

Intelligent Systems Assignment 1

Wessel Becker (1982362) & Sander ten Hoor (2318555)

September 20, 2016

1 MatLab code

1.1 Changes to tsp.m

The decrease T by 0.1% was removed to make T a constant value.

```
for jstep=1:ceil(maxsteps);  
%     temps = temps * 0.999;  
for ins = 1:100
```

1.2 plotmean.m

```
n = 250;  
maxsteps = 1000;  
temp = [1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001];  
method = 1;
```

```
meanresults = zeros(size(temp));  
standarddeviationresults = zeros(size(temp));  
  
% Try the different temperatures  
for index = 1:length(temp)  
    results = tsp(n, maxsteps, temp(index), method);  
    lastFifty = results(length(results) - 49:length(results));  
    meanresults(index) = mean(lastFifty);  
    standarddeviationresults(index) = sqrt(var(lastFifty));  
    fprintf('mean: %f \n', mean(lastFifty));  
    fprintf('variance: %f \n', var(lastFifty));  
end
```

```
% Plot the results in a graph  
figure(3); semilogx(0,0); hold on;  
semilogx(temp, meanresults);  
errorbar(temp, meanresults, standarddeviationresults)  
title([ 'n = ', num2str(n, '%d'), ' ...  
      ' Steps = ', num2str(maxsteps, '%d') ], ...  
      'fontsize', 16);
```

```

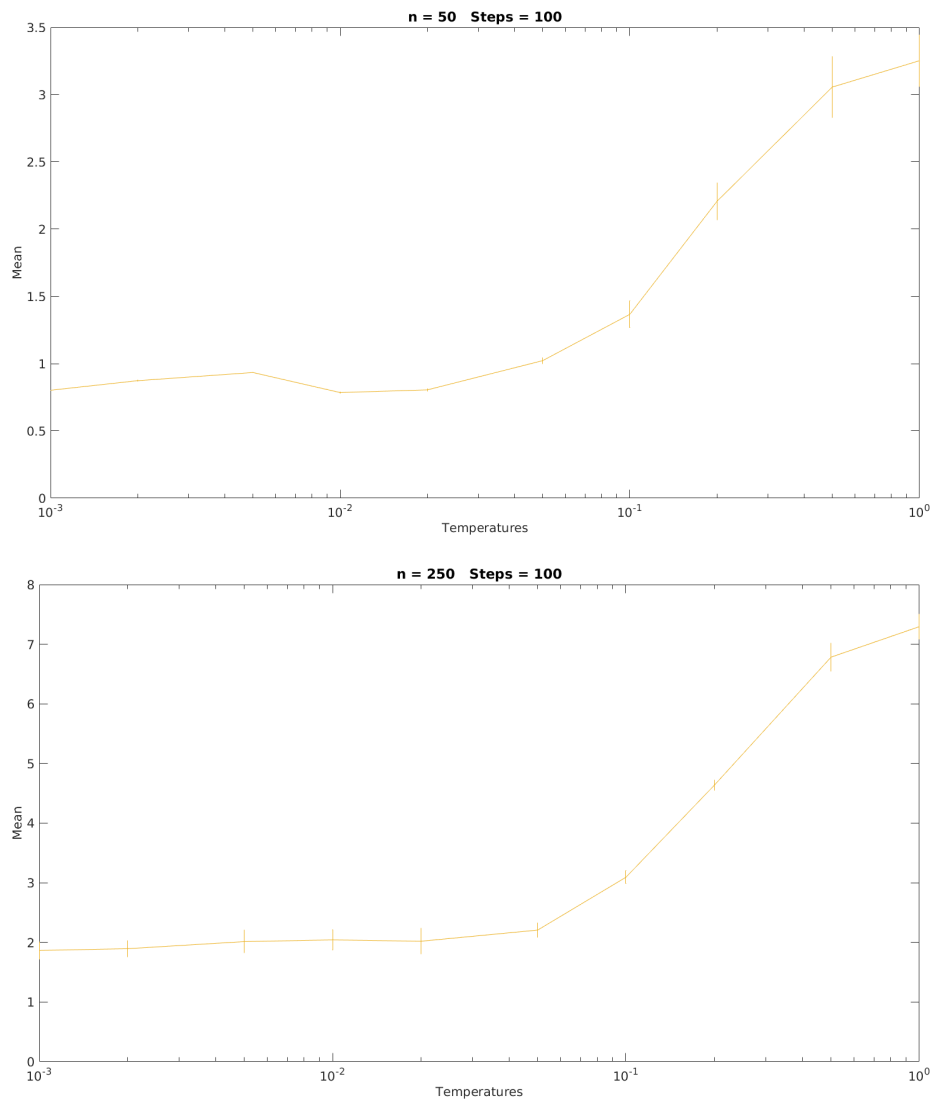
set(gca, 'fontsize', 16);
xlabel(['Temperatures'], 'fontsize', 16);
ylabel(['Mean'], 'fontsize', 16);
set(gca, 'fontsize', 16)

