CSCI3344

Applying Simulated Annealing on N-Queens Problem

In this project, we'll investigate an optimization method called simulated annealing (see textbook page 125). As the name implies, the search method mimics the process of annealing. Annealing is the physical process of heating and then cooling a substance in a controlled manner. The desired result is a strong crystalline structure, compared to fast untempered cooling which results in a brittle defective structure. The structure in question is our encoded solution, and the temperature is used to determine how and when new solutions are accepted.

NATURAL MOTIVATION

The structural properties of a solid depend upon the rate of cooling after the solid has been heated beyond its melting point. If the solid is cooled slowly, large crystals can be formed that are beneficial to the composition of the solid. If the solid is cooled in a less controlled way, the result is a fragile solid with undesirable properties.

When the solid has reached its melting point, a large amount of energy is present within the material. As the temperature is reduced, the energy within the material decreases as well. Another way of cooling at annealing is as a process of "shaking". Where at higher temperatures, there exists higher molecular activity within a physical system. Consider the shaking of a box containing a physical landscape, where a marble is moved around randomly searching for the global minimum. At higher temperatures, the marble is able to move freely around the landscape, while at lower temperatures the shaking is reduced, leading to less movement of the marble. The goal is to encounter the global minimum while shaking violently. As the temperature is reduced, it is less likely that the marble will be moved from the minimum. This is the search process as borrowed from annealing.

SIMULATED ANNEALING ALGOREITHM

Let's now look at how the metaphor of cooling a molten substance can be used to solve a problem. The simulated annealing algorithm is very simple and can be defined in six steps (see Figure 1)

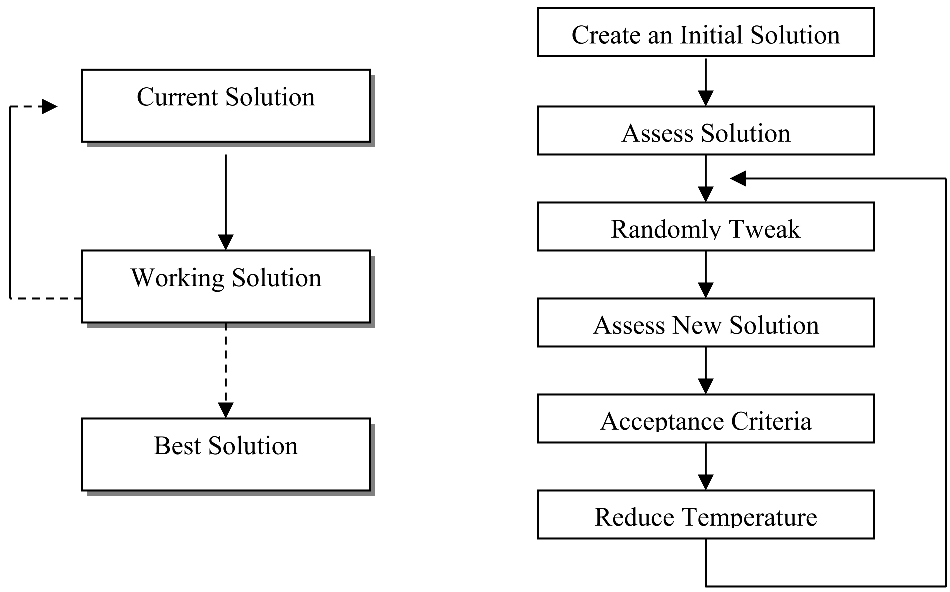

Figure 1 Simulated annealing algorithm

Initial Solution

For most problems, the initial solution will be a random one, or one found in a previous run. This gives the algorithm a base from which to search for a more optimal solution to the problem.

Assess Solution

Assessing the solution consists of decoding the current solution and then performing whatever action is necessary to evaluate it against the given problem. Note that the encoded solution may simply consist of a set of variables. These variables would be decoded from the current solution and the *energy* of the solution assessed based upon how well it solved the given problem.

Randomly Tweak Solution

We then randomly modify the working solution. How the working solution is modified depends upon the encoding. Consider an encoding of the Traveling Salesman Problem (TSP), where each element represents a unique city. To tweak this working solution, we pick two elements and swap them. This keeps the solution consistent by not allowing duplicates of cities or any cities from being removed from the solution.

Once the working solution has been tweaked, we assess the solution as defined in the previous step.

Acceptance Criteria

At this point in the algorithm, we have two solutions. The first is our original solution called the current solution and the second is the tweaked version called the working solution. Each has an associated energy, which is the strength of the solution (let's say that the lower the energy, the better the solution).

Our working solution is then compared to the current solution. If the working solution has less energy than the current solution (i.e., is better solution), then we copy the working solution to the current solution and move on to temperature reduction.

