**TSP – Protokoll**

**Enumeration:**

Über eine Backtrack-Rekursion umgesetzt. Aufwand O(n!), wobei unser Algorithmus einen effektiven Aufwand von (n-1)! hat. Das Optimum wäre (n-1)!/2, da es zu jeder Rundreise in die eine Richtung auch eine in die andere gibt, die nicht bestimmt werden müsste.

**Nearest Neighbour:**

Hat für einen bestimmten Startpunkt den Aufwand O(n²), falls man alle n Punkte betrachten möchte natürlich O(n³).

**Enumeration versus Nearest Neighbour**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **E Time** | **E Distance** | **N Time** | **N Distance** | **% Time ∆** | **% Distance ∆** |
| **6** | **0.0156s** | **25.418** | **0.0312s** | **25.418** | **+100%** | **+0%** |
| **8** | **0.0156s** | **142.888** | **0.0312s** | **142.888** | **+100%** | **+0%** |
| **10** | **0.0468s** | **337.617** | **0.0312s** | **373.739** | **-33%** | **+10.6%** |
| **11** | **0.3432s** | **1075.11** | **0.0156s** | **1075.11** | **-95%** | **+0%** |
| **12** | **3.6816s** | **2712.41** | **0.0468s** | **3148.01** | **-99%** | **+16%** |
| **13** | **46.613s** | **29.807** | **0.0468s** | **30.904** | **-99.9%** | **+3.7%** |

Man sieht eindeutig den Nachteil der Enumeration: Der faktorielle Aufwand benötigt viel zu lange Rechenzeit für größere Werte für n (wobei 13 eigentlich recht niedrig ist). Außerdem liefert das *Nearest Neighbour* Verfahren zwar sehr schnell eine akzeptable Lösung, achtet aber überhaupt nicht auf die optimal Lösung.

Schlussendlich sind beide Algorithmen nicht geeignet um das *Travelling Salesman Problem* zu lösen.

One sees clearly the disadvantage of enumeration: the factorial complexity takes too long for larger n-values. Furthermore *Nearest Neighbour* is very quickly but does not care about the perfect solution.

In the end both algorithms are not useful for solving the *Travelling Salesman Problem.*