Segmented Channel Routing

Vwani P. Roychowdhury, Jonathan W. Greene, *Member, IEEE*, and Abbas El Gamal, *Senior Member, IEEE*

Abstract—**Novel problems concerning routing in a segmented routing channel are introduced. These problems are fundamental to routing and design automation for field programmable gate arrays (FPGA's), a new type of electrically programmable VLSI. The first known theoretical results on the combinatorial complexity and algorithm design for segmented channel routing are presented. It is shown that the segmented channel routing problem is in general NP-complete. Efficient polynomial time algorithms for a number of important special cases are presented.**

I. Introduction

CONVENTIONAL channel routing [1] concerns the assignment of a set of connections to tracks within a rectangular region. The tracks are freely customized by the appropriate mask layers. Even though the channel routing problem is in general NP-complete [4], efficient heuristic algorithms exist and are in common use in many placement and routing systems.

In this paper we investigate the more restricted channel routing problem (see Fig. 3), where the routing is constrained to use fixed wiring segments of predetermined lengths and positions within the channel. Such segmented channels are incorporated in channeled field programmable gate arrays (FPGA's) [3]. In [10], [11] we demonstrated that a well-designed segmented channel needs only a few tracks more than a freely customized channel. This leads us to believe that segmented channel routing is fundamental to routing for FPGA's.

The architecture of channeled FPGA's [3] is similar to that of conventional (mask programmed) gate arrays, comprising rows of logic cells separated by segmented routing channels (Fig. 1). The inputs and outputs of the cells each connect to a dedicated vertical segment. Programmable switches are located at each crossing of vertical and horizontal segments and also between pairs of adjacent horizontal segments in the same track. By programming a switch, a low-resistance path is created between the two crossing or adjoining segments.

A typical example of routing in a channeled FPGA is shown in Fig. 1. The vertical segment connected to the output of cell 3 is connected by a programmed switch to a horizontal segment, which, in turn, is connected to the input of cell 4 through another programmed switch. In order to reach the inputs of cells 1 and 2, two adjacent horizontal segments are connected to form a longer one.

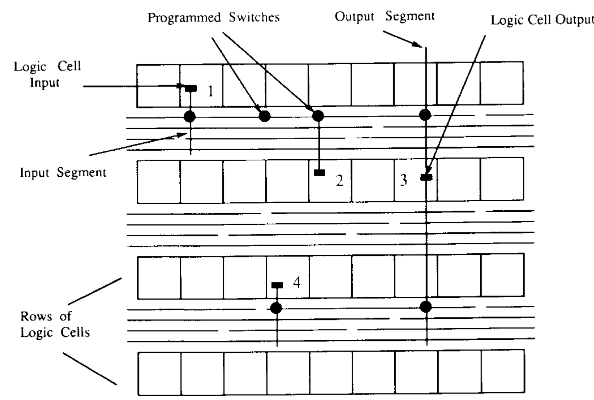


Fig. 1. FPGA routing architecture. ● denotes a programmed switch; unprogrammed switches are omitted for clarity.

The choice of the wiring segment lengths in a segmented channel is driven by tradeoffs involving the number of tracks, the resistance of the switches, and the capacitances of the segments. These tradeoffs are illustrated in Fig. 2.

Fig. 2(a) shows a set of connections to be routed. With the complete freedom to configure the wiring afforded by mask programming, the *left edge* algorithm [5] will always find a routing using a number of tracks equal to the density of the connections (Fig. 2(b)). This is the case since there are no "vertical constraints" in the problems we consider.

In an FPGA, achieving this complete freedom would require switches at every cross point. Furthermore, switches would be needed between each two cross points along a wiring track so that the track could be subdivided into segments of arbitrary length (Fig. 2(c)). Since all present technologies offer switches with significant resistance and capacitance, this would cause unacceptable delays through the routing. Another alternative would be to provide a number of continuous tracks large enough to accommodate all nets (Fig. 2(d)). Though the resistance is limited, the capacitance problem is only compounded, and the area is excessive.

A segmented routing channel offers an intermediate approach. The tracks are divided into segments of varying

Manuscript received January 29, 1991; revised February 10, 1992. V. P. Roychowdhury and A. El Gamal were supported in part by DARPA under Contract J-FBI-89-101. This paper was recommended by Associate Editor M. Marek-Sadowska.

V. P. Roychowdhury is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

J. W. Greene was with Actel Corporation, Sunnyvale, CA. He is now with BioCAD Corporation, Mountain View, CA 94043.

A. El Gamal is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305.

IEEE Log Number 9201137.

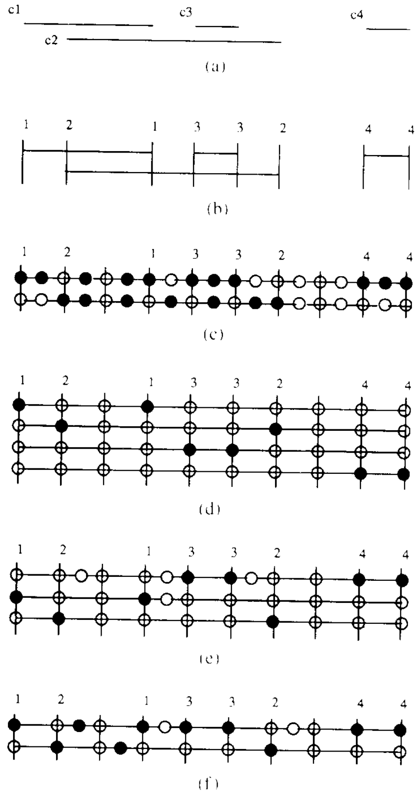


Fig. 2. Examples of channel routing. ○ denotes open switch; ● closed switch. (a) Set of connections to be routed. (b) Routing in unconstrained channels. (c) Routing in fully segmented channels. (d) Routing in an unsegmented channel. (e) Segmented for 1-segment routing. (f) Segmented for 2-segment routing.

lengths (Fig. 2(e)), allowing each connection to be routed using a single segment of the appropriate size. Greater routing flexibility is obtained by allowing limited numbers of adjacent segments in the same track to be joined end-to-end by switches (Fig. 2(f)). Enforcement of simple limits on the number of segments joined, or their total length, guarantees that the delay will not be unduly increased. Our results apply to the models of Fig. 2(e) and 2(f).

In Section II we formally define segmented channel routing and summarize the key results in the paper. Details of the algorithms and the proofs for theorems are given in Sections III–V and in the Appendix.

II. Definitions and Summary of Results

The input to a segmented channel routing problem, as depicted in Fig. 3, is a segmented channel consisting of a set $\mathfrak{T}$ of T tracks, and a set $\mathfrak{C}$ of M connections. The tracks are numbered from 1 to T. Each track extends from column 1 to column N, and is divided into a set of contiguous segments separated by switches. The switches are placed between two consecutive columns.

For each segment s, we define $left(s)$ and $right(s)$ to be the leftmost and rightmost columns in which the segment is present, $1 \leq left(s) \leq right(s) \leq N$. Each connection c_i, $1 \leq i \leq M$, is characterized by its leftmost and rightmost columns: $left(c_i)$ and $right(c_i)$. Without loss of generality, we assume throughout that the connections have been sorted so that $left(c_i) \leq left(c_j)$ for $i < j$.

A connection c may be *assigned* to a track t, in which case the segments in track t that are present in the columns spanned by the connection are considered *occupied*. More precisely, a segment s in track t is occupied by the connection c if $right(s) \geq left(c)$ and $left(s) \leq right(c)$. In Fig. 3 for example, connection c_3 would occupy segments s_{21} and s_{22} in track 2 or segment s_{31} in track 3.

Definition 1—Routing: A routing, R, of a set of connections is an assignment of each connection to a track such that no segment is occupied by more than one connection.

A *K-segment* routing is a routing that satisfies the additional requirement that each connection occupies at most K segments.

We can now define the following segmented channel routing problems:

Problem 1—Unlimited Segment Routing: Given a set of connections and a segmented channel, find a routing.

To reduce the delay through assigned connections, it may be desirable to limit the number of segments used for each connection.

Problem 2—K-Segment Routing: Given a set of connections and a segmented channel, find a K-segment routing.

If is often desirable to determine a routing that is optimal with respect to some criterion. We may thus specify a weight $w(c, t)$ for the assignment of connection c to track t, and define:

Problem 3—Optimal Routing: Given a set of connections and a segmented channel, find a routing which assigns each connection c_i to a track t_i such that $\Sigma_{i=1}^{M} w(c_i, t_i)$, is minimized.

For example, a reasonable choice for $w(c, t)$ would be the sum of the lengths of the segments occupied when connection c is assigned to track t. Note also that with appropriate choice of $w(c, t)$, Problem 3 subsumes Problem 2.

The problems defined above consider segmented channel routing with the restriction that each connection may be assigned only to a single track. It is easy to see that the routing capacity of a segmented channel may be increased if a connection is assigned to segments in different tracks. For example, consider the segmented channel routing problem in Fig. 4. It can be easily shown that if the assignment of each connection is constrained to a single track, successful routing does not exist. However, by assigning connection c_2 to segments s_{11} and s_{33}, which are located in tracks t_1 and t_3, successful routing may be achieved. We refer to such a routing as generalized routing.

Definition 2—Generalized Routing: A generalized routing R_G, of a *set* of connections consists of an assign-

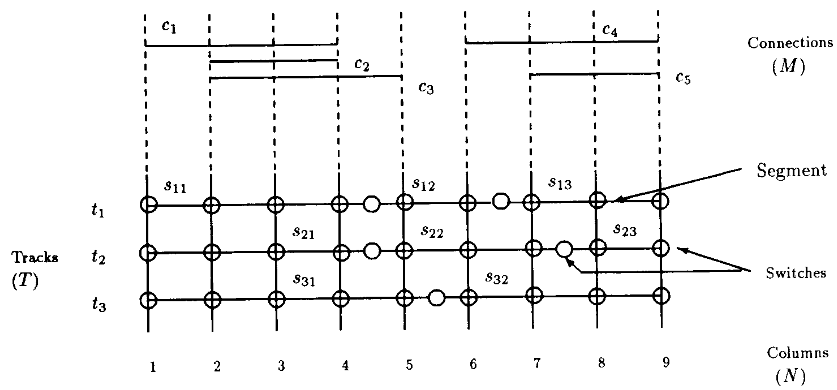


Fig. 3. An example of a segmented channel and a set of connections. $M = 5$, $T = 3$, $N = 9$. Connections: c_1, c_2, c_3, c_4, c_5. Segments: s_{11}, s_{12}, s_{13}, s_{21}, s_{22}, s_{23}, s_{31}, s_{32}.

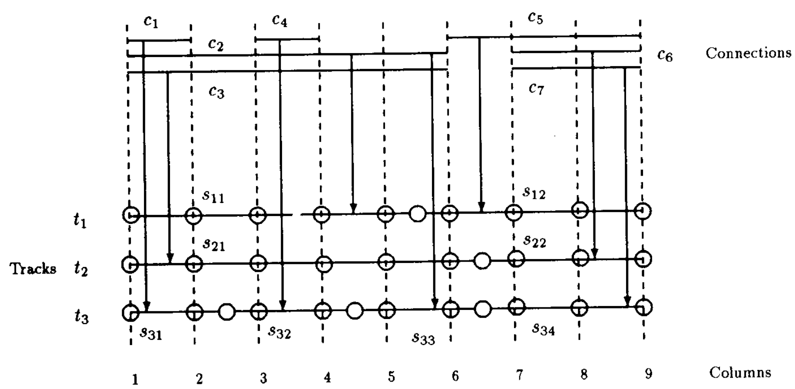


Fig. 4. An example where generalized routing is necessary for successful assignment.

ment of each connection to one or more tracks such that no segment is occupied by more than one connection.

