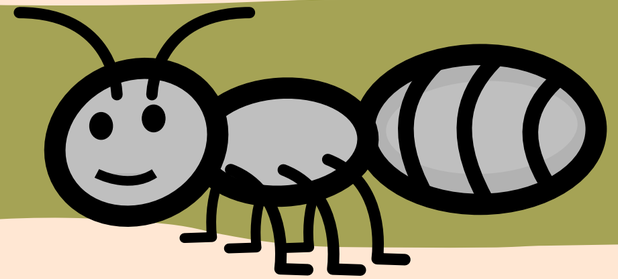
Ant colony optimiztion



combinatorial optimiztion

Combinatorial optimization is the subject of finding an optimal object from a finite set of objects.

In many such problems, exhaustive search is not tractable.

the domain of combinatorial optimization problem is the set of discrete feasible solutions

and in which the goal is to find the best solution.

Typical problems in this field are the traveling salesman problem knapsack and more.

The travelling salesman problemhe

The traveling salesman problem (TSP) asks the following question:
Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

It is an NP-hard problem in combinatorial optimization

The problem was first formulated in 1930 and is one of the most intensively studied problems in optimization.

Even though the problem is computationally difficult, many heuristics and exact algorithms are known, so that some instances with tens of thousands of cities can be solved completely and even problems with millions of cities can be approximated within a small fraction of 1%

metaheuristic algorithm approuch

in greek, meta (beyond, at a higher level) heuristic (to search, to discover)

a metaheuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem

metaheuristics do not guarantee a globally optimal solution and usually in their core are stochastic methods.

Advantages:

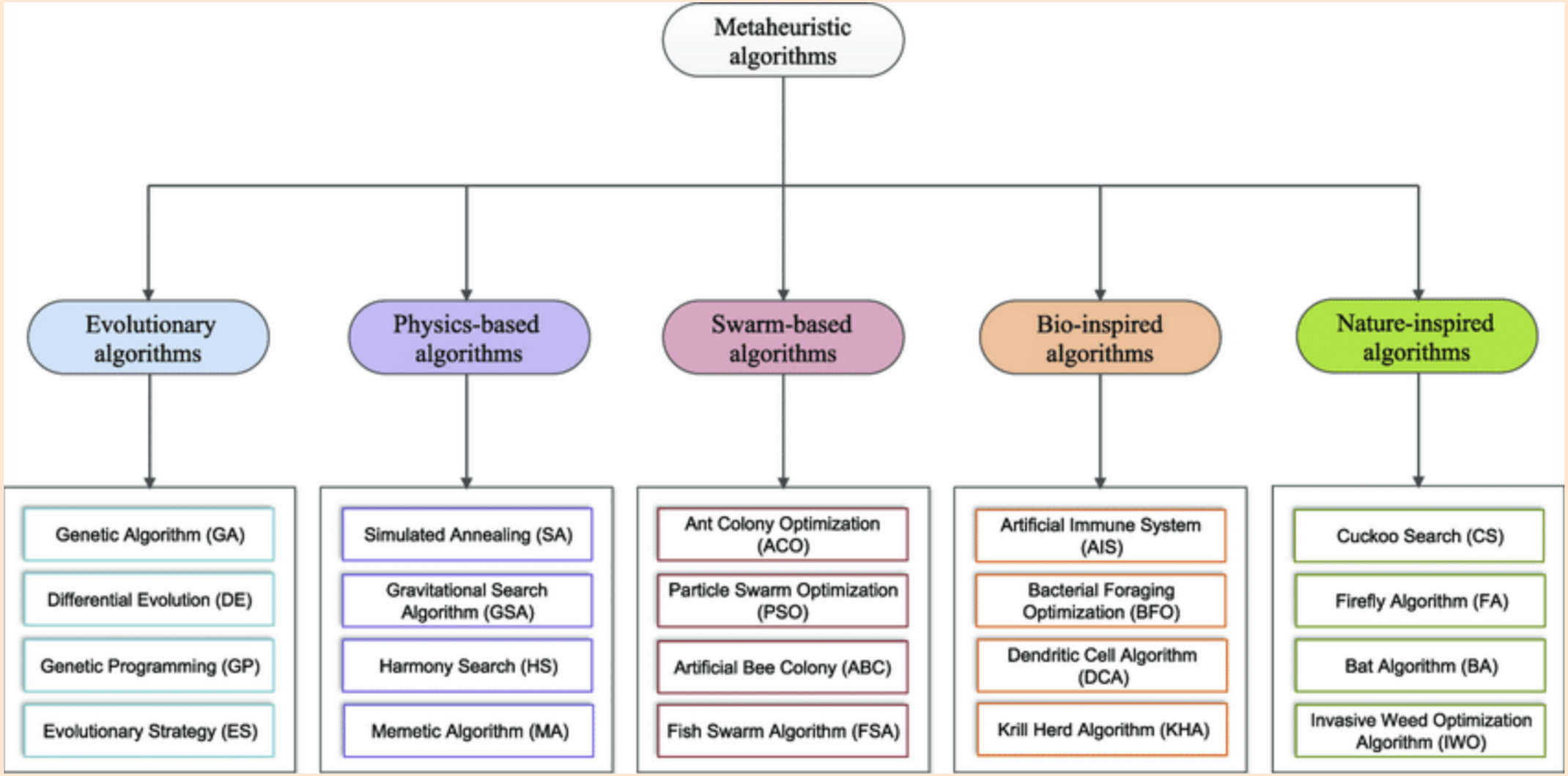
- General-purpose
- Successful in practice
- Easy implementation

Advantages:

- General-purpose
- Successful in practice
- Easy implementation

Drawbacks:

- they are not exact algorithms
- they are not deterministic
- poor theoretical foundations


Metaheuristic algorithms
Evolutionary algorithms
Physics-based algorithms
Swarm-based algorithms
Bio-inspired algorithms
Nature-inspired algorithms
Genetic Algorithm (GA)
Differential Evolution (DE)
Genetic Programming (GP)
Evolutionary Strategy (ES)
Simulated Annealing (SA)
Gravitational Search Algorithm (GSA)
Harmony Search (HS)
Memetic Algorithm (MA)
Ant Colony Optimization (ACO)
Particle Swarm Optimization (PSO)
Artificial Bee Colony (ABC)
Fish Swarm Algorithm (FSA)
Artificial Immune System (AIS)
Bacterial Foraging Optimization (BFO)
Dendritic Cell Algorithm (DCA)
Krill Herd Algorithm (KHA)
Cuckoo Search (CS)
Firefly Algorithm (FA)
Bat Algorithm (BA)
Invasive Weed Optimization Algorithm (IWO)

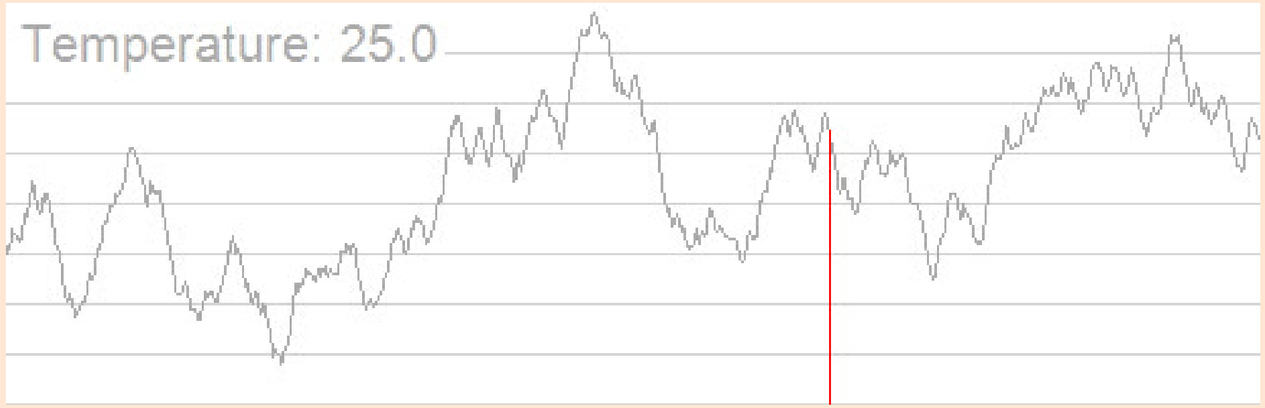
nature influenced algorithms

many metaheuristics are inspired by nature.

simulated annealing (pyshics)

The name of the algorithm comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material.

