BOSTON UNIVERSITY

THE p-MEDIAN MODEL AND THE LOCATION OF SCHOOLS IN THE CERDÀ PLAN OF EXPANSION OF BARCELONA

Montserrat Pallarès

DEPARTMENT OF GEOGRAPHY

APRIL 1986

I am fully indebted to my advisors, professors Jeffrey Osleeb and Samuel Ratick, for their guidance and continuous encouragement. Also, I would like to thank professor Thomas Glick for leading my attention to the ideas and work of Ildefonso Cerdà. Finally, I wish to thank the members of the Department of Geography for providing such a friendly and stimulating environment, as well as for the financial support received through these years. All remaining errors or shortcomings are entirely mine.

1. INTRODUCTION

Many of the present day 'modern' cities were transformed from their medieval pasts based on blueprints laid out by visionary city planners. These city planners resorted to methods of calculation prevailing in their day and age. This was probably combined with an uncarry instinct and intuition that identified the talented city planner from the rest. Ildefonso Cerdà was one such planner and the advent of the city structure of modern Barcelona may be attributed to his plan. Although Barcelona today is not entirely as Cerdà had visualized it, it is important to note that it is the result of changes imposed subsequently on the original Cerdà Plan. In order to chart out a course for future city planning in Barcelona it is essential to understand the original plan in the context of the objectives and anlytical tools currently in vogue.

Several programming techniques have been developed to resolve problems posed by various constraints to desirable objectives. Hence various programming models have been applied to the problem of suitable locating facilities like schools subject to criteria that define the user's convinience.

The objective of this paper is to demonstrate how the p-Median Model (15) can be used to test the optimality of the distribution of schools in Barcelona as proposed by Cerdà (2). The purpose here is to demonstrate a methodology for making such comparisons. A meaningful optimality check would ordinarily require using the whole plan. However, since that was found to be too ambitious, a small sample of nodes has been considered for the limited purpose of showing the application of a technique.

The Plan for the expansion of the city of Barcelona as proposed by Cerda, a 19th century urban planner, was adopted by the spanish goverment for several years. However, the documents, known as the Cerdà Plan were formally published only in 1961. Since a large part of Barcelona, as it exists today, was based on this important document, an understanding of the underlying framework is useful for further planning.

The orthogonal pattern of streets and the diagonal avenues, designed in the Cerdà Plan, constitute the skeleton of the city. This Plan was a guide for the transformation of a medieval city , as Barcelona at the end of the Nineteenth Century, to a modern city. Markets, schools, parks, hospitals, and other important urban services were planned but not accomplished by the subsequent implementation of the plan. The Plan was the focus of controversial discussions among important economic and power sectors. One of the results of this controversy was that a number of documents were 'lost'. Thus, part of the methodology of the Plan is not known. The distribution of facilities in the Plan reflects a sense of regularity and suggests the existence of some optimizing criterion. In this study a modern optimizing technique, namely the p-Median Model is used in the understanding of the underlying methodology of an existing part of a City Plan.

The theory of location of public facilities has led to the development of several modern optimizing techniques. The p-Median Model is one such technique whose objective is to locate facilities so that the average distance or time that users take to reach them is minimized. The basic constraint is that there is a fixed budget for the system of facilities and that all facilities have the same "investment" (per facility) and "expansion" (per user) costs. Hence, by comparing the distribution pattern from the p-Median Model (whose objectives and methodology are known) with that of the Cerdà Plan, some conclusions can be drawn about the optimality and possible objectives contained in it.

The reason for developing the p-Median Model in this paper is that the objective of the model corresponds to the goal of locating the schools minimizing the weighted distances . Which is the objective of this study.

This paper is organized as follows: section II is devoted to the literature and concepts relevant to this paper. In section III the p-Median Model's objective function and assumptions are explained. Section IV is devoted to the description of the data used in this paper. The application of the p-Median Model methodology is made using three algorithms. First a small sample is chosen from the Cerdà map in which two facilities are to be located. A mathematical package is used to run the linear programming algorithm. We assume in this part that nodes of population have the same distance to each other; i.e., all of them have a regular distance pattern, no node is different from the others. This is shown in sections V to VII. In section VIII another area has been chosen. This area is divided into four parts. A single facility is located in each one, assuming the so called L-1 Rectilinear distance (6,9). Two different procedures have being used to locate these facilities. The first one is called the Centroid Procedure (15). The second one is an algorithm in which the weighted distances from each point to the other in the area are calculated, and the smaller distance is chosen. This is the one that minimizes the weighted distances among all the nodes in the area. This is shown in sections IX and X. The objective of the p-Median Model is the same in all three cases, only the approaches to techniques are different. Section X contains the conclusions.

II. BACKGROUND

Hakimi's p-Median Model (15) is one which seeks to locate a fixed number of facilities. As it ignores the changes over time, the model is a static one. In our problem we are concerned with the location of schools. Facilities such as schools and libraries may be considered as "travelled for goods" opposed to fire engines and ambulances which are "delivered for goods" (12). In the first instance consumers must travel in order to get the good while in the latter the good is delivered to the consumer.

The p-Median Model belongs to the group of models, in Location Theory, called Consumer Oriented location models (15). In these models the objective consists of minimizing the average distance or time that users take to reach a facility subject to a constraint related to investment costs and/or operating costs of the facilities. In contrast, the objective function of the Goods-Oriented location models contain both the cost of goods movement and the cost of facilities (15).

Aternatively, Erlenkotter (5) distinguishes between public and private facility location models : "According to this differentiation public sector models typically have the objective of minimizing client cost for a given level of service subject to a public budget constraint while private sector models seek to minimize the total cost for meeting specified fixed demands". The p-Median Model, then falls in the category of public facilities sector models.

It is important to consider the characterization of <u>public goods</u> in the <u>spatial</u> context. According to Samuelson (11), a pure public good is generally defined as a good that remains undiminished following consumption. Similarly, Bigman and Revelle (11) define the public good as a good 'y' such that "each individual can consume the total amount of 'y' produced without diminishing anyone else's consumption of it". This implies that public goods are necessarily equally available to all consumers but does not indicate who those consumers will be.

The factor "space" gives to the public good a "difference in availability" of consumers to it. That is, when the location of consumers relative to the public good varies, even if everyone has the same utility function, the consumption of the public good will be different among the consumers. This idea is behind the objective function of the model used in this paper.

Thus far we have discussed a public good in general. We now define <u>public facilities</u> as those aspects of city structure whose primary function is to facilitate the provision of goods and services. They may be wholly or partly within the domain of government. The range of such goods and the degree of governement involvement may vary over time and place (18). For the purpose of the present paper, we assume that schools are the responsibility of the government. This is because the law in Spain states that a percentage of land has to be devoted to public schools. This percentage is in proportion to the population which lives in the given area.

III : THE p-MEDIAN MODEL

The p-Median Model can be developed in a multiobjective The first objective would be to minimize the travel distance or travel time between a population center and public facility. The second is to find and optimal or acceptable maximum time or distance that separates a user from its nearest facility (14). For example, consider a given region in which three similar facilities are located. Each facility now serves a particular population group. Thus, three areas are created such that the distance between the facility and points on the area's limits represent the maximum that the population can travel to reach it. However, this second objective has been set aside for future research. And hence will not be further explained here.

