Jeffrey Hyman's Summer Research at UCLA

J. D. Hyman*

August 18, 2006

Abstract

We examine three Monte Carlo sampling methods on idealized examples. Implementing the Metropolis algorithm on created functions, we compare the single temperature to four temperatures running in parallel. Using simulated annealing and parallel tempering on the traveling salesman problem, we examine what each of the methods is capable of exploring. We focus specifically on parallel tempering's ability to find local minima. Presenting a basic algorithm for temperature set selection in parallel tempering, I conclude with ideas for further research and a personal critique of my summer research at UCLA.

*St. Olaf College. Research partially supported by the REU program of the UCLA RTG, NSF grant DMS-0601395

1 Introduction

The methods known as Monte Carlo, gather rough estimates through sampling, and were named Monte Carlo by scientist at Los Alamos National Laboratory in the 1940's, because of their reliance on probability and resemblance to many games of chance. The existence of similar methods can be seen as far back as Buffon's needle experiment that arrived at an estimate of π through random sampling. The power of Monte Carlo comes from its ability to sample large regions with little computational force.[1].

Monte Carlo methods have been used consistently since their development and one particular method, the Metropolis Algorithm [2], has become a standard. Many other Monte Carlo methods are based upon the Metropolis Algorithm, including the two which we focus on, simulated annealing and parallel tempering. Our goal is to focus on these two particular methods and investigate what they are capable of doing.

We begin sampling on idealized examples to work on our methods. Then we apply the methods to a well known problem, the traveling salesman problem(TSP).

Our preliminary results show that the two methods are capable exploring different parts of a problem. Simulated annealing is a tool of optimization and can aid in finding the global minimum of a tour. While, more interestingly, parallel tempering is able to expose the energy landscape of the tour.

During our study we realized the importance of temperature sets in parallel tempering, and in order to expedite our work created a simple algorithm for the selection of temperatures. The method is feedback oriented and relies on educated guesses before attempting a temperature set.

[1]Kalos and Whitlock

[2]Outlined in detail : Section 2

2 Metropolis

The Metropolis Algorithm for advanced sampling was first presented in a paper by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller in 1953.[3] The algorithm is often refereed to as the Metropolis algorithm, due to his name being first on the paper.

The method is a Monte Carlo sweep method that randomly samples moves and accepts all moves that have a lower energy state, but also accepts a move to an increased energy state with a certain probability. That probability is

$$P(x \to y) = e^{-\Delta E(y-x)/T}. \qquad (1)$$

Where $\Delta E(y-x)$ is the difference in energy between the original point and the location of the new move. T is the temperature is used to control how big of uphill moves are accepted. Lower temperatures accept smaller moves, and larger temperatures accept greater ones. The power of this algorithm is found in its acceptance of some moves to an increased lever of energy. By accepting these uphill moves, the sampling may escape from local minima.

An extensive discussion of the Metropolis algorithm can be found in *Monte Carlo Methods, Volume I: Basics* By Kalos and Whitlock.

3 Simulated Annealing

The process of simulated annealing was created by S. Kirkpatrick, C. D. Gelatt, and

[3]Equation of State Calculations by Fast Computing Machines, Jour. Chem. Phys. June 1953

M.P. Vecchi[4]. Their process is based on the Metropolis algorithm where the temperature is slowly decreased to allow smaller and smaller moves. Figure 1 shows an a Metropolis sample being annealed. Notice how the width sampled is slowly decreased as the number of iterations increased. This change is the result of the temperature decreasing which allow for smaller uphill moves.

Kirkpatrick and colleagues found that simulated annealing could be used as a optimization method where,due to the size of the problem, standard methods failed. The process is able to sample large regions at the beginning of a run, and slowly moves into minimum as the temperature decreases. It hopes to retain a large enough temperature to exit local minimum but by the time the global minimum arises it cannot escape. The downfall of the method is its inability to confirm that it has found the optimal solution. It can only confirm that its has the best local minimum of the minima sampled.

4 Parallel Tempering

Parallel Tempering might be thought of as running multiple Metropolis samples simultaneously, and every so often a swap of two temperatures is attempted. This *swap* between two temperatures, exchanges part of the information between the two plots. The advantage of this swapping between temperatures comes in two different fashions. First, if a lower temperature in the set is in a local minimum for which it cannot escape, it may swap positions with a higher temperature and be placed in another area to explore. That high temperature can escape the local minimum and continue its sampling. Another advantage gained by parallel tempering is its lower computation time than simulated annealing. Simulated annealing has to check all the temperatures inbetween its highest to lowest temperature, while parallel tempering only has the number of temperatures assigned.

