運輸與配送

黃山琿

供應鏈管理，物流管理，配送管理?

- 供應鏈管理?also see this
  - jobs in SCM?
- 物流管理? Logistics involves the planning, design, coordination, management and improvement of the processes of moving goods and resources. The primary goal of logistics is to improve the efficiency of internal warehousing and transportation functions and to collaborate with distribution partners to maximize efficiency in the movement of information and goods.

配送管理

- 配送管理系統是管理、運作配送範圍內的配送業務之物流信息系統，對配送相關的運送資源、貨物資源、客戶資源、 人力資源等進行動態最佳化管理，以提供高效率、低成本之智慧化、網路化暨可視化之配送管理系統。See also 1, distribution strategy
- A central difference between logistics and distribution is that logistics encompasses more elements of planning and information flow, whereas distribution more often describes the physical movement of goods.
- Career in Warehousing & Distribution? What kind of job is it(0:50, 2:07)?

配送資訊管理系統

- 配送系統包含以下之資料管理系統：基本資料管理、車輛管理、配送業務管理、客戶管力、聯盟伙伴管理、財務管理、及決策分析系統等等。Illustration for DC, WareHouse
- 基本資料管理：包含貨物、行政區、道路、送貨區域、裝貨點、配送點、貨場(Yard Management)、車、路線資料場等。
- 車輛管理(Fleet management)：包括車輛基本資料管理、駕駛員（司機）管理、裝卸工管理、車輛維修/保養、車輛用油管理等。

  Example: Papa John's Food Service - Fleet Technology

配送資訊管理系統

- 配送業務管理：包括業務受理、裝載計畫、車輛調度、回車管理、事故管理等。
- 客戶管理：包含客戶基本資料、聯絡人資料、合約管理、項目管理、投訴與建議、競爭對手資料等。
- 聯盟伙伴管理：包括聯盟火但基本資料管理、聯絡人管理、合約管理等。
- 財務管理：包括固定大客戶配送價格、散客送貨價格、配送費用結算等。How To Set Up Your Product Price, on wholesale
- 決策分析系統：資料分析及挖掘、排程及預測分析等，並提供各類統計報表。

- 舊有行業，何以老調重彈？
  - 運輸的先進（陸海空機具，貨櫃，新式IT科技，ITS，GPS）
  - 資訊的普及（The Internet，E-Business & E-Commerce）
  - Grainger: stocks about 100,000 SKUs that can be sent to customers within a day of the order being placed.

HTTP://WWW.UPS.COM/CONTENT/TW/ZH/ABOUT/FACTS/WORLDWIDE.HTML

- 包裹業務
- **2015年營收總額 :** 489億美元
- **2015年全球貨運量 :** 47億件包裹和文件
- **每日遞送量 :** 1830 萬件包裹和文件
- **每日美國地區空運量 :** 260萬件包裹和文件
- **每日國際貨運量 :** 270萬件包裹和文件
- **服務涵蓋範圍 :** 超過220個國家及地區，在北美及歐洲遍及所有地區
- **客戶 :** 每日 980萬人 (160萬取件及 840萬送件)
- **線上追蹤 :** 平均每天有6940萬 個追蹤貨物的要求。
- **服務據點 :** 包括 The UPS Store® 4,841個; Mail Boxes Etc.®, 1 個 (全球); UPS 服務中心 1,001個; 授權服務據點 10,602個; UPS Drop Boxes 38,352個; Kiala / UPS Access Point超過 19,000地點
- **營運設備 :** 1,990
- **遞送機車隊 :** 99,984台運輸車、輕型貨車、拖車及摩托車，包括 5,088台油電混合車
- **UPS 噴氣式飛機隊 :** 237
- **包機 :** 412
- **每日飛機班次 :** 美國國內：940；國際：1,015
- **機場服務 :** 美國國內：382；國際：346
- 供應鏈及貨運業務
- **供應鏈及貨運業務2014營收淨額 :** 94 億美元
- UPS Supply Chain Solutions
- **主要服務項目 :** 物流配送, 貨運運輸（空運、海運、陸運、鐵路）, 195國家的貨運承攬代理商, 國際貿易管理, 海關報關行。
- **特殊服務項目 :** 服務零件配送; 技術維修及裝置; 物流設計與規劃; 退貨管理。
- **服務據點 :** 於全球120個以上的國家, 535個以上的服務據點; 32.2萬平方呎。
- **運輸隊伍 :** 541
- UPS Freight
- **主要服務 :** 提供併櫃服務領導者
- **運輸隊伍 :** 5,599運輸車、19,884拖車
- **設施 :** 超過 213間服務中心

HTTP://WWW.DHL.COM.TW/PUBLISH/TW/ZT/ABOUT/TW_ABOUT.HIGH.HTML

- **DHL Express Taiwan**
- 在台成立時間：西元1973年
- 員工人數：1,070
- 全台服務中心數：10
- 全台7-11寄件點：超過4,700點
- 全台台塑加油站寄件點：10點
- 車輛數：超過300部各式車輛
- 服務範圍：全球超過220個國家和地區

- **DHL Global Forwarding** 新加坡商敦豪全球貨運物流股份有限公司臺灣分公司，成立於1980年，現屬於德國郵政集團（Deutsche Post World Net）旗下公司
- 2006年空運總共運送超過186,000筆貨物(55,000噸)
- 2006年海運總共運送超過24,000TEUs
- 主要客戶總數超過8,500
- 員工超過300名
- DHL Global Forwarding 全台有7個據點

HTTP://WWW.TJOIN.COM/ABOUT/INFO.ASP

成立日期：1954年6月
資本額：新台幣49.7億
97年營業額：53.8億
全省據點：121處
員工人數：4,000+人
車輛數：2,000+輛
倉儲面積：120,000+(sq.m)

- 2008年11月21日大榮貨運之重要股東群與嘉里物流分別簽署合資協議書，大榮貨運同日正式成為嘉里物流成員之一。
- **目前嘉里大榮擁有4,000多名員工及3,000餘台車輛**

資訊科技輔助運輸配送：

- Advantages：
  - 速度、數量
- Disadvantages：
  - Security
- 是否引進？
  - 成本
  - 技術（軟硬體）
  - 人員
- 是否準備好接受？（Readiness）
- New Technologies: drone in warehouse, robot in warehouse, drone for delivery, robot for delivery, self-driving truck, machine learning, cloud, sustainable, Artificial Intelligence,Big Data

命

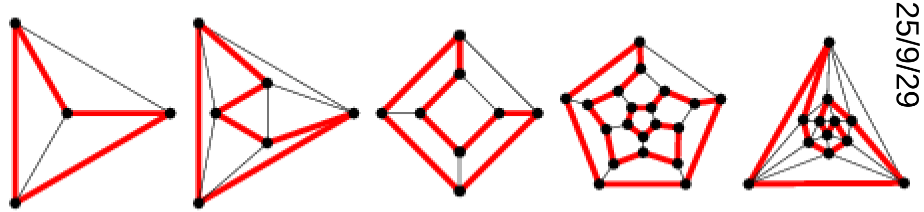
Hamiltonian circuit

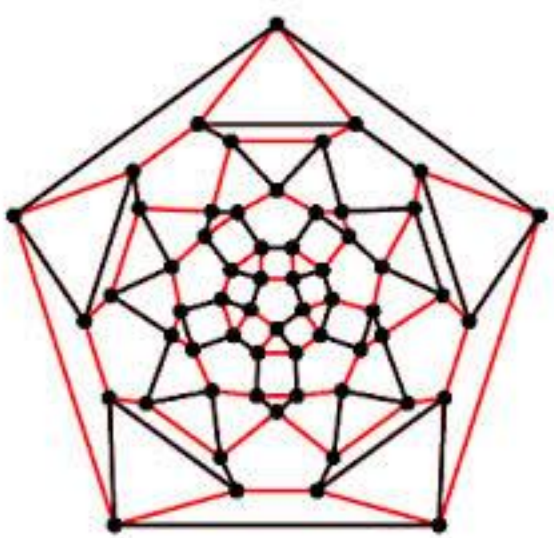
HAMILTONIAN CIRCUIT

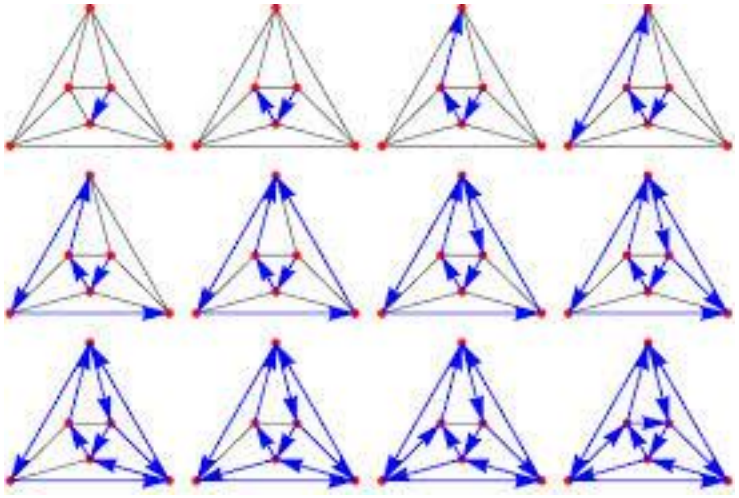
- A Hamiltonian cycle, also called a Hamiltonian circuit, Hamilton cycle, or Hamilton circuit, is a graph cycle (i.e., closed loop) through a graph that visits each node exactly once (Skiena 1990, p. 196). A graph possessing a Hamiltonian cycle is said to be a Hamiltonian graph. By convention, the singleton graph is considered to be Hamiltonian even though it does not posses a Hamiltonian cycle, while the connected graph on two nodes is not.

HAMILTONIAN CIRCUIT

- Hamilton circuit



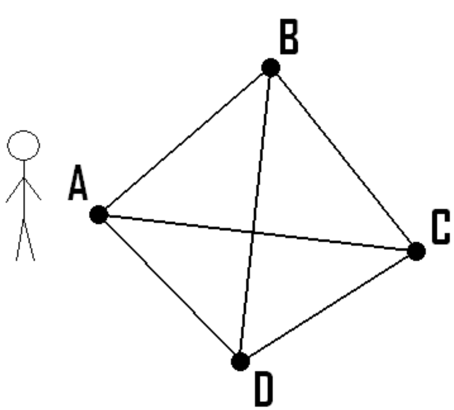


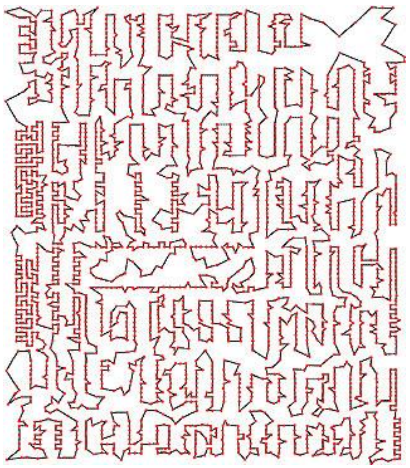


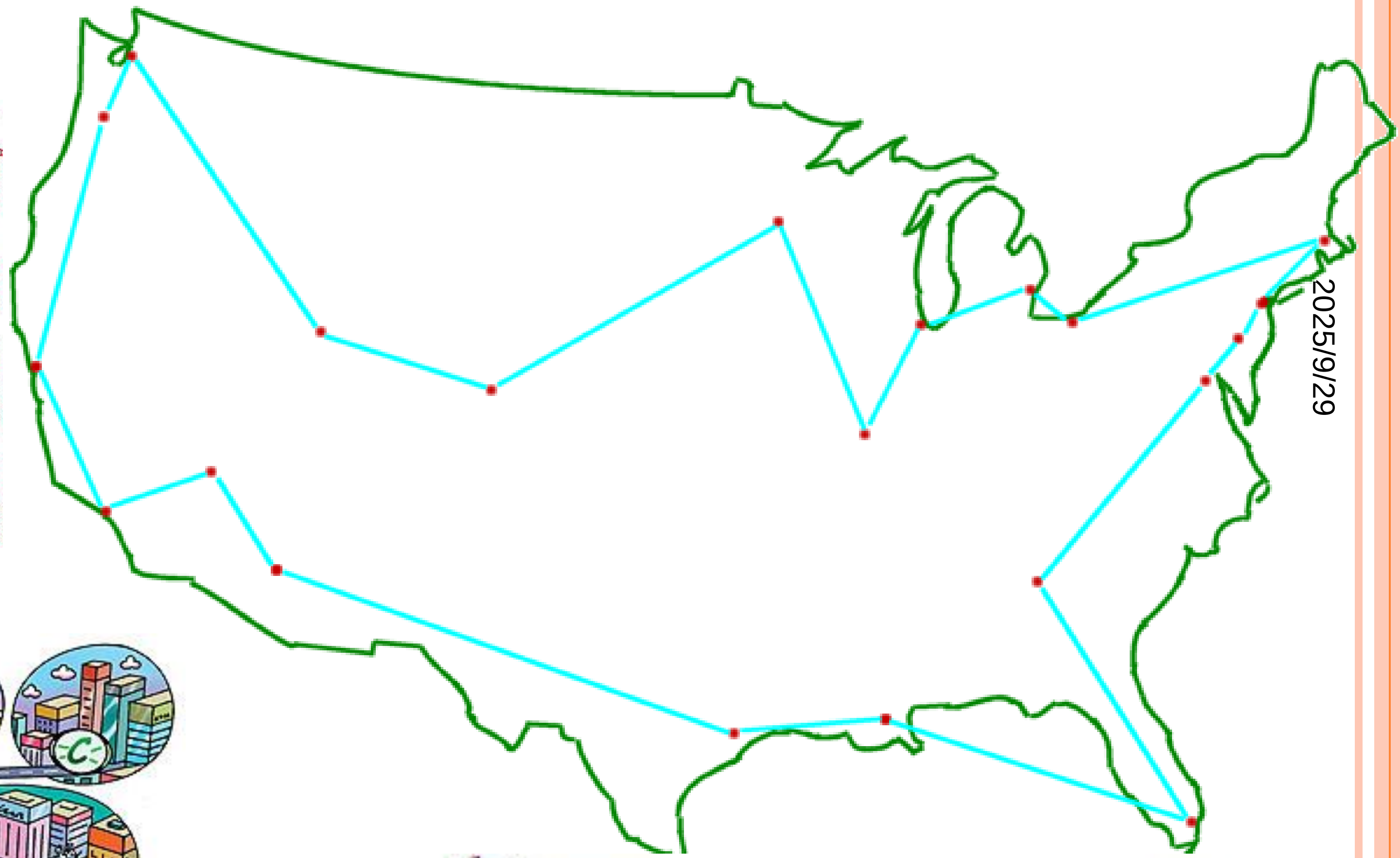
Travel salesman problem

TSP

Travel Salesman Problem (TSP)








A
B
C
D

TSP Math. Program

$$Minimize \quad \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij}$$

$$s.t.$$

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad (j = 1,\ldots,N)$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad (i = 1,\ldots,N)$$

$$x_{ij} = 0,1 \qquad for \quad all \quad i, j$$

- 變數x_{ij}表示是否行經節線（i, j）
- 變數c_{ij}表示節線成本，可為距離、時間或金錢等等。
- N為網路中之所有節點數。

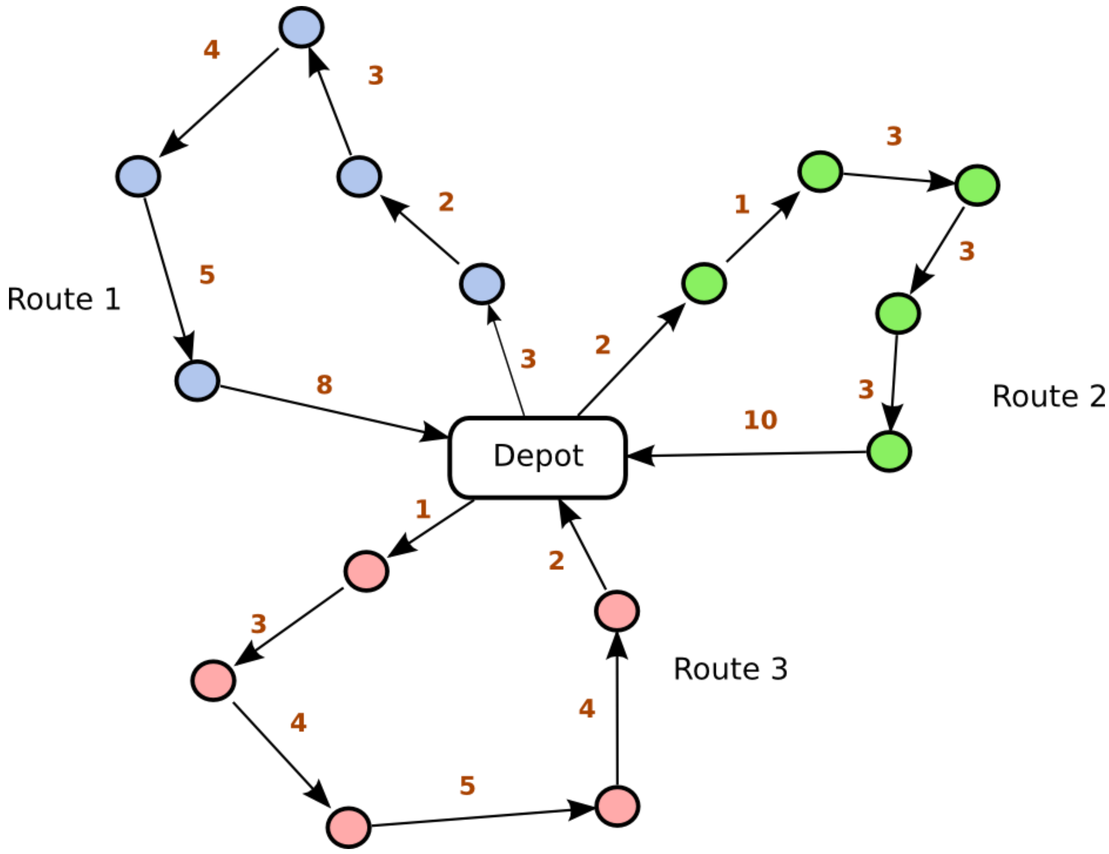
TSP ANALYSIS

- If there are N cities (including the depot) in a full connected network, then the number of feasible solutions is （N-1）！
- 1 2 3 …..N 1
- If the network is symmetric, then (N-1)!/2 answers can be obtained。
- Assume N＝20，then (N-1)!/2＝60822550204416000
- If 1,000,000 answers can be calculated by one computer per second, then 121645100408832000/1000000/60/60/24/365=1928.67 years are needed for obtaining all answers。

Vehicle routing problem

VRP

Vehicle Routing Problem (VRP)(Commercial software)



VRP Constraints -- Capacity

- Capacity
  - Volume
  - Weight
  - Units

VRP Constraints – Time Window

- Time Window
  - Distribution Center Time Window
  - Customer Time Window
  - Hard time window v.s. soft time window

VRP Mathematics programming

$$\min \sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij}$$

subject to

$$\sum_{i\in V} x_{ij} = 1 \quad \forall j \in V\setminus\{0\}; \qquad (1)$$

$$\sum_{j\in V} x_{ij} = 1 \quad \forall i \in V\setminus\{0\}; \qquad (2)$$

$$\sum_{i\in V} x_{i0} = K; \qquad (3)$$

$$\sum_{j\in V} x_{0j} = K; \qquad (4)$$

$$\sum_{i\notin S}\sum_{j\in S} x_{ij} \geq r(S) \quad \forall S \subseteq V\setminus\{0\}, S \neq \emptyset; \qquad (5)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in V. \qquad (6)$$

- Constraints 1 and 2 state that exactly one arc enters and exactly one leaves each vertex associated with a customer, respectively. Constraints 3 and 4 say that the number of vehicles leaving the depot is the same as the number entering. Constraints 5 are the capacity cut constraints, which impose that the routes must be connected and that the demand on each route must not exceed the vehicle capacity. Finally, constraints 6 are the integrality constraints. An alternative formulation may be obtained by transforming the capacity cut constraints into generalized sub-tour elimination

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - r(s)$$

- which imposes that at least *r(s)* arcs leave each customer set *S*.[2]

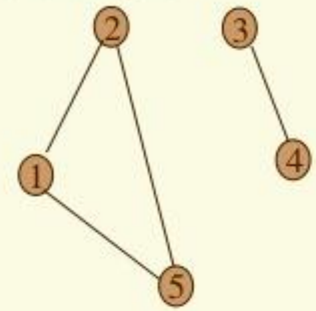
Sub-tour elimination constraint

Traveling Salesman Problem

Subtour elimination constraints for the example:

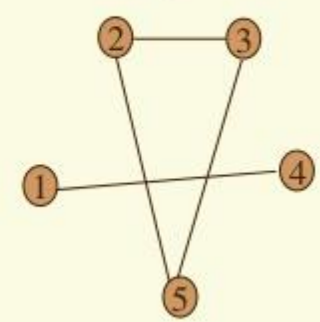
$$S = \{1,2,5\},\ \bar{S} = \{3,4\}$$

$$x_{13} + x_{14} + x_{23} + x_{24} + x_{53} + x_{54} \geq 1$$



$$S = \{1,4\},\ \bar{S} = \{2,3,5\}$$

$$x_{12} + x_{13} + x_{15} + x_{42} + x_{43} + x_{45} \geq 1$$



VRP CLASSIFICATION

Deterministic

- Time window
- Travel time (dynamic?)
- Demand (multi-item)
- # of depot
- Different Kinds of vehicles
  - With trailer
  - co-operate vehicle
- Multiple visits?
- Pickup & delivery
- Periodic

Stochastic

- Uncertain demand
- Stochastic demand(multi-item)
- Stochastic travel time
- Uncertain service time

Others

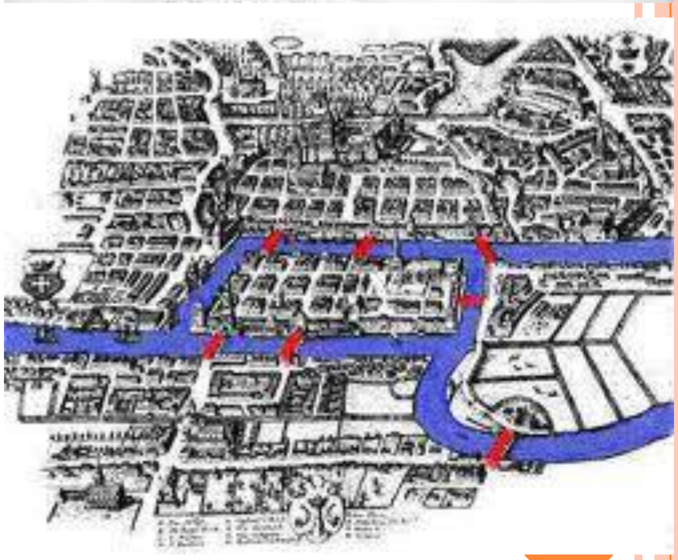
* Backhauls
* Open ended
* Limited number of vehicles
* Homogeneous vs. Heterogeneous

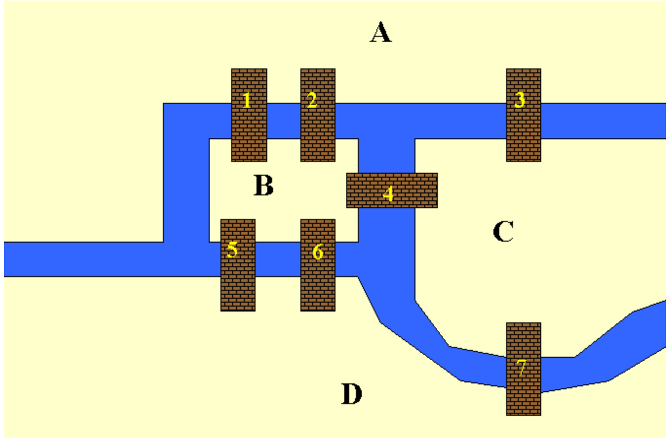
VRP TYPES – EXAMPLES

- **VRPTW** (time window)
- **DVRP** (dynamic)
  - *Traffic speed*
  - *Customer uncertainty*
- **MDVRP**
  - *multiple depot*
- **FSMVRP** (Fleet Size and Mix Vehicle Routing Problem)
  - Heterogeneous vehicles
- **PVRP**
  - Periodic planning
- **PDVRP** (Pick-up and Delivery Vehicle Routing Problem)
- **SVRP** (Stochastic Vehicle Routing Problem)
  - including customer appear, customer demand and traffic flow
- **SVRP** (Split VRP)

Euler circuit

- Euler circuitan: **Eulerian path** is a path in a graph which visits each edge exactly once. Similarly, an **Eulerian circuit** is an Eulerian path which starts and ends on the same vertex.
- Chinese Postman Problem
- Rural Postman Problem
- Windy Postman Problem





Chinese Postman Problem

- Directed Graph
- Windy Chinese Postman Problem
- Mixed Graph

The directed Chinese Postman Problem

Given a strongly connected directed graph $G = (V, A)$, with nonnegative costs c_{ij} for each $(i, j) \in A$, the *Directed Chinese Postman Problem* (DCPP) is the problem of finding a minimum cost tour passing through each arc in A at least once.

In order to formulate the problem we introduce the following notation. For each vertex i, let $\delta^+(i)$ denote the set of arcs leaving i, $\delta^-(i)$ the set of arcs entering i, and $s_i = |\delta^-(i)| - |\delta^+(i)|$. For each arc $(i, j) \in A$, let x_{ij} be the number of copies of each arc that must be added to G in order to obtain an Eulerian graph. Edmonds and Johnson [13] showed that the DCPP can be formulated as the following linear program (LP):

$$(4.1) \quad \text{(DCPP)} \quad \min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$(4.2) \quad \sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = s_i \qquad \forall i \in V,$$

$$(4.3) \quad x_{ij} \geq 0 \qquad \forall (i, j) \in A.$$

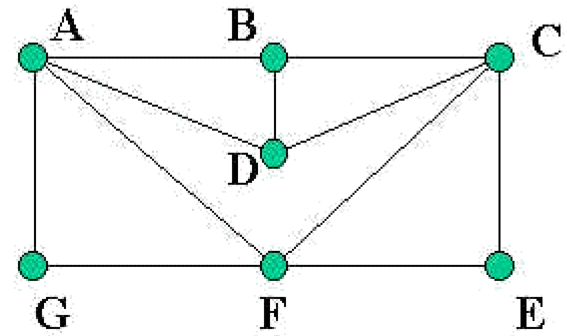
This corresponds to a minimum cost flow problem on G with supplies s_i for vertices i with $s_i > 0$, and demands $-s_i$ for vertices i with $s_i < 0$ (Liebling [26], Edmonds and Johnson [13]).

*Angel Corberan,Gilbert Laporte. (2014) **Arc Routing: Problems, Methods, and Applications**

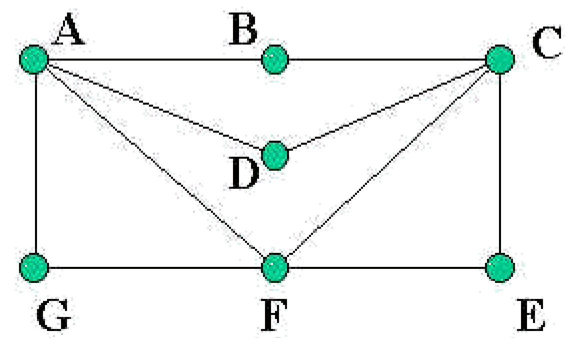
Arc Routing Problem

Arc Routing is the process of selecting the best path in a network based on the route. Contrary to normal routing problems, which usually involve mapping a route between nodes, arc routing focuses more heavily on the route itself. The goal of many arc routing problems is to produce a route with the minimum amount of dead mileage, while also fully encompassing the edges required. Examples of arc routing applications include garbage collection, road gritting, mail delivery, network maintenance, and snowploughing.

Arc Routing Problem



Euler path: **OK**. Euler circuit: **NO**.



One possible solution: **AGFADCFECBA**

Arc Routing Examples

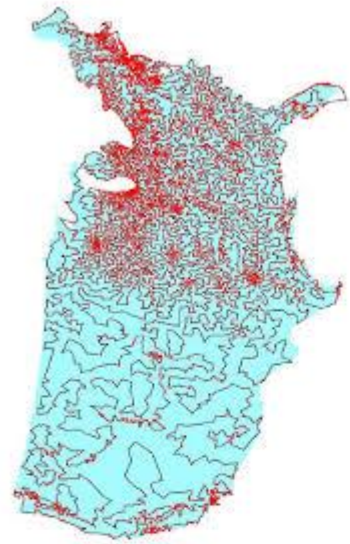
Arc Routing Constrains

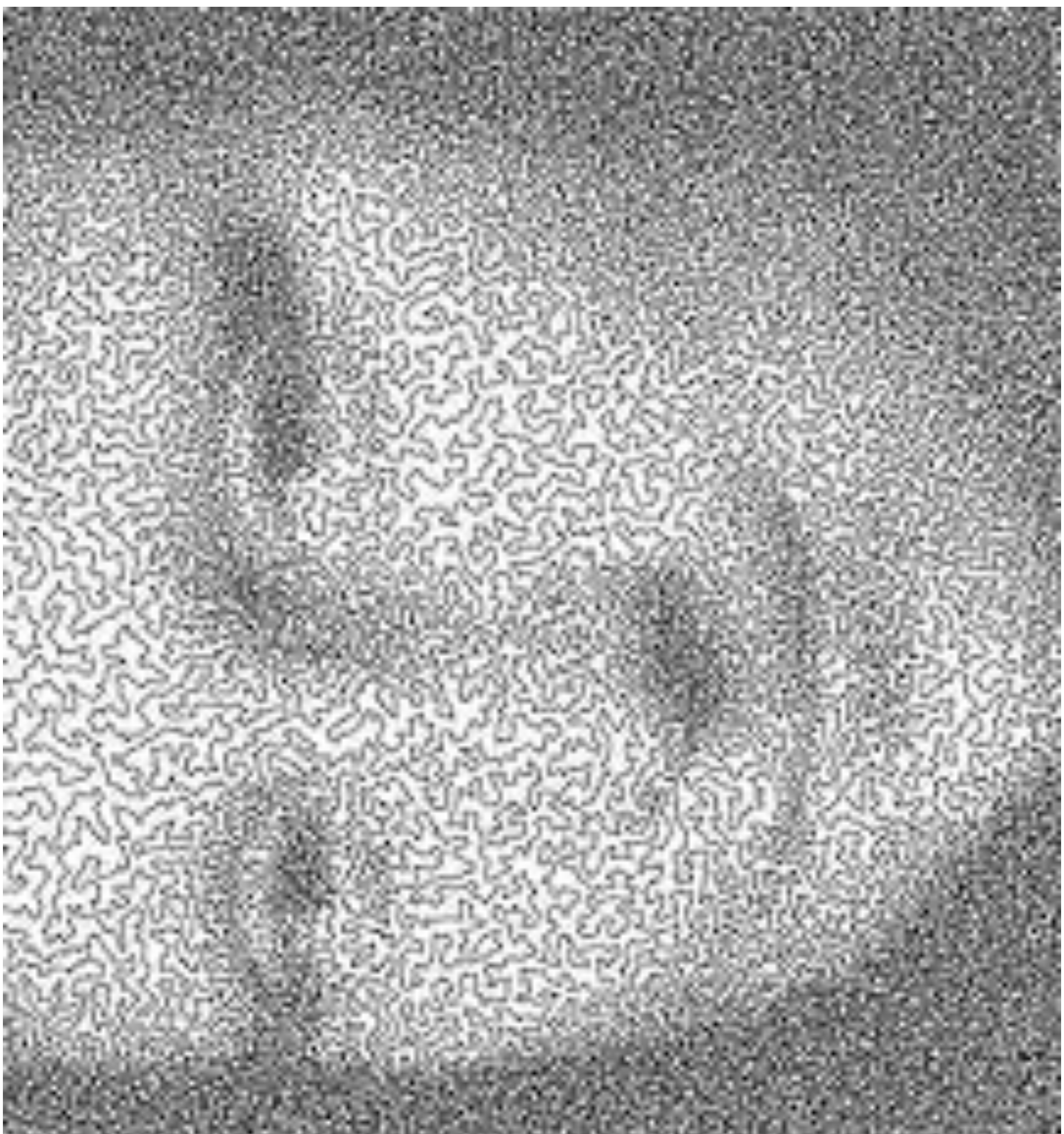
- Capacity
- Time
- Times of passing for an arc
- Turn penalties (U-turn)

