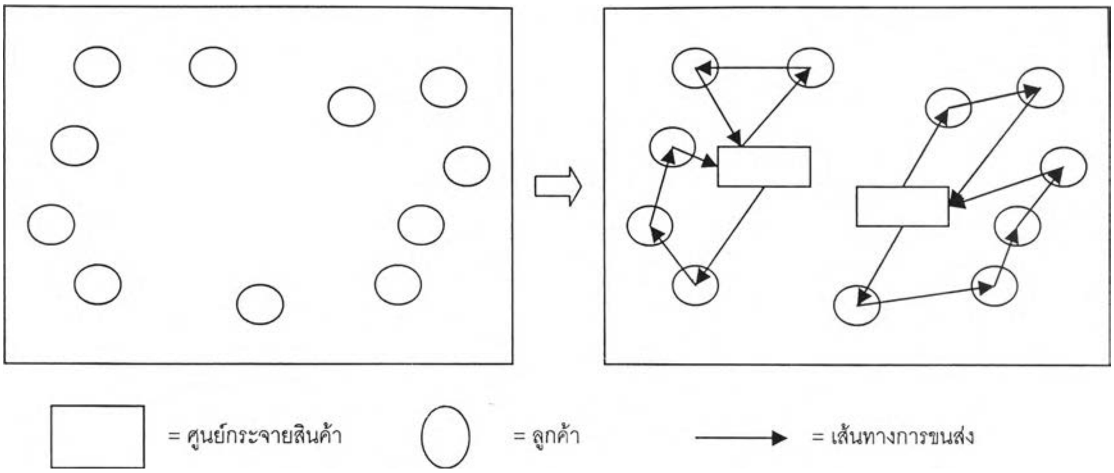
# บทที่ 2

# เอกสารและงานวิจัยที่เกี่ยวข้อง

ปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง จัดเป็นปัญหารูปแบบการหาค่าเหมาะสมที่สุดเชิงการจัด (Combinatorial Optimization) หรือ การหาค่าเหมาะสมที่สุดหลายระดับ (Multilevel Optimization) ที่มีความซับซ้อนของการคำนวณในระดับเอ็นพีแบบยาก (Nonpolynomial Hard: NP-Hard) ซึ่งใช้เวลาในการหาคำตอบยาวนาน และเมื่อขนาดของปัญหาเพิ่มขึ้นก็จะใช้เวลาในการหาคำตอบเพิ่มขึ้นอย่างเอ็กโปเนนเชียล ทำให้ต้องใช้เวลาในการหาคำตอบมากด้วยวิธีคำนวณตัวแบบทางคณิตศาสตร์ ดังนั้นจึงจำเป็นต้องอาศัยวิธีการหาคำตอบแบบฮิวริสติก (Heuristic Approach) สำหรับการแก้ไขปัญหา ในบทนี้จึงได้แบ่งเนื้อหาออกเป็น 3 เรื่อง ได้แก่ แนวคิดและทฤษฎีที่เกี่ยวข้องกับปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง แนวคิดและทฤษฎีที่เกี่ยวข้องกับการนำฮิวริสติกมาใช้ในการแก้ปัญหาและผลงานวิจัยที่เกี่ยวข้อง

## 2.1 แนวคิดและทฤษฎีที่เกี่ยวข้องกับปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง

"ปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง (Location Routing Problem : LRP)" นั้นสามารถให้คำอธิบายได้ดังนี้ คือ เซตคำตอบที่เป็นไปได้ของสถานที่ที่สามารถเป็นศูนย์กระจายสินค้าได้ ซึ่งลูกค้าแต่ละคนจะรับการจัดสรรให้กับเซตคำตอบเหล่านี้ ศูนย์กระจายสินค้าจะทำหน้าที่กระจายสินค้า โดยยานพาหนะที่ได้รับมอบหมายจากแต่ละศูนย์กระจายสินค้า และขนส่งไปตามเส้นทางที่กำหนดไว้ โดยมีค่าใช้จ่าย 3 ส่วนคือ ค่าใช้จ่ายในการตั้งศูนย์กระจายสินค้า ค่าใช้จ่ายเกี่ยวกับการจัดหายานพาหนะ และค่าใช้จ่ายในการดำเนินการขนส่ง โดยทั้งนี้เราพยายามทำให้ค่าใช้จ่ายรวมมีค่าต่ำที่สุด โดยตัวอย่างลักษณะของปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง แสดงในรูปที่ 2.1




รูปที่ 2.1: ตัวอย่างลักษณะของปัญหาการที่ตั้งของศูนย์กระจายสินค้าและเส้นทางขนส่ง

ซึ่งปัญหานี้ Tuzun, D., and Burke [1] ได้เสนอตัวแบบทางคณิตศาสตร์ได้ดังนี้

$$\text{Minimize} \sum_{i\in S}\sum_{j\in S}\sum_{k\in V} C_{ij}X_{ijk} + \sum_{k\in V}\Big(\varnothing \sum_{r\in G}\sum_{j\in H} X_{rjk}\Big) + \sum_{r\in G} F_r Z_r \qquad (1)$$

Subject to :

$$\sum_{k\in V}\sum_{i\in S} X_{ijk} = 1 \quad \forall j \in H, \qquad (2)$$

$$\sum_{j\in H}\sum_{i\in S} q_j X_{ijk} \le \text{Ç} \qquad (3)$$

$$\sum_{i\in S} X_{ipk} - \sum_{j\in S} X_{pjk} = 0 \quad \forall k \in V,\ p \in S, \qquad (4)$$

$$\sum_{r\in G}\sum_{i\in H} X_{rjk} \le 1 \quad \forall k \in V, \qquad (5)$$

$$\sum_{k\in V} X_{rmk} + Z_r + Z_m \le 2 \quad \forall m \in 1,\ldots,R,\ r \in G, \qquad (6)$$

$$\sum_{k\in V}\sum_{j\in H} X_{rjk} - Z_r \ge 0 \quad \forall r \in G, \qquad (7)$$

$$\sum_{j\in H\, i\in S} X_{rjk} - Z_r \le 0 \quad \forall k \in V,\ \forall r \in G, \qquad (8)$$

$$X_{ijk} = 0 \text{ or } 1 \quad \forall i,j \in S,\ k \in V, \qquad (9)$$

$$Z_r = 0 \text{ or } 1 \quad \forall r \in G. \qquad (10)$$

โดยที่

| | |
|---|---|
| G | {r I r = 1,..,R} เป็นเซตของ สถานที่ตั้งของศูนย์กระจายสินค้าที่เป็นไปได้ R แห่ง |
| H | {i I i = R+1,..,R+N} เป็นเซตของ ลูกค้า N คน |
| S | {G} U {H} เป็นเซตของ สถานที่ตั้งที่เป็นไปได้และลูกค้า โดยจะกำหนดให้เป็น NODE |
| V | {k I k = 1,..,K} เป็นเซตของ ยานพาหนะ K คัน |
| $C_{ij}$ | ค่าใช้จ่ายในการเดินทางจาก NODE ที่ i ไป NODE ที่ j |

