INFORMS Journal on Computing
Vol. 18, No. 2, Spring 2006, pp. 137–150
ISSN 1091-9856 | EISSN 1526-5528 | 06 | 1802 | 0137

DOI 10.1287/ijoc.1040.0127

Integer Linear Programming Models for Global Routing

Laleh Behjat
Department of Electrical and Computer Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada, laleh@ucalgary.ca

Anthony Vannelli
Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada, vannelli@uwaterloo.ca

William Rosehart
Department of Electrical and Computer Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada, rosehart@ucalgary.ca

Modern integrated circuit design involves the layout of circuits consisting of millions of switching elements or transistors. Due to the sheer complexity of the problem, optimizing the connectivity between transistors is very difficult. The circuit interconnection is the single most important factor in performance criteria such as signal delay, power dissipation, circuit size, and cost. These factors dictate that interconnections, i.e., wires, be made as short as possible. The wire-minimization problem is generally formulated as a sequence of discrete optimization subproblems that are known to be NP-hard. Hence, they can only be solved approximately using meta-heuristics. These methods are computationally expensive and the quality of the solution depends to a great extent on an appropriate choice of starting configuration and modeling techniques. In this paper, new modeling techniques are used to solve the routing problem formulated as an integer programming problem. The main contribution of this paper is a proposed global routing heuristic that combines the wire length, channel congestion, and number of pins in routes to find the best wiring layout of a circuit. By adding information such as channel congestion and the number of pins in each route as well as the wire length, the quality of the solution is improved. In addition, the solutions of the large relaxed linear programming problems are skewed towards a zero-one solution, resulting in faster convergence. The developed LP models in this paper are useful when solving the global routing problem for two reasons; first, the new interior-point algorithms to solve the LP problem are polynomial in time. Second, "near optimal wiring" is obtained in polynomial time without performing randomized rounding.

Key words: VLSI layout; integer programming; linear programming; linear relaxation; global routing
History: Accepted by John W. Chinneck, Guest Editor for the Special Cluster on Operations Research in Electrical and Computer Engineering; received October 2003; revised June 2004; accepted December 2004.

1. Introduction

As integrated circuit (IC) technology moves to deep submicron (DSM) designs, the delay of a circuit as well as the power dissipation and area becomes dominated by the amount of wire and the routing of the circuit. Since the number of transistors on ICs is growing exponentially, new modeling techniques and optimization tools are needed to develop better routings so that wire length and congestion of a circuit are minimized. This paper focuses on developing new modeling and optimization techniques for routing of standard cell-based very large scale integrated (VLSI) circuits.

Modern integrated circuits typically contain millions of elements. The large number of elements make the design of an IC an overwhelming task. In the design of VLSI circuits the *circuit layout* step is where a physical realization of the circuit is obtained from its functional description. A circuit is usually described by a *netlist* consisting of the circuit elements, modules, and their interconnecting wires. These interconnecting wires are called *nets*. Circuit layout determines the geometric coordinates of modules and the course of nets to optimize a given objective while satisfying certain design requirements. The circuit layout problem is NP-hard (Lengauer 1990). Therefore, it is typically divided into three subproblems: partitioning, placement, and routing. A flow chart of these subproblems and some typical feedback paths is given in Figure 1. Each one of these subproblems is briefly discussed.

The *partitioning* subproblem is used to divide the circuit into smaller subcircuits and eventually into switching elements called *cells*. These subcircuits can

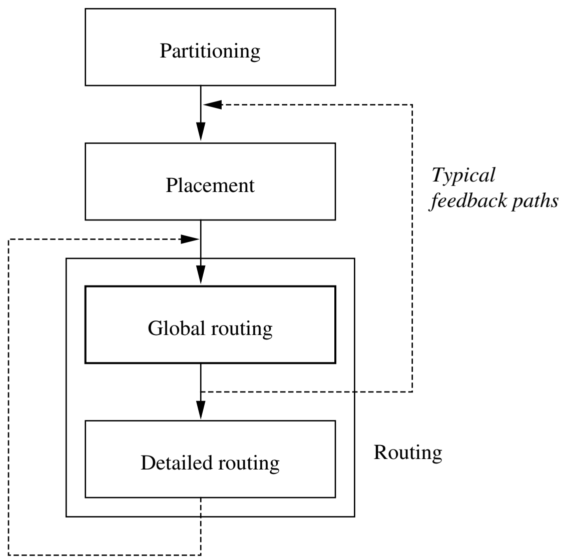

Figure 1 The Flowchart of the Circuit-Layout Subproblems

be designed and implemented as separate circuits and are connected at a later stage of the layout process.

In the next subproblem, *placement*, the position of the cells in each partition is determined while minimizing an estimate of the total wirelength. This problem has been traditionally formulated as an optimization problem with either linear or quadratic functions.

In the *routing* subproblem, the geometric layout of the nets that connect switching elements are determined. The routing subproblem is typically divided into two additional stages: *global routing* and *detailed routing*. In global routing, the approximate course of the wires is determined using a simplified description of the netlist. The solution from the global routing problem is then used in the detailed routing problem to determine the exact course of the wires in the channels (Sherwani 1999).

If, at any of the routing stages, the problem is shown to be infeasible, the designer will return to the placement stage and modify the placement of the problem. This procedure is shown by the feedback loops in Figure 1.

In standard cell design, every circuit is composed of modules, each with the same height and different widths (Lengauer 1990). At the placement stage the modules are placed in rows. The input and output ports of the modules are called *pins* and they provide connection points to other modules.

The global routing problem is NP-hard (Lengauer 1990). Therefore, heuristics capable of producing high-quality routes with little computational effort are required as ICs increase in size. In this paper a global routing heuristic that combines wire length, congestion in the path of the wires, and the number of bends in the routes is proposed. The main contribution of this paper is developing a model that combines different objectives when developing routes. In the first stage of the global routing a good set of routes is constructed for each net. This set is produced based on minimizing the wire length and an approximate estimation of congestion. The global routing problem is then formulated as an integer linear programming (ILP) problem with the objective of minimizing the wire length (Behjat et al. 2002). Congestion estimation and the number of bends are incorporated as penalty functions in the objective function. Two models are developed for the global routing problem. The first model focuses only on minimizing the wire length, while the second model considers minimization of wire length and channel capacity at the same time. Including channel capacity in the objective function tends to result in more uniform use of the channels throughout the circuit and lower channel capacity. The advantage of the proposed technique for global routing is that it can consider routing of all the nets at the same time, therefore providing a global view of the routing problem. Both models are relaxed to a linear programming (LP) problem. The relaxed model is solved using a predictor-corrector interior-point method.

We will show that using congestion in the formulation of the global routing as a penalty function not only improves the routability of the circuit, but also results in the solution of the relaxed LP to be driven either to zero or one. The LP models developed in this paper are solved using a polynomial-time interior-point algorithm. Therefore, by not performing randomized rounding, near-optimal solutions are obtained in polynomial time.

The reminder of the paper is structured as follows: in §2, the global routing problem is considered in detail. In §3, the new modeling techniques used to solve the global routing problem are presented. In §4, interior-point and randomized rounding methods are presented. In §5, numerical results obtained using industry test cases are discussed. Finally, in §6, the conclusions and future work are presented.

2. The Global Routing Problem

The main purpose of global routing is to find an initial course (path) for the wires according to a netlist. Global routing determines which routing channels the nets belong to. After global routing, in the detailed routing stage, each channel is individually routed by a channel router that assigns specific layers and tracks to wires to implement the connection patterns determined at the global routing stage.

Solution methodologies for the global routing problem can be grouped into sequential routing techniques and integer programming (IP) based techniques.

In sequential programming approaches, nets are first ordered according to their routing importance, then each net is separately routed. This method was first introduced by Lee (1961) as the basis for the maze runner algorithms. Various versions of the initial algorithm introduced by Lee have been proposed to improve the speed of sequential routing (Hadlock 1975, Soukup 1978), for example. Several approaches have been proposed to extend these algorithms to multiterminal nets. In approaches for routing multiterminal nets rectilinear Steiner trees are produced for each net (Alpert et al. 2001, Areibi et al. 2001, Chiang et al. 1994, Cong et al. 1998, Cong and Madden 1997, Hu and Sapatnekar 2002, Kastner et al. 2001).