Thus a generalized routing allows each connection $c = (left(c), right(c))$ to be *split* into $p (p \geq 1)$ parts: $(left(c), l_1)$, $(l_1 + 1, l_2)$, $(l_2 + 1, l_3)$, $\cdots$, $(l_{p-1} + 1, right(c))$, such that each part can be assigned to different tracks. A column l_i, where a connection is split, is referred to as a column where the connection c *changes* tracks.

Detailed hardware implementations may be developed to support generalized routing. For example, vertical wire segments may be added to facilitate track changing. In this case if a connection changes tracks, two switches must be programmed compared to only one if the connection is assigned to two contiguous segments in the same track. Thus allowing connections to occupy multiple tracks might lead to increase in area and to greater delays.

Motivated by such penalties, constraints may be imposed on the generalized segmented channel routing problem, leading to the following potentially important special cases.

1) Determine a generalized routing that uses at most k segments for routing any particular connection.

2) Determine a generalized routing that uses at most l different tracks for routing any connection.

3) Determine a generalized routing where connections can switch tracks only at predetermined columns.

We present preliminary results on the unconstrained version of generalized segmented channel routing problem.

Problem 4—Generalized Segmented Channel Routing: Given a set of connections and a segmented channel, find a generalized routing.

In this paper we establish the following results.

Theorem 1: Determining a solution to Problem 1 is strongly NP-complete.

Theorem 2: Determining a solution to Problem 2 is strongly NP-complete even when $K = 2$.

The reductions used to prove these theorems are rather tricky, and may have applications to problems in the area of task-scheduling on nonuniform processors. A proof of Theorem 1 is presented in Section III, and a proof of Theorem 2 is given in the Appendix.

We should note here that proving a given problem as NP-complete might not be enough to indicate its intractability. For example, many NP-complete problems such as Knap Sack have polynomial time solutions if all the input parameters are polynomially bounded in the input size. Strongly NP-complete problems however, remain

NP-complete even if input parameters are polynomially bounded; examples include TSP, Hamiltonian circuit, etc. (see [6] for detailed discussion on such issues). For Problems 1 and 2, the input parameters are indeed polynomially bounded in the number of columns (N) and tracks (T); for example, $M \leq TN$ and the lengths of the connections and the segments are bounded above by N. Hence, a proper approach would be to show that these problems are strongly NP-complete, which is what Theorems 1 and 2 establish.

Although Theorems 1 and 2 show that segmented channel routing is in general NP-complete, several special cases of the problem are tractable. We have developed polynomial-time algorithms for the following special cases:

Identically Segmented Tracks: Two tracks will be defined to be identically segmented if they have switches at the same locations, and hence, segments of the same length. The *left edge* algorithm used for conventional channel routing can be applied to solve Problems 1, 2, and 3.

1-Segment Routing: A routing can be determined by a linear time ($O(MT)$) greedy algorithm that exploits the geometry of the problem. The corresponding optimization problem can be also solved in polynomial time by reducing it to a weighted maximum bipartite matching problem.

At Most 2-Segments Per Track: If each track is segmented into at most two segments then also a greedy linear time algorithm (similar to the one for 1-Segment routing) can be designed to determine a routing.

We have also developed a general $O(T!M)$-time algorithm using dynamic programming for solving Problems 1, 2, and 3. This general algorithm can be adapted to yield more efficient algorithms for the following cases:

Fixed Number of Tracks: If the number of tracks is fixed, then the general algorithm directly yields a polynomial time algorithm.

K-Segment Routing: The general algorithm can be modified to yield an $O((K+1)^T M)$-time algorithm. Note that for small values of K the modified algorithm performs better than the general one.

Fixed Types of Tracks: If the number of tracks is *unbounded* but the tracks are chosen from a fixed set where T_i is the number of tracks of type i, then an $O((\Pi_1^l T_i^{K-2})M)$ time (hence, a polynomial-time) algorithm can be designed.

Furthermore, we have developed a heuristic algorithm based on linear programming for solving Problems 1 and 2 that appears to work surprisingly well in practice.

The general algorithm and the above-mentioned special cases are described in Section IV.

In Section V we present preliminary results on the generalized segmented channel routing problem. In particular we show that Problem 4 admits a polynomial time algorithm if the number of tracks is bounded. Determining the exact complexity of the generalized segmented channel routing problem remains an open problem.

III. Complexity of the Segmented Routing Problem

In this section we prove Theorem 1, i.e., determining a solution to Problem 1 is strongly NP-complete. The proof of Theorem 2, i.e., determining a solution to Problem 2 is strongly NP-complete even when $K = 2$, is presented in the Appendix. The NP-completeness reductions for both the theorems is from the *Numerical Matching Problem with Target Sums*, which has been shown to be strongly NP-complete [7].

Numerical Matching with Target Sums [7]*:* Given a set $S = \{1, \cdots, n\}$, and positive integers $x_1, \cdots, x_n, y_1, y_n, z_1, \cdots, z_n$ with

$$\sum_{i \in S} (x_i + y_i) = \sum_{i \in S} z_i$$

do there exist permutations α and β of S such that $x_{\alpha(i)} + y_{\beta(i)} = z_i$, for all $i \in S$?

We assume without loss of generality that $x_1 < x_2 < \cdots < x_n$, $y_1 < y_2 < \cdots < y_n$, and $z_1 < z_2 < \cdots < z_n$. Furthermore, we assume that for any instance of the problem, we have $x_{i+1} - x_i \geq n$ and $x_1 + y_1 \geq x_n + n$. If these conditions are not met for an instance of the problem then one can define an equivalent problem (i.e., the modified problem has a solution if and only if the original problem has a solution) for which the conditions are met by performing the following transformations:

1) Scaling: Define $m = \lceil n / min(x_i - x_{i-1}) \rceil$. If $m > 1$ then set $x_i \leftarrow mx_i$, $y_i \leftarrow my_i$, and $z_i \leftarrow mz_i$.

2) Translation: Define $p = x_n + n - (y_1 + x_1)$. If $p > 0$ then set $y_i \leftarrow y_i + p$, and $z_i \leftarrow z_i + p$.

Given an instance of the Numerical matching problem $\mathfrak{N}$, we now show how to construct an instance of Problem 1 in (pseudo)-polynomial time; we shall refer to the segmented channel and the set of connections generated by the reduction procedure as $\mathfrak{Q}$.

The set of connections $\mathfrak{C}$ is defined as follows.

1) For each x_i we define a connection a_i such that left(a_i) = 4, right(a_i) = $x_i + 3$. Thus, each connection a_i is of length $x_i - 1$, and starts at column number 4.

2) For each y_k, we define n connections $b_{k1}, \cdots, b_{kn}$ (one for each x_j) such that left(b_{kj}) = $x_j + 4 + (n - k)$ and right(b_{kj}) = $(y_k + x_j) + 4$. Note that right(b_{kj}) − left(a_j) = $x_j + y_k$.

3) n connections $d_1, \cdots, d_n$ are defined with left(d_i) = 1, and right(d_i) = 3.

4) $n^2 - n$ connections $e_1, \cdots, e_{n^2-n}$ are defined with left(e_i) = 1, and right(e_i) = 5.

5) n^2 connections $f_1, \cdots, f_{n^2}$ are defined with left(f_i) = $x_n + y_n + 5$ and right(f_i) = $x_n + y_n + 7$.

Set the number of columns in the construction to $N = x_n + y_n + 7$.

The set $\mathfrak{T}$ of n^2 tracks are then defined as follows.

1) For the first n tracks $t_1, \cdots, t_n$ each track t_i begins with a segment (1, 3), followed by unit-length segments that span the region from column 4 to column $z_i + 4$ (i.e., there is a switch between every two columns between col-

umn 4 and column $z_i + 4$), followed by a single segment of the form $(z_i + 5, N)$.

2) The rest of the $n^2 - n$ tracks are best described by dividing them into n blocks, each consisting of $n - 1$ tracks. Each such track comprises 3 segments.

The first block of $n - 1$ tracks, i.e., tracks $t_{n+1}, t_{n+2}, \cdots, t_{2n-1}$, are constructed using the definitions of the connections b_{1j}, $1 \le j \le n$. The segments in the track t_{n+1} are (1, left(b_{11}) − 1), (left(b_{11}), right(b_{12})), and (right(b_{12}) + 1, N). That is, the middle segment in the track t_{n+1} is defined such that the connections b_{11} or b_{12} can be assigned to it. In general, the segments in each track t_{n+j}, $1 \le j \le n - 1$, are defined as (1, left(b_{1j}) − 1), (left(b_{1j}), right($b_{1(j+1)}$)), and (right($b_{1(j+1)}$) + 1, N). That is, the middle segment in the track t_{n+j}, $1 \le j \le n - 1$, is designed such that the connections b_{1j} or $b_{1(j+1)}$ can be assigned to it.

The ith block of $n - 1$ tracks (i.e., tracks $t_{n+(i-1)(n-1)+1}, \cdots, t_{n+i(n-1)}$) is constructed using the definitions of the connections b_{ij}, $1 \le j \le n$. The segments in the track $t_{(n+(n-1)(i-1))+j}$ (i.e., the jth track in the ith block) are (1, left(b_{ij}) − 1), (left(b_{ij}), right($b_{i(j+1)}$)), and (right($b_{i(j+1)}$) + 1, N). That is, the middle segment in the track $t_{(n+(n-1)(i-1))+j}$ is designed such that the connections b_{ij} or $b_{i(j+1)}$ can be assigned to it.

The following example illustrates this construction.

Example 1: Consider the unlimited segment routing problem (see Fig. 5) corresponding to the instance of the Numerical matching problem with Target Sums:

$$x_1 = 2, x_2 = 5, x_3 = 8, \quad y_1 = 9, y_2 = 11, y_3 = 12,$$
$$z_1 = 11, z_2 = 17, z_3 = 19. \qquad \square$$

We might note here that our proof of the NP-complete reduction is geometric in nature and it is helpful to use the above example in understanding the statement and the proof of each of the following propositions and lemmas. Before we proceed, however, let us define the following.

Two connections c_1 and c_2 will be said to *overlap* if they are present in the same column(s), i.e., *left*(c_2) ≤ *left*(c_1) ≤ *right*(c_2) or *left*(c_1) ≤ *left*(c_2) ≤ *right*(c_1).

A connection c_1 is said to *fit* in a segment S_1 if *left*(c_1) ≥ *left*(S_1) and *right*(c_1) ≤ *right*(S_1).

A segment is said to be *available* for a set of connections if it is unoccupied by the rest of the connections in $\mathcal{C}$.

Proposition 1: In any routing R of $\mathcal{Q}$ the following prevail.

a) The connections f_i, $1 \le i \le n^2$, are assigned to n^2 different tracks.

b) The connections d_i, $1 \le i \le n$, and a_i, $1 \le i \le n$, are assigned to tracks $t_1, \cdots, t_n$, and connections e_i, $1 \le i \le n^2 - n$, are assigned to tracks t_{n+1} through t_{n^2}.

Proof: Claim a) follows directly from the construction; i.e., the connections f_i, $1 \le i \le n^2$ are all identical and overlapping.

Claim b) follows from the following observations that are based on the above reduction.

1) Each connection e_i, $1 \le i \le (n^2 - n)$, overlaps with every other e_i. Each e_i also overlaps with every connection d_j, $1 \le j \le n$ and every connection a_k, $1 \le k \le n$.

2) In tracks t_1 through t_n a d_i and a_j can be assigned to the same track; such assignment is not possible for tracks t_i, where $i > n$.

3) Finally, it follows from 1), 2) and from the pigeonhole principle that if any e_i is assigned to a track t_j, $j \le n$, then there would not be a sufficient number of tracks so as to assign all the connections d_i, $1 \le i \le n$, a_j, $1 \le j \le n$, and e_k, $1 \le k \le n^2 - n$. $\square$

Proposition 2: In any routing R of $\mathcal{Q}$, the segments available for assigning the connections a_i, $1 \le i \le n$, and b_{ij}, $1 \le i, j \le n$ are as follows.

a) In any track t_i, $1 \le i \le n$, the segments in columns 4 through $z_i + 4$ (i.e., the portion that is fully segmented) are available.

b) In any track t_i, $n + 1 \le i \le n^2$, only the middle segment is available.

