Directional p-Median in Two Dimensions is NP-Complete

Laura E. Jackson   Matthias F. Stallmann

March 16, 2004

Abstract

An instance of a (two-dimensional) p-median problem gives the locations of n demand points on the plane. The goal is to find p supply points that minimize the total distance of the demand points to their nearest supply point. Megiddo and Supowit [9] have shown that rectilinear and Euclidean versions of the p-median problem are NP-hard when p is part of the input. We show that the rectilinear version continues to be NP-hard under the restriction that the nearest supply point for a given demand point must lie above and to the right of it. This restriction has applications to providing more than one dimension of quality-of-service in, for example, computer networks — a customer is usually unwilling to accept less than the desired QoS along any dimension, but wants to pay as little as possible for what is offered.

1 Introduction

The continuous p-median problem in d-dimensional space, $d \geq 2$, is: given n demand points, find p supply points that minimize the sum of the distance from each demand point to its closest supply point, with respect to a particular distance metric. The continuous nature of the problem allows us to choose supply points anywhere in d-space, not merely from among the given demand points. When either the Euclidean or the rectilinear distance measure is used, this problem is NP-hard [9]. Under the rectilinear distance measure, it is well known that only "intersection points" need be considered as candidate sites for supply points. Thus this problem reduces to a discrete p-median problem (see [1] for a complete treatment of discrete location problems).

We define here the *directional* rectilinear distance measure and prove the continuous p-median problem in $d \geq 2$ dimensions is NP-hard under this new metric. In general, a c-directional, d-dimensional rectilinear metric (with $c \leq d$) defines distance from point $(p_1, \ldots, p_d)$ to $(q_1, \ldots, q_d)$ to be ∞ if $p_i > q_i$ for some $i \in \{1, \ldots, c\}$ and $\sum_{1 \leq i \leq d} |q_i - p_i|$ otherwise. Thus, in a directional p-median problem, a supply point must achieve or exceed the values of the first c coordinates of all its demand points.

The directional p-median problem arises when attempting to proportion a network to satisfy the various Quality of Service (QoS) demands of many customers, where a particular level of QoS is described by $d \geq 2$ parameters. A customer's d-tuple of coordinates defines its location in the demand space. For example, a service provider may characterize its service using the two parameters average transmission rate ρ and maximum burst size σ. Each customer requests a level of service described by an ordered pair (ρ_i, σ_i). The collection of thousands of service requests is a scatter plot in the (ρ, σ)-plane. Rather than meeting each customer's demand with a unique level of service (represented by a unique point), the service provider may prefer to group similar requests into a single service level, in such a way that any given customer receives *at least* the amount requested. That is, a given demand point in the plane will be mapped to a supply point that is located above and to the right of it, corresponding to a c-directional, d-dimensional distance metric with $c = d = 2$. The directional *rectilinear* metric is appropriate whenever cost of service is a linear combination of costs along each dimension.

In $d = 1$ dimension, the (non-directional) p-median problem can be optimally solved in time $O(pn)$ [4]. (On the real line, the rectilinear and Euclidean distance measures are identical.) The directional p-median problem with $c = d = 1$ can be solved within the same bound by restating the problem as a shortest path problem [7]. We show that the rectilinear c-directional, d-dimensional p-median problem is NP-hard when $c = d = 2$, which implies NP-hardness for all c, d satisfying $2 \leq c \leq d$.

The rest of this report is organized as follows. We begin in Section 2 with a precise statement of the decision version of the problem for $c = d = 2$ (DPM2). We show that DPM2 is NP-complete, which implies NP-completeness for all c, d satisfying $2 \leq c \leq d$. Sections 3 and 4 describe the two main components in the reduction from planar 3-SAT, circuits and clause configurations. We then present a sketch of the proof in Section 5. Section 6 gives the details of the proof.

2 Directional p-median in two dimensions

Our main result is that the (decision version of the) directional, rectilinear p-median problem is NP-complete in two dimensions. In the precise statement of the problem which follows, let $d((x_i, y_i), (x_j, y_j))$ be the 2-directional rectilinear distance from point (x_i, y_i) to point (x_j, y_j). That is,

$$d((x_i, y_i), (x_j, y_j)) = \begin{cases} x_j - x_i + y_j - y_i & \text{if } x_j \geq x_i \text{ and } y_j \geq y_i, \\ \infty & \text{otherwise} \end{cases}$$

Problem*(DPM2) Given a set $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ of points in the plane, an integer p, and a bound B, does there exist a set*

$$S = \{(z_1, t_1), (z_2, t_2), \ldots, (z_p, t_p)\}$$

of p points such that

$$\sum_{i=1}^{n} \min_{1 \leq j \leq p} \{d((x_i, y_i), (z_j, t_j))\} \leq B \text{ ?}$$

Our reduction is from planar 3-SAT, the non-polar version (see [10]), first proved NP-complete by Lichtenstein [8]. An instance of planar 3-SAT consists of a conjunctive normal form (CNF) formula with three literals per clause, each literal being either a variable or its negation. In addition, the graph representation of this formula must be planar. To build

Table 1: Summary of the reduction from planar 3-SAT to DPM2.

Planar 3-SAT	$\rightarrow$	DPM2
variable u_i	$\rightarrow$	circuit C^i of points $P_1^i, P_2^i, \ldots, P_{r_i}^i$
clause E_j	$\rightarrow$	clause point P_j^*
if u_i in E_j	$\rightarrow$	P_j^* is located near circuit C^i in a true configuration
if $\overline{u_i}$ in E_j	$\rightarrow$	P_j^* is located near circuit C^i in a false configuration

the graph representation, we create a vertex for each variable and a vertex for each clause. We add edges to connect variable v's vertex to the vertex of any clause in which v or its negation appears.

