IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 11, 2020, accepted February 19, 2020, date of publication March 2, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977825

An Efficient Rectilinear and Octilinear Steiner Minimal Tree Algorithm for Multidimensional Environments

MING CHE LEE[1], GENE EU JAN[2], (Member, IEEE), AND CHUNG CHIN LUO[3]
[1]Department of Computer and Communication Engineering, Ming Chuan University, Taoyuan 333, Taiwan
[2]Tainan National University of the Arts, Tainan 72045, Taiwan
[3]Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan

Corresponding author: Ming Che Lee (leemc@mail.mcu.edu.tw)

ABSTRACT The rectilinear/octilinear Steiner problem is the problem of connecting a set of terminals Z using orthogonal and diagonal edges with minimum length. This problem has many applications, such as the EDA, VLSI circuit design, fault-tolerant routing in mesh-based broadcast, and Printed Circuit Board (PCB). This paper proposes an obstacle-avoiding 4/8/10/26-directional heuristic algorithm for this problem based on the Areibi's concept, Higher Geometry Maze Routing, and Sollin's minimal spanning tree algorithm. The major contributions of this paper are (1) our work is the first report for the octilinear SMTs in the multidimensional environments, (2) we provide an optimal point-to-point routing without any refinement, and (3) the proposed algorithm has higher adaptability to deal with any irregular environment, and can be extended to the λ-geometry without any extra work, where $\lambda = 2, 4, 8$ and ∞ corresponding to rectilinear, 45°, 22.5° and Euclidean geometries respectively.

INDEX TERMS Higher geometry maze routing, octilinear, rectilinear, Steiner tree.

I. INTRODUCTION

Routing plays an important role in many applications, such as the Electronic Design Automation (EDA), Very Large Scale Integrated (VLSI) circuit design (X Initiative), fault-tolerant routing in mesh-based broadcast [1], [2], decision systems[3], and Printed Circuit Board (PCB) [4], [5]. Finding a minimal length spanning structure using orthogonal and diagonal segments connecting a set of vertices is critical for multi-pin net in global routing of VLSI design, as shown in Fig. 1. The objective is to connect a set of terminals with minimal wire length so that the there exists a path between any two terminals. The problem can be formalized as the minimal spanning tree (MST) problem. Optimal connection with shortest wire length of given terminals can be achieved via some additional terminals or Steiner Vertexes. The optimal connection problem is so-called the Steiner minimal tree (SMT) problem.

The Steiner minimal tree problem in Euclidean and rectilinear routing model (rectilinear Steiner minimal tree, RSMT) has been well studied in the literatures [6]–[14]. It is similar to the Minimal Spanning Tree (MST) problem. The difference is that a minimal spanning tree is formed by edges that link all the required vertices, while minimal Steiner trees can connect some auxiliary nodes, which are called *Steiner vertices*. Consider the five nodes in Fig. 2. A minimal spanning tree can be easily constructed by linking these five nodes, as shown in Fig. 2(a). However, Fig. 2(b) reveals that the Steiner minimal tree has 6 nodes, i.e. one Steiner vertex has been introduced. That is, the dashed edges in Fig. 2 depict the subtle difference between them.

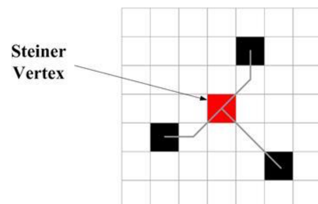


FIGURE 1. Octilinear steiner minimal tree.

Until now, there are some important issues that have not been fully solved: the first is to apply the RSMT to octilinear interconnection that is widely used in IC design, such as the multi-pin nets global routing of VLSI design [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Mehmood.

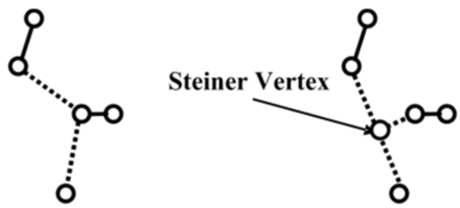


FIGURE 2. Minimal spanning trees vs. minimal steiner trees.

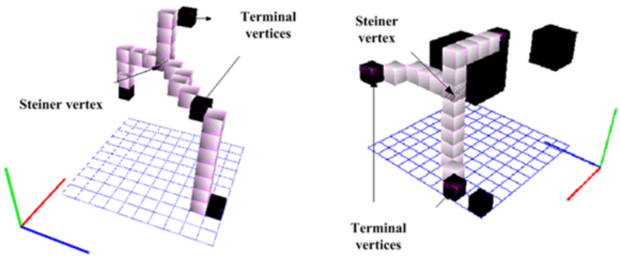


FIGURE 3. 3D Minimal Steiner Trees (a) Minimal Steiner Tree in a multilayer grid without obstacles, (b) Minimal Steiner Tree in a multilayer grid with obstacles.

The second is to develop non connection-graph or non-Delaunay triangulation algorithms for arbitrary obstacles. Recently many Connection Graph (CG) and Delaunay Triangulation (DT) based algorithms have been proposed and have outstanding theoretical worse case running time [4], [15]–[19]. These algorithms have the feature of obstacle avoidance for rectilinear and even polygon obstacles.

However, how to handle the environment with arbitrary obstacles, such as single-cell obstacle and non-polygon obstacles is still a challenge. The other issue concerns with measuring the performance of the algorithms while the minimal Steiner tree (MST) problem is known as NP-complete [10], [11].

In the previous research, besides the time and space complexities, experimental results or simulations of average wire length are usually provided to illustrate the effectiveness and efficiency for the proposed algorithms. This paper proposes an obstacle-avoiding 4/8/10/26-directional heuristic algorithm. Fig. 3 is a schematic diagram of multilayer MST. Fig. 3(a) depicts the minimal Steiner tree of four vertices in a multilayer grid without obstacles, while Fig. 3(b) illustrates a different minimal Steiner tree of three vertices with one Steiner vertex when there are obstacles around. The proposed algorithm adapts Areibi's concept and the recursive minimal spanning tree algorithm of Boruvka (also attributed to Sollin and commonly known as the Sollin's algorithm) to obtain minimal Steiner tree [20]–[23]. The rest of this paper is organized as follows. Section II briefly describes background and related works. Section III outlines the minimal Steiner tree algorithms. Section IV shows the experimental results, space and time complexities. Section V concludes this paper.

II. STEINER PROBLEM AND CURRENT RELATED WORKS

A. THE STEINER PROBLEM

The Steiner problem can be formulated as follows: Given a network $G = (V, E, C)$, where V is a set of vertices, E is a set of edges, and C: $E \rightarrow R$ is the cost function that maps every edge in E to a real-number cost. Find a subnetwork G_S of G for a set of required vertices Z ($Z \subseteq V$) that contains a path between every pair of required vertices and minimizes the cost $\sum_{(v_i, v_j \in G_S)} c(v_i, v_j)$.