The first objective of the p-Median Model, as already stated, is (A) to minimize average travel distance of all users to their nearest facility, with every user assumed to travel their nearest facility. This is subject to the constraints that ensure (B) that all users receive service, (C) that service is provided only from facilities that are open, and (D) the total number of facilities is not exceeded. This objective is the only one which is going to be used in this paper.

The mathematical formulation of this model is as follows :

$$\text{Minimize } Z = \Sigma^{n}{}_{i=1} \Sigma^{n}{}_{j=1} a_i \, d_{ij} \, x_{ij} \qquad \text{(A)}$$

Subject to

$$\Sigma^{n}{}_{j=1} \; x_{ij} = 1 \quad \text{for all } i \qquad \text{(B)}$$

$$x_{ij} - x_{jj} \leq 0 \quad \text{for all } i,j \quad i \neq j \qquad \text{(C)}$$

$$\Sigma^{n}{}_{j=1} \; x_{jj} = p \quad \text{for all } i,j \qquad \text{(D)}$$

$$x_{ij} = (0,1) \quad \text{for all } i,j$$

where,

a = relevant population of the ith node.

j = possible point of location of a facility.

i = node of population.

d = the shortest distance from node i to node j.

$$x_{ij} = \begin{cases} 1 \text{ if node i does assign to node j} \\ 0 \text{ otherwise} \end{cases}$$

$$x_{jj} = \begin{cases} 1 & \text{if a facility does open at node j.} \\ 0 & \text{otherwise} \end{cases}$$

SECTION IV. DATA

Data from population within each node, and distance from population to facility are needed to undertake this analysis.

The Cerdà Plan proposes a population of 250 inhabitants per hectare (2.47 acres), or 40 square meters for each inhabitant. These were standards requested by the public health committees. Most of the Cerdà blocks had an area of 12,370 square meters. Therefore, the population in each block consists of approximately 310 inhabitants. The normal distribution of population between 4 and 14 years old in the age pyramid is 27 percent. Thus, in each block 84 inhabitants go to primary school. In the smaller blocks the primary school population is counted proportional to the area of each particular block. By block, in this paper, we refer to the squares in which the map of Barcelona is divided (see Figure 1). Sometimes, a block is divided in a half or a quarter by the patern of the streets, these are the smaller blocks. We refer also in this paper to a block as a node of population or, shortly, as a node.

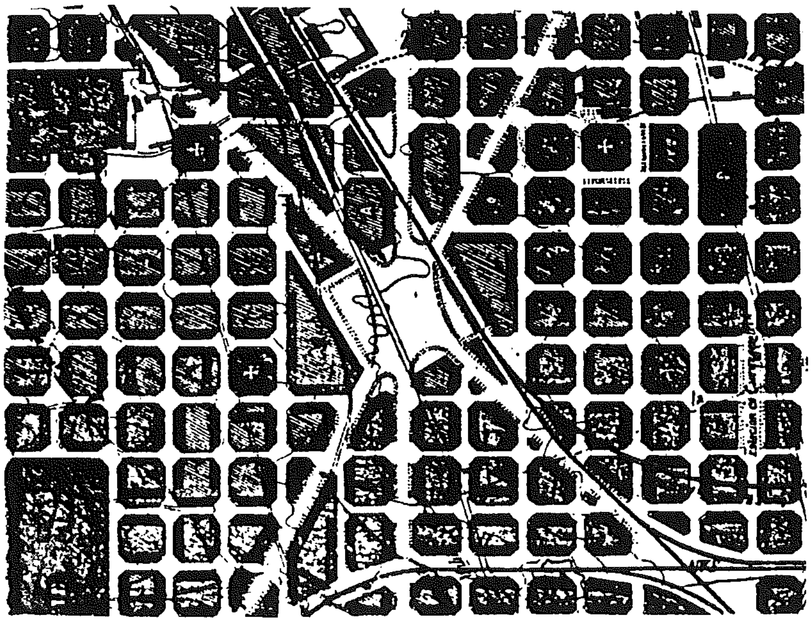


Figure 1. Part of Cerdà's Map with examples of nodes.

Throughout this paper, the term population nodes refers to small communities separated from one another by streets. It is assumed that the population of each node is concentrated at one point at the center of the node. This assumption is made because the area of each node is small enough to assume that people living in different parts of it will have on average to travel the same distance to the next node. Therefore, distances are measured from this point of each population node. In other words if community "i" is the site of a facility, the population of community "i" has to travel zero meters (by assumption) to the facility.

Distance is taken under the assumption of "Manhattan or L_1 Rectilinear Distance". That is people cannot travel diagonally between blocks, but must follow the assigned orthogonal pattern of streets.

V. ASSUMING REGULAR PATTERN/DISTANCE

Streets are calculated as 20 meters long and blocks as 113 meters long. Since a person must travel only along streets, the distance to be covered between one block and the next is 133 meters.

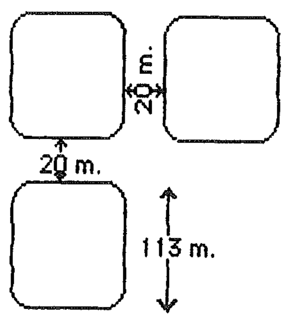


Figure 2. Calculation of Distances in Regular Pattern in the Cerdà Plan.

Space in the Cerdà Plan is divided into similar blocks but of different magnitudes. Blocks have the same size of 113 square meters with the exception of a few which range from one quarter to twice the area of a common-size block.

VI. APPLICATION OF THE MODEL IN REGULAR DISTANCES PATTERN

To choose a sample to demonstrate that the p-Median Model can be used to test the optimality of the location of schools in a given area, a group of four nodes of population was randomly selected from Cerdà's map (see Figure 3). Two schools were to be located to serve these four nodes.

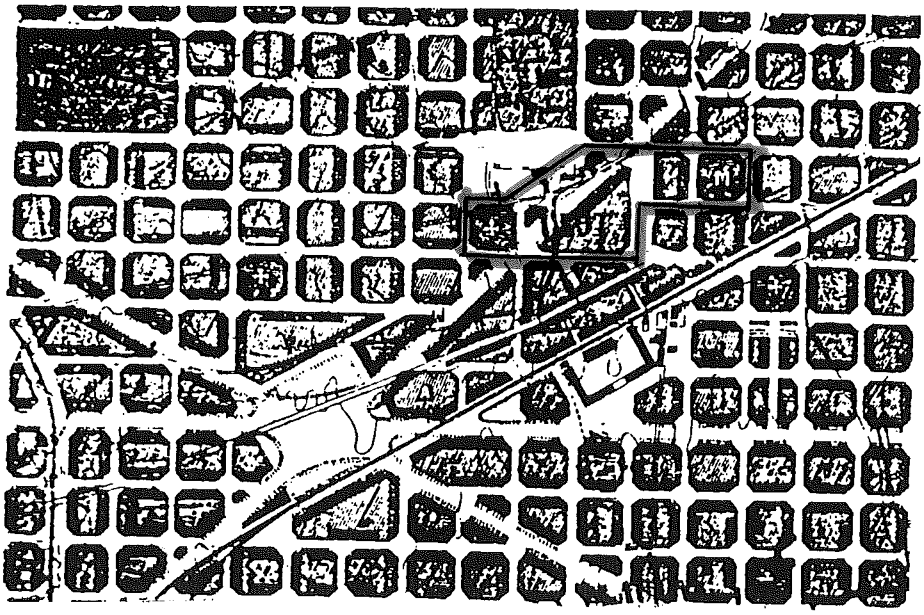


Figure 3. Area in which 2 schools have to be located.

