Mathematical Techniques Applied to Knowledge Based Engineering Design Systems

C. Van der Velden* C. Bil* X. Yu† A. Smith‡

31 March 2005

Abstract

A number of mathematical techniques used in automation of routing in different fields are presented and discussed for application to a Knowledge Based Engineering (KBE) design system. The connection of multiple terminals within a limited space with obstacles is a commonly encountered problem in numerous fields including computer microchip and printed circuit board routing, artificial intelligence , and GPS navigation systems. Algorithms used in these applications can be of several different types such as maze, channel switchbox and line search routers. They also use different strategies including best-first, depth-first and breadth-first searches, as well as applying penalties for turns or through vias. Also presented is a proposed configuration for a KBE software system for three dimensional routing design in aerospace vehicles. Such a system will have applications in the design of electrical wiring and hydraulic and pneumatic pipes on aircraft.

Contents

*School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia.

†School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia.

‡GKN Aerospace Engineering Services Pty Ltd, Brisbane, Australia.

1 Introduction

Knowledge Based Engineering (KBE) is a methodology of enhancing engineering design processes through effective knowledge management. KBE systems are software applications which collect, store and utilise engineering rules and knowledge and are used for design automation and verification, automated updating, and integration of design and production knowledge. Usage of KBE tools facilitates the practice of concurrent engineering allowing work to be carried out effectively over great distances and time-zones. Additionally, effective integration of design rules, three dimensional modelling and designer specific knowledge into a KBE software system can lead to greater product integrity, shorter development time and reduced cost. KBE tools are especially suited to applications involving repetitive design processes or rule checking, and designs which change a number of times before a final design is reached.

KBE systems employ various mathematical processes together with process and concept knowledge collected from experts and interpreted into "IF... THEN..." rules, driven by an inference engine which performs the reasoning process and makes decisions. KBE systems are applied across many problem domains in various industries such as aerospace, automotive, maritime engineering and manufacturing.

The problem of connecting multiple terminals within a set space with obstacles is commonly encountered in numerous fields ranging from electronics and micro-electronics, information flow, navigation systems, artificial intelligence. The following lists some common routing problems:
- Very Large Scale Integrated circuits (VLSI circuits) and Printed Circuit

Boards (PCBs)
- Electrical wiring through structures
- Pipe design including hydraulic, pneumatic, air ducting, etc.
- Artificial Intelligence (AI) for robots and computer game navigation
- Information packet flow in computer networks and over the internet
- GPS navigation systems

Practices in VLSI circuit routing automation can provide a good starting point for the problem of electrical loom and pipe design in aerospace vehicles. The VLSI routing problem is considered to be NP-complete which is one of the most difficult problems to find optimal solutions. Microprocessors contain hundreds of millions of logic components which must be interconnected within a limited space. To achieve the interconnection on such a large scale, autonomous systems are utilised employing algorithms and heuristics which generally find "good" solutions but not guaranteed to output the most efficient and fastest solution. Because non-optimal techniques are used, routing quality is measured by several metrics including: area taken by the routed nets, routing completion rate, length of interconnection wires, number of vias, and time taken to execute the routing run.

This paper presents an overview of some common search and routing algorithms and heuristics used in routing automation, and a description of a proposed Knowledge Based Engineering routing system for design of electrical wiring and pipe design for aerospace vehicles.

2 Routing Design Process and Algorithms

In general once the physical component placement for a system has been defined and the requirements given (usually in the form of a netlist), the routing process consists of four main steps:

1) **Definition of regions to be routed:** allows the problem to be split into a number of smaller routing problems.

2) **Global routing:** planning phase which reads the position of terminals and fixed objects and gives instructions for the detailed router to maximise completion rate and minimise path length, especially for critical nets.

3) **Order in which regions are routed:** determines ordering of such that congestion will be avoided.

4) **Detailed routing:** determines the exact path taken by wires including layers used for each segment and vias connecting between layers.

