Original scientific paper

https://doi.org/10.33180/InfMIDEM2023.302

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 53, No. 3(2023), 137 – 144

Parallel Routing for FPGA Using Improved Lagrange Heuristics with Sub-Gradient Method and Steiner Tree

Premalatha Balasubramaniam

Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore-14, Tamilnadu, India

Abstract: One of the most time-consuming processes in the Field Programmable Gate Array (FPGA) design cycle is the routing of the nets. For this reason, versatile Place and Route (VPR), a widely used algorithm, routes efficiently but takes a long time to complete. Using parallelization is one approach to quicken this design flow. A method based on Linear Programming (LP) might be employed to enhance the speed of this routing process further. This strategy, however, has two drawbacks: a local minimum and the solution's violation of boundaries. So, to overcome these issues, this paper proposed Parallel Routing for FPGA Using Improved Lagrange Heuristics with a Sub-Gradient method and Steiner tree (PR-ILH). The Lagrange relaxation process is enhanced by a series of innovative Lagrange heuristics presented in this work. Lagrange heuristics and sub-gradient optimization are both made use of by PR-ILH, which combines their advantages to provide more effective routing while reducing the complexity of parameter tuning. The use of the Steiner tree further improves the usage of resources and overall performance. It has also provided an improved method for updating the step size that this iterative process takes. Tests have been conducted on standard metrics, demonstrating that the proposed approach surpasses the ParaLaR and other existing improved techniques. Compared to ParaLaR, the proposed PR-ILH approach can enhance the minimum channel width and the violation of the restrictions by up to 25.1%. Compared to ParaLaR, PR-ILH realizes savings of roughly 15.4% in the total wire length. PR-ILH algorithm effectively reduces route latency, improving circuit speed and responsiveness in the FPGA routing process.

Keywords: Parallel Routing, FPGA, VPR, Sub-gradient, Improved Lagrange Heuristics, Steiner tree.

Vzporedno usmerjanje FPGA-ja z uporabo izboljšane Lagrangeove hevristike s podgradientno metodo in Steinerjevim drevesom

Izvleček: Usmerjanje mrež je eden najbolj zamudnih procesov v načrtovalskem ciklu FPGA (Field Programmable Gate Array). Zaradi tega pogosto uporabljen in vsestranski Place and Route (VPR) algoritem usmerja učinkovito, vendar časovno potratno. Uporaba paralelizacije je eden od pristopov za pospešitev tega načrtovanja. Za nadaljnje povečanje hitrosti tega procesa usmerjanja bi lahko uporabili metodo, ki temelji na linearnem programiranju (LP). Ta strategija pa ima dve pomanjkljivosti: lokalni minimum in kršitev robnih pogojev. Da bi premagali ta vprašanja, članek predlaga vzporedno usmerjanje za FPGA z uporabo izboljšane Lagrangeeve hevristike z metodo podgradienta in Steinerjevim drevesom (PR-ILH). Lagrangeev sprostitveni proces je izboljšan z nizom inovativnih Lagrangeovih hevristik. Lagrangevo hevristiko in podgradientno optimizacijo uporablja PR-ILH, ki združuje njune prednosti za zagotavljanje učinkovitejšega usmerjanja in hkrati zmanjšuje kompleksnost prilagajanja parametrov. Uporaba Steinerjevega drevesa dodatno izboljša uporabo virov in splošno učinkovitost. Opravljeni so bili testi na standardnih metrikah, ki dokazujejo, da predlagani pristop presega ParaLaR in druge obstoječe izboljšane tehnike. V primerjavi s ParaLaR lahko predlagani pristop PR-ILH poveča minimalno širino kanala in kršitev omejitev za do 25,1 %. V primerjavi s ParaLaR, PR-ILH realizira približno 15,4 % prihranek pri skupni dolžini žice. Algoritem PR-ILH učinkovito zmanjša zakasnitev poti, izboljša hitrost vezja in odzivnost v procesu usmerjanja FPGA.

Ključne besede: Vzporedno usmerjanje, FPGA, VPR, podgradient, izboljšana Lagrangeeva hevristika, Steinerjevo drevo.

** Corresponding Author's e-mail: premalatha.b@cit.edu.in*

How to cite:
P. Balasubramaniam, "Parallel Routing for FPGA Using Improved Lagrange Heuristics with Sub-Gradient Method and Steiner Tree", Inf. Midem-J. Microelectron. Electron. Compon. Mater., Vol. 53, No. 3(2023), pp. 137–144

1 Introduction to FPGA routing

Moore's law states that an integrated circuit's transistor count doubles roughly every two years. One of the most demanding processes in the FPGA [1] design cycle is the routing of nets, which are groups of two or more linked devices. One barrier to the broad usage of FPGA technology is the slow performance of conventional CAD software. Routing is frequently the most crucial stage in a normal CAD flow concerning runtime and overall performance since it directly impacts the clock frequency that may be used. Almost all contemporary FPGA routers can be traced back to 1995 by introducing the PathFinder algorithm [2].

