Service Intelligence Week 10.

[Service Optimization: Operational Decision Making with Computational Intelligence]

Chiehyeon Lim

2022. 10. 31

Today's Topic in the Framework of This Course

consumer behaviors (user behaviors)

service delivery and experience

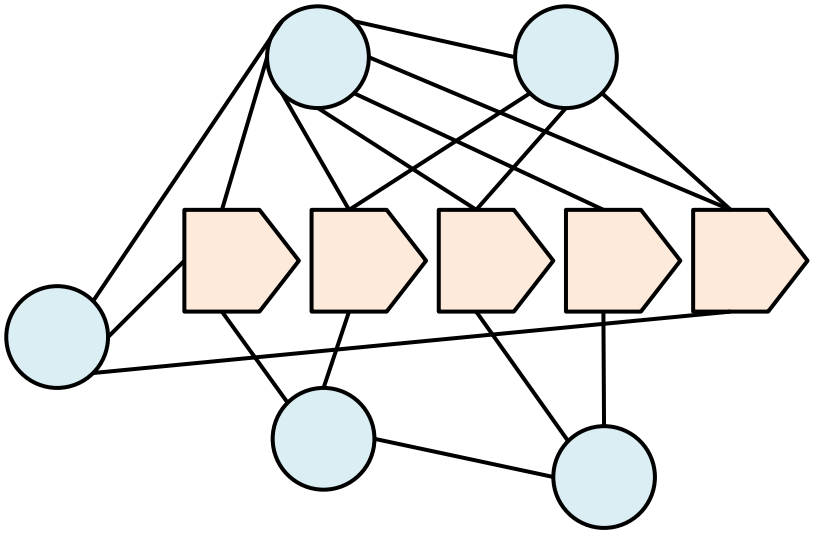
quality

value cocreation

technology and self-service

relationship

innovation



design and development

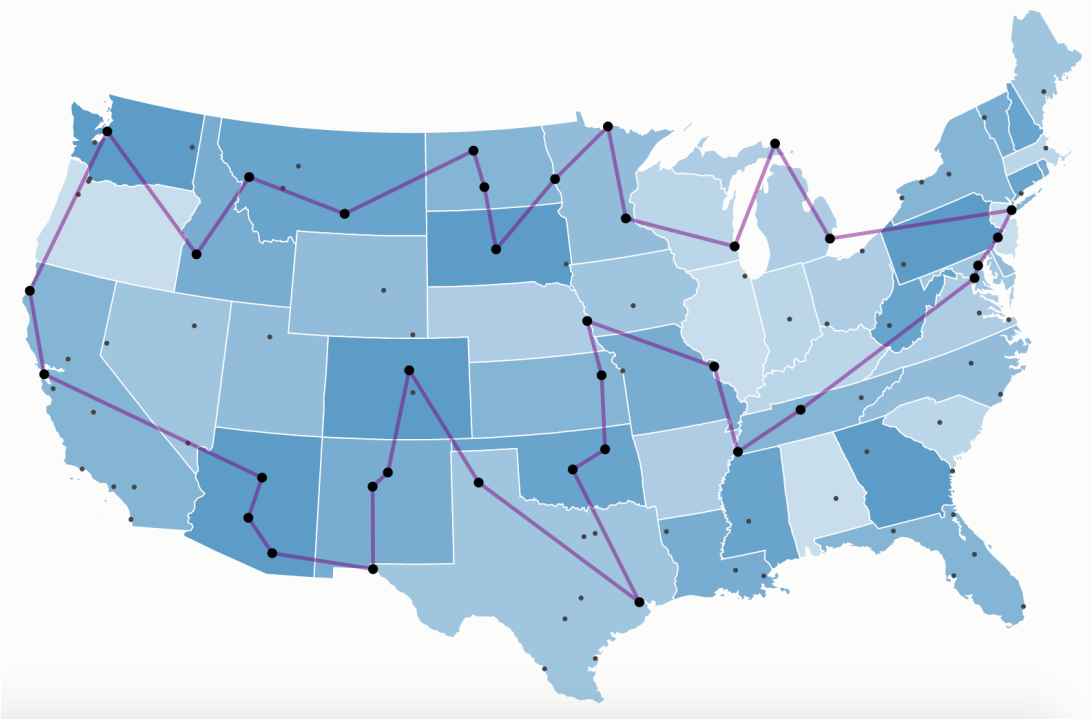
productivity

knowledge and employee management

cost and pricing

service operations and supply chain management

Network of Service Delivery: An Example



A TSP Model

$$Let\ x_{ij} = \begin{cases} 1 & if\ the\ tour\ passes\ i\ to\ j \\ 0 & if\ not \end{cases}$$

$$Minimize \quad Z = \sum_{i} \sum_{j \neq i} c_{ij} x_{ij}$$

$$subject\ \ to$$

$$\sum_{j=1} x_{ij} = 1, \quad \text{for all i}$$

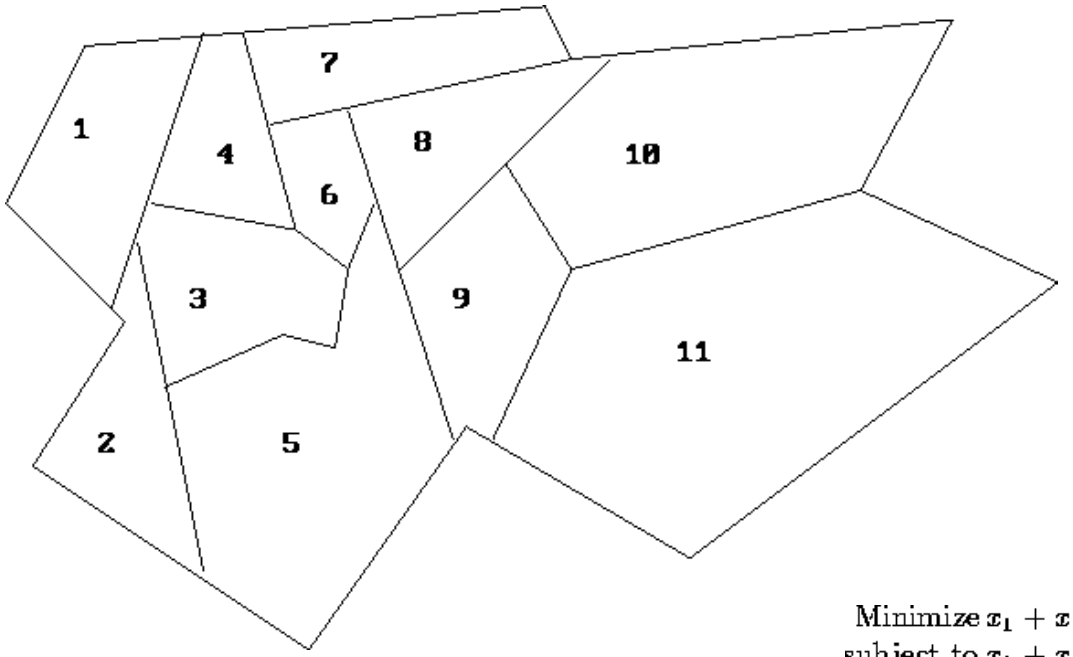
$$\sum_{i=1}^{n} x_{ij} = 1, \quad \text{for all j}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \forall S \subset \{1,2,\dots,n\}$$

$$x_{ij} = \ binary,\ for\ \ all\ i\ \ and\ \ j$$

TSP Problem in UNIST

Other Service-related Optimization Problems: Set Covering



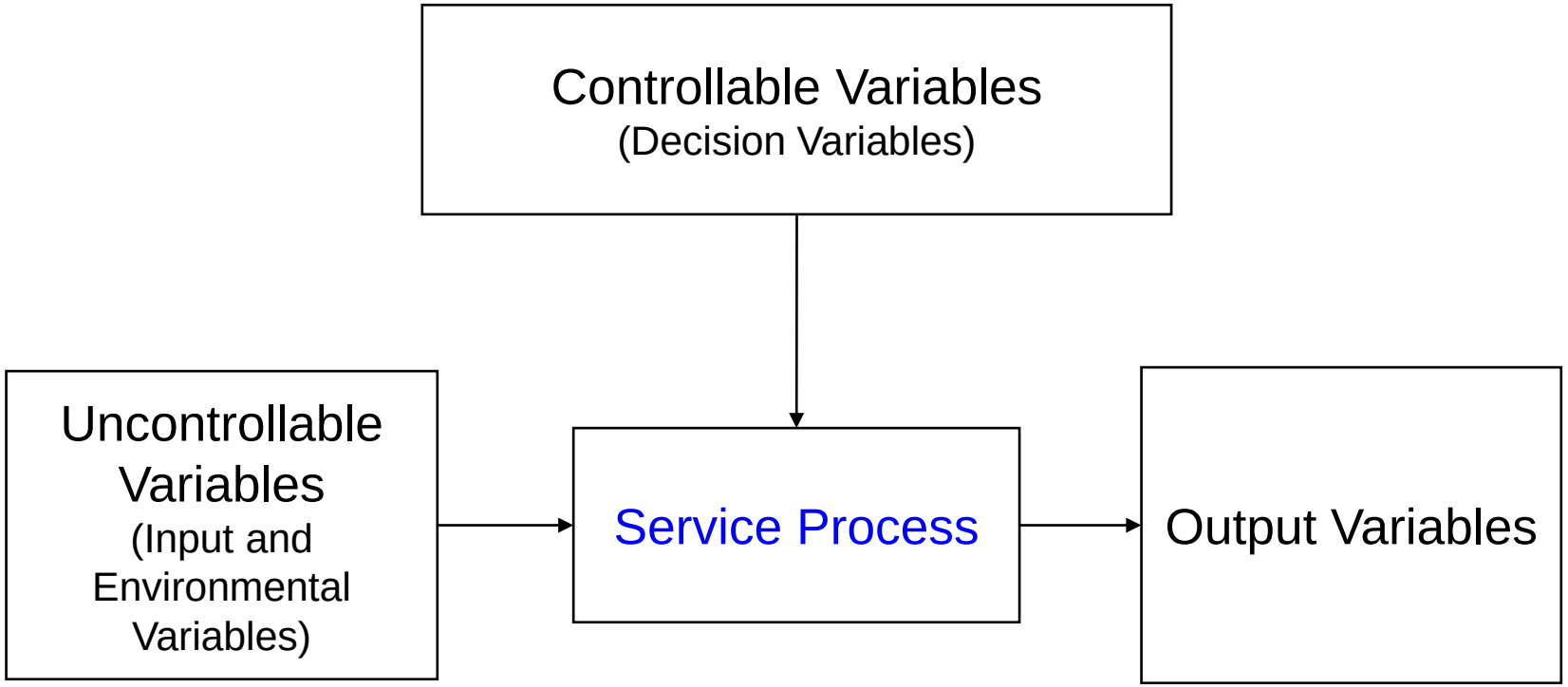
$$
\begin{aligned}
\text{Minimize } & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \\
\text{subject to } & x_1 + x_2 + x_3 + x_4 \geq 1 \\
& x_1 + x_2 + x_3 + x_5 \geq 1 \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\
& x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \\
& x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \\
& x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \\
& x_4 + x_6 + x_7 + x_8 \geq 1 \\
& x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \\
& x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_9 + x_{10} + x_{11} \geq 1 \\
& x_j \in \{0,1\} \quad j = 1, \ldots, 11
\end{aligned}
$$

Source: http://mat.gsia.cmu.edu/classes/integer/node8.html

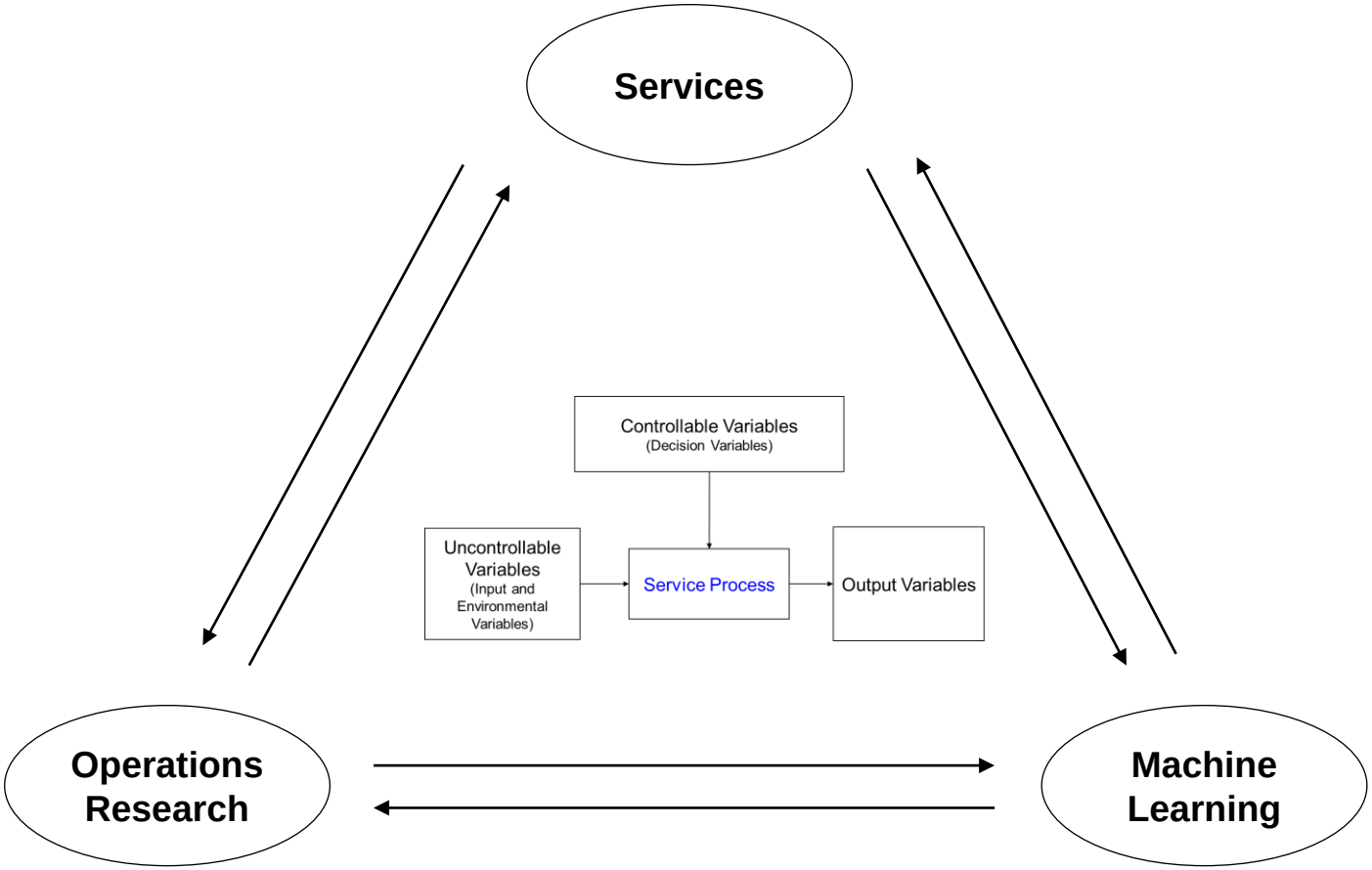
Service-related Optimization Problems

- Traveling Salesman Problem
- Vehicle Routing Problem
- Set Covering Problem
- Knapsack Problem
- Bin Packing Problem
- Facility Location Problem
- Scheduling Problem
- Etc.

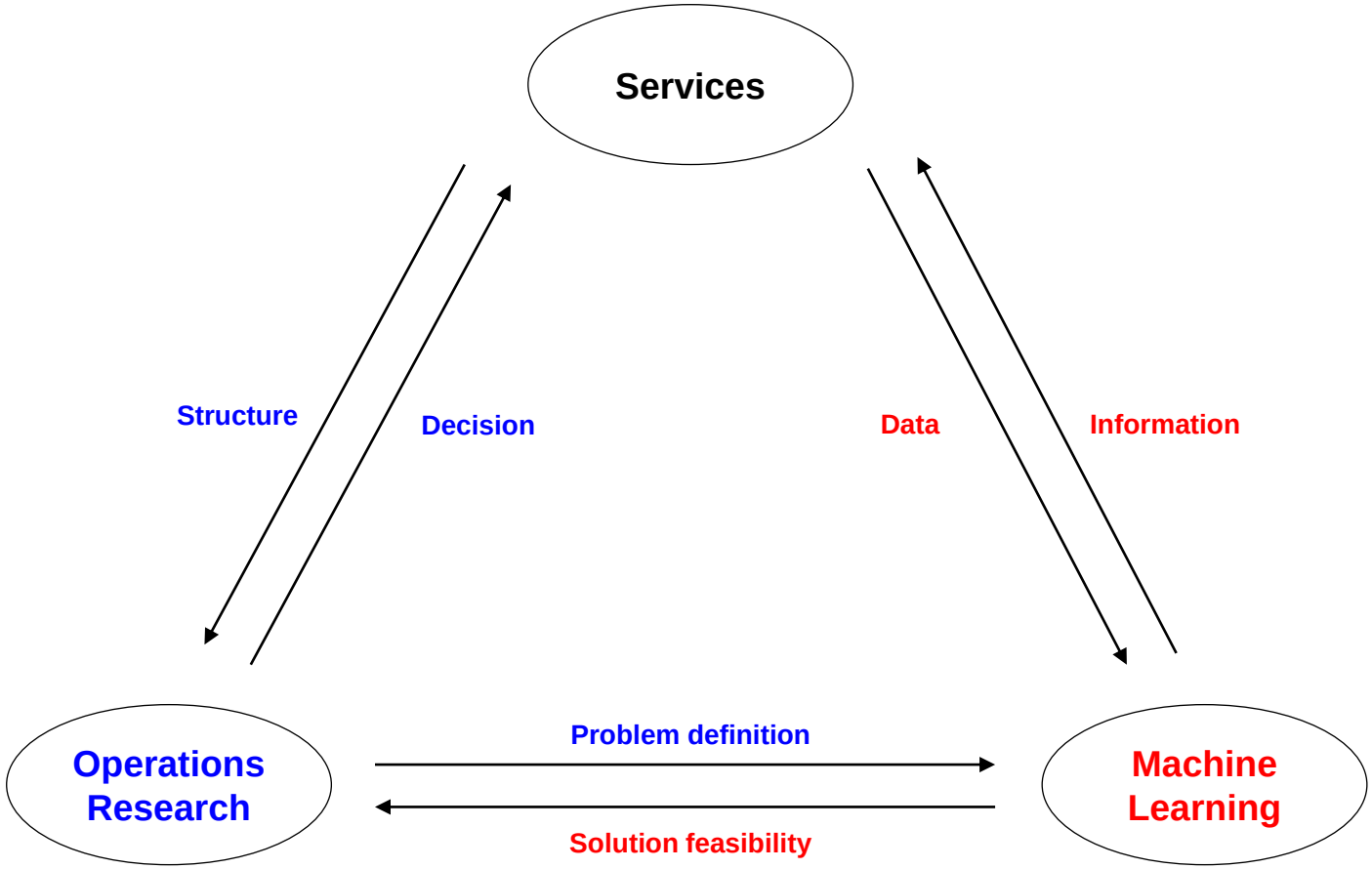
Challenges in Service Optimization



Challenges in Service Optimization



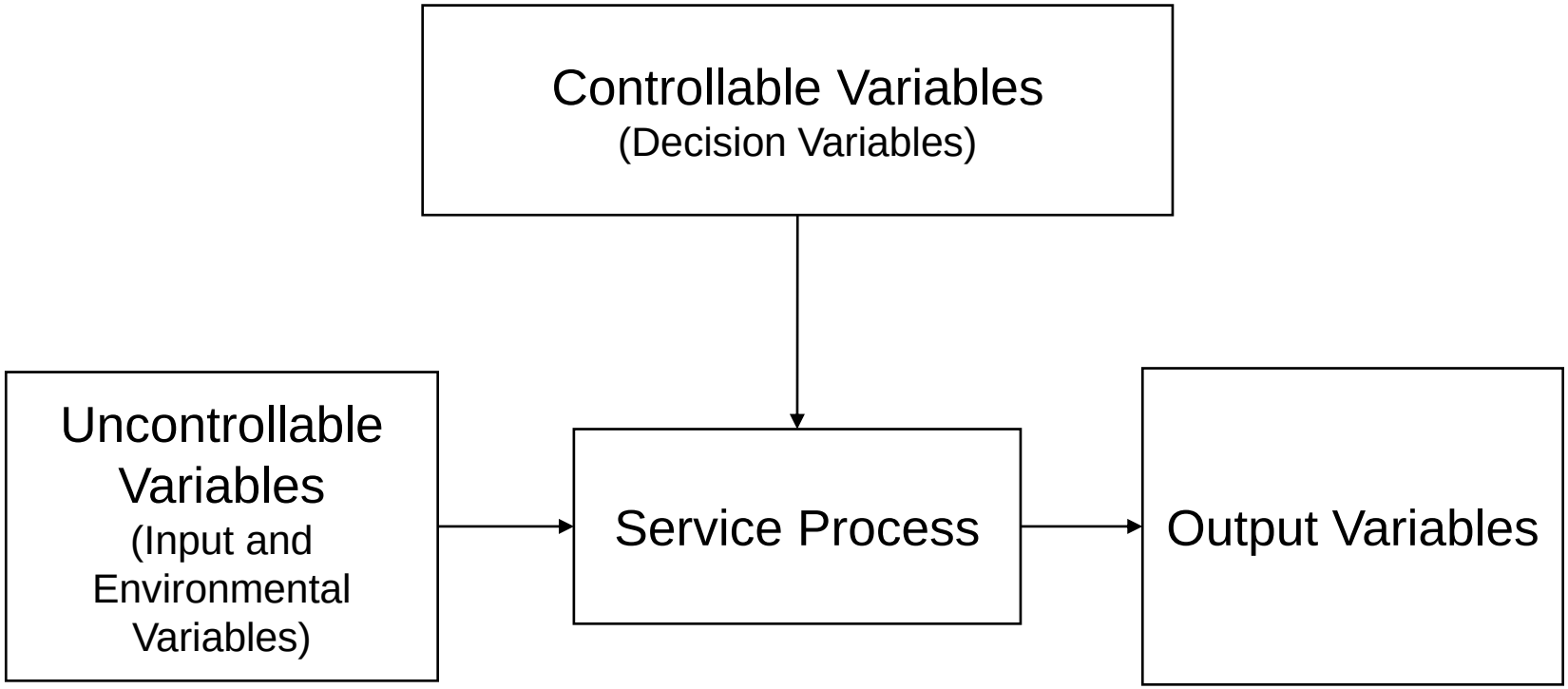
Challenges in Service Optimization: OR-based Modeling Approach



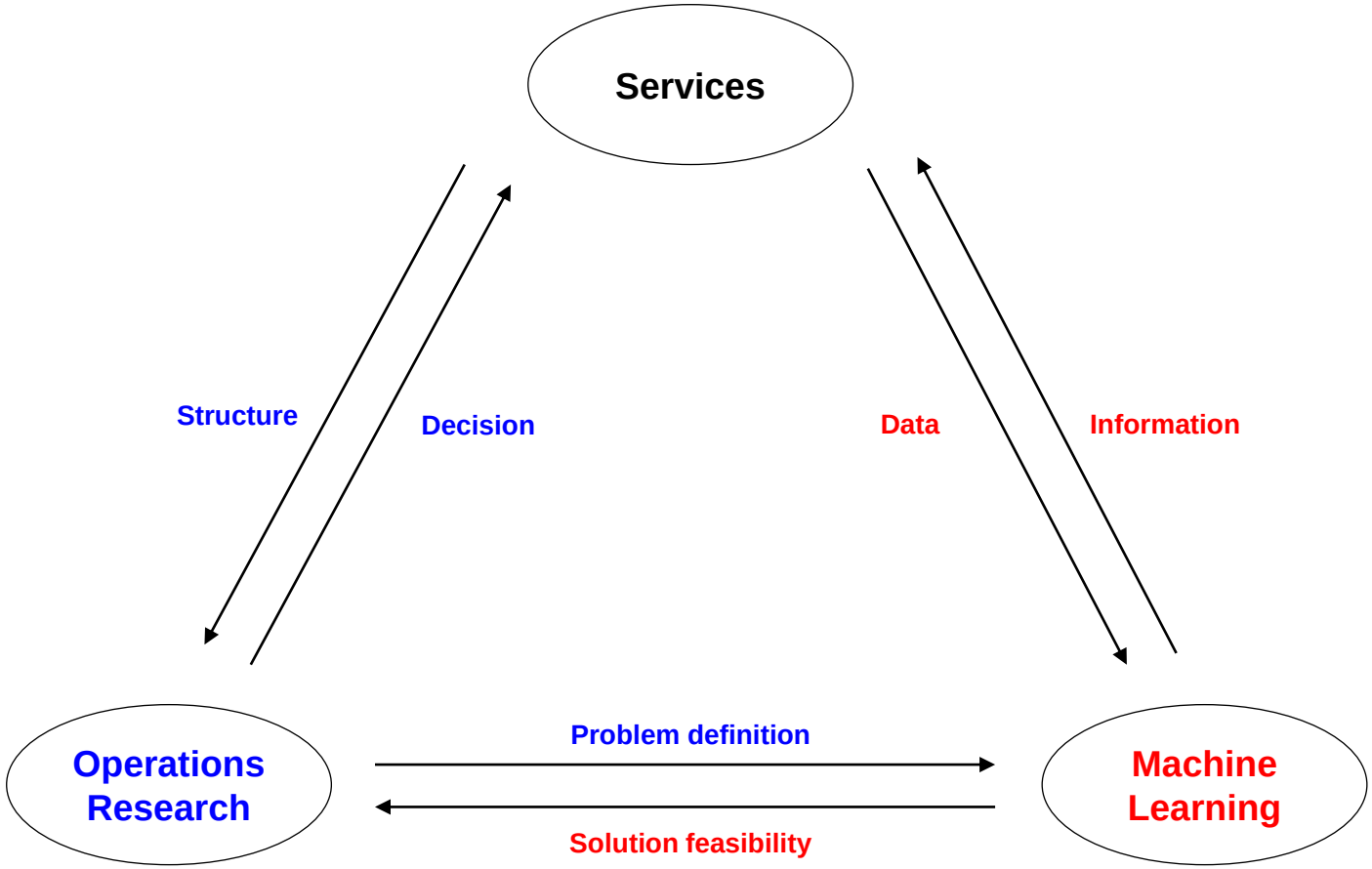
Challenges in Service Optimization: "Deterministic" Approach

"Model building" "Search optimal setting for **x**"

$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon \quad \rightarrow \quad y(\mathbf{x}) \quad \rightarrow \quad \underset{\mathbf{x}\in\Omega}{\text{optimize}}\ y(\mathbf{x})$$

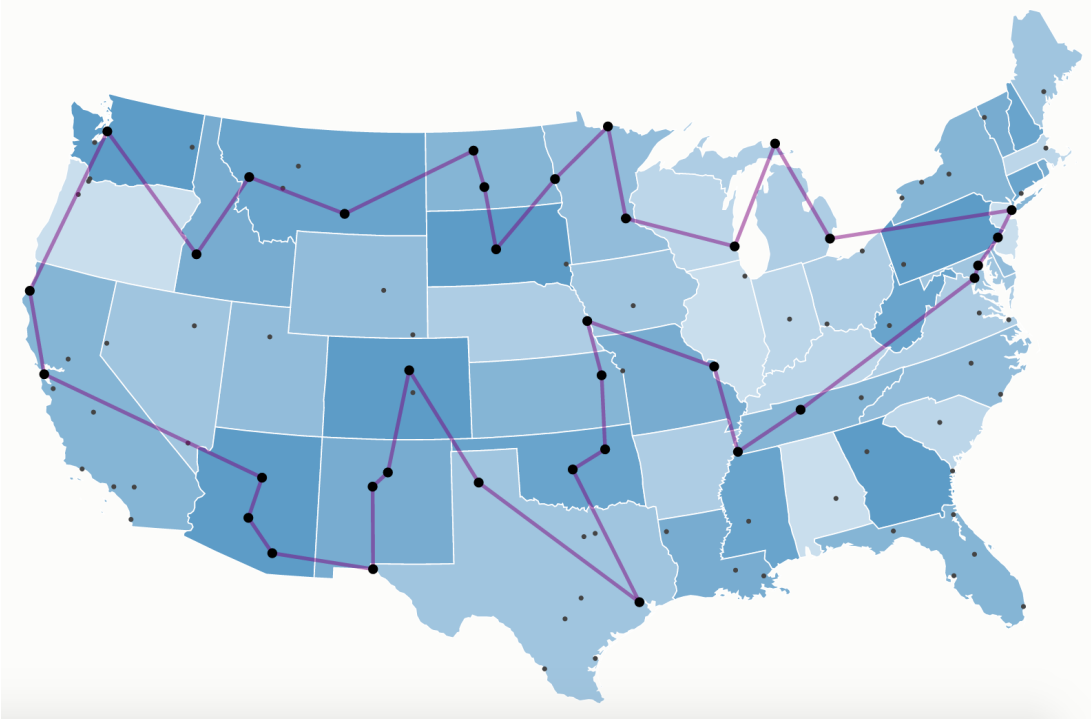


Challenges in Service Optimization: OR-based Modeling Approach

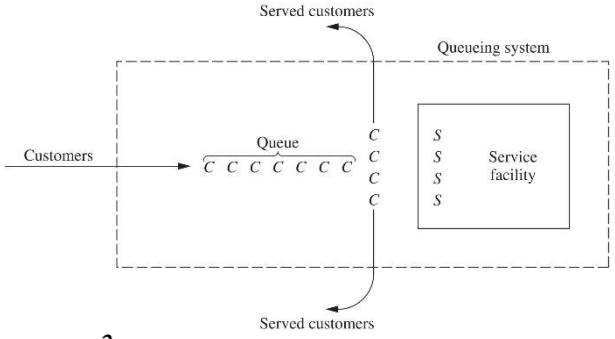


Solution Feasibility of TSP

Solution Feasibility of TSP



Solution Feasibility of Queuing



$$\rho = \frac{\lambda}{\mu}$$

$$P_0 = 1 - \rho$$

$$P_n = (1-\rho)\rho^n$$

$$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}$$

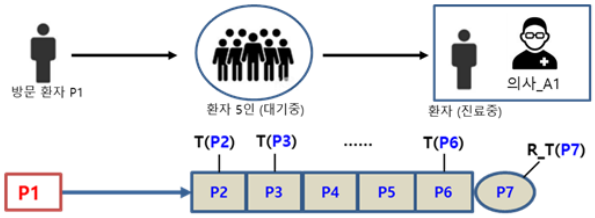
$$L_q = \frac{\lambda^2}{\mu(\mu-\lambda)} = \lambda W_q$$

$$W_q = \frac{\lambda}{\mu(\mu-\lambda)}$$

$$W = \frac{1}{\mu-\lambda} = W_q + 1/\mu$$

ML-based Data-driven Approach for Queuing

Input Data
(Pre-processed)



대기시간 예측용 데이터		
컬럼 명	**설명**	**변수 출처**
RCPN_NO	데이터 key	공통
PID	환자 ID	공통
DEPT_NM	진료과 명	공통
MDCR_DR_ID	의사 ID	공통
RCPN_APNT_DVNM	예약 여부	공통
RCPN_APNT_KIND_NM	접수경로	공통
MCCH_FREE_RESN_NM	특이사항	공통
FVNR_DVNM	환자종류	환자 접수 데이터
MDCR_DATE	날짜	공통
weekday	요일	파생
Time_Call	방문호출 시점 (진료시작시점)	호출 이벤트 데이터 가공
Time_next_patient_Call	다음 환자의 방문호출 시점 (진료종료시점)	호출 이벤트 데이터 가공
Time_resister	전산상 '대기' 등록 시점 (대기시작시점)	호출 이벤트 데이터 가공
Doctor_time	진료시간	파생
Waiting_time	대기시간	파생

Model

$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon$$

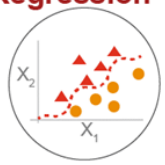
neural networks

Nonlinearity consideration

Decision trees

Rule-based explanation (e.g., RF model)

Regression

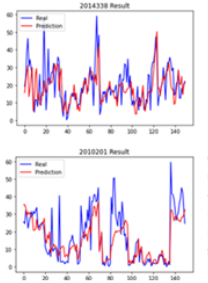


Linear coefficient calculation (LR with penalty)

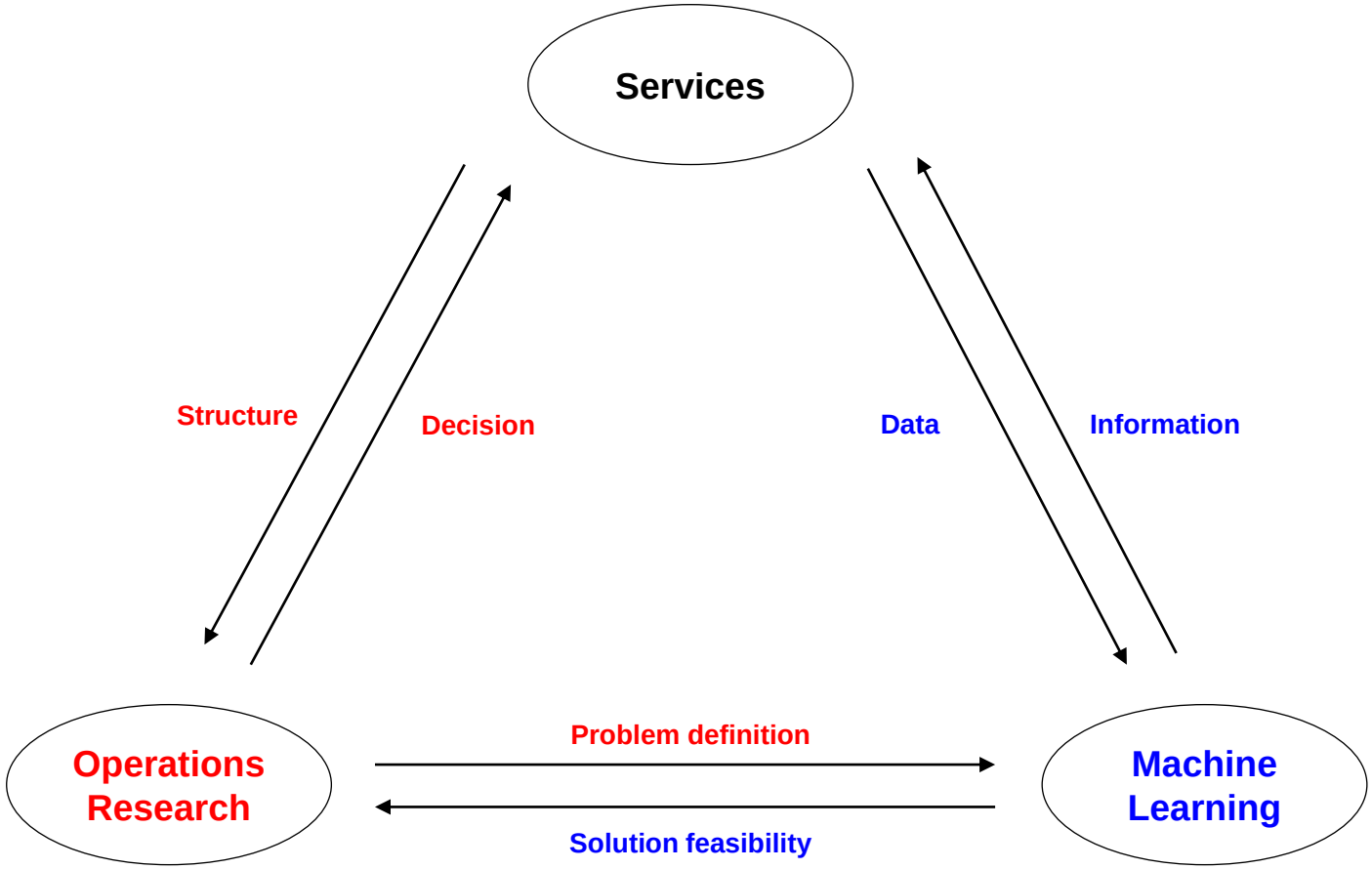
Result

$$\hat{y}(\mathbf{x})$$

Number of waiting patients



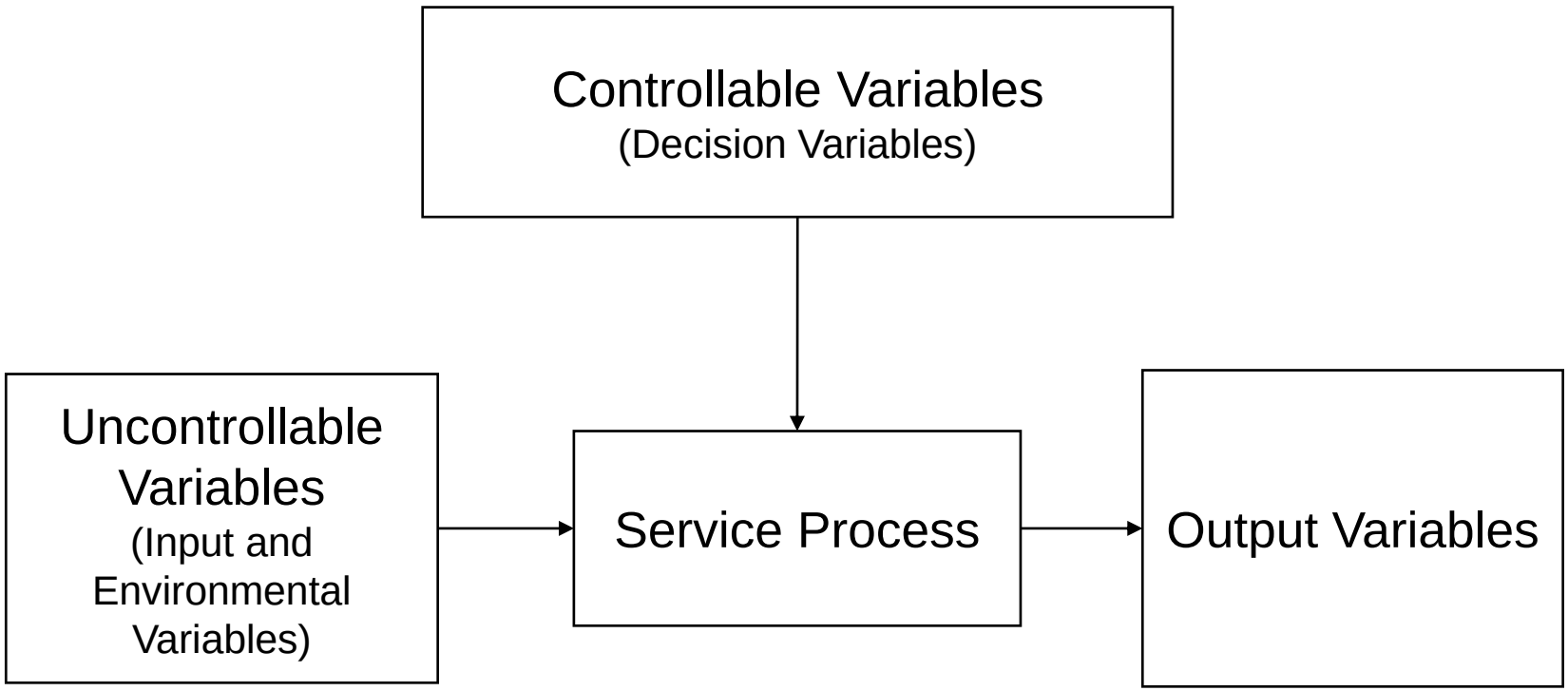
Challenges in Service Optimization: ML-based Data-driven Approach



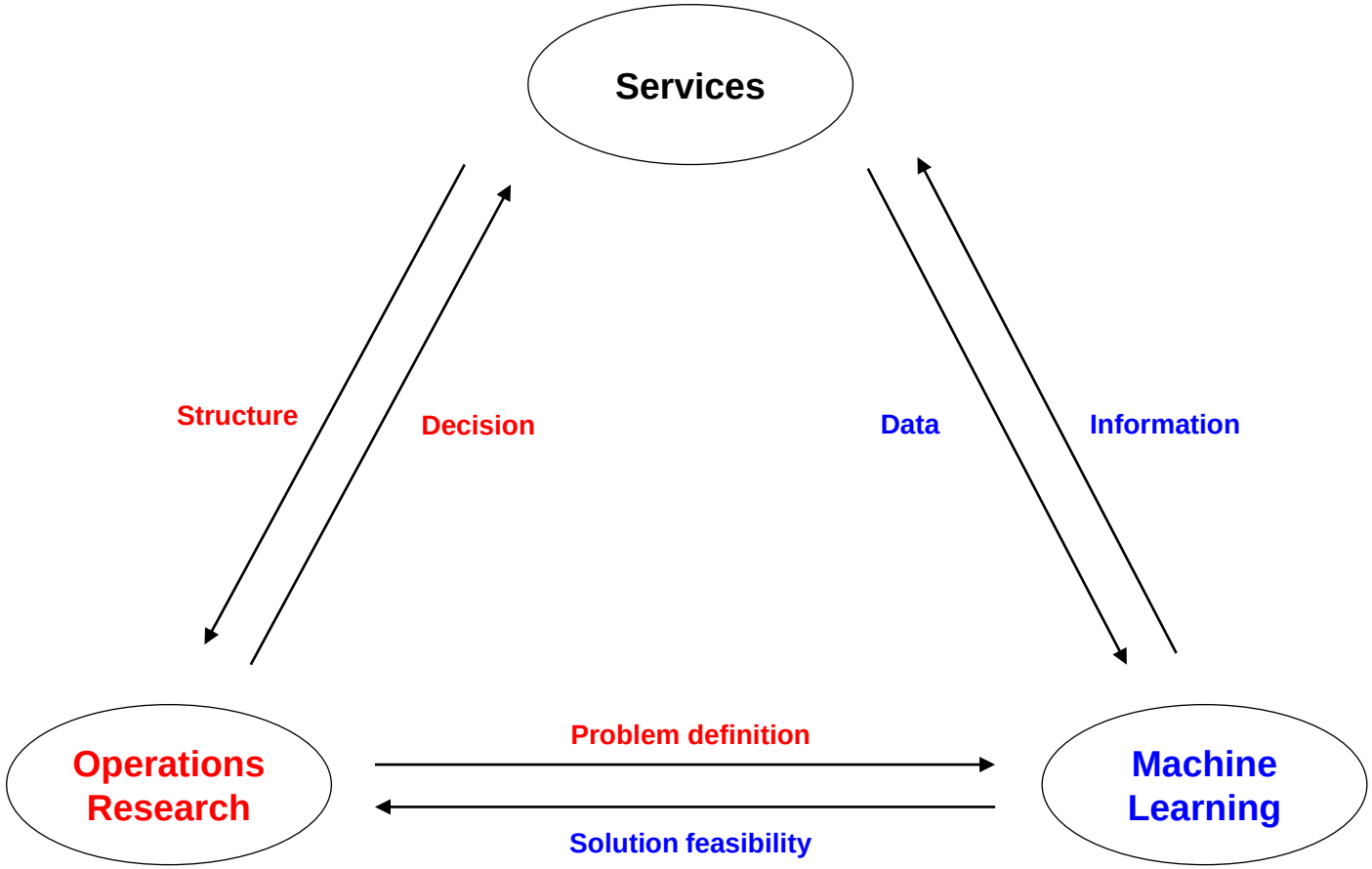
Challenges in Service Optimization: "Stochastic" Approach

