Minimum Crosstalk Switchbox Routing*

Tong Gao and C. L. Liu
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois, 61801

Abstract

As technology advances, interconnection wires are placed in closer proximity. Consequently, reduction of crosstalks between interconnection wires becomes an important consideration in VLSI design. In this paper, we study the gridded switchbox routing problems with the objectives of satisfying crosstalk constraints and minimizing the total crosstalk in the nets. We propose a new approach to the problems which utilizes existing switchbox routing algorithms and improves upon the routing results by re-assigning the horizontal and vertical wire segments to rows and columns, respectively, in an iterative fashion. This approach can also be applied to the channel routing problem with crosstalk constraints. A novel mixed ILP formulation and effective procedures for reducing the number of variables and constraints in the mixed ILP formulation are then presented. The experimental results are encouraging.

1 Introduction

As fabrication technology advances, interconnection wires are placed in closer proximity. Also, decrease in circuit delay will enable circuits to operate at higher frequencies. Consequently, reduction of crosstalks between interconnection wires becomes an important consideration in VLSI design. Crosstalk between two wires is proportional to their coupling capacitance, which in turn, is proportional to their coupling length (the total length of their overlapping segments) and inversely proportional to their separating distance.

Crosstalks produce noises which could lead to unexpected circuit behavior. Usually, in the design specification, the maximum tolerable crosstalk for each net that will guarantee the correct behavior of the circuit is given. Such constraints on crosstalks are called *crosstalk constraints*. To reduce the total amount of noise in a design, it is also desirable that the total crosstalk in the nets is minimized. Since crosstalks depend not only on the coupling capacitance between the nets, but also on the frequency of the signals traveling in the nets, in order to simplify our presentation, we assume that the circuit operates at a fixed frequency and the value of the crosstalk in a net is directly proportional to the coupling capacitance between the net and its neighboring nets. Moreover, we will use the terms "coupling capacitance" and "crosstalk" interchangeably throughout this paper.

Crosstalks between interconnection wires are determined by the routing of the wires. Routing problems with crosstalk constraints are more difficult to solve in comparison with the conventional routing problems because crosstalks between wire segments are not only decided by how each individual wire is routed, but also by the relative positions of the wire segments.

In [1, 2], routing algorithms for gridless routing problems with crosstalk constraints were presented. In [3], a permutation algorithm for gridded channel routing problems with crosstalk constraints was presented. In the permutation approach, the tracks in an initial routing solution are permuted to obtain the final routing solution. Since all the wire segments on a track in the initial routing solution are constrained to be permuted to the same track in the final routing solution, the solution space explored is relatively small. Because a switchbox has fixed pins on all four boundaries, the permutation approach can not be generalized to solve switchbox routing problems. To be able to solve the channel and switchbox routing problems with crosstalk constraints effectively, we propose here a new transformational approach which modifies a routing solution to obtain a new one that will satisfy the crosstalk constraints and minimize the total crosstalk all the nets. In our approach, a conventional routing algorithm is first used to generate an initial routing solution with conventional objectives (e.g., minimizing channel height for the channel routing problems). The wire segments in the initial routing solution are then re-assigned to satisfy the crosstalk constraints and to minimize the total crosstalk in the nets. By re-assigning a wire segment, we mean to move a horizontal wire segment to another row (track) or a vertical wire segment to another column[1] while maintaining the validity of the routing solution.

For a channel routing problem with crosstalk constraints, the horizontal wire segments in the initial routing solution are re-assigned to tracks to satisfy the crosstalk constraints and to minimize the total crosstalk in the nets. Re-assigning the horizontal wire segments changes the relative positions of the horizontal wire segments and the lengths of the vertical wire segments, which in turn, change the values of the crosstalks in the nets. The process of re-assigning hor-

*This work was partially supported by the National Science Foundation under Grant MIP-92-22408

[1]including the possibility of not moving the wire segment.

izontal (or vertical) wire segments will be referred to as the *assignment process*. Since the assignment process re-assign wire segments to existing rows, channel height does not change after the assignment process.

For a switchbox routing problem with crosstalk constraints, the assignment process is carried out iteratively between the horizontal and vertical wire segments in order to satisfy the crosstalk constraints and to minimize the total crosstalk in the nets. The iterative assignment process continues until no more improvement can be obtained on the values of the crosstalks. Since each intermediate routing solution is a valid routing solution, the final routing solution is also a valid routing solution.

