Lecture 37
Genetic and Random Search Algorithms (2)

Outline

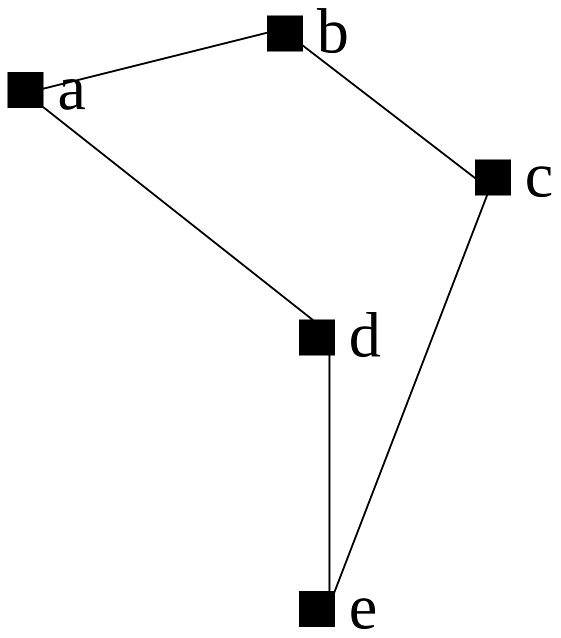
- Discrete optimization problem
- Random search method
- Simulated Annealing (SA) method

Discrete Optimization Problem

- An optimization problem of which the total number of possible solutions are countable.
- Often the number of solutions are finite but increases exponentially with respect to problem size. These lead to the class of NP-hard or NP-complete problem.
- Time to search the global optimal solution increases exponentially.
- Practical approaches:
  - Heuristic solutions
  - Random search solutions

Example: Traveling Salesperson Problem

- A traveling sales person needs to make a travel plan to visit N cities.
- Every city can only be visited once (1 arrival and 1 departure).
- The total distance of the tour needs to be minimized.
- Example: a-b-c-e-d-a is one possible tour.
- Total # of tours with potentially different distances = (N–1)!/2 where N = # of cities.



TSP Problem Representation

- Index each city with an integer in {1, 2, …, N}.
- Since the trips of 1-2-3, 2-3-1, or 3-1-2 are the same, hence one may designate city 1 as the fixed starting city without affecting the optimality of the solution.
- If one exhaustively search all permutations of the remaining N–1 cities, this leads to (N –1)! possible solutions.
- Also, reverse tour has the same distance. Eg. 1-3-2-4 and 1-4-2-3 have the same distance.
- Hence total number of distinct solution is (N–1)!/2
- $N \leq 3$, only 1 solution.

Greedy Algorithm

- Define the "distance" between two solutions x and y as

  d(x,y) = # of different cities in the tours.

  $$2 \le d(x,y) \le N-1$$

- Given the current best solution x(n), find the trip lengths of all neighboring solution x' in

  B={x'| d(x(n), x') = 2 (say)}.

- Choose x(n+1) to be a member in B such that the corresponding trip length is
  1) smaller than that of x(n), and
  2) smaller than that of all other x' in B.

  This is the steepest descent search direction.

- Repeat this process until no further improvement can be found. Then a local minimum is obtained.
- Proposed by Lin and Kernighan

Random Search Algorithm

- Randomly generate a permutation of {2, …, N}.
- Evaluate the trip length of the generated permutation.
- If the trip length is smaller than the current best solution, use the new solution to replace the current best solution.
- Otherwise, continue generate additional solutions until a termination criterion is met.

- Random search algorithm converges to the globally optimal solution exponentially as a function of repeated trials!
- To prove, let P (<<1) be the probability that the solution randomly generated in a trial is globally optimal. Since each trial is considered independent, the probability that global optimal solution has not been generated in M trials is $(1-P)^M$.

Simulated Annealing Algorithm

1. Generate a new tour based on current tour by interchanging up to k cities (distance = k).
2. Evaluate the trip length. If it is smaller than the current best solution, accept the new tour as current tour and update the current best solution.
3. Else, if the trip length is greater than the current best solution, then accept the new tour with a probability.
4. The probability to accept a new tour with higher trip length is varying according to a cooling schedule. Initially the temperature is high and the probability to accept such a step is high. As iterations progress, the probability to accept a step that increases the cost will be reduced.
5. Stop the algorithm after certain convergence criteria are met.

Simulation Example