The swaps between temperatures occur with a probability that looks similar to the Metropolis method's acceptance of uphill moves.

Let t_n and t_m be neighboring temperatures in a parallel tempering swapping set, where $t_n < t_m$. Given that a swap between the temperatures is attempted the acceptance of the swap is defined by the probability

$$P(\text{accepts swap}) = min\{1, e^{(\Delta B * \Delta E)}\} \quad (2)$$

Where $\Delta B = (1/t_m - 1/t_n)$ and $\Delta E = E(t_m^k) - E(t_n^k)$. Where $E(t^k)$ is the energy of the temperature at time k.

Figure 2 shows two temperatures swapping, and just below 500 iterations they move back and forth between two tour distances until one samples out of its minima.

Earl and Deem discuss the parallel tempering origins and possible future applications with great vigor [5]. The full potential of parallel tempering has yet to be tapped, along with its wide range of possible applications including work with protein folding, and polymers.

5 Implementation on Idealized examples

To begin our analysis we make a function to implement our methods on. The function

[4] **Optimization by Simulated Annealing** *Science*, May 13 1983

[5] Earl, D.J., Deem, M.W. *Parallel Tempering:Theory, Applications, and New Perspectives* Jan 4, 2006, Physical Chemistry Chemical Physics

we created has multiple minima, so that we could experiment with uphill acceptance of moves and swapping between temperatures in parallel tempering.

5.1 Parallel Tempering and Energy Functions

We created a function and ran both a single temperature Metropolis method and a four temperature parallel tempering method. The function $f(x)$ is defined as

$$g(x) = \begin{cases} 1+\sin(2\pi * x): -2 \leq x < -1.25 \\ 2(1+\sin(2\pi * x)): -1.25 \leq x < -.25 \\ 3(1+\sin(2\pi * x)): -.25 \leq x < .75 \\ 4(1+\sin(2\pi * x)): .75 \leq x < 1.75 \\ 5(1+\sin(2\pi * x)): .75 \leq x \leq 2 \end{cases}$$

$$f(x) = -g(x) + 9 \tag{3}$$

Figure 3 shows $f(x)$ graphed over $-2 < x < 2$.

We run a single temperature Metropolis sample on $f(x)$ and figure 4 shows a histogram of the sampling taking place. Some of the minima are exposed, but other parts of the graph are completely unexplored.

Next we apply parallel tempering with four temperatures swapping. Figure 5 is a histogram of the same temperature as the Metropolis but with parallel tempering also taking place. The peeks of the histogram correspond to the the valleys of the graph, and the temperature was able to explore multiple minimum and exposes the energy landscape of the function.

Parallel tempering not only shows the global minimum but also shows the location of the local minimum in a energy system.

6 TSP

A salesman wants to find the shortest distance for a trip he has to take to N different cities. He wishes to visit each city only once and end where he began. How does he go about finding the optimal tour distance for this traveling salesman?

The traveling salesman problem(TSP) has become a classic in the world of computational mathematics and has been studied extensively. Due to this wealth of information concerning the problem, it is a perfect subject to test new methods on. Comparing the solutions of the new methods to the existing data allows for the accuracy of the method to be gauged.

We choose to apply simulated annealing and parallel tempering on the TSP because it is a problem for which Monte Carlo sweeps and the Metropolis algorithm are applicable.

We created a tour of 100 cities with Matlab's pseudo-random generator, where we paired up i and $i+1$ as x and y coordinates to plot the cities on a unit square. Allon Percus[6] found the optimal tour lengthn for our particular city set to be 7.064.

6.1 Nearest Neighbor Heuristic Method

The Nearest Neighbor Heuristic Method is a construction heuristic method that helps to get a tour within 20 percent of its optimal tour length. The method is based on a greedy algorithm where only down hill moves are accepted. We start with selecting one city and calculate the distance from it

[6]Dr. Percus was our resident expert on the TSP and should be credited with greatly increasing the pace of the project.

to all other cities. Then select the shortest distance as the next city in the tour. The process repeats itself which each new city in the tour, but only checks the distance to unaffiliated cities. The method results in a tour distance with many good moves at the beginning and a few terrible distances at the end, due to limited cities available. Figure 6 shows our cities with the random tour, and Figure 7 shows our cities with the nearest neighbor construction and periodic boundary conditions implemented. Before sampling with either simulated annealing or parallel tempering, we use the nearest neighbor heuristic to rearrange our city set, to be closer to the optimal tour.

