

Research on Methods for Very Large Scale Integration Track Assignment Routing

Hui Yu¹, Shipeng Huang², Lu Ren^{1,*}

¹ The Network Information Center, Wuhan University of Technology, Wuhan, China

² School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China

Abstract. Routing is a crucial stage in the physical design of Very Large Scale Integration (VLSI) circuits, comprising three phases: global routing, track assignment routing, and detailed routing. With the development of VLSI circuits, scholars have proposed various track assignment routing algorithms. However, improving the efficiency of track assignment routing and optimizing conflicting design rules have become bottlenecks in track assignment routing problems. This study addresses these bottlenecks by utilizing single-level horizontal and vertical Steiner trees to extract routability information of local wire nets, resolving the adaptation issue between global routing and detailed routing. The proposed algorithm enhances routability information by an average of 16.07% across ten benchmark circuits. Additionally, a Generative Neural Network model based on Conditional Variational Autoencoder (CVAE) is employed to improve the efficiency of track assignment routing, yielding significant simulation results. Furthermore, a negotiation-based tear-and-reassign approach is utilized to address track congestion issues, resulting in an average optimization of 26.03% in overlap cost, with a tradeoff of sacrificing 6.67% of wirelength on average.

1 Research Background and Significance

In order to integrate more circuit components on smaller chips while meeting the requirements for chip integration and improving existing chip fabrication processes, higher demands have been placed on Very Large Scale Integration (VLSI) design. In the VLSI design process, physical design directly affects the cycle, production cost, and product quality of chip design [1]. Among them, routing is the most time-consuming and important aspect of VLSI design, and the prerequisite for the size and complexity of advanced process ultra-large-scale chips makes routing problems more challenging.

Due to the complexity of routing problems, they are typically divided into three stages: global routing, track assignment routing, and detailed routing. In the global

routing stage, all routing areas are divided into rectangular grid units, and a coarse-grained 3D routing graph is used to represent the routing results. However, since global routing can only provide approximate routing paths and cannot accurately capture local congestion areas in the results, it may increase unnecessary workload for subsequent detailed routing. Track assignment routing allocates the results of global routing to corresponding routing resources according to certain process design rules. Finally, detailed routing connects the routability information in the wire nets to the pins based on the allocation scheme of track assignment routing.

Currently, many routers overlook the process of track assignment routing, and only consider global routing and detailed routing. This traditional routing process is inefficient and has significant limitations for multilayer routing. In order to better complete the routing process,

* Corresponding Author: Lu Ren, Email: renlu1989@whut.edu.cn`

routers need to incorporate an effective track assignment routing method between global routing and detailed routing to optimize the process's adherence to process design rules. Track assignment routing allocates the results of global routing to routable resources on different layers, and when evaluating the routability of track assignment routing, it aims to satisfy process design rule constraints as much as possible. Therefore, improving the efficiency of track assign.

2 Research Content and Methodology

This study aims to address the bottleneck issue in track assignment routing regarding how to improve efficiency and optimize conflicting design rules. The main contents of this research include:

1. Considering fabricability information from both global and local nets before track assignment routing, while traditionally ignoring the latter. This aims to extract additional fabricability information from local nets.
2. Improving the efficiency of track assignment routing by optimizing wirelength cost through adjustments in the overall cost function.
3. Considering the impact of overlap cost on detailed routing and employing a refinement strategy based on negotiation-based tear-and-reassign method. This strategy optimizes overlap cost while sacrificing minimal wirelength cost, facilitating smoother progress in subsequent detailed routing stages.

The technical roadmap of this research is illustrated in Figure 1.