School population in each node and distance among nodes are shown, respectively, in Tables I and II.

Node No.	Population
1	84
2	126
3	74
4	84

Table I. Population

	1	2	3	4
1	0	153	439	572
2		0	153	286
3			0	133
4				0

Table II. Matrix of Distances

To obtain the results the MPSX (Mathematical Programming System-Extended) package was used. (Refer to pp. 39-40 for listing of the program). The results are discussed in the next section.

VII. INTERPRETATION OF RESULTS IN REGULAR DISTANCES PATTERN

On page 38 there is a listing of the results obtained. According to these, if the two schools are located in node number two and node number four, the value of the objective function is 22,700, this value represents the total weighted distance that population must travel to get school.

Assigments are as follows: populations in node one and node two are to be served by a school in node two, with a weighted distance of 12,900; while populations in node three and four must be served by the school in node four, with a weighted distance of 9,800. Based on the problem formulation, such an arrangement constitutes the minimum distance that the population in each node has to travel to reach a school. Hence, the location of schools is optimal.

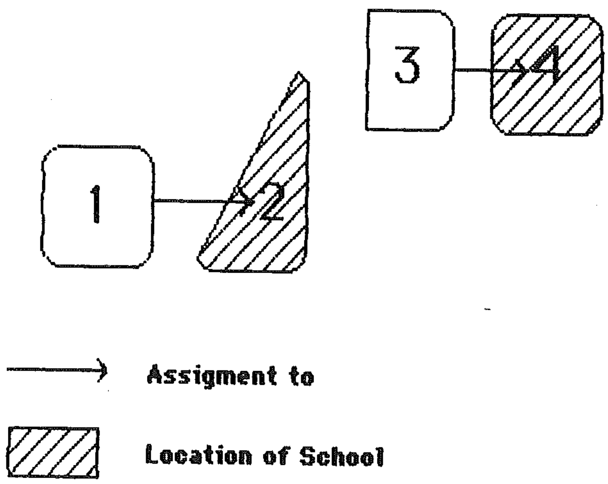


Figure 4. Distribution of schools and assigments of population according to p-Median Model in the chosen sample.

The entire problem was conceived so as to compare results obtained as demonstrated above with the actual locations marked in Cerdà's Plan.

For a meaningful comparison, merely taking a sample of nodes from the Plan is insufficient in order to answer the question raised. One would necessarily have to compare the whole plan and not just a part of it. From the map (page 37) it can be observed that no particular set of nodes can be exclusively considered. This is because the relationship of nodes to schools on the boundary of the sample is ambiguous. Assume that the program is run with the same proportion of schools to population nodes as in Cerdà's Plan, a node included in the sample for analysis may actually be served by a school outside the selected nodes in the Plan and vice versa.

Hence a meaningful application of the methodology demonstrated in this paper can occur only if the model is run with data for the entire Plan. However, the Plan constitutes more than 800 nodes of population in which 32 schools are to be located. Since a solution of the problem using all the data is beyond the scope of this section, only a sample has been demonstrated.

VIII. ASSUMMING L_1 RECTILINEAR-DISTANCE PATTERN. THE CENTROID PROCEDURE.

Defining L_1 Rectilinear-Distance or Manhatan Distance as the one where travel occurs along an orthogonal set of streets.

The rectilinear-distance problem can be stated mathematically as

$$\text{Minimize } f(x,y) = \Sigma^{m}_{i=1} \Sigma^{m}_{j=1} w_i(|x_i - x_j| + |y_i - y_j|)$$

where,

(x_i , y_i), and (x_j , y_j) are the coordinates of points of demand on a plane, and

w_i is the weight. In this paper w_i is the primary school population which lives in each demand node.

To work with L_1 Rectilinear-Distance we start with the assumption that thirty six facilities should be located on the whole map of Barcelona (page 37). In the Cerdà map there are only thirty two facilities to locate on the map. The reason why it is necessary to use thirty six instead of thirty two will become clear in the following paragraphs.

We divide the entire map into thirty six regions in such a way that in each region one faciltiy should be located. The division is made in the following way : the whole map is placed on the x and y axes. We break up the x and y axes in six equal area parts. Until now the whole map has been divided in thirty six equal areas and we might locate a facility in each of them. And we will now provide to the significance of the assumption stated above; i.e. we want to have thirty six facilities because that facilitates the division of axes in six equal parts.

The following step is to choose the area in which we want to work. Any area of the map is useful to make the study. We choose randomly four small areas, which come from the division stated above , and this will be the area in which we have to locate four facilities.

This area, which we call the Area of Study, has one hundred and thirty five nodes, most of them having a certain number of inhabitants. Few of them have no inhabitants because they are parks or other types of facilities where there is no population.

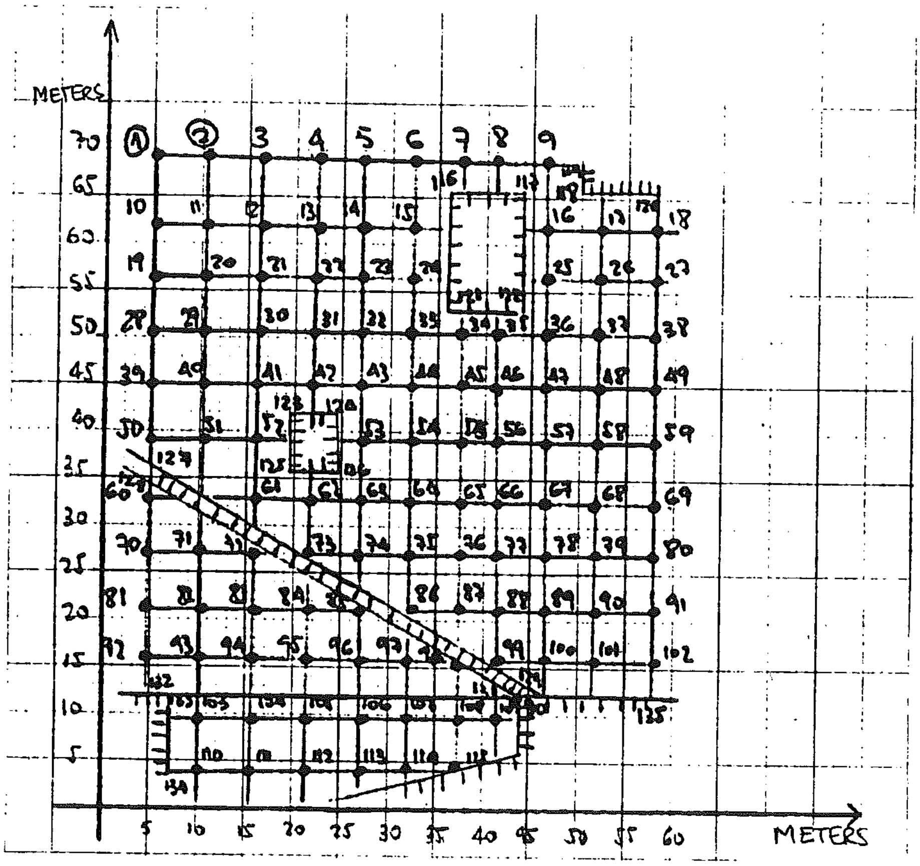


Figure 5. Area of Study assuming L_1 Rectilinear-Distance.

