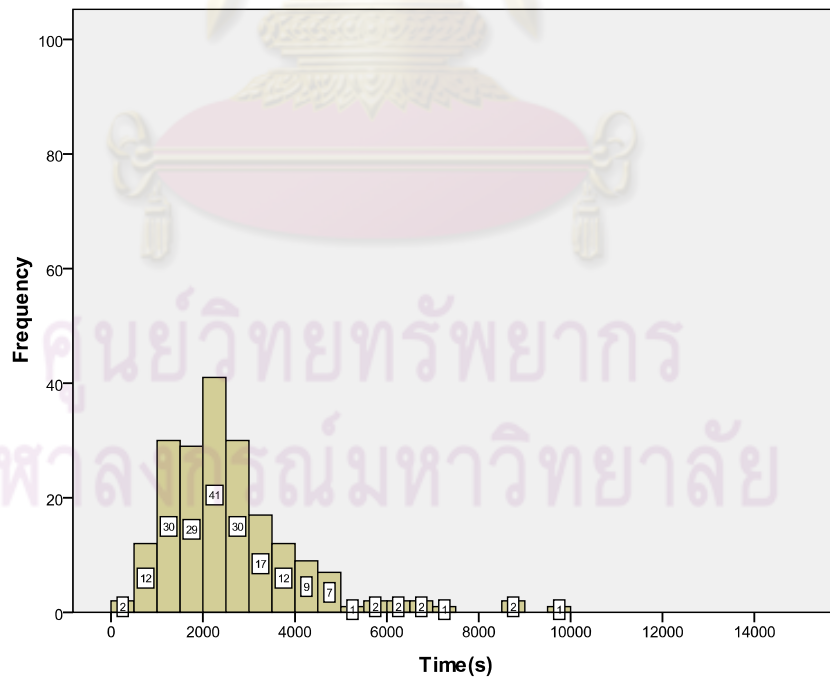
รูปที่ ง.14 ฮิสโทแกรมของปัญหาประติษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules



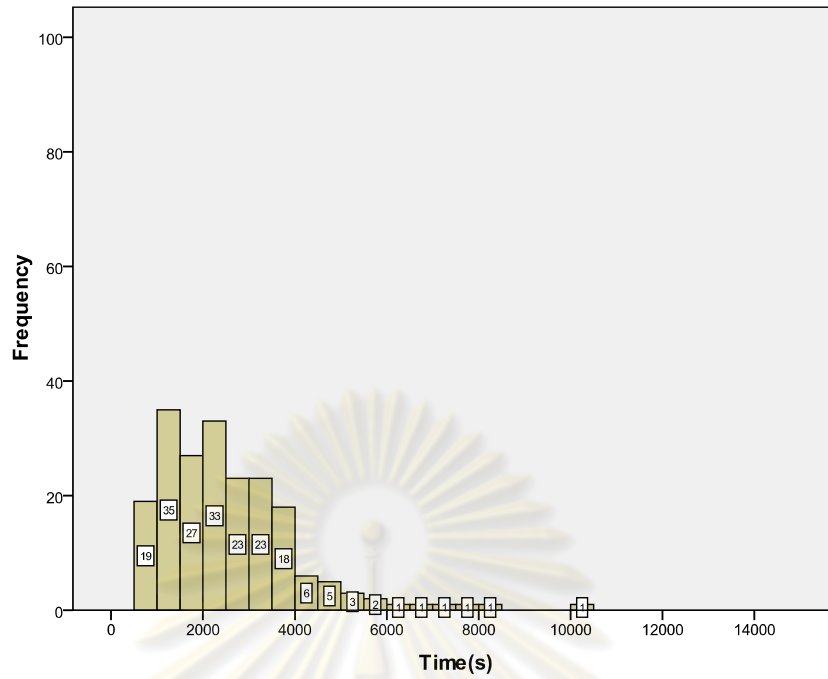
รูปที่ ง.15 ฮิสโทแกรมของปัญหาประติษฐ์ที่สร้างขึ้น-E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix



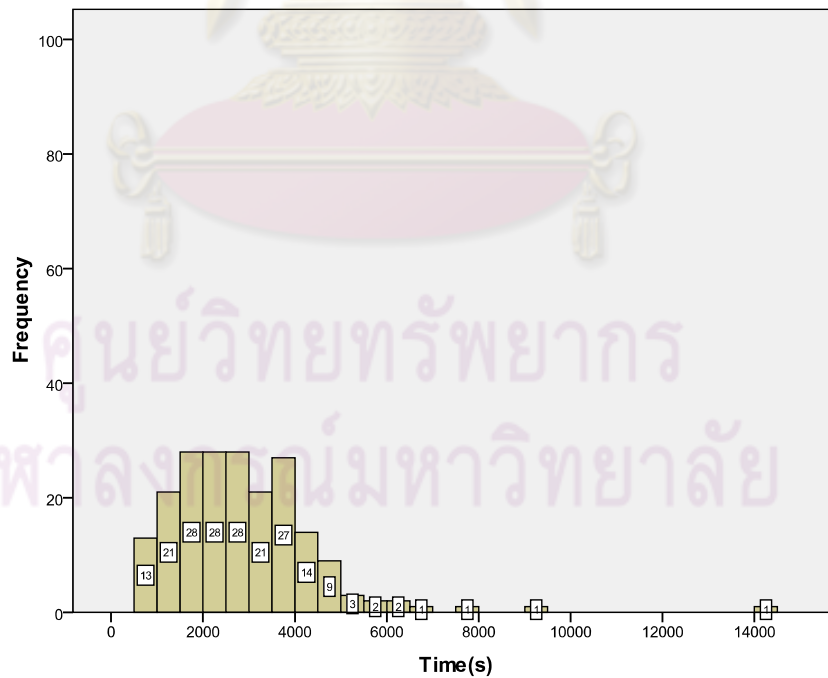
รูปที่ ง.16 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule



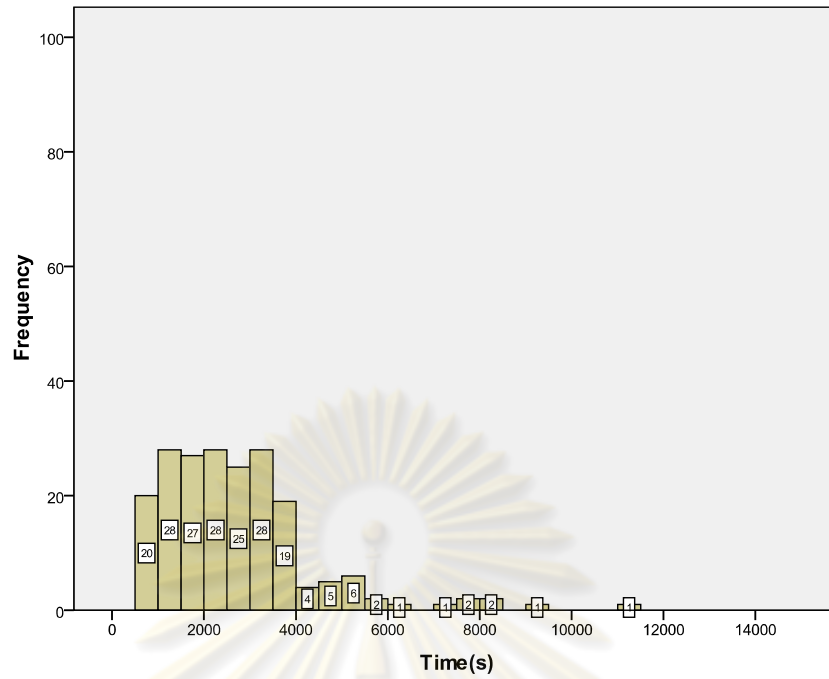
รูปที่ ง.17 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}



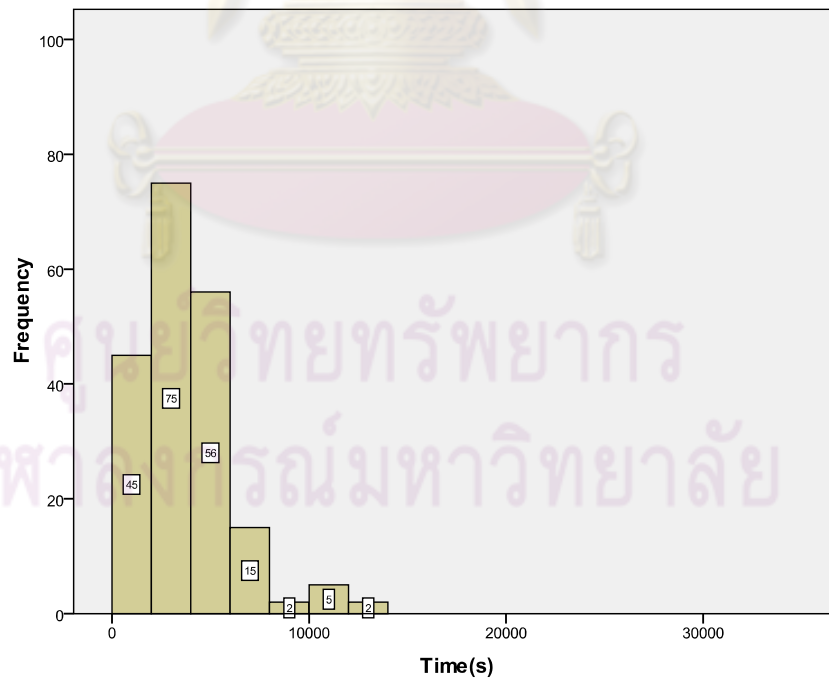
รูปที่ ง.18 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment



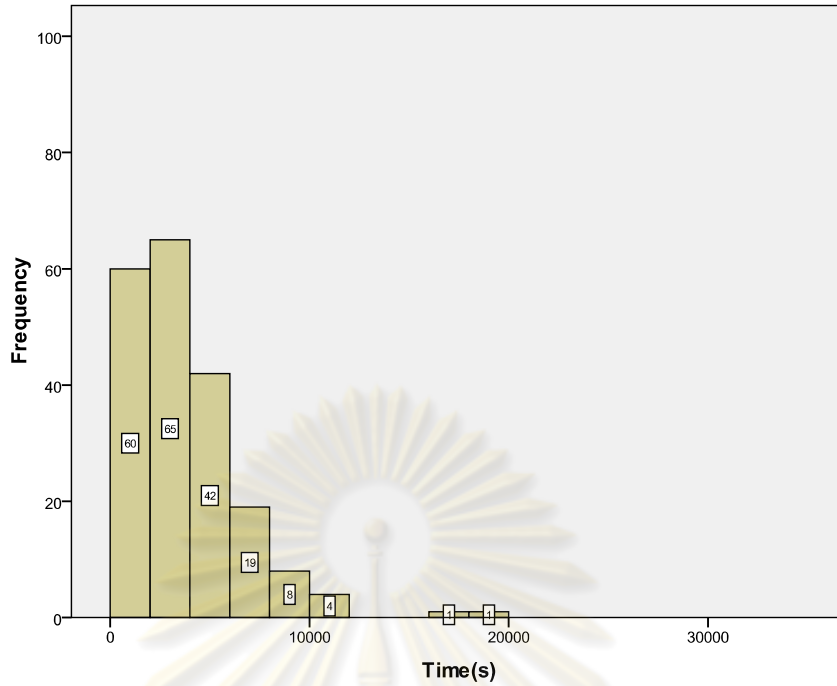
รูปที่ ง.19 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules



รูปที่ ง.20 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix

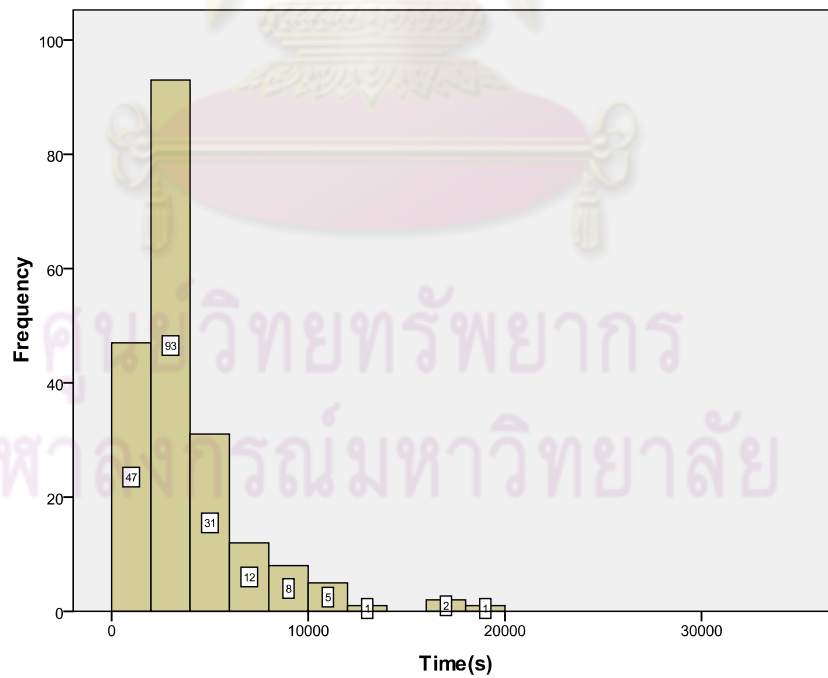


รูปที่ ง.21 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule

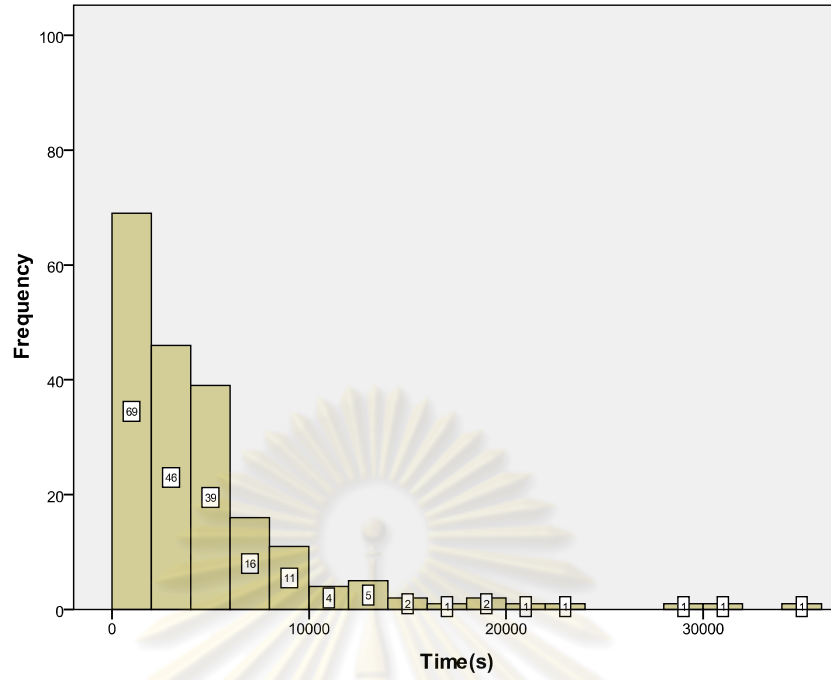


รูปที่ ง.22 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี

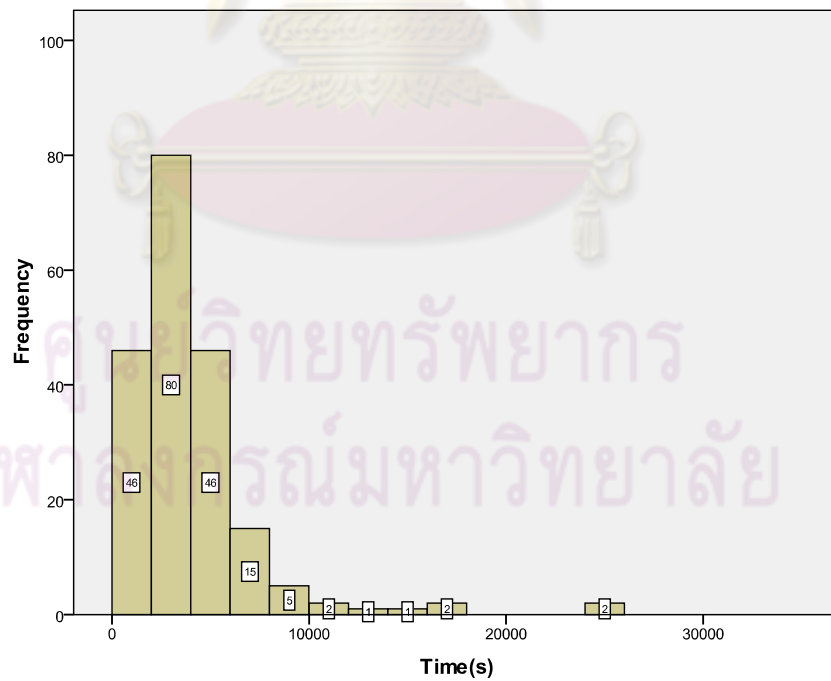
Change w_{inc} , w_{dec}



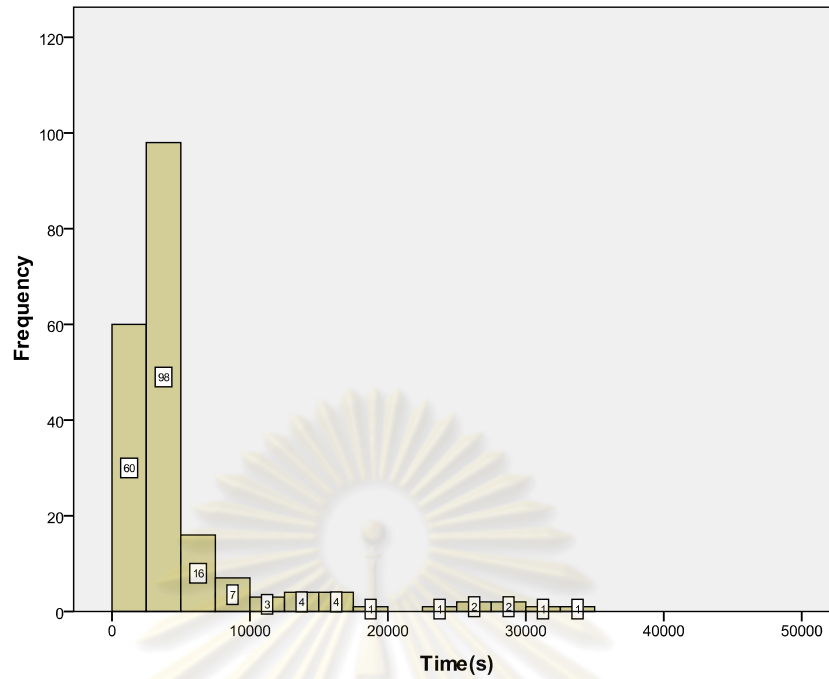
รูปที่ ง.23 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment



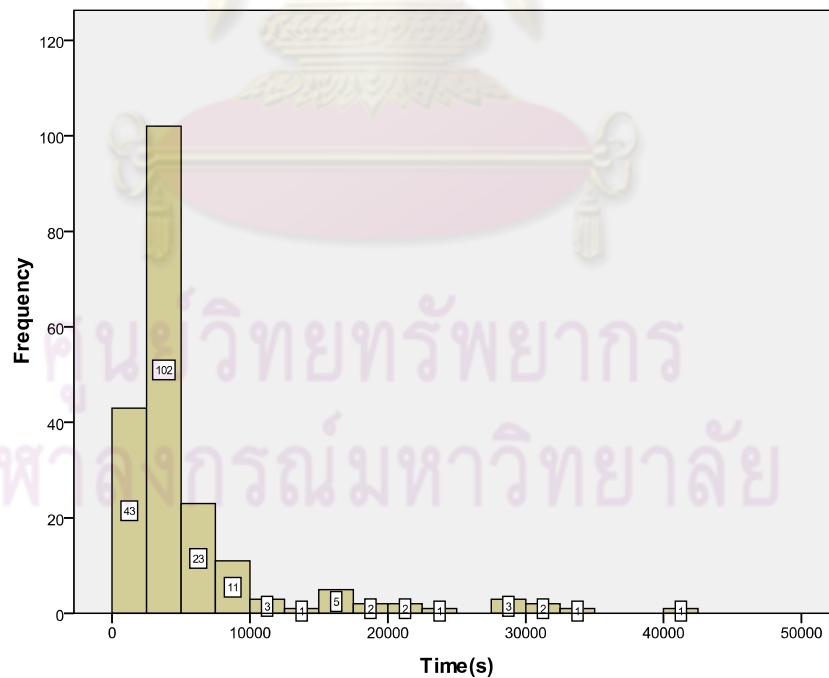
รูปที่ ง.24 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules



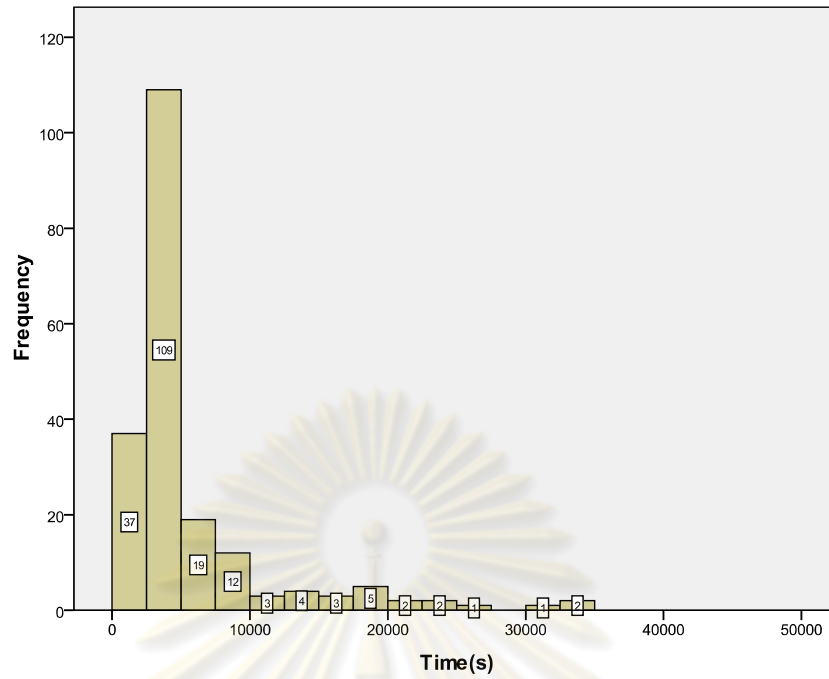
รูปที่ ง.25 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Mix



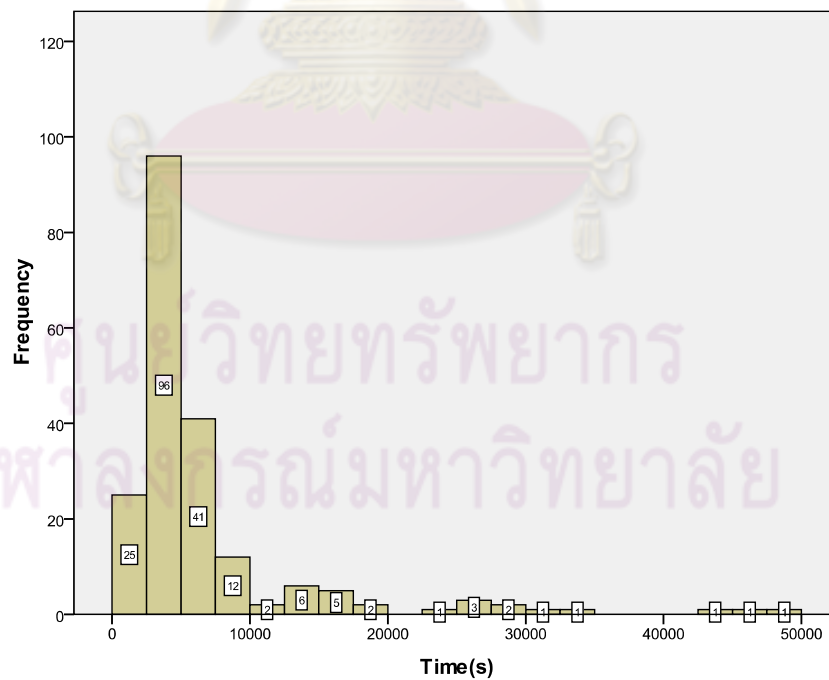
รูปที่ ง.26 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule



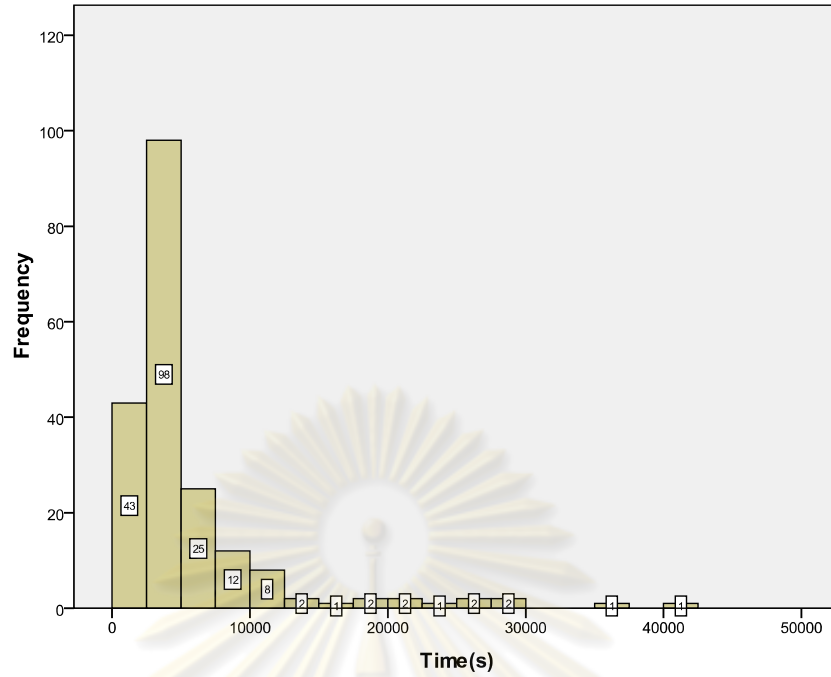
รูปที่ ง.27 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}



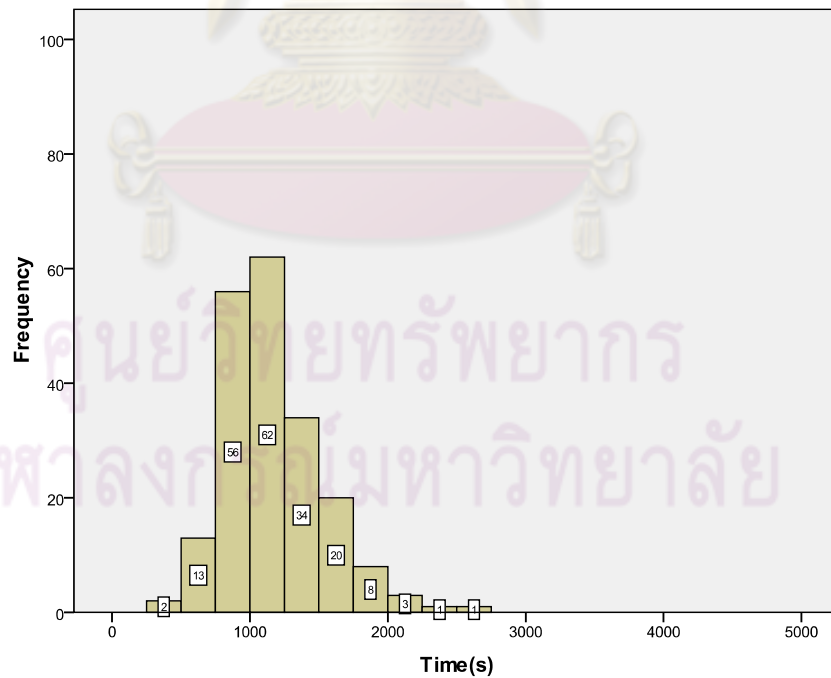
รูปที่ ง.28 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment



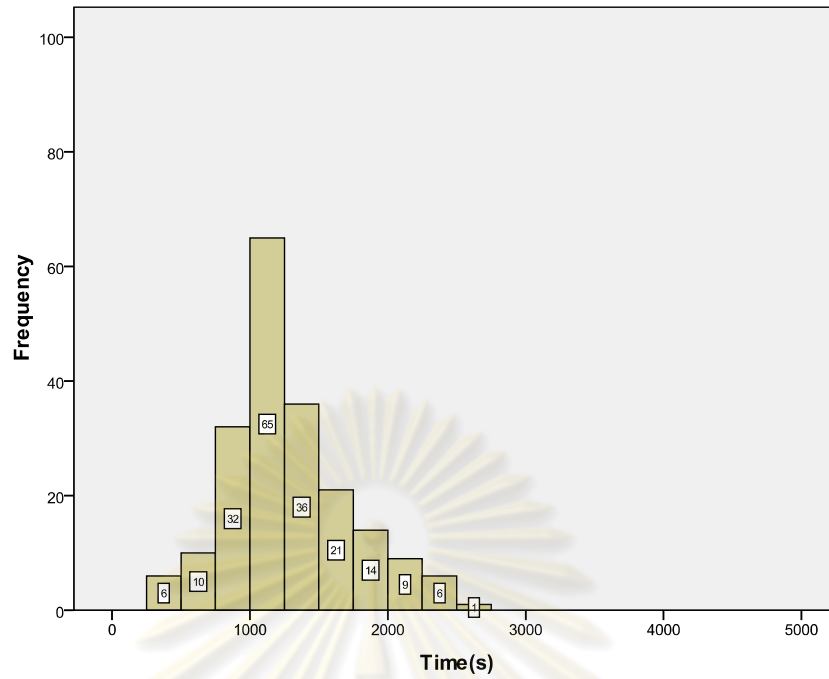
รูปที่ ง.29 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules



รูปที่ ง.30 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix

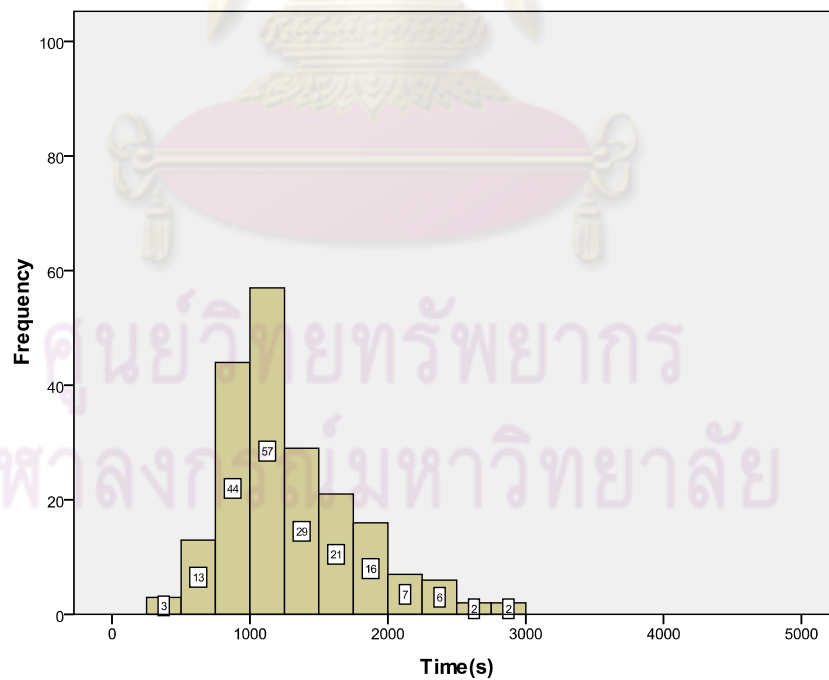


รูปที่ ง.31 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule



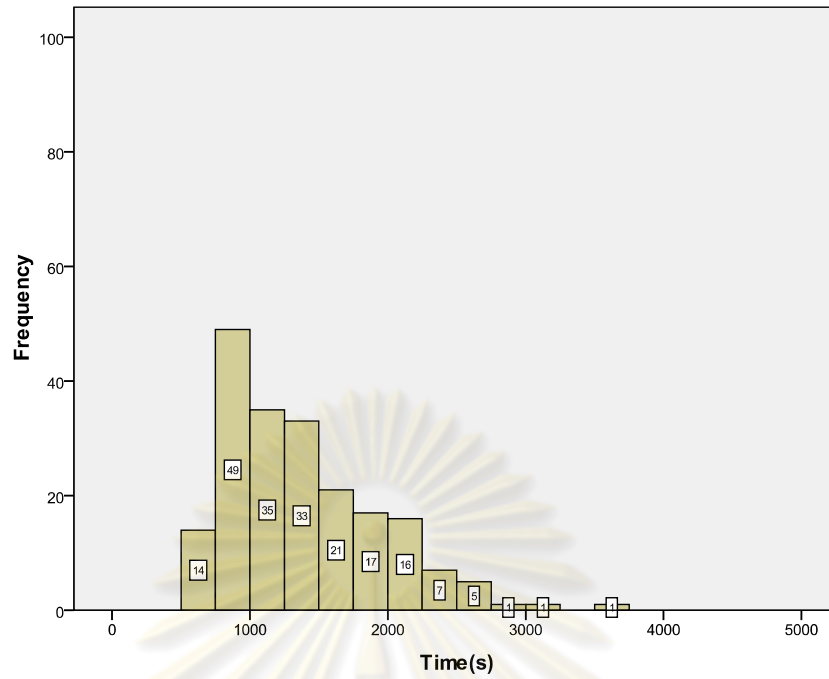
รูปที่ ง.32 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี

Change w_{inc} , w_{dec}

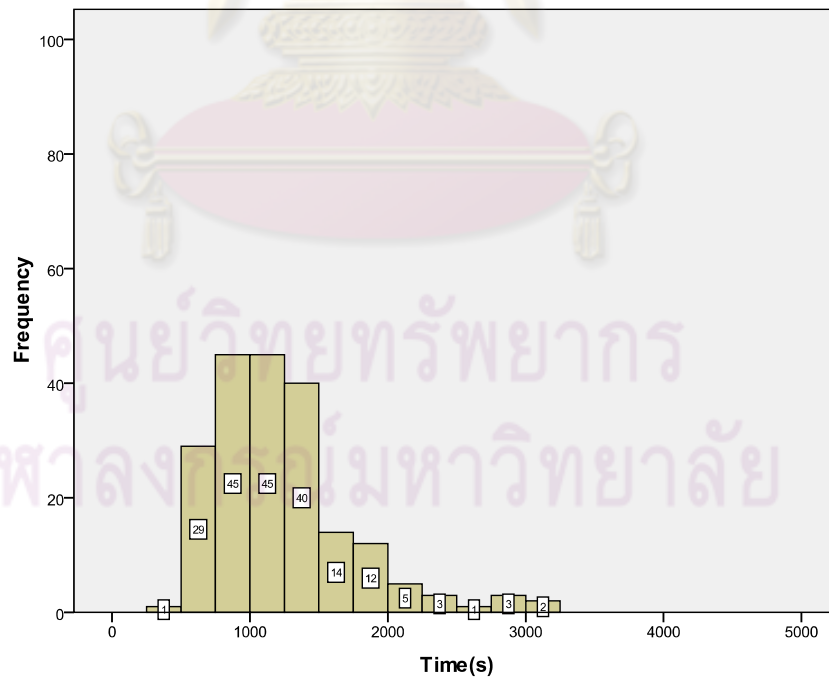


รูปที่ ง.33 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี

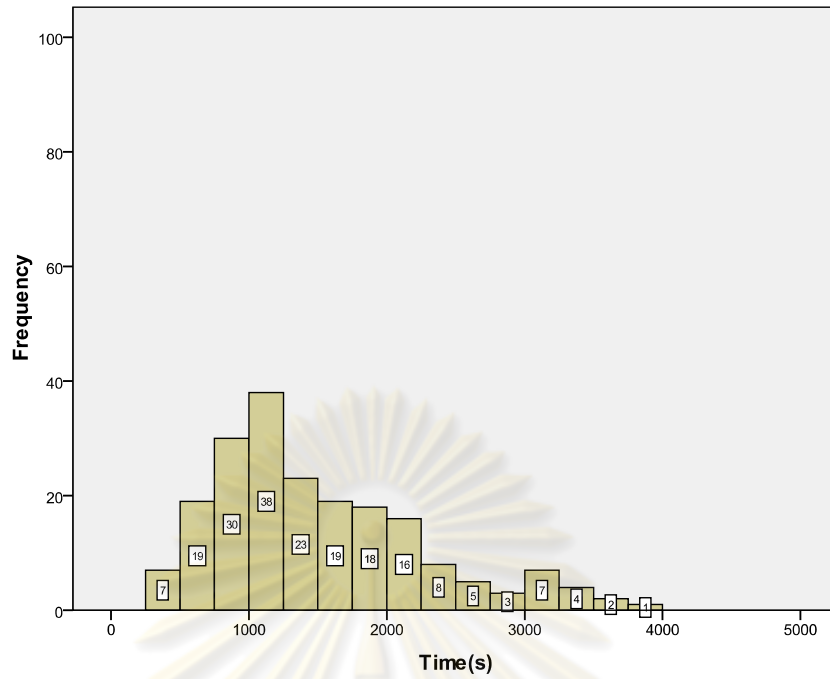
7-minute increment



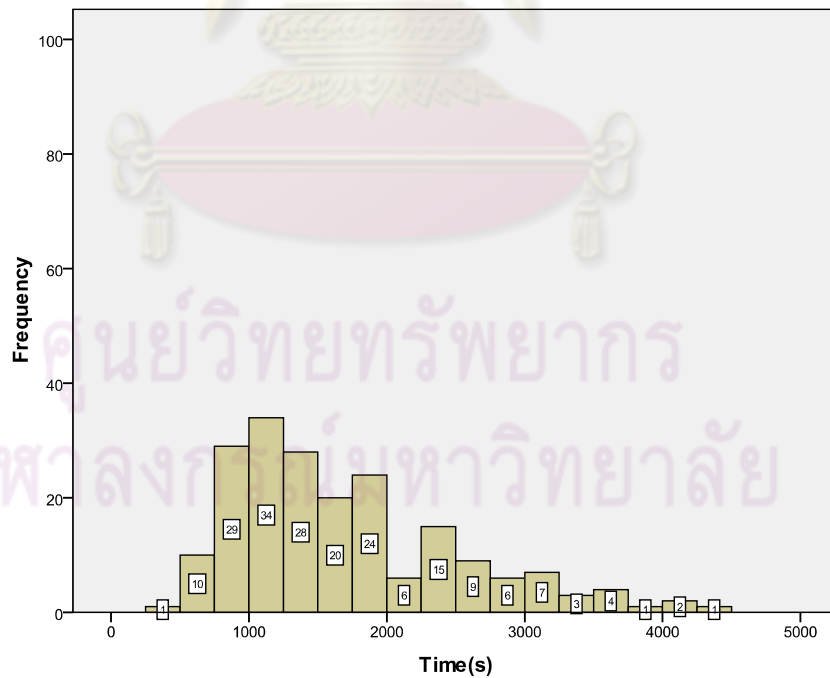
รูปที่ ง.34 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี
Change number of rules



