Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 34, No. 1, February 1987
Printed in Belgium

River Routing in VLSI

ANDRANIK MIRZAIAN

*Department of Computer Science, York University,
North York, Ontario M3J 1P3, Canada*

Received August 13, 1984; revised July 22, 1986

A common framework for solving several VLSI river routing problems is developed. The main result of this paper is an $O(n)$ time algorithm for the *optimum offset* problem. This improves upon the best previously known $O(n \log n)$ time bound. A new reduction technique called *halving* is used to achieve this result. A variety of other applications of the *halving* technique are also mentioned. Algorithms for the *minimum area, minimum longest wire length*, and *minimum total wire length* problems are also given that take $O(n^2)$ time. © 1987 Academic Press, Inc.

1. INTRODUCTION

River Routing is a VLSI layout problem that has received considerable attention recently. See, in particular, [4, 11, 15, 16]. The main result of this paper is an $O(n)$ time algorithm for the *optimum offset* problem. This improves the best previously known time bound of $O(n \log n)$ [4, 15]. Furthermore, the constant time factor in our $O(n)$ algorithm is much smaller than those known previously. Algorithms for the *minimum area, minimum longest wire length*, and *minimum total wire length* problems are also given that take $O(n^2)$ time.

Our VLSI model comprises two horizontal rows, each containing *n terminal* points. The bottom row contains terminals $a_0, a_1, \ldots, a_{n-1}$, and the top row contains terminals $b_0, b_1, \ldots, b_{n-1}$. Wire W_i is to connect terminals a_i and b_i, for $0 \leqslant i < n$. The wires should satisfy certain constraints defined by the *wiring model.* In this paper we consider the *single-layer-rectilinear* wiring model. However, as we shall see in the next sections, the methods developed in this paper may also be applied to some other wiring models, such as *multilayer, 45 degree lines*, etc. The area in which the wires are to be routed is called the *channel.* The channel consists of horizontal and vertical grid lines. The horizontal grid lines are called *tracks* and the vertical grid lines are called *columns*. Tracks are numbered 0, 1, 2,..., s from bottom to top. The topmost and bottommost tracks contain the terminals, $s \geqslant 0$ is called the channel *separation.* Wires should follow the grid lines and be separated from each other by at least one grid unit. The points a_i and b_i will represent the terminals as well as their horizontal positions (i.e., column numbers). We assume $s = 0$ is possible iff $a_i = b_i$ for all i. If $s > 0$, then we assume wires should be connected to track s

0022-0000/87 $3.00

vertically, while we allow them to be connected to track 0 either horizontally or vertically. (This is merely an assumption for convenience and is not a serious restriction.) Figure 1 shows a wiring example.

The terminals on a track are separated from each other by at least one grid unit. That is, $a_{i-1} \leqslant a_i - 1$ and $b_{i-1} \leqslant b_i - 1$, for $0 < i < n$. We let the track containing b_i's slide horizontally; the amount of horizontal displacement d is called the channel *offset*. That is, with offset d, terminal b_i moves to horizontal position $b_i + d$, for $0 \leqslant i < n$. We say a pair (s, d) is *feasible* if there is a feasible wiring with separation s and offset d. The following problems are considered.

(a) *The separation problem.* Given a fixed offset d; find the minimum separation s, such that (s, d) is feasible.

(b) *The offset range problem.* Given a fixed separation s; find the set of all offsets d, such that (s, d) is feasible.

(c) *The optimum offset problem.* Find a feasible pair (s, d) that minimizes s.

(d) *The feasible set problem.* Compute the set of all feasible pairs (s, d). This problem may be regarded as solving the offset range problem for each separation $s = 0, 1,..., n-1$. However, we give some indication that there might be a more effecient way of solving this problem. We will show how an efficient solution of this problem may be used to obtain solutions for many problems such as the following three.

(e) *The minimum area problem.* Find a feasible pair (s, d) for which the area of the smallest rectangle (with horizontal and vertical sides) circumscribing the entire wiring is minimized.

(f) *The minimum longest wire length problem.* Find a feasible pair (s, d) for which the length of the longest wire is minimized.