Proof: Follows from Proposition 1: a) in any track t_i, $1 \le i \le n$, the first segment is always occupied by a d_j (for some $1 \le j \le n$), and the last segment is occupied by an f_k, hence the only available portion is the fully segmented part of the track; b) every track t_i, $n + 1 \le i \le n^2$, has only three segments, and from Proposition 1 we know that the left segment is occupied by a connection e_j (for some $1 \le j \le n^2 - n$) and the right segment by another connection f_k, $1 \le k \le n^2$. $\square$

The following proposition shows that in any routing R of $\mathcal{Q}$, every track has exactly one b_{ij} assigned to it.

Proposition 3: All connections b_{ij}, $1 \le i, j \le n$, overlap; hence, they have to be assigned to different tracks.

Proof: Given the geometry of our construction, it suffices to show that b_{11} and b_{1n} overlap. Now right(b_{11}) $= x_1 + y_1 + 4$, and left(b_{1n}) $= x_n + 4 + (n - 1) = x_n + n + 3$. Hence, right($b_{11}$) − left($b_{1n}$) $= x_1 + y_1 - (x_n + n - 1)$, which is strictly greater than 0 by our assumptions. $\square$

We can now show one direction of the reduction procedure.

Lemma 1: If the given Numerical Matching problem with target sums has a solution, then there exists a routing R for $\mathcal{Q}$.

Proof: Suppose there exist permutations α and β such that $x_{\alpha(i)} + y_{\beta(i)} = z_i$ for all $1 \le i \le n$. Then we can define a routing R for $\mathcal{Q}$ as follows.

1) Connections d_i, $1 \le i \le n$, e_i, $1 \le i \le n^2 - n$, and f_i, $1 \le i \le n^2$, are assigned according to Proposition 1.

2) For every i, $1 \le i \le n$, connections $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$ are assigned to track t_i. Since $x_{\alpha(i)} + y_{\beta(i)} = z_i$, one can easily show that the connections can be appropriately assigned in the available segments (see also Proposition 2).

At this stage, for every i, $1 \le i \le n$, all except one connection among the connections b_{ij}, $1 \le j \le n$, need to be routed.

3) Consider the connections b_{1j}, $1 \le j \le n$. Let b_{1k} be

the connection that has been assigned to one of the tracks t_i, $1 \le i \le n$. Recall that the tracks t_{n+1}–$t_{n+(n-1)}$ were designed using the definitions of b_{1j}, $1 \le j \le n$, and that the middle segment in track t_{n+1} can accommodate either connection b_{11} or connection b_{12}. So assign b_{11} to track t_{n+1} and repeat this procedure by assigning connections b_{12}–$b_{1(k-1)}$ to tracks t_{n+2}–$t_{n+(k-1)}$. Now b_{1k} has already been assigned, hence one has to assign connections $b_{1(k+1)}$–b_{1n}. By construction, however, $b_{1(k+1)}$ can be assigned to track t_{n+k}, and this assignment procedure can be continued by assigning $b_{1(k+2)}$ to track $t_{n+(k+1)}$, and so on.

In general, for any i the unassigned $n - 1$ connections among b_{ij}, $1 \le j \le n$ can be assigned to the ith block of tracks (i.e., tracks $t_{n+(i-1)(n-1)+1}, \cdots, t_{n+i(n-1)}$) by following the same procedure as above. □

Next we show that if Q has a valid routing then there is a solution for the numerical matching problem $\mathfrak{N}$. The following definitions that capture the geometry of the routing problem Q will be helpful:

It is clear from Propositions 1, 2, and 3 that each track t_l, $1 \le l \le n$ has one connection from a_i, $1 \le i \le n$ and one connection from b_{kj}, $1 \le k, j \le n$ assigned to it. Also, note that since the parts of the first n tracks that are available for the connections a_i, $1 \le i \le n$ and b_{kj}, $1 \le k, j \le n$ are fully segmented, two connections, a_i and b_{kj}, can be assigned to the same track only if they do not overlap.

We define the *length* or *space* occupied by the connections a_i and b_{kj} assigned to some track t_l $(1 \le l \le n)$ as equal to right(b_{kj})–left(a_i). That is, the length (or space) occupied by the two connections is the geometrical length from the left end of the connection a_i to the right end of the connection b_{kj}.

Claim 1: It follows from Proposition 2 that the total length (or space) available in the first n tracks for assigning the connections a_i, $1 \le i \le n$ and b_{ij}, $1 \le i, j \le n$ is $\Sigma_1^n z_i$.

The above claim follows immediately from the observation that the only portion of each track t_i to which a_i and b_{ij} can be assigned is of length z_i.

Proposition 4: Connections a_i and b_{kj} cannot be assigned to the same track if $j < i$.

Proof: left(b_{kj}) $= x_j + 4 + (n - k)$ and right(a_i) $= x_i + 3$. Thus right(a_i) $-$ left(b_{kj}) $= x_i - (x_j + n) + k - 1$. However, $(k - 1) \ge 0$, and by our assumptions $x_i - (x_j + n) \ge 0$ for all $j < i$. Hence, a_i and b_{kj} overlap for $j < i$. □

Proposition 5: If a_i and b_{kj} $(j \ge i)$ are assigned to the same track t_l $(1 \le l \le n)$ then the length occupied in the track t_l is $x_j + y_k$ $(\ge x_i + y_k)$.

Proof: Left(a_i) $= 4$, and right(b_{kj}) $= x_j + y_k + 4$. Hence, right(b_{kj}) $-$ left(a_i) $= x_j + y_k \ge x_i + y_k$ (because by our assumption $j \ge i$ implies that $x_j \ge x_i$). □

The next two propositions use the definitions of the tracks $t_{n+1}, \cdots, t_{n^2}$, and determine the restrictions on possible assignments of the connections b_{ij} to these tracks.

Proposition 6: None of the connections b_{kj} for $k > 1$ can be assigned to tracks t_{n+1}–$t_{n+(n-1)}$.

Proof: Recall that the tracks t_{n+1}–$t_{n+(n-1)}$ were constructed using the connections b_{1j}, $1 \le j \le n$. Now consider any track t_{n+l}. From Proposition 1 we know that its end segments are already occupied. Hence, for any b_{kj} to be assigned to this track, it must fit within the middle segment (left(b_{1l}), right ($b_{1(l+1)}$)).

First, consider the case where $k > 1$ and $j \le l$. Recall that left(b_{kj}) $= x_j + (n - k) + 4$; since $j \le l$, we have $x_j \le x_l$ and since $k > 1$, we can write left(b_{kj}) $= x_j + (n - k) + 4 < x_l + (n - 1) + 4 =$ left(b_{1l}). Hence, b_{kj} cannot be assigned to track t_{n+l}.

Next, consider the case where $k > 1$ and $j > l$. Recall that right(b_{kj}) $= x_j + y_k + 4$; since $j \ge (l + 1)$, we have $x_j \ge x_{l+1}$; furthermore, $k > 1$ implies that $y_k > y_1$. Hence, right(b_{kj}) $= x_j + y_k + 4 > x_{l+1} + y_1 + 4 =$ right($b_{1(l+1)}$). Therefore, b_{kj} cannot be assigned to track t_{n+l}. □

Proposition 7: In general, none of the connections b_{kj} for $k > i$ can be assigned to tracks $t_{n+(n-1)(i-1)+1}$–$t_{n+(n-1)i}$. Hence, none of the connections b_{kj} for $k > i$ can be assigned to tracks t_{n+1}–$t_{n+(n-1)i}$.

Proof: Recall that the tracks under consideration were constructed using the definitions of b_{ij}, $1 \le j \le n$. The proof then follows along the lines of the previous proposition. □

Let R be any routing of $\mathbb{Q}$, then we define m_i as follows:

$$m_i = |\{b_{ij} : 1 \le j \le n, \text{ and } b_{ij} \text{ is assigned to some track } t_l, 1 \le l \le n, \text{ in } R\}|.$$

In other words, m_i is the number of connections from the set $\{b_{i1}, b_{i2}, \cdots, b_{in}\}$ that are assigned to the first n tracks (i.e., $t_1, t_2, \cdots, t_n$). Propositions 8–10, following show that in any valid routing R of $\mathbb{Q}$ $m_i = 1$, for all $1 \le i \le n$.

Proposition 8: $\Sigma_1^k m_i \le k$, $\forall\ 1 \le k \le n$ and $\Sigma_1^n m_i = n$.

Proof: Each track has exactly one connection b_{ij} (for some i and j) assigned to it. Hence, by definition $\Sigma_1^n m_i = n$.

To show that $\Sigma_1^k m_i \le k$ for every $1 \le k \le n$, first consider $k = 1$. Suppose that $m_1 > 1$, then exactly $n - m_1$ connections from among the connections b_{1j}, $1 \le j \le n$ are assigned to tracks $t_{n+1} - t_{n^2}$. Even if all of them were assigned to tracks in the first block (i.e., among $t_{n+1}, \cdots, t_{n+n-1}$), there would be $(m_1 - 1) \ge 1$ tracks in the block that are left unassigned. However, by Proposition 6, no connection b_{ij}, when $i > 1$ can be assigned to any track among $t_{n+1}, \cdots, t_{n+n-1}$. Thus, at least $(m_1 - 1)$ tracks among $t_{n+1}, \cdots, t_{n+n-1}$ have no connection b_{ij} assigned to them. This leads to a contradiction (because every track has exactly one b_{ij} assigned to it).

Using Proposition 7, the same arguments can be applied for any $k > 1$. That is for $k = 2$, one can show (using Proposition 7) that if $m_1 + m_2 > 2$, then some tracks among t_{n+1} through $t_{n+2(n-1)}$ do not have any connection b_{ij} assigned to them. □

Proposition 9: Let $w_1 < w_2 < \cdots w_n$ be a sequence of positive integers and let non-negative integers m_i, $1 \le i \le n$, satisfy the following relations: $\Sigma_1^k m_i \le k$, $\forall\ 1 \le k \le n$, and $\Sigma_1^n m_i = n$. Then $\Sigma_1^n m_i w_i > \Sigma_1^n w_i$ if and only if some of the m_i are 0.

Proof: First we observe that if there exists an $m_j > 1$, then there exists $l = m_j - 1$ distinct variables $m_{j_1}, \cdots, m_{j_l}$, such that all of them are 0 and $j_i < j$. If not, then one can easily show that $\Sigma_1^j m_i > j$, which is a contradiction. Thus, if any of the variables $m_i > 1$, then it always forces some m_k to equal 0 such that $k < i$. Hence, $\Sigma_1^n m_i w_i > \Sigma_1^n w_i$ if and only if some of the m_i are 0. □

Proposition 10: In any routing R, $m_i = 1\ \forall\ 1 \le i \le n$, i.e., in every routing only one connection from the set $\{b_{i1}, \cdots, b_{in}\}$ is assigned to one of the first n tracks.

Proof: If a_i and b_{kj} are assigned to the same track then from Proposition 5 we know that the length occupied is $\ge x_i + y_k$. Now, by definition m_k connections from among b_{kj}, $1 \le j \le n$ appear in the first n tracks. Hence, the total length occupied by the connections a_i, $1 \le i \le n$, and the connections b_{ij}, $1 \le i, j \le n$ that are assigned in the first n tracks is $\ge \Sigma_1^n x_i + \Sigma_1^n m_k y_k$.