The quality of a sequential global router depends on the ordering of the nets. Usually nets are ordered according to their criticality, half-perimeter wire length, and number of terminals (Sherwani 1999). The criticality of the nets is decided by the importance of the nets. For example, a clock net can determine the performance of the circuit and will be considered important, i.e., it would be routed in the early stages of a sequential approach. After a net has been routed it can block the path of the other nets. Methods such as *rip-up and re-route* algorithms (Dees and Karger 1982) have been developed to improve further the quality of the final solution. Because of the sequential nature of these methods they fail to give a global view of the routing problem.

IP-based global routing methods route all the nets in a region of the circuit concurrently. Routing all the nets concurrently is computationally hard, but this method tends to result in a better solution since no initial ordering of the nets is required. This technique formulates the global routing problem as a 0/1 IP problem. Given a routing graph and a set of Steiner trees for each net, the objective of the IP technique is to select one Steiner tree for each net, such that, the total wire length is minimized and the channel-capacity constraints are not violated.

The time to solve the IP problem increases exponentially with the number of Steiner trees generated in the formulation of the program. Therefore, direct methods to solve the IP problems, such as branch and bound (Floundas 1995), can become time consuming. Alternative approaches such as randomized rounding are used to solve the IP problem. Randomized rounding techniques have been proposed by Raghavan and Thompson (1991) and used by Lengauer and Lungering (2000). In these techniques, an integer solution is obtained from the fractional values of the linear program. The probability of a specific solution to obtain a value of one is equal to its fractional value. A randomized rounding technique uses the fractional outcome of the LP solution as the basis of the appropriate coin tosses to generate an integer solution. A *coin toss experiment* is repeated an appropriate number of times to generate a good solution.

One approach to solve the IP problem for the global routing is using linear relaxations. In this approach, the global routing problem is formulated as an ILP problem (Lengauer 1990). The linear relaxation of the ILP formulation of the problem is solved and an initial partial solution is found. In Hu and Shing (1985), an integer solution for the problem is obtained by choosing the Steiner tree with the highest fractional value for each tree. In Shragowitz and Keel (1987), a linear relaxation of the routing problem is formulated as a multicommodity network flow problem. In Vecchi and Kirkpatrick (1983), a simulated annealing approach for routing the nets is proposed. This method results in high quality routes but requires large running times and is highly dependent on the determination of critical annealing schedules (Aarts and Laarhoven 1985, Lundy 1986). In Vannelli (1991), an adaptation of Karmarkar (1984) interior-point method is used to solve the linear relaxation of the global routing problem. In Karp (1972), randomized rounding (Ng et al. 1987, Raghavan 1988, Raghavan and Thompson 1991) is used to generate an integer solution from the solution of the LP problem. Local search heuristics, sequential routing, genetic routing, and randomized procedures are employed in Lengauer and Lungering (2000) to produce global routes.

Sequential routing techniques can produce good global routes in reasonable time, but because of the sequential nature of these techniques, they tend to produce poor globally optimum results and cannot consider congestion in the formulation of the problem. On the other hand, IP techniques tend to produce global solutions, but the time to solve the integer programs tends to be very high.

In this paper, the global routing problem is formulated as an IP problem, and a linear relaxation is used to solve the IP problem. New modeling techniques are employed to avoid the need for randomized rounding while obtaining high quality results.

3. New Formulation for Global Routing

The approach proposed and implemented in this paper uses information such as congestion and via count in addition to wire length in both the problem formulation and the rectilinear spanning tree (RST) construction. The flowchart in Figure 2 shows the main steps in formulating and solving the global routing problem when using the proposed IP-based approach.

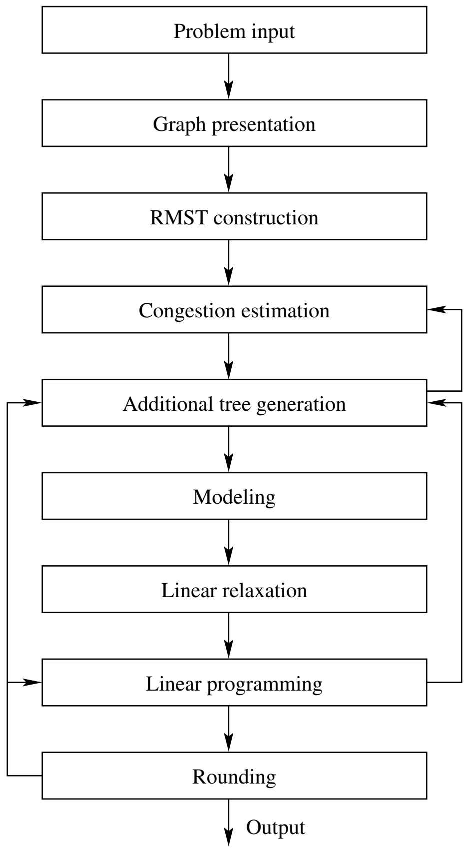

Figure 2 **Flowchart of the Proposed Global Routing Formulation**

Some of the important steps presented in this section for solving the global routing problem are:

RMST Construction. An approach to construct rectilinear minimum spanning trees (RMSTs) for the nets.

Congestion Estimation. A new approach to develop a congestion estimation technique to predict potentially congested areas in the routing graph at the RMST construction stage.

Additional Trees. Based on the proposed congestion estimation technique, a number of additional RMSTs are constructed. This method also incorporates important factors such as the number of bends and congestion, as well as wire length in constructing the RMSTs.

New ILP Formulations. Development of new models for the ILP formulation of the global routing problem. These models directly incorporate wire length, maximum channel capacity, congestion estimation and the number of vias in the objective of the global routing formulation. The most important contribution of the new ILP models is that the LP relaxations are solved efficiently without using rounding techniques.

3.1. Rectilinear Spanning Tree Construction

One of the most important steps in global routing is producing a set of admissible routes for each net. Often this set is produced using RMSTs that connect the cells of the net. Since the number of all the possible trees for each net can be substantial, considering all possible trees for a net results in a large number of variables and constraints in the optimization problem. To reduce the initial size of the problem, only RMSTs are considered for the nets.

The number of minimum-bend trees generated for each net grows exponentially with the number of the cells in the net. For a net consisting of r cells, the maximum number of minimum-bend trees is

$$r^{r-2}2^{r-1}.$$

From (1) it can be calculated that for 2-cell, 3-cell, 4-cell, and 5-cell nets there are at most 2, 12, 128, and 2,000 possible minimum-bend trees, respectively. Another disadvantage of producing minimum-bend trees (trees with the minimum number of bends in them) is that for the nets that contain only two cells, the maximum number of trees produced is two.

The Hanan (1966) method to produce a set of trees for the nets produces too many trees for the nets that have four or more cells. Therefore, the number of variables in the optimization problem can become very large. On the other hand, the number of trees for nets with two or three cells is very limited, but these nets constitute a high percentage of the nets. Therefore, the set of produced trees can be an infeasible set. To overcome the above shortcomings, a new method is proposed and implemented. First, only minimum Steiner trees are constructed for all the nets. An estimate of congestion in the path of each tree is calculated. Additional trees are produced only for the nets that pass through the congested areas or the congested edges. These trees have additional length or bends but do not pass through the congested edges.

The main contribution of the proposed technique is to construct a set of trees that contains the fewest trees that can result in a feasible solution.

