An Integer Linear Programming Algorithm for Crosstalk-Constrained River Routing

Abstract

As the advances in technology bring about smaller interconnection wire spacing and higher circuit operating frequency, the effect of crosstalks on performance and even on yield in integrated circuit design and manufacturing thus increases rapidly. Consequently, reduction of crosstalks between interconnection wires becomes important in VLSI design. In this paper, we present an approach to the gridded river routing problem with the objective of minimizing crosstalks. Given an initial routing solution generated by a conventional river routing algorithm, the reduction of crosstalks is carried out by an iterative reassignment of the horizontal and vertical wire segments. To effectively and optimally perform the reassignment process, an integer linear programming (ILP) formulation is proposed in company with procedures for reducing the number of ILP variables and constraints. The experimental results show that this approach of ILP is very encouraging.

An Integer Linear Programming Algorithm for Crosstalk-Constrained River Routing

I   Introduction

Crosstalks being a result of mutual capacitance and inductance occurred between neighboring wires [1], cause increased delays and/or inadvertent logic transitions in VLSI circuits. As the interconnection wire spacing is getting smaller and circuit operating frequency running higher, the increasing effect of crosstalks will greatly reduce the performance and yield in integrated circuit design and manufacturing. Consequently, the reduction of crosstalks between interconnection wires becomes an important consideration in VLSI design.

In the detailed routing phase of VLSI physical design, many researches have been focused on the crosstalk reduction problem [2]-[6] [9]-[11]. In fact, solving routing problems with crosstalk reduction consideration is more difficult in comparison with the conventional routing problems because the crosstalks between interconnection wires are determined not only by how each individual wire is routed, but also by the relative positions of neighboring wires. Usually, a gridless routing model is used to solve this problem such that the spacings between interconnection wires can be modified to reduce crosstalks. Dai *et al*. [3] proposed a gridless rubber-band area routing algorithm which can create an adjustable planar sketch to enlarge the spacings for those wires having potential crosstalk problems. Onozawa *et al*. [9] proposed a one-dimensional spacing algorithm to adjust the spacings between interconnection wires in order to optimize the circuit delay and to reduce crosstalks. Their spacing algorithm utilizes the repulsive constraints to expand the distances between interconnection wires such that crosstalks can be minimized.

On the other hand, there exist also researches concentrated on the gridded routing problems with the objective of minimizing crosstalks [4]-[6] [10] [11]. Unlike the gridless routing model, interconnection wires in the gridded routing model are to be placed on grid lines, thus the spacings between grid lines cannot be adjusted to reduce crosstalks. Some of previous works using the gridded routing model adopt a *two-phase* method to reduce crosstalks [4]-[6] [10]. In the first phase, conventional routing algorithms without considering crosstalks are used to obtain an initial routing solution with some objectives such as minimizing the channel width. In the second phase,

postprocessing algorithms are designed to modify the initial routing solution for reducing crosstalks. For example, Gao and Liu proposed a track permutation technique [4] and a wire segment reassignment technique [5] to modify a given channel and a given switchbox routing solutions, respectively, to obtain new crosstalk-reduced routing solutions. The above two postprocessing techniques are formulated as two mixed integer linear programming (ILP) problems to effectively solve the crosstalk problem. In addition to the mixed ILP technique, Jhang *et al.* [6] presented an efficient approach for reassigning the wire segments of a given channel routing solution. In [10], an optimal postprocessing algorithm was proposed for minimizing the crosstalks between vertical wire segments in a three-layer VHV channel by finding an optimal vertical layer assignment for them.

Recently, a crosstalk minimization technique for the gridded river routing has been proposed by Zhou and Wong [11]. They considered the crosstalks between interconnection wires in a global way during the routing process and developed an optimal river routing algorithm. In their approach, the crosstalk minimization river routing problem is reduced to a global space allocation problem. Corresponding to the latter problem, a flow network is constructed and the flows on the network are computed to route the nets with minimum crosstalks. Since their algorithm first tries to route as much as possible parts of wires to stairs (in [11], a stair is a wire which is composed of alternating unit horizontal and vertical wire segments) and then uses space allocation to solve the placement of the straight segments, there will be many jogs (horizontal wire segments) in the routing solution. Although a postprocessing process is used to reduce the number of jogs, there could still remain a large number of jogs in some wires.

In this paper, we deal with the gridded river routing problem with the objective of minimizing crosstalks. The two-phase method is applied to solve the problem. Firstly, the conventional river routing algorithm used to generate an initial routing solution must have restriction on the number of jogs in each wire (or route defined in Section II). There are practical reasons for this. High electrical resistance and high possibility of manufacturing faults occur at the endpoints of jogs where the wires change direction. After an initial routing solution is obtained, the wire segment reassignment technique is applied to the initial routing solution to derive a new solution satisfying the problem objective, i.e., crosstalk reduction. To reassign wire segments effectively and optimally, the ILP technique is used.

The rest of the paper is organized as follows. Section II gives the problem formulation. In Section III, we present an ILP formulation of the wire segment reassignment problem and techniques for reducing the number of variables and constraints in the ILP formulation. Experimental results are presented in Section IV. Section V gives a conclusion.

II   Problem Formulation

A *channel* in the river routing model is a single-layer rectangular routing area with pins placed at the top and bottom boundaries. We assume that there are grid lines imposed on the channel. Horizontal and vertical grid lines represent the rows (tracks) and columns, respectively. Let the pins on the top boundary be denoted by t_1, t_2,..., t_n, and the pins on the bottom boundary denoted by b_1, b_2,..., b_n where n is the number of nets. A *route* W_i of net N_i, for $i = 1, 2, ..., n$, is a path connecting t_i and b_i in a *monotonic* fashion on the grid lines, that is, W_i can always be designed to monotonically go only *down-and-left* or *down-and-right* from t_i to b_i. A route is called *left route* if it is routed in a down-and-left fashion, and called *right route* if routed in a down-and-right fashion. Therefore, a route is composed of alternating vertical and horizontal wire segments. Here, vertical and horizontal wire segments mean vertical and horizontal straight line segments, respectively, and each wire segment has two endpoints. An endpoint is called a *fixed endpoint* if it is connected to the top or bottom boundary, and called a *free endpoint* if it is connected to another wire segment. The conventional gridded river routing problem is to connect pins for each net in the channel such that no two routes cross each other.

