A Min-area Solution to Performance and RLC Crosstalk Driven Global Routing Problem *

Tong Jing, Ling Zhang, Jinghong Liang
Jingyu Xu, Xianlong Hong

Computer Science & Technology Department
Tsinghua University
Beijing 100084, P. R. China
Phone: +86-10-62785564
Fax: +86-10-62781489
E-mail: {jingtong, hxl-dcs}@tsinghua.edu.cn
zhangling@tsinghua.org.cn
{liangjh03, xjy99}@mails.tsinghua.edu.cn

Jinjun Xiong, Lei He

Electrical Engineering Department
UCLA
Los Angeles, California 90095-1594, USA
Phone: (310) 206-2037
Fax: (310) 206-4685
Email: jinjun@ucla.edu
lhe@ee.ucla.edu

Abstract -- This paper presents a novel global routing algorithm, AT-PO-GR, to minimize the routing area under both congestion, timing, and RLC crosstalk constraints. The proposed algorithm is consisted of three key parts: (1) timing and congestion optimization; (2) crosstalk budgeting and estimation; and (3) crosstalk elimination and local refinement. Compared with the recent work introduced in [9] and [10], the proposed algorithm can achieve smaller routing area and fewer shields under the same design constraints, yet use less running time.

I. INTRODUCTION

Global routing plays an important role in very/ultra large scale integrated circuit (VLSI/ULSI) physical design. New challenges to global routing are coupling noise (crosstalk) elimination and performance optimizations [1]. There are some works focusing on the above problems, which mainly fall into three categories, noise modeling [2-3], noise minimization [4-7], and simultaneous noise minimization and performance optimization [8-10].

Among noise minimization algorithms, post global routing optimization techniques have been studied in literature. For example, [4] described a two-pass algorithm that includes region-based crosstalk risk estimation and crosstalk reduction. [5] proposed a three-phase algorithm based on crosstalk budgeting, simultaneous shield insertion and net ordering (SINO), and local refinement.. As routing solution has been decided, there are limited design freedoms to leverage in order to reduce crosstalk. Therefore, it makes sense to consider crosstalk reduction early in the global routing phase. An early work on this is due to [6], in which a cost function that took crosstalk into consideration is used during the phase of constructing the routing Steiner tree. If the crosstalk of initial routing solution exceeds the given bound after routing, rip-up and reroute will be used to improve the solution.

The practical applications need simultaneous performance optimization (timing performance and routability) and crosstalk elimination. Ref. [8] and [9] proposed performance optimization global routing algorithms considering crosstalk reduction. The former mainly focuses on coupling capacitance and uses spacing method. The later considers coupling inductance and is based on shield insertion. The shortcoming of [9] is that the running time is long due to the simulated annealing (SA) method. An efficient RLC crosstalk reduction algorithm is presented in [10], which is much faster than [9] with the similar routing results. However, the routing area and shield number in [10] are comparably larger than those in [9].

The main contribution of this paper is a min-area solution to performance and RLC crosstalk driven global routing problem. The algorithm performs much faster compared with [9], and obtains routing solution with less routing area and fewer shields compared with [10].

The remainder of this paper is organized as follows. Section 2 gives necessary preliminaries. Section 3 introduces problem formulations. In Section 4, we discuss AT-PO-GR, our global routing algorithm in detail. Section 5 shows experimental results. Section 6 concludes and gives some possibilities for future work.

II. PRELIMINARIES

A. *Global Routing Problem*

With the progress in multi-layer routing technology, routing area is a whole chip plane. Thus, a net can be specified as a set of nodes in global routing graph (GRG). Then, the problem of routing a net can be described as a rectilinear Steiner minimal tree (RSMT) problem of specified nodes in GRG [11].

* This work was supported in part by the NSFC under Grant (China) No.60373012, the NSF CAREER Award (USA) CCR-0401682, the SRFDP of China under Grant No.20020003008, and the Hi-Tech Research and Development (863) Program of China under Grant No.2004AA1Z1050.

Fig.1 shows an example GRG that holds 4×4 GRCs. Node i represents the center point of GRC_i. The edge linking vertex i (v_i) and vertex j (v_j) is named as e, l is the length of edge e, equals the distance between vertex i and vertex j. A non-negative number c_e, called edge capacity, indicates the number of available tracks between two adjacent vertices of edge e.