(g) *The minimum total wire length problem.* Find a feasible pair (s, d) for which the sum of the wire lengths is minimized.

Dolev *et al.* [4] give $O(n)$ time algorithms for the *separation* problem and an $O(n \log n)$ time algorithm for the *optimum offset* problem. They also claim to have

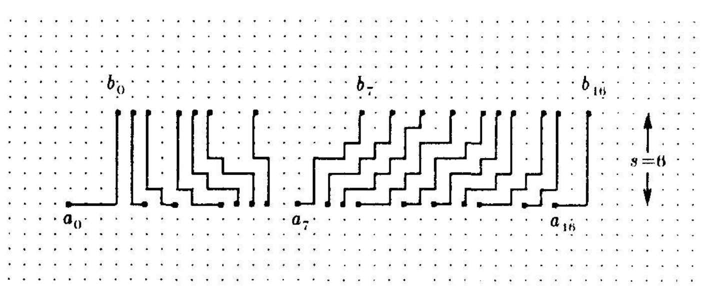

FIG. 1. A wiring example.

an $O(n^3)$ time algorithm for the *minimum area* problem. Siegel and Dolev [15] give $O(n)$ time algorithms for the *separation* and *offset range* problems and an $O(n \log n)$ time algorithm for the *optimum offset* problem. Leiserson and Pinter [11] report that the *minimum area* problem can be solved in $O(n^2)$ time, but do not give any algorithm.

Section 2 of this paper develops a methodology for approaching the solution of the problems mentioned above. The results in this section are the basis for the developments in subsequent sections. Consequently, we present $O(n)$ time algorithms for the *separation* and the *offset range* problems, although comparable results have already appeared elsewhere [4, 15].

Section 3 introduces the new reduction technique called *halving*, and mentions a variety of application areas which indicate the technique to be a good candidate as an algorithmic paradigm.

Section 4 uses the *halving* technique to obtain an $O(n)$ time algorithm for the *optimum offset* problem; thus improving the best previously known $O(n \log n)$ time bound. The constant time factor of our $O(n)$ algorithm is also smaller than those known previously. Consequently, this result is of both practical and theoretical interest.

Section 5 discusses the *feasible set* problem and shows how this problem may be regarded as a special case of the *discrete convolution* problem with the *min* and $-$ operators. We also show that an efficient solution of this problem implies efficient solutions of the *minimum area*, *minimum longest wire length*, and *minimum total wire length* problems.

Section 6 makes some concluding remarks.

2. The Methodology

By showing that the wires can take a special form, we obtain a simple necessary and sufficient condition for the feasibility of a pair (s, d), for separation s and offset d. This section adopts some of the ideas from [4, 16] and in particular [11].

As we said earlier, wire W_i connects terminals a_i and b_i. Wire W_i may be considered as being composed of two parts. The first part, which we call the *base wire*, extends from a_i and its shape depends only on the position of the terminals at track 0. The second part is a vertical wire segment (of possibly zero length) that extends downward from b_i and joins the base wire. There are two types of base wires, *left base wire*, denoted by L_i, and *right base wire*, denoted by R_i. Intuitively, the base wires L_i (R_i) are those that extend from a_i upward and as far to the left (right) as possible without violating the constraints of the wiring model. Figure 2 shows the base wires corresponding to the example of Fig. 1. Note how R_{i+1} may be obtained by an appropriate shift of R_i along the direction with slope -1. L_i may be obtained from L_{i+1} by a similar shift. In order to obtain wire W_i, we drop a vertical line from terminal b_i until it hits one of the base wires L_i or R_i (depending on whether terminal b_i is to the left or to the right of a_i). The vertical line plus the por-