"Model building" "Search optimal setting for **x**"

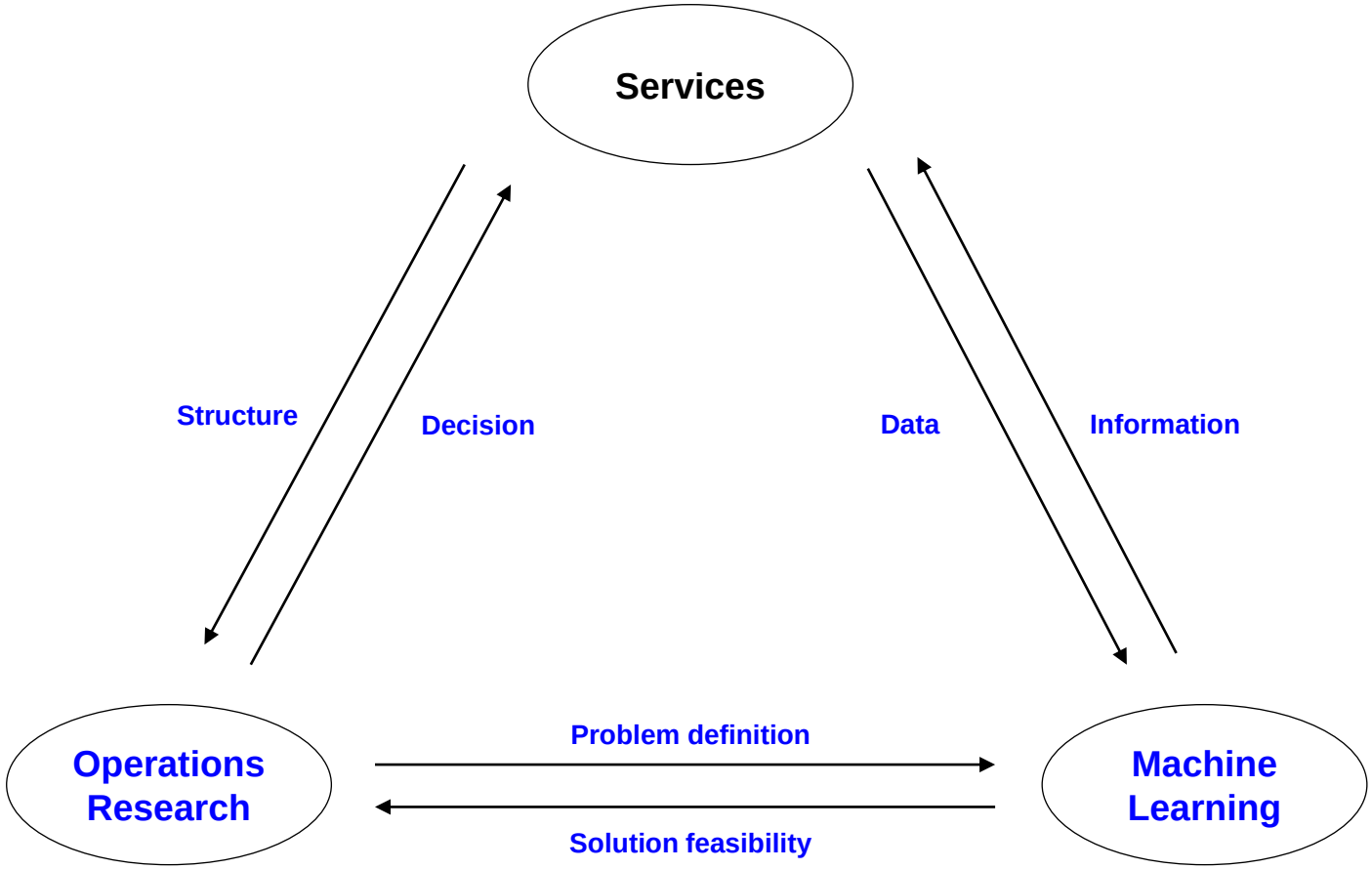
$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon \rightarrow \hat{y}(\mathbf{x}) \rightarrow \underset{\mathbf{x}\in\Omega}{\text{optimize}}\ \hat{y}(\mathbf{x})$$



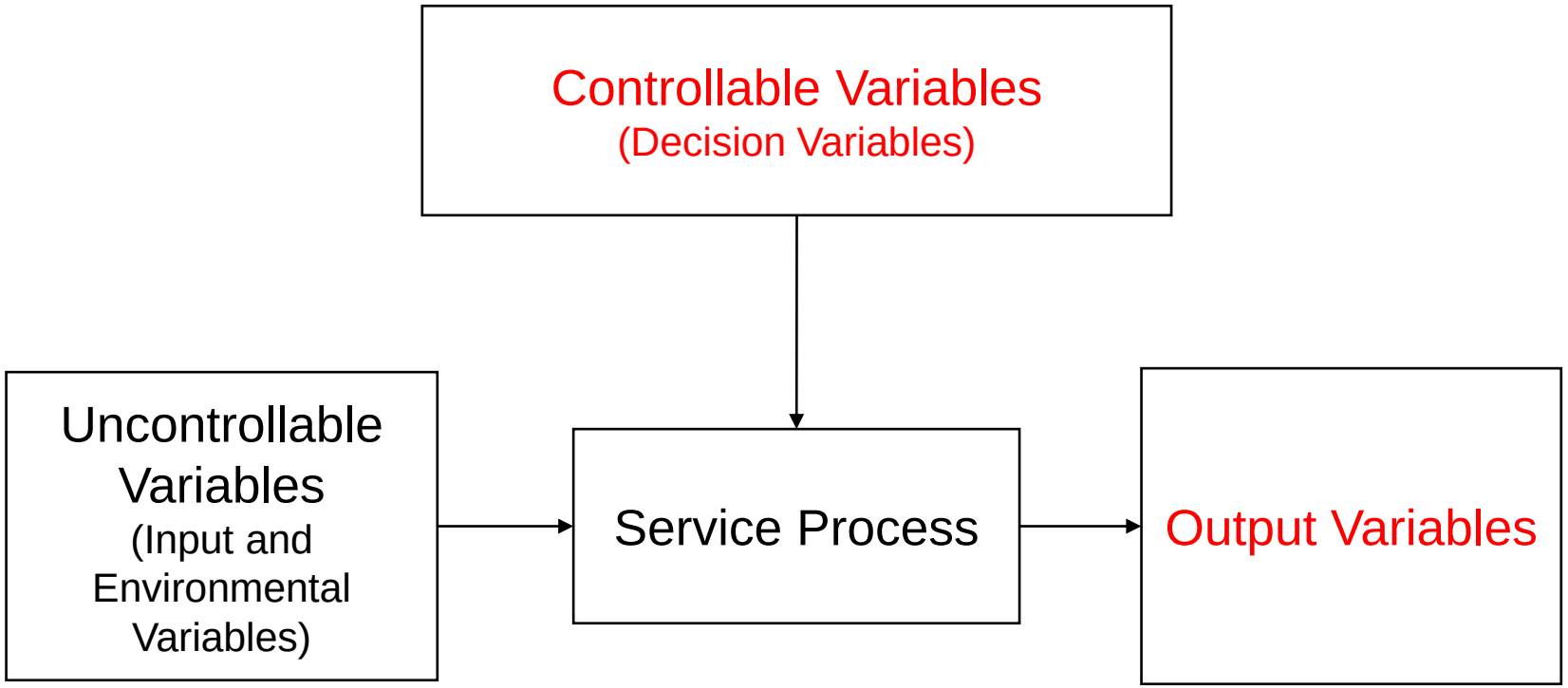
Challenges in Service Optimization: ML-based Data-driven Approach



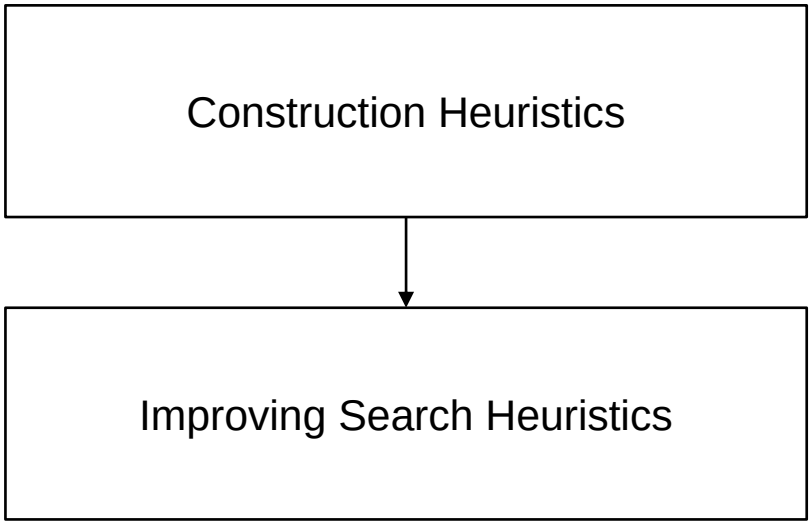
Service Optimization: Both Approaches are Needed



Challenges in Service Optimization: Optimization Algorithms



Heuristics

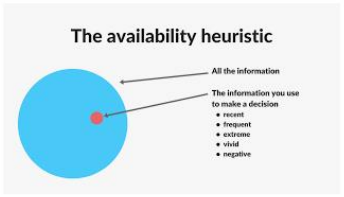


Heuristics

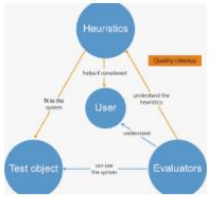
psychology | algorithm | problem solving | availability | representativeness | decision making | design | usability | evaluation | bias | nielsen | affect

Heuristics and Cognitive Biases
verywellmind.com

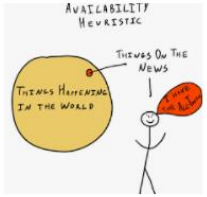
Availability Heuristic Affecting Your ...
verywellmind.com



The availability heuristic: Why your ...
kenthendricks.com



Quality model of heuristic evalu...
researchgate.net

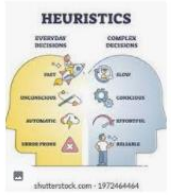


Availability Heuristic - Biases ...
thedecisionlab.com

What Is A Heuristic And Why Heuristics ...
fourweekmba.com


Heuristic Evaluation & Analysis in UX ...
xd.adobe.com

Applying Heuristic Evaluation ...
designorate.com

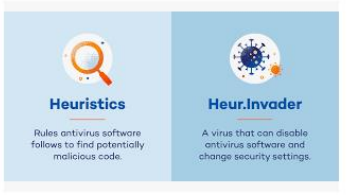


Heuristic Images, Stock ...
shutterstock.com

Heuristics - Overview, How It Works ...
corporatefinanceinstitute.com

The availability heuristic (and bia...
sketchplanations.com

Heuristics Definition | DeepAI
deepai.org



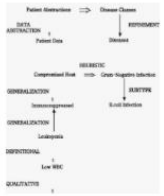
What Is a Heuristic Virus and How to ...
pandasecurity.com

5 Common Mental Errors That S
jamesclear.com

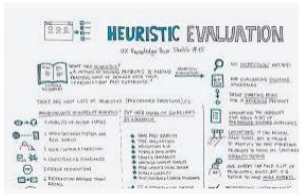

Reducing Consumer Decision Friction ...
paleofoundation.com


Using Heuristic Problem-Solving Methods ...
slidemodel.com

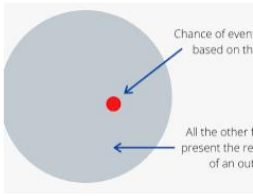
Create a heuristic evaluation and ...
fiverr.com


Heuristic Classification
aistudy.co.kr


What Is Heuristic Evaluation? Simple ...
adamfard.com



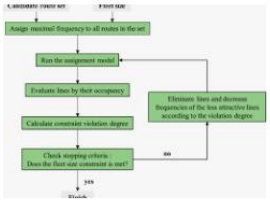
Heuristic Evaluation. UX Knowledge Base ...
uxknowledgebase.com

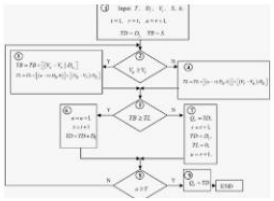


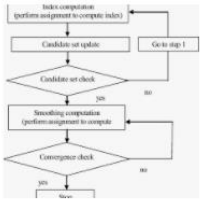
Availability Heuristic (Definition and ...
boycewire.com

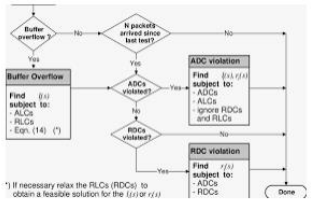
Heuristics

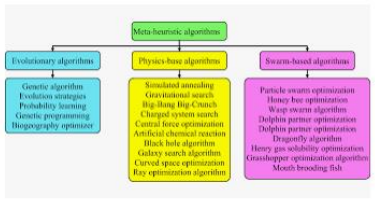
flowchart | singhal | diagram | problem | mystery | simple | classification | ppt | neh | tree | architecture | clustering | scheduling

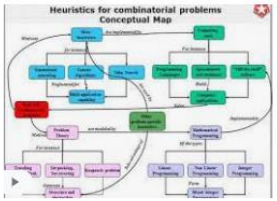

Heuristic algorithm scheme. | Download ...
researchgate.net

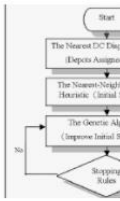

CB heuristic algorithm. | Download ...
researchgate.net

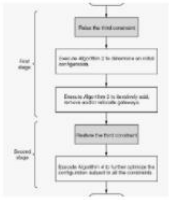

rule based heuristic algorithm...
researchgate.net

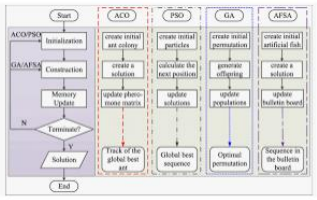

Outline of the Heuristic Algorithm ...
researchgate.net


A novel nature-inspired meta-heuristic ...
link.springer.com

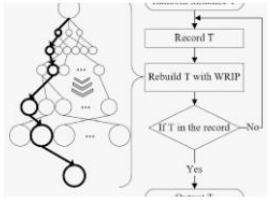

Heuristic Algorithms introduction - YouTube
youtube.com


The hybrid heuristic
researchgate.net

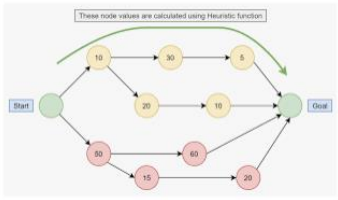

Heuristic Algorithm - an ...
sciencedirect.com


framework of meta-heuristic algorithms ...
researchgate.net

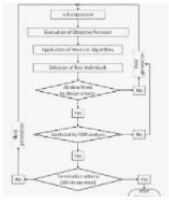

your algorithms from your heuristics ...
dataphilos.com

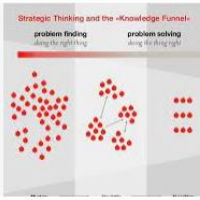


A Heuristic Algorithm Based on Tour ...
semanticscholar.org

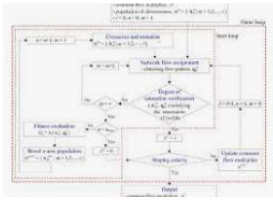


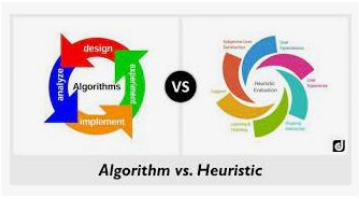
Greedy Vs. Heuristic Algorithm ...
baeldung.com


optimization of heuristic ...
researchgate.net

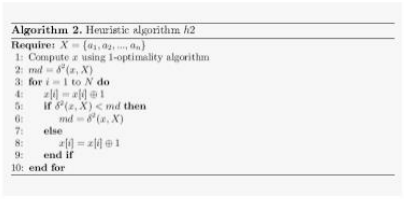


Mystery Heuristic Algorithm ...
in.pinterest.com

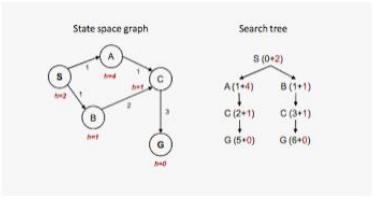

Procedure of the heuristic algorithm ...
researchgate.net



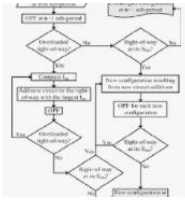
Heuristics Definition | DeepAI
deepai.org



Heuristic Algorithms for 2-Optimality ...
link.springer.com



heuristic function H is not monotonic ...
stackoverflow.com

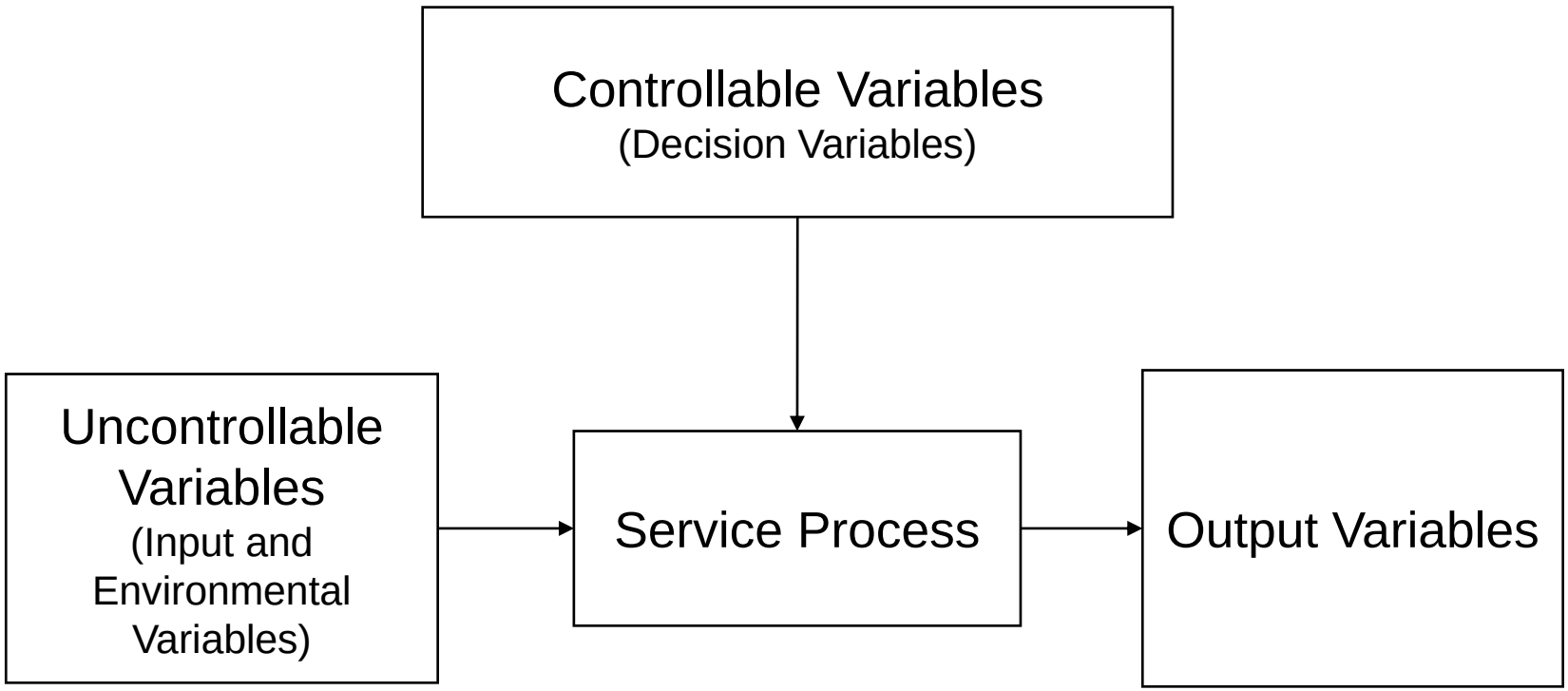

Heuristic algorithm embedd...
researchgate.net

Sookshum: Heuristic Algorithm
sookshum.blogspot.com

Optimization Algorithms for "Deterministic" Problems

"Model building" "Search optimal setting for **x**"

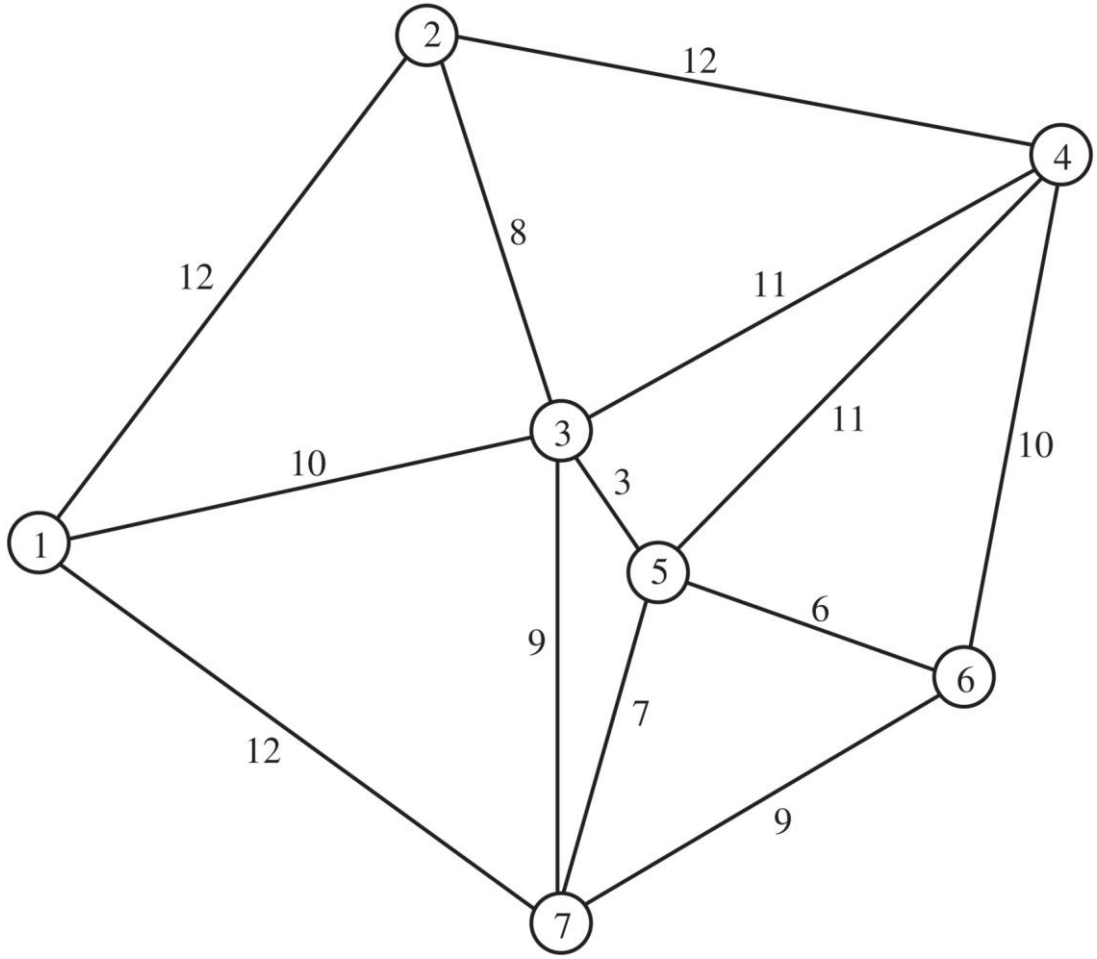
$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon \quad \rightarrow \quad y(\mathbf{x}) \quad \rightarrow \quad \underset{\mathbf{x} \in \Omega}{\text{optimize}}\ y(\mathbf{x})$$



TSP Problem in UNIST

TSP Problem in UNIST

TSP Problem in UNIST: A Text Book Format



Improving Search Heuristic Algorithm

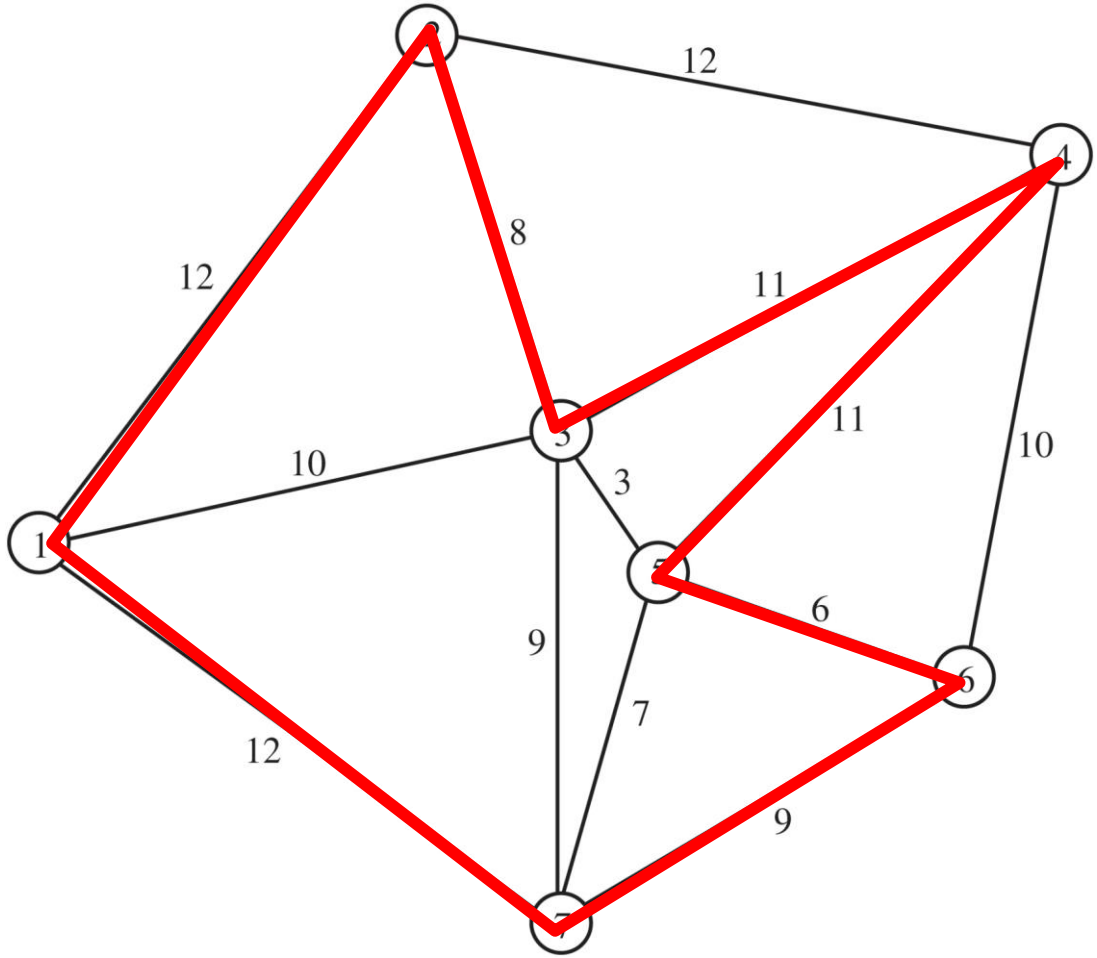
- **Step 0:** Choose any starting feasible solution x(0), and set solution index t = 0
- **Step 1:** If no move Δx in move set M is both improving and feasible at current solution x(t), stop. x(t) is a local optimum.
- **Step 2:** Choose some improving feasible move $\Delta x \in M$
- **Step 3:** Update x(t+1) = x(t) + Δx
- **Step 4:** T = t+1, return to Step 1

Some TSP Construction Heuristics

- Random Search
- Nearest Neighbor Heuristic
- Divide and Conquer
- Farthest Addition Heuristic
- Multi-fragment
- …

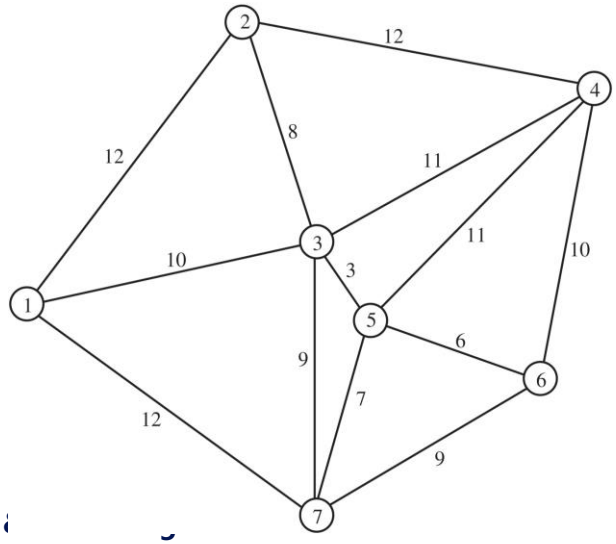
Source: R. L. Rardin, Optimization in Operations Research, Prentice Hall, 1998

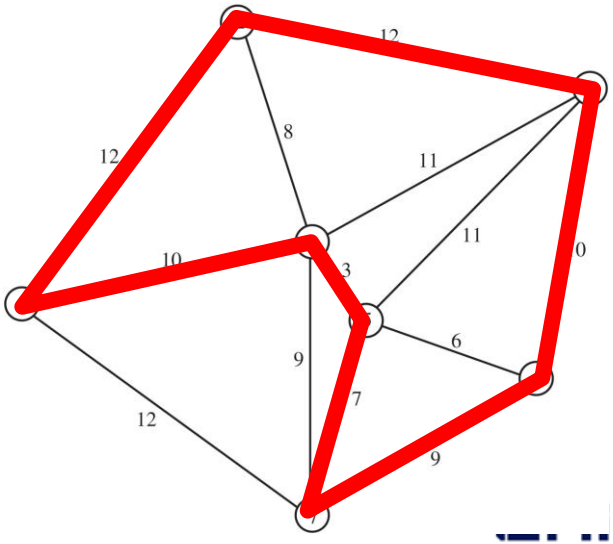
Random Search



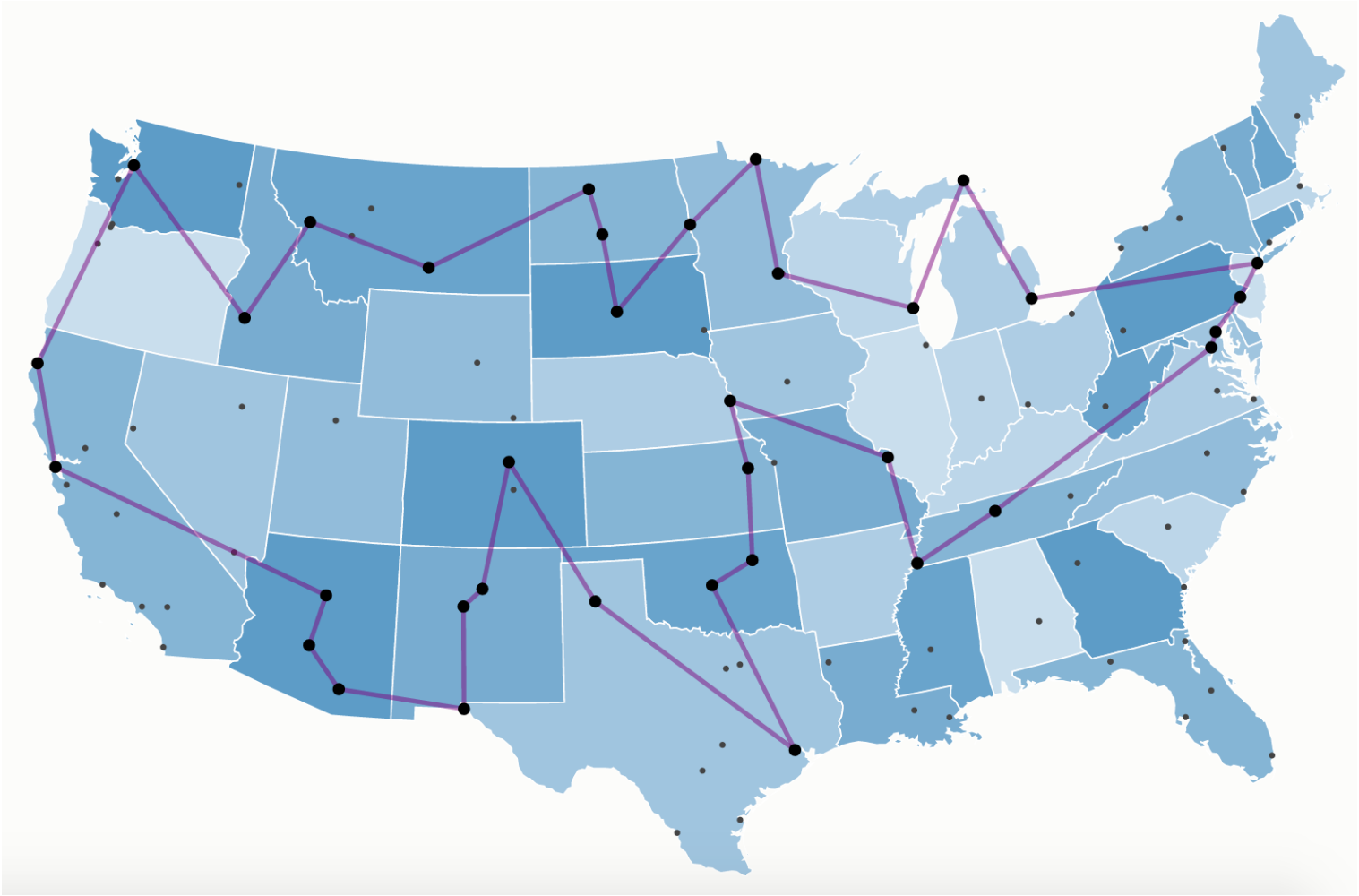
Nearest Neighbor (Greedy Constructive) Heuristic

- Select Greedily
  - Elect the next variable to fix and its value that does least damage to feasibility and most helps the objective function, based on what has already been fixed in the current partial solution
  - Pros: Simple & Fast
  - Cons: Looking only locally → may end up with poor objective value

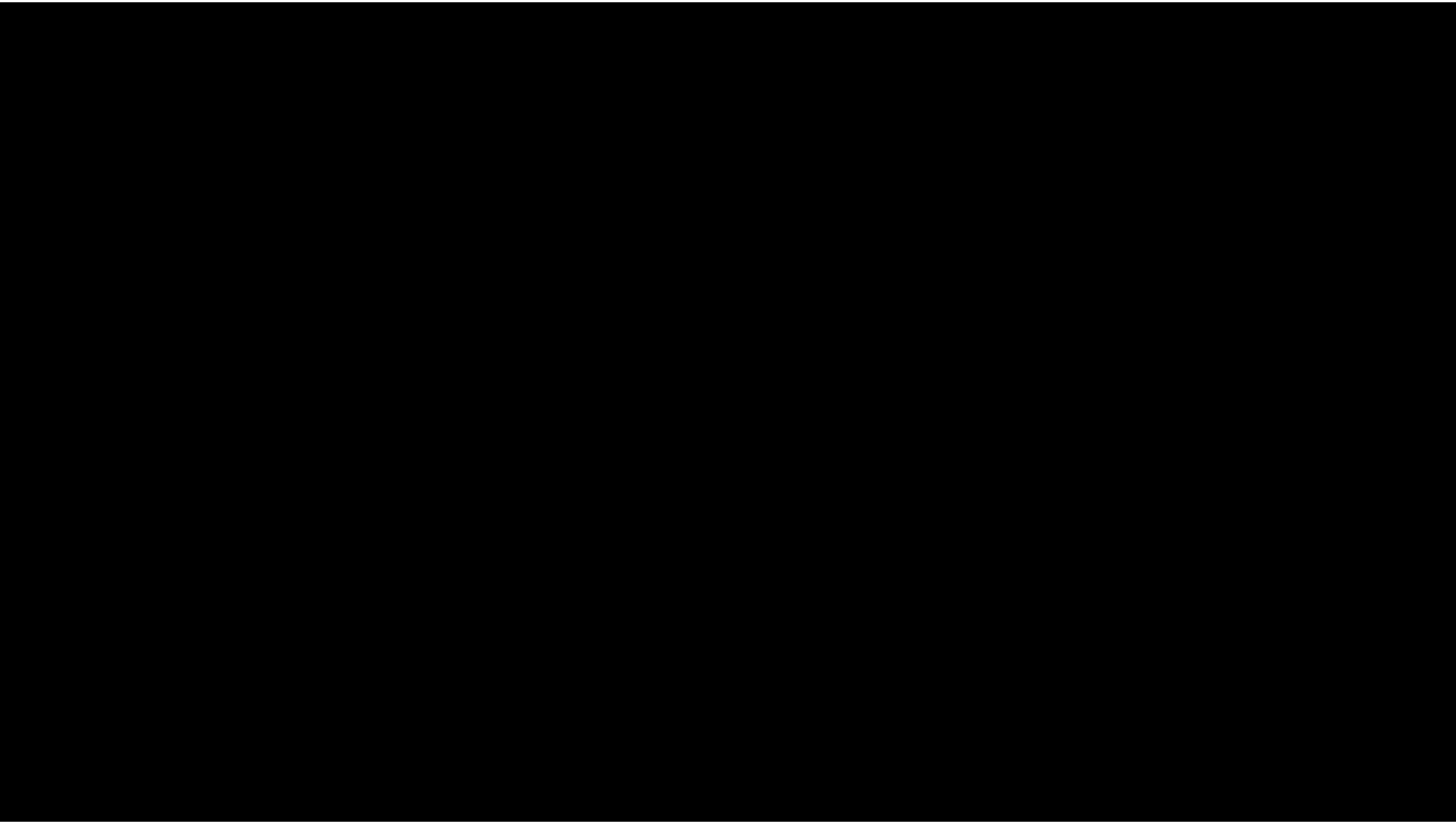




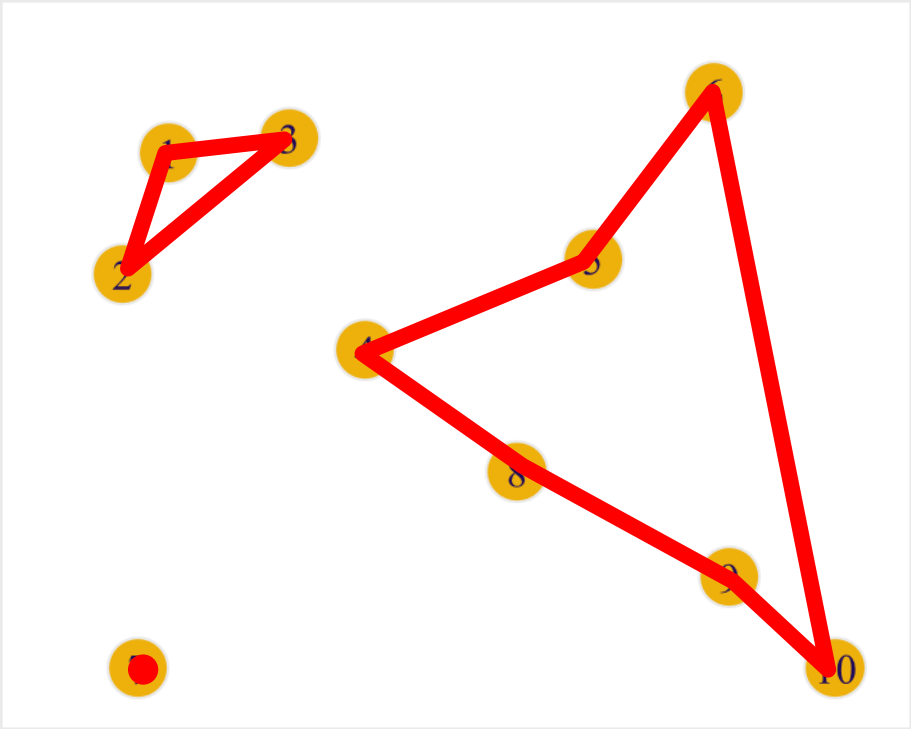
Greedy Constructive Heuristics: Real Example



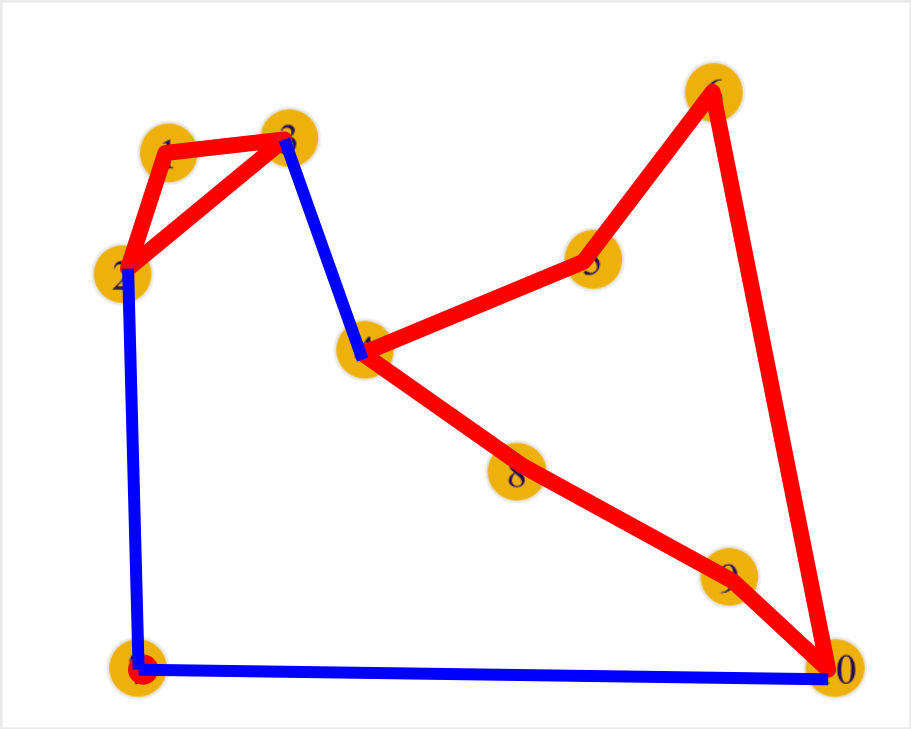
Greedy Constructive Heuristics: Real Example