If at least one m_k is 0, then Proposition 9 implies $\Sigma_1^n m_k y_k > \Sigma_1^n y_k$ (because $y_1 < y_2 < \cdots y_n$). Hence, the total length occupied by the connections a_i and b_{ij} in the first n tracks is $> \Sigma_1^n x_i + \Sigma_1^n y_k = \Sigma_1^n z_i$. This leads to a contradiction because Proposition 2 and Claim 1 show that the total space available is equal to $\Sigma_1^n z_i$. Hence, $m_i = 1$ $\forall\ 1 \le i \le n$. □

Lemma 2: If there is a routing for $\mathcal{Q}$, then there exists a solution to $\mathfrak{N}$.

Proof: Proposition 10 shows that $\forall\ i$ only one connection among $\{b_{i1}, \cdots, b_{in}\}$ is assigned to one of the first n tracks. By Proposition 5, if a_i and b_{kj} $(j \ge i)$ are assigned to the same track then the length occupied is $x_j + y_k$ $(\ge x_i + y_k)$. Hence, the total length occupied by the connections is $\ge \Sigma_1^n x_i + \Sigma_1^n y_k = \Sigma_1^n z_i$.

Claim a: A connection a_i can only be assigned to the same track with some b_{ki}.

Proposition 4 shows that if a_i and b_{kj} are assigned to the same track, then $j \ge i$. Now if a_i is matched with some b_{kj} and $j > i$, then the length occupied is $x_j + y_k > x_i + y_k$. Hence, the total length occupied by all the connections in the first n tracks is greater than $\Sigma_1^n x_i + \Sigma_1^n y_k = \Sigma_1^n z_i$. However, this leads to a contradiction since the total space available in the first n tracks is $\Sigma_1^n z_i$ (Claim 1). Hence, a_i can be assigned only to the same track as some b_{ki}.

It follows then that if we define the connections assigned to track t_i, $1 \le i \le n$, as $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$, then α and β are permutations of the set $\{1, \cdots, n\}$. Also, by our convention the total length occupied in track t_i by $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$ is $= x_{\alpha(i)} + y_{\beta(i)}$.

Claim b: $x_{\alpha(i)} + y_{\beta(i)} = z_i$.

Suppose this is not the case for some i, $1 \le i \le n$. Then it implies that in the track t_i, the length occupied by the connections $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$ is $< z_i$. Now by Proposition 2, in any track t_k $(1 \le j \le n)$ the space available for assigning the connections a_i and b_{ij} is z_k. Hence, the length occupied by the connections a_i and the connections b_{ij} in the first n tracks is $< \Sigma_1^n z_i$. However, this leads to a contradiction because we showed that the length occupied is $\ge \Sigma_1^n z_i$.

Thus, the assignment of connections to the first n tracks defines permutations α and β such that $\forall\ i$, $x_{\alpha(i)} + y_{\beta(i)} = z_i$. □

Theorem 1: Determining a solution to Problem 1 is strongly NP-complete.

Proof: Follows from Lemmas 1 and 2: □

IV. Algorithms for Segmented Routing

In this section we present algorithms for various special cases of Problems 1–3. We first discuss algorithms that exploit the geometry of the segmented channels. We then discuss a general algorithm based on dynamic programming. Finally, we discuss a heuristic algorithm (based on linear programming) that appears to work surprisingly well in practice.

A. Geometrical Algorithms

Identically Segmented Tracks: If all tracks are identically segmented (i.e., the locations of the switches are the same in every track), then Problems 1 and 2 can be solved by the left-edge algorithm [5] in time $O(MT)$. Assign the connections in order of increasing left ends as follows: assign each connection to the first track in which none of the segments it would occupy are yet occupied.

Note that the density of the connections does not provide an upper bound on the number of tracks required for routing (as is the case for conventional routing when the left-edge algorithm is used in the absence of vertical constraints). However, if prior to computing the density the ends of each connection are extended until a column adjacent to a switch is reached, then the density would be a valid upper bound.

1-Segment Routing: If we restrict consideration to 1-segment routings, Problem 2 can be solved by the following greedy algorithm.

The connections are assigned in order of increasing left ends as follows. For each connection, find the set of tracks in which the connection would occupy one segment. Eliminate any tracks where this segment is already occupied. From among the remaining tracks, choose one where the unoccupied segment's right end is closest to the left (i.e., the right end coordinate of the segment in the chosen track is the smallest), and assign the connection to it. If there is a tie, then it is broken arbitrarily. In the example of Fig. 3, the algorithm assigns c_1 to s_{11}, c_2 to s_{21}, c_3 to s_{31}, c_4 to s_{32}, and c_5 to s_{13}. The time required is $O(MT)$.

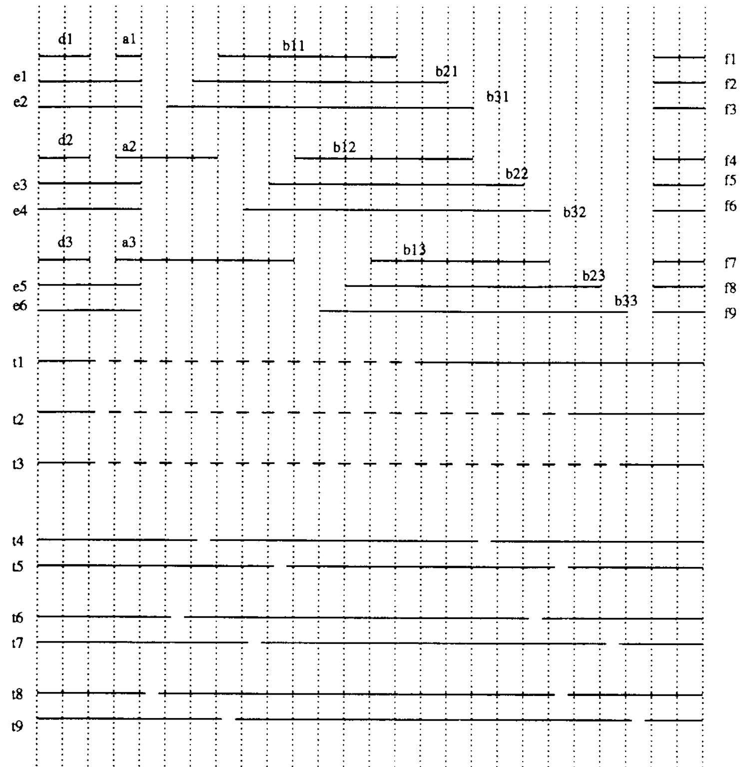


Fig. 5. The segmented channel and connections for Example 1.

Next, we show that if some connection cannot be assigned to any track, then no complete routing is possible.

Theorem 3: The above algorithm solves Problem 2 if $K = 1$.

Proof: It suffices to show that if there is a routing for Problem 2 with $K = 1$, then it can always be modified to obtain an assignment that the above algorithm would generate.

Let R be a routing with $K = 1$. Consider the leftmost connection c_1. Let F_1 be the set of segments that c_1 can be assigned to, and let S_1 be the set of segments in F_1 with the minimum right edge. There now are three possible cases.

1) In R, c_1 has been assigned to one of the segments in S_1. In such a case, no modification is necessary; the assignment of c_1 is according to the above algorithm.

2) In R, c_1 has been assigned to some $s \notin S_1$, and that there is at least one unoccupied segment in S_1. Then assign c_1 to one of the unoccupied segments in S_1.

3) In R, c_1 has been assigned to some $s \notin S_1$, and every segment in S_1 is occupied. In that case choose some c_i that occupies a segment $s_1 \in S_1$. We can now always interchange the assignments, i.e., assign c_i to s and assign c_1 to $s_1 \in S_1$. Thus a new assignment is obtained where c_1 is assigned according to the above algorithm.

The justification for swapping is as follows (see Fig. 6). Since $s_1 \in S_1$, c_1 can always be assigned to it. Moreover, the left edge of s is at or to the left of c_i (because left edge of c_i is at or to the right of c_1) and the right edge of s is to the right of c_i (because by definition of S_1 the right edge of s is to the right of s_1). Hence, c_i can be assigned to s.

The above procedure can be continued for c_2 and other connections until a modified routing R' is obtained that satisfies the conditions of the above algorithm. □

For 1-segment routing, Problem 3 may be solved efficiently by reducing it to a bipartite matching problem. Fig. 7 shows the graph corresponding to the routing problem in Fig. 3. The left side has a node for each connection and the right side a node for each segment. An edge is present between a connection and a segment if the connection can be assigned to the segment's track. The weight $w(c, t)$ is assigned to the edge between connection c and a segment in track t. A minimum-weight matching indicates an optimal routing. The time required using the best known matching algorithm (see [6]) is $O(V^3)$, where $V \leq M + NT$ is the number of nodes.

At Most 2-Segments Per Track: In a track with two-segments, the first segment from the left will be referred to as the *initial* segment and the next one will be referred to as the *end* segment. If the track is unsegmented, i.e., it has only one segment, then for our purposes we will refer to the only segment as an end segment.

The following greedy algorithm, which is similar to the

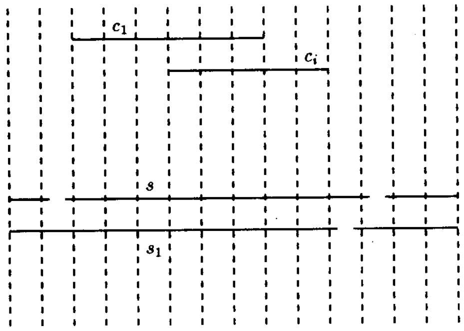


Fig. 6. An example of assignment modification in a 1-segment routing. If connection c_1 is assigned to s and c_i to s_1 such that $right(s) > right(s_i)$, then the assignment can always be swapped.

one for 1-segment routing, can be used to determine a solution to Problem 1:

The connections are assigned in order of increasing left ends (ties are resolved arbitrarily). During the execution of the algorithm a track will be considered *unoccupied* if no connection has been assigned to it.

Now for each connection, determine the set of tracks in which the connection would occupy a single segment. Eliminate any track where this segment is already *occupied*. Now consider the following two cases:

Case 1: If no track is available (i.e., after the above-mentioned elimination of tracks), then append the connection to the pool, P, of unassigned (but already examined) connections.

Case2: If tracks are available, then assign the connection to a track where the unoccupied segment's right end is closest to the left (i.e., the right-end coordinate of the segment in the chosen track is the smallest). If more than one track qualifies, then the tie is broken arbitrarily.

Next, if $|P|$ (i.e., the number of unassigned, but already examined, connections) equals the number of tracks unoccupied by any connection, then assign the connections in P to these unoccupied tracks in any order; mark these tracks as occupied, and remove the assigned connections from P. Else, if $|P|$ is greater than the number of such unoccupied tracks, then stop and signal that no valid routing is possible.

Continue with the next connection.

When all the connections are examined and pool P is nonempty, then assign the connections in P to unoccupied tracks.

In the example shown in Fig. 8, the above algorithm would assign c_1 to track t_1 and append c_2 to the pool P. For c_3, both tracks t_2 and t_3 are eligible, and let the tie be broken by assigning c_3 to track t_3. At this point, there is one unoccupied track (i.e., t_2) and there is one connection (i.e., c_2) in pool P. Hence, the number of unoccupied tracks equals the number of connections in P, and the algorithm would assign connection c_2 to the unoccupied track t_2. Next, the algorithm assigns c_4 to track t_1.

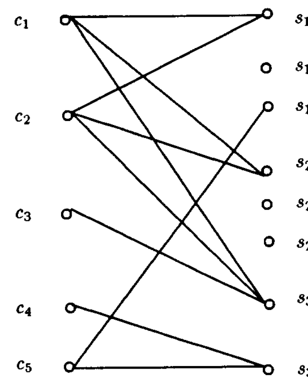


Fig. 7. Bipartite graph for 1-segment routing of the problem in Fig. 3.

Theorem 4: The above-mentioned algorithm determines a routing, if one exists, for the case where every track has at most two segments.

