หอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ปัญหาการจัดเส้นทางการขนส่งเวชภัณฑ์ในระบบการกระจายเวชภัณฑ์ของโรงพยาบาล จัดเป็นปัญหารูปแบบการหาค่าเหมาะสมที่สุดเชิงการจัด (Combinatorial Optimization) หรือการหาค่าเหมาะสมที่สุดหลายระดับ (Multilevel Optimization) ที่ขยายขอบเขตและเพิ่มความซับซ้อนขึ้นจากปัญหาการจัดเส้นทางเดินรถพื้นฐาน (Conventional Vehicle Routing Problem: VRP) และมีความซับซ้อนของการคำนวณในระดับเอ็นพีแบบยาก (Nonpolynomial Hard: NP-Hard) ซึ่งใช้เวลาในการหาคำตอบยาวนาน และเมื่อขนาดของปัญหาเพิ่มขึ้นก็จะใช้เวลาในการหาคำตอบเพิ่มขึ้นอย่างเอ็กโปเนนเชียล [1,2] ทำให้การหาคำตอบที่ดีที่สุดด้วยวิธีการคำนวณตัวแบบทางคณิตศาสตร์ใช้เวลาในการหาคำตอบที่ยาวนานเกินไป และจำเป็นต้องอาศัยวิธีการหาคำตอบแบบฮิวริสติก (Heuristic Approach) สำหรับการแก้ไขปัญหา ในบทนี้จึงได้แบ่งเนื้อหาออกเป็น 3 เรื่อง ได้แก่ ทฤษฎีที่เกี่ยวข้องกับปัญหาการจัดเส้นทางเดินรถ วิธีการหาคำตอบสำหรับปัญหาการจัดเส้นทางเดินรถ และผลงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้องกับปัญหาการจัดเส้นทางเดินรถ

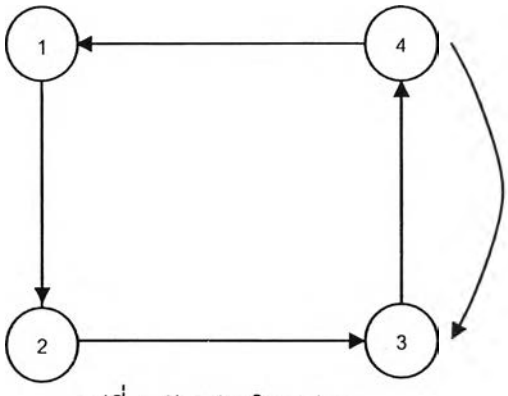
2.1.1 การวิเคราะห์โครงข่าย (Network Analysis)

เพื่อให้สามารถเข้าใจและวิเคราะห์ปัญหาได้ง่ายขึ้น ปัญหาการจัดเส้นทางเดินรถสามารถพิจารณาได้ในรูปแบบของการวิเคราะห์โครงข่ายประเภทหนึ่ง โดยโครงข่าย (Network) แสดงได้ในระบบขององค์ประกอบ 2 ส่วน คือ โนด (Node) และ อาค (Arc) เมื่อ

1) *โนด* คือ จุดยอดหรือจุดมุมของรูปหลายเหลี่ยมเมื่อแสดงในโครงข่ายในลักษณะของกราฟ และสำหรับปัญหาการจัดเส้นทางเดินรถ โนดจะหมายถึง จุดเชื่อม หรือจุดที่ใช้ในการขนส่งสินค้าโดยสามารถเป็นได้ทั้งจุดเริ่มต้นหรือจุดสุดท้ายของการขนส่ง
2) *อาค* คือ เส้นเชื่อม หมายถึง เส้นทางการขนส่งสินค้าหรือแสดงความสัมพันธ์ระหว่างจุดเชื่อมใดๆ อาจแสดงได้ทั้งระยะทาง ระยะเวลาเดินทางหรือค่าใช้จ่ายในการเดินทาง อาคมีทิศทางเดียวหรือสองทิศทางก็ได้ โดยอาคที่ต่อกับโนดอย่างเป็นลำดับเรียกว่า เชน (Chain) และเรียกเชนที่อาคมีลำดับของการต่อโนดที่โนด

ปลายเป็นโนดเริ่มต้นของอาคถัดไปว่า พาธ (Path) หรือเส้นทาง และเรียก อาค ที่ไม่มีทิศทางเรียกว่า เอ็ดจ์ (Edge)

ตัวอย่างของโครงข่ายแสดงดังรูปที่ 1



รูปที่ 1 ตัวอย่างโครงข่าย

การหาเส้นเชื่อมให้มีค่าประจำอาครวมน้อยที่สุด (Shortest Path Problem: SPP) จะมีความสัมพันธ์กับระยะทาง ระยะเวลาเดินทางหรือค่าใช้จ่ายที่น้อยที่สุดได้ ซึ่งตัวอย่างวิธีการแก้ปัญหาโครงข่ายโดยหาระยะทางที่น้อยที่สุดระหว่างจุดเชื่อม เช่น วิธีการของ Dijkstra (Dijkstra's Algorithm) [1]

2.1.2 ความหลากหลายของปัญหาการจัดเส้นทางเดินรถ

ปัญหาการจัดเส้นทางเดินรถได้รับการศึกษาอย่างมากในช่วงสามถึงสี่ทศวรรษที่ผ่านมาและมีผลงานด้านการวิจัยดำเนินงาน (Operation Research) ที่เกี่ยวข้องกับปัญหาการจัดเส้นทางเดินรถไว้หลากหลายรูปแบบ โดยปัญหาการจัดเส้นทางที่มีการศึกษาและรู้จักกันอย่างกว้างขวางที่สุด ได้แก่ ปัญหา Traveling Salesman Problem (TSP) ซึ่งเป็นรูปแบบปัญหาการเดินทางของพนักงานขาย 1 คน ที่มีเงื่อนไขว่าพนักงานขายจะต้องเดินทางผ่านทุกเมืองโดยไม่วนซ้ำ และเดินทางกลับมายังจุดเริ่มต้นให้มีระยะทางสั้นที่สุด [2,3,4] ซึ่งปัญหาที่มีรูปแบบคล้ายกัน ได้แก่ ปัญหาการจัดเส้นทางเดินรถแบบ Vehicle Routing Problem (VRP) ซึ่งทำการหาเส้นทางเดินรถหลายเส้นทางเริ่มจากคลังสินค้าและเดินทางไปหาลูกค้าในที่ต่างๆ โดยผ่านลูกค้าแต่ละจุดเพียงครั้งเดียว และปริมาณความต้องการรับบริการของลูกค้าสำหรับแต่ละเส้นทางเดินรถจะต้องไม่เกินความจุของพาหนะขนส่ง มีจุดประสงค์เพื่อให้เสียค่าใช้จ่ายในการเดินทางรวมน้อยที่สุด [2,3] เมื่อระบบที่ศึกษามีความไม่แน่นอนจะทำให้เกิดปัญหาแบบ Stochastic Vehicle Routing Problem (SVRP) ซึ่งเป็นรูปแบบปัญหาการจัดเส้นทางเดินรถเมื่อปัจจัยใดๆ ในระบบมีความไม่แน่นอน อาทิ ระยะเวลาเดินทาง ความต้องการของลูกค้า หรือ ความไม่แน่นอน

