Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02659-8

ORIGINAL RESEARCH

Efficient VLSI routing algorithm employing novel discrete PSO and multi-stage transformation

Genggeng Liu[1] · Weida Zhu[1] · Saijuan Xu[2] · Zhen Zhuang[1] · Yeh-Cheng Chen[3] · Guolong Chen[1]

Received: 28 July 2020 / Accepted: 29 October 2020

Abstract
For routing industrial circuits, the Steiner minimal tree (SMT) model can be applied in different routing problems, such as wirelength optimization, congestion reduction, and delay optimization. In this paper, an efficient VLSI routing algorithm employing novel discrete particle swarm optimization (PSO) and multi-stage transformation is proposed to build two types of SMT, including X-architecture Steiner minimal tree and rectilinear Steiner minimal tree. Firstly, to simultaneously handle two types of SMT problems, an effective encoding strategy is proposed to be more suitable for PSO and thus it can overcome the difficulty of designing different algorithms for different routing architectures. Secondly, a multi-stage transformation strategy is presented to expand the search space of the proposed algorithm and accelerate the convergence speed of the proposed algorithm. Various combinations of multi-stage transformation strategies have been tested to highlight the best combination. Furthermore, the various genetic operators combined with union-find data structure strategy are proposed to construct the novel and effective discrete particle update formula. Experimental simulation results on industrial circuits show that the proposed algorithm can get the best solutions among the existing algorithms.

Keywords Particle swarm optimization (PSO) · Multi-stage transformation · Genetic operation · X-architecture Steiner minimal tree · Rectilinear Steiner minimal tree

1 Introduction

Integrated circuit chip design includes mobile communication chip, Internet of things chip, RF chip, wireless connection chip, security chip, TV chip and other fields (Pan et al. 2019b; Chen et al. 2019b). Electronic design automation technology plays a very important role in the field of integrated circuit design, in which the routing problem is the key step to determine the final performance of the chip. The Steiner minimal tree (SMT) model can be applied in different routing problems, such as wirelength optimization, congestion reduction, and delay optimization. At present, most routing algorithms on industrial circuits were proposed for rectilinear architecture (Zhang et al. 2020; Liu et al. 2019; Siddiqi and Sait 2017). Those routing algorithms optimized the wire length and delay by optimizing the Steiner tree topology, changing the width of metal wire, inserting buffer and so on (Held et al. 2017; Dai et al. 2009, 2011; Chang et al. 2010; Liu et al. 2013). With the development of VLSI technology, interconnect effect is becoming the main challenge of the performance of chip.

However, the routing direction for rectilinear architecture is only horizontal and vertical. As a result, the optimization capability of rectilinear routing algorithm is limited, which leads to its limited ability to optimize chip performance. Therefore, it is necessary to change the traditional Manhattan architecture from the fundamental point of view, so that researchers begin to try to use the non Manhattan architecture as the basic model to optimize the overall performance of the chip (Thurber and Xue 1999; Coulston 2003; Chiang and Chiang 2002).

In the non Manhattan architecture, it can further reduce the chip area, shorten the wire length, reduce the number of vias and power consumption, and improve many performance indicators in physical design (Samanta et al. 2006,

✉ Saijuan Xu
feature19870206@163.com

[1] College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou, China

[2] Department of Information Engineering, Fujian Business University, Fuzhou, China

[3] Department of Computer Science, University of California, Davis, CA, USA

Published online: 13 November 2020

2011; Zhu et al. 2005; Yan 2008; Koh and Madden 2000). The proposed non Manhattan architecture brings new opportunities and challenges for VLSI circuit physical design. It leads to the update of the whole layout field algorithm, that is, it puts forward new requirements for layout planning and layout, routing, parameter extraction and other aspects in the physical design process based on non Manhattan architecture.

The minimal Steiner tree problem is an NP-hard problem (Garey and Johnson 1977). Some computational intelligence methods have shown good application prospects in dealing with NP hard problems (Wu et al. 2019b; Chen et al. 2019a; Pan et al. 2019a; Ma et al. 2020; Kundu et al. 2016, 2017). Especially, evolutionary algorithms are used to solve the two types of NP-hard problems including rectilinear Steiner minimal tree (RSMT) problem (Manna et al. 2015; Bhattacharya et al. 2014; Guo and Huang 2020; Liu et al. 2011) and the X-architecture Steiner minimal tree (XSMT) problem (Guo and Huang 2020; Liu et al. 2012). As a swarm-based evolutionary method, particle swarm optimization (PSO) proposed by Eberhart and Kennedy (1995) is a powerful optimization tool, which has the advantages of quick convergence, simple implementation, and powerful global optimization capability. Therefore, in recent years, more and more PSO algorithms are used to solve the NP-hard problem and achieve good results (Gandomi and Kashani 2017; Liu et al. 2014, 2015a, b, 2020; Jia et al. 2017; Guo et al. 2014; Yazdaninejadi et al. 2018; Feng et al. 2019).

For this reason, an efficient VLSI routing algorithm employing novel discrete PSO and multi-stage transformation (RA-PSO-MST) is proposed to construct two types of SMT, including non-Manhattan SMT and Manhattan SMT. The main contributions of this paper can be summarized as follows.

(1) An effective edge-vertex encoding strategy with four types of pseudo-Steiner point choice, which is more suitable for the proposed PSO algorithm, is proposed for the non-Manhattan Steiner tree. Furthermore, when the proposed encoding strategy adopts two types of pseudo-Steiner point choice, it can be effectively extended to construct the Manhattan Steiner tree. Therefore, it can effectively help the algorithm construct two types of SMT.
(2) A multi-stage transformation strategy is proposed to both expand the algorithm search space and ensure the effective convergence. We have tested three types from two to four stages and various combinations with each type to highlight the best combination for the proposed algorithm. Therefore, it can construct SMT with less wirelength.
(3) To bring uncertainty and diversity into the proposed PSO algorithm, four genetic operators are proposed. Genetic operators also help the proposed algorithm to address the discrete VLSI routing problem for industrial circuits.
(4) A series of experiments on industrial circuits (Alpert 1998; Zachariasen 2003) are designed to illustrate the feasibility and effectiveness of the proposed strategies and can achieve better solution with less runtime.

The remainder of this paper is organized as follows. In Sect. 2, the related work is described. In Sect. 3, the problem formulation is described. In Sect. 4, the implementation details of RA-PSO-MST algorithm are described. Several comparisons and experimental results are discussed in Sect. 5. Section 6 concludes this paper.

2 Related work

At present, the research on VLSI routing mainly focuses on rectilinear architecture and non Manhattan architecture. Next, we will introduce the research progress of these two types of work.

2.1 Routing in rectilinear architecture

The global routing problem of physical design is becoming more and more complex. The research work of the global routing problem is mainly focused on the Steiner minimal tree construction and the global routing algorithm in rectilinear architecture.

Aiming at the layer assignment problem of global routing in advanced manufacturing process, a delay driven layer allocation algorithm considering coupling effect and congestion was proposed (Zhang et al. 2020), and the proposed algorithm achieved the best routing quality and running time compared with other similar work. In view of the fact that routability has become a very important challenge in IC design, a routability driven track assignment algorithm was proposed, which can effectively reduce the mismatch between global routing and detailed routing (Liu et al. 2019). In order to further improve the quality of existing routing algorithms, a game theory algorithm as a post-processing step of the global routing algorithm was proposed to achieve good routing results. In order to further consider the static delay constraint in the global routing stage, based on the minimum maximum resource sharing model, Siddiqi and Sait (2017) dynamically adjusted the delay cost and balanced the wirelength congestion to optimize the delay. In order to find a routing solution with less overflow, i.e. high routability, two effective strategies were proposed in Held et al. (2017), including adaptive pseudo-random network scheduling routing strategy and evolution based rewiring strategy.

Especially in the VLSI global routing algorithm competition held in ISPD, NCTU-R (Dai et al. 2009), NTHU-Route2.0 (Chang et al. 2010), NCTU-GR (Dai et al. 2011), NCTU-GR2.0 (Liu et al. 2013) and other global routers have emerged, and stood out in the global routing competition. In the current process, if the number of metal layers is multi-layer, the above global router can be used for multi-layer routing. According to the different multi-layer routing methods, it can be divided into two categories: complete 3-D global routing and plane global routing close layer allocation (2.5-D global routing). The complete 3-D global routers, compared with other 2.5-D global routers can achieve more accurate global routing solution in some optimization objectives, but it takes a lot of time and space complexity.