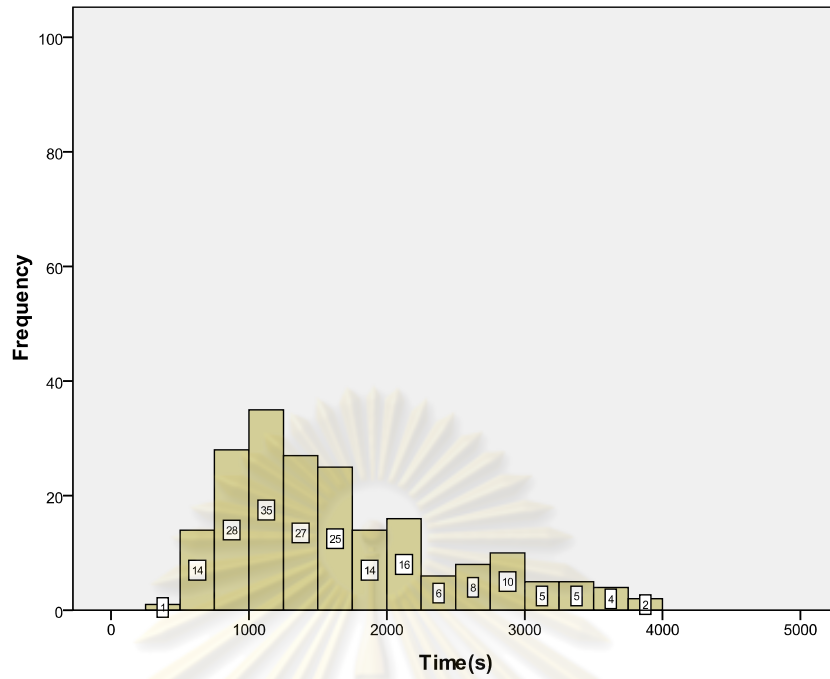
รูปที่ ง.35 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี
Mix



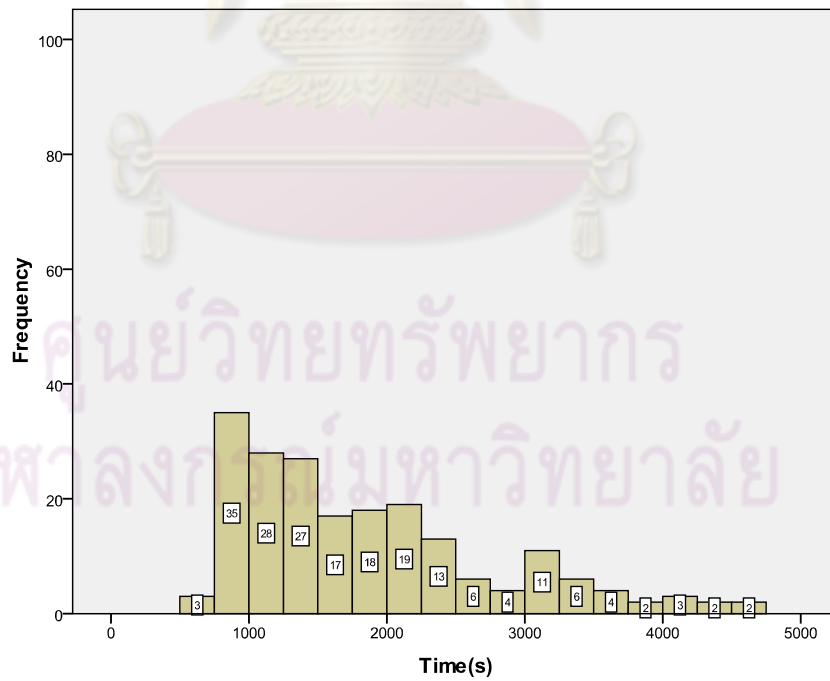
รูปที่ ง.36 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule



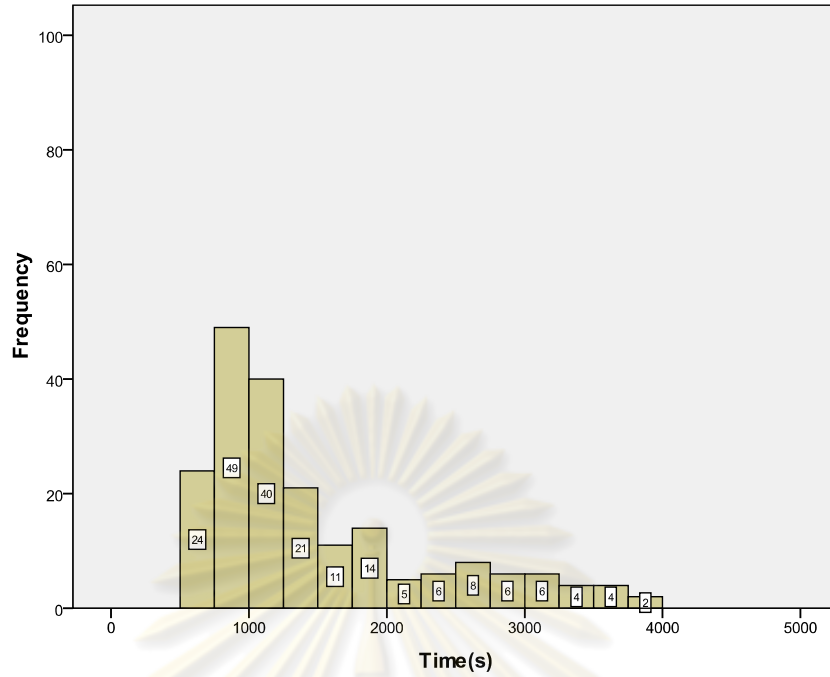
รูปที่ ง.37 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}



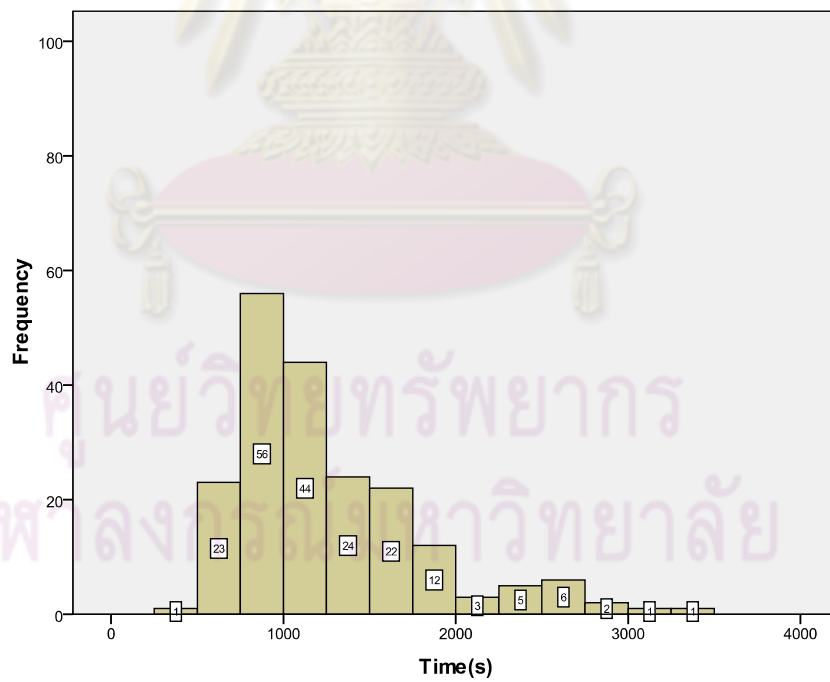
รูปที่ ง.38 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment



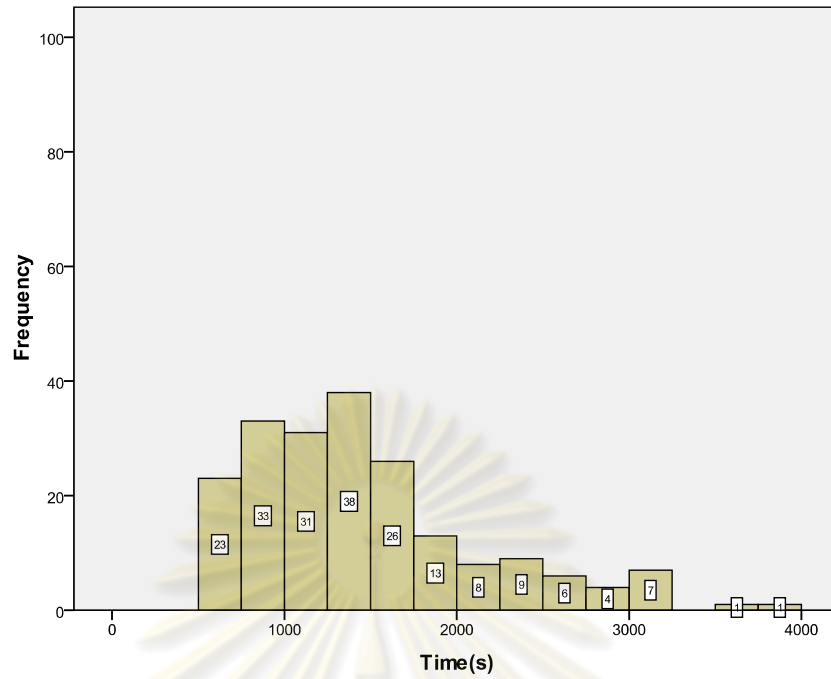
รูปที่ ง.39 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules



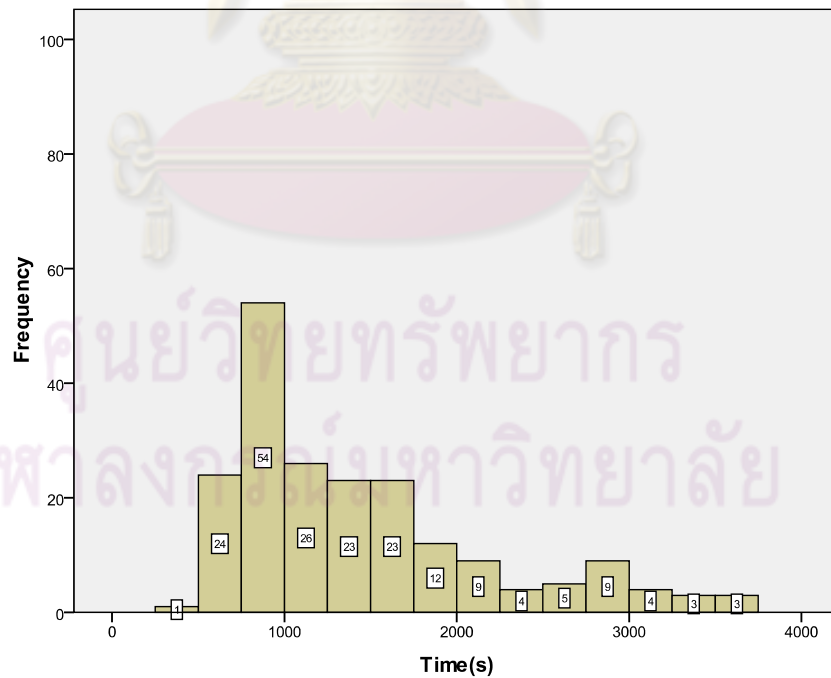
รูปที่ ง.40 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Mix



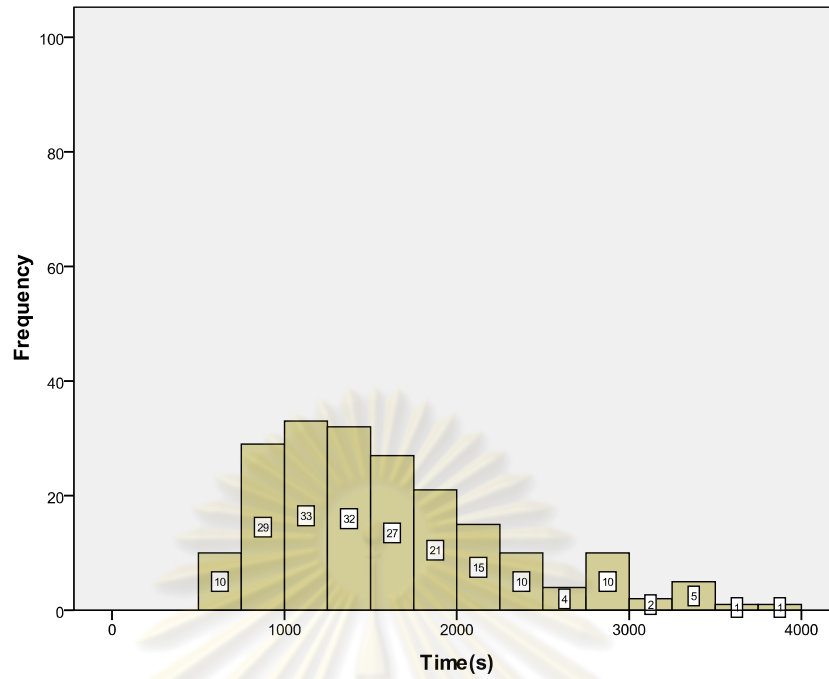
รูปที่ ง.41 ฮิสโทแกรมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule



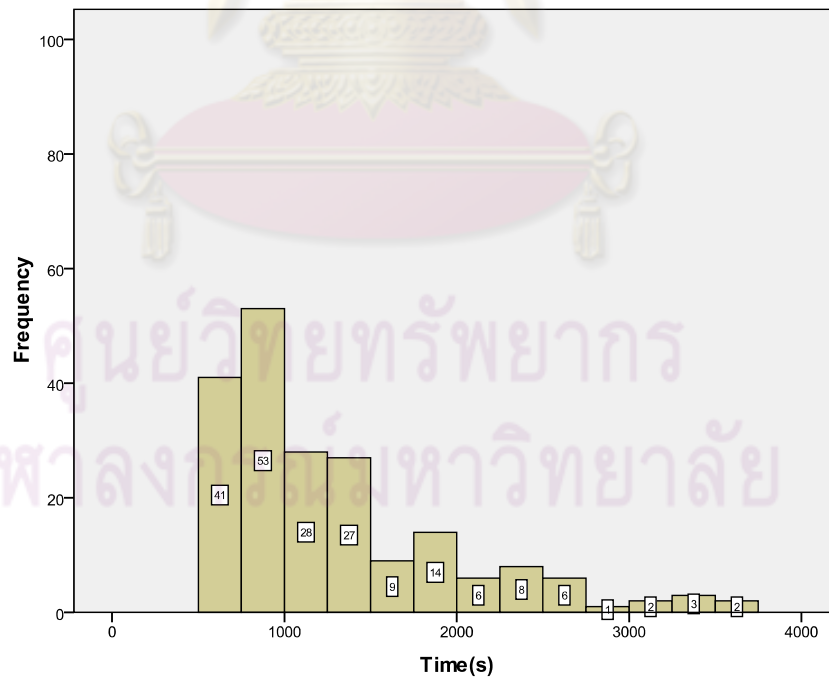
รูปที่ ง.42 ฮิสโทแกรมของปัญหาประติมากรรมที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}



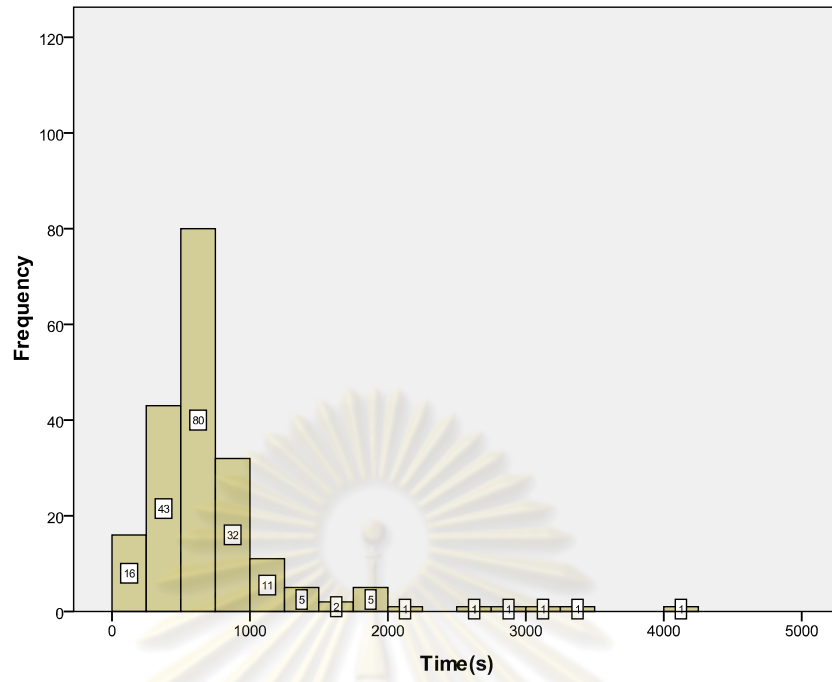
รูปที่ ง.43 ฮิสโทแกรมของปัญหาประติมากรรมที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment



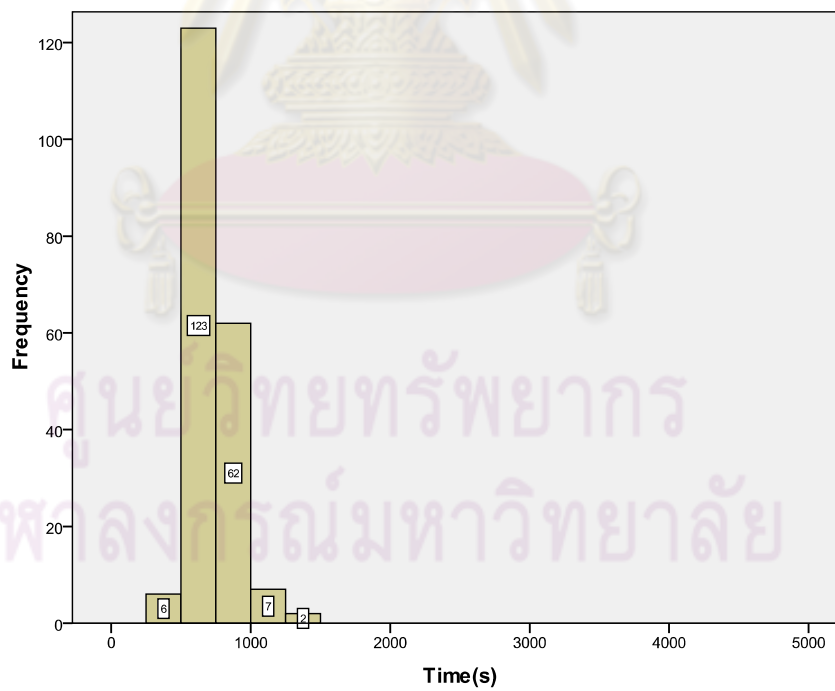
รูปที่ ง.44 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules



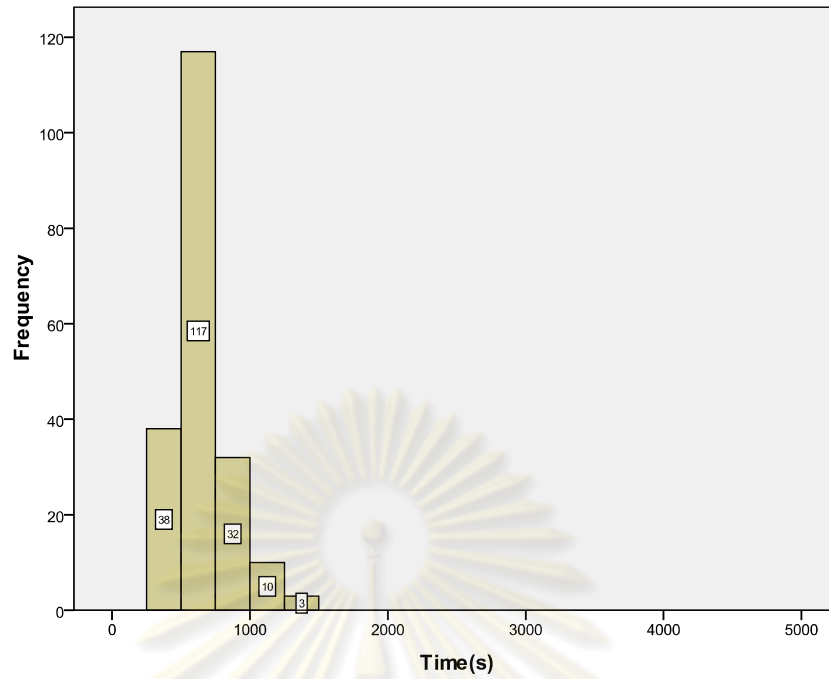
รูปที่ ง.45 ฮิสโทแกรมของปัญหาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix



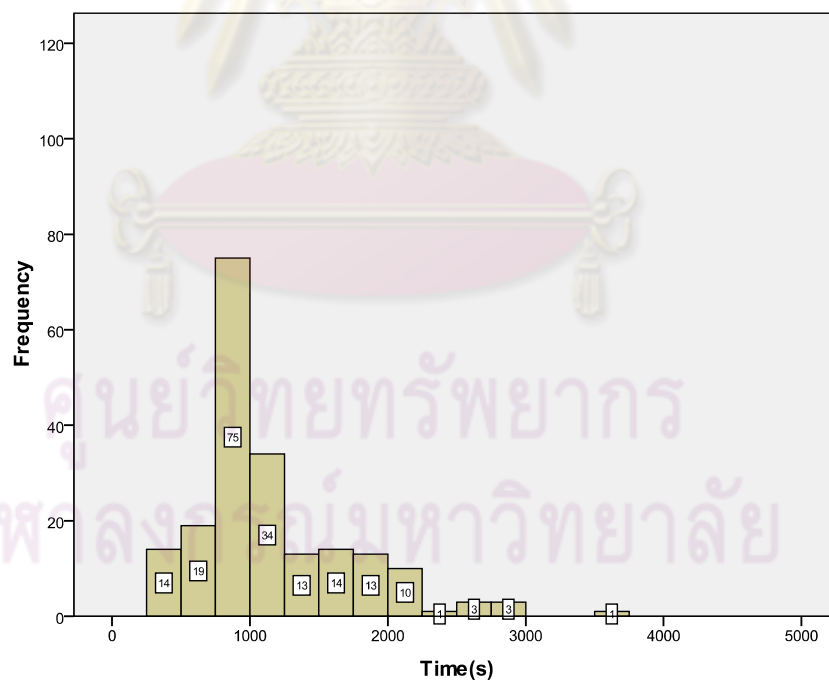
รูปที่ ง.46 ฮิสโทแกรมของ E323 - E323 ในแผ่นที่ 1 (SmallDivide)



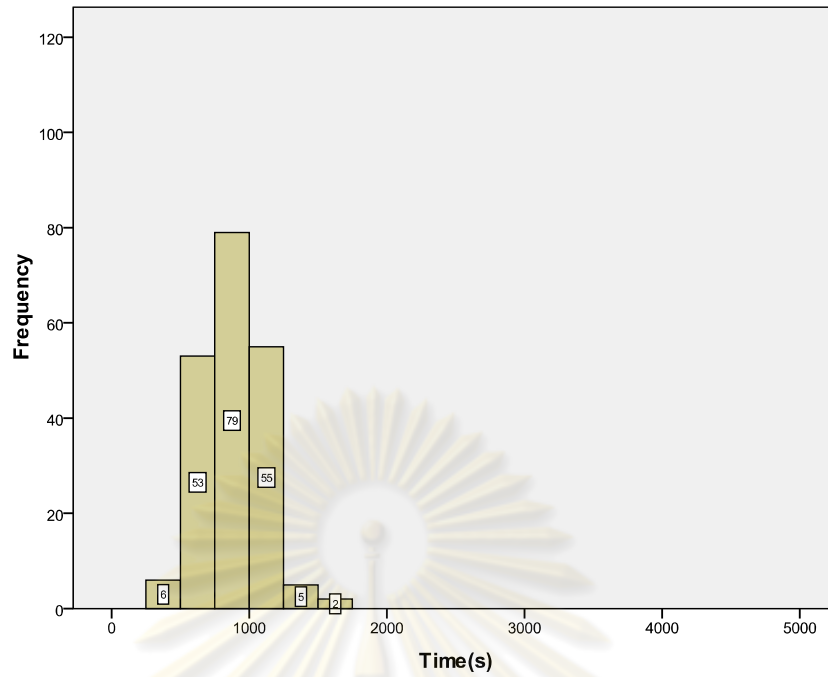
รูปที่ ง.47 ฮิสโทแกรมของ E323 - AAI ในแผ่นที่ 1 (SmallDivide)



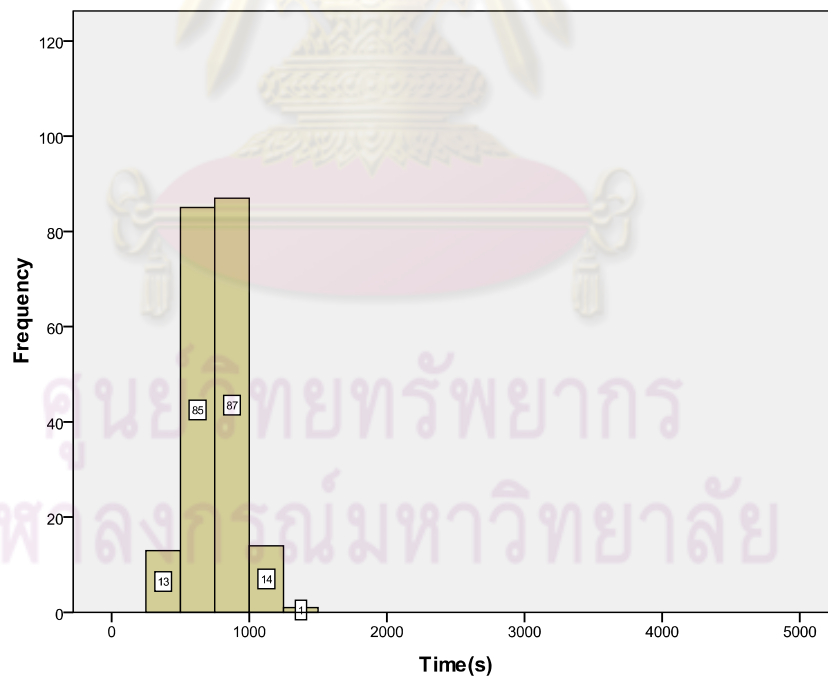
รูปที่ ง.48 ฮิสโทแกรมของ E323 - RAI ในแผ่นที่ 1 (SmallIDivide)



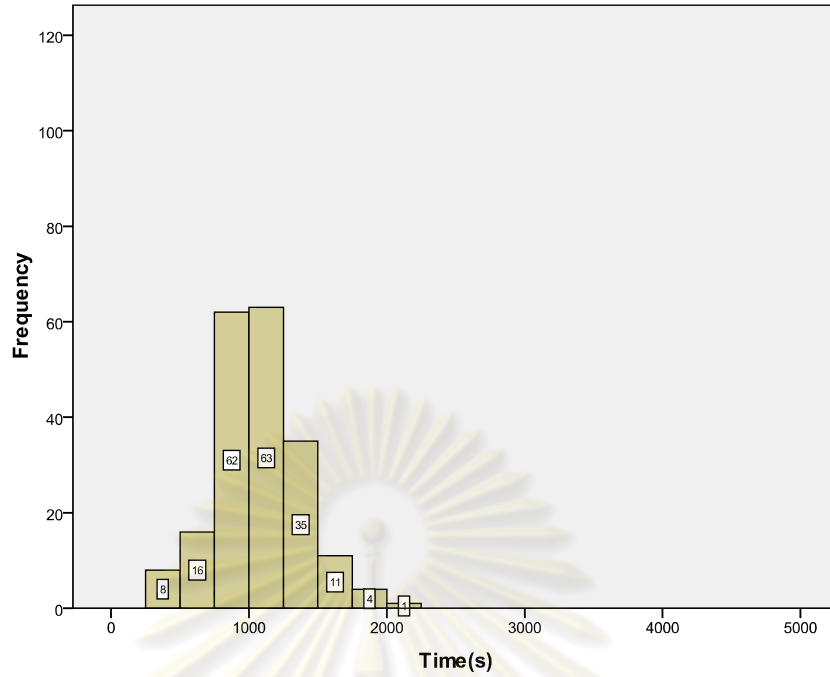
รูปที่ ง.49 ฮิสโทแกรมของ AAI - AAI ในแผ่นที่ 1 (SmallIDivide)



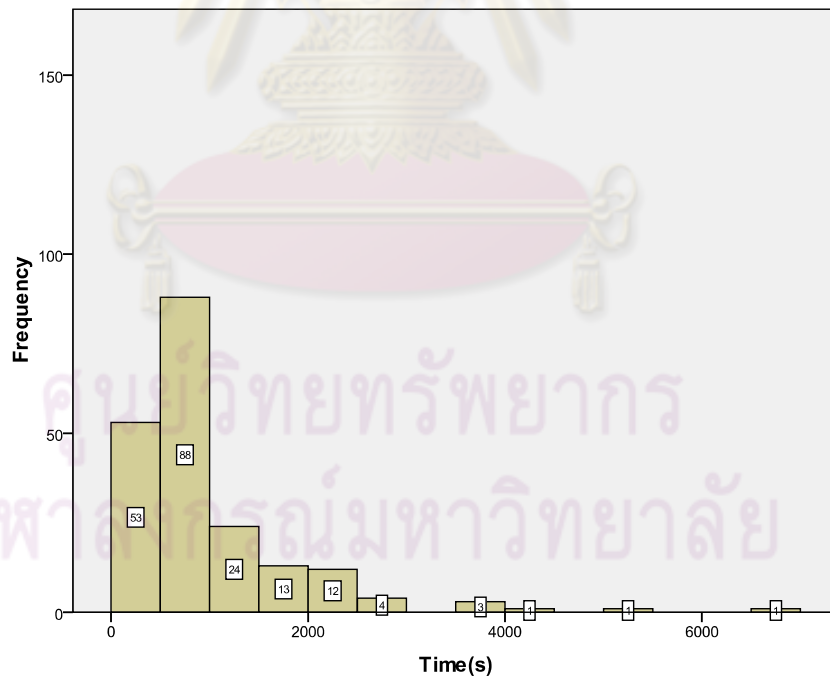
รูปที่ ๕.๕๐ ฮิสโทแกรมของ AAI - RAI ในแผนกที่ 1 (SmallDivide)



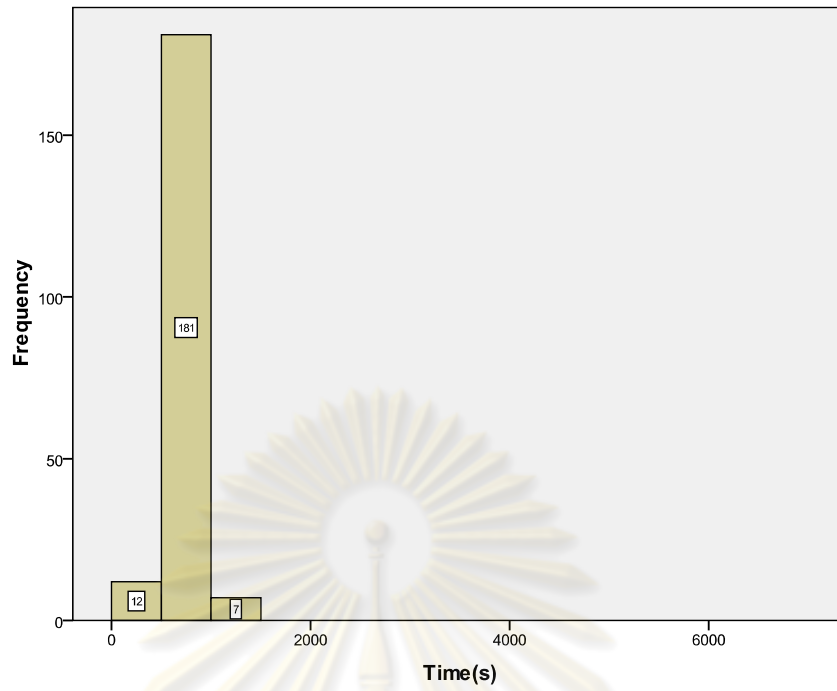
รูปที่ ๕.๕๑ ฮิสโทแกรมของ RAI - RAI ในแผนกที่ 1 (SmallDivide)



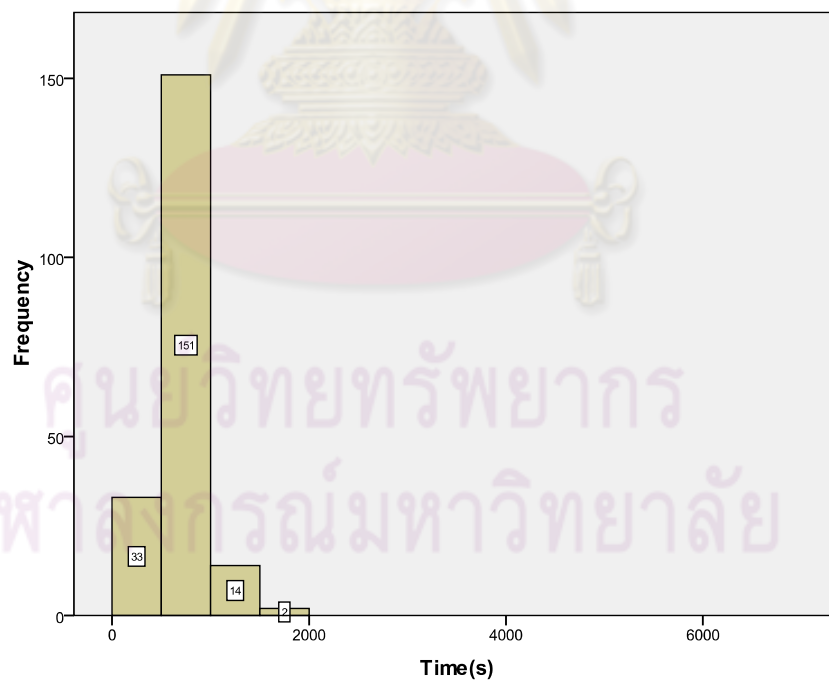
รูปที่ ๕.๕๒ ฮิสโทแกรมของ RAI - AAI ในแผ่นที่ 1 (SmallDivide)