Methodologies

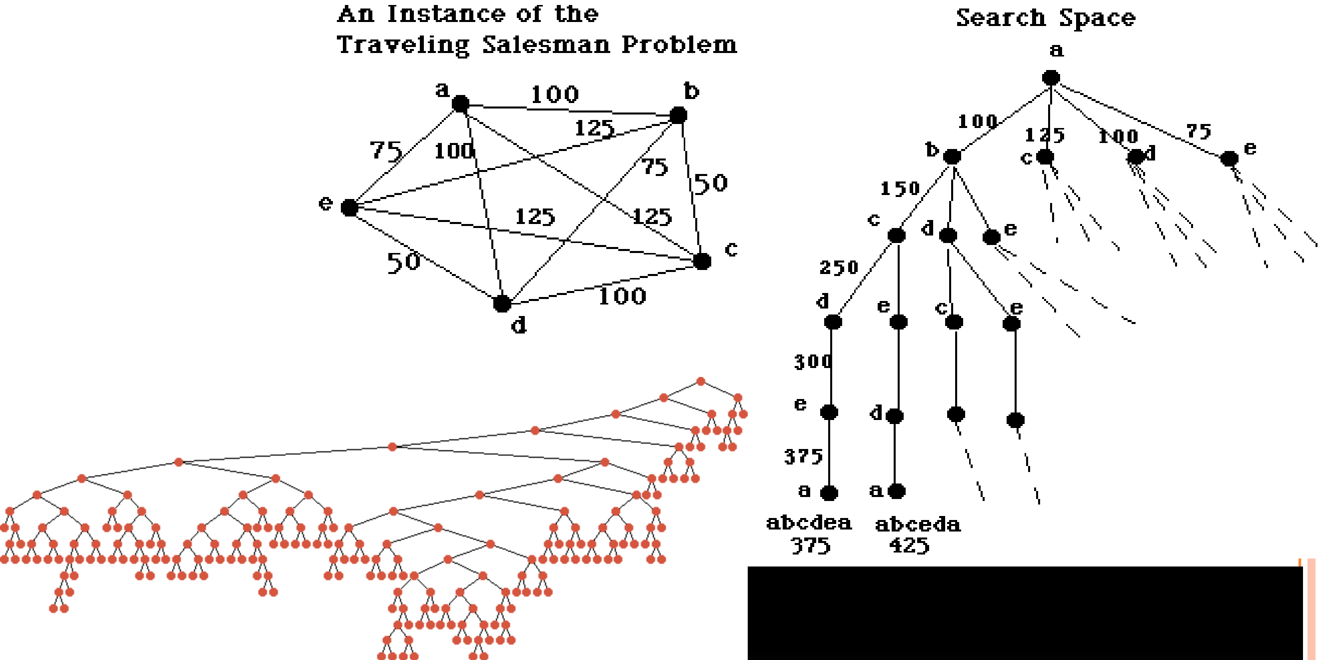
窮舉法（Enumerative Approach）

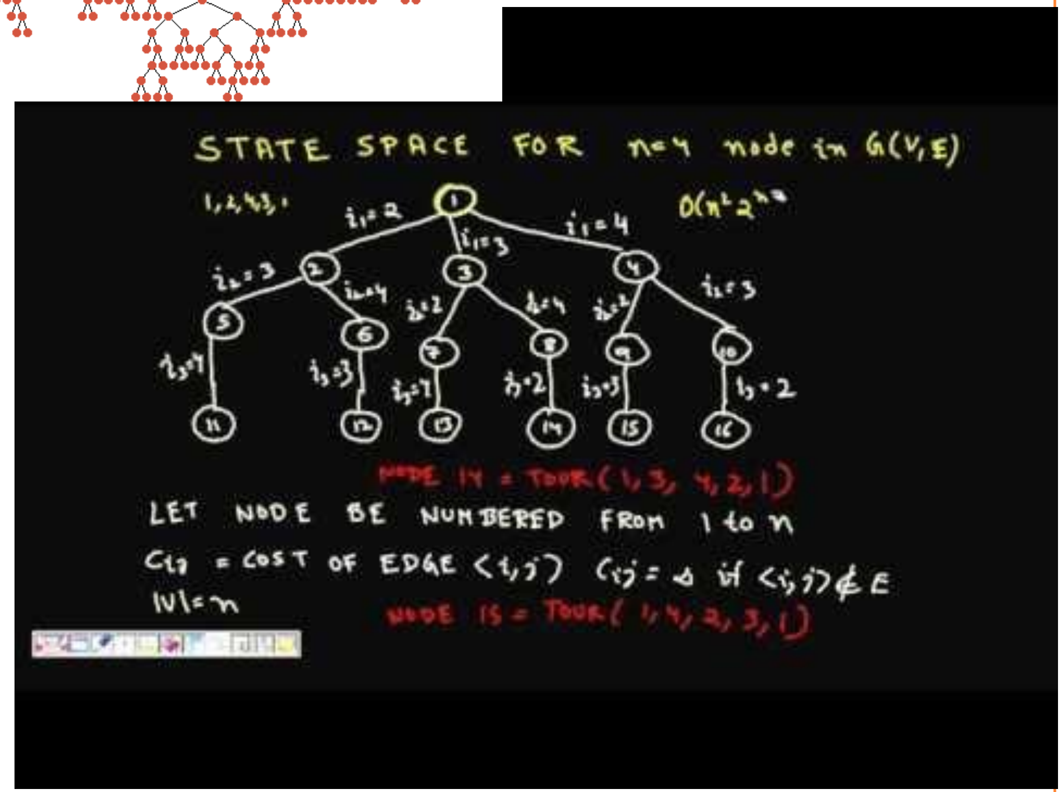
簡單但無效率



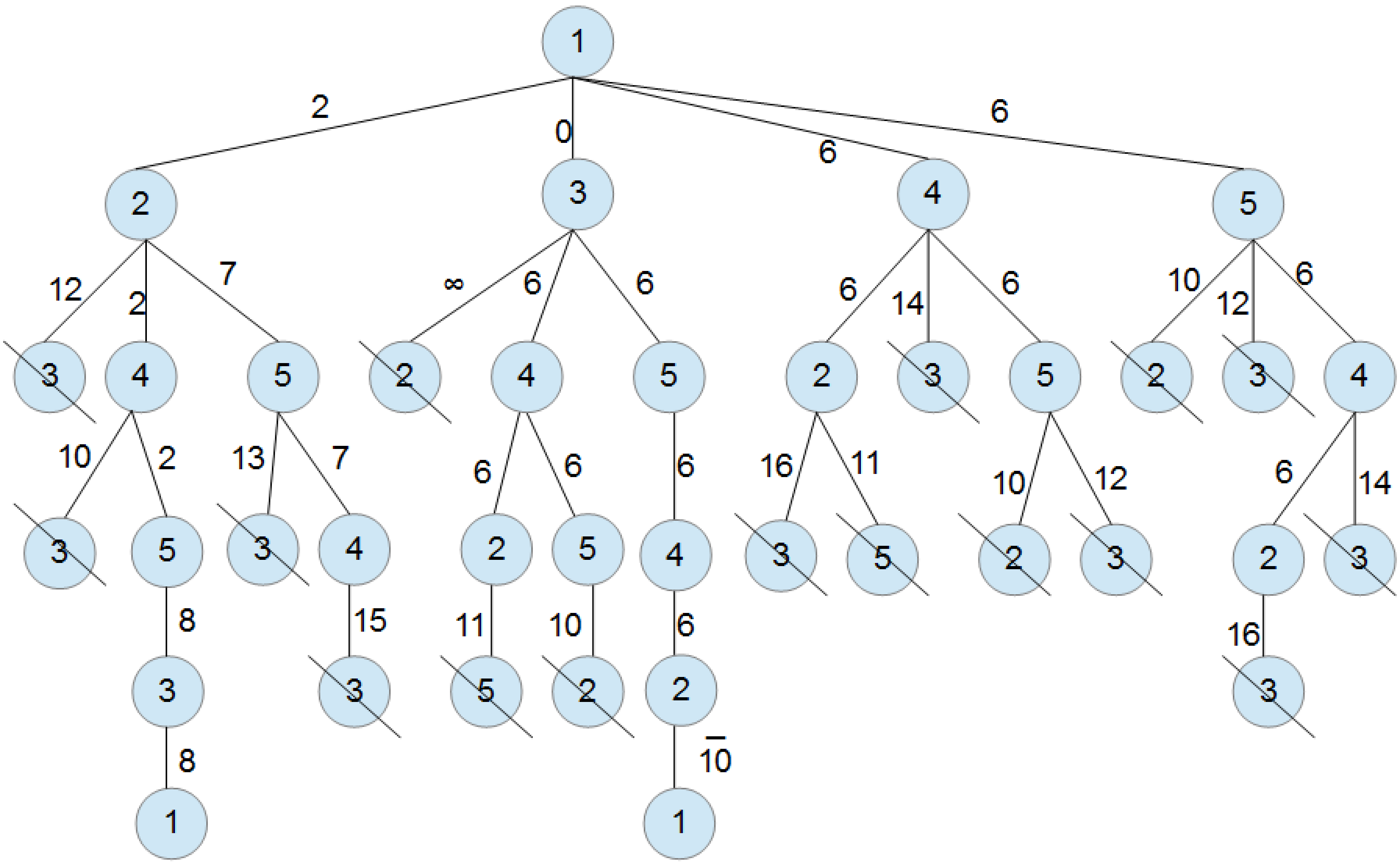


分枝界限法（Branch and Bound）





- 混合整數規劃問題：利用逐漸縮小上限（Upper Bound）與下限（Lower Bound）間的差距來搜尋最佳解。
- TSP: 分枝界限法把問題的可行解展開如樹的分枝，再經由各個分枝中尋找最佳解



1
2
0
6
6
2
3
4
5
12
2
7
∞
6
6
6
14
6
10
12
6
3
4
5
2
4
5
2
3
5
2
3
4
10
2
13
7
6
6
6
16
11
10
12
6
14
3
5
3
4
2
5
4
3
5
2
3
2
3
8
15
11
10
6
16
3
3
5
2
2
3
8
$\overline{1}0$
1
1

鬆弛法（Relaxation）

藉由放鬆原問題之某些限制條件來產生下陷值的方法，例如線性鬆弛（Linear Relaxation），拉氏鬆弛（Lagrangean Relaxation）

General Example

$$
\begin{aligned}
Z = \min \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& Dx \le e \\
& x \ge 0, x \text{ integer}
\end{aligned}
$$

$$
\begin{aligned}
L(u) = \min \quad & c^T x + u(Ax - b) \\
\text{s.t.} \quad & Dx \le e \\
& x \ge 0,\ x \text{ integer}
\end{aligned}
$$

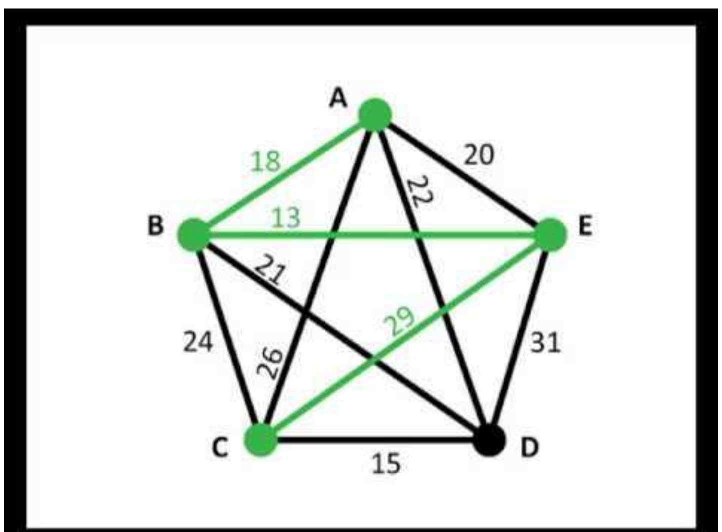
$$\max L(u)$$

Tour Construction

- 根據Bodin et al.
  - 路線建構（Tour Construction）：根據路往之距離或成本矩陣直接產生較佳的TSP可行解，常見的方法如：
    - 插入法（Insertion procedures）：如最近插入法、最省插入法、隨意插入法、最遠插入法、最大角度插入法等。。

Find an efficient circuit between the cities below using the Nearest Neighbor Algorithm starting in Portland

	Ashland	Astoria	Bend	Corvallis	Crater Lake	Eugene	Newport	Portland	Salem	Seaside
Ashland	-	374	200	223	108	178	252	285	240	356
Astoria	374	-	255	166	433	199	135	95	136	17
Bend	200	255	-	128	277	128	180	160	131	247
Corvallis	223	166	128	-	430	47	52	84	40	155
Crater Lake	108	433	277	430	-	453	478	344	389	423
Eugene	178	199	128	47	453	-	91	110	64	181
Newport	252	135	180	52	478	91	-	114	83	117
Portland	285	95	160	84	344	110	114	-	47	78
Salem	240	136	131	40	389	64	83	47	-	118
Seaside	356	17	247	155	423	181	117	78	118	-

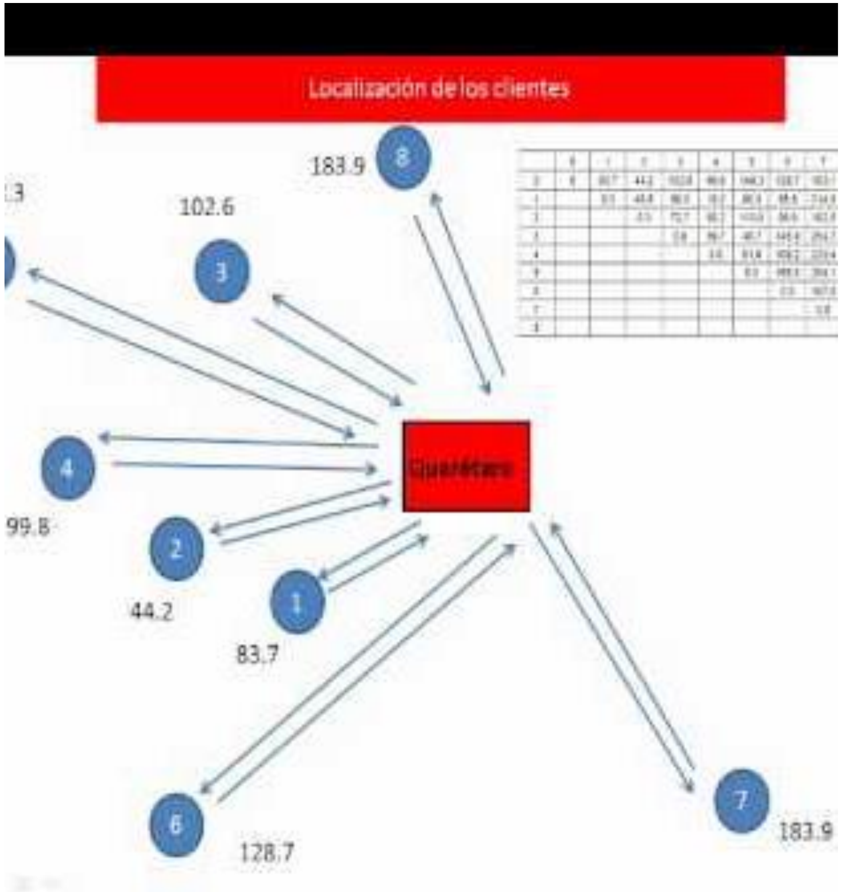


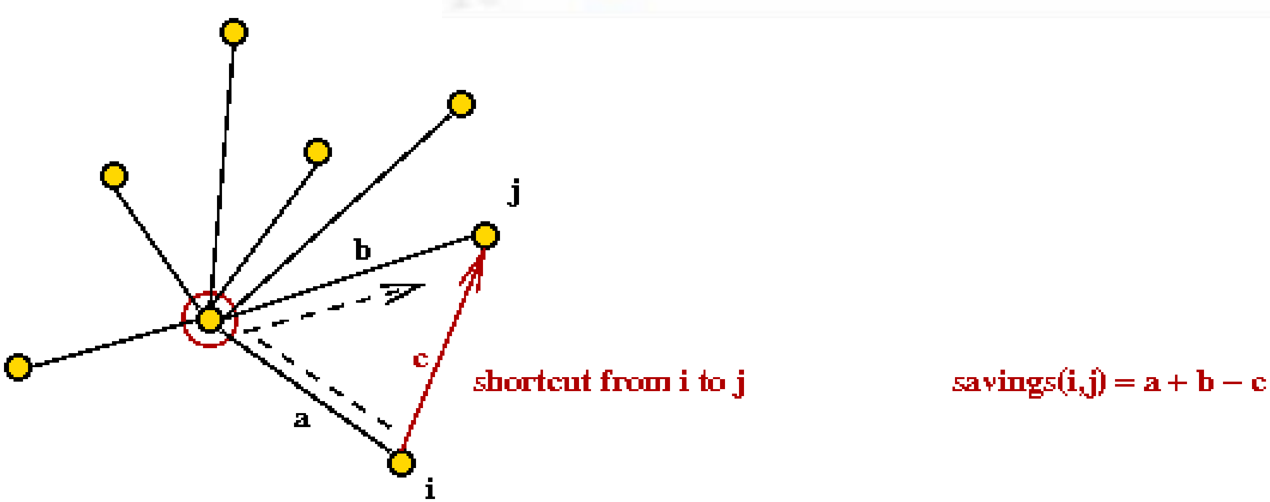
最近鄰點法（Nearest Neighbor Procedure）：

一開始以尋找離場站最近的需求點為起始路線的第一個顧客，此後尋找離最後加入路線的顧客最近的需求點，直到最後。

節省法（Clark and Wright Saving）：

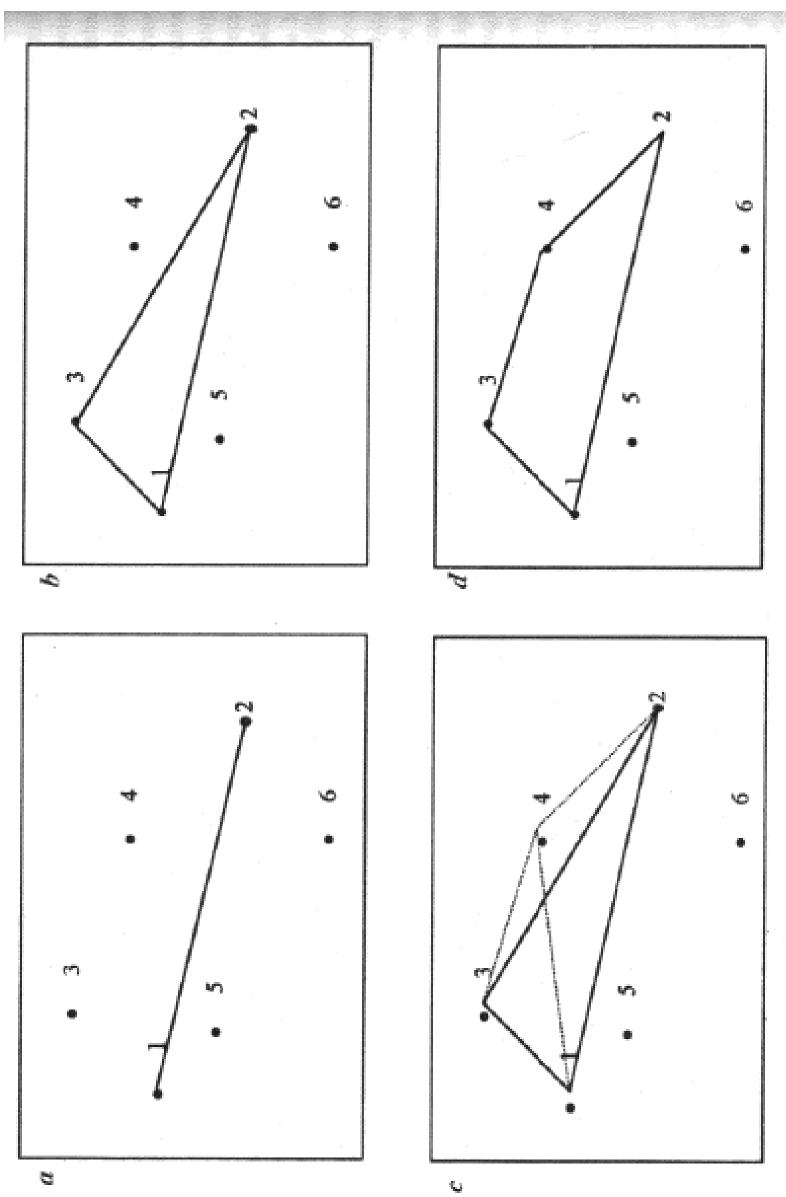
以服務每一個節點為起始解，根據三角不等式兩邊之和大於第三邊之性質，其起始狀況為每服務一個顧客後便回場站，而後計算路線間合併節省量，將節省量以降序排序而依次合併路線，直到最後。





插入法（Insertion procedures）：

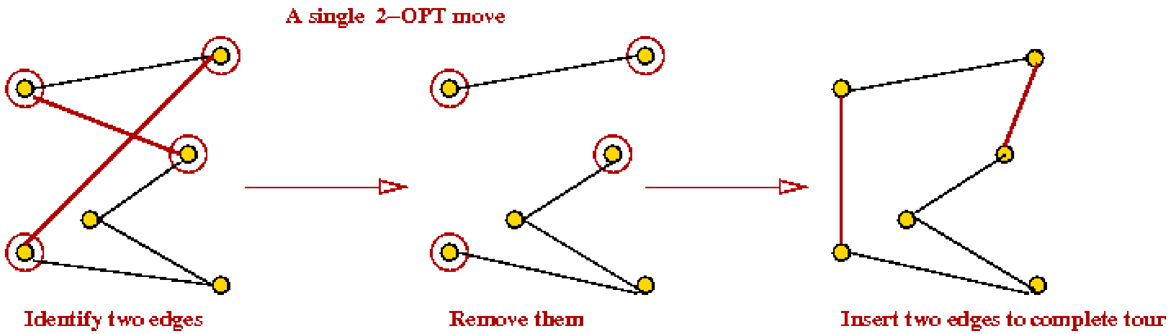
如最近插入法、最省插入法、隨意插入法、最遠插入法、最大角度插入法等。

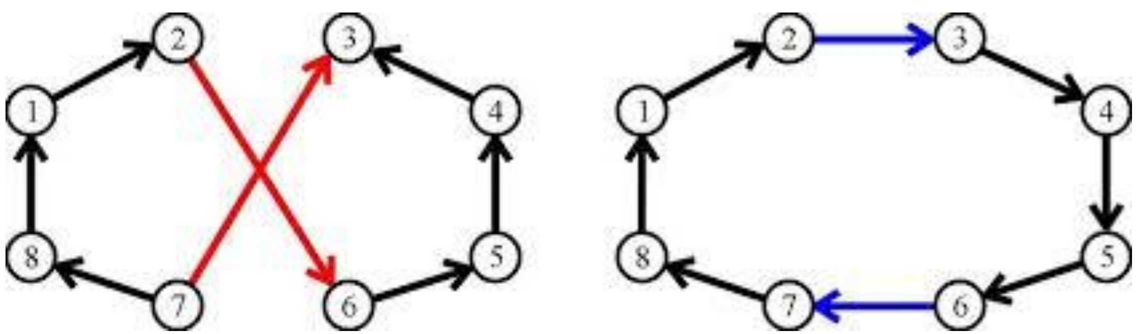


Tour Improvement

- 根據任意一個起使可行解，以交換型啟發式方法（Exchange Approach）改善路線成本，例如2-opt交換法、3-opt交換法、k-opt交換法、Lin-Kernighan交換法等。

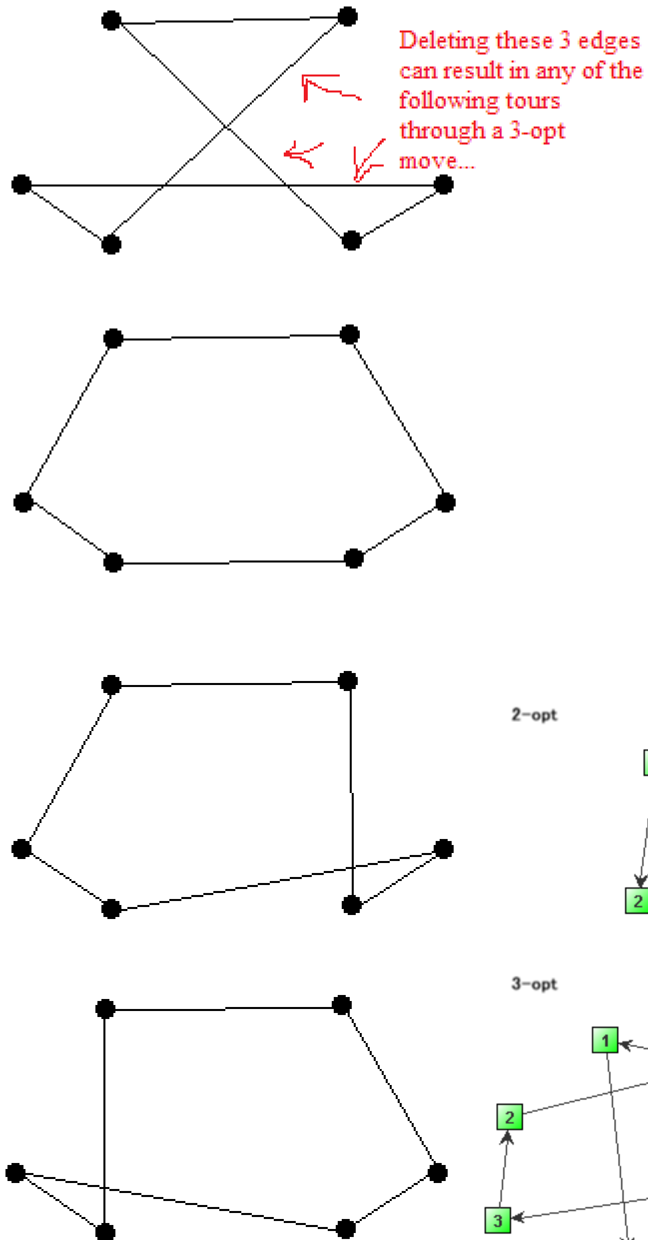
2-OPT

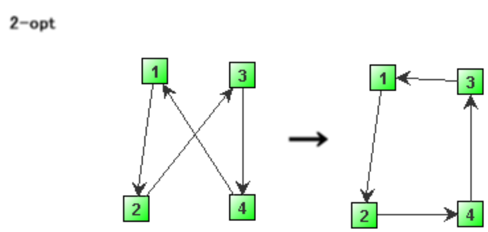


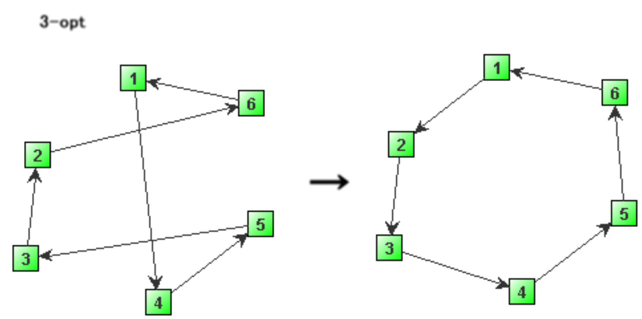


2-opt is a simple local search algorithm first proposed by Croes in 1958 for solving the traveling salesman problem. The main idea behind it is to take a route that crosses over itself and reorder it so that it does not.

3-OPT

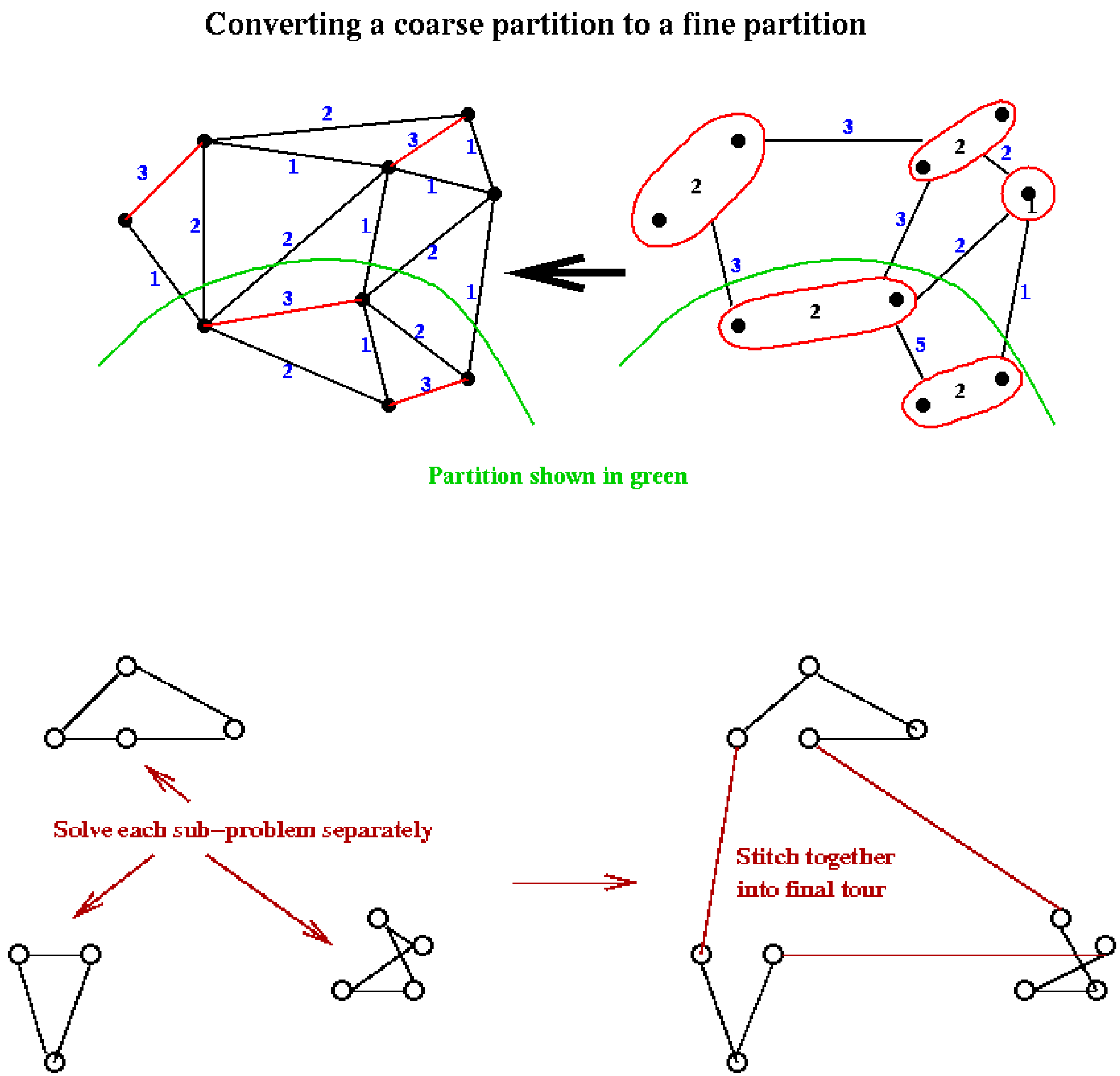






3-opt analysis involves deleting 3 connections (or edges) in a network (or tour), reconnecting the network in all other possible ways, and then evaluating each reconnection method to find the optimum one. This process is then repeated for a different set of 3 connections.

LIN–KERNIGHAN HEURISTIC



Briefly, it involves swapping pairs of sub-tours to make a new tour. It is a generalization of 2-opt and 3-opt. 2-opt and 3-opt work by switching two or three paths to make the tour shorter. Lin–Kernighan is adaptive and at each step decides how many paths between cities need to be switched to find a shorter tour.

