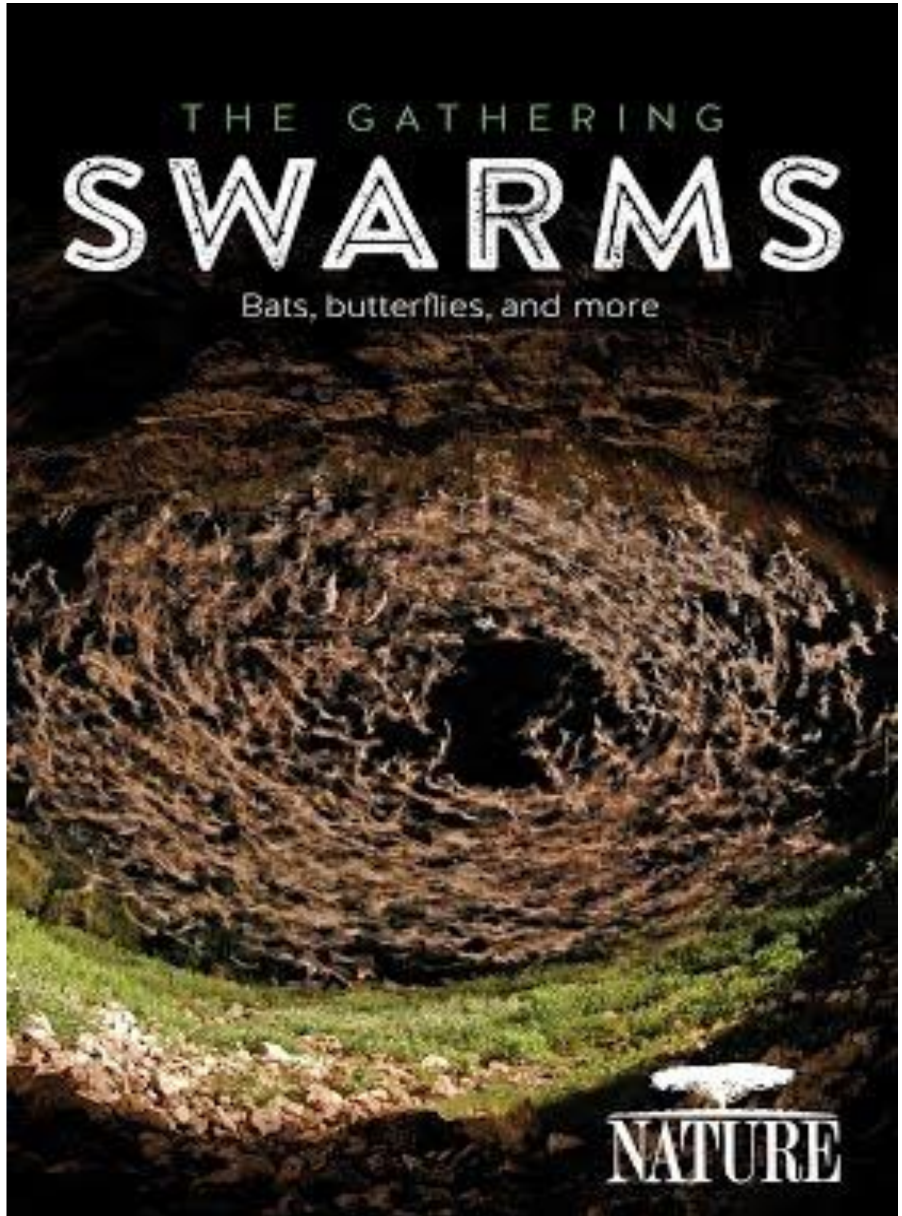
Computational Intelligence

Unit # 6



http://www.dailymotion.com/video/x5sa5hr

Swarms

- It is about individuals cooperating (knowingly or not) to achieve a definite goal. Such as,
  - ants finding the shortest path between their nest and a good source of food
  - bees finding the best sources of nectar within the range of their hive
  - Termites making huge mounds
  - Birds flocking
  - Fish schooling

- https://www.macs.hw.ac.uk/~dwcorne/Teaching/SIchapterforHandbook_NC.pdf

Swarm behaviors in animals

- Food foraging
- Flocking/Schooling
- Clustering
- Natural Construction

Key Principles

- Social Insects work without supervision.
- Individuals are largely homogenous
- Individuals act asynchronously in parallel;
- Self-organization: activities are neither centrally controlled nor locally supervised
- Although these interactions might be primitive, taken together they result in efficient solutions to difficult problems
- Flexibility: the colony can adapt to a changing environment
- Robustness: even when one or more individual fail, the group can still perform its task
- communication between agents is largely effected by some form of ‘stigmergy’

Stigmergy

- Stigmergy means communication via signs or cues placed in the environment by one entity, which affect the behavior of other entities who encounter them.
- Stigmergy is the name for the indirect communication that seems to underpin cooperation among social insects.

