European Journal of Operational Research 58 (1992) 284–292
North-Holland

# A heuristic for Euclidean and rectilinear Steiner problems

J.E. Beasley
*The Management School, Imperial College, London SW7 2AZ, England, UK*

Received March 1988; revised December 1990

**Abstract:** In this paper we present a heuristic for Euclidean and rectilinear Steiner problems. This heuristic is based upon finding optimal Steiner solutions for connected subgraphs of the minimal spanning tree of the entire vertex set. Computational results are given for randomly generated problems involving up to 10000 vertices.

**Keywords:** Euclidean, rectilinear, Steiner, heuristic

## 1. Introduction

Let $i$ and $j$ be any two points in the Euclidean plane and let their coordinates be $(x_i, y_i)$ and $(x_j, y_j)$, respectively. Define the cost of the edge connecting $i$ and $j$ together to be equal to the Euclidean distance $[(x_i-x_j)^2+(y_i-y_j)^2]^{1/2}$ between $i$ and $j$. Then if $V$ is a set of $n$ points (vertices) in the plane, the Euclidean Steiner problem is the problem of connecting together the vertices in $V$ so as to minimise the total cost of the edges used. It is well-known that the solution to this problem will be the minimal spanning tree (MST) on some set of vertices $V \cup S$ where $S$ is called the set of Steiner vertices.

The rectilinear Steiner problem is the same as the Euclidean Steiner problem except that the cost of the edge connecting any two points $i$ and $j$ together is given by the rectilinear distance $|x_i-x_j|+|y_i-y_j|$.

Both of these problems are NP-complete [8,9]. They have received a fair amount of attention in the literature and many algorithms, both heuristic and optimal, have been proposed.

Since a comprehensive survey of work relating to Euclidean and rectilinear Steiner problems has recently been given by Hwang and Richards [18] we shall not give a complete literature survey here but instead concentrate upon heuristic algorithms and upon work which reports computational experience. The reader interested in a more complete description of the work that has been done on Euclidean and rectilinear Steiner problems is referred to [18].

### 1.1. Euclidean problems

Optimal solution algorithms for the Euclidean Steiner problem have been presented by Boyce and Seery [4], Cockayne and Schiller [7], Winter [34] and Cockayne and Hewgill [6].

These algorithms work by examining 'topologies' (a topology being a set of vertices and their associated edges) corresponding to full Steiner trees (FST's). An FST is a Steiner tree containing $s+2$ original vertices, each with degree one, and $s$ Steiner vertices, each with degree three. Pictures of FST's for $s=0, 1, \ldots, 7$ can be found in Gilbert and Pollak [10]. Optimal solution algorithms can only solve problems involving up to 30 vertices.

Chang [5] presented an early heuristic algorithm based upon inserting vertices into the MST in order to reduce the cost of the tree. This is a natural approach and has been used in many algorithms (e.g. Korhonen [19] and Smith and Liebman [31]). Smith, Lee and Liebman [30] presented an algorithm based upon Voronoi dia-

grams and Delaunay triangulations. Lundy [22] presented an algorithm based upon simulated annealing.

### 1.2. Rectilinear problems

The only optimal algorithm for the rectilinear Steiner problem that has been presented in the literature is due to Yang and Wing [35,37,38]. This algorithm can only solve very small problems involving up to 10 vertices.

Yang and Wing [35–38] also presented a heuristic algorithm based upon branch and bound. Lee, Bose and Hwang [21] presented an algorithm similar to the Prim [24] algorithm for the MST (see also Hwang [15]). Smith and Liebman [31] and Smith, Lee and Liebman [29] presented algorithms for the rectilinear problem similar to their algorithms [30,31] for the Euclidean problem.

Servit [26] investigated the performance of eight simple heuristics on example problems drawn from the design of printed circuit boards. Hsu, Pan and Kubitz [14] presented algorithms based upon the Prim [24] and Kruskal [20] algorithms for the MST. Basart and Huguet [1] presented an algorithm based upon dividing the rectilinear plane into two. Richards [25] presented an efficient implementation of an algorithm due to Hanan [11]. Ho, Vijayan and Wong [13] presented an algorithm based upon transforming the rectilinear MST into a rectilinear Steiner tree.

## 2. Heuristic

In this section we present the heuristic algorithm we have developed and discuss its application to Euclidean and rectilinear Steiner problems.

Essentially the heuristic considers all connected subgraphs of the MST of $V$ which contain four vertices, finds the optimal Steiner tree for each such subgraph and adds to $V$ selected Steiner vertices from these subgraphs. The details are as follows:

(1) Let $T(K)$ represent the cost of connecting a set $K$ of vertices together via their MST and let $T_S(K)$ represent the cost of connecting the same set of vertices together via their *optimal* Steiner tree. Let $V_0$ be the initial vertex set and let $t$ be an iteration counter. Set $V = V_0$ and $t = 0$.