Proof: We shall provide an outline of the proof. Since every track has at most two segments, it follows that if a connection cannot be assigned to a single segment, then it has to occupy a whole track. Note also that the above-mentioned algorithm follows the greedy algorithm developed for 1-segment routing, and if a connection cannot be assigned to a single segment (by following the 1-segment routing algorithm) only then it is appended to the pool P.

The basic idea of the proof relies on the following observations. Since the algorithm developed for 1-segment routing was proved to be optimal, the connections in the pool P represent each of those connections that require a whole track. Moreover, these connections (i.e., which require whole tracks) are not assigned until a) all other connections are assigned to single segments and there are enough unoccupied tracks left to accommodate the connections in P; or b) during execution there are exactly as many unoccupied tracks as the number of connections in P (i.e., since, the connections in P must require whole tracks, these unoccupied tracks must be assigned to these connections). Thus the routing algorithm maximizes the connections that can be assigned to single segments and minimizes the connections that have to be assigned whole tracks.

A more rigorous proof, similar to the one for Theorem 3, can also be developed. More precisely, we can show that given any routing, one can always modify it such that the modified routing will be the same as one that the above-mentioned algorithm would generate. The details, however, get more involved; moreover, one may lose the intuitive appeal of the above explanations. □

B. A General Algorithm for Determining Routing

Although the problem of determining a routing for a given segmented channel and a set of connections is in general NP-complete, we describe below an algorithm that finds a routing in time linear in M (the number of connec-

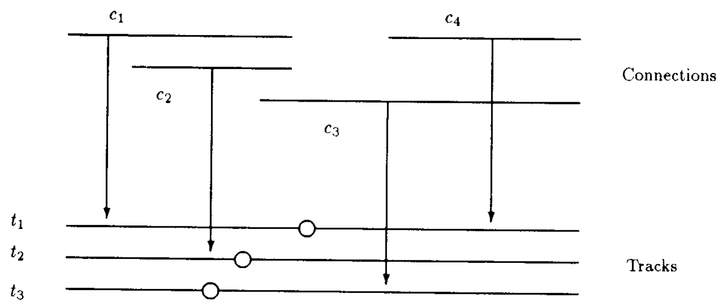


Fig. 8. An example of a routing with at most two segments per track.

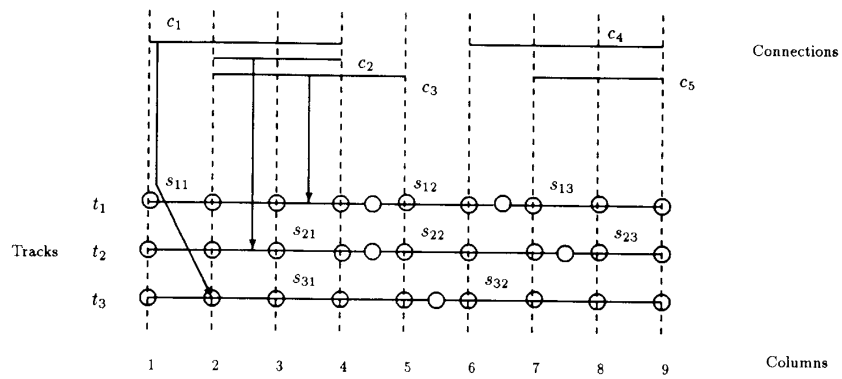


Fig. 9. A frontier for the example of Fig. 3. Connections c_1, c_2, and c_3 are assigned to segments s_{31}, s_{21}, and $\{s_{11}, s_{12}\}$, respectively. The frontier is $x = [7, 6, 6]$.

tions) when T (the number of tracks) is fixed. This is of interest since T is often substantially less than M. The algorithm may also be quite efficient when there are many tracks, but they are segmented in a limited number of ways (see Theorem 7). The algorithm first constructs a data structure called an *assignment graph* and then reads a valid routing from it. The same algorithm applies to both Problems 1 and 2, though with different time and memory bounds. It can also be extended to Problem 3.

Frontiers and the assignment graph: Given a valid routing for connections c_1 through c_i, it is possible to define a *frontier* which constitutes sufficient information to determine how the routing of $c_1 \cdots c_i$ may be extended to include an assignment of c_{i+1} to a track such that no segment occupied by any of c_1 through c_i will also be occupied by c_{i+1}. Fig. 9 shows an example of a frontier. It will be apparent that c_{i+1} may be assigned to any track t in which the frontier has not advanced past the left end of c_{i+1}. For example, in Fig. 9 connection c_4 can be assigned to track t_2 but not to track t_1.

More precisely, given a valid routing of $c_1, \cdots, c_i$, $1 \le i < M$, define the frontier $\boldsymbol{x}$ to be a T-tuple $(\boldsymbol{x}[1], \boldsymbol{x}[2], \cdots, \boldsymbol{x}[T])$ where $\boldsymbol{x}[j]$ is the leftmost unoccupied column in track t_j at or to the right of column $left(c_{i+1})$. (A column in track t_j is considered unoccupied if the segment present in the column is not occupied.) The frontier is thus a function $\boldsymbol{x} = F_i(t_{c_1}, \cdots, t_{c_i})$ of the tracks $t_{c_1}, \cdots, t_{c_i}$ to which $c_1 \cdots c_i$ are, respectively, assigned. For $i = 0$, let $\boldsymbol{x} = F_0$, where $F_0[t] = left(c_1)$ for all t. For $i = M$, let $\boldsymbol{x} = F_M$, where $F_M[t] = N + 1$ for all t.

Next, we describe a graph called the assignment graph, which is used to keep track of partial routings and the corresponding frontiers. A node at level i, $1 \le i < M$, of the assignment graph corresponds to a frontier resulting from some valid routing of c_1–c_i; see Fig. 10 for an illustration of the structure of an assignment graph. Level 0 of the graph contains the root node, which corresponds to F_0. If a complete valid routing for $c_1, \cdots, c_M$ exists, then level M of the graph contains a single node corresponding to F_M. Otherwise, level M is empty.

The assignment graph is constructed inductively. Given level $i \ge 0$ of the graph, construct level $i + 1$ as follows. (For convenience, we identify the node by the corresponding frontier.)

```
For each node x_i in level i {
   For each track t_j, 1 ≤ j ≤ T {
      If x_i[j] = left(c_{i+1}) {
         /* c_{i+1} can be assigned to track t_j.*/
         Let x_{i+1} be the new frontier after c_{i+1} is as-
         signed to track t.
         If x_{i+1} is not yet in level i + 1 {
            Add node x_{i+1} to level i + 1.
            Add an edge from node x_i to node x_{i+1}. Label
            it with t_j.
```

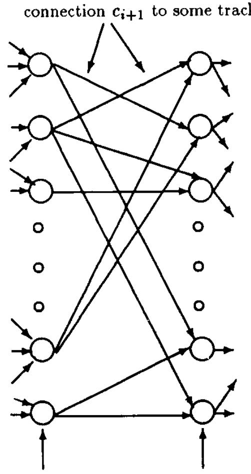


Fig. 10. An illustration of the structure of an assignment graph.

```
            }
        }
        Else {
            /* x_i[j] > left(c_{i+1}) so c_{i+1} cannot be assigned
            to track t_j. */
        Continue to next track t_{j+1}.
        }
    }
}
```

If there are no nodes added at level $i + 1$, then there is not valid assignment of c_1–c_{i+1}.

Searching for the node x_{i+1} in level $i + 1$ can be done in $O(T)$ time, using a hash table. Insertion of a new node in the table likewise requires time $O(T)$.

If there is a maximum of L nodes at each level, then construction of the entire assignment graph requires time $O(MLT^2)$. Once the assignment graph has been constructed, a valid routing may be found by tracing a path from the node at level M back to the root, reading the track assignment from the edge labels. (If there is no node at level M, then no complete valid assignment exists.) This takes only $O(M)$ time, so the overall time for the algorithm is $O(MLT^2)$. The memory required to store the assignment graph is $O(MLT)$.

A minor change allows us to solve the optimization problem as well. Each edge is labeled with the weight $w(c, t_c)$ of the corresponding assignment. Each node is labeled with the weight of its parent node plus the weight of the incoming edge. The algorithm is modified as follows. If a search in level $i + 1$ finds that the new node x_{i+1} already exists, we examine its weight relative to the weight of node x_i plus $w(c_{i+1}, t_{c_{i+1}})$. If the latter is smaller, we replace the edge entering x_{i+1} with one from x_i and update the weights accordingly. Thus the path traced back from the node at level M will correspond to a minimal weight routing. The order of growth of the algorithm's time remains the same, as does that of its memory.

Analysis for unlimited segment routing: The following theorem shows that for unlimited segment routing, $L \leq 2T!$, so that the time to construct the assignment graph and find an optimal routing is $O(MT^2 T!)$ and the memory required is $O(MTT!)$.

Theorem 5: For unlimited segment routing, the number of distinct frontiers that may occur for some valid assignment of c_1–c_i is at most $2T!$.

Proof: Let $l = left(c_{i+1})$. Let d be the number of connections among c_1 through c_i that are present in column l. Since the assignment of c_1 through c_i determining the frontier must be a valid one, we know that $d \leq T$. The d connections can be assigned to d of the T tracks in $T!/(T - d)!$ ways. Once we have assigned a connection to a track t_j, the value of $x[j]$ in that track is determined. For each of the remaining $(T - d)$ tracks, there are only two alternatives.

1) the track t_j may be unoccupied in column l, in which case, $x[j] = l$.

2) the track t_j may be occupied in column l by some connection c with $right(c) < l$, up to the first switch to the right of column l. In this case, $x[j]$ is the column just to the right of this switch, regardless of which such connection c is involved.

Thus the number of possible frontiers is at most $2^{(T-d)}T!/(T - d)! \leq 2T!$. □

Analysis for K-segment routing: The following theorem shows that for K-segment routing, $L \leq (K + 1)^T$, so that the time to construct the assignment graph and find an optimal routing is $O(MT^2(K + 1)^T)$ and the memory required is $O(MT(K + 1)^T)$.

Theorem 6: For K-segment routing, the number of distinct frontiers that may occur for some valid routing of c_1–c_i is at most $(K + 1)^T$.

Proof: Let $l = left(c_{i+1})$ and consider track t_i. Since the connections are sorted by increasing left edge, at most one connection from among $c_1 \cdots c_i$ may occupy track t_i in columns at or to the right of column l. Such a connection may occupy track t rightward through the segment appearing in column l, or through that segment plus the next one, or possibly as far as the Kth segment at or to the right of column l. Of course it is also possible that no connection from among c_1–c_i occupies the segment in column l of track t_i. Thus there are only $K + 1$ possible locations for the frontier $x[i]$ in track t_i, and at most $(K + 1)^T$ possible values for the frontier x overall. □

Case of many tracks of a few types: Suppose the T tracks fall into two types, with all tracks of each type segmented identically. Then two frontiers that differ only by a permutation among the tracks of each type may be considered equivalent for our purposes in that one frontier can be a precursor of a complete routing if and only if the other can. Thus we can restrict consideration to only one of each set of equivalent frontiers and strengthen the result of Theorem 6 as follows.

Theorem 7: Suppose there are T_1 tracks segmented in one way and $T_2 = T - T_2$ segmented another way. The number of distinct frontiers $\mathbf{x}$ that may occur for some valid K-segment routing of c_1–c_i, and that satisfy $\mathbf{x}[i] \le \mathbf{x}[j]$ for all $i < j$ with tracks t_i and t_j of the same type, is $O((T_1 T_2)^K)$.

Proof: As in Theorem 6, there are at most $K + 1$ possible values for $\mathbf{x}[i]$. Due to the inequality restriction (which eliminates all but one member of each set of equivalent frontiers), the number of possible frontiers is at most

$$\binom{T_1 + K}{T_1} \times \binom{T_2 + K}{T_2}$$

which for large T_1 and T_2 is $O((T_1 T_2)^K)$. □

It follows that a K-segment routing may be found in time $O(M(T_1 T_2)^K T^2)$, and memory $O(M(T_1 T_2)^K T)$.

