Timing-Driven X-architecture Steiner Minimum Tree Construction Based on Social Learning Multi-Objective Particle Swarm Optimization

Xiaohua Chen
College of Mathematics and Computer Sciences, Fuzhou University
Fuzhou, Fujian, China
sherryhchen@163.com

Ruping Zhou
College of Mathematics and Computer Sciences, Fuzhou University
Fuzhou, Fujian, China
zrp08200331@163.com

Genggeng Liu
College of Mathematics and Computer Sciences, Fuzhou University
Fuzhou, Fujian, China
liugenggeng@fzu.edu.cn

Zhen Chen
College of Mathematics and Computer Sciences, Fuzhou University
Fuzhou, Fujian, China
chenzhen@fzu.edu.cn

Wenzhong Guo
College of Mathematics and Computer Sciences, Fuzhou University
Fuzhou, Fujian, China
fzugwz@163.com

ABSTRACT

The construction of timing-driven Steiner minimum tree is a critical issue in VLSI routing design. Meanwhile, since the interconnection model of X-architecture can make full use of routing resources compared to the traditional Manhattan architecture, constructing a Timing-Driven X-architecture Steiner Minimum Tree (TDXSMT) is of great significance to improving routing performance. In this paper, an efficient algorithm based on Social Learning Multi-Objective Particle Swarm Optimization (SLMOPSO) is proposed to construct a TDXSMT with minimizing the maximum source-to-sink pathlength. An X-architecture Prim-Dijkstra model is presented to construct an initial Steiner tree which can optimize both the wirelength and the maximum source-to-sink pathlength. In order to find a better solution, an SLMOPSO method based on the nearest and best select strategy is presented to improve the global exploration capability of the algorithm. Besides, the mutation and crossover operators are utilized to achieve the discrete particle update process, thereby better solving the discrete TDXSMT problem. The experimental results indicate that the proposed algorithm has an excellent trade-off between the wirelength and maximum source-to-sink pathlength of the routing tree and can greatly optimize the timing delay.

CCS CONCEPTS

• **Hardware → Wire routing**; • **Theory of computation → Evolutionary algorithms**.

KEYWORDS

Particle Swarm Optimization, VLSI Routing, X-architecture Steiner Tree, Timing Delay

ACM Reference Format:
Xiaohua Chen, Ruping Zhou, Genggeng Liu, Zhen Chen, and Wenzhong Guo. 2021. Timing-Driven X-architecture Steiner Minimum Tree Construction Based on Social Learning Multi-Objective Particle Swarm Optimization. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3442442.3451143

1 INTRODUCTION

With the continuous shrinking of chip feature size and development of semiconductor technology, the delay effect caused by the interconnected wires is further intensified, thereby affecting the performance of the circuit. And Steiner Minimum Tree (SMT) is the best connection model for multi-terminal nets in global routing of Very Large Scale Integration (VLSI). Therefore, the construction of Timing-Driven Steiner Minimum Tree (TDSMT) becomes a critical issue in VLSI routing design.

At present, most researches on Steiner tree in VLSI routing are limited to wirelength optimization [6, 18, 20, 21, 24]. Considering the influence of source-to-sink delay on chip performance, researchers gradually regard the wirelength cost and transmission delay of SMT as the optimization objectives. For the reason that the transmission delay between two nodes in the spanning tree is measured according to its pathlength, Seo et al. [23] proposed a spanning tree model with minimum wirelength and minimum radius to construct a routing tree with optimal performance. In [12], a performance-driven routing model was proposed to satisfy the delay constraint of given sinks, and introduced the transformation operation of Steiner points to reduce the interconnect delay. Based on Prim-Dijkstra algorithm [3], a new problem model was designed in [2] to optimize the wirelength and radius at the same time and the edge flip technology was proposed to change the topological structure of spanning tree. For global routing that takes delay, congestion, etc. into account, the Steiner tree constructed by this method can achieve shorter wirelength compared with the spanning tree. Chen et al. [4] proposed a new Steiner shAllow-Light Tree (SALT) based on Manhattan architecture, and further optimized the experimental results by adopting post-processing methods such as safe refining and constrained edge replacement.

Most of the current researches on the routing tree are based on Manhattan architecture [22, 30–32]. Lin et al. [15] constructed Rectilinear Steiner Minimum Tree (RSMT) based on the Manhattan architecture to reduce time and routing costs. [16] minimized routing congestion and maximized distribution linearity through RSMT. Two new techniques were proposed in [26] for optimal minimum spanning tree-based obstacle-avoiding component-to-component routing. Guo and Huang [8] proposed a Physarum-inspired obstacle-avoiding rectilinear Steiner tree construction algorithm, which can lead to shorter wirelength compared to several existing methods. However, there are few related studies on TDSMT problem based on non-Manhattan architectures. Because of the more routing directions, the non-Manhattan architectures can obtain more and better candidate topologies to make full use of the limited routing resources. Based on the concept of Steiner-Point reassignment and non-Manhattan path reconstruction, Yan et al. [27] proposed a Timing-Driven X-architecture Steiner Minimum Tree (TDXSMT) construction algorithm, which can obtain shorter wirelength than the original Rectilinear Steiner Tree (RST) while optimizing the timing delay. [11] and [10] respectively proposed a discrete optimization method based on Particle Swarm Optimization (PSO) and a fast heuristic algorithm to shorten the wirelength of the obstacle-avoiding Steiner tree under X-architecture. In [19], an efficient algorithm is presented to construct the minimum cost spanning tree with a minimum radius for performance-driven routing under X-architecture based on PSO. But this routing tree model is only suitable for the placement where the source is in the center and sinks are distributed on the periphery.

