VLSI DESIGN
1996, Vol. 5, No. 1, pp. 23-36
Reprints available directly from the publisher
Photocopying permitted by license only

A Greedy Algorithm for Over-the-Cell Channel Routing

GUDNI GUDMUNDSSON[a,*] and SIMEON NTAFOS[b,†]

[a]*Sunnuhvoli B, IS-170 Seltjarnarnes Iceland;* [b]*Computer Science Program, The University of Texas at Dallas, Richardson, TX 75083-0688*

(Received 19 September 1994; In final form 20 June 1995)

Recent advances in VLSI technology have made the area over cells available for routing. In this paper we present a new over-the-cell channel router that uses greedy heuristics to make the over-the-cell connections and to define the nets needed to complete the connections inside the channel. The router tries to reduce the channel density by moving segments that cross maximum density columns to the over-the-cell areas. The layout model used allows only planar connections over each cell. The final stage is to use an existing channel router to route the connections inside the channel. An important characteristic of the new router is that there is interaction between the decisions made for the over-the-cell connections and the connections needed inside the channel. It performs significantly better than previous over-the-cell routers.

Keywords: Over-the-cell channel routing, VLSI

1. INTRODUCTION

Since being introduced by Hashimoto and Stevens in 1971 [13], channel routers have been among the most useful tools in the automatic layout of VLSI circuits. A *channel* consists of two horizontal rows of points called *terminals* that belong to two cells (*bottom*, *top cells*). Each terminal belongs to a *net* and has the name of the net as its label. In the *rectilinear*, *two reserved layer* model there are two wiring layers available for rectilinear routing. All horizontal wire segments use one layer and all vertical wire segments use the other layer. Connections between layers are done through contact windows called *vias*. It is convenient to impose a unit grid on the channel so that all terminals and vias are at grid points and all wire segments follow grid lines (called *tracks* and *columns*). The *local density* at a column *c* equals the number of nets for which a horizontal wire that intersects the column is necessary (in all routings). The *channel density* is the maximum local density in any of the columns. A channel of density *d* needs at least *d* tracks to be routed.

Certain technologies for the production of VLSI circuits open up the possibility that the space over the cells defining the channel might be available for rout-

*Phone: 011-354-561-1736. Fax: 011-354-560-8175. E-mail: gudnig@isbank.is.
†Phone: 214-883-2809. Fax: 214-883-2349. E-mail: ntafos@utdallas.edu.

ing [7, 20]. This leads to the question of how to use this space to further reduce the area needed for the VLSI circuit. As before, a routing that fully connects all the nets is required and the objective is to find a routing that minimizes the number of tracks needed between the two channel rows. The number of tracks used by the over-the-cell connections is limited by the size of the cell.

This over-the-cell channel routing problem (OCCRP) is, as most problems in VLSI design, computationally intractable [11]. There are two subproblems in developing efficient algorithms that produce good solutions: (a) Allocation of connections to the area inside the channel and the areas over the cells on either side, and (b) routing of the connections in each area. Many of the over-the-cell routers that have been reported [5, 7, 12] use the following three stage approach:

Step 1: Make the connections over each cell. This produces groups of terminals (*hyperterminals*) that are locally connected.

Step 2: Select terminals within the hyperterminals that will be used to completely connect each net inside the channel.

Step 3: Use existing channel routers to route the channel.

In this paper we describe a new greedy approach to solve the over-the-cell channel routing problem. The layout model used is the extended rectilinear, two reserved layer model, with planar over-the-cell routing (single interconnection layer available over each cell) and unlimited number of over-the-cell tracks. The planarity requirement for the over-the-cell connections severely limits the number of over-the-cell tracks that are used and the router can be easily adapted to handle a specified number of available over-the-cell tracks. In this router we use only two stages, where the first stage combines the first two stages of the three stage approach. This results in improved performance because we take into account the interaction between the over-the-cell and the internal connections (rather than treat them independently as in the three stage approach). As before, the second (final) stage is to use a conventional two layer channel router to route the connections that are necessary inside the channel.

In section 2 we give a brief overview of recent over-the-cell channel routers. In section 3 we describe the new greedy over-the-cell router. We compare the new router with the routers reported in [1, 5, 7, 9, 16] in section 4. The new router has the best performance in terms of final density inside the channel. It also does well in terms of the number of tracks used for over-the-cell routing.

2. OVERVIEW OF RECENT OVER-THE-CELL CHANNEL ROUTERS

Over-the-cell channel routing has been the topic of extensive research in the past few years. In this section we give a brief descriptions of recent over-the-cell channel routers that use the same interconnection model (two layers inside the channel and one layer over the cells) as ours. Five of them are later used in an experimental comparison with our router.

In [5], Cong and Liu describe and improve their over-the-cell channel router that they first reported in [4]. This channel router follows the three stage approach. They use a dynamic programming approach (adapted from [21]) for routing over the cells. However, they convert the multiterminal over-the-cell planar routing problem into an equivalent two terminal problem, and therefore can allow connections between terminals in the same net that are not "nearest neighbors". Furthermore, they assume that net segments coming into (from the left) and leaving (to the right) the channel can be routed over the cells. Since Cong and Liu are working with an increased number of over-the-cell connection candidates, the complexity of their first stage becomes $O(n^2N^2)$ (rather than $O(N^2)$ in [21]), where n is the maximum number of terminals on the terminal row that belong to the same net and N is the number of terminals on the terminal row.

Cong and Liu proceed to show that the general problem of choosing net segments inside the channel to fully connect the nets and minimize channel density after over-the-cell routing is NP-hard. To solve the net segment selection problem a greedy heuristic is used. First a *hyperterminal* (superterminal) connection graph is created, where the nodes correspond to hyperterminals and (possibly parallel) edges correspond to a connection between two hyperterminals. When looking at net segments inside the channel that can be used to connect a terminal to the rest of the net, it is only necessary to consider connecting the terminal to the terminals closest to it. The edges in the hyperterminal connection graph have their net segment intervals as weights. The objective is to find a spanning forest for the graph, such that the channel density resulting from the chosen edges (i.e., the ones in the spanning forest) is minimized. First all critical edges (edges whose removal increases the number of connected components) are identified. A connection is chosen for elimination in a greedy way, i.e., if it is estimated to contribute the most towards the density inside the channel and is not critical. This is repeated until we are left with a spanning forest for the connection graph. The complexity of this heuristic is $O(N^2 \log N)$.

