FREe: A Fast Routability Estimator *

Shenghua Liu, Xianlong Hong, Tong Jing, Jingyu Xu
Computer Science and Technology Department
Tsinghua University
Beijing 100084, P. R. China

***Abstract*—A fast routability estimator, namely FREe, is proposed in this paper. With the fast feedback on congestion, FREe serves incremental placement or guides routing to shorten the physical design time. FREe firstly extends an existing combinatorial model with the extended bounding box for flat tow-pin nets/sections to pre-estimate congestion probabilistically. Then, it routes all nets guided by the pre-estimation to get more exact congestion estimation, which is fast since it does not include any iteration/optimization as a router does. The idea of FREe is net-order-independent. FREe can get a uniform usage of routing resources by probability. The experimental results show that FREe can give more detailed congestion information in a short running time. Compared FREe with a recent router SSTT, we find that FREe has a good congestion correlation with SSTT, but FREe is more objective.**

I. INTRODUCTION

In recent years, with the profound development of VLSI /ULSI (very/ultra large scale integration) technology, routability estimator, as an optional module in physical design, is playing more and more important role. Its significance can be classified as the following two aspects. One is to give the feedback on congestion and serve incremental placement [1], in order to avoid congested routing solutions and achieve design closure in reasonable time [2]-[4]. The other is to give the guidance to succeeding global routing [5]-[7].

There exist some congestion estimations. [8] and [9] use empirical models. [10]-[13] employ simplified global routers to estimate congestion. However, most empirical models depend much on the characteristics of cases with a loss of generality. Even after simplification, global routers still face the trade off between accuracy and efficiency. In addition, different routers may lead to different solutions. So, the estimations with much detouring may lose objectivity and consistency.

Congestion estimation algorithms using probabilistic analysis are proposed in [2] and [3] to solve the above problem. [2] and [3] perform well in congestion estimation. The shortcoming is that they can not give more accurate details on congestion since the combinatorial model used in [2] leads to distributing wires averagely to possible paths of a bounding box, while [3] only counts L-/Z-shaped nets.

The major contribution of this paper is a routability estimator, namely FREe, with probabilistic pre-estimation in (extended) bounding box and fast actual routing. Here, we use both extended bounding box and bounding box in probabilistic pre-estimation. The reason is that extended bounding box for *flat two pins* (two pins laid in the same row or column) admits reasonable detour of nets, which can avoid over-congested areas resulted in just using bounding box. Guided by the probabilistic pre-estimation, we then perform fast actual routing without any iteration /optimization as in a router, which can get more exact congestion information and a uniform nets distribution. Some strategies, such as reusing routed wires and avoiding cycles in a net, are taken to shorten wire length. FREe is net-order-independent and can predict congested area more objectively.

The rest of this paper is organized as follows. Problem formulation and GRG are described in Section II. The details of pre-estimation model with (extended) bounding box are introduced in Section III. The algorithm of fast actual nets routing is described in Section IV. Experimental results are given in Section V. And Section VI concludes the whole paper.

II. PRELIMINARIES

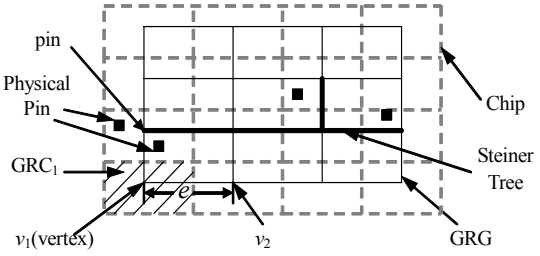


Figure 1. GRG (global routing graph)

According to the actual manufacture process, some assumptions are made as follows. 1) The routing tracks on

* This work was supported in part by the Key Project of Chinese Ministry of Education under Grant No.106008, the National Natural Science Foundation of China (NSFC) under Grant No.60373012, and the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) of China under Grant No.20050003099.

each layer of chip have a unique direction. 2) Horizontal and vertical tracks appear alternatively on each layer. 3) Every track is of width, so the number of tracks on each layer has an upper bound, which is called *capacity*. According to the first two assumptions, the projection of all the layers from top to bottom of the chip will be a 2-dimensional grid plane. A GRC (global routing cell) is defined as a grid of it. Our estimation model is based on GRG (global routing graph), which is the dual graph of GRC shown in Figure 1. Each edge has a capacity. All the *physical pins* in one GRC are abstracted as the crossing point in its center, which is called *pins* here. So, the GRG can be defined as $GRG = (V, E, C)$, where V (vertices) is the crossing points set, E (edges) is the set of horizontal and vertical edges, $a=|V|$ and $b=|E|$. A function $c: E \rightarrow Z^b_+$, where c_e is called the capacity of edge e, and Z^b_+ is the set of non-negative integral b-dimensional vectors. The congestion η_e of edge e is defined as the ratio of the number of used tracks λ_e and c_e, *i.e.* $\eta_e=\lambda_e/c_e$.