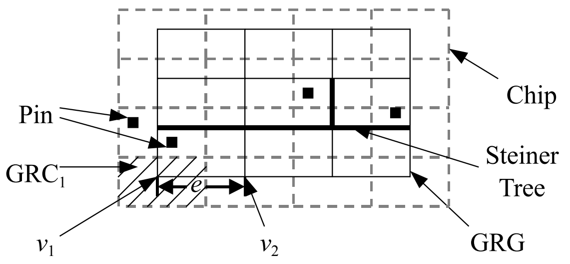

Fig. 1. Global routing graph (GRG).

B. RLC Noise Model

The *LSK* model for RLC crosstalk [3, 7] is used in this paper. Different from earlier noise model [2], the *LSK* model considers coupling inductance between adjacent and non-adjacent sensitive nets. For any two segments N_{it} and N_{jt} in region R_t, the inductive coupling coefficient between them is

$$k_{it,jt} = \frac{L_{it,jt}}{\sqrt{L_{it} \cdot L_{jt}}} \quad (1)$$

where $L_{it,jt}$ is the mutual inductance between N_{it} and N_{jt}, and L_{it} and L_{jt} are the self inductance for N_{it} and N_{jt}, respectively. A formula-based K_{eff} model has been developed in [3] to calculate the coupling coefficient $k_{it,jt}$. Furthermore, the total amount of inductive coupling induced on N_{it} can be represented by the sum of the inductive coupling coefficient $K_{it} = \sum_{j \neq i} k_{it,jt}$ for all net segments N_{jt} that are sensitive to N_{it}.

To consider the effect of interconnect length and the general case where the total coupling is not uniform in all routing regions, a length-scaled K_{eff} (*LSK*) model was proposed in [7], where the *LSK* value is defined as

$$LSK = \sum_t l_t \cdot K_{it} \quad (2)$$

where l_t is length of R_t and K_{it} is total coupling for N_{it} in region t.

C. Tabu Search

Tabu search has been widely used to cope with the overwhelming computational intractability of NP-hard combinatorial optimization problems since firstly proposed by Glover in 1986 [13], which is applied to crosstalk elimination in this paper.

The basic idea of this technology is simple, which records and taboos the local minimum points that has been reached so as to avoid getting stuck at these points and finds out new search ways that could lead to the global minimum point eventually. The outline of Tabu search algorithm can be described in Fig.2.

Step1. Select an initial solution x^{now}, and set Tabu list H=empty;
Step2. While not meet the stop conditions do
Generate a candidate list Can_N(x^{now}) from the neighborhood N(x^{now},H) of x^{now} that doesn't conflict with H;
Select the best solution from Can_N(x^{now}):x^{next};
x^{now}=x^{next};
Update Tabu list H;
End While

Fig.2. Outline of Tabu search algorithm.

Key factors of Tabu search are neighborhood, Tabu object, Tabu length and aspiration rule. The following are some concerns in applying Tabu search method. (i) How to choose proper Tabu object and Tabu length. (ii) How to search efficiently in neighborhood. (iii) How to set the reasonable aspiration rule.

III. PROBLEM FORMULATION

Let

$$Sit = \begin{cases} 1 & \text{edge } t \text{ is used by net } i, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where S_{it} is a kind of stamps indicating whether edge t contains net i.

Then we have

$$\textbf{Minimize} \quad L = \sum_{i=1}^{N_n} \sum_{t=1}^{N_e} S_{it} l_t$$

Subject to

$$f_t = \sum_{i=1}^{N_n} S_{it} + sn_t + o_t \le c_t, \forall t \in E; \quad (4)$$

$$T(i,j) \le T_D(i,j), \forall i \in N, \forall j \in s(i); \quad (5)$$

$$LSK_{ij} \le \overline{LSK_{ij}}, \forall i \in N, \forall j \in s(i); \quad (6)$$

Formula (4) is the congestion constraint, which forbids the overflow on each GRG edge. Formula (5) guarantees the actual delay value from source i to sink j, $T(i, j)$, is no more than the given timing constraint $T_D(i, j)$. Formula (6) sets the upper bound of *LSK*, $\overline{LSK_{ij}}$, for each source sink pair ij. The actual *LSK* value of this pair, LSK_{ij}, could not exceed the bound.

