International Journal of High Speed Electronics and Systems, Vol. 6, No. 3 (1995) 459–475

PIN REDISTRIBUTION PROBLEM FOR MULTI-CHIP MODULES: ALGORITHMS AND COMPLEXITY*

DOUGLAS CHANG† and TEOFILO F. GONZALEZ
Department of Computer Science
University of California, Santa Barbara, CA 93106, USA
e-mail: {dchang,teo}@cs.ucsb.edu

We investigate the pin redistribution problem (PRP) for multi-chip modules. We transform the PRP to the max-flow problem and obtain an efficient algorithm for finding a 2-layer solution, whenever one exists. A greedy heuristic to find a k-layer solution is described. Our approach can also construct a minimum layer solution for two variants; nets can be routed on more than one layer, and terminals (source and target) are drilled through all layers. Our algorithms take $O(min\{|S|, mk^{1/2}\}m^2k)$ time, except for the heuristic procedure which takes $O(km^4 \log^2 m)$ time, where S is the set of source terminals, m is the number of rows and columns in the grid, and k is the number of layers required. Several variations of the PRP when generalized to graphs can also be solved efficiently by our algorithms, whereas other variations are shown to be NP-complete.

1. Introduction

The packaging between computer chips has become a greater factor in system performance as chip speeds have increased. Fifty percent of the delay in high-performance systems can be attributed to packaging, and this is likely to increase in the future.[2] Multi-Chip Modules (MCMs) have been introduced to reduce inter-chip delay by removing one layer of packaging. This improves system performance and reliability. In MCM technology,[14] bare chips are placed on a common substrate called the *chip layer*. Directly below the chip layer there are a number of *pin redistribution layers*, and below them there are the *signal distribution layers* (see Fig. 1). Some MCMs use the bottom signal distribution layer as a power distribution layer. For simplicity, we omit this special layer, but our algorithms can be easily adapted to handle this situation. The pin redistribution layers are used to redistribute the chips' I/O pins to a set of pins with a minimum spacing, as required by the signal distribution layers. This redistribution can also be used to spread the pins uniformly over the MCM, which leads to fewer signal distribution layers, fewer vias, and minimal crosstalk.[4] Lastly, the signal distribution layers are used to connect the appropriate (redistributed) chip I/O pins.

*A preliminary version of this paper appeared in the *Proceedings of the Fourth Great Lakes Symposium on VLSI*, March 1994, pp. 114–119.
†Douglas Chang was partially supported by NSF Grant CCR-8918409.

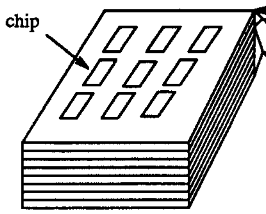


Fig. 1. Multi-chip module.

An early example of MCM technology is IBM's ceramic multichip technology, used in the IBM 3081 processor in the late '70s.[3] More recent examples are IBM's glass-ceramic/copper module for the System 390/9000 and DEC's multilevel thin film for the VAX 9000.[16] A detailed discussion of various MCM technologies is given in Ref. 15.

In this paper, we investigate the k-layer pin redistribution problem (PRP). The k-layer PRP is to connect (redistribute) the (source) I/O pins on the chip layer to target locations on the bottom redistribution layer, using k redistribution layers. A wiring with the minimum number of redistribution layers (i.e., the optimization version of the PRP) can be obtained by solving a set of k-layer problems. Since an algorithm for one of these problems can be easily derived from an algorithm to solve the other problem, we will refer to both of these problems as the PRP.

1.1. *The pin redistribution problem*

The basic model used is the k-layer routing model as described in Ref. 11. In this model, the routing graph consists of k stacked grid graphs (each representing one layer). The grid graphs (or layers) are numbered in increasing order from top (1) to bottom (k). Each edge in the graph can accommodate one wire segment. Vertical vias are available at a set of grid intersection points.

Given a set of source terminals on grid 1 (the top grid) and a set of target terminals on grid k (the bottom grid), the PRP is to connect each source terminal to a different target terminal by a wire in *only one* layer (grid) such that no two wires on the same layer (grid) intersect. Note that a source terminal does not have to be connected to a specific target terminal; it just has to be connected to some target terminal. This is the main difference between the PRP and conventional two pin routing problems.

A source (target) grid point is defined as an (r, c) grid point where there is a source (target) terminal on grid $1(k)$. Note that the only vias that can be used are at the source and target grid points, and these vias can only be used by the corresponding source or target terminal. This is because each net has to be routed on only one layer. Another observation is that if the net for source (target) terminal (r, c) is routed in layer i then the grid point (r, c) in any layer $j > i$ $(j < i)$ can be used to route another net in layer j. However, the grid point (r, c) in any layer

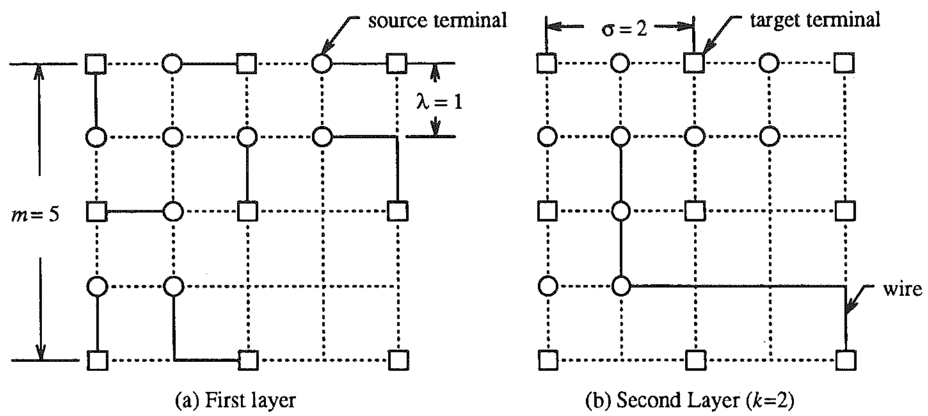


Fig. 2. PRP instance.