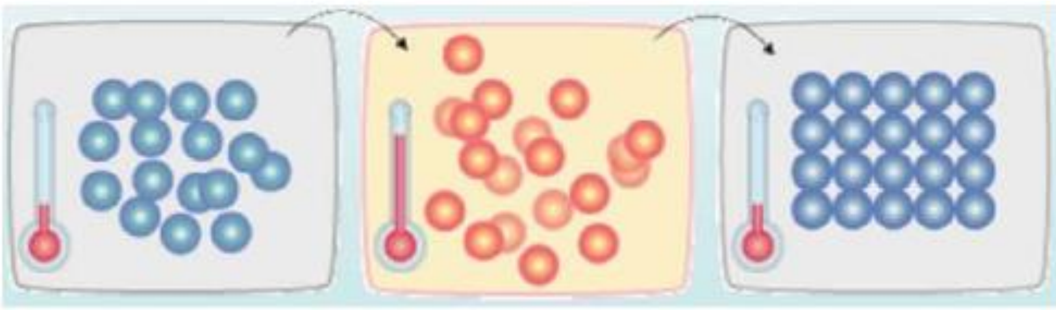
META-HEURISTICS

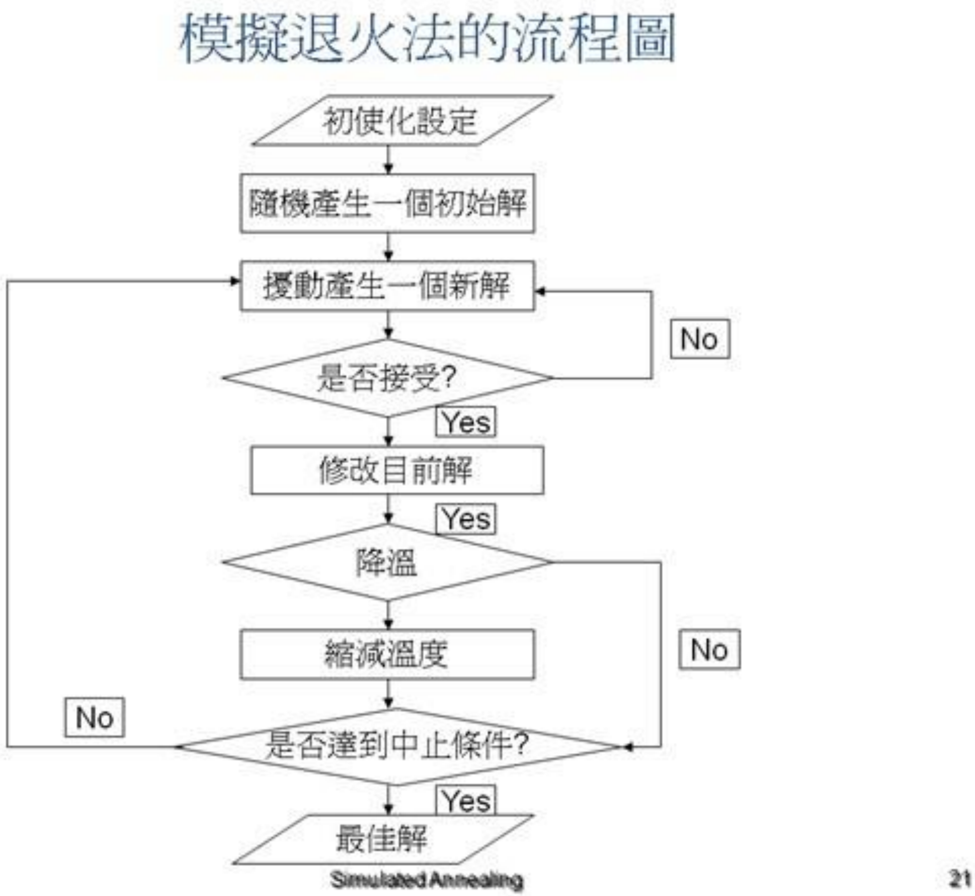
軌跡搜尋法	族群式搜尋法

- (1) 模擬退火法
- (2) 塔布搜尋法
- (3) 可變性鄰近搜尋法

- (1) 螞蟻理論
- (2) 進化演算法
- (3) 粒子群最佳化法

模擬退火法（SIMULATED ANNEALING）





模擬物質經加熱後，逐漸冷卻所產生的結晶現象，亦稱為「退火（Annealing）」。模擬退火的原理也和金屬退火的原理近似：我們將熱力學的理論套用到統計學上，將搜尋空間內每一點想像成空氣內的分子；分子的能量，就是它本身的動能；而搜尋空間內的每一點，也像空氣分子一樣帶有「能量」，以表示該點對命題的合適程度。演算法先以搜尋空間內一個任意點作起始：每一步先選擇一個「鄰居」，然後再計算從現有位置到達「鄰居」的機率。

http://toddwschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny/

The basic elements of simulated annealing (SA) are the following:

1. A finite set S.
2. A real-valued cost function J defined on S. Let $S^* \subset S$ be the set of global minima of the function J, assumed to be a proper subset of S.
3. For each $i \in S$, a set $S(i) \subset S - \{i\}$, called the set of neighbors of i.
4. For every i, a collection of positive coefficients q_{ij}, $j \in S(i)$, such that $\sum_{j \in S(i)} q_{ij} = 1$. It is assumed that $j \in S(i)$ if and only if $i \in S(j)$.
5. A nonincreasing function T: $N \to (0, \infty)$, called the *cooling schedule*. Here N is the set of positive integers, and $T(t)$ is called the *temperature* at time t.
6. An initial "state" $x(0) \in S$.

Acceptance probabilities

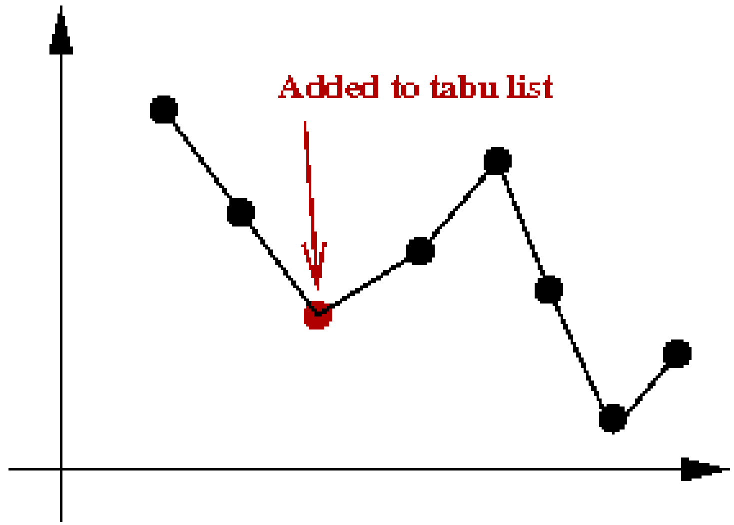
If $J(j) \leq J(i)$, then $x(t+1) = j$.
If $J(j) > J(i)$, then
$x(t+1) = j$
with probability $\exp[-(J(j) - J(i))/T(t)]$
$x(t+1) = i$ otherwise.

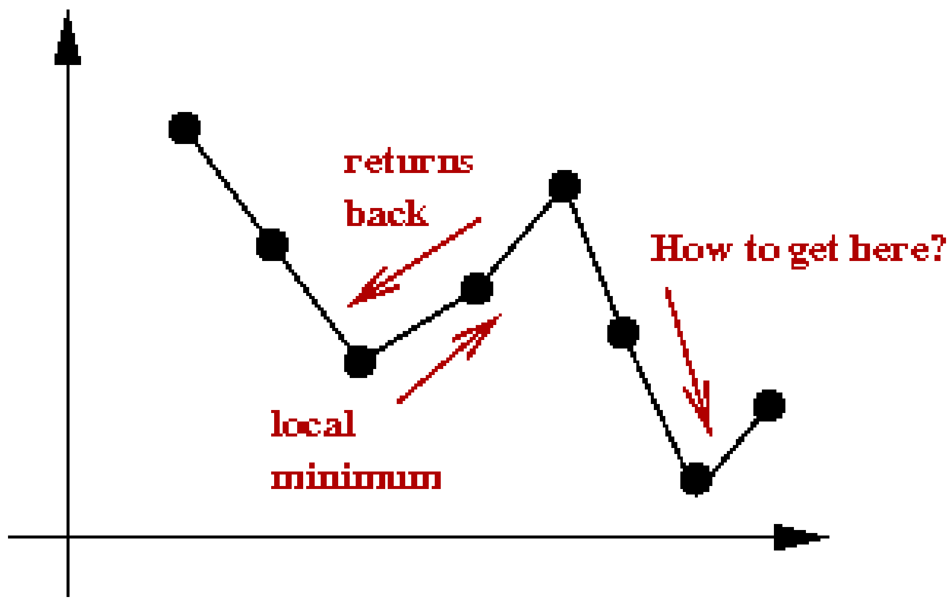
```
cost_previous = infinite
temperature = temperature_start
while temperature > temperature_end:
    cost_new = cost_function(...)
    difference = cost_new - cost_previous
    if difference < 0 or  e^(-difference/temperature) > random(0,1):
        cost_previous = cost_new
    temperature = temperature * cooling_factor
```

SA Example

Applied onto the TSP

塔布搜尋法（TABU SEARCH）

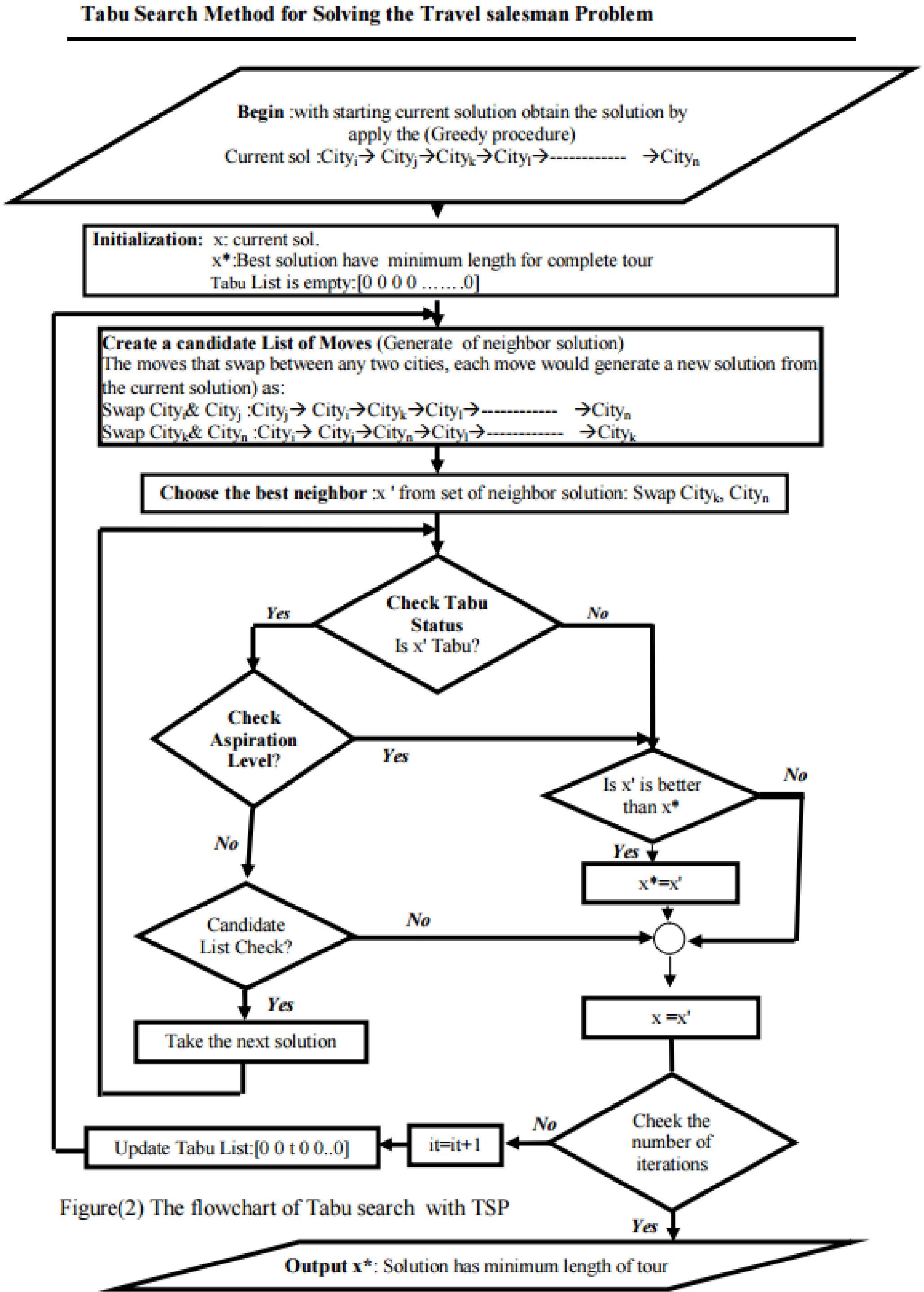




對局部領域搜索的一種擴展，是一種全局逐步尋優演算法，是對人類智力過程的一種模擬。TS演算法通過引入一個靈活的存儲結構和相應的禁忌準則來避免迂迴搜索，並通過藐視準則來赦免一些被禁忌的優良狀態，進而保證多樣化的有效探索以最終實現全局優化。

TS Example

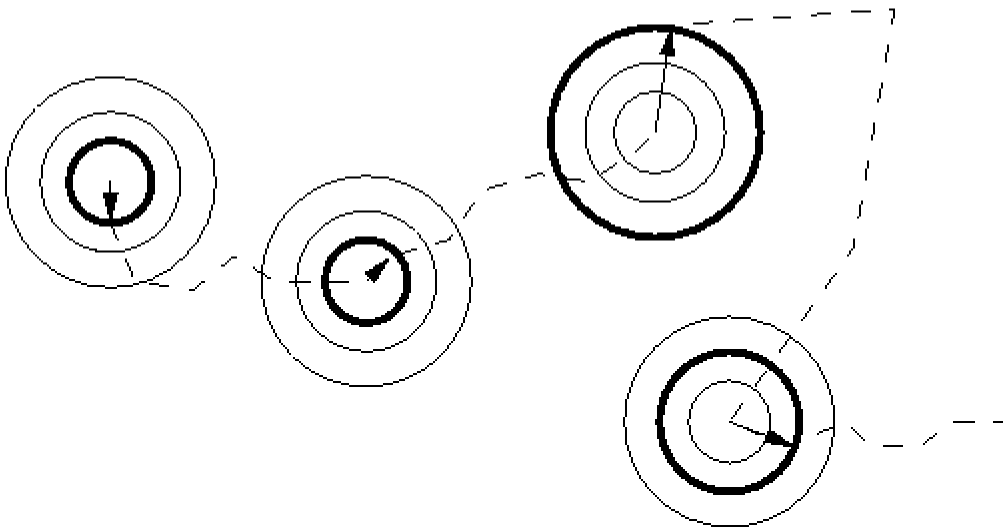
2025/9/29



Figure(2) The flowchart of Tabu search with TSP

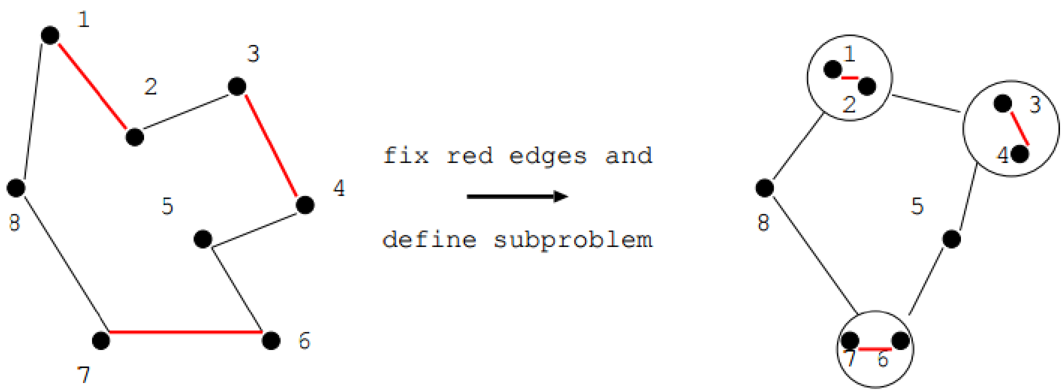
*Alkallak, I.N. and Sha'ban, R.Z. (2008) Tabu Search method for solving the Traveling Salesman Problem. Raf. J. of Comp. & Math's. 5(2), 141-153

可變性鄰近搜尋法(Variable Neighborhood Search)



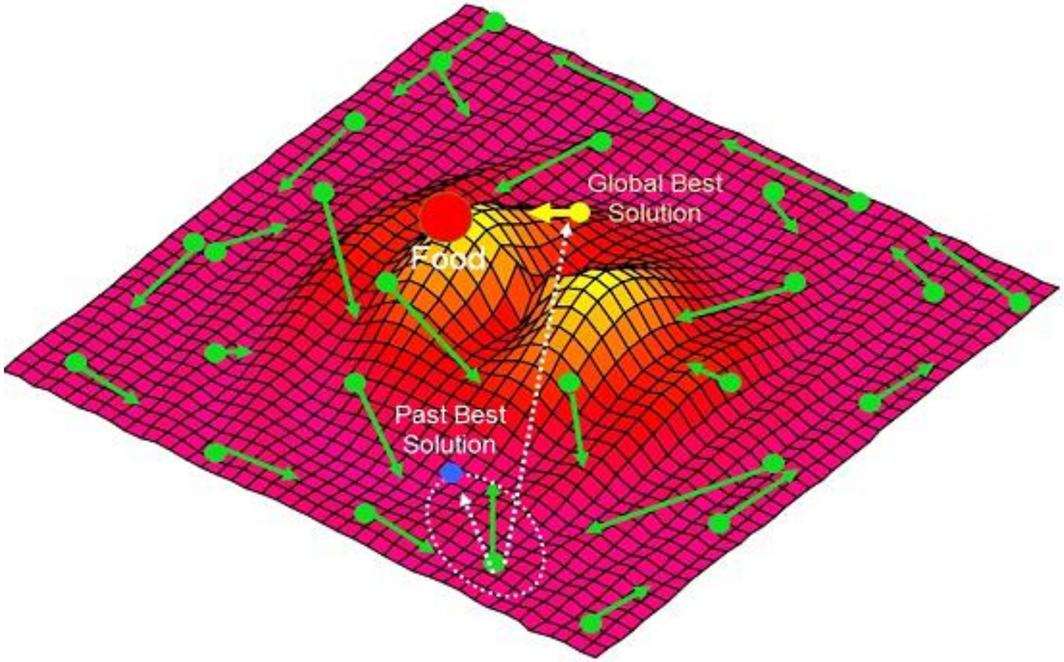
central idea

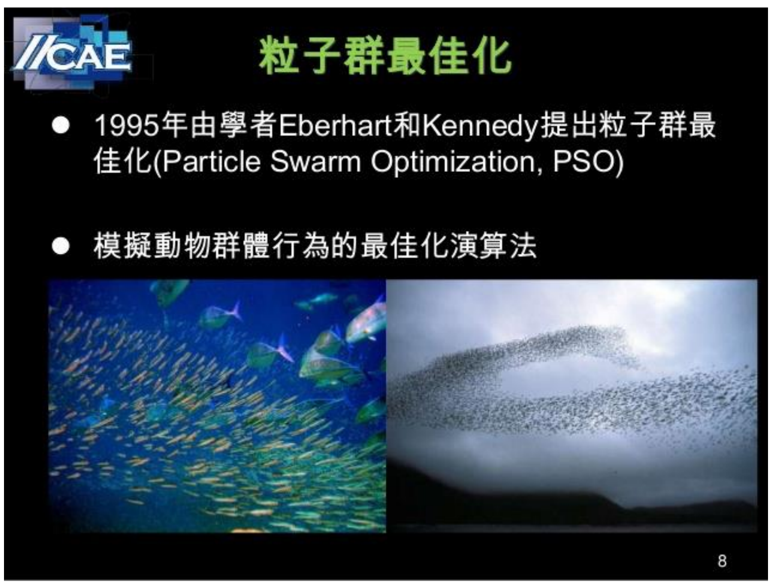
- generate subproblems by keeping all but k solution components fixed
- apply a local search only to the k "free" components



It explores distant neighborhoods of the current incumbent solution, and moves from there to a new one if and only if an improvement was made. The local search method is applied repeatedly to get from solutions in the neighborhood to local optima.

粒子群最佳化法(Particle Swarm Optimization)





粒子群最佳化法(Particle Swarm Optimization)中的每個體稱稱為「粒子」(particle)，是代表解空間(solution space)中的一個可能的解。每個粒子的移動除了出自本身的慣性(inertia)外，還參考個體本身最佳經驗移而產生認知學習(cognitive learning)的遷移，也參考群體整體最佳經驗做社會學習(social learning)，迭代演化前進，最後收斂而得到最佳解。

PSO Example

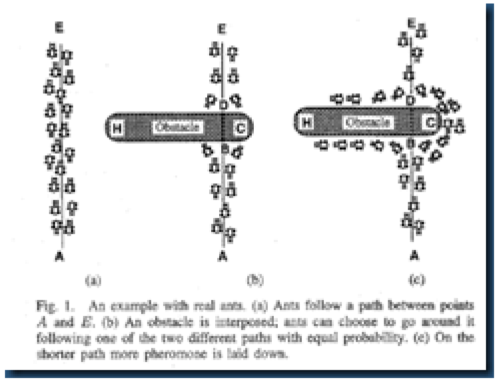
$$\boldsymbol{v}_i^{(t+1)} = a \cdot \boldsymbol{v}_i^{(t)} + r_{\text{loc}} \cdot b_{\text{loc}} \cdot (\boldsymbol{p}_i - \boldsymbol{x}_i^{(t)}) + r_{\text{glob}} \cdot b_{\text{glob}} \cdot (\boldsymbol{p}_{\text{glob}} - \boldsymbol{x}_i^{(t)}) \tag{1}$$

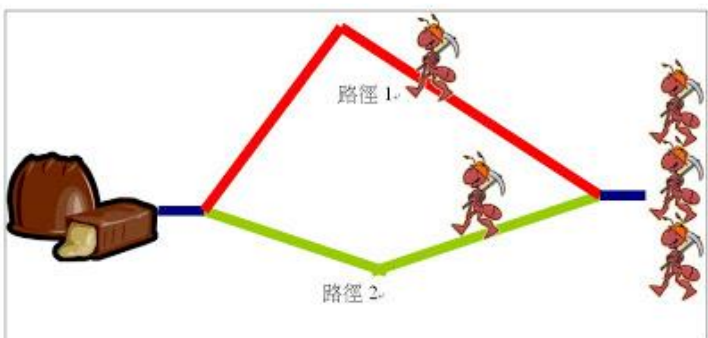
$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}. \tag{2}$$

The parameters $a, b_{\text{loc}}, b_{\text{glob}} \in \mathbb{R}$ are constant weights which can be selected by the user. The *inertia* a adjusts the relative importance of the inertia of the particles, and the so-called *acceleration coefficients* b_{loc} and b_{glob} determine the influence of the local and the global attractor, resp. In every iteration, r_{loc} and r_{glob} are drawn uniformly at random from $[0, 1]$. Particles exchange information about the search space exclusively via the global attractor $\boldsymbol{p}_{\text{glob}}$. $\boldsymbol{p}_i - \boldsymbol{x}_i^{(t)}$ is the *local attraction* exerted by the local attractor on particle i, and $\boldsymbol{p}_{\text{glob}} - \boldsymbol{x}_i^{(t)}$ is the *global attraction* exerted by the global attractor.

蟻群演算法（ANT COLONY OPTIMIZATION, ACO）

一種用來在圖中尋找優化路徑的機率型演算法，其靈感來源於螞蟻在尋找食物過程中發現路徑的行為。蟻群演算法是一種模擬進化演算法，初步的研究表明該演算法具有許多優良的性質.針對PID控制器參數優化設計問題，將蟻群演算法設計的結果與遺傳演算法設計的結果進行了比較，數值模擬結果表明，蟻群演算法具有一種新的模擬�i化優化方法的有效性和應用價值。





$$s = \begin{cases} \max\limits_{j \in \Omega_i}\{Att_{i,j}\} & \text{if} \quad q \leqslant q_0 \\ p_{i,j} & \text{otherwise} \end{cases}$$

where q is a random number generated from a
parameter q_0, then the next arc chosen is the arc
to the state transition probability $p_{i,j}$ given by:

$$p_{i,j} = \begin{cases} \dfrac{Att_{i,j}}{\sum_{k \in \Omega_i} Att_{i,k}} & \forall j \in \Omega_i \\ 0 & \forall j \notin \Omega_i \end{cases}$$

and the attraction of arc (i,j) for an ant is

$$Att_{i,j} = (\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}$$

3.1.2. The local pheromone-updating rule

Every time an artificial "ant" finishes cons
be represented as follows:

$$\tau_{i,j} = (1-\rho)\cdot\tau_{i,j} + \rho\cdot\tau_0$$

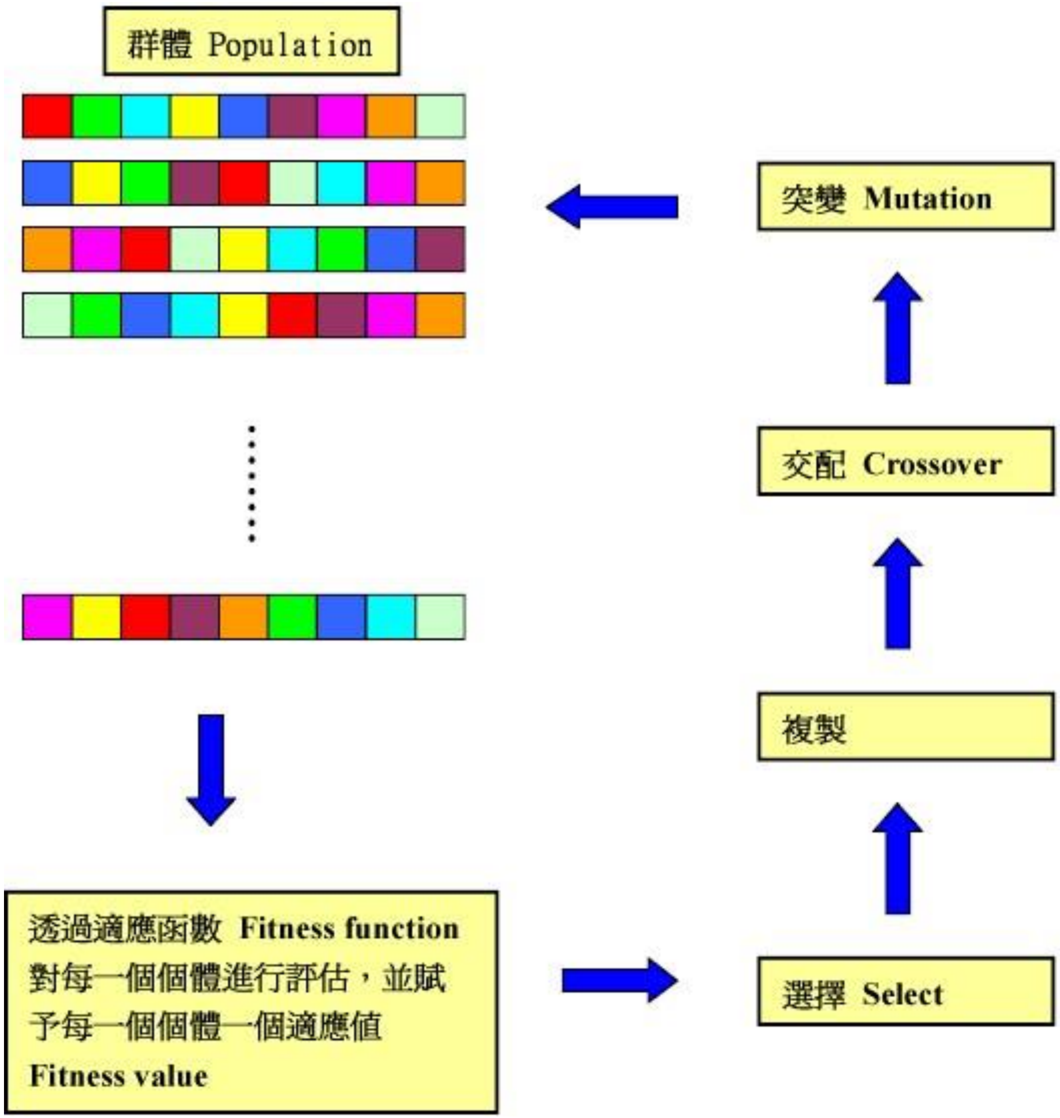
where $\tau_0 = 1/(N\cdot J_{\psi^{\mathrm{ini}}})$ is the amount of phe

3.1.3. The global pheromone-updating rule

The ACO will update the pheromone alloc
number of iterations. In addition, the pheron
omone-updating rule resets the ant colony's
update rule is as follows:

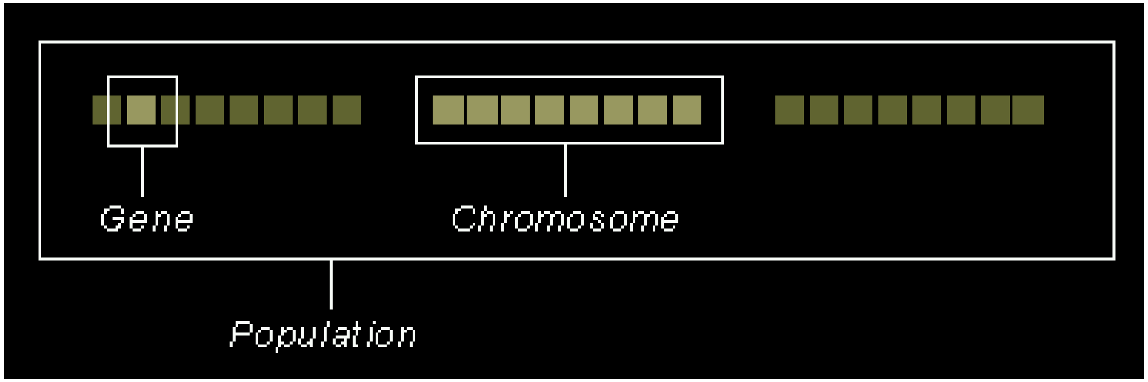
$$\tau_{i,j} = (1-\rho)\cdot\tau_{i,j} + \rho/J_{\psi^{gl}} \quad (i,j)\in\psi^{gl}$$

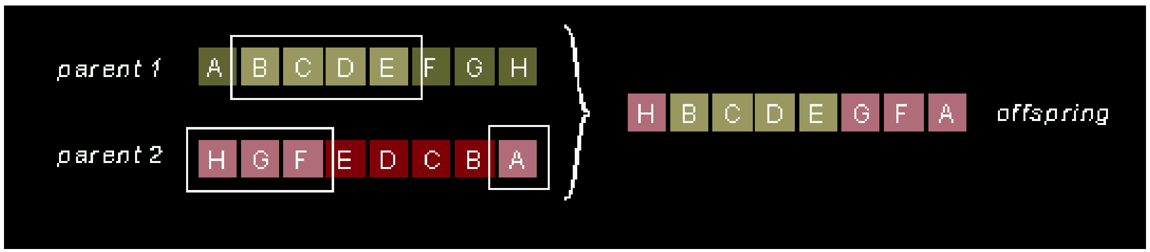
基因演算法(GA)

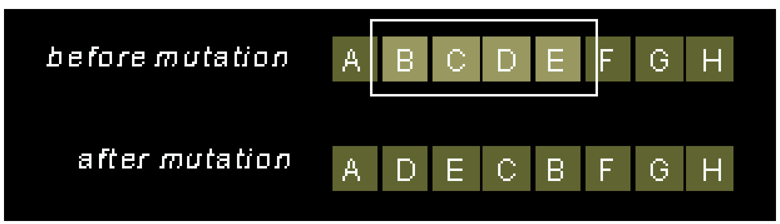


遺傳算法是模仿自然界生物進化機制發展起來的隨機全局搜索和優化方法，它借鑑了達爾文的進化論和孟德爾的遺傳學說。其本質是一種高效、並行、全局搜索的方法，它能在搜索過程中自動獲取和積累有關搜索空間的知識，並自適應的控制搜索過程以求得最優解。遺傳算法操作使用適者�穿s的原則，在潛在的解決方案種群中逐次產生一個近似最優解的方案，在遺傳算法的每一代中，根據個體在問題域中的適應度值和從自然遺傳學中借鑑來的再造方法進行個體選擇，產生一個新的近似解。這個過程導致種群中個體的進化，得到的新個體比原來個體更能適應環境，就像自然界中的改造一樣。

