DM812
METAHEURISTICS

Lecture 2
Simulated Annealing

Marco Chiarandini

Department of Mathematics and Computer Science
University of Southern Denmark, Odense, Denmark

Outline

Outline

Probabilistic Iterative Improv.

Key idea: Accept worsening steps with probability that depends on respective deterioration in evaluation function value:
bigger deterioration $\cong$ smaller probability

Realization:

- Function $p(g, s)$: determines probability distribution over neighbors of s based on their values under evaluation function g.
- Let $\texttt{step}(s, s') := p(f, s, s')$.

Note:

- Behavior of PII crucially depends on choice of p.
- II and RII are special cases of PII.

Example: Metropolis PII for the TSP

- **Search space** S**:** set of all Hamiltonian cycles in given graph G.
- **Solution set:** same as S
- **Neighborhood relation** $\mathcal{N}(s)$**:** 2-edge-exchange
- **Initialization:** an Hamiltonian cycle uniformly at random.
- **Step function:** implemented as 2-stage process:
  1. select neighbor $s' \in N(s)$ uniformly at random;
  2. accept as new search position with probability:

$$p(T, s, s') := \begin{cases} 1 & \text{if } f(s') \leq f(s) \\ \exp \frac{f(s) - f(s')}{T} & \text{otherwise} \end{cases}$$

  (Metropolis condition), where *temperature* parameter T controls likelihood of accepting worsening steps.
- **Termination:** upon exceeding given bound on run-time.

Inspired by statistical mechanics in matter physics:

- candidate solutions $\cong$ states of physical system
- evaluation function $\cong$ thermodynamic energy
- globally optimal solutions $\cong$ ground states
- parameter $T \cong$ physical temperature

Note: In physical process (*e.g.*, annealing of metals), perfect ground states are achieved by very slow lowering of temperature.

Simulated Annealing

Key idea: Vary temperature parameter, *i.e.*, probability of accepting worsening moves, in Probabilistic Iterative Improvement according to annealing schedule (aka *cooling schedule*).

Simulated Annealing (SA):

determine initial candidate solution s
set initial temperature T according to annealing schedule
while termination condition is not satisfied: **do**
    while maintain same temperature T according to annealing schedule **do**
        probabilistically choose a neighbor s' of s using proposal mechanism
        if s' satisfies probabilistic acceptance criterion (depending on T) **then**
            $s := s'$
    update T according to annealing schedule

- 2-stage step function based on
  - proposal mechanism (often uniform random choice from $N(s)$)
  - acceptance criterion (often *Metropolis condition*)
- Annealing schedule
  (function mapping run-time t onto temperature $T(t)$):
  - initial temperature T_0
    (may depend on properties of given problem instance)
  - temperature update scheme
    (*e.g.*, linear cooling: $T_{i+1} = T_0(1 - i/I_{max})$,
    geometric cooling: $T_{i+1} = \alpha \cdot T_i$)
  - number of search steps to be performed at each temperature
    (often multiple of neighborhood size)
  - may be *static* or *dynamic*
  - seek to balance moderate execution time with asymptotic behavior properties
- Termination predicate: often based on *acceptance ratio*,
  i.e., ratio of proposed *vs* accepted steps *or* number of idle iterations

Example: Simulated Annealing for the TSP

Extension of previous PII algorithm for the TSP, with

- *proposal mechanism:* uniform random choice from 2-exchange neighborhood;
- *acceptance criterion:* Metropolis condition (always accept improving steps, accept worsening steps with probability $\exp\left[(f(s) - f(s'))/T\right]$);
- *annealing schedule*: geometric cooling $T := 0.95 \cdot T$ with $n \cdot (n-1)$ steps at each temperature (n = number of vertices in given graph), T_0 chosen such that 97% of proposed steps are accepted;
- *termination:* when for five successive temperature values no improvement in solution quality and acceptance ratio $< 2\%$.

