GRAPH THEORETIC SOLUTION TO MULTIDIMENSIONAL OPTIMIZATION (A-TREE)

Swaroopa V Dugani*, Dr.Sunanda Dixit and Mahesh B V*****

*Dayanda Sagar College of Engineering /ISE, Banglore, India
E-mail:swroopadugani83@gmail.com
**Dayanda Sagar College of Engineering /ISE, Banglore, India
E-mail:sunanda.bms@gmail.com
***Dayanda Sagar College of Engineering /ISE, Banglore, India
E-mail: mahesh_ait@hotmail.com

ABSTRACT: Representation and traversal of rectangular area graphs is a problem well known in Electronic design Automation (EDA).A new approach to the problem is to use Area trees or A-tree that have quadruple network of area pointers that can be used to traverse area in any of the four planar directions. The work done in this research enumerates the problems faced while performing basic operations on rectangular areas .We have implemented bisect, sort, merge and insert operations in A-tree.

KEYWORDS: Electronic design Automation, Geographical Information Systems, R-tree, Quad Trees, A tree.

INTRODUCTION

EDA includes methodologies, algorithms and tools that assist and automate the design, verification, and testing of electronic systems. It is a general methodology for refining a high-level description down to a detailed physical implementation that are ranging from Integrated circuits, printed circuit boards (PCBs) and Electronic systems.
Current digital drifts are extremely modular. Analog EDA tools are less modular, since it requires many functions, they are strongly coupled, and the components are less ideal. Use of EDA tools in electronics has rapidly increased because of their continuous growth in semiconductor technology. Area representation is a basic requirement in many engineering problems. Thus there is a need to find best ways of representing areas in terms of storage efficiency, speed of search, and functional completeness.

AREA REPRESENTATION

Area representation is a basic requirement in many engineering problems. Geographical Information Systems (GIS) and mapping problems need 2D and 3D data structures to exhibit geospatial data. Mechanical engineers will stand in need of surface optimization to determine pattern nesting problems. Electronic engineers require area representation while designing integrated circuits. Thus there is a need to find best ways of representing areas in terms of storage efficiency, speed of search, and functional completeness.

Area Optimization

Area optimization is done using A-tree which performs different operations like insert, bisect, merge, sort. Optimizing for minimum area allows the designer to use fewer resources, and also allows the sections of the design to be tightly bound and interconnected [8]. This leads to shorter interconnect distances, less routing resources to be used, brisk end-to-end signal paths, and even swift and more consistent place and route times. Figure 1 shows the free space optimization process

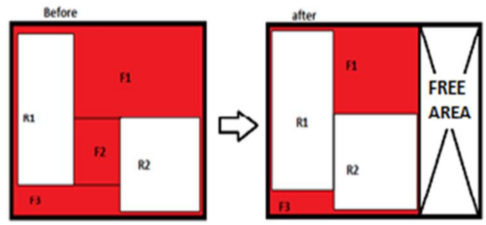

Figure 1. Area optimization process

Quad Tree
A quad tree is a tree data structure in which each internal node has four children [12]. Quad trees are used to partition a two dimensional space by recursively subdividing it into four partitions or quadrants. The resultant regions may be squares ir rectangle, or may have arbitrary shapes.It has some common countenance: They partition area into adaptable cells, Each cell has a maximum capacity.

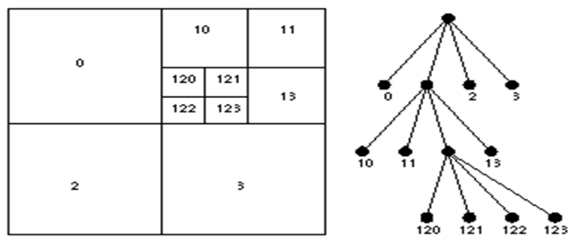

Figure 2. Structure of Quad Tree

R-Tree
These are tree data structures used for dimensional access methods [6] i.e., for indexing multi-dimensional information such polygons, rectangles or geographical coordinates. A common usage for an R-tree might be to store spatial objects such as polygons and joint objects that typical maps are built up of: streets, buildings, outlines of lakes, coastlines, etc [9]. Figure 3 shows the structure of R-Tree.

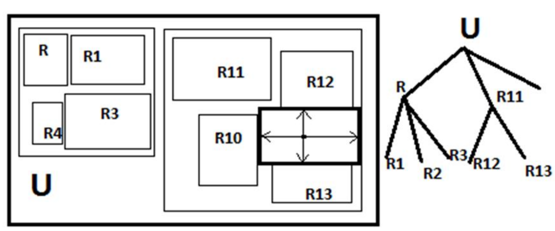

Figure 3. Structure of R-Tree

B tree
These are tree data structures called as Balanced Search Tree. We use insertion feature of this tree where all insertions start at a leaf node. To add a new element, search for a leaf node in the tree where the new element should be added.