2.2 Routing in non-Manhattan architecture

However, the the above optimization researches of wirelength and delay based on rectilinear architecture can not make full use of the routing area due to its limited routing direction, which leads to excessive redundancy of interconnection resources. Therefore, more and more attention has been paid to the routing model of non-Manhattan architecture which allows more routing directions to make full use of chip resources, and ultimately optimize chip performance more effectively.

Non-Manhattan routing mainly includes Y-architecture routing and X-architecture routing. In Thurber and Xue (1999), a method based on branch and bound was proposed to construct the Y-architecture Steiner minimum tree, which enables the proposed algorithm to construct an accurate Y-architecture Steiner minimum tree on small-scale circuits. In Coulston (2003), an exact algorithm and a series of pruning strategies were used to construct an X-architecture Steiner minimum tree. A 10% reduction in wirelength was achieved compared with the minimum rectilinear Steiner tree, but the cost of algorithm complexity is high. Due to the high time complexity of the precise algorithm, researcher began to explore the application of heuristic strategy in the construction of non-Manhattan architecture Steiner tree . In Chiang and Chiang (2002), an effective algorithm was proposed to construct X-architecture Steiner minimum trees, but it is only limited to constructing isomorphic X-architecture Steiner minimum tree. In Samanta et al. (2006), a heuristic algorithm was proposed to solve Y-architecture Steiner minimum tree and the algorithm was based on greedy idea. In Zhu et al. (2005), two algorithms for constructing X-architecture Steiner minimum trees based on octagonal spanning graph were proposed based on greedy strategy. Yan (2008) proposed a modified algorithm for the construction of delay driven X-architecture Steiner minimum tree, which can effectively reduce the delay of the longest path by adjusting the Y-shaped routing of the longest path. On the basis of Hanan grid and Elmore delay model, Samanta et al. (2011) constructed a delay driven Y-architecture Steiner tree. Based on graph theory, Koh and Madden (2000) used k-idea algorithm to generate multiple groups of routing solutions, and introduced the concept of branch and bound to accelerate the solution.

The above researches on non-Manhattan Steiner tree (Thurber and Xue 1999; Coulston 2003; Chiang and Chiang 2002; Samanta et al. 2006, 2011; Zhu et al. 2005; Yan 2008; Koh and Madden 2000) are both based on exact algorithms and traditional heuristic algorithms. However, the time complexity of exact algorithms increases exponentially with the scale of the problem. Most of the traditional heuristic algorithms are based on greedy strategies and easy to fall into local optima. The Steiner tree construction methods based on exact algorithms and traditional heuristic algorithms did not make full use of the geometric properties of non-Manhattan architecture and cannot guarantee the quality of the Steiner tree. Those algorithms provided less suitable method for the topology optimization and therefore it is not satisfactory in terms of time, wirelength, congestion, and delay. Therefore, this paper uses PSO algorithm which shows good application prospects in NP-hard problems to construct high-quality Steiner minimum tree.

3 Problem formulation

The notations used in this paper are listed in Table 1.

XSMT problem Given a set of n pins $\{P = P_1, P_2, P_3, \ldots, P_n\}$ and each P_i represented by a coordinate of (x_i, y_i), the XSMT problem is to construct an X-architecture Steiner minimal tree to connect the pins in

Table 1 Notations in this paper

Notation	The meaning of each notation
n	Number of pins in each net
P_i	The given pin with number i
x_i	The X-coordinate of P_i
y_i	The Y-coordinate of P_i
e_i	An edge in the global routing graph
$l(\mathrm{e}_i)$	The length of segment e_i in tree T_x
N_1	The mutation operator
N_2 and N_3	The crossover operators
w	Inertia weight factor
c_1 and c_2	Two types of acceleration factors
C_p	The crossover operator between the personal optimal particle and the particles
C_g	The crossover operator between the global optimal particle and the particles

Table 2 The coordinate information of pins

Pin. no.	1	2	3	4	5	6	7	8
X-coordinate	33	2	42	47	34	38	37	20
Y-coordinate	33	9	35	2	1	2	5	4

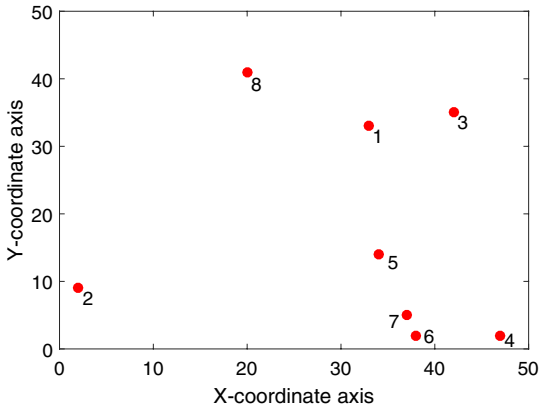

Fig. 1 A given net with eight pins

P by introducing some Steiner points. The routing direction of XSMT including 45^0 direction, 135^0 direction, horizontal direction and vertical direction. In Fig. 1, there are eight pins for the given net and the coordinate information of pin is shown in Table 1. For example, the coordinate information of Pin 3 in Fig. 1 is (42, 35) shown in Table 2.

3.1 Definitions

Definition 1 (*pseudo-Steiner point*) For convenience, we assume that the connection point except for the pins is called the pseudo-Steiner point ('PS' represents 'the pseudo-Steiner point' in subsequent content). In Fig. 2, the point *S* is PS, and PS contains the Steiner point.

Definition 2 (0 *Choice*) In Fig. 2a, let $A = (x_1, y_1)$ and $B = (x_2, y_2)$ be the two endpoints of a line segment *L*, where $x_1 < x_2$. The 0 Choice of *L* is given in Fig. 2b, which first from *A* leads rectilinear side to pseudo-Steiner point *S* and then leads octagonal side to *B*.

Definition 3 (1 *Choice*) The 1 Choice of *L* is given in Fig. 2b, which first from *A* leads octagonal side to pseudo-Steiner point *S* and then leads rectilinear side to *B*.

Definition 4 (2 *Choice*) The 2 Choice of *L* is given in Fig. 2c, which first from *A* leads vertical side to pseudo-Steiner point *S* and then leads horizontal side to *B*.

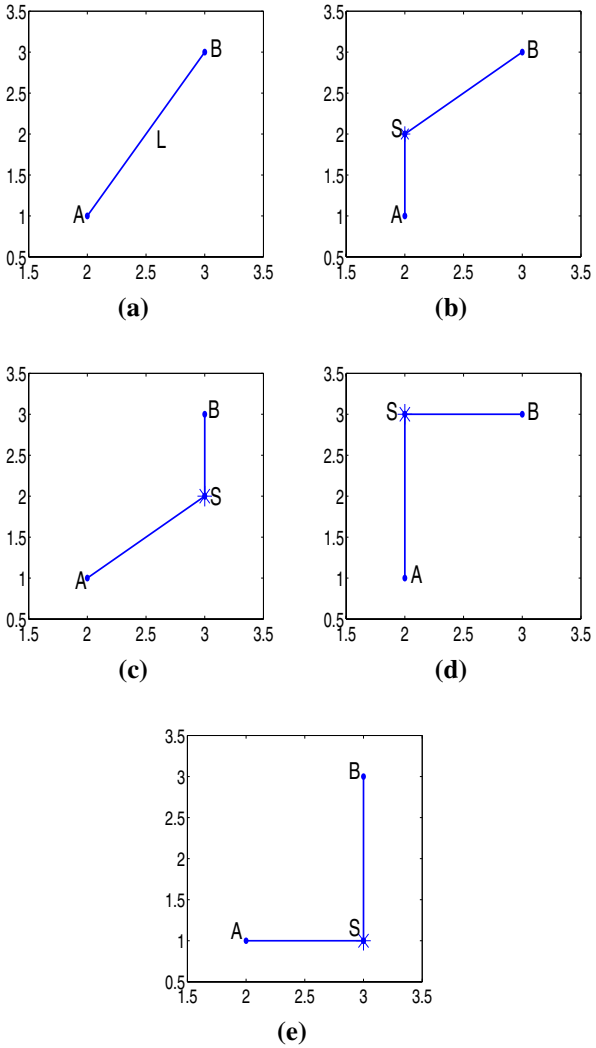

Fig. 2 Four options of Steiner point for the given line segment. **a** Line segment *L*, **b** 0 Choice, **c** 1 Choice, **d** 2 Choice, **e** 3 Choice

Definition 5 (3 *Choice*) The 3 Choice of *L* is given in Fig. 2d, which first from *A* leads horizontal side to pseudo-Steiner point *S* and then leads vertical side to *B*.

