ISE Industrial and Systems Engineering

A Provably Good Global Routing Algorithm in Multilayer IC and MCM Layout Design

Mohamed Saad
University of Sharjah

Anthony Vannelli
University of Guelph

Hu Zhang
Canadian Imperial Bank of Commerce

Tamás Terlaky
Lehigh University

Report: 08T-004

A Provably Good Global Routing Algorithm in Multilayer IC and MCM Layout Designs

Mohamed Saad * Tamás Terlaky† Anthony Vannelli‡ Hu Zhang§

June 5, 2008

Abstract

Given a multilayer routing area, we consider the global routing problem of selecting a maximum set of nets, such that every net can be routed entirely in one of the given layers without violating the physical capacity constraints. This problem is motivated by applications in multilayer IC and multichip module (MCM) layout designs. The contribution of this paper is threefold. First, we formulate the problem as an integer linear program (ILP). Second, we modify an algorithm by Garg and Könemann for packing linear programs to obtain an approximation algorithm for the global routing problem. Our algorithm provides solutions *guaranteed* to be within a certain range of the global optimal solution, and runs in polynomial-time even if all, possibly exponentially many, Steiner trees are considered in the formulation. Finally, we demonstrate that the complexity of our algorithm can be significantly reduced in the case of identical routing layers.

1 Introduction

Traditionally, the VLSI routing process is divided into two phases: global routing and detailed routing. Global routing is to find a routing tree for each net, and detailed routing assigns the actual tracks and vias. Advances in VLSI fabrication technology have made it possible to use multiple routing layers for interconnections. A significant amount of research exists on handling multiple routing layers in the detailed routing phase. However, only limited research exists on

*Department of Electrical and Computer Engineering, University of Sharjah, Sharjah, UAE. Email: msaad@sharjah.ac.ae.

†School of Computational Engineering and Science, Department of Computing and Software, McMaster University, Hamilton, ON, Canada. Email: terlaky@mcmaster.ca.

‡College of Physical and Engineering Sciences, University of Guelph, Guelph, ON, Canada. Email: vannelli@uoguelph.ca.

§Canadian Imperial Bank of Commerce, Toronto, ON, Canada. Email: hu.zhang@cibc.ca.

multilayer global routing [13]. In other words, multiple routing layers have been dealt with in the detailed routing phase rather than in the global routing phase [13].

Given a *multilayer* routing area and a set of nets, we consider the *global routing* problem of selecting a maximum (weighted) subset of nets, such that every net can be routed entirely in one of the given layers without violating the physical capacity constraints. This problem is motivated by the following.

- Routing the majority of nets each in a single layer significantly reduces the number of required vias in the final layout. It is known that vias increase fabrication cost and degrade system performance [7].
- Routing the majority of nets each in a single layer greatly simplifies the detailed routing problem in multilayer IC design [7].

A similar multilayer topological planar routing problem was addressed in [8] and [7]. Given a number of routing layers, these studies addressed problem of choosing the maximum (weighted) subset of nets such that each net can be topologically routed entirely in one of the given layers. In particular, Cong and Liu proved in [8] that the problem is NP-hard. A provably good greedy algorithm for the problem was presented by Cong, Hossain and Sherwany [7]. A limitation of these studies is that they considered only planar routing, i.e., physical capacity constraints were not considered [7]. Moreover, planar routing graphs cannot handle state-of-the-art technologies properly [13].

The contribution of this paper is threefold.

- We formulate the multilayer global routing problem of selecting the maximum subset of nets such that every net can be routed entirely in one of the given layers without violating the physical capacity constraints, as an integer linear program (ILP).
- We modify an algorithm by Garg and Könemann for packing linear programs to obtain an approximation algorithm for the global routing problem. Our algorithm provides solutions *guaranteed* to be within a certain range of the global optimal solution, and runs in polynomial-time even if all, possibly exponentially many, Steiner trees are considered in the formulation.

- We demonstrate that the complexity of our algorithm can be significantly reduced in the case of identical routing layers.

The remainder of this paper is organized as follows. In Section 2, we model the global routing problem as an ILP. In Section 3, we present a polynomial-time approximation algorithm for the linear programming (LP-) relaxation of the problem (i.e., for the *fractional* global routing problem), and establish its performance guarantee and computational complexity. A reduced complexity algorithm is introduced, in Section 4, for the case of identical routing layers. In Section 5, we derive our overall approximation guarantee for solving the *integer* global routing problem. Section 6 concludes the paper.