(2) Find the MST of $V$. This can be easily accomplished, e.g. using [24] or using [27,28] for Euclidean problems and [16] for rectilinear problems. Update the iteration counter using $t = t + 1$.

(3) Define:

$L = \{[p, q, r, s] \mid p, q, r, s \in V;\ p, q, r, s$ all distinct vertices and $[p, q, r, s]$ constitutes a connected subgraph with respect to the MST of $V\}$.

The set $L$ is easily found since it is trivial to show that to enumerate $L$ we need only consider those vertex sets $[p, q, r, s]$ for which we have either:

(a) a path in the MST consisting of edges $p - q$, $q - r$ and $r - s$; or

(b) a 'star' configuration in the MST centred on $p$ and consisting of edges $p - q$, $p - r$ and $p - s$.

(4) For all vertex sets $K \in L$ calculate the reduction in cost (if any) which occurs when the vertices in $K$ are connected via their optimal Steiner tree. This reduction is given by $R(K) = T(K) - T_S(K)$ where $R(K) \geqslant 0$.

(5) If $\max[R(K) \mid K \in L] = 0$, then we have no vertex set that we can use to reduce the cost of the MST of $V$, so stop where, if $V_F$ is the final vertex set, the cost of connecting together the vertices in $V_0$ has been reduced (in $t$ iterations) from $T(V_0)$ to $T(V_F)$ via introduction of the Steiner vertices in $V_F - V_0$.

(6) Let $M$ be a set of vertices where initially, at each iteration $t$, $M$ is empty. Sort the vertex sets $K \in L$ into descending $R(K)$ order and run down this list where, for each such $K$, if $|M \cap K| = 0$ and $R(K) > 0$, we:

(a) add the Steiner vertices associated with $T_S(K)$ to $V$; and

(b) set $M = M \cup K$ (i.e. add the vertices in $K$ to $M$).

Informally we are adding Steiner vertices associated with the maximum reduction we can find provided that we have no possibility of 'interfering' with previously added Steiner vertices.

(7) Find the MST of $V$ and let $E$ be the set of edges associated with this MST. If there exist any vertices $i \in V - V_0$ which have degree $\leqslant 2$ with respect to $E$, then remove these vertices from $V$ (since plainly we reduce, or leave unchanged, the

cost of the MST by so doing) and adjust $E$ accordingly.

(8) For each vertex $i \in V - V_0$ which has degree three with respect to $E$, move this vertex to its optimal Steiner location (which is easily found) and repeat this step until there is no further reduction in the cost of the tree based on the edge set $E$.

(9) If the problem is a Euclidean one then we use the algorithm of Hwang [17] for constructing full Steiner trees (FST's) in an attempt to improve the solution. This can be done as follows:

Find the MST of $V$ and for each subtree (vertex set $V^*$, cost $C^*$) of this MST which has a topology appropriate to a FST:

(a) apply the algorithm of Hwang [17] to find the FST for the subtree;

(b) if the cost of this FST is less than C*, move the vertices in $V^* - V_0$ to the locations given by the FST;

(c) if the cost of this FST is greater than $C^*$, then improve the current solution (if possible) by removing from $V$ the vertex $i$ corresponding to

$$T(V^* - \{i\}) = \min[T(V^* - [j]) \mid j \in V^* - V_0, \quad T(V^* - \{j\}) \leqslant C^*],$$

i.e. remove the vertex that leads to the lowest-cost subtree after removal.

(10) Go to step (2).

In applying the above heuristic to Euclidean and rectilinear Steiner problems, the only difference lies in the calculation of the optimal Steiner tree $T_S(K)$ where $K = [p, q, r, s]$ consists of four distinct vertices. We consider each problem type in turn.