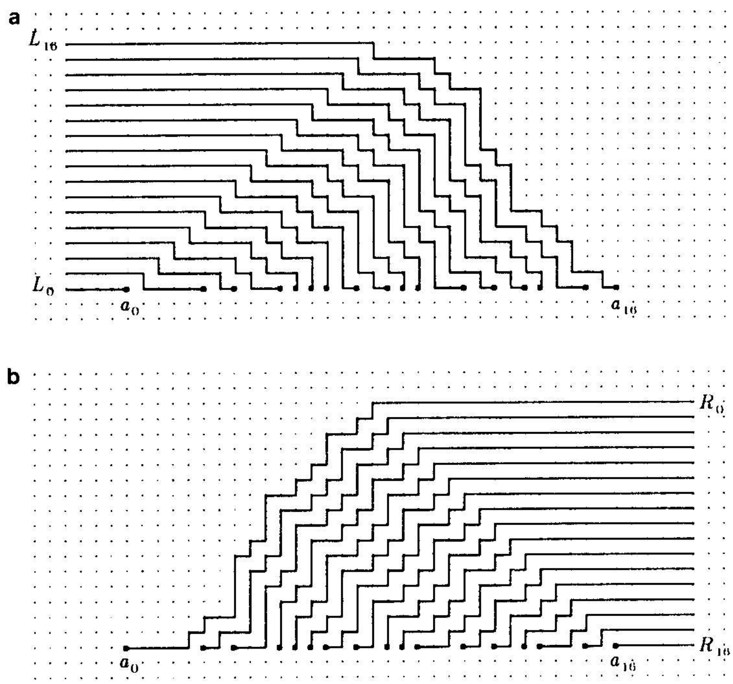

FIG. 2. (a) Left base wires. (b) Right base wires.

tion of the intersected base wire from the intersection point to a_i forms W_i, as illustrated in Fig. 3. It can be shown that this process will produce a feasible wiring whenever one such wiring exists (for a proof see, e.g., [11]).

From the above discussion we see that a necessary and sufficient condition for the feasibility of (s, d) is that terminal b_i, which is at track s and column $b_i + d$, must be above the base wires L_i and R_i, for $0 \leqslant i < n$. The condition is

$$a_{i-s} + s \leqslant b_i + d \leqslant a_{i+s} - s,$$

where the lhs inequality is for $s \leqslant i < n$ and the rhs inequality is for $0 \leqslant i < n - s$. These inequalities follow from the fact that L_{i-s} may be obtained by shifting L_i down along the direction with slope $+1$, and similarly, R_{i+s} may be obtained by shifting R_i down along the direction with slope -1 (see Fig. 3). This necessary and sufficient condition is essentially given in [11].

Let $x_i = a_i - i$ and $y_i = b_i - i$. We see that $x_0 \leqslant x_1 \leqslant \cdots \leqslant x_{n-1}$ and $y_0 \leqslant y_1 \leqslant \cdots \leqslant y_{n-1}$. Let $x = (x_0, x_1, ..., x_{n-1})$ and $y = (y_0, y_1, ..., y_{n-1})$. We conclude that the pair (s, d) is feasible iff $x_i - y_{i+s} \leqslant d \leqslant x_{i+s} - y_i$, for $0 \leqslant i < n - s$. Let

$$u(s, x, y) = \min\{x_{i+s} - y_i \mid 0 \leqslant i < n - s\} \tag{1}$$

$$l(s, x, y) = \max\{x_i - y_{i+s} \mid 0 \leqslant i < n - s\}. \tag{2}$$

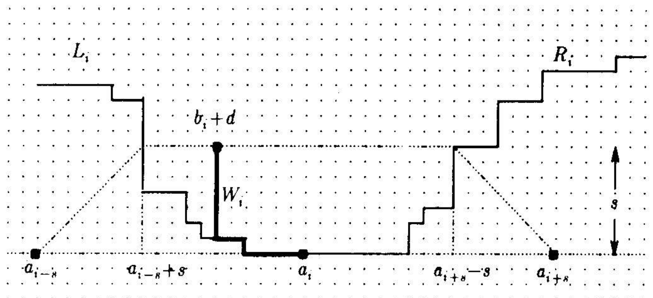

FIG. 3. Wires W_i, L_i, and R_i.

When x and y are understood, we use $l(s)$ and $u(s)$ instead of $l(s, x, y)$ and $u(s, x, y)$. We summarize the above discussion in Theorem 1, below.

THEOREM 1. *The pair (s, d) is feasible iff $l(s) \leqslant d \leqslant u(s)$.*

Theorem 1 is the basis of much of the subsequent development in this paper. An immediate consequence of Theorem 1 is the following theorem due to [15].

