

McMaster University

Advanced Optimization Laboratory



Title:

On Routing in VLSI Design
and Communication Networks

Authors:

Tamás Terlaky, Anthony Vannelli and Hu Zhang

AdvOl-Report No. 2005/14

August 2005, Hamilton, Ontario, Canada

On Routing in VLSI Design and Communication Networks*

Tamás Terlaky[†] Anthony Vannelli[‡] Hu Zhang[§]

Abstract

In this paper, we study the global routing problem in VLSI design and the multicast routing problem in communication networks. We first propose new and realistic models for both problems. In the global routing problem in VLSI design, we are given a lattice graph and subsets of the vertex set. The goal is to generate trees spanning these vertices in the subsets to minimize a linear combination of overall wirelength (edge length) and the number of bends of trees with respect to edge capacity constraints. In the multicast routing problem in communication networks, a graph is given to represent the network, together with subsets of the vertex set. We are required to find trees to span the given subsets and the overall edge length is minimized with respect to capacity constraints. Both problems are \mathcal{APX} -hard. We present the integer linear programming formulation of both problems and solve the linear programming (LP) relaxations by the fast approximation algorithms for min-max resource-sharing problems in [21]. For the global routing problem, we investigate the particular property of lattice graphs and propose a combinatorial technique to overcome the hardness due to the bend-dependent vertex cost. Finally, we develop asymptotic approximation algorithms for both problems with ratios depending on the best known approximation ratio for the minimum Steiner tree problem. They are the first known theoretical approximation bound results for these problems.

*Research supported by a MITACS grant for all the authors, the NSERC Discovery Grant #5-48923 for the first and third author, NSERC Grant #15296 for the second author, and the Canada Research Chair Program for the first author.

[†]Advanced Optimization Lab., Department of Computing and Software, McMaster University, Hamilton, ON, Canada. terlaky@mcmaster.ca.

[‡]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. a.vannelli@ece.uwaterloo.ca.

[§]Advanced Optimization Lab., Department of Computing and Software, McMaster University, Hamilton, ON, Canada. zhanghu@mcmaster.ca.

1 Introduction

Hardness of many combinatorial problems is represented by two characteristics in their mathematical programming formulation: First, the integrality of variables leads to \mathcal{NP} -hardness for many problems. Second, in general the limited resources represented by the constraints increase the complexity of the problems. An example of the second characteristics is the shortest path problem in graphs, which is polynomial time solvable, while the shortest edge-disjoint paths problem with multiple source-destination pairs is hard, where we are given multiple pairs of vertices to be connected by edge-disjoint paths and the total edge length is minimized, which is a special case of the unsplittable min-cost flow problem. In fact the occurrence of constraints changes the feasible set of the problem, i.e., the polytope of the mathematical program. Thus the characteristics of the problem differs much from the original problem. Furthermore, in many problems arising in graph theory, when trees are required to be generated instead of paths, \mathcal{NP} -hardness appears as well. For instance, the shortest path problem is easy, while the minimum Steiner tree problem is \mathcal{APX} -hard.

In this paper, we shall study routing problems in two different engineering applications: in VLSI design and in multicast communication networks. In both problems, there are only limited resources shared by multiple entities. Furthermore, the solutions of both problems consist of trees generated for these entities. Specifically, we shall study the *global routing problem in VLSI design* and the *multicast routing problem in communication networks*. There exist capacity constraints on all edges. The purpose of both problems is to minimize the *total cost* (e.g., the overall edge length of the trees or with some additional cost). As described above, both problems are hard. In fact we shall show that they are at least as hard as the *minimum Steiner tree problem* in graphs. We propose asymptotic approximation algorithms for both problems. They are the first results of this kind. In addition, as a byproduct, we develop a combinatorial technique for lattice graphs with bend-dependent vertex cost, which can be applied to many optimization problems in VLSI design and transportation networks.

1.1 Global Routing in VLSI Design

In the global routing stage of VLSI design, the circuits are assumed to be in a one-layer frame instead of the actual multi-layer frame. The channels on a chip form the edge set of a lattice graph, and their crosses consist of the vertex set. Pins of components are located on the vertices. A group of pins

to be connected is called a *net*. In general in the input of an instance there are many nets. Wires are to be routed along the channels to connect the nets with respect to the channel capacity, and certain objective functions (e.g., overall wirelength or maximum edge congestion) are optimized.

The global routing problem is \mathcal{NP} -hard [31]. Therefore, heuristics tend to be used to find reasonable solutions in fast time. The solution methodologies are usually broken into two classes: (i) sequential techniques and (ii) integer programming techniques.

In sequential global routing, nets are ordered to their routing importance, then each net is routed separately. This method is called the *maze runner* method and was initially introduced by Lee [29]. Enhancements to the maze runner heuristics were shown in [18]. Several approaches to extend these techniques to multi-terminal nets by decomposing these nets into two-terminal nets were shown in [26]. The quality of a sequential global router depends on the ordering of the nets. Nets are usually ordered based on their importance, half-perimeter wirelength and number of terminals [45].

Integer programming methodologies allow global routing problems to be solved in a “global” sense by routing all the nets simultaneously. This class of techniques is used to formulate the global routing problem as a 0-1 integer programming problem where the objective is to select one Steiner tree for each net such that the total wirelength (tree length) is minimized and channel capacity (edge capacity) constraints are not violated.

Recent approaches use linear relaxations to find approximate solutions to this problem [31]. In [46], a linear relaxation of the routing problem is described as a multicommodity network flow problem. Randomized rounding [41, 42] is used to generate an integer solution to the LP problem in [25]. Finally, local search techniques including sequential routing, genetic routing and randomized procedures are used in [32] to produce legal routings.

Three traditional models of the global routing by integer programming are listed in [3]. In the first model [47], the goal is to minimize the overall wirelength with respect to the capacity constraints, which was also studied in [40]. The goal of the second model is to minimize the maximum tree length [32]. The third model is to minimize the maximum edge congestion. In [42], based on the LP-relaxation given in [25], an approximation algorithm was proposed to find a satisfactory tree in a pre-specified polynomial size tree set.

Based on the above traditional models, two new models are proposed in [3, 4]. In the first model, the goal is to minimize a linear combination of overall wirelength, overall number of vias (bends of the routed paths or trees) and the maximum edge congestion. The weights of these three terms depend

on a pre-specified set of trees. A big improvement is that the impact of vias is counted, while vias increase the hardness of manufacturing and leads to more heat generation in general. In the second model, besides the objective in the first model, the maximum edge congestion is to be minimized. To solve the problems, first a polynomial size set of trees for nets is manually defined such that the trees in the set have relatively small wirelength and small number of vias. Then an integer program is proposed to find one tree in the set for each net with the capacity constraints. An interior point method is applied to solve the LP relaxation and randomized rounding is used to obtain a feasible solution. This is similar to the strategy given in [42]. However, there is no performance bound for the solution.

A related problem is to find shortest rectilinear paths or rectilinear minimum bends paths [30, 37, 38, 51, 52, 53]. However, here we study the case that the routing is realized by Steiner trees. More details of the global routing problem in VLSI design can be found in [3].

A feasibility version of the global routing problem to maximize the successfully routed nets was studied in [23, 15, 16, 17, 36, 43].

1.2 Multicast Routing in Communication Networks

We notice that a model of the global routing problem in VLSI design [32, 42] is to minimize the maximum edge congestion, which is essentially equivalent to the multicast congestion problem [2, 22] or multicast packing problem [9] in communication networks. In fact there are many similarities between the routing problems in VLSI design and the routing problems in communication networks. In general the former problem is only modelled on lattice graphs. This motivates us to find the link between these two categories.