Since there are many good conventional switchbox and channel routing algorithms, we shall focus our attention on the assignment problem for a given initial routing solution. Since the assignment process for the vertical wire segments is similar to the assignment process for the horizontal wire segments, we shall only present our algorithm in terms of the horizontal wire segments.

2 Problem Formulation

A routing area is a rectangular region with pins placed on the four boundaries. For a channel routing problem, the pins on the top and bottom boundaries are fixed. For a switchbox routing problem, the pins on all four boundaries are fixed. For a gridded routing problem, it is assumed that there is an orthogonal gird superimposed on the routing area. All pins and wires are laid along grid edges (Figure 1). Two routing layers are available, one for the horizontal wires and the other one for the vertical wires. No wire segments of different nets may overlap.

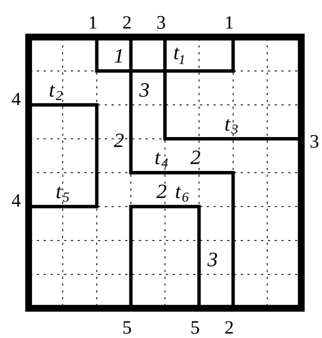

Figure 1. Crosstalks and constraints

Our algorithm is capable of considering crosstalks between any pair of parallel wire segments. Since crosstalk between two wire segments decreases as their separating distance increases, and since it is likely that crosstalk between two non-adjacent wire segments will be shielded by other wires between them, to simplify the computation, we shall assume that crosstalks only exist between adjacent wire segments. Our algorithm can be generalized immediately, however, to consider crosstalks between non-adjacent wires as well. Since the grid spacing is fixed, the crosstalk between two wire segments in adjacent rows or columns is proportional to their coupling length. Without loss of generality, we shall set the proportional constant to 1 and consider the coupling length the same as the crosstalk. The crosstalk C_k in a net N_k is the sum of the crosstalks in all the wire segments of N_k. As an example, the crosstalk in N_2 in Figure 1 is $1+2+3+2+2+3=13$. The difference between the maximum tolerable crosstalk M_k in N_k and C_k, $m_k - C_k$ is called the *crosstalk slack* of N_k. The minimum value of the crosstalk slacks among all the nets is called the *minimum slack*, and will be denoted by *minslack*. In order to satisfy all the crosstalk constraints, it is necessary that $minslack \geq 0$. Since *minslack* measures the safety margin on the crosstalk constraints, a good routing solution should have *minslack* as large as possible. The sum of the crosstalks in all the nets is denoted *totalcross*. To be able to maximize *minslack* and minimize *totalcross* at the same time, the objective of the assignment problem is defined as maximizing $w_s \times minslack - w_c \times totalcross$ where w_s and w_c are weight factors. By assigning a relatively large value to w_s, more emphasis will be placed on maximizing *minslack*. By assigning a relatively large value to w_c, more emphasis will be placed on minimizing *totalcross*.

For a given routing solution, there is a *vertical constraint* from t_i to t_j if t_i must be placed above t_j in a valid routing solution. For example, in Figure 1, since wire segment t_1 has a via which is connected to a pin on the top boundary that is directly above a via in t_4 which is connected to a pin on the bottom boundary, there is a vertical constraint from t_1 to t_4. The vertical constraints are transitive in the sense that a vertical constraint from t_i to t_j and a vertical constraint from t_j to t_k *imply* a vertical constraint from t_i to t_k. In Figure 1, a vertical constraint from t_1 to t_4 and a vertical constraint from t_4 to t_6 imply a vertical constraint from t_1 to t_6. Notice also that a vertical constraint might be *implied* by the possible locations of the horizontal wire segments. In Figure 1, since t_2 is connected to a fixed pin in row 2 and t_5 is connected to a fixed pin in row 5, a vertical constraint is implies from t_2 to t_5. Vertical constraints that are not implied by other vertical constraints or the possible locations of the horizontal wire segments are called *essential vertical constraints*, otherwise, they are called *implied vertical constraints*. If all the essential vertical constraints are satisfied, the implied vertical constraints are also satisfied.

For a given routing solution, there is a *horizontal constraint* between horizontal wire segment t_i and t_j if placing them in the same row causes them to overlap (or causes their endpoints to overlap if t_i and t_j belong to different nets). For example, in Figure 1, there is a horizontal constraint between t_3 and t_6.

