VLSI DESIGN
1998, Vol. 7, No. 1, pp. 59–72
Reprints available directly from the publisher
Photocopying permitted by license only

Power Distribution Synthesis for VLSI

ASHOK VITTAL and MALGORZATA MAREK-SADOWSKA*

Department of Electrical and Computer Engineering, University of California, Santa Barbara, 93106

The synthesis of the power distribution network is an important problem in the layout design of VLSI systems. In this paper we propose novel methods to solve the problem of designing minimal area power distribution nets, while satisfying voltage drop and electromigration constraints. We will see that our methods significantly improve upon current techniques. We propose two novel greedy heuristics for power net design-one based on bottom-up tree construction using greedy merging and the other based on top-down linearly separable partitioning. We test the efficacy of our techniques on benchmark instances. The areas required by our methods on typical instances are significantly smaller than those obtained using previous methods.

Keywords: Power and ground routing, P/G routing, routing, IC layout

1. INTRODUCTION

The power distribution network supplies reference voltages to the circuit elements of an electronic system. It is designed after the placement phase of layout design. Thus the positions of the devices and their circuit functionality are known. As the power distribution net is one of the largest nets to be routed and subsequent nets are routed with it as a blockage, the area it occupies is of concern. Good power distribution design is also essential as it affects the reliability and speed of the system. The reliability is affected as the mean time to electromigration is determined by the current densities in the power lines. The speed is degraded if the voltage drop on the line is excessive so that the current drive of the devices is smaller than required.

Several trends make optimized power distribution synthesis imperative [20]. As technologies scale to the sub-micron regime, the interconnect resistance increases and the total current drawn by the system increases, so that voltage drops on the power lines increases. If the supply voltage is scaled down for low power as in portable applications, the noise margin is further reduced and the problem becomes increasingly important. With device counts increasing, the problem becomes harder as its scale is larger. While it is possible to alleviate the problem by increasing the number of metal layers, increasing the process complexity considerably lowers the yield. There is

*Corresponding author.

clearly a need for better algorithms for power distribution synthesis that exploit all the available degrees of freedom in order to reduce the area requirements, while ensuring correct circuit operation.

Early work on power distribution design centered around single layer routing of power and ground networks [18, 17, 23]. The emphasis was on connectivity, and given the small circuit complexities, electrical effects were not considered. Subsequently, electromigration and line voltage drop problems became severe enough to warrant optimization techniques. Algorithms were developed [3, 6, 14] to widen the wires to satisfy electrical constraints, while minimizing the area, *given an input topology*. The problem had been broken into two parts: one of finding a topology and the other of sizing it.

Topology optimization was recognized as a powerful tool in [16]. The approach was meant specifically for standard cell layouts. The central idea was to add enhancement buses of minimum area to pre-existing power buses to meet performance constraints. However, high performance systems-the systems in need of such optimization, are invariably full custom, so that unrestricted topologies can be much better than constrained ones. Nevertheless, this work demonstrated that topology design was an important degree of freedom-one whose use could lead to much better solutions.

Simultaneous topology optimization and wire sizing were considered in [21]. The entire problem was solved in one step using simulated annealing, but all topologies in the search space are sub-sets of a general grid. This is extremely restrictive and misses a large class of optimum solutions. In particular, when voltage drop constraints are very stringent, optimum solutions may have sinks fed by a single path from the pad. Thus, if two sinks are in the same channel, the optimum topology might run parallel lines to each sink. This is not allowed in the general grid formulation. The origin of this weakness in the formulation can be traced back to the use of simulated annealing as the optimization engine. If a simulator is to be used in an optimization loop, the run time required by the simulator should be small. In order to keep the simulation fast, parallel lines in the same channel are not allowed. This automatically forfeits the possibility of obtaining optimum star routes in a channel. Yet another shortcoming of the simulated annealing approach is the large computational expense. The time required for problem instances with tens of sinks is tens of hours. While the power supply routing problem is usually solved only once during layout design, the use of simulated annealing for large scale designs with tens of thousands of sinks is ruled out even if the problem were partitioned into hundreds of smaller sub-problems. Algorithms that exploit problem structure and quickly return good solutions are needed. While we have railed against simulated annealing, to be fair it should be pointed out that the approach is meant primarily for small analog power distribution problems where noise coupling is critical.

Another topology optimization procedure has been proposed in [8]. The procedure starts with a sized power distribution *mesh* and returns a sized power distribution *tree* which satisfies the same electrical constraints while using less area. This shows that area minimal topologies are typically trees, so that fast heuristics that synthesize area minimal trees are potentially useful. We will see that this turns out to be the case.

Section II outlines our problem formulation and discusses many insights into the structure of the problem. We observe that the problem is closely related to other layout problems like timing driven routing and clock routing, enabling the re-use of many ideas. We will also see why present techniques for power net routing are insufficient.

The use of information about the times during which sinks draw currents has been omitted from previous research on power distribution design. Most formulations either accept the worst case currents of the sinks or require simulation over all patterns of interest. While analysis tools for power distribution networks are more sophisticated [15,

25, 2] all previous works on synthesis [3, 21] lose important temporal information in the formulation. In Section III, we will survey the current estimation techniques in use during synthesis and look at better ways of representing the necessary temporal information so that design methods can utilize it. This leads to two powerful representations – the current compatibility graph and current interval sets. We will see that in general, the current compatibility graph representation leads to intractable problems when used for design. However, important special cases lead to simple and elegant means to its use during design. When more temporal information is available, current interval sets can be applied.