A multicast communication network consists of a number of processors (usually routers) and other devices. The processors can receive, duplicate and deliver packets of data, and are linked by infrastructure backbone. The signals are transmitted among processors through fixed cables. Here we also assume that each processor is able to send the signals to any other processors linked to it. Furthermore, for each cable there is a bandwidth (capacity) constraint. There exist a certain number of *requests* in a multicast communication network. Each request utilizes some of the processors. One of them is the source processor to send the data packets, and others are destinations of the signal flows. We also assume that each processor is able to deal with the signal flows of different requests simultaneously. Thus, it is required to find a routing for each request in the network structure such that the transmission can be realized. We do not consider fault-tolerant systems,

so the routing of every request is realized by a tree spanning the processors involved in the request.

From algorithmic point of view, a communication network can be represented by an undirected graph, where the vertices are the processor and the edges are the cables. A request is a subset of the vertex set. A solution of a multicast routing problem is a set of trees spanning the requests with respect to the edge capacity constraints. And the goal is to minimize the total cost. Heuristics for this problem was studied in [50], but no theoretical approximation bound result is reported.

In the multicast congestion/multicast packing problem, there is no edge capacity nor cost of the routing. The goal is to minimize the maximum edge congestion. Approximation algorithms for this problem were proposed in [2, 7, 22, 48]. Furthermore, experimental studies of approximation algorithms [2, 34] show that the asymptotic approximation algorithm of [22] is promising.

Another related problem is the so-called group multicast routing problem [5, 24]. However, they deal with multiple identical requests in directed graphs with heuristics [24], or give an efficient algorithm for the problem in directed acyclic graphs [5]. The undirected version of the group multicast routing problem is called as the minimum Steiner forest problem with bandwidth constraints in [35], though the definition of the minimum Steiner forest problem is different from that in [12]. In fact it is proved that the minimum Steiner forest problem with bandwidth constraints can not be approximated within ratio $\exp(\text{poly}(n))$, unless $\mathcal{P} = \mathcal{NP}$ [35].

Furthermore, a variant of the above multicast routing problem in undirected graphs is studied in [6, 19, 33], where the source vertex can only send multicast message to a bounded number of destinations such that a request may be realized by more than one tree. Their studies, however, still deal with single request.

A feasibility version of the multicast routing problem to maximize the number of routing trees for one multicast request or its variant are studied in [10, 20, 27, 27]. Its generalization to maximize the throughput for different multicast request is studied in [44].

1.3 Our Contribution

In this paper, we propose generalized models for the global routing problem in VLSI design and the multicast routing problem in communication networks. Then we develop approximation algorithms for the linear relaxations and obtain approximate solutions for the routing problems by applying ran-

domized rounding. As a by product, we design a combinatorial technique for lattice graphs, which can be employed in many optimization problems in VLSI design and urban transportation networks.

For the global routing problem, our model generalizes the previous models developed in [3, 4, 42, 47]. Here we consider the three important factors in global routing: the total wirelength, the edge congestion, and the number of vias. The edge capacity can vary according to the local requirement instead of posing extra edge length as penalty for the highly congested edges. For instance, the edge capacity can be specified as a relatively small number in or around areas of potential hot spots. The goal is to optimize a combination of the total wirelength and total number of bends. According to the actual requirement, the values of the two weights corresponding to above terms can be adjusted to fulfil the required design criteria.

Similarly, for the multicast routing problem, our model also generalizes the previous models in [2, 5, 7, 22, 24, 35, 48]. It is a special case of the global routing problem, but in arbitrary graphs. We optimize the overall cost with respect to the edge capacity constraints, which can also vary locally to reduce the maximum edge congestion.

In our approximation algorithms for the routing problems, we first apply binary search strategy to reformulate the linear relaxations to the *convex min-max resource-sharing problem* (the *packing problem* in the linear case). Then we use the approximation algorithm in [21] as a subroutine to solve the packing problems. We show that the block problem of the multicast routing problem is the minimum Steiner tree problem in graphs. We also develop a combinatorial technique for lattice graphs, with which we also convert the block problem of the global routing problem to the minimum Steiner tree problem. Finally we obtain asymptotic approximation algorithms for the routing problems. To our best knowledge, no approximation results have been previously reported for these problems.

In addition, with the combinatorial technique for lattice graphs, we also present polynomial time algorithms for some generalized optimization problems in urban transportation networks.

The remainder of this paper is organized as follows: We give the formal models of the routing problems in Section 2 and reformulate them to the packing problems in Section 3. The approximation algorithms are presented in Section 4, and the approximation guarantee is shown in Section 5. Finally we address some more applications of the combinatorial technique in Section 6.

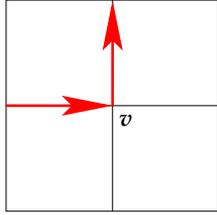


Figure 1: A path with a bend on v .

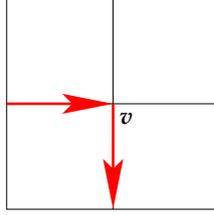


Figure 2: A path with a bend on v .

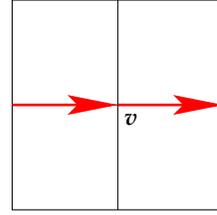


Figure 3: A path without bend on v .

2 Mathematical Formulation

The global routing problem in VLSI design we study is defined as follows: Given an edge-weighted lattice graph and a set of nets (subsets of the vertex set), the edge set is associated with a nonnegative length function and a positive capacity function. The given edge capacity can differ from the physical channel capacity in order to control the congestion and reduce the hot spots when the chip turns on. Furthermore, there also exists a vertex cost function. The solution is a set of trees spanning the given nets. The total cost of a solution consists of two parts: the edge cost and the bend-dependent vertex cost. The edge cost is the sum of edge length of the generated trees, and the bend-dependent vertex cost is caused by the structure of the generated trees. For a tree in a solution in the given lattice graph, if the tree has a bend at vertex v , then its bend-dependent vertex cost on v is the given vertex cost (see Figure 1 and 2). Otherwise, its bend-dependent vertex cost on v is 0 (see Figure 3). In fact a bend on a vertex corresponds a via in VLSI design, which leads to extra cost for manufacturing and results in risk of hot spots. The goal of the problem is to minimize the overall cost, which is a linear combination of the total edge cost and the total bend-dependent vertex cost, while the edge capacity constraints are fulfilled.

In the multicast routing problem in communication networks, we are given an edge-weighted undirected graph and requests (subsets of the vertex set). There are a nonnegative cost function and a positive capacity function associated to the edge set, while there is no vertex cost. The goal is to find a set of trees spanning the requests with respect to the edge capacity constraints, such that the overall cost is minimized.