3.2. Congestion-Estimation Technique

After producing an initial set of RMSTs for each net, the global routing problem can be formulated as an ILP problem. Solving the ILP problem is the most time-consuming part of the global routing problem. Therefore, it is very important to make sure that the global routing formulation will result in a feasible solution, i.e., that the trees produced for the nets constitute a feasible set. One of the main set of constraints in routing a circuit are the channel-capacity constraints which state that the number of wires passing through a channel cannot be more than the maximum channel capacity. This maximum channel capacity is

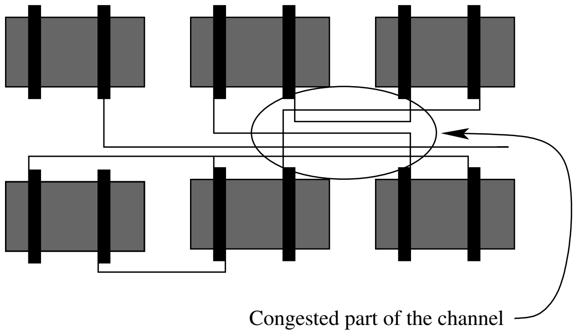

Figure 3 Illustration of a Channel with a Congested Area

set by the circuit specifications. A channel is considered to be congested if the number of the wires passing through it is close to, or exceeds, the maximum channel capacity. Figure 3 illustrates a channel with a congested area. If the number of wires passing through a channel exceeds the maximum channel capacity, the routing of a circuit becomes infeasible.

The proposed congestion estimation method can be outlined as follows. First, a method that estimates the congestion in each edge is applied. The term *edge* refers to a part of a routing channel. This estimation is then used to produce additional RSTs for the nets that pass through the congested areas. This method avoids producing a large number of trees for each net, therefore, minimizing the number of variables in the optimization problem. In addition, it avoids congestion in the edges of the routing graph. Furthermore, when formulating the ILP problem the congestion estimations are used as penalty terms for the trees that pass through the congested areas. These penalties are directly incorporated in the objective of the global routing problem. Congestion estimation is used make sure that the problem will have a feasible solution. In the rest of this section the congestion-estimation technique is discussed in further detail.

3.2.1. Determining Routing Demand. After constructing a set of trees for each net, a binary variable y_j is assigned to each tree. The variable y_j is equal to one if it is used in the final layout and is equal to zero otherwise, i.e.,

$$y_j = \begin{cases} 1 & \text{if tree } j \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$$

At this stage, the number of trees produced for each net and the edges through which each tree passes are known. Let N_k represent the set of all the trees produced for net k. The probability of $y_j \in N_k$ to be chosen in the final result is approximated by

$$p(y_j) = \frac{1}{\text{number of trees produced for net } k}.$$

For example, if net k has two trees, the probability of each tree being chosen is equal to 1/2. In the early stages of tree construction the trees produced for a net have equal lengths and number of vias, so the above probability measure is a reasonable measure to represent the probability of a tree's being chosen. In addition, since at the first stage of the route construction no other information such as congestion in the path of the trees exists, this measure is considered adequate. This probability measure can be further improved by adding congestion in the calculations in the subsequent iterations.

For each edge e_i, a new measure called the *routing demand*, $r(e_i)$, is presented. The routing demand estimates the number of trees that might pass through an edge and is equal to the summation of the probabilities of the trees that pass through edge e_i:

$$r(e_i) = \sum_{j=1}^{t} a_{ij} p(y_j),$$

where t is the total number of trees and a_{ij} is a binary constant and is equal to one if tree j passes through edge i and is zero otherwise, i.e.,

$$a_{ij} = \begin{cases} 1 & \text{if tree } j \text{ passes through edge } i, \\ 0 & \text{otherwise.} \end{cases}$$

In addition, an upper estimate $r_u(e_i)$ and a lower estimate $r_l(e_i)$ of the routing demand for each edge are calculated as follows:

$$r_u(e_i) = \sum_{j=1}^{t} a_{ij},$$

$$r_l(e_i) = \sum_{j=1}^{t} a_{ij} \lfloor p(y_j) \rfloor,$$

where $\lfloor p(y_j) \rfloor$ represents the floor function of $p(y_j)$ and is equal to one only if $p(y_j) = 1$. The upper estimate $r_u(e_i)$ indicates the maximum number of trees that can pass through e_i, i.e., the worst case. The lower estimate $r_l(e_i)$ indicates the number of trees that will definitely pass through e_i. The lower estimate $r_l(e_i)$ is equal to the sum of the nets that have only one tree and pass through edge e_i.

3.2.2. Congestion Estimation Functions. To find probable congested edges, two congestion functions are developed. The first function f_{cong} compares the maximum channel capacity and the routing demand of each edge:

$$f_{cong} = c - r(e_i),$$

where c is the maximum channel capacity. The function f_{cong} calculates the difference between the maximum capacity and the estimation of congestion.

A smaller value for f_{cong} means a higher possibility that the edge will become congested. If $f_{cong} < 0$, then the congestion estimation is higher than the maximum channel capacity and the problem will become infeasible. At the first iteration of the additional tree-construction phase, a good margin between the channel congestion and the maximum channel capacity is desirable. It is proposed that if $f_{cong}(e_i) < c/10$ then e_i is considered congested. Therefore, it is proposed that if any edge has more than 90% of its capacity estimated to be occupied, then that edge will be considered congested. Numerical simulations allowing occupancy less than 90% of the total capacity resulted in feasible solutions for all the test cases used. The edges that are marked congested are not used in the next iteration of the tree construction. In areas that have too many congested edges, no other possible trees can be constructed and the congestion in these edges cannot be reduced. It is proposed that the accepted gap between the congestion estimation and the maximum channel capacity should be reduced to allow additional trees to be introduced in highly congested areas. The following tolerance function is introduced for evaluating the congestion function:

$$f_{cong} < \theta(c, iteration) = \frac{c}{10\ iteration}, \quad iteration > 0.$$

where θ is the tolerance function and *iteration* indicates the iteration number for building additional trees. This tolerance function decreases with each iteration of the tree construction, allowing more trees to be constructed in the congested areas.

The function f_{cong} measures the difference between the maximum channel capacity and the channel congestion estimation but does not consider other factors such as the number of trees passing through an edge or the number of trees that will definitely pass through an edge. Therefore, another function f_{util} is proposed that measures the remaining utility of the edge and is expressed as

$$f_{util} = \frac{c - r_l(e_i)}{r_u(e_i) - r_l(e_i)}.$$

The numerator $(c - r_l(e_i))$ of f_{util} is the number of available tracks in a channel after routing the nets that have only one tree. The denominator $r_u(e_i) - r_l(e_i)$ is the number of trees that can pass through the remaining tracks. The f_{util} function indicates the number of tracks available in each edge. If f_{util} is too small it means that a high number of trees might pass through the few remaining tracks, so the probability of the edge becoming congested is high. The tolerance function proposed for this function is the same as the tolerance function θ. This tolerance will enforce a 10% remaining capacity to be available for the trees with probabilities less than one at the first iteration.

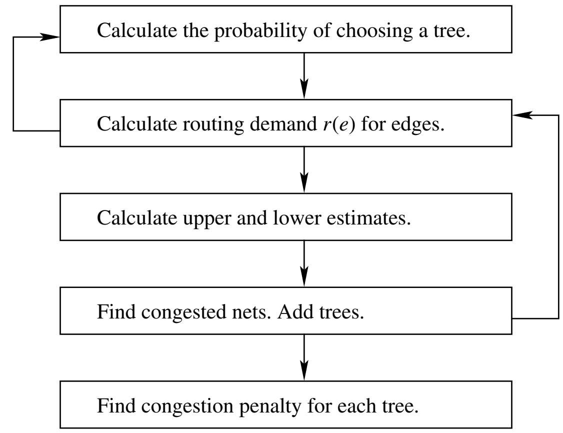

Figure 4 **Proposed Congestion Estimation Outline**

As the number of iterations becomes larger, the tolerance becomes too small to accommodate the large number of trees with low probability produced in the higher iterations.

For each edge, f_{cong} and f_{util} are calculated. If $f_{cong} < \theta$ or $f_{util} < \theta$, then additional trees are added to the nets passing through the congested edges. This procedure is repeated until the congestion requirements for all the edges are satisfied or an iteration limit is reached. After finding the congested edges, additional trees are made for these edges and the congestion estimate along with the upper and the lower estimates are calculated. An outline for the congestion estimation technique is shown in Figure 4.

3.2.3. Congestion Penalty. After a suitable set of trees has been produced for all the nets, the congestion penalty associated with a tree, w_c, is calculated. This congestion penalty is equal to the sum of the routing demands $r(\cdot)$ of the edges through which the tree passes

$$w_{c_j} = \sum r(e_i), \quad e_i \in y_j.$$

This proposed congestion penalty for the trees is used when formulating the problem as an ILP in §3.3.