2 Mathematical Model

In this section we introduce an ILP formulation for the global routing problem. Following [1], an undirected grid graph $G = (V, E)$ is constructed. In other words, a two-dimensional grid is placed over the chip. For each tile, there is a vertex $v \in V$, and two vertices corresponding to adjacent tiles are connected by an edge. In other words, each edge $e \in E$ represents a routing area between two adjacent tiles. In multilayer designs, an edge may consist of more than one layer [19]. In particular, the following are given as inputs to the problem.

- V: the set of vertices in the routing graph, $|V| = N$.
- E: the set of edges in the routing graph, $|E| = M$.
- $\mathcal{L} = \{1, 2, \ldots, L\}$: the set of available routing layers[1].
- $c_{e,l}$: the capacity of edge $e \in E$ on layer $l \in \mathcal{L}$.
- $\mathcal{I}$: the set of nets. Each net $i \in \mathcal{I}$ is defined by a subset of vertices $V_i \subseteq V$ that need to be connected. In particular, a net $i \in \mathcal{I}$ is realized by finding a Steiner tree that connects all vertices in V_i.

[1]A routing layer considered in this paper may, in practice, be implemented as a pair of layers: one for wiring in the x direction, and the other for wiring in the y direction. The problem formulation and algorithm presented in this paper avoids the use of stacked vias between different pairs of layers. However, vias used to connect wires within any pair of layers may be required. These vias are less expensive, and may be minimized in the detailed routing phase. In the sequel, pairs of layers are simply termed "layers".

- $\mathcal{T}_i$: the set of all Steiner trees in G that can be used to realize net $i \in \mathcal{I}$. In other words, every tree $T \in \mathcal{T}_i$ connects the vertices in V_i. It is worth noting that $\mathcal{T}_i$ can be exponentially sized. our algorithm, however, do not require that the sets $\mathcal{T}_i$ are explicitly given.

A net $i \in \mathcal{I}$ is realized by finding a Steiner tree $T \in \mathcal{T}_i$ that is routed entirely in one of the given layers $l \in \mathcal{L}$. The objective is to maximize the number of nets successfully realized. The design variables are $\{x_i(T,l) : i \in \mathcal{I}, T \in \mathcal{T}_i, l \in \mathcal{L}\}$, where for some $i \in \mathcal{I}$:

$$x_i(T,l) = \begin{cases} 1, & T \in \mathcal{T}_i \text{ is selected to route net } i \text{ on layer } l \in \mathcal{L}; \\ 0, & \text{otherwise.} \end{cases}$$

The global routing problem can be cast as ILP as follows:

$$\begin{aligned} \max \quad & \sum_{l \in \mathcal{L}} \sum_{i \in \mathcal{I}} \sum_{T \in \mathcal{T}_i} x_i(T,l) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} \sum_{T \in \mathcal{T}_i : e \in T} x_i(T,l) \le c_{e,l}, \ \forall e, l && \text{(1a)} \\ & \sum_{l \in \mathcal{L}} \sum_{T \in \mathcal{T}_i} x_i(T,l) \le 1, \ \forall i && \text{(1b)} \\ & x_i(T,l) \in \{0,1\}, \ \forall i, T, l. && \text{(1c)} \end{aligned}$$

Equation (1a) represents the capacity constraints. It ensures that the number of nets routed over any edge e and assigned to the same layer l does not exceed the capacity $c_{e,l}$ of that edge. Equation (1b) ensures that at most one tree is chosen for every net i. Equation (1c) represents the non-negativity and integrality constraints of the variables. The objective is to maximize the number of nets successfully routed.

It is straightforward to see that the global routing problem as formulated by (1) is NP-hard. In fact, it contains the unsplittable maximum multicommodity flow problem as a special case. In particular, let $\mathcal{L}$ contain only one layer. Also, let every net $i \in \mathcal{I}$ contain only two vertices, i.e., for every net $i \in \mathcal{I}$ let $V_i = \{s_i, d_i\}$ where $s_i, d_i \in V$. In this case $\mathcal{T}_i$ will contain only simple paths that join s_i and d_i. Under these restrictions ILP (1) will be equivalent to the following problem: Given a graph $G = (V, E)$, a capacity associated with every edge, and a set of commodities (each defined by a pair of vertices and associated with a unit demand), we seek to route a subset of the commodities of maximum total demand, such that every demand is routed along a single path and that total flow routed across any edge is bounded by its capacity. This is precisely the unsplittable maximum multicommodity flow problem, which is known to be NP-hard [12].