The difference between the global routing problem in VLSI design and the multicast routing problem in communication networks lies in the following two points: First, there is no vertex cost in the multicast routing problem while it occurs in the global routing problem. Second, we only deal

with lattice graphs in VLSI design, while a communication network can have arbitrary topologies. In fact, both problems are \mathcal{NP} -hard. Furthermore, we have the following complexity result:

Proposition 2.1 *Both the global routing problem in VLSI design and the multicast routing problem in communication networks are \mathcal{APX} -hard. They are at least as hard as the minimum Steiner tree problem in graphs.*

Proof: We consider a restricted version of the global routing problem in VLSI design and the multicast routing problem in communication networks. In this restricted version, there is only one net/request. In addition, the vertex cost in the global routing problem is zero, and all edge capacities are sufficiently large. Thus the problem is to find a tree to span the only net/request S_1 and the total edge length is minimized. In fact this is exactly the minimum Steiner tree problem in graphs. Since the minimum Steiner tree problem is \mathcal{APX} -hard, the lemma follows. \square

In the following we give the mathematical formulation of the global routing problem in VLSI design. Formally, given an planar edge-weighted lattice graph $G = (V, E)$ (with some rectangular holes) and nets $S_1, \dots, S_K \subseteq V$, the edge set is associated with a length function $l : E \rightarrow \mathbb{R}^+ \cup \{0\}$ and a capacity function $c : E \rightarrow \mathbb{R}^+$. We assume that $|S_k|$ is bounded by some constant for all $k = 1, \dots, K$. Furthermore, there also exists a vertex cost $w : V \rightarrow \mathbb{R}^+ \cup \{0\}$. A feasible solution is a set of K trees spanning S_1, \dots, S_K with respect to the edge capacity constraints. The goal is to minimize the overall cost defined as a linear combination $\alpha l_{\text{total}} + \beta v_{\text{total}}$, where l_{total} is the sum of edge length of all K trees and v_{total} is the sum of numbers of bends of all K trees, while $\alpha, \beta \geq 0$ are artificial weights corresponding to the impact of the total wirelength and the total number of vias whose values are set according to the design requirements and are given in advance. For simplicity, we denote by c_i the capacity of edge $e_i \in E$ from now on. In addition, by scaling, we can set $\alpha + \beta = 1$, i.e., the overall cost is a convex combination of the total edge length and the the total number of bends.

We now develop the integer linear program formulation of our generalized model. Denote by \mathcal{T}_k the set of all trees in G connecting the vertices in S_k . It is worth noting that $|\mathcal{T}_k|$ can be exponentially large. We also denote by $x_k(T)$ the indicator variable as follows:

$$x_k(T) = \begin{cases} 1, & \text{if } T \in \mathcal{T}_k \text{ is selected for the net } S_k; \\ 0, & \text{otherwise.} \end{cases}$$

In addition, we define by $l(T)$ and $v(T)$ the length of tree T and the number of vias of tree T , respectively. Therefore the integer linear program formulation of the global routing problem in VLSI design is as follows:

$$\begin{aligned}
\min \quad & \alpha \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T) + \beta \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} v(T)x_k(T) \\
\text{s.t.} \quad & \sum_{T \in \mathcal{T}_k} x_k(T) = 1, & \forall k = 1, \dots, K; \\
& \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T) \leq c_i, & \forall e_i \in E; \\
& x_k(T) \in \{0, 1\}, & \forall T \& k = 1, \dots, K.
\end{aligned} \tag{1}$$

Here the first set of constraints means that for any set \mathcal{T}_k we choose exactly one tree for S_k , and the second set of constraints are capacity constraints.

In the multicast routing problem G can be an arbitrary graph. Formally, we are given an edge-weighted undirected graph $G = (V, E)$ and requests $S_1, \dots, S_K \subseteq V$. There is a cost function $l : E \rightarrow \mathbb{R}^+ \cup \{0\}$ and a capacity function $c : E \rightarrow \mathbb{R}^+$ associated to the edge set. In this problem, there is no vertex cost. The goal is to find K trees spanning S_1, \dots, S_K with respect to the edge capacity constraints, such that the overall cost is minimized. The integer linear programming formulation of the multicast routing problem in communication networks is similar to (1) except that the weight β corresponding to the total number of bends is 0, and $\alpha = 1$. Accordingly, we just need to study the integer linear program (1) for both problems.

3 The Packing Formulation of the Generalized Model

In this section, we shall reformulate the integer linear program (1) of the global routing problem in VLSI design and the multicast routing problem in communication networks and its LP-relaxation. In fact we shall formulate the LP-relaxation as a convex min-max resource-sharing problem/a packing problem. In this way, existing approximation algorithms for convex min-max resource-sharing/packing problems can be employed for solving the LP-relaxation of (1).

First, we have the following equivalent formulation to (1):

Lemma 3.1 *The optimum value of the integer linear program (1) is equivalent to the optimum value of the following integer linear program:*

$$\begin{aligned}
\min \quad & g \\
\text{s.t.} \quad & \alpha \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T) + \beta \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} v(T)x_k(T) \leq g, \\
& \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T) \leq c_i, & \forall e_i \in E; \\
& \sum_{T \in \mathcal{T}_k} x_k(T) = 1, & \forall k = 1, \dots, K; \\
& x_k(T) \in \{0, 1\}, & \forall T \& k = 1, \dots, K.
\end{aligned} \tag{2}$$

Proof: We need to search for the minimum of g . Clearly for any fixed feasible solution x , when $g = \alpha \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T) + \beta \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} v(T)x_k(T)$ the minimum is attained. Therefore the optimum of (2) over all feasible solution is attained when $\alpha \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T) + \beta \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} v(T)x_k(T)$ is minimized under other constraints, which are same as the constraints in (1). Since exactly one tree is selected in any \mathcal{T}_k , the objective value of (1) is always positive. Thus the lemma is proved. \square

Furthermore, with the same argument we can show that any ρ -approximate solution of (2) is also a ρ -approximate solution of (1) for any $\rho \geq 1$. In this case we can study (2) instead of (1).

As usual, we shall study the LP-relaxation of (2), and then apply rounding techniques to obtain a feasible solution. It is worth noting that there may be exponentially many variables in (2) and its LP-relaxation. Thus many exact algorithms for LPs such as standard interior point methods can not be applied. The LP-relaxation of (2) may be solved by the volumetric-center [1] or the ellipsoid methods with separation oracle [14]. However, those approaches will lead to a large running time, which is very unsuitable to VLSI design or communication networks since these problems are of very large scale in general. Therefore, we shall study fast approximation algorithms for the LP-relaxation of (1).

Now we consider the LP-relaxation of (2). In fact we can show that the optimum solution of the LP-relaxation of (2) can be approximated by combining a binary search procedure and solving another linear program:

Lemma 3.2 *For any given $\varepsilon \in (0, 1)$, if we can solve the following linear program*

$$\begin{aligned}
\min \quad & \lambda \\
\text{s.t.} \quad & \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T)/c_i \leq \lambda, & \forall e_i \in E; \\
& \alpha \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T)/g + \beta \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} v(T)x_k(T)/g \leq \lambda, \\
& \sum_{T \in \mathcal{T}_k} x_k(T) = 1, & \forall k = 1, \dots, K; \\
& x_k(T) \in [0, 1], & \forall T \& k = 1, \dots, K,
\end{aligned} \tag{3}$$

then we can find a $(1 + \varepsilon)$ -approximate solution to the LP-relaxation of (2).

Proof: Assuming that we are given an oracle to solve (3), we shall apply the following strategy to solve the LP-relaxation of (2) approximately: We first guess a value of $g \in [g_L, g_U]$, where g_L and g_U are lower and upper bounds of g . With the guessed g we call the oracle to solve (3). In addition, we denote by OPT the optimum value of the LP-relaxation of (2). If in the solution, $\lambda > 1$, then the guessed g is too small such that OPT is larger than g because the capacity constraints must hold. If $\lambda \leq 1$, it means that g is too large such that OPT is smaller than the guessed g . In the first case we can set $g_L = g$ and $g = (g + g_U)/2$ to start the new iteration. In the second case we can set $g_U = g$ and $g = (g + g_L)/2$. This binary search procedure stops till the resulting objective value $\lambda \leq 1 + \varepsilon$ for the given accuracy $\varepsilon > 0$. This approach gives a $(1 + \varepsilon)$ -approximate solution to (2), and there are at most $\log(g_U/\varepsilon)$ iterations in this binary search procedure. Clearly, $g_U \leq K(m + n)$, because for any feasible tree the length can not be more than m and the number of bends is at most n . Therefore the number of iterations is bounded by $O(\log(Km\varepsilon^{-1}))$. \square

The remaining task is to solve (3). Here we call the linear program (3) the packing formulation of the global routing problem in VLSI design or the multicast routing problem in communication networks.