The congestion estimate, along with its upper and lower estimations, are used to identify edges that are likely to become congested. The nets passing through these edges are identified and additional trees are constructed for these nets. These trees have additional length or number of pins but cannot pass through congested areas.

3.3. ILP Formulation

The constraints of the global routing problem, and hence the constraints of the ILP problem, are as follows:

- Each variable y_j can be either zero or one:

$$y_j \in \{0, 1\}, \quad j = 1, 2, \ldots, t,$$

where t is the total number of trees.

• From all the different ways to connect a net, only one way can be chosen. Therefore, the following constraint is introduced for each net:

$$\sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n,$$

where N_k is the set of all the trees constructed for net k and n is the total number of nets.

• Each edge has a maximum capacity, so only a certain number of trees can pass through each edge. The constraints associated with the maximum capacity are referred to as *maximum capacity* constraints and are formulated as follows: 0/1 matrix, $\mathbf{A}_d = [a_{ij}] \in \Re^{p\times t}$, or the *adjacency matrix*, that represents the edge-tree connections is built. Each row of $\mathbf{A}_d$ corresponds to an edge in the graph, and each column corresponds to a tree. If tree j passes through edge i, $a_{ij} = 1$; otherwise, $a_{ij} = 0$. The elements of $\mathbf{A}_d$ are represented as

$$a_{ij} = \begin{cases} 1 & \text{if tree } j \text{ passes through edge } i, \\ 0 & \text{otherwise.} \end{cases}$$

The sum of all the trees passing through an edge should be less than the capacity of the edge. Therefore, the capacity constraints can be represented as

$$\sum_{j=1}^{t} a_{ij} y_j \le c, \quad i = 1, 2, \ldots, p,$$

where p is the total number of edges and c is the maximum allowable capacity of the edge i set by the circuit specifications and is fixed.

The global routing problem with the objective of *minimizing the total wire length* (MTWL) (Vannelli 1991) is formulated as:

$$\begin{aligned} \min \quad & \sum_{j=1}^{t} w_{l_j} y_j \\ \text{s.t.} \quad & \sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n, \\ & \sum_{j=1}^{t} a_{ij} y_j \le c, \quad i = 1, 2, \ldots, p, \\ & y_j \in \{0, 1\}, \quad j = 1, 2, \ldots, t, \end{aligned} \tag{1}$$

where w_{l_j} represents the weight associated with the length of tree y_j. This length is calculated as the normalized length of tree j with respect to the rest of the trees in net k:

$$w_{l_j} = \frac{\text{Length of tree } j}{\text{Max length of trees constructed for net } k}.$$

The MTWL formulation given in (1) has the major shortcoming that it takes into consideration only the wire length of the produced trees and does not consider other important factors such as congestion or number of vias. Therefore, the problem can become infeasible or, in the final results, parts of a channel can become congested, which in turn can result in *hotspots* and unroutability. In addition, the number of vias or bends play an important role in the routability and timing of the nets. Minimizing the number of vias will reduce the number of metal contacts in the routing, thus improving the routability of the circuit. In addition, if the channel capacities and congestion are reduced, unwanted effects such as hotspots and capacitive coupling are reduced. Hotspots refer to areas of a circuit where the number of wires is large. Because of the resistance of the wires, heat dissipates through these wires. If a part of a channel is congested then the heat dissipation can become too much, which in turn will increase the resistance of the wires and make a feed forward loop. The high temperature of a part of a circuit can result in burning the transistors of the circuit. RC coupling happens when two wires close to one another are both carrying signals. The signal in one wire can be changed from a logical 0/1 to 1/0 because of a change in the signal in another wire. To reduce RC coupling the distance between the wires are increased. If a channel is congested this distance has to be reduced and the reliability of a circuit will be reduced. Because of these factors, reducing congestion in a circuit is crucial issue.

3.3.1. Penalty Model. In the proposed formulation there are two other important measures: the congestion in the path of the trees, and the number of bends in the trees are added as penalty terms to the objective of the global routing problem. The proposed global routing problem formulation is

$$\begin{aligned} \min \quad & \sum_{j=1}^{t} (\beta_l w_{l_j}) y_j + \sum_{j=1}^{t} (\beta_v w_{v_j} + \beta_c w_{c_j}) y_j \\ \text{s.t.} \quad & \sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n, \\ & \sum_{j=1}^{t} a_{ij} y_j \le c, \quad i = 1, 2, \ldots, p, \\ & y_j \in \{0, 1\}, \quad j = 1, 2, \ldots, t, \end{aligned} \tag{2}$$

where w_{v_j} and w_{c_j} are the weighting factors associated with the number of bends and congestion in the path of tree j. The scalars β_l, β_v, and β_c are used to change the emphasis of the respective weighting factors. The weighting factor associated with the number of bends, w_{v_j} in tree j, belonging to net k and is calculated as

$$w_{v_j} = \frac{\text{Number of bends in tree } j}{\text{Max number of bends in trees produced for net } k}.$$

The congestion measure w_{c_j} is based on the probabilistic measure of congestion in the path of tree j, as calculated in §3.2.3.

In the final layout, the circuit is fabricated one layer at a time. Each layer must be aligned perfectly to fabricate the features such as vias that connects the different layers. Therefore, minimizing the number of vias will improve the routability of the circuit (Sherwani 1999). Including congestion in the objective will result in fewer tracks, reduced wire length, and fewer hotspots. Simulation results on the MCNC benchmarks (Kozminski 1991), given in §5, indicate that including the congestion as a penalty function in the objective of the global routing problem results in fewer tracks being used in the final routing. MCNC benchmarks are a set of circuits designed to verify the VLSI circuit problems and are widely used.

3.3.2. Combined Objective Function. The objective of the optimization problem presented in (2) does not consider minimization of channel capacities. To overcome the shortcomings of this model, a second formulation that is a combination of the two models is proposed. In this model the channel capacity becomes an integer variable with a lower bound of zero and upper bound of the set maximum channel capacity. The objective of the combined penalty and minimization of maximum capacity models is given in (2). The new formulation is

$$
\begin{aligned}
\min\ & \sum_{j=1}^{t} w_j y_j + \beta_z z \\
\text{s.t.}\ & \sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n, \\
& \sum_{j=1}^{t} a_{ij} y_j - z \le 0, \quad i = 1, 2, \ldots, p, \\
& y_j \in \{0, 1\}, \quad j = 1, 2, \ldots, t, \\
& z \in \{0, \ldots, c\},
\end{aligned} \tag{3}
$$

where $w_j = \beta_l w_{l_j} + \beta_v w_{v_j} + \beta_c w_{c_j}$. The variable z represents the variable assigned to the capacity of channels. The scalar β_z is the weighting factor associated with the maximum channel capacity. This model minimizes the channel capacity and the combination of wire length, number of vias, and congestion at the same time.

3.4. Linear Relaxation of the ILP Model

Because of the large sizes of today's VLSI problems, solving the ILP formulation using traditional solution methodologies, e.g., branch and bound, tend to be impractical. Therefore, the ILP is typically relaxed and solved as an LP problem followed by rounding heuristics to obtain an integer solution. This relaxation is performed by replacing the integer constraints imposed on the variables with linear boundary constraints. To reduce the number of constraints, the following observation is made: if each tree is assumed to be nonnegative, then the inequality constraint $0 \le y \le 1$ is redundant and this condition is enforced by the first set of constraints, enforcing that only one tree can be chosen for each net.