Due to the presence of obstacles, the previous works on the obstacle-avoiding Steiner minimal tree (OASMT) problem suffer the decrement of quality and increment of running time. Very recently, some OASMT algorithms with significant running time are proposed and most of the researches adopt the MST strategy. Lin *et al.* [24] proposed an obstacle-avoiding rectilinear Steiner minimal tree (OARSMT) construction algorithm that achieves an optimal solution for two-pin net. Their approach guarantees to provide a rectilinear shortest path between any two pins. Nevertheless, the approach is applicable only to rectangle obstacles and has the $O(n^3)$ theoretical time complexity. Jing *et al.* [15] presented a three stages OARSMT routing algorithm that is based on the Delaunay triangulation [25]. In Jing *et al.* [15] spanning graph is used and the obstacle avoiding feature is inherited from the connected graph automatically. This approach first constructs a connected graph by the boundaries of obstacles and then a spanning graph is generated from the graph. The algorithm of Jing *et al*. has $O(n\log n)$ worse-case time complexity. However, the performance of the proposed algorithm strongly depends on the global view of the pins or obstacles. And similarly, Long *et al.* [16] proposed a spanning graph based approach with the same time complexity and improved the local refinements to a global view. The above mentioned connected/spanning graph-based approaches try to capture the global blockage information and may have a large total wire length and running time both when the number of obstacles is large and in handling polygon/irregular-shaped obstacles. Besides, the additional cost of calculating the corner vertices of obstacles is not mentioned.

In a nutshell, CG/DT based approaches are difficult to deal with the situations shown in Fig. 4. In the case of Fig. 4(a), the path of CG/DT based approaches from source to destination is connected by the corner of the obstacle as the black-dashed line, and this may not be the shortest. In fact in the 3-D environment, the shortest path is the white-dashed line as shown in Fig. 4(a). Furthermore, these approaches are insufficient to meet both the cases of obstacles with smooth boundaries and obstacles with arbitrary shapes, such as single-cell obstacles. The two cases are depicted in Fig. 4(b) and Fig. 4(c), respectively. The researches mentioned above concern rectilinear SMTs, and recently many researchers developed their heuristics in the octilinear

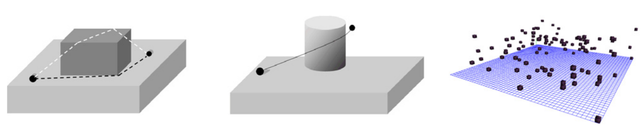


FIGURE 4. Three issues in the 3D environment. (a) 3D shortest path (b) Obstacle with smooth boundary (c) Single cell obstacles.

architecture. Traditional Manhattan architecture considers only horizontal and vertical directions, and clearly this restriction leads to non-optimal routing over the Euclidean plane.

The algorithm of Jing *et al*. can be extended to λ-geometry plane, such as the Y/X-architecture (λ_3/λ_4- geometry). The Steiner problem in rectilinear and octilinear routings has been well studied [4], [5], [17], [18], [26]–[29]. Zhu *et al.* [19] proposed two octilinear SMT (OSMT) construction algorithms (OST-E, octilinear Steiner tree by edge substitution and OST-T, octilinear Steiner tree by triple construction) based on octilinear spanning graphs (OSGs), which allows 45° diagonal interconnections. Their algorithms are easy to find neighbors for wire length refinement; however, the possible local optimization of four or more neighbors is not taken into account.

Lee *et al.* [30] proposed a DT based heuristic algorithm that considered only local substitutions and [31]–[34] presented liquid routing-based technologies in the X architecture. In Dong *et al.* [35], a tree splitting-merging based OARSMT and obstacle-avoiding octilinear Steiner minimal tree (OAOSMT) heuristic was presented to solve IC routing problems both in Manhattan Architecture and X-Architecture. In 2015, Huang *et al.* [36] presented a DT based heuristic algorithm with $O(n * m)$ time complexity, where n and m are the number of terminal nodes and obstacles, respectively. In their work, Delaunay triangulations (DT) was first generated for the given terminal nodes, then corner points of obstacles were selected as the Steiner point. Huang *et al.* is the current state of the art in OAOSMT construction problem. Furthermore, some artificial intelligence technologies were applied to multi-scale routing [37]–[39].

The other MST based approaches concern the concept of maze routing that was first mentioned by Lee [34]. Clow [40] presented the A∗ maze routing and Warren [41] presented an enhanced A∗ to reduce the time and space complexities. Hentschke *et al.* [42] presented a fast maze routing-based algorithm to build Steiner trees. A sharing factor and a path-length factor were introduced to trade-off wire length for delay. Researches presented a heuristic maze routing-based approach for constructing large scale OARSMTs [43]–[48]. These methods of maze routing mentioned so far only take the rectilinear cases into account. This is because in the octilinear condition, the time complexity and memory usage of the traditional maze routing algorithms grow prohibitively huge as the routing area becomes larger, and this drawback make the maze-routing based approaches less popular for modern applications. For the obstacle-avoiding octilinear MST (OAOMST) problem, Huang *et al.* [49] proposed a Particle Swarm Optimization-based (PSO) single-layer OAOMST approach 2015, and further improved its performance with the concept of DT [50] and applied it to the X-Architecture [51].

In this paper, we will show that the maze routing-based approach can also handle both the rectilinear and octilinear SMT problems effectively, due to the acceptable running time and quality of results.

In this paper, we construct a maze routing based octilinear OASMT, which is an extension of the Higher Geometry Maze Routing (HGMR) algorithm that was first introduced in [9]. Our previous work [52] improves Lee's rectilinear routing algorithm [34] to the λ-geometry plane, and the significant improvement is that it has the same time and space complexities of $O(N)$ as Lee's algorithm by a specially designed data structure. The λ-geometry allows edges with the angles of $i\pi/\lambda$, for all i, $\lambda = 2, 4, 8$ and ∞ corresponding to rectilinear, 45°, 22.5° and Euclidean geometries respectively. This paper introduces 2D 4/8- directional and 3D 10/26-directional heuristic OASMT algorithms that based on maze routing approach (see section 3.1). To our knowledge, this study is the first report of maze routing based approaches for octilinear SMT construction and arbitrary obstacles (not necessary maze routing based). The main contributions of this paper are listed as follows: (1) the proposed method has high flexibility and ability to handle arbitrary blockages, (2) this method produces optimal point-to-point paths both in 2D planes and 3D volumes without local refinement. This quality obviously outperforms other graph based approaches in wire length in multidimensional volumes, and (3) we demonstrate that the Steiner ratio of our algorithms is 1.25.