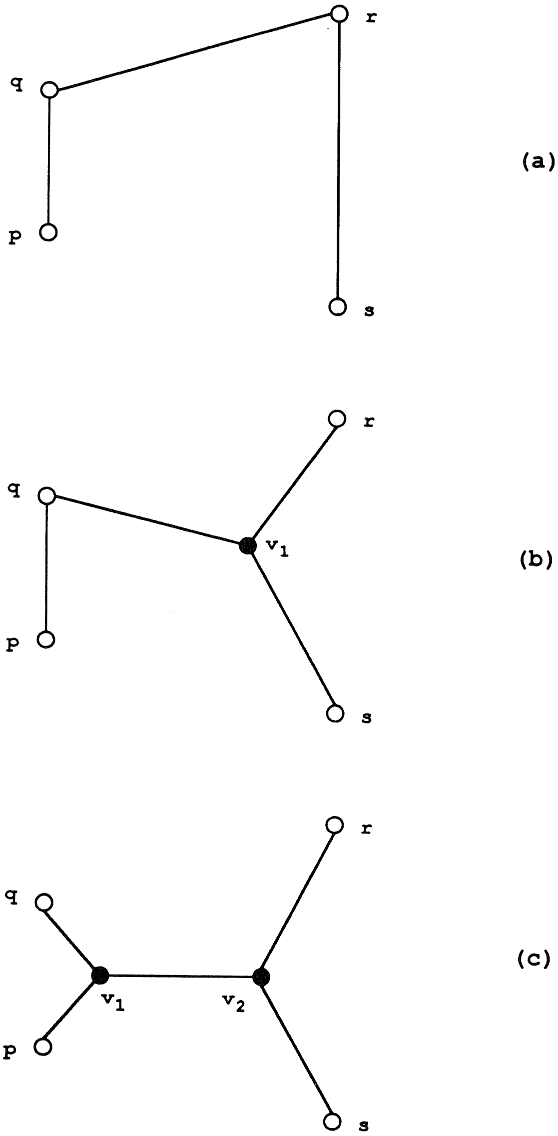

Figure 1

### 2.1. Euclidean problems

It is well-known (e.g. see [10]) that for Euclidean problems any Steiner vertices in the optimal Steiner tree must have (vertex) degree three with 120 degrees between the edges incident at the Steiner vertex. Then $T_S(K)$ is either:

(a) the MST of the vertex set $K$; or

(b) one of the four cases where we have three vertices from $K$ joined via a single Steiner vertex ($v_1$, say) and the remaining vertex ($i$, say) connected to its nearest vertex in $K - \{i\}$; or

(c) one of the two cases where, ordering the vertices in $K$ such that the four edges $p - q$, $q - r$, $r - s$ and $s - p$ constitute a quadrilateral whose interior angles add up to 360 degrees, we have $p$, $q$, $r$ and $s$ connected via two Steiner vertices ($v_1$ and $v_2$, say).

Figure 1 illustrates this diagrammatically. For a proof of the fact that (a), (b) and (c) above are sufficient to define the *optimal* Steiner tree for four vertices, see Gilbert and Pollak [10]. Note here that, depending upon the position of $p$, $q$, $r$ and $s$, some of the cases in (b) and (c) above may be redundant (i.e. it may not be possible to find Steiner vertices).

In order to locate the single Steiner vertex associated with three vertices some simple formulae are used (e.g. see Thompson [33]). To locate the two Steiner vertices associated with four vertices some simple geometry is used (e.g. see Melzak [23]).

### 2.2. Rectilinear problems

For rectilinear problems $T_S(K)$ can be calculated directly using the inner-rectangle construction given by Hanan [12]. However the difficulty with this construction, from our point of view, is that it does not uniquely define the position of the Steiner vertices. This contrasts with the situation when we are connecting three vertices together where the position of the Steiner vertex is uniquely defined (see Hanan [12]). Hence, for this reason, we prefer to regard $T_S(K)$ as being either:

(a) the MST of the vertex set $K$; or

(b) one of the four cases where we have three vertices from $K$ joined via a single Steiner vertex ($v_1$, say) and the remaining vertex ($i$, say) connected to its nearest vertex in $K - \{i\} + \{v_1\}$; or

(c) defined via the inner–rectangle construction of Hanan [12].

The advantage of this approach is that we can adopt (a) or (b) above (thereby uniquely defining the position of the Steiner vertex (if any)) if the value for $T_S(K)$ calculated from (a) or (b) above is equal to the value for $T_S(K)$ calculated from (c) above.

The location of the single Steiner vertex associated with three vertices $i$, $j$ and $k$ is (median $(x_i, x_j, x_k)$, median $(y_i, y_j, y_k)$) (see Hanan [12]). In the event that $T_S(K)$ was defined from the inner-rectangle construction ((c) above) we took as Steiner vertices those corners of the inner rectangle which had at least one vertex of $K$ transferred to them (see Hanan [12]).

## 3. Computational results

The heuristic presented in this paper was programmed in FORTRAN and run on a Cray X-MP/28 using the CF77 compiling system (with maximum optimisation) for a number of randomly generated problems. These problems consisted of $n$ points randomly distributed in a unit square where, for each value of $n$, 15 test problems were generated. Note here that all of the test problems considered in this paper are publically available via electronic mail from OR-Library [3].