$j \leq i$ ($j \geq i$) cannot be used by another net. An instance and solution of a PRP is shown in Fig. 2.

We also consider two variations of the PRP. In one variation nets are allowed to be routed on more than one layer, and vias are allowed at any given subset of grid points. Under this model, stack vias are allowed. The main advantage of this model is that the number of layers in an optimal solution is less than or equal to the optimal number of layers of the corresponding problem under the original model. However, there may be a larger number of vias. The second variation of the PRP is when source and/or target terminals are drilled through all the redistribution layers. This routing model is similar to the one used in printed circuited board (PCB) technologies. For instances that have a solution under this model, a minimum layer solution also has the minimum number of vias. However, the number of layers may be larger than in an optimal solution to the corresponding problem under the original model. In the following sections we discuss computational complexity aspects of the PRP under these different routing models.

1.2. *Previous work*

Cho and Sarrafzadeh[5] introduce and formalize the PRP. In their formulation, the PRP is a 6-tuple, (k, m, S, T, λ, σ) (see Fig. 2), where

1. k is the number of layers available for pin redistribution, including the chip layer.
2. m is the number of rows and columns in the grid.
3. S is the set of grid points on the chip layer where the source terminals are located.
4. T is the set of grid points on the bottom redistribution layer where the target terminals are located.

5. λ is the minimum legal distance between two parallel wires on the same layer.
6. target terminals are spread uniformly over grid k at a horizontal and vertical distance of σ.

A solution to the PRP is a wiring connecting all the source terminals to the target terminals in the k-layer grid, such that each net is wired on one layer, no two wires intersect on any layer, and the minimum distance λ between two parallel wires on the same layer is maintained. We assume $|S| \leq |T|$, since there is no solution when $|S| > |T|$. Our PRP formulation differs in two ways: (1) target terminals cannot be placed in adjacent grid points (less restricted than the above model), (2) the value of λ is equal to 1, and (3) in our routing model if the net for source (target) terminal (r, c) is routed in layer i then the grid point (r, c) in any layer $j > i$ $(j < i)$ can be used to route another net in layer j, whereas in their model an additional constraint is imposed.

Cho and Sarrafzadeh[5] present three heuristics to solve the PRP. Their first heuristic is based on concurrent maze routing. The other two heuristics are based on finding a global routing, and then performing the detailed routing. They also show that given a special type of global routing with density two (at most two wires can be assigned to each grid point), a 2-layer solution can be found in polynomial time. However this special type of global routing does not always exist, so in the worse case, the routing area must be doubled in order to generate a 2-layer solution.

Shiao *et al.*[13] and McBride *et al.*[12] study a different type of pin redistribution problem on multi-chip modules. Rather than redistributing the pins to targets spread over a grid, they redistribute the pins over different signal layers. Thus the problem becomes a layer assignment problem.[10] Shiao *et al.* finds the center of mass of each net from its pin locations. They then calculate the induced force between two nets as a function inversely proportional to the square of the distance between the center of masses. Then they use a greedy algorithm to reduce the induced forces. McBride *et al.* are interested in the case where the source pins are in several rows. They find that redistributing pins in inner rows to lower layers, and pins in outer rows to higher layers facilitates routing. Thus they use a heuristic similar to the one by Shiao *et al.*, but also taking into account the pin row number.

The one layer PRP with $\lambda = 1$ in which all the target terminals are located on the boundary is called the *escape problem* in Ref. 6, pp. 625–626. Cormen, Leiserson and Rivest[6] solve the escape problem by reducing it to the max-flow problem. Our approach, developed independently, is a generalization of the one in Ref. 6. The PRP is reduced to the maximum flow (max-flow) problem, which can be solved efficiently.[8] In Sec. 2, we show that given the restriction of $\lambda = 1$, a 2-layer solution can be found quickly, whenever one exists. The time complexity of our 2-layer algorithm is $O(min\{|S|, m\}m^2)$. We then present a heuristic procedure, based on the 2-layer algorithm to find a suboptimal solution to the optimization version of the PRP. Note that in most practical cases a solution using at most three layers exists.[5] We also present an algorithm for the k-layer PRP when nets are allowed to be routed on more than one layer. Lastly, we show that if we restrict each source and target grid point to be used only by the wire connecting that source or target terminal, a k-layer solution can be found efficiently, whenever one exists. These algorithms take $O(min\{|S|, mk^{1/2}\}m^2k)$ time, except for the heuristic procedure which takes $O(km^4 \log^2 m)$ time.

The versions of the PRP just discussed can also be solved efficiently by our algorithm when the grid graph is replaced by an arbitrarily connected graph. We show in Sec. 3 that the PRP, when generalized to arbitrarily connected graphs, is an NP-complete problem even when the number of layers is three. We also show that the problem remains NP-complete even when the targets or sources (not both) are drilled through.

2. Flow Solution

An input to the max-flow problem is a directed graph G, called the flow graph, with two special nodes labeled s (source) and t (sink). Each arc in the flow graph has a positive real capacity associated with it. A feasible flow F is any assignment of flow values to each of the arcs in the flow graph such that the flow along each arc is between 0 and the flow capacity of the arc, and the flow at each node (other than the source and sink) is conserved (i.e., the flow into a node must equal the flow out of it). The max-flow problem consists of finding a maximum feasible flow from the source s to the sink t. There are a number of efficient algorithms to solve the max-flow problem.[1] It is well known that when all the capacities are integers, a maximum flow in which all the arcs have integer flows exists, and algorithms such as Ford and Fulkerson's and Dinic's generate such a flow.[8]

Let us now consider the following reduction from the k-layer PRP to the max-flow problem. We map the routing grid to a flow graph as follows. Each grid point is represented by a flow cell (see Fig. 3). Each flow cell has a subset of the *in* arcs (IN, IE, IS, IW), *out* arcs (ON, OE, OS, OW), *layer* arcs (IA, OB), and the *inner layer* arc F. All the arcs in this construction have capacity 1. A flow cell with all its arcs is shown in Fig. 3(a). Flow cells corresponding to adjacent grid points are connected as follows. Flow cell X immediately to the west of flow cell Y has a correspondence between arcs X.OE and Y.IW. There is also a correspondence between arcs X.IE and Y.OW. A similar arrangement holds for flow cells immediately to the east, north, and south, as shown in Fig. 3(b). The inner layer arc is present in every flow cell. The only flow cells with layer arcs are those representing source and target grid points (called source and target flow cells). If flow cells X and Y correspond to the same source or target grid point in layers i and $i+1$, respectively, then X.OB corresponds to Y.IA.