6.2 Simulated Annealing

Applying simulated annealing to the TSP as a optimization method was done by Kirtpatrick, and has been a common use of the method ever sense. Our examination of simulated annealing on the TSP was short lived and over-shadowed by work with parallel tempering, due to the interesting new results parallel tempering exposed.

6.3 Parallel Tempering

We choose to run parallel tempering on the TSP with certain variables fixed and certain ones as what we would experiment changing. Fixed Variables: Number of cities = 100, Number of cites swapped per move 1 pair. Our variables were t_{max}, t_{min}, Spacing of temperatures(linear, geometric) and the number of iterations between attempted temperature swaps. After running the nearest neighbor heuristic construction we let the temperatures attempt any move for the first 100 iterations so that the tours would be different.

In our version of the TSP a move consisted of randomly selecting two cities in the city set and swapping their place in the order of the set.

The first three non-fixed parameters are discussed in section 6.3.1; in our trials changing the number of moves between swaps we tested: every iteration, every ten iterations, and every 1000 iterations. However, one should note that just because a swap is attempted it does not mean that one is completed. Figure 8, 9, 10 are all geometrically spaced samples running for 1000 iterations, with temperatures in the key. Figure 8 found a best tour distance of 8.0329. Figure 9 found a best tour distance of 8.0144. Figure 10 found a best tour distance of 8.015.

From these figures one sees how attempting swaps often can pull lower temperature out of minima before they are able to explore. Attempting swaps not often enough causes the system to lose the power of parallel tempering.

A major issue involved progress in the problem is finding a temperature set that optimizes exploration and is discussed in section 6.3.1.Often times a low temperature gets stuck in a minimum and is swapped out of the location, however, that low temperature leaves a trail that it was stuck for a while[7]. The swapping temperatures move downhill gradually, and thus the higher temperatures are able to get close enough to stuck low ones that a swap will be excepted and progress may continue. Starting with a nearest neighbor construction we were able to make progress towards the optimal tour distance. Our best local minimum of 7.6952 was found using a set of five temperatures of .01,.012,.015,.017,.02 run over 50,000 it-

[7]This is explored further in section 6.3.2

erations.

6.3.1 Comparing Temperature Sets

A major influence on the performance of parallel tempering on the TSP comes from the selection of the temperature sets. We set $t_{max} = .02$ and $t_{min} = .01$ and attempt temperature swap every 10 iterations. We chose two ways to selected the temperature, first geometrically and the second linearly. Figure 11 shows the geometrically selected temperatures, which find a minimum of 7.9946 after 10,000 iterations. Figure 12 shows the linearly selected set, which found a minimum of 8.0534 over 10,000 iterations.

In selection of temperature sets, focus should primarily be on t_{min} and t_{max}. t_{max} should be large enough to escape from any local minimum, but not so large that it undoes the tour distance gained by the nearest neighbor construction. We set an upper bound on t_{max} at .5101. Our reason for this selection was given periodic boundary conditions the maximum distance between any two cities is .7071 and the minimum is 0, therefore the average distance is .3536. Solving equation 1 for probability of accepting a move to equal .5 where ΔE was .3536, we received .5101 to be the largest temperature we would need.

On the other hand t_{min} needs to be small enough so that it could not escape most local minimum but would still be able to explore them. Which leads us to the next section, Local Minima exploration.

6.3.2 Local Minima Exploration

Similar to our idealized examples, parallel tempering on the TSP is able to find local minima. Creating a histogram of the lower temperatures we are able to see the locations of the local minima. Figure 13 and figure 14 are corresponding graphs to showing three distinct minima. Figure 14 shows three distinct clusters of samples at ≈ 7.75, ≈ 7.9 and ≈ 8.08 for tour distance. These clusters correspond to Three local minima shown in figure 13 that the lowest temperature get stuck in.

7 Temperature Selection Algorithm

A major problem that arose during implementation of parallel tempering on the TSP was being able to control how the temperatures interacted with one another. In particular, the probability that given an attempt for a swap occurs, that the swap will occur. This probability relies upon the temperature set. An algorithm that, given certain information about a problem, would select other temperatures was created. It is a very straight forward approach, solving equation 2 for t_n. It is described in terms of the TSP, but may be applied to other problems as well.

