

Routing Tree Topology Construction to Meet Interconnect Timing Constraints

Huibo Hou
Department of ECE
Iowa State University
Ames, IA 50011.

Sachin S. Sapatnekar
Department of ECE
University of Minnesota
Minneapolis, MN 55455.

Abstract

This work presents a Steiner tree construction procedure, MVERT, to meet specified sink arrival time constraints. It is shown that the optimal tree requires the use of non-Hanan points. The procedure works in two phases: a minimum-delay Steiner tree is first constructed, after which the tree is iteratively modified, using an efficient binary search method, to reduce its length. Experimental results show that this procedure works particularly well for technologies where the interconnect resistance dominates, and significant cost savings are generated.

1 Introduction

In recent years, interconnect delay has become an increasingly critical factor in VLSI systems. The increased effect of interconnect resistance has played a significant role as feature sizes have entered the deep submicron range and will become more dramatic in the future. To deal with this trend, many methods have been proposed to reduce interconnect delay, among which performance-driven routing has been an active area of research. Typically, the objective of a performance-driven routing algorithm has been to minimize the source-to-sink delay using various models of abstraction. Several types of problems have been tackled by researchers in the past, including minimizing the average source-to-sink delay over all sinks, minimizing the maximum source-to-sink delay, and minimizing a weighted sum of the delay from the source to some critical sink(s).

In this paper, we develop a performance-driven routing formulation whose objective is to meet a spec-

ified delay constraint at each sink. The approach can also be generalized to solve the problem of minimizing the maximum delay over all sink nodes. It is shown here that the use of Steiner points off the Hanan grid [5] can reduce the cost of the Steiner tree.

1.1 Previous work

Early work solved the problem of building performance-driven routing topologies by solving the minimum-length Steiner tree routing problem. This was based on the observation that in the domain where interconnect behaves purely as a capacitance, minimizing the tree length minimizes the source-sink delay. After Hwang's work [9] that proved that the ratio of the cost of a rectilinear minimum spanning tree (MST) to that of a minimum cost (length) rectilinear Steiner tree (RST) is within a factor of $3/2$, many heuristic algorithms [6, 7, 8, 10, 13] used the MST as a starting point to derive the RST. Kahng *et al.* [11] developed the iterated 1-Steiner heuristic method, which introduces only one Steiner point into a net at one time such that the current minimum spanning tree cost is minimized. A good overview of these techniques is provided in [12].

However, in the deep submicron range, the interconnect capacitance and resistance have become comparable to or dominate the gate capacitance and the output driver resistance and cannot be ignored during delay calculation any more. Therefore, it is apparent that for leading-edge technologies, minimum net lengths do not always yield minimum delay. Thus performance-driven methods are introduced to minimize the source-to-sink delay. The A-tree algorithm [4] and SERT [3] use the Elmore delay model, with a justification of its fidelity in [1] being provided as a basis for doing so. SERT is based on a greedy algorithm that optimizes the Elmore delay directly as the routing tree is constructed. Significant improvements over existing performance-driven routing tree constructions were demonstrated in this work. However, the area overhead of SERT is discouragingly large and for some

technologies, it tends to generate star-like topologies due to its greedy nature. Lillis [14] proposed a method that is a departure from the constructive greedy heuristics. The method builds routing topologies induced by sink permutations and then maps the topologies to a routing layout. The P – Tree_{AT} algorithm in this paper derives an area/delay tradeoff under the Elmore delay model and incorporates simultaneous wire-sizing by using dynamic programming.

1.2 Motivation

All of the above techniques have a common bottom-line: they are based on Hanan’s theory [5], which showed that if one were to draw horizontal and vertical grid lines through the source and sinks of the given signal net, there would be an minimum-length rectilinear Steiner tree whose Steiner points are all chosen from among the intersection points in the resulting grid. Thus the possible Steiner points can be chosen from a finite set, namely, the so-called Hanan grid. In [3], it was proved that only points on the Hanan grid need be considered while solving the problem of minimizing the weighted sum of critical sink delays. For the minmax problem of minimizing the maximum sink delay, it was shown in [2] that it is possible to build a better solution by considering points off the Hanan grid, but it was stated that such situations are uncommon and can be ignored. In this work we show that it is possible in cases to arrive at significantly better solutions by considering non-Hanan points during Steiner tree construction for two problems:

- (a) the minmax problem, and
- (b) the problem of achieving a specified delay at each sink node.