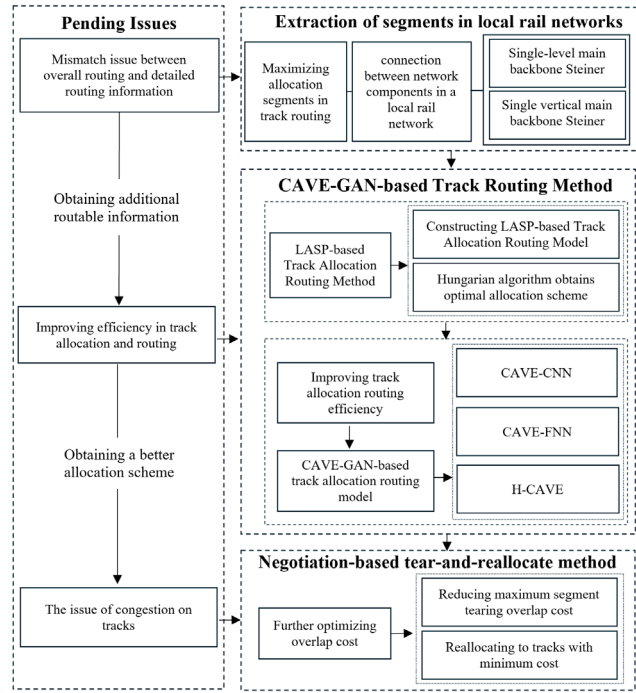


Fig. 1. Technical Roadmap.

3 Study on the Feasibility of Local Net Wiring Information

To address the mismatch issue between global routing and detailed routing, the study considers the routability information of local wire nets. This is achieved by extracting routability information from local wire nets using single vertical backbone Steiner trees and single horizontal backbone Steiner trees.

3.1 Extracting segments of local wire networks

In order to obtain more accurate routability information, this study defines line segments differently from previous research. In previous studies, only straight lines that pass through one or more global units were considered as line segments. As shown in Figure 2-1(a), which illustrates the overall wiring results with two wire networks, each small grid represents a global unit. n_1 represents a global wire network, with its overall wiring path highlighted by a red box, while n_2 represents a local wire network, with its pins all in the same global unit. As shown in Figure 2-1(b), considering only the global wire network, only line segment w_1 would be regarded as a line segment, while other line segments would be ignored. Therefore, ignoring the local wire network would result in a loss of a significant amount of wiring information.

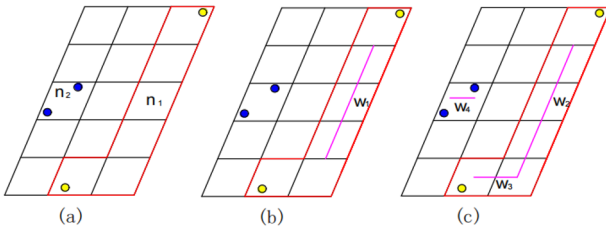


Fig. 2-1: (a)Overall routing results of n_1 and n_2 . (b)Line segment extraction results considering only the global wire network. (c)Line segment extraction results considering both global wire nets and local wire nets.

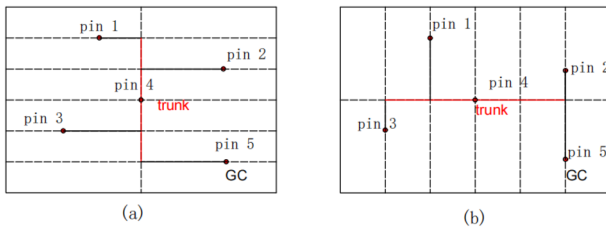


Fig. 2-2: (a)Single vertical backbone Steiner tree. (b)Single horizontal backbone Steiner tree.

This study simultaneously extracts routability information from both global wire nets and local wire nets. Given the overall routing result of a global wire net, a straight line from the center of one global unit to the center of another global unit is considered as a line segment. Since local wire nets are often overlooked in overall routing, there are no routing results associated with local wire nets. To capture the routability information of local wire nets, it is necessary to know the topological structure between wire net components in the local wire net.