4 Approximation Algorithms for LP-Relaxations

We shall present the approximation algorithms for the LP-relaxations of the multicast routing problem in communication networks and the global routing problem in VLSI design. As indicated in Section 3, we will consider the LP-relaxation (3) on the different types of graphs due to the particular properties of the problems. Here we call the linear relaxations of the multicast

routing problem (respectively, the global routing problem) the *fractional multicast routing problem* (respectively, the *fractional global routing problem*). We shall study the fractional multicast problem at first and then the fractional global routing problem.

4.1 Approximation Algorithms for the Fractional Multicast Routing Problem

In the fractional multicast routing problem in communication networks, we shall minimize the overall edge length of the trees spanning requests with respect to the capacity constraints (e.g., $\alpha = 1$ and $\beta = 0$ in (1), (2) and (3)). Thus we need to solve the following linear program:

$$\begin{aligned}
\min \quad & \lambda \\
\text{s.t.} \quad & \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T)/c_i \leq \lambda, \quad \forall e_i \in E; \\
& \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T)x_k(T)/g \leq \lambda, \\
& \sum_{T \in \mathcal{T}_k} x_k(T) = 1, \quad \forall k = 1, \dots, K; \\
& x_k(T) \in [0, 1], \quad \forall T \& k = 1, \dots, K.
\end{aligned} \tag{4}$$

We notice that (4) is indeed the formulation of the convex min-max resource-sharing problems (packing problems in the linear case) in [13, 21, 49]. The convex min-max resource sharing problem is defined as follows:

$$\begin{aligned}
\min \quad & \lambda \\
\text{s.t.} \quad & f_m(x) \leq \lambda, \quad m = 1, \dots, M; \\
& x \in B,
\end{aligned} \tag{5}$$

where $B \subseteq \mathbb{R}^N$ is a nonempty convex compact set and $f_m : B \rightarrow \mathbb{R}_+$ are nonnegative continuous convex functions on B for $m \in \{1, \dots, M\}$. Let $f(x) = (f_1(x), \dots, f_M(x))^T$ and $\lambda(x) = \max_{m \in \{1, \dots, M\}} f_m(x)$ for $x = (x_1, \dots, x_N)^T \in B$. Denote by x^* the optimal solution and by $\lambda^* = \lambda(x^*)$ the optimal value of (5). In [13, 49], algorithms are proposed to find $(1 + \varepsilon)$ -approximate solutions to (5) provided an oracle to deliver a $(1 + O(\varepsilon))$ -approximate solution to the corresponding block problem (which depends on the specified $f(x)$). However, in the fractional multicast routing problem in communication networks (also the fractional global routing problem in VLSI design), we shall show that there is no PTAS for the block problem. Therefore there only exist r -approximation algorithms for the block problem with $r > 1$. In such a case the algorithms in [13, 49] are not applicable.

However, an approximation algorithm for convex min-max resource-sharing problems with only weak block solvers is proposed in [21]. We shall use that algorithm to solve (4).

The algorithm is based on the Lagrangian relaxation. The algorithm maintains a pair of a feasible solution x and a *price vector* (dual vector) $p \in P = \{p \in \mathbb{R}^M \mid \sum_{m=1}^M p_m = 1, p_m \geq 0\}$. The block problem of (5) is as follows [21]: for a given price vector p

ABS(p, t, r) compute $\hat{x} = \hat{x}(p) \in B$ such that $p^T f(\hat{x}) \leq r(1+t) \min\{p^T f(y) \mid y \in B\}$,

where $t = O(\varepsilon)$ and $r \geq 1$ is the approximation ratio. The algorithm works with the scaling phase strategy. In each scaling phase the relative error tolerance σ_s is set. Based on the known pair of x and p , a solution \hat{x} is delivered by the approximate block solver. Afterwards an appropriate convex combination of the old solution x and the block solution \hat{x} is computed as the new solution. The iteration stops when the solution satisfies any one of the two stopping rules. After one scaling phase, the error tolerance σ_s is halved and the next scaling phase starts until the error tolerance $\sigma_s \leq \varepsilon$. In the last scaling phase an $r(1 + \varepsilon)$ -approximate solution is delivered. The algorithms \mathcal{L} for (5) in [21] is showed in Table 1. More details of packing problems or convex min-max resource-sharing problems can be found in [54].

Proposition 4.1 [21] *For any given $\varepsilon \in (0, 1)$, Algorithm \mathcal{L} finds an $r(1 + \varepsilon)$ -approximate solution to the convex min-max resource sharing problem (5) in $O(M(\log M + \varepsilon^{-2} \log \varepsilon^{-1}))$ iterations and an $r(1 + O(\varepsilon))$ -approximate block solver is called once in each iteration.*

The approximation algorithm for packing problems in [39] or [8] could also be applied here. Their algorithm is used for the decision version of the packing problems. To use their algorithm for the optimization version of the problem, the binary search strategy is also necessary. Furthermore, the bound on the number of iterations of their algorithm is $O(\varepsilon^{-2} \rho \ln(M\varepsilon^{-1}))$, where $\rho = \max_{x \in B} \max_{m \in \{1, \dots, M\}} f_m(x)$ is the width of B , which can lead to only pseudo polynomial time approximation algorithms. So we do not use their algorithms here.

In the fractional multicast routing problem in communication networks, the packing constraints are: $f_i = \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T) / c_i$, for $i = 1, \dots, m$, and $f_{m+1} = \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} l(T) x_k(T) / g$. Then we can directly apply algorithm \mathcal{L} here and the following theorem holds:

Theorem 4.1 *For any given accuracy $\varepsilon \in (0, 1)$, there exists an $r(1 + \varepsilon)$ -approximate algorithm for the fractional multicast routing problem in communication networks, where r is the ratio of the approximate minimum Steiner tree solver called as the oracle.*

Proof: From Proposition 4.1, if there is an r -approximate block solver for (4), then we can obtain an $r(1 + \varepsilon)$ -approximation algorithm for (4) by using Algorithm \mathcal{L} in [21]. Actually the key issue to apply Algorithm \mathcal{L} for problem (5) is to identify the block problem $\min\{p^T f(x)|x \in B\}$ for a pre-specified price vector $p \in P$ [21]. We shall find the block problem for (4).

We denote by x_k a vector whose components are $x_k(T)$ for all $T \in \mathcal{T}_k$ for a fixed $k \in \{1, \dots, K\}$. Here the size of $x_k(T)$ can be exponentially large. Clearly, $x = (x_1^T, \dots, x_K^T)^T$ for the packing formulation (4). According to the third set of constraints in (4), the sum of components of x_k is 1, for any $k = 1, \dots, K$. Therefore the set B that $x \in B$ can be completely decomposed to K blocks $B = B_1 \times \dots \times B_K$, where each B_k corresponds to a vector x_k and is a simplex. Thus the optimum must exist on one vertex of each simplex.

