# an algorithm for Erdős & Szekeres problem

Walter B. Gress V

Drexel University

Friday, May 27, 2016

The algorithm proposed can take convex graph and return a quadrilateral from it, should one exist.

# introduction

This paper quickly glosses over a method of algorithmically solving the Erdős & Szekeres problem with a divide and conquer recursive solution to the problem of deciding if a given graph has a quadrilateral and will actually, if applied to all vertices, find all the quadrilaterals in the graph.

# PROBLEM

Erdős & Szekeres problem can be stated as such: For five points that lie on the plane and which no three lie on the same straight line, it is always possible to find four points determining a convex quadrilateral.

We did not calculate the problem directly but devised an algorithm for any set of points on a graph, can a quadrilateral be found within them?

# ALGORITHMIC CONCEPT

The algorithm proceeds as so. Given a discrete graph, non-rooted, embed that graph into For any four points of the graph, if the area is a quadrilateral, the algorithm halts. If it does not find a square and is not a semicircle or triangle, divide the figure made by the four points so that each edge has a midpoint. Draw edges between those midpoints, dividing it into four cells. Now evaluate those four cells. If any are semicircles or triangles, the group of vetices are rejected. Continue dividing the quadrilateral cells by drawing edges between midpoints and rejecting semicircles and triangles. Continue this process until an area with a square root exists or a close approximation to a square root since we are in the reals.

# ALGORITHM IN PSEUDOCODE

Let G = {V, E}

Let v0, v1, v2, v3 V

Let Q V the current clique of vertices

Let A be the area given by the graph Q

Let q() be the decision of whether an area is a quadrilateral

Let s() be the decision of whether an area is a square

Let c() be the decision of whether an area is a circle

Let t() be the decision of whether an area is a triangle

Let d() be the division into midpoints

operation that generates the subgraphs Dn

Let m0, m1, m2, m3 be midpoints of the current graph Q

Let **this** be the current graph

Base:

If s(Q) = TRUE, return v0 to v3

If c(Q) or t(Q) = TRUE, return FAIL

D = d(v0 to v3)

Recursion(D0)

Recursion(D1)

Recursion(D2)

Recursion(D3)

Recursion:

Let Q = D

v0 to v3 D

If s(Q) = TRUE, return v0 to v3

s(), c(), t()

if q(Q) = TRUE:

d()

Recursion(this)

# PROOF AND TIME COMPLEXITY:

Base Case:

Given a five pointed graph, prove that there

is a four pointed quadrilateral in the graph.

For any given four point graph, an addition of another vertex does not change the original graph, and there is thus a vacuous case where the quadrilateral is nestled in the five pointed graph.

Inductive Case:

For n vertices, where n > 5, a quadrilateral exists for four points if: there does not exist a circle or semi circle, and if there is not a triangle. For every five points a quadrilateral exists iff for the kth iteration of an nxn figure (quadrilateral) its subdivision d() results in either a square or number of sides approximately close to a square, that is, as k.

Where a subdivision d() consists of finding the midpoint of all the edges of the quadrilateral and inscribing an edge between the midpoints as to divide it equally horizontally and vertically.