For a given river routing solution, there is a *vertical constraint* from horizontal wire segment h_i to horizontal wire segment h_j, denoted by $VC(h_i, h_j)$, if h_i must be placed above h_j in any valid routing solution obtained by reassigning the horizontal wire segments of a given initial routing solution. Unlike the vertical constraints in [13] which are defined between nets and are independent of any routing solution, the vertical constraints here are defined between horizontal wire segments in a given river routing solution. By scanning the channel of a given routing solution column by column, a set of vertical constraints can be found and they can be represented by a *vertical constraint graph* (VCG). A VCG is a directed graph where each vertex represents a horizontal wire segment and a directed edge from h_i to h_j means a vertical constraint from h_i to h_j. It should be clear that the VCG is acyclic for a valid routing solution.

If a route is traced from its pin on the top boundary to its pin on the bottom boundary, an ordered set of horizontal wire segments belonging to the route can be found. For each pair of h_i and h_j in the ordered set, if no other elements (horizontal wire segments) exist between them and placing h_i below h_j results in the violation of the monotonic property, there is a *monotonic constraint* from h_i to h_j denoted by $MC(h_i, h_j)$. Moreover, we trace other routes one by one in the same way, then a set of monotonic constraints can be found.

To obtain a high performance routing solution, it is important to take the crosstalk noise into consideration. The crosstalk between two routes is proportional to the coupling capacitance between them, which in turn is determined by the routing of the routes, i.e., by the relative positions between them. Generally speaking, the crosstalk between two parallel wire segments is proportional to their coupling length and is inversely proportional to their separating distance. Since the coupling capacitance between two parallel wire segments decreases rapidly as the distance between them increases, it is reasonable to assume that crosstalks only exist between wire segments in adjacent rows or columns. Since the distance between two adjacent rows or columns is fixed in a gridded routing area, without loss of generality, we shall regard the coupling length between two wire segments in adjacent rows or columns as the crosstalk, which is also the model used in [4], [5], [6], and [11]. The *actual crosstalk R_i* in a route W_i is defined as the sum of the crosstalks in all the wire segments of W_i. For example, in the routing of Fig. 1, considering route W_2 composed of v_3, h_2, v_4, h_3, and v_5, its crosstalk will be $1 + 0 + 2 + 1 + 2 = 6$. In practice, each route W_i can be specified by the designer a critical value called the *maximum tolerable crosstalk C_i* according to the electrical property of the circuit. The difference between C_i and R_i is called the *crosstalk slack* of W_i. The minimum value of the crosstalk slacks among all the routes is called the *minimum slack*, denoted by *minslack*. The sum of the crosstalk slacks in all the routes is denoted by *totalslack*. If the maximum tolerable crosstalk in each route is not exceeded by its actual crosstalk, i.e., minslack ≥ 0, we say that the routing solution satisfies the crosstalk constraints on all the routes, that is, the routing solution is within the safety margin. Obviously, a larger value of minslack means a better routing solution, and so for totalslack. Therefore, a good routing algorithm should generate routing solutions with both minslack and totalslack as large as possible.
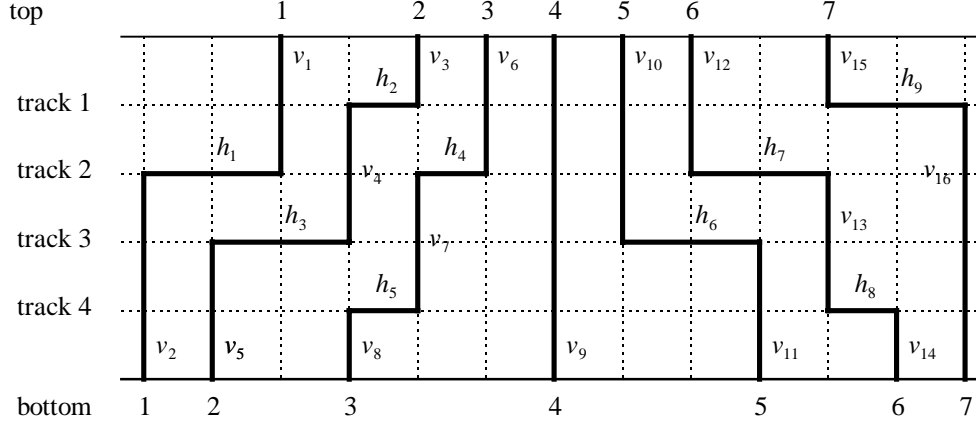
Figure 1: A river routing example.

The problem addressed in this paper is the gridded river routing problem with the objective of maximizing minslack and totalslack. Instead of directly devising algorithms for the solution of the problem, we propose here a new transformational approach which modifies a given routing solution to obtain a new one that will satisfy the problem objective. In our approach, a conventional gridded river routing algorithm having the objective of minimizing the channel separation with bounded number of jogs in each route is first used to generate an initial routing solution, then the wire segments of routes in the initial solution are reassigned to maximize minslack and totalslack. Here, the reassignment of a horizontal (or vertical) wire segment is to move it to another row (or column) while maintaining the validity of the routing solution. That is, the reassignment of the horizontal wire segments must satisfy the vertical constraints and the monotonic constraints. Our approach for the reassignment problem is to reassign the horizontal and vertical wire segments to rows and columns, respectively, in an iterative fashion. In other words, in order to maximize minslack and totalslack, the reassignment process is carried out iteratively between the horizontal and vertical wire segments until no improvement can be made on the crosstalk reduction.

Since there exist many good conventional gridded river routing algorithms generating solutions with a small number of jogs, like those presented in [8] and [12], we shall pay our attention on the reassignment problem for a given initial routing solution. To reassign the horizontal and vertical wire segments effectively and optimally, the ILP technique is used. But since the ILP formulation for the reassignment of vertical wire segments is similar to that of horizontal wire segments, we will only present the ILP formulation in terms of horizontal wire segments in the next

section. Note that after the reassignment process, channel separation does not change and the number of jogs in each route should be no greater than the previous one.