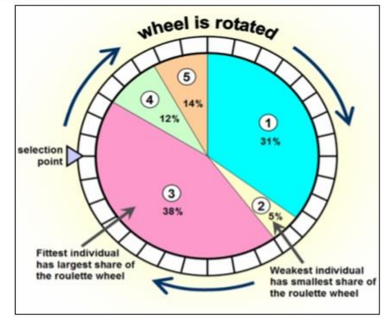
GA



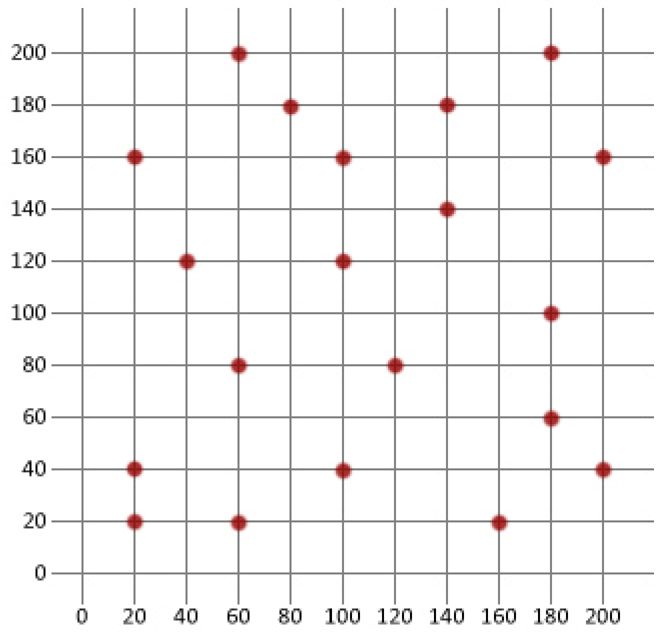




Roulette Wheel Selection



GA Example



Parents

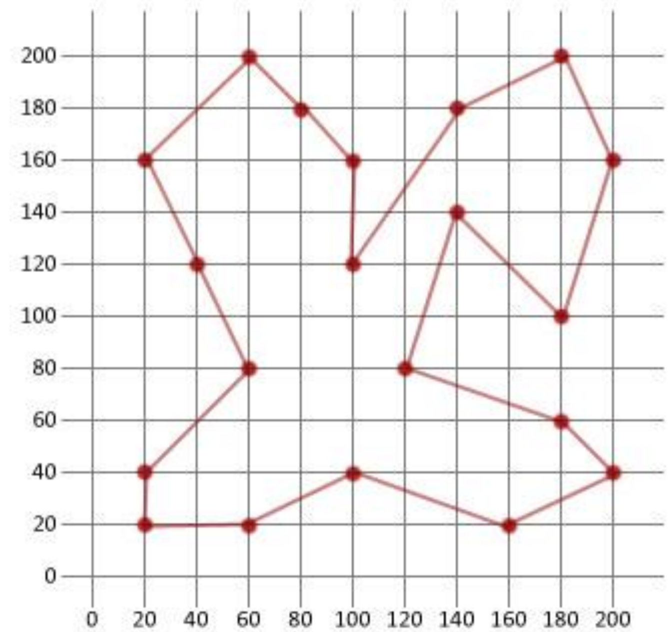
1	2	3	4	5	6	7	8	9

9	8	7	6	5	4	3	2	1

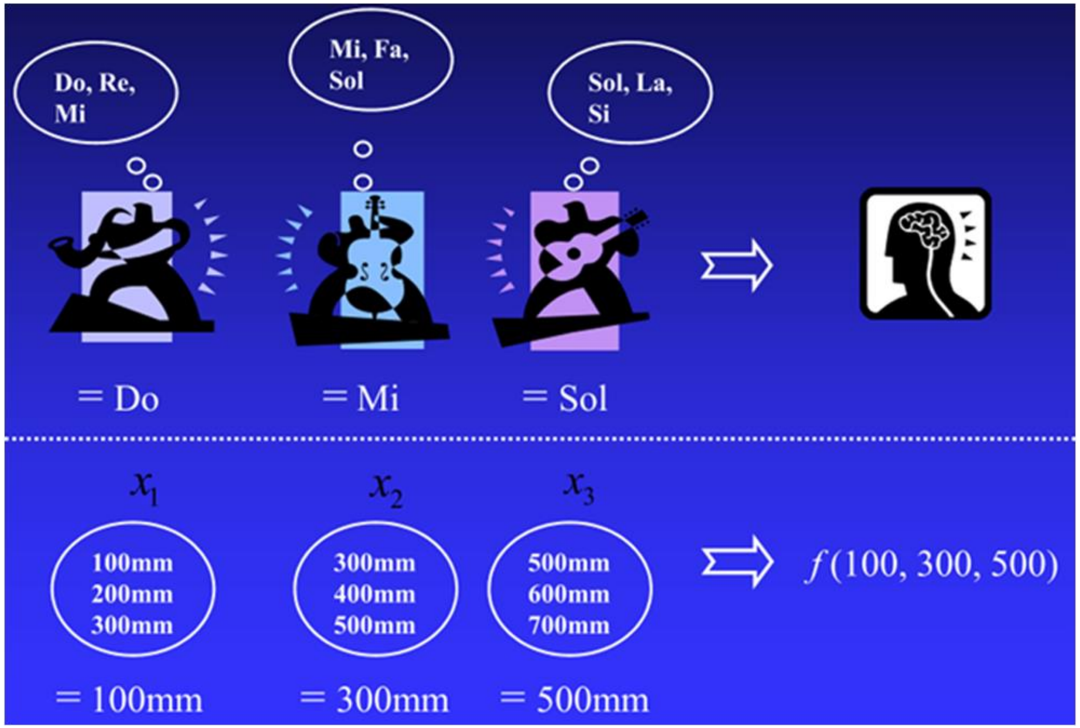
Offspring

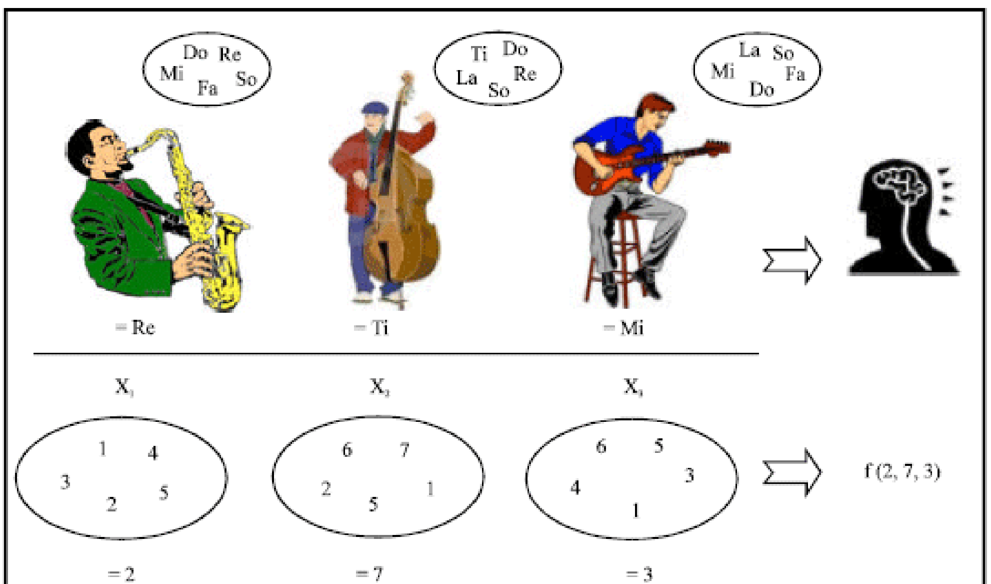
					6	7	8	

9	5	4	3	2	6	7	8	1



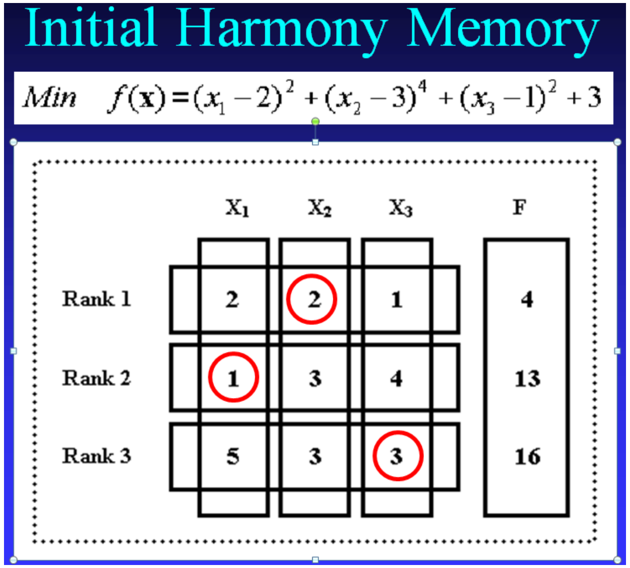
HARMONY SEARCH

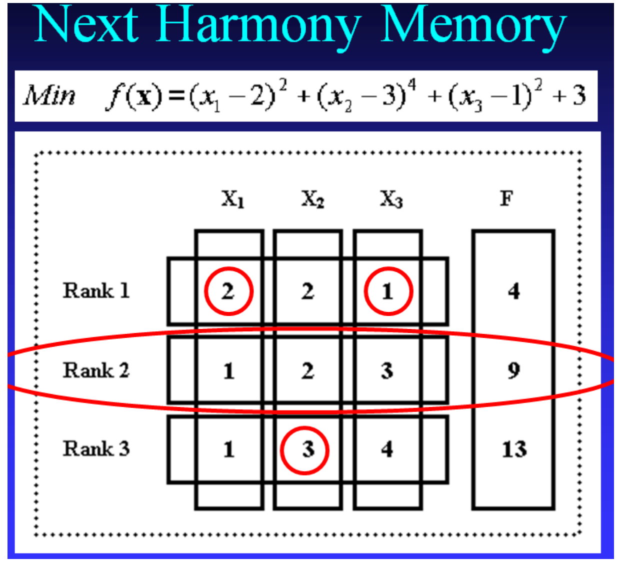


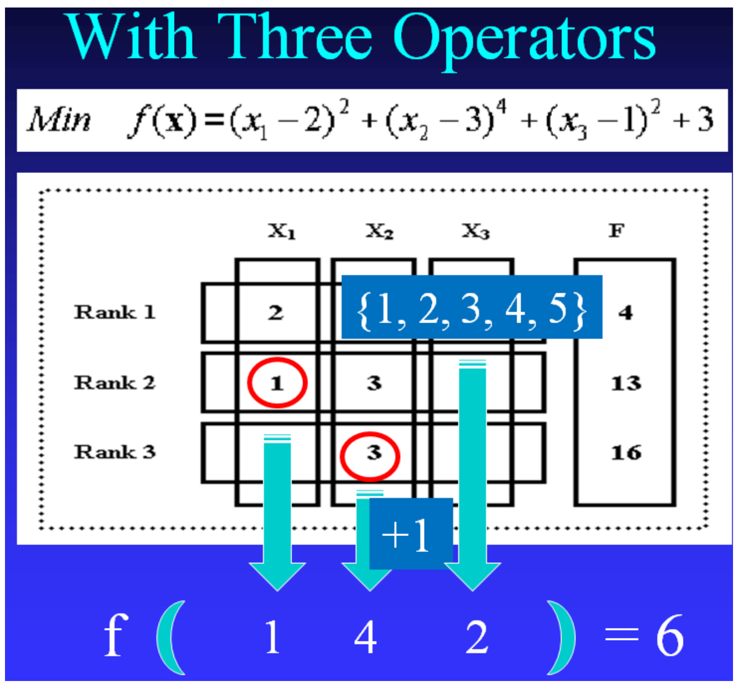


In the HS algorithm, each musician (= decision variable) plays (= generates) a note (= a value) for finding a best harmony (= global optimum) all together.

2025/9/29


Initial Harmony Memory
Min $f(\mathbf{x}) = (x_1-2)^2 + (x_2-3)^4 + (x_3-1)^2 + 3$
X1 X2 X3 F
Rank 1 2 2 1 4
Rank 2 1 3 4 13
Rank 3 5 3 3 16


Next Harmony Memory
Min $f(\mathbf{x}) = (x_1-2)^2 + (x_2-3)^4 + (x_3-1)^2 + 3$
X1 X2 X3 F
Rank 1 2 2 1 4
Rank 2 1 2 3 9
Rank 3 1 3 4 13


With Three Operators
Min $f(\mathbf{x}) = (x_1-2)^2 + (x_2-3)^4 + (x_3-1)^2 + 3$
X1 X2 X3 F
Rank 1 2 4
Rank 2 1 3 13
Rank 3 3 16
{1, 2, 3, 4, 5}
+1
f (1 4 2) = 6

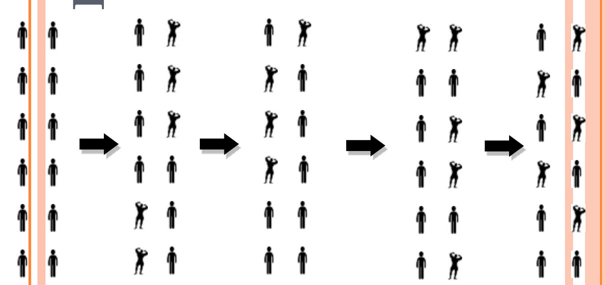
Harmony-Genetic Algorithm

(4) *Reproduction and production:* Two methods are used to generate the population for a new generation.

4.1. The first approach is to reproduce the population by conventional crossover methods. The single-point crossover is used in this research, and the offspring with the largest fitness value is retained for the next generation each time a crossover is complete. The mutation scheme remains affected on those chromosomes. If mutation is triggered, a randomly generated gene will replace another gene at a randomly selected position.

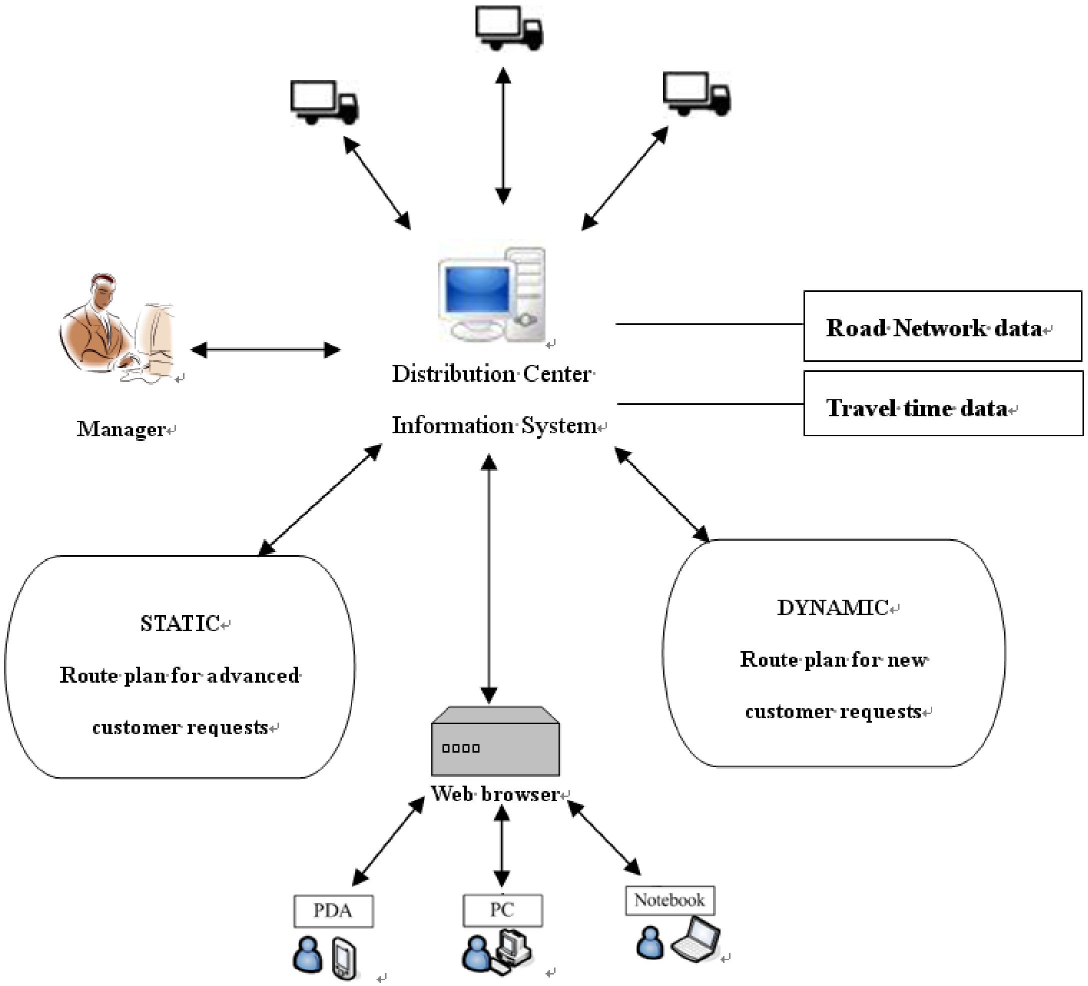
4.2. The other approach is to produce a Frankenstein. First, one noumenon is selected, and the genes are subsequently decided individually. The pathological changes may be applied on the gene under a specific probability, $\bar{\rho}$.

sponding gene in the noumenon. It is believed that the chromosome with the highest fitness value has superior gene traits. This step is called the "transplant." The transplant operation is executed only if $q < \bar{q}$ where q is randomly generated and $\bar{q}$ is given. The parameter $\bar{q}$ is named the "transplant ratio." When $q \geqslant \bar{q}$, the corresponding gene in the noumenon remains the same. A consequence that may occur after the transplantation is referred to as "pathological changes." If pathological changes occur, the gene will mutate to another form and a new gene will be generated to replace the old gene. A specific parameter called the "pathological changes ratio," symbolized as $\bar{\rho}$, was designed to control the occurrence probability of pathological changes. If $\rho < \bar{\rho}$, the pathological changes occur, where ρ is a random number.



Applications

A model for solving the dynamic vehicle dispatching problem with customer uncertainty and time dependent link travel time


Manager
Distribution Center
Information System
Road Network data
Travel time data
STATIC
Route plan for advanced
customer requests
DYNAMIC
Route plan for new
customer requests
Web browser
PDA
PC
Notebook

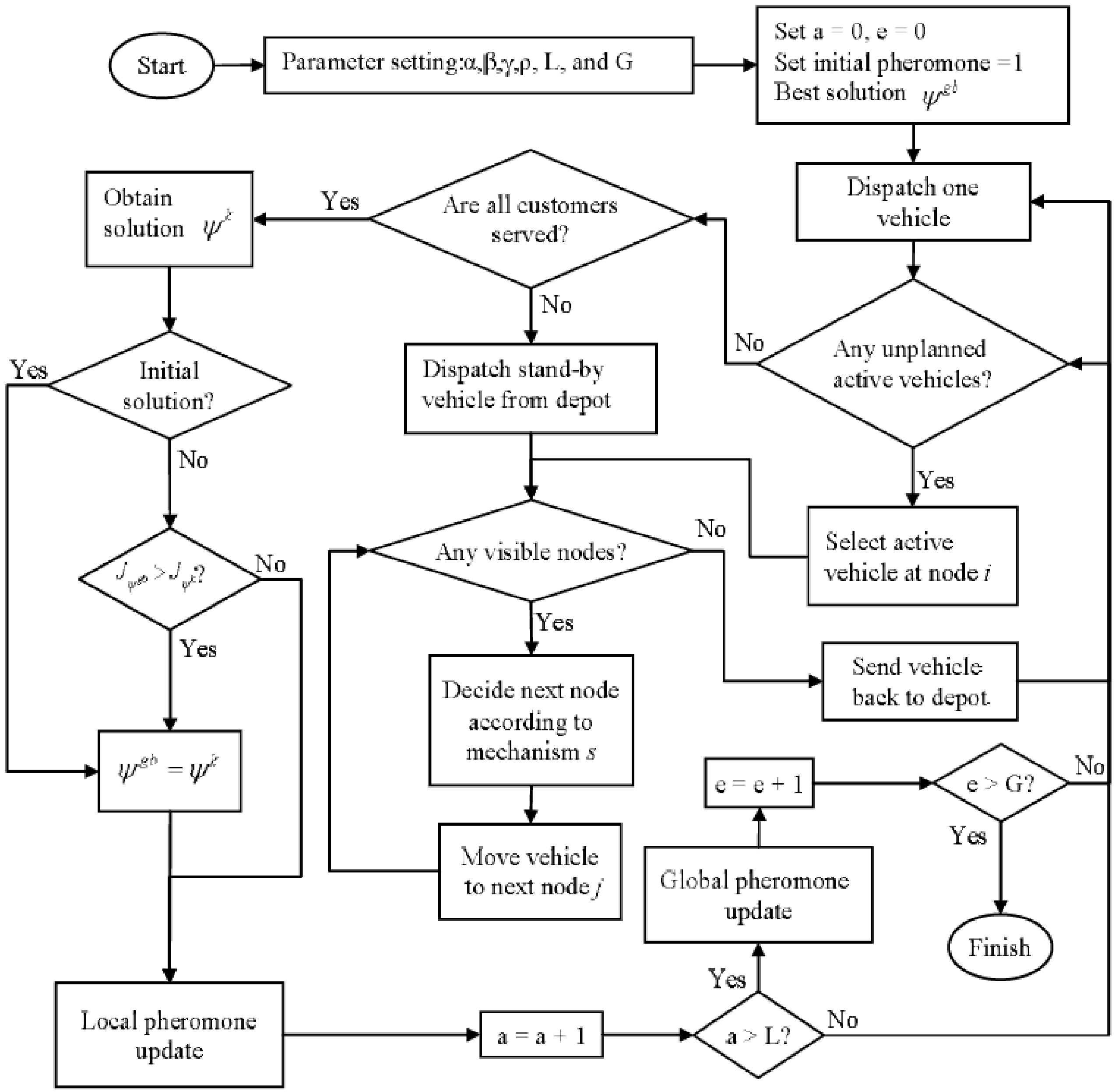
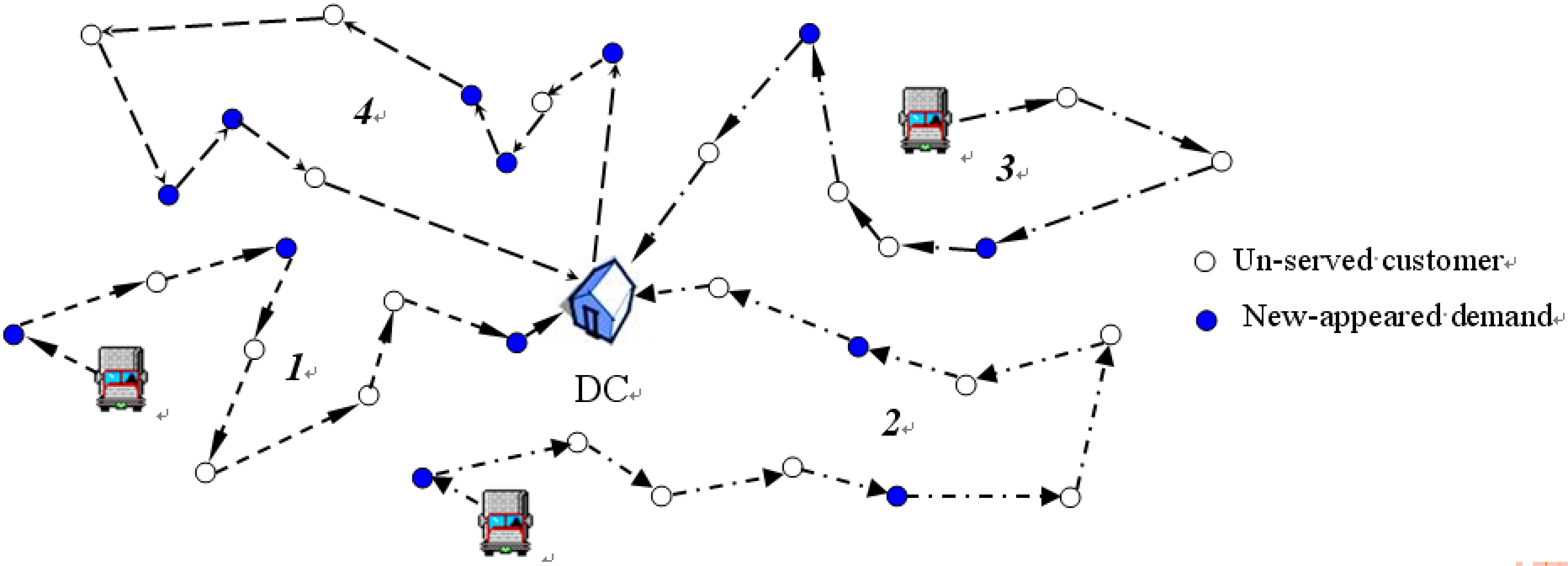


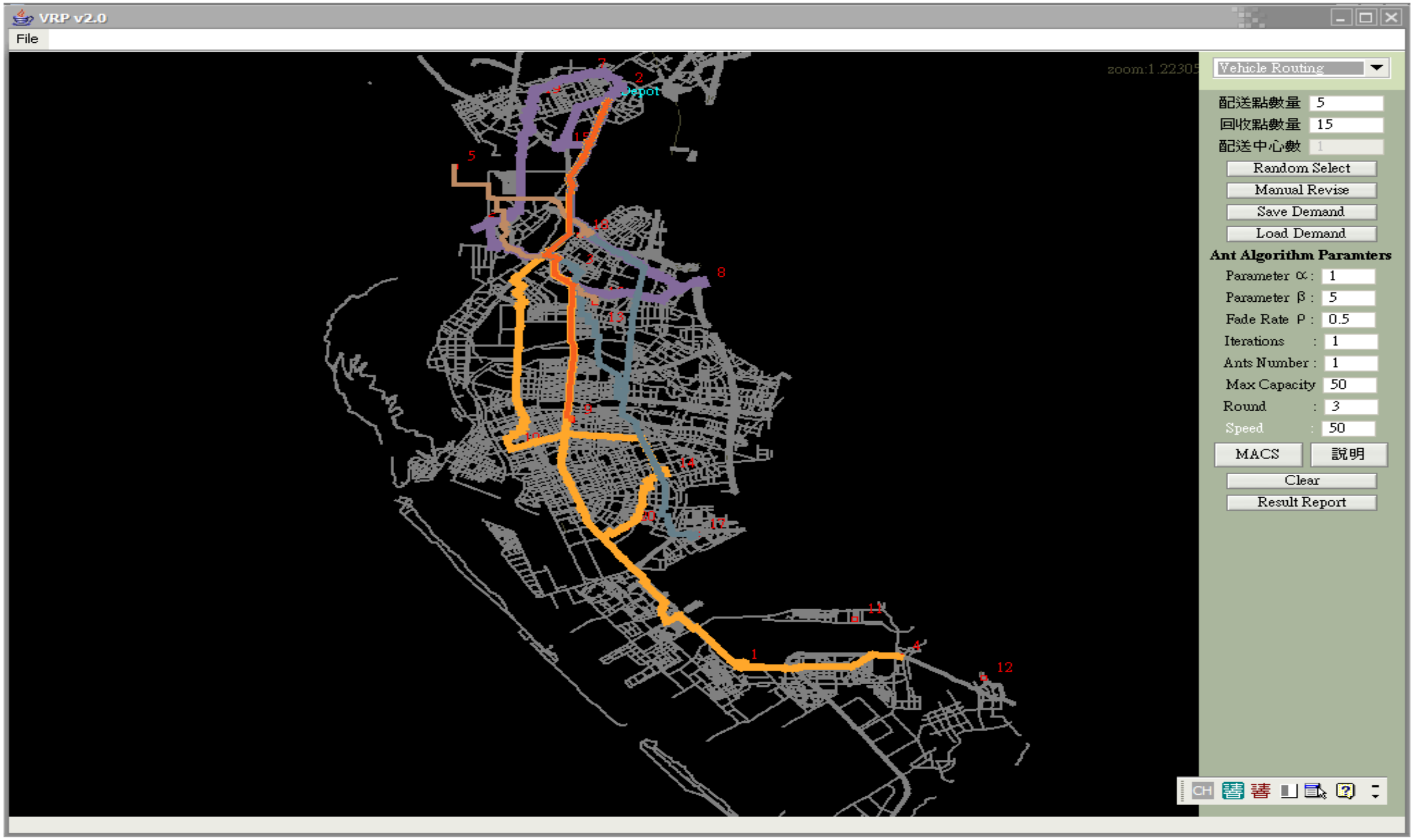
Figure 2 Flowchart of the ACO rerouting process

2025/9/29



202

2025/9/29


VRP v2.0
File
Vehicle Routing
配送點數量 5
回收點數量 15
配送中心數 1
Random Select
Manual Revise
Save Demand
Load Demand
Ant Algorithm Paramters
Parameter α : 1
Parameter β : 5
Fade Rate ρ : 0.5
Iterations : 1
Ants Number : 1
Max Capacity 50
Round : 3
Speed : 50
MACS
說明
Clear
Result Report
Depot

2025/9/29

A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty

stockout cost

$$SOC = \int_{x=1}^{x=\infty} \sum_{i \in V_2} \sum_{c=1}^{C} S_c \times P_i^c(x)\, dx$$

recourse

$$\sum_{\substack{i \in V_1;\\ i=1}}^{k} 2d_{i,0} \times \theta_{i,m}^{c}$$

Z = planned travel cost + recourse cost + stockout cost

$$z = \sum_{m=1}^{M} (D^m \times T) + \sum_{m=1}^{M} \sum_{i=1}^{k} (2d_{i,0} \times \theta_{i,m}^{c} \times T) + \int_{x=1}^{\infty} \sum_{i \in V_2} \sum_{c=1}^{C} S_c \times P_i^c(x)\, dx$$

$$\text{s.t.} \sum_{i\in V_1}\sum_{c=1}^{C} E[\xi_i^c] \times y_i^m \leqslant Q \tag{1}$$

$$\sum_{j\in V_1}\sum_{m=1}^{M} x_{i,j}^m = 1 \quad \forall i \in V_1, i\neq 0, j\neq 0, i\neq j \tag{2}$$

$$\sum_{i\in V_1}\sum_{m=1}^{M} x_{i,j}^m = 1 \quad \forall j \in V_1, i\neq 0, j\neq 0, i\neq j \tag{3}$$

$$\sum_{\substack{j\in V_2\\ j\neq i}}\sum_{m=1}^{M} x_{i,j}^m = 0 \quad \forall i\neq j \tag{4}$$

$$\sum_{\substack{i\in V_2\\ i\neq j}}\sum_{m=1}^{M} x_{i,j}^m = 0 \quad \forall j\neq i \tag{5}$$

$$\sum_{i\in V_1}\sum_{m=1}^{M} x_{i,0}^m = \sum_{j\in V_1}\sum_{m=1}^{M} x_{0,j}^m = m \tag{6}$$

$$x_{i,j}^m \leqslant y_i^m \quad \forall i \in V_1 \quad \forall m \in M \tag{7}$$

$$D^m = \sum_i \sum_j d_{i,j} \cdot x_{i,j}^m \quad \forall i,j \in \{0\} \cup V_1, i\neq j \tag{8}$$

$$\sum_{i=1}^{N} y_i^m w_i^m + D^m / s_{i,j}^m \leqslant W \quad \forall m \in M \tag{9}$$

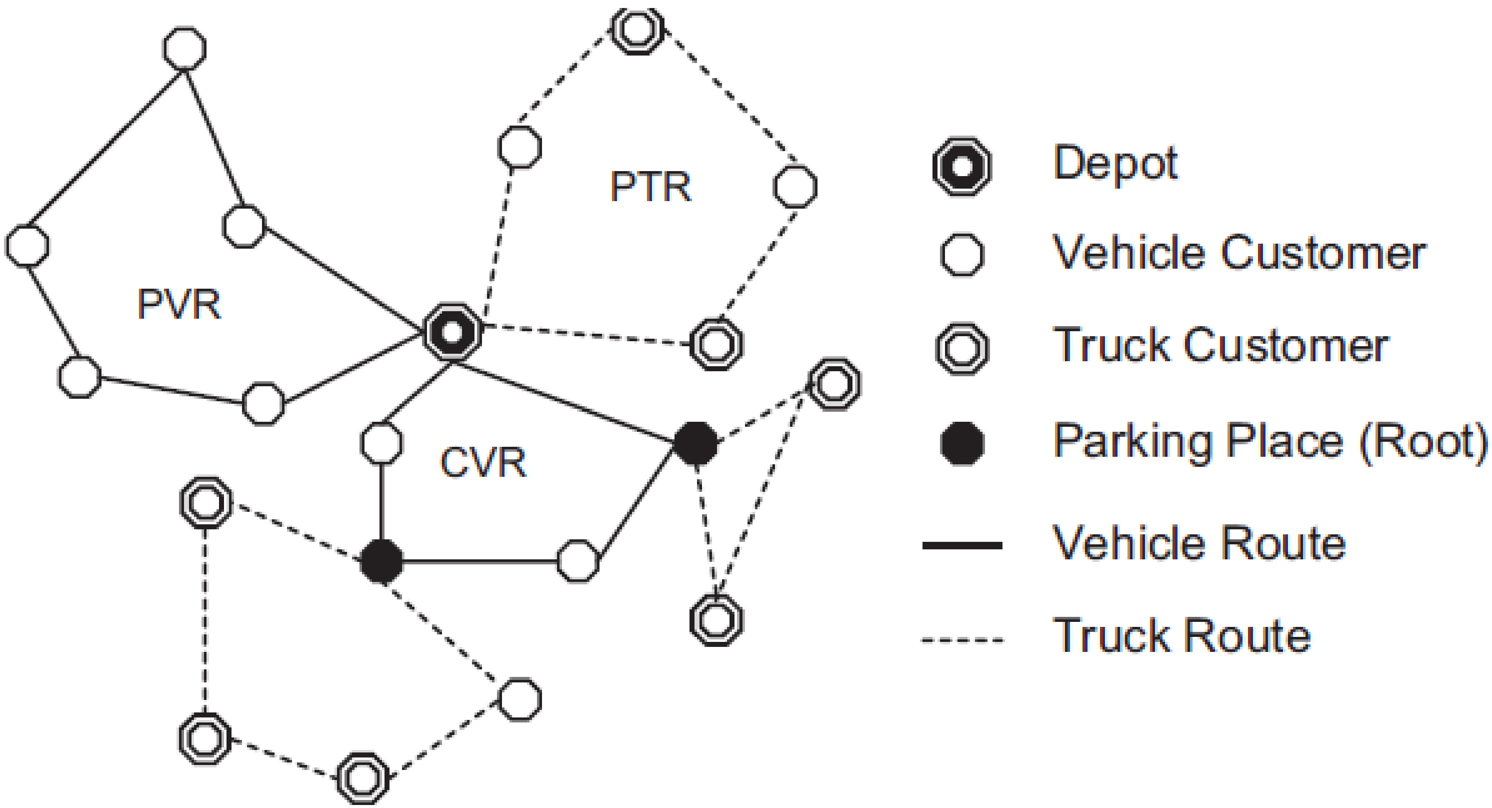
$$u_i^m - u_j^m + Q \cdot x_{i,j}^m \leqslant Q - \sum_{c=1}^{C} E[\xi_j^c] \quad \forall m \in M \tag{10}$$

$$\sum_{c=1}^{C} E[\xi_i^c] \leqslant u_i^m \leqslant Q \quad \forall i \in V_1 \quad \forall m \in M \tag{11}$$

$$x_{i,j}^m = 0 \text{ or } 1, \quad y_i^m = 0 \text{ or } 1, \quad \forall i \in V_1 \quad \forall m \in M$$

Truck with Trailer

Truck with Trailer



Fleet size determination for a truckload distribution center

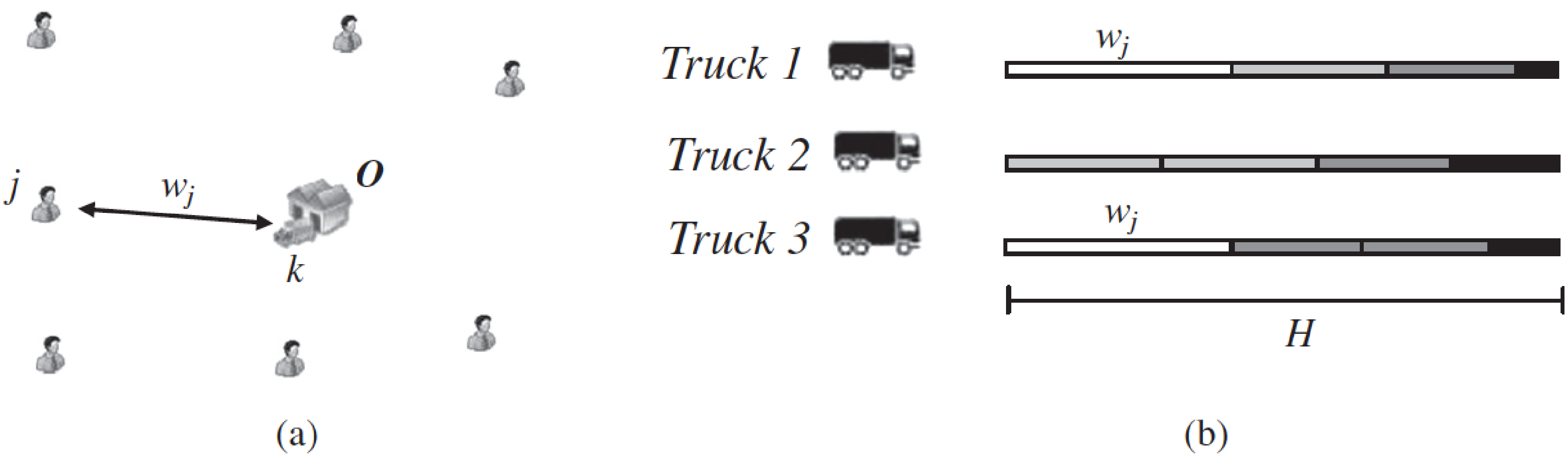


Figure 1. Illustration of the truck working time.

