First Joint Soviet-German Workshop
"Digital Image Processing"
Berlin, November 5-9, 1990

Vertex-Weighted Steiner Tree Problem in Graphs

Oleg G. Taraszow
Institute of Computer Architecture
in Central Institute of Cybernetics
and Information Processes
Kurstraße 33, O-1086 Berlin

ABSTRACT

We study the general vertex-weighted Steiner tree problem in graphs which is an extension of the standard Steiner tree problem by the addition of vertex-associated weights. This article deals also with a such special case of this problem as the vertex-weighted optimal paths problem and its modifications and with the vertex-weighted generalized 1-median-problem. Polynomial algorithms are presented for the exact solution of the last problems and some heuristic algorithms for the vertex-weighted Steiner tree problem in graphs. The memory requirement is linear with respect to sizes of graphs.

1. Introduction

Let $G = (V,E)$ be a finite, undirected graph with sets V and E of vertices and edges, respectively. The sets V and E of the graph G are denoted by V(G) and E(G), too. Let $V = Z \cup S$ and $n := |V(G)|$, $m := |E(G)|$, and $k := |S|$. The vertices belonging to Z are called Z-vertices and the remaining vertices are called *Steiner vertices* (or S-vertices). *Cost functions* φ and ψ for the vertex and edge sets are given as follows:

$$\varphi: V(G) \longrightarrow \mathbb{R} \tag{1}$$

$$\psi: E(G) \longrightarrow \mathbb{R} \tag{2}$$

For any subgraph $Q \subseteq G$ the total cost C(Q) of Q is defined in the following way

$$C(Q) := \sum_{x \in V(Q)} \varphi(x) + \sum_{r \in E(Q)} \psi(x) \tag{3}$$

Formally, the Vertex-weighted Steiner Tree Problem in Graphs (VSTG - problem) can be formulated as follows.

Problem (VSTG-problem).

GIVEN: An graph G; edge and vertex cost functions (1) and (2), respectively, and a subset $Z \subseteq V(G)$.

FIND: A subgraph $Q \subseteq V(G)$ such that there is a path between every pair of Z-vertices, and the total cost (3) of Q is a minimum.

In the case $\varphi \equiv 0$ the VSTG-problem reduces to the well-known standard Steiner tree problem in graphs. The Steiner tree problem in graphs is known to be NP-complete [1] and several exact and heuristic algorithms for its solutions have been proposed. The excellent survey on the Steiner tree problems in graphs has been given by *Winter* [2].

The special case of the VSTG-problem, where the set of vertices, which must be included in the solution tree, consists of a single node, and all vertices weights are negative, is investigated by *Segev* [3]. It is shown in [3] that this special case is also NP-complete, its integer programming formulation is given and heuristic procedures are proposed. Analogous to [3] can be proved that the VSTG-problem is also NP-complete.

We examine first following special polynomial cases of the VSTG-problem:

(a) $|Z| = 2$. The VSTG-problem reduces to a generalization of the well-known shortest path problem.

(b) $|Z| = n$. The VSTG-problem reduces to the well-known minimal spanning tree problem.

We study then a generalization of the well-known 1-median-problem and present finally some heuristics for solving the VSTG-problem.

2. Graph representation

The algorithms proposed in this paper use a linked adjacency list representations of graphs (e.g.*Taraszow* and *Richter* [4]). This representation requires at most $O(m+n)$ words of storage.

Fig.1 illustrates the suggested data representation for some digraph Q.

N_Q:

x	1	2	3	4	5	6
$N_Q(x)$	4	1	2	2	1	2

L_Q:

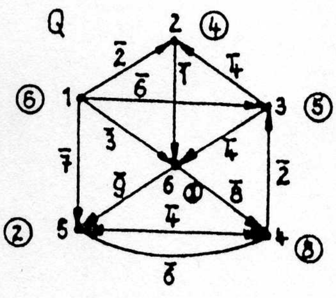
x	1	2	3	4	5	6
$L_Q(x)$	2,3,5,6	6	2,6	3,5	4	4,5

M_Q:

x	1	2	3	4	5	6
$\psi(r)$	2,6,7,3	1	4,4	2,6	4	8,9

K_Q:

x	1	2	3	4	5	6
$\varphi(x)$	6	4	5	8	2	3



i - vertex number

$\textcircled{j}$ - vertex cost

$\bar{k}$ - arc cost

i	1	2	3	4	5	6	
POIN:	1	5	6	8	10	11	13

j	1	2	3	4	5	6	7	8	9	10	11	12
NEI:	2	3	5	6	6	2	6	3	5	4	4	5
CARC:	2	6	7	3	1	4	4	2	6	4	8	9

i	1	2	3	4	5	6
CVERT:	6	4	5	8	2	3

Fig. 1: Example of a digraph Q and its data representation

3. Vertex-weighted optimal paths

We consider now the the special case of the VSTG-problem, where $|Z| = 2$ and therefore the subgraph Q is a path.

Problem (generalized one-pair problem)

GIVEN: An digraph G, the arc and vertex cost functions,and two vertices $x,y \in V(G)$.

FIND: A path $Q \leq G$ from the vertex $x \in V(G)$ to the vertex $y \in V(G)$ such that the total cost of Q is minimal.

In the case $\varphi \equiv 0$ we have the well-known shortest path problem. Polynomial time algorithms for this problem are known (e.g. *Bellman* [5], *Dijkstra* [6] or *Dreyfus* [7]).

The basic algorithms for the determination of shortest paths are algorithms for solving the following problem.

Problem (generalized one-to-all problem)

GIVEN: An digraph G, the arc and the vertex cost functions, and a vertex $x \in V(G)$.

FIND: Optimal paths in G from the vertex $x \in V(G)$ to all other vertices of G.

The our algorithm **Path1**(x) solving the generalized one-to-all shortest path problem is based on the *Dijkstra's* method [6].

