TR-87-010

A STUDY ON TWO GEOMETRIC LOCATION PROBLEMS

D. W. Wang
Depart. of Computer Science & Information Engineering
National Taiwan University

Y. S. Kuo
Institute of Information Science
Academia Sinica

Keywords: computational geometry, geometric location problems, p-center problem, circle intersection graph, maximum independent set, dynammic programming, NP-hard

Address for Correspondence:

Dr. Y. S. Kuo
Institute of Information Science
Academia Sinica
Nankang, Taipei, Taiwan
The Republic of China

1. Introduction

In this paper, we consider the following related geometric location problems:

(A) Given a set S of points and a positive real number d, find a largest subset of S in which the distance between any two points is greater than d.

(B) Given a set S of points and a positive integer p, find a p-point subset of S in which the two closest points are farthest away.

To be formal, let $S = \{p_1, p_2, \ldots, p_n\}$. For Problem A, we wish to maximize the size of a subset T which satisfies

$$\min \{ d(p_a, p_b) \mid p_a \in T,\ p_b \in T,\ p_a \neq p_b \} > d.$$

But for Problem B, we wish to find a subset T of given size p which can maximize

$$\min \{ d(p_a, p_b) \mid p_a \in T,\ p_b \in T,\ p_a \neq p_b \}.$$

Since we shall consider Problems A and B in either the 1-dimensional or the 2-dimensional Euclidean space, there are actually four problems to be considered, which will be referred to as A1, A2, B1 and B2 respectively.

The problems just described are most related to the Euclidean p-center problem [1]. The 1-dimensional Euclidean p-center problem has been shown to have an O(nlog n) upper bound on its time complexity [2]. In contrast, Problem A1 can be solved by sorting in time O(nlog n), and we have derived an elegant dynammic programming algorithm for Problem B1 which runs in time O(pn+nlog n). For the 2-dimensional case, we will prove that both Problems A2 and B2 are NP-hard as the Euclidean p-center problem is [1]. Incidentally, we also prove that the maximum independent set problem for circle intersection graphs is NP-hard.

2. The 1-dimensional case

In this section, we give efficient algorithms for Problems A1 and B1. Problem A1 can be solved by the following greedy algorithm: First sort the set S of points. Then scan the points from left to right and select a point whenever its distance from the last selected point is greater than d. (The leftmost point is selected initially.) It is easy to prove that this algorithm does find a largest subset as required and its run time is O(nlog n).

As for Problem B1, we use a dynammic programming approach. Let us assume that the points of S have been presorted in ascending order, i.e. $p_1 < p_2 < \cdots < p_n$. Let $S_i = \{p_1, p_2, \ldots, p_i\}$ and let

$$d_{ij} = \max_{\substack{T \subseteq S_i \\ |T| = j \\ p_i \in T}} \; \min_{\substack{p_a, p_b \in T \\ p_a \neq p_b}} |p_a - p_b| , \qquad 2 \leq j \leq i \leq n,$$

be an optimum solution for Problem B1 given S_i as the set of points and j as the selection size. (Thus d_{np} is the solution we are seeking for.) Then d_{ij} can be computed by the following recurrence relation:

$$\begin{aligned} d_{i2} &= p_i - p_1 \\ d_{ij} &= \max_{j-1 \leq h < i} \; \min \{ d_{h,j-1}, p_i - p_h \} , \quad 2 < j \leq i \leq n. \end{aligned} \qquad (1)$$

A straightforward implementation of (1) yields an $O(pn^2)$ algorithm. However, we can exploit some structures peculiar to d_{ij} and reduce the run time to O(pn).

Let $d_{ij}^{\ h} = \min\left\{ d_{h,j-1}, p_i - p_h \right\}$, $j \leq i$, $j-1 \leq h < i$.