4 Details of RA-PSO-MST algorithm

The XSMT problem is a discrete problem, while the original PSO is designed for continuous problems (Chen et al. 2020; Tang et al. 2020; Huang et al. 2015). Therefore, it

is necessary to design an effective discrete mechanism so that the proposed PSO in this paper can effectively solve the XSMT problem. In view of our previous research work, PSO has been effectively discretized and applied to a series of discrete problems such as VLSI routing (Huang et al. 2013, 2016, 2017). Therefore, this paper proposes an effective algorithm, RA-PSO-MST, to solve the XSMT problem.

4.1 Encoding strategy

One candidate Steiner tree is represented as lists of spanning tree edges and each edge augments with a pseudo-Steiner point ('PS' represents 'the pseudo-Steiner point' in subsequent content) choice which specifies the transformation from the spanning tree edge to the octagonal edge. Each pseudo-Steiner point choice, includes four types as shown in Liu et al. (2014) and the value is 0, 1, 2 or 3 which denotes 0 Choice, 1 Choice, 2 Choice or 3 Choice, respectively.

If a net has n pins, a spanning tree would have n-1 edges and one extra digit which is the particle's fitness. Besides, two digits represent the two vertices of each edge and another one represents the pseudo-Steiner point choice for each edge, so the length of one particle is $3(n-1)+1$.

Property 1 *An edge-vertex encoding strategy with four types of PS point Choice is more effective for XSMT than the one with two types of PS point Choice.*

The encoding strategy with two types of PS point Choice, i.e. including 0 Choice and 1 Choice. In contrast, the encoding strategy with four types of PS point Choice is applied in the proposed algorithm. And the algorithm with four types of PS point Choice can reduce more wirelength than the algorithm with two types of PS point Choice. The reason is that the existence of the last two pseudo-Steiner point choices may overlap with the first two choices with less wirelength. For example, the SMT in Fig. 3b, have the more overlapping location than the one in Fig. 3a. Thus the wirelength (15.070) of SMT in Fig. 3b is less than the wirelength (19.140) of SMT in Fig. 3a. Generally, this situation could frequently exist among the net with more pins. From Table 3, it can find that the encoding strategy with four pseudo-Steiner point choices can achieve 1.05% wirelength reduction. Meanwhile, for the circuits with large scale, it can reduce more wirelength with the proposed encoding strategy. Specifically, the term "Imp" of Table 3 and other tables in this paper represents the improvement rate of relevant algorithms or strategies.

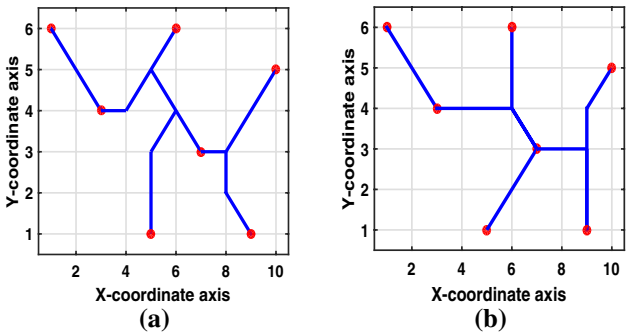

Fig. 3 Two kinds of encoding strategies for the potential impact on the wirelength reduction. **a** The encoding strategy with two pseudo-Steiner point choices, **b** the encoding strategy with four pseudo-Steiner point choices

Property 2 *An edge-vertex encoding strategy for the XSMT problem can be effectively extended to solve the RSMT problem.*

If the edge-vertex encoding strategy for the XSMT problem is designed to only contain two kinds pseudo-Steiner point choices, i.e. 2 Choice and 3 Choice, then the proposed algorithm can effectively solve the RSMT problem. Therefore, the proposed encoding strategy is helpful to effectively extend the proposed RA-PSO-MST algorithm to solve the RSMT problem. And this situation has been experimentally verified in latter section.

4.2 Fitness function

Definition 6 The wirelength of X-architecture Steiner tree is the sum of the lengths of all the edge fragments in the Steiner tree and the calculation function of wirelength is as follows.

$$L(T_X) = \sum_{e_i \in T_X} l(e_i) \tag{1}$$

Table 3 Comparison between the encoding strategy with four pseudo-Steiner point choices and the one with two choices

Circuit	Pin#	Two choices	Four choices	Imp (%)
1	8	16,951	16,900	0.30
2	9	18,041	18,023	0.10
3	10	19,435	19,397	0.20
4	20	32,303	32,038	0.82
5	50	48,469	47,875	1.23
6	70	56,397	55,902	0.88
7	100	69,205	68,454	1.09
8	410	142,098	139,254	2.00
9	500	155,146	151,942	2.07
10	1000	221,063	217,067	1.81
Average				1.05

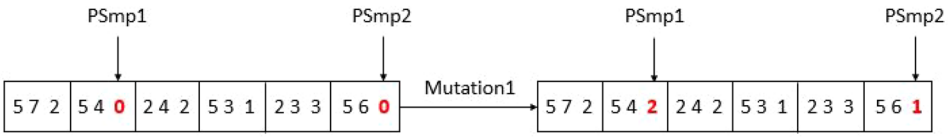

Fig. 4 Mutation operator with PS transformation

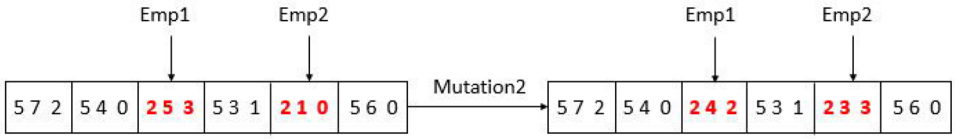

Fig. 5 Mutation operator with E transformation

where $l(e_i)$ represents the length of segment e_i in tree T_X. Then the particle fitness function of the algorithm is designed as follows.

$$fitness = L(T_X) \tag{2}$$

4.3 Particle update formula

The redefined particle update formula in this paper is shown below.

$$X_i^t = N_3(N_2(N_1(X_i^{t-1}, w), c_1), c_2) \tag{3}$$

where N_1 represents the mutation operator. N_2 and N_3 represent the crossover operators. w represents inertia weight factor. c_1 and c_2 represent two types of acceleration factors.

4.3.1 Mutation operator

For different problems, there are many ways to implement mutation operation. There are two kinds of mutation operators in our previous work, including the PS transformation strategy (Liu et al. 2012) in Fig. 4 and the E transformation strategy in Fig. 5.

For the XSMT construction (Liu et al. 2012), it proposed the mutation operator with PS transformation strategy and each PS selection method contains four choices. Based on the PS transformation strategy, we find that the proposed algorithm takes the wirelength as the optimization objective, and obtains a better wirelength optimization rate relative to RSMT.

Property 3 *If the mutation operator with PS transformation strategy is designed, the optimization ability of XSMT construction algorithm is limited.*

The algorithm adopting the mutation operator with PS transformation strategy is an algorithm based on MST to construct the SMT. For the rectilinear routing architecture, the wirelength of Steiner minimal tree based on MST construction may be 3/2 times than the wirelength of exact solution. Thus the wirelength of Steiner minimal tree based on MST construction is a little too long compared with the exact solution.

Therefore, in the process of constructing SMT, only if the PS transformation is adopted, the wirelength of Steiner minimal tree may be 3/2 times than the wirelength of exact solution and the solution space of the algorithm may not contain the optimal value. Thereby the new and effective transformation strategies need to be redesigned. Then another basic transformation strategy, namely edge transformation (E transformation), is introduced.

The updated operation of PSO for constructing XSMT (Liu et al. 2012) only adopts the pseudo-Steiner transformation. However, in the process of building rectilinear Steiner minimal tree or X-architecture Steiner minimal tree, the topology of the routing tree is changed. Therefore, in the particle update process of rectilinear Steiner minimal tree or X-architecture Steiner minimal tree construction algorithms, both edge transformation and pseudo-Steiner transformation need to be adopted. Furthermore, the proposed algorithm with edge transformation strategy can further reduce the wirelength of SMT. As shown in Table 4, the proposed

Table 4 Comparison among RSMT, E and PS transformation

Circuit	RSMT	PS	E	Imp PS-RSMT (%)	Imp E-RSMT (%)
1	17,928	16,918	16,900	5.63	5.73
2	20,478	18,041	18,023	11.90	11.99
3	21,969	19,696	19,397	10.35	11.71
4	35,675	32,193	32,144	9.76	9.90
5	53,518	48,208	48,057	9.92	10.20
6	62,633	56,360	56,341	10.02	10.05
7	75,584	68,626	68,618	9.21	9.22
8	158,206	141,316	141,917	10.68	10.30
9	170,924	153,832	155,073	10.00	9.27
10	246,731	220,427	221,826	10.66	10.09
Average				9.81	9.85

algorithm with edge transformation can reduce the smaller wirelength than the one with pseudo-Steiner transformation.