III. THE PROPOSED STEINER TREE ALGORITHM

A. λ-GEOMETRY MAZE ROUTING

Hanan Grid Theorem states that there exists a minimal Steiner tree with all the Steiner points chosen from the Hanan vertices. As shown in Fig. 5, a finite set $H(S)$ is obtained by constructing the horizontal and vertical lines of the terminal Set Z in a planar grid, i.e., the Hanan Vertices. According to Hanan's theorem, the Steiner Point of any rectilinear minimal Steiner tree will fall into the set $H(S)$. Subsequently Areibi developed an iterative algorithm based on this theorem [20] and Lee's shortest path connection algorithm [34], which is the most commonly used algorithm for finding a path between two vertices on a planar rectangular grid. Lee's shortest path algorithm can be divided into three phases:

(1) *Wave propagation phase*(*Exploration phase*): Starting from the source node, wavefronts are propagated

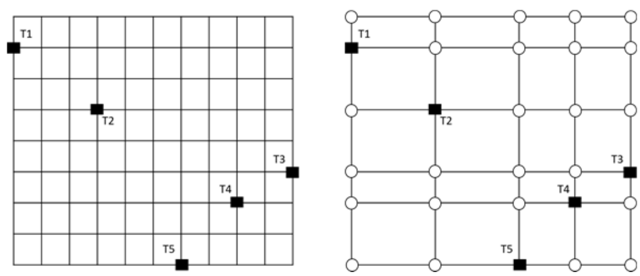


FIGURE 5. An example of rectilinear grid (a) Rectilinear grid with terminals T1∼T5 (b) Hanan Grid and Hanan vertices.

to neighboring nodes until the destination node is reached.

(2) *Backtracking phase*: A path is backtracked from the destination node to the source node.

(3) *Reversing phase*: The shortest path is obtained by reversing all the edges of the path.

Later Jan *et al.* improved Lee's algorithm by extending the propagation and backtracking processes from 4 directions to 8 directions [52]. The key to the popularity of Jan's algorithm is its simplicity and its guarantee of finding an optimal solution if it exists with the time complexity of O(λN) for the λ-geometry maze routing, where N is the number of vertices on the grid.

B. THE PROPOSED ALGORITHM

The proposed algorithm adapts Areibi's and Jan's algorithms to generate shortest paths in the multidimensional environments. The result of wave propagation in the proposed algorithm can be interpreted as the gravitational effect relative to starting point. The larger the value in the grid, the smaller the gravity of the point relative to the starting point, that is, the distance is inversely proportional to the magnitude of the attraction. The algorithm uses the concept of equal distance accumulation to simulate the result of the addition of all force fields, and then find the critical vertices closest to the average value of each node. Then a new minimal spanning tree will be constructed after adding one of Hanan vertices to Z and its improvement over the previous minimal spanning tree will be computed. This process will continue until the improvement becomes non-positive. A minimal spanning tree will be constructed with one of candidates plus all the points in Z and its length will be compared with the length of the minimal spanning tree of Z only.

The difference for every free vertex will be stored in a table, called the Improvement Table, and Steiner vertices will be chosen from the candidates based on this table by the greedy method. Consider the nodes Z1 to Z5 in Fig. 6. The length of the minimal spanning tree of these nodes is 16.73 units, as shown in Fig. 6(a). The Improvement Table in Fig. 6(b) reveals that the cost of connecting these nodes can be reduced if Steiner vertices are used. The length of the Steiner tree is reduced to 14.07 after adding one Steiner node, as shown in Fig. 6(c). The 4-directional weight of distance is 1 in each direction, and 1 and $\sqrt{2}$ corresponding to the angles of 90° and 45° in the 8-directions. The basic idea of the 3D 10/26-directional SMT algorithm is the same as the 2D 4/8-directional algorithm.

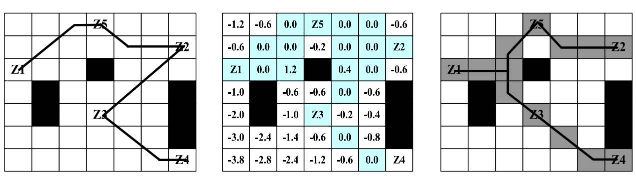


FIGURE 6. 2D Example of SMT and Improvement Table (a) Initial minimal spanning tree, (b) Improvement Table, and (c) The SMT after adding one Steiner node.

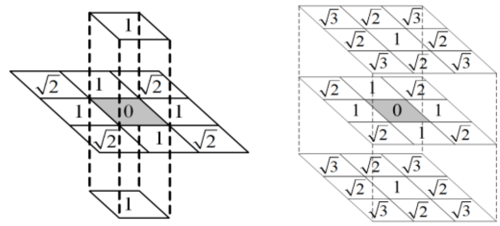


FIGURE 7. 10/26-Directional Propagation (a) 10 Directions and (b) 26 Directions.

TABLE 1. Notation used in the 2-dimentional 4/8-directional algorithm.

Notation	Meaning
$v_{i,j}$	Vertex, stores the information of each vertex in the $I \times J$ cell map, $0 \le i < I$, $0 \le j < J$.
$AD(i, j)$	Array of distance, records the distance from vertex (i,j) to other vertices.
LL^Z	Linked list of *Z*-vertices.
IT	Improvement Table. In *IT*, a positive number represents that the total wire length can be reduced by adding the vertex as a Steiner vertex.

The difference lies at the procedure of λ-geometry wave propagation, and the weight of distance can be 1, $\sqrt{2}$, and $\sqrt{3}$. The 10-directional connection is widely used in the VLSI design since multiple circuit layers of VLSI or PCB are linked through vertical vias.

As a result, every node in a multi-layer mesh has 10 neighbors, as shown in Fig. 7(a). In additional, the 26-directional connection is more general, and each node has 26 neighbors as illustrated in Fig. 7(b). Table 1 lists the notations of the proposed algorithm. The complete 4-directional algorithm is given in Table 2.

C. COMPLEXITY ANALYSIS

1) SPACE COMPLEXITY OF THE 2D 4/8-DIRECTION

Let N_{free} be the number of free nodes and $N_{obstacle}$ be the number of obstacle nodes, then $p + N_{free} + N_{obstacle} = N$. Step 1 of the algorithm creates a matrix $AD(i, j)$ for every vertex $v_{i,j}$ and every free node in the space and hence the space requirement is $(p+N_{free}) \times N$. Step 2 takes a space of N for the *Improvement Table IT*. In addition, two auxiliary data structures will be created during computation and each will takes at most N nodes. Consequently, the space complexity of this algorithm is $(p + N_{free}) \times N + N + 2 \times N = (p + N_{free} + 3)N = O(N^2)$.

2) SPACE COMPLEXITY OF THE 3D 10/26-DIRECTION

Step 1 of the algorithm will create a matrix $AD(i, j, k)$ for every vertex $v_{i,j,k}$ and every free node in the space and hence the space requirement is $(p + N_{free}) \times N$. Step 2 will take a space of N for the *Improvement Table IT*. The space

TABLE 2. The proposed SMT algorithm.