The block problem is now to find an x such that the inner product of the price vector and the vector of the constraints is minimized, i.e.

$$\begin{aligned} \min_{x \in B} p^T f(x) &= \min_{x \in B} \left\{ \sum_{i=1}^m \left[p_i \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} \frac{1}{c_i} x_k(T) \right] + p_{m+1} \sum_{k=1}^K \sum_{T \in \mathcal{T}_k} \frac{l(T)}{g} x_k(T) \right\} \\ &= \min_{x \in B} \sum_{k=1}^K \left[\sum_{i=1}^m \sum_{T \in \mathcal{T}_k \& e_i \in T} \frac{p_i}{c_i} x_k(T) + \sum_{T \in \mathcal{T}_k} \frac{p_{m+1} l(T)}{g} x_k(T) \right] \\ &= \sum_{k=1}^K \min_{x_k \in B_k} \left[\sum_{T \in \mathcal{T}_k} \sum_{e_i \in T} \frac{p_i}{c_i} x_k(T) + \sum_{T \in \mathcal{T}_k} \frac{p_{m+1} l(T)}{g} x_k(T) \right]. \end{aligned}$$

The last equality holds because the set B can be completely decomposed to K blocks. So we just need to find the minimum in each block and sum them up to obtain the overall minimum. For any tree T , its length $l(T) = \sum_{e_i \in T} l_i$, where l_i is the length of edge e_i . So

$$\sum_{T \in \mathcal{T}_k} \frac{p_{m+1} l(T)}{g} x_k(T) = \sum_{T \in \mathcal{T}_k} \sum_{e_i \in T} \frac{p_{m+1} l_i}{g} x_k(T).$$

Thus we have that

$$\begin{aligned}
\min_{x \in B} p^T f(x) &= \sum_{k=1}^K \min_{x_k \in B_k} \sum_{T \in \mathcal{T}_k} \sum_{e_i \in T} \left(\frac{p_i}{c_i} + \frac{p_{m+1} l_i}{g} \right) x_k(T) \\
&= \sum_{k=1}^K \min_{T \in \mathcal{T}_k} \sum_{e_i \in T} \left(\frac{p_i}{c_i} + \frac{p_{m+1} l_i}{g} \right).
\end{aligned}$$

The last equality holds because each B_k is a simplex and for each k we can just take that components of x_k as 1, which corresponds to the minimum of $W_k = \sum_{e_i \in T} (p_i/c_i + p_{m+1} l_i/g)$, while other components all set to be zeros. In this case we can attain the minimum of the dual value $p^T f(x)$. Now the goal of the block problem is for each net S_k to find a tree connecting the vertices in S_k such that W_k is minimized. In fact, the goal is exactly to find a minimum Steiner tree for S_k , where the weight for edge e_i is $p_i/c_i + p_{m+1} l_i/g$. Thus we can apply existing approximation algorithms for the minimum Steiner tree problem as the approximate block solver. \square

4.2 Approximation Algorithms for the Fractional Global Routing Problem

The fractional global routing problem in VLSI design is formulated as the linear program (3), where the weight for the overall number of bends (vias) $\beta \neq 0$. The packing constraints in this case are $f_i = \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} x_k(T)/c_i$, for $i = 1, \dots, m$, and $f_{m+1} = \sum_{k=1}^K [\alpha \sum_{T \in \mathcal{T}_k} l(T)x_k(T)/g + \beta \sum_{T \in \mathcal{T}_k} v(T)x_k(T)/g]$.

Similar to Subsection 4.1, we will also apply the approximation algorithm \mathcal{L} in [21] for convex min-max resource-sharing problems. However, here the block problem is different from the one for the multicast routing problem in communication networks due to the additional cost caused by the impact of tree bends.

However, we still have the following result for the fractional global routing problem in VLSI design:

Theorem 4.2 *There exists an $r(1 + \varepsilon)$ -approximation algorithm for the fractional global routing problem (3) provided an r -approximate minimum Steiner tree solver is available.*

Proof: Similar to the proof of Theorem 4.1, we need to identify the block problem for the fractional global routing problem. We use the same notations and arguments as in Subsection 4.1.

The block problem is now to find an x such that the inner product of the price vector and the vector of the constraints is minimized, i.e.

$$\begin{aligned}
\min_{x \in B} p^T f(x) &= \min_{x \in B} \left\{ \sum_{i=1}^m \left[p_i \sum_{k=1}^K \sum_{T \in \mathcal{T}_k \& e_i \in T} \frac{1}{c_i} x_k(T) \right] \right. \\
&\quad \left. + p_{m+1} \sum_{k=1}^K \left[\alpha \sum_{T \in \mathcal{T}_k} \frac{l(T)}{g} x_k(T) + \beta \sum_{T \in \mathcal{T}_k} \frac{v(T)}{g} x_k(T) \right] \right\} \\
&= \min_{x \in B} \sum_{k=1}^K \left[\sum_{i=1}^m \sum_{T \in \mathcal{T}_k \& e_i \in T} \frac{p_i}{c_i} x_k(T) + \sum_{T \in \mathcal{T}_k} \frac{\alpha p_{m+1} l(T)}{g} x_k(T) \right. \\
&\quad \left. + \sum_{T \in \mathcal{T}_k} \frac{p_{m+1} \beta v(T)}{g} x_k(T) \right] \\
&= \sum_{k=1}^K \min_{x_k \in B_k} \left[\sum_{T \in \mathcal{T}_k} \sum_{e_i \in T} \frac{p_i}{c_i} x_k(T) + \sum_{T \in \mathcal{T}_k} \frac{\alpha p_{m+1} l(T)}{g} x_k(T) \right. \\
&\quad \left. + \sum_{T \in \mathcal{T}_k} \frac{p_{m+1} \beta v(T)}{g} x_k(T) \right] \\
&= \sum_{k=1}^K \min_{x_k \in B_k} \left[\sum_{T \in \mathcal{T}_k} \sum_{e_i \in T} \left(\frac{p_i}{c_i} + \frac{\alpha p_{m+1} l_i}{g} \right) x_k(T) \right. \\
&\quad \left. + \sum_{T \in \mathcal{T}_k} \frac{p_{m+1} \beta v(T)}{g} x_k(T) \right] \\
&= \sum_{k=1}^K \min_{T \in \mathcal{T}_k} \left[\sum_{e_i \in T} \left(\frac{p_i}{c_i} + \frac{\alpha p_{m+1} l_i}{g} \right) + \frac{p_{m+1} \beta v(T)}{g} \right].
\end{aligned}$$

The last equality holds because each B_k is a simplex and for each k we can just take that component of x_k as 1, which corresponds to the minimum of $W_k = \sum_{e_i \in T} (p_i/c_i + \alpha p_{m+1} l_i/g) + p_{m+1} \beta v(T)/g$, while all the other components are set to zero. In this case we can attain the minimum of the dual value $p^T f(x)$. Now the goal of the block problem is for each net S_k to find a tree connecting the vertices in S_l such that W_k is minimized. If $\beta = 0$, then the goal is to find a minimum Steiner tree for S_k , where the weight for edge e_i is $p_i/c_i + \alpha p_{m+1} l_i/g$, same as the multicast routing problem in communication networks. However, for the second term of W_k we need to apply a special technique.

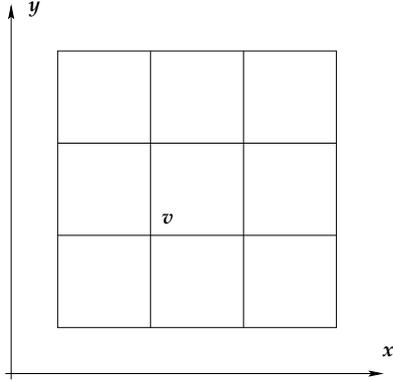


Figure 4: Original lattice graph G .