For a pair of vertical wire segments t_k and t_l in the same column, if there is no vertical constraint between any horizontal wire segment connected to t_k and any horizontal wire segment connected to t_l, then t_k can be placed either above or below t_l. (Obviously, both t_k and t_l are doglegs.) However, they can not overlap.

Such constraints between doglegs will be referred to as the *non-overlapping constraints*.

Given an initial routing solution, an *assignment* of the horizontal wire segments must satisfy the horizontal constraints, essential vertical constraints, and non-overlapping constraints. The objective of the assignment problem is to maximize $w_s \times minslack - w_c \times totalcross$.

3 A Mixed ILP Formulation

In this section, we shall formulate the assignment problem as a mixed integer linear programming (ILP) problem.

Since crosstalks in a wire segment are due to the wire segments in adjacent rows or columns, wire segment adjacency information is needed in the mixed ILP formulation. Since the vertical, horizontal, and non-overlapping constraints are constraints on the relative positions of the wire segments, information on the relative positions of the wire segments are also needed in the mixed ILP formulation. These two kinds of information are closely related, and must be expressed in an unified framework in the mixed ILP formulation.

T-variables: For an initial routing solution with n horizontal wire segments and m rows, denote the horizontal wire segments by t_1, t_2, ..., t_n. Assume that the rows and columns are labeled in increasing order with the top most row being row 1 and the leftmost column being column 1 (the top boundary is labeled as row 0 and the bottom boundary is labeled as row $m+1$). For each horizontal wire segment t_i, there is a general integer variable T_i the value of which is the row in which t_i is placed in the final solution. These variables will be referred to as T-variables. Obviously, $1 \leq T_i \leq m$ for $1 \leq i \leq n$.

Vertical constraints: An essential vertical constraint from t_i to t_j can be expressed by the linear constraint $T_i < T_j$. It is not necessary to introduce linear constraints corresponding to the implied vertical constraints in the mixed ILP formulation.

Horizontal constraints: To express a horizontal constraint between horizontal wire segments t_i and t_j, we need to express the condition "$T_i > T_j$ *or* $T_i < T_j$" in terms of linear constraints. To do so, one extra 0-1 integer variable z_{ij} and the following linear constraints are introduced:

$$-mz_{ij} < T_i - T_j \quad (1)$$

$$T_i - T_j < m(1 - z_{ij}) \quad (2)$$

Obviously, the difference between T_i and T_j is less than m. If $z_{ij} = 0$, Constraint 2 is redundant and Constraint 1 is equivalent to $T_j < T_i$. If $z_{ij} = 1$, Constraint 1 is redundant and Constraint 2 is equivalent to $T_i < T_j$. Since there is no other constraint on z_{ij}, Constraints 1 and 2 are equivalent to "$T_i < T_j$ *or* $T_i > T_j$".

Non-overlapping constraints between doglegs: To express the non-overlapping constraints between doglegs t_i and t_j, we need to express the condition "$u_i < d_j$ *or* $d_j < u_i$" in terms of linear constraints, where u_i is the row number of the bottom endpoint of t_i, d_i is the row number of the top endpoint of t_i, u_j is the row number of the bottom endpoint of t_j, and d_j is the row number of the top endpoint of t_j. u_i and d_i are defined as continuous variables and the following constraints are introduced:

$$u_i \geq T_k \text{ for each hor. seg. } t_k \text{ connected to } t_i \quad (3)$$

$$d_i \leq T_k \text{ for each hor. seg. } t_k \text{ connected to } t_i \quad (4)$$

The constraints in 3 force the value of u_i to be at least as large as the row number of the bottom endpoint of t_i, and the constraints in 4 forces the value of d_i to be at least as small as the row number of the top endpoint of t_i. Because of the objective function of the mixed ILP formulation and the form of the signs of the coefficient of u_i and d_i in the constraints, as will be shown later, the optimization process will force the values of u_i and d_i to be sufficiently close to their true values so that the non-overlapping constraints are satisfied if and only if the constraints in 3 and 4 are satisfied. Moreover, the optimization process will force the values of u_i and d_i to be sufficiently close to their true values so that the value of the objective function is computed correctly.

