SIMULATED ANNEALING

Fundamental Concept

- Motivation by an analogy to the statistical mechanics of annealing in solids. => to coerce a solid (i.e., in a poor, unordered state) into a low energy thermodynamic equilibrium (i.e., a highly ordered defect-free state) such as a crystal lattice.
- The material is annealed by heating the material to a temperature that permits many atomic rearrangements, then cooled *slowly* until the material freezes into a good crystal- Metropolis procedure
- Different from greedy algorithm, allows perturbation to move uphill in a controlled fashion to escape from local minima.

Design Metaphor

- Simulated annealing offers a physical analogy for the solution of optimization problems.
- Boltzmann distribution

$$\frac{N_{current\_state}}{N_{new\_state}} = \exp\left(\frac{E_{new\_state} - E_{current\_state}}{kT}\right)$$

  N_i: number of atoms with energy

  E_i, i = new_state or current_state

  k is the Boltzmann constant

  T is the absolute temperature

- In simulated annealing the left hand side is interpreted as a probability
- The probability of a uphill move of size ΔE at temperature T is $\Pr(accept) = e^{-\Delta E / kT}$
- If $\Delta E<0$, the new configuration is accepted and

  if $\Delta E>0$, probability of accepting the worse configuration is calculated based on Boltzmann distribution

- At higher temperatures, the probability of large uphill move in energy is large (permits an aggressive, essentially random search of the configuration space)
- At lower temperatures the probability is small (few uphill moves are allowed)
- By successfully lowering the temperature (cooling schedule) and running the algorithm, we can simulate the material coming into equilibrium at each newly reduced temperature.

Cooling Schedule

- A starting hot temperature and rule to determine when the temperature should be lowered and how much the temperature should be lowered and when annealing should be terminated.
- $T = \alpha T$, $\alpha < 1$
- If α is very small, the temperature reduces very fast and there is high possibility of being trapped in a local minimum
- If α is large the energy decreases very slowly
- Many schemes to reduce temperature
  - Fast Cauchy: $T_k = T_0 \frac{1}{k}$
  - Geometric: $T_k = T_0 \alpha k$
  - Boltzmann: $T_k = T_0 \frac{1}{\ln(k)}$

Where T_0 is the initial temperature, and T_k is the temperature after the k'th temperature decrement

Pseudo Codes

- M = number of moves (perturbations) to attempt
- T = current temperature
- For m = 1 to M

  Generate a random move

  Evaluate the change in energy ΔE

  IF ($\Delta E < 0$)

    Accept this move and update configuration /* downhill move

  ELSE

    Accept with probability, $P = e^{-\Delta E/T}$

    Update configuration if accepted

  ENDIF

  Update temperature $T = \alpha T$
- ENDFOR

Simulated Annealing for TSP

- Initial configuration: permutation =>1, 2, 3, 4, …, *N*

  temperature $T = 2$

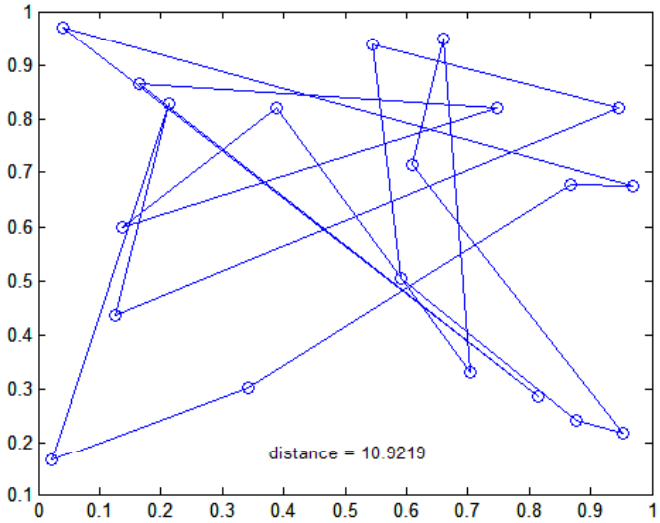
  cooling rate $\alpha = 0.99$

  energy = $d(1,2)+d(2,3)+\ldots+d(N,1)$

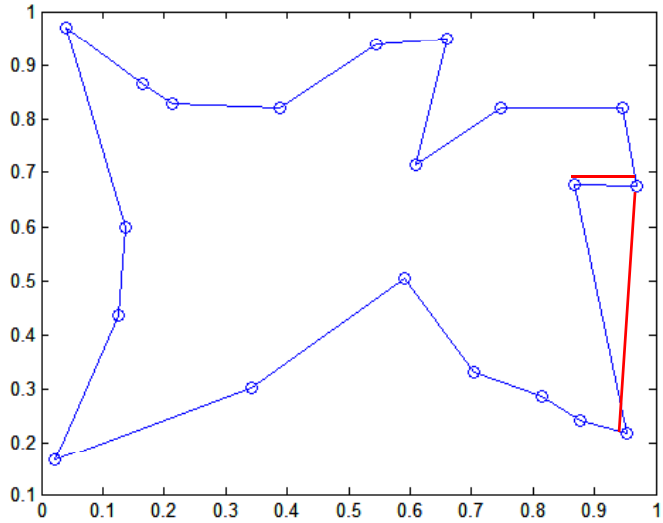
- Generate a new configuration from the current one at random
- Evaluate ΔE = current energy – previous energy
- If $\Delta E < 0$ accept the current configuration (downhill)

  Else accept configuration with probability $P = e^{-\Delta E / kT}$
- $T = \alpha T$
- Stopping criteria if energy < threshold or number of iterations is reached.