There are numerous algorithms for a variety of routing applications in the global and detailed stages in the routing process as well as for specialised applications. Figure 1 (from [1]) shows a family of routing algorithm types

and their use in different phases in the routing process. The list is by no means exhaustive, but covers many of the common routers which serve as a basis for more specific and complex algorithms. Several algorithms from this list will be discussed in the following pages including graph/tree searching, maze routing, channel and switchbox routing, line probes/searches, and the A* (A star) algorithm used in AI navigation.

Figure 1: Family of routers. (Adapted from [1])

2.1 Searches

Searches are used in algorithmic processes to find the sequences to go from an initial state to some goal state. In terms of path finding, an analysis of a search space and some method of choosing the best path is required. The most basic path finding algorithms analyse the search space in all directions until the target is found. While simple to implement, this form of searching has a high processing cost (in terms of processing time and storage of information). A common method of structuring data within a software system uses graph or tree structures which provides hierarchy information. There are a number of strategies which can be used to seek out targets more effectively depending on the nature of the problem, the size of the data tree, the number of solutions available and the location of the solution(s) within the tree. Some common search techniques include breadth-first, depth-first, best-first, and greedy searches. These are discussed below:

Breadth-first search is a systematic method of searching the nodes on a tree or graph. The search algorithm starts from the root and expands the child nodes, and continues to expand all the subsequent child nodes on the same level. This method will find shallowest solution in a search tree.

Depth-first search starts from the root and expands the first node and then its first child node and so on. When the bottom of the tree is reached, the next child in the previous level will be expanded. This method will find deep solutions more quickly than breadth-first, however some shallow grid points will not be visited early in the search. Depth first searches can have a lower memory requirement especially if there are numerous children on each branch.

Best-first search is an extension of depth first search which uses a heuristic to expand the most promising nodes based on a particular rule.

Greedy searches use a heuristic that takes the most optimum path at each step to bring closer to the global optimum. Greedy searches are not

guaranteed to find the optimum solution in all cases, but often gives a good approximation.

2.2 Maze Router

The maze router was one of the first algorithms to be used in automated routing systems and is often called Lees algorithm after its inventor [2]. The maze router is a breadth-first, grid-based path finder that will always return the shortest path for a single net. The algorithm functions by propagating a wave from the source and/or target terminals over a grided search space and assigns a value to each cell depending on its distance from the source or target. A backtracking phase then determines the shortest path between the two terminals. The maze router can be used in the global and detailed routing phases.

The process is relatively simple and begins by searching the space for the source location, **S**, from which a search wave is propagated iteratively. For each search step number, **k**, all cells with a Manhattan distance (i.e. no diagonals) of **k** from **S** are labelled **k**. The process repeats increasing **k** by 1 for each iteration until the target cell, **T**, is found. When the target is reached, the algorithm traces a path back to the source by iteratively searching for cells with **k** **1**, until **S** is again reached. At this point the path is defined.

Figure 2: Example of maze routing. (Left) wave propagation from source only, (Right) wave propagation from source and target

Run time can be improved by propagating waves from both source and target, reducing number of steps, k, by approximately half and reducing the number of cells visited by the algorithm. Figure 2 shows an example of maze routing in a small grid with obstacles with waves expanded from both the source, S, and source and target, T.

The algorithm can be implemented on multiple layers, although algorithmic complexity increases exponentially. The maze algorithm can be shown to be efficient with a large number of obstacles. In its basic form, the maze router is very simple to implement and its power can be extended by applying constraints (eg. preferred path directions, turns penalties, via penalties, etc.). Modifications can be made to the algorithm reduce run time.

A number of limitations to the maze algorithm exist and have been well documented. The main difficulty encountered with this algorithm is its sensitivity to sequential routing of nets. Paths from already routed nets form

obstacles for unrouted nets. In addition to this, the algorithm is inefficient when routing more than two terminals in a single net. In the case of multi-terminal nets, the connection between two terminals is found, then the partially routed net is treated as the a source for remaining terminals. Also, the algorithm is inefficient when routing terminals in large empty spaces.

2.3 A* (A star)

The A* (or A star) algorithm is an example of a best-first search algorithm and is commonly used in Artificial Intelligence (AI) navigation in computer games. The algorithm uses a grided search area similar to Lees Maze algorithm, however the method for determining which cells differs in several ways.