To express the condition "$u_i < d_j$ *or* $d_i > u_j$" in terms of linear constraints, one extra 0-1 integer variable z_{ij} and two linear constraints are introduced:

$$u_i \quad < \quad d_j + mz_{ij} \quad (5)$$

$$u_j \quad < \quad d_i + m(1 - z_{ij}) \quad (6)$$

Since t_i and t_j are both doglegs, $u_i - d_j < m$ and $u_j - d_i < m$. If $z_{ij} = 0$, Constraint 6 is redundant, and Constraint 5 is equivalent to $u_i < d_j$. If $z_{ij} = 1$, Constraint 5 is redundant, and Constraint 6 is equivalent to $u_j < d_i$. Since there is no other constraint on z_{ij}, Constraints 5 and 6 are equivalent to "$u_i < d_j$ *or* $u_j < d_i$".

Crosstalks between vertical wire segments: Denote the crosstalk between vertical wire segments t_k and t_l in adjacent columns by C. Obviously, $C \geq 0$. Without knowing the relative positions of the endpoints of t_k and t_l in the final solution, two extra 0-1 integer variables y_{kl} and z_{kl} and the following constraints are introduced:

$$C \geq u_l - d_k - (m+1)y_{kl} - (m+1)z_{kl} \quad (7)$$

$$C \geq u_l - d_l - (m+1)y_{kl} - (m+1)(1 - z_{kl}) \quad (8)$$

$$C \geq u_k - d_k - (m+1)(1 - y_{kl}) - (m+1)z_{kl} \quad (9)$$

$$C \geq u_k - d_l - (m+1)(1 - y_{kl}) - (m+1)(1 - z_{kl}) \quad (10)$$

We shall show here that the optimization process will force the value of C to its true value for the two possible relative positions of t_k and t_l shown in Figure 2. It can be shown in a similar way that this is also the case for other possible positions of t_k and t_l.

Since $u_k \leq m+1$, $u_l \leq m+1$, $d_k \geq 0$, and $d_l \leq 0$, it follows that $u_l - d_k \leq m+1$, $u_l - d_l \leq m+1$, $u_k - d_k \leq m+1$, and $u_k - d_l \leq m+1$. If $y_{kl} = 0$ and $z_{kl} = 0$, Constraint 7 is equivalent to $C \geq u_l - d_k$ and the other

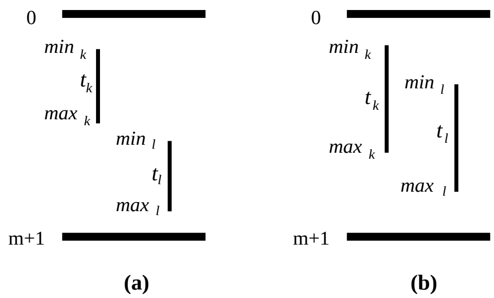

Figure 2. Possible relative positions of t_k and t_l

constraints become redundant. If $y_{kl} = 0$ and $z_{kl} = 1$, Constraint 8 is equivalent to $C \geq u_l - d_l$ and the other constraints become redundant. If $y_{kl} = 1$ and $z_{kl} = 0$, Constraint 9 is equivalent to $C \geq u_k - d_k$ and the other constraints become redundant. Finally, if $y_{kl} = 1$ and $z_{kl} = 1$, Constraint 10 is equivalent to $C \geq u_k - d_l$ and the other constraints become redundant. If the relative positions of the endpoints of t_k and t_l in the final solution are that shown in Figure 2a, the optimization process will force the value of y_{kl} to be 1, the value of z_{kl} to be 1, and the value of C to be 0. If the relative positions of the endpoints of t_k and t_l in the final solution are that shown in Figure 2b, the optimization process will force the value of y_{kl} to be 1, the value of z_{kl} to be 1, and the value of C to be $u_k - d_l$. Moreover, the value of u_k will be forced to be the row number of the bottom endpoint of t_k, and the value of d_l will be forced to be the row number of the top endpoint of t_l.

Crosstalks between horizontal wire segments: To compute the crosstalks between horizontal wire segments, we need the adjacency information between horizontal wire segments in the final solution. A 0-1 integer variable P_{ij} is defined for each pair of horizontal wire segments t_i and t_j if their coupling length is non-zero and they may be assigned to adjacent rows in the final solution. $P_{ij} = 1$ if and only if t_i and t_j are in adjacent rows in the final solution, and $P_{ij} = 0$ otherwise. These variables will be referred to as P-variables. So that the T-variables and the P-variables will be consistent in the sense that they will represent the same assignment, for each P_{ij}, an extra 0-1 integer variable z_{ij} and the following constraints are introduced:

$$1 - P_{ij} - (m+1)z_{ij} \leq T_i - T_j - 1 \qquad (11)$$

$$1 - P_{ij} - (m+1)(1 - z_{ij}) \leq T_j - T_i - 1 \qquad (12)$$

$$T_i - T_j - 1 \leq (1 - P_{ij})(m-1) \qquad (13)$$

$$T_j - T_i - 1 \leq (1 - P_{ij})(m-1) \qquad (14)$$

Notice that $T_i - T_j - 1 < m - 1$ and $T_j - T_i - 1 < m - 1$. If $T_i < T_j$, the right hand side of Constraint 11 is smaller than 0, and the value of z_{ij} is forced to be 1 by Constraint 11. Similarly, if $T_i > T_j$, the value of z_{ij} is forced to be 0 by Constraint 12. If $T_i - T_j = 1$ (t_i is adjacent to t_j), the right hand side of Constraint 11 is equal to 0. Since the value of z_{ij} is forced to be 0 by Constraint 12, the value of P_{ij} is forced to be 1 by Constraint 11. Similarly, if $T_j - T_i = 1$, the value of P_{ij} is forced to be 1 by Constraint 12. If $T_i - T_j > 1$ (t_i is not adjacent to t_j), the left hand side of Constraint 13 is greater than 0, and Constraint 13 forces the value of P_{ij} to 0. Similarly, if $T_j - T_i > 1$, Constraint 14 forces the value of P_{ij} to 0.

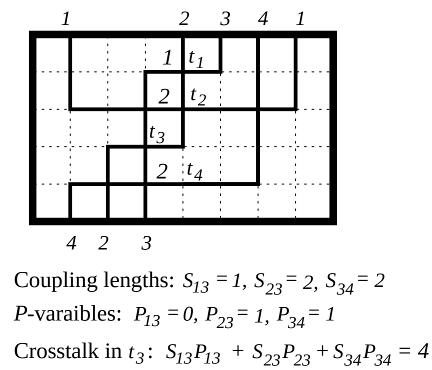

Figure 3. Crosstalk in horizontal wire segment t_3

With the P-variables, crosstalks in horizontal wire segment t_i can be computed as $\sum S_{ij} P_{ij}$ for all t_j with P_{ij} defined where S_{ij} is the coupling length between t_i and t_j (Figure 3). Notice that S_{ij} are constants which can be determined before the assignment problem is solved.

Minimum crosstalk slack: The crosstalk in a net is the sum of the crosstalks in all wire segments of the net. *minslack* is defined as a continuous variable and the following constraints are introduced:

$$minslack \leq M_i - C_i \text{ for every net } N_i \qquad (15)$$

where M_i is the maximum tolerable crosstalk in N_i, and C_i is the total crosstalk in N_i. Constraint 15 force *minslack* to be at least as small as the smallest crosstalk slack among all the nets. The optimization process will force the value of *minslack* to be exactly the smallest crosstalk slack among all the nets.

4 Reducing the Size of ILP Formulation

For large problem instances, the problem size of the mixed ILP formulation might become very large. In order to solve the mixed ILP problem efficiently, the size of the formulation need to be reduced.

A *constraint graph* is a graph with both directed and undirected edges in which the vertices represent the horizontal wire segments, solid directed edges represent the essential vertical constraints, dotted directed edges represent the implied vertical constraints, and undirected edges represent the horizontal constraints (Figure 4). Given an initial routing solution (Figure 4a), by scanning the routing area column by column, a set of vertical constraints which includes all the essential vertical constraints (and maybe some implied vertical constraints) can be found (Figure 4b). By checking the coupling length between horizontal wire segments, a set of horizontal constraints can be