4/8-Directional SMT Algorithm
Input O // O is the set of obstacles
V // V is an $I \times J$ matrix denotes the vertices.
Output MST //Minimal Spanning Tree
Distance Matrix AD = **null**
$LL^Z = \emptyset$
$MST_{initial}$ = **null**
$MST_{i,j}$ = **null**
//Step 1: Compute the flooding distances from every node
For each $v_{i,j} \in V$ **Do**
If $v_{i,j} \in O$ **Then CONTINUE**
$AD_{i,j} = HGMR(v_{i,j})$
If $v_{i,j} \in Z$ **Then**
Insert($LL^Z \cup \{ v_{i,j} \}$)
End For
// Step 2: Compute the minimal spanning tree of Z nodes
$MST_{initial} = Sollin's\_Algorithm(LL^Z)$
//Step 3: Compute the Improvement Table IT
For each $v_{i,j} \in V$ **Do**
If $v_{i,j} \notin (LL^Z \cup O)$ **Then** // $v_{i,j}$ is a free node
$MST_{i,j} = Minimal\_Spanning\_Tree(LL^Z \cup \{ v_{i,j} \})$
$IT_{i,j} = \lvert MST_{initial} \rvert - \lvert MST_{i,j} \rvert$
End For
//Step 4: Find a Steiner node
$LL^Z = LL^Z \cup \{ Maximum\_Positive\_node(IT) \}$
//Step 5: Find more Steiner nodes
Repeat Steps 2 to 4 until no more nodes can be found
//Step 6:
Return $MST = Minimal\_Spanning\_Tree(LL^Z)$

complexity of this algorithm applied to 3D model is $(p + N_{free}) \times N + N + 2 \times N = (p + N_{free} + 3)N = O(N^2)$.

3) TIME COMPLEXITY OF THE 2D 4/8-DIRECTION

In step 1, this algorithm computes the flooding distances from every node in Z and each free vertex. Since each computation of flooding distances costs $O(N)$, the time complexity of this step is $O(N) \times (p + N_{free}) \leq O(N) \times N = O(N^2)$. Step 2 of this algorithm finds the minimum spanning tree connecting all the nodes in Z using the Sollin's algorithm, which has the time complexity of $O(p\log_2 p)$. In Step 3, a free vertex will be picked and a minimum spanning tree will be constructed to connect this free vertex and all nodes in Z, and then the difference between the length of this MST and the MST computed in Step 2 will be stored in the *Improvement Table*. Since constructing a minimum spanning tree of $p + 1$ nodes costs $O((p+1)\log_2(p+1))$ and there are total N_{free} free vertices, this step takes $O((p+1)\log_2(p+1)) \times N_{free} \sim O(Np\log_2 p)$. Step 4 chooses the node with the largest positive number from the *Improvement Table*, and hence its cost is only at most $O(N)$. Step 5 will repeat the processes of Steps 2 to 4 for at most $p-2$ times, since it is proven that at most $p - 2$ Steiner nodes can be found from the set of p nodes Z [53]. As a result, the time complexity will be $(p - 2) \times (O(p\log_2 p) + O(Np \log_2 p) + O(N)) \sim O(Np^2 \log_2 p)$. The final Steiner minimal tree will be constructed in Step 6, which has the complexity of $O(p^2)$. Combining these steps, the time complexity of this algorithm should be $O(N^2) + O(Np^2 \log_2 p) + O(p^2) \sim O(N^2 + Np^2 \log_2 p)$.

4) TIME COMPLEXITY OF THE 3D 10/26-DIRECTION

The basic idea of the 3D 10/26-directional SMT is the same as the 2D 4/8-directional algorithm. The major difference between the 2D 4/8-directional algorithm and the 3D 10/26-directional algorithm is the flexibility of flooding. This variation corresponds to the size of N and the order remains unchanged. Therefore, the total time complexity of this algorithm is $O(N^2 + Np^2 \log_2 p)$.

D. ERROR ANALYSIS

The proposed algorithms find minimal Steiner trees with a Steiner ratio of 1.25, where the Steiner ratio is defined as the smallest upper bound on the ratio between the length of a minimum spanning tree and the length of a Steiner minimum tree of the same set of terminals [23], i.e., let M be the metric space and P a finite set of terminals in M, the Steiner ratio $\rho(M) = supremum L_{MST}(P) / L_{SMT}(P) \, | P \subset M$, where $L_{MST}(P)$ and $L_{SMT}(P)$ are the lengths of the minimal spanning tree and Steiner minimal tree on P, respectively.

Lemma 1: When a Steiner minimal tree or a minimal spanning tree of a set of nodes in the Euclidean space is translated to the octilinear architecture, its new length will be at most 1.0842 times of its original length. In other words,

$$|MST|_E \leq |MST|_X \leq 1.0842|MST|_E \quad \text{and}$$
$$|SMT|_E \leq |SMT|_X \leq 1.0842|SMT|_E,$$

where $|MST|_E$ and $|MST|_X$ represent the lengths of minimal spanning trees in the Euclidean space and octilinear architecture respectively, and $|SMT|_E$ and $|SMT|_X$ denote the lengths of Steiner minimal trees in the Euclidean space and octilinear architecture respectively.

Proof: When horizontal, vertical, or orthogonal edges are translated from the Euclidean space to the octilinear architecture, there will be no errors, whereas all other types of edges will cause discrepancies. Since the regions divided by the horizontal, vertical, and orthogonal links are symmetrical, only the grid nodes (x, y) in the first region (i.e. x, and y are integers, $x, y \geq 0$ and $x \geq y$) will be shown. For every grid node (x, y) in the first region, its Euclidean distance to the original (0,0) is $\sqrt{x^2 + y^2}$.

On the other hand, its distance on the octilinear architecture will be

$$(x - d) + (y - d) + d \times \sqrt{2} = (x + y) - (2 - \sqrt{2}) \times d$$

where d is the number of diagonal edges that are taken in any path from (0,0) to (x, y).

Therefore, the shortest distance on the octilinear architecture from (0,0) to (x, y) will be

$$\min((x + y) - (2 - \sqrt{2}) \times d)$$

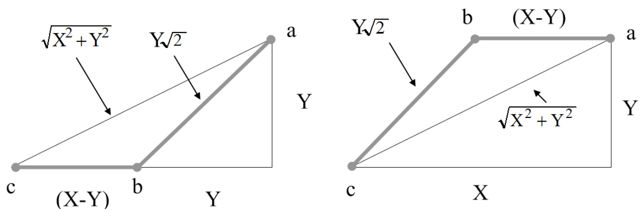


FIGURE 8. Shortest Paths to (x,y) on the octilinear architecture (a) One possible path (b) Another possible path.

which happens when $d = y$. That is, the shortest distance from (0,0) to (x, y) is

$$(x + y) - (2 - \sqrt{2}) \times y = (x - y) + \sqrt{2} \times y$$

Fig. 8 displays two possible shortest paths between (0,0) and (x, y). As a result, the ratio between the shortest distance on the octilinear architecture and the Euclidean distance will be

$$\alpha = \frac{((x\text{-}y) + \sqrt{2} \times y}{\sqrt{x^2 + y^2}}$$

The maximal value of α occurs when $y = (\sqrt{2} - 1)x$, and the value α_{max} is $\sim$1.0842. Since each edge of a minimal spanning tree in the octilinear architecture will be at most 1.0824 times of its original length in the Euclidean space, the overall length of the minimal spanning tree in the octilinear architecture will be at most 1.0824 times of its original length in the Euclidean space. That is,

$$|MST|_E \leq |MST|_X \leq 1.0824|MST|_E.$$

Similarly, it can be proven that

$$|SMT|_E \leq |SMT|_X \leq 1.0824|SMT|_E, \qquad (1)$$

for Steiner minimal trees.