Dijkstra procedure

```
begin
    u_1 ← ∅;
    for j ← 2 until n do
        if (1,i) ∈ E(G) then u_j ← c_1j;
        else u_j ← ∞;
        fi;
    od;
    S ← {2,3,...,n};
    while S ≠ ∅ do
        find k ∈ S such that u_k = min {u_j};
                                   j∈S
        if u_k = ∞ then STOP;
        [comment: no paths to the vertices remaining in S]
        else
            begin
                S ← S \ {k};
                for each j ∈ S do
                    u_j  ←      min         {u_j,u_k + c_kj};
                od;        (k,j) ∈ E(G)
            end;
        fi;
    od;
end;
```

where $\square_n := \{1,2,\ldots,n\}$; $V(G) = \square_n$;

$c_{ij} := \psi(i,j) \quad (i,j \in \square_n)$

u_j := length of optimal path from the vertex $1 \in \square_n$ to the vertex $j \in \square_n$.

The algorithm **Path1**(x) is a new modification of the *Dijkstra* procedure and uses a linked adjacency list representation, and the idea of the stack principle (e.g. *Yen* [8]) with the labeling finally processed vertices (e.g. *Dreyfus* [7]).

The algorithm **Path1**(x) has been modified for solving following optimization problems in digraphs:

- determination of an optimal path from a vertex $x \in V(G)$ to a vertex $y \in V(G)$ in a arc and vertex weighted digraph G
  [Algorithm **Path2(x,y)**]
- determination of an optimal path from a vertex $x \in V(G)$ to a vertex subset $Y \subseteq V(G)$ in a arc and vertex weighted digraph G
  [Algorithm **Path3(x,Y)**]
- determination of an optimal path from a vertex subset $X \subseteq V(G)$ to a vertex subset $Y \subseteq V(G)$ in a arc and vertex weighted digraph G
  [Algorithm **Path4(X,Y)**]

The algorithms **Path1(x)** - **Path4(X,Y)** are described in details by *Taraszow* [9], and *Taraszow* and *Richter* [4]. Fig. 2 - 5 present some computational results of these algorithms for a graph delivered by the random graph generator described by *Richter* and *Taraszow* [10]. For this graph n = 360 and m = 597. The vertex and the edge costs are uniformly distributed in the segments [0,200] and [0,10], respectively.

4. Vertex-weighted generalized 1-median problem

The median of a graph is any vertex in the graph that minimizes the sum of the shortest distances from it to all another vertices. We investigate now a vertex-weighted generalized 1-median problem. Let $X \subseteq V(G)$ be the set of admissible median placements and $Y \subseteq V(G)$ the set of all vertices to be connected with the median vertex. In addition to vertex and edge costs (1) and (2),respectively, there is also a cost function χ for admissible median placements:

$$\chi : V(G) \longrightarrow \mathbb{R} \qquad (4)$$

We denote by S(x,Y) a star consisting of optimal paths from its center $x \in X$ to all vertices $y \in Y$ and by J(S(x,Y)) its cost, respectively, i.e.

$$J(S(x,Y)) := \chi(x) + \sum_{y \in Y} C(P_n(x,y)) \qquad (5)$$

where $P_G(x,y)$ is an optimal path between vertices x and y, and $C(P_G(x,y))$ is the cost of this path determined by (3).

Problem (vertex-weighted generalization 1-median problem)

GIVEN: A graph G; vertex and edge costs; median placement cost; subsets $X \subseteq V(G)$ of admissible median placements and $Y \subseteq V(G)$ of terminal vertices.

FIND: A star $S(x_*,Y)$ with $x_* \in X$ such that the total cost of $S(x_*,Y)$ is minimal.

The standard 1-median problem is identical to the case of $X = Y = V(G)$ and $\varphi \equiv \chi \equiv 0$. For results regarding the p-median problem see e.g. Christofides [11].

The algorithm **Star**(X,Y) solving the problem above is presented in Pidgin Algol as follows (in detail see *Taraszow* [9]):

```
Algorithm Star(X,Y)
begin
      [comment: X,Y ⊆ V(G)]
  C ← ∞;
  for each x ∈ X do
      Call Path3(x,Y);
      C1 ← J(S(x,Y));
      if C1 < C then
         C ← C1;
         S ← S(x,Y);
      fi;
  od;
end
```

The computational complexity of the algorithm **Star**(X,Y) is $O(pn^2)$ in the worst case, where $p = |X|$.

A computational example for solving of the vertex-weighted generalized 1-median problem is given in Fig.6.

5. Vertex-weighted Steiner-problem in graphs

We consider now the common case of the VSTG-problem, formulated in the section 1. To solve the VSTG-problem we propose a common greedy scheme that can be mainly characterized by a stepwise extension of a tree and consists of three steps, namely (a) initialization, (b) extension, and (c) termination. Make use of this scheme we can obtain some greedy heuristics solving the VSTG-problem.

Common greedy scheme

Initialization: Fixing some subtree $T_0 \subseteq G$.
$T \leftarrow T$; $R \leftarrow S \setminus V(T)$.

Extension: Find a vertex $x \in R$ such that
$$C(P_G(x,T)) = \min_{y \in R} C(P_G(y,T)).$$
$T \leftarrow T \cup P_G(x,T)$; $R \leftarrow R \setminus V(P_G(x,T))$.

Termination: If $R = \emptyset$ then end.

Different heuristics for solving of the vertex-weighted Steiner problem in graphs can be obtained from this common greedy scheme by the choice of different start subtrees T_0 or different extensions of subtrees. Usual situations correspond to choice such start subtrees as a Z-vertex, the shortest path between two Z-vertices or between two arbitrary vertices, the longest path between two Z-vertices etc. and to choice such extensions as the shortest or the longest path between the vertices of the current Steiner tree and the remaining Z-vertices. Some of these algorithms are described in details by *Taraszow* [9]. The computational complexity of these algorithms is $O(kn^2)$ in the worst case. Two illustrative examples for these heuristics are given in Fig.7 ($T_0 = x \in Z$) and Fig.9 ($T_0 = P_G(x,y)$ where $x,y \in Z$).

