CSM6120 Essay: AI Methods for Finding Degree-Constraint Minimum Spanning Trees

Morgan Jones ([mwj7@aber.ac.uk](mailto:mwj7@aber.ac.uk))

# Introduction

This essay presents and compares current AI approaches to solving the Degree-Constrained Minimum Spanning Tree (DCMST) problem. I begin by introducing the problem definition and then proceed to present and compare various AI algorithms that solve it. Finally, I conclude my findings and state, along with motivation, my chosen approach to tackling the problem.

I used Google Scholar to find recent cited articles that define solutions to the DCMST. I read some of these articles and compared the background sections of the papers to piece together a sort of history/timeline of DCMST solutions. I searched for an API for solving DCMST instances and found a GitHub project that referenced papers that solve the problem in different ways, some of which I had already read. I chose the most recent and referenced approaches to compare; one from each type of AI approach. {Simulated Annealing, Genetic Algorithms & Ant Colony Optimization}

# The Degree-Constrained Minimum Spanning Tree Problem

The goal of the DCMST problem [1] is to find a Minimum Spanning Tree (MST) [2] of an asymmetric, unweighted and complete graph; such that the MST does not have a degree d 2 on any of its vertices. The degree of a vertex is the number edges attached to the vertex (incident edges). The problem is NP-Hard as shown by Garey and Johnson (1979) [3] through a reduction to an equivalent symmetric TSP.

Despite being difficult the DCSMT is a problem worth studying because many real life applications require a connected network that is subject to a degree-constraint. <Examples here..>

There have been multiple exact algorithms for solving the problem: branch & bound, Lagrangean Relaxation.

The problem is defined by the following objective function and constraints:

|  |  |  |
| --- | --- | --- |
| Minimise |  | Minimise the sum of edge weights included in the spanning tree |
|  |  | The number of edges on each node is no more than the degree. |
|  |  | The number of edges on each node is no less than one. |
|  |  | The number of edges in the MST is no more than one less than the number of vertices in the MST. |

Notation:

* i,j = Nodes in the graph.
* Bi = Number of incident edges on a vertex
* Cij = The weight of the edge connecting i and j.
* Xij = 1 if the edge eij is included in the MST, 0 otherwise
* V = Set of all vertices in the graph
* N = Set of vertices included in the MST

# AI Approaches for Computing Degree-Constrained Minimum Spanning Trees

## Ant-Based Optimisation

TODO

## Evolutionary Algorithms

TODO

## Simulated Annealing

TODO

# References

|  |  |
| --- | --- |
| [1] | S. C. Narula and C. A. Ho, “Degree-Constrained Minimum Spanning Tree,” *Computers and Operations Research,* vol. 7, pp. 239-249, 1980. |
| [2] | E. Horowitz and S. Sahni, “Minimum spanning trees,” in *Fundamentals of Computer Algorithms*, Potomac, Md, Computer Science Press, 1978, pp. 174-183. |
| [3] | M. Garey and D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, San Francisco, CA: Freeman, 1979. |