Fig. 1(a) shows the planar graph for the formula $E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5$, where $E_1 = (u_1 \vee \overline{u_2} \vee u_3)$, $E_2 = (\overline{u_1} \vee u_2 \vee \overline{u_3})$, $E_3 = (u_1 \vee u_3 \vee \overline{u_4})$, $E_4 = (\overline{u_3} \vee u_4 \vee u_5)$, and $E_5 = (\overline{u_1} \vee u_3 \vee \overline{u_5})$ (and also for $2^{15} - 1$ other formulae that have the same variables in the same clauses and differ only in whether or not a particular occurrence is negated). Our proof requires an **orthogonal** embedding of the formula graph as shown in Fig. 1(b). Such an embedding can be constructed in linear time and with $O(n^2)$ total area [11] (see also [2]). Even though the orthogonal drawing assumes a graph whose maximum degree is four, a simple transformation can ensure this requirement without affecting the remainder of the argument: clause vertices have degree three by design and edges from a variable vertex to $k > 4$ clauses can be replaced by a tree with k leaves and interior nodes of degree ≤ 3. The nodes p and q in Fig. 1(b) are interior nodes of the tree for u_3.

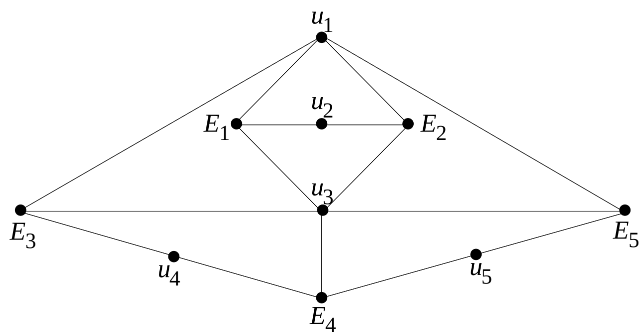
We also require that (a) no two clause vertices have the same x- or y-coordinate, and (b) no clause vertex has edges incident on both its west (smaller x-coordinates) and south (smaller y-coordinates) sides. Both requirements are easy to meet by shifting clause vertices of an existing drawing horizontally and/or vertically while extending edges in the appropriate direction (since clause vertices always have degree 3, conflicts can be resolved by rotating edges clockwise 90 degrees until the unused direction is reached).

The demand points for the directional p-median instance are organized into n rectilinear polygons called **circuits**, one for each variable, and m additional points, one per clause. Fig. 1(c) illustrates the schematic based on the orthogonal embedding of Fig. 1(b) (analogous to the schematic used by Megiddo and Supowit [9]). Table 1 gives an overview of the elements of the reduction.

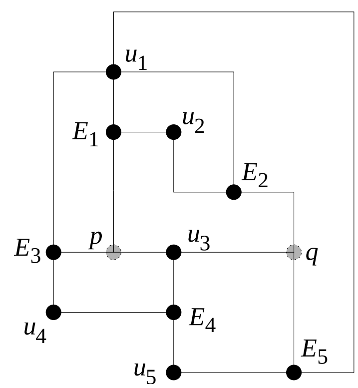
3 Construction of the circuit for each variable

A 3-SAT variable u_i is represented by a circuit C of r points arranged in a rectilinear polygon. The number r is divisible by 3 and circuits are designed in such a way that an optimal set of $r/3$ supply points partitions the demand points in one of two ways, to represent true and false values, respectively. In the circuit in Figure 2, subsets in the true (respectively, false) partition are enclosed with solid (respectively, dashed) lines.

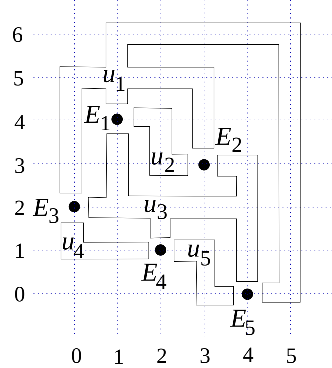
A *cluster* S is any subset of demand points that share a supply point. The optimal supply point for S is the point $(x_{\max}, y_{\max})$, where $x_{\max}$ and $y_{\max}$ are the maximum x- and



(a) The graph of a planar 3-SAT instance.



(b) An orthogonal embedding of the graph.



(c) The schematic for the p-median instance.

Figure 1: Transformation of the planar 3-SAT instance $E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5$, where $E_1 = (u_1 \vee \overline{u_2} \vee u_3)$, $E_2 = (\overline{u_1} \vee u_2 \vee \overline{u_3})$, $E_3 = (u_1 \vee u_3 \vee \overline{u_4})$, $E_4 = (\overline{u_3} \vee u_4 \vee u_5)$, and $E_5 = (\overline{u_1} \vee u_3 \vee \overline{u_5})$.

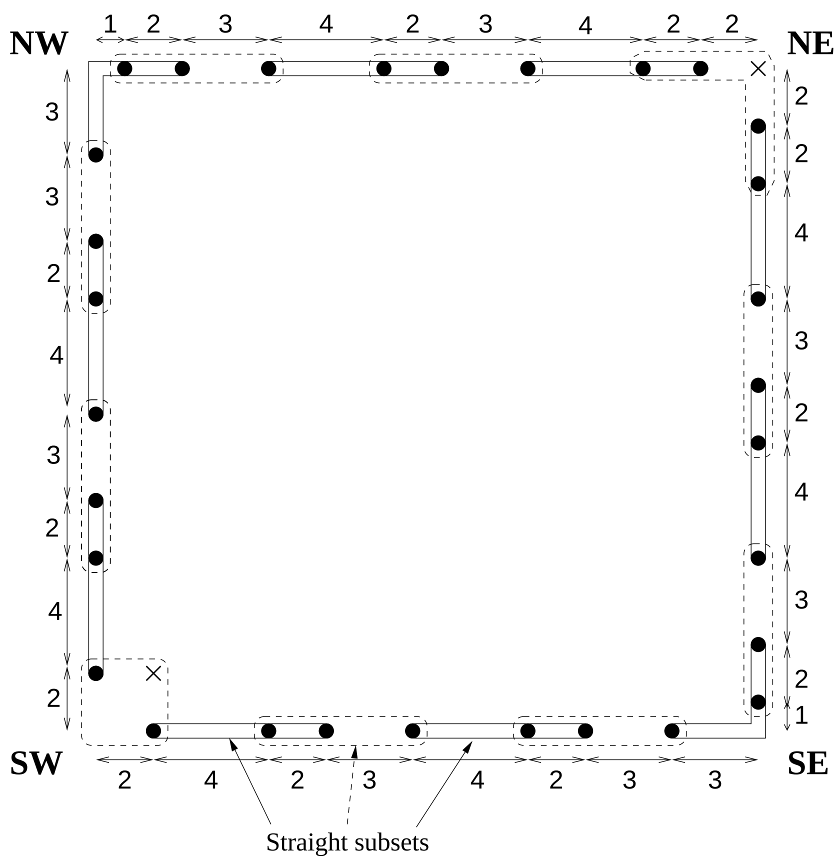
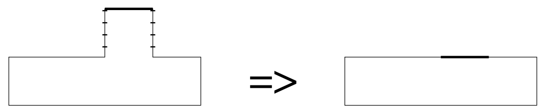
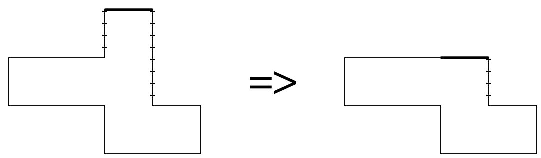


