รายงานผลการประดิษฐ์
เงินอุดหนุนโครงการสิ่งประดิษฐ์

เรื่อง

คอมพิวเตอร์หมากรุกไทย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จฬ
วศ 15
007918

โดย

บุญเสริม กิจศิริกุล

2539

จุฬาลงกรณ์มหาวิทยาลัย

โครงการสิ่งประดิษฐ์

รายงาน

คอมพิวเตอร์หมากรุกไทย

โดย

บุญเสริม กิจศิริกุล

มีนาคม 2539

จุฬาลงกรณ์มหาวิทยาลัย

I17900931 ... 2541

กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงไปด้วยดีได้ เพราะได้รับการสนับสนุนจากเงินอุดหนุนโครงการ-สิ่งประดิษฐ์จุฬาลงกรณ์มหาวิทยาลัย ขอขอบคุณไว้ ณ ที่นี้ และขอขอบคุณ นายบัญชร ตั้งปรัชญาวุธ นายธนะเทพ พันธุ์สุวรรณกิจ และนายพีรพงษ์ เจียรณัยที่ได้มีส่วนช่วยในงานวิจัยนี้

จุฬาลงกรณ์มหาวิทยาลัย

ชื่อโครงการ คอมพิวเตอร์หมากรุกไทย
ชื่อผู้ดำเนินงาน บุญเสริม กิจศิริกุล
เดือนและปีที่ทำวิจัยเสร็จ มีนาคม 2539

บทคัดย่อ

การวิจัยนี้ศึกษาเปรียบเทียบอัลกอริทึมค้นหาในหมากรุกไทย ได้แก่ อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา อัลกอริทึม B* และอัลกอริทึมตัวเลขคอนสไปเรซี นำอัลกอริทึมมาพัฒนาโปรแกรมเล่นเกมหมากรุกไทยและหาข้อสรุปของอัลกอริทึมที่เหมาะสมในเกมหมากรุกไทย เรานิยามฟังก์ชันฮิวริสติกที่เหมาะสมต่อการเขียนโปรแกรมของอัลกอริทึมในหมากรุกไทย และทำการทดลองเพื่อเปรียบเทียบอัลกอริทึม

การทดลองแบ่งออกเป็น 4 ชุดเพื่อใช้แข่งขันกันชุดละ 10 เกม การทดลองในแต่ละชุดได้ออกแบบขึ้นเพื่อวัดค่าคะแนนของหมากที่เหลือเมื่อควบคุมให้จำนวนโหนดที่สร้างได้ในแต่ละอัลกอริทึมใกล้เคียงกัน หรือวัดค่าคะแนนของหมากที่เหลือเมื่อควบคุมเวลาในการสร้างตาเดินจากแต่ละอัลกอริทึมภายในเวลาเท่ากัน ทำการวัดประสิทธิภาพของอัลกอริทึมตามการทดลองที่ออกแบบไว้ โดยใช้โปรแกรมหมากรุกไทยที่พัฒนาขึ้น

ผลการวิจัยพบว่า 1.)อัลกอริทึม B* ไม่เหมาะสมกับหมากรุกไทย เนื่องจากการนิยามฟังก์ชันฮิวริสติกสำหรับการลู่เข้าของอัลกอริทึมเป็นไปได้ยาก 2.)อัลกอริทึมตัวเลขคอนสไปเรซีสร้างตาเดินได้ดีกว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา เนื่องจากสืบค้นได้ลึกกว่าเมื่อกำหนดให้แต่ละอัลกอริทึมสร้างโหนดได้ใกล้เคียงกัน และ 3.)อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาสร้างตาเดินได้ดีกว่าอัลกอริทึมตัวเลขคอนสไปเรซี เนื่องจากสร้างโหนดได้มากกว่า เมื่อกำหนดให้แต่ละอัลกอริทึมสร้างตาเดินภายในเวลาเท่ากัน

โดยสรุป อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามีความเหมาะสมที่จะนำมาใช้ในเกมหมากรุกไทยมากกว่าอัลกอริทึมอื่น เนื่องจากประสิทธิภาพที่เหนือกว่าในด้านความเร็วของการสืบค้น และพื้นที่หน่วยความจำที่ใช้ อย่างไรก็ตามอัลกอริทึมตัวเลขคอนสไปเรซีเป็นอีกทางเลือกหนึ่งที่น่าสนใจ ในแง่ของการสร้างตาเดินที่มีคุณภาพ หากแต่ยังมีความต้องการศึกษาเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพด้านความเร็ว และพื้นที่หน่วยความจำที่ใช้

Project Title	Computer Thai Chess
Name of the Investigator	Boonserm Kijsirikul
Year	March 1996

Abstract

This research studies and compares three searching algorithms for a Thai Chess program which are the B* algorithm, the minimax with alpha-beta pruning algorithm and the conspiracy number algorithm. We implement these algorithms for a Thai Chess program to find out which one is most suitable for the Thai Chess program. To implement the algorithms for the Thai Chess program, we define appropriate heuristic-functions which significantly affect the performance of the algorithms. We then run experiments to compare the algorithms.

The experiments are divided into 4 sets, each of which is composed of 10 games. Every set of the experiments is designed for evaluating the material value scores when nodes generated by each algorithm are nearly the same, or when each algorithm is controlled to give a move within the same limited times. The performance of each algorithm is then measured by using the developed Thai Chess program.

The experimental results reveal that 1.) the B* algorithm is not appropriate for the Thai Chess program due to the difficulty of defining heuristic function for the convergency of the algorithm; 2.) the conspiracy number algorithm outperforms the minimax with alpha-beta pruning algorithm because of its deeper search, when given the same number of nodes to be explored; and 3.) the minimax with alpha-beta pruning algorithm generates better moves than the conspiracy number algorithm does, when each algorithm must produce a move within the limited time.

In summary, the minimax with alpha-beta pruning algorithm is more suitable for the Thai Chess program than the conspiracy number algorithm in terms of searching speed and memory used. However, conspiracy number algorithm is an interesting alternative as a quality of move obtained but further study is required to improve speed and memory.

สารบัญ

หน้า

กิตติกรรมประกาศ ii

บทคัดย่อ - ภาษาไทย iii

- ภาษาอังกฤษ iv

สารบัญ v

รายการตารางประกอบ vii

รายการภาพประกอบ viii

1. บทนำ 1

2. แนวความคิดของการประดิษฐ์ 3

2.1 กฎของหมากรุกไทย 3

2.2 อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา 6

2.3 อัลกอริทึม B* 9

2.4 อัลกอริทึม ตัวเลขคอนสไปเรซี 13

3. วิธีการประดิษฐ์ 18

3.1 การออกแบบโครงสร้างของโปรแกรมหมากรุกไทย 18

3.2 การเขียนโปรแกรมอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย 19

3.2.1 ฟังก์ชันฮิวริสติกของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย 20

3.2.2 พารามิเตอร์ที่กำหนดของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย 22

3.3 การเขียนโปรแกรมอัลกอริทึม B*ในหมากรุกไทย 22

3.3.1 ฟังก์ชันฮิวริสติกของอัลกอริทึม B* ในหมากรุกไทย 22

3.3.2 พารามิเตอร์ที่กำหนดของอัลกอริทึม B* ในหมากรุกไทย 23

3.4 การเขียนโปรแกรมอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย 23

3.4.1 ฟังก์ชันฮิวริสติกของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย 23

3.4.2 พารามิเตอร์ที่กำหนดของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย 26

4. การทดสอบการใช้งานและการวิเคราะห์ผล 27

4.1 การทดสอบและวิเคราะห์โปรแกรมอัลกอริทึมทั้ง 3 ในหมากรุกไทย 27

4.1.1 การวิเคราะห์ผลที่ได้ของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา

ในหมากรุกไทย 27
4.1.2 การวิเคราะห์ผลที่ได้ของอัลกอริทึม B* 27
4.1.3 การวิเคราะห์ผลของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย 28
4.2 การทดลองเพื่อเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา และอัลกอริทึมตัวเลขคอนสไปเรซีและการวิเคราะห์ผล 28
4.2.1 ผลการทดลองเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา และอัลกอริทึมตัวเลขคอนสไปเรซี 28
4.2.2 วิเคราะห์ผลการทดลองเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดย อัลฟาเบตาและอัลกอริทึมตัวเลขคอนสไปเรซี 33
5. ข้อสรุป 35
6. ข้อเสนอแนะ 36

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการตารางประกอบ

หน้า

ตารางที่ 1 ตัวเลขคอนสไปเรซีและโหนดที่ต้องเปลี่ยนค่า 14
ตารางที่ 2 คะแนนในการป้องกันหมากของฝ่ายเดียวกัน 24
ตารางที่ 3 แสดงคะแนนของแต่ละพื้นที่ที่ล้อมรอบขุนฝ่ายตรงข้าม 25
ตารางที่ 4 ผลการทดลองชุดที่ 1 31
ตารางที่ 5 ผลการทดลองชุดที่ 2 32
ตารางที่ 6 ผลการทดลองชุดที่ 3 32
ตารางที่ 7 ผลการทดลองชุดที่ 4 33

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการภาพประกอบ

หน้า

รูปที่ 1 กระดานหมากรุกไทย 3

รูปที่ 2 ตำแหน่งเริ่มต้นของตัวหมาก 4

รูปที่ 3 กฎการเดินของ “ขุน” 4

รูปที่ 4 กฎการเดินของ “เรือ” 5

รูปที่ 5 กฎการเดินของ “ม้า” 5

รูปที่ 6 กฎการเดินของ “โคน” 5

รูปที่ 7 กฎการเดินของ “เม็ด” 5

รูปที่ 8 (ก) กฎการเดินเพื่อเปลี่ยนตำแหน่งของ “เบี้ย” 6

รูปที่ 8 (ข) กฎการเดินเพื่อกินของ “เบี้ย” 6

รูปที่ 9 ต้นไม้เกม(ค้นหาล่วงหน้า 2 ก้าว) 6

รูปที่ 10 การส่งค่ากลับ(ค้นหาล่วงหน้า 2 ก้าว) 7

รูปที่ 11 การลดทอนโดยอัลฟา 8

รูปที่ 12 การลดทอนโดยอัลฟาเบตา 9

รูปที่ 13 จุดเริ่มต้นของการค้น B* 10

รูปที่ 14 ยุทธวิธี PROVEBEST 10

รูปที่ 15 ยุทธวิธี DISPROVEREST 10

รูปที่ 16 ต้นไม้เกม 13

รูปที่ 17 ตัวเลขคอนสไปเรซีสำหรับต้นไม้เกมในรูปที่ 16 15

รูปที่ 18 การกระจายโหนด E 16

รูปที่ 19 การกระจายโหนด D 17

รูปที่ 20 โครงสร้างของโปรแกรมหมากรุกไทย 18

รูปที่ 21 ส่วนติดต่อกับผู้ใช้ 19

รูปที่ 22 พื้นที่ที่ล้อมรอบขุนฝ่ายตรงข้าม 25

รูปที่ 23 รูปแบบที่ 1 ของสมุดมาตรฐาน 29

รูปที่ 24 รูปแบบที่ 2 ของสมุดมาตรฐาน 30

1. บทนำ

ปัญญาประดิษฐ์เป็นสาขาวิชาที่มีจุดมุ่งหมายเพื่อพัฒนาคอมพิวเตอร์ให้สามารถทำงานหนึ่งๆที่เดิมต้องใช้คนทำ (E. Rich & K. Knight, 1991) สาขานี้ได้รับความสนใจอย่างมากในต่างประเทศ แต่เนื่องจากเป็นสาขาค่อนข้างใหม่จึงมีผู้เชี่ยวชาญหรือผู้ที่ให้ความสนใจในประเทศน้อยเมื่อเทียบกับสาขาอื่นทางคอมพิวเตอร์ งานวิจัยนี้จึงมีความสำคัญที่จะศึกษาเทคนิคที่ใช้ในปัญญาประดิษฐ์เพื่อที่จะเป็นเครื่องมือในการเรียนการสอนในวิชานี้ และกระตุ้นให้เกิดความสนใจและการศึกษาทางด้านนี้เพิ่มขึ้น

ได้มีการนำเทคนิคทางปัญญาประดิษฐ์มาประยุกต์ใช้ในหลายด้านเช่น ระบบผู้เชี่ยวชาญ การประมวลผลภาษาธรรมชาติ เป็นต้น และการประยุกต์ใช้อย่างหนึ่งที่ได้รับความสนใจอย่างมาก คือ โปรแกรมหมากรุก ซึ่งเกมหมากรุกนี้เป็นเกมที่มีความซับซ้อนมาก จนไม่สามารถสร้างต้นไม้เกม (game tree) ซึ่งแสดงตาเดินในคอมพิวเตอร์ได้ทั้งหมด "เป็นที่ประมาณว่าต้นไม้-เกมของหมากรุกมีจำนวนโหนดมากกว่า 10^{100} โหนด แม้ว่าจะใช้คอมพิวเตอร์ที่มีความสามารถในการสร้าง 10^{11} โหนดต่อวินาที การสร้างต้นไม้เกมที่สมบูรณ์สำหรับเกมหมากรุกต้องใช้เวลามากกว่า 10^{80} ปี" (E.Horowitz, S. Sahni, 1978) จะเห็นได้ว่าเกมหมากรุกซึ่งเป็นเกมที่มีต้นไม้เกมขนาดใหญ่นั้น ยังมีความต้องการทางเทคนิค หรือวิธีการในการสร้างตาเดินที่มีประสิทธิภาพ การประเมินทางเลือกของตาเดินที่ดี และการเลือกตาเดินที่เหมาะสม

โปรแกรมประเภทหมากรุกนี้ในต่างประเทศมีการศึกษาอย่างมากจนถึงระดับที่มีความสามารถทัดเทียมกับคน เช่น โปรแกรมหมากรุกสากล "Deep Thought" (Hsu et al., 1990) หรือโปรแกรม checkers "Chinook" (Schaeffer et al., 1993) ซึ่งใช้เทคนิคต่างๆทางปัญญาประดิษฐ์มากมาย สำหรับโปรแกรมหมากรุกไทยที่จะพัฒนาขึ้นนี้ ในงานวิจัยนี้จะนำเทคนิคการค้นหาทางปัญญาประดิษฐ์ต่างๆที่มีอยู่ใช้เขียนเป็นโปรแกรม แล้วทำการวิเคราะห์เพื่อเลือกอัลกอริทึมที่เหมาะสมสำหรับโปรแกรมหมากรุกไทย

การค้นหาทางปัญญาประดิษฐ์ที่ใช้ในโปรแกรมหมากรุกที่รู้จักอย่างแพร่หลายคือ อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา (minimax with alpha-beta pruning) นอกจากอัลกอริทึมนี้แล้วยังมีอัลกอริทึมการค้นหาใหม่ๆเช่น อัลกอริทึมตัวเลขคอนสไปเรซี (conspiracy number) อัลกอริทึม B* (B* algorithm) ในงานวิจัยนี้เราเลือกศึกษาการประยุกต์ใช้ทั้งสามในโปรแกรมหมากรุกไทย แม้ว่าจะได้มีผู้ศึกษาวิจัยอัลกอริทึมทั้งสามในหมากรุกสากลอย่างมากแล้วก็ตาม เราก็ไม่สามารถนำผลมาใช้กับหมากรุกไทยได้โดยตรง โดยเฉพาะฟังก์ชันฮิวริสติกที่ใช้ในอัลกอริทึมหนึ่งๆสำหรับหมากรุกไทยก็แตกต่างไปจากของหมากรุกสากล ซึ่งฟังก์ชันฮิวริสติกนี้ก็เป็นสิ่งสำคัญอย่างหนึ่งที่จะกำหนดความสามารถของโปรแกรมหมากรุก เราได้นิยามฟังก์ชันฮิวริสติก

สำหรับอัลกอริทึมทั้งสามในหมากรุกไทยซึ่งประกอบด้วยส่วนสำคัญ 3 ส่วนคือ 1) คะแนนของตัวหมากที่เหลือในกระดาน 2) คะแนนในการป้องกันหมากของฝ่ายเดียวกัน และ 3) คะแนนของการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม แม้ว่าคะแนนทั้งสามส่วนจะเป็นคะแนนที่ใช้กันทั่วไปสำหรับอัลกอริทึมค้นหาในหมากรุกสากล แต่การที่จะนิยามค่าสำหรับตัวหมากในหมากรุกไทยเพื่อให้โปรแกรมสามารถเดินหมากได้อย่างดีนั้น ก็ต้องอาศัยการทดลอง การปรับค่าที่ค่อนข้างยุ่งยาก จากการทดลองเราได้สรุปและกำหนดค่าที่เหมาะสมสำหรับฟังก์ชันฮิวริสติกของอัลกอริทึมทั้งสามในหมากรุกไทย นอกจากนั้นเราได้ออกแบบโปรแกรมเพื่อเปรียบเทียบอัลกอริทึมในหมากรุกไทยโดยให้โปรแกรมสามารถเลือกอัลกอริทึมเพื่อแข่งขันกันเองได้

ผลการวิจัยพบว่า 1.)อัลกอริทึม B* ไม่เหมาะสมกับหมากรุกไทย เนื่องจากการนิยามฮิวริสติกฟังก์ชันสำหรับการลู่เข้าของอัลกอริทึมเป็นไปได้ยาก 2.)อัลกอริทึมตัวเลขคอนสไปเรซีมีข้อดีที่เลือกกระจายโหนดเฉพาะบางเส้นทางที่สนใจและไม่ถูกจำกัดด้วยความลึกทำให้สร้างโหนดจำนวนน้อยแต่มีคุณภาพสำหรับตัดสินใจเลือกตาเดินที่ดีได้ 3.) ส่วนอัลกอริทึมมินิแมกซ์+การลด-ทอนโดยอัลฟาเบตามีจุดดีที่สามารถสร้างโหนดได้รวดเร็วกว่า

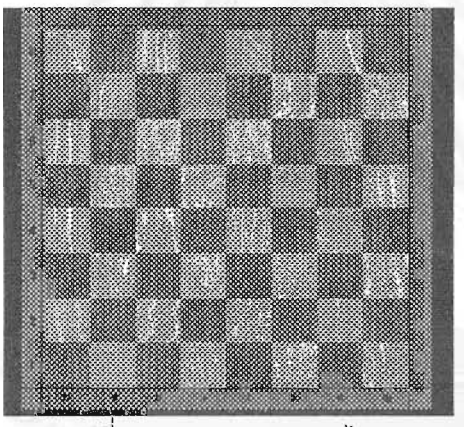
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. แนวความคิดของการประดิษฐ์