THEOREM 2. *The offset range problem can be solved in $O(n)$ time.*

Proof. All we need to do is to compute $l(s)$ and $u(s)$ using Eqs. (1) and (2). ∎

For most of what follows. It is helpful to imagine the $n \times n$ matrix $X - Y$, whose entry at row i and column j is $x_i - y_j$. In this matrix, the entries in a row are in descending order and entries in a column are in ascending order. By this convention, $u(s)$ may be described as being the minimum entry along the (sub)diagonal of $X - Y$, which is in the lower triangle of the matrix and a "distance" s away from the main diagonal. Similarly, $l(s)$ is the maximum entry along the (sub)diagonal, which is the symmetric image of the one described for $u(s)$. We emphasize that this matrix formulation is for exposition purposes; it is not used by our algorithms.

Another implication of Theorem 1 is a simple $O(n)$ time algorithm for the *separation problem* (for other solutions of this problem see [4, 15]). That is, given an offset d, find the minimum separation s such that $l(s) \leqslant d \leqslant u(s)$. More specifically, we have

THEOREM 3. *The function separation(n, d, x, y) below solves the separation problem in $O(n)$ time.*

Proof. The matrix formulation above allows us to easily establish the fact that at the end of the ith iteration of the loop, s is the minimum separation corresponding to the offset d and the vectors $(x_0, x_1, \ldots, x_i)$ and $(y_0, y_1, \ldots, y_i)$. ∎

```
function separation (n, d, x, y);
begin
  s := 0;
  for i := 0 to n − 1 do
    if not (x_{i−s} − y_i ≤ d ≤ x_i − y_{i−s}) then s := s + 1;
  return s
end.
```

The wiring process described earlier led us to Theorem 1. This process may be used even when other *single-layer* wiring models are used and a discrete number of tracks are available. Only the base wires L_i and R_i need to be redefined. For an l-layer wiring model (assume layers are numbered $0, 1, \ldots, l-1$), with some reasonable assumptions about the model, it can be shown (see [3]) that in order to achieve a wiring, we may lay out wire W_i entirely in layer $i \pmod{l}$. In this way the l-layer wiring of n terminal pairs is decomposed into l independent problems of single-layer wiring, each with at most $\lceil n/l \rceil$ terminal pairs. Thus, the wiring in each layer may proceed, independent of the other layers, in exactly the same way as we have described earlier in this section.

3. The Halving Technique

The *halving* technique may be described as follows. Suppose we are given a sequence $x = x_0 x_1 x_2 \cdots x_{n-1}$ and we need to compute an integer function $f(x)$. Let x^e, called the *even half of* x, be the subsequence of x comprising even indices. That is, $x^e = x_0 x_2 x_4 \cdots$. (We may similarly define x^o, called the *odd half of* x, to be the subsequence of x comprising odd indices.) We note that the length of each of the sequences x^e and x^o is about half the length of x. We recursively compute $f(x^e)$; then we use $2f(x^e)$ as an approximation to $f(x)$. A final adjustment will transform the approximate solution to the exact solution.

Some variations of the above general description are possible. For instance, instead of using $2f(x^e)$ as an approximation to $f(x)$, we may use some combination of $f(x^e)$ and $f(x^o)$. This line of approach can be seen in the *Fast Fourier Transform* [1]. One difference is that in the latter case we need two recursive calls instead of one. For efficiency reasons this should be avoided if possible. The *halving* technique can also be generalized from one-dimensional to multidimensional sequences, such as matrices.

The *halving* technique results in efficient algorithms for a variety of application areas. One such example is the *odd–even merge algorithm* [10, 17]. In [14] the *halving* technique is used to obtain an $O(n)$ time algorithm for the *selection* problem on $n \times n$ matrices with sorted rows and columns. (See also [6] for a solution of the latter problem.) In [13] a suitable variation of the *halving* technique is used to obtain a linear time algorithm for the *longest regular subsequence* problem. Finally, we should mention that the *scaling* method [5, 7–9] may be viewed as a *numerical halving* technique, whereas ours may be regarded as a *combinatorial halving* technique.

