S. Szykman
Manufacturing Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD

J. Cagan
Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA

Synthesis of Optimal Nonorthogonal Routes

This paper introduces a novel approach to three dimensional routing optimization. Examples of routing tasks for engineering applications include routing of pipes, wires and air ducts. Traditionally, routing algorithms perform Manhattan, or orthogonal, routing. Nonorthogonal routing can be less costly than Manhattan routing and for applications such as automotive or aerospace design, Manhattan routing is impractical due to spatial limitations. The research presented in this paper uses simulated annealing as the basis of a nonorthogonal routing optimization algorithm. Several examples comparing the two approaches are given.

1 Introduction

This research introduces a novel approach to three dimensional routing optimization that uses simulated annealing, a stochastic optimization technique. Routing problems are abundant in numerous engineering applications such as routing of pipes, wires and air ducts. Potential uses for a routing optimization tool include plant, building and machine layout, automobile and ship design as well as aerospace applications.

Traditionally, routing algorithms perform Manhattan routing, meaning that routing occurs along three orthogonal directions and that all bends are at 90 deg angles. For certain classes of problems, Manhattan routing may have advantages over nonorthogonal routing despite potentially higher routing costs. In some piping applications, for example, using Manhattan routes allows pipes to be grouped together for ease of mounting and maintenance, and to run parallel to walls to reduce intrusion of pipes into working spaces.

For problems where there is no specific need for Manhattan routing, this approach continues to be used for a number of reasons. One reason is that before the advent of CAD and optimization tools, routing was done manually by designers. Because a three dimensional space could only be represented in two dimensions on a drafting board, the easiest way to avoid obstacles was to use Manhattan routing where a path only moves along one coordinate direction at a time. Although more sophisticated CAD systems now exist, this vestigial characteristic persists in most routing algorithms. A second reason for the extensive use of Manhattan paths is that they are easier to generate. Because paths can only move in three orthogonal directions, search can be rapid since there are far fewer alternatives to explore than with nonorthogonal paths. Associated computation, such as interference testing, may also be simplified.

While Manhattan routing is well suited for certain types of problems, other applications that have historically utilized Manhattan routing could realize routing cost reductions or improvements in performance with nonorthogonal routing. Furthermore, for many mechanical engineering problems, such as automotive or aerospace applications, Manhattan routing is impractical due to spatial limitations and nonorthogonal routing becomes necessary (these issues are discussed in greater detail in Section 3). There is therefore a need for the development of computational approaches to nonorthogonal routing optimization.

The objectives a designer chooses to optimize will differ from one task to another. Examples of problem-specific design issues relevant to routing are: interference and crosstalk in wire routing, thermal considerations, pressure drops due to bends in piping, wall penetrations and undesirable routes such as routing water pipes above electronic equipment. This work omits problem-specific objectives and focuses on two objectives that are common to a majority of routing tasks: minimizing routing length and number of required bends. Using this general approach, other design considerations can be taken into account by incorporating additional problem-specific objectives in the optimization or post-processing the solution.

In order to describe this generic approach, we define three context-free terms that are used in the remainder of this paper. A *route* is the path between specified points produced by the routing algorithm. A *terminal* is the starting or ending point of a route. A route can represent a wire, a pipe, an air duct, etc., depending on the application. Similarly, a terminal might be an electronic terminal, a connector, a valve, a nozzle, and so on. An *obstacle* is an object that prevents a route from moving through a region; generally these are the components located between the routes. Note that a terminal may be located on or off the obstacle or component.

The remainder of the paper is organized as follows: Section 2 describes areas of related work and Section 3 discusses issues associated with Manhattan routing versus nonorthogonal routing. In Section 4 we describe the simulated annealing routing algorithm and Section 5 presents results of several problems solved using the routing algorithm.

2 Background and Related Work

In the electrical engineering domain there are many examples of two dimensional routing algorithms that have been developed for VLSI circuit layout tools (e.g., Sechen and Sangiovanni-Vincentelli, 1985; Ohtsuki, 1986; Shin and Sangiovanni-Vincentelli, 1987; Cohn et al., 1991). This body of literature consists exclusively of Manhattan routing algorithms.

Pipe routing is another common application for which numerous algorithms have been developed. Gunn and Al-Asadi (1987) have developed a CAD system for chemical plant layout that performs pipe routing. Mitsuta et al. (1986) present a rule-based approach to routing problems for industrial plants. The rule base contains knowledge relevant to routing (e.g. specifications on minimum clearance and wall penetrations) which is used to reduce the entire space into a set of rectangular spatial elements. A best-first maze running algorithm is then used to find the best path for each of the specified routes. Zhu and Latombe (1991) use an adaptation of the approximate cell decomposition method from the field of robotic motion planning to generate pipe routes. The technique for generating the actual path is similar to the above approach—free space is decomposed into a graph of connected cells and a best-first search is used to find the path.