Lemma 1. There exists an h such that

$$d_{ij}^{\ j-1} \leq d_{ij}^{\ j} \leq \ldots \leq d_{ij}^{\ h} \text{ and } d_{ij}^{\ h} > d_{ij}^{\ h+1} \geq \ldots \geq d_{ij}^{\ i-1}.$$

Proof: Note that $d_{h,j-1}$ is nondecreasing and $p_i - p_h$ is decreasing as h increases. Since $d_{ij}^{\ h}$ is the minimum of $d_{h,j-1}$ and $p_i - p_h$, $d_{ij}^{\ h}$ consists of a nondecreasing and a nonincreasing subsequences as shown in Fig.1(a). (One of the subsequences may not be present if the curves for $d_{h,j-1}$ and $p_i - p_h$ do not intersect.) Q.E.D.

Lemma 2. Let h(i,j) be the h specified in Lemma 1 with respect to i and j. Then $d_{ij} = d_{ij}^{\ h(i,j)}$ and $h(i,j) \leq h(i+1,j)$, $j \leq i$.

Proof: From (1), we have $d_{ij} = \max\limits_{j-1 \leq h < i} d_{ij}^{\ h}$.

Thus $d_{ij} = d_{ij}^{\ h(i,j)}$ due to Lemma 1.

If $h(i+1,j) = i$, then $h(i,j) \leq i-1 < h(i+1,j)$.

Assume $h(i+1,j) < i$. By definition, $d_{i+1,j}^{\ h} = \min\left\{ d_{h,j-1}, p_{i+1} - p_h \right\}$. With reference to Fig. 1(a) and (b), $d_{h,j-1}$ remains the same but $p_i - p_h$ is raised to $p_{i+1} - p_h$. Thus the intersection must shift to the right. In other words, $h(i,j) \leq h(i+1,j)$. Q.E.D.

Lemmas 1 and 2 can lead to the following algorithm for Problem B1:

```
Algorithm B1
    for i = 2 to n do $d_{i2} = p_i - p_1$ endfor
    for j = 3 to p do
        $h(j,j) = j-1$
        $d_{jj} = \min\{d_{j-1,j-1}, p_j - p_{j-1}\}$
        for i = j+1 to n do
            for h = h(i-1,j) to i-1 do      /* determine h(i,j) */
                if $h+1 < i$ and $\min\{d_{h,j-1}, p_i - p_h\} > \min\{d_{h+1,j-1}, p_i - p_{h+1}\}$
                then break
            endfor
            $h(i,j) = h$
            $d_{ij} = \min\{d_{h,j-1}, p_i - p_h\}$
        endfor
    endfor
```

Theorem 1. Algorithm B1 plus a presorting can solve Problem B1 in time $O(pn + n\log n)$.

Proof: Algorithm B1 correctly computes $h(i,j)$ and d_{ij} according to Lemmas 1 and 2. Its run time is $O(pn)$ since at most $O(n)$ statements are executed in the two inner for loops. The term $n\log n$ accounts for the presorting time. Q.E.D.

3. The 2-dimensional case

We shall establish the NP-hardness of Problems A2 and B2 by proving the maximum independent set problem for circle intersection graphs to be NP-complete. The maximum independent set problem is known to be NP-complete for general graphs, but it is solvable in polynomial time for many restricted classes of graphs. [3]

A graph is a circle intersection graph if each of its vertices corresponds to a unit circle in the plane and two vertices are joined by an edge iff their corresponding circles intersect. (This definition is not as general as that appeared in the literature due to the restriction of unit circles.)

Maximum Independent set for Circle Intersection Graphs (MISCIG). Given a circle intersection graph $G = G(V,E)$ and a positive integer p, does G contain an independent set of size p or more, i.e. a subset $W \subseteq V$ such that $|W| \geq p$ and such that no two vertices in W are joined by an edge in E ?

Theorem 2. MISCIG is NP-complete.

Proof: It is obvious that MISCIG is in NP.