4. The Optimum Offset Problem

In this section we use the *halving* technique to obtain an $O(n)$ time algorithm for the *optimum offset* problem. This improves the best previously known $O(n \log n)$ time bound of [4, 15]. Furthermore, the constant time factor of our $O(n)$ algorithm is much smaller than those known previously. Therefore, this result is of practical as well as theoretical interest.

Based on Theorem 1, we may describe the *optimum offset* problem as follows: Find the minimum separation s^*, such that $l(s^*, x, y) \leqslant u(s^*, x, y)$. We call s^* the *optimum separation with respect to x and y* and denote it by $s^* = \operatorname{optsep}(n, x, y)$. Once s^* is known, then any offset d^*, such that $l(s^*, x, y) \leqslant d^* \leqslant u(s^*, x, y)$, is an optimum offset. First we state two useful lemmas.

Lemma 1. *The following hold for all s.*

(i) $u(s) \leqslant u(s+1)$,

(ii) $l(s) \geqslant l(s+1)$.

Proof. (i) Let $u(s+1) = x_{j+s+1} - y_j$, for some j, $0 \leqslant j < n-s-1$. Since x is in ascending order we have $u(s+1) \geqslant x_{j+s} - y_j$. It is obvious that $x_{j+s} - y_j \geqslant u(s)$. Therefore $u(s) \leqslant u(s+1)$. A similar proof holds for (ii). ∎

Lemma 2. *Let* $s^* = \operatorname{optsep}(n, x, y)$. *Then* $s^* \leqslant \lfloor n/2 \rfloor$; *and this bound is tight.*

Proof. Since x and y are in ascending order, we can conclude that $l(\lfloor n/2 \rfloor) \leqslant x_{\lfloor n/2 \rfloor} - y_{\lfloor n/2 \rfloor} \leqslant u(\lfloor n/2 \rfloor)$. Thus $s^* \leqslant \lfloor n/2 \rfloor$. To see that this bound is tight, consider the following example. Suppose $b_i = i+1$ (i.e., $y_i = 1$) for $0 \leqslant i < n$, $a_i = i$ for $0 \leqslant i < \lfloor n/2 \rfloor$, $a_i = i+2$ for $\lceil n/2 \rceil \leqslant i < n$. If n is odd, then $a_i = i+1$ for $i = \lfloor n/2 \rfloor$. For this example we have $l(\lfloor n/2 \rfloor - 1) > u(\lfloor n/2 \rfloor - 1)$. We conclude $s^* = \lfloor n/2 \rfloor$ for this example. ∎

We first describe a simple $O(n \log s^*) = O(n \log n)$ time algorithm to compute s^*. We then use the *halving* technique to obtain an $O(n)$ time algorithm for the same problem. From Lemma 1 we see that $u(s) - l(s)$ is ascending in s. We need to find the smallest s^* such that $u(s^*) - l(s^*) \geqslant 0$. A binary search on s suffices. We start with $s = 1$, and as long as $u(s) - l(s) < 0$ we double s. When this phase of the process is done, let us assume s has a value s'. Then $\lfloor s'/2 \rfloor \leqslant s^* \leqslant s'$. In the second phase of the process a binary search in the range from $\lfloor s'/2 \rfloor$ to s' is sufficient. We have examined $O(\log s^*)$ values of s during the entire process. For each value of s it takes $O(n)$ time to compute $u(s) - l(s)$ using Eqs. (1) and (2) of Section 2.

Therefore, this algorithm takes $O(n \log s^*)$ time. This is already a much simpler algorithm than that of [4, 15]. However, the worst case asymptotic bound is still $O(n \log n)$.

Now we describe an $O(n)$ time algorithm to compute s^* based on the *halving* technique. First we give an intuitive description of the idea. Then we consider a more formal development. Consider a feasible wiring of the n terminal pairs. Remove all odd-numbered wires. The remaining wires are separated from each other by at least two grid units. Now for each i move wire W_{2i} to the left by i units. These wires are now separated from each other by at least one horizontal unit. We can now decrease the separation by almost half (+ or − one extra track). This can be done by shifting all horizontal wire segments as far down as possible and shrinking the vertical wire segments to almost half. Conversely, if we have a feasible wiring for the latter problem (that is, odd terminals removed and even terminals moved left by i units), we can move even wires back to their original position. Now even wires are separated by at least two horizontal units. We can now double the vertical segments of these wires and if necessary add an extra track at the bottom and extend the wires down vertically. Now there is enough room for the odd wires to be routed in between even wires. The formal development follows.