Ø ค่าใช้จ่ายในการจัดหายานพาหนะ

$F_r$ ค่าใช้จ่ายในการสร้างศูนย์กระจายสินค้าที่สถานที่ r

$q_j$ ความต้องการของลูกค้าคนที่ j

Ç ความจุของยานพาหนะ

ตัวแปรที่ทำการตัดสินใจ

$$X_{ijk} = \begin{cases} 1 & \text{ถ้า ยานพาหนะที่ k ออกจาก NODE ที่ i ไป NODE ที่ j},\ i \in S,\ j \in S,\ k \in V,\ i \neq j, \\ 0 & \text{ในกรณีอื่นๆ}, \end{cases}$$

$$Z_r = \begin{cases} 1 & \text{ถ้า ศูนย์กระจายสินค้าถูกตั้งขึ้นที่สถานที่ r},\ r \in G, \\ 0 & \text{ในกรณีอื่นๆ}, \end{cases}$$

โดยแต่ละสมการมีคำอธิบายดังนี้

(1) สมการวัตถุประสงค์

(2) ลูกค้าทุกคนต้องได้รับบริการจากยานพาหนะเพียงคันใดคันหนึ่งเท่านั้น

(3) ข้อจำกัดด้านความจุของยานพาหนะ

(4) หากยานพาหนะใดเข้าไปที่จุดใดก็ตามต้องออกมาจาก จุดนั้นด้วย

(5) ยานพาหนะแต่ละคันต้องออกมาจากศูนย์กระจายสินค้าแห่งใดแห่งหนึ่งเท่านั้น

(6) ไม่มีการเชื่อมโยงกันระหว่างศูนย์กระจายสินค้า 2 แห่ง

(7) ยานพาหนะจะออกมาจากสถานที่ตั้งใดได้ก็ต่อเมื่อ สถานที่นั้นถูกเลือกให้เป็นศูนย์กระจายสินค้า

(8) ยานพาหนะจะออกมาจากสถานที่ตั้งใดได้ก็ต่อเมื่อ สถานที่นั้นถูกเลือกให้เป็นศูนย์กระจายสินค้า

(9) กำหนดตัวแปร $X_{ijk}$ ให้เป็นแบบ ไบนารี่

(10) กำหนดตัวแปร $Z_r$ ให้เป็นแบบ ไบนารี่

## 2.2 การแก้ปัญหาด้วยวิธีค้นหาคำตอบ (Searching)

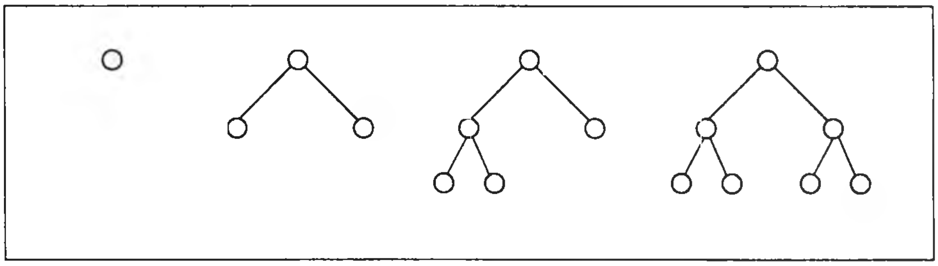
ในการหาคำตอบที่ดีเหมาะสมที่สุด สำหรับปัญหา Combinatorial Optimization ได้มีการพัฒนาเทคนิคสำหรับแก้ไขปัญหาเหล่านี้หลายวิธี ลักษณะของ Combinatorial Optimization คือ ไม่สามารถใช้วิธีการทางคณิตศาสตร์โดยตรงในการแก้ปัญหาได้ เมื่อปัญหามีขนาดใหญ่ขึ้น

ดังนั้นจึงมีการคิดค้นวิธีการหาคำตอบที่ดีที่สุดสำหรับแก้ปัญหา วิธีการหนึ่งที่นำมาใช้คือ วิธีค้นหาคำตอบ

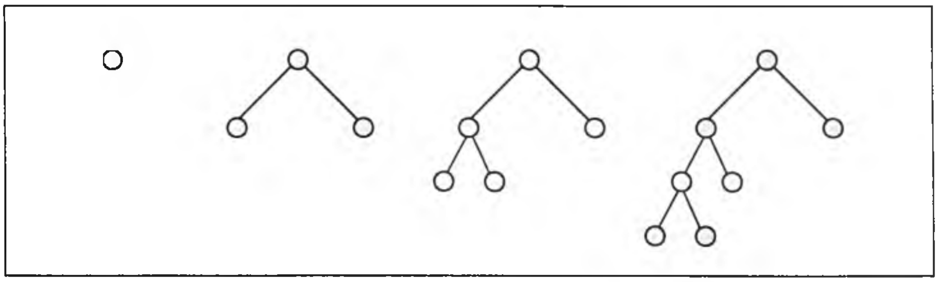
สามารถแบ่งประเภทของวิธีค้นหาคำตอบออกเป็น Uninformed Search หรือ Blind Search เป็นการหาคำตอบโดยไม่มีข้อมูลเกี่ยวกับจำนวนขั้นตอนหรือ Path Cost จากสภาวะปัจจุบัน (Current State) ไปสู่สภาวะเป้าหมาย (Goal State) และ Informed Search หรือ Heuristic Search ที่อาศัยข้อมูลในการเปลี่ยนจากสภาวะปัจจุบันไปสู่สภาวะเป้าหมาย [2] ตัวอย่างฮิวริสติกที่รู้จักกันดี ได้แก่ Greedy Search Admission Search Hill Climbing Search โดย Informed Search มีประสิทธิภาพมากกว่า Uninformed Search ตัวอย่างของ Uninformed Search ได้แก่ Bread-first Search (BFS) และ Depth-first Search (DFS)

ก. Bread-first Search (BFS) จะขยายการค้นหาคำตอบจากเป้าหมายที่อยู่ใกล้ที่สุดก่อน ดังแสดงในรูปที่ 2.2 ส่วนมากแล้ว BFS เป็นวิธีการค้นหาคำตอบที่สมบูรณ์และให้ค่าที่เหมาะสมที่สุด แต่ใช้หน่วยความจำสูง

ข. Depth-first Search (DFS) จะขยายการค้นหาคำตอบไปยังส่วนที่อยู่ลึกที่สุดของแผนผังต้นไม้ (Tree) ดังแสดงในรูปที่ 2.3 เป็นวิธีการค้นหาคำตอบที่ไม่สมบูรณ์และไม่ได้ให้ค่าที่เหมาะสมที่สุด แต่ใช้หน่วยความจำน้อยกว่าเมื่อเปรียบเทียบกับวิธีการ BFS



รูปที่ 2.2 ขั้นตอนในวิธีขยายการค้นหาคำตอบของ Bread-first search



รูปที่ 2.3 ขั้นตอนในวิธีขยายการค้นหาคำตอบของ Depth-first search

สำหรับวิธีการของ Informed Search หรือ Heuristic Search มีความหลากหลายและเหมาะสมกับลักษณะปัญหาต่างๆ กัน ได้แก่