Cong, Preas and Liu [7] (originally reported in [6]) modified the over-the-cell router above by adding weights to the over-the-cell connection candidates. The main purpose of [7] is to show what needs to be done to adapt the over-the-cell channel router to "real life situations", that is cases where the number of tracks for over-the-cell routing is limited. Two terminal rows are routed over each cell, i.e., the cell contains the top and bottom terminal rows for two separate channels. Finally the cell may be divided (due to feedthroughs from global routing) so that it is not always possible to connect two terminal on the same terminal row that belong to the same net.

Lin, Perng, Hwang and Lin [16] formulate OCCRP as a linear programming problem. They only consider over-the-cell connections between terminals in the same net that are "nearest neighbors" in each terminal row. The benefit from this approach is that they do not have to estimate (guess) how much each over-the-cell connection will help in reducing the density inside the channel. A drawback of this approach is the limitations on the over-the-cell connection candidates that are considered. The main drawback however is that the number of constraints is $O(N^2)$ (and therefore the complexity of the linear programming problem grows very fast with the number of terminals). Also, the number of variables in the linear programming problem grows linearly with the number of net segments. Therefore, as nets have more terminals, the time required to find a solution grows very fast.

Das, Nandy and Bhattacharya [9] approach the problem by using a blend of the three stage approach (reversing the order of the first two steps) and the greedy approach to determine the connections to be made over the cells. They start by using a greedy strategy to decide, for each net that has terminals on both sides of the channel, which net segment will always be routed inside the channel to complete the connection of the net between the two sides. They proceed to show how five different cases should be handled to maximize the density reduction possible inside the channel. After starting with all possible over-the-cell connection candidates (as in [5]) they select over-the-cell connections according to a weight function that tries to measure the impact of the connections that will be forced inside the channel because of the planarity restriction. In a way, the objective of Das, Nandy and Bhattacharya is to increase the density in the channel as little as possible, from a known minimum of connections that have to be carried out inside the channel. (On the other hand, in our greedy approach we start by assuming that everything is routed inside the channel and select an over-the-cell connection that maximizes the density reduction inside the channel.) The time complexity of the algorithm in [9] is $O(N\,D\,s(N + nI))$. Here, N and n are as defined above, D is the original density of the channel, s is the maximum span of a net and I is the size of the maximum independent set of over-the-cell connection candidates.

Holmes, Sherwani and Sarrafzadeh [14] extend the k-track over-the-cell channel router by Cong, Preas and Liu [6] to take advantage of vacant terminals. The weights of connection candidates are not based only on the density within the channel. They also try to estimate the impact of the connection on the vertical constraint graph for the resulting channel. Connection candidates that use vacant terminals are added if two terminals in the same net and on opposite sides of the channel each have a vacant terminal in the same column. Also, if there exists a column in which both terminals are vacant, they consider using this column as part of the net (i.e., there are "virtual" terminals for the net on both sides of the channel). Holmes et al. use a greedy heuristic to assign vacant columns to nets. The time complexity of this over-the-cell channel router is $O(N^2(k + v))$, where v is the maximum number of vacant columns over the span of any net that is considered for taking advantage of vacant columns.

Chang, Hsiao, Yan and Shew [1] use a two stage approach consisting of over-the-cell routing followed by conventional routing within the channel. The over-the-cell routing stage routes certain subnets that intersect the column with highest density over the cell and determines what net segments need to be connected inside the channel. A conventional channel router can then be used to complete the task. The overall approach in [1] is similar to ours but different methods are used to make the critical decisions and the results obtained by the two routers are significantly different. We note that the two routers were developed independently (ours was completed in 1992).

3. A GREEDY OVER-THE-CELL CHANNEL ROUTING ALGORITHM

To solve the over-the-cell routing problem, we start by assuming that all the connections will be made inside the channel. We create an initial set *OCC* of candidates for over-the-cell connections (*OC-candidates*). For each candidate we find a "best" net segment (*reduction interval*) in the portion of the net that is routed inside the channel that can be eliminated if the OC-candidate is routed over-the-cell. An OC-candidate is acceptable if connecting it over-the-cell lowers the density in a maximum density column inside the channel. Our greedy approach proceeds as follows:

```
    while OCC contains an acceptable OC-candidate
        do
begin
    let occ be a "best" OC-candidate (using rules
      c.1,2);
    route occ over-the-cell;
    remove from OCC all OC-candidates that conflict
      with occ;
    add new OC-candidates to OCC (rule b-iii; Figure
      2) and determine a reduction interval for each
      one.
    re-evaluate the reduction intervals for an appropri-
      ate subset of the OC-candidates and possibly add
      more OC-candidates (rules c-i,ii,iii).
end.
```

The motivation for our approach and for the way the various steps are implemented was to see if the level of performance of the linear programming approach in [5] could be achieved using a much faster algorithm. The router in [5] showed that good results can be obtained by focusing on net segments that span maximum density columns. The router in [7] uses weights that incorporate density information and selects a maximum-weight over-the-cell routing. However, the weights are static (i.e., they are assigned initially but are not updated as over-the-cell connections are made) and it seems that many of the over-the-cell connections reduce the local density without affecting the channel density. This led us to require that OC-candidates that are selected for routing over-the-cell should reduce the density in a maximum density column. In order to accomplish this, the routing of over-the-cell connections had to be combined with the selection of connections to be made inside the channel (resulting in a two stage approach).

The set of OC-candidates considered by the routers in [5,7] is larger that that of [16] because over-the-cell connections between terminals that are not nearest neighbors are used. In our router we allow a restricted set of OC-candidates connecting terminals that are not nearest neighbors. Our goal was to have enough OC-candidates so that the performance of [16] can be reached and, at the same time, limit the number of OC-candidates considered so that the complexity of the algorithm can be kept low. Complexity considerations also led us to disallow backtracking (i.e., once a connection is routed over the cell, it is never retracted).

In the following subsection we describe the way an over-the-cell connection can be used to reduce the density within the channel. Then we describe the set of possible over-the-cell connection candidates (OC-candidates) that we allow and how we determine which OC-candidate is chosen for routing over-the-cell. Finally, we analyze the time complexity of our algorithm.