Example 1: We illustrate a simple example showing that a non-Hanan node is required to minimize the maximum source-sink delay during tree construction. We caution the reader not to be unduly swayed by the modest performance improvements in this simple example as our experimental results will show that it is possible to achieve more significant improvements on other larger examples.

Consider a net with a source at (0,0) and two sinks, a and b, at (3,0) and (1,4), respectively. We assume, for simplicity, a unit resistance and a unit capacitance per unit length. The driver has a source resistance of 6, and the sinks a and b have load capacitances of 1 unit and 4.5 units, respectively. The delays are calculated here under the Elmore delay model, described in Section 2.1. The variation of the delay at each sink as the Steiner point x is moved from (0,0) to (1,0) is

shown in Figure 1¹. The maximum sink delay for the tree is minimized at $x = 0.33$.

This example illustrates that in real design problems, the timing requirements at different sinks are often contradictory and it is necessary to arrive at a solution where all of the sinks are considered together.

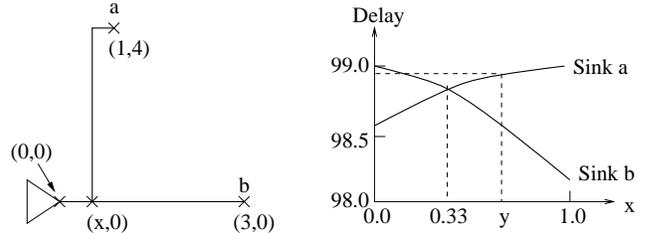


Figure 1: Illustrating the effects of non-Hanan Steiner points

For the problem of achieving a specified timing constraint T_{spec} , it is also easy to show that the optimal solution may lie at a non-Hanan point. Any procedure that restricts of Steiner points to Hanan points alone would lead to a larger than optimal tree cost. Therefore, for the problems of achieving a set of specified sink delays, and of minimizing the maximum source-sink delay, the best Steiner points do not necessarily lie on the Hanan grid.

However, the routing problem is clearly more difficult when the number of Steiner points becomes infinite, and the search space becomes extremely large. This paper utilizes the properties of the delay function to arrive at a simple and efficient method to overcome this challenge.

2 Preliminaries

2.1 Delay model

The delay in this work is modeled using Elmore delays [15], which are briefly described here. Given a routing tree $T(N)$ rooted at the source n_0 , let e_v denote the edge from node v to its parent in $T(N)$. The resistance and capacitance of edge e_v are denoted by r_{e_v} and c_{e_v} , respectively. Let T_v denote the subtree of T rooted at v , and $C_{d,v}$ denote the downstream tree capacitance of T_v , which is the sum of sink and edge capacitances in T_v . The Elmore delay from the predecessor of node v along edge e_v is equal to $r_{e_v}(c_{e_v}/2 + C_{d,v})$. This procedure can be used to recursively compute the delays. Given that the root node is driven by a driver of resistance r_d , the Elmore

¹The logic in [1] can be extended to show that a Steiner point to the right of (1,0) is suboptimal.

delay $t_{ED(n_i)}$ at sink n_i is calculated as:

$$t_{ED(n_i)} = r_d C_{no} + r_{ev} \sum_{\text{all fanouts}} \left(\frac{c_{ev}}{2} + C_{d,v} \right) \quad (1)$$

2.2 The SERT method

In this section, we provide a brief overview of the SERT (Steiner Elmore Routing Tree) algorithm in [3]. Before proceeding further, we define the following terms:

Definition [3]: A segment of a tree T is defined as a contiguous set of straight edges in T that are either all horizontal or all vertical. A maximal segment is a segment not properly contained in any other segment. The root of a segment is the extremal node of the segment that is closer to the root n_0 of the tree.