Case Study: 20-city problem

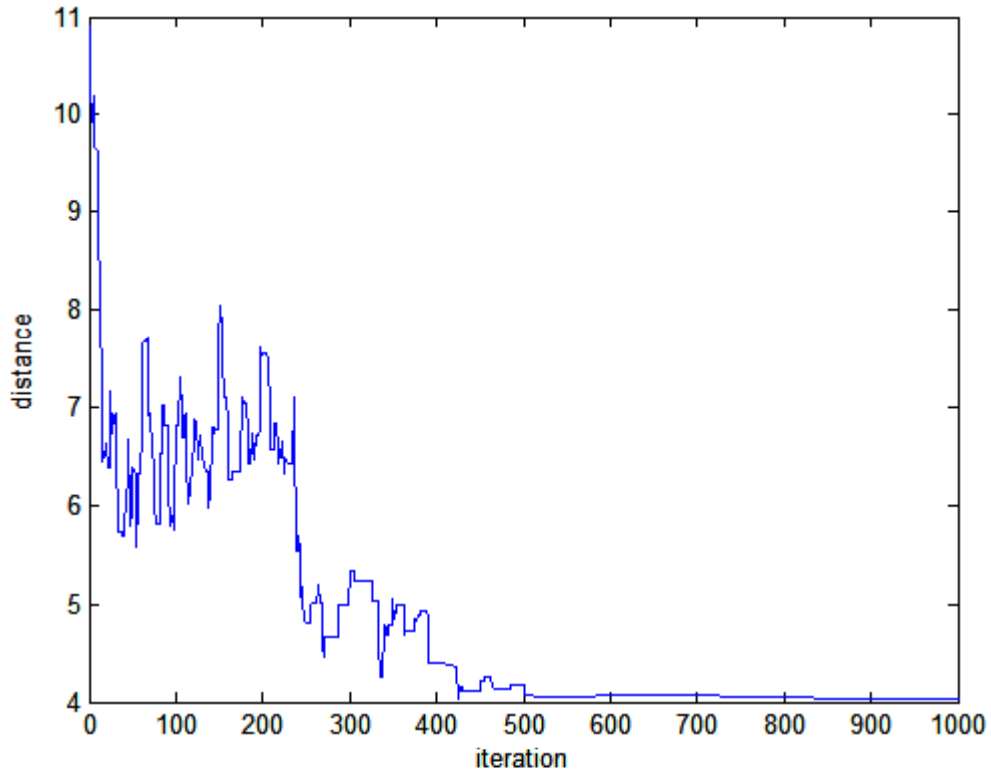


Initial random permutation



Best route found for 20-city problem
d=4.0185 (d=4.0192)

Initial temperature	2	2	2	2	2
Cooling rate	0.8	0.9	0.95	0.99	0.999
Converged iteration	87	155	298	851	5982
Minimal energy obtained	4.1056	4.0920	4.0427	4.0185	4.0185



T_0= 2
α=0.99
Iteration = 851
d = 4.0185

Case Study: 101-city problem (ali101)

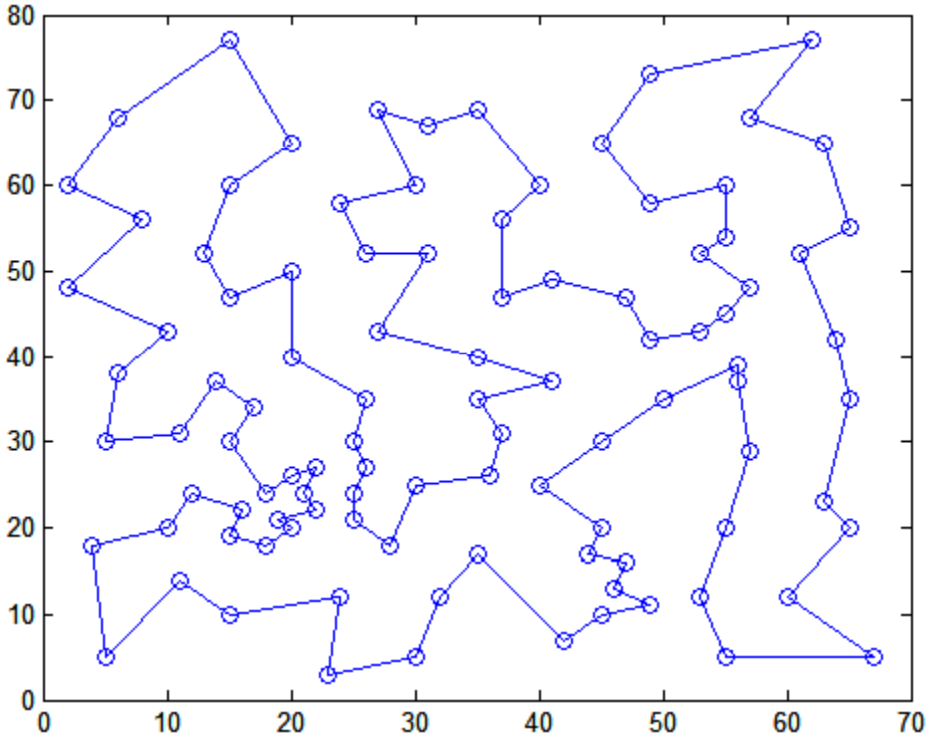
T_0=200

α= 0.999

Iterations = 100,000

d = 661.25

Global minima = 629

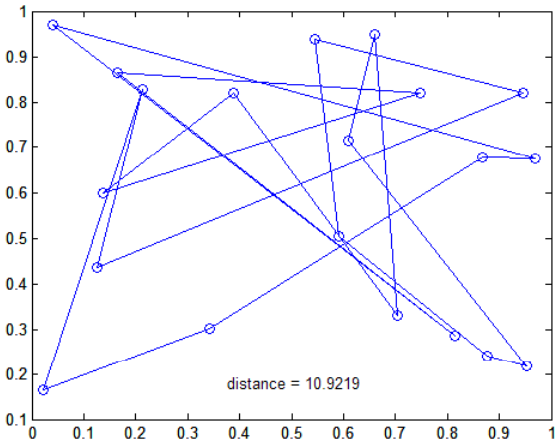


Research Issues

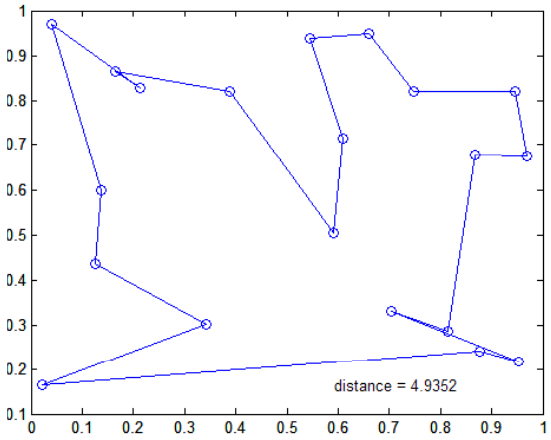
- Initial Temperature
- Initial Configuration
- Next Move (neighborhood size)
  - is it possible to design a truly local search for combinatorial optimization problem (as oppose to numerical optimization problem)?
- Cooling Schedule (cooling rate)
- Stopping Criteria
- Acceptance Probability