Therefore, it is necessary to create fast routing algorithms that address the issue of the growing number of transistors per chip and, as a result, the lengthened runtime of FPGA-CAD tools. There are two methods to do this. First, run the routing algorithms in parallel on systems with several cores. The pathfinder method [3] is essentially sequential among the most popular FPGA routing algorithms. Therefore, this method is unsuitable for parallel running the FPGA routing algorithms.

Second, users may split their designs, build each component one at a time, and combine all the pieces to create the complete design rather than compiling it all at once. This strategy has been suggested in [4]. There is no certainty that there will be balanced partitioning because another could own the routing assets needed by one partition. In other words, this strategy must address the issues when routing resources are shared.

The suggested routing method is built on the Improved Lagrange Heuristics. By iteratively changing the Lagrange multipliers for every constraint, the Lagrange heuristics have previously been used in FPGA routing to produce workable solutions [5]. Innovative techniques are added to the Lagrange heuristics in the present work to address the intrinsic complexity of contemporary FPGA designs. The suggested approach considerably speeds up the routing process while maintaining the ability to provide superior routing alternatives by using the parallelism present in FPGA systems.

The Sub-Gradient approach, a practical optimization approach, is used to improve the effectiveness of the routing process further [6]. The suggested strategy effectively examines the solution space by utilizing sub-gradients, gradually getting closer to almost ideal routing topologies. With this improvement, the FPGA router can handle more complicated designs with less computing effort and reach quicker convergence.

Another crucial component of the suggested strategy is the development of the Steiner tree. Wire length and communication delays are significantly decreased by using Steiner trees for connecting logic components efficiently and compactly [7]. The parallel routing approach displays its capacity to produce routing strategies that maximize crucial parameters, such as wire length and signal propagation time, by including Steiner tree building into the enhanced Lagrange heuristics and sub-gradient method. The suggested way demonstrates its superiority over conventional routing strategies through thorough simulations and evaluation across multiple FPGA designs, making it an intriguing contender for enhancing the performance of contemporary FPGA-based systems.

The rest of the paper has been organized as follows: Section 2 describes related research on heuristic modelling for parallel FPGA routing. Section 3 Parallel Routing for FPGA Using Improved Lagrange Heuristics with Sub-Gradient method and Steiner tree (PR-ILH). Results and discussion have been given in section 4. Finally, the conclusion, limitations, and scope for further research have been shown in section 5.

2 Related works on heuristic modelling for parallel FPGA routing

The success of the semiconductor sector over the past fifty years has primarily been attributed to the Electronic Design Automation (EDA) method. However, routing accounting for a sizable portion of this takes a lot of time. This work concentrates on a sizable portion of this issue, namely the expensive FPGA routing procedure. Heuristic modelling for parallel FPGA routing has received a lot of research attention.

A parallel FPGA router based on Lagrangian relaxation was suggested by Hoo et al. (2015) called ParaLaR [8]. The routing task was divided into minor problems that could be handled in parallel using Lagrangian relaxation. Compared to conventional approaches, the ParaLaR router reduced routing time by an average of 20%, considering several benchmarks. A real-time Monte Carlo optimization method for FPGA was provided by Lee and Kim (2019) to create efficient and dependable message chain topologies [9]. They used Monte Carlo simulations to optimize the message chain topology in FPGA designs and increase performance and reliability. The suggested method was tested on several FPGA architectures and showed a mean performance gain of 15%. A novel face algorithm employing Lower-Upper (LU) factorization for LP was presented by PAN (2020) [10]. The suggested approach sought to resolve LP issues using LU factorization techniques effectively. In several LP examples, the algorithm's performance was evaluated, and it produced results that were competitive with those of already available methods.

Fuzzy Linear Programming (FLP) problems were addressed by models and methods provided by Ghanbari et al. in 2020. To deal with limitations and goal function inconsistencies, the study developed an FLP technique [11]. Various FLP situations were used to test the suggested models and remedies, demonstrating how well they handled uncertainty. For multi-FPGA systems, Pui and Young (2020) presented a Lagrangian relaxation-based Time-Division Multiplexing (TDM) optimization [12]. To maximize resource efficiency and minimize communication overhead, their solution used Lagrangian relaxation to optimize the objectives of multi-FPGA systems. Two benefits of this method are the potential to reduce connectivity constraints and make optimum use of resources in multi-FPGA systems. The requirement for meticulous parameter tuning in the Lagrangian relaxation process is a constraint, though.

The parallel FPGA router Agrawal et al. (2018) developed uses the Steiner tree and sub-gradient approach [13]. By applying the sub-gradient approach to enhance Steiner trees, the method attempted to parallelize the routing procedure. Several metrics were used to assess the parallel router, which showed a 30% decrease in routing time compared to sequential techniques. Using the primal-dual sub-gradient approach, Agrawal et al. (2019) proposed ParaLarPD, a parallel FPGA router [14]. The primary goals of the strategy were to parallelize the router and apply the primal-dual sub-gradient technique to improve Steiner trees. On several evaluations, the ParaLarPD router reduced routing time by 25% compared to sequential methods. The authors suggested a parallel FPGA router called ParaLarH, based on Lagrange heuristics, in [15]. The approach sought to improve FPGA routing by parallelizing the router and applying Lagrange heuristics. On numerous evaluations, the ParaLarH router significantly shortened the routing time compared to sequential techniques.

The studies above suggest multiple parallel FPGA routers using optimization methods, including Steiner trees, sub-gradient techniques, and Lagrangian relaxation. There are, however, several flaws that these studies have. Firstly, the complete examination of various FPGA designs and circumstances is lacking, which hurts the generalizability and applicability of the suggested methodologies. Second, the techniques are less appropriate for resource-constrained FPGA designs due to the additional complexity of parallelization, which may require significant FPGA resources and provide implementation difficulties. Furthermore, tuning and parameter sensitivity concerns may impact the efficiency of some heuristic-based systems. A unified and thorough methodology has been required to overcome these shortcomings and enhance the effectiveness of FPGA routing. So, the PR-ILH approach has been suggested in this paper.

3 Parallel routing for FPGA using improved lagrange heuristics with sub-gradient method and steiner tree (PR-ILH)

Lagrange heuristics and sub-gradient optimization are both made use of by PR-ILH, which combines their advantages to provide more effective routing while reducing the complexity of parameter tuning. The use of Steiner trees further improves the usage of resources and general efficiency. PR-ILH guarantees increased performance, decreased latency, and wire length reductions by completing thorough assessments of various FPGA designs and scenarios. In the end, PR-ILH solves the shortcomings of prior works by providing a more usable, effective, and flexible parallel routing solution for FPGA designs.

A weighted matrix grid MG(Vx, Ed) with a collection of vertices Vx and edges Ed, where a cost is linked with each edge, is a common formulation for the routing problem in FPGA or EDA. There are three different kinds of vertices in this grid matrix: Net Vertices (NV), Steiner Vertices (SV), and Additional Vertices (AV). A set N⊆Vx that contains all of the NV is how a net is defined. The route of a net, or a sub-tree ST of the grid matrix MG, is built using an SV, which is not an integral component of the NV. The term "Steiner tree" can also refer to a net tree.

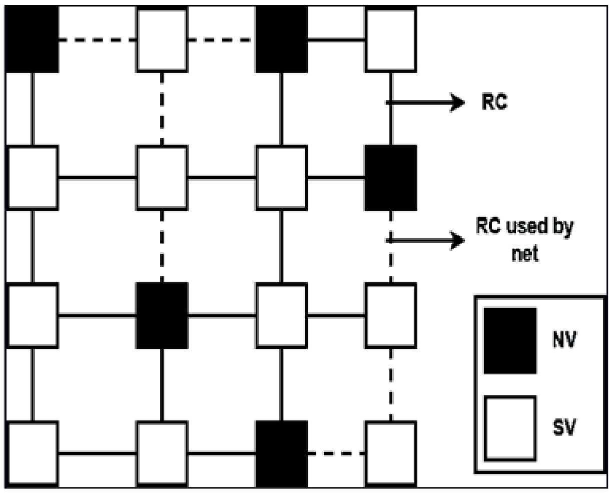

Figure 1: A weighted matrix grid MG(Vx, Ed)

An example of a 4x4 matrix grid is shown in Figure.1. In Figure 1, the NV is represented by the green colour squares, the SV by the blue colour squares, and the AV by the white colour squares. The horizontal and vertical lines represent the edges; these edges carry a cost, which is not indicated here. The dotted borders depict two net trees. The continuous lines represent the Routing Channel (RC), and the dotted lines represent the RC used by the net.

Each net's vertices and a total number of nets are specified. Finding a route for every net must be done so that the sum of all the routes minimizes the matrix MG overall path cost. Here, reducing the required channel width for every edge is also important. The routing of nets problem is presented as the following objective function (LP problem) to accomplish the two goals mentioned above:

$$F\left(y\right) = Min_{y_{ed,j}} \sum_{j=1}^{N_{net}} \sum_{ed \in Ed} C_{ed} y_{ed,j} \qquad \text{(1a)}$$

$$\sum_{j=1}^{N_{net}} y_{ed,j} < T, y_{ed,j} = 0/1 \qquad \text{(1b)}$$

where, $NA_j y_j = b_j, j = 1, 2, \ldots . N_{net}$

N_{net} is the number of nets, *ed* is the set of edges, and C_{ed} is the cost or latency linked with each edge Ed. The proposed optimization issue seeks to reduce the overall route cost of FPGA routing. NA_j is the node-arch occurrence matrix, b_j is the demand or supply vector, and y_i is an array of all $y_{ed,j}$ for net j corresponding to the jth net's path tree. $y_{ed,j}$ is the choice factor that signals whether an edge (routing channel) *ed* is employed by the net j (value 1) or not (value 0). The channel width restrictions, or disparity constraints, limit the number of nets that can use an edge to a constant T (which is also consistently reduced). The equality requirements, which ensure that a legitimate route tree is created for each net, have been inherently met by the technique to solve the problem. The LP mentioned above must be parallelized to locate a workable path for each net effectively. The following discussion will focus on the two primary difficulties present.

To limit the Channel Width (CW) of routing pathways in a design, CW restrictions are applied to the basic objective function in optimization problems. The purpose is to identify an ideal solution that minimizes or maximizes the objective function while satisfying these limitations. The Lagrange relaxation method addresses constraints in optimization problems by transforming restrictions into penalty factors inside the objective function. As a result, the limitations might be loosened up and considered indirectly throughout the optimization procedure. For every constraint, the approach entails the introduction of Lagrange multipliers, also referred to as dual factors. When the restrictions are broken, these Lagrange multipliers are weights that punish the goal function. For an LP issue with an objective function F(y) and CW constraints $\sum\left(j-1\right)\left(N_{net}\right)\left(y_{ed,j}\right)-T$ where $y_{ed,j}$ signifies the decision variables, the revised LP with Lagrange relaxation is given as ∂_{Ed} times (Lagrange multiplier) the CW constraints and has been characterized as:

$$Min_{y_{ed,j}} [\sum_{j=1}^{N_{net}} \sum_{ed \in Ed} C_{ed} y_{ed,j} + \sum_{ed \in Ed} \partial_{Ed} (\sum_{j=1}^{N_{net}} y_{ed,j} - T)] \qquad \text{(2)}$$

$$\left[NA\right] - jyj = b - j, j = 1, 2, \ldots\ldots.N, \partial_{ed,j} \geq 0$$

where, ⟦NA⟧_j y_j=b_j,j=1,2,….N_net,∂_Ed≥0

The Lagrange multiplier linked to the CW restriction

$$\left(\sum_{ed \in Ed} C_{ed} \partial_{ed} \sum_{(j=1)(N_{net})} [y_{ed,j} - T] \right)$$

is represented by the symbol ∂_Ed in the equation (2). The CW constraints in the problem are represented by the $\sum_{ed \in Ed} C_{ed} \partial_{ed}$. By including penalty factors proportionate to each constraint's violation, the Lagrange relaxation effectively integrates the restrictions into the objective function F(y).

The Simplex technique or interior-point methods are popular optimization algorithms used to find the solution to the modified LP with Lagrange relaxation. The technique converges to an ideal state that meets the CW constraints while lowering the objective function by iteratively changing the Lagrange multipliers and the decision factors. Lagrange relaxation is a versatile and powerful method for handling design constraints in optimization problems that may be used for a variety of optimization problems in engineering, operations management, and other disciplines.

The choice factors $y_{ed,j}$ present the second difficulty in solving the LP given by (1) or the revised LP with the Lagrange multiplier given by equation (2). As mentioned previously, if the net j uses the edge Ed, then the choice factor $y_{ed,j}$ must take a value of either 0 or 1; otherwise, $y_{ed,j}$ must take a value of 0. This is a Binary Integer Linear Program (BILP), which cannot be addressed with standard techniques like the Simplex or interior point approach because it is non-differentiable. Sub-gradient-based techniques, the approximation approach, etc., are some ways to handle non-differentiable optimization problems.

3.1 Sub-gradient method

The techniques based on sub-gradients are frequently used for minimizing non-differentiable functions F(y). These are recursive and refresh the factor y as $y^{l+1} = y^{1}$ - lh^{l}, where l and h^{l} are the step size and a sub-gradient of the objective function at iteration l, respectively. A sub-gradient-based technique only yields the Lagrange relaxation multipliers rather than directly solving the LP provided in equation (2). The minimal Steiner tree approach is then applied in parallel for FPGA routing after this (i.e., after calculating Lagrange relaxation multipliers). The choice variables $y_{ed,j}$∈ 1,2,….N_net in this situation can only have binary values. Sub-gradient methods alone will not always produce binary outcomes.