Figure 2: A simple circuit with four corners, one of each type. Solid lines indicate a true partition; the dashed lines indicate a false partition. The two ×'s mark supply points that do not coincide with demand points.



(a) Facing sides have equal length.



(b) Facing sides have unequal length.

Figure 3: The two cases of Lemma 1.

y-coordinates, respectively, in S. The points of a circuit are partitioned into clusters in two ways.

In the *true partition* every cluster is made up of three points whose cost is 8 (the demand points are at distances 0, 2, and 6). The *false partition* has three-point clusters with cost 8 (demand points at distances 0, 3, and 5) along the straight segments, and clusters with two and four points, respectively, in the SW and NE corners. The latter have supply points that are not demand points (they are indicated by $\times$ in Figure 2) and have costs of 4 and 12, respectively.

Although the circuit in Figure 2 has only four corners, this will never be the case in our construction. For example, the circuit for u_4 in Figure 1(c) has six corners; in clockwise order beginning above the u_4 label they face NW, NE, SW, NE, SE, and SW. The one for u_3 has 18 corners — from above the u_3 label clockwise they face SE, NE, SW, NW, NE, SE, SW, NE, NW, SE, SW, NE, SW, NW, SE, NW, NE, and SW.

It is important for the construction that the number of NE and SW corners be the same. This ensures that every cluster with cost 12 in a false partition is balanced out by one of cost 4.

Lemma 1 *In any circuit, the number of NE corners and SW corners is the same.*

Proof. (by induction on the number of circuit sides) This is obviously true for a circuit with 4 sides. Suppose it is true for all circuits (orthogonal polygons) with k or fewer sides. Consider a circuit with $k+1$ sides and reduce it to one with fewer sides by eliminating an "inward-facing" side and at least one of the two sides incident with it. There are two cases to consider.

Case 1. The two corners of the inward-facing side are equidistant from the remaining corners of the other two sides (see Fig. 3(a)). We can eliminate both incident sides and by doing so we remove four corners, one of each type. By induction the $(k-2)$-sided circuit that results has the same number of NE and SW corners, and so does the $(k+1)$-sided circuit.

Case 2. The two corners of the inward-facing side are different distances away from the other two corners (see Fig. 3(b)). We can move the inward-facing side until it merges with the closer of the two nearby sides. The effect is to eliminate two sides and two corners. The missing corners are either a NE and a SW corner or a NW and a SE corner (as in the figure) so adding them to the $(k-1)$-sided circuit does not change the relationship between the number of NE and SW corners. □

For technical reasons that will be evident later, circuits are constructed according to the following rules:

1. The number of points in a circuit is divisible by 3 (already noted).
2. Points in corners of a circuit are arranged as in Fig. 2, depending on the direction the corner faces (one of NW, NE, SW, or SE).
3. Points are situated according to the following inter-point spacing rules:
   (a) The first point north or east from a SW corner is at distance 2; subsequent inter-point distances are 4,2,3,4,2,3, *etc.*, until a point at distance 3 from a NW or SE corner is reached.
   (b) The first point south or west from a NE corner is at distance 2; subsequent inter-point distances are 2,4,3,2,4,3, *etc.*, until a point at distance 1 from a NW or SE corner is reached.
4. A side of a circuit must include at least one straight cluster each of the true and the false partition.
5. The distance from a demand point to another demand point that is not one of its two closest neighbors is at least 9.

Three types of circuit sides arise from the construction. An *inward* side is one that connects two corners in one of the sets $\{NW, NE\}$, $\{NE, SE\}$, $\{SE, SW\}$, or $\{SW, NW\}$ and must have length congruent with 5 modulo 9 with a minimum of 14. (the circuit in Fig. 2 has sides of length 23, the next larger possible size). A *downward* side connects a NE corner with a SW corner and has length 1 mod 9, with a minimum of 10 (a downward side with length 19 is shown in Fig. 4(a) — it could also be horizontal, with a SW corner to the west of a NE corner). An *upward* side (connecting a SE and NW corner) has length 0 mod 9, a minimum of 9 (Fig. 4(b) shows one with length 18; again, a horizontal orientation is possible).

Since each type of circuit size can only account for one mod-9 value in its length and we need more flexibility later to align variable circuits with clause points, extra length-adjusting elements may be needed in our circuits. Fig. 5 shows a gadget called a *kink* that increases length of a side by 2 mod 9 and, with repeated insertions along a side of a circuit, can achieve any length mod 9.

The corner configurations of the kink are similar to those of ordinary circuit corners. The first two are 3×3 squares in the true partition with a cost of 8. The next is a 4×4 square in the false partition identical to a normal northeast corner of a circuit. Its cost of 12 is balanced by the next cost-4 2×3 rectangle. The illustrated kink works only on a north

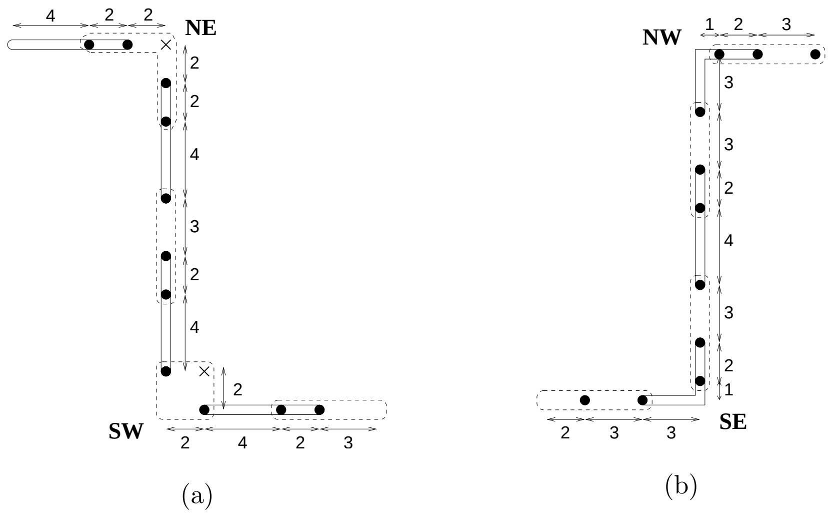


Figure 4: Downward and upward circuit sides.