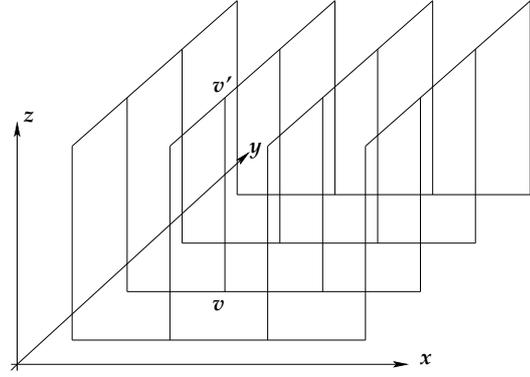


Figure 5: Two-layer graph H .

To solve this problem, we partition the edge set E of the given lattice graph G into two subsets E_x and E_y such that $E = E_x \cup E_y$ and $E_x \cap E_y = \emptyset$. E_x only consists of edges parallel to the x -axis (horizontal) while E_y consists of edges parallel to the y -axis (vertical) (see Figure 4). Now we construct a two-layer graph $H = (V_H, E_H)$ based on G as follows: For each vertex $v \in V$, there exist a pair of vertices v and $v' \in V_H$. The pair v and v' have the same x -coordinates and y -coordinates, but different z -coordinates. We call v' the *mirror* of v , and vice versa. The vertex set V_H is now on a 3-dimensional space, and locates in two layers. Each layer is a planar graph in an $x - y$ plane (with the same z -coordinates), and these two layers are parallel, i.e., for each vertex in one layer, there is a mirror vertex in the other layer with the same x - and y -coordinates. We define the layers with larger (respectively, smaller) z -coordinates by upper (respectively, lower) layer. Now we consider the edge set E_H . In the upper layer, there are only edges corresponding to those in E_y in the original graph G , i.e., the edge set E'_y in the upper layer is identical to E_y . Similarly, in the lower layer, there are only edges corresponding to those in E_x in G (the edge set E'_x is identical to E_x). Furthermore, we construct another edge set E'_z in H . Each edge in E'_z connects a pair of mirrors and is parallel to the z -axis. For simplicity we still use the notation E_x , E_y and E_z for the edge sets E'_x , E'_y and E'_z , respectively. Now for the new graph H the edge set $E_H = E_x \cup E_y \cup E_z$ (see Figure 5). It is obvious that one bend on vertex $v \in V$ in the original graph G corresponds to an edge $(v, v') \in E_z$ in the two-layer graph H . Furthermore, if a path in H passes through an edge $(v, v') \in E_z$, then it must have a bend on vertex v in the original graph G . Therefore, one time of using an edge in E_z in H is equivalent to a bend in G . If there is a path

in G with q bends, the corresponding path in H has the same edges in E_x and E_y and it is using q edges in E_z corresponding to the bends of the path in G . An example of the two-layer graph H of the original lattice graph G (see Figure 4) is showed in Figure 5.

We now set the weights to the edges in H . For any edge $e_i \in E_x \cup E_y$, we assign a weight $w_i = p_i/c_i + \alpha p_{m+1} l_i/g$ according to their indices in the original graph G . For every edge in E_z , we assign a weight $p_{m+1}\beta/g$. In this weighted two-layer graph H , a minimum Steiner tree for a net S_k corresponds to a tree for S_k in G with the minimum W_k . So when we apply Algorithm \mathcal{L} , the block problem corresponds to the classical Steiner tree problem in the graph H to minimize the overall edge weight of the Steiner tree connecting the vertices in S_k . We can apply an approximate solver for the Steiner tree problem as the block solver of Algorithm \mathcal{L} . According to Proposition 4.1, given any r -approximation algorithms for the minimum Steiner tree problem, there exists an $r(1 + \varepsilon)$ -approximation algorithm for the fractional global routing problem in VLSI design. This completes the proof. \square

Remark: In the fractional global routing problem in VLSI design, we need to deal with the influence of the number of bends in the generated trees spanning the nets, which makes the problem harder than the fractional multicast routing problem in communication networks. However, we make use of the particular structure of lattice graphs such that we can obtain the same approximation ratio for the fractional global routing problem as for the fractional multicast routing problem. The key issue is to partition the edges into two sets and assign them to two different layers, respectively. We call this combinatorial technique the *virtual layer method*. In fact later we will show that this technique can be also applied in some other optimization problems.

5 Approximation Algorithms

In Section 4 we have presented $r(1 + \varepsilon)$ -approximation algorithms for the fractional global routing problem in VLSI design and for the multicast routing problem in communication networks. Now we consider how to get a feasible solution (integer solution) for the original problem (1).

Our approximation algorithm for the global routing problem in VLSI design or the multicast routing problem in communication networks has two step. In the first step, Algorithm \mathcal{L} is applied to find a fractional solution

to the given instance. Then we need to find a feasible integer solution with guaranteed performance. To accomplish this, we use the randomized rounding in [41, 42]. Let OPT denote the optimal objective value of the LP relaxation of (3). Since the expectation of the rounded objective value of (3) is the objective value of the corresponding LP-relaxation given by our algorithm $r \cdot OPT$, then directly from [41] we have the following theorem:

Theorem 5.1 *There is an approximation algorithm for the global routing problem in VLSI design and the multicast routing problem in communication networks such that the objective value is bounded by*

$$\begin{cases} r(1 + \varepsilon)OPT + (\exp(1) - 1)(1 + \varepsilon)\sqrt{r \cdot OPT \ln m}, & \text{if } r \cdot OPT > \ln m; \\ r(1 + \varepsilon)OPT + \frac{\exp(1)(1 + \varepsilon) \ln m}{1 + \ln(\ln m / (r \cdot OPT))}, & \text{otherwise.} \end{cases}$$

If $r \cdot OPT > \ln m$, we can conclude that the objective value generated by our algorithm is in fact bounded by $\exp(1)r(1 + \varepsilon)OPT$. In the other case, the objective value generated by our algorithm is bounded by $r(1 + \varepsilon)OPT + O(\ln m)$. Therefore, we have obtained asymptotic approximation algorithms for the global routing problem in VLSI design and the multicast routing problem in communication networks.

In (1) the number of variables is exponential. However, by applying Algorithm \mathcal{L} , we just need to generate K minimum Steiner trees for the K nets/requests in each iteration corresponding to the current price vector. There are only a polynomial number of Steiner trees generated in Algorithm \mathcal{L} in total. The following theorem holds because each component of vector x corresponds to a tree:

Corollary 5.1 *Our approximation algorithms for the global routing problem in VLSI design and the multicast routing problem in communication networks generate only at most $O(Km(\log m + \varepsilon^{-2} \log \varepsilon^{-1}))$ Steiner trees.*

This is similar to the column generation technique for linear programs, and the hardness due to the exponential number of variables in (1) is overcome.

6 More Applications of the Virtual Layer Method

We have presented the virtual layer method in Subsection 4.2 to convert the block problem for the global routing problem in VLSI design to the

minimum Steiner tree problem in a two-layer graph. In this section we will give more applications of the virtual layer method.