III Reassignment Approach for Crosstalk Reduction in a River Channel

In this section, we propose a new approach using ILP technique which guarantees an optimal solution for the reassignment of horizontal wire segments. Without confusion, the reassignment process for horizontal wire segments is called the reassignment process for short in this section. We present an ILP formulation for the reassignment process in detail in Subsection A and effective techniques for reducing the number of variables and constraints in the ILP formulation in Subsection B.

A An Integer Linear Programming Formulation

To maximize minslack and totalslack at the same time, the objective of the reassignment problem can be formulated as:

$$\text{maximize} \qquad w_m * \text{minslack} + w_t * \text{totalslack} \qquad (1)$$

where w_m and w_t are the weighting factors. With the above objective, the ILP formulation for the reassignment process has two kinds of ILP constraints. The first kind of ILP constraints should include the vertical constraints and the monotonic constraints as defined in Section II to maintain the validity of the resultant final routing solution. In order to compute the crosstalks between adjacent wire segments in the final solution, another kind of constraints are needed to be formulated in such a way that adjacency information between wire segments is available. We will express these constraints in terms of variables and linear constraints in the following.

Given an initial routing solution with n nets, r rows ($r \geq 1$), p horizontal wire segments, and q vertical wire segments, we denote the horizontal wire segments by $h_1, h_2,..., h_p$ and the vertical wire segments by $v_1, v_2,..., v_q$. We assume that rows and columns are labeled in increasing order with the top-most row being row 1 and the left-most column being column 1 (the top boundary is labeled as row 0 and the bottom boundary is labeled as row $r + 1$). For each horizontal wire segment h_i, there is an integer variable H_i and the value of H_i is the row in which h_i is placed in the final solution. Obviously, there are p H-variables in the ILP formulation.

For a given routing solution, the *upper* (or *lower*) *boundary* of a horizontal wire segment h_i is the feasible top-most (or bottom-most) row in which h_i can be placed in a valid routing solution. It is easy to see that the row in which h_i is placed in the final solution must fall between its upper and lower boundaries. To express the feasible row ranges of horizontal wire segments, the following constraints are introduced:

$$upper_i \leq H_i \leq lower_i \qquad \text{for } i = 1, 2,..., p \qquad (2)$$

where the values of $upper_i$ and $lower_i$ represent the row numbers of the upper and lower boundaries of h_i, respectively. Note that for a given initial solution, we can identify the upper and lower boundaries for each horizontal wire segment by traversing the VCG before the reassignment process.

Since the final solution obtained after the reassignment process must still be a valid routing solution, vertical constraints and monotonic constraints must be satisfied during the reassignment process. To express the vertical constraint $VC(h_i, h_j)$ between any two horizontal wire segments h_i and h_j, the following constraint is introduced:

$$H_i < H_j \qquad (3)$$

To express the monotonic constraint $MC(h_i, h_j)$ (both horizontal wire segments h_i and h_j belong to the same route), the following constraint is introduced:

$$H_i \leq H_j \qquad (4)$$

To compute the crosstalks between horizontal wire segments, adjacency information between horizontal wire segments must be well included in the ILP formulation. For each pair of horizontal wire segments h_i and h_j, if their coupling length is nonzero and they could be adjacent in the final solution, one 0-1 integer variable T_{ij} is introduced. $T_{ij} = 1$ if and only if h_i and h_j are reassigned to become adjacent in the final solution; otherwise $T_{ij} = 0$. Assume that h_i is placed above h_j in the initial solution, the adjacency information between h_i and h_j is expressed in the following constraints:

$$1 - T_{ij} \leq H_j - H_i - 1 \qquad (5)$$

$$H_j - H_i - 1 \leq (1 - T_{ij})\,(r - 1) \qquad (6)$$

Note that h_i should still be placed above h_j after the reassignment process. If h_i is adjacent to h_j in the final solution, then $H_j - H_i = 1$; hence, Constraint (6) is redundant and T_{ij} is forced to be 1 by Constraint (5). If h_i is not adjacent to h_j, then $H_j - H_i > 1$; hence, Constraint (5) is redundant and T_{ij} is forced to be 0 by Constraint (6) because $H_j - H_i - 1 < r - 1$. By Constraints (5) and (6), $T_{ij} = 1$ iff h_i is adjacent to h_j in the final solution.

With the T-variables, crosstalks in horizontal wire segment h_i can be computed as $\sum Q_{ij} T_{ij}$ for all h_j with T_{ij} defined where Q_{ij} is the coupling length between h_i and h_j. Note that Q_{ij} is a constant and it can be determined before the reassignment process.

Vertical wire segments can be classified into three types as follows. A vertical wire segment v_i is considered *type*-0 wire segment (*feedthrough*) if both endpoints of v_i are fixed endpoints. A segment v_i is considered *type*-1 wire segment if one endpoint of v_i is a fixed endpoint and another endpoint is a free endpoint. If both endpoints of a segment v_i are free endpoints, v_i is considered *type*-2 wire segment (*dogleg*). For each vertical wire segment v_i, two integer variables L_i and U_i are defined where the value of L_i is the row number of the top endpoint of v_i and the value of U_i is the row number of the bottom endpoint of v_i. For these two variables, the following constraints are introduced:

$$L_i = \begin{cases} 0 & \text{if the top endpoint of } v_i \text{ is connected to the top boundary} \\ H_k & \text{if the top endpoint of } v_i \text{ is connected to a horizontal wire segment } h_k \end{cases} \quad (7)$$

$$U_i = \begin{cases} r+1 & \text{if the bottom endpoint of } v_i \text{ is connected to the bottom boundary} \\ H_k & \text{if the bottom endpoint of } v_i \text{ is connected to a horizontal wire segment } h_k \end{cases} \quad (8)$$

From Constraints (7) and (8), it is clear to see that if the horizontal wire segments connected to the two endpoints of v_i are reassgined to other rows during the reassignment process, the row numbers of the two endpoints of v_i are also forced to change. However, there is no change for the type of v_i.