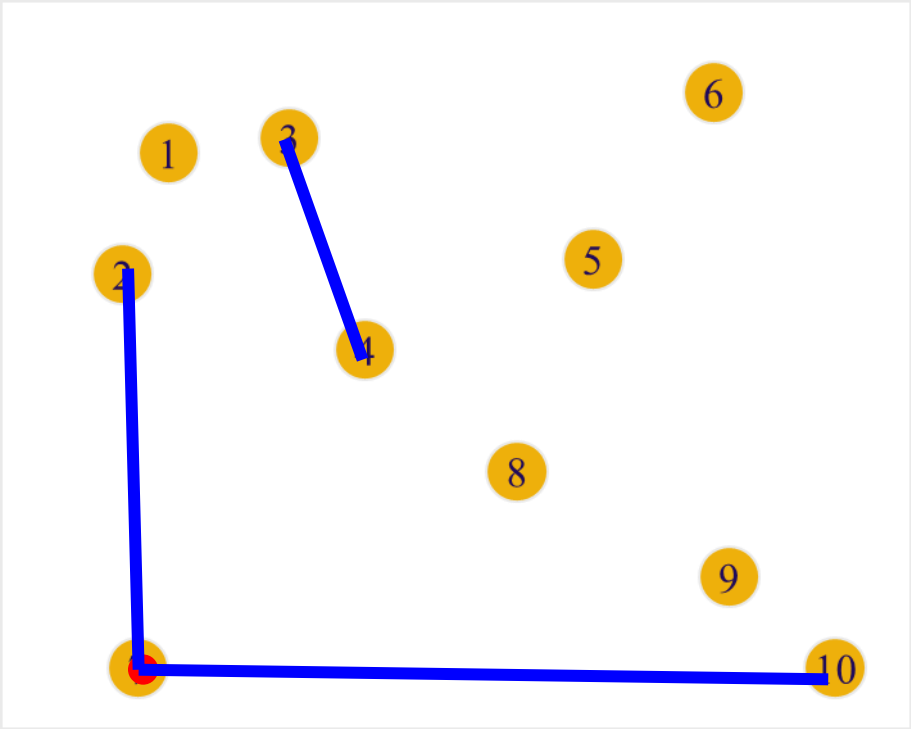
Divide and Conquer with Clustering



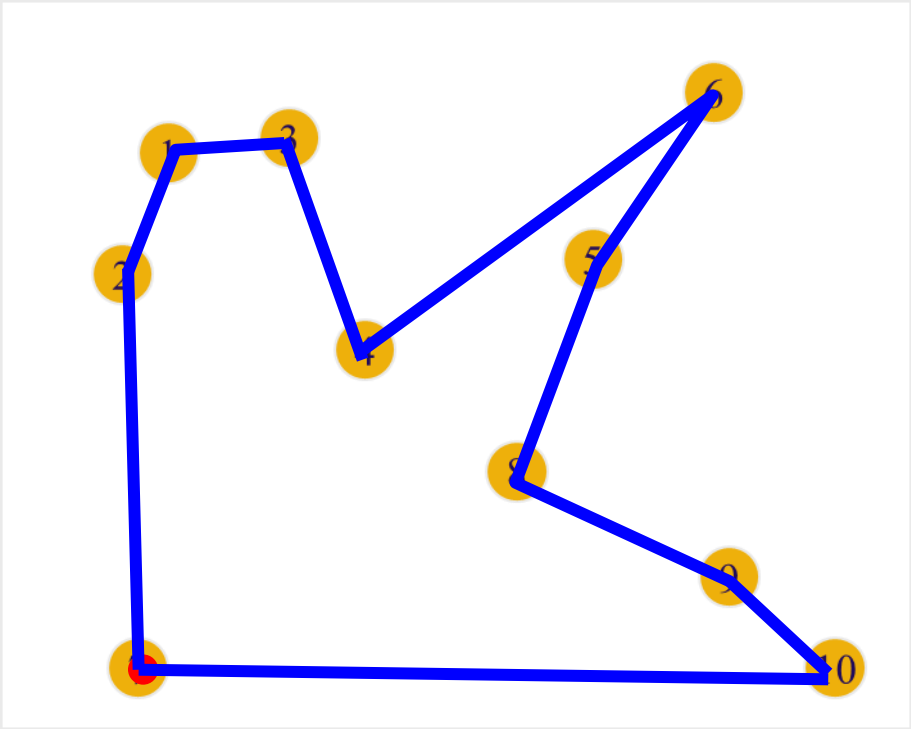
Divide and Conquer with Clustering



Divide and Conquer with Clustering



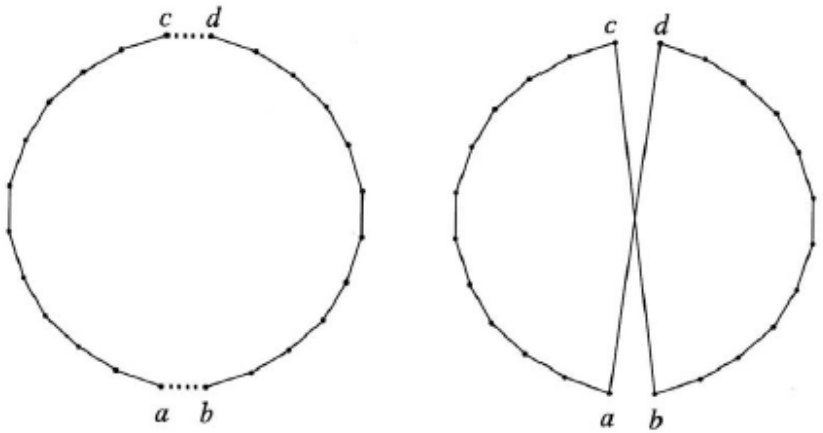
Divide and Conquer with Clustering



Improving Search Heuristic Algorithm

- **Step 0:** Choose any starting feasible solution x(0), and set solution index t = 0
- **Step 1:** If no move Δx in move set M is both improving and feasible at current solution x(t), stop. x(t) is a local optimum.
- **Step 2:** Choose some improving feasible move $\Delta x \in M$
- **Step 3:** Update x(t+1) = x(t) + Δx
- **Step 4:** T = t+1, return to Step 1

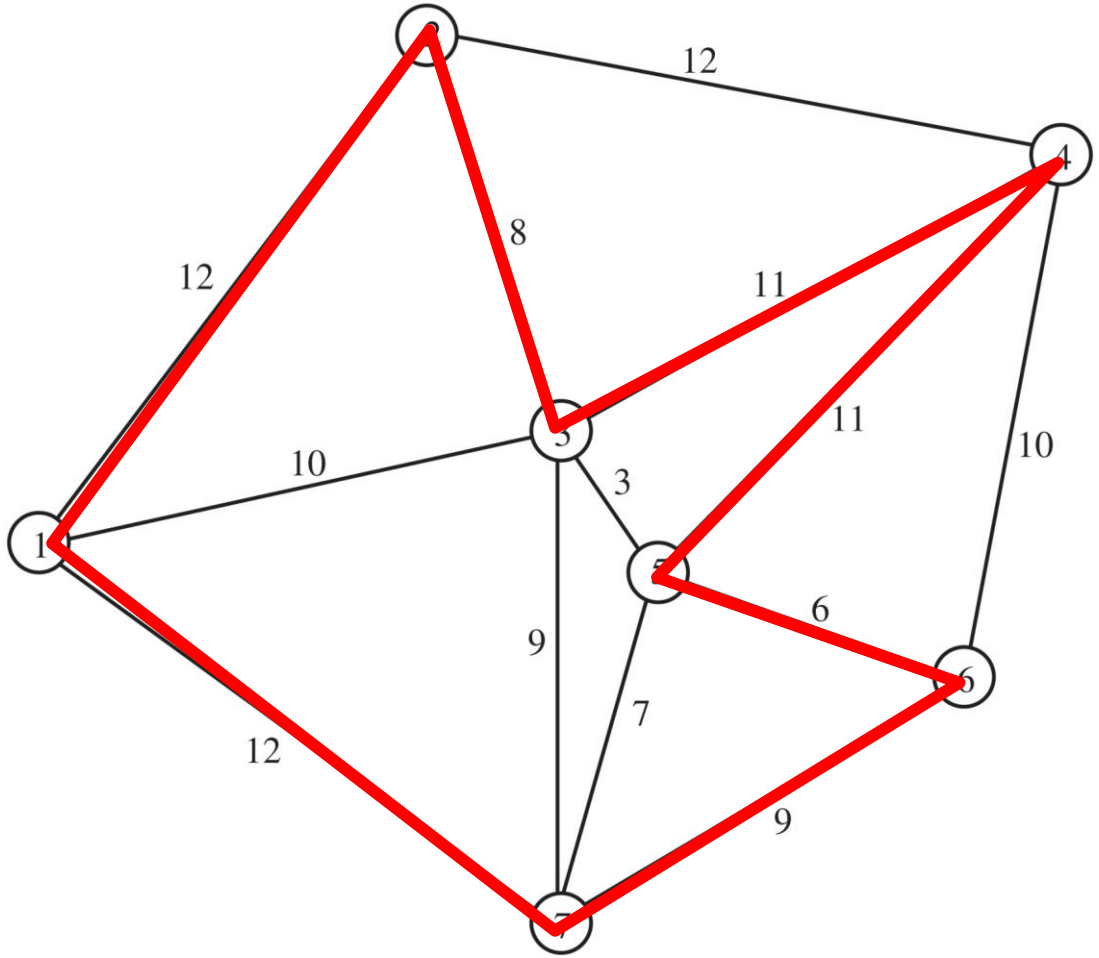
2-Opt



GAIN = (a,d)+(b,c)-(c,d)-(a,b)
Subpath (b,…,d) is reverted

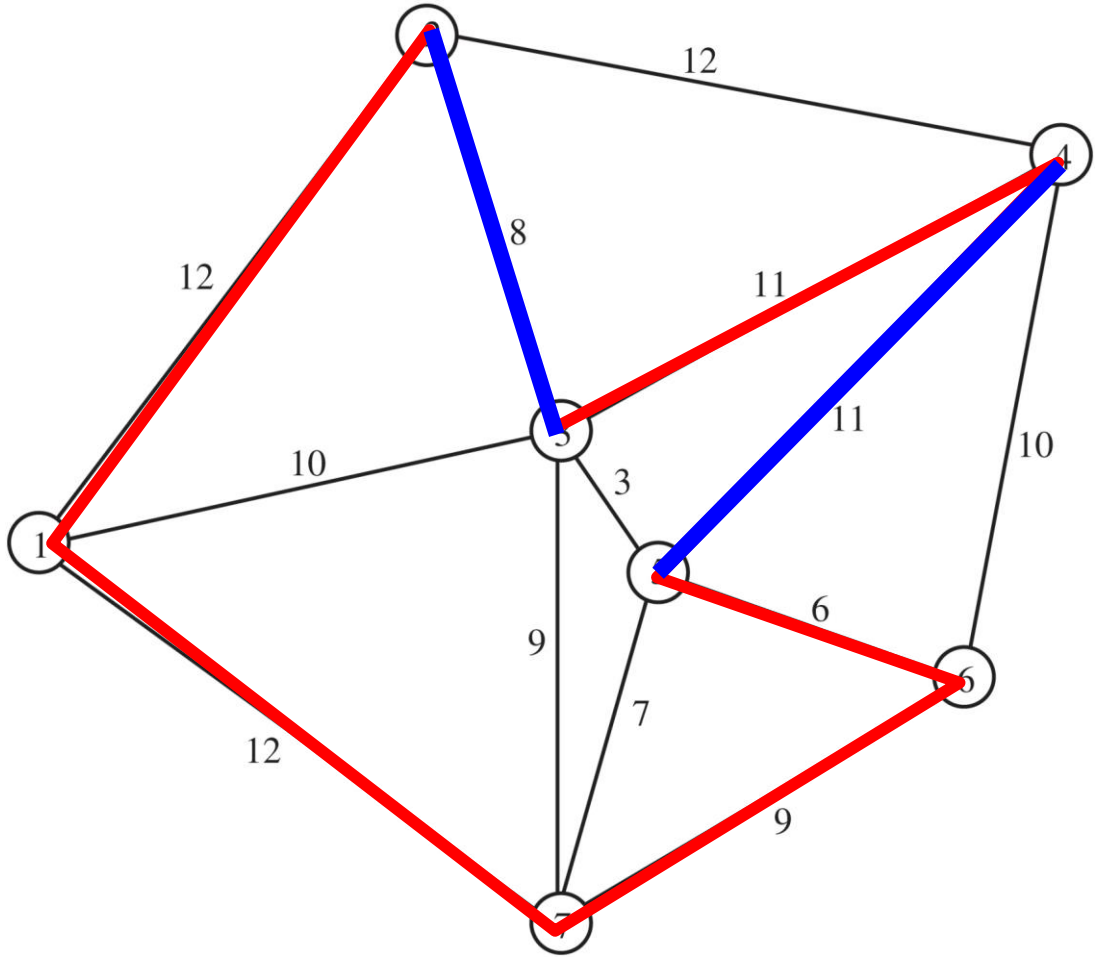
2-Opt

69



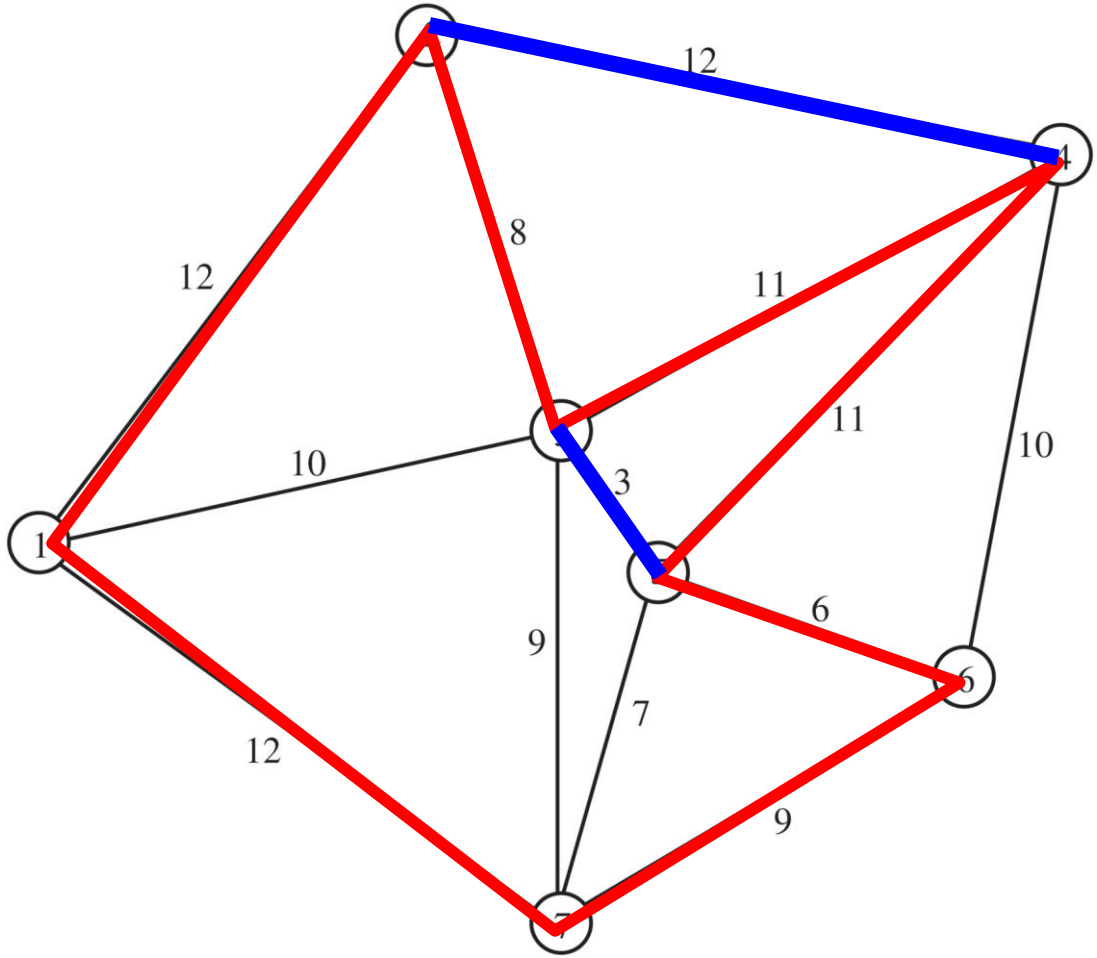
2-Opt

69



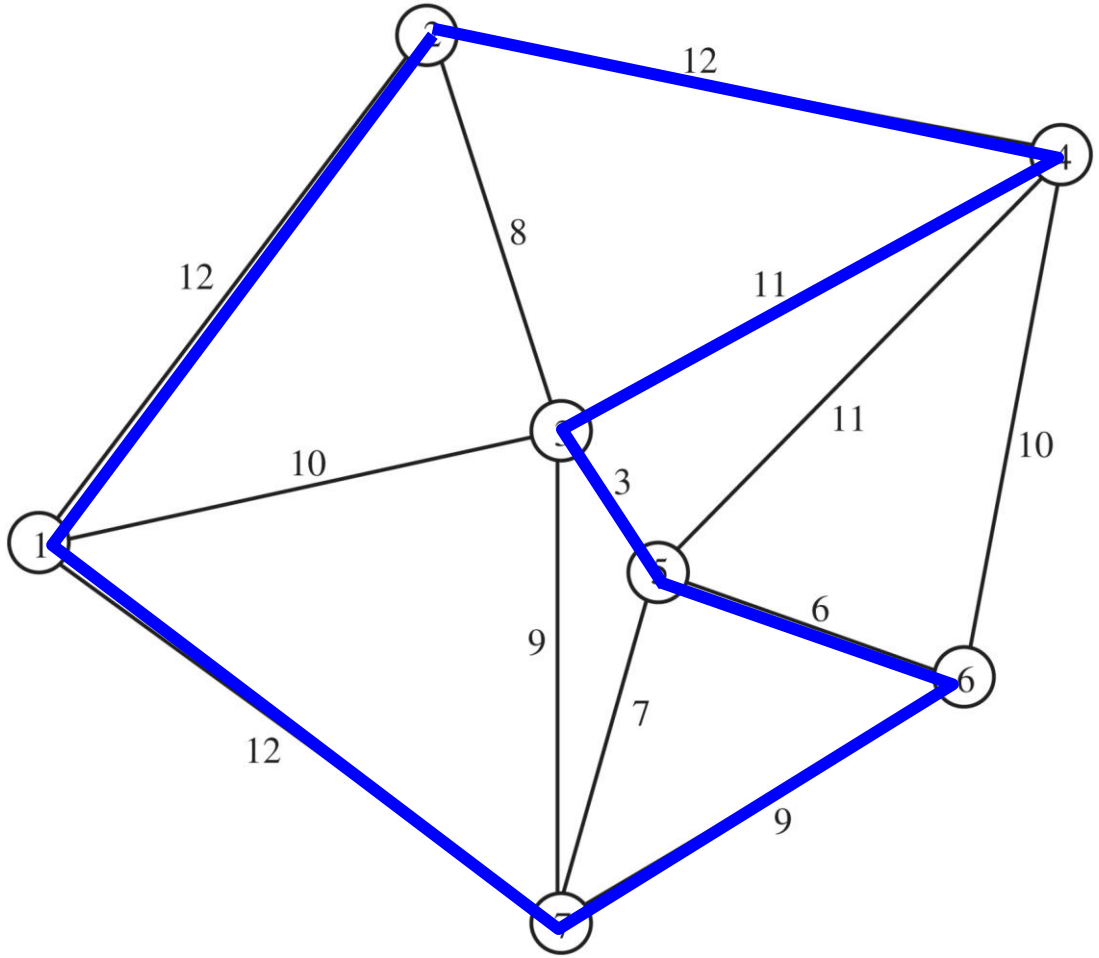
2-Opt

65



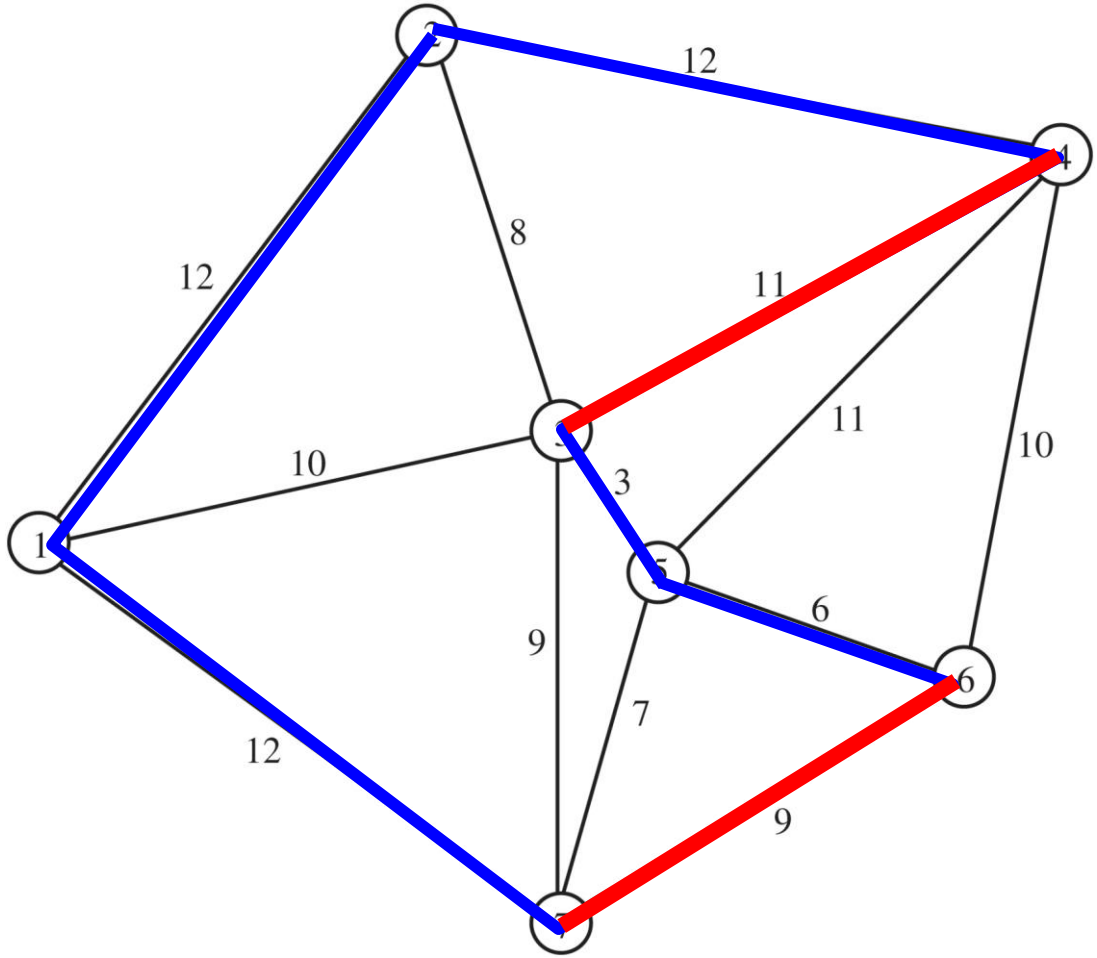
2-Opt

65



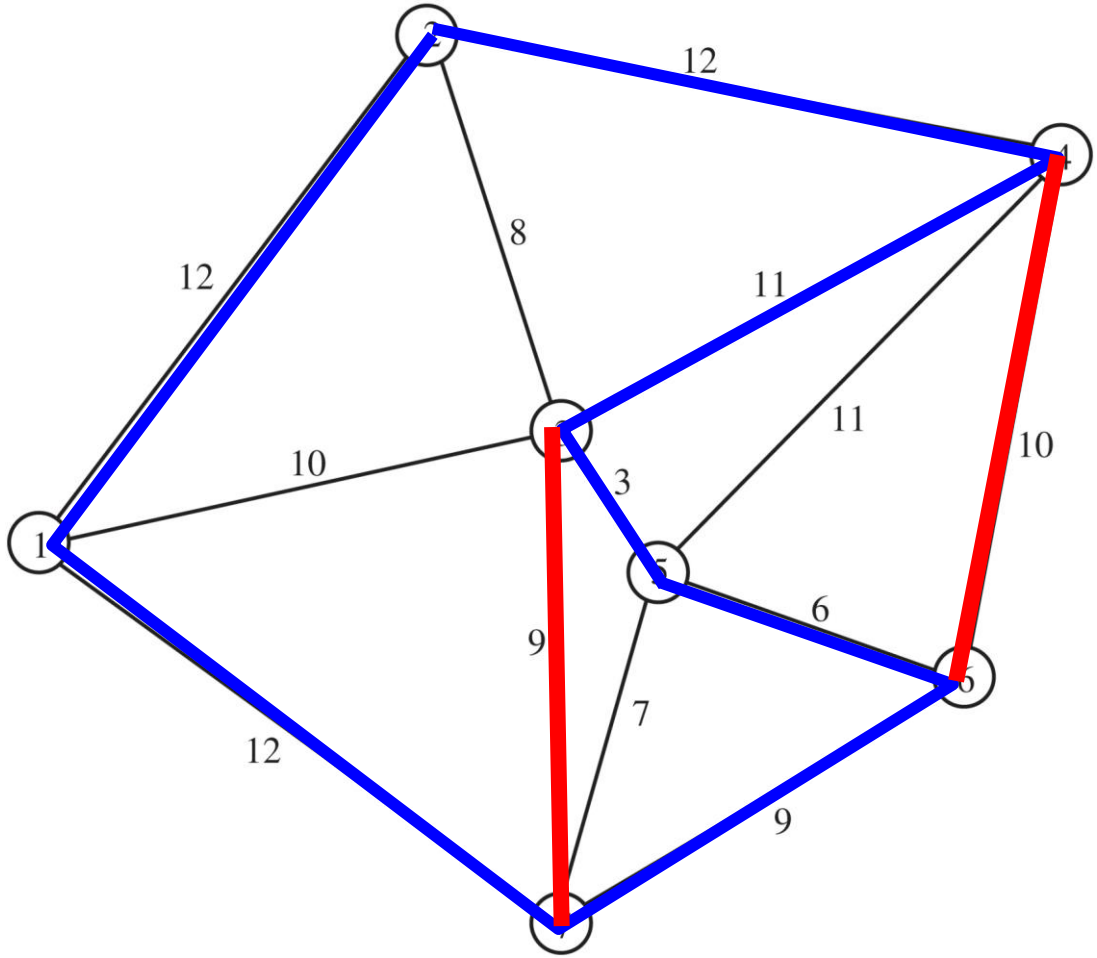
2-Opt

65



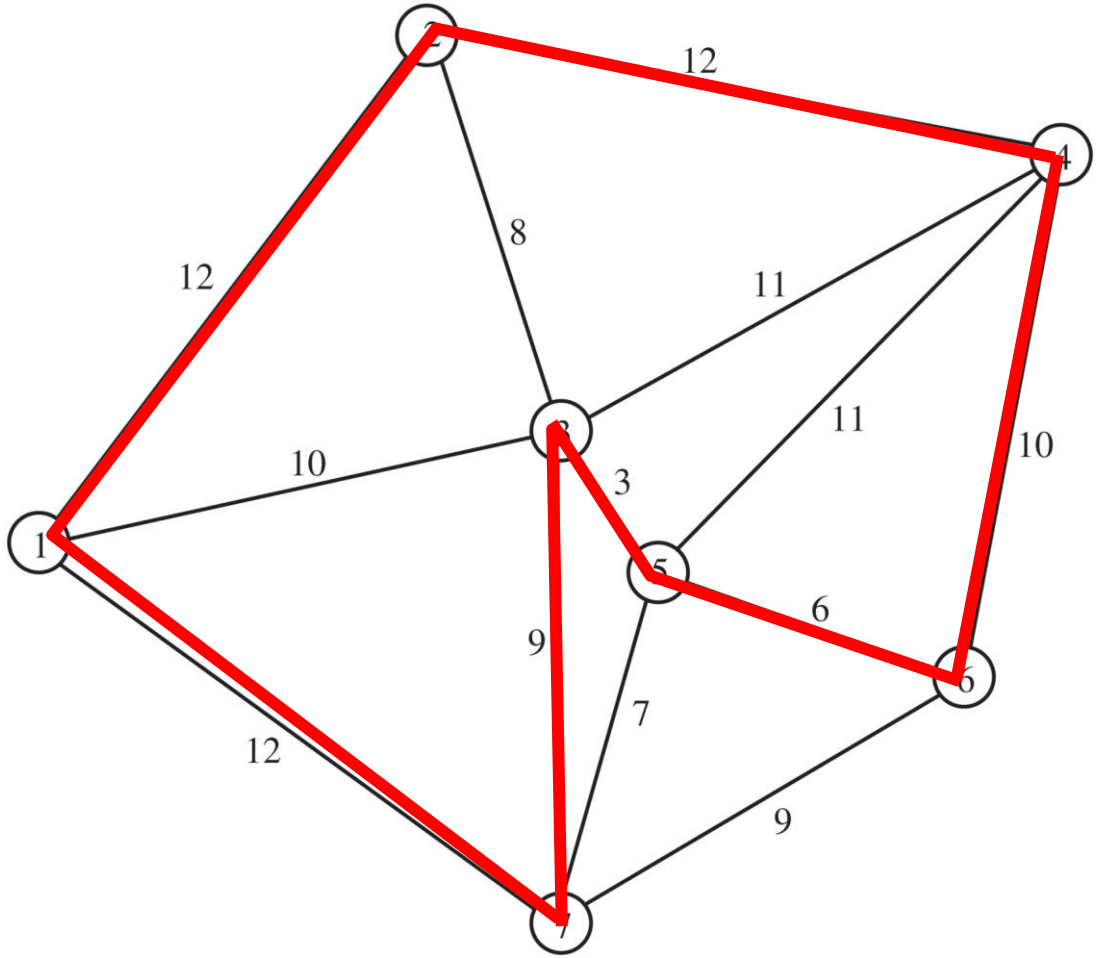
2-Opt

64



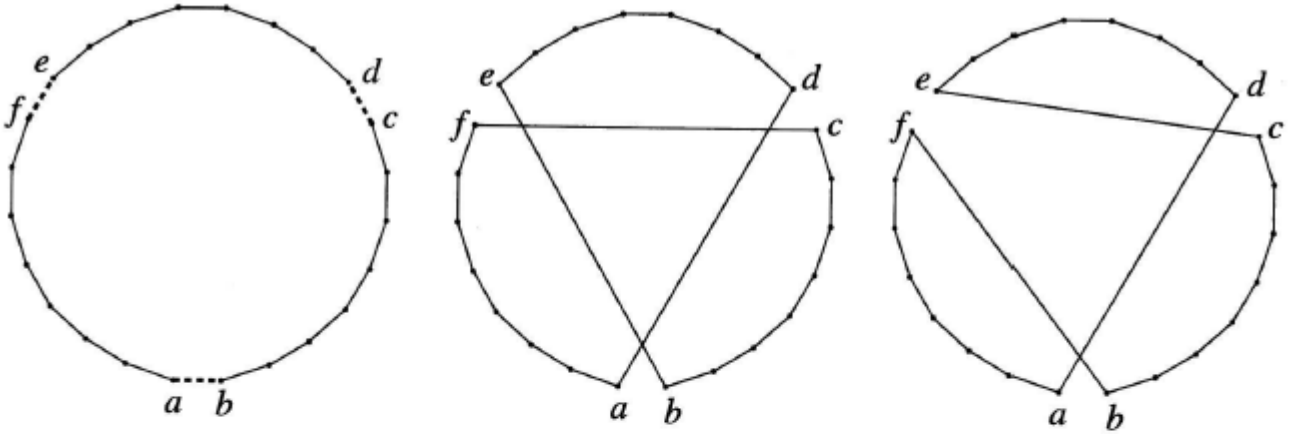
2-Opt

64



2-Opt: Real Example

3-Opt



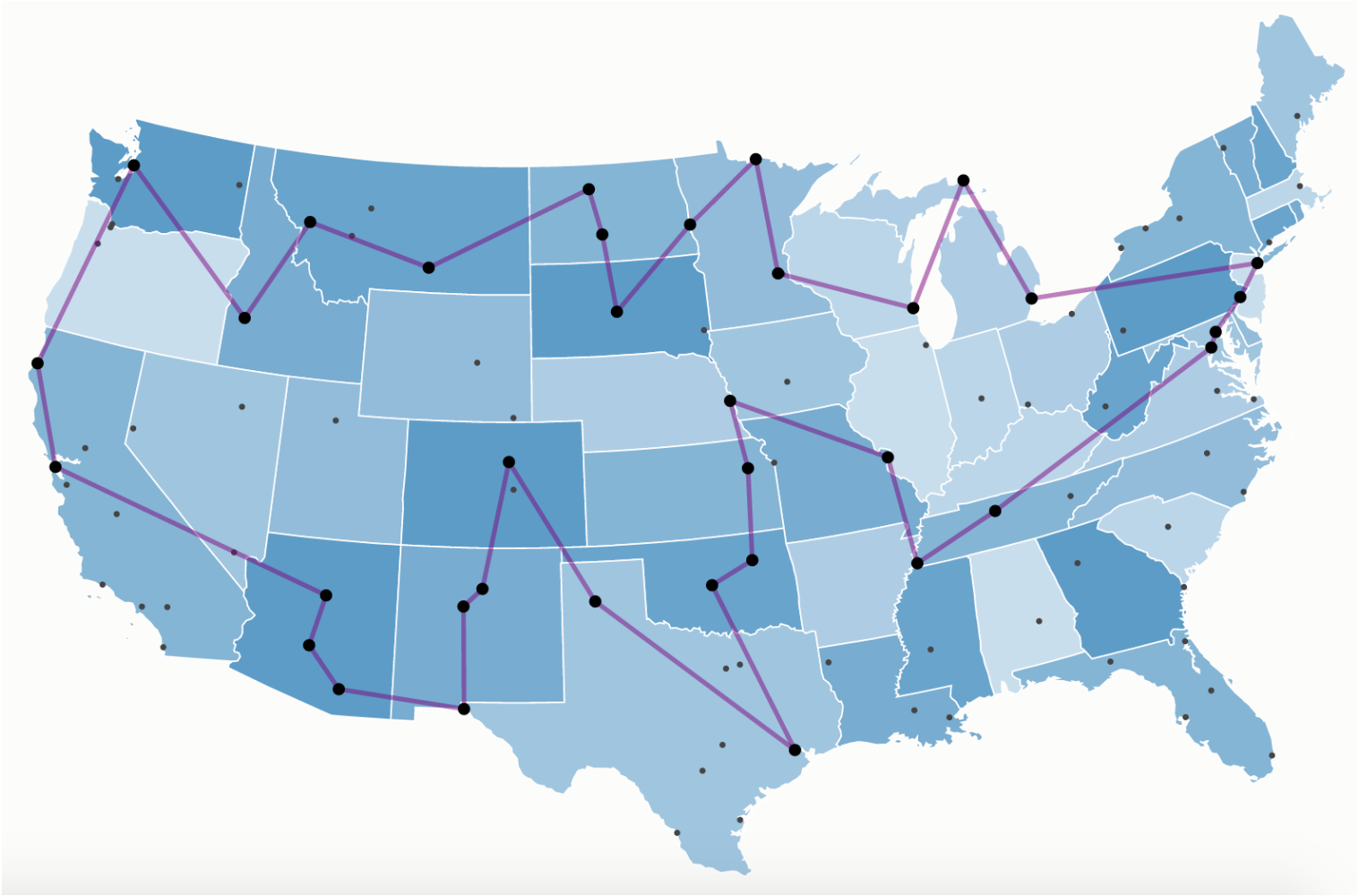
Two possible new tours

GAIN1 = (a,d)+(e,b)+(c,f)-(a,b)-(c,d)-(e,f) no path is reverted

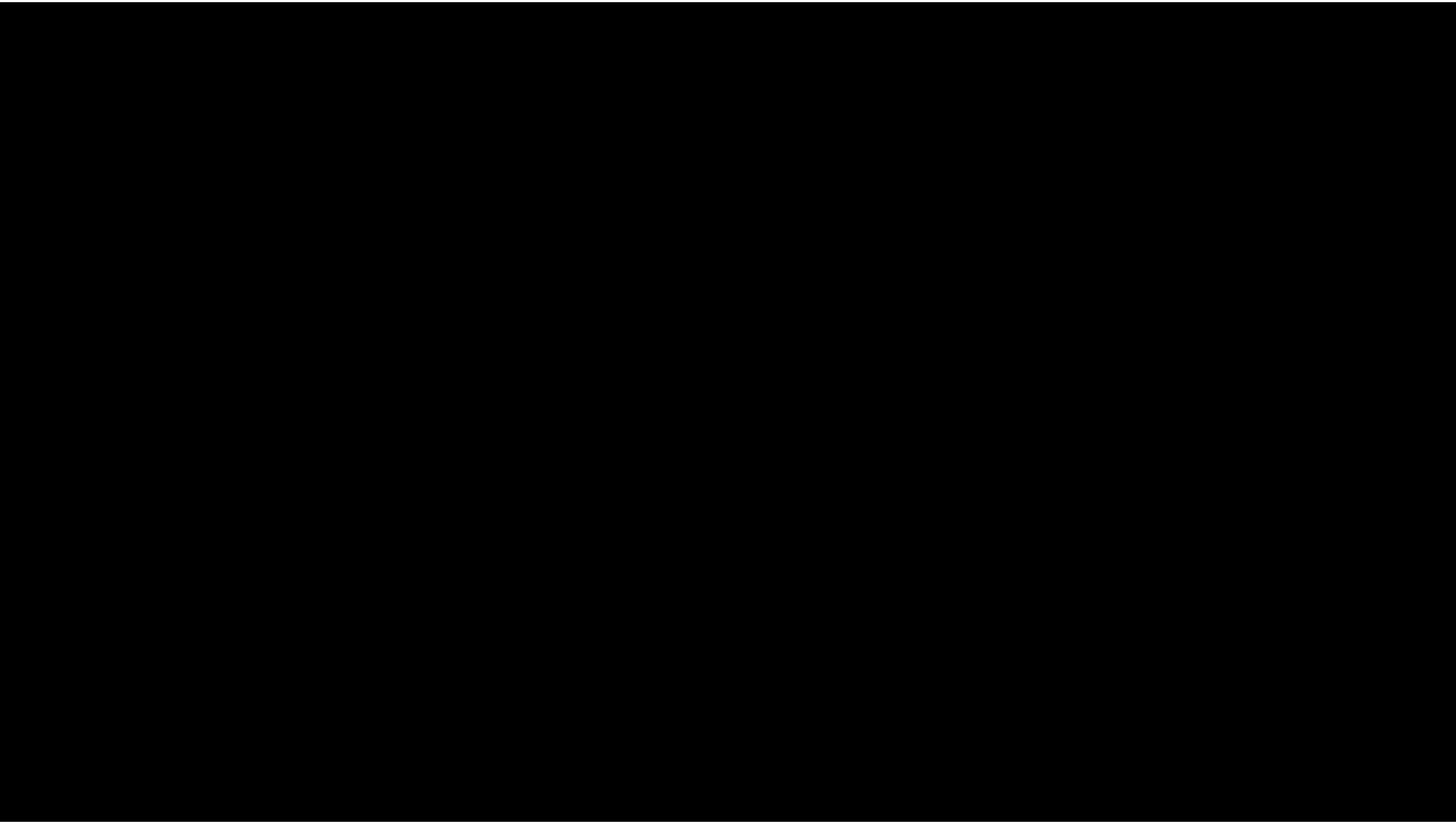
GAIN2 = (a,d)+(e,c)+(b,f)-(a,b)-(c,d)-(e-f) path (c,…,b) is reverted

Metaheuristics

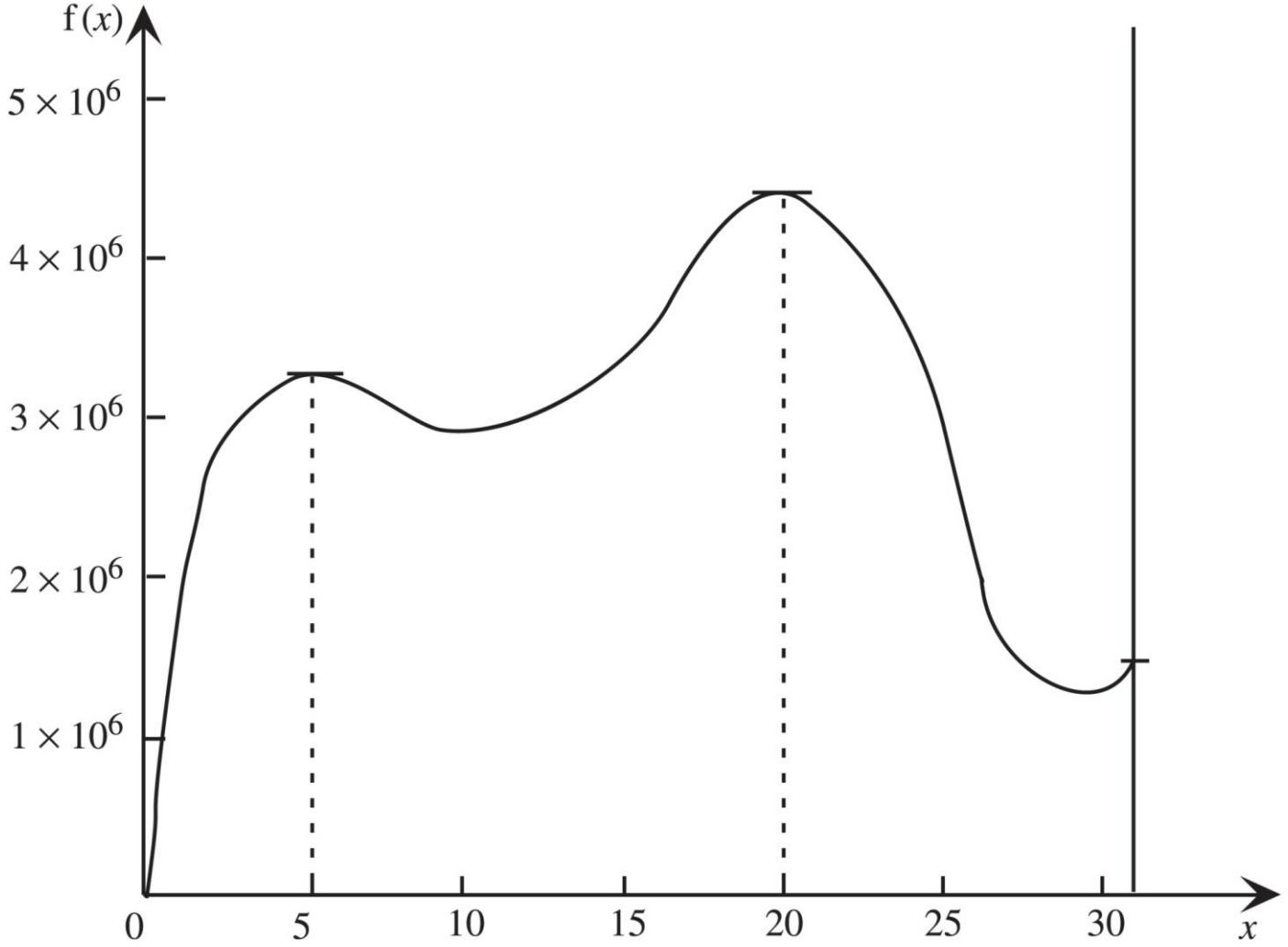
Limitation of Improving Search Heuristic Algorithm