Lemma 2: A Steiner minimal tree connecting a set of nodes in the Euclidean space can be found from the minimal spanning tree of the set of nodes, and the length of the Steiner minimal tree will be at most $2/\sqrt{3}$ times of the length of the minimal spanning tree.

In other words, $|MST|_E \leq 2/\sqrt{3}\ |SMT|_E$.

Proof: It has already been proven by Cieslik [23] and Feng *et al.* [54].

Lemma 3: The length of any Steiner minimal trees in octilinear architecture computed by this algorithm will be equal to or small than the length of the minimal spanning tree for the same set of nodes in octilinear architecture. That is,

$$|SMT|_X^{alg} \leq |MST|_X,$$

where $SMT|_X^{alg}$ be the length of a Steiner minimal tree in octilinear architecture computed by this algorithm.

Proof: Step 2 of this algorithm constructs the minimal spanning tree that connects the set of nodes Z, and then Step 4 identifies a Steiner minimal tree for Z by choosing the node with the maximum positive value (say c_{max}) in the *Improvement Table* as the Steiner vertex. As a result, the length of the new Steiner minimal tree $|SMT|alg\ X$ will be equal to $|MST|_X - c_{max}$.

Consequently, $|SMT|\ alg\ X \leq |MST|_X$.

Theorem 1: The Steiner ratio of the Steiner minimal trees computed by this algorithm is 1.25.

Proof:

$$SMT|_X^{alg} \leq |MST|_X, \quad \text{Lemma 3}$$
$$SMT|_X^{alg} \leq |MST|_X \leq 1.0842|MST|_E, \quad \text{Lemma 1}$$
$$SMT|_X^{alg} \leq 1.0842 \times 2/\sqrt{3}|SMT|_E, \quad \text{Lemma 2}$$

Therefore, $|SMT|alg\ X \leq 1.25|SMT|_E \leq 1.25|SMT|_X$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We have implemented the proposed algorithms in C++ language and all experiments were carried out on a server running Windows 7 with an Intel Core i7 processor (3.5GHz) and 8-gigabyte memory.

A. 2-D 3-D EXAMPLES AND ANALYSIS

Figs. 9$\sim$13 illustrate the experimental results of the proposed algorithms. In all cases, the black cells represent obstacles, gray cells represent terminals, and light gray cells represent branches of SMTs. Figs. 9$\sim$11 show the 2D planes and Figs. 12$\sim$13 are the 3D volumes. Fig. 9 and Fig. 10 are the experimental results of the 4-directional routing. Fig. 11 presents two 8-directional routing results.

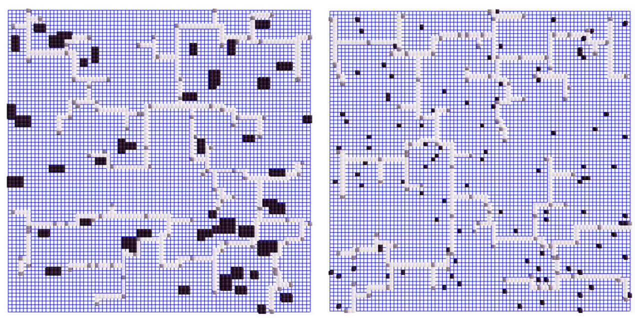


FIGURE 9. 4-directional routing results under 80 × 80 map, 100 terminals and with (a) 50 rectangular obstacles, (b) 100 single-cell obstacles.

Fig. 12 and Fig. 13 present the routing results of 10 and 26 directions, respectively. In the 3D cases, we present small scale routing results to provide a clear view of the diagrams. The space and time complexities are $O(N^2)$ and $O(N^2 + Np^2 \log_2 p)$ respectively, where N is the number of the grid nodes and p is the number of terminal vertices in Z ($|Z| = p$ and $p \leq N$). The results are summarized in Table 3.

B. EXPERIMENTAL RESULTS AND OBSERVATIONS

We compared our algorithms with those presented in [24], [36], [55] in the 2D rectilinear and octilinear planes, PORA [48], PSO [49], FH-OAOS [50], and Chow *et al.* [56],

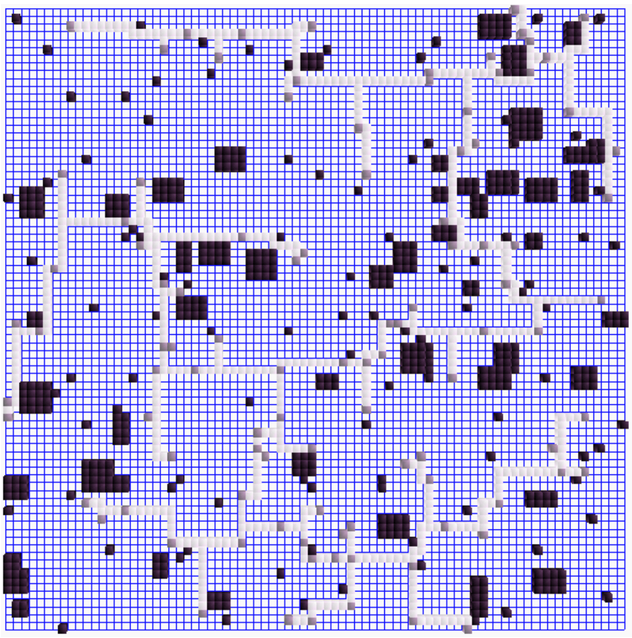


FIGURE 10. 4-directional routing result that combines 50 rectangular obstacles (non-single cell obstacles) and 100 single-cell obstacles.

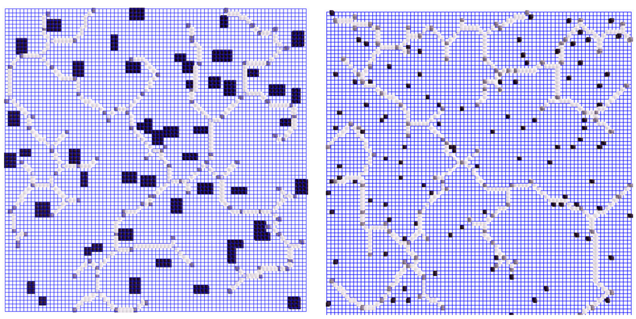


FIGURE 11. Two routing results of the 2D 8-Directional algorithm under 80 × 80 cell map, 100 terminals and (a) 50 rectangular obstacles, (b) 100 single-cell obstacles.

