

ผศ. ดร. คັນนีย์ เอื้อพันธุ์วิริยะกุล

บทความความฉลาดเชิงคำนวณสำหรับวิศวกรรมคอมพิวเตอร์

Introduction to Computational Intelligence for Computer Engineering

For Personal Use Only (Copyright คันนีย์ เอื้อพันธุ์วิริยะกุล 2556)



เอกสารคำสอน  
กระบวนวิชา 261456

บทนำความฉลาดเชิงคำนวณสำหรับวิศวกรรม  
คอมพิวเตอร์  
Introduction to Computational Intelligence  
for Computer Engineering

โดย

ผู้ช่วยศาสตราจารย์ ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเชียงใหม่

พ.ศ. 2556

For Personal Use Only (Copyright คำนสนีย เอื้อพัณธีวริยะกุล 2556)

สงวนลิขสิทธิ์ © พ.ศ. 2556 โดย คำนสนีย เอื้อพัณธีวริยะกุล  
All Rights Reserved

## คำนำ Preface



ปัจจุบันการนำความฉลาดเชิงคำนวณ ไปประยุกต์ใช้มีอยู่มากในงานหลายประเภท รวมถึง การรู้จำรูปแบบ ระบบควบคุม ระบบการตัดสินใจ การหาเส้นทางที่สั้นที่สุด ฯลฯ วิศวกรหรือผู้ที่ เกี่ยวข้องควรมีความรู้พื้นฐานเกี่ยวกับ ความฉลาดเชิงคำนวณ และสามารถนำไปใช้งานได้

เอกสารคำสอนเล่มนี้ถูกเขียนขึ้นมา โดยให้มีเนื้อหาครอบคลุมความรู้พื้นฐานในอัลกอริทึม ที่เกี่ยวข้องกับ ความฉลาดเชิงคำนวณ ทั้งนี้หมายถึงอัลกอริทึมในโครงข่ายประสาทเทียม ระบบ ฟัซซี การคำนวณเชิงวิวัฒนาการ และสุดท้ายอัลกอริทึมในความฉลาดเชิงกลุ่ม โดยมีการเขียนที่อยู่ใน ลักษณะที่ให้ผู้อ่านสามารถทำความเข้าใจได้ง่าย ดังนั้นเอกสารคำสอนเล่มนี้เหมาะสำหรับการ เรียนการสอนในระดับปริญญาตรี และยังสามารถเป็นแหล่งความรู้เบื้องต้นสำหรับนักศึกษาใน ระดับบัณฑิตศึกษา ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ วิทยาการคอมพิวเตอร์ และวิศวกรรมไฟฟ้า ทั้งนี้ผู้ที่สนใจในหัวข้อนี้ยังสามารถศึกษาได้ด้วยตนเองจากเอกสารคำสอนเล่มนี้เช่นกัน

ผู้เขียน เขียนเอกสารคำสอนเล่มนี้สำหรับการสอนกระบวนวิชา 261456 บทนำความฉลาด เชิงคำนวณสำหรับวิศวกรรมคอมพิวเตอร์ ซึ่งเป็นกระบวนวิชาเลือกสำหรับนักศึกษาปริญญาตรี ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ในภาคการศึกษา ที่ 1/2556 และ 1/2557 และได้มีการปรับปรุงเนื้อหาให้มีความเหมาะสมในการเรียนการสอนมากขึ้น ทั้งนี้ยังได้ปรับปรุงเนื้อหาให้ผู้อ่านสามารถทำความเข้าใจได้ง่ายขึ้นอีกด้วย

สำหรับคำศัพท์ทางเทคนิคที่เป็นภาษาอังกฤษที่ใช้ในเอกสารคำสอนเล่มนี้ ผู้เขียนพยายาม ใช้คำภาษาไทยที่เหมาะสมที่สุดเท่าที่พึงกระทำได้ และให้สื่อความหมายที่ถูกต้องให้มากที่สุด โดย อ้างอิงถึงศัพท์เทคนิคของวิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์ และศัพท์บัญญัติ ของราชบัณฑิตยสถาน ทั้งนี้ทั้งนั้น ผู้เขียนได้เขียนคำศัพท์ภาษาอังกฤษประกอบไว้ด้วย เพื่อความ เข้าใจที่ถูกต้อง นอกจากนี้ เพื่อความชัดเจนและเพื่อคุณภาพที่ดีของเอกสารคำสอน ผู้เขียนได้จัดทำ รูปในเนื้อหาใหม่เองทั้งหมด

ผู้เขียนหวังเป็นอย่างยิ่งว่า เอกสารคำสอนนี้จะเอื้อประโยชน์ให้ผู้อ่านได้รับความรู้ทางด้าน ความฉลาดเชิงคำนวณ นอกจากนี้ ผู้เขียนยังหวังว่าเอกสารคำสอนนี้จะประโยชน์สำหรับบุคคล ทั่วไปที่มีความสนใจทางด้านนี้อีกด้วย หากผู้อ่านมีข้อคิดเห็นหรือข้อเสนอแนะประการใด อันจะเป็นผล ให้เอกสารคำสอนนี้ มีความสมบูรณ์มากยิ่งขึ้น โปรดส่งอีเมลมายังผู้เขียนที่ [sansanee@eng.cmu.ac.th](mailto:sansanee@eng.cmu.ac.th)

คันสนีย์ เอื้อพันธ์วิริยะกุล

ผู้ช่วยศาสตราจารย์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเชียงใหม่

# ประมวลวิชา Course Syllabus



ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่  
วศ.คพ. 456 (261456) บทนำความฉลาดเชิงคำนวณสำหรับวิศวกรรมคอมพิวเตอร์  
จำนวนหน่วยกิต: 3 หน่วยกิต  
เงื่อนไขที่ต้องผ่านก่อน: ตามความเห็นชอบของภาควิชา

## คำอธิบายลักษณะกระบวนวิชา:

บทนำความฉลาดเชิงคำนวณ บทนำโครงข่ายประสาทเทียม บทนำระบบฟัซซี บทนำการคำนวณเชิงวิวัฒนาการ บทนำความฉลาดเชิงกลุ่ม

## วัตถุประสงค์:

นักศึกษาสามารถอธิบายและติดตามเทคโนโลยีทางด้านความฉลาดเชิงคำนวณโดยใช้คอมพิวเตอร์

## เนื้อหากระบวนวิชา

## จำนวนชั่วโมงบรรยาย

1. บทนำความฉลาดเชิงคำนวณ (Introduction to Computational Intelligence) 3
2. บทนำโครงข่ายประสาทเทียม (Introduction to Neural Networks)
  - 2.1 พื้นฐานโครงข่ายประสาทเทียม (Basic Neural Networks) 3
  - 2.2 เพอร์เซปตรอนหลายชั้น (Multi-Layer Perceptrons) 9
3. บทนำระบบฟัซซี (Introduction to Fuzzy System)
  - 3.1 พื้นฐานฟัซซีเซต (Basic Fuzzy Set) 6
  - 3.2 ระบบอนุมานฟัซซี (Fuzzy Inference System) 6
4. บทนำการคำนวณเชิงวิวัฒนาการ (Introduction to Evolutionary Computing)
  - 4.1 บทนำอัลกอริทึมแบบพันธุกรรม (Introduction to Genetic Algorithm) 7
  - 4.2 บทนำโปรแกรมแบบพันธุกรรม (Introduction to Genetic Programming) 2
  - 4.3 บทนำการคำนวณเชิงวิวัฒนาการ 3

(Introduction to Evolutionary Computing)	
5. บทนำความฉลาดเชิงกลุ่ม	6
(Introduction to Swarm Intelligence)	
<b>รวม</b>	<b><u>45</u></b>

**ลักษณะกิจกรรมการเรียนการสอน:** บรรยายโดยอาจารย์ผู้สอน มีการบ้านสำหรับเนื้อหาแต่ละบท มีการบ้านที่ใช้คอมพิวเตอร์ในการแก้ปัญหา โดยที่นักศึกษาจะต้องส่งรายงานและโปรแกรมคอมพิวเตอร์ของการบ้านแต่ละชิ้นตามระยะเวลาที่กำหนด

#### การประเมินผล:

1. การบ้าน รายงาน และการทดสอบย่อย	40 คะแนน
2. สอบกลางภาค	30 คะแนน
3. สอบปลายภาค	30 คะแนน
คะแนนรวม	100 คะแนน

#### แหล่งวิทยาการหลัก

1. A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd., West Sussex, England, 2005.
2. S. Auephanwiriyaikul, *Introduction to Computational Intelligence*, 2013.

#### แหล่งวิทยาการสำหรับอ่านประกอบ:

1. J. C. Bezdek, "What is computational intelligence?" in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks, and C. J. Robinson (Eds.) Piscataway, NJ: IEEE Press, 1994, pp 1 – 11.
2. E. Bonabeau, M. Dorigo, and G. Theraulez, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Inc., New York, USA., 1999.
3. M. Clerc and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space", *IEEE Transactions on Evolutionary Computation*, vol 6. 2002, pp. 58 – 73.
4. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents", *IEEE Trans. Syst. Man Cybern. B*, vol 26, 1996, pp. 29 – 41.
5. M. Dorigo, *Artificial Life: The swarm intelligence approach*, Tutorial TD1, Congress on Evolutionary Computing, Washington, DC., 1999.
6. D. Dumitrescu, B. Lazzerini, L.C. Jain, and A. Dumitrescu, *Evolutionary Computation*, CRC Press LLC, Florida, USA., 2000.
7. R. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*, Morgan Kaufmann Publishers, USA., 2007.

8. G. K. Eng, and A. M. Ahmad, "Malay Speech Recognition using Self-Organizing Map and Multilayer Perceptron", *Proceedings of the Postgraduate Annual Research Seminar*, 2005.
9. A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd., West Sussex, England, 2002.
10. A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd., West Sussex, England, 2007.
11. D. B. Fogel, "Review of Computational Intelligence: Imitating Life (book Review)", *Proceedings of the IEEE*, Vol. 83, Issue 11, 1995, pp.1588 – 1592.
12. D. B. Fogel, *Evolutionary Computation: Principles and Practice for Signal Processing*, SPIE-The international Society for Optical Engineering, Washington, USA., 2000.
13. D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, New York, USA., 2000.
14. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman, Inc., USA., 2005.
15. S. Haykin, *Neural Networks: A comprehensive Foundation*, Macmillan Publishing Company, Inc. New Jersey, USA., 1994.
16. S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, Inc., USA., 2009.
17. D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization", Technical Report –TRO6, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
18. J. Kennedy "The behavior of particles" in VW Porto, N. Saravanan, D. Waagen(eds), *Proceedings of the 7<sup>th</sup> International Conference on Evolutionary Programming*, 1998. pp. 581 – 589.
19. G. J. Klir and B. Yuan, *Fuzzy Stes and Fuzzy Logic: Theory and Applications*, Prentice Hall Inc., New Jersey, USA., 1995.
20. G. J. Klir, H. Yuan-Yu, B. Yuan and U. S. Clair, *Fuzzy Set Theory: Foundations and Applications*, Prentice Hall Inc., USA., 1997
21. R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.
22. B. Kosko, "Fuzzy associative memories", in *Fuzzy Expert Systems*, A. Kandel, Ed. Reading, MA: Addison-Wesley, 1987.
23. B. Kosko, "Bidirectional Associative Memories", *IEEE Transactions on Systems, Man, and Cybernetics*, vol 18(1), 1988, pp. 49 – 60.
24. J. R. Koza *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, England, 1992.

25. E. Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, Inc., New York USA., 2005.
26. R. Kruse, J. Gebhardt, and F. Klawonn, *Foundations of Fuzzy Systems*, John Wiley & Sons Ltd., West Sussex, England, 1995.
27. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Massachusetts Institute Technology, USA., 1998.
28. N. Ohnishi, A. Okamoto, and N. Sugiem, "Selective Presentation of Learning Samples for Efficient Learning in Multi-Layer Perceptron", *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol 1, 1990, pp. 688 – 691.
29. F. Rosenblatt, "The Perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, vol 65, 1958, pp. 386 – 408.
30. T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons Ltd., West Sussex, England, 2004.
31. P.N. Suganthan, "Particle swarm optimizer with neighborhood operators", *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999, pp. 1958 – 1961.
32. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Elsevier Inc., London, UK., 2009.
33. H. H. Thodberg, "Improving Generalization of Neural Networks through Pruning", *International Journal of Neural Systems*, 1(4), 1991, pp. 317 – 326.
34. A. M. Turing, Computing Machinery and Intelligence, *Mind*, vol 59, 1950, pp. 433 – 460.
35. ] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, 8(3), 1965, pp. 338–353.



# สารบัญ Contents



	หน้า
คำนำ	ค
ประมวลวิชา	ง
สารบัญ	ช
สารบัญตาราง	ฉ
สารบัญรูป	ฉ
<b>บทที่ 1 บทนำความฉลาดเชิงคำนวณ</b>	<b>1</b>
<b>Introduction to Computational Intelligence</b>	
1.1 ประวัติโดยย่อ (Short History)	2
1.2 ชนิดของความฉลาดเชิงคำนวณ (Computational Intelligence Paradigms)	4
1.2.1 โครงข่ายประสาทเทียม (Artificial Neural Networks)	5
1.2.2 ระบบฟัซซี (Fuzzy Systems)	8
1.2.3 การคำนวณเชิงวิวัฒนาการ (Evolutionary Computing)	9
1.2.3 ความฉลาดเชิงกลุ่ม (Swarm Intelligence)	10
คำถามท้ายบทที่ 1	11
<b>บทที่ 2 บทนำโครงข่ายประสาทเทียม</b>	<b>12</b>
<b>Introduction to Neural Networks</b>	
2.1 พื้นฐานโครงข่ายประสาทเทียม (Basic Neural Networks)	12
2.2 เพอร์เซปตรอนหลายชั้น (Multi-layer Perceptrons)	17
2.2.1 โครงข่ายป้อนค่าไปข้างหน้าหลายชั้น (Multi-layer Feedforward networks)	17
2.2.2 อัลกอริทึมการแพร่กระจายย้อนกลับ (Back-propagation Algorithm)	20
2.2.3 การวางนัยทั่วไป (Generalization) และการตรวจสอบความสมเหตุสมผลแบบไขว้ (Cross Validation)	33
2.2.4 แฟกเตอร์ของสมรรถนะ (Performance Factor)	36
2.3 การส่งการจัดระเบียบตนเอง (Self-Organizing Maps)	39
คำถามท้ายบทที่ 2	48
<b>บทที่ 3 บทนำระบบฟัซซี</b>	<b>49</b>
<b>Introduction to Fuzzy System</b>	
3.1 พื้นฐานฟัซซีเซต (Basic Fuzzy Set)	51
3.1.1 ฟังก์ชันสมาชิก (Membership Function)	51
3.1.2 ฟัซซีเซตรูปแบบอื่น	57

3.1.3	การสร้างฟัซซีเซต	58
3.1.4	การดำเนินการในฟัซซีเซต (Operations on Fuzzy Sets)	60
3.1.5	คุณสมบัติของฟัซซีเซต	62
3.2	ระบบอนุมานฟัซซี (Fuzzy Inference System)	67
3.2.1	เซตจํานวน (Linguistic Hedges)	68
3.2.2	การหาเหตุผลโดยประมาณ (Approximate Reasoning)	68
3.2.3	วิธีการในระบบควบคุมฟัซซี (Approached to Fuzzy Control)	84
3.2.3.1	วิธีการของ Mamdani (The Mamdani model)	84
3.2.3.2	วิธีการของ Takagi-Sugeno (The Takagi-Sugeno Model)	87
	คำถามท้ายบทที่ 3	89
<b>บทที่ 4</b>	<b>บทนำการคำนวณเชิงวิวัฒนาการ</b>	<b>90</b>
	<b>Introduction to Evolutionary Computing</b>	
4.1	บทนำอัลกอริทึมแบบพันธุกรรม (Introduction to Genetic Algorithm)	92
4.1.1	การแทน (Representation) โดยโครโมโซม (Chromosome)	92
4.1.2	ฟังก์ชันความเหมาะสม (Fitness Function)	94
4.1.3	ประชากรเริ่มต้น (Initial Population)	95
4.1.4	ตัวดำเนินการคัดเลือก (Selection Operator)	96
4.1.4.1	การคัดเลือกแบบสุ่ม (Random Selection)	97
4.1.4.2	การคัดเลือกตามสัดส่วน (Proportional Selection)	97
4.1.4.3	การคัดเลือกโดยขึ้นอยู่กับกลไกการสเกล และการจัดอันดับ	
	(Selection Based on Scaling and Ranking Mechanisms)	101
4.1.5	กระบวนการรักษานิยม (Elitism)	107
4.1.6	ตัวดำเนินการรวมกันใหม่ (Recombination Operator)	107
4.1.6.1	การข้ามสายพันธุ์แบบหนึ่งจุด (One-point Crossover)	108
4.1.6.2	การข้ามสายพันธุ์แบบสองจุด (Two-point Crossover)	109
4.1.6.3	การข้ามสายพันธุ์แบบ T จุด (T-point Crossover)	109
4.1.6.4	การข้ามสายพันธุ์แบบสลั (Punctuated Crossover)	112
4.1.6.5	การข้ามสายพันธุ์แบบเอกรูป (Uniform Crossover)	113
4.1.7	การกลายพันธุ์ (Mutation)	114
4.1.8	ตัวดำเนินการการผกผัน (Inversion Operator)	115
4.2	บทนำโปรแกรมแบบพันธุกรรม (Introduction to Genetic Programming)	119
4.3	บทนำการคำนวณเชิงวิวัฒนาการ (Introduction to Evolutionary Computing)	127
	คำถามท้ายบทที่ 4	129
<b>บทที่ 5</b>	<b>บทนำความฉลาดเชิงกลุ่ม</b>	<b>131</b>
	<b>Introduction to Swarm Intelligence</b>	
5.1	การหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (Particle Swarm Optimization)	133
5.1.1	อัลกอริทึมที่ดีที่สุดแบบรายบุคคล (Individual Best)	135

5.1.2	อัลกอริทึมที่ดีที่สุดแบบรวม (Global Best)	136
5.1.3	อัลกอริทึมที่ดีที่สุดแบบเฉพาะที่ (Local Best)	138
5.1.4	ตัวแปรในอัลกอริทึมการหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (Particle Swarm Optimization Parameters)	139
5.1.5	การปรับปรุง PSO (Modification PSO)	140
5.2	การหาค่าที่เหมาะสมที่สุดโดยอาณานิคมมด (Ant Colony Optimization)	142
5.2.1	ระบบมด (Ant System)	144
5.2.2	ระบบอาณานิคมมด (Ant Colony System)	148
	คำถามท้ายบทที่ 5	150
	<b>บรรณานุกรม</b>	151
	<b>Bibliography</b>	

## สารบัญตาราง List of Tables



ตารางที่	หน้า
2.1 ตารางความจริงของ AND logic	16
2.2 อินพุตและเอาต์พุตสำหรับตัวอย่างที่ 2.1	17
2.3 ตารางความจริงสำหรับ XOR logic	18
2.4 อินพุตและเอาต์พุตสำหรับตัวอย่างที่ 2.2	19
3.1 เซตสากลของระดับการศึกษา	53
3.2 พัชชีเซต $A$	54
3.3 คำตอบจากกรรมการทั้ง 10	58
3.4 ค่าความเป็นสมาชิกของคนไข้ในพัชชีเซต $A$ , $B$ และ $A \cup B$	60
3.5 ค่าความเป็นสมาชิกของคนไข้ในพัชชีเซต $A$ , $B$ และ $A \cap B$	61
3.6 กฎและค่าคงที่ที่ใช้ในระบบควบคุมพัชชี	88
4.1 ตัวอย่างของประชากรและค่าความเหมาะสม	95
4.2 ทายาทของโครโมโซมในตารางที่ 4.1	95
4.3 ประชากรในรุ่นที่ $t+1$	95
4.4 การคำนวณจากลออูเล็ด	99
4.5 การสร้างประชากรใหม่	100
4.6 ความน่าจะเป็นการคัดเลือกและค่าคาดหวังของจำนวนของทายาท	105
4.7 โครโมโซมรุ่นที่ 1	116
4.8 โครโมโซมลูก	117
4.9 เคสทดสอบ	125

# สารบัญรูป

## List of Figures



รูปที่		หน้า
1.1	ความเชื่อมโยงของชนิดของความฉลาดเชิงคำนวณ	4
1.2	เซลล์ประสาททางชีววิทยา (biological neuron)	5
1.3	เซลล์ประสาทเทียม (artificial neuron)	6
1.4	โครงข่ายประสาทเทียม (artificial neural networks) (ก) $p-4-2$ และ (ข) $p-4-3-2$	7
1.5	โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 1.1	8
2.1	เซลล์ประสาทเทียมในกรณีที่อินพุตของฟังก์ชันเป็น (ก) $net-\theta$ และ (ข) $net+\theta$	12
2.2	ฟังก์ชันเชิงเส้น	13
2.3	ฟังก์ชันขั้น	14
2.4	ฟังก์ชันแรนดัม	14
2.5	ฟังก์ชันซิกมอยด์ (ก) ฟังก์ชันลอจิสติก (ข) ฟังก์ชันไฮเพอร์โบลิคแทนเจนต์	15
2.6	ฟังก์ชันเกาส์เซียน	15
2.7	โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 2.1	16
2.8	เส้นขอบการตัดสินใจ	16
2.9	เปอร์เซปตรอนหลายชั้นสำหรับตัวอย่างที่ 2.2	18
2.10	เส้นขอบการตัดสินใจของ (ก) จุดที่ 1 (ข) จุดที่ 2 และ (ค) โครงข่ายประสาทเทียมทั้งหมด	19
2.11	(ก) โครงข่ายประสาทเทียม 3-4-1 (ข) ลักษณะของระนาบเกินของโครงข่ายในรูป (ก)	20
2.12	เกรเดียนต์เวกเตอร์ ( $\nabla \mathcal{E}(w)$ )	22
2.13	สัญญาณฟังก์ชัน (เส้นทึบ) และสัญญาณความผิดพลาด (เส้นประ)	23
2.14	รายละเอียดของสัญญาณที่เซลล์ที่ $j$ ที่ชั้นเอาต์พุต	24
2.15	รายละเอียดของสัญญาณที่เซลล์ที่ $j$ ที่ชั้นซ่อน	26
2.16	ลักษณะโครงข่ายประสาทเทียมที่มีเลขลำดับชั้นกำกับ	27
2.17	โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 2.5 ในการทำซ้ำที่ (ก) $t$ และ (ข) $t-1$	31
2.18	การปรับเส้นโค้ง (ก) ที่มีการวางนัยทั่วไป และ (ข) ที่มีการเหมาะสมเกินไป	34
2.19	การแบ่งชุดข้อมูลสำหรับการตรวจสอบความสมเหตุสมผลแบบไขว้ 4 โฟลต์	35
2.20	ผลกระทบของเอาต์ไลเออร์	37
2.21	การส่งการจัดระเบียบตนเองตามแบบจำลองของ Kohonen	39
2.22	ลักษณะของฟังก์ชันเพื่อนบ้านรอบเซลล์ผู้ชนะ $i(x)$ สำหรับแลตทิซ (ก) 1 มิติ และ (ข) 2 มิติ	41
2.23	ฟังก์ชันเพื่อนบ้านเกาส์เซียน	41
2.24	เวกเตอร์น้ำหนักของแลตทิซ 2 มิติ ขนาด $10 \times 10$ หลังจากที่ถูกฝึกสอนด้วยตัวเลข 0 - 9	43
2.25	ฟังก์ชันเพื่อนบ้านที่เป็นสี่เหลี่ยม ที่ขนาดต่างกัน เซลล์ที่เป็นจุดสีดำคือเซลล์ผู้ชนะ	44

2.26	(ก) แลตทิส และ (ข) เซลล์ผู้ชนะและฟังก์ชันเพื่อนบ้าน สำหรับตัวอย่างที่ 2.6	45
2.27	ความสัมพันธ์ระหว่างการส่งลักษณะ $\Phi$ และเวกเตอร์น้ำหนัก $w_i$ ของเซลล์ผู้ชนะ $i$	46
2.28	โครงข่ายประสาทเทียมที่ใช้การจัดระเบียบตนเองในการเลือกลักษณะและเปอร์เซปตรอนหลายชั้นในการจำแนกคำ	47
3.1	เซตของเหรียญไทย	50
3.2	ฟังก์ชันสมาชิกของฟัซซีเซต 'แก่'	50
3.3	ฟังก์ชันลักษณะของคริสป์เซต $A$	52
3.4	ฟังก์ชันสมาชิก $A$	52
3.5	ฟังก์ชันสมาชิกของ (ก) 'young' และ (ข) 'คนตัวใหญ่'	53
3.6	ฟังก์ชันสมาชิกของฟัซซีเซต 'การศึกษาน้อย(0)' 'การศึกษาสูง( $\diamond$ )' 'การศึกษาสูงมาก(+)'	54
3.7	การแทนฟังก์ชันสมาชิกด้วยเรขาคณิต	55
3.8	ฟังก์ชันสมาชิกของฟัซซีเซต 'ประมาณ 6'	55
3.9	(ก) ฟังก์ชันรูปสามเหลี่ยม (triangular shape) (ข) ฟังก์ชันรูปสี่เหลี่ยมคางหมู (trapezoidal) และ (ค) ฟังก์ชันระฆังคว่ำ (Bell-shaped)	56
3.10	ฟัซซีเซตแบบช่วง	57
3.11	ฟัซซีเซตแบบชนิดที่ 2 (type-2 fuzzy set)	57
3.12	(ก) ฟัซซีเซตแบบระดับที่ 2 'young' และ (ข) ฟัซซีเซต 'child'	58
3.13	(ก) S-function (ข) $\pi$ -function	59
3.14	ฟัซซีเซต 'Experience' และ 'Inexperience'	60
3.15	ฟังก์ชันสมาชิกของฟัซซีเซต 'Old' 'Very Old' และ 'Somewhat Old'	62
3.16	ฟัซซีเซต $E$	63
3.17	ตัวอย่างของทฤษฎีที่ 3.1	65
3.18	ตัวอย่างของทฤษฎีที่ 3.3 [Klir95] (ก) ฟัซซีเซต $A$ (ข) การแยกของฟัซซีเซต $A$ ให้เป็น $_{0.3}A$ $_{0.6}A$ และ $_{1}A$	66
3.19	แผนภาพคอนทัวร์ของ (ก) คอนเวกฟัซซีเซต (ข) ฟัซซีเซตที่ไม่คอนเวก	67
3.20	ฟัซซีเซตไม่ต่อเนื่อง	67
3.21	ความสัมพันธ์ระหว่าง 2 ตัวแปร (ก) $x \rightarrow y$ โดยที่ $y=f(x)$ (ข) $A \rightarrow B$ โดยที่ $B=\{y \in Y \mid y=f(x), x \in A\}$	69
3.22	ความสัมพันธ์ระหว่าง 2 ตัวแปรที่ไม่ใช่ฟังก์ชันและหาฟังก์ชันลักษณะได้จากสมการที่ 3.33 (ก) $\chi = u$ (ข) $\chi \in A$	70
3.23	การประกอบของกฎการสื่อความ (compositional rule of inference)	70
3.24	(ก) ใช้ correlation-min (ข) ใช้ correlation-product สำหรับกรณีที่ $A'$ เป็นคริสป์เซต	78
3.25	ตัวอย่างของระบบฟัซซีที่มี 2 กฎและความจริงอินพุตเป็นสิ่งที่รู้แน่นอน	79
3.26	ตัวอย่างของระบบฟัซซีที่มี 2 กฎและความจริงอินพุตเป็นสิ่งที่ไม่แน่นอน	80
3.27	ตัวอย่างของฟัซซีเซตเอาต์พุตของระบบฟัซซีที่ไม่ใช่คอนเวกซ์ฟัซซีเซต	81
3.28	ระบบควบคุมฟัซซี	82

3.29	ลูกตุ้มกลับหัว	83
3.30	ระบบควบคุมพีซีโดยใช้วิธี Mamdani	86
3.31	อินพุตของระบบควบคุมพีซีของการควบคุมรถผ่านทางไค้	87
4.1	ผังงานอย่างง่ายของการคำนวณเชิงวิวัฒนาการ	90
4.2	(ก) โครงข่ายประสาทเทียม และ ลักษณะของโครโมโซมของ (ก) สำหรับ (ข) ยีนที่เป็นจำนวนจริง และ (ค) ยีนที่เป็นค่าไบนารี	94
4.3	เชกเตอร์ในลัทธิของแต่ละโครโมโซม	99
4.4	โครโมโซม พ่อ-แม่ ที่ผ่านการถูกเลือก	108
4.5	โครโมโซมลูกหลังจากการข้ามสายพันธุ์	108
4.6	โครโมโซมพ่อ-แม่ สำหรับการข้ามสายพันธุ์แบบสองจุด	109
4.7	โครโมโซม	109
4.8	(ก) โครโมโซมพ่อ-แม่ $(x,y)$ และ (ข) โครโมโซมลูกหลังจากการข้ามสายพันธุ์ 3 จุด	110
4.9	(ก) โครโมโซมพ่อ-แม่ $(x,y)$ และ (ข) โครโมโซมลูกหลังจากการข้ามสายพันธุ์ 4 จุด	110
4.10	การข้ามสายพันธุ์แบบเอกรูป ด้วยเวกเตอร์หน้ากาก $m$	113
4.11	โครงข่ายประสาทเทียม 3-2-1	117
4.12	โครงข่ายประสาทเทียมที่เป็นโครโมโซมแม่	117
4.13	โครงข่ายประสาทเทียมที่เป็นโครโมโซมลูก	118
4.14	โครงข่ายประสาทเทียมในรูปที่ 4.11 ที่ถูกกลายพันธุ์	118
4.15	ทรีสำหรับ XOR	120
4.16	ทรีสำหรับสมการที่ 4.60	120
4.17	(ก) พ่อแม่ตัวที่ 1 (ข) พ่อแม่ตัวที่ 2 และ (ค) ลูกที่ได้ (ทรีส่วนย่อยที่อยู่ในกรอบเส้นประคือทรีส่วนย่อยในพ่อแม่ตัวที่ 1 ที่จะถูกแทนส่วนทรีส่วนย่อยในพ่อแม่ตัวที่ 2)	122
4.18	ลูกที่ได้จากการข้ามสายพันธุ์ของพ่อแม่ในรูปที่ 4.17	122
4.19	(ก) ทรีดั้งเดิม และทรีที่ถูกกลายพันธุ์ (ข) ที่ไหนดฟังก์ชัน (ค) ที่ไหนดปลายทาง (ง) สลับ (จ) ขยาย (ฉ) เกาส์เซียน และ (ช) ตัดปลาย (วงกลมที่เป็นเส้นประคือตำแหน่งที่เกิดการกลายพันธุ์)	124
4.20	ทรีสำหรับรายบุคคลที่ (ก) 1 (ข) 2 และ (ค) 3	126
4.21	การข้ามสายพันธุ์จากพ่อแม่ตัวที่ (ก) 1 และ (ข) 2 (ตำแหน่งการข้ามสายพันธุ์อยู่ที่เส้นประ) สร้างลูกตัวที่ (ค) 1 และ (ง) 2	127
4.22	การกลายพันธุ์ที่ได้ไหนด — ของรูปที่ 4.20(ก) (วงกลมเส้นประคือทรีส่วนย่อยที่ถูกสร้างใหม่)	127
5.1	รูปแบบของเพื่อนบ้านสำหรับการจัดระเบียบกลุ่มของอนุภาค (ก) ทอพอโลยีแบบดาว (ข) ทอพอโลยีแบบวงแหวน และ (ค) ทอพอโลยีแบบวงล้อ	134
5.2	การเดินทางหาอาหารของมดงาน (ก) ในตอนต้น และ (ข) เมื่อเวลาผ่านไป	143
5.3	กราฟ $(N,E)$ ที่มี 4 เมือง และทุกเมืองไม่ได้เชื่อมต่อกัน	144
5.4	กราฟสำหรับตัวอย่างที่ 5.3	147

# บทนำความฉลาดเชิงคำนวณ

## Introduction to Computational Intelligence

### บทที่ 1

เพื่อให้เกิดความเข้าใจอย่างถ่องแท้ถึงคำว่าความฉลาดเชิงคำนวณ (Computational Intelligence) จะขอกล่าวถึงความหมายของ ความฉลาด (intelligence) ปัญญาประดิษฐ์ (Artificial Intelligence) และ ความฉลาดเชิงคำนวณ (Computational Intelligence) ส่วนในหัวข้อย่อยในบทนี้จะกล่าวถึงประวัติ และสาขาย่อยในความฉลาดเชิงคำนวณที่จะกล่าวถึงอย่างคร่าวๆ ในเอกสารคำสอนนี้ เพื่อเป็นการทำความเข้าใจเบื้องต้น ซึ่งสาขาเหล่านั้นคือโครงข่ายประสาทเทียม (Artificial Neural Networks) ระบบฟัซซี (Fuzzy Systems) การคำนวณเชิงวิวัฒนาการ (Evolutionary Computing) และความฉลาดเชิงกลุ่ม (Swarm Intelligence)

ความฉลาด (intelligence) [Engelbrecht07] ตามการแปลจากพจนานุกรมคือความสามารถในการทำความเข้าใจ (comprehend) หรือความเข้าใจและสร้างประโยชน์จากประสบการณ์ หรือความสามารถในการแปล และมีความสามารถในการใช้ความคิด และสร้างเหตุผล ทั้งนี้ความฉลาดยังรวมถึงความคิดสร้างสรรค์ ทักษะ ความสำนึก อารมณ์ และการรู้เอง

ส่วนปัญญาประดิษฐ์ (Artificial Intelligence (AI)) [Engelbrecht07] ตามนิยามของ IEEE Neural Networks Council ปี ค.ศ. 1996 คือ การศึกษาในการทำให้คอมพิวเตอร์ทำในสิ่งที่มนุษย์ทำได้ดีกว่า ซึ่งโดยปกติแล้วเป็นการศึกษาเพื่อพัฒนาอัลกอริทึมที่ได้จากการสร้างต้นแบบ (model) ของความฉลาดทางชีววิทยาและความฉลาดทางธรรมชาติ ซึ่งถูกเรียกว่าระบบที่ฉลาด (intelligent system) ให้สามารถแก้ปัญหาที่มีความซับซ้อน (complex problem) หรือในอีกความหมายหนึ่ง ปัญญาประดิษฐ์เป็นการผสมผสานงานวิจัยในหลายสาขาวิชา เช่นวิทยาการคอมพิวเตอร์ (computer science) สรีรวิทยา (physiology) ปรัชญา (philosophy) สังคมวิทยา (sociology) และชีววิทยา (biology) เป็นต้น

ทั้งนี้ความฉลาดเชิงคำนวณ (Computational Intelligence (CI)) [Fogel95] คือ การรับรู้ของจิตในระดับต่ำ ("low-level cognition in the style of the mind" [Bexdek94, Fogel95]) และระบบจะมีความฉลาดเชิงคำนวณถ้าเป็นการคำนวณที่เกี่ยวข้องกับข้อมูลที่เป็นตัวเลข ซึ่งถือว่าเป็นข้อมูลระดับต่ำ โดยที่ระบบนั้นมีส่วนประกอบเป็นการรู้จำต้นแบบ (pattern recognition) และไม่ได้ใช้ความรู้ (knowledge) ในด้านของปัญญาประดิษฐ์ และนอกเหนือจากนี้ระบบนี้ต้องมีความสามารถในการปรับตัวในการคำนวณ (computational adaptivity) ทนต่อความผิดพลาดในการคำนวณ (computationally fault tolerance) มีความเร็วในการคำนวณใกล้เคียงกับมนุษย์ และมีอัตราความผิดพลาด (error rate) ที่ใกล้เคียงกับมนุษย์ ทั้งนี้ความฉลาดเชิงคำนวณต้องมีความสามารถในการปรับตัวเสมอ ทำให้ความหมายโดยสรุปของความฉลาดเชิงคำนวณ [Engelbrecht05] คือการศึกษาของกลไก (mechanisms) ที่สามารถปรับตัวได้ ทำให้มีพฤติกรรมที่ฉลาดในสภาพแวดล้อมที่ซับซ้อนและเปลี่ยนแปลง ซึ่งกลไกเหล่านี้รวมถึงกลไกใน AI ที่สามารถปรับตัวได้ เช่น โครงข่ายประสาทเทียม (Artificial Neural Networks) ระบบฟัซซี (Fuzzy Systems) การคำนวณเชิงวิวัฒนาการ (Evolutionary Computing) และความฉลาดเชิงกลุ่ม (Swarm Intelligence) เป็นต้น โดยกลไกเหล่านี้ไม่ใช่ปัญญาประดิษฐ์ดั้งเดิม (traditional AI) เนื่องจากปัญญาประดิษฐ์ดั้งเดิม ไม่มีความสามารถในการปรับตัวหรือการเรียนรู้แต่อย่างใด ถึงแม้ว่าในบางสถานการณ์ อัลกอริทึมในปัญญาประดิษฐ์แบบดั้งเดิมสามารถแก้ไขปัญหาได้ดี



ใกล้เคียงกับมนุษย์ แต่ ปัญญาประดิษฐ์ดั้งเดิมเหล่านี้ไม่สามารถปรับตัวเข้าสถานการณ์ใหม่ได้ ทำให้ไม่สามารถแก้ปัญหาที่มีความซับซ้อนมากขึ้นได้ดี นั่นเอง ดังนั้นสามารถสรุปได้ว่าข้อแตกต่างระหว่างปัญญาประดิษฐ์ดั้งเดิม และความฉลาดเชิงคำนวณคือ อัลกอริทึมในความฉลาดเชิงคำนวณสามารถปรับตัว หรือมีการเรียนรู้ ในขณะที่อัลกอริทึมในปัญญาประดิษฐ์ดั้งเดิม ไม่มีนั่นเอง

## 1.1 ประวัติโดยย่อ (Short History)

ก่อนที่จะกล่าวถึงหัวข้อต่างๆในความฉลาดเชิงคำนวณ ที่จะกล่าวถึงในเอกสารคำสอนนี้ จะขอกล่าวถึงประวัติโดยย่อ ของความพยายามที่จะทำให้เครื่องคอมพิวเตอร์ มีความฉลาดหรือแก้ปัญหาที่มีความซับซ้อน ได้ใกล้เคียงกับมนุษย์ โดยเริ่มต้นจาก [Engelbrecht07] Aristotle (384 – 322 ก่อนคริสตกาล) ที่น่าจะเป็นคนแรกที่พยายามอธิบาย และจัดระบบการหาเหตุผลที่สามารถอนุมานได้ (deductive reasoning) ซึ่งถูกเรียกว่าตรรกบท (syllogism) หลังจากนั้น Ramon Llull (ค.ศ. 1235 – 1316) พยายามสร้างเครื่อง Ars Magna ซึ่งเป็นเครื่องที่ประกอบไปด้วย เซตของวงล้อ และเป็นเครื่องที่ถูกสร้างขึ้นมาเพื่อตอบคำถามทุกคำถาม ต่อมา Gottfried Leibniz (ค.ศ. 1646 – 1716) ให้เหตุผลว่า การมีอยู่ของ แคลคูลัสฟีโลโซฟิคัส (calculus philosophicus) ซึ่งคือ พีชคณิตสากล (universal algebra) ทำให้สามารถแทนความรู้รวมถึงความจริง ในระบบอนุมาน (deductive system) ได้ หลังจากนั้นในปี ค.ศ. 1854 George Boole พัฒนาพื้นฐานของ ลอจิกเชิงประพจน์ (propositional logic) และพัฒนาพื้นฐานของ เพรดิเคตแคลคูลัส ในปี ค.ศ. 1879 ซึ่งทั้งแคลคูลัสเชิงประพจน์ (propositional calculus) และเพรดิเคตแคลคูลัส เป็นส่วนหนึ่งของเครื่องมือของปัญญาประดิษฐ์ นั่นเอง

ทั้งนี้นิยามของคำว่าปัญญาประดิษฐ์ [Engelbrecht07] ได้ถูกตั้งขึ้นในช่วงทศวรรษ 1950 โดย Alan Turing และในช่วงเดียวกันนี้ Turing ได้ทำการศึกษาว่า จะทำอย่างไรให้เครื่องจักรสามารถเลียนแบบสมองของมนุษย์ โดยผลงานนี้ได้ถูกตีพิมพ์ในชื่อว่า เครื่องจักรฉลาด (Intelligent Machinery) ซึ่งถือได้ว่าเป็นผลงานตีพิมพ์อันแรกในงานวิจัยทางด้านปัญญาประดิษฐ์นั่นเอง และหลังจากนั้น Turing ได้ตีพิมพ์ผลงานที่ชื่อว่า ฐานหลักทางเคมีของกำเนิดสิ่งมีชีวิต (The Chemical Basis of Morphogenesis) ซึ่งอาจจะเป็นงานวิจัยอันแรกของ ชีวิตประดิษฐ์ (artificial life) ในขณะเดียวกัน Turing ได้ตีพิมพ์ผลงานที่เกี่ยวข้องกับการทดสอบความฉลาดของเครื่องคอมพิวเตอร์ที่ถูกเรียกว่า การทดสอบของ Turing (Turing test) [Turing 1950] ซึ่งเป็นการทดสอบที่ให้ผู้ทดสอบหนึ่งถามคำถามโดยใช้คีย์บอร์ด ให้กับคนที่อยู่ห้องหนึ่ง และเครื่องคอมพิวเตอร์ที่ถูกตั้งไว้ในอีกห้องหนึ่ง ถ้าคนที่ถามคำถามไม่สามารถแยกได้ว่าห้องไหนที่คนหรือเครื่องคอมพิวเตอร์อยู่ แสดงว่าเครื่องคอมพิวเตอร์มีความฉลาด นอกเหนือจากนี้ ในปีค.ศ. 1956 John McCarthy ซึ่งถือว่าเป็นบิดาของปัญญาประดิษฐ์ ได้จัดงานประชุมวิชาการ Dartmouth และในงานนี้งานวิจัยทางด้านปัญญาประดิษฐ์ได้ถูกวางรากฐานไว้

ในปีค.ศ. 1943 [Haykin09] Warren McCulloch และ Walter Pitts ได้เสนอแนวคิดเรื่องโครงข่ายประสาทในมุมมองของเครื่องคอมพิวเตอร์ ในขณะที่ Donald O. Hebb ได้อ้างถึงกฎข้อแรกของการเรียนรู้แบบทำให้เป็นระบบเอง (self-organized learning) ในปีค.ศ. 1949 [Haykin09] และในช่วงปีค.ศ. 1956 – 1969 [Engelbrecht07] มีงานวิจัยมากมายที่เกี่ยวข้องกับการจำลอง เซลประสาททางชีววิทยา (biological neuron) ซึ่งงานวิจัยที่มีชื่อมากคืองานวิจัยของ

Frank Rosenblatt ที่เป็งานวิจัยที่เก็วข้องกับเปอร์เซปตรอน (perceptron) และงานวิจัยของ Bernard Widrow และ Marcian Edward Hoff, Jr. ที่เก็วข้องกับ อะดาลิน (adaline) แต่ในปี ค.ศ. 1969 Marvin Minsky และ Seymour Papert ได้ตีพิมพ์ผลงานวิจัยในหนังสือชื่อ Perceptron ที่พิสูจน์ว่า โครงข่ายประสาทเทียม (artificial neural networks) ที่ขยายมาจาก เปอร์เซปตรอน เป็นเปอร์เซปตรอนหลายชั้น (multi-layer perceptron) ไม่สามารถทำงานได้ดี ทำให้งานวิจัยที่เก็วข้องกับโครงข่ายประสาทเทียมหยุดชะงักลง จนถึงช่วงกลางศตวรรษที่ 1980 แต่ในช่วงนี้ยังคงมีนักวิจัยที่พยายามศึกษาโครงข่ายประสาทเทียม และทำให้โครงข่ายประสาท เทียมกลับมาเป็นที่นิยมอีก อาทิเช่น Stephen Grossberg, Gail A. Carpenter, Shun-ichi Amari, Teuvo Kohonen, Ken-Ichi Funahashi และนักวิจัยท่านอื่นๆ ที่ไม่ได้กล่าวถึง ณ. ที่นี้ แต่ผลงานวิจัยที่ถือได้ว่า เป็นเครื่องหมายบ่งชี้และทำให้งานวิจัยด้านโครงข่ายประสาทเทียมกลับมา เป็นที่นิยมอีกครั้ง คืองานวิจัยของ John Hopfield, Geoffrey E. Hinton, David E. Rumelhart และ James L. McClelland ซึ่งงานวิจัยเหล่านี้เกิดขึ้นในช่วงต้นถึงกลางศตวรรษที่ 1980 และใน ปลายศตวรรษเดียวกันนี้งานวิจัยด้านโครงข่ายประสาทเทียมกลับมาเป็นที่นิยมอีกครั้ง จนถึง ปัจจุบันนี้ [Engelbrecht07]

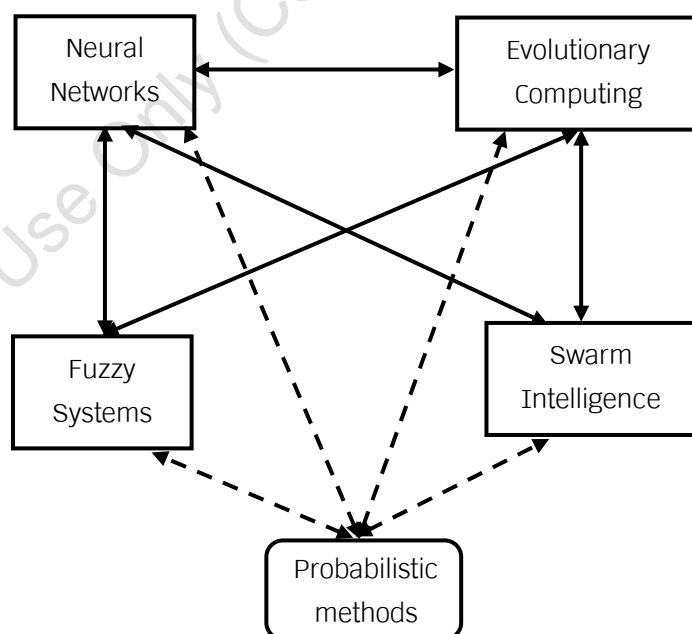
ทางด้านของงานวิจัยที่เก็วข้องกับทฤษฎีฟัซซี (Fuzzy Set Theory) [Engelbrecht07] นั้นมีความเชื่อว่า เริ่มมาตั้งแต่สมัยของพระสัมมาสัมพุทธเจ้า ซึ่งได้อธิบายถึงพื้นที่ที่ไม่ใช่พื้นที่สี ขาวหรือสีดำ แต่เป็นพื้นที่ที่เป็นสีเทา ในด้านความคิด แต่ทั้งนี้จุดเริ่มต้นจริงๆของฟัซซีลอจิก (Fuzzy Logic) คือเริ่มมาจากการคิดค้น ลอจิก 2 ค่า (2-valued logic) ของ Aristotle และในปี ค.ศ. 1920 Jan Łukasiewicz ได้คิดสิ่งที่เรียกว่า ลอจิก 3 ค่า (3-valued logic) ซึ่งเป็นลอจิกที่ เบี่ยงเบนมาจากลอจิก 2 ค่า และหลังจากนั้นได้ถูกขยายไปเป็นค่าที่เป็นอะไรก็ได้ (arbitrary number of value) จากนั้น Max Black ได้คิดสิ่งเป็นกึ่งฟัซซีเซต (quasi-fuzzy set) ที่ระดับของ ความเป็นสมาชิกในเซตจะถูกให้ค่ากับสมาชิกของเซต ในขณะที่ Lofti A. Zadeh ได้คิดค้นสิ่งที่ เรียกว่า ทฤษฎีฟัซซีเซต ในปี ค.ศ. 1965 และตั้งแต่นั้นจนกระทั่งถึงศตวรรษที่ 1980 ฟัซซีเซต หรือ ระบบฟัซซี ก็กลายเป็นงานวิจัยที่เป็นที่นิยม แต่ในศตวรรษที่ 1980 ฟัซซีเซตก็เข้าสู่ยุคมืดเช่นกัน จนกระทั่งนักวิจัยชาวญี่ปุ่น ได้ประยุกต์ใช้ฟัซซีเซตในการควบคุมสิ่งของหลายชนิดเช่น เครื่องดูดฝุ่น กล้องถ่ายรูป ลิฟต์ จนกระทั่งหุ่นยนต์ ในช่วงปลายศตวรรษที่ 1980 ซึ่งนักวิจัยชาวญี่ปุ่นที่ถือว่า เป็นผู้บุกเบิกคือ Michio Sugeno [Eberhart07] และหลังจากนั้นทฤษฎีฟัซซีจึงกลับมาเป็นที่นิยม อีกครั้ง ทั้งนี้นักวิจัยด้านนี้ที่มีชื่อเสียงเป็นที่รู้จัก มีด้วยกันหลายท่านอาทิเช่น Ebrahim H.Mamdani, Tomohiro Takagi, James Bezdek, James M. Keller และนักวิจัยท่านอื่นที่ไม่ได้ กล่าวถึง ณ. ที่นี้

สำหรับงานวิจัยทางด้านการคำนวณเชิงวิวัฒนาการ (Evolutionary Computing) นั้นเริ่ม มาจากงานวิจัยในด้านอัลกอริทึมแบบพันธุกรรม (genetic algorithm) ในปีค.ศ. 1950 ซึ่งเป็น งานวิจัยของ Alex S. Fraser แต่อย่างไรก็ตามบุคคลที่ถือว่าเป็นบิดาของการคำนวณเชิง วิวัฒนาการโดยเฉพาะอย่างยิ่งอัลกอริทึมแบบพันธุกรรม คือ John Holland ซึ่งในงานวิจัยนี้ เป็น การจำลองวิวัฒนาการจากทฤษฎีของ Darwin (Darwin's theory of evolution) ในรูปของ อัลกอริทึม หลังจากนั้นงานวิจัยด้านนี้ ถูกพัฒนาจนเป็นที่นิยมในปัจจุบัน ส่วนนักวิจัยที่เป็นที่รู้จัก ในด้านนี้มีด้วยกันหลายท่าน อาทิเช่น Kenneth A. De Jong, J. David Schaffer, David E. Goldberg, David B. Fogel, John R. Koza และนักวิจัยท่านอื่นที่ไม่ได้กล่าวถึง ณ. ที่นี้

ส่วนงานวิจัยที่จะกล่าวถึงอันดับสุดท้ายคืองานวิจัยด้านความฉลาดเชิงกลุ่ม (Swarm Intelligence) เป็นงานวิจัยที่ตั้งต้นมาจากการศึกษาพฤติกรรมเชิงกลุ่มของลิง และมด ซึ่งเป็นงานวิจัยของนักวิทยาศาสตร์ชาวแอฟริกาใต้ ที่ชื่อว่า Eugene N. Marais (1871 – 1936) ซึ่งหนังสือของท่านนี้ถูกพิมพ์ หลังจากที่ท่านได้เสียชีวิตไปแล้ว 30 ปี ซึ่งหนังสือทั้งสองเล่มชื่อ The Soul of White Ant ถูกพิมพ์ในปีค.ศ. 1970 และหนังสืออีกเล่มชื่อ The Soul of the Ape ที่ถูกพิมพ์ในปีค.ศ. 1969 ส่วนอัลกอริทึมที่เป็นการจำลองสวอร์ม (modelling of swarm) นี้ เริ่มมีการศึกษาอย่างจริงจังในต้นศตวรรษ 1990 โดยเริ่มจากงานวิจัยของ Marco Dorigo ที่ทำการจำลองคอโลนีของมด (ant colonies) และในปีค.ศ. 1996 Russell C. Eberhart และ James Kennedy ได้พัฒนาอัลกอริทึมการหาค่าที่เหมาะสมแบบสวอร์มของอนุภาค (particle swarm optimization) โดยเลียนแบบฝูงนก (bird flock) และตอนนี้นงานวิจัยด้านนี้เป็นงานวิจัยที่ได้รับความสนใจอย่างกว้างขวาง เช่นกัน

## 1.2 ชนิดของความฉลาดเชิงคำนวณ (Computational Intelligence Paradigms)

เอกสารคำสอนนี้จะกล่าวถึงความฉลาดเชิงคำนวณ (Computational Intelligence (CI) 4 ชนิดหลักๆ คือ โครงข่ายประสาทเทียม (Artificial Neural Networks) ระบบฟัซซี (Fuzzy Systems) การคำนวณเชิงวิวัฒนาการ (Evolutionary Computing) และความฉลาดเชิงกลุ่ม (Swarm Intelligence) ทั้งนี้ในแก้ปัญหา อาจจะใช้วิธีการในแก้ปัญหาจากหนึ่งใน 4 ชนิด หรือเป็นส่วนผสมของทั้ง 4 ชนิด และในบางครั้งอาจจะมีการนำวิธีการในความน่าจะเป็น (probabilistic methods) มาใช้ร่วมด้วยก็ได้ ดังแสดงในรูปที่ 1.1 [Engelbrecht07] และในหัวข้อนี้จะกล่าวอย่างย่อของทั้ง 4 ชนิด เพื่อให้ผู้อ่านมีความเข้าใจเบื้องต้นก่อนที่จะกล่าวถึงรายละเอียดในบทต่อไป



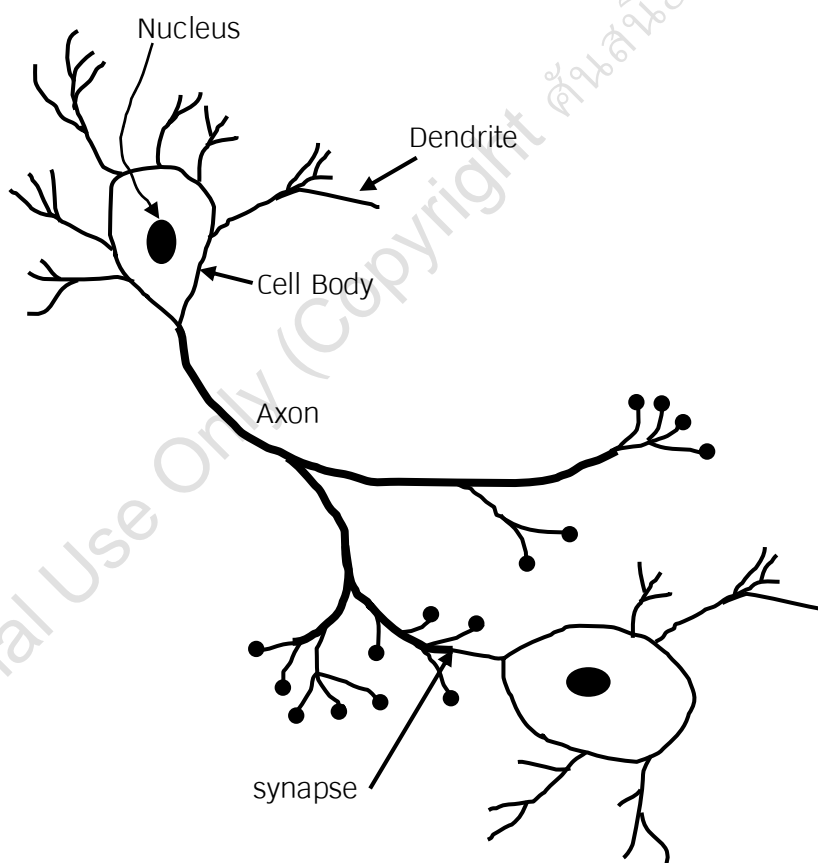
รูปที่ 1.1 ความเชื่อมโยงของชนิดของความฉลาดเชิงคำนวณ

ความฉลาดเชิงคำนวณทั้ง 4 ชนิดมีต้นกำเนิดมาจากระบบชีวภาพ (biological system) นั่นคือโครงข่ายประสาทเทียม มาจากการพยายามสร้างแบบจำลองของระบบประสาทชีวภาพ

(biological neural system) ในขณะที่ระบบพีซี มาจากการศึกษาของการโต้ตอบ (interact) ของสิ่งมีชีวิต (organism) กับสิ่งแวดล้อม ส่วนการคำนวณเชิงวิวัฒนาการ มีต้นกำเนิดจากการวิวัฒนาการในธรรมชาติ ทั้งนี้รวมถึงพันธุกรรม (genetic) และวิวัฒนาการทางพฤติกรรม (behavioral evolution) และสุดท้ายความฉลาดเชิงกลุ่ม มีต้นกำเนิดจากการจำลองพฤติกรรมทางสังคมของสิ่งมีชีวิตในการอยู่ร่วมกันในกลุ่มหรือสวอร์ม (swarm) หรือโคโลนี (colony)

### 1.2.1 โครงข่ายประสาทเทียม (Artificial Neural Networks)

โครงข่ายประสาทเทียมถูกสร้างขึ้นมาเพื่อเลียนแบบการทำงานของสมองของมนุษย์ ซึ่งสมองของมนุษย์มีลักษณะการทำงานที่เป็นคอมพิวเตอร์งานขนานที่ไม่เป็นเชิงเส้น และมีความซับซ้อน (complex, nonlinear and parallel computer) หรือที่เรียกว่าเป็นระบบการประมวลผลข้อมูล [Haykin09] ในการทำงานแบบนี้ สมองมีความสามารถในการทำให้เซลล์ประสาท (neuron) ทำงานด้วยกันแบบมีโครงสร้าง เพื่อให้ทำการคำนวณสำหรับการรู้จำ การรับรู้ และอื่นๆ แต่จนถึงปัจจุบันการทำงานของสมองยังคงมีความเร็วว่าการคำนวณของเครื่องคอมพิวเตอร์หลายเท่า และโดยปกติสมองของมนุษย์มีการสั่งสมประสบการณ์จากสิ่งแวดล้อมหรือมีการเรียนรู้จากสิ่งแวดล้อม และมีการปรับตัวไปตามสิ่งแวดล้อม เพื่อนำมาใช้ในภายหลัง

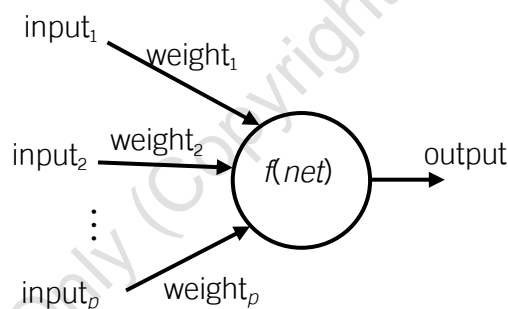


รูปที่ 1.2 เซลล์ประสาททางชีววิทยา (biological neuron)

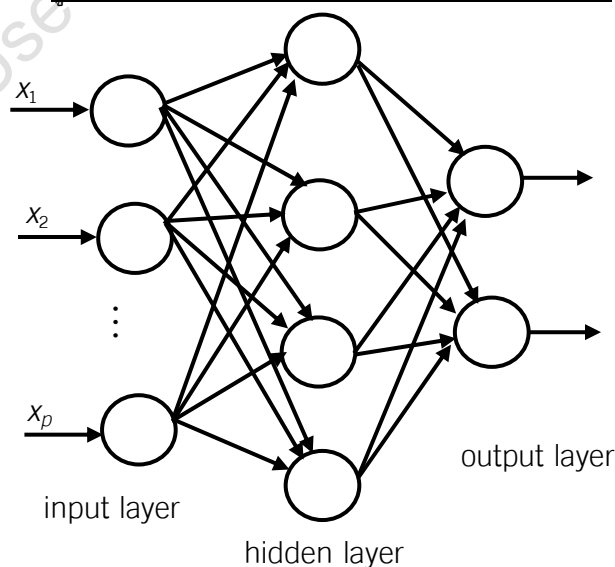
สมองของมนุษย์ประกอบด้วยเซลล์ประสาท (neuron) ที่มีจำนวนอยู่ในช่วงของ 10 – 500 พันล้าน และ 50 ล้านล้านจุดประสานประสาท (synapse) ในเปลือกสมอง ซึ่งเซลล์ประสาทเหล่านี้ถูกจัดเรียงได้ประมาณ 1000 มอดูล (module) และในแต่ละมอดูลมี โครงข่ายประสาท (neural network) ประมาณ 500 โครงข่าย และโครงข่ายเหล่านี้สามารถทำงาน และแก้ปัญหาได้พร้อมกัน

เนื่องจากคุณสมบัติของการกระจายแบบขนาน นั่นเอง ส่วนองค์ประกอบของเซลล์ประกอบแสดงในรูปที่ 1.2 นั้นประกอบไปด้วย ตัวเซลล์ (cell body) โยประสาทนำเข้า (dendrite) และโยประสาทนำออก (axon) ซึ่งการเชื่อมต่อระหว่างเซลล์ประสาทเป็นการเชื่อมต่อระหว่างโยประสาทนำออกของเซลล์หนึ่งไปยังโยประสาทนำเข้าของอีกเซลล์หนึ่ง และเส้นเชื่อมเหล่านี้ถูกเรียกว่าจุดประสานประสาท (synapse) ดังนั้นการส่งต่อสัญญาณจะเป็นการส่งต่อสัญญาณจากโยประสาทนำเข้า ผ่านตัวเซลล์ และโยประสาทนำออก ไปยังเซลล์อื่น นั่นเอง [Engelbrecht07] แต่การส่งต่อสัญญาณจะกระทำก็ต่อเมื่อเซลล์นั้นถูกกระตุ้นเท่านั้น ดังนั้นเซลล์ประสาทสามารถยับยั้ง (inhibit) หรือ กระตุ้น (excite) สัญญาณได้ ยกตัวอย่างเช่นการที่มีเข็มปักที่นิ้วมือ ก็จะมีการส่งสัญญาณจากปลายนิ้วมือ และส่งผ่านเซลล์ ในลักษณะเดียวกันจนกระทั่งสัญญาณนั้นไปถูกแปลผลที่สมอง ทำให้มนุษย์รู้สึกได้ถึงเจ็บที่ปลายนิ้วมือนั้น

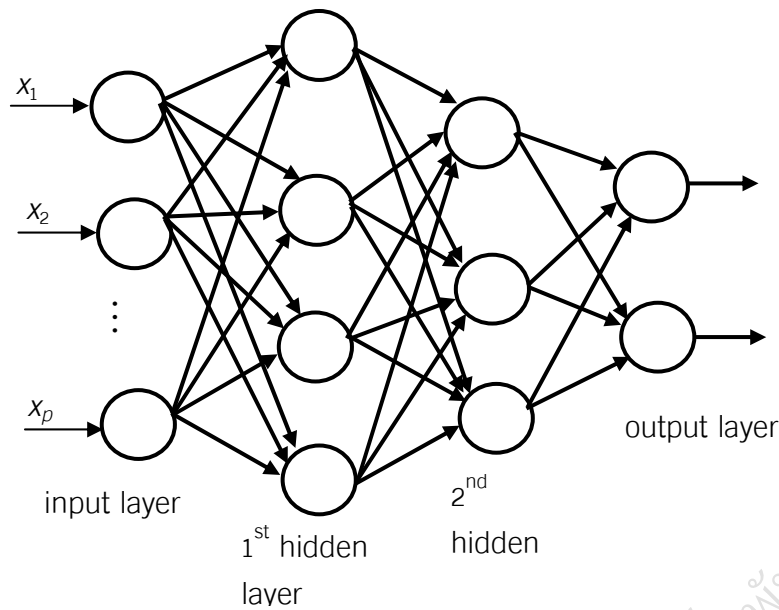
ส่วนโครงข่ายประสาทเทียม (artificial neuron network) เป็นแบบจำลองของเซลล์ประสาททางชีววิทยา (biological neuron) โดยที่แต่ละจุด (node) ได้รับสัญญาณจากสิ่งแวดล้อมหรือ จุดอื่น และทำการรวมสัญญาณเหล่านั้น และเมื่อจุดนั้นมีการทำงาน จะส่งต่อสัญญาณไปยังจุดอื่นที่เชื่อมต่อกับตัวอย่างของเซลล์เทียมแสดงในรูปที่ 1.3 ซึ่งสัญญาณอินพุตที่เข้ามาที่เซลล์นี้จะถูกยับยั้ง หรือกระตุ้น ด้วยค่าน้ำหนักของแต่ละเส้นเชื่อม ที่เป็นลบ หรือ บวกตามลำดับ และเมื่อสัญญาณอินพุตเหล่านี้ถูกรวมกันแล้ว นั้นค่าที่ได้จะเป็นอินพุตให้กับฟังก์ชันที่ถูกเรียกว่าฟังก์ชันการกระตุ้น (activation function) และค่าเอาต์พุตที่ได้จะเป็นค่าเอาต์พุตที่ออกจากเซลล์นั้น



รูปที่ 1.3 เซลล์ประสาทเทียม (artificial neuron)



(ก)



(ข)

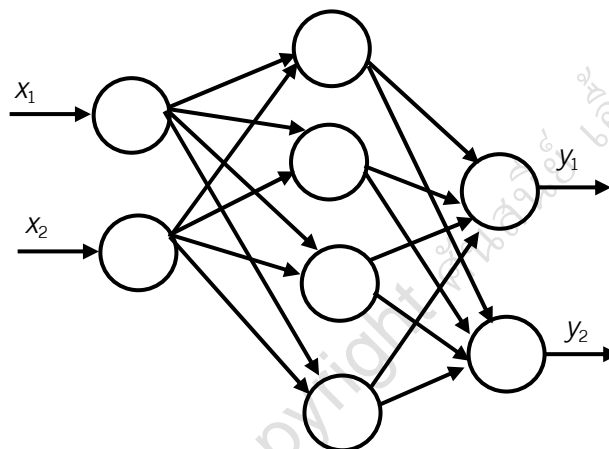
รูปที่ 1.4 โครงข่ายประสาทเทียม (artificial neural networks) (ก)  $p-4-2$  และ (ข)  $p-4-3-2$

ดังนั้นโครงข่ายประสาทเทียมแสดงในรูปที่ 1.4 จึงเป็นการเชื่อมต่อของเซลล์เทียมในลักษณะที่ ชั้นอินพุต (input layer) จะเป็นชั้นที่รับค่าอินพุตตั้งต้น สมมุติให้อินพุตเวกเตอร์ ( $x$ ) เป็นเวกเตอร์ที่มี  $p$  มิติ ( $p$ -dimensional vector) ดังนั้นที่ชั้นนี้จะมีจำนวนจุด (node) เท่ากับจำนวนมิติของอินพุตเวกเตอร์เสมอ และชั้นอินพุตนี้โดยปกติจะเชื่อมต่อกับชั้นซ่อน (hidden layer) ซึ่งการเชื่อมต่อนี้อาจจะเป็นการเชื่อมต่อแบบเต็มรูปแบบ (fully connected) หรือเป็นบางส่วน (partial connected) ก็ได้ และในขณะเดียวกันจำนวนชั้นซ่อนอาจจะมีมากกว่า 1 ก็ได้ โดยที่ในแต่ละชั้นซ่อนจะมีจำนวนจุด (node) เท่าใดก็ได้ ซึ่งไม่จำเป็นต้องเท่ากัน ส่วนชั้นสุดท้ายคือชั้นเอาต์พุต (output layer) ซึ่งในชั้นนี้จำนวนจุดเอาต์พุต (output node) จะเป็นไปตามการประยุกต์ใช้งานซึ่งจะกล่าวในรายละเอียดในบทต่อไป โครงข่ายประสาทเทียมนี้มีการส่งต่อสัญญาณในลักษณะที่คล้ายกันกับเซลล์ประสาททางชีววิทยา ดังนั้นโครงข่ายประสาทเทียมนี้จึงเปรียบเสมือนตัวประมวลผลกระจายแบบขนานจำนวนมาก ที่ประกอบด้วยหน่วยการประมวลผลง่ายๆ หลายหน่วย เพื่อทำการเก็บความรู้จากประสบการณ์ และสามารถนำมาใช้ได้ภายหลัง ซึ่งกระบวนการนี้เลียนแบบสมองของมนุษย์ใน 2 มุมมองคือ [Haykin09]

1. ความรู้ที่ได้โดยโครงข่ายเกิดจากสภาวะแวดล้อม โดยผ่านกระบวนการเรียนรู้
2. ความแข็งแรงของเส้นเชื่อมระหว่างเซลล์ประสาท หรือที่เรียกว่าน้ำหนักของจุดประสานประสาท (synaptic weight) ถูกใช้ในการเก็บความรู้ที่ได้

โดยปกติแล้วชื่อของโครงข่ายประสาทเทียมนี้จะขึ้นอยู่กับจำนวนจุดของแต่ละชั้น ดังนั้นชื่อของโครงข่ายในรูปที่ 1.4(ก) คือ  $p-4-2$  ( $p$  คือจำนวนจุดที่ชั้นอินพุต 4 คือจำนวนจุดที่ชั้นซ่อน และ 2 คือจำนวนจุดที่ชั้นเอาต์พุต) แต่ถ้าชื่อของโครงข่ายเป็น  $p-4-3-2$  หมายถึงโครงข่ายนี้มี  $p$  จุดที่ชั้นอินพุต 4 จุดที่ชั้นซ่อนชั้นแรก และ 3 จุดที่ชั้นซ่อนที่สอง ในขณะที่ มี 2 จุดที่ชั้นเอาต์พุต ดังแสดงในรูปที่ 1.4(ข) นั่นเอง

**ตัวอย่างที่ 1.1** ต้องการสร้างโครงข่ายประสาทเทียมในการจำแนก (classification) ผู้หญิงออกจากผู้ชาย โดยปกติจะมีการเก็บลักษณะ (feature) เช่น ส่วนสูง ( $x_1$ ) น้ำหนัก ( $x_2$ ) จากตัวอย่างที่เป็น ผู้หญิง และ ผู้ชาย สมมติให้ตัวอย่างที่เป็นผู้ชายที่ 100 คน และ ผู้หญิง 100 คน ดังนั้น เวกเตอร์ลักษณะ (feature vector ( $x$ )) มีทั้งหมด 200 เวกเตอร์ โดยที่เวกเตอร์ลักษณะนี้เป็นเวกเตอร์ที่มี 2 มิติ คือ  $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$  ส่วนเอาต์พุตที่ต้องการ (desire output) มี 2 ค่า ( $y_1, y_2$ ) โดยที่ถ้าค่าเอาต์พุตเป็น (1,0) แสดงว่าเวกเตอร์ลักษณะนั้นมาจากคลาส (class) 1 หรือเป็นผู้หญิง และถ้าค่าเอาต์พุตเป็น (0,1) แสดงว่าเวกเตอร์ลักษณะนั้นมาจากคลาส (class) 2 หรือเป็นผู้ชาย และเมื่อนำข้อมูลชุดนี้ไปทำการสร้างโครงข่ายโดยผ่านการเรียนรู้ และได้โครงข่ายที่ดีที่สุดเป็นโครงข่าย 2-4-2 ดังรูปที่ 1.5



รูปที่ 1.5 โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 1.1

ดังนั้นเมื่อมีตัวอย่างใหม่ที่ต้องการทดสอบ สามารถทำได้โดยส่งค่าเวกเตอร์ทดสอบ  $x$  ผ่านโครงข่าย และถ้าค่า  $y_1$  มากกว่า  $y_2$  แสดงว่าเวกเตอร์ทดสอบนั้นเป็นผู้หญิง แต่ในทางกลับกันถ้าค่า  $y_2$  มากกว่า  $y_1$  แสดงว่าเวกเตอร์ทดสอบนั้นเป็นผู้ชายเป็นต้น

การประยุกต์ใช้งานของโครงข่ายประสาทเทียมมีได้หลายประเภท ยกตัวอย่างเช่น การวินิจฉัยโรค การรู้จำรูปแบบ ไม่ว่าจะเป็นตัวอักษร สัญญาณ หรือ วัตถุ รวมทั้งใช้ในระบบควบคุมและอื่นๆ อีกมากมาย

### 1.2.2 ระบบฟัซซี (Fuzzy Systems)

การหาเหตุผลของมนุษย์ ไม่จำเป็นต้องเป็นการเลือกกระหว่าง ใช่ หรือ ไม่ใช่ เท่านั้น เช่นคำกล่าวที่ว่า “นกทุกชนิดบินได้” แต่ถ้าพิจารณาจริงๆ มีนกบางชนิดที่บินไม่ได้ เช่นนกเพนกวิน เป็นต้น ดังนั้นคำกล่าวนี้ไม่ถือว่าเป็นจริงทั้งหมด หรือในการเป็นสมาชิกของอะไรบางอย่างก็ไม่จำเป็นต้องมีให้เลือกแค่ เป็น หรือไม่เป็นเท่านั้น เช่นคำว่า “เพื่อน” ถ้าพิจารณาจริงแล้ว มนุษย์จะมีระดับของความ เป็นเพื่อนอยู่ นั่นคือเรื่องบางเรื่องคนเราจะเล่าให้เพื่อนบางคนฟังเท่านั้น เป็นต้น ซึ่งการเลือกในลักษณะเหล่านี้จะเป็นการเลือกที่เกิดขึ้นสำหรับลอจิกที่มีค่าเป็นไบนารี (binary-valued logic) แต่สำหรับฟัซซีเซต และฟัซซีลอจิก ไม่เป็นเช่นนั้น แต่จะมีกระบวนการที่เรียกว่า

การหาเหตุผลโดยประมาณ (approximate reasoning) ที่มีความใกล้เคียงกับกระบวนการคิดและหาเหตุผลของมนุษย์มากที่สุด

ระบบฟัซซีถูกนำไปใช้ในระบบควบคุม ระบบการตัดสินใจต่างๆ การวินิจฉัยความผิดปกติ (fault diagnosis) การจัดกลุ่ม (clustering) และการรู้จำวัตถุ เป็นต้น ส่วนรายละเอียดของระบบฟัซซี จะกล่าวถึงอย่างละเอียดในบทต่อไป

### 1.2.3 การคำนวณเชิงวิวัฒนาการ (Evolutionary Computing)

การคำนวณเชิงวิวัฒนาการ เป็นกระบวนการที่สอดคล้องกับการวิวัฒนาการในธรรมชาติ นั่นคือ เป็นกระบวนการเลือกผู้ที่แข็งแรงในมุมมองของความเหมาะสมที่สุด (fittest) ให้อยู่รอด ในขณะที่กำจัดผู้ที่อ่อนแอ ซึ่งกระบวนการอยู่รอด (survival) จะเป็นกระบวนการที่ผ่านการสืบพันธุ์ (reproduction) โดยที่ลูกหลาน (offspring) ที่อาจจะเกิดจากแม่ (parents) 2 ตัวหรือมากกว่า 2 จะมียีน (gene) ของแม่เหล่านั้น ซึ่งตามปกติจะคาดหวังว่าจะเป็นยีนที่ดีที่สุด ส่วนลูกหลานที่มีลักษณะเฉพาะ (characteristic) ที่ไม่ดีจะกลายเป็นตัวที่อ่อนแอ และถูกกำจัดไปในที่สุด [Engelbrecht07]

อัลกอริทึมเชิงวิวัฒนาการ (evolutionary algorithm) ใช้ประชากรของรายบุคคล (population of individuals) โดยที่รายบุคคลในที่นี้คือ โครโมโซม (chromosome) ซึ่งจะเป็นตัวกำหนดลักษณะเฉพาะ และลักษณะเฉพาะในที่นี้คือยีน (gene) นั่นเอง โดยปกติค่าของยีนจะถูกเรียกว่า แอลลีล (allele) โดยปกติในแต่ละรุ่น (generation) แต่ละรายบุคคลจะแข่งขันกันสร้างลูกหลาน รายบุคคลที่สามารถอยู่รอดได้ดีกว่าจะมีโอกาสมากกว่านั่นเอง ซึ่งกระบวนการในการสร้างลูกหลานหรือที่เรียกว่าคrossover คือการรวมบางส่วนของแม่ (parents) และในขณะเดียวกันแต่ละรายบุคคลจะผ่านกระบวนการกลายพันธุ์ (mutation) ซึ่งเป็นการเปลี่ยนแปลงค่าแอลลีลของยีนนั่นเอง ส่วนค่าความแข็งแรงของการอยู่รอดจะวัดได้จากฟังก์ชันความเหมาะสม (fitness function) ที่เป็นจุดประสงค์ (objective) และข้อกำหนดของปัญหาที่ต้องการแก้ หลังจากนั้นแต่ละรายบุคคลจะถูกเลือกให้รอด (survive) ไปยังรุ่นต่อไป (หรือที่เรียกว่าอีลิทิสซึม (elitism)) หรือกำจัด (culling) โดยที่ไม่ได้ถูกส่งต่อไปยังรุ่นต่อไป [Engelbrecht07] โดยปกติแล้วอัลกอริทึมเชิงวิวัฒนาการ ถูกแยกออกได้เป็น [Engelbrecht07]

1. อัลกอริทึมแบบพันธุกรรม (genetic algorithm) ซึ่งเป็นการจำลองวิวัฒนาการของยีน
2. โปรแกรมแบบพันธุกรรม (genetic programming) เป็นสิ่งที่มีพื้นฐานจากอัลกอริทึมแบบพันธุกรรม แต่รายบุคคลในกรณีนี้เป็นโปรแกรม
3. โปรแกรมแบบวิวัฒนาการ (evolutionary programming) เป็นสิ่งที่มาจากการจำลอง (simulation) ของพฤติกรรมที่ปรับตัวได้ในการวิวัฒนาการ (เป็นวิวัฒนาการของฟีโนไทป์ฟิค (phenotypic evolution))
4. กลยุทธ์แบบวิวัฒนาการ (evolution strategies) เป็นสิ่งที่พยายามจำลองตัวแปรของกลยุทธ์ที่ควบคุมการแปรผันในวิวัฒนาการ
5. วิวัฒนาการเชิงอนุพันธ์ (differential evolution) เป็นสิ่งที่คล้ายกับอัลกอริทึมแบบพันธุกรรม แต่กลไกในการสืบพันธุ์ ต่างกัน
6. วิวัฒนาการเชิงวัฒนธรรม (cultural evolution) เป็นการจำลองวิวัฒนาการของวัฒนธรรมของประชากร และวัฒนธรรมมีอิทธิพลต่อวิวัฒนาการของพันธุกรรม และวิวัฒนาการของฟีโนไทป์ฟิค ของรายบุคคลอย่างไร



7. วิวัฒนาการร่วม (coevolution) เป็นกระบวนการที่รายบุคคลที่มีวิวัฒนาการผ่านการร่วมมือ (coopration) หรือการแข่งขัน (competition) กับรายบุคคลอื่น ทำให้ได้ลักษณะเฉพาะที่จำเป็น ในการอยู่รอด

การคำนวณเชิงวิวัฒนาการถูกนำไปใช้ในหลายการประยุกต์ใช้ตัวอย่างเช่น การทำเหมืองข้อมูล (data mining) การหาค่าที่เหมาะสม (optimization) การวินิจฉัยความผิดปกติ (fault diagnosis) การจัดกลุ่ม (clustering) และการประมาณอนุกรมเวลา (time-series approximation) เป็นต้น และในเอกสารคำสอนเล่มนี้จะกล่าวถึงเพียงแค่อัลกอริทึมแบบพันธุกรรม โปรแกรมแบบพันธุกรรม และโปรแกรมแบบวิวัฒนาการ ในรายละเอียดภายหลังในบทต่อไปเท่านั้น

#### 1.2.4 ความฉลาดเชิงกลุ่ม (Swarm Intelligence)

ความฉลาดเชิงกลุ่มมีกำเนิดมาจากการศึกษาของโคโลนี (colony) หรือสวอร์ม (swarm) ของสิ่งมีชีวิตเชิงสังคม (social organism) ซึ่งการศึกษาพฤติกรรมทางสังคมของสิ่งมีชีวิต หรือรายบุคคลในสวอร์มนี้นำให้เกิดอัลกอริทึมในการหาค่าที่เหมาะสม และอัลกอริทึมในการจัดกลุ่มที่มีประสิทธิภาพ

ในเอกสารคำสอนนี้จะกล่าวในบทต่อไปถึง การหาค่าที่เหมาะสมจากสวอร์มของอนุภาค (particle swarm optimization (PSO)) [Engelbrecht07] ซึ่งเป็นกระบวนการที่เลียนแบบพฤติกรรมทางสังคมของการบินของนก (bird flock) โดยที่เป็นกระบวนการที่ใช้ประชากรเช่นกัน และในกรณีนี้รายบุคคลถูกเรียกว่าอนุภาค (particle) จะจับกันเป็นกลุ่มในสวอร์ม ซึ่งแต่ละอนุภาคในสวอร์มเป็นคำตอบที่ต้องการหาตัวเอง และเนื่องจากวิธีการนี้เป็นการเลียนแบบการบินของนก แต่ละอนุภาคจะทำการ “บิน” ไปในปริภูมิของการหา (search space) ที่มีหลายมิติ และจะมีการปรับตำแหน่งของตัวเองตามประสบการณ์ของตัวเอง และของเพื่อนรอบข้างด้วย ดังนั้นจะเป็นการปรับเข้าหาตำแหน่งที่ดีที่สุดของตัวเองและของเพื่อนรอบข้าง เพื่อให้เข้าใกล้คำตอบที่เหมาะสมที่สุดนั่นเอง ซึ่งกระบวนการนี้ถูกนำไปใช้ในหลายการประยุกต์ใช้เช่น การประมาณฟังก์ชัน (function approximation) การจัดกลุ่ม (clustering) และการหาคำตอบของระบบสมการ (solving systems of equations) เป็นต้น

นอกเหนือจากนี้ยังกล่าวถึงโคโลนีของมด (ant colony) [Engelbrecht07] ที่เป็นการจำลองการตกสะสม (deposit) ของฟีโรโมน (pheromone) ของมดในการหาเส้นทางที่สั้นที่สุดใน การหาอาหาร ซึ่งกระบวนการนี้จะให้เส้นทางที่สั้นในอัลกอริทึมการหาค่าที่เหมาะสมที่สุด ซึ่งถูกนำไปประยุกต์ใช้ในการจัดเส้นทางที่เหมาะสมที่สุด (routing optimization) ในโครงข่ายของโทรคมนาคม (telecommunication network) และการจัดกำหนดการ (scheduling) เป็นต้น ทั้งนี้ในตอนท้ายจะกล่าวถึงโคโลนีของผึ้ง (bee colony) ซึ่งเป็นการจำลองพฤติกรรมของผึ้ง (honey bee) โดยที่มีผึ้งอยู่ 3 ชนิดคือ ผึ้งลูกจ้าง (employed bee) ผึ้งชม (onlooker bee) และ ผึ้งสอดแนม (scout bee) ในการหาแหล่งอาหาร [Karaboga05] นั่นเอง ซึ่งกระบวนการนี้ถูกนำไปใช้ในการแก้ปัญหาการหาค่าที่เหมาะสมสำหรับฐานนิยมเดียว (unimodal optimization) และหลายฐานนิยม (multi-modal optimization)

---

### คำถามท้ายบทที่ 1

1.1 ให้อ่านผลงานทางวิชาการใดก็ได้ที่นำความฉลาดเชิงคำนวณ ไปประยุกต์ และทำการวิเคราะห์ว่ากระบวนการเหล่านั้น มีการปรับตัวได้ และเป็นไปตามนิยามของความฉลาดเชิงคำนวณหรือไม่อย่างไร

1.2 จากนิยามของความฉลาดเชิงคำนวณ จงวิเคราะห์ว่า โครงข่ายประสาทเทียม ระบบฟัซซี การคำนวณเชิงวิวัฒนาการ และความฉลาดเชิงกลุ่ม เป็นไปตามนิยามหรือไม่ อย่างไร

---

# บทนำโครงข่ายประสาทเทียม

## Introduction to Neural Networks

### บทที่ 2

บทนี้จะกล่าวถึงพื้นฐานโครงข่ายประสาทเทียม รวมทั้งโครงข่ายประสาทเทียมที่เป็นที่นิยมใช้ในการประยุกต์ใช้ต่างๆ มากที่สุด ซึ่งคือเปอร์เซปตรอนหลายชั้น (multi-layer perceptrons) และนอกเหนือจากนี้ยังกล่าวถึง การส่งของการจัดระเบียบลักษณะตนเอง (self-organizing feature maps) หรือการส่งการจัดระเบียบตนเอง (self-organizing maps)

### 2.1 พื้นฐานโครงข่ายประสาทเทียม (Basic Neural Networks)

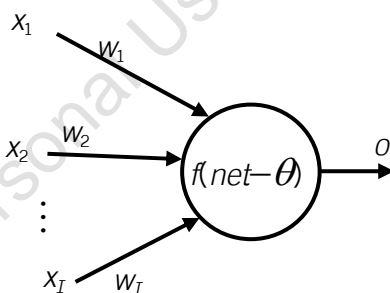
โครงข่ายประสาทเทียม (Neural Networks) [Engelbrecht07] เปรียบเสมือนฟังก์ชันที่ไม่เป็นเชิงเส้น (nonlinear function) ที่ส่งค่าจากเซตของจำนวนจริงใน  $I$  มิติ (อินพุตมี  $I$  มิติ) ไปยังเซตของจำนวนจริงใน  $K$  มิติ (เอาต์พุตมี  $K$  มิติ) นั่นคือ

$$f_{NN} : \mathcal{R}^I \rightarrow \mathcal{R}^K \quad (2.1)$$

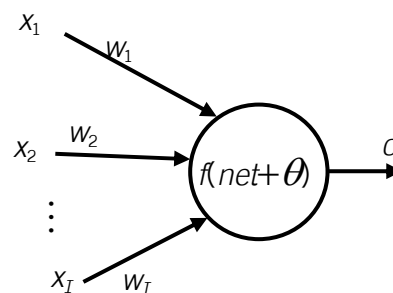
ซึ่งฟังก์ชัน  $f_{NN}$  เป็นฟังก์ชันซับซ้อนของเซตของฟังก์ชันที่ไม่เป็นเชิงเส้นของแต่ละเซลล์ประสาท (neuron) ในโครงข่าย และโดยปกติแล้วในการประยุกต์ใช้โครงข่ายประสาทเทียมในการจำแนกประเภท (classification) ฟังก์ชันนี้จะเป็นการส่งค่าจากเซตของจำนวนจริงใน  $I$  มิติ (อินพุตมี  $I$  มิติ) ไปยังค่าในช่วง  $[0,1]$  หรือ  $[-1,1]$  ทั้งนี้ขึ้นอยู่กับฟังก์ชันการกระตุ้น (activation function) นั่นคือ

$$f_{NN} : \mathcal{R}^I \rightarrow [0,1] \text{ หรือ } f_{NN} : \mathcal{R}^I \rightarrow [-1,1] \quad (2.2)$$

รูปที่ 2.1 แสดงถึงโครงข่ายที่มีเพียง 1 เซลล์ประสาทที่รับเวกเตอร์อินพุตที่มี  $I$  มิติ ( $\mathbf{x} = [x_1, x_2, \dots, x_I]^T$ ) และให้เอาต์พุตค่าเดียว ( $o$ ) ที่เป็นค่าเอาต์พุตจากฟังก์ชัน  $f$  ที่รับค่า  $net$  และ  $\theta$  เป็นอินพุตซึ่งค่า  $\theta$  คือค่าไบแอส (bias) ซึ่งโดยปกติแล้วค่าอินพุตของฟังก์ชันจะเป็น  $net - \theta$  หรือ  $net + \theta$  ก็ได้



(ก)



(ข)

รูปที่ 2.1 เซลล์ประสาทเทียมในกรณีที่อินพุตของฟังก์ชันเป็น (ก)  $net - \theta$  และ (ข)  $net + \theta$  ส่วนค่า  $net$  โดยปกติคือการนำค่าอินพุตแต่ละค่ามารวมกันโดยผ่านน้ำหนัก (weight) ที่เชื่อมต่อ ดังนั้นสามารถคำนวณได้ดังนี้ [Engelbrecht07]

$$net = \sum_{i=1}^I x_i w_i \quad (2.3)$$

แต่ในบางครั้งการหาค่า  $net$  สามารถหาได้จาก [Engelbrecht07]

$$net = \prod_{i=1}^I x_i^{w_i} \quad (2.4)$$

หลังจากค่าอินพุตและน้ำหนักถูกรวมกันโดยสมการที่ 2.3 หรือ 2.4 แล้ว เซลล์ประสาทนั้น จะทำการคำนวณหาค่าเอาต์พุตที่บ่งบอกถึงความแรงของการจุด (firing strength) ของเซลล์นั้น โดยผ่านค่า  $net$  และไบแอส ไปยังฟังก์ชันการกระตุ้น (activation function) ซึ่งฟังก์ชันการกระตุ้น จะเป็นฟังก์ชันเพิ่มทางเดียว (monotonically increasing function) (ไม่นับรวมฟังก์ชันเชิงเส้น (linear function)) ที่มีคุณสมบัติ ดังนี้

$$f(-\infty) = 0 \text{ หรือ } f(-\infty) = -1 \quad (2.5ก)$$

และ

$$f(\infty) = 1 \quad (2.5ข)$$

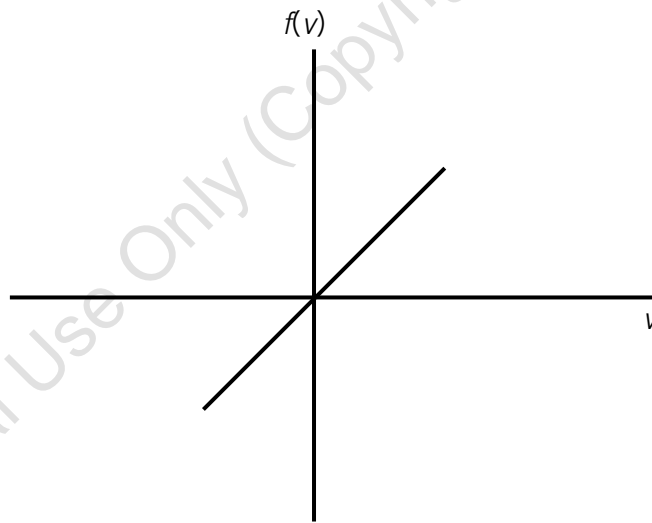
โดยปกติแล้วฟังก์ชันการกระตุ้นที่นิยมใช้มีดังต่อไปนี้

ฟังก์ชันเชิงเส้น (linear function) [Engelbrecht07]

เป็นฟังก์ชันที่มีลักษณะดังรูปที่ 2.2 และสามารถคำนวณได้ดังนี้

$$f(v) = \beta v \quad (2.6)$$

โดยที่  $\beta$  เป็นค่าคงที่ และเป็นค่าความชันของเส้นตรงนั้น ดังนั้นฟังก์ชันการกระตุ้นนี้จะให้ค่าเอาต์พุตที่เป็นเชิงเส้นเช่นกัน



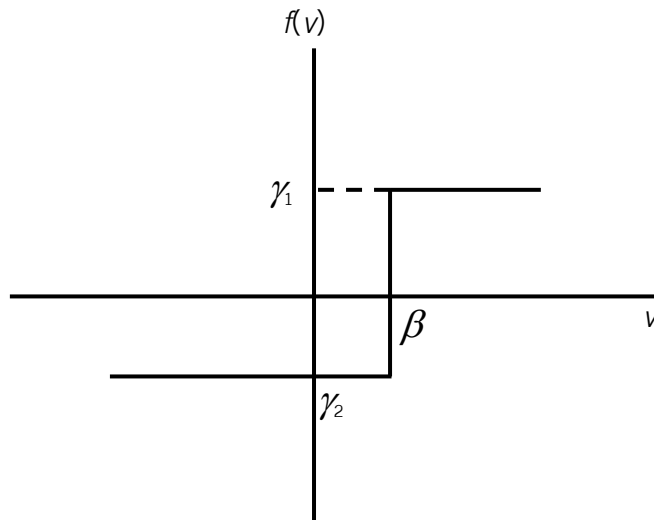
รูปที่ 2.2 ฟังก์ชันเชิงเส้น

ฟังก์ชันขั้น (step function) [Engelbrecht07]

บางครั้งถูกเรียกว่า ฟังก์ชันขั้นหนึ่งหน่วย (unit step function) เป็นฟังก์ชันที่ให้ค่าเอาต์พุต 2 ค่า โดยที่มีค่าขีดแบ่งเป็นตัวแบ่ง ( $\beta$ ) ดังรูปที่ 2.3 และสามารถคำนวณได้ดังนี้

$$f(v) = \begin{cases} \gamma_1 & \text{if } v \geq \beta \\ \gamma_2 & \text{if } v < \beta \end{cases} \quad (2.7)$$

ซึ่งโดยปกติแล้วค่า  $\gamma_1=1$  และ  $\gamma_2=0$  หรือ  $-1$

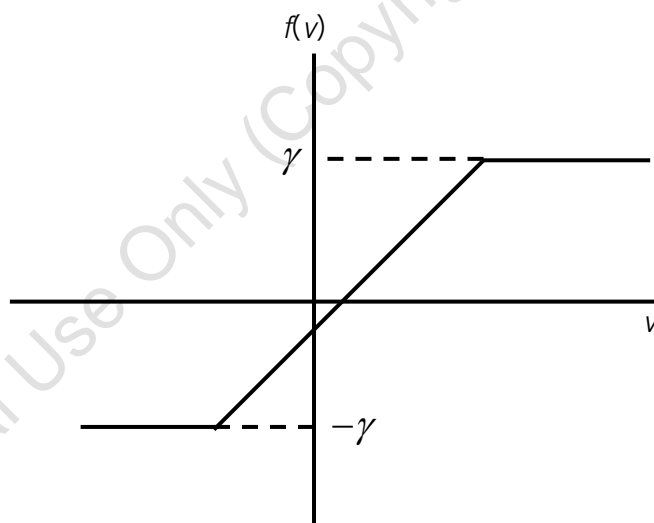


รูปที่ 2.3 ฟังก์ชันขั้น

ฟังก์ชันแรมป์ (ramp function) [Engelbrecht07]

เป็นฟังก์ชันที่เป็นส่วนผสมระหว่างฟังก์ชันเชิงเส้นและฟังก์ชันขั้น แสดงในรูปที่ 2.4 ซึ่งสามารถคำนวณได้ดังนี้

$$f(v) = \begin{cases} \gamma & \text{if } v \geq \beta \\ v & \text{if } -\beta < v < \beta \\ -\gamma & \text{if } v \leq -\beta \end{cases} \quad (2.8)$$



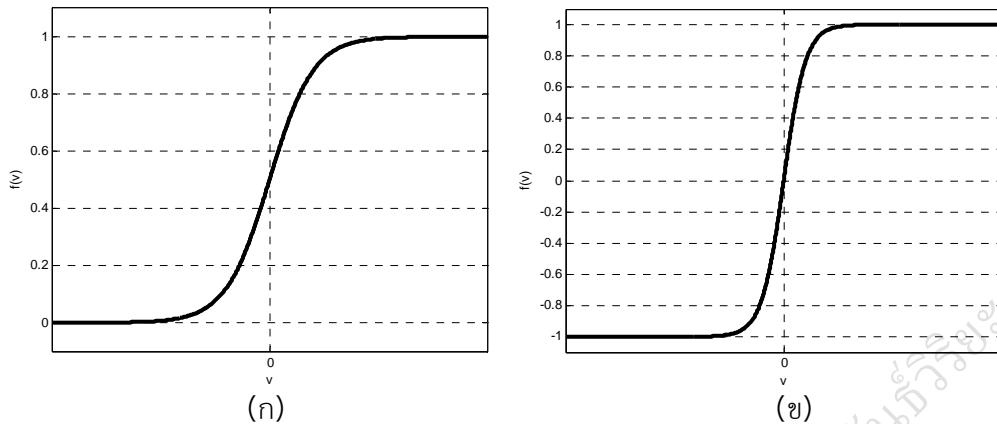
รูปที่ 2.4 ฟังก์ชันแรมป์

ฟังก์ชันซิกมอยด์ (sigmoid function) [Haykin09]

เป็นฟังก์ชันที่มีลักษณะเป็นรูปตัวอักษร S และเป็นฟังก์ชันประเภทที่เป็นฟังก์ชันการกระตุ้นที่เป็นที่นิยม ตัวอย่างของฟังก์ชันซิกมอยด์คือ ฟังก์ชันลอจิสติก (logistic function) แสดงในรูปที่ 2.5(ก) และคำนวณได้ดังสมการต่อไปนี้

$$f(v) = \frac{1}{1 + \exp(-av)} \quad (2.9)$$

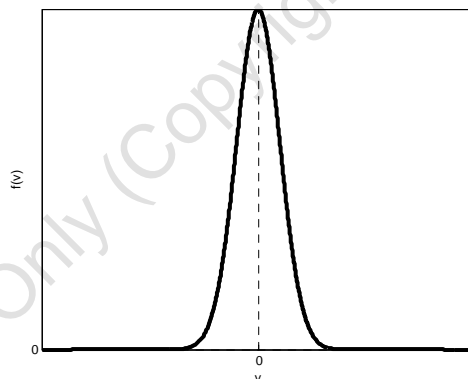
โดยที่ค่า  $a$  คือตัวแปรความชัน (slope parameter) ของฟังก์ชันซิกมอยด์ โดยปกติค่า  $a=1$  เป็นค่าที่นิยมใช้มากที่สุด ถ้าค่า  $a$  มีค่าเข้าใกล้อนันต์ (infinity) ฟังก์ชันนี้จะกลายเป็นฟังก์ชันขั้น และลักษณะฟังก์ชันนี้เป็นฟังก์ชันต่อเนื่องที่ให้ค่าตั้งแต่ค่าเข้าใกล้ 0 ถึงค่าเข้าใกล้ 1



รูปที่ 2.5 ฟังก์ชันซิกมอยด์ (ก) ฟังก์ชันลอจิสติก (ข) ฟังก์ชันไฮเพอร์โบลิคแทนเจนต์

แต่อย่างไรก็ตามในบางการประยุกต์ใช้งาน ต้องการค่าตั้งแต่ค่าที่เข้าใกล้  $-1$  ถึงค่าเข้าใกล้ 1 ดังนั้นฟังก์ชันซิกมอยด์อีกประเภทที่เป็นที่นิยมเช่นกันคือ ฟังก์ชันไฮเพอร์โบลิคแทนเจนต์ (Hyperbolic tangent function) ดังแสดงในรูปที่ 2.5(ข) และคำนวณได้ตามสมการต่อไปนี้

$$f(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)} = \frac{2}{1 + \exp(-v)} - 1 \quad (2.10)$$



รูปที่ 2.6 ฟังก์ชันเกาส์เซียน

ฟังก์ชันเกาส์เซียน (Gaussian function)

สามารถคำนวณได้ดังนี้

$$f(v) = \exp\left(-\frac{(v - \mu)^2}{\sigma^2}\right) \quad (2.11)$$

โดยที่  $\mu$  และ  $\sigma$  เป็นค่าเฉลี่ย (mean) และส่วนเบี่ยงเบนมาตรฐาน (standard deviation) ของฟังก์ชัน รูปที่ 2.6 แสดงถึงฟังก์ชันเกาส์เซียนที่มีค่าเฉลี่ยเป็น 0

ตัวอย่างที่ 2.1 สมมุติให้โครงข่ายประสาทเทียมมีลักษณะดังรูปที่ 2.7 เป็นโครงข่ายที่ถูกสร้างขึ้นมาเพื่อทำการแยกประเภท (classification) ของเอาต์พุตของลอจิกแอนด์ (AND logic) ดังตารางความจริง (truth table) ในตารางที่ 2.1 นั่นคือ เวกเตอร์อินพุต  $\mathbf{x} = [x_1, x_2]^T$  ที่มีค่า

เอาต์พุต  $y$  เป็น 0 จะอยู่ในคลาสเดียวกัน (สมมุติเป็นคลาส 1) ส่วน เวกเตอร์อินพุตที่มีค่าเอาต์พุตเป็น 1 จะอยู่อีกคลาสหนึ่ง (สมมุติเป็นคลาส 2) ในตัวอย่างนี้สมมุติให้ค่าไบแอส คือค่าน้ำหนักที่เชื่อมต่อกับค่าอินพุต 1 ส่วนค่าน้ำหนักอื่นๆ เป็นดังค่าที่กำกับในแต่ละเส้นเชื่อม ดังนั้นค่าอินพุตของฟังก์ชันการกระตุ้นตามสมการที่ 2.3 คือ

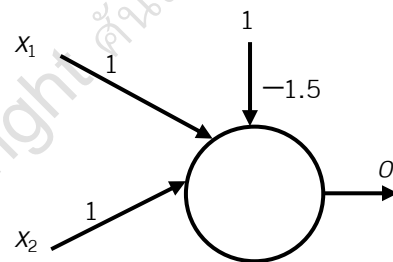
$$v = x_1 + x_2 - 1.5 \quad (2.12)$$

และในตัวอย่างนี้ฟังก์ชันการกระตุ้นเป็นฟังก์ชันขั้นที่มีค่า  $\gamma_1=1$  และ  $\gamma_2=0$  โดยที่  $\beta=0$  ดังนั้นที่เวกเตอร์อินพุต  $[0,0]^t$  จะได้ค่า  $v=-1.5$  ทำให้ค่า  $o=0$  ในขณะที่ถ้าเวกเตอร์อินพุต  $[0,1]^t$  หรือ  $[1,0]^t$  จะได้ค่า  $v=-0.5$  ทำให้ค่า  $o=0$  เช่นกัน แต่ถ้าเวกเตอร์อินพุต  $[1,1]^t$  จะได้ค่า  $v=0.5$  ทำให้ค่า  $o=1$  ซึ่งเป็นไปตามค่าเอาต์พุตที่ต้องการ (desire output) ทั้งหมด ดังแสดงในตารางที่ 2.2

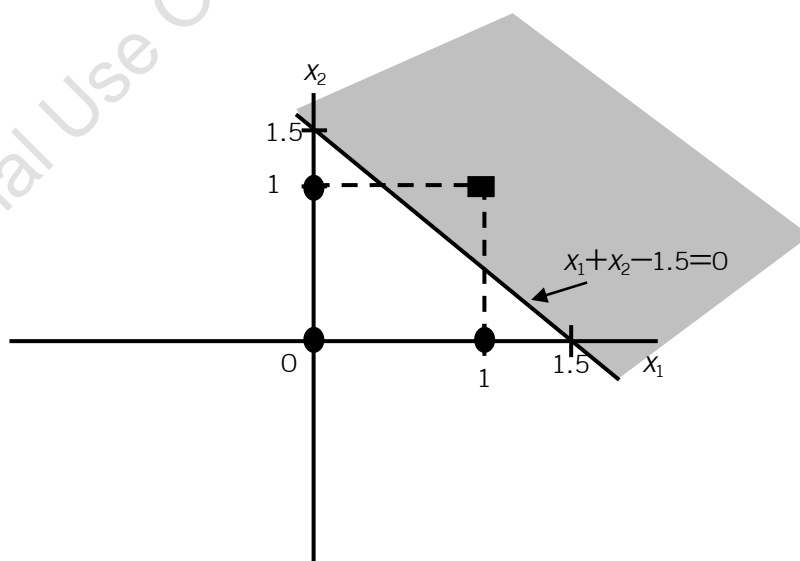
เส้นตรง  $x_1 + x_2 - 1.5 = 0$  แสดงในรูปที่ 2.9 เป็นเส้นตรงที่ได้จากสมการที่ 2.12 เป็นเส้นตรงที่ถูกเรียกว่าเส้นขอบการตัดสินใจ (decision boundary) เนื่องจากเวกเตอร์  $[x_1, x_2]^t$  ใดที่อยู่เหนือเส้นตรงนี้ (พื้นที่สีเทาในรูปที่ 2.9) จะถูกจัดเป็นคลาสที่ 2 เพราะค่า  $v$  ที่ได้จะมากกว่า 0 เสมอ ในขณะที่  $[x_1, x_2]^t$  ใดที่อยู่ใต้เส้นตรงนี้จะถูกจัดเป็นคลาสที่ 1 เพราะค่า  $v$  ที่ได้จะน้อยกว่า 0 เสมอ

ตารางที่ 2.1 ตารางความจริงของ AND logic

$x_1$	$x_2$	$y$
0	0	0
0	1	0
1	0	0
1	1	1



รูปที่ 2.7 โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 2.1



รูปที่ 2.8 เส้นขอบการตัดสินใจ

ตารางที่ 2.2 อินพุตและเอาต์พุตสำหรับตัวอย่างที่ 2.1

$x_1$	$x_2$	$v$	$o$
0	0	-1.5	0
0	1	-0.5	0
1	0	-0.5	0
1	1	0.5	1

เปอร์เซปตรอนในตัวอย่างที่ 2.1 เป็นลักษณะของ โครงข่ายป้อนค่าไปข้างหน้าชั้นเดียว (single-layer feedforward networks) ซึ่งเป็นการที่ชั้นอินพุตส่งค่าไปยังชั้นเอาต์พุตโดยตรง แต่จะไม่มีการส่งค่าย้อนกลับ

## 2.2 เปอร์เซปตรอนหลายชั้น (Multi-layer Perceptrons)

### 2.2.1 โครงข่ายป้อนค่าไปข้างหน้าหลายชั้น (Multi-layer Feedforward networks)

เปอร์เซปตรอนหลายชั้น (multi-layer perceptrons) เป็นโครงข่ายประสาทเทียมดังกล่าวมาแล้วในบทที่ 1 (รูปที่ 1.4) เป็นโครงข่ายที่มีชั้นซ่อน (hidden layer) ซึ่งคำว่าซ่อน (hidden) นี้หมายถึงการที่ชั้นนี้ไม่ถูกมองเห็นจากทั้งชั้นอินพุตและชั้นเอาต์พุต และการที่มีชั้นซ่อนไม่ว่าจะเป็น 1 หรือ 2 ชั้น เป็นการที่ทำให้โครงข่ายมีความสามารถในการสกัดข้อมูลในลำดับสูงของอินพุตได้ [Haykin09] ส่วนการคำนวณค่าเอาต์พุตนั้นทำได้โดยที่ เซลล์ที่ชั้นซ่อนชั้นแรกทำการคำนวณหาค่าเอาต์พุตจากเวกเตอร์อินพุต (จากชั้นอินพุต ซึ่งถือเป็นชั้นแรก (first layer)) และเมื่อได้ค่าเอาต์พุตจากทุกเซลล์ จะทำการส่งค่าเอาต์พุตที่ได้ไปยังชั้นซ่อนต่อไป และเซลล์ที่ชั้นต่อไปจะทำการคำนวณหาค่าเอาต์พุตด้วยวิธีการเดิม ซึ่งกระบวนการนี้จะทำซ้ำไปจนกระทั่งได้ค่าเอาต์พุตที่ชั้นเอาต์พุตหรือชั้นสุดท้าย (final layer) โดยที่โครงข่ายลักษณะนี้ถูกเรียกว่า โครงข่ายป้อนค่าไปข้างหน้าหลายชั้น (multi-layer feedforward networks)

ตัวอย่างที่ 2.2 [Haykin09] สมมติให้โครงข่ายประสาทเทียมมีลักษณะดังรูปที่ 2.9 เป็นโครงข่ายที่ถูกสร้างขึ้นมาสำหรับปัญหาของลอจิก XOR ดังตารางความจริง (truth table) ในตารางที่ 2.3 นั่นคือ เวกเตอร์อินพุต  $\mathbf{x} = [x_1, x_2]^T$  ที่มีค่าเอาต์พุต  $y$  เป็น 0 จะอยู่ในคลาสเดียวกัน (สมมติเป็นคลาส 1) ส่วนเวกเตอร์อินพุตที่มีค่าเอาต์พุตเป็น 1 จะอยู่อีกคลาสหนึ่ง (สมมติเป็นคลาส 2) ในตัวอย่างนี้สมมติให้ค่าไบแอสมีค่าเป็น 1 ส่วนค่าน้ำหนักเป็นดังค่าที่กำกับในแต่ละเส้นเชื่อม และให้ฟังก์ชันการกระตุ้นเป็นฟังก์ชันขั้นที่มีค่า  $\gamma_1=1$  และ  $\gamma_2=0$  โดยที่  $\beta=0$  ดังนั้นถ้าอินพุตของฟังก์ชันกระตุ้นมีค่าเป็นลบ จะได้ค่าเอาต์พุตเป็น 0 ในขณะที่ถ้าอินพุตเป็นบวกจะได้ค่าเอาต์พุตเป็น 1

อินพุตของฟังก์ชันกระตุ้นที่จุด 1 คือ

$$v_1 = x_1 + x_2 - 1.5 \quad (2.13)$$

ส่วนอินพุตของฟังก์ชันกระตุ้นที่จุด 2 คือ

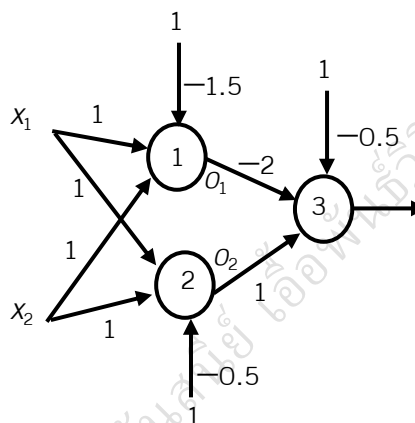


$$v_2 = x_1 + x_2 - 0.5 \quad (2.14)$$

ถ้าทำการสร้างกราฟของเส้นขอบการตัดสินใจของสมการเส้นตรงที่ได้จากสมการที่ 2.13 และ 2.14 จะได้กราฟดังรูปที่ 2.10(ก) และ 2.10(ข) โดยที่เวกเตอร์  $[x_1, x_2]^t$  ใดที่อยู่เหนือเส้นตรงนี้ (พื้นที่สีเทาในรูปที่ 2.10(ก)) ค่า  $v_1$  จะมากกว่า 0 ทำให้ค่า  $o_1$  เป็น 1 เสมอ ในขณะที่  $[x_1, x_2]^t$  ใดที่อยู่ใต้เส้นตรงนี้ ค่า  $v_1$  จะน้อยกว่า 0 ทำให้ค่า  $o_1$  เป็น 0 เสมอ ส่วนเวกเตอร์ที่อยู่เหนือเส้นตรงในรูปที่ 2.10(ข) ค่า  $v_2$  จะมากกว่า 0 ทำให้ค่า  $o_2$  เป็น 1 เสมอ ในขณะที่เวกเตอร์ที่อยู่ใต้เส้นตรง ค่า  $v_2$  จะน้อยกว่า 0 ทำให้ค่า  $o_2$  เป็น 0 เสมอ เช่นกัน

ตารางที่ 2.3 ตารางความจริงสำหรับ XOR

logic		
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



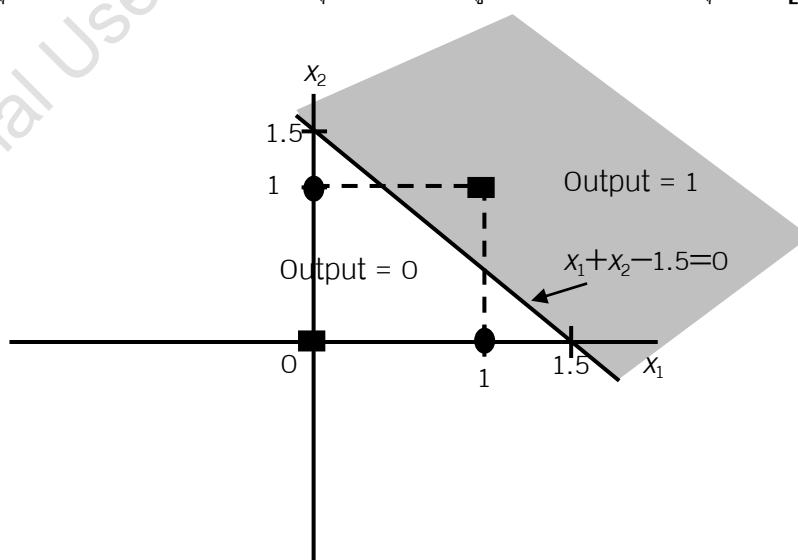
รูปที่ 2.9 เพอร์เซปตรอนหลายชั้นสำหรับ

ตัวอย่างที่ 2.2

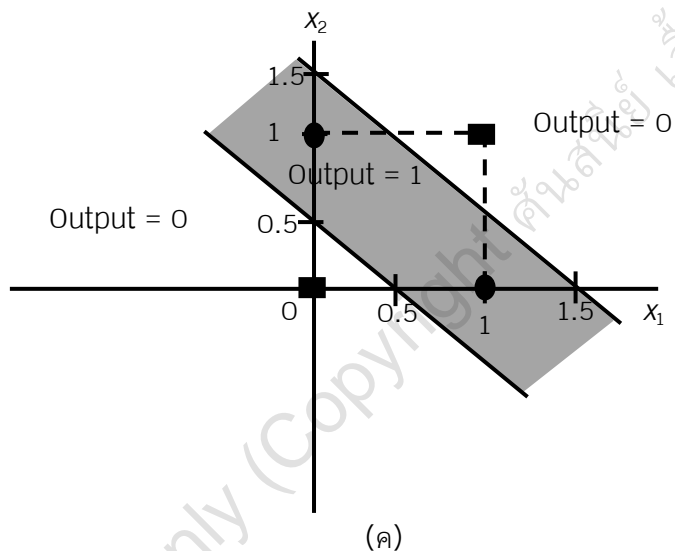
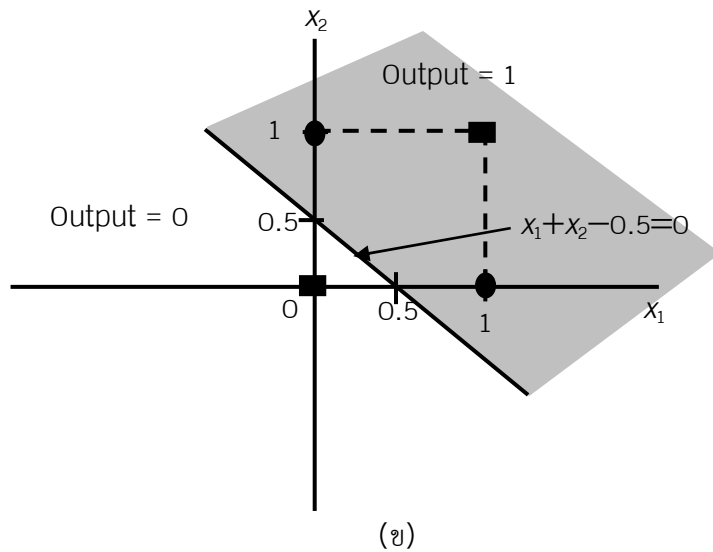
แต่เมื่อนำค่าเอาต์พุตที่ได้จากจุดที่ 1 และ 2 ไปเป็นอินพุตให้กับจุดที่ 3 ทำให้ได้ว่า

$$v_3 = -2o_1 + o_2 - 0.5 \quad (2.15)$$

ดังนั้นฟังก์ชันที่เอาต์พุต เป็นผลรวมเชิงเส้น (linear combination) ของเส้นขอบการตัดสินใจจากเซลล์ในชั้นซ่อนทั้ง 2 จุด และจะเห็นได้ว่าที่จุดที่ 1 มีการเชื่อมต่อแบบยับยั้ง (inhibitory connection) หรือการเชื่อมต่อเป็นลบ (negative connection) ไปยังชั้นเอาต์พุต ในขณะที่ที่จุดที่ 2 มีการเชื่อมต่อแบบกระตุ้น (excitatory connection) หรือการเชื่อมต่อแบบบวก (positive) ไปยังชั้นเอาต์พุต ดังนั้นเมื่อเซลล์ทั้งสอง (จุด 1 และ 2) ถูกปิด คือกรณีที่อินพุตเป็น  $[0,0]^t$



(ก)



รูปที่ 2.10 เส้นขอบการตัดสินใจของ (ก) จุดที่ 1 (ข) จุดที่ 2 และ (ค) โครงข่ายประสาทเทียมทั้งหมด

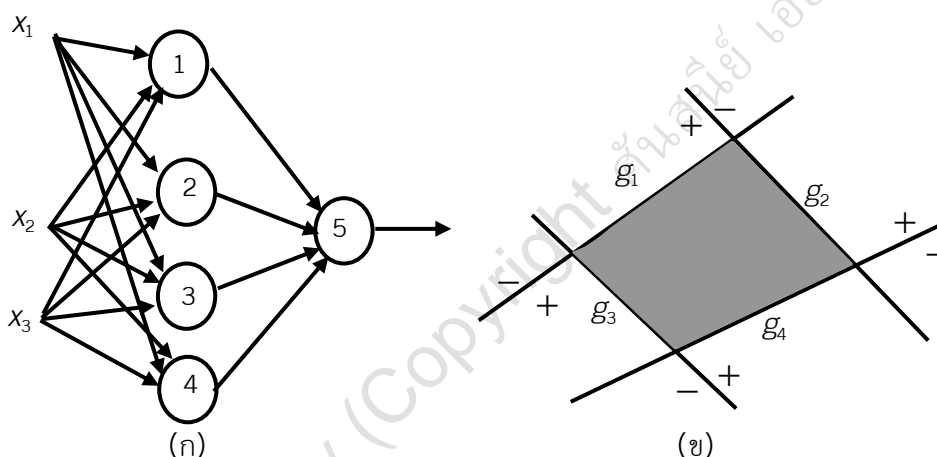
ตารางที่ 2.4 อินพุตและเอาต์พุตสำหรับตัวอย่างที่ 2.2

$x_1$	$x_2$	$v_1$	$o_1$	$v_2$	$o_2$	$v$	$o$
0	0	-1.5	0	-0.5	0	-0.5	0
0	1	-0.5	0	0.5	1	0.5	1
1	0	-0.5	0	0.5	1	0.5	1
1	1	0.5	1	1.5	1	-1.5	0

ค่าเอาต์พุตที่ได้เป็น 0 ด้วย และเมื่อทั้งสองเซลล์ถูกเปิดคือกรณีที่อินพุตเป็น  $[1,1]^t$  ค่าเอาต์พุตที่ได้เป็น 0 เช่นกัน ทั้งนี้เป็นเพราะค่าน้ำหนักของการเชื่อมต่อแบบยับยั้งมีค่ามากกว่าค่าน้ำหนักของการเชื่อมต่อแบบกระตุ้นแต่ถ้าค่าอินพุตของทั้งสองเซลล์เป็น  $[0,1]^t$  หรือ  $[1,0]^t$  ค่าเอาต์พุตเอาต์พุตที่ได้เป็น 1 ทั้งนี้เพราะค่าน้ำหนักของการเชื่อมต่อแบบกระตุ้นมีผลมากกว่านั่นเอง (ดัง

แสดงในตารางที่ 2.4) หรือถ้ามองจากรูปที่ 2.10(ค) จะเห็นว่าค่าเอาต์พุตที่ได้จากจุดที่ 1 จะถูกกลับทิศ ก่อนที่จะนำมาอินเตอร์เซก (intersect) กับค่าเอาต์พุตที่ได้จากจุดที่ 2 นั้นเอง

จากตัวอย่างที่ 2.2 ทำให้เห็นว่าที่จริงแล้วชั้นซ่อนทำหน้าที่ในการส่งผ่าน (mapping) ปริภูมิอินพุต (input space) ไปยังปริภูมิซ่อน (hidden space) ซึ่งในชั้นซ่อนนี้มี ระนาบเกิน (hyperplane)  $p$  ระนาบ (เท่ากับจำนวนเซลล์ในชั้นซ่อน) ซึ่งแต่ละระนาบจะถูกสร้างจากแต่ละเซลล์ในชั้นซ่อน และเช่นเดียวกับในตัวอย่างที่ 2.2 แต่ละระนาบจะเปรียบเสมือนระนาบการตัดสินใจของแต่ละเซลล์ ในขณะที่ผลรวมหรือการอินเตอร์เซกของระนาบเหล่านี้ เป็นระนาบการตัดสินใจของทั้งโครงข่าย [Theodoridis09] ดังนั้นถ้าที่ชั้นซ่อนมี 4 เซลล์ ดังรูปที่ 2.11(ก) ลักษณะของระนาบเกินทั้ง 4 ( $g_1-g_4$  โดยที่ระนาบ  $g_i$  มากจากเซลล์ที่  $i$  ที่ชั้นซ่อนนั่นเอง) จะเป็นดังรูปที่ 2.11(ข) โดยที่พื้นที่ข้างในกรอบ (สีเทา) เกิดจากการอินเตอร์เซกของพื้นที่ที่เป็นบวกของทุกระนาบ ทำให้เป็นพื้นที่ของเอาต์พุตที่เป็นคลาส (class) 1 หรือคลาสบวก ส่วนพื้นที่ข้างนอกเกิดจากการอินเตอร์เซกของพื้นที่ที่เป็นลบของทุกระนาบ ทำให้เป็นพื้นที่ของคลาส 2 หรือคลาสลบ



รูปที่ 2.11 (ก) โครงข่ายประสาทเทียม 3-4-1 (ข) ลักษณะของระนาบเกินของโครงข่ายในรูป (ก)

### 2.2.2 อัลกอริทึมการแพร่กระจายย้อนกลับ (Back-propagation Algorithm)

จากการกล่าวถึงโครงข่ายป้อนค่าไปข้างหน้า (feedforward networks) ไม่ว่าจะเป็นโครงข่ายชั้นเดียว หรือหลายชั้นก็ตามเป็นการกล่าวถึงในลักษณะที่ค่าน้ำหนัก (synaptic weights หรือ weight) ที่กำกับในแต่ละเชื่อม เป็นค่าน้ำหนักที่มีค่าที่แน่นอนแล้ว แต่ในการประยุกต์ใช้งาน ค่าน้ำหนักเหล่านี้เป็นค่าที่ไม่ทราบค่า และนอกเหนือผู้ใช้งานไม่ทราบว่าโครงสร้าง (structure) ของโครงข่ายควรเป็นลักษณะเช่นใด ตัวอย่างเช่นผู้ใช้งานต้องการสร้างโครงข่ายประสาทเทียมที่สามารถแยกแยะ ผู้หญิงออกจากผู้ชาย โดยมีตัวอย่างผู้หญิง และผู้ชายอยู่อย่างละ 100 คน โดยปกติผู้ใช้งานจะทำการคัดลักษณะ (feature extraction) เช่นอาจจะเป็นส่วนสูง ( $x_1$ ) น้ำหนัก ( $x_2$ ) และ ดัชนีมวลกาย ( $x_3$ ) ของแต่ละคน ดังนั้นจะมีเวกเตอร์ลักษณะ (feature vector) ( $\mathbf{x}=[x_1, x_2, x_3]^t$ ) ของผู้หญิง 100 เวกเตอร์ และเวกเตอร์ลักษณะของผู้ชาย 100 เวกเตอร์ โดยที่เวกเตอร์ลักษณะเหล่านี้แต่ละเวกเตอร์จะมีเวกเตอร์เอาต์พุตกำกับอยู่ด้วย เช่นถ้าเป็นเวกเตอร์ในคลาสผู้หญิง เวกเตอร์เอาต์พุตจะเป็น  $[1,0]^t$  แต่ถ้าเป็นเวกเตอร์ในคลาสผู้ชาย เวกเตอร์เอาต์พุตจะเป็น  $[0,1]^t$  ซึ่งเวกเตอร์อินพุตและเอาต์พุตทั้ง 200 เวกเตอร์นี้จะถูกเรียกว่าชุดข้อมูลฝึกสอน (training data set)

และเมื่อต้องการสร้างโครงข่ายประสาทเทียม ผู้ใช้จะต้องทำการสร้างโครงข่ายประสาทเทียมในหลายๆ ลักษณะ โดยมีเซลล์ที่ชั้นอินพุต 3 จุดเนื่องจากเวกเตอร์อินพุตมี 3 มิติ และเซลล์ที่ชั้นเอาต์พุตเป็น 2 เนื่องจากมี 2 คลาส เช่นโครงข่าย 3-4-2 หรือ 3-5-3-2 เป็นต้น สมมติให้ผู้เลือกใช้โครงข่าย 3-4-2 จำเป็นที่จะต้องทำให้โครงข่ายมีการเรียนรู้จากชุดข้อมูลฝึกสอน และหนึ่งในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมคือ อัลกอริทึมการแพร่กระจายย้อนกลับ (back-propagation algorithm) ซึ่งจะได้กล่าวถึงต่อไป แต่เนื่องจากฟังก์ชันการกระตุ้นที่นิยมใช้เป็นฟังก์ชันซิกมอยด์ เช่นฟังก์ชันลอจิสติก ซึ่งฟังก์ชันนี้มีค่าที่ปลายด้าน  $-\infty$  เป็นค่าเข้าใกล้ 0 ไม่ใช่ 0 ส่วนค่าที่ปลายด้าน  $\infty$  เป็นค่าเข้าใกล้ 1 ไม่ใช่ 1 ดังนั้นค่าเอาต์พุตที่ต้องการ จะเป็น  $0+\epsilon$  และ  $1-\epsilon$  แทนที่จะเป็น 0 หรือ 1 โดยปกติ  $\epsilon=0.1$  ดังนั้นเวกเตอร์เอาต์พุตจะเป็น  $[0.9, 0.1]^t$  และ  $[0.1, 0.9]^t$  สำหรับคลาสผู้หญิงและผู้ชาย แต่ถ้าฟังก์ชันการกระตุ้นที่ใช้เป็นฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ ค่าเอาต์พุตที่ต้องการก็ต้องถูกปรับเป็น  $-1+\epsilon$  และ  $1-\epsilon$  ดังนั้นเวกเตอร์เอาต์พุตจะเป็น  $[0.9, -0.9]^t$  และ  $[-0.9, 0.9]^t$  สำหรับเวกเตอร์จากทั้งสองคลาสเช่นเดียวกัน ทั้งนี้เพื่อป้องกันการที่อัลกอริทึมการแพร่กระจายย้อนกลับ ทำให้ค่าของพารามิเตอร์ที่ต้องการกลายเป็นค่าอนันต์ และทำให้กระบวนการเรียนรู้ช้าลงด้วยอัตราของขนาด [Haykin09] นั่นเอง

เมื่อโครงข่ายประสาทเทียมนี้ทำการเรียนรู้จากชุดข้อมูลฝึกสอน และได้ค่าน้ำหนักที่ดี ผู้ใช้สามารถนำโครงข่ายนี้ไปทดสอบกับเวกเตอร์ลักษณะจากชุดข้อมูลทดสอบ (testing data set) ด้วยวิธีการคำนวณจากโครงข่ายป้อนค่าไปข้างหน้าหลายชั้นดังที่กล่าวแล้วในหัวข้อที่แล้ว และถ้าค่าที่เอาต์พุตที่หนึ่งของชั้นเอาต์พุตมีค่ามากกว่า เวกเตอร์ทดสอบนั้นจะถูกจัดให้อยู่คลาสผู้หญิง แต่ถ้าค่าเอาต์พุตที่สองของชั้นเอาต์พุตมีค่ามากกว่า เวกเตอร์ทดสอบนั้นจะถูกจัดให้อยู่คลาสผู้ชาย นั่นเอง

ก่อนที่จะกล่าวถึงอัลกอริทึมการแพร่กระจายย้อนกลับ จะขอกล่าวถึงกฎการเรียนรู้การตกลงของเกรเดียนต์ (gradient descent learning rule) [Engelbrecht07] หรือ การตกลงมาที่ชันที่สุด (steepest descent) [Haykin09] ซึ่งเป็นวิธีการที่นิยมใช้กันมากที่สุดในการเรียนรู้ของโครงข่ายประสาทเทียม โดยที่กระบวนการนี้เป็นการหาค่าที่เหมาะสมที่ไม่มีเงื่อนไขบังคับ (unconstrained optimization) ซึ่งอยู่บนพื้นฐานของความคิด (idea) ของการทำซ้ำของการตกลงมา (iterative descent) นั่นคือ [Haykin09] เป็นกระบวนการที่เริ่มจากการเดาเริ่มต้น (initial guess) ของเวกเตอร์น้ำหนัก  $w$  ที่การทำซ้ำครั้งที่ 0 ( $w(0)$ ) หลังจากนั้นปรับค่าเวกเตอร์น้ำหนัก ( $w(1), w(2), \dots$ ) ไปเรื่อยๆ จนกระทั่งฟังก์ชันคอส (cost function) หรือฟังก์ชันความผิดพลาด (error function) ( $\mathcal{E}(w)$ ) เป็นฟังก์ชันที่หาอนุพันธ์ได้ต่อเนื่อง (continuously differentiable function) ของพารามิเตอร์ หรือเวกเตอร์ของน้ำหนักที่ไม่ทราบค่า ( $w$ ) ลดลง นั่นคือ

$$\mathcal{E}(w(t+1)) < \mathcal{E}(w(t)) \quad (2.16)$$

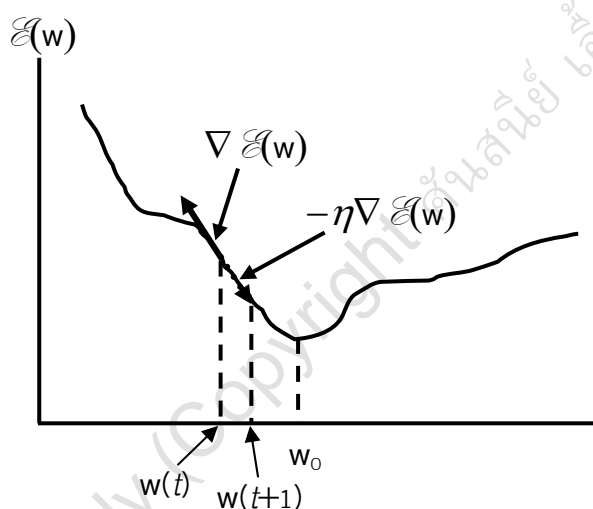
โดยที่  $w(t)$  และ  $w(t+1)$  เป็นเวกเตอร์น้ำหนักเดิม และเวกเตอร์น้ำหนักที่ถูกปรับค่า

ดังนั้นกฎการเรียนรู้การตกลงของเกรเดียนต์ (gradient descent learning rule) [Engelbrecht07] หรือ การตกลงมาที่ชันที่สุด (steepest descent) [Haykin09] เป็นการปรับเวกเตอร์น้ำหนัก  $w$  ด้วยทิศทางของการตกลงมาที่ชันที่สุด (direction of steep descent) ซึ่งทิศทางนี้เป็นทิศทางที่ตรงกันข้ามกับทิศทางของเกรเดียนต์เวกเตอร์ ( $\nabla \mathcal{E}(w)$ ) โดยที่เวกเตอร์  $w$  ที่ต้องการหาเป็นเวกเตอร์ที่ทำให้  $\mathcal{E}(w)$  มีค่าน้อยที่สุด โดยปกติแล้วทิศทางของเกรเดียนต์

เวกเตอร์จะมีทิศทางที่พุ่งเข้าหาค่าสูงเสมอ [Kreyszig05] ดังนั้นการปรับเวกเตอร์น้ำหนัก  $w$  ด้วยทิศทางของการตกลงมาที่ชันที่สุด ซึ่งตรงกันข้ามกับทิศทางของเกรเดียนต์เวกเตอร์นั้น เป็นการปรับค่า  $w$  เพื่อให้มีค่า  $\mathcal{E}(w)$  มีค่าน้อยลงจนกระทั่งได้  $w$  ที่ทำให้  $\mathcal{E}(w)$  มีค่าน้อยที่สุดนั่นเอง ดังเช่นตัวอย่างในรูปที่ 2.12 สมมติให้ ณ. การทำซ้ำ (iteration) ที่  $t$  ให้  $w(t)$  เป็นเวกเตอร์ที่ให้ค่า  $\mathcal{E}(w)$  ค่าหนึ่ง และเมื่อทำการหา  $\nabla \mathcal{E}(w)$  จะได้เวกเตอร์ดังในรูป 2.12 แต่เมื่อทำการกลับทิศของ  $\nabla \mathcal{E}(w)$  จะได้เวกเตอร์ที่มีทิศทางดังรูป 2.12 และมีขนาดที่ถูกคูณด้วยค่าคงที่ ( $\eta$ ) ที่เป็นค่าบวกค่าหนึ่งที่เราเรียกว่าอัตราการเรียน (learning rate) หรือขนาดของขั้น (step size) ทำให้เมื่อมีการปรับค่า  $w$  จะได้

$$w(t+1) = w(t) - \eta \nabla \mathcal{E}(w) \quad (2.17)$$

และเมื่อทำการปรับค่า  $w$  โดยการทำซ้ำไปหลายครั้ง ก็จะมีโอกาสที่จะได้  $w$  ที่เป็นเวกเตอร์ที่ทำให้  $\mathcal{E}(w)$  น้อยที่สุด คือที่  $w_0$  นั่นเอง ซึ่งกระบวนการนี้ได้ถูกพิสูจน์มาแล้ว [Haykin09] ว่าเป็นไปตามสมการที่ 2.16



รูปที่ 2.12 เกรเดียนต์เวกเตอร์ ( $\nabla \mathcal{E}(w)$ )

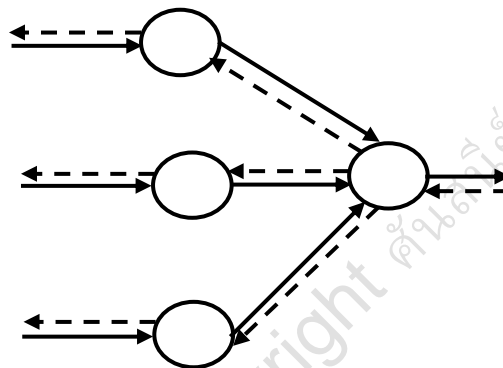
วิธีการตกลงมาที่ชันที่สุด (steepest descent) จะให้คำตอบที่ลู่ออกหา  $w_0$  อย่างช้าๆ [Haykin09] และอัตราการเรียนรู้อิทธิพลต่อการลู่ออกหาคำตอบดังนี้

1. เมื่อ  $\eta$  มีค่าน้อย การตอบสนองชั่วคราว (transient response) ของอัลกอริทึม จะเป็นแบบหน่วงมาก (overdamped) ดังนั้นการปรับค่า  $w(t)$  จะเป็นการปรับแบบค่อยเป็นค่อยไป
2. เมื่อ  $\eta$  มีค่ามาก การตอบสนองชั่วคราว (transient response) ของอัลกอริทึม จะเป็นแบบหน่วงน้อย (underdamped) ดังนั้นการปรับค่า  $w(t)$  จะเป็นการปรับแบบซิกแซก หรือมีการกระโดดไปมา หรือแบบแกว่งกวัด (oscillate)
3. เมื่อ  $\eta$  มีค่ามากกว่าค่าวิกฤติ (critical value) ค่าหนึ่ง อัลกอริทึมจะไม่เสถียร (unstable) หรือเป็นการลู่ออก (diverge)

จากรูปที่ 1.4 ในบทที่ 1 นั้น มีสัญญาณอยู่ด้วยกัน 2 ประเภท ดังรูปที่ 2.13 นั่นคือ [Haykin09]

1. สัญญาณฟังก์ชัน (function signals) เป็นสัญญาณอินพุต (ตัวกระตุ้น (stimulus)) ที่เข้ามาที่ชั้นอินพุต และแพร่กระจายไปข้างหน้าจากเซลล์หนึ่งไปยังอีกเซลล์หนึ่ง ในโครงข่าย จนกระทั่งได้เอาต์พุตที่ชั้นเอาต์พุต ซึ่งเป็นสัญญาณที่คำนวณมาจากอินพุตและค่าน้ำหนักต่างๆในโครงข่าย
2. สัญญาณความผิดพลาด (error signals) เป็นค่าความผิดพลาดที่ชั้นเอาต์พุต และแพร่กระจายย้อนกลับ ชั้นต่อชั้นในโครงข่าย และเป็นค่าที่มีการคำนวณที่ทุกเซลล์ในโครงข่ายที่เกี่ยวข้องกับความผิดพลาดนั้น

ดังนั้นแต่ละเซลล์ในชั้นซ่อน และชั้นเอาต์พุตในเปอร์เซปตรอนหลายชั้นจะทำหน้าที่ในการคำนวณหาสัญญาณฟังก์ชันที่ได้รับจากชั้นอินพุตและส่งต่อไปชั้นอื่น จนกระทั่งได้สัญญาณที่ชั้นเอาต์พุต ซึ่งขั้นตอนนี้เรียกว่าการผ่านไปข้างหน้า (forward pass) และคำนวณค่าของการประมาณเกรเดียนต์เวกเตอร์ของตัวอย่างนั้นๆ กลับไปยังชั้นแรกๆ ของโครงข่าย ซึ่งขั้นตอนนี้ถูกเรียกว่าการผ่านย้อนกลับ (backward pass)



รูปที่ 2.13 สัญญาณฟังก์ชัน (เส้นทึบ) และสัญญาณความผิดพลาด (เส้นประ)

ในส่วนของการผ่านไปข้างหน้า ได้กล่าวถึงมาแล้วในเรื่องของโครงข่ายป้อนค่าไปข้างหน้า ส่วนการผ่านย้อนกลับจะเป็นกระบวนการของ อัลกอริทึมการแพร่กระจายย้อนกลับ (back-propagation algorithm) แต่ก่อนที่จะกล่าวถึงอัลกอริทึม จะขออธิบายถึงสัญลักษณ์ที่ถูกใช้สำหรับอัลกอริทึมนี้ ซึ่งมีดังต่อไปนี้

- $E(t)$  หมายถึงผลรวมของความผิดพลาดกำลังสอง (sum of squared errors) ของเซลล์ที่ชั้นเอาต์พุต สำหรับการทำซ้ำ (iteration) ครั้งที่  $t$
- $E_{av}$  หมายถึงค่าเฉลี่ยของความผิดพลาดกำลังสอง (mean squared error) ของทุกเวกเตอร์อินพุต ในชุดข้อมูลฝึกสอน
- $e_j(t)$  หมายถึงความผิดพลาดของเซลล์ที่  $j$  ที่ชั้นเอาต์พุตสำหรับการทำซ้ำครั้งที่  $t$
- $d_j(t)$  หมายถึงค่าเอาต์พุตที่ต้องการ (desire output) ของเซลล์ที่  $j$  ที่ชั้นเอาต์พุต สำหรับการทำซ้ำครั้งที่  $t$
- $o_j(t)$  หมายถึงค่าเอาต์พุตที่คำนวณได้จากโครงข่าย (actual/program output) ของเซลล์ที่  $j$  ที่ชั้นเอาต์พุตสำหรับการทำซ้ำครั้งที่  $t$
- $y_j(t)$  หมายถึงค่าเอาต์พุตจากเซลล์ที่  $j$  สำหรับการทำซ้ำครั้งที่  $t$
- $w_{ji}(t)$  หมายถึงค่าน้ำหนักของจุดประสานประสาท (synaptic weight) หรือค่าน้ำหนักที่กำกับกับการเชื่อมต่อจากเซลล์ที่  $i$  ไปยังเซลล์ที่  $j$  สำหรับการทำซ้ำครั้งที่  $t$

- $\Delta w_{ji}(t)$  หมายถึงการเปลี่ยนแปลงค่าน้ำหนักที่กำกับการเชื่อมต่อกับเซลล์ที่  $i$  ไปยังเซลล์ที่  $j$  สำหรับการกระทำซ้ำครั้งที่  $t$
- $v_j(t)$  หมายถึงค่าระดับกิจกรรม (activity level) ของเซลล์ที่  $j$  สำหรับการกระทำซ้ำครั้งที่  $t$
- $\phi_j(\cdot)$  หมายถึงฟังก์ชันการกระตุ้นของเซลล์ที่  $j$
- $w_{jo}(t)$  หรือ  $\theta_j(t)$  หมายถึงค่าไบแอส (bias) ที่กำกับการเชื่อมไปยังเซลล์ที่  $j$
- $N$  หมายถึงจำนวนตัวอย่างในชุดข้อมูลฝึกสอน

สำหรับการคำนวณในอัลกอริทึมการแพร่กระจายย้อนกลับ จำเป็นต้องคำนวณหาค่าความผิดพลาดของเซลล์ที่  $j$  (ดังรูปที่ 2.14 โดยที่สมมุติให้ชั้นก่อนหน้ามี  $p$  เซลล์) ที่ชั้นเอาต์พุตสำหรับการกระทำซ้ำที่  $t$  ซึ่งในการกระทำซ้ำครั้งนี้มีการนำตัวอย่าง 1 ตัวอย่างในชุดข้อมูลฝึกสอน และส่งผ่านโครงข่ายโดยการคำนวณหาค่าเอาต์พุตที่ได้จากการคำนวณแบบโครงข่ายป้อนค่าไปข้างหน้าของตัวอย่างนี้ ดังนั้นค่าความผิดพลาดคือ

$$e_j(t) = d_j(t) - y_j(t) \quad (2.19)$$

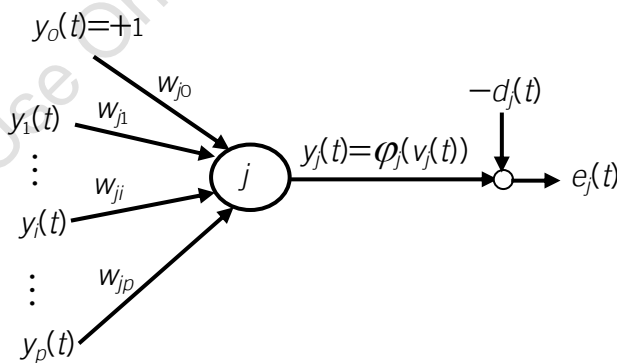
ส่วนผลรวมของความผิดพลาดกำลังสอง ของเซลล์ที่ชั้นเอาต์พุต (โดยที่สมมุติให้  $C$  คือจำนวนเซลล์ที่ชั้นเอาต์พุต) สำหรับการกระทำซ้ำครั้งที่  $t$  คือ

$$\mathcal{E}(t) = \frac{1}{2} \sum_{j \in C} e_j^2(t) \quad (2.20)$$

แต่ถ้าต้องการคำนวณหาเฉลี่ยของความผิดพลาดกำลังสอง ของทุกเวกเตอร์อินพุต ในชุดข้อมูลฝึกสอน จะได้

$$\mathcal{E}_{av} = \frac{1}{N} \sum_{n=1}^N \mathcal{E}(n) \quad (2.21)$$

โดยที่  $\mathcal{E}(n)$  เป็นผลรวมของความผิดพลาดกำลังสอง ของเซลล์ที่ชั้นเอาต์พุต สำหรับตัวอย่างที่  $n$



รูปที่ 2.14 รายละเอียดของสัญญาณที่เซลล์ที่  $j$  ที่ชั้นเอาต์พุต

และจากโครงข่ายป้อนค่าไปข้างหน้าได้ว่า

$$v_j(t) = \sum_{i=0}^p w_{ji}(t) y_i(t) \quad (2.22)$$

โดยที่  $p$  คือจำนวนเซลล์ในชั้นก่อนหน้าชั้นเอาต์พุต โดยไม่รวมอินพุตจากไบแอส ซึ่งเซลล์เหล่านี้จะสร้างเอาต์พุต ที่จะเป็นอินพุตให้กับเซลล์ที่  $j$  ที่ชั้นเอาต์พุต ส่วนค่าเอาต์พุตของเซลล์ที่  $j$  คือ

$$y_j(t) = \varphi_j(v_j(t)) \quad (2.23)$$

เนื่องจากกระบวนการของการตกลงมาที่ชันที่สุด (steepest descent) [Haykin09] จำเป็นต้องหาเกรเดียนต์เวกเตอร์ โดยที่สามารถหาจากแต่ละสมาชิก (element) ในเวกเตอร์ และเนื่องจากสมการที่ 2.20 ไม่ปรากฏ  $w_{ji}$  ดังนั้นในกรณีนี้จำเป็นต้องใช้การคำนวณจาก กฎลูกโซ่ (chain rule) [Haykin09] ได้ดังนี้

$$\frac{\partial \mathcal{E}(t)}{\partial w_{ji}(t)} = \frac{\partial \mathcal{E}}{\partial e_j(t)} \frac{\partial e_j(t)}{\partial y_i(t)} \frac{\partial y_i(t)}{\partial v_i(t)} \frac{\partial v_i(t)}{\partial w_{ji}(t)} \quad (2.24)$$

ซึ่งจากสมการที่ 2.20 จะได้ว่า

$$\frac{\partial \mathcal{E}}{\partial e_j(t)} = e_j(t) \quad (2.25)$$

และจากสมการที่ 2.19 จะได้ว่า

$$\frac{\partial e_j(t)}{\partial y_i(t)} = -1 \quad (2.26)$$

ส่วนการหาอนุพันธ์อันดับ 1 ของ  $y_i$  นั้นที่จริงคือการหาอนุพันธ์อันดับ 1 ของฟังก์ชันการกระตุ้น ดังนี้

$$\frac{\partial y_i(t)}{\partial v_i(t)} = \varphi_j'(v_j(t)) \quad (2.27)$$

และจากสมการที่ 2.22 จะได้ว่า

$$\frac{\partial v_i(t)}{\partial w_{ji}(t)} = y_i(t) \quad (2.28)$$

ดังนั้นสมการที่ 2.24 คือ

$$\frac{\partial \mathcal{E}(t)}{\partial w_{ji}(t)} = -e_j(t) \varphi_j'(v_j(t)) y_i(t) \quad (2.29)$$

ดังนั้นการเปลี่ยนแปลงค่าน้ำหนักที่กำกับการเชื่อมต่อจากเซลล์ที่  $i$  ไปยังเซลล์ที่  $j$  สำหรับการซ้ำครั้งที่  $t$  คือ

$$\Delta w_{ji}(t) = -\eta \frac{\partial \mathcal{E}(t)}{\partial w_{ji}(t)} = \eta e_j(t) \varphi_j'(v_j(t)) y_i(t) \quad (2.30)$$

โดยที่  $\eta$  เป็นอัตราการเรียนรู้ (learning rate) ที่เป็นค่าคงที่บวก และจากสมการที่ 2.30 สามารถสรุปได้ว่า

$$\Delta w_{ji}(t) = \eta \delta_j(t) y_i(t) \quad (2.31)$$

โดยที่  $\delta_j(t)$  เป็นค่าเกรเดียนต์เฉพาะที่ (local gradient) ซึ่งคือ

$$\delta_j(t) = -\frac{\partial \mathcal{E}(t)}{\partial e_j(t)} \frac{\partial e_j(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)} = e_j(t) \varphi_j'(v_j(t)) \quad (2.32)$$

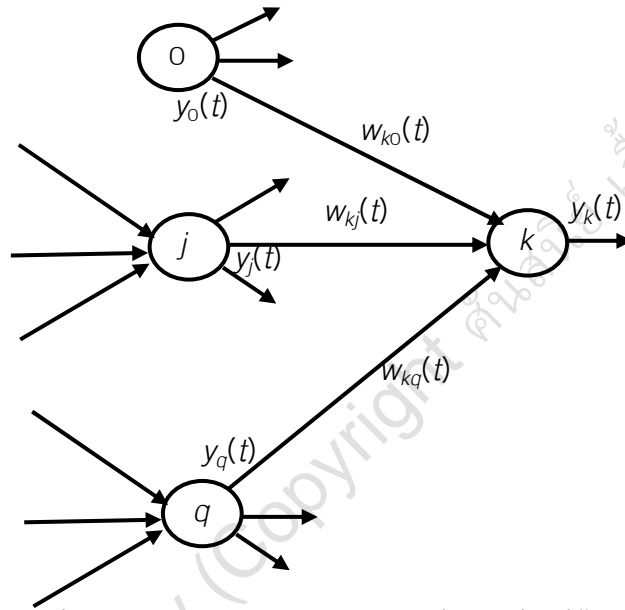
ซึ่งค่าเกรเดียนต์เฉพาะที่นี้ จะบ่งบอกถึงการเปลี่ยนแปลงของค่าน้ำหนักของจุดประสานประสาท (synaptic weight) โดยที่มีค่าเท่ากับผลคูณของค่าความผิดพลาดของค่าเอาต์พุตที่เซลล์ที่  $j$  และ



อนุพันธ์อันดับ 1 ของฟังก์ชันการกระตุ้น นั่นเอง ส่วนการปรับค่าน้ำหนักของเส้นเชื่อมที่พุ่งเข้าหาเซลล์ที่  $j$  ที่ชั้นเอาต์พุตคือ

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \quad (2.33)$$

แต่การปรับค่านี้เป็นการปรับค่าเนื่องจากการเปลี่ยนแปลงค่าน้ำหนักที่กำกับการเชื่อมต่อจากเซลล์ที่  $i$  ไปยังเซลล์ที่  $j$  ที่อยู่ชั้นเอาต์พุต สำหรับการซ้ำครั้งที่  $t$  ดังนั้นจึงต้องทำการหาการปรับค่าน้ำหนักสำหรับเส้นเชื่อมที่ชั้นซ่อนด้วย แต่ที่ชั้นซ่อนนั้นไม่สามารถคำนวณหาค่าความผิดพลาดได้โดยตรง เนื่องจากไม่ทราบค่าเอาต์พุตที่ต้องการของแต่ละเซลล์ที่ชั้นซ่อน ดังนั้นจำเป็นต้องหาจากค่าความผิดพลาดจากเซลล์ (เซลล์ที่  $k$  จากรูปที่ 2.15) ของชั้นที่ตามหลังชั้นซ่อนนั้น โดยคำนวณจากเซลล์ทุกเซลล์ที่เชื่อมต่อกับเซลล์นั้นซ่อน ซึ่งเป็นกระบวนการของการแพร่กระจายย้อนกลับ นั่นเอง



รูปที่ 2.15 รายละเอียดของสัญญาณที่เซลล์ที่  $j$  ที่ชั้นซ่อน

สมมติให้เซลล์ที่  $j$  เป็นเซลล์ที่ชั้นซ่อน ดังรูปที่ 2.15 ค่าเกรเดียนต์เฉพาะที่ของเซลล์นี้คือ

$$\delta_j(t) = -\frac{\partial \mathcal{E}(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)} = -\frac{\partial \mathcal{E}(t)}{\partial y_j(t)} \phi_j'(v_j(t)) \quad (2.34)$$

เนื่องจากค่าความผิดพลาดของเซลล์ที่ชั้นเอาต์พุตทุกเซลล์เกิดจากค่าเอาต์พุตของเซลล์นั้น ซึ่งค่าเอาต์พุตเหล่านั้นเกิดจากอินพุตที่มาจากเซลล์ที่  $j$  ที่ชั้นซ่อนนี้ด้วย และจากสมการที่ 2.20 จะได้ว่า

$$\frac{\partial \mathcal{E}(t)}{\partial y_j(t)} = \sum_k e_k(t) \frac{\partial e_k(t)}{\partial y_j(t)} = \sum_k e_k(t) \frac{\partial e_k(t)}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial y_j(t)} \quad (2.35)$$

จากสมการที่ 2.19 และ 2.23 จะได้ว่า

$$\frac{\partial e_k(t)}{\partial v_k(t)} = \frac{\partial e_k(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial v_k(t)} = -\phi_k'(v_k(t)) \quad (2.36)$$

เมื่อนำสมการที่ 2.22 มาเขียนใหม่จะได้ว่า

$$v_k(t) = \sum_{j=0}^q w_{kj}(t) y_j(t) \quad (2.37)$$

โดยที่  $q$  คือจำนวนเซลล์ ในชั้นก่อนหน้า โดยไม่รวมอินพุตจากไบแอส ซึ่งเซลล์เหล่านี้จะสร้างเอาต์พุต ที่จะเป็นอินพุตให้กับเซลล์ที่  $k$  ที่ชั้นเอาต์พุต ดังนั้น

$$\frac{\partial v_k(t)}{\partial y_j(t)} = w_{kj}(t) \quad (2.38)$$

และสมการที่ 2.36 และ 2.38 ทำให้สมการที่ 2.35 กลายเป็น

$$\frac{\partial \mathcal{E}(t)}{\partial y_j(t)} = -\sum_k e_k(t) \phi'_k(v_k(t)) w_{kj}(t) \quad (2.39)$$

แต่จากสมการที่ 2.32 ทำให้สมการที่ 2.39 กลายเป็น

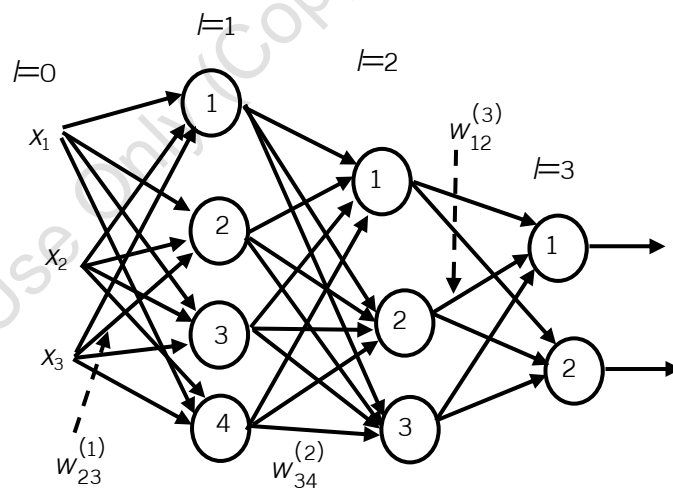
$$\frac{\partial \mathcal{E}(t)}{\partial y_j(t)} = -\sum_k \delta_k(t) w_{kj}(t) \quad (2.40)$$

ดังนั้นสมการที่ 2.34 กลายเป็น

$$\delta_j(t) = \phi'_j(v_j(t)) \sum_k \delta_k(t) w_{kj}(t) \quad (2.41)$$

ซึ่งจะเป็นว่าเกรเดียนต์เฉพาะที่ที่เซลล์ที่  $j$  ที่ชั้นซ่อนนี้มาจากผลรวมของเกรเดียนต์เฉพาะที่ของทุกเซลล์ที่เชื่อมต่อกับเซลล์นี้นั่นเอง ส่วนการปรับค่าน้ำหนักสามารถใช้สมการที่ 2.31 และ 2.33 ได้เช่นกัน

เพื่อให้เกิดความเข้าใจที่ง่ายขึ้น สัญลักษณ์ / ในรูปที่ 2.16 หมายถึงลำดับชั้นในโครงข่าย ซึ่งโดยปกติ ที่ชั้นอินพุต  $I=0$  เสมอ และในรูปที่ 2.16 นี้ ชั้นเอาต์พุตเป็นชั้นสุดท้ายที่  $I=L=3$  และสัญลักษณ์ที่ใช้ในส่วนนี้คือ



รูปที่ 2.16 ลักษณะโครงข่ายประสาทเทียมที่มีเลขลำดับชั้นกำกับ

- $w^{(l)}$  หมายถึงเวกเตอร์น้ำหนักของเซลล์ที่อยู่ในชั้นที่  $l$
- $\theta^{(l)}$  หมายถึงค่าไบแอส (bias) ของเซลล์ที่อยู่ในชั้นที่  $l$
- $v^{(l)}$  หมายถึงเวกเตอร์ค่าระดับกิจกรรม (activity level) ของเซลล์ที่อยู่ในชั้นที่  $l$
- $y^{(l)}$  หมายถึงเวกเตอร์ค่าเอาต์พุตจากเซลล์ที่อยู่ในชั้นที่  $l$
- $e$  หมายถึงเวกเตอร์ความผิดพลาดที่ชั้นเอาต์พุต

—  $\delta^{(l)}$  หมายถึงเวกเตอร์ค่าเกรเดียนต์เฉพาะที่ของเซลล์ที่อยู่ในชั้นที่  $l$

ดังนั้นสามารถสรุปกระบวนการการแพร่กระจายย้อนกลับ ได้ดังนี้

1. การผ่านไปข้างหน้า (forward pass) หรือ การแพร่กระจายไปข้างหน้า (forward propagation)

ในขั้นตอนนี้ไม่มีการเปลี่ยนค่าน้ำหนักในโครงข่าย แต่ค่าสัญญาณของโครงข่ายจะถูกคำนวณ จากเซลล์ไปสู่เซลล์ ซึ่งค่าเอาต์พุตของเซลล์ที่  $j$  ที่ชั้น  $l$  คือ

$$y_j^{(l)}(t) = \varphi_j^{(l)}(v_j^{(l)}(t)) \quad (2.42)$$

โดยที่  $v_j^{(l)}(t)$  คือค่าระดับกิจกรรม (activity level) ของเซลล์ที่  $j$  ที่ชั้น  $l$  สำหรับการเข้าครั้ง  $t$

$$v_j^{(l)}(t) = \sum_{i=0}^q w_{ji}^{(l)}(t) y_i^{(l-1)}(t) \quad (2.43)$$

โดยที่  $q$  คือจำนวนเซลล์ที่เชื่อมต่อนมายังเซลล์ที่  $j$  หรือจำนวนอินพุตของเซลล์ที่  $j$  นั้นเอง ส่วน  $w_{ji}^{(l)}(t)$  คือค่าน้ำหนักของเส้นเชื่อมระหว่างเซลล์ที่  $i$  ที่ชั้น  $(l-1)$  มายังเซลล์ที่  $j$  ที่ชั้น  $(l)$  ดังเช่น  $w_{23}^{(1)}(t)$   $w_{34}^{(2)}(t)$  และ  $w_{12}^{(3)}(t)$  ในรูปที่ 2.16 และ  $y_i^{(l-1)}(t)$  คือสัญญาณอินพุตของเซลล์ที่  $j$  ดังนั้นถ้าเซลล์ที่  $j$  เป็นเซลล์ที่ชั้นซ่อนชั้นแรก ( $l=1$ ) แสดงว่าเซลล์ที่  $i$  จะเป็นเซลล์ที่ชั้นอินพุต ( $l=0$ ) ดังนั้น

$$y_i^{(0)}(t) = x_i(t) \quad (2.44)$$

และถ้าเซลล์ที่  $j$  เป็นเซลล์ที่ชั้นเอาต์พุต ( $l=L$ ) แสดงว่าค่าเอาต์พุตที่ได้เป็นเอาต์พุตของโครงข่าย ซึ่งคือ

$$y_j^{(L)}(t) = o_j(t) \quad (2.45)$$

2. การผ่านย้อนกลับ (backward pass) หรือ การแพร่กระจายย้อนกลับ (back-propagation)

ในขั้นตอนนี้เมื่อได้ค่าเอาต์พุต  $o_j(t)$  ของทุกเซลล์ในชั้นเอาต์พุต ( $l=L$ ) แล้ว ทำให้สามารถคำนวณหาค่าความผิดพลาด  $e_j^{(L)}(t)$  ของทุกเซลล์ในชั้นเอาต์พุต ( $l=L$ ) ได้ดังสมการที่

2.19 และทำการคำนวณหาค่าเกรเดียนต์เฉพาะที่ของเซลล์ที่  $j$  ที่ชั้นเอาต์พุต ( $l=L$ ) ได้ดังนี้

$$\delta_j^{(L)}(t) = e_j^{(L)}(t) \varphi_j^{(L)'}(v_j^{(L)}(t)) \quad (2.46)$$

และทำการคำนวณหาค่าเกรเดียนต์เฉพาะที่ของเซลล์ที่  $j$  ที่ชั้นซ่อน ( $l>0$  และ  $l<L$ ) ได้ดังนี้

$$\delta_j^{(l)}(t) = \varphi_j^{(l)'}(v_j^{(l)}(t)) \sum_k \delta_k^{(l+1)}(t) w_{kj}^{(l+1)}(t) \quad (2.47)$$

จากนั้นทำการปรับค่าน้ำหนัก  $w_{ji}^{(l)}$  สำหรับทุกเส้นเชื่อมได้ดังนี้

$$w_{ji}^{(l)}(t+1) = w_{ji}^{(l)}(t) + \Delta w_{ji}^{(l)}(t) \quad (2.48)$$

$$\Delta w_{ji}^{(l)}(t) = \eta \delta_j^{(l)}(t) y_i^{(l-1)}(t) \quad (2.49)$$

และ  $\eta$  อัตราการเรียนรู้ (learning rate) และเช่นที่เคยอธิบายมาแล้วว่า ถ้าค่า  $\eta$  น้อย ค่า น้ำหนักที่ถูกปรับจะมีค่าน้อยด้วย และการปรับค่าจะค่อนข้างเป็นการปรับที่เรียบ และ อาจจะให้อัลกอริทึมลู่อืเข้าหาคำตอบได้ช้า แต่ถ้าให้ค่า  $\eta$  มากเกินไปจะทำให้การปรับค่า น้ำหนักเป็นแบบแกว่งกวัด (oscillate) ดังนั้นวิธีที่เพิ่มอัตราการเรียนรู้แต่หลีกเลี่ยงการ เกิดการแกว่งกวัด สามารถทำได้โดยรวมพจน์ที่มีอัตราโมเมนตัม (momentum rate) ด้วย ดังนั้นสมการที่ 2.49 จะกลายเป็น

$$\Delta w_{ji}^{(l)}(t) = \alpha \Delta w_{ji}^{(l)}(t-1) + \eta \delta_j^{(l)}(t) y_i^{(l-1)}(t) \quad (2.50ก)$$

$$\text{หรือ } \Delta w_{ji}^{(l)}(t) = \alpha [w_{ji}^{(l)}(t) - w_{ji}^{(l)}(t-1)] + \eta \delta_j^{(l)}(t) y_i^{(l-1)}(t) \quad (2.50ข)$$

โดยที่  $\alpha$  คืออัตราโมเมนตัม ส่วน  $\Delta w_{ji}^{(l)}(t-1)$  คือการเปลี่ยนแปลงค่าน้ำหนักที่ทำการ เชื่อมต่อจากเซลล์ที่  $i$  ที่ชั้น  $(l-1)$  ไปยังเซลล์ที่  $j$  ที่ชั้น  $(l)$  สำหรับการเข้าครั้ง  $t-1$  นั้นเอง ซึ่งจากสมการที่ 2.50ก และ 2.50ข ทำให้เห็นว่าถ้า  $\Delta w_{ji}^{(l)}(t-1)$  มีทิศทาง เดียวกับ  $\eta \delta_j^{(l)}(t) y_i^{(l-1)}(t)$  (มีเครื่องหมายตรงกัน) แสดงว่าเป็นการปรับค่าที่ไปใน ทิศทางเดียวกันทำให้  $\Delta w_{ji}^{(l)}(t)$  มีขนาดที่มากขึ้น แต่ถ้า  $\Delta w_{ji}^{(l)}(t-1)$  มีทิศทาง ตรงกันข้ามกับ  $\eta \delta_j^{(l)}(t) y_i^{(l-1)}(t)$  (มีเครื่องหมายต่างกัน) แสดงว่าเป็นการปรับค่าที่ ไม่ได้ไปในทิศทางเดียวกันทำให้  $\Delta w_{ji}^{(l)}(t)$  มีขนาดที่น้อยลง [Haykin09]

ตัวอย่างที่ 2.3 สมมติให้ฟังก์ชันการกระตุ้นของทุกเซลล์เป็นฟังก์ชันลอจิสติก (logistic function) ดังนี้

$$y_j^{(l)}(t) = \varphi_j^{(l)}(v_j^{(l)}(t)) = \frac{1}{1 + \exp(-v_j^{(l)}(t))} \quad (2.51)$$

ทำให้

$$\frac{\partial y_j^{(l)}(t)}{\partial v_j^{(l)}(t)} = \varphi_j^{(l)'}(v_j^{(l)}(t)) = \frac{\exp(-v_j^{(l)}(t))}{[1 + \exp(-v_j^{(l)}(t))]^2} \quad (2.52)$$

ซึ่งคือ

$$\varphi_j^{(l)'}(v_j^{(l)}(t)) = \frac{1}{1 + \exp(-v_j^{(l)}(t))} \left[ 1 - \frac{1}{1 + \exp(-v_j^{(l)}(t))} \right] \quad (2.53)$$

$$\varphi_j^{(l)'}(v_j^{(l)}(t)) = y_j^{(l)}(t) [1 - y_j^{(l)}(t)] \quad (2.54)$$

ดังนั้นสำหรับเซลล์  $j$  ที่ชั้นเอาต์พุต ( $l=L$ ) สมการที่ 2.46 กลายเป็น

$$\delta_j^{(L)}(t) = e_j^{(L)}(t) [o_j(t) [1 - o_j(t)]] \quad (2.55)$$

ส่วนเซลล์  $j$  ที่ชั้นซ่อน ( $l > 0$  และ  $l < L$ ) สมการที่ 2.47 กลายเป็น

$$\delta_j^{(l)}(t) = y_j^{(l)}(t) [1 - y_j^{(l)}(t)] \sum_k \delta_k^{(l+1)}(t) w_{kj}^{(l+1)}(t) \quad (2.56)$$

ตัวอย่างที่ 2.4 สมมติให้ฟังก์ชันการกระตุ้นของทุกเซลล์เป็นฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ (Hyperbolic tangent function) ดังนี้

$$y_j^{(l)}(t) = \phi_j^{(l)}(v_j^{(l)}(t)) = \tanh\left(\frac{v_j^{(l)}(t)}{2}\right) = \frac{2}{1 + \exp(-v_j^{(l)}(t))} - 1 \quad (2.57)$$

ทำให้

$$\frac{\partial y_j^{(l)}(t)}{\partial v_j^{(l)}(t)} = \phi_j^{(l)'}(v_j^{(l)}(t)) = \frac{2 \exp(-v_j^{(l)}(t))}{[1 + \exp(-v_j^{(l)}(t))]^2} \quad (2.58)$$

ซึ่งคือ

$$\phi_j^{(l)'}(v_j^{(l)}(t)) = \frac{2}{1 + \exp(-v_j^{(l)}(t))} \left[ 1 - \frac{1}{1 + \exp(-v_j^{(l)}(t))} \right] \quad (2.59)$$

$$\phi_j^{(l)'}(v_j^{(l)}(t)) = 2y_j^{(l)}(t) [1 - y_j^{(l)}(t)] \quad (2.60)$$

ดังนั้นสำหรับเซลล์  $j$  ที่ชั้นเอาต์พุต ( $l=L$ ) สมการที่ 2.46 กลายเป็น

$$\delta_j^{(L)}(t) = e_j^{(L)}(t) [2o_j(t) [1 - o_j(t)]] \quad (2.61)$$

ส่วนเซลล์  $j$  ที่ชั้นซ่อน ( $l>0$  และ  $l<L$ ) สมการที่ 2.47 กลายเป็น

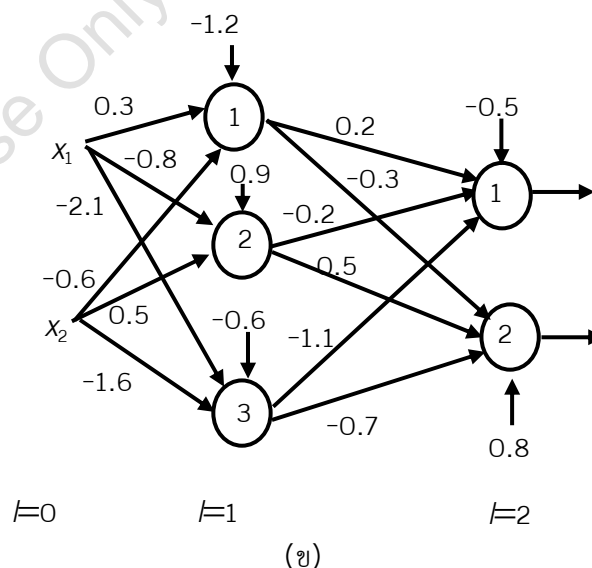
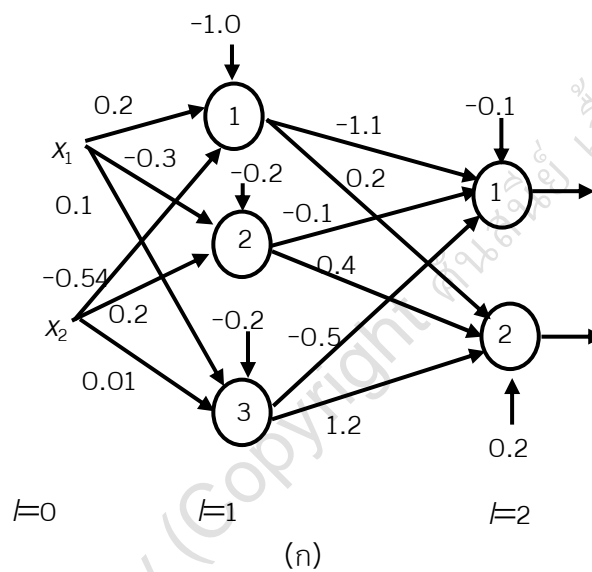
$$\delta_j^{(l)}(t) = 2y_j^{(l)}(t) [1 - y_j^{(l)}(t)] \sum_k \delta_k^{(l+1)}(t) w_{kj}^{(l+1)}(t) \quad (2.62)$$

เพื่อให้เกิดความเข้าใจที่ง่าย สำหรับชุดข้อมูลฝึกสอน  $\{[x(n), d(n)] | n=1, 2, \dots, N\}$  โดยที่เวกเตอร์อินพุต  $x$  เป็นเวกเตอร์  $p$  มิติ ส่วนเวกเตอร์เอาต์พุตที่ต้องการเป็นเวกเตอร์ที่มี  $q$  มิติ อัลกอริทึมสำหรับการเรียนการแพร่กระจายย้อนกลับ (back propagation algorithm) เป็นดังนี้

1. กำหนดค่าเริ่มต้น (Initialization) ของค่าน้ำหนักสำหรับทุกเส้นเชื่อมของทุกเซลล์ รวมทั้งไบแอส และกำหนดจำนวนเอพพรอก (epoch) (epoch = 1)
2. For  $t=1$  ถึง  $N$
3. สุ่มเวกเตอร์  $x$  จากชุดข้อมูลฝึกสอน โดยไม่ให้ซ้ำ
4. ทำขั้นตอนการผ่านไปข้างหน้า (forward pass) ด้วยสมการที่ 2.42 และ 2.43
5. ทำขั้นตอนการผ่านย้อนกลับ (backward pass) ด้วยสมการที่ 2.46 – 2.48 และ 2.49 หรือ 2.50ก หรือ 2.50ข
6. End For
7. ตั้งค่า epoch = epoch+1
8. คำนวณค่าเฉลี่ยของความผิดพลาดกำลังสองของทุกเวกเตอร์อินพุตในชุดข้อมูลฝึกสอน ด้วยสมการที่ 2.21

9. ถ้า  $\mathcal{E}_{av} < \mathcal{E}_1$  หรือ  $epoch > \mathcal{E}_2$  (โดยที่  $\mathcal{E}_1$  เป็นค่าคงที่บวกเล็กๆ และ  $\mathcal{E}_2$  เป็นจำนวนเอพพรอกที่มากที่สุด) ให้หยุดการเรียนรู้

ตัวอย่างที่ 2.5 การเรียนรู้ (train) โครงข่ายประสาทเทียมในรูปที่ 2.17(ก) และ 2.17(ข) เป็นโครงข่ายที่ค่าน้ำหนักทุกค่าเป็นค่าที่การทำซ้ำครั้งที่  $t-1$  และ  $t$  ตามลำดับ ซึ่งการเรียนรู้ที่ใช้ในตัวอย่างนี้เป็นการเรียนรู้ที่ใช้การแพร่กระจายย้อนกลับ โดยที่อินพุตและเอาต์พุตที่ต้องการของการทำซ้ำครั้งนี้ ( $t$ ) คือ  $\mathbf{x}(t) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  และ  $\mathbf{d}(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  จงคำนวณหา  $w_{32}^{(1)}(t+1)$  โดยที่อัตราการเรียนรู้เท่ากับ 0.2 และอัตราโมเมนตัมเป็น 0.1 โดยที่ฟังก์ชันการกระตุ้นเป็นฟังก์ชันลอจิสติกในสมการที่ 2.51



รูปที่ 2.17 โครงข่ายประสาทเทียมสำหรับตัวอย่างที่ 2.5 ในการทำซ้ำที่ (ก)  $t$  และ (ข)  $t-1$

ทำการหาค่าเอาต์พุตของแต่ละเซลล์จะได้

$$v_1^{(1)}(t) = \sum_{i=0}^2 w_{1i}^{(1)} x_i = -1.2 + (0.3)(1) + (-0.6)(1) = -1.5 \quad \text{ทำให้}$$

$$y_1^{(1)}(t) = \frac{1}{1 + \exp(-v_1^{(1)}(t))} = \frac{1}{1 + \exp(1.5)} = 0.1824 \quad \text{และ}$$

$$v_2^{(1)}(t) = \sum_{i=0}^2 w_{2i}^{(1)} x_i = 0.9 + (-0.8)(1) + (0.5)(1) = 0.6 \quad \text{ทำให้}$$

$$y_2^{(1)}(t) = \frac{1}{1 + \exp(-0.6)} = 0.6457 \quad \text{และ}$$

$$v_3^{(1)}(t) = \sum_{i=0}^2 w_{3i}^{(1)} x_i = -0.6 + (0.5)(1) + (-1.6)(1) = -1.7 \quad \text{ทำให้}$$

$$y_3^{(1)}(t) = \frac{1}{1 + \exp(1.7)} = 0.1545 \quad \text{ส่วน}$$

$$v_1^{(2)}(t) = \sum_{i=0}^3 w_{1i}^{(2)} y_i^{(1)} = -0.5 + (0.2)(0.1824) + (-0.2)(0.6457) + (-1.1)(0.1545) = -0.7626$$

$$\text{ทำให้ } y_1^{(2)}(t) = \frac{1}{1 + \exp(0.7626)} = 0.3181 \quad \text{ทำให้ } e_1^{(2)}(t) = 0 - 0.3181 = -0.3181 \quad \text{ในขณะที่}$$

$$\text{ที่ } v_2^{(2)}(t) = \sum_{i=0}^3 w_{2i}^{(2)} y_i^{(1)} = 0.8 + (-0.3)(0.1824) + (0.5)(0.6457) + (-0.7)(0.1545) = 0.96$$

$$\text{ทำให้ } y_2^{(2)}(t) = \frac{1}{1 + \exp(-0.96)} = 0.7231 \quad \text{ทำให้ } e_2^{(2)}(t) = 1 - 0.7231 = 0.2769 \quad \text{ดังนั้นค่า}$$

เกรเดียนต์เฉพาะที่ที่ชั้นเอาต์พุตตามสมการที่ 2.55 คือ

$$\delta_1^{(2)}(t) = e_1^{(2)}(t) [o_1(t) [1 - o_1(t)]] = (-0.3181) [0.3181(1 - 0.3181)] = -0.0690$$

$$\text{และ } \delta_2^{(2)}(t) = e_2^{(2)}(t) [o_2(t) [1 - o_2(t)]] = (0.2769) [0.7231(1 - 0.7231)] = 0.0554$$

$$\text{ดังนั้น } \delta_3^{(1)}(t) = y_3^{(1)}(t) [1 - y_3^{(1)}(t)] \sum_{k=1}^2 \delta_k^{(2)}(t) w_{kj}^{(2)}(t) \quad \text{ซึ่งคือ}$$

$$\delta_3^{(1)}(t) = 0.1545 [1 - 0.1545] [(-0.0690)(-1.1) + (0.0554)(-0.7)] = 0.0048$$

ถ้าต้องการปรับค่าน้ำหนักที่ชั้นเอาต์พุตสามารถทำได้ด้วยวิธีการเดียวกับการปรับค่า

$$\text{น้ำหนัก } w_{12}^{(2)}(t+1) \quad \text{นั่นคือ } \Delta w_{12}^{(2)}(t) = \alpha \Delta w_{12}^{(2)}(t-1) + \eta \delta_1^{(2)}(t) y_2^{(1)}(t) \quad \text{ซึ่งคือ}$$

$$\Delta w_{12}^{(2)}(t) = 0.1(-0.2 - (-0.1)) + 0.2(-0.0690)(0.6457) = -0.0189 \quad \text{ดังนั้น}$$

$$w_{12}^{(2)}(t+1) = w_{12}^{(2)}(t) + \Delta w_{12}^{(2)}(t) = -0.2 - 0.0189 = -0.2189$$

ส่วนการปรับค่าน้ำหนักที่ชั้นซ่อน สามารถทำได้ดังการปรับค่าน้ำหนัก  $w_{32}^{(1)}(t+1)$  นั้น คือ

$$\Delta w_{32}^{(1)}(t) = \alpha \Delta w_{32}^{(1)}(t-1) + \eta \delta_3^{(1)}(t) x_2(t) \quad \text{ซึ่งคือ}$$

$$\Delta w_{32}^{(1)}(t) = 0.1(-1.6 - 0.01) + 0.2(0.0048)(1) = -0.16 \quad \text{ดังนั้น}$$

$$w_{32}^{(1)}(t+1) = w_{32}^{(1)}(t) + \Delta w_{32}^{(1)}(t) = -1.6 - 0.16 = -1.76$$

### 2.2.3 การวางนัยทั่วไป (Generalization) และการตรวจสอบความสมเหตุสมผลแบบไขว้ (Cross Validation)

โดยปกติการแพร่กระจายย้อนกลับจะถูกใช้ในการเรียนรู้โครงข่ายประสาทเทียมจากชุดข้อมูลฝึกสอน โดยที่มีความหวังว่าโครงข่ายประสาทเทียมนั้นจะมีลักษณะเป็นการวางนัยทั่วไป (generalization) ซึ่งคือโครงข่ายนั้นจะทำการส่ง อินพุต-เอาต์พุต (input-output mapping) ได้ถูกต้องหรือใกล้เคียงค่าที่ถูกต้องที่สุด สำหรับชุดข้อมูลทดสอบ ที่ไม่ถูกใช้ในการเรียนรู้โครงข่าย ยกตัวอย่างเช่นโครงข่ายประสาทเทียมที่ถูกสร้างขึ้นมาเพื่อให้รู้จำใบหน้าของคน 10 คน โดยที่มี อินพุตเป็นลักษณะ (feature) ของใบหน้าของคนทั้ง 10 คน ในสถานการณ์ต่าง ส่วนเอาต์พุตก็คือ คลาสที่ระบุตัวบุคคล ถ้าโครงข่ายนี้เป็นโครงข่ายที่มีการวางนัยทั่วไป เมื่อเวลาผ่านไป เช่น 10 ปี คนทั้ง 10 อาจจะมีใบหน้าที่เปลี่ยนแปลงไป โครงข่ายนี้ก็ต้องสามารถแยกแยะบุคคลทั้ง 10 ได้ค่อนข้างดี แต่ถ้าโครงข่ายนี้ไม่มีการวางนัยทั่วไป ก็จะไม่สามารถแยกแยะได้ถูกต้องนั่นเอง

กระบวนการเรียนรู้ของโครงข่ายประสาทเทียมอาจจะถูกมองได้ว่าเป็นกระบวนการปรับเส้นโค้ง (curve fitting) ได้ โดยที่ตัวโครงข่ายถูกมองว่าเป็นการส่ง อินพุต-เอาต์พุต ดังนั้นถ้าโครงข่ายนั้นมีการวางนัยทั่วไปการส่งนี้จะมีลักษณะดังรูปที่ 2.18(ก) [Haykin09] นั่นคือโครงข่ายจะทำการสร้างเส้นจากคู่อินพุต-เอาต์พุต ดังรูปและเมื่อมีอินพุตที่มาจากชุดข้อมูลทดสอบ จะได้ผลลัพธ์ที่เกิดจากการประมาณค่าในช่วง (interpolate) ของโครงข่าย ดังนั้นโครงข่ายที่มีการวางนัยทั่วไป จะสามารถให้คำตอบที่ถูกต้องได้ถึงแม้ว่าอินพุต ที่เข้ามานั้นจะมีความแตกต่างจากอินพุต ที่มาจากชุดข้อมูลฝึกสอน แต่ถ้าโครงข่ายนั้นไม่มีการวางนัยทั่วไป หรือโครงข่ายนั้นทำการจำคู่อินพุต-เอาต์พุต หรือทำตัวเองให้เป็นตารางลुकอัป (look-up table) ในกรณีนี้เกิดจากการเรียนรู้มากเกินไป (overtrain) หรือการเหมาะสมเกินไป (overfit) ซึ่งอาจจะมีลักษณะของการปรับเส้นโค้งดังรูปที่ 2.18(ข) [Haykin09] ซึ่งในรูปนี้คำตอบของอินพุตจากชุดข้อมูลทดสอบเดียวกันกับรูปที่ 2.18(ก) จะเป็นคำตอบที่อยู่ห่างไกลจากคำตอบจริงมาก

ปัจจัยที่มีผลต่อการวางนัยทั่วไปคือ [Engelbrecht07]

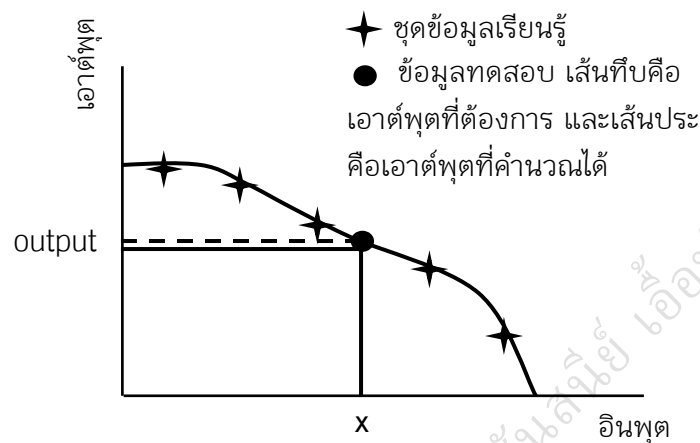
1. จำนวนของชุดข้อมูลฝึกสอน
2. สถาปัตยกรรม (architecture) ของโครงข่ายประสาทเทียม และ
3. ความซับซ้อนทางกายภาพ (physical complexity) ของปัญหา

โดยปกติแล้วเราไม่สามารถจัดการกับปัจจัยข้อที่ 3 ได้แต่ปัจจัยข้อที่ 1 และ 2 นั้นแท้จริงแล้วมีความสัมพันธ์กันซึ่งที่ว่าถ้าสถาปัตยกรรมของโครงข่ายถูกกำหนดเป็นสิ่งที่คงที่ไว้ สามารถกำหนดจำนวนของชุดข้อมูลเพื่อให้เกิดการวางนัยทั่วไปได้ และในทางกลับกันถ้าจำนวนของชุดข้อมูลเป็นค่าคงที่ สามารถสร้างโครงข่ายให้มีการวางนัยทั่วไปได้ ดังนั้นทั้งสองปัจจัยมีความสัมพันธ์กันดังนี้ [Haykin09]

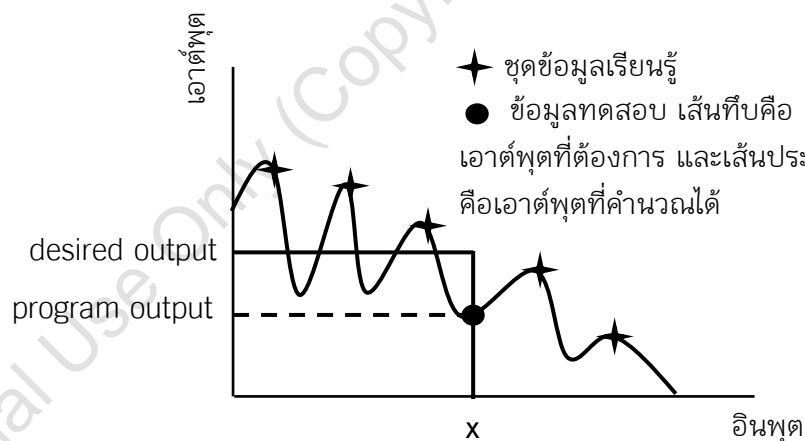


$$N = O\left(\frac{W}{\epsilon}\right) \quad (2.63)$$

โดยที่  $N$  คือจำนวนของชุดข้อมูลฝึกสอน ส่วน  $W$  คือจำนวนตัวแปรทั้งหมด หมายถึงจำนวนค่าน้ำหนักและไบแอส และ  $\epsilon$  คือค่าความผิดพลาด (error) ที่ต้องการ ในขณะที่  $O(\cdot)$  คืออันดับของปริมาณของตัวเลขที่อยู่ในวงเล็บ เช่นถ้ามีค่าความผิดพลาดเป็น 10 เปอร์เซ็นต์ จำนวนของชุดข้อมูลฝึกสอนควรจะเป็น 10 เท่าของจำนวนของตัวแปรที่ต้องการหาทั้งหมดเป็นต้น



(ก)



(ข)

รูปที่ 2.18 การปรับเส้นโค้ง (ก) ที่มีการวางนัยทั่วไป และ (ข) ที่มีการเหมาะสมเกินไป

การสร้างโครงข่ายให้มีการวางนัยทั่วไปนั้น อาจเป็นการมองได้อีกแบบหนึ่งว่าเป็นการเลือกโครงข่ายที่ดีที่สุด จากโครงข่ายที่มีทั้งหมดของชุดข้อมูลชุดนั้น ซึ่งการมองในลักษณะนี้สามารถทำได้จากการตรวจสอบความสมเหตุสมผลแบบไขว้ (cross validation) ซึ่งสามารถทำได้โดยการแบ่งชุดข้อมูลฝึกสอนออกเป็น

1. ซับเซตการประมาณ (estimation subset) ซึ่งเป็นซับเซต (subset) ของชุดข้อมูลฝึกสอน ที่ถูกใช้ในการเลือกแบบจำลอง (model) หรือใช้ในการฝึกสอน (train) โครงข่าย

2. ซับเซตการตรวจสอบความสมเหตุสมผล (validation subset) ซึ่งเป็นซับเซต (subset) ของชุดข้อมูลฝึกสอนที่ถูกใช้ในการตรวจสอบแบบจำลอง หรือประเมินสมรรถนะ (evaluate performance) ของแบบจำลองที่ได้จากซับเซตการประมาณ

โดยปกติการทำการตรวจสอบความสมเหตุสมผลแบบไขว้ จะทำในลักษณะของการตรวจสอบความสมเหตุสมผลแบบไขว้  $k$  โฟลด์ ( $k$ -fold cross validation) ซึ่งคือการแบ่งชุดข้อมูลแบบสุ่มออกเป็น  $k$  ชุดข้อมูลย่อย โดยที่ในแต่ละชุดข้อมูลมีจำนวนข้อมูลประมาณเท่าๆ กัน และให้ทำการฝึกสอนโครงข่ายด้วย ชุดข้อมูล  $k-1$  ชุด และให้ใช้ ชุดข้อมูลย่อยที่เหลือในการทดสอบโครงข่าย หลังโครงข่ายนั้นได้ถูกฝึกสอนจนเสร็จสิ้นแล้ว ทำกระบวนการนี้ซ้ำ  $k$  ครั้งด้วยชุดข้อมูลย่อยสำหรับทดสอบ ที่ไม่ซ้ำกัน [Kohavi95] ดังตัวอย่างการแบ่งชุดข้อมูลในรูปที่ 2.19 สำหรับการตรวจสอบความสมเหตุสมผลแบบไขว้ 4 โฟลด์ ในรูปนี้ชุดข้อมูลฝึกสอนถูกแบ่งเป็น 4 ส่วน โดยที่ส่วนที่ถูกระบายสีเทาคือซับเซตการตรวจสอบความสมเหตุสมผลสำหรับการฝึกสอนแต่ละครั้ง และส่วนที่ไม่มีการระบายสีเทาเป็นซับเซตการประมาณที่ใช้ในการฝึกสอนในแต่ละครั้ง ซึ่งในกรณีนี้เป็นการฝึกสอนทั้งหมด 4 ครั้งโดยที่ในการฝึกสอนแต่ละครั้งจะใช้ชุดข้อมูลที่ไม่ได้ถูกระบายสีเทา (ซับเซตการประมาณ) จนกระทั่งการฝึกสอนในครั้งสิ้นสุด ชุดข้อมูลที่ถูกระบายสีเทา (ซับเซตการตรวจสอบความสมเหตุสมผล) จะถูกใช้ในการทดสอบโครงข่ายนั้น



รูปที่ 2.19 การแบ่งชุดข้อมูลสำหรับการตรวจสอบความสมเหตุสมผลแบบไขว้ 4 โฟลด์

ข้อที่น่าสังเกตสำหรับการทำการตรวจสอบความสมเหตุสมผลแบบไขว้  $k$  โฟลด์ นั้นในการฝึกสอนแต่ละครั้งนั้นจำเป็นที่จะต้องมีการตั้งค่าเริ่มต้น และโครงสร้างของโครงข่ายให้เหมือนเดิมทั้ง  $k$  ครั้ง และหลังจากเสร็จสิ้นกระบวนการนี้ทั้ง  $k$  ครั้ง ให้ทำการคำนวณหาค่าความผิดพลาดเฉลี่ยของค่าความผิดพลาดที่ได้จาก ซับเซตการตรวจสอบความสมเหตุสมผลในแต่ละครั้ง เพื่อเป็นการบ่งบอกถึงสมรรถนะของโครงข่ายนั่นเอง

นอกจากการคำนวณหาค่าความผิดพลาด โดยปกติการรายงานผลสามารถรายงานในรูปของคอนฟิวนซ์เมทริกซ์ (confusion matrix) ซึ่งเป็นการรายงานดังนี้

$$\begin{array}{c}
 \text{calculated class} \\
 \begin{array}{cccc}
 & 1 & 2 & \cdots & C \\
 \text{desired class} : & \begin{array}{l} 1 \\ 2 \\ \vdots \\ C \end{array} & \begin{array}{l} a_{11} \\ a_{21} \\ \vdots \\ a_{C1} \end{array} & \begin{array}{l} a_{12} \\ a_{22} \\ \vdots \\ a_{C2} \end{array} & \begin{array}{l} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} & \begin{array}{l} a_{1C} \\ a_{2C} \\ \vdots \\ a_{CC} \end{array}
 \end{array}
 \end{array}
 \quad (2.64)$$

ซึ่งค่า  $a_{ij}$  คือจำนวนชุดข้อมูลที่มาจากคลาส  $i$  แต่ถูกโครงข่ายจัดไปอยู่คลาส  $j$  ดังนั้นจำนวนข้อมูลที่ถูกต้องกลุ่มผิดพลาดมีจำนวนเท่ากับผลรวมของ  $a_{ij}$  ที่  $i \neq j$  นั่นเอง

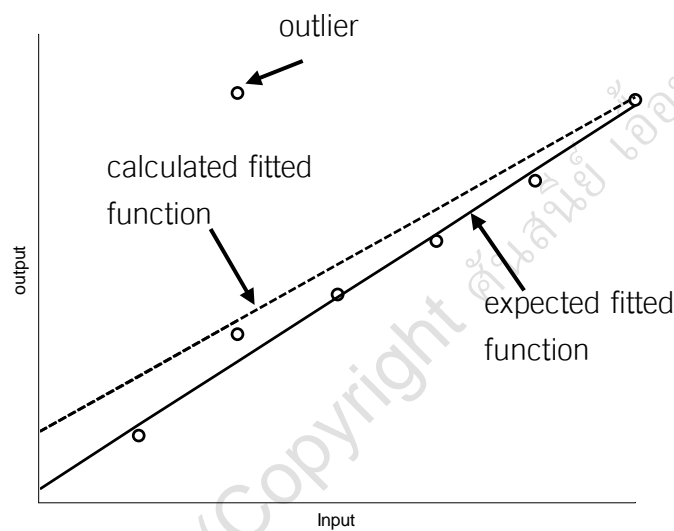
#### 2.2.4 แฟกเตอร์ของสมรรถนะ (Performance Factor)

ในหัวข้อนี้จะกล่าวถึงปัจจัยต่างๆ ที่มีผลต่อสมรรถนะของการทำงานของโครงข่ายประสาทเทียม โดยมีรายละเอียดดังนี้ [Engelbrecht07,Haykin09]

1. การเตรียมชุดข้อมูล (data preparation) เป็นการเตรียมข้อมูลให้อยู่ในรูปแบบที่เป็นที่ยอมรับของการฝึกโครงข่าย ดังต่อไปนี้

- ค่าที่หายไป (missing value) ถ้าจำนวนข้อมูลในการฝึกสอนมีเพียงพอให้ทำการจำกัดข้อมูลที่มีค่าลักษณะ (feature value) หายไป แต่ถ้ามีจำนวนข้อมูลไม่เพียงพอให้ทำการแทนที่ค่าลักษณะที่หายไปนั้นด้วยค่าเฉลี่ยของค่าลักษณะในมิติ นั้น เช่นค่าลักษณะในมิติที่ 2 ของข้อมูลที่ 10 หายไป ให้ทำการหาค่าเฉลี่ยของค่าลักษณะในมิติที่ 2 จากข้อมูลทั้งหมด และแทนที่ค่าลักษณะในมิติที่ 2 ของข้อมูลที่ 10 ด้วยค่าเฉลี่ยที่ได้ แต่ในบางครั้งอาจจะใช้ค่าลักษณะในมิติอื่นที่เกิดบ่อยที่สุดมาแทนที่ค่าลักษณะในมิติที่หายไปก็ได้ หรืออาจจะเพิ่มอินพุตพารามิเตอร์ที่บ่งบอกว่าข้อมูลนั้นมีค่าลักษณะที่หายไป และเมื่อทำการฝึกสอนเสร็จสิ้นแล้วสามารถระบุได้ว่าค่าลักษณะที่หายไปมีความสำคัญมากน้อยเพียงไร
- ค่าอินพุต (coding input value) โดยปกติแล้วการฝึกสอนโครงข่ายจะเป็นการฝึกสอนด้วยค่าที่เป็นตัวเลข แต่ในการเก็บข้อมูลบางครั้งค่าลักษณะที่ได้ไม่ใช่ตัวเลข เช่นเป็นค่าลักษณะที่ระบุถึง ใช่ หรือ ไม่ใช่ ดังนั้นจำเป็นต้องแปลงค่าลักษณะเหล่านั้นให้อยู่ในรูปของตัวเลขก่อน
- เอาต์ไลเออร์ (outlier) เป็นข้อมูลที่เบี่ยงเบนจากการกระจายของชุดข้อมูลมาก ดังแสดงในรูปที่ 2.20 และข้อมูลนี้จะทำให้การฝึกสอนโครงข่ายเบี่ยงเบนออกไปจากสิ่งที่ควรจะเป็นได้ หรือการวางนัยทั่วไปของการฝึกสอนลดลงได้ ดังนั้นโดยปกติจะทำการกำจัดเอาต์ไลเออร์ออกจากชุดข้อมูลฝึกสอนก่อน หรือทำให้ฟังก์ชันจุดประสงค์ (objective function) มีความเป็นทนทาน (robust) มากขึ้น
- การสเกล (scaling) และการทำให้เป็นบรรทัดฐาน (normalization) ในการทำให้สมรรถนะของโครงข่ายดีขึ้น จำเป็นต้องมีการสเกลข้อมูลให้อยู่ในโดเมนและเรนจ์ที่แอ็กทีฟ (active) ของฟังก์ชันการกระตุ้น (activation function) ทั้งนี้ค่าเอาต์พุตที่ต้องการของเซลล์ที่ชั้นเอาต์พุต จำเป็นที่จะต้องอยู่ในเรนจ์ของฟังก์ชันการกระตุ้น เช่นเอาต์พุตที่ต้องการที่เซลล์  $j$  ที่ชั้นเอาต์พุตมีค่าเท่ากับ  $d_j$  ควรจะมีการตั้งค่าออฟเซต  $\mathcal{E}$  เพื่อให้ค่าเอาต์พุตที่ต้องการอยู่ในเรนจ์ของฟังก์ชันการกระตุ้น หรืออยู่ห่างจากค่าลิมิตของเรนจ์ของฟังก์ชันการกระตุ้น นั่นคือสมมติให้ฟังก์ชันการกระตุ้นคือฟังก์ชันลอจิสติก (logistic function) ค่าลิมิตของฟังก์ชันนี้คือ 0 และ 1 ดังนั้นค่าเอาต์พุตที่ต้องการควรจะเป็น  $0+\mathcal{E}$  หรือ  $1-\mathcal{E}$  โดยปกติ  $\mathcal{E}$  จะมีค่าเท่ากับ 0.1 ดังนั้น  $d_j$  จะมีค่าเป็น 0.1 หรือ 0.9 ตัวอย่างเช่นต้องการสร้างโครงข่ายประสาทเทียมให้มีความสามารถในการแยกแยะข้อมูลออกเป็น 3 คลาส

ดังนั้นโครงข่ายนี้จะมีเซลล์ที่ชั้นเอาต์พุต 3 เซลล์ โดยที่ข้อมูล  $x_i$  ที่เข้ามาทำการฝึกสอนโครงข่าย ณ ขณะนี้อยู่คลาส 1 ค่าเอาต์พุตที่ต้องการจะเป็น 0.9 0.1 0.1 แต่ถ้าเป็นคลาส 2 จะเป็น 0.1 0.9 0.1 และถ้าเป็นคลาส 3 จะเป็น 0.1 0.1 0.9 สำหรับทั้ง 3 เซลล์ตามลำดับ แต่ถ้าฟังก์ชันการกระตุ้นเป็นฟังก์ชันไฮเพอร์โบลิคแทนเจนต์ (Hyperbolic tangent function) ที่มีค่าลิมิตเป็น  $-1$  และ  $1$  นั้น ค่าเอาต์พุตที่ต้องการจะเป็น  $-1+\varepsilon$  หรือ  $1-\varepsilon$  ซึ่งเช่นเดิมค่า  $\varepsilon$  จะมีค่าเท่ากับ 0.1 ดังนั้น  $d_j$  จะมีค่าเป็น  $-0.9$  หรือ  $0.9$  ดังนั้นด้วยตัวอย่างเดียวกันค่าเอาต์พุตที่ต้องการของ  $x_i$  ในทั้ง 3 กรณีคือ 0.9  $-0.9$   $-0.9$  สำหรับคลาส 1 แต่ถ้าเป็นคลาส 2 จะเป็น  $-0.9$  0.9  $-0.9$  และถ้าเป็นคลาส 3 จะเป็น  $-0.9$   $-0.9$  0.9 สำหรับทั้ง 3 เซลล์ตามลำดับ



รูปที่ 2.20 ผลกระทบของเอาต์ไลเออร์

ในทำนองเดียวกันสำหรับค่าลักษณะในแต่ละมิติของเวกเตอร์อินพุตนั้นควรจะถูกทำให้เป็นบรรทัดฐาน เพื่อให้ค่าลักษณะในมิติใดมิติหนึ่งมีผลต่อโครงข่ายมากกว่าค่าลักษณะในมิติอื่น [Theodoridis09] เช่นเวกเตอร์อินพุต  $x_i$  มี 5 มิติ และในแต่ละมิติมีขนาดอยู่ในช่วง  $10, 10^2, 10^6, 10^{-2}, 10^{-6}$  แ่่นอนว่าค่าลักษณะในมิติที่ 3 ต้องมีผลกับการปรับค่าน้ำหนักมากกว่าค่าในมิติที่ 5 ดังนั้นจำเป็นต้องทำให้เป็นบรรทัดฐาน ซึ่งมีกระบวนการดังนี้ สมมุติให้มีจำนวนข้อมูลที่ใช้ในการฝึกสอนเท่ากับ  $N$  และแต่ละเวกเตอร์มี  $p$  มิติ  $\left(x_i = [x_{i1}, x_{i2}, \dots, x_{ip}]^t\right)$  สำหรับการทำให้เป็นบรรทัดฐานของค่าลักษณะในมิติที่  $k$  คือ

$$\text{ค่าเฉลี่ยของค่าในมิติ } k \text{ คือ } \bar{x}_k = \frac{1}{N} \sum_{i=1}^N x_{ik} \text{ for } k = 1, 2, \dots, p \quad (2.65)$$

$$\text{ค่าความแปรปรวน (variance) ของค่าในมิติ } k \text{ คือ } \sigma_k^2 = \frac{1}{N-1} \sum_{i=1}^N (x_{ik} - \bar{x}_k)^2 \quad (2.66)$$

ค่าลักษณะในมิติ  $k$  ของเวกเตอร์  $\mathbf{x}_i$  หลังจากทำให้เป็นบรรทัดฐานคือ  $\hat{x}_{ik} = \frac{x_{ik} - \bar{x}_k}{\sigma_k}$  (2.67)

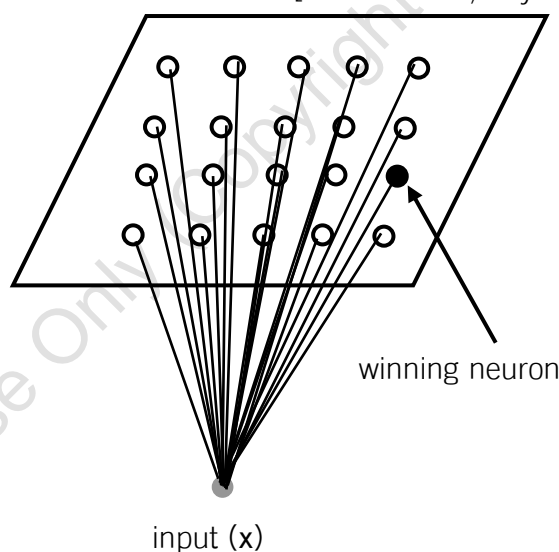
หรืออาจจะเป็นการทำให้เป็นบรรทัดฐานวิธีอื่นก็ได้เช่นกัน

- การเติมสิ่งรบกวน (noise injection) ในกรณีที่จำนวนข้อมูลในชุดข้อมูลฝึกสอนมีน้อยมาก สามารถเติมสิ่งรบกวนที่ถูกสร้างจากการแจกแจงปกติ (normal distribution) ที่มีค่าเฉลี่ยเป็น 0 และความแปรปรวนน้อย ให้เป็นส่วนหนึ่งของชุดข้อมูลฝึกสอนได้ และถ้าสิ่งรบกวนที่เติมเข้าไปอยู่รอบๆ เส้นขอบการตัดสินใจ (decision boundary) ทำให้สมรรถนะของโครงข่ายดีขึ้นอีกด้วย
  - การจัดการชุดข้อมูลฝึกสอน (training set manipulation) ในงานวิจัยหลายชิ้นได้มีความพยายามควบคุมลำดับของข้อมูลในการฝึกสอนโครงข่าย เช่นงานวิจัยของ Ohnishi และคณะ [Ohnishi90] ได้พัฒนาวิธีการที่เรียกว่าการนำเสนอแบบเลือกได้ (selective presentation) ซึ่งเป็นการแบ่งชุดข้อมูลออกเป็น 2 ชุดคือแบบตามแบบอย่าง (typical pattern) ซึ่งเป็นข้อมูลที่อยู่ไกลจากเส้นขอบการตัดสินใจ และแบบสับสน (confusing pattern) ซึ่งเป็นข้อมูลที่อยู่ใกล้กับเส้นขอบการตัดสินใจ ดังนั้นวิธีการนี้จะทำการส่งข้อมูลในแบบตามแบบอย่าง ให้กับการฝึกสอน สลับกับข้อมูลในแบบสับสน แต่ทั้งนี้ทั้งนั้นวิธีการนี้จำเป็นต้องมีข้อมูลเบื้องต้น (prior information) ก่อนถึงจะดำเนินการฝึกสอนตามนี้ได้ แต่ในการใช้งานจริง ไม่มีข้อมูลเบื้องต้นในการฝึกสอน
2. การตั้งค่าน้ำหนักเริ่มต้น (weight initialization) สำหรับกระบวนการตกลงของเกรเดียนต์ (gradient descent) นั้นค่าน้ำหนักเริ่มต้นมีผลต่อกฎการเรียนรู้แบบนีมาก โดยปกติแล้วค่าเริ่มต้นที่นิยมใช้กันคือการสุ่มค่าเริ่มต้นจากช่วง  $\left[ \frac{-1}{\sqrt{fanin}}, \frac{1}{\sqrt{fanin}} \right]$  โดยที่  $fanin$  คือจำนวนเส้นเชื่อมขาเข้ามายังเซลล์นั้น
  3. อัตราการเรียนรู้ (learning rate) และโมเมนตัม (momentum) โดยปกติแล้วอัตราการเรียนรู้จะควบคุมสเกลของการเรียนรู้ ถ้ามีค่าน้อยเกินไปการปรับค่าน้ำหนักจะเป็นไปที่ละน้อย แต่ถ้ามากเกินไปการปรับค่าน้ำหนักจะมีค่ามากในแต่ละครั้งและอาจจะทำให้แกว่งกวัด (oscillate) ได้ ดังนั้นโดยปกติการเลือกอัตราการเรียนรู้ที่ดีที่สุด สามารถทำได้โดยใช้การลองผิดลองถูก (trial and error) ผ่านการตรวจสอบความสมเหตุสมผลแบบไขว้ (cross validation) ในทำนองเดียวกันค่าโมเมนตัมเป็นค่าที่ช่วยในการถ่วงการเกิดการแกว่งกวัดจากอัตราการเรียนรู้ และการเลือกค่าโมเมนตัมที่ดีที่สุด ก็ทำได้เช่นเดียวกับการเลือกอัตราการเรียนรู้ที่ดีที่สุดเช่นกัน
  4. วิธีการหาค่าที่เหมาะสม (optimization method) กระบวนการตกลงของเกรเดียนต์ (gradient descent) เป็นเพียงหนึ่งในกระบวนการหาค่าที่เหมาะสมที่สุด ซึ่งในความเป็นจริงแล้วยังมีวิธีการอื่นอีกที่สามารถใช้ในการปรับค่าน้ำหนัก เพื่อหาค่าน้ำหนักที่ดีที่สุดได้ซึ่งจะไม่ขอก้าวในที่นี้
  5. การเลือกสถาปัตยกรรม (architecture) ดังคำกล่าวที่ว่าถ้ามีโครงข่ายหลายโครงข่ายที่เหมาะสม (fit) กับชุดข้อมูลฝึกสอนได้ดีเหมือนกัน แล้วโครงข่ายที่อยู่ในรูปโครงสร้างที่ง่ายที่สุด หรือโครงข่ายที่มีจำนวนน้ำหนักน้อยที่สุด จะเป็นโครงข่ายที่มีการวางนัยทั่วไปได้ดี

ที่สุด [Thodberg91] โดยปกติแล้วในการฝึกสอนโครงข่าย ถ้าโครงข่ายนั้นมีจำนวน  
น้ำหนักที่ต้องการมากเกินไป อาจจะทำให้โครงข่ายไม่มีการวางนัยทั่วไป แต่กลายเป็น  
การจำรูปแบบไปนั่นเอง  
นอกเหนือจากนี้ยังคงมีปัจจัยอื่นๆ อีกมากมายที่จะไม่ขอก้าวในที่นี้

## 2.3 การส่งการจัดระเบียบตนเอง (Self-Organizing Maps)

เปอร์เซปตรอนหลายชั้น (multi-layer perceptrons) ที่กล่าวถึงในหัวข้อที่แล้วนั้นมีการ  
ฝึกสอนที่เป็นการฝึกสอนแบบมีการควบคุม (supervised training) ซึ่งหมายถึงข้อมูลในชุดข้อมูล  
ฝึกสอนจะมีค่าเอาต์พุตที่ต้องการกำกับ ส่วนการส่งการจัดระเบียบตนเอง (self-organizing  
maps (SOMs)) หรือการส่งการจัดระเบียบลักษณะของตนเอง (self-organizing feature  
maps (SOFMs)) ในหัวข้อนี้ นั้นเป็นโครงข่ายประสาทเทียมที่มีการฝึกสอนแบบไม่มีการควบคุม  
(unsupervised training) หรือแต่ละข้อมูลในชุดข้อมูลฝึกสอนไม่มีคลาส หรือค่าเอาต์พุตที่  
ต้องการกำกับ และกระบวนการเรียนรู้จะเป็นการเรียนรู้ตามการเรียนรู้แบบแข่งขันกัน  
(competitive learning) โดยที่เซลล์เอาต์พุตที่ชนะการแข่งขันจะถูกเรียกว่า เซลล์ผู้ชนะได้ทั้งหมด  
(winner-takes-all neuron) หรือเซลล์ผู้ชนะ (winning neuron) ส่วนกระบวนการของการทำ  
เซลล์ผู้ชนะได้ทั้งหมดนั้น ยึดหลักการของเส้นเชื่อมการยับยั้งส่วนข้าง (lateral inhibitory  
connections) ซึ่งเป็นความคิดจาก Rosenblatt [Rosenblatt58, Haykin09]



รูปที่ 2.21 การส่งการจัดระเบียบตนเองตามแบบจำลองของ Kohonen

โครงสร้างของการส่งการจัดระเบียบตนเองนั้น แต่ละเซลล์จะถูกวางในแต่ละจุดของ  
แลตทิซ (lattice) ที่อาจจะเป็น 1 หรือ 2 มิติแลตทิซก็ได้ ซึ่งการส่งการจัดระเบียบตนเองนี้เป็นการ  
จัดการเรียงตัวของในการส่งทอพอโลยี (topological map) ของรูปแบบอินพุต (input pattern)  
โดยที่ตำแหน่งเชิงพื้นที่ (spatial location) หรือพิกัด (coordinate) ของเซลล์ในแลตทิซ (lattice)  
เป็นตัวบ่งชี้ถึง ลักษณะทางสถิติภายใน (intrinsic statistical feature) ที่อยู่ในรูปแบบ  
อินพุต และโดยปกติแล้วการส่งนี้จะไม่เป็นเชิงเส้น (nonlinear) ในส่วนของการพัฒนาการส่งการ

จัดระเบียบตนเองนั้นมีที่มา มาจากลักษณะเด่นจำแนก (distinct feature) ของสมองของมนุษย์นั่นเอง

กระบวนการการส่งการจัดระเบียบตนเองนี้ เป็นการสร้างการส่งภูมิลักษณะเทียม (artificial topographic map) โดยผ่านการเรียนรู้จากการจัดระเบียบตนเองในวิธีการตามแบบอย่างของเซลล์ประสาททางชีววิทยา (neurobiological) และแบบจำลองการส่งการจัดระเบียบตนเองที่เป็นที่นิยม เป็นแบบจำลองของ Kohonen [Haykin09] ดังแสดงในรูปที่ 2.21 ซึ่งในรูปนี้เป็นการส่งในลักษณะของอาร์เรย์  $4 \times 5$  ซึ่งโดยปกติการเชื่อมต่ออินพุตจะเชื่อมต่อไปยังทุกเซลล์ในการส่งเสมอ กระบวนการของ Kohonen นี้มีรากฐานมาจาก ตำแหน่งเชิงพื้นที่ของเซลล์เอาต์พุต (output neuron) ในการส่งภูมิลักษณะสมนัย (correspond to) กับโดเมนเฉพาะ (particular domain) หรือลักษณะ (feature) ของข้อมูล (data) จากปริภูมิอินพุต (input space)

อัลกอริทึมในการก่อเกิดของการส่งการจัดระเบียบนี้เริ่มจากการเริ่มต้นค่าน้ำหนักของเซลล์ในโครงข่าย ซึ่งสามารถทำได้ด้วยการสุ่ม หลังจากนั้น มี 3 กระบวนการ [Haykin09] ที่จำเป็นสำหรับการเรียนรู้ก็คือ

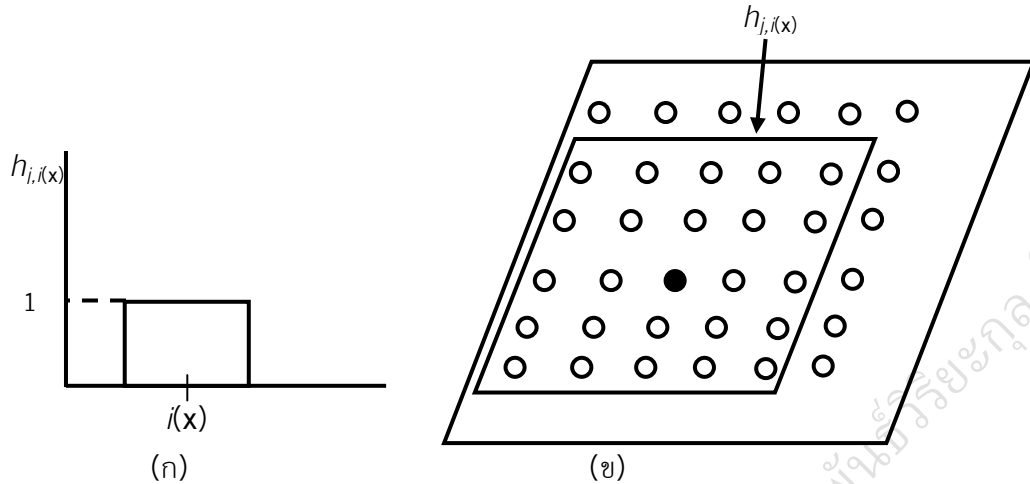
1. การแข่งขัน (competitive) เซลล์ในโครงข่ายทำการคำนวณหาค่าของฟังก์ชันดิสคริมิแนนต์ และเซลล์ที่มีค่านี้น้อยมากจะเป็นผู้ชนะ สำหรับการคำนวณหาค่าของฟังก์ชันดิสคริมิแนนต์นี้ทำได้โดย ให้เวกเตอร์อินพุต  $\mathbf{x} = [x_1, x_2, \dots, x_m]^t$  เป็นเวกเตอร์ที่มี  $m$  มิติ และให้เซลล์ที่  $j$  ในโครงข่ายมีเวกเตอร์น้ำหนัก  $\mathbf{w}_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^t$  ที่มีจำนวนมิติเท่ากับเวกเตอร์อินพุต สำหรับทุกเซลล์ที่  $j=1, 2, \dots, J$  (ในโครงข่ายมี  $J$  เซลล์) ดังนั้นในการหาเซลล์ที่มีค่าของฟังก์ชันดิสคริมิแนนต์ นั้นที่จริงเป็นการหาว่าเซลล์ใดมีค่าผลคูณภายใน (inner product  $(\mathbf{w}_j^t \mathbf{x})$ ) ซึ่งเทียบเท่ากับการหาระยะแบบยูคลิด (Euclidean distance) ที่สั้นที่สุดนั่นเอง ดังนั้นดัชนี (index) ของเซลล์ที่เป็นผู้ชนะ ( $i(\mathbf{x})$ ) คือ

$$i(\mathbf{x}) = \underset{j}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{w}_j\| \quad \text{for } j \in \mathcal{N} \quad (2.68)$$

โดยที่  $\mathcal{N}$  คือแลตทิซของเซลล์ในโครงข่าย และจากสมการที่ 2.68 นี้มีข้อสังเกตคือ ปริภูมิต่อเนื่องของอินพุตของมีรูปแบบที่มีการกระตุ้น จะถูกส่งไปยังปริภูมิที่ไม่ต่อเนื่องของเอาต์พุตของเซลล์ โดยกระบวนการแข่งขันระหว่างเซลล์ในโครงข่าย [Haykin09]

2. ความร่วมมือ (cooperation) เซลล์ผู้ชนะเป็นตัวกำหนดตำแหน่งเชิงพื้นที่ ของทอพอโลยีเพื่อนบ้าน (topological neighbor) ของเซลล์ที่ถูกกระตุ้น ซึ่งเป็นการให้พื้นฐานของความร่วมมือกันระหว่างเซลล์เพื่อนบ้าน โดยสามารถทำได้ดังนี้ ให้  $h_{j,i(\mathbf{x})}$  เป็นฟังก์ชันเพื่อนบ้าน (neighborhood function) รอบเซลล์ผู้ชนะ ( $i(\mathbf{x})$ ) โดยที่ในตอนเริ่มต้นจะเป็นฟังก์ชันที่มีขนาดใหญ่ เพื่อให้ทิศทางการปรับเวกเตอร์น้ำหนักของเซลล์ส่วนใหญ่ในแลตทิซมีสหสัมพันธ์กัน และมีขนาดลดลงเมื่อเวลาผ่านไปเพื่อเซลล์ที่มีสหสัมพันธ์กันเท่านั้นที่ถูกปรับเวกเตอร์น้ำหนัก [Haykin94] โดยปกติแล้วรูปร่างของฟังก์ชันนี้จะมีลักษณะคล้ายฟังก์ชันสี่เหลี่ยมดังรูปที่ 2.22(ก) ซึ่งเป็นฟังก์ชันเพื่อนบ้านในกรณีที่แลตทิซมี 1 มิติ แต่ถ้าแลตทิซเป็น 2 มิติ จะมีลักษณะดังรูปที่ 2.22(ข) ความหมายของฟังก์ชัน

เพื่อนบ้านรอบเซลล์ผู้ชนะในกรณีนี้คือ เฉพาะเซลล์ที่อยู่ในฟังก์ชันเพื่อนบ้านเท่านั้นที่มีค่า  $h_{j,i}(x)$  เท่ากับ 1 แต่เซลล์ที่อยู่ชานนอก  $h_{j,i}(x)$  จะมีค่าเป็น 0



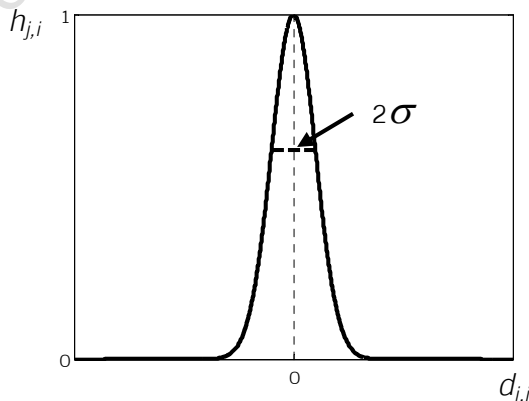
รูปที่ 2.22 ลักษณะของฟังก์ชันเพื่อนบ้านรอบเซลล์ผู้ชนะ  $i(x)$  สำหรับแลตทิซ (ก) 1 มิติ และ (ข) 2 มิติ

แต่ในสมองของมนุษย์นั้นเซลล์ที่ถูกจุด (firing) จะกระตุ้นเซลล์ที่อยู่ใกล้เคียงมากกว่าเซลล์ที่อยู่ห่างออกไป ดังนั้นฟังก์ชันเพื่อนบ้านควรจะมีลักษณะดังนี้ [Haykin09]

- $h_{j,i}(x)$  มีลักษณะสมมาตรรอบจุดที่สูงที่สุด หรือที่ระยะส่วนข้าง (lateral distance) ระหว่างเซลล์ผู้ชนะ  $i$  และเซลล์ที่ถูกกระตุ้น  $j$  ( $d_{j,i}$ ) มีค่าเท่ากับ 0
  - แอมพลิจูดของ  $h_{j,i}(x)$  จะลดลงทางเดียว (monotonic decreasing) ด้วยระยะ  $d_{j,i}$  ที่เพิ่มขึ้น จนกระทั่งมีค่าเข้าใกล้ 0 ที่  $d_{j,i}$  เข้าใกล้ค่าอนันต์
- ดังนั้นฟังก์ชันที่นิยมใช้เป็นฟังก์ชันเพื่อนบ้าน อีกฟังก์ชันหนึ่งคือ ฟังก์ชันเกาส์เซียน คือ

$$h_{j,i}(x) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \text{ for } j \in \mathcal{N} \quad (2.69)$$

โดยที่  $\sigma$  จะเป็นตัวกำหนดขนาดความกว้างประสิทธิผลของการกระตุ้น ดังรูปที่ 2.23



รูปที่ 2.23 ฟังก์ชันเพื่อนบ้านเกาส์เซียน

ส่วนค่าระยะ [Haykin09] ระหว่างเซลล์ผู้ชนะ  $i$  และเซลล์ที่ถูกกระตุ้น  $j$  นั้น ในกรณีของแลตทิซ 1 มิติ จะมีค่าเท่ากับ

$$d_{j,i} = |j - i| \quad (2.70n)$$



สำหรับแลตทิซ 2 มิติ คือ 
$$d_{j,i}^2 = \|r_j - r_i\|^2 \quad (2.70ข)$$

โดยที่  $r_i$  และ  $r_j$  คือเวกเตอร์ของตำแหน่งของเซลล์ที่  $i$  และ  $j$  ตามลำดับ และเช่นเดิม ฟังก์ชันเพื่อนบ้านจะมีขนาดที่ลดลงเมื่อเวลาผ่านไปดังนั้นโดยปกติแล้ว  $\sigma$  จะแปรผกผันกับเวลา  $n$  (หรือจำนวนครั้งของการทำซ้ำที่  $n$  ( $n^{\text{th}}$  iteration)) ซึ่งอาจจะเป็น

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \quad \text{for } n = 0, 1, 2, \dots, \quad (2.71)$$

โดยที่  $\sigma_0$  คือค่าตั้งต้นของ  $\sigma$  ส่วน  $\tau_1$  เป็นค่าคงที่ของเวลา (time constant) ซึ่งเป็นค่าที่ผู้ใช้เป็นคนกำหนด ดังนั้นฟังก์ชันเพื่อนบ้านในสมการที่ 2.69 จะกลายเป็น [Haykin09]

$$h_{j,i}(x)(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right) \quad \text{for } n = 0, 1, 2, \dots, \quad (2.72)$$

3. การปรับตัวของจุดประสานประสาท (synaptic adaptation) เป็นกระบวนการที่ทำให้เซลล์ที่ถูกกระตุ้นมีการเพิ่มค่าของฟังก์ชันดิสคริมิแนนต์ โดยให้สัมพันธ์กับรูปแบบอินพุต โดยผ่านการปรับค่าน้ำหนักของจุดประสานประสาท ซึ่งจากสัจพจน์ของการเรียนรู้ของ Hebb (Hebb's postulate of learning) นั้น ค่าน้ำหนักของจุดประสานประสาทจะเพิ่มขึ้นพร้อมกับการเกิด กิจกรรมก่อน และหลังของจุดประสานประสาท (presynaptic and post synaptic activities) แต่ในกรณีนี้เป็นการเรียนรู้แบบไม่มีการควบคุม (unsupervised learning) ทำให้ไม่เป็นไปตามสมมติฐานของ Hebbian เนื่องจากการปรับค่าน้ำหนักเป็นไปทางเดียว ซึ่งอาจจะทำให้ค่าน้ำหนักเกิดการอิ่มตัว (saturation) ได้ ดังนั้นจึงทำการปรับสมมติฐานของ Hebbian โดยการเพิ่มเทอมฟอร์เก็ตติง (forgetting term ( $g(y_j)w_j$ )) ที่  $w_j$  เป็นเวกเตอร์น้ำหนักของเซลล์ที่  $j$  และ  $g(y_j)$  เป็นฟังก์ชันค่าบวกของการตอบสนอง  $y_j$  ซึ่งมีข้อจำกัดที่ว่า เทอมของค่าคงที่ในการกระจายตัวอนุกรมของ Taylor (Taylor series expansion) ของฟังก์ชัน  $g(y_j)$  เป็น 0 ดังนั้น

$$g(y_j)=0 \quad \text{for } y_j=0 \quad (2.73)$$

ดังนั้นการเปลี่ยนแปลงของเวกเตอร์น้ำหนักของเซลล์  $j$  ในแลตทิซคือ

$$\Delta w_j = \eta y_j x - g(y_j) w_j \quad \text{for } j \in \mathcal{A} \quad (2.74)$$

โดยที่  $\eta$  คืออัตราการเรียนรู้ (learning rate) เทอมแรกในสมการที่ 2.74 นั้นเป็นเทอมของ Hebbian ส่วนเทอมที่สองเป็นเทอมฟอร์เก็ตติง และเพื่อให้สมการที่ 2.73 เป็นจริง  $g(y_j)$  ควรจะเป็นฟังก์ชันเชิงเส้น ดังนั้น

$$g(y_j) = \eta y_j \quad (2.75)$$

และ 
$$y_j = h_{j,i}(x) \quad (2.76)$$

ดังนั้นสมการที่ 2.74 คือ 
$$\Delta w_j = \eta h_{j,i}(x) (x - w_j) \quad \text{for } j \in \mathcal{A} \quad (2.77)$$

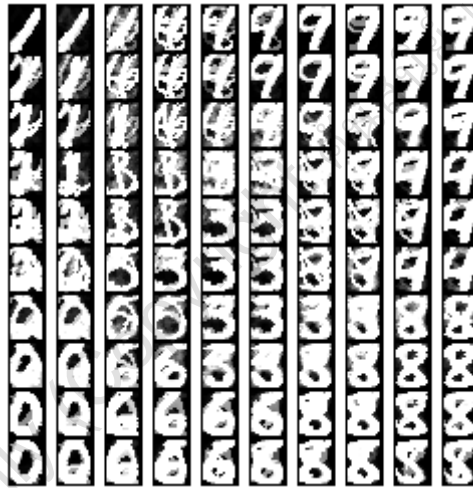
สำหรับฟังก์ชันเพื่อนบ้านที่เป็นฟังก์ชันสี่เหลี่ยม สมการที่ 2.77 คือ [Haykin94]

$$\Delta w_j = \begin{cases} \eta (x - w_j) & \text{if } j \text{ is inside neighborhood function} \\ 0 & \text{if } j \text{ is outside neighborhood function} \end{cases} \quad \text{for } j \in \mathcal{A} \quad (2.78)$$

แต่ถ้า  $h_{j,i}(x)$  เป็นฟังก์ชันเกาส์เซียน ให้ทำการแทนค่า  $h_{j,i}(x)$  ในสมการที่ 2.77 ด้วยสมการที่ 2.72 ได้เลย ดังนั้นเวกเตอร์น้ำหนักในการทำซ้ำที่  $t+1$  คือ [Haykin94, Haykin09]

$$w_j(t+1) = w_j + \eta(t) h_{j,i}(x)(t) (x(t) - w_j(t)) \quad \text{for } j \in \mathcal{N} \quad (2.79)$$

ซึ่งการปรับค่านี้ทำกับทุกเซลล์ในแลตทิซที่อยู่ข้างในทอพอโลยีเพื่อนบ้านของเซลล์ผู้ชนะ  $i$  และสมการที่ 2.79 นี้จะทำให้การเคลื่อนที่ของเวกเตอร์น้ำหนัก  $w_j$  ของเซลล์ผู้ชนะ  $i$  เข้าใกล้เวกเตอร์อินพุต  $x$  และเมื่อมีสิ่งอินพุตในชุดข้อมูลฝึกสอนเข้ามาฝึกสอนระบบเรื่อยๆ จะทำให้ค่าน้ำหนักของจุดประสานประสาทเป็นไปตาม การกระจายของเวกเตอร์อินพุต และการปรับในเพื่อนบ้านนั่นเอง และอัลกอริทึมนี้จะทำการเรียงลำดับทอพอโลยี (topology ordering) ของการส่งลักษณะ (feature map) จากปริภูมิอินพุต ในทำนองที่ว่าเซลล์ที่อยู่ใกล้กันจะค่อนข้างมีเวกเตอร์น้ำหนักที่คล้ายกัน [Haykin09] ดังเช่นรูปที่ 2.24 เป็นรูปของเวกเตอร์น้ำหนักของแลตทิซ 2 มิติ ขนาด  $10 \times 10$  หลังจากที่ถูกฝึกสอนด้วยตัวเลข 0 – 9 จะเห็นว่า เซลล์ที่อยู่ในกลุ่มซ้ายบน จะมีรูปร่างคล้ายกับตัวเลข 1 ในขณะที่กลุ่มซ้ายล่างจะเป็น 0 หรือขวาบนและขวาล่าง เป็น 9 และ 8 ตามลำดับ



รูปที่ 2.24 เวกเตอร์น้ำหนักของแลตทิซ 2 มิติ ขนาด  $10 \times 10$  หลังจากที่ถูกฝึกสอนด้วยตัวเลข 0 – 9

โดยปกติแล้วอัตราการเรียนรู้จะเป็นค่าคงที่ แต่ในบางครั้งอัตราการเรียนรู้สามารถเปลี่ยนแปลงค่าได้ตามเวลาดังเช่น

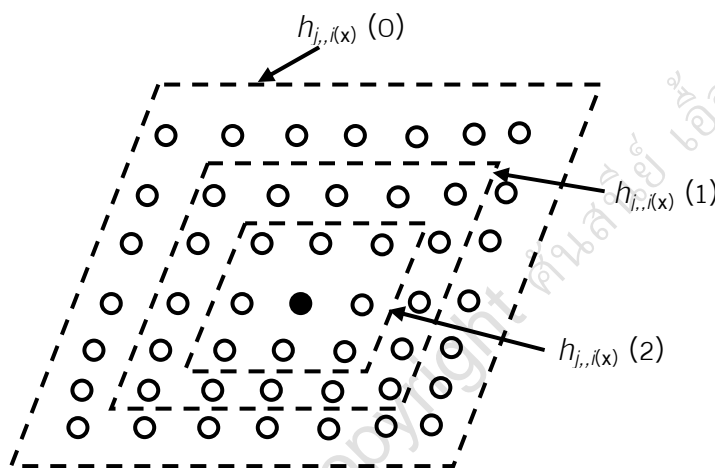
$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \quad \text{for } n = 0, 1, 2, \dots, \quad (2.80)$$

โดยที่  $\tau_2$  เป็นค่าคงที่ของเวลา (time constant) อีกค่าหนึ่ง

จากที่กล่าวมาข้างต้นจะเห็นได้ว่าอัลกอริทึม SOM นั้นเริ่มต้นจากการตั้งค่าเริ่มต้นที่ไม่มีรูปแบบหรือการจัดระเบียบใดๆ แต่เมื่อเวลาผ่านไปอัลกอริทึมจะทำการจัดระเบียบตัวแทนของรูปแบบที่ได้จากปริภูมิอินพุต ผ่านทางพารามิเตอร์ที่ถูกเลือกที่เหมาะสม และการปรับเวกเตอร์น้ำหนักตามสมการที่ 2.79 ซึ่งในกระบวนการนี้สามารถแบ่งออกเป็น 2 เฟส (phase) ดังนี้

- เฟสการจัดระเบียบตนเอง (self-organizing) หรือการจัดอันดับ (ordering) การฝึกสอนในเฟสนี้ เป็นการฝึกสอนโครงข่ายโดยการส่งเวกเตอร์อินพุตให้กับโครงข่ายและ

ทำการแข่งขัน ความร่วมมือ และการปรับตัวของจุดประสานประสาท สำหรับเวกเตอร์อินพุตนี้ และทำกระบวนการซ้ำสำหรับเวกเตอร์อื่นๆ ในชุดข้อมูลฝึกสอน โดยเป็นการทำซ้ำถึง 1,000 ครั้ง (1,000 iteration) ของอัลกอริทึม SOM หรืออาจจะมากกว่า 1,000 ครั้งก็ได้ โดยที่ค่าอัตราการเรียนรู้ตั้งต้น อาจจะมีค่าเป็น 0.1 และทำการเปลี่ยนค่าอัตราการเรียนรู้ตามสมการที่ 2.80 โดยที่  $\tau_2$  อาจจะมีค่าเท่ากับ 1,000 เพื่อให้อัตราการเรียนรู้มีค่าลดลงตามเวลา ซึ่งเวลาในที่นี้หมายถึงเอพพรอก (epoch) ตามที่เคยอธิบายไว้แล้วในหัวข้อ 2.2.2 แต่ค่าอัตราการเรียนรู้นี้ต้องไม่มีค่าลดลงจนกระทั่งเป็น 0 ส่วนฟังก์ชันเพื่อนบ้าน จะถูกเริ่มต้นจากทุกเซลล์เป็นเซลล์ที่อยู่ในฟังก์ชันเพื่อนบ้าน โดยมีเซลล์ผู้ชนะเป็นศูนย์กลาง และฟังก์ชันนี้มีขนาดลดลงตามเวลา นั่นคือถ้าฟังก์ชันเพื่อนบ้านเป็นฟังก์ชันสี่เหลี่ยมดังรูป 2.22(ข) แล้วจะตั้งต้นด้วยทุกเซลล์อยู่ในสี่เหลี่ยม และขนาดของสี่เหลี่ยมจะลดลงทุกด้านด้านละ 1 ดังรูปที่ 2.25



รูปที่ 2.25 ฟังก์ชันเพื่อนบ้านที่เป็นสี่เหลี่ยม ที่ขนาดต่างกัน เซลล์ที่เป็นจุดสีดำคือเซลล์ผู้ชนะ

แต่ถ้าเป็นฟังก์ชันเกาส์เซียนตามสมการที่ 2.72 จะมีการลดขนาดด้วย  $\sigma(n)$  ด้วยสมการที่ 2.71 โดยการกำหนดค่าเริ่มต้น  $\sigma_0$  ให้มีขนาดเท่ากับรัศมีของแลตทิซ (ทำให้ทุกเซลล์ถูกปรับค่าเวกเตอร์น้ำหนักในช่วงแรก) และลดค่า  $\sigma(n)$  โดยการกำหนดให้  $\tau_1$  มีค่าเป็น

$$\tau_1 = \frac{1000}{\log \sigma_0} \quad (2.81)$$

- เฟสการลู่เข้า (convergence phase) เป็นเฟสที่ทำการปรับที่ละเอียด (fine-tuning) ของการส่งลักษณะ (feature map) เพื่อความถูกต้องยิ่งขึ้นสำหรับปริภูมิอินพุตในเชิงปริมาณทางสถิติ โดยปกติแล้วจำนวนการทำซ้ำในเฟสนี้จะอยู่ที่ อย่างน้อย 500 เท่าของจำนวนของเซลล์ในโครงข่าย โดยที่อัตราการเรียนรู้จะเป็นค่าคงที่และมีค่าน้อย อยู่ในช่วงของ 0.01 และต้องไม่มีค่าเป็น 0 ส่วนฟังก์ชันเพื่อนบ้านจะเป็นฟังก์ชันที่มีเฉพาะเซลล์ที่ใกล้ที่สุดของเซลล์ผู้ชนะเท่านั้น นั่นคือลดลงจนเป็น 1 หรือ 0 (ในกรณีที่มีเฉพาะเซลล์ผู้ชนะเท่านั้น)

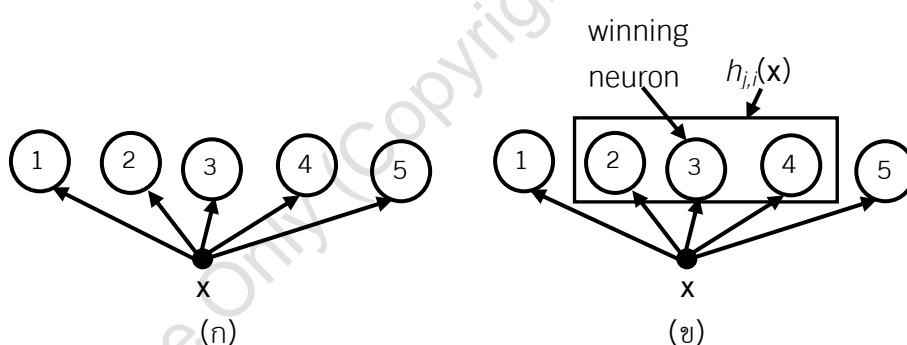
จากที่กล่าวมาข้างต้น สามารถสรุปอัลกอริทึม SOM ได้ดังนี้ [Haykin09]

1. การกำหนดค่าเริ่มต้น (Initialization): กำหนดค่าตัวต้นของเวกเตอร์น้ำหนักของทุกเซลล์ในแลตทิซ  $w_j(0)$  ให้แตกต่างกันสำหรับ  $j \in \mathcal{N}$  โดยที่อาจสุ่มให้มีค่าน้อย หรืออาจจะสุ่มเลือกจากเวกเตอร์อินพุตในชุดข้อมูลฝึกสอนที่มีเวกเตอร์อินพุต  $N$  เวกเตอร์  $\{x_i\}_{i=1}^N$  ซึ่งในกรณีนี้จะมีข้อดีคือการส่งเริ่มต้น (initial map) จะอยู่ในเรนจ์ของการส่งสุดท้าย (final map)
2. การชักตัวอย่าง (sampling): ชักตัวอย่าง  $x$  จากชุดข้อมูลฝึกสอน
3. การจับคู่ความเหมือน (similarity matching): หาเซลล์ผู้ชนะโดยใช้สมการที่ 2.68
4. การปรับค่า (updating): ปรับเวกเตอร์น้ำหนักตามสมการที่ 2.79
5. ทำซ้ำ (iterate): กลับไปทำขั้นตอนที่ 2 จนกระทั่งไม่มีการเปลี่ยนแปลงของการส่งลักษณะ (feature map) อีก

ตัวอย่างที่ 2.6 สมมติให้แลตทิซมีลักษณะดังรูปที่ 2.26(ก) และเวกเตอร์น้ำหนักดังนี้

$$w_1 = \begin{bmatrix} 0.5 \\ 0.9 \end{bmatrix}, w_2 = \begin{bmatrix} -0.3 \\ -0.1 \end{bmatrix}, w_3 = \begin{bmatrix} -0.9 \\ 0.9 \end{bmatrix}, w_4 = \begin{bmatrix} 0.3 \\ -0.6 \end{bmatrix}, \text{ และ } w_5 = \begin{bmatrix} 1.3 \\ -1.6 \end{bmatrix}$$

สมมติให้  $h_{ji}(x)$  เป็นฟังก์ชันเพื่อนบ้านรอบเซลล์ผู้ชนะ ที่มีรัศมีเป็น 1 และให้อัตราการเรียนรู้มีค่าเป็น 0.1 จงปรับเวกเตอร์น้ำหนักเมื่อมีเวกเตอร์อินพุต เป็น  $x = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$



รูปที่ 2.26 (ก) แลตทิซ และ (ข) เซลล์ผู้ชนะและฟังก์ชันเพื่อนบ้าน สำหรับตัวอย่างที่ 2.6

คำนวณหาระยะยูคลิด  $d = \sqrt{(x_1 - w_1)^2 + (x_2 - w_2)^2}$  ระหว่างเวกเตอร์อินพุตและเวกเตอร์น้ำหนักได้  $d(1) = 1.5033$ ,  $d(2) = 1.3038$ ,  $d(3) = 0.1414$ ,  $d(4) = 2.0616$ ,  $d(5) = 3.4713$  ทำให้เซลล์ผู้ชนะคือเซลล์ที่ 3 ( $i(x) = 3$ ) เมื่อฟังก์ชันเพื่อนบ้านเป็นสี่เหลี่ยมที่มีรัศมีเป็น 1 ดังรูปที่ 2.26(ข) ทำให้เซลล์ที่ถูกปรับเวกเตอร์น้ำหนักนอกเหนือจากเซลล์ที่ 3 คือ เซลล์ที่ 2 และ 4 นั่นเอง

$$\text{ดังนั้นจากสมการที่ 2.79 จะได้ว่า } w_2 = \begin{bmatrix} -0.3 \\ -0.1 \end{bmatrix} + 0.1 \left( \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} -0.3 \\ -0.1 \end{bmatrix} \right) = \begin{bmatrix} -0.37 \\ 0.01 \end{bmatrix},$$

$$w_3 = \begin{bmatrix} -0.9 \\ 0.9 \end{bmatrix} + 0.1 \left( \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} -0.9 \\ 0.9 \end{bmatrix} \right) = \begin{bmatrix} -0.91 \\ 0.91 \end{bmatrix}$$

$$\text{และ } w_4 = \begin{bmatrix} 0.3 \\ -0.6 \end{bmatrix} + 0.1 \left( \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.3 \\ -0.6 \end{bmatrix} \right) = \begin{bmatrix} 0.17 \\ -0.44 \end{bmatrix}$$

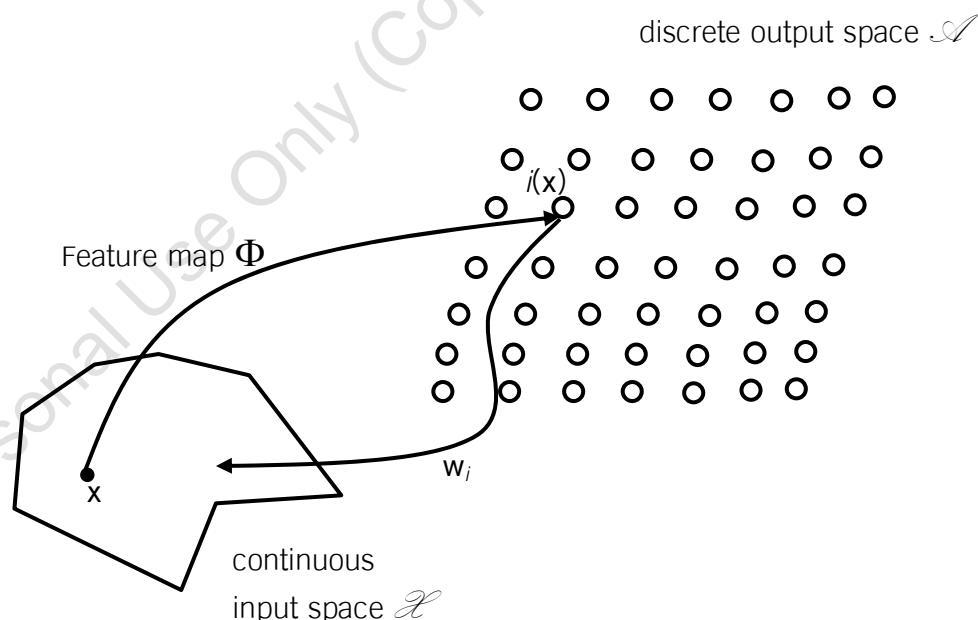
ส่วน  $w_1$  และ  $w_5$  มีค่าคงเดิมคือ  $w_1 = \begin{bmatrix} 0.5 \\ 0.9 \end{bmatrix}$  และ  $w_5 = \begin{bmatrix} 1.3 \\ -1.6 \end{bmatrix}$  นั่นเอง

เมื่ออัลกอริทึม SOM สู้เข้าแล้ว การส่งลักษณะ (feature map ( $\Phi$ )) ที่ได้ จะเป็นการแปลงไม่เชิงเส้น (nonlinear transform) ของปริภูมิอินพุต  $\mathcal{X}$  ไปยังปริภูมิเอาต์พุต  $\mathcal{A}$  หรือ

$$\Phi: \mathcal{X} \rightarrow \mathcal{A} \quad (2.82)$$

ซึ่งหมายความว่า เมื่อส่งเวกเตอร์อินพุตให้กับอัลกอริทึม SOM แล้วเซลล์ที่มีเวกเตอร์น้ำหนักใกล้เคียงกับเวกเตอร์อินพุต  $x$  มากที่สุดจะถูกเลือกเป็นเซลล์ผู้ชนะ ในปริภูมิเอาต์พุต  $\mathcal{A}$  และเวกเตอร์น้ำหนัก  $w_i$  ของเซลล์ผู้ชนะ  $i(x)$  จะถูกมองว่าเป็นตัวชี้ (pointer) สำหรับเซลล์นั้นๆ ในปริภูมิอินพุต  $\mathcal{X}$  ซึ่งสรุปดังรูปที่ 2.27 ได้ว่า

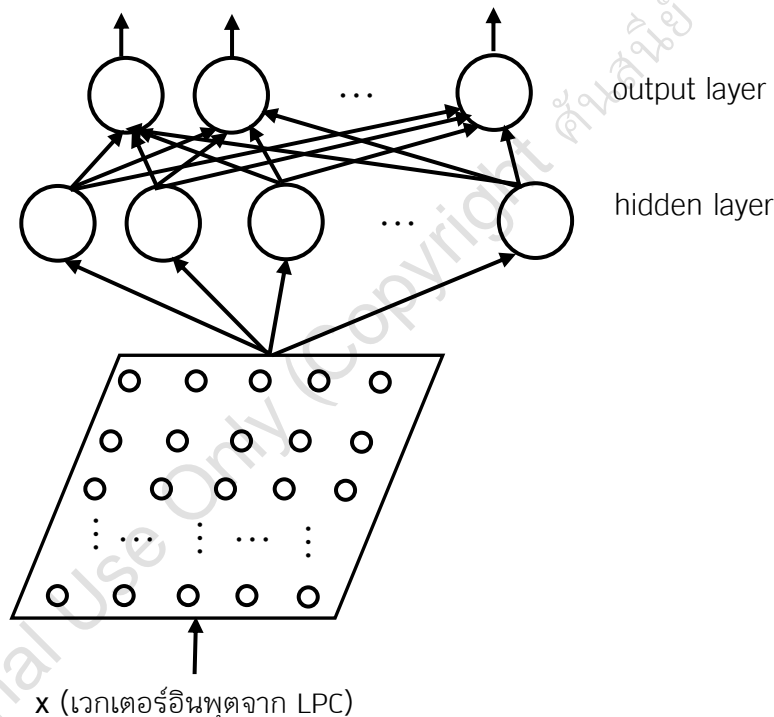
- การฉาย (projection) จากปริภูมิต่อเนื่องอินพุต (continuous input space ( $\mathcal{X}$ )) ไปยังปริภูมิไม่ต่อเนื่องเอาต์พุต (discrete output space ( $\mathcal{A}$ )) เป็นการหาเซลล์ผู้ชนะในแลตทิซโดยหาจากเซลล์ที่มีเวกเตอร์น้ำหนักที่ใกล้เคียงกับเวกเตอร์อินพุตนั้นที่สุด ซึ่งกรณีนี้ใช้ได้ในการหาเอาต์พุตของเวกเตอร์อินพุตในชุดข้อทดสอบได้ด้วย
- เป็นตัวชี้ (pointer) จากปริภูมิเอาต์พุตไปยังปริภูมิอินพุต ซึ่งเวกเตอร์น้ำหนักของเซลล์ผู้ชนะจะบ่งชี้จุดในปริภูมิอินพุต ซึ่งเปรียบเหมือนภาพสะท้อนของเซลล์ผู้ชนะในปริภูมินี้ ในกรณีนี้เป็นการทำซ้ำของการปรับค่าเวกเตอร์น้ำหนักในอัลกอริทึมนั่นเอง



รูปที่ 2.27 ความสัมพันธ์ระหว่างการส่งลักษณะ  $\Phi$  และเวกเตอร์น้ำหนัก  $w_i$  ของเซลล์ผู้ชนะ  $i$

คุณสมบัติที่สำคัญของการส่งการจัดระเบียบตนเอง (self-organizing map) คือ [Haykin09]

1. การประมาณของปริภูมิอินพุต (Approximation of the input space): การส่งลักษณะ (feature map ( $\Phi$ )) ที่ถูกแทนด้วยเซตของเวกเตอร์น้ำหนัก  $\{w_i\}$  ในปริภูมิเอาต์พุต  $\mathcal{A}$  จะให้การประมาณที่ดีของปริภูมิอินพุต  $\mathcal{X}$
2. การเรียงลำดับทอพอโลยี (Topological ordering): การส่งลักษณะ (feature map ( $\Phi$ )) ที่ถูกคำนวณจากอัลกอริทึม SOM เป็นการเรียงลำดับทอพอโลยี ในแง่ของตำแหน่งเชิงพื้นที่ (spatial location) ของเซลล์ในแลตทิซ ที่สมนัยกับบางโดเมน หรือลักษณะของรูปแบบอินพุต
3. การจับคู่ความหนาแน่น (Density matching): การส่งลักษณะ (feature map ( $\Phi$ )) เป็นตัวสะท้อนการกระจายทางสถิติของการกระจายอินพุต นั่นคือพื้นที่ในปริภูมิอินพุต  $\mathcal{X}$  ที่เวกเตอร์อินพุต  $x$  ถูกสุ่มด้วยการที่มีโอกาสการเกิดสูง จะถูกส่งไปยังโดเมนส่วนใหญ่ของปริภูมิเอาต์พุต และทำให้มีความละเอียดที่ดีกว่า พื้นที่ในปริภูมิอินพุตที่เวกเตอร์อินพุต  $x$  ถูกสุ่มด้วยการที่มีโอกาสการเกิดต่ำ
4. การเลือกลักษณะ (Feature selection): การส่งการจัดระเบียบตนเองสามารถที่จะเลือกเซตของลักษณะที่ดีที่สุดจากข้อมูลในปริภูมิอินพุต เพื่อประมาณการกระจายของข้อมูลได้



รูปที่ 2.28 โครงข่ายประสาทเทียมที่ใช้การจัดระเบียบตนเองในการเลือกลักษณะและเปอร์เซปตรอนหลายชั้นในการจำแนกคำ

การจัดระเบียบตนเองนั้นถูกนำไปใช้ในการจัดกลุ่มข้อมูล และเนื่องจากคุณสมบัติการเลือกลักษณะที่ดี จึงถูกนำไปใช้ควบคู่กับเปอร์เซปตรอนหลายชั้น ยกตัวอย่างเช่นงานวิจัยของ Eng และคณะ [Eng05] เป็นงานวิจัยที่เกี่ยวข้องกับการรู้จำคำพูดภาษามาลาย โดยในงานวิจัยนี้ใช้ SOM ในการแปลงการเข้ารหัสการทำนายเชิงเส้น (linear prediction coding (LPC)) ให้อยู่ในรูปของไบนารีเมทริกซ์ นั่นคือเมื่อส่งลำดับของเสียงสำหรับคำ 1 คำ ให้ SOM เซลล์ที่เป็นเซลล์ผู้ชนะของในแต่ละเสียง จะให้ค่าเอาต์พุตออกมาเป็น 1 ส่วนเซลล์อื่นจะให้ค่าเอาต์พุตเป็น 0 ดังนั้นสำหรับคำ 1 คำ จะมีเอาต์พุตจากหลายตำแหน่งที่ให้ค่าเป็น 0 และเมทริกซ์นี้จะถูกแปลงเป็นเวกเตอร์คอลัมน์

เพื่อเป็นเวกเตอร์อินพุตให้กับเปอร์เซปตรอนหลายชั้น เพื่อการจำแนกค่าต่อไป ลักษณะโดยรวมของโครงข่ายนี้แสดงในรูปที่ 2.28

---

## คำถามท้ายบทที่ 2

2.1 ให้เขียนโปรแกรมสำหรับการฝึกสอนเปอร์เซปตรอนหลายชั้น สำหรับชุดข้อมูลมาตรฐานใดก็ได้ (ให้บอกแหล่งที่มาของชุดข้อมูลนั้นด้วย) และให้ใช้ 10% การตรวจสอบความสมเหตุสมผลแบบไขว้ โดยให้ทำการเปลี่ยนแปลงจำนวนเซลล์ในชั้นซ่อน จำนวนชั้นซ่อน ค่าอัตราการเรียนรู้ ค่าโมเมนตัม และเปลี่ยนการตั้งค่าน้ำหนัก หลังจากทำการฝึกสอน และทดสอบแล้วให้ทำการวิเคราะห์ว่าแต่ละการเปลี่ยนแปลงมีผลกระทบต่อค่าความถูกต้อง หรือไม่อย่างไร

2.2 ให้ทำการเขียนโปรแกรมสำหรับการฝึกสอนการจัดระเบียบตนเอง ในการจัดกลุ่มชุดข้อมูลมาตรฐาน (ให้บอกแหล่งที่มาของชุดข้อมูลนั้นด้วย) และให้แสดงเวกเตอร์น้ำหนักของแต่ละเซลล์สุดท้าย ที่คล้ายกับรูปที่ 2.24

---

# บทนำระบบฟัซซี

## Introduction to Fuzzy System

### บทที่ 3

มนุษย์เราใช้ความรู้ที่ได้จากประสบการณ์เกี่ยวกับโลกที่เราใช้ชีวิตอยู่ และเราใช้ความสามารถที่มีอยู่ในการให้เหตุผลและผล ในการสร้างความสำคัญของข้อมูลต่างๆ หรืออีกนัยหนึ่งก็คือเราสามารถทำความเข้าใจกับสิ่งที่อยู่รอบตัวเรา เรียนรู้สิ่งใหม่ๆ วางแผนการสำหรับอนาคตได้ แต่ทั้งนี้ทั้งนั้นความสามารถของเราถูกจำกัดในการรับรู้ต่างๆ หรือเราถูกจำกัดความสามารถในการให้เหตุผลให้ผลอย่างถ่องแท้ นั่นคือเราถูกจำกัดจาก ความไม่แน่นอน (uncertainty)

ความไม่แน่นอน (uncertainty) คือสถานะที่มีความเป็นไปได้ที่จะเกิดข้อผิดพลาดเนื่องจากการที่เราได้รับข้อมูล (information) ไม่ครบทั้งหมด เกี่ยวกับสิ่งแวดล้อมที่อยู่รอบตัวเรา หรืออีกนัยหนึ่งคือมนุษย์พยายามที่จะทำการตัดสินใจ จัดการ หรือวิเคราะห์ ข้อมูล เหตุการณ์ หรือทำนายเหตุการณ์ในอนาคต แต่เราไม่มีข้อมูลทั้งหมดในมือทำให้มีโอกาสที่จะเกิดความผิดพลาดได้ ตัวอย่างเช่นการขับรถบนถนน ถ้าเราขับไปในที่ที่เราไม่คุ้นเคยเราไม่สามารถบอกได้ว่าเราสามารถขับรถได้เร็วที่ความเร็วเท่าไร เนื่องจากเราไม่สามารถบอกได้ว่าข้างหน้าจะเป็นถนนตรงหรือทางโค้ง หรือมีสิ่งกีดขวางหรือไม่

#### ตัวอย่างที่ 3.1 ความไม่แน่นอนกับความซับซ้อน (uncertainty and complexity) [Klir97]

การขับรถเกียร์ธรรมดา หรือเกียร์กระปุก เป็นการขับรถที่มีความซับซ้อนมากกว่าการขับรถเกียร์อัตโนมัติ ซึ่งการที่จะขับรถเกียร์ธรรมดาได้ดีต้องอาศัยความรู้และความชำนาญมากกว่า แต่ทั้งนี้ทั้งนั้นการขับรถทั้งสองก็มีความไม่แน่นอนเข้ามาเกี่ยวข้องด้วยนั่นคือ เราไม่รู้อย่างแน่นอนว่าเมื่อไหร่ควรจะหยุด หรือ อ้อม เพื่อหลบสิ่งกีดขวาง ถ้าขับรถในที่ที่มีการจราจรหนาแน่น ความไม่แน่นอนก็ยิ่งเพิ่มขึ้น เช่นเดียวกันกับความซับซ้อน นั่นคือความซับซ้อนเพิ่มขึ้นเมื่อเราตระหนักว่าเรามีความรู้เท่าใด และมีสิ่งที่เราไม่รู้เท่าใด

#### ตัวอย่างที่ 3.2 ความไม่แน่นอนกับการวัด (uncertainty and measurement) [Klir97]

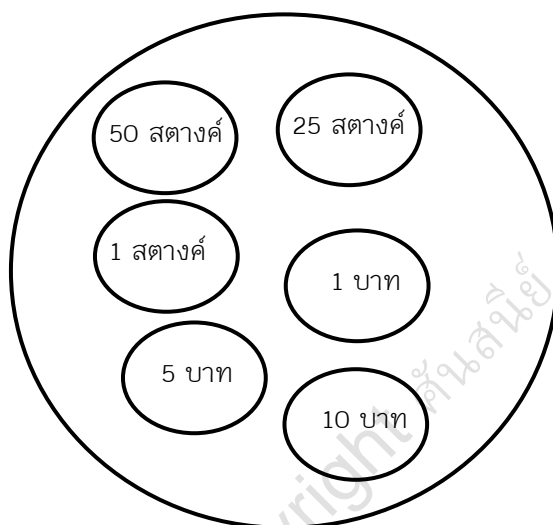
ไม่ว่าเครื่องมือวัดจะมีความเที่ยงตรงขนาดใด ความไม่แน่นอนจะมีอยู่ตลอดเวลาถึงแม้ว่าขนาดเล็กเพียงใดก็ตาม ตัวอย่างเช่นเครื่องมือวัดที่มี เป็นไม้บรรทัดที่มีหน่วยเป็น ฟุต และหน่วยย่อยเป็น นิ้ว โดยที่แต่ละช่องที่แบ่งไว้คือ  $1/16$  นิ้ว แต่ถ้าเป็นไม้บรรทัดที่ใช้ในการทดลอง อาจเป็นช่องละ  $1/64$  นิ้ว หรือเล็กไปกว่านั้นซึ่งอาจจะเป็น  $1/106$  แต่ไม่ว่าจะมีความละเอียดมากเพียงไรก็ยังมีความเป็นไปได้ที่จะเกิดความผิดพลาด เช่นต้องการวัดวัตถุอันหนึ่งแต่ปรากฏว่าไม่สามารถบอกความยาวได้แน่นอนเนื่องจากความยาวของวัตถุนั้นไปสิ้นสุดระหว่างช่องที่แบ่งไว้นั่นเอง

ส่วนความเข้าใจผิดที่เป็นผลมาจากการใช้คำในความหมายที่แตกต่างกัน คือความหมายของความคลุมเครือ (vagueness) หรืออีกนัยหนึ่งคือ มนุษย์เรามีความหมายหลักที่เหมือนกัน และสามารถสื่อสารกันได้ถึงระดับหนึ่งแต่ความหมายอาจจะไม่ถูกต้องนัก เช่นคำว่า ‘หนาว (cold)’ สำหรับประชากรที่อาศัยอยู่ที่กรุงเทพอาจจะมีความรู้สึกถึงคำว่า ‘หนาว’ ที่แตกต่างจากประชากรที่อาศัยอยู่ที่เชียงใหม่ แต่ประชากรทั้งสองที่ รู้ว่าคำว่า ‘หนาว’ หมายถึงอะไร นั่นคือทั้งสองจะไม่

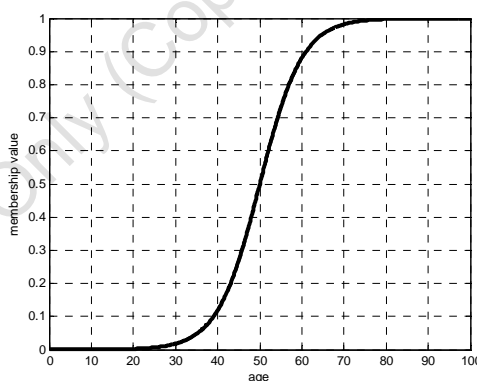


บอกว่า ‘หนาว’ ที่อุณหภูมิเท่ากับ 35 องศาเซลเซียส แต่คนที่อาศัยอยู่ที่กรุงเทพฯ จะบอกว่าหนาวที่อุณหภูมิเท่ากับ 20 องศาเซลเซียส ในขณะที่คนที่อาศัยอยู่ที่เชียงใหม่มีความทนทานกับอากาศมากกว่าและถ้าอุณหภูมิต่ำกว่า 10 องศาเซลเซียสคนที่นี่ถึงบอกว่า ‘หนาว’ ซึ่งแตกต่างจากคนที่กรุงเทพฯ

นอกเหนือจากในภาษาแล้วยังมีปรากฏการณ์ที่เรียกว่า ‘heap paradox’ ที่สามารถอธิบายคำว่า vagueness ได้เช่นเดียวกัน ปรากฏการณ์นี้เกี่ยวข้องกับ ของที่อยู่รวมกันมากๆ เช่น กองหิน ถ้าเราเอาหินออกหนึ่งก้อนจากกอง ก็ยังเรียกกองนี้ว่ากองหินอยู่ แต่ถ้าเราเอาหินออกทีละก้อนไปเรื่อยๆ จนเหลือแค่ 2 ก้อนเราก็จะไม่เรียกว่ากองหินอีกแล้ว แต่ในระหว่างทางคำถามที่เกิดขึ้นคือเมื่อไหร่เราจะหยุดเรียกกองหินกองนี้ว่ากองหิน



รูปที่ 3.1 เซตของเหรียญไทย



รูปที่ 3.2 ฟังก์ชันสมาชิกของฟัซซีเซต ‘แก่’

ในปี ค.ศ. 1965 L. A. Zadeh [Zadeh65] ได้เผยแพร่ผลงานวิจัยในเรื่องฟัซซีเซต (Fuzzy Sets) ใน วารสารวิชาการ ซึ่งในผลงานนี้ได้อธิบายเกี่ยวกับแนวคิดของฟัซซีเซต (fuzzy set) ซึ่งเป็น set ที่มีขอบที่ไม่คมชัด (un-sharp boundary) ทั้งนี้ตรงข้ามกับคริสป์เซต (crisp set) ปกติที่ขอบเขตต้องเด่นชัด จากรูปที่ 3.1 เป็นรูปของเซต (set) ของเหรียญไทย ถ้าสมมุติว่ามีเหรียญบางเหรียญที่ชำรุดหรือหัก ถ้าถามว่าเหรียญนั้นอยู่ในเซตนี้หรือไม่ ถ้าเป็น คริสป์เซต (crisp set) คำตอบคืออยู่หรือไม่อยู่เท่านั้น แต่ในทางตรงข้ามฟัซซีเซตเป็นเซตที่มีขอบที่ไม่ชัดเจนดังนั้นคำว่า เป็นสมาชิกของฟัซซีเซต ไม่ใช่แค่เป็นหรือไม่เป็น หรืออยู่หรือไม่อยู่เท่านั้น หรืออีกนัยหนึ่งคืออยู่ในเซตด้วยระดับ (degree) ของความเป็นสมาชิก ที่มากกว่าหรือน้อยกว่า ดังเช่นฟังก์ชันสมาชิก (membership function) ในรูปที่ 3.2 ซึ่งเป็นฟังก์ชันที่อธิบายถึงฟัซซีเซต ‘แก่’ (‘old’) ถ้าถามว่า

ถึงปริมาณใดที่คนอายุ 55 65 75 หรือ 85 เป็นคนแก่ คำตอบก็ขึ้นอยู่กับมุมมองของแต่ละคน แต่ทั้งนี้ทั้งนั้น 85 เป็นคนแก่แน่นอน แต่ 55 และ 65 ไม่แน่ว่าจะเป็นคนแก่หรือไม่ ทั้งนี้จะขึ้นกับแต่ละสถานการณ์ ส่วน 50 อาจจะไม่ชัดสำหรับเราทุกคนว่าอยู่ในเซตหรือไม่ ดังนั้นสามารถสรุปได้ว่าฟัซซีเซต ไม่ใช้การยืนยันหรือปฏิเสธสิ่งใดสิ่งหนึ่ง

ทฤษฎีความน่าจะเป็นถูกนำไปใช้ในหลายสาขาวิชาแต่ก็ยังไม่สามารถครอบคลุมความไม่แน่นอน (uncertainty) ในหลายด้านได้ โดยเฉพาะอย่างยิ่งความไม่แน่นอนที่เกิดจากความคลุมเครือ (vagueness) ในด้านภาษา ดังนั้นจะเห็นได้ว่า ทั้งทฤษฎีความน่าจะเป็น และ ทฤษฎีฟัซซี สามารถอธิบายความไม่แน่นอนในรูปแบบที่แตกต่างกันนั่นคือ ทฤษฎีความน่าจะเป็นเกี่ยวข้องกับความคาดหวังของเหตุการณ์ในอนาคต ที่ขึ้นกับสิ่งที่รู้อยู่แล้ว เช่น เราสนใจว่ามีโอกาสมากน้อยเท่าไรที่คนต่อไปที่เดินเข้ามาในห้องเรียนเป็นคนสูง ซึ่งแนวคิดของความสูงในที่นี้มาจากการกระจายของส่วนสูงของคนไทยทั้งหมด ถ้าในห้องเรียนที่เรานั่งอยู่มีแต่นักกีฬาบาสเกตบอลซึ่งเป็นคนสูงส่วนใหญ่ เราก็คาดหวังว่าคนที่เดินเข้ามาจะสูงด้วย แต่ถ้าเราอยู่ในห้องที่มีคนที่มีส่วนสูงปกติอยู่ด้วย ความคาดหวังของเราอาจลดต่ำลง ในที่นี้ความรู้สึกของความไม่แน่นอนอยู่ที่การทำนายเกี่ยวกับเหตุการณ์

แต่ความไม่แน่นอนในความหมายของฟัซซี ไม่ใช่ความไม่แน่นอนของความคาดหวัง แต่เป็นความไม่แน่นอนจากแนวคิดที่ถูกอธิบายโดยคำพูดเช่นเราอยู่ในห้องที่มีแต่นักกีฬาบาสเกตบอล ถ้ามีคนสูง 180 เซนติเมตรเดินเข้ามา เราอาจตั้งคำถามว่ามีโอกาสเท่าไรที่คนผู้นี้ (หมายถึงคนที่เดินเข้ามาในห้อง) เป็นคนสูง คำตอบคือไม่มีความคาดหวังอีกต่อไปเนื่องจากเขาอยู่ในห้องแล้ว แต่ถ้าถามว่ามีความถูกต้องเท่าไรที่จะกล่าวได้ว่าคนผู้นี้เป็นคนสูง ถ้าเราเปรียบเทียบกับประชากรในประเทศไทย คำตอบจะเป็น ถูกต้องมาก แต่ถ้าเราเปรียบเทียบกับประชากรในห้อง คำตอบจะเป็น ถูกต้องน้อย ถ้าคนส่วนใหญ่สูง จากที่กล่าวมาทั้งหมดเป็นการอธิบายถึงความสัมพันธ์ระหว่างคุณสมบัติของแต่ละบุคคล คือ ส่วนสูง(height) กับความไม่ชัดเจนของแนวคิดคือ สูง (tallness)

จากที่กล่าวมาข้างต้นสามารถกล่าวได้ว่าความน่าจะเป็น เป็นทฤษฎีของเหตุการณ์สุ่ม (random event) ซึ่งเกี่ยวข้องกับโอกาสที่จะเกิดเหตุการณ์ใดเหตุการณ์หนึ่ง ในขณะที่ ทฤษฎีฟัซซีเซตไม่ขึ้นกับเหตุการณ์ แต่เกี่ยวข้องกับแนวคิด (concept) เช่น สูง (tall) อุ่น (warm) หนาว (cold) เป็นต้น [Klir97]

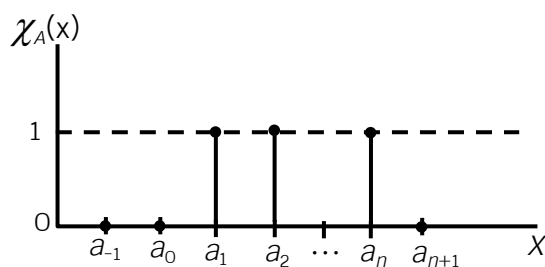
### 3.1 พื้นฐานฟัซซีเซต (Basic Fuzzy Set)

เพื่อให้เกิดความเข้าใจในฟัซซีเซตมากขึ้นในหัวข้อนี้จะกล่าวถึง ฟังก์ชันสมาชิก (membership function) ในทุกรูปแบบ การสร้างฟัซซีเซต การดำเนินการในฟัซซีเซต (Operations on Fuzzy Sets) รวมทั้งคุณสมบัติของฟัซซีเซต

#### 3.1.1 ฟังก์ชันสมาชิก (Membership Function)

ก่อนที่จะกล่าวถึงฟังก์ชันสมาชิกของฟัซซีเซต จะขอกล่าวถึงฟังก์ชันลักษณะ (characteristic function) [Klir95] ของเซตดั้งเดิม (classical set) หรือ คริสป์เซต (crisp set) ซึ่งจากการเขียนคริสป์เซต  $A = \{a_1, a_2, \dots, a_n\}$  ที่ถูกเรียกว่าวิธีการแสดง (list method) โดยที่เป็นการแทนคริสป์เซตด้วยการแจกแจงส่วนประกอบในเซต ที่  $a_i$  สำหรับ  $i$  เท่ากับ 1 ถึง  $n$  เป็นสมาชิกหรือส่วนประกอบของเซต  $A$  ( $a_i \in A$ ) ดังนั้นฟังก์ชันลักษณะของคริสป์เซต  $A$  คือ  $a_i$  สำหรับ  $i$  เท่ากับ 1

ถึง  $n$  จะมีค่าในฟังก์ชันลักษณะ เป็น 1 ในขณะที่  $a_j$  สำหรับ  $j$  ที่ไม่เท่ากับ 1 ถึง  $n$  แต่เป็นสมาชิกของเซตสากล (universal set ( $X$ )) จะมีค่าในฟังก์ชันลักษณะเป็น 0 ดังรูปที่ 3.3



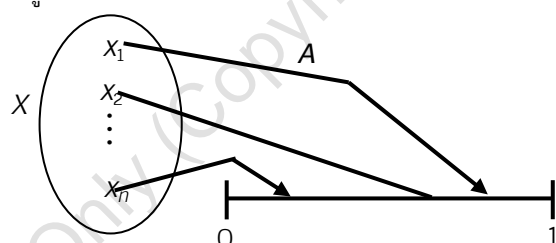
รูปที่ 3.3 ฟังก์ชันลักษณะของคริปเซต  $A$

ในขณะที่สมาชิกของฟัซซีเซตเป็นเรื่องของระดับ (degree) เช่นคนคนหนึ่งเป็นสมาชิกของเซต ‘คนสูง’ ถึงระดับ (degree) ที่คนคนนั้นมีคุณสมบัติเข้าข่ายของแนวคิดของ ‘สูง’ หรือกล่าวอีกนัยหนึ่งคือ ระดับของสมาชิก ของแต่ละสมาชิกในฟัซซีเซตบ่งบอกถึง ระดับของความใช้แทนกันได้ (degree of compatibility) ของสมาชิกต่อแนวคิดที่แทนฟัซซีเซตนั้น

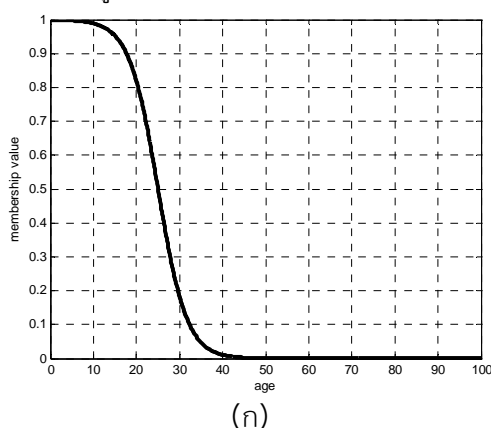
ฟัซซีเซต  $A$  ถูกกำหนดบนเซตสากล (universal set ( $X$ )) โดยเป็นฟังก์ชันแบบเดียวกับฟังก์ชันลักษณะ ซึ่งถูกเรียกว่าฟังก์ชันสมาชิก (membership function) ที่จะให้ค่าเป็นตัวเลข ( $A(x)$ ) กับสมาชิก  $x$  ในเซต  $X$  ซึ่งตัวเลขนี้เป็นสมาชิกของช่วงปิด  $[0,1]$  ซึ่งเป็นค่าที่บอกลักษณะของระดับของสมาชิก  $x$  ใน  $A$  [Klir95]

$$A: X \rightarrow [0,1] \text{ หรือ } \mu_A: X \rightarrow [0,1] \quad (3.1)$$

ฟังก์ชันลักษณะนี้แสดงในรูปที่ 3.4

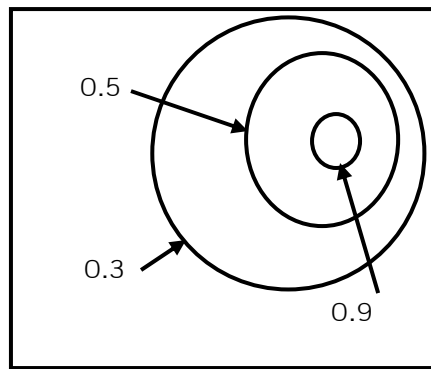


รูปที่ 3.4 ฟังก์ชันสมาชิก  $A$



(ก)

น้ำหนัก



ส่วนสูง

(ข)

รูปที่ 3.5 ฟังก์ชันสมาชิกของ (ก) ‘young’ และ (ข) ‘คนตัวใหญ่’

ในบางครั้งการแทนฟังก์ชันสมาชิกสามารถทำได้รูปที่ 3.5(ก) ซึ่งเป็นฟังก์ชันสมาชิกของฟัซซีเซต ‘young’ หรือ ตัวอย่างในรูปที่ 3.5(ข) เป็นการแทนฟังก์ชันสมาชิกของฟัซซีเซตที่มี 2 มิติโดยที่แต่ละมิติมีความสัมพันธ์กันเช่นเซตของ ‘คนตัวใหญ่’ ซึ่งสามารถอธิบายได้ด้วย ส่วนสูงและน้ำหนัก โดยปกติการแทนแบบนี้ถูกเรียกว่าการแทนแบบแผนภาพคอนทัวร์ (contour diagram) จะเห็นว่าจุดที่อยู่ในวงกลม 0.9 จะมีค่าสมาชิกมากกว่าจุดอื่น และที่ตำแหน่งอื่นก็เป็นเช่นเดียวกัน

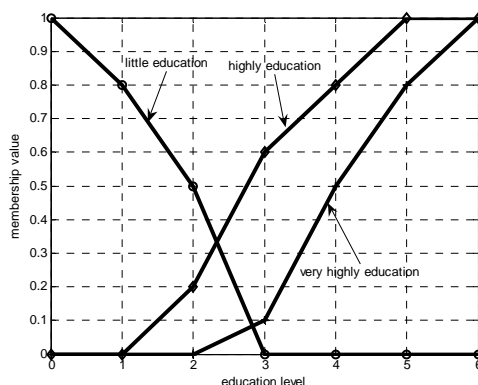
ตารางที่ 3.1 เซตสากลของระดับการศึกษา

เลขหมาย	ระดับการศึกษา
0	ไม่มีการศึกษา
1	ประถมศึกษา
2	มัธยมศึกษา
3	หลักสูตรวิชาชีพชั้นสูง
4	ปริญญาตรี
5	ปริญญาโท
6	ปริญญาเอก

อีกตัวอย่างของการแทนแบบรูปคือ สมมติให้เซตสากล (universal set) ประกอบด้วยระดับของการศึกษาทั้ง 7 ระดับดังตารางที่ 3.1 ในชีวิตปัจจุบันคนมักจะพูดว่ามีการศึกษาสูง หรือต่ำ ซึ่งจากการพูดแบบนี้แสดงให้เห็นความคลุมเครือในภาษาเช่นกัน ถ้าเราพยายามจะอธิบายความคลุมเครือเหล่านี้ด้วย ฟังก์ชันสมาชิก ‘การศึกษาน้อย’ ‘การศึกษาสูง’ ‘การศึกษาสูงมาก’ ดังรูปที่ 3.6 ซึ่งเป็นการพยายามอธิบายแนวคิดเรื่องการศึกษา เช่นถ้าคนที่จบปริญญาตรีจะเทียบเท่ากับมีค่าความเป็นสมาชิก 0.8 ในฟัซซีเซต ‘การศึกษาสูง’ และมีค่าความเป็นสมาชิก 0.5 ในฟัซซีเซต ‘การศึกษาสูงมาก’ นี่แสดงให้เห็นว่าคนที่จบปริญญาตรีมีการศึกษาในระดับที่ดีแต่เนื่องจากมีปริญญาที่สูงกว่าอีก 2 ระดับทำให้ฟัซซีเซต ‘การศึกษาสูง’ มีความเหมาะสมกับปริญญาตรีมากกว่า ในขณะที่ฟัซซีเซต ‘การศึกษาสูงมาก’ มีความเหมาะสมน้อยลงไป

รูปร่างของการเปลี่ยนแปลงของค่าความเป็นสมาชิกจาก 0 ไป 1 ในฟัซซีเซตต่างๆ นั้นไม่ใช่สิ่งที่ต้องคำนึงถึงมาก เนื่องจากเราไม่รู้แน่นอนเกี่ยวกับการเปลี่ยนแปลงให้เป็นไปตามความหมายในภาษาในแต่ละใจความ ดังนั้นรูปร่างจะเป็นไปตามประสบการณ์ว่าคำนั้นถูกใช้อย่างไรในใจความ

นั้น และในหลายงานหรือการประยุกต์ใช้ไม่สนใจในรูปร่างที่แท้จริง ดังนั้นรูปร่างของฟัซซีเซตที่ง่าย จึงถูกใช้ในงานส่วนใหญ่



รูปที่ 3.6 ฟังก์ชันสมาชิกของฟัซซีเซต ‘การศึกษาน้อย(0)’ ‘การศึกษาสูง(1)’ ‘การศึกษาสูงมาก (+)’

นอกเหนือจากการแทนฟังก์ชันสมาชิกด้วยรูปยังมีการแทนด้วยตาราง (tabular) หรือการแสดง (list) เพราะโดยปกติการแทนด้วยรูปทำได้ถ้าเป็นฟังก์ชันใน 1 หรือ 2 มิติ ในยูคลิเดียนสเปซ (Euclidean space) เท่านั้น การแทนด้วยตารางทำได้เช่นตารางที่ 3.2 ซึ่งเป็นฟัซซีเซต A

ตารางที่ 3.2 ฟัซซีเซต A

ชื่อนักศึกษา (สัญลักษณ์)	ค่าความเป็นสมาชิกในฟัซซีเซต A
Carry ( $x_1$ )	0.8
Bill ( $x_2$ )	0.3
J-H ( $x_3$ )	0.5
Wabei ( $x_4$ )	0.9

ส่วนการแทนด้วยการแสดงของฟัซซีเซต A ทำได้โดย  $A = \{ \langle \text{Carry}, 0.8 \rangle, \langle \text{Bill}, 0.3 \rangle, \langle \text{J-H}, 0.5 \rangle, \langle \text{Wabei}, 0.9 \rangle \}$  หรือถ้าใช้สัญลักษณ์จะได้  $A = \{ \langle x_1, 0.8 \rangle, \langle x_2, 0.3 \rangle, \langle x_3, 0.5 \rangle, \langle x_4, 0.9 \rangle \}$  แต่ในหลายครั้งเราด้วยการแสดงดังสมการที่ 3.2

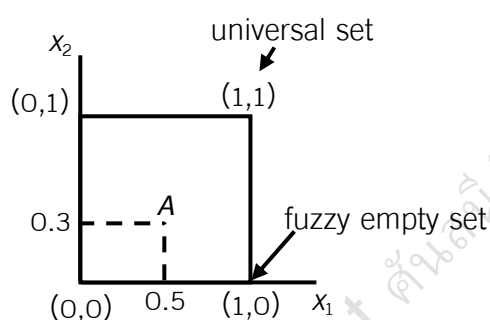
$$A = \frac{0.8}{\text{Carry}} + \frac{0.3}{\text{Bill}} + \frac{0.5}{\text{J-H}} + \frac{0.9}{\text{Wabei}} \quad (3.2)$$

การแทนด้วยสมการที่ 3.2 ไม่ได้เป็นการหารหรือบวกตามความหมายทางคณิตศาสตร์จริง แต่เป็นเพียงแค่การแสดงซึ่งบ่งบอกว่าแต่ละตัวมีค่าความเป็นสมาชิกเป็นเช่นไรเช่น Carry มีค่าความเป็นสมาชิกในฟัซซีเซต A เท่ากับ 0.8 และ Bill มีค่าความเป็นสมาชิกในฟัซซีเซต A เท่ากับ 0.3 เป็นต้น ดังนั้นสามารถเขียนสมการของฟังก์ชันสมาชิกของฟัซซีเซต ในกรณีที่เป็นฟังก์ชันดิสครีต (discrete function) ได้ดังสมการที่ 3.3ก และถ้าเป็นฟังก์ชันต่อเนื่อง (continuous function) สามารถเขียนได้ดังสมการที่ 3.3ข

$$A = \sum_x \frac{A(x)}{x} \quad (3.3ก)$$

$$A = \int_x \frac{A(x)}{x} \quad (3.3ข)$$

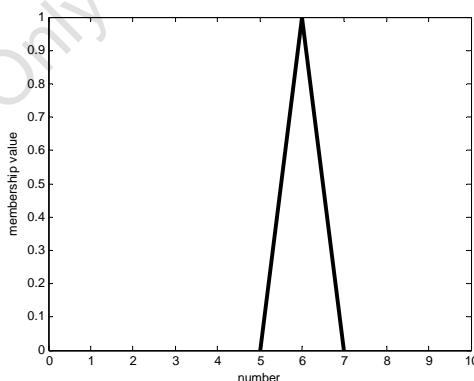
นอกเหนือจากการแทนดังที่กล่าวแล้วยังมีการแทนด้วยเรขาคณิต (geometric) สมมุติให้เซตสากล  $X = \{x_1, x_2, \dots, x_n\}$  โดยที่  $x_i$  สำหรับ  $i=1,2,\dots,n$  ถูกมองเป็นแกน (axis) ใน  $n$  มิติ ในยูคลิเดียนสเปซ ถ้าจำกัดค่าของแต่ละแกนให้อยู่ในช่วงปิด  $[0,1]$  เราจะได้ซับเซตของยูคลิเดียนสเปซ ที่เรียกว่า  $n$  มิติ ในหน่วยลูกบาศก์ (unit cube) โดยที่แต่ละจุดที่อยู่ในหน่วยลูกบาศก์นั้นจะเป็นค่าความเป็นสมาชิกของ  $x_i$  ในฟัซซีเซตใดๆ ดังนั้นตำแหน่งพิกัด (coordinate) ของแต่ละจุดในยูคลิเดียนสเปซนี้ก็คือค่าความเป็นสมาชิกของ  $x_i$  ในฟัซซีเซตนั่นเอง เช่น  $X = \{x_1, x_2\}$  ฟัซซีเซต  $A = 0.5/x_1 + 0.3/x_2$  ดังแสดงในรูปที่ 3.7 และฟัซซีเซตว่างคือจุด (0,0) และ ฟัซซีเซตของเซตสากลคือจุดที่ (1,1) นั่นเอง ฟัซซีพาวเวอร์เซต (power set) ( $\tilde{P}(X)$ ) คือเซตของฟัซซีเซตทุกฟัซซีเซตบน  $X$  ซึ่งไม่จำเป็นต้องมีจำนวนจำกัด หรือหมายถึงทุกจุดที่อยู่ในกรอบใน  $n$  มิติ ในยูคลิเดียนสเปซ เช่นทุกจุดในกรอบในรูปที่ 3.7 เป็นต้น โดยปกติ  $\tilde{P}(X)$  คือ  $[0,1]^n = I^n$



รูปที่ 3.7 การแทนฟังก์ชันสมาชิกด้วยเรขาคณิต

การแทนในรูปแบบสุดท้ายคือการแทนโดยใช้การวิเคราะห์ (analytic) โดยปกติการแทนแบบนี้จะใช้เมื่อเซตสากล มีสมาชิกไม่จำกัดดังตัวอย่างที่ 3.3

ตัวอย่างที่ 3.3 ฟัซซีเซต  $A$  หรือ ‘ประมาณ 6’ สามารถเขียนได้ดังรูป 3.8



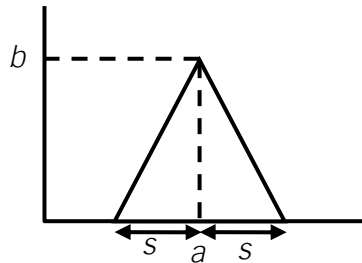
รูปที่ 3.8 ฟังก์ชันสมาชิกของฟัซซีเซต ‘ประมาณ 6’

หรือเขียนเป็นสมการได้ดังนี้

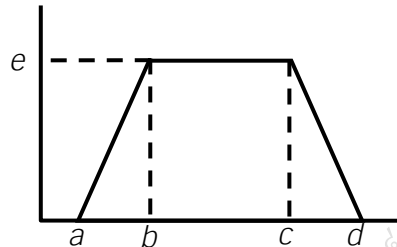
$$A(x) = \begin{cases} x - 5 & 5 \leq x \leq 6 \\ 7 - x & 6 \leq x \leq 7 \\ 0 & \text{else} \end{cases} \quad (3.4)$$

โดยปกติแล้วฟังก์ชันที่มีรูปร่างเป็นสามเหลี่ยมที่สมมาตรดังรูป 3.9(ก) สามารถแทนได้ด้วยฟังก์ชันในสมการที่ 3.5

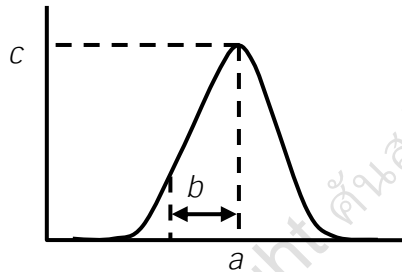
$$A(x) = \begin{cases} b \left( 1 - \frac{|x-a|}{s} \right) & a-s \leq x \leq a+s \\ 0 & \text{else} \end{cases} \quad (3.5)$$



(ก)



(ข)



(ค)

รูปที่ 3.9 (ก) ฟังก์ชันรูปสามเหลี่ยม (triangular shape) (ข) ฟังก์ชันรูปสี่เหลี่ยมคางหมู (trapezoidal) และ (ค) ฟังก์ชันระฆังคว่ำ (Bell-shaped)

และฟังก์ชันที่มีรูปร่างเป็นสี่เหลี่ยมคางหมู (trapezoidal) ซึ่งในบางครั้งอาจเขียนเป็นพีชคณิต ( $a, b, c, d$ ) ดังรูปที่ 3.9(ข) สามารถแทนได้ด้วยฟังก์ชันในสมการที่ 3.6

$$A(x) = \begin{cases} \frac{(a-x)e}{a-b} & a \leq x \leq b \\ e & b \leq x \leq c \\ \frac{(d-x)e}{d-c} & c \leq x \leq d \\ 0 & \text{else} \end{cases} \quad (3.6)$$

นอกเหนือจากนี้ยังมีฟังก์ชันในรูปแบบระฆังคว่ำ (Bell-shaped) ดังรูป 3.9(ค) ซึ่งถูกใช้ในหลายงาน โดยที่เขียนในรูปสมการได้ดังนี้

$$A(x) = ce^{-\frac{(x-a)^2}{b}} \quad (3.7)$$

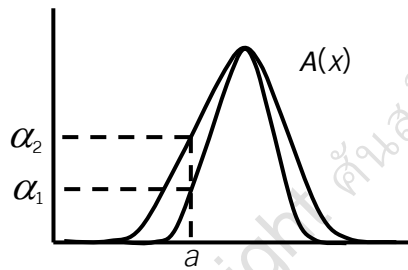
### 3.1.2 ฟัชซีเซตรูปแบบอื่น

นอกเหนือจากฟัชซีเซตดังที่กล่าวมาแล้ว ยังมีฟัชซีเซตในรูปแบบอื่นที่ถูกเรียกว่า ฟัชซีเซตแบบช่วง (Interval-valued fuzzy sets) ฟัชซีเซตแบบ ชนิดที่ 2 (Type-2 fuzzy Sets) และฟัชซีเซตแบบระดับที่ 2 (Level 2 fuzzy sets)

ฟัชซีเซตแบบช่วง(Interval-valued fuzzy sets) [Klir95] เป็นฟัชซีเซตที่แต่ละสมาชิกในเซตมีค่าความเป็นสมาชิกเป็นช่วงปิดของจำนวนจริงที่มีค่าขอบเขตล่างและขอบเขตบน ไม่ใช่ตัวเลขดังเช่นฟัชซีเซตในหัวข้อที่แล้ว นั่นคือ

$$A : X \rightarrow \mathcal{E}([0,1]) \quad (3.8)$$

โดยที่  $\mathcal{E}([0,1])$  เป็นครอบครัว (family) ของช่วงปิดของจำนวนจริงทั้งหมดในช่วง  $[0,1]$  นั่นคือ  $\mathcal{E}([0,1]) \subset P([0,1])$  ตัวอย่างของฟัชซีเซตแบบช่วงแสดงในรูป 3.10 จากตัวอย่างในรูป ค่าความเป็นสมาชิกของ  $a$  ( $A(a)$ ) มีค่าเท่ากับ  $[\alpha_1, \alpha_2]$  ฟัชซีเซตแบบนี้จะไม่จำเพาะ (specific) ดังเช่นฟัชซีเซตปกติ แต่อย่างไรก็ตามฟัชซีเซตแบบนี้ถูกใช้ในงานหลายๆอย่าง แต่ข้อเสียคือ เสียเวลาในการทำงานกับฟัชซีเซตแบบนี้มาก

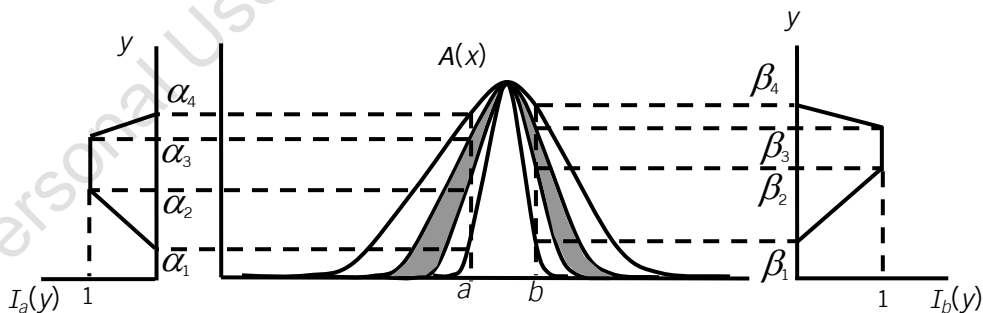


รูปที่ 3.10 ฟัชซีเซตแบบช่วง

ฟัชซีเซตชนิดที่ 2 (Type-2 fuzzy sets) [Klir95] เป็นฟัชซีเซตที่มีค่าความเป็นสมาชิกเป็นฟัชซีเซต นั่นคือ

$$A : X \rightarrow \tilde{\mathcal{P}}([0,1]) \quad (3.9)$$

โดยที่  $\tilde{\mathcal{P}}([0,1])$  เป็นเซตของฟัชซีเซตปกติที่ถูกนิยามให้อยู่ในเซตสากล (universal set)  $[0,1]$  ดังนั้นเราสามารถเรียก  $\tilde{\mathcal{P}}([0,1])$  ได้อีกอย่างหนึ่งว่าฟัชซีฟาวเวอร์เซต ของ  $[0,1]$  ดังรูปที่ 3.11



รูปที่ 3.11 ฟัชซีเซตแบบชนิดที่ 2 (type-2 fuzzy set)

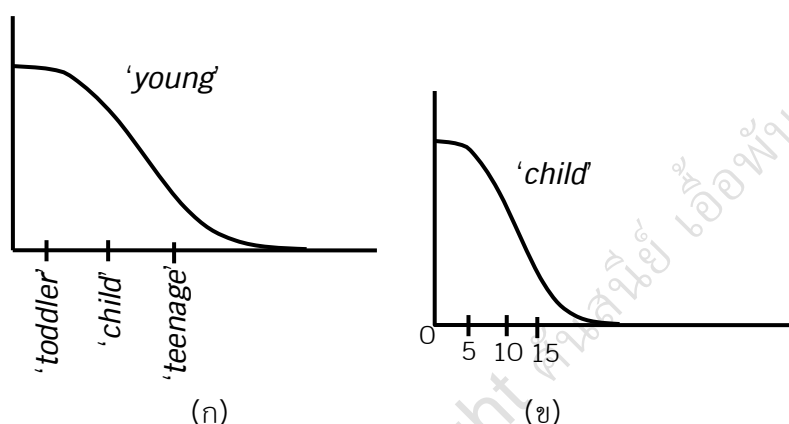
จากตัวอย่างในรูปที่ 3.11 ค่าความเป็นสมาชิกของ  $a$  เป็นฟัชซีเซต  $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$  และค่าความเป็นสมาชิกของ  $b$  เป็นฟัชซีเซต  $(\beta_1, \beta_2, \beta_3, \beta_4)$  นอกเหนือจากฟัชซีเซตแบบชนิดที่ 2 แล้ว ยังมีฟัชซีเซตแบบชนิดที่ 3 หรือที่สูงกว่า 3 ได้ โดยการเรียกซ้ำ (recursive) ในรูปแบบที่กล่าวมาแล้ว



ฟัซซีเซตแบบระดับที่ 2 (level 2 fuzzy sets) [Klir95] มีขึ้นมาใช้กับเหตุการณ์ที่แต่ละสมาชิกของเซตสากลไม่สามารถถูกระบุได้แน่นอน นั่นคือ แต่ละสมาชิกเป็นฟัซซีเซต และฟังก์ชันสมาชิกมีรูปแบบดังนี้

$$A : \tilde{P}(x) \rightarrow ([0,1]) \quad (3.10)$$

โดยที่  $\tilde{P}(x)$  เป็นฟัซซีพาวเวอร์เซตของ  $x$  ในรูปที่ 3.12(ก) แสดงตัวอย่างของฟัซซีเซต 'young' ที่เป็นฟัซซีเซตแบบระดับที่ 2 โดยที่เซตสากลคือ {'toddler', 'child', 'teenage'} ส่วนฟังก์ชันสมาชิกของฟัซซีเซต 'child' ที่ถูกกำหนดให้อยู่ในเซตสากลของอายุ {0,1,2,...} แสดงในรูปที่ 3.12(ข) และเช่นเดียวกับฟัซซีเซตแบบชนิดที่ 2 ฟัซซีเซตแบบนี้สามารถมีระดับที่สูงๆขึ้นไปได้และมีวิธีการเรียกซ้ำเช่นเดียวกัน



รูปที่ 3.12 (ก) ฟัซซีเซตแบบระดับที่ 2 'young' และ (ข) ฟัซซีเซต 'child'

### 3.1.3 การสร้างฟัซซีเซต

โดยปกติแล้วมีวิธีการสร้างฟัซซีเซต [Klir97] หลายรูปแบบ และในการทำงานหลายครั้ง การสร้างเกิดจากสอบถามผู้เชี่ยวชาญ แต่ถ้ามีผู้เชี่ยวชาญมากกว่า 1 คน เราสามารถทำการรวมความคิดเห็นเหล่านั้นเพื่อสร้างระดับของค่าสมาชิกที่น่าเชื่อถือได้

**ตัวอย่างที่ 3.4** [Klir97] สมมติให้คนขับรถ 5 คนคือ สมชาย วรพจน์ กานต์ เกิด และ อัน และมีกรรมการทั้งหมด 10 คน นั่นคือ  $r_1$   $r_2$  จนถึง  $r_{10}$

ตารางที่ 3.3 คำตอบจากกรรมการทั้ง 10

	สมชาย( $x_1$ )	วรพจน์( $x_2$ )	กานต์( $x_3$ )	เกิด( $x_4$ )	อัน( $x_5$ )
$r_1$	1	1	1	1	1
$r_2$	0	0	1	1	1
$r_3$	0	1	0	1	0
$r_4$	1	0	1	1	1
$r_5$	0	0	1	1	1
$r_6$	0	1	1	1	1
$r_7$	0	0	0	0	0
$r_8$	1	1	1	1	1
$r_9$	0	0	0	1	0
$r_{10}$	0	0	0	1	0

ในการแข่งขันนี้ต้องการหาคนที่ขับรถดีที่สุด จึงทำการถามกรรมการทั้ง 10 เกี่ยวกับคนทั้ง 5 ว่าแต่คนขับรถดีหรือไม่ โดยที่ให้ตอบเป็น 1 (ใช่) และ 0 (ไม่ใช่) คำตอบทั้งหมดแสดงในตารางที่ 3.3

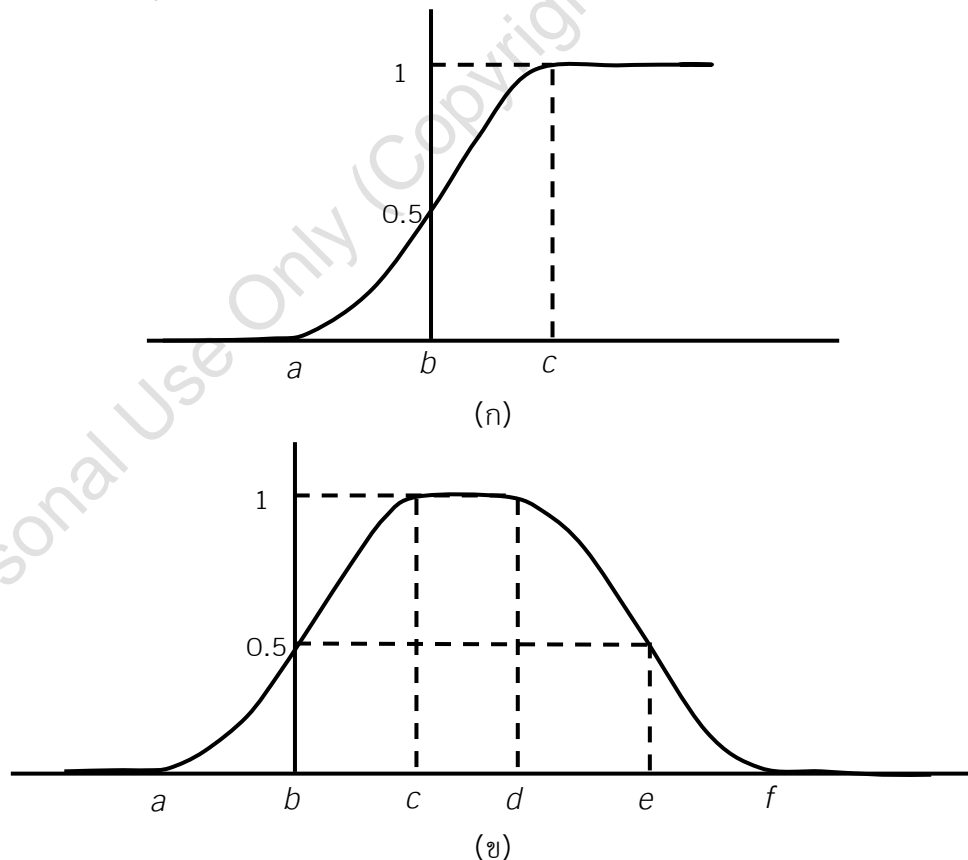
การรวมความเห็นทั้งหมดทำได้โดยการหาสัดส่วนของคำตอบ ใช่ต่อสัดส่วนทั้งหมด ดังนั้นจะได้ฟังก์ชันเซตของคนขับรถดี ( $A$ ) เป็น

$$A = \frac{0.3}{\text{สมชาย}} + \frac{0.4}{\text{วรพจน์}} + \frac{0.6}{\text{กานต์}} + \frac{0.9}{\text{เกิด}} + \frac{0.6}{\text{อัน}} \quad (3.11)$$

ยังมีฟังก์ชันในทางคณิตศาสตร์ที่เรานำมาใช้ในการสร้างฟังก์ชันสมาชิกเช่น S-function ซึ่งมีลักษณะดังนี้

$$s(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \frac{1}{2} \left( \frac{x-a}{b-a} \right)^2 & a \leq x \leq b \\ 1 - \frac{1}{2} \left( \frac{x-c}{c-b} \right)^2 & b \leq x \leq c \end{cases} \quad (3.12)$$

และ  $\pi$ -function ซึ่งมีลักษณะเป็น S-function + ภาพสะท้อนของ S-function ทั้ง S- และ  $\pi$ -function แสดงในรูปที่ 3.13



รูปที่ 3.13 (ก) S-function (ข)  $\pi$ -function

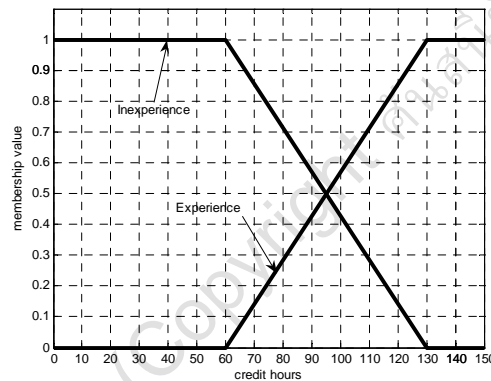
### 3.1.4 การดำเนินการในฟัซซีเซต (Operations on Fuzzy Sets)

ในหัวข้อนี้จะกล่าวถึงการดำเนินการต่างๆ เช่น ฟัซซีคอมพลีเมนต์มาตรฐาน (standard fuzzy complement) ฟัซซียูเนียนมาตรฐาน (standard fuzzy union) ฟัซซีอินเตอร์เซกชันมาตรฐาน (standard fuzzy intersection) และคุณสมบัติของการดำเนินการเหล่านี้ [Klir97]

ให้ฟัซซีเซต  $A$  ถูกกำหนดให้อยู่บนเซตสากล  $X$  และ ฟัซซีคอมพลีเมนต์ ( $\bar{A}$ ) เป็นฟัซซีเซตในเซตสากล  $X$  เช่นกัน โดยปกติ  $A(x)$  บอกถึงระดับของความเป็นสมาชิกของ  $x$  ใน  $A$  ดังนั้นฟัซซีคอมพลีเมนต์มาตรฐานของฟัซซีเซต  $A$  ( $\bar{A}(x)$ ) จะเป็นการบอกถึงระดับของความไม่เป็นสมาชิกของ  $x$  ใน  $A$  ซึ่งเราสามารถเขียนให้เป็นสมการทางคณิตศาสตร์ได้คือ  $\forall x \in X$

$$\bar{A}(x) = 1 - A(x) \quad (3.13)$$

ตัวอย่างของฟัซซีคอมพลีเมนต์แสดงในรูปที่ 3.14 ซึ่ง 'Inexperience' เป็นฟัซซีคอมพลีเมนต์ของฟัซซีเซต 'Experience' จากรูปจะเห็นว่าที่จำนวนชั่วโมงเท่ากับ 80 มีระดับความเป็นสมาชิกในฟัซซีเซต 'Inexperience' เท่ากับ 0.7 ในขณะที่ระดับความเป็นสมาชิกในฟัซซีเซต 'Experience' เป็น 0.3 ซึ่งจะเห็นได้ว่า คอมพลีเมนต์ของฟัซซีเซตแตกต่างจากในเซตแบบดั้งเดิม นั่นคือค่าความเป็นสมาชิกหรือฟังก์ชันสมาชิกสามารถสลับกันได้



รูปที่ 3.14 ฟัซซีเซต 'Experience' และ 'Inexperience'

ในกรณีของฟัซซียูเนียนมาตรฐานของฟัซซีเซต  $A$  และ  $B$  สามารถเขียนฟังก์ชันสมาชิกที่แสดงถึงค่าความเป็นสมาชิกของทุก  $x \in X$  ในฟัซซียูเนียนได้เป็น

$$(A \cup B)(x) = \max[A(x), B(x)] \quad (3.14)$$

ตัวอย่างที่ 3.3 [Klir97] ให้  $X$  เป็นเซตของคนใช้จำนวน  $n$  คน โดยที่ชื่อของแต่ละคนถูกแทนด้วยตัวเลข  $1, 2, \dots, n$  และให้ฟัซซีเซต  $A$  และ  $B$  เป็นฟัซซีเซตของคนใช้ที่มี ความดันสูง และใช้สูงตามลำดับ

ตารางที่ 3.4 ค่าความเป็นสมาชิกของคนใช้ในฟัซซีเซต  $A$ ,  $B$  และ  $A \cup B$

คนใช้	$A$	$B$	$A \cup B$
1	1	1	1
2	0.5	0.6	0.6
3	1	0.1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	0.1	0.7	0.7

ดังนั้น  $A \cup B$  ของคนไข้ในเซต  $X$  แสดงถึง การมีความดันสูงหรือไข้สูง ซึ่งค่าความเป็นสมาชิกของคนไข้แต่ละคนในฟัซซีเซตทั้งสามแสดงในตารางที่ 3.4

กฎนิรฆณ (law of excluded middle) ในเซตแบบดั้งเดิมที่กล่าวว่า  $A \cup \bar{A} = X$  ไม่เป็นจริงในกรณีของฟัซซีเซตมาตรฐานและฟัซซีคอมพลิเมนต์มาตรฐาน เช่นถ้า  $A(x)$  มีค่าเท่ากับ 0.6 และ  $\bar{A}(x)$  จะมีค่าเท่ากับ 0.4 ดังนั้น  $(A \cup \bar{A})(x)$  จะมีค่าเท่ากับ 0.6 ดังนั้น  $x$  เป็นสมาชิกของเซตสากลด้วยค่าความเป็นสมาชิกที่ไม่เท่ากับ 1 ดังนั้นกฎนี้ไม่เป็นจริง

ในกรณีของฟัซซีอินเตอร์เซกชันมาตรฐานของฟัซซีเซต  $A$  และ  $B$  สามารถเขียนฟังก์ชันสมาชิกที่แสดงถึงค่าความเป็นสมาชิกของทุก  $x \in X$  ในฟัซซีอินเตอร์เซกชันได้เป็น

$$(A \cap B)(x) = \min[A(x), B(x)] \quad (3.15)$$

ตัวอย่างที่ 3.4 [Klir97] ให้  $A$  และ  $B$  เป็นฟัซซีเซตของแม่น้ำที่ยาว และแม่น้ำที่ใช้เดินเรือได้ตามลำดับ โดยที่เซตสากลเป็น {อะเมซอน, ไนล์, เจ้าพระยา, โขง, แม่ปิง} และ  $A \cap B$  เป็นฟัซซีเซตของแม่น้ำที่ยาวและใช้เดินเรือได้ ซึ่งค่าความเป็นสมาชิกของแม่น้ำในฟัซซีเซตทั้งสามแสดงในตารางที่ 3.5

ตารางที่ 3.5 ค่าความเป็นสมาชิกของคนไข้ในฟัซซีเซต  $A$ ,  $B$  และ  $A \cap B$

แม่น้ำ	$A$	$B$	$A \cap B$
อะเมซอน	1	0.8	0.8
ไนล์	0.9	0.7	0.7
เจ้าพระยา	0.8	0.8	0.8
โขง	0.5	0.6	0.5
แม่ปิง	0.4	0.3	0.3

เช่นเดียวกับกฎนิรฆณ กฎของความขัดแย้ง (law of contradiction) ที่กล่าวว่า  $A \cap \bar{A} = \emptyset$  ไม่เป็นจริงสำหรับฟัซซีอินเตอร์เซกชันมาตรฐานและฟัซซีคอมพลิเมนต์มาตรฐาน เช่นถ้า  $A(x)$  มีค่าเท่ากับ 0.6 และ  $\bar{A}(x)$  มีค่าเท่ากับ 0.4 ดังนั้น  $(A \cap \bar{A})(x)$  จะมีค่าเท่ากับ 0.4 ซึ่งแสดงให้เห็นว่า  $x$  ยังเป็นสมาชิกของ  $A \cap \bar{A}$  ถึงระดับหนึ่งซึ่งขัดกับนิยามของฟัซซีเซตว่างที่กล่าวว่า ทุก  $x \in X$  จะมีค่าความเป็นสมาชิกเท่ากับ 0 เสมอ

แต่อย่างไรก็ตามคุณสมบัติอื่นเช่น คอมมิวเททีฟ (commutative) แอสโซซิเอทีฟ (associativity) ไอเดมโพเทนซี (idempotency) การกระจาย (distribution) และ De Morgan's Law ยังคงเป็นความจริงสำหรับการดำเนินการฟัซซีเซตโดยวิธีมาตรฐานเหล่านี้ แต่ถ้าใช้การดำเนินการที่ไม่ใช่วิธีมาตรฐาน คุณสมบัติเหล่านี้อาจจะไม่เป็นจริงอีกต่อไป

ตัวอย่างที่ 3.5 จงพิสูจน์ว่าคุณสมบัติของการกระจาย (distribution)  $(A \cap (B \cup C)) = (A \cap B) \cup (A \cap C)$  ของการดำเนินการฟัซซีเซตโดยวิธีมาตรฐานเป็นจริง

ทำได้โดย

$$(A \cap (B \cup C))(x) = A(x) \cap (B \cup C)(x)$$

$$= A(x) \cap (B(x) \cup C(x))$$

เนื่องจากคุณสมบัติของการหา minimum และ maximum ทำให้

$$\begin{aligned} A(x) \cap (B(x) \cup C(x)) &= (A(x) \cap B(x)) \cup (A(x) \cap C(x)) \\ &= (A \cap B)(x) \cup (A \cap C)(x) \\ &= ((A \cap B) \cup (A \cap C))(x) \end{aligned}$$

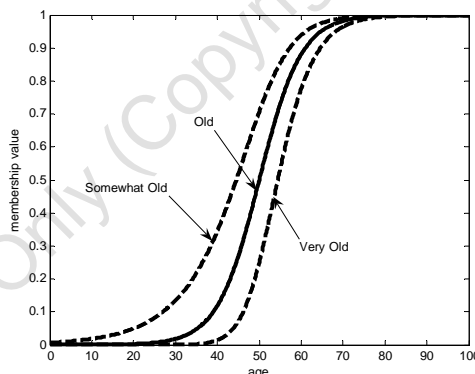
ทำให้คุณสมบัติการกระจายเป็นจริง

ในเซตแบบดั้งเดิมยังมีการดำเนินการเกี่ยวกับการเป็นเซตย่อย (inclusion) และความเท่ากันของเซต ซึ่งในฟัซซีเซตก็มีการดำเนินการเหล่านี้เช่นกัน นั่นคือ  $A \subseteq B$  ถ้า  $A(x) \leq B(x)$  สำหรับทุกๆ  $x$  และ  $A=B$  ถ้า  $A(x)=B(x)$  สำหรับทุกๆ  $x$  นั่นเอง [Klir95]

นอกเหนือจากการดำเนินการข้างต้น ยังมีการดำเนินการอีกประเภทหนึ่งที่อาจจะมี ความสำคัญในการทำอนุมาน (inference) นั่นคือ การดำเนินการเอกภาพ (unary operation) [Klir95] ซึ่งเป็นการจัดการกับฟัซซีเซตโดยตรง นั่นคือ

$$A^a(x) = (A(x))^a \quad (3.16)$$

โดยที่ถ้า  $a$  มีค่ามากกว่า 1 จะทำให้ฟัซซีเซต  $A$  มีความเป็นจำเพาะ (specific) มากขึ้นเช่น 'very  $A$ ' และถ้า  $a$  มีค่าน้อยกว่า 1 จะทำให้ฟัซซีเซต  $A$  มีความเป็นจำเพาะ (specific) น้อยลงเช่น 'somewhat  $A$ ' ดังแสดงในรูปที่ 3.15 ซึ่งมีฟัซซีเซต 'Old' ฟัซซีเซต 'Very Old' ที่มีฟังก์ชันสมาชิกเป็น 'Old' และฟัซซีเซต 'Somewhat Old' ที่มีฟังก์ชันสมาชิกเป็น 'Old'<sup>1/2</sup> นั่นเอง



รูปที่ 3.15 ฟังก์ชันสมาชิกของฟัซซีเซต 'Old' 'Very Old' และ 'Somewhat Old'

### 3.1.5 คุณสมบัติของฟัซซีเซต

ในหัวข้อนี้จะกล่าวถึงคุณสมบัติของฟัซซีเซตบางส่วน นั่นคือ [Klir95]

ซัพพอร์ต (support) ของฟัซซีเซต  $A$  คือเซตของสมาชิกในเซตสากลที่มีค่าความเป็นสมาชิกในฟัซซีเซต  $A$  ไม่เท่ากับ 0

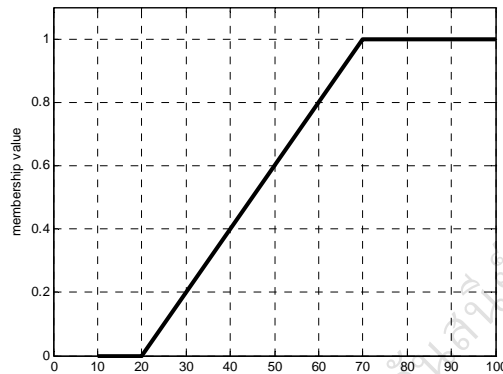
$$\text{supp}(A) = \{x \in X \mid A(x) > 0\} \quad (3.17)$$

ความสูง (height) ของฟัซซีเซต  $A$  ( $h(A)$ ) คือค่าความเป็นสมาชิกที่มากที่สุดของ  $x$  ใดๆ ในฟัซซีเซต  $A$  ถ้า  $h(A)=1$  ฟัซซีเซตนั้นจะเป็น นอร์มัล (normal) ฟัซซีเซต ถ้า  $h(A)<1$  ฟัซซีเซตนั้นจะเป็น ซับนอร์มัล (subnormal) ฟัซซีเซต และถ้า  $h(A)>1$  ฟัซซีเซตนั้นจะเป็น ซูเปอร์นอร์มัล (supernormal) ฟัซซีเซต

คอร์ (core) ของฟัซซีเซต A คือ เซตของสมาชิกในเซตสากลที่มีค่าความเป็นสมาชิกในฟัซซีเซต A เท่ากับ  $h(A)$

$$\text{core}(A) = \{x \in X \mid A(x) \geq h(A)\} \text{ หรือ } \text{core}(A) = \{x \in X \mid A(x) = h(A)\} \quad (3.18)$$

$\alpha$ -cut ของฟัซซีเซต จากรูปที่ 3.16 ถ้าดูที่ช่วงปิดของค่าความเป็นสมาชิก  $[0.2, 0.6]$  จะเห็นว่าค่าความเป็นสมาชิกช่วงนี้เป็นค่าความเป็นสมาชิกของช่วงปิด  $[30, 50]$  และเช่นเดียวกันกับที่ค่าความเป็นสมาชิก  $\leq 0.8$  เป็นค่าความเป็นสมาชิกของช่วงปิด  $[0, 60]$  สำหรับช่วงอื่นๆก็เช่นเดียวกัน ดังนั้นอาจกล่าวได้ว่า ฟัซซีเซตโดยปกติจะมีตัวร่วมเป็นครอบครัวของเซตแบบดั้งเดิม (family of crisp subsets) ของ  $X$  เสมอ



รูปที่ 3.16 ฟัซซีเซต E

ดังนั้นที่ค่าความเป็นสมาชิกที่มากกว่าหรือเท่ากับ ค่าใดๆ ( $\alpha$ ) ที่อยู่ในช่วงปิด  $[0,1]$  เราจะได้เซตแบบดั้งเดิม  ${}^\alpha A$  ( $\alpha$ -cut ของ A) ซึ่งคือ

$${}^\alpha A = \{x \in X \mid A(x) \geq \alpha\} \quad (3.19)$$

จากรูปที่ 3.16  ${}^0 E = [0,100]$  หรือ  ${}^{0.2} E = [30,100]$  หรือ  ${}^1 E = [70,100]$  จากรูปจะสังเกตเห็นได้ว่า ถ้า  $\alpha_1 < \alpha_2$  แล้ว  ${}^{\alpha_1} A \supseteq {}^{\alpha_2} A$  และ  ${}^{\alpha_1} A \cap {}^{\alpha_2} A = {}^{\alpha_2} A$  ในขณะเดียวกัน  ${}^{\alpha_1} A \cup {}^{\alpha_2} A = {}^{\alpha_1} A$

นอกเหนือจาก  $\alpha$ -cut ที่กล่าวมาข้างต้นยังมี  $\alpha$ -cut แบบเข้ม (strong  $\alpha$ -cut) นั่นคือเซตที่มีแต่สมาชิกที่มีค่าความเป็นสมาชิกมากกว่าค่า  $\alpha$  เท่านั้น นั่นคือ

$${}^{\alpha+} A = \{x \in X \mid A(x) > \alpha\} \quad (3.20)$$

จากรูปที่ 3.16  ${}^{0+} E = (20,100]$  หรือ  ${}^{0.2+} E = (30,100]$  หรือ  ${}^{1+} E = \emptyset$  แต่อย่างไรก็ตามคุณสมบัติอื่นๆยังคงอยู่ นั่นคือ ถ้า  $\alpha_1 < \alpha_2$  แล้ว  ${}^{\alpha_1+} A \supseteq {}^{\alpha_2+} A$  และ  ${}^{\alpha_1+} A \cap {}^{\alpha_2+} A = {}^{\alpha_2+} A$  ในขณะเดียวกัน  ${}^{\alpha_1+} A \cup {}^{\alpha_2+} A = {}^{\alpha_1+} A$

เซตระดับ (level set) ของฟัซซีเซต A คือเซตของ  $\alpha$ -cut เด่นๆ ของฟัซซีเซต A นั่นคือ

$$L_A = \wedge_A = \{\alpha \mid A(x) = \alpha; \exists x \in X\} \quad (3.21)$$

ถ้าต้องการแปลง  $\alpha$ -cut ให้เป็นฟัซซีเซตชนิดพิเศษ  ${}_\alpha A$  ทำได้โดย

$${}_\alpha A(x) = \alpha({}^\alpha A(x)) \quad (3.22)$$

ตัวอย่างที่ 3.6 [Klir95] จงให้  $A = 0.2/x_1 + 0.4/x_2 + 0.6/x_3 + 0.8/x_4 + 1/x_5$  และ  $L_A = \{0.2, 0.4, 0.6, 0.8, 1\}$

$$^{0.2}A = \{x_1, x_2, x_3, x_4, x_5\} = 1/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5$$

$$^{0.4}A = 0/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5$$

$$^{0.6}A = 0/x_1 + 0/x_2 + 1/x_3 + 1/x_4 + 1/x_5$$

$$^{0.8}A = 0/x_1 + 0/x_2 + 0/x_3 + 1/x_4 + 1/x_5$$

$$^1A = 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5$$

แปลง  $\alpha$ -cut เหล่านี้ให้เป็นฟังก์ชันเซตจะได้

$$_{0.2}A = 0.2/x_1 + 0.2/x_2 + 0.2/x_3 + 0.2/x_4 + 0.2/x_5$$

$$_{0.4}A = 0/x_1 + 0.4/x_2 + 0.4/x_3 + 0.4/x_4 + 0.4/x_5$$

$$_{0.6}A = 0/x_1 + 0/x_2 + 0.6/x_3 + 0.6/x_4 + 0.6/x_5$$

$$_{0.8}A = 0/x_1 + 0/x_2 + 0/x_3 + 0.8/x_4 + 0.8/x_5$$

$$_1A = 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5$$

ทฤษฎีการแยก (Decomposition Theorem) ของฟังก์ชันเซต จากตัวอย่างที่ 3.6 จะเห็นได้ว่า ถ้าเราทำการยูเนียน ฟังก์ชันเซตพิเศษ  $(_{0.2}A \cup _{0.4}A \cup _{0.6}A \cup _{0.8}A \cup _1A)$  เข้าด้วยกันจะได้ฟังก์ชันเซต  $A$  กลับมา การทำแบบนี้เป็นไปตามทฤษฎีที่จะกล่าวถึงต่อไปนี้เป็น

ทฤษฎีที่ 3.1 [Klir95] กล่าวว่าสำหรับฟังก์ชันเซต  $A$  ใดๆ ที่เป็นสมาชิกของ  $\tilde{\mathcal{P}}(X)$  ฟังก์ชันพาวเวอร์เซตของ  $X$  จะได้ว่า

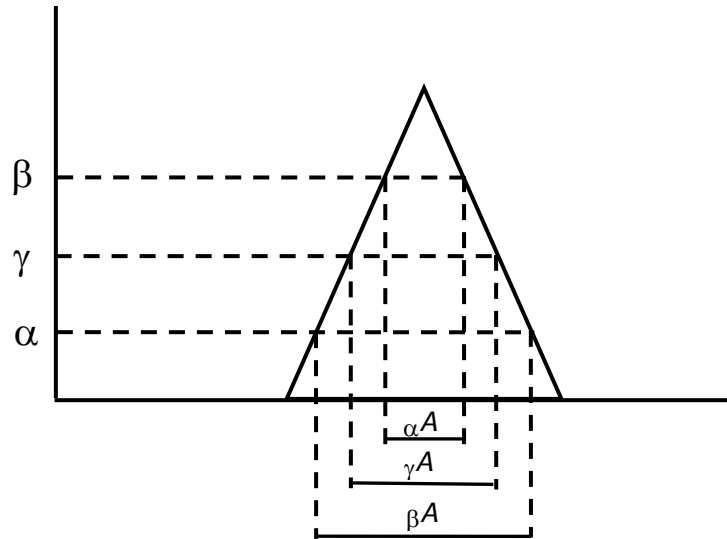
$$A = \bigcup_{\alpha \in [0,1]} \alpha A \quad (3.23)$$

โดยที่  $\alpha A$  ถูกนิยามดังสมการที่ 3.22 และการยูเนียนเป็นการยูเนียนด้วยวิธีมาตรฐาน ตัวอย่างของทฤษฎีนี้แสดงในรูปที่ 3.17 ซึ่งแสดงแค่  $\alpha$ -cut ของค่า  $\alpha$  3 ค่าเท่านั้น และ  $A$  เกิดจากการยูเนียน  $\alpha A$  สำหรับ ทุกๆ  $\alpha$

ทฤษฎีที่ 3.2 [Klir95] กล่าวว่าสำหรับฟังก์ชันเซต  $A$  ใดๆ ที่เป็นสมาชิกของ  $\tilde{\mathcal{P}}(X)$  ฟังก์ชันพาวเวอร์เซตของ  $X$  จะได้ว่า

$$A = \bigcup_{\alpha \in [0,1]} \alpha_+ A \quad (3.24)$$

โดยที่  $\alpha_+ A$  ถูกนิยามเป็น  $\alpha_+ A(x) = \alpha^{(\alpha_+)} A(x)$  และการยูเนียนเป็นการยูเนียนด้วยวิธีมาตรฐาน



รูปที่ 3.17 ตัวอย่างของทฤษฎีที่ 3.1

ทฤษฎีที่ 3.3 [Klir95] กล่าวว่าสำหรับฟัซซีเซต  $A$  ใดๆ ที่เป็นสมาชิกของ  $\tilde{\mathcal{P}}(X)$  ฟัซซีพาวเวอร์เซตของ  $X$  จะได้ว่า

$$A = \bigcup_{\alpha \in \Lambda_A} \alpha A \quad (3.25)$$

โดยที่  $\Lambda_A$  เป็นเซตระดับของฟัซซีเซต  $A$  และ  $\alpha A$  ถูกนิยามดังสมการที่ 3.22 และการยูเนียนเป็นการยูเนียนด้วยวิธีมาตรฐาน ตัวอย่างของทฤษฎีนี้แสดงในรูปที่ 3.18 ซึ่งในรูป  $\Lambda_A = \{0, 0.3, 0.6, 1\}$  และ  ${}_0A = \emptyset$  ดังนั้น  $A$  จึงถูกแทนได้ด้วย  ${}_{0.3}A$   ${}_{0.6}A$  และ  ${}_1A$  นั่นเอง

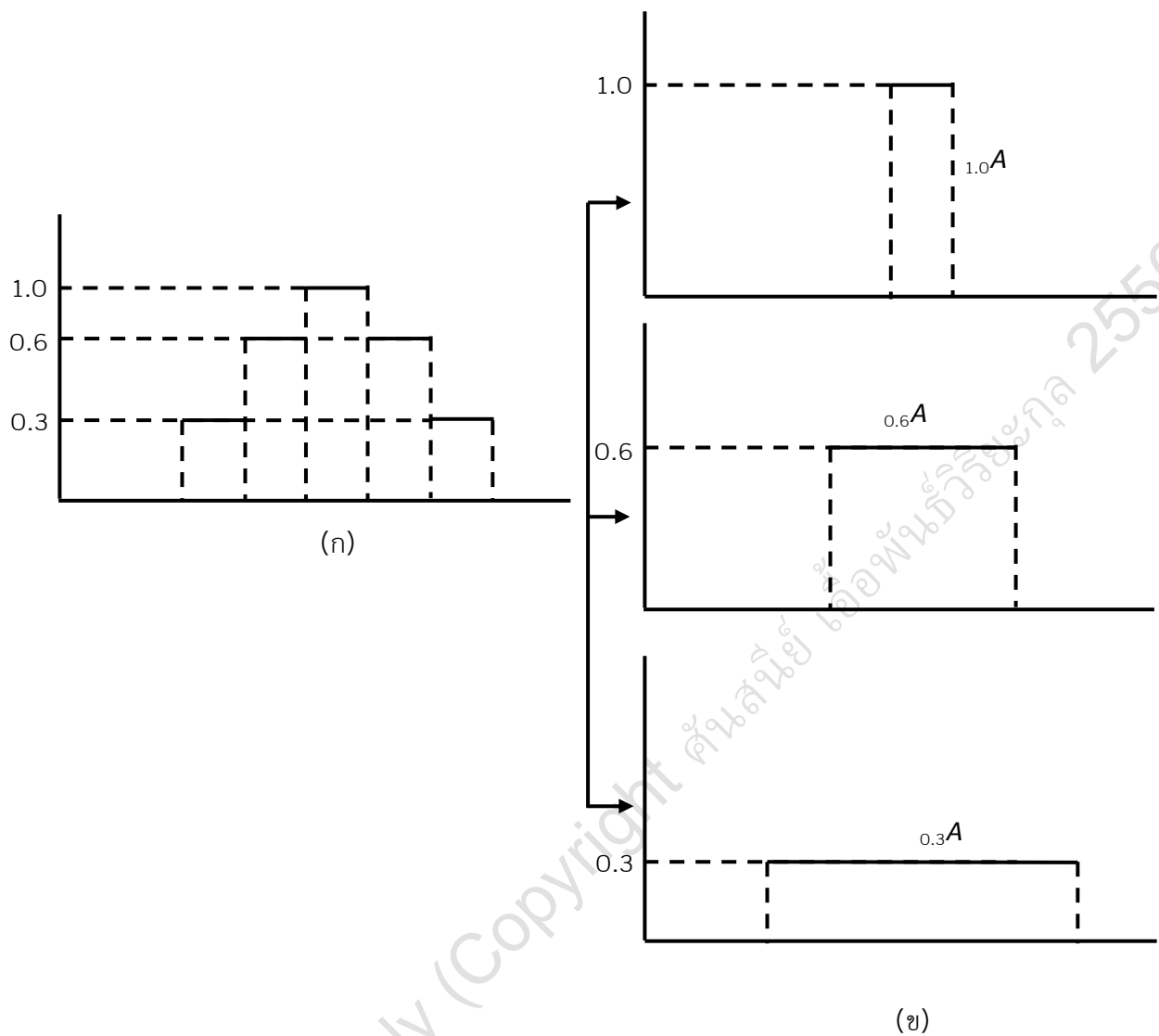
คอนเวกฟัซซีเซต (Convex Fuzzy Set) [Klir95] ฟัซซีเซต  $A$  เป็นคอนเวกฟัซซีเซตก็ต่อเมื่อ  $A(\lambda \vec{r} + (1-\lambda)\vec{s}) \geq \min(A(\vec{r}), A(\vec{s}))$  นั่นคือค่าความเป็นสมาชิกของจุดที่อยู่บนเส้นที่ลากระหว่างจุด  $\vec{r}$  และ  $\vec{s}$  มากกว่าหรือเท่ากับค่าความเป็นสมาชิกที่น้อยที่สุดของทั้งสองจุด ดังรูปที่ 3.19 ที่แสดงแผนภาพคอนทัวร์ (contour diagram) และเส้นประแสดงค่าความเป็นสมาชิกของจุดที่อยู่ระหว่าง  $\vec{r}$  และ  $\vec{s}$  จะเห็นได้ว่า แผนภาพคอนทัวร์ในรูปที่ 3.19(ก) เป็นแผนภาพคอนทัวร์ของคอนเวกฟัซซีเซต เพราะไม่มีจุดบนเส้นประที่มีค่าความเป็นสมาชิกน้อยกว่า  $\vec{r}$  และ  $\vec{s}$  และรูปที่ 3.19(ข) แสดงฟัซซีเซตที่ไม่ใช่คอนเวกฟัซซีเซต เพราะมีบางจุดบนเส้นที่หลุดออกไปจากคอนทัวร์นั่นคือมีค่าความเป็นสมาชิกเท่ากับ 0 ซึ่งน้อยกว่าของ  $\vec{r}$  และ  $\vec{s}$  หรือถ้าจะทำให้การพิจารณาคอนเวกฟัซซีเซตทำได้ง่าย เราสามารถใช้คุณสมบัติของ เซตดั้งเดิมผ่านทาง  $\alpha$ -cut ได้ นั่นคือ ถ้าฟัซซีเซต  $A$  เป็นคอนเวกฟัซซีเซต แล้ว (ก)  $\alpha$ -cut ของทุกค่า  $\alpha$  จะต่อเนื่องกันตลอด และ (ข) แต่ละ  $\alpha$ -cut จะเป็นคอนเวกในแนวคิดของเซตดั้งเดิม

ตัวเลขฟัซซี (Fuzzy Number) [Klir95] คือฟัซซีเซตที่เป็นคอนเวกฟัซซีเซต ต่อเนื่องแบบพิชไวส์ (piece-wise continuous) และ นอร์แมล (normal) ฟัซซีเซต

สเกลาร์คาร์ดินาลิตี้ (Scalar Cardinality) หรือ การนับซิกมา (Sigma Count) ( $|A|$ ) อธิบายได้ด้วย [Klir95]

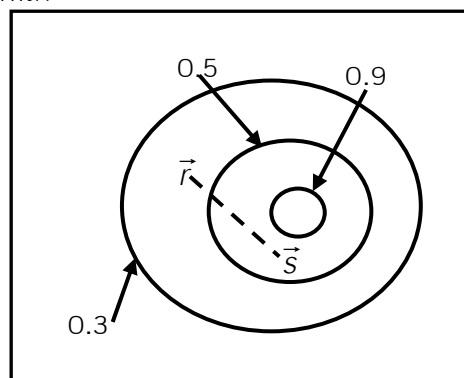
$$\text{ฟัซซีเซตไม่ต่อเนื่อง (discrete fuzzy set): } |A| = \sum_x A(x) \quad (3.26)$$





รูปที่ 3.18 ตัวอย่างของทฤษฎีที่ 3.3 [Klir95] (ก) ฟังก์ชัน  $A$  (ข) การแยกของฟังก์ชัน  $A$  ให้เป็น  $0.3A$ ,  $0.6A$  และ  $1A$

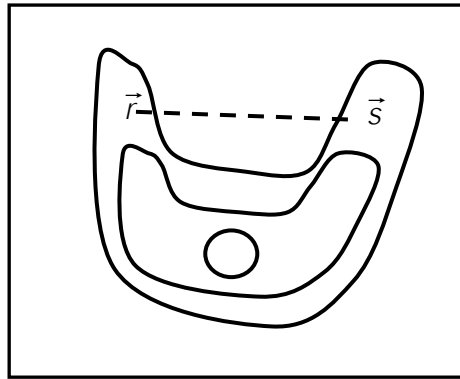
น้ำหนักร



ส่วนสูง

(ก)

น้ำหมัก



ส่วนสูง

(ข)

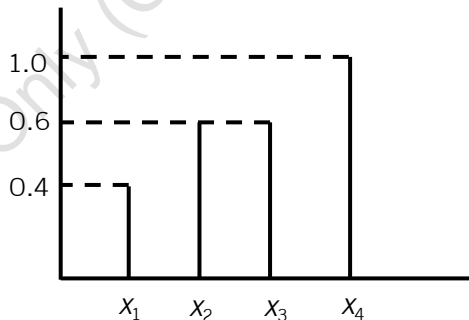
รูปที่ 3.19 แผนภาพคอนทัวร์ของ (ก) คอนเวกฟัชซีเซต (ข) ฟัชซีเซตที่ไม่คอนเวก

ฟัชซีเซตต่อเนื่อง (continuous fuzzy set):  $|A| = \int_x A(x) dx$  (3.27)

ฟัชซีคาดีนาลิตี (Fuzzy Cardinality)  $|\tilde{A}|$  [Klir95] อธิบายได้ด้วย

$$|\tilde{A}|(\alpha A) = \alpha, \quad \forall \alpha \quad (3.28)$$

ตัวอย่างที่ 3.7 จากฟังก์ชันสมาชิกของฟัชซีเซต  $A$  ในรูปที่ 3.20 จะได้ว่า  $|^{0.4}A| = 4$ ,  $|^{0.6}A| = 3$ ,  $|^{1.0}A| = 1$  ดังนั้น  $|\tilde{A}| = \frac{0.4}{4} + \frac{0.6}{3} + \frac{1.0}{1}$



รูปที่ 3.20 ฟัชซีเซตไม่ต่อเนื่อง

### 3.2 ระบบอนุมานฟัชซี (Fuzzy Inference System)

จากความรู้พื้นฐานในหัวข้อที่แล้ว เราสามารถทำความเข้าใจกับระบบอนุมานฟัชซีที่จะกล่าวถึงในหัวข้อนี้ได้ โดยในหัวข้อนี้จะกล่าวถึง เฮดจ์ภาษา (linguistic hedges) การหาเหตุผลโดยประมาณ (approximate reasoning) และวิธีการในระบบควบคุมฟัชซี (approached to fuzzy control) ซึ่งเป็นระบบอนุมานฟัชซีในนั่นเอง

### 3.2.1 เสด็จภาษา (Linguistic Hedges)

ส่วนประกอบในการสร้างระบบอนุมานฟัซซี คือการสร้างฟังก์ชันสมาชิก ซึ่งเสด็จภาษา [Klir97] เป็นกระบวนการหนึ่ง ที่ถูกใช้ในการสร้างฟังก์ชันสมาชิกเหล่านั้น

ถ้ามีพจน์ “Tina is young” และพจน์ “Tina is very young” ตัวดัดแปร (modifier) “very” ทำให้พจน์ที่ 2 เน้นไปที่ young มากขึ้น นั่นคือ very ทำหน้าที่เป็นเสด็จ หรือที่เราเรียกว่า เสด็จภาษา (linguistic hedges) นั่นเอง ดังนั้นเสด็จภาษาจึงเป็นพจน์ภาษาชนิดพิเศษที่ทำการดัดแปรพจน์ภาษาอื่น เช่น very, more or less, fairly หรือ extreme เป็นต้น

เสด็จภาษาใดๆ เป็นการดำเนินการเอกภาพ (unary operation) ของช่วงปิด  $[0,1]$  เช่น สำหรับ เพรดิเคตแบบฟัซซี (fuzzy predicate)  $A$  ที่อยู่บนเซตสากล  $X$  และ ตัวดัดแปร  $h$  ซึ่งถูกใช้แทนเสด็จภาษา  $H$  เพรดิเคตแบบฟัซซีที่ถูกดัดแปร  $HA$  มีฟังก์ชันสมาชิกสำหรับ  $x \in X$  เป็น

$$HA(x) = h(A(x)) \quad (3.29)$$

ตัวอย่างเช่น ฟังก์ชันในการเน้น (concentration function) เป็น  $HA(x) = (A(x))^2$  ซึ่งโดยปกติใช้กับเสด็จภาษา very ส่วนฟังก์ชันที่ทำให้พจน์ขยาย (dilation function) เป็น  $HA(x) = (A(x))^{1/2}$  ซึ่งโดยปกติใช้กับเสด็จภาษา more or less หรือ fairly หรือ somewhat และถ้าต้องการใช้ฟังก์ชันที่อยู่ระหว่าง  $A$  และ very  $A$  ที่เรียกว่า plus สามารถใช้ฟังก์ชัน  $HA(x) = (A(x))^{1.25}$  ได้และบางครั้งสำหรับเสด็จภาษา slightly สามารถใช้ฟังก์ชัน  $\text{int}[\text{plus } A \text{ and not very } A]$  โดยที่ฟังก์ชัน  $\text{int}$  มีค่าเป็น

$$HA(x) = \begin{cases} 2A(x)^2 & \text{if } A(x) \in \left[0, \frac{1}{2}\right] \\ 1 - 2(1 - A(x))^2 & \text{otherwise} \end{cases} \quad (3.30)$$

### 3.2.2 การหาเหตุผลโดยประมาณ (Approximate Reasoning)

หนึ่งในเป้าหมายของลอจิกแบบดั้งเดิมคือการหาเหตุผลโดยใช้กฎของการส่อความ เช่น modus ponens และ hypothetical syllogism ที่เป็นรูปแบบของการส่อความ และในหัวข้อนี้เราจะพูดถึง modus ponens เท่านั้น ซึ่งรูปแบบหนึ่งของ modus ponens ที่เป็นประโยคซ้ำความ (tautology) คือ  $((p \rightarrow q) \wedge p) \rightarrow q$  การส่อความแบบนี้เป็นไปได้เพราะเราคิดว่าข้อตั้ง (premise)  $p$  ตัวที่ 2 เป็นตัวเดียวกับ  $p$  ตัวที่ 1 และข้อตาม (consequent)  $q$  ทั้งสองที่เหมือนกัน แต่ถ้าไม่เหมือนกันก็ไม่สามารถหาคำตอบได้ และอีกปัญหาหนึ่งก็คือกฎนี้สนใจแค่ค่าความเป็นจริง (truth) 0 (เป็นเท็จ) และ 1 (เป็นจริง) เท่านั้น ดังนั้นเราสามารถเลียนแบบการหาเหตุผลแบบนี้ในชีวิตประจำวันด้วยการใช้ การหาเหตุผลโดยประมาณ (approximate reasoning) [Klir97]

การหาเหตุผลโดยประมาณ (approximate reasoning) เป็นการประยุกต์ใช้ทฤษฎีฟัซซีเซตที่สำคัญอันหนึ่ง นั่นคือเลียนแบบการหาเหตุผลของมนุษย์ และสามารถใช้งานได้ในสถานะที่ไม่แน่นอนตัวอย่างของการเหตุผลแบบนี้คือ

กฎ:	If a book is large, then it is expensive.	
ความจริง:	Book x is fairly large	
บทสรุป:	Book x is fairly expensive	(3.31)

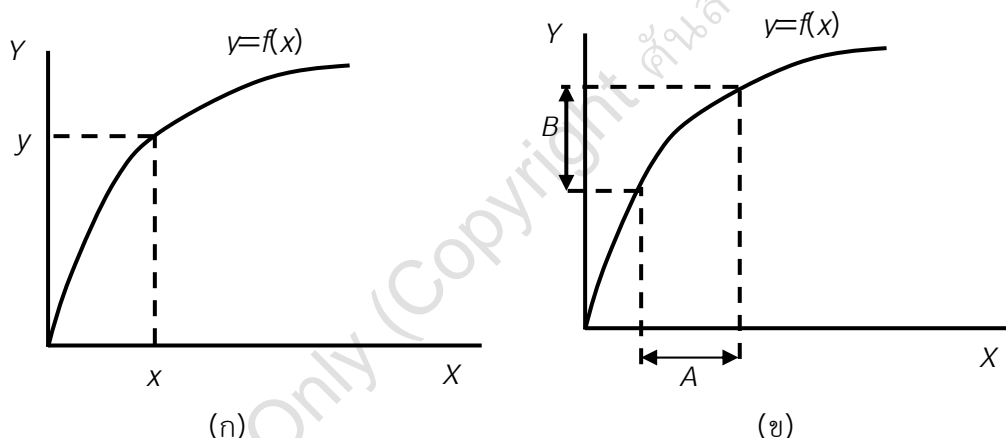
ในตัวอย่างนี้ไม่สามารถใช้การหาเหตุผลแบบดั้งเดิมได้เนื่องจากมีแนวคิดที่เป็นแนวคิดแบบฟัซซี เช่น large, fairly large และ expensive และอีกเหตุผลหนึ่งคือข้อตั้งใน modus ponens แบบดั้งเดิมต้องเหมือนกันกับข้อตั้งใน ความจริง ซึ่งในตัวอย่างนี้ไม่เป็นเช่นนั้น

การหาเหตุผลในสมการที่ 3.31 เป็นตัวอย่างอันหนึ่งของนัยทั่วไปของ modus ponens (generalized modus ponens) ในการหาเหตุผลโดยประมาณ ซึ่งโดยปกติมีรูปแบบเป็น

$$\begin{array}{ll} \text{กฎ:} & \text{If } \chi \text{ is } A, \text{ then } Y \text{ is } B \\ \text{ความจริง:} & \underline{\chi \text{ is } A'} \\ \text{บทสรุป:} & Y \text{ is } B' \end{array} \quad (3.32)$$

โดยที่  $\chi$  และ  $Y$  เป็นตัวแปรที่อยู่ในเซตสากล  $X$  และ  $Y$  และ  $A$  และ  $A'$  เป็นฟัซซีเซตที่อาจจะไม่เหมือนกันบน  $X$  ในขณะที่  $B$  และ  $B'$  เป็นฟัซซีเซตที่อาจจะไม่เหมือนกันบน  $Y$

จากสมการนัยทั่วไปของ modus ponens ในสมการที่ 3.32 [Klir95] ให้  $\chi$  และ  $Y$  เป็นตัวแปรที่อยู่ในเซตสากล  $X$  และ  $Y$  และสำหรับทุก  $x \in X$  และ  $y \in Y$  มีความสัมพันธ์กันโดยที่  $y=f(x)$  และเช่นเดียวกับตัวแปร  $\chi=x$  สามารถหา  $Y=y=f(x)$  ได้ดังรูป 3.21(ก) และถ้า  $\chi$  อยู่ในเซต  $A$  สามารถหา  $Y$  ที่อยู่ในเซต  $B = \{y \in Y \mid y=f(x), x \in A\}$  ดังรูป 3.21(ข)



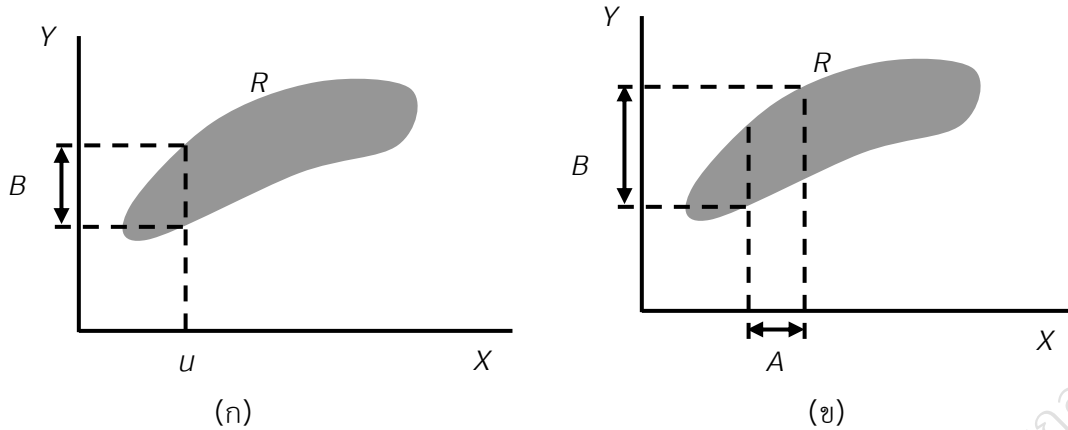
รูปที่ 3.21 ความสัมพันธ์ระหว่าง 2 ตัวแปร (ก)  $x \rightarrow y$  โดยที่  $y=f(x)$  (ข)  $A \rightarrow B$  โดยที่

$$B = \{y \in Y \mid y=f(x), x \in A\}$$

ถ้าตัวแปรเหล่านี้มีความสัมพันธ์  $R$  บน  $X \times Y$  ถ้าให้  $\chi=u$  และความสัมพันธ์  $R$  สามารถหา  $Y \in B$  ที่  $B = \{y \in Y \mid \langle x, y \rangle \in R\}$  ได้ดังรูปที่ 3.22(ก) และเช่นเดียวกันถ้า  $\chi \in A$  สามารถหา  $Y \in B$  โดยที่  $B = \{y \in Y \mid \langle x, y \rangle \in R, x \in A\}$  ได้ดังรูปที่ 3.22(ข) โดยที่  $B$  มีฟังก์ชันลักษณะ (characteristic function) เป็น [Klir95]

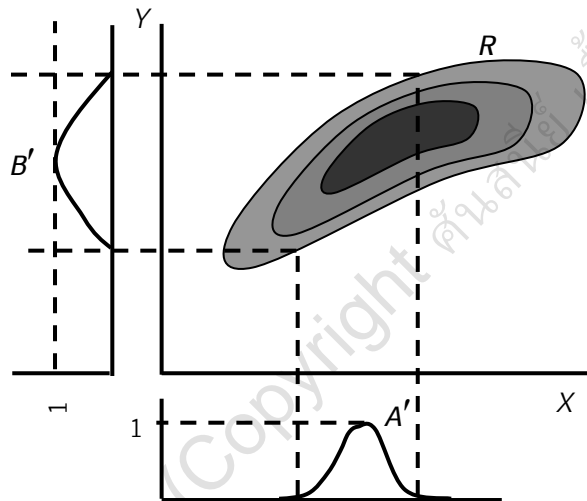
$$X_B(y) = \sup_{x \in X} \min[X_A(x), X_R(x, y)] \quad (3.33)$$

สำหรับทุก  $y \in Y$  โดยที่  $\chi_A$ ,  $\chi_B$  และ  $\chi_R$  เป็นฟังก์ชันลักษณะของเซต  $A$ ,  $B$  และ  $R$



รูปที่ 3.22 ความสัมพันธ์ระหว่าง 2 ตัวแปรที่ไม่ใช่ฟังก์ชันและหาฟังก์ชันลักษณะได้จากสมการที่

$$3.33 \quad (ก) \chi = u \quad (ข) \chi \in A$$



รูปที่ 3.23 การประกอบของกฎการส่อความ (compositional rule of inference)

ถ้าความสัมพันธ์  $R$  เป็นความสัมพันธ์แบบฟัซซีบน  $X \times Y$  โดยที่  $A$  และ  $A'$  เป็นฟัซซีเซตบน  $X$  และ  $B$  และ  $B'$  เป็นฟัซซีเซตบน  $Y$  ดังนั้นถ้ารู้ความสัมพันธ์  $R$  (ซึ่งเป็นความสัมพันธ์ระหว่าง  $A$  และ  $B$ ) และ  $A'$  เราสามารถหา  $B'$  ในบทสรุปได้โดยใช้ [Klir95]

$$B'(y) = \sup_{x \in X} \min[A'(x), R(x, y)] \quad (3.34)$$

สำหรับทุก  $y \in Y$  ซึ่งสมการที่ 3.34 ที่จริงก็คือ

$$B' = A' \bullet R \quad (3.35)$$

โดยที่  $\bullet$  คือ ตัวดำเนินการของการประกอบ (composition operator) และสมการที่ 3.35 นี้ถูกเรียกว่า การประกอบของกฎการส่อความ (compositional rule of inference) นั่นคือถ้ามีการประกอบของกฎการส่อความ ไปใช้ในการหาบทสรุปได้ แสดงว่ากฎนั้นสามารถแทนได้ด้วยความสัมพันธ์ และความหมายของสมการที่ 3.34 คือการฉาย (Projection) ลงในทิศทาง  $Y$  ของอินเตอร์เซกชัน (intersection) ระหว่างการขยายแบบไซลินดริก (cylindrical extension) ของ  $A'$  ไปในทิศทาง  $Y$  และความสัมพันธ์  $R$  ซึ่งนิยามของการฉาย คือ ให้ความสัมพันธ์แบบฟัซซีเป็นความสัมพันธ์ที่เป็นซัพเซตของคาร์ทีเซียนโปรดักส์  $X = X_1 \times X_2 \times \dots \times X_n$  ( $RL(x_1, x_2, \dots, x_n)$ ) และ

ต้องการสร้างความสัมพันธ์แบบฟัซซีอันใหม่บนคาร์ทีเซียนโปรดักส์  $Y$  ซึ่ง  $Y = \{x_j | j \in J \subset N_n\}$  โดยที่  $|J| = r$  ดังนั้นการฉายความสัมพันธ์  $RL$  ลงบน  $Y$  เป็น

$$[RL \downarrow Y](\vec{y}) = \max_{\vec{x} \succ \vec{y}} RL(\vec{x}) \quad (3.36)$$

โดยที่  $\vec{y}$  เป็น  $r$  ทูเพิลใน  $Y$  และ  $\vec{x}$  เป็น  $n$  ทูเพิลใน  $X$  และ  $\vec{x} \succ \vec{y}$  คือ  $\vec{y}$  เป็นซับซีควเอน (subsequence) ของ  $\vec{x}$  (ถ้า  $y_j = x_j$  สำหรับทุก  $j \in J$  ตัวอย่างเช่น  $\vec{x} = (x_1, x_2, \dots, x_n)$  และ  $\vec{y} = (y_1, y_2, y_3)$  ที่  $J = (2, 3, 6)$  ดังนั้น  $\vec{y} \prec \vec{x}$  ก็ต่อเมื่อ  $y_1 = x_2, y_2 = x_3$  และ  $y_3 = x_6$ ) ส่วนนิยามของการขยายแบบไชลินดิก คือให้คาร์ทีเซียนโปรดักส์  $X$  และ  $Y$  เป็นเซตที่กล่าวข้างต้น และ  $RL$  เป็นความสัมพันธ์แบบฟัซซีที่เป็นซับเซตของ  $Y$  ให้  $[RL \uparrow X - Y]$  เป็นการขยายแบบไชลินดิกของ  $RL$  ไปยังเซต  $X_i$  โดยที่  $i \in N_n$  ที่อยู่ใน  $X$  แต่ไม่ได้อยู่ใน  $Y$

$$[RL \uparrow X - Y](\vec{x}) = RL(\vec{y}) \quad (3.37)$$

สำหรับแต่ละ  $\vec{x}$  ที่  $\vec{x} \succ \vec{y}$

ดังนั้นในบางครั้งการประกอบการส่อความนี้ถูกเรียกว่า การประกอบการส่อความ max-min (max-min composition rule of inference) และการประกอบของกฎการส่อความนี้แสดงในรูปที่ 3.23 แต่ในบางครั้งการประกอบการส่อความที่ถูกใช้จะเป็นการประกอบการส่อความ max-product (max-product composition rule of inference) ดังสมการนี้

$$B'(y) = \sup_{x \in X} [A'(x) \times R(x, y)] \quad (3.38)$$

### ตัวอย่างที่ 3.8 สมมุติให้มี

กฎ: If  $x$  and  $y$  are approximately equal.

ความจริง:  $x$  is little.

บทสรุป: ?

และสมมุติให้  $A$  เป็นฟัซซีเซต little ที่มีฟังก์ชันสมาชิกเป็น

$$A = \{(1,1), (2,0.6), (3,0.2), (4,0)\}$$

และความสัมพันธ์ของกฎคือ

$$R = \begin{bmatrix} 1.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 1.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 1.0 & 0.5 \\ 0.0 & 0.0 & 0.5 & 1.0 \end{bmatrix}$$

โดยที่สมมุติให้เซตสากล  $X$  และ  $Y$  เท่ากันจะหาบทสรุปได้จาก

$$B = A \bullet R$$

$$B = \begin{bmatrix} 1.0 & 0.6 & 0.2 & 0.0 \end{bmatrix} \bullet \begin{bmatrix} 1.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 1.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 1.0 & 0.5 \\ 0.0 & 0.0 & 0.5 & 1.0 \end{bmatrix}$$

โดยที่ตัวดำเนินการของการประกอบเป็น max-min composition นั้นเอง

$$B = \begin{bmatrix} 1.0 & 0.6 & 0.5 & 0.2 \end{bmatrix}$$

ดังนั้นค่าความเป็นสมาชิกของ 2 ของ  $B$  เกิดจากการหา  $\max(\min(1,0.5), \min(0.6,1), \min(0.2, 0.5), \min(0.0,0.0))$  ซึ่งเท่ากับ 0.6 และค่าความเป็นสมาชิกของสมาชิกตัวอื่นก็หาเช่นเดียวกัน

เนื่องจากความสัมพันธ์แบบฟัซซีซ่อนอยู่ในความหมายของพจน์ที่มีเงื่อนไข (conditional proposition) หรือ กฎใน นัยทั่วไปของ modus ponens นั่นคือ

$$p: \text{If } X \text{ is } A, \text{ then } Y \text{ is } B$$

สำหรับทุก  $x \in X$  และ  $y \in Y$  จะได้ความสัมพันธ์ฟัซซีเป็น

$$R(x,y) = I(A(x), B(y)) \quad (3.39)$$

โดยที่  $I$  คือการสื่อความแบบฟัซซี (fuzzy implication) เช่น [Klir95]

1. การสื่อความแบบฟัซซีของ Lukasiewicz

$$I(A(x), B(y)) = \min[1, 1 - A(x) + B(y)] \quad (3.40)$$

2. การสื่อความแบบฟัซซีของ Zadeh (1973)

$$I(A(x), B(y)) = \max((1 - A(x)), \min(A(x), B(y))) \quad (3.41)$$

3. การสื่อความแบบฟัซซีของ Kleen-Dienes (1938, 1949)

$$I(A(x), B(y)) = \max((1 - A(x)), B(y)) \quad (3.42)$$

4. การสื่อความแบบฟัซซีของ Mamdani (Correlation-min)

$$I(A(x), B(y)) = \min(A(x), B(y)) \quad (3.43)$$

5. การสื่อความแบบฟัซซี Correlation-product

$$I(A(x), B(y)) = A(x) \times B(y) \quad (3.44)$$

6. การสื่อความแบบฟัซซีของ Goedel (1976)

$$I(A(x), B(y)) = \begin{cases} 1 & \text{if } A(x) \leq B(y) \\ B(y) & \text{if } A(x) > B(y) \end{cases} \quad (3.45)$$

ส่วนการหานัยทั่วไปการสื่อความแบบอื่นเช่น นัยทั่วไปของ modus tollens (generalized modus tollens) ซึ่งมีลักษณะเป็น [Klir95]

$$\text{กฎ:} \quad \text{If } X \text{ is } A, \text{ then } Y \text{ is } B$$

$$\text{ความจริง:} \quad Y \text{ is } B'$$

$$\text{บทสรุป:} \quad X \text{ is } A' \quad (3.46)$$

หาได้โดยใช้การประกอบของกฎการสื่อความดังนี้คือ

$$A' = B' \bullet R^T \quad (3.47)$$

โดยที่  $R$  หาได้เช่นเดียวกับกรณีของ modus ponens

ตัวอย่างที่ 3.9 [Klir95] ให้เซตสากล  $X = \{x_1, x_2, x_3\}$  และ  $Y = \{y_1, y_2\}$  และให้กฎ If  $\chi$  is  $A$ , then  $Y$  is  $B$  มาโดยที่ฟัซซีเซต  $A = 0.5/x_1 + 1/x_2 + 0.6/x_3$  และ  $B = 1/y_1 + 0.4/y_2$  และความจริง  $A'$  ให้มาโดยที่  $A' = 0.6/x_1 + 0.9/x_2 + 0.7/x_3$  และสมมุติใช้การส่อความแบบฟัซซีของ Lukasiewicz ต้องการหาบทสรุป  $Y$  is  $B'$  สำหรับกรณีของ นัยทั่วไปของ modus ponens ใช้การส่อความแบบฟัซซีของ Lukasiewicz จะได้

$$R = \frac{1}{\langle x_1, y_1 \rangle} + \frac{0.9}{\langle x_1, y_2 \rangle} + \frac{1}{\langle x_2, y_1 \rangle} + \frac{0.4}{\langle x_2, y_2 \rangle} + \frac{1}{\langle x_3, y_1 \rangle} + \frac{0.8}{\langle x_3, y_2 \rangle}$$

$$\text{ดังนั้นจากสมการที่ 3.34 จะได้ } B' = \begin{bmatrix} 0.6 & 0.9 & 0.7 \end{bmatrix} \bullet \begin{bmatrix} 1.0 & 0.9 \\ 1.0 & 0.4 \\ 1.0 & 0.8 \end{bmatrix}$$

$$\text{ดังนั้นบทสรุปที่ได้ คือ } B' = \begin{bmatrix} 0.9 & 0.7 \end{bmatrix} \text{ ซึ่งคือ } B' = \frac{0.9}{y_1} + \frac{0.7}{y_2}$$

สำหรับกรณีของ นัยทั่วไปของ modus tollens ให้ความจริงที่ให้มา  $Y$  is  $B'$  โดยที่  $B' = 0.9/y_1 + 0.7/y_2$  และสมมุติใช้การส่อความแบบฟัซซีของ Lukasiewicz เช่นกัน ต้องการหาบทสรุป  $\chi$  is  $A'$

$$\begin{aligned} \text{จากสมการ 3.47 จะได้ว่า } A' &= \begin{bmatrix} 0.9 & 0.7 \end{bmatrix} \bullet \begin{bmatrix} 1.0 & 0.9 \\ 1.0 & 0.4 \\ 1.0 & 0.8 \end{bmatrix}^T \\ A' &= \begin{bmatrix} 0.9 & 0.7 \end{bmatrix} \bullet \begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 0.9 & 0.4 & 0.8 \end{bmatrix} \\ A' &= \begin{bmatrix} 0.9 & 0.9 & 0.9 \end{bmatrix} \end{aligned}$$

$$\text{ซึ่งก็คือ } A' = \frac{0.9}{x_1} + \frac{0.9}{x_2} + \frac{0.9}{x_3} \text{ นั่นเอง}$$

พจน์ที่มีเงื่อนไขที่อยู่ในรูปแบบของ If  $\chi$  is  $A$ , then  $Y$  is  $B$  อาจจะมองได้เป็น If  $\chi$  is  $A$ , then  $Y$  is  $B$  else  $V$  is **UNKNOWN** โดยที่  $\chi$   $Y$  และ  $V$  เป็นตัวแปรในเซตสากล  $X$   $Y$  และ  $V$  และ  $A$   $B$  และ **UNKNOWN** เป็นฟัซซีเซตบน  $X$   $Y$  และ  $V$  นั่นเองโดยที่ **UNKNOWN** เป็นฟัซซีเซตที่สมาชิกทุกตัวในเซตสากลมีค่าความเป็นสมาชิกเป็น 1 และการหาความสัมพันธ์ของพจน์ที่มีเงื่อนไขแบบนี้ทำได้โดย [Ross04]

$$R = A \times B + \bar{A} \times \text{UNKNOWN} \quad (3.48)$$



โดยที่ใช้การหา minimum แทนการคูณและ หา maximum แทนการบวก และถ้าเป็นพจน์ที่มีเงื่อนไขเป็น If  $\chi$  is  $A$ , then  $Y$  is  $B$  else  $Z$  is  $C$  โดยที่  $Z$  เป็นตัวแปรในเซตสากล  $Z$  และ  $C$  เป็นฟัซซีเซตบน  $Z$  การหาความสัมพันธ์ของพจน์นี้จะเป็น

$$R = A \times B + \bar{A} \times C \quad (3.49)$$

เนื่องจาก  $B \subseteq UNKNOWN$  ทำให้  $\bar{A} \times B \subseteq \bar{A} \times UNKNOWN$  ดังนั้นเราสามารถพิสูจน์ได้ว่าถ้า  $A$  และ  $B$  เป็นคริสป์ สมการที่ 3.48 จะกลายเป็น  $\bar{A} \vee B$  ซึ่งเท่ากับการ implication ในลอจิกแบบดั้งเดิม และถ้าอินพุตที่เข้ามาเป็น  $A$  สำหรับความสัมพันธ์ตามสมการที่ 3.48 จะได้เอาต์พุตเป็น

$$B' = A \bullet (A \times B + \bar{A} \times UNKNOWN) \quad (3.50ก)$$

$$B' = (A \bullet A \times B) + (A \bullet \bar{A} \times UNKNOWN) \quad (3.50ข)$$

ซึ่งถ้า  $A$  เป็นนอร์มัล (normal) ฟัซซีเซต จะได้ว่า  $A \bullet A$  เท่ากับ 1 และ  $A \bullet \bar{A}$  เท่ากับ  $\beta$  (ค่าคงที่ค่าหนึ่ง) ดังนั้น  $B' = B + \beta \times UNKNOWN$  และถ้า  $A$  เป็นคริสป์เซตจะเห็นได้ว่า  $A \bullet A$  เท่ากับ 1 และ  $A \bullet \bar{A}$  เท่ากับ 0 ดังนั้น  $B' = B$

เช่นเดียวกันสำหรับสมการที่ 3.49 จะได้ว่า

$$B' = A \bullet (A \times B + \bar{A} \times C)$$

$$B' = (A \bullet A \times B) + (A \bullet \bar{A} \times C)$$

ซึ่งถ้า  $A$  เป็นนอร์มัล (normal) ฟัซซีเซต จะได้ว่า  $A \bullet A$  เท่ากับ 1 และ  $A \bullet \bar{A}$  เท่ากับ  $\beta$  (ค่าคงที่ค่าหนึ่ง) ดังนั้น  $B' = B + \beta C$  และถ้า  $A$  เป็นคริสป์เซตจะเห็นได้ว่า  $A \bullet A$  เท่ากับ 1 และ  $A \bullet \bar{A}$  เท่ากับ 0 ดังนั้น  $B' = B$

แต่ถ้าอินพุตที่เข้ามาเป็น  $\bar{A}$  จะได้ว่า

$$B' = \bar{A} \bullet (A \times B + \bar{A} \times C)$$

$$B' = (\bar{A} \bullet A \times B) + (\bar{A} \bullet \bar{A} \times C)$$

ซึ่งถ้า  $\bar{A}$  เป็นนอร์มัล (normal) ฟัซซีเซต จะได้ว่า  $\bar{A} \bullet A$  เท่ากับ  $\beta$  (ค่าคงที่ค่าหนึ่ง) และ  $\bar{A} \bullet \bar{A}$  เท่ากับ 1 ดังนั้น  $B' = \beta B + C$  และถ้า  $\bar{A}$  เป็นคริสป์เซตจะเห็นได้ว่า  $\bar{A} \bullet A$  เท่ากับ 1 และ  $\bar{A} \bullet \bar{A}$  เท่ากับ 0 ดังนั้น  $B' = C$

ตัวอย่างที่ 3.10 ให้เซตสากล  $X$   $Y$  และ  $Z$  เป็นเซตเดียวกันคือ  $\{1,2,3\}$  และฟัซซีเซต  $A = 1/1 + 0.4/2$  ฟัซซีเซต  $B = 0.4/2 + 1/3$  และ ฟัซซีเซต  $C = 1/1 + 0.6/2$

ดังนั้นความสัมพันธ์ของพจน์ If  $\chi$  is  $A$ , then  $Y$  is  $B$  else  $Z$  is  $C$  ได้เป็น

$$R = \begin{bmatrix} 1.0 \\ 0.4 \\ 0.0 \end{bmatrix} \begin{bmatrix} 0.0 & 0.4 & 1.0 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.6 \\ 1.0 \end{bmatrix} \begin{bmatrix} 1.0 & 0.6 & 0.0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0 & 0.4 & 1.0 \\ 0.0 & 0.4 & 0.4 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} + \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.6 & 0.6 & 0.0 \\ 1.0 & 0.6 & 0.0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0 & 0.4 & 1.0 \\ 0.6 & 0.6 & 0.4 \\ 1.0 & 0.6 & 0.0 \end{bmatrix}$$

และถ้าต้องการหาความสัมพันธ์ของพจน์ If  $\chi$  is  $A$ , then  $Y$  is  $B$  โดยใช้ฟัซซีเซต **UNKNOWN** จะได้

$$R = \begin{bmatrix} 1.0 \\ 0.4 \\ 0.0 \end{bmatrix} \begin{bmatrix} 0.0 & 0.4 & 1.0 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.6 \\ 1.0 \end{bmatrix} \begin{bmatrix} 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0 & 0.4 & 1.0 \\ 0.0 & 0.4 & 0.4 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} + \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.6 & 0.6 & 0.6 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0 & 0.4 & 1.0 \\ 0.6 & 0.6 & 0.6 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}$$

สำหรับหน่วยความจำแบบฟัซซีแอสซิซิเอทีฟ (Fuzzy Associative Memories (FAM)) [Kosko87, Ross04] เป็นการปฏิบัติต่อ การหาเหตุผลโดยประมาณเหมือนเป็นระบบฟัซซี (fuzzy system) ที่ส่งทอด (map) ฟัซซีเซตบน  $X$  ไปยังฟัซซีเซตบน  $Y$  นั่นคือทำให้พจน์ If  $\chi$  is  $A$ , then  $Y$  is  $B$  เป็นการเชื่อมความสัมพันธ์ระหว่างฟัซซีเซต  $A$  และฟัซซีเซต  $B$  และเนื่องจากในระบบฟัซซีนี้ สามารถมีกฎได้มากกว่า 1 กฎนั้นคือ

กฎ 1: If  $\chi$  is  $A_1$ , then  $Y$  is  $B_1$

กฎ 2: If  $\chi$  is  $A_2$ , then  $Y$  is  $B_2$

⋮

กฎ  $n$ : If  $\chi$  is  $A_n$ , then  $Y$  is  $B_n$

ความจริง:  $\chi$  is  $A'$

บทสรุป:  $Y$  is  $B'$  (3.51)

ดังนั้นระบบฟัซซีจะทำการเก็บความสัมพันธ์ของกฎเหล่านี้  $(A_1, B_1), (A_2, B_2), \dots, (A_n, B_n)$  ในรูปของ  $R_1, R_2, \dots, R_n$  และถ้ามีอินพุต  $A'$  เข้ามาในระบบแต่  $A'$  ไม่เหมือน  $A_1, A_2, \dots, A_n$  เลย ทุกกฎจะทำงานเหมือนกันแต่ด้วยระดับที่ต่างกัน และเอาต์พุตที่ได้จะมีฟังก์ชันสมาชิกเป็น

$$B' = w_1 B_1' + w_2 B_2' + \dots + w_n B_n' \quad (3.52)$$

โดยที่  $w_i$  สำหรับ  $i = 1, \dots, n$  เป็นน้ำหนักที่เหมาะสม (อาจจะเป็นระดับความน่าเชื่อถือ ระดับความเข้ากันได้ หรือระดับความสำคัญ) และ

$$B_i' = A' \bullet R_i \quad (3.53)$$

การหาความสัมพันธ์ของกฎ  $R_i$  ทำได้ 2 วิธีคือ correlation-min หรือ correlation-product สมมุติให้ฟังก์ชันสมาชิกถูกแทนด้วย เวกเตอร์แถว (row vector) เช่น

$$A_i = a_1/x_1 + a_2/x_2 + \dots + a_m/x_m \quad (3.54ก)$$

จะเขียนได้เป็น

$$A_i = [a_1, a_2, \dots, a_m] \quad (3.54ข)$$

ดังนั้นการหาความสัมพันธ์แบบ correlation-min [Klir95] ระหว่าง  $A_i$  และ  $B_j = [b_1, b_2, \dots, b_p]$  ทำได้โดย

$$R_i = A_i^T \circ B_j \quad (3.55)$$

$$R_i = \begin{bmatrix} a_1 \wedge B_j \\ a_2 \wedge B_j \\ \vdots \\ a_m \wedge B_j \end{bmatrix} \quad (3.56)$$

ซึ่งมีค่าเท่ากับ 
$$R_i = [b_1 \wedge A_i^T \quad b_2 \wedge A_i^T \quad \dots \quad b_p \wedge A_i^T] \quad (3.57)$$

จะเห็นได้ว่าสมการ 3.56 ความสัมพันธ์ที่ได้จะเห็นว่าในแต่ละแถวจะเป็น ฉบับขลิบ (clipped version) ของฟังก์ชันเซต  $B_j$  และจากสมการ 3.56 จะเห็นว่าในแต่ละคอลัมน์จะเป็น ฉบับขลิบ (clipped version) ของฟังก์ชันเซต  $A_i^T$  นั่นเอง และมีทฤษฎีที่เกี่ยวกับการหาความสัมพันธ์แบบนี้ [Kosko87, Kosko88, Ross04] คือ

**ทฤษฎี 3.4** ทฤษฎี correlation-min bidirectional FAM

ถ้า  $R = \text{correlation-min}(A^T, B)$  แล้ว

1.  $A \bullet R = B$  ก็ต่อเมื่อ (iff)  $h(A) \geq h(B)$
2.  $B \bullet R^T = A$  ก็ต่อเมื่อ (iff)  $h(B) \geq h(A)$
3.  $A' \bullet R \subseteq B$  สำหรับฟังก์ชันเซต  $A'$  ใดๆ
4.  $B' \bullet R^T \subseteq A$  สำหรับฟังก์ชันเซต  $B'$  ใดๆ

สำหรับการหาความสัมพันธ์แบบ correlation-product ระหว่าง  $A_i$  และ  $B_j$  ทำได้โดย

$$R_i = A_i^T \circ B_j \quad (3.58)$$

$$R_i = \begin{bmatrix} a_1 B_j \\ a_2 B_j \\ \vdots \\ a_m B_j \end{bmatrix} \quad (3.59)$$

หรือเท่ากับ 
$$R_i = [b_1 A_i^T \quad b_2 A_i^T \quad \dots \quad b_p A_i^T] \quad (3.60)$$

ซึ่งสมการ 3.59 ในแต่ละแถวของความสัมพันธ์ที่ได้เป็น ฉบับสเกล (scaled version) ของ  $B_j$  และจากสมการ 3.60 จะเห็นว่าในแต่ละคอลัมน์จะเป็น ฉบับสเกล (scaled version) ของฟัซซีเซต  $A_j^T$  นั้นเอง และมีทฤษฎีที่เกี่ยวกับการหาความสัมพันธ์แบบนี้คือ

ทฤษฎี 3.5 ทฤษฎี correlation-product bidirectional FAM

ถ้า  $R = \text{correlation-product}(A^T, B)$  แล้ว

1.  $A \bullet R = B$       ก็ต่อเมื่อ (iff)  $h(A) = 1$
2.  $B \bullet R^T = A$       ก็ต่อเมื่อ (iff)  $h(B) = 1$
3.  $A' \bullet R \subseteq B$       สำหรับฟัซซีเซต  $A'$  ใดๆ
4.  $B' \bullet R^T \subseteq A$       สำหรับฟัซซีเซต  $B'$  ใดๆ

ในการหาบทสรุปในกรณีที่มีกฎมากกว่า 1 กฎ ทำได้โดยใช้ สมการที่ 3.52 โดยทั่วไปค่าน้ำหนักของแต่ละกฎจะเป็นค่าความน่าเชื่อถือของกฎนั้นๆ และโดยมากจะมีค่าเป็น 1 หรือจะทำโดยการรวมความสัมพันธ์ของทุกกฎก่อน คือ [Klir95]

$$R = \bigcup_{j \in N_n} R_j \quad (3.61)$$

โดยที่  $N_n = \{1, 2, \dots, n\}$  ซึ่งความหมายของสมการนี้คืออย่างน้อย 1 กฎที่เข้าข่าย (fire) และการทำยูเนียนนี้ก็ใช้การทำฟัซซียูเนียนมาตรฐาน ที่เคยกล่าวแล้ว หรืออาจจะใช้ [Klir95]

$$R = \bigcap_{j \in N_n} R_j \quad (3.62)$$

ซึ่งความหมายของสมการนี้คือทุกกฎต้องเข้าข่าย (fire) และเช่นเดียวกันการทำอินเตอร์เซกชันนี้จะเป็นการทำฟัซซีอินเตอร์เซกชันมาตรฐาน นั้นเอง หลังจากนั้นหาบทสรุป โดยที่สมมุติให้ความจริงหรืออินพุตที่เข้ามาเป็น  $A'$  ได้โดย

$$\begin{aligned} B' &= A' \bullet R \\ &= A' \bullet \bigcup_{j \in N_n} R_j \\ &= A' \bullet \sup_{j \in N_n} R_j \\ &= \sup_{j \in N_n} (A' \bullet R_j) \\ &= \sup_{j \in N_n} (A' \bullet A_j^T \circ B_j) \\ &= \sup_{j \in N_n} ((A' \bullet A_j^T) \circ B_j) \end{aligned}$$

ดังนั้น

$$B = \sup \left[ (A' \bullet A_1^T) \circ B_1, (A' \bullet A_2^T) \circ B_2, \dots, (A' \bullet A_i^T) \circ B_i, \dots, (A' \bullet A_n^T) \circ B_n \right] \quad (3.63)$$

ถ้าค่าความจริงที่เข้ามาเป็นเซตแบบดั้งเดิมหรือรู้ความจริงแน่นอน นั่นคือ

$$A' = 0/x_1 + 0/x_2 + \dots + 1/x_i + 0/x_{i+1} + \dots + 0/x_m$$

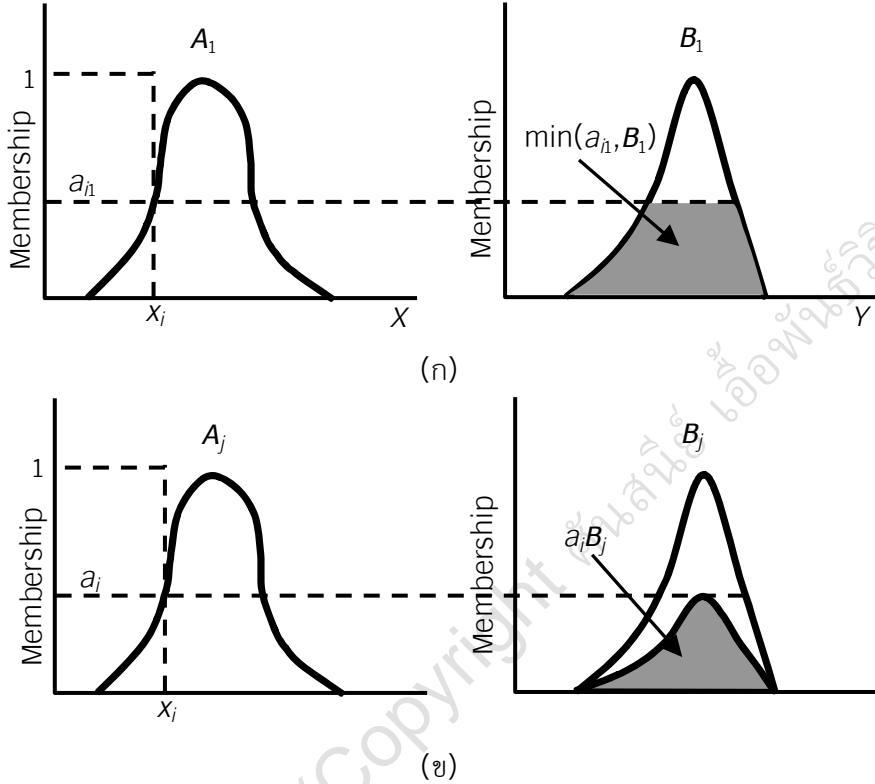
และใช้ correlation-min ดังนั้นสำหรับพจน์ที่  $j$  ในสมการที่ 3.63 จะเป็น

$$(A' \bullet A_j^T) \circ B_j = \min(A_i(x_i), B_j) \quad (3.64)$$

และถ้าใช้ correlation-product จะได้

$$(A' \bullet A_j^T) \circ B_j = A_i(x_i)B_j \quad (3.65)$$

สมการที่ 3.64 และ 3.65 แสดงในรูปที่ 3.24 (ก) และ 3.24 (ข)



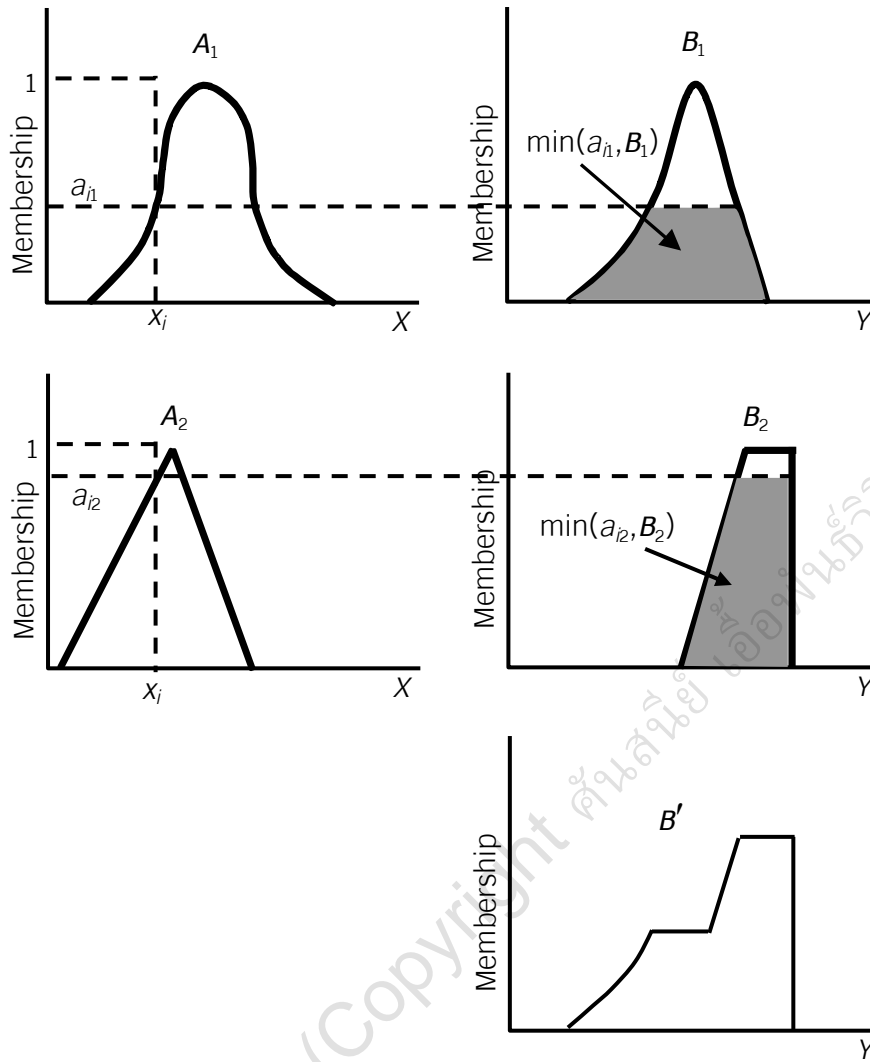
รูปที่ 3.24 (ก) ใช้ correlation-min (ข) ใช้ correlation-product สำหรับกรณีที่  $A'$  เป็นคริสป์

เซต

ตัวอย่างที่ 3.11 สมมุติระบบฟัซซีมีกฎ 2 กฎที่แต่ละกฎใช้ correlation-min และการรวมความสัมพันธ์ใช้ฟัซซียูเนียนมาตรฐาน ดังรูปที่ 3.25 และให้ความจริงที่เข้ามาเป็นคริสป์เซต เป็น  $A' = 0/x_1 + 0/x_2 + \dots + 1/x_i + 0/x_{i+1} + \dots + 0/x_m$  เมื่อผ่านความจริงนี้เข้าไปในระบบ จะได้ว่า  $(A' \bullet A_1^T) = a_{11}$  และ  $(A' \bullet A_2^T) = a_{12}$  และเอาต์พุตของระบบฟัซซีคือฟัซซีเซต

$$B' = \sup [(A' \bullet A_1^T) \circ B_1, (A' \bullet A_2^T) \circ B_2] \quad (3.66)$$

ซึ่งเป็นฟัซซีเซตที่แสดงในรูป 3.25

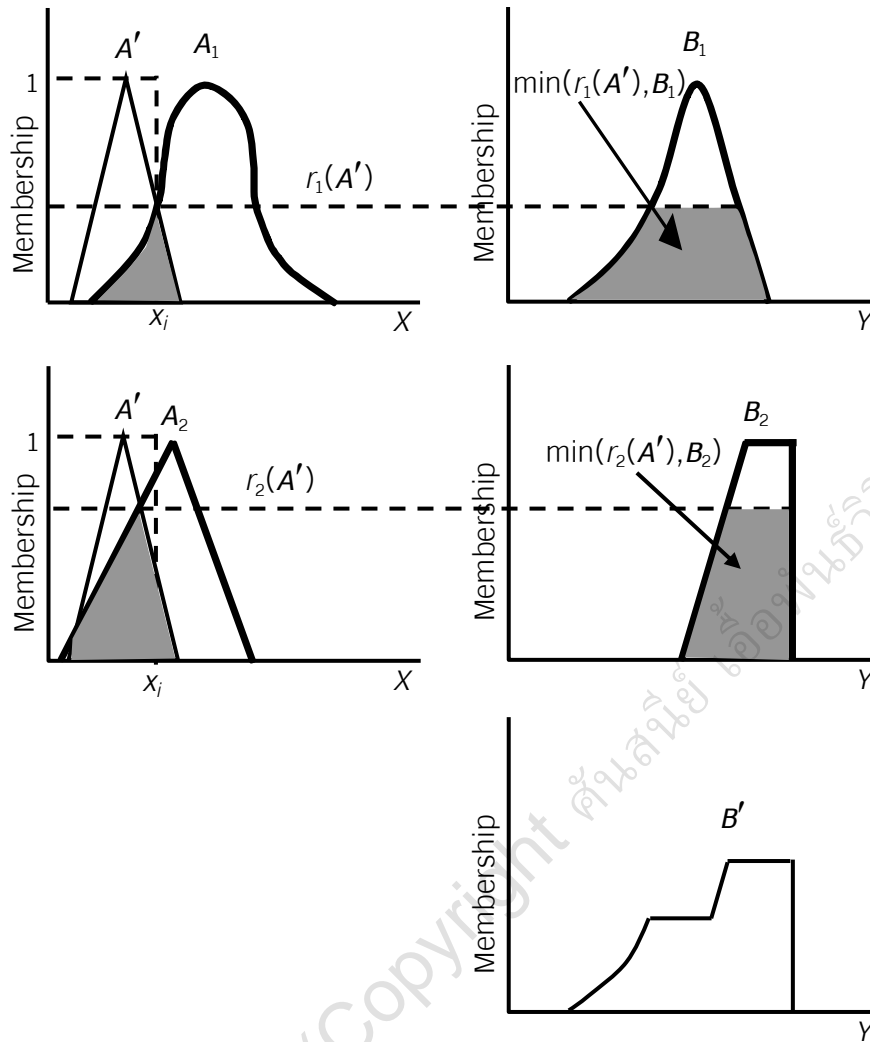


รูปที่ 3.25 ตัวอย่างของระบบฟัซซีที่มี 2 กฎและความจริงอินพุตเป็นสิ่งที่รู้แน่นอน

ตัวอย่างที่ 3.12 ถ้าความจริงที่เข้ามามีความไม่แน่นอนคือเป็นฟัซซีเซต  $A'$  และสมมุติระบบฟัซซีมีกฎ 2 กฎที่แต่ละกฎใช้ correlation-min และการรวมความสัมพันธ์ใช้ฟัซซียูเนียนมาตรฐาน จากสมการที่ 3.63 การหา  $(A' \bullet A_j)^{\circ} B_j$  ทำได้โดยการหา  $(A' \bullet A_j)^{\top}$  ก่อนซึ่ง

$$r_j(A') = (A' \bullet A_j)^{\top} = \sup_{x \in X} \min[A'(x), A_j(x)] \quad (3.67)$$

ซึ่งการหา  $\min[A'(x), A_j(x)]$  ก็คือการหาฟัซซีอินเตอร์เซกชันมาตรฐาน และการหา supremum ของการอินเตอร์เซกชัน คือการหา height ของอินเตอร์เซกชันระหว่าง  $(A', A_j)$  ซึ่งคือระดับของความต้อกัน (degree of consistency) นั่นเองและการหาเอาต์พุตฟัซซีเซต ทำเช่นเดียวกับในตัวอย่างที่ 3.11 นั่นเอง [Klir95] ระบบนี้แสดงในรูปที่ 3.26



รูปที่ 3.26 ตัวอย่างของระบบฟัซซีที่มี 2 กฎและความจริงอินพุตเป็นสิ่งที่ไม่แน่นอน

การทำการหาเหตุผลโดยประมาณไม่ว่าความจริงอินพุตจะเป็นที่รู้แน่นอนหรือไม่แน่นอน เอาต์พุตที่ได้จะเป็นฟัซซีเซต แต่ในการประยุกต์ใช้ต่างๆเช่นในระบบควบคุมแบบฟัซซี (fuzzy control) จำเป็นที่เอาต์พุตที่ออกจากระบบต้องเป็นตัวเลขเพื่อนำไปใช้งาน และจะเป็นตัวบอกว่าระบบควรจะทำงานอย่างไร ในกรณีต่างๆตามสภาวะแวดล้อม และในปัจจุบันมีวิธีการแปลงฟัซซีเซตเป็นตัวเลข (defuzzification) มากมาย และการเลือกใช้วิธีที่ต่างกันอาจทำให้ได้ตัวเลขที่ต่างกัน ได้ และในหัวข้อนี้จะพูดถึงการแปลง 2 วิธีเท่านั้น [Klir95] โดยให้  $B'$  เป็นฟัซซีเซตเอาต์พุตบนเซตสากล  $Y$

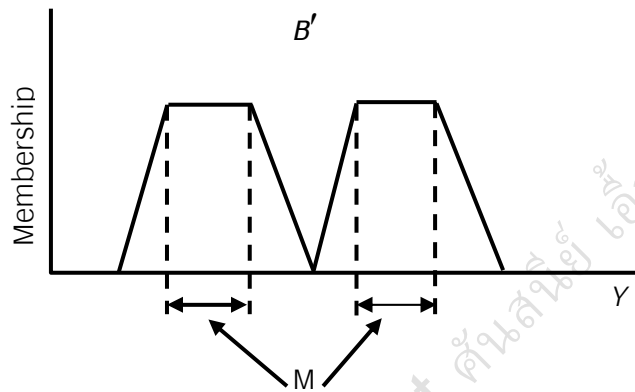
1. *max membership*: ให้เลือก  $de\_y$  ที่  $B'(de\_y) \geq B'(de\_y')$  สำหรับทุก  $de\_y' \in Y$  เป็นตัวเลขที่ใช้แทนฟัซซีเซต  $B'$  แต่ถ้า  $de\_y$  มีมากกว่า 1 ค่า ให้ใช้ค่า mean ของ  $de\_y$  เหล่านั้น ในกรณีที่ฟัซซีเซตที่ได้มีฟังก์ชันสมาชิกเป็นฟังก์ชันดิสครีต (discrete function) นั่นคือ ให้  $M = \{y \in [y_1, y_2] \mid B'(y) = h(B')\}$  โดยที่  $h(B')$  เป็น height ของ  $B'$  และค่า  $de\_y$  คือ

$$de\_y = \frac{\sum_{y_k \in M} y_k}{|M|} \quad (3.68)$$

และแปลงโดยใช้ center ของค่า  $de\_y$  เหล่านั้นถ้าเป็นฟังก์ชันสมาชิกเป็นฟังก์ชันต่อเนื่อง (continuous function) ค่า  $de\_y$  คือ

$$de\_y = \frac{\inf M + \sup M}{2} \quad (3.69)$$

แต่ถ้าฟังก์ชันที่ได้ไม่ใช่คอนเวกซ์ฟังก์ชันดังรูป 3.27 การหาค่าเฉลี่ยหรือ center ของค่า  $de\_y$  ก็อาจจะไม่ถูกต้องนัก



รูปที่ 3.27 ตัวอย่างของฟังก์ชันเอาต์พุตของระบบฟัซซีที่ไม่ใช่คอนเวกซ์ฟังก์ชัน

2. *Center of Area (Centroid)*: ให้เลือก  $de\_y$  ที่เป็นค่า centroid ของฟังก์ชันนั้นคือ ถ้าฟังก์ชัน  $B'$  ที่มีฟังก์ชันสมาชิกเป็นฟังก์ชันดิสครีต (discrete function) ค่า  $de\_y$  คือ

$$de\_y = \frac{\sum_k B'(y_k) y_k}{\sum_k B'(y_k)} \quad (3.70)$$

ถ้าเป็นฟังก์ชันที่มีฟังก์ชันสมาชิกเป็นฟังก์ชันต่อเนื่อง (continuous function) ค่า  $de\_y$  คือ

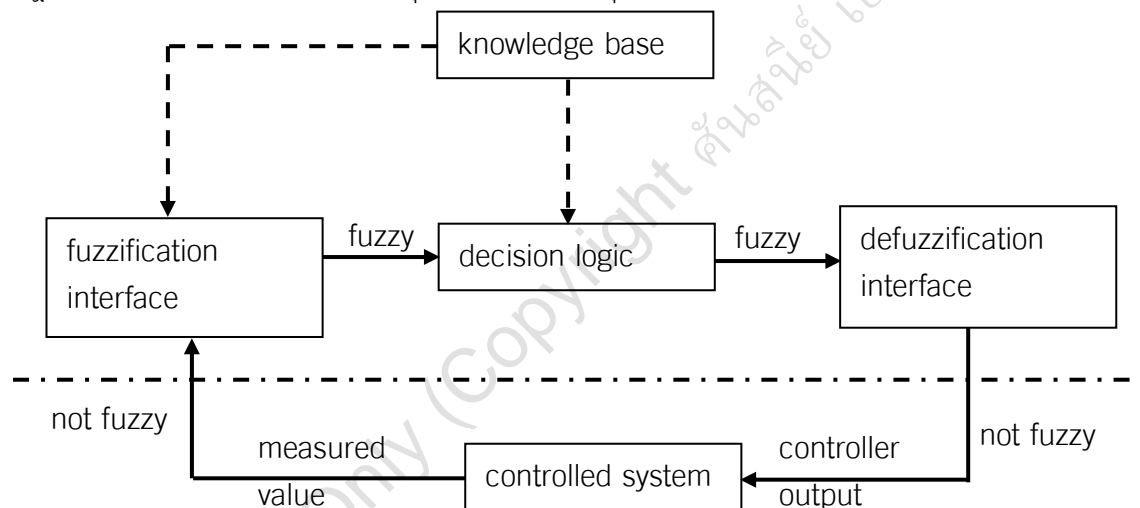
$$de\_y = \frac{\int B'(y) y dy}{\int B'(y) dy} \quad (3.71)$$

และในกรณีนี้ก็เช่นเดียวกันกับการหา max membership นั่นคือถ้าฟังก์ชันที่ได้ไม่ใช่คอนเวกซ์ฟังก์ชันและมีรูปร่างดังรูป 3.27 ค่า  $de\_y$  ที่ได้จะเป็นค่ากึ่งกลางซึ่งมีค่าสมาชิกเป็น 0 ทำให้เป็นคำตอบที่ผิดจากที่ควรจะเป็นมากนั่นเอง และอีกปัญหาหนึ่งก็คือเสียเวลาในการคำนวณมากเพราะใช้ข้อมูลทุกอย่างที่มีจากฟังก์ชัน ในการลด computational expensive สามารถใช้ ฐานนิยม (mode) [Klir95] ของ  $B_j(y_{B_j}^0)$  ในแต่ละกฎมาช่วยได้นั่นคือให้  $N_n = \{1, 2, \dots, n\}$  และ  $c_j$  สำหรับ  $1 \leq j \leq n$  เป็น ระดับของความตึงเครียด นั่นคือ  $a_{ij}$  ในกรณีของความจริงที่รู้แน่นอน หรือ  $r_i(A')$  ในกรณีของความจริงที่ไม่รู้แน่นอนดังนั้นค่า  $de\_y$  คือ



$$de\_y = \frac{\sum_{j \in N_n} c_j y_{B_j}^0}{\sum_{j \in N_n} c_j} \quad (3.72)$$

ที่กล่าวมาข้างต้นนั้นเป็นการหาคำตอบสำหรับระบบที่มีอินพุตเดียว แต่สำหรับการประยุกต์ใช้ระบบฟัซซีที่ใช้กันมากในปัจจุบัน คือระบบควบคุมฟัซซี (fuzzy controller) [Kruse95] โดยส่วนใหญ่เป็นระบบฟัซซีที่มีหลายอินพุต (fuzzy system with multiple inputs) แสดงในรูปที่ 3.28 ซึ่งระบบนี้ประกอบด้วย ส่วนที่เป็น fuzzification interface ซึ่งเป็นส่วนที่รับค่าอินพุต และถ้าเป็นอินพุตที่รู้แน่นอนเช่นอุณหภูมิ 30 องศา และในส่วนนี้จะแปลงอินพุตให้เป็นฟัซซีเซต ส่วนที่ 2 คือ knowledge base เป็นส่วนที่เก็บข้อมูลทุกอย่างรวมทั้งกฎต่างๆ ส่วนที่ 3 คือ decision logic เป็นส่วนในการหาเหตุผลโดยการประมาณตามกฎที่มีใน knowledge base และส่วนสุดท้ายคือ defuzzification interface เป็นส่วนที่จะแปลงฟัซซีเซตให้เป็นตัวเลขเพื่อนำไปใช้ในการควบคุมนั่นเอง และโดยมากระบบนี้จะเป็นระบบที่มีอินพุตมากกว่า 1 พจน์แรก (antecedent) ในกฎต่างๆ หรือเรียกอีกอย่างว่ามีอินพุตมากกว่า 1 อินพุตนั่นเอง



รูปที่ 3.28 ระบบควบคุมฟัซซี

ตัวอย่างที่ 3.13 [Kruse95] ระบบควบคุมฟัซซีที่ใช้ในการควบคุมเสถียรภาพของลูกตุ้มกลับหัว (inverted pendulum) แสดงในรูปที่ 3.29 ซึ่งสิ่งที่ระบบนี้ต้องทำคือพยายามทำให้ลูกตุ้มอยู่ตรงกลางนั่นคือ ทำให้ลูกตุ้มตีกลับไปในทิศทางตรงข้ามโดยการให้กระแสไฟฟ้ากับมอเตอร์ และถ้าการหมุนกลับเกินไปในทิศตรงข้ามก็ต้องให้กระแสไฟฟ้าในทิศตรงข้าม อีกเพื่อให้ลูกตุ้มตีกลับอีกครั้ง

ลูกตุ้มเคลื่อนที่ไปในทิศทางที่ทำมุมกับเส้นกึ่งกลางเป็นมุม  $\theta$  ด้วยความเร็ว  $\Delta\theta$  ซึ่งทั้งมุมและความเร็วน่าจะเป็นอินพุตให้กับระบบควบคุมฟัซซี และระบบฟัซซีจะให้เอาต์พุตคือกระแสไฟฟ้า  $v$  โดยที่ให้อินพุตทั้งสอง และเอาต์พุตมีค่าตัวแปรภาษาเป็น negative large (NL) negative medium (NM) negative small (NS) zero (ZE) positive small (PS) positive medium (PM) และ positive large (PL) โดยที่ negative หมายถึงลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา หรือให้กระแสไฟฟ้าในทิศไปทางซ้าย ในขณะที่เคลื่อนที่ตามเข็มนาฬิกาหรือให้กระแสไฟฟ้าในทิศไปทางขวาถ้าเป็น positive ดังนั้นกฎที่  $j$  จะมีลักษณะเป็น

กฎ  $j$ : If  $\theta$  is  $A_j$ , และ  $\Delta\theta$  is  $B_j$  then  $v$  is  $C_j$

เช่น

If  $\theta$  is NL และ  $\Delta\theta$  is NL, then  $v$  is PL

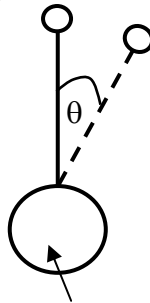
หรือ

If  $\theta$  is ZE และ  $\Delta\theta$  is ZE, then  $v$  is ZE

โดยปกติเราสามารถเขียนกฎเหล่านี้ให้อยู่ในลักษณะทั่วไปได้เช่น  $(\theta, \Delta\theta; v)$  ซึ่งในตัวอย่างของกฎทั้งสองเขียนได้เป็น (NL,NL;PL) สำหรับกฎแรก และ (ZE,ZE;ZE) และเราสามารถเขียนกฎโดยใช้เมตริกซ์ได้เช่น สมมุติให้มี 13 กฎ จะได้เมตริกซ์เป็น

$\theta \backslash \Delta\theta$	NL	NM	NS	ZE	PS	PM	PL
NL				PL			
NM				PM			
NS				PS			
ZE	PL	PM	PS	ZE	NS	NM	NL
PS				NS			
PM				NM			
PL				NL			

โดยที่ค่าที่อยู่ในเมตริกซ์คือค่าเอาต์พุตนั่นเอง เช่นกฎ (NL,ZE;PL) คือกฎที่อยู่คอลัมน์ที่ 1 แถวที่ 4 นั่นเอง จะเห็นได้ว่าในระบบนี้มีกฎที่เป็นไปได้ทั้งหมด 343 กฎคือแต่ละกฎให้ค่าตัวแปรภาษาของอินพุต 2 อินพุตและ 1 เอาต์พุตโดยที่แต่ละอินพุตมี 7 ตัวแปรภาษาและเอาต์พุตมี 7 ตัวแปรภาษาเช่นกัน ซึ่งจะได้  $7 \times 7 \times 7$  แต่ในการใช้งานจริงบางกฎอาจจะไม่ถูกใช้เลยดังนั้นกฎที่ใช้งานจริงจึงมีน้อยกว่า 343 กฎ



รูปที่ 3.29 ลูกตุ้มกลับหัว

ให้ระบบมีกฎ 1 กฎและถูกเขียนเป็น  $(A,B;C)$  โดยที่พจน์ภาษา (linguistic term) ของอินพุตที่ 1 และของอินพุตที่ 2 เป็น  $A = a_1/x_1 + a_2/x_2 + \dots + a_m/x_m$  และ  $B = b_1/y_1 + b_2/y_2 + \dots + b_m/y_m$  ให้การหาบทสรุปสามารถทำได้โดยที่ให้แยกออกเป็น 2 ส่วนคือ  $(A,C)$  และ  $(B,C)$  ดังนั้นเราสามารถความสัมพันธ์ของแต่ละส่วนได้เป็น

$$M_{AC} = A^T \circ C \text{ และ } M_{BC} = B^T \circ C \quad (3.73)$$

สมมุติให้ความจริงอินพุตเป็น  $A' = I_x^i = 0/x_1 + 0/x_2 + \dots + 1/x_i + 0/x_{i+1} + \dots + 0/x_m$  และอินพุตที่ 2 เป็น  $B' = I_y^j = 0/y_1 + 0/y_2 + \dots + 1/y_j + 0/y_{j+1} + \dots + 0/y_m$  ดังนั้นสำหรับแต่ละอินพุตเราสามารถหา  $A' \bullet M_{AC}$  และ  $B' \bullet M_{BC}$  และนำเอาต์พุตทั้งสองรวมเข้าด้วยกันเป็น

$$F(A', B') = C' = [A' \bullet M_{AC}] \cap [B' \bullet M_{BC}] \quad (3.74)$$

สมมุติว่าระบบนี้ใช้ correlation-min ในการหาความสัมพันธ์จะได้ว่า

$$A' \bullet M_{AC} = I_x^i \bullet M_{AC} \quad (3.75)$$

$$= I_x^i \bullet \begin{bmatrix} a_1 \wedge C \\ a_2 \wedge C \\ \vdots \\ a_m \wedge C \end{bmatrix}$$

$$= a_i \wedge C$$

และ

$$B' \bullet M_{BC} = I_y^j \bullet M_{BC} \quad (3.76)$$

$$= I_y^j \bullet \begin{bmatrix} b_1 \wedge C \\ b_2 \wedge C \\ \vdots \\ b_m \wedge C \end{bmatrix}$$

$$= b_j \wedge C$$

ดังนั้นฟัซซีเอาต์พุตจะเป็น

$$C' = (a_i \wedge C) \cap (b_j \wedge C)$$

$$= (\min(a_i, b_j)) \wedge C \quad (3.77)$$

ซึ่งค่า  $a_i$  เป็นระดับที่สอดคล้องกัน (degree of satisfaction) ของ  $x$  ใน พจน์ภาษา  $A$  ทำนองเดียวกันค่า  $b_j$  เป็นระดับที่สอดคล้องกัน ของ  $y$  ใน พจน์ภาษา  $B$  .ในทำนองเดียวกันถ้าในการหาความสัมพันธ์ใช้ correlation-product จะได้

$$C' = (a_i C) \cap (b_j C)$$

$$= (\min(a_i, b_j)) C \quad (3.78)$$

### 3.2.3 วิธีการในระบบควบคุมฟัซซี (Approached to Fuzzy Control)

วิธีการที่ใช้กันมากที่สุดในระบบควบคุมฟัซซี มีอยู่ด้วยกัน 2 วิธี คือ Mamdani model และ Takagi-Sugeno model ทั้งสองวิธีการเน้นไปที่ลักษณะ (specification) ของฟังก์ชันควบคุม (control function) และสิ่งที่เป็นฟัซซีของทั้งสองวิธีคือ ตัวแบบ (model) และวิธีการ (method) เท่านั้นส่วนฟังก์ชันควบคุม (control function) จะเป็นคริสป (crisp) เสมอ

#### 3.2.3.1 วิธีการของ Mamdani (The Mamdani model)

ระบบมีอินพุตมากกว่า 1 ( $X_1, X_2, \dots, X_n$ ) และแต่ละอินพุตถูกนิยามด้วยแปรภาษา (linguistic variable) ( $x_1, x_2, \dots, x_n$ ) และพจน์ภาษา (linguistic term)  $T(x_i)$  ของตัวแปรภาษา  $x_i$  ในเซตสากล  $X_i$  สำหรับ  $1 \leq i \leq n$  ไว้แล้ว ในขณะที่เดียวกันเอาต์พุต  $Y$  ก็ถูกนิยามด้วยแปรภาษา (linguistic

variable)  $y$  และพจน์ภาษา (linguistic term)  $\pi(y)$  ของตัวแปรภาษา  $y$  ในเซตสากล  $Y$  ไว้แล้ว เช่นกัน และกฎมีลักษณะดังนี้ [Kruse95]

If  $\xi_1$  is  $A^{(1)}$ , และ  $\xi_2$  is  $A^{(2)}$  และ ... และ  $\xi_n$  is  $A^{(n)}$  then  $\eta$  is  $B$  (3.79)  
โดยที่  $A^{(1)}$ ,  $A^{(2)}$  และ ... และ  $A^{(n)}$  เป็นพจน์ภาษาใน  $\pi(x_i)$  สำหรับ  $1 \leq i \leq n$  และ  $B$  พจน์ภาษาใน  $\pi(y)$  นั้นเองและถ้ามีกฎมากกว่า 1 กฎและให้

$$\begin{aligned} \pi(x_1) & \text{ ประกอบด้วย } A_1^{(1)}, A_2^{(1)}, \dots, A_{N_1}^{(1)} \\ \pi(x_2) & \text{ ประกอบด้วย } A_1^{(2)}, A_2^{(2)}, \dots, A_{N_2}^{(2)} \\ & \vdots \\ \pi(x_n) & \text{ ประกอบด้วย } A_1^{(n)}, A_2^{(n)}, \dots, A_{N_n}^{(n)} \end{aligned} \quad (3.80)$$

$$\text{และ } \pi(y) \text{ ประกอบด้วย } B_1, B_2, \dots, B_{N_0} \quad (3.81)$$

จะเขียนกฎได้ว่า

กฎ  $j$ : If  $\xi_1$  is  $A_{i_1,j}^{(1)}$ , และ  $\xi_2$  is  $A_{i_2,j}^{(2)}$  และ ... และ  $\xi_n$  is  $A_{i_n,j}^{(n)}$  then  $\eta$  is  $B_{i,j}$  (3.82)  
โดยที่  $i_1 \in \{1,2,\dots,N_1\}$ ,  $i_2 \in \{1,2,\dots,N_2\}$ , ...,  $i_n \in \{1,2,\dots,N_n\}$  และ  $i \in \{1,2,\dots,N_0\}$   
นั้นเอง จากตัวอย่าง 3.13 สามารถบอกได้ว่ามีกฎที่เป็นไปได้ทั้งหมดเท่ากับ  $N_1 \times N_2 \times \dots \times N_n \times N_0$   
แต่ในการใช้งานจริงมีเพียงบางกฎเท่านั้นที่นำไปใช้ได้จริง

ในการหาเหตุผล จะเริ่มจากการหาระดับความเข้ากันได้ของแต่ละอินพุต ( $x_i$  โดยที่  $i \in \{1,2,\dots,n\}$ ) กับพจน์ภาษาในกฎนั้น และเนื่องจากลักษณะของข้อตั้ง (premise) ของกฎ ต้องการให้ทุกอินพุตเป็นไปตามพจน์ภาษา ดังนั้นค่าความเป็นสมาชิกของแต่ละอินพุตในแต่ละพจน์ภาษาจะถูกรวมกันในลักษณะของตัวเชื่อม conjunction นั่นคือ ที่กฎ  $j$  [Kruse95]

$$\alpha_j = \min \{A_{i_1,j}^{(1)}(x_1), A_{i_2,j}^{(2)}(x_2), \dots, A_{i_n,j}^{(n)}(x_n)\} \quad (3.83)$$

และเอาต์พุตของกฎ  $j$  เป็นฟัซซีเซตที่เกิดจากการตัด (cut off) พจน์ภาษา  $B_{i,j}$  ด้วย  $\alpha_j$  หรืออาจจะพูดได้ว่า [Kruse95]

$$OUT_{x_1, x_2, \dots, x_n}^{(j)}(y) = \min [A_{i_1,j}^{(1)}(x_1), A_{i_2,j}^{(2)}(x_2), \dots, A_{i_n,j}^{(n)}(x_n), B_{i,j}(y)] \quad (3.84)$$

และเมื่อได้เอาต์พุตของแต่ละกฎแล้ว ฟัซซีเอาต์พุตจากทุกกฎจะถูกรวมกันโดยการหาฟัซซียูเนียนมาตรฐาน (ซึ่งจะได้ฟัซซีเอาต์พุตรวม (OUT)) สมมติให้มียกกฎทั้งหมด  $k$  กฎ จะได้ OUT เป็น [Kruse95]

$$OUT_{x_1, x_2, \dots, x_n}(y) = \max_{j \in \{1,2,\dots,k\}} \min [A_{i_1,j}^{(1)}(x_1), A_{i_2,j}^{(2)}(x_2), \dots, A_{i_n,j}^{(n)}(x_n), B_{i,j}(y)] \quad (3.85)$$

เอาต์พุตที่ออกไปที่ระบบควบคุม จะเป็นค่าที่ได้จากการแปลงฟัซซีเอาต์พุตรวมเป็นตัวเลข โดย defuzzification interface

**ตัวอย่างที่ 3.14** สมมติให้ระบบมีกฎ 2 กฎ โดยที่แต่ละกฎจะมีอินพุต 2 อินพุต และแต่ละอินพุตในแต่ละกฎมีพจน์ภาษาดังรูปที่ 3.30 โดยที่มีกฎดังนี้คือ

กฎ 1: If  $x_1$  is  $L_1$  และ  $x_2$  is  $H_2$ , then  $y$  is  $L$

กฎ 2: If  $x_1$  is  $M_1$  และ  $x_2$  is  $M_2$ , then  $y$  is  $H$

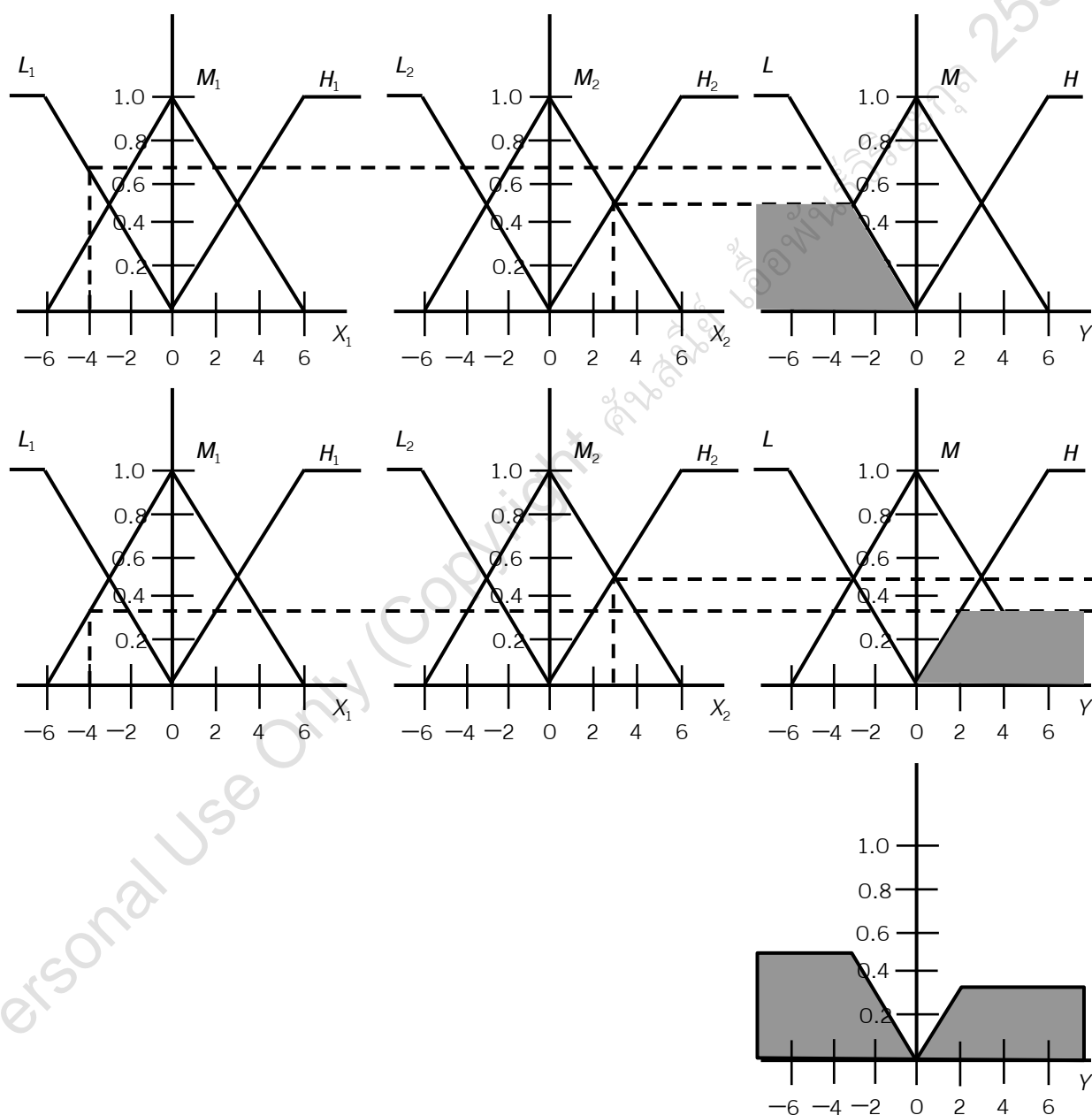
และสมมติให้อินพุตที่เข้ามาที่ fuzzification interface มีค่าเป็น -4 และ 3 จะเห็นได้ว่า

$$\alpha_1 = \min(L_1(-4), H_2(3)) = \min(0.67, 0.5) = 0.5$$

และ

$$\alpha_2 = \min(M_1(-4), M_2(3)) = \min(0.33, 0.5) = 0.33$$

ฟัซซีเอาต์พุตของกฎที่ 1 และ 2 และฟัซซีเอาต์พุตรวม (OUT) แสดงในรูปที่ 3.30 เช่นกัน สมมติว่าในส่วนของการ defuzzification interface ใช้วิธีการแปลงโดยใช้ centroid จะได้ค่า output เท่ากับ 1.2



รูปที่ 3.30 ระบบควบคุมฟัซซีโดยใช้วิธี Mamdani

### 3.2.3.2 วิธีการของ Takagi-Sugeno (The Takagi-Sugeno Model)

ให้นิยามต่างๆ เกี่ยวกับตัวแปรภาษาของอินพุต หรือ ข้อตั้งของกฎเป็นเช่นเดียวกับในหัวข้อ 3.2.3.1 แต่ในวิธีนี้ไม่มีตัวแปรภาษาของเอาต์พุต หรือบทรูป เนื่องจากแต่ละกฎ จะมีเอาต์พุตที่อยู่ในรูปของสมการ ดังนี้ [Kruse95]

$$\begin{aligned} \text{กฎ } j: \quad & \text{If } \xi_1 \text{ is } A_{1,j}^{(1)}, \text{ และ } \xi_2 \text{ is } A_{2,j}^{(2)} \text{ และ ... และ } \xi_n \text{ is } A_{n,j}^{(n)} \\ & \text{then } \eta_j = f_j(\xi_1, \xi_2, \dots, \xi_n) \end{aligned} \quad (3.86)$$

สำหรับ  $1 \leq j \leq k$  โดยที่  $f_j$  เป็นฟังก์ชันที่ส่งทอด (map) จาก  $x_1 \times x_2 \times \dots \times x_n$  ไปยัง  $Y$  ซึ่งโดยปกติแล้วฟังก์ชัน  $f_j$  เป็นฟังก์ชันเชิงเส้น (linear function) คือ สมมติให้  $a_j^i$  เป็นค่าคงที่ ที่ใช้ในกฎ  $j$  และค่าเหล่านี้จะแตกต่างกันในแต่ละกฎที่แตกต่างกัน ดังนั้น

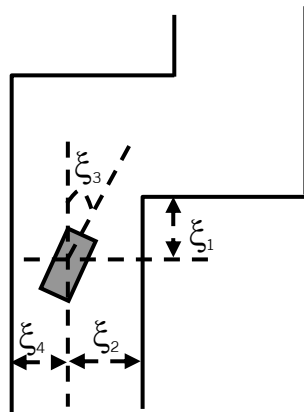
$$f_j(x_1, x_2, \dots, x_n) = a_1^j x_1 + a_2^j x_2 + \dots + a_n^j x_n + a_0^j \quad (3.87)$$

ในการหาเหตุผลของแต่ละกฎ  $j$  คือการหาระดับความเข้ากันได้ของแต่ละอินพุต ( $x_i$  โดยที่  $i \in \{1, 2, \dots, n\}$ ) กับพจน์ภาษาในกฎนั้น และเนื่องจากลักษณะของข้อตั้ง (premise) ของกฎต้องการให้ทุกอินพุตเป็นไปตามพจน์ภาษา เช่นเดียวกับกรณีของ Mamdani ดังนั้นค่าความเป็นสมาชิกของแต่ละอินพุตในแต่ละพจน์ภาษาจะถูกรวมกันในลักษณะของตัวเชื่อม conjunction เป็น  $\alpha_j$  สำหรับกฎที่  $j$  โดยใช้สมการที่ 3.83 และเมื่อคำนวณหา  $f_j$  สำหรับทุกกฎแล้ว นำค่าที่ได้มารวมกันโดย

$$\eta = \frac{\sum_{j=1}^k \alpha_j f_j(x_1, x_2, \dots, x_n)}{\sum_{j=1}^k \alpha_j} \quad (3.88)$$

ดังนั้นค่าเอาต์พุต ( $\eta$ ) ที่ได้จะเป็นตัวเลขไม่ใช่ฟัซซีเซต จึงไม่จำเป็นต้องใช้ defuzzification interface ในกรณีนี้เลย

**ตัวอย่างที่ 3.15** [Kruse95] ต้องการสร้างระบบที่ควบคุมรถให้ผ่านทางโค้งในขณะที่มีความเร็วคงที่ โดยให้อินพุตมีค่า  $\xi_1$  เป็นระยะของรถจนถึงขอบทางโค้ง  $\xi_2$  เป็นระยะของรถจนถึงไหล่ทางข้างขวา  $\xi_3$  เป็นมุมของรถ และ  $\xi_4$  เป็นระยะของรถถึงไหล่ทางข้างซ้าย ดังแสดงในรูปที่ 3.31



รูปที่ 3.31 อินพุตของระบบควบคุมฟัซซีของการควบคุมรถผ่านทางโค้ง

ให้  $\eta$  คือความเร็วของการหมุน (rotation speed) ของพวงมาลัยและให้เซตสากล  $X_1=[0, 150]$  เซนติเมตร  $X_2=[0, 150]$  เซนติเมตร  $X_3=[-90, 90]$  องศา และ  $X_4=[0, 150]$  เซนติเมตร และให้กฎที่  $j$  สำหรับ  $1 \leq j \leq 20$  มีลักษณะดังนี้

กฎ  $j$ : If  $\xi_1$  is  $A_{1,j}^{(1)}$ , และ  $\xi_2$  is  $A_{2,j}^{(2)}$  และ  $\xi_3$  is  $A_{3,j}^{(3)}$  และ  $\xi_4$  is  $A_{4,j}^{(4)}$

then  $\eta = p_0 + p_1\xi_1 + p_2\xi_2 + p_3\xi_3 + p_4\xi_4$

โดยที่  $A_{1,j}^{(1)} \in \{small, medium, big\}$   $A_{2,j}^{(2)} \in \{small, big\}$   $A_{3,j}^{(3)} \in \{outwards, forward, inwards\}$  และ  $A_{4,j}^{(4)} \in \{small\}$  ซึ่งฟuzzyเซตหรือพจน์ภาษาเหล่านี้ถูกกำหนดไว้แล้ว กฎและค่าคงที่ที่ใช้ในการคำนวณแต่ละกฎแสดงในตารางที่ 3.6

ตารางที่ 3.6 กฎและค่าคงที่ที่ใช้ในระบบควบคุมฟuzzy

กฎ	$\xi_1$	$\xi_2$	$\xi_3$	$\xi_4$	$p_0$	$p_1$	$p_2$	$p_3$	$p_4$
1	—	—	outwards	small	3.000	0.000	0.000	-0.045	-0.004
2	—	—	forward	small	3.000	0.000	0.000	-0.030	-0.090
3	small	small	outwards	—	3.000	-0.041	0.004	0.000	0.000
4	small	small	forward	—	0.303	-0.026	0.061	-0.050	0.000
5	small	small	inwards	—	0.000	-0.025	0.070	-0.075	0.000
6	small	big	outwards	—	3.000	-0.066	0.000	-0.034	0.000
7	small	big	forward	—	2.990	-0.017	0.000	-0.021	0.000
8	small	big	inwards	—	1.500	0.025	0.000	-0.050	0.000
9	medium	small	outwards	—	3.000	-0.017	0.005	-0.036	0.000
10	medium	small	forward	—	0.053	-0.038	0.080	-0.034	0.000
11	medium	small	inwards	—	-1.220	-0.016	0.047	-0.018	0.000
12	medium	big	outwards	—	3.000	-0.027	0.000	-0.044	0.000
13	medium	big	forward	—	7.000	-0.049	0.000	-0.041	0.000
14	medium	big	inwards	—	4.000	-0.025	0.000	-0.100	0.000
15	big	small	outwards	—	0.370	0.000	0.000	-0.007	0.000
16	big	small	forward	—	-0.900	0.000	0.034	-0.030	0.000
17	big	small	inwards	—	-1.500	0.000	0.005	-0.100	0.000
18	big	big	outwards	—	1.000	0.000	0.000	-0.013	0.000
19	big	big	forward	—	0.000	0.000	0.000	-0.006	0.000
20	big	big	inwards	—	0.000	0.000	0.000	-0.010	0.000

สมมุติให้รถกำลังเข้าทางโค้งและวัดอินพุตทั้ง 4 ค่า ได้  $\xi_1 = 10$  เซนติเมตร  $\xi_2 = 30$  เซนติเมตร  $\xi_3 = 0$  องศา (นั่นคือไปตรง) และ  $\xi_4 = 50$  เซนติเมตร เมื่อผ่านค่าเหล่านี้เข้าไปในระบบจะมีแค่กฎที่ 4 และ 7 เท่านั้นที่ค่า  $\alpha$  มากกว่า 0 นั่นคือในกฎที่ 4 จากการอ่านค่าความเป็นสมาชิกของ  $\xi_1$  ใน *small* เท่ากับ 0.8  $\xi_2$  ใน *small* เท่ากับ 0.25  $\xi_3$  ใน *forward* เท่ากับ 1 ทำให้ได้  $\alpha_4 = 0.25$  และ

$$\eta_4 = 0.303 - 0.026(10) + 0.061(30) - 0.050(0) + 0.000(50) = 1.873$$

และในกฎที่ 7 จากการอ่านค่าความเป็นสมาชิกของ  $\xi_1$  ใน *small* เท่ากับ 0.8  $\xi_2$  ใน *big* เท่ากับ 0.167  $\xi_3$  ใน *forward* เท่ากับ 0 ทำให้ได้  $\alpha_7 = 0.167$  และ

$$\eta_7 = 2.990 - 0.017(10) + 0.000(30) - 0.021(0) + 0.000(50) = 2.820$$

และค่าเอาต์พุตรวมเท่ากับ

$$\eta = \frac{0.25(1.873) + 0.167(2.820)}{0.25 + 0.167} = 2.252$$

ดังนั้นความเร็วในการหมุนเท่ากับ 2.252 นั้เอง

ในการทำระบบควบคุมฟัซซีไม่ว่าจะใช้วิธี Mamdani หรือ Takagi-Sugeno ก็ยังมีคำถามที่ยังคงไม่มีคำตอบนั้นคือ

1. จะแบ่งเซตสากลของทั้งอินพุตและเอาต์พุตอย่างไร
2. ฟังก์ชันสมาชิกของพจน์ภาษาที่ใช้ควรจะเป็นอย่างไร
3. กฎควรจะมีเท่าไร ระบบถึงจะทำงานได้ดี

### คำถามท้ายบทที่ 3

3.1 จงตัวอย่างคำพูดที่มีความไม่แน่นอนในเนื้อความมาอย่างน้อย 3 ตัวอย่าง

3.2 ให้ทำการคำนวณตามตัวอย่างที่ 3.6 สำหรับ

$$A = \frac{0.1}{1} + \frac{0.2}{2} + \frac{0.2}{3} + \frac{0.3}{4} + \frac{0.5}{5} + \frac{0.8}{6} + \frac{0.8}{7} + \frac{0.9}{8} + \frac{1.0}{9}$$

3.3 ให้ทำการคำนวณตามตัวอย่างที่ 3.8 สำหรับ

กฎ: If a book is large, then it is expensive.

ความจริง: Book x is fairly large

บทสรุป: Book x is fairly expensive (3.31)

โดยที่  $A = \frac{0.1}{x_1} + \frac{0.2}{x_2} + \frac{0.2}{x_3} + \frac{0.3}{x_4} + \frac{0.5}{x_5} + \frac{0.8}{x_6} + \frac{0.8}{x_7} + \frac{0.9}{x_8} + \frac{1.0}{x_9}$  แทนฟัซซีเซต large

ในขณะที่  $B = \frac{0.5}{500} + \frac{0.8}{1000} + \frac{0.8}{1500} + \frac{0.9}{1800} + \frac{1.0}{2000}$  แทนฟัซซีเซต expensive และ  $A'$  แทน

ฟัซซีเซต fairly large ที่ค่าความเป็นสมาชิกถูกคำนวณจาก การดำเนินการเอกภาพ สำหรับคำว่า fairly

3.4 ให้เขียนโปรแกรมสำหรับการทำ Mamdani inference system ในการควบคุมระบบ มา 1 ระบบ

3.5 ให้เขียนโปรแกรมสำหรับการทำ Takagi-Sugeno inference system ในการควบคุมระบบ มา 1 ระบบ

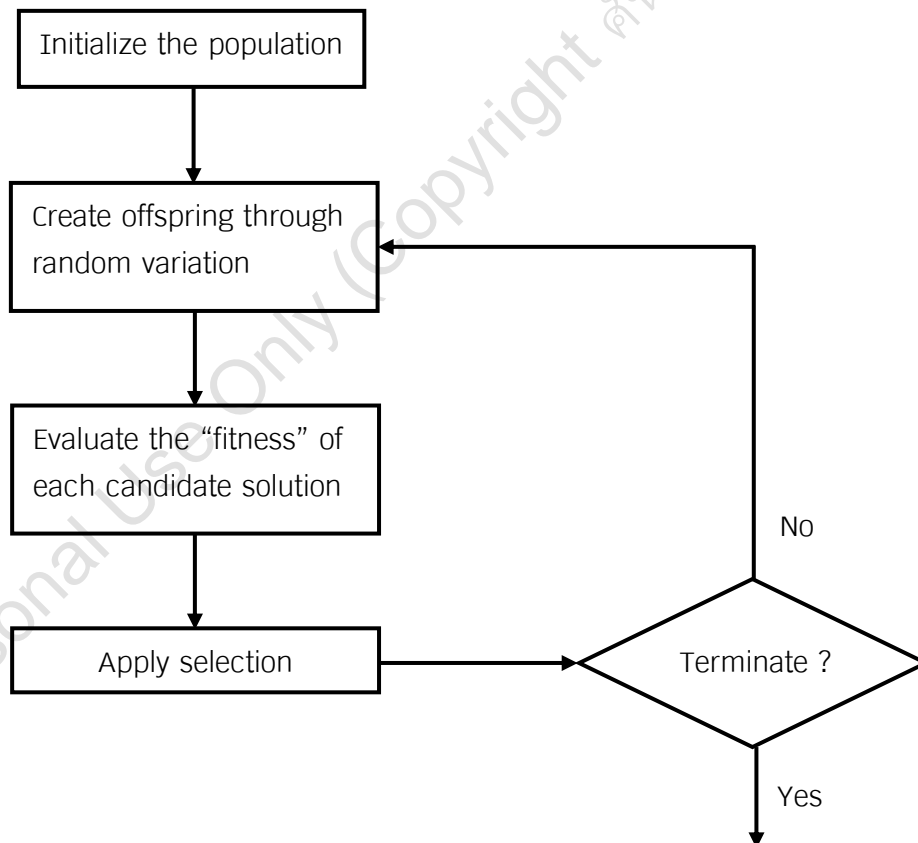


# บทนำการคำนวณเชิงวิวัฒนาการ

## Introduction to Evolutionary Computing

### บทที่ 4

การวิวัฒนาการ (evolution) เป็นกระบวนการหาค่าที่เหมาะสม (optimization) ที่มุ่งหวังในการพัฒนาความสามารถของออร์แกนิซึม (organism) หรือระบบ (system) ให้อยู่รอดได้ในสิ่งแวดล้อมที่มีการเปลี่ยนแปลงแบบไดนามิก (dynamic) หรือสิ่งแวดล้อมที่มีการแข่งขัน ส่วนการคำนวณเชิงวิวัฒนาการ (evolutionary computing) คือระบบการแก้ปัญหาโดยใช้คอมพิวเตอร์ ที่อาศัยแบบจำลองเชิงคำนวณ (computational model) ของกระบวนการวิวัฒนาการ เช่นการคัดเลือกทางธรรมชาติ ความอยู่รอดของตัวที่เหมาะสมที่สุด (fittest) และกระบวนการสืบพันธุ์ (reproduction) [Engelbrecht07] หรืออีกนัยหนึ่งการคำนวณเชิงวิวัฒนาการ คือการเลียนแบบ (emulation) ของกระบวนการคัดเลือกทางธรรมชาติในกระบวนการค้นหา [Engelbrecht02] ส่วนอัลกอริทึมเชิงวิวัฒนาการ (evolutionary algorithm) คือกระบวนการคัดเลือกทางธรรมชาติของการเลือกประชากร (population) แบบสุ่ม (random) ของรายบุคคล (individual) ซึ่งคือการค้นหาในปริภูมิ (space) ของค่าโครโมโซม (chromosome) ที่เป็นไปได้ หรือคือการค้นหาคำตอบที่มีค่าที่เหมาะสมที่สุดแบบสุ่ม (stochastic) สำหรับปัญหานั้นๆ



รูปที่ 4.1 ผังงานอย่างง่ายของการคำนวณเชิงวิวัฒนาการ

ซึ่งองค์ประกอบที่สำคัญของอัลกอริทึมเชิงวิวัฒนาการ มีดังนี้

1. การเข้ารหัส (encoding) ของคำตอบของปัญหาในรูปของโครโมโซม

2. ฟังก์ชันที่ประเมินค่าความเหมาะสม (fitness value) หรือค่าความแรงของความอยู่รอด (survival strength) ของรายบุคคล
3. ประชากรเริ่มต้น (initial population)
4. ตัวดำเนินการคัดเลือก (selection operator)
5. ตัวดำเนินการสืบพันธุ์ (reproduction operator)

โดยปกติแล้วการคำนวณเชิงวิวัฒนาการจะมีการดำเนินการตามแบบผังงานอย่างง่าย (simple flow chart) [Fogel00a] ในรูปที่ 4.1 เสมอ แต่ในบางอัลกอริทึมอาจจะมีการปรับเปลี่ยนได้ตามความเหมาะสม ทั้งนี้ตามผังงานจะเห็นว่าการดำเนินการ (process) จะเริ่มจากการเริ่มต้นประชากรของคำตอบสำหรับปัญหานั้นๆ ซึ่งอาจจะมาจากการสุ่มหรือจากอัลกอริทึมอื่น หลังจากนั้นคำตอบใหม่ (หรือบางครั้งเรียกว่าทายาท (offspring)) จะถูกสร้างจากการเปลี่ยนแปลงคำตอบเดิมจากพ่อแม่ 1 หรือ มากกว่า 1 ตัว โดยเป็นการกระทำแบบสุ่ม ซึ่งคำตอบใหม่เหล่านี้ประกอบด้วยข้อมูลของพ่อแม่ โดยในบางครั้งโครโมโซมเหล่านี้อาจจะถูกกลายพันธุ์ (mutation) ซึ่งเป็นการเปลี่ยนบางลักษณะ (characteristic) ที่อาจจะเป็นผลลบหรือผลบวกต่อค่าความเหมาะสม และถ้าไม่มีการกลายพันธุ์เลยอาจจะทำให้ประชากรลู่เข้าหาความเป็นเอกพันธ์ (homogeneous) ก็เป็นไปได้ ทั้งนี้คำตอบใหม่ที่ได้จะถูกนำไปคำนวณหาค่าความเหมาะสม ซึ่งคำตอบที่ดีจะถูกคัดเลือกจากตัวดำเนินการคัดเลือก และอัลกอริทึมจะสร้างทายาทรุ่นใหม่ต่อไปจนกว่าอัลกอริทึมจะหยุดการทำงานเมื่อถึงเกณฑ์ที่กำหนด ซึ่งในการทำงานแต่ละรอบเรียกว่า 1 รุ่น (generation) นั่นเอง ในส่วนของการหยุดอัลกอริทึมนั้น [Engelbrecht07] อาจจะเป็นการหยุดเนื่องจากเงื่อนไขดังต่อไปนี้

1. จำนวนรุ่นถึงจำนวนรุ่นที่มากที่สุดที่ตั้งไว้ตั้งแต่ต้น
2. ไม่มีการเปลี่ยนแปลงหรือดีขึ้นในจำนวนรุ่นที่ติดกันมากกว่า 1 รุ่น ซึ่งถ้าค่าความเหมาะสมของรายบุคคล (individual) ที่ดีที่สุดในรุ่นไม่ดีขึ้นในช่วงจำนวนรุ่นหนึ่ง ก็สามารถหยุดอัลกอริทึมได้
3. ไม่มีการเปลี่ยนแปลงในประชากร ซึ่งคือในจำนวนรุ่นที่ติดกันจำนวนหนึ่ง ค่าเฉลี่ยการเปลี่ยนแปลงของข้อมูลสีโนไทป์มีค่าน้อย ก็สามารถหยุดอัลกอริทึมได้
4. เมื่อเจอคำตอบที่ยอมรับได้ เช่น ให้  $x$  เป็นคำตอบที่ดีที่สุดที่ต้องการหา และค่าความผิดพลาด (error ( $\epsilon$ )) ระหว่างคำตอบที่หาได้กับคำตอบที่ดีที่สุดมีค่าน้อย ก็สามารถหยุดอัลกอริทึมได้

การคำนวณเชิงวิวัฒนาการถูกนำไปประยุกต์ในหลายการใช้งาน รวมทั้ง

- การวางแผน (planning) เช่นการหาเส้นทางที่เหมาะสมที่สุด (routing optimization) หรือ การจัดกำหนดการ (scheduling)
- การออกแบบ (design) เช่นการออกแบบตัวกรอง (filter) การออกแบบโครงสร้างเครือข่ายประสาทเทียมที่เหมาะสมที่สุด
- การควบคุม (control) เช่นระบบควบคุมของเครื่องยนต์กังหัน (turbine engine) และระบบนำทางด้วยการมองเห็นของหุ่นยนต์
- การแยกแยะ (classification) และการจัดกลุ่ม (clustering)
- การประมาณฟังก์ชัน (function approximation) และการจำลองอนุกรมเวลา (time series modeling)

- ทั้งนี้การคำนวณเชิงวิวัฒนาการ ยังถูกนำไปประยุกต์ใช้ในงานอื่นๆ มากขึ้น อีกด้วย

ในหัวข้อนี้จะกล่าวถึง อัลกอริทึมแบบพันธุกรรม (Genetic algorithm) ในแบบที่มีใช้กันในปัจจุบัน ซึ่งอัลกอริทึมอย่างง่ายมีลักษณะดังนี้ [Dumitrescu00]

- จากอัลกอริทึมที่นั่นในบางครั้งขั้นตอนที่ 7 นั้นถ้าจำนวนโครโมโซมใน  $P^3$  ( $|P^3|$ ) เท่ากับจำนวนโครโมโซม ( $N$ ) ในแต่ละรุ่น โครโมโซมใน  $P^3$  จะเป็นโครโมโซมใน  $P(t+1)$  หรือประชากรของรุ่น (generation) ถัดไปได้ทันที แต่ถ้า  $|P^3|$  น้อยกว่า  $N$  จำนวนโครโมโซมที่หายไป ( $q$ ) นั้นจะถูกเลือกจาก  $P(t)$  หรือเลือกจากโครโมโซมที่ไม่อยู่ใน  $P^1$  ทั้งนี้ในการเลือกจะเป็นไปตามตัวดำเนินการเลือก (selection operator) นั้นเอง ซึ่งโดยปกติคำตอบของปัญหาคือรายบุคคลที่ดีที่สุดในการสุดท้าย แต่ในบางครั้งก็อาจจะไม่ใช่คำตอบที่ดีที่สุดดังนั้นบางครั้งรายบุคคลที่ดีที่สุดจากทุกรุ่นจะถูกเลือกเป็นคำตอบแทน ตอนนี้จะขอลำเอียงถึงรายละเอียดของแต่ละขั้นตอนดังนี้

ในธรรมชาตินั้น สิ่งมีชีวิต (organism) จะมีลักษณะ (characteristics) ที่มีอิทธิพลต่อความสามารถในการเอาอยู่รอด (survive) และการสืบพันธุ์ (reproduction) ซึ่งลักษณะเหล่านี้จะอยู่ในรูปของสายอักขระ (string) ของข้อมูล ที่อยู่ในโครโมโซม (chromosome) ของสิ่งมีชีวิตเหล่านั้น ทั้งนี้โครโมโซมแต่ละอันจะมียีน (gene) ที่เป็นส่วนหนึ่งของพันธุกรรม เป็นส่วนประกอบจำนวนมาก โดยที่ยีนจะเป็นตัวกำหนดกายวิภาค (anatomy) และสรีรวิทยา (physiology) ผ่าน

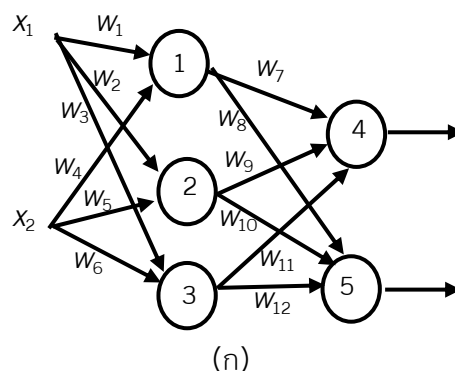
ทางการควบคุมการผลิตโปรตีน ทั้งนี้ยีนแต่ละอันจะมีรูปแบบ (อัลลีล (allele)) ที่ต่างกันได้ แต่รายบุคคล (individual) จะมียีนเฉพาะของตนเองที่ไม่เหมือนยีนของบุคคลอื่น [Engelbrecht07]

สำหรับการคำนวณเชิงวิวัฒนาการนั้น ประชากรของรายบุคคล (population of individual) ถูกใช้ในการหาคำตอบ โดยที่รายบุคคล (individual) เป็นคำตอบให้เลือก (candidate solution) ของปัญหา ซึ่งลักษณะของรายบุคคลอยู่ในรูปของโครโมโซม (chromosome) หรือ จีโนม (genome) [Engelbrecht02] ทั้งนี้ลักษณะเหล่านี้จะเกี่ยวข้องกับตัวแปรของการหาค่าที่เหมาะสม (optimization) ของปัญหา โดยที่แต่ละตัวแปรนั้นถูกเรียกว่ายีน และโดเมนของค่าของตัวแปรนั้นคืออัลลีลนั่นเอง ดังนั้นอาจจะบอกได้ว่า 1 โครโมโซมคือ 1 คำตอบในปริภูมิการหา (search space) ก็เป็นไปได้ โดยปกติแล้วลักษณะนี้จะถูกแบ่งออกเป็น [Engelbrecht07]

- จีโนไทป์ (genotype) ที่อธิบายองค์ประกอบของยีนของรายบุคคล ที่สืบทอดจากพ่อแม่ (parent)
- ฟิโนไทป์ (phenotype) ที่บ่งบอกถึงลักษณะสืบสายพันธุ์ทางพฤติกรรม (behavior trait) ของรายบุคคล ในสถานะแวดล้อมเฉพาะ ซึ่งเป็นตัวกำหนดรูปร่างหน้าตาของรายบุคคล ทั้งนี้จีโนไทป์ และฟิโนไทป์ มีความสัมพันธ์กันในลักษณะดังนี้ [Engelbrecht07]
- โพลีโอโทรปี (pleiotropy) คือปรากฏการณ์ที่ยีน 1 อันมีผลต่อฟิโนไทป์ หรือการที่ยีนมีการเปลี่ยนแปลงแบบสุ่ม ทำให้เกิดการเปลี่ยนในลักษณะสืบสายพันธุ์ทางฟิโนไทป์
- โพลีจีนิ (polygeny) คือการที่ยีนมากกว่า 1 อันที่ปฏิกริยาต่อกันทำให้เกิดลักษณะสืบสายพันธุ์ทางฟิโนไทป์เฉพาะ

โดยปกติแล้วยีนจะอยู่ในรูปของเลขไบนารี และ 1 โครโมโซมจะมีจำนวนยีนที่เท่าเดิม ดังนั้น 1 โครโมโซม เปรียบเสมือนเวกเตอร์ของยีนที่มีมิติคงเดิม และถ้าคำตอบของตัวแปรเป็นจำนวนจริง จึงจำเป็นต้องมีวิธีแปลงหรือเข้ารหัสให้จำนวนจริงเป็นเลขไบนารี ด้วยนั่นเอง ทั้งนี้ในการใช้งานในปัจจุบัน มีหลายอัลกอริทึมที่ให้ยีนมีค่าเป็นจำนวนจริงได้เช่นกัน นอกเหนือจากนี้ยังมีค่ายีนที่เป็นเลขจำนวนเต็มอีกด้วย

**ตัวอย่างที่ 4.1** ต้องการหาค่าน้ำหนักที่เหมาะสมที่สุดของโครงข่าย 2-3-2 ดังรูปที่ 4.2(ก) โดยที่แต่ละจุดในโครงข่ายไม่มีไบแอส ทำให้ต้องหาค่าน้ำหนัก 12 ค่าซึ่งค่าน้ำหนัก 1 ค่าคือ 1 ยีนในโครโมโซม ดังรูปที่ 4.2(ข) ในกรณีที่ยีนมีค่าเป็นจำนวนจริง ทำให้มี 12 ยีน แต่ถ้ายีนมีค่าเป็นไบนารี จึงต้องทำการแปลงเลขจำนวนจริงให้อยู่ในรูปของไบนารี เช่น เลขจำนวนจริง 1 เลข แปลงเป็นไบนารี 3 บิต (bit) ดังนั้นจำนวนยีนคือ  $3 \times 12 = 36$  นั่นเอง ดังนั้นค่าน้ำหนักของ  $w_1$  อยู่ที่บิตที่ 1 ถึง 3 ส่วนค่าน้ำหนักของ  $w_2$  อยู่ที่บิตที่ 4 ถึง 6 ดังรูปที่ 4.2(ค) นั่นเอง



$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------

(ข)

$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------

(ค)

รูปที่ 4.2 (ก) โครงข่ายประสาทเทียม และ ลักษณะของโครโมโซมของ (ก) สำหรับ (ข) ยีนที่เป็นจำนวนจริง และ (ค)ยีนที่เป็นค่าไบนารี

#### 4.1.2 ฟังก์ชันความเหมาะสม (Fitness Function)

ในการกำหนดความสามารถในการอยู่รอดของรายบุคคล นั้นสามารถใช้สมการทางคณิตศาสตร์ ในการหาคำตอบที่ดีที่สุดจากโครโมโซมได้ ซึ่งฟังก์ชันความเหมาะสม ( $F_{EA}$ ) จะเป็นฟังก์ชันที่แมปแต่ละโครโมโซมไปยังค่าจำนวนจริงได้ ดังนี้

$$F_{EA} : C^I \rightarrow \mathbb{R} \quad (4.1)$$

โดยที่  $C$  เป็นเวกเตอร์โครโมโซมที่มี  $I$  มิติ (dimension)

ตัวอย่างที่ 4.2 ตัวอย่าง [Mitchell98] ของการนำอัลกอริทึมแบบพันธุกรรมไปใช้คือ ต้องการหาค่า  $y$  ที่ทำให้

$$\max_{y \in [0, \pi]} f(y) = \max_{y \in [0, \pi]} (y + |\sin(32y)|) \quad (4.2)$$

ดังนั้นในกรณีนี้ค่า  $y$  ที่เป็นจำนวนจริงที่อยู่ในปริภูมิที่กำหนดจะถูกแปลงเป็นเลขไบนารี โดยที่อาจจะกำหนดไว้ว่า เลขจำนวนจริง 1 เลข เทียบเท่ากับเลขไบนารี 5 บิต ดังนั้น 1 โครโมโซมจะมี 5 ยีนเป็นต้น และสามารถใช้สมการที่ 4.2 เป็นฟังก์ชันความเหมาะสมได้อีกด้วย ดังนี้

$$F_{EA} = f(y) \quad (4.3)$$

ตัวอย่างที่ 4.3 สมมติให้ประชากรในรุ่นที่  $t$  ที่ประกอบไปด้วยรายบุคคล  $A, B, C$  และ  $D$  ที่แต่ละโครโมโซมมี 8 ยีน เป็นตารางที่ 4.1 และสมมติให้ฟังก์ชันความเหมาะสมเป็นจำนวนเลข 1 ในโครโมโซม ดังนั้นโครโมโซม  $A, B, C$  และ  $D$  มีค่าความเหมาะสมเป็น 2, 6, 1 และ 3 ตามลำดับ

และค่าความเหมาะสมที่มากที่สุดคือ 6 ส่วนค่าเฉลี่ยของค่าความเหมาะสมคือ  $\frac{12}{4}$  ดังนั้นโครโมโซม

ที่ถูกเลือกสำหรับการสร้างทายาทโดยการข้ามสายพันธุ์ (crossover) คือ  $B$  และ  $D$  เนื่องจากมีค่าความเหมาะสมมากเป็นอันดับ 1 และ 2 สมมติให้การข้ามสายพันธุ์เกิดในตำแหน่งหลังบิตที่ 1 (การข้ามสายพันธุ์จะถูกอธิบายอีกครั้งภายหลัง) ทำให้ได้โครโมโซมใหม่คือ  $E$  และ  $F$  ที่มีค่าเป็น

10110100 และ 01101110 และทายาทที่เหลือถูกเลือกมาจาก  $B$  และ  $C$  ตามลำดับดังนั้นจะได้ตามตารางที่ 4.2 หลังจากนั้นสมมติให้โครโมโซม  $B$  ถูกเลือกให้กลายเป็นพันธุ์ที่ปิดที่ 1 และ  $E$  ถูกเลือกให้กลายเป็นพันธุ์ที่ปิดที่ 6 ซึ่งเป็นดังตารางที่ 4.3 ซึ่งโครโมโซม  $B'$ ,  $C$ ,  $E'$  และ  $F$  คือโครโมโซมที่อยู่ในรุ่นที่  $t+1$  นั้นเอง ทั้งนี้ค่าความเหมาะสมที่มากที่สุดคือ 5 ในขณะที่ค่าเฉลี่ยของค่าความเหมาะสมคือ

14

4

ตารางที่ 4.1 ตัวอย่างของประชากรและค่าความเหมาะสม

ป้ายโครโมโซม (chromosome label)	สายอักขระของโครโมโซม (chromosome string)	ค่าความเหมาะสม (fitness value)
$A$	00000110	2
$B$	11101110	6
$C$	00100000	1
$D$	00110100	3

ตารางที่ 4.2 ทายาทของโครโมโซมในตารางที่ 4.1

ป้ายโครโมโซม (chromosome label)	สายอักขระของโครโมโซม (chromosome string)	ค่าความเหมาะสม (fitness value)
$B$	11101110	6
$C$	00100000	1
$E$	10110100	4
$F$	01101110	5

ตารางที่ 4.3 ประชากรในรุ่นที่  $t+1$

ป้ายโครโมโซม (chromosome label)	สายอักขระของโครโมโซม (chromosome string)	ค่าความเหมาะสม (fitness value)
$B'$	01101110	5
$C$	00100000	1
$E'$	10110000	3
$F$	01101110	5

#### 4.1.3 ประชากรเริ่มต้น (Initial Population)

ในขั้นตอนแรกของอัลกอริทึมแบบพันธุกรรมนั้นคือการเริ่มต้นประชากร ด้วยจำนวนรายบุคคลเท่ากับ  $N$  โดยที่รายบุคคลในประชากรจะเป็นคำตอบให้เลือก ซึ่งโครโมโซมหรือรายบุคคลแต่ละตัว จะมีจำนวนยีน ( $L$ ) ที่เท่ากัน โดยที่แต่ละยีนจะถูกสุ่ม (random) จากโดเมน (domain) ที่กำหนด ซึ่งการสุ่มนี้ต้องการให้ประชากรเริ่มต้นนั้นเป็นตัวแทนของปริภูมิการค้นหา (search space) แบบเอกรูป (uniform) ถ้ามีพื้นที่ในปริภูมิการค้นหาไม่ถูกครอบคลุมด้วยประชากรเริ่มต้น จะมีโอกาสที่พื้นที่เหล่านั้นจะไม่ถูกพิจารณาเลยก็เป็นได้

ทั้งนี้ขนาดของประชากร ( $N$ ) มีผลต่อความซับซ้อนของการคำนวณ และความสามารถในการค้นหา ดังนี้ [Engelbrecht07] จำนวนรายบุคคลมากเป็นการเพิ่มความหลากหลาย ทำให้ความสามารถในการค้นหาดีขึ้น และจำนวนรุ่นในการคำนวณหาคำตอบที่ดีที่สุดจะน้อยลง แต่ความซับซ้อนของการคำนวณต่อรุ่นจะเพิ่มขึ้น แต่ถ้าจำนวนรายบุคคลน้อย จะครอบคลุมพื้นที่ในการค้นหาเล็กๆ ทำให้ความซับซ้อนของการคำนวณต่อรุ่นจะลดลง ในขณะที่ต้องการจำนวนรุ่นในการค้นหาคำตอบที่ดีที่สุดมีมากกว่า ทั้งนี้ในกรณีนี้จำเป็นที่อัลกอริทึมต้องการกลายพันธุ์ด้วยอัตราการกลายพันธุ์ (rate of mutation) ที่มากกว่า เพื่อให้ครอบคลุมพื้นที่การค้นหาให้ได้มากที่สุด

#### 4.1.4 ตัวดำเนินการคัดเลือก (Selection Operator)

การคัดเลือกเป็นขั้นตอนหนึ่งในอัลกอริทึมแบบพันธุกรรม ซึ่งในการคัดเลือกเป็นการคัดเลือกเพื่อ [Engelbrecht07]

1. การคัดเลือกของประชากรใหม่ หรือเป็นการคัดเลือกเพื่อเลือกที่จะเก็บ หรือแทน (replace) รายบุคคลในประชากรนั่นเอง ซึ่งในกรณีนี้เป็นการคัดเลือกกว่าในประชากรใหม่จะประกอบไปด้วยทายาท (offspring) หรือพ่อแม่ของทายาทนั้นๆ ทั้งนี้การคัดเลือกนี้เป็นการยืนยันว่ารายบุคคลที่ดีจะอยู่รอดไปอยู่ในรุ่นถัดไป
2. การคัดเลือกเพื่อกระบวนการสืบพันธุ์ (reproduction) หรือเป็นการคัดเลือกเพื่อให้รายบุคคลมีการเปลี่ยนแปลงทางยีนนั่นเอง ซึ่งในกรณีนี้เป็นการคัดเลือกเพื่อสร้างทายาทจากการข้ามสายพันธุ์ หรือการกลายพันธุ์ ซึ่งการข้ามสายพันธุ์นี้ รายบุคคลที่ดีกว่าจะมีโอกาสในการสืบพันธุ์มากกว่าเพื่อยืนยันว่าทายาทที่ได้จะประกอบไปด้วยยีนที่ดี ส่วนในการกลายพันธุ์นั้นการคัดเลือกจะมุ่งไปที่รายบุคคลที่อ่อนแอกว่า เพื่อที่ว่ารายบุคคลที่อ่อนแอกว่าจะกลายพันธุ์แล้วทำให้เกิดลักษณะสืบสายพันธุ์ (trait) ที่ดีเพื่อโอกาสในการอยู่รอดที่ดีขึ้น

จากอัลกอริทึมที่กล่าวถึงก่อนหน้านั้น ตัวดำเนินการคัดเลือกจะเลือกรายบุคคลเพื่อเป็นพ่อแม่ให้กับรุ่นถัดไป ซึ่งตัวดำเนินการการรวมกันใหม่ (recombination) การกลายพันธุ์ หรือการผกผัน (inversion) จะถูกดำเนินการกับรายบุคคลที่ถูกเลือกนั้น

ทั้งนี้การคัดเลือกเพื่อการรวมกันใหม่ (selection for recombination) คือการคัดเลือกเพื่อสร้างทายาทจากการรวมกัน จะเป็นการเพิ่มโอกาสการสืบพันธุ์กับรายบุคคลที่เหมาะสมกว่าให้สูงขึ้น ทั้งนี้ยังต้องคงความหลากหลาย (diversity) เพื่อประโยชน์ในการค้นหา ส่วนการคัดเลือกเพื่อการแทน (selection for replacement) เป็นการคัดเลือกรายบุคคลจากพ่อแม่หรือทายาทเพื่อเป็นส่วนประกอบของรุ่นถัดไปนั่นเอง [Dumitrescu00]

ในกระบวนการคัดเลือกนั้นนอกเหนือจากคัดเลือกเพื่อการข้ามสายพันธุ์ การกลายพันธุ์แล้วยังมีกระบวนการบังคับให้อัลกอริทึมคงไว้ซึ่งรายบุคคลที่มีดีที่สุดในจำนวนหนึ่ง หรือการสำเนา (copy) รายบุคคลที่ดีที่สุดจำนวนหนึ่งไปยังรุ่นถัดไป ทั้งนี้รายบุคคลเหล่านี้อาจจะหายไปในที่สุดถ้าไม่ถูกคัดเลือกให้สืบพันธุ์ หรือถูกทำลายโดยการข้ามสายพันธุ์หรือการกลายพันธุ์ ซึ่งกระบวนการนี้ถูกเรียกว่ากระบวนการอภิชนนิยม (elitism) [Mitchell98]

ในตัวดำเนินการคัดเลือกที่จะกล่าวถึงบางตัวดำเนินการในที่นี่นั้น ตัวแปรที่สำคัญที่จะขอกกล่าวถึงก่อนคือ ความดันการคัดเลือก (selection pressure) [Dumitrescu00] หรือ ความดันคัดเลือก (selective pressure) ซึ่งคือระดับ (degree) ที่รายบุคคลที่เหมาะสม (fit) ถูกอนุญาตให้สร้างทายาท หรือสำเนา ในรุ่นถัดไป ทั้งนี้ถ้าความดันการคัดเลือกต่ำจะทำให้รายบุคคลในรุ่นมี

ความน่าจะเป็น (probability) ในการถูกคัดเลือกที่มีเหตุผล แต่ถ้าความดันการคัดเลือกมีค่ามากจะเป็นการสนับสนุนรายบุคคลที่ดีที่สุดโดยรวมเป็นอย่างมาก ดังนั้นการคัดเลือกจะเป็นคล้ายกระบวนการการอธิบายนิยมมากไปด้วย นอกเหนือจากนี้ยังทำให้ประชากรลดความหลากหลายลงอย่างรวดเร็ว ทำให้กระบวนการค้นหาเข้าสู่อย่างรวดเร็ว ดังนั้นเพื่อเป็นการหลีกเลี่ยงการชะงักไหล (stagnation) หรือการลู่เข้าก่อนเวลาอันควร (premature converge) ของกระบวนการค้นหา จึงจำเป็นต้องทำการรวมกันใหม่ และการกลายพันธุ์ อย่างมาก หรือมีค่าความน่าจะเป็นของการรวมกันใหม่ และการกลายพันธุ์สูง ทั้งนี้ถ้าจำนวนรุ่นในอัลกอริทึมมีจำนวนน้อย จำเป็นต้องมีค่าความดันการคัดเลือกที่สูง เพื่อให้การลู่เข้าของการค้นหาดีขึ้น ส่วนเวลาในการครอบครอง (takeover time) [Dumitrescu00] ซึ่งเป็นจำนวนรุ่นในวิวัฒนาการที่จะลู่เข้าหา (converge) ประชากรที่ประกอบไปด้วยรายบุคคลที่เหมาะสมที่สุดทั้งรุ่น โดยที่มีแต่การคัดเลือก และไม่มีตัวดำเนินการอื่นๆ เลย ดังนั้นถ้าความดันการคัดเลือกมีค่าน้อยจะทำให้เวลาในการครอบครองสูง ซึ่งความดันการคัดเลือกจะสามารถบ่งบอกได้ถึงความกรี้ดี (greedy) ของตัวดำเนินการคัดเลือกในการทำให้ประชากรเป็นเอกภาพ ส่วนเวลาในการครอบครองอาจจะเป็นตัววัดความช้าของการลู่เข้าของอัลกอริทึมได้ นั่นเอง

ฟังก์ชันความเหมาะสมเป็นส่วนสำคัญ ในกระบวนการคัดเลือก ซึ่งฟังก์ชันความเหมาะสมถูกแบ่งออกเป็น [Engelbrecht02]

1. การส่งใหม่ของความเหมาะสมชัดเจน (explicit fitness remapping) เป็นการที่ค่าความเหมาะสมของรายบุคคลถูกส่งไปที่เรนจ์ (range) ใหม่ เช่นทำให้เป็นบรรทัดฐาน (normalization) ให้อยู่ในเรนจ์ [0,1] หลังจากนั้นค่าใหม่นี้จะถูกใช้ในการคัดเลือก
  2. การส่งใหม่ของความเหมาะสมโดยปริยาย (implicit fitness remapping) เป็นการใช้ค่าความเหมาะสมของรายบุคคลในการคัดเลือก
- ในหัวข้อนี้จะขอกล่าวถึงตัวดำเนินการคัดเลือกเป็นบางชนิดดังนี้

#### 4.1.4.1 การคัดเลือกแบบสุ่ม (Random Selection)

การคัดเลือกแบบสุ่มนั้น เป็นการคัดเลือกที่ไม่ได้ใช้ค่าความเหมาะสมในการคัดเลือกแต่อย่างใด แต่เป็นการสุ่มเลือก ดังนั้นแต่ละรายบุคคลจะมีโอกาสถูกเลือกที่เท่ากัน

#### 4.1.4.2 การคัดเลือกตามสัดส่วน (Proportional Selection)

การคัดเลือกตามสัดส่วน [Dumitrescu00] นั้นเป็นการคัดเลือกที่เป็นที่นิยม นอกเหนือจากนี้ยังมีการแปรผันของการคัดเลือกตามสัดส่วนอีกด้วย แต่ส่วนความน่าจะเป็นที่เป็นพื้นฐานคือ ความน่าจะเป็นการคัดเลือก (selection probability) ซึ่งคือสมมุติให้ประชากรในรุ่นปัจจุบัน ( $P(t)$ ) มีจำนวนโครโมโซม  $N$  ตัว ดังนี้  $P(t) = \{x^1, x^2, \dots, x^N\}$  โดยที่  $x^i$  คือโครโมโซมตัวที่  $i$  เราสามารถหาค่าความเหมาะสมรวม (total fitness) ได้จาก

$$F = \sum_{i=1}^N f(x^i) \quad (4.4)$$

โดยที่  $f(x^i)$  คือค่าความเหมาะสมของโครโมโซมตัวที่  $i$  และความน่าจะเป็นการคัดเลือกของโครโมโซม  $x^i$  คือ



$$p_i = \frac{f(x^i)}{F} \quad \text{for } i = 1, 2, \dots, N \quad (4.5)$$

สำหรับประชากรที่มีรายบุคคล  $N$  ตัวดำเนินการคัดเลือกจะถูกใช้  $N$  ครั้งสำหรับการคัดเลือกตามสัดส่วน ซึ่งในกรณีนี้เป็นการดำเนินงานที่คล้ายกับการกระจายทวินาม (binomial distribution) ของตัวแปรสุ่ม (random variable) ดังนั้นสำหรับโครโมโซม  $x^i$  จะมีค่าเฉลี่ย หรือค่าคาดหวัง (expected number) ของจำนวนครั้งที่ถูกเลือกไปอยู่ในประชากรกลางคือ

$$n_i = Np_i \quad \text{for } i = 1, 2, \dots, N \quad (4.6)$$

หรือ

$$n_i = N \frac{f(x^i)}{F} = \frac{f(x^i)}{\bar{f}} \quad \text{for } i = 1, 2, \dots, N \quad (4.7)$$

โดยที่  $\bar{f}$  คือค่าเฉลี่ยของค่าความเหมาะสมของประชากร

ในการคัดเลือกตามสัดส่วนนี้ วิธีมอนติคาร์โล (Monte Carlo) หรือ วิธีล้อรูเล็ต (roulette wheel) เพื่อเป็นการยืนยันการเลือกโครโมโซม  $x^i$  ที่มีความน่าจะเป็น  $p_i$  ทั้งนี้ล้อรูเล็ตจะมี  $N$  เซกเตอร์ (sector) แต่ละเซกเตอร์จะสอดคล้องกับแต่ละโครโมโซมโดยที่ขนาดของแต่ละเซกเตอร์จะแปรผันตรงกับความน่าจะเป็นของแต่ละโครโมโซม และล้อรูเล็ตนี้จะถูกหมุน  $N$  ครั้ง ในแต่ละครั้งโครโมโซมจะถูกเลือก 1 ตัว และโครโมโซมนั้นจะถูกสำเนาไปที่ประชากรกลาง อัลกอริทึมที่กล่าวมาข้างต้นเป็นดังนี้ [Dumitrescu00, Mitchell98]

1. For  $i=1$  ถึง  $N$
2. คำนวณหาค่า

$$q_i = \sum_{k=1}^i p_k \quad \text{for } i = 1, 2, \dots, N \quad (4.8)$$

3. End For

4. For  $i=1$  ถึง  $N$

4.1. สุ่มตัวเลข  $\zeta \in [0,1]$  โดยใช้การกระจายเอกรูป (uniform distribution)

4.2. If  $0 \leq \zeta \leq q_1$  โครโมโซม  $x^1$  ถูกเลือก

4.3. If  $q_{i-1} \leq \zeta \leq q_i$  สำหรับ  $i=1, 2, \dots, N$

5. End For

ในการคัดเลือกแบบนี้เป็นตัวแทนของการชักตัวอย่างแบบสโตแคสติก (stochastic sampling) ด้วยการแทนที่ ถึงแม้ว่าจะมีไบแอส (bias) เป็นศูนย์ กระบวนการนี้จะยอมให้โครโมโซมที่ดีที่สุดจะถูกเลือกได้มากกว่า 1 ครั้ง ทำให้มีผู้สืบทอด (descendent) ได้มากกว่า นอกจากนี้รายบุคคลที่มีค่าความเหมาะสมเป็นบวกยังถูกเลือกเพื่อเติมเต็มประชากรทั้งหมด ทั้งนี้กระบวนการล้อรูเล็ตนั้นมีหลายชุด (version) ด้วยกัน แต่อัลกอริทึมที่กล่าวถึงเป็นอัลกอริทึมที่เป็นมาตรฐาน (standard)

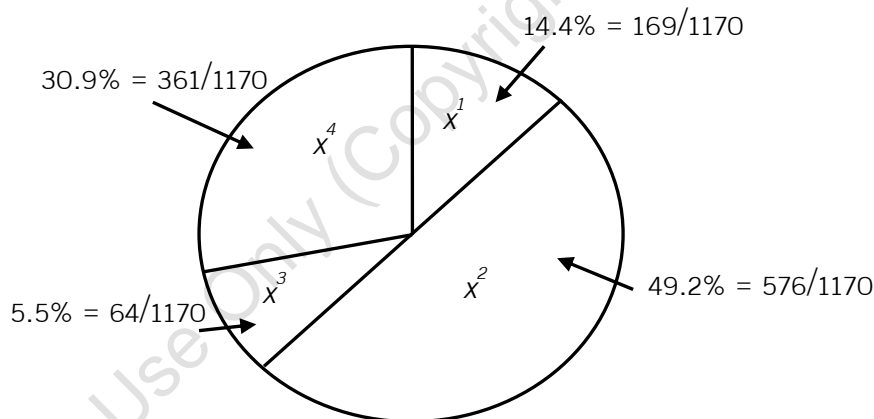
ตัวอย่างที่ 4.4 [Goldberg05] ต้องการหาค่า  $x$  ที่ทำให้

$$\max_x f(x) = \max_x x^2 \quad (4.9)$$

โดยที่  $x$  มีค่าอยู่ระหว่าง 0 ถึง 31 ดังนั้นสามารถใช้สมการที่ 4.3 ในการคำนวณหาค่าความเหมาะสมได้ สมมุติให้แต่ละโครโมโซมมียีนที่มีค่าเป็นไบนารีและมี 5 บิต และสมมุติให้จำนวนประชากรเป็น 4 รายบุคคล สามารถคำนวณหาค่าเฉลี่ยของจำนวนสำเนาในประชากรกลางได้ดังตารางที่ 4.4 และสมมุติให้ในลัทธิการสุ่มค่า  $\xi$  ครั้งที่ 1 ถึง 4 คือ 0.8 0.5 0.1 และ 0.6 ทำให้ครั้งที่ 1 เลือกได้  $x^4$  ในครั้งที่ 2 3 และ 4 เลือกได้  $x^2$   $x^1$  และ  $x^2$  ตามลำดับ

ตารางที่ 4.4 การคำนวณจากลัทธิการสุ่ม

$x$			$f(x)$	$p_i$	$q_i$	$n_i$	จำนวนครั้งที่ถูกเลือกจากลัทธิการสุ่ม
ลำดับที่	โครโมโซม	ค่าจำนวนจริง					
1	01101	13	169	0.14	0.14	0.58	1
2	11000	24	576	0.49	0.63	1.97	2
3	01000	8	64	0.06	0.69	0.22	0
4	10011	19	361	0.31	1.00	1.23	1
ผลรวม			1170	1.00			
ค่าเฉลี่ย			293	0.25			
ค่าที่มากที่สุด			576	0.49			



รูปที่ 4.3 เซกเตอร์ในลัทธิการสุ่มของแต่ละโครโมโซม

จำนวนสำเนาของโครโมโซมในประชากรกลางแสดงในตารางที่ 4.4 เช่นกัน ถ้ามองในลักษณะของลัทธิการสุ่มจะได้ดังรูปที่ 4.3 ซึ่งจะเห็นได้ว่าเซกเตอร์ของโครโมโซม  $x^2$  และ  $x^4$  จะกว้างกว่า  $x^1$  และ  $x^3$  มาก และจากตารางที่ 4.4 จะเห็นว่าถ้า  $n_i$  มากจะบ่งบอกว่าโครโมโซมนั้นจะถูกเลือกมากกว่าในกรณีที่ถ้า  $n_i$  มีค่าน้อยอาจจะไม่ถูกเลือกเลย หลังจากที่ได้เลือก  $x^1$   $x^2$  และ  $x^4$  ถูกเลือก 1 2 และ 1 ครั้ง สำเนาของโครโมโซมเหล่านี้จะถูกส่งไปอยู่ในประชากรกลางดังตารางที่ 4.5 สมมุติให้ทั้งหมดถูกส่งไปยังกลุ่มการจับคู่ จากนั้นทำการสุ่มเพื่อเลือกคู่ ซึ่งปรากฏว่าโครโมโซมที่ 1 จับคู่กับโครโมโซมที่ 2 และเป็นการข้ามสายพันธุ์ที่บิตหลังจากบิตที่ 4 (ตำแหน่งการข้ามสายพันธุ์ (crossover site) ที่บิตที่ 4) การทำการข้ามสายพันธุ์จะอธิบายโดยละเอียดภายหลัง และโครโมโซมที่ 4 จับคู่กับโครโมโซมที่ 3 และเป็นการข้ามสายพันธุ์ที่บิตหลังจากบิตที่ 2 จะได้โครโมโซมในประชากรใหม่ดังตารางที่ 4.5 หลังจากนั้นมีการกลายพันธุ์ (การกลายพันธุ์จะอธิบายโดยละเอียดภายหลัง) โดยที่

สมมุติให้ความน่าจะเป็นการกลายพันธุ์มีค่าเป็น 0.001 ซึ่งการกลายพันธุ์จะเกิดที่แต่ละบิต ทำให้มีจำนวนบิตที่ต้องทำการกลายพันธุ์ทั้งหมด 20 บิต ดังนั้นค่าคาดหวัง (expected value) ของการกลายพันธุ์เท่ากับ  $20(0.001)=0.02$  บิต ดังนั้นในตัวอย่างจึงสมมุติให้ว่าไม่มีการกลายพันธุ์เกิดขึ้นนั่นเอง ดังนั้นประชากรใหม่ที่จะถูกส่งไปที่รุ่นถัดไปได้ และจะเห็นได้ว่าค่าความเหมาะสมที่มากที่สุดคือ 729 ในขณะที่ค่าเฉลี่ยคือ 439 ซึ่งมากกว่าประชากรในรุ่นที่แล้ว

ตารางที่ 4.5 การสร้างประชากรใหม่

ลำดับที่	โครโมโซม	คู่	ตำแหน่งการข้ามสายพันธุ์ (crossover site)	ประชากรใหม่	ค่าของโครโมโซมจำนวนจริง	$f(x)$
1	01101	2	4	01100	12	144
2	11000	1	4	11001	25	625
3	11000	4	2	11011	27	729
4	10011	3	2	10000	16	256
					ผลรวม	1754
					ค่าเฉลี่ย	439
					ค่าที่มากที่สุด	729

ถึงแม้ว่าการคัดเลือกตามสัดส่วนจะเป็นที่นิยม แต่ก็มีข้อเสียเช่นกัน ดังนี้ [Dumitrescu00]

1. การลู่เข้าก่อนเวลาอันควร (premature convergence) เนื่องจากโครโมโซมที่มีความเหมาะสมมากจะถูกเลือกมากกว่าทำให้ทายาทของโครโมโซมเหล่านี้มีอิทธิพล (dominate) ในรุ่นต่อไปมากกว่า ทำให้ความหลากหลายลดลง ทำให้บางพื้นที่ในปริภูมิการค้นหาไม่ถูกรอบคลุม และอาจจะทำให้ลู่เข้าหาตำแหน่งที่เหมาะสมที่สุดเฉพาะที่ (local optimum) ในการที่จะหลีกเลี่ยงเหตุการณ์ อาจจะได้โดยการ ปรับวิธีการเลือกโครโมโซม ซึ่งวิธีที่ง่ายที่สุดคือการใช้ การแปลงสเกล (scale transform) เพื่อให้ค่าความเหมาะสมอยู่ในเรนจ์ หรือใช้กลไกการคัดเลือกสโตแคสติก (stochastic selection mechanism) แบบอื่น
2. การลู่เข้าช้า (slow convergence) ในกรณีที่ประชากรเป็นเอกพันธ์ (homogeneous population) ทำให้แต่ละรายบุคคลมีค่าความเหมาะสมที่แตกต่างกันน้อย ทำให้ความน่าจะเป็นการคัดเลือกเกือบจะเท่ากันสำหรับแต่ละโครโมโซม ทำให้เป็นการเลือกแบบสุ่ม และทำให้การคัดเลือกไม่ใช้การค้นหาเข้าหาตำแหน่งที่เหมาะสมส่วนกลาง (global optimum) และรายบุคคลจะเคลื่อนที่ในปริภูมิค้นหาได้ช้า ทำให้กระบวนการค้นหาลู่เข้าหาตำแหน่งที่เหมาะสมส่วนกลางอย่างช้า ซึ่งทางแก้ในกรณีก็เป็นเช่นเดียวกับการแก้ปัญหาการลู่เข้าก่อนเวลาอันควร

#### 4.1.4.3 การคัดเลือกโดยขึ้นอยู่กับกลไกการสเกล และการจัดอันดับ (Selection Based on Scaling and Ranking Mechanisms)

จากการคัดเลือกที่กล่าวก่อนหน้านี้ทำให้เห็นว่ารายบุคคลที่มีค่าความเหมาะสมที่สูงที่สุดจะมีอิทธิพลมากที่สุด และในบางครั้งอาจจะทำให้กระบวนการค้นหาผิดพลาดได้ ซึ่งหนึ่งในกระบวนการที่จะแก้ปัญหานี้ได้คือการสเกลทั้งที่เป็นแบบอพลวัต (static) และพลวัต (dynamic) และการจัดอันดับดังเช่น [Dumitrescu00]

##### กลไกการสเกลแบบอพลวัต (Static Scaling Mechanisms)

มีได้หลายรูปแบบดังเช่น [Dumitrescu00]

1. การสเกลเชิงเส้น (linear scaling) คือการแปลงโดยฟังก์ชัน  $S$  ดังนี้

$$S(x) = ax + b \quad (4.10)$$

โดยที่ค่าสัมประสิทธิ์  $a$  และ  $b$  ที่เป็นค่าจำนวนจริงที่ไม่เกี่ยวข้องกับปัญหา และฟังก์ชันความเหมาะสมที่ถูกแปลงมา (transformed fitness function ( $tf$ )) คือ

$$tf(x) = S(f(x)) \quad (4.11)$$

โดยที่  $f(x)$  คือค่าความเหมาะสมเดิม และโดยปกติแล้วค่าความเหมาะสมที่ใช้ ( $eval(x)$ ) ก็คือ

$$eval(x) = tf(x) = S(f(x)) = af(x) + b \quad (4.12)$$

2. การสเกลด้วยกฎยกกำลัง (power law scaling) ค่าความเหมาะสมหลังจากการแปลงคือ

$$eval(x) = tf(x) = S(f(x)) = (f(x))^u \quad (4.13)$$

โดยที่  $u$  คือค่าจำนวนจริง และโดยปกติจะมีค่าเข้าใกล้ 1

3. การสเกลแบบลอการิทึม (logarithmic scaling) ในการสเกลแบบนี้การแปลงแบบลอการิทึมที่สามารถนำมาใช้ได้มีอยู่หลายรูปแบบ ซึ่งหนึ่งในนั้นคือการคัดเลือกแบบโบซแมน (Boltzmann selection) ดังนี้

$$tf(x) = S(f(x)) = e^{\frac{f(x)}{T}} \quad (4.14)$$

โดยที่  $T$  คือตัวแปรในการควบคุมความดันการคัดเลือก และจะมีค่าลดลงในระหว่างกระบวนการค้นหา

##### กลไกการสเกลแบบพลวัต (Dynamic Scaling Mechanisms)

มีได้หลายรูปแบบดังเช่น [Dumitrescu00]

1. การตัดซิกมาร์ (sigma truncation) หรือบางครั้งถูกเรียกว่าการสเกลแบบซิกมาร์ (sigma scaling) ซึ่งเป็นการสเกลที่ใช้ข้อมูลของประชากรปัจจุบัน ( $P(t)$ ) ด้วย ซึ่งการแปลงเป็นดังนี้ ให้  $T$  เป็นฟังก์ชันที่เป็นค่าไม่เป็นค่าลบ (non-negative valued function) ดังนี้

$$T(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else} \end{cases} \quad (4.15)$$

ดังนั้นฟังก์ชันการสเกลด้วยการตัดซิกมาร์  $S$  คือ

$$S(x) = T\left(x - (m_t - c\sigma_t)\right) \quad (4.16)$$

โดยที่  $m_t$  คือค่าเฉลี่ยของค่าความเหมาะสมของประชากร  $P(t)$  ในขณะที่  $\sigma_t$  คือค่าเบี่ยงเบนมาตรฐาน (standard deviation) ของค่าความเหมาะสมของประชากร  $P(t)$  ส่วน  $c$  คือค่าจำนวนเต็ม (integer) เล็กๆ ที่เป็นตัวแทนของปัญหา โดยปกติจะถูกตั้งค่าไว้ที่ 2 ดังนั้นค่าความเหมาะสมหลังจากการแปลงคือ

$$eval(x) = tf(x) = s(f(x)) = T(f(x) - (m_t - c\sigma_t)) \quad (4.17)$$

จากสมการจะเห็นได้ว่าการแปลงนี้จะทำให้ค่าที่เป็นลบถูกตั้งค่าเป็น 0 และในการแปลงแบบนี้จะคงค่าความดันการคัดเลือกให้เป็นค่าคงที่มากกว่าที่จะให้ขึ้นอยู่กับค่าความเบี่ยงเบนมาตรฐานของค่าความเหมาะสมในประชากร ซึ่งทำให้จำกัดผลกระทบที่จะเกิดจากรายบุคคลที่มีค่าความเหมาะสมสูง ในประชากรนั่นเอง ทำให้

$$eval(x) = \begin{cases} f(x) - (m_t - c\sigma_t) & \text{if } f(x) > (m_t - c\sigma_t) \\ 0 & \text{if } f(x) \leq (m_t - c\sigma_t) \end{cases} \quad (4.18)$$

แต่ในบางครั้งการสเกลแบบตัดซิกมาร์อาจจะเป็นแบบ ทำให้

$$eval(x) = \begin{cases} 1 + \frac{f(x) - m_t}{\sigma_t} & \text{if } \sigma_t \neq 0 \\ 0 & \text{if } \sigma_t = 0 \end{cases} \quad (4.19)$$

2. การสเกลด้วยหน้าต่าง (window scaling) เป็นการสเกลที่ใช้ข้อมูลจาก  $g$  รุ่นสุดท้าย โดยที่  $g \geq 1$  เช่นถ้าสนใจเฉพาะประชากรในรุ่นปัจจุบัน  $P(t)$  ทำให้ค่าความเหมาะสมหลังจากการแปลงคือ

$$eval(x) = f(x) - \min_{y \in P(t)} f(y) \quad (4.20)$$

แต่ถ้าสนใจในจำนวนรุ่นที่มากขึ้นทำให้

$$eval(x) = f(x) - \min_{y \in W} f(y) \quad (4.21)$$

โดยที่  $W$  คือเซตของรายบุคคลที่อยู่ในประชากร  $g$  รุ่นสุดท้าย ที่  $g \geq 1$  รวมทั้งรุ่นปัจจุบัน การส่งใหม่ของค่าความเหมาะสมสำหรับปัญหาการหาค่าน้อย (Fitness remapping for minimization problem) [Dumitrescu00]

โดยปกติค่าความเหมาะสมมากจะเป็นการบ่งบอกว่ารายบุคคลนั้นดี ซึ่งในกรณีนี้จะเหมาะสมในการหาค่าตอบที่เหมาะสมที่สุดสำหรับการหาค่ามาก แต่ถ้าเป็นปัญหาการหาค่าน้อย เช่น ต้องการหาค่า  $x$  ที่ทำให้ฟังก์ชัน  $f(x)$  ที่ค่าน้อยที่สุด จึงจำเป็นต้องมีการแปลงค่าจากฟังก์ชันเพื่อให้เป็นไปตามอัลกอริทึมแบบพันธุกรรม หนึ่งใน การแปลงที่เป็นไปได้คือ

$$tf(x) = M - f(x) \quad (4.22)$$

โดยที่  $M$  คือค่าขอบเขตบน (upper bound) ของฟังก์ชันจุดมุ่งหมาย (objective function) แต่ถ้าไม่ทราบค่าที่มากที่สุดส่วนกลาง (global maximum) ของฟังก์ชันจุดมุ่งหมาย สมการค่าความเหมาะสมหลังการแปลงจะเป็น

$$eval(x(t)) = tf(x(t)) = M_t - f(x(t)) \quad (4.23)$$

โดยที่  $x(t) \in P(t)$  และ  $M_t$  เป็นค่าที่มากที่สุดของฟังก์ชัน  $f$  ที่ได้จนถึงเวลา  $t$  หรือรุ่นที่  $t$  หรืออาจจะ เป็นค่าที่มากที่สุดของฟังก์ชัน  $f$  ในรุ่นของประชากร  $P(t)$  นอกเหนือจากการแปลงดังที่กล่าวข้าง สามารถใช้การแปลงดังนี้

$$eval(x(t)) = \frac{1}{K + f(x(t)) - f_{\min}(t)} \quad (4.24)$$

โดยที่  $K$  คือค่าคงที่บวก ในขณะที่  $f_{\min}(t)$  คือค่าที่น้อยที่สุดของฟังก์ชัน  $f(x)$  ที่ได้จนถึงเวลาที่  $t$  หรือรุ่นที่  $t$  หรืออาจจะ เป็นค่าที่น้อยที่สุดของฟังก์ชัน  $f(x)$  ในรุ่นของประชากร  $P(t)$  ทั้งนี้อาจจะใช้ การแปลงอีกรูปแบบหนึ่งคือ

$$eval(x) = \frac{1}{K + f(x)} \quad (4.25)$$

โดยที่  $K$  คือค่าคงที่บวก

#### การคัดเลือกโดยใช้การจัดอันดับ (Rank-based Selection)

การคัดเลือกโดยใช้การจัดอันดับเป็นการคัดเลือกที่สนใจไปที่การจัดอันดับของค่าความ

เหมาะสมของรายบุคคล ดังนั้นรายบุคคลในประชากรที่มีขนาด  $N$  ( $|P(t)|=N$ ) จะถูกจัดอันดับตาม ความเหมาะสมของรายบุคคลในกระบวนการค้นหา ซึ่งรายบุคคลที่อยู่ในอันดับหนึ่งจะเป็น รายบุคคลที่ไม่เหมาะสมที่สุด ในขณะที่รายบุคคลที่อยู่ในอันดับที่  $N$  จะเป็นรายบุคคลที่เหมาะสม ที่สุด ทั้งนี้ความน่าจะเป็นการคัดเลือกจะถูกตั้งให้กับแต่ละรายบุคคลจากฟังก์ชันของอันดับใน ประชากรนั่นเอง

ในการคัดเลือกจากการจัดอันดับนั้นการให้ค่า (assign) คะแนนแบบนามธรรม (subjective score) ให้กับคำตอบของปัญหาได้ง่ายกว่าการระบุฟังก์ชันจุดมุ่งหมายที่แท้จริง นอกเหนือจากนี้ในการคัดเลือกตามสัดส่วนนั้น กลุ่มของรายบุคคลที่มีค่าความเหมาะสมมากอาจจะ มีผลต่อประชากรทั้งหมดได้ การจัดอันดับนี้สามารถช่วยในการหลีกเลี่ยงการลู่เข้าก่อนเวลาอันควร เนื่องจากการมีอิทธิพลของกลุ่มนี้ได้ และการจัดอันดับยังช่วยในการลดความดันการคัดเลือก เมื่อ ความเบี่ยงเบนของค่าความเหมาะสมมีค่าสูง และเพิ่มความดันการคัดเลือกถ้าความเบี่ยงเบนการ คัดเลือกต่ำอีกด้วย

ทั้งนี้การคัดเลือกจากการจัดอันดับมีดังนี้

1. การคัดเลือกจากการจัดอันดับเชิงเส้น (linear ranking selection) ซึ่งในการจัดอันดับเชิง เส้นนี้ ความน่าจะเป็นการคัดเลือกของแต่ละรายบุคคลจะถูกกำหนดจากฟังก์ชันเชิงเส้น ของอันดับของแต่ละรายบุคคลนั้น ส่วนการจัดอันดับเชิงเส้นสามารถทำได้โดยการ กำหนดค่าคาดหวังของจำนวน (expected number) ของทายาทของรายบุคคลที่ดีที่สุด ในแต่ละรุ่น สมมติให้  $Max$  เป็นค่าคาดหวังของจำนวนของทายาทของรายบุคคลที่ดีที่สุด ในประชากร ค่าความน่าจะเป็นการคัดเลือกของโครโมโซม  $x^i$  คือ [Dumitrescu00]

$$p_i = \frac{1}{N} \left[ Min + (Max - Min) \frac{r_i - 1}{N - 1} \right] \quad (4.26)$$

โดยที่  $r_i$  เป็นอันดับของโครโมโซม  $x^i$  ดังนั้น  $r=i$  เสมอเช่น  $r_1=1$  ในขณะที่  $r_N=N$  นั่นเอง และ  $min$  คือค่าคาดหวังของจำนวนของทายาทของรายบุคคลที่มีอันดับเป็น 1 ดังนั้น อาจจะบอกได้ว่า  $Max$  และ  $Min$  เป็นอัตราสุ่ม (sampling rate) ของรายบุคคลที่ดีที่สุด และแย่ที่สุดตามลำดับ และเช่นเดิมค่าคาดหวังของทายาทของโครโมโซม  $x^i$  คือ

$$n_i = Np_i \quad (4.27)$$

ถ้าจำนวนประชากรในแต่ละรุ่นคงที่จะได้ว่า

$$\sum_{i=1}^N n_i = \sum_{i=1}^N Np_i = N \quad (4.28)$$

จากสมการที่ 4.28 และ 4.26 ทำให้สมการที่ 4.26 กลายเป็น

$$\sum_{i=1}^N Np_i = N \quad (4.29)$$

$$\sum_{i=1}^N N \frac{1}{N} \left[ \text{Min} + (\text{Max} - \text{Min}) \frac{r_i - 1}{N - 1} \right] = \sum_{i=1}^N \text{Min} + \sum_{i=1}^N (\text{Max} - \text{Min}) \frac{r_i - 1}{N - 1} = N \quad (4.30)$$

$$N\text{Min} + \frac{1}{N - 1} (\text{Max} - \text{Min}) \sum_{i=1}^N (r_i - 1) = N \quad (4.31)$$

แต่เนื่องจาก 
$$\sum_{i=1}^N (r_i - 1) = \sum_{i=1}^N (i - 1) = \sum_{i=0}^{N-1} i = \frac{N(N-1)}{2} \quad (4.32)$$

ทำให้สมการที่ 4.31 กลายเป็น

$$\text{Min} + \frac{1}{2} (\text{Max} - \text{Min}) = 1 \quad (4.33)$$

ดังนั้น 
$$\text{Min} = 2 - \text{Max} \text{ หรือ } \text{Max} + \text{Min} = 2 \quad (4.34)$$

แต่โดยปกติมีเงื่อนไขว่า 
$$\text{Min} \geq 0 \quad (4.35)$$

และโดยปกติแล้ว 
$$\text{Min} \leq \text{Max} \quad (4.36)$$

ทำให้ 
$$\text{Max} \leq 2 \quad (4.37)$$

และ 
$$\text{Max} \geq 1 \quad (4.38)$$

ดังนั้นสรุปได้ว่า 
$$1 \leq \text{Max} \leq 2 \quad (4.39)$$

จากสมการที่ 4.26 ค่าความน่าจะเป็นของโครโมโซมอันดับที่  $N$  หรือโครโมโซมที่ดีที่สุดคือ

$$p_N = \frac{\text{Max}}{N} \quad (4.40)$$

ส่วนค่าความน่าจะเป็นของโครโมโซมอันดับที่ 1 หรือโครโมโซมที่แย่ที่สุดคือ

$$p_1 = \frac{2 - \text{Max}}{N} = \frac{\text{Min}}{N} \quad (4.41)$$

แต่ในบางครั้งค่าความน่าจะเป็นการคัดเลือกเป็นฟังก์ชันของตัวแปร  $\text{Max}$  ทำให้

$$p_i = \frac{1}{N} \left[ \text{Max} + 2(\text{Max} - 1) \left( 1 - \frac{r_i - 1}{N - 1} \right) \right] \quad (4.42)$$

2. การคัดเลือกจากการจัดอันดับไม่เชิงเส้น (nonlinear ranking selection) ในการจัดอันดับแบบนี้เป็นเช่นเดียวกับกรณีแรกนั่นคือ รายบุคคลที่ดีที่สุดจะอยู่อันดับที่  $N$  ในขณะที่รายบุคคลที่แย่ที่สุดจะอยู่อันดับที่ 1 เพียงแต่ค่าความน่าจะเป็นการคัดเลือกจะเป็นฟังก์ชัน

ไม่เชิงเส้นของอันดับนั่นเอง เช่นความน่าจะเป็นการคัดเลือกของโครโมโซม  $x^i$  คือ [Dumitrescu00]

$$p_i = \frac{1}{c} \left( 1 - e^{1-r_i} \right) \quad (4.43)$$

แต่สำหรับ  $r_1 = 1$  นั้นความน่าจะเป็นการคัดเลือกจะเป็น  $p_1 = 0$  เสมอ

ไม่ว่าจะใช้อันดับเชิงเส้นหรือไม่เชิงเส้นการคัดเลือกจะเป็นอีกรูปแบบของการคัดเลือกตามสัดส่วนซึ่งคือการสุ่มแบบสโทแคสติกสากล (stochastic universal sampling (SUS)) ซึ่งอัลกอริทึมนี้จะต่างกับวิธีสุ่มโดยตรงที่แทนที่ล๊อตจะถูกหมุน  $N$  ครั้งเพื่อเลือก  $N$  โครโมโซมด้วยตัวชี้ (pointer) 1 อัน แต่จะเป็นการหมุนเพียงแค่ 1 ครั้ง แต่มีตัวชี้  $N$  อันที่อยู่ห่างกันด้วยระยะประมาณเท่ากันเพื่อเลือก  $N$  โครโมโซม ดังนั้นอัลกอริทึมนี้เป็นดังนี้ [Mitchell98]

1.  $ptr = \text{Rand}()$ ; /\*สุ่มตัวเลขในช่วง  $[0,1]$  ด้วยการกระจายเอกรูป (uniform distribution)\*/
2. For ( $sum=i=0$ ;  $i<N$ ;  $i++$ )
  - 2.1. For ( $sum += n_i$ ;  $sum > ptr$ ;  $ptr++$ )
  - 2.2. Select( $i$ );
  - 2.3. End For
3. End For

ซึ่ง SUS จะรับประกันว่าโครโมโซม  $x^i$  จะถูกสืบทอดอย่างน้อย  $\lfloor n_i \rfloor$  ครั้งแต่ไม่เกิน  $\lceil n_i \rceil$  ครั้งเสมอ

ตัวอย่างที่ 4.5 สมมติให้ประชากรปัจจุบันมีรายบุคคลดังนี้  $x^1=15$ ,  $x^2=10$  และ  $x^3=20$  ซึ่งในตัวอย่างนี้ต้องการแก้ปัญหาของสมการที่ 4.9 และต้องการใช้การคัดเลือกจากการจัดอันดับเชิงเส้น ทำให้ต้องจัดอันดับใหม่ ทำให้โครโมโซมที่มีอันดับ 1 2 และ 3 กลายเป็น  $x^1=10$ ,  $x^2=15$  และ  $x^3=20$  สมมติให้ค่าความน่าจะเป็นคำนวณจากสมการที่ 4.26 โดยที่กำหนดให้  $Max=1.2$  ทำให้  $Min=0.8$  และค่าคาดหวังของจำนวนของโครโมโซม  $x^i$  ตามสมการที่ 4.27 ได้ดังตารางที่ 4.6

ตารางที่ 4.6 ความน่าจะเป็นการคัดเลือกและค่าคาดหวังของจำนวนของทายาท

โครโมโซม	$f(x)=x^2$	$r_i$	$p_i$	$n_i$
$x^1=10$	100	1	$\frac{2-1.2}{3} = 0.27$	$3(0.27)=0.81$
$x^2=15$	225	2	$\frac{1}{3} \left[ 0.8 + (1.2 - 0.8) \frac{2-1}{3-1} \right] = 0.33$	$3(0.33)=0.99$
$x^3=20$	400	3	0.4	$3(0.4)=1.2$

ในการคัดเลือกโดยใช้ SUS สมมติให้ค่า  $ptr$  ถูกสุ่มแล้วได้ค่าเท่ากับ 0.8 และจากอัลกอริทึมทำให้โครโมโซมที่ถูกเลือกเป็น  $x^1$ ,  $x^3$  และ  $x^3$  ตามลำดับ



### การคัดเลือกจากการแข่งขัน (Tournament Selection)

การคัดเลือกตามสัดส่วนที่ได้กล่าวมาแล้วนั้นมีการคำนวณที่เกี่ยวข้องกับการหาค่าเฉลี่ยของค่าความเหมาะสมรวมทั้งการหาค่าคาดหวังของจำนวนของทายาท ในขณะที่การคัดเลือกจากการจัดอันดับต้องทำการเรียงลำดับ (sorting) ของค่าความเหมาะสมของทั้งประชากร ซึ่งอาจจะทำให้เสียเวลา ส่วนการคัดเลือกจากการแข่งขันจะคล้ายกับการคัดเลือกจากการจัดอันดับ ในเรื่องของความดันการคัดเลือก แต่การดำเนินการมีประสิทธิภาพและสามารถทำคู่ขนานกันได้ด้วย ซึ่งตัวอย่างของการคัดเลือกจากการแข่งขันที่จะกล่าวถึงเป็นการแข่งขันแบบไบนารี (binary tournament) ซึ่งเป็นการแข่งขันระหว่าง 2 โครโมโซม และในการคัดเลือกแบบนี้อาจจะ หรือ อาจจะไม่มี การนำโครโมโซมที่เข้าแข่งขันกลับไปยังประชากรเดิมหรือไม่ก็ได้ มีด้วยกันหลายรูปแบบ ดังนี้ [Dumitrescu00]

1. การแข่งขันเชิงกำหนด (deterministic tournament) ซึ่งกระบวนการเป็นดังนี้
  - 1.1. สุ่มโครโมโซมจากประชากรมา 2 โครโมโซม
  - 1.2. คำนวณหาค่าความเหมาะสมของทั้งสองโครโมโซม
  - 1.3. โครโมโซมที่มีค่าความเหมาะสมมากกว่า (ผู้ชนะ (winner)) จะถูกเลือกและสำเนาไปยัง  $P^1$  (ประชากรกลาง (intermediate population))

ปกติแล้วทั้งสองโครโมโซมจะถูกส่งกลับไปยังประชากรเดิม (original population) ซึ่งนี่คือการแข่งขันที่มีการส่งกลับ (tournament with reinsertion) และทั้งสองโครโมโซมอาจจะถูกเลือกอีกครั้งก็เป็นได้ และกระบวนการคัดเลือกแบบนี้จะถูกทำซ้ำ  $N$  ครั้งเพื่อให้ได้โครโมโซม  $N$  ตัวในประชากรกลาง ในบางครั้งโครโมโซมผู้ชนะจะถูกกลายพันธุ์ และถูกส่งกลับไปยังประชากรเดิมเพื่อแทนที่ผู้แพ้ (loser) ในการแข่งขัน และในบางครั้งแทนที่จะสุ่มคู่การแข่งขัน แต่จะเป็นการให้ทุกโครโมโซมสุ่มเลือกคู่เพื่อการแข่งขันก็เป็นได้

2. การแข่งขันตามความน่าจะเป็น (probabilistic tournament) หรือ การแข่งขันแบบสโตแคสติก (stochastic tournament) ซึ่งจะจำลองการแข่งขันเชิงกำหนดเพียงแต่ในขั้นตอนที่ 1.3 จะถูกเปลี่ยนเป็น

- 1.3. สุ่มค่า  $R \in [0,1]$  แล้วถ้า  $R < p$  โดยที่  $p$  คือค่าตัวแปรที่ถูกกำหนด (ตัวอย่างเช่น  $p=0.3$ ) รายบุคคลที่มีค่าเหมาะสมน้อยกว่าจะถูกเลือก มิฉะนั้นรายบุคคลที่เหมาะสมกว่าจะถูกเลือก

โดยปกติแล้วค่า  $p$  จะมีค่าน้อยลงตามเวลา (ตามรุ่น) เช่น  $p$  สำหรับรุ่นที่  $t$  มีค่าเป็น

$$p(t) = p_0 e^{-ct} \quad (4.44)$$

โดยที่  $p_0 \in (0,1)$  และค่า  $c$  คือค่าคงที่บวก กระบวนการนี้เป็นการรับประกันว่าการค้นหายังคงดำเนินการต่อไปในขณะที่ความหลากหลายยังคงอยู่ ซึ่งในการค้นหาที่ดำเนินไปเรื่อยๆ ความน่าจะเป็นของการเก็บรักษาคำตอบที่ดีที่สุดมีเพิ่มขึ้น ในบางครั้งค่า  $p$  ก็อาจจะเป็น

$$p(t) = p_0 \frac{c}{1 + \ln t} \quad (4.45)$$

โดยที่  $c > 0$  เสมอ

3. การแข่งขันแบบโบลทซ์แมน (Boltzmann tournament) สมมติให้  $f$  เป็นฟังก์ชันที่ต้องการหาค่าน้อย (minimization problem) และให้  $x$  และ  $y$  เป็นคำตอบปัจจุบันและคำตอบทางเลือก ในการแข่งขันแบบไบนารี โครโมโซม  $x$  จะชนะหรือถูกเลือกด้วยความน่าจะเป็น

$$p(x) = \frac{1}{1 + e^{\frac{f(x)-f(y)}{T}}} \quad (4.46)$$

โดยที่  $T$  เป็นอุณหภูมิ ที่จะมีค่าลดลงตามรุ่น ส่วนผู้ชนะในการแข่งขันจะเป็นคำตอบปัจจุบันในการแข่งขันครั้งต่อไป

#### 4.1.5 กระบวนการอภิชนนิยม (Elitism)

กระบวนการนี้เกี่ยวข้องกับการเลือกเซตของรายบุคคลจากรุ่นปัจจุบันไปยังรุ่นต่อไป แต่รายบุคคลเหล่านี้อาจจะหายไปภายหลังถ้าไม่ถูกเลือกในการจับคู่ หรือถ้าถูกทำให้รวมกันใหม่ หรือกลายพันธุ์ ดังนั้นกระบวนการนี้ทำให้รายบุคคลที่มีค่าความเหมาะสมสูงอยู่รอดได้มากกว่า 1 รุ่น ซึ่งเป็นการทำให้คำตอบที่ดีคงอยู่ในประชากรมากกว่า 1 รุ่น นั่นเอง ถึงแม้ว่ากระบวนการนี้อาจจะทำให้เกิดการลู่เข้าก่อนเวลาอันควร แต่ในหลายครั้งพบว่าทำให้อัลกอริทึมแบบพันธุกรรมมีประสิทธิภาพมากขึ้น [Dumitrescu00] ทั้งนี้จำนวนของรายบุคคลที่อยู่รอดไปยังรุ่นต่อไปโดยที่ไม่มีการกลายพันธุ์ ถูกเรียกว่าช่องว่างระหว่างรุ่น (generation gap) ถ้าช่องว่างระหว่างรุ่นเป็น 0 แสดงว่าในรุ่นต่อไปมีแต่รายบุคคลใหม่ ไม่ได้มีรายบุคคลเดิมแม้แต่หนึ่งราย [Engelbrecht02]

#### 4.1.6 ตัวดำเนินการรวมกันใหม่ (Recombination Operator)

ตัวดำเนินการรวมกันใหม่นั้นบางครั้งถูกเรียกว่าตัวดำเนินการข้ามสายพันธุ์ (cross over operator) ซึ่งเป็นการรวมกันหรือข้ามสายพันธุ์ระหว่างโครโมโซมที่ถูกเลือก โดยที่กระบวนการนี้เป็นการเลียนแบบการข้ามสายพันธุ์ที่เกิดขึ้นในธรรมชาติ ซึ่งเป็นการข้ามสายพันธุ์ระหว่าง 2 โครโมโซม ที่โดยปกติถูกเรียกว่าการข้ามสายพันธุ์แบบ (2,2) หมายถึงการข้ามสายพันธุ์ระหว่าง 2 โครโมโซม (พ่อ-แม่) และทำให้ได้โครโมโซมลูก 2 โครโมโซม สมมุติให้  $X$  เป็นเซตของของโครโมโซมที่มีความยาวคงที่เป็น  $L$  และ  $C$  คือ ตัวดำเนินการข้ามสายพันธุ์แบบ (2,2) ดังนั้นการข้ามสายพันธุ์แบบนี้คือ [Dumitrescu00]

$$C: X \times X \rightarrow X \times X \quad (4.47)$$

ซึ่งคือ โครโมโซม พ่อ-แม่  $x = x_1 x_2 x_3 \dots x_L$  และ  $y = y_1 y_2 y_3 \dots y_L$  ทำให้ตัวดำเนินการข้ามสายพันธุ์

$$C(x,y) = (x',y') \quad (4.48)$$

โดยที่  $x' = x_1 x_2 x_3 \dots x_{k-1} x_k y_{k+1} \dots y_L$  และ  $y' = y_1 y_2 y_3 \dots y_{k-1} y_k x_{k+1} \dots x_L$  ซึ่งค่า  $k$  คือตำแหน่งการข้ามสายพันธุ์ (crossover site) หรือจุดการข้ามสายพันธุ์ (crossover point) หรือจุดพัก (breakpoint) ที่เป็นตัวเลขที่ถูกสุ่มมาจากเซต  $\{1, 2, \dots, L-1\}$  โดยเป็นการสุ่มจากการแจกแจงเอกรูป (uniform distribution) นั่นเอง

หน้าที่ของการข้ามสายพันธุ์คือการกระทำที่กระตุ้นให้การค้นหามีความก้าวหน้า และเพื่อการรับประกันในปริภูมิการค้นหา ทั้งนี้กระบวนการข้ามสายพันธุ์มีได้หลายรูปแบบดังที่จะกล่าวต่อไป

#### 4.1.6.1 การข้ามสายพันธุ์แบบหนึ่งจุด (One-point Crossover)

ให้  $L$  เป็นความยาวของโครโมโซมในประชากร และตำแหน่งการข้ามสายพันธุ์ (crossover site) หรือจุดการข้ามสายพันธุ์ (crossover point) หรือจุดหัก (breakpoint) จะเป็นตัวเลขจำนวนเต็ม  $k \in \{1, 2, \dots, L-1\}$  ซึ่ง  $k$  จะระบุตำแหน่งในโครโมโซมที่ถูกตัด และรวมกันใหม่ทำให้เกิดโครโมโซมลูก [Dumitrescu00] โดยปกติแล้ว ค่า  $k$  จะถูกเลือกแบบสุ่มโดยที่ไม่ให้มีความมากกว่า  $L$  [Dumitrescu00]

สมมติให้โครโมโซม พ่อ-แม่  $(x, y)$  มีความยาว  $L$  คือ

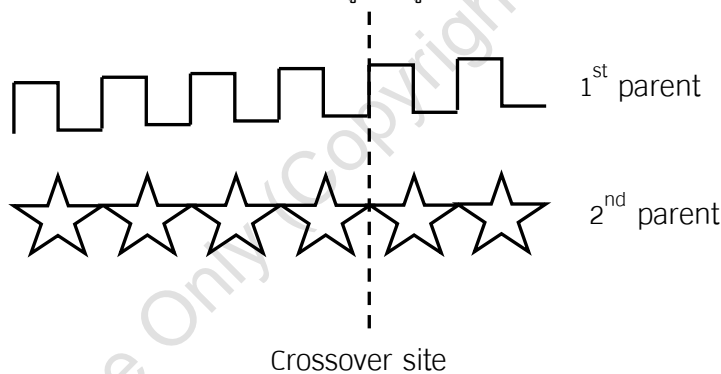
$$x = x_1 x_2 x_3 \dots x_{k-1} x_k x_{k+1} \dots x_L \text{ และ } y = y_1 y_2 y_3 \dots y_{k-1} y_k y_{k+1} \dots y_L$$

และตำแหน่งการข้ามสายพันธุ์อยู่ที่ปิดที่  $k$  ทำให้ได้โครโมโซมลูก  $x'$  และ  $y'$  เป็น

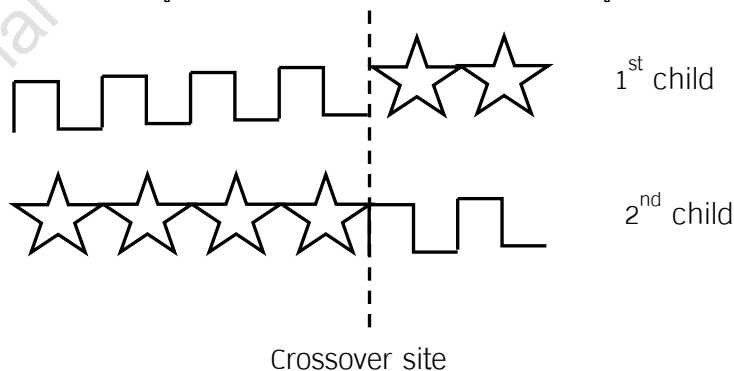
$$x' = x_1 x_2 x_3 \dots x_{k-1} x_k y_{k+1} \dots y_L \text{ และ } y' = y_1 y_2 y_3 \dots y_{k-1} y_k x_{k+1} \dots x_L \text{ นั่นเอง}$$

ตัวอย่างที่ 4.6 สมมติให้โครโมโซม พ่อ-แม่คือ  $x=10001110$  และ  $y=00111001$  ถ้าตำแหน่งการข้ามสายพันธุ์อยู่ที่  $k=5$  จะได้โครโมโซมลูกเป็น  $x'=10001001$  และ  $y'=00111110$

ตัวอย่างที่ 4.7 สมมติให้โครโมโซม พ่อ-แม่เป็นดังรูปที่ 4.4 และให้ตำแหน่งการข้ามสายพันธุ์ (crossover site) ที่ปิดที่ 4 หรือ  $k=4$  จะได้รูปดังรูปที่ 4.5



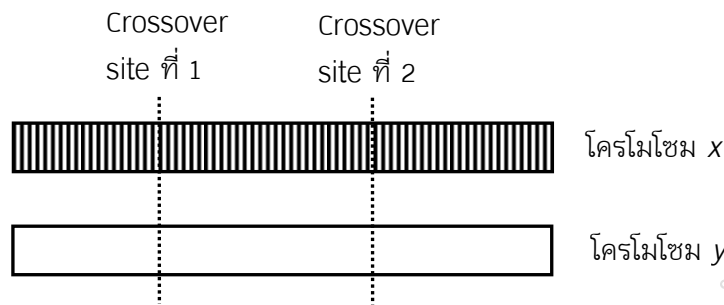
รูปที่ 4.4 โครโมโซม พ่อ-แม่ ที่ผ่านการถูกเลือก



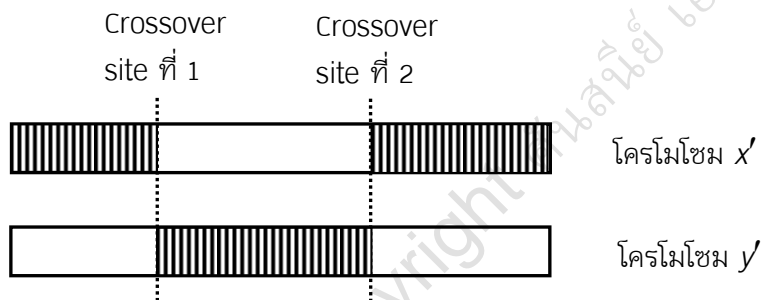
รูปที่ 4.5 โครโมโซมลูกหลังจากการข้ามสายพันธุ์

#### 4.1.6.2 การข้ามสายพันธุ์แบบสองจุด (Two-point Crossover)

ในธรรมชาติการรวมกันใหม่นั้นอาจจะมีจุดการข้ามสายพันธุ์ได้มากกว่า 1 จุด ซึ่งการสร้างลูกก็จะเป็นไปตามกฎที่ถูกตั้งไว้ในแต่ละกรณี [Dumitrescu00] สำหรับการข้ามสายพันธุ์แบบสองจุดคือการข้ามสายพันธุ์ที่มีจุดการข้ามสายพันธุ์ 2 จุดด้วยกัน และทั้ง 2 จุดนี้จะถูกเลือกแบบสุ่ม โดยที่ตัวดำเนินการนี้เป็นไปตามสมการที่ 4.47 สมมติให้โครโมโซมพ่อ-แม่ มีลักษณะดังรูปที่ 4.6 และโครโมโซมลูกทั้งสองโครโมโซม จะเกิดการสลับข้อมูลระหว่างจุดทั้งสอง ดังรูปที่ 4.7



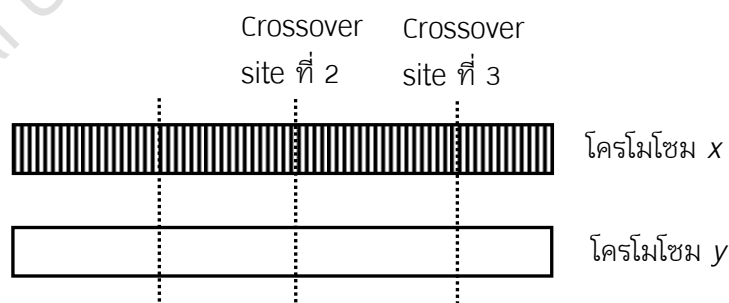
รูปที่ 4.6 โครโมโซมพ่อ-แม่ สำหรับการข้ามสายพันธุ์แบบสองจุด



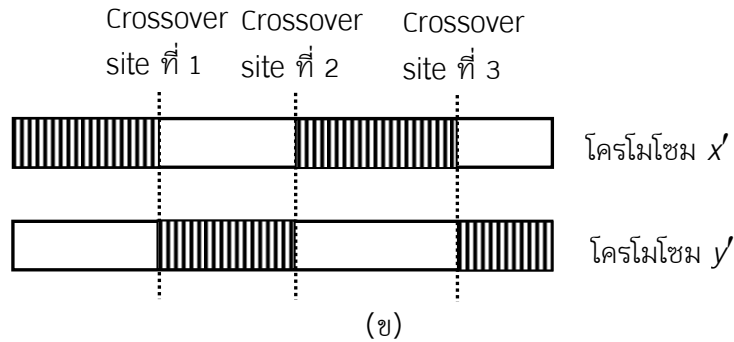
รูปที่ 4.7 โครโมโซม

#### 4.1.6.3 การข้ามสายพันธุ์แบบ T จุด (T-point Crossover)

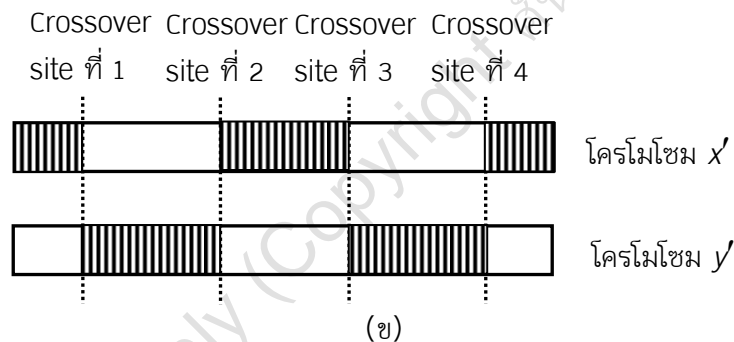
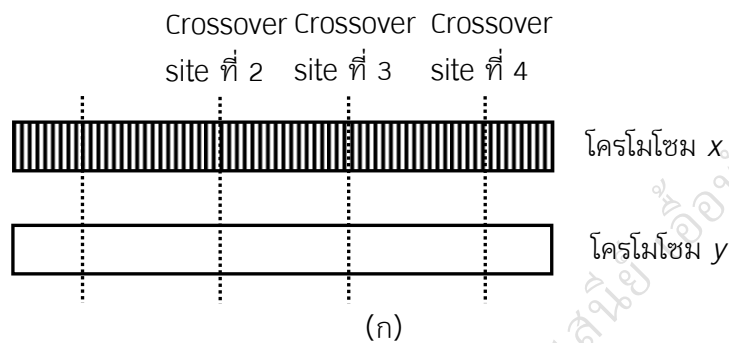
ในกรณีนี้คือการข้ามสายพันธุ์ที่มีการข้ามสายพันธุ์มากกว่า 2 จุด เช่นมีจุดข้ามสายพันธุ์ 3 จุด สมมติให้โครโมโซมพ่อแม่เป็นดังรูปที่ 4.6 และมีจุดข้ามสายพันธุ์ 3 จุด จะได้โครโมโซมลูก ที่เกิดจากการสลับข้อมูลดังรูปที่ 4.8 หรือตัวอย่างการข้ามสายพันธุ์ 4 จุด ซึ่งโครโมโซมลูกที่ได้ก็เกิดจากการสลับข้อมูล ระหว่างจุดข้ามสายพันธุ์ทั้ง 4 ดังรูปที่ 4.9



(ก)



รูปที่ 4.8 (ก) โครโมโซมพ่อ-แม่ ( $x, y$ ) และ (ข) โครโมโซมลูกหลังจากการข้ามสายพันธุ์ 3 จุด



รูปที่ 4.9 (ก) โครโมโซมพ่อ-แม่ ( $x, y$ ) และ (ข) โครโมโซมลูกหลังจากการข้ามสายพันธุ์ 4 จุด  
ทั้งนี้การข้ามสายพันธุ์แบบ 2 จุดจะถูกใช้มากที่สุดเนื่องจากการทำแบบนี้เป็นการลดผลกระทบจากการทำให้สับสน (disruptive effect) แต่ทั้งนี้ทั้งนั้น การข้ามสายพันธุ์แบบใดมีข้อดีหรือข้อเสียอย่างไร ไม่เคยปรากฏการพิสูจน์แต่อย่างใด

ทั้งนี้จะขอกล่าวถึงเค้าร่าง (schema) [Dumitrescu00] เพื่อที่จะให้การอธิบายเรื่องผลกระทบการทำให้สับสน ง่ายขึ้น ซึ่งโดยปกติอัลกอริทึมแบบพันธุกรรมใช้การเข้ารหัสแบบไบนารี ดังนั้นแต่ละตำแหน่งขอลยีน ถูกแทนด้วยค่า 0 หรือ 1 และเช่นเดิมให้  $X$  เป็นเซตของของโครโมโซมที่มีความยาวคงที่เป็น  $L$  ดังนั้น  $X = \{0, 1\}^L$  ซึ่งโครโมโซมอาจจะถูกมองเป็นเวกเตอร์ในไฮเพอร์คิวบ์ (hypercube)  $X$  ดังนั้นนิยามที่เกี่ยวข้องกับเค้าร่างมีดังนี้

นิยาม 4.1 [Dumitrescu00] เค้าร่าง (schema) ของปริภูมิโครโมโซม (chromosome space)  $X$  คือสายอักขระ (string) ? มีความยาว  $L$  ที่ประกอบไปด้วย 0, 1 หรือ \* ซึ่งสัญลักษณ์ \* แทน ไม่สนใจ หรือที่ตำแหน่งนั้นอาจจะเป็น 0 หรือ 1 ก็ได้ ดังนั้นเค้าร่างจะส่วยย่อยของเซต  $\{0, 1, *\}^L$  และเค้าร่างอาจจะถูกแปลเป็นระนาบเกินของปริภูมิการค้นหาได้

