# Ant Colony Optimization (ACO)

Developed in 1991 by Dorigo (PhD dissertation) in collaboration with Colorni & Maniezzo

10/6/04 1

## Basis of all Ant-Based Algorithms

- Positive feedback
- Negative feedback
- Cooperation

10/6/04 2

## Positive Feedback

- To reinforce portions of good solutions that contribute to their goodness
- To reinforce good solutions directly
- Accomplished by *pheromone accumulation*

10/6/04 3

## Negative Feedback

- To avoid premature convergence (*stagnation*)
- Accomplished by *pheromone evaporation*

10/6/04 4

## Cooperation

- For simultaneous exploration of different solutions
- Accomplished by:
  - *multiple ants* exploring solution space
  - *pheromone trail* reflecting multiple perspectives on solution space



## Ant System for Traveling Salesman Problem (AS-TSP)

- During each iteration, each ant completes a tour
- During each tour, each ant maintains *tabu list* of cities already visited
- Each ant has access to
  - distance of current city to other cities
  - intensity of local pheromone trail
- Probability of next city depends on both



## Transition Rule

- Let $\eta_{ij} = 1/d_{ij}$ = "nearness" of city $j$ to current city $i$
- Let $\tau_{ij}$ = strength of trail from $i$ to $j$
- Let $J_i^k$ = list of cities ant $k$ still has to visit after city $i$ in current tour
- Then transition probability for ant $k$ going from $i$ to $j \in J_i^k$ in tour $t$ is:

$$p_{ij}^k = \frac{\left[\tau_{ij}(t)\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum_{l \in J_i^k} \left[\tau_{il}(t)\right]^\alpha \left[\eta_{il}\right]^\beta}$$



## Pheromone Deposition

- Let $T^k(t)$ be tour $t$ of ant $k$
- Let $L^k(t)$ be the length of this tour
- After completion of a tour, each ant $k$ contributes:

$$\Delta\tau_{ij}^k = \begin{cases} Q / L^k(t) & \text{if } (i,j) \in T^k(t) \\ 0 & \text{if } (i,j) \notin T^k(t) \end{cases}$$

## Pheromone Decay

- Define total pheromone deposition for tour $t$:

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

- Let ρ be decay coefficient
- Define trail intensity for next round of tours:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

10/6/04 9

## Number of Ants is Critical

- Too many:
  - suboptimal trails quickly reinforced
  - ∴ early convergence to suboptimal solution
- Too few:
  - don't get cooperation before pheromone decays
- Good tradeoff:
  number of ants = number of cities
  ($m = n$)

10/6/04 10

## Improvement: "Elitist" Ants

- Add a few ($e \approx 5$) "elitist" ants to population
- Let $T^+$ be best tour so far
- Let $L^+$ be its length
- Each "elitist" ant reinforces edges in $T^+$ by $Q/L^+$
- Add $e$ more "elitist" ants
- This applies accelerating positive feedback to best tour

10/6/04 11

## Time Complexity

- Let $t$ be number of tours
- Time is $\mathcal{O}(tn^2m)$
- If $m = n$ then $\mathcal{O}(tn^3)$
  - that is, cubic in number of cities

10/6/04 12

## Evaluation

- Both “very interesting and disappointing”
- For 30-cities:
  - beat genetic algorithm
  - matched or beat tabu search & simulated annealing
- For 50 & 75 cities and 3000 iterations
  - did not achieve optimum
  - but quickly found good solutions
- I.e., does not scale up well
- Like all general-purpose algorithms, it is out-performed by special purpose algorithms

10/6/04 13

## Improving Network Routing

1. Nodes periodically send *forward ants* to some recently recorded destinations
2. Collect information on way
3. Die if reach already visited node
4. When reaches destination, estimates time and turns into *backward ant*
5. Returns by same route, updating routing tables

10/6/04 14

## Some Applications of ACO

- Routing in telephone networks
- Vehicle routing
- Job-shop scheduling
- Constructing evolutionary trees from nucleotide sequences
- Various classic NP-hard problems
  - shortest common supersequence, graph coloring, quadratic assignment, …

10/6/04 15

## Improvements as Optimizer

- Can be improved in many ways
- E.g., combine local search with ant-based methods
- As method of stochastic combinatorial optimization, performance is promising, comparable with best heuristic methods
- Much ongoing research in ACO
- But optimization is not a principal topic of this course

10/6/04 16

## The Nonconvergence Issue

- AS often does not converge to single solution
- Population maintains high diversity
- A bug or a feature?
- Potential advantages of nonconvergence:
  - avoids getting trapped in local optima
  - promising for dynamic applications
- Flexibility & robustness are more important than optimality in natural computation

10/6/04 17

## Natural Computation

*Natural computation* is computation that occurs in nature or is inspired by computation occurring in nature

10/6/04 18

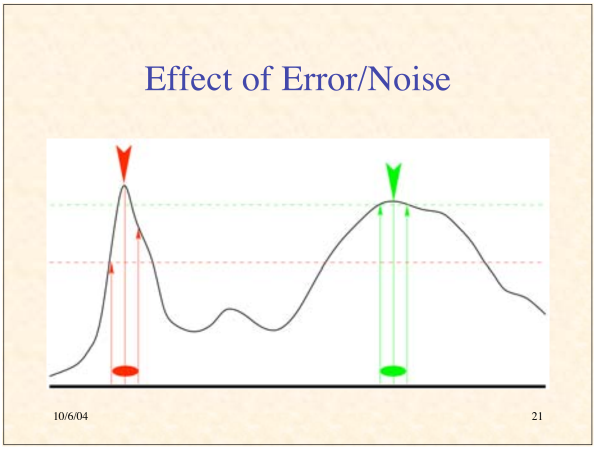
## Optimization in Natural Computation

- Good, but suboptimal solutions may be preferable to optima if:
  - suboptima can be obtained more quickly
  - suboptima can be adapted more quickly
  - suboptima are more robust
  - an ill-defined suboptimum may be better than a sharp optimum
- "The best is often the enemy of the good"

10/6/04 19

## Robust Optima

10/6/04 20


Effect of Error/Noise
10/6/04
21