25	4	3	1	10	14	12	23	24	6	19	7	17	18	2	16	21	22	8	5	9	20	13	15	11

Truck 1 Truck 2 Truck 3 Truck 4 Truck 5

Figure 2. Demonstration of a chromosome and the number of trucks needed.

A

20	**14**	**9**	**4**	**23**	**22**	**17**	**7**	**1**	**13**	**25**	**3**	**12**	**10**	**21**	**18**	**2**	**15**	**16**	**19**	**8**	**11**	**5**	**24**	**6**
0	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	1

A’

B

16	8	4	1	9	3	12	15	14	10	7	19	22	5	23	2	21	25	24	17	18	20	13	11	6
1	0	0	1	0	1	1	0	1	0	1	0	0	0	1	1	0	0	1	0	1	1	1	1	0

B’

C

20	**14**	**23**	**7**	**1**	**13**	**3**	**12**	**18**	**2**	**16**	**11**	**24**	8	4	9	15	10	19	22	5	21	25	17	6
16	1	3	12	14	7	23	2	24	18	20	13	11	**9**	**4**	**22**	**17**	**25**	**10**	**21**	**15**	**19**	**8**	**5**	**6**

C’

Figure 3. An example of crossover.

Multi-treatment capacitated arc routing of construction machinery in Taiwan's smooth road project

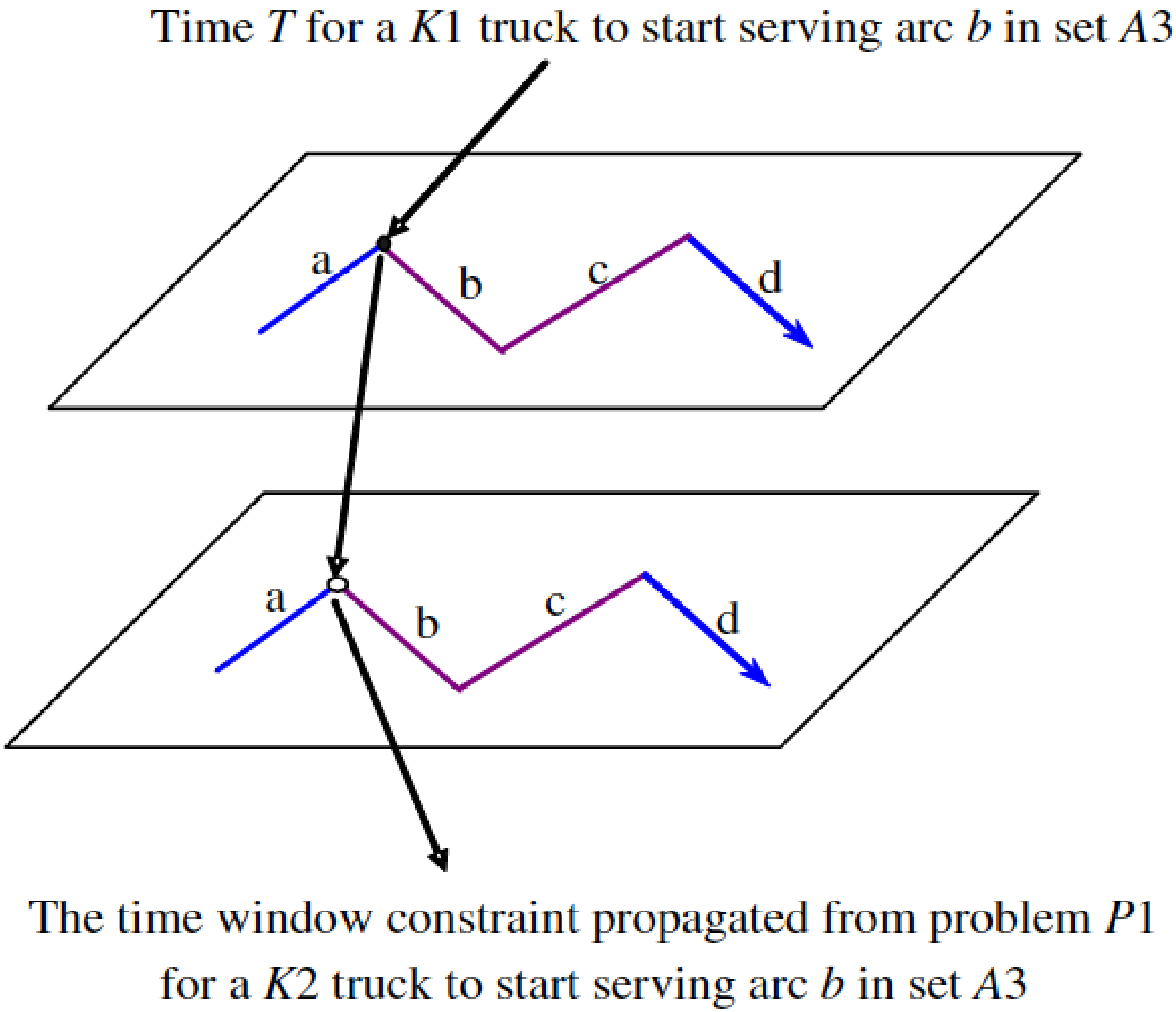


Fig. 1. The formation of time window constraints for $A3$ arcs.

$$\underset{x_{ij}^{k};\ x_{ij}^{\kappa};\ y_{ij}^{k};\ y_{ij}^{\kappa};\ t_{i}^{k};\ t_{i}^{\kappa}}{\text{minimize}} \quad \begin{cases} |K1| + |K2| \\ \sum_{i,j\in V}\left(\sum_{k\in K1} c_{ij}^{k}\cdot x_{ij}^{k} + \sum_{\kappa\in K2} c_{ij}^{\kappa}\cdot x_{ij}^{\kappa}\right) \end{cases}$$

s.t.

$$\sum_{(i,j)\in A1\cup A3} q_{ij}^{k} y_{ij}^{k} \le Q^{k} \qquad \forall k\in K1 \qquad (1)$$

$$\sum_{(i,j)\in A2\cup A3} q_{ij}^{\kappa} y_{ij}^{\kappa} \le Q^{\kappa} \qquad \forall \kappa\in K2 \qquad (2)$$

$$\sum_{k\in K1} y_{ij}^{k} = 1 \qquad \forall (i,j)\in A1\cup A3 \qquad (3)$$

$$\sum_{k\in K2} y_{ij}^{\kappa} = 1 \qquad \forall (i,j)\in A2\cup A3 \qquad (4)$$

$$y_{ij}^{k}\left(t_{i}^{k} + \frac{D_{ij}}{s^{k}} - t_{j}^{k}\right) \le 0 \qquad \forall (i,j)\in A1\cup A3,\quad \forall k\in K1 \qquad (5)$$

$$y_{ij}^{\kappa}\left(t_{i}^{\kappa} + \frac{D_{ij}}{s^{\kappa}} - t_{j}^{\kappa}\right) \le 0 \qquad \forall (i,j)\in A2\cup A3,\quad \forall \kappa\in K2 \qquad (6)$$

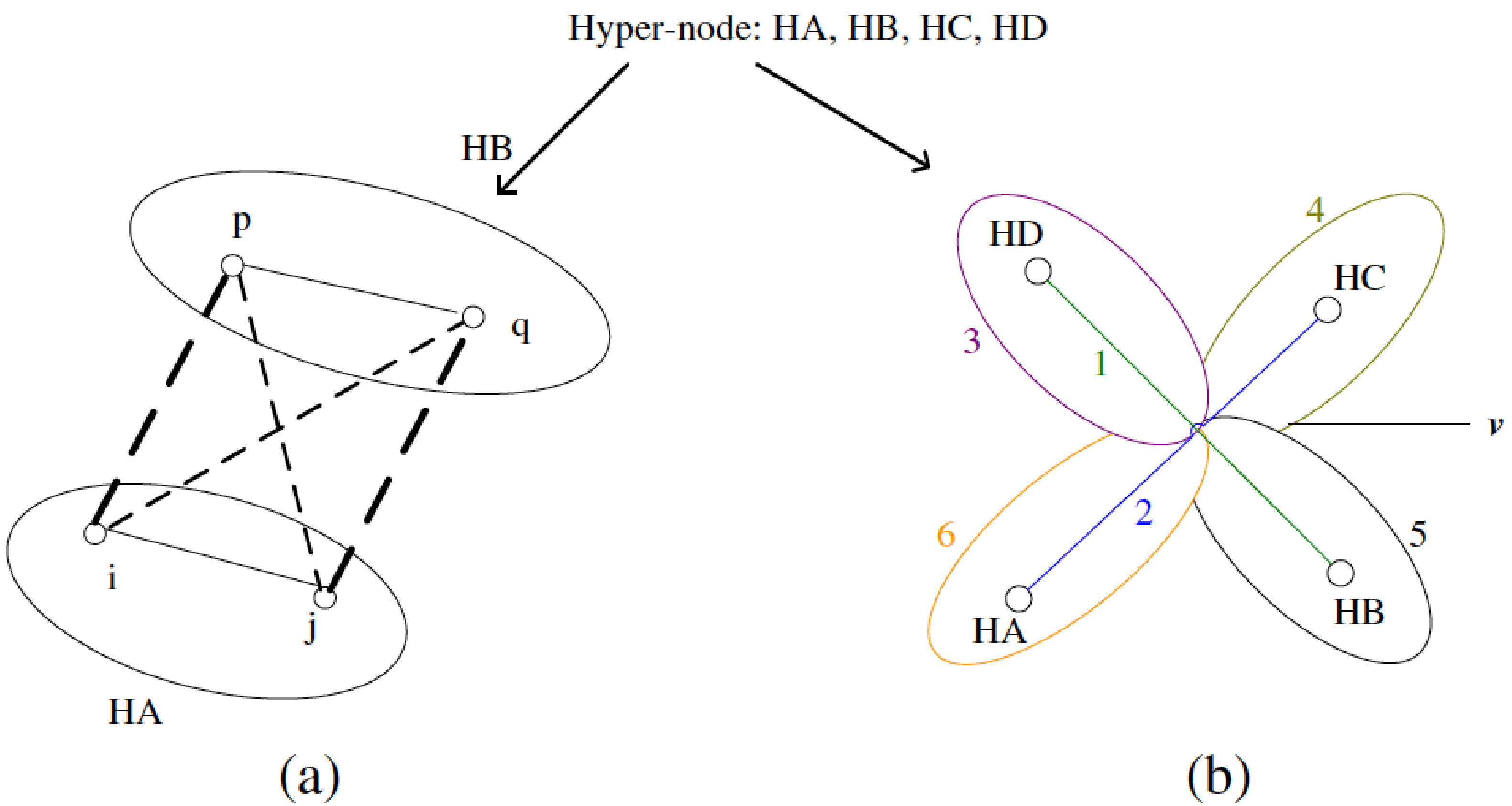


Fig. 2. Illustration of two hyper-node types: (a) separated and b) adjacent.


r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
Depot

Using Ant Colony Optimization to solve Periodic Arc Routing Problem with *Refill Points*

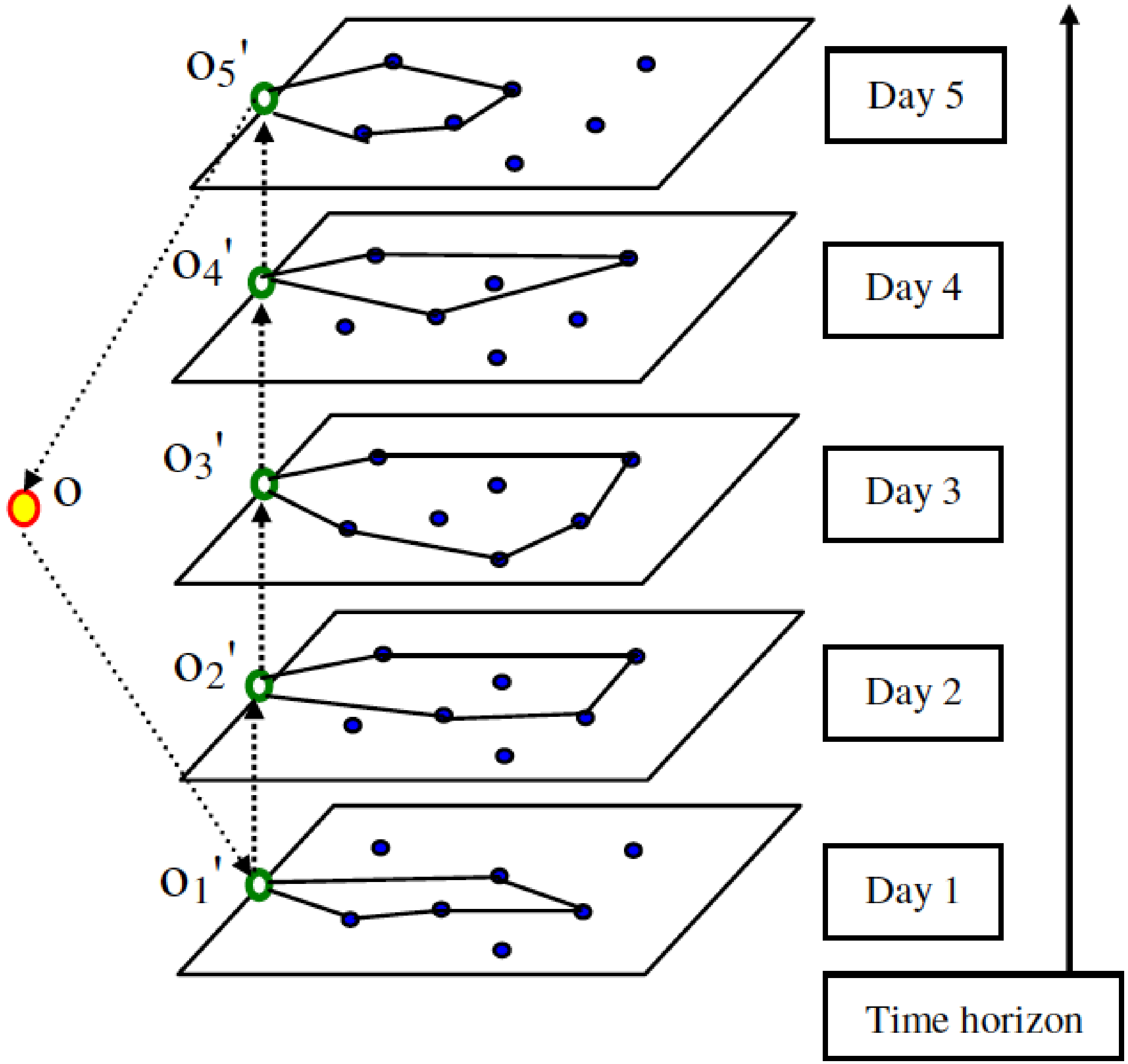
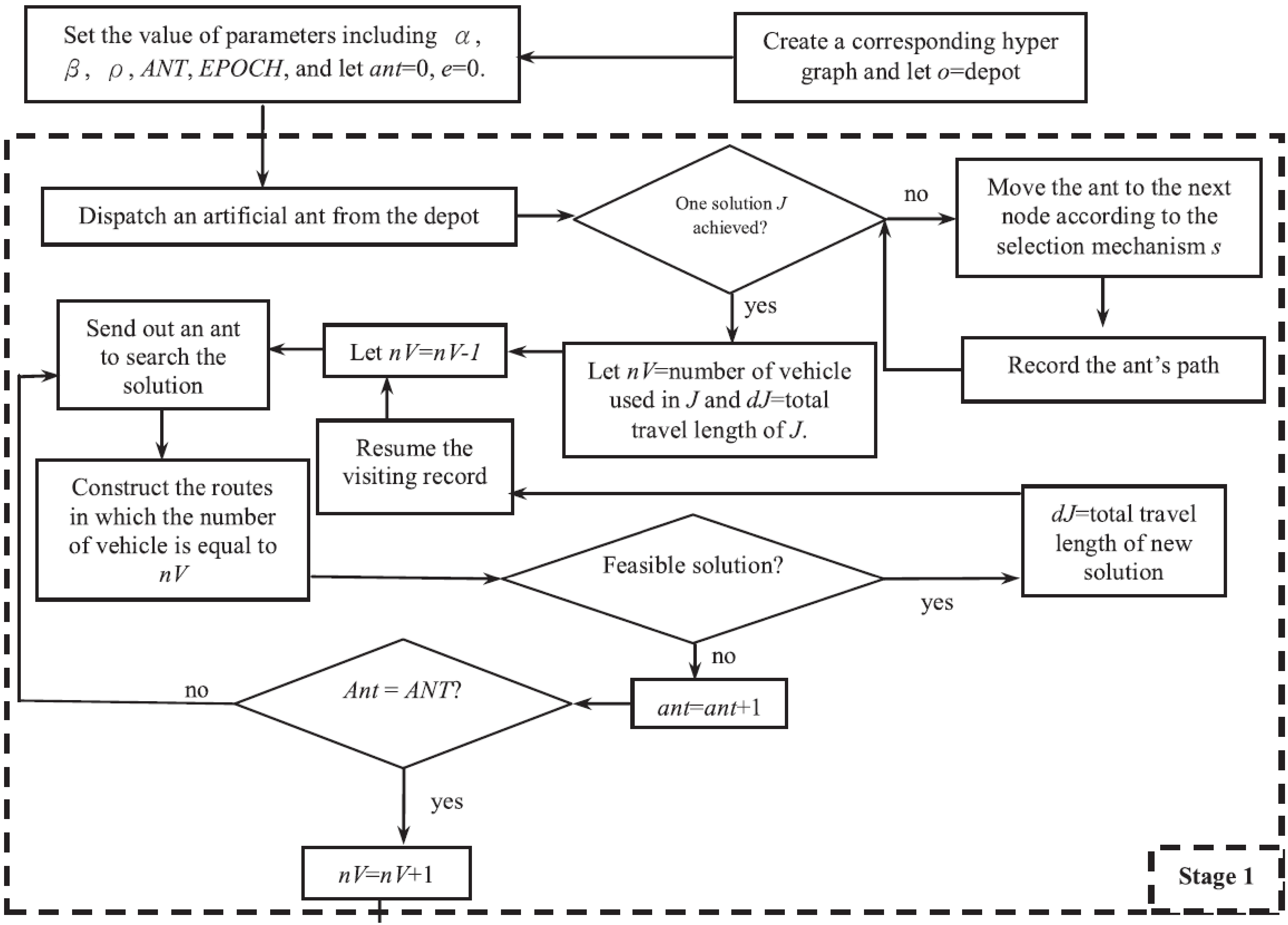
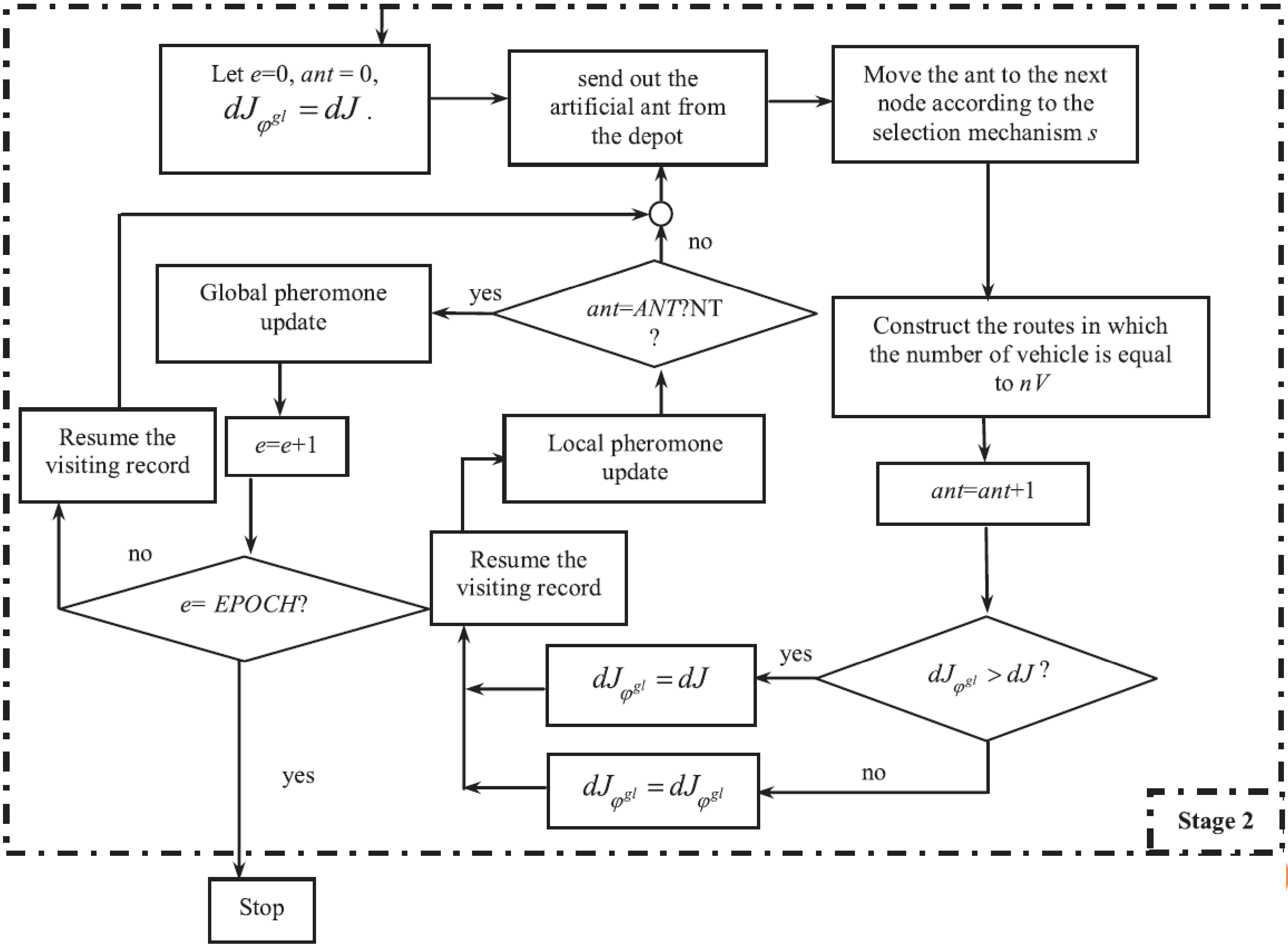
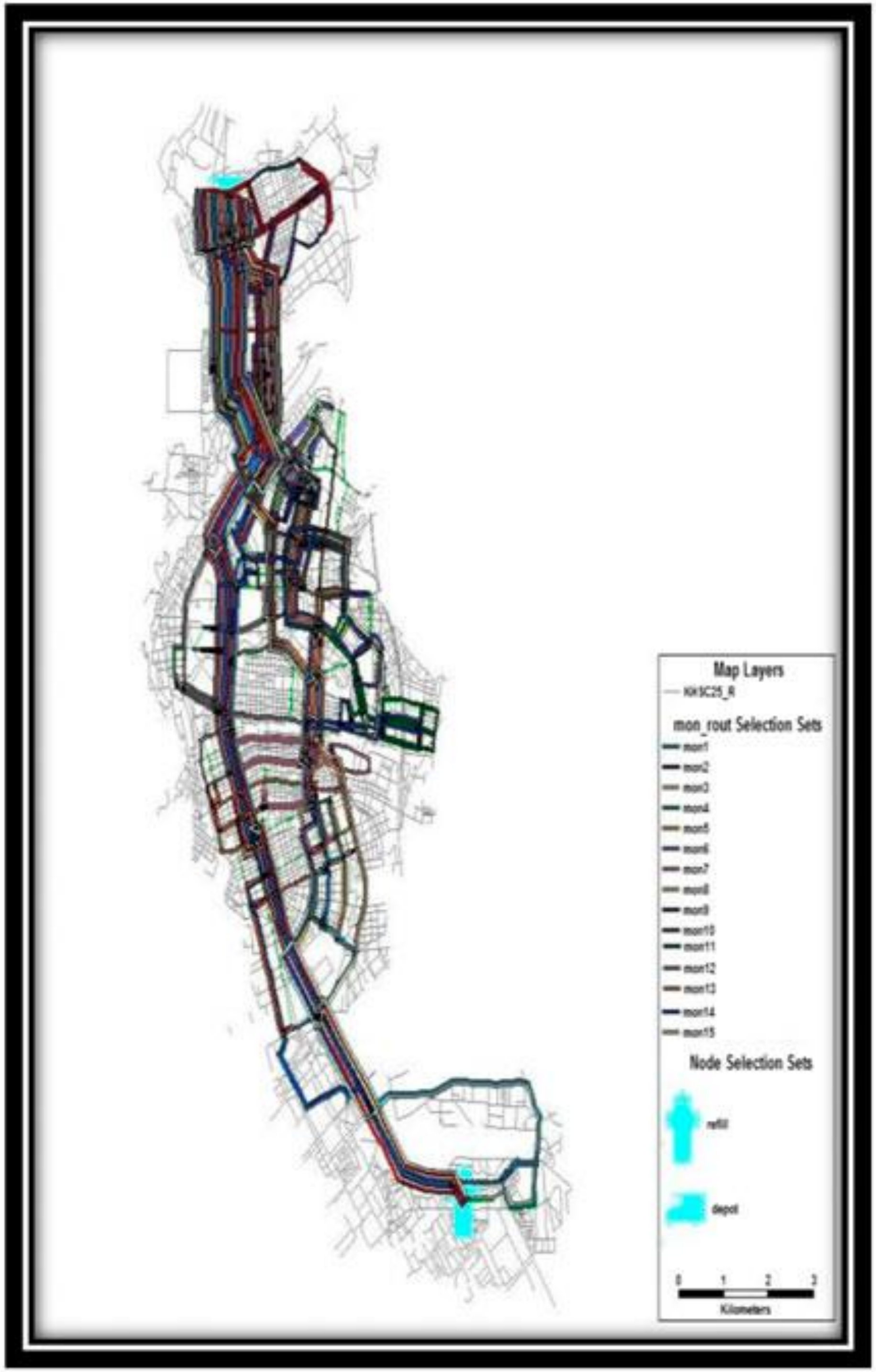


Figure 1. An illustration of the PARPRP network with a weekly route.


Set the value of parameters including α, β, ρ, ANT, EPOCH, and let ant=0, e=0.
Create a corresponding hyper graph and let o=depot
Dispatch an artificial ant from the depot
One solution J achieved?
no
Move the ant to the next node according to the selection mechanism s
yes
Record the ant's path
Send out an ant to search the solution
Let nV=nV-1
Let nV=number of vehicle used in J and dJ=total travel length of J.
Resume the visiting record
Construct the routes in which the number of vehicle is equal to nV
dJ=total travel length of new solution
Feasible solution?
yes
no
no
Ant = ANT?
ant=ant+1
yes
nV=nV+1
Stage 1



Let e=0, ant = 0,
$dJ_{\varphi^{gl}} = dJ$.
send out the artificial ant from the depot
Move the ant to the next node according to the selection mechanism s
no
Global pheromone update
yes
ant=ANT?NT ?
Construct the routes in which the number of vehicle is equal to nV
Resume the visiting record
e=e+1
Local pheromone update
ant=ant+1
no
e= $EPOCH$?
Resume the visiting record
yes
$dJ_{\varphi^{gl}} > dJ$?
$dJ_{\varphi^{gl}} = dJ$
yes
no
$dJ_{\varphi^{gl}} = dJ_{\varphi^{gl}}$
Stage 2
Stop


Map Layers
mon_rout Selection Sets
mon1
mon2
mon3
mon4
mon5
mon6
mon7
mon8
mon9
mon10
mon11
mon12
mon13
mon14
mon15
Node Selection Sets
refill
depot
0
1
2
3
Kilometers

Vehicle routing-scheduling for municipal waste collection system under the "Keep Trash off the Ground" policy

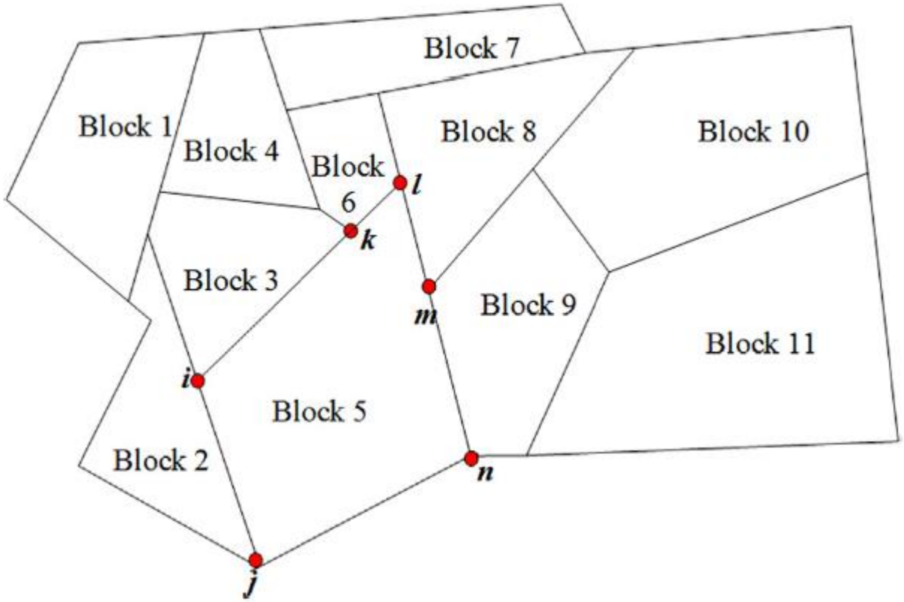


Fig. 1. Adjacency of blocks and intersections.