The result of Theorem 7 may easily be generalized to the case of l types of tracks, in which case the time is $O(M(\Pi_1^l T_i^K))$, and the memory is $O(M(\Pi_1^l T_i^K)T)$.

C. A Linear Programming Approach

Problems 1 and 2 can be reduced to $0 - 1$ linear programming (LP) problems via a straightforward reduction procedure. The $0 - 1$ LP is in general NP-complete. For our purposes, however, such a reduction is interesting because our simulations showed (see [12]) that for almost all cases the corresponding $0 - 1$ LP problems could be solved by viewing them as *ordinary LP problems* for which efficient algorithms are known. In particular, our simulation results indicated that whenever a randomly generated instance of Problem 1 had a feasible solution, one could always find $0 - 1$ feasible solutions for the corresponding integer LP problem by solving it as an ordinary LP. The simulations were carried out for fairly large-sized instances, e.g., $M = 60$ and $T = 25$.

We now describe briefly the reduction procedure for Problem 1. The corresponding reduction for Problem 2 follows after minor modifications. Let us define binary variables x_{ij}, for $1 \le i \le M$, and $1 \le j \le T$ as follows: if $x_{ij} = 1$, then connection c_i is assigned to track t_j, else if $x_{ij} = 0$, then connection c_i is not assigned to track t_j. Since in a routing each connection is assigned to at most one track, one has the following constraints:

$$\sum_{j=1}^{T} x_{ij} \le 1, \qquad \forall\ 1 \le i \le M.$$

One also has to make sure that in any routing two connections assigned to the same track must not share a segment. Consider a track t_j; one can then easily determine sets of connections $P_{j1}, \cdots, P_{jl_j}$ (not necessarily disjoint) such that at most one from each set can be assigned to the track t_j. Hence for each such set P_{jk}, one must satisfy

$$\sum_{c_i \in P_{jk}} x_{ij} \le 1.$$

Finally, one must make sure that all the connections are routed; this can be ensured by maximizing the following objective function:

$$\sum_{i=1}^{M} \sum_{j=1}^{T} x_{ij}.$$

One can now easily verify the following facts about the above $0 - 1$ LP.

1) The objective function achieves the maximum value of M if there is a solution to Problem 1. This is because in a feasible routing each c_i is assigned to some track (thus there exists only one j such that $x_{ij} = 1$ for every i) and the constraints are never violated.

2) If the objective function achieves the value of M, then there is a solution to Problem 1. This follows directly from our construction of the $0 - 1$ LP.

Note that one can derive a $0 - 1$ LP for solving Problem 2 if one assigns $x_{ij} = 0$ whenever a connection c_i cannot be assigned to track t_j because it would require more than K segments.

V. An Algorithm for Determining Generalized Routing

We present here an algorithm for solving Problem 4. The algorithm has a time complexity of $O(T^{T+3}M)$, and is derived by modifying the construction of assignment graphs introduced in the last section. Thus, for a constant number of tracks the generalized segmented routing problem can be solved in time linear in M (the number of connections). We should note here that efficient algorithms for various special cases of the generalized segmented routing problem and results on their computational complexity remain as open problems.

Given an instance of the generalized segmented routing problem with a set of connections $\mathcal{C}$ (with M connections) and a set of tracks $\mathcal{T}$, the first step in our algorithm involves defining a new set of connections $\mathcal{C}'$ as follows:

For every connection $c_i = (left(c_i), right(c_i))$ in $\mathcal{C}$ we will define $p = right(c_i) - left(c_i) + 1$ connections in $\mathcal{C}'$, each spanning a single column. That is, the corresponding p connections in $\mathcal{C}'$ are: $(left(c_i), left(c_i))$, $(left(c_i) + 1, left(c_i) + 1), \cdots, (right(c_i), right(c_i))$.

Note that every connection in $\mathcal{C}'$ spans only a single column and the total number of connections in $\mathcal{C}'$ is at most MN (because each connection in $\mathcal{C}$ can generate at most N connections in $\mathcal{C}'$).

Proposition 11: A generalized segmented routing (as defined in Definition 2) for a set of connections $\mathcal{C}$ and a set of tracks $\mathcal{T}$ can be determined by finding a usual segmented routing (as defined in Definition 1) for the set of connections $\mathcal{C}'$ and the set of tracks $\mathfrak{T}$ if two connections in $\mathcal{C}'$ that are derived from the same parent connection in $\mathcal{C}$ are allowed to share the same segment.

Proof: The proof follows directly from the above construction. □

In order to determine a routing where certain connections from $\mathcal{C}'$ are allowed to occupy the same segment, we shall modify the construction of the assignment graph that was introduced in the previous section. As before, given a valid routing for connections $c_1, \cdots, c_i$ (in $\mathcal{C}'$), it is possible to define a *frontier* which constitutes sufficient information to determine how the routing of $c_1, \cdots, c_i$ may be extended to include an assignment of c_{i+1}. However, since connections in $\mathcal{C}'$ that originate from the same connection in $\mathcal{C}$ are allowed to occupy the same segment, it is not sufficient to just keep track of the occupancy of the segments (this is what is done in Section IV-B). In other words, one has to keep the additional information that would indicate whether connection c_{i+1} can be assigned to an already occupied segment. This can be done by storing the information that if a segment at a frontier is occupied, then which connection from $\mathcal{C}$ occupies it; *a segment will be said to be occupied by a connection* c_j *in* $\mathcal{C}$ if a connection in $\mathcal{C}'$ that is derived from c_j occupies the given segment.

More precisely, given a valid routing of $c_1, \cdots, c_i$ in $\mathcal{C}'$, define the frontier $\boldsymbol{x}$ to be a T-tuple $(\boldsymbol{x}[1], \boldsymbol{x}[2], \cdots, \boldsymbol{x}[T])$ where $\boldsymbol{x}[j] = (\boldsymbol{x}_1[j], \boldsymbol{x}_2[j])$. $\boldsymbol{x}_1[j]$ is defined as before, i.e., it is the leftmost unoccupied column in track t_j at or to the right of column $left(c_{i+1})$. (Recall that a column in track t_j is considered unoccupied if the segment present in the column is not occupied.) On the other hand, $\boldsymbol{x}_2[j]$ indicates that if the column $left(c_{i+1})$ is occupied (i.e., $\boldsymbol{x}_1[j] > left(c_{i+1})$) then which connection in $\mathcal{C}$ occupies it. $\boldsymbol{x}_2[j]$ can take two types of values:

1) $1 \le \boldsymbol{x}_2[j] \le M$: In this case the value of $\boldsymbol{x}_2[j]$ gives the connection in $\mathcal{C}$ that occupies the segment of the frontier (i.e., the segment present in column $left(c_{i+1})$) in track t_j. Thus, if c_{i+1} is derived from connection $c_{\boldsymbol{x}_2[j]}$ in $\mathcal{C}$ then c_{i+1} can be assigned to track t_j *irrespective* of the value of $\boldsymbol{x}_1[j]$.

2) $\boldsymbol{x}_2[j] = \phi$: This case would imply that whether c_{i+1} can be assigned to track t_j is determined *only* by the value of $\boldsymbol{x}_1[j]$. Thus if $\boldsymbol{x}_2[j] = \phi$ then c_{i+1} can be assigned to track t_j only if $\boldsymbol{x}_1[j] = left(c_{i+1})$.

Thus a frontier is a function $\boldsymbol{x} = F_i(t_{c_1}, \cdots, t_{c_i})$ of the tracks $t_{c_1}, \cdots, t_{c_i}$ to which $c_1, \cdots, c_i$ are respectively assigned. For $i = 0$, let $\boldsymbol{x} = F_0$, where $F_0[t] = (left(c_1), \phi)$ for all t. For $i = M$, let $\boldsymbol{x} = F_M$, where $F_M[t] = (N + 1, \phi)$ for all t.

As in Section IV-B, an assignment graph can now be used to keep track of the partial routings and the corresponding frontiers. A node at level i, $1 \le i < M$, of the assignment graph corresponds to a frontier resulting from some valid routing of c_1 through c_i. Level 0 of the graph contains the root node, which corresponds to F_0. If a complete valid routing for $c_1, \cdots, c_M$ exists, then level M of the graph contains a single node corresponding to F_M. Otherwise, level M is empty.

As in Section IV-B, the assignment graph is constructed inductively, and a modified algorithm for its construction can be stated as described below.

Given level $i \ge 0$ of the graph, construct level $i + 1$ as follows. (For convenience, we identify the node by the corresponding frontier.)

```
For each node x^i in level i {
  For each track t_j, 1 ≤ j ≤ T {
    If (x^i_1[j] = left(c_{i+1})) or (c_{i+1} is derived from
    c_{x^i_2[j]} in C) {
      /* c_{i+1} can be assigned to track t_j. */
      Let x^{i+1} be the new frontier after c_{i+1} is as-
      signed to track t.
      If x^{i+1} is not yet in level i + 1 {
        Add note x^{i+1} to level i + 1.
        Add an edge from node x^i to node x^{i+1}. Label
        it with t_j.
        }
    }
    Else {
      /* c_{i+1} cannot be assigned to track t. */
    Continue to next track t_{j+1}.
    }
  }
}
```

If there are no nodes added at level $i + 1$, then there is no valid assignment of c_1 through c_{i+1}.

Theorem 8: There is an $O(T^{T+3}M)$ time algorithm for solving Problem 4.

Proof: Recall from Section IV-B that if L is the maximum number of nodes at any level of the assignment graph then the time complexity of the above algorithm is $O(MLT^2)$. We will show here that $L = O(T^{T+1})$.

Let $l = left(c_{i+1})$, and consider a frontier $\boldsymbol{x} = (\boldsymbol{x}[1], \boldsymbol{x}[2], \cdots, \boldsymbol{x}[T])$ after a valid routing for connections $c_1, \cdots, c_i$. Recall that *every connection in* $\mathcal{C}'$ *spans a single column*. Hence in any track t_j, *only* the segment present in column l can either be occupied or unoccupied by connections $c_1, \cdots, c_i$; in other words, any segment to the right of the segment present in column l cannot be occupied by $c_1, \cdots, c_i$. Hence, $\boldsymbol{x}_1[j]$ can assume only *two values*, namely, $\boldsymbol{x}_1[j] = l$ or $\boldsymbol{x}_1[j]$ equals the column where the segment present in column l ends.

Let us next consider the possible values of $\boldsymbol{x}_2[j]$. We claim that, given a frontier, in order to correctly assign connection c_{i+1}, it is sufficient to know whether a segment at column l is occupied by connections (in $\mathcal{C}$) present *only* in column $l - 1$. This claim follows easily from the geometry of our segmented routing problem. In other words, connections in $\mathcal{C}$ were broken up into disjoint but contiguous units to generate connections of $\mathcal{C}'$. Hence, if c_{i+1} shares a segment with another connection then that connection must be derived from a connection in $\mathcal{C}$ that occupies column $l - 1$. Thus, in the frontier if a segment in t_j (spanning column l) is occupied by a connection present in column $l - 1$, then the value of this connection is stored in $\boldsymbol{x}_2[j]$; otherwise, $\boldsymbol{x}_2[j]$ is set to ϕ.

Let d be the connections present in column $l - 1$, then $\boldsymbol{x}_2[j]$ can take at most $d + 1$ values. We have already shown that $\boldsymbol{x}_1[j]$ can take at most two values. Hence, the maximum number of distinct frontiers possible is $2^T T^{d-1}$. However, since connections present at the same column has to be assigned to different tracks, $d \leq T$. Hence, $L \leq 2^T T^{T-1}$; in other words, $L = O(T^{T-1})$.

The above algorithm could be easily modified to solve the following restricted versions of the generalized segmented routing problem.

1) Each connection can switch tracks only at prespecified columns.