Effect on Initialization

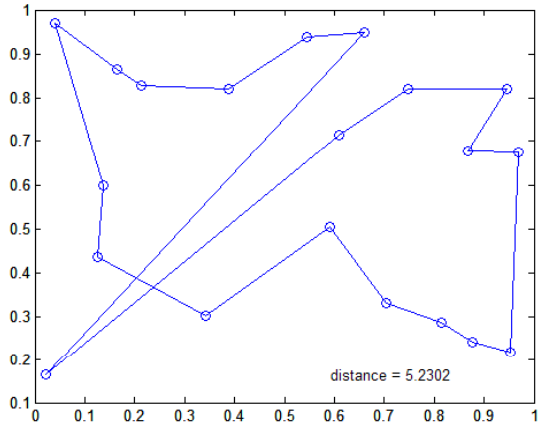
Random sequence



Hilbert space filling curve



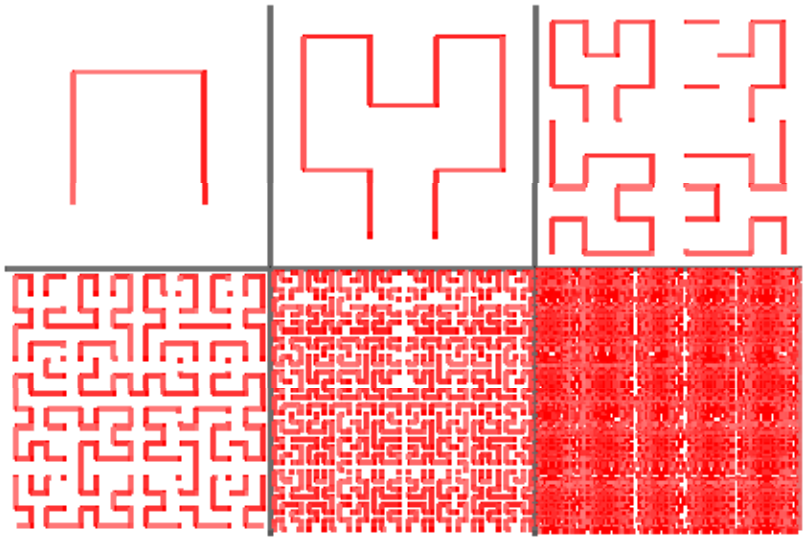
Fastest closest path

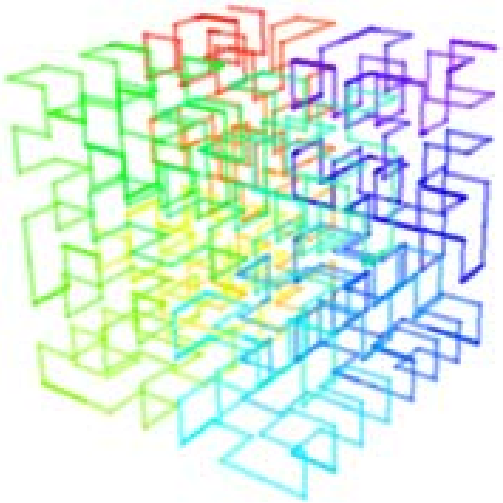


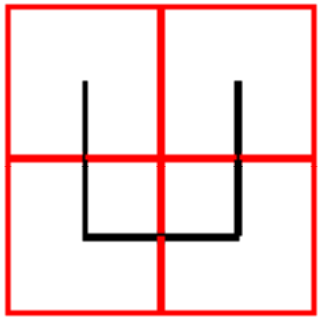
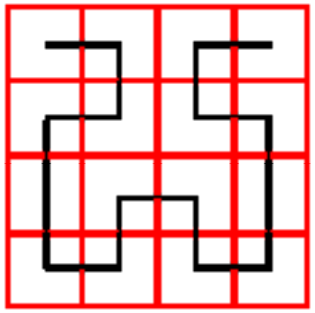
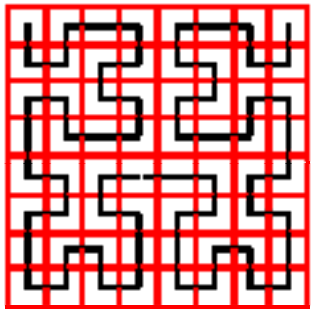
Experimental results indicate that the initialization method does not affect much the probability of the system to converge to the global optimal point.