There are two additional nodes, the source s and the sink t. The IA arcs of all the source flow cells on grid 1 emanate from s, and all the OB arcs of the target flow cells on grid k end at t (see Fig. 4).

Let us discuss our flow algorithm that finds a 2-layer routing for the PRP whenever one exists. The algorithm takes as input an instance of the PRP and constructs a flow graph as shown above. It then finds a maximum flow using Dinic's algorithm. If the flow value is less than $|S|$, we claim that no 2-layer routing exists, otherwise a 2-layer routing can be constructed as follows. Because the capacities are integer, from max-flow theory we know there is always a flow with integer values on all the edges, and such a flow is obtained by Dinic's algorithms. This flow can be seen as consisting of a set of edge disjoint flow paths, and these flow paths can be easily translated to a routing of each one of the nets in exactly one of the two layers.

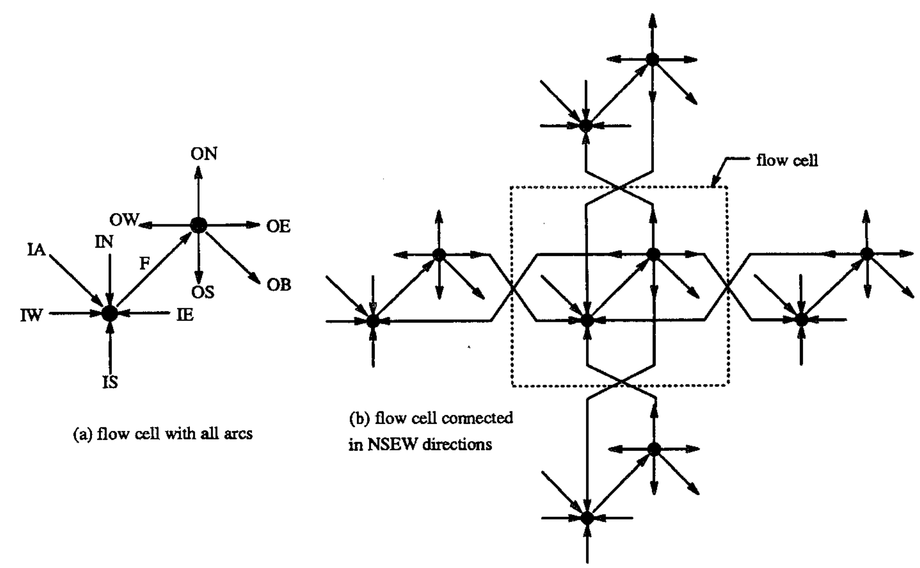


Fig. 3. Flow cell (all arcs have capacity 1).

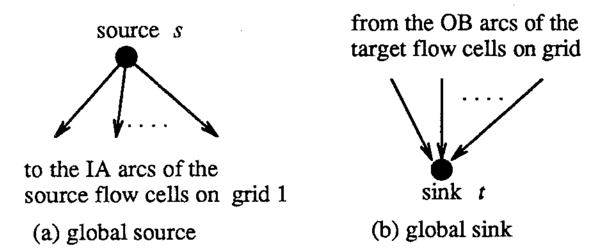


Fig. 4. Global source and sink (all arcs have capacity 1).

The following theorem shows that our algorithm finds in $O(min\{|S|, m\}m^2)$ time a 2-layer solution, whenever one exists.

Theorem 2.1: Our flow algorithm determines in $O(min\{|S|, m\}m^2)$ time whether or not any instance of the PRP is two layer routable. Furthermore, for PRP instances that have a 2-layer solution, our algorithm also constructs a wiring.

Proof: Suppose that a problem instance I of the PRP has a k-layer routing. We claim that the max-flow problem instance, $M(I)$, generated by our reduction for problem instance I has a maximum flow from source to sink equal to $|S|$. We prove this by showing that for any valid routing R, we can find a maximum flow from source to sink equal to $|S|$. Let R be any valid routing for I. Now consider net j routed on layer i in I. For this net we construct a flow of one unit from s to t (i.e., its flow path) as follows. Starting at the source s send a flow of one unit through the source flow cells for the net until you reach the flow cell in layer i. Then send a flow of one unit from that flow cell in layer i to the flow cell that represents the target grid point for net j along the path corresponding to the route followed by the wire connecting net j in layout R. Then from that target flow cell to the sink, a flow of one is sent that goes through only the corresponding target flow cells for the net. The flows for all the nets can be easily combined into a valid flow from s to t with value $|S|$, because they follow the same paths as the routing and no two wires in the routing intersect in any layer. Since there are at most $|S|$ arcs emanating from the source, the maximum flow is at most $|S|$. Therefore, if I has a k-layer routing, then $M(I)$ has a maximum flow equal to $|S|$.

