PERFORMANCE AND RLC CROSSTALK DRIVEN GLOBAL ROUTING*

Ling Zhang[1], Tong Jing[1], Xianlong Hong[1], Jingyu Xu[1], Jinjun Xiong[2], Lei He[2]

[1] Dept. of CST, Tsinghua Univ., Beijing 100084, P. R. China
[2] Dept. of EE, UCLA, Los Angeles, CA 90095-1594, USA
E-mail: {zhangling, jingtong}@tsinghua.org.cn; jinjun@ucla.edu; lhe@ee.ucla.edu

ABSTRACT

This paper presents a global routing algorithm that minimizes total wire length and satisfies RLC crosstalk constraints specified at sinks. Our algorithm is based on critical network concept and search space traversing technology (SSTT) for global routing synthesis and Tabu search for shield insertion and net ordering (SINO) to eliminate noise. The algorithm achieves about 20x speedup compared with a recent work using iterative deletion based global routing and simulated annealing based SINO. Furthermore, our algorithm increases the wire length by 4% compared with global routing without crosstalk constraints, achieving a 2.5x reduction compared with the aforementioned recent work.

1. INTRODUCTION

With the progress of VDSM technology and giga-hertz clock frequencies, performance optimization becomes an increasingly dominant factor in global routing [1]. One of the major concerns is coupling noise elimination. There are some works focusing on noise reduction, which mainly fall into two categories, noise modeling [2-3] and noise minimization [4-7]. Among noise minimization algorithms, some post optimizations are performed after global routing. Ref. [4] described a two-part algorithm of region-based crosstalk risk estimation and crosstalk reduction. In [5], the iSINO algorithm is proposed, which eliminates crosstalk by inserting shields.

Researchers find that it is more flexible if they reduce noise in the global routing phase. In [6], it constructs Steiner tree with a cost function including crosstalk consideration. If the crosstalk of initial routing solution still exceeds the given bound, then do rip up. Ref. [7] proposed the GSINO Algorithm. Since the simulated annealing (SA) method is used, it takes long running time. Meanwhile, the objective of these two algorithms is to minimize crosstalk. They do not take timing performance and routability into consideration.

This paper studies RLC coupling noise elimination problem in the process of global routing. The main contribution of this paper is that an efficient crosstalk elimination algorithm based on Tabu search is proposed. Moreover, timing performance and routability are simultaneously considered at global routing level. That is, it regards wire length as the objective and considers timing, RLC coupling noise, and routability as the constraints. Then, the performance optimization is performed throughout the global routing phase under multi-constraints.

The remainder of this paper is organized as follows. Section 2 gives necessary preliminaries. In Section 3, the coupling noise elimination algorithm based on Tabu search is described in detail. In Section 4, we discuss global routing with performance optimization. Section 5 shows experimental results. Section 6 concludes and gives some possibilities for future work.

* This work was partially supported by the SRFDP 20020003008, the NSFC (China) 60176016, the NSF CAREER Award (USA) CCR-0093273, Hi-Tech Research and Development (863) Program of China 2002AA1Z1460, and the Key Faculty Support Program of Tsinghua Univ. [2002] 4.

2. PRELIMINARIES

2.1. 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 tree (RST) problem of specified nodes in GRG [9].

2.2. 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

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

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 coefficients $k_{it,jt}$. Furthermore, the total amount of inductive coupling induced on N_{it} can be represented by the sum of the inductive coupling coefficients $K_{it} = \sum_{j \neq i} k_{it,jt}$ 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

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

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

2.3. 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 [8]. 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.1.

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.1. 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. (1) How to choose proper Tabu object and Tabu length. (2) How to search efficiently in neighborhood. (3) How to set the reasonable aspiration rule.

3. NOISE ESTIMATION AND ELIMINATION

3.1. The Three-Step Method

The flow chart of crosstalk estimation and elimination is shown in Fig.2. There are three main steps described as follows.

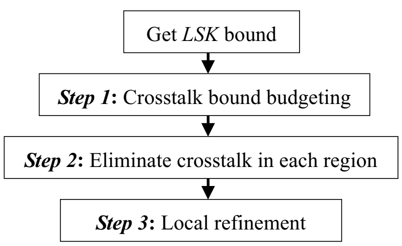

Fig.2. The three-step method.