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received May 1995; revised Feb. 1996. Associate Technical Editor: A. R. Parkinson.

The primary limitation of the approaches described above is that they perform Manhattan routing.[1] These approaches were developed to solve problems where Manhattan routing was either specifically required or found to be acceptable; however, they cannot be used to generate nonorthogonal routes for classes of problems where nonorthogonal routes are desirable due to reduced routing costs or necessary due to spatial limitations. Asano et al. (1987) developed an algorithm that runs in polynomial time for finding the guaranteed shortest (nonorthogonal) path between two dimensional polygons. While it may be possible to extend this work to three dimensional routing, this algorithm is limited by the fact that the shortest path is not necessarily the optimal path since length is usually not the sole objective. This and other shortest path algorithms use purely geometric information and do not consider other important routing design criteria or constraints.

Jain et al. (1992) have developed a knowledge-based system which is able to generate nonorthogonal pipe routes. A drawback to this approach is that the total space is discretized and divided into "occupied cells" and "free cells." This fragmentation of free space can cause a combinatorial explosion in the number of free cells (and therefore the number of possible pipe paths) leading to excessive computational time requirements. By eliminating the restriction to 90 deg bends to allow nonorthogonal routes, the number of possible routes for nontrivial problems could become extremely large making a complete exploration of the design space impractical.

Because of obstacles and other routing constraints, the objective function space associated with this design space is often ill-behaved, containing many nonlinearities and/or discontinuities. This type of objective function space causes complications for traditional gradient optimization techniques, which are prone to becoming trapped in local optima. The routing approach presented in this paper uses simulated annealing (described in Section 4) to avoid such difficulties.

3 Manhattan vs. Nonorthogonal Routing

The extensive use of Manhattan routing algorithms for wiring VLSI circuits is primarily due to two factors. First, since a single circuit component may have many electrical connections, the large number of wires in VLSI circuits leads to very high wiring densities. In these cases, the parallel arrangement of wires given by Manhattan routes makes the most efficient use of available area. Second, circuit components are almost always rectangular. Given dense packings of rectangular components, there is little or no advantage to allowing non-Manhattan routes. However, these reasons often do not apply to general routing tasks.

The primary drawback to Manhattan routing approaches for engineering applications is that they generally lead to increased routing lengths and increased numbers of bends. For any two terminals separated by a distance of $(\Delta x, \Delta y, \Delta z)$, the length of the shortest Manhattan route (i.e., the Manhattan distance) between the terminals is $D_M = \Delta x + \Delta y + \Delta z$. This distance is always longer than the standard (Euclidean) distance between the terminals.[2] Route length affects material costs, while costs associated with bends can have multiple sources including equipment and time required for bending pipes, fittings used to connect multiple pipe segments, and increased energy requirements due to pressure drops in fluids moving through the pipe.

Obstacles, which are present in virtually all routing problems, can make the difference between Manhattan and nonorthogonal routes even more pronounced. In Fig. 1, the Manhattan route has a longer length than the nonorthogonal route and requires more bends to avoid the obstacle. Also note that nonorthogonal routing allows bends that are less sharp, resulting in smaller pressure drops across the bends in fluid-carrying pipes.

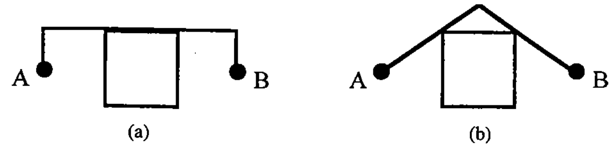


Fig. 1 Manhattan (*a*) vs. nonorthogonal (*b*) routing with obstacles

Consider layout tasks that occur in the mechanical engineering domain where the effect of obstacles is much more significant due to spatial limitations. It would be impossible, for instance, to route all the pipes, wires and hoses in an automobile engine compartment using Manhattan routes without greatly increasing the volume of the compartment. Other routing problems, such as those occurring in airplane or satellite design, share similar characteristics that make the use of Manhattan routing impractical.

Since a nonorthogonal route can include 90 deg bends, Manhattan routes can be thought of as a subset of general nonorthogonal routes. Thus, the best nonorthogonal route will never be worse (i.e., longer and having more bends) than the best Manhattan route; but, as Fig. 1 illustrates, it can be better. Although the routing algorithm developed through this research is aimed at engineering applications that require nonorthogonal routing, the approach can also be used to generate nonorthogonal routes for problems for which Manhattan routes are often considered adequate, such as piping in chemical plants or routing air ducts in buildings. Further, the algorithm presented in this paper can be implemented to generate Manhattan routes should they be required for a particular application.

