Traveling Salesman Problem

Walter Gress

This is an attempt to solving the NP-Complete problem: The Traveling Salesman Problem. The solution, from what I could find, is O(n4) worst case scenario, well below exponential time.

To start we begin by defining a function for flow through the graph (network). If there is a perfect flow, where all the edges needed to provide a circuit are covered, the flow function will equal 0.

The algorithm uses multiple agents, each attached to a vertex and emitting its own flow. Because we use an agent for each vertex, we cover all incident edges.

The flow equation is as follows:

where a is the number of agents, |V| is the number of vertices, |E| is the number of edges, 1/a is a normalizing factor across the agents. is the flow for the previous edge. is the resistance (distance in the TSP problem) for the ith edge.

The algorithm is:

For every agent

Let V = Set of all Vertices

Let P = Solution Set

Let Z = Set of Edges

Let = Set of working Edges

While (f(Z) != 0)

Let e = A.pop() //get one of the edges from the collection

Z.add(e) //add edge to test set

If (f(Z) == 0) //if flow through the set of edges is perfect

//use this as a solution

P.add(Z)

The problem with this solution, despite any others, is the issue of the initial value for flow.

The algorithm as far as I can tell is O(n4). n2 for the processing in the algorithm, and n2 for the flow function integrals totaling O(n4)