This notion of slow cooling implemented in the simulated annealing algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored



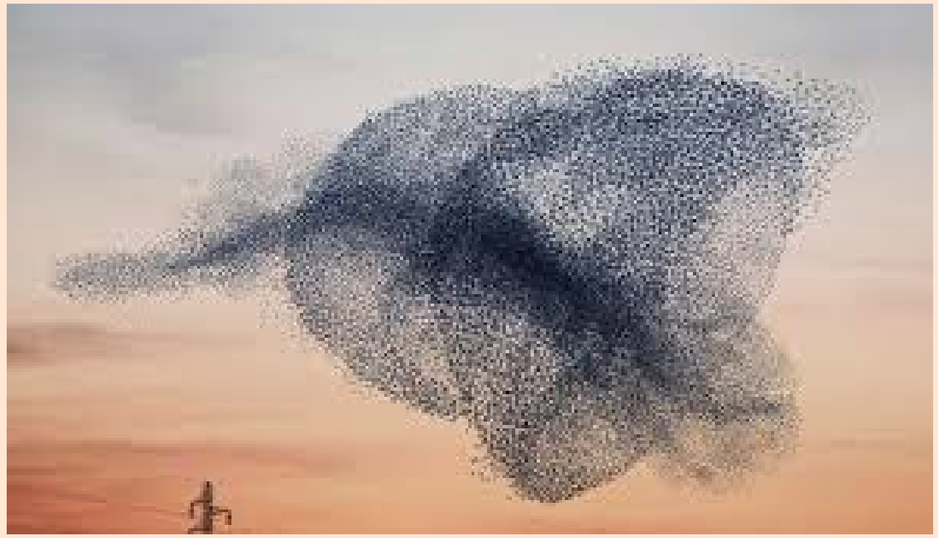
genetic algorithm (evolutionary)

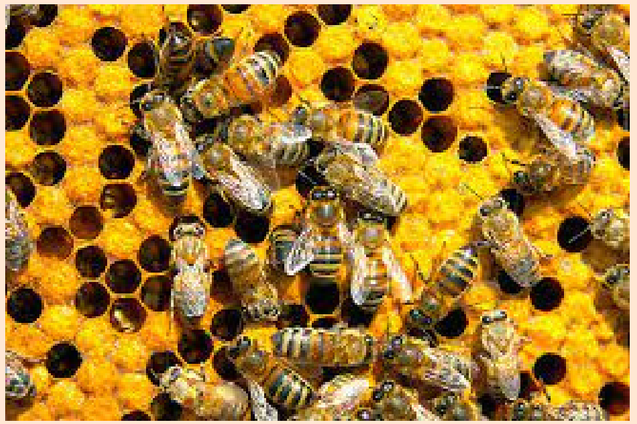
is an Evolutionary algorithm.
we usually start from a population of randomly generated individuals (solutions)
In each generation (iteration of the algorithm), the fitness of every individual in the population is evaluated.
the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation.

A typical genetic algorithm requires:

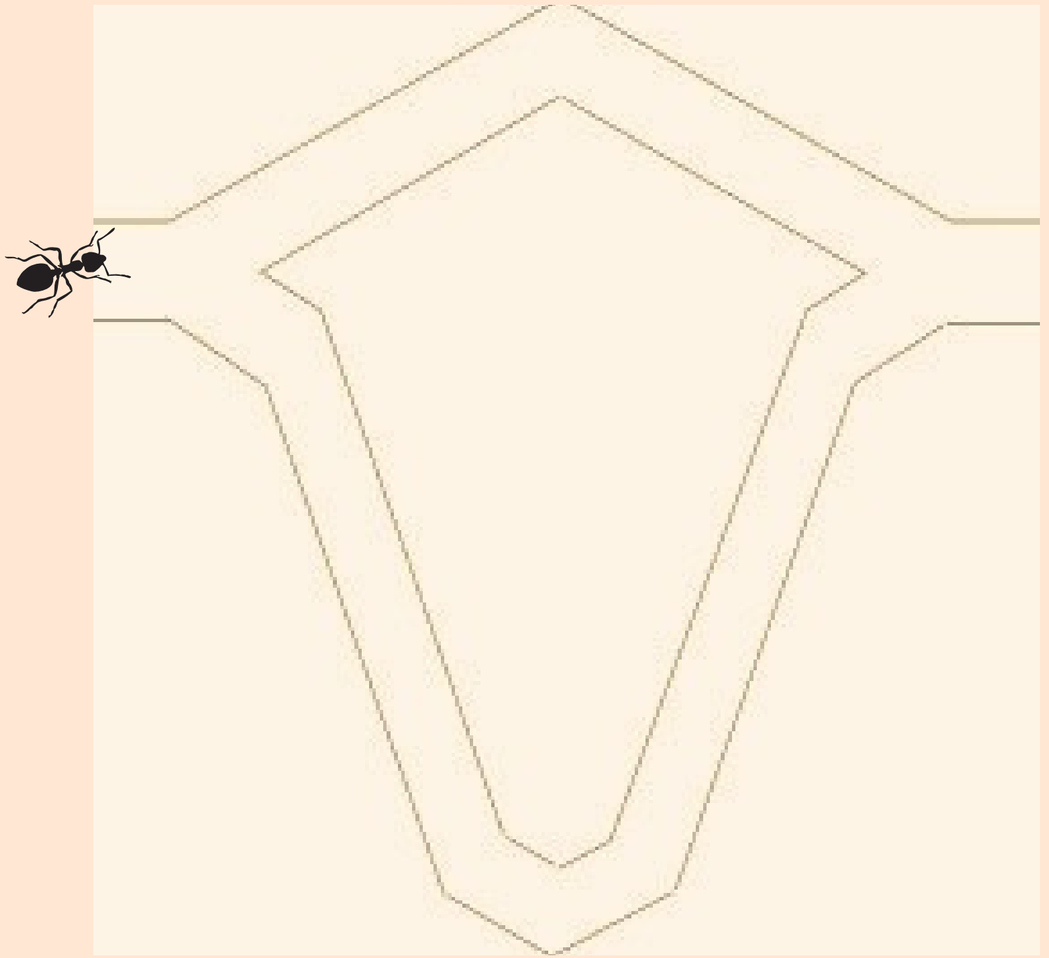
1. a genetic representation of the solution domain,
2. a fitness function to evaluate the solution domain.

