EE496 : COMPUTATIONAL INTELLIGENCE
EA05 : ANT COLONIES

UGUR HALICI
METU: Department of Electrical and Electronics Engineering (EEE)
METU-Hacettepe U: Neuroscience and Neurotechnology (NSNT)

Ant Colony Optimization

- since food has to be fetched from its source and carried to the nest, ants form transportation roads
- to do this, they label all their routes with scents (a distinctive smell) for other ants may then trace their routes
- pheromones are this kind of chemical substances produced by insects
- this way, routes to food sources are minimized

Ant Colony Optimization

Ant Colony Optimization [Dorigo and Stützle, 2004]

motivation: some ant species are able to find shortest route to food sources by placing and tracing pheromones (scents)

- intuitively: short routes are labeled with more pheromone during the same time
- routes are randomly chosen according to the current pheromone distribution: the more pheromone there is, the more probable is it for ants to choose this way
- the amount of pheromone might vary according to the quality and amount of food found

main principle: stigmergy (traces left in the environment)

- ants are communicating implicitly by placing pheromones
- stigmergy (indirect communication by changing the environmental circumstances) allows for globally adapted behaviour due to locally found information

Double-Bridge Experiment [Goss et al., 1989]

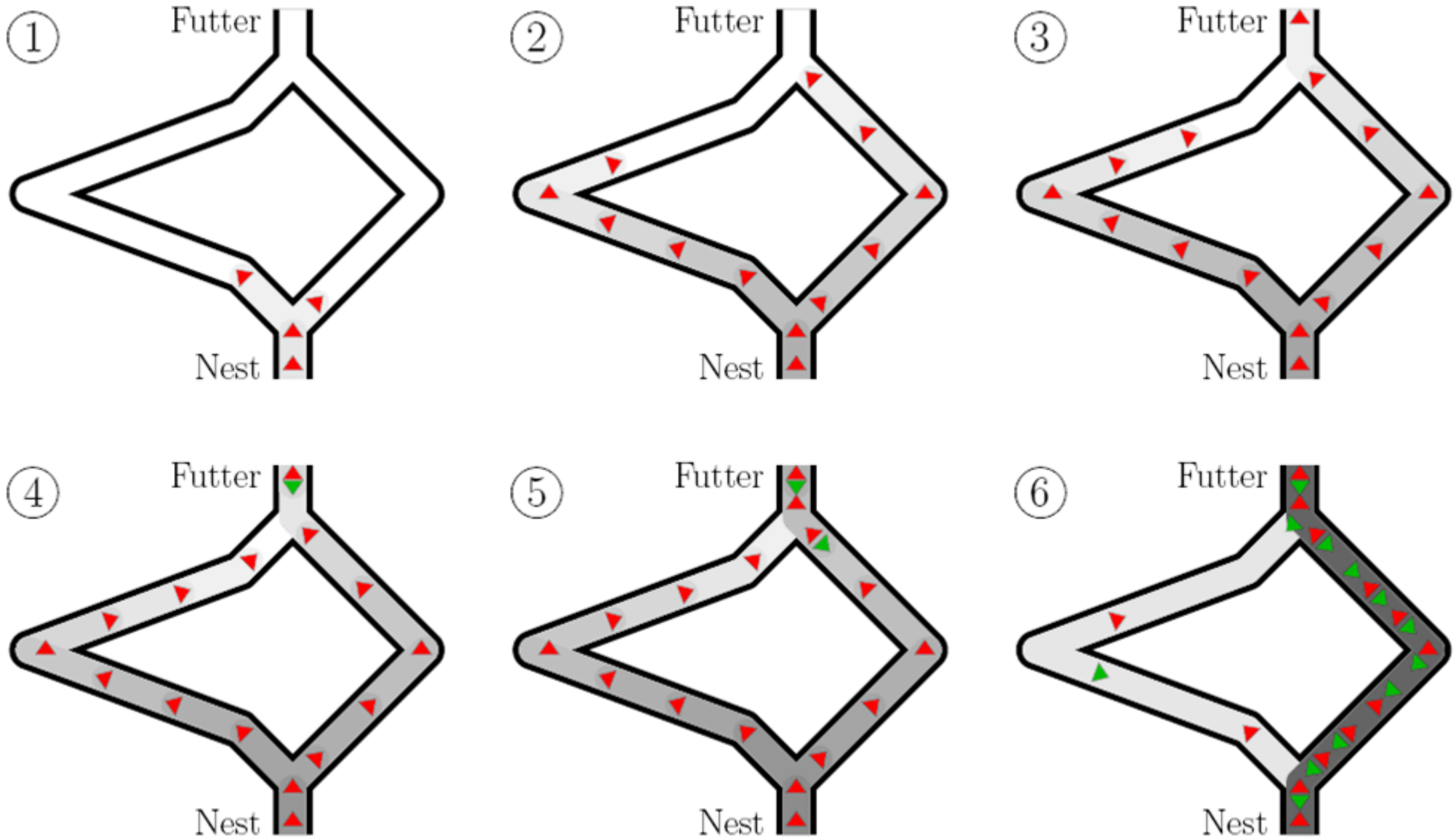
- ant nest and food source are connected by 2 bridges that differ in length
- experiment has been run with Argentinian Ants Iridomyrmex Humilis: these ants are almost blind (as most other ants, too) so they can't “see“ which bridge is shorter
- in most runs: after just several minutes most ants were using the shorter bridge