ก. Relaxation Heuristic คือ วิธีการแก้ปัญหาอย่างง่ายของปัญหาแบบ Exact Optimization อาศัยการลดความยุ่งยากของการปัญหาบางส่วน เช่น การ Relax ค่าที่ดีที่สุด (Optimum) โดยการปัดตัวเลข (Rounding) เพื่อให้สามารถหาคำตอบได้ง่ายและรวดเร็วขึ้นอย่างเช่นวิธี Lagrangian Relaxation ทำการแก้ปัญหาโดยแบ่งปัญหาออกเป็นส่วนๆ และหาคำตอบสำหรับแต่ละส่วนโดยอาศัยชุดของคำตอบจากปัญหาย่อยเพื่อปรับปรุงค่าที่ดีที่สุด ซึ่งวิธี Relaxation Heuristic จะไม่เหมาะสมกับปัญหามีความซับซ้อนในการสร้างโมเดลหรือปัญหาที่เป็น Integer-Infeasible

ข. Decomposition Heuristic เป็นการแบ่งปัญหาออกเป็นส่วนๆ และแก้ปัญหาแต่ละส่วนแยกจากกันซึ่งมีวิธีการแก้ปัญหาหลายแบบ ได้แก่ Iteration ซึ่งจะหาคำตอบสำหรับปัญหาย่อยโดยคงค่าของตัวแปรอื่นๆ และปรับปรุงค่าของตัวแปรตัดสินใจสำหรับชุดคำตอบย่อยนั้นๆ Column generation เป็นวิธีที่เหมาะสมเมื่อปัญหามีแง่มุมที่แสดงในรูปแบบทางคณิตศาสตร์ได้ยาก

ค. Constructive Search แก้ปัญหาโดยอาศัย Null solution และทำการหาแก้ปัญหาแบบครั้งต่อครั้งโดยอาศัยข้อมูลจากการแก้ปัญหาครั้งก่อน เพื่อให้ได้คำตอบของปัญหา (Full solution) โดยการหาคำตอบของวิธี Constructive Search มีลักษณะที่คำนึงผลประโยชน์เฉพาะหน้า ซึ่งข้อสังเกตหนึ่งของวิธีการนี้คือคำตอบที่ได้จะขึ้นอยู่กับลำดับของการแก้ปัญหา ดังนั้นหากการแก้ปัญหาครั้งก่อนหน้าให้คำตอบที่ไม่ดีก็จะส่งผล (Degrade)ต่อคำตอบในครั้งถัดไป[3] ตัวอย่างของ Constructive Search ได้แก่ Greedy Search และ Admission Search

1. Greedy Search มีวิธีการขยายการค้นหาคำตอบจากคำตอบที่ดีที่สุดในปัจจุบันไปยังคำตอบใหม่ โดยประเมินค่า Cost ในการขยายจากสภาวะเฉพาะ( Particular State) ไปยังสภาวะเป้าหมาย ฟังก์ชันในการคำนวณตัวประเมินค่านี้เรียกว่า Heuristic Function ที่สามารถสร้างเป็นฟังก์ชันความสัมพันธ์ของอะไรก็ได้ โดยเราต้องการให้ $h(n) = 0$ เมื่อ $n$ คือสภาวะเป้าหมาย ถึงแม้ว่า Greedy Search จะให้ค่า Minimal Search Cost แต่เป็นวิธีการที่ไม่มีการขยายการค้นหาคำตอบไปในแนวทางอื่นเลยและไม่การันตีว่าจะได้ค่าที่เหมาะสมที่สุด แต่อัลกอริทึมของ Greedy Search สามารถ

ใช้งานได้ดี และหาคำตอบได้อย่างรวดเร็ว ข้อด้อยของวิธีการ Greedy Search ค่อนข้างแย่ (weak) ต่อการเริ่มต้นที่ไม่ดี ทำให้เกิดการขยายการค้นหาคำตอบไปยังส่วนที่ไม่จำเป็น และถ้าไม่มีการป้องกันการทำซ้ำก็จะไม่สามารถหาคำตอบได้

2. Admission Search (A Search) เป็นการนำเอาส่วนดีของทั้ง Greedy Search กับ Uniform-Cost Search มารวมกัน เนื่องจาก Greedy Search ใช้หลักการ Minimize Estimate Costไปยังสภาวะเป้าหมายโดยใช้ Heuristic Function, *h(n)* เป็นวิธีการค้นหาคำตอบที่ไม่สมบูรณ์และไม่ได้ให้ค่าที่เหมาะสมที่สุด ส่วน Uniform-Cost Search ใช้หลักการ Minimize the Cost of Path, *g(n)* เป็นวิธีการที่สมบูรณ์และให้คำตอบที่เหมาะสมที่สุดแต่ไม่มีประสิทธิภาพ A search จึงสร้างฟังก์ชันการประเมินค่า (Evaluation Function)ขึ้นมาใหม่ ดังสมการ

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

โดยที่ *g(n)* คือ path cost จากโนดเริ่มต้นไปยังโนดที่ n

*h(n)* คือ estimated cost ที่น้อยที่สุดจากโนดที่ n ไปยังโนดเป้าหมาย

*f(n)* คือ estimated cost ของคำตอบที่น้อยที่สุดผ่านโนดที่ n

ข้อควรระวังในการเลือก Heuristic Function คือต้องไม่เป็นตัวประเมินค่าที่มากเกิน (Over Estimate Cost) ในการขยายไปยังสภาวะเป้าหมาย ดังนั้น h จะถูกเรียกใหม่ว่า Admission Heuristic ข้อสังเกตที่พบจาก A Search คือ f-cost ไม่เคยลดค่าลงซึ่งฮิวริสติกที่มีพฤติกรรมเช่นนี้เรียกว่า Monotonicity

ง. Improving Search การหาคำตอบจะเริ่มต้นที่คำตอบเบื้องต้นที่มีจำนวนตัวแปรอยู่ครบทุกตัว (Full Solution) และจะทำการปรับปรุงคำตอบโดยการปรับค่าตัวแปรตามแนวทางการปรับค่าของตัวแปร (Move Set or Neighborhood) ซึ่งทำให้คำตอบของปัญหาจะขึ้นอยู่กับการเลือกคำตอบเบื้องต้น โดยข้อเสียของวิธีนี้คือ หากปัญหามีขนาดของ Neighborhood ใหญ่เกินไปจะทำให้การปรับปรุงค่าในแต่ละครั้งไม่มีประสิทธิภาพ แต่หาก Neighborhood มีขนาดเล็กเกินไปก็จะทำให้มีโอกาสที่ไม่ครอบคลุมคำตอบที่ดีได้ทั้งหมด นอกจากนี้สำหรับการปรับปรุงค่าในแต่ละครั้งควรจะมีความเป็นไปได้ของคำตอบ (Feasibility) และหากจำกัดแนวทางการปรับปรุงค่าเพื่อให้คำตอบเป็นไปได้ทุกครั้งย่อมจะทำให้การนำไปใช้นั้นยุ่งยากมากขึ้น ดังนั้นแนวทางการปรับปรุงค่าจึงยอม