Property 4 *The fact is not that the more points the operator mutates, the better the convergence effect of the algorithm. Furthermore, based on a lot of experiments, it can find that two-point mutation operator is more effective for the proposed algorithm than others.*

To find more effective mutation operator for our algorithm, we also test the single point mutation operator, two-point mutation operator, three-point mutation operator, and four-point mutation operator (Wu et al. 2019a). From Table 5, we can find that two-point mutation operator achieve the best wirelength. Therefore, two-point mutation operator is adopted in the proposed algorithm.

With the proposed best mutation operator, the mutation operator of particles is shown below.

$$W_i^t = N_1(X_i^{t-1}, mp) = \begin{cases} M(X_i^{t-1}), & r_1 < mp \\ X_i^{t-1}, & \text{otherwise} \end{cases} \quad (4)$$

where M represents the mutation operator and mp represents the corresponding mutation probability.

Since invalid solutions may occur in the iteration process after the introduction of edge transformation operation, it is necessary to introduce union-find data structure strategy in mutation operation to ensure that the mutated spanning tree is effective. As shown in Fig. 6, after the mutation operator is executed, one edge is randomly selected as the mutation edge in the spanning tree. To ensure that the mutated spanning tree is valid, union-find data structure strategy records the point set of two subtrees after deleting the mutated edge. Then, a new spanning tree is constructed as an effective solution by selecting one point from the point set of two subtrees as two endpoints of the new edge. As shown in Fig. 6, M_1 represents the mutated edge, M_2 represents the new edge after mutation. The pseudo code of mutation operator is shown in Algorithm 1.

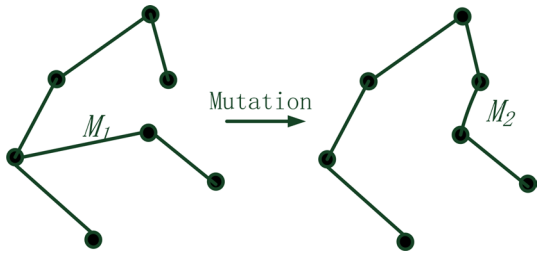

Fig. 6 Mutation operation with the union-find data structure strategy

Algorithm 1 Mutation operator(p)

Input: Particle p
Output: New particle
  Initialize each pin's partition to singletons
  $r = random(1, n-1)$; //n is the number of pins
  for *each edge* e_i *of* p **do**
    if $e_i \neq e_r$ **then**
      $Union\_partition(u, v)$;//u and v is endpoint of e_i, u and v are merged into the same set
    end if
  end for
  while *true* **do**
    $p_1 = random(1, n-1)$;// generate a random number between (1,n-1)
    $p_2 = random(1, n-1)$;
    if $Find\_set(p_1) \neq Find\_set(p_2)$ **then**
      $Union\_partition(p_1, p_2)$;//$p_1$ and p_2 are not in the same set
      $generate\_edge(p_1, p_2)$;
      break;
    end if
  end while

Table 5 Comparison among four mutation operators

Circuit	RSMT (%)	1-M (%)	2-M (%)	3-M (%)	4-M (%)
1	17,928	5.70	5.74	5.74	5.70
2	20,478	11.99	11.99	11.99	11.99
3	21,969	11.71	11.71	11.71	11.71
4	35,675	10.05	10.21	10.14	10.09
5	53,518	10.46	10.50	10.42	10.48
6	62,633	10.74	10.95	10.49	10.58
7	75,584	9.66	9.84	9.79	9.77
8	158,206	11.62	11.74	11.63	11.52
9	170,924	10.85	10.89	10.85	10.84
10	246,731	11.93	11.99	11.92	11.85
Average		10.47	10.56	10.47	10.45

4.3.2 Crossover operator

Similar to the mutation operator, there are two kinds of crossover operator, shown in Figs. 7 and 8, respectively. In Fig. 7, the intersection point is only included in the PS point, while the intersection point is included in both the PS point and the edge location in Fig. 8.

With the introduction of crossover operator, the cognitive personal experience of particles is shown below.

$$S_i^t = N_2(W_i^t, cp_1) = \begin{cases} C_p(W_i^t), & r_2 < cp_1 \\ W_i^t, & \text{otherwise} \end{cases} \tag{5}$$

where C_p represents the crossover operator between the personal optimal particle and the particles. cp_1 represents the corresponding crossover probability.

With the introduction of crossover operator, the cooperative global experience of particles is shown below.

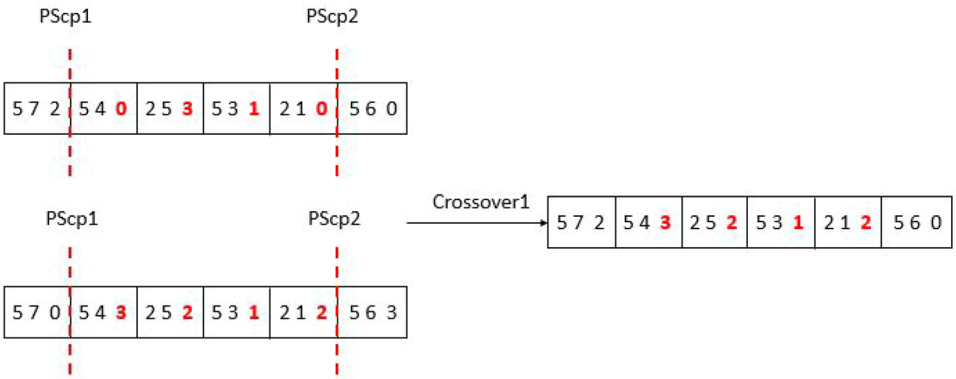

Fig. 7 Crossover operator with PS transformation

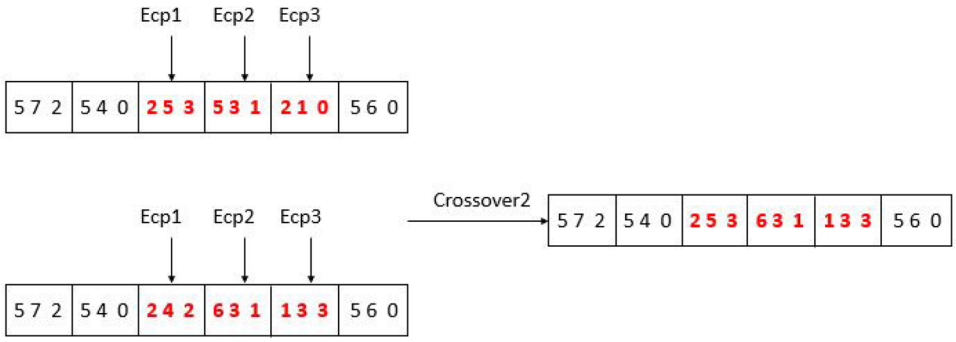

Fig. 8 Crossover operator with E transformation

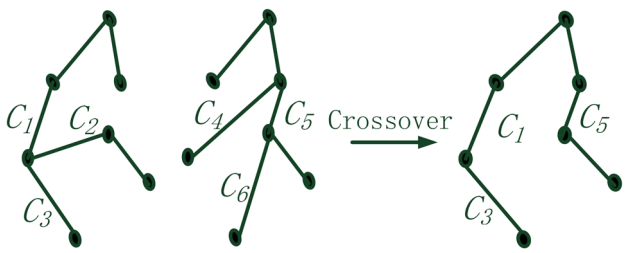

Fig. 9 Crossover operation with the union-find data structure strategy

$$X_i^t = N_3(S_i^t, cp_2) = \begin{cases} C_g(S_i^t), & r_3 < cp_2 \\ S_i^t, & \text{otherwise} \end{cases} \tag{6}$$

where C_g represents the crossover operator between the global optimal particle and the particles. cp_2 represents the corresponding crossover probability.

As shown in Fig. 9, when implementing the crossover operator, we sort the edges of two Steiner trees from small to large according to the number order of pins, and then

use union-find data structure strategy to select the common edges of the two Steiner trees as an edge set S and the remaining edges of two Steiner trees as the other edge set D. Then we randomly select an edge from edge set D and add it to edge set S until we construct an effective new Steiner tree. Here we use union-find data structure strategy to ensure that the new Steiner tree does not have a ring. As shown in Fig. 9, the remaining edge set $D = C_1, C_2, C_3, C_4, C_5, C_6$, and the new edge set C_1, C_3, C_5 denotes the selected edges from edge set D which are the new edges of the new Steiner tree. The pseudo code of crossover operator is shown in Algorithm 2.