However, if the working solution is worse than the current solution, we evaluate the acceptance criteria to figure out what to do with the current working solution. The probability of acceptance is based on Equation 1 (which is based upon the law of thermodynamics)

$$P(\Delta E)=\exp\left(-\frac{\Delta E}{T}\right) \qquad (1)$$

What this means is more easily visualized in Figure 2. At high temperatures (> 60 degrees Celsius), worse solutions are accepted more often than they are rejected. If the delta energy is lower, the probability is even higher for acceptance. As the temperature decreases, so does the probability for accepting a worse solution. With decreasing temperature, higher delta energies contribute to lower acceptance probabilities.

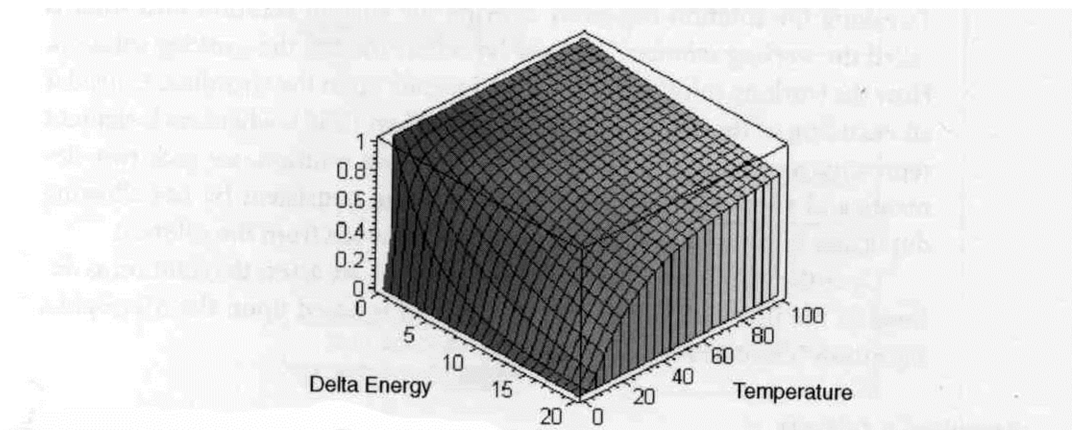

Figure 2 Visualization of Acceptance Probability

At higher temperatures, simulated annealing permits acceptance of worse solutions in order to search more of the available solution landscape. As the temperature decreases, the search allowable also decreases until equilibrium is reached when the temperature reaches 0 degrees C.

Reduce Temperature

After some number of iterations through the algorithm at this temperature, we reduce the temperature by small amount. Large varieties of cooling schedules exist, but in this example we'll use a simple geometric function (see Equation 2):

$$T_{i+1} = \alpha T_i \qquad (2)$$

Constant α is less than one. Many other cooling strategies are possible, including linear and non-linear functions.

Repeat

A number of iterations will be performed at a single temperature. When that set of iterations is complete, the temperature is reduced and the process continues until the temperature reaches zero.

SAMPLE ITERATION

To help illustrate the algorithm, let's run through a couple of iterations to see how it works. Note that when the working solution has a lower energy (representing a better solution) than the current solution, it's always used. Only when the working solution is worse than the current solution does the acceptance probability equation come into play.

Let's say in our current environment that the temperature is 50 C and the current solution has energy of 10. We copy the current solution to the working solution and tweak it. After assessing the energy, the new working solution has energy of 20. In this case, the energy for the working solution is worse (higher) than the original solution and therefore we use the acceptance criteria.

Current solution energy	10
Working solution energy	20

The Delta energy for this sample is (working solution energy-current solution energy), or 10. Plugging this delta, and our temperature of 50, into Equation 1, we get a probability:

$$P = \exp\left(\frac{-10}{50}\right) = 0.818731$$

Therefore, for this sample we see that it's very likely that the worse solution will be propagated forward.

Let's now look at an example near the end of the temperature schedule. Our temperature is now 2 and the energies for the current and working solutions are:

Current solution energy	3
Working solution energy	7

The delta energy for this sample is four. Using this delta and the temperature, we again use Equation 1 to find the probability of acceptance:

$$P = \exp\left(\frac{-4}{2}\right) = 0.135335$$

Given the low probability result of this sample, it's very unlikely that the working solution will be propagated on to the subsequent iterations.

That’s the basic flow of the algorithm. Let’s now apply simulated annealing to a problem and see how well it actually works.

<u>SAMPLE PROBLEM</u>

For this algorithm, we’ll look at a very famous problem that has been attacked by a wide variety of search algorithms. The N-Queens Problem (or NQP) is defined as the placement of N queens on an N-by-N board such that no queen threatens any other queen using the standard rules of chess (see Figure 3).