Space Filling Curve

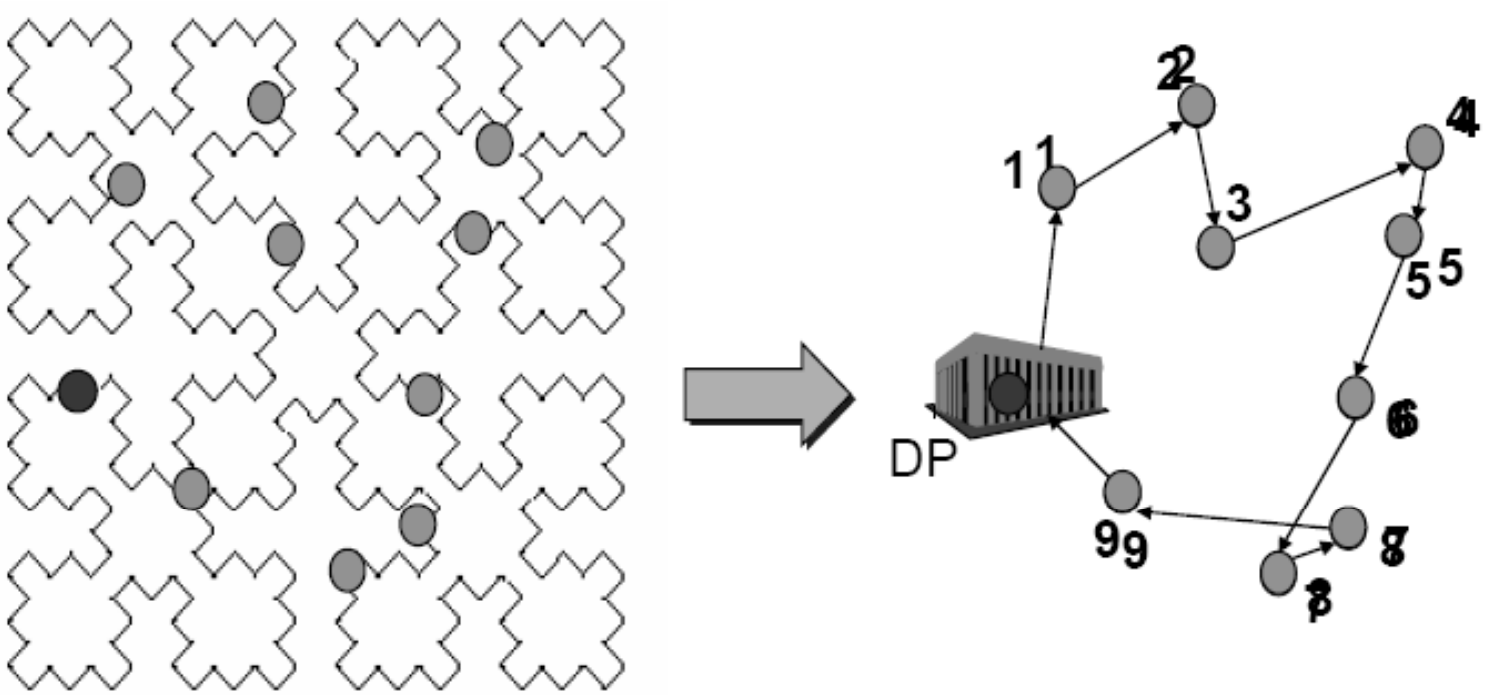
- A ***space filling curve*** is a continuous mapping from a lower dimensional space into a higher one. A spacefilling curve is formed by repeatedly copying and shrinking a simple pattern.
- A ***Hilbert curve*** (also known as a Hilbert space-filling curve) is a continuous fractal space-filling curve first described by the German mathematician David Hilbert in 1891.





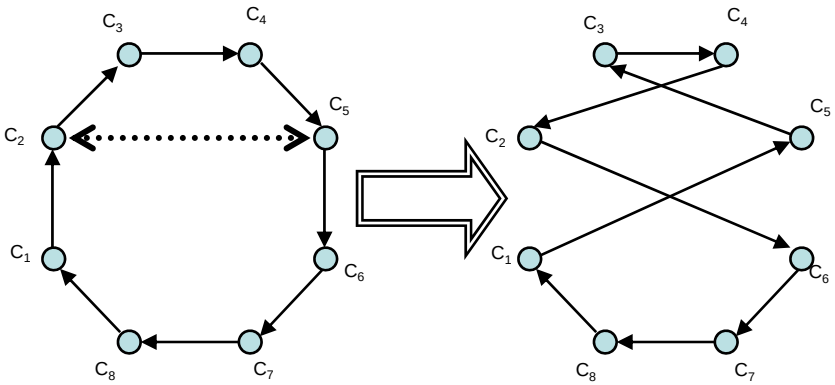




- Divide the square into four smaller squares $\mathbf{I_{00}}$, $\mathbf{I_{01}}$, $\mathbf{I_{10}}$, and $\mathbf{I_{11}}$
- Define $\mathbf{f_0}$ so that it maps [0,1/4] into $\mathbf{I_{00}}$, [1/4,1/2] into $\mathbf{I_{01}}$ and so on…