Step 1**:** *LSK* bound budgeting

We partition the *LSK* bound at each sink of a net into the GRG edges belonging to the source-sink paths by using CBUD (uniform distributed crosstalk budgeting) strategy. This strategy uniformly partitions the *LSK* bound into edges according to their length. Let $\overline{LSK_{ij}}$ be the crosstalk bound at sink p_{ij} for net N_i, *len* be the total length from the source p_{io} to sink p_{ij}, each routing region (one GRG edge) R_t on the path is then assigned a uniform crosstalk sub-bound $\overline{K_{it}}$:

$$\overline{K_{it}} = \frac{\overline{LSK_{ij}}}{len} \qquad (3)$$

If the segment N_{it} is shared with multiple paths starting from the same source to different sinks, we use the minimum value computed for these paths according to equation (3). If we focus on the same region and compare the K_{eff} with the sub-bound of one net $\overline{K_{it}}$, $\overline{K_{it}}$ will be expressed as K_{th}.

Step 2**:** Crosstalk elimination in each region

According to each $\overline{K_{it}}$ computed in ***Step 1***, this step applies optimization method in each region to insert shields, so that the crosstalk of all regions is within the given bound.

Ref. [3] and [7] introduced the strategy of using SA method to insert shields in each region. SA method could obtain good results. But it takes comparative long runtime. In the following, we will introduce the strategy of using Tabu search method for this step, which obtains similar performance as SA while greatly shortening runtime (see Section 5).

Step 3**:** Local refinement

Check each net to eliminate possible remnant crosstalk and delete unnecessary shields so that the final area is minimized.

First, to eliminate remnant crosstalk, the net N_{it} with most critical crosstalk violation is chosen, and shields will be inserted in the least congested region R_t on N_i's path.

Second, to reduce total area, the most congested region R_t is chosen, and the slack K_{it} - K_{th} of all nets in R_t is computed. If possible, shields could be deleted when K_{th} increases properly.

3.2. Crosstalk Elimination Based on Tabu Search

We use Tabu search in ***Step 2*** to reduce runtime. The method, described in Fig.3, is much faster than SA but can obtain similar performance.

Set the global solution in one GRG edge as initial solution x^{cur}*;*
Set Tabu list H=empty; a=0; c=0;
While($a < N_a$*)*
tmpcost = ∞ *;*
b = 0;
While ($b < N_b$*)*
$x^{new} = x^{cur}$*;*
randommove (x^{new} *);*
If cost (x^{new}*) is in H*
c++;
If $c < N_c$*, then continue;*
Else c = 0;
If cost (x^{new}*) < tmpcost, then*
$x^{tmp} = x^{new}$*;*
tmpcost = cost (x^{new}*);*
b++;
Insert x^{cur} *into H;*
$x^{cur} = x^{tmp}$*;*
If cost (x^{cur}*) < cost (*x^{min}*), then* $x^{min} = x^{cur}$*; a = 0;*
Else a++;
Update H;

Fig.3. Tabu search in ***Step 2***.

In the crosstalk elimination problem, a solution is a sequence of net ordering in a certain GRG edge. There are often hundreds of GRG edges in middle-scale circuits. Meanwhile, Tabu search algorithm is used for each GRG edge. So, the runtime of Tabu search greatly affects the efficiency of the whole routing algorithm.

In Fig.3, we can see that Tabu search finds a best legal candidate in a candidate set of x^{cur}'s neighborhood, taboos the current solution x^{cur}, and accepts that best legal candidate as new x^{cur}. It records the best solution x^{min} throughout the whole search process. Following such a flow, this method has the ability to traverse from the local minimal solution in the search space and can record the best solution it has ever reached. For convenience, we use x^{new} as a copy of x^{cur} at the beginning of iteration one time and do random movement on x^{new}. We record the best neighbor solution of x^{cur} in x^{tmp}.

We use the cost of each solution as the Tabu object because it is convenient to taboo a set of solutions having the same cost. The cost function includes the following four factors: (1) Total number of nets that are adjacent to their sensitive nets; (2) Total number of shields in a GRG edge; (3) Summation of (K_{eff} - K_{th}) for all nets with $K_{eff} > K_{th}$ in a GRG edge; (4) Total number of nets with $K_{eff} > K_{th}$ in a GRG edge.