IV. OUR GLOBAL ROUTING ALGORITHM AT-PO-GR

To obtain routing solution with less routing area and fewer shields, yet less running time, we designed the new flow for AT-PO-GR instead of partial improvements from our previous work [9], [10].

A. The Main Flow of AT-PO-GR

The main flow chart of AT-PO-GR is shown in Fig. 3 and the corresponding pseudo code is in Fig. 4, which are different from those in [9] and [10].

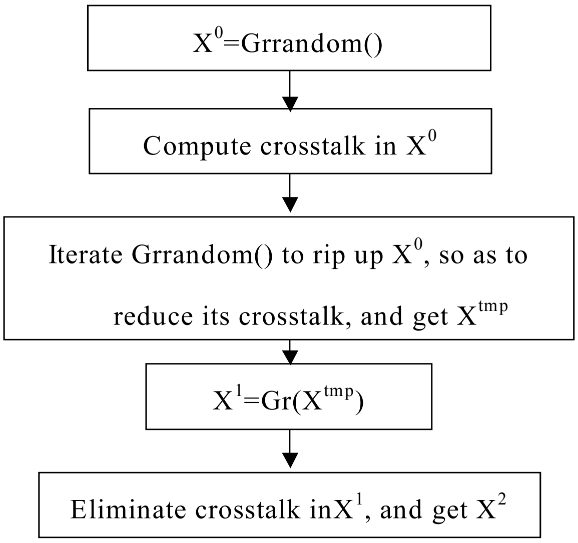

Fig. 3. The main flow chart of AT-PO-GR.

Step1. X^0=Grrandom( );
Step2. CtkEst(X^0);
Step3. n=0;tmp=vionum(X^0);
while(n<nBound) *do*
Step4. X^{n+1}=Grrandom(X^n);
Step5. CtkEst(X^n);
Step6. *If* vionum(X^{n+1})<tmp
Step7. *Then* tmp=vionum(X^{n+1});n=0; X^{tmp}=X^{n+1};
Step8. *else* n++;
Step9. X^1=Gr(X^{tmp});
Step10. X^2=CtkEli(X^1);

Fig. 4. Pseudo code of AT-PO-GR.

AT-PO-GR mainly consists of the following 3 parts.

(1) Gr() and Grrandom(): timing optimization and congestion reduction;

(2) CtkEst(): crosstalk budgeting and estimation;

(3) CtkEli(): crosstalk elimination and local refinement.

AT-PO-GR firstly uses Grrandom() to generate an initial routing solution X^0. Then, CtkEst() budgets and computes its crosstalk. After that, we apply an iteration procedure to reduce the crosstalk in X^0 and obtain X^{tmp}. We consider X^{tmp} as a good mid-solution and call Gr() to reduce its wire length, congestion, and delay. Then, we get X^1. Finally, CtkEli() eliminates crosstalk in X^1 and gets the final result X^2.

B. Part 1: Gr() and Grrandom()

Since Gr() and Grrandom() are used in different situations, they were designed in different ways. The complete congestion reduction and timing optimization are performed in Gr(). Gr() was designed following 3 different strategies, stochastic optimization, deterministic optimization, and local enumeration optimization strategy [12], which represents 3 different search directions in global routing solution space so as to transit the local minimum point and make a fast search. That is, the hybrid optimization method can dynamically reconstruct the problem structure and make "transition" from a local minimum point (see Fig. 5).

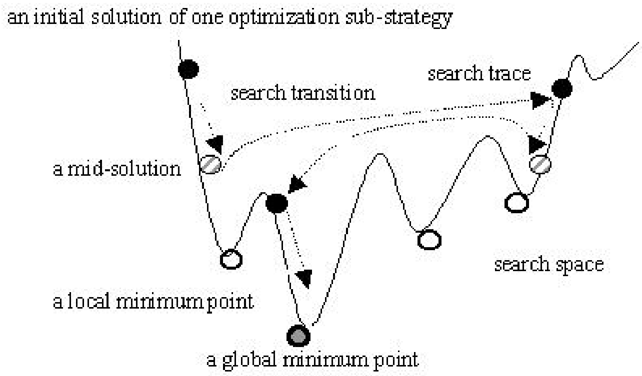

Fig. 5. Transittion from a local minimum point.

