Convex Hull Problem and P=NP

Walter B. Gress V

Drexel University

Friday, June 7 2013

This concept of operations details a “Divide and Defend” Firewall. Recently there has been a surge of malware attacks, those attacks typically being orchestrated through botnets. Modern enterprise and personal firewalls are not designed by default to handle massive number of attacks at once like botnets can incur. D&D is designed with that very thought in mind.

# introduction

This paper presents an exercise in the determination of whether P=NP by use of the Convex Hull Problem and Turing Machines. We will begin by defining briefly a Turing Machine, followed by the complexity problem and finally the Convex Hull Problem. We will finish with a sketch of a proof that may lead towards a verification of Convex Hull being P.

# The convex hull problem

The convex hull problem is thus: In mathematics the convex hull of a set of X points in the Euclidean Plane or Space (we will refer to Euclidean Plane for the matter of this paper) For example, when X is in a bounded subset of a plane, the convex hull over those points can be represented to be like a tight rubberband around the outermost points. Convex is described as such:

A set S is convex if whenever two points P and Q are inside S, then the whole line segment PQ is also in S. The definition for a convex hull being

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The convex hull of a finite point set S = {P} is the smallest 2D convex polygon  (or polyhedron in 3D) that contains S. That is, there is no other convex polygon (or polyhedron)  with . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

There are seveal hull algorithms. One such isd Graham’s

(list more examples here)

# VISUAL CONVEX HULL PROBLEM

I am proposing a new sort of algorithm for dealing with the problem of the Convex Hull.

Fig 1.0

Given a collection of points (Fig 1.0) a convex hull can be mathematically or programmatically challenging to find. However, I believe it can be found almost instantly for a large number of points if this problem was given to a human, or, an algorithm that uses visual algorithms to mimic human vision. However, suppose we use the visual approach to an extant. So, instead of going around the edge of the convex hull, we go through it. For every y CONVEX, where CONVEX is the set of points in the Euclidean plane significant to the problem, we create horizontal scanline, capturing the x position.

# TURING MACHINES

# turing machines and scan lines

Let us assume each scan-line is a tape in a Turing Machine. So for every scan line there is a tape. Let us mark the beginning and ending of each tape as stop-reverse points. Let us mark the tape also with its x coordinate. Now let us place each Turing Machine Tape at location 0. Now all of the Turing Machine’s tapes will, simultenously, move forward and backward, on and off of their marker until the correct configuration is found

The question is, how does it know its in the right configuration? Does it know when all of the tapes are locked on their x values? Is there some intrinsic propery? There must, otherwise the multiple taps wouldn’t be reducible to one.

# COMPLEXITY

I am not certain of the Complexity. Each individual tape runs at no greater than O(n/2). However there is multiple tapes.

On the other hand, it has been shown that multiple tape Turing Machines are equivalent to a single Machine. If that is true, then theoretically we can combine the multiple tapes into one for a complexity of O(n/2), making the Convex Hull Problem an element of P.