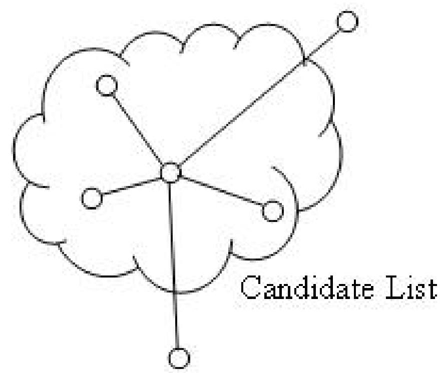
“Ant Colony Optimization: The Traveling Salesman Problem”

Section 2.3 from *Swarm Intelligence: From Natural to Artificial Systems* by Bonabeau, Dorigo, and Theraulaz

Ant Colony System (ACS)

- Improvements on Ant System made by Dorigo and Gambardella
- Four modifications to AS:
  - Candidate list to restrict choices of next city
  - Different transition rule
  - Different pheromone trail update rule
  - Local updates of pheromone trail

Candidate List



Transition Rule

- Uses a tunable parameter q_0 to adjust the probability that an ant will explore or reinforce existing paths
- If $q > q_0$, ACS transition rule is identical to AS
- If $q \leq q_0$, ACS exploits knowledge available about the problem. Use best edge rather than probabilistically choosing among edges.
- Gradually increasing q_0 would allow the algorithm to favor exploration early and existing paths afterward. This is a component of simulated annealing.

Initialization

Algorithm 2.2 High-level description of ACS-TSP

/* **Initialization** */
For every edge (i, j) **do**
  $\tau_{ij}(0) = \tau_0$
End For
For $k = 1$ to m **do**
  Place ant k on a randomly chosen city
End For
Let T^+ be the shortest tour found from beginning and L^+ its length

Transition Rule

/* Main loop */
For $t = 1$ to $t_{\max}$ **do**
  For $k = 1$ to m **do**
    Build tour $T^k(t)$ by applying $n - 1$ times the following steps:
    If exists at least one city $j \in$ candidate list **then**
      Choose the next city j, $j \in J_i^k$, among the cl cities in the candidate list as follows

$$j = \begin{cases} \arg\max_{u \in J_i^k} \{[\tau_{iu}(t)] \cdot [\eta_{iu}]^\beta\} & \text{if } q \leq q_0; \\ J & \text{if } q > q_0, \end{cases}$$

      where $J \in J_i^k$ is chosen according to the probability:

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

      and where i is the current city
    Else
      choose the closest $j \in J_i^k$

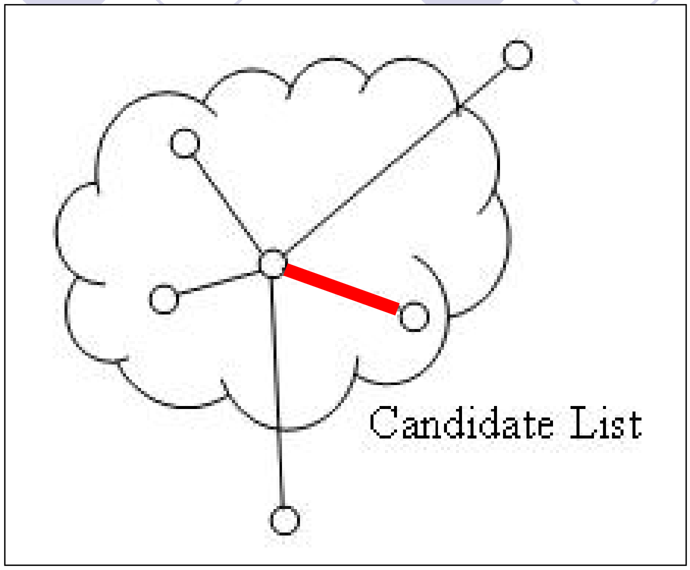
Local Pheromone Trail Updates

- Each time an ant moves from one city to another, the pheromone concentration on that edge is reduced
- This makes visited edges less attractive and encourages exploration
- τ_0: $(n * L_{nn})^{-1}$
  - n: number of cities
  - L_{nn}: Length of tour as produced by nearest neighbor heurstic