First, the Area of Study is treated as a single unit. We want to divide the area into four equally weighted parts. In order to do that we look for the centroid of the area. Given the x and y coordinates of this area, we look for the centroid in each of the axes. A centroid is a point on the plain which minimizes the weighted distances of the x and y coordinates.

There are different ways to calculate the centroid solution, all of which give us the same result. One way to calculate it is given by Francis and White (6, pp.184):

The formulation of the problem is :

$$\text{Minimize } f(x,y) = \Sigma^m_{i=1} w_i[(x-a_i)^2 + (y-b_i)^2]$$

where, x, y, a_i, b_i are the coordinates of points in a given area.

Any point (x^*, Y^*) that minimizes this objective function must satisfy the conditions :

$$\left(\frac{\partial f(x^*, y^*)}{\partial x^*}, \frac{\partial f(x^*, y^*)}{\partial y^*}\right) = (0,0)$$

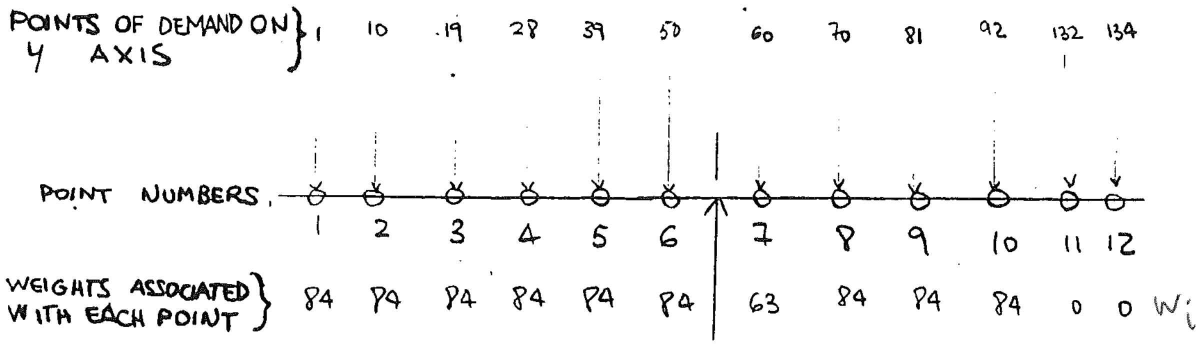
Then, computing the partial derivatives of the objective function with respect to x and y and setting them to zero gives the following unique solution :

$$x^* = \frac{\Sigma^m_{i=1} w_i a_i}{\Sigma^m_{i=1} w_i}$$

$$y^* = \frac{\sum^{m}_{i=1} w_i b_i}{\sum^{m}_{i=1} w_i}$$

The coordinates x^* and y^* are the ones which minimize the objective function stated above. This formulation corresponds to the Gravity Problem formulation, but the same could be applied to the L_1 Rectilinear-Distance. Similar way to calculate the centroid solution on a plain is given by Revelle and Church (14, pp. 10), and is the one used in this paper:

To find a centroid we first consider locating a single facility on a line on which are scattered points from which people travel. These points represent nodes of demand.



$$\sum_{i=1}^{6} w_i = 504 \qquad \left(\sum_{i=1}^{6} w_i - \sum_{i=7}^{12} w_i\right)\Delta$$

$$\sum_{i=7}^{12} w_i = 315 \qquad +189\Delta$$

$$\sum_{i=1}^{5} w_i = 420 \qquad +21\Delta$$

$$\sum_{i=6}^{12} w_i = 399$$

$$\sum_{i=1}^{4} w_i = 336 \qquad -147\Delta$$

$$\sum_{i=5}^{12} w_i = 483$$

Figure 6. Calculation of Centroid on the Y axis.

The facility point is moved across the line from left to right in an effort to find the point from which the weighted distance is a minimum. At each point where the facility temporarily lodges, it is necessary to evaluate the sum of weights multiplied by distances to that point.

In fact, an easier procedure is the following: suppose the trial site where we locate the facility is between points 6 and 7 (where the arrow is located). If the trial site is moved to the right by Δ distance units, the weighted distance from points to the left is increased and the weighted distance from points to the right is decreased. In the example shown in Figure 6, the weight to the left of the trial site is $\Sigma^6_{i=1} w_i = 504$; the weight to the right is $\Sigma^{12}_{i=7} w_i = 315$. The change in total weighted distance on movement by Δ to the left is

$$(\Sigma^6_{i=1} w_i - \Sigma^{12}_{i=7} w_i)\, \Delta$$

where the term in parenthesis is positive since the value is going up. We want to calculate the change in movement of Δ if the trial site were between points to the left of the one we started, because we want our weighted distance to go down. The point where the term in parenthesis is zero is where the sum of the weights to the left equals the sum of the weights to the right. The minimum weighted distance occurs either at the point where the slope of the objective changes sign or at the points where the objective is constant. In this case the minimum point is number ~~4~~ 5; as shown in figure IV the objective changes sign (from +21 to -147). Thus the centroid (or median) in the vertical axis (y) is point 28. 39.

This concept is extended to two dimensions. The centroid in the horizontal direction is established independently of the centroid in the vertical direction. These two centroids give us two lines which intersect and separate the Area of Study in four smaller areas.

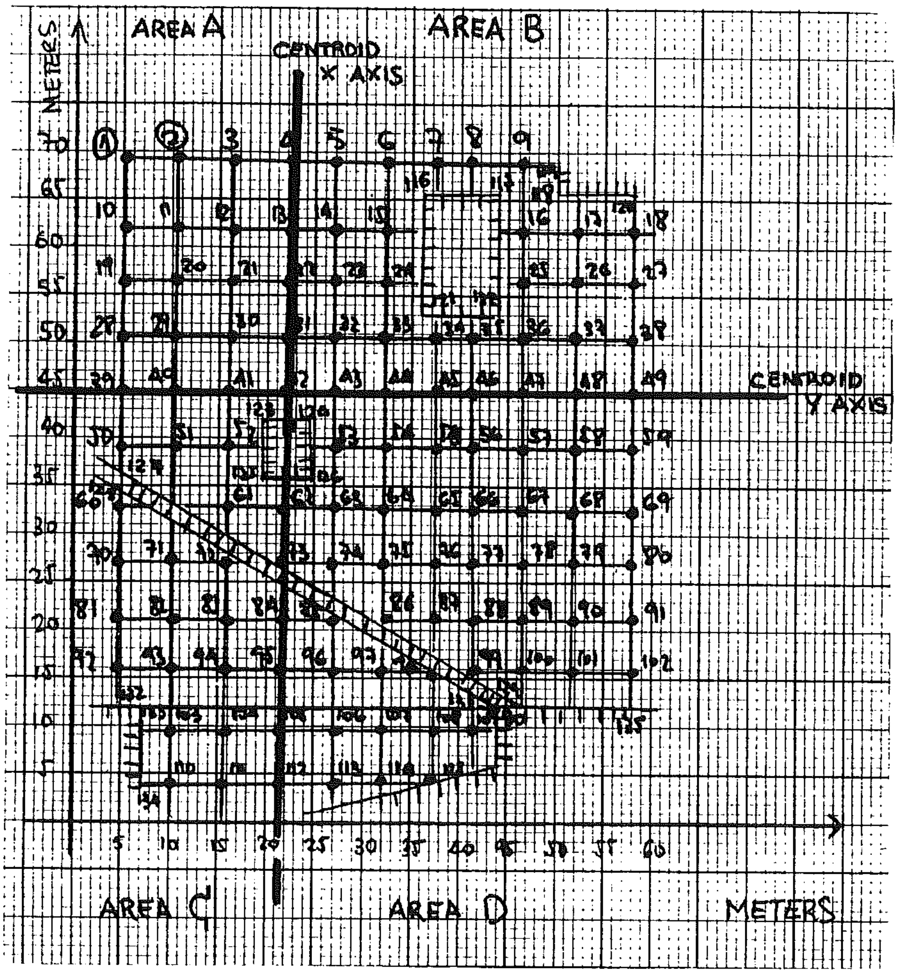


Figure 7. Division into four smaller areas by the centroids of the X and Y axes.