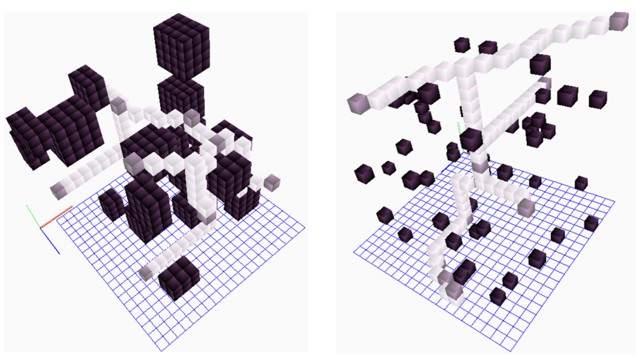


FIGURE 12. Routing results of the 3D 10-directional algorithm under 20 × 20×20 cell volume, 10 terminals and (a) 20 non-single cell obstacles (b) 50 single-cell obstacles.

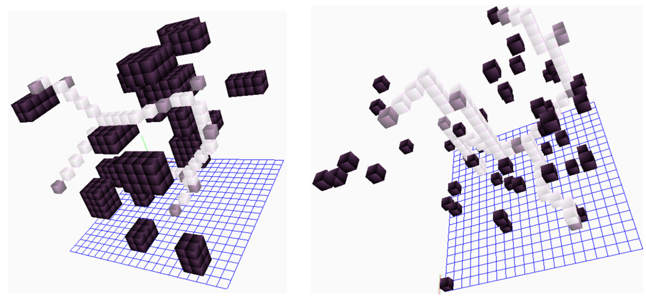


FIGURE 13. Routing results of the 3D 26-directional algorithm under 20 × 20×20 cell volume, 10 terminals and (a) 20 non-single cell obstacles (b) 50 single-cell obstacles.

TABLE 3. Summarization of Figs. 9∼13 in Map size, number of terminals, number of obstacles, number of Steiner points, MST/SMT ratio, and wire length.

Examples	Map Size	Term #	Obs#	Steiner points#	Length
Fig. 9-(a)	80×80	100	50 Rectangle	17	486
Fig. 9-(b)	80×80	100	100 Single-cell	14	516
Fig. 10	80×80	100	50 Rectangle	12	484
Fig. 11-(a)	80×80	100	50 Rectangle	15	393.65
Fig. 11-(b)	80×80	100	100 Single-cell	12	471.88
Fig. 12-(a)	20×20×20	10	20 Rectangle	4	73.113
Fig. 12-(b)	20×20×20	10	50 Single-cell	3	65.213
Fig. 13-(a)	20×20×20	10	20 Rectangle	3	46.445
Fig. 13-(b)	20×20×20	10	50 Single-cell	3	51.517

all works were proposed from 2006 to 2020. There are totally 12 test cases used in [24] and Table 4 lists the total wire lengths of these algorithms. In Table 4, "Term #" is the number of terminals and "Obs#" is the number of obstacles, which were random generated from size 1 × 1 to 200 × 200. "Rec" and "Oct" are the rectilinear and octilinear routings, respectively. Among them, Chow and PSO did not provide the results of C12, while PORA lacked C11 and C12. The percentage of improvement was calculated by ((Other Result – Our Result) / Other Result) × 100%. The respective improvements on the total wire length of rectilinear routing are −11.1% ∼ +9.5% and +0.73% ∼ +60% compared with the algorithms in [24] and [55], −12.9% ∼ + 6.3% in Chow, −17.2% ∼ + 1.79% in PSO, −18.63 ∼ + 2.95 in FH-OAOS, and −9.6% ∼ + 7.56% in PORA. In the octilinear case, the improvements on the total wire length are −9.9% ∼ + 47.1% and −17.4% ∼ + 11.5% compared with [55] and Huang *et al.* [36], −12.48% ∼ + 14.62 in Chow, −15.4% ∼ + 10.26% in PSO, −17.6% ∼ + 11.57% in FH-OAOS, and −9.2% ∼ + 15.77% in PORA. Compared with the latest research in 2020 [48], the proposed approach achieves better performance in most cases of both rectilinear and octilinear routing.

TABLE 4. **Comparison between [55]-Rectilinear, [55]-Octilinear, [24]-Rectilinear, Huang[36], CHOW [56], PSO [49], FH-OAOS [50], PORA [48] and ours on wire length.**

C#	Term#	Obs#	Total Wire Length									
			[55]-Rec	[55]-Oct	[24]-Rec	Huang [36]	[56]	PSO [49]	FH-OAOS [50]	PORA 2020 [48]	Ours-Rec	Ours-Oct
C1	10	10	30,410	27,279	26,900	25,084	25,980	24,717	25,084	26,334	24,343	22,182
C2	30	10	45,640	41,222	42,210	39,488	41,740	40,751	39,488	42,462	40,019	39,962
C3	50	10	58,570	52,432	55,750	54,177	55,500	52,033	54,177	54,722	52,747	51,481
C4	70	10	63,340	57,699	60,350	59,988	60,120	57,250	59,643	60,925	59,913	58,332
C5	100	10	83,150	73,090	76,330	72,833	75,390	72,738	72,738	75,146	73,242	72,710
C6	100	500	149,725	135,454	83,365	78,079	81,340	78,643	77,592	84,030	82,378	80,491
C7	200	500	181,470	162,762	113,260	105,950	110,952	105,542	105,480	11,3056	107,636	106,336
C8	200	800	202,741	182,056	118,747	113,943	115,663	116,204	113,110	11,8277	116,065	115,285
C9	200	1,000	214,850	193,228	116,168	111,258	114,275	111,385	110,642	11,7722	129,078	128,539
C10	500	100	198,010	176,497	170,690	156,329	167,830	157,520	155,579	16,7781	182,290	180,782
C11	1,000	100	250,570	222,758	236,615	216,937	235,866	219,037	216,401	N/A	256,738	244,882
C12	1,000	10,000	1,723,990	1,564,170	789,097	703,669	N/A	N/A	702,544	N/A	833,071	826,189

The results clearly show that our approaches outperform the CG/DT based approaches in many cases, since the proposed algorithms seek the shortest path in the free space instead of the path connected from the source, corners of obstacles to the destination. As described in section 2, the other major drawback of the CG/DT based approaches is that the Steiner vertices must be set in the corners of obstacles; the total wire length grows prohibitively huge as the number of obstacles increases sharply. In C12, there are nearly 2 times improvements to [55] in wire length (1,723,990 vs. 833,071 in rectilinear and 1,564,170 vs. 826,189 in octilinear) since there are 10,000 obstacles.

Furthermore, to handle single cell obstacles with size 1×1, a virtual bound must be attached to the obstacles to at least 2×2, and this also leads to the increase of wire length.