explanation

- ants travelling the shorter bridge are reaching the food earlier so in the first place the end of the shorter bridge gets more pheromon
- when returning to the nest, choosing the shorter bridge again is more probable for it is labeled with more pheromone now, thus increasing the difference in pheromone even more

Double-Bridge Experiment

- futter = food



Double-Bridge Experiment

notes:

- shorter route is found because of both bridges being available in the very beginning, and not differing in their amount of pheromone
- end of the shorter bridge is reached earlier

  ⇒ different amount of pheromone on both bridges

  ⇒ self-intensifying process

questions:

- What if a new even shorter route is added later by changing the environment?
- Will the ants change for the new route?

answer: No! [Goss et al., 1989]

- once a solution route has been established, the ants will stick to it
- proof: by a second bridge experiment: initializing the experiment with only one (longer) path), later adding a second (shorter) one
- most ants go on using the longer path, only few ants change.

Natural and Artificial Ants

reduce the problem to a search for the best path within a weighted graph

- **problem: self-intensifying cycles** (being visited by an ant a cycle becomes even more probable to be again visited by an ant)
- **solution:** labelling routes only after the ant has completed it's whole tour (thus cycles may be removed from the path)
- **problem: early conver**gence to a candidate solution found in the very beginning
- **solution:** pheromone **evaporation** (not of importance in nature)

useful extensions/improvements

- amount of pheromone dependent on the quality of the solution
- considering heuristics when choosing graph edges (e.g. their weights)

Ant Colony Optimization

- **preconditions:** combinatorial optimization problem with a constructive method for creating a solution candidate
- **procedure:** solutions are constructed according to a sequence of random choices, where every choice extends a partial solution
- sequence of choices = path in a decision graph (or construction graph)
- ants ought to explore the paths through a decision graph and find the best (shortest, cheapest) one
- ants label the graph edges with pheromone ⇒ other ants will be guided towards promising solutions
- pheromone "evaporates" after every iteration so once placed it won't affect the system for too long ("forgetting" outdated information)

Application to the TSP

- represent the problem by n x n matrix $D = (d_{ij})_{1 \le i,j \le n}$
- n cities with distances d_{ij} between city i and j
- note: D may be asymmetrical, but $\forall\, i \in \{1, . . . , n\} : d_{ii} = 0$
- pheromone information as n xn matrix $\Phi = (\phi_{ij})1 \le i,j \le n$
- pheromone value ϕ_{ij} $(i \neq j)$ indicates the desirability of visiting city j directly after visiting city i (ϕ_{ii} not used)
- there is no need in keeping Φ symmetrical
- initialize all ϕ_{ij} with the same small value (same amount of pheromone on all edges in the beginning)
- ants run Hamilton tour by labelling the edges of the Hamilton tour with pheromone (with the added pheromone value corresponding to the quality of the found solution)

Constructing a solution

- every ant possesses a “memory“ C where indices of not-yet visited cities are stored
- every visited city is removed from the set C
- there is no such memory in the nature!