Following the technique developed by Megiddo [1], we establish a reduction from 3-satisfiability [3]. Formally, given a boolean expression

$$E = E_1 \wedge E_2 \wedge \ldots \wedge E_m$$

where $E_j = x_j \vee y_j \vee z_j$ ($\{x_j, y_j, z_j\} \subseteq \{u_1, \bar{u}_1, u_2, \bar{u}_2, \ldots, u_q, \bar{u}_q\}$), the 3-satisfiability problem is to decide whether there exists an assignment

$A \subseteq \{u_1,\bar{u}_1,u_2,\bar{u}_2,\ldots,u_q,\bar{u}_q\}$ such that

$$A \cap \{x_j,y_j,z_j\} \neq \emptyset \qquad (j = 1,2,\ldots,m),$$

and $$A \cap \{u_i,\bar{u}_i\} = 1 \qquad (i = 1,2,\ldots,q).$$

In the reduction from 3-satisfiability to MISCIG, each variable u_i $(i = 1,2,\ldots,q)$ will be represented by a "circuit" of vertices in the plane. The clauses E_j $(j = 1,2,\ldots m)$ are represented by "clause configurations" which determine how the different circuits meet each other. Circuits must cross each other, without interfering with each other's properties; this requires that we design the "junctions". A schematic view of the circuits and their relations to the clause configurations is shown in Fig. 2. Note that the constructed graph will be a circle intersection graph laid out in the plane, each vertex of the graph is positioned at the center of its corresponding unit circle.

We now describe the 3 components in details. Associated with each variable u_i is a circuit of vertices $C_i = \{v_0^{\,i},v_1^{\,i},\ldots,v_{r_i}^{\,i}\}$, where $v_0^{\,i} = v_{r_i}^{\,i}$, r_i is even and $v_k^{\,i}$ and $v_h^{\,i}$ are joined by an edge iff $|k - h| = 1 \pmod{r_i}$. As shown in Fig. 3, a circuit C_i consists of a vertical stem and some horizontal branches, each branch corresponds to a clause that u_i or $\bar{u}_i$ appears. It can be observed that r_i is bounded by qm to a constant factor, and the coordinates of the vertices in all the circuits can be computed in polynomial time. In a circuit, a vertex $v_k^{\,i}$ is called an even vertex or an odd vertex depending on whether k is even or odd. There are essentially two different ways to choose these vertices into a maximum independent set, namely, either all even vertices or all odd vertices. The former case will correspond to the assignment of "true" to u_i and the latter case correspond to the assignment of "false" to u_i.

In the reduction, each clause $E_j = x_j \vee y_j \vee z_j$ is represented by a configuration of three vertices $v_x^{\,j}$, $v_y^{\,j}$ and $v_z^{\,j}$ as shown in Fig. 4. The vertex $v_x^{\,j}$, for example, is adjacent to a vertex $v_f^{\,i}$ in the circuit C_i for variable u_i where i is such that $x_j \in \{u_i, \bar{u}_i\}$. Moreover, $v_f^{\,i}$ is an even vertex if $x_j = \bar{u}_i$ and an odd vertex if $x_j = u_i$. Thus, if E_j is satisfied by an assignment A containing x_j, then $v_f^{\,i}$ is not in the independent set and $v_x^{\,j}$ can be included in the independent set. In fact, this clause configuration has the property that if not all of the vertices $v_f^{\,i}$, $v_g^{\,j}$ and $v_h^{\,k}$ are in an independent set, then one and only one vertex among $v_x^{\,j}$, $v_y^{\,j}$ and $v_z^{\,j}$ can be included in the independent set.

As we can see in Fig. 2, the vertical stem of a circuit may intersect a horizontal branch of another circuit, and each intersection produces 4 cross points. To avoid such interference, we introduce junctions, one junction for each cross point. As shown in Fig. 5, two circuits C_i and C_j cross each other and a junction consisting of 4 new vertices is created to join vertices $v_k^{\,i}$ and $v_{k+1}^{\,i}$ and vertices $v_h^{\,j}$ and $v_{h+1}^{\,j}$. We insist that both k and h be even numbers. This ensures that the segments of circuits between consecutive junctions have even number of vertices. Furthermore, such a junction component has the following property: If both $v_k^{\,i}$ and $v_{k+1}^{\,i}$ or both $v_h^{\,j}$ and $v_{h+1}^{\,j}$ are in an independent set, then none of the 4 vertices in the junction can be included in the independent set. Otherwise, one and only one vertex in the junction can be included in the independent set. In other words, if in each circuit only even vertices or only odd vertices are in the independent set (consistent assignment of truth value to each variable), then an additional vertex for each junction can be included in the independent set. Let us denote the number of junctions by J.