Stochastic optimization strategy randomly selects a subset from current congested net set in each iteration, and simultaneously reroutes them to reduce congestion. It is a fast tentative optimization method. Deterministic optimization strategy sequentially rips-up and reroutes all congested net with a random order to reduce congestion in current solution. Local enumeration optimization strategy selects the best Steiner tree for each congested net, so congestion has been minimized after applying this optimization strategy.

Grrandom() focuses on tentatively finding a good mid-solution X^{tmp} with comparatively lower crosstalk violation. Then, Grrandom() only uses stochastic optimization strategy.

In Gr() and Grrandom(), we use the following new cost formulas of GRG edge that can take crosstalk into account, which are different from those in [9] and [10].

$$\overline{w_t} = \frac{f_t + \delta}{c_t + \delta} + \frac{nv_t}{c_t + \delta} \tag{7}$$

$$w_t = \begin{cases} \overline{w_t}, (f_t \le c_t) \\ K \cdot \overline{w_t}, otherwise \end{cases} \tag{8}$$

where c_t is the capacity of edge t, f_t is the number of used tracks in edge t, δ is a small real number that validates formula (7) while c_t is 0, $\overline{w_t}$ is the actual congestion of edge t, K is a large integer used as the penalty factor, w_t is the weighted cost of edge t, and nv_t is the number of net segments in edge t that violate crosstalk constraint.

Considering possible shield may be inserted due to these net segments, we add nv_t in $\overline{w_t}$ such that these edges tend to become more congested. Then, nets crossing such edges will have higher cost and thus it will be avoided.

The timing optimization follows the idea of critical network concept introduced in [11].

C. Part 2: CtkEst()

In this part, it firstly partitions the *LSK* bound for each sink of a net into the GRG edges that belong to the source-sink paths. Let $\overline{LSK_{ij}}$ be the crosstalk bound at sink p_{ij} for net N_i (given by designers, see the *Benchmark Data* in Section V.*A*.), *len* be the total length from the source p_{io} to sink p_{ij}. We then can get a uniform crosstalk sub-bound $\overline{K_{it}}$ for net N_i at each routing region (i.e., one GRG edge) R_t as follows.

$$\overline{K_{it}} = \frac{\overline{LSK_{ij}}}{len} \quad (9)$$

Secondly, having got $\overline{K_{it}}$, CtkEst() computes actual K_{it} with *LSK* model.

At last, we can obtain K_{slack} for each source-sink pair *ij*. K_{slack} has the following definition.

$$K_{slack} = \sum_t (\overline{K_{it}} - K_{it}) \quad (10)$$

In Fig.4, procedure vionum() means to compute the number of source-sink pair whose K_{slack} is less than 0 in a solution. So it measures how serious crosstalk is in a solution.

D. Part 3: CtkEli()

To eliminate crosstalk, this part applies a 3-step optimization method: (i) firstly, insert shields in each GRG region so that the crosstalk of most regions is within the given bound, (ii) secondly, insert shield in those regions which have possible remnant crosstalk, so that crosstalk is completely eliminated, (iii) finally, delete unnecessary shields so that the final area is minimized.

CtkEli() uses Tabu search method to do crosstalk elimination throughout all the 3 steps but [10] only uses Tabu search in its first 2 steps. The 3rd step based on SA method is time consuming. So, AT-PO-GR performs much faster than [10] does to accomplish crosstalk elimination and local refinement by using CtkEli().

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

The global router AT-PO-GR has been implemented in C language. It performs on a SUN V880 workstation with Unix OS. We compared our results with PO-GR [9] and T-PO-GR [10].

A. Benchmark Data

We tested four MCNC benchmarks under 0.2*um* technology, which are C2, C5, C7, and avq. Sensitivity rate of 0.5 is given to all nets and a random sensitivity matrix is created. *LSK* bound at each sink, $\overline{LSK_{ij}}$, is set to be 1000.