ในบทนี้เราจะกล่าวถึงแนวความคิดและทฤษฎีที่เกี่ยวข้องกับการค้นหาในโปรแกรมหมากรุก ซึ่งแบ่งออกเป็น 4 ส่วนคือ 1. กฎของหมากรุกไทย 2. อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา 3. อัลกอริทึม B* และ 4.อัลกอริทึมตัวเลขคอนสไปเรซี

2.1 กฎของหมากรุกไทย

หมากรุกไทยจัดเป็นเกมบนกระดานชนิดหนึ่ง ที่มีผู้เล่น 2 ฝ่าย ผลัดกันเดินหมากคราวละหนึ่งตา เดินจนกว่าฝ่ายใดฝ่ายหนึ่งจะสามารถกินขุนของฝ่ายตรงข้ามได้ก็จะถือว่าเป็นผู้ชนะ โดยที่มีอุปกรณ์การเล่นคือ กระดานหมากรุกซึ่งใช้เป็นตัวจำกัดตำแหน่งของการเดิน กระดานหมากรุกไทยจะมีลักษณะเป็นรูปสี่เหลี่ยมจัตุรัส กว้าง 8 ช่อง ยาว 8 ช่อง แต่ละช่องมีลักษณะเป็นสี่เหลี่ยมจัตุรัสเช่นเดียวกันดังแสดงในรูปที่ 1 ด้านล่างนี้



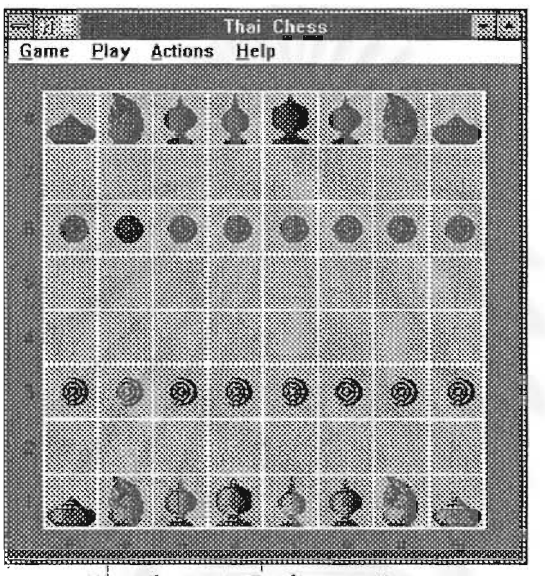
รูปที่ 1 กระดานหมากรุกไทย

ตัวหมากรุกไทยมีทั้งหมด 5 ประเภท คือ ขุน เรือ ม้า โคน เม็ด และ เบี้ย โดยที่แต่ละตัวหมากจะมีลักษณะการเดินเฉพาะเพื่อเปลี่ยนตำแหน่งหมากของตัวเอง หรือเดินเพื่อกินหมากของฝ่ายตรงข้าม

-- การเดินเฉพาะเพื่อการเปลี่ยนตำแหน่งตัวหมากจะทำได้ก็ต่อเมื่อไม่มีหมากตัวอื่นไม่ว่าจะเป็นของฝ่ายตนเองหรือฝ่ายตรงข้ามวางขวางอยู่บนตำแหน่งของตารางที่กำลังจะเดินไป และการเดินนั้นจะต้องถูกต้องตามกฎการเดินสำหรับหมากตัวนั้น ๆ นอกจากนี้เมื่อเบี้ยของฝ่ายใดฝ่ายหนึ่งสามารถเดินขึ้นตรงได้เป็นจำนวน 3 ช่องตาราง เบี้ยตัวนั้นจะถูกเปลี่ยนให้เป็นเบี้ยหงายซึ่งมีลักษณะการเดินเช่นเดียวกับเม็ด เฉพาะสำหรับเรือเท่านั้นที่ไม่สามารถเดินข้ามตัวหมากอื่นๆ ได้ไม่ว่าจะเป็นหมากของฝ่ายตนเองหรือฝ่ายตรงข้าม

– การเดินเพื่อกินตัวหมากของฝ่ายตรงข้ามจะสามารถทำได้ก็ต่อเมื่อ มีตัวหมากของฝ่ายตรงข้ามอยู่บนตารางตำแหน่งที่กำลังจะเดินไป และการเดินนั้นต้องถูกต้องตามกฎการเดินของหมากตัวนั้น

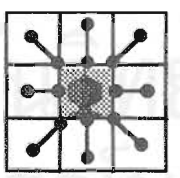
ในการเริ่มต้นเล่นหมากรุกไทยตามปกติทั่วๆไป จะมีจำนวนตัวหมากเริ่มต้นของแต่ละฝ่าย และมีการวางตำแหน่งเริ่มต้นของตัวหมากของแต่ละฝ่ายดังแสดงในรูปที่ 2



รูปที่ 2 ตำแหน่งเริ่มต้นของตัวหมาก

กฎการเดินตัวหมากในหมากรุกไทยเป็นดังต่อไปนี้

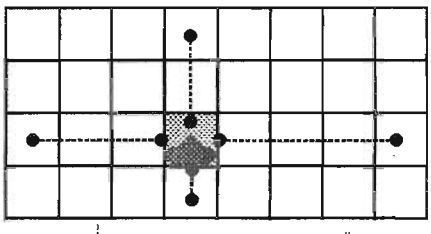
ขุน สามารถเดินเพื่อเปลี่ยนตำแหน่งหรือกินได้ไกลไม่เกินคราวละหนึ่งตาราง และสามารถเดินได้รอบตัวทั้ง 8 ทิศ ดังรูปที่ 3



รูปที่ 3 กฎการเดินของ “ขุน”

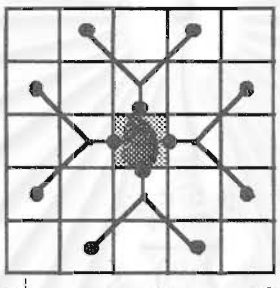
เรือ สามารถเดินเพื่อเปลี่ยนตำแหน่งหรือกินได้ไกลมากกว่าหนึ่งตาราง และเดินในแนวตรงนอนหรือแนวตั้งเท่านั้น ห้ามเดินในแนวเฉียงดังรูปที่ 4

หอสมุดกลาง ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



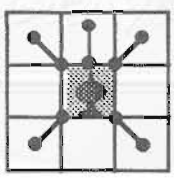
รูปที่ 4 กฎการเดินของ “เรือ”

ม้า สามารถเดินเพื่อเปลี่ยนตำแหน่งหรือกินโดยที่จะเดินตามแนวตรงก่อนแล้วจึงเดินในแนวเฉียง (หรืออาจเดินในแนวเฉียงก่อนแล้วเดินแนวตรงก็ได้) ดังแสดงรูปที่ 5



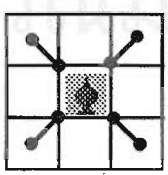
รูปที่ 5 กฎการเดินของ “ม้า”

โคน สามารถเดินเพื่อเปลี่ยนตำแหน่งหรือกินได้ไม่เกินคราวละหนึ่งตาราง และเดินได้ในแนวขึ้นและแนวเฉียง 4 ทิศดังรูปที่ 6



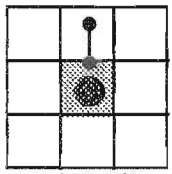
รูปที่ 6 กฎการเดินของ “โคน”

เม็ด สามารถเดินเพื่อเปลี่ยนตำแหน่งหรือกินได้ไม่เกินคราวละหนึ่งตาราง และเดินได้ในแนวเฉียงทั้ง 4 ทิศดังรูปที่ 7

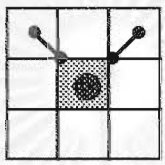


รูปที่ 7 กฎการเดินของ “เม็ด”

เบี้ย สามารถเดินเพื่อเปลี่ยนตำแหน่งได้ไกลไม่เกินคราวละหนึ่งตารางโดยที่ต้องเดินในแนวขึ้นตรงดังรูป และสามารถเดินเพื่อกินได้เฉพาะในแนวเฉียงขึ้นบน 2 ทิศ ดังรูปที่ 8 (ก) และ 8 (ข) ตามลำดับ



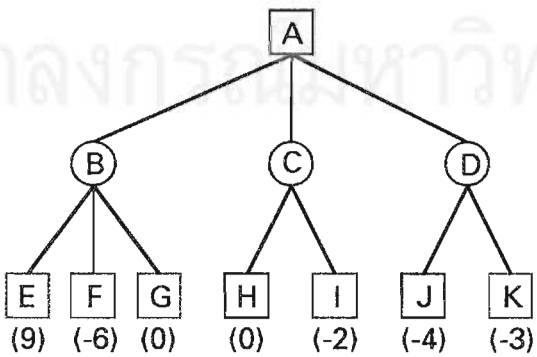
รูปที่ 8 (ก) กฎการเดินเพื่อเปลี่ยนตำแหน่งของ "เบี้ย"



รูปที่ 8 (ข) กฎการเดินเพื่อกินของ "เบี้ย"

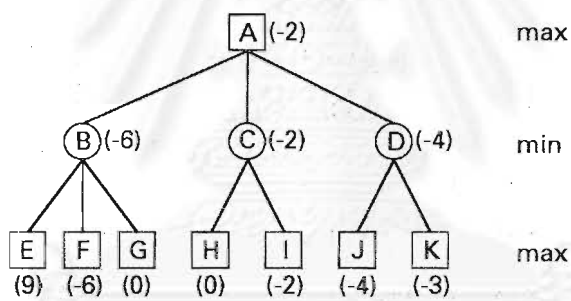
2.2 อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา

อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา (E.Rich & K.Knight, 1990) (M.A.Weiss, 1993) เป็นกระบวนการค้นหาตามแนวลึกก่อนโดยมีการกำหนดความลึกสูงสุด อัลกอริทึมเริ่มค้นหาจากโหนดราก(โหนดเริ่มต้น)ลึกล่วงหน้าหลายก้าวตามที่กำหนดไว้พร้อมกับสร้างโหนดใหม่ขึ้นมา โหนดแต่ละโหนดแทนกระดานหมากรุกที่ขณะหนึ่งๆ โหนดที่ปลายของต้นไม้เรียกว่าโหนดใบ(leaf node) ต้นไม้ที่ได้จากกระบวนการค้นหาเรียกว่าต้นไม้เกม รูปที่ 9 แสดงต้นไม้เกมหนึ่งในกรณีที่ค้นหาลึกล่วงหน้า 2 ก้าว ที่โหนดใบอัลกอริทึมจะคำนวณค่าสถิต(static value) หรือเรียกว่าค่าฮิวริสติก เพื่อหาว่าโหนดนั้นมีโอกาสชนะเท่าไร ค่าสถิตนี้โดยทั่วไปคำนวณจากจำนวนเบี้ยในกระดานและตำแหน่งของเบี้ยของฝ่ายเราและฝ่ายตรงข้าม ตัวเลขในรูปที่ 9 แสดงค่าสถิตของโหนดใบต่างๆ โหนดสี่เหลี่ยมในรูปแทนฝ่ายเราและโหนดวงกลมแทนฝ่ายตรงข้าม



รูปที่ 9 ต้นไม้เกม(ค้นหาล่วงหน้า 2 ก้าว)

ค่าสถิตที่ได้จะถูกส่งกลับไปยังโหนดรากเพื่อประเมินค่าโหนดรากว่ามีโอกาสชนะหรือคะแนนเท่าไร ในกรณีของโหนดสี่เหลี่ยมจะต้องพยายามเลือกตาเดินที่ทำให้โอกาสชนะของตนเองมากสุด ในกรณีนี้โหนด A จะต้องเลือกตาเดินที่ทำให้เกิดโหนด B หรือ C หรือ D อย่างใดอย่างหนึ่ง ดังนั้นถ้าเรารู้ค่าสถิตที่โหนดทั้งสาม ก็จะสามารถตัดสินใจได้ว่า A ควรเลือกโหนดใด โดยจะเลือกโหนดที่มีค่าสถิตสูงสุด (โอกาสชนะของฝ่ายเรามากสุด) ในกรณีของโหนดวงกลมก็เช่นเดียวกันจะต้องเลือกตาเดินที่ทำให้ตนเองมีโอกาสชนะมากสุดหรือทำให้ค่าสถิตที่โหนดรากมีค่าน้อยสุด ตัวอย่างเช่นโหนด B มีโอกาสเลือก E หรือ F หรือ G ซึ่งจะต้องเลือกให้ค่าที่โหนดตนเองมีค่าน้อยสุด (เพราะต้นไม้เกมนี้เริ่มจากโหนดสี่เหลี่ยมดังนั้นค่าที่โหนดต่างๆจึงเป็นโอกาสที่โหนดสี่เหลี่ยมจะชนะ) ในกรณีนี้โหนด B ควรเลือกโหนด F เมื่อ F ถูกเลือกค่าสถิตของ F จะถูกส่งกลับมาให้ B ซึ่งเราเรียกค่าที่ B นี้ว่าค่ามินิแมกซ์ เราเรียกผู้เล่นที่แทนด้วยโหนดสี่เหลี่ยมว่า max ซึ่งเป็นผู้เล่นที่พยายามทำให้ค่าของโหนดตัวเองสูงสุด และเรียกผู้เล่นที่แทนด้วยโหนดวงกลมว่า min ซึ่งเป็นผู้เล่นที่พยายามทำให้ค่าของโหนดตัวเองต่ำสุด รูปที่ 10 แสดงการส่งค่ากลับจากโหนดใบไปยังโหนดราก



รูปที่ 10 การส่งค่ากลับ(ค้นหาล่วงหน้า 2 ก้าว)

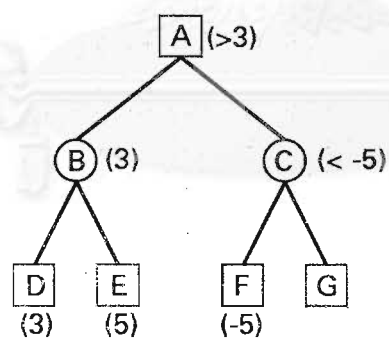
อัลกอริทึมจะค้นหาล่วงหน้าเท่ากับความลึกที่กำหนดไว้ โดยค้นหาโหนดในแนวลึกก่อน ทุกครั้งที่สร้างโหนดใบค่าสถิตจะถูกส่งกลับไปยังโหนดที่อยู่ด้านบนหรือโหนดที่ระดับสูงกว่า เพื่อแก้ไขค่ามินิแมกซ์ของโหนดที่อยู่ด้านบนนั้นรวมทั้งโหนดรากด้วย ในการเล่นเกมหมากรุกโดยคอมพิวเตอร์นั้น ถ้าอัลกอริทึมสามารถค้นหาล่วงหน้าได้ลึกเท่าไร ความถูกต้องของค่ามินิแมกซ์ที่แสดงโอกาสชนะของโหนดรากก็จะยิ่งถูกต้องมากขึ้น แต่เมื่อเราสร้างต้นไม้เกมที่มีความลึกมากขึ้นจำนวนโหนดก็จะเพิ่มขึ้นเป็นเลขยกกำลัง ทำให้ประสิทธิภาพของอัลกอริทึมมินิแมกซ์ต่ำ อัลกอริทึมการลดทอนโดยอัลฟาเบตาเพิ่มประสิทธิภาพของอัลกอริทึมมินิแมกซ์โดยการช่วยลดการสร้างโหนดที่ไม่จำเป็น ซึ่งมีวิธีการดังต่อไปนี้

อัลกอริทึมการลดทอนโดยอัลฟาเบตาจะกำหนดให้มีค่าอัลฟา(α) ซึ่งแสดงขอบเขตล่างของค่ามินิแมกซ์ที่โหนด max และค่าเบตา(β) ซึ่งแสดงขอบเขตบนของค่ามินิแมกซ์ที่โหนด min แล้วใช้ 2 ค่านี้ในการลดทอนการสร้างโหนดที่ไม่จำเป็น พิจารณาต้นไม้เกมในรูปที่ 11 หลังจากที่โหนด F ถูกสร้างขึ้น เราจะได้ว่าฝ่ายตรงข้ามหรือ min จะมีค่าอย่างน้อยที่สุดคือ -5 ที่โหนด C และเราประกันได้ว่าฝ่ายเราหรือ max จะได้ค่าอย่างน้อยที่สุดคือ 3 ที่โหนด A (โดยการเลือกโหนด B) ถ้าโหนดอื่นๆที่มีค่าน้อยกว่า 3 (เช่นโหนด C) max ก็จะไม่เลือกโหนดเหล่านั้น ดังนั้นหลังจากที่เราสร้างโหนด F เราสามารถแน่ใจได้ว่าโหนด C ไม่ดีกว่าโหนด B โดยไม่ต้องสนใจว่าโหนด G จะมีค่าเท่าไร ที่จริงแล้วเราไม่จำเป็นต้องสร้างโหนด G เลย ซึ่งในกรณีนี้ 3 คือค่าขอบเขตล่างที่โหนด A หรือค่า α นั่นเอง

การลดทอนโดยอัลฟาสามารถสรุปเป็นกฎได้คือ

"ให้ α ของโหนด max คือขอบเขตล่างสำหรับโหนดนั้น ถ้าพบว่าค่าของโหนด min น้อยกว่าหรือเท่ากับ α ของโหนดพ่อแม่แล้ว ให้หยุดการสร้างและคำนวณค่าโหนดลูกที่เหลือของโหนด min นั้น"

ในตัวอย่างรูปที่ 11 จะเห็นว่าการลดทอนทำได้เพียงแค่โหนด G โหนดเดียว แต่ถ้าต้นไม้เกมนี้มีขนาดใหญ่เช่นลึกมากกว่านี้และแต่ละโหนดมีโหนดลูกมากกว่านี้แล้วผลของการลดทอนจะเด่นชัดยิ่งขึ้น