The NP-hardness of the global routing problem as given by (1) justifies the use of heuristics. In this paper, however, we are interested in polynomial-time approximation algorithms that have a theoretically proven worst-case performance guarantee. We start by giving an efficient algorithm to solve the linear programming (LP-) relaxation of ILP (1).

3 A Provably Good Algorithm for Fractional Global Routing

We briefly digress from the global routing problem to a more general packing problem, which is a special kind of LPs. In fact the LP-relaxation of (1) is a packing problem. In this section we will design a fast approximation algorithm for the LP-relaxation of (1) based on the method in [18].

We consider the following *fractional packing problem*:

$$\begin{array}{rl} \max & c^T x \\ \text{s.t.} & Ax \leq b; \\ & x \geq 0. \end{array} \tag{2}$$

Here A is a $m \times n$ positive matrix, and $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ are positive vectors. Since in (1), there are exponentially many variables, which can not be solved by many exact algorithms for LPs, e.g., standard interior point methods. The volumetric cutting plane method [2] or the ellipsoid method with separation oracle [10] may be employed, but in general they lead to large complexity. Therefore, we are interested in approximation algorithms.

The approximation algorithms for fractional packing problems {refpacking) are well studied in [9, 14, 18, 20]. All these algorithms are based on the duality relation for LPs. However, the algorithms in [14, 20] run in a time depending on the input data, and therefore only lead to polynomial time algorithms. The algorithm in [9] is the first with a strictly polynomial time but the block problem (subproblem) is required to be polynomial time solvable. Unfortunately it is not the case as we shall show later that the block problem of LP-relaxation for (1) is $\mathcal{NP}$-hard. Hence, we will apply the algorithm proposed in [18].

The approximation algorithm in [18] is an iterative approach. It maintains a sequence of a pair of the primal solution x and the dual solution y. At each iteration, for a pre-computed $y \in \mathbb{R}^m$, an approximate block solver $ABS(y)$ is called once that finds a column index q that:

$$(A_q)^T y / c_q \leq r \min_j (A_j)^T y / c_j,$$

where $r \geq 1$ is the approximation ratio of the block solver, which plays a role similar to the separation oracle in [10]. It is shown in [18] that their algorithm can find a $(1-\epsilon)/r$-approximate solution within coordination complexity (bounds on the number of iterations) of $O(m\epsilon^{-2}\ln m)$. The approximation algorithm for fractional packing problem (2) is in Table 1.

$\delta = 1-\sqrt{1-\epsilon}$, $u = (1+\delta)((1+\delta)m)^{-1/\delta}$, $f = 0$, $y_i = u/b_i$, $D = um$;
while $D < 1$ **do** {iteration}
call $ABS(y)$ to find a column index q;
$p = \arg\min_i b_i/A_{i,q}$;
$x_q = x_q + b_q/A_{p,q}$;
$f = f + c_q b_p/A_{p,q}$;
$y_i = y_i \left[1 + \delta \frac{b_p/A_{p,q}}{b_i/A_{i,q}}\right]$;
$D = \sum_{i=1}^{m} b_i y_i$;
end do

Table 1: Approximation algorithm for fractional packing problems.

In the algorithm the parameters f and D are in fact the objective values of the primal and dual programs for current pair x and y. It is showed in [18] that the scaled solution $x/\log_{1+\delta}((1+\delta)/u)$ at the final iteration is a feasible solution and the corresponding objective value is at least $(1-\epsilon)OPT/r$, where OPT is the optimum value of (2). For the complexity, the following result holds:

Proposition 1 *[18] There exists a $(1-\epsilon)/r$-approximation algorithm for the packing problem (2) running in $O(m\epsilon^{-2}\ln m)$ iterations, each iteration calling an r-approximate block solver once.*

It is worth noting that the complexity of the algorithm in [18] is independent of the input data or the approximation ratio r, which is similar to the result in [11] for convex min-max resource-sharing problems.