To compute the crosstalk between vertical wire segments v_i and v_j in adjacent columns, we need to know the actual relative positions of the endpoints of v_i and v_j in the final solution. However, since the actual relative positions in the final solution are unknown before completing the reassignment process, all possible relative positions of the endpoints of v_i and v_j will be explored. With these possible relative positions information, some constraints which will enable us to compute

the crosstalk between v_i and v_j in the final solution are introduced. According to the types of v_i and v_j in the initial solution, the possible relative positions and constraints can be classified into six categories as discussed in the following cases (Case 1 to Case 6). Note that for those constraints in Cases 1 to 6, R_{ij}^{α} denotes the crosstalk between v_i and v_j in the final solution where α indicates the types of v_i and v_j. Obviously, $R_{ij}^{\alpha} \geq 0$ for each adjacent pair of v_i and v_j. We also assume that v_i is to the left of v_j for all possible relative positions between them.

Case 1, both vertical wire segments are type-0 wire segments. Obviously, no constraint is needed in this case because the crosstalk between these two vertical wire segments is always equal to a constant $r + 1$.
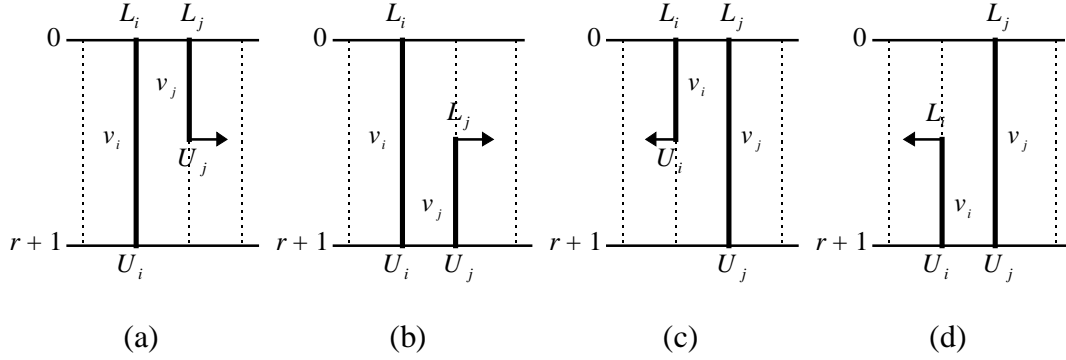

Figure 2: Relative positions of v_i and v_j in Case 2.

Case 2, one vertical wire segment is a type-0 wire segment and another is a type-1 wire segment. The possible relative positions of the endpoints of both wire segments v_i and v_j in this case are shown in Fig. 2. Note that the direction of arrow in the figure indicates that the corresponding route is a left route or a right route. Having this relative position information, the following constraint is introduced in order to compute the crosstalk R_{ij}^{01} between v_i and v_j:

$$\begin{cases} R_{ij}^{01} \geq U_j - L_j & \text{for the case of Fig. 2(a) or Fig. 2(b)} \\ R_{ij}^{01} \geq U_i - L_i & \text{for the case of Fig. 2(c) or Fig. 2(d)} \end{cases} \tag{9}$$

From Constraint (9), the crosstalk between these two wire segments completely depends on the type-1 wire segment. Since our objective is to maximize (w_m * minslack + w_t * totalslack), the

optimization process will force the value of R_{ij}^{01} to be the smallest possible value which is the exact crosstalk between v_i and v_j in the final solution.

Case 3, one vertical wire segment is a type-0 wire segment and another is a type-2 wire segment. Since these two wire segments are always non-adjacent and we assume that crosstalks only exist between wire segments in adjacent rows or columns, there is no crosstalk between these two wire segments.

Case 4, both vertical wire segments are type-1 wire segments. To simplify the constraints used to compute the crosstalk in this case, the relative positions of these two vertical wire segments v_i and v_j in the final solution can be classified into two subcases according to whether or not the fixed endpoints of v_i and v_j are connected to the same boundary (top boundary or bottom boundary).
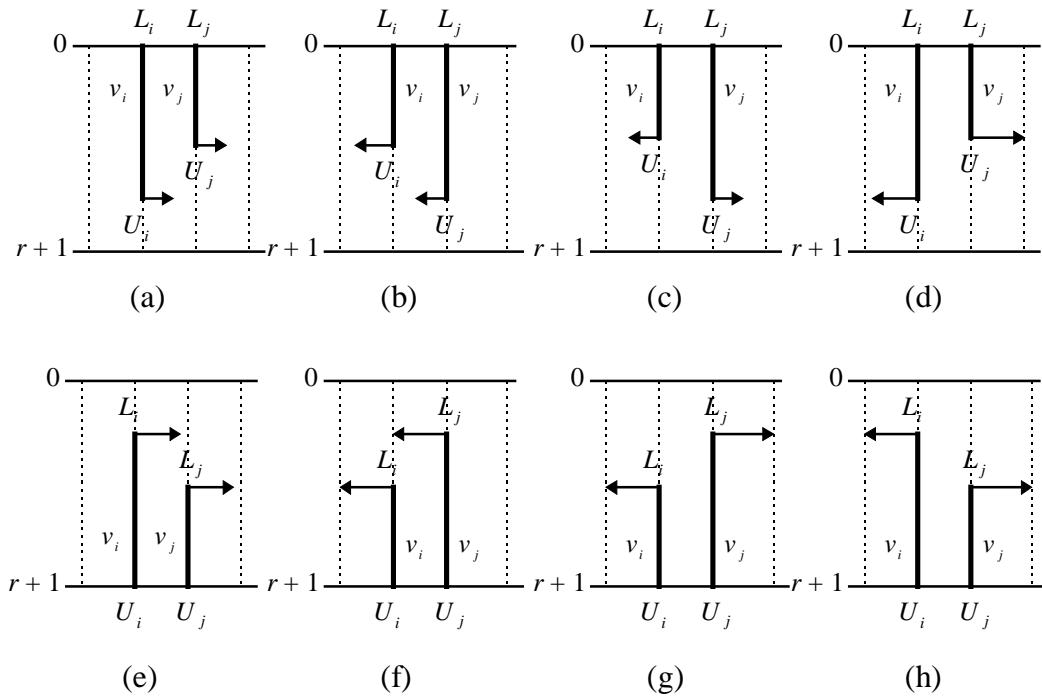

Figure 3: Relative positions of v_i and v_j in the first subcase of Case 4.