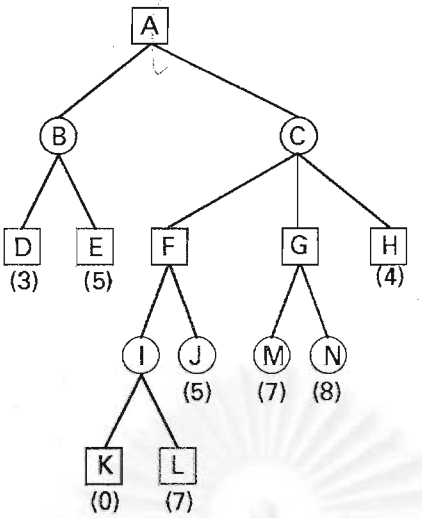


รูปที่ 11 การลดทอนโดยอัลฟา

ในทำนองเดียวกันเราได้กฎสำหรับการลดทอนโดยเบตาคือ

"ให้ β ของโหนด min คือขอบเขตบนสำหรับโหนดนั้น ถ้าพบว่าค่าของโหนด max มากกว่าหรือเท่ากับ β ของโหนดพ่อแม่แล้ว ให้หยุดการสร้างและคำนวณค่าโหนดลูกที่เหลือของโหนด max นั้น"

รูปที่ 12 แสดงการใช้ α และ β ร่วมกันลดทอนการสร้างโหนดที่ไม่จำเป็น



รูปที่ 12 การลดทอนโดยอัลฟาเบตา

ในรูปที่ 12 เริ่มแรกต้นไม้ย่อยที่มีโหนด B เป็นโหนดต้นถูกสร้างขึ้น ที่จุดนี้เราพบว่าอย่างน้อยที่สุดโหนด A จะได้ค่าเป็น 3 (หรือมากกว่า) ซึ่ง 3 นี้คือค่า α เมื่อ α ถูกส่งลงมาให้ F ก็จะทำให้เราสามารถตัดการสร้างและกระจายโหนด L ได้ ที่เป็นเช่นนี้เพราะว่า เมื่อ K ถูกสร้างขึ้นเราพบว่า I มีขอบเขตบนคือ 0 ซึ่งหมายถึง F มีขอบเขตล่างคือ 0 แต่ค่า 0 นี้น้อยกว่า α (=3) ดังนั้นโหนดอื่นๆที่เป็นโหนดลูกของ I จึงสามารถตัดทิ้งได้ ในทำนองเดียวกันเราสามารถแสดงผลของการลดทอนโดย β ได้ดังนี้ สมมุติว่าที่จุดนี้โหนด J ถูกสร้างขึ้นมีค่าสถิตเท่ากับ 5 ซึ่งทำให้ F มีค่ามินิแมกซ์เป็น 5 ด้วย ค่า 5 นี้เป็นค่า β ที่โหนด C หมายความว่าที่ C ประกันได้ว่าจะได้ค่ามากสุดคือ 5 (min พยายามทำให้ค่าที่ C ต่ำสุด) ที่จุดนี้โหนด G ถูกกระจายได้โหนดแรกเป็น M มีค่าสถิตเท่ากับ 7 ค่า 7 ถูกส่งกลับไปยัง G และถูกเปรียบเทียบกับค่า β (=5) ผลการเปรียบเทียบได้ 7 มีค่ามากกว่า ดังนั้นโหนดอื่นๆที่เป็นโหนดลูกของ G เช่น N จะไม่ถูกสร้างขึ้น

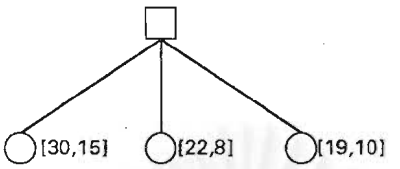
2.3 อัลกอริทึม B*

อัลกอริทึม B* (H. Berliner,1979,1991) ใช้ค่าประเมิน 2 ค่า คือ ขอบเขตบน(optimistic) และขอบเขตล่าง(pessimistic) ค่าของโหนดในต้นไม้ย่อยจะอยู่ระหว่างช่วงของขอบเขตทั้งสองนี้ เมื่อโหนดในระดับล่างถูกกระจายเพิ่มขึ้นจะทำให้ช่วงกว้างนี้ลดลง จนกระทั่งเหลือเพียงค่าเดียว ซึ่งเป็นเงื่อนไขของการสิ้นสุดของอัลกอริทึมในการหากิ่งที่ดีที่สุด ที่ขณะหนึ่งๆอัลกอริทึมจะเลือกยุทธวิธี(strategy) หนึ่งจาก 2 ยุทธวิธีต่อไปนี้

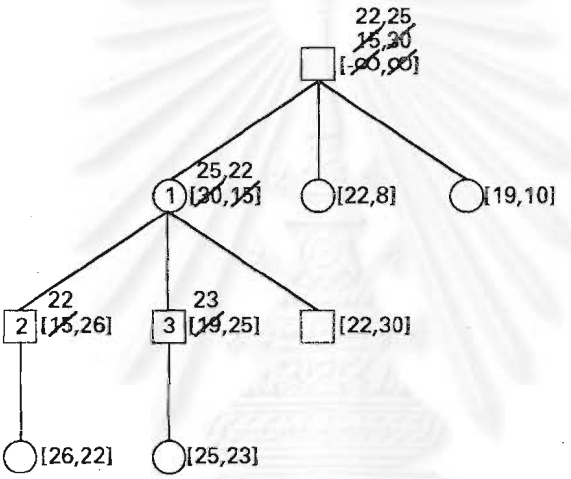
(1) พยายามเพิ่มค่าของขอบเขตล่างของโหนดที่มีขอบเขตบนสูงสุด เพื่อที่จะทำให้ขอบเขตล่างนั้นมีค่ามากกว่าขอบเขตบนของโหนดพี่น้อง(โหนดในระดับเดียวกัน)ทั้งหมด ยุทธวิธีนี้เรียกว่า PROVEBEST

(2) พยายามลดค่าของขอบเขตบนของโหนดพี่น้อง เพื่อที่จะทำให้ขอบเขตบนเหล่านั้นมีค่าน้อยกว่าขอบเขตล่างของที่มีค่าขอบเขตบนสูงสุด ยุทธวิธีนี้เรียกว่า DISPROVEREST

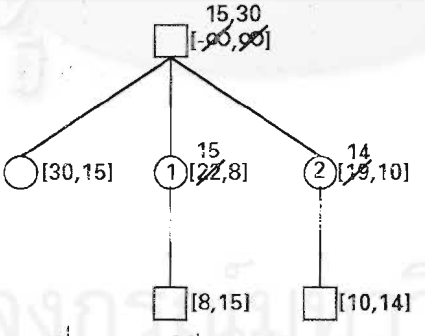
ตัวอย่างด้านล่าง (รูปที่ 13-15) แสดงต้นไม้เกมและการใช้ยุทธวิธีทั้งสอง ในรูปโหนดซ้ายมือสุดเป็นโหนดที่มีขอบเขตบนสูงสุด



รูปที่ 13 จุดเริ่มต้นของการค้น B*



รูปที่ 14 ยุทธวิธี PROVEBEST



รูปที่ 15 ยุทธวิธี DISPROVEREST

โหนดสี่เหลี่ยมในรูปแทนตำแหน่งของ max ส่วนโหนดวงกลมแทนตำแหน่งของ min ตัวเลขแสดงลำดับของโหนดที่ถูกกระจาย เมื่อโหนดถูกกระจายค่าของมันจะถูกส่งกลับไปและถูกแทนที่ในโหนดพ่อแม่ซึ่งแสดงไว้เหนือตัวเลขในวงเล็บ จะเห็นได้ว่าโหนดที่กระจายในแต่ละยุทธวิธีไม่เท่ากัน ซึ่งในตัวอย่างด้านบน DISPROVEREST จะมีประสิทธิภาพดีกว่าคือกระจายโหนดน้อยกว่า

อัลกอริทึม B* ทำงานโดยเลือกยุทธวิธี PROVEBEST หรือ DISPROVEREST ที่โหนดราก จากนั้นอัลกอริทึมจะค้นหาจนกระทั่งพบโหนดใบแล้วกระจายโหนดใบเหล่านั้น ในกรณีที่โหนดใหม่ที่ได้จากการกระจายมีค่าที่ทำให้ช่วงค่าของโหนดพ่อแม่เปลี่ยนไป ค่าเหล่านั้นจะถูกส่งกลับไปให้โหนดพ่อแม่และเปลี่ยนช่วงค่าของโหนดพ่อแม่ การส่งค่าจะถูกกระทำขึ้นไปทีละระดับในต้นไม้เกมจนกระทั่งการเปลี่ยนแปลงสิ้นสุด กระบวนการค้นหาจะสิ้นสุดเมื่อขอบเขตล่างของโหนดที่ดีที่สุดมีค่ามากกว่าขอบเขตบนของโหนดพี่น้อง

ในขณะที่มีการส่งค่ากลับขอบเขตบนที่ดีที่สุดของโหนดลูกๆจะเป็นขอบเขตล่างของโหนดพ่อแม่ และขอบเขตล่างที่น้อยที่สุดของโหนดลูกๆจะเป็นขอบเขตบนของโหนดพ่อแม่ สำหรับ max แล้ว ขอบเขตบนจะมีค่ามากกว่าขอบเขตล่าง แต่สำหรับ min ขอบเขตล่างจะมีค่ามากกว่าขอบเขตบน

สำหรับการเลือกยุทธวิธีและโหนดที่จะกระจายนั้นได้มีการศึกษาใน (A.J.Palay, 1982) ซึ่งสรุปได้ดังต่อไปนี้

ในการเลือกยุทธวิธีและโหนดที่จะกระจาย เราจะสมมุติให้ที่โหนดใดๆมีโหนดลูกเพียง 2 โหนด สำหรับกรณีที่มีโหนดใดๆมีโหนดลูกเกิน 2 โหนดสามารถทำได้ในลักษณะเดียวกัน ให้ N1 และ N2 เป็นโหนดลูก 2 โหนดของโหนดใดๆ ช่วงค่าของโหนด N1 และ N2 เป็น [A1,B1] และ [A2,B2] ตามลำดับ โดยที่ Ai > Bi และสมมุติว่า A1 $\geq$ A2 > B1 ดังนั้นมีกรณีที่เป็นไปได้ 6 กรณีในการเลือกยุทธวิธีสำหรับการค้นหาและโหนดที่จะกระจาย

-- กรณีที่ 1 : A1 = A2 และ B1 > B2

-- กรณีที่ 2 : A1 = A2 และ B1 = B2

-- กรณีที่ 3 : A1 = A2 และ B1 < B2

-- กรณีที่ 4 : A1 > A2 และ B1 > B2

-- กรณีที่ 5 : A1 > A2 และ B1 = B2

-- กรณีที่ 6 : A1 > A2 และ B1 < B2

กฎที่ใช้ในการเลือกยุทธวิธีและโหนดที่จะกระจายวิธีมี 2 กฎดังนี้

-- กฎที่ 1 : ถ้าต้องกระทำการอะไรสักอย่างในกระบวนการค้นหา ให้ทำทันที

-- กฎที่ 2 : พยายามเลือกยุทธวิธีที่ทำให้ความน่าจะเป็นของความสำเร็จรวมมีค่ามากสุด

นอกจากนี้เราตั้งสมมุติฐานว่า ค่าของโหนดหนึ่งๆมีการกระจายอย่างสม่ำเสมอตลอดช่วงกว้างของโหนดนั้น

<u>กรณีที่ 1</u>

ในกรณีนี้เราอาจพยายามพิสูจน์ว่าโหนดทั้งสองเป็นโหนดดีที่สุด แต่จากสมมุติฐานแสดงว่ามีโอกาสที่โหนดดีที่สุดเป็น N1 มากกว่า ดังนั้นจึงมีทางเลือก 2 ทางระหว่างพิสูจน์เลือก

PROVEBEST สำหรับ N1 หรือ DISPROVEREST สำหรับ N2 จากกฎที่ 2 ความน่าจะเป็นที่จะสำเร็จของ PROVEBEST มีค่าน้อยกว่าของ DISPROVEREST เพราะว่ามีหนทางเดียวที่การค้นหาโดย PROVEBEST จะสำเร็จคือพยายามเพิ่มขอบเขตล่างของโหนด N1 ดังนั้นในกรณีนี้เราจะเลือก DISPROVEREST กรณีที่ 3 สมมาตรกับกรณีที่ 1

กรณีที่ 2

โหนดทั้งสองเหมือนกันเราจึงไม่สามารถเลือกยุทธวิธีหรือโหนดที่จะกระจายได้ ดังนั้นในกรณีนี้เราจึงจำเป็นที่จะต้องลองกระจายโหนดในระดับล่างดูก่อน เพื่อที่จะได้ข้อมูลเพิ่มเติมในการตัดสินใจ

กรณีที่ 4

ในกรณีนี้ช่วงค่าของโหนดทั้งสองคาบเกี่ยวกันอยู่ เราต้องเลือกยุทธวิธีใดยุทธวิธีหนึ่งเพื่อแสดงว่าโหนด N1 เป็นโหนดดีที่สุด กฎที่ 2 บอกให้เรารู้ว่ายุทธวิธีที่จะเลือกจะต้องทำให้ความน่าจะเป็นของความสำเร็จมีค่าสูงสุดหรือความน่าจะเป็นของความล้มเหลวต่ำสุด และจากสมมุติฐานทำให้เราสามารถเลือกยุทธวิธีได้ง่ายขึ้น เราทำได้โดยคำนวณความน่าจะเป็นที่ค่าที่แท้จริงของโหนดที่ดีที่สุด (N1) จะมีค่าน้อยกว่าหรือเท่ากับขอบเขตบนของ N2 ซึ่งก็คือความน่าจะเป็นของความล้มเหลวในการเลือกยุทธวิธี PROVEBEST ค่าความน่าจะเป็นนี้เท่ากับ

$$P_{FP} = (A2 - B1)/(A1-B1)$$

สำหรับโหนด N2 เราคำนวณค่าความน่าจะเป็นของความล้มเหลวในการเลือกยุทธวิธี DISPROVEREST ความน่าจะเป็นนี้เป็นความน่าจะเป็นที่ค่าที่แท้จริงของ N2 จะมากกว่าหรือเท่ากับขอบเขตล่างของ N1 ซึ่งเท่ากับ

$$P_{FD} = (A2 - B1)/(A2-B2)$$

ถ้า P_{FP} มีค่าน้อยกว่า P_{FD} แสดงว่าควรเลือก PROVEBEST แต่ถ้า P_{FD} มีค่าน้อยกว่า P_{FP} แสดงว่าควรเลือก DISPROVEREST

กรณีที่ 5

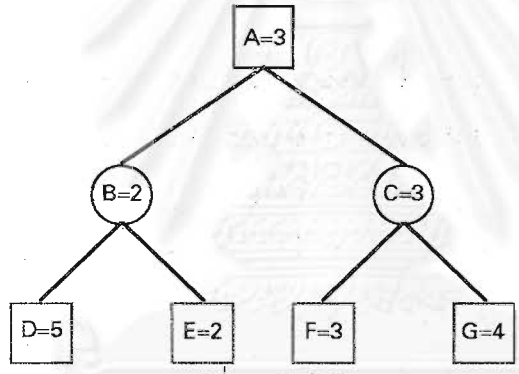
กรณีนี้เราแสดงให้เห็นว่าโหนด N1 เป็นโหนดดีที่สุดโดยยุทธวิธี PROVEBEST ซึ่งยุทธวิธีนี้จะมีโอกาสสำเร็จมากกว่า DISPROVEREST

กรณีที่ 6

สำหรับกรณีนี้ช่วงค่าของโหนด N1 จะคลุมช่วงค่าของ N2 ดังนั้นความน่าจะเป็นของความสำเร็จของยุทธวิธี DISPROVEREST จะเท่ากับ 0 คือเราไม่มีโอกาสที่จะลดขอบเขตบนของ N2 ให้น้อยกว่าของ N1 ดังนั้นยุทธวิธีที่เลือกได้มีเพียง PROVEBEST

2.4 อัลกอริทึมตัวเลขคอนสไปเรซี

อัลกอริทึมตัวเลขคอนสไปเรซีนี้ (D.A.McAllester,1988; J.Schaeffer,1990; L.Lister & J. Schaeffer,1994) จะเลือกสร้างโหนดในต้นไม้เกมโดยพยายามที่จะทำให้ช่วงของค่ามินิแมกซ์ที่เป็นไปได้สำหรับรากแคบลง ความน่าจะเป็นของรากที่จะมีค่า v หนึ่งๆแสดงโดยจำนวนโหนดใบที่น้อยที่สุดที่ต้องเปลี่ยนค่าของโหนดใบเหล่านั้นเพื่อส่งผลให้ได้ค่า v ที่ราก โหนดเหล่านี้กล่าวได้ว่าร่วมกัน (conspire) เพื่อให้ได้ผลที่ต้องการ และเราเรียกจำนวนโหนดเหล่านี้ว่าเป็น ตัวเลขคอน-สไปเรซี(CN)สำหรับค่า v นั้น อัลกอริทึมนี้จะใช้ค่าขีดแบ่ง (threshold) เพื่อเป็นเกณฑ์ในการหยุดการทำงาน ถ้าค่าของตัวเลขคอนสไปเรซีถึงค่าขีดแบ่งนี้แสดงว่ามีความน่าจะเป็นน้อยที่รากจะมีค่าเป็น v อัลกอริทึมจะหยุดก็ต่อเมื่อรากมีค่าที่เป็นไปได้ค่าเดียวและมีตัวเลขคอนสไปเรซีน้อยกว่าค่าขีดแบ่ง



รูปที่ 16 ต้นไม้เกม

ตัวอย่างในรูป 16 แสดงต้นไม้เกม โหนด A มีค่ามินิแมกซ์ 3 โหนด B มีค่า 2 โหนดอื่นๆมีค่าตามที่แสดงในรูป ถ้าเราค้นหาต่อจากโหนดใบ E ลึกต่อลงไปและพบว่า E เปลี่ยนค่าเป็น 4 เราจะได้ว่า A เปลี่ยนค่าเป็น 4 ด้วย นั่นคือจำนวนโหนดใบที่ต้องเปลี่ยนค่าเพื่อให้ค่าที่รากเปลี่ยนเป็น 4 คือ 1 หรือตัวเลขคอนสไปเรซีสำหรับ 4 คือ 1 แต่ถ้าเราต้องการเปลี่ยนค่า A เป็น 1 เราต้องเปลี่ยนค่า D หรือ E เป็น 1 และ F หรือ G เป็น 1 ในกรณีนี้จำนวนโหนดใบที่น้อยที่สุดที่ต้องเปลี่ยนค่าเพื่อให้ A เปลี่ยนเป็น 1 คือ 2 ตารางที่ 1 แสดงตัวเลขคอนสไปเรซีสำหรับค่าต่างๆของโหนดราก