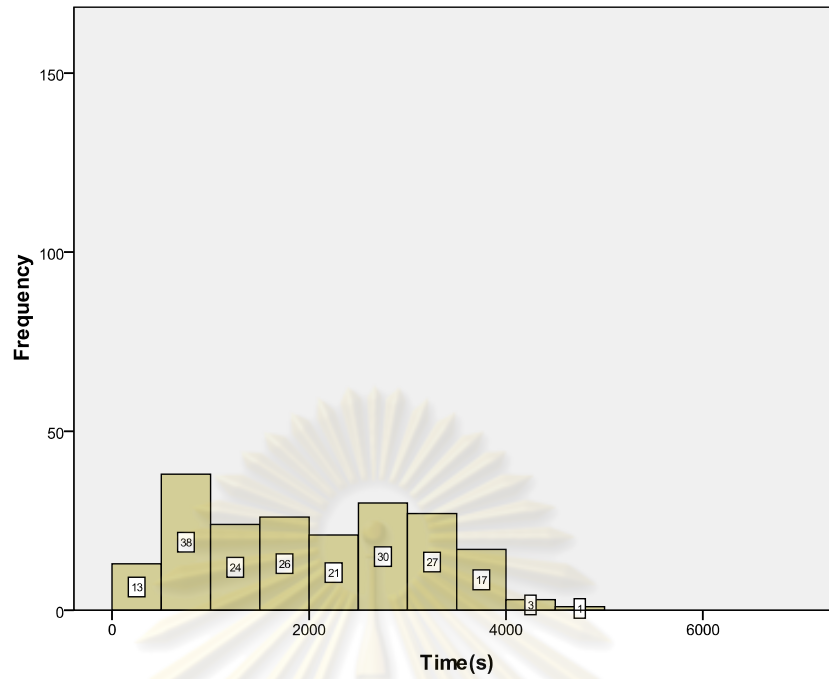
รูปที่ ๕.๕๓ ฮิสโทแกรมของ E323 - E323 ในแผ่นที่ 2 (Barren)



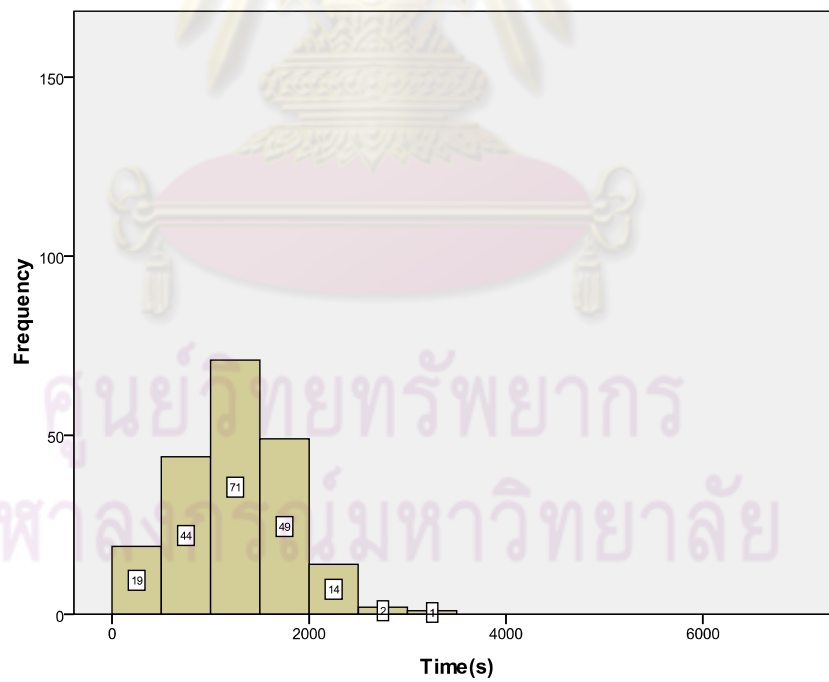
รูปที่ ง.54 ฮิสโทแกรมของ E323 - AAI ในแผนที่ 2 (Barren)



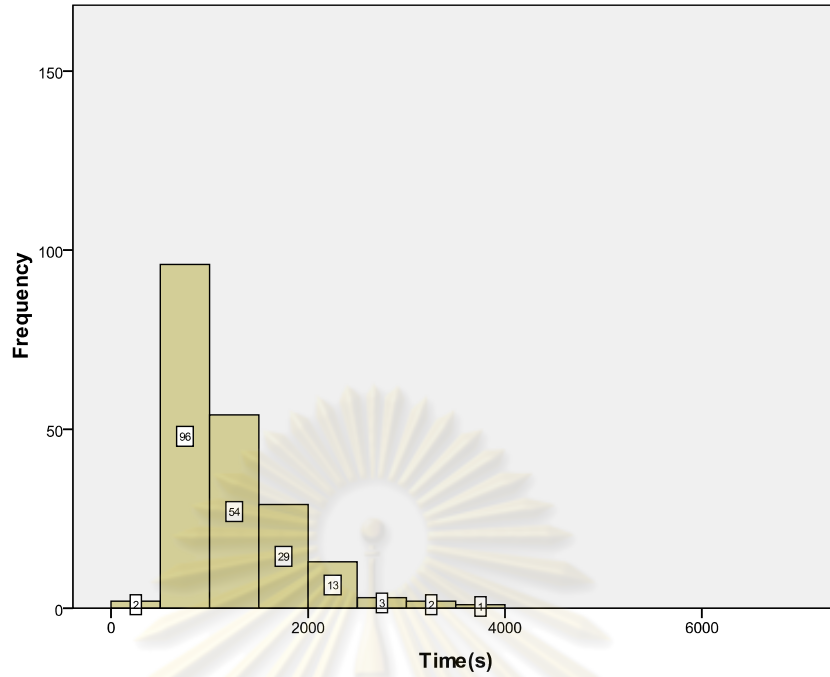
รูปที่ ง.55 ฮิสโทแกรมของ E323 - RAI ในแผนที่ 2 (Barren)



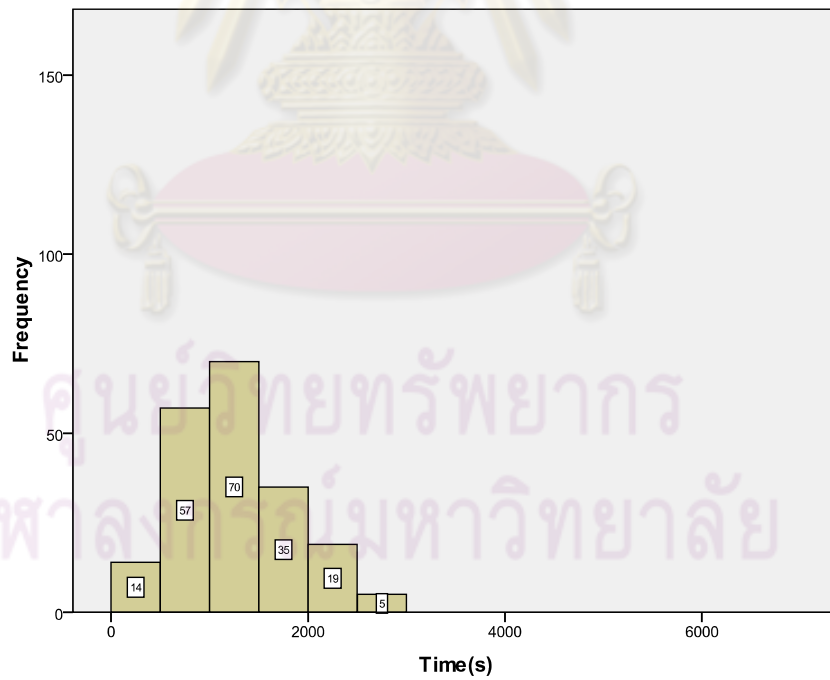
รูปที่ ง.56 ฮิสโทแกรมของ AAI - AAI ในแผนที่ 2 (Barren)



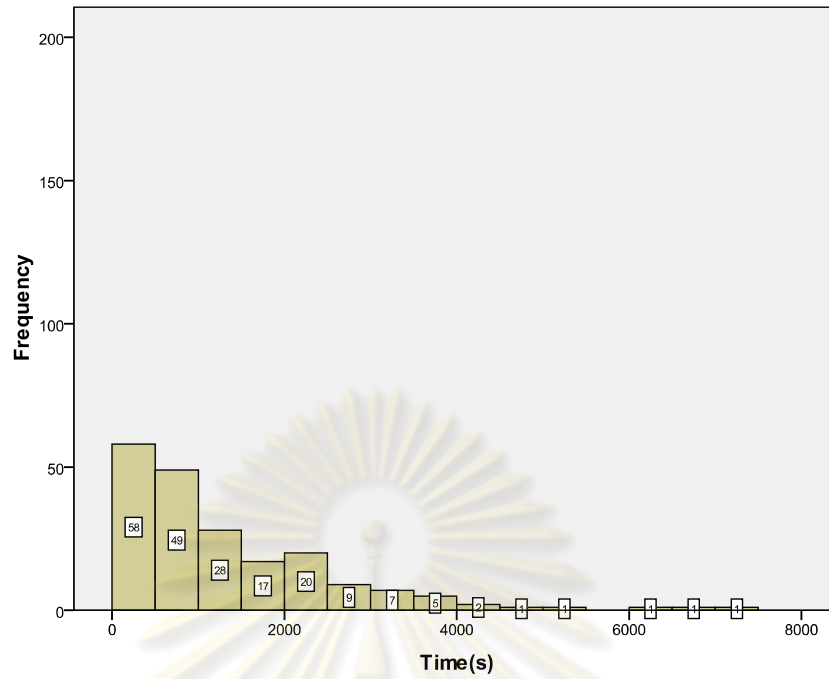
รูปที่ ง.57 ฮิสโทแกรมของ AAI - RAI ในแผนที่ 2 (Barren)



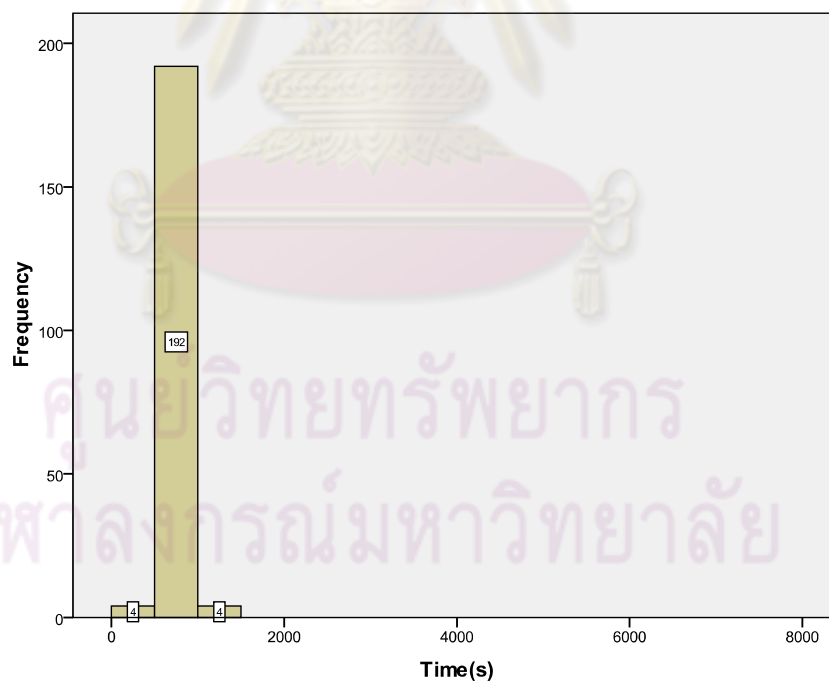
รูปที่ ๕.๕๘ ฮิสโทแกรมของ RAI - RAI ในแผนที่ ๒ (Barren)



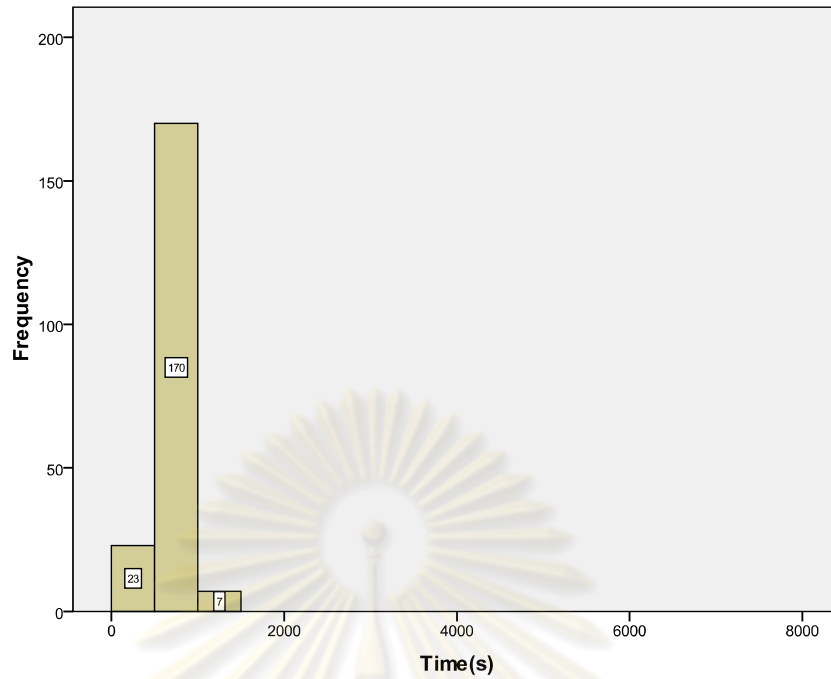
รูปที่ ๕.๕๙ ฮิสโทแกรมของ RAI - AAI ในแผนที่ ๒ (Barren)



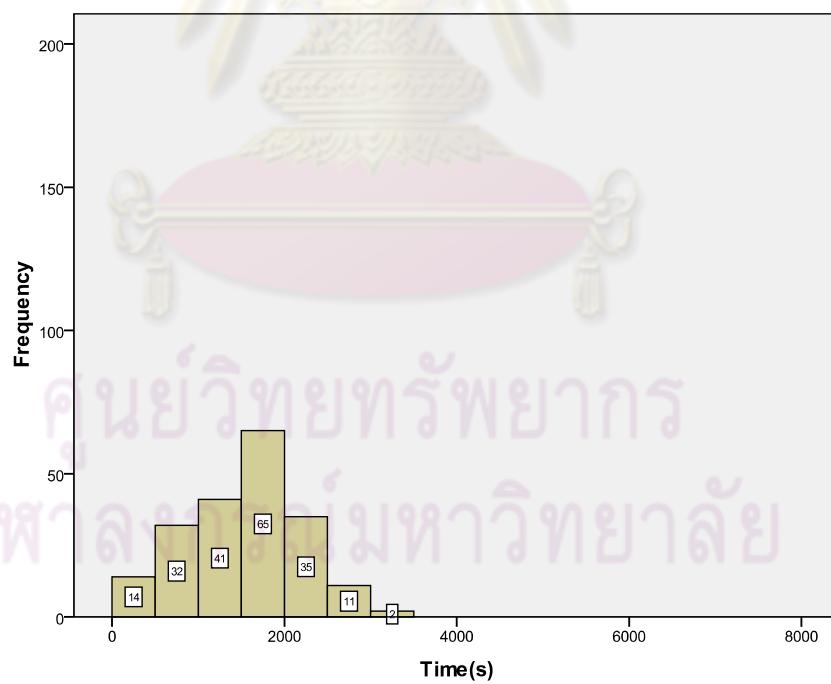
รูปที่ ง.60 ฮิสโทแกรมของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2)



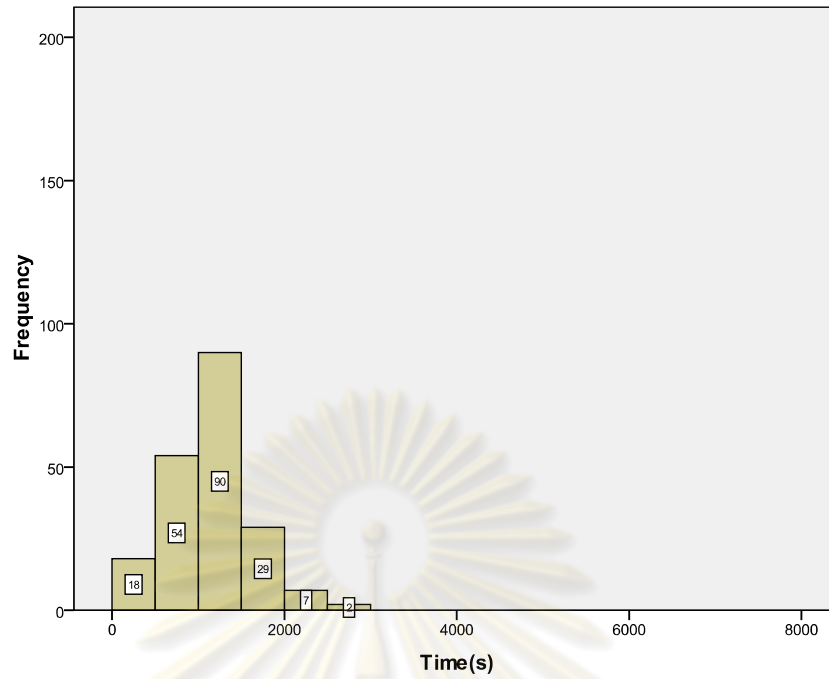
รูปที่ ง.61 ฮิสโทแกรมของ E323 - AAI ในแผนที่ 3 (Tomb Stone Desert V2)



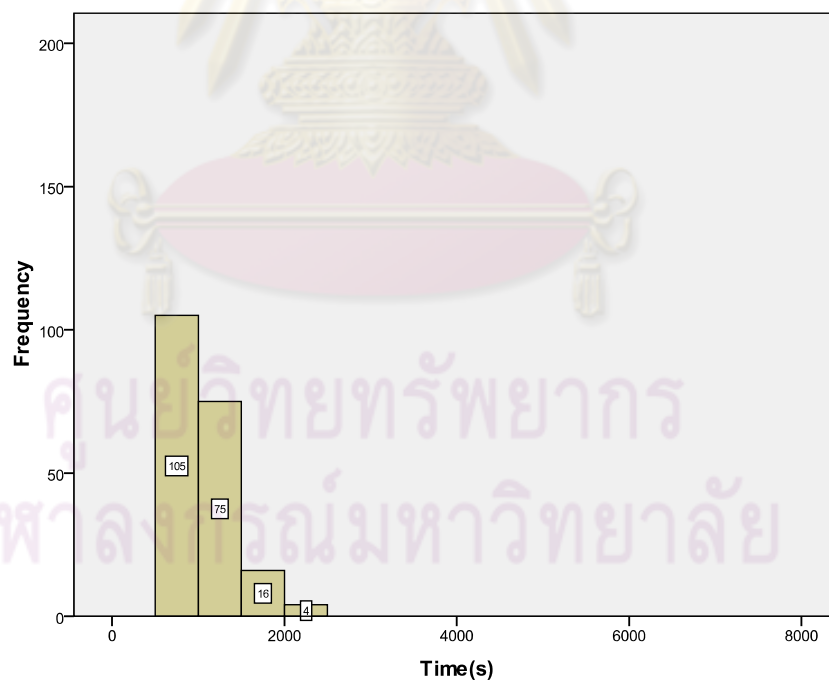
รูปที่ ง.62 ฮิสโทแกรมของ E323 - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