To this end, two straight Steiner trees are constructed, namely, a single vertical main stem Steiner tree and a single horizontal main stem Steiner tree for each local wire network. The single vertical (horizontal) main stem Steiner tree connects all pins in the local wire network to the main stem via vertical (horizontal) segments. The x-coordinate (y-coordinate) of the vertical (horizontal) main stem is determined by the median of all pin x-coordinates (y-coordinates) in the local wire network, with the top and bottom y-coordinates (x-coordinates) of the vertical (horizontal) main stem set to the maximum and minimum y-coordinates (x-coordinates) among all pins, respectively. As illustrated in Figure 2-2, the local wire network has 5 pins, where Figure 2-2(a) represents the single vertical main stem Steiner tree constructed for this local wire network, and Figure 2-2(b) represents the single horizontal main stem Steiner tree constructed for the same. The position of the selected main stem is determined

during the construction of the corresponding Steiner tree. After constructing the single vertical main stem and single horizontal main stem Steiner trees, their lengths are calculated and compared, and the Steiner tree with the smaller length is chosen, with its corresponding main stem becoming the line segment of the local wire network. As shown in Figure 2-1(c), line segments w_2 and w_3 are extracted from the global wire network n_1 , while w_4 represents the line segment extracted from the local wire network n_2 .

3.2 Track assignment routing evaluation metrics

This study evaluates the effectiveness of track assignment routing based on wirelength cost, overlap cost, blockage cost, via obstruction cost, planar anchoring cost, and via anchoring cost.

3.3 Comparison and analysis of experimental results for local wire networks.

The experiments were conducted using C/C++ language in a Linux environment with an Intel(R) Core(TM) i7-10750H CPU and 64GB RAM. The routability information of both global wire nets and local wire nets was considered. By extracting the routability information of local wire nets, the allocation of line segments in track assignment routing was maximized. Table 1 shows the extraction status of all wire segments in 10 benchmark circuits [2]. The layout results were obtained from the layout placer NTUplace4, and then the overall routing results of the benchmark circuits used in the experiments were obtained using the global router NCTUgr. N_1 represents the total number of wire nets for each benchmark circuit, while N_2 represents the number of local wire nets for each benchmark circuit. The proportion of local wire nets to the total number of wire nets is as follows:

$$N = \frac{N_2}{N_1} \times 100\% \quad (1-1)$$

I_1 represents the total number of extracted line segments for each benchmark circuit, while I_2 represents the number of line segments extracted from local wire nets for each benchmark circuit. The proportion of routability information from local wire nets to the total routability information is as follows:

$$I = \frac{I_2}{I_1} \times 100\% \quad (1-2)$$

The ratio I_0 between the number of line segments considering local wire nets and the number of line segments without considering local wire nets reflects the allocation of routability information in track assignment routing.

Table 1. Number of wire networks and extracted segments for each benchmark circuit.

Circuit	N_1	N_2	N/ %	I_1	I_2	I / %	I_0 / %
sb2	9908 99	304 025	30. 68	2134 703	304 025	14. 24	1.1 7
sb3	8980 01	339 078	37. 76	2017 313	339 078	16. 81	1.2 0
sb6	1006 629	350 741	34. 84	2037 526	350 714	17. 21	1.2 1
sb7	1340 418	440 442	32. 86	2883 168	440 442	15. 28	1.1 8
sb9	8338 08	317 925	38. 13	1671 043	317 925	19. 03	1.2 3
sb1 1	9357 31	275 351	29. 43	1811 827	275 351	15. 20	1.1 8
sb1 2	1293 436	361 900	27. 98	2719 712	361 900	13. 31	1.1 5
sb1 4	6198 15	178 114	28. 74	1261 573	178 114	14. 12	1.1 6
sb1 6	6974 58	227 932	32. 68	1371 803	227 932	16. 62	1.2 0
sb1 9	5116 85	187 695	36. 68	9956 11	187 695	18. 85	1.2 3
mean	—	—	32. 98	—	—	16. 07	1.1 9

From Table 1, it can be observed that the total number of wire nets for the benchmark circuit sb2 is 990,899, with 304,025 of them being local wire nets. The proportion of local wire nets to the total number of wire nets is 30.68%. Across all benchmark circuits, the average proportion of global wire nets to the total number of wire nets is 67.02%, while the average proportion of local wire nets is 32.98%. The average proportion of routability information extracted from local wire nets to the total routability information is 16.07%. The average ratio between considering routability information from local wire nets and not considering it is 1.19. This clearly indicates that there is a substantial amount of routability information in local wire nets. Ignoring this information would result in significant mismatch between global routing and detailed routing. Thus, it can be proven that considering local wire nets in the track assignment routing

stage is necessary in order to maximize the allocation of routability information.