Algorithm 2 Crossover operator(p, q)

Input: Particle p and q
Output: New particle
Initialize each pin's partition to singletons
$Sort\_edge(p, u)$;//sort edge of p according to the serial number of the first endpoint u
$Sort\_edge(p, v)$;//sort edge of p according to the serial number of the second endpoint v
$Sort\_edge(q, u)$;
$Sort\_edge(q, v)$;
$set1 = Selecct\_same\_edge(p, q)$;
$set2 = Select\_different\_edge(p, q)$;
$Union\_partition(u, v, set1)$;//merge each edge of set1
$New\ particle = Generate\_edge(set1)$;
while *New particle is not a complete tree* **do**
  $L(u, v) = Random\_select_edge(set2)$;
  if $Find\_set(u) \neq Find\_set(v)$ **then**
    add $L(u, v)$ *to New particle*;
    $Union\_partition(u, v, L)$;
  end if
end while

4.4 Multi-stage transformation

The pseudo-Steiner transformation strategy and edge transformation strategy show their respective advantages in different benchmark circuits. Therefore, we intend to combine the two transformation strategies in different ways, and carry out experimental verification on different benchmark circuits to find the best multi-stage transformation.

Firstly, we divide the iterative process of the algorithm into two stages, including the first half and the second half, namely two-stage transformation strategy. In Table 6, we try to implement all the combinations of transformations, including E:E, E:PS, PS:E, PS:PS. In Table 6, for CM1 and CM4, the transformation strategy of the first half and the second half is the same. CM4 of Table 6 is PS transformation, and CM1 of Table 6 is E transformation.

Table 6 Comparison among various combinations of the two-stage transformation strategy

Number	First half	Second half	Imp RSMT (%)
CM1	E	E	9.75
CM2	E	PS	9.91
CM3	PS	E	9.85
CM4	PS	PS	9.26

Table 7 Comparison among various combinations of the three-stage transformation strategy

Number	First	Second	Third	Imp RSMT (%)
CM1	E	E	E	9.73
CM2	E	E	PS	10.42
CM3	E	PS	E	10.44
CM4	E	PS	PS	10.45
CM5	PS	E	E	10.21
CM6	PS	E	PS	10.46
CM7	PS	PS	E	10.48
CM8	PS	PS	PS	10.24

Table 8 Comparison among various combinations of the four-stage transformation strategy

Number	First	Second	Third	Fourth	Imp RSMT (%)
CM1	E	E	E	E	9.73
CM2	E	E	E	PS	10.43
CM3	E	E	PS	E	10.54
CM4	E	E	PS	PS	10.48
CM5	E	PS	E	E	10.45
CM6	E	PS	E	PS	10.56
CM7	E	PS	PS	E	10.44
CM8	E	PS	PS	PS	10.43
CM9	PS	E	E	E	10.15
CM10	PS	E	E	PS	10.48
CM11	PS	E	PS	E	10.52
CM12	PS	E	PS	PS	10.49
CM13	PS	PS	E	E	10.43
CM14	PS	PS	E	PS	10.52
CM15	PS	PS	PS	E	10.48
CM16	PS	PS	PS	PS	10.23

Property 5 *The edge transformation strategy can expand the search space of the proposed algorithm, while the pseudo-Steiner transformation strategy can accelerate the convergence speed of the proposed algorithm.*

E:PS transformation strategy, known as CM2, represents that edge transformation strategy is introduced in the first half iterative process of the algorithm, and pseudo-Steiner transformation strategy is introduced in the second half iterative process of the algorithm. CM2 can reduce 9.91% wirelength than the rectilinear Steiner minimal tree construction algorithm. CM2 is the best combination for the two-stage transformation strategy in Table 6.

Secondly, the iterative process of the algorithm is divided into the first stage, the second stage, and the third stage, namely three-stage transformation. PS:PS:E transformation strategy, known as CM7 can reduce 10.48% wirelength than the rectilinear Steiner minimal tree construction algorithm, as shown in Table 7. CM7 is the best combination for the three-stage transformation strategy in Table 7.

Thirdly, the the iterative process of the algorithm is divided into the first stage, the second stage, the third stage, and the fourth stage, namely four-stage transformation. E:PS:E:PS transformation strategy, known as CM6 can reduce 10.56% wirelength than the rectilinear Steiner minimal tree construction algorithm, as shown in Table 8. CM6 is the best combination for the four-stage transformation strategy in Table 8.

Finally, we also test five-stage transformation, six-stage transformation, and then the best combination of these transformations is not better than the best one of four-stage transformation. The simulation results of the experiment show that the algorithm does not always get better solution when the algorithm is divided into more stages. Therefore, we select the best four-stage transformation for the proposed multi-stage transformation.

Table 10 The effectiveness of the RSMT construction algorithm based on the proposed four-stage transformation

Circuit	Pin#	RSMT	MRSMT	Imp RSMT (%)
1	8	17,928	17,693	1.31
2	9	20,478	19,797	3.33
3	10	21,969	21,143	3.76
4	20	35,675	34,827	2.38
5	50	53,518	51,631	3.53
6	70	62,633	60,297	3.73
7	100	75,584	74,663	1.22
8	410	158,206	153,050	3.26
9	500	170,924	165,664	3.08
10	1000	246,731	237,666	3.67
Average				2.93

Table 9 Comparison results among various SMT algorithms

Circuit	Pin#	[29]	[30]	RA-PSO-MST	Imp [29] (%)	Imp [30] (%)
1	8	17,928	16,918	16,900	5.73	0.11
2	9	20,478	18,041	18,023	11.99	0.10
3	10	21,969	19,696	19,397	11.71	1.52
4	20	35,675	32,207	32,038	10.19	0.52
5	50	53,518	48,020	47,875	10.54	0.30
6	70	62,633	56,433	55,902	10.75	0.94
7	100	75,584	68,855	68,454	9.43	0.58
8	410	158,206	141,894	139,254	11.98	1.86
9	500	170,924	154,825	151,942	11.11	1.86
10	1000	246,731	221,090	217,067	12.02	1.82
Average					10.55	0.96

5 Experimental results

5.1 Validation of multi-stage transformation

Based on the best threshold parameter, the detail comparison among our proposed algorithm and the two similar algorithms (Liu et al. 2011, 2012) is given in Table 9. We can see that the constructed XSMT in this paper achieves the 0.96% wirelength reduction compared with the constructed XSMT (Liu et al. 2012) and the 10.55% wirelength reduction compared with the constructed RSMT (Liu et al. 2011). The author pointed out that the reduction rate of XSMT length to RSMT length is generally $9.75 \pm 2.29\%$ (Coulston 2003). And the average reduction rate of our algorithm is 10.55%, which is within the optimum rate range.

Furthermore, for the large-scale problems, such as Circuits 8, 9, and 10, our algorithm exhibits stronger optimization capability. Therefore, our algorithm (RA-PSO-MST) has more advantages in solving large-scale problems.

5.2 Validation of being extended to rectilinear architecture

We further extend the efficient multi-stage transformation proposed in this paper to the rectilinear architecture, and then the rectilinear Steiner minimal tree construction algorithm with the proposed multi-stage transformation is proposed. In Table 10, the experimental results are listed. 'MRSMT' represents the rectilinear Steiner minimal tree construction algorithm with the proposed multi-stage transformation, while 'RSMT' indicates that the multi-stage transformation is not adopted in the rectilinear Steiner minimal tree construction algorithm . Experimental results have shown that the proposed multi-stage transformation adopted in the rectilinear architecture routing can reduce

Table 11 Comparing experiments with multiple swarm intelligence algorithms on ISPD circuits (1)

Circuit	Best value						Mean Value					
	Absolute values			Normalised values			Absolute values			Normalised values		
	ABC	DE	OURS	ABC	DE	OURS	ABC	DE	OURS	ABC	DE	OURS
IBM1	56,721	57,247	56,114	1.011	1.020	1.000	56,729	57,252	56,124	1.011	1.020	1.000
IBM2	15,6470	158,374	155,077	1.009	1.021	1.000	156,487	158,397	155,084	1.009	1.021	1.000
IBM3	135,232	136,516	134,093	1.008	1.018	1.000	135,246	136,527	134,109	1.008	1.018	1.000
IBM4	150,999	152,476	149,826	1.008	1.018	1.000	151,015	152,486	149,833	1.008	1.018	1.000
IBM6	258,955	262,259	256,883	1.008	1.021	1.000	258,984	262,290	256,903	1.008	1.021	1.000
IBM7	338,492	342,195	335,707	1.008	1.019	1.000	338,516	342,215	335,718	1.008	1.019	1.000
IBM8	376,292	380,768	372,984	1.009	1.021	1.000	376,316	380,811	373,002	1.009	1.021	1.000
IBM9	385,517	389,583	382,581	1.008	1.018	1.000	385,540	389,612	382,595	1.008	1.018	1.000
IBM10	537,805	543,717	532,995	1.009	1.020	1.000	537,841	543,760	533,006	1.009	1.020	1.000
Average				1.009	1.020	1.000				1.009	1.020	1.000