3.2 Steiner tree approach

Determining the shortest tree that links a specific number of nodes (terminals) in a graph is a common optimization issue that may be solved using the minimum Steiner tree methodology. Due to the Steiner tree problem's NP-hardness, it can be time-consuming and costly to compute the precise answer for large graphs. In the framework of FPGA routing, the graph depicts the routing architecture of the FPGA, and the terminals stand in for the source and destination locations that have to be linked.

The minimum Steiner tree method seeks to concurrently optimize the routing pathways for multiple links when used in parallel for FPGA routing. It considers all source and destination pairs and identifies the shortest trees that effectively link them. This method is particularly helpful in FPGA routing, where several routing pathways between different design elements must be built. The FPGA router can effectively determine the best routing pathways for multiple links simultaneously by addressing the minimum Steiner tree issue in parallel. This decreases the total routing time and boosts the efficiency of the FPGA design.

4 Improved lagrange heuristics with sub-gradient method and steiner tree (PR-ILH)

At first, the ParaLarPD technique described in [14] was used to perform FPGA routing in the suggested PR-ILH strategy. Since the acquired solution frequently violates some restrictions, a heuristic has been designed that makes the impossible solution possible (i.e., addresses the problem of the constraint violation). The Lagrangian multipliers are introduced, and the sub-gradient approach and minimal Steiner tree method are used to solve them. As anticipated, the discovered solutions are not always workable. Thus, a Lagrange heuristic has been developed. This allocates the likelihood of the restrictions violation based on particular solution aspects.

Figure.2 depicts the Lagrange heuristics verified with a sub-graph. The fundamental Lagrangian heuristic to fix the constraints violation in [14] comprises the following phases. Here, at first, these phases are described with the help of the example in Fig. 2, and then an algorithm is provided. For example, the channel widths calculated by [14] have been written adjacent to the relevant edge in fig. 2. Three edges have been present where the constraints are violated since T is assumed to be forty, and the three edges in Fig. 2 that are depicted as bolded lines, such as AE, BF, and DH.

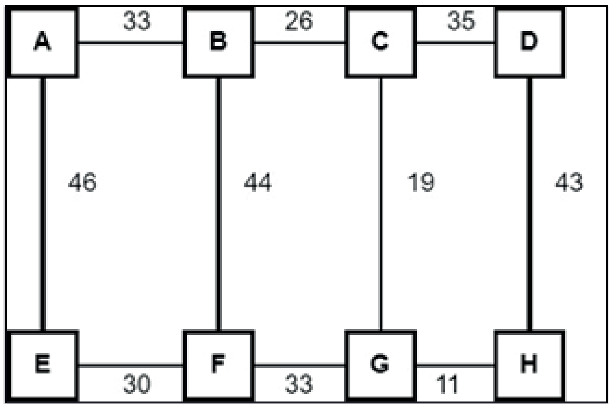

Figure 2: The Lagrange Heuristics verified with a sub-graph

Step 2: Calculate each edge's ability to route more nets down the new route without violating the limitations. The lowest of these capabilities is the Threshold and is employed when in the future. Mathematically, the Threshold has been given as follows:

$$\text{Threshold} = minimize\left[T - \sum_{j=1}^{N_{net}} y_{ed_l,j}\right], \quad (3)$$

l∈(Ed in the new route)

From equation (3), the value of Threshold is found to be 8.

Step 3: The amount of constraint violation has been given as:

$$f = \sum_{j=1}^{N_{net}} y_{ed,j} - T \quad (4)$$

Compute the number of nets where the constraints-violating edge has to be substituted with the chosen new route for the edge being considered, Ed. It is calculated such that no Ed in the new path has the constraint violation and is given as

$$R = minimum\left(Threshold, f\right) \tag{5}$$

For the edge BF, based on equation (4) f=44-40=4 and from equation (5), R=minimum (8,4) = 4.

Step 4: In R number of nets, finally, swap out this edge under investigation with the chosen path. This equates to changing BF in three nets from BF to BC→CG→GF in this case.

Step 5: The restriction violation in the edge under examination has not been removed entirely if the Threshold in equation (5) is less than f. If this is the case, the search for a different route must be restarted until the violation is eradicated or a different route cannot be found.