The global routing problem can be written as

$$
\begin{aligned}
\min\ & \sum_{j=1}^{t} w_j y_j \\
\text{s.t.}\ & \sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n, \\
& \sum_{i=1}^{p} a_{ij} y_i \le c, \quad i = 1, 2, \ldots, p, \\
& 0 \le y_j, \quad j = 1, 2, \ldots, t.
\end{aligned} \tag{4}
$$

or

$$
\begin{aligned}
\min\ & \sum_{j=1}^{t} w_j y_j + \beta_z z \\
\text{s.t.}\ & \sum_{y_j \in N_k} y_j = 1, \quad k = 1, 2, \ldots, n, \\
& \sum_{i=1}^{t} a_{ij} y_i - z \le 0, \quad i = 1, 2, \ldots, p, \\
& 0 \le y_j, \quad j = 1, 2, \ldots, t, \\
& 0 \le z \le c,
\end{aligned} \tag{5}
$$

or in matrix format for (4)

$$
\begin{aligned}
\min\ & \mathbf{w}_1^T \mathbf{y}, \\
\text{s.t.}\ & \mathbf{A}_1 \mathbf{y} = \mathbf{1}, \\
& \mathbf{A}_2 \mathbf{y} \le \mathbf{c}, \\
& -\mathbf{y} \le \mathbf{0},
\end{aligned} \tag{6}
$$

where $\mathbf{w}_1 \in \Re^t$ is the vector of weights placed on the different trees. The vector $\mathbf{y} \in \Re^t$ is the vector of unknown variables, y_j. The constraint matrices $\mathbf{A}_1 \in \Re^{n \times t}$ and $\mathbf{A}_2 \in \Re^{p \times t}$ represent the constraints. The right-hand side vectors are $\mathbf{1} \in \Re^p$, where $\mathbf{1} = [1, 1, \ldots, 1]^T$, and $\mathbf{c} \in \Re^p$, $\mathbf{c} = [c, \ldots, c]^T$, where c is the maximum capacity of the edges. The matrix format of (5) can be written as

$$
\begin{aligned}
\min\ & \mathbf{w}_1^T \mathbf{y} + \beta_z z, \\
\text{s.t.}\ & \mathbf{A}_1 \mathbf{y} = \mathbf{1}, \\
& \mathbf{A}_2 \mathbf{y} - \mathbf{1} z \le \mathbf{0}, \\
& -\mathbf{y} \le \mathbf{0}, \\
& 0 \le z \le c.
\end{aligned} \tag{7}
$$

The equality constraints in (6) and (7) can be transformed to inequality constraints without changing the nature of the problem (Vannelli 1991). Therefore, the problem can be represented as

$$\begin{aligned} \min \quad & \mathbf{w}^T\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\ & \mathbf{0} \le \mathbf{x}, \end{aligned} \tag{8}$$

where for (6)

$$\begin{aligned} \mathbf{x} &= \mathbf{y} \\ \mathbf{w} &= \mathbf{w}_1 \end{aligned}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{1} \\ \mathbf{c} \end{bmatrix},$$

and for (7) we have

$$\begin{aligned} \mathbf{x} &= [\mathbf{y}; z] \\ \mathbf{w} &= [\mathbf{w}_1; \beta_z] \end{aligned}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{1} \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ c \end{bmatrix}.$$

Several different methods have been introduced to solve the relaxed LP problem. In §4, a primal-dual interior-point (PD-IP) method and a predictor-corrector interior-point (PC-IP) method to solve general nonlinear optimization problems with linear constraints are derived. The latter interior-point method is used to solve all the LP problems considered.

4. Interior-Point Optimization Techniques

The relaxed global routing program formulated in §3 can be solved in polynomial time using an interior-point method. Because of the new modeling techniques, the solution of the LP model yields 0/1 answers. Therefore, the IP model used for the global routing problem is solved in polynomial time.

The general formulation of the global routing problem can be written as

$$\begin{aligned} \min \quad & \mathbf{w}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0}, \end{aligned} \tag{9}$$

where $\mathbf{x} \in \Re^q$ is a vector of unknown variables, usually referred to as the *primal* variables, and q is the total number of variables. Matrix $\mathbf{A} \in \Re^{m\times q}$ is the constraint matrix, consisting of constraints on the nets and capacity constraints. The total number of constraints is denoted by m.

The first step in the derivation of an interior-point algorithm to solve (9) is to add nonnegative slack variables $\mathbf{s}_1 \in \Re^m_+$ and $\mathbf{s}_2 \in \Re^q_+$ to transform all inequality constraints to equality constraints (Andersen et al. 1996, Vanderbei 1998). The optimization problem then has the form

$$\begin{aligned} \min \quad & \mathbf{w}^T\mathbf{x} \\ \text{s.t.} \quad & -\mathbf{A}\mathbf{x} - \mathbf{s}_1 + \mathbf{b} = \mathbf{0} \\ & \mathbf{x} - \mathbf{s}_2 = \mathbf{0} \\ & \mathbf{s}_1, \mathbf{s}_2 \ge \mathbf{0}. \end{aligned} \tag{10}$$

The nonnegativity conditions $\mathbf{s}_1, \mathbf{s}_2 \ge \mathbf{0}$ are incorporated into the objective function of (10) using logarithmic barrier terms as follows

$$\begin{aligned} \min \quad & \mathbf{w}^T\mathbf{x} - \mu^k \sum_{i=1}^{m} \log(s_{1_i}) - \mu^k \sum_{i=1}^{q} \log(s_{2_i}) \\ \text{s.t.} \quad & -\mathbf{A}\mathbf{x} - \mathbf{s}_1 + \mathbf{b} = \mathbf{0} \\ & \mathbf{x} - \mathbf{s}_2 > \mathbf{0}, \end{aligned} \tag{11}$$

where $\mu^k > 0$ is a monotonically decreasing series of *barrier parameters*, and s_{1_i} and s_{2_i} represent the ith element of $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. The logarithmic terms impose the strict positivity conditions on the nonnegative variables. To solve the equality-constrained problem stated in (11), a Lagrange-Newton method is applied: first, the Lagrangian function associated with (11) is formed, then an approximate local minimum of the Lagrangian function based on the Karush-Kuhn-Tucker (KKT) conditions is obtained (Bertsekas 1995).

To simplify the presentation of the Lagrangian function, the vector $\mathbf{r} := (\mathbf{s}_1^T, \mathbf{s}_2^T, \boldsymbol{\lambda}^T, \boldsymbol{\pi}^T, \mathbf{x}^T)^T \in \Re^m \times \Re^q \times \Re^m \times \Re^q \times \Re^q$ is introduced where $\boldsymbol{\lambda} \in \Re^m$ and $\boldsymbol{\pi} \in \Re^q$ are Lagrangian multipliers, usually referred to as the *dual* variables. The Lagrangian function $L_\mu(\mathbf{r})$ associated with (11) is

$$\begin{aligned} L_\mu(\mathbf{r}) := {} & \mathbf{w}^T\mathbf{x} - \mu^k \sum_{i=1}^{m} \log(s_{1_i}) - \mu^k \sum_{i=1}^{q} \log(s_{2_i}) \\ & - \boldsymbol{\lambda}^T(-\mathbf{A}\mathbf{x} - \mathbf{s}_1 + \mathbf{b}) - \boldsymbol{\pi}^T(\mathbf{x} - \mathbf{s}_2). \end{aligned}$$

To find a local minimum of the Lagrangian function $L_\mu(\mathbf{r})$, a stationary point of the Lagrangian function is found. This point satisfies the following KKT first-order necessary conditions

$$\nabla_{\mathbf{s}_1} L_\mu = \boldsymbol{\lambda} - \mu^k \mathbf{S}_1^{-1}\mathbf{1}_m = \mathbf{0} \tag{12}$$

$$\nabla_{\mathbf{s}_2} L_\mu = \boldsymbol{\pi} - \mu^k \mathbf{S}_2^{-1}\mathbf{1}_q = \mathbf{0} \tag{13}$$

$$\nabla_{\boldsymbol{\lambda}} L_\mu = \mathbf{A}\mathbf{x} + \mathbf{s} - \mathbf{b} = \mathbf{0} \tag{14}$$

$$\nabla_{\boldsymbol{\pi}} L_\mu = -\mathbf{x} + \mathbf{s}_2 = \mathbf{0} \tag{15}$$

$$\nabla_{\mathbf{x}} L_\mu = \mathbf{w} + \mathbf{A}^T\boldsymbol{\lambda} - \boldsymbol{\pi} = \mathbf{0}, \tag{16}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are diagonal matrices with their diagonal elements equal to the elements of $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. The vector $\mathbf{1} := [1, 1, \ldots, 1]^T$ stands for a vector of ones with its dimension indicated by the subscript. The KKT conditions can be interpreted as follows (Bertsekas 1995): The first two terms are the μ-*complementary* conditions. The third and fourth terms, along with $\mathbf{s}_1, \mathbf{s}_2 \geq \mathbf{0}$, ensure primal feasibility. The last term, along with $\boldsymbol{\lambda}, \boldsymbol{\pi} \geq \mathbf{0}$, ensures dual feasibility.