Let x^e be the $\lceil n/2 \rceil$-vector, which is obtained from x by removing the odd indexed entries of x. That is, $x_i^e = x_{2i}$, for $0 \leqslant i < \lceil n/2 \rceil$. x^e is called the *even half of* x. Define y^e, the *even half of* y, similarly. Recall the definitions $u(s, x^e, y^e)$ and $l(s, x^e, y^e)$ from Eqs. (1) and (2). Let s^e be the optimum separation with respect to x^e and y^e. The following theorem states that $2s^e$ closely approximates s^*.

THEOREM 4. *Let* $s^* = \text{optsep}(n, x, y)$ *and* $s^e = \text{optsep}(\lceil n/2 \rceil, x^e, y^e)$. *Then* $|s^* - 2s^e| \leqslant 1$.

Proof. It follows immediately from our matrix formulation (in Section 2) that for any separation s, we have $u(2s, x, y) \leqslant u(s, x^e, y^e)$ and $l(2s, x, y) \geqslant l(s, x^e, y^e)$. If $s^e > 0$, then we have $u(2s^e - 2, x, y) - l(2s^e - 2, x, y) \leqslant u(s^e - 1, x^e, y^e) - l(s^e - 1, x^e, y^e) < 0$. This implies $s^* \geqslant 2s^e - 1$. The latter inequality obviously holds when $s^e = 0$. It remains to show that $s^* \leqslant 2s^e + 1$. Let $u(2s^e + 1, x, y) = x_{j+2s^e+1} - y_j$, for some j, $0 \leqslant j < n - 2s^e - 1$. Let $k = \lceil j/2 \rceil$. Then $u(2s^e + 1, x, y) \geqslant x_{2k+2s^e} - y_{2k} \geqslant u(s^e, x^e, y^e)$. Thus $u(2s^e + 1, x, y) \geqslant u(s^e, x^e, y^e)$. Similarly, $l(2s^e + 1, x, y) \leqslant l(s^e, x^e, y^e)$. Therefore $l(2s^e + 1, x, y) \leqslant u(2s^e + 1, x, y)$. This implies $s^* \leqslant 2s^e + 1$ and completes the proof. ∎

It follows from Theorem 4 that the algorithm below computes the optimum separation in $O(n)$ time.

function optsep (n, x, y);
  begin
    if $n \leqslant 1$ **then return** 0
    else begin
      $s :=$ opstep$(\lceil n/2 \rceil, x^e, y^e)$;

```
            if l(2s, x, y) > u(2s, x, y) then return 2s + 1
            else if s > 0 and l(2s - 1, x, y) ≤ u(2s - 1, x, y) then return 2s - 1
              else return 2s
    end
end.
```

THEOREM 5. *The function optsep(n, x, y) computes the optimum separation in $O(n)$ time. Thus, the optimum offset problem can be solved in $O(n)$ time.*

Proof. The correctness of the algorithm follows from Theorem 4, Lemma 1, and induction on n. Let $T(n)$ be the time complexity of the algorithm. The algorithm makes a recursive call with vectors x^e and y^e of length $\lceil n/2 \rceil$. This takes $T(\lceil n/2 \rceil)$ time. Then it computes $l(s)$ and $u(s)$ for at most two values of s. This takes an extra $O(n)$ time. From above we conclude that $T(n) = T(\lceil n/2 \rceil) + O(n)$ for $n > 1$. Thus $T(n) = O(n)$. ∎

5. OTHER PROBLEMS

The *feasible set* problem is to compute $l(s)$ and $u(s)$ for $s = 0, 1, \ldots, n-1$, according to Eqs. (1) and (2). The straightforward method takes $O(n^2)$ time. Lemma 1 asserts that $u(s)$ is ascending and $l(s)$ is descending with respect to s. We also see from Eqs. (1) and (2) that the problem is essentially that of computing the *discrete convolution* of two sorted integer vectors x and y with the operators *min* and —. An algorithm similar to the *Fast Fourier Transform* might exist; this question is open. Another approach might be based on the halving technique. The importance of this problem is justified by the following theorems.