Limitation of Improving Search Heuristic Algorithm



Limitation of the Improving Search Heuristic Algorithm in General



Limitation of the Improving Search Heuristic Algorithm in General

Max 18 x1 + 25 x2 + 11 x3 + 14 x4
s.t. 2 x1 + 2 x2 + x3 + x4 ≤ 3
x1,…, x4 = 0 or 1

Assume x(0) = (1,0,0,0), Obj value(z) = 18

Move set: { single complement move (0-1) }

Neighborhood: { (0,0,0,0), (1,1,0,0), (1,0,1,0), (1,0,0,1) }
(0,0,0,0): feasible, z = 0
(1,1,0,0): infeasible
(1,0,1,0): feasible, z = 29
(1,0,0,1): feasible, z = 32

x(1) = (1,0,0,1)
Neighborhood: { (0,0,0,1), (1,1,0,1), (1,0,1,1), (1,0,0,0) }
(0,0,0,1): feasible, z = 14
(1,1,0,1): infeasible
(1,0,1,1): infeasible
(1,0,0,0): feasible, z = 18

Stop

Heuristics vs. Metaheuristics

- Heuristics
  - Ad hoc: Chosen method depends on the specific problem
  - Often iterative: Each iteration focuses on finding a better solution than the one found previously
  - Drawback of heuristics (i.e., local search procedures):

    Procedure stops when a local optimum is reached

    Which optimum is found depends on where procedure begins the search

Heuristics vs. Metaheuristics

- Metaheuristics
  - Some complex optimization problems may not be possible to solve with exact algorithms presented in the OR I and OR II lectures
  - A metaheuristic provides a general structure and strategy guidelines to develop a heuristic method to fit a particular type of problem
  - Metaheuristics can escape from a local optimum: Trial solutions that immediately follow a local optimum are allowed to be inferior
  - Metaheuristics are useful to deal with non-dominated solutions of complex problems

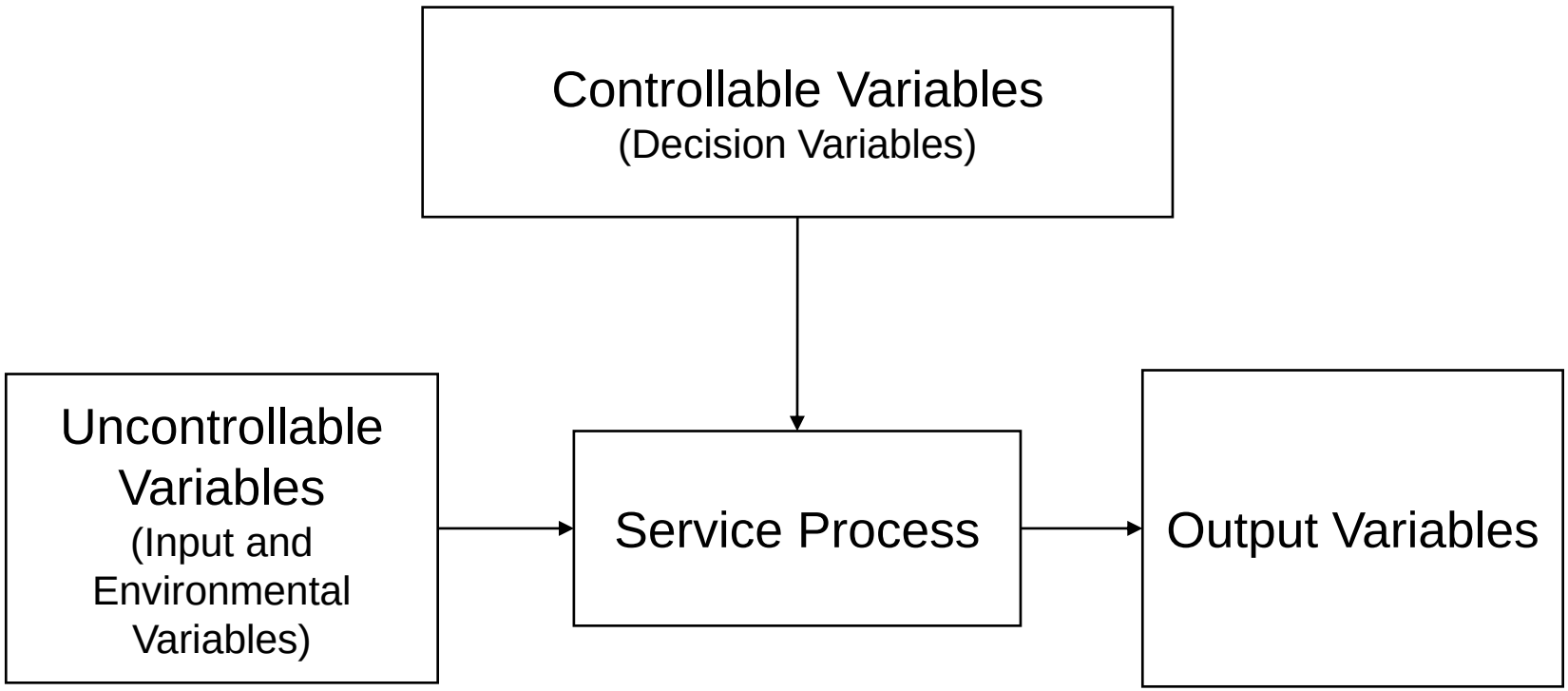
Concept of “Meta”

- Higher order thinking skills
  - Cognition about cognition
  - Thinking about thinking
  - Knowing about knowing
  - Algorithm of algorithms

Metaheuristic Algorithms for "Stochastic" Problems

"Model building" "Search optimal setting for **x**"

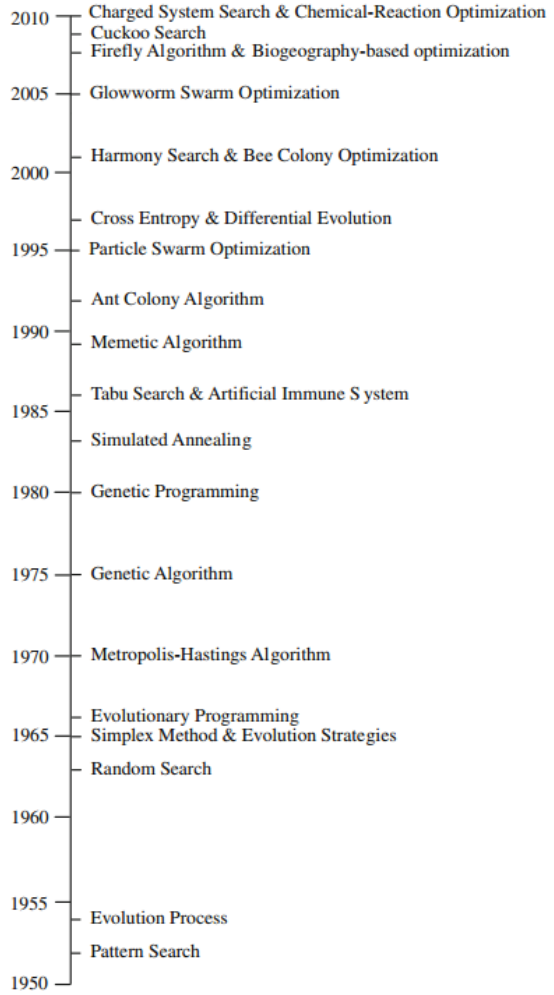
$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon \quad \rightarrow \quad \hat{y}(\mathbf{x}) \quad \rightarrow \quad \underset{\mathbf{x}\in\Omega}{\text{optimize}}\ \hat{y}(\mathbf{x})$$



Metaheuristics Algorithms (Computational Intelligence Algorithms)

- Higher-level procedure or heuristic designed to find a sufficiently good solution

- Often inspired from nature or society
  - Genetic Algorithm
  - Tabu Search
  - Simulated Annealing
  - Ant Colony Algorithm
  - Particle Swarm Optimization
  - Cuckoo Search Algorithm
  - …



Genetic Algorithm

Genetic Algorithm: An Old Champion

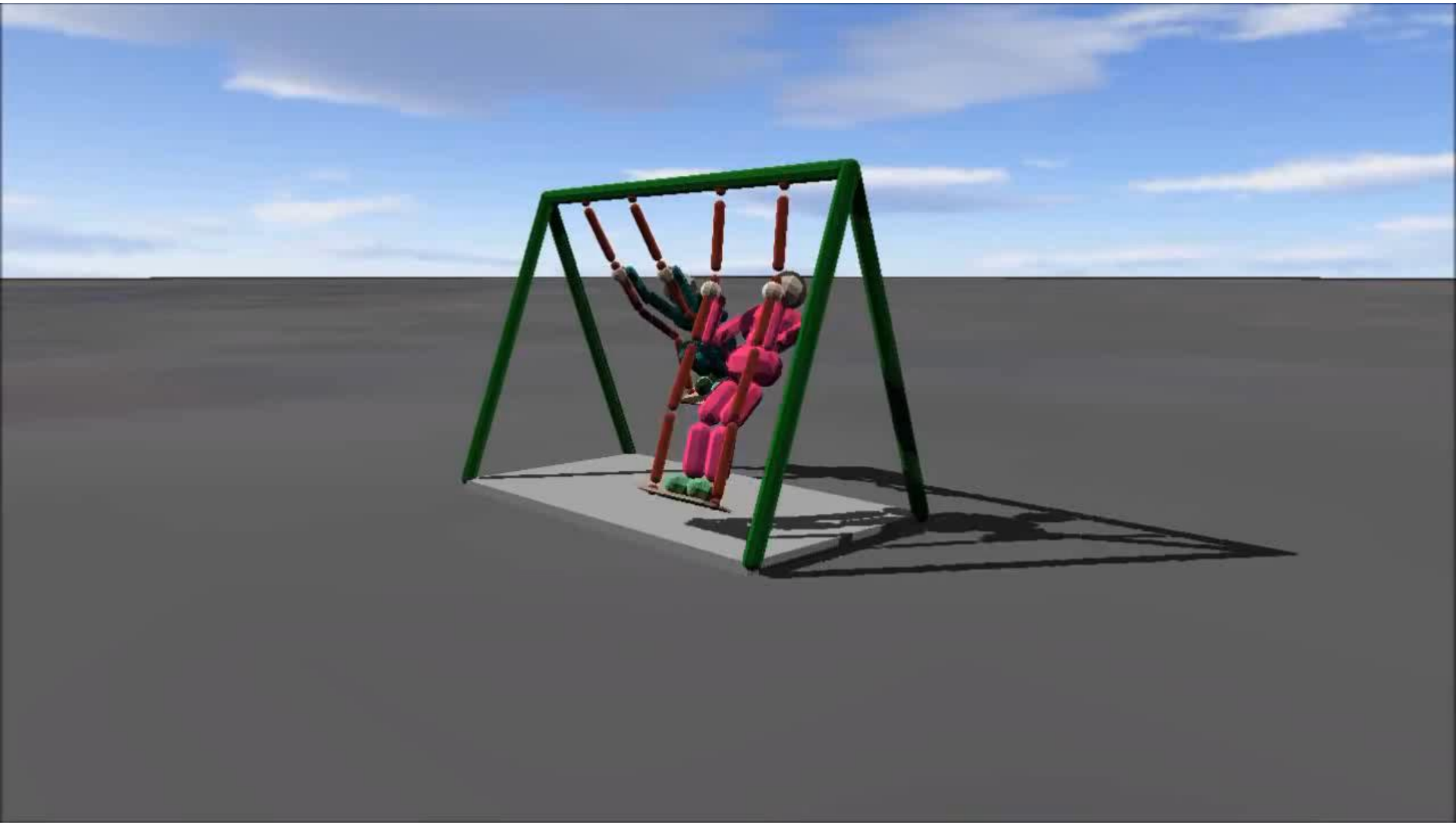
- Basic concepts of genetic algorithms
  - Analogous to Darwin's theory of evolution, "survival of the fittest," and natural selection
  - Concerned with populations of solutions, rather than dealing with one trial solution at a time
  - Generate improving populations of trial solutions as they proceed
  - Mutations can occur

Genetic Algorithm: Overview

- Select algorithm's parameters
  - Population size
  - Objective function f (*) for fitness evaluation
  - Maximum no. of generations
  - Mutation probability
  - Etc.
- Begin
  - Generate initial population
  - Calculate fitness value for initial population
- Repeat until stopping criterion is met
  - Select parents from current population based on their fitness
  - Generate children through crossover and mutation
  - Generate new population using some replacement strategy

Genetic Algorithm: Illustration

Genetic Algorithm: Illustration



Genetic Algorithm: Procedure

- Encoding methods
  - Such as strings of binary digits
  - Makes it easier to generate children and create mutations
  - Develop an appropriate encoding scheme is an important part of applying a genetic algorithm
- Random numbers are used to generate children from a pair of parents
- Procedure for generating a child should be designed as well

Genetic Algorithm: Procedure

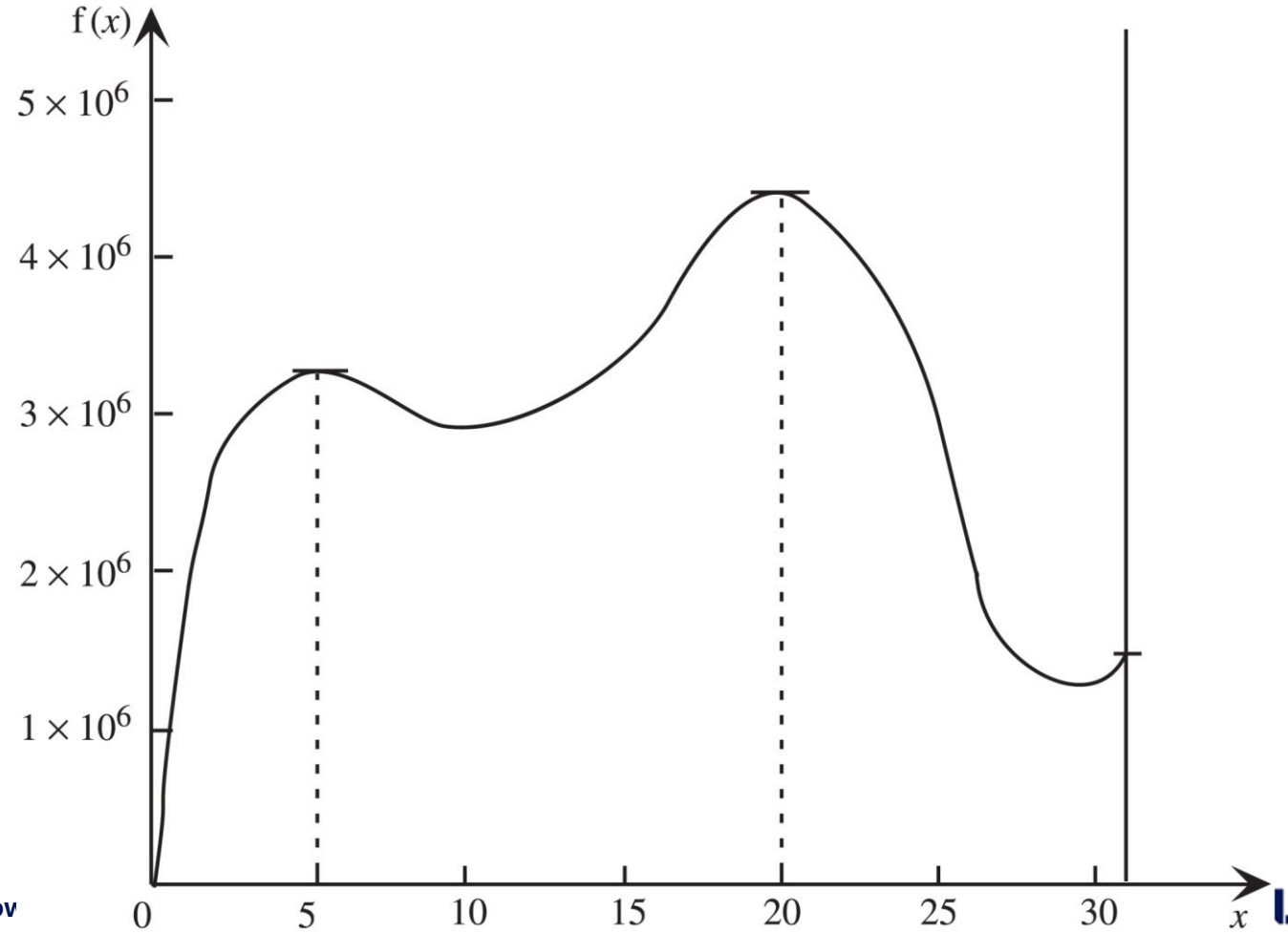
- Outline of a basic genetic algorithm
  - Initialization
  - Iteration
  - Stopping rule

- First ask these questions: Answer based on the nature of the specific problem
  - What population size should be used?
  - How should parents be selected?
  - How should children's features be derived from parents‘ features?
  - How should mutations be injected?
  - Which stopping rule should be used?

GA Example

$$Maximize\ f(x) = 12x^5 - 975x^4 + 28000x^3 - 345000x^2 + 1800000x$$

$$subject\ to\ 0 \le x \le 31, x\ is\ integer$$



GA Example: The Concept of Encoding

- If x = 3,

  the gene is 00011

- If x = 10,

  the gene is 01010

- If x = 25,

  the gene is 11001

GA Example: The Concept of Inheritance and Use of rand()

- Father: 00011

  Mother: 01010

- Child 1: 0x01x

  Child 2: 0x01x

- We determine the value of x (i.e., from whom to inherit) randomly (i.e., using a rand())

  0.0000–0.4999 corresponds to the digit being 0

  0.5000–0.9999 corresponds to the digit being 1

GA Example: The Concept of Mutation

- Assume that our rand() generated 0.73, 0.52, 0.04, 0.36, then our children should be
  - Father: 00011, Mother: 01010

    Child 1: 0x01x, Child 2: 0x01x
  - Child 1: 01011 (11)

    Child 2: 00010 (2)

- However, we should consider the possibility of a mutation

  (if p = 0.1, then 0.0000–0.0999 corresponds to a mutation, otherwise no mutation)

  Possible result of mutation:

  Child 1: 01010 (11)

  Child 2: 00110 (6)

GA Example

Member	Initial Population	Value of x	Fitness (Objective)
1	01111	15	3,628,125
2	00100	4	3,234,688
3	01000	8	3,055,616
4	10111	23	3,962,141
5	01010	10	2,950,000
6	01001	9	2,978,613
7	00101	5	3,303,125
8	10010	18	4,239,216
9	11110	30	1,350,000
10	10101	21	4,353,237

GA Example

Member	Initial Population	Value of x	Fitness (Objective)
1	01111	15	3,628,125
2	00100	4	3,234,688
3	01000	8	3,055,616
4	10111	23	3,962,141
5	01010	10	2,950,000
6	01001	9	2,978,613
7	00101	5	3,303,125
8	10010	18	4,239,216
9	11110	30	1,350,000
10	10101	21	4,353,237

GA Example

Member	Initial Population	Value of x	Fitness (Objective)
1	01111	15	3,628,125
2	00100	4	3,234,688
3	01000	8	3,055,616
4	10111	23	3,962,141
5	01010	10	2,950,000
6	01001	9	2,978,613
7	00101	5	3,303,125
8	10010	18	4,239,216
9	11110	30	1,350,000
10	10101	21	4,353,237

GA Example

Member	Initial Population	Value of x	Fitness (Objective)
1	01111	15	3,628,125
2	00100	4	3,234,688
3	01000	8	3,055,616
4	10111	23	3,962,141
5	01010	10	2,950,000
6	01001	9	2,978,613
7	00101	5	3,303,125
8	10010	18	4,239,216
9	11110	30	1,350,000
10	10101	21	4,353,237

Member	Parents	Children	Value of x	Fitness (Objective)
10	10101	00101	5	3,303,125
2	00100	10101	21	4,353,237
8	10010	10011	19	4,357,164
4	10111	10110	22	4,207,984
1	01111	01011	11	2,980,637
6	01001	01111	15	3,628,125

GA Example

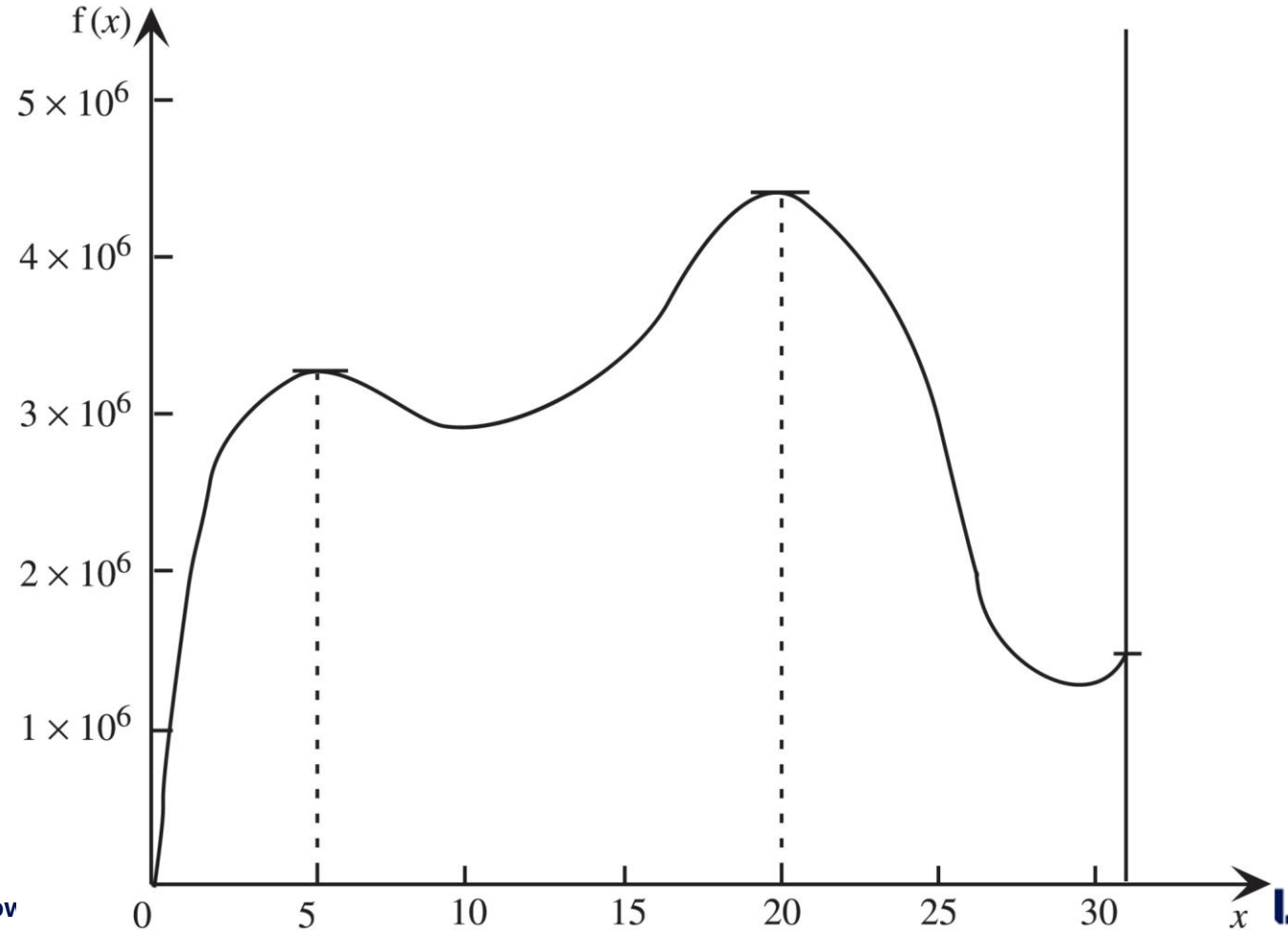
Member	Initial Population	Value of x	Fitness (Objective)
1	01111	15	3,628,125
2	00100	4	3,234,688
3	01000	8	3,055,616
4	10111	23	3,962,141
5	01010	10	2,950,000
6	01001	9	2,978,613
7	00101	5	3,303,125
8	10010	18	4,239,216
9	11110	30	1,350,000
10	10101	21	4,353,237

Member	Parents	Children	Value of x	Fitness (Objective)
10	10101	00101	5	3,303,125
2	00100	10001	17	4,064,259
8	10010	10011	19	4,357,164
4	10111	10100	20	4,400,000
1	01111	01011	11	2,980,637
6	01001	01111	15	3,628,125

GA Example: Inheritance and Mutation

$$Maximize\ f(x) = 12x^5 - 975x^4 + 28000x^3 - 345000x^2 + 1800000x$$

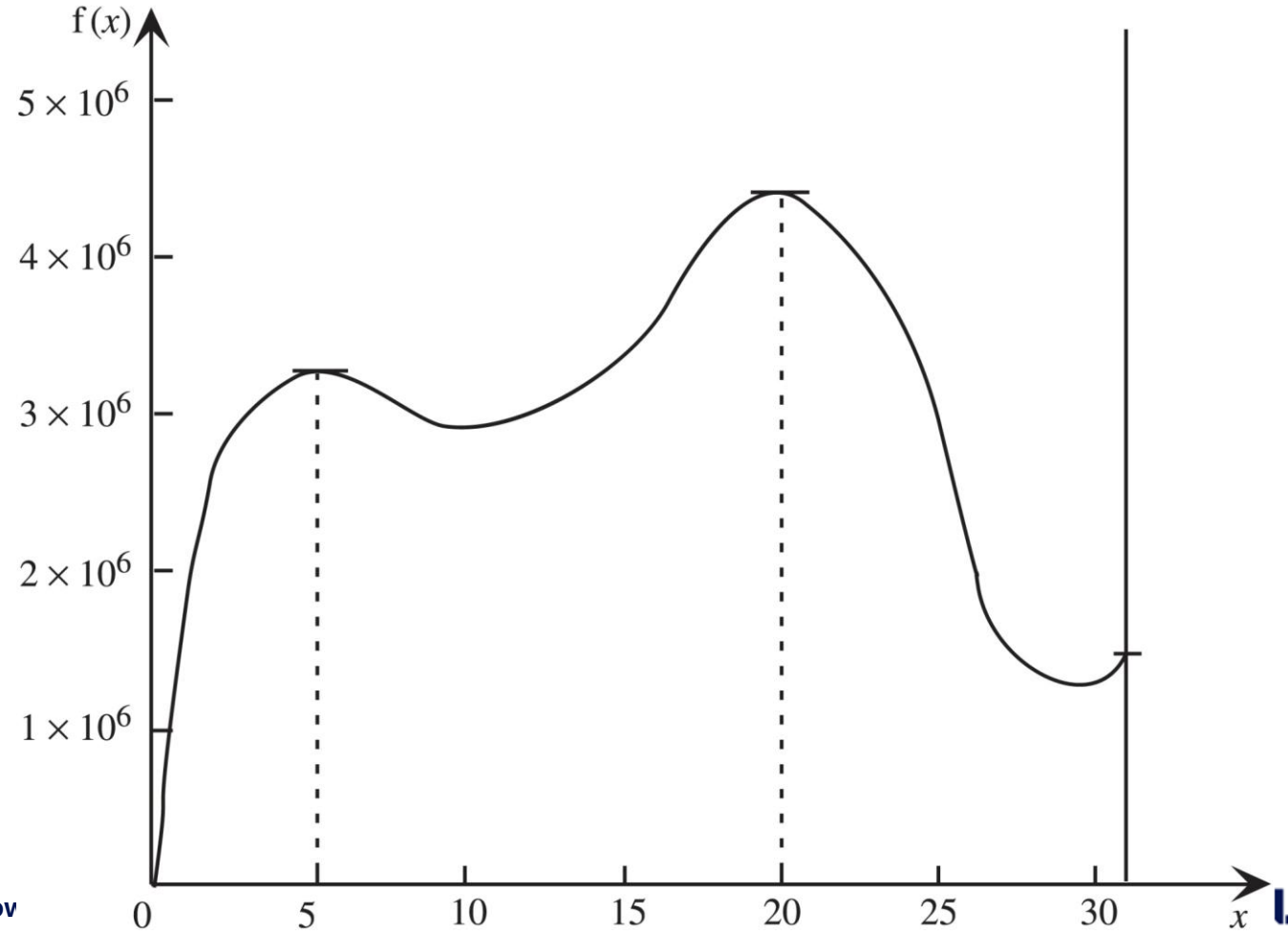
$$subject\ to\ 0 \le x \le 31, x\ is\ integer$$



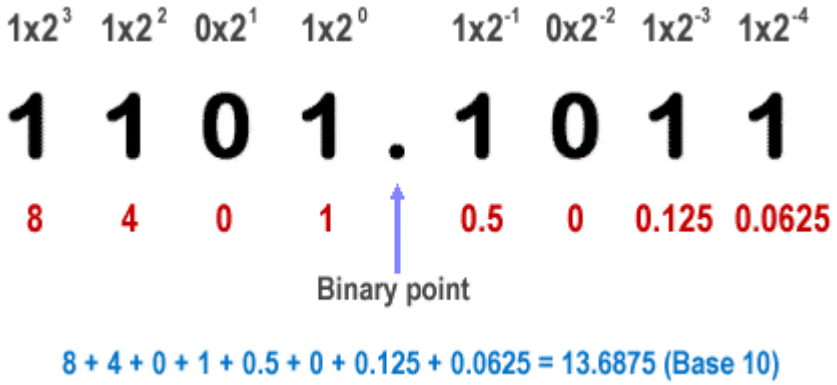
GA Example

$$Maximize\ f(x) = 12x^5 - 975x^4 + 28000x^3 - 345000x^2 + 1800000x$$

$$subject\ to\ 0 \le x \le 31, \sout{x\ is\ integer}$$



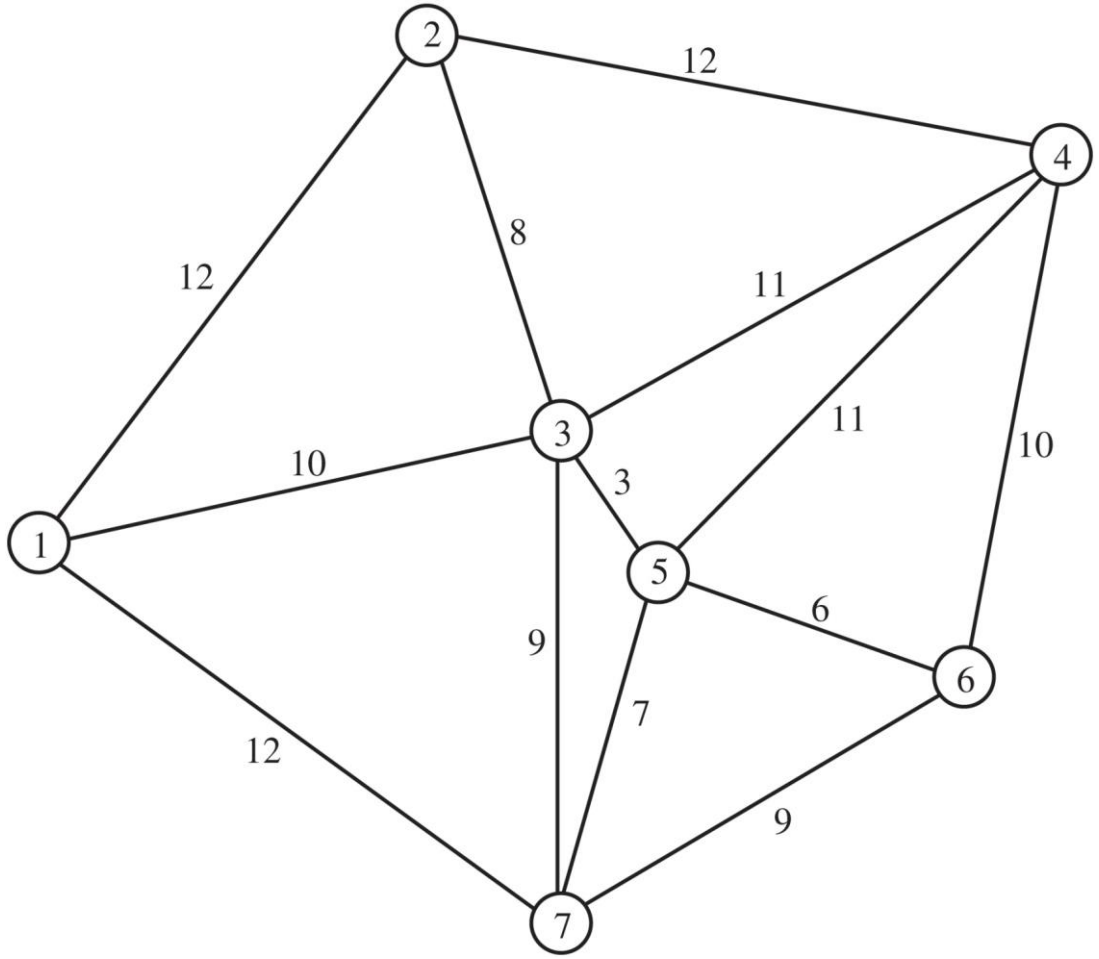
Decimal Fraction to Binary Conversion



- Example
  - $23.675_{10} = 10111.10100_2$
  - $23.66_{10} \approx 10111.10101_2$

Let's Solve a TSP Problem with GA

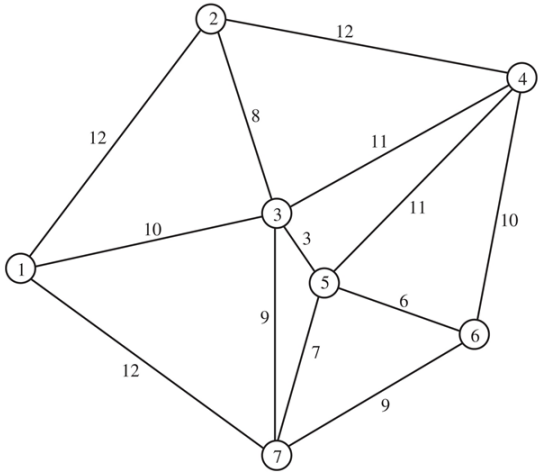
Let's Solve a TSP Problem with GA

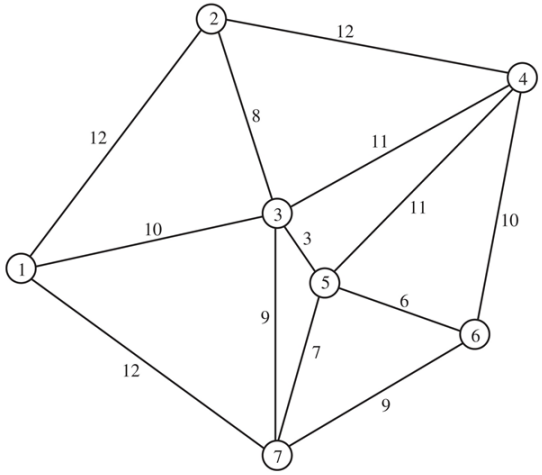


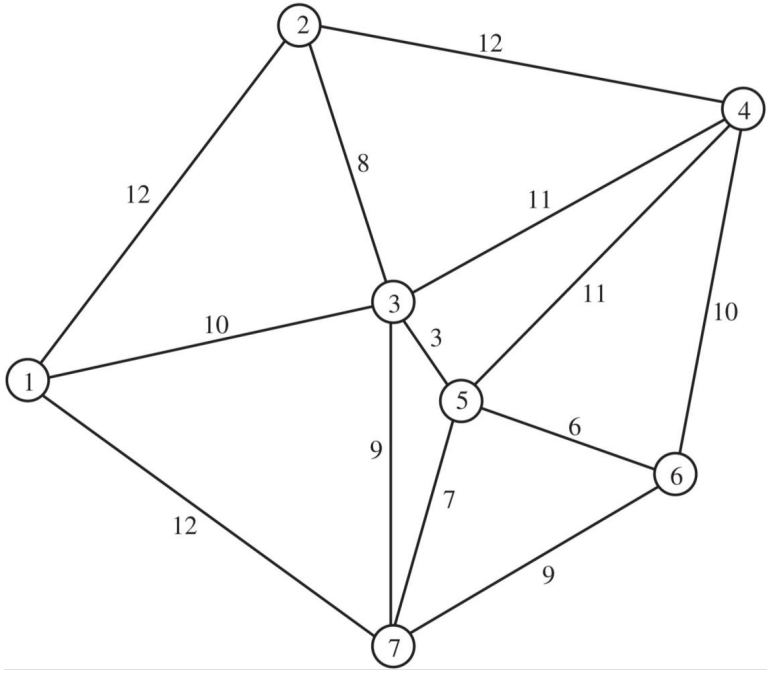
Let's Solve a TSP Problem with GA

- Encoding scheme: ex) 1-2-3-4-5-6-7-1 & 1-2-4-6-5-7-3-1
- Procedure for generating a child
  - Use a random number to select between link options
  - Check for mutation: If the next random number is less than 0.10, a mutation occurs, and link selected in the last step is rejected
  - Continuation: Add the selected link to the end of the child's current tour and start from the added city
  - Iterate until all cities but one have been visited
  - Add link to the final remaining city

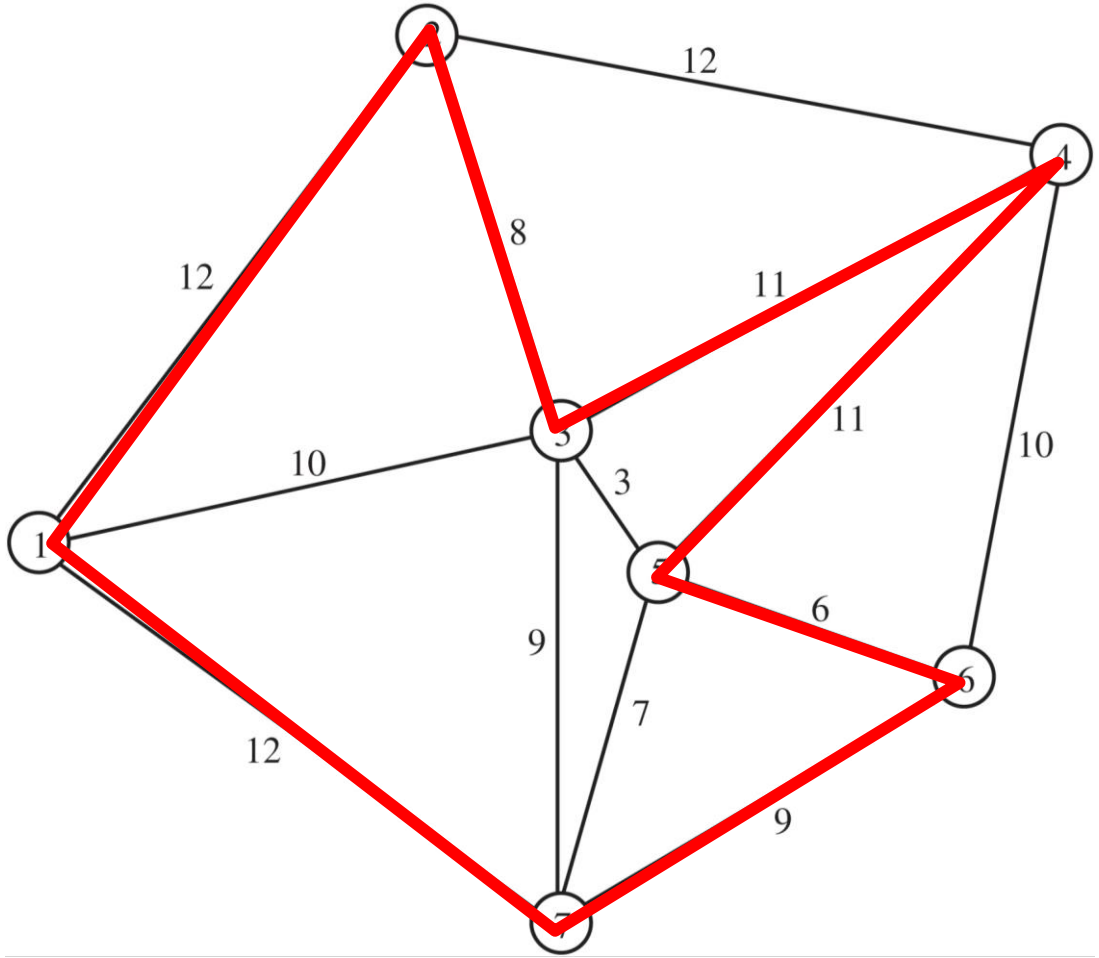
Inheritance



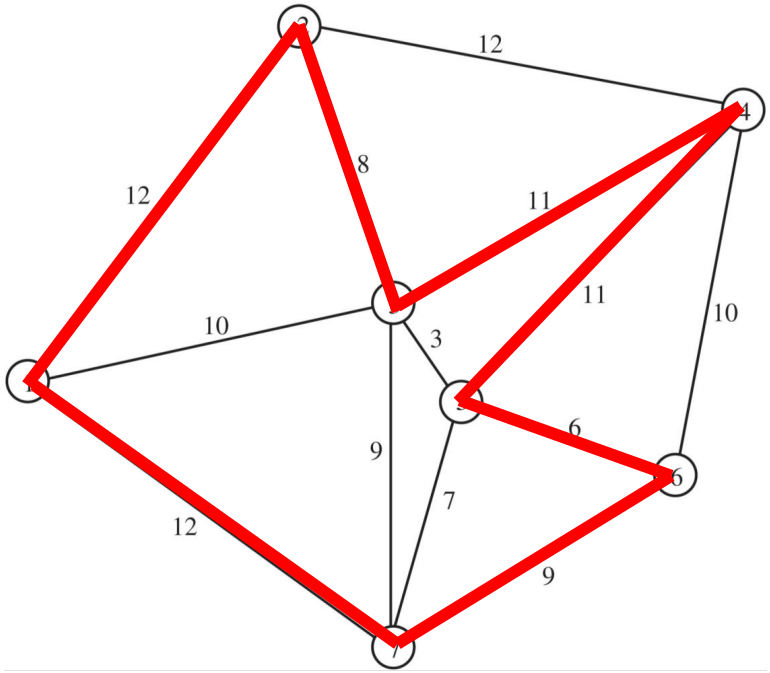


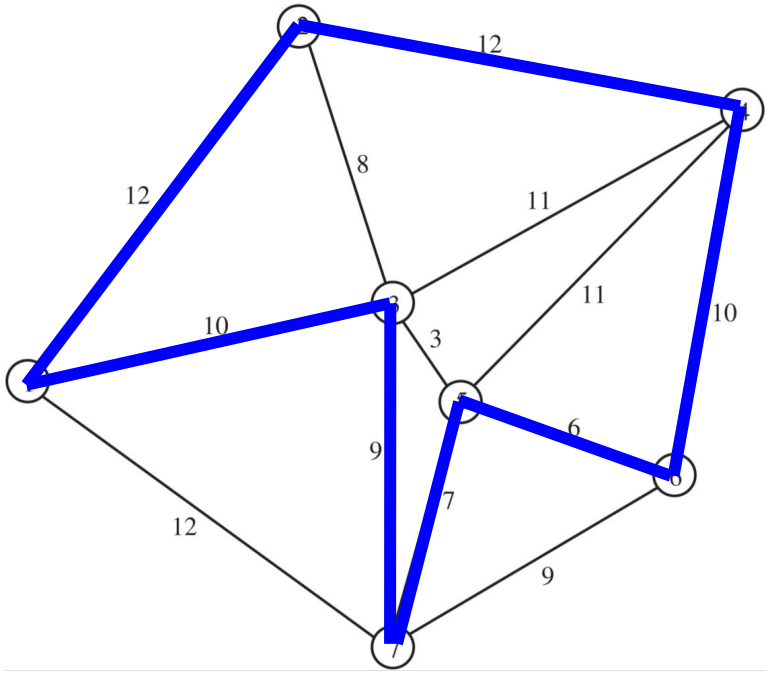


Random Search



Parents





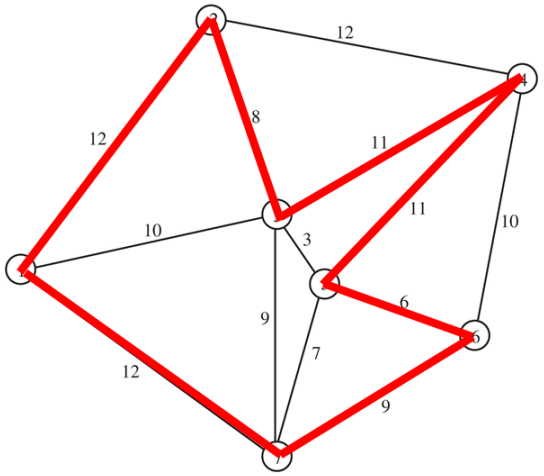
Let's Solve a TSP Problem with GA

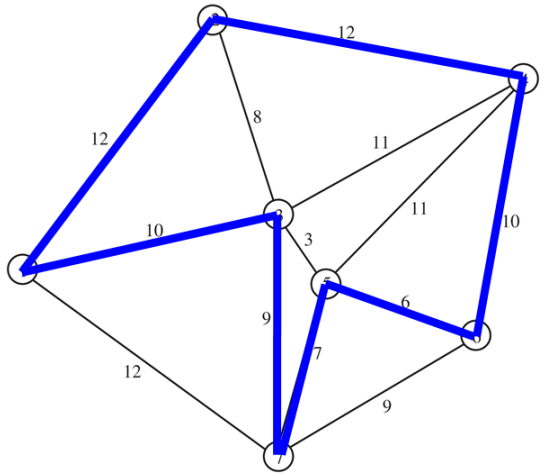
Parent P1: **1-2-3-4-5-6-7-1**
Parent P2: **1-2-4-6-5-7-3-1**