The algorithm uses an "open list" to define all cells which to be considered in the search, and a closed list for those which are illegal. The algorithm works by examining all neighbouring nodes in a grided space and calculating a path score, **F**, based on a movement cost, **G**, from starting cell to given cell (calculated by taking cost of parent and adding movement cost to next cell), and an estimated cost to target, **H**, (determined using a heuristic eg. Manhattan cost).

Figure 3 shows an example of the A* algorithm for a simple path finding problem. The movement cost, G, estimated remaining cost, H, and path score, F, are shown for each cell.

Figure 3: Example of A* algorithm (Adapted from [3])

2.4 Channel router

Channel routing is a technique used for wiring rectangular regions called "channels" in VLSI circuits which have a number of pins running along the top and bottom. A net-list for the top and bottom of the channel with terminals labelled from 1 to N indicates which pins are to be connected. Terminals with the same number, i, where 1 = i = N are to be connected with a single net, while a terminal labelled 0 is unconnected. In most cases routing is carried out on two layers with vertical wire segments (branches) on one layer and horizontal segments (trunks) are on the other, with small electrical contacts called "vias" connecting the two layers when vertical and horizontal segments meet. This allows overlapping of wires from other nets to reduce the size of the channel.

Figure 4: (Above left) Unrouted channel with ordered nets, (Above right) Horizontal Constraint Graph, (Below left) Vertical Constraing Graph, (Below right) Solution of channel routing problem (Adapted from [4])

The goal of the channel routing process is to connect terminals of the same netlist number while minimising the number of horizontal tracks and thusly, number of vias. The order in which nets are routed is determined by Horizontal Constraint Graphs (HCG) and Vertical Constraint Graphs (VCG). The graphs analyse the net lists on the top and bottom of the channel and search for possible conflicts. Figure 4 shows a channel routing problem and its associated HCG, VCG and optimal solution to the channel. The channel width is minimised by placing multiple nets on the same track while considering vertical conflicts.

2.5 Switchbox router

Switchbox routing problems are similar to the channel router, except connections occur on all sides of the rectangle rather than the top and bottom as in Figure 5 (left). Despite the problem being notionally similar to channel routing, it is much harder to implement in an algorithm. Whereas in channel routing there are several metrics for measuring routing performance (path length, area of nets, number of tracks, etc.), in switchbox routing emphasis is placed on finding the existence of a solution. Figure 5 (right) shows an example of a routed switchbox.

Figure 5: (Left) Channels and Switchboxes, (Right) Example of switchbox routing

2.6 Line search

Line search or line probe routers are used for discovering paths in large grided search areas which can significantly reduce the number of nodes required to be searched compared to the classic maze routing method described above. The basic algorithm works by extending search lines from the source and target nodes, extending additional lines from "escape points" perpendicular to the initial search line until the lines intersect and the path defined. There are two main implementations of the line probe algorithm, Mikami-Tabuchi's Algorithm which generates many escape points per search line (Figure 6 left),

and Hightowers algorithm [5] which generates only one escape point per line (Figure 6 right).

Figure 6: (Left) Example of line search algorithm using Mikami-Tabuchi's Algorithm, (Right) Example of line search algorithm using Hightowers Algorithm

2.7 Other algorithms

The search and routing algorithms discussed above represent only a few of many router types. The maze and channel routers are two of the most common routers. There have been numerous variations to these algorithms to increase routing speed, speed reduce memory requirements, and extend their capability to other applications.

The specialised routers as indicated in Figure 1 will not be discussed except to say that power/ground routing involves finding minimum resistance paths in electrical networks, and clock routing involves routing nets such that clock delay is minimised, and delay synchronized for multiple nets.