```

2 Plots

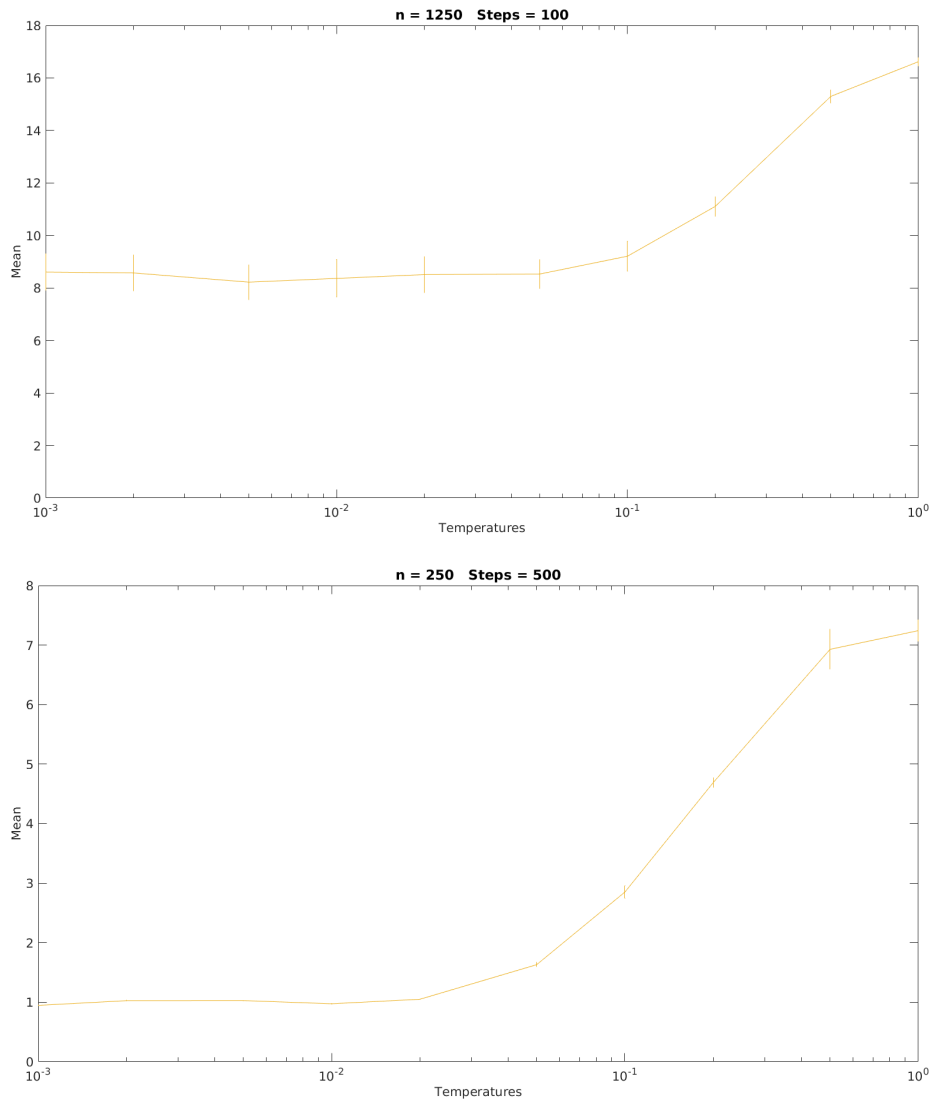
To show the impact of the temperature, several values for the temperature have been used to use with the tsp function. The tsp function returns the generated distance values. The last fifty of these values are used to plot the mean normalised path distance against the used temperature.