2) If a connection c switches from track t_1 to track t_2 at column l then the segments in the two different tracks (to which parts of c are assigned to) must be include l. It is easy to see that the algorithm described in this section might assign connections such that the segments in t_1 and t_2 to which parts of c, are assigned are separated by one column; this might not be desirable in certain hardware models.

We will not go into the details of the modifications, however, the general idea is as follows: the assignment graph as described above enumerates all possible routings, and restricted routings can be easily obtained by disallowing assignments that violate the premises.

VI. Concluding Remarks

We have introduced novel problems concerning the design and routing for segmented channels. We also have presented the first known theoretical results on the algorithm design, and combinatorial complexity of the routing problem for segmented channels. In particular, we showed that 1) the problem of determining a routing for a given segmented channels and connections is in general NP-complete; 2) efficient polynomial time algorithms can be designed for several special cases; and 3) efficient algorithms can be designed for some cases of a generalized segmented routing problem, where connections can occupy segments in different tracks.

There are several open issues in this new area of routing. For example, although we have developed efficient algorithms for many special cases of the routing problem (as listed in Section II), several other interesting cases are yet to be solved; following are some relevant ones: 1) channel length (N) is bounded, 2) connection lengths are bounded, and 3) connections are nonoverlapping. Also, efficient algorithms for the generalized routing problems are not known.

The routing scheme using segmented channels may also be considered as a model for a communication network in a multiprocessor architecture. The logic modules in Fig. 1 can be replaced by processing elements (PE's); the segmented routing network can then be used for dynamically reconfiguring interconnections among the PE's (by programming the appropriate switches as described for the FPGA's). In [8] a preliminary network model that uses specially segmented channels (referred to as express channels) has already been proposed. Tradeoffs similar to those discussed in Section I also appear to hold for such multiprocessor communication networks; however, this area needs further investigation.

Appendix

We showed in Section III that the unlimited segment routing problem is strongly NP-complete. We shall now use the instance of the unlimited segment routing problem that was used in proving Theorem 1 and reduce it to a 2-segment routing problem, thereby showing that the latter problem is also strongly NP-complete.

Let us briefly recall the construction of the unlimited segment routing problem $\mathcal{Q}$.

Given: Integers $x_1 < x_2 < \cdots x_n$, $y_1 < y_2 < \cdots < y_n$, and $z_1 < z_2 < \cdots < z_n$, such that 1) $\Sigma_{i \in S}(x_i + y_i) = \Sigma_{i \in S} z_i$ and 2) $x_{i+1} - x_i \geq n$ for every $1 \leq i \leq n - 1$ and $x_1 + y_1 \geq x_n + n$. For this section, without loss of generality, we shall further assume that $z_1 \geq x_n + n$.

The set of connections, $\mathcal{C}$, is then defined as follows.

1) For each x_i we defined a connection a_i such that $\text{left}(a_i) = 4$, $\text{right}(a_i) = x_i + 3$.

2) For each y_k, we defined n connections $b_{k1}, \cdots, b_{kn}$ (one for each x_j) such that $\text{left}(b_{kj}) = x_j + 4 + (n - k)$ and $\text{right}(b_{kj}) = (y_k + x_j) + 4$.

3) n connections $d_1, \cdots, d_n$ are defined with $\text{left}(d_i) = 1$, and $\text{right}(d_i) = 3$.

4) $n^2 - n$ connections $e_1, \cdots, e_{n^2-n}$ are defined with $\text{left}(e_i) = 1$, and $\text{right}(e_i) = 5$.

5) n^2 connections $f_1, \cdots, f_{n^2}$ are defined with $\text{left}(f_i) = x_n + y_n + 5$ and $\text{right}(f_i) = x_n + y_n + 7$.

The number of columns is set to $N = x_n + y_n + 7$.

The set $\mathcal{T}$ of n^2 tracks is then defined as follows:

1) For the first n tracks $t_1, \cdots, t_n$ each track t_i begins with a segment $(1, 3)$ followed by unit length segments that span the region from column 4 to column $z_i + 4$, followed by a single segment of the form $(z_i + 5, N)$.

2) The rest of the $n^2 - n$ tracks are best described by dividing them into n blocks, each consisting of $n - 1$ tracks. Each such track comprises three segments.

The first block of $n - 1$ tracks, i.e., tracks $t_{n+1}, t_{n+2}, \cdots, t_{2n-1}$, are constructed using the definitions of the connections b_{1j}, $1 \leq j \leq n$. The segments in each track t_{n+j}, $1 \leq j \leq n - 1$, are defined as $(1, \text{left}(b_{1j}) - 1)$, $(\text{left}(b_{1j}), \text{right}(b_{1(j+1)}))$, and $(\text{right}(b_{1(j+1)}) + 1, N)$. That is, the middle segment in the track t_{n+j} is defined such that the connections b_{1j} or $b_{1(j+1)}$ can be assigned to it.

The ith block of $n - 1$ tracks (i.e., tracks $t_{n+(i-1)(n-1)+1}, \cdots, t_{n+i(n-1)}$) is constructed using the definitions of the connections b_{ij}, $1 \leq j \leq n$. The segments in the track $t_{(n+(n-1)(i-1))+j}$ (i.e., the jth track in the ith block) are $(1, \text{left}(b_{ij}) - 1)$, $(\text{left}(b_{ij}), \text{right}(b_{i(j+1)}))$, and $(\text{right}(b_{i(j+1)}) + 1, N)$.

Given the above instance $\mathcal{Q}$ of unlimited segment routing, one can generate an instance $\mathcal{Q}_2$ of a 2-segment routing problem as described below.

The number of columns is set to the same value as in $\mathcal{Q}$, i.e., $N = x_n + y_n + 7$. The set of connections in the 2-segment problem $\mathcal{Q}_2$ is defined as follows:

1) The connections a_i, $1 \le i \le n$, e_i, $1 \le i \le n^2 - n$, and b_{ij}, $1 \le i, j \le n$ are defined as in the unlimited segment routing problem $\mathcal{Q}$.

The connections f_i are again defined as in $\mathcal{Q}$, except that there are now $2n^2 - n$ of them, i.e., $1 \le i \le 2n^2 - n$.

The connections d_i, defined in problem instance $\mathcal{Q}$, are omitted in $\mathcal{Q}_2$.

2) $n^2 - n$ new connections g_{ij}, where $1 \le i \le n$, and $1 \le j \le (n - 1)$, are added such that $\text{left}(g_{ij}) = 4$ and $\text{right}(g_{ij}) = z_i + 4$. Note that for a fixed value of i, all the $n - 1$ connections, g_{ij}, where $1 \le j \le (n - 1)$, are identical and have the same left and right end points.

The set of tracks (comprising $2n^2 - n$ tracks) is defined as follows:

1) Each track t_i, $1 \le i \le n$ in the construction of the unlimited segment routing problem $\mathcal{Q}$, is replaced by a set of n tracks that we label as t_{ij}, $1 \le j \le n$. Each such track comprises five segments. Let us first describe the five segments in the tracks, t_{1j}, $1 \le j \le n$: they are (1, 2), (3, 3), $(4, \text{right}(a_j))$, $(\text{right}(a_j) + 1, z_1 + 4)$, and $(z_1 + 5, N)$.

In general, for any $i (1 \le i \le n)$, the segments in the tracks, t_{ij}, $1 \le j \le n$, are defined as follows: (1, 2), (3, 3), $(4, \text{right}(a_j))$, $(\text{right}(a_j) + 1, z_i + 4)$, and $(z_i + 5, N)$.

2) The last $n^2 - n$ tracks, i.e., $t_{(n+1)}, \cdots, t_{n^2}$, in the unlimited segment routing is kept the same in the 2-segment routing problem, $\mathcal{Q}_2$.

Before we proceed, let us review the properties that routings must satisfy in the unlimited segment problem, which we proved in Section III.

1) In every track, t_i, $1 \le i \le n$, a connection a_i can only be assigned to the same track with some b_{ki} (see Lemma 2).

2) In every track, t_i, $1 \le i \le n$, the length occupied by connections $a_{\alpha(i)}$ and $b_{\alpha(i)\beta(i)}$ that are assigned to it is z_i. Note that the length occupied is defined as $\text{right}(b_{\alpha(i)\beta(i)}) - \text{left}(a_{\alpha(i)})$; (see Lemma 2).

Proposition 12: In any routing of $\mathcal{Q}_2$:

a) the connections e_i, $1 \le i \le n^2 - n$, are assigned to tracks $t_{n+1}, \cdots, t_{n^2}$, i.e., the last $n^2 - n$ tracks.

b) the connections f_i, $1 \le i \le 2n^2 - n$, occupy the last segment in every track.

c) the connections a_i, $1 \le i \le n$, are assigned to tracks t_{ij}, $1 \le i, j \le n$.

d) the connections g_{ij}, $1 \le i \le n$, $1 \le j \le n - 1$ cannot be assigned to tracks $t_{n+1}, \cdots, t_{n^2}$.

e) only n connections from b_{ij}, $1 \le i, j \le n$, can be assigned to tracks t_{ij}, $1 \le i, j \le n$, and the rest of the $n^2 - n$ connections are assigned to tracks $t_{n+1}, \cdots, t_{n^2}$. Thus, each track among $t_{n+1}, \cdots, t_{n^2}$, has one connection from b_{ij}, $1 \le i, j \le n$, assigned to it.

Proof: In Proposition 12, a) follows directly from the construction: if an e_i is assigned to any track among t_{ij}, $1 \le i, j \le n$, then it would occupy three segments: this is not permitted in 2-segment routings. Hence, the first segment in each of the last $n^2 - n$ tracks must be occupied by an e_i connection.

b) follows again from the construction: all the connections f_i are overlapping and there are as many of these connections as the total number of tracks. Hence, they occupy the last segment in each track.

c) follows from a): the connections a_i and e_j overlap for every i and j; hence, none of the a_i connections can be assigned to tracks $t_{n+1}, \cdots, t_{n^2}$.

d) again follows from a): every connection g_{ij} overlaps with every connection e_k. Since the connections e_k are assigned to the last $n^2 - n$ tracks, the connections g_{ij} must be assigned to the tracks t_{ij}, $1 \le i, j \le n$.

e) follows from d): every b_{ij} overlaps with every g_{ij} (because by assumption $z_1 \ge x_n + n$); since all the g_{ij} ($n^2 - n$ of them) are assigned to the top t_{ij}, $1 \le i, j \le n$ tracks, there are only n tracks left that connections b_{ij} can be assigned to. This also implies that each track among $t_{n+1}, \cdots, t_{n^2}$, has one connection from b_{ij}, $1 \le i, j \le n$ assigned to it. □

Proposition 13: The total length required by the connections among a_i and b_{ij} that are assigned to n tracks among t_{ij}, $1 \le i, j \le n$, is $\ge \Sigma_{i=1}^n z_i$.

Proof: Since the connections a_i, b_{ij} and the tracks $t_{n+1}, \cdots, t_{n^2}$ are defined identically in problems $\mathcal{Q}$ and $\mathcal{Q}_2$, Propositions 4, 5, 6, and 7 (proved in Section III) are also true for $\mathcal{Q}_2$.

If R_2 is any routing for $\mathcal{Q}_2$, then we can define a quantity l_i (similar to m_i defined in Section III) as follows:

$$l_i = |\{b_{ij}\colon 1 \le j \le n, \text{ and } b_{ij} \text{ is assigned to some track } t_{kl}, 1 \le k, l \le n, \text{ in } R_2\}|.$$

In other words, l_i is the number of connections from the set $\{b_{i1}, b_{i2}, \cdots, b_{in}\}$ that are assigned to tracks t_{kl}. We can now exactly follow the arguments of Propositions 8–10 and show that a) $\Sigma_{i=1}^k l_i \le k$ for all $1 \le k \le n$, and $\Sigma_{i=1}^n l_i = n$; b) the total length occupied by the connections a_i and b_{ij} in the tracks t_{ij}, $1 \le i, j \le n$, is $> \Sigma_1^n x_i + \Sigma_1^n l_k y_k$, and c) finally, (using Proposition 9 and the arguments in Proposition 10)

$$\sum_1^n x_i + \sum_1^n l_k y_k \ge \sum_1^n x_i + \sum_1^n y_k = \sum_{i=1}^n z_i$$

where equality is met if and only if $l_k = 1$ for all k. □

The next proposition shows that among n tracks t_{ij}, $1 \le j \le n$ (i is fixed), there is exactly one track that can be occupied by connections a_k and b_{lm}; the rest are occupied by $n = 1$ connections g_{ii}, $1 \le j \le n - 1$.