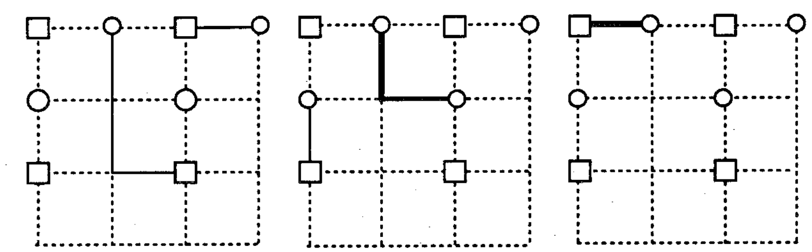
For the case when $k = 2$, the converse claim also holds. The reason for this is the following. The only illegal flow path that may be found is one that flows from s to a source cell on the first layer to a target cell on the first layer, to a target cell on the second layer, to a different target cell on the second layer, and finally flows to t. This would correspond to a routing in two layers, which is not allowed. However this flow path can be modified by rerouting it directly to the sink t when it reaches the second layer, thus converting the two layer flow path to a one layer flow path.

Because the capacities are integer, from max-flow theory we know there is always a flow with integer values on all the edges, and such a flow is obtained by all well known maximum flow algorithms. This flow can be seen as consisting of a set of edge disjoint flow paths, and these flow paths can be easily translated to a routing of each one of the nets in exactly one of the two layers.

Therefore, we claim that for the 2-layer case, there is a solution to the PRP if and only if the maximum flow is equal to the number of source terminals ($|S|$). Furthermore, the layout can easily be constructed from any maximum integer flow.

In the flow graph we have defined all arc capacities as one. For flow graphs of this type, Even and Tarjan[7] have shown that Dinic's max-flow algorithm has $O(min\{V^{2/3}, E^{1/2}\})$ phases, and each phase takes $O(E)$ time, where V is the number of nodes and E is the number of arcs in the flow graph. This leads to a time complexity of $O(min\{V^{2/3}E, E^{3/2}\})$. For a ($k$-layer) flow graph, V is $O(m^2k)$ and E is $O(m^2k)$ so for the 2-layer case (i.e., solution in at most 2 layers), we get the time complexity of $O(m^3)$. For our special type of flow graph we can establish another time bound. Since the maximum possible flow is $|S|$ and each phase increases the flow by at least one, we can bound the number of phases by $|S|$. Thus the time complexity of our 2-layer algorithm is $O(min\{|S|, m\}m^2)$. □

In the proof of Theorem 2.1, we showed that in the case $k = 2$, if there is a flow with value $|S|$ from source to sink in $M(I)$ then the PRP instance I has a routing. For $k > 2$, this does not hold, because the flow might imply a routing for a net in the PRP in more than one layer. An example of this is shown in Fig. 5. For this simple example there is an obvious 2-layer solution. For more complex examples (see Appendix) there are no 3-layer solutions, but an illegal 3-layer solution is found by the algorithm (i.e., some nets are routed in two or more layers).



(a) First layer (b) Second layer (c) Third layer

Fig. 5. Illegal wiring (thick line) that could occur with a 3-layer PRP.

We now present a heuristic algorithm to construct a routing for any instance of the PRP. Note that the number of layers k, is not necessarily minimum.

The heuristic begins by connecting the maximum number of nets in the first two layers, while allowing the unconnected source terminals to reach layer three. Then repeat the same operation for the unconnected nets in layers three and four, and so on. This can be done by modifying the previous flow graph to have the OB arcs of *both* the source and target flow cells on the second grid enter t. Then a cost of 1 is placed on each arc in the flow graph, except for the arcs from source flow cells to t, which get a cost of $22m^2 + 1$. This cost is larger than any set of paths from the source s to target flow cells to the sink t because each flow cell has at most 11 arcs and there are $2m^2$ flow cells. Therefore a minimum cost flow will include a minimum number of these arcs, which implies that a maximum number of nets are connected. Now the PRP has been transformed to the minimum cost flow problem which can be solved efficiently.[1] This gives the maximum number of connections on two layers, and source terminals not connected on the first two layers are available for connecting on lower layers. When we reapply the algorithm on lower layers the grid points of the target terminals that have been connected above are no longer available for wires, so their inner layer arc is removed.

Theorem 2.2: Our heuristic algorithm constructs in $O(km^4 \log^2 m)$ time a routing for any instance of the PRP, where k is the number of layers in the solution.

Proof: The proof that a feasible wiring is generated follows from the fact that the source terminals not connected on the first two layers are available for connecting on lower layers, the target terminals that have been connected above are no longer available for wiring, and the restriction that target terminals are not located in adjacent grid points. Note that with this last condition our heuristic algorithm will be able to generate a solution to all problem instances.

The time complexity of the minimum cost flow algorithm is $O(m^4 \log^2 m)$ for our flow graphs using Orlin's algorithm.[1] Since we need to repeat this $\lceil k/2 \rceil$ times, the algorithm has a total time complexity of $O(km^4 \log^2 m)$. Note that the solution does not necessarily have the minimum number of layers. □

2.1. *Variations of the PRP*

Two interesting variations of the PRP can also be solved by our flow strategy. One variation allows nets to be wired on more than one layer. A k-layer solution for this variation is found by the max-flow algorithm on our original flow graph construction. Note that the PRP solution in Fig. 5 would be a legal wiring under this variation. The flow graph can easily be modified to allow vias at locations other than the source and target grid points. If vias are allowed at grid point (r, c) then the corresponding (r, c) flow cell will have the layer arcs. Note that the solution allows stacked vias. Algorithm VPRP given below finds a minimum layer solution to this variation of the PRP.

```
Algorithm VPRP

begin
  create appropriate (depends on the PRP variation) flow graph for 1-layer PRP;
  apply Dinic's max-flow to the flow graph and let F be the flow value;
  while (F < |S|)
    begin
      add one layer to the bottom of the flow graph;
      extend flows that have already been found through added layer;
      apply Dinic's max-flow to the new flow graph and let F be the flow value;
    end
    output the flow paths found;
end
```

Theorem 2.3: Our VPRP algorithm constructs in $O(min\{|S|, mk^{1/2}\}m^2k)$ time a minimum k-layer solution to any PRP instance when nets are allowed to be routed on any number of layers.