References

[1] R.M.Garey and D.S.Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H.Freeman and Company, San Francisco (1979).

[2] A. Segev. The node-weighted Steiner tree problem. Networks 17 (1987) 1-17.

[3] P.Winter. Steiner problem in networks: a Survey. Networks 17 (1987) 129-167.

[4] O.Taraszow and P.Richter. Efficient algorithms for the determination of optimal paths in non-negatively weighted digraphs. In: A.Iwainsky (Eds.) Optimization of Connection Structures in Graphs. CICIP, Berlin (1985) 13-28.

[5] R.E.Bellman. On a routing problem. Q.Appl.Math. 16(1958) 87-90.

[6] E.W.Dijkstra. A note on two problems in connection with graphs. Numer.Math. 1(1959) 269-271.

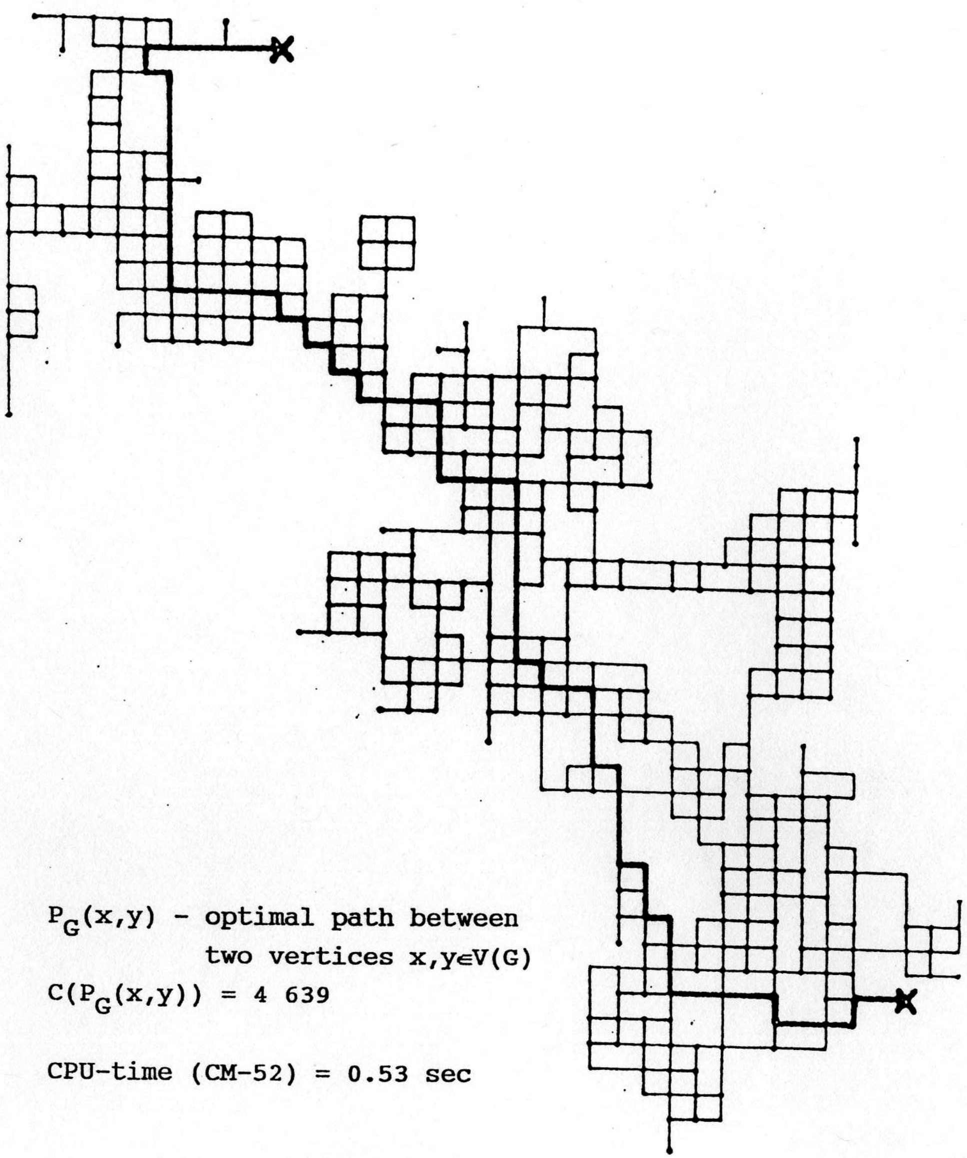
[7] S.E.Dreyfus. An appraisal of some shortest-path algorithms. Oper.Res. 17(1967) 3, 395-412.

[8] J.X.Yen. Shortest Path Networks Problems. (Mathematical Systems in Economics 18) Verlag Anton Hain, Meisenheim am Glan (1975).

[9] O.Taraszow. Entwicklung graphentheoretischer Methoden zur Untersuchung von Systemstrukturen. Diss. B, Berlin (1985).

[10] P.Richter and O.Taraszow. Random graph generation. In: A.Iwainsky (Eds.) Optimization of Connection Structures in Graphs. CICIP, Berlin (1985) 213-225.

[11] N.Christofides. Graph Theory: An Algorithmical Approach. Acad.Press, New York (1975).