ของลูกค้า จัดเป็นปัญหาประเภท A-Priori Optimization Based Method [2] ซึ่งสามารถแบ่งย่อยปัญหาออกได้อีกหลายประเภท โดยรูปแบบของปัญหา SVRP ที่ได้รับความสนใจศึกษามากที่สุด คือ รูปแบบปัญหาที่มีความต้องการของลูกค้าไม่แน่นอน Vehicle Routing Problem with Stochastic Demand (VRPSD) [5,2] ส่วนปัญหา SVRP แบบอื่นๆ ได้แก่ Probabilistic Traveling Salesman Problem (PTSP) หรือ TSP with Stochastic Customer (TSPSC) เป็นรูปแบบปัญหาที่ปรับแนวคิดจากกลุ่มของเมืองที่พนักงานขายจะเดินทางไปให้มีความน่าจะเป็นเท่ากับ p_i และจะทำการวางแผนการเดินทางไปตามข้อมูลความน่าจะเป็น โดยหากไม่มีลูกค้าที่เมืองใด ก็จะเดินทางข้ามเมืองนั้นๆ [2,6] Traveling Salesman Problem with Stochastic Travel Time (TSPST) เป็นรูปแบบปัญหาที่ตำแหน่งของลูกค้าทราบแน่นอนแต่ระยะเวลาเดินทาง หรือค่าประจำอาคเป็นตัวแปรสุ่ม โดยทั่วไปจุดประสงค์ในการจัดเส้นทางเดินรถเพื่อทำให้ค่าความน่าจะเป็นของการเดินทางครบรอบภายในกำหนดมีค่ามากที่สุด โดยปัญหาสามารถขยายขอบเขตไปสู่ m-TSP with Stochastic Travel Time (m-TSPST) [2,3] นอกจากนี้ Probabilistic Vehicle Routing Problem (PVRP) หรือ PVRP with Stochastic Customer and Demand (VRPSCD) เป็นรูปแบบปัญหาที่ปรับแนวคิดจากทั้งความต้องการและลูกค้าที่ทราบล่วงหน้าแน่นอน (Deterministic) ให้มีลักษณะของความน่าจะเป็น [2] สำหรับปัญหาการจัดเส้นทางเดินรถที่ระบบไม่คงที่ Dynamic Traveling Salesman Problem (DTSP) เป็นปัญหาที่จัดอยู่ในประเภท Real-Time Optimization Method โดยปรับปรุงแนวคิดเดิมของ TSP ซึ่งระบบคงที่ (Static) โดยให้ความต้องการเป็นอิสระขึ้นอยู่กับลูกค้าพนักงานขายเดินทางไปด้วยความเร็วคงที่เพื่อให้บริการลูกค้าที่จุดต่างๆ และต้องมีการกำหนดนโยบายในการให้บริการลูกค้าเพื่อการจัดเส้นทาง เช่น First Come First Serve (FCFS), Stochastic Queue Median (FCFS-SQM), Nearest Neighbor (NN), Traveling Salesman Problem (TSP), Space Filling Curve (SFC), Partitioning Policy (PART) [2] สำหรับความไม่คงที่ของระยะเวลาเดินทางระหว่างคู่ของจุดเชื่อม Time-Dependent TSP and VRP (TD-TSP and TD-VRP) เป็นรูปแบบปัญหาที่ระยะเวลาเดินทางระหว่างจุดเชื่อมที่เป็นลูกค้าหรือระหว่างจุดเชื่อมที่เป็นคลังสินค้าและลูกค้าเปลี่ยนไปตามช่วงเวลา ตัวอย่างเช่น ความเร็วของรถขนส่งไม่คงที่เนื่องจากความหนาแน่นของเส้นทาง (Traffic Congestion) โดยปัจจัยอื่นๆ คงที่และทราบล่วงหน้าแน่นอน [2,7]

นอกจากนี้สำหรับปัญหาการจัดเส้นทางเดินรถที่พิจารณาแง่มุมเฉพาะ ด้านตัวสินค้าหรือพาหนะขนส่งที่จะทำการจัดเส้นทาง อาทิ Fleet Size and Mix Vehicle Routing

Problem (FSMVPR) เป็นรูปแบบปัญหาที่ทำการจัดเส้นทางเดินรถโดยมีชนิดของสินค้าหลายแบบซึ่งมีขนาดไม่เท่ากัน และพาหนะที่ใช้ในการขนส่งมีหลายประเภทซึ่งมีค่าใช้จ่ายแตกต่างกัน โดยทำการจัดเส้นทางให้เสียค่าใช้จ่ายน้อยที่สุด [8] Truck and Trailer Vehicle Routing Problem (TTVRP) พิจารณาพาหนะที่ใช้ในการขนส่งและความสามารถของเส้นทางร่วมกัน โดยเส้นทางแบ่งออกเป็น 3 ประเภทคือ Pure Truck Route, Pure Vehicle Route without Sub Tour และ Complete Vehicle Route [9] Multimodal Shortest Variable Path Problem (MSVPP) เป็นรูปแบบปัญหาที่เลือกเส้นทางในการวิเคราะห์โครงข่ายที่สั้นที่สุด โดยมีข้อแตกต่างคือ เส้นเชื่อมที่เชื่อมระหว่างจุดเชื่อมคู่ใดๆ จะมีมากกว่า 1 เส้น และการเลือกเส้นเชื่อมใดๆ จะให้ค่าของระยะเวลาที่ใช้ในการเดินทางและค่าใช้จ่ายไม่เท่ากัน [10] นอกจากนี้ Traveling Salesman Problem with Backhaul (TSPB) เป็นรูปแบบหนึ่งของปัญหา The Traveling Salesman Problem with Precedence Constraint (TSPPC) ซึ่งมีข้อกำหนดเฉพาะสำหรับการจัดเส้นทางเดินรถของจุดเชื่อม ซึ่งสำหรับ TSPB ทำการจัดเส้นทางเดินรถโดยคำนึงถึงลำดับของลูกค้าโดยเพิ่มข้อกำหนดให้แก่เส้นทางตามลำดับก่อนหลัง ซึ่งลักษณะของปัญหา เช่น การขนส่งสินค้าจากแหล่งผลิตไปยังตลาดสินค้า [11]

สำหรับแง่มุมทางด้านกรอบเวลาในการจัดเส้นทางเดินรถ Vehicle Routing Problem with Time Window (VRPTW) เป็นรูปแบบปัญหาที่มีข้อจำกัดด้านระยะเวลาในการรับบริการของลูกค้า ณ จุดเชื่อม หรือข้อกำหนดด้านระยะเวลาสินค้า หรือการให้บริการต้องเริ่มและเสร็จสิ้นภายในช่วงเวลาที่กำหนด โดยวัตถุประสงค์เพื่อให้ได้ค่าใช้จ่ายน้อยที่สุดและไม่เกิดงานที่เกินเวลา (Tardy Customer) หรือใช้ระยะเวลาเดินทางมากเกินไป [2,12] และปัญหาการจัดเส้นทางเดินรถที่พิจารณาข้อจำกัดด้านเวลาและความไม่คงที่ของระบบร่วมกัน คือ Dial-A-Ride Problem (DARP) ซึ่งเป็นรูปแบบปัญหาที่มีความต้องการเพื่อไปรับจากตำแหน่งเริ่มต้นและนำไปส่งในตำแหน่งเป้าหมาย โดยมีช่วงเวลาสำหรับการรับและส่งที่จำกัด [2,6] สำหรับปัญหาที่มีความผสมผสานระหว่างปัญหาการจัดเส้นทางเดินรถและปัญหาแบบ Assignment Problem เช่น Period Vehicle Routing Problem (PVRP) พิจารณาเส้นทางสำหรับช่วงระยะเวลาใดๆ และความต้องการของลูกค้าจะมีลักษณะซ้ำๆ กันในช่วงระยะเวลาที่พิจารณา [13,14] รูปแบบปัญหาการจัดเส้นทางเดินรถที่แตกต่างออกไป เช่น Chinese Postman Problem (CPP) เป็นรูปแบบปัญหาที่หาระยะทางที่สั้นที่สุดซึ่งจะต้องผ่านเส้นเชื่อมระหว่างจุดเชื่อมต่างๆ อย่างน้อย 1 ครั้ง และคำนึงถึงทิศทางของเส้นเชื่อม โดยแบ่งออกได้เป็น 3 ประเภทตามทิศทาง คือ Undirected CPP, Directed CPP และ Mixed CPP [3,15] Asymmetric

Traveling Salesman Problem (ATSP) เป็นรูปแบบของปัญหาการจัดเส้นทางเดินรถ ที่เส้นเชื่อมระหว่างคู่ของจุดเชื่อมใดๆ มีทิศทางของเส้นเชื่อม 2 ทิศทาง (Bi-Directional Distance) และค่าประจำเส้นเชื่อมไม่จำเป็นต้องเท่ากันเสมอไป [16]

2.2 วิธีการหาคำตอบสำหรับปัญหาการจัดเส้นทางเดินรถ

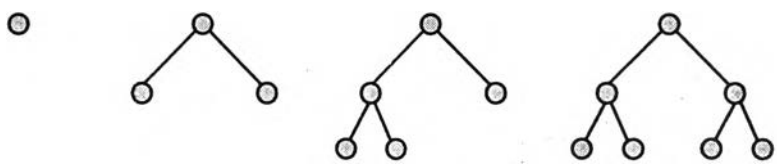
การหาคำตอบสำหรับปัญหาการจัดเส้นทางเดินรถแบ่งออกเป็น 2 ประเภท ได้แก่ วิธีการหาคำตอบแบบ Exact Optimization และ Heuristic Optimization ซึ่งความแตกต่างของทั้งสองวิธีนี้จะอยู่ที่ผลลัพธ์ของวิธีการหาคำตอบ โดยสำหรับ Exact Optimization นั้นคำตอบจะอยู่ในรูปของค่าที่ดีที่สุด (Mathematical Optima) ซึ่งจำเป็นต้องสร้างตัวแบบทางคณิตศาสตร์สำหรับการแก้ปัญหา ในขณะที่วิธีการแบบ Heuristic Optimization จะได้คำตอบในรูปของค่าใกล้เคียงค่าที่ดีที่สุด (Approximate Optima) แต่เนื่องจากรูปแบบของปัญหาในทางปฏิบัติจะมีขนาดใหญ่และมีความซับซ้อนมากกว่าลักษณะปัญหาในเชิงทฤษฎีที่มีลักษณะแบบ Formal Framework จึงทำให้การแก้ไขปัญหาในทางปฏิบัติจำเป็นต้องอาศัยการประยุกต์ใช้วิธีการแก้ปัญหาแบบ Heuristic Optimization ในการหาคำตอบ [17]