Applying the approximation algorithm for fractional packing problems to the LP-relaxation of (1) yields the following result:

Theorem 1 *There is a $(1-\epsilon)/r$-approximation algorithm for the LP-relaxation of (1) with a running time $O((ML+|\mathcal{I}|)L|\mathcal{I}|\epsilon^{-2}\beta\ln(ML+|\mathcal{I}|))$, where r and β are the ratio and the running time of the approximate minimum Steiner tree solver called as the approximate block solver, respectively.*

Proof. We just need to consider the block problem. There are two types of components in the dual vector y. The first type of components corresponding to the first set of constraints (capacity constraints) in (1) are $y_1, \ldots, y_M, y_{M+1}, \ldots, y_{2M}, \ldots, y_{M(L-1)+1}, \ldots, y_{ML}$, which corresponds to edge $e_i \in E$ and the layer l. The remaining components $y_{ML+1}, \ldots, y_{ML+|\mathcal{I}|}$ corresponds to the second set of constraints in (1). It is easy to verify that the block problem of the LP-relaxation of (1) is to find a tree T such that

$$\min_i \min_l \min_{T \in \mathcal{T}_i} (\sum_{e \in T} y_{Ml+e} + y_{ML+i} \delta_{i,T}), \tag{3}$$

where the indicator variable $\delta_{i,T} = 1$ if $T \in \mathcal{T}_i$, and otherwise $\delta_{i,T} = 0$. To find the minimum, we can just search a number of $L|\mathcal{I}|$ trees to attain the following minima:

$$\min_{T \in \mathcal{T}_i} \sum_{e \in T} y_{Ml+e},$$

for all $i = 1, \ldots, |\mathcal{I}|$ and $l = 1, \ldots, L$. Then we can find the minimal objective value of (3) over all these $L|\mathcal{I}|$ trees. If we regard the first ML components of the dual vector $y_1, \ldots, y_{ML}$ as the length associated to all edges in the given graph for all layers, then $\min_{T \in \mathcal{T}_i} \sum_{e \in T} y_{Ml+e}$ is equivalent to finding a tree on the l-th layer for net i with a minimum total length. Now the block problem is in fact the minimum Steiner tree problem in graphs. With an r-approximate minimum Steiner tree solver and using the approximation algorithm for fractional packing problems in [18], we can prove the theorem. ■

Unfortunately, the minimum Steiner tree problem is $\mathcal{APX}$-hard [3, 4]. The best known lower and upper bounds on the approximation ratio are $96/95 \approx 1.0105$ [6] and $1 + (\ln 3)/2 \approx 1.550$ [17], respectively. Thus, we can only use the approximation algorithm in [18] with an approximate minimum Steiner tree solve to obtain a feasible solution to the LP-relaxation of (1) to obtain an approximation algorithm with theoretical performance bounds.

4 A Reduced Complexity Algorithm for Identical Layers

In this section we consider the special case of the global routing problem where for every edge $e \in E$:

$$c_{e,1} = c_{e,2} = \ldots = c_{e,L} = c_e$$

This corresponds to the situation of all routing layers being identical. In this case, the LP-relaxation of the global routing problem will be given as follows.

$$\begin{aligned}
\max \quad & \sum_{l\in\mathcal{L}}\sum_{i\in\mathcal{I}}\sum_{T\in\mathcal{T}_i} x_i(T,l) && \\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}}\sum_{T\in\mathcal{T}_i:e\in T} x_i(T,l) \le c_e,\ \forall e,l && \text{(4a)}\\
& \sum_{l\in\mathcal{L}}\sum_{T\in\mathcal{T}_i} x_i(T,l) \le 1,\ \forall i && \text{(4b)}\\
& 0 \le x_i(T,l) \le 1,\ \forall i,T,l. && \text{(4c)}
\end{aligned}$$

Now, consider another special case of the global routing problem as given by (1), where the number of routing layers is reduced to one and the capacity of every edge e is set to $c_e \cdot L$. Let this problem be termed *single-layer problem.* It is straightforward to see that the LP-relaxation for the single-layer problem is given as follows.