Section IV outlines two heuristics for the problem: a bottom-up greedy merging construction and a top-down partitioning-based divide-and-conquer strategy. Section V reports the performance of these heuristics as compared to previous methods. Finally, Seciton VI concludes with directions for future work.

2. PROBLEM FORMULATION

We wish to design minimal area, sized power supply net, given only the positions of the sinks, currents drawn at sinks, temporal sink current information, source pad position, and technology parameters. The temporal sink current information will be discussed in detail in Section III. It represents information about the times at which sinks draw current from the power net.

Thus we state the Power Tree Construction (PTC) problem as: Given $P = \{p_1, p_2, \ldots, p_n\}$ a set of sink positions on the Manhattan plane, the interconnect resistance per unit length R_0, the source pad position, the maximum allowed voltage drop V_{vm} from the pad to each sink (the vertical voltage drop constraint), the maximum voltage drop difference V_{hm} between any pair of sinks (the horizontal constraint), the peak sink currents $\{I_i\}$, temporal information regarding the times at which the sinks draw currents and the maximum allowed current density $J_{\max}$,

$$Minimize \sum_{j \in B} w_j \bullet l_j$$

where B is the set of all branches of a tree whose topology is to be determined, w_j and l_j are the width and length of a branch j. The cost function is therefore the area of the power distribution net.

The electromigration constraints are given by formula (1):

$$\left(w_j \geq \frac{I_j}{J_{\max}} \right) \forall j \tag{1}$$

This means that the wire width of each tree branch should be large enough to keep the estimated maximum current density within limits.

The vertical voltage drop constraints for each leaf node (sink) i are as follows:

$$\left(\left(\sum_{j \in P_i} I_{j\max} R_0 \frac{l_j}{w_j} \leq V_{\max} \right) \forall i \right) \tag{2}$$

In (2) P_i is the unique path from the root (pad) of the tree to the i^{th} sink, R_0 is the resistance per unit length and $I_{j\max}$ is the current in the sub-tree downstream of the branch j.

The vertical voltage drop constraints are imposed so that the current drive of the sinks does not fall below prescribed limits.

The horizontal voltage drop constraints impose the need for bounding the difference between reference voltages seen by communicating circuit elements. For CMOS, they are implied by the vertical voltage drop constraints [3], as the minimum non-zero current drawn by a sub-tree is very small. However, this may not be the case for custom ECL circuits which essentially draw large, constant currents. In this case, optimization is possible if the horizontal drop constraints are stringent and the vertical constraints are not.

The decision version of the PTC problem is NP-complete by restriction to the minimum Steiner tree problem, which is NP-complete [9]. Without

electromigration and voltage drop constraints, the problem is one of finding a Steiner tree of minimum cost – and NP-complete problem. This means that efficient exact solution in polynomial time is unlikely and leads to the quest for good polynomial time approximation algorithms.

We need to see if standard signal routing techniques such as minimum Steiner tree routing or bounded-radius bounded-cost trees are sufficient for our problem.

Figure 1 shows a small example with three sinks A, B and C. The pad position is indicated by the square. The instance also gives us the times at which the sinks draw current – A and B draw current at the same time, followed by the sink C. For this instance neither a minimum Steiner tree nor a bounded-radius bounded cost tree is optimum in terms of area. The geometric formulations simply do not consider some of the available electrical information. Note that critical temporal information is used to avoid sub-optimal design. This motivates the need for good representation schemes for temporal information, which we study next.

3. BRANCH CURRENT ESTIMATION

The currents flowing in the branches of a power distribution network determine the wire widths necessary for satisfying electromigration and voltage drop constraints. It is therefore important to be able to accurately estimate the worst case currents in any branch. Previous synthesis approaches typically assume that the current in a branch of a tree is just the sum of the leaf currents. This is a very poor estimate and the artificial nature of this formulation has been recognized [6]. In this section we propose a remedy to this problem. First we will discuss a graph-theoretic representation for sink current temporal information. We then study computational complexity issues related to finding good branch current estimates using this information. We will also consider representation of currents as a set of time intervals corresponding to the time at which current is drawn.

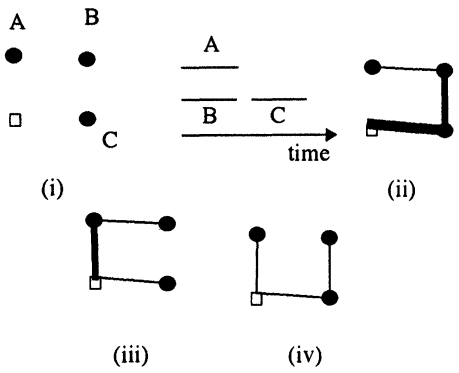

FIGURE 1 Insufficiency of geometric formulations. i) A problem instance with 3 sinks. ii) A minimum Steiner tree. iii) A bounded-radius bounded cost tree. iv) Best design.

We call two sinks *current-compatible* if the sinks never draw current at the same time. This relation is used to define a *current-compatibility graph G* whose vertex set corresponds to the set of sinks and whose edge set consists of current-compatible sink pairs. Note that information about disjoint temporal utilization of functional units is not difficult to obtain, especially from a high-level synthesis system. In a self-timed system, there is a partial order on the events occurring in the system [5]. This might lead to some sub-systems never being activated at the same time as others. In domino logic, a stage evaluates only after its predecessor is done. In a Boolean network, a zeroth order delay model would give gate level as the time instant at which it draws current. Thus, such temporal information can be obtained not only by simulation, but also by straightforward structural analysis.