The 8-Queens problem was first solved in 1850 by Carl Friedrich Gauss. The search algorithm, as can be inferred by the date of the solution, was trial and error. The N-Queens Problem has since been solved using depth-first search (1987), divide and conquer (1989), genetic algorithms (1992), and a variety of other methods. In 1990, Rok Socic and Jun Gu solved the 3,000,000-Queens problem using local search and conflict minimization.

		Q					
Q							
						Q	
				Q			
							Q
	Q						
			Q				
					Q		

Figure 3 One of 92 8-Queens solutions

Solution Encoding

The encoding for the N-Queens solution is standard one that takes into account the final solution, and thus reduces the search space. Note from Figure 3 that only one queen can be found in each row and in each column. This constraint makes it much easier to create an encoding that will be manipulated by the simulated annealing algorithm.

Since each column contains only one queen, an N-element array will be used to represent the solution (see Figure 4).

The N-element array represents the row indices of queen placement. For example, in Figure 4, column 1 contains the value 5, which represents the row for which the queen will be placed.

5	7	1	4	2	8	6	3

	1	2	3	4	5	6	7	8
1			Q					
2					Q			
3								Q
4				Q				
5	Q							
6							Q	
7		Q						
8						Q		

Figure 4 N-Queens solution encoding

Creating a random solution is also a very simple process. We initialize our solution with each queen occupying the same row as its column. We then walk through each column and pick a random number from 1 to N for each. The two elements are then swapped (the current column with the randomly selected column). When we reach the end, the solution will be randomly perturbed.

Finally, given the encoding, there are never horizontal or vertical conflicts on the board. Therefore, only diagonal conflicts need to be checked when assessing the energy of the solution.

Energy

The energy of a solution is defined as the number of conflicts that arise given an encoding. The goal is to find an encoding that exhibits energy of zero, or no conflicts on the board.

Temperature Schedule

For this problem, we'll begin with a temperature of 30 degrees C and step down to zero using a geometric schedule (as defined in Equation 2). Our source will utilize 0.98 as α. As we'll see later, the temperature schedule shows a fast decline and slow convergence to the final temperature of zero.

At each temperature change, we'll perform 100 steps. This permits the algorithm a number of search steps at this plateau.

<u>SAMPLE RUN</u>

Let's look at the result of a sample run. In this run, we'll start the temperature at 100 to illustrate the algorithm (as shown in Figure 5).

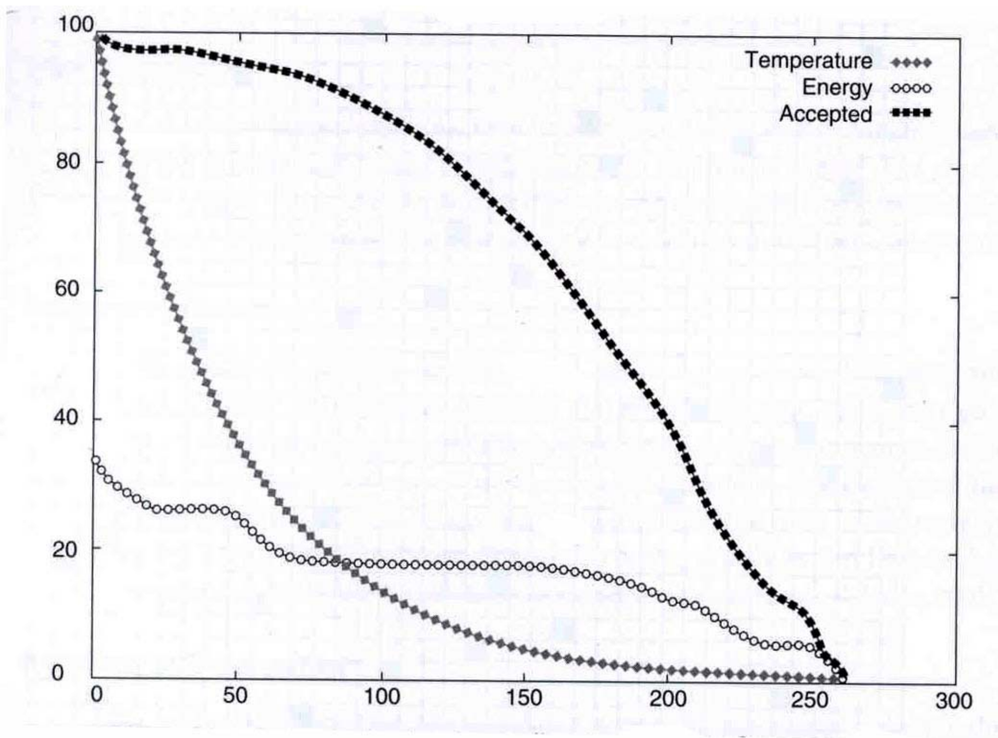

Figure 5 Plot of sample simulated annealing run for the 40-Queens problem

The element falling sharply from 100 to 0 is the temperature. The cooling schedule uses Equation 2. The element falling less sharply is the number of accepted worse solutions (based upon the acceptance probability Equation 1). Since the acceptance probability is a function of the temperature, a correlation is easily seen. As this graph shows, the perfect solution is not found until very near the end. This is true in all runs because of the acceptance criteria equation. The sharp drop of the "accepted" graph element at the end of the plot is indicative if this – a sharp reduction in the acceptance of worse solutions can be seen.