Letting

$$p = \sum_{i=1}^{q} r_i/2 + m + J,$$

we claim that E is satisfiable iff the constructed graph has an independent set W of size p or more. Before we give the proof, note that the constructed graph is a circle intersection graph since each of the components shown in Fig. 3, 4 and 5 is a circle intersection graph laid out in the plane. Also from the construction of these components, we know that the graph can have an independent set of size at most p.

Assume that E is satisfied by a truth assignment A. For $i = 1,2,\ldots,q$, if A contains u_i, then include in W the even vertices of the circuit for u_i; if A contains u_i, then include in W the odd vertices of the circuit for u_i. We can add an additional vertex in W for each clause and junction as we have argued. Thus W is an independent set of size p.

To prove the converse, let W be an independent set of size p. Since each junction and clause can contribute at most one vertex in W, each circuit C_i, $i = 1,2,\ldots,q$, must have $r_i/2$ vertices in W. This ensures that each segment of C_i between consecutive junctions has either the even vertices or the odd vertices in W. Suppose that the segments of C_i have both even vertices and odd vertices in W. Then there exists a junction joining two vertices of C_i both of which are in W. Such a junction can not contribute a vertex in W. Thus W can not have p vertices which is a contradiction. We have shown that each circuit C_i has either the even vertices or the odd vertices in W. Let us include u_i or $\bar{u}_i$ in an assignment A depending on whether C_i has the even vertices or the odd vertices in W. E is satisfied by this assignment since each clause configuration has contributed a vertex in W.

Q.E.D.

It is easy to prove that each of Problems A2 and B2 is polynomially reducible to the other and MISCIG is polynomially reducible to Problem A2. Thus we have the following corollary.

Corollary 1. Both Problems A2 and B2 are NP-hard.

4. Concluding remarks

Both Problems A1 and B1 have a lower bound of Ω(nlog n) for the ε-closeness problem [4] can be transformed to these problems in linear time. Thus, sorting for Problem A1 is optimal. The dynammic programming algorithm for Problem B1, however, has the run time O(pn+nlog n). Further research should be undertaken to bring close the upper and lower bounds.