The computational complexity associated with calculating the branch currents using this graph is of interest. Consider the homogeneous case where all sinks draw the same current. We define

[WEIGHTED CLIQUE COVER]
INSTANCE: Graph $G=(V,E)$, integer $K<|V|+1$
QUESTION: Can V be partitioned into disjoint sub-sets $V_1,V_2,\ldots,V_k$ such that the sub-graph induced by each V_i is a clique and $\Sigma_i|V_i| - 1 \geq K$.

We call the maximum such gain attainable the weight of G and denote as $W(G)$. The intuition behind this problem is as follows. If we have a clique of sizen n as the current compatibility graph, the current in the root of the tree is that of a single sink – the worst case current estimate has been reduced from n to 1 indicating a gain of $n-1$ in terms of the current bound. Clearly, if we have k cliques of size $V_1, V_2, \ldots, V_k$, for each V_i we get a gain of V_i-1 in the current bound. Thus maximum weight corresponds to a maximum gain in the current bound. Conversely, if the current compatibility graph is not a clique of size n, we cannot get a gain of $n-1$ as this would mean that only one of the sinks is conducting at a given time – a contradiction. Thus, the bound is tight. Theorem 1 formalizes this argument – the branch current estimation problem is translated into a graph theoretic problem.

THEOREM 1 *The worst case current drawn by the root of a tree with a current compatibility graph $G(V, E)$ is given by*

$$I_{\max} = \sum_{i \in V} I_i - W(G)$$

The computational complexity of finding this partitioning is given by the following theorem.

THEOREM 2 [WEIGHTED CLIQUE COVER] *is NP-complete.*

Proof The problem is in NP as we can test each of the sub-graphs for the presence of all edges in time polynomial in the number of vertices and the number of such sub-graphs is linear in vertex set size. The transformation from [EXACT COVER BY 3-SETS (X3C)] to [PARTITION INTO TRIANGLES] given in [10] can be used for this problem as well. The statement of the X3C problem is as follows.

[X3C]
INSTANCE: A finite set X with $|X| = 3q$ and a collection C of 3-element subsets of X.
QUESTION: Does C contain an exact (pair wise disjoint) cover for X.

Figure 2 below shows the local replacement for each clause $c_i = (x_i, y_i, z_i)$ in the [X3C] instance to the [WEIGHTED CLIQUE COVER] problem. Note that if a clause is in the set cover, we can get a gain corresponding to 4 triangles in the [WEIGHTED CLIQUE COVER] problem. If the clause is not in the set cover we get a gain corresponding to 3 triangles. Thus, the [X3C] instance has an exact cover if and only if the weighted clique cover instance has a weight of at least $2*(q + 3C)$, where C is the number of clauses and $3q$ is the size of X. In essence, the [WEIGHTED CLIQUE COVER] problem for any transformation from [X3C] is equivalent to the [PARTITION INTO TRIANGLES] problem as there are no cliques of size greater than 4 and we are done. □

The NP-completeness result motivates the search for polynomial time estimates that are only a constant factor above optimal. This is in fact possible as proved in Theorem 3 below.

THEOREM 3 *The weight of the graph $W(G)$ is at most twice that of any maximal matching on the graph G.*

Proof Consider an optimum partitioning $V_1, V_2, \ldots, V_k$, where each V_i induces a clique in G. Clearly, the maximal matching on G has at least $\lfloor |V_i|/2 \rfloor$ edges contributed by each such a clique else it is not maximal. The weight of G is sum of the weights of the k cliques each of which

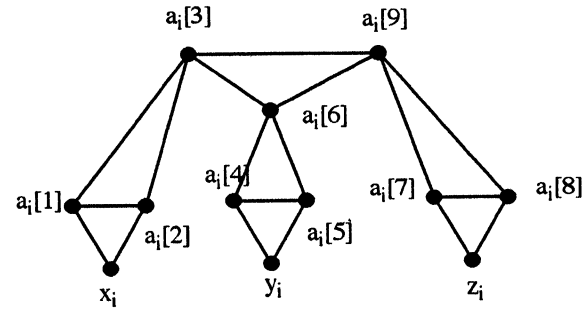

FIGURE 2 Local replacement for $c_i = (x_i, y_i, z_i)$ for transforming X3C to [Weighted Clique cover].

contribute at least half their size $|V_i|$ to the matching. The weight of G is therefore at most twice that of the matching. □

The bound above is tight. For a triangle, the weight is 2, while the matching is a single edge giving a weight of 1. Maximal matching on a graph with V vertices can be determined in $O(V)$ time using a simple scan algorithm – add an edge from the next scanned vertex and skip the adjacent vertex during the scan.

The current-compatibility graph carries information only about pair wise relationships between sinks regarding the current drawn, but not about the actual individual sink current waveform envelopes (worst case sink current drawn during a clock period). This graph representation is appropriate when little is known about the exact temporal behavior. However, an alternative representation is possible which captures information about the actual time instants at which sinks draw current. Note that in the context of design flow, we are past the placement phase but signal routing has not been done. Voltage-controlled circuits draw currents when they drive their respective loads, so that sub-intervals of the clock period when there are demands on the power net can be determined. Such interval information would use bounds on delays of nets [19] as placement is done, but not routing. We explain this with an example.

Figure 3 shows a netlist and the intervals for each sink. The sink correspond to the gates in the netlist. Assuming that each of the inputs is available at the same time and the levels of the gates give the time at which the gates draw currents, the intervals for the gates are shown. If each gate draws one unit of current, the bound on the sink current is 2, while the sum of currents is 3. Note that the use of temporal information not only leads to better wire sizing, but also allows for better topology design.