Now we consider an optimization problem in lattice graphs as follows: In a given lattice graph $G = (V, E)$, there is a source-destination pair (s, t) and an amount of demand $d \leq \min_e c(e)$ between the pair, where $c(e)$ is the capacity of edge e . There is an edge cost function $l : E \rightarrow \mathbb{R}^+ \cup \{0\}$. Furthermore, for each vertex there is a vertex cost $w : V \rightarrow \mathbb{R}^+ \cup \{0\}$. A feasible solution is a routed path connecting the source and the destination. The overall cost of the path is the sum of edge costs and bend-dependent vertex costs, where the bend-dependent vertex cost of a vertex v is defined as follows: If the path has a bend on v (see Figure 1 and 2), then the bend-dependent vertex cost of v is its vertex cost $w(v)$; Otherwise (see Figure 3) the bend-dependent vertex cost of v is 0. The goal of the problem is to find a routing path between the source and the destination such that the overall cost of the path is minimized.

This problem is a variant of the classical shortest path problem in lattice graphs. The difference is that in this problem the vertex cost of the path also depends on its bends. This problem has a realistic background in urban transportation networks. In many cases the urban transportation networks are lattice graphs or can be represented as lattice graphs. When a vehicle reaches an intersection, it can turn left or right, or drive straight through the intersection. The delays corresponding to these three choices are different. We assume that the delays to turn left and right are identical due to the traffic situation, pedestrians and traffic signal lights. The delay to pass straight is much less than the delays of turns and can also be merged to the edge cost. To deliver a certain amount commodity from a source to a destination in such a transportation network, a route for the vehicle with a minimum overall time is desired. The vehicle should try to avoid turns in this case unless turns can lead to edges with lower costs such that the overall cost is reduced.

Theorem 6.1 *The shortest path problem in lattice graphs, where the total cost is the sum of edge the costs and the bend-dependent vertex costs, is polynomial time solvable.*

Proof: To solve this problem, we can directly apply the virtual layer method to generate a two-layer graph H based on the given lattice graph G . Now we consider the two-layer graph H . For any edge in $E_x \cup E_y$, we assign the same cost as in the original graph G to it. Furthermore, we define the cost of the edges in E_z as follows: For each edge $(v, v') \in E_z$, its cost $l(v, v') = w(v)$,

i.e., the edge cost is identical to the corresponding vertex cost in the original graph G . There is no vertex cost in H . If there is no bend of the path on v , then the corresponding path in H will not change to the other layer, i.e., will not pass through the mirror of v . Otherwise if there is a bend on v , then the path has to change to the other layer via the edge (v, v') . In the optimum path in the original graph G , the total cost (the sum of edge costs and the bend-dependent vertex costs) is minimized. Therefore, it corresponds to a shortest path in the graph H . Thus the original problem in G has been converted to find a shortest path in the new graph H , which can be solved in polynomial time. \square

In addition, the virtual layer method can be further employed to the fractional minimum cost flow problem and the fractional minimum cost multicommodity flow problem in lattice graphs, where the total cost is also the sum of edge costs and the bend-dependent vertex costs of the routed paths. By the virtual layer method, they are formulated as the fractional minimum cost flow problem and the fractional minimum cost multicommodity flow problem in the corresponding two-layer graphs, where the total cost is only the sum of the edge costs.

Corollary 6.1 *The fractional min-cost flow problem (or the fractional min-cost multicommodity flow problem) in lattice graphs, where the total cost is the sum of edge costs and the bend-dependent vertex costs, is polynomial time solvable.*

The virtual layer method makes use of the unique property of lattice graphs such that the change of direction is realized by a turn on a vertex. This technique is a crucial step for the design of our approximation algorithm for the global routing problem in VLSI design. We also believe that this method can find many applications in optimization problems in lattice graphs.

7 Conclusion

In this paper we have presented asymptotic approximation algorithms for the global routing problem in VLSI design and the multicast routing problem in communication networks. Our models generate many previous models for these routing problems. In fact, with binary search technique one can study a bicriteria version of the minimization problem with two objective functions: the overall cost and the maximum edge congestion.

There are many interesting topics for further study. First, we intend to apply the virtual layer method to more optimization problems in lattice graphs. Second, we would like to propose more general models for the global routing problem in VLSI design and for the multicast routing problems in communication networks such that the impact of more factors are included. Nice problems arise when one studies the 3-dimensional lattice graphs for the global routing problem. We are working on an implementation of our algorithm to explore the power of our approximation algorithm in computational practice.

References

- [1] K. M. Anstreicher, Towards a practical volumetric cutting plane method for convex programming, *SIAM Journal on Optimization*, 9 (1999), 190-206.
- [2] A. Baltz and A. Srivastav, Fast approximation of minimum multicast congestion - implementation versus theory, *RAIRO Operations Research*, 38 (2004), 319-344.
- [3] L. Behjat, New modeling and optimization techniques for the global routing problem, *Ph.D. Thesis*, University of Waterloo, 2002.
- [4] L. Behjat, A. Vannelli, W. Rosehart, Linear programming models for the global routing problem, *Inform Journal on Computing*, to appear in 2005.
- [5] M. Cai, X. Deng, and L. Wang, Minimum k arborescences with bandwidth constraints, *Algorithmica*, 38 (2004), 529-537.
- [6] Z. Cai, G. Lin, and G. L. Xue, Improved approximation algorithms for the capacitated multicast routing problem, *Proceedings of the 11th International Computing and Combinatorics Conference (COCOON 2005)*, LNCS 3595, 136-145.
- [7] R. Carr and S. Vempala, Randomized meta-rounding, *Proceedings of the 32nd ACM Symposium on the Theory of Computing (STOC 2000)*, 58-62.
- [8] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin, Approximating a finite metric by a small number of tree metrics, *Proceedings*

- of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998), 379-388.
- [9] S. Chen, O. Günlük, and B. Yener, The multicast packing problem, *IEEE/ACM Transactions on Networking*, 8 (3) (2000), 311-318.
 - [10] J. Cheriyan and M. Salavatipour, Hardness and approximation results for packing Steiner trees problems, *Proceedings of the 12th Annual European Symposium on Algorithms (ESA 2004)*, LNCS 3221, 180-191.
 - [11] 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, 170-179.
 - [12] N. Garg and R. Khandekar, Fast approximation algorithms for fractional Steiner forest and related problems, *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*, 500-509.
 - [13] M. D. Grigoriadis and L. G. Khachiyan, Coordination complexity of parallel price-directive decomposition, *Mathematics of Operations Research*, 2 (1996), 321-340.
 - [14] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, 1 (1981), 169-197.
 - [15] M. Grötschel, A. Martin, and R. Weismantel, Packing Steiner trees: a cutting plane algorithm and computational results, *Mathematical Programming*, 72 (1996), 125-145.
 - [16] M. Grötschel, A. Martin, and R. Weismantel, Packing Steiner trees: separation algorithms, *SIAM Journal on Discrete Mathematics*, 9 (1996), 233-257.
 - [17] M. Grötschel, A. Martin, and R. Weismantel, The Steiner tree packing problem in VLSI-design, *Mathematical Programming*, 78 (1997), 265-281.
 - [18] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, *SIAM Journal on Computing*, 4(3) (1975), 221-225.