In recent years, the optimization algorithms based on Swarm Intelligence (SI), which is inspired by the simple behaviors and self-organizing interactions between biologies [28], have been widely applied in the field of VLSI physical design and achieve good results. In particular, PSO algorithm has obvious advantages in improving the solution quality of Steiner tree problems due to its fewer control parameters, simple implementation and excellent search capability [5]. Because the relatively single learning objects of particles in traditional PSO will make exploitation intensity of the algorithm too large and tend to fall into local extrema, [7] proposed Social Learning Particle Swarm Optimization (SLPSO), an improved PSO, to greatly enhance the global searching capability of PSO and improve the algorithm performance. On the basis of SLPSO, [29] designed a single example learning and an example-mean learning to replace the individual cognition component and social cognition component of SLPSO respectively. But none of these updated formulas can be directly applied to discrete problems. Therefore, this paper proposes a Timing-Driven X-architecture Steiner Minimum Tree construction algorithm based on Social Learning Multi-Objective Particle Swarm Optimization (SLMOPSO-TDXSMT). The proposed algorithm mainly includes the following contents:

- An X-architecture Prim-Dijkstra (X-PD) model is presented to construct an initial Steiner tree which can consider both the wirelength and the maximum source-to-sink pathlength, so as to better optimize interconnection delay.
- A social learning multi-objective particle swarm optimization method based on the nearest and best select strategy is proposed to enhance the diversity of population.
- Combining with the characteristics of the TDXSMT problem, the mutation and crossover operator are integrated into the particle updating process to better solve this discrete problem.

2 PROBLEM DESCRIPTION

2.1 Timing-Driven XSMT

The TDXSMT problem takes timing delay as the optimization goal. Through introducing additional points (Steiner points) to connect given pins, an X-architecture Steiner tree with the minimum cost is finally constructed. And interconnected model of X-architecture means that in addition to the traditional horizontal and vertical directions, the routing directions also include 45° and 135°. This paper takes the maximum source-to-sink delay and total delay of the routing tree as the final optimization goal, and constructs a TDXSMT by optimizing the wirelength and the maximum source-to-sink pathlength of the Steiner tree.

The TDXSMT problem can be description as follows: Given a net N, where s_0 is the source, $S(N) = \{s_1, s_2, ..., s_{n-1}\}$ is the set of sinks, $d(s_i)$ is the timing delay from s_0 to sink s_i, $d(N)$ is the timing delay of the net N. And constructing an X-architecture Steiner tree that can well balance the total wirelength and the maximum source-to-sink pathlength will finally optimize the maximum $d(s_i)$ and $d(N)$.

2.2 Elmore Delay Model

For the reason that Elmore model [13] has the characteristics of low time complexity and high calculation accuracy, it is introduced in the proposed SLMOPSO-TDXSMT algorithm to evaluate the timing delay of routing tree. For net N, an edge in routing tree T represents a segment in the net and is regarded as an individual RC cell. The timing delay of edge e_i in the tree is calculated as follows:

$$d(e_i) = R(e_i) \times (C(e_i)/2 + C_{down}(e_i)) \quad (1)$$

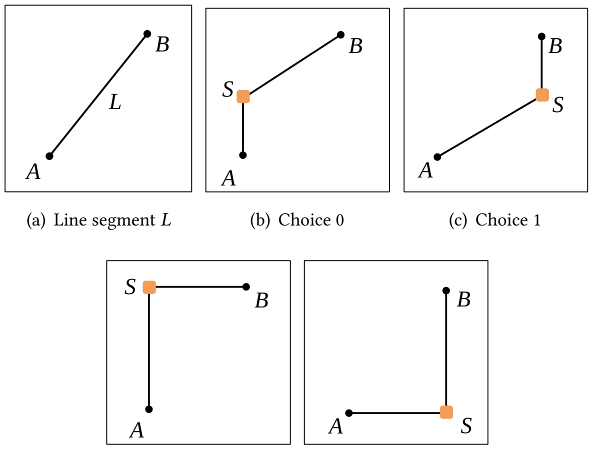
where $d(e_i)$, $R(e_i)$, $C(e_i)$, and $C_{down}(e_i)$ respectively represent the timing delay, resistance, capacitance and downstream capacitance of the edge. And the downstream capacitance of e_i denotes the sum of capacitances of all sinks and edges in the subtree rooted at the sink s_i.

The timing delay $d(s_i)$ at sink s_i is the sum of the timing delay of each edge that the path passes through between s_i and the source, which is calculated as follows:

$$d(s_i) = \sum\nolimits_{e_i \in path(s_0, s_i)} d(e_i) \quad (2)$$

where $path(s_0, s_i)$ is the set of all edges in the path from source to sink s_i.

The total timing delay $d(N)$ of the net is the sum of the weight of all sinks timing delay, in which every sink has a user-defined



(a) Line segment L (b) Choice 0 (c) Choice 1

(d) Choice 2 (e) Choice 3

Figure 1: Four choices of Steiner point for the given segment.

weight. The calculation formula of $d(N)$ is as follows:

$$d(N) = \sum\nolimits_{s_i \in S(N)} \alpha(s_i) \times d(s_i) \tag{3}$$

where $\alpha(s_i)$ is the weight of sink s_i. In order to make the delay of each sink equally important, the weight of each sink is set to $1/|S(N)|$. And $S(N)$ is the number of sinks in the net.

3 SLMOPSO-TDXSMT ALGORITHM

3.1 X-PD Model

Prim-Dijkstra model [3] can well balance the wirelength and the maximum source-to-sink pathlength of the spanning tree. And considering that X-architecture is more flexible than traditional Manhattan architecture, the PD model based on X-architecture is designed to generate the initial routing tree.