Each test problem was solved both as a Euclidean problem and as a rectilinear problem. The results are shown in Table 1. In that table we give, for each value of $n$, the minimum, mean, maximum and standard deviation for:

(a) the number of iterations;

(b) the percentage reduction achieved $[100(T(V_0) - T(V_F))/T(V_0)]$;

(c) the number of Steiner vertices $[|V_F| - n]$; and

(d) the total time (in Cray X-MP/28 seconds).

Note here that a least-squares linear regression using the results shown in Table 1 indicates that the average computer time required to solve a problem on the Cray X-MP/28 is $O(n^{1.317})$ for Euclidean problems and $O(n^{1.253})$ for rectilinear problems.

In Tables 2 and 3 we compare the average percentage reduction and the average computer time for the heuristic presented in this paper with the results presented by other workers (for varying values of $n$). It is clear from those tables that the heuristic presented in this paper gives larger percentage reductions (better-quality solutions) in most instances.

Recently Richards [25] reported solving rectilinear problems of size $n = 10000$ with an average percentage reduction ranging from 3.95 to 3.99%. In order to compare the heuristic presented in this paper with this result we generated and solved one problem of size $n = 10000$. The percentage reduction achieved was 9.981% in 6 iterations, involved 4438 Steiner vertices and was obtained in 291.305 Cray X-MP/28 seconds. When this problem was solved as a Euclidean problem the percentage reduction achieved was 3.000% in 15 iterations, involved 4005 Steiner vertices and was obtained in 2728.923 Cray X-MP/28 seconds.

In order to compare the heuristic presented in this paper with some optimal solutions we solved the 46 test problems given by Soukup and Chow [32]. For a number of these test problems the Euclidean optimal solution is known from previous work [32]. In order to generate rectilinear optimal solutions for these test problems each rectilinear test problem was converted into an

