Travelling Salesman Investigation

Zoe Wall
40182161@napier.ac.uk
Edinburgh Napier University - Algorithms and Data Structures (SET09117)

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords – Algorithms, Travelling Salesman, Nearest Neighbour, Two-Opt

1 Introduction

Problem summary including limitations of your sol.

2 Method

Description of the experiment conducted Intances to be examined how did you

3 Results

Tables and charts to show performance of sol. demonstration that sol is valid demonstration of quality of sol.

4 Conclusions

Summary of results Reflection on performance of your assessment

5 Appendix

source code

Referencing You should cite References like this: [1]. The references are saved in an external .bib file, and will automatically be added ot the bibliography at the end once cited.

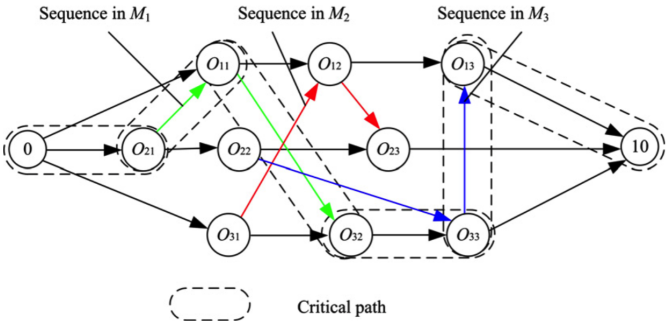

Figure 1: **ImageTitle** - Some Descriptive Text

6 Formatting

Some common formatting you may need uses these commands for **Bold Text**, *Italics*, and underlined.

6.1 LineBreaks

Here is a line

Here is a line followed by a double line break. This line is only one line break down from the above, Notice that latex can ignore this

We can force a break
with the break operator.

6.2 Maths

Embedding Maths is Latex's bread and butter

$$J = \left[\frac{\delta e}{\delta \theta_0} \frac{\delta e}{\delta \theta_1} \frac{\delta e}{\delta \theta_2}\right] = e_{current} - e_{target}$$

6.3 Code Listing

You can load segments of code from a file, or embed them directly.

Listing 1: Hello World! in c++

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!" << std::endl;
5     std::cin.get();
6     return 0;
7 }
```

6.4 PseudoCode

```
for i = 0 to 100 do
    print_number = true;
    if i is divisible by 3 then
        print "Fizz";
        print_number = false;
    end
    if i is divisible by 5 then
        print "Buzz";
        print_number = false;
    end
    if print_number then
        print i;
    end
    print a newline;
end
```

Algorithm 1: FizzBuzz

7 Conclusion

References

[1] S. Keshav, “How to read a paper,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 83–84, July 2007.