The essential idea of the SERT method is based on building a greedy Steiner tree using an approach similar to Prim's algorithm. Starting with a trivial tree consisting of only the source n_0 , the tree is iteratively built by adding a pin n_i in the tree and a sink n_j outside the tree so that adding edge (n_i, n_j) yields a tree with the minimum Elmore delay. The iterations continue until all sinks have been included in the tree. The algorithm is predicated on two results:

- (a) Selecting n_i to be downstream of the closest connection (CC) from n_j to the partial tree will yield a larger delay than connecting it to the CC .
- (b) For a maximal segment, if x is the distance from its root m_0 to the point n_i in the partial tree, then it was shown that the Elmore delay to any sink is a concave function of x when the connection point n_i lies between m_0 and CC , both inclusive. Therefore, any weighted sum of the Elmore delays to a fixed set of critical sinks is a concave function of x and is minimized by setting n_i to be either the root m_0 or CC .

These observations reduce the search space considerably as m_0 and CC are the only candidate connection points for a segment in each iteration. As both m_0 and CC lie on the Hanan grid, all Steiner points in the SERT tree must necessarily lie on the Hanan grid.

Note, however, that the concavity logic does not extend to the use of minmax formulations or formulations where the delay at each sink is a constraint. This is due to the fact that while the positive weighted sum of concave functions is concave, the maximum of concave functions is not concave, as seen from Figure 1.

3 The MVERT algorithm

3.1 Problem formulation

The procedure described in this work is referred to as the Maximum delay Violation Elmore Routing Tree (MVERT) algorithm. A basic concept that is used in this work is the idea of a delay violation, which is defined as the amount by which the delay $d(n_i)$ at sink i violates its specification, i.e.,

$$Violation(i) = d(n_i) - T_{spec}(i) \quad (2)$$

Clearly, a positive value of the violation implies that the constraints could not be met. A large negative value of the violation, on the other hand, indicates the possibility of overdesign, and it is possible in some cases to reduce the cost of the Steiner tree by bringing the violation value to be closer to zero. This idea motivates the formal statement of the MVERT problem as follows:

MVERT Problem: Given a signal net $N = \{n_0, n_1, \dots, n_k\}$ with source n_0 , construct a Steiner routing tree $T(N)$ such that the total length of the net is minimized while the delay violation at each sink node is less than 0, i.e.,

$$\begin{aligned} & \text{minimize} && \sum_{\text{all segments } k} w_k l_k && (3) \\ & \text{subject to} && d(n_i) \leq T_{spec}(i) \text{ for } i = 1, 2, \dots, n. \end{aligned}$$

3.2 Properties of the formulation

The MVERT problem formulation is quite general in nature. If the value of T_{spec} at each node is specified to be zero, then the problem is identical to the maximum source-sink delay problem. The user may specify different delay specifications at different sinks as desired.

Theorem 1: Consider any Steiner tree T connecting a source n_0 to sinks n_1, n_2, \dots, n_m . Let the node n_j be connected to the tree at a point n_k along a maximal segment s of the tree, where n_k is possibly a Steiner point. Let T' be the subtree rooted at n_j , let CC be a closest connection connecting n_j to the tree $T \setminus T'$ along maximal segment s' (with s' being possibly the same as s), and let m_0 be the root of s' . Let T_1 be the tree formed by connecting n_j to $T \setminus T'$ at CC . Then

- (a) The tree obtained by connecting n_j to $T \setminus T'$ at a point downstream of CC on segment s' results in a tree with a larger length and larger delays at each sink than T_1 .
- (b) If x is the distance from m_0 to the point n_i in the partial tree, then the Elmore delay to any sink is a concave function of x when the connection point n_i lies between m_0 and CC , both inclusive.

Proof: Parallel to that of a similar result in [3].

3.3 Tree construction procedure

As mentioned earlier, the set of candidate Steiner points is infinite, and it is necessary to find an efficient method to find the best Steiner points. The MVERT algorithm is divided into two phases:

- (a) The initial tree construction phase, where an initial tree is heuristically built to minimize delay.
- (b) The cost-improvement phase, where the tree is iteratively refined to reduce its cost while ensuring that it meets all timing specifications.

3.3.1 Phase 1: Initial tree construction

The first step in tree construction is similar to the ERT construction procedure in [3], with the difference that we minimize the maximum delay *violation* rather than the maximum delay. This procedure considers only candidate points on the Hanan grid.