ให้เกิดการปรับปรุงที่เป็นไปไม่ได้ (Infeasible) แต่จะทำการหักค่าของคำตอบในฟังก์ชันวัตถุประสงค์ลง

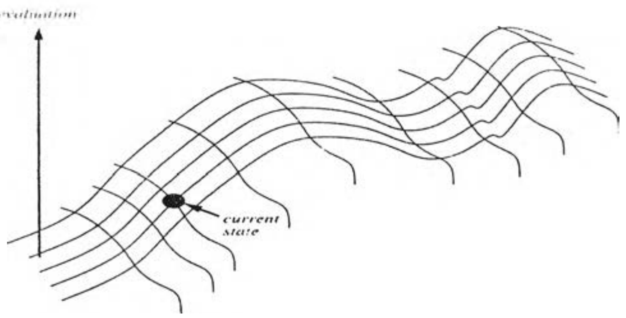
จ. Local Search หรือ Hill-climbing เป็นวิธีการที่ประยุกต์มาจาก Improving Search โดยเริ่มต้นจากคำตอบตั้งต้นที่มีความเป็นไปได้ ( Initial Feasible Solution)จากนั้นทำการปรับปรุงค่าโดยพิจารณาตามคำตอบที่อยู่ถัดกัน (Neighborhood)หากคำตอบที่ได้มีค่าที่ดีขึ้นก็จะทำการปรับปรุงค่าต่อไป แต่หากคำตอบที่ได้ไม่ดีขึ้นก็จะหยุดการปรับปรุงค่าโดยได้คำตอบแบบ Local Optimum โดยแนวทางหนึ่งในการแก้ปัญหาคำตอบวนอยู่ใน Local Optimum ซึ่งจะเกิดเมื่อการปรับปรุงค่าตัวแปรที่ตำแหน่งนั้นๆ พิจารณา Neighborhood ชุดเดียวกันก็คือวิธีการ Multi-start ซึ่งเลือกจุดสำหรับใช้เป็น Initial Feasible Solution หลายๆ จุดซึ่งแต่ละจุดจะทำให้ได้ค่า Local Optimum ที่แตกต่างกันโดยคำตอบของ Heuristic ใช้ค่า Local Optimum ที่ดีที่สุด

อัลกอริทึมในการค้นหาคำตอบของ Hill-climbing แสดงในรูปที่ 2.4 ซึ่งจะมีการเคลื่อนที่ไปในทิศทางที่เพิ่มค่าของคำตอบขึ้น เนื่องจากอัลกอริทึมนี้ไม่ได้ใช้รูปแบบของแผนผังต้นไม้ ดังนั้นโครงสร้างของโนดจึงใช้หน่วยความจำในเรื่องของสภาวะและค่าประเมินที่เรียกว่า Value สิ่งสำคัญอีกอย่างหนึ่งคือ เมื่อมีจุดที่ดีที่สุดหลายๆจุดที่สามารถเลือกได้ อัลกอริทึมนี้สามารถเลือกจุดนั้นขึ้นมาอย่างสุ่มได้ อุปสรรคของวิธีค้นหาคำตอบ ได้แก่

1. Local Maxima :Local Maximum เป็นตัวขัดขวางการหา Global Maximum แม้ว่าLocal Maxima เป็นจุดสูงสุดแต่เมื่อเทียบกับทุกสภาวะแล้วจะมีจุดอื่นที่สูงกว่า เมื่อพบกับ Local Maxima ก็จะหยุดค้นหาคำตอบแม้ว่าคำตอบที่ได้จะไม่ใช่ค่าที่ดีที่สุดก็ตาม
2. Plateaux : Plateau เป็นสภาวะที่ฟังก์ชันในการประเมินค่ามีค่าเท่ากัน ซึ่งต้องใช้การขยายพื้นที่แบบสุ่ม (Random Walk)
3. Ridges :Ridge เป็นแนวที่มีความชันมากที่สุด ดังนั้นวิธีค้นหาคำตอบจะไปยังจุดสูงสุดของแนวที่มีความชันมากที่สุด ในบางกรณีการเคลื่อนที่จากจุดปัจจุบันไปยังจุดสูงสุดของ ridge วิธีการค้นหาคำตอบจะแกว่งจากด้านหนึ่งไปอีกด้านหนึ่ง (Oscillation) ทำให้มีการปรับปรุงคำตอบน้อย

ในแต่ละกรณี อัลกอริทึมอาจจะเคลื่อนที่ไปยังจุดที่ไม่ทำให้เกิดการปรับปรุงค่า สิ่งที่ต้องทำคือเริ่มต้นซ้ำอีกครั้งจากจุดเริ่มต้นที่แตกต่างกัน การสุ่มสภาวะเริ่มต้น (Random Start Hill Climbing) จะทำให้ได้จุดเริ่มต้นที่แตกต่างกัน และทำไปเรื่อยๆ

จนกว่าจะหยุดหรือไม่มีการปรับปรุงค่า เพื่อเป็นการประหยัดเวลาในการค้นหาคำตอบ สามารถกำหนดจำนวนของตารางในการหาคำตอบ( Iteration) ได้ และถ้าจำนวนตารางในการหาคำตอบเพียงพอ การสุ่มสภาวะเริ่มต้นจะสามารถหาคำตอบที่เหมาะสมที่สุดได้ ความสำเร็จของ Hill -climbing ขึ้นอยู่กับรูปร่างของพื้นผิว ถ้ามี Local Maxima น้อย การสุ่มสภาวะเริ่มต้นจะสามารถหาคำตอบที่ดีได้อย่างรวดเร็ว แต่ปัญหาจริงๆมักจะมีพื้นผิวที่มีลักษณะขรุขระ ถ้าปัญหาเป็น NP-complete จะใช้เวลาในการหาคำตอบนานขึ้น ซึ่งอาจจะมีจำนวนครั้งที่ติดอยู่กับ Local Maxima มากขึ้น แต่โดยทั่วไปแล้วสามารถหาคำตอบที่ดีได้โดยใช้จำนวนตารางในการหาคำตอบไม่มากนัก

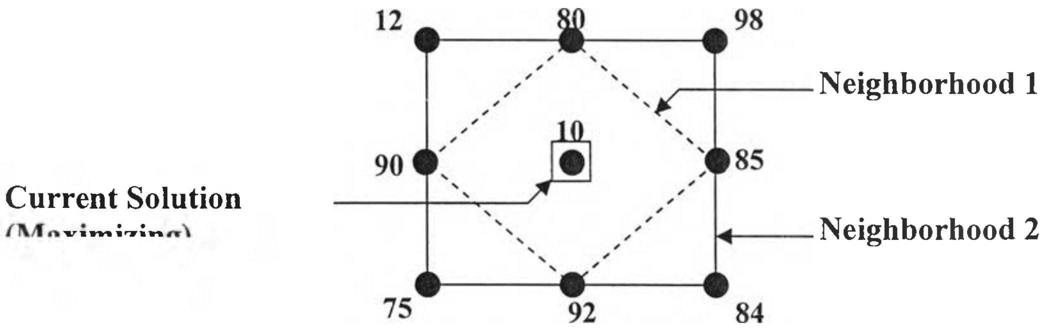



รูปที่ 2.4 อัลกอริทึมในการค้นหาคำตอบของ Hill-climbing

ในการใช้ Heuristic Search เพื่อหาคำตอบนั้นจะมีอุปสรรคของวิธีค้นหาคำตอบเนื่องจากการเกิดการวนซ้ำ (Cycling) จึงมีการพัฒนาจาก Heuristic Search ให้เป็น Meta-Heuristic เพื่อปรับปรุงประสิทธิภาพของวิธีค้นหาคำตอบให้ดีขึ้น Meta-Heuristic ที่ถูกนำมาใช้ในการหาคำตอบอย่างแพร่หลาย ได้แก่ TABU Search ,Simulated Annealing และ Genetic Algorithms