If each sink is associated with a set of intervals corresponding to the time instants at which it draws current, the following procedure gets the best current estimate.

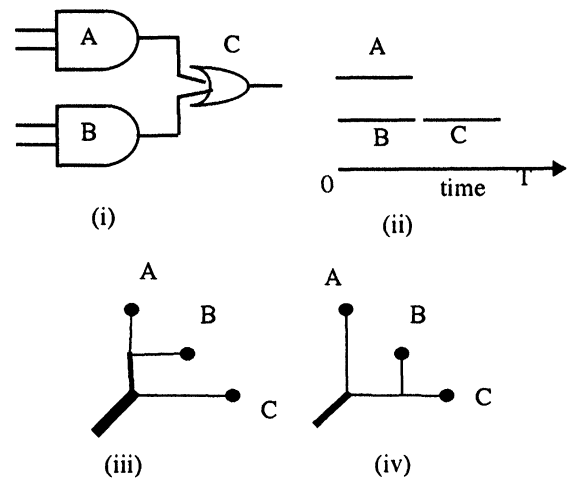

FIGURE 3 The use of sink current temporal information in design. i) The netlist. ii) The sink current intervals. iii) Design without temporal inforamation. iv) Best design.

```
Current Estimation

Input: A family F of sets of time intervals
Output: The worst case current estimate

begin
   For each interval in F
      increment bins corresponding to times contained in the interval
   worst case estimate = size of the largest bin
end
```

FIGURE 4 Estimating the worst case currents given a family of time interval sets.

For simplicity, the time intervals are assumed to be drawn from a small finite set, as this is usually the case. This means that we can have a bin for each time instant between 0 and T (the clock period).

Example Let the family of interval sets consist of three sets $\{[0,1],[3,4]\}$, $\{[2,4]\}$ and $\{[0,2],[3,4]\}$. In our current estimation problem this means there are three sinks that draw a unit of current at the given times during a clock period. Four bins corresponding to unit times $[0,1]$ to $[3,4]$ are initialized. After the loop, the first bin has two units in it, the second has 1, the third has 1 and the fourth has 3. The worst case current is 3. During

the time interval [3, 4], three units of current are drawn from the root (pad) of the tree. The generalization to the case of unequal sink currents is straightforward – we increment the bin by the amount of the corresponding current. The intervals could also be over an arbitrary set and our procedure would still work with minor changes.

4. GREEDY HEURISTICS

In this section, we propose two greedy heuristics for the problem. The first heuristic is a greedy bottom-up tree construction strategy which simultaneously designs the topology and sizes the wires. The second heuristic employs the divide-and-conquer approach based on linearly separable partitioning to solve the problem. First we discuss the characteristics of the problem and motivate the algorithms. Then we highlight some features of the heuristics.

Voltage drop at a sink i can be expressed by the formula given on the left hand side of inequality (2). Note that this expression is isomorphic to the one of calculating Elmore delay in the performance-driven interconnect design problem [4, 19]. The delay to the j^{th} sink when resistance per unit length is R_0, driver resistance is R_d, capacitance per unit length is C_0 and load capacitances is C_{Li} can be determined from the following formula:

$$D_i = \sum_{k \in P_i} C_k R_0 \frac{l_k}{w_k} + R_d C_0 L(T) + R_d \sum_{\forall i} C_{Li}$$

There, C_k is the downstream capacitance seen by a branch k (this is the sum of sub-tree load capacitances and sub-tree interconnect capacitance), $L(T)$ is the net length, l_k and w_k are the length and width of the branch k.

The expressions for voltage drop and sink delay are isomorphic if $C_0 = 0$ and $R_d = 0$ in the sink delay equation. The delay in performance-driven interconnect design corresponds to the voltage drop in our problem and the sink capacitances correspond to the sink currents. Wire sizing behavior is the same too – wire widening decreases the interconnect resistance in both cases. There are differences however – the load capacitances are constant while the load currents are time-varying. The delay expression has a term which is quadratic in the wire length. This is absent in the voltage drop expression.

The horizontal constraints are exactly the same as bounded skew constraints in clock routing [24]. Horizontal constraints require the difference between leaf voltages to be small, while bounded skew routing asks for delays to sinks to differ by at most a given constant. Therefore our topology design should reduce to the problem of clock routing when the only constraints are the horizontal voltage drops.

4.1. Bottom-Up Greedy Merging

The PTC problem appears to be closely related to that of the performance-driven interconnect design problem and to the clock routing – two problems which have seen considerable research over the past few years. A common thread that runs through both of the problems is the tremendous, almost unreasonable effectiveness of greedy methods. In clock routing, the greedy algorithm proposed in [7] returns remarkably small wire length. Similarly, in performance driven interconnect design, greedy methods [1, 4] has proven effective. It is therefore natural to expect a greedy algorithm to do well for power supply net routing too.

In this section we propose to build the power net in a bottom up fashion. The basic operation preformed by our greedy algorithm is the merging of sub-trees. Below, we discuss in detail the greedy merge operation.