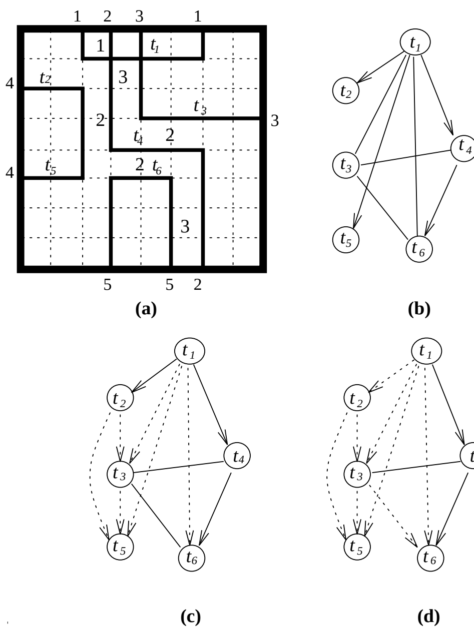

Figure 4. Constraint graph

found (some of them might be implied by the vertical constraints) (Figure 4b). Without knowing whether a vertical constraint is essential or implied, it is conservatively assumed that all the vertical constraints are essential. Since a vertical constraint implies a horizontal constraint, if there is a vertical constraint from t_i to t_j, the horizontal constraint between t_i and t_j can be removed from the constraint graph. To minimize the number of horizontal and essential vertical constraints that need to be included in the mixed ILP formulation, it is desirable to find as many implied vertical constraints as possible. Here, we will demonstrate the techniques of finding the implied vertical constraints by using the example shown in Figure 4a.

First, because of the transitivity of the vertical constraints, it is easy to find some of the implied vertical constraints (e.g., the vertical constraint from t_1 to t_6 in Figure 4c).

Next, the top and bottom most row that each horizontal wire segment may occupy are computed. The top and bottom most row that t_i may occupy gives the range of values of T_i may assume. Reducing the ranges of the T-variables lead to reduction in the number of essential vertical constraints and horizontal constraints.

To reduce the range of T_i, we first check if t_i is connected to a fixed pin on the boundaries. In Figure 4a, since t_2 is connected to a fixed pin on row 2, the top and bottom most row that t_2 may occupy are both 2. Similarly, the top and bottom most row that t_3 may occupy are both 3, and the top and bottom most row that t_5 may occupy are both 5. Since the rows that t_2, t_3, and t_5 may occupy are fixed, variables T_2, T_3, and T_5 can be eliminated. Moreover, from the top and bottom most row that t_2, t_3, and t_5 may occupy, we can derive immediately the vertical constraints from t_2 to t_3, from t_3 to t_5, and from t_2 to t_5 are implied vertical constraints. The vertical constraint from t_1 to t_5 is also an implied one (Figure 4c).

The range of T_i can be further reduced by checking the vertical constraints (both essential and implied) from t_i to other wire segments and from other wire segments to t_i. In Figure 4, since the bottom most row that t_2 may occupy is row 2, the vertical constraint from t_1 to t_2 implies that the bottom most row that t_1 may occupy is $2-1=1$. Since the top most row that t_1 may occupy is also 1, T_1 has fixed value of 1 and can be eliminated as a variable. Since the bottom most row that t_1 may occupy (row 1) is higher than the top most row that t_2 may occupy (row 2), the vertical constraint from t_1 to t_2 is an implied vertical constraint (Figure 4d). Similarly, because of the vertical constraint from t_1 to t_4 and the vertical constraint from t_4 to t_6, the top most row that t_6 may occupy is at least 3. Because t_3 is fixed in row 3, the top most row that t_6 may occupy is 4, and the vertical constraint from t_3 to t_6 is an implied one (Figure 4d).

As was discussed in Section 3, without knowing the relative positions of the endpoints of the vertical wire segments t_k and t_l in adjacent columns, two 0-1 integer variables y_{kl} and z_{kl} and Constraints 7, 8, 9, and 10 are needed to compute the value of the crosstalk C between t_k and t_l. From the range of the T-variables, the ranges of u_k, d_k, u_l, and d_l can be computed. If the minimum possible value of u_k is greater than the maximum possible value of u_l, only one 0-1 integer variable z_{kl} and two constraints are needed:

$$C \geq u_l - d_k - (m+1)z_{kl} \qquad (16)$$

$$C \geq u_l - d_l - (m+1)(1-z_{kl}) \qquad (17)$$

Similarly, if the maximum possible value of u_k is smaller than the minimum possible value of u_l, the number of variables and constraints can be reduced.

As was discussed in Section 3, for a P-variable defined between t_i and t_j where t_i was in a row above t_j in the initial routing solution, a 0-1 integer variable z_{ij} and Constraints 11, 12, 13, and 14 are needed to force the value of P_{ij} to be consistent with the values of T_i and T_j. Since either Constraint 11 or Constraint 12 will be violated if $T_i = T_j$, Constraints 11 and 12 imply a horizontal constraint between t_i and t_j. Therefore, if P_{ij} is defined, the horizontal constraint between T_i and T_j can be excluded in the mixed ILP formulation. Moreover, if there exists a vertical constraint from t_i to t_j, only two constraints are needed to force the value of P_{ij} to be consistent with the values of T_i and T_j:

$$1 - P_{ij} \leq T_j - T_i - 1 \qquad (18)$$

$$T_j - T_i - 1 \leq (1 - P_{ij})(m-1) \qquad (19)$$

If the difference between the top most row that t_j may occupy and the bottom most row that t_i may occupy is greater than 1, $P_{ij} = 0$ and the corresponding variables and constraints can be eliminated.