ตารางที่ 1 ตัวเลขคอนสไปเรซีและโหนดที่ต้องเปลี่ยนค่า

ค่า	CN	โหนดที่ต้องเปลี่ยนค่า
1	2	(D หรือ F) และ (F หรือ G)
2	1	(F หรือ G)
3	0	--
4	1	(E หรือ F)
5	1	E
6	2	(D และ E) หรือ (F และ G)

ตัวเลขคอนสไปเรซีแสดงความน่าจะเป็นที่โหนดจะเปลี่ยนค่าไปเป็นค่าหนึ่งๆ ในรูปที่ 16 จำนวนโหนดที่ต้องเปลี่ยนค่าเพื่อให้โหนดรากมีค่าเป็น 5 คือ 1 และเพื่อให้โหนดรากมีค่าเป็น 6 คือ 2 นั่นคือโหนดรากมีโอกาสเปลี่ยนค่าเป็น 6 น้อยกว่าเปลี่ยนค่าเป็น 5

การคำนวณตัวเลขคอนสไปเรซีที่โหนดใดๆ (ไม่จำกัดว่าต้องที่โหนดราก) คำนวณได้ดังต่อไปนี้

-- กรณีของโหนดใบ : ถ้าค่ามินิแมกซ์โหนดใบคือ v แล้ว ตัวเลขคอนสไปเรซีสำหรับค่า v คือ 0 สำหรับค่ามินิแมกซ์อื่นๆตัวเลขคอนสไปเรซีจะมีค่าเป็น 1

– กรณีของโหนดภายใน : การพิจารณาตัวเลขคอนสไปเรซีแยกออกเป็น 2 กลุ่ม คือ จำนวนที่ใช้ในการเพิ่มค่าโหนด ($\uparrow$needed) และจำนวนที่ใช้ในการลดค่าโหนด($\downarrow$needed) ค่าที่อยู่ต่ำกว่าค่ามินิแมกซ์ใน $\uparrow$needed และเหนือกว่าค่ามินิแมกซ์ใน $\downarrow$needed จะมีค่าเป็น 0 ทั้ง $\uparrow$needed และ $\downarrow$needed ของโหนดลูกภายในสามารถร่วมกันเพื่อกำหนดค่าตัวเลขคอนสไปเรซี (CN) ของโหนดพ่อแม่โดยใช้สมการต่อไปนี้ (ต้องการหาตัวเลขคอนสไปเรซีสำหรับ v และ m คือค่ามินิแมกซ์)

$$CN(v) = 0 \text{ ; ถ้า } v = m$$

สำหรับโหนด max :

$$CN(v) = \sum_{\text{all son } i} \downarrow\text{needed}_i(v) \text{ ; สำหรับทุกค่าที่ } v < m$$

$$CN(v) = \min_{\text{all son } i} \uparrow\text{needed}_i(v) \text{ ; สำหรับทุกค่าที่ } v > m$$

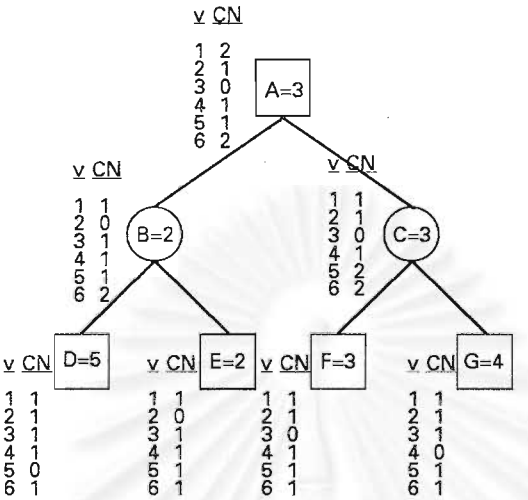
สำหรับโหนด min :

$$CN(v) = \min_{\text{all son } i} \downarrow\text{needed}_i(v) \text{ ; สำหรับทุกค่าที่ } v < m$$

$$CN(v) = \sum_{\text{all son } i} \uparrow\text{needed}_i(v) \text{ ; สำหรับทุกค่าที่ } v > m$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

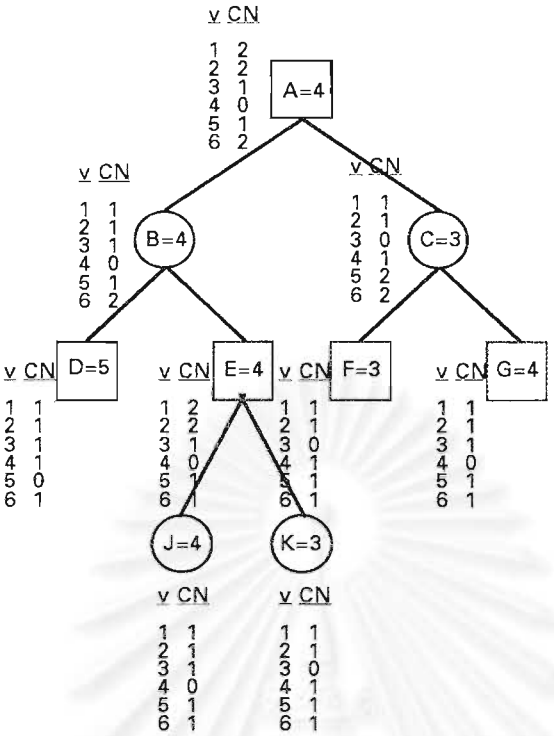
รูปที่ 17 แสดงตัวเลขคอนสไปเรซีสำหรับต้นไม้เกมในรูปที่ 16 ตัวเลขด้านซ้ายมือของแต่ละโหนดแสดงคู่ลำดับของ v (ค่า) และ CN (ตัวเลขคอนสไปเรซี) ของแต่ละโหนด



รูปที่ 17 ตัวเลขคอนสไปเรซีสำหรับต้นไม้เกมในรูปที่ 16

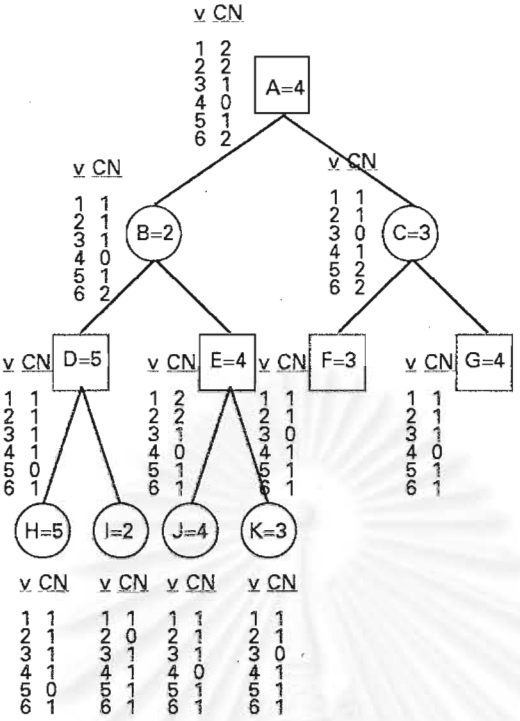
การใช้ตัวเลขคอนสไปเรซีในการค้นหานี้ทำได้โดยหลักการต่อไปนี้ ตัวเลขคอนสไปเรซีแสดงความน่าจะเป็นที่รากจะเปลี่ยนค่าเป็นค่าหนึ่งๆ ดังนั้นเราจะทำการค้นหาจนกระทั่งความน่าจะเป็นที่รากจะเปลี่ยนค่ามีน้อย ซึ่งกำหนดโดยขีดแบ่ง(CT) ถ้าตัวเลขคอนสไปเรซีสำหรับ v มากกว่าหรือเท่ากับขีดแบ่งแสดงว่าโอกาสที่จะเปลี่ยนเป็นค่า v มีน้อย และอัลกอริทึมจะหยุดที่จุดนั้น

โหนดจะถูกกระจายเพื่อที่จะเปลี่ยนค่ามากที่สุดและน้อยที่สุดของโหนดราก พิจารณาตัวอย่างในรูปที่ 17 สมมุติให้ค่าขีดแบ่งเป็น 2 เราพิจารณาว่าค่า 1 และ 6 ไม่เหมาะสมเพราะว่าค่าทั้งสองต้องการโหนดที่จะเปลี่ยนค่าอย่างน้อย 2 โหนด ค่าน้อยที่สุด(V_{min})และค่ามากที่สุด(V_{max}) ที่เป็นไปได้สำหรับโหนดรากคือ 2 และ 5 ตามลำดับ ค่าที่โหนดราก(V_{root})คือ 3 ในกรณีนี้อัลกอริทึมจะลองเปลี่ยนค่าของ V_{max} เนื่องจากมีค่าที่เป็นไปได้ที่อยู่เหนือ V_{root} มากกว่าที่อยู่ใต้ V_{root} ดังนั้นอัลกอริทึมจะกระจายโหนดเพื่อจะเปลี่ยนค่า V_{root} เป็น 5 ที่จุดนี้โหนด B มีตัวเลขคอนสไปเรซีเป็น 1 สำหรับค่า 5 ในขณะที่โหนด C มีตัวเลขคอนสไปเรซีเป็น 2 ดังนั้นอัลกอริทึมจะเลือก B จากการเลือกโหนด B อัลกอริทึมต้องตัดสินใจต่อว่าจะเลือกโหนด D หรือ E เพื่อที่จะกระจายต่อไป เนื่องจากโหนด D มีค่าที่ต้องการอยู่แล้ว ดังนั้นโหนด E จะถูกเลือก โหนด E เป็นโหนดใบอัลกอริทึมจึงค้นหาลึกลงไปอีกหนึ่งก้าว สมมุติว่าโหนดที่ถูกสร้างขึ้นที่จุดนี้คือโหนด J และ K แสดงในรูปที่ 18 ผลที่ได้คือ V_{root} เปลี่ยนเป็น 4 และช่วงของค่าที่เป็นไปได้ลดลงจาก 2-5 เป็น 3-5



รูปที่ 18 การกระจายโหนด E

ในรูปที่ 18 ช่วงค่าที่เป็นไปได้ที่อยู่เหนือและล่าง 4 เท่ากัน ดังนั้นอัลกอริทึมเลือกค่าที่ จะลองเปลี่ยนแบบสุ่ม สมมุติว่าอัลกอริทึมเลือกที่จะเปลี่ยนค่าที่โหนดรากเป็น 3 ในกรณีนี้โหนด B จะถูกเลือกเพราะว่าโหนด C ได้ค่าที่ต้องการแล้ว จากนั้นอัลกอริทึมจะเลือกระหว่างโหนด D กับโหนด E เนื่องจากโหนดทั้งสองมีตัวเลขคอนสไปเรซีสำหรับค่า 3 เท่ากัน ดังนั้นจึงต้องเลือกแบบสุ่ม สมมุติว่าโหนด D ถูกเลือก รูปที่ 19 แสดงต้นไม้เกมหลังจากโหนด D ถูกกระจาย การค้นหาจะทำในทำนองเดียวกันจนกระทั่งค่าที่โหนดรากเหลือเพียงค่าเดียวที่เป็นไปได้



รูปที่ 19 การกระจายโหนด D

จุฬาลงกรณ์มหาวิทยาลัย

3. วิธีการประดิษฐ์

การเขียนโปรแกรมหมากรุกไทยโดยใช้อัลกอริทึมทั้งสามที่ได้กล่าวในบทที่แล้วนั้น ประกอบด้วยการออกแบบโครงสร้างของโปรแกรมหมากรุกไทย การเขียนโปรแกรมของแต่ละอัลกอริทึมซึ่งต้องคำนึงถึงฟังก์ชันฮิวริสติกและการกำหนดพารามิเตอร์ต่างๆที่เหมาะสมสำหรับอัลกอริทึมนั้นๆ ดังต่อไปนี้

3.1 การออกแบบโครงสร้างของโปรแกรมหมากรุกไทย

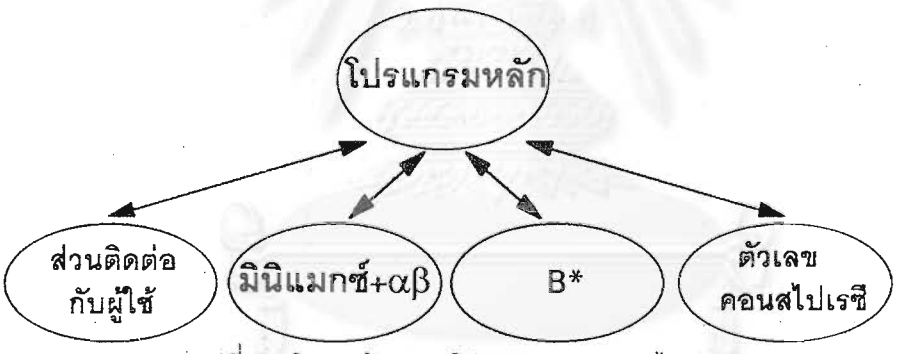
โครงสร้างของโปรแกรมหมากรุกไทยประกอบด้วยส่วนต่างๆดังต่อไปนี้คือ

-- โปรแกรมหลัก

-- ส่วนติดต่อกับผู้ใช้

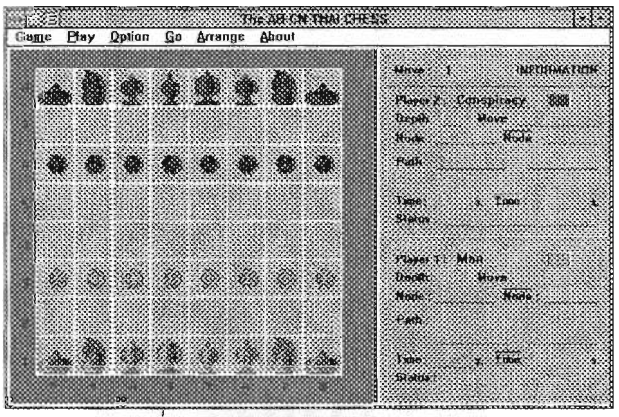
-- ส่วนที่เป็นอัลกอริทึมต่างๆเช่น อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา อัลกอริทึม B* อัลกอริทึมตัวเลขคอนสไปเรซี

ส่วนต่างๆติดต่อประสานงานกันดังแสดงในรูปที่ 20



รูปที่ 20 โครงสร้างของโปรแกรมหมากรุกไทย

โปรแกรมหลักจะทำหน้าที่ประสานการทำงานระหว่างส่วนอื่นๆ เช่น เรียกส่วนติดต่อกับผู้ใช้เพื่อรับข้อมูลที่ผู้ใช้ป้อนเข้า (ตัวอย่างเช่น ข้อมูลแสดงการเดินของตัวหมากที่ผู้ใช้กำหนด) แล้วส่งต่อไปยังอัลกอริทึมหนึ่งๆ หลังจากที่อัลกอริทึมค้นหาได้ตาเดินที่ดีที่สุดแล้วจะส่งข้อมูลกลับมายังโปรแกรมหลักเพื่อส่งต่อไปยังส่วนติดต่อกับผู้ใช้ต่อไป หน้าที่ของโปรแกรมหลักอีกอย่างคือ การตรวจสอบว่าการเดินตัวหมากจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งที่ผู้ใช้กำหนดนั้นถูกต้องตามกฎการเล่นหมากรุกหรือไม่ ถ้าไม่ถูกต้องก็จะเรียกส่วนติดต่อกับผู้ใช้ให้รับอินพุตจากผู้ใช้ใหม่



รูปที่ 21 ส่วนติดต่อกับผู้ใช้

ส่วนติดต่อกับผู้ใช้แสดงในรูปที่ 21 ส่วนติดต่อกับผู้ใช้นี้จะทำหน้าที่รับอินพุตจากผู้ใช้เพื่อส่งต่อไปยังโปรแกรมหลัก และรับเอ๊าท์พุตจากโปรแกรมหลักแล้วแสดงให้ผู้ใช้ ผู้ใช้สามารถส่งอินพุตต่างๆให้ส่วนติดต่อกับผู้ใช้ได้เช่น กำหนดว่าจะให้โปรแกรมเลือกอัลกอริทึมใดในการเล่นหมากรุก กำหนดพารามิเตอร์ต่างๆให้กับอัลกอริทึมหนึ่งๆ (ตัวอย่างเช่นกำหนดให้ความลึกของการค้นหาเป็น 3 สำหรับอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา) หรือรับเอ๊าท์พุตจากอัลกอริทึมหนึ่งๆแล้วแสดงผลว่าอัลกอริทึมเลือกเดินหมากอย่างไรให้กับผู้ใช้ได้รู้ ส่วนติดต่อกับผู้ใช้ได้ออกแบบให้ใช้งานได้สะดวกโดยใช้เมาส์ เช่นถ้าต้องการเปลี่ยนตำแหน่งของตัวหมากก็สามารถทำโดยใช้เมาส์เลือกตัวหมากที่ต้องการแล้วลากไปยังตำแหน่งใหม่ เป็นต้น

ส่วนที่เป็นอัลกอริทึมจะรับข้อมูลจากโปรแกรมหลักซึ่งบอกตำแหน่งต่างๆของตัวหมากแล้วนำมาใช้ในการค้นหาตาเดินที่เหมาะสมโดยอัลกอริทึมนั้นๆ ในการเล่นเกมหมากรุกหนึ่งๆ ได้ออกแบบให้ผู้ใช้สามารถกำหนดโดยผ่านส่วนติดต่อกับผู้ใช้ได้ว่าต้องการให้โปรแกรมเลือกอัลกอริทึมใดเล่นแข่งกับผู้ใช้ และสามารถกำหนดได้ว่าให้ 2 อัลกอริทึมใดๆเล่นแข่งกันเองก็ได้

3.2 การเขียนโปรแกรมอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย

การเขียนโปรแกรมของอัลกอริทึมต่างๆในหมากรุกไทยนั้น สิ่งสำคัญที่จะเป็นตัวกำหนดพฤติกรรมของโปรแกรมคือ ฟังก์ชันฮิวริสติกที่ใช้และพารามิเตอร์ต่างๆสำหรับอัลกอริทึมนั้นๆ ในหัวข้อนี้จะกล่าวถึงฟังก์ชันฮิวริสติก(ในกรณีของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาบางครั้งเราเรียกค่าฮิวริสติกว่าค่าสถิต) และพารามิเตอร์ที่กำหนดขึ้น