4 The Routing Algorithm

4.1 Simulated Annealing. Simulated annealing is a stochastic optimization technique introduced by Kirkpatrick et al. (1983). The algorithm is summarized as follows: an initial (design) state is chosen and the value of the objective function for that state is evaluated. A step is taken to a new state by applying a move, or operator, from an available move set. This new state is evaluated; if the step leads to an improvement in the objective function, the new design is accepted and becomes the current design state.

If the step leads to an inferior state, the step may still be accepted with some probability. This probability is a function of a decreasing parameter called *temperature,* based on an analogy with the annealing of metals, given by:

$$P_{\text{accept}} = e^{-(\Delta C/T)}, \tag{1}$$

where ΔC is the change in objective function due to the move and T is the current temperature. The temperature starts out high and decreases with time. Initially, steps taken through the state space (and therefore the objective function space) are almost random, resulting in a broad exploration of the objective function space. As the probability of accepting inferior steps decreases, those steps tend to get rejected, allowing the algorithm to converge to an optimum once promising areas of the objective function space have been found.

The temperature in a simulated annealing algorithm is controlled by an *annealing schedule,* which consists of an initial temperature, the number of steps taken at each temperature, the amount by which the temperature is decreased for each reduction, and an overall number of temperatures. The simulated annealing routing algorithm described in this paper uses an adaptive annealing schedule taken from Huang et al. (1986) which automatically tailors the temperature profile to a given

[1] The limitations of Manhattan routing for mechanical engineering applications are discussed in the following section.

[2] Except for the case in which two of the "deltas" are zero, in which case the two distances are equal.

problem based on statistical information about states visited during the optimization.

In simple simulated annealing algorithms, the probabilities of selecting given moves from the move set are fixed by the user before the algorithm is run. To further improve efficiency, the routing algorithm also uses a move selection strategy introduced by Hustin and Sangiovanni-Vincentelli (1987). This technique dynamically modifies move selection probabilities based on past performance, so that moves which are likely to improve the design have a high probability of being selected and those which are less likely to lead to better designs have a lower probability of being chosen. The selection probabilities are updated periodically as the algorithm runs since the utility of various moves may change as the optimization proceeds. By decreasing the probability of applying moves that will be rejected, the amount of wasted search is reduced.

Szykman and Cagan (1995) present the adaptive annealing schedule and the move selection strategy in greater detail, along with a quantitative comparison with simulated annealing algorithms that do not use these techniques. The equilibrium and freezing conditions, which determine when the temperature should be reduced and when the algorithm should terminate, are described in Huang et al. (1986).

4.2 The Objective Function. To allow designers to optimize a variety of design objectives, the problem formulation utilizes a generic objective function, F, consisting of a weighted sum of the form:

$$F = W_1C_1f_1 + W_2C_2f_2 + \cdots + W_nC_nf_n, \tag{2}$$

where f_i is the value of the ith objective, C_i is the ith normalizing coefficient, and W_i is the weight for the ith term. To avoid scaling problems between objective terms that differ greatly in order of magnitude, each of the terms, f_i, is multiplied by a coefficient, C_i. The value of C_i is one divided by the maximum value of f_i seen at the current temperature, which keeps values of the C_if_i terms between zero and one. Once the objective function terms are scaled using the coefficients, the weights, W_i, allow the user to specify their relative importance.

For this research, the goal of the optimization is to minimize routing length and number of bends. If we define L_{TOTAL} as the sum of the lengths for all the required routes, and B as the number of bends needed for the routes, L_{TOTAL} and B are the first two objectives, f_1 and f_2, in the objective function. To make a broad exploration of the design space easier, the problem formulation permits routes to penetrate obstacles. Clearly, such routes are undesirable in a final routing; thus, a third term is added to the objective function that penalizes obstacle penetrations. As the algorithm runs, the optimization drives the penalty to zero by modifying routes that penetrate objects. The final objective function consists of three terms and takes the form:

$$F = W_1C_1L_{\mathrm{TOTAL}} + W_2C_2B + W_3C_3P_{OP}. \tag{3}$$

P_{OP}, the penalty for obstacle penetrations, is defined as the sum (over all routes) of the squares of the lengths of obstacle penetrations (L_{OP}):

$$P_{OP} = \sum_{\mathrm{Routes}} L_{OP}^2. \tag{4}$$

The objective function given by Eq. (3) is used to solve general routing problems. The number of bends in piping is often an important factor and a reduction in bends is often desirable (the costs associated with bends were discussed in Section 3). However, there may be little or no cost associated with bending wires, which are flexible. To solve wire routing problems when this is the case, the number of bends is ignored by setting the weight in the second term in the objective function to a small or to zero value. Examples of tradeoffs between the cost of route length and bends are given in Section 5. The objective function can also be extended to include additional objectives that may apply to a given problem. Due to the large number of iterations associated with simulated annealing algorithms, the use of this approach becomes less practical for objectives that require computationally expensive evaluations.