V. CONCLUSION

This paper has proposed an obstacle-avoiding heuristic for the minimal Steiner tree problem for multidimensional rectilinear and octilinear architectures. The 4/8/10/26-directional algorithm is developed by adapting Areibi's concept and Sollin's minimal spanning tree algorithm. The proposed algorithm sequentially adds one potential Hanan vertex to Z and reconstructs the MST to compute the improvement over the old MST. Among this procedure, the HGMR is invoked for λ-geometry shortest path routing. This procedure will continue until the improvement becomes non-positive. Traditionally, the CG/DT-based approaches need local refinement to each triangle connected by any three terminals and Steiner nodes, which may limit the capacity for extending their approaches to higher dimensional environments. Compared to these algorithms, one can replicate our methodology to any λ-geometry environment without complicated refinement, and the point-to-point shortest path is guaranteed. Compared with researches up to 2020, our approach archived up to 1.79%~7.56% improvement on the total wire length in rectilinear cases, and 10.26%~47.1% in octilinear cases.

REFERENCES

[1] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model," *IEEE Trans. Comput.*, vol. 52, no. 9, pp. 1154–1169, Sep. 2003.

[2] J. Cong, A. B. Kahng, and K.-S. Leung, "Efficient algorithms for the minimum shortest path Steiner arborescence problem with applications to VLSI physical design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 1, pp. 24–39, Jan. 1998.

[3] M. Santos, V. López, and G. Botella, "Dyna-*H*: A heuristic planning reinforcement learning algorithm applied to role-playing game strategy decision systems," *Knowl.-Based Syst.*, vol. 32, pp. 28–36, Aug. 2012.

[4] T.-Y. Ho, "PIXAR: A performance-driven X-architecture router based on a novel multilevel framework," *Integration*, vol. 42, no. 3, pp. 400–408, Jun. 2009.

[5] G. Liu, X. Huang, W. Guo, Y. Niu, and G. Chen, "Multilayer obstacle-avoiding X-Architecture Steiner minimal tree construction based on particle swarm optimization," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1003–1016, May 2015.

[6] W. M. Boyce, "An improved program for the full Steiner tree problem," *ACM Trans. Math. Softw.*, vol. 3, no. 4, pp. 359–385, Dec. 1977.

[7] E. J. Cockayne and D. G. Schiller, "Computation of Steiner minimal trees," in *Combinatorics*, D. J. A. Welsh, and D. R. Woodall, Eds. Essex, U.K.: Institute for Mathematics and Application, 1972, p. 5271.

[8] Z. A. Melzak, "On the problem of Steiner," *Can. Math. Bull.*, vol. 4, no. 2, pp. 143–148, 1961.

[9] P. Winter, "Steiner problem in networks: A survey," *Networks*, vol. 17, no. 2, pp. 129–167, 1987.

[10] R. M. Karp, *Reducibility among Combinatorial Problems*. New York, NY, USA: Springer, 1972, pp. 85–103.

[11] P. Korhonen, "An algorithm for transforming a spanning tree into a Steiner tree," *Surv. Math. Program.*, vol. 2, pp. 349–357, Mar. 1979.

[12] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Appl. Math.*, vol. 14, no. 2, pp. 255–265, 1966.

[13] F. Hwang, "An O(n log n) algorithm for suboptimal rectilinear Steiner trees," *IEEE Trans. Circuits Syst.*, vol. 26, no. 1, pp. 75–77, Jan. 1979.

[14] J. M. Smith, D. T. Lee, and J. S. Liebman, "An 0 (N log N) heuristic algorithm for the rectilinear Steiner minimal tree problem," *Eng. Optim.*, vol. 4, no. 4, pp. 179–192, 1980.

[15] T. T. Jing, Z. Feng, Y. Hu, X. L. Hong, X. D. Hu, and G. Y. Yan, "λ-OAT: λ-Geometry obstacle-avoiding tree construction with O(nlogn) complexity," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 11, pp. 2073–2079, Nov. 2007.

[16] J. Long, H. Zhou, and S. O. Memik, "EBOARST: An efficient edge-based obstacle-avoiding rectilinear Steiner tree construction algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 12, pp. 2169–2182, Dec. 2008.

[17] C. Chiang and C. S. Chiang, "Octilinear Steiner tree construction," in *Proc. 45th Midwest Symp. Circuits Syst.*, vol. 1, Aug. 2002, p. I-603.

[18] A. B. Kahng, I. I. Măndoiu, and A. Z. Zelikovsky, "Highly scalable algorithms for rectilinear and octilinear Steiner trees," in *Proc. Conf. Asia South Pacific Des. Autom. (ASPDAC)*, 2003, pp. 827–833.

[19] Q. Zhu, H. Zhou, T. Jing, X.-L. Hong, and Y. Yang, "Spanning graph-based nonrectilinear Steiner tree algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 7, pp. 1066–1075, Jul. 2005.

[20] S. Areibi, M. Xie, and A. Vannelli, "An efficient rectilinear Steiner tree algorithm for VLSI global routing," in *Proc. Can. Conf. Elect. Comput. Eng.*, vol. 2, May 2001, pp. 1067–1072.

[21] S. Vajda, C. Berge, and A. Ghouila-Houri, "Programming, games and transportation networks," *Math. Gazette*, vol. 50, no. 373, p. 347, Oct. 1966.

[22] R. E. Tarjan, *Data Structures and Network Algorithms*, vol. 14. Philadelphia, PA, USA: Society for industrial and Applied Mathematics, 1983.

[23] D. Cieslik, *The Steiner Ratio*. Boston, MA, USA: Kluwer, 2001.

[24] C.-W. Lin, S.-Y. Chen, C.-F. Li, Y.-W. Chang, and C.-L. Yang, "Obstacle-avoiding rectilinear Steiner tree construction based on spanning graphs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 643–653, Apr. 2008.

[25] S. Fortune, "A sweepline algorithm for Voronoi diagrams," *Algorithmica*, vol. 2, nos. 1–4, pp. 153–174, 1987.

[26] T. T. Jing, Y. Hu, Z. Feng, X.-L. Hong, X. Hu, and G. Yan, "A full-scale solution to the rectilinear obstacle-avoiding Steiner problem," *Integration*, vol. 41, no. 3, pp. 413–425, May 2008.

[27] C.-K. Koh and P. H. Madden, "Manhattan or non-Manhattan?: A study of alternative VLSI routing architectures," in *Proc. 10th Great Lakes Symp. VLSI*, Mar. 2000, pp. 47–52.

[28] S. L. Teig, "The X architecture: Not your father's diagonal wiring," in *Proc. Int. Workshop Syst.-Level Interconnect Predict.*, Apr. 2002, pp. 33–37.

[29] J. Xu, X. Hong, T. Jing, Y. Cai, and J. Gu, "An efficient hierarchical timing-driven Steiner tree algorithm for global routing," *Integration*, vol. 35, no. 2, pp. 69–84, Aug. 2003.

[30] D. T. Lee, C. F. Shen, and C. L. Ding, "On Steiner tree problem with 45 routing," in *Proc. Int. Symp. Circuits Syst.*, vol. 3, May 1995, pp. 1680–1682.