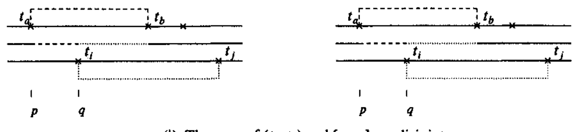
(a) Reduction Intervals for OC-Candidates

Let us consider an OC-candidate (t_i, t_j), where t_i, and t_j, are two terminals on the same side of the channel that belong to the same net n' and are in columns $cl(t_i)$ and $cl(t_j)$ respectively (cl is the column function). Let net n' have terminals (on either side of the channel) in columns $c_1, c_2, \ldots, c_m$, such that $cl(t_i) = c_1 < c_2 < \ldots < c_m = cl(t_j)$. Note that net n' may have terminals in other columns that are not in $[c_1, c_m]$. If there is no over-the-cell connection to the terminals of n' in columns $c_1, \ldots, c_m$ (except possibly from c_1 going to the left or from c_m going to the right) then there must be a wire inside the channel that spans $[c_1, c_m]$. Then we can eliminate a segment of this wire between columns c_l and c_{l+1} for some l, $1 \leq l < m$. If two or more wire segments in $[c_1, c_m]$ are eliminated, the net n' is clearly no longer fully connected. We call the interval $[c_l, c_{l+1}$ a *(density) reduction interval.* (Note that if $1 < l < m - 1$ then the reduction interval is open; otherwise it may be half open or closed, depending on whether there are wires inside the channel that connect to c_1 from the left and to c_m from the right.)

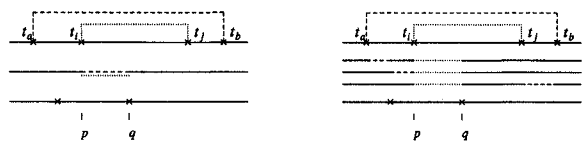
Let us now assume that one over-the-cell connection has been made for net n', say between terminals t_a and t_b where $cl(t_a) < cl(t_b)$. Then there is one break in the connection for n' inside the channel. This net segment corresponds to a reduction interval, say $[p, q]$. Note that there can be no column between columns p and q that contains a terminal of n'. Now, consider some other OC-candidate (t_i, t_j) for n'. There are four possibilities (see Figure 1):

(i) The spans of (t_i, t_j) and $[p, q]$ are disjoint (they may have a common endpoint). Then there must be a wire inside the channel for net n' spanning $[cl(t_i), cl(t_j)]$. Any reduction interval in $[cl(t_i), cl(t_j)]$ is allowable because the connection (t_i, t_j) would keep n' fully connected when routed over-the-cell. Note that if (t_i, t_j) and (t_a, t_b) are on the same terminal row they must be either side by side or (t_i, t_j) is contained within (t_a, t_b). Otherwise, the spans of (t_i, t_j) and (t_a, t_b) may partially overlap as long as the span of (t_i, t_j) does not intersect $[p, q]$.

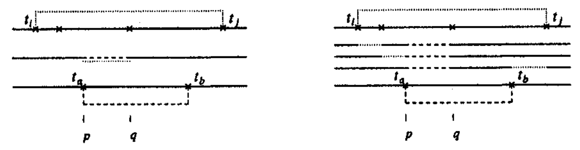
(ii) The span of (t_i, t_j) contains $[p, q]$ and is contained totally within the span of the already routed connection (t_a, t_b). If the OC-candidate (t_i, t_j) were routed over the cell, it would keep the terminals in columns p and q connected to the rest of net n'. Therefore, we can now look for a reduction interval that is within the span of



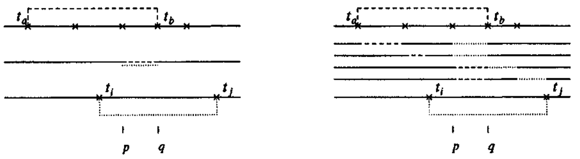
(i) The span of (t_i, t_j) and $[p, q]$ are disjoint.



(ii) The span of (t_i, t_j) contains $[p, q]$ and is contained within the span of (t_a, t_b).



(iii) The span of (t_i, t_j) contains $[p, q]$ and the span of (t_a, t_b).



(iv) The span of (t_i, t_j) contains $[p, q]$ and exactly one of t_a, t_b.

FIGURE 1 Examples of finding reduction intervals when an over-the-cell connection has been made for the net.

(t_a, t_b) and outside the span of (t_i, t_j). If we find such an interval, we associate $[p, q]$ with (t_i, t_j) and the newly found interval with (t_a, t_b). Notice that we do not gain anything by routing two over-the-cell connections (on opposite sides of the channel) for the same net and between the same two columns.

(iii) The case when the new OC-candidate contains $[p, q]$ and spans the already routed candidate is analogous to (ii) above, where the roles of the two candidates are reversed.

(iv) The span of (t_i, t_j) contains $[p, q]$ and one of (t_i, t_j) is inside the span of (t_a, t_b) while the other is outside (planarity forces the candidates to be on opposite sides of the channel). Then routing the OC-candidate (t_i, t_j) over the cell would keep the terminals in columns p and q connected to the rest of net n'. Therefore, we can now look for a reduction interval that is within the span of either one of the candidates but is not inside their intersection. The association of $[p, q]$ will change to (t_i, t_j) if the new reduction interval is found within the span of (t_a, t_b).

Figure 1 (i)–(iv) illustrates the four cases for the interaction between the OC-candidates and their reduction intervals that we described above. For each case, the left side of the figure shows the original OC-candidate and the corresponding reduction interval (dashed lines) and the new OC-candidate and its reduction interval (dotted lines). The right side of the figure shows the possible locations of the reduction intervals after both OC-candidates are routed (there is reassignment of reduction intervals in cases (ii) and (iv)). Note that only one reduction interval will be selected from among the various possibilities.

When more that one over-the-cell connection has been made, allowable reduction intervals for an OC-candidate can be found by repeated application of the analysis in cases (i)—(iv) above. It is worth noting that because of case (iv) it is possible that we have to evaluate reduction intervals anywhere in the span of the net (if there is a chain of partially overlapping over-the-cell connections). Also, when a reduction interval has been used because of a routed over-the-cell connection, we never retract this choice and select a different reduction interval at a later time. However, as we saw in cases (ii) and (iv), reduction intervals may be associated with a different over-the-cell connection as more connections are routed over-the-cell for the same net.

(b) Building and Maintaining the Set of OC-Candidates

At any time, we allow each terminal to be part of at most two *two-terminal (or regular) OC-candidates*, one going to the left and the other to the right (initially, these OC-candidates connect nearest neighbors). A *three terminal OC-candidate (3TOC-candidate)* is introduced at any terminal t' that has two regular OC-candidates and the current local density in its column is maximum. The reduction interval associated with a 3TOC-candidate is the one we get by joining the two intervals that have t' as a common endpoint. A 3TOC-candidate is not introduced if either one of these reduction intervals is not allowed. In constructing our set of OC-candidates, vacant terminals are not used. Connecting exits (nets that enter the channel from the left or leave the channel to the right) over-the-cell is allowed. Moreover, the relative ordering of these exits is not fixed (i.e., we do not introduce new terminals for these nets at the beginning and end of the channel).