3.3.2 Phase 2: Cost improvement

The initial tree constructed above attempts to minimize the maximum delay violation at any leaf node. Since the criticality of a sink is dependent on the timing constraints and the positions of the other sinks, this provides a natural technique for automatically detecting critical sinks and providing more importance to them in the tree construction process.

However, the tree so constructed may be conservative as its objective is to minimize the violation, rather than to simply ensure that the constraints are never violated². Moreover, the Phase 1 procedure considers only Hanan grid points as candidate Steiner points. Therefore, it attempts to connect each point either to the closest connection (*CC*) in the partial tree, or to the root node m_0 of a maximal segment. If the delay violation associated with a *CC* connection were larger than the delay associated with a connection to m_0 , then the algorithm would make a connection to m_0 . However, due to the interactions between paths the MVERT solution may lie at a different (and possibly non-Hanan) point, and the connection to m_0 results in a larger net length than is necessary. Therefore, we examine the tree constructed in Phase 1 and move node connections from the root of a segment towards *CC* in a bid to reduce the tree length, as shown in Figure 2, while ensuring that all timing constraints are satisfied.

Thus in Phase 2, we refine the tree built in Phase 1 to reduce its length while ensuring that the timing

²The idea is that the minimum-length Steiner tree minimizes tree length and the minimum-delay Steiner tree minimizes the delay; the middle ground between these extremes would solve the MVERT problem.

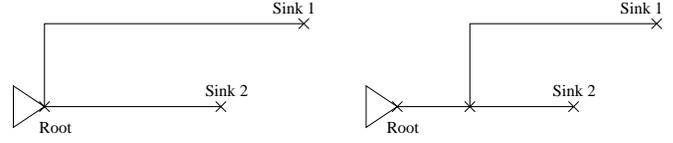


Figure 2: Creating a non-Hanan Steiner point that meets timing specifications

constraints are satisfied. The pseudocode for this procedure is shown below:

```

Input: Routing tree  $T_1 = (V, E)$  from Phase 1
Output: An optimal routing tree  $T_2$ 
if ( $T_1$  did not satisfy constraints) OR
    (all sinks were connected to CC in  $T_1$ )
     $T_2 = T_1$ ; exit
Sort sinks  $v_1, v_2, v_3 \dots$  that are not
    connected to a CC in  $T_1$  in descending
    order of the distance to  $n_0$ 
for (each such sink  $v_i$ ) do Improve_Cost( $v_i$ )
Output resulting routing tree  $T_2 = (V, E)$ 

```

If the original tree T_1 did not satisfy the given timing constraints, then moving the Steiner points towards the *CC* for a sink node will not help meet the constraints. Also, if T_1 only has *CC* connections, then no further improvements are possible in this phase. In either case, the output of this phase will be the tree T_1 . However, if neither of the two conditions hold, then the Improve_Cost routine is invoked to reduce the cost of the tree T_1 as follows:

```

Function Improve_Cost( $v_i$ )
Find path  $p$  from  $n_0$  to  $v_i$   $Q = \{u_1, u_2, \dots, u_k\}$ ,
    where  $u_i$  is a sink connected to a node
    on the path between  $n_0$  and  $v_i$ .
For each ( $u_i$  connected to  $m_0 \neq CC$ ) do
Remove connection of  $u_i$  to  $m_0$ , while keeping
    the connection of  $u_i$  to its
    downstream nodes.
Define  $T' = T \setminus \{(e_i)(T_{u_i})\}$ , where  $e_i$ 
    is the edge that connects  $u_i$  to  $n_0$ ,
     $T_{u_i}$  is the downstream tree of  $u_i$  in  $T$ .
Find CC for  $u_i$  in  $T'$ 
Reconnect  $e_i$  to  $T'$  at the nearest point to
    CC where constraints are satisfied.

```

The idea is illustrated in Example 1 for the constraint of 98.8 units, where we see that a connection to $(y, 0)$ is preferable to a connection to $(0.33, 0)$.