4.3 The Move Set. The specifications of a routing task consist of locations for a pair of terminals that must be connected for each route, and a set of obstacles. Given these specifications, an initial state is chosen by connecting the terminals with the simplest routes possible. Generally, these routes consist of straight lines between terminals. For some piping problems a pipe must extend in a straight line from a nozzle a certain distance before it can bend. In such cases, the simplest route consists of short extensions from the terminals which are connected by a straight line. The optimization then begins according to the simulated annealing algorithm described previously. The simulated annealing algorithm moves from one design state to another by applying a move from a predefined move set. For the routing problem, a state is a set of routes that may intersect obstacles.

The move set consists of four different types of moves that perturb a route: *add, remove, relocate* and *vector-relocate.* A route is represented as a list of point coordinates specifying the locations of the endpoints and each of the bends. From this list, the vectors and lengths of route segments are determined. The add move selects a route at random and inserts a new bend at a random location in the list. The remove move selects a route and randomly deletes one of the bends.

The relocate move changes the location of a bend by randomly choosing a bend and moving it a distance DV (i.e., a distance D in the direction V) where V is a randomly generated unit vector. In contrast, vector-relocate moves a bend along the direction of one of the two route segments (chosen at random) that meet at a bend. More formally, if V_s is the unit vector corresponding to the line defined by one of the two segments that meet at a bend, vector-relocate shifts the location of that bend by a distance DV_s. The advantage of the vector-relocate move is that it has the effect of changing the length of one of the route segments that meet at a bend without changing its orientation (though the orientation of the other segment does change). Typically, there are a number of relocate and vector-relocate moves having varying move distances.

The routing algorithm also incorporates several different types of constraints on routing. For instance, the user can set minimum and maximum bend angles, as well as a minimum required distance between bends. Constraints such as these may occur in piping applications to prevent excessive pressure drops for fluids in pipes or to reflect limitations of bending equipment or materials. A move is only made if doing so does not result in the violation of any constraints that have been specified.

5 Experimental Results

In this section, solutions to several routing problems generated by the simulated annealing routing algorithm are given. The first three examples are simple problems of increasing difficulty that illustrate various characteristics of the algorithm. The fourth example is a more complicated routing task of routing components in a chemical production plant. The locations of terminals are defined as points at specific X-Y-Z coordinates relative to the location of the components. These locations may or may not be on obstacle surfaces, depending on the nature of the routing problem at hand.

5.1 Example: Routing with Obstacles. For this example, the task is to find the optimal route from a terminal located at $(-0.5, -0.5, -0.5)$ to a terminal at $(0.5, 0.5, 0.5)$. A cube with edges of length 1.0 is centered at the origin and acts as an obstacle. Because of the penalty for obstacle intersections described in the previous section, the optimization drives inter-

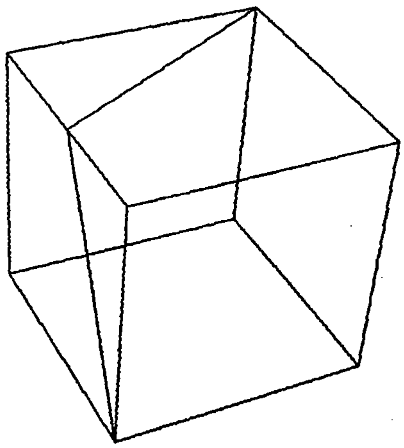


Fig. 2 Optimal routing across a cube

sections to zero by driving the route segments outside of the cube.