Improvements:

- neighborhood pruning (*e.g.*, candidate lists for TSP)
- greedy initialization (*e.g.*, by using NNH for the TSP)
- *low temperature starts* (to prevent good initial candidate solutions from being too easily destroyed by worsening steps)

1. Call READ_INSTANCE() to read input, compute an upper bound c^* on the optimal solution value, and return the average neighborhood size N.
2. Call INITIAL_SOLUTION() to generate an initial solution S and return $c = cost(S)$.
3. Choose an initial temperature $T > 0$ so that in what follows the *changes*/*trials* ratio starts out approximately equal to *INITPROB*.
4. Set *freezecount* = 0.
5. While *freezecount* < *FREEZE_LIM* (i.e., while not yet "frozen") do the following:

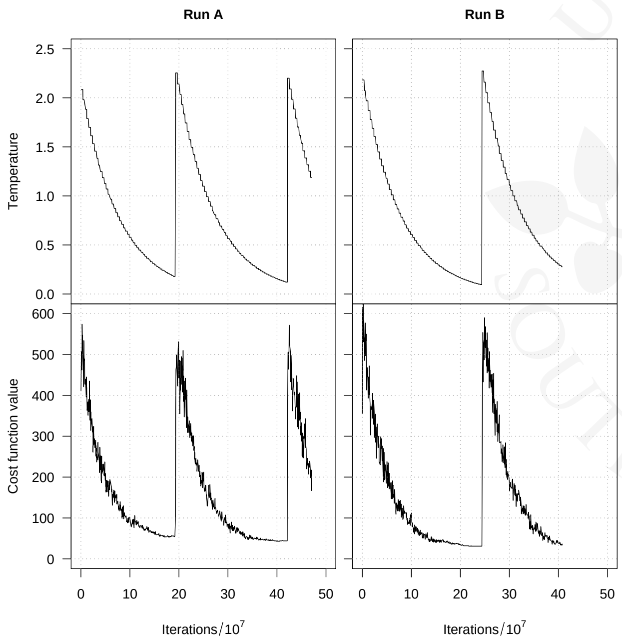
   5.1 Set *changes* = *trials* = 0.
   While *trials* < *SIZEFACTOR*·N and *changes* < *CUTOFF*·N, do the following:

   5.1.1 Set *trials* = *trials* + 1.
   5.1.2 Call NEXT_CHANGE() to generate a random neighbor S' of S and return $c' = cost(S')$.
   5.1.3 Let $\Delta = c' - c$.
   5.1.4 If $\Delta \le 0$ (downhill move),
   Set *changes* = *changes* + 1 and $c = c'$.
   Call CHANGE_SOLN() to set $S = S'$ and, if S' is feasible and $cost(S') < c^*$, to set $S^* = S'$ and $c^* = cost(S')$.
   5.1.5 If $\Delta > 0$ (uphill move),
   Choose a random number r in [0,1].
   If $r \le e^{-\Delta/T}$ (i.e., with probability $e^{-\Delta/T}$),
   Set *changes* = *changes* + 1 and $c = c'$.
   Call CHANGE_SOLN().

   5.2 Set T = *TEMPFACTOR* · T (reduce temperature).
   If c^* was changed during 5.1, set *freezecount* = 0.
   If *changes*/*trials* < *MINPERCENT*, set *freezecount* = *freezecount* + 1.
6. Call FINAL_SOLN() to output S^*.

Profiling

Simulated Annealing
Convergence



Related Approaches (1)

Simulated Annealing
Convergence

Noising Method
Perturb the objective function by adding random noise.
The noise is gradually reduced to zero during algorithm's run.

Threshold Method
Removes the probabilistic nature of the acceptance criterion

$$p_k(\Delta(s,s')) = \begin{cases} 1 & \Delta(s,s') \le Q_k \\ 0 & \text{otherwise} \end{cases}$$

Q_k deterministic, non-increasing step function in k.
Suggested: $Q_k = Q_0(1 - i/I_{MAX})$

Related Approaches (2)

Critics to SA:
The annealing schedule strongly depends on

- the time bound
- the search landscape and hence on the single instance

Evidence that there are search landscapes for which optimal annealing schedules are non-monotone [Hajek and Sasaki, Althofer and Koschnick, Hu, Kahng and Tsao].