Proof: It is simple to prove that the PRP on a line (1-dimensional PRP) always has a solution. Since the grid (2-dimensional) PRP can be traversed in a snake fashion, corresponding to a 1-D PRP, we know that the grid PRP always has a solution. Thus the VPRP algorithm always finds a solution because for large k the solution to the snake 1-D PRP can be generated. The algorithm finds a minimum k-layer solution because the max-flow algorithm insures the maximum number of nets are connected in the k layers.

Note that the flows found by Algorithm VPRP during iteration i are *not* discarded at iteration $i + 1$. Since the maximum flow is obtained by Dinic's algorithm, the total number of phases that are needed is $O(min\{mk^{1/2}, |S|\})$, and the size of the final flow graph is $O(m^2k)$. Therefore the time complexity bound is $O(min\{|S|, mk^{1/2}\}m^2k)$. □

It is worthwhile noting that algorithm VPRP finds k sequentially rather than in a binary search fashion. The reason for this is that for problem instance that require few layers, the procedure does not have to construct huge graphs for $k = |S|/2$ layers, and in the case of a sequential search one can use all previously computed flows at each iteration.

A second PRP variation is restricting each source and target grid point to be used only by the wire connecting the corresponding source or target terminal. In other words, the source and target terminals are drilled through all the layers. In this variation, we can also obtain a k-layer solution efficiently, whenever one exists. The flow graph for this variation is a modification of the original flow graph by removing the *in* arcs for each source flow cell, and removing the *out* arcs for each terminal flow cell.

Our algorithm MVPRP is a slight modification of algorithm VPRP. The modification needed is to test whether or not the flow value increases in each iteration. If the flow value does not increase and is less than $|S|$, then terminate without a solution. This test is needed because unlike the first variation, there is no solution to certain problem instances.

Theorem 2.4: Our MVPRP algorithm, determines in $O(min\{|S|, mk^{1/2}\}m^2k)$ time whether or not a k-layer solution to any PRP instance exists when all the sources and targets are drilled through. Furthermore, for PRP instances that have a solution, our algorithm can also construct a minimum layer solution.

Proof: It is simple to prove that a flow of $|S|$ can be achieved in the graph if and only if there is a k-layer layout, and that a minimum layer solution is found by MVPRP (if one exists) in time $O(min\{|S|, mk^{1/2}\}m^2k)$. □

3. Generalized PRP

Our algorithms can easily be adapted to handle the case when the grid has holes in it, where routing is not possible. This could occur, for example, if areas are reserved for routing power and ground. We now consider a more general PRP (GPRP) in which the grid graph is replaced by an arbitrarily connected graph. Our flow technique and algorithms can easily be adapted to handle this case. For the GPRP, we can obtain a 2-layer solution (whenever one exists), and we can obtain a

k-layer solution (whenever one exists) for the GPRP, under the two variations given in the conditions of Theorems 2.3 and 2.4.

In the following subsection, we show that the 3-layer GPRP is NP-complete. We also show that the 3-layer GPRP with drilled through sources or drilled through targets is NP-complete.

3.1. *Complexity of the GPRP*

We show that the 3-layer generalized PRP (GPRP) is NP-complete. This is shown by reducing the three-dimensional matching problem (3DM)[9] to it. The 3DM problem is defined as follows.

> *Instance*: Given $M \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets, each having h elements, and $|M| = m$.
>
> *Question*: Does M contain a matching, i.e., a subset $M' \subseteq M$ such that $|M'| = h$ and no two elements of M' agree in any coordinate?

Theorem 3.1: The 3-layer Generalized PRP (3GPRP) is NP-complete.

Proof: Since showing that 3GPRP is in NP is straight forward, we only show that 3DM α 3GPRP. We assume without loss of generality that there is a triple containing each element in X, Y, and Z, as otherwise the 3DM problem has a "no" solution.

Before presenting the reduction, we first describe two 3GPRP structures with special properties that will be used in the reduction. Secondly, we describe the sources, targets, and structures to be used in the 3GPRP instance that we generate from the 3DM instance. Then we show how these components are connected to create the 3GPRP instance. Lastly we show that there is a solution to the 3DM instance iff there is a solution to the 3GPRP instance, and show that the transformation takes polynomial time.

The FS1 graph structure is shown in Fig. 6(a). It has three sources and two targets. Thus at least one of the sources must be connected to a target through the edge labeled "out". Since there are only three layers one of those connections must be on layer 1. This is because there are three sources that need to be connected through target $T3$, so we must have each of these sources connected on different layers. If the source connected on layer 1 does not go through the "out" edge, then it must be connected to $T2$ or $T3$, but then at least one of the other two sources cannot be connected to any target. But in a solution to the 3GPRP all sources must be connected. Thus the FS1 graph structure forces one source to be connected to a target through the "out" edge on layer 1. We will call that source the FS1 source.

The FT2,3 graph structure is shown in Fig. 6(b). It has four targets and three sources. Under the assumption that there are an equal number of source and target terminals, which will hold in our final reduction, at least one of the targets must be connected to a source through the edge labeled out. In particular the target labeled X must be connected to a source through the "out" edge, because if X is connected to one of the sources in FT2,3 then it must be connected on layer 1. However in that case target $T1$ cannot be connected to any source. Thus the FT2,3 graph structure forces there to be at least one target that must be connected to a source through the "out" edge on layer 2 or 3.