ก. TABU Search ถูกคิดค้นโดย Glover ในปี ค.ศ.1989 ขยายแนวคิดในการหาคำตอบจากวิธีการแบบ Local Search โดยยอมให้สามารถย้ายค่าไปในตำแหน่งที่ไม่ปรับปรุงค่าคำตอบให้ดีขึ้นได้ (Nonimproving Move) ตัวอย่างเช่น สำหรับ Neighborhood ที่ 1 ในรูปที่ 2.5 ซึ่งทำให้คำตอบออกจากค่า Local optimum ได้โดยย้ายไปที่จุดที่ดีที่สุดถัดไป คือ 92 แม้ว่าค่าคำตอบที่ได้จะไม่ดีขึ้นก็ตาม แต่ด้วยวิธีการเช่นนี้อาจนำไปสู่การปรับปรุงค่าโดยการย้ายจุดอย่างไม่สิ้นสุด (Infinite Cycling) และการปรับปรุงค่าครั้งต่อไปจะทำให้คำตอบย้ายกลับไปสู่จุดเดิม คือ 100 TABU Search มีกลไกเพื่อป้องกันปัญหาที่จะเกิดขึ้นโดยการห้ามปรับปรุงค่าบางแนวทาง (Tabu Move) รวมถึงการปรับปรุงค่าที่จะทำให้คำตอบย้ายเข้าสู่ตำแหน่งก่อนหน้า เป็นการป้องกันการเกิดการวนรอบ (Short Term Cycling) โดยใช้ TABU List บันทึกการย้ายตำแหน่ง และเก็บ

คำตอบไว้เสมอ ดังนั้น ทุกๆ Iteration จะเกิดการปรับปรุงค่าคำตอบอย่าง TABU แม้ว่าค่าคำตอบจะไม่ดีขึ้น แต่ยังคงเก็บค่าคำตอบจากทุกๆ Iteration ไว้และเลือกค่าที่ดีที่สุดจากจำนวนการปรับปรุงค่าที่ตั้งไว้เป็นค่า Heuristic Optimum พารามิเตอร์ที่สำคัญของวิธีการนี้คือการกำหนดขนาดของ TABU List (TABU List Size) ซึ่งค่าที่เหมาะสมขึ้นอยู่กับปัญหา ถ้าขนาดของ TABU List มากเกินไปจะทำให้คำตอบที่ได้มีค่าไม่ดี แต่ถ้าขนาดของ TABU List น้อยเกินไปจะทำให้คำตอบยังคงวนอยู่ในค่า Local Optimum




รูปที่ 2.5 Alternative Neighborhoods

ข. Simulated Annealing : SA อัลกอริทึมของ Simulated Annealing คือการยอมรับการเคลื่อนที่ที่ไม่ปรับปรุงค่าคำตอบให้ดีขึ้น โดยการทดสอบความน่าจะเป็นจากค่าสุ่มเพื่อหลีกเลี่ยงการวนรอบซ้ำ เริ่มต้นจากคำตอบที่เป็นไปได้ (Feasible Solution) กระบวนการเคลื่อนที่ของแต่ละ Iteration เริ่มต้นจากค่าสุ่มซึ่งได้มาจากการเคลื่อนที่ครั้งก่อนหน้า ซึ่งโดยทั่วไปไม่ได้พิจารณาเพียงค่าฟังก์ชันวัตถุประสงค์ (Objective Function) แต่จะคำนวณเป็น ค่าฟังก์ชันวัตถุประสงค์สุทธิ (Net Objective Function Improvement ; obj) ถ้าคำตอบที่ได้มีค่าดีขึ้น ( obj > 0) จะยอมรับคำตอบนั้นทันที หรือในกรณีที่คำตอบที่ได้มีค่าไม่ดีขึ้น ( obj <= 0) จะยอมรับคำตอบด้วยความน่าจะเป็นดังนี้

$$\text{Probability of acceptance} = e^{\frac{\Delta \text{ obj}}{q}}$$

ค่าที่ดีที่สุดจะถูกเก็บเอาไว้จะเป็นค่า Heuristic Optimum ซึ่งค่าการลดค่าคำตอบ ( obj <= 0) ในความน่าจะเป็นแบบเอ็กโปเนนเชียลจะแสดงถึงปริมาณที่ลดลงของค่าฟังก์ชันวัตถุประสงค์เทียบกับคำตอบจากการปรับปรุงค่า ดังนั้นการปรับปรุงค่าครั้งใดๆ ที่ให้ค่าคำตอบน้อยย่อมถูกยอมรับด้วยโอกาสน้อยกว่าการปรับปรุงค่าครั้งที่มี

ค่ามากกว่าเสมอ พารามิเตอร์ q เรียกว่าค่าควบคุมอุณหภูมิของการค้นหาคำตอบ (Tempearature Controlling) เป็นตัวกำหนดความสุ่มของการปรับปรุงค่า ถ้าพารามิเตอร์ q มีค่ามากจะทำให้สมการมีค่าเข้าใกล้ศูนย์ หมายความว่าโอกาสที่จะยอมรับการเคลื่อนที่เท่ากับ 1 เสมอแม้จะไม่มีการปรับปรุงค่าคำตอบและถ้าพารามิเตอร์ q มีค่าน้อย โอกาสที่จะยอมรับการเคลื่อนที่ที่ไม่มีการปรับปรุงคำตอบก็จะลดลง

การนำ SA ไปใช้ในการหาคำตอบจะเริ่มต้นจากค่าพารามิเตอร์ q มากก่อนแล้วค่อยๆลดค่าลงจนเป็นศูนย์ตลอดช่วงการทำงานของการค้นหาคำตอบ โดยแนวคิดของวิธีการนี้ก็คือ ให้การหาคำตอบสามารถขยายการค้นหาคำตอบ (Explore)ได้อย่างทั่วถึงในขั้นแรกและจำกัดพื้นที่สำหรับการค้นหาลงมาให้เหลือเฉพาะพื้นที่ที่มีศักยภาพในภายหลัง (Exploit) และโดยปกติแล้วจำนวนครั้งของการปรับปรุงค่าโดยวิธีนี้จะสูงกว่าวิธี TABU Search หรือ Hill Climbing และมีการปฏิเสธการปรับปรุงค่าน้อยกว่ารวมถึงใช้ Neighborhood จำนวนน้อยกว่าในการพิจารณา เมื่อเทียบกับวิธีแบบ Local Search ที่มักทดสอบกับ Neighborhood เกือบทุกค่า จากการศึกษาพบว่าจำนวนครั้งของการปรับปรุงค่าอย่างสุ่มที่มากพอจะทำให้ SA หาคำตอบที่ดีได้