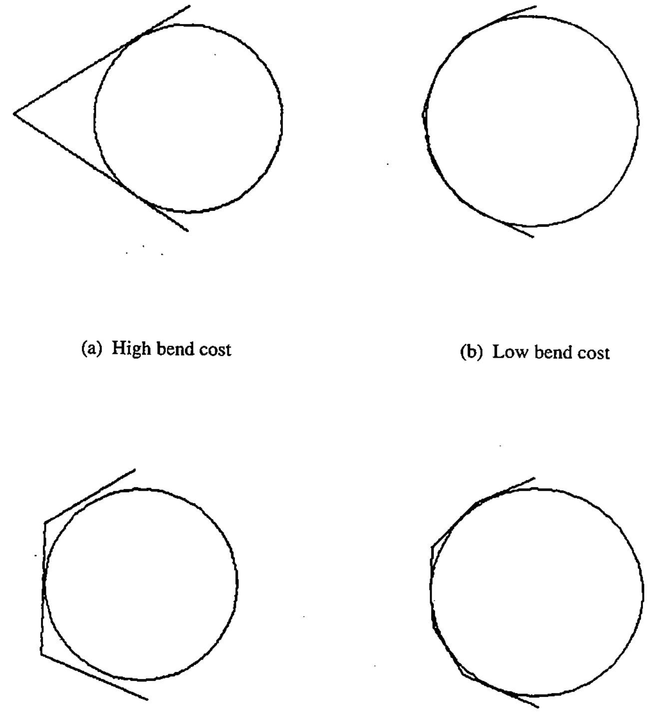
Because of symmetry, the true optimal route is observed to be one with a bend at the center of any one of the cube edges whose endpoint is not one of the terminals, and has a length of 2.236. The solution found by the routing algorithm is shown in Fig. 2. This route has a length of 2.240, which is within 0.18 percent of the true optimum. The solution was generated in slightly over 13,300 iterations and took seven seconds on a DEC Alpha 3000 workstation at a rate of over 1900 iterations per second.

5.2 Example: Tradeoffs in Routing Objectives. The two routing objectives considered in the current formulation are route length and number of bends. The relative costs of route length (raw materials) and bends (time and manufacturing operations) may vary from one task to the next and will affect the kinds of solutions generated by the algorithm. Knowledge about these tradeoffs is reflected by the designer's choice of weights in the objective function given by Eq. (1). If bends are very expensive compared to raw materials, one would expect routes having few bends, possibly at the cost of additional route length, and vice versa.

A potential difficulty in making this type of determination is to correctly quantify tradeoffs between route length and number of bends. One approach to doing this is to reduce these objectives to a common metric. For instance, one common metric for comparing route length and number of bends in pipe routing problems is pressure drop. If the pressure drop per unit length of pipe and the pressure drop across a bend of a given angle are known, weights can be set in the objective function to accurately represent the tradeoff between the two objectives. Another typical metric is a dollar cost, which would assign weights in the objective function based on the cost per unit length of raw materials and an estimated cost for bending a tube or the cost of an angled pipe fitting with a given angle.

In this example, a cylindrical obstacle is placed between two terminals that must be connected by a route. The cost of bends is initially made high relative to the cost of route length (i.e., $W_2 \gg W_1$ in Eq. (3)). The result, shown in Fig. 3*a* is a route with a minimal number of bends. One bend is required because a zero-bend route would intersect the obstacle.

The other extreme is one in which bends have very little cost compared to route length. The weights are changed so that $W_1 \gg W_2$ and the algorithm is run again. This time, the route has 40 segments and very closely approximates the curve along the cylinder circumference that minimizes the route length (see Fig. 3*b*). By modifying the tradeoff between the weights, solutions between the two extremes can be found, as shown in Figures 3*c* and 3*d*.



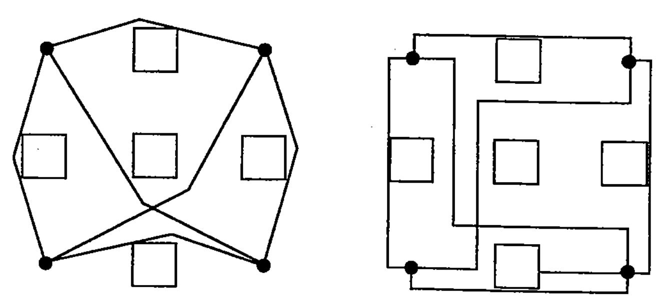
(a) High bend cost (b) Low bend cost

(c) and (d) Intermediate weightings

Fig. 3 Tradeoffs between objectives

5.3 Example: Nonorthogonal vs. Manhattan Routing. As discussed in Section 3, one of the motivations for a nonorthogonal routing algorithm is savings both in route length and number of required bends. This example compares the nonorthogonal and Manhattan routing approaches applied to the same problem. For this task, four terminals are located at (10, 10, 0), (10, −10, 0), (−10, 10, 0), (−10, −10, 0), and six routes are specified connecting each possible pair of terminals. Five obstacles of dimensions $4 \times 4 \times 10$ are placed halfway between each pair of points.