In the first subcase as shown in Fig. 3, the fixed endpoints of v_i and v_j are connected to the same boundary. To compute the crosstalk R_{ij}^{11} between v_i and v_j in this subcase, one extra 0-1 integer variable S_{ij} and the following constraints are introduced:

$$R_{ij}^{11} \geq U_i - L_i - r * S_{ij} \tag{10}$$

$$R_{ij}^{11} \geq U_j - L_j - r * (1 - S_{ij}) \tag{11}$$

The values of $(U_i - L_i)$ and $(U_j - L_j)$ represent the lengths of v_i and v_j in the final solution, respectively. From Fig. 3, if the length of v_i is less than the length of v_j in the final solution, the optimization process will force S_{ij} to be 0 and R_{ij}^{11} to be the value of $(U_i - L_i)$. Similarly, S_{ij} will be forced to be 1 and R_{ij}^{11} will be forced to be the value of $(U_j - L_j)$ if the length of v_j is less than the length of v_i in the final solution.
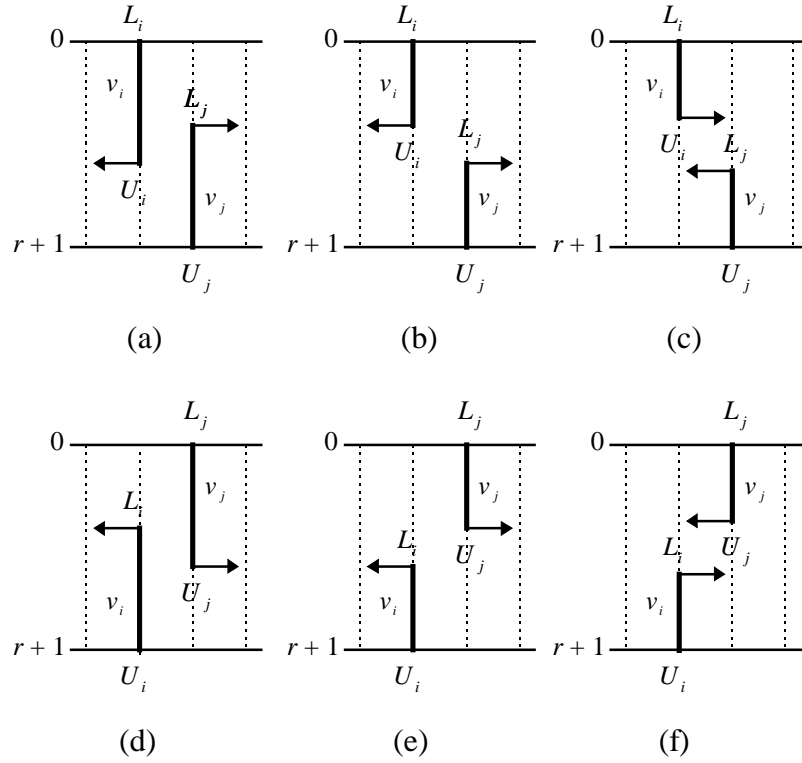

(a) (b) (c)

(d) (e) (f)

Figure 4: Relative positions of v_i and v_j in the second subcase of Case 4.

In the second subcase of Case 4 as shown in Fig. 4, the fixed endpoints of v_i and v_j are connected to the opposite boundaries. To compute the crosstalk between v_i and v_j in this subcase, one extra 0-1 integer variable O_{ij} and the following constraints are introduced:

$$R_{ij}^{11} \geq U_i - L_j - (r + 1) * O_{ij} \tag{12}$$

$$R_{ij}^{11} \geq U_j - L_i - (r + 1) * (1 - O_{ij}) \tag{13}$$

If the relative positions of the endpoints of v_i and v_j in the final solution are as shown in Fig. 4(a), the optimization process will force O_{ij} to be 0 and R_{ij}^{11} to be the value of $(U_i - L_j)$ by Constraint (12). Similarly, in the case of Fig. 4(d), $O_{ij} = 1$ gives the value $(U_j - L_i)$ for R_{ij}^{11} by Constraint (13). If the relative positions of the endpoints of v_i and v_j in the final solution are as shown in Fig. 4(b) or 4(c), the value of $(U_i - L_j)$ is negative and the value of $(U_j - L_i)$ is equal to $r + 1$, therefore, $O_{ij} = 0$ gives the smallest possible value 0 for R_{ij}^{11} by Constraint (13). Similarly, $O_{ij} = 1$ gives the smallest possible value 0 for R_{ij}^{11} by Constraint (12) for the cases in Fig. 4(e) or 4(f).
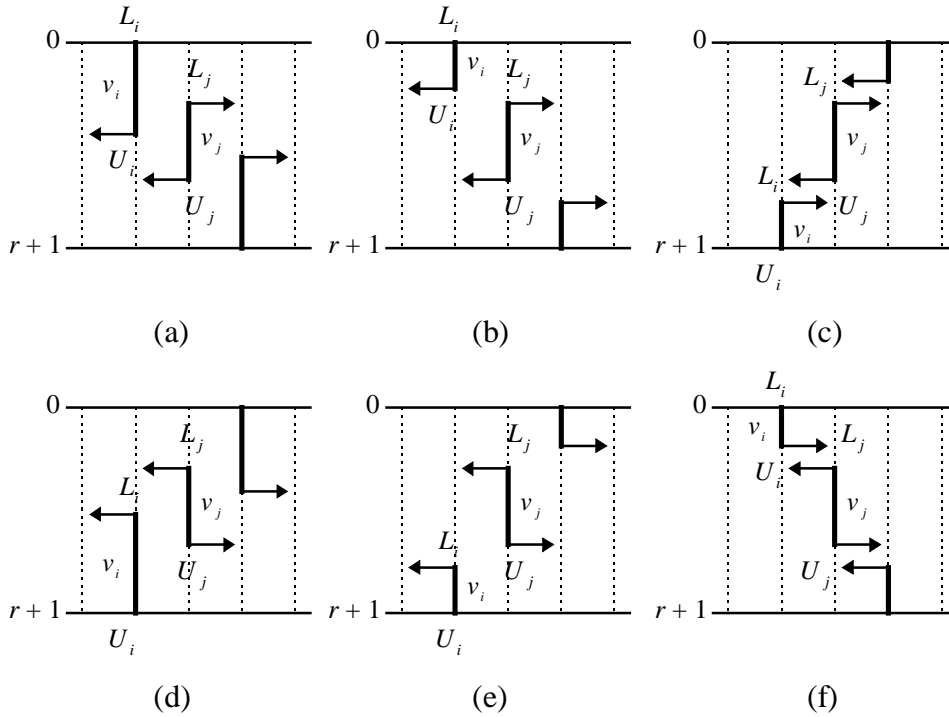