Table 1
Computational results

| Problem type | n | Number of iterations | | | | Percentage reduction | | | | Number of Steiner vertices | | | | Total time (Cray X-MP/28 seconds) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | min | mean | max | sd | min | mean | max | sd | min | mean | max | sd | min | mean | max | sd |
| Euclidean | 10 | 2 | 3.000 | 4 | 0.655 | 0.477 | 3.138 | 6.168 | 1.863 | 2 | 3.400 | 6 | 1.183 | 0.018 | 0.044 | 0.091 | 0.023 |
| | 20 | 3 | 3.600 | 5 | 0.737 | 1.389 | 3.015 | 4.737 | 1.008 | 6 | 7.667 | 10 | 1.447 | 0.076 | 0.114 | 0.191 | 0.033 |
| | 30 | 3 | 3.533 | 5 | 0.640 | 2.059 | 2.868 | 4.752 | 0.721 | 9 | 11.667 | 16 | 2.160 | 0.112 | 0.171 | 0.239 | 0.048 |
| | 40 | 3 | 5.133 | 21 | 4.642 | 1.669 | 3.024 | 4.174 | 0.631 | 13 | 15.733 | 18 | 1.668 | 0.157 | 0.335 | 1.394 | 0.318 |
| | 50 | 3 | 4.067 | 7 | 0.961 | 2.138 | 2.841 | 3.620 | 0.400 | 16 | 19.333 | 22 | 2.093 | 0.197 | 0.326 | 0.594 | 0.010 |
| | 60 | 3 | 4.533 | 11 | 1.885 | 2.347 | 2.946 | 3.576 | 0.404 | 18 | 23.733 | 28 | 2.685 | 0.259 | 0.478 | 1.326 | 0.246 |
| | 70 | 3 | 4.133 | 6 | 0.990 | 2.142 | 2.844 | 3.592 | 0.363 | 22 | 26.467 | 31 | 2.973 | 0.302 | 0.480 | 0.733 | 0.143 |
| | 80 | 3 | 4.467 | 9 | 1.407 | 1.973 | 2.817 | 4.288 | 0.623 | 28 | 31.667 | 35 | 2.193 | 0.359 | 0.665 | 1.694 | 0.318 |
| | 90 | 3 | 4.733 | 8 | 1.335 | 1.989 | 2.935 | 3.687 | 0.454 | 32 | 36.733 | 41 | 2.120 | 0.406 | 0.812 | 1.476 | 0.272 |
| | 100 | 4 | 5.333 | 9 | 1.543 | 2.286 | 2.952 | 3.467 | 0.370 | 32 | 39.333 | 47 | 3.519 | 0.675 | 0.979 | 1.650 | 0.322 |
| | 250 | 4 | 5.867 | 15 | 2.642 | 2.611 | 2.950 | 3.279 | 0.206 | 92 | 98.467 | 106 | 3.292 | 1.936 | 2.877 | 7.738 | 1.376 |
| | 500 | 5 | 6.400 | 9 | 1.298 | 2.668 | 3.052 | 3.316 | 0.169 | 190 | 200.533 | 209 | 5.963 | 5.017 | 6.855 | 9.842 | 1.508 |
| | 1000 | 5 | 8.067 | 14 | 2.840 | 2.807 | 3.017 | 3.283 | 0.128 | 383 | 401.400 | 416 | 8.951 | 11.754 | 21.174 | 35.212 | 7.105 |
| Rectilinear | 10 | 3 | 3.467 | 5 | 0.640 | 4.377 | 9.947 | 17.147 | 3.772 | 2 | 3.867 | 5 | 0.834 | 0.010 | 0.016 | 0.030 | 0.006 |
| | 20 | 3 | 3.733 | 5 | 0.594 | 6.098 | 10.590 | 14.218 | 2.114 | 6 | 8.467 | 10 | 1.246 | 0.024 | 0.032 | 0.046 | 0.005 |
| | 30 | 3 | 4.067 | 5 | 0.704 | 7.448 | 10.250 | 15.095 | 1.957 | 10 | 12.667 | 15 | 1.447 | 0.030 | 0.048 | 0.066 | 0.011 |
| | 40 | 4 | 4.133 | 5 | 0.352 | 7.237 | 9.556 | 12.783 | 1.715 | 15 | 17.533 | 21 | 1.642 | 0.059 | 0.067 | 0.087 | 0.010 |
| | 50 | 3 | 4.133 | 5 | 0.640 | 7.585 | 9.522 | 11.929 | 1.351 | 17 | 21.467 | 25 | 2.295 | 0.058 | 0.084 | 0.110 | 0.016 |
| | 60 | 4 | 4.267 | 5 | 0.458 | 8.302 | 10.146 | 12.893 | 1.273 | 23 | 26.400 | 30 | 1.993 | 0.090 | 0.117 | 0.160 | 0.020 |
| | 70 | 4 | 4.067 | 5 | 0.258 | 7.383 | 9.779 | 12.063 | 1.100 | 26 | 30.133 | 38 | 2.997 | 0.108 | 0.125 | 0.149 | 0.015 |
| | 80 | 4 | 4.467 | 5 | 0.516 | 7.967 | 9.831 | 12.044 | 1.151 | 31 | 35.800 | 40 | 2.336 | 0.141 | 0.180 | 0.225 | 0.025 |
| | 90 | 4 | 4.400 | 5 | 0.507 | 7.910 | 10.128 | 11.941 | 1.037 | 37 | 39.333 | 44 | 2.350 | 0.160 | 0.195 | 0.266 | 0.031 |
| | 100 | 4 | 4.267 | 5 | 0.458 | 8.337 | 10.139 | 11.904 | 1.125 | 38 | 43.800 | 51 | 3.364 | 0.170 | 0.218 | 0.281 | 0.033 |
| | 250 | 4 | 4.733 | 6 | 0.594 | 9.099 | 9.964 | 10.708 | 0.431 | 100 | 111.400 | 119 | 5.705 | 0.486 | 0.676 | 0.960 | 0.134 |
| | 500 | 4 | 5.133 | 6 | 0.516 | 9.091 | 9.879 | 10.856 | 0.482 | 207 | 219.533 | 234 | 6.906 | 1.260 | 1.619 | 2.125 | 0.220 |
| | 1000 | 5 | 5.600 | 7 | 0.632 | 9.492 | 9.888 | 10.615 | 0.338 | 426 | 440.600 | 455 | 8.509 | 4.070 | 5.071 | 6.371 | 0.631 |

Table 2
Algorithm comparison for Euclidean problems [a]

| $n$ | Heuristic | | Chang [5] | | Korhonen [19] | | Smith and Liebman [31] | | Smith, Lee and Liebman [30] | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Average percentage reduction | Average computer time Cray X-MP/28 seconds | Average percentage reduction | Average computer time IBM 360/65 seconds | Average percentage reduction | Average computer time Burroughs B6700 seconds | Average percentage reduction | Average computer time DEC-10 seconds | Average percentage reduction | Average computer time DEC-10 seconds |
| 10 | 3.138 | 0.044 | 2.200 | 0.922 | 1.50 | 0.327 | 2.873 | 0.458 | 3.173 | 0.293 |
| 20 | 3.015 | 0.114 | 3.012 | 10.561 | 2.25 | 0.945 | 1.749 | 2.145 | 2.333 | 0.569 |
| 30 | 2.868 | 0.171 | 3.087 | 54.906 | 2.51 | 1.706 | 2.214 | 5.790 | 2.769 | 0.799 |
| 40 | 3.024 | 0.335 | | | 2.54 | 2.280 | 1.380 | 12.790 | 2.663 | 1.089 |
| 50 | 2.841 | 0.326 | | | 2.39 | 2.757 | | | 2.568 | 1.375 |
| Number of test problems | 15 | | 20 | | 5 | | 15 | | 15 | |

[a] Lundy [22] has presented his results in a way that makes direct comparison with the above work impossible.