Table 12 Comparing experiments with multiple swarm intelligence algorithms on ISPD circuits (2)

Circuit	Standard deviation					
	Absolute values			Normalised values		
	ABC	DE	OURS	ABC	DE	OURS
IBM1	3.9	2.8	3.5	1.114	0.800	1.000
IBM2	11.6	11.7	5.9	1.966	1.983	1.000
IBM3	10.5	5.8	8.1	1.296	0.716	1.000
IBM4	11.2	7.3	4.6	2.435	1.587	1.000
IBM6	21.3	22.6	13.9	1.532	1.626	1.000
IBM7	16.5	14.2	7	2.357	2.029	1.000
IBM8	17.88	25.1	15.5	1.154	1.619	1.000
IBM9	10.42	15.8	7.4	1.408	2.135	1.000
IBM10	16.4	22.2	8.3	1.976	2.675	1.000
Average				1.693	1.686	1.000

Table 13 Comparing experiments with multiple swarm intelligence algorithms on ISPD circuits (3)

Circuit	Best value				Mean value				Standard deviation			
	Absolute values		Normalised values		Absolute values		Normalised values		Absolute values		Normalised values	
	GA	OURS	GA	OURS	GA	OURS	GA	OURS	GA	OURS	GA	OURS
IBM1	56,725	56,114	1.011	1.000	56,736	56,124	1.011	1.000	6.8	3.5	1.943	1.000
IBM2	156,975	155,077	1.012	1.000	156,996	155,084	1.012	1.000	13.3	5.9	2.254	1.000
IBM3	135,674	134,093	1.012	1.000	135,690	134,109	1.012	1.000	8.3	8.1	1.025	1.000
IBM4	150,922	149,826	1.007	1.000	150,940	149,833	1.007	1.000	11.9	4.6	2.587	1.000
IBM6	260,878	256,883	1.016	1.000	260,938	256,903	1.016	1.000	32.6	13.9	2.345	1.000
IBM7	339,430	335,707	1.011	1.000	339,479	335,718	1.011	1.000	22.9	7.0	3.271	1.000
IBM8	378,575	372,984	1.015	1.000	378,618	373,002	1.015	1.000	26.0	15.5	1.677	1.000
IBM9	387,118	382,581	1.012	1.000	387,155	382,595	1.012	1.000	17.1	7.4	2.311	1.000
IBM10	539,040	532,995	1.011	1.000	539,082	533,006	1.011	1.000	26.8	8.3	3.229	1.000
Average			1.012	1.000			1.012	1.000			2.294	1.000

2.93% wirelength than the method without the one without the proposed multi-stage transformation.

5.3 Industrial applications

Our algorithm (OURS) can be applied to industrial circuits and achieve the solutions with high quality, which is helpful to the construction of VLSI router in EDA design. In ISPD benchmark suite (Alpert 1998), a set of industrial circuits has been released. GEO benchmark circuit suites (Zachariasen 2003) are also used to verify the superiority of the proposed method. Following is the application of our algorithm to industrial circuits, and the router in this paper is compared with others.

5.3.1 Comparing experiments with multiple swarm intelligence algorithms on ISPD circuits

Comparing experiments with three swarm intelligence algorithms on ISPD circuits are shown in Tables 11, 12 and 13. These three swarm intelligence algorithms are differential evolution (DE) technique (Manna et al. 2015), artificial bee colony optimization technique (ABC) (Bhattacharya et al. 2014), and GA, respectively.

Each algorithm is executed 20 times. In Tables 11, 12 and 13, for the best value, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 0.9%, 2.0%, and 1.2%, respectively. For the mean value, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 0.9%, 2.0%, and 1.2%,

Table 14 Comparing experiments with multiple swarm intelligence algorithms on GEO circuits (1)

Circuit	Best value						Mean value					
	Absolute values			Normalised values			Absolute values			Normalised values		
	ABC	DE	OURS	ABC	DE	OURS	ABC	DE	OURS	ABC	DE	OURS
1	16,918	17,506	16,900	1.001	1.036	1.000	16,918	17,540	16,921	1.000	1.037	1.000
2	18,041	18,041	18,023	1.001	1.001	1.000	18,041	18,041	18,023	1.001	1.001	1.000
3	19,696	19,703	19,397	1.015	1.016	1.000	19,696	19,734	19,397	1.015	1.017	1.000
4	32,250	32,810	32,039	1.007	1.024	1.000	32,405	32,849	32,063	1.011	1.025	1.000
5	48,196	49,138	47,883	1.007	1.026	1.000	48,603	49,184	48,027	1.012	1.024	1.000
6	56,984	57,547	56,086	1.016	1.026	1.000	57,447	57,596	56,350	1.019	1.022	1.000
7	70,414	70,368	68,427	1.029	1.028	1.000	70,548	70,416	68,625	1.028	1.026	1.000
8	143,242	142,996	140,520	1.019	1.018	1.000	143,681	143,183	140,898	1.020	1.016	1.000
9	156,352	155,742	153,377	1.019	1.015	1.000	156,533	155,929	153,708	1.018	1.014	1.000
10	222,450	221,818	219,637	1.013	1.010	1.000	222,626	222,028	219,954	1.012	1.009	1.000
Average				1.013	1.020	1.000				1.014	1.019	1.000

Table 15 Comparing experiments with multiple swarm intelligence algorithms on GEO circuits (2)

Circuit	Standard deviation					
	Absolute values			Normalised values		
	ABC	DE	OURS	ABC	DE	OURS
1	0.0	100.8	15.7	0.000	6.404	1.000
2	0.0	0.0	0.0	–	–	1.000
3	0.0	93.6	0.0	–	–	1.000
4	71.2	99.9	31.0	2.300	3.227	1.000
5	259.3	70.4	93.9	2.762	0.750	1.000
6	231.4	41.0	144.8	1.598	0.283	1.000
7	55.1	25.9	149.5	0.369	0.173	1.000
8	168.5	102.9	187.6	0.898	0.549	1.000
9	142.7	113.9	180.5	0.790	0.631	1.000
10	100.3	103.2	184.1	0.545	0.560	1.000
Average				1.158	1.572	1.000

Table 16 Comparing experiments with multiple swarm intelligence algorithms on GEO circuits (3)

Circuit	Best value				Mean value				Standard deviation			
	Absolute values		Normalised values		Absolute values		Normalised values		Absolute values		Normalised values	
	GA	OURS	GA	OURS	GA	OURS	GA	OURS	GA	OURS	GA	OURS
1	17,611	16,900	1.042	1.000	17,524	16,921	1.036	1.000	51.2	15.7	3.253	1.000
2	18,041	18,023	1.001	1.000	18,041	18,023	1.001	1.000	0.0	0.0	–	1.000
3	19,703	19,397	1.016	1.000	19,741	19,397	1.018	1.000	11.6	0.0	–	1.000
4	32,711	32,039	1.021	1.000	32,955	32,063	1.028	1.000	81.2	31.0	2.623	1.000
5	49,023	47,883	1.024	1.000	49,268	48,027	1.026	1.000	60.8	93.9	0.648	1.000
6	57,671	56,086	1.028	1.000	57,716	56,350	1.024	1.000	33.0	144.8	0.228	1.000
7	70,264	68,427	1.027	1.000	70,412	68,625	1.026	1.000	41.0	149.5	0.274	1.000
8	141,862	140,520	1.010	1.000	143,019	140,898	1.015	1.000	73.6	187.6	0.392	1.000
9	155,825	153,377	1.016	1.000	155,918	153,708	1.014	1.000	107.2	180.5	0.594	1.000
10	221,911	219,637	1.010	1.000	222,075	219,954	1.010	1.000	74.2	184.1	0.403	1.000
Average			1.019	1.000			1.020	1.000			1.052	1.000