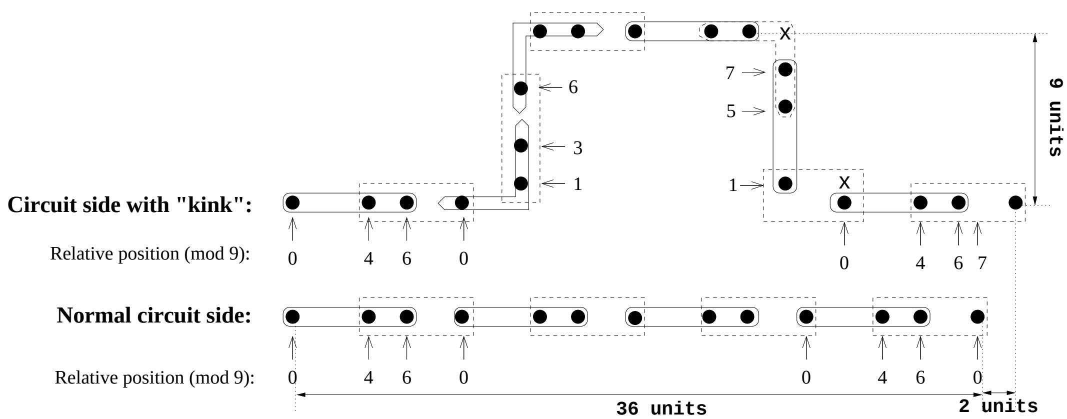


Figure 5: A *kink* that can be used to adjust the length of a side mod 9.

side of a circuit, but a reflection with respect to a horizontal line followed by a 90-degree rotation leads to a kink that has the same effect on an east side.

Within a length of $38 \cdot 8 = 284$ units and a width of 23, we can maneuver the corner that ends a side into any mod 9 position. We can achieve this end easily if we scale the length of any side that does not abut a clause point by a factor of at least $38 \cdot 10 = 380$ (an extra two segments at the two ends ensure that the kinks of two adjacent sides do not come too close to each other) and make the width of any circuit at least 23.

For the example illustrated in Fig. 1(c), we later show what happens if the schematic is scaled up a factor of 900 (this is overkill, but easy to calculate) The example turns into a roughly 4500×5400 unit rectangle.

4 Alignment of clause points with variable circuits

For each clause $E = \ell_1 \vee \ell_2 \vee \ell_3$, we add a demand point P located where the circuits corresponding to the variables of ℓ_1, ℓ_2, and ℓ_3 come together. If $\ell_j = u$, for some variable u, then point P is positioned in one of the four true clause configurations $\mathsf{T_N}$, $\mathsf{T_E}$, $\mathsf{T_S}$, or $\mathsf{T_W}$ with u's circuit C, as shown in Fig. 6. If $\ell_j = \overline{u}$, then point P is positioned in one of the three false clause configurations $\mathsf{F_{NE}}$, $\mathsf{F_S}$, or $\mathsf{F_W}$ with u's circuit C, as Fig. 7 illustrates. It is now clear why we could not have a clause point with circuits coming from both the west and the south (i.e., a clause point cannot be to the east of one circuit and to the north of another).

Figure 8 shows that the circuits of the three variables of a clause will never come too close to each other (the critical distance is 9) regardless of which variables are true or false.

Like the proof in [9], we consider the directional p-median problem on some circuit C with $p = r/3$. We may rephrase the problem as: Partition C into p clusters $C_1, C_2, \ldots, C_p$ so as to minimize $\sum_{j=1}^{p} \text{cost}(S_j)$. For convenience, we call this sum the *cost of the partition* $C_1, \ldots, C_p$. For the true partition, the cost is equal to $8r/3$ because the cost of each cluster is 8. For the false partition, the cost is also $8r/3$ because the cost of each cluster on a straight segment is 8 and, as a consequence of Lemma 1, the number of NE clusters, with 4 points at cost = 12, and SW clusters, with 2 points at cost = 4, is the same.

Each true (respectively, false) configuration is carefully constructed so that the point P can join a cluster of the true (respectively, false) partition and add 12 to the cost for that cluster. Joining any other cluster would result in a higher (non-optimal) cost for the circuit and clause point combined. Therefore, if the truth values are chosen so that E is satisfied, then there is at least one cluster that P may join such that the circuit (with clause point added) will have a new cost that is 12 more than its previous cost.

Suppose we have constructed a DPM2 instance from a 3-SAT instance with r total demand points in the variable circuits and m additional points for m 3-SAT clauses. The decision problem will ask whether or not there exist $p = r/3$ supply points with total cost (*i.e.* sum of costs of all circuits) at most $8r/3 + 12m$.

The use of 12 as the critical distance to add for a clause point is not an arbitrary choice. Any value less than 12 makes it impossible to determine a clause point location to the north or east of a circuit whose variable is negated in the clause (see Fig. 7) without getting too close to the circuit. Fig. 9 illustrates that a value of 13 or more allows too much freedom. In particular, it allows enough freedom so that the partition of the circuit into either the true or the false clusters can be violated.