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{I}}\sum_{T\in\mathcal{T}_i} y_i(T) && \\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}}\sum_{T\in\mathcal{T}_i:e\in T} y_i(T) \le c_e\cdot L,\ \forall e && \text{(5a)}\\
& \sum_{T\in\mathcal{T}_i} y_i(T) \le 1,\ \forall i && \text{(5b)}\\
& 0 \le y_i(T) \le 1,\ \forall i,T. && \text{(5c)}
\end{aligned}$$

Recall that M and L denotes the number of edges in the routing graph and the number of routing layers, respectively. Moreover, let $|\mathcal{T}|$ and $|\mathcal{I}|$ denote the total number of Steiner trees in the graph and the total number of nets, respectively. The number of constraints in the multilayer LP (4) is $M\cdot L+|\mathcal{I}|$, while the number of constraints in the single-layer LP (5) is $M+|\mathcal{I}|$. Moreover, the number of variables in (4) is $|\mathcal{T}|\cdot L$, while the number of variables in (5) is $|\mathcal{T}|$. To give more insight, note that, in the case of ten routing layers, the single-layer LP as given by (5) has an order of magnitude less constrai.nts and variables than the multilayer LP as given by (4). In the following theorem we establish the fact that solving LP (5) provides the same solution and objective function value as solving LP (4).

Theorem 2 *Let OPT_m denote the optimal objective function value of the multlayer LP given by (4). Also, let OPT_s denote the optimal objective function value of the single-layer LP given by (5). Then, $OPT_m = OPT_s$.*

Proof. We establish the proof by showing that $OPT_m \le OPT_s$ and $OPT_m \ge OPT_s$.

Let $\{x_i^*(T,l) : i \in \mathcal{I}, T \in \mathcal{T}_i, l \in \mathcal{L}\}$ be an optimal solution to LP (4). Define $y_i^*(T) = \sum_{l\in\mathcal{L}} x_i^*(T,l)$ for every $i \in \mathcal{I}$ and $T \in \mathcal{T}_i$. By (4b) and (4c), for every $i \in \mathcal{I}$ and $T \in \mathcal{T}_i$, we have $0 \le y_i^*(T) = \sum_{l\in\mathcal{L}} x_i^*(T,l) \le 1$. Also, by (4a), for every $e \in E$, we have $\sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i:e\in T} y_i^*(T) = \sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i:e\in T} x_i^*(T,l) \le \sum_{l\in\mathcal{L}} c_e = c_e \cdot L$. Furthermore, by (4b), for every $i \in \mathcal{I}$, we have $\sum_{T\in\mathcal{T}_i} y_i^*(T) = \sum_{l\in\mathcal{L}\sum_{T\in\mathcal{T}_i} x_i^*(T,l)} <= 1$. In other words, $\{y_i^*(T) : i \in \mathcal{I}, T \in \mathcal{T}_i\}$ is a feasible solution for LP (5). Consequently,

$$\sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i} y_i^*(T) \le OPT_s. \tag{6}$$

By replacing $y_i^*(T)$ by its definition in terms of $x_i^*(T,l)$ in (6), we conclude that

$$OPT_m \le OPT_s. \tag{7}$$

Conversely, let $\{y_i^*(T) : i \in \mathcal{I}, T \in \mathcal{T}_i\}$ be an optimal solution to LP (5). Define $x_i^*(T,l) = \frac{1}{L} \cdot y_i^*(T)$ for every for every $i \in \mathcal{I}$, $T \in \mathcal{T}_i$ and $l \in \mathcal{L}$. By (5c), for every $i \in \mathcal{I}$, $T \in \mathcal{T}_i$ and $l \in \mathcal{L}$, we have $0 \le x_i^*(T,l) \le 1$. Also by (5a), for every $e \in E$ and $l \in \mathcal{L}$, we have $\sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i:e\in T} x_i^*(T,l) = \frac{1}{L} \cdot \sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i:e\in T} y_i^*(T) \le \frac{1}{L} \cdot c_e \cdot L = c_e$. Furthermore by (5b), for every $i \in \mathcal{I}$, we have $\sum_{l\in\mathcal{L}} \sum_{T\in\mathcal{T}_i} x_i^*(T,l) = \frac{1}{L} \cdot \sum_{l\in\mathcal{L}} \sum_{T\in\mathcal{T}_i} y_i^*(T) = \sum_{T\in\mathcal{T}_i} y_i^*(T) \le 1$. In other words, $\{x_i^*(T,l) : i \in \mathcal{I}, T \in \mathcal{T}_i, l \in \mathcal{L}\}$ is a feasible solution for LP (4). Consequently,

$$\sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{I}} \sum_{T\in\mathcal{T}_i} x_i^*(T,l) \le OPT_m. \tag{8}$$

By replacing $x_i^*(T,l)$ by its definition in terms of $y_i^*(T)$ in (8), we conclude that

$$OPT_s \le OPT_m. \tag{9}$$

Combining (7) and (9) completes the proof. ∎

Moreover, we have the following result.

Corollary 1 *Let $\{y_i^*(T) : i \in \mathcal{I}, T \in \mathcal{T}_i\}$ be an optimal solution to LP (5). Then, $\{x_i^*(T,l) : i \in \mathcal{I}, T \in \mathcal{T}_i, l \in \mathcal{L}\}$, where $x_i^*(T,l) = \frac{1}{L} \cdot y_i^*(T)$ for every for every $i \in \mathcal{I}$, $T \in \mathcal{T}_i$ and $l \in \mathcal{L}$, is an optimal solution to LP (4).*

Proof. Follows directly from the proof of Theorem 2. ∎

Furthermore, using the same argument we can show that if $y_i^*(T)$ is a ρ-approximate solution to LP (5), then $x_i^*(T,l) = \frac{1}{L} \cdot y_i^*(T)$ is a ρ-approximate solution to LP (4). Therefore, the algorithm presented in Section 3 can be used at reduced complexity to obtain a provably good solution to the single-layer LP as given by (5). This solution can then be used to obtain a solution of precisely the same quality to the multilayer LP as given by (4).

5 The Approximation Algorithm

As usual, our approximation algorithm for the global routing problem in multilayer VLSI design is as follows: We first solve the LP-relaxation of (1) to obtain a fractional solution; Then we round the fractional solution to obtain feasible solution to (1).

By the algorithm in [18], we are able to obtain a $(1-\epsilon)/r$-approximate solution for the LP-relaxation of (1). Then we apply the randomized rounding in [15, 16] to generate the integer solution. Based on the scaling technique in [15, 16], for any real number v satisfying $(ve^{1-v})^c < 1/(m+1)$, where $c = \min_{e,l} c_{e,l}$ is the minimal capacity, we can obtain a bound for the integer solution by randomized rounding:

Theorem 3 *There is an approximation algorithms for the global routing problem in multilayer VLSI design such that the objective value is no less than*

$$\begin{cases} (1-\epsilon)vOPT/r - (\exp(1)-1)(1-\epsilon)v\sqrt{OPT\ln(M+1)/r}, & \text{if } OPT > r\ln(M+1); \\ (1-\epsilon)vOPT/r - \dfrac{\exp(1)(1-\epsilon)v\ln(M+1)}{1+\ln\left(r\ln(M+1)/OPT\right)}, & \text{otherwise}, \end{cases}$$

where OPT is the optimal integer solution to (1).

Another strategy to obtain an approximate solution to (1) is to directly apply the approach to find $(1-\epsilon)/r$-approximate solution for integer packing problems in [18]. Thus we have the following result:

Theorem 4 *If all edge capacities are not less than $(1+\log_{1+\delta}(ML+|\mathcal{I}|))/\delta$, then there exists an algorithm that finds a $(1-\epsilon)/r$-approximate integer solution to the global routing problem in multilayer VLSI design (1) within $O((ML+|\mathcal{I}|)L|\mathcal{I}|\epsilon^{-2}c_{\max}\beta\ln(ML+|\mathcal{I}|))$ time, where r and β are the ratio and the running time of the approximate minimum Steiner tree solver called as the oracle, and $c_{\max}$ is the maximum edge capacity.*

Though this approach has complexity depending on the input data, i.e., it is only a pseudo polynomial time approximation algorithm, it is worth using this method for some instances as the rounding stage is avoided.

In addition, at each iteration there are only $L|\mathcal{I}|$ Steiner trees generated. Thus, there are only a polynomial number of Steiner trees generated by using the approximation algorithm for fractional packing problem in [18], though there are exponentially many variables. This is similar to the column generation technique for LPs.

Corollary 2 *The approximation algorithms for the global routing problem in multilayer VLSI design only generates at most $O((ML + |\mathcal{I}|)L|\mathcal{I}|\epsilon^{-2}\ln(ML + |\mathcal{I}|))$ Steiner trees.*

6 Concluding Remarks

Given a multilayer routing area, this paper has addressed the problem of selecting the maximum (weighted) set of nets, such that every net can be routed entirely in one of the given routing layers without violating the physical capacity constraints. This problem is motivated by the following.