Stigmergy

- Stigmergy is a form of self-organization. It produces complex, seemingly intelligent structures, without need for any planning, control, or even direct communication between the agents.
- As such it supports efficient collaboration between extremely simple agents, who lack any memory, intelligence or even individual awareness of each other.

Stigmergy

- For example, ants exchange information by laying down pheromones (the trace) on their way back to the nest when they have found food.

Termites

- Termite mounds are among the largest structures built by any non-human. They reach as high as thirty feet, which, proportional to the insects' tiny size, is the equivalent of our building something twice as tall as the 2,722-foot Burj Khalifa, in Dubai.

https://www.newyorker.com/magazine/2018/09/17/what-termites-can-teach-us

Termites

- The interior of a termite mound is an intricate structure of interweaving tunnels and passageways, radiating chambers, galleries, archways, and spiral staircases.

Termites

- Termites use pheromones to build their complex nests by following a simple decentralized rule set. Each insect scoops up a 'mudball' or similar material from its environment, invests the ball with pheromones, and deposits it on the ground, initially in a random spot.
- However, termites are attracted to their nestmates' pheromones and are therefore more likely to drop their own mudballs on top of their neighbors'. The larger the heap of mud becomes, the more attractive it is, and therefore the more mud will be added to it (positive feedback). Over time this leads to the construction of pillars, arches, tunnels and chambers.

https://www.newyorker.com/magazine/2018/09/17/what-termites-can-teach-us

Swarm Intelligence and Computer Science

- **Ant Colony Optimization (ACO)**


Inspired by food foraging behavior of ants/bees

- **Particle Swarm Optimization (PSO)**


Inspired by bird flocking and fish schooling

Videos

- Lens of Time: Secrets of Schooling
- https://www.youtube.com/watch?v=Y-5ffl5_7AI&t=107s
- Starlings Murmuration https://www.youtube.com/watch?v=eakKfY5aHmY&t=145s

Ant Colony Optimization (ACO)

Interesting Facts About Ants (Source: Engelbrecht)

- Ants appeared on earth some 100 million years ago, and have a current total population estimated at 10^{16} individuals.
- Most of these ants are social insects, living in colonies of 30 to millions of individuals.

The Social Conquest of Earth,

Ant Forging Behavior

- Studies of the foraging behavior of several species of real ants revealed an initial random or chaotic activity pattern in the search for food.
- As soon as a food source is located, activity patterns become more organized with more and more ants following the same path to the food source.
- “Auto-magically”, soon all ants follow the same, shortest path.
- This emergent behavior is a result of a recruitment mechanism whereby ants that have located a food source influence other ants towards the food source.
- The recruitment mechanism differs for different species, and can either be in the form of direct contact, or indirect “communication.”

Simulation

- http://netlogoweb.org/launch#http://netlogoweb.org/assets/modelslib/Curricular%20Models/GenEvo/GenEvo%203%20Genetic%20Drift%20and%20Natural%20Selection.nlogo

Working of Ant Colonies

- The colony's efficient behavior emerges from the collective activity of individuals following two very simple rules:
  - Lay pheromone
  - Follow the trails of others

Working of Ant Colonies

- In a simple case, two ants leave the nest at the same time and take different paths to a food source, marking their trail with pheromone.
- The ant that took the shortest path will return first, and this trail will now be marked with twice as much pheromone (from the nest to the food and back) as the path taken by the second ant, which has yet to return.
- Their nest mates will be attracted to the shorter path because of its higher concentration of pheromone.
- As more and more ants take that route, they too lay pheromone, further amplifying the attractiveness of the shorter trail.

Working of Ant Colonies

- The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate.
- A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.
- Pheromone evaporation has also the advantage of avoiding the convergence to a locally optimal solution.
- If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained.