Within each of these smaller areas we want to calculate the centroid to locate a single facility to minimize the weighted distance. <u>This will</u>

give us the solution to our p-Median Problem under the L_1 metric assumption. In pages 48-55 the calculations of these centroids are shown. And the results are:

1. In area A the points in which we want to locate the facility are numbers: 20 and 21. Thus there are two alternative solutions in this area. The computation of weights on the X axis gives as candidate, points 40 and 41; thus the centroid on the X axis has two equally weighted points (page 48). Computation of weights on the Y axis gives as candidate, point 19 (page 49). Thus the point where the two lines (the centroid line on the Y and on the X axes) cross gives the results stated above.

2. In Area B the point in which we want to locate the facility is located inside a park; i. e.; the intersection between the Y centroid and the X centroid are points inside an area in which the construction of buildings is not allowed. The candidate point to be the centroid for the X axis is number 45; and the one to be the centroid for the Y axis is number 22 (pages 50-51). The point where the two lines cross lies on a park. So, we need to look for another technique to locate a school in this area. We study this problem in the following section.

3. In Area C the point in which we want to locate the facility is: 83. There is only one optimal solution in this area. The candidate point to be the centroid for the X axis is number 111 (page 52). The one for the Y axis

is number 81. Hence, the point where the two lines cross gives the solution stated above.

4. In Area D the point in which we want to locate the facility is : 76. Again there is only one optimal solution in this area. The centroid for the X axis is point number 115 (page 53). The one for the Y axis is number 73 (page54). The point where the two centroid lines cross is the one stated above.

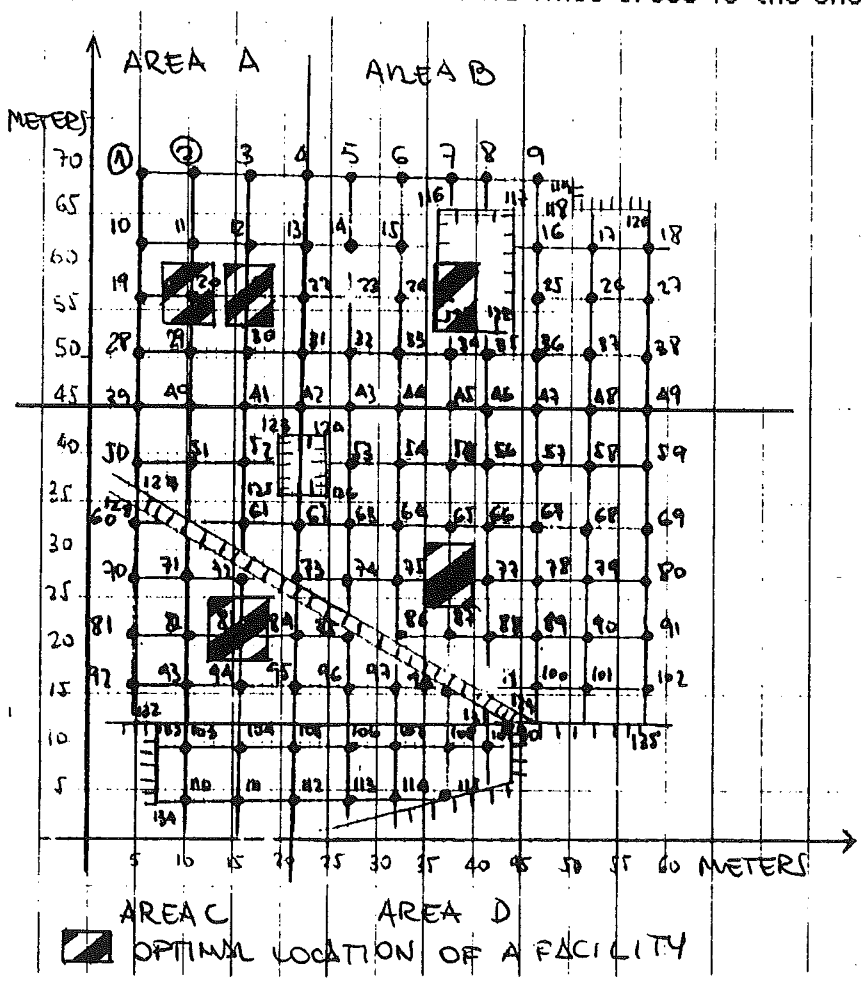


Figure VI. Optimal Location of Facilities Given by Centroids in the Smaller Areas.

Thus, in this Section we have found a solution for the initial problem of locating a number of facilities minimizing the weighted distances, the objective function of the p-Median Problem. All p-Median points with minimum distances are allocated, but a problem is discovered in Area B where these minimum points are inside a park. In a park, construction of buildings is not allowed; so, another technique to find the optimal point is developed in the next section.

IX. COMPUTATION OF THE MINIMUM WEIGHTED DISTANCES IN L_1 RECTILINEAR-PATTERN USING THE THIRD ALGORITHM

We want to minimize the sum of the weighted absolute distance between each node of demand and each node of supply. i. e.; points of demand are all points aimed in each of the Smaller Areas defined in section VIII (i.e.; areas A, B, C, D); points of supply are the points where we will locate a facility.

The procedure to get the optimal solution and, hence, the solution to our p-Median Problem is the following.

1. Suppose you take Area B (any of the areas can be choosen) and assume a school is located in node number one, with coordinates (x_1, y_1);

2. Now, the absolute weighted distance among node one and all of the rest in Area B is calculated. This is done in the following way : suppose coordinates in point one are (5 , 69), and coordinates in point two are (10,69). Population in point two is 84. Thus, the absolute weighted distance between point one and point two is

$$[|5 - 10| + |69 - 69|] \times 84 = 420$$

Suppose coordinates of point three are (19 , 62), and population is 21. Thus, the absolute weighted distance between point one and point three is

$$[|5 - 19| + |69 - 62|] \times 84 = 1764$$

the same method may be continued until all nodes in Area B are exhausted.

3. Find the sum of the absolute weighted distance of each point in Area B and point one, i. e.; following the example started in part 2:

$$420 + 1764 + \ldots$$

This will give us the absolute weighted distance that all the primary school inhabitants in Area B have to travel if we locate the school at point one.

4. For each node of Area B, steps 1 to 3 should be done. The node where the absolute weighted distance is minimum is the one where we should locate the school. Thus, in this point we will locate the facility, and this will be our solution to our p-Median problem.