THEOREM 6. *If the feasible set problem can be solved in $T(n)$ time, then the minimum area problem can be solved in $T(n) + O(n)$ time.*

Proof. The channel area for the feasible pair (s, d) is

$$A(s, d) = s[\max(b_{n-1} + d, a_{n-1}) - \min(b_0 + d, a_0)].$$

First solve the feasible set problem. In order to find the feasible pair (s, d) for the minimum area problem, we can consider the four critical offsets $l(s)$, $u(s)$, $a_0 - b_0$, and $a_{n-1} - b_{n-1}$, for each separation s. (Some of these offsets might be infeasible for some separations.) In this way we can compute the area for each of these separation and offset pairs and select the best choice. ∎

THEOREM 7. *If the feasible set problem can be solved in $T(n)$ time, then the minimum longest wire length problem can be solved in $T(n) + O(n)$ time.*

Proof. First solve the feasible set problem. Suppose (s, d) is a feasible pair. Then the length of wire W_i, denoted l_i, is the rectilinear distance of the terminals a_i and b_i. That is,

$$l_i = s + |a_i - b_i - d| = s + |x_i - y_i - d|. \tag{3}$$

Let LWL(s, d) denote the longest wire length with respect to (s, d). Let $p = \max(x_i - y_i)$ and $q = \min(x_i - y_i)$, where the min and max are taken over all i. (Note that $p = l(0)$ and $q = u(0)$.) Then

$$\mathrm{LWL}(s, d) = s + \max(|d - p|, |d - q|).$$

To simplify this expression even further, let $P = (p + q)/2$ and $Q = (p - q)/2$. Then

$$\mathrm{LWL}(s, d) = s + |d - P| + Q. \tag{4}$$

Note that P and Q in Eq. (4) are independent of (s, d) and can be computed in $O(n)$ time. Now we consider the three critical offsets $l(s)$, $u(s)$, and P for each separation and pick the best choice. Among these three offsets the best choice is the one that is in between the other two, assuming that $l(s) \leqslant u(s)$. Then we select the separation, along with the corresponding best offset, that gives the overal best result. ▮

THEOREM 8. *If the feasible set problem can be solved in $T(n)$ time, then the minimum total wire length problem can be solved in $T(n) + O(n \log n)$ time.*

Proof. Suppose (s, d) is a feasible pair. Then the length of wire W_i is as in Eq. (3). Therefore, for a fixed separation s, the total wire length decreases as the offset d approaches the median of $x_i - y_i$, $i = 0, 1, \ldots, n - 1$. We first sort $x_i - y_i$ for $i = 0, 1, \ldots, n - 1$. Let the sorted order be $d_0, d_1, \ldots, d_{n-1}$. This takes $O(n \log n)$ time. Let TWL(s, d) be the total wire length with respect to (s, d). Then

$$\mathrm{TWL}(s, d) = ns + \sum_{i=0}^{n-1} |d - d_i|. \tag{5}$$

Now for $k = 0, 1, \ldots, n$ we compute $A_k = \sum_{i=0}^{k-1} d_i$. This can be done in $O(n)$ time. Then we solve the feasible set problem, computing $l(s)$ and $u(s)$ for $s = 0, 1, \ldots, n - 1$. Then for each separation $s \geqslant s^*$, in ascending order, we select that offset d such that $l(s) \leqslant d \leqslant u(s)$ and $d_{k-1} \leqslant d \leqslant d_k$ and k is as close to $n/2$ as possible, and $(2k - n)\, d$ is minimized subject to the restriction on k. The total wire length for this pair (s, d) is

$$\mathrm{TWL}(s, d) = ns + (2k - n)\, d + A_n - 2A_k.$$

During this phase of the process, as s increases, d changes monotonically. Thus, this phase of the process can be done in $O(n)$ time as well. Now for each separation we have a minimum total wire length. We select the minimum among these. ▮

COROLLARY 1. *The following problems can be solved in $O(n^2)$ time.*

(i) *the feasible set problem,*

(ii) *the minimum area problem,*

(iii) *the minimum longest wire length problem,*

(iv) *the minimum total wire length problem.*

Proof. $T(n)$ in Theorems 6, 7, and 8 is $O(n^2)$ using the straightforward computation of the feasible set problem. ▮

6. CONCLUDING REMARKS

Our main result in this paper is an $O(n)$ time algorithm for the *optimum offset* problem based on the *halving* technique. We also show how an efficient solution of the *feasible set* problem implies efficient solutions of the *minimum area*, the *minimum longest wire length*, and the *minimum total wire length* problems. The following problems may be considered.

