An Analytical-based Hybrid Algorithm for FPGA Placement

Chengyu Hu[1], Qinghua Duan[2], Liran Hu[1], Peng Lu[1],
Zhengjie Li[1], Meng Yang[1], Jian Wang[1], and Jinmei Lai[1] *
[1]Fudan University, [2]Chengdu Sino Microelectronic Technology Co., Ltd
[1]{15110720009, 17212020078,13110720020, 18112020031, mengyang, wjian, jmlai }@fudan.edu.cn,
[2]qh_duan@csmsc.com

ABSTRACT

As the capacity of FPGA increases, FPGA placers that adopt Simulated Annealing (SA) algorithm take more and more runtime. To solve this problem, this paper presents HCAS, a Hybrid algorithm Combining Analytical method and SA. There are three modifications in HCAS: (1) In global placement, faster and better result is realized by modified analytical algorithm. (2) In detailed placement, proper tradeoff is made between quality and runtime through improvement of SA. (3) Optimization workload of timing and wirelength is reasonably assigned between global and detailed placement according to algorithm features. HCAS is implemented in the newest VPR. Compared to VPR placer, it obtains a speedup of 11.1x, with 3% shorter wirelength and 5% smaller critical path delay. Compared to other analytical-based hybrid placers, HCAS achieves greater speedup and enhancement of placement quality is similar or better.

KEYWORDS

FPGA; Placement; Analytical; Simulated annealing; Hybrid algorithm;

ACM Reference format:

Chengyu Hu, Qinghua Duan, Liran Hu, Peng Lu, Zhengjie Li, Meng Yang, Jian Wang and Jinmei Lai. 2019. An Analytical-based Hybrid Algorithm for FPGA Placement. In Proceedings of 2019 Great Lakes Symposium on VLSI (GLSVLSI '19). *Tysons Corner, VA, USA,* 4 pages. https://doi.org/10.1145/3299874.3318035

1 INTRODUCTION

Simulated annealing is popular in the domain of FPGA placement, academic tool VPR [1] and commercial tool Quartus II [2] both adopt this algorithm. However, rapid growth of FPGA capacity makes SA placers take more and more runtime. Placement for some large benchmarks even takes several hours [3]. In order to reduce placement runtime without degrading placement quality, analytical-based hybrid algorithm is widely studied. It mainly includes two stages: global placement and detailed placement. In global placement, analytical algorithm is used to obtain a high-quality result in a short time. In detailed placement, algorithms of greedy or heuristic kind are applied to improve quality further.

In paper [4], hybrid placement algorithm includes three stages: building and solving analytical equation many times; legalizing analytical placement; low-temperature SA. Compared to VPR 4.3, a speedup of 5.8x is achieved and wirelength increase is 1.9%; In paper [5], global placement performs multiple analytical iterations and different types of blocks are solved and legalized separately. Then greedy algorithm is used in detailed placement. Compared to non-timing placer of Quartus II, a speedup of 4.1x is got at the cost of 5% longer wirelength. In paper [6], timing optimization is supported in global placement. During detailed stage, different algorithms are applied in different optimization targets. Compared to VPR 5.0, a speedup of 6.91x is achieved with 1% shorter wirelength and 7% smaller critical path delay. In paper [7], global placement aims at wirelength optimization. In detailed placement, cell swapping and cell matching are applied. Compared to VPR 7.0, a speedup of 3.07x is obtained with 6% shorter wirelength.

Although significant acceleration has been achieved in above researches, there is still some space for modifications: (1) global placement can be more efficient. On the one hand, it is not worth putting too much effort in solving analytical system [4] [5] [6] [7], because the analytical result will be broken by legalization. On the other hand, more considerations are needed for heterogeneous blocks, during building analytical system and legalizing analytical results [4] [5] [6] [7]. (2) Proper tradeoff between quality and runtime is required in detailed placement. Greedy-like algorithms are applied in [5] [7]. Solution space of them is small, leading to poor quality improvement. SA algorithm is used in [4] [6]. Its solution space is large, leading to much runtime. (3) Optimization workload should be assigned reasonably between global and detailed placement [4] [5] [6] [7]. Different Algorithms are used in these two stages, so optimization workload should be assigned according to algorithm features.

In order to overcome these shortcomings in previous work [4] [5] [6] [7] and achieve better speedup without quality degrading, this paper presents a modified hybrid placement algorithm, named HCAS. It adopts analytical algorithm in global placement and applies improved SA algorithm in detail placement.