We use four kinds of random movements to find a neighbor solution: swap two net randomly, move one net randomly, insert one shield randomly and remove one shield randomly. Each of these random movements has restrictions so that a neighbor solution is still a feasible solution (that is, to exclude the cases such as two shields are next to each other) and each movements has some certain possibility to be conducted with the control of different weights on them.

Four parameters, N_a, N_b, N_c, and Tabu length, could affect the running time and performance of Tabu search method. N_a is the total iteration times if Tabu search couldn't find a new best solution. N_b is the total number of neighbor solutions that Tabu search method regards them as legal candidates. N_c is the times that this method will try to find one legal candidates. N_c is also a kind of aspiration criterion for that if there are no more legal neighbor solution after N_c times search, this method will accept the last solution as a legal candidates, even if it has been tabooed actually. Tabu length is the times that one cost value is labeled as illegal.

Some of the parameters should fit for the scale of search space, which means for a larger search space, we need larger N_a , N_b , and Tabu length to obtain better results, and for a smaller search space, we need smaller N_a , N_b , and Tabu length to shorten the running time. In our problem formulation, the scale of search space directly depends on the number of nets in one GRG edge. When the number of nets is from 20 to 50, Tabu length and N_c do not have great effects on the final results. But all these parameters have effects on the runtime.

Based on large number of experiments, we find that the proper value of these parameters are as follows: N_a=350, N_b=20, N_c=10, Tabu length=3.

4. GLOBAL ROUTING WITH PERFORMANCE OPTIMIZATION

Besides the above RLC crosstalk elimination, we also perform performance optimization in global routing phase, which includes timing performance, routability, and coupling noise. It has been implemented as the performance and RLC crosstalk driven global router, called PO-GR, which consists of the following two parts.

(1) ***Part 1*:** timing performance and routability

(2) ***Part 2*:** crosstalk estimation and elimination

In ***Part 1***, it firstly generates an initial routing solution considering congestion and timing optimization. The timing analysis and optimization method follows the critical network concept introduced in [9] and the congestion reduction uses the search space traversing technology (SSTT) introduced in [10]. Then, ***Part 2*** eliminates the crosstalk from the solution by inserting shields and gets a mid-result. Finally, regard the mid-result as input and send it to ***Part 1*** for iterations. The flow chart and pseudo code of PO-GR are shown in Fig.4 and Fig.5, respectively.

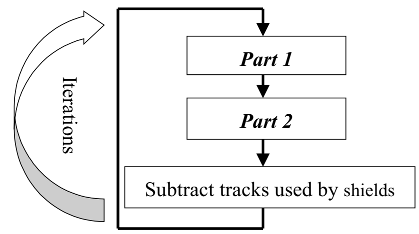

Fig.4. The flow chart of PO-GR.

*1. Call **Part 1** to generate a minimum wire length initial solution X^0 without congestion and timing violation;*
*2. Call **Part 2** to obtain X^1 = CEE(X^0);*
3. If(no edge overflow in X^1) do go to step 4.;
Else do go back to step 1. to generate a new solution;
*4. Call **Part 1** again to obtain congestion and timing optimized solution X^2 from X^1;*

Fig.5. The pseudo code of PO-GR.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The PO-GR is implemented in the C language. It runs on a SUN Enterprise 450 workstation with Unix OS. There are two running modes. *T* mode is timing-driven mode. *W* mode is non-timing-driven mode. We only reduce congestion in *W* mode, and consider both congestion and timing performance in *T* mode. We compared our crosstalk elimination algorithm with the recent work [3, 7] based on SA algorithm.

5.1. Benchmark Data

We tested three MCNC benchmarks under 0.2*um* technology. Table 1 summarizes the benchmark data sets. Sensitivity rate of 0.5 is given to all nets and a random sensitivity matrix is created. *LSK* bound at each sink is set to be 1000. N_a=350, N_b=20, N_c=10, and Tabu length=3.

Table 1 Benchmark data

Circuits	Number of nets	Grids
C2	745	9×11
C5	1764	16×18
C7	2356	16×18

5.2. Results

The experimental results are shown in Table 3, Table 4, and Table 5, respectively. Table 2 shows the needed notations.

5.3. Discussions

(1) From row 4 in Table 3, we can see that Tabu search sharply decreases runtime in XSINO step (about 20x speedup) compared with simulated annealing method [3, 7]. From row 10 in Table 3, we also can see that the total runtime is greatly shortened. The local refinement step has not been changed, but its runtime has been decreased slightly (row 7 in Table 3), which means Tabu search doesn't make any bad effects on succeeding optimization.

(2) From Table 4, we can find that Tabu search obtains similar results in routing area compared with simulated annealing method [3, 7]. The shielding number only increases a little.
(3) Row 4 in Table 5 shows that the wire length increment of PO-GR is no more than 4.65%. So, crosstalk estimation and elimination does slight bad effects on wire length. Furthermore, the wire length increment is about 10% in [7]. Therefore, we achieve 2.5x wire length reduction.
(4) The minimum delay slack (i.e., required delay – current delay), denoted as Min-R and shown in row 9 and row 10 in Table 5, is almost unaffected. So, PO-GR keeps the effectiveness in timing optimization in ***Part 1*** (i.e., P1 in Table 5).

Table 2 Needed notations

SA	The existing crosstalk elimination algorithm based on simulated annealing in [3, 7]
Tabu search	Our crosstalk elimination algorithm based on Tabu search
XSINO	***Step 2*****:** Crosstalk elimination in each region
LR	***Step 3*****:** Local refinement
TT	Total runtime (XSINO + LR)
Sn	Shield number
PO-GR	Our two-step router with performance optimization
P1	***Part 1*****:** timing and congestion optimization
Min-R	Minimum delay slack (required delay–current delay)

Table 3 Comparison of runtime (*s*) between Tabu search and SA

Circuits		C2	C5	C7
XSINO	SA	901.97	2140.36	3748.78
	Tabu search	45.75	112.87	237.80
	Runtime reduction	856.22	2027.49	3510.98
LR	SA	153.53	56.36	453.70
	Tabu search	91.44	34.08	227.50
	Runtime reduction	62.09	22.28	226.20
TT (XSINO + LR)	SA	1055.50	2196.72	4202.48
	Tabu search	137.19	146.95	465.30
	Total runtime reduction	918.31	2049.77	3737.18

Table 4 Comparison of results between Tabu search and SA

Circuits		C2	C5	C7
Area	SA	149×196	271×301	342×395
	Tabu search	149×202	273×307	346×393
Sn	SA	158	460	589
	Tabu search	165	501	621
	Sn increment	7	41	32

6. CONCLUSIONS AND FUTURE WORK

A performance and RLC crosstalk driven global routing algorithm is presented. The experimental results show that this algorithm is able to: (1) Preserve the good routing result and greatly decrease the running time. (2) Tackle coupling noise, timing performance and routability simultaneously. It efficiently eliminates crosstalk throughout the process of global routing by inserting shields, which has little influence on wire length and timing performance. (3) Take coupling inductance into consideration.
For future work, we plan to improve the timing efficiency of this algorithm. We will try to reduce the running time of local refinement step and find better strategies for crosstalk partitioning.

Table 5 Comparison of results between P1 and PO-GR

Circuits		C2	C5	C7
W mode	(P1) Wire length (*um*)	480350	1307456	1552916
	(PO-GR) Wire length (*um*)	477326	1368198	1575922
	Wire length increment	-0.63%	4.65%	1.48%
T mode	(P1) Wire length (*um*)	476424	1346876	1569366
	(PO-GR) Wire length (*um*)	479100	1280352	1567818
	Wire length increment	0.56%	-4.94%	-0.10%
	(P1) Min-R	-0.009243	0.012124	0.000034
	(PO-GR) Min-R	-0.007195	0.003439	0.001243

7. REFERENCES

[1] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "An exact algorithm for coupling-free routing", in: *Proc. ACM ISPD*, Sonoma, CA, USA, 2001, pp.10-15.
[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, J. Chen, J. Ma, and L. He. "Post global routing RLC crosstalk budgeting", in: *Proc IEEE/ACM ICCAD*, San Jose, CA, USA, 2002, pp.504-509.
[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] F. Glover, "Future paths for integer programming and links to artificial intelligence", *Computers and Operations research*, 1986, 13(5): pp.533-549.
[9] 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-I168.
[10] 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.