4 Research on Track Assignment Routing Method Using a Generative Neural Network Model Based on CVAE

Existing track assignment routing algorithms mainly include the weighted bipartite matching-based track assignment routing algorithm^[3], negotiation-based track assignment routing algorithm^[4], and hybrid discrete particle swarm optimization-based track assignment routing algorithm^[5]. However, these algorithms still have limitations in solving large-scale track assignment routing problems.

Generally, allocation problems can be classified based on the nature of the objective function to be optimized, such as linear allocation problems, quadratic allocation problems, etc. Depending on the size of the allocation problem, it can be classified as two-dimensional allocation problems, three-dimensional allocation problems, etc. LSAP (Linear Sum Assignment Problem) is a special case of two-dimensional allocation problems with a linear objective function, which can be seen as a branch of classical combinatorial optimization problems.

Many resource allocation problems in VLSI can be modeled as LSAP problems. Similarly, track assignment routing problems can be modeled as two-dimensional LSAP problems, where the assignment scheme is obtained by generating the LSAP cost matrix and invoking the Hungarian algorithm for track assignment. However, traditional algorithms require a considerable amount of time to find the optimal solution when solving large-scale LSAP problems. LSAP problems are a special type of combinatorial optimization problem, and with recent breakthroughs in the application of deep neural networks to solving combinatorial optimization problems, there is significant potential for addressing large-scale LSAP problems. Additionally, deep neural networks are increasingly being applied in the field of Electronic Design Automation (EDA), as mentioned by Lee^[6] and others regarding the application of graph convolutional neural networks and DREAMPlace in Very Large Scale Integration (VLSI) circuits. Therefore, this study proposes a deep learning approach based on Conditional

Variational Autoencoder (CVAE) [7] to accurately solve track assignment routing problems based on LSAP.

The experimental research principle is as follows: The Conditional Variational Autoencoder (CVAE)-based generative neural network model takes the cost matrix and decision matrix of the LSAP-based track assignment routing problem as input data and transforms them into the form of the decision matrix for the LSAP problem. The CVAE generative neural network model with multiple hidden layers can be used to enhance feature extraction capabilities and effectively adapt to more data. As the number of layers increases, the model's training parameters also increase, so the minimum reconstruction error is introduced. This model can address the issue where traditional training techniques require a large number of training samples to cover the results when dealing with large-scale problems. It can generate more data samples following the same distribution as the given training dataset.

4.1 Model construction

In comparing the performance of three different network architectures of CVAE neural network models (i.e., CVAE-CNN, CVAE-FNN, and H-CVAE) in solving the LSAP-based track assignment routing problem, the H-CVAE generative neural network model was ultimately selected.

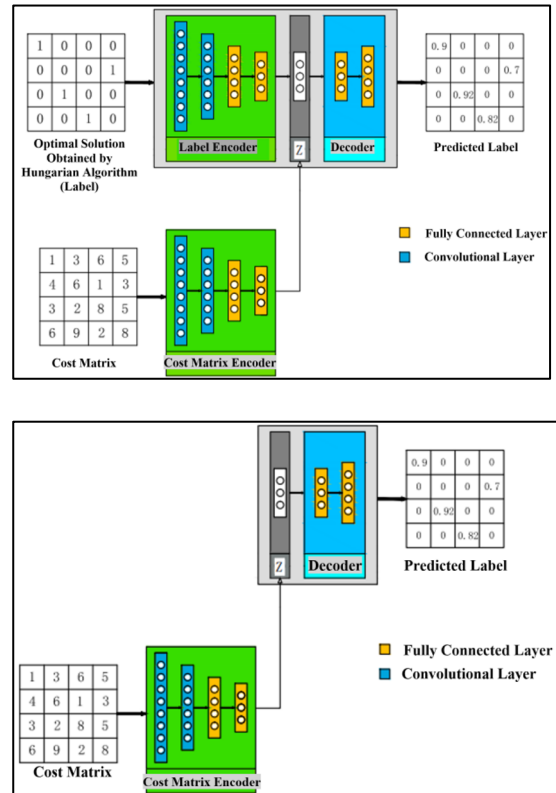


Fig.3: Generation neural network model based on H-CVAE.