Ant System: An Example

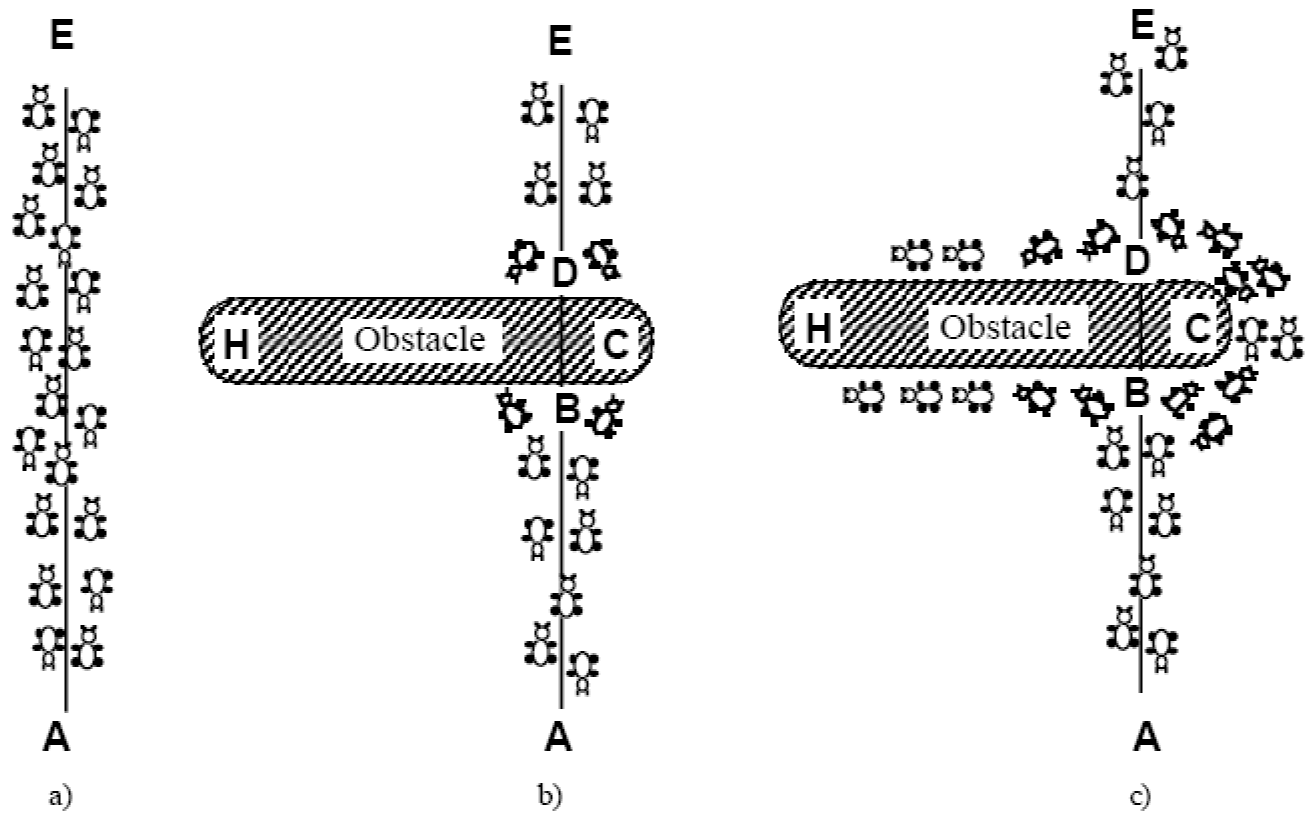


Fig. 1. An example with real ants.

a) Ants follow a path between points A and E.
b) An obstacle is interposed; ants can choose to go around it following one of the two different paths with equal probability.
c) On the shorter path more pheromone is laid down.