3.3.3 Finding the optimal reconnection point

Consider the set of constraints on the routing tree from Equation (3). Rewriting them in the form $d(n_i) - T_{spec}(i) \leq 0$ for all sinks i , we see that the maximum violation must always be nonpositive. Since each of the $d(n_i)$'s is a concave function of the connection point x by Theorem 1, and since any concave function shifted by a constant is a concave function, this implies that we must find a reconnection point x such that the maximum of the set of concave functions is nonpositive. This is pictorially shown in Figure 3 below for a net with four sinks, v , w , y , and z ; the maximum violation function is shown by the darkened line. Note that the graph shows that sink z is never critical in this case, for any value of x . The delay violation at each sink as a function of x is a concave function and the objective is to find the value of x that is closest to CC (corresponding to a minimal increase in the net length) that satisfies all constraints. In Figure 3, this point is found to be x^* . This point would, in general, be a non-Hanan point.

Corollary 2: The search space for the reconnection point in the pseudocode in Section 3.3.2 can be restricted to the interval between n_0 and a closest connection point, CC .

Proof: This follows directly from Theorem 1.

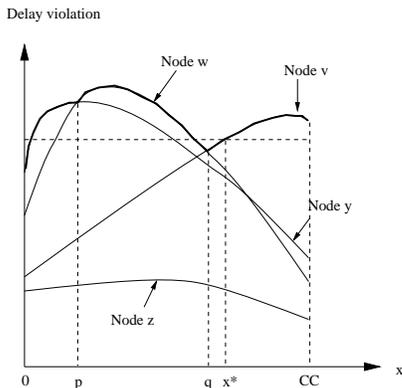


Figure 3: Finding the optimal value of x that satisfies the timing constraints

In searching for x^* , we observe that it is possible to perform a binary search on the value of x from 0 to CC , while taking advantage of the fact that the value on each concave segment is minimized at its intersection with the concave segment on either side (if such a segment exists), or at 0 or CC otherwise. In Figure 3, this translates to the fact that for the minmax problem, the only candidate solutions are 0, p , q and CC . This permits a reduction of the search space from the

Table 1: Results on technology 1 (IC)

| Circuit | Cost | | |
|-----------|---------------|---------------|------------------------|
| | After Phase 1 | After Phase 2 | Percentage Improvement |
| Net 1_IC5 | 883 | 727 | 21.4% |
| Net 2_IC5 | 1049 | 954 | 10.0% |
| Net 3_IC5 | 820 | 758 | 8.2% |
| Net 4_IC5 | 936 | 813 | 15.1% |
| Net 5_IC5 | 645 | 563 | 14.6% |
| Net 6_IC5 | 745 | 525 | 41.9% |

infinity of points between 0 and CC .

For the problem of meeting timing specifications at each sink, several pruning strategies are possible for the binary search, and just one is described here due to space constraints. If the value of $\max[d(n_i) - T_{spec}(i)]$ at each end of the concave segment is positive, then the search would find the end point of the concave segment away from CC and repeat this procedure on the next concave segment since each point on the concave segment must necessarily have a positive violation.

4 Experimental results

The MVERT algorithm was applied to nets in two technologies: an IC technology and an MCM technology. Design parameters for each technology are taken from [3]. The results are shown in Tables 1 and 2, respectively. The CPU time required for the computations was insignificant (under a second).

It is seen that in each case, for the nets shown and the timing constraints chosen, significant reductions in the cost function are permitted by this procedure. Overall, it is seen that better results are achievable for technology 2 than for technology 1. The explanation for this stems from the fact that the driver resistance is lower in technology 2, due to which the resistive effects of interconnect are more pronounced. Under such a situation, a SERT-like algorithm (or Phase 1 of our procedure) is more likely to generate star-like connections that connect more nodes to the root node. Since many of these connections may, under certain timing constraints, be overkill, in Phase 2 these connections are moved to be closer to CC . This has the added benefit of reducing the routing congestion near the root node.

5 Conclusion

A new technique for finding a minimum cost Steiner tree subject to timing specifications at the sink nodes