Table 3
Algorithm comparison for rectilinear problems [b]

| $n$ | Heuristic | | Lee, Bose and Hwang [21] [a] | | Smith and Liebman [31] | | Smith, Lee and Liebman [29] | | Basart and Huguet [1] | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Average percentage reduction | Average computer time Cray X-MP/28 seconds | Average percentage reduction | Average computer time PDP-10 seconds | Average percentage reduction | Average computer time DEC-10 seconds | Average percentage reduction | Average computer time DEC-10 seconds | Average percentage reduction | Average computer time |
| 10 | 9.947 | 0.016 | 7– 8 | 0.4 | 7.100 | 0.746 | 8.316 | 0.256 | 9.4 | ? |
| 20 | 10.590 | 0.032 | 8–10 | 1.6 | 7.621 | 5.454 | 7.650 | 0.519 | 6.5 | ? |
| 30 | 10.250 | 0.048 | 8–10 | 3.0 | 7.978 | 16.309 | 8.306 | 0.839 | | |
| 40 | 9.556 | 0.067 | | | 5.887 | 34.835 | 8.641 | 1.051 | | |
| Number of test problems | 15 | | 50 | | 15 | | 15 | | 5 | |

[a] For Lee, Bose and Hwang [21] a range for the *median* percentage reduction is given.

[b] For Yang and Wing [35–38], Hwang [15] and Hsu, Pan and Kubitz [14] the computational results given are insufficient to enable a comparison to be made with the work presented above. Servit [26] has presented his results in a way that makes direct comparison with the above work impossible. Ho, Vijayan and Wong [13] report that the average percentage reduction for their algorithm is approximately constant at 9.1% for all values of $n$ between 5 and 100. Richards [25] reports an average percentage reduction for $n = 100$ ranging from 4.02 to 5.93%.

Table 4
Computational results – Soukup and Chow [32] test problems [a]

| Problem number | $n$ | Euclidean | | Rectilinear | |
|---|---|---|---|---|---|
| | | Heuristic solution | Optimal solution | Heuristic solution | Optimal solution |
| 1 | 5 | 1.66440 | – [b] | 1.87 | – |
| 2 | 6 | 1.50050 | – | 1.68 | 1.64 |
| 3 | 7 | 2.07767 | – | 2.36 | – |
| 4 | 8 | 2.13879 | – | 2.54 | – |
| 5 | 6 | 2.04405 | – | 2.29 | 2.26 |
| 6 | 12 | 2.22239 | 2.1842 | 2.48 | 2.42 |
| 7 | 12 | 2.20529 | n/k [c] | 2.54 | 2.48 |
| 8 | 12 | 2.17779 | – | 2.42 | 2.36 |
| 9 | 7 | 1.58783 | n/k | 1.72 | 1.64 |
| 10 | 6 | 1.64728 | 1.5988 | 1.84 | 1.77 |
| 11 | 6 | 1.27411 | – | 1.44 | – |
| 12 | 9 | 1.64853 | n/k | 1.80 | – |
| 13 | 9 | 1.27338 | n/k | 1.50 | – |
| 14 | 12 | 2.20492 | – | 2.60 | – |
| 15 | 14 | 1.23041 | – | 1.48 | n/k |
| 16 | 3 | 1.16678 | – | 1.60 | – |
| 17 | 10 | 1.64279 | n/k | 2.01 | 2.00 |
| 18 | 62 | 3.85130 | n/k | 4.06 | n/k |
| 19 | 14 | 1.72225 | n/k | 1.90 | n/k |
| 20 | 3 | 1.03962 | – | 1.12 | – |
| 21 | 5 | 1.81818 | n/k | 2.16 | 1.92 |
| 22 | 4 | 0.50329 | – | 0.63 | – |
| 23 | 4 | 0.51303 | – | 0.65 | – |
| 24 | 4 | 0.25282 | – | 0.30 | – |
| 25 | 3 | 0.19897 | – | 0.23 | – |
| 26 | 3 | 0.12435 | – | 0.15 | – |
| 27 | 4 | 1.17817 | – | 1.33 | – |
| 28 | 4 | 0.20442 | – | 0.24 | – |
| 29 | 3 | 1.46598 | – | 2.00 | – |
| 30 | 12 | 1.03323 | n/k | 1.10 | – |
| 31 | 14 | 2.34009 | n/k | 2.60 | n/k |
| 32 | 19 | 2.85677 | n/k | 3.23 | n/k |
| 33 | 18 | 2.22953 | n/k | 2.69 | n/k |
| 34 | 19 | 2.13813 | n/k | 2.54 | n/k |
| 35 | 18 | 1.35545 | n/k | 1.54 | n/k |
| 36 | 4 | 0.87891 | – | 0.90 | – |
| 37 | 8 | 0.76603 | n/k | 0.90 | – |
| 38 | 14 | 1.43501 | 1.4248 | 1.66 | n/k |
| 39 | 14 | 1.43125 | – | 1.66 | n/k |
| 40 | 10 | 1.41803 | n/k | 1.62 | 1.55 |
| 41 | 20 | 1.97672 | n/k | 2.24 | n/k |
| 42 | 15 | 1.31535 | n/k | 1.53 | – |
| 43 | 16 | 2.36719 | n/k | 2.66 | n/k |
| 44 | 17 | 2.19744 | n/k | 2.61 | n/k |
| 45 | 19 | 1.93584 | n/k | 2.26 | n/k |
| 46 | 16 | 1.42209 | n/k | 1.50 | – |