Ant System: An Example

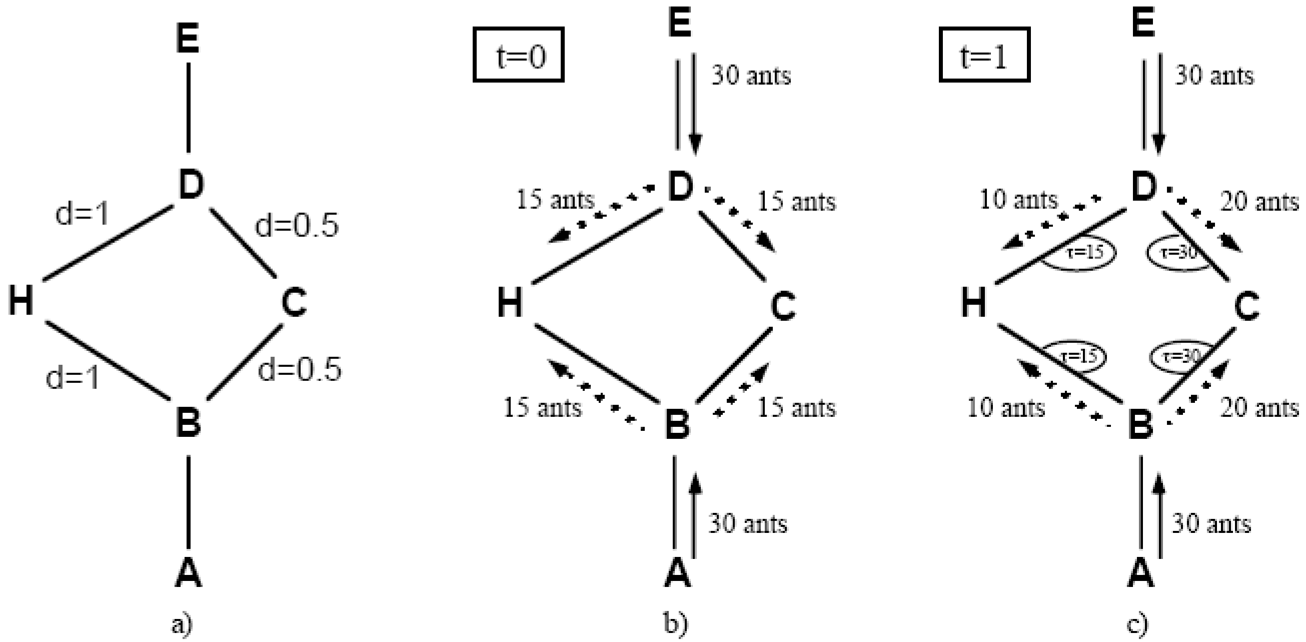


Fig. 2. An example with artificial ants.

a) The initial graph with distances.
b) At time t=0 there is no trail on the graph edges; therefore, ants choose whether to turn right or left with equal probability.
c) At time t=1 trail is stronger on shorter edges, which are therefore, in the average, preferred by ants.

Demonstration

NetLogo Web: Ants

ACO*

- The first algorithmic models of the foraging behavior of ants, as developed by Marco Dorigo in 1992 (PhD thesis)
- A general-purpose heuristic algorithm which can be used to solve different combinatorial optimization problems.
- Not interested in simulation of ant colonies, but in the use of artificial ant colonies as an optimization tool.
- Major differences with a real (natural) ant:
  - Artificial ants will have some memory
  - They will not be completely blind
  - They will live in an environment where time is discrete

* The Ant Systems: Optimization by a colony of cooperating agents by Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni

ACO Algorithm

- Let bi(t) (i=1, ..., n) be the number of ants in town i at time t and let m = sum [bi (t)] be the total number of ants.
- Each ant is a simple agent with the following characteristics:
  - it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge;
  - to force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed (this is controlled by a tabu list);
  - when it completes a tour, it lays a substance called *trail* on each edge (i,j) visited.

ACO Algorithm

- Let $\tau_{ij}(t)$ be the ***intensity of trail*** on edge (i,j) at time t.
- Each ant at time t chooses the next town, where it will be at time t+1.
- In n iterations, each ant completes a tour.
- At this point the trail intensity is updated according to the following formula

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij} \qquad (1)$$

where

ρ is a coefficient such that $(1 - \rho)$ represents the *evaporation* of trail between time t and t+n,

ACO Algorithm

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (2)$$

where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i,j) by the k-th ant between time t and t+n; it is given by

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if k - th ant uses edge (i, j) in its tour (between time t and t + n)} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where Q is a constant and L_k is the tour length of the k-th ant.

- In order to satisfy the constraint that an ant visits all the n different towns, a data structure, called ***tabu list***, is associated with each ant that saves the towns already visited upto time t and forbids the ant to visit them again before n iterations (a tour) have been completed.