Consider two sub-trees T_1 and T_2, rooted at positions Z_1 and Z_2 on the Manhattan plane as show in Figure 5. Let the maximum voltage drops from the sub-tree roots to any of the sinks be V_{M1} and V_{M2} respectively. Let their sub-tree currents be I_1 and I_2. We need to decide a merge point as the position of the new root. This is chosen to be

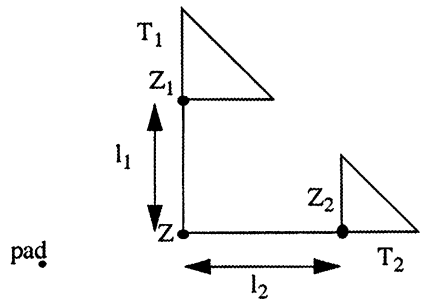

FIGURE 5 Merging sub-trees T_1 and T_2 to get a new sub-tree T.

the point on the periphery of the bounding box of the two sub-tree roots that is closest to the pad. This makes our tree an arborescence tree, using the terminology of [4]. The motivation for using arborescence trees is that the sum of sink to pad distances is minimized. This term appears in a lower bound on the sum of voltage drops to sinks.

The widths of the segments w_1 and w_2 are determined next. This is done using the wire sizing techniques of [14] for homogeneous vertical voltage drop constraints.

We have the electromigration constraints

$$\left(w_i \geq \frac{I_i}{J_{\max}}\right) i = 1,2$$

The voltage drop constraints are

$$\left(V_{\max} \geq V_{Mi} + I_i R_0 \frac{l_i}{w_i}\right) i = 1,2$$

If the minimum widths given by the electromigration constraints are insufficient for satisfying the voltage drop requirements, we size up the sub-tree by a factor α. This decreases the sub-tree voltage drop by a factor of α but increases the sub-tree area A_i, so that there is an optimum α for minimum area increase, while satisfying constraints [14]. The optimum α and w_i can be computed in $O(1)$ time for any pair of sub-trees. The new largest voltage drop to any sink can also be found in $O(1)$ time. Note that we size up sub-trees so that the width may be increased during subsequent merges.

Our heuristic chooses the sub-trees to merge, using a minimum area increase criterion. In other-words, we find the cost increase for the merge of each pair of current sub-trees and choose the merge that gives the smallest area increase. Following the merge, the current drawn by the new sub-tree is determined. This uses the temporal information provided by our formulation. Note that if a sub-tree formed during this process draws a lot of current, it would not be merged right away with other sub-trees, because the topology design uses the exact current information to make better-informed decisions. The algorithm is outlined below.

Note that the topology design is dynamically driven by wire sizing considerations, and therfore the wire sizing and topology design are determined simultaneously. At any stage of the above algorithm, we have sized sub-trees with exact information about the current they draw and the maximum voltage drop so far from the root of the sub-tree to any of its sinks. If the maximum voltage drop from the root to a sink of a sub-tree is approaching the maximum allowed, the area increase incurred by a merge with this sub-tree goes up, so that topolgy decisions are guided in the right direction.

```
Algorithm GM (Greedy Merging)
Input: Sink positions {Zi}, voltage drop & electromigration constraints, technology
Output: Sized topology of minimal area tree
-----------------------------------------------------------------------
begin
    initialize list of sub-trees to the list of sinks
    repeat{
        minimum_cost_merge()
    } n-1 times
end
-----------------------------------------------------------------------
minimum_cost_merge()
 begin
   find min{cost of sub-tree merges for all pairs of sub-trees}
       where cost = area increase due to the merge (as described in Figure 4)
   merge the sub-trees and append to the list
end
```

FIGURE 6 Greedy merging algorithm.

The time complexity of the greedy merging algorithm is $O(n^3)$ as we have n merges each of which is decided upon in $O(n^2)$ time. Each sub-tree is represented by its maximum voltage drop, its children set and the current. Each of these can be computed in linear time for a new sub-tree that merges two others. Thus $O(n^3)$ time is sufficient. In our experience, this greedy heuristic takes a few seconds for instances with a hundred sinks, so that the constants associated with this growth are not large. When combined with partitioning, this enables us to solve problem instances with tens of thousands of sinks in minutes as opposed to many days as required by competing approaches such as simulated annealing.

4.2. Top-Down Topology Design

If the tree is designed in a top-down manner, it is possible to ensure single-layer routable trees. This can be done by dividing the problem into two pieces whose convex hulls do not intersect so that the routing internal to the two pieces do not intersect. We also have to ensure that at each stage the wire segment connecting the roots of the two pieces to the pad does not intersect any of the other branches that are topologically lower down in the tree. This idea has been used for finding single-layer routable clock trees [13]. We use linearly separable partitions for dividing the problem into smaller pieces. A partition of a set of points is said to be linearly separable if there exists a line separating the points into two clusters. Linearly separable partitions naturally give us smaller problem instances whose convex hulls do not intersect. Besides, the number of such partitions grows at most quadratically with the number of points if no three of the points are collinear. We can clearly choose one such partition that results in the smallest estimated area. If we insist on arborescence tress, the root of each sub-tree corresponding to the sub-problems is just the point closest to the pad and contained in the bounding box (the smallest rectangle enclosing all pins). We therefore know the length of the top-level routing. The current in the branches going from the tree root to each of its children can be determined exactly as it depends solely on the children in each sub-tree. The only information that we do not have when deciding on the partition is the routing area that each sub-problem will require. Estimates are required for this and we approximate the unknown sub net area by a product the diameter of the bounding box containing all sinks of the sub tree and the square root of the sinks cardinality. This is motivated by the probabilistic results of [22]. Our top-down heuristic is outlined below. Note that it decides on the topology completely, before sizing the wires and therefore corresponds to conventional methods of choosing a topology first. The topology chosen, however, is cognizant of the electrical constraints that will be encountered later. Once again, note the pivotal role played by the current estimation which enables good partition cost evaluation.

