Further Comments on "An Optimal Solution for the Channel-Assignment Problem"

FRANK RUBIN

In the above paper[1] Gupta, Lee, and Leung present a new algorithm for making an optimal assignment of wire segments to channels on a PC board. The new algorithm has a running time of $0(N \log N)$ in the general case, but can be executed in $0(N)$ time if the wire segment endpoints are close to uniformly distributed.

An earlier algorithm due to Hashimoto and Stevens appeared in [1]. The execution time for their algorithm cannot be directly stated, since they do not specify the method for organizing and searching the table of wire segments. The most naive algorithm would leave the wire segments in input order and make a serial search for each wire segment to be assigned. This implementation would have $0(N^2)$ execution time. It will be shown here that the Hashimoto and Stevens algorithm can be implemented with $0(N)$ execution time if the left endpoints of the wire segments have near uniform distribution.

At each stage of the Hashimoto and Stevens algorithm, the next segment to be assigned to the current channel is sought. The search is for the leftmost segment lying to the right of the previously assigned segment. The searching occupies most of the time in this algorithm.

Let the number of wire segments be N, and let C be a convenient constant. Divide the length of the channel into N/C intervals or bins of size C. The wire segments can be assigned to these N/C intervals in $0(N)$ operations. It is not necessary to sort the wire segments within the bins.

Assume that the bin size is approximately $3C/4$ when the ith wire is being assigned, and that the ith wire is to be assigned to the right of wire segment (x, y). The assignment will involve a serial search of the bin that would contain the value y. There is about a chance of $4/3C$ that all of the segments in this bin will lie to the left of y, hence requiring a search of the next bin as well. Thus, the mean search time will be $1 + 3C/4$ per wire segment, or $0(N)$ for all wire segments.

The above estimate would fail to hold if many bins became empty, thus requiring skipping over the empty bins during the search. To prevent this, the size of the bins can be doubled when half of the wire segments have been assigned. The ith doubling procedure would involve changing $N/2^i$ pointers, for a total of N pointer changes. The average bin size would then be maintained at $3C/4$.

All operations involved in the above implementation of the Hashimoto and Stevens algorithm are $0(N)$. Consequently, there is no difference in the computational order of the new algorithm and the older agorithm [1].

References

[1] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Des. Automation Workshop*, June 1971, pp. 155–169.

Manuscript received January 16, 1980; revised February 8, 1980.

The author is with IBM, Poughkeepsie, NY 12602.

[1] U. I. Gupta, D. T. Lee, and J.Y-T. Leung, *IEEE Trans. Comput.*, vol. C-28, pp. 807–810, Nov. 1979.

Additional Comments on "An Optimal Solution for the Channel-Assignment Problem"

U. LAUTHER

The problem stated and solved in the above paper[1] has been solved before with essentially the same algorithm by Hashimoto and Stevens.

The problem discussed by Gupta, Lee, and Leung–how to partition a set of intervals (x_i, y_i) into a minimal number of subsets (channels) such that no two intervals of one subset overlap each other–has already been solved by Hashimoto and Stevens [1] as early as 1971. They used essentially the same method as Gupta *et al.* The main difference to the new algorithm is that they filled the channels not simultaneously, but sequentially: Starting with the leftmost interval always the closest fitting interval is selected for the next assignment until no further interval fits in the current channel; then the next channel is started. Implementation details were not discussed in the paper.

In [2] this method was called "left-edge algorithm" and a time complexity $0(N \log N)$ was stated without proof.

This proof can easily be given: N times we have to search for the interval i with smallest left value x_i and with $x_i \geq y_j$ (j being the last assigned interval in the current channel) and then to delete interval i from the list of unassigned intervals. This search for the closest fitting interval takes $0(\log N)$ steps when an appropriate list structure is used which takes $0(N \log N)$ steps to initialize.

References

[1] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Des. Automation Workshop*, June 1971, pp. 155–169.

[2] B. W. Kernighan, D. G. Schweikert, and G. Persky, "An optimum channel-routing algorithm for polycell layouts of integrated circuits," in *Proc. 10th Des. Automation Workshop*, 1973, pp. 50–59.

Authors' Reply[2]

U. I. GUPTA, D. T. LEE, AND J. Y-T. LEUNG

Since the publication of this paper it has been brought to our notice by Lauther, Persky *et al.*, and Rubin that the channel assignment problem has been studied by Hashimoto and Stevens [2] and Kernighan *et al.* [3]. These papers should have been cited and we regret the omission.

In [2] an algorithm is presented which works channel by channel, making several passes from left to right. Although no mention of complexity is made, a straightforward implementation would take $0(n^2)$ time in the worst case. In [3] this method is referred to as the "left edge algorithm" and it is stated without proof that an $0(n \log n)$ implementation is possible. It is indeed possible to implement the algorithm to run in $0(n \log n)$ time using balanced tree schemes.

We disagree with Persky *et al.* that our algorithm is a "detailed restatement" of the left-edge algorithm. Unlike the left-edge algorithm, our algorithm makes a single pass over the input and takes only $0(n)$ time after the initial sorting step. In the left-edge algorithm, after the initial balanced tree has been constructed in $0(n \log n)$ time it will take another $0(n \log n)$ time to find the closest fitting intervals for the n intervals.

We believe that the overhead involved in the use of balanced tree structure is more than in our algorithm. Also significant is the fact that after the initial sorting, our algorithm requires only $0(n)$ time. Thus, in the special case where the endpoints are drawn from a set of cardinality p, our algorithm runs in $0(n + p)$ time. We do not see how the Hashimoto & Stevens algorithm can be adapted for this situation to run in $0(n + p)$ time or less.

Manuscript received March 4, 1981.

The author is with Siemens AG, Munich, West Germany.

[1] U. I. Gupta, D. T. Lee, and JY-T. Leung, *IEEE Trans. Comput.*, vol. C-28, pp. 807–810, Nov. 1979.

[2] Manuscript received March 4, 1981.

The authors are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201.

0018-9340/81/0600-0455$00.75