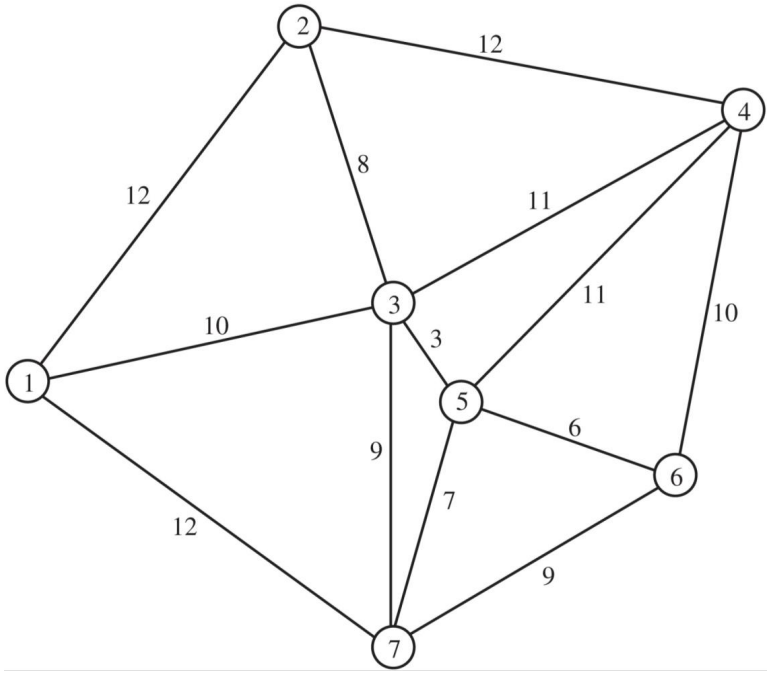
Link	Options	Random Selection	Tour
1	1-2, 1-7, 1-2, 1-3	1-2	1-2
2	2-3, 2-4	2-4	1-2-4
3	4-3, 4-5, 4-6	4-3	1-2-4-3
4	3-5*, 3-7	3-5*	1-2-4-3-5
5	5-6, 5-6, 5-7	5-6	1-2-4-3-5-6
6	6-7	6-7	1-2-4-3-5-6-7
7	7-1	7-1	1-2-4-3-5-6-7-1