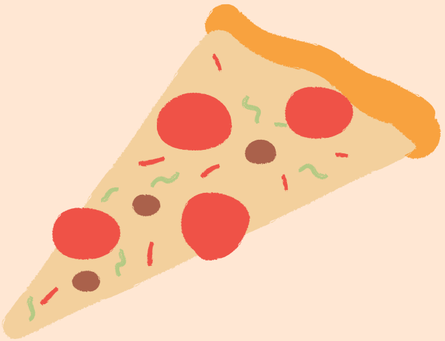
Swarm intelligence



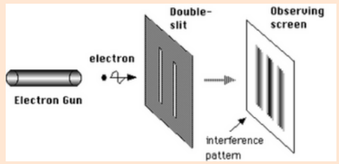


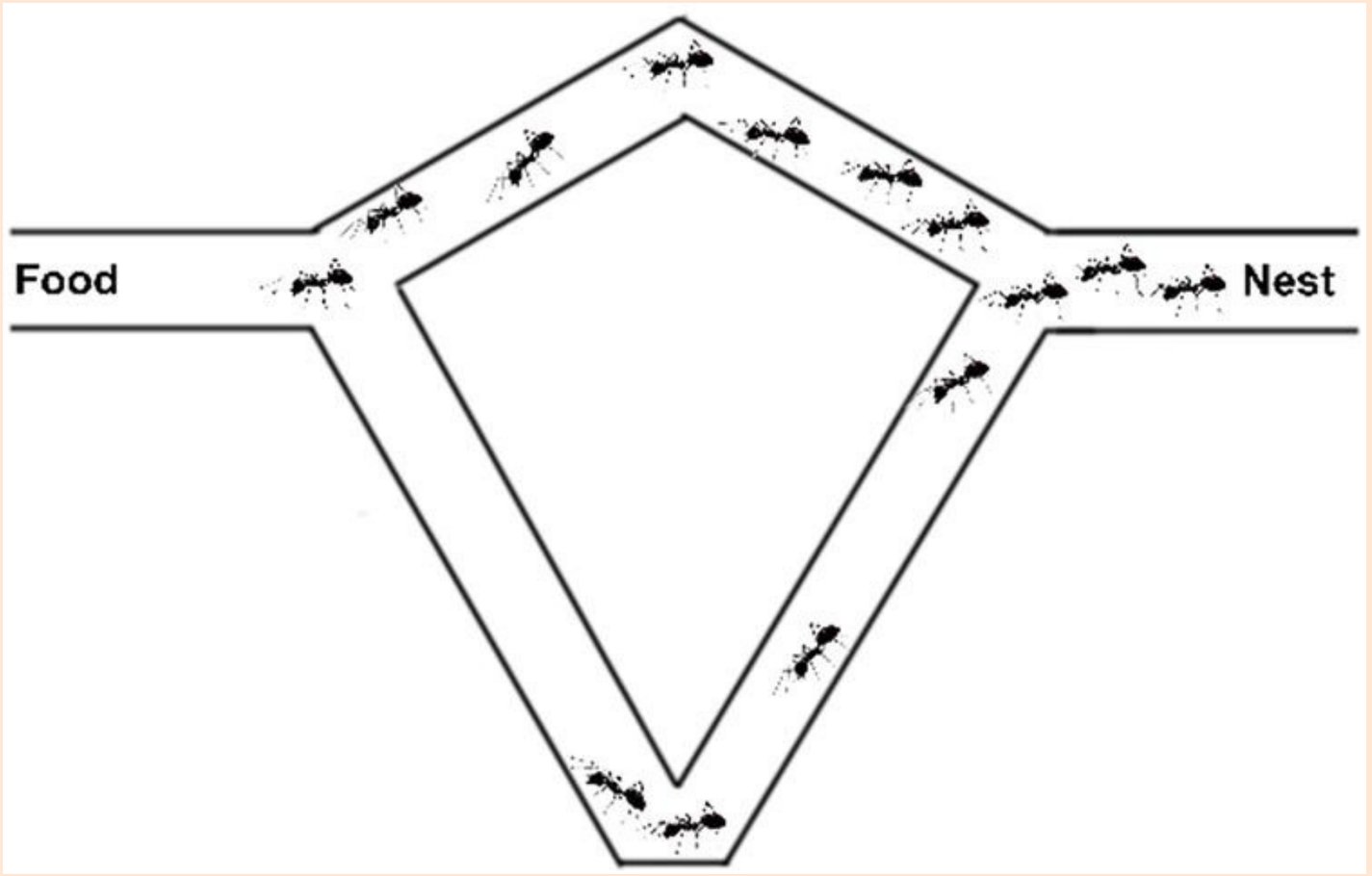
the double bridge experiment

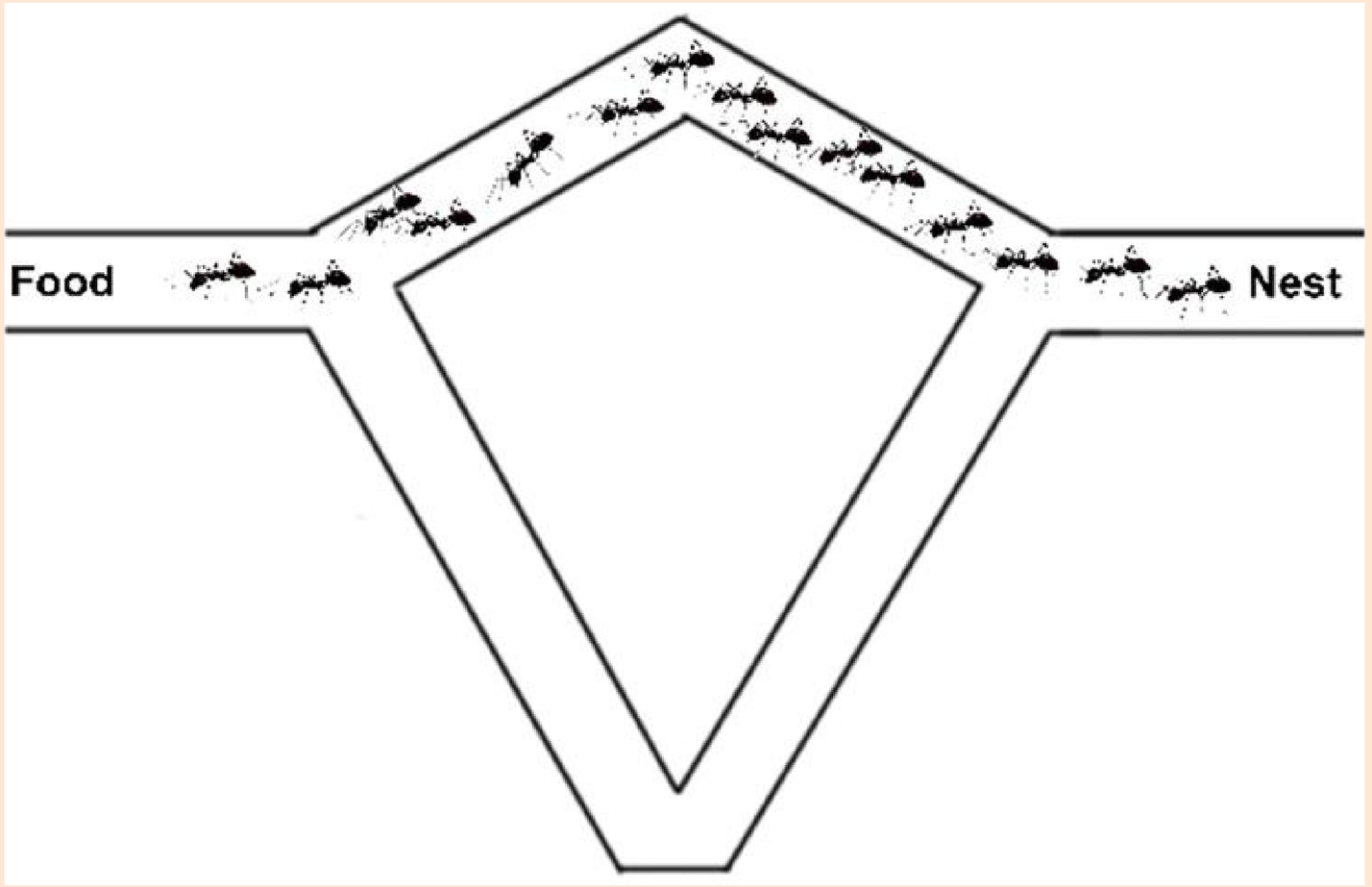




not slits




Food
Nest

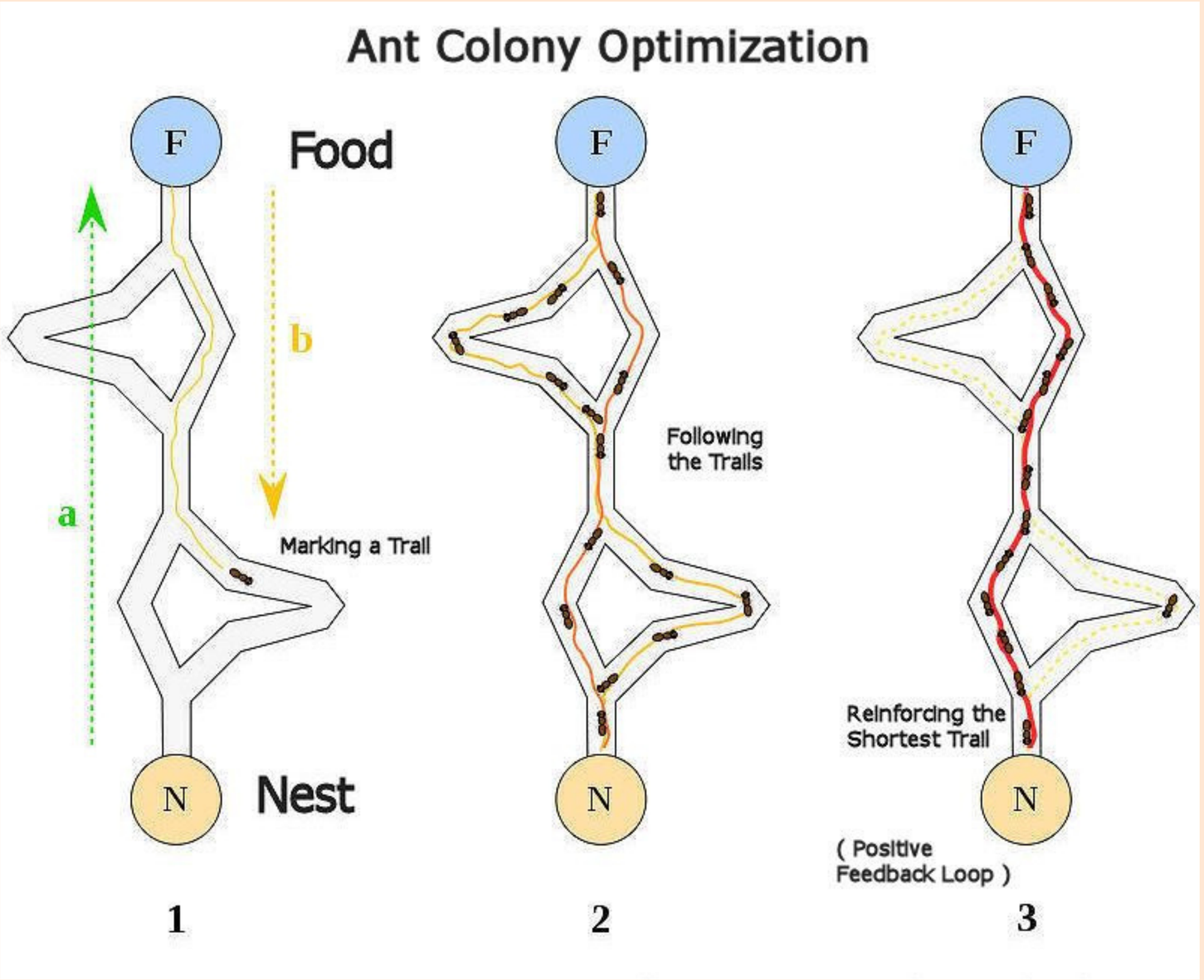

Food
Nest

pheromons

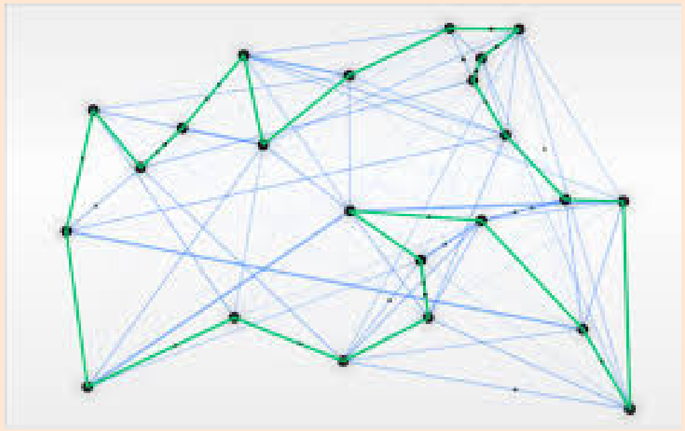
A pheromone is a chemical that an animal produces which changes the behavior of another animal of the same species.

In many ant species, individual ants may deposit pheromones on the ground while walking.
By depositing pheromone, ants create a trail that is used,
for example, to mark the path from the nest to food sources and back.
Foragers can sense the pheromone trails and follow the path to food discovered by other ants.
Several ant species are capable of exploiting pheromone trails to determine the shortest among the available paths leading to the food.

pheromons

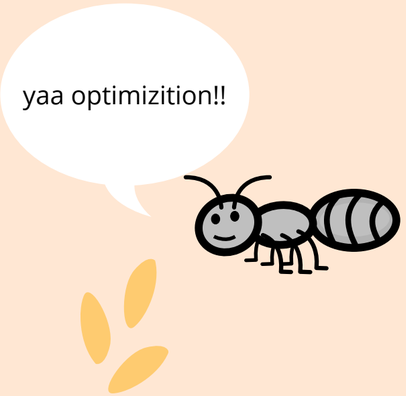

Ant Colony Optimization
F
Food
b
a
Marking a Trail
N
Nest
1
F
Following
the Trails
N
2
F
Reinforcing the
Shortest Trail
N
(Positive
Feedback Loop)
3

Ant Colony optimization algorithem



Initially proposed by Marco Dorigo in 1992 in his Ph.D. thesis.
the first algorithm was aiming to search for an optimal path in a graph.

The ACO algorithm is an optimization algorithm, recognized for being very efficient in problems of finding routes and planning paths in roads. In terms of the problem of the traveling salesman, ACO algorithm has been able to find optimal solutions to the problem



the general algorithem

```
procedure ACO_MetaHeuristic is
    while not terminated do
        generateSolutions()
        Local_search()
        pheromoneUpdate()
    repeat
end procedure
```