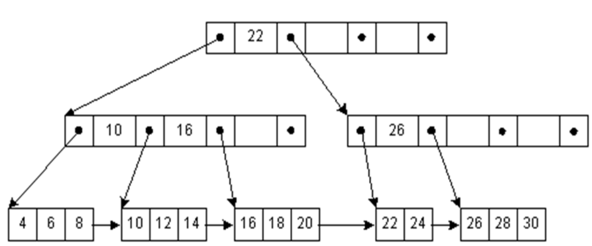

Figure 4. Structure of B Tree

Constraint Graphs

These are used to find neighbour nodes or elements and to search horizontal and vertical constraints [7]. Figure 5 shows constraint graph. With this one can determine the distance between two nodes that may be top, bottom, left and right nodes and their relationship and can create graph to show the best possible ways to shift the elements in accordance with their relation and operation to optimize the area [10].

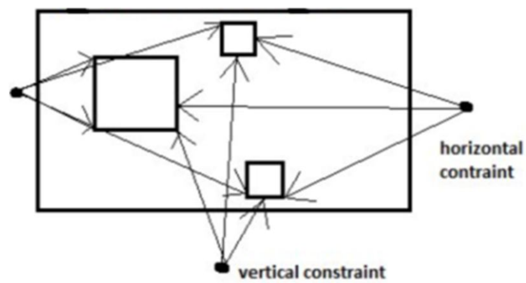

Figure 5. Constraint graph

PROPOSED DATA STRUCTURE A-TREE

A-Tree also known as Area trees can be used to traverse area in all four directions. The x and y co-ordinates of each area is represented by origin and its size (height and width) is represented by range. Each sub-tree has four neighbours top, bottom, left and right. Initially all the neighbours must be null. As and when the operations are performed on it the neighbours are adjusted. The data structure gets updated as and when operations are performed on the components. Non slicing geometry which is difficult to obtain from other tool sets based on B-trees, O-trees etc. can be obtained these primitive operations. While designing integrated circuits, we use this data structure to develop a toolset for area representation.

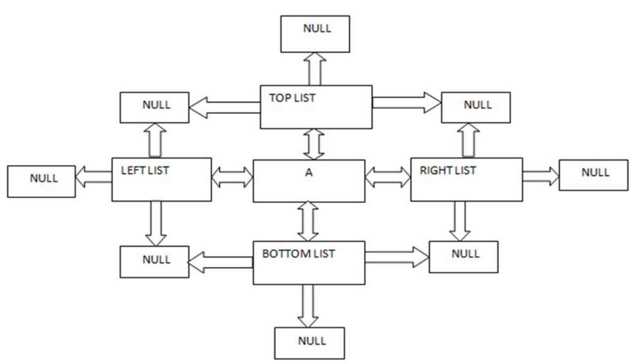

Figure 6. Structure of A-tree

Work Flow

Original Area (OA) is total available area to insert components. When new element is inserted, value of original area changes that is calculated by following relation. RA=OA – Area of Inserted Component.
Value of Original Area (OA) will be renewed after calculating the Remaining Area(RA) i. e original area will be made equal to the value of remaining area. RA=OA. Insert allows to dynamically inserting the component and splitting of rectangular area. Once the rectangle is inserted the original rectangle is spitted into three rectangles (Old.a, Old.b and Old.c) where each rectangle is considered as one node. Create graph creates the graph and shows pointers to four neighbours of every node (left, right, top bottom). Figure 8 shows inserted rectangle, spitted regions and graph.
Bisect Divides the selected rectangle in two equal parts either horizontally or vertically as per the radio buttons selected. Figure 9 shows how rectangles are selected to merge area to get wider space to add some more components. Common areas between selected rectangles are merged. Figure 9 shows result of optimization, old.c.a is a large space where we can insert more componenets.