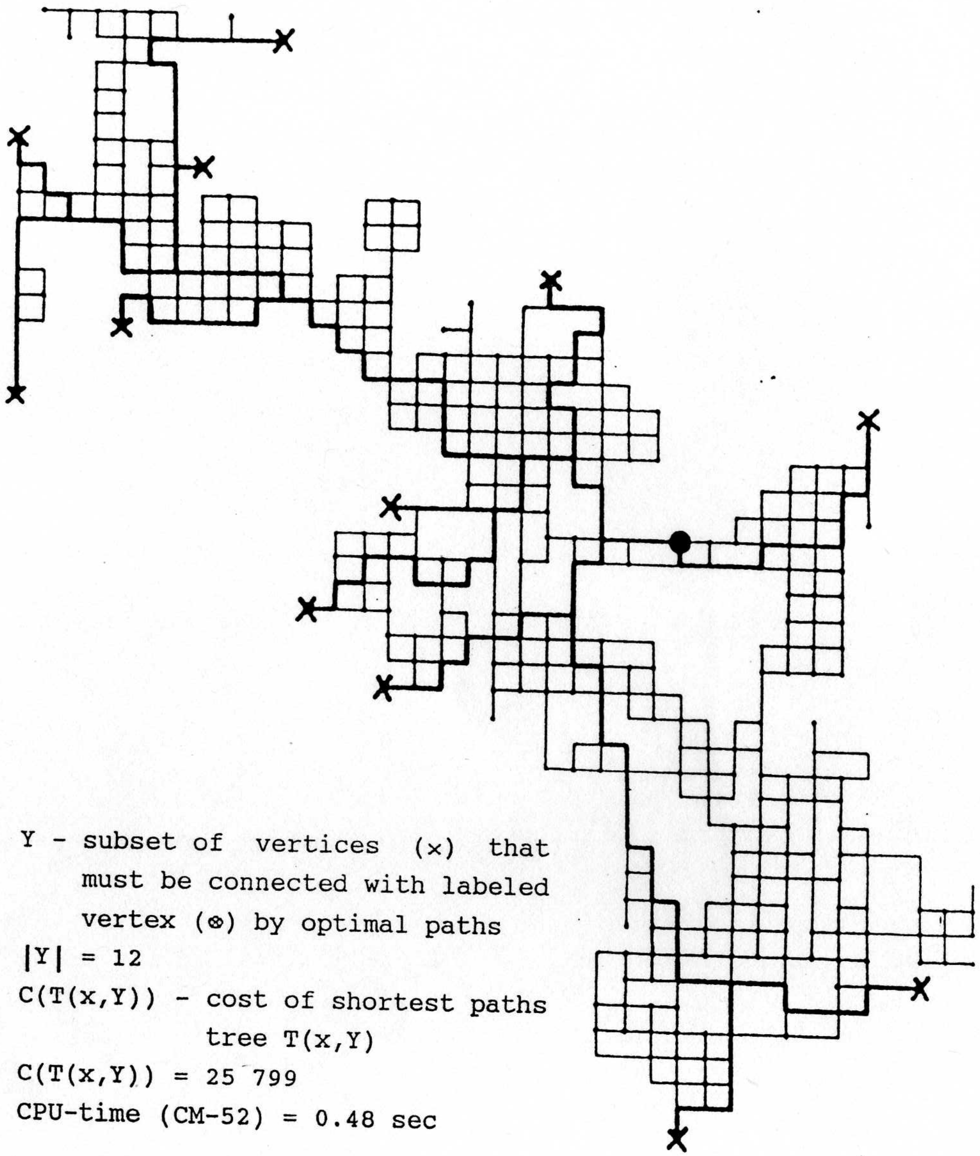


$P_G(x,y)$ - optimal path between two vertices $x,y \in V(G)$

$C(P_G(x,y)) = 4\ 639$

CPU-time (CM-52) = 0.53 sec

Fig. 2: Optimal path between two labeled vertices (×) [Algorithm Path2(x,y)]



Y - subset of vertices (×) that must be connected with labeled vertex (⊗) by optimal paths