DP
1
2
3
4
5
6
9
8

Effect on Next Move Modification

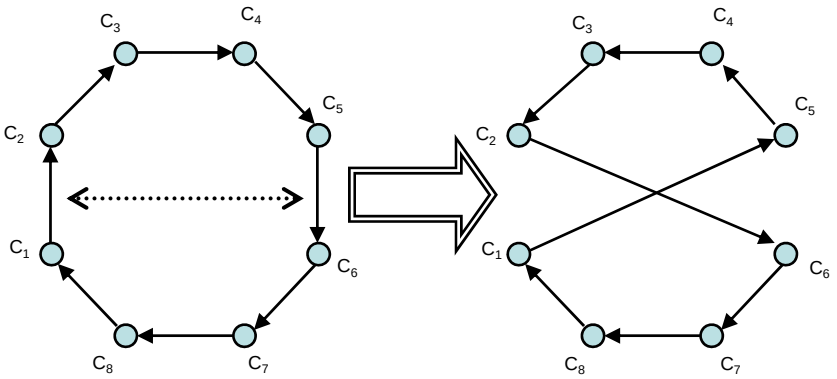
Random switch of two vertices



1 2 3 4 5 6 7 8

1 5 3 4 2 6 7 8

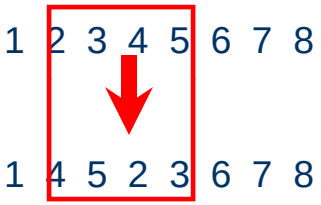
Random switch of two edges

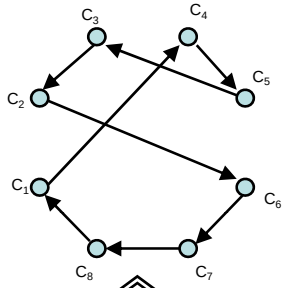


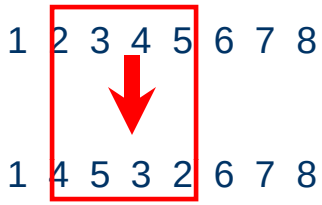
1 2 3 4 5 6 7 8

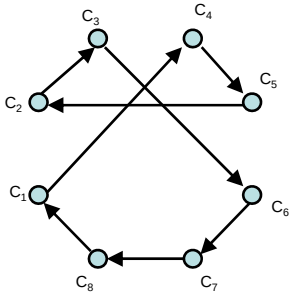
1 5 4 3 2 6 7 8

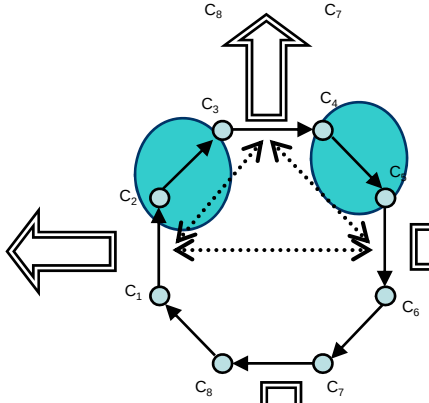
Random switches of three edges

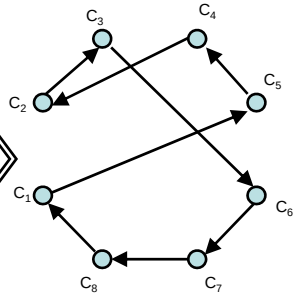


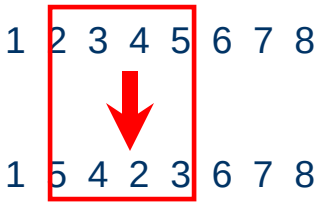


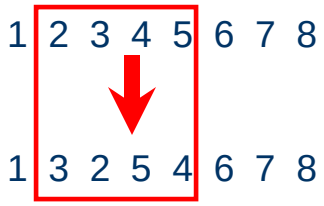


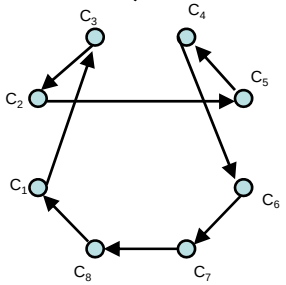






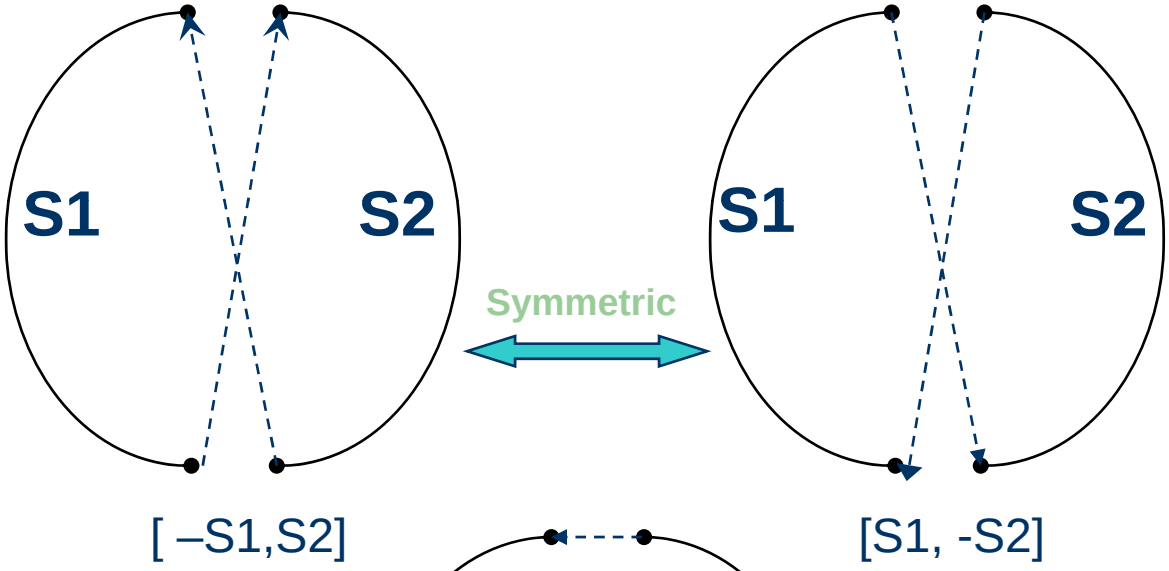






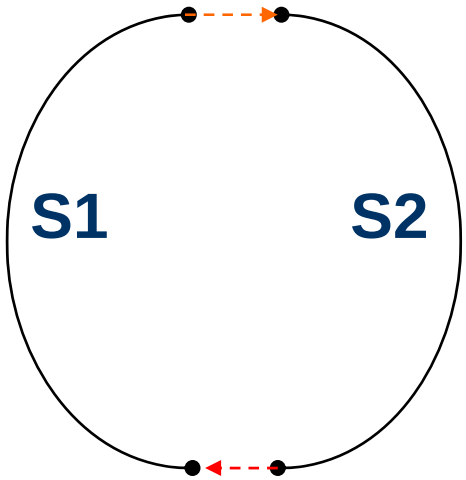
Experimental results show that the three edge change is much more powerful in making the method converge to the global minimum than the other methods. However, its processing cost is also higher, as one would expect.

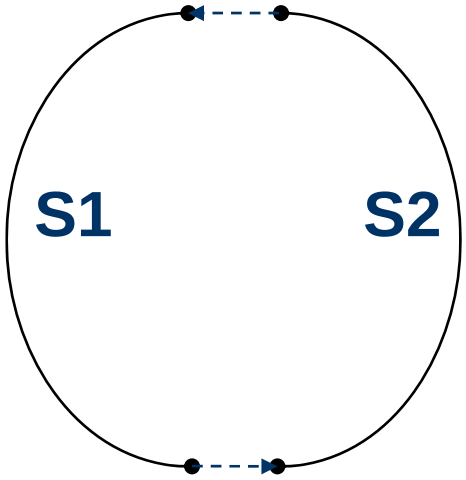
2-opt



[–S1,S2]

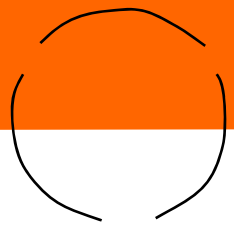
[S1, -S2]



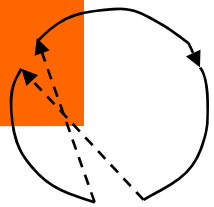


[-S1, –S2]