the algorithem has 3 main parts.

1. a group of ants generates feasible solutions by traversing the graph
2. Localsearch() - usually a local search to generate better solutions also known as deamon action
3. update the pheromones on the edges of the graph increasing the amount at the good solutions path and evaporate pheromones by some rate from all the edges

generate solutions

single ant next node local selection

To select the next node in its tour, an ant will consider the length of each edge available from its current position, as well as the corresponding pheromone level

each ant k computes a set Ak(x) of feasible expansions to its current state in each iteration, and moves to one of these in probability.

$$p_{xy}^k = \frac{(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}{\sum_{z \in \text{allowed}_x} (\tau_{xz}^\alpha)(\eta_{xz}^\beta)}$$

generate solutions

tau is the amount of pheromone deposited for transition from state **x** to **y (xy).**

alpha is a parameter to control the influence of **tau** at **xy.**

eta is the desirability of state transition **xy** (a priori knowledge, typically **1/d** at **xy**, where **d** is the distance).

beta is a parameter to control the influence of **eta** at **xy**.

$$p_{xy}^{k} = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\sum_{z \in \text{allowed}_x} (\tau_{xz}^{\alpha})(\eta_{xz}^{\beta})}$$

generate solutions

ants full path solution

1. an ant will keep touring the graph until no more feasible nodes can be visited.
while choosing the next node by its probability

2. iterate this process for **m** ants (usually 10 - 20 ants)

local search (Deamon action)

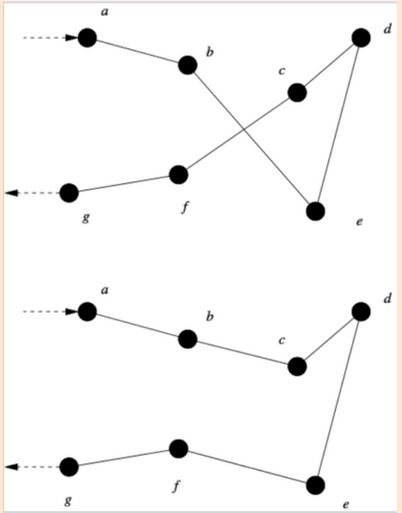
local search is a heuristic method for solving optimization problems.