2.2.1 วิธีการหาคำตอบที่ดีที่สุด (Exact Optimization)

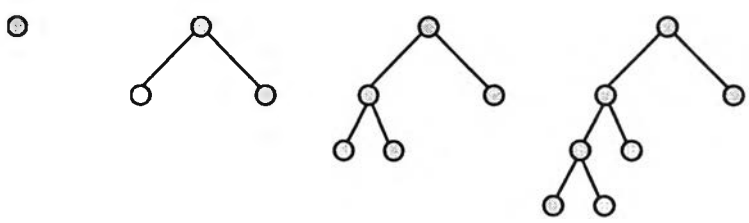
วิธีการหาคำตอบที่ดีที่สุดสำหรับปัญหาการจัดเส้นทางเดินรถแบ่งออกเป็น 3 แบบ ได้แก่ การโปรแกรมเชิงเส้นตรง (Linear Programming: LP) เป็นการประยุกต์ใช้วิธีการแก้ไขปัญหาทางการจัดสรรทรัพยากรที่มีลักษณะความสัมพันธ์ของตัวแปรต่างๆ ที่เกี่ยวข้องเป็นเชิงเส้นตรงทั้งหมดเพื่อทำการกำหนดเส้นทางเดินรถ โดยมีจุดมุ่งหมายเพื่อแก้ปัญหาและตัดสินใจให้ได้ค่าของฟังก์ชันเป้าหมายที่ดีที่สุด การโปรแกรมแบบไดนามิก (Dynamic Programming) เป็นวิธีการหาค่าที่ดีที่สุดโดยแบ่งกลุ่มของจุดเชื่อมที่พิจารณาจัดเส้นทางออกเป็น 2 กลุ่ม คือ กลุ่มที่จัดเส้นทางไปแล้วและกลุ่มที่ยังไม่ได้จัดเส้นทาง จากนั้นพิจารณาเส้นทางที่ดีที่สุดทุกครั้งที่เลือกเชื่อมจุดเชื่อมหนึ่งจุดใดๆ เข้ากับกลุ่มของจุดเชื่อมที่จัดเส้นทางไปแล้ว เทคนิคการแตกกิ่งและจำกัดขอบเขตของตัวแปร (Branch and Bound Technique) ใช้วิธีการจำกัดค่าเมตริกซ์ ในการคำนวณหาขอบเขต และใช้เมตริกซ์ที่ถูกจำกัดค่าในการแตกกิ่งของเส้นทางเดินจาก 1 กลุ่มย่อยไปเป็น 2 กลุ่มย่อย [3]

2.2.2 วิธีการค้นหาคำตอบ (Searching Approach)

เมื่อวิธีการทางคณิตศาสตร์ไม่สามารถใช้ในการแก้ปัญหาที่มีขนาดใหญ่และมีความซับซ้อนมากๆ ได้ จึงมีการคิดค้นวิธีการหาคำตอบที่เหมาะสมที่สุดสำหรับสำหรับปัญหาแบบ Combinatorial Optimization และวิธีการหนึ่งที่นำมาใช้เรียกว่า "วิธีการค้นหาคำตอบ" ซึ่งสามารถแบ่งออกเป็น 2 ประเภท ได้แก่ Uninformed Search หรือ Blind Search เป็นการหาคำตอบโดยไม่มีข้อมูลเกี่ยวกับจำนวนขั้นตอนหรือ Path Cost จากสภาวะปัจจุบัน (Current State) ไปสู่สภาวะเป้าหมาย (Goal State) ตัวอย่างของ Uninformed Search ได้แก่ Bread-First Search (BFS) มีวิธีขยายการค้นหาคำตอบจากเป้าหมายที่อยู่ใกล้ที่สุดก่อน ดังแสดงในรูปที่ 2 ส่วนมากแล้ว BFS เป็นวิธีการค้นหาคำตอบที่สมบูรณ์และให้ค่าที่เหมาะสมที่สุด แต่ใช้หน่วยความจำสูง Depth-First Search (DFS) มีวิธีขยายการค้นหาคำตอบไปยังส่วนที่อยู่ลึกที่สุดของแผนผังต้นไม้ (Tree Diagram) ดังแสดงในรูปที่ 3 เป็นวิธีการค้นหาคำตอบที่ไม่สมบูรณ์และไม่ได้ให้ค่าที่เหมาะสมที่สุด แต่ใช้หน่วยความจำน้อยกว่าเมื่อเปรียบเทียบกับวิธีการ BFS [18]



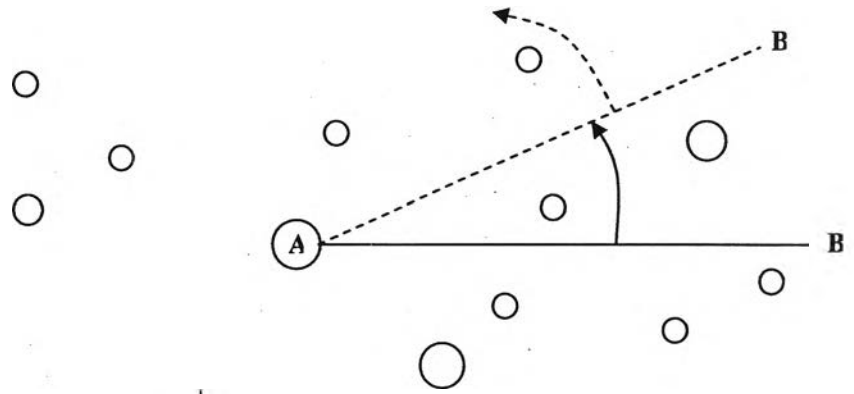
รูปที่ 2 วิธีการขยายการค้นหาคำตอบของ Bread-First Search



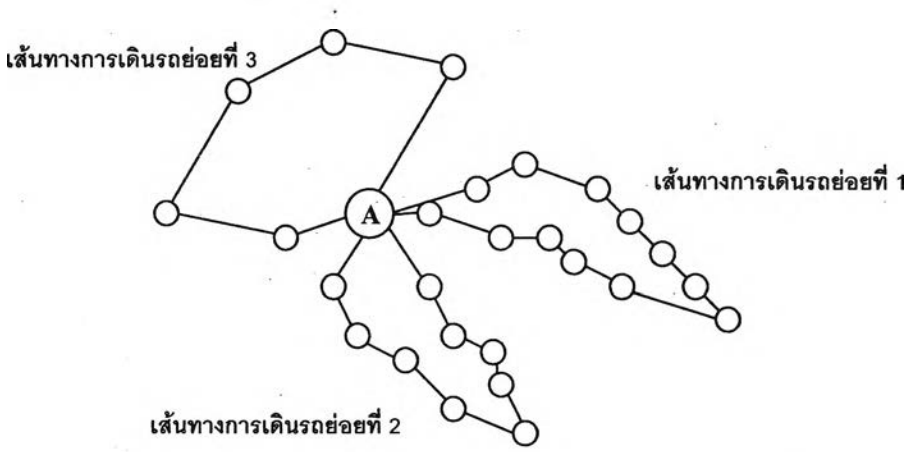
รูปที่ 3 วิธีการขยายการค้นหาคำตอบของ Depth-First Search

สำหรับ Informed Search หรือ Heuristic Search จะอาศัยข้อมูลในการเปลี่ยนจากสภาวะปัจจุบันไปสู่สภาวะเป้าหมายซึ่ง Informed Search จะมีประสิทธิภาพมากกว่า Uninformed Search และมีอยู่ด้วยกันหลากหลายวิธี ได้แก่ Sweep Approach เป็นวิธีการที่กำหนดจุดเชื่อมและบอกลำดับที่ต้องเดินทางผ่านจุดต่างๆ โดยกำหนดตำแหน่งที่ตั้งของ