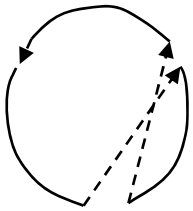
3-opt



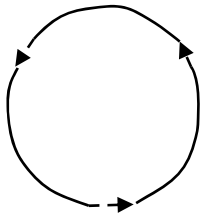
S1 S2 S3



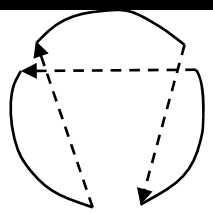
S1 S2 –S3



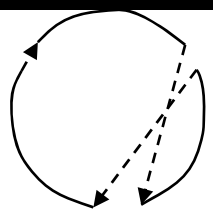
-S1 -S3 S2



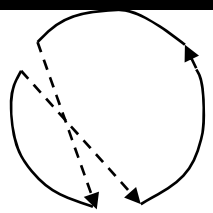
-S1 -S3 -S2



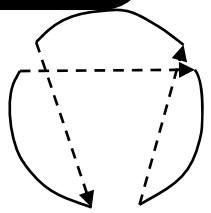
S1 -S2 -S3



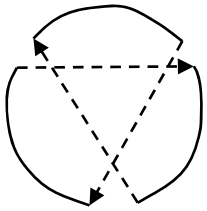
S1 -S2 S3



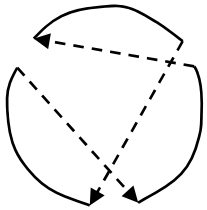
-S1 S3 -S2



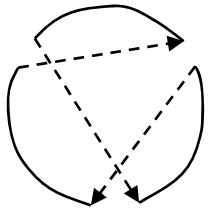
-S1 S3 S2



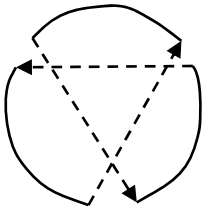
S1 S3 S2



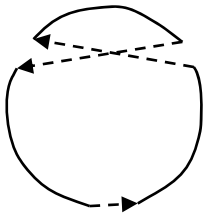
S1 S3 –S2



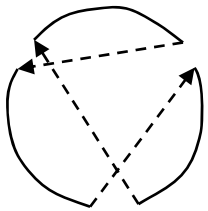
-S1 –S2 S3



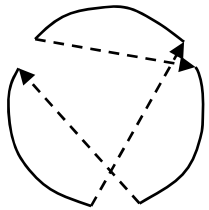
-S1 –S2 –S3



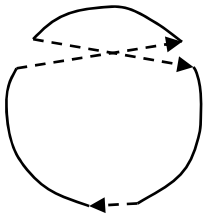
S1 -S3 -S2



S1 -S3 S2



-S1 S2 -S3



-S1 S2 S3

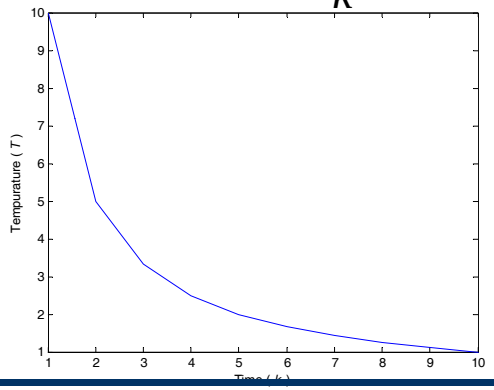
- For symmetric TSP,
  - The new tour is the best one out of 7 new tours.
- For asymmetric TSP,
  - The new tour is the best one in the 15 new tours.
- Larger neighborhood Vs. Smaller neighborhood

  Pro: Larger is better in terms of improvement in each iteration and convergence speed

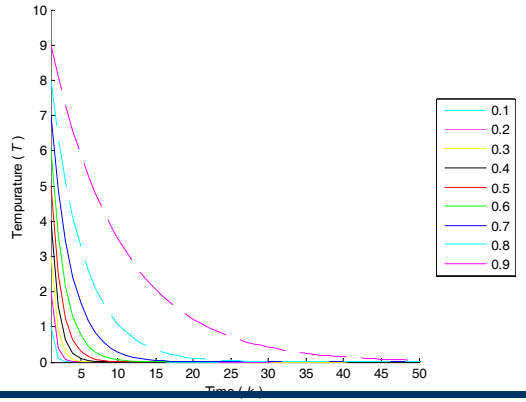
  Con: The searching space increases exponentially

Effect on Cooling Schedule

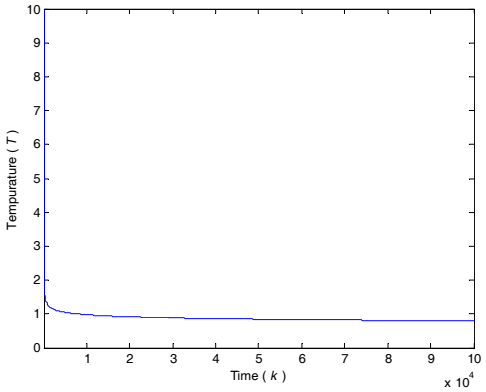
Fast Cauchy $T_k = \frac{1}{k} T_0$



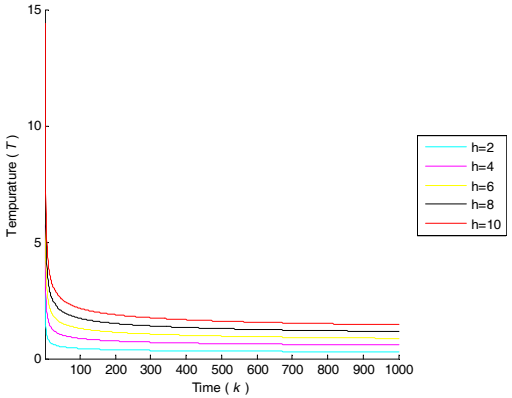
Geometric $T_k = \alpha T_{k-1}$



Boltzmann $T_k = \frac{1}{\ln(k)+1} T_0$



Logarithmic $T_k = \frac{1}{\ln(k+1)} h$



Theoretical Results

The generic SA starts with an initial configuration generated at random. At each step, it selects the next configuration Y from the neighborhood N_x of the current configuration X. The next configuration will be accepted as the current one if its cost is *no greater* than that of the current configuration, otherwise it will only be accepted with probability in Botzmann distribution. The procedure is repeated with a slow decrease of the control parameter T, called temperature, until a sufficiently good solution has been found.

