Mutated Near Optimal Vertex Cover Algorithm (NOVCA) Visualization on Tile Display
Sanjaya Gajurel and Roger Bielefeld
High Performance Computing (HPC), Advanced Research Computing (ARC)
IEEE Cluster 2015, Sep. 8-11, Chicago, IL, USA

ABSTRACT
This paper describes the mutated version of extremely fast polynomial time algorithm: NOVCA (Near Optimal Vertex Cover Algorithm). NOVCA is based on the idea of including the vertex having higher degree in the cover. Mutated is introduced in NOVCA by randomly selecting any remaining vertex having degree greater than 1 is the cover as an exception.

INTRODUCTION

The Vertex Cover (VC) of a graph G(V,E) with vertex set V and edge set E is a subset of vertices C of V (C ⊆ V) such that every edge of G has at least one endpoint in C. In 1972, Richard Karp showed that identification of minimal VC in a graph is an NP-complete problem.

MUTATED NEAR OPTIMAL VERTEX COVER ALGORITHM (MNOVCA)

NOVCA [2][3][4] (Fig. 1) is based on the concept that vertex cover candidates are those that are adjacent to the minimum degree vertex. It prevents the minimum degree vertex from being included in the cover. In case of a tie in a minimum degree vertex, the one having higher sum of the degrees of its adjacent vertices is chosen.

In MNOVCA, mutation is introduced by randomly including any of the remaining vertices not in the cover, having degree at least greater than one. NOVCA always returns minimum cover for all sets of random graphs including the Benchmark random graphs. MNOVCA tackles the family of benchmark graphs [5] having minimum cover consisting of lower degree vertices that defeats NOVCA’s fundamental heuristics of including vertices of higher degree in the cover.

IMPLEMENTATION AT CWRU HPC & VIZ-WALL

The NOVCA algorithm is rewritten in VTK/Cxx [6], implemented in CWRU High Performance Computing (HPC) Cluster, and visualized on the CWRU Viz-Wall.

In the VTK/Cxx implementation, the output of the code is dumped as VTK files in VTK unstructured (.vtu) format, where points and cells are defined by the vertices and their respective connection to form the graph. The ParaView engine, then produces high resolution animation in the tile display (Fig. 2, Fig. 3 and Fig. 4 depict high resolution ParaView screenshots of 2D and 3D animations respectively).

CONCLUSIONS

The ability to visualize combinatorial optimization problems such as NOVCA on a large display (Fig. 2, Fig. 3 and Fig. 4) makes researchers to observe and analyze their behavior at different stages of the algorithm. NOVCA, which always returns an optimal value for small benchmark graphs, sometimes produces suboptimal results on larger benchmark graphs.

ACKNOWLEDGEMENT

This work made use of the High Performance Computing Resource in the Case Facility for Advanced Research Computing at Case Western Reserve University.

REFERENCES