5 Sketch of proof

First we need a polynomial algorithm to reduce an arbitrary 3-SAT instance to DPM2. The tricky part of this is adding kinks in the right places to ensure that all circuits line up properly with the clause points. A simple algorithm puts all clause points at coordinates that are multiples of 900 and then moves around each circuit in clockwise order, adding kinks as needed. Kinks are added only to the two segments immediately before the one facing a clause point. We ensured that there would be at least two segments between any two clause points by not allowing two clause points to share either coordinate. Full details

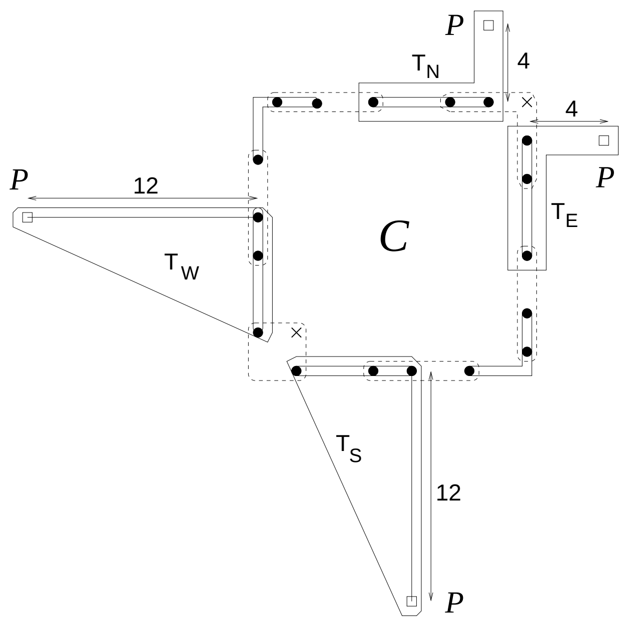


Figure 6: Possible locations of clause points for the true partition.

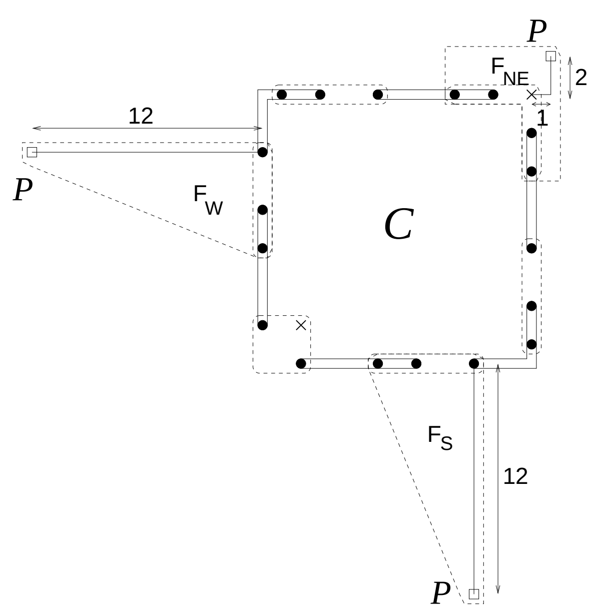


Figure 7: Possible locations of clause points for the false partition.

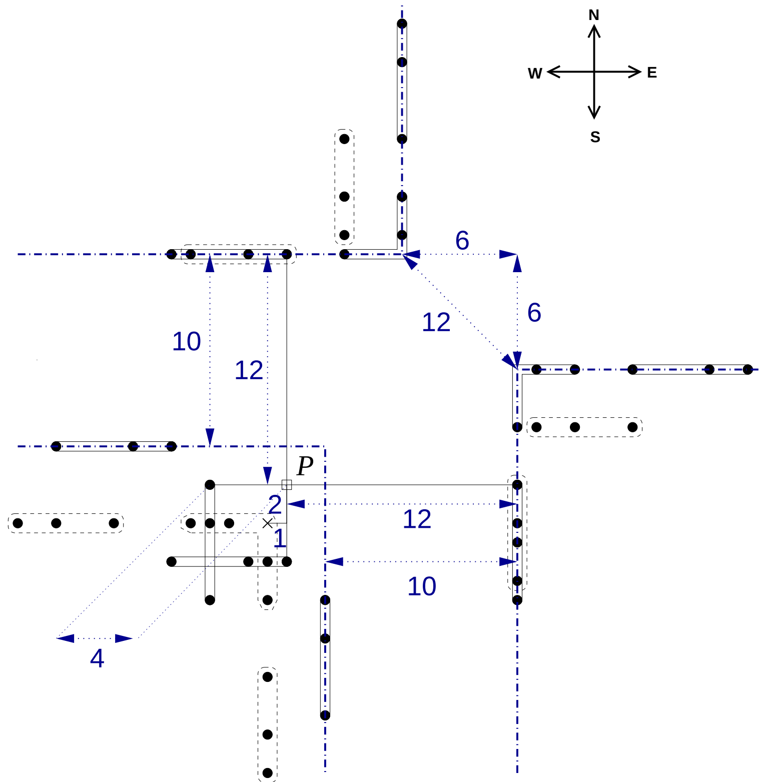


Figure 8: All possible interactions among circuits abutting a clause point.

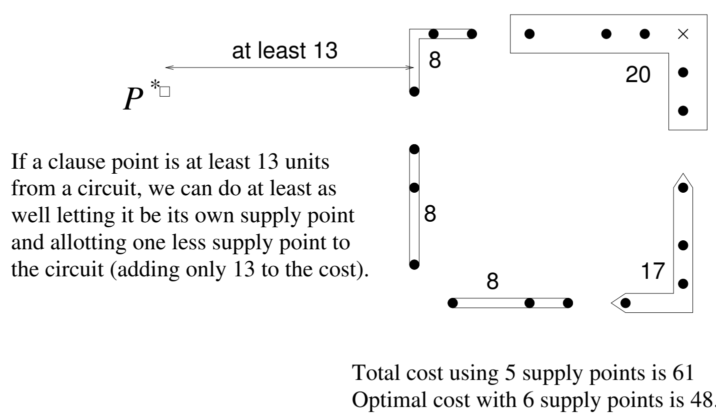


Figure 9: When a clause point adds too much to the cost of a cluster.

are in Section 6.

It is not hard to see that a satisfiable formula implies the existence of p supply points with total cost $8r/3 + 12m$. First, cluster the points in each variable circuit using either the true partition or the false partition, depending on whether the satisfying assignment makes that variable true or false. With no clause points, this leads to a solution with $r/3$ supply points and a cost of $8r/3$. Second, because each clause has at least one true literal, each clause point has a circuit with which it aligns so that it can join a cluster and add no more than 12 to the total cost. The total cost will therefore be $8r/3 + 12m$ as desired.