- The performance of generic SA can be analyzed by *nonstationary Markov chain theory*. Suppose the probability of generating the next configuration Y from the current one X is g_{XY}, which is independent of temperature T, and the probability of accepting Y as the new configuration is a_{XY},

$$g_{XY} = \begin{cases} \dfrac{1}{|N_X|}, & X \in S, Y \in N_X \\ 0, & X \in S, Y \notin N_X \end{cases}$$

$$a_{XY}(T) = \min\left\{1, \exp\left(-\frac{E_Y - E_X}{T}\right)\right\}$$

- The one-step transition probability of the nonstationary Markov chain associated with SA are

$$p_{XY}(T) = \begin{cases} \dfrac{1}{|N_X|}\min\left\{1, \exp\left(-\dfrac{E_Y - E_X}{T}\right)\right\}, & X \in S, Y \in N_X \\ 0, & X \in S, Y \notin N_X, X \neq Y \\ 1 - \sum_{Z \in N_X} \dfrac{1}{|N_X|}\min\left\{1, \exp\left(-\dfrac{E_Z - E_X}{T}\right)\right\}, & X \in S, Y \notin N_X, X = Y \end{cases}$$

- S: the configuration space
- N_X: the neighborhood space of the current configuration X
- The convergence of SA has been proven by several researchers, e.g.,
  - S. Geman and D. Geman, “Stochastic rlaxation, Gibbs distribution and the Bayesian restoration of images” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp. 721-742, 1984.
- Given that SA’s generation and acceptance probabilities satisfy the above equations in the previous slide, the SA converges to global minima

$$\lim_{n\to\infty} \text{Prob}\{X \in S^*\} = 1$$

$$S^* = \{X \,|\, X \in S, E_X \le E_Y, \forall Y \in S\}$$

- The rate of temperature decrease is

$$T_n = \frac{h}{\ln(n+1)}$$

where *h* is a problem dependent constant and T_n stands for the temperature at time *n*. A better result is found with

$$T_n = \frac{\lambda}{\sum_{i=1}^{N}\left((\ln)^i\left(\left\lfloor \frac{n}{t} \right\rfloor + n_0\right)\right)}$$

X. Yao and G.J. Li, “Performance analysis of simulated annealing,” in L.W. Chen, B.C. Chen and Q.N. Sun, editors, Proceedings of the Third Pan-pacific Computer Conference, p. 792-980, 1989.

- In practice, we use $T_{n+1} = \rho T_n$
where ρ is a constant selected between 0.8 to 0.99

- *Conjecture*: In earlier stage, *raising the temperature* makes escaping from local minima easier. However, how much changes should be allowed was not addressed?
- *Theorem*: SA with *larger neighborhoods* has a greater probability of arriving at a global optimum than generic SA has if the other conditions, i.e., the initial configuration, initial temperature and cooling rate are kept the same.
- *Guideline*: In earlier stage, allow the next_state to be generated from a larger neighborhood of the current state. As the process continues, the neighborhood should be reduced accordingly. But HOW?

Applications

- Computer-aided VLSI design
  - Simulated annealing for VLSI design, Kluwer Academic, 1988
- Combinatorial optimization
  - Simulated annealing: theory and application, Reidel Publishing, 1987
- Neural network training
  - A learning algorithm for Botzman machine, Cognitive Science, 9: 147-169, 1985
- Image processing
  - Image processing by simulated annealing, IBM Journal of Research and Development, 29: 569-579, 1985
- Code design
  - Using simulated annealing to design good codes, IEEE Trans Information Theory, 33: 116-123, 1987
- Function optimization
  - An empirical study of bit vector function optimization, in Genetic Algorithm and Simulated Annealing, Chapter 13, 170-204, 1987
- and etc

Selected References

- S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by simulated annealing,”,*Science*, **220**(4598), pp. 671-680, 1983.
- B. Moon, H.V. Jagadish, C. Faloutsos, and J.H. Saltz, “Analysis of the clustering properties of the Hilbert space-filling curve,” *IEEE Transactions on Knowledge and Data Engineering*, **13**(1), pp. 124-141, 2001.
- S. Lin and B. Kernighan, “An effective heuristic algorithm for the traveling salesman problem,” *Operations Research,* **21**(2), pp. 498-516, 1973.

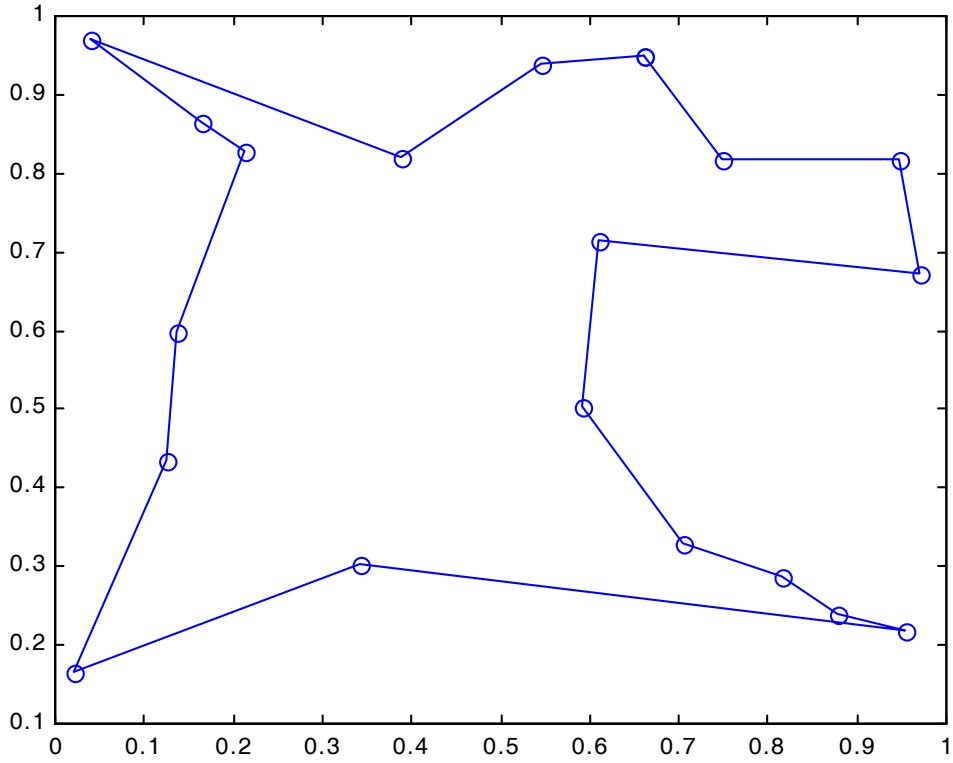
Homework #2- due 10/17/2009

- **Problem #1 (*Combinatorial Optimization*)**

  Develop a *generic* simulated annealing algorithm to solve the traveling salesman problem with 20 cities that are uniformly distributed within a unit square in a 2-dimensional plane. The coordinates of 20 cities are given below in a matrix:

