Introduction to Artificial Intelligence Lab

Lab 6: Simulated Annealing Algorithm

Taxonomy

Simulated Annealing is a global optimization algorithm that belongs to the field of Stochastic Optimization and Metaheuristics. Simulated Annealing is an adaptation of the Metropolis-Hastings Monte Carlo algorithm and is used in function optimization. Like the Genetic Algorithm, it provides a basis for a large variety of extensions and specialization's of the general method not limited to Parallel Simulated Annealing, Fast Simulated Annealing, and Adaptive Simulated Annealing.

Inspiration

Simulated Annealing is inspired by the process of annealing in metallurgy. In this natural process a material is heated and slowly cooled under controlled conditions to increase the size of the crystals in the material and reduce their defects. This has the effect of improving the strength and durability of the material. The heat increases the energy of the atoms allowing them to move freely, and the slow cooling schedule allows a new low-energy configuration to be discovered and exploited.

Metaphor

Each configuration of a solution in the search space represents a different internal energy of the system. Heating the system results in a relaxation of the acceptance criteria of the samples taken from the search space. As the system is cooled, the acceptance criteria of samples is narrowed to focus on improving movements. Once the system has cooled, the configuration will represent a sample at or close to a global optimum.

Strategy

The information processing objective of the technique is to locate the minimum cost configuration in the search space. The algorithms plan of action is to probabilistically re-sample the problem space where the acceptance of new samples into the currently held sample is managed by a probabilistic function that becomes more discerning of the cost of samples it accepts over the execution time of the algorithm. This probabilistic decision is based on the Metropolis-Hastings algorithm for simulating samples from a thermodynamic system.

Procedure

Algorithm (below) provides a pseudocode listing of the main Simulated Annealing algorithm for minimizing a cost function.

Input: ProblemSize, $iterations_{max}$, $temp_{max}$
Output: S_{best}
$S_{current} \leftarrow$ CreateInitialSolution(ProblemSize)
$S_{best} \leftarrow S_{current}$
For ($i = 1$ **To** $iterations_{max}$)
  $S_i \leftarrow$ CreateNeighborSolution($S_{current}$)
  $temp_{curr} \leftarrow$ CalculateTemperature(i, $temp_{max}$)
  If (Cost(S_i) $\leq$ Cost($S_{current}$))
    $S_{current} \leftarrow S_i$
    If (Cost(S_i) $\leq$ Cost(S_{best}))
      $S_{best} \leftarrow S_i$
    End
  ElseIf (Exp($\frac{CostS_{current} - CostS_i}{temp_{curr}}$) > Rand())
    $S_{current} \leftarrow S_i$
  End
End
Return (S_{best})

Figure 1. Pseudocode for Simulated Annealing.

Heuristics

- Simulated Annealing was designed for use with combinatorial optimization problems, although it has been adapted for continuous function optimization problems.

- The convergence proof suggests that with a long enough cooling period, the system will always converge to the global optimum. The downside of this theoretical finding is that the number of samples taken for optimum convergence to occur on some problems may be more than a complete enumeration of the search space.

- Performance improvements can be given with the selection of a candidate move generation scheme (neighborhood) that is less likely to generate candidates of significantly higher cost.

- Restarting the cooling schedule using the best found solution so far can lead to an improved outcome on some problems.

- A common acceptance method is to always accept improving solutions and accept worse solutions with a probability of $P(accept) \leftarrow \exp(\frac{e-e'}{T})$, where T is the current temperature, e is the energy (or cost) of the current solution and e' is the energy of a candidate solution being considered.

- The size of the neighborhood considered in generating candidate solutions may also change over time or be influenced by the temperature, starting initially broad and narrowing with the execution of the algorithm.

- A problem specific heuristic method can be used to provide the starting point for the search.

Exercise

In this week, you implement a Simulated Annealing algorithm to solve a scheduling problem. Table 1 and 2 describes a scheduling problem. Each entry in Table 1 represents how long it takes for switching from one job to another job. For example, the entry [1,2] ([Job1, Job2]=12) is the time for switching job 1 to job 2 in Table 1. The entries in Table 2 is the time a machine completes the corresponding job. For example, the entry [2,3] ([Machine2, Job3]=5) is the time machine 2 completes job 3.

	Job1	Job2	Job3
Job1	0	12	10
Job2	4	0	8
Job3	6	10	0

Table 1

	Job1	Job2	Job3
Machine1	10	4	8
Machine2	12	9	5

Table 2

The problem in here is to assign each of the 3 jobs to one of the machines in such a way that the total time to complete all jobs is minimized. Implement a simulated annealing algorithm for this problem.