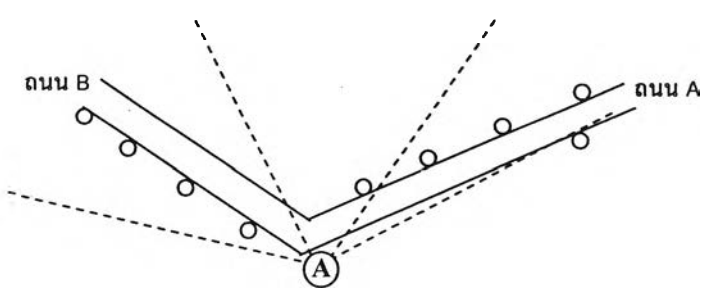
จุดเชื่อมเป็นพิกัดเชิงมุม (Polar Coordinate) ตัวอย่างวิธีการแสดงดังรูปที่ 4 ให้คลังสินค้าอยู่ตรงกลางเป็นจุด A ทำการเลือกจุดเริ่มต้นแบบสุ่มและกวาดแขน AB โดยหมุนทวนหรือตามเข็มนาฬิกา จากคลังไปยังจุดต่างๆเพื่อตอบสนองความต้องการของแต่ละจุด โดยข้อบกพร่องของวิธีการนี้คือในกรณีที่คลังสินค้าไม่ได้อยู่ที่จุดศูนย์กลางของพื้นที่ จะทำให้ได้เส้นทางมีขนาดไม่สมดุล ดังรูปที่ 5 แสดงให้เห็นว่าวิธีนี้ไม่สามารถจ่ายงานให้รถแต่ละคันได้อย่างสมดุล และวิธีการนี้ไม่ได้คำนึงถึงถนนทำให้จุดที่ใกล้เคียงกันที่อยู่บนถนนเส้นเดียวกันอาจไม่ได้อยู่ในเส้นทางเดียวกัน ดังรูปที่ 6 [3,8]



รูปที่ 4 ตัวอย่างวิธีการ Sweep Approach



รูปที่ 5 รูปร่างที่ไม่สมดุลของเส้นทางการเดินรถย่อย



รูปที่ 6 ผลกระทบของถนนต่อวิธีการ Sweep Approach

Saving Algorithm เป็นวิธีการที่สามารถใช้กับปัญหาที่มีจุดขนส่งจำนวนมากได้ โดยหาเส้นทางเดินรถที่จะให้ค่าเป้าหมายมากที่สุด เช่น ค่าใช้จ่ายหรือค่าปรับน้อยที่สุดสำหรับการเดินทางไปยังจุดเชื่อมต่างๆ โดยทำการเปรียบเทียบค่าเป้าหมายของเส้นทางเดินรถแบบ m เส้นทางและค่าเป้าหมายสำหรับเส้นทางเดินรถที่ทำการลดจำนวนเส้นทางลงเหลือ m - i เส้นทาง เมื่อ $m > i \geq 1$ [3] Decomposition Heuristic วิธีนี้เป็นการลดขนาดของปัญหาโดยแบ่งเป็นปัญหาเล็กๆ และปัญหาเล็กๆ เหล่านี้จะถูกแก้โดยใช้เวลาที่น้อยกว่าการแก้ปัญหาขนาดใหญ่ เป็นการแบ่งปัญหาออกเป็นส่วนๆ และแก้ปัญหาแต่ละส่วนแยกจากกันซึ่งมีวิธีการแก้ปัญหาหลายแบบ ได้แก่ Iteration ซึ่งจะหาคำตอบสำหรับปัญหาย่อยโดยคงค่าของตัวแปรอื่นๆ และปรับปรุงค่าของตัวแปรตัดสินใจสำหรับชุดคำตอบย่อยนั้นๆ Column Generation เป็นวิธีที่เหมาะสมเมื่อปัญหามีแง่มุมที่แสดงในรูปแบบทางคณิตศาสตร์ได้ยาก โดยสำหรับปัญหาการจัดเส้นทางเดินรถการปรับปรุงคำตอบอาจเลือกใช้วิธีการ K-OPT โดยที่ K เป็นจำนวนเต็ม ทำการลดเส้นเชื่อมในเส้นทางออก K เส้น วิธีการนี้เหมาะสำหรับใช้แก้ปัญหาขนาดใหญ่และใช้เวลาไม่มากนัก กรณีเลือกใช้ K-OPT ที่มีค่า K เป็นเลขจำนวนเต็มที่มีค่ามากๆ จะให้ผลลัพธ์ที่ดีกว่า แต่จะใช้เวลานานกว่า [3,17]

Constructive Search แก้ปัญหาโดยอาศัย Null Solution และทำการแก้ปัญหาแบบครั้งต่อครั้งโดยอาศัยข้อมูลจากการแก้ปัญหาครั้งก่อน เพื่อให้ได้คำตอบของปัญหา (Full Solution) โดยการหาคำตอบของวิธี Constructive Search มีลักษณะที่คำนึงผลประโยชน์เฉพาะหน้า (Greedy และ Myopic) ซึ่งข้อสังเกตหนึ่งของวิธีการนี้คือคำตอบที่ได้จะขึ้นอยู่กับลำดับของการแก้ปัญหา ดังนั้นหากการแก้ปัญหาครั้งก่อนหน้าให้คำตอบที่ไม่ดีก็จะลดค่า (Degrade) ของคำตอบในครั้งถัดไป [17] โดยตัวอย่างของ Constructive Search ได้แก่ Greedy Search ซึ่งมีวิธีการขยายการค้นหาคำตอบจากคำตอบที่ดีที่สุดในปัจจุบันไปยังคำตอบใหม่ โดยประเมินค่าใช้จ่ายในการขยายจากสภาวะเฉพาะ (Particular State) ไปยังสภาวะเป้าหมาย ฟังก์ชันในการคำนวณตัวประเมินค่าใช้จ่ายนี้เรียกว่า Heuristic Function ที่สามารถสร้างเป็นฟังก์ชันความสัมพันธ์ใดๆ โดยต้องการให้ $h(n) = 0$ เมื่อ n คือสภาวะเป้าหมาย และถึงแม้ว่า Greedy Search จะให้ค่า ใช้จ่ายต่ำที่สุด (Minimal Search Cost) แต่เป็นวิธีการที่ไม่มีการขยายการค้นหาคำตอบไปในแนวทางอื่นอีกและไม่ประกันว่าจะได้ค่าที่เหมาะสมที่สุด วิธีการของ Greedy Search สามารถหาคำตอบได้อย่างรวดเร็ว แต่จุดเริ่มต้นที่ไม่ดีจะมีผลต่อวิธีการ Greedy Search โดยทำให้เกิดการขยายการค้นหาคำตอบไปยังส่วนที่ไม่จำเป็น และถ้าไม่มีการป้องกันการทำซ้ำก็จะไม่สามารถหาคำตอบได้ [18] สำหรับ Admission Search (A Search) จะสร้างฟังก์ชันการประเมินค่า (Evaluation Function) ขึ้นมาใหม่ซึ่งเป็นการรวมข้อดีของทั้ง Greedy Search กับ Uniformed-Cost Search เนื่องจาก Greedy Search ใช้

หลักการหาค่าใช้จ่ายคาดหมายต่ำที่สุด (Minimize Estimate Cost) ไปยังสภาวะเป้าหมาย โดยใช้ Heuristic Function เป็นวิธีการค้นหาคำตอบที่ไม่สมบูรณ์และไม่ได้ให้ค่าที่เหมาะสมที่สุด แต่ Uniformed-Cost Search ใช้หลักการหาค่าใช้จ่ายประจำพาธต่ำที่สุด (Minimize The Cost of Path), *g(n)* เป็นวิธีการที่สมบูรณ์และให้ค่าตอบที่เหมาะสมที่สุดแต่ไม่มีประสิทธิภาพ ดังนั้น A search จึงสร้างฟังก์ชันการประเมินค่า (Evaluation Function) ขึ้นมาใหม่ ดังสมการ