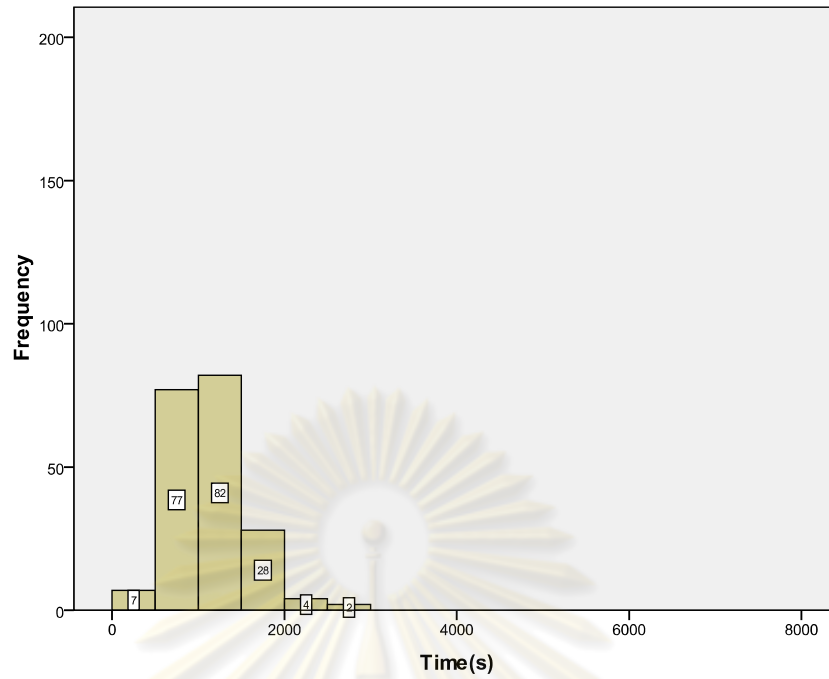
รูปที่ ง.63 ฮิสโทแกรมของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ ง.64 ฮิสโทแกรมของ AAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ ง.65 ฮิสโทแกรมของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ ง.66 ฮิสโทแกรมของ RAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)

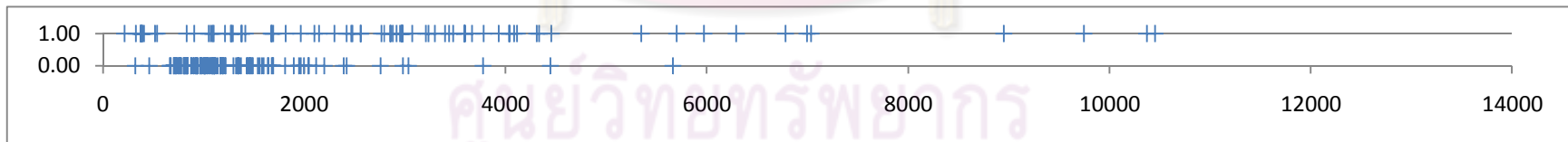
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



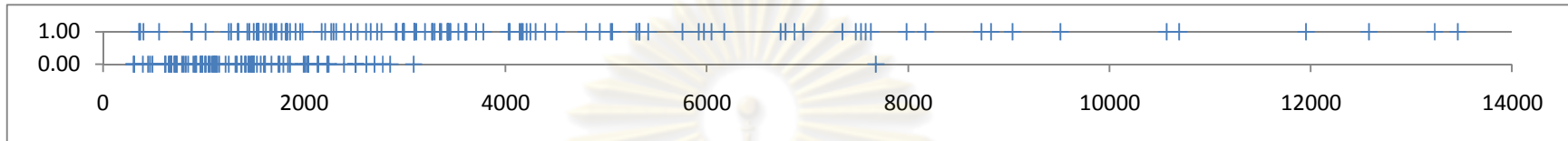
รูปที่ ง.67 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule



รูปที่ ง.68 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} w_{dec}



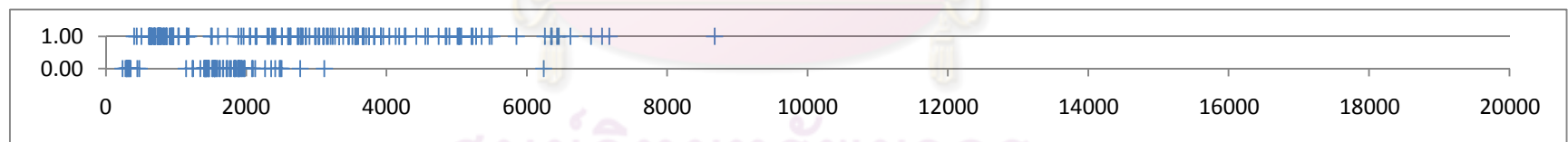
รูปที่ ง.69 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment



รูปที่ ง.70 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules

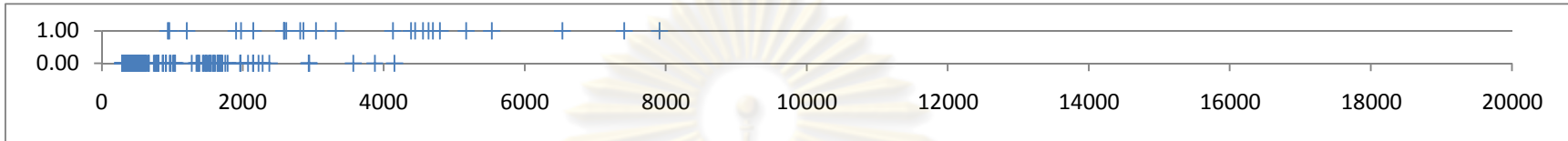


รูปที่ ง.71 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 1 (SmallDivide) โดยวิธี Mix

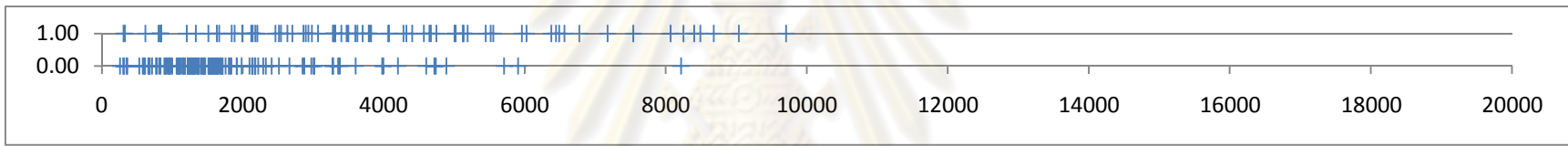


รูปที่ ง.72 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule

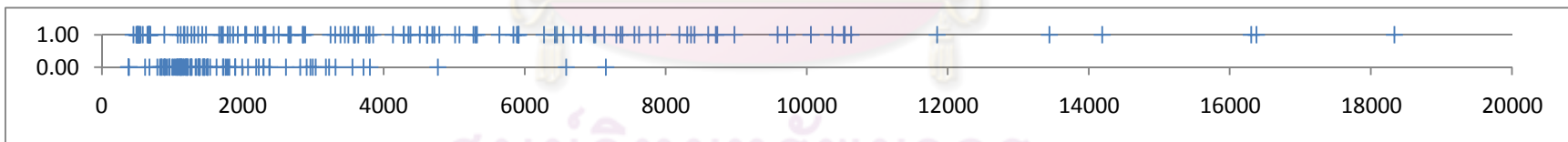
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.73 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}

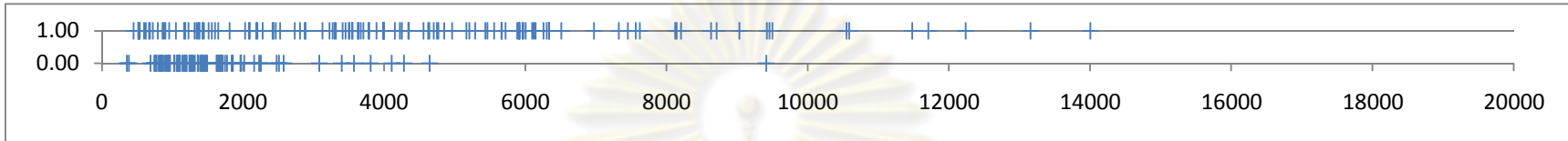


รูปที่ ง.74 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment

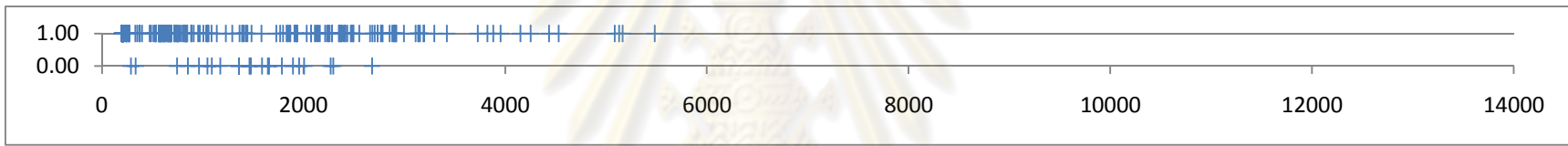


รูปที่ ง.75 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Change number of rules

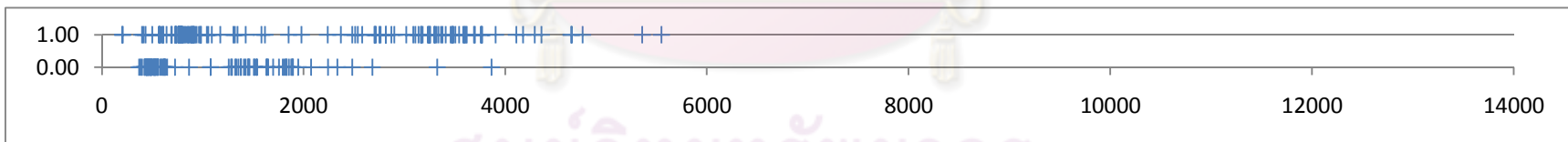
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.76 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 2 (Barren) โดยวิธี Mix

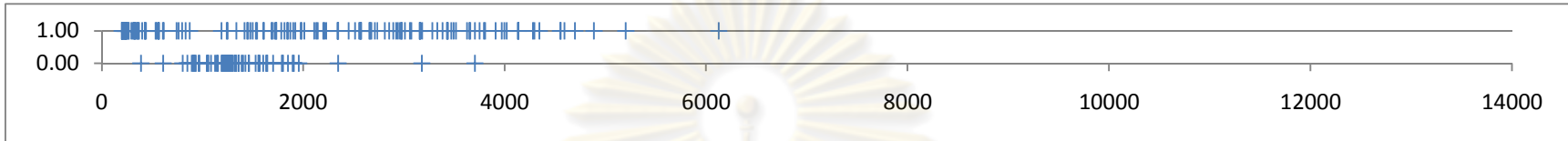


รูปที่ ง.77 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule



รูปที่ ง.78 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change winc, wdec

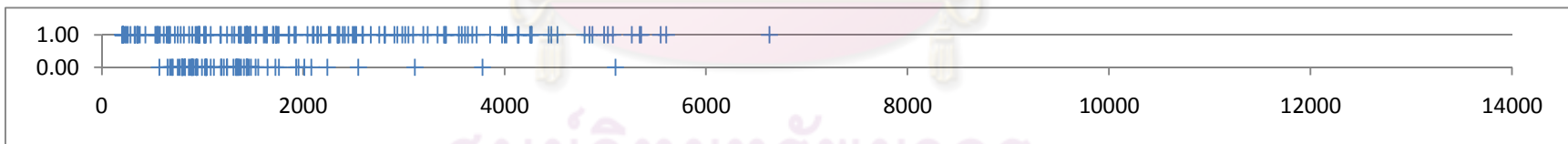
ศูนย์วิทยพัชการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.79 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment

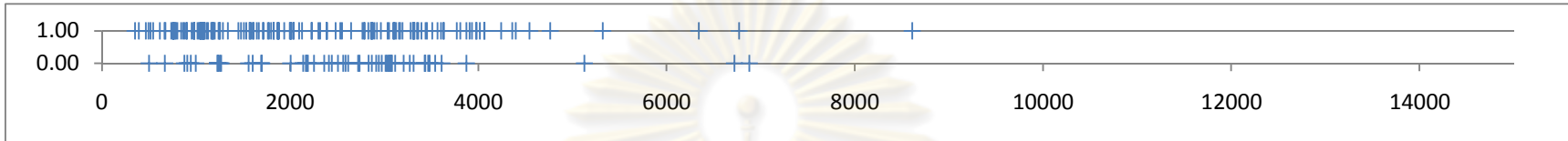


รูปที่ ง.80 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules

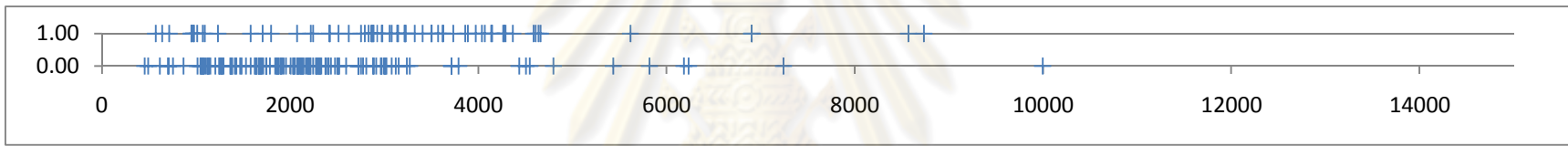


รูปที่ ง.81 กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - E323 ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix

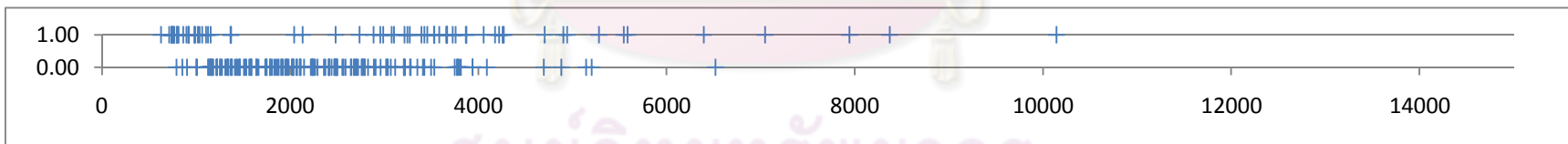
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.82 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule

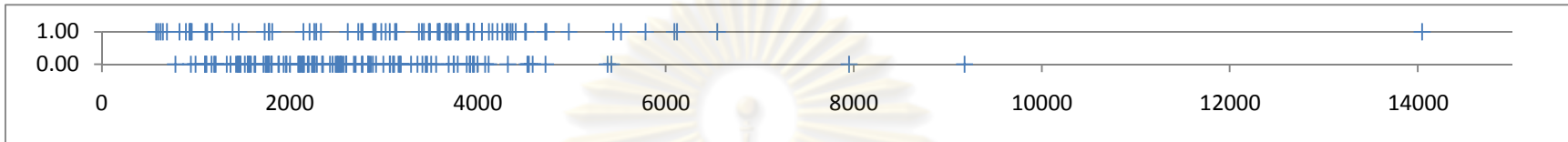


รูปที่ ง.83 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}

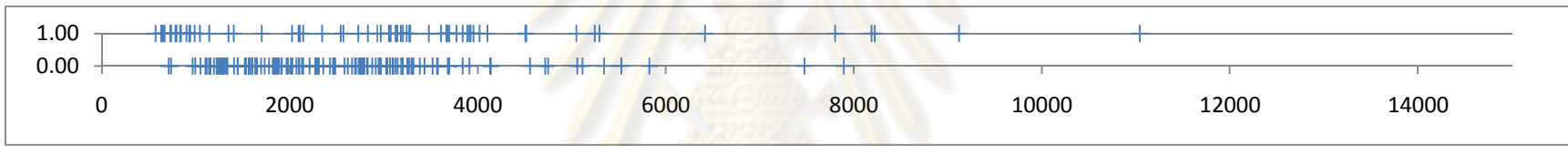


รูปที่ ง.84 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment

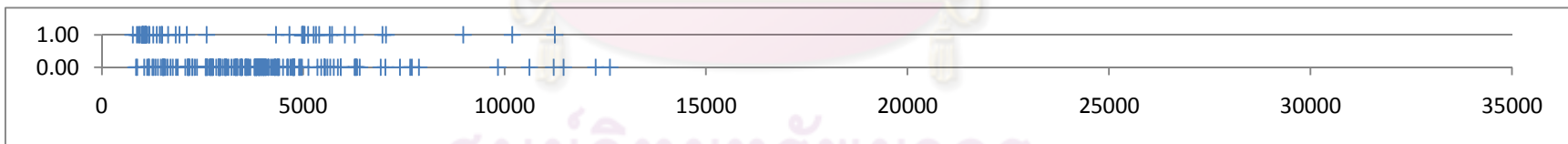
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ๘.๘๕ กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules

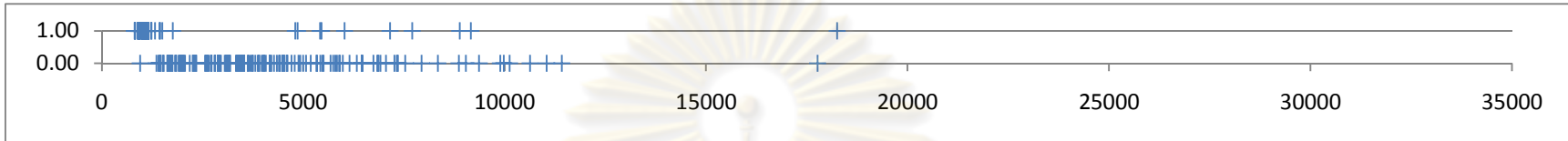


รูปที่ ๘.๘๖ กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix



รูปที่ ๘.๘๗ กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule

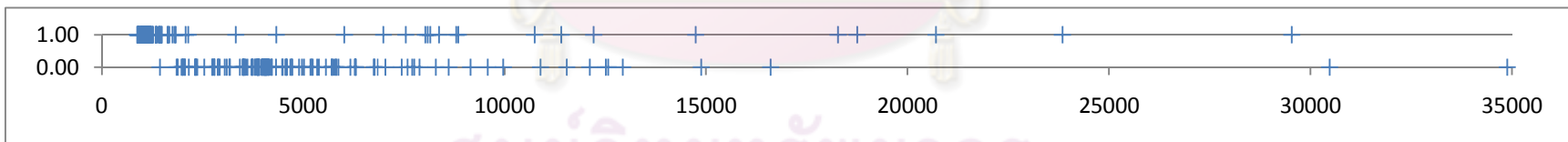
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ๘.88 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}