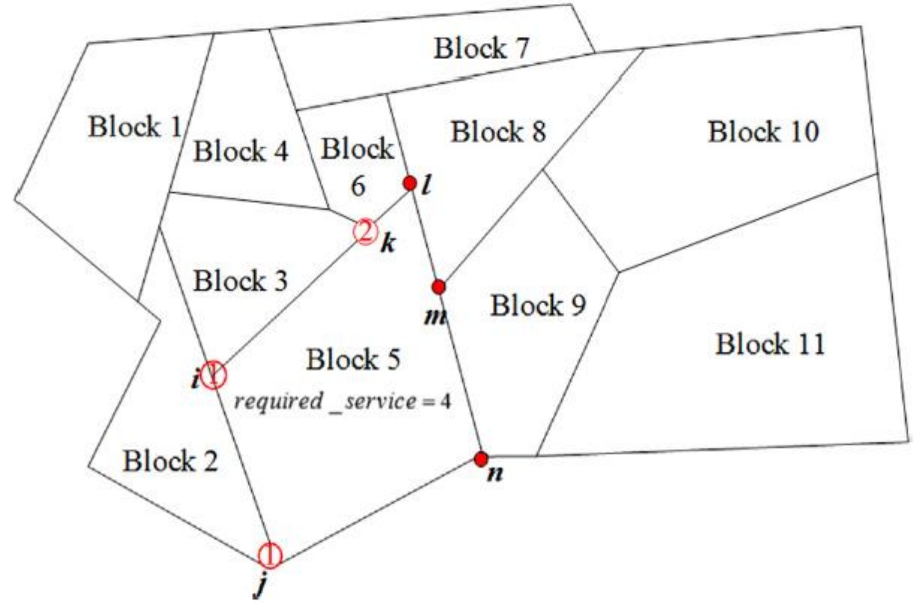
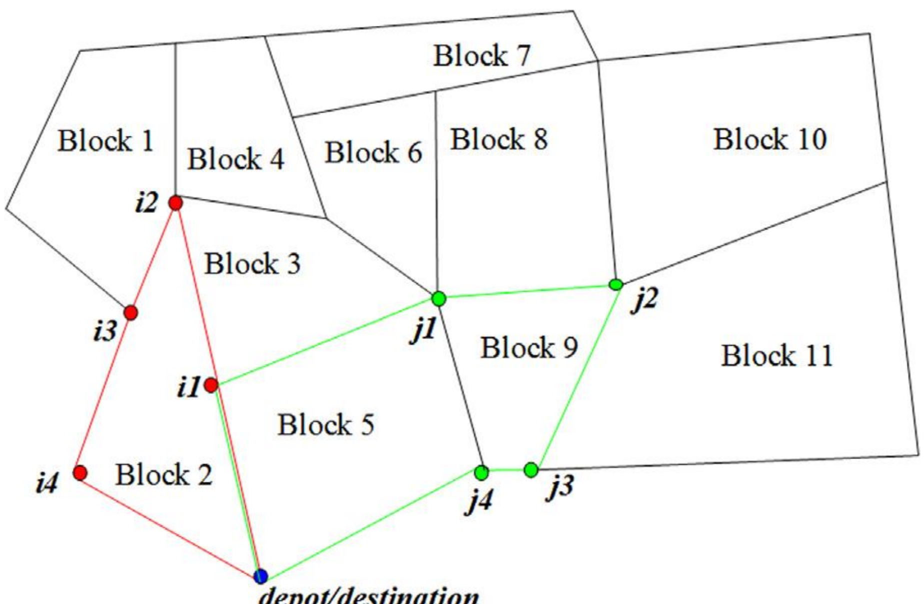
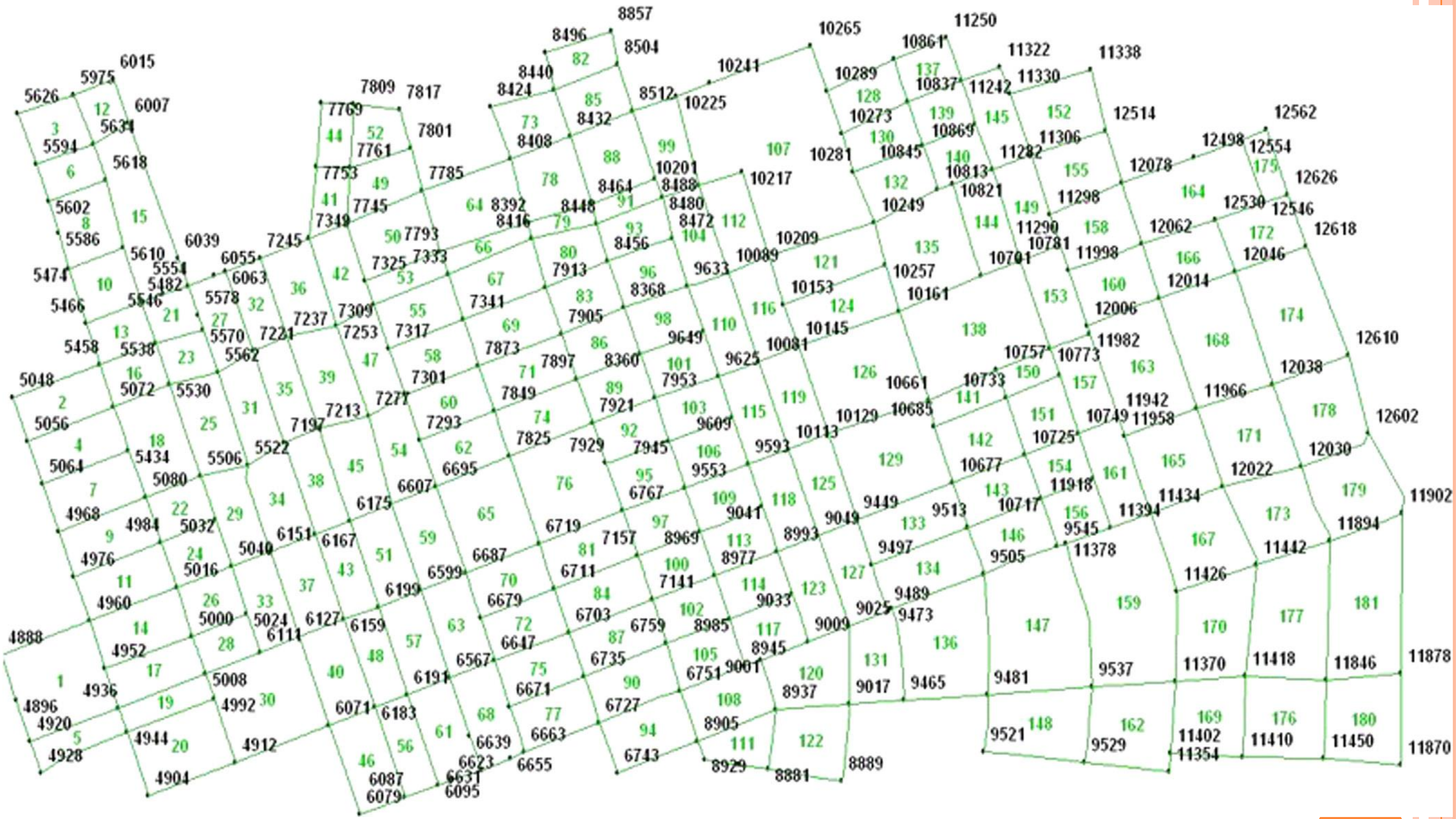
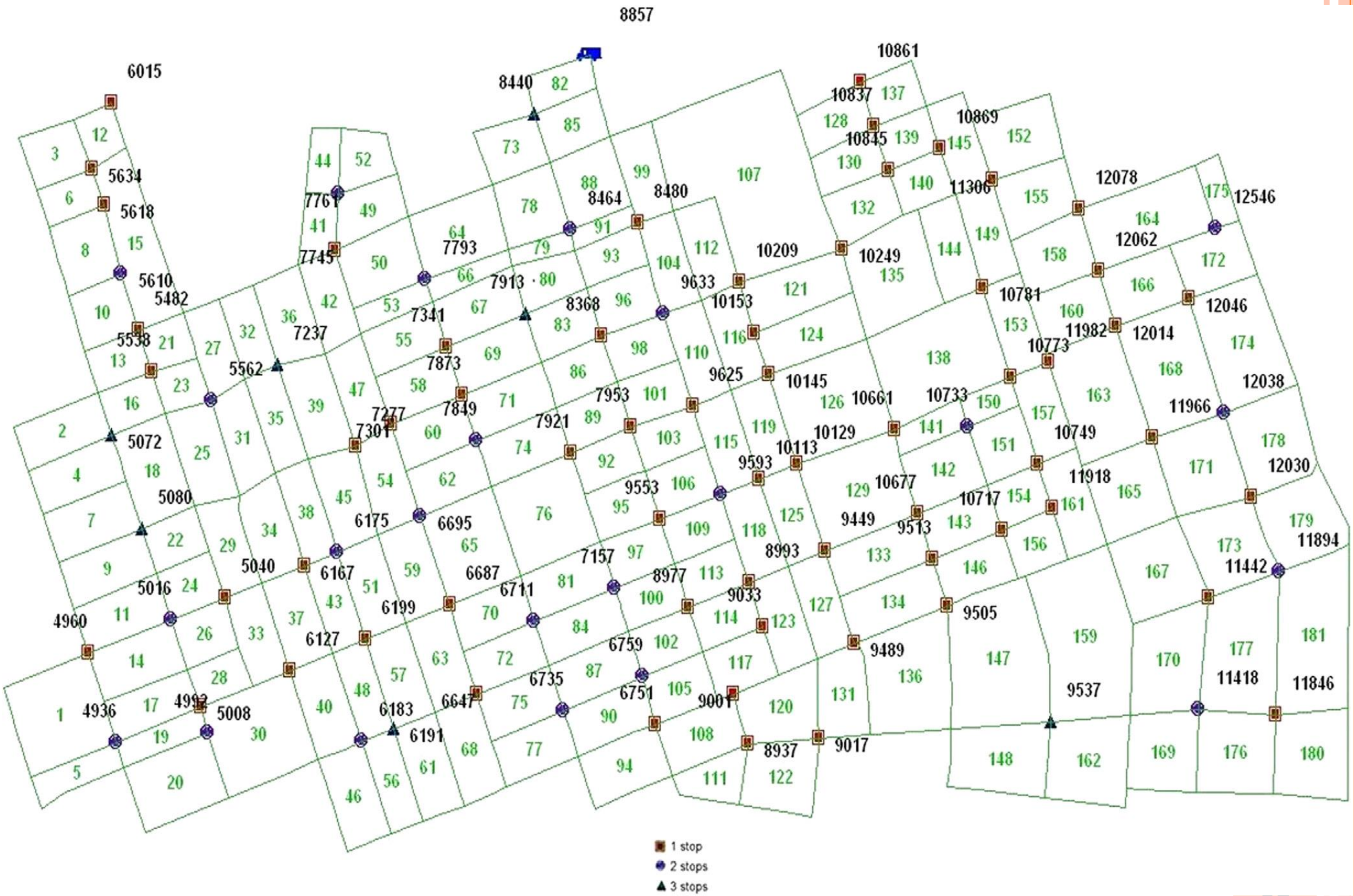
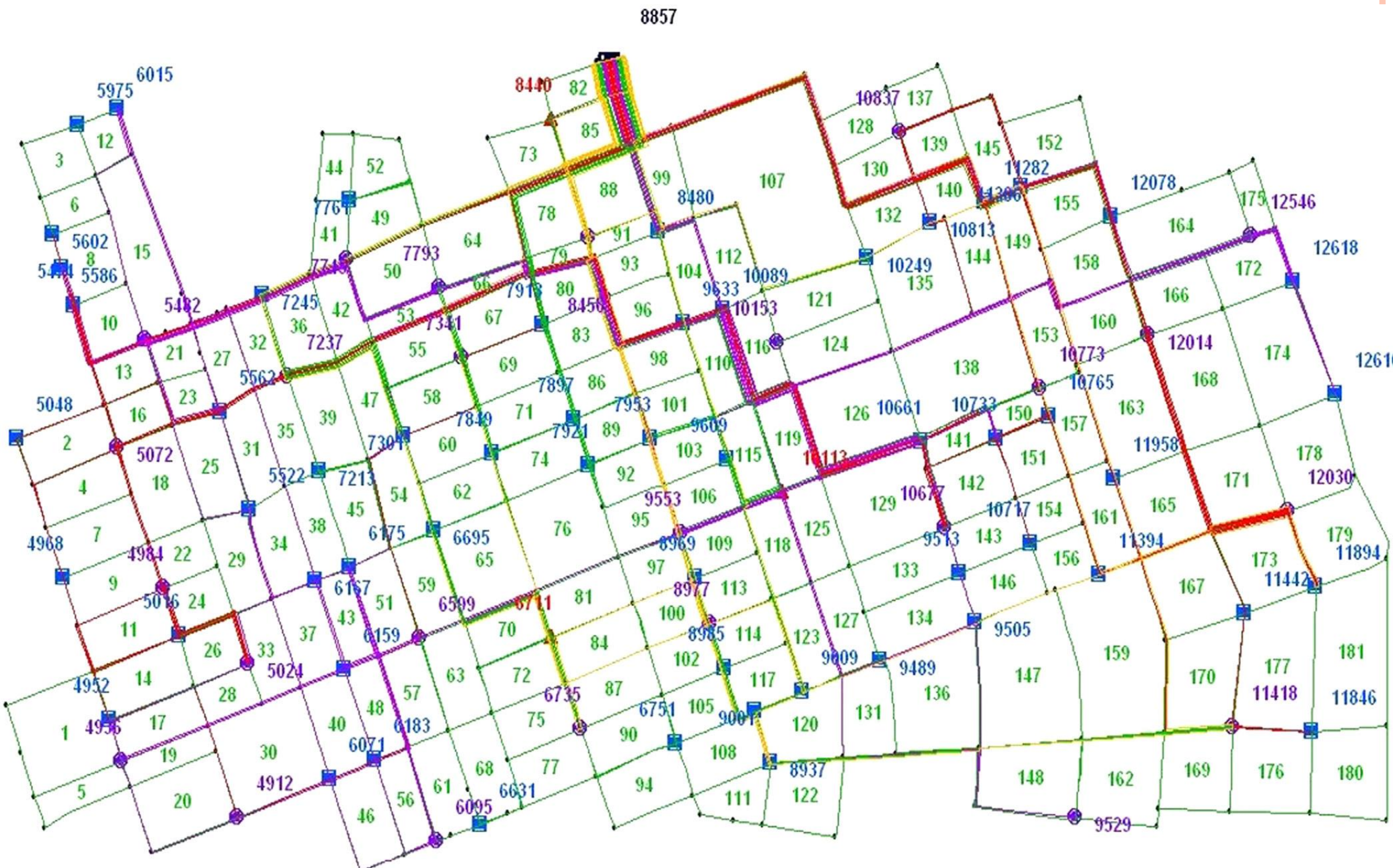


Fig. 2. A possible way to fulfill number of collection services for block 5.

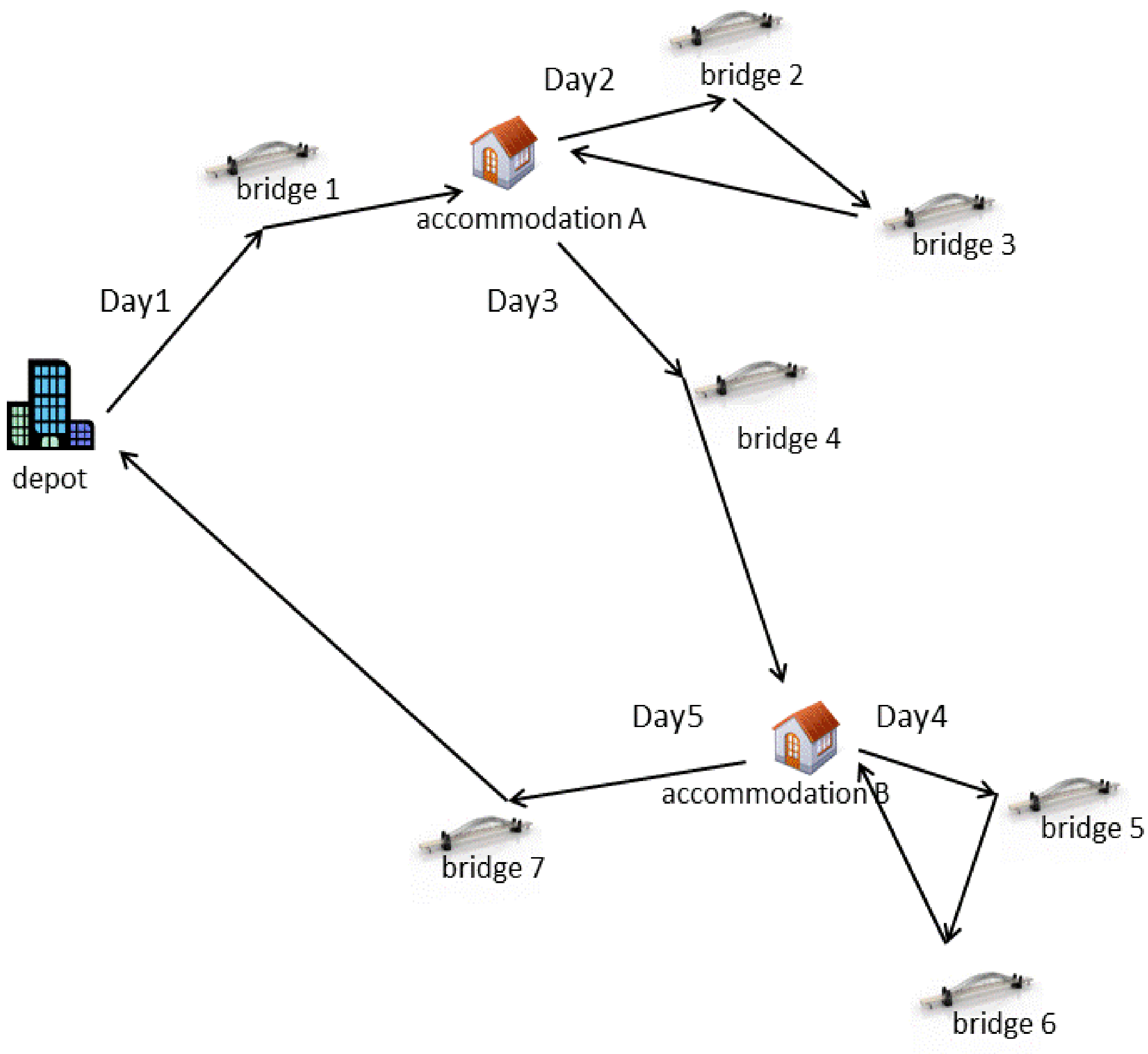





1 stop
2 stops
3 stops

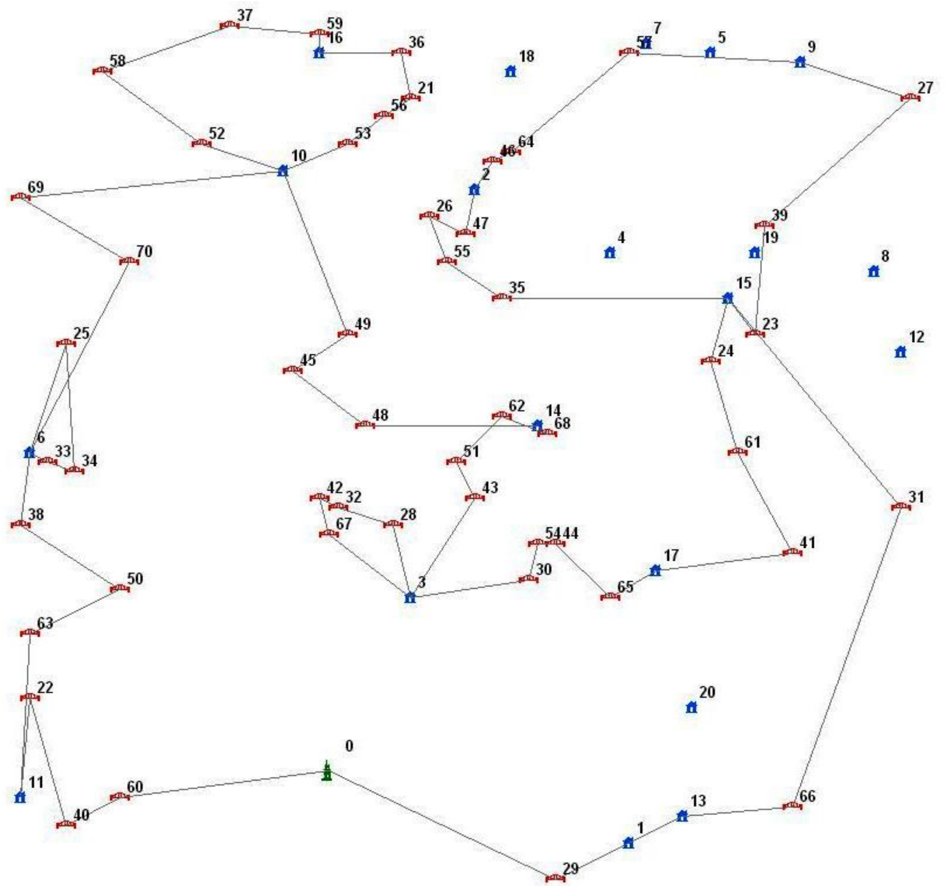


Solving the Bridge Inspection Routing Problem by Ant Colony Optimization


Day2
bridge 2
bridge 1
accommodation A
bridge 3
Day1
Day3
bridge 4
depot
Day5
Day4
accommodation B
bridge 5
bridge 7
bridge 6

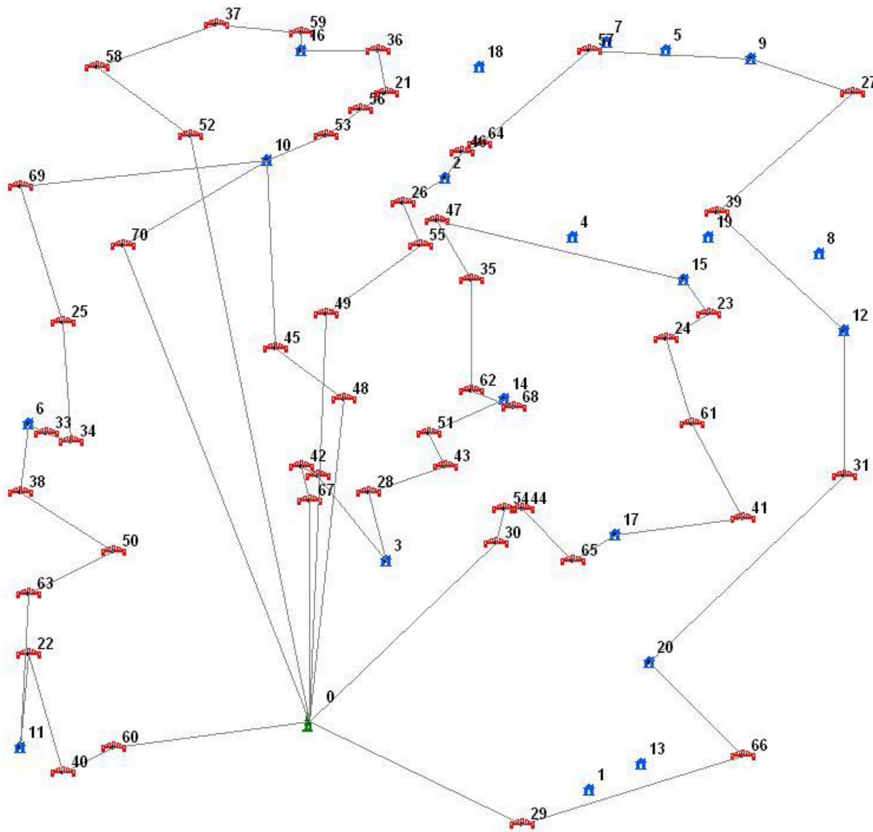
2025/9/29

TSP-based



0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

VRP-BASED



37
59
16
36
58
18
7
57
5
9
27
21
56
53
52
10
46
64
2
69
26
47
39
19
4
8
70
55
35
15
23
12
49
25
24
45
62
14
68
48
61
6
33
34
51
43
42
31
38
28
67
54
44
41
17
30
3
65
50
63
20
22
0
11
60
66
13
40
1
29

Ant colony optimization for railway driver crew scheduling: from modeling to implementation

For crew scheduling models with DA approach, the basic mathematics program can be expressed as

$$\min \sum_{j \in M} c_j x_j \tag{1}$$

$$\text{s.t.} \sum_{j \in M} a_{ij} x_j \geq 1 \quad \forall i \in N \tag{2}$$

$$x_j = 0, 1, \tag{3}$$

where c_j is the cost of duty j, N is the set of all trips in the schedule, and M is the set of all duties generated; $a_{ij}=1$ if trip i participates in duty j, $a_{ij}=0$ otherwise; and $x_i=1$ if duty j participates in the solution, $x_j=0$ otherwise.

dead-heading-not-allowed approach vs. dead-heading allowed approach

DNA approach:

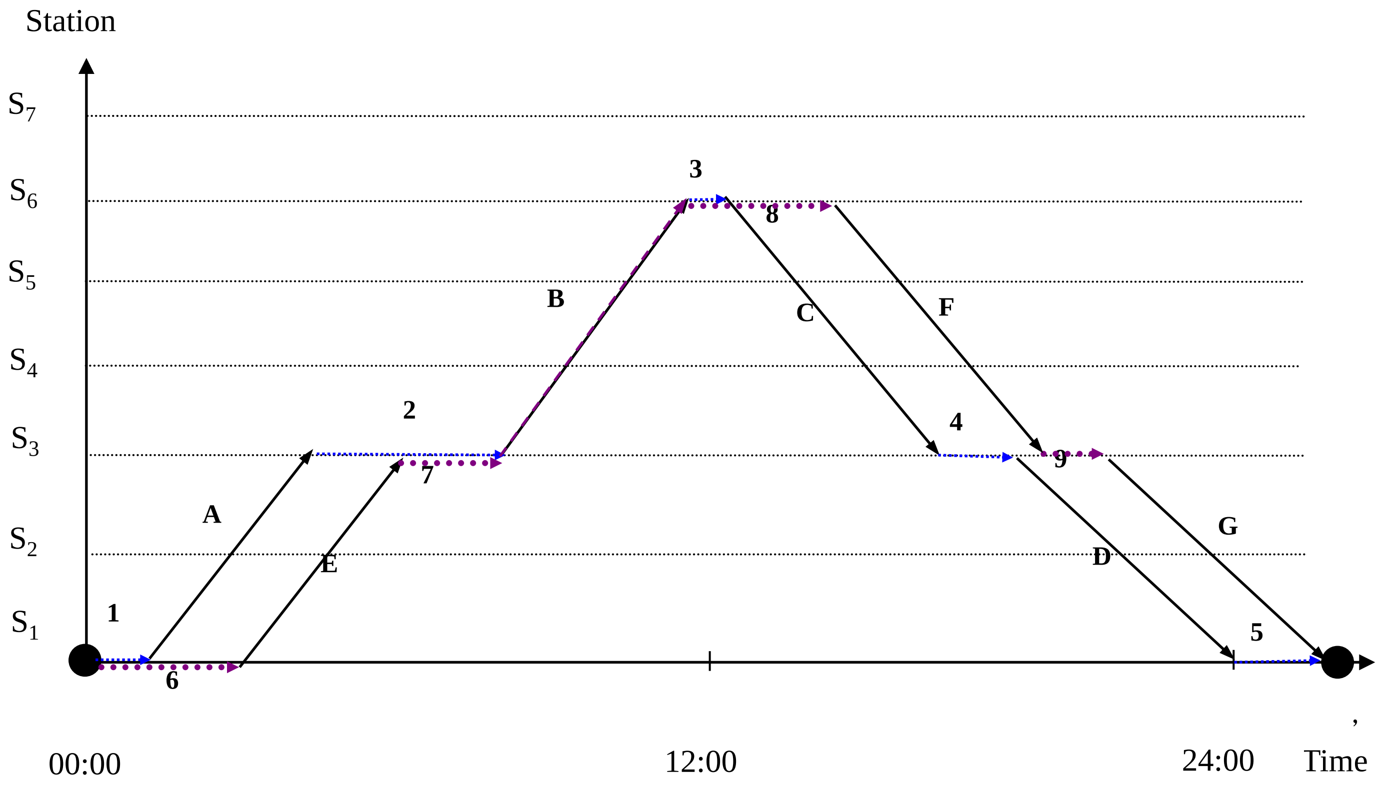
$$\min \left[|K|, \sum_{k \in K} \sum_{\substack{(i,j) \in A' \\ i \neq j}} c_{ij} x_{ij}^{k} \right] \tag{5}$$

$$\text{s.t.} \sum_{k \in K} \sum_{i \in N} y_i^k = 1 \tag{6}$$

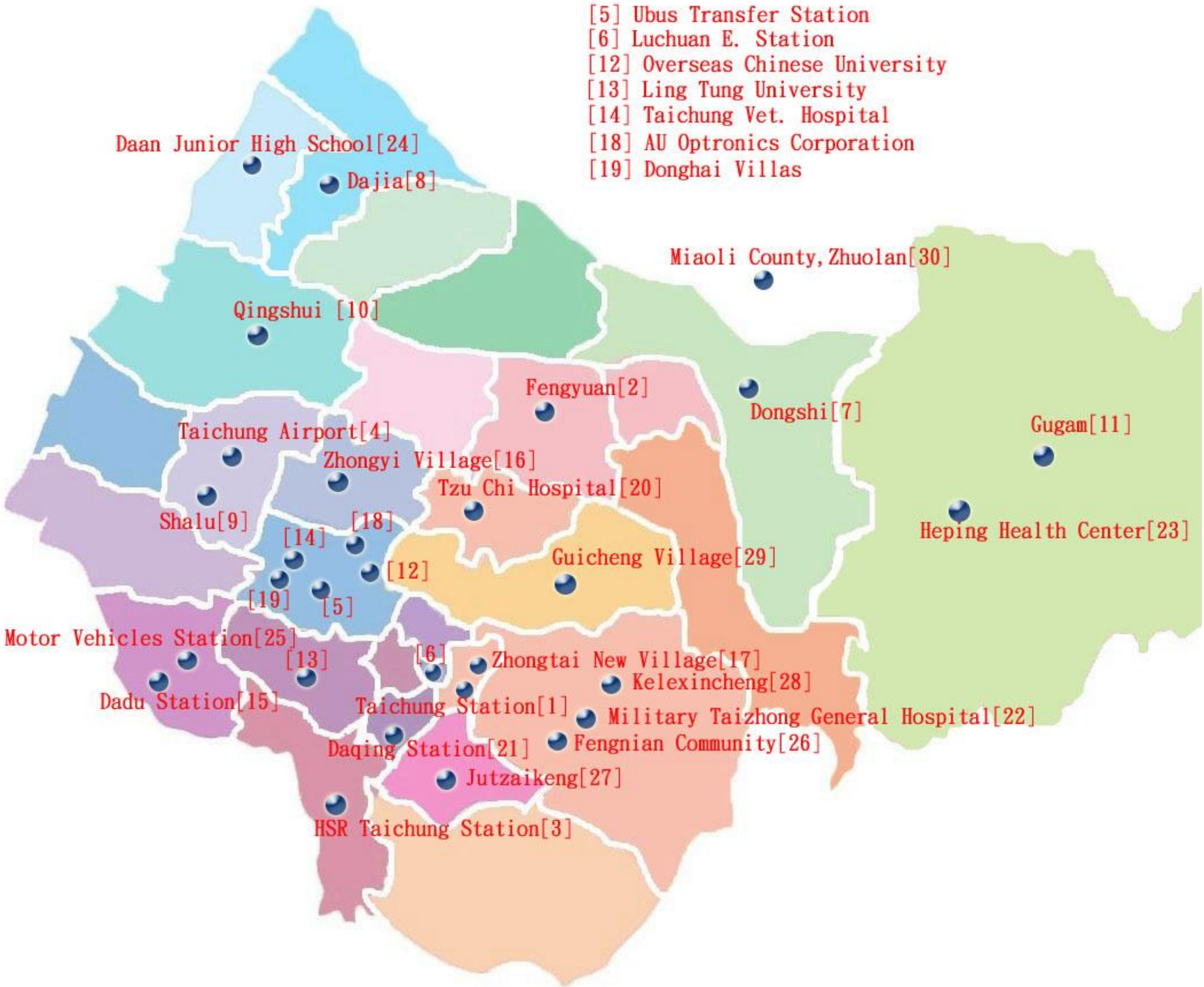
$$\sum_{k \in K} \sum_{i \in N, i \neq O} x_{Oi}^{k} = |K| \tag{7}$$

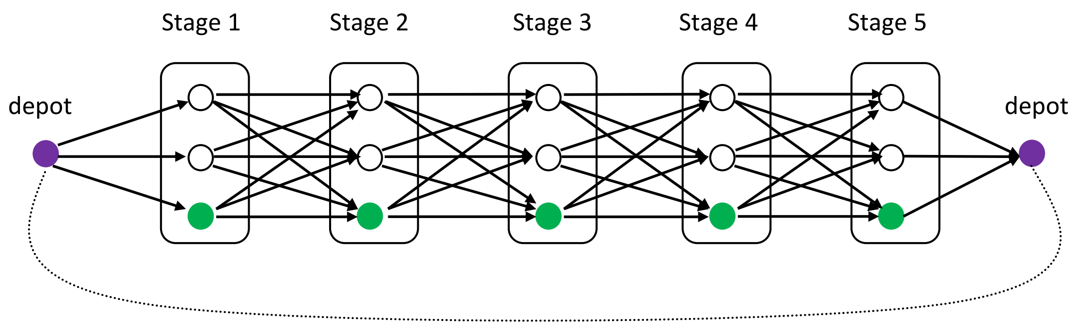
VRP應用

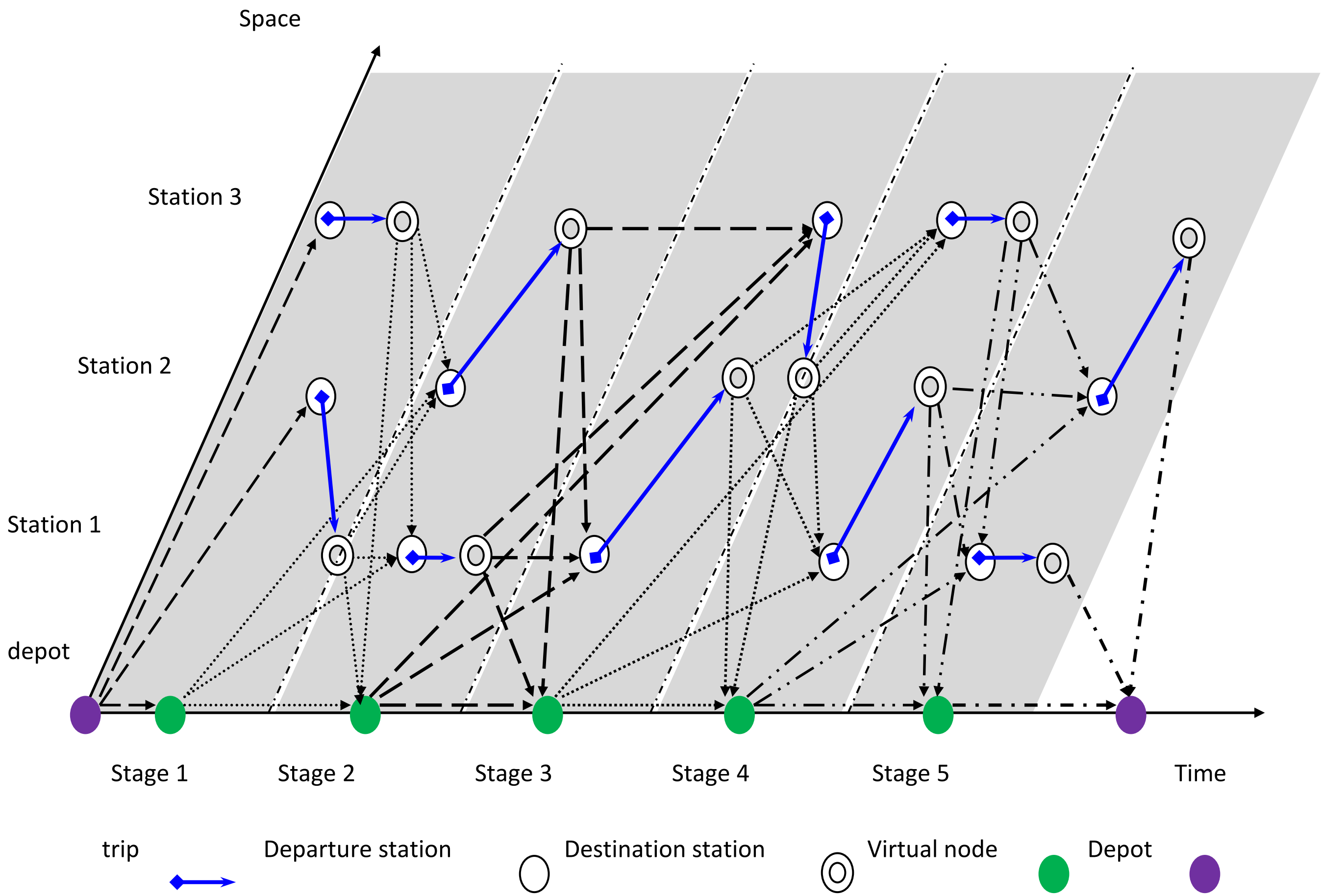
- 駕駛(司機員)指派



Multi-Depot Bus Scheduling Problem


[5] Ubus Transfer Station
[6] Luchuan E. Station
[12] Overseas Chinese University
[13] Ling Tung University
[14] Taichung Vet. Hospital
[18] AU Optronics Corporation
[19] Donghai Villas
Daan Junior High School[24]
Dajia[8]
Miaoli County, Zhuolan[30]
Qingshui [10]
Fengyuan[2]
Dongshi[7]
Gugam[11]
Taichung Airport[4]
Zhongyi Village[16]
Tzu Chi Hospital[20]
Shalu[9]
[18]
[14]
[12]
Heping Health Center[23]
Guicheng Village[29]
[19]
[5]
Motor Vehicles Station[25]
[13]
[6]
Zhongtai New Village[17]
Kelexincheng[28]
Dadu Station[15]
Taichung Station[1]
Military Taizhong General Hospital[22]
Daqing Station[21]
Fengnian Community[26]
Jutzaikeng[27]
HSR Taichung Station[3]


Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
depot
depot


Space
Station 3
Station 2
Station 1
depot
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Time
trip
Departure station
Destination station
Virtual node
Depot

Open Vehicle Routing Problem

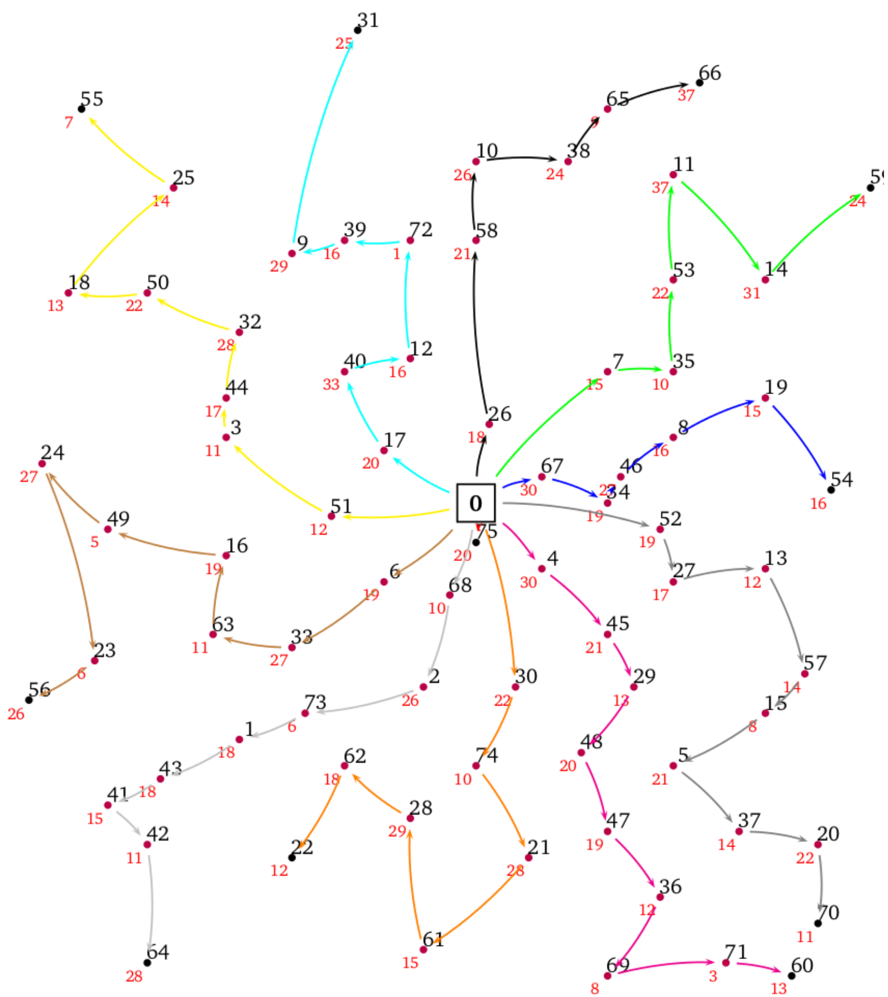


Figure 1: Solution for C7

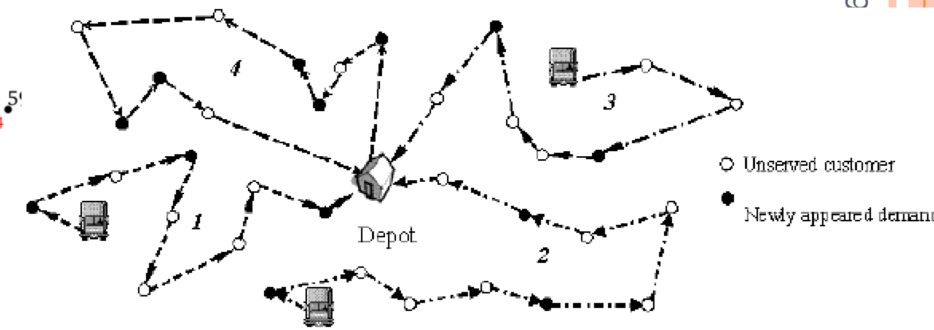
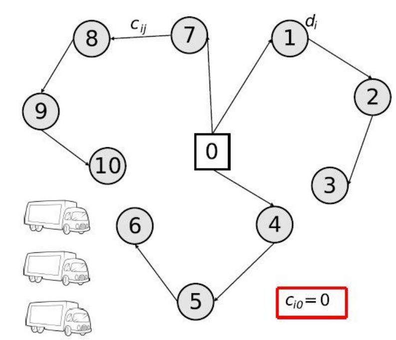
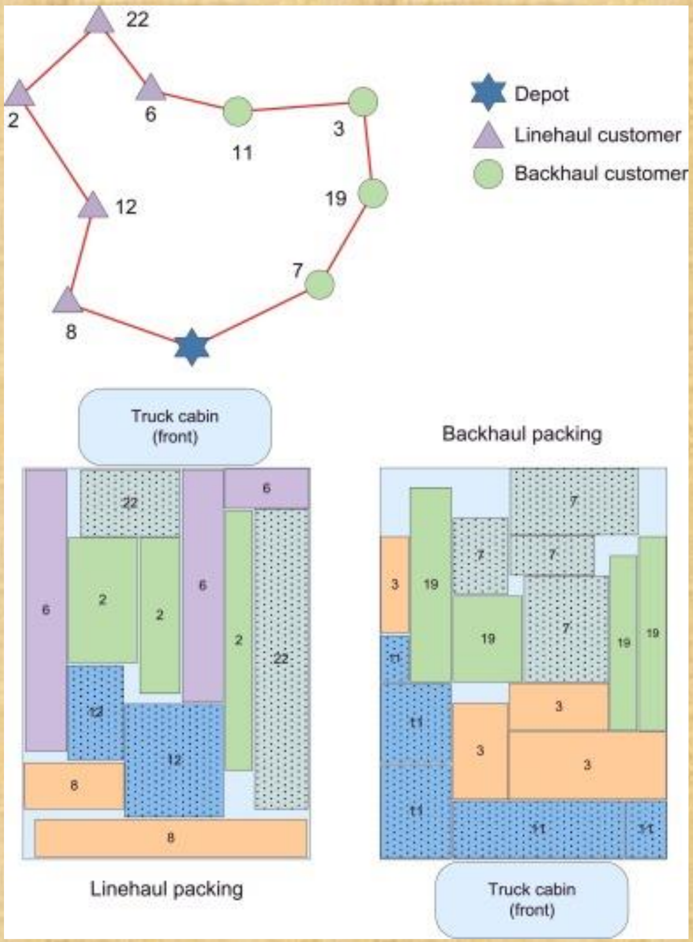


Figure 1 An instance of the DVRP problem

OVRP (open vehicle routing problem)

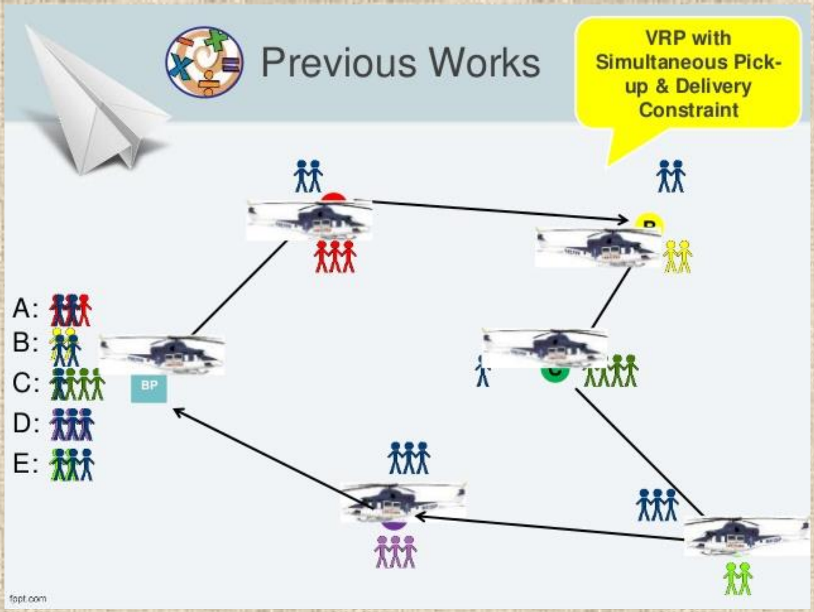


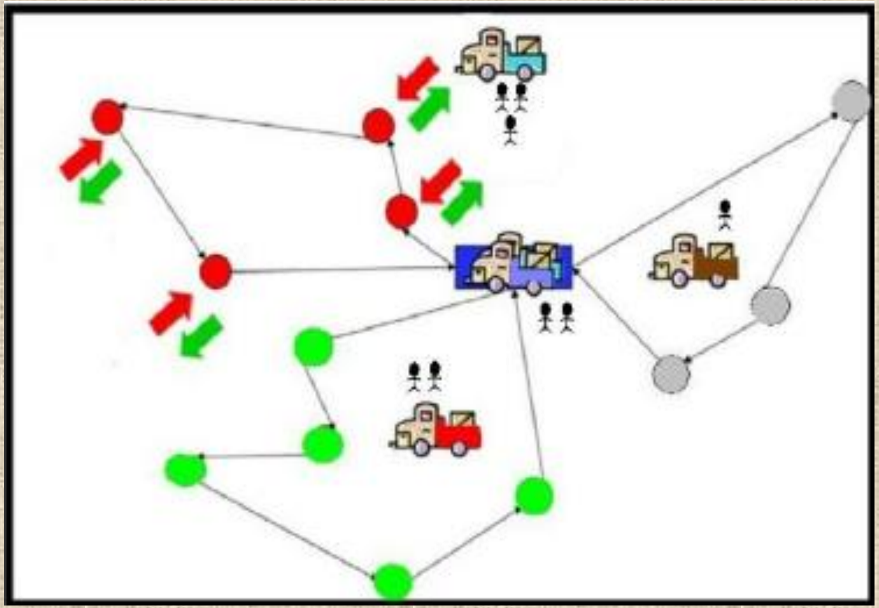
VRP WITH BACKHAUL



Dominguez et al. (2016)

Pickup and Delivery VRP





- one-to-many-to-one pickup and delivery problem(1-M-1 PDP)
- one-to-one pickup and delivery problem (1-1 PDP)
- many-to-many pickup and delivery problem (M-M PDP)

Balanced VRP

Possible objectives:
Put it to the constraints as threshold

A formal definition of each OF under study is presented:

All-min

$$min \sum_{t \in T} (l_t - min_{u \in T} l_u) \quad (1)$$

Max-min

$$min(max_{u \in T} l_u - min_{u \in T} l_u) \quad (2)$$

Min-max

$$min\ max_{u \in T} l_u \quad (3)$$

Rel

$$min \frac{1}{|T|} \sum_{t \in T} \left(\frac{max_{u \in T} l_u - l_t}{max_{u \in T} l_u} \right) \quad (4)$$

Var

$$min \left(\frac{\sum_{t \in T} l_t^2}{|T|} - \left(\frac{\sum_{t \in T} l_t}{|T|} \right)^2 \right) \quad (5)$$

MAD

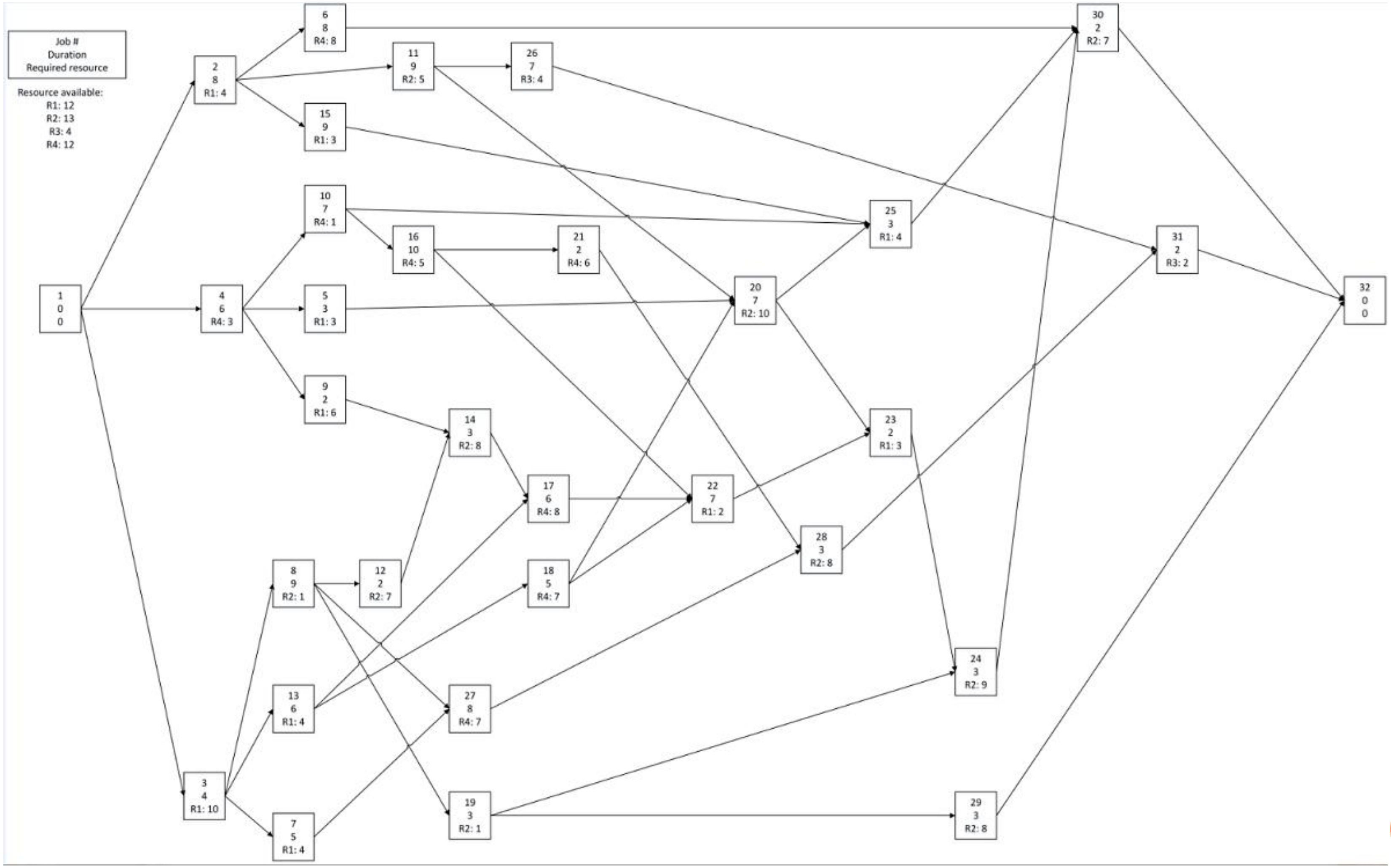
$$min \frac{1}{|T|} \sum_{t \in T} |l_i - \bar{l}| \quad (6)$$

Gini

$$min \frac{1}{2n^2 \bar{l}} \sum_{t \in T} \sum_{t' \in T} |l_t - l_{t'}| \quad (7)$$

DRONE DISTRIBUTION

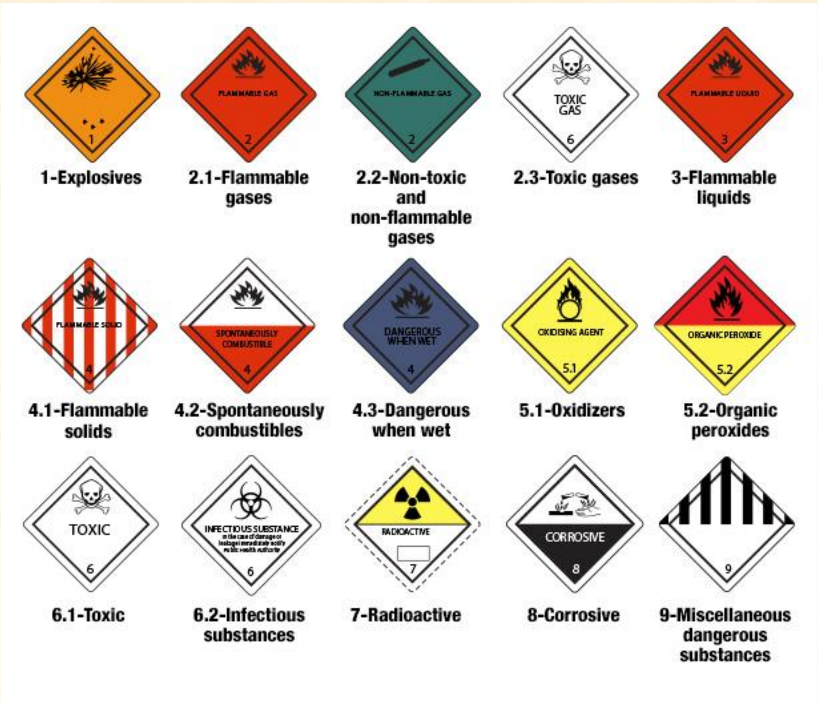
Project scheduling



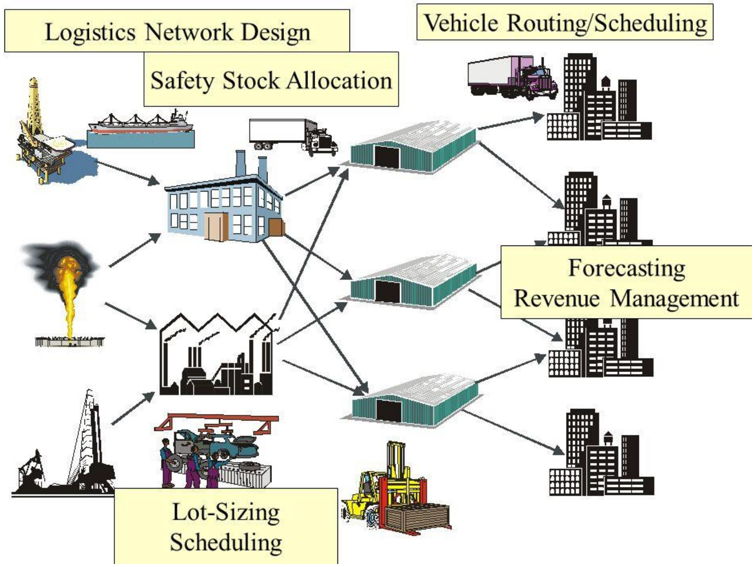
Feeder Vehicle Routing Problem

- Smaller capacity vehicles to cooperate with a truck.

VEHICLE ROUTING PROBLEM FOR HAZARDOUS MATERIALS TRANSPORTATION



Emergency Vehicle Routing Problem



Alleviate carbon dioxide emissions in Vehicle Routing Problem

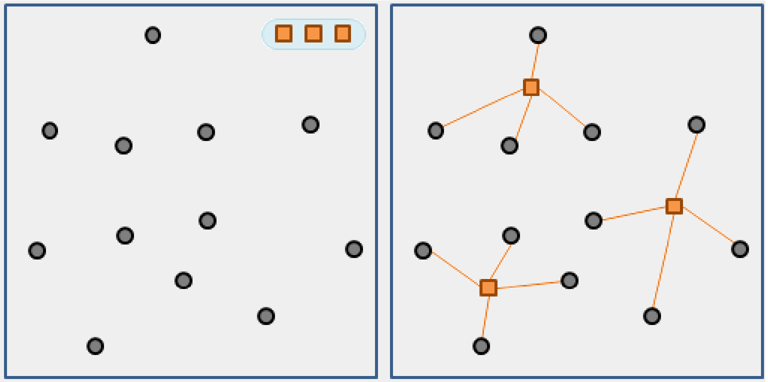
- Alleviate carbon dioxide emissions in Vehicle Routing Problem is actually the side effect of the objective in which the travel length is minimized.

Electric vehicle

Facility Location Problem

Facility Location Problem

- **P-median problem (min-sum)**
  - In the p-median problem p facilities have to be located on a graph such that the sum of distances between the nodes of the graph and the facility located nearest is minimized.



$$y_i = \begin{cases} 1 \text{ if faciluty } i \text{ is opened} \\ 0, \text{ otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 \text{ if client } j \text{ is serviced by facility } i, \\ 0 \text{ otherwise.} \end{cases}$$

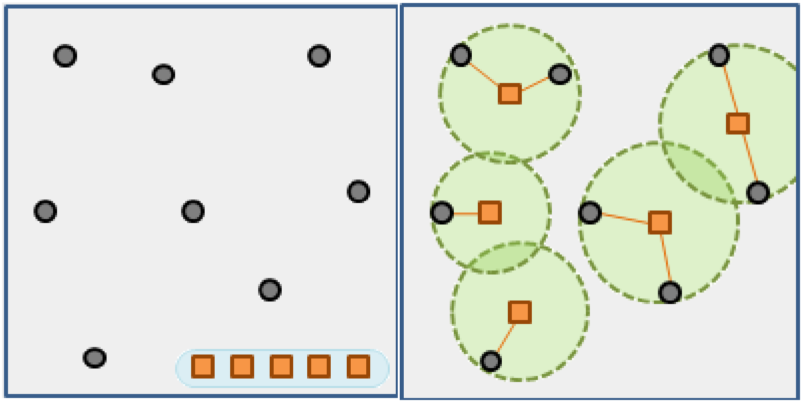
$$\sum_{j \in J} p_{ij} x_{ij} \le V_i y_i, \quad i \in I,$$

$$\min \left\{ \sum_{i \in I} c_i y_i + \sum_{i \in I} \sum_{j \in J} g_{ij} x_{ij} \right\}$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J,$$

$$x_{ij},\ y_i \in \{0, 1\},\ i \in I,\ j \in J.$$

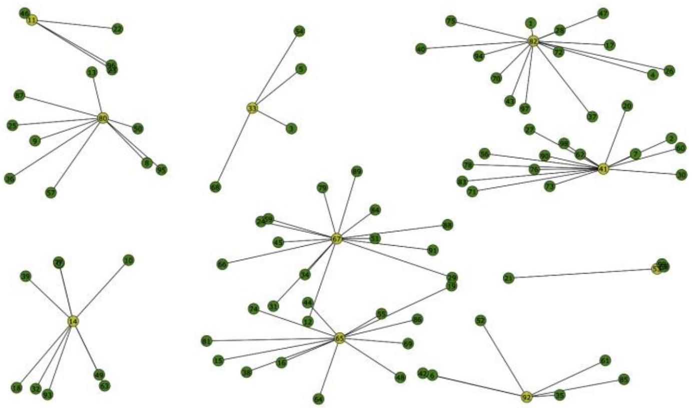
- **P-center problem (min-max)**
  - p-center problem the aim of which is to locate p facilities such that the maximum distance is minimized.



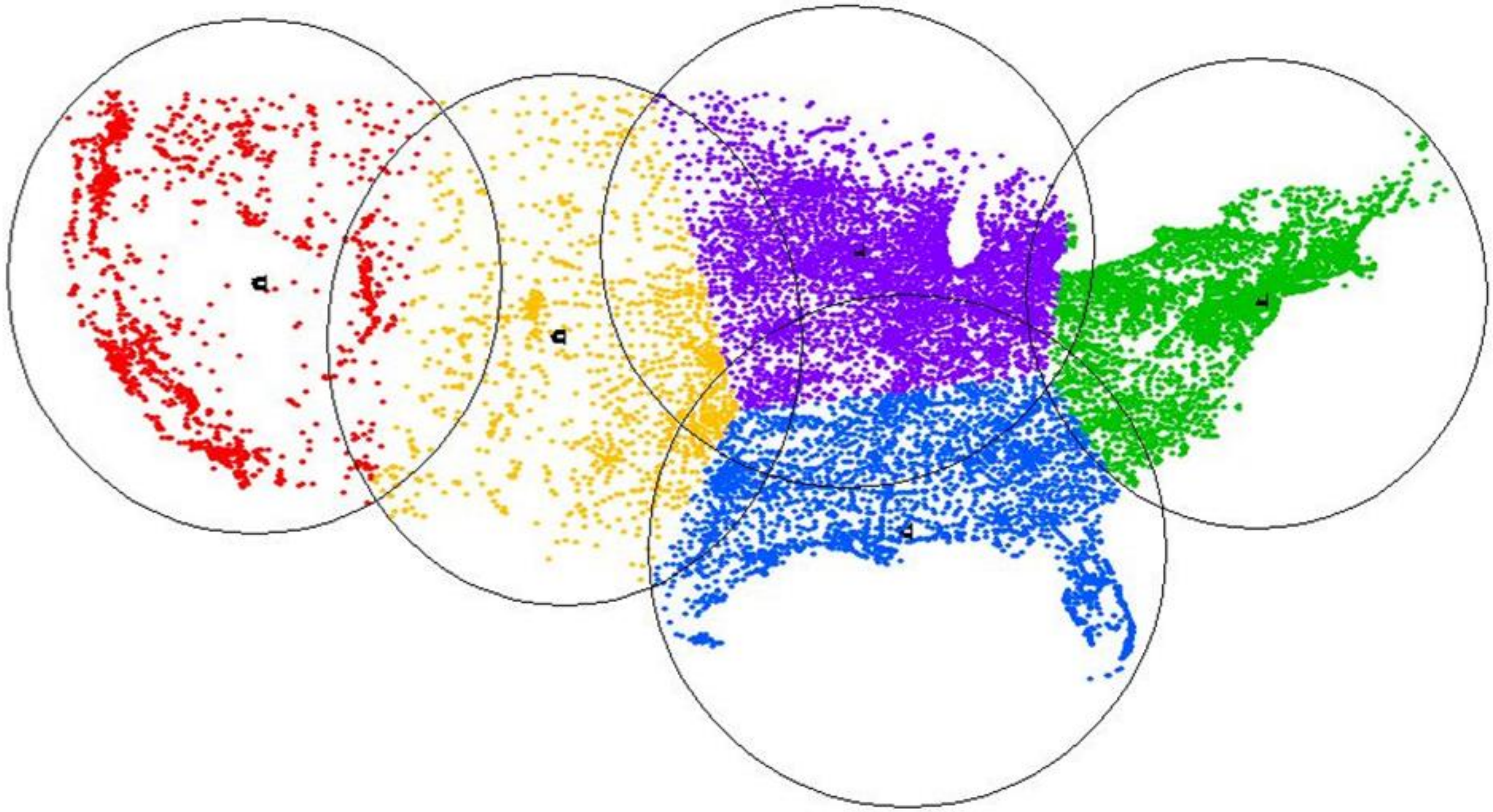
- **Minimax facility location**
  - The **minimax facility location** problem seeks a location which minimizes the maximum distance to the sites.
  - It the case of the Euclidean metric it is known as the smallest enclosing sphere problem or 1-center problem. Its study traced a least to the year of 1860. The planar case (smallest enclosing circle problem) may be solved in optimal time.

2025/9/29

- **Maxmin facility location**
  - The **maxmin facility location** or **obnoxious facility location** problem seeks a location which maximizes the minimum distance to the sites. It the case of the Euclidean metric it is known as the largest empty sphere problem. The planar case (smallest enclosing circle problem) may be solved in optimal time



- **Nondeterministic p-median problem**
  - Find p facilities (p is not determined) to minimize the sum of distances.
- **Nondeterministic p-center problem**
- **Nondeterministic demand p-median problem**
- **Nondeterministic demand p-center problem**



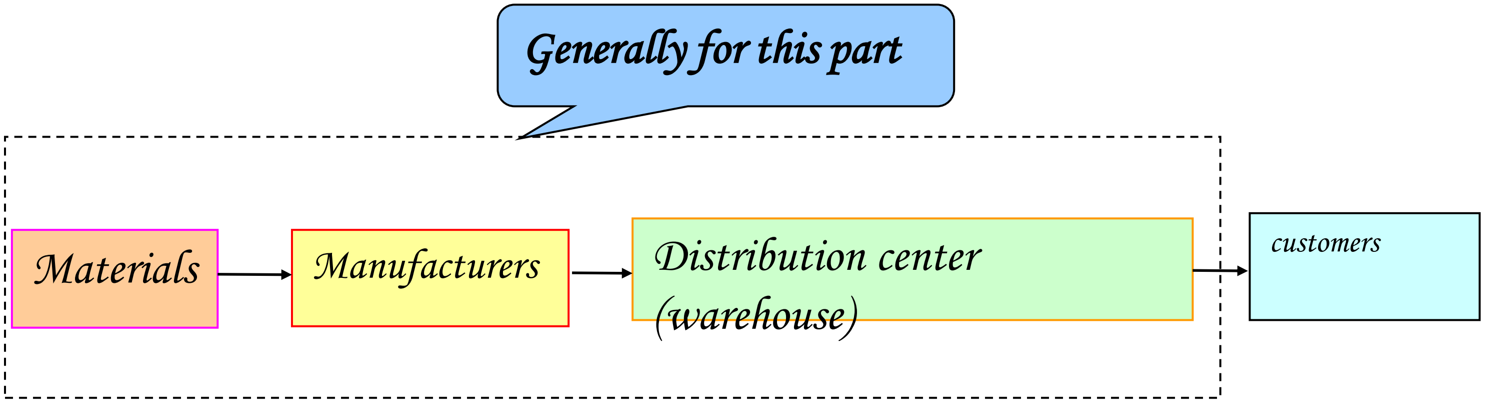
2025/9/29

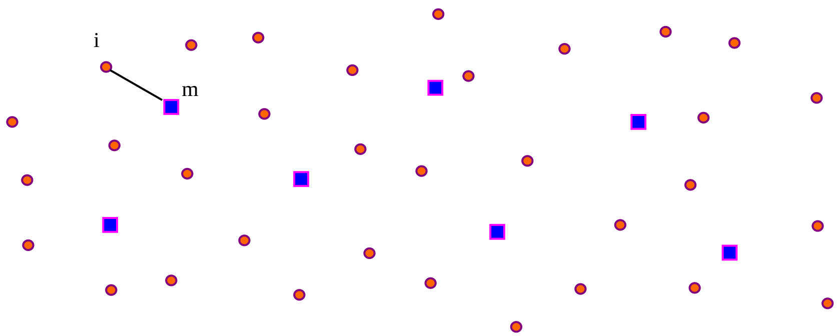
Location Routing Problem

Location Routing Problem

- **the location-routing problem** (LRP)
  - seeks to minimize total cost by simultaneously selecting a subset of candidate facilities and constructing a set of delivery routes that satisfy the following constraints:
    - (i) customer demands are satisfied without exceeding vehicle or facility capacities;
    - (ii) the number of vehicles, route lengths, and route durations do not exceed the specified limits; and
    - (iii) each route begins and ends at the same facility.