[31] Z. Cao, T. Jing, Y. Hu, Y. Shi, X. Hong, X. Hu, and G. Yan, "DraXRouter: Global routing in X-architecture with dynamic resource assignment," in *Proc. Asia South Pacific Conf. Design Automat.*, Jan. 2006, pp. 618–623.

[32] T.-Y. Ho, C.-F. Chang, Y.-W. Chang, and S.-J. Chen, "Multilevel full-chip routing for the X-based architecture," in *Proc. 42nd Design Automat. Conf.*, Jun. 2005, pp. 597–602.

[33] Y. Hu, T. Jing, X. Hong, X. Hu, and G. Yan, "A routing paradigm with novel resources estimation and routability models for X-architecture based physical design," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*. Berlin, Germany: Springer, 2005, pp. 344–353.

[34] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 346–365, Sep. 1961.

[35] J. Dong, H. Zhu, M. Xie, and X. Zeng, "Graph Steiner tree construction and its routing applications," in *Proc. IEEE 10th Int. Conf. ASIC*, Oct. 2013, pp. 1–4.

[36] X. Huang, W. Guo, and G. Chen, "Fast obstacle-avoiding octilinear Steiner minimal tree construction algorithm for VLSI design," in *Proc. 16th Int. Symp. Qual. Electron. Des.*, Mar. 2015, pp. 46–50.

[37] E. Gelenbe, A. Ghanwani, and V. Srinivasan, "Improved neural heuristics for multicast routing," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 2, pp. 147–155, Feb. 1997.

[38] A. Meyer-Bäse, G. Botella, and L. Rybarska-Rusinek, "Stochastic stability analysis of competitive neural networks with different time-scales," *Neurocomputing*, vol. 118, pp. 115–118, Oct. 2013.

[39] F. R. Amoza, "A neural heuristic for access network planning," in *Proc. IEEE Workshop Hybrid Intell. Models Appl.*, Mar. 2009, pp. 31–38.

[40] G. W. Clow, "A global routing algorithm for general cells," in *Proc. 21st Design Automat. Conf.*, Jun. 1984, pp. 45–51.

[41] C. W. Warren, "Fast path planning using modified A* method," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 1993, pp. 662–667.

[42] R. F. Hentschke, J. Narasimham, M. O. Johann, and R. L. Reis, "Maze routing Steiner trees with effective critical sink optimization," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2007, pp. 135–142.

[43] S. Bharathi and K. Alagarsamy, "Steiner tree rectilinear model for data aggregation and sensor node query processing in WSN," *Int. J. Manage., IT Eng.*, vol. 4, no. 10, pp. 65–76, 2014.

[44] J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanità, "Steiner tree approximation via iterative randomized rounding," *J. ACM*, vol. 60, no. 1, pp. 1–33, Feb. 2013.

[45] K. W. Lin, Y. S. Lin, Y. L. Li, and R. B. Lin, "A maze routing-based methodology with bounded exploration and path-assessed retracing for constrained multilayer obstacle-avoiding rectilinear Steiner tree construction," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 23, no. 4, p. 45, 2018.

[46] R.-Y. Wang, C.-C. Pai, J.-J. Wang, H.-T. Wen, Y.-C. Pai, Y.-W. Chang, J. C. M. Li, and J.-H. Jiang, "Efficient multi-layer obstacle-avoiding region-to-region rectilinear Steiner tree construction," in *Proc. 55th Annu. Design Autom. Conf. (DAC)*, 2018, p. 45.

[47] P. P. Saha and T. Samanta, "Obstacle avoiding rectilinear clock tree construction with skew minimization," in *Proc. 27th Int. Conf. VLSI Des. 13th Int. Conf. Embedded Syst.*, Jan. 2014, pp. 387–392.

[48] W. Guo and X. Huang, "PORA: A physarum-inspired obstacle-avoiding routing algorithm for integrated circuit design," *Appl. Math. Model.*, vol. 78, pp. 268–286, Feb. 2020.

[49] X. Huang, G. Liu, W. Guo, Y. Niu, and G. Chen, "Obstacle-avoiding algorithm in X-Architecture based on discrete particle swarm optimization for VLSI design," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 20, no. 2, pp. 1–28, Mar. 2015.

[50] X. Huang, W. Guo, G. Liu, and G. Chen, "FH-OAOS: A fast four-step heuristic for obstacle-avoiding octilinear Steiner tree construction," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 21, no. 3, p. 48, 2016.

[51] X. Huang, W. Guo, G. Liu, and G. Chen, "MLXR: Multi-layer obstacle-avoiding X-architecture Steiner tree construction for VLSI routing," *Sci. China Inf. Sci.*, vol. 60, no. 1, p. 19102, 2017.

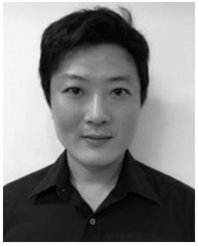
[52] G. E. Jan, K.-Y. Chang, S. Gao, and I. Parberry, "A 4-geometry maze router and its application on multiterminal nets," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 10, no. 1, pp. 116–135, Jan. 2005.

[53] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*. Chelmsford, MA, USA: Courier Corporation, 1976.

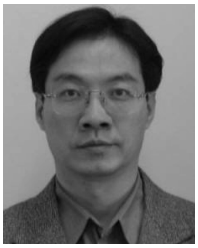
[54] Z. Feng, Y. Hu, T. Jing, X. Hong, X. Hu, and G. Yan, "An O(nlogn) algorithm for obstacle-avoiding routing tree construction in the λ-geometry plane," in *Proc. Int. Symp. Phys. Design*, Apr. 2006, pp. 48–55.

[55] J. H. Rubinstein and J. F. Weng, "Compression theorems and Steiner ratios on spheres," *J. Combinat. Optim.*, vol. 1, no. 1, pp. 67–78, 1997.

[56] W.-K. Chow, L. Li, E. F. Y. Young, and C.-W. Sham, "Obstacle-avoiding rectilinear Steiner tree construction in sequential and parallel approach," *Integration*, vol. 47, no. 1, pp. 105–114, Jan. 2014.



MING CHE LEE received the B.S. degree in computer science from National Taiwan Ocean University, in 2002, and the M.S. and Ph.D. degrees from National Cheng Kung University, Taiwan, in 2003 and 2008, respectively. Since 2011, he has been an Associate Professor of computer and communication engineering with Ming Chuan University, Taoyuan, Taiwan. His research interests are in deep learning, algorithms, and artificial intelligence.



GENE EU JAN (Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, the E.M.B.A. degree from Business School, National Taiwan University, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland. He was a Professor and the Chair with the Institute of Electrical Engineering, National Taipei University. He was an Associate Professor with the Department of Computer Science and Navigation, National Taiwan Ocean University. He is currently the President of the Tainan National University of the Arts. He is also the Board Director of Taiwan Taoyuan International Airport Company Ltd.

CHUNG CHIN LUO received the M.S. degree from the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan, in 2003. His research topics are in path planning and Steiner tree problem on X-Architectures.