ค. Genetic Algorithms :GAs เป็นวิธีแก้ปัญหาที่พัฒนามาจากวิธีการแบบ Evolutionary Algorithm ซึ่งใช้แก้ปัญหาประเภท Stochastic Optimization โดยการประยุกต์หลักการคัดเลือกโดยธรรมชาติและหลักการพันธุศาสตร์ในการหาคำตอบ เริ่มต้นโดยการสุ่มชุดของคำตอบเริ่มต้น (Initial Solution) ขึ้นมาเป็นประชากร (Population) และสำหรับแต่ชุดของคำตอบเริ่มต้น ซึ่งเรียกว่าโครโมโซม (Chromosome) ประกอบไปด้วยคำตอบของปัญหาในขณะนั้นๆ โดยการปรับปรุงชุดคำตอบ (Population Solution) ตลอดการค้นหาคำตอบ โดยประชากรบางส่วนจะถูกเลือกโดยคำนึงถึงค่าความเหมาะสมของคำตอบ (Fitness Value) ซึ่งส่งผลทำให้โอกาสในการถูกเลือกสำหรับประชากรแต่ละตัวไม่เท่ากัน จากนั้นคู่ของประชากรที่ได้รับการคัดเลือกจะแลกเปลี่ยนชุดตัวแปรของคำตอบย่อยในตำแหน่งเดียวกันเพื่อทำการปรับปรุงค่าคำตอบ เรียกขั้นตอนการปรับปรุงนี้ว่า Crossover จากนั้นจะทำการเลือกคู่ของประชากรสำหรับรุ่นถัดไป (Offspring) โดยทำซ้ำตามขั้นเดิม และในบางครั้งก็จะทำการปรับปรุงค่าของคำตอบด้วยวิธีการ Mutation โดยการเปลี่ยนค่าของตัวแปรอย่างสุ่มภายในประชากรตัวใดๆ โดยที่ความน่าจะเป็นในการเลือกคู่ประชากรใดๆ คำนวณจากสมการดังนี้

$$\text{probability} = \frac{\text{Individual solution value}}{\text{population total}}$$

ในทางทฤษฎีจะเห็นว่า Genetic Algorithms เป็นวิธีการหาคำตอบที่ไม่ขึ้นกับ Neighborhood แต่การปรับปรุงคำตอบด้วยวิธีการ Crossover ไม่สามารถประกันได้ว่าคำตอบที่ได้มาเป็นประชากรในรุ่นถัดไปจะเป็นคำตอบที่เป็นไปได้ และวิธีการจัดตำแหน่งตัวแปรคำตอบสำหรับประชากรก็มีผลอย่างมากต่อกระบวนการ Crossover ขึ้นอยู่กับว่าจุดตัดจะอยู่ในตำแหน่งใด นอกจากนี้วิธีการกำหนดความน่าจะเป็นสำหรับการถูกคัดเลือกเป็นประชากรก็เหมาะสมเฉพาะกับปัญหาแบบหาค่ามากที่สุดเท่านั้น[4]

## 2.3 ผลงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้จะกล่าวถึงงานวิจัยในอดีตที่เกี่ยวข้องซึ่งสามารถได้เป็น 3 หมวด คือ ปัญหาการกำหนดเส้นทางการขนส่งภายใต้กรอบเวลา (Vehicle routing problem with time window : VRPTW) ปัญหาการกำหนดที่ตั้งของสถานประกอบการ (Facility Location Problem) และปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง

### 2.3.1 ปัญหาการกำหนดเส้นทางการขนส่งภายใต้กรอบเวลา (Vehicle routing problem with time window : VRPTW)

ปัญหาการกำหนดเส้นทางการขนส่งภายใต้กรอบเวลา เป็นปัญหาที่ได้รับความสนใจและนำไปใช้ในการจัดการการขนส่งด้านต่างๆ เช่น เส้นทางเดินรถโรงเรียน การส่งจดหมาย การส่งหนังสือพิมพ์ การขนส่งน้ำมัน โดยในการแก้ปัญหามีจุดประสงค์คือ เพื่อทำให้ค่าใช้จ่ายในการขนส่งมีค่าต่ำที่สุด คือมีค่าใช้จ่ายในการเดินทางต่ำที่สุดและจำนวนของยานพาหนะน้อยที่สุด โดยไม่ละเมิดข้อจำกัดของระบบ การมีเส้นทางการขนส่งที่ดีนั้นสามารถประหยัดค่าใช้จ่ายได้มาก เช่น ในสหรัฐอเมริกาทั้งในส่วนรัฐบาลและภาคอุตสาหกรรม สามารถประหยัดค่าใช้จ่ายได้ถึง 1,000,000ดอลลาร์ การสำรวจเกี่ยวกับการจำแนกชนิดและการประยุกต์ใช้ของ VRPTW สามารถค้นพบในงานของ Osman [5], Laporte [6], Fisher [7] และ Bodin et al. [8]

ในอดีตนั้นได้มีการแสดงให้เห็นถึงการหาคำตอบที่เป็นไปได้สำหรับ การแก้ปัญหาการกำหนดเส้นทางของพนักงานขาย (traveling salesman problem : TSP) ซึ่งเป็นปัญหาที่เรียบง่ายที่สุดในปัญหาการกำหนดเส้นทางการขนส่ง ว่าเป็นปัญหาแบบเอ็นพีอย่างสมบูรณ์ (NP-complete) ดังนั้นจะเห็นได้ว่าการแก้ปัญหา VRPTW มีความซับซ้อน

มากกว่าตั้งแต่ในด้านที่ต้องให้บริการลูกค้าหลายราย โดยใช้ยานพาหนะหลายคัน และถึงแม้ว่าเราสามารถแก้ปัญหา VRPTW เพื่อให้ได้คำตอบที่ดีที่สุด (Optimal solution) โดยใช้วิธีแบบแม่นตรง (exact methods) แต่เวลาที่ใช้นั้นก็มาก [9] เมื่อวิธีการทางฮิวริสติกสามารถที่จะสร้างคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุด ภายใต้เวลาที่ไม่นานเกินไป ทำให้มีหลายงานวิจัย ที่ทำการออกแบบฮิวริสติกใหม่ๆ สำหรับแก้ปัญหา VRPTW

มีงานวิจัยมากมายที่ได้ทำการพัฒนาฮิวริสติกสำหรับแก้ปัญหา VRPTW ยกตัวอย่างเช่น Christofides [10], Fisher [11], Federgruen and Simchi-Levi [12], Bertsimas and Simichi-Levi [13],Kontoravdis and Bard [14], Kohl and Madsen [15], Kohl et al. [16], Thompson and Psaraftis [17] หรือ Kolen et al. [18] โดยที่ฮิวริสติกเหล่านี้ สามารถจำแนกได้เป็น 4 ประเภทโดยแบ่งเป็น Constructive methods, Route first-cluster second method, Cluster first route second methods และ Incomplete optimization methods