$$f(n) = g(n) + h(n)$$

เมื่อ *g(n)* คือ ค่าใช้จ่ายประจำพาธ (Path Cost) จากโนดเริ่มต้นไปยังโนดที่ n

h(n) คือ ค่าใช้จ่ายคาดหมายที่น้อยที่สุดจากโนดที่ n ไปยังโนดเป้าหมาย

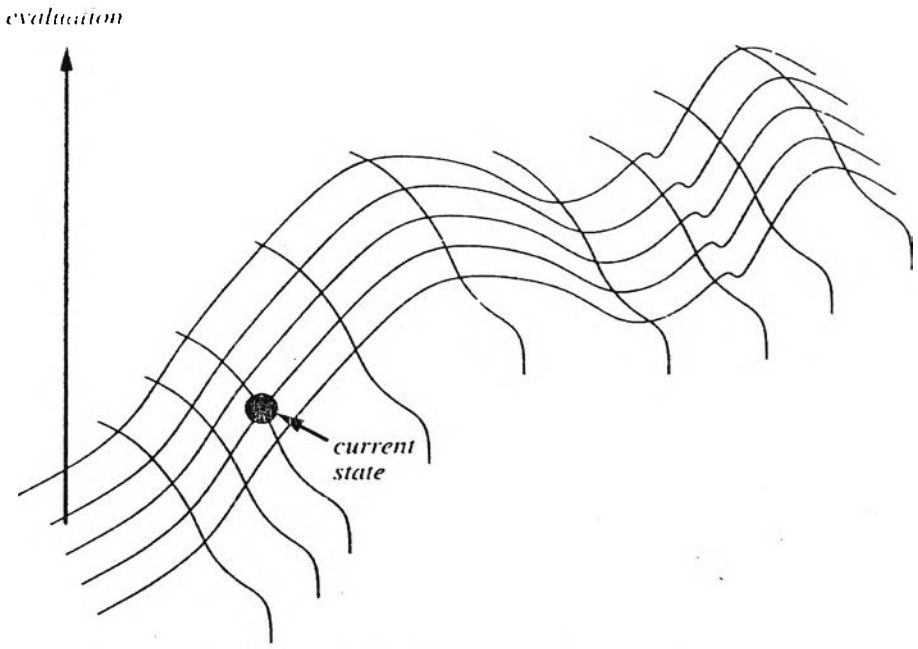
f(n) คือ ค่าใช้จ่ายคาดหมายของคำตอบที่น้อยที่สุดผ่านโนดที่ n

ข้อควรระวังในการเลือก Heuristic Function คือต้องไม่เป็นตัวประเมินค่าที่มากเกินจริง (Over Estimate Cost) ในการขยายไปยังสภาวะเป้าหมาย ดังนั้น h จะถูกเรียกใหม่ว่า Admission Heuristic ข้อสังเกตที่พบจาก A Search คือ ฟังก์ชันการประเมินค่าจะไม่ลดลงซึ่งฮิวริสติกที่มีพฤติกรรมเช่นนี้เรียกว่า Monotonicity [18] นอกจากนี้ Improving Search จะเริ่มต้นที่คำตอบเบื้องต้นที่มีจำนวนตัวแปรอยู่ครบทุกตัว (Full solution) และจะทำการปรับปรุงคำตอบโดยการปรับค่าตัวแปรตามแนวทางการปรับค่าของตัวแปร (Move Set หรือ Neighborhood) ซึ่งทำให้คำตอบของปัญหาจะขึ้นอยู่กับการเลือกคำตอบเบื้องต้น โดยข้อเสียของวิธีนี้คือ หากปัญหามีขนาดของ Neighborhood ใหญ่เกินไปจะทำให้การปรับปรุงค่าในแต่ละครั้งไม่มีประสิทธิภาพ แต่หาก Neighborhood มีขนาดเล็กเกินไปก็จะทำให้มีโอกาสที่ไม่ครอบคลุมคำตอบที่ดีได้ทั้งหมด นอกจากนี้สำหรับการปรับปรุงค่าในแต่ละครั้งควรจะมีความเป็นไปได้ของคำตอบ (Feasibility) และหากจำกัดแนวทางการปรับปรุงค่าเพื่อให้คำตอบเป็นไปได้ทุกครั้งย่อมจะทำให้การนำไปใช้นั้นยุ่งยากมากขึ้น ดังนั้นแนวทางการปรับปรุงค่าจึงยอมให้เกิดการปรับปรุงที่เป็นไปไม่ได้ (Infeasible) แต่จะทำการหักค่าของคำตอบในฟังก์ชันเป้าหมายลง [17] Local Search หรือ Hill Climbing เป็นวิธีการที่ประยุกต์มากจาก Improving Search โดยเริ่มต้นจาก Initial Feasible Solution จากนั้นทำการปรับปรุงค่าโดยพิจารณาตาม Neighborhood หากคำตอบที่ได้มีค่าที่ดีขึ้นก็จะทำ Iteration ซ้ำๆ ต่อไป แต่หากคำตอบที่ได้ไม่ได้ค่าที่ดีขึ้นแล้วก็จะหยุดการปรับปรุงค่าโดยได้คำตอบที่ดีที่สุดสัมพัทธ์ (Local Optimum) โดยกระบวนการค้นหาคำตอบแสดงในรูปที่ 7 ซึ่งจะมีการเคลื่อนที่ไปในทิศทางที่เพิ่มค่าของคำตอบขึ้น เนื่องจากวิธีการนี้ไม่ได้ใช้รูปแบบของแผนผังต้นไม้ ดังนั้นโครงสร้างของโนดจึงใช้หน่วยความจำในเรื่องของสภาวะและค่าประเมิน สิ่งสำคัญอีกอย่าง

หนึ่งคือ เมื่อมีจุดที่ดีที่สุดหลายๆ จุดที่สามารถเลือกได้ วิธีการนี้สามารถเลือกจุดนั้นขึ้นมาอย่างสุ่มได้ โดยอุปสรรคของวิธีการค้นหาคำตอบ ได้แก่

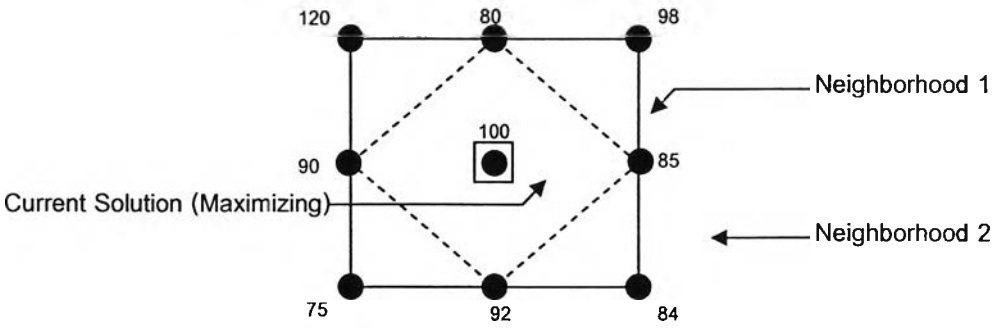
- Local Optimum เป็นตัวขัดขวางการหาค่าที่ดีที่สุดสากล (Global Optimum) โดยเมื่อพบกับ Local Optimum วิธีการ Hill Climbing ก็จะหยุดค้นหาคำตอบแม้ว่าคำตอบที่ได้จะไม่ใช่ค่าที่ดีที่สุดก็ตาม
- Plateau เป็นสภาวะที่ฟังก์ชันในการประเมินค่ามีค่าเท่ากัน ซึ่งต้องใช้การขยายพื้นที่แบบสุ่ม (Random Walk)
- Ridge เป็นแนวที่มีความชันมากที่สุด ดังนั้นวิธีการค้นหาคำตอบจะไปยังจุดสูงสุดของแนวที่มีความชันมากที่สุด ในบางกรณีการเคลื่อนที่จากจุดปัจจุบันไปยังจุดสูงสุดของ Ridge วิธีการค้นหาคำตอบจะแกว่งจากด้านหนึ่งไปอีกด้านหนึ่ง (Oscillation) ทำให้มีการปรับปรุงคำตอบน้อย

ในแต่ละกรณี วิธีการหาคำตอบอาจจะเคลื่อนที่ไปยังจุดที่ไม่ทำให้เกิดการปรับปรุงค่า สิ่งที่ต้องทำ คือเริ่มต้นซ้ำอีกครั้งจากจุดเริ่มต้นที่แตกต่างกัน ซึ่งแนวทางหนึ่งในการแก้ปัญหาคำตอบวนอยู่ใน Local Optimum ซึ่งจะเกิดเมื่อการปรับปรุงค่าตัวแปรที่ตำแหน่งนั้นๆ พิจารณา Neighborhood ชุดเดียวกันก็คือวิธีการ Multistart หรือ Random Start Hill Climbing เป็นการใช้วิธีการของ Hill Climbing จากการสุ่มสภาวะเริ่มต้น และทำไปเรื่อยๆ จนกว่าจะหยุดหรือไม่มีการปรับปรุงค่าซึ่งใช้หลักการเลือกจุดสำหรับใช้เป็น Initial Feasible Solution หลายๆ จุดซึ่งแต่ละจุดจะทำให้ได้ค่า Local Optimum ที่แตกต่างกันโดยคำตอบของ Heuristic ใช้ค่า Local Optimum ที่ดีที่สุด เพื่อเป็นการประหยัดเวลาในการค้นหาคำตอบสามารถกำหนดจำนวนของ Iteration ได้ และถ้าจำนวน Iteration เพียงพอ Random Start Hill Climbing จะสามารถหาคำตอบที่เหมาะสมที่สุดได้ ความสำเร็จของ Hill Climbing ขึ้นอยู่กับรูปร่างของพื้นผิว ถ้ามี Local Optima น้อย Random Start Hill Climbing จะสามารถหาคำตอบที่ดีได้อย่างรวดเร็ว แต่ปัญหาจริงๆ มักจะมีพื้นผิวที่มีลักษณะขรุขระ ถ้าปัญหาเป็น NP-Complete จะใช้เวลาในการหาคำตอบนานขึ้น ซึ่งอาจจะมีจำนวนครั้งที่ติดอยู่กับ Local Optima มากขึ้น แต่โดยทั่วไปแล้วสามารถหาคำตอบที่ดีได้โดยใช้จำนวน Iteration ที่ไม่มากนัก [17,18]