Doing all of the steps for each of the Smaller Areas already defined, we will have the solution to the problem of locating four schools in the four Areas.

Once those computations have been completed (refer to pages 56-[illegible] for computational results), the solutions are:

1. Area A. The optimal points are 20 and 21. Note that the solution is not single and that the solution coincides with the one given in section VIII using the centroid method. If a school is located at one of these two points the total weighted distance that population has to travel to reach the school is of 20,496 (see pages 56-[illegible]).

2. In Area B the opimal point is 34. Thus, using this method we have obtained an optimal point outside the park . In section VIII, the optimal solutions given by the centroid method were points located in a park. If a school is located in point 34, the total weighted distance that population has to travel to reach school is of 50,820 (see pages 66-69).

3. In Area C the optimal point is 83. This point does coincide with the one given by the centroid procedure. Total weighted distance that population in this area has to travel to get the school is 36,015. (pp.70-73)

4. In Area D the optimal point is 76. Again this point is the same as the optimal point given by the centroid procedure. Total weighted distance that population has to travel to get the school is 85,848. (pp.74-79)

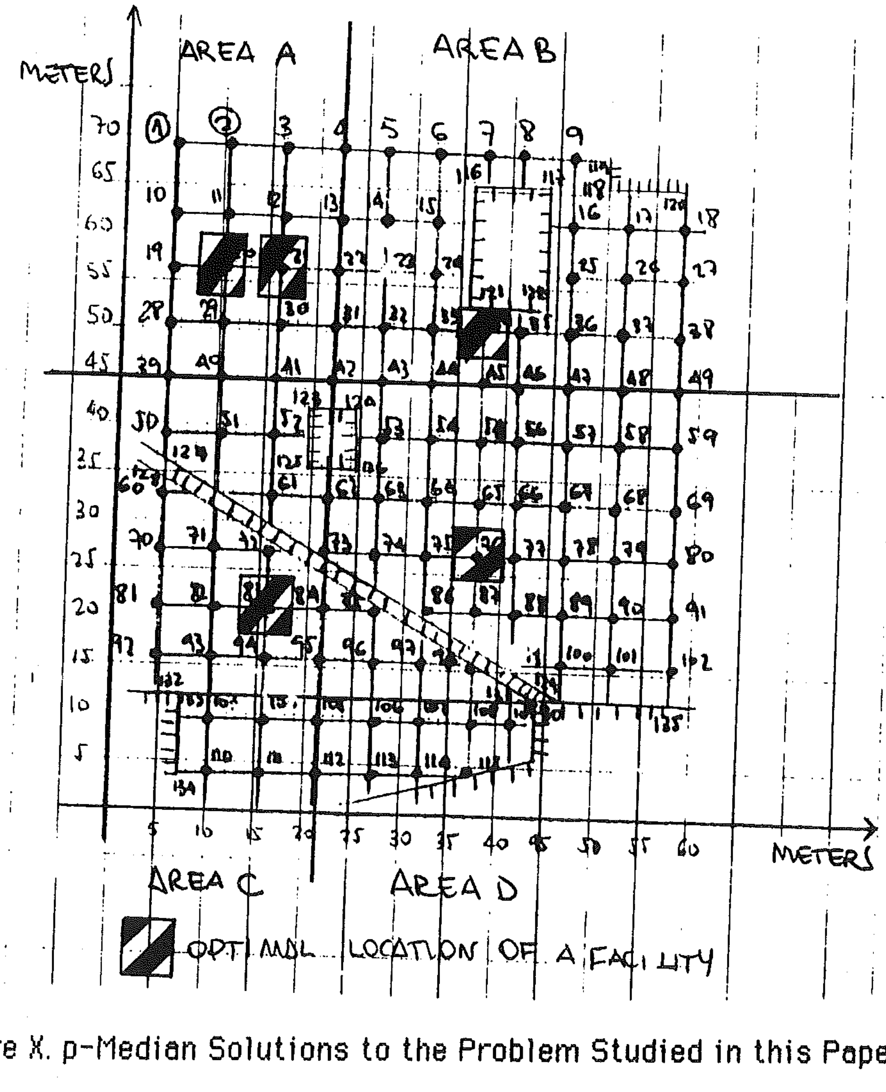


Figure X. p-Median Solutions to the Problem Studied in this Paper.

X. CONCLUSION.

The objective of this paper was to demonstrate how the p-Median Model (15) could be used to test the optimality of the distribution of schools in Barcelona as proposed by Cerdà (2). Three algorithms have been developed in this paper in order to operationalize the p-Median Model. The accomplishment of this objective has been demonstrated in two different samples taken from the Cerdà Map. The results necessary to actually carry out a comparative study are not obtained in this paper. The complete comparison between the optimal locations of schools as obtained by the L_1 techniques used in this paper and the locations of all schools in the Cerdà map has to be posponed for future research. Such a comparison should ideally include the entire map or relevant zone, and not just a sample. Within the limitations of the samples selected for study here, it is found that the algorithms for the p-Median Model are appropriate to test the optimality of facility distributions in city plans such as the Cerda Plan.

Thus far the model is based on the assumption of a omogeneous and isotropic plane. For a realistic test of the entire Cerda plan barriers to travel will also have to be included. The imposition of such barriers (9, 10) to travel on the areas of study may lead to different results to our problem. In urban location of facilities the L_1 metric is a reasonable approximation to travel behavior. Using the L_1 metric approximation, introduction of urban barriers, such as parks, lakes, rivers, etc., to the foregoing problem can constitute a continuation of the problem planned in this paper. Under such circumpstances the results may be similar to the facility distributions in the Cerda Plan.

Therefore, a useful application of a modern optimizing technique, namely, the p-Median Model in understanding the underlying methodology of the Cerdà City Plan has been demonstrated.

XI. BIBLIOGRAPHY

1. BOHIGAS GUARDIOLA, ORIOL "Le Plan Cerdà", Arquitecture Forms and Fonctions, 6 annee, 1959, pp. 58-60.

2. BRONSON, RICHARD Operations Research, McGraw-Hill Book Company (1982).

3. CERDÀ, ILDEFONS Teoria General de la Urbanización, vol. 1: Teoría General de la Urbanización; vol. 2: La Urbanización Como un Hecho Concreto. Estadística Urbana de Barcelona; vol. 3: Vida y Obra de Ildefonso Cerdà. Anexo Documental y Bibliografía, by Fabián Estapé; 3 vols. (Barcelona : Ariel and Vicens Vives, 1968-1971).

4. CERDÀ, ILDEFONS La Theorie Generale de l'Urbanization. (Paris : Editions du Seuil, 1979).

5. COHON, J.L., et al. "Population/ Cost Tradeoffs for Nuclear Reactor Siting Policies". Operations Research Group Series, Paper 81-04, The johns Hopkins University, Baltimore.

6. ERLENKOTTER, D. "On the Choice of Models for Public Facility Location", Working Papaer, International Institute for Applied Systems Analysis, (Loxenburg, Austria, 1980).

7. FRANCIS, RICHARD L. and WHITE, J.A. Facility Layout Location, Prentice-Hall, Englewood Cliffs, New Jersey (1974).

8. GOODCHILD, M.F. "The Aggregation Problem in Location-Allocation", Geographycal Analysis, vol. 11 no. 3 (1979).

9. HILLIER F. S. and LIEBERMAN, G.J. Operations Research, Holden-Day, Inc. (1980).

10. HILLSMAN, E. and R. RHODA "Errors Measuring Distances from Populations to Service Centers", Annalas of Regional Science, vol. 12 no. 3 (1978).