ฮิวริสติก saving algorithm ของ Gaskel [19], Yellow [20] และ Russel [21] ถูกจัดอยู่ในประเภท Constructive methods The optimal partitioning heuristic [22] และ The sweep algorithm [23] ถูกจัดอยู่ในประเภท Route first-cluster second method โดยสำหรับประเภท Cluster first route second methods มี3 อัลกอริทึม ซึ่งรวมถึง two-phase method [24], generalized assignment heuristic [25] และ location-based heuristic [26] สำหรับประเภทสุดท้าย Incomplete optimization methods นั้นรวมถึงวิธี cutting plane methods [27] และ minimum K-tree methods [28]

อัลกอริทึมที่กล่าวมาทั้งหมดมีความสามารถที่จะให้คำตอบที่เป็นไปได้ สำหรับปัญหา VRPTW แต่คำตอบที่ได้อาจไม่เป็นที่น่าพอใจ เพราะความสามารถที่จำกัดในการค้นหาพื้นที่ของคำตอบ (solution space) เมื่อไม่นานมานี้ได้มีการนำฮิวริสติกเพื่อการค้นหา (Heuristics search) อย่างเช่น เจเนติกอัลกอริทึม(GA) และ วิธีค้นหาแบบตาบู (TS) (อัลกอริทึมที่มีพื้นฐานบนปัญญาประดิษฐ์ (AI))มาใช้เพื่อปรับปรุงการแก้ปัญหา VRPTW เช่น Thangiah และ Petrovic [29] Thangiah et al. [30]และ Thangiah [31] และมีการพิสูจน์ว่า วิธีฮิวริสติกเพื่อการค้นหา ที่พื้นฐานอยู่บน เจเนติกอัลกอริทึม [32, 33] สามารถแก้ปัญหาได้ดีสำหรับ VRPTW

2.3.2 ปัญหาการกำหนดที่ตั้งของสถานประกอบการ (Facility Location Problem)

ปัญหาการกำหนดที่ตั้งของสถานประกอบการ เป็นปัญหาที่สำคัญปัญหาหนึ่งของการตัดสินใจเกี่ยวกับระบบขนส่ง โดยเป็นการตัดสินใจว่าสถานประกอบการ แต่ละแห่งควรถูกกำหนดให้อยู่ที่ไหน โดยจุดประสงค์ของการตัดสินใจกำหนดที่ตั้งสถานประกอบการจะขึ้นกับนโยบายหรือความต้องการของวิสาหกิจนั้นๆ เช่น ต้องการให้ครอบคลุมความต้องการมากที่สุดโดยใช้ค่าใช้จ่ายต่ำที่สุด หรือต้องการให้อยู่ใกล้แหล่งความต้องการมากที่สุด เป็นต้น การที่การจัดตั้งหรือก่อสร้างสถานประกอบการ นั้นจะต้องใช้ค่าใช้จ่ายจำนวนมาก การตัดสินใจนี้จึงเป็นการวางแผนตัดสินใจลงทุนระยะยาว ซึ่งหากสภาพแวดล้อมของการแข่งขันเปลี่ยนไป อาจทำให้ที่ตั้งในปัจจุบันไม่เป็นที่ตั้งที่เหมาะสมอีกก็เป็นได้ ดังนั้นการตัดสินใจกำหนดที่ตั้งสถานประกอบการ จึงเป็นความท้าทายที่สำคัญด้านกลยุทธ์การแข่งขันเช่นกัน

ในงานวิจัยของ Owen and Daskin [34] เป็นการสำรวจงานวิจัยต่างๆเกี่ยวกับปัญหาการกำหนดที่ตั้งของศูนย์อำนวยความสะดวก ซึ่งกล่าวถึงปัญหาและตัวแบบทางคณิตศาสตร์สำหรับปัญหาในแบบต่างๆซึ่งสามารถแบ่งได้เป็น 3ประเภทหลักๆ คือ static and deterministic ,dynamic และ stochastic

2.3.3 ปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่ง (Location Routing Problem : LRP)

ปัญหาการหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางการขนส่งนั้นสามารถให้คำอธิบายได้ดังนี้ คือ เซตคำตอบที่เป็นไปได้ของสถานที่ที่สามารถเป็นศูนย์กระจายสินค้าได้ ซึ่งลูกค้าแต่ละคนจะรับการจัดสรรให้กับเซตคำตอบเหล่านี้ ศูนย์กระจายสินค้าจะทำหน้าที่กระจายสินค้า โดยยานพาหนะที่ได้รับมอบหมายจากแต่ละศูนย์กระจายสินค้า และขนส่งไปตามเส้นทางที่กำหนดไว้

รูปแบบทั่วไปของปัญหาการเลือกที่ตั้งและเส้นทางสามารถสร้างรูปแบบปัญหาเป็น Integer Programming หรือ Mixed Integer Programming โดยใช้ 3 index variable แสดงถึงการเชื่อมต่อกันระหว่างจุดความต้องการสินค้าและการจัดเส้นทาง งานวิจัยของ Laporte and Nobert [35] เสนอวิธีการสร้างรูปแบบปัญหาเป็น Integer Linear Programming และแก้ปัญหาด้วย branch and bound base on constraint relaxation technique และในงานวิจัยของ Laporte et al. [36] ได้ใช้เทคนิคที่ Laporte and Nobert [35] พัฒนาขึ้นเพื่อแก้ปัญหา uncapacitated multi-facility location problem โดยไม่พิจารณาข้อจำกัดด้านความจุของรถ ความจุของจุดกระจายสินค้าและ

เวลาในการเดินทาง วิธีการ exact integer program สามารถแก้ปัญหาได้มากถึง 50 โนด

และงานวิจัยของ Bookbinder and Reece [37] ได้พิจารณาข้อจำกัดด้านความจุของรถ,ความจุของจุดกระจายสินค้า (capacitated vehicle and facilities) และเสนอวิธีการสร้างรูปแบบปัญหาเป็น nonlinear mixed integer program ได้ประยุกต์ใช้ Bender's decomposition แยกปัญหาออกเป็นปัญหาที่ตั้งและปัญหาการจัดเส้นทาง

เนื่องจากเวลาที่ใช้ในการคำนวณจะเพิ่มขึ้นอย่างรวดเร็ว เมื่อปัญหามีขนาดใหญ่ขึ้น (Exponential Time) ทำให้ exact method ของปัญหาการเลือกที่ตั้งและเส้นทาง มีข้อจำกัดกับปัญหาขนาดเล็กและขนาดกลางเท่านั้น (ไม่เกิน 20-50 จุดความต้องการ) ดังนั้นจึงนิยมใช้ฮิวริสติกในการหาคำตอบมากขึ้น เช่น Saving Concept , Deterministic heuristic base on location-allocation งานวิจัยของ Madsen[38] ได้สร้างรูปแบบปัญหาในการจัดส่งหนังสือพิมพ์ (2-level newspaper delivery problem) เพื่อพิจารณาถึงที่ตั้งของโรงงาน จุดขนถ่ายสินค้าและเส้นทางการขนส่งไปยังลูกค้า ได้เสนอ 2 Algorithm คือ alternate location-allocation saving และ saving drop procedure