The set of OC-candidates is built and maintained as follows:

(i) Initially the set of OC-candidates consists of all the regular OC-candidates that we get by connecting to "nearest neighbors" in the net and on the same row. If a net has exits, we also introduce a regular OC-candidate going to the left from the leftmost terminal and/or one going to the right from the rightmost terminal for the net on each terminal row as is appropriate (we introduce columns -1 and $N + 2$ that we treat in a special way). 3TOC-candidates are included as warranted by the initial set of regular candidates.

(ii) When an OC-candidate *occ* for net n' is routed we delete from the current set of OC-candidates all other candidates for net n' that are on the same terminal row as *occ* and either contain *occ* or are contained within the span of *occ*.

(iii) When *occ* is routed we eliminate candidates from other nets on the same terminal row according to the planarity restriction. Assume two regular OC-candidates from some other net n'' are eliminated, where one spans the leftmost terminal in *occ* and the other spans the rightmost terminal of *occ*. Then we create a new regular OC-candidate for net n'' that connects the terminals outside the span of *occ* that were endpoints for the two eliminated OC-candidates (unless they both correspond to exits), see Figure 2. Note that the OC-candidate connecting the inner pair of terminals in Figure 2 is not affected. 3TOC-candidates are included as warranted by the newly created OC-candidates.

When an OC-candidate is created, we associate a reduction interval with it. This is the reduction interval that would be used if this OC-candidate is chosen for over-the-cell routing. We choose a reduction interval based on the maximum local density that this interval would reduce inside the channel. Ties are broken by how many columns in the reduction interval have this local density. In case of another tie, the first interval that we find is used.

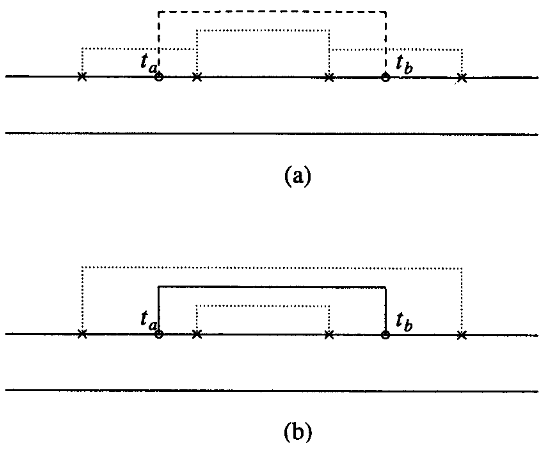


FIGURE 2 Maintenance of active set of OC-candidates using rule (iii). (a) Before routing (t_a, t_b). (b) After routing (t_a, t_b).

(c) Selection of OC-Candidates for Routing

We choose the next OC-candidate to be routed over-the-cell by following the criteria below:

(1) Select an OC-candidate that reduces the current density for the maximum number of maximum density columns in the current channel. When routing of the OC-candidate will result in reassignment of reduction intervals (cases (ii) and (iv) in figure 1), we use the"new" reduction interval in the evaluation. If no OC-candidate has a reduction interval that spans a maximum density column, then we stop.

(2) If there is a tie in (1), select the OC-candidate that minimizes the number of OC-candidates that (a) would have to be discarded because of planarity and (b) reduce the density in a maximum density column. In case of another tie, select the OC-candidate that minimizes the number of OC-candidates that reduce the density in a column with local density one less than the maximum density and have to be discarded because of planarity. Note that this count includes OC-candidates that reduce the density in a maximum density column but are such that those columns also get reduced by the reduction interval associated with the candidate under consideration. In case of another tie, select the first OC-candidate.

When choosing OC-candidates for routing we travel the terminal rows from left to right. We found that first considering the opposite terminal row from the one over which the last connection was made (initially we consider the bottom) slightly improves the quality of our solutions.

Because of the way we choose OC-candidates for routing, it is logical to store the relevant information for the reduction interval currently associated with the candidate. So for each OC-candidate we store the terminals it connects, the columns that define the reduction interval, the highest density reduced and the number of columns covered that have this density. So that this information is current when the next over-the-cell connection is chosen, we must do the following after routing over-the-cell connection *occ* (and reducing the density inside the channel as implied by its reduction interval):

(i) Reevaluate the reduction intervals for any remaining OC-candidates from the same net as *occ*.
(ii) Reevaluate the reduction interval for any OC-candidate (from a different net than *occ*) whose current reduction interval intersects the reduction interval associated with *occ*.
(iii) If the density of the channel has just been reduced by one, then we must consider each terminal in a maximum density column for introduction of a 3TOC-candidate.

Finally, since a secondary goal of ours is to minimize the number of tracks used for over-the-cell routing, we roll back the last OC-candidates that were routed and did not result in further reduction of the density inside the channel. This can be easily accomplished by keeping routed OC-candidates on a stack and clearing the stack every time the density of the channel has been reduced by one.

(d) Complexity Analysis

In order to be able to analyze the time complexity of our implementation it is necessary to describe briefly the data structures that are used. The data for the original channel is kept in a two dimensional array ($2 \times N$) of terminal structures. Each terminal structure contains the net the terminal belongs to as well as the columns of the nearest terminals (both left and right) on each side of the channel that belong to the same net. The over-the-cell routing information is similarly kept in a two dimensional array. Each element of this array stores the column, to the left and right, that is (possibly) connected by an over-the-cell wire to this terminal. Also, there are pointers for up to three active OC-candidates associated with this terminal, two regular OC-candidates that go left and right, and a 3TOC-candidate that has this terminal as its middle terminal. The current density inside the channel is kept in a one dimensional array; each element stores the current local density for the corresponding column of the channel. Finally, for each net we have a linked list of intervals that show the wires inside the channel that remain for the net.

In the following analysis of the time complexity N is the number of terminals on each side of the channel, n the maximum number of terminals on either terminal row that belong to the same net, D the original density of the channel, s the maximum span of a net, and I the size of the maximum independent set of over-the-cell connection candidates.