Pheromone Update

End If
After each transition ant k applies the local update rule:
$\tau_{ij}(t) \leftarrow (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\tau_0$
End For
For $k = 1$ to m **do**
Compute the length $L^k(t)$ of the tour $T^k(t)$ produced by ant k
End For
If an improved tour is found **then**
update T^+ and L^+
End If
For every edge $(i,j) \in T^+$ **do**
Update pheromone trails by applying the rule:
$\tau_{ij}(t) \leftarrow (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}(t)$ where $\Delta\tau_{ij}(t) = 1/L^+$
End For
For every edge (i,j) **do**
$\tau_{ij}(t+1) = \tau_{ij}(t)$
End For

Local Pheromone Update



Pheromone Trail Update Rules

- Only the ant with the best tour lays pheromones
- Encourages ants to search paths near the best found so far

Example Solution

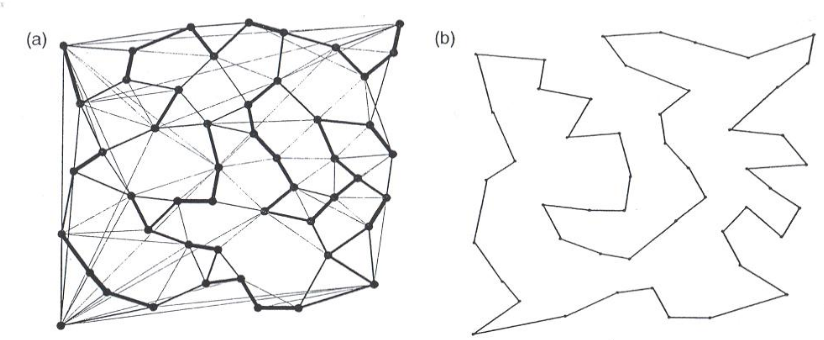


FIGURE 2.15 (a) An example of a trail configuration found by ACS in a 50-city problem (Eil50). Line thickness reflects pheromone concentration. (b) Best solution found by ants.

ACS Conclusions

TABLE 2.2 Comparison of ACS-TSP with other algorithms on randomly generated prob lems. Results for EN are from Durbin and Willshaw [117] and results for SA and SOM ar from Potvin [262]. For ACS-TSP, averages are over 25 runs. Best results are in boldface From Dorigo and Gambardella [111].

	ACS-TSP	SA	EN	SOM
City set 1 (50-city problem)	**5.88**	**5.88**	5.98	6.06
City set 2 (50-city problem)	6.05	**6.01**	6.03	6.25
City set 3 (50-city problem)	**5.58**	5.65	5.70	5.83
City set 4 (50-city problem)	**5.74**	5.81	5.86	5.87
City set 5 (50-city problem)	**6.18**	6.33	6.49	6.70

TABLE 2.3 Comparison of ACS-TSP with GA, EP, and SA on four test problems (available at TSPLIB: http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html). Results for ACS-TSP are from Dorigo and Gambardella [111]. ACS-TSP is run for 1250 iterations using 20 ants, which amounts to approximately the same number of tours searched by the other heuristics; averages are over 15 runs. Results for GA are from Bersini et al. [20] for the KroA100 problem, and from Whitley et al. [327] for the Oliver30, Eil50 and Eil75 problem; results for EP are from Fogel [125]; and results for SA from Lin et al. [222]. For each algorithm results are given in two columns: the first one gives the best integer tour length (that is, the tour length obtained when distances among cities are given as integer numbers), and, in parenthesis, the best real tour length (that is, the tour length obtained when distances among cities are given as real numbers); the second one gives the number of tours that were generated before the best integer tour length was discovered. Best results are in boldface. (N/A: Not available.)

	ACS-TSP best	ACS-TSP # iter.	GA best	GA # iter.	EP best	EP # iter.	SA best	SA # iter.
Eil50 (50-city problem)	**425** (427.96)	**1830**	428 (N/A)	25000	426 **(427.86)**	100000	443 (N/A)	68512
Eil75 (75-city problem)	**535** **(542.37)**	**3480**	545 (N/A)	80000	542 (549.18)	325000	580 (N/A)	173250
KroA100 (100-city problem)	**21282** **(21285.44)**	4820	21761 (N/A)	103000	N/A (N/A)	N/A	N/A (N/A)	N/A

- ACS performs comparably or better than other TSP algorithms

Further Enhancements

- Added 3-opt procedure to locally minimize tour lengths between ACS iterations
  - Performed comparably with STSP algorithm that won the First International Contest on Evolutionary Optimization
- Other unimplemented improvements:
  - Allow the best r ants to update the pheromone trail, instead of a single ant
  - Remove pheromone from edges of worst tours
  - More powerful local search