TABLE I summarizes the benchmark data sets. PO-GR and T-PO-GR tested three MCNC benchmarks, which are C2, C5, and C7.

TABLE I
BENCHMARK DATA

Circuit	Number of nets	Grids
C2	745	9×11
C5	1764	16×18
C7	2356	16×8
avq	21851	65×67

B. Results and Discussions

The experimental results are shown in TABLE II and TABLE III, respectively.

(1) From the second, third, and forth row in TABLE II, we see that the iteration procedure can reduce crosstalk violation number by about 2% to 7%. After the iteration, Gr() tries to minimize the total wire length and considers timing constraint, so the violation number will rise up a little in X^1.

(2) TABLE II also shows that using Tabu search method greatly shortens the runtime of AT-PO-GR, which is no more than 5% of PO-GR. That is to say, the speed of AT-PO-GR is at least 20 times of PO-GR. For the larger scale circuit avq, PO-GR did not give the runtime.

(3) AT-PO-GR can reduce wire length by more than 4% compared with PO-GR as shown in TABLE II. It's reasonable since AT-PO-GR contains crosstalk factors in edge cost. It can adjust the topology of net if there are crosstalk violations.

For example, if net *i* has crosstalk violation passing edge *t*, it can change the solution to net *j* passing edge *t* without crosstalk violation. While in PO-GR, once a track in edge *t* is used by shield, maybe neither net *i* nor net *j* can pass that edge, but to find a topology with longer wire length.

(4) AT-PO-GR has smaller routing area compared with PO-GR as shown in TABLE II.

(5) TABLE III shows that the running time of AT-PO-GR is about half of the runtime of T-PO-GR. That is, CtkEli() uses Tabu search method to do crosstalk elimination throughout all the 3 steps but [10] only uses Tabu search in its first 2 steps. And the adjustment made by Gr() is helpful for succeeding CtkEli() step.

TABLE II
THE COMPARISON OF WIRE LENGTH, RUNNING TIME, AND ROUTING AREA BETWEEN AT-PO-GR AND PO-GR [9]

Circuit		C2	C5	C7	avq
AT-PO-GR	Vionum (X^0)	654	1600	1960	9885
	vionum (X^{tmp})	608	1485	1902	9690
	Decrease	5.66%	7.19%	2.96%	1.97%
	vionum (X^1)	622	1522	1902	9690
	Wire length (X^0)	477516	1415238	1588218	10154788
	Wire length (X^1)	450730	1266044	1530654	9906136
	Running time (s)	84.33	245.55	336.94	6277.5
	Area	150×187	269×304	337×378	1206×986
PO-GR [9]	Wire length	471840	1327942	1606928	-
	Running time (s)	2457.39	5738.45	9985.52	-
	Area	160×190	269×309	364×366	-
AT-PO-GR running time / PO-GR running time		3.43%	4.28%	3.37%	-
The decrease of AT-PO-GR wire length compared with PO-GR wire length		4.47%	4.66%	4.75%	-
The decrease of AT-PO-GR area compared with PO-GR area		7.73%	1.62%	4.38%	-

The symbol “-” in the table means not available.

TABLE III
THE COMPARISON OF WIRE LENGTH, RUNNING TIME, AND ROUTING AREA BETWEEN AT-PO-GR AND T-PO-GR [10]

Circuit		C2	C5	C7	avq
AT-PO-GR	Running time (s)	84.33	245.55	336.94	6277.5
	Area	150×187	269×304	337×378	1206×986
	Shield number	166	484	665	4131
	Wire length	450730	1266044	1530654	9906136
T-PO-GR [10]	Running time (s)	169.07	417.93	630.71	-
	Area	169×211	284×330	337×405	-
	Shield number	204	527	684	-
	Wire length	460384	1308622	1616152	-
The decrease of AT-PO-GR running time compared with T-PO-GR running time		50.12%	42.25%	46.58%	-
The decrease of AT-PO-GR wire length compared with T-PO-GR wire length		2.10%	3.25%	5.29%	-
The decrease of AT-PO-GR area compared with T-PO-GR area		21.34%	12.74%	6.67%	-
The decrease of AT-PO-GR shield number compared with T-PO-GR shield number		18.63%	8.16%	2.78%	-

(6) Compared with T-PO-GR, the wire length of AT-PO-GR is also shorter with the same reason of item (3) as shown in TABLE III.