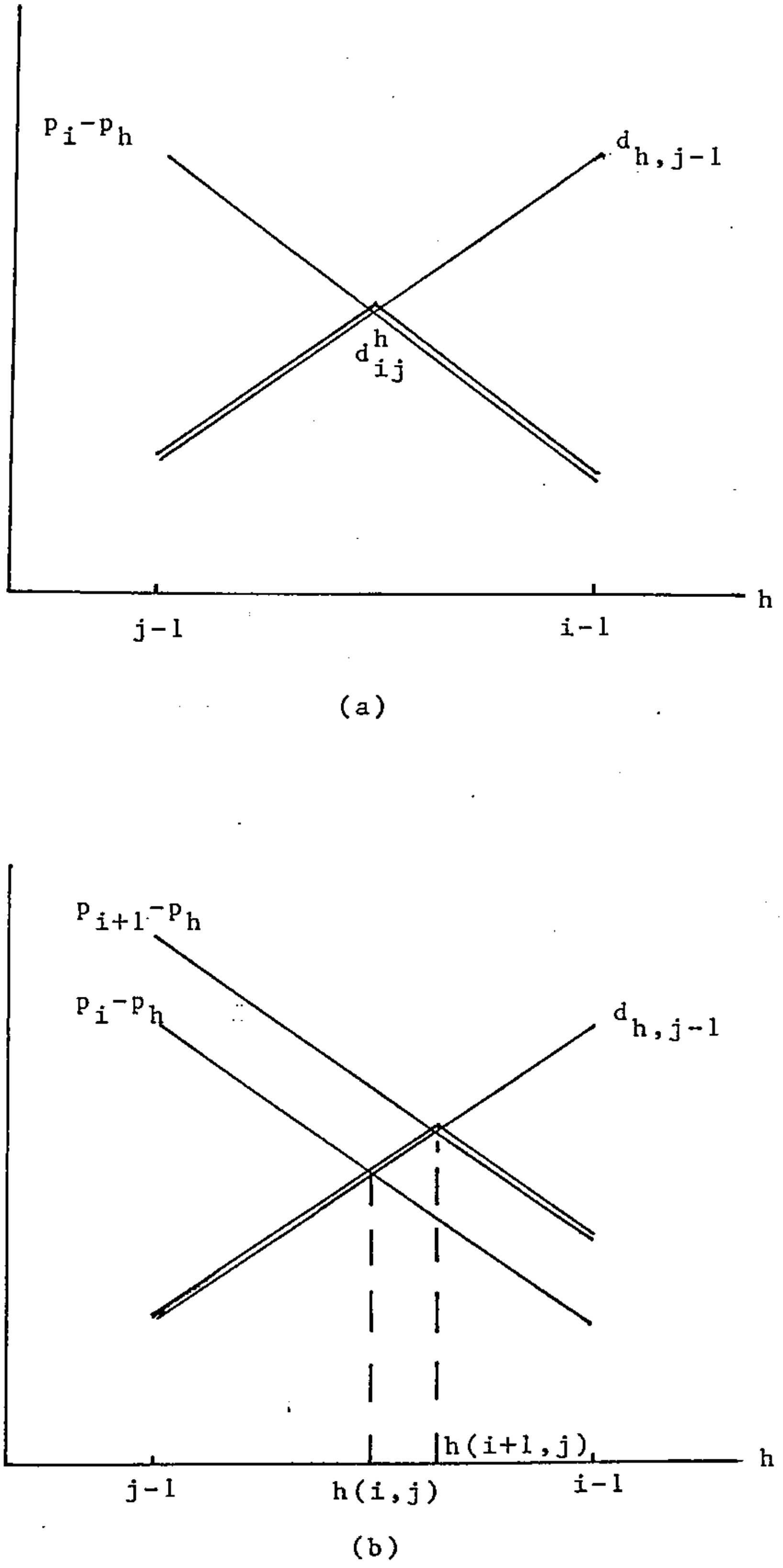
We can consider Problems A2 and B2 relative to the rectilinear distance $d((x_1,y_1),(x_2,y_2)) = |x_1-x_2| + |y_1+y_2|$. These two problems can be proven to be NP-hard in exactly the same way as their Euclidean counterparts.

Acknowledgment

We thank Dr. D. T. Lee for suggesting these problems and giving some valuable guidance.

References

1. N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," SIAM J. Comput., Vol. 13, No. 1 (1984) 182-196.
2. M. I. Shamos, "Geometry and Statistics: Problems at the Interface," in Algorithms and Complexity: New Directions and Recent Results, J. F. Traub, ed., Academic Press, New York (1976) 251-280.
3. M. R. Garey and D. S. Johnson, Computers and Intractability, a Guide to the Theory of NP-completeness, W. H. Freeman and Company, San Francisco (1979) 194-195.
4. F. P. Preparata and M. I. Shamos, Computational Geometry, Springer-Verlag, New York (1985) 325-326.



(a)

(b)