Proposition 14: In any routing of $\mathcal{Q}_2$, for any fixed i, the $(n-1)$ connections g_{ij}, $1 \le j \le (n-1)$, can only be assigned to tracks in the set of t_{ik}, $1 \le k \le n$.

Proof: Let us define h_i to be the number of tracks among t_{ij}, $1 \le j \le n$, that are unoccupied by the connections g_{kl}, $1 \le k \le n$, $1 \le l \le (n-1)$. These are also the tracks available for connections a_i and b_{ij}; hence, $\Sigma_{i=1}^{n} h_i = n$. Moreover, the total space available in these tracks (for connections a_i and b_{ij}) is $\Sigma_{i=1}^{n} h_i z_i$ (because the total length provided by any track t_{ij} for connections a_i and b_{ij} is z_i).

Next, let us observe that any connections g_{ij} can never be assigned to a track t_{kl}, where $k < i$. This is because the connection g_{ij} is defined as $(4, z_i + 4)$, and one can easily verify that if it is assigned to track t_{kl}, $k < l$, then it would occupy three segments which is not permitted in a 2-segment routing. Using the above property we can show that

$$\sum_{j=1}^{i} h_{(n+1)-j} \le i, \qquad \forall 1 \le i \le n.$$

For example, for $i = 1$ the above relationship follows easily: none of the connections g_{nj} can be assigned to tracks t_{kl} where $k \le n$, hence, all of them have to be assigned to tracks t_{nl} (see Proposition 12); thus, the maximum number of tracks unoccupied by g_{nj}, $1 \le j \le n-1$, among t_{nk}, $1 \le k \le n$, is at most 1, or equivalently, $h_n \le 1$. For other values of i, the above relationship can be showed by induction.

Now, we know that $z_n > z_{n-1} > \cdots > z_1$; using this property and the fact that 1) $\Sigma_{j=1}^{i} h_{(n+1)-j} \le i$, $\forall 1 \le i \le n$; b) $\Sigma_{i=1}^{n} h_{(n+1)-i} = n$, we can easily derive (applying arguments analogous to those in Propositions 9 and 10) that $\Sigma_{i=1}^{n} z_i h_i \le \Sigma_{i=1}^{n} z_i$ and the equality results if and only if $h_i = 1$ for all $1 \le i \le n$.

Thus if $h_i \ne 1$ for all i then it leads to a contradiction with Proposition 13.

Note that we already showed that all the connections g_{nj}, $1 \le j \le n-1$, have to be assigned to tracks t_{nk}, $1 \le k \le n$. Now, $h_n = 1$, hence, for connections $g_{(n-1)j}$, $1 \le j \le n-1$, the only available tracks are $t_{(n-1)k}$, $1 \le k \le n$. Since $h_{n-1} = 1$, the same arguments can be continued to show that the $(n-1)$ connections g_{ij}, $1 \le j \le (n-1)$, can only be assigned to tracks in the set of t_{ij}, $1 \le k \le n$. □

Theorem 2: Determining a solution to Problem 2 is strongly NP-complete even when $K = 2$.

Proof: First let us show that if there is a solution to the unlimited segment routing problem for $\mathcal{Q}$, then there is a solution to the 2-segment routing problem for $\mathcal{Q}_2$. The assignments for $\mathcal{Q}_2$ are as follows:

1) The connections e_i, $1 \le i \le n^2 - n$ are assigned to tracks $t_{n+1}, \cdots, t_{n^2}$. Since, the last $n^2 - n$ tracks are identical in both instances and e_i gets assigned to single segments in every track, this is a valid step.

The connections f_i, $1 \le i \le 2n^2$ are assigned the last segment in every track.

2) Since, the last $n^2 - n$ tracks are identical in both $\mathcal{Q}$ and $\mathcal{Q}_2$ (and so are the connections b_{ij}), the connections b_{ij} assigned to the these tracks in $\mathcal{Q}$ are also assigned to the same tracks in $\mathcal{Q}_2$. This leaves n connections among b_{ij}, $1 \le i, j \le n$ to be routed (precisely those which are assigned to tracks $t_1, \cdots, t_n$ in the routing for $\mathcal{Q}$).

3) Next consider the connections a_i and b_{ij} that are assigned to the first n tracks $t_1, \cdots, t_n$ in $\mathcal{Q}$. First consider, t_1, and let the connections assigned to it be $a_{\alpha(1)}$ and $b_{\beta(1)\alpha(1)}$ (recall from Section III that a connection a_i can only be assigned to the same track with some b_{ki}). Now consider the track $t_{1\alpha(1)}$ in $\mathcal{Q}_2$, it has a segment (4, right $(a_{\alpha(1)})$) to which the connection $a_{\alpha(1)}$ can be assigned and a segment (right$(a_{\alpha(1)}) + 1$, $z_i + 4$) to which the connection $b_{\beta(1)\alpha(1)}$ can be assigned. Next, the $n-1$ connections g_{1j}, $1 \le j \le n-1$ can be assigned to the $n-1$ tracks among t_{1j}, $1 \le j \le n$ that are not occupied by the connections $a_{\alpha(1)}$ and $b_{\beta(1)\alpha(1)}$.

This procedure can be continued, i.e., consider track t_i in $\mathcal{Q}$ and let $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$ be the connections assigned to it. Then for a routing of Q_2, assign the connections $a_{\alpha(i)}$ and $b_{\beta(i)\alpha(i)}$ to track $t_{i\alpha(i)}$. To the rest of the $(n-1)$ tracks among t_{ij} assign the connections g_{ij}, $1 \le j \le n-1$.

One can easily verify that after following the above three steps, all the connections of $\mathcal{Q}_2$ are appropriately routed.

We now state how to get a routing for $\mathcal{Q}$ given a routing for $\mathcal{Q}_2$.

1) Assign d_i, $1 \le i \le n$, f_i, $1 \le i \le n^2$ and e_i, $1 \le i \le n^2 - n$ according to Proposition 1.

2) The $n^2 - n$ connections among b_{ij}, $1 \le i, j \le n$ that are assigned to tracks $t_{n+1}, \cdots, t_{n^2}$ in $\mathcal{Q}_2$ are assigned to the identical tracks in $\mathcal{Q}$.

3) After the above steps, one is left with the connections a_i, $1 \le i \le n$, and n connections among b_{ij} (precisely those that are assigned to tracks t_{ij} in $\mathcal{Q}_2$) that need to be assigned.

Consider the particular track among t_{1j}, $1 \le j \le n$ in $\mathcal{Q}_2$ (note that by Proposition 14 there always exists such a track), that has one connection each from a_i and b_{ij} assigned to it and let these connections be $a_{\alpha(1)}$ and $b_{\alpha(1)\beta(1)}$. Then in $\mathcal{Q}$ assign $a_{\alpha(1)}$ and $b_{\alpha(1)\beta(1)}$ to track t_1 (the validity of this assignment follows immediately from the construction of the track t_1).

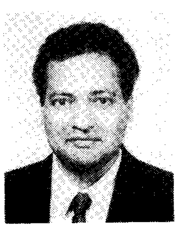
In general, let $a_{\alpha(i)}$ and $b_{\alpha(i)\beta(i)}$ be the connections assigned to one track among the n tracks t_{ij} $1 \le j \le n$. Then assign $a_{\alpha(i)}$ and $b_{\alpha(i)\beta(i)}$ to track t_i in $\mathcal{Q}$. □

REFERENCES

[1] M. Lorenzetti and D. S. Baeder, "Routing," *Physical Design Automation of VLSI Systems*, vol. B. Preas and M. Lorenzetti, Eds. Menlo Park, CA: Benjamin/Cummings, 1988, chap. 5.

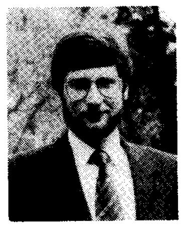
[2] H. Hsieh *et al.*, "A second generation user programmable gate array," in *Proc. Custom Integrated Circuits Conf.*, May 1987, pp. 515–521.

[3] A. El Gamal, J. Greene, J. Reyneri, E. Rogoyski, K. El-Ayat, and A. Mohsen, "An architecture for electrically configurable gate arrays," *IEEE J. Solid-State Circuits*, vol. 24, pp. 394–398, Apr. 1989.
[4] T. Szymanski, "Dogleg channel routing is NP-complete," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 31–41, Jan. 1985.
[5] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th IEEE Design Automation Workshop*, 1971.
[6] C. H. Papadimitrou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[7] M. R. Garey and D. S. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
[8] W. J. Dally, "Express cubes: Improving the performance of the k-ary n-cube interconnection networks," MIT, Cambridge, MA, VLSI Memo 89-564, Oct. 1989.
[9] A. El Gamal, "Two dimensional stochastic model for interconnections in master slice integrated circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 127–128, Feb. 1981.
[10] A. El Gamal, J. Greene, and V. P. Roychowdhury, "Segmented channel routing is as efficient as conventional routing (and just as hard)," in *Proc. 13th Conf. Advanced Research VLSI*, Santa Cruz, CA, Mar. 1991, pp. 192–211.
[11] J. Greene, V. P. Roychowdhury, S. Kaptanoglu, and A. El Gamal, "Segmented channel routing," in *Proc. 27th ACM/IEEE Design Automation Conf.*, June 1990, pp. 567–572.
[12] T. Varvarigou, "A linear programming approach to segmented channel routing," Dept. Elect. Eng., Stanford Univ., Stanford, CA, Project Rep. EE391.



Vwani P. Roychowdhury received the B.Tech. degree from the Indian Institute of Technology, Kanpur, India and the Ph.D. degree from Stanford University, Stanford, CA, in 1982 and 1989, respectively, all in electrical engineering.

From September 1989 to August 1991 he was a research associate in the Department of Electrical Engineering, Stanford University. He is currently an Assistant Professor in the Electrical Engineering Department, Purdue University, West Lafayette, IN. His research interests include parallel algorithms and architectures, design and analysis of neural networks, special purpose computing arrays and VLSI design, and fault-tolerant computation.



Jonathan W. Greene (S'80–M'84) received the Sc.B. degree in biology from Brown University, Providence, RI, in 1979, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1983.

He is currently with BioCAD Corp., Mountain View, CA, applying computer-aided design and information theoretic techniques to the process of drug discovery. He has worked on various aspects of computer-aided IC design at Hewlett-Packard Laboratories and the LSI Logic Systems Research Lab. From 1986 to 1990 he was with Actel Corporation, where he helped develop the architecture and software for their field-programmable gate arrays and became Director of System Architecture. He has also been a Visiting Scholar in the Information Systems Laboratory at Stanford University, and at Columbia University, NY, Chemistry Department.

Abbas El Gamal (S'71–M'73–SM'83) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1978.

He is currently an Associate Professor of Electrical Engineering at Stanford University. From 1978–1980 was an assistant professor of electrical engineering at the University of Southern California, Los Angeles, CA. From 1981–1984, he was an assistant professor of electrical engineering at Stanford. He was on leave from Stanford from 1984 to 1987, first as director of the LSI Logic Research Lab, then as founder and Chief Scientist of Actel Corp. His research interests include semi-custom VLSI architectures, design automation and synthesis for VLSI, configurable VLSI, complexity theory, and information theory.