Example 1 Capacitated Facility Location Problem

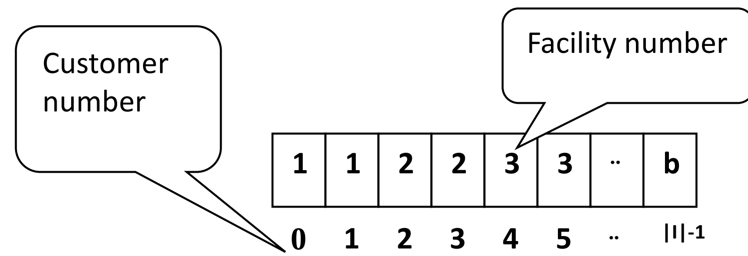




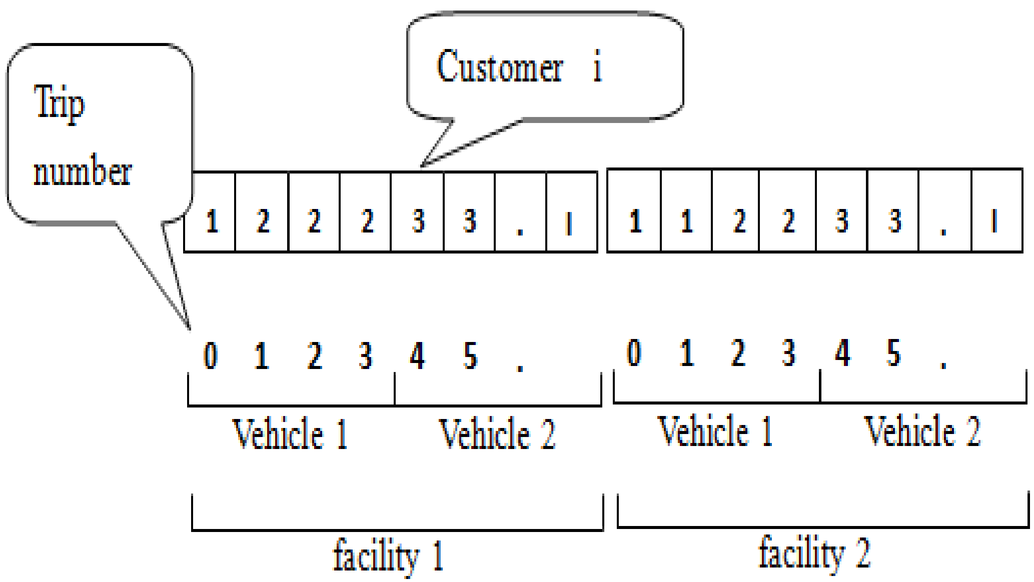
A NESTED GENETIC OPTIMIZATION ALGORITHM FOR THE CAPACITATED FACILITY LOCATION PROBLEM

0	1	0	1	1	0	1.	0	1.	0

Chromosome of GA-I



chromosome of GA-II



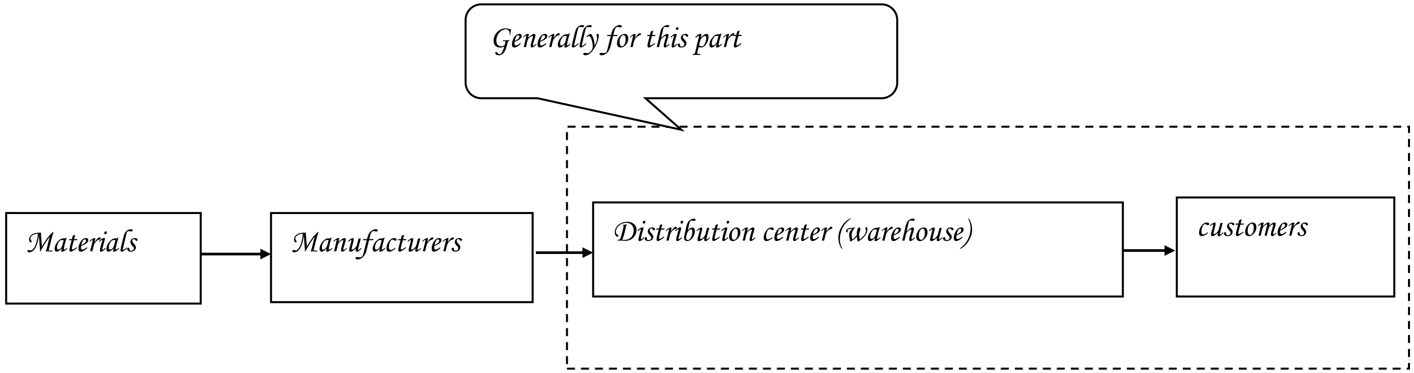
Chromosome of GA-III

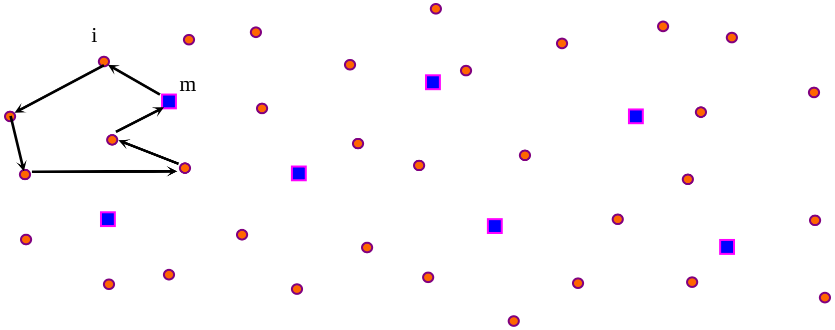
EXAMPLE 2 STOCHASTIC CAPACITATED FACILITY LOCATION PROBLEM

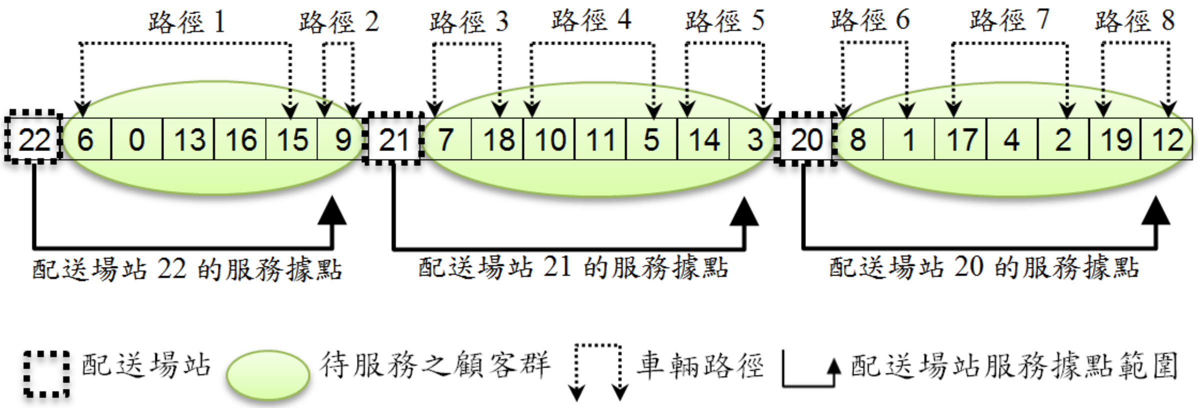
Facility cost + transportation cost + vehicle cost + penalty.

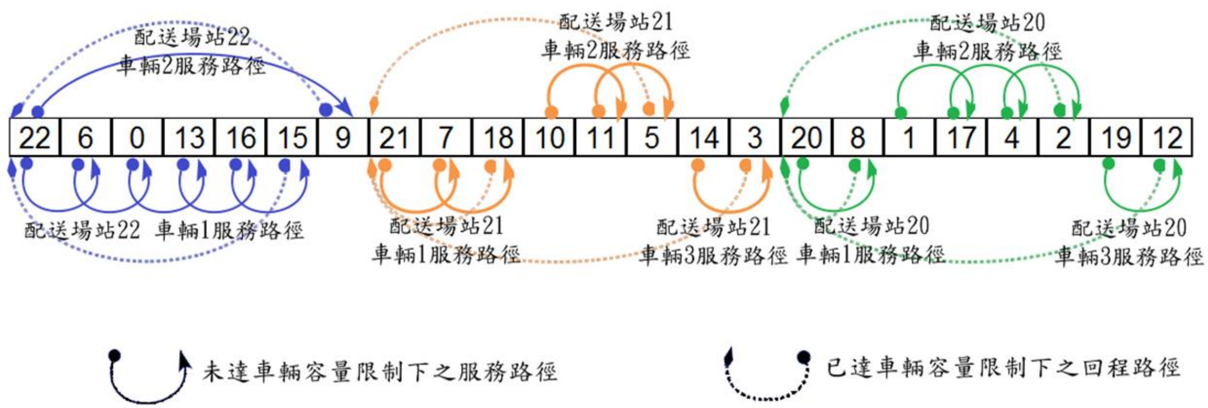
$$\min \quad \sum_m C_m(\pi_m) y_m + H \sum_m \sum_i 2 c_{mi} \lambda_i x_m^i + O_k \sum_m |K_m| + \sum_i p_i \theta_i$$

Example 3 Capacitated Location Routing Problem






路徑 1
路徑 2
路徑 3
路徑 4
路徑 5
路徑 6
路徑 7
路徑 8
22 6 0 13 16 15 9 21 7 18 10 11 5 14 3 20 8 1 17 4 2 19 12
配送場站 22 的服務據點
配送場站 21 的服務據點
配送場站 20 的服務據點
配送場站
待服務之顧客群
車輛路徑
配送場站服務據點範圍


配送場站22
車輛2服務路徑
配送場站21
車輛2服務路徑
配送場站20
車輛2服務路徑
22 6 0 13 16 15 9 21 7 18 10 11 5 14 3 20 8 1 17 4 2 19 12
配送場站22 車輛1服務路徑
配送場站21
車輛1服務路徑
配送場站21
車輛3服務路徑
配送場站20
車輛1服務路徑
配送場站20
車輛3服務路徑
未達車輛容量限制下之服務路徑
已達車輛容量限制下之回程路徑

2025/9/29

Chromosome❶為隨機產生之二進制

25	2	17	9	10	22	3	7	5	13	20	18	12	1	4	23	14	15	16	19	8	11	6	24	21	Chromosome①
0	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	0	Chromosome❶

Chromosome❷為 Chromosome①與 Chromosome②比對後

所產生之相反二進制編碼

24	8	16	1	9	22	18	15	14	10	7	5	6	2	23	20	21	12	19	17	4	3	13	11	25	Chromosome②
1	0	1	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	Chromosome❷

合併至 Chromosome③：

Chromosome①對應 Chromosome❶之基因型串資料為 0 的基因

Chromosome②對應 Chromosome❷之基因型串資料為 0 的基因

合併至 Chromosome❸：

Chromosome①對應 Chromosome❶之基因型串資料為 1 的基因

Chromosome②對應 Chromosome❷之基因型串資料為 1 的基因

交配點

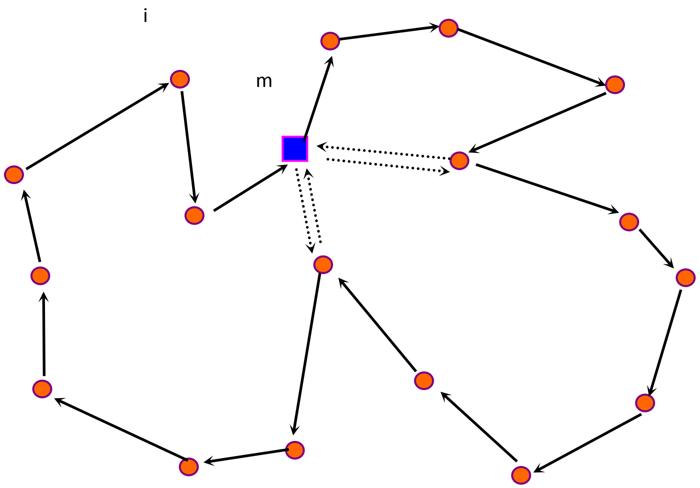
25	2	10	7	5	13	18	12	23	14	16	11	24	21	8	1	9	22	15	6	20	19	17	4	3	Chromosome③
24	16	18	14	10	7	5	2	23	21	12	13	11	25	17	9	22	3	20	1	4	15	19	8	6	Chromosome❸

交配點

Example 4 Stochastic Capacitated Location Routing Problem

(facility opening cost+ route cost + traveling cost + penalty cost)

$$\min \quad z = FC + RC + TC + PC$$



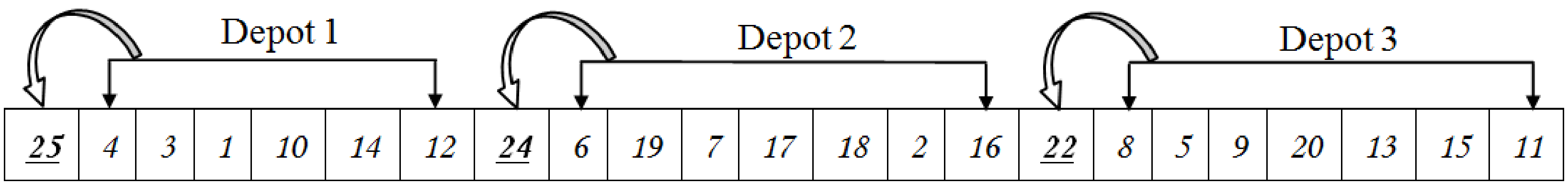
$$FC = \sum_m F_m y_m$$

$$RC = O_k \sum_m \sum_n (n \Pi_n^m y_m)$$

$$TC = \sum_m (\sum_i \sum_j \sum_k c_{ij} z_{ij}^k x_m^i x_m^j + 2 \sum_{\eta=2} c_{m\varphi_\eta} \Pi_\eta^m) y_m$$

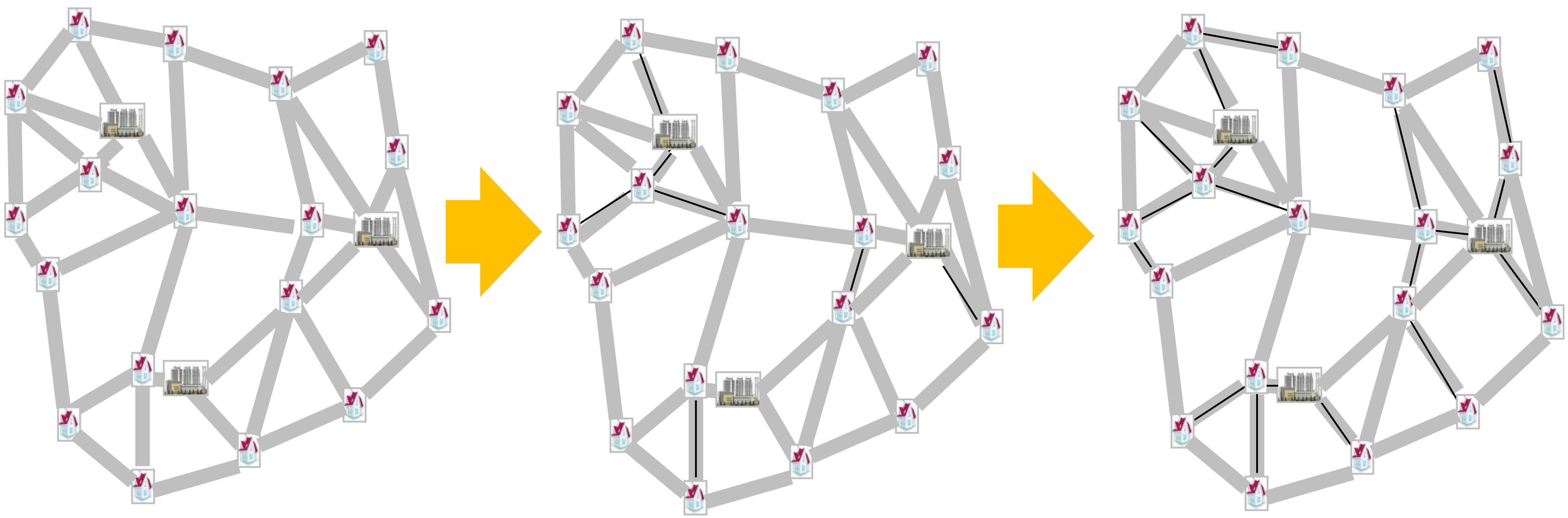
$$PC = p \sum_{ex=0}^{\infty} (ex(\Pr((\sum_m \sum_i x_m^i d_i - \pi_m) = ex)))$$

Using Genetic Algorithm to Solve the Capacitated Facility Location Routing Problem with Stochastic Demand

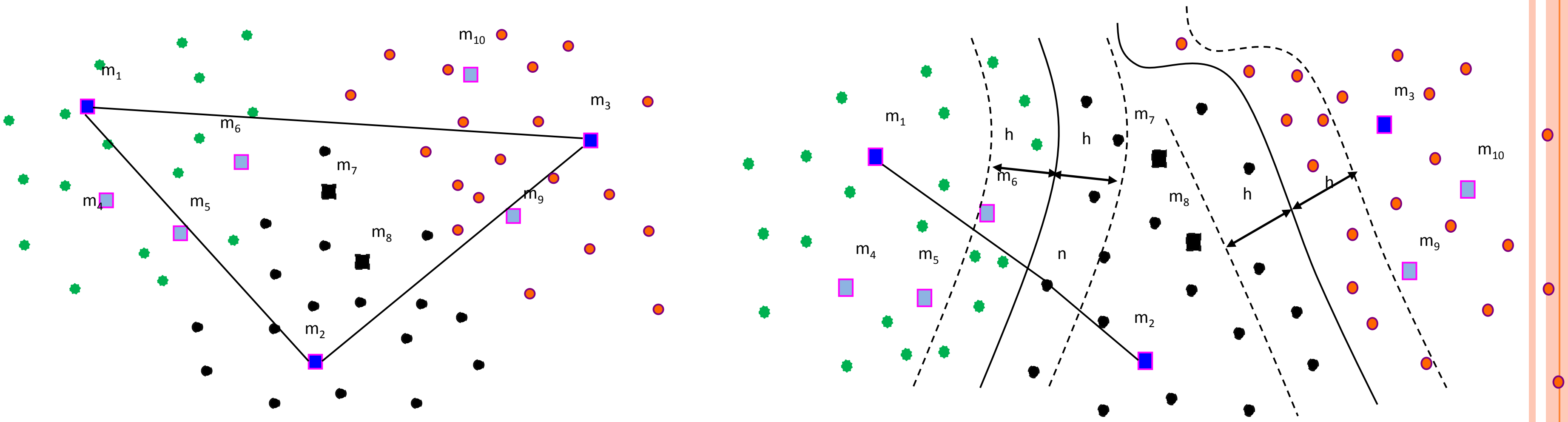


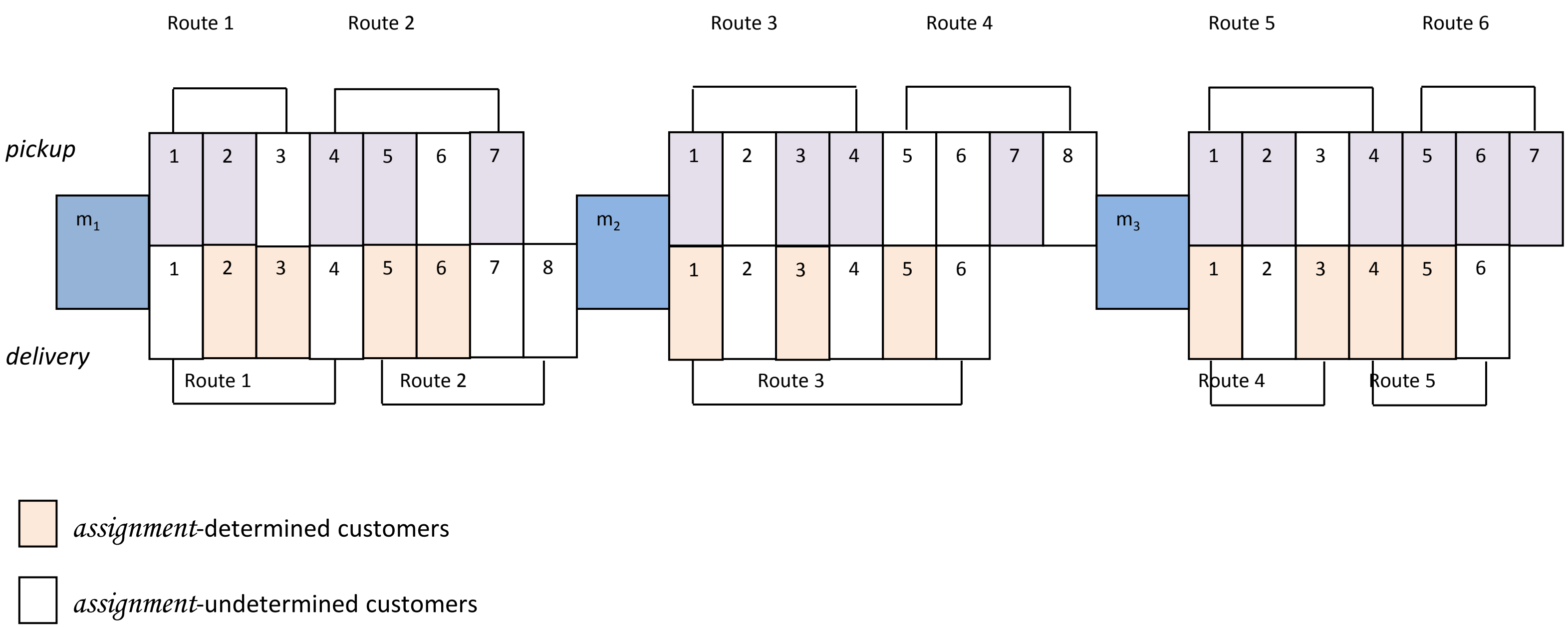
Solve Facility Location Problem in Upstream of Supply Chain using Harmony Search

2025/9/29



Solving the Capacitated Location Routing Problem with Pickup and Delivery Routes




Route 1
Route 2
Route 3
Route 4
Route 5
Route 6
pickup
delivery
m_1
m_2
m_3
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6
Route 1
Route 2
Route 3
Route 4
Route 5
assignment-determined customers
assignment-undetermined customers

2025/9/29

Solving the Multi-Compartment Capacitated Location Routing Problem with Pickup-Delivery Routes and Stochastic Demands

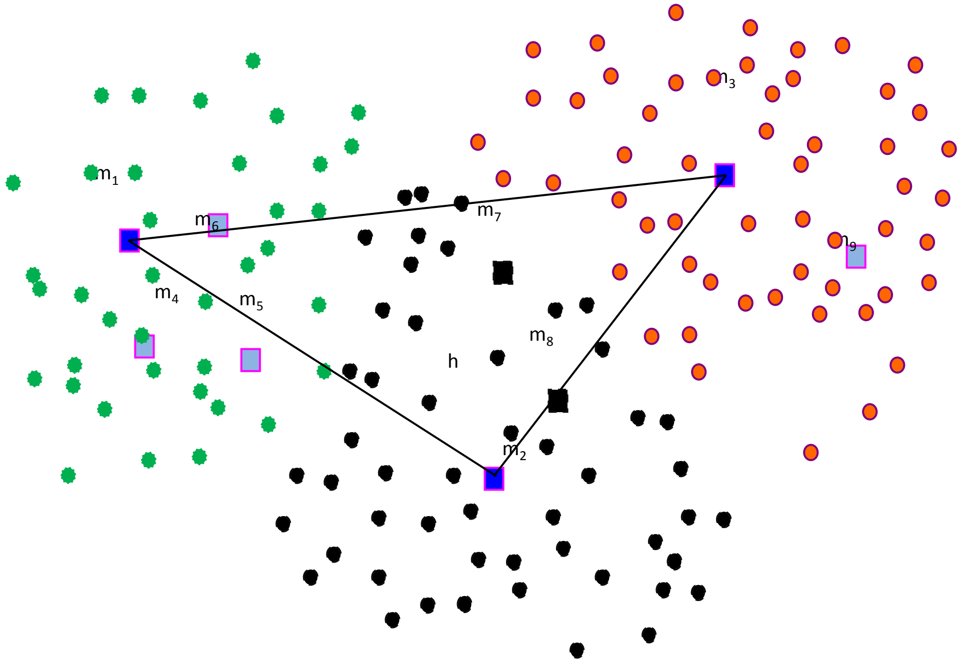
min

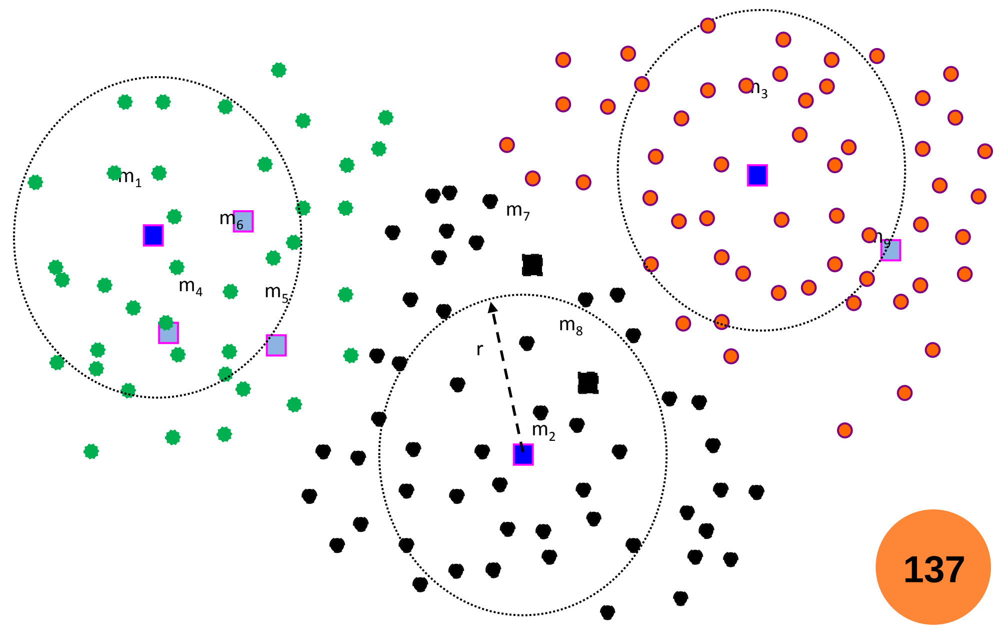
$$\sum_{m\in V_o} O_m y_m + \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij} x_{ij}^k + F_p \sum_{k\in K_p}\sum_{m\in V_o}\sum_{j\in V_p} x_{mj}^k + F_d \sum_{k\in K_d}\sum_{m\in V_o}\sum_{j\in V_d} x_{mj}^k +$$

$$2\sum_{m\in V_o}\sum_{k}\sum_{\eta=1}^{n} y_m x_{\varphi_{\eta-1}\varphi_\eta}^k (c_{m\varphi_\eta} \times \Pr(\delta_{\varphi_{\eta-1}}^{pu} < Q_{pu}^k) \times \Pr(\delta_{\varphi_\eta}^{pu} > Q_{pu}^k) \times \Pr(\delta_{\varphi_{n-\eta}}^{pu} < Q_{pu}^k)) +$$

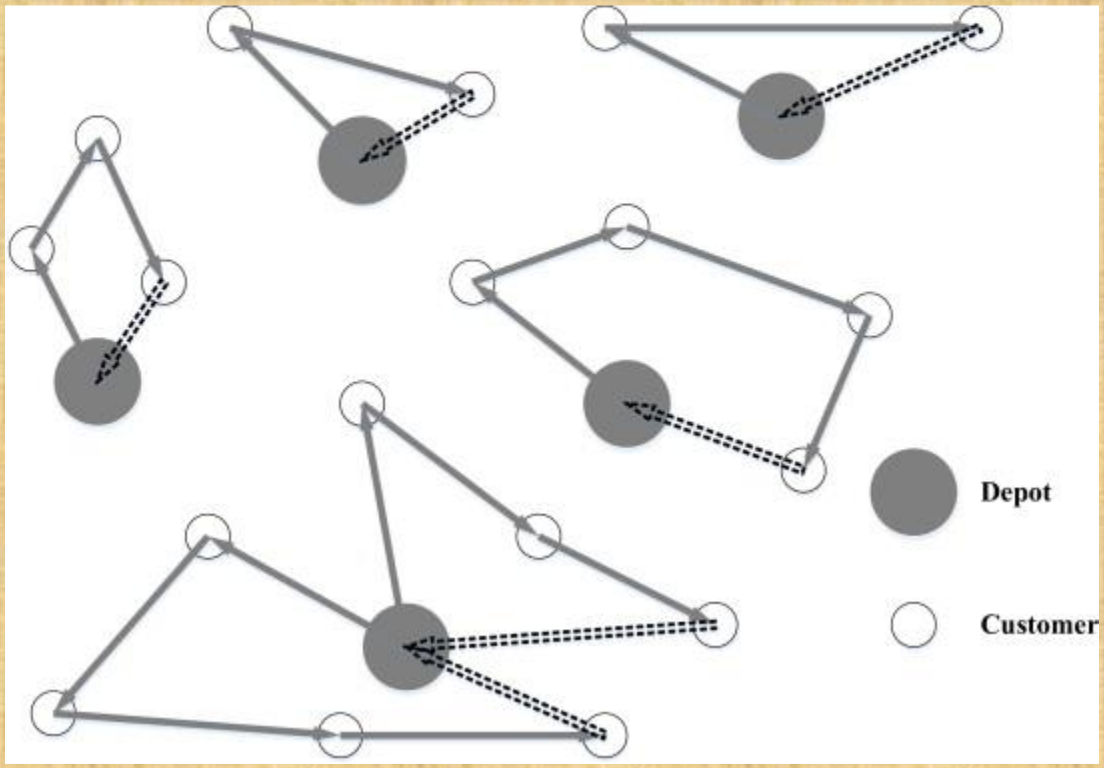
$$2\sum_{m\in V_o}\sum_{k}\sum_{\eta=1}^{n} y_m x_{\varphi_{\eta-1}\varphi_\eta}^k (c_{m\varphi_\eta} \times \Pr(\delta_{\varphi_{\eta-1}}^{du} < Q_{du}^k) \times \Pr(\delta_{\varphi_\eta}^{du} > Q_{du}^k) \times \Pr(\delta_{\varphi_{n-\eta}}^{du} < Q_{du}^k)) +$$

$$\sum_{e=0}^{\infty}\sum_{u} \varepsilon_u (e(\Pr((\sum_{m}\sum_{i} z_{im} d_{iu} - W_{pu}^m) = e))) + \sum_{e=0}^{\infty}\sum_{u} \varepsilon_u (e(\Pr((\sum_{m}\sum_{i} z_{im} d_{iu} - W_{du}^m) = e)))$$


m_1
m_6
m_7
m_3
m_9
m_4
m_5
m_8
h
m_2


m_1
m_6
m_4
m_5
m_7
m_3
m_9
m_8
r
m_2

Open Location Routing Problem



Questions or Comments?

- https://www.youtube.com/watch?v=mjbCUr2uFMc
- https://www.youtube.com/watch?v=Rm_V75Jzapg
- https://www.youtube.com/watch?v=cwbHVhZYu2k
- https://www.youtube.com/results?search_query=%E7%8D%A8%E7%AB%8B%E7%89%B9%E6%B4%BE%E5%93%A1+%E7%89%A9%E6%B5%81