2 OVERVIEW OF HCAS ALGORITHM

HCAS has been incorporated in the newest version of VPR [8]. As shown in Fig.1, process of HCAS placer can be divided into three stages: (1) random initial placement; (2) analytical global placement. (3) SA detailed placement.

Some strategies are adopted in global placement to get faster and better result. In building step, connections of heterogeneous blocks will get unbalanced weight. In solving step, runtime is reduced by gradient descent with momentum [9]; in legalization step, two-step method is used for heterogeneous blocks.

Some strategies are used in detailed placement to realize proper tradeoff between runtime and quality. Before inner loops of SA start, temperature is initialized by formulated method and move distance is initialized by self-adaptive method, in order to reduce solution space of SA. During inner loops, directed moves are applied to accelerate SA convergence.

In addition, analytic algorithm is connection-based and good at optimizing critical path delay, while SA is net-based and prefers reducing wirelength. Thus, global placement is assigned more timing optimization by dynamic weight factor of timing, while detailed placement is assigned more wirelength optimization by dynamic tradeoff factor of wirelength.

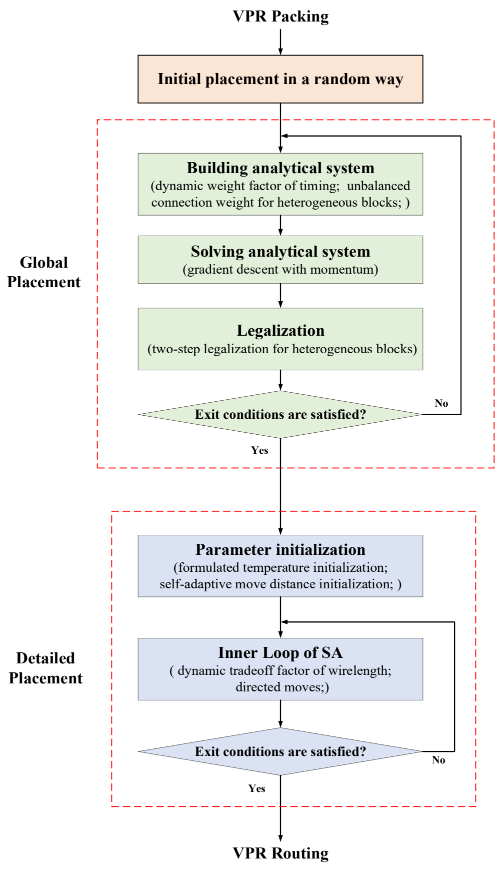
VPR Packing
Initial placement in a random way
Building analytical system
(dynamic weight factor of timing; unbalanced connection weight for heterogeneous blocks;)
Solving analytical system
(gradient descent with momentum)
Legalization
(two-step legalization for heterogeneous blocks)
Exit conditions are satisfied?
No
Yes
Global Placement
Parameter initialization
(formulated temperature initialization; self-adaptive move distance initialization;)
Inner Loop of SA
(dynamic tradeoff factor of wirelength; directed moves;)
Exit conditions are satisfied?
No
Yes
Detailed Placement
VPR Routing

Figure 1: Flow of HCAS

3 GLOBAL PLACEMENT

During global placement, analytical algorithm is modified to be more efficient and obtain more timing optimization.

3.1 Building Analytical System

Here, wirelength and timing cost are computed. Besides, each block is expected to move to its legal position of the last iteration, so legalization cost needs to be included. Directions of x and y can be solved separately in the same way. Cost function of "x" direction is described as formula (1).

$$\begin{aligned} COST(x) = &\sum_{ij\in C_{bb}} w_{bb}(i,j)\left(x_i - x_j\right)^2 \\ &+ \sum_{ij\in C_{timing}} w_{timing}(i,j)\left(x_i - x_j\right)^2 \\ &\sum_{i,i0\in C_{legal}} w_{legal}(i)\left(x_i - x_{i0}\right)^2 \end{aligned} \tag{1}$$

C_{bb}, C_{timing}, C_{legal} are sets of connections for wirelength, timing and legalization; $w_{bb}(i,j)$, $w_{timing}(i,j)$, $w_{legal}(i,j)$ are corresponding weights of connection between block i and block j; x_i and x_j are x coordinates of block i and block j; x_{i0} is legal coordinate of block i in the last iteration.

Bound2Bound net module [10] is used to compute wirelength cost. About Timing cost, connections on critical paths should be considered; timing weight of a connection is related to timing criticality [6]. About legal cost, it is taken into account after the second analytical iteration. Pseudo connection is built between the placement block and its legal position of the last iteration [5].

3.1.1 Unbalanced connection weight for heterogeneous blocks

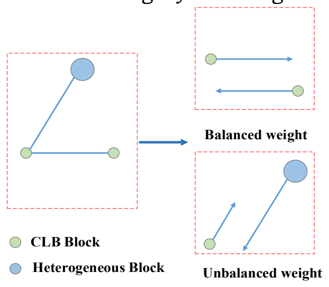

Figure2: Unbalanced connection weight for heterogeneous blocks

Heterogeneous blocks may deviate far from original analytical positions after legalization. This can result in increasing cost of connected CLB blocks. Therefore, unbalanced weight is designed for connections of a heterogeneous block and a CLB block. As shown in Fig.2, for a connection between two CLB blocks, we hope they will approach each other in solving step, so same weight will be added for the two blocks. However, for connection between a CLB block and a heterogeneous block, we hope a CLB block will approach other connected CLB blocks, rather than the connected heterogeneous block. Therefore, a smaller weight is added for the CLB block.

3.1.2 Dynamic weight factor of timing.

In global placement, timing optimization is favored by tuning timing weight factor. As shown in formula (2), timing weight factor is designed to increase dynamically with the number of iterations, so solutions of better timing can be found out when analytical algorithm is close to convergence.

$$\alpha = \alpha_0 + \mathrm{Min}(\alpha_{max},\ \mathrm{delta}_\alpha * (\mathrm{iter\_num} + 1)) \qquad (2)$$

3.2 Solving Analytical System

Gradient descent with momentum is used for solving analytical system. Its property of fast solving has been verified in [11]. The momentum is added to help stepping out of local optimal solution. Solving process can be formulized as formula (3) (4), where $\mathbf{x}$ is position vector and $\mathbf{v}$ is momentum.

$$\mathbf{x}_{iter\_num} = \mathbf{x}_{iter\_num-1} + \beta * \mathbf{v}_{iter\_num-1} \qquad (3)$$

$$\mathbf{v}_{iter\_num} = \mathbf{v}_{iter\_num-1} - \gamma * \nabla COST(\mathbf{x}_{iter\_num}) \qquad (4)$$

3.3 Legalization

At the beginning, a closest legal position is found for movable blocks and overlap is allowed. For "CLB" blocks, partition-based method [10] is adopted to eliminate overlap. For heterogeneous blocks, two-step method is applied separately for different block types: (1) all blocks of the same type are ranked according to their current cost. Then each block finds its best position in priority order. Overlap is forbidden here; (2) SA algorithm is used to further optimize positions of heterogeneous blocks.

4 DETAILED PLACEMENT

The target of detailed placement is improving global placement within acceptable time. This can be achieved by reducing solution space and accelerating search speed of SA process. Besides, more effort should be put in wirelength optimization.

4.1 Parameter Initialization

Temperature and move distance are critical parameters of SA. The former decides the probability that a block move of worse quality is accepted. The latter determines movable range of a block [1].

4.1.1 Formulated temperature initialization

$$T_0 = \frac{T_{vpr\_sa}}{\beta_0 * (1 + Num_{Blocks})} \qquad (5)$$

In detailed placement, initial temperature (T_0) should be smaller than normal SA algorithm. It decides the starting point of solution search. On the one hand, T_0 needs to be small enough to reject block moves that degrade global results seriously. On the other hand, T_0 cannot be too small to turn SA into greedy-like algorithm. Therefore, T_0 is designed as formula (5), where T_{vpr_sa} indicates initial temperature of VPR placer and β_0 is a factor.

4.1.2 Self-adaptive move distance initialization

Initial move distance (Rlim_0) decides size of solution space. Small Rlim_0 leads to small solution space and poor optimization, while large Rlim_0 results in great change of cost functions and low acceptance rate of block moves. Therefore, Rlim_0 should take a proper value. In experiments, if Rlim_0 makes move acceptance rate be 10%~15%, good quality is obtained. According to this clue, a self-adaptive method is proposed for Rlim_0 initialization: The detail is shown in Fig.3, where grid_height and grid_width indicate device size, inner_num means number of block move of one SA inner loop, a takes value about 10%~15%.

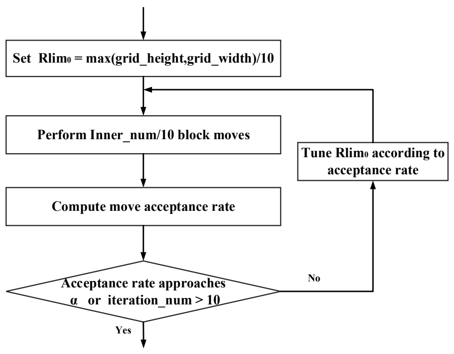
Set Rlim0 = max(grid_height,grid_width)/10
Perform Inner_num/10 block moves
Compute move acceptance rate
Acceptance rate approaches α or iteration_num > 10
No
Yes
Tune Rlim0 according to acceptance rate

Figure 3: Self-adaptive initialization of move distance

4.2 Inner loop of SA

Compared to VPR SA, HCAS made two modifications in inner loops. First, dynamic tradeoff factor of wirelength is used to achieve more wirelength optimization. Second, directed moves are applied to speed up search of solution space.

4.2.1 Dynamic tradeoff factor of wirelength

$$\Phi_{bb} = \Phi_0 + \frac{\mathrm{delta\_}\Phi}{1 + \beta * \mathrm{iter\_num}} \qquad (6)$$

Tradeoff factor of wirelength means proportion of wirelength cost to total placement cost. It will influence optimization bias. As shown in formula (6), it is designed to decrease with iteration number. Φ_0 is usually take the value of 0.5, $\mathrm{delta\_}\Phi$ can take the value of 0.1~0.2. This formula makes wirelength have more chances to be optimized in the early phase of detailed placement. In the final phase of detailed placement, timing will get same optimization as wirelength.

4.2.3 Directed Moves

As number of inner loops increases, acceptance rate of block moves becomes lower and lower. At this time, strategy of directed move is used to accelerate convergence. As shown in Table 1, there are two types of directed moves existing in inner loops. About directed move of timing, blocks of critical paths will be put into a set. One block will be randomly selected from the set, and then the block will be moved to a random position. As for directed moves of wirelength, after a block is randomly selected, a position will be randomly selected from its median region [12].

Table1: Two types of directed moves

	From block	To Position
Directed move of timing	Timing-critical	Random
Directed move of wirelength	Random	Median [12]

5 EXPERIMENTAL RESULTS

HCAS is implemented in the newest VPR [8]. Benchmark circuits are packed by VPR packer and placed by different placers (VPR Placer, HCAS Placer). Placement results (critical path delay, wirelength, and runtime) will be compared. All circuits placed by HCAS placer will be routed by VPR to verity their validity.

Hardware platform of our experiments is a quad-core CPU, named Intel(R) Core (TM) i7-4790. Operating system is a 64-bit Linux system (ubuntu-16.04.4). Seven largest benchmarks of VPR are selected. Resources of them are shown in Table 2.

Table2: Resources of benchmarks

Circuit name	Blocks and nets number of each circuit				
	#IO	#CLB	Mult_36	Memory	#Nets
bgm	289	16613	0	0	49032
LU32PEEng	216	7252	32	168	57053
LU64PEEng	216	14005	64	340	109854
LU8PEEng	216	30685	0	0	71378
mcml	69	6472	27	159	65271
stereovision1	278	9339	0	0	28968
stereovision2	331	16791	0	0	55545

5.1 Compared to VPR

In order to test performance, we run VPR placer in timing-driven mode (timing tradeoff factor is set to 0.5) and compared HCAS with VPR placer in three aspects: wirelength, critical path delay and runtime. As shown in Table 3, HCAS gets a speedup of 11.1x, with 3% short wirelength and 5% smaller critical path delay.

Table3: Comparison between VPR placer and HCAS placer

Circuit name	VPR Placer			HCAS placer		
	WL	CPD	RT	WL	CPD	RT
bgm	910	41.2	561	869	42.1	56
LU32PEEng	1519	74.8	694	1592	73.5	49
LU64PEEng	3775	77.9	1983	3765	72.8	162
LU8PEEng	1788	174.7	1616	1827	167.5	113
mcml	1056	45.2	647	950	46.3	44
stereovision1	404	15.6	210	378	13.9	26
stereovision2	769	32.1	583	726	28.0	70
Ratio Geomean	**1.03**	**1.05**	**11.1**			

(WL indicates wirelength and it has been divided by 1000, CPD and RT means critical path delay and run time, their measure unit is second)

5.2 Compared to Other Hybrid Placers

Here, HCAS placer is compared with other analytical-based hybrid placers in [4] [5] [6] [7]. Data of [4] [5] [6] [7] is got from comparison to SA placer. As shown in Table 4, HCAS placer outperforms all listed placers in speedup. Placement quality enhancement of HCAS is better than [4] [5] and similar to [6] [7]. This proves that strategies adopted in HCAS are effective.

Table4: Comparison HCAS placer with other hybrid placers

	[4]	[5]	[6]	[7]	HCAS
Speedup	5.8x	4.1x	6.91x	3.07 x	11.1x
Quality	WL: +1.9%;	WL: +5%	WL: -1% CPD: -7%	WL: -6%	WL: -3% CPD: -5%

(WL: wirelength, CPD: critical path delay, +: worse, - : better)

6 CONCLUSIONS

In this paper, an analytical-based hybrid algorithm named HCAS is presented for FPGA placement. HCAS improves performance in three aspects: (1) Global placement is sped by faster analytical solving method and better result is achieved by two heterogeneous strategies. (2) Detailed placement realizes quality refinement in acceptable runtime by decreasing solution space and accelerating search during SA. (3) Optimization workload is reasonably assigned according to algorithm features. Finally, HCAS obtains a speed up of 11.1x with 3% short wirelength and 5% smaller critical path delay, compared to placer of VPR. This verifies that HCAS outperforms previous hybrid placers [4] [5] [6] [7].

REFERENCES

[1] J. Luu, J. Goeders, M. Wainberg, A. Somerville, T. Yu, K. Nasartschuk, M. Nasr, S. Wang, T. Liu, N. Ahmed, K. B. Kent, J. Anderson, J. Rose and V. Betz "VTR 7.0: Next Generation Architecture and CAD System for FPGAs," ACM TRETS, Vol. 7, No. 2, June 2014, pp. 6:1 - 6:30.

[2] Adrian Ludwin and Vaughn Betz. 2011. Efficient and Deterministic Parallel Placement for FPGAs. ACM Trans. Des. Autom. Electron. Syst. 16, 3, Article 22 (June 2011), 23 pages. DOI=http://dx.doi.org/10.1145/1970353.1970355

[3] K. E. Murray, S. Whitty, S. Liu, J. Luu and V. Betz, "Titan: Enabling large and complex benchmarks in academic CAD," 2013 23rd International Conference on Field programmable Logic and Applications, Porto, 2013, pp. 1-8. doi: 10.1109/FPL.2013.6645503

[4] Yonghong Xu and M. A. S. Khalid, "QPF: efficient quadratic placement for FPGAs," International Conference on Field Programmable Logic and Applications, 2005., Tampere, 2005, pp. 555-558. doi: 10.1109/FPL.2005.1515784

[5] M. Gort and J. H. Anderson, "Analytical placement for heterogeneous FPGAs," 22nd International Conference on Field Programmable Logic and Applications (FPL), Oslo, 2012, pp. 143-150. doi: 10.1109/FPL.2012.6339278

[6] Tzu-Hen Lin, P. Banerjee and Y. Chang, "An efficient and effective analytical placer for FPGAs," 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2013, pp. 1-6.

[7] Y. Chen, S. Chen and Y. Chang, "Efficient and effective packing and analytical placement for large-scale heterogeneous FPGAs," 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, 2014, pp. 647-654. doi: 10.1109/ICCAD.2014.7001421

[8] VTR: https://github.com/verilog-to-routing/vtr-verilog-to-routing

[9] G. Liu, Z. Zhou, H. Zhong and S. Xie, "Gradient descent with adaptive momentum for active contour models," in IET Computer Vision, vol. 8, no. 4, pp. 287-298, August 2014. doi: 10.1049/iet-cvi.2013.0089

[10] M. Kim, D. Lee and I. L. Markov, "SimPL: An Effective Placement Algorithm," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 1, pp. 50-60, Jan. 2012. doi: 10.1109/TCAD.2011.2170567

[11] E. Vansteenkiste, S. Lenders and D. Stroobandt, "Liquid: Fast placement prototyping through steepest gradient descent movement," 2016 26th International Conference on Field Programmable Logic and Applications (FPL), Lausanne, 2016, pp. 1-4. doi: 10.1109/FPL.2016.7577303

[12] K. Vorwerk, A. Kennings and J. W. Greene, "Improving Simulated Annealing-Based FPGA Placement With Directed Moves," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 2, pp. 179-192, Feb. 2009. doi: 10.1109/TCAD.2008.2009167