Every set of cities has a particular temperature set that optimizes sampling. Once one has a run a few random temperatures and found a range that does not get immediately stuck, nor climbs away, they may input certain parameters into the algorithm and gain temperature sets. These parameters are: the maximum distance between temperatures for which a swap of temperatures wishes t occur, the probability that a swap occurs, and a temperature at a tail ends of the set, either t_{min} or t_{max}.

One should make sure that the computation of the sampling only attempts a swap with one pair of temperatures for a given attempt of swapping, else the algorithm does not work. The failure occurs because if t_1

and t_2 swap and then an attempt of swapping t_2 and t_3 occurs, the difference of distance between t_2 and t_3 is no longer the original distance; rather, it is the distance between t_1 and t_3 and the probability of accepting the swap is decreased.

1 Choose t_{min}, (call this t_1) and find average distance for t_1 call this (d_{t_1}).

2 Choose a ΔD that is the largest difference in distances between d_{t_1} and d_{t_2} for which you wish to accept a swap of temperatures. d_{t_2} is the average distance for t_2, Note $d_{t_2} = d_{t_1} + \Delta D$.

3 Define $\Delta B = 1/t_2 - 1/t_1$ and $\Delta E = (d_{t_1} + \Delta D) - d_{t_1}$.

4 Let S_1 be the event that the opportunity for a swap between t_1 and t_2 occurs. Let R_1 be the event that a swap of temperature t_1 and t_2 occurs. Define P_1 as the probability that a swap occurs between t_1 and t_2 given an attempt occurs.

$$P_1 = P(R_1|S_1)$$

Pick a value for P_1 which is the probability you wish to accept the given swap.

5

$$\begin{aligned} P_1 &= e^{(\Delta B * \Delta E)} \\ P_1 &= e^{(1/t_2 - 1/t_1) * ((d_{t_1} + \Delta D) - d_{t_1})} \\ 1/t_2 &= ln(P_1)/\Delta D + 1/t_1 \end{aligned}$$

Note that the right side of the equation has a numeric value given our parameter choices. Solve this equation for t_2 and we have our second temperature.

6 Once we have t_2, implement these temperatures in parallel tempering. Find the average distance for t_2 when it is swapping with t_1, then restart the algorithm to find the rest of your temperatures.

This algorithm may also be used starting at a given T_{max} and working backwards to lower temps. Note that for the larger temps one will want a larger ΔD because the average distance between temperatures is larger.

8 Summary and Further Research

From my work this summer I found that this most influential parameter in parallel tempering is the probability that given a swap is attempted, the event will occur. It is fair to say this parameter controls parallel tempering. If the probability of swapping out of a energy location is very high, the temperatures will never have time to explore. Conversely if the probability of swapping temperatures is too low, the power of parallel tempering is lost because swapping rarely occurs. Further research ought be done to find the optimal probability of swapping.

Simulated annealing was used, but our focus was primarily on parallel tempering, and thus no new information was found concerning simulated annealing. However, if a good temperature set was found for parallel tempering, one could create an annealing process that mimics parallel tempering and compare the two as optimization methods.

Using parallel tempering to explore the distribution of the local minima is clearly

the next step that this research should take. Support for the effectiveness of this method comes from both our trials on functions and the TSP. The ability to expose energy landscapes carries great potential and ought be explored.

However, the process of parallel tempering is dependent upon finding a optimal temperature for any set of cities. In taking a step back and working on a good algorithm for temperature selection, one finds a problem that has many more applications than the TSP. To my knowledge there is no universal system for temperature set selection in parallel tempering. The lack of such a system could be due to the differences of the problems that parallel tempering can be applied.

Another direction that this research could lead would be using simulated annealing with a fast cooling rate to freeze the system in a local minimum. This fast paced annealing, might also expose the energy landscape of the tour. An idea I had which I barely put into practice, was a combination parallel tempering simulated annealing process where the temperature set in parallel tempering would also be annealed. The value of this would be the temperatures would always be moving down and yet keep their configurations with one another, and still have the power of swapping. Clearly the entire temperature set would have to start higher, but it would be able to explore more valleys fuller and still be able to get out of the minima.

9 Personal Critique

I am including a section called personal critique both for my own good as part of reflection and to acknowledge those who aided in this summer's research. I arrived at UCLA not sure as to what my project would be. I knew that I had some preparation in the subject matter from course work at St. Olaf College[8], but was unsure of the direction or the circumstances of the research. I had been sent a couple of papers, and a title. I was fortunate to have seemingly unlimited resources in both of my mentors, Russ Caflisch and Richard Wang, and to have connected with Allon Percus who aided my project in in-numerous ways during the time my mentors were away on travel.

I am unsure of how much other undergraduate research typically covers in one summer but I feel like our project covered a lot of ground. This expedient work was the result of pre-arrival preparation, luck in finding resources, and an open direction of the project. I arrived every morning having a idea to try out and leaving in the afternoon with twenty more to try tomorrow. But more importantly I had a wonderful group that encouraged me to continue exploring my ideas. Occasionally I would create a tangent for myself, but my mentors would check in and reset me on course. This research has helped me personally to understand what a career in mathematics might be like. My deepest thanks to Russ Caflisch, Richard Wang, Allon Percus, Andrew Bernoff, Jon Azose, James M. Hyman, Matt Richey IPAM, NSF, and UCLA for their aid in preparation and execution of this summer REU. Cheers!

List of Figures

1 A Metropolis algorithm with simulated annealing 9
2 Two Metropolis samples with swapping occurring 9

[8] Matt Richey in particular

3 f(x) 9
4 A histogram of a single temperature sampling f(x) 9
5 A histogram of a single temperature sampling f(x), while swapping with 3 other temperatures running in parallel . 10
6 A tour of 100 cities without the nearest Neighbor Heuristic method 10
7 A tour of 100 cities with the nearest Neighbor Heuristic Method and Periodic boundary conditions 10
8 Parallel tempering on the TSP attempting swaps every iteration 10
9 Parallel tempering on the TSP attempting swaps every 10 iterations 11
10 Parallel tempering on the TSP attempting swaps every 100 iterations 11
11 Parallel Tempering with five temperatures selected Geometrically run for 10,000 Iterations 11
12 Parallel Tempering with five temperatures selected linearly run for 10,000 Iterations . . . 11
13 Parallel Tempering with five temperatures used to show how parallel tempering can explore local minima 12
14 A histogram of the temperature .0075 from figure **??** exposing local minima 12

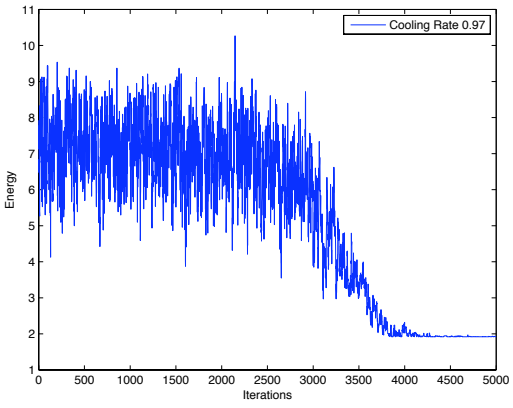

Figure 1: A Metropolis algorithm with simulated annealing

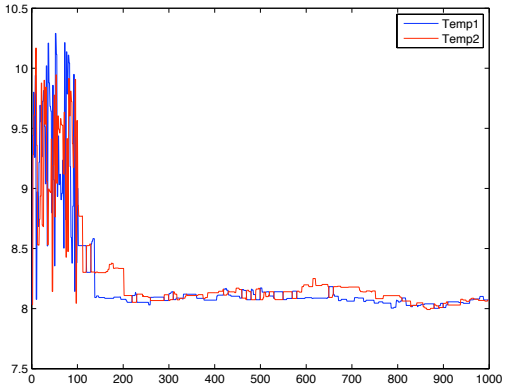

Figure 2: Two Metropolis samples with swapping occurring

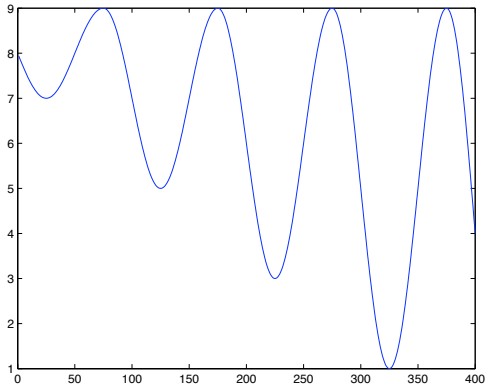

Figure 3: f(x)

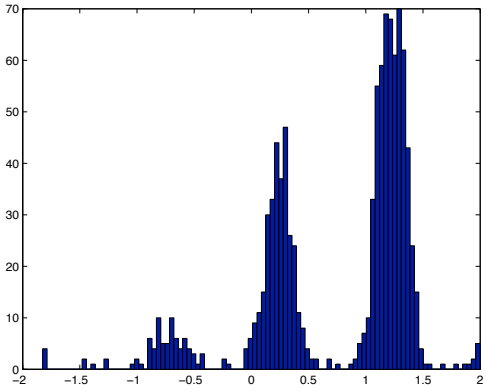

Figure 5: A histogram of a single temperature sampling f(x), while swapping with 3 other temperatures running in parallel

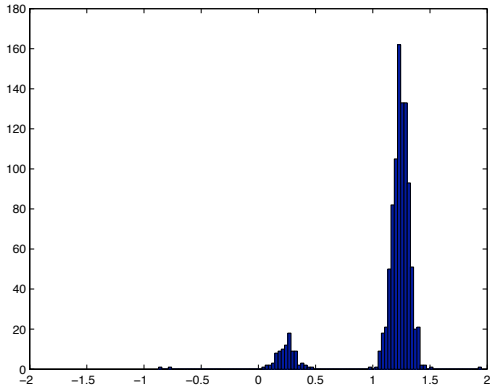

Figure 4: A histogram of a single temperature sampling f(x)

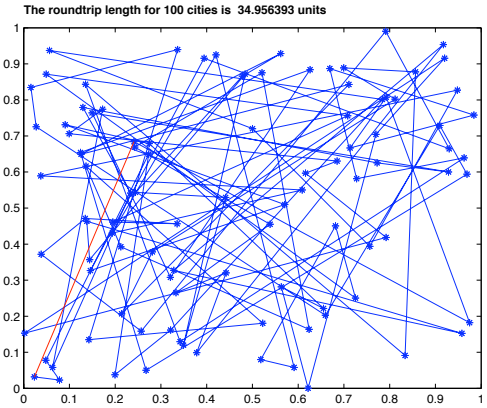

Figure 6: A tour of 100 cities without the nearest Neighbor Heuristic method

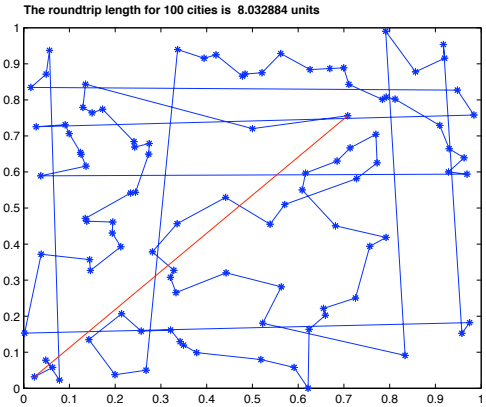

Figure 7: A tour of 100 cities with the nearest Neighbor Heuristic Method and Periodic boundary conditions

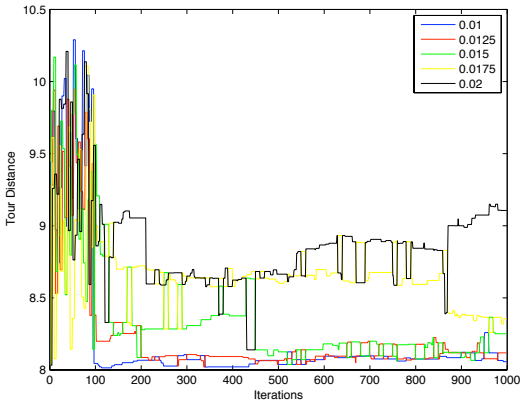

Figure 9: Parallel tempering on the TSP attempting swaps every 10 iterations

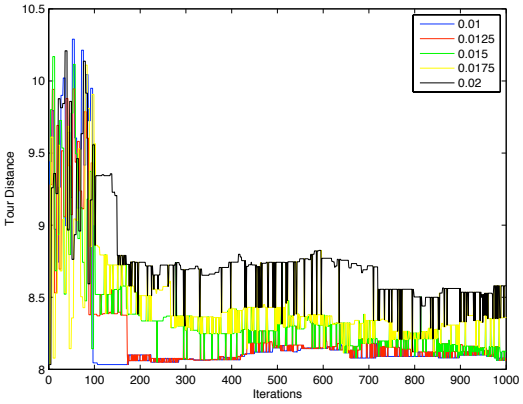

Figure 8: Parallel tempering on the TSP attempting swaps every iteration

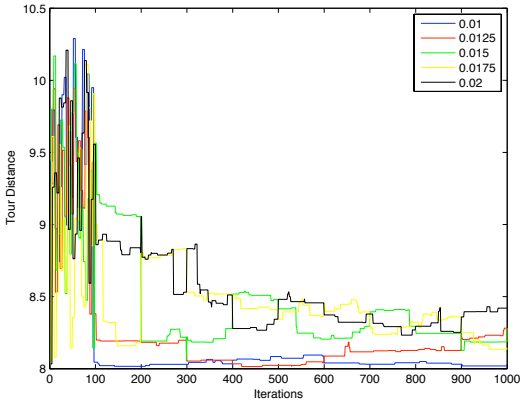

Figure 10: Parallel tempering on the TSP attempting swaps every 100 iterations

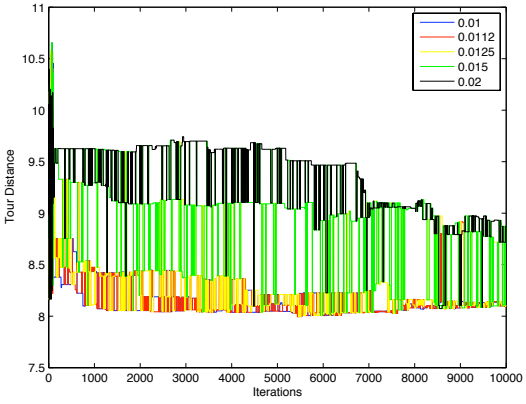

Figure 11: Parallel Tempering with five temperatures selected Geometrically run for 10,000 Iterations

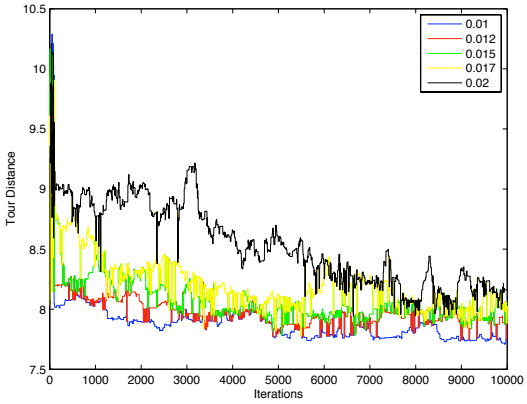

Figure 13: Parallel Tempering with five temperatures used to show how parallel tempering can explore local minima

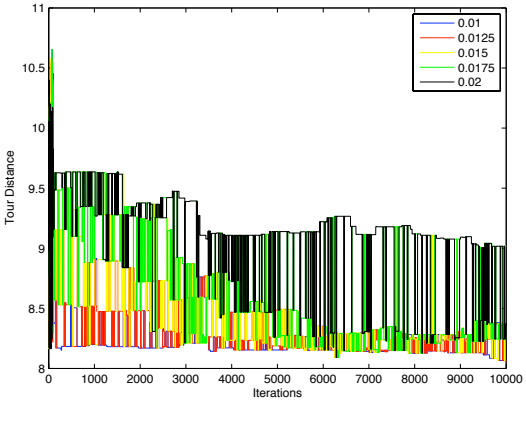

Figure 12: Parallel Tempering with five temperatures selected linearly run for 10,000 Iterations

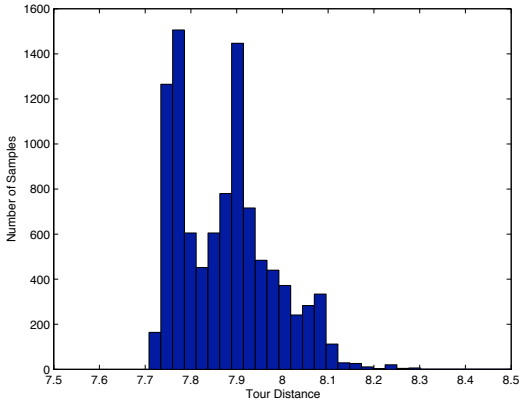

Figure 14: A histogram of the temperature .0075 from figure 13 exposing local minima