Assuming that an initial solution satisfying the strict positivity constraints (interior point) $(\mathbf{s}_1, \mathbf{s}_2, \boldsymbol{\lambda}, \boldsymbol{\pi}, \mathbf{x}) > \mathbf{0}$ is provided, one step of Newton's method is applied to find a solution that is closer to solving the first-order optimality conditions. The Newton step, $\Delta\mathbf{r} := (\Delta\mathbf{s}_1^T, \Delta\mathbf{s}_2^T, \Delta\boldsymbol{\lambda}^T, \Delta\boldsymbol{\pi}^T, \Delta\mathbf{x}^T)^T$, is computed by solving the nonsymmetric indefinite system of equations

$$\nabla_{\mathbf{r}}^2 L_\mu(\mathbf{r})\Delta\mathbf{r} = -\nabla_{\mathbf{r}} L(\mathbf{r}). \quad (17)$$

A suitable step size α is calculated by performing a line search along the Newton direction and the new iteration is calculated by

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \alpha^k \Delta\mathbf{r},$$

where the scalar $\alpha^k \in (0, 1]$ is used to limit the step size and ensure that the nonnegativity conditions are satisfied at each iteration. Different schemes to obtain an appropriate step length are reviewed in (Vanderbei and Shanno 1997).

At the end of each iteration, the updated variables are tested to determine if a local minimum has been obtained. At the global minimum, primal feasibility conditions, dual feasibility conditions, and the complementary conditions should be satisfied. The program terminates when the convergence criteria are satisfied; otherwise, k is incremented, a new barrier parameter $\mu^{k+1} < \mu^k$ is computed and another Newton direction is determined. Different schemes exist for reducing μ and scaling the Newton step.

The primal-dual interior-point methods presented in this section can solve large problems, but long running times and numerical difficulties in some cases led to the development of predictor-corrector interior-point methods (Vanderbei 1998).

Mehrotra (1992) proposed a predictor-corrector primal-dual interior-point method that can be computationally more efficient than the original primal-dual interior-point methods. The main idea behind the predictor-corrector methods is to incorporate higher-order information to improve the accuracy of the Newton step for solving the KKT conditions.

To implement a predictor-corrector variant of the primal-dual algorithm, a new point $(\mathbf{r} + \Delta\mathbf{r})$ is defined as being on the central path, i.e., the exact solution to (12)–(16) for μ^k. Therefore, the Newton step stated in (17) that was used to estimate an approximate solution from the current point $\mathbf{r} := (\mathbf{s}_1, \mathbf{s}_2, \boldsymbol{\lambda}, \boldsymbol{\pi}, \mathbf{x})$ is not used. (12)–(16) are rewritten in matrix format by substituting $\mathbf{r} + \Delta\mathbf{r}$ in place of $\mathbf{r}$

$$\begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} & \mathbf{S}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi} & \mathbf{0} & \mathbf{S}_2 & \mathbf{0} \\ \mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_q & \mathbf{0} & \mathbf{0} & -\mathbf{I}_q \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & -\mathbf{I}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{s}_1 \\ \Delta\mathbf{s}_2 \\ \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\pi} \\ \Delta\mathbf{x} \end{bmatrix} = -\begin{bmatrix} \bar{\boldsymbol{\beta}}_1 \\ \bar{\boldsymbol{\beta}}_2 \\ \bar{\boldsymbol{\beta}}_3 \\ \bar{\boldsymbol{\beta}}_4 \\ \bar{\boldsymbol{\beta}}_5 \end{bmatrix} + \begin{bmatrix} \mu^k \mathbf{1}_m \\ \mu^k \mathbf{1}_q \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Delta\mathbf{S}_1 \Delta\boldsymbol{\lambda} \\ \Delta\mathbf{S}_2 \Delta\boldsymbol{\pi} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (18)$$

where $\Delta\mathbf{S}_1$ and $\Delta\mathbf{S}_2$ are diagonal matrices containing the elements of $\Delta\mathbf{s}_1$ and $\Delta\mathbf{s}_2$, respectively. $\boldsymbol{\Lambda}$ and $\boldsymbol{\Pi}$ are diagonal matrices with their diagonal elements equal to the components of $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$, respectively, and

$$\begin{bmatrix} \bar{\boldsymbol{\beta}}_1 \\ \bar{\boldsymbol{\beta}}_2 \\ \bar{\boldsymbol{\beta}}_3 \\ \bar{\boldsymbol{\beta}}_4 \\ \bar{\boldsymbol{\beta}}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1 \boldsymbol{\lambda} \\ \mathbf{S}_2 \boldsymbol{\pi} \\ \mathbf{A}\mathbf{x} + \mathbf{s}_1 - \mathbf{b} \\ -\mathbf{x} + \mathbf{s}_2 \\ \mathbf{w} + \mathbf{A}^T \boldsymbol{\lambda} - \boldsymbol{\pi} \end{bmatrix}.$$

The column vector $\Delta\mathbf{r} = (\Delta\mathbf{s}_1^T, \Delta\mathbf{s}_2^T, \Delta\boldsymbol{\lambda}^T, \Delta\boldsymbol{\pi}^T, \Delta\mathbf{x}^T)^T$ in $\nabla_{\mathbf{r}}^2 L_\mu(\mathbf{r})\Delta\mathbf{r} = \nabla_{\mathbf{r}} L(\mathbf{r})$ consists of three components corresponding to the three vectors on the right-hand side of (18)

$$\Delta\mathbf{r} = \Delta\mathbf{r}_{aff} + \Delta\mathbf{r}_{cen} + \Delta\mathbf{r}_{cor}.$$

The first component $\Delta\mathbf{r}_{aff}$ is responsible for reducing the complementary gap $\boldsymbol{\lambda}^T\mathbf{S}_1 + \boldsymbol{\pi}^T\mathbf{S}_2$. The second component $\Delta\mathbf{r}_{cen}$ is used to bring the solution back to the center of the feasible region. Finally, the last component $\Delta\mathbf{r}_{cor}$ is a corrector direction used to compensate for some of the nonlinear terms.

An overview of the predictor-corrector method is as follows: first, the linear system (18) is solved with only $\bar{\boldsymbol{\beta}}$ on the right-hand side, to get an intermediate solution $\Delta\mathbf{r}_{aff} := (\Delta\hat{\mathbf{s}}_1, \Delta\hat{\mathbf{s}}_2, \Delta\hat{\boldsymbol{\lambda}}, \Delta\hat{\boldsymbol{\pi}}, \Delta\hat{\mathbf{x}})$. This solution is used to estimate a value for μ for the corrector step and approximate $\Delta\mathbf{S}_1$, $\Delta\boldsymbol{\lambda}$, $\Delta\mathbf{S}_2$, and $\Delta\boldsymbol{\pi}$. Then, (18) is solved again, substituting $\Delta\mathbf{r}_{aff}$ and the estimate of μ and solving (18) again by using the last two vectors on the right-hand side to get an update $\Delta\mathbf{r}$.

Full details of the implementation of a predictor-corrector method can be found in Andersen et al. (1996) and Vanderbei and Shanno (1997). In this paper, the predictor-corrector method developed in Vanderbei (1998) has been used.