แนวคิดในการสร้างอัลกอริทึมอีกแนวคิดหนึ่งคือการเปิด(ปิด)จุดกระจายสินค้าทั้งหมดก่อนตั้งแต่แรก แล้วจึงค่อยๆปิด(เปิด) ไปทีละ 1 จุดโดยใช้หลักการความประหยัดค่าใช้จ่าย (cost saving) ในการตัดสินใจ พบในงานวิจัยของ Srivastana [39] ได้เสนออัลกอริทึม 3 แบบ คือ SAV1 SAV2 และ CLUS เพื่อแก้ปัญหาการเลือกที่ตั้งและเส้นทางที่ซับซ้อนมากขึ้น และสำรวจปัจจัยต่างๆที่ส่งผลต่อสมรรถนะของอัลกอริทึม SAV1 จะเปิดจุดกระจายสินค้าทั้งหมดตั้งแต่เริ่มต้น และใช้ค่าใช้จ่ายในการขนส่งเพื่อตัดสินว่าควรเปิดหรือปิดจุดใด ปรับปรุงจาก saving algorithm for the multiple depot ของ Tillman (1969) SAV2 เป็นการเปิดจุดกระจายสินค้าเพิ่มขึ้นทีละหนึ่งจุดแล้วใช้ค่าใช้จ่ายในการขนส่งเพื่อตัดสินว่าควรเปิดหรือปิดจุดใด และ CLUST เป็นการสร้าง minimal spanning tree ระหว่างจุดความต้องการสินค้าแล้วแยกออกเป็น cluster ด้วย density search technique ซึ่งจากผลการคำนวณพบว่าอัลกอริทึมทั้งหมดนี้ให้ผลลัพธ์ที่ดีกว่าการใช้ sequential approach (Sequential approach เป็นวิธีที่นิยมใช้ในทางปฏิบัติ ในขั้นแรกจะตัดสินที่ตั้งของจุดกระจายสินค้าจาก Moment sum approximation จากนั้นจึงแก้ปัญหา multi-depot routing ด้วย saving algorithm)

การแบ่งปัญหา LRP ออกเป็นปัญหาย่อยและแก้ปัญหาย่อยนั้นด้วย exact method หรือ heuristic พบในงานวิจัยของ Or and Pierskalla [40]เพื่อหาคำตอบถึงที่ตั้งของคลังโลหิตและเส้นทางการขนส่งเพื่อกระจายโลหิตให้แก่โรงพยาบาล โดยผลิตภัณฑ์ที่ขนส่งคือโลหิต ซึ่งมีการจัดส่ง 2 รูปแบบคือแบบช่วงเวลาและแบบฉุกเฉิน นำเสนอฮิวริ

สติก 2 รูปแบบคือ อัลกอริทึม 1 อัลกอริทึม 2 และใช้ข้อมูลของ Greater Metropolician area ในการทดสอบฮิวริสติก งานวิจัยของ Jossed and Daskin [41]สร้างรูปแบบปัญหา Warehouse location-routing problem(WLRP) เพื่อหาที่ตั้งของศูนย์กระจายสินค้าและเส้นทางเดินรถไปพร้อมๆกัน ได้เสนอ Mixed Integer Programming Formulation ซึ่งไม่สามารถแก้ปัญหาที่มีขนาดใหญ่และซับซ้อนได้ และนำเสนอฮิวริสติกเพื่อแก้ปัญหาโดยแยกปัญหาออกเป็นปัญหาย่อย 3 ปัญหาที่มีผลต่อกัน เพื่อประยุกต์ใช้กับบริษัทขนส่งสินค้า

การใช้ meta-heuristic เพื่อรวมปัญหาทั้งสองเข้าด้วยกันพบในงานวิจัยของ Tuzun and Burke [1] เสนอ two-phase tabu search algorithm เพื่อหาคำตอบของ LRP เพื่อตัดสินใจทั้งที่ตั้งและเส้นทางพร้อมๆกัน โดยแบ่งออกเป็นปัญหาย่อยที่มีตัวแปรการตัดสินใจต่างกัน เมื่อเปรียบเทียบในด้านคุณภาพคำตอบพบว่า two-phase tabu search algorithmให้คำตอบที่มีคุณภาพดีกว่า SAV1 และในด้านเวลาในการคำนวณพบว่า two-phase tabu search algorithm ใช้เวลาในการคำนวณมากกว่า งานวิจัยของ Lin et al.[42] พัฒนา meta-heuristic base on threshold accepting และ Simulated Annealing ประยุกต์ใช้กับการจัดส่งใบเสร็จรับเงินรายเดือนไปยังลูกค้า เพื่อหาที่ตั้งของศูนย์ออกใบเสร็จรับเงิน เส้นทางการจัดส่ง และการโหลดงาน ผลการใช้ meta-heuristic พบว่าสามารถลดค่าใช้จ่ายของบริษัทลงได้ สามารถแก้ปัญหาได้อย่างรวดเร็วและได้คำตอบที่เข้าใกล้ค่าที่เหมาะสม งานวิจัยของ Jayaraman and Ross [43] ได้พัฒนา PLOT Model เป็นลักษณะที่มีหลายผลิตภัณฑ์ มีศูนย์กลางการผลิต 1แห่ง จุดกระจายสินค้า cross docking และลูกค้าหลายราย ผลของระบบเน้นที่ 2 ขั้นตอน คือขั้นตอนการวางแผนซึ่งเกี่ยวข้องกับกระบวนการตัดสินใจเชิงกลยุทธ์ เพื่อเลือกที่ตั้งที่ดีที่สุดของจุดกระจายสินค้าและ cross docking ในขั้นตอนที่ 2 เป็นกระบวนการตัดสินใจในระดับปฏิบัติการเพื่อตัดสินถึงกลุ่มผลิตภัณฑ์ที่จะจัดส่งจากโรงงานไปยังจุดกระจายสินค้าต่อไปยังคลังสินค้า และจัดส่งให้แก่ลูกค้า โดยใช้วิธีการ Simulated Annealing และใช้ข้อมูลจากการดำเนินงานประจำวันของ Walgreens in Houston ในการทดสอบความถูกต้อง (validate) ของการออกแบบโมเดล ข้อจำกัดของงานวิจัยนี้คือข้อสมมติที่มีความต้องการคงที่ เพื่อใช้โมเดลนี้กับสภาวะที่การผลิตมีความแน่นอน

งานวิจัยที่ได้รับการพัฒนาขึ้นใน 30 ปีที่ผ่านมาถูกรวบรวมโดย Min et al.[44] เพื่อวิเคราะห์ถึงวิวัฒนาการของปัญหาที่ตั้งและการจัดเส้นทางและแนวทางในการพัฒนาวิธีการแก้ปัญหาให้เหมาะสมและประยุกต์ใช้ได้กับปัญหาจริง (realistic problem)