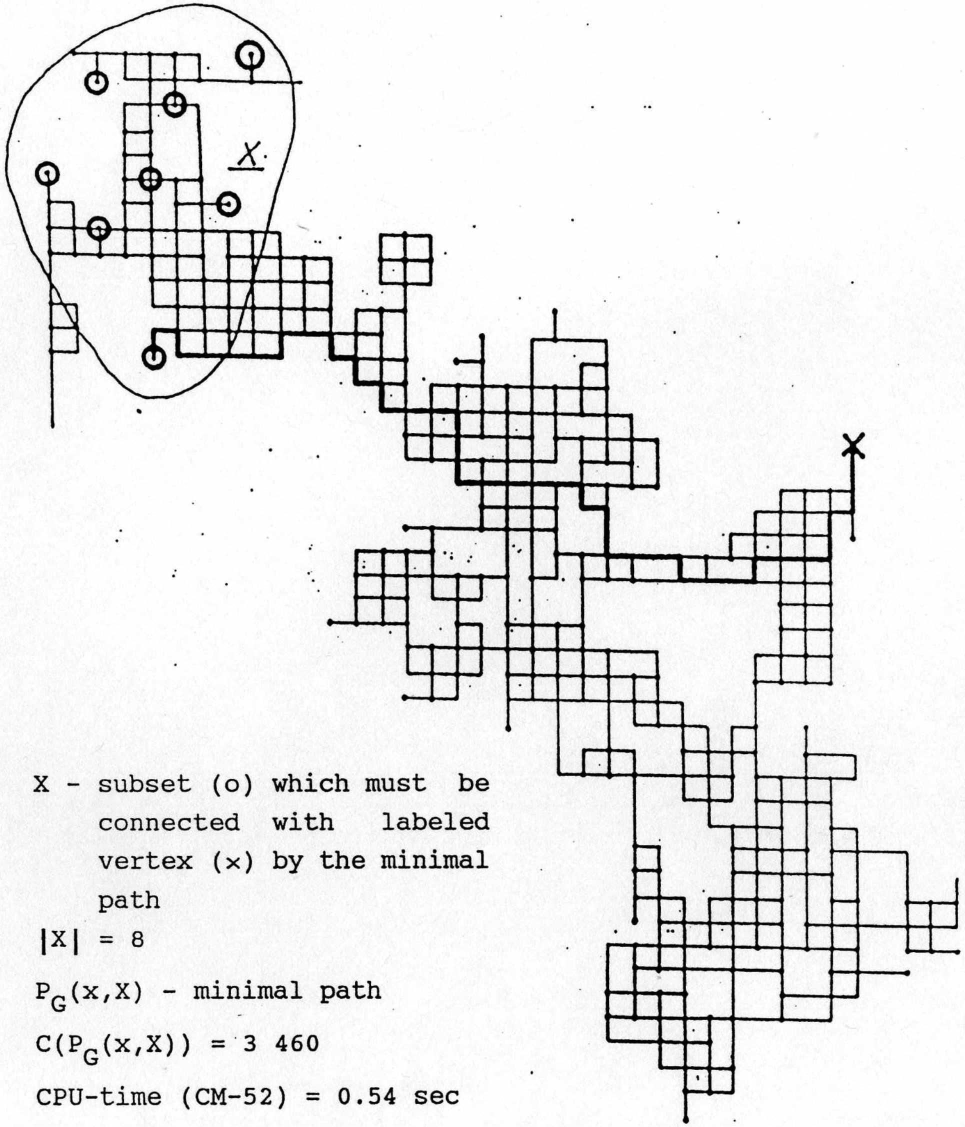
$|Y| = 12$

$C(T(x,Y))$ - cost of shortest paths tree $T(x,Y)$

$C(T(x,Y)) = 25\ 799$

CPU-time (CM-52) = 0.48 sec

Fig. 3: Optimal paths from a source vertex (⊗) to the vertices (×) of a subset $Y \subseteq V(G)$



X - subset (o) which must be connected with labeled vertex (×) by the minimal path

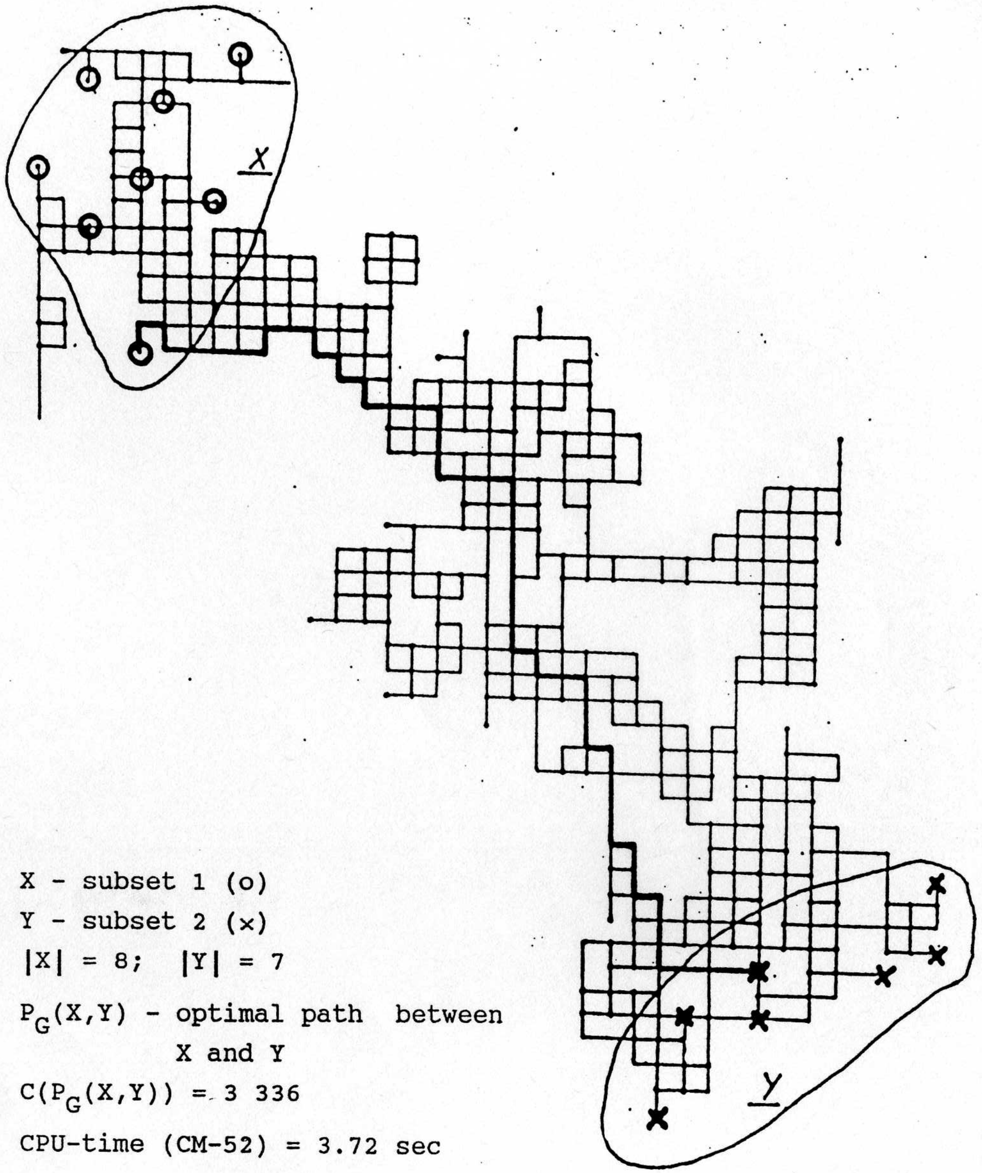
$|X| = 8$

$P_G(x,X)$ - minimal path

$C(P_G(x,X)) = 3\ 460$

CPU-time (CM-52) = 0.54 sec

Fig. 4: Optimal path between a vertex (×) and a subset $X \subseteq V(G)$ (o)
[Algorithm Path3(x,X)]