In Figure 3, the left diagram illustrates the input of the generative neural network model based on H-CVAE, which consists of an $m \times m$ cost matrix and the decision matrix obtained from the Hungarian algorithm. The output is in the form of the decision matrix for the LSAP-based track assignment routing problem. The right diagram depicts the testing phase of the network model, where only the cost matrix is used as the input to the cost matrix encoder.

4.2 Dataset generation

To train the proposed Conditional Variational Autoencoder (CVAE)-based generative neural network model for solving the LSAP-based track assignment routing problem, we use MATLAB to generate a cost matrix with the same distribution as the track assignment routing problem. Then, we use the optimal solution obtained from the Hungarian algorithm as the decision matrix for this cost matrix. The resulting decision matrix and the cost matrix form a tuple, which is referred to as a sample. These samples are used as both the validation dataset and the training dataset for the generative neural network model.

4.3 Experimental result and analysis

The Conditional Variational Autoencoder (CVAE) models used in the experiments were implemented using Python programming language in a PyCharm environment, utilizing TensorFlow GPU 2.0 and Python 3.7. The hardware setup includes an Intel(R) Core(TM) i7-10750H CPU and an 8GB NVIDIA GeForce GTX 1650Ti GPU. In this research, nine generative neural networks were established, comprising three different network models (FNN, CNN, and hybrid neural network) of three different sizes (n=4, 8, and 16). All network models were initially trained for 180 iterations from scratch. The learning rate was set to 0.001 for n=4 and n=8, and 0.0001 for n=16. Adam optimizer was used, and the batch size was set to 128. All networks were trained based on the CVAE generative neural network model, consisting of one decoder and two encoders. Table 2 shows the number of layers and neurons per layer in the encoders of the H-CVAE generative neural network models corresponding to LSAP-based track assignment routing problems of different scales.

Table 2: Number of Layers and Neurons per Layer in the Encoder of the H-CVAE Generated Neural Network Model

n	Convolutional layers	Fully connected layers
4	256-128-64	512-256-128-64
8	1024-512-256-128-64	4096-1024-512-256
16	4096-2048-1024-512-128-64	4096-2048-1024-512-256

In comparison to the classification-based deep neural network model described in [8], our model does not use binary accuracy metric but employs a precision measurement algorithm suitable for the sparsity of the output decision matrix.

As shown in Figure 4, the x-axis represents the number of training iterations, and the y-axis represents the accuracy of the allocation scheme obtained compared to the Hungarian algorithm. The graph depicts the accuracy of LSAP-based track assignment routing problem solutions obtained during the training iterations of three different network architectures of the CVAE generative neural network model when solving an LSAP track assignment routing problem of size n=16. The red line represents the accuracy of H-CVAE, the blue line represents the accuracy of CVAE-CNN, and the purple line represents the accuracy of CVAE-FNN. The

simulation results indicate that CVAE-FNN, CVAE-CNN, and H-CVAE achieve convergence in accuracy at around 130 iterations, with final convergence accuracies of 83%, 88.7%, and 92.3%, respectively. These results suggest that the H-CVAE-based generative neural network model is more effective in extracting hidden features and their nonlinear relationships. Additionally, the accuracy of the H-CVAE-based generative neural network model surpasses the best results from the classification-based deep neural network model (which achieved 67.7% accuracy), as shown in Table 3.

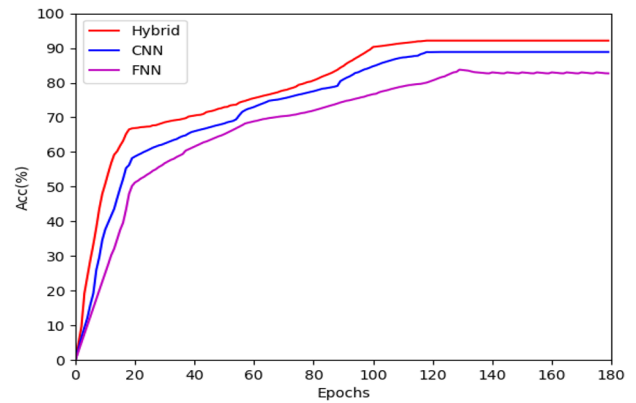


Fig. 4: Accuracy of three different network architectures of CVAE-based generative neural network models.

Table 3: Experimental results comparison between three different network architectures of CVAE-based generative neural network models and the classification-based deep neural network model.

n	DNN Accuracy (%)			CVAE Accuracy (%)			CVAE Runtime (ms)		
	FNN	CNN	Hybrid	FNN	CNN	Hybrid	FNN	CNN	Hybrid
4	90.8	92.76	92.2	94.5	97.46	97.46	3.2	5.5	6.9
8	72.3	77.8	89	93	94.5	94.5	3.6	6.9	7.2
16	59.8	67.7	83	88.7	92.3	92.3	4.2	7.1	7.6

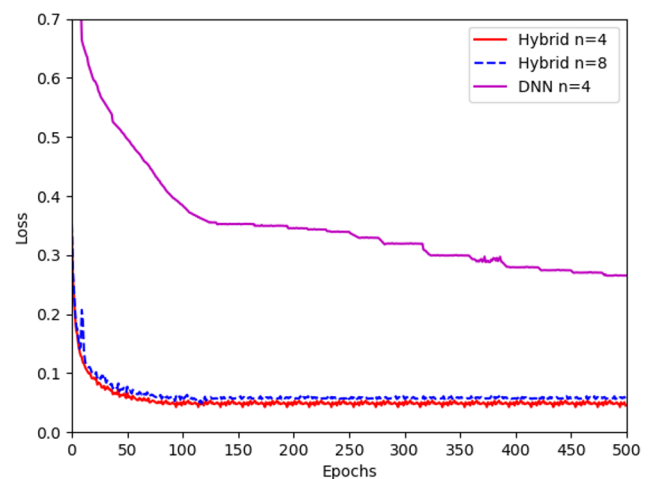


Fig.5. Convergence Comparison of Loss Functions between H-CVAE and Deep Neural Network (DNN).

As shown in Figure 5, the x-axis represents the number of training iterations, and the y-axis represents the loss value of the loss function. The purple line represents the convergence of the loss function for the classification-based deep neural network model when $n=4$, the blue dashed line represents the convergence of the loss function for the H-CVAE-based generative neural network model when $n=8$, and the red line represents the convergence of the loss function for the H-CVAE-based generative neural network model when $n=4$. The classification-based deep neural network model was trained with a learning rate of 0.001 and a batch size of 128.

The simulation results indicate that the convergence speed of the H-CVAE-based generative neural network model used in this study is much faster compared to the classification-based deep neural network model. In scenarios where the expected output results are sparse, the L2 regularization loss function employed by our generative model is suitable for the LSAP track assignment routing problem. The output of the generative model uses ReLU as the activation function, ensuring that the expected output is a matrix containing only 0 and 1. Additionally, it is evident that the H-CVAE-based generative neural network model can converge to lower error rates more quickly compared to the classification-based deep neural network model. Overall, these research findings demonstrate that employing the H-CVAE-based generative neural network model can ensure high accuracy in the track assignment routing phase while significantly improving routing efficiency.

5 Refined Strategy of Negotiation-based Tear-and-Redistribute Approach

The track assignment routing schemes generated by the Conditional Variational Autoencoder (CVAE)-based generative neural network model will result in lower wirelength costs but higher overlap costs. In this study, a refinement strategy based on negotiation-driven tear-and-reassign method is employed. This strategy sacrifices a smaller wirelength cost to optimize the overlap cost, aiming to strike a good balance between overlap cost and wirelength cost.