Local search algorithms move from solution to solution in the space of candidate solutions (the search space) by applying local changes

the 2-opt algorithm is a simple local search algorithm for solving the traveling salesman problem.

The main idea behind it is to take a route that crosses over itself and reorder it so that it does not.

A complete 2-opt local search will compare every possible valid combination of the swapping mechanism.



update pheromones

Trails are usually updated when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions

ρ is the evaporation rate,

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k}^{m} \Delta\tau_{xy}^{k}$$

update pheromones

Trails are usually updated when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions

ρ is the evaporation rate,

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k}^{m} \Delta\tau_{xy}^{k}$$

$$\Delta\tau_{xy}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

where L_k is the cost of the kth ant's tour (typically length) and Q is a constant.

improvements to the algorithm

TABLE 1 **A non-exhaustive list of successful ant colony optimization algorithms (in chronological order).**

ALGORITHM	AUTHORS	YEAR	REFERENCES
ANT SYSTEM (AS)	DORIGO ET AL.	1991	[6]—[8]
ELITIST AS	DORIGO ET AL.	1992	[7], [8]
ANT-Q	GAMBARDELLA & DORIGO	1995	[9]
ANT COLONY SYSTEM	DORIGO & GAMBARDELLA	1996	[10]—[12]
$\mathcal{MAX}$-$\mathcal{MIN}$ AS	STÜTZLE & HOOS	1996	[13]—[15]
RANK-BASED AS	BULLNHEIMER ET AL.	1997	[16], [17]
ANTS	MANIEZZO	1999	[18]
BWAS	CORDON ET AL.	2000	[19]
HYPER-CUBE AS	BLUM ET AL.	2001	[20], [21]

more improvments..

Problem type	*Problem name*	*Authors*	*Year*	*References*
Routing	Traveling salesman	Dorigo et al.	1991, 1996	[64, 65]
		Dorigo & Gambardella	1997	[61]
		Stützle & Hoos	1997, 2000	[147, 148]
	TSP with time windows	López Ibáñez et al.	2009	[102]
	Sequential ordering	Gambardella & Dorigo	2000	[76]
	Vehicle routing	Gambardella et al.	1999	[77]
		Reimann et al.	2004	[127]
		Favoretto et al.	2007	[72]
		Fuellerer et al.	2009	[73]
	Multicasting	Hernández & Blum	2009	[91]
Assignment	Quadratic assignment	Maniezzo	1999	[105]
		Stützle & Hoos	2000	[148]
	Frequency assignment	Maniezzo & Carbonaro	2000	[106]
	Course timetabling	Socha et al.	2002,2003	[138, 139]
	Graph coloring	Costa & Hertz	1997	[39]
Scheduling	Project scheduling	Merkle et al.	2002	[111]
	Weighted tardiness	den Besten et al.	2000	[42]
		Merkle & Middendorf	2000	[109]
	Flow shop	Stützle	1997	[142]
		Rajendran, Ziegler	2004	[125]
	Open shop	Blum	2005	[18]
	Car sequencing	Solnon	2008	[140]
Subset	Set covering	Lessing et al.	2004	[101]
	l-cardinality trees	Blum & Blesa	2005	[20]
	Multiple knapsack	Leguizamón & Michalewicz	1999	[100]
	Maximum clique	Solnon, Fenet	2006	[141]
Machine learning	Classification rules	Parpinelli et al.	2002	[124]
		Martens et al.	2006	[107]
		Otero et al.	2008	[121]
	Bayesian networks	Campos, Fernández-Luna	2002	[40, 41]
	Neural networks	Socha, Blum	2007	[135]
Bioinformatics	Protein folding	Shmygelska & Hoos	2005	[132]
	Docking	Korb et al.	2006	[97, 98]
	DNA Sequencing	Blum et al.	2008	[25]
	Haplotype Inference	Benedettini et al.	2008	[11]

Ant Colony System (ASC)

In the ant colony system algorithm, the original ant system was modified in three aspects:

Ant Colony System (ASC)

In the ant colony system algorithm, the original ant system was modified in three aspects:

1. the edge selection is biased towards exploitation (favoring the selection of the shortest edges with a large amount of pheromone)

Ant Colony System (ASC)

In the ant colony system algorithm, the original ant system was modified in three aspects:

1. the edge selection is biased towards exploitation (favoring the selection of the shortest edges with a large amount of pheromone)
2. while building a solution, ants change the pheromone level of the edges they are selecting by applying a local pheromone updating rule.

Ant Colony System (ASC)

In the ant colony system algorithm, the original ant system was modified in three aspects:

1. the edge selection is biased towards exploitation (favoring the selection of the shortest edges with a large amount of pheromone)
2. while building a solution, ants change the pheromone level of the edges they are selecting by applying a local pheromone updating rule.
3. at the end of each iteration, only the best ant is allowed to update the trails by applying a modified global pheromone updating rule.

1. State transition rule

In the ACS the state transition rule is as follows: an ant positioned on a node chooses the city to move to by applying the rule given by:

$$s = \begin{cases} \arg\ \max_{u \in J_k(r)} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\}, & \\ \qquad \text{if } q \leq q_0 & \text{(exploitation)} \\ S, \qquad \text{otherwise} & \text{(biased exploration)} \end{cases}$$

where q0 is aparameter s,t 0 < q0 < 1and q is choosen from a uniform distribution between 0 to 1.

The parameter determines the relative importance of exploitation versus exploration: every time an ant in a city has to choose a city to move to, it samples a random number. If q<=q0 the best edge is chosen (exploitation), otherwise an edge is chosen according to AS previews function (biased exploration).

2. local pheromone update

The most interesting contribution of ACS is probably the introduction of a **local pheromone update** in addition to the pheromone update performed at the end of the construction process.

The local pheromone update is performed by all the ants after each construction step by the rules:

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s)+\rho\cdot\Delta\tau(r,s)$$

$$\tau_0 = (n\cdot L_{nn})^{-1}$$

where n is the number of nodes and Lnn is the total distance of a greedy TSP algorithm

The main goal of the local update is to diversify the search performed by subsequent ants during an iteration by decreasing the pheromone concentration on the traversed edges.