(1) Are there other applications of the *halving* technique? For instance, can we efficiently apply the *halving* technique to any of the feasible set, minimum area, and minimum wire length problems?

(2) The feasible set problem is a discrete convolution problem on sorted vectors and the *min* and — operators. Does there exist a *Fast Fourier Transform* type of algorithm for such a problem?

(3) It would be interesting to generalize the techniques developed in this paper to some other wire routing problems such as *placement of a circuit within a ring of pads* [2].

ACKNOWLEDGMENTS

I would like to thank the referees for their valuable suggestions. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant A5516. A preliminary version of this paper appeared in [12].

REFERENCES

1. A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, Mass., 1974.
2. B. S. BAKER AND R. Y. PINTER, An algorithm for the optimum placement and routing of a circuit within a ring of pads, *in* "Proceedings 24th Annual Symp. on Foundations of Computer Science, 1983," pp. 360–370.
3. A. E. BARATZ, "Algorithms for Integrated Circuit Signal Routing," Ph.D. dissertation, Department of EECS, MIT, Cambridge, Mass., 1981.

4. D. Dolev, K. Karplus, A. Siegel, A. Strong, and J. D. Ullman, Optimal wiring between rectangles, *in* "Proceedings 13th Annual ACM Symp. on Theory of Computing, 1981, pp. 312–317.
5. J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *J. Assoc. Comput. Mach.* **19**, No. 2 (1972), 248–264.
6. G. N. Frederickson and D. B. Johnson, Generalized selection and ranking: Sorted matrices, *SIAM J. Comput.* **13**, No. 1 (1984), 14–30.
7. H. N. Gabow, A scaling algorithm for weighted matching on general graphs, *in* "Proceedings, 26th Annual IEEE Symp. on Foundations of Computer Science, 1985," pp. 90–100.
8. H. N. Gabow, Scaling algorithms for network problems, *J. Comput. System Sci.* **31**, No. 2 (1985), 148–168.
9. H. N. Gabow, J. L. Bentley, and R. E. Tarjan, Scaling and related techniques for geometry problems, *in* "Proceedings, 16th Annual ACM Symp. on Theory of Computing, 1984," pp. 135–143.
10. D. E. Knuth, "The Art of Computer Programming," Vol. III, "Sorting and Searching," Addison–Wesley, Reading, Mass., 1973.
11. C. E. Leiserson and R. Y. Pinter, Optimal placement for river routing, *SIAM J. Comput.* **12**, No. 3 (1983), 447–462.
12. A. Mirzaian, Channel routing in VLSI, *in* "Proceedings, 16th Annual ACM Symp. on Theory of Computing, 1984," pp. 101–107.
13. A. Mirzalan, "A Halving technique for the Longest Regular Subsequence Problem," Technical Report CS-86-04, Department of Computer Science. York University, Toronto, Ontario, 1986.
14. A. Mirzaian and E. Arjomandi, Selection in $X+Y$ and matrices with sorted rows and columns, *Inform. Process. Lett.* **20** (1985), 13–17.
15. A. Siegel and D. Dolev, The separation for general single-layer wiring barriers, *in* "VLSI Systems and Computations" (H. T. Kung, B. Sproull, and G. Steele, Eds.), pp. 143–152, Comput. Sci. Press, Rockville, Md., 1981.
16. M. Tompa, An optimal solution to a wire-routing problem, *J. Comp. System Sci.* **23** (1981), 127–150.
17. J. D. Ullman, "Computational Aspects of VLSI," Comput. Sci. Press, Rockville, Md., 1983.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium