Multi-Clustering Net Model for Placement Algorithms

Andrey Ziyatdinov
Universitat Politècnica de Catalunya
Barcelona, Spain

David Baneres
Universitat Oberta de Catalunya
Barcelona, Spain

Jordi Cortadella
Universitat Politècnica de Catalunya
Barcelona, Spain

***Abstract*—Current placement algorithms aim at routable layouts with shortest wirelength and mostly minimize the total Half-Perimeter Wirelength (HPWL) of nets. A new clustering net model is proposed for better handling of high degree hyperedges, for which the HPWL can significantly underestimate wirelength.**

Splitting a net into several lower degree subnets, the total HPWL of the subnets estimates wirelength significantly better than HPWL of the original net. An efficient clustering approach is proposed with complexity linear on the number of pins. The final Steiner tree wirelength is improved with no or little penalty in runtime by transforming the circuit netlist between global and detailed placement stages accordingly to the new net model. The reduction in the wirelength also leads to shorter delays of circuits.

I. Introduction

Placement and routing are the two steps that produce the physical layout of a circuit. The cost function of algorithms for physical layout incorporate factors responsible for the quality of the circuit, such as area, delay and routability.

The estimation of the routing cost is crucial during placement. On one hand, it is desirable to use models that provide an accurate estimation of the final wirelength. On the other hand, the models must be simple enough for the algorithms to have a manageable computational complexity. This trade-off is a continuous area of research in physical synthesis.

Half-perimeter bounding box wirelength (HPWL) is widely used in placement as an objective function, since it is easy to compute and it represents a reasonable first-order estimation for routed wirelength [1], [2]. The HPWL is exactly equivalent to rectilinear Steiner minimal tree (RSMT) cost for low-pin nets with two or three pins and it is approximately proportional for multi-pin nets [3]. Since the HPWL minimization placement problem is NP-hard [4] and inapproximable [5], placers optimize HPWL heuristically by applying such methods as min-cut partitioning, quadratic or analytical solvers, or simulated annealing. In order to produce routable designs, placers typically combine HPWL cost optimization with different congestion-aware techniques [6], [7], [8].

More accurate netlength estimation can be achieved by calculating Steiner tree wirelength (StWL). However, the construction of RSMT is an NP-complete problem [9] and, thus, too computationally expensive in placement, even using heuristics [10], [11]. In addition, the RSMT, that is not a convex objective, may imply mathematical obstacles in StWL optimization process. The most recent placer [12] that employs StWL cost function suffers from long runtime and highly depends on effectiveness of RSMT heuristic methods.

Since most of placement algorithms aim at HPWL minimization, we propose a *HPWL-based multi-clustering* net (MCN) model for better handling of high degree hyperedges. The main contribution of our work is formulated by the following statements.

- **Clustering approach.** We introduce a clustering technique for the netlength modeling problem and present an efficient implementation with linear complexity on the number of pins. When a net is split into several lower degree subnets, the total HPWL of the subnets approximates RSMT length significantly better than HPWL of the original net.
- **Practical StWL minimization.** By transforming a circuit netlist between global and detailed placement stages accordingly to the MCN model, the total StWL of all nets is improved with no or little penalty in runtime. The proposed approach neither depends on any placement algorithm or RSMT heuristic. In practice, it can be applied between any consecutive placement steps.

An overview of the approach is presented in Section II. Section III describes the MCN model. Finally, Section IV reports the experimental results.

II. Overview

The paper addresses the problem of wirelength evaluation for multi-pin nets (more than three pins) in placement. The traditional HPWL model is adequate for nets with two or three pins, but it can crucially underestimate wirelength for nets with more pins. To overcome the deviation, a high degree net is broken into several subnets by the clustering approach described in Section III. In order to estimate the netlength, the HPWL measure is applied to the resulting union of subnets.

On the MCN model, we assume that the pins with closest position form the subnets. Furthermore, the Steiner tree of the original net is likely to be within the HPWL bounding boxes of the subnets. The sum of the HPWL of the subnets also gives more accurate estimation, because the subnets have smaller pincount. Our empirical results demonstrate the efficiency of the MCN heuristic and prove more precision with regard to the HPWL.