Table 17 Experimental comparison with two recent similar research algorithms

Circuit	# Nets	# points	Wirelength					Runtime (in min)				
			SAT	KNN	OURS	Imp SAT (%)	Imp KNN (%)	SAT	KNN	OURS	Imp SAT	Imp KNN
IBM1	11,507	44,266	61,005	61,071	56,124	8.00	8.10	152.63	3.77	1.08	141.33	3.49
IBM2	18,429	78,171	172,518	167,359	155,084	10.11	7.33	1144.17	9.75	1.93	592.83	5.05
IBM3	21,621	75,710	150,138	147,982	134,109	10.68	9.37	1817.97	10.75	1.61	1129.17	6.68
IBM4	26,163	89,591	164,998	164,838	149,833	9.19	9.10	1993.45	11.55	1.94	1027.55	5.95
IBM6	33,354	124,299	289,705	280,998	256,903	11.32	8.57	3952.85	20.85	2.97	1330.93	7.02
IBM7	44,394	164,369	368,015	368,780	335,718	8.78	8.97	4385.32	23.95	3.65	1201.46	6.56
IBM8	47,944	198,180	431,879	413,201	373,002	13.63	9.73	5844.37	27.82	5.10	1145.95	5.45
IBM9	50,393	187,872	418,382	417,543	382,595	8.55	8.37	3333.63	23.08	4.46	747.45	5.18
IBM10	64,227	269,000	588,079	583,102	533,006	9.36	8.59	5250.10	34.50	6.30	833.35	5.48
Average						9.96	8.68				905.56	5.65

respectively. In Tables 12 and 13, for the standard deviation, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 69.3%, 68.6%, and 29.4%, respectively.

5.3.2 Comparing experiments with multiple swarm intelligence algorithms on GEO circuits

On GEO circuits, comparing experiments with three swarm intelligence algorithms are shown in Tables 14, 15 and 16. Each algorithm runs 20 times. In Tables 14 and 16, for the best value, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 1.3%, 2.0%, and 1.9%, respectively. For the mean value, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 1.4%, 1.9%, and 2.0%, respectively. In Tables 15 and 16, for the standard deviation, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 15.8%, 57.2%, and 5.2%, respectively.

5.3.3 Experimental comparison with two recent similar research algorithms

We compare the proposed algorithm (OURS) with two recent similar research algorithms (Kundu et al. 2017), namely K-nearest neighbour (KNN) in Table 17 (Kundu et al. 2016), namely Scholastic Assessment Test (SAT) in Table 17, and the experimental results are listed in Table 17. The proposed algorithm can reduce 9.96% and 8.68% wirelength than Scholastic Assessment Test and K-nearest neighbour, respectively. Furthermore, the proposed algorithm speeds up 905.56× and 5.65× than Cholastic Assessment Test and K-nearest neighbour, respectively.

6 Conclusions

For industrial circuits, an efficient VLSI routing algorithm employing novel discrete PSO and multi-stage transformation is proposed to build two types of SMT, including X-architecture Steiner minimal tree and rectilinear Steiner minimal tree. Firstly, an effective edge-vertex encoding strategy with four pseudo-Steiner point choices is proposed for the non-Manhattan Steiner tree. It can be effectively extended to construct the Manhattan Steiner tree. Secondly, a multi-stage transformation strategy is presented to expand the search space of the proposed algorithm and accelerate the convergence speed of the proposed algorithm. Finally, a series of experiments on industrial circuits are designed to illustrate the feasibility and effectiveness of the proposed strategies and can achieve better solution with less runtime. Specifically, for the best value, on ISPD circuits, the proposed algorithm can outperform artificial bee colony optimization, differential evolution, and genetic algorithm by 0.9%, 2.0%, and 1.2%, respectively, while on GEO circuits, the proposed algorithm can outperform by 1.3%, 2.0%, and 1.9%, respectively. Furthermore, it can be extended to the global routing problem in both Manhattan and non-Manhattan architecture. Specifically, the proposed algorithm can reduce 9.96% and 8.68% wirelength than Scholastic Assessment Test and K-nearest neighbour, respectively, while it can speed up 905.56X and 5.65X, respectively.

In general, the proposed RA-PSO-MST algorithm based on the edge-vertex encoding strategy needs to be further developed in the solution space, and the corresponding refining strategy needs to be presented, which makes it possible for the proposed algorithm to explore the better solutions. Meanwhile, delay which is not considered in the proposed RA-PSO-MST algorithm is also an important optimization goal in the industrial circuits design. In the future, we will study the construction of delay-driven SMT by integrated the corresponding refining strategy.

Acknowledgements The authors thank the reviewers for their valuable comments/suggestions which helped to improve the quality of this paper significantly.

Funding This work was supported in part by the National Natural Science Foundation of China under Grants No. 61877010 and No.11501114, the National Basic Research Program of China No. 2011CB808000, and the Fujian Natural Science Funds under Grant No. 2019J01243, the project of Technology Innovation Platform of Fujian Province (Grant No. 2014H2005 and No. 2009J1007), and the Fujian Collaborative Innovation Center for Big Data Applications in Governments.

Compliance with ethical standards

Conflict of interest All the authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Alpert CJ (1998) The ispd98 circuit benchmark suite. In: Proceedings of the 1998 international symposium on physical design, pp 80–85. https://doi.org/10.1145/274535.274546

Bhattacharya P, Khan A, Sarkar SK (2014) A global routing optimization scheme based on ABC algorithm. In: Advanced computing, networking and informatics, vol 2. Springer, Cham, pp 189–197. https://doi.org/10.1007/978-3-319-07350-7_21

Chang YJ, Lee YT, Gao JR, Wu PC, Wang TC (2010) NTHU-Route 2.0: a robust global router for modern designs. IEEE Trans Comput Aided Des Integr Circuits Syst 29(12):1931–1944, https://doi.org/10.1109/TCAD.2010.2061590

Chen CM, Wang KH, Yeh KH, Xiang B, Wu TY (2019a) Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. J Ambient Intell Hum Comput 10(8):3133–3142. https://doi.org/10.1007/s12652-018-1029-3

Chen CM, Xiang B, Liu Y, Wang KH (2019b) A secure authentication protocol for internet of vehicles. IEEE Access 7:12047–12057. https://doi.org/10.1109/ACCESS.2019.2891105

Chen X, Liu G, Xiong N, Su Y, Chen G (2020) A survey of swarm intelligence techniques in VLSI routing problems. IEEE Access 8:26266–26292. https://doi.org/10.1109/ACCESS.2020.2971574

Chiang C, Chiang CS (2002) Octilinear Steiner tree construction. In: The 2002 45th Midwest symposium on circuits and systems, 2002 (MWSCAS-2002), vol 1. IEEE, pp I–603. https://doi.org/10.1109/MWSCAS.2002.1187293

Coulston CS (2003) Constructing exact octagonal Steiner minimal trees. In: Proceedings of the 13th ACM Great Lakes symposium on VLSI, pp 1–6. https://doi.org/10.1145/764808.764810

Dai KR, Liu WH, Li YL (2009) Efficient simulated evolution based rerouting and congestion-relaxed layer assignment on 3-d global routing. In: 2009 Asia and South Pacific design automation conference. IEEE, pp 570–575. https://doi.org/10.1109/TVLSI.2010.2102780

Dai KR, Liu WH, Li YL (2011) Nctu-gr: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing. IEEE Trans Very Large Scale Integr (VLSI) Syst 20(3):459–472. https://doi.org/10.1109/TVLSI.2010.2102780

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43. https://doi.org/10.1109/MHS.1995.494215

Feng L, Ali A, Iqbal M, Bashir AK, Hussain SA, Pack S (2019) Optimal haptic communications over nanonetworks for e-health systems. IEEE Trans Ind Inf 15(5):3016–3027. https://doi.org/10.1109/TII.2019.2902604

Gandomi AH, Kashani AR (2017) Construction cost minimization of shallow foundation using recent swarm intelligence techniques. IEEE Trans Ind Inf 14(3):1099–1106. https://doi.org/10.1109/TII.2017.2776132

Garey MR, Johnson DS (1977) The rectilinear Steiner tree problem is NP-complete. SIAM J Appl Math 32(4):826–834. https://doi.org/10.2307/2100192

Guo W, Huang X (2020) PORA: a physarum-inspired obstacle-avoiding routing algorithm for integrated circuit design. Appl Math Model 78:268–286. https://doi.org/10.1016/j.apm.2019.10.027

Guo W, Liu G, Chen G, Peng S (2014) A hybrid multi-objective pso algorithm with local search strategy for VLSI partitioning. Front Comput Sci 8(2):203–216. https://doi.org/10.1007/s11704-014-3008-y

Held S, Müller D, Rotter D, Scheifele R, Traub V, Vygen J (2017) Global routing with timing constraints. IEEE Trans Comput Aided Des Integr Circuits Syst 37(2):406–419. https://doi.org/10.1109/TCAD.2017.2697964