Table 1 Statistics of MCNC Benchmarks Used as Test Cases

Circuit	No. of cells	No. of pads	No. of nets	No. of rows	Total wire length (m)
Fract	125	24	147	6	0.034
Primary 1	752	81	876	17	0.823
Primary 2	3,014	107	3,136	24	4.088
Struct	1,888	64	1,920	2	0.435
Industry 1	2,271	814	2,478	25	1.551
Biomed	6,417	97	5,742	44	2.191
Industry 2	12,142	495	13,419	69	18.731
Industry 3	15,059	374	21,938	52	52.245
Avg. small	21,854	64	22,124	80	9.742
Avg. large	25,114	64	25,384	86	10.979

5. Numerical Results

The global routing algorithm was implemented on MCNC benchmarks circuits given in Table 1. The MCNC benchmarks were used in this research because of the availability of the placement results to the authors. The MCNC benchmark circuits are sample circuits used verify placement and routing algorithms. These benchmarks were planned for 1.2μ technology and have been used extensively in place and route packages. These benchmark circuits were part of larger circuits used in industrial circuits and have been modified for testing place and route algorithms. These benchmarks can be obtained at Madden (2004). The functionality of each of these circuits are as follows (Kozminski 2004):

Fract: This circuit is a fractional multiplier with 125 logic elements.

Primary 1: This is a medium-size peripheral interface chip. It has a fairly random mix of sequential and combinational logic and is constructed based on a one layer of metal technology.

Primary 2: This circuit is a 16-bit microprocessor. It includes a sizeable register stack and some large pieces of decode logic. It also uses only one layer of metal.

Struct: This circuit implements a 16-bit multiplier. The circuit is very regular and contains many repeating configurations of cells.

Industry 1: No data regarding the functionality of this circuit exist.

Biomed: This benchmark circuit is the only benchmark circuit with more cells than net. It also has a very high number of pins.

Industry 2: No data regarding the functionality of this circuit exist.

Industry 3: No data regarding the functionality of this circuit exist.

Avg. small: To our knowledge no data regarding the functionality of this circuit exist.

Avg. large: To our knowledge no data regarding the functionality of this circuit exist.

All the experiments were done on a Sun Ultra 10 workstation. A summary of the placement results was obtained using the attractor-repeller placer (ARP) (Etawil et al. 1999).

The names and sizes in terms of number of cells, input/output pads, and nets for the MCNC benchmark test circuits used to analyze the characteristics of the proposed formulations and congestion estimation models are given in Table 1.

The placement results obtained from the ARP placement package (Etawil et al. 1999), along with the number of rows for the test cases, are given in Table 1.

5.1. Tree Construction and Congestion Estimation

After producing trees for these nets, the congestion estimate of each edge is calculated using the method given in §3.2. Table 2 lists the congestion results when producing trees for the nets for the three cases: (i) only minimum-bend trees are made; (ii) additional trees are made for all the trees; and (iii) additional trees are made based on congestion. In the second, third, and fourth columns represent the number of trees, the maximum congestion estimate for the edges, and the average congestion estimate over all the edges are given when only minimum-bend, minimum-length trees are made. In columns 5, 6, and 7, the maximum congestion estimate and average congestion estimate are given when a set of additional trees are made for

Table 2 Congestion Results for Different Tree Construction Schemes

	Only RMSTs			No congestion			With congestion		
Circuit	Trees	Max	Avg.	Trees	Max	Avg.	Trees	Max	Avg.
Fract	130	6.5	1.630	170	6.6	1.683	165	5.5	1.725
Primary 1	800	7.5	2.206	1,106	7.57	2.288	821	7.25	2.211
Primary 2	2,121	14.5	3.772	3,510	14.55	3.846	2,338	11.0	3.791
Struct	1,713	11.5	2.689	2,319	11.15	2.775	1,785	9.25	2.701
Industry 1	2,433	25.5	5.043	4,677	19.54	5.168	3,084	15.889	5.046
Biomed	3,370	9.00	1.375	4,899	8.77	1.448	3,423	7.7	1.377
Industry 2	12,556	27.5	4.586	30,481	23.65	4.677	13.875	24.38	4.588
Industry 3	21,754	49	12.719	43,334	47.97	12.893	24,501	41.14	12.47
Avg. small	21,433	18.5	5.120	41,882	17.58	5.203	24,385	14.58	5.118
Avg. large	23,270	17	5.097	47,000	16.46	5.178	26,166	13.12	5.101

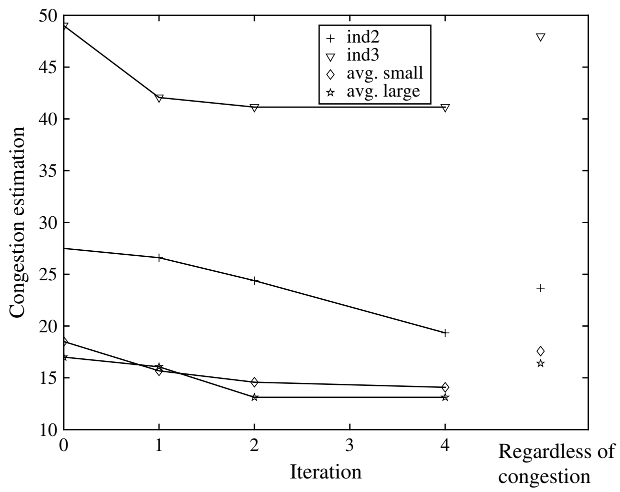

Figure 5 Comparing the Maximum Congestion Estimates Results for Large Test Circuits at Each Iteration of Tree Construction

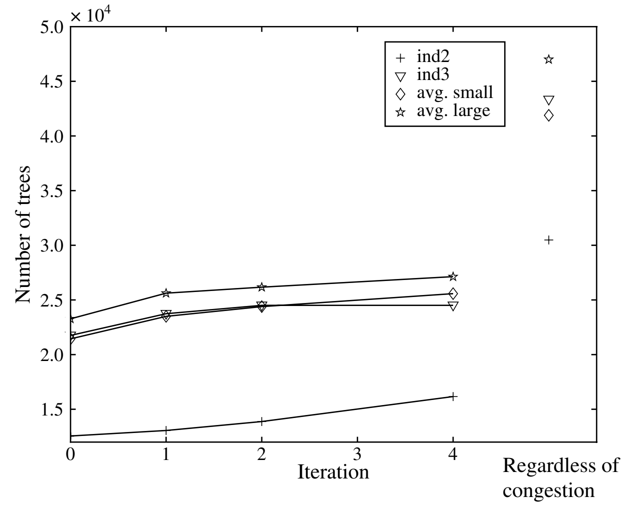

Figure 6 Number of Constructed Trees at Each Iteration of Tree Construction for Large Test Circuits

the all the nets regardless of congestion . Finally, in the eighth, ninth, and tenth columns, the maximum and the average estimation are given when producing additional trees for the nets in the congested areas.

In Figure 5, a comparison between the maximum congestion-estimation results for the larger test circuits in Table 2 is shown. In this figure, the x-axis represents the tree-construction iterations. The last entry in this axis is the maximum congestion estimate when no congestion calculation is used to make one iteration of additional trees. As can be seen from this figure, for all of the test circuits, the maximum congestion estimate becomes substantially lower in the first two iterations.

Figure 6 plots of the number of trees made for each of these test circuits during the iterations of tree construction. This figure illustrates that the rate of increase in the number of trees for different iterations is not high when congestion is incorporated in making trees. On the other hand, when congestion is not considered in making trees, the number of trees constructed becomes very high.

5.2. Modeling Results

To obtain binary solutions from the relaxed problem, a simple rounding scheme is used. A large number of rounding iterations is needed to obtain an optimal solution if the distinction between the trees constructed for a net is not apparent. This problem happens mostly for the minimization-of-total-wire-length model and the minimization-of-maximum-capacity model. In Table 3, the number of the nets that have a tree with a fractional value under 0.9 are given. This table illustrates that for the penalty and the combined models proposed in this paper, number of the nets that have maximum values less than 0.9 is very low compared to the total number of optimization-problem variables. In all of the test circuits, the cost of the integer solution is equal to the cost of the linear solution. In other words, the solution of the relaxed LP model yields *optimal* 0/1 solutions in polynomial time for these test problems.

In Table 4, a comparison of the results for the maximum capacities for all the four models is given. The results in this table indicate that the maximum channel capacity is substantially lowered for the penalty and combined models. By comparing the results in Tables 3 and 4 it can be seen that the new modeling formulations enhance the solution of the global routing problem and improve the quality of the results. The other characteristics of the penalty and combined models is their flexibility in handling different aspects of the global routing problem. These models result in the lowest wire length, via count and congestion.