X - subset 1 (o)

Y - subset 2 (×)

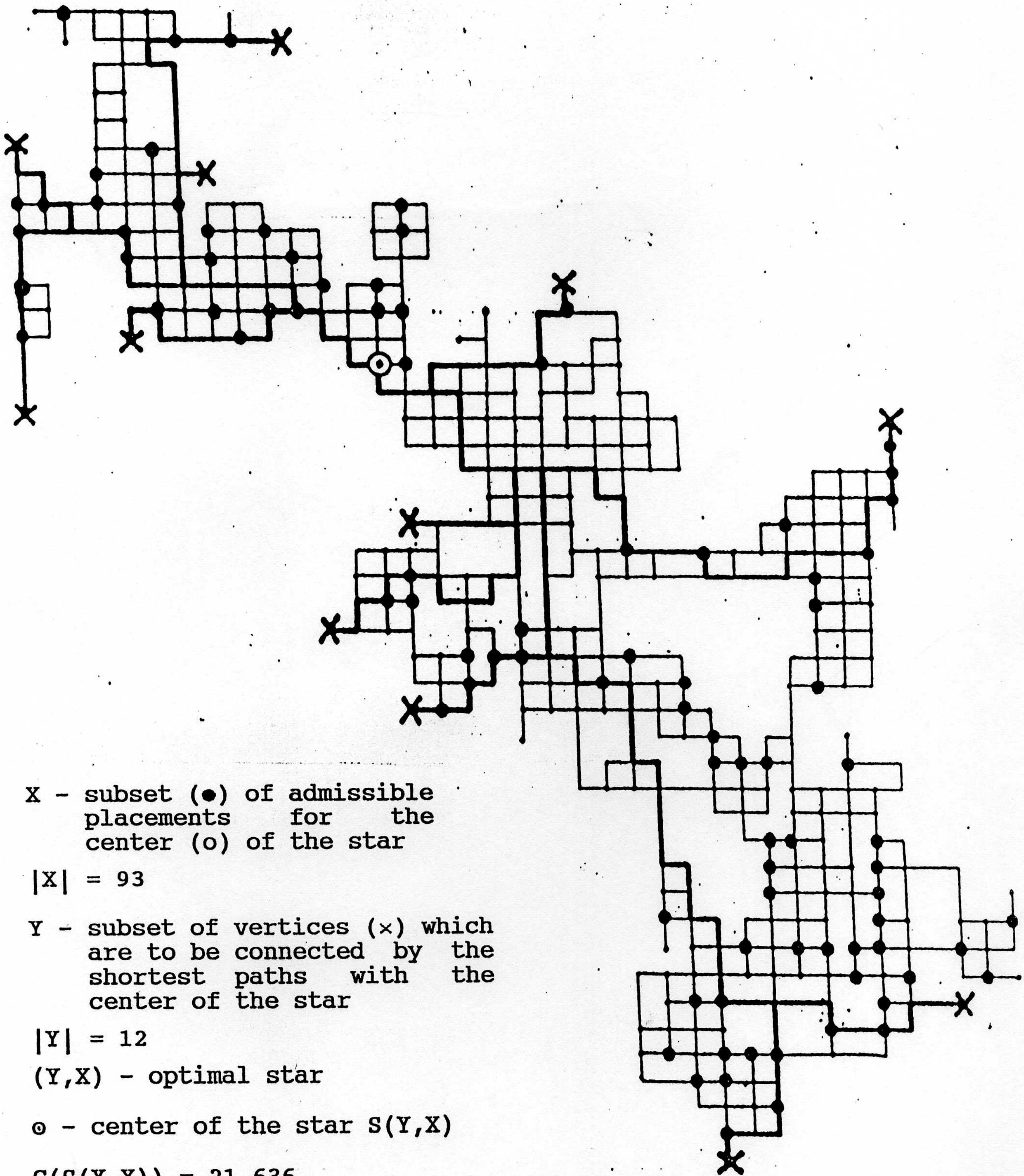
$|X| = 8$; $|Y| = 7$

$P_G(X,Y)$ - optimal path between X and Y

$C(P_G(X,Y)) = 3\ 336$

CPU-time (CM-52) = 3.72 sec

Fig. 5: Optimal path between two subsets $X,Y \subseteq V(G)$
[Algorithm Path4(X,Y)]



X - subset (•) of admissible placements for the center (o) of the star

$|X| = 93$

Y - subset of vertices (×) which are to be connected by the shortest paths with the center of the star