Figure 5: Relative positions of v_i and v_j in Case 5.

Case 5, one vertical wire segment is a type-1 wire segment and another is a type-2 wire segment. To minimize the number of subfigures used to show all the possible relative positions of v_i and v_j in this case, we illustrate two possible relative positions in each subfigure as shown in Figs. 5(a) to 5(f). In these subfigures, the relative positions of each pair of adjacent vertical wire segments represent one possible relative positions of v_i and v_j (as mentioned previously, v_i is always to the left of v_j for all possible relative positions between them). With this relative position information,

we can compute the crosstalk R_{ij}^{12} between v_i and v_j by using one extra 0-1 integer variable F_{ij} and the constraints below:

$$R_{ij}^{12} \geq U_i - L_j - r * F_{ij} \tag{14}$$

$$R_{ij}^{12} \geq U_j - L_i - r * (1 - F_{ij}) \tag{15}$$

$$R_{ij}^{12} \geq 0 \tag{16}$$

Using the above three constraints, the optimization process will force R_{ij}^{12} to be the exact crosstalk between v_i and v_j in the final solution.

Case 6, both vertical wire segments are type-2 wire segments. The possible relative positions of these two vertical wire segments in this case are the same as those shown in Fig. 5 except that both wire segments are type-2 wire segments. Therefore, the constraints used to compute the crosstalk R_{ij}^{22} are similar to the constraints used in Case 5 and these constraints can be shown below:

$$R_{ij}^{22} \geq U_i - L_j - (r - 1) * E_{ij} \tag{17}$$

$$R_{ij}^{22} \geq U_j - L_i - (r - 1) * (1 - E_{ij}) \tag{18}$$

$$R_{ij}^{22} \geq 0 \tag{19}$$

where E_{ij} is an extra 0-1 integer variable.

After the crosstalks in all wire segments in the final solution are computed, the actual crosstalk R_i in a route W_i, for $i = 1, 2, ..., n$, can be derived. Since the value of $(C_i - R_i)$ represents the crosstalk slack of W_i and minslack is the smallest crosstalk slack among all the routes, the following constraints are introduced:

$$\text{minslack} \leq C_i - R_i \qquad \text{for every route } W_i \tag{20}$$

Since our objective is to maximize ($w_m *$ minslack $+ w_t *$ totalslack), the value of minslack will be forced by the optimization process to be the exact smallest crosstalk slack among all the routes.

So far, the ILP formulation for the reassignment of the horizontal wire segments has been presented. After the reassignment process, a valid routing solution with better solution quality can be obtained. Next, the reassignment of the vertical wire segments will be performed to futher maximize minslack and totalslack. The reassignment process will be carried out iteratively between the horizontal and vertical wire segments until no improvement can be made on the crosstalk reduction.

In practice, the iterative reassignment process converges in few iterations, no more than three iterations, for each example in our experiments presented in Section IV.

B   Techniques for Reducing the Number of Variables and Constraints

Generally speaking, as the problem size increases, the size of the ILP formulation will proportionally become larger. In order to solve the ILP problems efficiently, it is necessary to reduce the number of variables and constraints in the ILP formulation. In this subsection, some effective techniques are proposed to minimize the number of variables and constraints.

If there exist three vertical constraints $VC(h_i, h_j)$, $VC(h_j, h_k)$, and $VC(h_i, h_k)$ in a VCG, $VC(h_i, h_k)$ is called *redundant* vertical constraint and can be deleted from the VCG because vertical constraints are transitive in the sense that $VC(h_i, h_j)$ and $VC(h_j, h_k)$ imply $VC(h_i, h_k)$, i.e., if $VC(h_i, h_j)$ and $VC(h_j, h_k)$ are satisfied, the redundant vertical constraint $VC(h_i, h_k)$ is also satisfied. Therefore, we can find all redundant vertical constraints in a VCG and remove the corresponding constraints in the ILP formulation to reduce the number of constraints.

Consider Constraints (5) and (6) which are used to express adjacency information between horizontal wire segments h_i and h_j. If the row number of the upper boundary of h_j minus the row number of the lower boundary of h_i is greater than 1 (assuming h_i is placed above h_j in the initial solution), h_i and h_j cannot become adjacent in any valid routing solution. Therefore, T_{ij} and the corresponding constraints can be removed from the ILP formulation.

In Section III.A, Cases 4, 5, and 6 may be further classified into more detailed subcases in order to minimize the number of variables and constraints. For example, in the first subcase of Case 4, if the length of a vertical wire segment is constrained to be shorter than the length of another vertical wire segment, no extra 0-1 integer variable and only one constraint are needed to compute the value of R_{ij}^{11} (assuming v_i is the shorter one):

$$R_{ij}^{11} \geq U_i - L_i \tag{21}$$

On the other hand, in the second subcase of Case 4, if we find that the fixed endpoints of v_i and v_j are connected to the top (or bottom) and bottom (or top) boundaries, respectively, and there exists a vertical constraint $VC(h_i, h_j)$ (or $VC(h_j, h_i)$) where h_i and h_j are the horizontal wire

segments connected to the free endpoints of v_i and v_j, respectively, then h_i must be constrained to be placed above (or below) h_j, i.e., there is no coupling length between v_i and v_j in the final solution. Therefore, variables R_{ij}^{11} and O_{ij} and the corresponding constraints can be completely removed from the ILP formulation. Similarly, if the two vertical wire segments in Case 5 (or Case 6) have been constrained to have no coupling length between them in the final solution, the corresponding variables and constraints can be removed from the ILP formulation.