Initializing the terminal array, calculating initial densities and setting up initial net intervals can clearly be done in $O(N)$ time from an input that lists the nets coming into and leaving the channel, as well as listing the number of the net in each column. To determine the time complexity of finding the initial set of OC-candidates we notice that each possible reduction interval for a net figures in the calculation of at most two regular OC-candidates. These are the two regular OC-candidates, one on each side of the channel, that span the reduction interval. This same reduction interval can appear in at most four 3TOC-candidates, two on each side. These have their middle terminal in one of the columns that determine the interval. Therefore the density reduction calculations for the initial OC-candidates can be done in time proportional to the total span of all the nets in the channel. This is clearly $O(D\,N)$, that is proportional to the total area inside the original channel.

Now we turn our attention to the choosing of which OC-candidate should be routed next over-the-cell. By scanning the connection array we find all the OC-candidates that have reduction intervals covering a maximum density column. Because of our rules for the set of active OC-candidates, at most $O(D)$ OC-candidates have to be discarded because of planarity. For any given OC-candidate, finding the candidates it interferes with can be done by scanning the span of the OC-candidate in question (in $O(s)$ time). Actually calculating the amount of interference for a candidate takes $O(n + s)$ time (the n is because we need to check whether the reduction interval is open or closed). The time complexity for choosing the next OC-candidate for routing is therefore $O(N\,(s + D(n + s))) = O(N\,D\,s)$, since $n \leq s$. Since the number of times that we choose an OC-candidate for routing is $\leq I$, the overall time complexity of this step is $O(I\ N\ D\ s)$.

Routing the chosen OC-candidate (call it *occ*) is done in constant time. Reducing the density over its reduction interval and updating the inside the channel wire intervals for the net that *occ* belongs to (call it n') can be done in $O(s + n)$ time.

Removing the OC-candidates for net n' that overlap the span of *occ* takes $O(n)$ time. It takes $O(n(n^2 + s))$ time to reevaluate the density reduction interval for the OC-candidates that remain for net n'. As before, there are $O(D)$ OC-candidates blocked by *occ*, and those candidates can be found in $O(s)$ time. We create at most $O(D)$ new OC-candidates by merging two candidates for the same net that are blocked by *occ*. Finding the best density reduction interval for each of these new OC-candidates can be done in $O(n^2 + s)$ time. The time required to obey the planarity restriction is therefore $O(s + (n + D)(n^2 + s)) = O((n + D)\ (n^2 + s))$.

Next we turn our attention to reevaluating the reduction intervals for those OC-candidates whose current best reduction interval intersects the reduction interval of the newly routed candidate. There is no need to reevaluate OC-candidates whose best reduction interval does not intersect the reduction interval for *occ* (since any such interval gets worse). To estimate the number of the OC-candidates that need to be reevaluated, we need to know how many nets may have such reduction intervals. This is clearly linear in the span of the reduction interval for *occ*. The time complexity for each update is therefore $O(sn(n^2 + s))$. If d is the final density inside the channel after our over-the-cell routing, then the total density reduction is $O(N(D - d))$. Thus, the overall time complexity of this step is $O(N\ D\ n(n^2 + s))$.

Creating new 3TOC-candidates when the density inside the channel has been reduced by one takes $O(D\ N)$ time, each time. Since this is done at most D times, the overall time complexity for this step is $O(D^2N)$.

Since the over-the-cell routing is planar, the size of the maximum independent set of over-the-cell connection candidates (I) is linear in the number of terminals. This then means that the complexity of removing and creating new OC-candidates after routing is dominated by the complexity of choosing the candidates for over-the-cell routing. From the above we see that the total time complexity for our algorithm is then $O(D\ N(I\ s + n(n^2 + s) + D))$.

According to [5, p. 411] it is safe to assume that the number of terminals per net is bounded by a small constant. They claim that in industrial examples the average number of terminals per net is between 2 and 3, and the maximum is between 16 and 18. In theory n could be $O(N)$. However, the above observation leads us to believe that the fact that we may have to check most of the density reduction segments for a net for each active OC-candidate is not too time consuming. Thus, we probably would not gain much time by using more complicated data structures than a linked list to store the current wire segments inside the channel.

4. EXPERIMENTAL RESULTS

We implemented the first stage of the over-the-cell channel router in C and ran our experiments on a SUN SPARCstation SLC. Our experiments consisted of "routing" (i.e., deciding which connections to make over-the-cell and inside the channel so that the resulting density inside the channel is minimized) the seven channels used in [8,22] and the two circuits used in [7]. The seven channels used in [8,22] were originally published in [15] (Ex1–Ex5) and in [10] (Deutsch). We compare the over-the-cell routers based on the final density inside the channel rather than the number of tracks used for routing the channel. This is reasonable since almost all channels will be routed at density, or at most one track over density, by a good channel router.

In Table I we show the results of using the new router on routing the examples with those for the routers in [1, 5, 7, 9, 16]. The router in [14] is not included in the comparison because it uses vacant terminals (resulting in a very different set of over-the-cell routing candidates). After determining how to use the vacant terminals, the router in [14] uses a slightly modified version of the three stage approach in [5]. Holmes et al. [14] report that on standard examples this router (without the use of vacant terminals) performs similarly to previous routers. From the results presented here, we believe that combining our router with the initial stages of the router in [14] will improve the results obtained by either router alone.

Table I lists the densities (except in the case of [9], see below) after over-the-cell routing by all the over-the-cell routers. The old routers are identified by the references to the corresponding paper. The columns marked T, B and I, give the resulting densities over the top (T) and bottom (B) cells as well as the resulting density inside the channel (I). For our router we also have a column marked #C; this gives the total number of over-the-cell connections (i.e., the number of iterations). The densities reported here for the improved three stage approach by Cong et al. [7] were not reported in the paper, but were obtained through personal communication with Dr. Jason Cong.

From Table I we see that our new router always reduces the density inside the channel more than the three stage approach in [5]. We also see that our router outperforms the improved three stage approach [7]. For all but two of the channels we reduce the density inside the channel more, and when there is a tie we use fewer tracks for over-the-cell routing. Our router performs better than the one reported in [1] as well. It reduces the density inside the channel more in six of the seven examples; it achieves the same density in one example but uses less over-the-cell connections for it. A contributing factor to the difference in performance between the router in [1] and the new router is that the router in [1] considers OC-candidates that connect to nearest neighbors only. Also, the router in [1] estimates the expected benefit of making an over-the-cell connection; it determines how to use the selected connection in order to reduce the density inside the channel after the connection has been selected. Our router first decides how a connection will be used to reduce the density inside the channel and that information is used in the selection process.