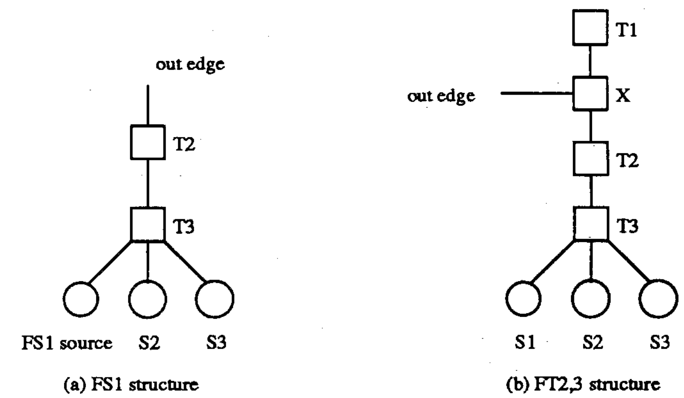


Fig. 6. Special graph structures.

Given an instance of 3DM, we construct an instance of 3GPRP as follows. For each element $y_i \in Y$ ($z_i \in Z$) there is a source vertex named S_{y_i} (S_{z_i}). For each element $x_i \in X$ there is a FT2,3 structure. We label the X target in the FT2,3 as T_{x_i}. For each triple $(x_i,\, y_j,\, z_k) \in M$ there is a target vertex named T_{x_i,y_j,z_k}. Lastly, there are $m-h$ FS1 structures. This gives a total of $h+h+3h+3(m-h) = 3m+2h$ sources and $4h+m+2(m-h) = 3m+2h$ targets. Since there are an equal number of sources and targets, all sources and targets must be connected in a feasible solution.

We partition the triples in M into h sets as follows. The triples which contain x_1, the triples which contain x_2, etc. We now describe the portion of the 3GPRP graph corresponding to the triples containing x_1. By assumption we know there are $T \geq 1$ such triples. Then associated with this partition there are $T-1$ FS1 structures, one FT2,3 structure, and targets T_{x_1,y_j,z_k} for each $(x_1,\, y_j,\, z_k) \in M$. These components are put together as follows. There is an edge from each T_{x_1,y_j,z_k} to S_{y_j}, S_{z_k}, T_{x_1} in FT2,3, and lastly to each $T2$ in the associated FS1 structures. The portion of the 3GPRP graph corresponding to the other $h-1$ partitions are constructed similarly. See Fig. 7 for an example of this construction. The reduction takes time $O(|M|)$ which is polynomial in the size of the 3DM problem instance.

Now we must show that there is a solution to the 3GPRP instance iff there is a solution to the 3DM instance. The basic idea is that S_{y_j} and S_{z_k} are connected to T_{x_i,y_j,z_k} and T_{x_i} iff $(x_i,\, y_j,\, z_k) \in M'$.

$\Leftarrow$ Let $M' \subset M$ be any solution to the 3DM. Then the corresponding 3GPRP has the following solution. For each $(x_i,\, y_j,\, z_k) \in M'$, the corresponding T_{x_i,y_j,z_k}

will connect to S_{z_k} on layer 3. Then T_{x_i} will connect to S_{y_j} on layer 2 along the path through T_{x_i,y_j,z_k}. Since M' is a 3-dimensional matching, it follows that all the sources S_{y_j} and S_{z_k} have been connected. For each $(x_i, y_j, z_k) \notin M'$, the corresponding T_{x_i,y_j,z_k} will connect to an FS1 source on layer 1. At this point the only remaining unconnected source and targets are in the FS1 and FT2,3 structures. Within each structure Si will connect to Ti on layer i, $1 \leq i \leq 3$. This is a solution to the 3GPRP. Part of the solution to the 3GPRP instance in Fig. 7 is shown in Fig. 8. In Fig. 8, wire connections are shown by the numbered lines; dashed lines have no connections on them. The numbers indicate the layers of the wires connections.

$\Rightarrow$ Suppose there is a solution to the 3GPRP problem instance constructed from the 3DM instance. Let's examine any FS1 structure, introduced because of x_1. This FS1 structure has an FS1 source A. Source A must be connected to a target since there is a solution to the 3GPRP. And, as we established before, source A must connect to a target on layer 1. Potential targets for A to connect to are: a target in a different FS1 structure, a target in the T_{x_i} FT2,3 structure, or a T_{x_i,y_j,z_k}. The source A cannot connect to a target in a different FS1 structure, because that

M = {(x1, y2, z2), (x1, y1, z1), (x2, y1, z3), (x2, y2, z3), (x3, y1, z2), (x3, y2, z3), (x3, y3, z1)}

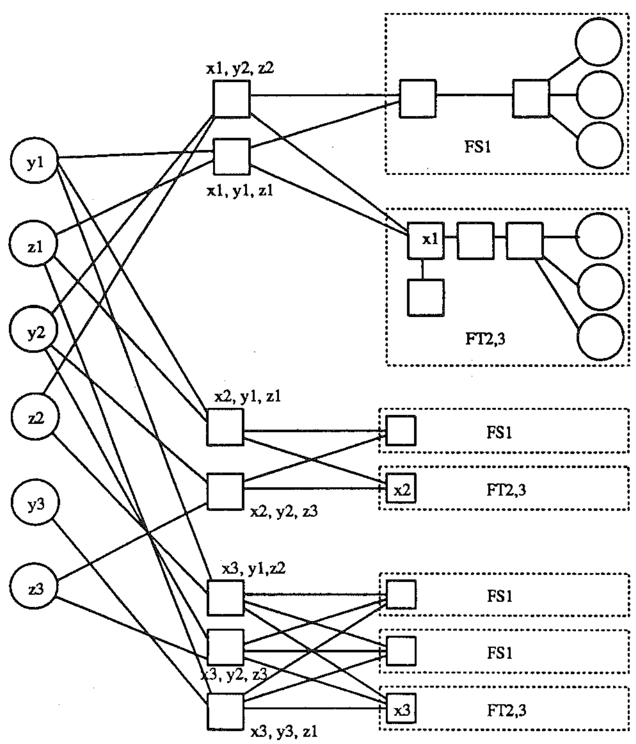


Fig. 7. Example reduction from 3DM to 3GPRP. For clarity only the subscripts of S sources and T targets are shown.