5 Experimental Results

Our algorithm was implemented in C and executed on an IBM RISC6000 workstation. The Optimization Subroutine Library (OSL) distributed by IBM was used to solve the mixed ILP problems. Our algorithm was tested on several benchmark circuits and many randomly generated circuits. Here, we shall present the results of running our algorithm on two randomly generated switchbox routing circuits Example1 and Example2, two well known switchbox benchmark circuits Burstein's Difficult problem and the Extended Burstein's Difficult problem, and four channel routing circuits used in [3]. The input specifications are summarized in Table 1.

cir. name	initial solution		
	no. of rows	no. of columns	no. of nets
Example1	18	18	18
Example2	21	21	27
Burstein	15	23	24
Ext-Burstein	21	22	24
Random	5	20	10
YK3C	18	79	54
D1	18	155	65
Deutsch	19	174	72

Table 1. Switchbox routing circuit specifications

cir. name	before reduction		after reduction	
	no. of var.	no. of constr.	no. of var.	no. of constr.
Example1	536	822	128	239
Example2	854	1339	143	279
Burstein	829	1371	50	105
Ext-Burstein	1044	1650	62	128

Table 2. Sizes of the mixed ILP formulation

To demonstrate the effectiveness of the techniques for reducing the problem size presented in Section 4, Table 2 lists the problem size for the switchbox routing problems before and after reduction. Substantial reduction was obtained in the number of variables and constraints for all four circuits.

cir. name	init. solution		final solution		
	min-slack	*total-cross*	*min-slack*	*total-cross*	time (min.)
Example1	-10	510	1	414	1.0
Example2	-9	772	8	720	2.8
Burstein	-3	778	4	762	0.1
Ext-Burst.	-20	1092	11	1008	1.4
Random	-12	200	11	134	0.4
YK3C	-10	2902	24	2732	20.7
D1	-8	3752	75	3092	37.9
Deutsch	-11	7570	6	7474	117.1

Table 3. Routing results before and after assignment

Since no maximum tolerable crosstalks in the nets were specified for any of the circuits we tested, the maximum tolerable crosstalks for the nets are assigned randomly. Since the channel routing benchmark circuits are large, the mixed ILP problems formulated for these circuits are also large. To solve the mixed ILP problems effectively, the routing channels are broken into several parts from left to right. A mixed ILP problem is then formulated and solved for each part. The results generated by our algorithm for both the switchbox and channel routing problems are summarized in Table 3. Before assignment, *minslack* for all the circuits are negative which means that some crosstalk constraints are violated. After the assignment process, all the crosstalk constraints are satisfied with margins. Moreover, the total crosstalk in the nets is reduced significantly for every circuit.

To compare the assignment approach with the permutation approach presented in [3], the routing solutions generated by the permutation algorithm are summarized in Table 4. As shown in the tables, our algorithm produces better results for all the circuits.

cir. name	init. solution		final solution		
	min-slack	*total-cross*	*min-slack*	*total-cross*	time (min.)
Random	-12	200	7	156	0.1
YK3C	-10	2902	11	2752	6.6
D1	-8	3752	21	3518	5.1
Deutsch	-11	7570	5	7556	0.3

Table 4. Results by the permutation algorithm

6 Conclusion

In this paper, we proposed a new approach to the solution of the gridded channel routing problem and the switchbox routing problem which utilizes existing switchbox and channel routing algorithms and improves upon the routing solution by re-assigning the positions of the wire segments in the initial routing solution. A novel mixed integer linear programming formulation and an effective procedure for reducing the number of variables and constraints in the ILP formulation were then presented. The experimental results are encouraging.

References

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

[2] W. M. Dai, R. Kong, J. Jue, and M. Sato. Rubber Band Routing and Dynamic Data Representation. In *Proc. International Conference on Computer-Aided Design*, pages 52–55, 1990.

[3] T. Gao and C. L. Liu. Minimum Crosstalk Channel Routing. In *Proc. International Conference on Computer-Aided Design*, pages 692–696, 1993.