ACO Algorithm

- ***Visibility/Desirability***, η_{ij}, is defined as the inverse of the length of the edge (i,j)
- The transition probability from town i to town j for the k^{th} ant is defined as

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{k \in allowed_k} \left[\tau_{ik}(t)\right]^{\alpha} \cdot \left[\eta_{ik}\right]^{\beta}} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

- α and β are parameters that control the relative importance of trail versus visibility.
- The transition probability is a trade-off between visibility and trail intensity.

Example

	A	B	C	D	E
A	0	5	2	9	3
B	5	0	8	1	9
C	2	8	0	10	2
D	9	1	10	0	4
E	3	9	2	4	0

- Initial Solutions
  - A C D B E 22
  - B A C D E 21
  - E D B C A 15
  - E B C A D 28
  - A D C E B 30

- Given
  - $\rho = 0.6$,
  - $Q = 1$
  - $\alpha = 0.8$
  - $\beta = 0.8$
  - $\tau_{ij} = 0$

Computing τ_{ij}

	A	B	C	D	E
A	0	(1/21)	(1/22+1/21+1/15+1/28)	(1/28+1/30)	0
B		0	(1/15)	(1/22)	(1/22+1/28+1/30)
C			0	(1/22+1/21+1/30)	(1/30)
D				0	(1/21+1/15)
E					0

- Initial Solutions
  - A C D B E 22
  - B A C D E 21
  - E D B C A 15
  - E B C A D 28
  - A D C E B 30

Computing τ_{ij} and η_{ij}

τ_{ij}

	A	B	C	D	E
A	0.000	0.048	0.195	0.069	0.000
B	0.048	0.000	0.067	0.045	0.115
C	0.195	0.067	0.000	0.126	0.033
D	0.069	0.045	0.126	0.000	0.114
E	0.000	0.115	0.033	0.114	0.000

η_{ij}

inverse of the length of the edge (i,j)

	A	B	C	D	E
A	0.00	0.20	0.50	0.11	0.33
B	0.2	0.00	0.13	1.00	0.11
C	0.5	0.13	0.00	0.10	0.50
D	0.11	1.00	0.10	0.00	0.25
E	0.33	0.11	0.50	0.25	0.00

Computing Transition Probabilities

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

- Numerators only
  - $p_{AB}^{1} = (0.048)^{0.8} (0.20)^{0.8} = 0.024$
  - $p_{AC}^{1} = (0.195)^{0.8} (0.5)^{0.8} = 0.156$
  - $P_{AD}^{1} = (0.069)^{0.8} (0.11)^{0.8} = 0.020$
  - $P_{AE}^{1} = 0.00$
- After Normalization (dividing by the denominator)
  - $p_{AB}^{1} = 0.121$
  - $p_{AC}^{1} = 0.778$
  - $p_{AD}^{1} = 0.101$
- So Ant 1, starting from node A, would decide about the next node based on these probabilities.
- The process is repeated for the other nodes in the sequence as well.
- All the ants follow this process.

Observations

- From the studies of the simple ACO algorithm, the importance of the exploration–exploitation trade-off becomes evident.
- Care should be taken to employ mechanisms to ensure that ants do not exploit pheromone concentrations such that the algorithm prematurely stagnates on sub-optimal solutions, but that ants are forced to explore alternative paths.
- ACO has an advantage over evolutionary algorithms from similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

Refinements in SACO

- Maximum and minimum limits on the pheromone levels are imposed to avoid stagnation.
- Rank-based elitist strategy is adopted in an attempt to prevent the algorithm from being trapped in a local minimum. In this strategy, ‘k’ best ranked ants are used to update the pheromone levels and the **amount of pheromone deposited by each ant decreases with its rank**. Furthermore, at each iteration, the global-best ant is allowed to deposit the largest amount of pheromone.

Refinements in SACO

- The ACS pseudo-random-proportional state transition rule provides a direct way to balance between exploration of new states and exploitation of a priori and accumulated knowledge. **The best state is chosen with probability q0 (that is a parameter $0 \le q0 \le 1$ usually fixed to 0.9) and with probability (1-q0) the next state is chosen randomly with a probability distribution based on ηij and τ ij** weighted by α (usually equal to 1) and β (usually equal to 2) .