$$\text{cities} = \begin{bmatrix} 0.6606,0.9695,0.5906,0.2124,0.0398,0.1367,0.9536,0.6091,0.8767,0.8148 \\ 0.9500,0.6740,0.5029,0.8274,0.9697,0.5979,0.2184,0.7148,0.2395,0.2867 \\ 0.3876,0.7041,0.0213,0.3429,0.7471,0.5449,0.9464,0.1247,0.1636,0.8668 \\ 0.8200,0.3296,0.1649,0.3025,0.8192,0.9392,0.8191,0.4351,0.8646,0.6768 \end{bmatrix}$$

  Show the “best” route you find and the associated distance with attached computer coding (with *documentation- show your recipe*). An example is given below for reference.

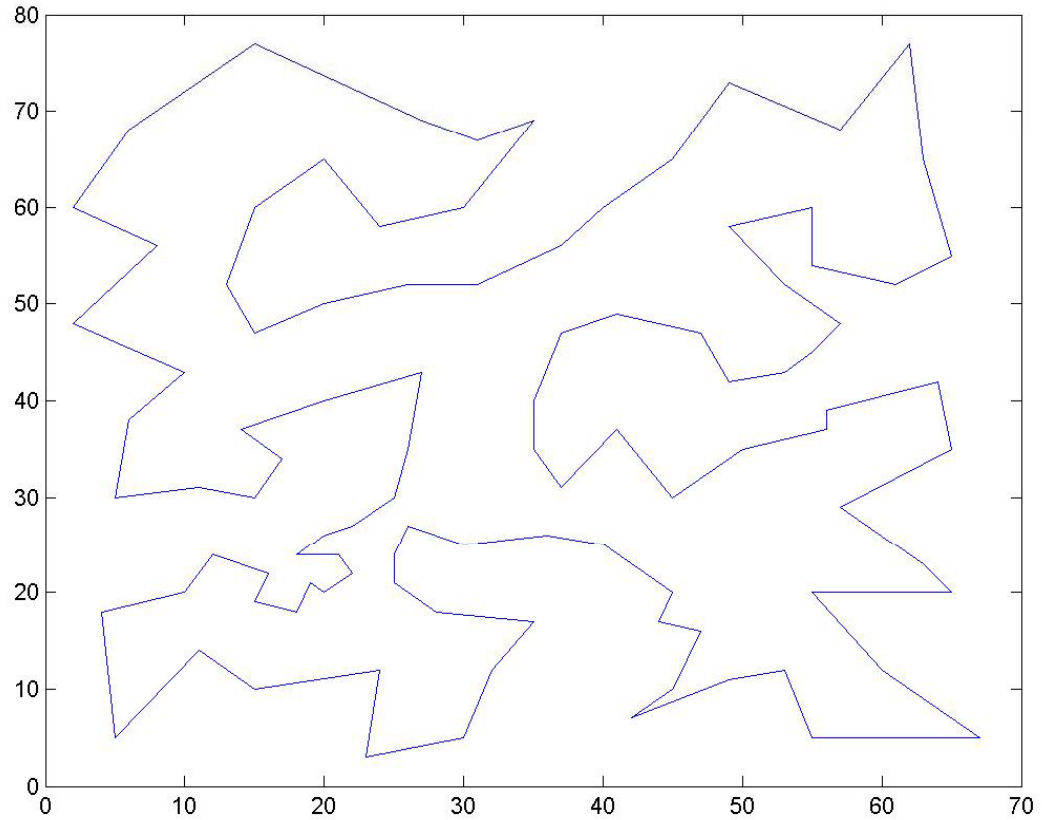

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

- **Problem 2 (*Scalability*)**

  Extend your simulated annealing algorithm to solve the benchmark 101-city symmetric TSP problem (i.e., eil101 due to Christofides and Eilson).The benchmark problem can be found from TSPLIB archive at

  http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

  No need to turn in the codes. Only the best route found (i.e., its configuration and distance in the form of above figure) is to be turned in. This problem is to test if your algorithm can scale up properly.


80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70

- **Problem 3**: Extend your simulated annealing algorithm to solve the benchmark 17-city asymmetric TSP problem (i.e., br17 due to Repetto). The benchmark problem can be found from TSPLIB archive. Turn in only the best route found.

9999	3	5	48	48	8	8	5	5	3	3	0	3	5	8	8	5
3	9999	3	48	48	8	8	5	5	0	0	3	0	3	8	8	5
5	3	9999	72	72	48	48	24	24	3	3	5	3	0	48	48	24
48	48	74	9999	0	6	6	12	12	48	48	48	48	74	6	6	12
48	48	74	0	9999	6	6	12	12	48	48	48	48	74	6	6	12
8	8	50	6	6	9999	0	8	8	8	8	8	8	50	0	0	8
8	8	50	6	6	0	9999	8	8	8	8	8	8	50	0	0	8
5	5	26	12	12	8	8	9999	0	5	5	5	5	26	8	8	0
5	5	26	12	12	8	8	0	9999	5	5	5	5	26	8	8	0
3	0	3	48	48	8	8	5	5	9999	0	3	0	3	8	8	5
3	0	3	48	48	8	8	5	5	0	9999	3	0	3	8	8	5
0	3	5	48	48	8	8	5	5	3	3	9999	3	5	8	8	5
3	0	3	48	48	8	8	5	5	0	0	3	9999	3	8	8	5
5	3	0	72	72	48	48	24	24	3	3	5	3	9999	48	48	24
8	8	50	6	6	0	0	8	8	8	8	8	8	50	9999	0	8
8	8	50	6	6	0	0	8	8	8	8	8	8	50	0	9999	8
5	5	26	12	12	8	8	0	0	5	5	5	5	26	8	8	9999