The X-architecture interconnection model defines four choices for the interconnection between two pins, as shown in Fig. 1.

Definition 3.1. Pseudo-Steiner point. In addition to original points formed by given pins, the final TDXSMT can be constructed by introducing additional points called Pseudo-Steiner Points (PSP). In Fig. 1, the point S is PSP, and PSP contains the Steiner point.

Definition 3.2. Choice 0 (as shown in Fig. 1(b)). The Choice 0 of PSP corresponding to edge L is defined as leading rectilinear side first from A to PSP S, and then leading non-rectilinear side to B.

Definition 3.3. Choice 1 (as shown in Fig. 1(c)). The Choice 1 of PSP corresponding to edge L is defined as leading non-rectilinear side first from A to PSP S, and then leading rectilinear side to B.

Definition 3.4. Choice 2 (as shown in Fig. 1(d)). The Choice 2 of PSP corresponding to edge L is defined as leading vertical side first from A to PSP S, and then leading horizontal side to B.

Definition 3.5. Choice 3 (as shown in Fig. 1(e)). The Choice 3 of PSP corresponding to edge L is defined as leading horizontal side first from A to PSP S, and then leading vertical side to B.

The process of generating the initial routing tree with the X-PD model is as follows.

The vertex set of net N is $V = \{s_0, s_1, ..., s_{n-1}\}$, with s_0 the source and the remaining vertexes sinks. A routing graph $G = (V, E)$ is defined as a connected weighted graph, where each edge $e_{ij} \in E$ has a cost $cost(s_i, s_j)$. And routing tree $T = (V, E')$ is a spanning subgraph of G with $|E'| = n - 1$. At the beginning, tree T is only composed of s_0, and then sink s_i and edge e_{ij} are iteratively added into T to minimize the cost function:

$$\gamma \cdot cost(s_i) + cost(s_i, s_j) \quad s.t. \quad s_i \in T, s_j \in V - T \tag{4}$$

where $cost(s_i)$ is the total cost of the shortest path from s_0 to s_i, and $cost(s_i, s_j)$ is the cost of the edges between s_i and s_j. When $\gamma = 0$, PD algorithm is identical to Prim algorithm and the constructed T is a Minimum Spanning Tree (MST). When $\gamma = 1$, PD algorithm is identical to Dijkstra algorithm and the constructed T is a Shortest Path Tree (SPT). The proposed SLMOPSO-TDXSMT algorithm randomly generates value $\gamma_i = [0, 0.75]$ for each particle in the initial population, and constructs the initial routing tree to obtain a variety of candidate topologies and increase the search space of the population.

In the proposed X-PD model, the source-to-sink pathlength is calculated as follows:

$$Odis(s_i, s_j) = \begin{cases} |x_{s_i} - x_{s_j}| + |y_{s_i} - y_{s_j}|, & pspc = 2 \, or \, 3 \\ (\sqrt{2} - 1)\min(|x_{s_i} - x_{s_j}|, |y_{s_i} - y_{s_j}|) + \\ \max(|x_{s_i} - x_{s_j}|, |y_{s_i} - y_{s_j}|), & \text{otherwise} \end{cases} \tag{5}$$

where (x_{s_i}, y_{s_i}) and (x_{s_j}, y_{s_j}) respectively represent the position of s_i and s_j, $pspc$ is the PSP choice between s_i and s_j.

3.2 SLMOPSO Method

3.2.1 Particle Encoding. The construction of TDXSMT needs to consider both the wirelength and the maximum source-to-sink pathlength. Thus, in addition to all edges of the Steiner tree and the PSP choice of each edge, the particle encoding should also include the wirelength cost and radius cost of the routing tree. The wirelength cost of the routing tree is the sum of the length of all edge segments. And radius cost is the maximum source-to-sink pathlength of the routing tree, which is equal to the sum of the length of all the edge segments along the path.

For a net with n pins, its routing tree has $n - 1$ edges, the corresponding encoding length of the particle is $3 \times (n - 1) + 2$. Each edge is represented by $(e_{start}, e_{end}, pspc)$, where $pspc$ is the PSP choice between edge (e_{start}, e_{end}). The last two digits of encoding respectively represent the wirelength cost and the radius cost of the routing tree. For example, the encoding of a TDXSMT ($n = 9$) tree can be expressed as follows:

6 7 *0* 7 3 *1* 3 9 *2* 3 1 *3* 9 1 *0* 10 8 *3* 6 2 *0* 2 4 *0* 1 5 *1* **120.0118 34**

where the bold numbers '120.0118' and '34' are the wirelength cost and the radius cost of the routing tree respectively, and the first substring of the encoding (6,7,*0*) means that pin 6 and pin 7 are connected with Choice 0.

3.2.2 Objective Function. In MOPSO algorithm, the individual fitness assessment directly affects the learning process of the optimal historical experience of individual (selecting *pbest*) and social learning in the swarm (selecting *gbest*). The proposed algorithm determines the dominant relationship between particles according to Pareto dominance principle [9].

In the SLMOPSO method, two optimization goals are considered, namely wirelength and maximum source-to-sink pathlength. f_{wl} and f_{pl} are the wirelength cost function and radius cost function of the routing tree, and the specific formula is as follows.

$$f_{wl} = \sum_{e_i \in T} l(e_i) \tag{6}$$

$$f_{pl} = \sum_{e_j \in pathm(s_0, s_i)} l(e_j) \tag{7}$$

3.2.3 Social Learning Based on the Nearest and Best Select Strategy. It is necessary for MOPSO algorithm to maintain an external archive to preserve the Pareto optimal set of the population so far. Therefore, how to select a Pareto optimal solution from the external archive as the social learning object of the particle is very important. The proposed SLMOPSO-TDXSMT algorithm adopts ε-dominance [14] to maintain the external archive and control its size.

After completing inertia component and the learning of its own historical optimal experience, the particle enters the social learning phrase. Social learning is an important process to realize the information exchange among individuals in the population and also the most important way to promote particles to move towards the global optimal solution. In traditional PSO, every iteration generates only one *gbest*, so the social learning objects of all the particles in an iteration are the same one, that is *gbest*. Moreover, when the algorithm falls into the local optimal solution, the *gbest* found in the population may remain unchanged after several iterations, leading to the movement of all particles toward the local optimal particle, resulting in the aggregation phenomenon and finally obtaining the suboptimal solution. While in this paper, a social learning mechanism based on the nearest and best select strategy is proposed to make the social learning objects of all particles in the same iteration can be as different as possible, so as to have the opportunity to jump out of the local extrema. The details are bellow.

(1) The distance between the particle and each Pareto optimal solution is calculated by using the target distance function, which is shown in Eq. 8:

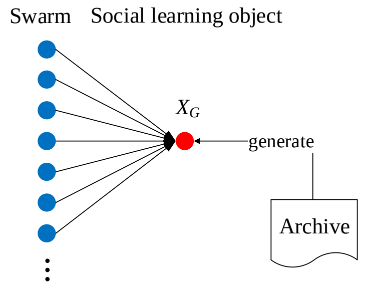
$$fd_{ij} = \left| f_{wl}(X_i) - f_{wl}(X_j) \right| + \left| f_{pl}(X_i) - f_{pl}(X_j) \right|, i \neq j \tag{8}$$

where fd_{ij} is the distance between particle X_i and X_j, and X_j is one of the optimal solution particles in the external archive.

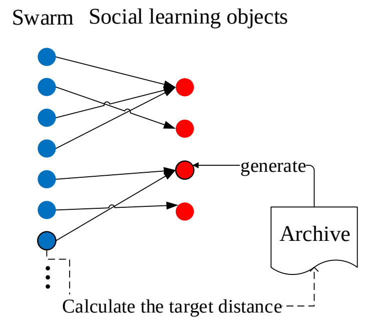
(2) The particle with the smallest fd_{ij} is selected as the nearest and best solution of the current particle X_i, namely the social learning object of X_i, which is denoted as *nbest*.

(3) The particle follows the update process described in Section 3.2.4 for social learning.

As shown in Fig. 2, most MOPSO algorithms generate a *gbest* from external archive as the social learning object for all particles in a specific way, such as random selection, selection based on density distance or fitness function and so on. The disadvantage of this kind of learning mode is that the learning object is single and the algorithm tends to be precocious. While in SLMOPSO-TDXSMT algorithm, the social learning object of each particle is generated from the external archive by calculating the distance from its own target. This makes it possible for different particles to learn from different optimal objects in the same iteration. And the objects learned by the same particle in different iteration are also as different as possible so that the exploration ability of the algorithm is improved. At the same time, the quality of learning is guaranteed because the *nbest* of the particle is also selected from external archive and the algorithm will not degrade its exploitation capability due to excessive exploration.



(a) Traditional social learning mode



(b) Social learning mode based on the nearest and best select strategy

Figure 2: Social learning mode of SLMOPSO-TDXSMT algorithm.

3.2.4 Particle Update Formula. In order to better solve the discrete problem of TDXSMT, particle updating process is realized by introducing genetic operators in proposed SLMOPSO-TDXSMT algorithm, and follows the following update formula:

$$X_i^t = \mathrm{NF}_3(\mathrm{NF}_2(\mathrm{NF}_1(X_i^{t-1}, \omega), c_1), c_2) \tag{9}$$

where ω is inertia weight, c_1 and c_2 are acceleration factors, NF_1, NF_2 and NF_3 respectively represent the inertia component, individual cognition and social cognition. The specific operation is as follows.

(1) Inertia Component.

The particle velocity is updated through NF_1 by introducing mutation operator and edge transformation strategy [17], which is express as follows:

$$W_i^t = \mathrm{NF}_1(X_i^{t-1}, \omega) = \begin{cases} \mathrm{M_e}(\mathrm{M_p}(X_i^{t-1})), & r_1 < \omega \\ X_i^{t-1}, & \text{otherwise} \end{cases} \tag{10}$$

where ω determines the probability of mutation operation, r_1 is a random number in $[0, 1]$, $\mathrm{M_p}()$ is the mutation operation for PSP transformation, and $\mathrm{M_e}()$ is the mutation operation for edge transformation.

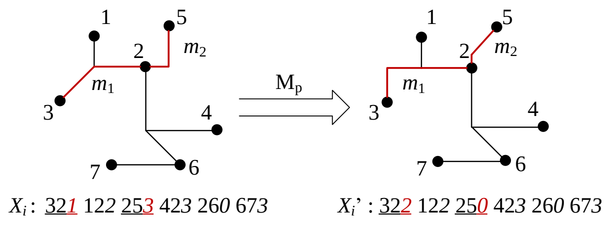

1
5
2
m_2
m_1
3
4
7
6
$\mathrm{M_p}$
X_i: 321 122 253 423 260 673
X_i' : 322 122 250 423 260 673

Figure 3: Mutation operator for PSP transformation.

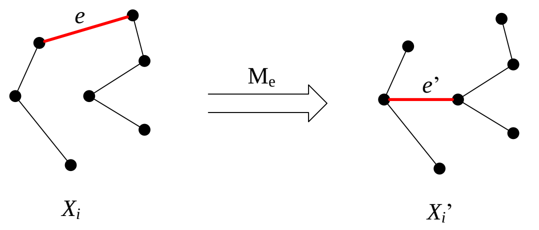


Figure 4: Mutation operator for edge transformation.