3.2.1 ฟังก์ชันฮิวริสติกของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย

ฟังก์ชันฮิวริสติกเป็นฟังก์ชันที่ใช้ประเมินค่าหรือให้คะแนนกับโหนดหนึ่งๆในต้นไม้เกมเพื่อหาว่าโหนดนั้นมีโอกาสชนะเท่าไร โหนดในต้นไม้เกมจะแสดงสภาพของกระดานสำหรับโหนดนั้นว่ามีตัวหมากเหลืออยู่เท่าใดและอยู่ในตำแหน่งใดบ้าง ในที่นี้ค่าฮิวริสติกของโหนดหนึ่งๆเป็นคะแนนรวมของกระดานนั้น ซึ่งคะแนนรวมของกระดานจะมาจากคะแนนย่อยๆซึ่งได้จากของผู้เล่นคนที่หนึ่งและคนที่สองรวมกัน ส่วนของคะแนนย่อยได้จาก

-- คะแนนของตัวหมากที่เหลืออยู่ในกระดาน

-- คะแนนของการคุมกำลังพลกันเองของตัวหมาก

-- คะแนนของการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม

คะแนนทั้งสามส่วนนี้ถ้าเป็นของฝ่ายเราจะมีค่าเป็นบวก(+) และถ้าเป็นของฝ่ายตรงข้ามจะมีค่าเป็นลบ(-)

<u>คะแนนของตัวหมากที่เหลืออยู่ในกระดาน</u>

คะแนนนี้คิดมาจากค่าที่จัดให้สำหรับตัวหมากแต่ละตัวดังต่อไปนี้

ขุน	มีค่าคะแนนของตัวหมากเป็น	50,000,000
เรือ	มีค่าคะแนนของตัวหมากเป็น	10,000,000
ม้า	มีค่าคะแนนของตัวหมากเป็น	5,000,000
โคน	มีค่าคะแนนของตัวหมากเป็น	1,000,000
เม็ด	มีค่าคะแนนของตัวหมากเป็น	500,000
เบี้ย	มีค่าคะแนนของตัวหมากเป็น	100,000

คะแนนเหล่านี้กำหนดขึ้นเพื่อให้อัลกอริทึมหลีกเลี่ยงการเอาตัวหมากที่มีค่ามากไปกินแลกกับตัวหมากของฝ่ายตรงข้าม เช่นให้หลีกเลี่ยงการนำเรือไปแลกกับม้า ซึ่งตามหลักการทั่วไปของการเล่นหมากรุกไทย จะถือกันว่าเรือมีความสำคัญมากกว่าม้าหรือพูดได้ว่าถ้ามีเรือก็จะมีโอกาสชนะมากกว่าการที่มีม้า สำหรับตัวเลขที่กำหนดขึ้นมานั้นได้จากการทดลองในหลายๆค่าแล้วดูพฤติกรรมของอัลกอริทึม ซึ่งตัวเลขที่กำหนดนี้ให้ผลการทดลองดีกว่าค่าอื่นๆ

<u>คะแนนของการคุมกำลังพลกันเองของตัวหมาก</u>

คะแนนนี้คิดจากค่าซึ่งตัวหมากพยายามคุมกำลังพลกันเองของผู้เล่นฝ่ายตน ดังต่อไปนี้

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขุน	คุมจะได้คะแนน	5
เรือ	คุมจะได้คะแนน	10
ม้า	คุมจะได้คะแนน	500
โคน	คุมจะได้คะแนน	1,000
เม็ด	คุมจะได้คะแนน	5,000
เบี้ย	คุมจะได้คะแนน	10,000

สาเหตุที่กำหนดให้คะแนนเป็นเช่นนี้เพื่อให้อัลกอริทึมพยายามใช้ตัวหมากหลายๆตัวในการคุมกันให้ได้คะแนนมากที่สุด ไม่ใช่ว่าใช้ตัวหมากที่มีคะแนนของตัวหมากสูงเพียงตัวเดียวคุมก็พอแล้ว เนื่องจากในการเล่นหมากรุกจริงๆนั้นจะเกิดการแลกตัวหมากกันดังนั้นจึงต้องพิจารณาว่าการแลกนั้นคุ้มหรือไม่ ซึ่งต้องใช้ตัวหมากอื่นๆมากกว่าหนึ่งตัวคุมตัวหมากที่อยู่ในจุดบนตารางที่ทำการแลก สำหรับตัวเลขที่กำหนดขึ้นนั้นได้มาจากการทดลองเช่นเดียวกับในกรณีของคะแนนของตัวหมากที่เหลืออยู่ในกระดาน

<u>คะแนนของการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม</u>

คะแนนนี้คิดจากค่าซึ่งตัวหมากต่างๆของผู้เล่นฝ่ายตนพยายามที่จะรุกเข้าไปหาขุนของผู้เล่นฝ่ายตรงข้าม ถ้าตัวหมากของตนมีตำแหน่งอยู่บนกระดานยิ่งใกล้ขุนฝ่ายตรงข้ามมากขึ้นเท่าใด ค่าการบุกรุกเข้าหาขุนของฝ่ายตรงข้ามก็จะยิ่งมากขึ้นตามไปด้วย

-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 1 ช่องทุกทิศทาง ได้คะแนน 12
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 2 ช่องทุกทิศทาง ได้คะแนน 10
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 3 ช่องทุกทิศทาง ได้คะแนน 8
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 4 ช่องทุกทิศทาง ได้คะแนน 7
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 5 ช่องทุกทิศทาง ได้คะแนน 6
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 6 ช่องทุกทิศทาง ได้คะแนน 4
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 7 ช่องทุกทิศทาง ได้คะแนน 2
-- อยู่ห่างขุนฝ่ายตรงข้ามไม่เกิน 0 ช่องทุกทิศทาง ได้คะแนนเพิ่มอีก 1

การกำหนดคะแนนเป็นเช่นนี้เพื่อให้อัลกอริทึมซึ่งเมื่อคุมกำลังพลกันดีแล้ว ก็ให้พยายามที่จะบุกรุกเข้าหาขุนฝ่ายตรงข้าม เพื่อพยายามกินขุนของฝ่ายตรงข้ามด้วย

3.2.2 พารามิเตอร์ที่กำหนดของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย

ความสามารถในการมองตาเดินล่วงหน้าของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตานั้นขึ้นอยู่กับความลึกที่กำหนดไว้ เนื่องจากอัลกอริทึมนี้จะค้นหาแบบลึกก่อนและเมื่อถึงความลึกที่กำหนดก็จะหยุดค้นหาในเส้นทางนั้นและทำการค้นหาต่อในเส้นทางอื่นแล้วนำมาเปรียบเทียบกันเพื่อหาโหนดที่มีค่าสูงสุด (โดยมีการลดทอนโดยอัลฟาเบตาช่วยในการตัดโหนดที่ไม่เป็นโหนดที่ดีออกไปโดยไม่มีผลกระทบกระเทือนต่อการค้นหา) กล่าวคือถ้าเรากำหนดให้ความลึกเป็น n แล้ว โหนดที่อยู่ในระดับ n+1 จะไม่ถูกกระจายหรือไม่สามารถเห็นได้โดยอัลกอริทึม ในเกมหมากรุกนั้นยิ่งเราสามารถค้นหาได้ลึกเท่าไรก็จะยิ่งเลือกตาเดินได้ดีมากขึ้นเท่านั้น ดังนั้นประสิทธิภาพของอัลกอริทึมจะขึ้นกับความลึกที่กำหนดนี้อย่างมาก อย่างไรก็ดีการกำหนดให้ความลึกมีค่ามากๆก็ไม่สามารถทำได้ในทางปฏิบัติ เนื่องจากมีข้อจำกัดในเวลาและหน่วยความจำที่เครื่องสามารถใช้ได้

ในงานวิจัยนี้เรากำหนดความลึกได้สูงสุดเป็น 3 ถ้ากำหนดมากกว่านี้โปรแกรมจะใช้เวลามากเกินไปกว่าเวลาสมควร (เกิน 1 นาทีต่อ 1 ก้าวเดินบนเครื่องไมโครคอมพิวเตอร์ที่มีไมโครโพรเซสเซอร์ 80486)

3.3 การเขียนโปรแกรมอัลกอริทึม B*ในหมากรุกไทย

3.3.1 ฟังก์ชันฮิวริสติกของอัลกอริทึม B* ในหมากรุกไทย

เนื่องจากอัลกอริทึม B* ต้องการค่าฮิวริสติก 2 ค่าสำหรับโหนดหนึ่งๆในต้นไม้เกม ซึ่งคือขอบเขตบน(optimistic) และขอบเขตล่าง(pessimistic) ดังนั้นเราไม่สามารถใช้ค่าฮิวริสติกของอัลกอริทึมมินิแมกซ์ในหัวข้อที่แล้วได้ เรานิยามฟังก์ชันฮิวริสติกสำหรับอัลกอริทึมนี้ดังต่อไปนี้

การคำนวณค่าขอบเขตบนทำโดยนำผลรวมของค่าของจำนวนตัวหมากทั้งหมดบวกกับค่าที่เราโจมตีฝ่ายตรงข้าม สำหรับค่าขอบเขตล่างได้จากผลรวมของค่าของจำนวนตัวหมากทั้งหมดบวกกับค่าที่เราถูกฝ่ายตรงข้ามโจมตี โดยที่ค่าของตัวหมากเป็นดังนี้

ขุน	มีค่า	6
เรือ	มีค่า	5
ม้า	มีค่า	4
โคน	มีค่า	3
เม็ดหรือเบี้ยหงาย	มีค่า	2
เบี้ย	มีค่า	1

ค่าของตัวหมากแต่ละตัวจะเป็นบวกถ้าเป็นหมากของฝ่ายเรา และเป็นลบถ้าเป็นของฝ่ายตรงข้าม และถ้าฝ่ายเรามีการหงายเบี้ยจะเพิ่มค่าอีก 1 แต่ถ้าเป็นการหงายเบี้ยของฝ่ายตรงข้ามจะลดลง 1

ค่าที่เราโจมตีฝ่ายตรงข้ามจะคิดจากค่าของตัวหมากของฝ่ายตรงข้ามทั้งหมดที่ถูกฝ่ายเราจับกิน และค่าที่เราถูกโจมตีคิดจากค่าของตัวหมากของฝ่ายเราทั้งหมดที่ถูกฝ่ายตรงข้ามจับกิน

3.3.2 พารามิเตอร์ที่กำหนดของอัลกอริทึม B* ในหมากรุกไทย

เงื่อนไขในการหยุดของอัลกอริทึม B* นั้นคือ ค่าขอบเขตล่างของโหนดที่ดีที่สุดมีค่ามากกว่าค่าขอบเขตบนของโหนดอื่นที่ระดับความลึก 1 หลังจากที่ได้เขียนโปรแกรม B* ขึ้นสำหรับงานวิจัยนี้พบว่าเงื่อนไขในการหยุดนี้เป็นเงื่อนไขที่แข็งเกินไป กล่าวคือเริ่มต้นจากโหนดรากหนึ่งเราอาจต้องใช้เวลานานเกินกว่าเวลาที่สมควร (เกิน 1 นาทีต่อ 1 ก้าวเดิน) และใช้หน่วยความจำมากเกินไป จึงได้กำหนดเงื่อนไขที่อ่อนลงคือ

1. อัลกอริทึมจะหยุดค้นหาเมื่อค่าขอบเขตล่างของโหนดที่ดีที่สุดมีค่ามากกว่าค่าขอบเขตบนของโหนดอื่นที่ระดับความลึก 1 หรือ

2. เมื่อได้กระจายโหนดทั้งหมด 1000 โหนด (ค่านี้สามารถกำหนดได้)

ในกรณีที่ 2 นั้น จะถือว่าเป็นการค้นหาที่ไม่สมบูรณ์ของอัลกอริทึมนี้ ซึ่งอาจทำให้เกิดข้อผิดพลาดได้ เพราะว่าโหนดที่มีค่าขอบเขตบนสูงสุดอาจไม่ใช่โหนดที่ดีที่สุดก็ได้ โหนดนี้อาจมีค่าขอบเขตล่างน้อยกว่าโหนดอื่นๆ ในกรณีนี้จะถือว่า โหนดที่มีค่าขอบเขตล่างสูงสุดเป็นโหนดที่จะถูกเลือก

3.4 การเขียนโปรแกรมอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย

3.4.1 ฟังก์ชันฮิวริสติกของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย

ฟังก์ชันฮิวริสติกที่ใช้ในอัลกอริทึมตัวเลขคอนสไปเรซีประกอบด้วย 3 ส่วนคือ

-- คะแนนของตัวหมากที่เหลืออยู่ในกระดาน

-- คะแนนของการคุมกำลังพลกันเองของตัวหมาก

-- คะแนนของการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม

คะแนนทั้งสามส่วนนี้ถ้าเป็นของฝ่ายเราจะมีค่าเป็นบวก(+) และถ้าเป็นของฝ่ายตรงข้ามจะมีค่าเป็นลบ(-)

คะแนนของตัวหมากที่เหลืออยู่บนกระดาน

คะแนนส่วนนี้เป็นผลรวมของคะแนนตัวหมากของทั้ง 2 ฝ่ายที่เหลืออยู่บนกระดานดังต่อไปนี้

ขุน	มีค่าคะแนนของตัวหมากเป็น	50
เรือ	มีค่าคะแนนของตัวหมากเป็น	35
ม้า	มีค่าคะแนนของตัวหมากเป็น	25
โคน	มีค่าคะแนนของตัวหมากเป็น	15
เม็ด	มีค่าคะแนนของตัวหมากเป็น	10
เบี้ย	มีค่าคะแนนของตัวหมากเป็น	5

<u>คะแนนของการคุมกำลังพลกันเองของตัวหมาก</u>

ในที่นี้กำหนดให้จำนวน 10 ตาเดินแรกหมายถึง สภาวะเริ่มต้นของเกม จากการสังเกตพบว่าในช่วงสภาวะเริ่มต้นของเกม การป้องกันทางด้านข้างและด้านหลังจะมีความสำคัญน้อยกว่าการป้องกันทางด้านหน้าเนื่องจากคู่ต่อสู้เริ่มบุกมาทางด้านหน้าของฝ่ายเรา แต่หลังจากที่ผ่านสภาวะเริ่มต้นไปแล้ว การป้องกันจากทุก ๆ ด้านจะมีความสำคัญเท่ากันหมด เพราะในตอนนี้หมากของคู่ต่อสู้มิได้อยู่ตรงด้านหน้าเพียงอย่างเดียว แต่อาจจะอยู่ทางด้านข้างและทางด้านหลังของหมากฝ่ายตนเองได้ ดังนั้นจึงได้ออกแบบคะแนนในการคุมกำลังพลหรือการป้องกันหมากของฝ่ายตนเองดังตารางที่ 2 ด้านล่างนี้

<u>ตารางที่ 2 คะแนนในการป้องกันหมากของฝ่ายเดียวกัน</u>

สภาวะของเกม	คะแนนจากการป้องกัน		
	ด้านหน้า	ด้านข้าง	ด้านหลัง
เริ่มต้น	2	1	1
หลังจากนั้น	2	2	2

<u>คะแนนของการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม</u>

คะแนนในส่วนนี้แบ่งย่อยออกเป็น 3 ส่วนดังนี้

1 คะแนนตำแหน่งของหมากที่อยู่ห่างจากขุนฝ่ายตรงข้าม

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 1 ช่อง ได้ 2 คะแนน

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 2 ช่อง ได้ 2 คะแนน

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 3 ช่อง ได้ 2 คะแนน

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 4 ช่อง ได้ 2 คะแนน

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 5 ช่อง ได้ 1 คะแนน

หมากอยู่ห่างจากขุนฝ่ายตรงข้าม 6 ช่อง ได้ 1 คะแนน

2 คะแนนที่เพิ่มให้ตัวหมากที่อยู่ในตำแหน่งที่สามารถกินขุนฝ่ายตรงข้ามได้ คะแนนนี้ขึ้นอยู่กับว่าเป็นหมากตัวใด โดยกำหนดให้

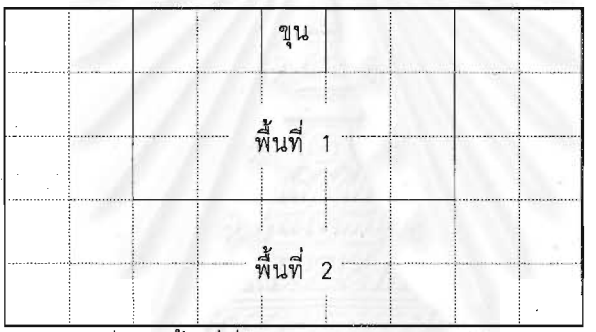
ม้า ได้คะแนน 5 คะแนน

เรือ ได้คะแนน 7 คะแนน

อื่น ๆ ได้คะแนน 3 คะแนน

3 คะแนนที่เพิ่มให้เมื่อมีหมากหลาย ๆ ตัวอยู่ในตำแหน่งที่เข้าใกล้ขุนฝ่ายตรงข้าม

เพื่อให้มีหมากหลาย ๆ ตัวเข้าช่วยกันบุกรุกเข้าหาขุนของฝ่ายตรงข้าม โดยแบ่งพื้นที่ที่ล้อมรอบขุนคู่ต่อสู้ออกเป็น 2 ส่วน ตามรูปที่ 22 และกำหนดให้คะแนนในแต่ละส่วนเป็นไปตามตารางที่ 3 ดังนี้



รูปที่ 22 พื้นที่ที่ล้อมรอบขุนฝ่ายตรงข้าม

ตารางที่ 3 แสดงคะแนนของแต่ละพื้นที่ที่ล้อมรอบขุนฝ่ายตรงข้าม

จำนวนหมากในพื้นที่	คะแนนในพื้นที่ 1	คะแนนในพื้นที่ 2
0	0	0
1	1	2
2	2	4
3	4	8
4	8	16
>4	8	16

3.4.2 พารามิเตอร์ที่กำหนดของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย

ในการพัฒนาโปรแกรมหมากรุกไทยโดยอัลกอริทึมตัวเลขคอนสไปเรซีนี้มีพารามิเตอร์ที่ต้องกำหนดให้ดังนี้

- ค่าขีดแบ่ง (CT)
- ช่วงกว้าง (Vmax-Vmin)
- จำนวนกิ่งแยกของโหนด (BF)

ค่าขีดแบ่งคือตัวเลขคอนสไปเรซีที่น้อยที่สุดที่กำหนดให้ในการพิจารณาค่าที่ไม่เหมาะสม นั่นคือค่าที่มากกว่าค่าขีดแบ่งจะไม่นำมาพิจารณาหาค่ามากที่สุด(Vmax)และค่าน้อยที่สุด(Vmin)ที่เป็นไปได้ ในงานวิจัยนี้เรากำหนดให้ค่าโดยปริยายของค่าขีดแบ่งเป็น 2

ช่วงกว้างเป็นพารามิเตอร์ที่ใช้วัดความถูกต้องของค่ามินิแมกซ์ของโหนดราก โปรแกรมจะหยุดการกระจายโหนดเมื่อ Vmax - Vmin น้อยกว่าหรือเท่ากับช่วงกว้างนี้ เรากำหนดให้ค่านี้มีค่าเท่ากับ 0

จำนวนกิ่งแยกของโหนดเป็นจำนวนทางเลือกของตาเดิน โดยเฉลี่ยแล้วจำนวนกิ่งแยกของโหนดสำหรับหมากรุกไทยเป็น 37 แต่เนื่องจากอัลกอริทึมตัวเลขคอนสไปเรซีต้องเก็บช่วงค่าที่เป็นไปได้ทั้งหมดไว้ที่แต่ละโหนด เพื่อคำนวณตัวเลขคอนสไปเรซี จึงต้องเก็บต้นไม้เกมทั้งต้นในหน่วยความจำทำให้สิ้นเปลืองและอาจเกิดปัญหาหน่วยความจำไม่พอ และในกรณีที่ช่วงค่าที่เป็นไปได้ทั้งหมดมีมากเกินไปจะเกิดปัญหาของหน่วยความจำและเวลามาก ดังนั้นเราจึงไม่สามารถนำโหนดทั้ง 37 โหนดมาใช้ในการค้นหาได้หมด ในงานวิจัยนี้จึงกำหนดให้มีการเลือกจำนวนกิ่งแยกที่มีค่ามินิแมกซ์ดีที่สุดจาก 37 โหนดเป็นโหนดที่ใช้ในการค้นหาต่อไป (ค่าโดยปริยายเท่ากับ 15)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. การทดสอบการใช้งานและการวิเคราะห์ผล

บทนี้กล่าวถึงการทดสอบโปรแกรมของอัลกอริทึมทั้งสามและวิเคราะห์ผลที่ได้ หลังจากนั้นจะกล่าวถึงการทดลองและผลการเปรียบเทียบระหว่างอัลกอริทึมมินิแมกซ์+อัลฟาเบตากับอัลกอริทึมตัวเลขคอนสไปเรซี

4.1 การทดสอบและวิเคราะห์โปรแกรมอัลกอริทึมทั้ง 3 ในหมากรุกไทย

4.1.1 การวิเคราะห์ผลที่ได้ของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทย

จากการทดลองเขียนโปรแกรมของอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาในหมากรุกไทยนั้น พบว่าจำนวนกิ่งแยกของโหนดหนึ่งๆมีค่าประมาณ 37 (โหนดหนึ่งๆมีโหนดลูกทั้งหมด 37 โหนดโดยประมาณ) ซึ่งตัวเลขนี้เป็นตัวเลขของหมากรุกไทยโดยไม่ขึ้นอยู่กับอัลกอริทึมใดๆ ดังนั้นถ้าเราค้นหาล่วงหน้า 2 ก้าวเดิน โหนดที่ถูกสร้างขึ้นทั้งหมดจะประมาณ 37*37=1,369 หรือ ถ้าค้นหาล่วงหน้า 3 หรือ 4 ก้าวเดิน โหนดทั้งหมดจะประมาณ 50,653 หรือ 1,874,161 ตามลำดับ อัลกอริทึมมินิแมกซ์สามารถสร้างโหนดได้ประมาณ 700 โหนดต่อ 1 วินาที (เป็นค่าที่วัดได้จากการทดลองบนเครื่องไมโครคอมพิวเตอร์ซีพียู 80486 หน่วยความจำ 4 เมกะไบต์ ความเร็ว 66 เมกะเฮิรตซ์) ดังนั้นถ้าค้นหาล่วงหน้า 2 ก้าวเดิน ต้องใช้เวลาประมาณ 2 วินาที และถ้าค้นหาล่วงหน้า 3 ก้าวเดิน ต้องใช้เวลาประมาณ 72 วินาที ซึ่งใช้เวลาเกินกว่าเวลาที่สมควร(เกิน 1 นาที) ซึ่งก็หมายความว่าถ้าเราใช้อัลกอริทึมมินิแมกซ์โดยไม่ใช้การลดทอนโดยอัลฟาเบตา เราสามารถค้นหาล่วงหน้าได้เพียง 2 ก้าวเดิน

แต่ถ้าเราใช้การลดทอนโดยอัลฟาเบตาแล้วจากการทดลองพบว่าจำนวนกิ่งแยกของโหนดหนึ่งๆโดยเฉลี่ยจะลดลงเหลือ 19 ซึ่งหมายถึงถ้าค้นหาล่วงหน้า 2 ก้าวเดินต้องสร้างโหนดทั้งหมดประมาณ 361 โหนดและใช้เวลาประมาณ 0.5 วินาที จากการใช้การลดทอนโดยอัลฟาเบตาทำให้เราสามารถค้นหาล่วงหน้าได้ 3 ก้าวเดินซึ่งใช้เวลาประมาณ 10 วินาที (สร้างโหนดได้ทั้งหมดประมาณ 6900 โหนด)

ผลการทดลองแสดงให้รู้ว่าอัลกอริทึมนี้สามารถนำมาประยุกต์ใช้กับหมากรุกไทยได้แต่ถ้าต้องการให้โปรแกรมค้นหาล่วงหน้าได้มากกว่านี้ยังจำเป็นต้องใช้เทคนิคย่อยอื่นๆเข้าช่วย

4.1.2 การวิเคราะห์ผลที่ได้ของอัลกอริทึม B*

จากการเขียนโปรแกรมโดยใช้ทฤษฎีของอัลกอริทึม B* พบว่าอัลกอริทึม B* ไม่เหมาะสมกับหมากรุกไทยด้วยเหตุผลดังต่อไปนี้ (ธนะเทพ, 2538)

1.อัลกอริทึม B* ต้องการค่าฮิวริสติก 2 ค่าสำหรับประเมินโหนดหนึ่งๆ ซึ่งได้แก่ ขอบเขตล่างและขอบเขตบน การนิยามค่าฮิวริสติกนี้จะมีผลต่อการลู่เข้าเพื่อทำให้อัลกอริทึมหยุดแล้วส่งคำตอบที่เป็นโหนดที่ดีที่สุด ผลจากการทดลองพบว่าการนิยามค่าฮิวริสติกของอัลกอริทึมนี้สำหรับหมากรุกไทยทำได้ยากและบ่อยครั้งมากที่อัลกอริทึมไม่ลู่เข้าหรือไม่หยุด

2.การหยุดของอัลกอริทึมใช้เงื่อนไขที่แข็งเกินไป ซึ่งปัญหานี้มีส่วนสัมพันธ์กับการนิยามฟังก์ชันฮิวริสติก และถ้าเราพยายามลดให้เงื่อนไขอ่อนลงซึ่งอาจทำโดยกำหนดจำนวนโหนดที่กระจายในแต่ละครั้งดังที่ได้ทดลองในงานวิจัยนี้ ก็จะทำให้คำตอบที่ได้จากอัลกอริทึมไม่เป็นคำตอบที่ดีและส่งผลให้อัลกอริทึมเลือกตาเดินที่ไม่ดี

4.1.3 การวิเคราะห์ผลของอัลกอริทึมตัวเลขคอนสไปเรซีในหมากรุกไทย

อัลกอริทึมคอนสไปเรซีซึ่งเสนอโดย D.A.McAllester (1988) และ J.Schaeffer (1990) นั้นใช้ฟังก์ชันฮิวริสติกที่พิจารณาเฉพาะคะแนนของตัวหมากที่เหลืออยู่ในกระดานเพียงอย่างเดียว แต่หลังจากที่เราได้เขียนโปรแกรมทดสอบดูพบว่า ฟังก์ชันฮิวริสติกนี้ไม่เหมาะสมกับหมากรุกไทย เนื่องจากไม่สามารถแยกตาเดินที่เหมาะสมในกรณีที่ไม่มีการกินตัวหมากกันได้ จึงได้เพิ่มคะแนนส่วนที่เป็นการคุมกำลังพลกันเองของตัวหมากและส่วนที่เป็นการบุกรุกเข้าหาขุนของฝ่ายตรงข้าม ซึ่งผลที่ได้คือทำให้อัลกอริทึมสามารถเลือกตาเดินได้สมเหตุสมผลมากขึ้น

ผลที่ได้จากการเขียนโปรแกรมอัลกอริทึมทั้งสามทำให้รู้ว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาและอัลกอริทึมตัวเลขคอนสไปเรซีเหมาะสมกับโปรแกรมหมากรุกไทยมากกว่าอัลกอริทึม B* ดังนั้นในหัวข้อต่อไปเราจะศึกษาเปรียบเทียบอัลกอริทึมทั้งสองนี้

4.2 การทดลองเพื่อเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาและอัลกอริทึมตัวเลขคอนสไปเรซีและการวิเคราะห์ผล

4.2.1 ผลการทดลองเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาและอัลกอริทึมตัวเลขคอนสไปเรซี

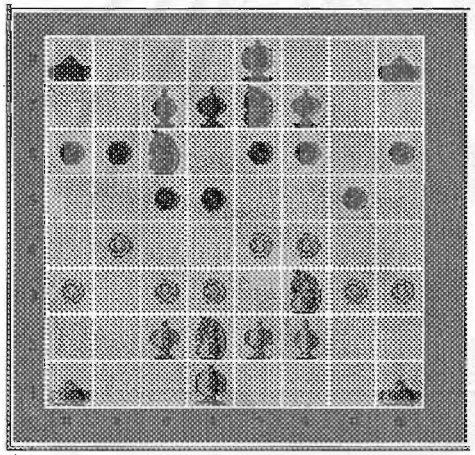
เราได้ออกแบบการทดลองออกเป็น 4 ชุด แต่ละชุดมี 10 เกม ซึ่งมีทั้งการเปิดและไม่เปิดสมุดมาตรฐาน(สมุดมาตรฐานแสดงรูปแบบการเดินตัวหมากในช่วงเริ่มต้นของเกม) การทดลองได้ออกแบบขึ้นมาเพื่อเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา(ในตารางที่ 4-7 แทนด้วย $\alpha\beta$)และ อัลกอริทึมตัวเลขคอนสไปเรซี (ในตารางที่ 4-7 แทนด้วย CN) โดยควบคุมจำนวนโหนดที่สร้างได้ให้ใกล้เคียงกัน หรือควบคุมเวลาในการสร้างตาเดินให้ได้ใกล้เคียงกัน

สมุดมาตรฐานที่ใช้ในการทดลองมีรูปแบบตาเดิน 2 รูปแบบดังต่อไปนี้

ตาเดินมาตรฐานรูปแบบที่ 1

ฝ่ายขาว	ฝ่ายดำ
1. ม, ข1-ง1	ม, ข8-จ7
2. ค, ค1-ค2	ค, ฉ8-ฉ7
3. บ, ข3-ข4	บ, ข6-ข5
4. บ, ฉ3-ฉ4	บ, ค6-ค5
5. ม, ข1-ฉ3	ม, ข8-ค6
6. บ, จ3-จ4	บ, ง6-ง5
7. ด, จ1-ฉ2	ด, ง8-ค7
8. ค, ฉ1-จ2	ค, ค8-ง7

กระดานหลังจากเปิดสมุดรูปแบบที่ 1



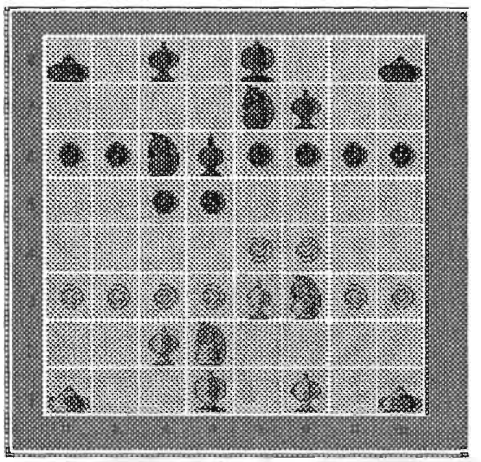
รูปที่ 23 รูปแบบที่ 1 ของสมุดมาตรฐาน

ตาเดินมาตรฐานรูปแบบที่ 2

ฝ่ายขาว	ฝ่ายดำ
1. ค, ค1-ค2	ค, ฉ8-ฉ7
2. บ, ฉ3-ฉ4	บ, ค6-ค7
3. ม, ข1-ฉ3	ม, ข8-ค6
4. ม, ข1-ง2	ม, ข8-จ7
5. บ, จ1-ฉ2	บ, ง6-ง5
6. บ, จ3-จ4	ด, ง8-ง5
7. ด, จ1-ฉ2	ด, ค7-ง6
8. ด, ฉ2-จ3	บ, ง5-ง4

จุฬาลงกรณ์มหาวิทยาลัย

กระดานหลังจากเปิดสมุดรูปแบบที่ 2



รูปที่ 24 รูปแบบที่ 2 ของสมุดมาตรฐาน

การทดลองทั้งหมดแบ่งออกเป็น 4 ชุดซึ่งแต่ละชุดกำหนดพารามิเตอร์ของอัลกอริทึมทั้งสองดังต่อไปนี้

การทดลองชุดที่ 1 กำหนดความลึกของ $\alpha\beta$(D) = 2, ค่าขีดแบ่งของ CN (CT) = 2, จำนวนกิ่งแยกของ CN (BF) = 15

การทดลองชุดที่ 2 กำหนดความลึกของ $\alpha\beta$(D) = 3, ค่าขีดแบ่งของ CN (CT) = 4, จำนวนกิ่งแยกของ CN (BF) = 30

การทดลองชุดที่ 3 กำหนดความลึกของ $\alpha\beta$(D) = 3, ค่าขีดแบ่งของ CN (CT) = 2, จำนวนกิ่งแยกของ CN (BF) = 15

การทดลองชุดที่ 4 กำหนดความลึกของ $\alpha\beta$(D) = 4, ค่าขีดแบ่งของ CN (CT) = 2, จำนวนกิ่งแยกของ CN (BF) = 30

การทดลองที่ 1 และ 2 กำหนดให้โหนดที่กระจายได้มีจำนวนใกล้เคียงกัน (หลักร้อยและหลักพันตามลำดับ) แล้วทดสอบว่าอัลกอริทึมใดเลือกตาเดินดีกว่าโดยให้อัลกอริทึมทั้งสองเล่นแข่งกัน การทดลองที่ 3 และ 4 กำหนดให้เวลาที่ใช้ในการเดินแต่ละครั้งใกล้เคียงกัน (ไม่เกิน 15 วินาทีและ 1 นาทีตามลำดับ) แล้วเปรียบเทียบอัลกอริทึมทั้งสอง เราควบคุมจำนวนโหนดหรือเวลาที่ใช้ให้ใกล้เคียงกันโดยปรับพารามิเตอร์ต่างๆของอัลกอริทึม พารามิเตอร์สำหรับอัลกอริทึมตัวเลขคอนสไปเรซีในที่ใช้ปรับคือค่าขีดแบ่ง(CT ของ CN ในตารางที่ 4-7) และจำนวนกิ่งแยกของโหนด(BF ของ CN ในตารางที่ 4-7) ส่วนพารามิเตอร์ของอัลกอริทึมมินิแมกซ์+การลดทอนโดย-อัลฟาเบตา คือความลึกสูงสุด(D ของ $\alpha\beta$ ในตารางที่ 4-7)

ผลการทดลองแสดงในตารางที่ 4-7 ค่า MV ในตารางแสดงคะแนนของตัวหมากที่เหลืออยู่บนกระดานทั้งหมดรวมกัน (ตามฟังก์ชันฮิวริสติกในหัวข้อ 3.4.1) ของแต่ละฝ่ายเมื่อสิ้นสุดเกม การสิ้นสุดของเกมกำหนดโดย (1) เมื่อเกิดการรุกฆาต (2) เมื่อทั้งสองฝ่ายเดินครบฝ่ายละ 130 ก้าวเดิน หรือ (3) เมื่อเกิดการเดินวนซ้ำ รูปแบบ P1/P2 ในตารางหมายถึงรูปแบบของสมุดมาตรฐานในการเดินตัวหมากในช่วงเปิดเกม (0/0 หมายถึงทั้งสองฝ่ายไม่ใช้สมุดมาตรฐาน 1/2 หมายถึงฝ่ายแรกใช้รูปแบบที่ 1 ฝ่ายหลังใช้รูปแบบที่ 2 เป็นต้น) "ฝ่ายเริ่ม" ในตารางแสดงอัลกอริทึมที่เดินก่อน

เพื่อควบคุมการทดลองเรากำหนดให้ฟังก์ชันฮิวริสติกที่ใช้ของทั้งสองอัลกอริทึมเป็นฟังก์ชันเดียวกันคือฟังก์ชันตามหัวข้อที่ 3.4.1 (เราได้ทดลองเปลี่ยนฟังก์ชันเป็นฟังก์ชันที่ใช้ในหัวข้อ 3.2.1 ซึ่งผลที่ได้ใกล้เคียงกัน) "ผู้ชนะ" ในตารางแสดงถึงอัลกอริทึมที่มีค่า MV มากกว่าเมื่อสิ้นสุดเกม (ค่า MV คือค่าตัวหมากที่เหลือในกระดาน) ตัวเลข X:Y ในตารางแสดงถึงจำนวน X ครั้งที่อัลกอริทึม CN เป็นผู้ชนะและ Y ครั้งที่อัลกอริทึม $\alpha\beta$ เป็นผู้ชนะ ตัวอย่างเช่นในการทดลองที่ 1 เรากำหนดให้ D = 2, CT = 2, BF = 15 จำนวนโหนดเฉลี่ยที่อัลกอริทึม CN สร้างคือ 251 โหนดต่อการเดินหมากแต่ละครั้ง ส่วนของอัลกอริทึม $\alpha\beta$ เป็น 313 เวลาที่ใช้โดยเฉลี่ยของอัลกอริทึม CN และ $\alpha\beta$ ต่อการเดินหมากแต่ละครั้งเป็น 10.6 และ 1.0 ตามลำดับ ผลที่ได้คือ 6:1 หมายถึงอัลกอริทึม CN ชนะ 6 ครั้ง อัลกอริทึม $\alpha\beta$ ชนะ 1 ครั้ง ผลการทดลองทั้ง 4 ชุดแสดงในตารางที่ 4-7 ด้านล่างนี้