III. CONGESTION PRE-ESTIMATION USING COMBINATORIAL MODEL

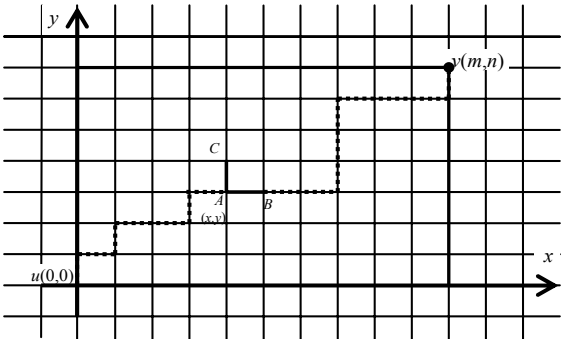


Figure 2. Grid path model on GRG

Definition 1: *Two-pin section* is defined as the two pins of a multi-pin net which are preferred to be connected directly (Steiner points can be included) by the path.

A. Two-pin nets

First, the two-pin net in GRG is considered in congestion pre-estimation.

1) Lower left and upper right

Two pins of the net are relatively lower left (pin u) and upper right (pin v). We assume u is the starting pin for the route and v is the end pin. So, a bounding box u-v is determined, and it is assumed that,

- the path between the two pins u, v cannot exceed the bounding box
- *turning back* is not allowed; i.e. only a right edge or an upper edge can be selected when standing on a vertex
- all the possible paths between the two pins are selected with the equal probability

So, the problem to calculate the possible paths from u to v is the same as the grid paths problem [14], as shown in Figure 2.

The relative Euclidean coordinates where u is the origin (0, 0) is considered. The coordinate of v is (m, n). The total number of paths from u to v is

$$T(m, n) = \frac{(m+n)!}{m!\, n!}, \text{ where } m>0, n>0 \quad (1)$$

A recursive expression can be derived as formula (2).

$$T(m, n) = T(m\text{-}1, n) + T(m, n\text{-}1), \text{ where } m, n \geq 1. \quad (2)$$

And $T(m, 0) = 1$, $T(0, n) = 1$. Due to the recursive expression, look-up table method can be employed to speed up calculation of $T(m, n)$.

As for the number of paths passing through one specific edge in the bounding box u-v, it can be calculated as follows.

$$\begin{cases} H(x, y)=T(x, y)T(m-x-1, n-y) & 0\leq x\leq m-1,\ 0\leq y\leq n \quad (3) \\ V(x, y)=T(x, y)T(m-x, n-y-1) & 0\leq x\leq m,\ 0\leq y\leq n-1 \quad (4) \end{cases}$$

where m and n are non-negative integers, $H(x, y)$ represents the path number through a specific horizontal edge in the bounding box u-v with $A(x, y)$ as the left end and $V(x, y)$ represents the path number through specific vertical edge with $A(x, y)$ as the lower end.

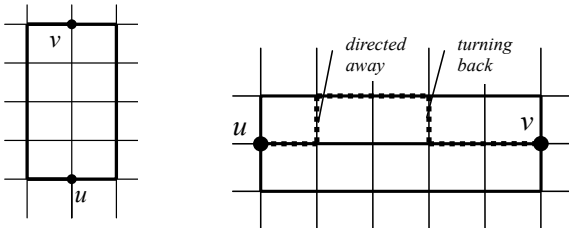
With the above analysis and assumptions, the probability of passing through the specific paths AB and AC in the bounding box u-v can be calculated by dividing $T(m, n)$ as formula (5) and formula (6).

$$\begin{cases} H(x, y)/T(m, n) & 0 \leq x \leq m-1,\ 0 \leq y \leq n \quad (5) \\ V(x, y)/T(m, n) & 0 \leq x \leq m,\ 0 \leq y \leq n-1 \quad (6) \end{cases}$$

2) Upper left and lower right

In this case, we just mirror the coordinate system, and it is the same to the above case.