ACO Simulation

- built by Osama Yousuf in CI (Spring 2019)

  Ant Colony Optimization - Simulation - Processing.py - YouTube

- **Ants Colony Simulation AI game experiment**

Combinatorial Optimization

Ant Colony **Optimization** (ACO) is a meta-heuristic algorithm that has been successfully applied to various Combinatorial Optimization Problems (COPs). For Example,

- Assignment problem
- Closure problem
- Constraint satisfaction problem
- Cutting stock problem
- Integer programming
- Knapsack problem
- Minimum spanning tree
- Nurse scheduling problem
- Traveling salesman problem
- Vehicle rescheduling problem
- Vehicle routing problem
- Weapon target assignment problem

Applications
Vehicle Routing Problem

VRPs are problems where a set of vehicles stationed at a depot has to serve a set of customers before returning to the depot, and the objective is to minimize:

– the number of vehicles used and

– the total distance traveled by the vehicles

Capacity constraints are imposed on vehicle trips, as well as possibly a number of other constraints deriving from real-world applications, such as time windows, rear loading, vehicle objections, maximum tour length, etc.

Multiple Ant Colony System (MACS)

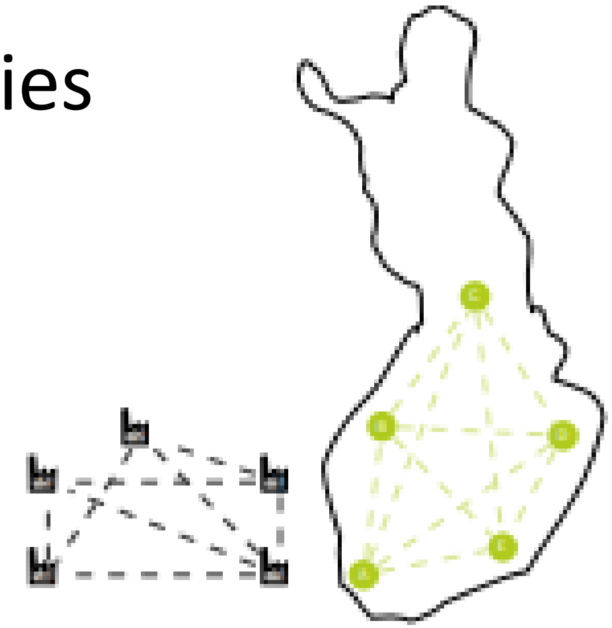
- Multiple Objective functions
  - Minimum number of vehicles
  - Minimum travelling time
- MACS-VRPTW employs two ant colonies; one colony (called ACS-VEI) to diminish the number of vehicles, and one colony (called ACS-TIME) to optimize the total travel time, and to optimize the solutions produced by ACSVEI.
- The two colonies are independent, possessing distinct pheromone trails, but they collaborate via a shared variable into which the currently best solution is stored.

Quadratic Assignment Problems

- There are a set of *n* facilities and a set of *n* locations. For each pair of locations, a *distance* is specified and for each pair of facilities a *weight* or *flow* is specified (e.g., the amount of supplies transported between the two facilities). The problem is to assign all facilities to different locations with the goal of minimizing the sum of the distances multiplied by the corresponding flows.

Quadratic Assignment Problems

- Optimal assignment of factories to the cities marked in green and Distances between the cities and flows between the factories shown below :