ตารางที่ 4 ผลการทดลองชุดที่ 1

D ของ $\alpha\beta$	CT ของ CN	BF ของ CN	รูปแบบ P1/P2	ฝ่ายเริ่ม	CN			$\alpha\beta$			ผู้ชนะ
					จำนวนโหนด	เวลา (วินาที)	MV	จำนวนโหนด	เวลา (วินาที)	MV	
2	2	15	0/0	CN	246	7.9	250	97	1.0	250	D
			0/0	$\alpha\beta$	247	7.6	250	106	1.0	250	D
			1/1	CN	245	10.3	140	301	1.0	85	CN
			1/1	$\alpha\beta$	194	8.6	180	362	1.0	50	CN
			2/2	CN	259	12.8	230	268	1.0	150	CN
			2/2	$\alpha\beta$	257	12.0	240	444	1.0	230	CN
			1/2	CN	268	13.8	255	275	1.0	225	CN
			1/2	$\alpha\beta$	260	9.5	135	412	1.0	150	$\alpha\beta$
			2/1	CN	260	11.0	170	387	1.0	165	CN
			2/1	$\alpha\beta$	275	12.7	195	477	1.0	195	D
ค่าเฉลี่ย					251	10.6	205	313	1.0	175	6:1

ตารางที่ 5 ผลการทดลองชุดที่ 2

D ของ αβ	CT ของ CN	BF ของ CN	รูปแบบ P1/P2	ฝ่ายเริ่ม	CN			αβ			ผู้ชนะ
					จำนวนโหนด	เวลา (วินาที)	MV	จำนวนโหนด	เวลา (วินาที)	MV	
3	4	30	0/0	CN	1637	67.7	225	4201	3.3	165	CN
			0/0	αβ	1505	78.1	195	3251	2.3	210	αβ
			1/1	CN	1020	59.3	80	2804	2.1	55	CN
			1/1	αβ	1545	99.2	180	8667	5.6	160	CN
			2/2	CN	1421	85.8	190	7932	5.7	175	CN
			2/2	αβ	1676	114.1	225	6151	4.4	200	CN
			1/2	CN	1358	75.6	160	3650	2.3	155	CN
			1/2	αβ	1194	53.7	165	4489	1.7	65	CN
			2/1	CN	1422	77.4	100	6160	4.2	140	αβ
			2/1	αβ	1402	76.7	115	5440	3.7	120	αβ
ค่าเฉลี่ย					1418	78.8	164	5275	3.5	145	7:3

ตารางที่ 6 ผลการทดลองชุดที่ 3

D ของ αβ	CT ของ CN	BF ของ CN	รูปแบบ P1/P2	ฝ่ายเริ่ม	CN			αβ			ผู้ชนะ
					จำนวนโหนด	เวลา (วินาที)	MV	จำนวนโหนด	เวลา (วินาที)	MV	
3	2	15	0/0	CN	257	8.4	240	3786	3.1	245	αβ
			0/0	αβ	232	8.3	105	3404	2.4	55	CN
			1/1	CN	263	13.6	225	5449	3.9	190	CN
			1/1	αβ	259	10.6	190	6228	4.5	220	αβ
			2/2	CN	237	11.6	175	7683	5.6	180	αβ
			2/2	αβ	248	11.1	155	4705	3.6	80	CN
			1/2	CN	261	12.1	240	4464	1.6	240	D
			1/2	αβ	242	7.9	90	3842	2.6	140	αβ
			2/1	CN	222	6.8	115	4952	3.7	170	αβ
			2/1	αβ	235	9.1	110	5769	4.0	105	CN
ค่าเฉลี่ย					246	9.9	164	5028	3.5	163	4:5

ตารางที่ 7 ผลการทดลองชุดที่ 4

D ของ αβ	CT ของ CN	BF ของ CN	รูปแบบ P1/P2	ฝ่ายเริ่ม	CN			αβ			ผู้ชนะ
					จำนวนโหนด	เวลา (วินาที)	MV	จำนวนโหนด	เวลา (วินาที)	MV	
4	2	30	0/0	CN	568	10.0	200	40720	13.3	225	αβ
			0/0	αβ	480	8.2	180	16028	5.5	235	αβ
			1/1	CN	582	14.7	130	26804	8.1	125	CN
			1/1	αβ	805	15.0	120	41862	13.4	165	αβ
			2/2	CN	237	5.7	165	7683	2.8	200	αβ
			2/2	αβ	791	18.0	170	51162	17.2	200	αβ
			1/2	CN	650	13.2	145	21976	6.6	135	CN
			1/2	αβ	743	14.1	145	30546	9.5	150	αβ
			2/1	CN	630	13.0	180	21085	7.2	170	CN
			2/1	αβ	944	22.0	205	13705	46.1	210	αβ
ค่าเฉลี่ย					643	13.4	164	27157	12.9	181	3:7

4.2.2 วิเคราะห์ผลการทดลองเปรียบเทียบอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาและอัลกอริทึมตัวเลขคอนสไปเรซี

ผลการทดลองที่ 1 ในตารางที่ 4 แสดงให้เห็นว่าเมื่อกำหนดให้จำนวนโหนดที่กระจายได้ใกล้เคียงกันคืออยู่ในหลักร้อยวินาทีเหมือนกัน อัลกอริทึมตัวเลขคอนสไปเรซีเลือกตาเดินได้ดีกว่าและเล่นชนะอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาเกือบทุกเกม ยกเว้นเกมที่ 8 จากการทดลองพบว่าอัลกอริทึมตัวเลขคอนสไปเรซีค้นหาล่วงหน้าได้ 3-4 ก้าวเดินในแต่ละครั้งซึ่งมากกว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาซึ่งค้นหาล่วงหน้าได้เพียง 2 ก้าวเดินในจำนวนโหนดที่ใกล้เคียงกัน (จำนวนโหนดเฉลี่ยของทุกเกมของอัลกอริทึมตัวเลขคอนสไปเรซีเป็น 251 ส่วน อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาเป็น 313) ซึ่งแสดงให้เห็นถึงคุณภาพการกระจายโหนดที่ดีกว่าของอัลกอริทึมตัวเลขคอนสไปเรซีโดยจะเลือกกระจายโหนดในแนวลึกในเส้นทางที่สนใจเท่านั้น

ผลการทดลองที่ 2 ยืนยันผลการทดลองที่ 1 คือเมื่อให้ทั้งสองอัลกอริทึมสร้างโหนดได้ใกล้เคียงกันอยู่ในหลักพันวินาที ผลที่ได้คืออัลกอริทึมตัวเลขคอนสไปเรซีชนะอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามากกว่า (ชนะ 7 แพ้ 3)

ผลการทดลองที่ 3 และ 4 แสดงให้เห็นว่าเมื่อกำหนดเวลาให้อัลกอริทึมต้องเดินภายใน 15 วินาทีและ 1 นาทีตามลำดับ อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาสามารถสร้าง

โหนดได้มากกว่าและเป็นฝ่ายชนะมากกว่า อย่างเช่นในตารางที่ 7 ซึ่งแสดงการทดลองที่ 4 จะเห็นได้ว่าอัลกอริทึมใช้เวลาเฉลี่ยในการเดินหมากแต่ละครั้งเป็น 12.9 วินาทีและสร้างโหนดได้เฉลี่ย 27157 โหนด ซึ่งมากกว่าจำนวนโหนดเฉลี่ยของอัลกอริทึมตัวเลขคอนสไปเรซีที่สร้างได้ 643 โหนด ใช้เวลาเฉลี่ย 13.4 วินาที (ตัวเลขแสดงเวลาเฉลี่ยในการทดลองนี้และทุกการทดลองวัดบนเครื่องไมโครคอมพิวเตอร์เพนเทียม ความเร็ว 100 เมกะเฮิรตซ์)

ผลการทดลองแสดงให้เห็นถึงข้อดีของแต่ละอัลกอริทึมคืออัลกอริทึมตัวเลขคอนสไปเรซีสร้างโหนดที่มีคุณภาพกว่าโดยเลือกสร้างเฉพาะเส้นทางที่น่าสนใจ ซึ่งการเลือกสร้างโหนดของอัลกอริทึมนี้มีลักษณะที่คล้ายกับคน ส่วนอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามีข้อดีที่สามารถสร้างโหนดได้จำนวนมากในเวลาที่กำหนด แม้ว่าความเร็วของอัลกอริทึมทั้งสองจะขึ้นกับวิธีการเขียนโปรแกรม แต่โดยธรรมชาติของอัลกอริทึมตัวเลขคอนสไปเรซีแล้วจะช้ากว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามาก เนื่องจากต้องแก้ไขค่า CN ของแต่ละโหนดในต้นไม้เกม ผลโดยรวมแล้วจะเห็นว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาได้เปรียบอัลกอริทึมตัวเลขคอนสไปเรซี อย่างไรก็ดีเราเห็นว่า อัลกอริทึมตัวเลขคอนสไปเรซีก็เป็นทางเลือกหนึ่งที่น่าสนใจสำหรับโปรแกรมหมากรุกไทยแต่ยังคงต้องการการศึกษาเพิ่มเติมในการเพิ่มประสิทธิภาพทางด้านความเร็วในการค้นหา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. ข้อสรุป

งานวิจัยนี้ได้กล่าวถึงการประยุกต์ใช้อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา อัลกอริทึม B* และอัลกอริทึมตัวเลขคอนสไปเรซีในโปรแกรมหมากรุกไทย และหาข้อสรุปว่าอัลกอริทึมใดมีความเหมาะสมที่จะนำมาใช้ในการพัฒนาโปรแกรมหมากรุกไทย โดยที่โปรแกรมหมากรุกไทยที่พัฒนาขึ้นนี้ นอกจากเล่นแข่งกับมนุษย์ได้แล้ว ยังสามารถให้โปรแกรมเล่นแข่งกันเองโดยเลือกใช้อัลกอริทึมที่แตกต่างกันได้

เราได้ทำการทดลองเพื่อศึกษาเปรียบเทียบอัลกอริทึม ผลการทดลองแสดงให้เห็นถึงข้อดีข้อเสียของทั้งสามอัลกอริทึม ผลที่ได้แสดงให้เห็นว่า 1.)อัลกอริทึม B* ไม่เหมาะสมกับหมากรุกไทย เนื่องจากการนิยามฟังก์ชันฮิวริสติกสำหรับการลู่เข้าของอัลกอริทึมเป็นไปได้ยาก 2.)อัลกอริทึมตัวเลขคอนสไปเรซีสร้างตาเดินได้ดีกว่าอัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา เนื่องจากสืบค้นได้ลึกกว่าเมื่อกำหนดให้แต่ละอัลกอริทึมสร้างโหนดได้ใกล้เคียงกัน และ 3.)อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตาสร้างตาเดินได้ดีกว่าอัลกอริทึมตัวเลขคอนสไปเรซี เนื่องจากสร้างโหนดได้มากกว่า เมื่อกำหนดให้แต่ละอัลกอริทึมสร้างตาเดินภายในเวลาเท่ากัน

โดยสรุป อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามีความเหมาะสมที่จะนำมาใช้ในเกมหมากรุกไทยมากกว่าอัลกอริทึมอื่น เนื่องจากประสิทธิภาพที่เหนือกว่าในด้านความเร็วของการสืบค้น และพื้นที่หน่วยความจำที่ใช้ อย่างไรก็ตามอัลกอริทึมตัวเลขคอนสไปเรซีเป็นอีกทางเลือกหนึ่งที่น่าสนใจ ในแง่ของการสร้างตาเดินที่มีคุณภาพ หากแต่ยังมีความต้องการศึกษาเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพด้านความเร็ว และพื้นที่หน่วยความจำที่ใช้

ผลที่ได้จากการวิจัยคือเราได้โปรแกรมหมากรุกไทยที่สามารถเลือกอัลกอริทึมในการเดินหมากคืออัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา และอัลกอริทึมตัวเลขคอนสไปเรซีที่จะสามารถนำมาใช้เพิ่มประสิทธิภาพและความเข้าใจในการเรียนการสอนวิชาปัญญาประดิษฐ์ นอกจากนั้นส่วนหนึ่งของงานวิจัยได้นำเสนอในการประชุมวิชาการไฟฟ้า ครั้งที่ 18 (พีรพงษ์, บุญเสริม 2538) ซึ่งกระตุ้นให้เกิดการศึกษาและความสนใจในสาขานี้เพิ่มขึ้น

6. ข้อเสนอแนะ

ข้อเสนอแนะเพื่อเป็นแนวทางการวิจัยต่อไปมีดังต่อไปนี้

1. การพัฒนาโปรแกรมหมากรุกไทยต่อไปควรเลือกใช้อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา ซึ่งผลจากการวิจัยได้แสดงให้เห็นแล้วว่า อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตามีความเหมาะสมกว่าอัลกอริทึม B* และอัลกอริทึมตัวเลขคอนสไปเรซี เพราะสร้างตาเดินได้เร็วและสามารถเลือกตาเดินได้ดีกว่าอัลกอริทึมอื่น นอกจากนี้ ควรมีการเปิดสมุดตาเดินในช่วงระยะสุดท้ายของเกม เนื่องจากพบว่า ในช่วงท้ายเกมการเล่นมีรูปแบบของการเดินหมากที่เป็นสูตรสำเร็จแล้วหากปล่อยให้อัลกอริทึมทำการสืบค้นต่อไปจะทำให้การจบเกมเป็นไปได้ช้า

2. เพื่อเพิ่มประสิทธิภาพในการสร้างโหนดของอัลกอริทึมอัลฟาเบตา ควรมีโครงสร้างข้อมูลอีกชุดหนึ่งไว้ใช้เก็บตัวหมากและตำแหน่ง โดยเรียงลำดับของตัวหมากที่มีความสำคัญต่อการเดินจากมากไปน้อย และเมื่อจะสร้างตาเดินในระดับแรกก็จะพิจารณาจากโครงสร้างข้อมูลชุดนี้แทนการพิจารณาจากกระดานตามปกติ ทั้งนี้เนื่องจากวิธีการที่ใช้ในขณะนี้เริ่มพิจารณาจากมุมบนซ้ายมือสุดของกระดานซึ่งยังไม่มีประสิทธิภาพ เพราะตัวหมากที่มีความสำคัญในการเดินขณะนั้นไม่ได้อยู่ที่มุมบนด้านซ้ายสุดเสมอไป อย่างไรก็ตาม แม้ว่าวิธีใหม่นี้จะเสียเวลาในระดับแรก แต่การสืบค้นที่ความลึกระดับถัดไปจะทำได้เร็วขึ้นเพราะโหนดที่มีความสำคัญได้ถูกจัดเรียงไว้ทางซ้ายของต้นไม้เกมแล้ว นอกจากนั้นแล้วการเพิ่มประสิทธิภาพของอัลกอริทึมมินิแมกซ์อาจทำได้โดยใช้เทคนิคย่อยอื่นๆเสริมร่วมกับอัลกอริทึมการลดทอนโดยอัลฟาเบตาเช่น Windowing, Iterative deepening (R.E.Korf, 1985) เป็นต้น

บรรณานุกรม

H.Berliner (1979) The B* Tree Search Algorithm: A Best-First Proof Procedure, Artificial Intelligence 12, pp.23-40.

H.Berliner (1991) Producing Behavior in a Searching Program, CMU Computer Science, A 25th Anniversary Commemorative, ACM Press, pp.311-344.

E.Horowitz, S. Sahni (1978) Fundamentals of Computer Algorithms, Maryland:Computer Science Press, Inc.

F.H.Hsu, T.Anantharaman, M.Campbell and A.Nowatzyk (1990) A Grandmaster Chess Machine, Scientific American 263(4), pp.44-50.

R.E.Korf, (1985) Depth-First Iterative-Deepening:An Optimal Admissible Tree Search, Artificial Intelligence 27, pp.97-109.

L.Lister and J.Schaeffer(1994) An Analysis of the Conspiracy Numbers Algorithm, Journal of Computers & Mathematics with Applications, Vol.27 pp.41-64.

D.A.McAllester (1988) Conspiracy Numbers for Min-Max Search, Artificial Intelligence 35, pp. 287-310

A.J.Palay (1982) The B* Tree Search Algorithm--New Results, Artificial Intelligence 19, pp.145-163.

E.Rich and K.Knight (1991) Artificial Intelligence, McGraw-Hill.

J.Schaeffer (1990) Conspiracy Numbers, Artificial Intelligence 43, pp.67-84.

J.Schaeffer, N.Treloar, P.Lu and R.Lake (1993) "For the World Checkers Championship", AI-magazine vol.14 No.2, pp.28-35.

M.A.Weiss (1993) Data Structures and Algorithm Analysis in C, The Benjamin/Cummings Pub.