$|Y| = 12$

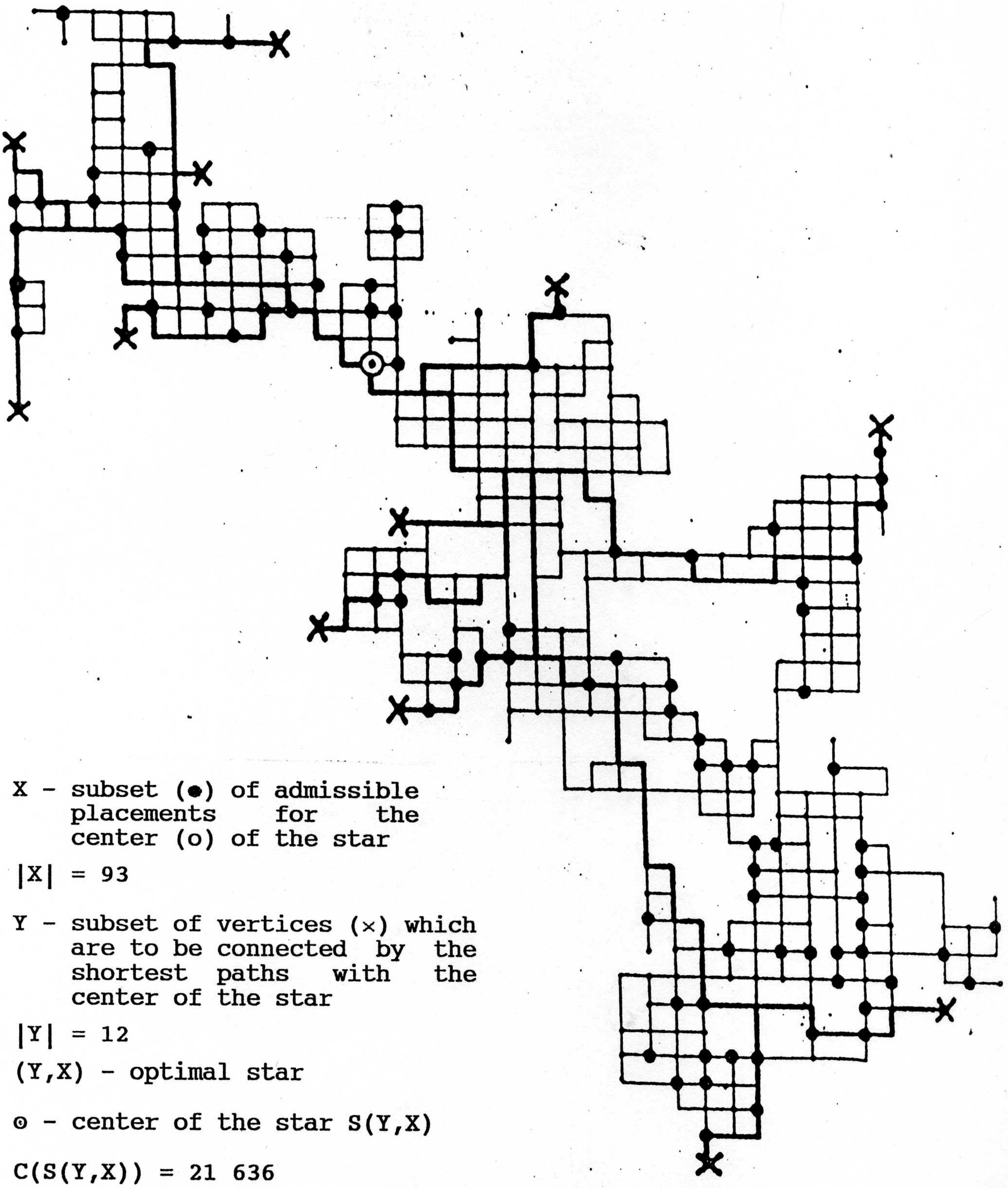
(Y,X) - optimal star

⊙ - center of the star S(Y,X)

C(S(Y,X)) = 21 636

CPU-time (CM-52) = 18.62 sec

Fig. 6: Optimal placement of a star structure [Algorithm Star(Y,X)]



X - subset (•) of admissible placements for the center (o) of the star

|X| = 93

Y - subset of vertices (×) which are to be connected by the shortest paths with the center of the star