3 Voxels

Current conventional CAD software is highly visual-based and uses complex mathematical equations to represent surfaces and other geometry, requiring high end computer graphics cards to display 3D models. An alternate method for displaying solid 3D models uses stacks of voxels (or volume pixels) which are 3D elements analogous to 2D pixels. Voxels can be stacked to produce complex 3D images of varying precision in the much the same way pixels are arranged to create 2D images. Because the connection between neighbouring voxels is based on fixed x,y,z coordinates, 3D hardware acceleration is not necessary for display. An example of voxel representation of a sphere is shown in Figure 7.

Figure 7: Voxel representation of a curved surface

Voxel technology is currently used in the medical industry for visualisation of scans of the human body including CT and CAT scans, and MRIs. Voxels are also used in height maps for display of terrain. In this case a 2D grid can be used with a height specified for each cell, reducing the memory requirement from a 3D array to 2D array with height parameter. Another

use emerging from the use of voxel displays is in development of training aids which can be deployed on handheld devices such as palm pilots.

In a computer environment, voxels are defined by a number of characteristics including size, reference position, state of the voxel (on or off), colour, density, etc.. One of the limitations on the use of voxels is the precision. Since there is no mathematical relationship between neighbouring voxels, details in a model such as curved surfaces, fillets chamfers, holes etc. must be simplified and approximated by solid cube elements. Precision of the model can be improved by increasing the number of voxels used and reducing their size, however, information storage and display requirements will increase.

The use of voxels to discretely model three dimensional geometry can provide a simple method of preparing geometry suitable to apply a grid based routing algorithm. Many voxel modelling software applets can be found on the internet. Some of these applets can read CAD geometry files and convert geometry to a voxel format.

4 KBE Router

The theory of various routing algorithms presented above will be applied to a KBE system for routing automation in aerospace vehicles. The objective of this research project is to extend the current body of the knowledge in the field of KBE by development of a general three dimensional router which will read 3D CAD geometry with source and target terminals and output 3D wire/pipe geometry and other information required to describe the system path. The system will maximise flexibility by allowing multiple rule sets for different routing applications to be implemented (for example wires / pipes / ducts etc.). Voxel technology will be used to discretise geometry for input into grid based algorithmic processes.

Figure 8: KBE router flow diagram

Electrical looms in aircraft consist of thousands of cables and are usually routed by hand using CAD workstations with engineers using personal knowledge and experience of how to route cables through the structure. This is a repetitive process and results will often vary between engineers. Also, as structural changes are made with further design iterations, often the entire system requires time-consuming rerouting, making it a prime candidate for application to a KBE system. The proposed system configuration is shown in Figure 8.

5 Conclusion

This paper has presented an overview of some of the algorithms and process commonly used in routing automation, and some of the principles of searching in knowledge based engineering systems. Also included was a description of voxel technology for representation of three dimensional geometry which can notionally be used to simplify and discretise complex CAD models for application to grid based algorithmic processes. This overview forms the initial stages in the development of a knowledge based router for aerospace routing applications such as electrical wiring and pipe design.

The next stages in the system development process include developing a code library for routing implementing, various methods described above, and determining the problem domains which are best suited to these various algorithms. The KBE system architecture will be developed using existing capabilities wherever possible with an emphasis on developing effective and novel methods of knowledge capture and representation. The knowledge and rule base will be developed for a limited number of selected routing applications while ensuring flexibility for expansion into other problem domains.

References

[1] M. Tehranipoor. *CAD Algorithms Routing*. Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 2005.

[2] C. Y. Lee. *An Algorithm for Path Connections and its Applications*. IRE Transactions on Electronic Computers vol. EC-10, no. 2, 1961.

[3] P. Lester. *A* Pathfinding for Beginners*. Almanac of Policy Issues website, 2005. URL: http://www.policyalmanac.org/games/aStarTutorial.htm

[4] P. Groeneveld. *Electronic Design Automation, Part 3*. Technische Universiteit Eindhoven, The Netherlands, 2005.

[5] D. W. Hightower. *A solution to line-routing problems on the continuous Plane*. Proceedings of the Sixth Annual Design Automation Workshop, 1969.

[6] M. Guruswamyl, and D. F. Wong. *A General Multi-layer Area Router*. The University of Texas at Austin, Department of Electrical and Computer Engineering, 1991.