รูปที่ 7 วิธีการค้นหาคำตอบของ Local Search

นอกจากนี้การค้นหาคำตอบที่เป็น Meta-Heuristic เช่น TABU Search ได้ขยายแนวคิดในการหาคำตอบจากวิธีการแบบ Local Search โดยยอมให้สามารถปรับปรุงค่าที่ไม่ทำให้คำตอบดีขึ้นได้ (Nonimproving Move) ตัวอย่างเช่น สำหรับ Neighborhood ที่ 1 ในรูปที่ 8 การปรับปรุงค่าสามารถทำให้คำตอบออกจากค่า Local Optimum ได้โดยย้ายไปที่จุดที่ดีที่สุดถัดไป คือ 92 แม้ว่าค่าคำตอบที่ได้จะไม่ดีขึ้นก็ตาม แต่ด้วยวิธีการเช่นนี้อาจนำไปสู่การปรับปรุงค่าโดยการย้ายจุดอย่างไม่สิ้นสุด (Infinite Cycling) และการปรับปรุงค่าครั้งต่อไปจะทำให้คำตอบย้ายกลับไปสู่จุดเดิม คือ 100 ซึ่ง TABU Search ได้ป้องกันปัญหาที่จะเกิดขึ้นโดยการห้ามการปรับปรุงค่าบางแนวทาง (TABU Move) รวมถึงการปรับปรุงค่าที่จะทำให้คำตอบย้ายเข้าสู่ตำแหน่งก่อนหน้า ดังนั้นทุกๆ Iteration จะเกิดการปรับปรุงค่าคำตอบอย่าง TABU แม้ว่าจะคำตอบจะไม่ดีขึ้น และจะเก็บค่าคำตอบจากทุกๆ Iteration ไว้และเลือกค่าที่ดีที่สุดจากจำนวนการปรับปรุงค่าที่ตั้งไว้เป็นค่า Heuristic Optimum ซึ่งจะเห็นว่าส่วนสำคัญอันหนึ่งของวิธีการนี้ก็คือ แนวทางในการห้ามการปรับปรุงค่า (TABU Move) หากการปรับปรุงค่าจำนวนมากถูกห้ามไว้ก็จะทำให้คำตอบที่ได้มีค่าไม่ดี แต่หากจำนวนการห้ามการปรับปรุงมีน้อยเกินไปก็จะทำให้คำตอบยังคงวนอยู่ในค่า Local Optimum [17,18]



รูปที่ 8 Alternative Neighborhoods

Simulated Annealing (SA) ใช้การสุ่มเพื่อหลีกเลี่ยงการวนรอบของการปรับปรุงค่าคำตอบที่ไม่ทำให้ค่าคำตอบดีขึ้น เริ่มต้นจาก Feasible Solution และกำหนดชุดของการปรับปรุงค่า ในแต่ละครั้งของการปรับปรุงค่าอย่างสุ่ม หากการปรับปรุงได้ผลแล้วคำตอบที่ได้จะมีค่าดีขึ้นทันที แต่หากการปรับปรุงไม่ทำให้ค่าคำตอบดีขึ้นแล้วก็ยังคงยอมรับคำตอบนั้นด้วยความน่าจะเป็นเท่ากับ $e^{-degradation/temperature}$ ไม่เช่นนั้นการปรับปรุงครั้งต่อไปจะถูกเลือกและทำซ้ำขั้นตอนเดิม ค่าที่ดีที่สุดที่เก็บเอาไว้จะถือเป็นค่า Heuristic Optimum ซึ่งค่าการลดค่าคำตอบ (Degradation) ในความน่าจะเป็นแบบเอ็กโปเนนเชียลจะแสดงถึงปริมาณที่ลดลงของค่าฟังก์ชันเป้าหมายเทียบกับคำตอบจากการปรับปรุงค่า ดังนั้นการปรับปรุงค่าครั้งใดๆ ที่ให้ค่าคำตอบน้อยย่อมมีโอกาสได้รับคัดเลือกน้อยกว่าการปรับปรุงค่าครั้งที่มีค่ามากกว่าเสมอ นอกจากนี้ค่าอุณหภูมิ จะเป็นตัวกำหนดความเป็นแบบสุ่มของการปรับปรุงค่าซึ่งที่อุณหภูมิสูงจะทำให้ค่า e มีค่าเข้าใกล้ศูนย์และทำให้ทุกๆ การปรับปรุงค่ามีโอกาสถูกยอมรับ การนำ Simulated Annealing ไปใช้ในการหาคำตอบจะขึ้นอยู่กับอุณหภูมิที่สูงในขั้นต้นและลดลงจนเป็นศูนย์ตลอดช่วงการทำงานของ Search และแนวคิดของวิธีการนี้ก็คือ ให้การหาคำตอบสามารถตรวจสอบ (Explore) ได้อย่างทั่วถึงในขั้นแรกและจำกัดพื้นที่สำหรับการค้นหาลงมาให้เหลือเฉพาะพื้นที่ที่มีศักยภาพในภายหลัง และโดยปกติแล้วจำนวนครั้งของการปรับปรุงค่าโดยวิธีนี้จะสูงกว่าวิธี TABU Search หรือ Hill Climbing และมีการปฏิเสธการปรับปรุงค่าน้อยกว่า รวมถึงใช้ Neighbor จำนวนน้อยกว่าในการพิจารณา เมื่อเทียบกับวิธีแบบ Local Search ที่มักทดสอบแทบจะทั้งหมดของ Neighbor โดยที่จากการศึกษาพบว่าจำนวนครั้งของการปรับปรุงค่าอย่างสุ่มที่มากจะทำให้ Simulated Annealing หาคำตอบ Heuristic Optimum ที่ดีได้ [17,18]

Genetic Algorithms (GAs) เป็นวิธีแก้ปัญหาที่พัฒนามาจากวิธีการแบบ Evolutionary Algorithm ซึ่งใช้แก้ปัญหาประเภท Stochastic Optimization โดยการประยุกต์หลักการคัดเลือกโดยธรรมชาติและหลักการ พันธุศาสตร์ในการหาคำตอบ เริ่มต้นโดยการสุ่มชุดของ

Initial Solution ขึ้นมาเป็นประชากร (Population) และสำหรับแต่ละ Initial Solution ซึ่งเรียกว่า โครโมโซม (Chromosome) ประกอบไปด้วยผลคำตอบของปัญหาในขณะนั้นๆ โดยการปรับปรุงชุดคำตอบ (Population Solution) ตลอดการ search โดยประชากรบางส่วนจะถูกเลือกโดยคำนึงถึงค่าของคำตอบ (Fitness Value) ซึ่งส่งผลทำให้โอกาสในการถูกเลือกสำหรับประชากรแต่ละตัวไม่เท่ากัน จากนั้นคู่ของประชากรที่ได้รับการคัดเลือกจะแลกเปลี่ยนชุดตัวแปรของคำตอบย่อยในตำแหน่งเดียวกันเพื่อทำการปรับปรุงค่าคำตอบ เรียกขั้นตอนการปรับปรุงนี้ว่า Crossover จากนั้นจะทำการเลือกคู่ของประชากรสำหรับรุ่นถัดไป (Offspring) โดยทำซ้ำตามขั้นเดิม และในบางครั้งก็จะทำการปรับปรุงค่าของคำตอบด้วยวิธีการ Mutation ซึ่งเป็นการเปลี่ยนค่าของตัวแปรอย่างสุ่มภายในประชากรตัวใดๆ โดยที่ความน่าจะเป็นในการเลือกคู่ประชากรใดๆ คำนวณจาก $\frac{\text{Individual Solution Value}}{\text{Population Total}}$ ในทางทฤษฎีจะเห็นว่า Genetic Algorithm เป็นวิธีการหาคำตอบที่ไม่ขึ้นกับ Neighborhood แต่ทั้งนี้การปรับปรุงคำตอบด้วยวิธีการ Crossover ไม่สามารถประกันได้ว่าคำตอบที่ได้มาเป็นประชากรในรุ่นถัดไปจะเป็นไปได้ (Feasible) และวิธีการจัดตำแหน่งตัวแปรคำตอบสำหรับประชากรก็มีผลอย่างมากต่อกระบวนการ Crossover ขึ้นอยู่กับว่าจุดตัดจะอยู่ในตำแหน่งใด นอกจากนี้วิธีการกำหนดความน่าจะเป็นสำหรับการถูกคัดเลือกเป็นประชากรก็เหมาะสมเฉพาะกับปัญหาแบบหาค่ามากที่สุดเท่านั้น [17,19]