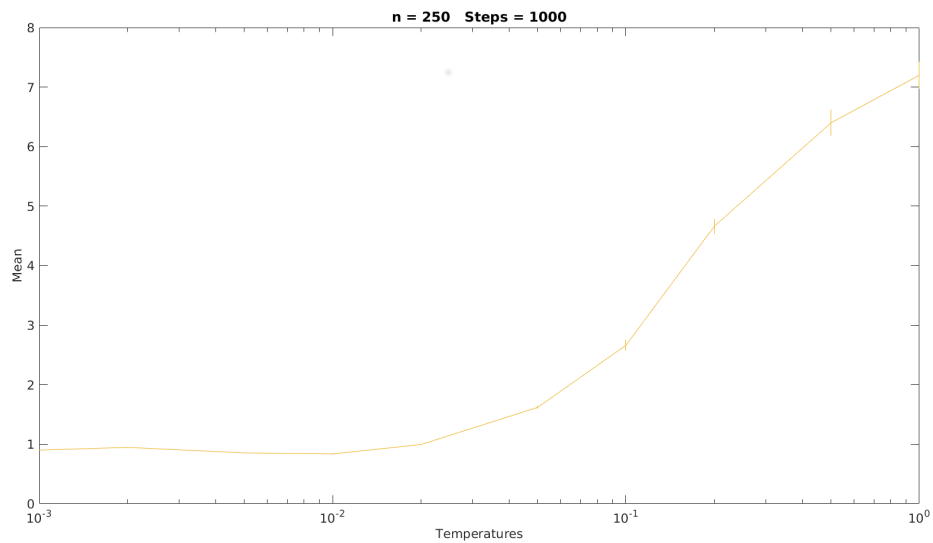
These plots show the value of the temperature on the x-axis against the mean normalised path distance on the y-axis. The standard deviation is displayed around the data points.



Increasing the amount of cities from 50 to 250 also shows an increase of the mean

(the y-axis has increased to 8 from 4 to accomodate for the mean) and the standard deviation. It also shows something odd with the standard deviation. At the higher temperatures, it seems to be quite small compared to the standard deviation at lower temperatures. It is possible that the cause for this is the high degree of random route-changes. The high temperature results in a less greedy algorithm, and thus the distance needs to decrease less. However, this is true for the other plots as well, as shown with the large means at high temperatures. But here, because the number of steps are not nearly enough for 500 cities, very little progress is made. That combined with the lack of direction (i.e., high temperature), results in a somewhat constant distance, which then leads to a small standard deviation.





3 T-Dependence

In every plot seen, a lower temperature resulted in a shorter distance. Running the algorithm with a lower temperature makes it more greedy. In a sense, it encourages the algorithm to go for the shorter distances. When the temperature is higher, this push for a short distance is made a lot less, thus resulting in a longer distance. Apparently, local minima do not seem to be an issue even with very small values for the temperature.

4 Work done

The MatLab code was written by Sander and refactored by Wessel. This code was then used to generate several plots, as done by Wessel. Sander wrote the text for the document, with verbal input and suggestions from Wessel.