[a] The average computation time needed to produce the heuristic results was 0.060 Cray X-MP/28 seconds for Euclidean problems and 0.012 Cray X-MP/28 seconds for rectilinear problems.

[b] – means that the optimal solution is the same as the heuristic solution.

[c] n/k means that the optimal solution is not known.

equivalent Steiner problem on a graph (see Hanan [12]) and this was then solved using the algorithm given by Beasley [2]. The results are shown in Table 4.

Examining Table 4 it is clear that, as we would expect, when $n \leqslant 4$ the heuristic presented in this paper always finds the optimal solution (for Euclidean or rectilinear problems).

Of the 13 Euclidean problems with $n \geqslant 5$ for which the optimal solution is known, the heuristic presented in this paper finds the optimal solution in all but 3 instances.

Of the 21 rectilinear problems with $n \geqslant 5$ for which the optimal solution is known, the heuristic presented in this paper finds the optimal solution in all but 10 instances.

## 4. Conclusions

In this paper we have presented a heuristic for Euclidean and rectilinear Steiner problems based upon finding optimal Steiner solutions for connected subgraphs of the minimal spanning tree of the entire vertex set. Computational results indicated that this heuristic gives better-quality solutions than other heuristics.

## Acknowledgement

We would like to acknowledge the comments provided by referees on earlier versions of this paper.

## References

[1] Basart, J.M., and Huguet, L., "An heuristic algorithm for rectilinear Steiner minimal trees", Working paper, Departament d'Informàtica, Facultat de Ciències, Universitat Autònoma de Barcelona, 08193 Bellaterra, 1987.

[2] Beasley, J.E., "An SST-based algorithm for the Steiner problem in graphs", *Networks* 19 (1989) 1–16.

[3] Beasley, J.E., "OR-Library: distributing test problems by electronic mail", *Journal of the Operational Research Society* 41 (1990) 1069–1072.

[4] Boyce, W.M., and Seery, J.B., "STEINER72, an improved version of Cockayne and Schiller's program STEINER for the minimal network problem", Computer Science Technical Report Number 35, Bell Laboratories, Murray Hill, NJ, 1972.

[5] Chang, S.K., "The generation of minimal trees with a Steiner topology", *Journal of the ACM* 19 (1972) 699–711.

[6] Cockayne, E.J., and Hewgill, D.E., "Exact computation of Steiner minimal trees in the plane", *Information Processing Letters* 22 (1986) 151–156.

[7] Cockayne, E.J., and Schiller, D.G., "Computation of Steiner minimal trees", in: D.J.A. Welsh and D.R. Woodall (eds.), *Combinatorics*, Institute of Mathematics and its Applications, Southend, 1972, 53–71.

[8] Garey, M.R., Graham, L.R., and Johnson, D.S., "The complexity of computing Steiner minimal trees", *SIAM Journal on Applied Mathematics* 32 (1977) 835–859.

[9] Garey, M.R., and Johnson, D.S., "The rectilinear Steiner tree problem is NP-complete", *SIAM Journal on Applied Mathematics* 32 (1977) 826–834.

[10] Gilbert, E.N., and Pollak, H.O., "Steiner minimal trees", *SIAM Journal on Applied Mathematics* 16 (1968) 1–29.