- [19] X.-D. Hu, X. Jia, T. Shuai, and M.-H. Zhang, Multicast routing and wavelength assignment in WDM networks with limited drop-offs, *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2004).
- [20] K. Jain, M. Mahdian and M. R. Salavatipour, Packing Steiner trees, *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms* (SODA 2003), 266-274.
- [21] 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), 255-266.
- [22] K. Jansen and H. Zhang, An approximation algorithm for the multicast congestion problem via minimum Steiner trees, *Proceedings of the 3rd International Workshop on Approximation and Randomized Algorithms in Communication Networks* (ARACNE 2002), 77-90.
- [23] G. W. Jeong, K. Lee, S. Park, and K. Park, A branch-and-price algorithm for the Steiner tree packing problem, *Computers and Operations Research*, 29 (2002), 221-241.
- [24] X. Jia and L. Wang, A group multicast routing algorithm by using multiple minimum Steiner trees, *Computer Communications* 20 (1997), 750-758.
- [25] R. M. Karp, Reducibility among combinatorial problems, in *R. E. Miller and J. W. Thatcher (Eds.), Complexity of Computer Computations*, Plenum Press, NY, (1972), 85-103.
- [26] E. S. Kuh and M. Marek-Sadowska, Global routing, Elsevier Science Publishers B.V., Amsterdam, 1985, vol. 1, 133-168.
- [27] L. C. Lau, An approximate max-Steiner-tree-packing min-Steiner-cut theorem, *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science* (FOCS 2004), 61-70.
- [28] L. C. Lau, Packing Steiner forests, *Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization* (IPCO 2005), LNCS 3509, 362-376.
- [29] C. Y. Lee, An algorithm for path connection and its application, *IRE Transactions on Electronic Computers*, 10 (1961), 346-365.

- [30] D. T. Lee, C. D. Yang and C. K. Wong, Rectilinear paths among rectilinear obstacles, *Discrete Applied Mathematics*, 70 (3) (1996), 185-215.
- [31] T. Lengauer, Combinatorial algorithms for integrated circuit layout, J. Wiley, New York, 1990.
- [32] T. Lengauer and M. Lungering, Provably good global routing of integrated circuits, *SIAM Journal on Optimization*, 11(1) (2000), 1-30.
- [33] G. Lin, An improved approximation algorithm for multicast k-tree routing, *Journal of Combinatorial Optimization*, 9(3) (2005), 349-356.
- [34] Q. Lu and H. Zhang, Implementation of approximation algorithms for the multicast congestion problem, *Proceedings of the 4th International Workshop on Efficient and Experimental Algorithms (WEA 2005)*, LNCS 3503, 152-164.
- [35] B. Ma and L. Wang, On the inapproximability of disjoint paths and minimum Steiner forest with bandwidth constraints, *Journal of Computer and System Science*, 60 (2000), 1-12.
- [36] A. Martin and R. Weismantel, Packing paths and Steiner trees: routing of electronic circuits, *CWI Quarterly*, 6 (1993), 185-204.
- [37] J. S. B. Mitchell, L_1 shortest paths among polygonal obstacles in the plane, *Algorithmica*, 8(1) (1992), 55-88.
- [38] J. S. B. Mitchell, C. D. Piatko, and E. M. Arkin, Computing a shortest k-Link path in a polygon, *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science (FOCS 1992)*, 573-582.
- [39] S. A. Plotkin, D. B. Shmoys, and E. Tardos, Fast Approximation algorithms for fractional packing and covering problems, *Mathematics of Operations Research*, 2 (1995), 257-301.
- [40] W. R. Pulleyblank, Two Steiner tree packing problems, *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC 1995)*, 383-387.
- [41] P. Raghavan, Probabilistic construction of deterministic algorithms: approximating packing integer programs, *Journal of Computer and System Sciences*, 37 (1988), 130-143.

- [42] P. Raghavan and C. D. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, *Combinatorica* 7 (4) (1987), 365-374.
- [43] M. Saad, T. Terlaky, A. Vannelli, and H. Zhang, A provably good global routing algorithm in multilayer IC and MCM layout designs, *Technical Report*, AdvOL #2005-15, Advanced Optimization Lab., McMaster University, Hamilton, ON, Canada. <http://www.cas.mcmaster.ca/~oplab/research.htm>.
- [44] M. Saad, T. Terlaky, A. Vannelli, and H. Zhang, Packing Trees in Communication Networks, *Technical Report*, AdvOL #2005-13, Advanced Optimization Lab., McMaster University, Hamilton, ON, Canada. <http://www.cas.mcmaster.ca/~oplab/research.htm>.
- [45] N. Sherwani, Algorithms for VLSI physical design automation, Kluwer Academic Publishers, Dordrecht, 1999.
- [46] E. Shragowitz and S. Keel, A global router on a multi-commodity flow model, *Interaction*, 5 (1987), 3-16.
- [47] A. Vannelli, An adaptation of the interior point method for solving the global routing problem, *IEEE Transactions on Computer-Aided Design*, 10 (2) (1991), 193-203.
- [48] S. Vempala and B. Vöcking, Approximating multicast congestion, *Proceedings of the 10th International Symposium on Algorithms and Computation (ISAAC 1999)*, LNCS 1741, 367-372.
- [49] J. Villavicencio and M. D. Grigoriadis, Approximate Lagrangian decomposition with a modified Karmarkar logarithmic potential, *Network Optimization*, P. Pardalos, D. W. Hearn and W. W. Hager (Eds.), *Lecture Notes in Economics and Mathematical Systems 450*, Springer-Verlag, Berlin, (1997), 471-485.
- [50] C.-F. Wang, C.-T. Liang, and R.-H. Jan, Heuristic algorithms for packing multiple-group multicasting, *Computers and Operations Research*, 29 (2002), 905-924.
- [51] C. D. Yang, D. T. Lee, and C. K. Wong, On bends and lengths of rectilinear paths: a graph-theoretic approach, *International Journal of Computational Geometry and Applications*, 2(1) (1992), 61-74.

- [52] C. D. Yang, D. T. Lee, and C. K. Wong, Rectilinear path problems among rectilinear obstacles revisited, *SIAM Journal on Computing*, 24 (1992), 457-472.
- [53] C. D. Yang, D. T. Lee, and C. K. Wong, On bends and distance paths among obstacles in two-layer interconnection model, *IEEE Transactions on Computers*, 43 (1994), 711-724.
- [54] H. Zhang, Approximation algorithms for min-max resource sharing and malleable tasks scheduling, *Ph.D. Thesis*, University of Kiel, 2004.

```

Algorithm  $\mathcal{L}(f, B, \varepsilon, c)$ :
initialize:  $\sigma := 1, t := \sigma/6, p := \vec{1}/M, x := ABS(p, t, r)$ ;
            $\omega := (1 + \sigma)/((1 + \sigma/6)M), \bar{\lambda} := \lambda(x)$ ;
finished_scaling := false;
while not(finished_scaling) do {scaling phase}
  finished_coordination := false;
  while not(finished_coordination) do {coordination step}
    solve equation  $\frac{t}{M} \sum_{m=1}^M \frac{\theta}{\theta - f_m(x)} = 1$  for  $\theta$ ;
     $p_m := \frac{t}{M} \frac{\theta}{\theta - f_m(x)}$ ;
     $\hat{x} := ABS(p, t, r)$ ;
     $\nu := \frac{p^T f(x) - p^T f(\hat{x})}{p^T f(x) + p^T f(\hat{x})}$ ;
    if ( $\nu \leq t$  or  $\lambda(x) \leq \omega \bar{\lambda}$ ) then
      finished_coordination := true;
    else
       $\tau := \frac{t\nu}{2M(p^T f(x) + p^T f(\hat{x}))}$ ;
       $x := (1 - \tau)x + \tau\hat{x}$ ;
    end
  end
if ( $\sigma \leq \varepsilon$ ) then
  finished_scaling := true;
else
   $\sigma := \sigma/2, t := \sigma/6, \omega := (1 + \sigma)/(1 + 2\sigma), \bar{\lambda} := \lambda(x)$ ;
end
end

```

Table 1: Algorithm \mathcal{L} for convex min-max resource sharing problems.