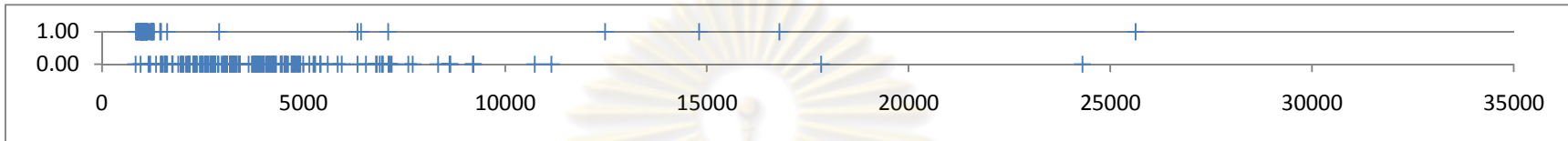


รูปที่ ๘.89 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment

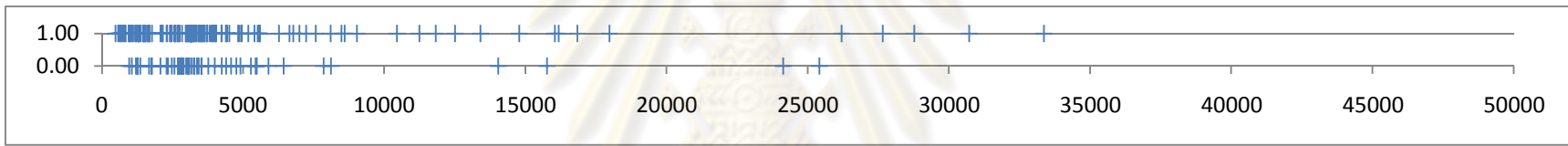


รูปที่ ๘.90 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules

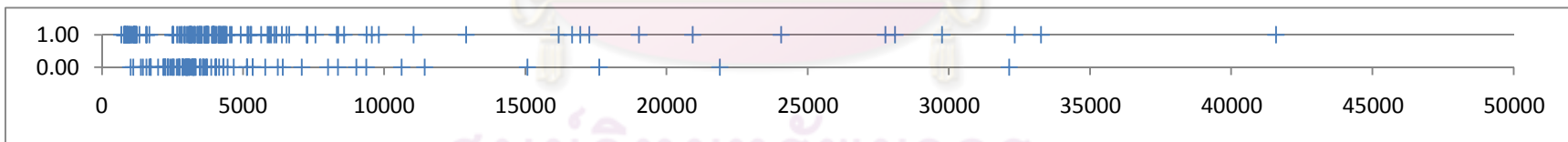
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.91 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 2 (Barren) โดยวิธี Mix

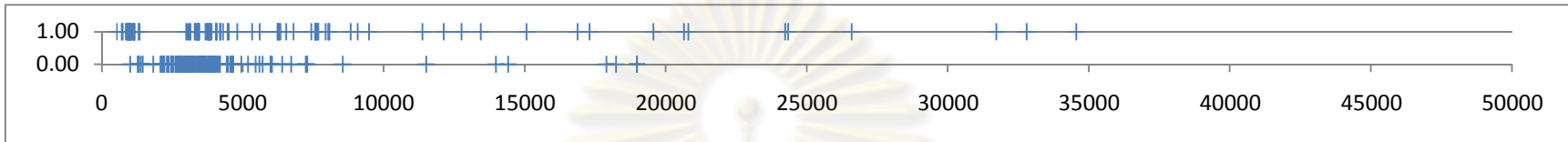


รูปที่ ง.92 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule

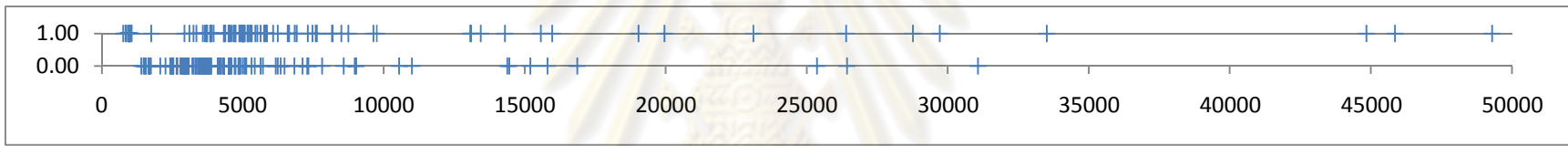


รูปที่ ง.93 กราฟผล ชนะ/แพ้ ท้ายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}

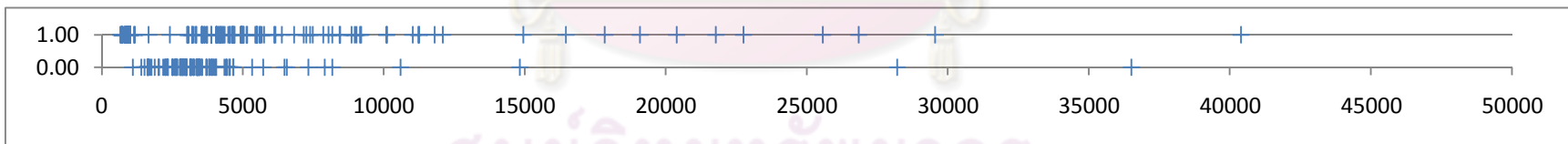
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ๙.๙๔ กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment

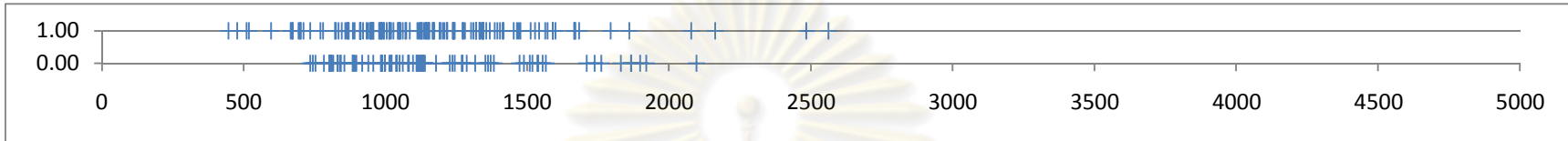


รูปที่ ๙.๙๕ กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules

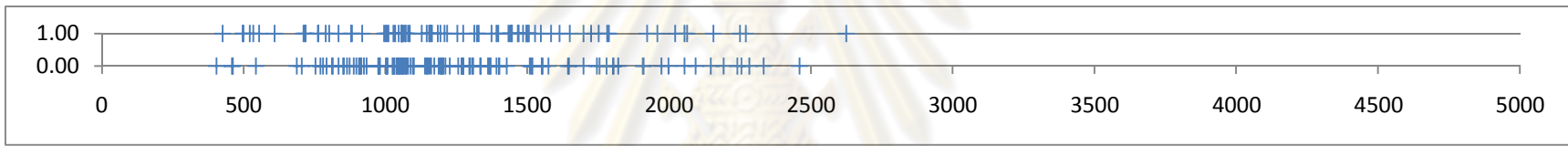


รูปที่ ๙.๙๖ กราฟผล ชนะ/แพ้ ทำายเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - AAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix

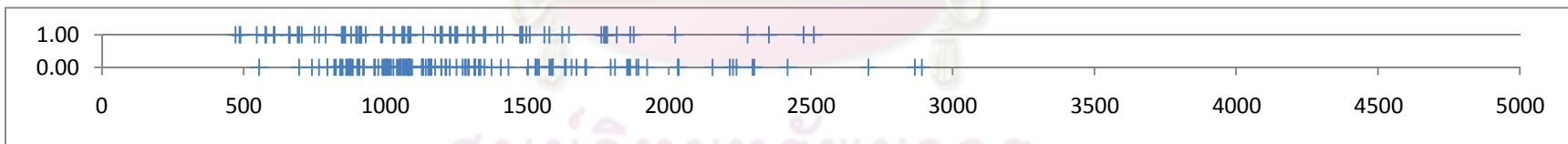
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.97 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Fix 1 rule

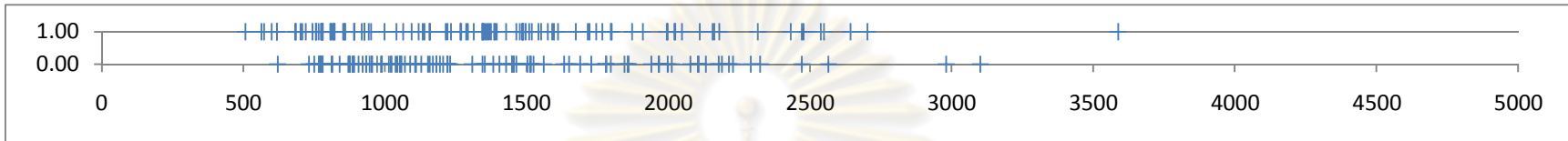


รูปที่ ง.98 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change w_{inc} , w_{dec}

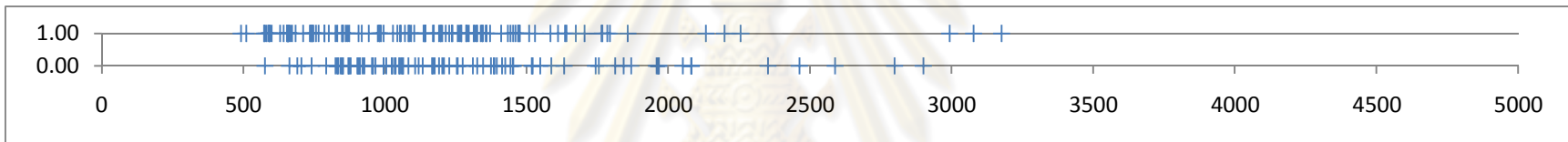


รูปที่ ง.99 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี 7-minute increment

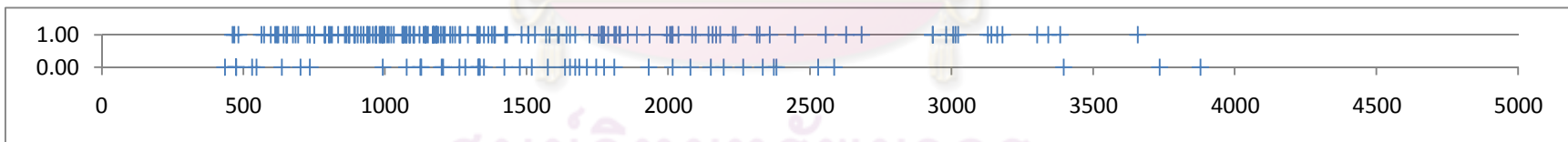
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.100 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Change number of rules



รูปที่ ง.101 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 1 (SmallDivide) โดยวิธี Mix

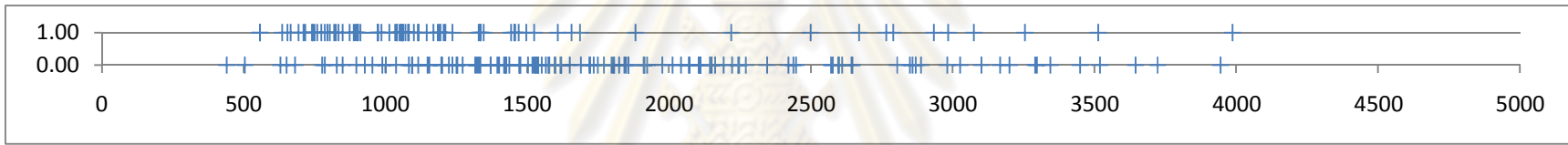


รูปที่ ง.102 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Fix 1 rule

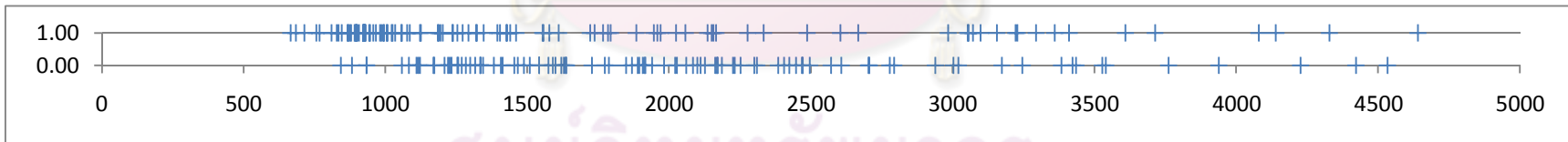
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.103 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change w_{inc} , w_{dec}

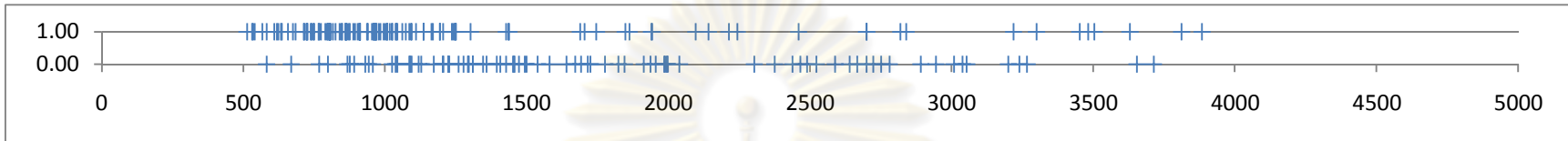


รูปที่ ง.104 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี 7-minute increment

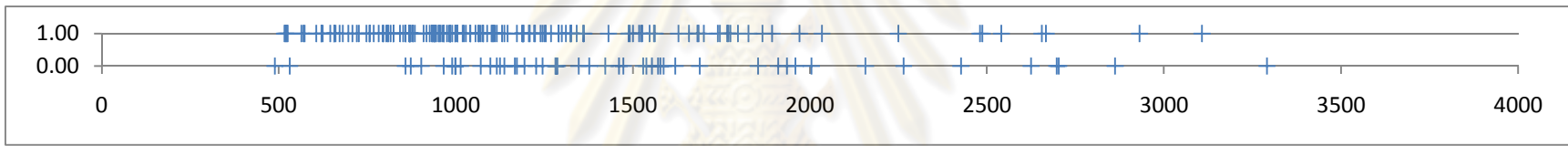


รูปที่ ง.105 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Change number of rules

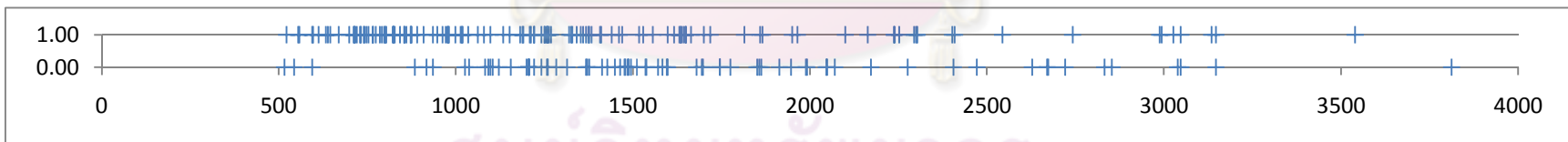
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.106 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 2 (Barren) โดยวิธี Mix

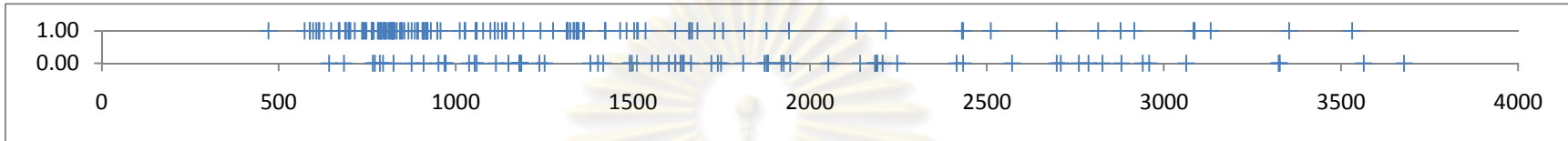


รูปที่ ง.107 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Fix 1 rule

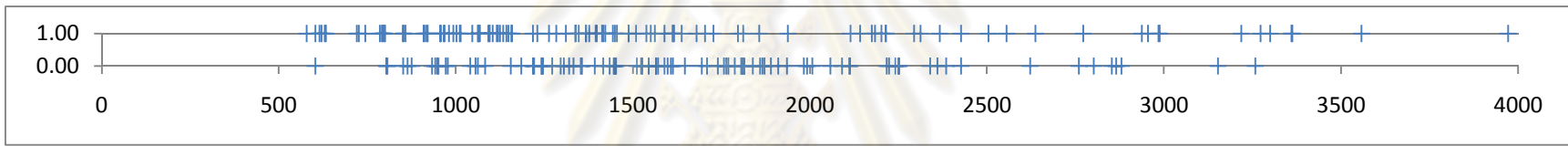


รูปที่ ง.108 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change w_{inc} , w_{dec}

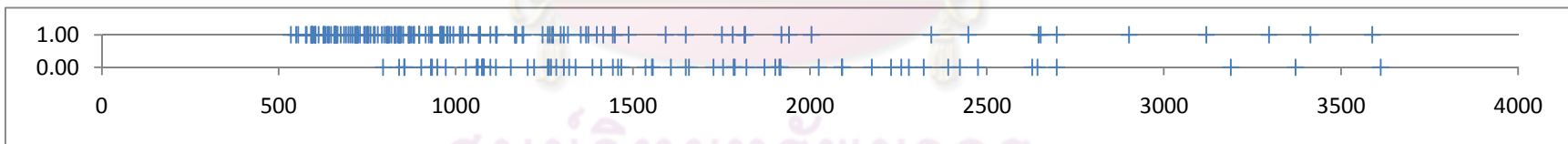
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.109 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี 7-minute increment