Now we need to ensure that any choice of $r/3$ supply points having cost $\leq 8r/3 + 12m$ corresponds to a satisfying solution for the 3-SAT formula. This will be argued in two stages. First, we observe that any partition of the demand points can only include *legitimate* clusters, ones that are part of either a true partition, a false partition, a T cluster (see Figure 6) on a circuit with a true partition, or an F cluster (see Figure 7) on a circuit with a false partition. Then the satisfying assignment can be derived from the true and false partitioning of the circuits for the variables; that it is a satisfying assignment follows from the fact that each clause point can only be part of a cluster belonging to a circuit with the appropriate partition.

The first stage of this second proof direction requires careful analysis of several cases (full details are given in Section 6). The reader should become convinced that any cluster that is not legitimate will incur a penalty in cost. This assertion is obvious for clusters that have legitimate ones as proper subsets (adding a new point to a cluster, unless it is the supply point, will increase its cost). Less obvious is the analysis of clusters that are proper subsets of legitimate ones or that overlap with legitimate ones. Using an argument similar to that of Megiddo and Supowit [9] we look at *marginal costs* of increasing the number of demand points associated with each type of possible supply point. Obviously a smaller-than-legitimate cluster in one place forces a larger one elsewhere. As long as the marginal cost of going from a smaller to a legitimate cluster is less than that of going from a legitimate to larger cluster, any optimal solution consists of legitimate clusters. Due to the subtleties of the construction, there are some complications in the details, as discussed below.

6 Proof details

We begin by reviewing the algorithm that constructs an instance of DPM2 from an arbitrary planar 3-SAT instance F.

1. Let G be the planar graph corresponding to F, modified so that no vertex has degree larger than 4 (e.g., repeatedly replace two rotationally-contiguous edges incident to a vertex by a three-edge tree).

2. Construct an orthogonal embedding of G with the following properties: (a) no two clause vertices share either their x- or y-coordinate; and (b) no clause vertex has edges incident to it on both its south and west sides. Both of these are easy to arrange.

3. Turn each variable vertex and its incident edges (including all tree edges that arose from them in step 1) into a circuit as described in Section 3; details are —

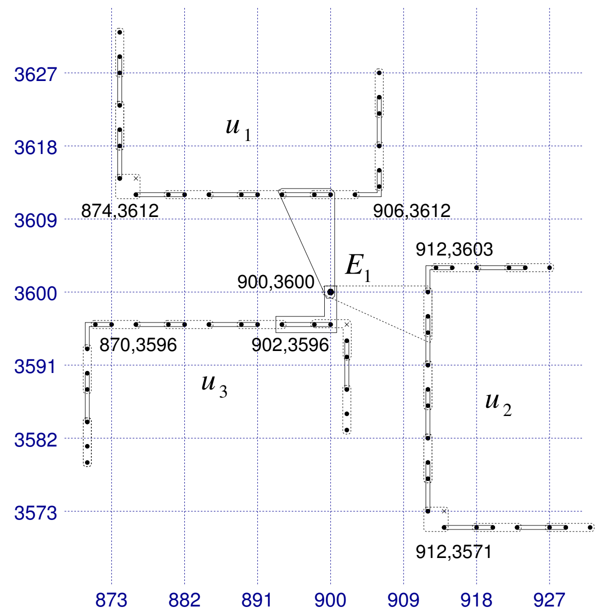


Figure 10: The clause point for $E_1 = (u_1 \vee \overline{u}_2 \vee u_3)$ and its neighboring circuits.

(a) Multiply coordinates of the embedding in step 2 by a large number M (as described earlier, 900 is sufficient) and put clause points in their correct positions with respect to the new coordinate system.

(b) For each edge incident to clause vertex i, lay out a length-32 circuit side orthogonal to the edge and with its interior away from i. Align it with the clause point using the appropriate true or false configuration in Fig. 6 or 7. The circuit sides of edges next to the unused side of i (south or west) should be shifted as far as possible in the unused direction. The third, middle, circuit side can be centered (or close to it). This step is illustrated for the example of Fig. 1 by Figs. 10–14. Coordinates of the corners of the circuits are given, as well as the coordinates of each clause point.

(c) For each variable circuit, pick an arbitrary clause and create the circuit starting at the length-32 side next to it, and continuing in clockwise order. Add the right number of kinks to the two sides preceding each subsequent length-32 side so that the sides line up properly. Fig. 15 shows the result for the example, assuming that each circuit is begun at the lowest-numbered clause. Only the mod 9 values of the circuit-corner coordinates are shown.

4. Set the number of supply points to be $r/3$ where r is the total number of demand points in all the circuits.

5. Set the target cost to be $8r/3 + 12m$, where m is the number of clauses.

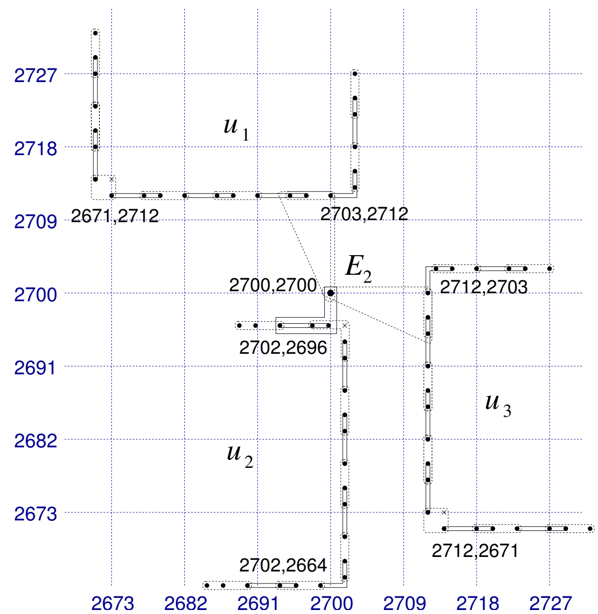


Figure 11: The clause point for $E_2 = (\overline{u}_1 \vee u_2 \vee \overline{u}_3)$ and its neighboring circuits.