The nonorthogonal routes generated by the simulated annealing routing algorithm are shown in Fig. 4*a* and the optimal Manhattan routes are shown in Fig. 4*b*. The total length for the routes in Fig. 4*a* is 146.9. For the Manhattan routes, the length of each of the four "outside" routes is 24 and the length of the



(a) Non-orthogonal routing (b) Manhattan routing

Fig. 4 Optimal routes for example problem

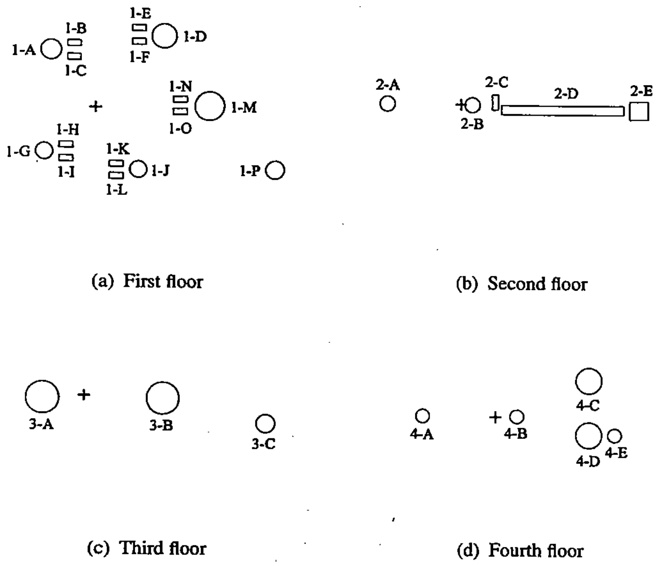


Fig. 5 Chemical plant layout based on optimized layout from Gunn and Al-Asadi (1987)

two "diagonal" routes is 40 giving a total length of 176, which is 20 percent longer than the nonorthogonal route. Each of the six routes in Fig. 4*a* has one bend, for a total of six bends. The Manhattan routes have a total of 14 bends, more than twice as many. Furthermore, had the terminals had different Z coordinates, additional bends would have been required to extend the Manhattan routes out of the plane, whereas the nonorthogonal routes can move out of the plane without additional bends.

The nonorthogonal routes in Fig. 4*a* were generated in 51,100 iterations and took 36 seconds to generate, for a rate of over 1400 iterations per second. This rate is lower than the rate cited in an earlier example because the additional routes (six versus one) leads to an increased number of intersection tests.

5.4 Example: Routing of a Chemical Plant. In this section, we illustrate the use of the simulated annealing routing algorithm on a more realistic problem. Gunn and Al-Asadi (1987) describe a CAD tool for chemical plant layout, and use as a case study the layout of a four-story polyester production plant containing four parallel production lines. From that optimized layout, a reduced example of a single production line plant was adapted.

Figure 5 shows a planar (top view) layout for each of the four floors in the plant.[3] Crosses mark the same X and Y coordinates on each floor; a key to the components in the plant layout is given in Appendix A. A material flow diagram for the process is given in (Gunn and Al-Asadi, 1987). For the original case study, layout was done for components in the process but not for the entire plant. Thus, only connections between components are considered in the routing problem; inputs to the process (raw materials, water for heat exchangers) and outputs from the process (final product, waste chemicals) are not shown.

Figure 6*a* shows the layout of the chemical plant before routing (spacing between floors has been slightly exaggerated to make viewing components easier). Note that geometry of the components has been simplified and they are represented using blocks and cylinders.

The shortest possible Manhattan routing is one in which no additional route segments are required to avoid obstacles. If no additional segments are required, the length of a Manhattan route is simply the Manhattan distance between the two terminals. Thus, if we assume such a routing is possible we can obtain a best bound on the optimal Manhattan routing, without having to perform the actual routing, by taking the sum of the Manhattan distances over all of the routes. For this problem, the optimal Manhattan route has a lower bound on length of 1832.1. We can also determine the lower bound on number of bends required for a Manhattan routing. At best, a Manhattan route from point A to point B will require no bends if A and B have two identical coordinates, one bend if the points have one identical coordinate, and two bends if they have no identical coordinates. Consequently, the fewest number of bends needed for the Manhattan routing of the chemical plant is 59.

Figure 6*b* illustrates the three dimensional view of the chemical plant after routing with the simulated annealing algorithm. The routing shown has a total length of 1315.8, 28.1 percent less than the shortest possible Manhattan routing. The number of bends required for the routes was 23, well under half of what would be needed for the best Manhattan routing. The routing shown in Fig. 6*b* was generated in slightly under 260,000 iterations at a rate of approximately 1000 iterations per second.

6 Concluding Remarks

This research introduces a novel approach to nonorthogonal routing optimization for engineering applications. The use of the routing algorithm is demonstrated using a realistic routing problem: a chemical production plant. Although the work in this paper is aimed at problems for which Manhattan routing is impractical, the algorithm can also be used to perform nonorthogonal routing for tasks for which Manhattan routes are generally considered adequate, such as piping for chemical plants or routing air ducts in buildings.