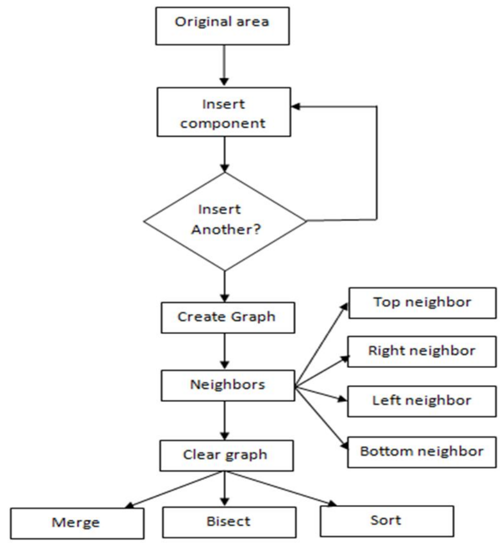

Figure 7. Work Flow

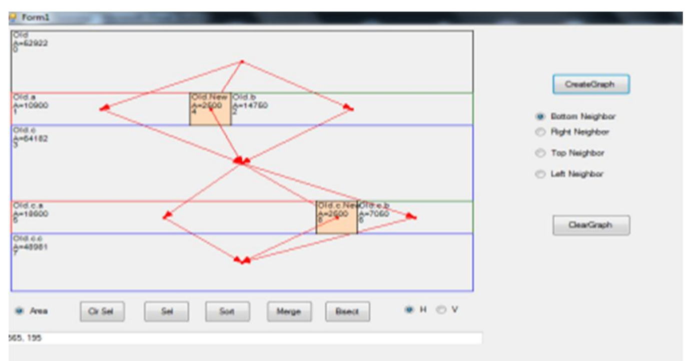

Figure 8. Output Screen shot

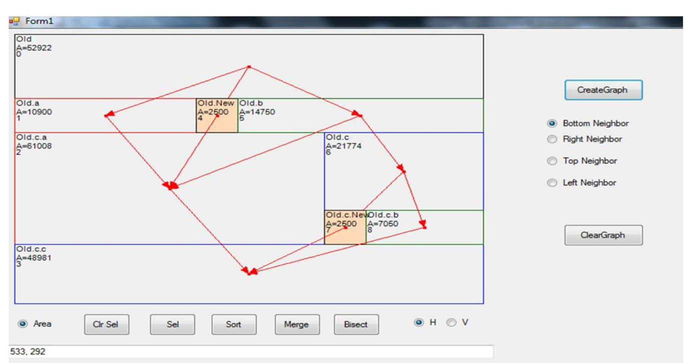

Figure 9. Output for Optimized Area

CONCLUSION

We have implemented a generalized graph structure called A-Tree. A node in A-Tree is rectangular area connectivity between four intermediate neighbours from edge. The graph can handle rectangles with operations like insert, merge, bisect, sort etc. Hence we optimize the area by shifting elements in such a way that we get more places to add more components. These structures used in many Electronic design automation tools.

REFERENCES

[1] Programming C# -2nd Edition by Jesse Liberty
[2] C# by Andrew Troelsen
[3] en.wikipedia.org/wiki/Electronic_design_automation
[4] en.wikipedia.org/wiki/Quadtree
[5] Ze-bao Zhang, Jian-pei Zhang, Ruo-yu Li, Jing Yang, Harbin Engineering University, Harbin, China. A R-tree-based Fine Directional Query Filtering
[6] Ravik. Kothuri, Siva Ravada. Efficient Processing of Large Spatial Queries Using Interior Approximation. International Symposium on Advances in Spatial and Temporal Database.2001.pp404-421.
[7] L. Arge, K. Hinrichs, J. Vahrenhold and J. Vitter, “Efficient Bulk Operations on Dynamic R-tress,” in Proc. ALENEX,1999, pp. 328–348.
[8] N. Beckmann, H.-P. Kriegel, R.“The R-tree: an Efficient and Robust Access Method for Points and Rectangles,” SIGMOD, 1990, pp. 322–331.
[9] S. Berchtold, C. B ohm and H.-P. Kriegel, “Improving the Query performance of High Dimensional Index Structures by Bulk Load Operations,” in Proc. EDBT, 1998, pp. 216–230.
[10] J. Bercken, P. Widmayer and B. Seeger, “A Generic Ap-proach to Bulk Loading Multidimensional Index Structures,” inProc. VLDB, 1997, pp. 406–415.
[11] C. B ohm and H.-P. Kriegel, “Efficient Bulk Loading of Large High Dimensional Indexes,” in Proc. DaWaK, 1999, pp.251–260.
[12] Graphics Interchange Format(sm) Version 89a (c) 1987, 1988, 1989, 1990, Copyright CompuServe Incorporated Columbus, Ohio, 31 July 1990.
[13] Mark Nelson, “LZW data compression,” Dr. Dobb’s Journal, Oct. 1989.
[14] ISO/IEC 15948:2003 (E), Portable Network Graphics (PNG) Specification (Second Edition) Functional specification, November 2003.
[15] ISO/IEC 15444-1:2004, Information technology -- JPEG 2000 image coding system Part 1: Code coding system, Sept. 2004.