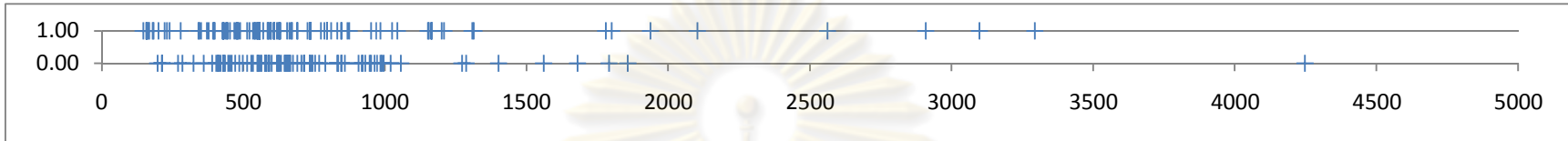


รูปที่ ง.110 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Change number of rules

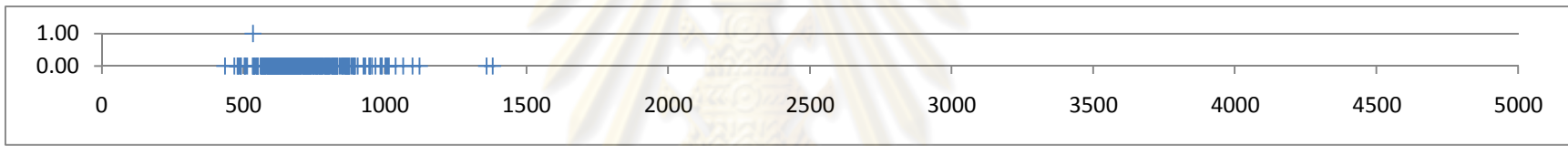


รูปที่ ง.111 กราฟผล ชนะ/แพ้ ทำเกมของปัญญาประดิษฐ์ที่สร้างขึ้น - RAI ในแผนที่ 3 (Tomb Stone Desert V2) โดยวิธี Mix

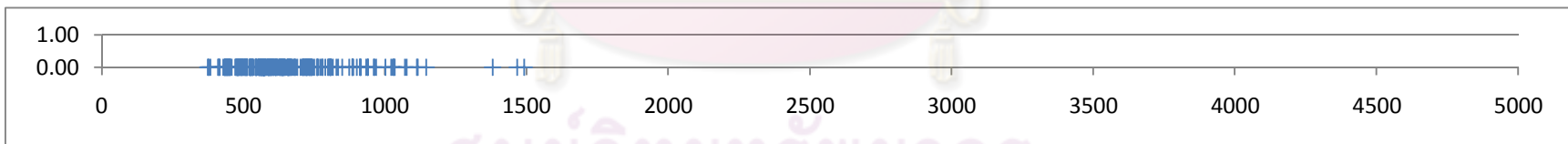
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.112 กราฟผล ชนะ/แพ้ ทำเกมของ E323 - E323 ในแผนที่ 1 (SmallDivide)

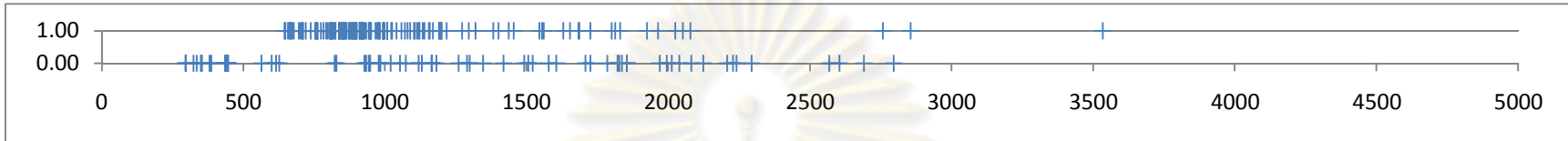


รูปที่ ง.113 กราฟผล ชนะ/แพ้ ทำเกมของ E323 - AAI ในแผนที่ 1 (SmallDivide)

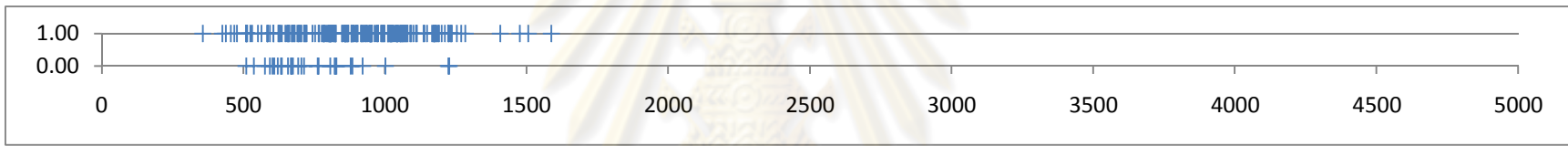


รูปที่ ง.114 กราฟผล ชนะ/แพ้ ทำเกมของ E323 - RAI ในแผนที่ 1 (SmallDivide)

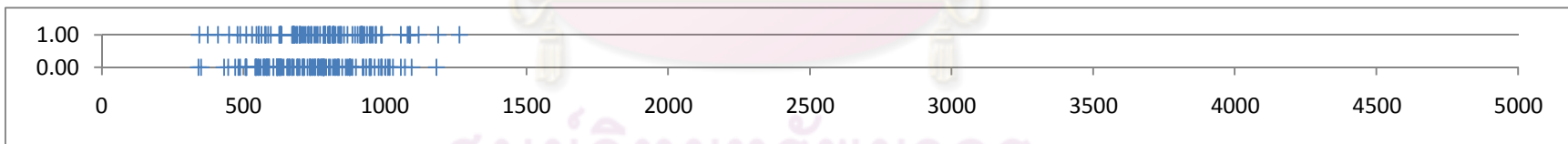
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.115 กราฟผล ชนะ/แพ้ ทำเกมของ AAI - AAI ในแผนที่ 1 (SmallDivide)

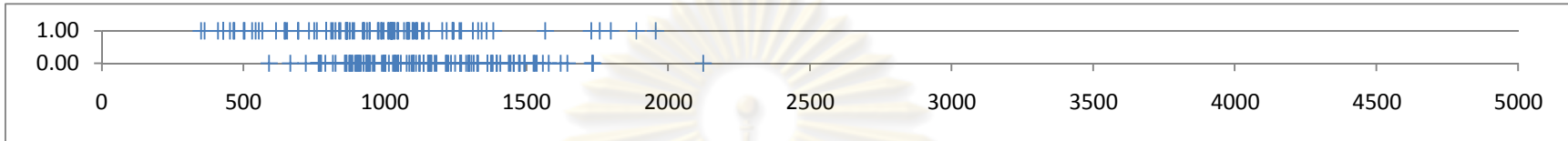


รูปที่ ง.116 กราฟผล ชนะ/แพ้ ทำเกมของ AAI - RAI ในแผนที่ 1 (SmallDivide)

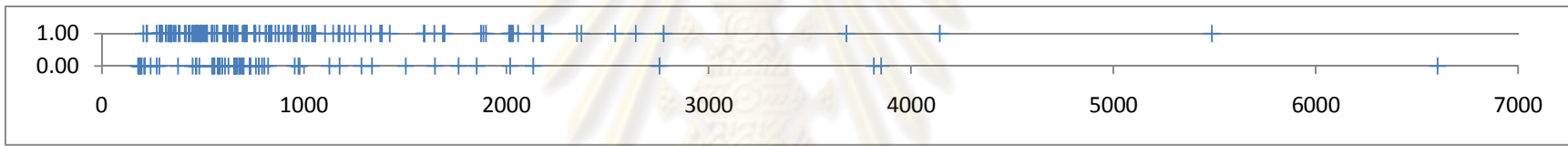


รูปที่ ง.117 กราฟผล ชนะ/แพ้ ทำเกมของ RAI - RAI ในแผนที่ 1 (SmallDivide)

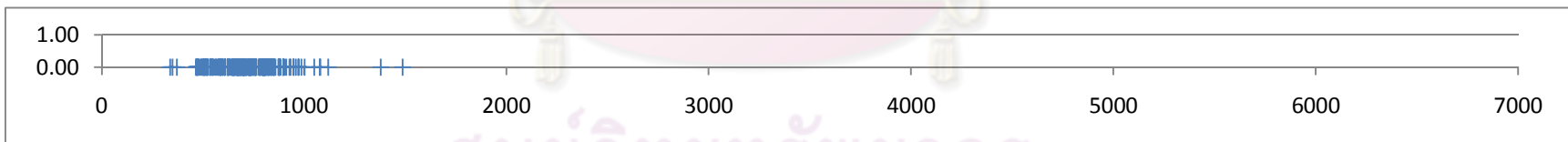
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.118 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - AAI ในแผนที่ 1 (SmallDivide)

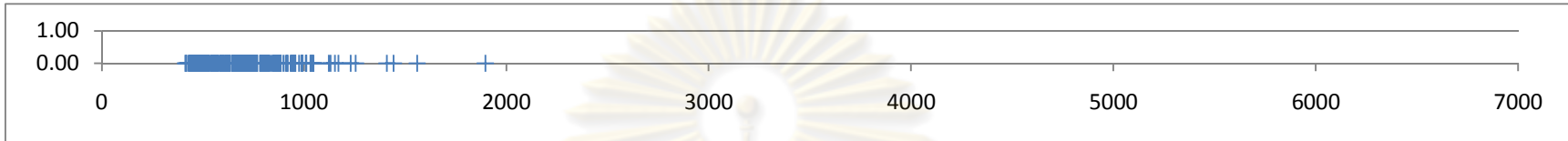


รูปที่ ง.119 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - E323 ในแผนที่ 2 (Barren)



รูปที่ ง.120 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - AAI ในแผนที่ 2 (Barren)

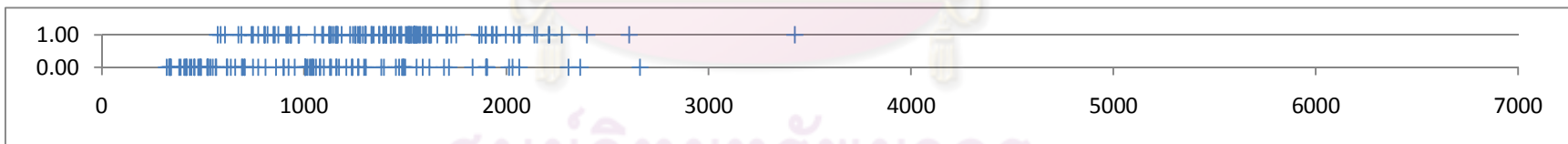
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.121 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - RAI ในแผนที่ 2 (Barren)



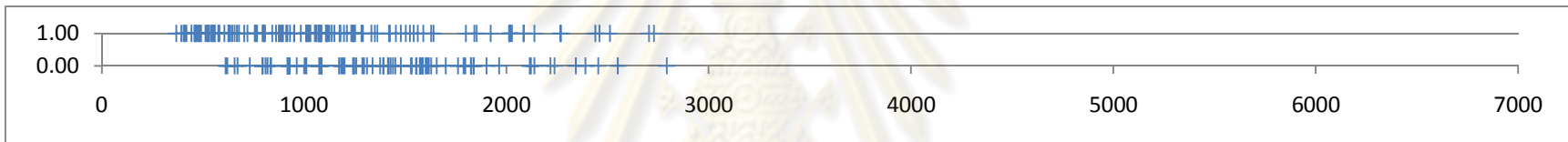
รูปที่ ง.122 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - AAI ในแผนที่ 2 (Barren)



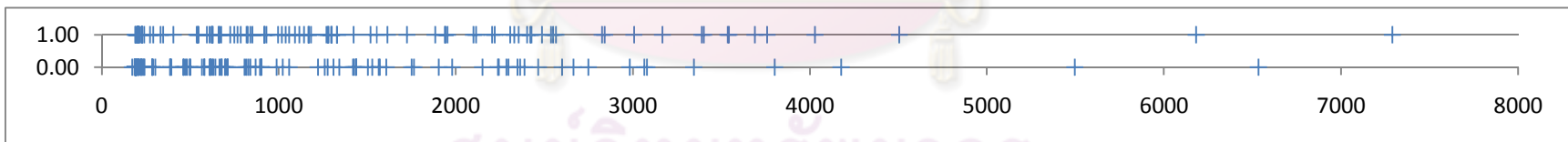
รูปที่ ง.123 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - RAI ในแผนที่ 2 (Barren)



รูปที่ ง.124 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - RAI ในแผนที่ 2 (Barren)

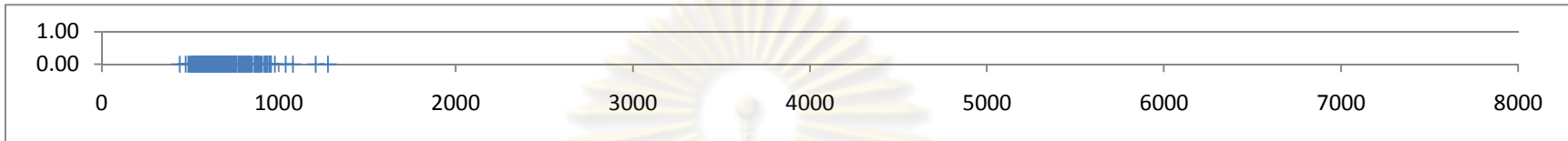


รูปที่ ง.125 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - AAI ในแผนที่ 2 (Barren)

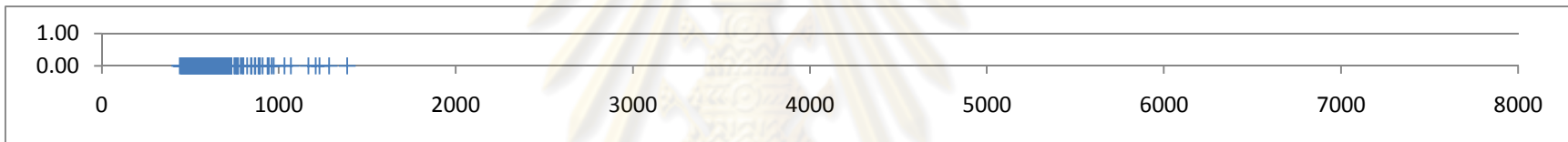


รูปที่ ง.126 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - E323 ในแผนที่ 3 (Tomb Stone Desert V2)

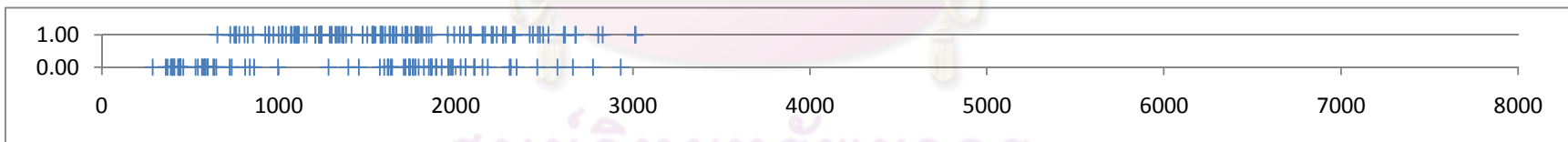
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.127 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - AAI ในแผนที่ 3 (Tomb Stone Desert V2)

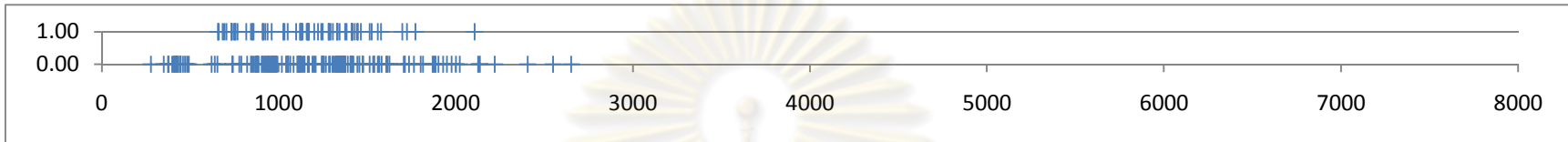


รูปที่ ง.128 กราฟผล ชนะ/แพ้ ท้ายเกมของ E323 - RAI ในแผนที่ 3 (Tomb Stone Desert V2)

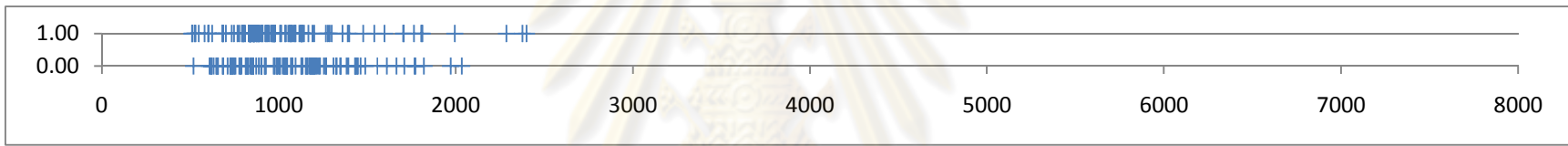


รูปที่ ง.129 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)

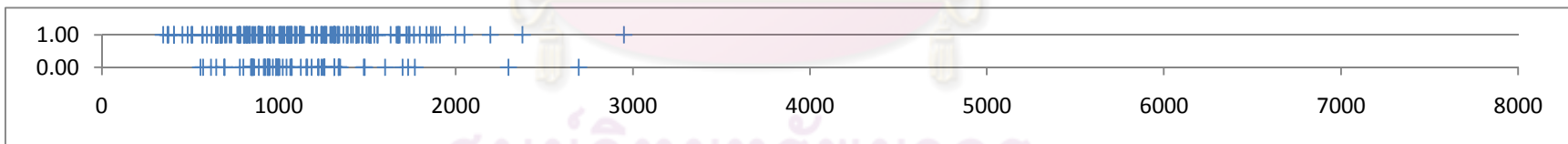
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.130 กราฟผล ชนะ/แพ้ ท้ายเกมของ AAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ ง.131 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - RAI ในแผนที่ 3 (Tomb Stone Desert V2)



รูปที่ ง.132 กราฟผล ชนะ/แพ้ ท้ายเกมของ RAI - AAI ในแผนที่ 3 (Tomb Stone Desert V2)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายปิยพจน์ เกษมภักดีพงษ์ เกิดวันที่ 29 สิงหาคม พ.ศ. 2527 สำเร็จ การศึกษาระดับมัธยมศึกษาที่โรงเรียนเตรียมอุดมศึกษา จากนั้นจึงเข้าศึกษาต่อที่คณะ วิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2546 และในปีการศึกษา 2549 จึง สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า และเข้าศึกษาใน หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรม คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2550



ศูนย์วิทยพัชร์พยากร
จุฬาลงกรณ์มหาวิทยาลัย