If the random number $r_1 < \omega$, the particle will take mutation operation, otherwise it will maintain the current state itself. The steps of the mutation operation are as follows: 1) Firstly, the particle performs the mutation operation of PSP transformation, as shown in Fig. 3. The algorithm randomly selects two edges, $m_1(3, 2, 1)$ and $m_2(2, 5, 3)$, and then replaces the PSP choices of these two edges, so as to achieve the purpose of randomly changing the topology of the routing tree. 2) Then the mutation for edge transformation is performed, as shown in Fig. 4. The algorithm randomly selects edge e to be mutated. And through using Union-Find partition, two pins randomly selected from the two subtrees are connected and finally get new edge e'.

(2) Individual Cognition.

The algorithm uses NF_2 to complete the individual cognition of particles, which is expressed as follows:

$$S_i^t = \mathrm{NF}_2(W_i^t, c_1) = \begin{cases} \mathrm{C_u}(W_i^t, X_i^P), r_2 < c_1 \\ W_i^t, \quad \text{otherwise} \end{cases} \tag{11}$$

where $\mathrm{C_u}()$ represents the crossover operation, the particle W_i^t crosses with its historical optimal X_i^P, and r_2 is a random number in $[0, 1]$.

(3) Social Cognition.

The algorithm uses NF_3 to complete the social cognition of particles, which is expressed as follows:

$$X_i^t = \mathrm{NF}_3(S_i^t, c_2) = \begin{cases} \mathrm{C_u}(S_i^t, X_i^N), r_3 < c_2 \\ S_i^t, \quad \text{otherwise} \end{cases} \tag{12}$$

where X_i^N is the nearest and best particle (*nbest*) that the particle S_i^t crosses with, r_3 is a random number in $[0, 1]$.

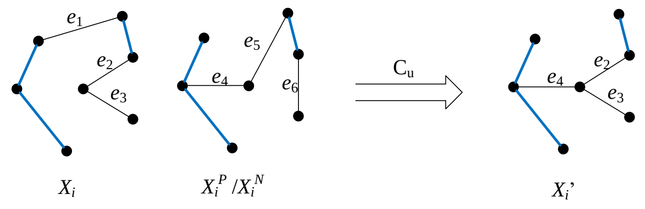


Figure 5: Crossover operator.

NF_2 and NF_3 implement the particle learning by introducing crossover operators. When the generated random number $r_2 < c_1$ (or $r_3 < c_2$), the algorithm performs crossover operation, otherwise the current state of the particle is maintained. Assume that $E_i = \{E_s, E_i^r\}$ is the edge set of the routing tree corresponding to the particle that need to be crossed. And the edge set of the routing tree corresponding to its learning object is $E_{i,best} = \{E_s, E_{i,best}^r\}$, where E_s is the same edge set of both. The steps of the crossover operations are as follows: 1) Select the same edges E_s of the particle to be crossed and its learning object, and E_s constitutes part of the new particle's edges. 2) Establish the set of remaining edges, $E_r = \{E_i^r, E_{i,best}^r\}$. 3) Randomly select edges from set E_r as the edges of the new particle until a completed tree is constructed. The algorithm uses Union-Find partition in Step 3) to avoid the occurrence of illegal trees. Fig. 5 gives an example of the crossover process for edge transformation where the blue edges are the same edges of the particles. For set $E_r = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, the algorithm randomly selects e_2, e_3 and e_4 as the new particle's edges.

3.3 Overall Flow of SLMOPSO-TDXSMT Algorithm

The overall flow of the proposed SLMOPSO-TDXSMT algorithm can be summarized as follows.

Step 1. Initialize parameters. Based on X-PD model, initial routing trees are constructed as initial population.

Step 2. Calculate the objective function values of each particle according to Eqs. 6-7. Then initialize each particle's *pbest* to itself, and store non-inferior solutions in the external archive.

Step 3. Determine the nearest and best solution *nbest* for each particle according to Eq. 8.

Step 4. Update the velocity and position of each particle according to Eqs. 9-12.

Step 5. Recalculate the objective function of the particle.

Step 6. Update *pbest* and the external archive for each particle according to Pareto dominance principle, and update *nbest* for the particle according to objective distance formula.

Step 7. If the termination condition is met (the set maximum number of iterations is reached), select the optimal solution from external archive by Neighbor Density Measure [19], and end the algorithm. Otherwise, return to Step 4.

The update for the external archive in Step 6 above is as follows (the maximum capacity of the external archive is K, X is the updated particle, A and A' are the external archives before and after updating, respectively).

(1) If $A = \emptyset$, return $A' = \{X\}$.

(2) If $X \succ X^P$, update the individual historical optimal solution as the current particle $X^P = X$.

(3) For $\forall a_i \in A$, if $a_i \succ X$, return $A' = A$.

(4) For $\forall a_i \in A$, if $X \succ a_i$, return $A = A/\{a_i\}$.

(5) If $|A| < K$, return $A' = A \cup \{X\}$.

4 EXPERIMENTAL RLESULTS

In order to verify the effectiveness of the proposed SLMOPSO-TDXSMT algorithm, experiments are performed on the benchmark circuit ISPD [1] and GEO [25], and detailed comparison results are given in this section. In this paper, the population size is set to 100, the max iteration number is set to 1000, and other parameters are consistent with [19]. The algorithm uses the Elmore delay model based on the 0.18μm IC technology, with the specific parameters shown in Table 1.

Table 1: Parameters on 0.18μm IC Technology.

Symbol	Description	Value
r_0	Unit-length wire resistance	0.0075(Ω/μm)
c_0	Unit-length wire capacitance	0.118(fF//μm)
C_L	Load capacitance on sink	23.4(fF)
R_D	Driver resistance on source	180(Ω)

The ultimate purpose of optimizing wirelength and maximum source-to-sink pathlength is to optimize maximum source-to-sink delay and total timing delay. Therefore, the proposed SLMOPSO-TDXSMT algorithm aims to greatly reduce the source-to-sink delay while ensuring the increase of wirelength can be controlled in an acceptable range. In order to verify the effectiveness of the proposed algorithm on the timing delay optimization, this section compares the performance of SLMOPSO-TDXSMT algorithm with TOST_BR_MOPSO [19] algorithm on the wirelength (WL), maximum source-to-sink pathlength (PL), maximum source-to-sink delay (MD) and total delay (TD). Table 2 and Table 3 are the experimental results in benchmark circuits ISPD and GEO, respectively. Each test of benchmark circuit ISPD consists of multiple nets, and the second column of Table 2 is the number of selected nets (using 500 as the interval, equidistant sampling to select some of the nets). And each test of benchmark circuit GEO is a single net, the second column of Table 3 is the number of pins of the net.

In Table 2, compared with TOST_BR_MOPSO algorithm, our algorithm can achieve significant optimization on total delay for each test case with an average TD reduction of 19.23% and average MD optimization of 2.56%, while the WL only increases by 0.21%. For test 3, the WL obtained by ours has increased by 0.29%, but the PL can be reduced by 1.88%, which better balances the optimization between them, thus achieving 1.86% MD optimization and 17.88% TD optimization.

Table 3 shows the experimental results on nets with different scale. In terms of WL optimization capabilities, the performance of the two algorithms is comparable, and the average WL of ours increases by 0.17%, but average PL is optimized by 9.44%. Therefore, both the maximum source-to-sink delay and the total timing-delay are greatly improved, which are optimized by 11.89% and 10.16%, respectively. Especially for large scale nets (more than 20 pins), PL can be greatly reduced with sacrificing a little WL. It is because the initial solution of our algorithm is generated through X-PD model, and the algorithm adopts social learning mode based on the nearest and best select strategy, which expands the scope of learning objects of the particles and the exploration capability is enhanced, so as to have the opportunity to explore better solutions.

5 CONCLUSIONS

Aiming at timing-driven X-architecture Steiner minimum tree problem, this paper proposes a social learning multi-objective particle swarm optimization algorithm which can achieve an excellent trade-off between the wirelength and maximum source-to-sink pathlength of the routing tree. Firstly, the proposed algorithm generates an initial routing tree through the X-PD model to obtain high-quality initial solutions. Secondly, in order to better solve the multi-objective problem of TDXSMT, the MOPSO method based on Pareto dominance is presented to search for the optimal solution. At the same time, since the particles of traditional PSO algorithms tend to learn a single object during social learning stage, which leads the algorithm to converge too fast or fall into the local extrema easily, the nearest and best select strategy is designed to enhance the exploration capability of the algorithm. Finally, in the discretization process of PSO, the velocity and position update of the particle are realized by mutation operator and crossover operator, and the PSP transformation and edge transformation strategies are combined to obtain more topologies.

The proposed SLMOPSO-TDXSMT algorithm can effectively balance wirelength and maximum source-to-sink pathlength of Steiner tree, thereby optimizing timing delay. Compared with the similar timing-driven routing tree algorithm, the proposed algorithm can greatly reduce the maximum source-to-sink delay and the total timing delay, so as to further improve the routing quality.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grants No. 61877010, No. 11501114, No. 11271002 and No. 11141005, National Basic Research Program of China under Grant No. 2011CB808000), State Key Laboratory of Computer Architecture (ICT,CAS) under Grant No. CARCHB202014, Fujian Natural Science Funds under Grants No. 2019J01243 and No. 2018J07005, and Fuzhou University under Grants No. GXRC-20060 and No. XRC-1544. Genggeng Liu is the corresponding author of the article.

REFERENCES

[1] Charles J. Alpert. 1998. The ISPD98 Circuit Benchmark Suite. In *Proceedings of the 1998 International Symposium on Physical Design* (Monterey, California, USA) *(ISPD '98)*. Association for Computing Machinery, New York, NY, USA, 80–85. https://doi.org/10.1145/274535.274546

[2] Charles J. Alpert, Wing-Kai Chow, Kwangsoo Han, Andrew B. Kahng, Zhuo Li, Derong Liu, and Sriram Venkatesh. 2018. Prim-Dijkstra Revisited: Achieving Superior Timing-Driven Routing Trees. In *Proceedings of the 2018 International Symposium on Physical Design*. Association for Computing Machinery, New York, NY, USA, 10–17. https://doi.org/10.1145/3177540.3178239

[3] Charles J. Alpert, T. C. Hu, Jen-Hsin Huang, Andrew B. Kahng, and David Karger. 1995. Prim-Dijkstra tradeoffs for improved performance-driven routing tree design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 14, 7 (1995), 890–896. https://doi.org/10.1109/43.391737

Table 2: Comparison with TOST_MOPSO algorithm on benchmark circuit ISPD.

Circuit	Nets	TOST_MOPSO				SLMOPSO-TDXSMT				Imp (%)			
		WL (μm)	PL (μm)	MD (fs)	TD (fs)	WL (μm)	PL (μm)	MD (fs)	TD (fs)	WL	PL	MD	TD
1	24	102	85	30	55	102	85	30	36	0.00	0.00	0.59	34.78
2	37	250	171	56	128	252	168	54	112	-1.02	1.56	2.32	12.28
3	44	457	280	143	372	459	275	140	305	-0.29	1.88	1.86	17.88
4	53	284	204	49	82	284	205	49	75	0.06	-0.31	0.76	8.86
5	67	531	284	120	378	532	282	106	288	-0.26	0.92	11.66	23.79
6	89	358	289	61	83	358	290	61	80	0.05	-0.14	-0.18	3.40
7	96	676	365	230	1058	677	363	224	781	-0.07	0.65	2.46	26.16
8	101	826	592	256	697	828	590	254	498	-0.18	0.21	0.76	28.53
9	129	1126	760	294	637	1128	757	286	527	-0.17	0.40	2.82	17.39
Avg										-0.21	0.57	2.56	19.23

Table 3: Comparison with TOST_MOPSO algorithm on benchmark circuit GEO.

Circuit	Pins	TOST_MOPSO				SLMOPSO-TDXSMT				Imp (%)			
		WL (μm)	PL (μm)	MD (ns)	TD (ns)	WL (μm)	PL (μm)	MD (ns)	TD (ns)	WL	PL	MD	TD
1	8	17170	6503	0.034	0.113	16951	6503	0.034	0.111	1.28	0.00	0.00	1.86
2	9	18041	6747	0.022	0.061	18039	6747	0.022	0.060	0.01	0.00	0.00	1.20
3	10	19586	6747	0.022	0.069	19467	6735	0.022	0.071	0.61	0.17	0.28	-3.61
4	20	32700	11523	0.085	0.294	32953	10883	0.077	0.272	-0.77	5.56	9.37	7.62
5	50	49139	14414	0.168	1.009	50467	11092	0.126	0.928	-2.70	23.05	24.97	8.01
6	70	57584	15730	0.254	2.283	57833	13009	0.189	2.017	-0.43	17.30	25.52	11.67
7	100	70483	15325	0.254	3.716	70363	13118	0.201	2.831	0.17	14.40	20.73	23.81
8	410	143535	23635	0.904	58.726	143764	21591	0.898	53.161	-0.16	8.65	0.71	9.48
9	500	156269	23490	1.279	96.154	155936	19818	0.960	65.983	0.21	15.63	24.92	31.38
10	1000	222378	21071	1.877	267.552	222115	19031	1.645	240.415	0.12	9.68	12.38	10.14
Avg										-0.17	9.44	11.89	10.16

[4] Gengjie Chen and Evangeline F. Y. Young. 2020. SALT: Provably Good Routing Topology by a Novel Steiner Shallow-Light Tree Algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 6 (2020), 1217–1230. https://doi.org/10.1109/TCAD.2019.2894653

[5] Xiaohua Chen, Genggeng Liu, Naixue Xiong, Yaru Su, and Guolong Chen. 2020. A Survey of Swarm Intelligence Techniques in VLSI Routing Problems. *IEEE Access* 8 (2020), 26266–26292. https://doi.org/10.1109/ACCESS.2020.2971574

[6] Xiaohua Chen, Ruping Zhou, Genggeng Liu, and Xin Wang. 2020. SLPSO-Based X-Architecture Steiner Minimum Tree Construction. In *Web Information Systems and Applications*. Springer International Publishing, Cham, 131–142. https://doi.org/10.1007/978-3-030-60029-7_12

[7] Ran Cheng and Yaochu Jin. 2015. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences* 291 (2015), 43–60. https://doi.org/10.1016/j.ins.2014.08.039

[8] Wenzhong Guo and Xing Huang. 2020. PORA: A Physarum-inspired obstacle-avoiding routing algorithm for integrated circuit design. *Applied Mathematical Modelling* 78 (2020), 268–286. https://doi.org/10.1016/j.apm.2019.10.027

[9] Wang Hu, Gary G. Yen, and Xin Zhang. 2014. Multiobjective Particle Swarm Optimization Based on Pareto Entropy. *Journal of Software* 25 (2014), 1025–1050. Issue 05. https://doi.org/10.13328/j.cnki.jos.004496 [In Chinese].

[10] Xing Huang, Wenzhong Guo, and Guolong Chen. 2015. Fast obstacle-avoiding octilinear steiner minimal tree construction algorithm for VLSI design. In *Sixteenth International Symposium on Quality Electronic Design*. IEEE, Santa Clara, CA, USA, 46–50. https://doi.org/10.1109/ISQED.2015.7085396

[11] Xing Huang, Genggeng Liu, Wenzhong Guo, and Guolong Chen. 2013. Obstacle-Avoiding Octagonal Steiner Tree construction based on Particle Swarm Optimization. In *2013 Ninth International Conference on Natural Computation (ICNC)*. IEEE, Shenyang, China, 539–543. https://doi.org/10.1109/ICNC.2013.6818035

[12] Huibo Hou, Jiang Hu, and Sachin S. Sapatnekar. 1999. Non-Hanan routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18, 4 (1999), 436–444. https://doi.org/10.1109/43.752927

[13] Yehea I. Ismail, Eby G. Friedman, and Jose L. Neves. 2000. Equivalent Elmore delay for RLC trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 19, 1 (2000), 83–97. https://doi.org/10.1109/43.822622

[14] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. 2002. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation* 10, 3 (2002), 263–282. https://doi.org/10.1162/106365602760234108

[15] Kuen-Wey Lin, Yeh-Sheng Lin, Yih-Lang Li, and Rung-Bin Lin. 2018. A Maze Routing-Based Methodology With Bounded Exploration and Path-Assessed Retracing for Constrained Multilayer Obstacle-Avoiding Rectilinear Steiner Tree Construction. *ACM Trans. Des. Autom. Electron. Syst.* 23, 4, Article 45 (2018), 26 pages. https://doi.org/10.1145/3177878

[16] Sheng-En David Lin and Dae Hyun Kim. 2018. Construction of All Rectilinear Steiner Minimum Trees on the Hanan Grid. In *Proceedings of the 2018 International Symposium on Physical Design* (Monterey, California, USA) *(ISPD '18)*. Association for Computing Machinery, New York, NY, USA, 18–25. https://doi.org/10.1145/3177540.3178240

[17] Genggeng Liu, Zhisheng Chen, Zhen Zhuang, Wenzhong Guo, and Guolong Chen. 2020. A unified algorithm based on HTS and self-adapting PSO for the construction of octagonal and rectilinear SMT. *Soft Computing* 24 (2020), 3943–3961. Issue 6. https://doi.org/10.1007/s00500-019-04165-2

[18] Genggeng Liu, Wenzhong Guo, Rongrong Li, Yuzhen Niu, and Guolong Chen. 2015. XGRouter: high-quality global router in X-architecture with particle swarm

optimization. *Frontiers of Computer Science* 9, 4 (2015), 576–594. https://doi.org/10.1007/s11704-015-4107-1

[19] Genggeng Liu, Wenzhong Guo, Yuzhen Niu, Guolong Chen, and Xing Huang. 2015. A PSO-based timing-driven Octilinear Steiner tree algorithm for VLSI routing considering bend reduction. *Soft Computing* 19 (2015), 1153–1169. Issue 5. https://doi.org/10.1007/s00500-014-1329-2

[20] Genggeng Liu, Xing Huang, Wenzhong Guo, Yuzhen Niu, and Guolong Chen. 2015. Multilayer Obstacle-Avoiding X-Architecture Steiner Minimal Tree Construction Based on Particle Swarm Optimization. *IEEE Transactions on Cybernetics* 45, 5 (2015), 1003–1016. https://doi.org/10.1109/TCYB.2014.2342713

[21] Genggeng Liu, Weida Zhu, Saijuan Xu, Zhen Zhuang, and Guolong Chen. 2020. Efficient VLSI routing algorithm employing novel discrete PSO and multi-stage transformation. *Journal of Ambient Intelligence and Humanized Computing* (2020), 1–16. https://doi.org/10.1007/s12652-020-02659-8

[22] Genggeng Liu, Zhen Zhuang, Wenzhong Guo, and Ting-Chi Wang. 2019. RDTA: An Efficient Routability-Driven Track Assignment Algorithm. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI*. Association for Computing Machinery, New York, NY, USA, 315–318. https://doi.org/10.1145/3299874.3318026

[23] Dae Young Seo. 1999. *On the complexity of bicriteria spanning tree problems for a set of points in the plane.* Ph.D. Dissertation. USA: Northwestern University.

[24] Hao Tang, Genggeng Liu, Xiaohua Chen, and Naixue Xiong. 2020. A Survey on Steiner Tree Construction and Global Routing for VLSI Design. *IEEE Access* 8 (2020), 68593–68622. https://doi.org/10.1109/ACCESS.2020.2986138

[25] David Warme, Pawel Winter, and Martin Zachariasen. 2003. GeoSteiner Software for Computing Steiner Trees. http://geosteiner.net.

[26] Emilio. Wuerges. 2020. 3-Step Rectilinear Minimum Spanning Tree Construction for Obstacle-Avoiding Component-to-Component Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 12 (2020), 5123–5127. https://doi.org/10.1109/TCAD.2020.2972534

[27] Jin-Tai Yan. 2008. Timing-driven octilinear Steiner tree construction based on Steiner-point reassignment and path reconstruction. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 13, 2 (2008), 1–18. https://doi.org/10.1145/1344418.1344422

[28] Feng Yang, Pengxiang Wang, Yizhai Zhang, Litao Zheng, and Jianchun Lu. 2017. Survey of swarm intelligence optimization algorithms. In *2017 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE, Beijing, China, 544–549. https://doi.org/10.1109/ICUS.2017.8278405

[29] Xinming Zhang, Xia Wang, Qiang Kang, and Jinfeng Cheng. 2019. Differential mutation and novel social learning particle swarm optimization algorithm. *Information Sciences* 480 (2019), 109–129. https://doi.org/10.1016/j.ins.2018.12.030

[30] Xinghai Zhang, Zhen Zhuang, Genggeng Liu, Xing Huang, Wen-Hao Liu, Wenzhong Guo, and Ting-Chi Wang. 2020. MiniDelay: Multi-Strategy Timing-Aware Layer Assignment for Advanced Technology Nodes. In *Proceedings of the 23rd Conference on Design, Automation and Test in Europe* (Grenoble, France) *(DATE '20)*. EDA Consortium, San Jose, CA, USA, 586–591. https://doi.org/10.5555/3408352.3408484

[31] Weida Zhu, Xinghai Zhang, Genggeng Liu, Wenzhong Guo, and Ting Chi Wang. 2020. MiniDeviation: An Efficient Multi-Stage Bus-Aware Global Router. In *2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*. IEEE, Hsinchu, Taiwan, 1–4. https://doi.org/10.1109/VLSI-DAT49148.2020.9196219

[32] Zhen Zhuang, Genggeng Liu, Xing Huang, Xiaotao Jia, Wen-Hao Liu, and Wenzhong Guo. 2020. MSFRoute: Multi-Stage FPGA Routing for Timing Division Multiplexing Technique. In *Proceedings of the 2020 on Great Lakes Symposium on VLSI* (Virtual Event, China) *(GLSVLSI '20)*. Association for Computing Machinery, New York, NY, USA, 107–112. https://doi.org/10.1145/3386263.3406902