A direct comparison with the greedy approach by Das et al. [9] is difficult because: (i) They only report the total number of tracks used for over-the-cell routing (shown here in column O). (ii) Only the number of tracks used inside the channel after routing the channel with the Greedy Channel Router [19] are reported (i.e., we do not know the actual density inside the channel). (iii) They do not report results for example 5. We believe that using actual density reduction inside the channel to select candidates for over-the-cell routing (whereas Das et al. use an estimation formula) should make our algorithm produce better results.

Finally, let us compare our router to the linear programming approach of Lin et. al [16]. These two algorithms do produce very similar results. Each has better inside the channel density in one case (the new router decrease the density by 3 more for example 3b, while Lin et al. do 1 better for example 3c) and the overall number of over-the-cell tracks goes back and forth. The inherent weakness of using linear programming leads us to believe that the new router is preferable to the linear programming approach. First, the necessity of having a linear programming solver

TABLE I Channel densities for various over-the-cell routers.

Channel	Cols	Density	[5]			[16]			[7]			[9]		[1]			New Router			
			T	B	I	T	B	I	T	B	I	O	I	T	B	I	T	B	I	#C
Ex1	35	12	4	3	9	2	5	8	4	4	8	7	9	1	3	9	3	3	8	7
Ex3a	62	15	3	6	12	3	5	11	3	6	12	6	11	2	3	12	2	7	11	13
Ex3b	61	17	2	5	13	2	4	13	2	5	13	7	13	2	3	13	6	6	10	13
Ex3c	79	18	3	4	14	4	4	12	4	4	14	6	14	2	4	14	2	4	13	10
Ex4b	119	17	5	4	16	5	4	12	4	6	12	9	13	2	3	13	2	5	12	12
Ex5	119	20	4	3	14	6	4	11	6	4	11	—	—	6	4	11	5	4	11	12
Deutsch	174	19	8	7	16	2	4	15	10	7	16	11	16	3	2	16	4	4	15	17

makes this approach very unlikely to be intergrated into"real world" computer aided design tools. Furthermore, the unpredictability of the running time to solve the linear programming problem makes it unappealing. While the running times for the new router on the example channels ranges form 0.090 to 0.530 seconds with an average of 0.267 seconds, Lin et al. report a running time of 5.4 to 96.6 seconds (average 39.6 sec.) for all channels except 4b, which takes 629.4 seconds to be solved. (Obviously the running times are not directly comparable because different computers are used. This only shows the way that the complexity of the channel, rather than its size, plays a major role in the time needed by the linear programming approach.)

Table II shows the density reduction inside the channel (except for [9] where we use number of tracks) as a percentage of the original density of the channel. The average density reduction shows that the new router outperforms all the other routers. In the case of Lin et al. this is on the strength of the solution for channel 3b.

Tables III and IV compare the density reduction obtained by the new greedy router with the results in [7] for the horizontally-connected vertically-divided (HCVD) layout model. In this model it is assumed that power and ground buses are routed at the middle of the cells, so the over-the-cell routing region is divided in two. In this case 5 tracks are available for over-the-cell routing for each terminal row. Each table shows the results of routing the channels of one standard cell circuit. These circuits are the Reed-Solomon Decoder (RSD) and Primary1. Primary1 is a benchmark example used in the Physical Design Workshop [18]. The actual channel definitions of the circuits, i.e., the results of global routing, were obtained from [2].

Cong et al. do not show the densities over-the-cells. We can however get an idea as to whether the 5 track limit has an effect on the resulting densities by comparing with the results reported for the horizontally-connected vertically-connected (HCVC) layout model. In HCVC the over-the-cell routing region of a cell is treated as common area for both terminal rows of the cell, and moreover there are now 13 tracks available for over-the-cell routing. It turns out that the inside the channel densities for the RSD do not get further reduced, and are only reduced in 3 channels in Primary1 (see [7]). Furthermore, the results for Primary1 for the HCVC model are just obtained by combining the solution for unlimited number of over-the-cell tracks for each terminal row. It can therefore be concluded that the limitation of having only 5 tracks for over-the-cell routing in the HCVD model does not severely affect the outcome of the density reduction by the improved three stage approach of [7].

Table III compares the results in the case of the Reed-Solomon Decoder. In only one case (channel 10) is the final density found by the new router higher than that obtained by Cong et al. The final densities are the same in 6 cases and the new router reduces the final densities more for the remaining 8 channels. It should be noted that Cong et al. report the initial density for channel 8 as being 13. This seems to be because of a net that only goes straight across the channel in a maximum density column (has only two terminals). This does not really increase the density

TABLE II Density reduction inside the channel for various over-the-cell routers.

Channel	Density	[5]	[16]	[7]	[9]	[1]	New Router
Ex1	12	25.0%	33.3%	33.3%	25.0%	25.0%	33.3%
Ex3a	15	20.0%	26.7%	20.0%	26.7%	20.0%	26.7%
Ex3b	17	23.5%	23.5%	23.5%	23.5%	23.5%	41.2%
Ex3c	18	11.1%	33.3%	22.2%	22.2%	11.1%	27.8%
Ex4b	17	5.9%	29.4%	29.4%	23.5%	23.5%	29.4%
Ex5	20	30.0%	45.0%	45.0%	-	45.0%	45.0%
Deutsch	19	15.8%	21.1%	15.8%	15.8%	15.8%	21.1%
Average reduction		18.8%	30.3%	27.0%	22.8%	23.4%	32.1%

TABLE III Channel densities in Reed-Solomon Decoder after over-the-cell routing by Cong et al. and our new router.

				New Router			
Channel	Cols	Orig. density	[5] Inside	Top	Bottom	Inside	# OCC
2	58	11	9	0	2	9	2
3	61	9	8	3	2	5	7
4	57	11	9	1	2	8	3
5	56	7	7	1	1	6	2
6	63	11	8	2	1	8	3
7	68	10	8	1	3	7	5
8	67	12	10	0	3	9	3
9	63	10	10	1	1	9	3
10	61	9	7	1	0	8	1
11	60	11	9	0	2	9	2
12	56	10	8	0	2	8	2
13	58	10	9	0	2	8	2
14	58	13	11	1	3	9	4
15	51	7	5	0	2	5	2
16	59	12	8	1	3	8	4
Total density		153	126	12	29	116	45
Reduction			17.6%			24.2%	

of the channel since that column can simply be ignored. Therefore, it is possible that the final density for this channel should be 9 instead of 10 in [7] (that would increase the number of ties by one). The overall density reduction for the RSD circuit is 24.2% for the new greedy router, compared with 17.6% for Cong et al. The running times for the new router varied from 0.070 to 0.110 seconds, with an average running time of 0.084 seconds.