*A link that completes a sub-tour reversal

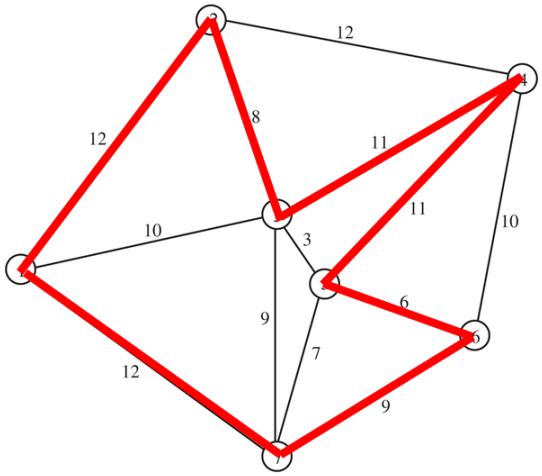
Inheritance

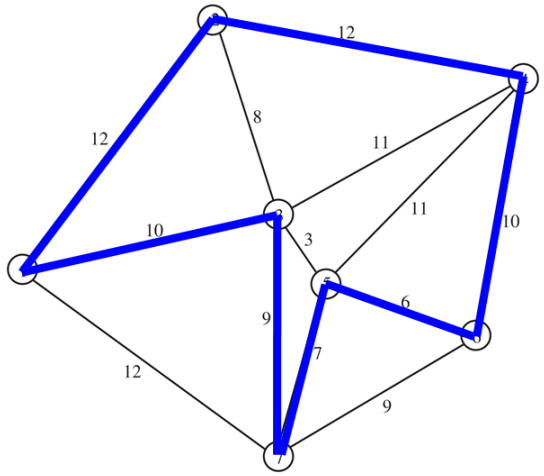


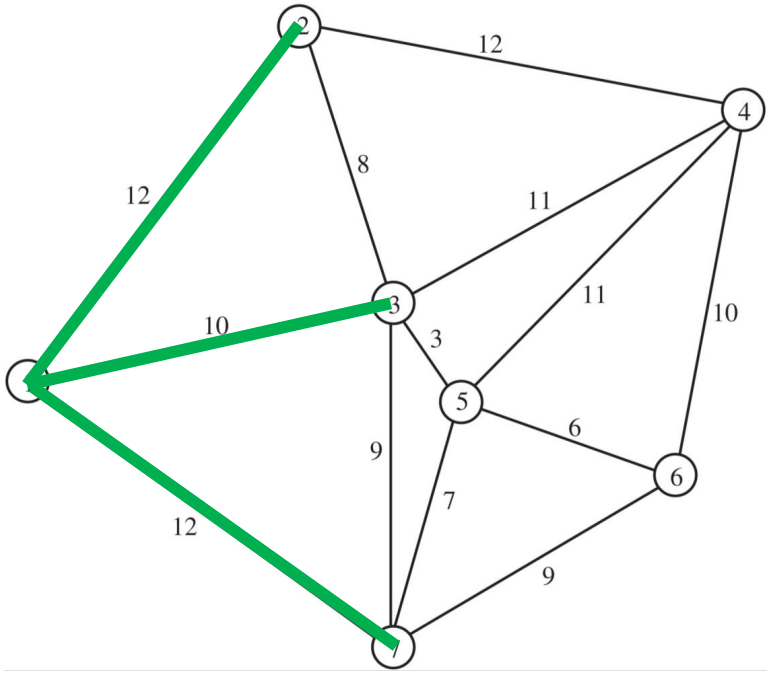




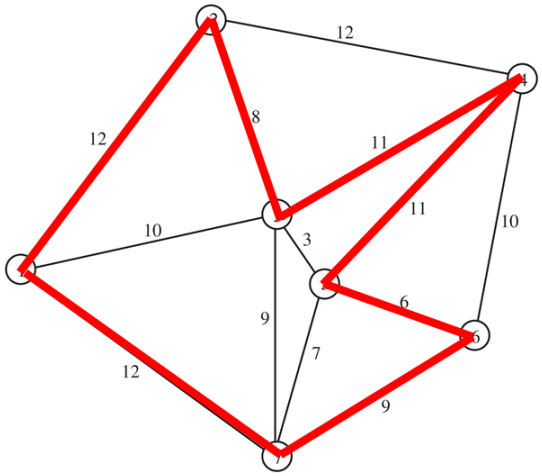
Inheritance

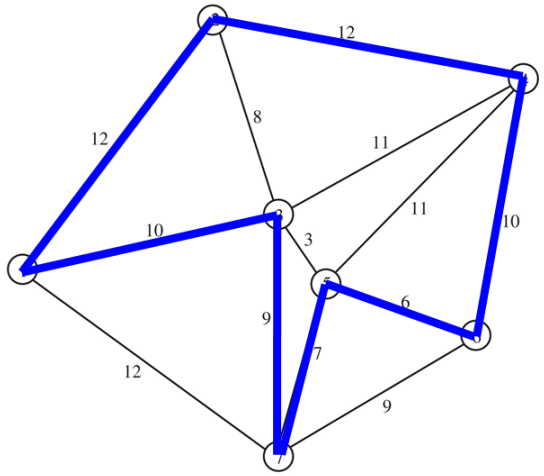


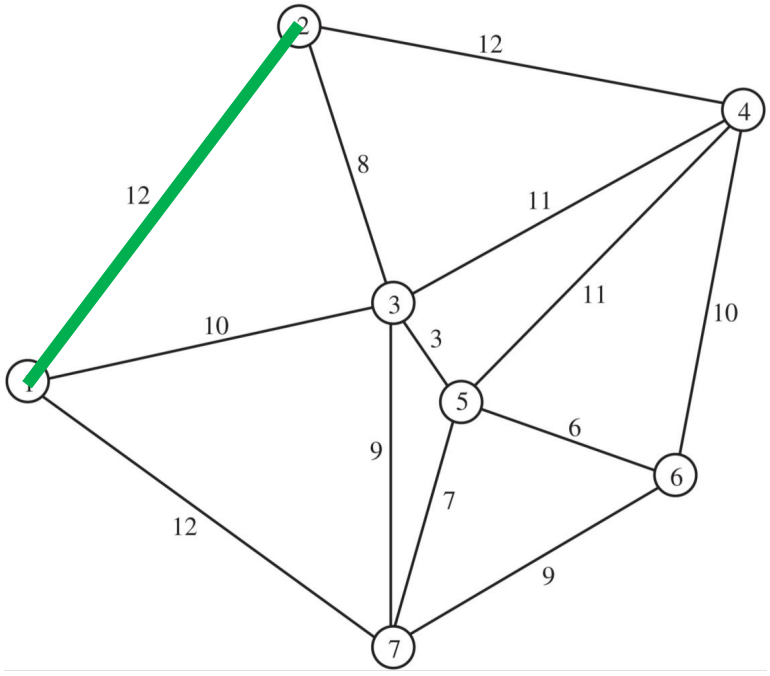




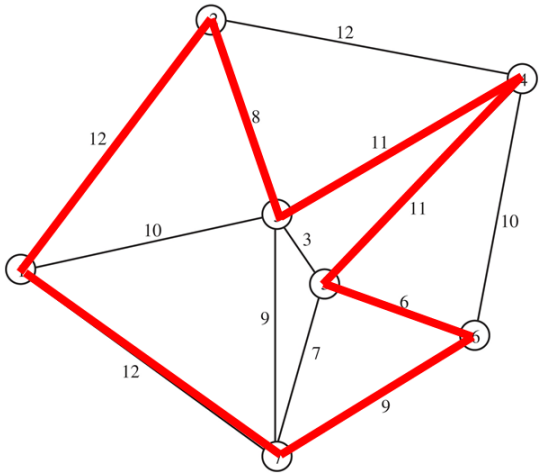
Inheritance

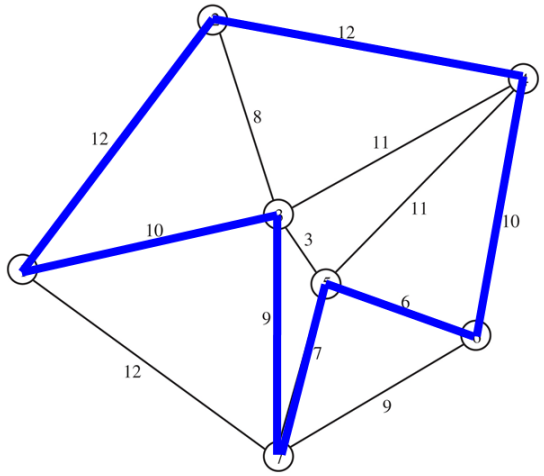


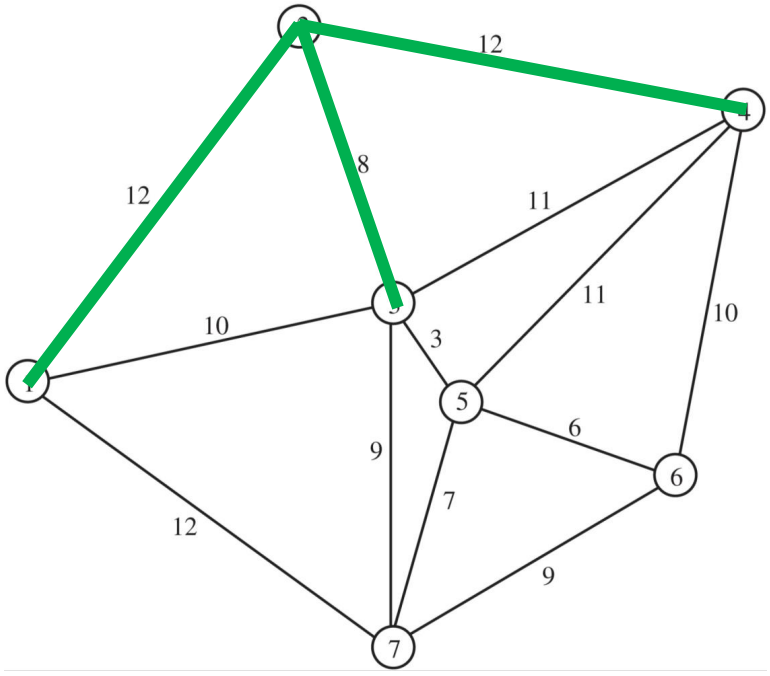




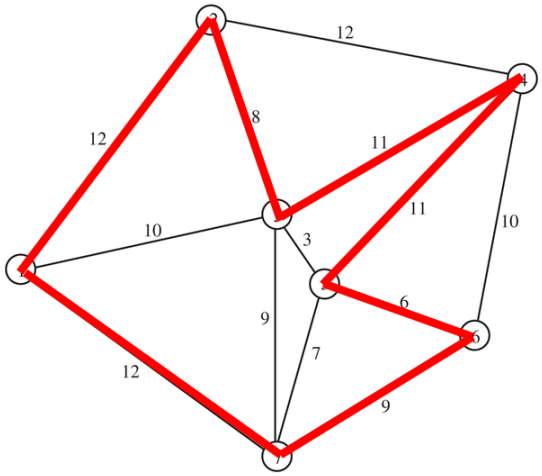
Inheritance

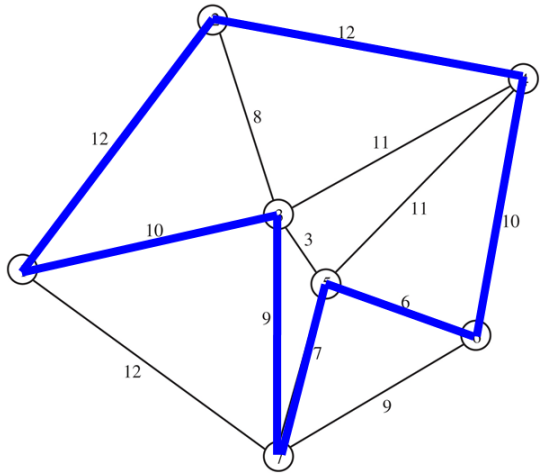


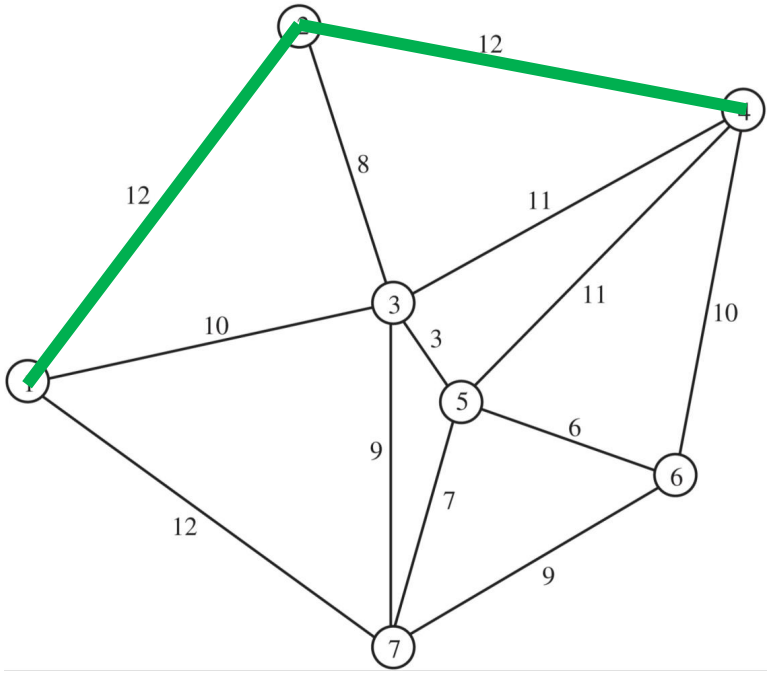




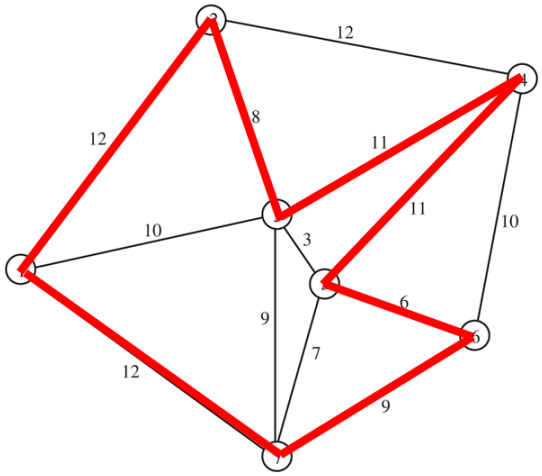
Inheritance

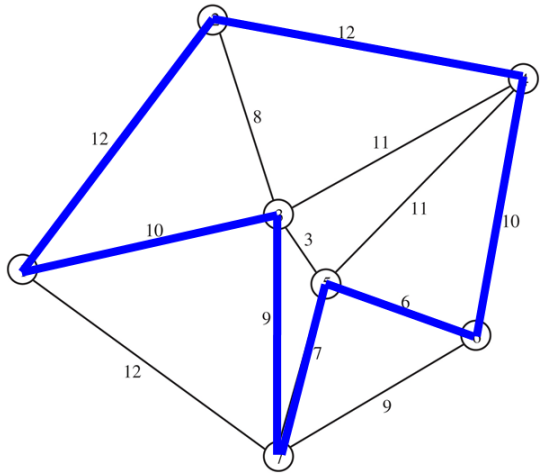


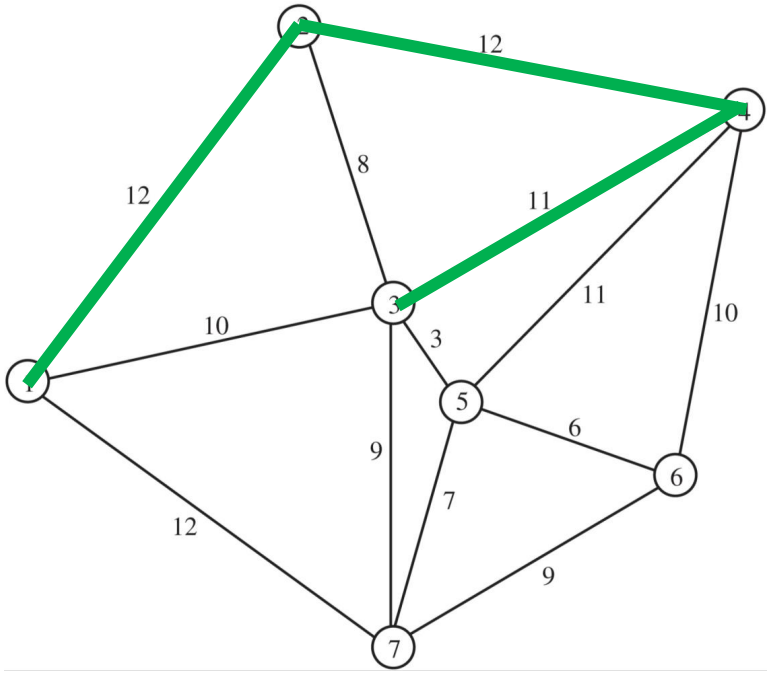




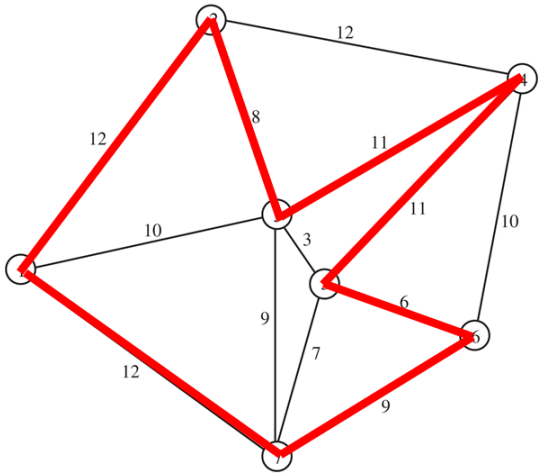
Inheritance

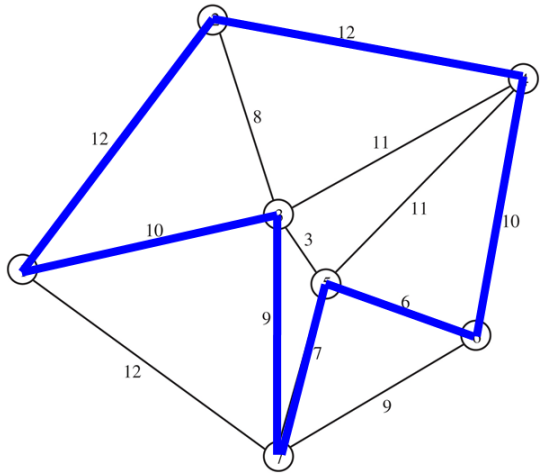


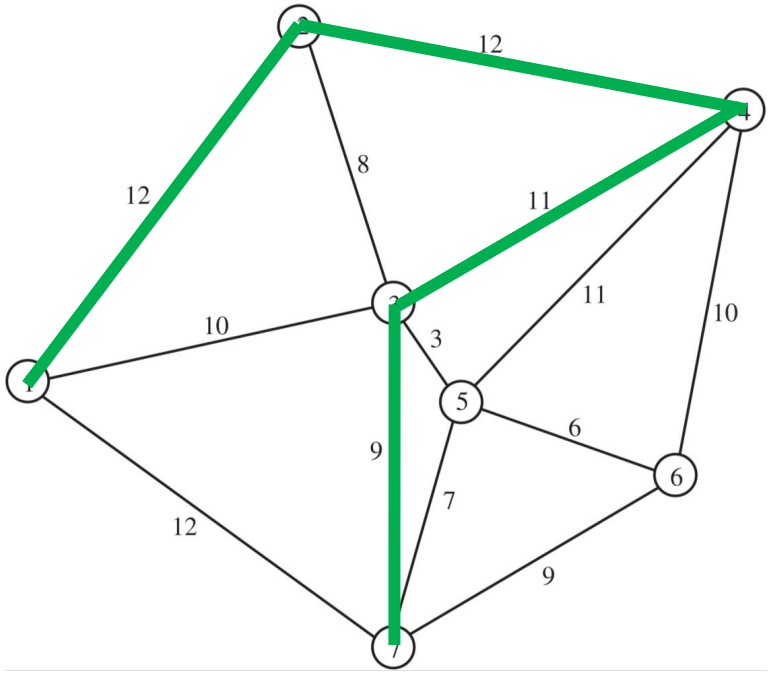




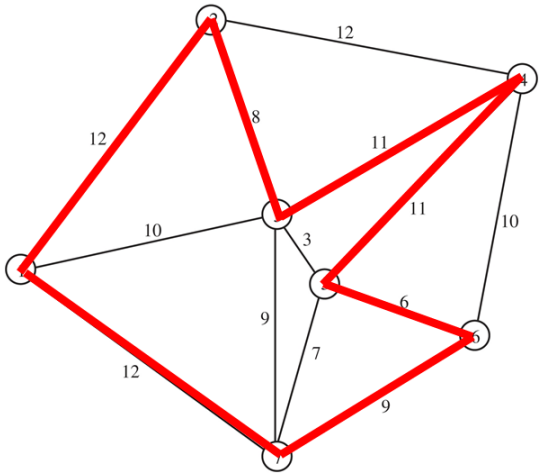
Inheritance

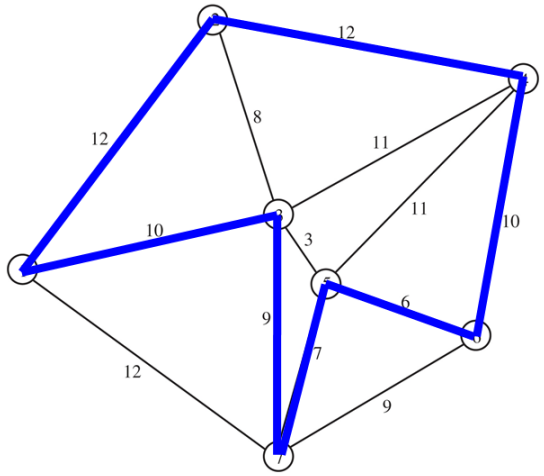


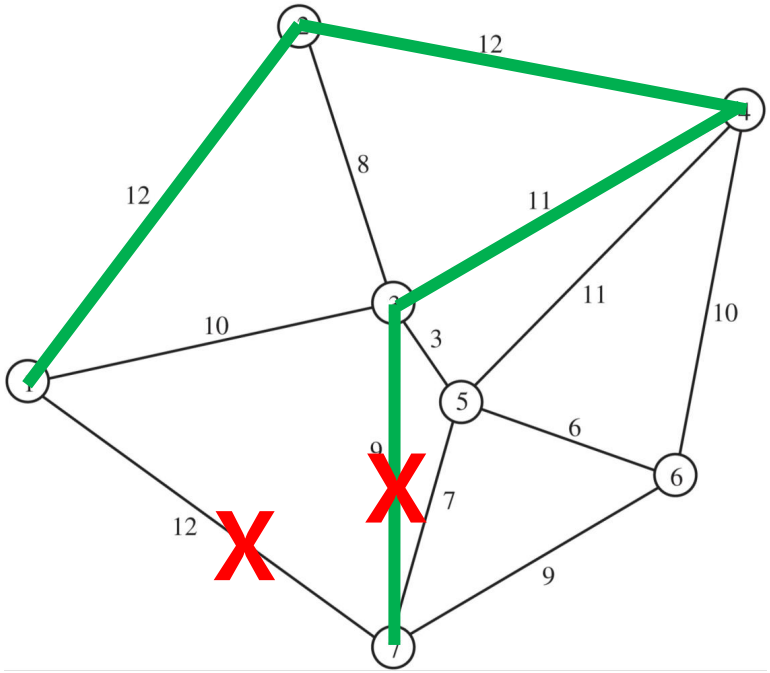




Inheritance

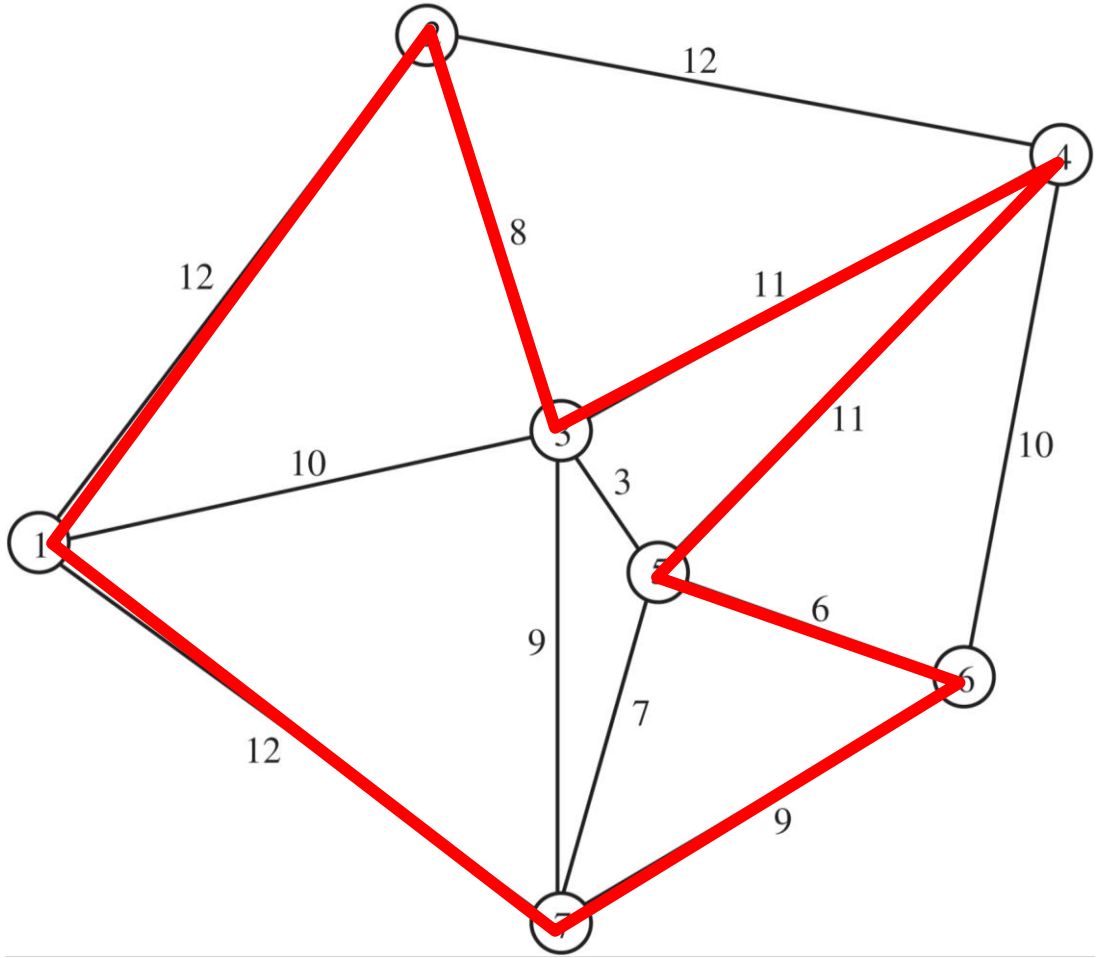






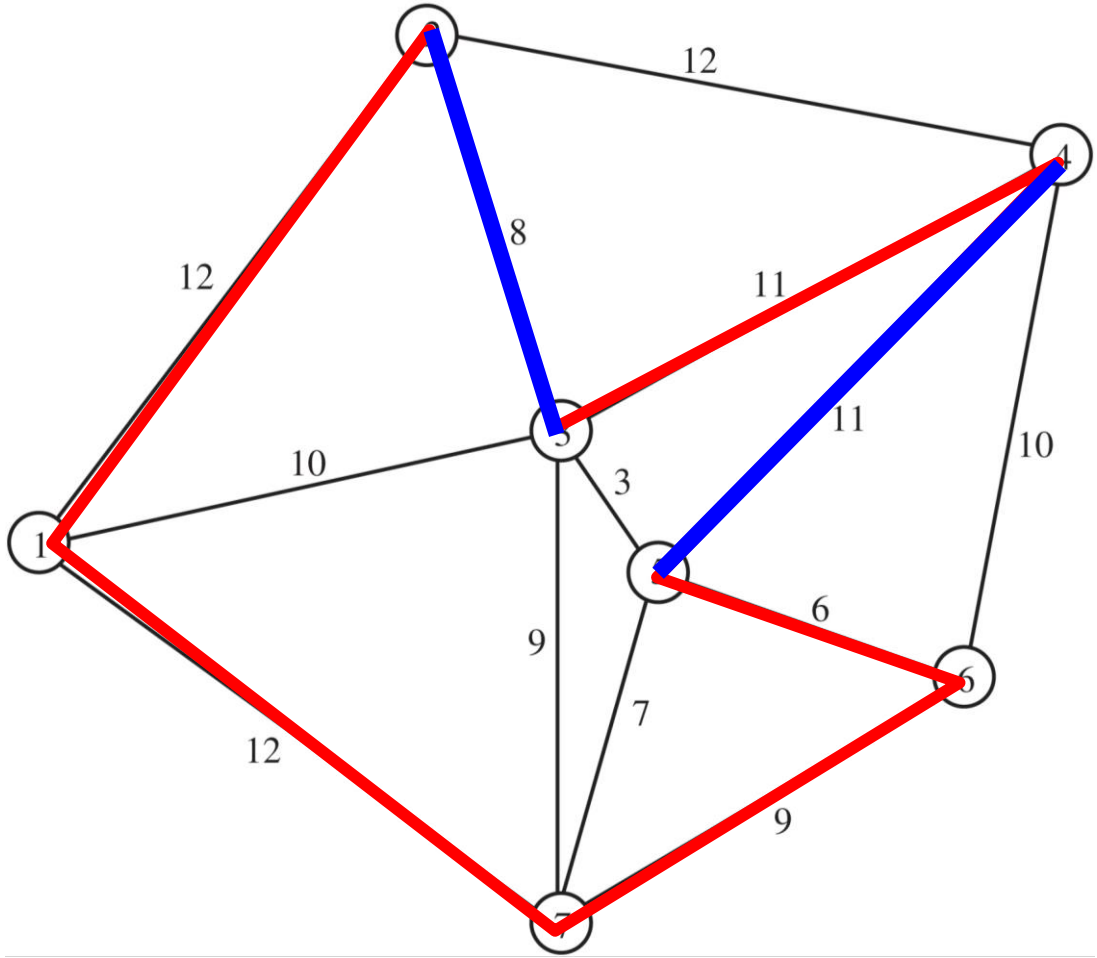
2-Opt

69



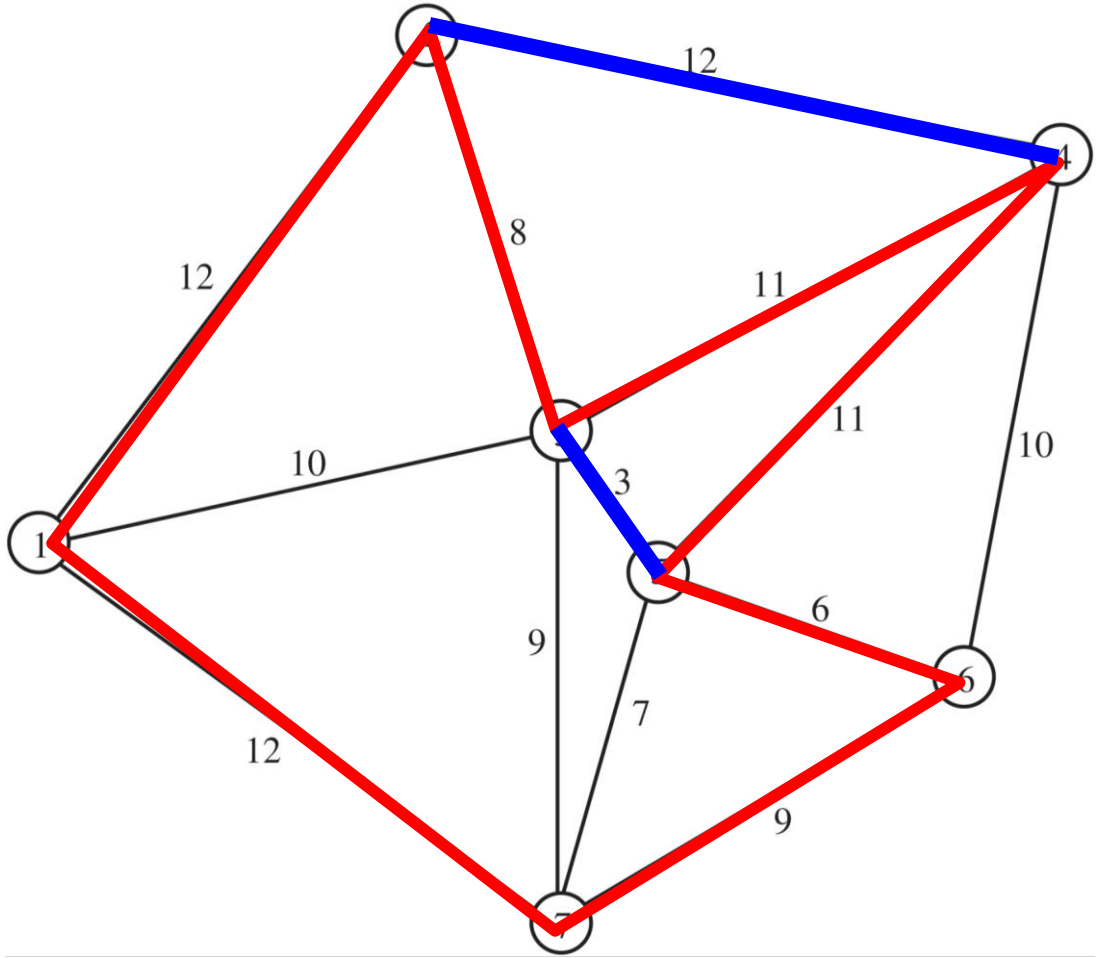
2-Opt

69



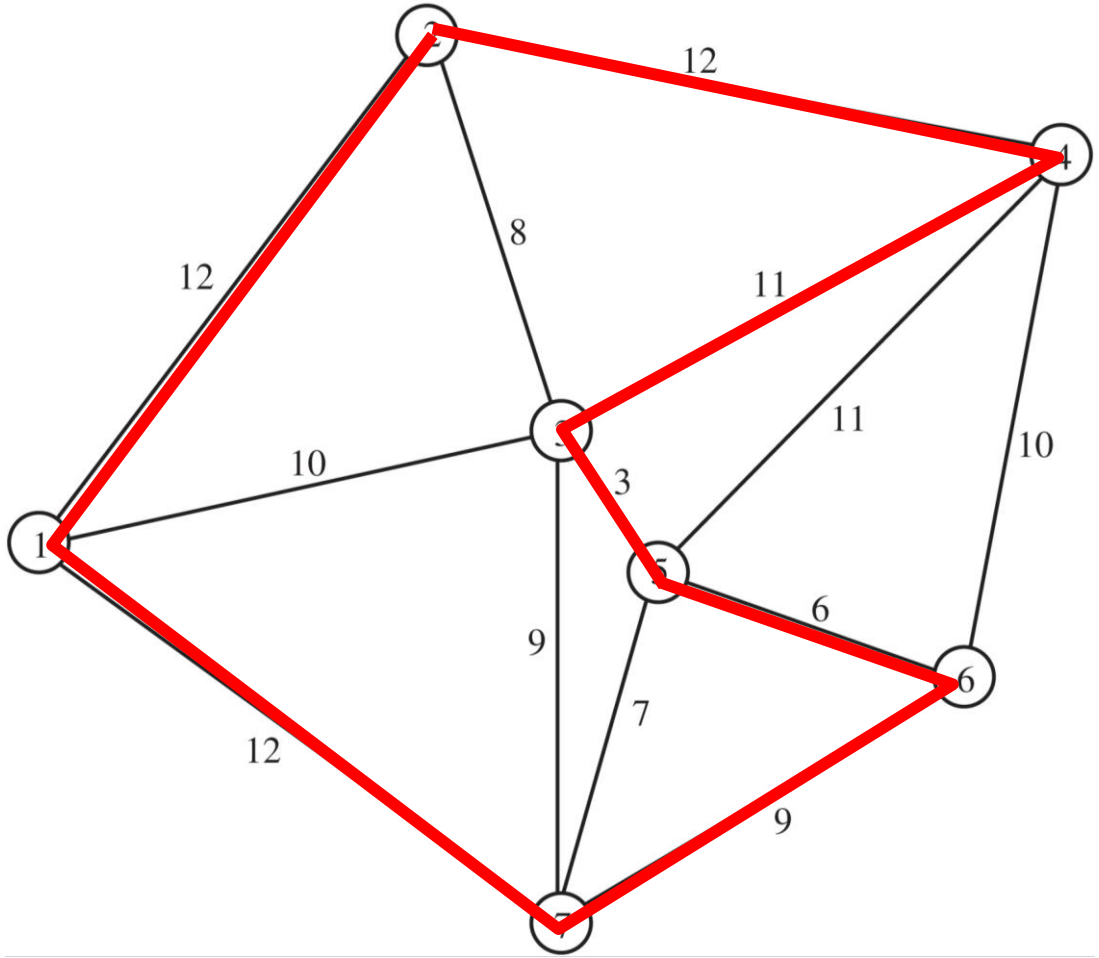
2-Opt

65

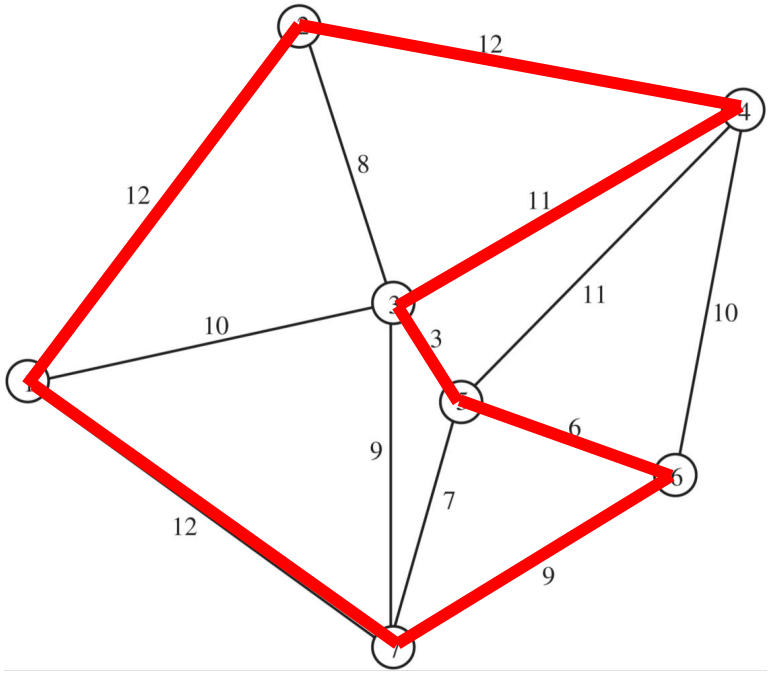


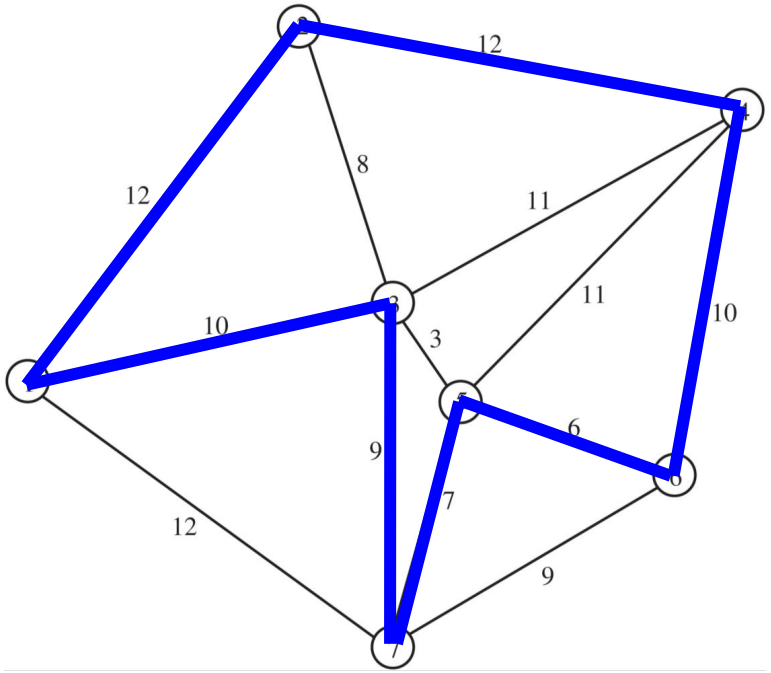
2-Opt

65

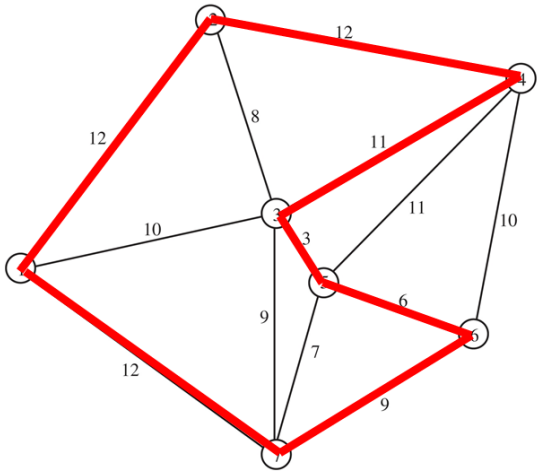


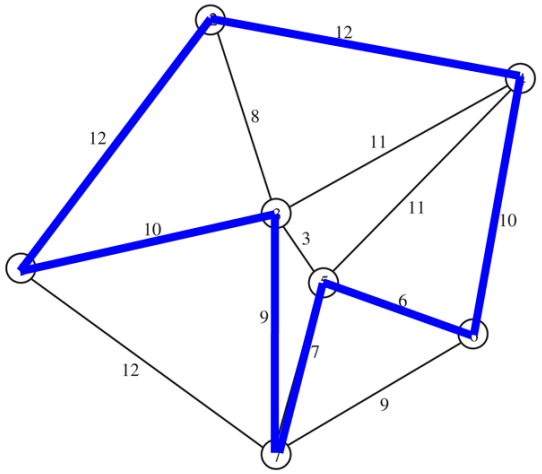
Parents Edited

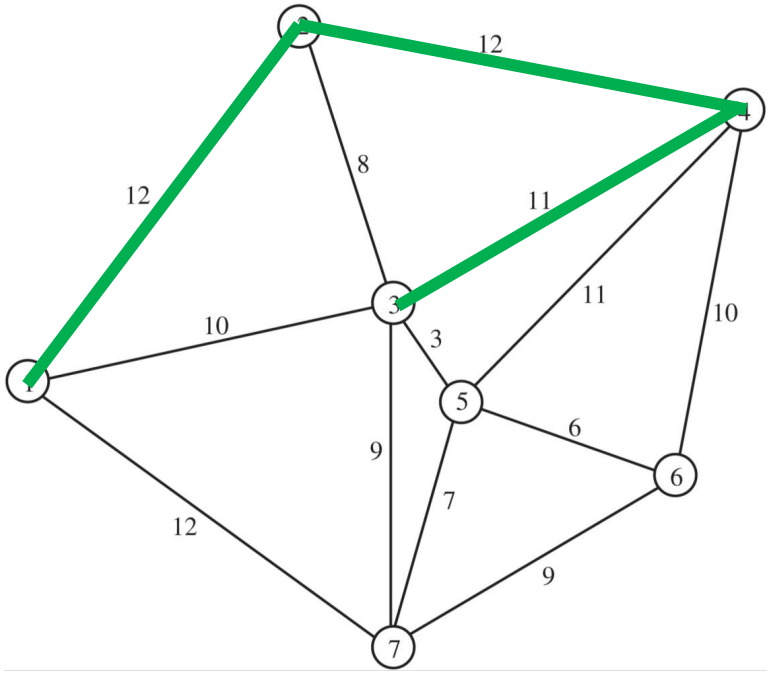




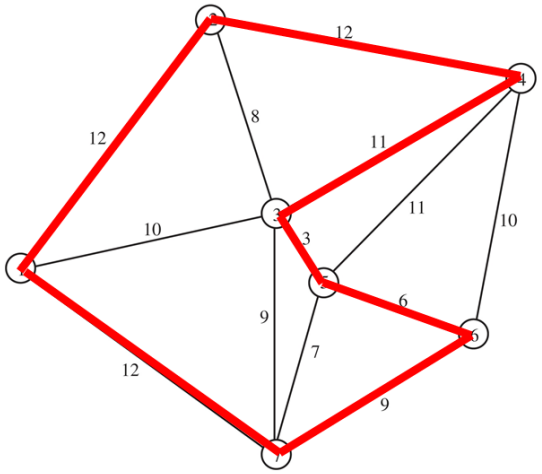
Inheritance

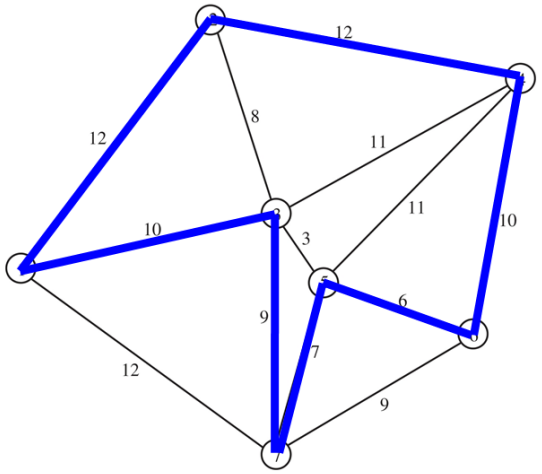


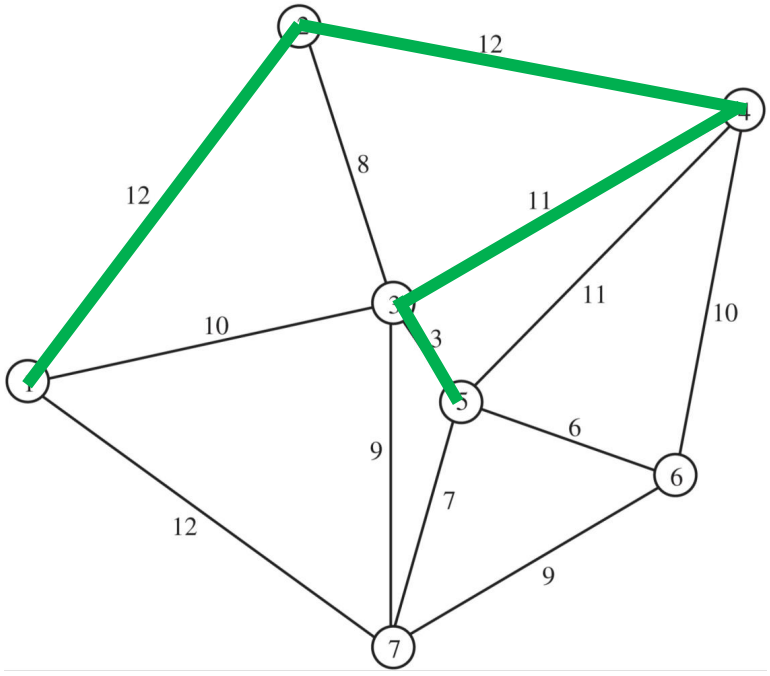




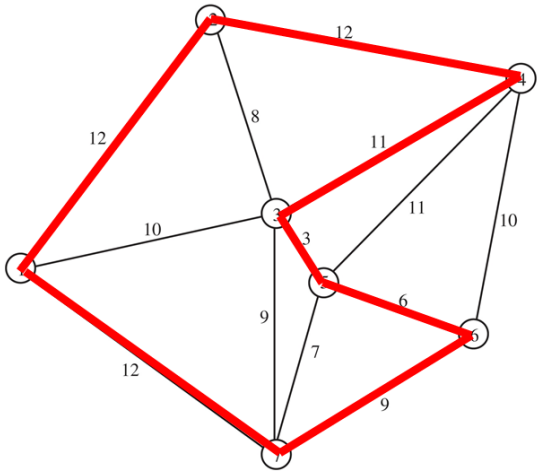
Inheritance

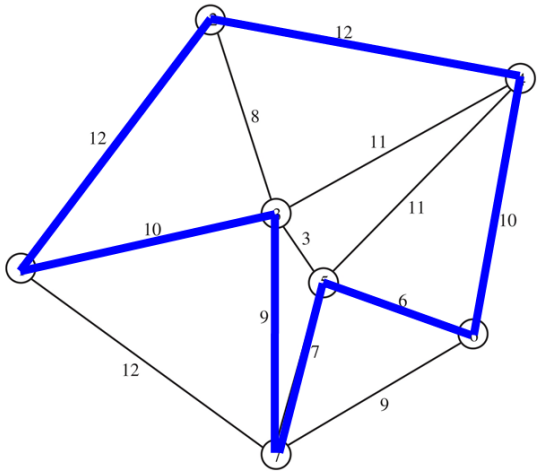


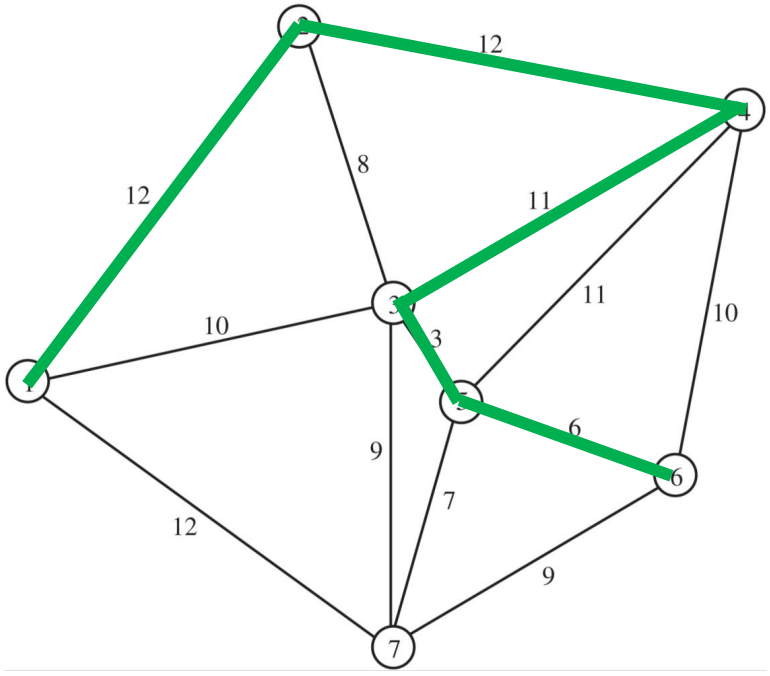




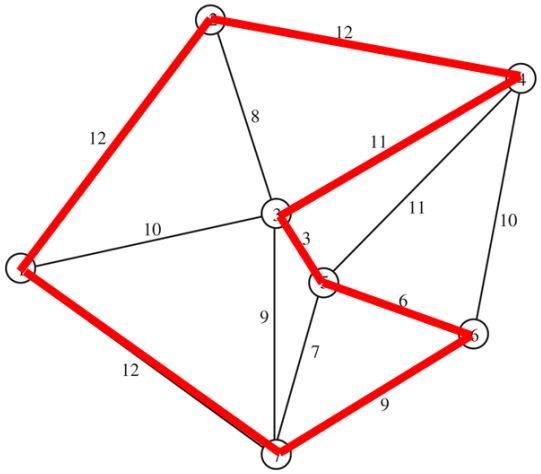
Inheritance

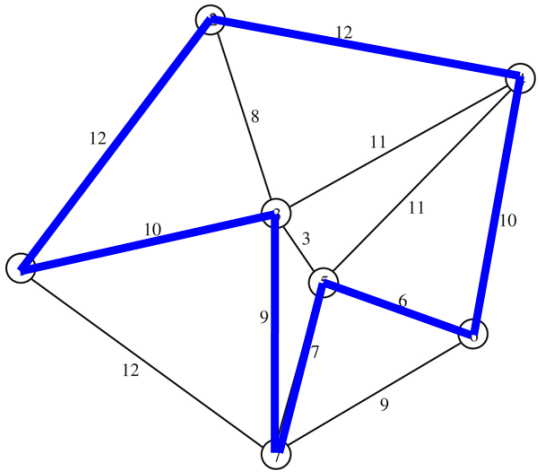


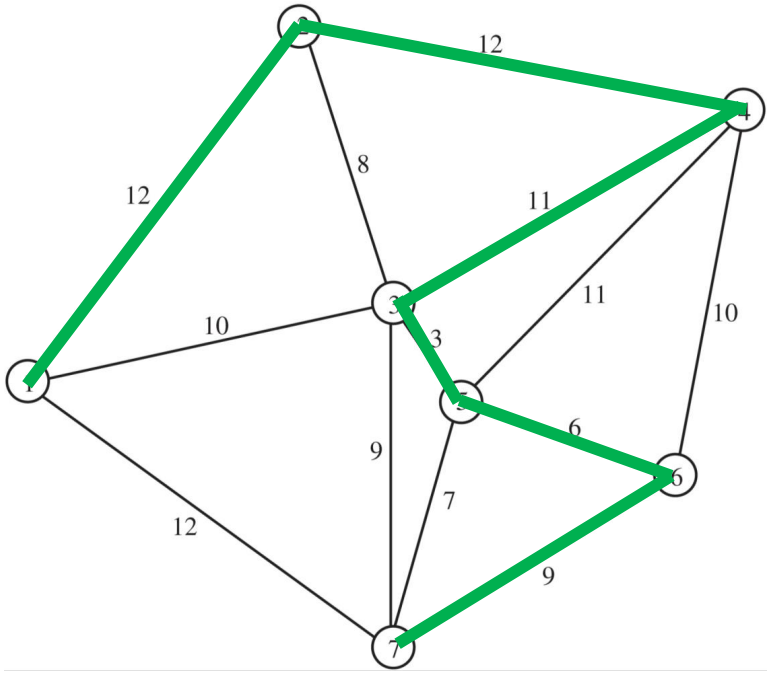




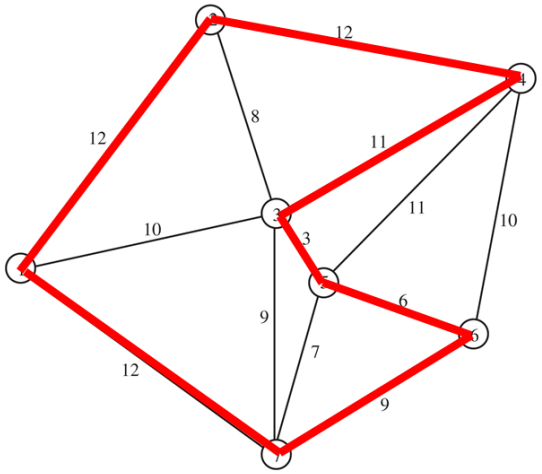
Inheritance

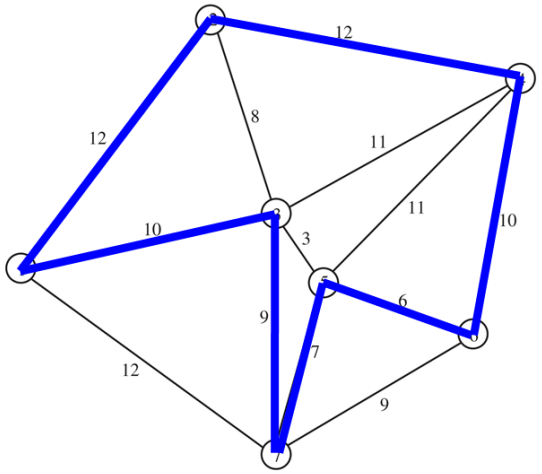


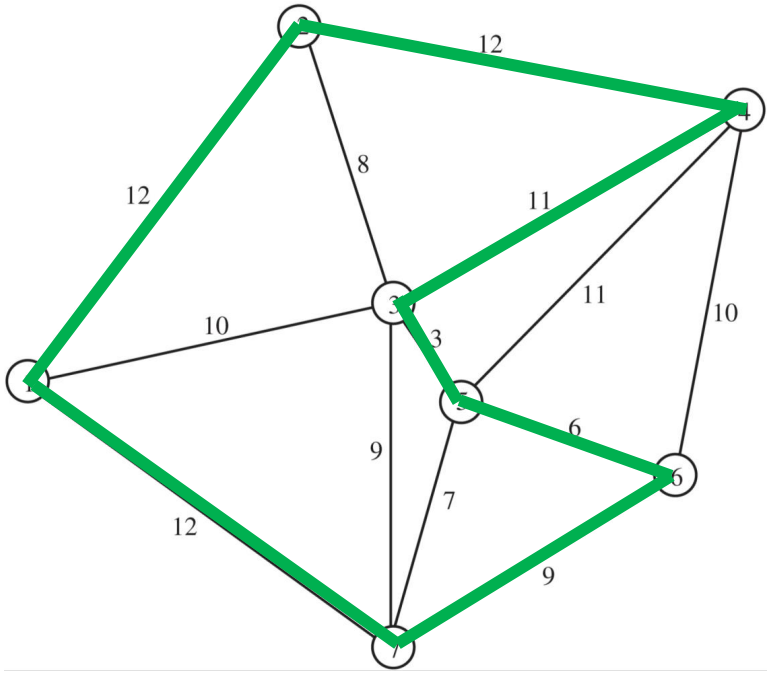




Inheritance







Let's Solve a TSP Problem with GA

Parent P1: **1-2-3-4-5-6-7-1**
Parent P2: **1-2-4-6-5-7-3-1**

Link	Options	Random Selection	Tour
1	1-2, 1-7, 1-2, 1-3	1-2	1-2
2	2-3, 2-4	2-4	1-2-4
3	4-3, 4-5, 4-6	4-3	1-2-4-3
4	3-5*, 3-7	3-5*	1-2-4-3-5
5	5-6, 5-6, 5-7	5-6	1-2-4-3-5-6
6	6-7	6-7	1-2-4-3-5-6-7
7	7-1	7-1	1-2-4-3-5-6-7-1

*A link that completes a sub-tour reversal

Let's Solve a TSP Problem with GA

- Procedure for generating a child
  - Options for the next link
    - All links from the current city to another city not already in the child's tour used by either parent in either direction
    - Add any link needed to complete a sub-tour reversal of the child's tour in a portion of a parent's tour
  - Use a random number to select between link options
  - Check for mutation: If the next random number is less than 0.10, a mutation occurs, and link selected in the last step is rejected
  - Continuation: Add the selected link to the end of the child's current tour and start from the added city
  - Iterate until all cities but one have been visited
  - Add link to the final remaining city

GA Example

Member	Initial Population	Fitness (Distance)
1	1-2-4-6-5-3-7-1	64
2	1-2-3-5-4-6-7-1	65
3	1-7-5-6-4-2-3-1	65
4	1-2-4-6-5-3-7-1	64
5	1-3-7-5-6-4-2-1	66
6	1-2-4-6-5-3-7-1	64
7	1-7-6-4-5-3-2-1	65
8	1-3-7-6-5-4-2-1	69
9	1-7-6-4-5-3-2-1	65
10	1-2-4-6-5-3-7-1	64

GA Example

Member	Initial Population	Fitness (Distance)
1	1-2-4-6-5-3-7-1	64
2	1-2-3-5-4-6-7-1	65
3	1-7-5-6-4-2-3-1	65
4	1-2-4-6-5-3-7-1	64
5	1-3-7-5-6-4-2-1	66
6	1-2-4-6-5-3-7-1	64
7	1-7-6-4-5-3-2-1	65
8	1-3-7-6-5-4-2-1	69
9	1-7-6-4-5-3-2-1	65
10	1-2-4-6-5-3-7-1	64

GA Example

Member	Initial Population	Fitness (Distance)
1	1-2-4-6-5-3-7-1	64
2	1-2-3-5-4-6-7-1	65
3	1-7-5-6-4-2-3-1	65
4	1-2-4-6-5-3-7-1	64
5	1-3-7-5-6-4-2-1	66
6	1-2-4-6-5-3-7-1	64
7	1-7-6-4-5-3-2-1	65
8	1-3-7-6-5-4-2-1	69
9	1-7-6-4-5-3-2-1	65
10	1-2-4-6-5-3-7-1	64

Member	Parents	Children	Member	Fitness (Distance)
1	1-2-4-6-5-3-7-1	1-2-4-5-6-7-3-1	11	69
7	1-7-6-4-5-3-2-1	1-2-4-6-5-3-7-1	12	64
2	1-2-3-5-4-6-7-1	1-2-4-5-6-7-3-1	13	69
6	1-2-4-6-5-3-7-1	1-7-6-4-5-3-2-1	14	65
4	1-2-4-6-5-3-7-1	1-2-4-6-5-3-7-1	15	64
5	1-3-7-5-6-4-2-1	1-3-7-5-6-4-2-1	16	66

GA Example

Member	Initial Population	Fitness (Distance)
1	1-2-4-6-5-3-7-1	64
2	1-2-3-5-4-6-7-1	65
3	1-7-5-6-4-2-3-1	65
4	1-2-4-6-5-3-7-1	64
5	1-3-7-5-6-4-2-1	66
6	1-2-4-6-5-3-7-1	64
7	1-7-6-4-5-3-2-1	65
8	1-3-7-6-5-4-2-1	69
9	1-7-6-4-5-3-2-1	65
10	1-2-4-6-5-3-7-1	64

Member	Parents	Children	Member	Fitness (Distance)
1	1-2-4-6-5-3-7-1	1-2-4-5-6-7-3-1	11	69
7	1-7-6-4-5-3-2-1	1-2-4-6-5-3-7-1	12	64
2	1-2-3-5-4-6-7-1	1-2-4-5-6-7-3-1	13	69
6	1-2-4-6-5-3-7-1	1-7-6-4-5-3-2-1	14	65
4	1-2-4-6-5-3-7-1	1-2-4-6-5-3-7-1	15	64
5	1-3-7-5-6-4-2-1	1-3-7-5-6-4-2-1	16	66

GA Example

Member	Initial Population	Fitness (Distance)
1	1-2-4-6-5-3-7-1	64
2	1-2-3-5-4-6-7-1	65
3	1-7-5-6-4-2-3-1	65
4	1-2-4-6-5-3-7-1	64
5	1-3-7-5-6-4-2-1	66
6	1-2-4-6-5-3-7-1	64
7	1-7-6-4-5-3-2-1	65
8	1-3-7-6-5-4-2-1	69
9	1-7-6-4-5-3-2-1	65
10	1-2-4-6-5-3-7-1	64

Member	Parents	Children	Member	Fitness (Distance)
1	1-2-4-6-5-3-7-1	1-2-4-5-6-7-3-1	11	69
7	1-7-6-4-5-3-2-1	1-2-4-6-5-3-7-1	12	64
2	1-2-3-5-4-6-7-1	1-2-4-5-6-7-3-1	13	69
6	1-2-4-6-5-3-7-1	1-7-6-4-5-3-2-1	14	65
4	1-2-4-6-5-3-7-1	1-2-4-6-7-5-3-1	15'	63
5	1-3-7-5-6-4-2-1	1-3-7-5-6-4-2-1	16	66

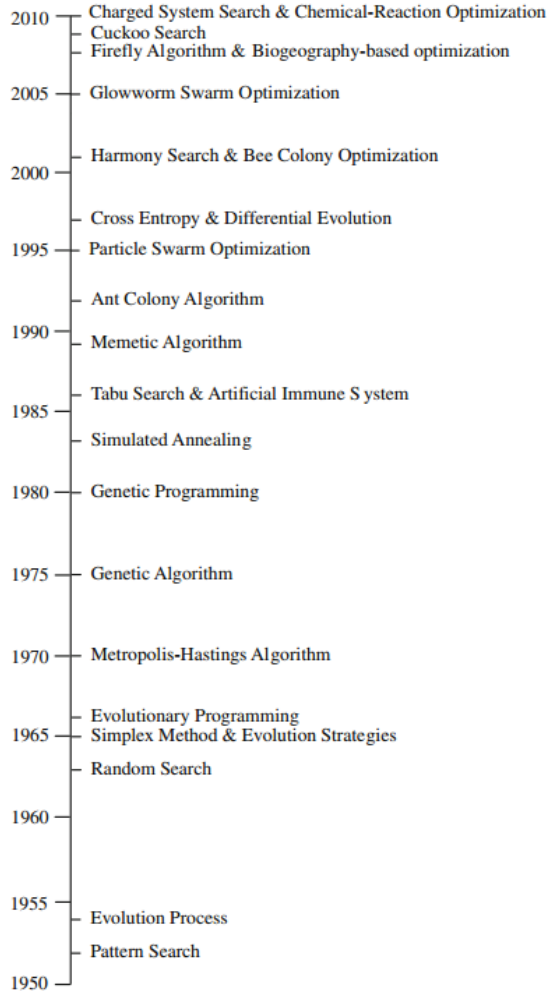
Think About

- Encoding method
- Initialization method
- Population size
- Fitness evaluation function
- Parents selection method
- Inheritance method
- Mutation method
- Parameters for the methods
- Homogeneity of population
- Etc.

Other Metaheuristics (Computational Intelligence) Algorithms: See the Appendix

- Higher-level procedure or heuristic designed to find a sufficiently good solution

- Often inspired from nature or society
  - Genetic Algorithm
  - Tabu Search
  - Simulated Annealing
  - Ant Colony Algorithm
  - Particle Swarm Optimization
  - Cuckoo Search Algorithm
  - …



Assignments 7 and 6

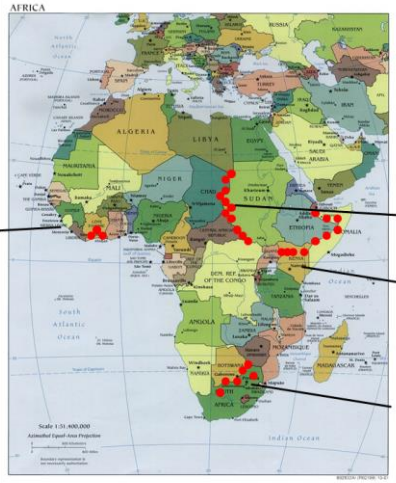
Assignment 7 (by 11.18 11:59 pm)

- (0) Based on the practice demonstration by TA Yoon, complete the Assignment 6 (i.e., Optimize the production process and describe the results - identify a set of optimal control values). First, based on a linear regression model, use the scipy package for the mathematical optimization. Second, based on a non-linear regression model (e.g., tree-based models, neural network models), complete the provided GA code and use it for the optimization through simulation (refer to the next week practice); of course you can develop your own heuristic optimization algorithm if you want.
- (1) Select more than 9 nodes in the UNIST map you would like to introduce to a visitor. Connect the nodes, estimate the distance between nodes, and represent the network similar to the case introduced in the class. Find the optimal route, using a genetic algorithm or your own heuristic.
- (2) Identify a real-world service optimization problem that you are interested or concerned, and develop a mathematical model of this problem, based on existing reference models and your own creativity. Describe the importance of your problem and model in detail.
- (3) Develop your own heuristic algorithm to solve this problem. Your algorithm should reflect a mechanism “how we, humans make decisions”. An idea level is fine. Try to think long and propose your new algorithm concisely. Describe in detail the algorithm in a format of flow chart, pseudocode, or your own visualization. If possible, try to identify and use a good mathematical model for your conceptual basis. For your reference, you may want to study a metaheuristic algorithm besides GA that is most interesting to you.

Per the Question (1), Consider the Current Map of UNIST

Per the Question (1), Consider the Future Map of UNIST

Per the Questions (2) and (3), an Example is…



Sierra Leone refugee camp

Darfur refugee camp

Kenya Kakuma refugee camp

Republic of South Africa refugee camp

Parameters

d_i	demand of area i
c_j	capacity of relief goods distribution center j
B	budget for relief goods support
F_j	fixed cost of establishing center j
m	unit cost of relief good supported
I_i	interference factor to demand i (not fixed, randomly change)
I_j	interference factor to center j (not fixed, randomly change)
f_{ij}	proportion of demand i satisfied by center j
p_i	probability of occurrence of demand i
p_j	probability of construction of camp j
h_j	portion of worker needed to camp j
P	total number of workers needed for relief work
TP	threshold of probability
TW	threshold of worker number

Decision variable

x_j {1 If distribution center j is constructed, 1. Otherwise, 0

y_i {1 If demand of area i is originated, 1. Otherwise, 0

The formulation for the problem is as follows:

Maximize $\sum_i \sum_j x_j y_i f_{ij} p_i p_j h_j d_i I_i$ (1)

Subject to

$\sum_j x_j \geq 1$ (2)

$\sum_j F_j x_j + m \sum_j c_j x_j \leq B$ (3)

$\sum_i f_{ij} d_i x_j \leq c_j I_j$ (4)

If $p_i \geq TP$, then p_i becomes 1, otherwise 0 (5)

If $p_j \geq TP$, then p_j becomes 1, otherwise 0 (6)

If $h_j \geq TW$ h_j becomes 1, otherwise 0 (7)

Assignment 7 (by 11.18 11:59 pm)

- (0) Based on the practice demonstration by TA Yoon, complete the Assignment 6 (i.e., Optimize the production process and describe the results - identify a set of optimal control values). First, based on a linear regression model, use the scipy package for the mathematical optimization. Second, based on a non-linear regression model (e.g., tree-based models, neural network models), complete the provided GA code and use it for the optimization through simulation (refer to the next week practice); of course you can develop your own heuristic optimization algorithm if you want.
- (1) Select more than 9 nodes in the UNIST map you would like to introduce to a visitor. Connect the nodes, estimate the distance between nodes, and represent the network similar to the case introduced in the class. Find the optimal route, using a genetic algorithm or your own heuristic.
- (2) Identify a real-world service optimization problem that you are interested or concerned, and develop a mathematical model of this problem, based on existing reference models and your own creativity. Describe the importance of your problem and model in detail.
- (3) Develop your own heuristic algorithm to solve this problem. Your algorithm should reflect a mechanism “how we, humans make decisions”. An idea level is fine. Try to think long and propose your new algorithm concisely. Describe in detail the algorithm in a format of flow chart, pseudocode, or your own visualization. If possible, try to identify and use a good mathematical model for your conceptual basis. For your reference, you may want to study a metaheuristic algorithm besides GA that is most interesting to you.

Assignment 6 (by 11.18 11:59 pm)

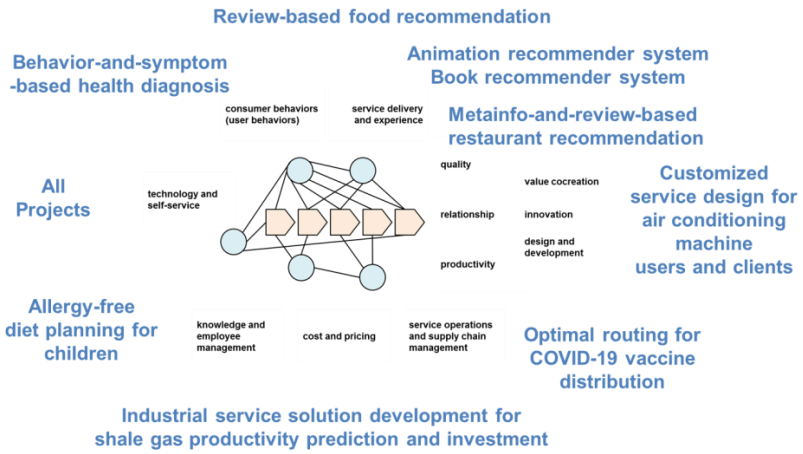
- Based on the practice demonstration by TA Cho, (1) complete the development of a yield prediction model for the sugar manufacturer by yourself. Through trials and tests, develop your own best prediction model. You should compare and interpret multiple prediction models.
- (2) In the prediction model development, think carefully about the controllable variables. You should analyze the variables around the process based on your own descriptive and predictive analyses. For example, interpret the analytics outcomes (e.g., describe the controllable variables you identified significant, interpret their coefficient/importance values in your yield prediction models). As a result, describe what controllable variables should be prioritized in the optimization of the process.
- (3) Optimize the production process and describe the results (e.g., a set of optimal control values). First, based on a linear regression model, use the scipy package for the mathematical optimization. Second, based on a non-linear regression model (e.g., tree-based models, neural network models), complete the provided GA code and use it for the optimization through simulation (refer to the next week practice); of course you can develop your own heuristic optimization algorithm if you want.
- (4) Assume you actually need to use your machine for the sugar manufacturer. Using the finally selected prediction model and your optimizer, think how to manage effectively and improve the sugar production process. Design and develop your own industrial service intelligence solution for this manufacturer (e.g., develop an automated prediction-optimization code package). You must provide visualization contents (e.g., visualization of the predicted values of yield flow, visualization of suggested optimal control values for specific controllable variables) Describe your service intelligence solution in detail. Think beyond the class examples in your own creative, unique way!
- (5) Think about your concerned industrial/business service around UNIST, in Ulsan, in your hometown, or any other interested service that require a machine for its management and improvement. Describe the specific tasks of the service that require the support of a machine. Discuss the requirements of developing such a machine for the service in detail.
- (6) If you would actually conduct a study on developing a machine for the industrial/business service, how would you conduct the research in your own creative, unique way? What kinds of data are you going to collect, analyze, and learn, and what methods are you going to use? Describe your service intelligence development plan in detail. If possible, visualize your plan clearly (e.g., draw an image, construct a mathematical model). To facilitate your thinking, you may want to identify and review a paper or any other reference in the Internet, related to the service you are interested or concerned.
- Upload your code and a several paragraph essay on the tasks (1)~(6) in the Blackboard.

Term Project Announcement:

“Develop Your Own Service Intelligence”

Term Project

- By yourself or with a team member, suggest a term project topic that you would like to solve an important real-world service problem by developing your own service intelligence. Or, you can participate in the following competition and develop a service intelligence that supports the decision making of credit card users.

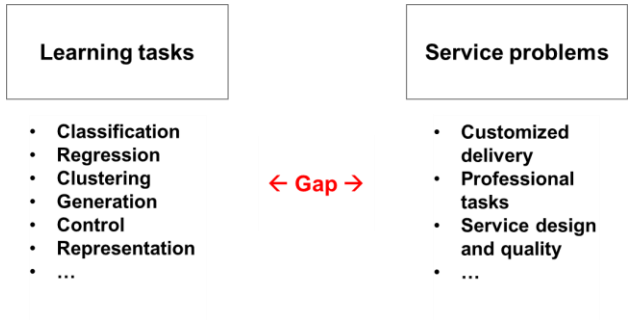


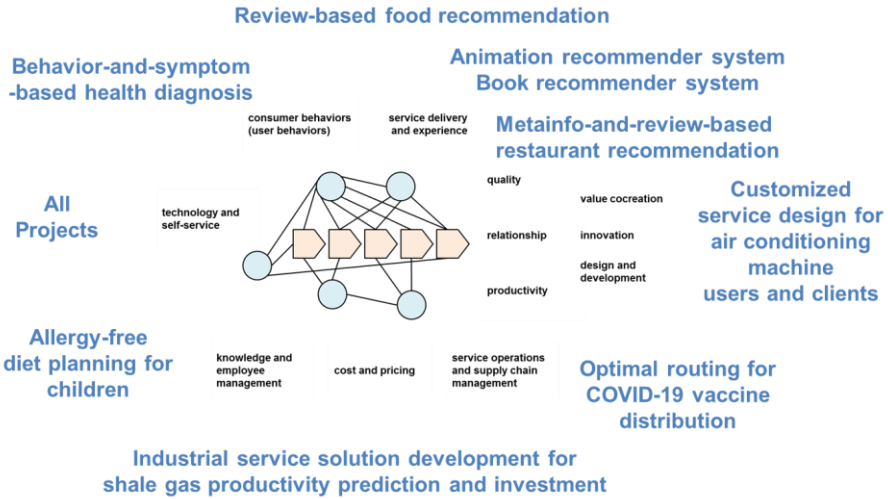
Hana Financial Group

- Review your previous assignment outcomes. Identify and describe the rooms for improvement. You do not have to redo the assignments. Just try how to improve them.

Guideline

- Topic selection
  - Type 1: Application of an existing theory or method to an important problem in your own way
  - Type 2: New theory or method development for service problems
  - These are just examples. Do whatever research you are interested, relevant to this course





- Important considerations
  - Start from the course material and/or a specific key paper and make your own contribution reasonably
  - Collection and use of real or at least realistic data is mandatory; Validation is important

Term Project Proposals (by 11/6 Sunday 11:59 pm)

- Presentation file upload due by 11/6 Sunday 11:59 pm

- Presentation operations
  - Presentation file: PPT format within 8 pages
  - Presentation content (examples):

    What service?, What problem and task? (i.e., Motivation of the task support, Clear problem definition),

    What data to analyze and learn? How to collect the data? (real-world or at least realistic simulated data required)

    What intelligence to develop? (e.g., learning and optimization for descriptive, predictive, or prescriptive task)

    Related literature and key reference, Project plan, Expected outcome and contribution, etc…
  - Presentation at: 11/7 Mon & 11/9 Wed
  - Presentation order each day: Random

Expected Schedule about the Term Project

- Week 10 11/2 Wed: Computational Intelligence Practice Demo
- Week 11
  - 11/7 Mon: Term Project Proposals I
  - 11/9 Wed: Term Project Proposals II
- Week 12
  - 11/14 Mon: Discussion on the assignment outcomes + Special lecture on an AICP competition win topic
  - 11/16 Wed: Special lectures on a sequential recommendation topic + a customer experience clustering topic
- Week 13
  - 11/21 Mon: Term Project Progress Presentation I
  - 11/23 Wed: Term Project Progress Presentation II

Service Intelligence Week 10.