5.1 Segments requiring reallocation

The cost function used to select the segments that need to be reassigned is defined as follows:

$$\begin{aligned} \maxCost_{ir,t} = & \text{overlapCost}(ir, t) + \\ & \text{historyCost}(ir, t) \end{aligned} \quad (2-1)$$

Where $\text{overlapCost}(ir, t)$ represents the reduction in overlap cost when segment ir is removed from track t , and $\text{historyCost}(ir, t)$ denotes the sum of historical costs for all unit intervals covered by segment ir on track t . Each unit interval has a length of one unit, and segment ir covers a contiguous set of unit intervals on track t . At the beginning of the reallocation process, the historical cost for each unit interval on track t is initially set to 0. After each reallocation to track t , the historical cost for track t increases by 1, indicating overlap with the segment.

Through experimentation, it was observed that a greedy selection strategy tends to select the same set of segments, leading to a locally optimal allocation scheme. To avoid repeatedly splitting and reallocating the same set of segments, once a segment is split and reallocated, it must go through f iterations before it can be selected again as a segment requiring reallocation. In our implementation, f is set to 20.

5.2 Redistribution of segmented segments

The steps for reallocating segments involve using the following cost function to find the track with the minimum cost for reallocation:

$$\begin{aligned} \minCost = & 0.1 * \text{wlCost}(ir, t) + \alpha_1 * \\ & \text{overlapCost}(ir, t) + \beta * \text{blkCost}(ir, t) + \\ & \text{historyCost}(ir, t) + C \end{aligned} \quad (2-2)$$

Where $\text{wlCost}(ir, t)$ represents the wirelength cost incurred when segment ir is allocated to track t , considering all net components on the same routing track as ir . $\text{overlapCost}(ir, t)$ and $\text{blkCost}(ir, t)$ respectively denote how much the overlap and obstruction costs on track t increase when segment ir is allocated to it.

5.3 Experimental results comparison and analysis

This paper's proposed track assignment routing method was implemented using C/C++ and Python languages in a Linux environment on an Intel(R) Core(TM) i7-10750H CPU with 64GB of RAM. To obtain the overall routing results as test cases for the proposed track assignment routing method, layout results were obtained using the NTUplace4 layouter and then routed using the NCTUgr global router.

The refined strategy of negotiation-based tear-and-reassign involves tearing apart segments from the allocation scheme based on the cost function defined in Equation (2-1), reducing the overlap cost of the segments with the highest overlap, and then selecting the track with the minimum cost based on the cost function defined in Equation (2-2). After iterations, a refined routing assignment solution is obtained that further optimizes the overlap cost. To verify the feasibility of adding the negotiation-based tear-and-reassign refined strategy, experimental results were compared using evaluation metrics such as wirelength cost, overlap cost, and obstruction cost, with and without the refinement strategy.

As shown in Table 4, WL, OC, and BC represent the wirelength cost, overlap cost, and obstruction cost, respectively, in the assignment scheme without the refinement strategy. T-BC, T-OC, and T-WL represent the ratio of obstruction cost, overlap cost, and wirelength cost with and without the negotiation-based tear-and-reassign refined strategy. For example, for benchmark circuit sb2, the actual wirelength cost is 26.7608×10^6 , the actual overlap cost is 2.7146×10^6 , and the actual obstruction cost is 0.3370×10^6 . The average ratio of overlap cost with and without the refinement strategy is 73.97%, while the average ratio of wirelength cost increase is 6.67%. This indicates that the negotiation-based tear-and-reassign refined strategy reduces the overlap cost by an average of 26.03% while increasing the wirelength cost by 6.67%. Thus, it shows that further optimizing the overlap cost within the design rule constraints is feasible by sacrificing a minor increase in wirelength cost while maintaining the obstruction cost.

Table 4: Experimental Results Comparison Before and After Adding the Refined Strategy of Tear-Redistribute.