Table 2: Results on technology 2 (MCM)

| Circuit | Cost | | |
|-------------|---------------|---------------|------------------------|
| | After Phase 1 | After Phase 2 | Percentage Improvement |
| Net 1_MCM5 | 670 | 633 | 5.8% |
| Net 2_MCM5 | 659 | 600 | 9.8% |
| Net 3_MCM5 | 733 | 649 | 12.9% |
| Net 4_MCM5 | 734 | 425 | 72.7% |
| Net 5_MCM5 | 696 | 568 | 22.5% |
| Net 6_MCM5 | 760 | 726 | 4.7% |
| Net 7_MCM5 | 792 | 654 | 21.1% |
| Net 8_MCM5 | 812 | 676 | 20.1% |
| Net 9_MCM5 | 811 | 763 | 6.3% |
| Net 10_MCM5 | 669 | 605 | 10.6% |
| Net 11_MCM5 | 878 | 595 | 47.6% |
| Net 12_MCM5 | 819 | 731 | 12.0% |
| Net 13_MCM5 | 898 | 693 | 29.6% |
| Net 14_MCM5 | 619 | 513 | 20.7% |
| Net 15_MCM5 | 760 | 666 | 14.1% |
| Net 16_MCM5 | 794 | 685 | 15.9% |

is presented. It is shown that the use of non-Hanan Steiner nodes can provide noticeable benefits in improving the cost of the tree. If the timing specifications at the sinks are very loose, a minimum length Steiner tree will be the correct solution. If the timing specifications are very tight, a minimum-delay Steiner tree is desirable. It is in between these two extremes that the MVERT proposed here paper is useful. In real problems such situations are likely to occur often when routing trees have to be built to conform to timing budgets. This work also shows utility of considering non-Hanan points as candidate Steiner points.

Acknowledgement: The authors thank Jiang Hu for his useful comments.

References

- [1] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Fidelity and near-optimality of Elmore-based routing constructions," *Proc. IEEE Int. Conf. Computer Design*, pp. 81-84, 1993.
- [2] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Rectilinear Steiner trees with minimum Elmore delay," *Proc. ACM/IEEE Design Automat. Conf.*, pp. 381-386, 1994.
- [3] K. Boese, A. Kahng, B. McCoy, G. Robins, "Near-optimal critical sink routing tree constructions," *IEEE Trans. Computer-Aided Design*, Vol. 14, pp. 1417-1436, December 1995.
- [4] J. Cong, K. S. Leung and D. Zhou, "Performance-driven interconnect design based on distributed RC delay model," *Proc. ACM/IEEE Design Automat. Conf.*, pp. 606-611, 1993.
- [5] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Appl. Math.*, Vol.14, pp. 255-265, 1966.
- [6] N. Hasan, G. Vijayan, and C. K. Wong, "A neighborhood improvement algorithm for rectilinear Steiner trees," *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 2869-2872, 1990.
- [7] J. Ho, G. Vijayan and C. K. Wong., "Constructing the optimal rectilinear Steiner tree derivable from a minimum spanning tree," *Proc. IEEE Int. Conf. Computer Design*, pp. 6-9, 1989.
- [8] J. Ho, G. Vijayan, and C. K. Wong, "New algorithms for the rectilinear Steiner tree problem," *IEEE Trans. Computer-Aided Design*, Vol. 9, pp. 185-193, Feb.1990.
- [9] F. K. Hwang, "On Steiner minimal trees with rectilinear distance," *SIAM J. Applied Math.*, Vol.30, pp. 104-114, Jan.1976.
- [10] F. K. Hwang, "An $O(n \log n)$ algorithm for suboptimal rectilinear Steiner tree," *IEEE Trans. Circuits Syst.*, Vol.CAS-26, pp. 75-77, Jan. 1979.
- [11] A. Kahng and G. Robins, "A new class of iterative Steiner tree heuristics with good performance," *IEEE Trans. Computer-Aided Design*, Vol.11, pp. 893-902, July 1992.
- [12] A. Kahng and G. Robins, *On Optimal Interconnections for VLSI*, Kluwer Academic Publishers, Norwell, MA, 1995.
- [13] J. H. Lee, N. K. Bose, and F. K. Hwang, "Use of Steiner's problem in suboptimal routing in rectilinear metric," *IEEE Trans. Circuits Syst.*, Vol. CAS-23, pp. 470-476, July 1976.
- [14] J. Lillis, C.-K. Cheng, T.-T. Y. Lin and C.-Y. Ho, "New performance-driven routing techniques with explicit area/delay tradeoffs and simultaneous wire sizing," *Proc. ACM/IEEE Design Automat. Conf.*, pp. 395-400, 1996.
- [15] J. Rubinstein, P. Penfield and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, pp. 202-211, Vol. CAD-2, No. 3, July 1983.