The time complexity of this procedure depends on the size of the acceptable partitions. If we insist on balanced partitions, the time complexity is $O(n^3)$. This is determined as follows.

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + k \cdot n^3$$

There are two sub-problems of equal size and the sub-division takes cubic time because $O(n^2)$ possible partitions have to be evaluated and each evaluation takes linear time. However, if we allow partitions of arbitrary size, $O(n^4)$ will be the worst case time complexity as we might end up with a sub-problem of size $n-1$. Just as with the bottom-up construction, the constants associated with this growth are small so that this heuristic takes about a minute for an instance with 150 sinks.

5. RESULTS

We implemented our algorithm in C on a SPARC 5 workstation. We have compared our heuristics with the previous approach of starting off with a

```
Algorithm TD (Top-Down topology design)
Input: Sink positions, current information, electromigration &voltage drop
       constraints, technology information
Output:Power distribution topology

begin
   if only two sinks say l,r present then
      return binary tree as topology with l and r as children
   for each linearly separable partition (induced by a pair of points)
       find_cost(partition)
   choose partition (L,R) of smallest cost
   embed root at smallest x-, y-coordinate of all sinks in the sub-tree
    TD(L)
    TD(R)
end

find_cost(partition:L,R)
begin
return(I(L)*sqrt|L|*diameter(L) + I(R)*sqrt|R|*diameter(R) + root_branch_ area)
end
```

FIGURE 7 Top-Down topology design algorithm.

topology and sizing it to satisfy the constraints. The two topologies that we considered were the near-optimal minimum Steiner tree and the star topology. Choosing the topology and then sizing it represents the conventional solution to the problem [17]. We tested how well our heuristics performed in the first set of experiments. In the second set of experiments we checked how useful the time domain current information was, as represented by the current interval sets.

There are no benchmarks for power supply routing. We therefore introduce our own benchmarks, based on the widely available clock routing benchmarks from [24]. We derived 8 benchmarks from the benchmarks R1 and R2 introduced in [24]. A typical high performance design has tens of thousands of gates and close to a hundred power supply pads. The number of sinks per pad is therefore a few hundred. We chose each of R1 and R2 to have four supply pads at the four corners of the die. The assignments of pins to pads was done by a closest point heuristic, i.e., a pin was assigned to its closest pad – a reasonable partition of the problem into smaller sub-problems. The total current, i.e., the sum of sink currents was chosen to be 1A. The distribution of sink currents I_i was chosen to be proportional to the corresponding load capacitances C_{Li} which is a valid assumption since large capacitive loads occur when circuit elements are sized up to increase speed, therefore increasing current requirements. The die size for R1 and R2 were set to be 7.5 mm × 7.5 mm and 1 cm × 1 cm respectively. The number of sinks in each of our eight bench-marks are shown in Table I. The suffix of each name indicates which corner the pad is at. E.g., R1.LL represents the benchmark obtained from R1 with sinks closest to the lower left corner.

The results of our first set of experiments are shown in Table II.

TABLE I Benchmark sizes

Name	Number of sinks	Sum of sinks currents (A)
R1.LL	72	0.274
R1.LR	76	0.282
R1.UL	87	0.325
R1.UR	32	0.119
R2.LL	116	0.193
R2.LR	160	0.277
R2.UL	142	0.237
R2.UR	180	0.294

TABLE II Area Comparison

	50 mV				200 mV				500 mV			
	I1S	*	GM	TD	I1S	*	GM	TD	I1S	*	GM	TD
R1.LL	18.68	15.21	5.402	5.481	5.078	5.772	1.844	1.727	3.672	5.772	1.819	1.622
R1.LR	21.94	14.56	4.674	5.548	5.956	5.716	1.654	1.873	3.843	5.716	1.654	1.604
R1.UL	19.53	15.55	5.562	8.022	5.732	6.243	1.935	2.489	4.176	6.243	1.935	2.102
R1.UR	7.722	6.942	2.199	2.728	2.094	2.481	0.753	0.840	1.529	2.481	0.753	0.705
R2.LL	21.57	21.59	6.768	6.823	5.950	7.928	2.047	2.091	3.615	7.739	1.745	1.606
R2.LR	36.29	28.48	7.790	8.222	9.766	10.19	2.537	2.713	5.254	10.01	2.191	2.061
R2.UL	20.79	20.21	6.196	6.201	6.204	8.061	2.114	2.082	3.795	7.871	1.859	1.790
R2.UR	34.33	30.31	8.349	9.569	9.757	11.63	2.711	3.172	5.110	11.35	2.411	2.205

We compared the areas of sized Iterated 1-Steiner trees [12], sized star routing, our greedy merging algorithm and the top down heuristic. The run time for the largest example is about one minute for R2.LR on a SPARC 5 and grows cubically with the number of sinks on both our implementations. As the time required is reasonably small, we do not list the run-time requirements for all the benchmarks. Very large examples will have to be partitioned into several smaller sub-problems involving a single pad each. The size of these sub-problems will typically be in the range that we cover. The iterated 1-Steiner heuristic returns trees with small net length and would therefore be used by layout tools which use standard signal routing procedures for power distribution net topology design. The star route supplies each pin with an individual route from the pad. This topology was studied in [3] and the conclusion was that star routing could be competitive in row-based routing schemes used for standard cells. The resistance per unit grid is 1 milliohm and the maximum current density is 1 mA/micron^2. The time interval for current drawn by each sink was chosen to be the level in a Boolean network. The number of sinks at the highest levels is usually larger than those at smaller levels. The level is therefore randomly generated from a triangular probability distribution. The levels correspond to a tree-like circuit with 7 levels. Sizing was done using the techniques of [14]. We report areas and net lengths for three different vertical voltage drop constraints. The areas are in units of 10^6 grids (1 grid = 0.1 micron × 3 micron). The net lengths are shown in Table III. The lengths are in terms of 10^6 grid units (0.1 micron).