As to the *L-* and *U*-variables, to reduce the number of these variables and the corresponding constraints (Constraint (7) for the *L*-variable and Constraint(8) for the *U*-variable), we can check if the top (or bottom) endpoint of a vertical wire segment v_i for which L_i and U_i are defined is connected to the top (or bottom) boundary. If so, L_i (or U_i) can be replaced by 0 (or $r + 1$) in each constraint having L_i (or U_i) in it and the corresponding Constraint (7) (or Constraint (8)) can also be removed from the ILP formulation.

Considering a horizontal wire segment h_i in a given routing solution, if the upper boundary of h_i is the same as the lower boundary of h_i, h_i is *unmovable*, i.e., it cannot be moved during the reassignment process. Therefore, H_i can be replaced by a value in each constraint having H_i in it where the value is the row in which h_i is placed in the initial solution. Meanwhile, the corresponding constraint used to express the feasible row ranges of h_i can be deleted from the ILP formulation. Furthermore, for a pair of unmovable horizontal wire segments h_i and h_j, if there exists $VC(h_i, h_j)$, $VC(h_j, h_i)$, $MC(h_i, h_j)$, or $MC(h_j, h_i)$, the corresponding constraint can also be removed from the ILP formulation. On the other hand, if h_i and h_j with T_{ij} defined are unmovable, it is not necessary to use Constraints (5) and (6) to express adjacency information between them and the crosstalk between h_i and h_j can be computed before the reassignment process if h_i and h_j are adjacent in the initial solution. Similarly, the number of variables and constraints used to compute the crosstalks between adjacent vertical wire segments can be greatly reduced if the ednpoints of some vertical wire segments are connected to the unmovable horizontal wire segments. For example, no variables and constraints are needed to compute the crosstalk between adjacent vertical wire segment v_i and v_j in Case 6 if all endpoints of v_i and v_j are connected to the unmovable horizontal wire segments. Of course, the crosstalk between v_i and v_j can be computed before the reassignment process if it exists.

In practice, there may have many unmovable horizontal wire segments in a given routing solution. For instance, for the examples Ex3 and Ex4 listed in Table 1, there are up to 18 and 26 unmovable horizontal wire segments, respectively (26 and 37 horizontal wire segments in totality, respectively). With this relatively large amount of unmovable horizontal wire segments, we shall have more chances to reduce the number of variables and constraints as many as possible.

Obviously, different problem instance will result in defferent reduction in the number of variables and constraints. In our experiments as shown in the next section, large reductions were obtained by using the effective techniques presented in this subsection.

IV   Experimental Results

Our algorithm was coded in C language and implemented on a SUN SPARC 10 workstation. We applied the algorithm to five river routing examples. The initial routing results for these five examples are summarized in Table 1. The example Ex5 is from the Fig. 1 published in [7] with little modification. The modification is that the bottom pins in Fig. 1 of [7] are shifted downward to the immediate next track to fit our routing model.

Table 1: Characteristics of test examples.

Ex.	# of tracks	# of columns	# of nets	# of h.w.s(jogs)	# of v.w.s.	References
Ex1	4	13	7	9	16	Fig. 1 in this paper
Ex2	5	25	9	17	26	Fig. 1 in [8]
Ex3	9	33	15	26	41	Fig. 2.8(c) in [12]
Ex4	13	38	19	37	56	Fig. 5(ii) in [8]
Ex5	6	35	17	40	57	Fig. 1 in [7]

Note that h.w.s and v.w.s represent horizontal and vertical wire segments, respectively.

It should be noted that in the initial routing solution of each example, there is a small number of jogs (horizontal wire segments) in each route. Let J denote the global upper bound on the number of jogs per route. For Ex1, Ex2, Ex3, and Ex4, $J = 2$. For Ex5, $J = 4$. Since restricting the number of jogs per route is one way to reduce the manufacturing faults, we choose these routing solutions with small J as the initial routing solutions. Of course, any routing solution with large J or no restriction

on the number of jogs per route can also be the initial routing solution. Note that the number of jogs per route in any initial routing solution should not increase after the reassignment process.

The number of variables and constraints in the ILP formulation for the first iteration of the reassignment process (i.e., the reassignment of horizontal wire segments) before and after the reduction process are summarized in Table 2. From the table, significant reduction in the number of variables and constraints is obtained for each example. Especially, for Ex4, the number of variables drops from 286 in the initial formulation to 67 in the final formulation and the number of constraints drops from 505 in the initial formulation to 118 in the final formulation. The main reason for the relatively large reduction in the number of variables and constraints for Ex4 is due to the fact that there is a large number of redundant vertical constraints and unmovable horizontal wire segments in the initial solution.

Table 2: The number of variables and constraints before and after the reduction process.

Ex.	Before reduction		After reduction	
	# of variables	# of constraints	# of variables	# of constraints
Ex1	72	95	38	59
Ex2	125	194	35	62
Ex3	220	350	49	90
Ex4	286	505	67	118
Ex5	333	517	179	283

The experimental results obtained by applying the reassignment process iteratively between the horizontal and vertical wire segments are summarized in Table 3. Since no maximum tolerable crosstalks in the routes were specified for any test example, they are assigned randomly. Meanwhile, in order to emphasize minslack in the objective of the reassignment problem, w_m and w_t were set to 10 and 1, respectively. Table 3 lists the values of minslack and totalslack of the initial solution and the final solution, respectively. As shown in the table, minslack for each example is negative in the initial solution, in other words, there exist some routes in the initial solution which do not satisfy the corresponding crosstalk constraints. After applying the reassignment technique, all the crosstalk constraints are satisfied with some margins. Moreover, the totalslack among all the routes is increased for each example. That is, the total decrement in the crosstalks in all routes is greater than

the total increment in the crosstalks in all routes. Larger totalslack means a more reliable routing solution.