3) Flat: in one row or column



(a) In one column (b) In one row

Figure 3. Extended bounding box with flat two pins

Equation (5) and equation (6) can be directly applied in this case if we use one direct line connecting the two pins. However, it differs greatly from practical cases in which a significant quantity of flat-two-pin nets or sections are allowed acceptably detour. So, *extended bounding box* strategy is necessary.

Definition 2: *Extended bounding box* is an enlarged bounding box of flat two pins, where one row or column is extended in two directions of vertical or horizontal separately, shown in Figure 3 a) and Figure 3 b).

So, detouring is allowed in this bounding box as Figure 3 shows. In addition, "not allowing turning *back*" is

modified as follows. At most one turning back is allowed in width direction, *i.e.* only one step *directed away* from target is allowed as Figure 3 b) shows. It may lead the increase ratio $2\sigma/m$ of original wire length of the net at worst, where σ is upper bound of the ratio between distances of adjacent rows and columns, and m is the row or column number between u and v. This assumption is aimed at avoiding too many *turning backs* resulting in triple wire length.

Theorem 1: there are $F(m) = m^2 + m + 1$ paths between pin u and pin v in the *extended bounding box* mode, where m is the row or column number between pin u and pin v.

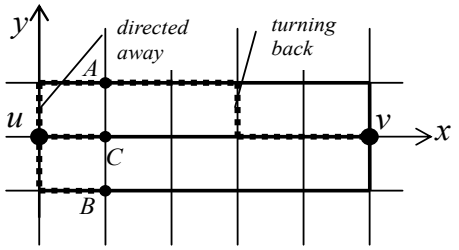


Figure 4. Augmented bounding box with flat two pins

Proof:

Without loss of generality, it is assumed that $u(0, 0)$, $v(m, 0)$ in the coordinate system shown in Figure 4. There are three ways from pin u to pin v. One is moving above directed away from v firstly, and A is reached. The following steps then are the same to *upper left and lower right* case, which path number is $T(m\text{-}1, 1)$. The second is moving below directed away from v at first, and B is reached. According to the analysis above, path number is $T(m\text{-}1, 1)$ as well. And the third is moving directly forward to C, which the path number is $F(m\text{-}1)$. Then $F(m) = 2T(m\text{-}1, 1) + F(m\text{-}1)$, where $F(1)=3$.

So, $F(m) = m^2 + m + 1$ is easily derived from it.

□

As for the number of paths passing through one specific edge it can be calculated in the following way.

$$\begin{cases} H^*(x,\ 0) = F(x) + F(m-x-1) - 1 \quad 0 \le x \le m-1 & (7) \\ H^*(x,\ \pm 1) = (x+1)(m-x) \quad 0 \le x \le m-1 & (8) \\ V^*(x,\ 0) = m \qquad 0 \le x \le m & (9) \end{cases}$$

where function $H^*(x, y)$ is the same as $H(x, y)$, and $V^*(x, 0)$ presents the number of paths passing through the above or below edge incident with vertex $(x, 0)$.

Thus, the probability of passing through one specific path in the extended bounding box u-v can be calculated by dividing $F(m)$, the same as equation (5) and equation (6).

B. Multi-pin nets

Constructing multi-pin net in GRG is defined as the rectilinear Steiner minimal tree (RSMT) problem [15][16]. Garey and Johnson [17] proved that the RSMT problem is NP-complete. And most heuristic algorithms constructing the SMT (Steiner minimal tree) are based on MST (minimum spanning tree), since there is a special relationship between them [15]. Hwang [18][19] has proved that the length of MST is at most 3/2 of that of optimal rectilinear Steiner tree. The branches of MST determine the two-pin sections preferred to be connected. Thus, the congestion estimation of multi-pin net can be calculated among every tow-pin section separately by two-pin net estimation method above.

IV. FAST ACTUAL ROUTING

A. Greedy and probabilistic selecting strategy

After pre-estimation of every net of $GRG = (V, E, C)$, a function $p: E \rightarrow R^b_+$ is established, where p_e is the probability of wires passing through edge e, and R^b_+ is the set of non-negative real b-dimensional vectors. Let $P=\{\ p_e \mid e \in E\ \}$. Thus, in order to make mathematical expectations of pre-estimated congestion on selectable edges equal, a method of connecting nets by two-pin sections is designed to choose edges in *bounding box* according to complementary probability. By using this method, relatively uniform routing is achieved. In addition, greedy and local search strategy is employed to speed up routing. The collection of selectable edges is the neighborhood in local search.

Routing in bounding box guarantees relatively shorter total wire length, and routing guided by pre-estimation guarantees a uniform result, which reflects the congestion of the chip objectively.

B. Cycle avoiding and wire reusing strategy

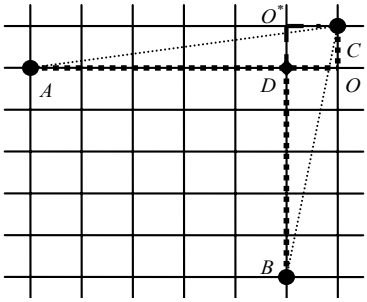


Figure 5. Cycle avoiding and wire reusing

In order to avoid cycle when routing every two-pin section of the net separately, cycle avoiding and wire reusing strategy is taken. As for the order of two-pin sections decomposed by Prim MST algorithm, one of the pins in a section must appear at least once in the previous sections, except the first one. Due to the property, the strategy succeeds by flagging. It is shown in Figure 5 that cycle DO^*OC is avoided, and wire DO and OC is reused, when AC and BC is the sequential two-pin sections.

C. Algorithm of constructing single net

Actual nets routing can be completed by invoking the algorithm in Figure 6 b) iteratively.

Construct tow-pin section

Input *GRG*=(*V*, *E*, *C*), starting pin *u*(*i.e.* vertex), flagged end pin *v* and *P*

Output A sequence of vertices *S* (including pins and Steiner points) to be connected.

1. $S=\varnothing$ $u^* \leftarrow u$
2. If pin u^* is flagged, return *S*.
3. Else flag u^* and add u^* into *S*, and find the set of all the possible following vertices V_p adjacent to and edges E_p incident with u^*, according to the assumption of routing in bounding box. For each $v_i \in V_p$, there exists a unique $e_i \in E_p$, otherwise is the same.
4. Select one edge $e_i \in E_p$, according to the probability which is reversely proportional to $\frac{p_{e_i}}{c_{e_i}}$.
5. Let vertex v_i, the other end of the selected edge e_i, be a new starting pin u^* and then go to 2.
6. Endif

(a) Sub-procedure of algorithm for routing one net

Construct Steiner tree directed by probability

Input *GRG*=(*V*, *E*, *C*) and *Q* the set of two-pin sections of the net (two-pin net can be viewed as the one of single two-pin section)

Output Steiner tree presented by a sequence of vertices ξ including pins and Steiner points.

1. (*initial*) For the first two-pin section $q_1 \in$ Q, flag either one of the pins as the end pin while the other is the starting one, and then invoke **Construct tow-pin section** sub-procedure and ξ =Output of it.
2. While there exist unrouted two-pin sections, from the next one, find a flagged pin as the end pin, and the other as starting one (MST algorithm guarantees it), and then invoke **Construct two-pin section** sub-procedure and add output into ξ.

(b) Main procedure of algorithm for routing one net

Figure 6. Algorithm for routing one net

TABLE I. CHARACTERISTICS OF TEST CASES

Test case	# of cells	# of nets	Grids
freecpu	3111	3040	74×68
u05614	32498	36452	205×205
u08421	48810	54743	251×251
u11228	65058	72968	290×290
u14035	81242	91129	324×324
u28070	162162	181934	458×458

V. EXPERIMENTAL RESULTS

The routability estimator FREe has been implemented in *C++* programming language and on *Open Access* (*OA*). *OA* is an open-source database *application program interface* (*API*) based on a standard *information model* (*IM*). The estimation of available routing resources (*i.e.* capacity on every edge of GRG) has been implemented based on the method in [20].

We compare FREe with other two related methods. One is a recent router SSTT [7], which has been implemented on *OA*. The other is a typical congestion estimator using probabilistic analysis in bounding box [2]. We implemented it here and call it *EOPA*. As shown in Table I, six cases are tested on a Linux server with 4GB memories and two 3.0GHz CPUs. FREe and the other two related methods use the same routing resource estimation [20].

We use the *minimum usage ratio (MUR)* to show the approximate lower congestion bound of each test case (if we can get the exact SMT, then it will be the exact lower congestion bound), which is the ratio of estimated minimum total length to $\sum_{e \in E} c_e \cdot l_e$, where c_e is the capacity and l_e is the length of edge *e*. *MUR* and minimum wire length of all cases are given in column 2 and column 3 in Table II a), respectively, which are estimated by FREe. The total wire length got by SSTT [7] is also given in column 4 for comparison.

TABLE II. EXPERIMENTAL RESULTS

a) Total resource used by FREe and SSTT

Test Cases	MUR	Estimated Minimum Total Length (μm)	SSTT Total Length(μm)
freecpu	57.22%	1.79×10^7	2.07×10^7
u05614	60.67%	4.44×10^8	5.02×10^8
u08421	83.09%	6.78×10^8	7.78×10^8
u11228	74.91%	8.64×10^8	1.03×10^9
u14035	167.78%	1.16×10^9	1.33×10^9
u28070	174.12%	2.43×10^9	2.83×10^9

Estimated Minimum Total Length is the total wire length got by FREe.

b) Performance comparison

Test Cases	EOPA	FREe		SSTT	
	running time (s)	running time (s)	sigma	running time (s)	sigma
freecpu	0.08	0.11	1.42	7.93	1.29
u05614	0.63	1.09	2.26	182.69	1.99
u08421	0.99	1.68	2.35	426.33	2.08
u11228	1.06	2.13	2.24	488.74	2.07
u14035	1.67	3.08	2.42	813.80	2.13
u28070	3.61	7.21	2.45	1728.25	2.20

In Table II b), column 2, column 3, and column 5 give the running time of *EOPA*, FREe, and SSTT, respectively. It is obviously that both *EOPA* and FREe are much faster than SSTT.

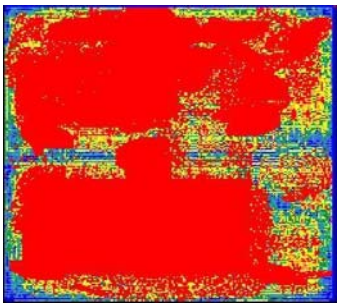
In Table II b), column 4 and column 6 show the sigma of FREe and SSTT, respectively. Here, sigma is the standard error of congestions η_e on edges of *GRG* and used to measure the distribution of the congested area, which reflects how uniform the routing resource is used. To sigma, the closer to zero from positive, the more uniform the routing resource is used. We can see that the sigma of FREe is quite close to that of SSTT but the latter is smaller, which indicates in some extent the more detoured, the more uniform the routing resource is used. The formula of sigma is as follows.

Sigma = $\sqrt{\sum_{e\in E}(\eta_e-\bar{\eta})^2}$, where $\bar{\eta}=\frac{1}{|E|}\sum_{e\in E}\eta_e$

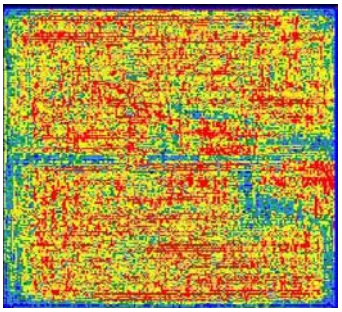
In addition, without loss of generality, density maps of case *u05614* generated by the three programs are shown in Figure 7 a), Figure 7 b), and Figure 7 c), respectively. It is obvious to see that Figure 7 b) is closer to Figure 7 c). Although there is not much detouring in FREe, it is able to correlate well with SSTT router due to the pre-estimation. And, FREe avoids that different router may lead to different routing solutions [2]. That is, FREe could estimate congestion more objectively.

Compared Figure 7 a) with Figure 7 b), we can find that FREe can give more details of congestion information, which due to the actual routing. *EOPA* just indicates roughly the most probably congested areas.

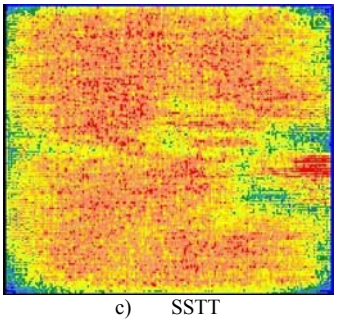
In addition, FREe can give a detailed estimating report, which includes maximal congestion of vertical/horizontal edges and summation of overflow of horizontal/vertical congested edges.



a) *EOPA*



b) FREe



c) SSTT

Figure 7. Density maps of case *u05614*

VI. CONCLUSIONS

We propose a routability estimator namely FREe. FREe can lead to more accurate and objective congestion estimation without loss of efficiency. Furthermore, the estimator is routing-order-independent and also able to obtain shorter total wire length.

With the estimated congestion information, the placement can be improved to remove congestion with many existing techniques [11][13][21]. Meanwhile, the actual nets routing in FREe can be used as a well-designed initial solution for a global router, and the congestion information can be used to guide the rip-up and rerouting in the routing phase.

REFERENCES

[1] Olivier Coudert, Jason Cong, Sharad Malik, Majid Sarrafzadeh, "INCREMENTAL CAD," *ICCAD* 2000, pp. 236-243

[2] J. Lou, S. Krishnanioorthy, and H. S. Sheng, "Estimating routing congestion using probabilistic analysis," in *PTOC. Int.* Symp**.** *on Phgsacal* Design, 2001.

[3] Jurjen Westra, Chris Bartels, Patrick Groeneveld, "Probabilistic Congestion Prediction," ISPD'04, April, 2004, pp. 204-209.

[4] Ulrich Brenner and André Rohe, "An Effective Congestion-Driven Placement Framework," *IEEE Tran. Computer-Aided Design*, *vol.* 22, *NO.* 4, April 2003 pp387.

[5] Jinjun Xiong, Lei He, "Probabilistic Congestion Model Considering Shielding for Crosstalk Reduction," *ASP-DAC* 2005 pp. 739-742.

[6] Raia T. Hadsell and Patrick H. Madden, "Improved Global Routing through Congestion Estimation," in *DAC'03,* June 2-6, 2003, Anaheim, California, USA.

[7] 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.

[8] C.-L. E. Cheng, "RISA: Accurate and efficient placement routability modeling," in *Proc. Int. Conf. Computer-Aided Design*, June 1994, pp. 690-695.

[9] M.Wang and M. Sarrafzadeh, "On the behavior of congestion minimization during placement," in *Proc. Int. Symp. Physical Design*, Apr. 1999, pp. 145-150.

[10] S. Mayrhofer and U. Lauther, "Congestion-driven placement using a new multipartitioning heuristic," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 332-335.

[11] P. N. Parakh, R. B. Brown, and K. A. Sakallah, "Congestion driven quadratic placement," in *Proc. 35th Design Automation Conf.*, June 1998, pp. 275-278.

[12] Maogang Wang, Majid Sarrafzadeh, "Modeling and minimization of routing congestion," in *Proc. Asia-Pacific Design Automation Conf.*, Jan. 2000, pp. 185-190.

[13] R. S. Tsay, S. C. Chang, and J. Thorvaldson, "Early wireability checking and 2-D congestion-driven circuit placement," in *Proc. 5th Annu. IEEE Int. ASIC Conf. and Exhibit*, Sept. 1992, pp. 50-53.

[14] Lu Kaicheng, Lu Huaming, "Combinatorial maths," Tsinghua univ. press 2002-7 version III, 2003-11 5th press, pp 35.

[15] X.L.Hong, X.L.Yan, and C.G.Qiao, "Theories and Algorithms for VLSI Layout Design", *Sicience Press*, Beijing, 1998, pp. 122, 125.

[16] Yu Hu, Tong Jing, Zhe Feng, Xian-Long Hong, Xiao-Dong Hu, Gui-ying Yan, "ACO-Steiner: Ant Colony Optimization Based Rectilinear Steiner Minimal Tree Algorithm," *J. Comput. Sci. & Technol. Jan. 2006, Vol.21, No.1*, pp.147-152.

[17] Garey M R, Johnson D S. The rectilinear Steiner tree problem is NP-complete. *SIAN J. on Applied Mathematics*, 1977, 32: 826-834.

[18] F.K.Hwang, "An O(nlogn) Algorithm for Rectilinear Steiner Trees", J. of the Association for Computing Machinery, 1976, 26(1): 177-182.

[19] F.K.Hwang, "An O(nlogn) Algorithm for Suboptimal Rectilinear Steiner Trees", IEEE Trans. on CAS, 1979, 26(1): 75-77.

[20] Andrew B. Kahng, Stefanus Mantik, and Dirk Stroobandt, "Toward accurate models of achievable routing" *IEEE Tran. Computer-Aided Design*, *vol.* 20, *Issue* 5, May 2001 pp. 648 – 659.

[21] J. M. Kleinhans, S. Sigl, F. M. Johannes, and K. J. Antreich, "GORDIAN: VLSI placement by quadratic programming and slicing optimization," *IEEE /Trans. CAD*, vol. 10, pp. 356–365, Mar. 1991.