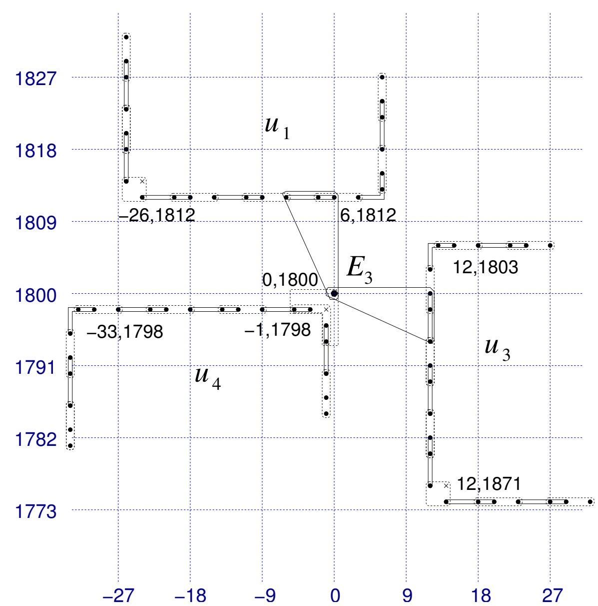


Figure 12: The clause point for $E_3 = (u_1 \vee u_3 \vee \overline{u}_4)$ and its neighboring circuits.

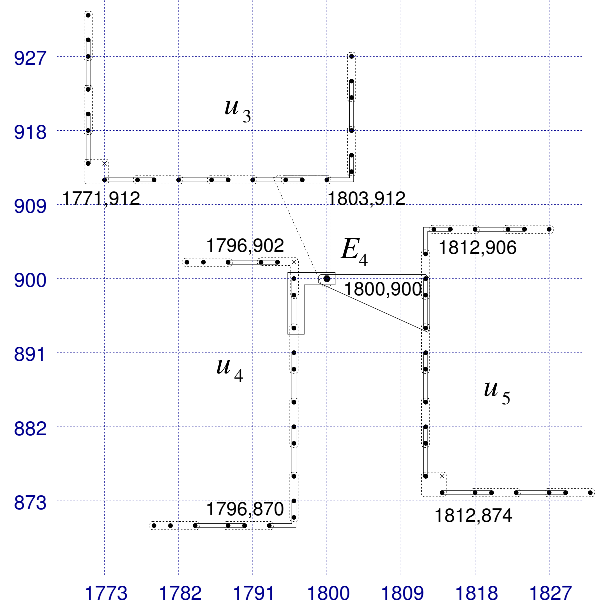


Figure 13: The clause point for $E_4 = (\overline{u}_3 \vee u_4 \vee u_5)$ and its neighboring circuits.

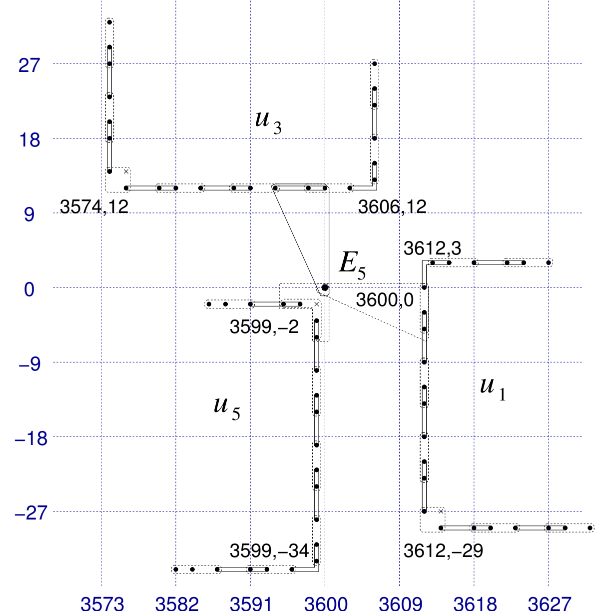
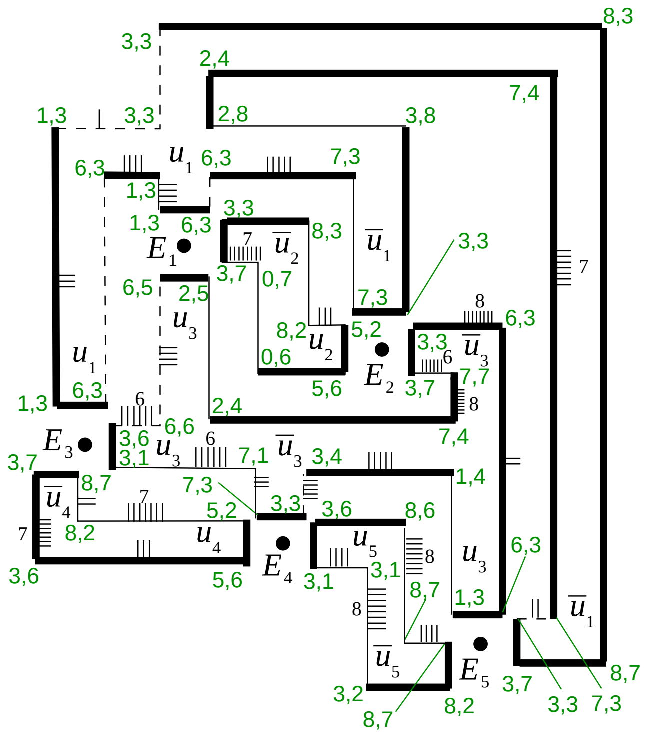


Figure 14: The clause point for $E_5 = (\overline{u}_1 \vee u_3 \vee \overline{u}_5)$ and its neighboring circuits.



This schematic shows the coordinates mod 9 of every circuit corner for the example in Figure 1. Every clause point has coordinates divisible by 9. Using this fact and the clause point location information in Figs. 6 and 7, we compute the relative coordinates of the circuit corners at every clause point. A circuit side abutting a clause has length 32 to allow room for kinks inside a the rectilinear polygon.

The heavy lines are inward sides (length is 5 mod 9), lighter lines are downward (length is 1 mod 9), and dashed lines are upward (length is divisible by 9).

The hash marks on circuit sides indicate kinks. These are calculated as we follow each circuit in clockwise order from an arbitrary clause point (we use the lowest number in this example). The two circuit sides immediately preceding the one that abuts the next clause point are adjusted to correctly match the nearest corner.