The processes as mentioned above have been repeated for each edge that deviates from the restrictions. The minimal CW is directly involved in this violation. . For large optimization problems, Lagrangian relaxation may be used to take advantage of the problem's structure and generate constraints on the optimum goal. These limitations form the backbone of several numerical algorithms and serve as an indicator of the algorithm's overall performance. Many heuristics and approximation methods have a Lagrangian solution as their starting point or reference point

5 Results and discussion

Tests have been carried out on a computer with a single Intel(R) Xeon(R) CPU E5-1620 v3 operating at 2.5 GHz and 64 GB of RAM. The kernel version is 3.13.0-100, and the operating system is Ubuntu 14.04 LTS. The code is written in C++11 and compiled with GCC version 4.84; the code that has been compiled is then executed utilizing various thread counts. ParaLaR [8], ParaLarPD [14], and ParaLarH [15] methods have been compared to the proposed approach. ParaLaR and VPR, 7.0 from the Verilog-to-Routing (VTR) package, were built using the same GCC version for comparison. Several settings in the input-output pad and the configuration logic blocks (CLBs) of ParaLarPD and ParaLaR have been updated to operate them identically as in the proposed model. MCNC benchmark circuits [16], which come in various sizes for the logic blocks, have been evaluated. The maximum number of sub-gradient technique iterations that have been utilized is 45.

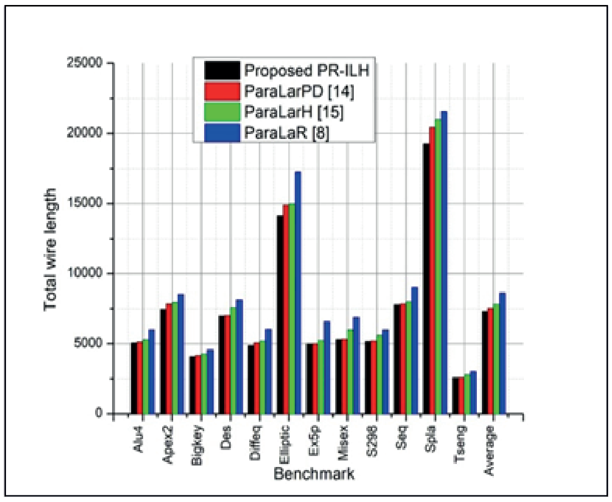

Figure. 3: Comparison of total wire length for the proposed PR-ILH, ParaLarPD [14], ParaLarH [15], and ParaLaR [8].

ParaLarH [15] and ParaLaR [8]. With an average total wire length of 7307.5, the suggested PR-ILH algorithm exhibits comparable performance across most benchmarks. In contrast, the average wire lengths for ParaLarPD [14], ParaLarH [15], and ParaLaR [8] are 7543.83, 7813.42, and 8632.42, respectively. The proposed PR-ILH represents a percentage reduction of approximately 15.4% compared to ParaLaR [8], which has an average total wire length of 8632.42. Additionally, PR-ILH reduces the overall length of the wire by around 3.3% and 6.0% when compared to ParaLarPD [14] and ParaLarH [15], respectively, underscoring its competitive advantage in wire routing optimization. Better

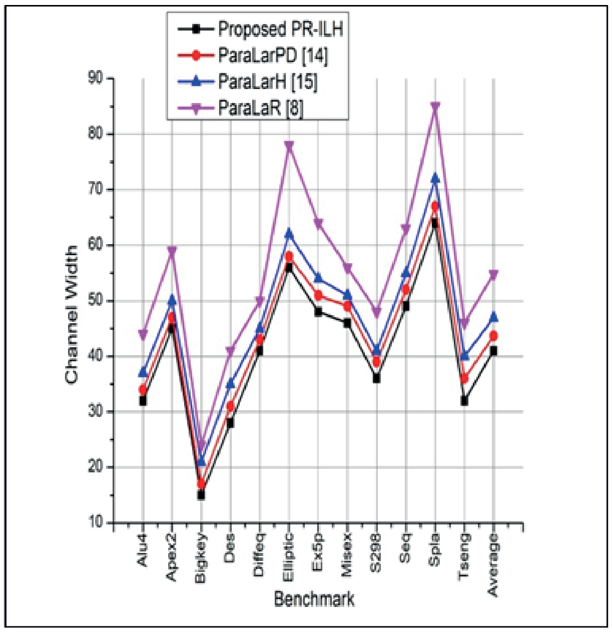

Figure 4: Comparison of Channel Width (CW) for the proposed PR-ILH, ParaLarPD [14], ParaLarH [15], and ParaLaR [8].

GPGA circuit designs with shorter wire lengths, which are essential for maximizing the effectiveness and productivity of FPGA routing, are indicated by lower overall wire lengths. The results reveal that the new PR-ILH algorithm outperforms some of the current ParaLaR-based approaches in this assessment, indicating that it has good potential for decreasing wire lengths. More in-depth study and benchmarking are required to draw firm conclusions on the algorithm's superiority over a wider range of circuits and design scenarios.