Figure 1 illustrates how the wirelength is estimated for a net with six pins. Figure 1(a) depicts the RSMT with 17 units

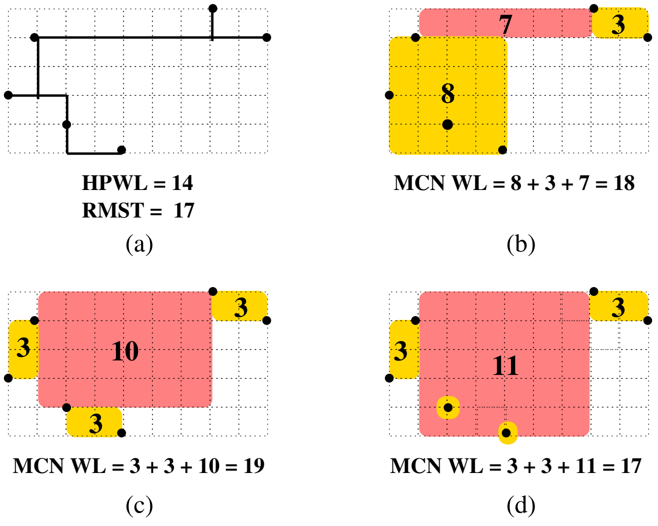


Fig. 1. Example of wirelength estimation.

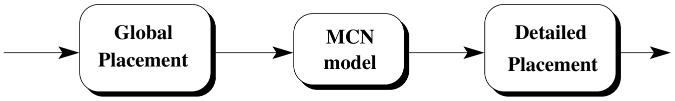


Fig. 2. Placement flow.

length, and the HPWL of 14 units undervalues the netlength. As shown in Fig. 1(b), the net is split into two subnets (light rectangles) and one additional subnet that connects them (dark rectangle). The total HPWL of the subnets is 18 units. Figures 1(c,d) depict the MCN approach for 4 and 5 subnets respectively. The MCN model is able to reach the exact wirelength of the RSMT in Fig. 1(d).

In the presented work, the effect of the MCN approach on real placement algorithms is examined by transforming a circuit netlist between global and detailed placement steps. The experimental scheme is based on the outline of Fig. 2.

The layout produced by global placement is captured and a new netlist is built, where each multi-pin net is substituted by the union of the subnets. The new netlist is passed to detailed placement that minimizes the wirelength accordingly to the MCN net model. Finally, our approach results in shorter Steiner tree wirelength in comparison with common global and detailed placement flow.

III. Multi-Clustering Net Model

The MCN model aims at obtaining a better approximation of the netlength for a given net. The accuracy of the wirelength estimation is increased by splitting the original net into several subnets.

An optimization algorithm has been designed to perform the partition. The technique selects the best set of subnets based on a local search algorithm. Iteratively, several groups of subsets of pins are explored and the best one is chosen. Internally, the problem is simplified to cluster the pins of the net by closest position. The well-known k-means algorithm [13] is used as the clustering technique. The construction of the subnets is straightforward from the subsets.

Algorithm 1 Clustering Algorithm

```
Require: A net $N$
Ensure: A group of subsets of pins
 1: Cost of clustering: $Cost \Leftarrow$ Cost of net $N$
 2: Number of subsets: $k \Leftarrow 2$
 3: repeat
 4:     centroids of subsets: $\{C_1,\dots,C_k\} \Leftarrow k$ random points
 5:     while changes in $\{S_1,\dots,S_k\}$ do
 6:         $\forall i \in \{1,\dots,k\}$, $S_i \Leftarrow$ Pins of net $N$ closer to $C_i$
 7:         $\forall i \in \{1,\dots,k\}$, $C_i \Leftarrow$ centroid of $S_i$
 8:     end while
 9:     $Cost \Leftarrow$ Cost of the $k$ subsets $\{S_1,\dots,S_k\}$
10:     $k \Leftarrow k+1$
11: until no improvement in $Cost$
12: return $\{S_1,\dots,S_k\}$
```

In this section, the k-means algorithm is initially presented because it is the core of our approach. Then, the algorithmic details of the optimization algorithm and the method applied to construct the subnets are described.