|Y| = 12

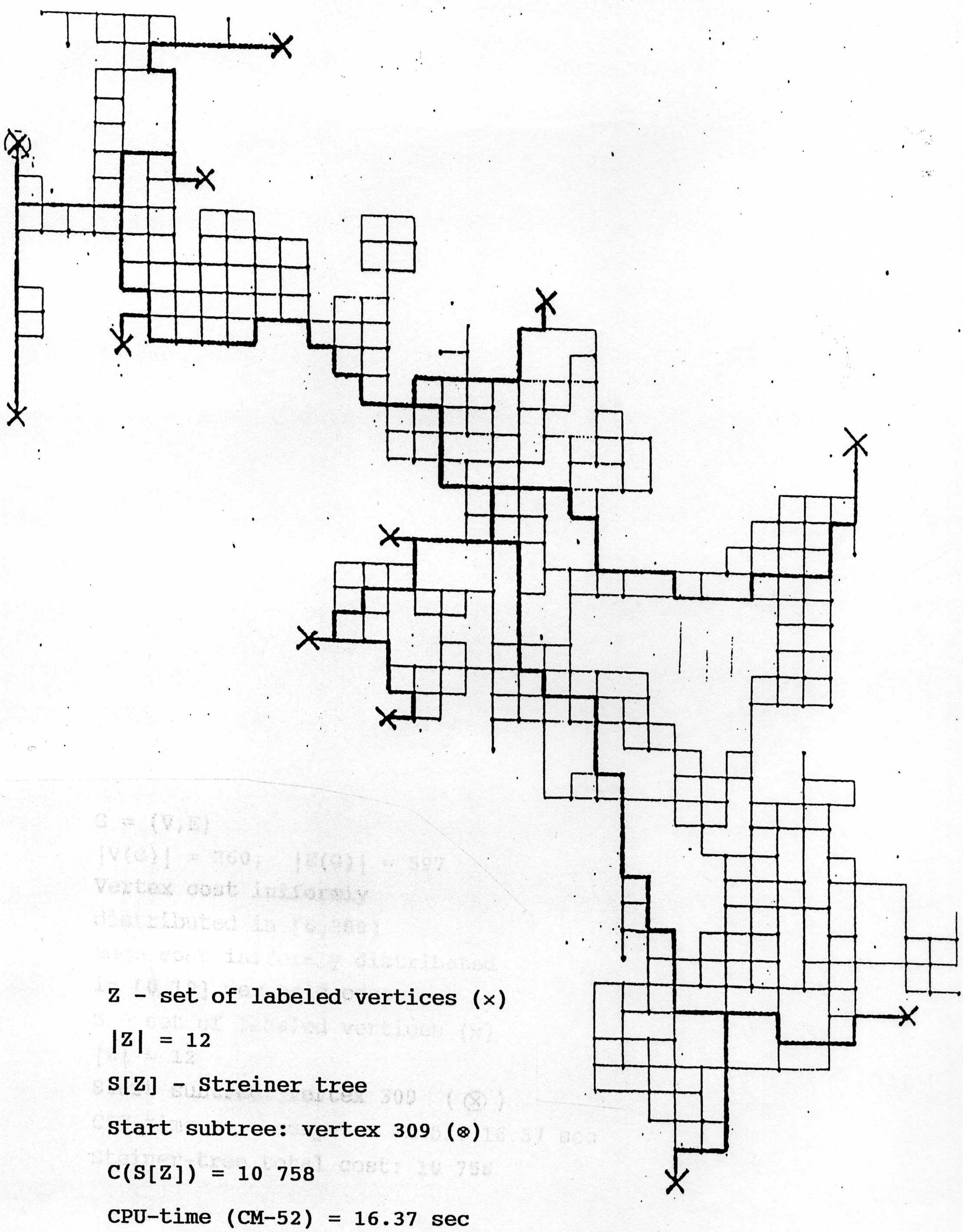
(Y,X) - optimal star

⊙ - center of the star S(Y,X)

C(S(Y,X)) = 21 636

CPU-time (CM-52) = 18.62 sec

Fig. 6: Optimal placement of a star structure [Algorithm Star(Y,X)]



Z - set of labeled vertices (×)

|Z| = 12

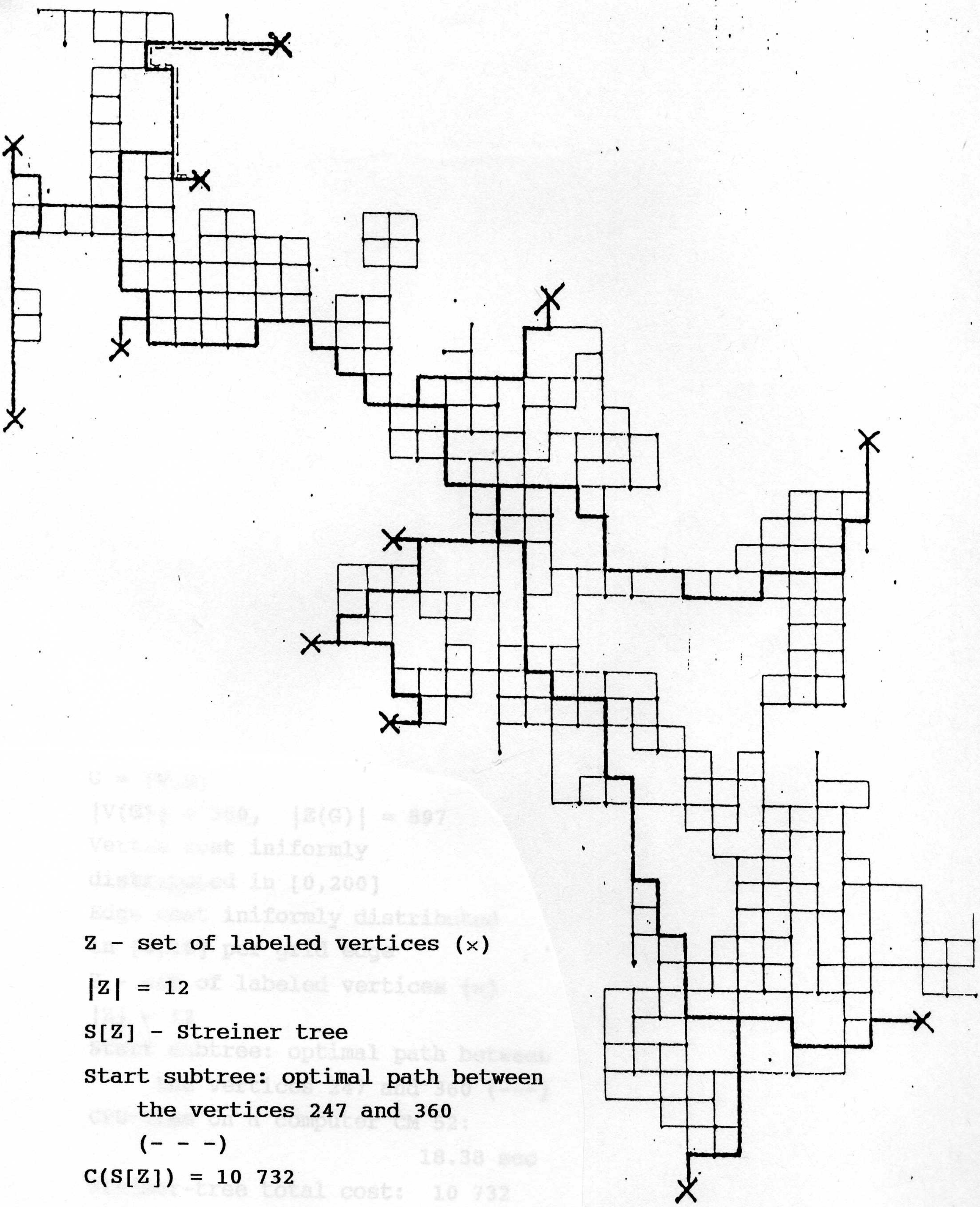
S[Z] - Streiner tree

Start subtree: vertex 309 (⊗)

C(S[Z]) = 10 758

CPU-time (CM-52) = 16.37 sec

Fig.7: Steiner tree [Algorithm STree1(x)]



Z - set of labeled vertices (×)

|Z| = 12

S[Z] - Streiner tree

Start subtree: optimal path between
the vertices 247 and 360
(- - -)

C(S[Z]) = 10 732

CPU-time (CM-52) = 18.38 sec

Fig. 8: Steiner tree [Algorithm STree2(x,y)]