1. ant is put randomly to a city where it begins its cycle
2. ant chooses not-yet visited city and goes there: in city i an ant chooses a (not-yet visited) city j with a probability of

$$p_{ij} = \frac{\phi_{ij}}{\sum_{k \in C} \phi_{ik}}.$$

3. repeat step 2 until every city has been visited

Updating the pheromone

1. **evaporation**

   all ϕ_{ij} are reduced by a fraction η (evaporation):

   $$\forall \quad i, j \in \{1, \ldots, n\} : \phi_{ij} = (1 - \eta) \cdot \phi_{ij}$$

2. **intensifying a constructed solution:**

   pheromone is put on all edges of the constructed solution corresponding to it's quality:

   $$\forall \pi \in \Pi_t : \phi_{\pi(i)\pi((i \bmod n)+1)} = \phi_{\pi(i)\pi((i \bmod n)+1)} + Q(\pi)$$

   Π_t is the amount used for the tour (permutation) constructed during step t, function of quality: e.g. inverse travelling length

   $$Q(\pi) = c \cdot \left(\sum_{i=i}^{n} d_{\pi(i)\pi((i \bmod n)+1)} \right)^{-1}$$

   ”The better the solution, the more pheromone is added.“

Travelling Salesman Problem

Algorithm 3 Ant colony optimization for TSP

initialize all elements ϕ_{ij}, $1 \leq i, j \leq n$ of the matrix, with small ϵ
do {
    for each Ant {   /* generate candidate solution */
        $C \leftarrow \{1, \ldots, n\}$ /* set of cities to be visited */
        $i \leftarrow$ draw a hometown randomly from C
        $C \leftarrow C \setminus \{i\}$ /* remove it from the set of not-yet visited cities */
        while $C \neq \emptyset$ {   /* while there are not-yet visited cities */
            $j \leftarrow$ draw the next city from C with probability p_{ij}
            $C \leftarrow C \setminus \{j\}$ /* remove it from the set of not-yet visited cities */
            $i \leftarrow j$ /* and move there */.
        }
    }
    update matrix of pheromones Φ according to the fitness of the solution
} **while** termination condition is not satisfied

Extensions and Alternatives

- **prefer nearby cities:** (analogical to next neighbor heuristics) move from city i to city j with probability

$$p_{ij} = \frac{\phi_{ij}^{\alpha}\tau_{ij}^{\beta}}{\sum_{k\in C}\phi_{ik}^{\alpha}\tau_{ik}^{\beta}}$$

  with C = set of indices of not-yet visited cities and $\tau_{ij} = d^{-1}_{ij}$

- **tend to choosing the best edge**: (greedy)

  with probability $p_{exploit}$ move from city i to city j_{best} with

  $j_{best} = \mathrm{argmax}_{j\in C}\, \phi_{ij}$ or $j_{best} = \mathrm{argmax}_{j\in C}\, \phi^{\alpha}_{ij}\, \tau^{\beta}_{ij}$

  and use p_{ij} with probability $1 - p_{exploit}$

- **intensify best known tour:** (elitism)

  label it with extra pheromone (e.g. the fraction of additional ants that pass it)

Extensions and Alternatives

ranking based updates

- place pheromone only on edges of last iteration's best solution (and possibly on the best overall solution, too)
- amount of pheromone depends on the rank of the solution

strict elite principles

- place pheromone only on the last iteration's best solution
- place pheromone only on the best solution found so far

Summary

Genetic and Evolutionary Algorithms

representation: (classical) $\{0, 1\}^L$ with fixed length L (also R^L and S_n, decoding)

mutation: bit flipping, uniformly distributed real-valued mutation, special operations for permutation

recombination: k-point- and uniform crossover, arithmetical crossover, order-based recombination

selection: parental selection, fitnessproportional or tournament selection

population: mid-sized populations

Summary

Local Search

representation: arbitrary

mutation: arbitrary

recombination: none

selection: improvements always, degradiation with a certain probability

population: one individual

features: early convergence is a central problem

Population-Based Incremental Learning

representation: $\{0, 1\}^L$

mutation: changing the population statistics

recombination: implicitly

selection: best child individual enters statistics

population: is replaced by population statistics

features: whenever individuals are needed, they are drawn from the statistics

Summary

Particle Swarm Optimization

representation: R^L

mutation: based on lethargy (inactivity) and neighbors

recombination: none

selection: based on the best (population/own memory)

population: small/mid-sized

features: synchronously searching the search space

Ant colony optimization

representation: several different

mutation: every ant generates one solution candidate

recombination: none

selection: quality determines influence on global pheromones

population: quantity of ants during one iteration

features: global amount of pheromones represents candidate solutionssimilar to statistics in PBIL