Consider, for example, the traversal of u_1's circuit from E_2 to E_5. Starting with the corner marked $3,3$ to the northeast of E_2, we traverse corners marked $3,8$ then $2,8$ then $2,4$ then $7,4$ simply using the standard length of each side type. The next corner must match the one northeast of E_5 in its y-coordinate (and its x-coordinate must be 7). Normally, this side has length 5 mod 9, yielding a y-coordinate of 8 mod 9 instead of 4 at its upper end. Since each kink gives an increase of 2, we must use kinks to increase from 8 to 22, a total of 14, or 7 kinks. The next side normally has length divisible by 9 and must be increased to 4 mod 9 by adding 2 kinks.

Figure 15: A more detailed schematic of the example showing where kinks are used to adjust circuit corners modulo 9.

Table 2: The marginal cost of adding demand points to the cluster of various types of supply points.

$k =$	2	3	4	5	6
type					
A_2	$\boxed{2}$	$\boxed{6}$	9	11	15
A_3	$\boxed{3}$	$\boxed{5}$	9	12	14
A_4	4	7	9	13	
NE (+2)	$\boxed{2}$	$\boxed{4}$	$\boxed{4}$	8	8
SW (+2)	$\boxed{2}$	—	—	—	—
T_{NE} (+4)	$\boxed{6}$	$\boxed{10}$	13	15	19
F_{NE} (+5)	$\boxed{5}$	$\boxed{7}$	$\boxed{7}$	11	11

The proof that a satisfying assignment for F implies a solution for the corresponding instance D of DPM2 follows easily from earlier discussions.

We turn to the other direction, that a solution for D implies a satisfying assignment for F.

There are 7 potentially useful types of supply points (the reader should be convinced that any other supply points can be moved to become one of these without increasing cost):

- A_i for $i = 2, 3, 4$ is a point whose nearest south- or westward neighbor is i units away. When $i = 4$, this includes the points 1 unit away from a NW or SE corner;
- NE for a point in a NE corner;
- SW for a point 2 units north and 2 units east (or 1 unit in one direction, 3 in the other for a kink) of a SW corner;
- T_{NE} for a clause point that is either 4 units north and 2 units west or 4 units east and 2 units south of a NE corner;
- F_{NE} for a clause point that is 2 units north and 1 unit east of a NE corner;

Table 2 shows the cost of adding each additional demand point (from a circuit) to a cluster whose supply point is of a given type. Squares are drawn around the entries that correspond to legitimate clusters. The entries for A_2 and A_3 are for clusters in the true and false partitions, respectively. An A_4 supply point already has an extra cost of 3 for a cluster of three points ($7 + 4 = 11$) and cannot be used unless offset by savings elsewhere.

If we ignore the NE and SW supply points, it is clear that, for circuit clusters, the marginal cost of adding a point to any legitimate cluster outweighs any possible savings (the most that can be saved with respect to circuit clusters is 8 and the marginal cost of extra points in them is at least 9). Because we ensured that the points not in the same cluster are at least distance 9 away, it also follows that incorporating an unrelated point into a cluster must increase overall cost.

The marginal cost of adding each of the first two additional points to a NE cluster is 8, which suggests the possibility of "breaking even" by enlarging such a cluster. However, as Fig. 9 illustrates (create, e.g., a one-point cluster from the topmost point of the cost-17 cluster, reducing its cost to 8 — this is the cheapest alternative), the ability to use the two-point cluster with cost 4 is lost (unless some even more costly "re-synchronization" along the circuit is done).

Clause points to the south or west of a circuit cannot be used as supply points for any points in the circuit. As demand points, they can be added with a marginal cost of 12, but only if aligned with a true/false cluster; the marginal cost is at least 15 otherwise.

That leaves the T_{NE} and F_{NE} supply points. The former yields total cost 20 when a three-point true cluster is added, thus increasing cost by 12 as desired. The latter yields total cost 24 when added to the four-point cluster whose cost without it is 12. Adding more demand points from the circuit to the cluster of one of these supply-point types does not allow a decrease in circuit cost — the marginal costs, 13 and 11, respectively, are too great.

7 Conclusions and Future Work

We have shown that 2DPM is NP-complete for the rectilinear metric when p is part of the input. Extending this result to the Euclidean case is likely to be tedious, but not fundamentally different (and we are not aware of applications for the directional p-median with Euclidean metric). The use of a large value for p in the construction appears to be necessary, and fixed-parameter tractability (see, e.g., [3]) with respect to p is still open, as far as we are aware. It is also not clear to what extent the optimization version of the problem has polynomial approximation algorithms. Good approximations do appear to be commonplace for practical and randomly-generated instances in our related experimental work [5, 6].

Acknowledgment

We are indebted to the anonymous referee who reviewed our initial submission of this paper to *Information Processing Letters* for suggesting that there were nontrivial issues related to lining up circuits with clause points. We also appreciate associate editor Susanne Hambrusch for her willingness to allow a resubmission after correcting these details.

References

[1] Mark Daskin. *Network and Discrete Location: Models, algorithms, and applications*. John Wiley and Sons, New York, 1995. 1

[2] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999. 3

[3] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995. 18

[4] R. Hassin and A. Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10:395–402, 1991. 2

[5] L. E. Jackson. *The Directional p-Median Problem with Applications to Traffic Quantization and Multiprocessor Scheduling*. PhD thesis, North Carolina State University, 2003. 18

[6] L. E. Jackson, G. N. Rouskas, and M. F. Stallmann. The directional p-median problem with applications to traffic quantization and multiprocessor scheduling. Submitted to *European Journal on Operations Research*, Dec 2003. 18

[7] Laura E. Jackson and George N. Rouskas. Optimal quantization of periodic task requests on multiple identical processors. *IEEE Transactions on Parallel and Distributed Computing*, August 2003. 2

[8] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329 – 343, 1982. 2

[9] Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, February 1984. 1, 3, 9, 12

[10] Bernard M. Moret. *The Theory of Computation*. Addison-Wesley, Reading, MA, 1998. 2

[11] A. Papakostas and I. G. Tollis. Algorithms for area-efficient orthogonal drawings. *Computational Geometry: Theory and Applications*, 9:83 – 110, 1998. 3