Table IV shows the results for the Primary1 circuit. In the three cases (channels 3, 5 and 12) when the 5 track over-the-cell limit comes into play for Cong et al. the density for the HCVC case is shown in parentheses. The new router goes over the 5 track limit for one channel (channel 2) and there the solution for the unlimited number of over-the-cell tracks is shown in parentheses. This is indeed a valid solution for the HCVC layout model, since channel 2 is the lowest

TABLE IV Channel densities in Primary 1 after over-the-cell routing by Cong et al. and our new router.

				New Router			
Channel	Cols	Orig. density	[5] Inside	Top	Bottom	Inside	# OCC
2	264	19	14	0	5(7)	14(13)	5(8)
3	262	14	13(11)	3	3	9	17
4	206	18	12	3	5	11	11
5	272	25	22(21)	2	5	20	8
6	272	16	13	3	3	12	8
7	309	24	19	4	2	19	11
8	293	22	18	2	2	18	5
9	300	25	21	4	4	19	11
10	313	27	22	2	5	20	7
11	306	26	22(21)	3	4	19	7
12	254	19	15	2	4	14	10
13	247	19	15	1	3	15	4
14	318	21	15	1	4	16	7
Total density		275	221	30	51	206	114
Reduction			19.6%			25.1%	

channel in the circuit and only 7 tracks are used for over-the-cell routing (when 13 are available). The 5 track solution by the new router for channel 2 is just an intermediate step on the way towards the solution when there is no limit to the number of over-the-cell tracks. Therefore, a modification of the new greedy router that works with limited number of over-the-cell tracks might give a better solution (note that there is no routing over the top cell). Again, Cong et al. find a solution with lower final density in only one case (channel 14). The final densities are the same in 4 cases and the new router results in lower final densities for 8 channels. The overall density reduction for the Primary1 circuit is 25.1% for our greedy approach and 19.6% for the improved three stage approach. The running times for the new router on the channels in Primary1 varied from 0.340 to 0.600 seconds, with an average running time of 0.455 seconds.

5. CONCLUDING REMARKS

We described a new algorithm for over-the-cell channel routing. The new router tries to reduce the density inside the channel by moving segments that span maximum-density columns to the over-the-cell areas. The new router performs better than previous over-the-cell routers. The improved performance results from combining the routing of over-the-cell connections with the selection of connections to be made inside the channel and from more effective greedy heuristics for selecting the over-the-cell connections.

The worst case time complexity for the algorithm is quite high but the actual running time turns out to be quite good. We should note that in the channels in tables 3 and 4, which came from actual circuits, from 20% to nearly 50% of all the terminals were vacant (this is after elimination of all columns where both terminals are vacant). This cuts down on the number of OC-candidates that we create and therefore reduces the running time. The time complexity (and the running time, for sufficiently large channels) of our algorithm can be reduced by storing the densities inside the channel in a segment tree [17] instead of the simple array that we use. The time for updating the densities remains $O(s)$, but the query time (to find the maximum density and the number of maximum density columns over an interval) is reduced to $O(\log s)$. This leads to overall time complexity of $O(I\,N(s + D \log s) + D\,N\,n^2\,(n + \log s))$ instead of the previous $O(D\,N(I\,s + n(n^2 + s) + D))$.

There are a number of avenues for improving the algorithm. The most obvious of these is to make use of vacant terminals. Assuming that we do not use a vacant terminal assignment preprocessing step (such as [14]) a possibility for improvement is to increase the number of OC-candidates by using the vacant terminals. That is, in a column where one of the terminals is vacant, we allow the net for the active terminal to go straight across the channel and use the vacant terminal for a possibility of over-the-cell routing on the opposite terminal row.

The restriction of not allowing the overlap of two over-the-cell wires for the same net and on the same side (rule (ii) in maintaining the active set of OC-candidates) is unnecessary. It can be dropped without affecting the time complexity of the algorithm and that would increase the size of the set of active OC-candidates. It would also be interesting to see how much effect it would have to allow every possible two terminal OC-candidate in the initial active set of OC-candidates (instead of just candidates connecting nearest neighbors). However, this would clearly increase the time complexity of the algorithm by a factor of n.

We have observed an example where an OC-candidate chosen for routing has two possible reduction intervals, and the "wrong" one is chosen as its best reduction interval. That is, the maximum density columns in the reduction interval that was not chosen can not be covered by any other active OC-candidate while it is possible to cover the maximum density columns in the chosen interval by other OC-candidates. This indicates a weakness in our approach for choosing reduction intervals for OC-candidates. However, it is not clear how to fix this problem even if we associate the list of all possible maximum density reduction intervals with each OC-candidates.

Finally, the rules that we use for selecting the next OC-candidate for over-the-cell routing is an area worth studying. There is the question of whether it would be better to reduce the priority of maximizing the number of maximum density columns that are covered. We could try to create a weight function that balances the number of maximum density columns covered against the number of (or columns covered by) the OC-candidates that the candidate interferes with in a planar routing.

Acknowledgement

We thank the reviewers for their helpful suggestions.

References

[1] Chang, P.-Y. Hsiao, J.-T. Yan and P.-W. Shew, "A Robust Over-the-Cell Channel Router". *I1EEE Trans. Computer-Aided Design*, Vol. 12, No. 10, pp. 1592–1599, October 1993.

[2] J. Cong, *Private Communications*, 1992.

[3] H. H. Chen, "TRIGGER: A Three-Layer Gridless Channel Router". *Dig. Tech. Papers, IEEE Int. Conf. on Computer-Aided Design*, pp. 196–199, 1986.

[4] J. Cong and C.L. Liu, "Over-the-Cell Channel Routing," Digest of Technical Papers, *IEEE Int'l Conf. on Computer Aided Design*, pp. 80–83, 1988.

[5] J. Cong and C.L. Liu, "Over-the-Cell Channel Routing". *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, pp. 408–418, April 1990.

[6] J. Cong, B. Preas and C.L. Liu, "General Models and Algorithms for Over-the-Cell Routing in Standard Cell Design". *Proc. 27th Design Automation Conf.*, pp. 709–715, 1990.

[7] J. Cong, B. Preas and C.L. Liu, "Physical Models and Efficient Algorithms for Over-the-Cell Routing in Standard Cell Design". *IEEE Trans. Computer-Aided Design*, Vol. 12, No. 5, pp. 723–734, May 1993.