2.3 ผลงานวิจัยที่เกี่ยวข้อง

ปัญหาการจัดเส้นทางเดินรถพื้นฐาน (Basic Vehicle Routing Problem: VRP) เป็นปัญหาที่ได้รับการศึกษาและวิจัยอย่างแพร่หลายในช่วงสามทศวรรษที่ผ่านมา ก่อให้เกิดความหลากหลายในแง่มุมของปัญหาและมีลักษณะใกล้เคียงกับปัญหาในชีวิตจริงมากยิ่งขึ้น ทั้งนี้จากการศึกษาสามารถแบ่งงานวิจัยที่เกี่ยวข้องกับปัญหาการจัดเส้นทางเดินรถออกได้ดังนี้

ปัญหาการจัดเส้นทางเดินรถสามารถแบ่งออกเป็น 2 ลักษณะตามรูปแบบความต้องการที่แตกต่างกัน ได้แก่ ปัญหาการจัดเส้นทางเดินรถเพื่อการขนส่งสินค้าที่ทราบความต้องการอย่างแน่ชัดหรือสามารถทำนายความต้องการได้แน่นอน (Deterministic Demand) และปัญหาการจัดเส้นทางเดินรถเพื่อการขนส่งสินค้าให้สามารถตอบสนองต่อความต้องการแบบไม่แน่นอน (Stochastic Demand) ซึ่งปัญหาประเภทนี้เรียกอีกอย่างหนึ่งว่า Stochastic Vehicle Routing Problem (SVRP) Dror และคณะ [20] ได้แบ่งวิธีการแก้ปัญหา SVRP ออกเป็น 2 แบบ คือ Wait and See Situation ซึ่งมีแนวทางการจัดเส้นทางเดินรถหลังจากรับทราบความต้องการแล้ว หรือ Baptista และคณะ [13] เสนอให้กำหนดค่าความน่าจะเป็นที่จะเกิดความต้องการจากค่าข้อมูลในอดีตและแปลงความต้องการให้อยู่ในรูปของค่าความต้องการคาดหมาย (Expected Demand)

วิธีเช่นนี้มีแนวโน้มที่จะทำให้เกิดการแก้ปัญหาแบบ Deterministic Vehicle Routing Problem (DVRP) ซึ่ง Bertsimas และคณะ [21] เสนอว่าปัญหา SVRP ที่ทำการแก้ปัญหาด้วยการแทนในรูปแบบของ Deterministic จะทำให้การแก้ปัญหาง่ายขึ้นมาก สำหรับวิธีการแก้ปัญหาอีกแบบหนึ่ง คือ Here and Now Situation ถูกจัดเป็นปัญหาประเภท Priori Optimization หรือ Probabilistic Optimization เส้นทางเดินรถจะเป็นเซตของความต้องการคาดหมาย (Anticipated Demands) โดย Chan และคณะ [5] เสนอให้คาดคะเนความต้องการน่าจะเป็น (Probable Demand) โดยใช้วิธีการ Queuing Network

สำหรับ DVRP เมื่อความต้องการถูกกำหนดแล้วจะแก้ปัญหาได้ง่ายขึ้นด้วยทฤษฎีและวิธีการต่างๆ เช่น Clark-Wright Heuristic (Saving Algorithm) ซึ่งเป็นวิธีที่ได้รับการยอมรับและถูกนำไปใช้และพัฒนาอย่างหลากหลาย Mandl [23] ได้ปรับปรุงวิธีการ Clark-Wright Heuristic โดยเพิ่มจำนวนจุดรับของเส้นทางเดินรถเพื่อประหยัดเส้นทางได้มากที่สุดและสามารถลดจำนวนรถขนส่งสินค้าที่จะต้องไปบริการได้ นอกจากนี้ Naddef [23] เสนอทางแก้ปัญหา DVRP โดยใช้ Mixed-Integer Linear-Programming Formulation ซึ่งประยุกต์วิธีการ Maximizing Triangular Saving ของ Clark-Wright และ Dror และ Trudeau [24] เสนอแนวทางในการลดระยะทางที่เรียกว่า Split-Split Delivery Heuristic ที่ทำการแก้ปัญหาการจัดเส้นทางเดินรถโดยแยกเส้นทางการขนส่งเพื่อลดระยะทางรวม

สำหรับ SVRP วิธีการแก้ปัญหาที่เหมาะสมยังคงได้รับการพัฒนาอย่างต่อเนื่อง โดยเทคนิคที่สามารถประยุกต์ใช้จริงเทคนิคหนึ่งคือ Space-Filling Curve (SFC) ซึ่งเป็นการปรับรูปแบบจากปัญหา Location-Routing Network แบบ 2 มิติให้เป็นมิติเดียวโดยที่จุดรับจะถูกวางตำแหน่งบนแผ่นโค้ง SFC นี้ [5] นอกจากนี้ Barthodoli [25] และ Bowerman และคณะ [26] ยังได้เสนอให้แบ่งแผ่นโค้งออกเป็นส่วนๆ ซึ่งช่วงของพื้นที่ย่อยๆ (Subinterval) จะแสดงค่าพื้นที่ของการกระจายการให้บริการที่ทำให้รถหนึ่งคันสามารถบริการได้ไม่เกินความจุของรถ หรือไม่เกินระยะทางเดินรถที่มากที่สุด และจุดรับที่อยู่ใกล้กับจุดกลางของพื้นที่ย่อยๆ มากที่สุดจะทำหน้าที่เสมือนเป็นคลังสินค้า จากนั้นเส้นทางการเดินรถจึงถูกจัดขึ้นจากคลังสินค้าไปยังจุดรับภายในพื้นที่ย่อยนั้นๆ

นอกจากแง่มุมของความแตกต่างในด้านความต้องการแล้ว ปัญหาการจัดเส้นทางเดินรถพื้นฐานยังได้ขยายขอบเขตออกไปอย่างหลากหลาย ซึ่งชนิดของปัญหาจะแตกต่างกันไปตามเงื่อนไขและข้อจำกัดหรือลักษณะเฉพาะของปัญหาซึ่งจำเป็นต้องสร้างวิธีเฉพาะที่เหมาะสมขึ้นมาเพื่อให้สามารถแก้ไขปัญหาได้อย่างสมบูรณ์ทำให้มีลักษณะของปัญหาแบบ Multilevel Optimization Problem โดยจากการศึกษาพบว่าปัญหาการจัดเส้นทางเดินรถที่มีความต้องการในการรับบริการจากรถขนส่งเป็นรูปแบบซ้ำๆ ของแต่ละรอบเวลาเรียกว่า Period Traveling

Salesman Problem (PTSP) Paletta [14] เสนอทางแก้ไขปัญหาโดยทำการปรับเปลี่ยนวิธีการหาคำตอบสำหรับปัญหาการจัดเส้นทางเดินรถโดยทำการปรับปรุงคำตอบย่อย (Partial Solution) เพื่อให้ได้คำตอบที่ดีขึ้นทุกครั้งที่จำนวนจุดรับที่จัดเส้นทางไปแล้ว (Processed Cities) เท่ากับค่าพารามิเตอร์ที่กำหนดขึ้นสำหรับช่วงนั้นๆ และ Bapitsta และคณะ [13] ได้ขยายแนวคิดจากฮิวริสติกของ Christofides และ Beasley สำหรับปัญหาแบบ PVRP โดยประยุกต์กับหลักการของ TSP เมื่อกำหนดวันนัดหมาย (Day Combination) สำหรับการไปยังจุดรับโดยปรับปรุงคำตอบขั้นต้นด้วยวิธีการ Record-to-Record Approach