3. ACS Global Updating Rule

only the globally best ant (i.e., the ant which constructed the shortest tour from the beginning of the trial) is allowed to deposit pheromone. This choice, together with the use of the first rule, is intended to make the search more directed: ants search in a neighborhood of the best tour found

The pheromone level is updated by applying the global updating rule of:

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s)$$

where

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1}, & \text{if } (r,s) \in \text{global-best-tour} \\ 0, & \text{otherwise} \end{cases}$$

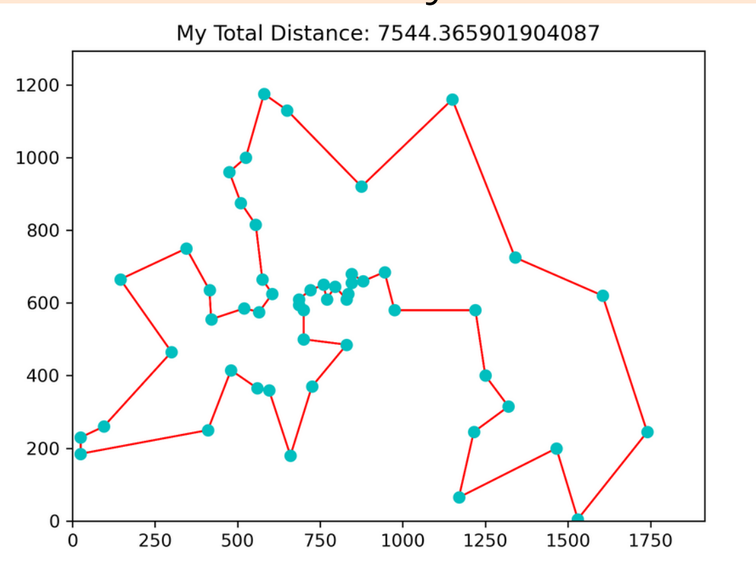
results compersion

A known set of real coordinantes in real cities

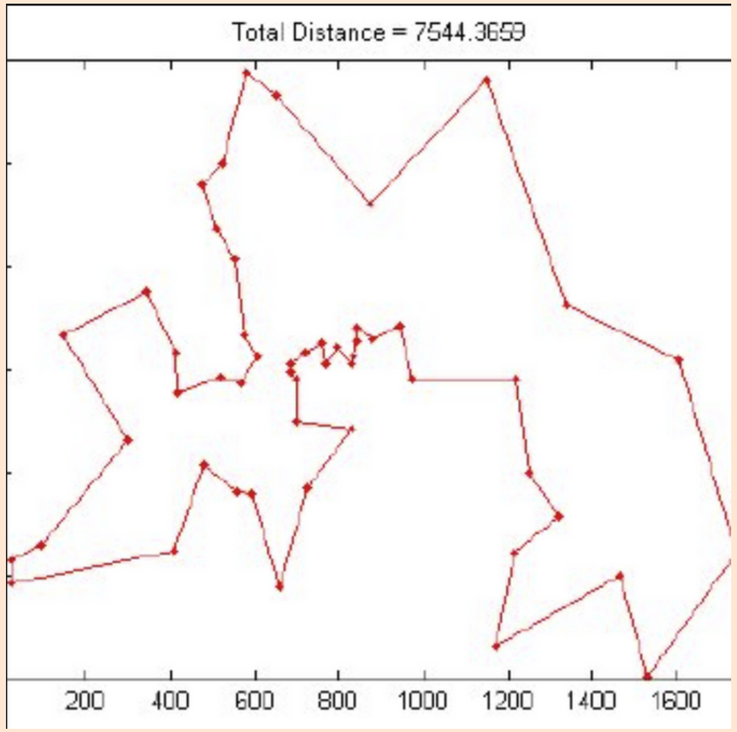
link to the dataset of the coordinantes
http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/

Benchmark	Cities Number	Optimal solution
A280	280	2579
Berlin52	52	7542
Ch130	130	6110
Ch150	150	6528
D1291	1291	50801
Gil262	262	2378
Rat195	195	2323
Rat575	575	6773
Tsp225	225	3916

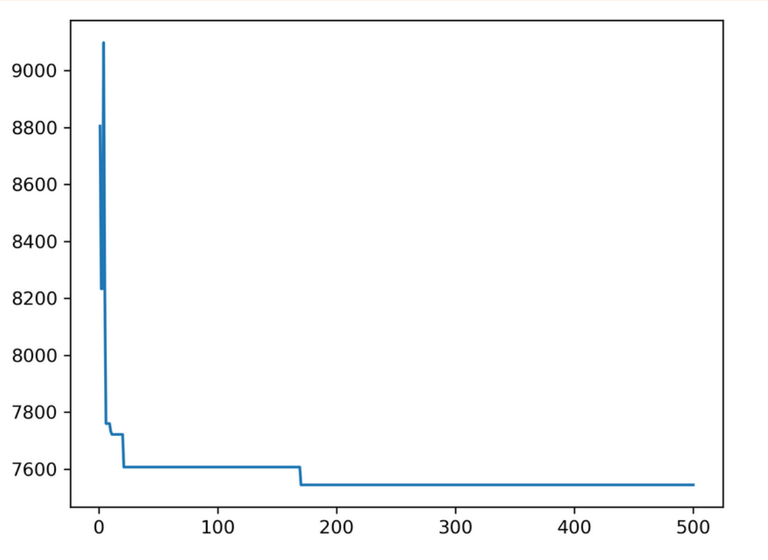
berlin52 my result



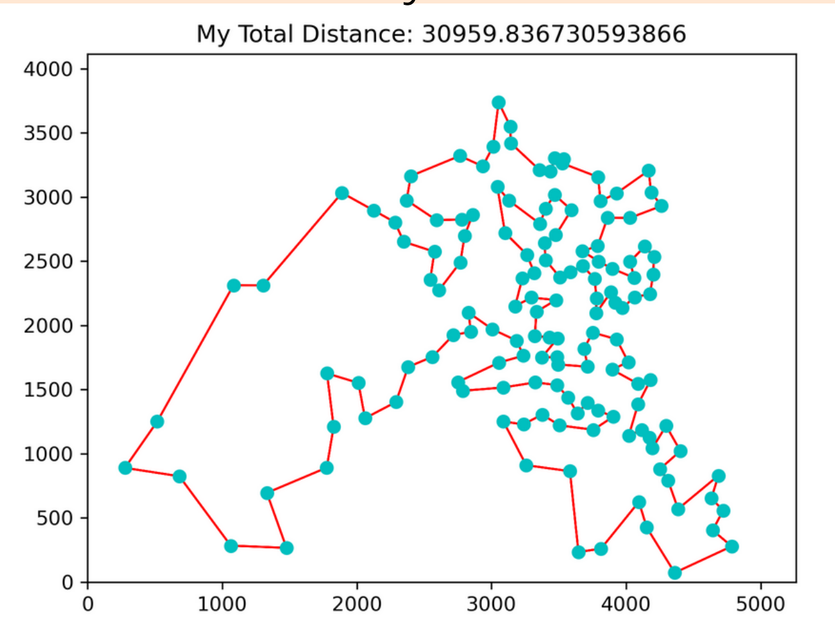
berlin52 real optimal



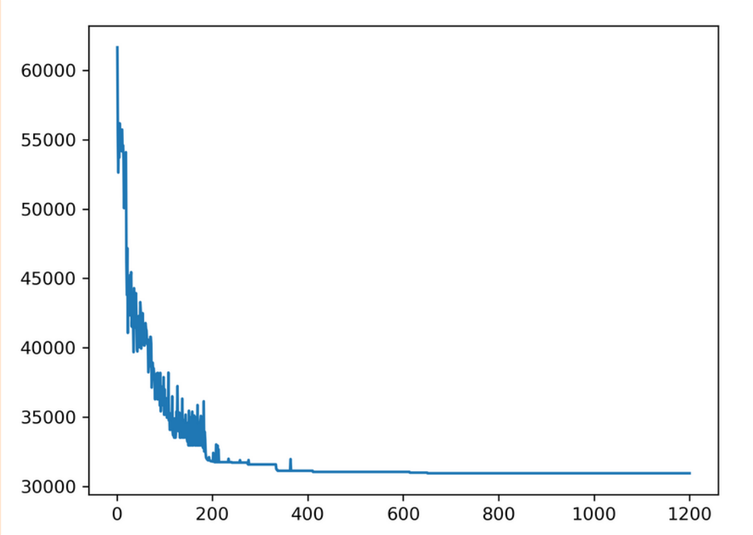
tour distance as a function
of the number of itrations