[8] J. Cong, D.F. Wong and C.L. Liu, "A New Approach to the Three Layer Channel Routing Problem". *Dig. Tech. Papers, IEEE Int. Conf. on Computer-Aided Design*, pp. 378–381, 1987.

[9] S. Das, S.C. Nandy and B.B. Bhattacharya, "An Improved Heuristic Algorithm for Over-the-Cell Routing". *Proc. 1991 IEEE Int. Symp. on Circuits and Systems*, vol. 5, pp. 3106–3109, 1991.

[10] Deutsch, D.N., "A Dogleg Channel Router", *Proc. of 13th Design Automation Conference*, pp. 425–433, 1976.

[11] G. Gudmundsson, "On Problems in Over-the-Cell Routing", Ph.D. Dissertation, University of Texas at Dallas, Dec. 1992.

[12] G. Gudmundsson and S. Ntafos, "Channel Routing with Superterminals". *Proc. 25th Allerton Conf. on Computing, Control and Communication*, pp. 375–376, 1987.

[13] A. Hashimoto and J. Stevens, "Wire Routing by Optimizing Channel Assignment within Large Apertures". *Proc. 8th Design Automation Workshop*, pp. 155–169, 1971.

[14] N.D. Holmes, N.A. Sherwani and M. Sarrafzadeh, "New Algorithm for Over-the-Cell Routing Using Vacant Terminals". *Proc. 28th Design Automation Conf.*, pp. 126–131, 1991.

[15] Kernighan, B.W., D.G. Schweikert and G. Persky, "An Optimum Channel-Routing Algorithm for Polycell Layouts of Integrated Circuits," *Proc. of 10th Design Automation Workshop*, pp. 50–59, 1973.

[16] M.-S. Lin, H.-W. Perng, C.-Y. Hwang and Y.-L. Lin, "Channel Density Reduction by Routing Over the Cells". *IEEE Trans. Computer-Aided Design*, vol. 10, no. 8, pp. 1067–1071, Aug. 1991.

[17] F.P. Preparata and M.I. Shamos, *Computational Geometry*. Springer-Verlag, New York, New York, 1985.

[18] B. Preas, "Benchmarks for Cell-Based Layout Systems". *Proc. 24th Design Automation Conf.*, pp. 319–320, 1987.

[19] R.L. Rivest and C.M. Fiduccia, "A "Greedy" Channel Router". *Proc. 19th Design Automation Conf.*, pp. 418–424, 1982.

[20] Y. Shiraishi and J. Sakemi, "A Permeation Router". *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 3, pp. 462–471, May 1987.

[21] K.J. Supowit, "Finding a Maximum Planar Subset of a Set of Nets in a Channel". *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 1, pp. 93–94, Jan. 1987.

[22] Yoshimura, T. and E.S. Kuh, "Efficient Algorithms for Channel Routing", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. CAD-1 (1), pp. 25–35, January 1982.

Authors' Biographies

Gudni Gudmundsson received his Ph.D. degree in Computer Science from the University of Texas at Dallas in 1992. Currently, he is a senior software engineer with Islandsbanki in Reykjavik, Iceland. His research interests include routing algorithms for VLSI design, and the design and implementation of data structures. He is a member of the IEEE and ACM.

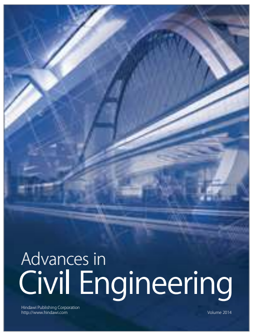
Simeon Ntafos received his Ph.D. in Computer Science from Northwestern University, Evanston, in 1979. Currently, he is a Professor in the Computer Science Program at the University of Texas at Dallas. His research interests include computational Geometry, program testing, software reliability and VLSI algorithms. He is a member of the ACM and EATCS.

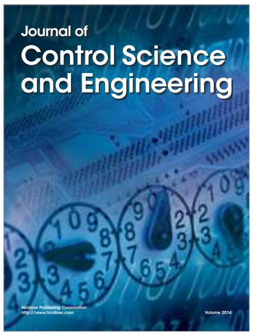
Journal of
Engineering

The Scientific World Journal

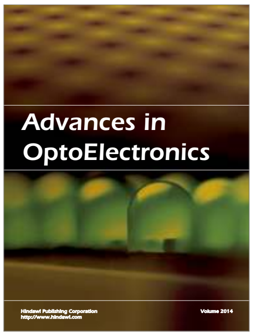
International Journal of
Rotating Machinery

Journal of
Sensors

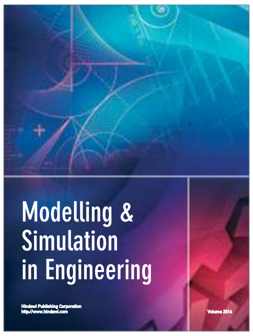
International Journal of
Distributed Sensor Networks

Advances in
Civil Engineering

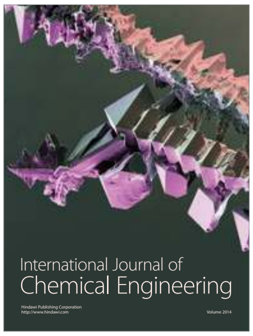
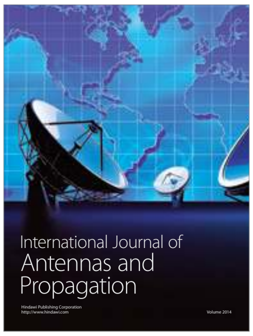

Journal of
Control Science and Engineering

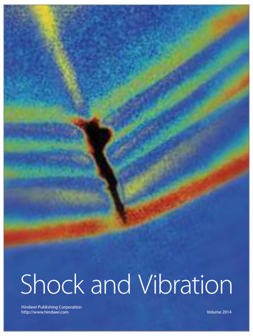
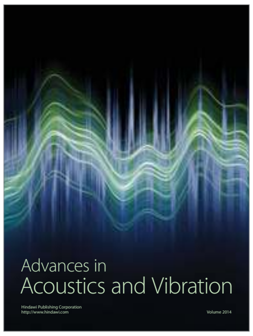
Journal of
Robotics

Journal of
Electrical and Computer Engineering

Advances in
OptoElectronics

VLSI Design

International Journal of
Navigation and Observation

Modelling & Simulation in Engineering

International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering

International Journal of
Antennas and Propagation

Active and Passive Electronic Components

Shock and Vibration

Advances in
Acoustics and Vibration