The actual 40-Queens solution, as represented by Figure 5, is shown in Figure 6.

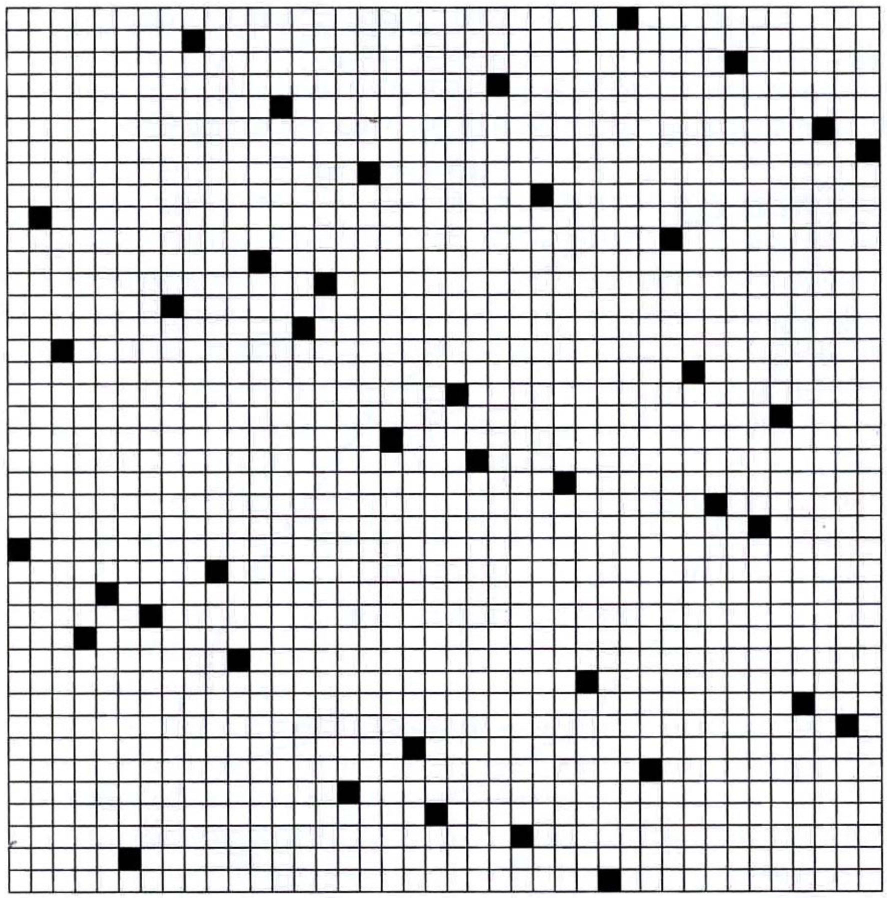

Figure 6 Sample solution to the 40-Queens problem

ADJUSTING THE ALGORITHM PARAMETERS

The parameters for the algorithm can be adjusted depending upon the complexity of the problem to be solved. This section will define some of these parameters along with the effects that can be realized.

Initial Temperature

The initial temperature must be high enough to permit movement of solutions to other parts of the solution landscape. The initial temperature can be dynamically defined. Given statistics on the rate of acceptance of worse solutions and discovery of new better solutions, the temperature can be raised until a sufficient number of acceptances/discoveries have occurred. This is similar to heating a solid until it is liquid, after this there is no point in further increasing the temperature.

Final Temperature

While zero is a good symbolic final temperature, the geometric function used within this example means that the algorithm will run far longer than practically useful. Therefore, the final temperature function used, this value may vary.

Temperature Function

The actual temperature function used can be varied depending upon the problem to be solved. Figure 5 shows the temperature over time using a geometric function. A large variety of other temperature cooling functions can be used. These functions may result in a steep reduction in the first half of the temperature schedule, or a slow reduction followed by a steep loss in temperature.

Iterations at Temperature

At higher temperatures, the simulated annealing algorithm looks for the global optimum over the entire solution landscape. As cooling occurs, there is less movement and the algorithm searches the local optimum to optimize it. The number of iterations to perform at each temperature step is therefore important. You can start with 100 iterations to be performed. It’s useful to manually experiment with the number of iterations to find what’s optimal for the problem at hand.