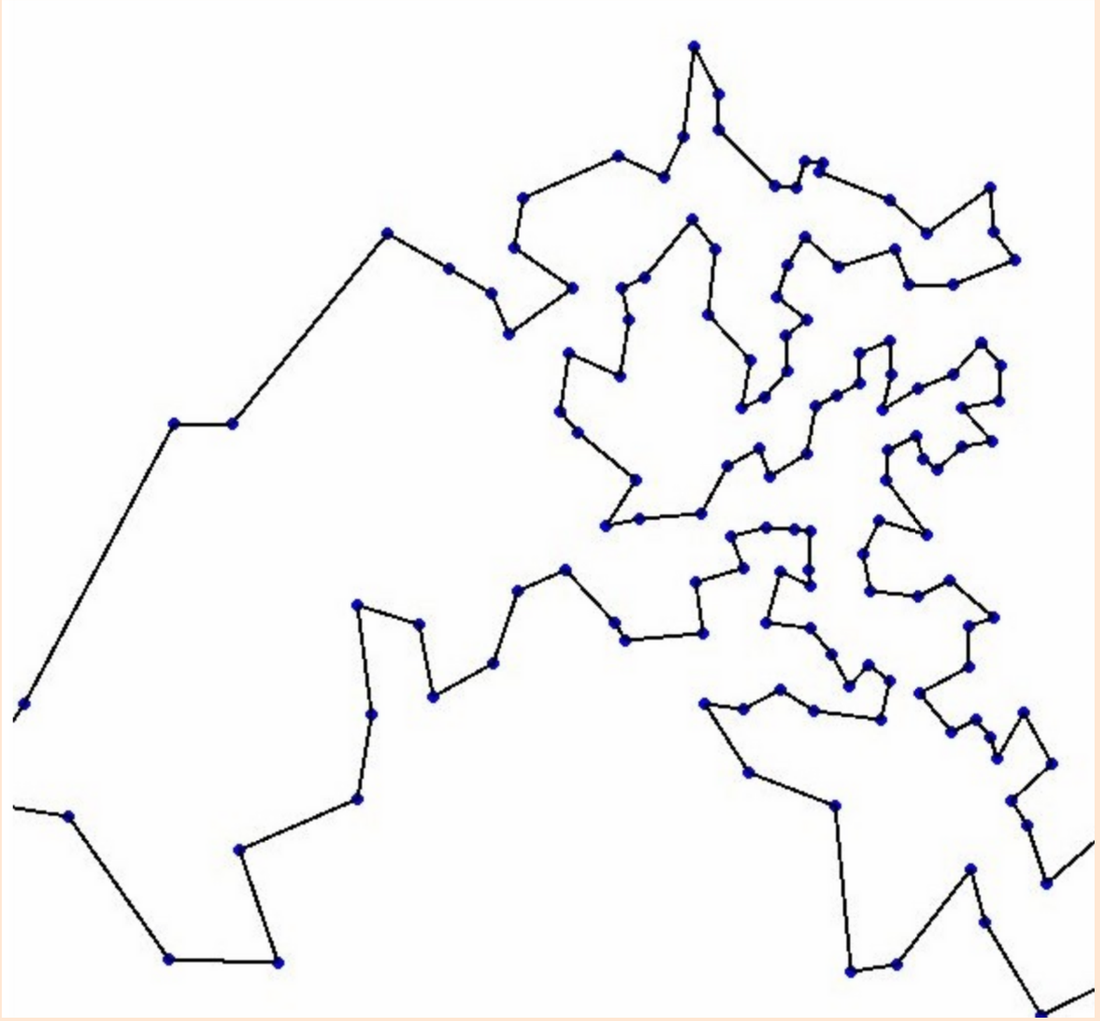
chn144 my result



tour distance as a function
of the number of itrations

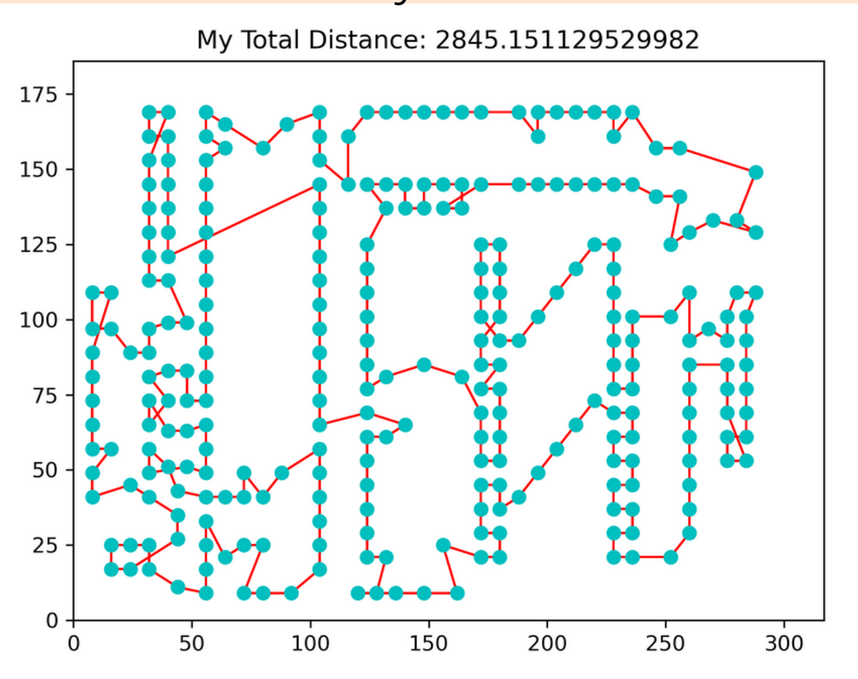


chn144 real optimal

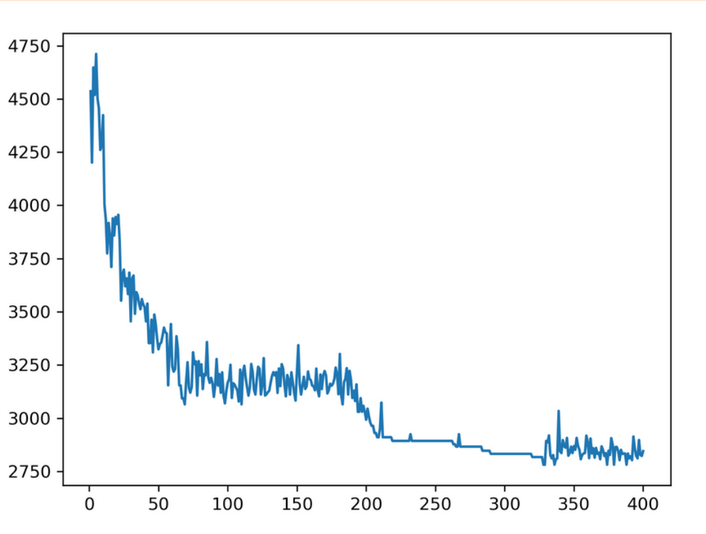


Optimal total distance: 30445

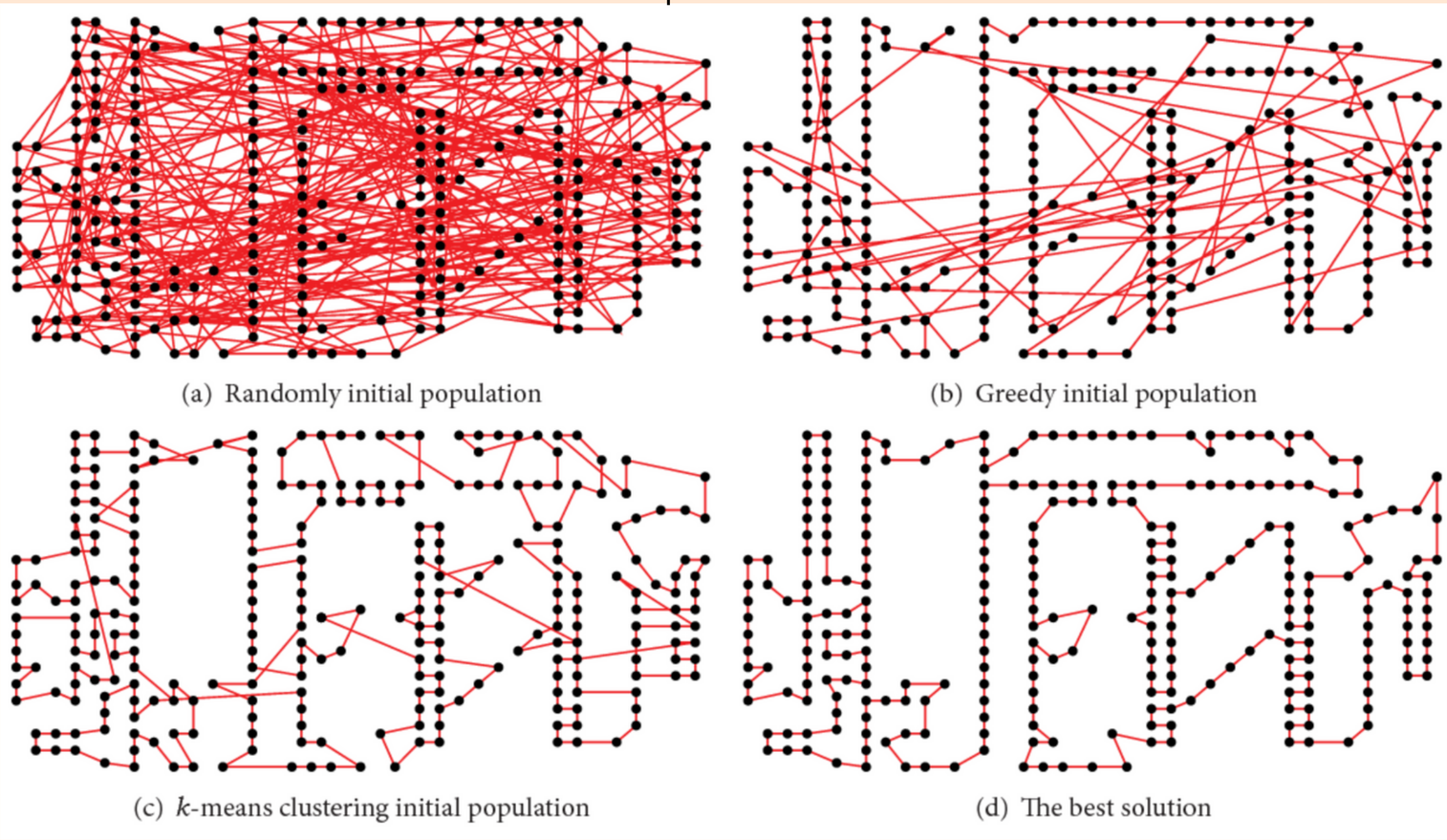
A280 my result



tour distance as a function of the number of itrations



Real optimal: 2579



Smart Random

Spotify smoothest playlist playing order

The main idea:

to generate a good smooth playing order of the songs in a given playlist.

we can do so by **measuring how similar two songs are** and give this property a scaler number.

in other words, **we can measure the distance between any 2 songs,**

so we can come back to the TSP problem.

but in this case, we will minimize the changes between songs that sound different, by that make our playing order sound smooth and with less abrupt changes.

the songs features

Spotify uses some features from the audio tracks to analyze the songs
with this data, we can measure the distance between each of the songs

Python can be used to acquire the data using the library Spotipy

the songs features

some of the features:

- Python can be used to acquire the data using the library Spotipy

the songs features

some of the features:

bpm: beat per minutes the tempo of the song

- Python can be used to acquire the data using the library Spotipy

the songs features

some of the features:

- Python can be used to acquire the data using the library Spotipy

bpm: beat per minutes the tempo of the song

Instrumentalness: This value represents a measure of the vocals in the song. The closer it is to 1.0, the more instrumental the song is.

the songs features

some of the features:

- Python can be used to acquire the data using the library Spotipy

bpm: beat per minutes the tempo of the song

Instrumentalness: This value represents a measure of the vocals in the song. The closer it is to 1.0, the more instrumental the song is.

Acousticness: This value describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.

the songs features

some of the features:

- Python can be used to acquire the data using the library Spotipy

bpm: beat per minutes the tempo of the song

Instrumentalness: This value represents a measure of the vocals in the song. The closer it is to 1.0, the more instrumental the song is.

Acousticness: This value describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.

Liveness: This value describes the probability that the song was recorded with a live audience. According to the official documentation “a value above 0.8 provides strong likelihood that the track is live”.

the songs features

some of the features:

- Python can be used to acquire the data using the library Spotipy

bpm: beat per minutes the tempo of the song

Instrumentalness: This value represents a measure of the vocals in the song. The closer it is to 1.0, the more instrumental the song is.