Fig. 1. Illustration for d^h_{ij}

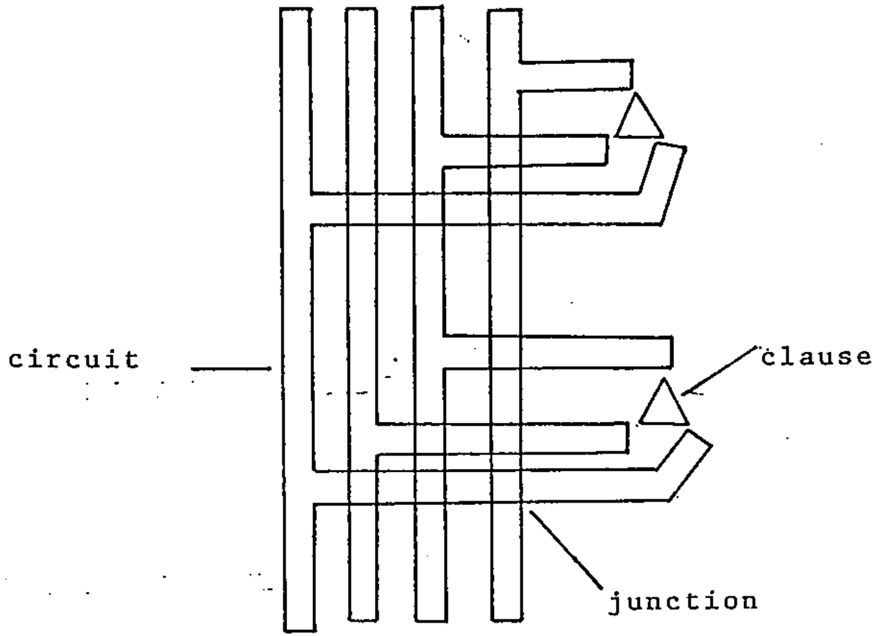
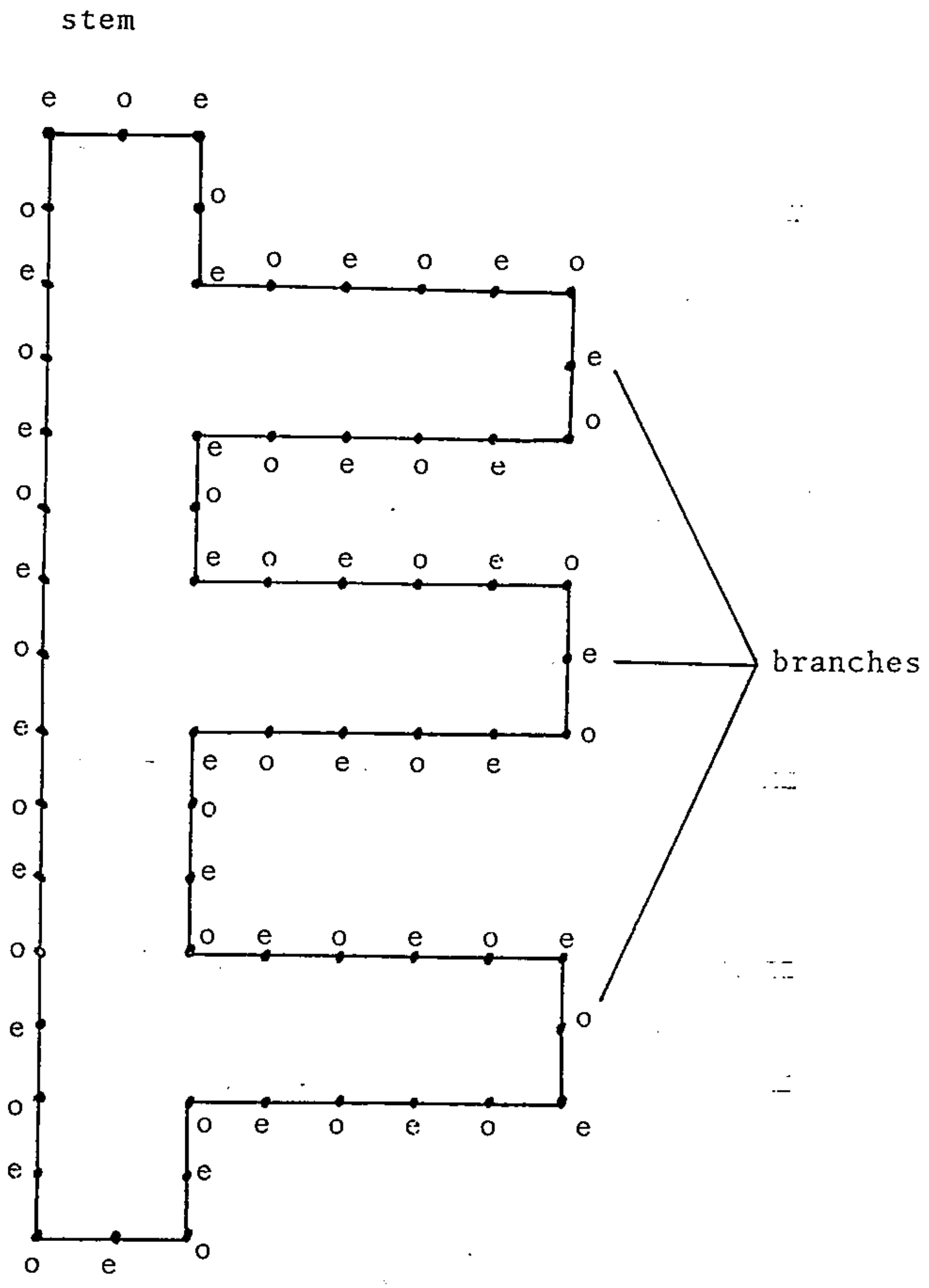


