Mutated Near Optimal Vertex Cover Algorithm (NOVCA) Visualization on a 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 minimal degree in the cover. In case of a tie in a minimum degree vertex, an additional vertex having higher sum of degree of adjacent vertices is chosen from the remaining candidates. In NOVCA+, mutation is introduced by randomly selecting any remaining vertex having degree greater than or equal to the minimum degree vertex. NOVCA is based on the concept that vertex cover candidates are those that are adjacent to the minimum degree vertex. If a vertex v is selected in the cover, then 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 NOVCA+, mutation is introduced by randomly selecting any remaining vertex having degree greater than or equal to the minimum degree vertex. NOVCA is based on the concept that vertex cover candidates are those that are adjacent to the minimum degree vertex. If a vertex v is selected in the cover, then 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 NOVCA+, mutation is introduced by randomly selecting any remaining vertex having degree greater than or equal to the minimum degree vertex. NOVCA is based on the concept that vertex cover candidates are those that are adjacent to the minimum degree vertex. If a vertex v is selected in the cover, then 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.

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 (VC ⊆ V) such that every edge of G has at least one endpoint in C. In 1972, Richard [1] showed that identification of minimal VC is a graph as an NP-complete problem.

MUTATED NEAR OPTIMAL VERTEX COVER ALGORITHM (NOVCA+)
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 NOVCA+, mutation is introduced by randomly selecting any remaining vertex having degree greater than or equal to the minimum degree vertex. NOVCA+ (Near Optimal Vertex Cover Algorithm) always returns optimal value for all sorts of graphs including the Benchmark random graphs. NOVCA always returns minimum cover for all sorts of random graphs including the Benchmark random graphs. MNOVCA, which always returns optimal value for small benchmark graphs, produces suboptimal results on some larger benchmark graphs.

IMPLEMENTATION @ CWRU HPC & VIZ-WALL
The NOVCA algorithm is re-written in VTK/Cxx [6], implemented in the 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 unstructured (.vtu) format, where points and cells represent vertices and edges of the graph respectively. ParaView [7] that uses VTK under the hood as the data processing and rendering engine, then produces high resolution animation in a tile display (Fig. 2). Fig. 3 and Fig. 4 depict high resolution ParaView screenshots of 2D and 3D animations respectively.

CONCLUSIONS
The ability to visual datasets from combinatorial optimization problems such as Vertex Cover on a large display (Fig. 2) helps researchers to observe and analyze their behavior at different stages of the algorithm. NOVCA, which always returns optimal value for small benchmark graphs, produces suboptimal results on some larger benchmark graphs. The animated visualization permitted detection of patterns of failure on these graphs, which would have been nearly impossible to achieve using static renderings. Moreover, because the small displays can quickly get cluttered when size of the graph increases, there is a great benefit to using a larger high resolution tile display engine, then produces high resolution animation in a tile display (Fig. 2). Fig. 3 and Fig. 4 depict high resolution ParaView screenshots of 2D and 3D animations respectively.

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

REFERENCES