Huang X, Liu G, Guo W, Chen G (2013) Obstacle-avoiding octagonal Steiner tree construction based on particle swarm optimization. In: 2013 Ninth international conference on natural computation (ICNC). IEEE, pp 539–543. https://doi.org/10.1109/ICNC.2013.6818035

Huang X, Liu G, Guo W, Niu Y, Chen G (2015) Obstacle-avoiding algorithm in x-architecture based on discrete particle swarm optimization for VLSI design. ACM Trans Des Autom Electron Syst (TODAES) 20(2):1–28. https://doi.org/10.1145/2699862

Huang X, Guo W, Liu G, Chen G (2016) FH-OAOS: a fast four-step heuristic for obstacle-avoiding octilinear Steiner tree construction. ACM Trans Des Autom Electron Syst (TODAES) 21(3):1–31. https://doi.org/10.1145/2856033

Huang X, Guo W, Liu G, Chen G (2017) MLXR: multi-layer obstacle-avoiding X-architecture Steiner tree construction for VLSI routing. Sci China Inf Sci 60(1):19102. https://doi.org/10.1007/s11432-015-0850-4

Jia YH, Chen WN, Gu T, Zhang H, Yuan H, Lin Y, Yu WJ, Zhang J (2017) A dynamic logistic dispatching system with set-based particle swarm optimization. IEEE Trans Syst Man Cybern Syst 48(9):1607–1621. https://doi.org/10.1109/TSMC.2017.2682264

Koh CK, Madden PH (2000) Manhattan or non-Manhattan? A study of alternative VLSI routing architectures. In: Proceedings of the 10th Great Lakes symposium on VLSI, pp 47–52. https://doi.org/10.1145/330855.330961

Kundu S, Roy S, Mukherjee S (2016) SAT based rectilinear Steiner tree construction. In: 2016 2nd International conference on applied and theoretical computing and communication technology (iCATccT). IEEE, pp 623–627. https://doi.org/10.1109/ICATCCT.2016.7912075

Kundu S, Roy S, Mukherjee S (2017) K-nearest neighbour (KNN) approach using SAT based technique for rectilinear Steiner tree construction. In: 2017 7th International symposium on embedded computing and system design (ISED). IEEE, pp 1–5. https://doi.org/10.1109/ISED.2017.8303915

Liu G, Chen G, Guo W, Chen Z (2011) DPSO-based rectilinear Steiner minimal tree construction considering bend reduction. In: 2011 Seventh international conference on natural computation. IEEE, vol 2, pp 1161–1165. https://doi.org/10.1109/ICNC.2011.6022221

Liu G, Chen G, Guo W (2012) DPSO based octagonal Steiner tree algorithm for VLSI routing. In: 2012 IEEE Fifth International conference on advanced computational intelligence (ICACI). IEEE, pp 383–387. https://doi.org/10.1109/ICACI.2012.6463191

Liu WH, Kao WC, Li YL, Chao KY (2013) NCTU-GR 2.0: Multithreaded collision-aware global routing with bounded-length maze routing. IEEE Trans Comput Aided Des Integr Circuits Syst 32(5):709–722. https://doi.org/10.1109/TCAD.2012.2235124

Liu G, Huang X, Guo W, Niu Y, Chen G (2014) Multilayer obstacle-avoiding X-architecture Steiner minimal tree construction based on particle swarm optimization. IEEE Trans Cybern 45(5):1003–1016. https://doi.org/10.1109/TCYB.2014.2342713

Liu G, Guo W, Li R, Niu Y, Chen G (2015a) XGRouter: high-quality global router in X-architecture with particle swarm optimization. Front Comput Sci 9(4):576–594. https://doi.org/10.1007/s11704-015-4107-1

Liu G, Guo W, Niu Y, Chen G, Huang X (2015b) A PSO-based timing-driven octilinear Steiner tree algorithm for VLSI routing considering bend reduction. Soft Comput 19(5):1153–1169. https://doi.org/10.1007/s00500-014-1329-2

Liu G, Zhuang Z, Guo W, Wang TC (2019) RDTA: an efficient routability-driven track assignment algorithm. In: Proceedings of the 2019 on Great Lakes symposium on VLSI, pp 315–318. https://doi.org/10.1145/3299874.3318026

Liu G, Chen Z, Zhuang Z, Guo W, Chen G (2020) A unified algorithm based on HTS and self-adapting PSO for the construction of octagonal and rectilinear SMT. Soft Comput 24(6):3943–3961. https://doi.org/10.1007/s00500-019-04165-2

Ma T, Liu Q, Cao J, Tian Y, Al-Dhelaan A, Al-Rodhaan M (2020) Lgiem: Global and local node influence based community detection. Fut Gen Comput Syst 105:533–546. https://doi.org/10.1016/j.future.2019.12.022

Manna S, Chakrabarti T, Sharma U, Sarkar SK (2015) Efficient VLSI routing optimization employing discrete differential evolution technique. In: 2015 IEEE 2nd international conference on recent trends in information systems (ReTIS). IEEE, pp 461–464. https://doi.org/10.1109/ReTIS.2015.7232923

Pan JS, Hu P, Chu SC (2019a) Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power. Processes 7(11):845. https://doi.org/10.3390/pr7110845

Pan JS, Lee CY, Sghaier A, Zeghid M, Xie J (2019b) Novel systolization of subquadratic space complexity multipliers based on toeplitz matrix–vector product approach. IEEE Trans Very Large Scale Integr (VLSI) Syst 27(7):1614–1622. https://doi.org/10.1109/TVLSI.2019.2903289

Samanta T, Ghosal P, Rahaman H, Dasgupta P (2006) A heuristic method for constructing hexagonal Steiner minimal trees for routing in VLSI. In: 2006 IEEE international symposium on circuits and systems. IEEE, p 4. https://doi.org/10.1109/ISCAS.2006.1692953

Samanta T, Rahaman H, Dasgupta P (2011) Near-optimal y-routed delay trees in nanometric interconnect design. IET Comput Digit Tech 5(1):36–48. https://doi.org/10.1049/iet-cdt.2009.0074

Siddiqi UF, Sait SM (2017) A game theory based post-processing method to enhance VLSI global routers. IEEE Access 5:1328–1339. https://doi.org/10.1109/ACCESS.2017.2665601

Tang H, Liu G, Chen X, Xiong N (2020) A survey on Steiner tree construction and global routing for VLSI design. IEEE Access 8:68593–68622. https://doi.org/10.1109/ACCESS.2020.2986138

Thurber A, Xue G (1999) Computing hexagonal Steiner trees using PCX for VLSI. In: ICECS'99. Proceedings of ICECS'99. 6th IEEE international conference on electronics, circuits and systems (Cat. No. 99EX357), vol 1. IEEE, pp 381–384. https://doi.org/10.1109/ICECS.1999.812302

Wu H, Xu S, Zhuang Z, Liu G (2019a) X-architecture Steiner minimal tree construction based on discrete differential evolution. In: The international conference on natural computation, fuzzy systems and knowledge discovery. Springer, pp 433–442. https://doi.org/10.1007/978-3-030-32456-8_47

Wu TY, Chen CM, Wang KH, Meng C, Wang EK (2019b) A provably secure certificateless public key encryption with keyword search. J Chin Inst Eng 42(1):20–28. https://doi.org/10.1080/02533839.2018.1537807

Yan JT (2008) Timing-driven octilinear Steiner tree construction based on Steiner-point reassignment and path reconstruction. ACM Trans Des Autom Electron Syst (TODAES) 13(2):1–18. https://doi.org/10.1145/1344418.1344422

Yazdaninejadi A, Golshannavaz S, Nazarpour D, Teimourzadeh S, Aminifar F (2018) Dual-setting directional overcurrent relays for protecting automated distribution networks. IEEE Trans Ind Inf 15(2):730–740. https://doi.org/10.1109/TII.2018.2821175

Zachariasen M (2003) GeoSteiner Homepage [Online]. http://www.diku.dk/geosteiner/

Zhang X, Zhuang Z, Liu G, Huang X, Liu WH, Guo W, Wang TC (2020) Minidelay: multi-strategy timing-aware layer assignment for advanced technology nodes. In: 2020 Design, automation & test in Europe conference & exhibition (DATE), IEEE, pp 586–591. https://doi.org/10.23919/DATE48585.2020.9116269

Zhu Q, Zhou H, Jing T, Hong XL, Yang Y (2005) Spanning graph-based nonrectilinear Steiner tree algorithms. IEEE Trans Comput Aided Des Integr Circuits Syst 24(7):1066–1075. https://doi.org/10.1109/TCAD.2005.850862