ธนะเทพ พันธุ์สุวรรณกิจ (2538) หมากรุกคอมพิวเตอร์, เอกสารประกอบโครงงาน COMPUTER CHESS ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

พีรพงษ์ เจียรณัย และ บุญเสริม กิจศิริกุล (2538) การเปรียบเทียบอัลกอริทึมค้นหาในโปรแกรมหมากรุกไทย, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18 หน้า 730 - 735

จุฬาลงกรณ์มหาวิทยาลัย

โครงการสิ่งประดิษฐ์

รายงานการประดิษฐ์

คอมพิวเตอร์หมากรุกไทย

โดย

บุญเสริม กิจศิริกุล

มีนาคม 2539

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายงานการประดิษฐ์
การออกแบบโปรแกรมหมากรุกไทยและวิธีการใช้โปรแกรม

ในรายงานนี้กล่าวถึงการออกแบบโปรแกรมหมากรุกไทย และวิธีการใช้โปรแกรม

1. การออกแบบโปรแกรมหมากรุกไทย

1.1 กระดานหมากรุกไทย

ในโปรแกรมคอมพิวเตอร์ กระดานหมากรุกไทยจะถูกแทนด้วยอาเรย์ขนาด 8 x 8 เพื่อใช้เป็นกรอบในการเดินของแต่ละอัลกอริทึม และใช้ในการอ่านค่าเพื่อทำการแสดงผลออกทางจอภาพ โดยเขียนอธิบายเป็นภาษาซี ได้ดังนี้

```
int     board[8][8];
```

โดยที่มิติที่ 1 จะบอกแถวนอนของกระดาน และ มิติที่ 2 จะบอกแถวหลักของกระดาน ซึ่งแถวนอนและแถวหลักจะเป็นเลขจำนวนเต็มที่มีค่าอยู่ระหว่างตั้งแต่ 0 จนถึง 7 ส่วนค่าของ board นั้นจะมีค่าอยู่ระหว่างตั้งแต่ -6 จนถึง 6

1.2 ตัวหมากรุกไทย

ตัวหมากรุกไทยจะถูกแปลงให้กลายเป็นเลขจำนวนเต็ม กำหนดให้ดังนี้

ฝ่ายขาว	ขุน	แทนด้วย	6
	เรือ	แทนด้วย	5
	ม้า	แทนด้วย	4
	โคน	แทนด้วย	3
	เม็ด	แทนด้วย	2
	เบี้ย	แทนด้วย	1
ฝ่ายดำ	ขุน	แทนด้วย	-6
	เรือ	แทนด้วย	-5
	ม้า	แทนด้วย	-4
	โคน	แทนด้วย	-3
	เม็ด	แทนด้วย	-2
	เบี้ย	แทนด้วย	-1

จุฬาลงกรณ์มหาวิทยาลัย

1.3 ช่องว่างในกระดานหมากรุกไทย

สำหรับตำแหน่งของกระดานที่ไม่มีหมากตัวใดวางอยู่นั้นจะแทนค่าด้วยเลขจำนวนเต็ม 0 ซึ่งเป็นการบอกให้โปรแกรมคอมพิวเตอร์ได้รับรู้ว่า ที่ตำแหน่งแถวนอนและแถวหลักของกระดานไม่มีหมากตัวใดวางอยู่ ตัวอย่างเช่น board[4][4] = 0 หมายความว่าที่แถวนอนที่ 4 และแถวตั้งที่ 4 เป็นตำแหน่งที่ไม่มีตัวหมากใดวางอยู่

1.4 การอ้างอิงตำแหน่งของหมากที่อยู่บนกระดาน

จากการที่ได้กำหนดให้อาเรย์ 2 มิติที่มีขนาด 8 x 8 ใช้แทนกระดาน ดังนั้นในการอ้างอิงตำแหน่งของหมากในโปรแกรมคอมพิวเตอร์ จึงต้องระบุด้วยเลขจำนวนเต็ม 2 ชุด ชุดแรกจะบอกแถวนอน ชุดที่สองจะบอกแถวหลัก เลขจำนวนเต็มดังกล่าวมีค่าอยู่ระหว่างตั้งแต่ 0 ถึง 7 ซึ่งค่าดังกล่าวมีความสมนัยกับการกำกับตำแหน่งของการเล่นเกมหมากรุกไทยดังนี้

สำหรับแถวนอน :	เกมหมากรุกไทย		โปรแกรมหมากรุกไทย
	8	แทนด้วย	0
	7	แทนด้วย	1
	6	แทนด้วย	2
	5	แทนด้วย	3
	4	แทนด้วย	4
	3	แทนด้วย	5
	2	แทนด้วย	6
	1	แทนด้วย	7

สำหรับแถวตั้ง :	เกมหมากรุกไทย		โปรแกรมหมากรุกไทย
	ก	แทนด้วย	0
	ข	แทนด้วย	1
	ค	แทนด้วย	2
	ง	แทนด้วย	3
	จ	แทนด้วย	4
	ฉ	แทนด้วย	5
	ช	แทนด้วย	6
	ญ	แทนด้วย	7

จุฬาลงกรณ์มหาวิทยาลัย

1.5 ตำแหน่งเริ่มต้นของตัวหมาก

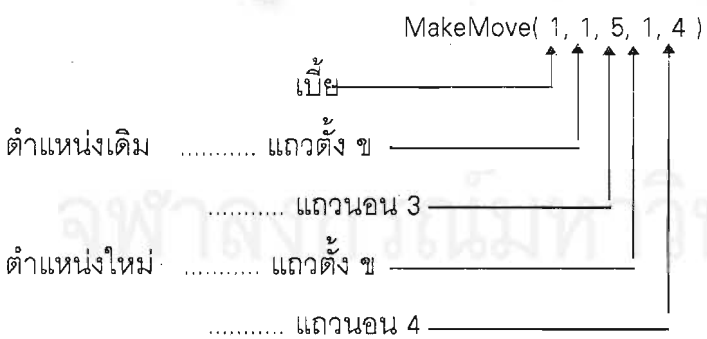
เมื่อเริ่มต้นเกม โปรแกรมคอมพิวเตอร์จะอ่านค่าเริ่มต้นของกระดานจาก board ซึ่งค่าที่อ่านได้ทำให้คอมพิวเตอร์รับรู้ได้ว่า มีหมากอยู่ที่ตำแหน่งใดบ้างบนกระดาน ซึ่งคอมพิวเตอร์ก็จะทำการเก็บค่าในรูปที่แสดงดังต่อไปนี้

0	-5	-4	-3	-2	-6	-3	-4	-5
1	0	0	0	0	0	0	0	0
2	-1	-1	-1	-1	-1	-1	-1	-1
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	0	0
7	5	4	3	6	2	3	4	5
	0	1	2	3	4	5	6	7

รูปที่ 1 ค่ากระดานตอนเริ่มต้นในโปรแกรม

1.6 การเดินของตัวหมาก

การเดินของตัวหมากรุกจะถูกแปลงให้อยู่ในรูปของการย้ายค่าตัวเลขจำนวนเต็มที่ใช้เป็นตัวแทนของหมาก จากตำแหน่งเดิมไปยังตำแหน่งใหม่ และทำการเปลี่ยนค่าหมากของตำแหน่งเดิมให้เป็นเลข 0 เพื่อให้เกิดเป็นช่องว่างหลังจากที่หมากตัวนั้นได้ย้ายไปอยู่ในตำแหน่งใหม่แล้ว ตัวอย่างเช่น การเดินเบี้ยฝ่ายขาว จากตำแหน่ง ข3 ไปยังตำแหน่ง ข4 สามารถเขียนแทนได้ดังนี้



นั่นคือ ที่ตำแหน่งเดิมคือ board[5][1] มีค่า 1 คือมีเบี้ยของฝ่ายขาววางอยู่ และต้องการเดินหมากตัวนี้ไปยังที่ตำแหน่งใหม่คือ board[4][1] ก่อนที่จะทำการเดินนั้นต้องมีการตรวจสอบว่าการเดินนี้ถูกต้องตามกฎการเดินของเบี้ยหรือไม่ ถ้าไม่ถูกต้องก็จะไม่ทำการเดิน ถ้าถูกต้องก็จะมีการย้ายค่าดังนี้

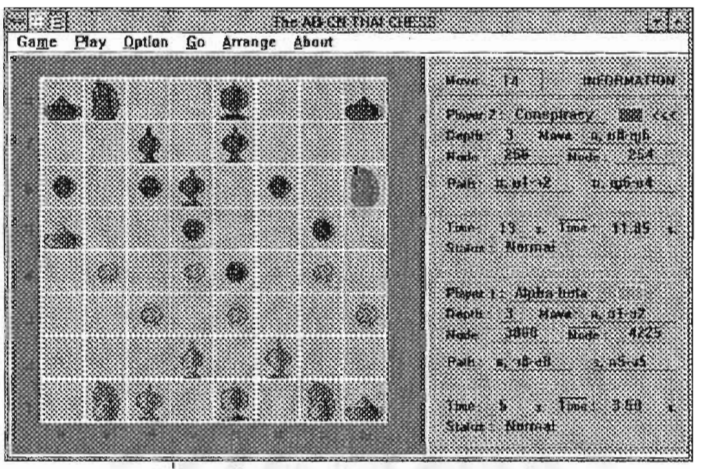
ที่ตำแหน่งใหม่ : board[4][1] = 1;

ที่ตำแหน่งเดิม : board[5][1] = 0;

หลังจากนั้นก็จะทำการปรับปรุงรูปภาพหมากที่อยู่บนจอภาพให้สอดคล้องตรงกับข้อมูลที่อยู่ในอาเรย์

2. การใช้งานของโปรแกรม

โปรแกรมหมากรุกไทยมีหน้าจอดังรูปที่ 2

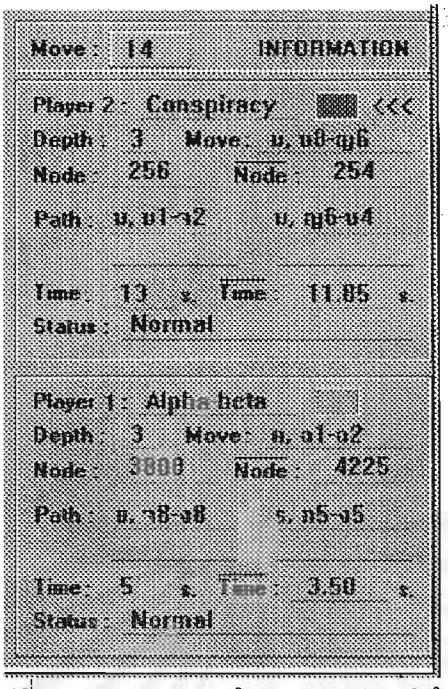


รูปที่ 2 หน้าจอของโปรแกรมหมากรุกไทย

โปรแกรมประกอบด้วย 3 ส่วนสำคัญคือ

1. ส่วนแสดงกระดานดังแสดงในด้านซ้ายของรูปที่ 2 ส่วนนี้ทำหน้าที่แสดงตาเดินและสภาวะของเกมในปัจจุบัน

2. ส่วนแสดงการทำงานของแต่ละอัลกอริทึม ซึ่งมีค่าต่างๆที่แสดงคือ ความลึกสูงสุดของต้นไม้ที่อัลกอริทึมค้นหาได้ (Depth ในรูป) ก้าวเดินล่าสุด (Move) จำนวนโหนดที่กระจายได้ครั้งล่าสุด (Node) จำนวนโหนดเฉลี่ย ($\overline{\text{Node}}$) เส้นทางที่อัลกอริทึมตัดสินใจ (Path) เวลาที่ใช้ในการเดินครั้งล่าสุด (Time) เวลาเฉลี่ยในการเดินแต่ละก้าว ($\overline{\text{Time}}$) และสถานภาพของอัลกอริทึม (Status) ดังแสดงในรูปที่ 3

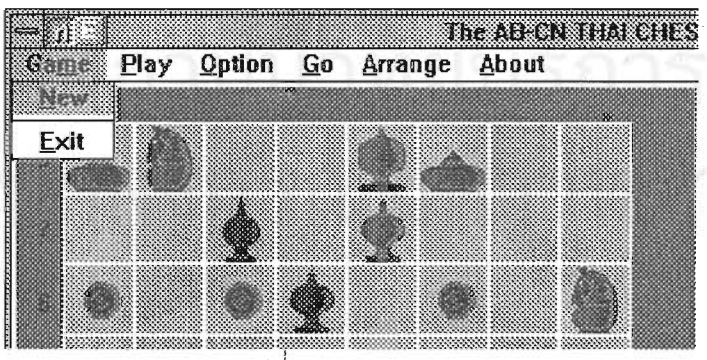


รูปที่ 3 ส่วนแสดงการทำงานของอัลกอริทึม

3. ส่วนของเมนูสำหรับกำหนดค่าต่างๆให้กับโปรแกรม เมนูประกอบด้วย

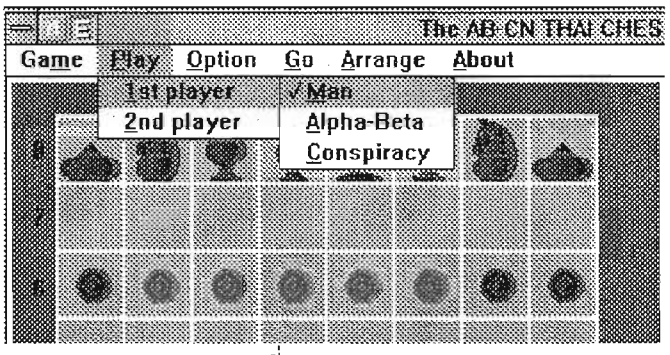
- Game
- Play
- Option
- Go
- Arrange
- About

เมนู GAME ใช้สำหรับกำหนดการเล่นเกมใหม่ (New) หรือจบการทำงาน(Exit) ของโปรแกรม ดังรูปที่ 4



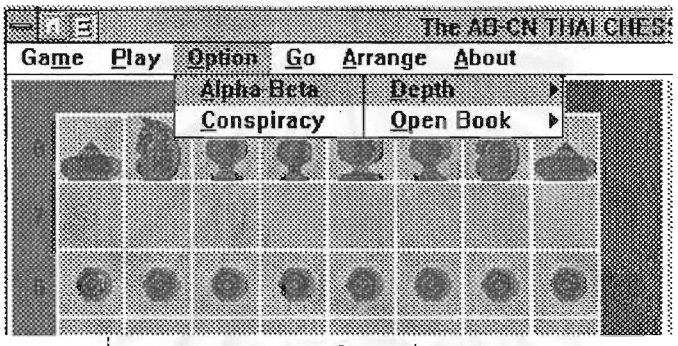
รูปที่ 4 เมนู "Game"

เมนู Play ใช้สำหรับกำหนดผู้เล่นคนแรก (1st player) และผู้เล่นคนที่สอง (2nd player) ว่าแต่ละผู้เล่นเป็นคน (Man) อัลกอริทึมมินิแมกซ์+การลดทอนโดยอัลฟาเบตา(Alpha-Beta) หรืออัลกอริทึมตัวเลขคอนสไปเรซี (Conspiracy) ดังแสดงในรูปที่ 5

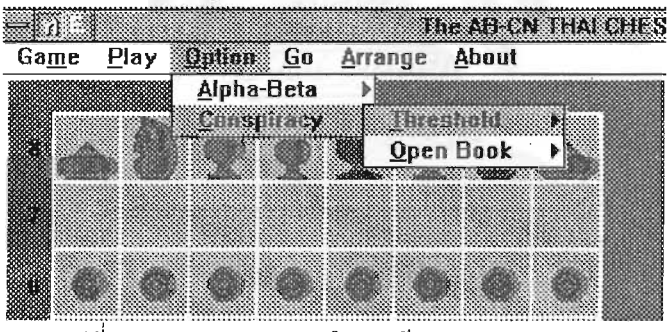


รูปที่ 5 เมนู "Play"

เมนู Option ใช้สำหรับกำหนดพารามิเตอร์ให้กับอัลกอริทึมต่างๆเช่น ความลึกของการค้นหา (Depth) ค่าขีดแบ่ง (Theshold) และการเปิดสมุดมาตรฐาน ดังแสดงในรูปที่ 6 (ก) และ 6(ข)



รูปที่ 6 (ก) เมนู "Option" ในกรณีของ "Alpha-Beta"



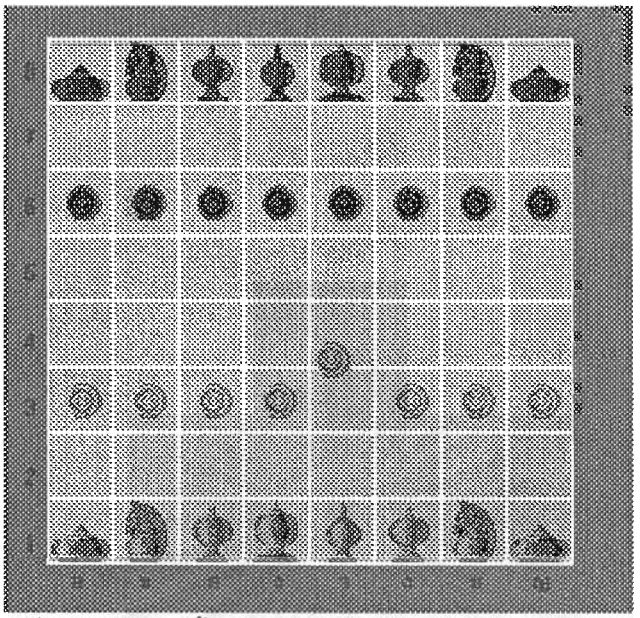
รูปที่ 6 (ข) เมนู "Option" ในกรณีของ "Conspiracy"

เมนู Go ใช้สำหรับให้โปรแกรมเริ่มทำงานหลังจากที่เราได้กำหนดพารามิเตอร์ต่างๆให้กับโปรแกรมโดยเมนู Option แล้ว โปรแกรมจะไม่เริ่มทำงานถ้าไม่สั่งโดยเมนู Go นี้

เมนู Arrage สำหรับให้ผู้ใช้จัดเรียงตำแหน่งเริ่มต้นของแต่ละตัวหมากได้ตามที่ตนเองต้องการ

เมนู About สำหรับแสดงข้อมูลของโปรแกรม

เมื่อเรากำหนดค่าต่างๆให้โปรแกรมเรียบร้อยและสั่งให้โปรแกรมเริ่มทำงานโดยเมนู Go แล้ว ในกรณีที่เรากำหนดให้โปรแกรมใช้อัลกอริทึมหนึ่งๆเล่นกับคน เราสามารถที่จะกำหนดการเดินหมากโดยใช้เมาส์คลิกซ์บนตัวหมากที่ต้องการแล้วลากไปวางยังตำแหน่งที่ต้องการให้ตัวหมากเดินไป ในรูปที่ 7 แสดงการเดิน "เบี้ย" จากตำแหน่ง "จ3" กำลังจะไปตำแหน่ง "จ4"



รูปที่ 7 การเดิน “เบี้ย” จากตำแหน่ง “จ3” ไปตำแหน่ง “จ4”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย