

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

บทนำ

โดยทั่วไปปัญหาการตัดสินใจหลาย ๆ ปัญหาที่มีทางเลือกของคำตอบที่มากกว่าสองทางเลือก จึงเป็นการยากในการค้นหาคำตอบที่ดีที่สุด แม้ว่าจะสามารถสร้างขั้นตอนวิธีทางคอมพิวเตอร์ (อัลกอริทึม) มาใช้ในการค้นหาคำตอบ แต่อาจจะต้องประสบปัญหาการค้นหาคำตอบที่ดีที่สุดเมื่อทำการค้นหาคำตอบในปัญหาการตัดสินใจที่มีขนาดใหญ่และซับซ้อน ทำให้ต้องใช้เวลานานในการค้นหาคำตอบหรือไม่สามารถทำการค้นหาคำตอบที่ดีที่สุดของปัญหาการตัดสินใจนั้น ๆ ได้ วิธีโลคอลเสิร์ช (Local search) ก็เป็นอีกวิธีหนึ่งที่สามารถนำมาใช้ในการค้นหาคำตอบที่เป็นไปได้ (Feasible solution) ได้ แต่เมื่อนำมาใช้ในการค้นหาคำตอบของปัญหาการตัดสินใจที่มีขนาดใหญ่และซับซ้อน อัลกอริทึมจะพลาดโอกาสในการค้นพบคำตอบภายในพื้นที่ที่น่าจะมีคำตอบที่มีคุณภาพดีกว่าหรือคำตอบที่ดีที่สุด และเมื่ออัลกอริทึมทำการค้นหาคำตอบจนครบรอบการกระทำซ้ำที่กำหนด คำตอบที่ได้นั้นอาจจะมีคุณภาพแย่กว่าคำตอบที่ดีที่สุดมาก ดังนั้นหากทราบถึงทิศทางของค่าการตัดสินใจของตัวแปรเพียงจำนวนหนึ่ง และนำไปใช้สำหรับกำหนดค่าการตัดสินใจของตัวแปรในขั้นตอนของการค้นหาคำตอบ จะทำให้ทางเลือกของปัญหาการตัดสินใจนั้นมีจำนวนลดลง และสามารถลดจำนวนรอบของการทำงานในการค้นหาคำตอบได้อีกด้วย งานวิจัยนี้จึงศึกษาวิธีการเรียนรู้และการพยากรณ์ค่าการตัดสินใจของตัวแปรร่วมกับการประยุกต์ใช้โลคอลเสิร์ชแบบสร้างเงื่อนไข (Constraint-based local search: CLS) เราประเมินประสิทธิภาพของวิธีการที่นำเสนอ โดยใช้ปัญหาการมอบหมายงาน (Generalized Assignment Problem: GAP) เนื่องจากมีคำตอบที่เป็น Benchmark จากการคำตอบด้วยวิธีอื่น ๆ

วิธีการหาคำตอบสำหรับปัญหาการตัดสินใจมีอยู่มากมาย วิธีการหาคำตอบที่ให้ค่าใกล้เคียงคำตอบที่ดีที่สุดในระยะเริ่มต้นเป็นวิธีหนึ่งที่ใช้สำหรับค้นหาคำตอบสำหรับปัญหาการตัดสินใจ ซึ่งอาศัยแนวคิดของวิธีการหาคำตอบแบบเนเบอร์ฮูด (Neighborhood search) ที่มีโครงสร้างแบบง่าย ๆ (การค้นหาคำตอบที่ดีกว่าคำตอบปัจจุบันภายในเนเบอร์ฮูดที่กำหนด) เช่น วิธี saving (Clarke & Wright, 1964) วิธีวิวิธศักดิ์ อื่น ๆ (Solomon, 1987) นอกจากวิธีการหาคำตอบที่ให้ค่าใกล้เคียงคำตอบที่ดีที่สุดในระยะเริ่มต้นแล้ว ยังมีวิธีอื่น ๆ เพื่อการปรับปรุงและการพัฒนาวิธีการหาคำตอบที่ให้ค่าที่ดีที่สุด (Exact method) เช่น วิธี Branch-and-bound ซึ่งอาจทำได้โดยการลดขนาดของพื้นที่ในการค้นหาคำตอบ (Search space) หรือการลดจำนวนทางเลือกของคำตอบที่

เป็นไปได้ (Mingozzi et al., 1999) และการใช้วิธีสถิติในการหาค้นหาคำตอบที่ดีภายในระยะเวลาอันรวดเร็ว (Golden et al., 1998) เช่น วิธีจำลองการตกผลึกทางเคมี (Simulated annealing: SA), วิธีทาบู (Tabu search) (Osman, 1993; Potvin et al., 1996a), วิธีเชิงพันธุกรรม (Genetic algorithm: GA) (Potvin et al., 1996b) และวิธีระบบปัญญาประดิษฐ์ต่าง ๆ (Artificial intelligence) เช่น วิธีจำลองการเลือกเส้นทางของมด (Ant colony optimization) (Bullnheimer et al., 1997) เป็นต้น สำหรับแนวความคิดในการพัฒนาวิธีสถิติสำนักวิจัยได้พยายามลดจำนวนพารามิเตอร์ที่ใช้ในอัลกอริทึมให้น้อยลงและเพิ่มการเรียนรู้ด้วยตนเองเพื่อให้วิธีสถิติมีความยืดหยุ่นและมีประสิทธิภาพมากยิ่งขึ้น (Indra-Payooong et al., 2003)

โครงสร้างและลักษณะปัญหาการตัดสินใจ

โดยทั่วไปปัญหาในภาคธุรกิจและอุตสาหกรรมเป็นปัญหาการตัดสินใจที่มีขนาดใหญ่และซับซ้อน วิธีการสร้างแบบจำลองการตัดสินใจของปัญหามีหลายวิธีที่แตกต่างกันไป โดยวัตถุประสงค์ของปัญหาเพื่อต้องการลดต้นทุน หรือเพิ่มกำไรในการดำเนินงาน ความซับซ้อนของปัญหาการตัดสินใจสามารถแบ่งออกเป็น 2 ส่วนที่สำคัญด้วยกันคือ 1) ความซับซ้อนของการสร้างแบบจำลองการตัดสินใจ (Decision model) และ 2) ความซับซ้อนของอัลกอริทึมเพื่อใช้ในการหาคำตอบ (Solution method) วิธีสถิติเป็นวิธีหนึ่งที่ใช้สำหรับหาคำตอบของปัญหาการตัดสินใจ วิธีสถิติที่ใช้ในการหาคำตอบของปัญหาการตัดสินใจมีอยู่มากมาย อาทิเช่น วิธี Greedy ร่วมกับ วิธี Local search โดย Martello และ Toth (Martello et al., 1981; Martello et al., 1990), การค้นหาคำตอบด้วยวิธีทาบู (Tabu search) ร่วมกับวิธีจำลองการตกผลึก (Simulated annealing) โดย Osman (Osman, 1995), วิธี Variable depth search โดย Racer และ Amini (Amini et al., 1995; Racer et al., 1994) เป็นต้น สำหรับความซับซ้อนของการสร้างแบบจำลองการตัดสินใจนั้นแบบจำลองในการตัดสินใจในรูปแบบต่าง ๆ มีหลากหลายวิธีที่แตกต่างกันไป อาจจะเขียนอยู่ในรูปของลำดับขั้นตอนการทำงาน (Procedure) หรือแบบจำลองทางคณิตศาสตร์ (Mathematical model) เป็นต้น การใช้แบบจำลองการตัดสินใจแต่ละแบบนั้นขึ้นอยู่กับลักษณะของปัญหานั้น ๆ ว่าเหมาะสมกับแบบจำลองแบบใด ในงานวิจัยนี้นำเสนอวิธีการจำลองปัญหาการตัดสินใจที่อยู่ในรูปแบบของเงื่อนไข (Constraint-based modeling: CBM) เนื่องจากส่วนใหญ่ปัญหาการตัดสินใจเป็นปัญหาที่มีตัวแปรการตัดสินใจมีค่าเป็น 0 หรือ 1 นั้นหมายถึง การตัดสินใจเลือกหรือไม่เลือกดำเนินงานนั้น ๆ ดังนั้นการกำหนดปัญหาเพื่อแก้ปัญหาหนึ่ง ต้องกำหนดปัญหาดังกล่าวให้อยู่ในรูปแบบตารางเมตริกซ์ 0/1 โดยเขียนปัญหาให้อยู่ในรูปแบบเงื่อนไขทั้งหมด โดยใช้เงื่อนไขแบบเซต (Set partitioning) (Balas & Padberg, 1976) เงื่อนไขแบบเซตประกันว่าจะต้องเลือกสมาชิกตัว

ใดตัวหนึ่งและเลือกได้ 1 ตัวเท่านั้น จากจำนวนทางเลือกทั้งหมดที่กำหนดให้ ซึ่งสามารถเขียนเงื่อนไขให้อยู่ในรูปของสมการได้ดังสมการที่ 2.1

$$\sum_{i=1}^n x_i \leq 1 \quad \text{โดย} \quad x_i \in \{0,1\} : i = 1,2,3,\dots,n \quad (2.1)$$

การแก้ปัญหาด้วยวิธีโลคอลเสิร์ช

วิธีการหาคำตอบหรือแก้ปัญหาคัดสินใจต่างๆ นั้นมีด้วยกันหลายวิธี ขึ้นอยู่กับเงื่อนไขและปัญหาว่าเหมาะสมที่จะใช้วิธีใดในการแก้ปัญหา วิธีโลคอลเสิร์ชซึ่งเป็นอีกวิธีหนึ่งของวิธีวิวิธวิธีที่สามารถหาคำตอบได้ดีเพียงพอกายในเวลาอันจำกัด โดยในแต่ละรอบการทำซ้ำ (Iteration) วิธีการหาคำตอบแบบเนเบอร์ฮูดจะทำการค้นหาคำตอบที่ดีขึ้นภายในเนเบอร์ฮูดที่กำหนด วิธีการหาคำตอบแบบเนเบอร์ฮูดจะเริ่มการค้นหาคำตอบจากคำตอบเริ่มต้นที่ไม่ขัดแย้งกับเงื่อนไข (Initial feasible solution) คำตอบเริ่มต้นสามารถเขียนแทนด้วยสัญลักษณ์ s^1 สำหรับแต่ละคำตอบ s (solution s) ที่ไม่ขัดแย้งเงื่อนไข กลุ่มคำตอบ (หรือจำนวนของคำตอบหลายๆ คำตอบ) ที่อยู่ภายในเนเบอร์ฮูดของ s เขียนแทนด้วยสัญลักษณ์ $N(s)$ ซึ่งกลุ่มคำตอบภายในเนเบอร์ฮูด s สามารถหาได้โดยการดัดแปลง เปลี่ยนแปลง หรือปรับปรุ้ง (Perturbation) คำตอบ s ตามกฎบางอย่างที่ผู้ออกแบบอัลกอริทึมตั้งขึ้นมา โดยกลุ่มคำตอบที่อยู่ภายใน $N(s)$ เรียกว่า “เนเบอร์ (Neighbor) ของ s ”

ต่อจากนั้นอัลกอริทึมจะทำการหาคำตอบแบบซ้ำไปซ้ำมาจนได้คำตอบ $s^1, s^2, s^3, \dots, s^{k-1}$ ที่เป็นคำตอบที่ไม่ขัดแย้งกับเงื่อนไข อัลกอริทึมจะให้คำตอบ s^{k-1} ซึ่งมีค่าฟังก์ชันวัตถุประสงค์ที่ดีกว่า (น้อยกว่า) คำตอบ s^k ที่มีอยู่ ณ เวลานั้น อัลกอริทึมจะหยุดหาคำตอบเมื่อไม่สามารถหาคำตอบที่ดีกว่าคำตอบที่มีอยู่ภายในเนเบอร์ฮูดนั้นได้ ซึ่งเรียกคำตอบนี้ว่า “Local optima” สำหรับปัญหาคัดสินใจที่ต้องการค่าสูงสุด เรียกว่า “Local maxima” และสำหรับปัญหาการหาค่าต่ำสุดเรียกว่า “Local minima”

หลังจากนั้น อัลกอริทึมจะเริ่มทำการหาคำตอบใหม่ที่มีจุดเริ่มต้นของคำตอบที่แตกต่างกัน (Different starting points) และดำเนินการหาคำตอบจนถึงระยะเวลาหรือรอบการกระทำซ้ำทั้งหมดที่กำหนด ซึ่งเราจะได้คำตอบที่เป็น Local optima เป็นจำนวนมาก Local optima ที่มีค่าน้อยที่สุดคือคำตอบสุดท้ายของวิธีนี้

สิ่งสำคัญในการหาคำตอบของปัญหาคัดสินใจโดยวิธีการหาคำตอบแบบเนเบอร์ฮูดคือการแบบโครงสร้างเนเบอร์ฮูด (Neighborhood หรือ Local move structure) ซึ่งเป็นการกำหนด

ว่าเนเบอร์ฮูด $N(s)$ จะนำไปสู่คำตอบ s ได้อย่างไร ซึ่งโครงสร้างเนเบอร์ฮูดนี้จะส่งผลโดยตรงต่อประสิทธิภาพของอัลกอริทึม

โดยหลักการแล้ว การใช้เนเบอร์ฮูดที่มีขนาดใหญ่จะทำให้เราสามารถหาคำตอบ (Local optima) ที่มีคุณภาพมากขึ้น อย่างไรก็ตาม การค้นหาคำตอบภายในเนเบอร์ฮูดที่มีขนาดใหญ่จะทำให้สูญเสียเวลาในการหาคำตอบมากด้วยเช่นกัน และส่งผลทำให้จำนวนครั้งของการเริ่มต้นการหาคำตอบภายในเนเบอร์ฮูดอันใหม่ลดลงตามไปด้วย ซึ่งทำให้อัลกอริทึมสูญเสียโอกาสในการค้นหาคำตอบภายในบริเวณอื่นซึ่งอาจจะมีคำตอบที่ดีมากกว่าอยู่ในบริเวณนั้นก็ได้

การแก้ปัญหาโดยวิธีวิวิธติสต์ และเมตาวิวิธติสต์

Holland (1975) นำเสนออัลกอริทึมสำหรับแก้ปัญหาที่ต้องการคำตอบที่ดีที่สุด (Optimization problem) มาจากกระบวนการเลียนแบบวิวัฒนาการของสิ่งมีชีวิตที่เกิดขึ้นในธรรมชาติโดยเกี่ยวข้องกัน โดย Genetic Algorithm อยู่บนพื้นฐานการพัฒนาการของสิ่งมีชีวิตตามหลักของธรรมชาติที่ว่าด้วย ทฤษฎีการคัดเลือกตามธรรมชาติ และทฤษฎีการสืบทอดลักษณะทางพันธุกรรมของยีนโดยการนำค่าของตัวแปรต่าง ๆ ที่มีผลต่อกำหนดค่าใช้เข้ามาแปลงให้อยู่ในรูปของโครโมโซม ซึ่งอาจจะแสดงในรูปของ binary bit string

Chu and Beasley (1997) นำเสนอวิธีเชิงพันธุกรรมในการแก้ปัญหา GAP โดยการเริ่มปรับปัญหาให้เป็นตามโครงสร้างของโครโมโซม จากนั้นวัดค่าความเหมาะสมของแต่ละโครโมโซม โครโมโซมทุกตัวถูกนำมาเข้าสู่ขั้นตอน Selection เพื่อคัดเลือกเข้าสู่กระบวนการ Genetic Operator โดยทำการคัดเลือกเฉพาะโครโมโซมที่มีค่าความเหมาะสม (Fitness Function) เพื่อที่จะได้โครโมโซมที่ยอมรับได้ตามที่ได้กำหนดเอาไว้ จากนั้นโครโมโซมที่คัดเลือกไว้จะเข้าสู่กระบวนการข้ามสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) ได้เป็นโครโมโซมชุดใหม่ ซึ่งเราจะนำโครโมโซมชุดใหม่นี้มาดำเนินการต่อไปจนสิ้นสุดตามเงื่อนไขที่ได้กำหนดไว้ ก็จะได้โครโมโซมที่มีค่าความเหมาะสมที่ดีที่สุด เป็นคำตอบของปัญหา

Diaz and Fernandez (2001) เสนอการแก้ปัญหา GAP โดยวิธีทาบูลาร์ สำหรับวิธีนี้ อัลกอริทึมจะยอมให้ค้นหาคำตอบทั้งที่เป็นไปได้ และคำตอบที่เกินไปได้ เพื่อเพิ่มโอกาสในการค้นหาคำตอบที่ดีที่สุด โดยเทคนิคการกระจายพื้นที่ในการค้นหาคำตอบ (Diversification) และการพยายามค้นหาคำตอบในพื้นที่ที่น่าจะมีคำตอบที่ดีที่สุดอยู่ (Intensification) งานวิจัยนี้ใช้การเก็บหน่วยความจำแบบถี่ (Frequency-based memory)

วิธีพันธุ์ผสม

Osman (1995) เสนอวิธีพันธุ์ผสมโดยนำวิธี SA ทำงานร่วมกับวิธีทาบู โดยอัลกอริทึมจะใช้วิธี SA สำหรับสร้างเนเบอร์ฮูดใหม่ เพื่อเป็นการกระจายพื้นที่ในการค้นหาคำตอบให้สามารถทำการค้นหาคำตอบในพื้นที่อื่น ๆ ได้เพิ่มมากขึ้น ในงานวิจัยนี้ได้ใช้หน่วยความจำแบบความถี่ในการจัดเก็บข้อมูล และใช้สำหรับป้องกันการค้นหาคำตอบในพื้นที่ที่ไม่น่ามีคำตอบที่ดีอยู่