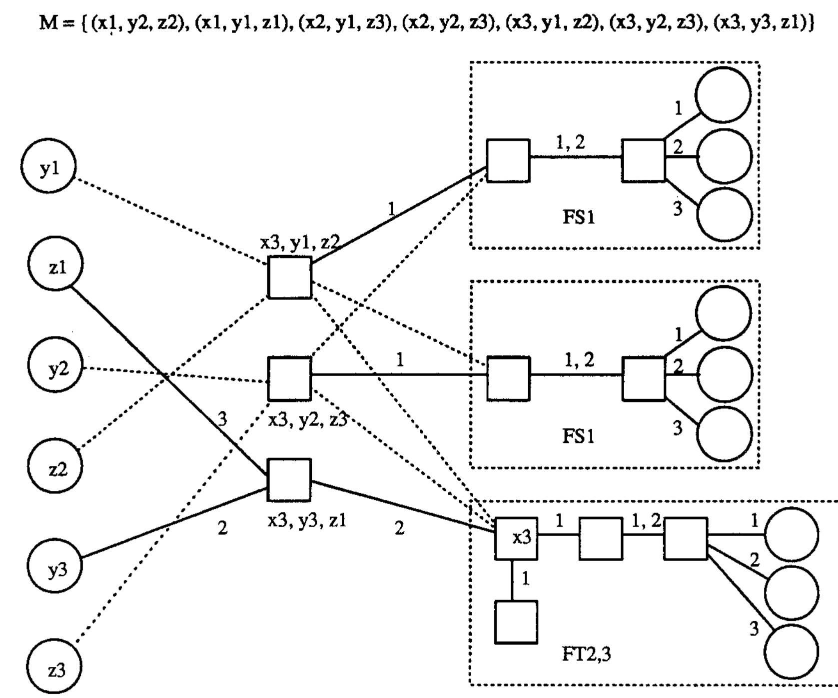


Fig. 8. A portion of the solution of 3GPRP shown in Fig. 7, with $(x_3, y_3, z_1) \in M'$. For clarity only the subscripts of S sources and T targets are shown.

FS1 structure also has an FS1 source, which must come out on layer 1. The source A cannot connect to a target in the FT2,3 structure because that would leave at least one of the sources in FT2,3 without any possible connection. So source A must connect to one of the T_{x_i,y_j,z_k} targets. In order to reach a target T_{x_i,y_j,z_k} with $i \neq 1$ A has to go through a source node. This is not possible since A must be connected on layer 1. Thus it follows that A can only be connected to a target of the form T_{x_1,y_j,z_k}. Since there are $T-1$ FS1 sources in the same partition as source A, $T-1$ targets of the form T_{x_1,y_j,z_k} will connect to FS1 sources. That leaves one target, $T_{x_1,y_{j'},z_{k'}}$, and the FT2,3 structure with T_{x_1} remaining in this partition. By the construction of the FT2,3 structure T_{x_1} must be connected on layer 2 or 3. Since it must be connected through $T_{x_1,y_{j'},z_{k'}}$, we must have T_{x_1} connected on layer 2 and $T_{x_1,y_{j'},z_{k'}}$ connected on layer 3.

Also $T_{x_1,y_{j'},z_{k'}}$ and T_{x_1} must connect to $S_{z_{k'}}$ and $S_{y_{j'}}$. Suppose that T_{x_1} does not connect to $S_{z_{k'}}$ or $S_{y_{j'}}$. Then it must connect to a source after passing through $S_{z_{k'}}$ or $S_{y_{j'}}$, but that means $S_{z_{k'}}$ or $S_{y_{j'}}$ must be connected on layer 1 which is not possible since all remaining targets must be connected on layer 2 or 3. Suppose $T_{x_1,y_{j'},z_{k'}}$ does not connect to $S_{z_{k'}}$ or $S_{y_{j'}}$. Then it must pass through $S_{z_{k'}}$ or $S_{y_{j'}}$ to another source, but $S_{z_{k'}}$ and $S_{y_{j'}}$ are only adjacent to targets so $T_{x_1,y_{j'},z_{k'}}$ cannot get to another source. Thus in a solution the set of terminals, $\{S_{y_j}, S_{z_k}, T_{x_i,y_j,z_k}, T_{x_i}\}$, are connected to each other. Since there is only one S_{y_j}, S_{z_k}, and T_{x_i} we define M' to include all the triples (x_i, y_j, z_k) representing the previously defined sets. Since by construction we only allow T_{x_i} to be adjacent to S_{y_j} and S_{z_k} when $(x_i, y_j, z_k) \in M$, and since all sources are connected in a solution to the 3GPRP, it then follows that M' is a solution to the 3DM. □

Theorem 3.2: The 3GPRP with drilled through sources is NP-complete.

Proof: In the reduction shown for Theorem 3.1 no connections were made through any source. Thus the sources can be drilled through and the same reduction works for this case. □

Theorem 3.3: The 3GPRP with drilled through targets is NP-complete.

Proof: This follows from the symmetry between sources and targets. The reduction for Theorem 3.2 can be modified by switching sources and targets. Then layer 3 is renamed layer 1 and layer 1 is renamed layer 3. □

4. Conclusion

We solved versions of the PRP using maximum flow and minimum cost flow techniques. This led to an efficient algorithm for the 2-layer PRP. We also proposed a greedy heuristic for the k-layer PRP. Two variations on the k-layer PRP were solved by the flow technique. This technique also can be used for the generalized k-layer PRP. We can show this problem is NP-Complete for $k > 2$. Currently we are studying the case when $\lambda > 1$, and trying to extend the reductions in the NP-completeness proofs to the grid PRP. Our algorithms can be adapted to the problem of reconfiguring 2-dimensional arrays in the presence of defective cells studied in Ref. 17. An interesting open problem is to develop faster algorithms for the versions of the PRP that can be solved in polynomial time. It is not unlikely that at least restricted versions of the PRP, such as the escape problem,[6] may be solvable by faster greedy algorithms.