Old Bachelor Acceptance
Dwindling expectations

$$Q_{i+1} = \begin{cases} Q_i + incr(Q_i) & \text{if failed acceptance of } s' \\ Q_i - decr(Q_i) & \text{if } s' \text{ accepted} \end{cases}$$

- $decr(Q_i) = incr(Q_i) = T_0/M$
- $Q_i = \left((\frac{age}{a})^b - 1\right) \cdot \Delta \cdot \left(1 - \frac{i}{M}\right)^c$
- ... (self-tuning, non-monotonic)

Outline

'Convergence' result for SA:

Theorem ([Geman and Geman, 1984; Hajek, 1998])

Let $\langle S, f, N\rangle$ be the search landscape of a combinatorial optimization problem with $S^* \neq S$ and S finite. Furthermore, let N be a neighborhood function defined on S that induces a strongly connected, symmetric neighborhood graph with diameter d.

Then the finite homogeneous Markov chain associated with a run of simulated annealing at a fixed value c of the control parameter is strongly ergodic and the unique stationary distribution $q(c)$ to which its probability distribution converges satisfies

$$\lim_{c\to 0} q_i(c) = 0$$

for any non-optimal solution $i \in S$.

'Convergence' result for SA:

Theorem ([Geman and Geman, 1984; Hajek, 1998])

Let $\langle S, f, N\rangle$ be the search landscape of a combinatorial optimization problem with $S^* \neq S$ and S finite. Furthermore, let N be a neighborhood function defined on S that induces a strongly connected, symmetric neighborhood graph with diameter d.

If a cooling schedule is assumed in which the sequence $\{c_k\}_{k=1}^{\infty}$ of control parameter values is non-increasing and satisfies both $\lim_{k\to\infty} = 0$ and

$$c_k \geq \frac{d\Delta}{\log k}$$

with $\Delta = \max_{i\in S, j\in N(i)}(f(j) - f(i))$, then the inhomogeneous Markov chain associated with a run of simulated annealing is strongly ergodic and the stochastic vector q to which its probability distribution converges satisfies $q_i = 0$ for any non-optimal solution.

Example

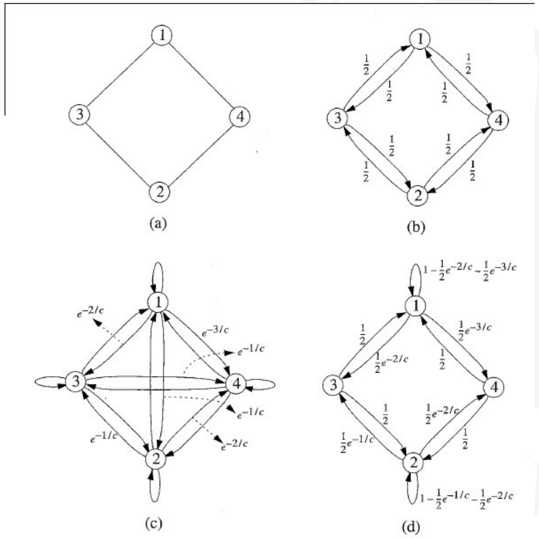
Mathematical modelling of SA

$$G = \begin{pmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix},$$

r solutions are numbered such that $f(i) = i$. The gener
Figure 8.2(b). The acceptance matrix

$$A(c) = \begin{pmatrix} 1 & \exp\left(-\frac{1}{c}\right) & \exp\left(-\frac{2}{c}\right) & \exp\left(-\frac{3}{c}\right) \\ 1 & 1 & \exp\left(-\frac{1}{c}\right) & \exp\left(-\frac{2}{c}\right) \\ 1 & 1 & 1 & \exp\left(-\frac{1}{c}\right) \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$P(c) = \begin{pmatrix} \alpha_1(c) & 0 & \frac{1}{2}\exp\left(-\frac{2}{c}\right) & \frac{1}{2}\exp\left(-\frac{3}{c}\right) \\ 0 & \alpha_2(c) & \frac{1}{2}\exp\left(-\frac{1}{c}\right) & \frac{1}{2}\exp\left(-\frac{2}{c}\right) \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix}$$



$q(3) = (0.38, 0.28, 0.20, 0.14)$
$q(1) = (0.64, 0.24, 0.09, 0.03)$
$q(0.1) =$
$(1, 5 \cdot 10^{-5}, 2 \cdot 10^{-9}, 9 \cdot 10^{-14})$

Simulated Annealing
Convergence

Note:

- Practical relevance for combinatorial problem solving is very limited (impractical nature of necessary conditions)
- In combinatorial problem solving, *ending* in optimal solution is typically unimportant, but *finding* optimal solution during the search is (even if it is encountered only once)!