[11] Hanan, M., "Net wiring for large scale integrated circuits", IBM Research Report RC 1375, 1965.

[12] Hanan, M., "On Steiner's problem with rectilinear distance", *SIAM Journal on Applied Mathematics* 14 (1966) 255–265.

[13] Ho, J., Vijayan, G., and Wong, C.K., "A new approach to the rectilinear Steiner tree problem", *26th ACM/IEEE Design Automation Conference*, 1989, 161–166.

[14] Hsu, Y.C., Pan, Y., and Kubitz, W.J., "A path selection global router", *24th ACM/IEEE Design Automation Conference*, 1987, 641–644.

[15] Hwang, F.K., "An O(*n* log *n*) algorithm for suboptimal rectilinear Steiner trees", *IEEE Transactions on Circuits and Systems* 26 (1979) 75–77.

[16] Hwang, F.K., "An O(*n* log *n*) algorithm for rectilinear minimal spanning trees", *Journal of the ACM* 26 (1979) 177–182.

[17] Hwang, F.K., "A linear time algorithm for full Steiner trees", *Operations Research Letters* 4 (1986) 235–237.

[18] Hwang, F.K., and Richards, D.S., "Steiner tree problems", Working paper, available from the first author at AT&T Bell Laboratories, Murray Hill, NJ, 1989.

[19] Korhonen, P., "An algorithm for transforming a spanning tree into a Steiner tree", *Survey of Mathematical Programming (Proceedings of the 9th International Mathematical Programming Symposium)* 2 (1979) 349–357.

[20] Kruskal, J.B., "On the shortest spanning subtree of a graph and the travelling salesman problem" *Proceedings of the American Mathematical Society* 7 (1956) 48–50.

[21] Lee, J.L., Bose, N.K., and Hwang, F.K., "Use of Steiner's problem in suboptimal routing in rectilinear metric", *IEEE Transactions on Circuits and Systems* 23 (1976) 470–476.

[22] Lundy, M., "Applications of the annealing algorithm to combinatorial problems in statistics", *Biometrika* 72 (1985) 191–198.

[23] Melzak, Z.A., *Companion to Concrete Mathematics*, Wiley, New York, 1973, 138–145.

[24] Prim, R.C., "Shortest connection networks and some generalisations", *Bell System Technical Journal* 36 (1957) 1389–1401.

[25] Richards, D., "Fast heuristic algorithms for rectilinear Steiner trees", *Algorithmica* 4 (1989) 191–207.

[26] Servit, M., "Heuristic algorithms for rectilinear Steiner trees", *Digital Processes* 7 (1981) 21–32.

[27] Shamos, M.I., "Geometric complexity", *Proceedings of the 7th Annual ACM SIGACT Symposium*, 1975, 224–233.

[28] Shamos, M.I., and Hoey, D., "Closest point problems", *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, 1975, 151–162.

[29] Smith, J.M., Lee, D.T., and Liebman, J.S., "An O(*n* log *n*) heuristic algorithm for the rectilinear Steiner minimal tree problem", *Engineering Optimization* 4 (1980) 179–192.

[30] Smith, J.M., Lee, D.T., and Liebman, J.S., "An O(*n* log *n*) heuristic for Steiner minimal tree problems on the Euclidean metric", *Networks* 11 (1981) 23–39.

[31] Smith, J.M., and Liebman, J.S., "Steiner trees, Steiner circuits and the interference problem in building design", *Engineering Optimization* 4 (1979) 15–36.

[32] Soukup, J., and Chow, W.F., "Set of test problems for the minimum length connection networks", *ACM/SIGMAP Newsletter* 15 (1973) 48–51.

[33] Thompson, E.A., "The method of minimum evolution", *Annals of Human Genetics London* 36 (1973) 333–340.

[34] Winter, P., "An algorithm for the Steiner problem in the Euclidean plane," *Networks* 15 (1985) 323–345.

[35] Yang, Y.Y., and Wing, O., "An algorithm for the wiring problem", *Digest of the IEEE International Symposium on Electrical Networks* (1971) 14–15.

[36] Yang, Y.Y., and Wing, O., "Suboptimal algorithm for a wire routing problem", *IEEE Transactions on Circuit Theory* 19 (1972) 508–510.

[37] Yang, Y.Y., and Wing, O., "Optimal and suboptimal solution algorithms for the wiring problem", *Proceedings of the IEEE International Symposium on Circuit Theory*, 1972, 154–158.

[38] Yang, Y.Y., and Wing, O., "On a multinet wiring problem", *IEEE Transactions on Circuit Theory* 20 (1973) 250–252.