A. *k-means algorithm*

The k-means clustering algorithm [13] is commonly used in data mining where efficient algorithms were proposed to process large quantity of data [14]. The complexity of this algorithm is $O(kni)$, where k is the number of clusters, n is the number of points to be clustered, and i the number of iterations to converge. In our case, k is the number of subsets of pins and n is the total number of pins of the net, which are typically small. Experimentally, the algorithm converges very fast when n is small, thus showing linear complexity on n.

The k-means algorithm (lines 4-8 in Algorithm 1) aims at clustering the pins into k subsets. Initially, k pins are arbitrarily chosen as the potential centers of the clusters (also known as *centroids*) and the remaining pins are assigned to the cluster with the closest center. The centroid of each cluster is re-calculated at each iteration as the center of gravity of the components of the cluster. The calculation stops when a fixpoint is reached for all the centroids.

Figure 3 depicts an example of the k-means algorithm when two clusters ($k = 2$) are sought. A net with eight pins is depicted in Fig. 3(a). Figures 3(b,c,d) show the locations of the centroids (shadowed circles) and the two subsets S_1 and S_2 at each iteration[1]. The initial points A and B are selected randomly (Fig. 3(b)). These initial centroids classify the pins into two subsets: $S_1 = \{A,C\}$ and $S_2 = \{B,D,E,F,G,H\}$. After re-clustering, point B is moved to the cluster S_1 and convergence is reached.

B. *Clustering algorithm*

The algorithm is shown in Algorithm 1. The initial solution is the original net that corresponds to a single subset of all pins. The clustering technique iteratively explores several solutions by incrementing the number of possible subsets. The k-means algorithm is applied in the innermost loop to obtain

[1] To be precise, the figure shows the state of the loop after the calculation of S_1 and S_2 and before the re-calculation of the centroids.

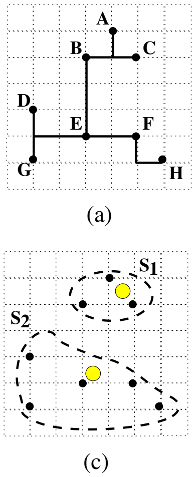


Fig. 3. Example of k-means algorithm when $k = 2$: (a) Initial net, (b,c,d) evolution of the k-means algorithm.

the clustering of the pins. Note that the optimal clustering is not guaranteed. The quality of the final solution depends on the initial centroids, which are commonly initialized to random coordinates.

To evaluate the clustering quality, we define a cost function that aims at minimizing the total inter- and intra-clustering variance:

$$Cost = \underbrace{\sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - C_i)^2}_{inter-clustering} + \underbrace{\sum_{C_i \in C} (C_i - C_T)^2}_{intra-clustering}$$

where k is the number of clusters, C_i is the centroid of the cluster S_i, and C_T is the center of gravity of the centroids of the clusters. The inter-clustering variance penalizes far-away pins and leads to more compact clusters. On the other hand, the intra-clustering variance constraints large connections among clusters.

The clustering algorithm stops when there is no improvement in the cost. Experimentally, we observed no significantly improvement on further exploration in larger number of subsets after a worst solution is found.

In terms of performance, the complexity considered to be linear with regard to the number of pins due to the small number of k-means algorithm runs and the small number of explored subnets that does not exceed 6 in practice and equals to 3-4 in average in the experiments on ISPD05 benchmarks.

Figures 4(a,b,c,d) show an example of the clustering algorithm. In the figures, the centroids of the subsets are labeled with C_i where i is the number of the subset and C_T corresponds to the center of gravity of the centroids. The two elements in the cost function, shown also in the figure, correspond to the inter- (light color) and intra-clustering (dark color) variance respectively. The original net is assumed as the initial solution in Fig. 4(a). Fig. 4(b) corresponds to the clustering into two subsets derived from the example shown in Fig. 3. The algorithm stops when $k = 4$ because the cost is not improved.

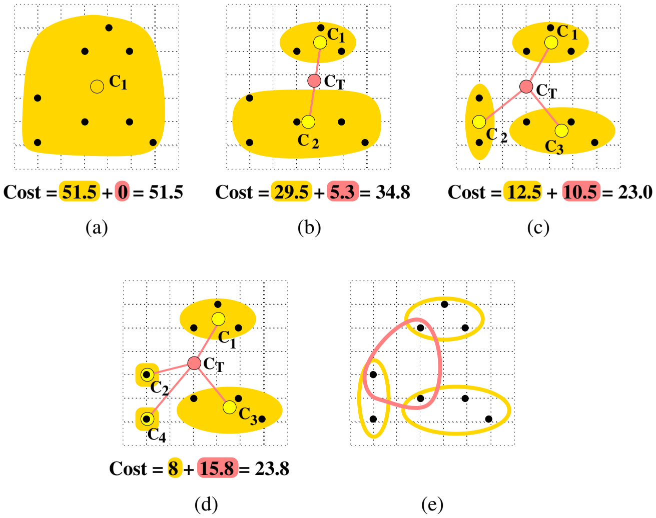


Fig. 4. Example of the clustering algorithm. (a) Initial net. Cluster pins into (b) 2 subsets, (c) 3 subsets and (d) 4 subsets. (e) Resulting set of subnets (with hyperedges) for (c) solution.

Finally, the solution with 3 subsets reported on Fig. 4(c) is selected.

C. Construction of the subnets

The subnets are constructed after the clustering to form the new netlist. Assuming that the pins of the net are split into k subsets, $k+1$ new hyperedges are created. A hyperedge is selected to interconnect the pins, because it gives total freedom to the posterior placement steps to apply any model on wirelength estimation. The subnets are built in two steps:

- The pins of each subset are connected with a hyperedge.
- For each subset, the closest pin to the center of gravity C_T is selected. The set of closest pins are interconnected with another hyperedge. It is not necessary to introduce any additional nodes like in Star net model [15].

Figure 4(e) depicts the resulting group of subnets: the three subnets derived from the solution in Fig. 4(c) and an extra hyperedge (with dark color) that binds them.

IV. Experimental Results

To test the application of our MCN model in existing placers, we have performed experiments based on the outline of Fig. 2. An intermediate step has been inserted between the two placement steps. The new netlist built after applying the MCN model is passed to the following detailed placement phase. Thus, all positive benefits compared to the common placement flow are produced because of the netlist modifications. Moreover, the contribution on the total CPU time of the MCN netlist construction is meaningless.

The experimental objective is to obtain placement layouts with shorter StWL. FastSteiner package [11] is used to measure the wirelength. We rely on the statement that improvement in StWL leads to better results on routing stage [12].

Before carrying out the main experiments, we evaluate the accuracy of the MCN model on ISPD05 benchmarks [1]. In

ISPD05	Statistics		mPL 6					FastPlace 3					Capo 10.5				
bench.	#nodes	Avr.	HPWL ($\times 10^8$)		StWL ($\times 10^8$)		CPU	HPWL ($\times 10^8$)		StWL ($\times 10^8$)		CPU	HPWL ($\times 10^8$)		StWL ($\times 10^8$)		CPU
	($\times 10^5$)	ND	GP+DP	MCN	GP+DP	MCN	Ratio	GP+DP	MCN	GP+DP	MCN	Ratio	GP+DP	MCN	GP+DP	MCN	Ratio
adaptec1	2.1	4.3	0.78	0.79	0.88	0.86	1.00	0.80	0.83	0.90	0.89	1.01	0.88	0.89	0.98	0.97	1.13
adaptec2	2.6	4.0	0.92	0.93	1.07	1.03	1.00	0.94	0.97	1.10	1.08	0.96	0.99	1.00	1.15	1.13	1.13
adaptec3	4.5	4.0	2.14	2.16	2.40	2.32	1.00	2.14	2.20	2.39	2.37	0.93	2.44	2.50	2.62	2.58	1.15
adaptec4	5.0	3.7	1.94	1.95	2.13	2.06	0.99	2.01	2.06	2.19	2.18	0.95	2.16	2.19	2.36	2.34	1.17
bigblue1	2.8	4.0	0.97	0.99	1.11	1.07	1.00	0.98	1.02	1.12	1.11	0.97	1.08	1.09	1.22	1.20	1.15
bigblue2	5.6	3.7	1.52	1.53	1.75	1.70	0.96	1.56	1.59	1.78	1.76	0.99	1.62	1.64	1.85	1.83	1.14
bigblue3	11.0	3.4	3.44	3.48	4.07	3.95	0.98	3.77	3.78	4.36	4.28	0.94	4.30	4.35	4.99	4.91	1.14
bigblue4	22.8	4.0	8.30	8.37	9.41	9.15	0.95	8.60	8.62	9.58	9.41	0.94	9.74	9.80	10.80	10.60	1.15
Norm.			**1.000**	**1.013**	**1.000**	**0.970**	**0.98**	**1.000**	**1.024**	**1.000**	**0.988**	**0.98**	**1.000**	**1.011**	**1.000**	**0.985**	**1.14**

TABLE II
MCN APPROACH ON ISPD05 CIRCUITS.

ISPD05		Wirelength error (%)			
bench.	% nets	all nets		nets >3 pins	
	> 3 pins	HPWL	MCN	HPWL	MCN
adaptec1	32	-3.73	0.45	-11.46	1.37
adaptec2	23	-2.90	0.19	-12.54	0.81
adaptec3	24	-2.62	0.25	-10.62	1.00
adaptec4	21	-2.19	0.22	-10.07	1.02
bigblue1	27	-3.36	0.32	-12.18	1.14
bigblue2	20	-2.38	0.19	-11.50	0.93
bigblue3	18	-1.94	0.12	-10.44	0.65
bigblue4	22	-2.55	0.17	-11.25	0.75
Norm.		**-2.71**	**0.24**	**-11.26**	**0.96**

TABLE I
AVERAGE ERROR IN WIRELENGTH ESTIMATION OF HPWL AND MCN MODELS IN RESPECT WITH FASTSTEINER [11] HEURISTIC.

the following series of experiments, we evaluate the MCN approach on circuits from different benchmark suites:

- *ISPD05 benchmarks [1].* We examine the performance of the MCN approach on different academic placers and report the improvement in StWL.
- *PEKU benchmarks [16].* These experiments show the efficiency of the MCN model on circuits with large number of multi-pin nets, where significant reduction in StWL is achieved.
- *ISCAS99 benchmarks.* Timing analysis is performed on the circuits mapped with different process technologies. The experiments show the improvement in wirelength and delay.

On these experiments, the traditional flow with global and detailed placement (referenced in the tables of results as GP+DP) is compared with the new flow with our intermediate step (referenced in the tables as MCN). All tables present the number of nodes and the average netdegree (Avr. ND) for all the circuits. For each circuit and placer tool, we report the HPWL, the StWL and the ratio of CPU time of the flow with the MCN approach with regard to the traditional flow. Both HPWL and StWL values are measured in dimensionless units of BookShelf format [17]. The last row shows the normalized sum of each column.

A. Accuracy of wirelength estimation

The goal of this section is to compare the accuracy of the wirelength estimation for HPWL and MCN net models with regard to FastSteiner [11] heuristic. The experiment has been performed on the ISPD05 circuits placed by mPL6 [18]. As described above, each net is firstly split into several subnets according to the MCN clustering algorithm, and the total HPWL of the subnets gives the estimation.

In Table I, the average error is computed for all nets and for nets with more than three pins separately. The table also reports the percentage of high-degree nets. The normalized sum of the errors is shown in the last row.

The HPWL underestimates the wirelength (shown with negative numbers). Our MCN model consistently produces an accurate wirelength estimation for all the circuits. Comparing with HPWL, the MCN approach increases the accuracy from -2.71% to 0.24% for all nets. The lost of precision for both net models comes from multi-pin nets that are $20-30\%$ in the netlist. This small amount of nets produces a significant increment of the inaccuracy for HPWL measure (-11.26%). On the contrary, the MCN model is able to estimate the same netlength with considerably smaller error (0.96%).

These promising results for the MCN model justify our heuristic for the optimization of the StWL in placement processes.

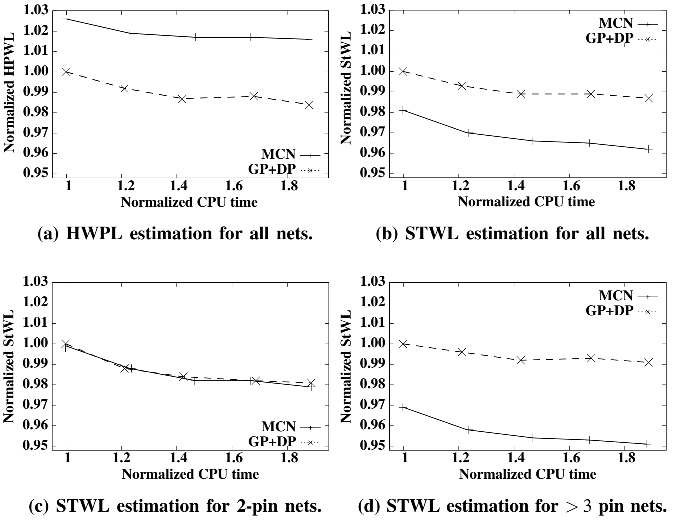
B. Experiments on ISPD05 benchmarks

The ISPD05 suite includes circuits with sizes ranging from 210 thousand to 2.1 million objects. The circuits also contain considerable number of multi-pin nets. Three placers, mPL6 [18], FastPlace 3 [19] and Capo 10.5 [12][2] have been run to demonstrate the performance of the MCN model.

Table II reports the results of the experiments. The HPWL increases for all placers and circuits, because the MCN model targets to the StWL optimization rather than improving HPWL. The last column shows the average reduction in StWL by 3.0%, 1.2% and 1.5% for mPL6, FastPlace and Capo respectively.

We have 14% of overhead in CPU time for Capo, but we spend less CPU time for the other placers. This highly depends on details of the placement algorithm, since there is always a trade-off between the increment on the number of nets and the reduction in the average net degree due to the MCN approach.

[2] We do not use ROOSTER feature, because the produced placement layouts of ISPD05 circuits are not competitive in the final wirelength. The point is that ROOSTER is aimed strictly for routability, and comparison in the wirelength is not fair(according to the personal reference to the authors).



(a) HWPL estimation for all nets.

(b) STWL estimation for all nets.

(c) STWL estimation for 2-pin nets.

(d) STWL estimation for > 3 pin nets.

Fig. 5. Iterative detailed placement on `adaptec1` circuit.

However, the runtime of constructing the MCN netlist is not of great concern and typically is less than 1% of overall placer CPU time.

Iterative Detailed Placer. One possible improvement of the scheme described in Fig. 2 consists on the idea of feedback line (not shown in the scheme). We can produce the MCN netlist and run detailed placer iteratively. The input placement for building the netlist is the detailed placement produced in the previous iteration. The proposed iterative approach is reasonable, because detailed placement contributes a small portion in overall CPU time compared to the global placement.

Figure 5 presents the results for *adaptec1* circuit during 5 runs of the detailed placer mPL6 [18]. We compare the iterative MCN approach (denoted as MCN) with an iterative scheme of the traditional flow (denoted as GP+DP). Figure 5(a) shows the expected HPWL reduction due to the iterative application of detailed placer. However, the MCN approach gives superior results in StWL reduction because of employing the MCN model as plotted in Fig. 5(b).

Figures 5(c,d) present the main feature of the MCN approach. The StWL is measured separately for low degree and multi-pin nets. The wirelength of low degree nets for both MCN and GP+DP placement flows is being reduced at the same ratio from iteration to iteration as shown on Fig. 5(c). However, the MCN approach can improve the wirelength for high degree nets significantly (Fig. 5(d)).

C. Experiments on PEKU benchmarks

In this section, the performance of the model is shown on circuits with the large number of multi-pin nets. We used the artificial PEKU benchmarks, because they contain only multi-pin nets, while the ISPD05 circuits tend to have low-pin nets.

mPL6 [18] is selected as global and detailed placer because it can work on PEKU benchmarks (FastPlace3 [19] fails on some circuits).

PEKU	Statistics		mPL 6				
bench.	#nodes	Avr.	HPWL ($\times 10^6$)		StWL ($\times 10^6$)		CPU
	($\times 10^5$)	ND	GP+DP	MCN	GP+DP	MCN	Ratio
dp01	1.3	111.7	0.91	0.92	2.96	2.62	1.00
dp05	2.9	167.5	1.95	1.96	7.41	6.59	1.01
dp10	7.0	261.7	5.62	5.52	24.77	23.76	1.04
dp15	16.3	401.5	11.99	11.77	69.92	65.23	0.98
dp18	21.2	458.3	16.19	14.43	99.48	86.71	1.00
Norm.			**1.000**	**0.974**	**1.000**	**0.908**	**1.01**

TABLE III
MCN APPROACH ON PEKU CIRCUITS.

The results are summarized in Table III. The MCN strategy leads to superior results in both HPWL and StWL. The StWL is improved by 9.2% at the expense of 1% CPU time increase.

D. Experiments on ISCAS99 benchmarks

ISCAS99 allows to track improvement not only in Steiner tree wirelength but also in delays. We selected the largest circuits and used FastPlace as a placement tool.

The $0.13\mu m$ *vxlib ALLIANCE* library [20] has been used for technology mapping. The technological parameters have been scaled down for the different technologies (65nm and 32nm), using the *Predictive Technology Model* [21]. For instance, the wire capacitance and resistance for 65nm are $2.71\Omega/\mu m$ and $0.19 fF/\mu m$, respectively, that approximately correspond to M2/M3 metal layers of the 65nm technology described in [22].

The initial circuits have been obtained by using the tree-height reduction technique `speed_up` [23] and the tree-mapping algorithm in the SIS tool [24]. A square layout with 25% whitespace has been created, with the terminals uniformly distributed around the bounding box.

Table IV summarizes the results. Besides the Steiner tree wirelength and the CPU ratio, the worst negative slack (`WNS`) and the total negative slack (`TNS`) are also reported based on the Steiner tree wirelength estimation provided by Fast-Steiner [11]. The same tendency is observed in wirelength and runtime with respect to the previous experiments. Although, the MCN model is not a delay-oriented approach, the improvement in wirelength is also reflected in delay. The reduction of delay in future semiconductor technologies (from 2.8% of improvement in 65nm to 5.6% in 32nm) corroborates the increasing relevance of interconnect optimization due to the dominant role of wire delays.

V. CONCLUSIONS

The new clustering approach for better wirelength modeling in placement has been proposed. We experimentally proved that our MCN model approximates the Steiner tree wirelength more accurately than the traditional HPWL model. Circuits with shorter wirelength and delays have been produced on the explored placement flows.

As future work, we will integrate our model in a global placer and track improvements in routability. Another line of investigation is to design a delay-aware clustering model, in order to group critical pins to the same subnet. This new model applied in placement could considerably reduce delays on critical paths.

ISCAS99	Statistics		Wirelength			65 nm				32 nm			
bench.	#nodes	Avr.	StWL ($\times 10^6$)		CPU	WNS ($\times 10^3$ ps)		TNS ($\times 10^6$ ps)		WNS ($\times 10^3$) ps		TNS ($\times 10^6$ ps)	
	($\times 10^5$)	ND	GP+DP	MCN	Ratio	GP+DP	MCN	GP+DP	MCN	GP+DP	MCN	GP+DP	MCN
b14_1	4.6	3.0	2.85	2.72	0.96	5.71	5.39	1.12	1.08	7.62	7.01	1.31	1.26
b15_1	7.3	3.2	4.81	4.69	0.98	6.85	6.69	1.93	1.89	8.33	8.03	2.36	2.27
b17_1	22.5	3.2	14.61	14.30	0.97	7.18	6.82	6.37	6.22	9.32	8.39	7.74	7.58
b20_1	8.9	3.1	5.77	5.65	0.98	8.04	8.05	2.56	2.51	10.27	9.12	2.99	2.87
b21_1	9.1	3.1	5.85	5.69	0.96	8.21	8.08	2.67	2.66	9.66	9.38	3.14	3.10
b22_1	13.4	3.1	7.41	7.25	0.97	9.53	9.60	4.35	4.27	10.59	10.61	5.03	4.94
s13207	2.7	2.8	1.49	1.38	0.99	2.88	2.74	0.44	0.43	3.01	0.95	0.50	0.49
s15850	10.7	2.9	1.95	1.88	0.97	3.94	3.85	1.20	1.22	9.58	9.39	5.50	4.93
s38584	10.0	2.9	6.00	5.94	0.99	0.99	5.01	4.17	3.63	13.96	13.57	9.24	7.36
Norm.			**1.000**	**0.969**	**0.98**	**1.000**	**0.974**	**1.000**	**0.972**	**1.000**	**0.950**	**1.000**	**0.944**

TABLE IV
MCN APPROACH ON ISCAS99 CIRCUITS.

ACKNOWLEDGEMENTS

This work was supported in part by CICYT TIN2007-66523, the UPC research scholarship and a grant from Intel Corporation.

REFERENCES

[1] G.-J. Nam, C. J. Alpert, P. Villarrubia, B. Winter, and M. Yildiz, "The ISPD2005 placement contest and benchmark suite," in *ISPD '05: Proceedings of the 2005 international symposium on Physical design*. New York, NY, USA: ACM, 2005, pp. 216–220.

[2] A. B. Kahng and S. Reda, "A tale of two nets: studies of wirelength progression in physical design," in *SLIP '06: Proceedings of the 2006 international workshop on System-level interconnect prediction*. New York, NY, USA: ACM, 2006, pp. 17–24.

[3] C. C. E., "RISA: accurate and efficient placement routability modeling," in *Proc. International Conference on Computer-Aided Design*, 1994, pp. 690–695.

[4] S. Sahni and T. Gonzalez, "P-Complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, 1976.

[5] M. Queyranne, "Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problem," *Operations Research Letters*, vol. 4, pp. 231–342, 1986.

[6] C. Li, M. Xie, C.-K. Koh, J. Cong, and P. H. Madden, "Routability-driven placement and white space allocation," in *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 394–401.

[7] A. B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," in *ISPD '04: Proceedings of the 2004 international symposium on Physical design*. New York, NY, USA: ACM Press, 2004, pp. 18–25.

[8] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," in *DATE '07: Proceedings of the conference on Design, automation and test in Europe*. San Jose, CA, USA: EDA Consortium, 2007, pp. 1226–1231.

[9] M. R. Garey and D. S. Johnson, "The rectilinear Steiner problem is NP-Complete," in *SIAM Journal of Applied Mathematics*, 1977, pp. 826–834.

[10] C. Chu, "FLUTE: fast lookup table based wirelength estimation technique," in *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 696–701.

[11] A. B. Kahng, I. I. Măndoiu, and A. Z. Zelikovsky, "Highly scalable algorithms for rectilinear and octilinear steiner trees," in *ASPDAC: Proceedings of the 2003 conference on Asia South Pacific design automation*. New York, NY, USA: ACM, 2003, pp. 827–833.

[12] J. A. Roy, J. F. Lu, and I. L. Markov, "Seeing the forest and the trees: Steiner wirelength optimization in placement," in *ISPD '06: Proceedings of the 2006 international symposium on Physical design*. New York, NY, USA: ACM Press, 2006, pp. 78–85.

[13] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability*, vol. 1. Berkeley: University of California Press, 1967, pp. 281–297.

[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.

[15] N. Viswanathan and C. C.-N. Chu, "FastPlace: efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model," in *ISPD '04: Proceedings of the 2004 international symposium on Physical design*. New York, NY, USA: ACM, 2004, pp. 26–33.

[16] J. Cong, T. Kong, J. R. Shinnerl, M. Xie, and X. Yuan, "Large-scale circuit placement: Gap and promise," in *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2003, p. 883.

[17] GSRC/bookshelf format,
vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement/plFormats.html.

[18] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, "mPL6: enhanced multilevel mixed-size placement," in *ISPD '06: Proceedings of the 2006 international symposium on Physical design*. New York, NY, USA: ACM, 2006, pp. 212–214.

[19] N. Viswanathan, M. Pan, and C. Chu, "FastPlace 3.0: A fast multilevel quadratic placement algorithm with placement congestion controlr," in *ISPD '04: Proceedings of the 2004 international symposium on Physical design*, 2007, pp. 135–140.

[20] Alliance library,
www.vlsitechnology.org/html/vx_description.html.

[21] Interconnection calculator,
www.eas.asu.edu/˜ptm/cgi-bin/interconnect/local.cgi.

[22] P. Bai, C. Auth, and et al., *A 65nm Logic Technology Featuring 35nm Gate Lengths, Enhanced Channel Strain, 8 Cu Interconnect Layers, Low-k ILD and 0.57μm^2 SRAM Cell*, Intel Developer Forum, Aug. 2005.

[23] K. Singh, A. Wang, R. Brayton, and A. Sangiovanni-Vincentelli, "Timing optimization of combinational logic," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 1988, pp. 282–285.

[24] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A system for sequential circuit synthesis," U.C. Berkeley, Tech. Rep., May 1992.