Yagiura (1999) เสนอวิธี variable depth search สำหรับแก้ปัญหา GAP เป็นอัลกอริทึมที่ทำการปรับปรุงคำตอบโดยการเคลื่อนย้าย (Shift) และการสลับค่า (Swap) ในเนเบอร์ฮูด โดยที่ห้ามทำการเคลื่อนย้ายหรือสลับค่าไปยังคำตอบที่ต้องห้ามเพื่อป้องกันการเกิด Local minima ซึ่งอัลกอริทึมจะยอมให้ค้นพบคำตอบที่เป็นไปไม่ได้ได้ แต่จะทำการปรับปรุงคุณภาพคำตอบให้ได้คำตอบที่เป็นไปได้ และมีคุณภาพมากยิ่งขึ้น

Lourenço and Serra (2002) เสนอวิธี greedy randomized ร่วมกับวิธี MAX-MIN ant system ซึ่งพัฒนามาจากวิธี ant colony ในการแก้ปัญหา GAP โดยอัลกอริทึมทั้งสองเป็นวิธีเมต้าฮิวริสติกส์ โดยอัลกอริทึมจะสร้างคำตอบเริ่มต้นซึ่งพิจารณาจากคำตอบของรอบการทำงานที่ผ่านมา ในงานวิจัยนี้จะหาความน่าจะเป็นของผู้รับผิดชอบงาน ว่าผู้รับผิดชอบงานรายใดสามารถดำเนินงานนั้น ๆ ได้สูงสุด จากนั้นจะใช้วิธีโลคอลเสิร์ชและทาบูในการพัฒนาคุณภาพคำตอบให้ดียิ่งขึ้น

สรุป

วิธีการที่นำเสนอนี้ เป็นการปรับปรุงประสิทธิภาพในการค้นหาคำตอบของวิธีฮิวริสติกส์ (นั่นคือวิธี C.L.S) สำหรับการแก้ปัญหาการตัดสินใจที่มีความซับซ้อน โดยพัฒนาวิธีการเรียนรู้และการพยากรณ์ล่วงหน้าเพื่อลดการตัดสินใจ เพื่อบอกถึงทิศทางของคำตอบให้ทำการค้นหาคำตอบในพื้นที่ที่น่าจะมีคำตอบที่ดีที่สุดอยู่ ทั้งนี้วิธีดังกล่าวมีความยืดหยุ่นในการประยุกต์ใช้กับวิธีฮิวริสติกส์อื่น ๆ สำหรับแก้ปัญหาการตัดสินใจ

ในการแก้ปัญหา GAP ด้วยวิธีฮิวริสติกส์และเมต้าฮิวริสติกส์ หากพิจารณาจากเทคนิค diversification และ intensification แล้ว วิธีการที่นำเสนอจะคล้ายกับ Diaz and Fernandez (2001): Osman (1995) และ Yagiura (1999) ซึ่งแต่ละวิธีการแก้ปัญหานั้นจะยอมรับคำตอบที่เป็นไปไม่ได้เพื่อให้เกิดการกระจายพื้นที่ในการค้นหาคำตอบ ป้องกันไม่ให้เกิด local optima จากนั้นจึงทำการปรับปรุงคุณภาพคำตอบให้ใกล้เคียง หรือได้คำตอบที่ดีที่สุดสำหรับปัญหานั้น ๆ ซึ่งจะแตกต่างกับ Holland (1975): Chu and Beasley (1997) และ Lourenço and Serra (2002) เนื่องจากอัลกอริทึมแต่ละวิธีจะเริ่มต้นด้วยการพยายามค้นหาคำตอบที่เป็นไปได้ก่อน จากนั้นจึงจะพัฒนาคำตอบที่ได้ให้มีคุณภาพดียิ่งขึ้น

สำหรับเทคนิคการเรียนรู้และการพยากรณ์ค่าตัวแปรการตัดสินใจในงานวิจัยนี้พัฒนาขึ้นเพื่อสร้างความเสถียรภาพในการค้นหาคำตอบ ให้กับวิธีโลคอลเสิร์ช โดยที่จะมีความแตกต่างกับวิธีโลคอลเสิร์ชอื่น ๆ คือ มีความยืดหยุ่นในการเพิ่ม หรือลดเงื่อนไข และหากค่าพยากรณ์ที่ได้มีความถูกต้องจะช่วยในการลดพื้นที่ในการค้นหาคำตอบ และลดจำนวนรอบในการทำงานได้อย่างรวดเร็ว