Table 3: Routing results after the iterative reassignment process.

Ex.	Initial solution		Final solution				Improvement	
	minslack	totalslack	minslack	totalslack	iterations	time(sec)	minslack	totalslack
Ex1	-2	13	1	17	2	3.7	3	4
Ex2	-4	42	2	58	2	6.5	6	16
Ex3	-5	145	2	170	3	18.4	7	25
Ex4	-3	133	3	153	3	15.7	6	20
Ex5	-2	144	3	174	3	49.8	5	30

To gain a better understanding on how the crosstalks in the routes change as the wire segments are reassigned, Fig. 6 plots the value of actual crosstalk in each route before and after the reassignment technique for example Ex3. As shown in the figure, to increase the values of minslack and totalslack, the optimization process reassigns the wire segments so that the crosstalks in most routes decrease. However, since the routing area is limited, the decrease in the crosstalks in some routes may result in the increase in the crosstalks in some other routes like W_4 and W_5 in Fig. 6. Comparing the two crosstalk curves between W_1 and W_7 and between W_8 and W_{15}, we observe that relatively smaller improvements on the crosstalks are achieved between W_1 and W_7. The reason for this is that the initial routing between W_1 and W_7 is more tightly packed than that of other routing area, and therefore, there is less freedom to reassign the wire segments.
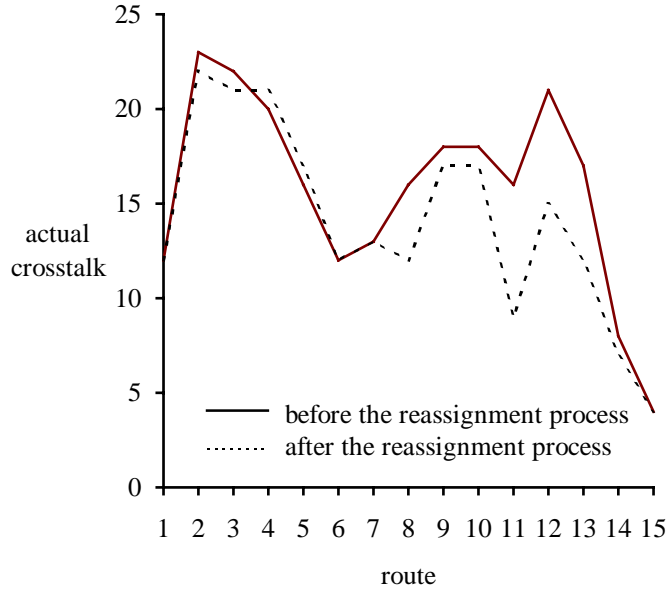
Figure 6: The actual crosstalk in each route before and after the reassignment process for example Ex3.

V   Conclusion

We have just proposed a new approach for the gridded river routing problem considering crosstalk constraints. Our approach is based on applying the wire segment reassignment technique to the initial routing solution generated by an existing conventional river routing algorithm. The reassignment of the wire segments is done by reassigning the horizontal and vertical wire segments iteratively. An ILP formulation and some effective techniques for reducing the ILP size have been presented to achieve the goal of reassigning horizontal and vertical wire segments efficiently and optimally. The obtained experimental results are very promising.

References

[1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.

[2] H. H. Chen and C. K. Wong, ″Wiring and Crosstalk Avoidance in Multi-Chip Module Design,″ in *Proc. Custom Integrated Circuits Conference*, 1992, pp. 28.6.1-28.6.4.

[3] W. M. Dai, R. Kong, J. Jue and M. Sato, ″Rubber Band Routing and Dynamic Data Representation,″ in *Proc. IEEE Int. Conf. Computer Aided Design*, 1990, pp. 52-55.

[4] T. Gao and C. L. Liu, ″Minimum Crosstalk Channel Routing,″ in *Proc. IEEE Int. Conf. Computer Aided Design*, 1993, pp. 692-696.

[5] T. Gao and C. L. Liu, ″Minimum Crosstalk Switchbox Routing,″ in *Proc. IEEE Int. Conf. Computer Aided Design*, 1994, pp. 610-615.

[6] K.-S. Jhang, S. Ha and C. S. Jhon, ″COP: A Crosstalk Optimizer for Gridded Channel Routing,″ *IEEE Trans. Computer-Aided Design*, vol. 15, no. 4, pp. 424-429, Apr. 1996.

[7] A. Mirzaian, ″River Routing in VLSI,″ *J. Comput. Syst. Sci.*, vol. 34, no. 1, pp. 43-54, 1987.

[8] A. Mirzaian, ″A Minimum Separation Algorithm for River Routing with Bounded Number of Jogs,″ in *Proc. IEEE Int. Conf. Computer Aided Design*, 1989, pp. 10-13.

[9] A. Onozawa, K. Chaudhary and E. S. Kuh, ″Performance Driven Spacing Algorithms Using Attractive and Repulsive Constraints for Submicron LSI′s,″ *IEEE Trans. Computer-Aided Design*, vol. 14, no. 6, pp. 707-719, Jun. 1995.

[10] S. Thakur, K. Y. Chao and D. F. Wong, ″An Optimal Layer Assignment Algorithm for Minimizing Crosstalk for Three Layer VHV Channel Routing,″ in *Proc. ISCAS*, 1995, pp. 207-210.

[11] H. Zhou and D. F. Wong, ″An Optimal Algorithm for River Routing with Crosstalk Constraints,″ in *Proc. IEEE Int. Conf. Computer Aided Design*, 1996, pp. 310-315.

[12] T. C. Tuan and S. L. Hakimi, ″River Routing with a Small Number of Jogs,″ *SIAM J. Discrete Math.*, vol. 3, no. 4, pp. 585-597, 1990.

[13] T. Yoshimura and E. S. Kuh, ″Efficient Algorithms for Channel Routing,″ *IEEE Trans. Computer-Aided Design*, vol. 1, no. 1, pp. 25-35, Jan. 1982.