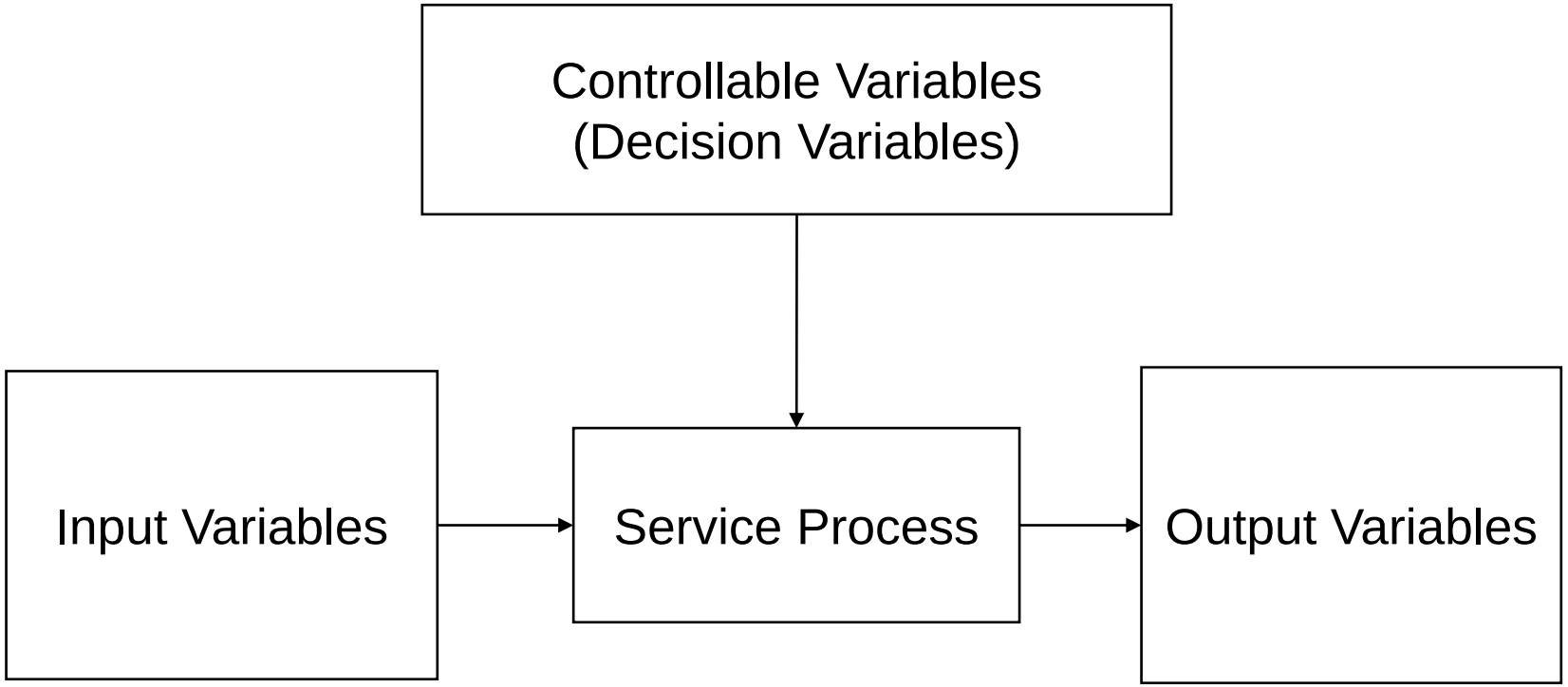
[Service Optimization: Operational Decision Making with Computational Intelligence]

Chiehyeon Lim

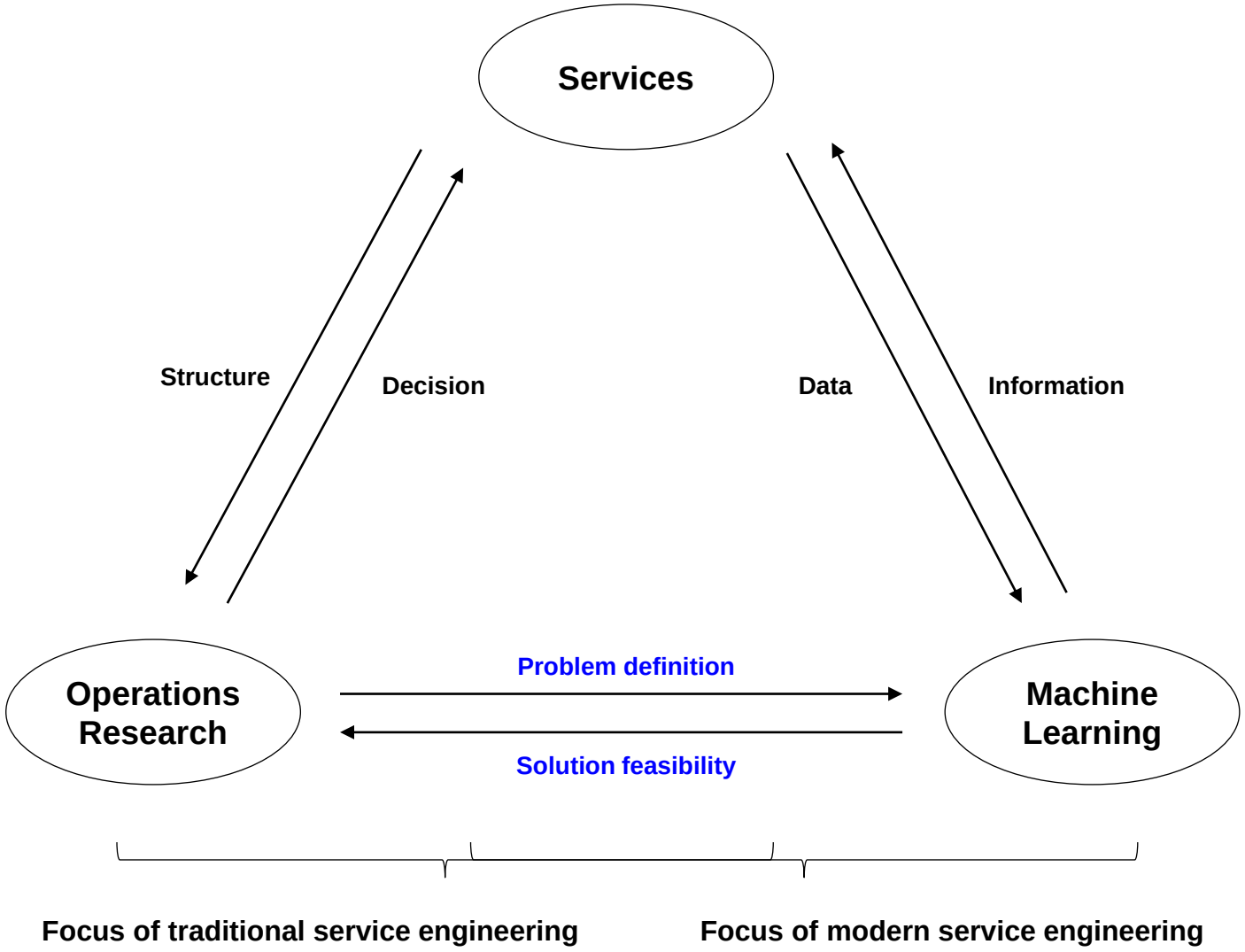
2022. 11. 2

Another Approach for Service Improvement

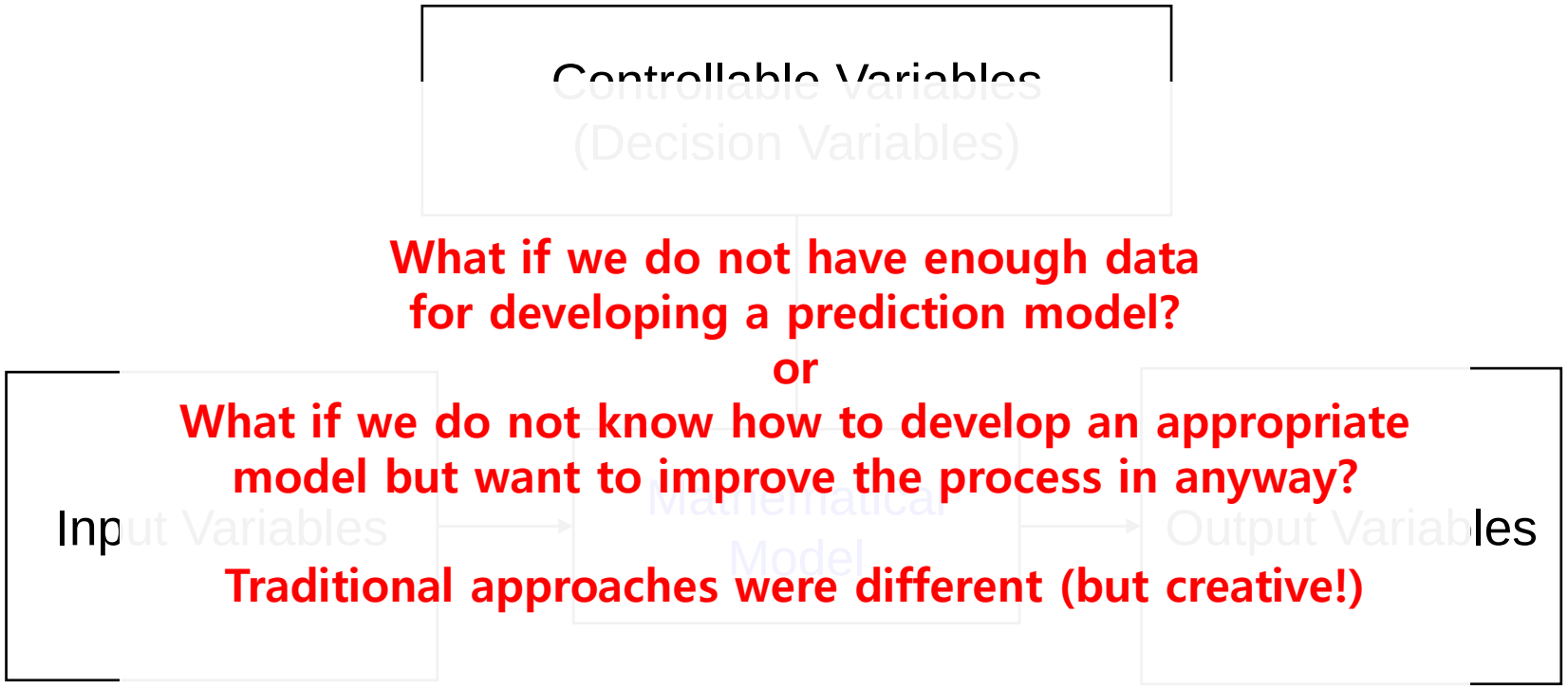
A Framework of Service Improvement



Effective Approach: Operations Research + Machine Learning



A Framework of Service Operations Management

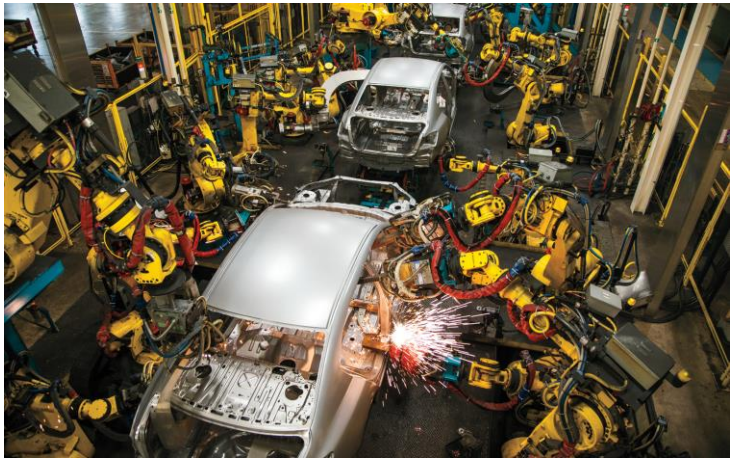


What if we do not have enough data for developing a prediction model?
or
What if we do not know how to develop an appropriate model but want to improve the process in anyway?

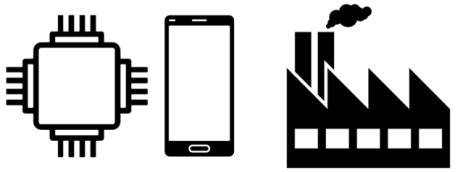
Traditional approaches were different (but creative!)

Data Envelopment Analysis and Its Relevance with the Previous Classes

Processes to be Improved/Optimized

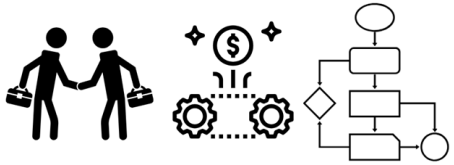


Processes around Us Involve Problems and Need Improvement



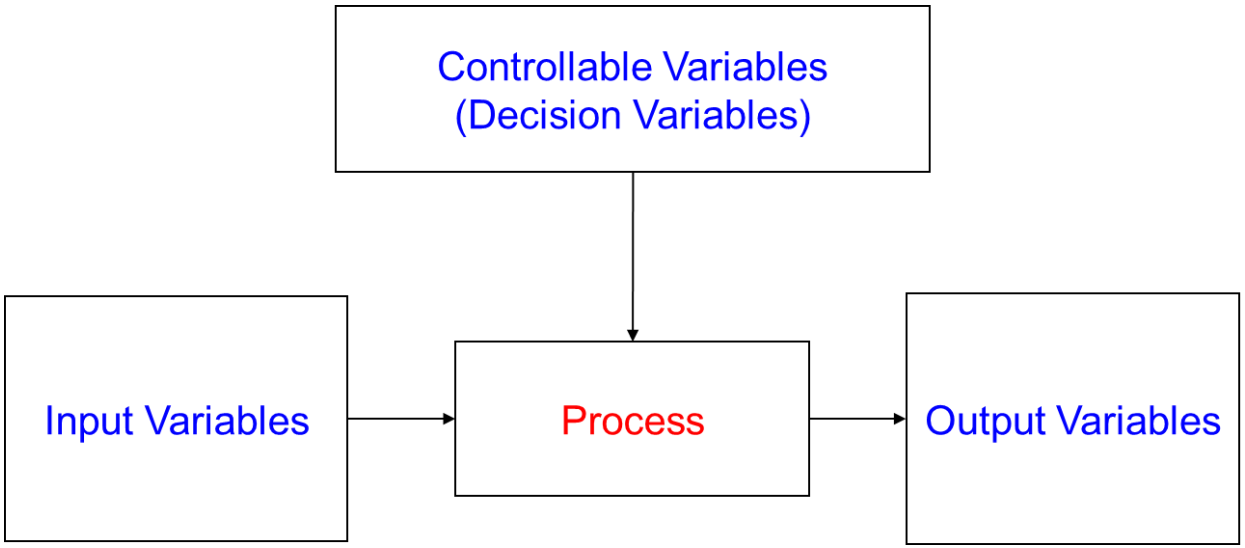
Manufacturing Process Management

vs.



Service Process Management

Personal Process Management



How Can We Evaluate and Improve Processes with the Process Variables?

- Efficiency
  - A measure of the efficient use of inputs for making outputs
  - Usually expressed as the ratio of “multiple” outputs to “multiple” inputs

- Efficiency ratios are used for
  - Planning workforce requirements and resource allocation
  - Financial and economic analysis
  - Benchmarking competitive organizations
  - Service business process improvement

How Can We Evaluate and Improve Processes with the Process Variables?

- Fundamental difficulty of measuring efficiency
  - A policeman captured 10 criminals in a month. Is the number “10” high?
  - The different nature of measurement scales:

    Nominal (A: 1, B: 2), Ordinal (rank #1 and #2), Interval (20 and 40 ℃), Ratio (2 and 4 cm)

- A lot of objectives and attributes of organizations can be measured relatively only
  - Library A may be better than Library B because they have more visitors with less asset
  - In many cases, we should consider multiple criteria in a relative efficiency measurement

How Can We Evaluate and Improve Processes with the Process Variables?

- The relative efficiency can be measured in various ways

Partial measures

$$\frac{\text{Output}}{\text{Labor}} \quad \frac{\text{Output}}{\text{Machine}} \quad \frac{\text{Output}}{\text{Capital}} \quad \frac{\text{Output}}{\text{Energy}}$$

Multifactor measures

$$\frac{\text{Output}}{\text{Labor + Machine}} \qquad \frac{\text{Output}}{\text{Labor + Capital + Energy}}$$

Total measure

$$\frac{\text{Goods or Services Produced}}{\text{All inputs used to produce them}}$$

Efficiency Analysis Data: Example

- Depots of a large retailing organization which distributes goods to supermarkets

Depot	Input 1 (Workers)	Input 2 (Wages)	Output 1 (Quotes)	Output 2 (Orders)	Output 3 (Sales)
Depot 1	3	5	40	55	30
Depot 2	2.5	4.5	45	50	40
⋮	⋮	⋮	⋮	⋮	⋮
Depot 10	5	7	70	65	48
Depot 11	5	7	45	65	40
⋮	⋮				
Depot 19	3				
Depot 20	5	6	57	60	40

It looks Depot 10 is obviously more efficient than Depot 11.
How can we judge the “extent” of superiority?

Source: http://deazone.com/en/resources/tutorial/introduction

Data Envelopment Analysis: Concept

- Definition of the efficiency in DEA

$$\text{Efficiency} = \frac{\text{Weighted Sum of Outputs}}{\text{Weighted Sum of Inputs}}$$

$$\text{Efficiency of unit j} = \frac{u_1 y_{1j} + u_2 y_{2j} + \ldots}{v_1 x_{1j} + v_2 x_{2j} + \ldots}$$

where

u_1 = the weight given to output i

y_{1j} = amount of output 1 from unit j

v_1 = weight given to input 1

x_{1j} = amount of input 1 to unit j.

Data Envelopment Analysis: Concept

- Essential problem
  - Outputs and inputs are criteria that should be considered in relative performance measurement
  - Finding weights in a measurement case is the essential problem

- Idea
  - If the unit in question is most efficient (e.g., 1),

    then the efficiency of other units varies between 0 and 1
  - We can find the weights of the unit in question

    as we maximize its efficiency to 1,

    subject to the efficiency of all units being <= 1

$$\text{Max } h_0 = \frac{\sum_r u_r y_{rj_0}}{\sum_i v_i x_{ij_0}}$$

subject to

$$\frac{\sum_r u_r y_{rj}}{\sum_i v_i x_{ij}} \le 1 \quad \text{for each unit } j$$

$$u_r, v_i \ge \varepsilon$$

Efficiency Analysis Data: Example

- Depots of a large retailing organization which distributes goods to supermarkets

Depot	Input 1 (Workers)	Input 2 (Wages)	Output 1 (Quotes)	Output 2 (Orders)	Output 3 (Sales)
Depot 1	3	5	40	55	30
Depot 2	2.5	4.5	45	50	40
⋮	⋮	⋮	⋮	⋮	⋮
Depot 10	5	7	70	65	48
Depot 11	5	7	45	65	40
⋮	⋮	⋮	⋮	⋮	⋮
Depot 19	3	4	45	67	32
Depot 20	5	6	57	60	40

Source: http://deazone.com/en/resources/tutorial/introduction

Data Envelopment Analysis: Example

- The depot case (Depot 1)

$$\text{Efficiency} = \frac{u_1 y_1 + u_2 y_2 + u_3 y_3}{v_1 x_1 + v_2 x_2}$$

$y_1 = Quotes$
$y_2 = Orders$
$y_3 = Sales$
$x_1 = Workers$
$x_2 = Wages$

$$\text{maximize} \quad \frac{40u_1 + 55u_2 + 30u_3}{3v_1 + 5v_2}$$

subject to

$$(1) \quad \frac{40u_1 + 55u_2 + 30u_3}{3v_1 + 5v_2} \leq 1$$

$$(2) \quad \frac{45u_1 + 50u_2 + 40u_3}{2.5v_1 + 4.5v_2} \leq 1$$

$$\vdots \qquad \vdots$$

$$(20) \quad \frac{57u_1 + 60u_2 + 40u_3}{5v_1 + 6v_2} \leq 1$$

$$(21) \quad u_1, u_2, v_1, v_2, v_3 \geq \epsilon$$

Data Envelopment Analysis: Translation to LP Form

- The depot case (Depot 1)

$$\text{maximize} \quad \frac{40u_1 + 55u_2 + 30u_3}{3v_1 + 5v_2}$$

$$\text{subject to}$$

$$(1) \quad \frac{40u_1 + 55u_2 + 30u_3}{3v_1 + 5v_2} \leq 1$$

$$(2) \quad \frac{45u_1 + 50u_2 + 40u_3}{2.5v_1 + 4.5v_2} \leq 1$$

$$\vdots \quad \vdots$$

$$(20) \quad \frac{57u_1 + 60u_2 + 40u_3}{5v_1 + 6v_2} \leq 1$$

$$(21) \quad u_1, u_2, v_1, v_2, v_3 \geq \epsilon$$

$$\text{maximize} \quad 40u_1 + 55u_2 + 30u_3$$

$$\text{subject to}$$

$$(1) \quad 40u_1 + 55u_2 + 30u_3 - 3v_1 - 5v_2 \leq 0$$

$$(2) \quad 45u_1 + 50u_2 + 40u_3 - 2.5v_1 - 4.5v_2 \leq 0$$

$$\vdots \quad \vdots$$

$$(20) \quad 57u_1 + 60u_2 + 40u_3 - 5v_1 - 6v_2 \leq 0$$

$$(21) \quad 3v_1 + 5v_2 = 1$$

$$(22) \quad u_1, u_2, v_1, v_2, v_3 \geq \epsilon$$

Data Envelopment Analysis: Implementation of the Example

File Edit Format Run Options Window Help

```
from gurobipy import *

for i in range(20):

    m = Model()

    u1 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="u1")
    u2 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="u2")
    u3 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="u3")
    v1 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="v1")
    v2 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="v2")

    dmu1_output = 40*u1 + 55*u2 + 30*u3
    dmu2_output = 45*u1 + 50*u2 + 40*u3
    dmu3_output = 55*u1 + 45*u2 + 30*u3
    dmu4_output = 48*u1 + 20*u2 + 60*u3
    dmu5_output = 28*u1 + 50*u2 + 25*u3
    dmu6_output = 48*u1 + 20*u2 + 65*u3
    dmu7_output = 80*u1 + 65*u2 + 57*u3
    dmu8_output = 25*u1 + 48*u2 + 30*u3
    dmu9_output = 45*u1 + 64*u2 + 42*u3
    dmu10_output = 70*u1 + 65*u2 + 48*u3
    dmu11_output = 45*u1 + 65*u2 + 40*u3
    dmu12_output = 45*u1 + 40*u2 + 44*u3
    dmu13_output = 65*u1 + 25*u2 + 35*u3
    dmu14_output = 38*u1 + 18*u2 + 64*u3
    dmu15_output = 20*u1 + 50*u2 + 15*u3
    dmu16_output = 38*u1 + 20*u2 + 60*u3
    dmu17_output = 68*u1 + 64*u2 + 54*u3
    dmu18_output = 25*u1 + 38*u2 + 20*u3
    dmu19_output = 45*u1 + 67*u2 + 32*u3
    dmu20_output = 57*u1 + 60*u2 + 40*u3

    dmu1_input = 3*v1 + 5*v2
    dmu2_input = 2.5*v1 + 4.5*v2
    dmu3_input = 4*v1 + 6*v2
    dmu4_input = 6*v1 + 7*v2
    dmu5_input = 2.3*v1 + 3.5*v2
    dmu6_input = 4*v1 + 6.5*v2
    dmu7_input = 7*v1 + 10*v2
    dmu8_input = 4.4*v1 + 6.4*v2
    dmu9_input = 3*v1 + 5*v2
    dmu10_input = 5*v1 + 7*v2
    dmu11_input = 5*v1 + 7*v2
    dmu12_input = 2*v1 + 4*v2
    dmu13_input = 5*v1 + 7*v2
    dmu14_input = 4*v1 + 4*v2
    dmu15_input = 2*v1 + 3*v2
    dmu16_input = 3*v1 + 6*v2
    dmu17_input = 7*v1 + 11*v2
    dmu18_input = 4*v1 + 6*v2
    dmu19_input = 3*v1 + 4*v2
    dmu20_input = 5*v1 + 6*v2
```

File Edit Format Run Options Window Help

```
    dmu20_input = 5*v1 + 6*v2

    DMU_output = [dmu1_output, dmu2_output, dmu3_output, dmu4_output, dmu5_output,
                  dmu6_output, dmu7_output, dmu8_output, dmu9_output, dmu10_output,
                  dmu11_output, dmu12_output, dmu13_output, dmu14_output, dmu15_output,
                  dmu16_output, dmu17_output, dmu18_output, dmu19_output, dmu20_output]
    DMU_input = [dmu1_input, dmu2_input, dmu3_input, dmu4_input, dmu5_input,
                 dmu6_input, dmu7_input, dmu8_input, dmu9_input, dmu10_input,
                 dmu11_input, dmu12_input, dmu13_input, dmu14_input, dmu15_input,
                 dmu16_input, dmu17_input, dmu18_input, dmu19_input, dmu20_input]

    m.setObjective(DMU_output[i], GRB.MAXIMIZE)
    c1 = m.addConstr(DMU_output[i] - DMU_input[i] <= 0)
    c2 = m.addConstr(DMU_output[i-1] - DMU_input[i-1] <= 0)
    c3 = m.addConstr(DMU_output[i-2] - DMU_input[i-2] <= 0)
    c4 = m.addConstr(DMU_output[i-3] - DMU_input[i-3] <= 0)
    c5 = m.addConstr(DMU_output[i-4] - DMU_input[i-4] <= 0)
    c6 = m.addConstr(DMU_output[i-5] - DMU_input[i-5] <= 0)
    c7 = m.addConstr(DMU_output[i-6] - DMU_input[i-6] <= 0)
    c8 = m.addConstr(DMU_output[i-7] - DMU_input[i-7] <= 0)
    c9 = m.addConstr(DMU_output[i-8] - DMU_input[i-8] <= 0)
    c10 = m.addConstr(DMU_output[i-9] - DMU_input[i-9] <= 0)
    c11 = m.addConstr(DMU_output[i-10] - DMU_input[i-10] <= 0)
    c12 = m.addConstr(DMU_output[i-11] - DMU_input[i-11] <= 0)
    c13 = m.addConstr(DMU_output[i-12] - DMU_input[i-12] <= 0)
    c14 = m.addConstr(DMU_output[i-13] - DMU_input[i-13] <= 0)
    c15 = m.addConstr(DMU_output[i-14] - DMU_input[i-14] <= 0)
    c16 = m.addConstr(DMU_output[i-15] - DMU_input[i-15] <= 0)
    c17 = m.addConstr(DMU_output[i-16] - DMU_input[i-16] <= 0)
    c18 = m.addConstr(DMU_output[i-17] - DMU_input[i-17] <= 0)
    c19 = m.addConstr(DMU_output[i-18] - DMU_input[i-18] <= 0)
    c20 = m.addConstr(DMU_output[i-19] - DMU_input[i-19] <= 0)
    c21 = m.addConstr(DMU_input[i] == 1)

    print('₩n')
    m.optimize()
    print(m.getVars(), '₩n')

    for v in m.getVars():
        print('%s: %g' % (v.varName, v.x))
    print('Objective value of depot', i+1, 'is:', m.objVal)
```

Ln: 1 Col: 0

Data Envelopment Analysis: Implementation of the Example

- Weight and objective values

Depot	W of quote (u_1)	W of order (u_2)	W of sales (u_3)	Weight of worker (v_1)	W of wage (v_2)	Efficiency (objective)
Depot 1 case	0.00524723	0.0110999	0	0.299697	0.0201816	0.8203835
Depot 2 case	0.00624625	0.0132132	0	0.356757	0.024024	0.9417417
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Depot 10 case	0.0126984	0	0	0	0.142857	0.8888888
Depot 11 case	0	0.0047079	0.00813182	0.0295314	0.121763	0.6312861
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Depot 19 case	0	0.00862069	0.0132004	0	0.25	1
Depot 20 case	0.0148148	0	0	0	0.166667	0.8444444

Efficiency Data: Example

- Depots of a large retailing organization which distributes goods to supermarkets

Depot	Input 1 (Workers)	Input 2 (Wages)	Output 1 (Quotes)	Output 2 (Orders)	Output 3 (Sales)
Depot 1	3	5	40	55	30
Depot 2	2.5	4.5	45	50	40
⋮	⋮	⋮	⋮	⋮	⋮
Depot 10	5	7	70	65	48
Depot 11	5	7	45	65	40
⋮	⋮	It looks Depot 10 is obviously more efficient than Depot 11. How can we judge the “extent” of superiority?			
Depot 19	3				
Depot 20	5	6	57	60	40

Source: http://deazone.com/en/resources/tutorial/introduction

Data Envelopment Analysis: Interpretation of the Example

Depot	Efficiency	Depot	Efficiency
Depot 19	1	Depot 13	0.83
Depot 15	1	Depot 6	0.83
Depot 14	1	Depot 1	0.82
Depot 12	1	Depot 3	0.82
Depot 9	0.96	Depot 7	0.71
Depot 5	0.95	Depot 4	0.65
Depot 2	0.94	**Depot 11**	**0.63**
Depot 16	0.91	Depot 17	0.55
Depot 10	**0.89**	Depot 8	0.52
Depot 20	0.84	Depot 18	0.42

Data Envelopment Analysis: Interpretation of the Example

Depot	Efficiency	Depot	Efficiency
Depot 19	**1**	Depot 13	0.83
Depot 15	**1**	Depot 6	0.83
Depot 14	**1**	Depot 1	0.82
Depot 12	**1**	Depot 3	0.82
Depot 9	0.96	Depot 7	0.71
Depot 5	0.95	Depot 4	0.65
Depot 2	0.94	Depot 11	0.63
Depot 16	0.91	**Depot 17**	**0.55**
Depot 10	0.89	**Depot 8**	**0.52**
Depot 20	0.84	**Depot 18**	**0.42**

Data Envelopment Analysis: Interpretation of the Example

- How can we improve Depot 18?

  We may want to benchmark efficient alternatives from the analysis of Depot 18

Depot 18	Criteria	Depot 12	Depot 15	Depot 19
4.0	Workers	2.0	2.0	3.0
6.0	Wages	4.0	3.0	4.0
25.0	Quotes	45.0	20.0	45.0
38.0	Orders	40.0	50.0	67.0
20.0	Sales	44.0	15.0	32.0

Data Envelopment Analysis: Interpretation of the Example

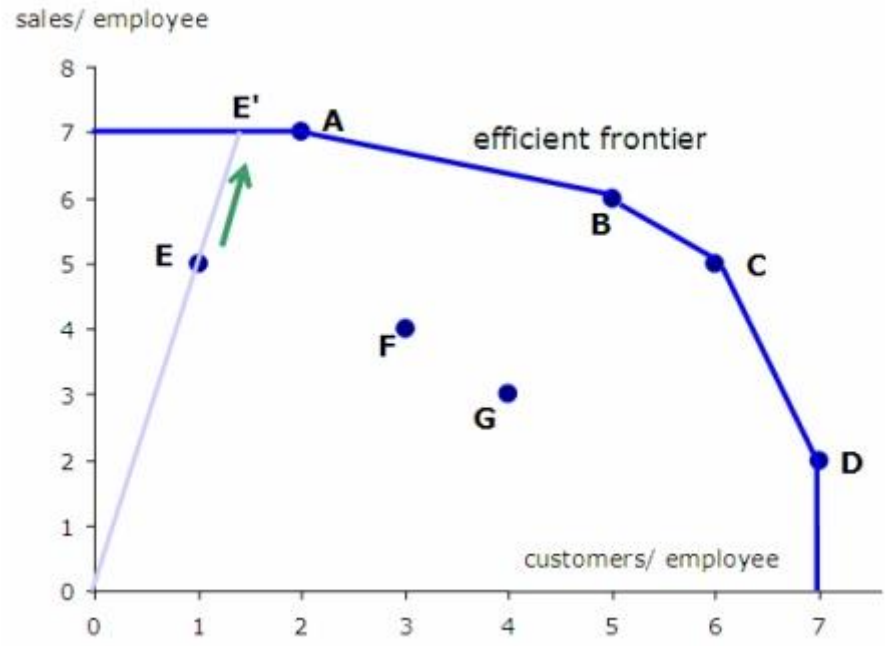
- How can we improve Depot 18?

  We may want to control the inputs to make the efficiency from 0.42 to 1

	Actual	Target
Workers	4.0	1.68
Wages	6.0	2.52
Quotes	25.0	25.0 ?
Orders	38.0	38.0 ?
Sales	20.0	20.0 ?

Data Envelopment Analysis: Visual Illustration of the Concept

- Efficient DMUs form a frontier and envelop the data

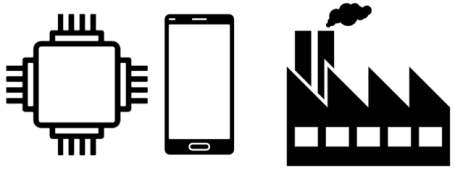


Source: http://blog.datumbox.com/data-envelopment-analysis-tutorial/

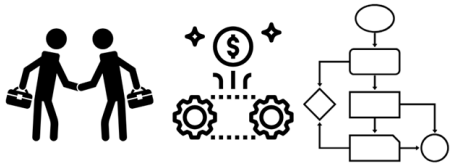
Discussion

- DEA is highly applicable to real world problems with real records:
  There are a lot of records related to specific organizations, systems, and events, and such records show their inputs and outputs (multiple criteria). Thus, DEA has been applied to evaluate complex service systems (e.g., bank, library, university, hospital) and events (e.g., bankruptcy)
- The concept of “efficiency” can aggregate numerous criteria in complexity in one index
- DEA can be applied to various forms of data without knowing an exact function (i.e., y=f(x))
- DEA identifies an empirical frontier (i.e., a set of efficient units to benchmark)
- It is difficult to interpret the weights derived
- It is also difficult to identify and define important inputs and outputs
- The linear additive assumption may not be correct

Processes around Us Involve Problems and Need Improvement



Manufacturing Process Management



Business Process Management

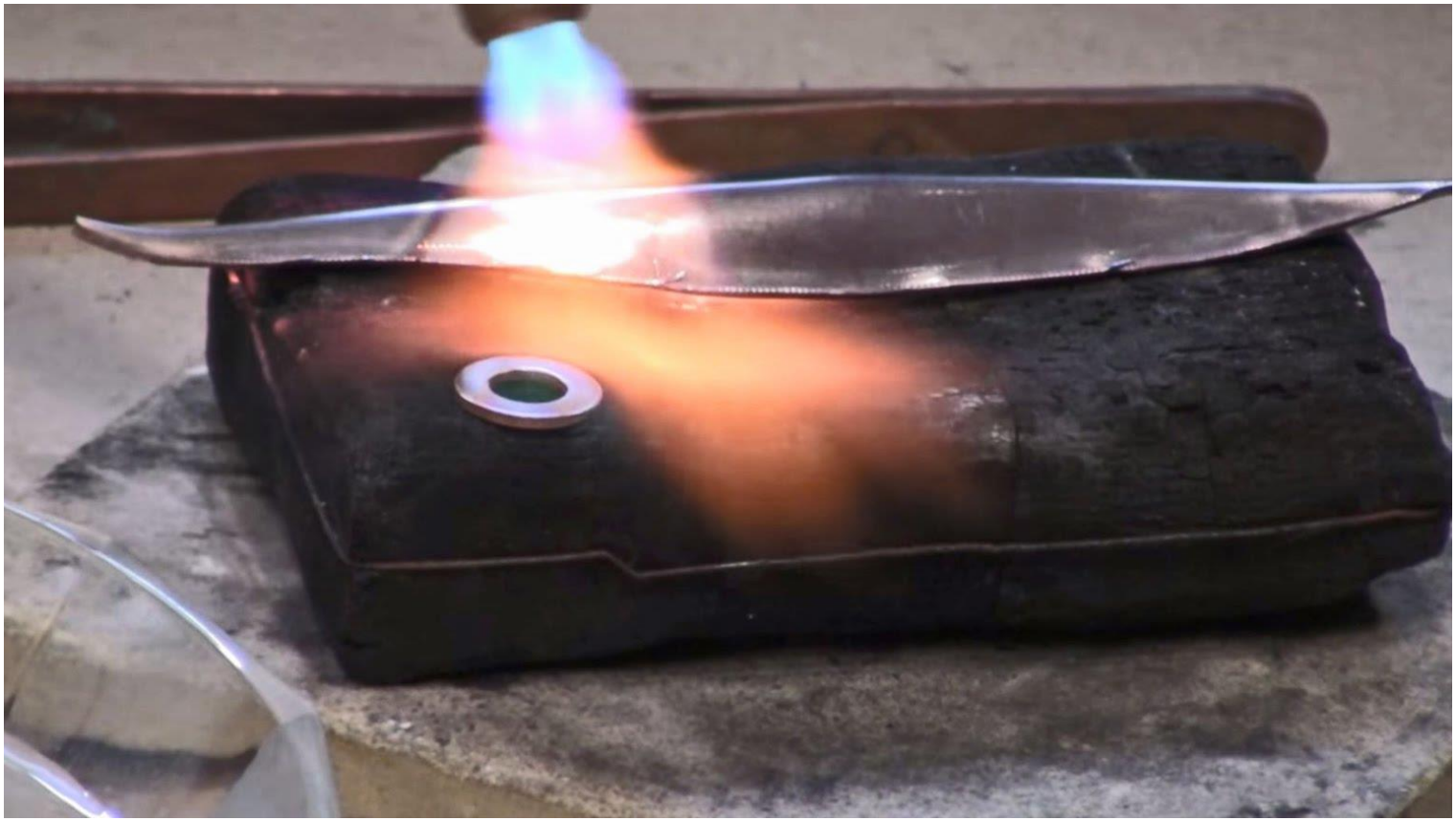
Personal Process Management

Service/ variables	Inputs				Outputs		
	I1	I2	I3	I4	O1	O2	O3
S2-a	…	…	…	…	…	…	…
S2-b	…	…	…	…	…	…	…
S2-c	…	…	…	…	…	…	…
S2-d	…	…	…	…	…	…	…
S2-e	…	…	…	…	…	…	…
S3-a	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
S5-a	…	…	…	…	…	…	…
S5-b	…	…	…	…	…	…	…
S5-c	…	…	…	…	…	…	…
S5-d	…	…	…	…	…	…	…
S5-e	…	…	…	…	…	…	…
S5-f	…	…	…	…	…	…	…
S5-g	…	…	…	…	…	…	…

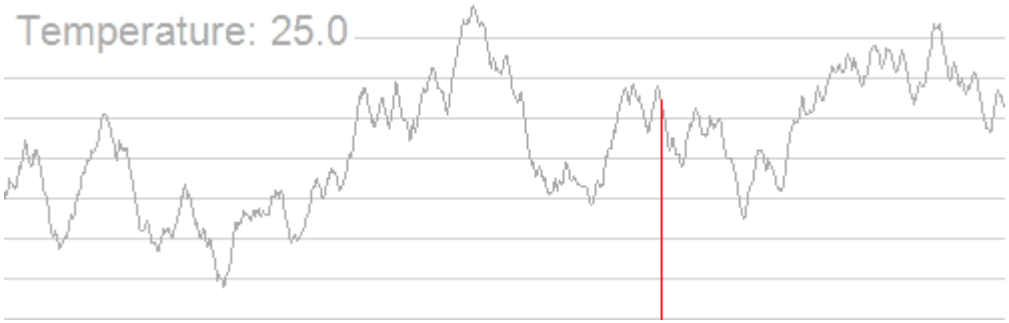
Appendix: Other Metaheuristic Algorithms

Simulated Annealing

Annealing Metal



SA Example



Simulated Annealing

- Basic concepts of simulated annealing
  - Early emphasis on taking steps in random directions
  - Gradually moves the emphasis on climbing upward (downward) by rejecting more moves that go in a downward (upward) direction
  - Think about “How we, humans make decisions” from the perspective of annealing process

- Move selection rule: Approach for selecting the next trial solution
  - If the candidate under consideration is better than the current trial solution, it is selected
  - If it is worse, a certain acceptance probability is applied

    Probability depends on how much worse it is (e.g., see the table in the next page)

    Probability also depends on the current temperature

Concept of Probabilistic Acceptance of a Worse Solution

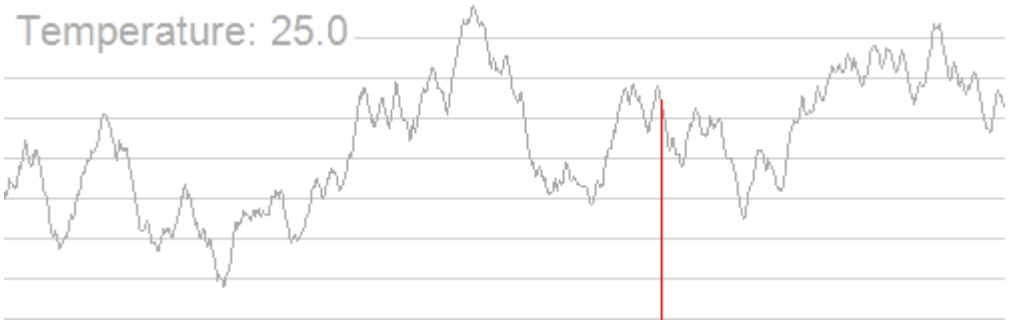
- Accepting a worse solution contributes to escape from a local optimum

$x = \frac{Z_n - Z_c}{T}$	Prob{acceptance} = e^x
−0.01	0.990
−0.1	0.905
−0.25	0.779
−0.5	0.607
−1	0.368
−2	0.135
−3	0.050
−4	0.018
−5	0.007

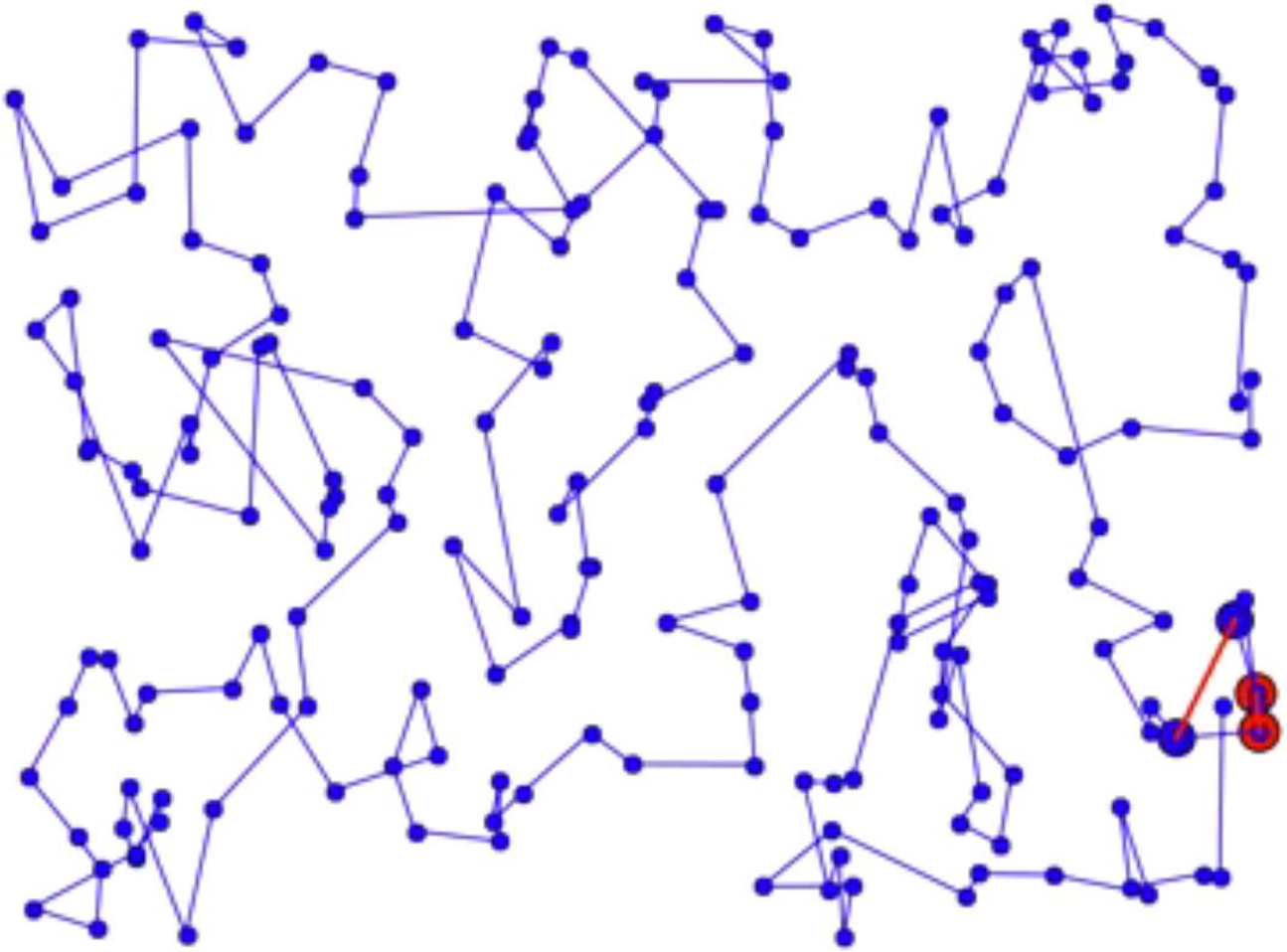
Algorithm

- Select algorithm's parameters
  - Initial temperature $T_0 > 0$
  - Cooling schedule CT(t)
  - Maximum iteration at fixed temperature
  - Objective function f (*)
- Begin
  - Generate initial solution s
  - Start iteration at initial temperature T_0
  - Calculate fitness value for initial solution
- Repeat until stopping criterion is met
  - Generate candidate solution in neighborhood of s, s'
  - Compute fitness value for candidate
  - If s' is better, s <- s'
    Elseif, exp(-Δf/KT) > rand(), then s <- s'
  - Reduce temperature as per CT(t)

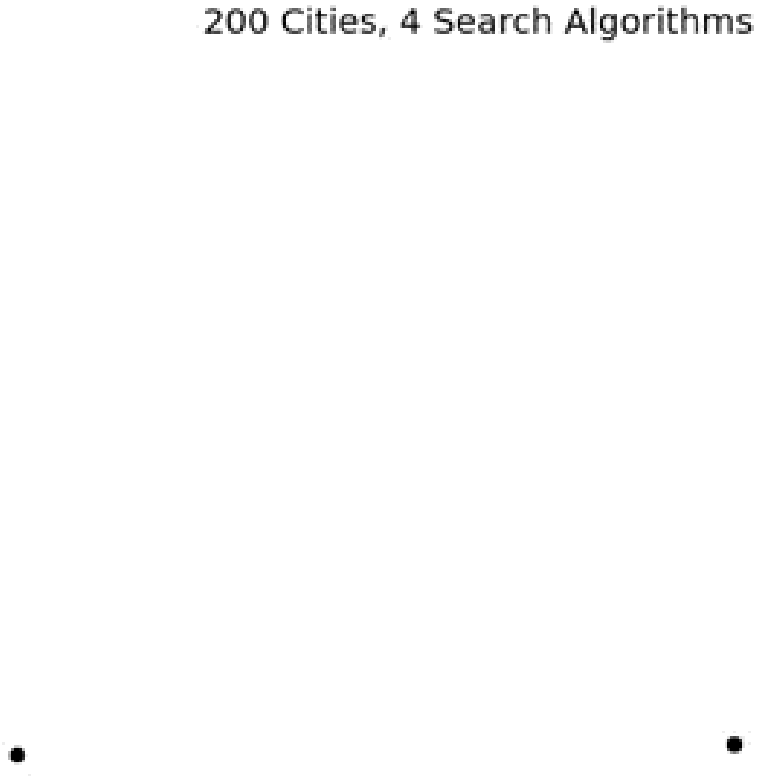
SA Example



SA Example



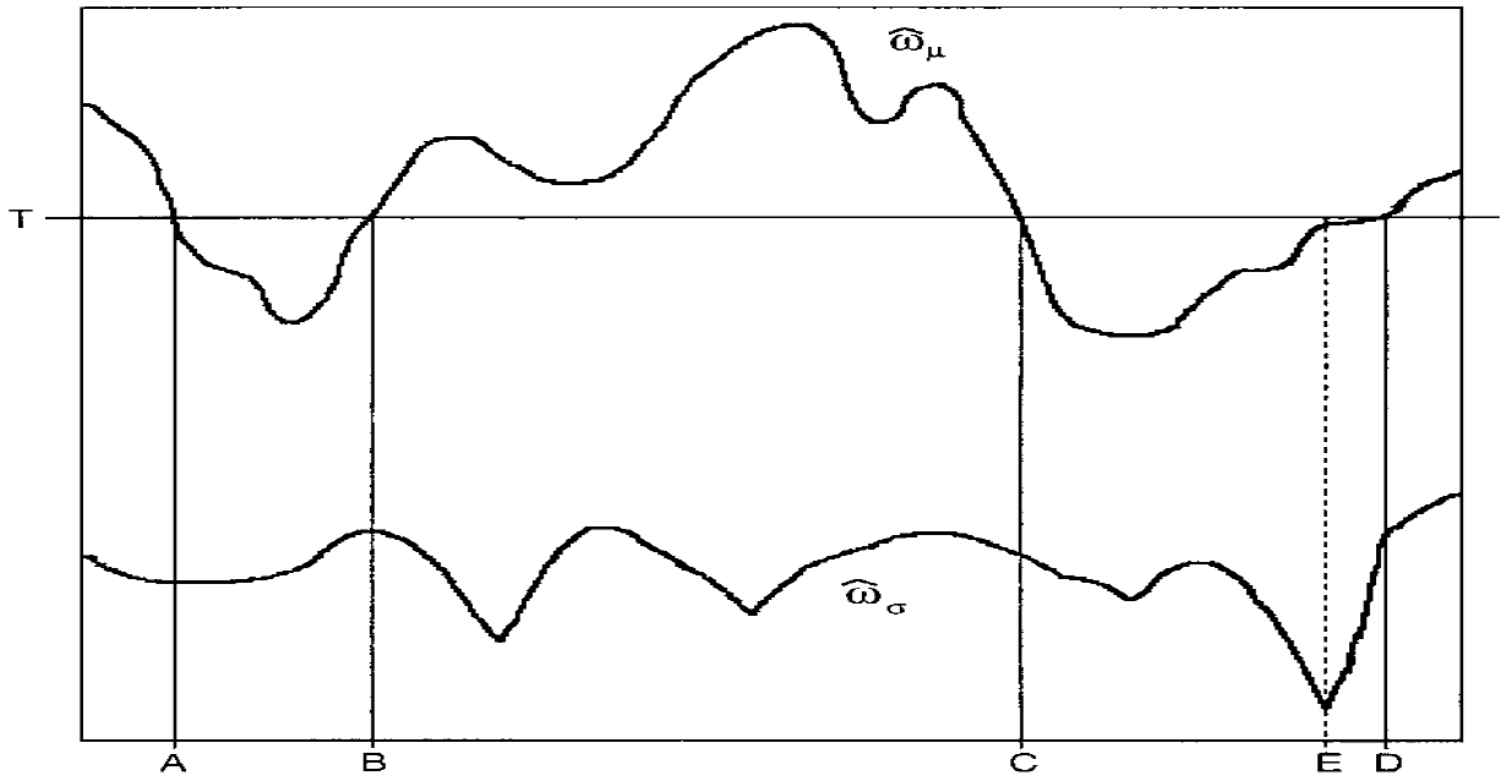
SA Example



Think About

- Initial temperature should be set adequately high
  - Too low a temperature can cause the system to get stuck in local optima
  - A very high value of T can cause difficulty in reaching the optimum solution
  - There should be a gradual reduction in control parameter T
    Cooling schedule affects the search process; several cooling strategies have been used
    (e.g., Exponential, Linear, Logarithmic, Non-monotonic, and Adaptive)
- Iterations at each temperature should be enough to stabilize the system
- Generation of candidate solutions in neighborhood can be customized
- SA is well suited to problems with a rough landscape (large number of local optima), but if the optimization problem has a smooth landscape with limited local optima, SA is not of much use and unnecessarily delays convergence

Example



$\hat{\omega}_\mu$ vs $\hat{\omega}_\sigma$?

