Single-Row Routing in Narrow Streets

SANYONG HAN AND SARTAJ SAHNI, MEMBER, IEEE

***Abstract*—We develop fast linear time algorithms for single row routing when the upper and lower street capacities are less than or equal to three. A similarly fast algorithm is developed for the case when one of the streets has a capacity 1 and the other has an arbitrary capacity. Experimental results show that our algorithms are many times faster than previously developed algorithms.**

I. Introduction

So has proposed a systematic approach to the interconnection problem of large multilayer printed circuit boards in which pins and feedthroughs are uniformly spaced on a rectangular grid [6]. This approach consists of a systematic decomposition of the general multilayer wiring problem into a number of independent single layer, single-row routing problems. There are five phases in this decomposition [8], [5]:

1. Via assignment,
2. Linear placement of via columns,
3. Layering,
4. Single row routing,
5. Via elimination.

In this paper, we are concerned only with the fourth phase: Single row routing. In the single-row routing problem, we are given a set $V = \{1, 2, \cdots, n\}$ of n nodes that are evenly spaced along a straight line; and a set $L = \{N_1, N_2, \cdots, N_m\}$ of nets. Each net represents a set of nodes that are to be made electrically equivalent. The nets satisfy the following conditions:

(i) $N_i \cap N_j = \phi, \quad i \neq j$

(ii) $\bigcup_{i=1}^{m} N_i = \{1, 2, \cdots, n\}$

$j \epsilon N_i$ is a *touch point* of net i. The nets are to be realized in a single layer by the use of nonoverlapping wires that are composed solely of horizontal and vertical segments. Fig. 1(a) shows some of the legal ways to realize the net $\{1, 3; 6, 9\}$. The wire layout must satisfy the additional requirement that a vertical cut made at any point along the axis formed by the nodes can intersect at most one horizontal segment from each net. Thus the wiring of Fig. 1(d) is illegal.

The area above the line of nodes is called the *upper street* while that below this line is the *lower street*. Each street has *tracks* that run parallel to the line of nodes (Fig. 2). Horizontal wire segments must be layed in tracks and no track may hold more than one wire segment at any point (of course, several non overlapping wire segments may be packed into the same track). Let K_u and K_l, repectively, denote the number of tracks in the upper and lower streets. Fig. 2 shows one way to realize the nets $(N_1, N_2, \cdots, N_5) = (\{1, 7\}, \{2, 8\}, \{3, 6\}, \{4, 9\}, \{5, 10\})$ when $K_u = 2$ and $K_l = 3$.

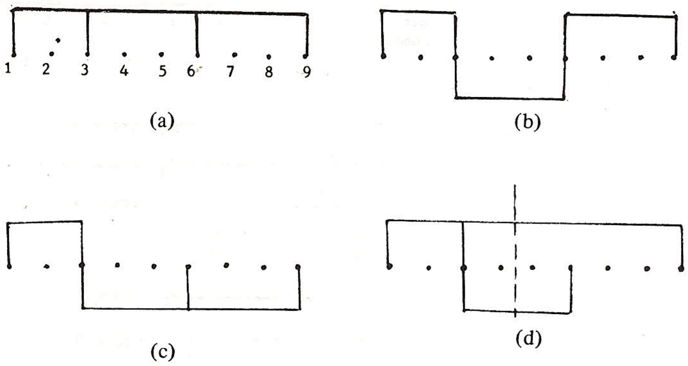

Fig. 1.

In this paper, we are specifically concerned with obtaining net realizations (when they exist) given V (node set), L (net set), K_u (upper street capacity), and K_l (lower street capacity).

This problem has been studied before. Kuh *et al.* [3] and Tsukiyama *et al.* [9] developed necessary and sufficient conditions for a net set to be realizable. In [3], a simple construction is used to show that when K_u and K_l are sufficiently large, L is realizable. This construction, however, results in an algorithm of complexity $O(m!n)$ where $|V| = n$ and $|L| = m$. Raghavan and Sahni [4] have developed an algorithm of complexity $O(k! * k * n * \log k)$ where $k = K_u + K_l$ for this problem. This algorithm is quite practical when k is small but impractical when k is large. For the case $K_u \leqslant 2$ and $K_l \leqslant 2$, Tsukiyama *et al.* [9] have developed an $O(mn)$ algorithm. This algorithm is, however, slower than that of [4].

Here, we shall develop fast $O(n)$ algorithms for the case $K_u \leqslant 3$ and $K_l \leqslant 3$ as well as the case $K_l = 1$ and K_u arbitrary. Experimental results show these algorithms to be several times faster than that of [4] for these street capacities. The case K_u and $K_l \leqslant 3$ is actually solved by developing algorithms for the subcases $K_u = K_l = 2$; $K_u = K_l = 3$; and $K_u = 3$, $K_l = 2$. These algorithms together with that for $K_l = 1$ and K_u arbitrary cover all the possibilities for $K_u \leqslant 3$ and $K_l \leqslant 3$. Note that the case of $K_l = 0$ and K_u arbitrary is trivially solved in linear time.

Before presenting the algorithms, we introduce some notation. Following [4], nodes are classified by type:

(a) Node i is of *type B* if it is the left extreme node of a net,

Manuscript received April 22, 1983. This work was supported in part by the National Science Foundation under Grant MCS 80-005856.

The authors are with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455.

0028-0070/84/0700-0235$01.00 © 1984 IEEE