ตัวอย่างที่ 4.8 [Dumitrescu00] สมมติให้มีเค้าร่าง

$$S = 1 * * 0$$

ซึ่งเค้าร่างนี้อาจจะเป็นโครโมโซมใดก็ได้ ใน 4 โครโมโซมดังต่อไปนี้

$$x_1 = 1 0 0 0$$

หรือ

$$x_2 = 1 0 1 0$$

หรือ

$$x_3 = 1 1 0 0$$

หรือ

$$x_4 = 1 1 1 0$$

ซึ่งโครโมโซม  $x_1, x_2, x_3, x_4$  คือกรณีตัวอย่าง (instance) หรือตัวแทน (representative) ของเค้าร่าง  $S$  นั้นเอง

**นิยาม 4.2 [Dumitrescu00]** สัญลักษณ์ที่ถูกนิยามดี (well-defined symbol) ในเค้าร่างจะไม่ใช้สัญลักษณ์  $*$  ดังเช่นยีนในตำแหน่งที่ 1 และ 4 ของ เค้าร่างในตัวอย่าง 4.8 จะเป็นสัญลักษณ์ที่ถูกนิยามดี

**นิยาม 4.3 [Dumitrescu00]** โครโมโซม  $x$  เป็นตัวอย่างของเค้าร่าง  $S$  ถ้าสัญลักษณ์ที่ถูกนิยามดีของ  $S$  อยู่ในตำแหน่งเดียวกันกับค่าของยีนในโครโมโซม ในตำแหน่งเดียวกัน ดังเช่นโครโมโซม  $x_1, x_2, x_3, x_4$  ในตัวอย่างที่ 4.8 นั้นเอง

ดังนั้นสำหรับโครโมโซมที่มีความยาว  $L$  จะเป็นตัวอย่างของเค้าร่างที่ต่างกันอยู่  $2^L$  เค้าร่าง

**นิยาม 4.4 [Dumitrescu00]** ลำดับ (order) ของเค้าร่างคือจำนวนของสัญลักษณ์ที่ถูกนิยามดีในเค้าร่างนั้น

**นิยาม 4.5 [Dumitrescu00]** ความยาวของเค้าร่างคือระยะระหว่าง ตำแหน่งแรกและตำแหน่งสุดท้าย เฉพาะ

**ตัวอย่างที่ 4.9 [Dumitrescu00]** สมมติให้มีเค้าร่าง

$$S = * 0 * 1 0 1 * * 0 *$$

จากนิยามที่ 4.9 ลำดับของเค้าร่างนี้คือ 5 ส่วนความยาวหาได้จาก ตำแหน่ง (loci) ของสัญลักษณ์ที่ถูกนิยามดีอยู่ที่ตำแหน่งที่ 2 และ 9 ดังนั้นความยาวของเค้าร่างจะเป็น 7 นั้นเอง

**นิยาม 4.6 [Dumitrescu00]** ค่าความเหมาะสมของเค้าร่าง  $S$  ในประชากรหนึ่งคือค่าเฉลี่ยของค่าความเหมาะสมของตัวอย่างของ  $S$  ในประชากรนั้น

สำหรับการอธิบายว่าทำไมการข้ามสายพันธุ์แบบหลายจุดนั้น ในบางกรณี การข้ามสายพันธุ์บางอย่างไม่สามารถเกิดขึ้นได้ในการข้ามสายพันธุ์จุดเดียว ดังตัวอย่างที่ 4.10

**ตัวอย่างที่ 4.10 [Dumitrescu00]** สมมติให้มีเค้าร่าง

$$S_1 = 0 1 * * * * * 1 1$$

และ

$$S_2 = * * * 1 0 1 * * * * *$$

ในการข้ามสายพันธุ์แบบจุดเดียวของตัวอย่างจากเค้าร่าง  $S_1$  และ  $S_2$  นั้นไม่สามารถสร้างลูกตามเค้าร่าง  $S_3 = 0 1 * 1 0 1 * * * 1 1$  ได้ทั้งนี้เพราะ  $S_1$  ถูกทำให้สับสนแต่  $S_3$  อาจจะถูกสร้างได้จาก

การข้ามสายพันธุ์แบบ 2 จุดนั่นเอง แต่อย่างไรก็ตามบางเค้รางก็ไม่ได้ถูกสร้างจากการข้ามสายพันธุ์แบบ 2 จุด แต่ถูกสร้างได้จากการข้ามสายพันธุ์แบบหลายจุด

อัลกอริทึมของการข้ามสายพันธุ์แบบ  $T$  จุด มีลักษณะเป็นดังนี้ [Dumitrescu00]

1. สำหรับแต่ละโครโมโซมจากประชากรกลาง  $P^1$ 
  - 1.1. สุ่ม ค่า  $q \in [0,1]$  ซึ่งในการสุ่มให้คำนึงถึงการกระจายแบบเอกรูป (uniform distribution)
  - 1.2. If  $q < p_c$  โดยที่  $p_c$  คือความน่าจะเป็นของการจับคู่ (mating probability) แล้วโครโมโซมนั้นจะถูกพิจารณาให้จับคู่ มิฉะนั้นโครโมโซมนั้นก็จะไม่เข้าไปอยู่ในกลุ่มการจับคู่
2. ให้จำนวนโครโมโซมที่ถูกเลือกในขั้นที่ 1 เป็น  $m^*$ 

ถ้า  $m^*$  เป็นเลขคู่ แล้วให้ทำการจับคู่แบบสุ่ม

ถ้า  $m^*$  เป็นเลขคี่ โครโมโซมที่ถูกเลือกจะถูกเอาออกจากกลุ่ม หรือ ทำการเลือกโครโมโซมตัวใหม่จาก  $P^1$  เข้ามาเพิ่มในกลุ่ม ซึ่งการเอาออกหรือการเพิ่มใหม่นี้เป็นการกระทำแบบสุ่ม (โดยการสร้างตัวเลขแบบสุ่มที่ใช้ในการเลือก) เช่นกัน
3. การข้ามสายพันธุ์จะถูกกระทำในโครโมโซมที่ได้ในขั้นที่ 2 ตามขั้นตอนดังนี้
  - 3.1. สำหรับแต่ละคู่ สุ่มค่า  $k$  โดยที่  $1 \leq k \leq L$  ( $L$  คือความยาวของโครโมโซม และค่า  $k$  เป็นตำแหน่งของการข้ามสายพันธุ์)

ถ้าการข้ามสายพันธุ์ ที่  $T \geq 2$  แล้วให้ทำการสุ่มค่า  $k_1, k_2, \dots, k_T$  ซึ่งเป็นตำแหน่งของการข้ามสายพันธุ์ และคู่โครโมโซมนั้นจะทำการข้ามสายพันธุ์ตามตำแหน่งเหล่านี้
  - 3.2. ลูก (descendant) ที่ได้จากขั้นที่ 3.1 จะเป็นสมาชิกที่เป็นไปได้ของประชากร  $P(t+1)$  ซึ่งยังคงเป็นสมาชิกของ  $P^2$  เพื่อโอกาสในการกลายพันธุ์
  - 3.3. พ่อแม่ (parent) ของโครโมโซมที่ถูกสร้างขึ้นใหม่นี้จะถูกเอาออกจาก  $P^1$
  - 3.4. ส่วนโครโมโซมที่เหลือใน  $P^1$  จะถูกเติมไปในประชากร  $P^2$  เช่นกัน

#### 4.1.6.4 การข้ามสายพันธุ์แบบสลับ (Punctuated Crossover)

การข้ามสายพันธุ์แบบนี้ [Dumitrescu00] ในบางครั้งจะถูกเรียกว่าการข้ามสายพันธุ์แบบปรับตัวได้ (adaptive crossover) ซึ่งในการข้ามสายพันธุ์แบบนี้จะต้องทำการเก็บตำแหน่งการข้ามสายพันธุ์ เพื่อที่ ทำให้รู้ได้ว่า ถ้าตำแหน่งการข้ามสายพันธุ์ที่สร้างลูกที่ไม่ดี ตำแหน่งการข้ามสายพันธุ์นั้นจะถูกพิจารณาอีกในอนาคต แต่ถ้าตำแหน่งการข้ามสายพันธุ์นั้นสร้างลูกที่ดี ตำแหน่งนั้นจะยังคงถูกเก็บไว้ หรือทำงานตามปกติ

ตำแหน่งการข้ามสายพันธุ์นี้จะผ่านกระบวนการวิวัฒนาการเช่นกัน ดังนั้นตำแหน่งการข้ามสายพันธุ์แบบสลับนี้ จะถูกเข้ารหัสเข้าไปในโครโมโซมด้วย ซึ่งในส่วนนี้จะถูกเรียกว่าลิสของการสลับ (punctuation list) ซึ่งลิสนี้จะใช้ในการระบุตำแหน่งการข้ามสายพันธุ์สำหรับการข้ามสายพันธุ์หลายจุด ซึ่งการนำลิสการสลับเข้าไปในโครโมโซมทำให้โครโมโซมมีความยาวเป็น  $2L$  โดยที่มีลักษณะดังนี้

$$C = x_1 \ x_1 \dots \ x_L \ k_1 \ k_1 \dots \ k_L \quad (4.49)$$

โดยที่ถ้า  $k_i = 1$  แสดงว่าตำแหน่ง  $i$  เป็นตำแหน่งการข้ามสายพันธุ์ แต่ถ้า  $k_i = 0$  แสดงว่าตำแหน่ง  $i$  ไม่เป็นตำแหน่งการข้ามสายพันธุ์

ทั้งนี้ถ้ามีการสร้างลูก ลูกจะมีลิสของการสลับที่ได้มาจากการยูเนียน (union) ของลิสนี้จากพ่อแม่ และโดยปกติในการตั้งประชากรเริ่มต้นค่า  $k_i$  ของโครโมโซมในประชากรจะมีโอกาสที่ค่า  $k_i = 1$  นี้น้อย (โดยปกติค่า  $p_i = 0.04$ ) และค่า  $k_i$  ในลิสของการสลับก็มีการกลายพันธุ์ และมีการปรับค่าเองได้ด้วย

#### 4.1.6.5 การข้ามสายพันธุ์แบบเอกรูป (Uniform Crossover)

การข้ามสายพันธุ์แบบนี้ไม่ใช่ตำแหน่งการข้ามสายพันธุ์ที่ถูกกำหนดไว้แล้ว และสำหรับแต่ละยีนในโครโมโซมของลูก (offspring) จะมีตัวแปรกลาง (global parameter) มากำหนดว่ายีนนั้นจะมาจาก พ่อแม่ตัวแรก หรือตัวที่สอง ดังนั้นสำหรับแต่ละตำแหน่งของโครโมโซมในลูกตัวที่หนึ่ง พ่อแม่ตัวที่จะให้ค่าที่ตำแหน่งนั้นของลูกตัวที่หนึ่งจะถูกเลือกด้วยความน่าจะเป็น  $p$  ซึ่งการข้ามสายพันธุ์แบบเอกรูป ด้วยความน่าจะเป็น  $p$  นี้จะให้ค่าเฉลี่ยของตัวเลขของตำแหน่งการข้ามสายพันธุ์เป็น  $pL$  และที่ตำแหน่งเดียวกันของลูกตัวที่สองจะเอาค่ามาจากพ่อแม่ตัวที่เหลือ ซึ่งการข้ามสายพันธุ์แบบนี้เป็นนิยามทั่วไปของการข้ามสายพันธุ์แบบหลายจุด (generalization of multiple-point crossover) ดังนั้นในการสร้างตัวแปรกลางนี้จะอยู่ในรูปของเวกเตอร์หน้ากาก (mask vector ( $m$ )) โดยที่อัลกอริทึมในการเวกเตอร์เป็นดังนี้ [Engelbrecht05]

1. ตั้งค่า  $m_i = 0$  สำหรับ  $i = 1, 2, \dots, L$

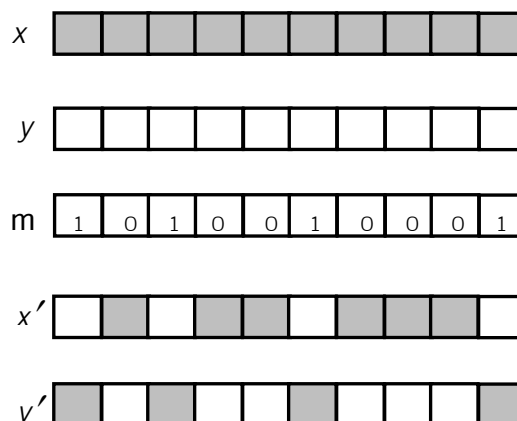
2. For each  $i$

2.1. คำนวณค่าตัวแปรสุ่ม  $\xi \sim U(0,1)$

2.2. If  $\xi \leq p_{at\_i}$  then  $m_i = 1$  โดยที่  $p_{at\_i}$  คือความน่าจะเป็นของการข้ามสายพันธุ์ที่ตำแหน่ง  $i$

จากอัลกอริทึมจะเห็นว่าถ้า  $p_{at\_i} = 0.5$  แสดงว่าทุกตำแหน่งมีโอกาสที่จะเป็นตำแหน่งของการข้ามสายพันธุ์เท่ากัน

ตัวอย่างที่ 4.11 [Engelbrecht05] สมมติให้โครโมโซมพ่อแม่  $x$  และ  $y$  และเวกเตอร์หน้ากาก  $m$  มีลักษณะดังรูปที่ 4.10



รูปที่ 4.10 การข้ามสายพันธุ์แบบเอกรูป ด้วยเวกเตอร์หน้ากาก  $m$



ในการสร้างลูกตัวที่หนึ่ง  $x'$  นั้นจะนำค่ามาจากพ่อแม่  $y$  ถ้า  $m_i = 1$  และจากพ่อแม่  $x$  ถ้า  $m_i = 0$  ส่วนโครโมโซมลูก  $y'$  นั้น จะทำในสิ่งที่ตรงข้ามกันดังรูปที่ 4.10

ทั้งนี้ในการข้ามสายพันธุ์นี้อาจจะมีการข้ามสายพันธุ์แบบอื่นเช่น การสร้างพ่อแม่หลายตัว แต่ได้ลูก 1 ตัว หรือชนิดอื่นๆ อีก หรือแม้กระทั่งการข้ามสายพันธุ์ในกรณีของโครโมโซมที่มีค่ายีนเป็นจำนวนจริง [Engelbrecht05] นั้นอาจจะใช้การสลับค่าตำแหน่งแบบที่เคยกล่าวมาแล้ว หรืออาจจะมีการข้ามสายพันธุ์โดยใช้ตัวดำเนินการเชิงเส้น (linear operator) เช่นมีพ่อแม่  $x$  และ  $y$  ทำการสร้างลูก 3 ตัวด้วยสมการที่ 4.50, 4.51 และ 4.52 โดยที่จะทำการเลือกลูกที่ดีที่สุดไว้ 2 ตัว

$$x'_j = x_j + y_j \quad \text{for } 1 \leq j \leq L \quad (4.50)$$

$$x'_j = 1.5x_j - 0.5y_j \quad \text{for } 1 \leq j \leq L \quad (4.51)$$

$$x'_j = -0.5x_j + 1.5y_j \quad \text{for } 1 \leq j \leq L \quad (4.52)$$

หรืออาจจะมีการสร้างลูก 1 ตัวจากพ่อแม่  $x$  และ  $y$  โดยใช้สมการต่อไปนี้

$$x'_j = U(0,1)[y_j - x_j] + y_j \quad \text{for } 1 \leq j \leq L \quad (4.53)$$

โดยมีข้อแม้ว่า  $y$  ต้องไม่แย่กว่า  $x$  เป็นต้น นอกจากนี้ยังมีกระบวนการข้ามสายพันธุ์ในกรณีจำนวนจริงอีกมากมาย

#### 4.1.7 การกลายพันธุ์ (Mutation)

กระบวนการทำให้มีการสร้างรายบุคคลที่ไม่สามารถสร้างจากกระบวนการอื่น และการกลายพันธุ์นี้เป็นกระบวนการที่สนับสนุนการข้ามสายพันธุ์ให้ยีนมีค่าที่ครอบคลุมทั้งมิติการค้นหา ทั้งนี้ในการกลายพันธุ์นี้จะมีค่าความน่าจะเป็นของการกลายพันธุ์ (mutation probability ( $p_m$ )) หรือบางครั้งถูกเรียกว่าอัตราการกลายพันธุ์ (mutation rate) ประกอบการพิจารณาการกลายพันธุ์ของแต่ละตำแหน่ง ดังนั้น ถ้ามี  $N$  โครโมโซม และแต่ละโครโมโซมมีจำนวนยีน  $L$  และถ้าทุกตำแหน่งมี  $p_m$  ที่เท่ากันจะทำให้ค่าเฉลี่ยของจำนวนบิตที่จะกลายพันธุ์เป็น

$$B = NLp_m \quad (4.54)$$

โดยปกติค่า  $p_m$  จะมีค่าน้อย และอาจจะเป็น  $p_m \in [0.001, 0.01]$

ในปัจจุบันมีอัลกอริทึมสำหรับการกลายพันธุ์มากมายแต่ที่นิยมและง่าย จะมีการกลายพันธุ์แบบแข็ง (strong mutation) และการกลายพันธุ์แบบอ่อน (weak mutation) ซึ่งอัลกอริทึมสำหรับการกลายพันธุ์แบบแข็งเป็นดังนี้ [Dumitrescu00]

1. For each chromosome และ For each position of the chromosome

1.1. คำนวณหาค่า  $q$  โดยสุ่มจาก  $U(0,1)$

1.2.If  $q < p_m$  ค่าที่ตำแหน่งนั้นจะสลับค่าจาก 1 เป็น 0 หรือ จาก 0 เป็น 1

If  $q \geq p_m$  ค่าตำแหน่งนั้นจะไม่มีการเปลี่ยนแปลง

ทั้งนี้สำหรับการกลายพันธุ์แบบอ่อนนั้นจะมีความแตกต่างกับการกลายพันธุ์แบบแข็ง ตรงที่ถ้าจะมีการกลายพันธุ์ที่ตำแหน่ง ค่าของยีนจะถูกเลือกแบบสุ่มอีกครั้งว่าจะป็น 0 หรือ 1 นั่นคืออัลกอริทึมสำหรับการกลายพันธุ์แบบอ่อนจะเหมือนกับอัลกอริทึมของการกลายพันธุ์แบบแข็งทุกอย่าง ยกเว้นในขั้นที่ 1.2 ซึ่งจะเปลี่ยนเป็นดังนี้ [Dumitrescu00]

1.2.(ใหม่) If  $q < p_m$  ค่าที่ตำแหน่งนั้นจะกลายพันธุ์ โดยที่ค่า 0 หรือ 1 จะถูกเลือกแบบสุ่ม และนำมาแทนที่ค่าเดิมที่ตำแหน่งนั้น

If  $q \geq p_m$  ค่าตำแหน่งนั้นจะไม่มีเปลี่ยนแปลง

นอกเหนือจากนี้ ยังมีการกลายพันธุ์แบบไม่เป็นเอกรูป (non-uniform mutation) อีกด้วย ซึ่งการกลายพันธุ์แบบนี้จะเกี่ยวข้องกับค่าความน่าจะเป็นของการกลายพันธุ์ นั่นคือ ในรุ่นแรกค่าความน่าจะเป็นของการกลายพันธุ์จะมีค่าสูง และจะมีค่าลดลงตามกาลเวลา (รุ่น) ซึ่งกระบวนการนี้เป็นการทำให้เห็นว่า การค้นหาจะกระทำในวงกว้าง ในขณะที่ในรุ่นหลังๆ ค่าความน่าจะเป็นนี้ลดลง เพื่อให้การค้นหาเป็นการค้นหาในวงจำกัดเฉพาะที่

ค่าของความน่าจะเป็นในกรณีของการกลายพันธุ์แบบไม่เป็นเอกรูป อาจจะมีลักษณะดังนี้

$$p_m(t) = p_m e^{-\beta t} \quad (4.55)$$

โดยที่  $p_m$  เป็นค่าความน่าจะเป็นการกลายพันธุ์ในรุ่นแรก และ  $t$  เป็นดัชนีของรุ่น ในขณะที่  $\beta$  เป็นค่าจำนวนจริงที่  $\geq 1$  หรืออาจจะใช้สมการ

$$p_m(t) = \left( \frac{\alpha}{\delta} \right)^{\frac{1}{2}} \left( \frac{e^{\frac{-\beta t}{2}}}{\frac{1}{L^2 N}} \right) \quad (4.56)$$

โดยที่  $\alpha, \beta$  และ  $\delta$  เป็นตัวแปรที่มีค่าจำนวนจริงบวก หรืออาจจะเป็น

$$p_m(t) = \frac{1}{2 + \frac{L-2}{T}t} \quad (4.57)$$

โดยที่  $0 \leq t \leq T$  ซึ่ง  $T$  เป็นจำนวนรุ่นที่มากที่สุด และโดยปกติสำหรับกรณีนี้ ค่า  $p_m(0) = 0.5$  ดังนั้น

ค่าที่น้อยที่สุดจะเป็น  $p_m(T) = \frac{1}{L}$  นั่นเอง

หรือในบางครั้งเราอาจจะคำนวณหาความน่าจะเป็นที่ขึ้นกับค่าความเหมาะสมก็ได้ดังเช่น

$$p_m(x) = \frac{1}{2(f(x) + 1) - L} \quad (4.58)$$

ซึ่งเป็นค่าความน่าจะเป็นของการกลายพันธุ์ของโครโมโซม  $x$  นั่นเอง นอกจากนี้ยังมีการกลายพันธุ์แบบอื่นๆ อีกมากมาย ที่ไม่ได้กล่าวถึงในที่นี้

#### 4.1.8 ตัวดำเนินการการผกผัน (Inversion Operator)

โดยปกติตัวดำเนินการที่ใช้เป็นประจำในอัลกอริทึมแบบพันธุกรรมนั้น จะไม่ค่อยมีการใช้ตัวดำเนินการการผกผันมากนัก ทั้งนี้ตัวดำเนินการการผกผันนี้เป็นการกระทำบนโครโมโซม 1 ตัว โดยที่จะมีการ กลับค่าระหว่างค่าใน 2 ตำแหน่งของโครโมโซม ซึ่งตำแหน่งทั้ง 2 นี้จะถูกเลือกมาแบบสุ่ม [Dumitrescu00] เช่นสมมติให้โครโมโซมมีค่าเป็น 10111011 และตำแหน่งที่ถูกสุ่มมาคือ บิตที่ 2 และ 6 จะทำให้ได้โครโมโซม 10011111 เป็นต้น

โดยสรุปแล้วอัลกอริทึมแบบพันธุกรรมมีลักษณะดังต่อไปนี้ [Dumitrescu00]

1. ตั้งค่า  $t = 0$

- ทำการตั้งค่า โครโมโซมในประชากรรุ่นแรก ( $P(0)$ ) แบบสุ่ม  
ทำการเลือกวิธีการคัดเลือก และถ้าจำเป็นวิธีการแปลงสเกล
2. ทำการคำนวณหาค่าความเหมาะสมของโครโมโซมในประชากร  $P(t)$   
โครโมโซมที่ดีที่สุด (most successful) จะถูกเก็บไว้
  3. ทำการคัดเลือก  $N$  ครั้ง  
โครโมโซมที่ถูกเลือกถูกส่งไปที่ประชากรกลาง  $P^1$  ซึ่งที่จำนวนโครโมโซมเป็น  $N$  เช่นกันโดยที่บางโครโมโซมอาจจะมีมากกว่า 1 สำเนา (copy) ซึ่งโครโมโซมเหล่านี้จะเป็นโครโมโซมให้เลือก (candidate) สำหรับกลุ่มจับคู่ (MP)
  4. ทำการข้ามสายพันธุ์ กับโครโมโซมใน MP  
โครโมโซมที่ถูกสร้างใหม่จะถูกไปยัง  $P^2$   
พ่อแม่ของโครโมโซมที่ถูกสร้างใหม่ จะถูกเอาออกไปจาก  $P^1$   
ส่วนโครโมโซมที่เหลือใน  $P^1$  จะเป็นสมาชิกของ  $P^2$  ด้วยเช่นกัน
  5. ทำการกลายพันธุ์ให้กับโครโมโซมใน  $P^2$  แล้วส่งต่อไปยัง  $P(t+1)$
  6. ตั้งค่า  $t = t+1$   
ถ้า  $t < T$  (ซึ่ง  $T$  เป็นจำนวนรุ่นที่มากที่สุด) แล้วให้กลับไปทำขั้น 2 มิฉะนั้นให้หยุดการทำงาน  
ทั้งนี้อัลกอริทึมแบบพันธุกรรมที่สรุปนี้อาจจะมีการปรับโดยให้มีการใช้ตัวดำเนินการผกผันหรือตัวดำเนินการอื่นๆ ก็ได้ และหลังจากอัลกอริทึมหยุดการทำงาน โดยปกติคำตอบจะเป็นรายบุคคลที่ดีที่สุดในเรื่องสุดท้าย หรือรายบุคคลที่ดีที่สุดในรุ่นทั้งหมด

ตัวอย่างที่ 4.12 สมมติว่าต้องการหาค่า  $x_1$  และ  $x_2$  ที่ทำให้

$$\max_{x_1, x_2} f(x_1, x_2) \quad (4.59)$$

โดยที่  $f(x_1, x_2) = \frac{1}{1 + x_1^2 + x_2^2}$  และสำหรับตัวอย่างนี้การแทนค่าตอบไม่ได้ใช้สายปิด แต่ใช้

สายที่แต่ละตำแหน่งเป็นค่าจำนวนจริงแทน ดังนั้น 1 โครโมโซมจะมี 2 ยีน และในแต่ละรุ่นมี 4 โครโมโซม และเนื่องจากในตัวอย่างนี้เป็นปัญหาที่ทำให้ฟังก์ชัน  $f$  มีค่ามากที่สุดจึงสามารถใช้ค่าจากฟังก์ชัน  $f$  เป็นค่าเหมาะสมได้

ตารางที่ 4.7 โครโมโซมรุ่นที่ 1

โครโมโซม	$x_1$	$x_2$	ค่าเหมาะสม
$C_0$	-1	2	0.167
$C_1$	-2	3	0.007
$C_2$	1.5	0.0	0.31
$C_3$	0.5	-1.0	0.44

สมมติให้รุ่นที่ 1 โครโมโซมทั้ง 4 หลังจากถูกสุ่มค่ายีนจะมีค่าดังตารางที่ 4.7 สมมติให้หลังจากการคัดเลือก 4 ครั้ง ได้โครโมโซม  $C_3$ ,  $C_3$ ,  $C_2$  และ  $C_0$  ซึ่งถูกส่งไปประชากร  $P^1$  และเมื่อทำการเลือกคู่จะได้คู่  $(C_3, C_2)$  และ  $(C_3, C_0)$  ซึ่งการข้ามสายพันธุ์จะทำการสลับค่าที่ตำแหน่งที่ 2 เสมอ ซึ่งโครโมโซมลูก (แสดงในตารางที่ 4.8) ที่ได้นี้จะถูกส่งไปยังประชากร  $P^2$  ส่วนการกลายพันธุ์

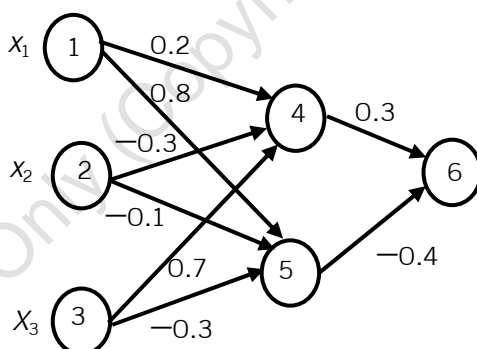
นั่นจะเป็นการเปลี่ยนแปลงค่าอื่น โดยการลุ่มค่าจำนวนจริงมาเพิ่มในตำแหน่งอื่นที่จะมีการเปลี่ยนค่านั่นเอง

อัลกอริทึมนี้จะทำงานไปจนครบจำนวนรุ่นที่มากที่สุดที่ตั้งไว้ หรือหยุดตามเงื่อนไขของค่าความเหมาะสม

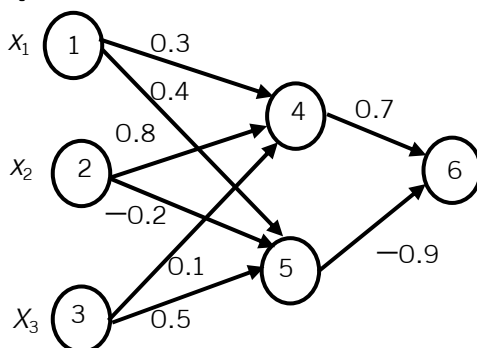
ตารางที่ 4.8 โครโมโซมลูก

โครโมโซม	$x_1$	$x_2$	ค่าเหมาะสม
$C_0$	0.5	0	0.8
$C_1$	1.5	-1.0	0.24
$C_2$	0.5	2.0	0.19
$C_3$	-1.0	-1.0	0.33

ตัวอย่างที่ 4.13 [Mitchell98] ตัวอย่างนี้จะแสดงการหาเวกเตอร์น้ำหนักที่ดีที่สุดสำหรับเปอร์เซปตรอนหลายชั้น โดยที่การแทนค่าตอบหรือโครโมโซมไม่ได้ใช้สายปิด แต่ใช้สายที่แต่ละตำแหน่งเป็นค่าจำนวนจริงแทน ดังเช่นตัวอย่างที่ 4.2 ซึ่งโดยปกติค่าน้ำหนักและค่าไบแอสของแต่ละเซลล์จะอยู่ที่ตำแหน่งเดิมในสายแบบนี้เสมอ ซึ่งโดยปกติจะเป็นการอ่านจากบนลงล่างและจากเอาต์พุตมายังอินพุต สมมติให้เปอร์เซปตรอนหลายชั้นเป็นโครงข่าย 3-2-1 และเป็นโครงข่ายประสาทเทียมที่ไม่มีไบแอส มีลักษณะดังรูปที่ 4.11 ดังนั้นโครโมโซมที่เป็นตัวแทนของโครงข่ายนี้คือ  $(0.3, -0.4, 0.2, 0.8, -0.3, -0.1, 0.7, -0.3)$



รูปที่ 4.11 โครงข่ายประสาทเทียม 3-2-1



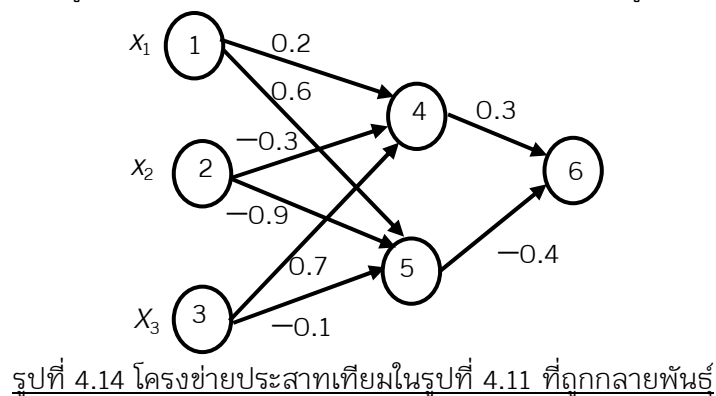
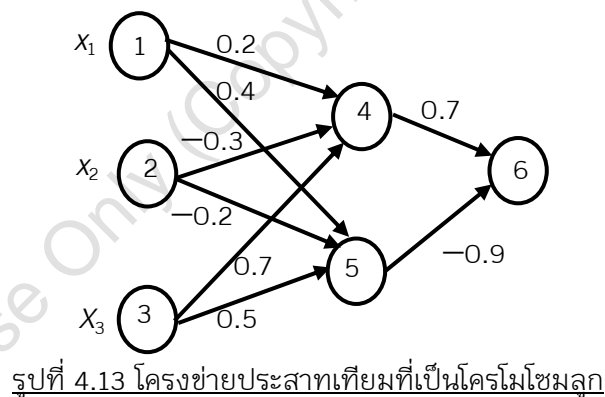
รูปที่ 4.12 โครงข่ายประสาทเทียมที่เป็นโครโมโซมแม่

ในการคำนวณหาค่าเหมาะสมในกรณีนี้คือการหาค่าเฉลี่ยของความผิดพลาดกำลังสอง ( $E_{av}$ ) ของทุกเวกเตอร์อินพุต ซึ่งโครโมโซมที่ให้ค่านี้น้อยที่สุดจะเป็นโครโมโซมที่ดีที่สุด ดังนั้นต้องทำการ

คำนวณหาสเกลของค่าที่ได้เพื่อให้เป็นค่าความเหมาะสมที่ใช้ ( $eval(x)$ ) ซึ่งเป็นการแปรผกผันกับ  $\mathcal{E}_{av}$  นั่นเอง

สมมติให้มี 50 โครโมโซมใน 1 รุ่น และสำหรับตัวอย่างนี้อัลกอริทึมจะทำงานไปจนถึงรุ่นที่ 200 ซึ่งในตอนเริ่มต้นจะทำการตั้งค่าน้ำหนักของแต่ละโครโมโซมโดยการสุ่มให้แต่ละค่าน้ำหนักมีค่าอยู่ในช่วง  $-1.0$  ถึง  $1.0$  และในการคัดเลือกเป็นการคัดเลือกโดยใช้การจัดอันดับ ส่วนกระบวนการอื่นๆก็เป็นไปตามอัลกอริทึมพันธุกรรมดังที่เคยกล่าวมาแล้ว แต่สำหรับการข้ามสายพันธุ์ในกรณีนี้จะเป็นการข้ามสายพันธุ์ที่มีพ่อแม่ 2 ตัวแต่ได้ลูก 1 ตัว โดยที่สำหรับแต่ละเซลล์ที่ไม่ใช่เซลล์อินพุตในโครโมโซมลูก จะทำการเลือกค่าน้ำหนักที่พุ่งเข้า (incoming link) เซลล์นั้นจากโครโมโซมพ่อหรือแม่ สมมติให้โครโมโซมพ่อเป็นตัวแทนของโครงข่ายประสาทเทียมดังรูปที่ 4.11 ส่วนโครโมโซมแม่เป็นตัวแทนของโครงข่ายประสาทเทียมเป็นดังรูปที่ 4.12 ดังนั้นเซลล์ที่ไม่ใช่เซลล์อินพุตคือ 4 5 และ 6 สมมติให้ค่าน้ำหนักที่พุ่งเข้าเซลล์ที่ 5 และ 6 ของลูกถูกเลือกมาจากแม่ ส่วนค่าน้ำหนักที่พุ่งเข้าเซลล์ที่ 4 มาจากพ่อ จะได้โครโมโซมลูกเป็นตัวแทนของโครงข่ายประสาทเทียมดังรูปที่ 4.13

ส่วนการกลายพันธุ์ของโครโมโซมคือการที่ เซลล์ที่ไม่ใช่เซลล์อินพุตจะถูกเลือกมา  $n$  เซลล์ และค่าน้ำหนักที่พุ่งเข้าเซลล์นั้นแต่ละค่า จะถูกเพิ่มค่าด้วยค่าที่สุ่มมาจากค่าที่อยู่ในช่วง  $-1.0$  ถึง  $1.0$  เช่นสมมติให้โครโมโซมเป็นตัวแทนของโครงข่ายประสาทเทียมในรูปที่ 4.11 ถูกเลือกให้กลายพันธุ์ และเซลล์ที่ถูกเลือกคือเซลล์ที่ 5 โดยที่ค่าน้ำหนัก  $w_{51}$  ถูกเพิ่มด้วย  $-0.2$  ดังนั้น  $w_{51}$  จะกลายเป็น  $0.6$  และค่าน้ำหนัก  $w_{52}$  ถูกเพิ่มด้วย  $-0.8$  จึงกลายเป็น  $-0.9$  ส่วนค่าน้ำหนัก  $w_{53}$  ถูกเพิ่มด้วย  $0.2$  จึงกลายเป็น  $-0.1$  ดังนั้นโครโมโซมที่ถูกกลายพันธุ์จะมีลักษณะโครงข่ายประสาทเทียมดังรูปที่ 4.14 นั่นเอง



โดยสรุปแล้วอัลกอริทึมแบบพันธุกรรมนี้จะใช้การแทนคำตอบด้วยสายบิต (bit string representation) โดยส่วนมากใช้การคัดเลือกตามสัดส่วน และการข้ามสายพันธุ์ จะเป็นอุปกรณ์หลักในการสร้างรายบุคคลรายใหม่ ซึ่งจะแตกต่างจากโปรแกรมแบบพันธุกรรม (genetic programming) และการคำนวณเชิงวิวัฒนาการ (evolutionary computing)

## 4.2 บทนำโปรแกรมแบบพันธุกรรม (Introduction to Genetic Programming)

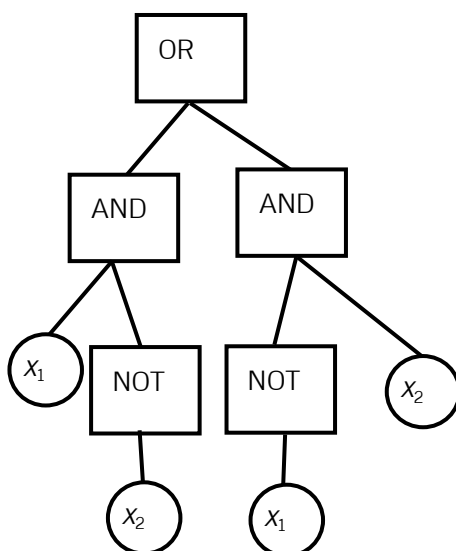
โปรแกรมแบบพันธุกรรม [Engelbrecht07] นี้ถูกมองว่าเป็นเคสพิเศษของอัลกอริทึมแบบพันธุกรรม นั่นคือโปรแกรมแบบพันธุกรรมจะมุ่งไปที่การวิวัฒนาการของจีโนไทป์ เพียงแต่การแทนคำตอบไม่ได้เป็นการแทนโดยโครโมโซมที่เป็นสายอักขระ (string) หรือเวกเตอร์ แต่เป็นการแทนด้วยทรี (tree) ซึ่งโปรแกรมแบบพันธุกรรมนี้ถูกคิดค้นครั้งแรกโดย John Koza [Koza92] สำหรับการวิวัฒนาการของโปรแกรมที่ทำงานได้ (executable program) โดยที่ในแต่ละรุ่น โปรแกรม (ในกรณีนี้เป็นรายบุคคล) จะถูกประเมินประสิทธิภาพจากการคำนวณหาคำตอบในโดเมนของปัญหา โดยที่คำตอบที่ได้จากการคำนวณจะถูกใช้เป็นค่าที่เหมาะสมของโปรแกรมนั้น (จะกล่าวโดยละเอียดต่อไป)

เนื่องจากในการสร้างลูกในแต่ละรุ่นจะคล้ายกับอัลกอริทึมแบบพันธุกรรม ดังนั้นในหัวข้อนี้จะอธิบายการทำงานของโปรแกรมแบบพันธุกรรม อย่างกระชับ และเนื่องจากในการทำโปรแกรมแบบพันธุกรรมนั้น 1 โครโมโซมหรือ 1 รายบุคคลในกรณีนี้ คือ โปรแกรมที่ทำงานได้ 1 โปรแกรม และจะถูกแทนด้วยโครงสร้างทรี ดังนั้นแต่ละรายบุคคลอาจจะมีรูปร่าง (ปัจจัยของกิ่งของแต่ละโหนดในทรี) และขนาด (ความลึกของทรี (tree depth)) ที่แตกต่างกันได้ และในแต่ละรุ่นก็อาจจะแตกต่างกันได้ ทั้งนี้แกรมมา (grammar) ที่สะท้อนถึงปัญหาและคำตอบ ต้องถูกกำหนดไว้ ซึ่งส่วนประกอบของแกรมมาคือ เซตปลายทาง (terminal set) จะประกอบไปด้วยตัวแปร (variable) และค่าคงที่, เซตฟังก์ชัน (function set) จะประกอบไปด้วยฟังก์ชันทางคณิตศาสตร์ และ/หรือ บูลีนฟังก์ชัน (Boolean function) ทั้งนี้โครงสร้างของการตัดสินใจ เช่น if-then-else และลูป (loop) ก็จะเป็นส่วนประกอบในเซตฟังก์ชันเช่นกัน และสุดท้าย กฎความหมาย (semantic rules) ก็ต้องถูกกำหนดไว้ด้วย

ในโครงสร้างทรีนี้ สมาชิกของเซตปลายทางจะอยู่ในตำแหน่งของโหนดใบ (leaf node) ในขณะที่สมาชิกของเซตฟังก์ชันจะอยู่ในตำแหน่งของโหนดที่ไม่ใช่ใบ (non-leaf node) และใน 1 รุ่น จะประกอบไปด้วยเซตของทรี ที่ถูกสร้างตามแกรมมาที่กำหนดไว้ และคำตอบของปัญหาจะเป็นทรี ที่ดีที่สุด นั่นเอง

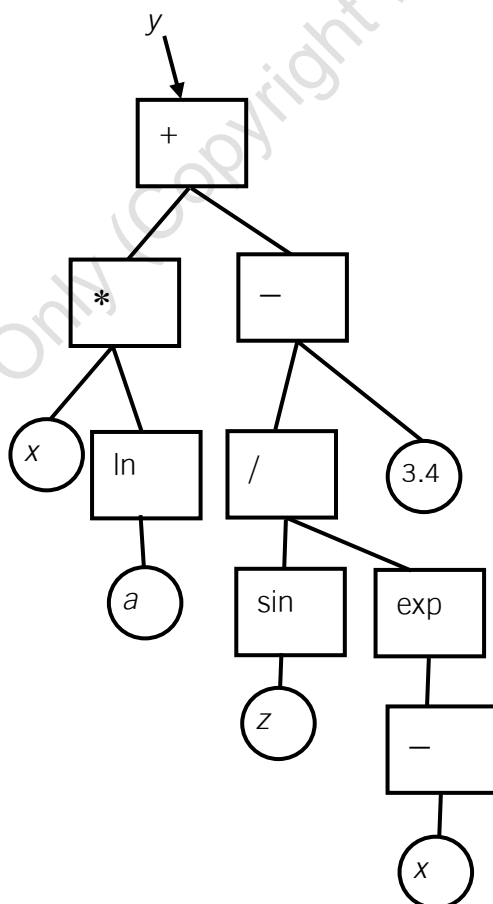
---

ตัวอย่างที่ 4.14 [Engelbrecht07] สมมติให้เซตฟังก์ชันเป็น  $\{\text{AND}, \text{OR}, \text{NOT}\}$  และเซตปลายทางเป็น  $\{x_1, x_2\}$  โดยที่  $x_1, x_2 \in \{0,1\}$  ดังนั้นสำหรับสมการบูลีน ของ XOR ซึ่งคือ  $(x_1 \text{ AND NOT } x_2) \text{ OR } (\text{NOT } x_1 \text{ AND } x_2)$  แสดงเป็นทรีได้ดังรูปที่ 4.15



รูปที่ 4.15 ทรีสำหรับ XOR

ตัวอย่างที่ 4.15 [Engelbrecht07] สมมติให้เซตฟังก์ชันเป็น  $\{-, +, *, /\}$  และเซตปลายทางเป็น  $\{a, x, z, 3.4\}$  โดยที่  $a, x, z \in \mathbb{R}$



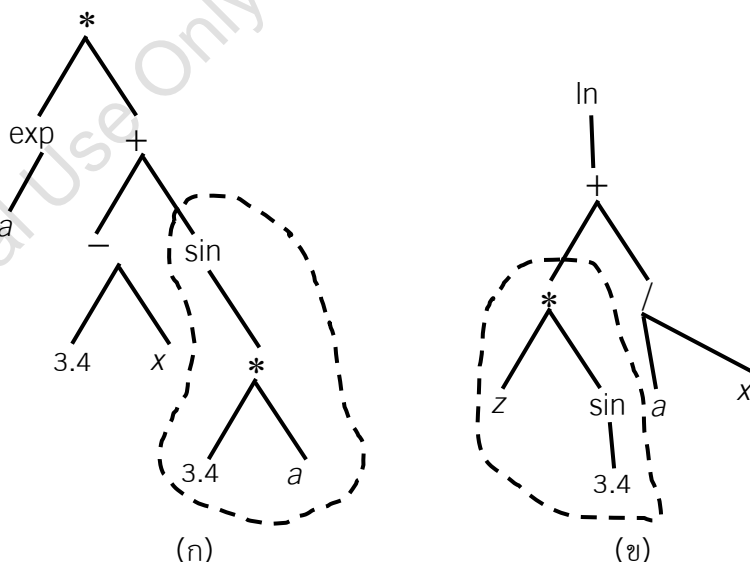
รูปที่ 4.16 ทรีสำหรับสมการที่ 4.60

สามารถสร้างทรีที่เป็นการแทนสมการ ที่ 4.60 ได้ดังรูปที่ 4.16

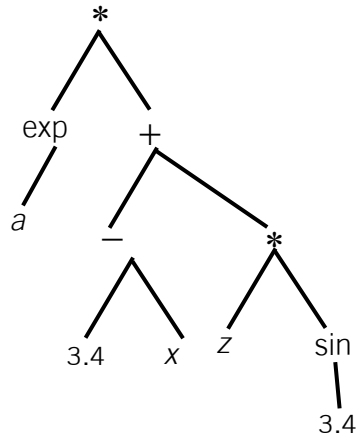
$$y = x * \ln(a) + \sin(z) / \exp(-x) - 3.4 \quad (4.60)$$

สำหรับกระบวนการตั้งประชากรเริ่มต้นนั้น [Engelbrecht07] ยังคงเป็นเช่นเดียวกับ อัลกอริทึมแบบพันธุกรรม นั่นคือ เป็นการสุ่มโดยอยู่ภายใต้ข้อจำกัดของความลึกของทรีที่มากที่สุด และความหมาย (semantic) ที่ถูกกำหนดในรูปของแกรมมา ดังนั้นสำหรับแต่ละรายบุคคล รากจะถูกเลือกแบบสุ่มจากสมาชิกของเซตฟังก์ชัน จำนวนลูกหรือกิ่งของรากนั้นจะถูกกำหนดจากชนิดของฟังก์ชันที่ถูกเลือกเป็นราก นั่นคือจำนวนกิ่งที่ออกจากโหนดฟังก์ชันจะเท่ากับจำนวนอาร์กิวเมนต์ของฟังก์ชันนั้น ส่วนโหนดที่ไม่ใช่รากนั้น จะถูกเลือกแบบสุ่มจากสมาชิกของเซตปลายทาง หรือเซตฟังก์ชัน และถ้าสมาชิกของเซตปลายทางถูกเลือก โหนดนั้นจะกลายเป็นโหนดใบทันที โดยปกติแล้วในการตั้งประชากรเริ่มต้นนั้น ทรีจะถูกสร้างอย่างง่าย และเมื่อผ่านกระบวนการวิวัฒนาการทรีจะโตและเพิ่มความซับซ้อนไปเอง

สำหรับการคำนวณฟังก์ชันความเหมาะสมนั้น [Engelbrecht07] ในโปรแกรมแบบพันธุกรรมจะขึ้นกับปัญหา เนื่องจากแต่ละรายบุคคลคือโปรแกรม ดังนั้นการคำนวณค่าความเหมาะสมจะเป็นการคำนวณจากเคสทดสอบ (test case) หรือเคสความเหมาะสม (fitness case) ซึ่งประสิทธิภาพของโปรแกรมที่ได้จากเคสทดสอบเหล่านั้น จะกลายเป็นค่าความเหมาะสมของโปรแกรมนั้นเอง เช่นจำนวนครั้งที่โปรแกรมคำนวณได้คำตอบที่ถูกจากเคสทดสอบ หรือในกรณีของการหาสมการที่เหมาะสมนั้น เคสทดสอบจะประกอบด้วย คู่อินพุตและเอาต์พุต หลายคู่ และทำการคำนวณหาค่าเอาต์พุตจากโปรแกรมเมื่อให้ค่าอินพุต และทำการเปรียบเทียบค่าเอาต์พุตที่ได้กับค่าเอาต์พุตจริง โดยใช้การคำนวณหาความผิดพลาด (error) และในท้ายที่สุดทำการหาค่าเฉลี่ยความผิดพลาดกำลังสอง (mean squared error (MSE)) ดังนั้นถ้า MSE น้อยแสดงว่าโปรแกรมนั้นดี ถ้าค่า MSE มากแสดงว่าโปรแกรมนั้นไม่ดี ซึ่งกรณีนี้อาจจะจำเป็นที่จะต้องใช้การสเกลที่เหมาะสมดังที่เคยกล่าวมาแล้วในหัวข้อที่แล้ว หรือค่าความเหมาะสมอาจจะเป็นจำนวนครั้งที่โปรแกรมชนะในการเล่นเกมสต่อจำนวนเกมที่เล่นทั้งหมด



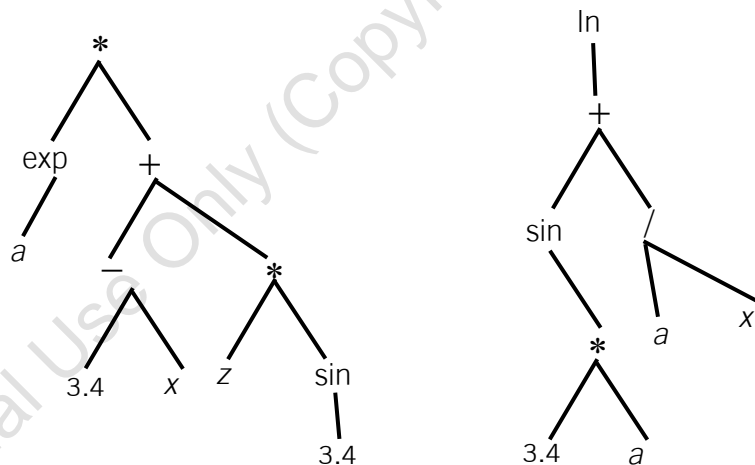




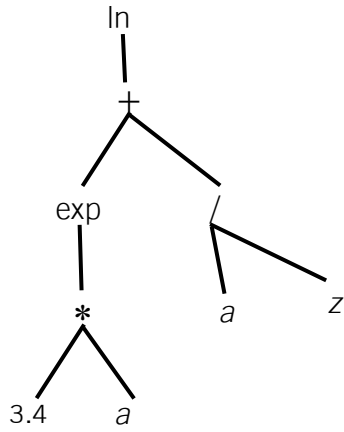
(ค)

รูปที่ 4.17 (ก) พ่อแม่ตัวที่ 1 (ข) พ่อแม่ตัวที่ 2 และ (ค) ลูกที่ได้ (ทรีส่วนย่อยที่อยู่ในกรอบเส้นประ คือทรีส่วนย่อยในพ่อแม่ตัวที่ 1 ที่จะถูกแทนส่วนทรีส่วนย่อยในพ่อแม่ตัวที่ 2)

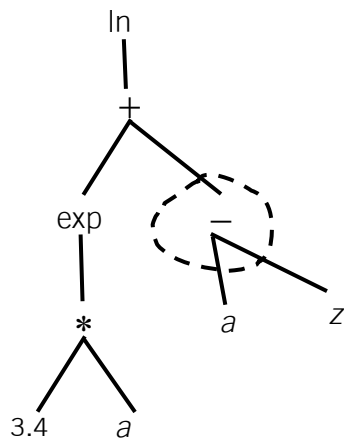
สำหรับการข้ามสายพันธุ์นั้น [Engelbrecht07] ก็เป็นเช่นเดียวกับที่เคยกล่าวมาแล้วในหัวข้อที่แล้ว เช่นในกรณีที่พ่อแม่ 1 คู่สร้างลูก (offspring) 1 ตัว สามารถทำได้ด้วยการสุ่มโหนดของพ่อแม่ แล้วกระบวนการข้ามสายพันธุ์นั้นทำได้โดยการสุ่มเลือก ทรีส่วนย่อย (subtree) ของพ่อแม่ตัวหนึ่ง และแทนที่ทรีส่วนย่อยนั้นจากพ่อแม่อีกตัวหนึ่ง ดังรูปที่ 4.17 (เพื่อความง่ายต่อไปนี้ในการแสดงรูปทรีที่เหลือ จะไม่มีการเขียนสัญลักษณ์วงกลมหรือสี่เหลี่ยม) ซึ่งจากรูปนี้ ทรีส่วนย่อย sin ของพ่อแม่ตัวที่ 1 ถูกแทนด้วยทรีส่วนย่อย \* ของพ่อแม่ตัวที่ 2 เพื่อสร้างลูก 1 ตัว



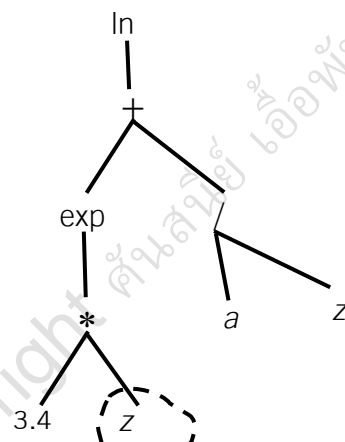
รูปที่ 4.18 ลูกที่ได้จากการข้ามสายพันธุ์ของพ่อแม่ในรูปที่ 4.17



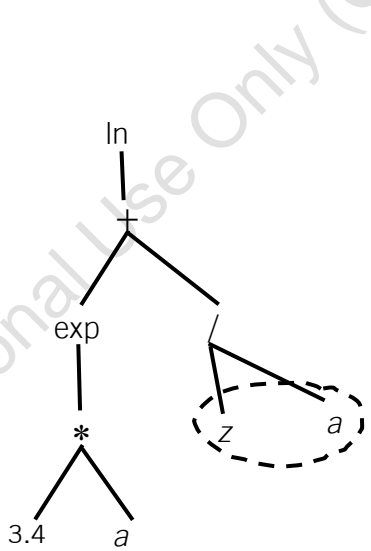
(ก)



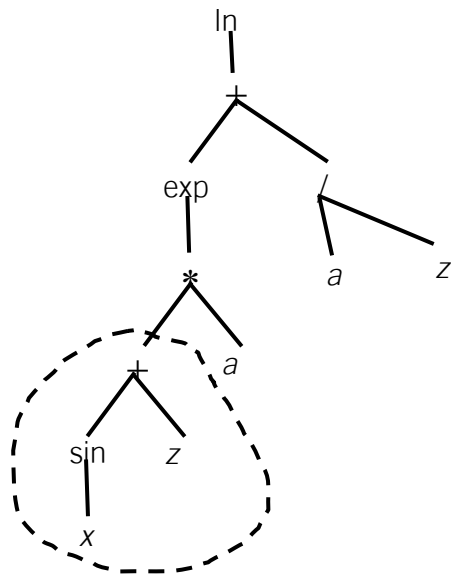
(ข)



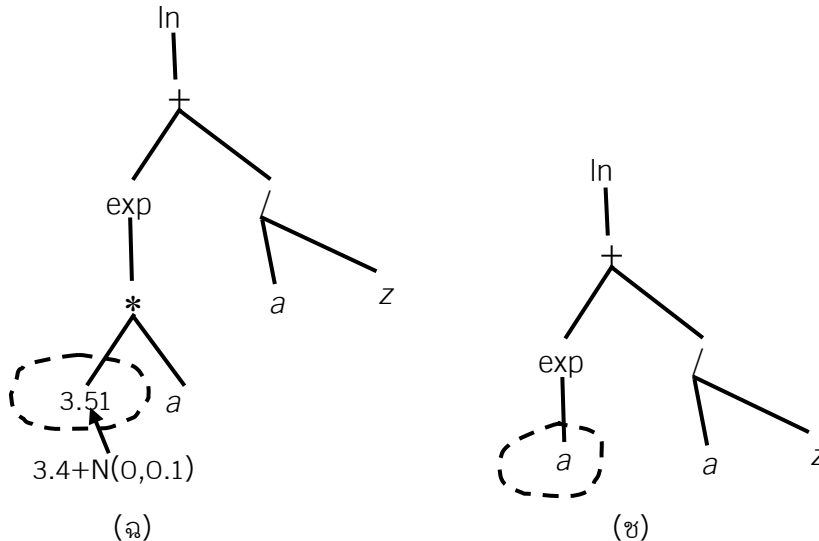
(ค)



(ง)



(จ)



รูปที่ 4.19 (ก) ทรีดั้งเดิม และทรีที่ถูกกลายพันธุ์ (ข) ที่โหนดฟังก์ชัน (ค) ที่โหนดปลายทาง (ง) สลับ (จ) ขยาย (ฉ) เกาส์เซียน และ (ช) ตัดปลาย (วงกลมที่เป็นเส้นประคือตำแหน่งที่เกิดการกลายพันธุ์)

แต่สำหรับการข้ามสายพันธุ์ที่เกิดจากพ่อแม่ 2 ตัวและได้ลูก 2 ตัวนั้นเป็นการสลับที่ทรีส่วนย่อยของพ่อแม่ทั้ง 2 โดยที่มีการสุ่มเลือกโหนด ดังรูปที่ 4.18 เป็นลูกที่เกิดจากพ่อแม่ในรูปที่ 4.17(ก) และ 4.17(ข) โดยที่โหนดที่ถูกเลือกคือโหนด sin ในพ่อแม่ตัวที่ 1 และ โหนด \* ในพ่อแม่ตัวที่สอง และลูกที่ได้เกิดจากการสลับทรีส่วนย่อยทั้งสอง นั้นเอง

สำหรับการกลายพันธุ์ [Engelbrecht07] ในโปรแกรมแบบพันธุกรรมนั้น ถึงแม้จะยังคงยึดหลักเดียวกันอัลกอริทึมแบบพันธุกรรม แต่ในกรณีนี้ก็มีลักษณะการกลายพันธุ์ได้หลายรูปแบบ ดังนี้

1. การกลายพันธุ์ที่โหนดฟังก์ชัน (Function node mutation) โหนดที่ไม่ใช่โหนดปลายทาง หรือเป็นโหนดฟังก์ชัน ที่ถูกเลือกแบบสุ่มนั้น จะถูกแทนที่ด้วยฟังก์ชันที่มีจำนวนกึ่งที่เท่ากัน ที่ถูกเลือกแบบสุ่ม ดังแสดงในรูปที่ 4.19(ข) ที่โหนด + ถูกแทนที่ด้วย -
2. การกลายพันธุ์ที่โหนดปลายทาง (Terminal node mutation) โหนดใบหรือโหนดปลายทาง จะถูกเลือกแบบสุ่ม และถูกแทนที่ด้วยสมาชิกจากเซตปลายทางที่ถูกเลือกมาแบบสุ่มเช่นกัน ดังแสดงในรูปที่ 4.19(ค) ที่โหนดที่มีค่า a ถูกเปลี่ยนเป็น z
3. การกลายพันธุ์สลับ (Swap mutation) ทำการเลือกโหนดฟังก์ชันแบบสุ่ม และอาร์กิวเมนต์ของโหนดนั้น จะมีการสลับกัน ดังรูปที่ 4.19(ง) ที่อาร์กิวเมนต์ของโหนด / มีการสลับค่ากัน
4. การกลายพันธุ์ขยาย (Grow mutation) โหนดจะถูกเลือกแบบสุ่ม และโหนดนั้นจะถูกแทนที่ด้วยทรีส่วนย่อยที่สร้างขึ้นใหม่แบบสุ่ม ดังรูปที่ 4.19(จ) ที่โหนดที่ 3.4 ถูกแทนที่ด้วยทรีส่วนย่อย
5. การกลายพันธุ์เกาส์เซียน (Gaussian mutation) โหนดปลายทางที่เป็นค่าคงที่ที่จะถูกเลือกมาแบบสุ่ม และจะถูกเพิ่มค่าด้วยค่าที่สุ่มมาแบบเกาส์เซียน ดังแสดงในรูปที่ 4.19(ฉ) ที่โหนด 3.4 ถูกเปลี่ยนเป็น  $3.4+N(0,0.1)$

6. การกลายพันธุ์ตัดปลาย (Trunc mutation) โหนดฟังก์ชันจะถูกเลือกแบบสุ่ม และถูกแทนที่ด้วยโหนดปลายทางที่ถูกเลือกมาแบบสุ่ม ดังแสดงในรูปที่ 4.19(ข) ที่โหนด + ถูกแทนที่ด้วยค่า  $a$

เพื่อให้เป็นภาพในการทำงานของโปรแกรมแบบพันธุกรรมชัดขึ้น จะขออธิบายตามตัวอย่างอัลกอริทึมของ John Koza [Koza92, Mitchell98] โดยสมมุติให้ต้องการหาฟังก์ชันที่รับค่าอินพุต 1 ตัว ( $A$ ) โดยที่มีค่าเอาต์พุต 1 ตัว ( $P$ ) ซึ่งในความเป็นจริงแล้วฟังก์ชันที่ต้องการหาคือ

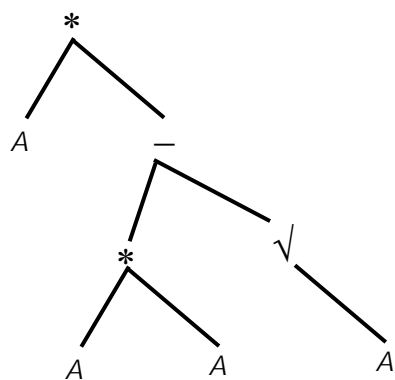
$$P^2 = cA^3 \quad (4.61)$$

ตารางที่ 4.9 เสนอทดสอบ

Input ( $A$ )	0.72	1.00	1.52	5.20	9.53	19.1
Desired Output ( $P$ )	0.61	1.00	1.84	11.90	29.40	83.5

ดังนั้นเสนอทดสอบเป็นดังตารางที่ 4.9 และอัลกอริทึมของ John Koza เป็นดังนี้

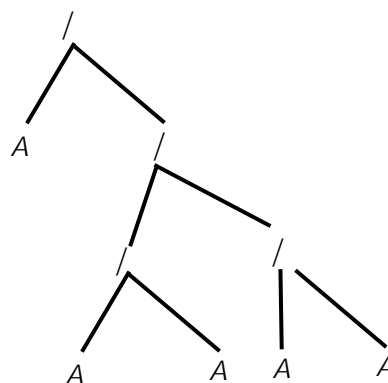
1. เลือกเซตฟังก์ชัน และเซตปลายทาง ที่เป็นไปได้สำหรับโปรแกรม ซึ่งผู้ใช้งานจำเป็นที่จะต้องทำการเดาอย่างชาญฉลาด เนื่องจากข้อมูลในส่วนนี้ไม่ได้มีให้ล่วงหน้า สมมุติให้เซตฟังก์ชันและเซตปลายทางที่ผู้ใช้เลือกคือ  $\{+, -, *, /, \sqrt{\phantom{x}}\}$  และ  $\{A\}$  ตามลำดับ
2. ทำการตั้งประชากรเริ่มต้นโดยการสุ่มสร้างทรี โดยใช้สมาชิกจากเซตฟังก์ชัน และเซตปลายทาง ซึ่งทรีที่สร้างนี้ต้องถูกต้องตามวากยสัมพันธ์ (syntax) ดังที่เคยกล่าวมาแล้ว และถึงแม้ว่าความลึกของทรีจะเป็นเท่าไรก็ได้ แต่สำหรับ John Koza จะกำหนดความลึกที่มากที่สุดไว้สำหรับขั้นตอนนี้ สมมุติให้ทรีที่ถูกสร้างมี 3 ทรี ดังรูปที่ 4.20
3. ทำการคำนวณหาค่าความเหมาะสมของแต่ละทรี โดยคำนวณค่าเอาต์พุตที่ได้จากแต่ละเซลล์ในเซลล์ทดสอบ และค่าความเหมาะสมจะเป็นค่าที่ได้จากฟังก์ชันที่เป็นฟังก์ชันของจำนวนเซลล์ที่ได้ผลลัพธ์ถูกต้อง ตัวอย่างเช่นจำนวนของเอาต์พุตที่ได้อยู่ในช่วง 20% ของค่าเอาต์พุตที่ต้องการ ซึ่งค่าความเหมาะสมของแต่ละทรีแสดงในรูปที่ 4.20
4. ทำการคัดเลือก การข้ามสายพันธุ์ ละการกลายพันธุ์ ซึ่งในอัลกอริทึมของ John Koza นั้นกำหนดไว้ว่า 10% ของทรีจะถูกคัดลอกไปยังประชากรรุ่นใหม่โดยไม่มีการเปลี่ยนแปลง และที่เหลือ 90% จะผ่านกระบวนการคัดเลือกและการข้ามสายพันธุ์ ซึ่งในการข้ามสายพันธุ์ในกรณีนี้ เป็นการสุ่มเลือก โหนดจากพ่อแม่แต่ละตัว และทำการสลับทรีส่วนย่อยของโหนดทั้งสอง ดังแสดงในรูปที่ 4.21 ส่วนการกลายพันธุ์ จะคล้ายกับการกลายพันธุ์ขยายเพียงแต่ถ้ามีโหนดที่ถูกเลือกแล้ว ทรีส่วนย่อยที่อยู่ใต้โหนดนั้นจะถูกเปลี่ยนเป็นทรีส่วนย่อยที่ถูกสร้างขึ้นใหม่แบบสุ่ม ดังแสดงในรูปที่ 4.22 ซึ่งเป็นกลายพันธุ์ของรูปที่ 4.20(ก) ที่ทรีส่วนย่อยใต้โหนด — แต่ในอัลกอริทึมของ Koza ไม่มีการใช้การกลายพันธุ์



$$A * [(A * A) - \sqrt{A}]$$

Fitness value = 1

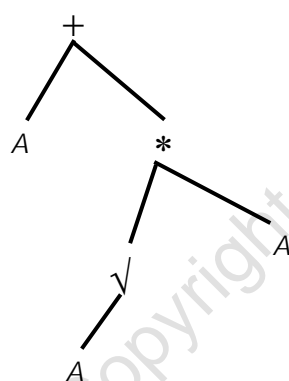
(ก)



$$A / [(A / A) / (A / A)]$$

Fitness value = 3

(ข)

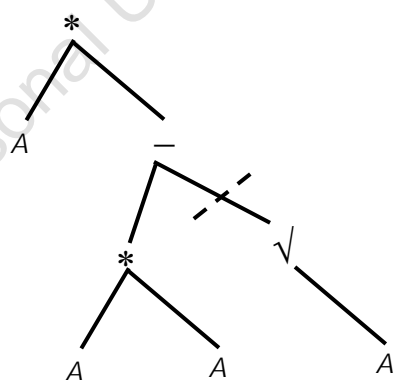


$$A + [\sqrt{A * A}]$$

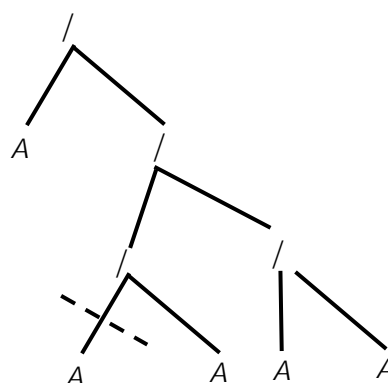
Fitness value = 0

(ค)

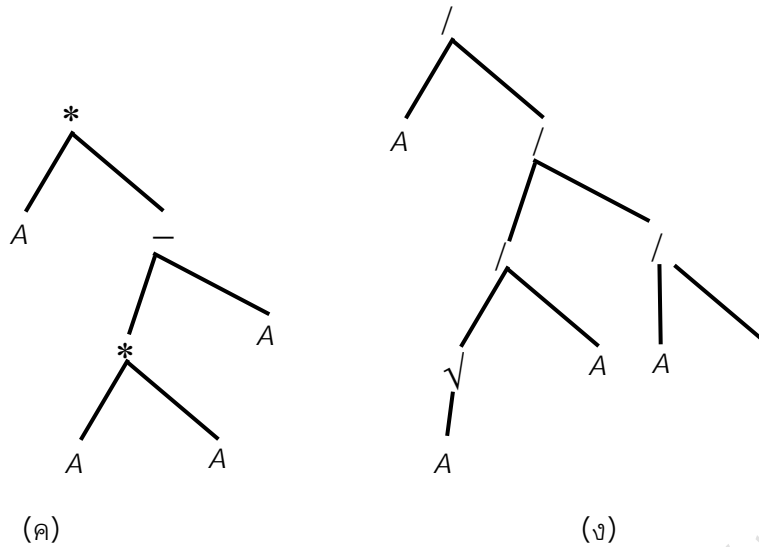
รูปที่ 4.20 ทรีสำหรับรายบุคคลที่ (ก) 1 (ข) 2 และ (ค) 3



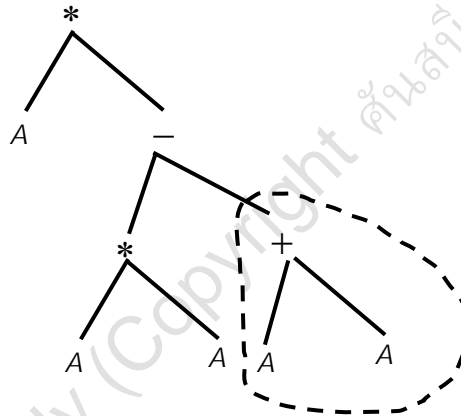
(ก)



(ข)



รูปที่ 4.21 การข้ามสายพันธุ์จากพ่อแม่ตัวที่ (ก) 1 และ (ข) 2 (ตำแหน่งการข้ามสายพันธุ์อยู่ที่เส้นประ) สร้างลูกตัวที่ (ค) 1 และ (ง) 2



รูปที่ 4.22 การกลายพันธุ์ที่ได้เหนด — ของรูปที่ 4.20(ก) (วงกลมเส้นประคือทรีส่วนย่อยที่ถูกสร้างใหม่)

### 4.3 บทนำการคำนวณเชิงวิวัฒนาการ (Introduction to Evolutionary Computing)

การคำนวณเชิงพันธุกรรม [Engelbrecht07] แตกต่างจากอัลกอริทึมแบบพันธุกรรมและโปรแกรมแบบพันธุกรรมอย่างมาก เนื่องจากการคำนวณเชิงพันธุกรรมจะเน้นไปที่การพัฒนาแบบจำลองพฤติกรรม (behavior model) ไม่ใช่แบบจำลองพันธุกรรม (genetic model) ซึ่งการคำนวณเชิงพันธุกรรมจะคำนึงถึงการวิวัฒนาการของฟีโนไทป์ (phenotypic evolution) ดังนั้นเป้าหมายหลักจึงเป็นการหาเซตของพฤติกรรมที่เหมาะสมที่สุดจากปริภูมิของพฤติกรรมที่สังเกตได้ (space of observable behavior) โดยที่การหาค่าที่เหมาะสมจะเป็นการหา “ความผิดพลาดทางพฤติกรรม (behavior error) ของแต่ละรายบุคคล และโดยปกติแล้วการคำนวณเชิงพันธุกรรมไม่มีการข้ามสายพันธุ์ มีเพียงการกลายพันธุ์เท่านั้น เนื่องจากกระบวนการทุกอย่างคล้ายกับอัลกอริทึมเชิงพันธุกรรม ในหัวข้อจะทำการอธิบายกระบวนการโดยใช้ตัวอย่างที่ 4.16 แทน

ตัวอย่างที่ 4.16 สมมุติให้ในการเล่นเกมสโ-เอ็กซ์นั้น [Fogel00a] ผู้เล่นที่สร้างจากการคำนวณเชิงวิวัฒนาการเริ่มเล่นก่อน และทำการวางเครื่องหมาย X โดยที่ผู้เล่นนี้จะทำการเล่นแข่งกับผู้เล่นที่สร้างจากฐานกฎ (rule-based) ซึ่งผู้เล่นฐานกฎจะทำการวางเครื่องหมาย O ทั้งนี้เมื่อผู้เล่นที่สร้างจากการคำนวณเชิงวิวัฒนาการเล่นก่อน ทำให้ผู้เล่นฐานกฎสามารถเลือกเล่นได้ 8 ความเคลื่อนไหวที่เป็นไปได้ และฐานกฎที่ใช้เป็นดังนี้

1. จากอาเรย์ของทุกความเคลื่อนไหวที่เป็นไปได้ เลือกการเคลื่อนไหวที่ยังไม่ได้เล่น
2. สำหรับการเคลื่อนไหวที่ตามมา (subsequent move)
  - a. ด้วยโอกาส 10% ให้เคลื่อนไหวแบบสุ่ม มิฉะนั้น
  - b. ถ้ามีช่องที่จะชนะ ให้วางเครื่องหมายในช่องนั้น มิฉะนั้น
  - c. ถ้ามีความเป็นไปได้ในการป้องกัน ให้วางเครื่องหมายที่ช่องนั้น มิฉะนั้น
  - d. ถ้ามีช่องว่าง 2 ช่องที่อยู่ในเส้นที่มีเครื่องหมาย O อยู่ ให้วางเครื่องหมายในช่องใดช่องหนึ่งแบบสุ่ม มิฉะนั้น
  - e. สุ่มวางเครื่องหมายในช่องที่ว่าง
3. ทำขั้นที่ 2 ไปจนกว่าเกมจะจบ
4. ทำขั้นที่ 1 ไปจนกว่าเกมจะจบครบ 8 ครั้งที่เป็นไปได้ สำหรับการเคลื่อนไหวลำดับที่ 2

สำหรับผู้เล่นที่สร้างจากการคำนวณเชิงวิวัฒนาการนั้น สร้างจากเปอร์เซปตรอนป้อนค่าไปข้างหน้าหลายชั้น (multi-layered feedforward perceptrons) ที่รับค่าอินพุตจากลักษณะของกระดานขณะนั้น และให้ค่าเอาต์พุตเป็นการเคลื่อนไหวของผู้เล่นนี้ ดังนั้นอินพุต และเอาต์พุตจะมี 9 ค่าเสมอ โดยที่ตำแหน่งที่มีเครื่องหมาย X จะมีค่าเป็น 1.0 และถ้ามีเครื่องหมายเป็น O จะมีค่าเป็น -1.0 ส่วนช่องที่ว่างจะมีค่าเป็น 0.0 และถ้าโหนดเอาต์พุตใดให้ค่ามากที่สุดผู้เล่นนี้จะเคลื่อนไปทางช่องนั้น ทั้งนี้โหนดเอาต์พุตที่เป็นช่องที่มีเครื่องหมายอยู่แล้วจะไม่ได้รับการพิจารณาเลย ซึ่งในกระบวนการวิวัฒนาการนั้นจะไม่ได้ใช้ความดันการคัดเลือก ในการทำให้ค่าเอาต์พุตของโหนดเหล่านั้นเป็น 0 เลย

ในตัวอย่างนี้เปอร์เซปตรอนหลายชั้นนี้มี 1 ชั้นซ่อน ที่มีจำนวนโหนดซ่อน (hidden nodes) ตั้งแต่ 1 ถึง 10 ดังนั้นเปอร์เซปตรอนหลายชั้นนี้คือ 9-?-9 และฟังก์ชันกระตุ้นในกรณีนี้คือ

$$f(x) = \frac{1}{(1 + e^{-x})} \quad (4.62)$$

สำหรับผู้เล่นจากการคำนวณเชิงวิวัฒนาการนี้ ประชากรเริ่มต้นมี 50 โครงข่าย โดยที่จำนวนโหนดซ่อนของแต่ละโครงข่ายจะถูกเลือกแบบสุ่มจาก  $[1, 2, \dots, 10]$  และค่าน้ำหนักและไบแอสที่เชื่อมกับแต่ละโหนดในโครงข่ายจะถูกเลือกแบบสุ่มจากการกระจายเอกรูปในช่วง

$[-0.5, 0.5]$  ทั้งนี้ลูก 1 ตัวจะถูกคัดลอกจากแต่ละพ่อแม่ และมีการกลายพันธุ์เกิดขึ้น 2 รูปแบบคือ

1. ค่าน้ำหนักและไบแอสทุกค่าในโครงข่ายจะถูกบวกเพิ่มด้วยค่าที่ถูกสุ่มแบบเกาส์เซียนที่มีค่าเฉลี่ยเป็น 0 และส่วนเบี่ยงเบนมาตรฐานเป็น 0.05
2. ด้วยความน่าจะเป็น 0.5 จำนวนโหนดในชั้นซ่อนจะถูกเปลี่ยน ในการเปลี่ยนนี้อาจจะเป็นการเพิ่มหรือการลดจำนวนโหนดด้วยความน่าจะเป็นที่เท่ากัน (equal likelihood) ทั้งนี้

จำนวนต้องอยู่ในช่วง  $[1, 2, \dots, 10]$  เสมอ และถ้าเป็นการเพิ่มจำนวนโหนด โหนดที่เพิ่มจะมี  
ค่าน้ำหนักและไบแอสเท่ากับ 0.0

สำหรับการหาค่าความเหมาะสมนั้น คำนวณจาก 4 เซตของ 8 เกมส์ โดยมีฟังก์ชันผลที่  
ตามมา (payoff function) เป็น  $\{+1, -1, 0\}$  หรือ  $\{+1, -10, 0\}$  หรือ  $\{+10, -1, 0\}$  โดยที่ ค่า  
บวกหมายถึงชนะ ค่าลบหมายถึงแพ้ และ 0 หมายถึงเสมอ ดังนั้นสำหรับฟังก์ชันผลที่ตามมา 2  
อันดับแรกนั้น คะแนนที่มากที่สุดที่เป็นไปได้คือ 32 ในขณะที่คะแนนที่มากที่สุดที่เป็นไปได้ของ  
ฟังก์ชันผลที่ตามมาอันดับสุดท้ายคือ 320 นั่นเอง หลังจากทีแต่ละโครงข่ายเล่นเกมส์ครบ 32 ครั้ง  
แล้ว แต่ละโครงข่ายจะถูกเปรียบเทียบกับคะแนนที่ได้กับอีก 10 โครงข่ายที่เลือกมาแบบสุ่ม ถ้า  
โครงข่ายนี้มีคะแนนสูงกว่าหรือเท่ากับโครงข่ายที่นำมาเปรียบเทียบ โครงข่ายนั้นจะชนะ และ  
โครงข่ายที่มีจำนวนครั้งที่ชนะมากที่สุด จะได้เป็นพ่อแม่ของรุ่นถัดไป ทั้งนี้ในการเล่นเกมนั้นจะเป็น  
การทดลองเล่นทั้งหมด 30 ครั้ง สำหรับแต่ละฟังก์ชันผลที่ตามมา และกระบวนการคำนวณเชิง  
วิวัฒนาการจะหยุดที่ 800 รุ่นของแต่ละครั้งการทดลองเล่น

---

โดยสรุปแล้วอัลกอริทึมสำหรับการคำนวณเชิงวิวัฒนาการเป็นดังนี้

1. ตั้งค่ารุ่น  $t=0$
2. ตั้งค่าตัวแปรต่างๆ
3. สร้างประชากรรุ่นแรก  $C(0)$  ด้วย  $n_s$  รายบุคคล
4. For แต่ละรายบุคคล  $x_i(t) \in C(t)$   
    คำนวณหาค่าความเหมาะสม  $f(x_i(t))$   
    End
5. While เชื้อนไขการหยุดไม่เป็นจริง
  - a. For แต่ละรายบุคคล  $x_i(t) \in C(t)$   
        สร้างลูก  $x'_i(t)$  โดยการใช้การกลายพันธุ์  
        คำนวณหาค่าความเหมาะสม  $f(x'_i(t))$   
        เพิ่ม  $x'_i(t)$  ในเซตของลูก  $C'(t)$   
        End
  - b. เลือกประชากรใหม่  $C(t+1)$  จาก  $C(t) \cup C'(t)$  โดยใช้ตัวดำเนินการการคัดเลือก
  - c.  $t = t+1$End

---

#### คำถามท้ายบทที่ 4

4.1 ให้เขียนโปรแกรมสำหรับการฝึกสอนเปอร์เซปตรอนหลายชั้น โดยใช้อัลกอริทึมแบบพันธุกรรม  
โดยที่สามารถเลือกใช้ตัวดำเนินการใดก็ได้ อย่างเหมาะสม



4.2 ให้อธิบายว่าถ้าต้องการสร้างโปรแกรมแบบพันธกรรมในการควบคุมหุ่นยนต์นั้นจะสามารถทำได้  
ได้อย่างไร โดยที่มีจุดประสงค์คือให้หุ่นยนต์สามารถเคลื่อนที่ออกจากห้องทางประตู ได้โดยที่ไม่ชน  
กับสิ่งกีดขวางที่วางอยู่ในห้อง

4.3 ให้เขียนโปรแกรมตามตัวอย่างที่ 4.16 และทำการจำลองการเล่นเกมสโ-เอ็กซ์นี้

---

# บทนำความฉลาดเชิงกลุ่ม

## Introduction to Swarm Intelligence

### บทที่ 5

สมมุติว่ากลุ่มคนกลุ่มหนึ่งที่เป็นเพื่อนกันออกตามหาที่ซ่อนสมบัติ ซึ่งมีอยู่ 1 คนที่รู้คร่าวๆ ว่า ที่ซ่อนสมบัตินั้นอยู่ประมาณที่ตรงไหน และทุกคนในกลุ่มตกลงกันว่าไม่ว่าใครจะเป็นตนเจอสมบัติก็ตาม สมบัติที่ได้มาจะถูกแบ่งให้ทุกคนในกลุ่ม ด้วยการที่คนที่เจอจะได้ส่วนแบ่งมากที่สุด และคนที่อยู่ห่างไปตามระยะทางจะได้ส่วนแบ่งน้อยลงไปตามสัดส่วนของระยะทาง ทั้งนี้ในการออกหาสมบัตินี้ ทุกคนจะมีอุปกรณ์ในการหาสมบัติเหมือนกัน และทุกคนสามารถส่งความแรงของสัญญาณ (signal strength) ของการตรวจเจอสมบัติ และตำแหน่งของตนเองให้กับเพื่อนที่อยู่ใกล้ที่สุดได้ (nearest neighbor) ดังนั้นแต่ละคนจะทราบว่าเป็นเพื่อนของตัวเองอยู่ใกล้สมบัติมากกว่าคนอื่นหรือไม่ ซึ่งในสถานการณ์แบบนี้ ถ้าแต่ละคน ไม่ช่วยเหลือกัน ต่างคนต่างหา จะมีคนที่ได้สมบัติเพียงคนเดียว แต่ถ้าทุกคนช่วยกันหาโดยอาศัยข้อมูลจากเพื่อน จะทำให้หาสมบัติได้ง่ายขึ้น และแต่ละคนจะเคลื่อนที่เข้าหาคนที่มีความแรงของสัญญาณของการตรวจเจอสมบัติ ซึ่งการทำเช่นนี้จะทำให้โอกาสที่จะเจอสมบัติมีมากขึ้น หรือเป็นการเพิ่มส่วนแบ่งของตนเอง [Engelbrecht07] ซึ่งที่กล่าวมาข้างต้นเป็นตัวอย่างของการทำงานร่วมกัน (cooperation) ในสถานการณ์ที่ไม่มีความรู้โดยรวม (global knowledge) ของสถานะแวดล้อม และแต่ละรายบุคคลมีการสื่อสารระหว่างกัน (interact) ในการหาคำตอบของจุดมุ่งหมายโดยรวม (global objective) โดยการแลกเปลี่ยนข้อมูลเฉพาะที่ (local information) ซึ่งภายหลังจะแพร่ขยายไปทั้งกลุ่ม

คำว่า “กลุ่ม (swarm)” [Engelbrecht07] ในที่นี้หมายถึงกลุ่มของเอเจนต์ (agent) (โดยปกติสามารถเคลื่อนที่ได้ (mobile)) ที่สามารถติดต่อสื่อสารกับเอเจนต์อื่นในกลุ่มได้ไม่ว่าจะเป็นทางตรง (direct) หรือทางอ้อม (indirect) โดยการกระทำต่อสิ่งแวดล้อมเฉพาะที่ ซึ่งการติดต่อสื่อสารระหว่างเอเจนต์นี้ทำให้เกิดกลยุทธ์การแก้ปัญหาแบบรวมแจกแจง (distributive collective problem-solving strategy) ดังนั้น ความฉลาดเชิงกลุ่ม (swarm intelligence (SI)) หมายถึง พฤติกรรมการแก้ปัญหาที่มาจากการติดต่อสื่อสารระหว่างเอเจนต์ และการคำนวณความฉลาดเชิงกลุ่ม (computational swarm intelligence (CSI)) [Engelbrecht07] หมายถึง แบบจำลองอัลกอริทึมที่เป็นไปตามพฤติกรรมนี้ ดังนั้นโดยปกติแล้วความฉลาดเชิงกลุ่มจะเป็นระบบที่เป็นพฤติกรรมร่วม (collective behavior) ของเอเจนต์ที่ไม่ซับซ้อน (unsophisticated agent) ที่ติดต่อสื่อสารกันเฉพาะที่กับสิ่งแวดล้อมของตนเอง แล้วทำให้เกิดรูปแบบฟังก์ชันโดยรวมที่สอดคล้องกัน (coherent functional global pattern) ในบางครั้งความฉลาดเชิงกลุ่มอาจถูกเรียกว่าความฉลาดร่วม (collective intelligence)