Fig. 4 shows the comparison of Channel Width (CW) for the proposed PR-ILH, ParaLarPD [14], ParaLarH [15] and ParaLaR [8]. As it directly affects the functionality and dependability of the circuits, CW is a critical component in the design of FPGA circuits. The suggested PR-ILH method outperforms ParaLarPD [14], ParaLarH [15], and ParaLaR [8], which have mean CW of 43.67, 46.92, and 54.83, respectively. Comparing this to ParaLaR [8], which has a mean CW of 54.83, results in an amazing percentage decrease of almost 25.1%. Additionally, PR-ILH reduces CW by around 6.5% and 12.0% when compared to ParaLarPD [14] and ParaLarH [15], respectively, demonstrating its capacity to create circuit topologies that are smaller and more efficient. It also shows good performance with an average CW of 41. Lower CW suggests more effective use of the area and resources, which is crucial for developing high-performance and compact FPGA routing circuits. The findings imply that the proposed PR-ILH algorithm excels at CW optimization, possibly making it a good contender for sophisticated FPGA circuit design applications.

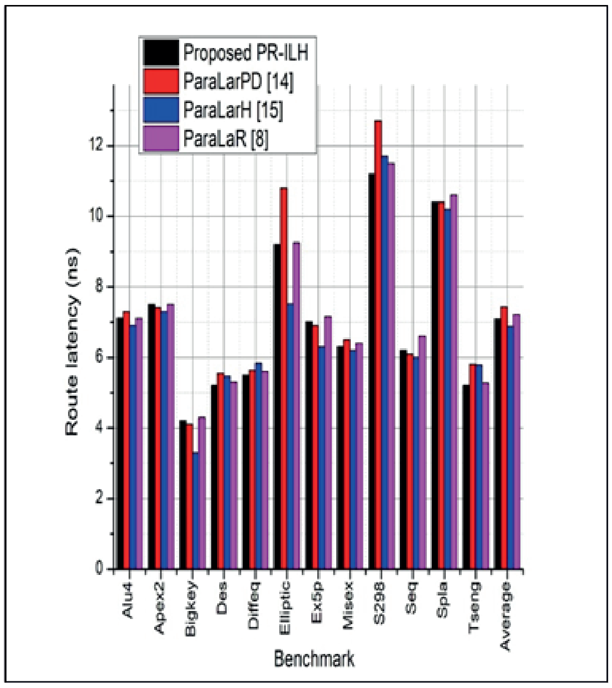

Figure 5: Comparison of route latency (ns) for the proposed PR-ILH, ParaLarPD [14], ParaLarH [15] and ParaLaR [8].

Fig. 5 shows the comparison of route latency (ns) for the proposed PR-ILH, ParaLarPD [14], ParaLarH [15] and ParaLaR [8]. The suggested PR-ILH method outperforms ParaLarPD [14], ParaLarH [15], and ParaLaR [8], which have mean route latencies of 7.433 ns, 6.875 ns, and 7.214 ns, respectively, is notable for its competitive average route latency of 7.085 ns. This shows that the PR-ILH algorithm effectively reduces route latency, improving circuit speed and responsiveness in the process. The lowest average route latency is demonstrated by ParaLarH [15], highlighting its competence in this area of circuit design. However, PR-ILH has the potential to be a dependable and effective method for lowering route latency and improving overall circuit performance because of its consistent performance across a range of evaluations.

6 Conclusion and scope of future work

This work proposed Parallel Routing for FPGA Using Improved Lagrange Heuristics with a Sub-Gradient method and Steiner tree (PR-ILH). Several novel Lagrange heuristics introduced in this study improve the Lagrange relaxation process. By combining the benefits of Lagrange heuristics with sub-gradient optimization, PR-ILH offers more efficient routing while lessening the complexity of parameter tweaking. Steiner trees are used to optimize resource use and overall efficiency further. Comparison of the proposed PR-ILH to ParaLaR (which has an average total wire length of 8632.42), indicates a decrease of about 15.4% in total wire length.

Additionally, compared to ParaLarPD and ParaLarH, PR-ILH shortens the wire's total length by around 3.3% and 6.0%, highlighting its competitive edge in wire routing optimization. Furthermore, PR-ILH decreases CW by around 6.5% and 12.0% when compared to ParaLarPD and ParaLarH, respectively, proving its ability to design circuit topologies that are more compact and effective. The additional effort required to implement the heuristic significantly reduces ParaLarH's parallelization speedups relative to ParaLarPD, which is readily remedied by adding more threads. In the future, we intend to focus on creating algorithms that would eliminate the violation of.constraints. The work can also be intended to adapt the proposed methods to the Internet of Things (IoT) industry, which faces significant design difficulties.

7 References

1. Jiang, Y., Chen, H., Yang, X., Sun, Z., & Quan, W.,." Design and Implementation of CPU &

FPGA Co-Design Tester for SDN Switches. Electronics",Vol.8,no.9,pp. 950.2019. https://doi.org/10.3390/electronics8090950

2. Zapletina, M. A., Zheleznikov, D. A., & Gavrilov, S. V. , "Improving pathfinder algorithm perfomance for FPGA routing", In 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering(ElConRus),pp.2054-2057,January 2019. IEEE. https://doi.org/10.1109/ElConRus51938.2021.9396608
3. McMurchie, L., & Ebeling, "PathFinder: A negotiation-based performance-driven router for FPGAs", In Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays,pp.111-117, February1995. https://doi.org/10.1109/FPGA.1995.242049
4. Zhou, Y., Vercruyce, D., & Stroobandt, D,"Accelerating FPGA routing through algorithmic enhancements and connection-aware parallelization",ACM Transactions on Reconfigurable Technology and Systems(TRETS),Vol.13,no.4,pp.1-26,2020. https://doi.org/10.1145/3406959
5. Martin, T., Barnes, C., Grewal, G., & Areibi, S.," Integrating Machine-Learning Probes in FPGA CAD: Why and How?", IEEE Design & Test,2023. https://doi.org/10.1109/MDAT.2023.3286334
6. Shen, M., Luo, G., & Xiao, N., "Coarse-grained parallel routing with recursive partitioning for FPGAs. IEEE Transactions on Parallel and Distributed Systems, Vol.32,no.4,pp.884-899,2020. https://doi.org/10.1109/TPDS.2020.3035787
7. Huang, P., Guo, L., Sun, L., & Zhang, X., "A Two-Stage Method for Routing in Field-Programmable Gate Arrays with Time-Division Multiplexing",Tsinghua Science and Technology, Vol.27,no.6,pp.902-911,2022. https://doi.org/10.26599/TST.2021.9010092
8. Hoo, C. H., Kumar, A., & Ha, Y, "ParaLaR: A parallel FPGA router based on Lagrangian relaxation", In 2015 25th International Conference on Field Programmable Logic and Applications (FPL)pp. 1-6. September 2015. IEEE. https://doi.org/10.1109/FPL.2015.7294012
9. Lee, H., & Kim, K, "Real-Time Monte Carlo Optimization on FPGA for the Efficient and Reliable Message Chain Structure. Electronics, Vol.8,no.8, pp:866,2019. https://doi.org/10.3390/electronics8080866
10. PAN, P. Q., "A new face algorithm using LU factorization for linear programming", Preprint, 2020. https://optimization-online.org/wpcontent/ uploads/ 2020/11/ 8085.pdf
11. Ghanbari, R., Ghorbani-Moghadam, K., Mahdavi-Amiri, N., & De Baets, B.," Fuzzy linear programming problems: models and solutions",Soft Com putting,Vol.24,no.13,pp.10043-10073,2020. https://doi.org/10.1007/s00500-019-04519-w
12. Pui, C. W., & Young, E. F," Lagrangian relaxation-based time-division multiplexing optimization for multi-FPGA systems",ACM Transactions on Design Automation of Electronic Systems (TODAES), Vol.25,no.2, pp.1-23,2019. https://doi.org/10.1109/ICCAD45719.2019.8942125
13. Agrawal, R., Hoo, C. H., Ahuja, K., & Kumar, A. (2018). Parallel FPGA router using sub-gradient method and Steiner tree. arXiv preprint arXiv:1803.03885. https://doi.org/10.3390/electronics8121439
14. Agrawal, R., Ahuja, K., Hau Hoo, C., Duy Anh Nguyen, T., & Kumar, A. (2019). ParaLarPD: Parallel FPGA router using primal-dual sub-gradient method. Electronics,8(12),1439. https://doi.org/10.3390/electronics8121439
15. Agrawal, R., Ahuja, K., Maheshwari, D., & Kumar, A.," ParaLarH: parallel FPGA router based upon Lagrange heuristics", arXiv preprint arXiv:2010.11893, 2020. https://doi.org/10.48550/arXiv.2010.11893
16. Yang, S.,"Logic synthesis and optimization benchmarks user guide: version 3.0", Research Triangle Park, NC, USA: Microelectronics Center of North Carolina(MCNC),pp.502-508,1991. https://api.semanticscholar.org/CorpusID:61228243

Arrived: 11. 08. 2023
Accepted: 14. 11. 2023