นอกจากนั้นปัญหาการจัดเส้นทางเดินรถที่มีความแตกต่างในด้านของรูปแบบของสินค้าที่ขนส่ง รูปแบบของเส้นทาง รูปแบบของรถขนส่งหรือวิธีการที่ใช้สำหรับการขนส่งยังเป็นแง่มมที่ได้รับความสนใจศึกษาซึ่งปัญหาประเภทนี้ได้แก่ Fleet Size and Mix Vehicle Routing Problem (FSMVRP) ที่ทำการจัดประเภทของรถขนส่งและสินค้าที่จะทำการจัดส่งเพื่อให้สามารถไปยังจุดรับได้โดยใช้ระยะทางหรือเสียค่าใช้จ่ายน้อยที่สุด Renaud และ Boctor [8] เสนอวิธีการแก้ปัญหาการจัดเส้นทางเดินรถประเภทที่ขนาดของสินค้าและชนิดของรถขนส่งมีความหลากหลายโดยทำการปรับปรุงวิธีการ Sweep-Based Algorithm เพื่อให้สามารถหาคำตอบได้ทั้งระนาบแบบ Euclidean และ Non-Euclidean หรือสำหรับปัญหาที่รถขนส่งประกอบด้วย Truck และ Trailer เรียกว่า Truck and Trailer Vehicle Routing Problem (TTVRP) ซึ่ง Chao [9] แสดงแนวทางการแก้ปัญหาการจัดเส้นทางเดินรถประเภท TTVRP ที่เส้นทางแบ่งได้เป็น 3 ประเภท คือ Pure Truck Route, Pure Vehicle Route without Subtour และ Complete Vehicle Route โดยปรับปรุงวิธีการ TABU Search Heuristic ร่วมกับแนวคิดแบบ Deterministic Annealing และพัฒนา TABU Restriction เรียกว่า Objective-Based TABU Restriction หรือ OTB เพื่อทำให้สามารถหาคำตอบที่มีค่าดีขึ้น และสำหรับเส้นทางที่มีข้อกำหนดเฉพาะดังเช่นปัญหา Traveling Salesman Problem with Backhaul (TSPB) Hassan และ Osman [11] ได้เสนอฮิวริสติกที่พัฒนาจาก Kohonen's Self-Organizing Feature Map for TSP with Backhaul โดยมีลักษณะเป็นโครงข่ายประสาท (Neural Network) และหาค่าใช้จ่ายแบบฮาลมิโทเนียน (Halmitonian Cost) ที่น้อยที่สุดบนกราฟโดยเริ่มจากเส้นทางขาไป (Linehaul) ไปยังเส้นทางขากลับ (Backhaul) และสำหรับปัญหา Multimodal Shortest Variable Path Problem (MSVPP) ซึ่งเส้นทางจากจุดตั้งต้นไปยังจุดปลายหนึ่งๆ มีทางเลือกหลายเส้นทางและแต่ละเส้นทางจะมีคุณสมบัติที่แตกต่างกัน และการเดินทางจากจุดตั้งต้นไปยังจุดปลายยังมีวิธีการเดินทางได้หลายแบบซึ่งจะใช้ทรัพยากรและระยะเวลาไม่เท่ากัน Lozano และ Storchi [10] ใช้วิธี Label Correcting Techniques และวิธีการ Ad Hoc Algorithm ในการแก้ปัญหา MSVPP

จากการศึกษาพบว่า Genetic algorithms (Gas) เป็น Meta-Heuristic วิธีหนึ่งที่ได้รับความสนใจและถูกนำมาใช้ในการแก้ปัญหาการจัดเส้นทางเดินรถเป็นจำนวนมากโดยงานวิจัยส่วนมากจะประยุกต์ใช้ GAs กับงาน VRP ในด้านที่แตกต่างกันโดยทำการปรับปรุงให้เหมาะสมกับเงื่อนไขของ VRP ประเภทนั้นๆ แต่สำหรับงานวิจัยที่ใช้ฮิวริสติกแบบเดียวกันสำหรับงาน VRP ทางด้านเดียวกันก็จะสร้างข้อแตกต่างโดยเน้นที่ระยะเวลาในการหาคำตอบหรือเน้นที่ค่าคำตอบที่ดีกว่าฮิวริสติกแบบเดิม

Hwang [27] ทำการออกแบบระบบ Supply-Chain Logistics ซึ่งคำนึงถึงผลกระทบระหว่างจำนวนคลังสินค้าและระยะทางจากคลังสินค้าไปยังจุดรับต่างๆ เพื่อให้สามารถตอบสนองต่อระดับความต้องการที่กำหนด โดยใช้ 0-1 Integer Programming ในการแก้ปัญหาการกำหนดจำนวนและตำแหน่งคลังสินค้าแบบ Deterministic และทำการพัฒนาวิธีการ Sector-Clustering Module Transform เพื่อวิเคราะห์การจัดสรรจุดรับให้แก่คลังสินค้าโดยปรับจาก Multi-Warehouse/Distribution Center ให้เป็น Single-Warehouse/Distribution Center นอกจากนี้ยังทำการปรับปรุง GA Operator ทั้งในส่วนของ Crossover Operator, Mutation Operator และประชากรขั้นแรกสำหรับ Genetic Algorithm ที่ใช้ในการหาเส้นทางการขนส่งที่เหมาะสมตามหลักการของ Traveling Salesman Problem และ Hwang [15] ยังได้เสนอการปรับปรุง GA Operator เพื่อให้สามารถแก้ปัญหาสำหรับการจัดเส้นทางเดินรถที่มีการจำกัดระยะเวลาในการให้บริการให้เป็นไปตามที่ตกลงกัน Choi [16] ทำการประยุกต์ใช้ GAs โดยปรับปรุง GA Operator ให้มีขอบเขตการค้นหาคำตอบที่กว้างมากขึ้นโดยทำการปรับให้ GA Operator หาค่าคำตอบจากคำตอบที่เป็นไปไม่ได้ (Infeasible Solution) ที่มีโอกาสเป็นคำตอบที่ดีได้และประยุกต์ใช้วิธีนี้กับงาน TSP ซึ่งมีค่าของระยะทางไม่สมมาตร (Asymmetric Traveling Salesman Problem) และ Moon และคณะ [6] ได้ใช้ GAs โดยประยุกต์ใช้กับ TSP with Precedence Constraints (TSPPC) ซึ่งเป็นลักษณะของการบังคับเงื่อนไขลำดับการเดินทางไปยังจุดรับแต่ละแห่ง โดยใช้แนวคิดของ Topological Sort Procedure เข้าไปกับการหาคำตอบโดยใช้ GAs นอกจากนี้ยังมีงานวิจัยที่ศึกษาเกี่ยวกับผลกระทบที่เกิดจากข้อจำกัดทางด้านระยะเวลาที่ส่งผลต่อคุณสมบัติของเส้นทางที่ถูกนำมาพิจารณาในการจัดเส้นทางเดินรถในแต่ละช่วงเวลาหรือ Time-Dependent Traveling Salesman Problem (TD-TSP) ซึ่ง Schneider [7] ได้ทำการเสนอให้พิจารณาการจัดเส้นทางที่เลี่ยงเส้นทางที่มีการจราจรติดขัด โดยอาจทำให้ระยะทางที่ใช้มากขึ้นได้ ซึ่งเป็นผลให้การตัดสินใจเลือกเส้นทางเปลี่ยนไปตามช่วงเวลาและความหนาแน่นของเส้นทางการจราจร โดยพิจารณาค่า Hamiltonian Cost และใช้วิธีการหาคำตอบแบบ Simulated Annealing

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.4 บทสรุป

ปัญหาการจัดเส้นทางเดินรถเป็นปัญหารูปแบบการหาค่าเหมาะสมที่สุดที่มีความซับซ้อนของการคำนวณในระดับเอ็นพีแบบยากและมีการศึกษาแยกย่อยออกไปอย่างหลากหลายตามแง่มุมเฉพาะของผู้ศึกษาซึ่งก่อให้เกิดรูปแบบที่แตกต่างกันของปัญหาการจัดเส้นทางเดินรถ โดยปัญหาการจัดเส้นทางเดินรถที่ได้รับความสนใจมากที่สุด ได้แก่ Traveling Salesman Problem และ Vehicle Routing Problem นอกจากนี้กรรมวิธีการแก้ไขปัญหาแบบการค้นหาคำตอบสำหรับปัญหาการจัดเส้นทางการเดินรถที่สำคัญสามารถแบ่งออกได้เป็น 2 ประเภท คือ วิธีการแก้ปัญหาแบบ Local Search และวิธีการแก้ปัญหาแบบ Meta-Heuristic เช่น Genetic Algorithm และ TABU Search สำหรับงานวิจัยที่เกี่ยวข้องสามารถแบ่งออกเป็น 2 แนวทาง คือ แนวทางการประยุกต์ใช้กรรมวิธีการแก้ไขปัญหาการจัดเส้นทางเดินรถเข้ากับระบบงานที่มีรูปแบบเฉพาะ หรือมีความซับซ้อนของระบบงานมาก และแนวทางการปรับปรุงประสิทธิภาพของกรรมวิธีการหาคำตอบ