Setting Parameters for Simulated Annealing

• All heuristic algorithms (and many nonlinear programming algorithms) are affected by “algorithm parameters”

• For Simulated Annealing the algorithm parameters are

• $T_0$, $M$, $\alpha$, $\beta$, maxtime

• So how do we select these parameters to make the algorithm efficient?

Handout for videotaped lecture labeled 9-2-11 (taped after regular lecture on 9-2-11)
Setting Parameters in Simulated Annealing

• As we saw in the first simulated annealing problem, the results can depend a great deal on the values of the parameter $T$ (“temperature”), which depends upon $T_0$ and upon $\alpha$.

• How should we pick $T_0$ and $\alpha$?

• We can use some simple procedures to pick an estimate a reasonable value (not necessarily optimal) of $T_0$.
  
  – user decides on the probability of $P_1$ of accepting an uphill move initially and $P_2$ the probability of accepting an uphill move after $G$ iterations.
Rough Estimates for Good Algorithm Parameter Values

• Note that the best algorithm parameters cannot be determined exactly and that the results are probably not sensitive to small changes in algorithm parameter values.

• Recall in previous SA example there was a difference between a $T_o$ of 100 and a $T_o$ of 500 in the two examples. This is a difference of 500% in the algorithm parameter value—not a small change.

• Hence you are trying to get a rough estimate of what would be good algorithm parameter values.
BOARD: change in search over time

- Allow uphill moves (not greedy) => P large enough to reasonably allow uphill move.
- Greedy => P is small. Unlikely to make uphill moves.

Period to determine good parameter value

"Global" search period

"Local" search period

Numbers of objective function evaluations

0 G MaxTime (stop here)
Simple Approach to Parameterizing SA

- Assume M=1
- Assume
  - you want to do Maxtime iterations and you want Maxtime-G of these to be greedy. (Here greedy means having a probability of less than P2 for uphill move.)
  - The estimated range of values of Cost(S) = R
  - R= Max Cost (S)-Min Cost (S)
  - Initially you want to accept $P_1$ fraction of the moves.
  - By the time you have done G iterations you want to have an essentially greedy search and accept $P_2$ or fewer fraction of the moves.
  - $P_2<<P_1$ (For example, $P_1=.95$ and $P_2 = .01$)
Selecting Algorithm Parameters

• So how do we select these parameters to make the algorithm efficient?
• Recall that the principle is that initially you want to accept a relatively high percentage of uphill moves (for a minimization problem), but the more iterations you have done, the less you want to accept uphill moves.
• Acceptance of uphill moves is determined by $T_0$, $M$, and $\alpha$. Assume $M = 1$, then $T_i = \alpha^i T_0$ starting with $T = T_0$.

Let $P1$ be your goal for the average probability of accepting an UPHILL move initially and $P2$ be your goal for the probability of accepting an Uphill move after $G$ iterations.
Simulated Annealing Morphs into What?

• Initially Simulated Annealing has a significant probability of accepting an uphill move.

• After many iterations, Simulated Annealing becomes similar to what algorithm (Random Walk, Random Sampling, or Greedy Search)? Why?
Given $T_0$, Select $\alpha$

- Once $\alpha^i$ gets small enough after $i$ iterations, then SA operates like ________search. WHY? $P$ is too small unlikely to go uphill

- You have maxtime as the maximum number of iterations.

- Decide what fraction of these you want to have spent in greedy search.

- Let $G = \text{maxtime} - \text{number of iterations for greedy search}$ and let $P_2 > 0$, be a small number that allows very few uphill movements. ($P_2$ selected by user)
BOARD $P$ is probability of acceptance of uphill move

$$P_{c} = e^{\frac{-\Delta \text{cost}}{B \alpha i T_{0}}}$$

$P$ is plotted against $i$, the number of iterations, reaching 0 at $maxiter$. The graph shows a decreasing trend as $i$ increases.
Effect of iteration number $i$ on Probability

- After $i$ iterations, $T = \alpha^i T_0$. (Decreasing)
- We know that $P$ decreases as $T$ decreases. Thus,

Pick the number of iterations $G$ when you want the probability of acceptance to be a low $P_2$. 

$P_1$ \hspace{3cm} $P_2$

$P$ \hspace{3cm} Iteration $i$ \hspace{3cm} maxtime
Simple Method for Setting $T_o$

- Estimate the average of $\Delta\text{Cost}$, (called $\text{avg}\Delta\text{Cost}$) for uphill moves in the initial iteration. We will discuss later ways to estimate this.
- Decide that you want initially to allow approximately $P_o$ percent of the uphill moves to $S_{\text{new}}$ moves to be accepted. (User decision)
- Then given $P_o$ and $\text{avg}\Delta\text{Cost}$, can you give an equation that includes $T_o$?

$$P_1 = e^{-\frac{(\text{avg}\Delta\text{cost})}{T_o}}$$

is approximately what you want, and you can solve for $T_o$. So $T_o = (-\text{avgCost})/(\ln P1)$
Estimate for $\alpha$

- Recall that our estimates for $T_0$ and $\alpha$ are rough estimates of what would be good values, i.e. within an order of magnitude.

- To estimate a good value for $\alpha$, we need to consider how fast do we want the probability of accepting an uphill move to decline.

- We define this by the user deciding what probability ($P_2$) do you want after $G$ iterations.

- Recall Simulated annealing becomes like Greedy Search if the probability of uphill moves gets small and this probability is determined by $\alpha^g T_0$ after $g$ iterations.
Selecting a value for $\alpha$

- The probability $P$ in $k$th iteration of accepting uphill move is

$$P_k = \exp\left(-\frac{\text{COST}(\text{NewS})-\text{COST}(\text{CurS})}{T_k}\right)$$

We also have

$$T_k = T_{k-1} \alpha = (T_{k-2} \alpha) \alpha = \ldots = T_0 \alpha^k$$

(The lower numbers are subscripts for iteration number. However the upper $k$ on $\alpha$ is an exponential power). So the term that appears in the probability equation in the $G$th iteration is

$$P_2 = \exp\left(-\frac{\text{COST}(\text{NewS})-\text{COST}(\text{CurS})}{\alpha^G T_0}\right)$$

$$P_2 \text{ approximately } = \exp\left(-\frac{\text{avg} \Delta \text{COST}}{\alpha^G T_0}\right)$$

If we have already computed $T_0$, and we are given $P_2$, $G$ and $\text{avg} \Delta \text{COST}$.

Hence we can compute $\alpha$. 

Selecting $\alpha$

After $G$ iterations and average $\Delta Cost$, $T = \alpha^G \, T_o$, so we compute the appropriate value of $\alpha$ using the equations on the following slide. Recall $P_2$ is the probability of accepting an uphill move after $G$ iteration. If we assume average $\Delta cost$ (="avgCost" for short) then

$$P_2 = \exp \left( - \frac{\text{avgCost}}{T} \right)$$
Computing $\alpha$, Given $T_0$, $P2$ and $G$

\[
P2 = \exp(-\text{avgCost} / T)
\]

\[
P2 = \exp(-\text{avgCost} / \alpha^G T_0)
\]

\[
\ln P2 = \ln \exp(-\text{avgCost} / \alpha^G T_0)
\]

\[
\ln P2 = -\text{avgCost} / \alpha^G T_0
\]

\[
\alpha^G = -\text{avgCost} / T_0 \ln P2
\]

\[
\alpha = (-\text{avgCost} / (T_0 \ln P2))^{1/G}
\]

Previous handwritten slide in videotaped lecture erroneously had “log” instead of “ln”