Circuit	Without refinement strategy			With refinement strategy					
	WL	OC	BC	WL	T-WL	OC	T-OC	BC	T-BC
sb2	25.3102	3.2571	0.3370	26.7608	1.0573	2.7146	0.8334	0.3370	1
sb3	24.1035	2.3856	0.3251	25.0249	1.0382	1.8217	0.7636	0.3251	1
sb6	24.5631	2.0139	0.2305	25.7926	1.0501	1.7159	0.8520	0.2305	1
sb7	37.2609	2.1508	0.2114	39.9673	1.0726	1.5297	0.7112	0.2114	1
sb9	19.9682	1.3726	0.1459	21.5739	1.0804	0.9246	0.6736	0.1459	1
sb11	21.9864	1.1431	0.2235	23.7826	1.0817	0.7649	0.6691	0.2235	1
sb12	36.0154	1.3967	0.0351	38.9625	1.0818	0.7982	0.5714	0.0351	1
sb14	15.2138	0.9687	0.2596	15.9367	1.0475	0.8019	0.8278	0.2596	1
sb16	16.2018	1.4935	0.1422	17.3691	1.0720	1.2317	0.8247	0.1422	1
sb19	12.2306	0.7218	0.0863	13.2716	1.0851	0.4837	0.6703	0.0863	1
Ratio	1	1	1	—	1.0667	—	0.7397	—	1

6 Conclusion

Through a series of experiments, it has been validated that the information regarding local routing networks cannot be ignored. This ensures that the track assignment routing phase can maximize the allocation of available routing information. Additionally, the CVAE-based generative neural network model has been shown to improve routing efficiency while maintaining high accuracy in solution generation. Moreover, the negotiation-driven tear-and-reassign algorithm has been employed to further optimize overlap costs by selecting segments with the highest reduction in overlap costs in the allocation scheme and reallocating them to tracks with the lowest cost, thereby sacrificing minimal wirelength costs.

Although the research on CVAE-based track assignment routing methods has yielded promising results, improving routing efficiency while balancing wirelength costs, overlap costs, and obstruction costs, the track assignment routing problem still faces significant challenges with the advancement of Very Large Scale Integration (VLSI) circuits. There are still many areas worthy of further research, such as the extraction of routing information, the efficiency of track assignment routing, and other design rule constraints in track assignment routing.

This study received support from two funds: (1) Hubei Province Teaching Research Project "Improving the Teaching Academic Ability of New Engineering Teachers under the Background of First-Class Engineering Construction" (2021107); (2) Wuhan University of Technology Independent Innovation Research Fund Project (No. 2020IVB071)

Reference

1. Xu Ning, Hong Xianlong. Physical Design Theory and Algorithm of VLSI [M]. Beijing: Tsinghua University Press,2009.
2. Viswanathan N, Alpert C, Sze C, et al. The DAC 2012 routability-driven placement contest and benchmark suite[C].DAC Design Automation Conference 2012,2012: 774-782.
3. Batterywala S, Shenoy N, Nicholls W, et al. Track assignment: A desirable intermediate step between global routing and detailed routing[C].Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design,2002: 59-66.
4. Wong M-P, Liu W-H, Wang T-C. Negotiation-based track assignment considering local nets[C].2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC),2016: 378-383.
5. Guo Wenzhong, Chen Xiaohua, Liu Genggeng et al. Trajectory allocation algorithm based on hybrid discrete particle swarm optimization [J]. Pattern Recognition and Artificial Intelligence,2019, 32 (08): 758-770.
6. Lee C-K. Deep Learning Creativity in EDA[C].2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT),2020: 1-1.
7. Kingma D P, Mohamed S, Jimenez Rezende D, et al.Semi-supervised learning with deep generative models[J].Advances in neural information processing systems,2014, 27.
8. Lee M, Xiong Y, Yu G, et al.Deep neural networks for linear sum assignment problems[J].IEEE Wireless Communications Letters,2018, 7 (6): 962-965.
9. Hsu M-K, Chen Y-F, Huang C-C, et al. Routability-driven placement for hierarchical mixed-size circuit designs[C].Proceedings of the 50th Annual Design Automation Conference,2013: 1-6.
10. Liu W H, Kao W C, Li Y L, et al.NCTU-GR 2.0: Multithreaded Collision-Aware Global Routing with Bounded-Length Maze Routing[J].IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,2013, 32 (5): 709-722.