The current algorithm contains several limitations that need to be addressed in future work. Presently, the route segments are considered to be one dimensional. For wire routing problems or applications where the route diameter is small compared to spaces between obstacles, the route thickness can be ignored. For other problems, however, route diameters can not be neglected and must be considered when performing interference tests between routes and obstacles or other routes. Extending the representation of routes to three dimensions would provide a more accurate characterization of designs and would also allow additional problem-specific knowledge, such as constraints on bend radii, to be incorporated into the optimization.

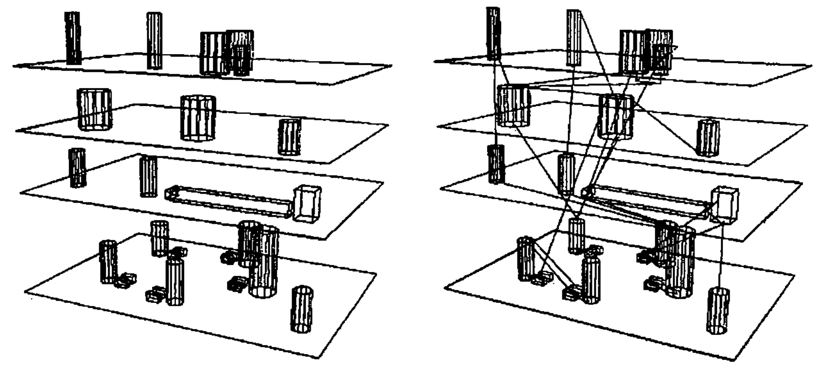
A second limitation is that obstacles are modeled by rectangular blocks or cylinders. Many objects have shapes that can be approximated using one or more of these primitive shapes; however, because components in some routing problems may have rather complex geometries, a more sophisticated object representation may be necessary.

Another potentially useful extension would be to replace the current move set with a similar one which is formalized as a shape grammar. Using this approach, called shape annealing (Cagan and Mitchell, 1993), domain knowledge (such as information about pipe bending constraints) would be represented implicitly within the design generation process rather than testing constraints after a step is taken. In particular, a grammatical approach is one way to constrain the routing algorithm described in this paper to 90 deg bends to perform Manhattan routing.

In some instances, a routing task is only part of a larger design problem. For instance, wire harness design consists of several subproblems including configuration design, part selection, and routing (e.g., Wu et al., 1992; Conru and Cutkosky, 1993; Park et al., 1994). In cases such as these, it may be possible to improve the performance of existing design tools by using them in conjunction with the algorithm described in this paper.

Many problems that require routing also involve layout of components. Typically, the first step is to lay out components, after which the routing is performed. This often makes the

[3] Because of incomplete data in the original paper, terminal (i.e., nozzle) locations and some dimensions have been estimated from figures.



(a) before routing (b) after routing

Fig. 6 Three-dimensional view of chemical plant

routing task more difficult because routing objectives are ignored during the component layout phase. An additional area of future work is to incorporate this routing algorithm into a component layout algorithm developed by Szykman and Cagan (1994) to allow simultaneous layout and routing. Because routing considerations would be taken into account at the layout stage, it is expected that this approach will lead to improvements over traditional layout-then-routing procedures.

Acknowledgments

The authors would like to thank the National Science Foundation (under grant #DDM-9258090) and United Technologies (supplying matching funds) for supporting this research. We also wish to thank Rick Clark, Pratip Dastidar and Paul Weisser for their valuable discussions regarding this work.

References

Asano, T., Asano, T., and Imai, H., 1987, "Shortest Path Between Two Simple Polygons," *Information Processing Letters,* Vol. 24, pp. 285–288.

Cagan, J., and Mitchell, W. J., 1993, "Optimally Directed Shape Generation by Shape Annealing," *Environment and Planning B,* Vol. 20, pp. 5–12.

Cohn, J. M., Garrod, D. J., Rutenbar, R. A., and Carley, L. R., 1991, "KOAN/ANGRAM II: New Tools for Device-Level Analog Placement and Routing," *IEEE Journal of Solid-State Circuits,* Vol. 23, No. 3, pp. 330–342.

Conru, A. B., and Cutkosky, M. R., 1993, "Computational Support for Interactive Cable Harness Routing and Design," *Advances in Design Automation 1993: Proceedings of the 19th ASME Design Automation Conference,* DE-Vol. 65-1, Albuquerque, NM, September, pp. 551–558.

Gunn, D. J., and Al-Asadi, H. D., 1987, "Computer-Aided Layout of Chemical Plant: A Computational Method and Case Study," *Computer-Aided Design,* Vol. 19, No. 3, pp. 131–140.