Acknowledgements

We would like to thank Oscar H. Ibarra and Ton Kloks for their comments and suggestions to preliminary versions of this paper.

Appendix

In Fig. 9(a) we show a sample PRP that does not have a 3-layer solution, yet an illegal solution will be found by our flow algorithm. We first show that no 3-layer solution is possible for this PRP instance. Then we will show the illegal 3-layer solution found by our flow algorithm.

In the PRP shown in Fig. 9(a), there are three differently colored sources. The white sources can be connected on any layer (1, 2 or 3). The shaded sources can only be connected on layer 2 or 3 because they must be connected under at least one white source. The black sources can only be connected on layer 3 because they must be connected under at least one shaded source. The source labeled A must be connected through (under) the source labeled B. Thus the B source must be connected on layer 2. Then we must have source B connected through (under) the source labeled C. Thus source C must be connected on layer 1 and must be connected to target D (because of the other sources). However in that case source B cannot be connected to any target in layer 2. Therefore, there is no solution.

In Fig. 9(b), we show an illegal 3-layer solution that could be generated by the flow algorithm. Source B is connected on layer 2 and on layer 3, switching layers under source C.

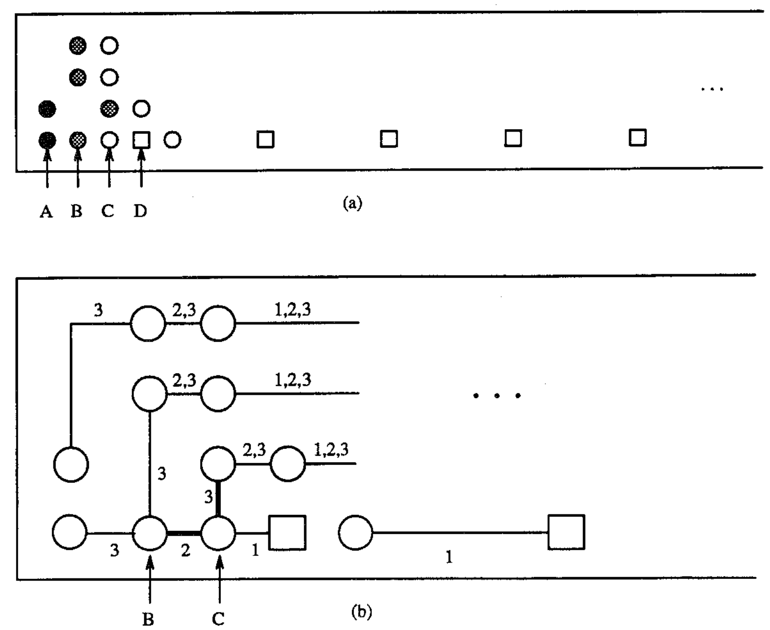


Fig. 9. There is no 3-layer solution, but an illegal solution will be found by our flow algorithm. The grid continues to the right so that there are 11 targets. Grid lines are not drawn for clarity.

One can create a PRP instance with targets spread uniformly over a square grid that does not have a 3-layer solution, but does have an illegal 3-layer solution. This is done by viewing the above PRP instance as one horizontal strip and replicating it 11 times in the vertical direction. One can show in a similar manner as above that this PRP would have no 3-layer solution, but an illegal 3-layer solution would be found by our flow algorithm. There are an infinite number of problem instances with this property.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.
2. H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, The VLSI systems series, Addison-Wesley, Reading, Mass., 1990.
3. A. J. Blodgett and D. R. Barbour, "Thermal conduction module: A high-performance multilayer ceramic package", *IBM J. Research and Development* **26** (1982) 30–36.
4. J. D. Cho, K. F. Liao, and M. Sarrafzadeh, "Multilayer routing algorithm for high performance MCMs", in *IEEE Int. ASIC Conf. and Exhibit*, 1992, pp. 226–229.
5. J. D. Cho and M. Sarrafzadeh, "The pin redistribution problem in multi-chip modules", *Mathematical Programming* **B63** (1994) 297–330.
6. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, New York, 1990.
7. S. Even and R. E. Tarjan, "Network flow and testing graph connectivity", *SIAM J. Computing* **4** (1975) 507–518.
8. S. Even, *Graph Algorithms*, Computer software engineering series, Computer Science Press, Potomac, MD, 1979.
9. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York, 1979.
10. J. M. Ho, M. Sarrafzadeh, G. Vijayan, and C. K. Wong, "Layer assignment for multichip modules", *IEEE Trans. Computer-Aided Design* **9** (1990) 1272–1277.
11. T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Applicable Theory in Computer Science Series, John Wiley & Sons, New York, 1990.
12. R. R. McBride, J. Chung, E. C. Shi, and C.-K. Cheng, "Pin redistribution for multichip module designs", in *Int. Symp. on Microelectronics*, 1993, pp. 605–609.
13. M.-F. Shiao, C. Changfan, S.-J. Chen, and C.-C. Tsai, "An efficient signal redistribution algorithm for MCM", in *Custom Integrated Circuits Conf.*, 1993, pp. 29.6.1–29.6.4.
14. N. A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers, Boston, Mass., 1993.
15. R. R. Tummala, *Microelectronics Packaging Handbook*, Van Nostrand Reinhold, New York, 1989.
16. R. R. Tummala, "Electronic packaging in the 1990's — A perspective from America", *IEEE Trans. Components, Hybrids, and Manufacturing Technology* **14** (1991) 262–271.
17. H. Y. Youn and A. D. Singh, "An efficient channel routing algorithm for defective arrays", *IEEE Int. Conf. on Computer-Aided Design*, 1989, pp. 432–435.