The net length comparison clearly shows that the iterated 1-Steiner trees have the smallest net lengths (the iterated 1-Steiner approach is one of the best heuristics for the minimum rectilinear Steiner tree problem) and the star routes the largest, as expected. The areas of the trees returned by the greedy merging procedure however are significantly better than the other algorithms-

TABLE III Net length comparison

Benchmark	Net length			
	I1S	STAR	GM	TD
R1.LL	0.241	3.371	0.446	0.542
R1.LR	0.233	3.483	0.425	0.495
R1.UL	0.252	3.809	0.423	0.568
R1.UR	0.150	1.601	0.220	0.261
R2.LL	0.356	7.739	0.670	0.770
R2.LR	0.418	10.01	0.753	0.968
R2.UL	0.415	7.871	0.750	0.965
R2.UR	0.424	11.35	0.813	0.967

mainly due to the simultaneous topology design and wire sizing.

Table IV lists the maximum current drawn from the pad for each case. Note that this is much smaller than the sum of the sink currents. Thus, using current intervals does indeed translate into better estimates and smaller areas.

In the next set of experiments we test how the granularity of the temporal information affects our results. Recall that each sink could draw at any of 7 time instants. Clearly, if our delay estimation capabilities in the form of net delay bounds are not so precise, there is more uncertainty in the time interval during which a circuit element draws current. In other words, the granularity of the interval will be larger. Note that standard methods of using the sum of sink currents for a branch current is attained when the time interval spans the entire clock period. In Table V we show how reduction of temporal information granularity affects the results. There, we compared the top-down and bottom-up greedy heuristics with different number of levels, for a 200mV vertical constraint. The corresponding root currents are shown in Table VI.

We see that the improved current estimates influence significantly the areas of power distribution trees.

6. CONCLUSIONS AND FUTURE WORK

We have proposed two new heuristics for synthesizing the power distribution network. These heuristics use temporal information about the sinks to obtain net whose areas are much smaller than those obtained by the previous methods.

Several issues remain to be solved. While we have developed greedy heuristics, no exact exponential-time algorithm has been proposed. Note that the Hanan grid [11], which gives the positions of optimum internal nodes of the tree for rectilinear minimum Steiner trees, is no longer sufficient for the PTC problem. This can be seen considering an instance with only a horizontal constraint that requires the internal node to be

TABLE IV Peak pad currents

Benchmark	Using temporal information (in mA)	Sum of sink currents (in mA)
R1.LL	70.4	274
R1.LR	68.1	282
R1.UL	89.0	325
R1.UR	29.6	119
R2.LL	51.8	193
R2.LR	70.6	277
R2.UL	60.3	237
R2.UR	77.3	294

TABLE VI Improvement in root current estimate (mA)

	2 levels	4 levels	7 levels
R1.LL	147	107	70.4
R1.LR	160	112	68.1
R1.UL	188	139	89.0
R1.UR	63.8	53.3	29.6
R2.LL	120	76.0	51.8
R2.LR	177	116	70.6
R2.UL	147	93.4	60.3
R2.UR	184	119	77.3

TABLE V Effect of smaller temporal information granularity

Benchmark	2 levels		4 levels		7 levels	
	GM	TD	GM	TD	GM	TD
R1.LL	3.180	3.103	2.362	2.020	1.844	1.727
R1.LR	3.022	2.850	2.196	2.614	1.654	1.873
R1.UL	3.700	3.680	2.517	2.598	1.935	2.489
R1.UR	1.372	2.132	1.118	1.000	0.753	0.840
R2.LL	3.820	4.832	2.655	2.954	2.047	2.091
R2.LR	4.973	6.627	3.628	4.284	2.537	2.713
R2.UL	3.617	3.623	2.755	3.453	2.114	2.082
R2.UR	4.839	5.650	3.586	4.063	2.711	3.172

outside the Hanan grid. Thus, it is of interest to be able to find characterizations of optimum solutions. It is also of interest to generate solutions in polynomial time with performance guarantees, i.e., the area is at most a constant factor away from optimal. Our work has been motivated primarily by practical applications and our approach seems reasonable for all test cases seen.

Acknowlegements

We would like to thank Professor T. Gonzalez of UC, Santa Barbara for useful discussions on the weighted clique cover problem. We would also like to thank Dr. Ken Boese and Professor Andrew Kahng of the University of California, Los Angeles for the use of their iterated 1-Steiner tree code. This work was supported in part by the Advanced Research Projects Agency under contract DABT 63-93-C-0039, the National Science Foundation under grant MIP 949119 and in part by Xilinx through the California MICRO Program.

References

[1] Boese, K. D., Kahng, A. B., McCoy, B. A. and Robins, G. (1994). "Rectilinear Steiner trees with minimum Elmore delay", *Proceedings of the Design Automation Conference*, pp. 381–386.

[2] Chowdhury, S. and Barkatullah, J. S., "Estimation of maximum currents in MOS IC logic circuits", *IEEE Transactions on CAD*, **9**(6), pp. 642–654, June 1990.