ระบบกลุ่มชีวภาพ (biological swarm system) ที่ถูกนำมาใช้ในความฉลาดเชิงกลุ่ม มีด้วยกันหลายประเภท เช่น มด ปลวก ผึ้ง แมงมุม ผีเสื้อ และ ผีเสื้อเป็นต้น ซึ่งสัตว์เหล่านี้มีโครงสร้างที่ง่าย แต่พฤติกรรมร่วมจะซับซ้อน และพฤติกรรมร่วมที่ซับซ้อนนี้เป็นผลมาจากรูปแบบของการติดต่อสื่อสารระหว่างรายบุคคลในกลุ่มบนช่วงเวลาหนึ่ง ซึ่งพฤติกรรมร่วมที่ซับซ้อนนี้ ไม่ใช่คุณสมบัติของรายบุคคลใดบุคคลหนึ่ง และไม่สามารถคาดเดาหรือสรุปจากพฤติกรรมของรายบุคคลได้ง่ายๆ ซึ่งพฤติกรรมนี้ถูกเรียกว่าการอุบัติ (emergence) หรืออาจหมายถึง กระบวนการของการอนุพัทธ์ (derive) โครงสร้างใหม่หรือโครงสร้าง รูปแบบ และคุณสมบัติ (พฤติกรรม) ร่วมในระบบที่ซับซ้อน ซึ่งโครงสร้างที่ได้นี้ไม่ได้ถูกสร้างจากระบบควบคุมที่ถูก

ประสานงาน (coordinated control system) แต่เป็นการอุปมาจากการติดต่อสื่อสารระหว่างรายบุคคลกับสิ่งแวดล้อมของตนเอง [Engelbrecht07]

พฤติกรรมร่วมของกลุ่มของสิ่งมีชีวิตสังคม เกิดจากพฤติกรรมของรายบุคคลในสังคมที่ไม่เป็นเชิงเส้น ซึ่งรายบุคคลและพฤติกรรมร่วมจะมีความสัมพันธ์ที่แนบแน่น นั่นคือพฤติกรรมร่วมของรายบุคคลจะกำหนดพฤติกรรมกลุ่ม ในขณะที่เดียวกันพฤติกรรมกลุ่มจะมีอิทธิพลต่อเงื่อนไขที่แต่ละรายบุคคลจะกระทำการใดๆ การกระทำเหล่านี้อาจเปลี่ยนสิ่งแวดล้อม และทำให้พฤติกรรมของรายบุคคลและเพื่อนบ้าน (neighbor) เปลี่ยนด้วย และจะทำให้พฤติกรรมร่วมเปลี่ยนตามมา การติดต่อสื่อสารของรายบุคคลช่วยให้ความรู้จากประสบการณ์เกี่ยวกับสิ่งแวดล้อมดีขึ้น ซึ่งการติดต่อสื่อสารเป็นได้ทั้งทางตรง โดยผ่านทาง การติดต่อทางร่างกาย หรือการมองเห็น การได้ยิน หรือการได้รับสารเคมี เป็นต้น และการติดต่อสื่อสารอาจเป็นทางอ้อม โดยผ่านทาง การเปลี่ยนแปลงสิ่งแวดล้อมเฉพาะที่ก็เป็นได้ และคำว่าสติกเมอจี (stigmergy) หมายถึงการติดต่อสื่อสารทางอ้อมระหว่างรายบุคคลนั่นเอง [Engelbrecht07]

ดังนั้นสิ่งที่สำคัญในความฉลาดเชิงกลุ่มคือ การติดต่อสื่อสาร และการทำงานร่วมกัน ซึ่งเป็นการทำงานร่วมกันที่ไม่มีใครเป็นหัวหน้า ไม่มีใครออกคำสั่งที่แต่ละรายบุคคลต้องทำตามนั่นเอง ทั้งนี้ในกระบวนการสร้างโครงสร้างเหล่านี้อยู่บนพื้นฐานหลัก 2 ประการคือ [Bonabeau99]

1. การจัดระเบียบตนเอง (self-organization) คือเซตของกลไกพลวัต (dynamic mechanisms) ที่โครงสร้างในระดับรวมของระบบนั้นเกิดจากการติดต่อสื่อสารระหว่างส่วนประกอบในระดับล่าง ซึ่งการติดต่อสื่อสารนี้เกิดจากข้อมูลเฉพาะที่อย่างเดียว ทั้งนี้การจัดระเบียบตนเองมีส่วนประกอบพื้นฐาน 4 ข้อ ดังนี้
  - ป้อนกลับแบบบวก (positive feedback) เป็นพฤติกรรมง่ายๆ ที่ถูกนำไปใช้ในการส่งเสริมการสร้างโครงสร้าง ซึ่งพฤติกรรมนี้รวมถึงการระดมคน (recruitment) และการเสริมกำลัง (reinforcement) เช่นการระดมคนไปยังแหล่งอาหาร ซึ่งการระดมคนในลักษณะนี้จะขึ้นอยู่กับฟีโรโมน (pheromone) ที่แต่ละรายบุคคลปล่อยไว้ตามทางที่ตนเอง ถ้าเป็นเส้นทางที่เดินไปยังแหล่งอาหาร จะมีการสะสมฟีโรโมนมากกว่าเส้นทางอื่น และคนที่มาเพิ่มจะเดินทางเส้นทางที่มีปริมาณฟีโรโมนมากกว่า เส้นทางอื่น
  - ป้อนกลับแบบลบ (negative feedback) เป็นพฤติกรรมในฉันทตรงข้ามกับป้อนกลับแบบบวก และเป็นพฤติกรรมที่ช่วยให้การรวบรวมรูปแบบเสถียรขึ้น ซึ่งพฤติกรรมนี้อาจจะมาในรูปของ ความอิ่มตัว (saturation), ความหมดแรง (exhaustion) หรือการแข่งขัน (competition) เช่นในการออกหาอาหาร (foraging) พฤติกรรมป้อนกลับที่เกิดจากจำนวนคนที่ออกหาอาหารที่ว่างที่จำกัดทำให้เกิดความพึงพอใจ (satiation), การหมดแรงในการหาแหล่งอาหาร, มีรายบุคคลมากที่แหล่งอาหาร หรือเกิดการแข่งขันระหว่างแหล่งอาหาร เป็นต้น
  - การจัดระเบียบตนเองจะขึ้นอยู่กับ การขึ้นลงแบบอิมิตัว (amplification of fluctuation) เช่นการเดินแบบสุ่ม (random walk), ความผิดพลาด (errors) การสลับหน้าที่แบบสุ่ม (random task-switching) เป็นต้น ซึ่งโครงสร้างของกลุ่มไม่ได้เกิดจากความสุ่ม (randomness) เท่านั้น แต่ความสุ่มนี้ยังเป็นส่วน

สำคัญ เนื่องจากทำให้การค้นหาคำตอบใหม่เกิดขึ้นได้ และการขึ้นลงนี้ยังทำตัวเองเป็นเมล็ดที่โครงสร้างเริ่มและขยาย

- ทุกอัลกอริทึมในการจัดระเบียบตนเองจะขึ้นกับการติดต่อสื่อสารหลายครั้ง ทั้งนี้รายบุคคลสามารถสร้างโครงสร้างของการจัดระเบียบตนเองได้ เช่น ความเสถียรของทางที่ได้จากช่วงชีวิตของฟีโรโมนก็เพียงพอแล้ว เนื่องจากการเดินทางเป็นการติดต่อสื่อสารทางการวางฟีโรโมน นั่นเอง

2. การแบ่งหน้าที่ (division of labor) ซึ่งขึ้นกับกลไกที่ความร่วมมือถูกกำหนดทางพันธุกรรม เช่นรายบุคคลจะมีลักษณะกายวิภาค ที่ต่างกัน เช่นชนิดของมดที่มีการแบ่งหน้าที่กันทำตามขนาด และรูปร่าง เป็นต้น

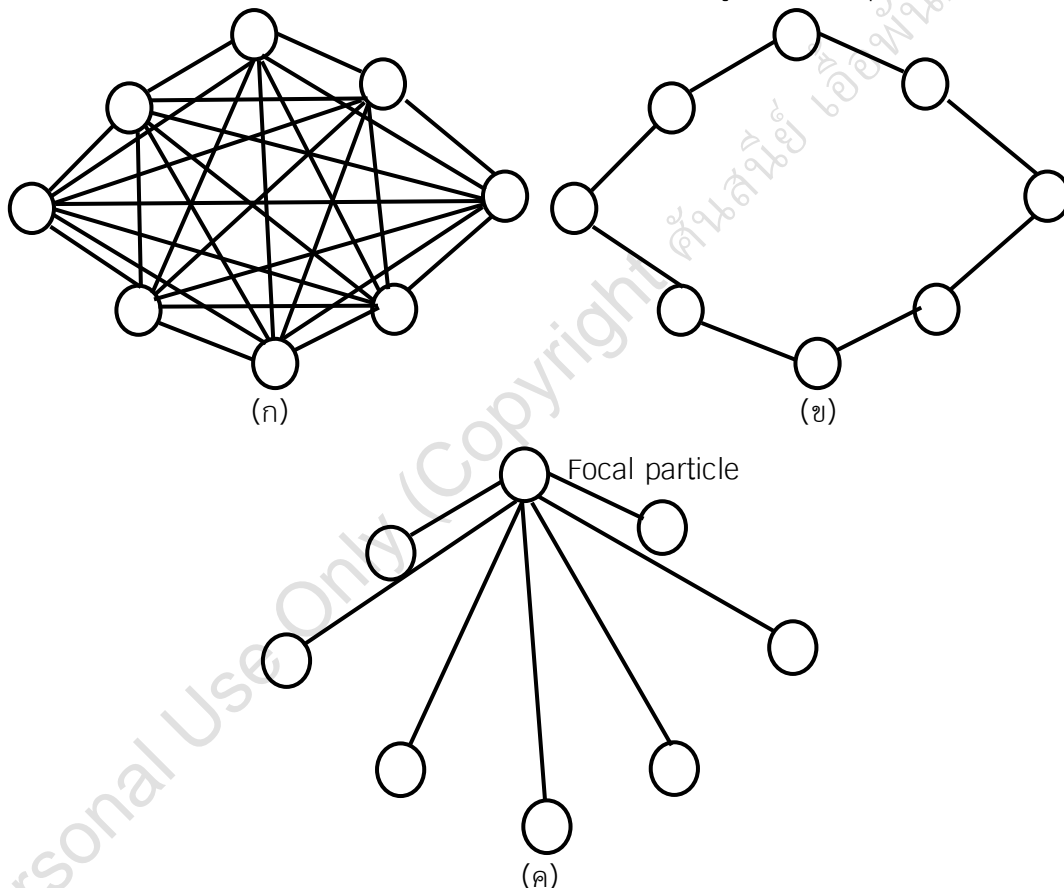
ทั้งนี้ในบทนี้จะกล่าวถึงความฉลาดเชิงกลุ่มเพียง 2 ประเภทคือ การหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (particle swarm optimization) ซึ่งเป็นวิธีการหาค่าที่เหมาะสมแบบเฟ้นสุ่ม (stochastic optimization) ที่จำลองพฤติกรรมเชิงสังคมของฝูงนก ดังนั้นรายบุคคลในที่นี้เป็นอนุภาค และรวมกลุ่มกัน โดยที่แต่ละอนุภาคเป็นตัวแทนของคำตอบที่เป็นไปได้ในปัญหาการหาค่าที่เหมาะสมนั้น นั่นเอง ซึ่งแต่ละอนุภาคจะบินในปริภูมิการค้นหามีหลายมิติ (multidimensional search space) และจะมีการปรับตำแหน่งของแต่ละอนุภาคตามประสบการณ์ของตนเองและของเพื่อนบ้าน ดังนั้นจะเป็นการนำตำแหน่งที่ดีที่สุดที่ตนเอง และเพื่อนบ้านค้นพบมาใช้ในการปรับตำแหน่งตนเองนั่นเอง ทำให้อนุภาคมีโอกาสในการบินเข้าหาคำตอบที่เหมาะสมที่สุดนั่นเอง สำหรับความฉลาดเชิงกลุ่มประเภทที่ 2 ที่จะกล่าวถึงคือ การหาค่าที่เหมาะสมที่สุดโดยอาณานิคมมด (ant colony optimization) ซึ่งเป็นการจำลองการวางฟีโรโมนของมดในการเดินทางหาแหล่งอาหาร ซึ่งมดจะพยายามหาเส้นทางที่สั้นที่สุดในการเดินทาง ทำให้อัลกอริทึมนี้เหมาะแก่ปัญหาการหาเส้นทางที่สั้นที่เหมาะสมสุด (shortest path optimization)

## 5.1 การหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (Particle Swarm Optimization)

การหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (particle swarm optimization (PSO)) [Engelbrecht07] นี่เป็นอัลกอริทึมการค้นหาค่าที่ขึ้นกับประชากร ซึ่งเป็นการจำลองพฤติกรรมเชิงสังคมของฝูงนก ทั้งนี้ความตั้งใจเริ่มแรกของแนวคิดเรื่องกลุ่มอนุภาคนี้คือการเลียนแบบ ทำทางของฝูงนกเชิงภูมิศาสตร์ที่คาดเดาไม่ได้ โดยที่มีจุดประสงค์ในการค้นพบรูปแบบที่ควบคุมความสามารถของนกในการบินพร้อมกัน และสามารถเปลี่ยนทิศทางการบินได้อย่างกะทันหัน โดยการรวมกลุ่มกันใหม่ในลักษณะที่เหมาะสมที่สุด ซึ่งจากการศึกษาที่กล่าวมานี้ทำให้เกิดอัลกอริทึมสำหรับการจัดระเบียบกลุ่มของอนุภาค ที่ง่ายและเป็นอัลกอริทึมการหาค่าที่เหมาะสมที่มีประสิทธิภาพ

ดังที่เคยกล่าวมาแล้วคือ อนุภาค 1 อนุภาค คือคำตอบที่เป็นไปได้ของปัญหาการหาค่าที่เหมาะสม และอนุภาคจะบินในปริภูมิการค้นหามีหลายมิติ ซึ่งการเปลี่ยนตำแหน่งของอนุภาคในปริภูมิการค้นหาคือขึ้นอยู่กับจิตวิทยาเชิงสังคมของรายบุคคล ที่ต้องการเลียนแบบความสำเร็จของรายบุคคลอื่น ดังนั้นการเปลี่ยนแปลงของอนุภาคในกลุ่มนั้นมีอิทธิพลมาจากประสบการณ์ หรือความรู้ของเพื่อนบ้าน ทั้งนี้รูปร่างของเพื่อนบ้านมีหลายรูปแบบ และมีการสร้างอัลกอริทึมตามแต่ละรูปแบบ มากมาย ซึ่งรูปแบบของเพื่อนบ้านที่ถูกนำมาใช้มีตัวอย่างดังนี้ [Engelbrecht02]

1. ทอพอโลยีแบบดาว (star topology) ดังแสดงในรูปที่ 5.1(ก) ซึ่งรูปแบบนี้ทำให้แต่ละอนุภาคสามารถติดต่อกับอนุภาคอื่นได้ทุกอนุภาค ดังนั้นแต่ละอนุภาคจะสนใจอนุภาคที่ดีที่สุดในกลุ่ม และแต่ละอนุภาคจะเลียนแบบอนุภาคที่ดีที่สุดในกลุ่มนี้เอง ซึ่งอัลกอริทึมที่จำลองสถานการณ์นี้คือ อัลกอริทึมที่ดีที่สุดแบบรวม (global best (gbest))
2. ทอพอโลยีแบบวงแหวน (ring topology) ในกรณีนี้แต่ละอนุภาคจะติดต่อกับเพื่อนบ้านที่ใกล้ที่สุด  $n$  อนุภาค ดังแสดงในรูปที่ 5.1(ข) ที่  $n=2$  ดังนั้นอนุภาคเคลื่อนที่ตามเพื่อนที่ดีที่สุดในกลุ่มเพื่อนบ้านที่ติดต่อกับได้ ซึ่งอัลกอริทึมที่จำลองสถานการณ์นี้คือ อัลกอริทึมที่ดีที่สุดแบบเฉพาะที่ (local best (lbest))
3. ทอพอโลยีแบบวงล้อ (wheel topology) ดังแสดงในรูปที่ 5.1(ค) ในกรณีนี้มีเพียง 1 อนุภาคเท่านั้นที่ติดต่อกับอนุภาคอื่นๆ ได้ ซึ่งอนุภาคนี้ถูกเรียกว่า อนุภาคจุดรวม (focal particle) ที่การเคลื่อนที่จะเป็นไปตามอนุภาคที่ดีที่สุด ถ้าการเปลี่ยนแปลงที่อนุภาคจุดรวมทำให้ประสิทธิภาพของตนเองดีขึ้น การเคลื่อนที่นี้จะถูกส่งไปยังอนุภาคอื่นๆ



รูปที่ 5.1 รูปแบบของเพื่อนบ้านสำหรับการจัดระเบียบกลุ่มของอนุภาค (ก) ทอพอโลยีแบบดาว (ข) ทอพอโลยีแบบวงแหวน และ (ค) ทอพอโลยีแบบวงล้อ

อัลกอริทึมที่จะกล่าวถึงในหัวข้อนี้คือ อัลกอริทึมที่ดีที่สุดแบบรายบุคคล (individual best) บางครั้งถูกเรียกว่า pbest, อัลกอริทึม gbest และอัลกอริทึม lbest ทั้งนี้เพื่อนบ้าน จะเป็นเพื่อนบ้านตามตัวเลขตัวชี้ของอนุภาค (numerical index of particle) ไม่ใช่การวัดทางภูมิศาสตร์ เช่น ระยะห่างแบบยูคลิด

ทั้ง 3 อัลกอริทึมที่จะกล่าวต่อไปนี้จะอยู่บนหลักการเดียวกันคือ กลุ่ม (swarm) จะประกอบด้วยเซตของอนุภาค โดยที่แต่ละอนุภาคเป็นตัวแทนของคำตอบที่เป็นไปได้ และเช่นที่เคย

กล่าวมาแล้วอนุภาคจะบินอยู่ในปริภูมิการค้นหาลายมิติ โดยที่ตำแหน่งของอนุภาคจะเปลี่ยนไปตามประสบการณ์ของอนุภาคเอง หรือของเพื่อนบ้าน ให้  $x_i(t)$  เป็นตำแหน่งของอนุภาค  $P_i$  ในปริภูมิไฮเปอร์ (hyperspace) ที่เวลา  $t$  และตำแหน่งของอนุภาคจะเปลี่ยนได้โดยการเพิ่มความเร็วก  $v_i(t)$  ให้กับตำแหน่งปัจจุบันดังนี้ [Engelbrecht02]

$$x_i(t) = x_i(t-1) + v_i(t) \quad (5.1)$$

ซึ่งความเร็วนี้ เป็นตัวขับในกระบวนการหาค่าที่เหมาะสม และสะท้อนถึงการแลกเปลี่ยนข้อมูลในสังคม

สำหรับทั้ง 3 อัลกอริทึมที่จะกล่าวถึงต่อไปนี้ จะมีสิ่งที่ใช้ร่วมกันคือการคำนวณหาค่าความเหมาะสม ซึ่งเปรียบเหมือนประสิทธิภาพของแต่ละอนุภาค เช่นต้องการหาค่า  $x_1$  และ  $x_2$  ที่

$$\min_{(x_1, x_2)} \left( \sin x_1 \sin x_2 \sqrt{x_1 x_2} \right) \quad (5.2)$$

ดังนั้นฟังก์ชันจุดประสงค์ (objective function)

$$f(x_1, x_2) = \sin x_1 \sin x_2 \sqrt{x_1 x_2} \quad (5.3)$$

สามารถถูกใช้เป็นฟังก์ชันหาค่าความเหมาะสมได้ และในการลู่เข้า (converge) นั้น คืออัลกอริทึมจะหยุดการทำงานในกรณีที่จำนวนรอบการเคลื่อนที่มีค่าเท่ากับจำนวนรอบที่สูงที่สุด หรือความเปลี่ยนแปลงของความเร็วของทุกอนุภาคมีค่าเข้าใกล้ 0 หรือคือไม่มีการเคลื่อนที่อีกแล้วนั่นเอง

#### 5.1.1 อัลกอริทึมที่ดีที่สุดแบบรายบุคคล (Individual Best)

สำหรับอัลกอริทึมนี้ [Engelbrecht02] แต่ละรายบุคคลจะเปรียบเทียบกับตำแหน่งปัจจุบันของตนเอง กับตำแหน่งที่ดีที่สุดของตนเอง (pbest) เท่านั้น และไม่มีการแลกเปลี่ยนข้อมูลอื่นกับอนุภาคอื่นแต่อย่างใด อัลกอริทึมนี้มีลักษณะดังนี้

1. ตั้งค่ากลุ่ม ( $P(t)$  ที่  $t=0$ ) ของอนุภาค โดยที่ ตำแหน่ง ( $x_i(t)$ ) ของอนุภาค  $i$  ( $P_i \in P(t)$ ) จะถูกสุ่มโดยให้ค่าอยู่ภายในปริภูมิไฮเปอร์ ที่ต้องการค้นหาคำตอบ
2. คำนวณค่าประสิทธิภาพ  $F$  ของแต่ละอนุภาค โดยใช้ตำแหน่งปัจจุบัน  $x_i(t)$
3. เปรียบเทียบค่าที่ได้ในข้อ 2 ของอนุภาค  $i$  กับค่าที่ดีที่สุดของตนเอง (pbest) ดังนี้

$$\text{If } F(x_i(t)) < pbest_i$$

$$pbest_i = F(x_i(t))$$

$$x_{pbest_i}(t) = x_i(t)$$

End

4. ปรับความเร็วของแต่ละอนุภาคดังนี้

$$v_i(t) = v_i(t-1) + \rho(x_{pbest_i} - x_i(t)) \quad (5.4)$$

โดยที่  $\rho$  เป็นค่าคงที่บวกที่ถูกสุ่มมา

5. ปรับตำแหน่งของแต่ละอนุภาค ตามสมการที่ 5.1 และตั้งค่า  $t=t+1$
6. กลับไปยังข้อ 2 และทำซ้ำ จนกระทั่งจะลู่เข้า (converge)

สำหรับอัลกอริทึมนี้ ยิ่งอนุภาคอยู่ไกลจากคำตอบที่ดีที่สุดที่เคยพบ การเปลี่ยนแปลงความเร็วยิ่งมีค่ามาก เพื่อที่จะดึงอนุภาคนั้นกลับมายังคำตอบที่ดีที่สุด ค่าขอบเขตบนของ  $\rho$  นั้นจะ

ถูกกำหนดโดยผู้ใช้ ซึ่งถ้าค่านี้มีมาก จะทำให้แนววิถี (trajectory) ของอนุภาคแกว่งกวัด (oscillate) ได้ แต่ถ้ามีค่าน้อย การเคลื่อนที่ของอนุภาคจะมีแนววิถีที่ค่อยเป็นค่อยไป

### 5.1.2 อัลกอริทึมที่ดีที่สุดแบบรวม (Global Best)

อัลกอริทึม gbest นี้เป็นการใช้โครงสร้างทอพอโลยีแบบดาว ดังนั้นการเคลื่อนที่ของอนุภาคจะขึ้นอยู่กับตำแหน่งที่ดีที่สุดของอนุภาคตัวที่ดีที่สุดในกลุ่ม และประวัติจากประสบการณ์ของตนเอง ดังนั้น อัลกอริทึมนี้สามารถสรุปได้ดังนี้ [Engelbrecht02]

1. ตั้งค่ากลุ่ม ( $P(t)$  ที่  $t=0$ ) ของอนุภาค โดยที่ ตำแหน่ง ( $x_i(t)$ ) ของอนุภาค  $i$  ( $P_i \in P(t)$ ) จะถูกสุ่มโดยให้ค่าอยู่ภายในปริภูมิไฮเปอร์ ที่ต้องการค้นหาคำตอบ
2. คำนวณค่าประสิทธิภาพ  $F$  ของแต่ละอนุภาค โดยใช้ตำแหน่งปัจจุบัน  $x_i(t)$
3. เปรียบเทียบค่าที่ได้ในข้อ 2 ของอนุภาค  $i$  กับค่าที่ดีที่สุดของตนเอง ( $pbest_i$ ) ดังนี้

$$\text{If } F(x_i(t)) < pbest_i$$

$$pbest_i = F(x_i(t))$$

$$x_{pbest_i}(t) = x_i(t)$$

End

4. เปรียบเทียบค่าที่ได้ในข้อ 2 ของอนุภาค  $i$  กับค่าที่ดีที่สุดของกลุ่ม ( $gbest$ ) ดังนี้

$$\text{If } F(x_i(t)) < gbest$$

$$gbest = F(x_i(t))$$

$$x_{gbest}(t) = x_i(t)$$

End

5. ปรับความเร็วของแต่ละอนุภาคดังนี้

$$v_i(t) = v_i(t-1) + \rho_1(x_{pbest_i} - x_i(t)) + \rho_2(x_{gbest} - x_i(t)) \quad (5.5)$$

โดยที่  $\rho_1$  และ  $\rho_2$  เป็นค่าที่ถูกสุ่มมา ทั้งนี้เทอมที่ 2 ในสมการที่ 5.3 นั้นเป็นส่วนประกอบ การรับรู้ (cognitive component) ในขณะที่เทอมที่ 3 เป็นส่วนประกอบทางสังคม (social component)

6. ปรับตำแหน่งของแต่ละอนุภาค ตามสมการที่ 5.1 และตั้งค่า  $t=t+1$
7. กลับไปยังข้อ 2 และทำซ้ำ จนกระทั่งจะลู่เข้า (converge)

ถ้าอนุภาคเคลื่อนที่ไปไกลจากตำแหน่งของอนุภาคที่ดีที่สุดและตำแหน่งที่ดีที่สุดของตนเอง จะทำให้มีการเปลี่ยนแปลงในความเร็วมาก เพื่อย้ายอนุภาคกลับมาใกล้ตำแหน่งที่ดีที่สุด ส่วนค่าตัวแปรสุ่ม  $\rho_1$  และ  $\rho_2$  นั้นโดยปกติจะถูกกำหนดดังนี้

$$\rho_1 = r_1 c_1 \text{ และ } \rho_2 = r_2 c_2 \quad (5.6)$$

โดยที่  $r_1$  และ  $r_2 \sim U(0,1)$  และค่า  $c_1$  และ  $c_2$  เป็นค่าคงที่ความเร่งบวก และได้มีการศึกษาว่าควรให้

$$c_1 + c_2 \leq 4 \quad (5.7)$$

เนื่องจากจะมีผลต่อแนวการเคลื่อนที่ของอนุภาค แต่ถ้า

$$c_1 + c_2 > 4 \quad (5.8)$$

อาจจะทำให้ความเร็วและตำแหน่งของอนุภาคจะเป็นอนันต์ (infinity) [Kennedy98]

ตัวอย่างที่ 5.1 ต้องการหาค่า  $(x,y)$  ที่ทำให้  $f(x,y) = x^2+2y$  น้อยที่สุด โดยที่  $x$  และ  $y$  เป็นจำนวนจริงใดๆ โดยใช้การหาค่าที่เหมาะสมโดยกลุ่มของอนุภาค และสมมุติให้มีอนุภาค 4 อนุภาค และอยู่ที่ตำแหน่ง  $(1,1)$ ,  $(3,-2)$ ,  $(-2,5)$ ,  $(-3,-2)$  ที่เวลา  $t$  ตามลำดับ และให้ค่า pbest ของแต่ละอนุภาคที่เวลา  $t-1$  เป็น 4, 6, 7, 8 และ gbest อยู่ที่ตำแหน่ง  $(0.5,1)$  และสมมุติให้ความเร็วของอนุภาค 1 ถึง 4 ที่เวลาที่  $t-1$  เป็น  $(0.5,0.5)$ ,  $(1,2)$ ,  $(0.4,3)$ ,  $(2,0.5)$  ตามลำดับ และค่า  $\rho_1$  และ  $\rho_2$  เป็น 1 ทั้งคู่

ถ้า gbest อยู่ที่ตำแหน่ง  $(0.5,1)$  ทำให้ค่าของ gbest = 2.25

ค่าความเหมาะสมของอนุภาคที่ 1 เป็น  $(1)^2+2(1) = 3$

ค่าความเหมาะสมของอนุภาคที่ 2 เป็น  $(3)^2+2(-2) = 5$

ค่าความเหมาะสมของอนุภาคที่ 3 เป็น  $(-2)^2+2(5) = 6$

ค่าความเหมาะสมของอนุภาคที่ 4 เป็น  $(-3)^2+2(-2) = 5$

ทั้ง 4 อนุภาคมีค่าความเหมาะสมที่เวลา  $t$  ที่ดีกว่าค่า pbest ของตนเองทำให้  $x_{pbest_1} = x_1(t)$ ,  $x_{pbest_2} = x_2(t)$ ,  $x_{pbest_3} = x_3(t)$  และ  $x_{pbest_4} = x_4(t)$  แต่ทั้ง 4 อนุภาคไม่ได้มีค่าความเหมาะสมที่ดีกว่า gbest ดังนั้นความเร็วของทั้ง 4 อนุภาคเป็น

$$\begin{aligned} v_1(t) &= v_1(t-1) + \rho_1(x_1(t) - x_1(t-1)) + \rho_2(x_{gbest} - x_1(t-1)) \\ &= v_1(t-1) + \rho_2 \left( \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} + 1 \left( \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \\ v_2(t) &= v_2(t-1) + \rho_2 \left( \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} - \begin{bmatrix} 3 \\ -2 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 1 \left( \begin{bmatrix} -2.5 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} -1.5 \\ 5 \end{bmatrix} \\ v_3(t) &= v_3(t-1) + \rho_2 \left( \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} - \begin{bmatrix} -2 \\ 5 \end{bmatrix} \right) = \begin{bmatrix} 0.4 \\ 3 \end{bmatrix} + 1 \left( \begin{bmatrix} 2.5 \\ -4 \end{bmatrix} \right) = \begin{bmatrix} 2.9 \\ -1 \end{bmatrix} \\ v_4(t) &= v_4(t-1) + \rho_2 \left( \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} - \begin{bmatrix} -3 \\ -2 \end{bmatrix} \right) = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} + 1 \left( \begin{bmatrix} 3.5 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} 5.5 \\ 3.5 \end{bmatrix} \end{aligned}$$

และสามารถทำการปรับตำแหน่ง  $x_i(t+1)$  สำหรับ  $i=1,2,3,4$  ได้ตามสมการที่ 5.1

$$\begin{aligned} x_1(t+1) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} \\ x_2(t+1) &= \begin{bmatrix} 3 \\ -2 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 5 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 3 \end{bmatrix} \\ x_3(t+1) &= \begin{bmatrix} -2 \\ 5 \end{bmatrix} + \begin{bmatrix} 2.9 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 4 \end{bmatrix} \end{aligned}$$



และ

$$\mathbf{x}_4(t+1) = \begin{bmatrix} -3 \\ -2 \end{bmatrix} + \begin{bmatrix} 5.5 \\ 3.5 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1.5 \end{bmatrix}$$

### 5.1.3 อัลกอริทึมที่ดีที่สุดแบบเฉพาะที่ (Local Best)

อัลกอริทึมนี้ lbest [Engelbrecht02] นี้เป็นการใช้เพื่อนบ้านในลักษณะของทอพอโลยีแบบวงล้อ ดังนั้นอนุภาคที่มีผลต่อการเคลื่อนที่คืออนุภาคที่อยู่ในเพื่อนบ้าน ที่ดีที่สุดและตำแหน่งที่ดีที่สุดของตนเอง ซึ่งอัลกอริทึมจะคล้ายกับ gbest เพียงแต่ในขั้นที่ 4 และ 5 เปลี่ยนจาก gbest เป็น lbest นั่นเอง

อัลกอริทึม lbest นี้จะซ้ำในการลู่เข้ามากกว่า gbest แต่จะให้คำตอบที่ดีกว่า และเป็นการค้นหา โดยครอบคลุมพื้นที่ได้กว้างกว่า โดยสรุปแล้วอัลกอริทึม lbest เป็นดังนี้

1. ตั้งค่ากลุ่ม ( $P(t)$  ที่  $t=0$ ) ของอนุภาค โดยที่ ตำแหน่ง ( $\mathbf{x}_i(t)$ ) ของอนุภาค  $i$  ( $P_i \in P(t)$ ) จะถูกสุ่มโดยให้ค่าอยู่ภายในปริภูมิไฮเปอร์ ที่ต้องการค้นหาคำตอบ
2. คำนวณค่าประสิทธิภาพ  $F$  ของแต่ละอนุภาค โดยใช้ตำแหน่งปัจจุบัน  $\mathbf{x}_i(t)$
3. เปรียบเทียบค่าที่ได้ในข้อ 2 ของอนุภาค  $i$  กับค่าที่ดีที่สุดของตนเอง ( $pbest_i$ ) ดังนี้

If  $F(\mathbf{x}_i(t)) < pbest_i$

$pbest_i = F(\mathbf{x}_i(t))$

$\mathbf{x}_{pbest_i}(t) = \mathbf{x}_i(t)$

End

4. เปรียบเทียบค่าที่ได้ในข้อ 2 ของอนุภาค  $i$  กับค่าที่ดีที่สุดของอนุภาคในเพื่อนบ้าน (lbest ของตนเอง) ดังนี้

If  $F(\mathbf{x}_i(t)) < lbest$  ของตนเอง

$lbest \text{ ของตนเอง} = F(\mathbf{x}_i(t))$

$\mathbf{x}_{own lbest}(t) = \mathbf{x}_i(t)$

End

5. ปรับความเร็วของแต่ละอนุภาคดังนี้

$$\mathbf{v}_i(t) = \mathbf{v}_i(t-1) + \rho_1(\mathbf{x}_{pbest_i} - \mathbf{x}_i(t)) + \rho_2(\mathbf{x}_{own lbest} - \mathbf{x}_i(t)) \quad (5.9)$$

โดยที่  $\rho_1$  และ  $\rho_2$  เป็นค่าที่ถูกสุ่มมา ทั้งนี้ทอมที่ 2 ในสมการที่ 5.3 นั้นเป็นส่วนประกอบ การรับรู้ (cognitive component) ในขณะที่ทอมที่ 3 เป็นส่วนประกอบทางสังคม (social component)

6. ปรับตำแหน่งของแต่ละอนุภาค ตามสมการที่ 5.1 และตั้งค่า  $t=t+1$
7. กลับไปยังข้อ 2 และทำซ้ำ จนกระทั่งจะลู่เข้า (converge)

#### 5.1.4 ตัวแปรในอัลกอริทึมการหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาค (Particle Swarm Optimization Parameters)

โดยปกติแล้วมีตัวแปร [Engelbrecht02] ที่มีผลกับอัลกอริทึมอยู่ 6 ตัวแปรคือ มิติของปัญหาซึ่งได้จากการพิสูจน์แล้วว่า PSO จะทำงานได้ดีในปัญหาที่มีมิติสูง (high dimensional problem) จำนวนของรายบุคคล ขอบเขตบนของ  $\rho$  ขอบเขตบนของความเร็ว ขนาดของเพื่อนบ้าน และน้ำหนักความเฉื่อย (inertia weight)

ในส่วนขอบเขตบนของความเร็วนั้น โดยปกติจะมีการกำหนดความเร็วสูงสุด (maximum velocity ( $v_{max}$ )) ทั้งนี้ขอบเขตบนนี้จะถูกกำหนดไว้ในทุกมิติ ซึ่งเป็นการป้องกันไม่ให้อนุภาคเคลื่อนที่เร็วเกินไปจากจุดหนึ่งไปอีกจุดหนึ่งในปริภูมิการค้นหา ดังนั้นการกำหนดจะเป็นดังนี้

$$\text{If } v_{ij}(t) > v_{max} \text{ then } v_{ij}(t) = v_{max} \quad (5.10)$$

$$\text{และ } \text{If } v_{ij}(t) < -v_{max} \text{ then } v_{ij}(t) = -v_{max} \quad (5.11)$$

โดยที่  $v_{ij}(t)$  เป็นความเร็วของอนุภาค  $i$  ( $P_i$ ) ในมิติ  $j$  และในเวลา  $t$  ทั้งนี้  $v_{max}$  ไม่ได้กำหนดขอบเขตตำแหน่งของอนุภาค เป็นเพียงกำหนดขอบเขตของขั้นในการเคลื่อนที่ในปริภูมิการค้นหาหลายมิติเท่านั้น

โดยปกติแล้วการกำหนด  $v_{max}$  นั้นจะเป็นการกำหนดให้เป็นฟังก์ชันของเรนจ์ของปัญหา ตัวอย่างเช่น ถ้าเรนจ์ของคำตอบในทุกมิติ ควรอยู่ในช่วง  $[-50, 50]$  แล้ว  $v_{max}$  อาจจะถูกตั้งให้แปรผันตรงกับค่า 50 ก็ได้ แต่อย่างไรก็ตามงานวิจัยของ Clerc และ Kennedy [Clerc02] ได้สรุปไว้ว่า อาจจะไม่จำเป็นต้องมี  $v_{max}$  ถ้าสมการในการปรับความเร็วในกรณี gbest เป็นดังนี้

$$v_i(t) = \kappa \left( v_i(t-1) + \rho_1 (x_{pbest_i} - x_i(t)) + \rho_2 (x_{gbest} - x_i(t)) \right) \quad (5.12)$$

โดยที่  $\kappa$  เป็นค่าสัมประสิทธิ์การหด (constriction coefficient) มีค่าดังนี้

$$\kappa = 1 - \frac{1}{\rho} + \frac{\sqrt{|\rho^2 - 4\rho|}}{2} \quad (5.13)$$

และ  $\rho = \rho_1 + \rho_2 > 4.0$  ทั้งนี้สมการที่ 5.12 สามารถใช้ได้กับ lbest เพียงแต่เปลี่ยนตัวแปรของ gbest เป็นตัวแปรของ lbest เท่านั้น

สำหรับขนาดของเพื่อนบ้านนั้น สามารถมองได้ว่า gbest เป็นฉบับง่ายของ lbest ที่มีเพื่อนบ้านเป็นทั้งกลุ่ม แต่อย่างไรก็ตาม gbest จะไวต่อ จุดต่ำเฉพาะที่ (local minimum) มากกว่า lbest เพราะทุกอนุภาคจะถูกดึงไปตามอนุภาคที่ดีที่สุดในกลุ่ม แต่ถ้าขนาดของเพื่อนบ้านเล็กลงและมีจำนวนกลุ่มเพื่อนบ้านมากกว่า 1 จะทำให้มีความไวต่อจุดต่ำเฉพาะที่น้อยลง และพื้นที่ในการค้นหาวางกว้าง แต่จะลู่เข้าช้ากว่าเช่นกัน

ส่วนน้ำหนักความเฉื่อยนั้น ถือได้ว่าเป็นการเพิ่มประสิทธิภาพของอัลกอริทึม ทั้งนี้ในการปรับความเร็วจะเปลี่ยนไปเป็น

$$v_i(t) = \phi v_i(t-1) + \rho_1 (x_{pbest_i} - x_i(t)) + \rho_2 (x_{gbest} - x_i(t)) \quad (5.14)$$

โดยที่ค่า  $\phi$  เป็นค่าน้ำหนักความเฉื่อย ซึ่งค่าน้ำหนักความเฉื่อยนี้ จะควบคุมผลความเร็วครั้งก่อนต่อความเร็วใหม่ ถ้าค่าน้ำหนักนี้มีค่ามากอาจจะทำให้เกิดการสำรวจในปริภูมิการค้นหาที่กว้าง

ในขณะที่ถ้าค่าน้ำหนักความเฉื่อยน้อย จะเป็นการค้นหาในพื้นที่แคบๆ โดยปกติแล้ว PSO จะเริ่มจากค่าน้ำหนักความเฉื่อยมาก และจะลดค่าลงไปตามเวลา และเพื่อให้อัลกอริทึมลู่เข้าได้มีการพิสูจน์แล้วว่า

$$\phi > \frac{1}{2}(c_1 + c_2) - 1 \quad (5.15)$$

โดยที่  $\phi \leq 1$  แต่ถ้าค่าน้ำหนักความเฉื่อยไม่เป็นไปตามสมการที่ 5.15 อัลกอริทึมการหาค่าที่เหมาะสมที่สุดโดยกลุ่มของอนุภาคจะลู่ออก (diverge)

#### 5.1.5 การปรับปรุง PSO (Modification PSO)

ในปัจจุบันมีการปรับปรุง PSO ในหลายรูปแบบรวมถึงที่จะกล่าวถึงต่อไปนี้ [Engelbrecht02]

1. การใช้การคัดสรร (selection) ซึ่งการคัดสรรนี้จะคล้ายกับการคัดสรรในการคำนวณเชิงวิวัฒนาการในบทที่ 4 โดยที่มีกระบวนการดังนี้

- สำหรับแต่ละอนุภาคในกลุ่ม ให้เปรียบเทียบคะแนน หรือค่าความเหมาะสมของตนเองกับเพื่อนที่ถูกเลือกมาแบบสุ่ม  $k$  อนุภาค
- นำค่าความเหมาะสมมาเรียง
- เลือกครึ่งบนของอนุภาค และคัดลอกตำแหน่งปัจจุบันของอนุภาคเหล่านี้ไปยังอนุภาคที่อยู่ครึ่งล่าง โดยไม่มีการเปลี่ยนค่า pbest ของอนุภาคครึ่งล่าง

ซึ่งการคัดสรรนี้จะทำก่อนที่จะถึงขั้นของการเปลี่ยนความเร็วในอัลกอริทึม ทั้งนี้กระบวนการนี้จะลดความหลากหลาย และเพิ่มความสามารถในการค้นหา แต่ขัดแย้งกับวัตถุประสงค์ของการคัดสรรในธรรมชาติ

2. การผสมพันธุ์ (breeding) ใน PSO ซึ่งมีกระบวนการดังนี้

- คำนวณความเร็วและตำแหน่งใหม่ของอนุภาค
- แต่ละอนุภาคจะถูกตั้งค่าความน่าจะเป็นของการผสมพันธุ์  $p_b$
- เลือกอนุภาคที่จะเป็นพ่อแม่มา 2 อนุภาค ( $P_a$  และ  $P_b$ ) และทำการสร้างลูก 2 อนุภาค โดยที่

$$x_a(t+1) = r_1 x_a(t) + (1-r_1) x_b(t) \quad (5.16)$$

$$x_b(t+1) = r_2 x_b(t) + (1-r_2) x_a(t) \quad (5.17)$$

$$v_a(t+1) = \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \|v_a(t)\| \quad (5.18)$$

$$v_b(t+1) = \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \|v_b(t)\| \quad (5.19)$$

โดยที่  $r_1, r_2 \sim U(0,1)$

- ตั้งค่าตำแหน่ง pbest ( $x_{pbest}$ ) ของแต่ละอนุภาคที่เกี่ยวข้องกับการผสมพันธุ์นี้ไปยังตำแหน่งปัจจุบัน

ทั้งนี้การเลือกพ่อแม่ในกระบวนการนี้ไม่ขึ้นอยู่กับค่าความเหมาะสมของอนุภาค ดังนั้นจะ  
เป็นการป้องกันไม่ให้อนุภาคที่ดีที่สุดมีอิทธิพลต่อการสร้างลูกมากกว่าอนุภาคอื่น

3. ทอพอโลยีของเพื่อนบ้าน โดยปกติแล้วเพื่อนบ้านใน lbest จะถูกกำหนดจากดัชนีของเพื่อน  
บ้าน ไม่ใช่ระยะห่าง แต่มีงานวิจัยของ Suganthan [Suganthan99] ที่ใช้เพื่อนบ้านเชิง  
พื้นที่ (spatial neighborhood) โดยที่อนุภาค  $P_a$  เป็นเพื่อนบ้านกับอนุภาค  $P_b$  ถ้า

$$\frac{\|x_a - x_b\|}{d_{\max}} < \xi \quad (5.20)$$

โดยที่  $d_{\max}$  เป็นระยะที่ห่างที่สุดระหว่างอนุภาค และ

$$\xi = \frac{3t + 0.6t_{\max}}{t_{\max}} \quad (5.21)$$

โดยที่  $t$  คือจำนวนรอบปัจจุบัน และ  $t_{\max}$  เป็นจำนวนรอบที่มากที่สุด สำหรับเพื่อนบ้านเชิง  
พื้นที่แบบนี้จะมีประสิทธิภาพของขนาดเพื่อนบ้านที่เล็กในตอนต้น และขนาดของเพื่อนบ้าน  
จะเพิ่มขึ้นตามเวลา และในที่สุดอัลกอริทึมจะกลายเป็น gbest ที่  $t_{\max}$  ซึ่งการมีขนาดเพื่อน  
บ้านที่เล็กในตอนต้นทำให้มีความหลากหลายมากกว่า และครอบคลุมพื้นที่การค้นหา  
มากกว่า ทำให้การลู่เข้าก่อนเวลา (premature convergence) อาจจะได้ยาก และใน  
ตอนปลายเมื่อเพื่อนบ้านเป็นทั้งกลุ่มจะทำให้ทุกอนุภาคเคลื่อนที่เข้าหาคำตอบที่ดีที่สุด  
ที่สุด

ตัวอย่างที่ 5.2 จำตัวอย่างที่ 5.1 สมมติอนุภาคที่ 1 และ 2 ถูกเลือกให้ทำการผสมพันธุ์ และ  
ค่า  $r_1=0.5$  และ  $r_2=0.3$  ทำให้อนุภาค 1 และ 2 มีค่าตำแหน่งและความเร็วใหม่เป็น

$$x_1(t+1) = r_1 x_1(t) + (1-r_1) x_2(t) = 0.5 \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} + 0.5 \begin{bmatrix} 1.5 \\ 3 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 2.25 \end{bmatrix}$$

$$\text{และ } x_2(t+1) = r_2 x_2(t) + (1-r_2) x_1(t) = 0.3 \begin{bmatrix} 1.5 \\ 3 \end{bmatrix} + 0.7 \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 1.15 \\ 1.95 \end{bmatrix}$$

$$\text{ทั้งนี้ค่า } \|v_1(t) + v_2(t)\| = \left\| \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 5 \end{bmatrix} \right\| = 5.7, \quad \|v_1(t)\| = \left\| \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \right\| = 0.5$$

$$\text{และ } \|v_2(t)\| = \left\| \begin{bmatrix} -1.5 \\ 5 \end{bmatrix} \right\| = 5.22$$

$$\text{ดังนั้นจากสมการที่ 5.18 และ 5.19 จะได้ } v_1(t+1) = \frac{\begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 5 \end{bmatrix}}{5.7} \cdot 0.5 = \begin{bmatrix} -0.13 \\ 0.48 \end{bmatrix} \text{ และ}$$

$$v_1(t+1) = \frac{\begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 5 \end{bmatrix}}{5.7} \cdot 5.22 = \begin{bmatrix} -1.37 \\ 5.04 \end{bmatrix}$$

ทั้งนี้ตำแหน่งของ  $pbest$  ของทั้งสองอนุภาคจะเป็น  $x_{pbest_1} = x_1(t+1)$  และ  $x_{pbest_2} = x_2(t+1)$

## 5.2 การหาค่าที่เหมาะสมที่สุดโดยอาณานิคมมด (Ant Colony Optimization)

การหาค่าที่เหมาะสมที่สุดโดยใช้อาณานิคมมดนี้เกี่ยวข้องกับหน้าที่ที่แตกต่างกันหลายหน้าที่ ซึ่งแต่ละหน้าที่จะเป็นงานของมดแต่ละชนิดที่ต่างกันเช่น [Engelbrecht02]

- กระบวนการสืบพันธุ์ (reproduction) เป็นหน้าที่ของนางพญามด
- กระบวนการป้องกัน (defense) เป็นหน้าที่ของมดทหาร
- กระบวนการหาอาหาร (food collection) เป็นหน้าที่ของมดงานที่ทำหน้าเฉพาะหน้าที่นี้
- กระบวนการเลี้ยงดูลูก (brood care) เป็นหน้าที่ของมดงานที่ทำหน้าเฉพาะหน้าที่นี้
- กระบวนการทำความสะอาดรัง (nest brooming) รวมทั้งการดูแลหลุมศพ (cemetery maintenance) เป็นหน้าที่ของมดงานที่ทำหน้าเฉพาะหน้าที่นี้
- กระบวนการสร้างและดูแลรัง (nest building and maintenance) เป็นหน้าที่ของมดงานที่ทำหน้าเฉพาะหน้าที่นี้

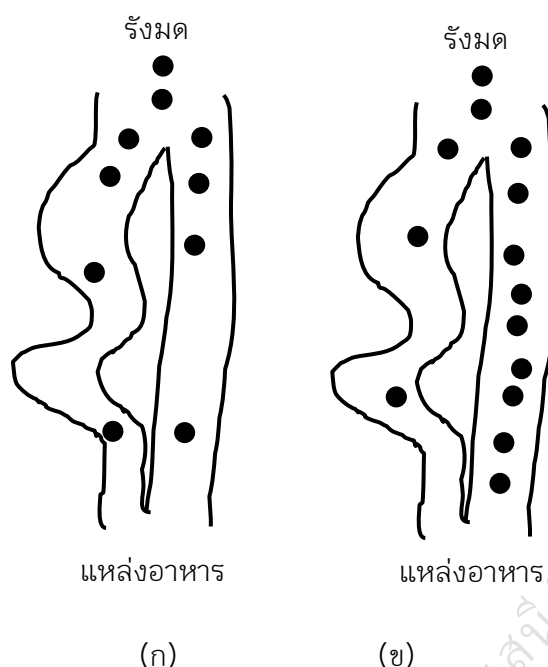
ซึ่งหน้าที่ที่กล่าวมาทั้งหมดนั้นจะถูกแบ่งตามลักษณะทางกายวิภาค (anatomy) เช่นขนาดตัว ลักษณะของกราม ซึ่งจะแบ่งระหว่างมดทหาร และมดงาน และสติกเมอร์จี (stigmergy) ซึ่งเป็นการกระจายทางพฤติกรรมภายในอาณานิคม ทั้งนี้สติกเมอร์จีในธรรมชาติ นั้นมีลักษณะดังนี้ [Dorigo99]

- การขาดการประสานงานจากตรงกลาง (central coordination)
- การติดต่อสื่อสารและการทำงานร่วมกันระหว่างรายบุคคลในอาณานิคมเป็นไปตามการปรับเปลี่ยนสิ่งแวดล้อมเฉพาะที่
- ป้อนกลับแบบบวก ซึ่งคือการเสริมกำลังของการกระทำ เช่นการติดตามรอยเดิน (trail-following) ไปยังแหล่งอาหาร เป็นต้น

ดังนั้นอัลกอริทึมจะถูกสร้างโดยเลียนแบบสติกเมอร์จีของมดแต่ละชนิด ซึ่งในที่นี้จะกล่าวถึงมดงานที่ทำหน้าที่หาอาหารชนิดเดียว เท่านั้น

มดงานที่ทำหน้าที่หาอาหารนี้จะพยายามหาเส้นทางที่สั้นที่สุดระหว่างรังและแหล่งอาหารเสมอ ซึ่งโดยปกติแล้วในการเดินทางของมดนั้น มดจะมีการปล่อยฟีโรโมน ระหว่างการเดินทางจากรังไปค้นหาอาหาร และเดินทางกลับรัง ซึ่งฟีโรโมนที่มดแต่ละตัวปล่อยออกไปจะถูกสะสมไปตามทางที่เดินเสมอ เมื่อมดตัวใหม่จะออกไปหาอาหารก็จะเลือกทางเดินที่มีฟีโรโมนสะสมมากกว่า ทำให้เส้นทางที่สั้นที่สุด มีฟีโรโมนสะสมมากที่สุด เพราะมีมดเดินทางไปและกลับตามเส้นทางนี้เร็วที่สุด และเนื่องจากทางที่ยาวกว่าต้องใช้เวลาในการเดินทางไปและกลับนานกว่าทำให้มีการสะสมฟีโรโมนน้อยกว่า และในขณะเดียวกันในธรรมชาตินั้นการสะสมฟีโรโมนนี้จะมีการระเหยไปตามเวลาอีกด้วย นี่จึงทำให้เส้นทางที่ยาวกว่ามีการสะสมที่น้อยกว่าเสมอ เช่นกัน ดังตัวอย่างในรูปที่ 5.2 ที่

ตอนต้นทั้งสองเส้นทางดูเหมือนจะมีการสะสมของฟีโรโมนไม่ต่างกัน และมดจะเลือกทางเดินแบบสุ่ม แต่เมื่อเวลาผ่านไปทางที่สั้นจะมีการสะสมฟีโรโมนมากกว่า และมดจะเลือกเส้นทางนี้มากกว่า



รูปที่ 5.2 การเดินทางหาอาหารของมดงาน (ก) ในตอนต้น และ (ข) เมื่อเวลาผ่านไป

การเดินทางของมดในลักษณะนี้ ถูกนำมาใช้สร้างแบบจำลองในการหาค่าที่เหมาะสมที่สุดเชิงการจัด (combinatorial optimization) เช่นการหาเส้นทางที่สั้นที่สุด (shortest path search) รวมทั้งการแก้ปัญหาพนักงานเดินตลาด (traveling salesman problem (TSP)) หรือปัญหาการกำหนดแบบกำลังสอง (quadratic assignment problem) หรือการจัดลำดับงาน (job-shop scheduling) หรือการหาเส้นทางเดินรถ (vehicle routing) เป็นต้น

ปัญหาพนักงานเดินตลาด คือปัญหาที่มีรายชื่อเมืองที่พนักงานต้องเดินทางไป และระยะทางระหว่างเมือง ดังนั้นปัญหาคือเส้นทางที่สั้นที่สุดในการเดินทางให้ครบทุกเมือง และเดินทางไปแต่ละเมืองเพียงครั้งเดียว และในท้ายที่สุดกลับมาถึงเมืองตั้งต้น ซึ่งปัญหานี้มีลักษณะที่เข้ากับการหาค่าที่เหมาะสมโดยอาณานิคมมดดังนี้ [Bonabeau99]

- เป็นปัญหาการหาเส้นทางที่สั้นที่สุด ที่สามารถอธิบายการทำงานของอาณานิคมได้ง่าย
- เป็นปัญหาแบบ NP-hard
- เป็นปัญหาที่มีการศึกษาอย่างกว้างขวาง จนปัญหานี้กลายเป็นเกณฑ์เปรียบเทียบสมรรถนะ (benchmark) สำหรับปัญหาการหาค่าที่เหมาะสมที่สุดเชิงการจัด
- เป็นปัญหาที่มีเอาไว้อ่อน เพราะง่ายในการทำความเข้าใจและอธิบายพฤติกรรมของอัลกอริทึม ซึ่งไม่ได้ถูกปิดกั้นด้วยเทคนิคมากนัก

อัลกอริทึมที่เกี่ยวข้องกับมดโดยส่วนใหญ่จะมีแนวความคิดพื้นฐานเหมือนกัน คือ ใช้กระบวนการป้อนกลับแบบบวก ที่อุปมาได้กับพฤติกรรมการวางรอยเดิน (trail-laying) การติดตามรอยเดิน ของมดบางชนิด และแมลงชนิดอื่น ในการเสริมกำลังในส่วนที่เป็นคำตอบที่ดี ที่สามารถสนับสนุนคุณภาพของคำตอบเหล่านั้น หรือเป็นการเสริมกำลังในส่วนคำตอบที่ดีโดยตรง และฟีโรโมนเสมือน (virtual pheromone) จะถูกใช้ในกระบวนการเสริมกำลังนี้ ซึ่งคำตอบที่ดีจะถูกเก็บในความจำ เพื่อที่จะทำให้สามารถพัฒนาให้ได้คำตอบที่ดีกว่าเดิม และจำเป็นที่จะต้องหลีกเลี่ยง

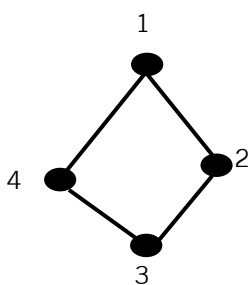
คำตอบที่ดีแต่ยังไม่ดีพอ ที่จะทำให้มีการเสริมกำลังไปยังตำแหน่งที่จะจำกัดการค้นหามากเกินไป ทำให้เกิดการลู่เข้าก่อนกำหนด หรือการชะงัก (stagnation) ของอัลกอริทึม จึงต้องมีการป้อนกลับแบบลบ ที่ดำเนินการผ่านการระเหยของฟีโรโมนตามเวลา ซึ่งสเกลของเวลาต้องไม่มากเกินไป มิฉะนั้น อาจเกิดการที่พฤติกรรมจะลู่เข้าก่อนกำหนด โดยเข้าสู่ส่วนย่อยของค่าที่เหมาะสมที่สุด (suboptimal) และสเกลเวลาต้องไม่น้อยเกินไป มิฉะนั้นจะไม่เกิดพฤติกรรมการทำงานร่วมกัน อัลกอริทึมจะทำการสำรวจพร้อมๆ กัน ของคำตอบที่แตกต่างกัน โดยการสะสมของมดตัวเดียวกัน และมดที่ทำงานดีในรอบนั้นๆ จะมีอิทธิพลกับการสำรวจของมดในรอบถัดไป เพราะมดสำรวจในคำตอบที่ต่างกัน ทำให้มีรอยเดินของฟีโรโมน ไปตามเส้นทางเหล่านั้น ถึงแม้จะมีเพียงมดที่ทำงานดีที่สุดจะถูกใช้ในการเสริมกำลังคำตอบของตนเอง แต่ยังมีผลของการทำงานร่วมกัน ที่คาบช่วงเวลา เพราะมดในรอบถัดไปจะใช้รอยเดินฟีโรโมน ช่วยในการนำทางในการสำรวจนั่นเอง

### 5.2.1 ระบบมด (Ant System)

ในปัญหาพนักงานเดินตลาดนั้น [Bonabeau99] มีเป้าหมายในการหาทัวร์ปิด (closed tour) ที่ใช้ระยะที่สั้นที่สุดในการเชื่อมต่อเมือง  $n$  เมือง และแต่ละเมืองจะถูกไปเยี่ยมเพียง 1 ครั้ง โดยปกติระยะระหว่างเมือง  $i$  ไปยังเมือง  $j$  เป็นระยะยูคลิด ดังนี้

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5.22)$$

โดยที่พิกัดของเมือง  $i$  และเมือง  $j$  เป็น  $(x_i, y_i)$  และ  $(x_j, y_j)$  ตามลำดับ โดยปกติแล้วปัญหาลักษณะนี้สามารถอธิบายได้ด้วยกราฟ  $(N, E)$  โดยที่โหนด  $N$  และเส้นเชื่อมระหว่างเมืองเป็นขอบ (edge)  $E$  ดังตัวอย่างในรูปที่ 5.3 ซึ่งกราฟสำหรับปัญหานี้ไม่จำเป็นต้องเป็นการเชื่อมต่อทั้งหมด (fully connect) ทั้งนี้ระยะทางไปและกลับระหว่างเมือง  $i$  และเมือง  $j$  ก็ไม่จำเป็นต้องเท่ากัน ถ้าระยะทางไปกลับระหว่างเมืองไม่เท่ากัน ( $d_{ij} \neq d_{ji}$ ) ปัญหานี้จะถูกเรียกว่า ปัญหาพนักงานเดินตลาดไม่สมมาตร (asymmetric TSP (ATSP)) แต่ไม่ว่าจะเป็นปัญหาพนักงานเดินตลาดไม่สมมาตร หรือสมมาตรก็ไม่เป็นปัญหาในการแก้ปัญหด้วยระบบมด (AS)



รูปที่ 5.3 กราฟ  $(N, E)$  ที่มี 4 เมือง และทุกเมืองไม่ได้เชื่อมต่อกัน

มดในระบบมดจะสร้างคำตอบสำหรับ TSP โดยการเคลื่อนที่ไปตามกราฟปัญหาจากเมือง 1 ไปยังอีกเมืองหนึ่ง จนกว่าจะเสร็จสิ้นการทัวร์ ดังนั้นในแต่ละรอบทัวร์ มด  $k$  (โดยที่  $k=1, 2, \dots, m$ ) จะสร้างทัวร์โดยการเดิน  $n = |N|$  ขั้น โดยใช้กฎความน่าจะเป็นของการเปลี่ยน (probabilistic transition rule) รอบของทัวร์มีดัชนีเป็น  $t$  โดยที่  $t=1, 2, \dots, t_{max}$  ทั้งนี้การเลือกเมือง  $j$  จากเมือง  $i$  นั้นจะเป็นไปตาม [Bonabeau99]

1. เมืองนั้นเคยถูกเยี่ยมชมแล้วหรือไม่ สำหรับมดแต่ละตัวจะมีความจำที่เรียกว่า รายการทาบู (tabu list) ซึ่งรายการนี้จะขยายภายในทัวร์ และจะว่างเปล่าระหว่างทัวร์ ดังนั้นความจำนี้จะถูกใช้ในการสร้างเซต  $J_i^k$  ซึ่งเซตนี้จะเป็นเซตของเมืองที่มด  $k$  ยังไม่ได้ไปเยี่ยมชมเมื่อมดตัวนี้อยู่ที่เมือง  $i$  ดังนั้นในตอนต้นของทัวร์ เซตนี้จะเป็นทุกเมืองยกเว้นเมือง  $i$
2. ทักษะวิสัย (visibility) ซึ่งเป็นส่วนกลับของระยะทาง ดังนี้

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (5.23)$$

เป็นข้อมูลที่ขึ้นอยู่กับข้อมูลเฉพาะที่เท่านั้น และเป็นตัวแทนของความต้องการเชิงสำนึก (heuristic desirability) ของการเลือกเมือง  $j$  เมื่ออยู่เมือง  $i$

3. จำนวนของรอยเคยฟีโรโมนเสมือน  $\tau_{ij}(t)$  บนขอบที่เชื่อมเมือง  $i$  ไปยังเมือง  $j$  ค่าฟีโรโมนนี้จะถูกปรับแบบเชื่อมต่อตรง (on-line) และค่านี้เปรียบเสมือนความต้องการในการเรียนรู้ (learned desirability) ในการเลือกเมือง  $j$  เมื่ออยู่เมือง  $i$  ซึ่งค่านี้จะเป็นข้อมูลโดยรวม (global information) ทั้งนี้รอยเคยฟีโรโมนนี้จะเปลี่ยนในระหว่างคำตอบของปัญหา เพื่อสะท้อนประสบการณ์ของมดที่ได้จากการหาคำตอบของปัญหา สำหรับกฎการเปลี่ยนสถานะ (transition rule) สำหรับมด  $k$  ในการเลือกเมือง  $j$  จากเมือง  $i$  เป็นดังนี้ [Bonabeau99]

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } j \in J_i^k \\ 0 & \text{else} \end{cases} \quad (5.24)$$

โดยที่ค่า  $\alpha$  และ  $\beta$  เป็นค่าที่ปรับได้ ซึ่งถ้าค่า  $\alpha=0$  ทำให้เมืองที่อยู่ใกล้ที่สุดถูกเลือก แต่ถ้า  $\beta=0$  มีเพียงฟีโรโมนที่มีผลต่อการเลือกเมือง แต่การทำแบบนี้อาจจะทำให้เป็นการเลือกเส้นทางที่จะนำไปสู่คำตอบที่ไม่ดีที่สุดก็เป็นได้

ในตอนต้น ( $t=0$ ) ของอัลกอริทึมฟีโรโมนของแต่ละขอบจะถูกกำหนดค่าให้มีค่าเป็นค่าบวกเล็กๆ  $\tau_0$  และหลังจากที่มดแต่ละตัวทำจนเสร็จทัวร์ มดจะมีการปล่อยฟีโรโมน  $\Delta\tau_{ij}^k(t)$  ไว้บนขอบ  $(i,j)$  ที่อยู่บนเส้นทางที่มด  $k$  เลือก โดยสมการคำนวณได้ดังนี้

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{if } (i,j) \in T^k(t) \\ 0 & \text{if } (i,j) \notin T^k(t) \end{cases} \quad (5.25)$$

โดยที่  $T^k(t)$  เป็นทัวร์ของมด  $k$  ในรอบทัวร์ที่  $t$  ส่วนค่า  $L^k(t)$  เป็นระยะรวมหรือความยาวของทัวร์นี้ ในขณะที่ค่า  $Q$  เป็นค่าที่ผู้ใช้ตั้งขึ้นมาโดยปกติแล้วจะเป็นค่าที่อยู่ในลำดับของขนาด (order of magnitude) เดียวกันกับความยาวของทัวร์ที่เหมาะสมที่สุด (optimal tour) และเพื่อให้สมจริงกับกระบวนการในธรรมชาติค่าฟีโรโมนเหล่านี้จะมีการระเหยไปตามกาลเวลาเช่นกัน ดังนี้

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (5.26)$$



โดยที่ค่า  $\rho$  เป็นค่าสัมประสิทธิ์ของความเสื่อม (coefficient of decay) ซึ่ง  $0 \leq \rho < 1$  และ

$$\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (5.27)$$

ซึ่งค่า  $m$  เป็นจำนวนมดทั้งหมด ซึ่งจะเป็นค่าเดิมตลอดทั้งอัลกอริทึม ทั้งนี้ถ้ามีจำนวนมากเกินไป อาจจะทำให้ได้คำตอบที่เป็นส่วนย่อยค่าเหมาะสมที่สุด หรือเป็นการลู่เข้าหาคำตอบที่ไม่ดี ในขณะที่ถ้าจำนวนมดมีน้อยเกินไป อาจจะไม่มีความหลากหลายของการทำงานร่วมกัน เนื่องจากกระบวนการความเสื่อมฟีโรโมน แต่มีงานวิจัย [Dorigo96] ที่แนะนำให้ใช้จำนวนมดเท่ากับจำนวนเมือง จะทำให้ได้คำตอบที่เหมาะสม และในตอนเริ่มต้นแต่ละตัวร มดอาจจะถูกปล่อยในเมืองที่ถูกเลือกมาแบบสุ่ม หรือเมือง 1 เมืองก็มีมดอยู่ 1 ตัวก็เป็นได้

นอกเหนือจากการปรับค่าฟีโรโมนดังสมการที่ 5.26 แล้วยังมีการใช้มดอภิชนนิยม (elitist ant) (e) ซึ่งมดตัวนี้จะเป็นมดที่ได้ทัวร์ที่ดีที่สุด ( $T^+$ ) ที่เจอตั้งแต่ตอนต้นของอัลกอริทึม โดยที่การปล่อยฟีโรโมนจะคล้ายกับสมการที่ 5.25 ดังนี้

$$\Delta \tau_{ij}^e(t) = \begin{cases} \frac{Q}{L^+} & \text{if } (i, j) \in T^+ \\ 0 & \text{if } (i, j) \notin T^+ \end{cases} \quad (5.28)$$

ทำให้สมการการปรับค่าฟีโรโมนในสมการที่ 5.26 เปลี่ยนเป็นดังนี้

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta \tau_{ij}(t) + e\Delta \tau_{ij}^e(t) \quad (5.29)$$

ซึ่งเป็นการบังคับให้การค้นหาเส้นทางที่ดีที่สุด อัลกอริทึมระบบมดสรุปได้ดังนี้

1. For ทุกขอบ  $(i,j)$

$$\tau_{ij}(0) = \tau_0$$

End For

2. For  $k=1$  ถึง  $m$

วางมด  $k$  ตามเมืองแบบสุ่ม

End For

3. ตั้ง  $T^+$  และ  $L^+$

4. For  $t=1$  ถึง  $t_{max}$

For  $k=1$  ถึง  $m$

สร้างทัวร์  $T^k(t)$  โดยทำ  $n-1$  ครั้งของการเลือกเมือง  $j$  ในขณะที่เมืองปัจจุบันเป็นเมือง  $i$  ตามสมการที่ 5.24

End For

For  $k=1$  ถึง  $m$

คำนวณ  $L^k(t)$  จาก  $T^k(t)$  ที่ได้ในขั้นตอนข้างต้น

End For

If มีทัวร์ที่ดีกว่า  $T^+$

ปรับ  $T^+$  และ  $L^+$

End If

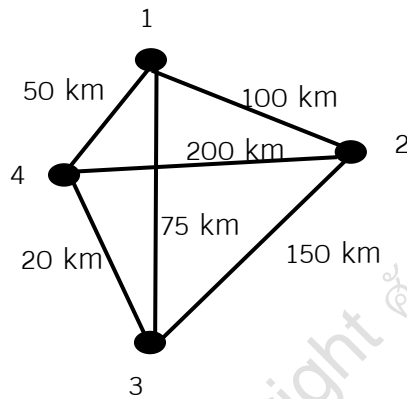
For ทุกขอบ (i,j)

    ปรับค่าฟีโรโมนตามสมการที่ 5.28

End For

End For

ตัวอย่างที่ 5.3 สมมุติปัญหาพนักงานเดินตลาดต้องไปทั้งหมด 4 เมือง และระยะทางไปกลับระหว่างเมืองเท่ากัน ดังรูปที่ 5.4 ในตอนต้นของการอัลกอริทึมทุกขอบจะมีฟีโรโมนเป็น 0.5 สมมุติให้รอบของการทำทัวร์เป็น  $t=1$  และมดตัวที่ 2 อยู่ที่เมืองที่ 3 ในตอนต้นของการทำทัวร์ ทำให้รายการทาบูของมดตัวนี้ที่เมืองนี้เป็นเซตว่างในขณะที่  $J_3^2 = \{1,2,4\}$



รูปที่ 5.4 กราฟสำหรับตัวอย่างที่ 5.3

เมืองที่มดตัวนี้จะไปต้องเลือกจากเมืองที่มีค่าความน่าจะเป็นการเปลี่ยนสถานะมากที่สุดดังนี้

$p_{33}^2(t) = 0$  จากสมการที่ 5.24 และสมมุติให้  $\alpha=\beta=1$  ทำให้

$$\sum_{l \in J_i^k} [\tau_{3l}(t)]^\alpha [\eta_{3l}]^\beta = [\tau_{31}(t)][\eta_{31}] + [\tau_{32}(t)][\eta_{32}] + [\tau_{34}(t)][\eta_{34}]$$
$$= 0.5 \frac{1}{75} + 0.5 \frac{1}{150} + 0.5 \frac{1}{20} = 0.035$$

$$\text{จะได้ } p_{31}^2(t) = \frac{[\tau_{31}(t)][\eta_{31}]}{\sum_{l \in J_i^k} [\tau_{3l}(t)]^\alpha [\eta_{3l}]^\beta} = \frac{0.5 \frac{1}{75}}{0.035} = 0.1905,$$

$$p_{32}^2(t) = \frac{[\tau_{32}(t)][\eta_{32}]}{\sum_{l \in J_i^k} [\tau_{3l}(t)]^\alpha [\eta_{3l}]^\beta} = \frac{0.5 \frac{1}{150}}{0.035} = 0.0952$$

$$\text{และ } p_{34}^2(t) = \frac{[\tau_{34}(t)][\eta_{34}]}{\sum_{l \in J_i^k} [\tau_{3l}(t)]^\alpha [\eta_{3l}]^\beta} = \frac{0.5 \frac{1}{20}}{0.035} = 0.7143$$

ดังนั้นมดตัวนี้จะเลือกเมืองที่ 4 ต่อไป และกระบวนการนี้จะถูกทำซ้ำจนกระทั่งมดตัวที่ 2 จะเสร็จสิ้นการทำทัวร์ และมดตัวอื่นก็จะดำเนินการในลักษณะเดียวกัน เมื่อมดทุกตัวดำเนินการจนเสร็จสิ้นทัวร์รอบนี้แล้วจึงมีการปรับค่าฟีโรโมนตามสมการที่ 5.28 ต่อไป และมดทุกตัวจะเริ่มการทำทัวร์ในรอบต่อไป และกระบวนการทั้งหมดจะถูกทำซ้ำจนกระทั่งจำนวนรอบทัวร์เป็นจำนวนที่สูงสุดที่กำหนดไว้ และคำตอบจะเป็นทัวร์ที่ดีที่สุดของทั้งหมดนั่นเอง

### 5.2.2 ระบบอาณานิคมมด (Ant Colony System)

ระบบอาณานิคมมด (Ant colony system (ACS)) นี้ถูกสร้างขึ้นมาเพื่อปรับปรุงคุณภาพของ AS ที่สามารถหาคำตอบที่เหมาะสมที่สุดในช่วงเวลาที่เป็เหตุเป็นผล แต่เฉพาะปัญหาที่มีขนาดเล็ก ทั้งนี้กฎการเปลี่ยนสถานะ การปรับรอยเดินฟีโรโมนเปลี่ยนไป โดยเป็นการปรับเฉพาะที่ของรอยเดินฟีโรโมนที่เป็นการให้ความเห็นด้วยกับการสำรวจ และมีการใช้รายการให้เลือก (candidate list) ที่จำกัดตัวเลือกของเมืองต่อไปที่จะไปเยี่ยม

กฎการเปลี่ยนสถานะ ในกรณีนี้เป็นดังนี้ มดตัวที่  $k$  เมื่ออยู่เมือง  $i$  จะเลือกเมือง  $j$  จาก

$$j = \begin{cases} \arg \max_{u \in J_i^k} \{ [\tau_{iu}(t)][\eta_{iu}]^\beta \} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \quad (5.30)$$

โดยที่  $q \sim U(0,1)$  และ  $q_0$  เป็นตัวแปรที่  $0 \leq q_0 \leq 1$  และ  $J \in J_i^k$  และจะถูกเลือกมาแบบสุ่มตามความน่าจะเป็น นี้

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)][\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)][\eta_{il}]^\beta} \quad (5.31)$$

ซึ่งใกล้เคียงกับสมการที่ 5.24 มาก ดังนั้น กฎการเปลี่ยนสถานะของ ACS จะเหมือนกับของ AS เมื่อ  $q > q_0$  และจะต่างเมื่อ  $q \leq q_0$  ซึ่งเป็นการสำรวจที่ขึ้นกับความรู้ที่ได้จากปัญหา นั่นคือระยะทางระหว่างเมือง และความรู้ที่ถูกเรียนรู้มาที่ถูกเก็บไว้ในรูปของฟีโรโมน ในขณะที่ถ้า  $q > q_0$  จะไปในทางการสำรวจมากกว่า ทั้งนี้ถ้า  $q_0$  เข้าใกล้ 1 คำตอบที่เหมาะสมที่สุดเฉพาะที่ (local optimal solution) จะถูกเลือก ในขณะที่ถ้า  $q_0$  เข้าใกล้ 0 ทุกคำตอบเฉพาะที่จะถูกนำมาพิจารณา

ในส่วนของการปรับฟีโรโมน ในกรณีของ AS มดทุกตัวจะปล่อยฟีโรโมน และจะเป็นการปรับในทุกขอบ แต่ใน ACS เฉพาะมดตัวที่ได้ทัวร์ที่ดีที่สุดตั้งแต่ตอนต้นของอัลกอริทึมจะเป็นคนปล่อยฟีโรโมน และจะปรับเฉพาะขอบที่อยู่ในทัวร์ที่ดีที่สุดเท่านั้น โดยการปรับค่าเป็นดังนี้

$$\tau_{ij}(t) \leftarrow (1 - \rho) \tau_{ij}(t) + \rho \Delta \tau_{ij}(t) \quad (5.32)$$

โดยที่ขอบ  $(i,j)$  อยู่ใน  $T^+$  และเช่นเดิมค่า  $\rho$  เป็นค่าสัมประสิทธิ์ความเสื่อม และ

$$\Delta \tau_{ij}(t) = \frac{1}{L^+} \quad (5.33)$$

โดยที่  $L^+$  เป็นความยาวของ  $T^+$

นอกเหนือจากนี้ยังมีการปรับเฉพาะที่ของรอยเดินฟีโรโมน ซึ่งเป็นการปรับเมื่อมดที่  $k$  กำลังทำทัวร์ และอยู่เมือง  $i$  และเลือกเมือง  $j \in J_i^k$  ซึ่งการปรับเป็น

$$\tau_{ij}(t) \leftarrow (1 - \rho) \tau_{ij}(t) + \rho \tau_0 \quad (5.34)$$

โดยที่  $\tau_0$  ฌเป็นค่าฟีโรโมนตั้งต้น โดยปกติค่านี้อาจจะเป็น

$$\tau_0 = (nL_{nn})^{-1} \quad (5.35)$$

โดยที่  $n$  เป็นจำนวนเมืองและ  $L_{nn}$  เป็นความยาวของทัวร์ที่ถูกสร้างจากสำนึกจำเพื่อนบ้านที่ใกล้ที่สุด (nearest neighbor heuristic) ที่ให้คำตอบที่ดี

เมื่อมดผ่านขอบนั้น สมการที่ 5.35 จะทำให้ระดับฟีโรโมนลดลง ซึ่งทำให้การผ่านขอบนั้นลดลงไปเรื่อยๆ ซึ่งเป็นการชักจูงให้ไปสำรวจเส้นทางที่ยังไม่เคยไปมากบขึ้น ทำให้มดไม่ลู่เข้าหาเส้นทางเดียวกัน ทำให้มีโอกาสที่มีมดบางตัวเข้าใกล้คำตอบมากขึ้น

ใน ACS มีการใช้รายการให้เลือก (candidate list) ซึ่งคือรายการของเมืองที่ต้องการไปเมื่ออยู่ที่เมืองเมืองหนึ่ง แทนที่จะหาเมืองที่ไปจากเมืองทั้งหมด มดจะเลือกเมืองจากรายการให้เลือกก่อน โดยใช้สมการที่ 5.30 และ 5.31 และเมื่อมดไปทุกเมืองในรายการให้เลือกแล้ว จึงจะเลือกเมืองจากรายการที่เหลือ (เมืองที่ยังไม่ได้ไปที่เหลือ) โดยเลือกจากเมืองที่ใกล้ที่สุด ในรายการนี้มีจำนวนเมือง  $cl$  เมือง ซึ่งเป็นเมืองที่ใกล้ที่สุด  $cl$  เมือง นั้นเอง และเมืองในรายการให้เลือกจะถูกเรียงไว้โดยเรียงจากเมืองที่ใกล้ที่สุดไปยังเมืองที่ไกลที่สุด อัลกอริทึม ACS โดยสรุปเป็นดังนี้

1. For ทุกขอบ  $(i,j)$

$$\tau_{ij}(0) = \tau_0$$

End For

2. For  $k = 1$  ถึง  $m$

วางมด  $k$  ตามเมืองแบบสุ่ม

End For

3. ตั้ง  $T^+$  และ  $L^+$

4. For  $t = 1$  ถึง  $t_{max}$

For  $k = 1$  ถึง  $m$

สร้างทัวร์  $T^k(t)$  โดยทำ  $n-1$  ครั้งของการเลือกเมือง  $j$  ในขณะที่เมืองปัจจุบันเป็นเมือง  $i$  ดังนี้

If มีเมือง  $j \in$  รายการให้เลือก

เลือกเมืองตามสมการที่ 5.30 และ 5.31

Else

เลือกเมือง  $j \in J_i^k$  ที่ใกล้ที่สุด

End If

หลังจากการเปลี่ยนสถานะของมดตัวที่  $k$  ให้ทำการปรับค่าฟีโรโมนตามสมการที่ 5.34

```

End For
For  $k = 1$  ถึง  $m$ 
    คำนวณ  $L^k(t)$  จาก  $T^k(t)$  ที่ได้ในขั้นตอนข้างต้น
End For
If มีตัวที่ต่ำกว่า  $T^+$ 
    ปรับ  $T^+$  และ  $L^+$ 
End If
For ทุกขอบ  $(i,j) \in T^+$ 
    ปรับค่าฟีโรโมนตามสมการที่ 5.32
End For
End For

```

---

### คำถามท้ายบทที่ 5

5.1 ให้เขียนโปรแกรมสำหรับการฝึกสอนเปอร์เซปตรอนหลายชั้น โดยใช้การหาค่าที่เหมาะสมที่สุด โดยกลุ่มของอนุภาค

5.2 สมมุติให้มด  $A_1$  เดินไปตามทางที่สั้นที่สุดในระหว่างสองเส้นทาง จากรังไปหาแหล่งอาหาร ในขณะที่มด  $A_2$  เดินไปในทางที่ยาวกว่า หลังจากมด  $A_2$  ถึงแหล่งอาหาร เส้นทางใดในระหว่างสองเส้นทางนี้จะมีโอกาสถูกเลือกมากกว่ากัน จงอธิบาย

5.3 จากตัวอย่างที่ 5.3 ถ้าเป็นการคำนวณโดยใช้ ACS จงหาว่าเมืองต่อไปที่จะไปคือเมืองอะไร โดยที่รายการให้เลือกเมื่อมดตัวที่ 2 อยู่เมืองที่ 3 คือ  $\{4\}$

---

## บรรณานุกรม Bibliography



[Bezdek94] J. C. Bezdek, "What is computational intelligence?" in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks, and C. J. Robinson (Eds.) Piscataway, NJ: IEEE Press, 1994, pp 1 – 11.

[Bonabeau99] E. Bonabeau, M. Dorigo, and G. Theraulez, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Inc., New York, USA., 1999.

[Clerc02] M. Clerc and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space", *IEEE Transactions on Evolutionary Computation*, vol 6. 2002, pp. 58 – 73.

[Dorigo96] M. Dorigo, V. Maniezzo, and A. Coloni, "The ant system: optimization by a colony of cooperating agents", *IEEE Trans. Syst. Man Cybern. B*, vol 26, 1996, pp. 29 – 41.

[Dorigo99] M. Dorigo, *Artificial Life: The swarm intelligence approach*, Tutorial TD1, Congress on Evolutionary Computing, Washington, DC., 1999.

[Dumitrescu00] D. Dumitrescu, B. Lazzerini, L.C. Jain, and A. Dumitrescu, *Evolutionary Computation*, CRC Press LLC, Florida, USA., 2000.

[Eberhart07] R. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*, Morgan Kaufmann Publishers, USA., 2007.

[Eng05] G. K. Eng, and A. M. Ahmad, "Malay Speech Recognition using Self-Organizing Map and Multilayer Perceptron", *Proceedings of the Postgraduate Annual Research Seminar*, 2005.

[Engelbrecht02] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd., West Sussex, England, 2002.

[Engelbrecht07] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, Ltd., West Sussex, England, 2007.

[Fogel95] D. B. Fogel, "Review of Computational Intelligence: Imitating Life (book Review)", *Proceedings of the IEEE*, Vol. 83, Issue 11, 1995, pp.1588 – 1592.

[Fogel00a] D. B. Fogel, *Evolutionary Computation: Principles and Practice for Signal Processing*, SPIE-The international Society for Optical Engineering, Washington, USA., 2000.

[Fogel00b] D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, USA., 2000.

[Goldberg05] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman, Inc., USA., 2005.

[Haykin94] S. Haykin, *Neural Networks: A comprehensive Foundation*, Macmillan Publishing Company, Inc. New Jersey, USA., 1994.

[Haykin09] S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, Inc., USA., 2009.

[Karaboga05] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization", Technical Report –TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[Kennedy98] J. Kennedy "The behavior of particles" in VW Porto, N. Saravanan, D. Waagen(eds), *Proceedings of the 7<sup>th</sup> International Conference on Evolutionary Programming*, 1998. pp. 581 – 589.

[Klir95] G. J. Klir and B. Yuan, *Fuzzy Stes and Fuzzy Logic: Theory and Applications*, Prentice Hall Inc., New Jersey, USA., 1995.

[Klir97] G. J. Klir, U. H. St. Clair, and B. Yuan, *Fuzzy Set Theory: Foundations and Applications*, Prentice Hall Inc., New Jersey, USA., 1997.

[Kohavi95] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[Kosko87] B. Kosko, "Fuzzy associative memories", in *Fuzzy Expert Systems*, A. Kandel, Ed. Reading, MA: Addison-Wesley, 1987.

[Kosko88] B. Kosko, “Bidirectional Associative Memories”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol 18(1), 1988, pp. 49 – 60.

[Koza92] J. R. Koza *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, England, 1992.

[Kreyszig05] E. Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, Inc., New York USA., 2005.

[Kruse95] R. Kruse, J. Gebhardt, and F. Klawonn, *Foundations of Fuzzy Systems*, John Wiley & Sons Ltd., West Sussex, England, 1995.

[Mitchell98] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Massachusetts Institute Technology, USA., 1998.

[Ohnishi90] N. Ohnishi, A. Okamoto, and N. Sugiem, “Selective Presentation of Learning Samples for Efficient Learning in Multi-Layer Perceptron”, *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol 1, 1990, pp. 688 – 691.

[Rosenblatt58] F. Rosenblatt, “The Perceptron: A probabilistic model for information storage and organization in the brain”, *Psychological Review*, vol 65, 1958, pp. 386 – 408.

[Ross04] T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons Ltd., West Sussex, England, 2004.

[Suganthan99] P. N. Suganthan, “Particle swarm optimizer with neighborhood operators”, *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999, pp. 1958 – 1961.

[Theodoridis09] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Elsevier Inc., London, UK., 2009.

[Thodberg91] H. H. Thodberg, “Improving Generalization of Neural Networks through Pruning”, *International Journal of Neural Systems*, 1(4), 1991, pp. 317 – 326.



[Turing50] A. M. Turing, Computing Machinery and Intelligence, *Mind*, vol 59, 1950, pp. 433 – 460.

[Zadeh95] L. A. Zadeh, “Fuzzy Sets”, *Information and Control*, 8(3), 1965, pp. 338–353.

For Personal Use Only (Copyright คำนวณด้วย เอื้อพันธ์วิริยะกุล 2556)