Tabu Search

Tabu

Tabu Search

- Basic concepts of tabu search
  - Subroutine: a local search procedure
    - Searches the whole neighborhood deterministically unlike SA that employs a random search
    - Danger with this approach: after moving away from a local optimum, the process will cycle back to the same local optimum
  - Solution: tabu search temporarily forbids moves that would take it back to the last optimum
    - Tabu moves are saved in a tabu list
    - This helps the algorithm from being trapped in cycles
  - If no better solution exists, the best neighbor is selected to replace the existing solution
    → Such worse solutions are also accepted to move out of local optima
  - Think about “How we, humans make decisions” from the perspective of our tabu lists

Concept of Memory Usage in Optimization

- Tabu list = Short-term memory

- Medium- and long-term memories can also be used
  - Medium-term memory for “intensification” stores the best solution obtained during the search and explores a portion of the feasible region more thoroughly than usual
  - Long-term memory for “diversification” helps in forcing the search into previously unexplored areas of the feasible region

Algorithm

- Select algorithm's parameters
  - Objective function f (*)
  - Maximum size of tabu list, s_t
- Begin
  - Initialize tabu list, short term and long term memory
  - Generate initial solution s
  - Calculate fitness value for initial solution
- Repeat
  - Generate candidate solution in neighborhood of s, s'
  - if s' ∉ tabu list, compute fitness value for candidate and update s <- s' as necessary
  - Update tabu list, medium and long term memory

Think About

- An efficient way is to use the elite (best) solution found so far to generate new solutions to speed up convergence

- The effectiveness of TS algorithm depends on appropriate selection of neighborhood operator and search space
  - This requires in-depth knowledge of problem at hand
  - An aspiration threshold is often used
  - Diversification should be very carefully handled to avoid getting stuck at local minima
  - Tabu list should be carefully formulated to make search effective and minimizing computation time
    The size of tabu list may be static, dynamic or adaptive
    Store moves or solution attributes vs. Store actual solutions in the tabu list

- The best feature of TS is its ability to avoid cycles. However, unlike SA, sometimes it is not that efficient enough in moving out of the local minima

Particle Swarm Optimization

Classification of Metaheuristics

- Nature inspired versus non-nature inspired

  e.g., GA vs. TS

- Memory usage versus memory less

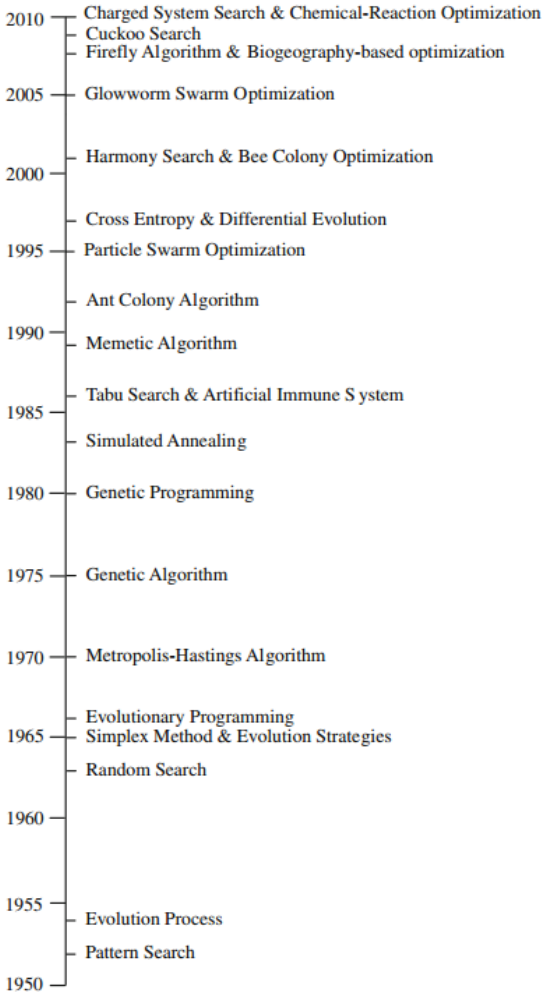
  e.g., TS vs. SA

- Population-based versus single-point search

  e.g., GA vs. SA

Population-based Metaheuristics Algorithms

- Genetic Algorithm (GA)
- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)
- Memetic Algorithm (MA)
- Artificial Immune System (AIS)

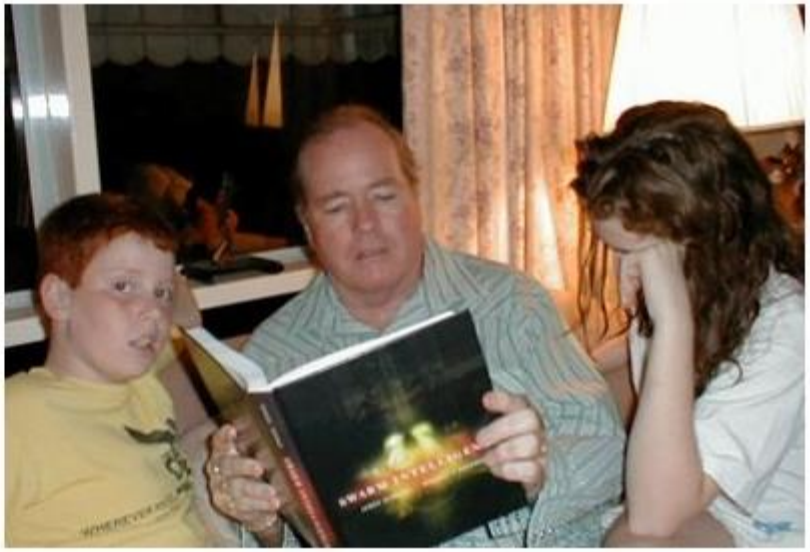


Particle Swarm Optimization

- Swarm intelligence
  - Artificial intelligence based on swarm behavior of systems (e.g., PSO based on birds flocking)
  - Think about “How we, humans make decisions” from the perspective of swarm behavior

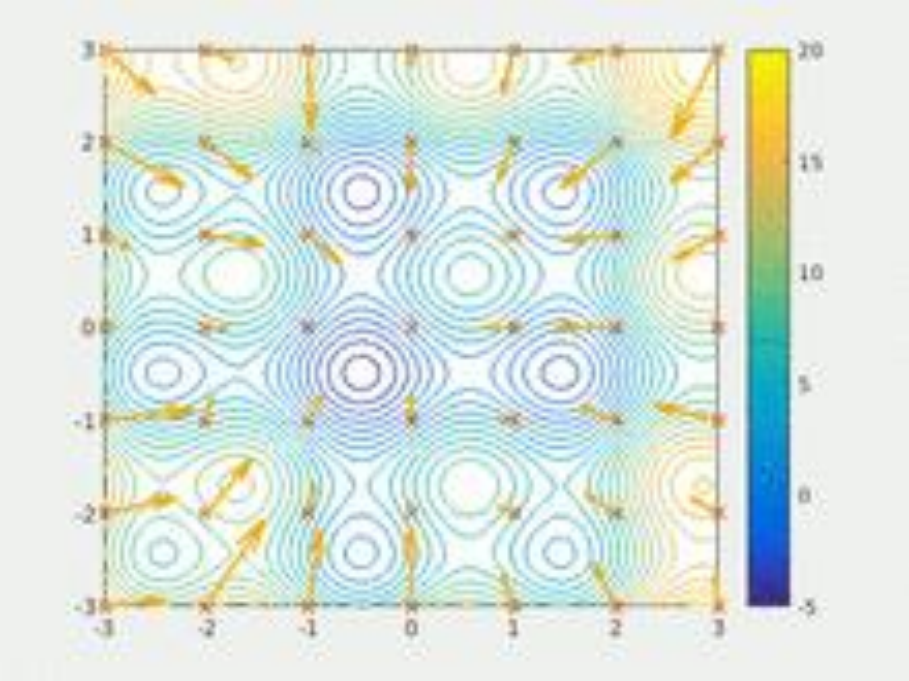
Inventors

Russell Eberhart
electrical engineer

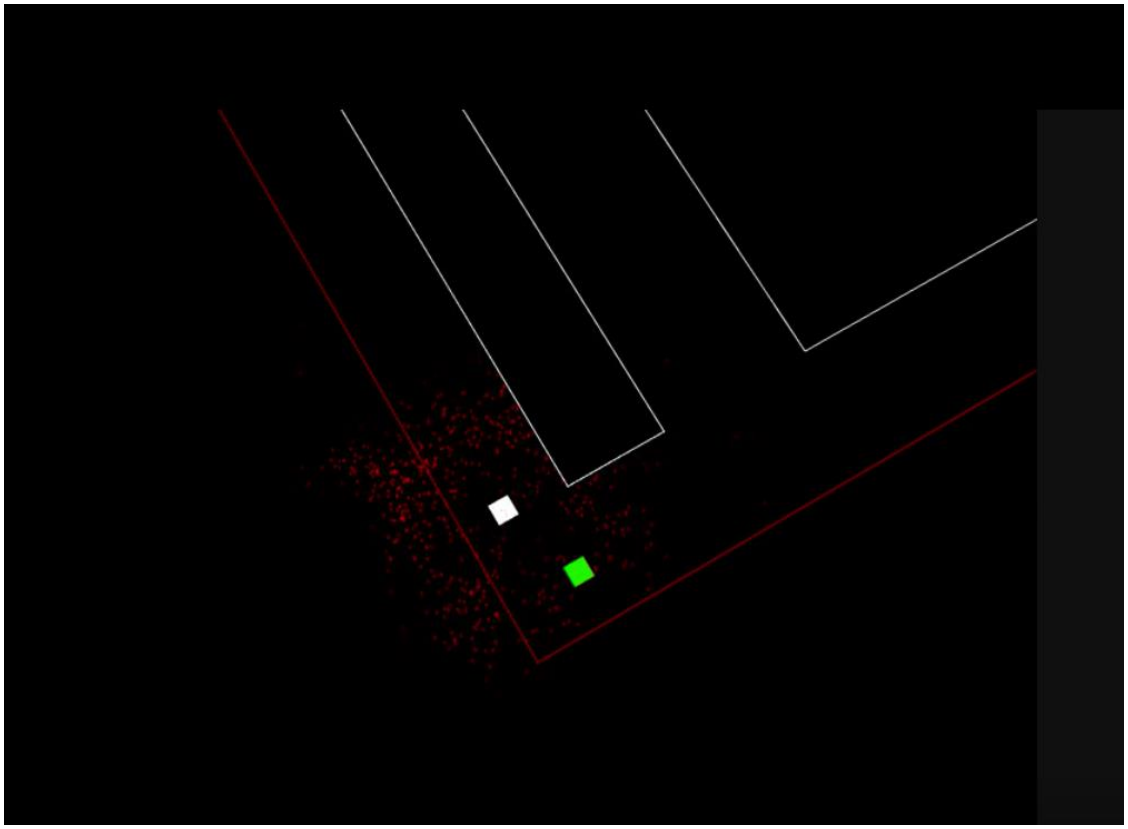


James Kennedy
social-psychologist

Example

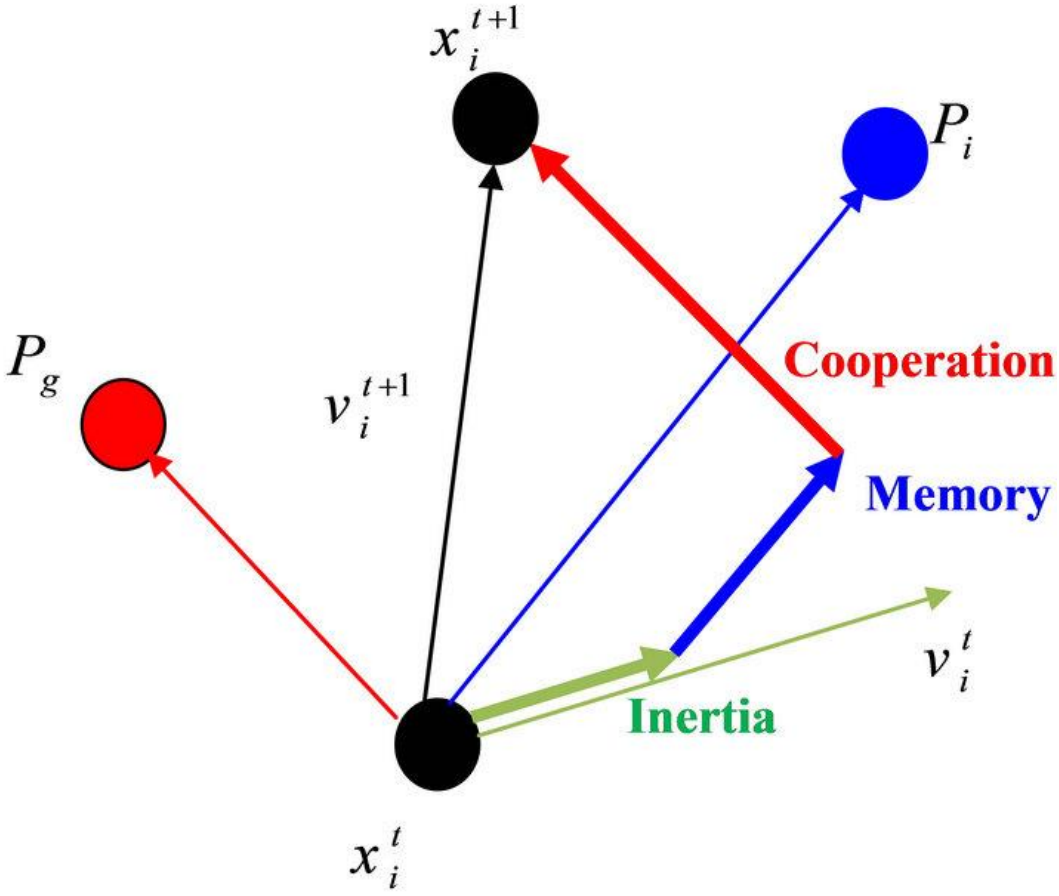


Example

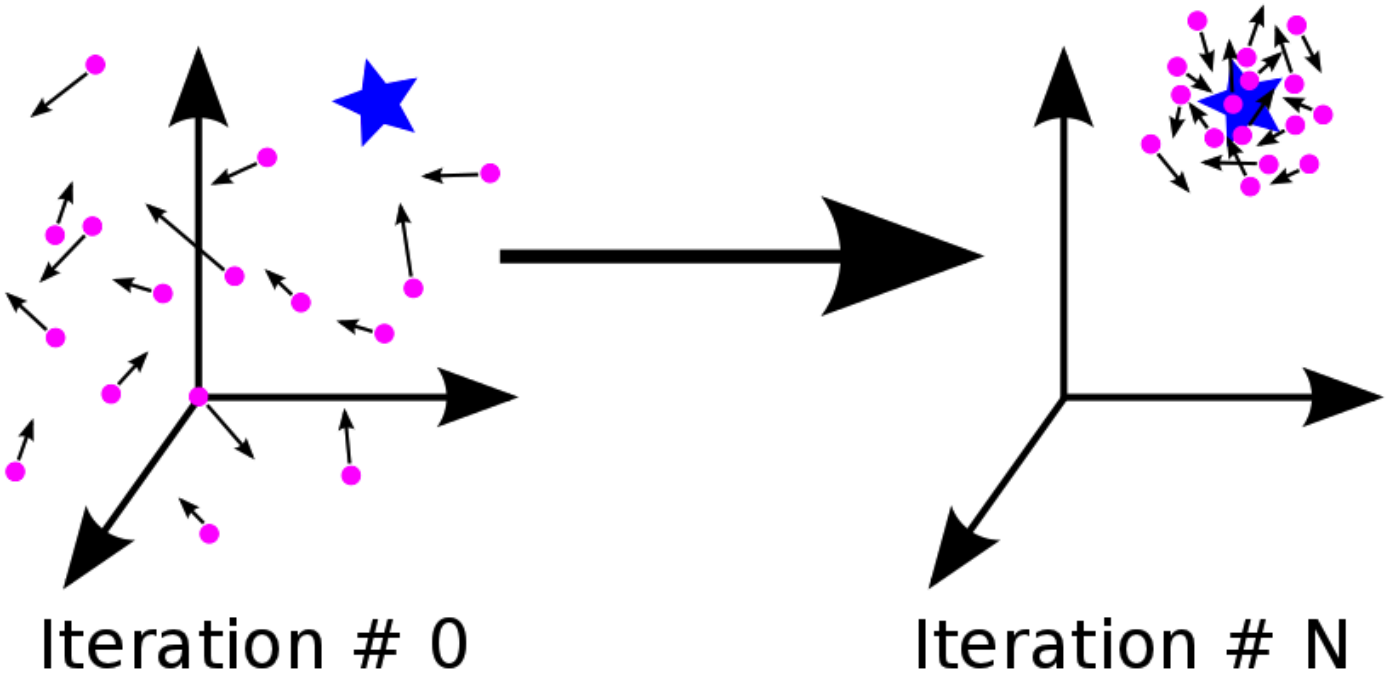


Example

A Mathematical Representation of PSO

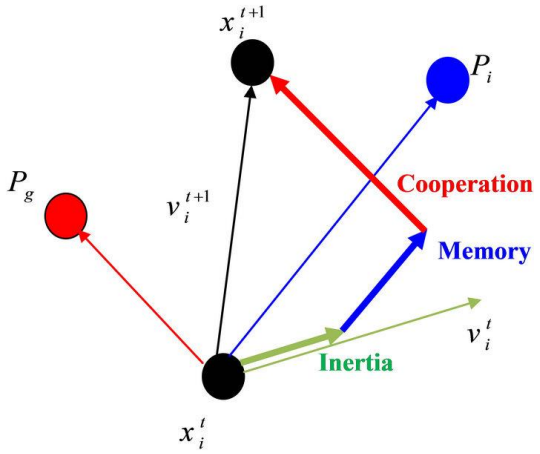


A Mathematical Representation of PSO



Algorithm of PSO

- Select algorithm's parameters
  - Swarm size
  - Dimension of particle
  - Objective function f (*)
  - Maximum no. of iteration
- Begin
  - Generate initial swarm
  - Initialize velocity vector
  - Calculate fitness value for initial swarm
  - Compute gBest and pBest
- Repeat
  - Generate new velocity vector using the above equation:
  - Compute new position vector of new_swarm using this equation:
  - Replace pBest and gBest values



$$v(t+1) = w(t) \times v(t) + c_1 \times r_1(\text{gBest} - x(t))c_2 \times r_2(\text{pBest} - x(t))$$

$$x(t+1) = x(t) + v(t+1)$$

PSO Practice

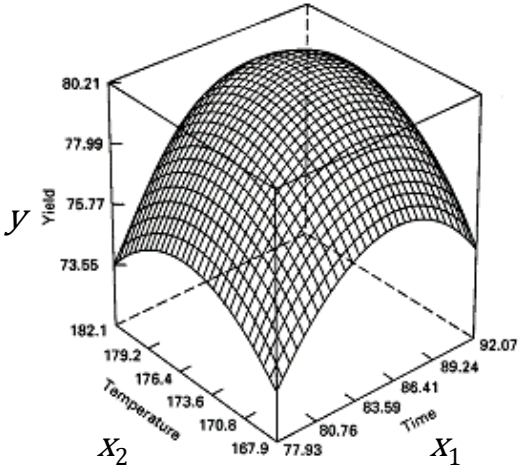
"Model building" "Search optimal setting for **x**"

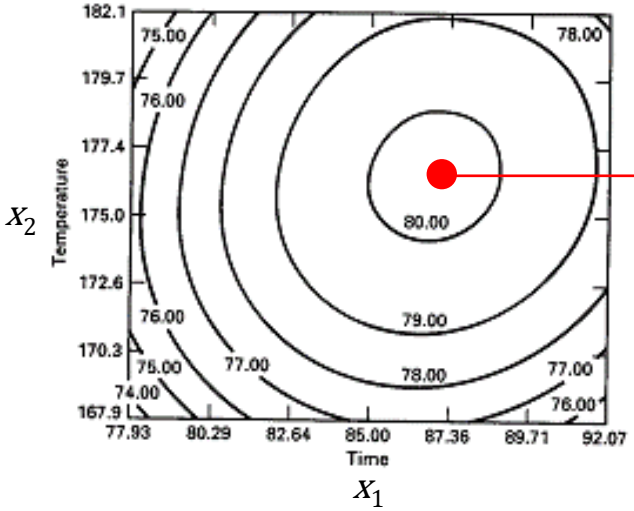
$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon \rightarrow \hat{y}(\mathbf{x}) \rightarrow \underset{\mathbf{x} \in \Omega}{\text{optimize}}\ \hat{y}(\mathbf{x})$$

Polynomial approximation for true f(x)

Example*

y = process yield (Larger the better type),
x_1 = reaction time, x_2 = reaction temperature





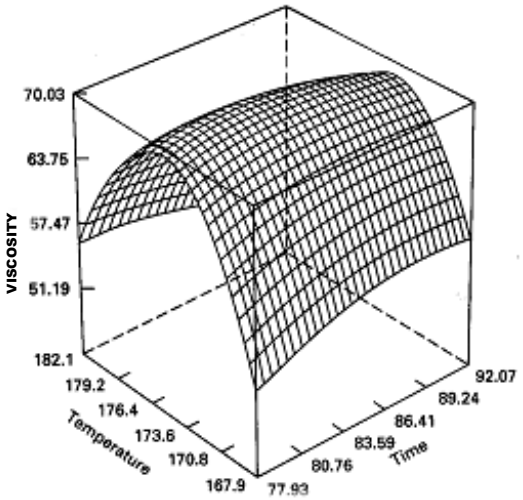
$\hat{y}_1^* = 80.21$ at $\mathbf{x}^* = (87.0, 176.5)$

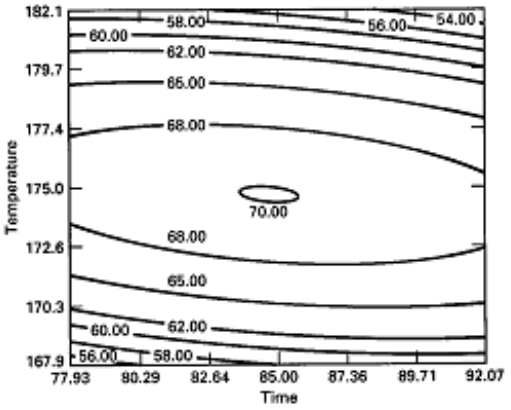
Source: Response Surface Methodology (Myers and Montgomery, 1995)

PSO Practice

- What if y_2(viscosity; NTB) and y_3(molecular weight; STB) are added?

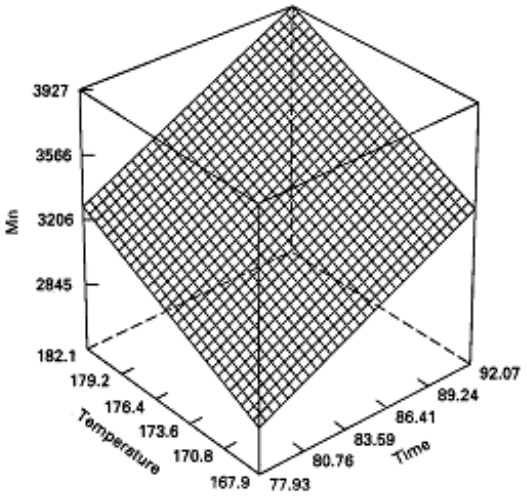
- Response Surface of y_2

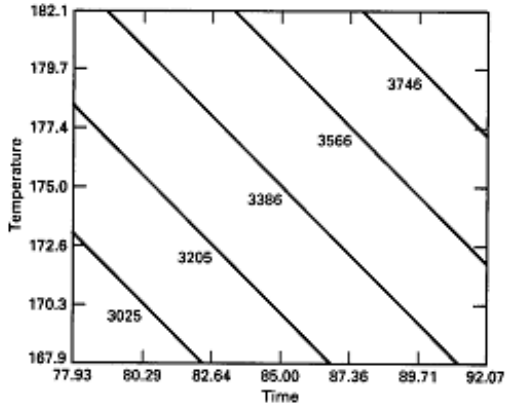




PSO Practice

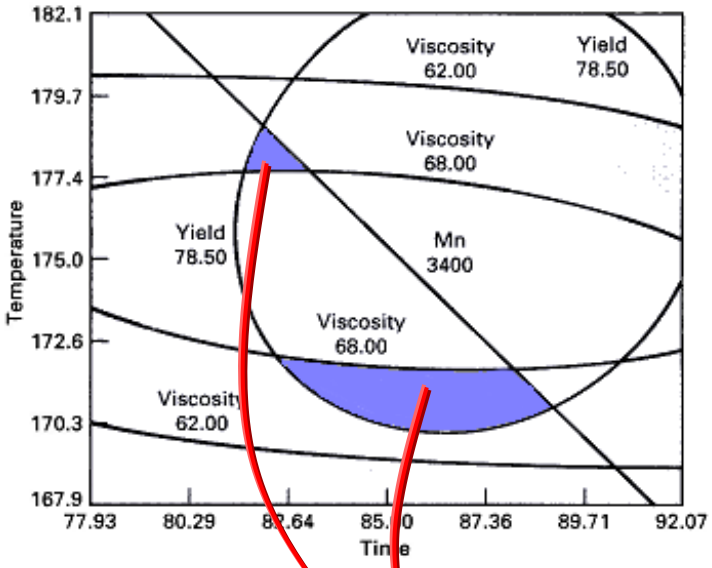
- Response Surface of y_3





PSO Practice

- Acceptable Region : $y_1 \geq 78.5$, $62 \leq y_2 \leq 68$, $y_3 \leq 3400$



Optimal $\mathbf{x}^* = (x_1^*, x_2^*) = ?$

Think About

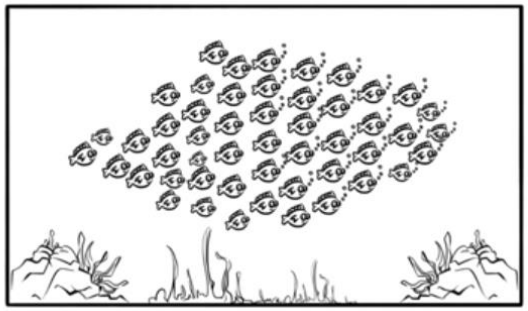
- Apart from using the entire swarm to compute gBest, local small neighborhood-based structures can be used as well

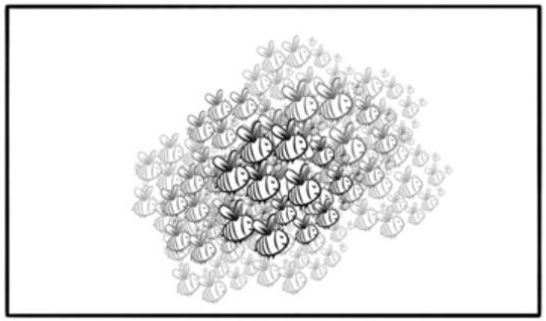
- Inertia weight decides the expanse of search space explored
  - A high value of inertia weight w, at the start of iteration helps in exploring a wide search space
    As iterations proceed, w can be reduced in order to intensify the search
  - Acceleration factors, c_i are used to make the particles follow the best obtained solution i: Low values of acceleration constants allow particles to roam far from target regions before being tugged back, while high values result in movement towards the target regions

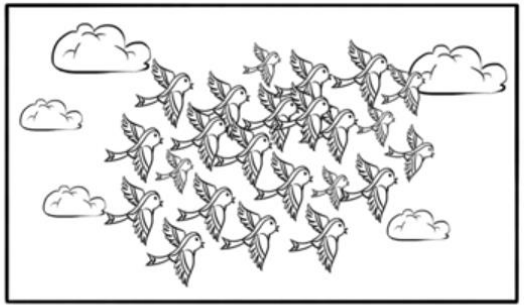
- To reduce the possibility of particles flying out of the problem space, a restriction can be imposed on the velocity increment at each stage

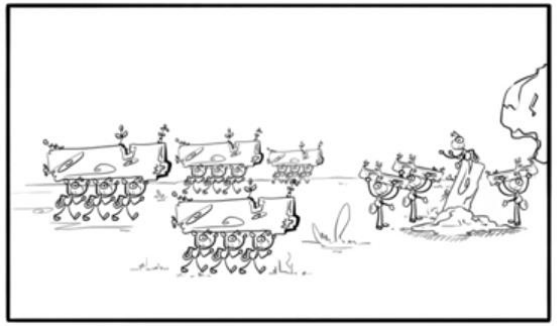
- The performance of PSO is known relatively insensitive to swarm size, if it is not too small

Swarm Intelligence



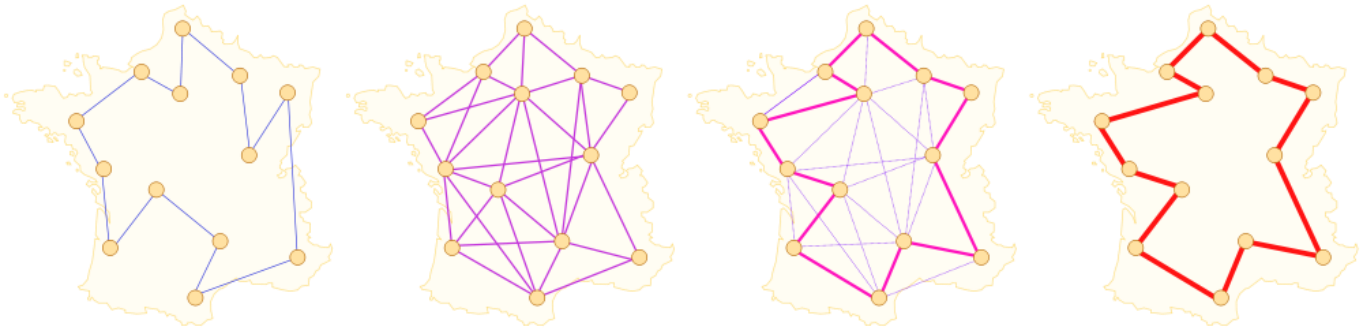






Ant Colony Algorithm

- ACO is very powerful to solve a dynamic network routing
- Think about “How we, humans make decisions” from the perspective of ant system



$$\max_{j \notin S}\{\tau_{i,j}{}^{\alpha}\eta_{i,j}{}^{\beta}\}$$

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_k \Delta\tau_{i,j}{}^{k}$$

$$\Delta\tau_{i,j}{}^{k} = \begin{cases} 1/c_k & \text{if ant } k \text{ travels on arc } i,j \\ 0 & \text{otherwise} \end{cases}$$

c_k = Cost of ant k's tour

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}{}^{best}$$

$$\Delta\tau_{i,j}{}^{best} = \begin{cases} 1/c_{best} & \text{if best ant travels on arc } i,j \\ 0 & \text{otherwise} \end{cases}$$

c_{best} = Cost of the best ant's tour

Cuckoo Search Algorithm

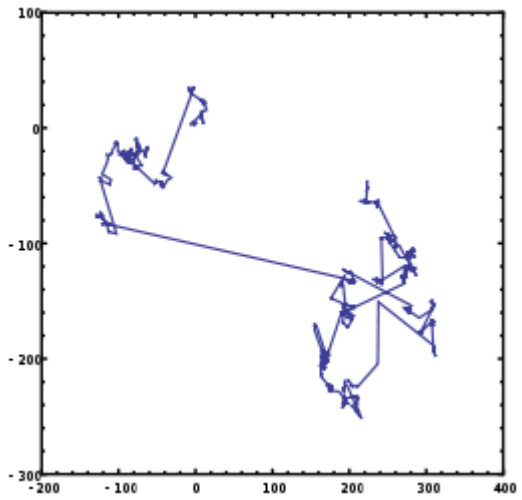
Cuckoo Search Algorithm

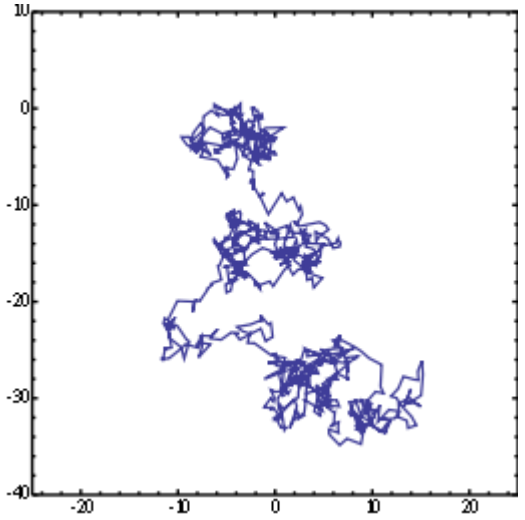
begin
  Objective function $f(x)$, $x=(x_1,\ldots,x_d)^T$;
  Initial a population of n host nests x_i $(i=1,2,\ldots,n)$;
    while ($t <$ Maximum Generation) or (stop criterion);
      Get a cuckoo (say i) randomly
                and generate a new solution by Lévy flights;
      Evaluate its quality/fitness; F_i
      Choose a nest among n (say j) randomly;
      if ($F_i > F_j$),
          Replace j by the new solution;
      end
      Abandon a fraction (P_a) of worse nests
          [and build new ones at new locations via Lévy flights];
       Keep the best solutions (or nests with quality solutions);
      Rank the solutions and find the current best;
    end while
    Post process results and visualization;
end

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda)$$

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high quality of eggs (solutions) will carry over to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with a probability $P_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

Lévy Flight


100
0
-100
-200
-300
-200
-100
0
100
200
300
400


10
0
-10
-20
-30
-40
-20
-10
0
10
20

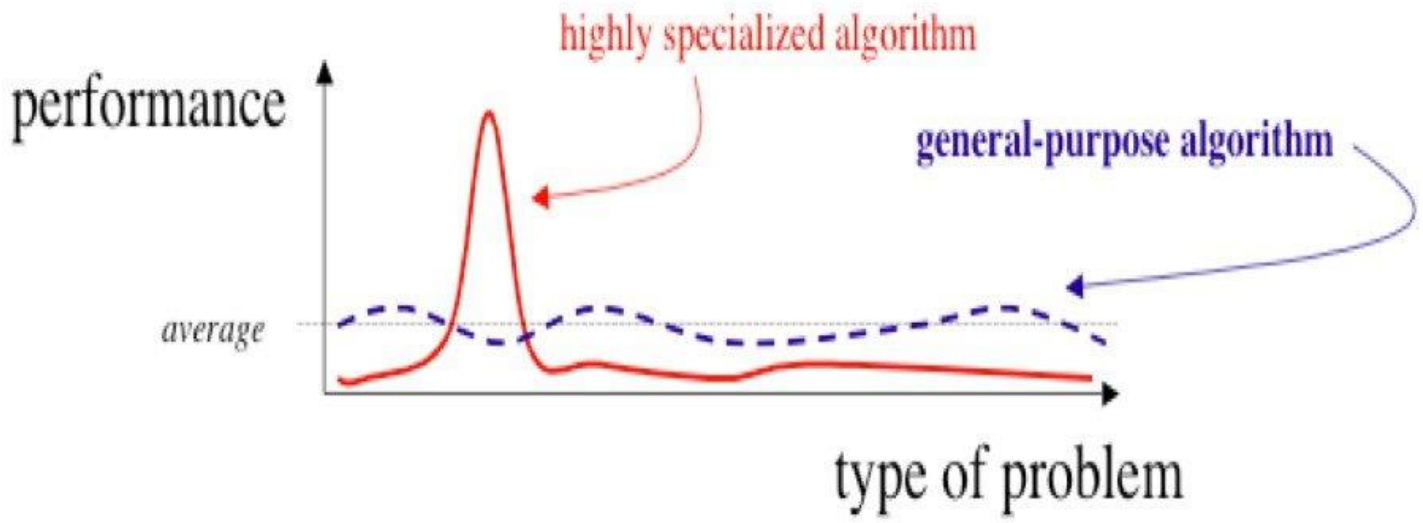
Applications of Metaheuristics

Algorithm	Area (s) of Application
Simulated Annealing (SA)	• Job shop scheduling in production system • Vehicle routing in transport and logistics management • Communication: mobile network design, routing, channel allocation
Tabu Search (TS)	• Resource allocation in industry, university etc. • Engineering technology: cell placement in VLSI, power distribution, structural design • Artificial Intelligence: pattern recognition, data mining, clustering
Genetic Algorithm (GA)	• Financial planning, stock predictions • Image processing: compression, segmentation • Sequencing in FMS (flexible management systems)
Differential Evolution (DE)	• Signal and image processing • Product design: aerodynamic • Chemical engineering
Particle Swarm Optimization (PSO)	• Telecommunications: network design, routing • Neural network training • System simulation and identification • Decision making and planning • Signal processing
Ant Colony Optimization (ACO)	• Set problems: set partitioning and covering, maximum independent set, bin packing • Scheduling: flow shop scheduling, process planning • Bioinformatics: protein folding, DNA sequencing
Memetic Algorithm (MA)	• Machine learning: neural network training, pattern classification • System engineering • Scheduling: production planning, rostering • Set problems: bin packing, set covering
Artificial Immune System (AIS)	• Data mining and data analysis • Clustering and classification • Network and computer security • Engineering design optimization

No Free Lunch Theorem (Wolpert and Macready, 1997)

- The averaged performance for all possible problems is the same for all algorithms

- No global distinction can be made between performances of any two algorithms apart from conditions where one algorithm is more suited to an application than the other



Think About

- It is not easy to prove the efficiency of metaheuristic algorithms “theoretically”

  Studies usually rely on empirical experimental results

- Each metaheuristic algorithm has its own set of advantages and limitations that makes it more suitable to a particular kind of application. Finding the best suited algorithm is important

- Combination of population-based and single-point search algorithms is powerful

- Power of good conceptual basis in analytics (e.g., Lévy Flight for CS and Dirichlet Distribution for LDA)

- Think about “How we, humans make decisions” and translate the logic into a model