[3] Chowdhury, S. and Breuer, M. A., "Optimum design of IC power/ground nets subject to reliability constraints", *IEEE Transactions on CAD*, **7**(7), pp. 787–796, July 1998.

[4] Cong, J., Leung, K.-S. and Zhou, D. (1993). "Performance-driven interconnect design based on a distributed RC delay model", *Proceedings of the Design Automation Conference*, pp. 606–611.

[5] Conway, L. and Mead, C. (1980). *Introduction to VLSI Systems*, Addison-Wesley, Chapter 7.

[6] Dutta, R. and Marek-Sadowska, M. (1992). "Algorithm for wire sizing of power and ground networks in VLSI designs", *Journal of Circuits, Systems and Computers*, **2**(2), pp. 141–157.

[7] Edahiro, M. (1993). "A clustering based optimization algorithm in zero-skew routings", *Proceedings of the Design Automation Conference*, pp. 612–616.

[8] Erhard, K. -H., Johannes, F. M. and Dachauer, R. (1992). "Topology optimization techniques for power/ground networks in VLSI", *Proceeding of the European Design Automation Conference*, pp. 362–367.

[9] Garey, M. R. and Johnson, D. S. (1997). "The rectilinear Steiner tree problem is NP-complete", *SIAM Journal of Applied Mathematics*, **32**(4), pp. 826–834.

[10] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability – a guide to the theory of NP-completeness*, W. H. Freeman and Co., pp. 68–69.

[11] Hanan, M. (1966). "On Steiner's problem with rectilinear distance", *SIAM Journal of Applied Mathematics*", **14**, pp. 255–265.

[12] Kahng, A. B. and Robins, G. (1990). "A new class of Steiner tree heuristics with good performance: the iterated 1-Steiner approach", *Digest of Technical Papers of the International Conference on Computer Aided Design*, pp. 428–431.

[13] Kahng, A. B. and Tsao, C. -W. A. (1994). "Low cost single layer routable clock trees with exact zero Elmore delay skew", *Digest of Technical Papers of the International Conference on Computer Aided Design*, pp. 213–218.

[14] Kolla, R., "A dynamic programming approach to the power supply net sizing problem", *Proceedings of the European Design Automation Conference*, pp. 600–604, March 1990.

[15] Kriplani, H., Najm, F. and Hajj, I. (1992). "Maximum current estimation in CMOS circuits", *Proceedings of the Design Automation Conference*, pp. 2–7.

[16] Mitsuhashi, T. and Kuh, E. S. (1992). "Power and ground network topology optimization for cell-based VLSIs", *Proceedings of the Design Automation Conference*, pp. 524–529.

[17] Moulton, A. S. (1983). "Laying the power and ground wires on a VLSI chip", *Proceedings of the Design Automation Conference*, pp. 754–755.

[18] Rothermel, H. -J. and Mlynski, D. A. (1981). "Computation of power and ground supply nets in VLSI layout", *Proceedings of the Design Automation Conference*, pp. 37–47.

[19] Rubinstein, J., Penfield, P. Jr. and Horowitz, M. A. (1983). "Signal delay in RC tree networks", *IEEE Transactions on CAD*, **2**(3), pp. 202–210.

[20] Song, W. S. and Glasser, L. A., "Power distribution techniques for VLSI circuits", *IEEE Journal of Solid State Circuits*, **21**(1), pp. 150–156, February 1986.

[21] Stanisic, B. R. (1993). "Automatic analog power distribution synthesis in RAIL", Ph.D. thesis, Carnegie Mellon University.

[22] Steele, J. M. (1988). "Grwoth rates of Euclidean minimal Steiner trees with power weighted edges", *Annals of Probability*, **16**, pp. 1767–1787.

[23] Syed, Z. A. and El Gamal, A (1982). "Single layer routing of power and ground networks in integrated circuits", *Journal of Digital Systems*, **6**(1), pp. 53–63.

[24] Tsay, R.-S. (1993). An exact zero-skew clock routing algorithm", *IEEE Transactions on CAD*, **12**, pp. 242–249.

[25] Vanoostende, P., Six, P. and de Man, H. J. (1993). "PRITI: estimation of maximal currents and current derivatives in complex CMOS circuits using activity waveforms", *Proceedings of the European Design Automation Conference*, pp. 347–353.

Authors' Biographies

Ashok Vittal received the Bachelor of Technology degree in electrical and electronic engineering from the Indian Institute of technology, Madras in 1992 and the M.S. degree in computer engineering from the University of California, Santa Barbara in 1993. He is currently a Ph.D. candidate at UCSB and is working on layout synthesis for high speed systems. In 1995 he was a summer intern at Interconnectix, Portland, OR, and worked on interconnect design for high speed packages.

Malgorzata Marek-Sadowska received the M.S. degree in applied mathematics (1971) and the Ph.D. degree in electrical engineering (1976) from Politechnika Warszawska (Technical University of Warsaw), Poland. From 1976 to 1982, she was an Assistant Professor at the Institute of Electron Technology at the Technical University of Warsaw, and was a Visiting Professor in the Electrical Engineering Department of the University of California at Berkeley from 1979 to 1980. She became a Research Engineer at the Electronics Research Laboratory in 1979 and then joined the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, as a Professor in 1990. From 1993 to 1995 she served as Editor-In-Chief of IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Her research interests are in the area of computer-aided design with an emphasis on layout and logic synthesis of VLSI circuits and systems.