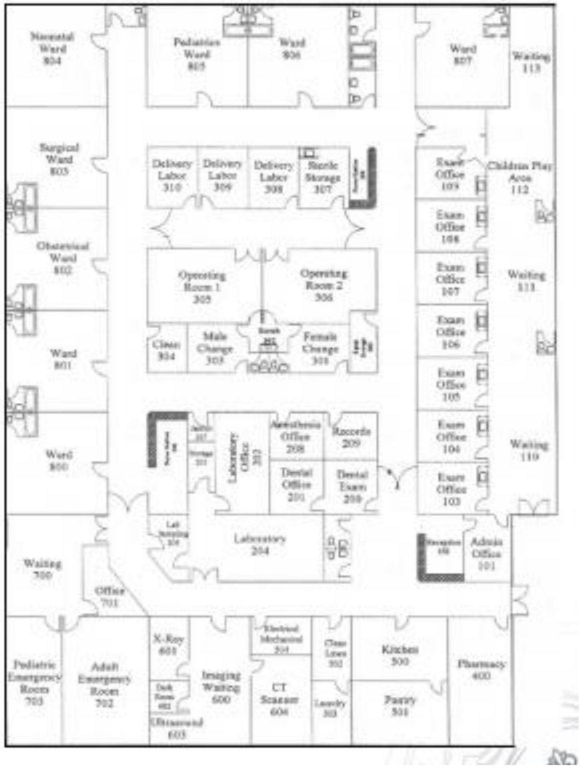


$$A = \begin{bmatrix} 0 & 3 & 6 & 4 & 2 \\ 3 & 0 & 2 & 3 & 3 \\ 6 & 2 & 0 & 3 & 4 \\ 4 & 3 & 3 & 0 & 1 \\ 2 & 3 & 4 & 1 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 10 & 15 & 0 & 7 \\ 10 & 0 & 5 & 6 & 0 \\ 15 & 5 & 0 & 4 & 2 \\ 0 & 6 & 4 & 0 & 5 \\ 7 & 0 & 2 & 5 & 0 \end{bmatrix}$$

http://web.abo.fi/fak/tkf/at/ose/doc/Pres_15112013/Axel%20Nyberg.pdf

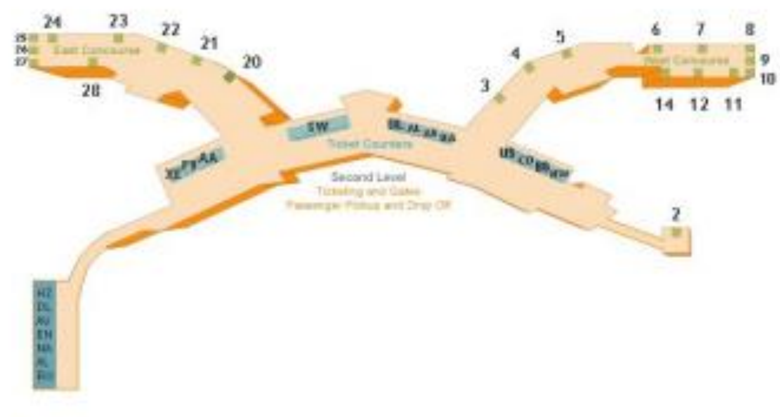
Hospital Layout

- Hospital Layout - German university hospital, Klinikum Regensburg, built 1972 I Optimality proven in the year 2000 I [Krarup and Pruzan(1978)]



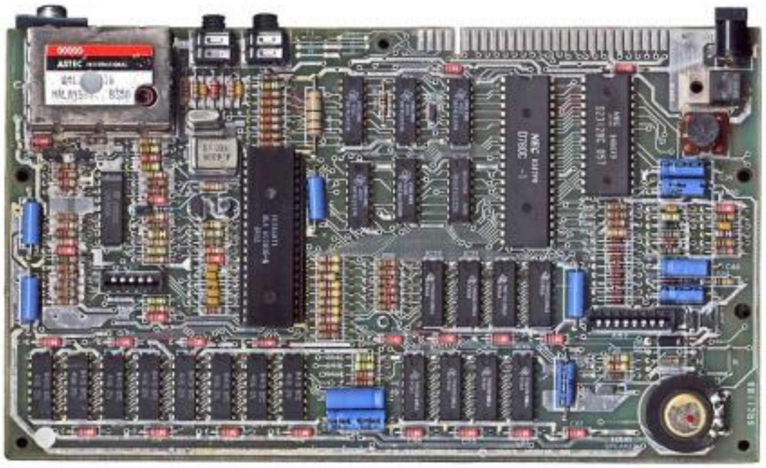
Airport Gate Assignment

- Airport gate assignment I Minimize total passenger movement I Minimize total baggage movement I [Haghani and Chen(1998)]



Wiring Problem

- Steinberg wiring problem I [Steinberg(1961)]
- Component placing on circuit boards [Rabak and Sichman(2003)]
- Minimizing the number of transistors needed on integrated circuits Burkard et al.(1993)



Heuristic vs Meta-heuristic

- A heuristic is a problem-specific technique that uses some information about the problem to find promising solutions. For example, a greedy algorithm is a heuristic that always chooses the best local option at each step.
- A meta heuristic is a problem-independent and generic technique that can be applied to a variety of problems. It usually combines or modifies existing heuristics to explore the solution space more efficiently.
- **The heuristics are the guidelines, metaheuristics is the framework that uses those.**

Thanks