Fig. 2. A schematic view of the reduction



e:even o:odd

Fig. 3. A circuit of vertices

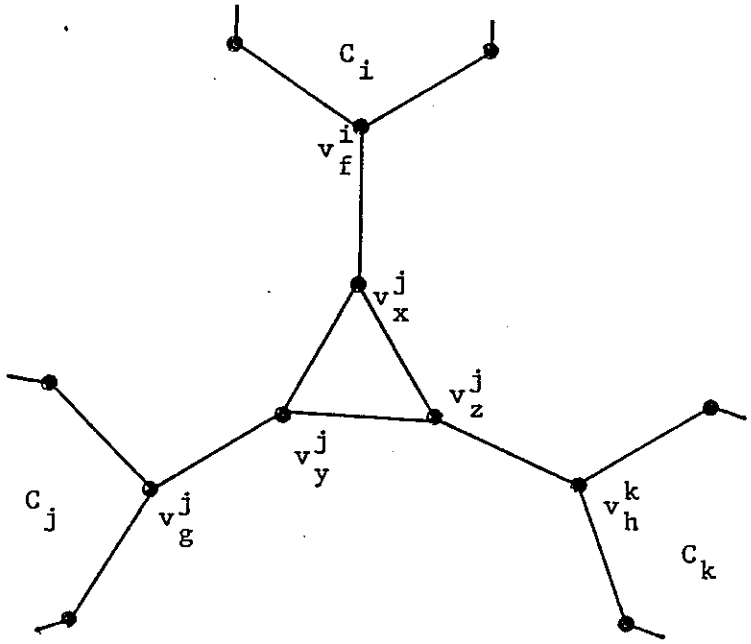


Fig. 4. A clause configuration

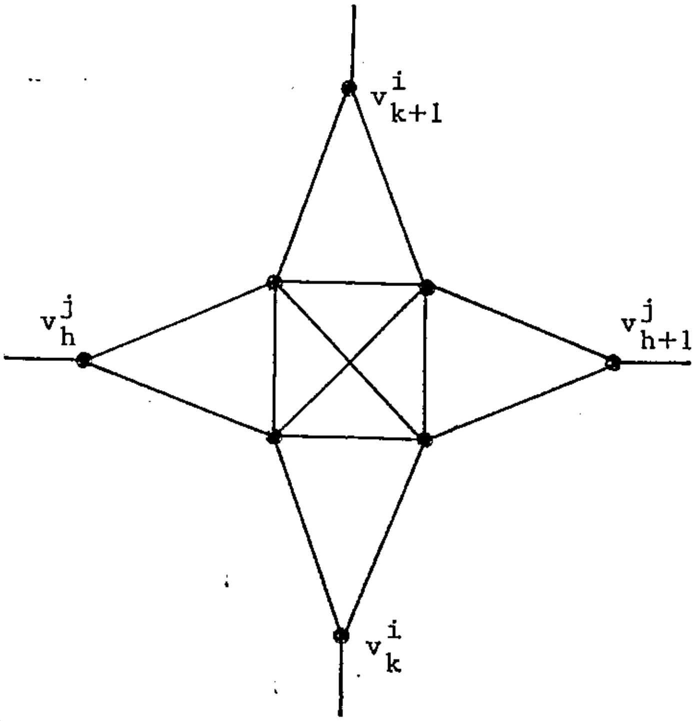


Fig. 5. A junction