In practical applications in VLSI computer-aided design the results obtained from the models developed in this paper can be used to develop routings with less congestion and lower channel capacity. Reducing congestion during routing of a circuit can

Table 3 Comparison of the Number of Unrouted Nets Between Different ILP Models

Circuit	MTWL	MMC	Penalty	Combined
Fract	20	30	0	5
Primary 1	157	178	4	4
Primary 2	129	135	12	10
Struct	310	346	7	6
Industry 1	604	DNC	1	27
Biomed	55	590	32	32
Industry 2	2,960	DNC	64	81
Industry 3	5,115	DNC	31	151
Avg. small	4,964	DNC	51	68
Avg. large	5,277	DNC	78	119

Note. DNC = did not converge.

Table 4 Comparison Between the Used Channel Capacity of Different ILP Models

Circuit	MTWL	MMC	Penalty	Combined
Fract	6	6	5	5
Primary 1	7	8	7	7
Primary 2	12	13	12	11
Struct	9	10	9	9
Industry 1	23	—	15	11
Biomed	7	9	7	7
Industry 2	30	—	27	15
Industry 3	55	—	45	33
Avg. small	22	—	17	13
Avg. large	33	—	17	13

reduce the probability that the solution will become infeasible, resulting in time savings.

6. Conclusions

In this paper a route-construction technique and two novel models for the global routing problem have been proposed and implemented. First, the routing area was presented as a graph. A method to produce RSTs has been implemented. This method considers congestion and via count when constructing trees. In addition, congestion estimates along with lower and upper estimates are calculated. The congestion estimates were later used in the formulation of the ILP model.

Different ILP models were developed for the global routing problem. The objective functions of these models incorporate minimization of wire length, minimization of the number of vias, congestion, and maximum channel capacity in the objective of the optimization problem. Numerical results obtained from the benchmark circuits illustrate that new models result in better routings compared to conventional routings.

The route-construction technique is most effective for constructing a set of trees for short nets. For long nets, efficient heuristics to produce RMSTs can be used. Comparing the models developed, penalty and combined models show that the channel capacity is much lower for the combined model with minimal loss in the number of unrouted nets. The trade-off for using the combined model is the higher number of variables in the global routing formulation and dense columns in the constraint matrix. The results for the combined and the penalty model are not overly dependent on the rounding heuristic because of the very high percentage of nets with fractional values higher than 0.9.

Acknowledgments

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding this research. This research is partially supported by NSERC Discovery Grants 15296 and 262463.

References

Aarts, E. H. L., P. J. M. V. Laarhoven. 1985. Statistical cooling: A general approach to combinatorial optimization problems. *Phillips J. Res.* **4** 193–226.

Alpert, C. J., M. Hrkic, J. Hu, A. B. Kahng, J. Lillis, B. Liu, S. T. Quay, S. S. Sapatnekar, A. J. Sullivan, P. Villarrubia. 2001. Buffered Steiner trees for difficult instances. *Proc. Internat. Sympos. Physical Design*. ACM Press, New York, 4–9.

Andersen, E. D., J. Gondzio, C. Meszaros, X. Xu. 1996. *Implementation of Interior-Point Methods for Large Scale Linear Programming*. Kluwer Academics Publishers, Dordrecht, The Netherlands.

Areibi, S., M. Xie, A. Vannelli. 2001. An efficient Steiner tree algorithm for VLSI global routing. *Proc. Canadian Conf. Electrical and Comput. Engrg*. IEEE Press, Toronto, Ontario, Canada, 1067–1072.

Behjat, L., D. Kucar, A. Vannelli. 2002. A novel eigenvector technique for large scale combinatorial problems in VLSI layout. *J. Combin. Optim.* **6** 271–286.

Bertsekas, D. P. 1995. *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Chiang, C., C. K. Wong, M. Sarrafzadeh. 1994. A weighted Steiner tree-based global router with simultaneous length and density minimization. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **13** 1461–1469.

Cong, J., P. H. Madden. 1997. Performance driven routing with multiple sources. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **16** 410–419.

Cong, J., A. B. Kahng, K. Leung. 1998. Efficient algorithms for minimum shortest path Steiner arborescence problem with application to VLSI physical design. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **17** 24–39.

Dees, W. A., P. G. Karger. 1982. Automated rip-up and reroute techniques. *Proc. 19th IEEE/ACM Design Automation Conf.* IEEE Press, Piscataway, NJ, 432–439.

Etawil, H., S. Areibi, A. Vannelli. 1999. Attractor-repeller approach for global placement. *IEEE/ACM Internat. Conf. Comput.-Aided Design*. IEEE Press, Piscataway, NJ, 7–11.

Floundas, C. A. 1995. *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, Oxford, UK.

Hadlock, F. 1975. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.* **4** 221–225.

Hanan, M. 1966. Steiner's problem with rectilinear distance. *SIAM J. Appl. Math.* **14** 255–265.

Hu, J., S. S. Sapatnekar. 2002. A timing-constrained simultaneous global routing algorithm. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **21** 1025–1036.

Hu, T. C., M. T. Shing. 1985. A decomposition algorithm for circuit routing. IEEE Press, New York, 144–152.

Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* **4** 373–395.

Karp, R. M. 1972. *Reducibility Among Combinatorial Problems*. Plenum Press, New York.

Kastner, R., E. Bozorgzadeh, M. Sarrafzadeh. 2001. An exact algorithm for coupling-free routing. *Proc. Internat. Sympos. Physical Design*. ACM Press, New York, 10–15.

Kozminski, K. 1991. Benchmarks for layout synthesis-evaluation and current status. *Proc. 28th ACM/IEEE Design Automation Conf.* IEEE Press, Piscataway, NJ, 265–270.

Kozminski, K. 2004. Benchmarks for layout synthesis—Evolution and current status. ACM Sigda, New York. http://www.sigda.org/Archives/ProceedingArchives/Dac/Dac91/papers/1991/dac91/17_3/17_3.htm.

Lee, C. Y. 1961. An algorithm for path connection and its application. *IRE Trans. Electronic Comput.* **10** 346–365.

Lengauer, T. 1990. *Combinatorial Algorithms for Integrated Circuit Layout*. Wiley, New York.

Lengauer, T., M. Lungering. 2000. Provably good global routing of integrated circuits. *SIAM J. Optim.* **11** 1–30.

Lundy, M. 1986. Convergence of an annealing algorithm. *Math. Programming* **34** 111–124.

Madden, P. 2004. Standard cell benchmark circuits. *BLAC CAD Binghamton Laboratory for Algorithms, Circuits, and Computer-Aided Design*. Binghamton University, Binghamton, NY. http://vlsicad.cs.binghamton.edu/benchmarks.html.

Mehrotra, S. 1992. On the implementation of a primal-dual interior-point method. *SIAM J. Optim.* **2** 575–601.

Ng, A. P., P. Raghavan, C. D. Thompson. 1987. Experimental results for a linear program global router. *Comput. Artificial Intelligence* **6** 130–143.

Raghavan, P. 1988. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *J. Comput. Sci.* **37** 130–143.

Raghavan, P., C. D. Thompson. 1991. Multi-terminal global routing: A deterministic approximation scheme. *Algorithmica* **6** 73–82.

Sherwani, N. 1999. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Shragowitz, E., S. Keel. 1987. A global router on a multi-commodity flow model. *Interaction* **5** 3–16.

Soukup, J. 1978. Fast maze router. *Proc. 15th Design Automation Conf.* IEEE Press, Piscataway, NJ, 100–102.

Vanderbei, R. 1998. LOQO: An interior point code for quadratic programming. Technical report, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ.

Vanderbei, R., D. F. Shanno. 1997. An interior point algorithm for non-convex nonlinear programming. Technical Report SOR-97-21, School of Engineering and Operational Research, Princeton University, Princeton, NJ.

Vannelli, A. 1991. An adaptation of the interior point method for solving the global routing problem. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **10** 193–203.

Vecchi, M. P., A. Kirkpatrick. 1983. Global wiring by simulated annealing. *IEEE Trans. Comput.-Aided Design Integrated Circuits Systems* **2** 215–222.