Acousticness: This value describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.

Liveness: This value describes the probability that the song was recorded with a live audience. According to the official documentation “a value above 0.8 provides strong likelihood that the track is live”.

Speechiness: “Speechiness detects the presence of spoken words in a track”. If the speechiness of a song is above 0.66, it is probably made of spoken words, a score between 0.33 and 0.66 is a song that may contain both music and words, and a score below 0.33 means the song does not have any speech.

the songs features

- Python can be used to acquire the data using the library Spotipy

some of the features:

bpm: beat per minutes the tempo of the song

Instrumentalness: This value represents a measure of the vocals in the song. The closer it is to 1.0, the more instrumental the song is.

Acousticness: This value describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.

Liveness: This value describes the probability that the song was recorded with a live audience. According to the official documentation “a value above 0.8 provides strong likelihood that the track is live”.

Speechiness: “Speechiness detects the presence of spoken words in a track”. If the speechiness of a song is above 0.66, it is probably made of spoken words, a score between 0.33 and 0.66 is a song that may contain both music and words, and a score below 0.33 means the song does not have any speech.

Danceability: “Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable”.

Test example

Test example

Playlist of 23 songs:

[Everlong, Miss Murder, A$AP Forever (feat. Moby), In Da Club, All About That Bass, P.I.M.P. , Ms. Jackson , Hey, Soul Sister, Shape of You, Viva La Vida, In the End, Oops!...I Did It Again, Get Lucky (feat. Pharrell Williams & Nile Rodgers), I Gotta Feeling, Dance Monkey, Blinding Lights, Candy Shop, Rolling In The Deep, Stronger (What Doesn\'t Kill You), Party Rock Anthem, Faint, Thnks fr th Mmrs, Bring Me To Life]

Test example

Playlist of 23 songs:

[Everlong, Miss Murder, A$AP Forever (feat. Moby), In Da Club, All About That Bass, P.I.M.P. , Ms. Jackson , Hey, Soul Sister, Shape of You, Viva La Vida, In the End, Oops!...I Did It Again, Get Lucky (feat. Pharrell Williams & Nile Rodgers), I Gotta Feeling, Dance Monkey, Blinding Lights, Candy Shop, Rolling In The Deep, Stronger (What Doesn\'t Kill You), Party Rock Anthem, Faint, Thnks fr th Mmrs, Bring Me To Life]

The playlist in "smooth" order:

['Rolling In The Deep', 'Hey, Soul Sister', 'Oops!...I Did It Again', 'Shape of You', 'Dance Monkey', 'Faint', 'Bring Me To Life', 'Blinding Lights', 'Everlong', 'Thnks fr th Mmrs', 'Miss Murder', "Stronger (What Doesn't Kill You)", 'In the End', 'A$AP Forever (feat. Moby)', 'Viva La Vida', 'Party Rock Anthem', 'I Gotta Feeling', 'Candy Shop', 'In Da Club', 'Get Lucky (feat. Pharrell Williams & Nile Rodgers)', 'All About That Bass', 'P.I.M.P.', 'Ms. Jackson']

thank you