Huang, M. D., Romeo, F., and Sangiovanni-Vincentelli, A., 1986, "An Efficient General Cooling Schedule for Simulated Annealing," *ICCAD-86: IEEE International Conference on Computer-Aided Design—Digest of Technical Papers,* Santa Clara, CA, November 11–13, pp. 381–384.

Hustin, S., and Sangiovanni-Vincentelli, A., 1987, "TIM, a New Standard Cell Placement Program Based on the Simulated Annealing Algorithm," IEEE Physical Design Workshop on Placement and Floorplanning, Hilton Head, SC, April.

Jain, D., Chatterjee, M., Unemori, A., and Thangam, N., 1992, "A Knowledge Based Automotive Pipe Routing System," *Computers in Engineering: Proceedings of the 1992 ASME International Computers in Engineering Conference and Exposition,* San Francisco, CA, August, pp. 127–132.

Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P., 1983, "Optimization by Simulated Annealing," *Science,* Vol. 220, No. 4598, pp. 671–679.

Mitsuta, T., Kobayashi, Y., Wada, Y., and Kiguchi, T., 1986, "A Knowledge-Based Approach to Routing Problems in Industrial Plant Design," *Proceedings of the 6th International Workshop on Expert Systems & Their Applications,* Avignon, France, April, pp. 237–256.

Ohtsuki, T., ed., 1986, *Layout Design and Verification,* (Chapter 3: "Maze-Running and Line-Search Algorithms"; Chapter 4: "Channel Routing"; Chapter 5: "Global Routing"), Elsevier Science Publishing Company.

Park, H., Cutkosky, M. R., Conru, A. B., and Lee, S-H, 1994, "An Agent-Based Approach to Concurrent Cable Harness Design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol. 8, No. 1, pp. 45–61.

Sechen, C., and Sangiovanni-Vincentelli, A., 1985, "The TimberWolf Placement and Routing Package," *IEEE Journal of Solid-State Circuits,* Vol. 20, No. 2, pp. 510–522.

Shin, H., and Sangiovanni-Vincentelli, A., 1987, "A Detailed Router Based on Incremental Routing Modifications: Mighty," *IEEE Transactions on Computer Aided Design,* Vol. 6, No. 6, pp. 942–955.

Szykman, S., and Cagan, J., 1994, "Constrained Three Dimensional Component Layout Using Simulated Annealing," *Concurrent Product Design: Proceedings of the 1994 International Mechanical Engineering Congress and Exposition,* ASME, DE-Vol. 74, Chicago, IL, November, pp. 175–184 (also accepted in ASME JOURNAL OF MECHANICAL DESIGN, 1996.)

Szykman, S., and Cagan, J., 1995, "A Simulated Annealing-Based Approach to Three Dimensional Component Packing," ASME JOURNAL OF MECHANICAL DESIGN, Vol. 117, No. 2(A), pp. 308–314.

Wu, Y., Champaneri, R., and Mehta, P., 1992, "A Knowledge-Based Cable Harness Design Rule Checker to Improve Effectiveness and Efficiency," *Proceedings of the First International Conference on Intelligent Systems Engineering,* Edinburgh, UK, August, pp. 185–187.

Zhu, D., and Latombe, J-C, 1991, "Mechanization of Spatial Reasoning for Automatic Pipe Layout Design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol. 5, No. 1, pp. 1–20.

APPENDIX A

Key to Plant Components

Floor 1
1-A. Condensate tank
1-B. Condensate pump
1-C. Condensate pump
1-D. Demineralized water tank
1-E. Demineralized water pump
1-F. Demineralized water pump
1-G. Titanium dioxide slurry preparation tank
1-H. Titanium dioxide pump
1-I. Titanium dioxide pump
1-J. Ethylene glycol measure vessel
1-K. Ethylene glycol pump
1-L. Ethylene glycol pump
1-M. Crude mixture collection vessel
1-N. Crude mixture pump
1-O. Crude mixture pump
1-P. Chip-collecting vessel

Floor 2
2-A. Ethylene glycol and methyl alcohol collector
2-B. Ethylene glycol and methyl alcohol collector
2-C. Water pump
2-D. Cooling trough
2-E. Slicer-dicer

Floor 3
3-A. Esterification reactor
3-B. Polycondensation reactor
3-C. Catalyst A storage tank

Floor 4
4-A. Ethylene glycol and methyl alcohol condenser
4-B. Ethylene glycol and methyl alcohol condenser
4-C. Dimethyl terepthalate feed vessel
4-D. Ethylene glycol heater
4-E. Catalyst B preparation vessel