- Routing the majority of nets each in a single layer significantly reduces the number of required vias in the final layout. It is known that vias increase fabrication cost and degrade system performance [7].

- Routing the majority of nets each in a single layer greatly simplifies the detailed routing problem in multilayer IC design [7].

First, we have formulated the problem as an integer linear program (ILP). Second, we have modified an algorithm by Garg and Könemann [9] for packing linear programs to obtain a $(1-\varepsilon)/r$ approximation algorithm for the LP-relaxation of the global routing problem, where r is the approximation ratio of solving the minimum Steiner tree problem[2]. This has led also to an algorithm for the *integer* global routing problem that provides solutions *guaranteed* to be within a certain range of the optimal solution, and runs in polynomial-time even if all, possibly exponentially many, Steiner trees are considered in the formulation. Finally, we have demonstrated that the complexity of our algorithm can be significantly reduced in the case of identical routing layers.

[2]The best known apoproximation guarantee for the minimum Steiner tree problem is 1.55 [17]

References

[1] C. Albrecht, "Gloabal Routing by New Approximation Algorithms for Multicommodity Flow", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 622–632, May 2001.

[2] K. M. Anstreicher,"Towards a Practical Volumetric Cutting Plane Method for Convex Programming", *SIAM Journal on Optimization*, vol. 9, pp. 190–206, 1999.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, "Proof Verification and Hardness of Approximation Problems", *Journal of the Association for Computing Machinery*, vol. 45, pp. 501–555, 1998.

[4] M. Bern and P. Plassmann, "The Steiner Problem with Edge Lengths 1 and 2", *Information Professing Letters*, vol. 32, pp. 171–176, 1989.

[5] C.-C. Chang and J. Cong, "An Efficient Aproach to Multilayer Layer Assignment with an Application to Via Minimization", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 18, no. 5, pp. 608–620, May 1999.

[6] M. Chlebík and J. Chlebíková, "Approximation Hardness of the Steiner Tree Problem", *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory*, SWAT 2002, LNCS 2368, pp. 170–179.

[7] J. Cong, M. Hossain and N.A. Sherwani, "A Provably Good Multilayer Topological Planar Routing Algorithm in IC layout Designs", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 12, no. 1, pp. 70–78, Jan. 1993.

[8] J. Cong and C.L. Liu, "On the k-Layer Planar Subset and Via Minimization Problems", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 10, no. 8, pp. 972–981, Aug. 1991.

[9] N. Garg and J. Könemann, "Fast and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems", *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 1998, pp. 300–309.

[10] M. Grötschel, L. Lovász and A. Schrijver, "The Ellipsoid Method and Its Consequences in Combinatorial Optimization", *Combinatorica*, vol. 1, pp. 169-197, 1981.

[11] K. Jansen and H. Zhang, "Approximation Algorithms for General Packing Problems with Modified Logarithmic Potential Function", *Proceedings of the 2nd IFIP International Conference on Theoretical Computer Science*, TCS 2002, pp. 255-266.

[12] J.M. Kleinberg, "Approximation Algorithms for Disjoint Paths Problems", Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1996.

[13] L.-C. E. Liu and C. Sechen, "Multilayer Chip-Level Global Routing Using an Efficient Graph-Based Steiner Tree Heuristic", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 18, no. 10, pp. 1442–1451, Oct. 1999.

[14] S.A. Plotkin, D.B. Shmoys and E. Tardos, "Fast Approximation Algorithms for Fractional Packing and Covering Problems", *Mathematics of Operations Research*, vol. 2, pp. 257–301, 1995.

[15] P. Raghavan, "Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs", *Journal of Computer and System Sciences*, vol. 37, pp. 130–143, 1988.

[16] P. Raghavan and C. D. Thompson, "Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs", *Combinatorica* vol. 7, no. 4, pp. 365–374, 1987.

[17] G. Robins and A. Zelikovsky, "Improved Steiner Tree Approximation in Graphs", *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2000, pp. 770–779.

[18] M. Saad, T. Terlaky, A. Vennelli and H. Zhang, "Packing Trees in Communication Networks", accepted for publication in *Journal of Combinatorial Optimization*, 2008.

[19] J. Xiong and L. He, "Extended Global Routing With *RLC* Crosstalk Constraints", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 3, pp. 319–329, March 2005.

[20] N.E. Young, "Randomized Rounding without Solving the Linear Program", *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, SODA 1995, pp. 170–178.