(7) AT-PO-GR has smaller routing area and fewer shields compared with T-PO-GR as shown in TABLE III. It shows that the iteration procedure in AT-PO-GR is efficient in minimizing routing area and reducing shield number.

VI. CONCLUSIONS AND FUTURE WORK

A min-area solution to performance and RLC crosstalk driven global routing problem has been presented in this paper. The experimental results have shown that this algorithm is able to: (i) obtain routing solutions with less routing area compared with [9] and [10], and (ii) preserve the good routing result and greatly decrease the running time compared with [9] and [10].

As our future work, we plan to find more specific methods to construct the Steiner tree set for crosstalk minimization, and better strategies for crosstalk budgeting.

VII. ACKNOWLEDGEMENTS

This paper describes research work performed cooperatively at Tsinghua University, Beijing, P. R. China and University of California, Los Angeles (UCLA), USA. The authors wish to thank Zuying Luo in University of Toronto, Canada for valuable discussions. Thanks also go to the anonymous reviewers for their helpful comments.

REFERENCES

[1] T. Jing, X. L. Hong. "The Key Technologies of Performance Optimization for Nanometer Routing". In: Proc. IEEE ASICON, Beijing, China, 2003, pp.118-123.

[2] T. Sakurai, S. Kobayashi, and M. Node. "Simple expressions for interconnecting delay, coupling and crosstalk in VLSI's", In: Proc. IEEE ISCAS, Singapore, 1991, pp.2375-2378.

[3] L. He and K. M. Lepak. "Simultaneous shield insertion and net ordering for capacitive and inductive coupling minimization", In: Proc. ACM ISPD, San Diego, CA, USA, 2000, pp.56-61.

[4] T. X. Xue, E. S. Kuh, and D. S. Wang. "Post global routing crosstalk synthesis", IEEE Trans on CAD, 1997, 16(12): pp.1418-1430.

[5] J. J. Xiong and L. He. "Full-Chip Routing Optimization With RLC Crosstalk Budgeting", IEEE Trans on CAD, 2004, 23(3) pp. 366-377.

[6] H. Zhou and D. F. Wong. "Global routing with crosstalk constraints", IEEE Trans on CAD, 1999, 18(11): pp.1683-1688.

[7] J. Ma and L. He. "Towards Global routing with RLC crosstalk constraints", in: Proc ACM/IEEE DAC, New Orleans, Louisiana, USA, 2002, pp.669-672.

[8] J. Y. Xu, X. L. Hong, T. Jing, L. Zhang, J. Gu. "A Coupling and Crosstalk Considered Timing-Driven Global Routing Algorithm for High Performance Circuit Design", In: Proc. IEEE/ACM ASP-DAC, 2004, Yokohama, Japan， pp.677-682.

[9] L. Zhang, T. Jing, X. L. Hong, J. Y. Xu, J. J. Xiong, L. He. "Performance Optimization Global Routing with RLC Crosstalk Constraints", In: Proc. IEEE ASICON, Beijing, China, 2003, pp.191-194.

[10] L. Zhang, T. Jing, X. L. Hong, J. Y. Xu, J. J. Xiong, L. He. "Performance and RLC Crosstalk Driven Global Routing", In: Proc. IEEE ISCAS, 2004, Vancouver, Canada, pp.V65-68.

[11] T. Jing, X. L. Hong, H. Y. Bao, *et al*, "A novel and efficient timing-driven global router for standard cell layout design based on critical network concept", In: Proc. IEEE ISCAS, Scottsdale, Arizona, USA, 2002, pp.I165-168.

[12] T. Jing, X. L. Hong, H. Y. Bao, *et al*, "SSTT: Efficient Local Search for GSI Global Routing", J. Comput. Sci. & Technol., 2003, 18(5): pp.632-639.

[13] F. Glover, "Future paths for integer programming and links to artificial intelligence", Computers and Operations research, 1986, 13(5): pp.533-549.