11. LARSON, RICHARD C. and LI, V.O.K. "Finding the Minimum Rectilinear Distance Paths in the Presence of Barriers", Working paper OR-088-79, Operations Research Center, M.I.T., (Cambridge, 1979).

12. LARSON, RICHARD C. and SADIQ, G. "Facility Locations with the L_1 Metric in the Presence of Barriers to Travel", Preliminary Version, Presented at the Secon International Conference on Locational Decisions (ISOLDE II), June 15-19, 1981 (Denmark).

13. LEA, A.C. "Welfare Theory, Public Goods, and Public Facilities Location", <u>Geographical Analysis</u>, vol 11 (1979).

Lynch, K. A Theory of good city form. Cambridge, Ma: MIT Press, 1981.

14. NARULA, S. C. et al. "An Algorithm for the p-Median Problem", <u>Operations Research,</u> vol. 25 (1977).

15. PARK, S. "Aggregation Effects and Disagregation of Demand in Location-Allocation Analysis", Discussion Paper no. 36, Department of Geography the University of Iowa, December 1983.

16. REVELLE, CH. et al. "Multiple Objectives in Facility Location : a Review", <u>Operations Research Group Report Series</u>, Paper 81-01, The Johns Hopkins University, Baltimore.

17. REVELLE, CH. and CHURCH <u>Location Analysis Theory</u>, unpublished material.

18. ROJESKI, P. and REVELLE, CH. "Central Facilities Location Under an Investment Constraint", <u>Geographycal Analysis</u>, vol. 2 (1970).

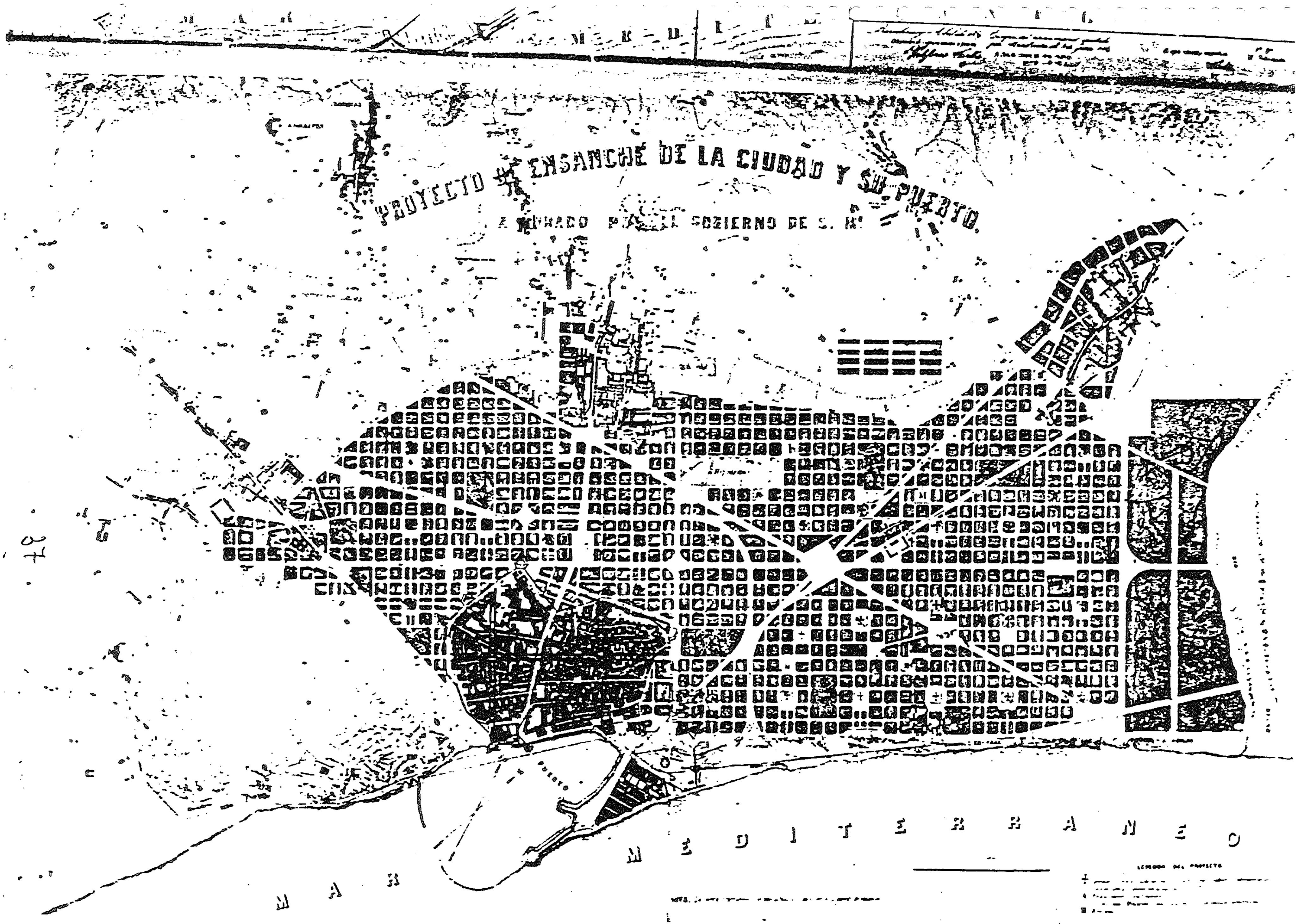
Samuelson, P. "The pure theory of public expenditures". Review of Economics & Statistics 36(4): 350-356, 1954.

19. SOLA MORALES, M. et al. <u>Los Ensanches (1) El Ensanche de Barcelona</u>, (Barcelona; Escuela Técnica Superior de Arquitectura, 1979).

Taylor,

20. TEITZ, M.B. "Toward a Theory of Public Facility Location", Papers of the Regional Science Association, 21 (1968).

APPENDIX I



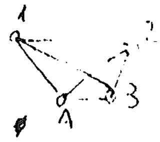
Proyecto de Ensanche de Ildefonso Cerdá (versión 1860).

SOLUTION TO THE SECTION VI

number	.columns	at	...activity...	..input cost..	..lower limit.	..upper limit.	.reduced cost.
0 19	x11	bs	.	.	.	none	.
20	x12	bs	1.00000	12.90000	.	none	.
21	x13	ll	.	36.90000	.	none	24.00000
22	x14	ll	.	48.00000	.	none	35.10000
23	x21	ll	.	19.30000	.	none	6.40000
24	x22	bs	1.00000	.	.	none	.
25	x23	ll	.	19.50000	.	none	6.60000
26	x24	ll	.	36.00000	.	none	23.10000
27	x31	ll	.	32.50000	.	none	21.20000
28	x32	bs	.	11.30000	.	none	.
29	x33	ll	.	.	.	none	1.40000
30	x34	bs	1.00000	9.80000	.	none	.
31	x41	ll	.	48.00000	.	none	36.60000
32	x42	ll	.	24.00000	.	none	12.60000
33	x43	bs	.	11.20000	.	none	.
34	x44	bs	1.00000	.	.	none	.

```
.mpsx/370 r1.6        mpscl execution                                   page
)punch
)time =    0.00

'exit - time =    0.00
'eof
```



$$\text{Min} \sum_i \sum_j d_{ij} a_i x_{ij}$$

s.t.

$$\sum_j x_{ij} = 1 \quad \forall i \qquad (1)$$

$$x_{ij} - x_{jj} \le 0 \quad \forall i,j,\ i \ne j \qquad (2)$$

$$\sum_j x_{jj} = p \qquad (3)$$

$$\text{Min } d_{11} a_1 x_{11} + d_{12} a_1 x_{12} + d_{13} a_1 x_{13} + d_{14} a_1 x_{14}$$
$$+ d_{21} a_2 x_{21} + d_{22} a_2 x_{22} + d_{23} a_2 x_{23} + d_{24} a_2 x_{24}$$
$$+ d_{31} a_3 x_{31} + d_{32} a_3 x_{32} + d_{33} a_3 x_{33} + d_{34} a_3 x_{34}$$
$$+ d_{41} a_4 x_{41} + d_{42} a_4 x_{42} + d_{43} a_4 x_{43} + d_{44} a_4 x_{44}$$

s.t. 1) a) $x_{12} + x_{13} + x_{14} = 1$

b) $x_{21} + x_{23} + x_{24} = 1$

c) $x_{31} + x_{32} + x_{34} = 1$

d) $x_{41} + x_{42} + x_{43} = 1$

2) a) $x_{12} - x_{22} \le 0$

b) $x_{13} - x_{33} \le 0$

c) $x_{14} - x_{44} \le 0$

d) $x_{21} - x_{11} \le 0$

e) $x_{23} - x_{33} \le 0$

f) $x_{24} - x_{44} \le 0$

g) $x_{31} - x_{11} \le 0$

h) $x_{32} - x_{22} \le 0$

i) $x_{34} - x_{44} \le 0$

j) $x_{41} - x_{11} \le 0$

k) $x_{42} - x_{22} \le 0$

l) $x_{43} - x_{33} \le 0$

3) $x_{11} + x_{22} + x_{33} + x_{44} = 2$

MPSX DATA ARRAY (SECTION VI)

ROW	X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{24}	X_{31}	X_{32}	X_{33}	X_{34}	X_{41}	X_{42}	X_{43}	X_{44}	Rel	
DMD1	1	1	1	1													=	1
DMD2					1	1	1	1									=	1
DMD3									1	1	1	1					=	1
DMD4													1	1	1	1	=	1
ASG12		1				-1											≤	0
ASG13			1								-1						≤	0
ASG14				1												-1	≤	0
ASG21	-1				1												≤	0
ASG23							1				-1						≤	0
ASG24								1								-1	≤	0
ASG31	-1								1								≤	0
ASG32						-1				1							≤	0
ASG34												1				-1	≤	0
ASG41	-1												1				≤	0
ASG42						-1								1			≤	0
ASG43											-1				1		≤	0
PFC	1					1					1					1	=	2
APROP1	0	12,9	36,9	48,0	19,3	0	19,3	36,0	32,5	11,3	0	9,8	48,0	24,0	11,2	0		

APPENDIX II

AREA	NODE	X &Y COORDINATES	WEIGHT OR POPULATION
A	1	(5 , 69)	84
A	2	(10 , 69)	84
A	3	(16 , 69)	84
B, A	4	(19 , 69)	84
B	5	(27 , 69)	84
B	6	(32 , 69)	84
B	7	(37 , 69)	84
B	8	(41 , 69)	84
B	9	(46 , 69)	84
A	10	(5 , 62)	84
A	11	(10 , 62)	84
A	12	(16 , 62)	84
B, A	13	(19 , 62)	84
B	14	(27 , 62)	84
B	15	(32 , 62)	84
B	16	(46 , 62)	84
B	17	(52 , 62)	84
B	18	(58 , 62)	84
A	19	(5 , 56)	84
A	20	(10 , 56)	84
A	21	(16 , 56)	84
B, A	22	(19 , 56)	84

B	23	(27 , 56)	84
B	24	(32 , 56)	84
B	25	(46 , 56)	84
B	26	(52 , 56)	84
B	27	(58 , 56)	84
A	28	(5 , 50)	84
A	29	(10 , 50)	84
A	30	(16 , 50)	84
B, A	31	(19 , 50)	84
B	32	(27 , 50)	84
B	33	(32 , 50)	84
B	34	(37 , 50)	84
B	35	(41 , 50)	84
B	36	(46 , 50)	84
B	37	(52 , 50)	84
B	38	(58 , 50)	84
C, A	39	(5 , 45)	84
C, A	40	(10 , 45)	84
C, A	41	(16 , 45)	84
D,B,C,A	42	(19 , 45)	84
D, B	43	(27 , 45)	84
D, B	44	(32 , 45)	84
D, B	45	(37 , 45)	84
D, B	46	(41 , 45)	84

D, B	47	(46 , 45)	84
D, B	48	(52 , 45)	84
D, B	49	(58 , 45)	84
C	50	(5 , 39)	84
C	51	(10 , 39)	84
C	52	(16 , 39)	84
D	53	(27 , 39)	84
D	54	(32 , 39)	84
D	55	(37 , 39)	84
D	56	(41 , 39)	84
D	57	(46 , 39)	84
D	58	(52 , 39)	84
D	59	(58 , 39)	84
C	60	(5 , 33)	63
C	61	(16 , 33)	63
D, C	62	(19 , 33)	84
D	63	(27 , 33)	84
D	64	(32 , 33)	84
D	65	(37 , 33)	84
D	66	(41 , 33)	84
D	67	(46 , 33)	84
D	68	(52 , 33)	84
D	69	(58 , 33)	84
C	70	(5 , 27)	84

C	71	(10 , 27)	84
C	72	(16 , 27)	42
D, C	73	(19 , 27)	42
D	74	(27 , 27)	84
D	75	(32 , 27)	84
D	76	(37 , 27)	84
D	77	(41 , 27)	84
D	78	(46 , 27)	84
D	79	(52 , 27)	84
D	80	(58 , 27)	84
C	81	(5 , 21)	84
C	82	(10 , 21)	84
C	83	(16 , 21)	84
D, C	84	(19 , 21)	84
D	85	(27 , 21)	42
D	86	(32 , 21)	42
D	87	(37 , 21)	84
D	88	(41 , 21)	84
D	89	(46 , 21)	84
D	90	(52 , 21)	84
D	91	(58 , 21)	84
C	92	(5 , 16)	84
C	93	(10 , 16)	84
C	94	(16 , 16)	84

D, C	95	(19 , 16)	84
D	96	(27 , 16)	84
D	97	(32 , 16)	84
D	98	(37 , 16)	42
D	99	(41 , 16)	42
D	100	(46 , 16)	84
D	101	(52 , 16)	84
D	102	(58 , 16)	84
C	103	(10 , 10)	84
C	104	(16 , 10)	84
D, C	105	(19 , 10)	84
D	106	(27 , 10)	84
D	107	(32 , 10)	84
D	108	(37 , 10)	84
D	109	(41 , 10)	84
C	110	(10 , 4)	84
C	111	(16 , 4)	84
D, C	112	(19 , 4)	84
D	113	(27 , 4)	63
D	114	(32 , 4)	42
D	115	(37 , 4)	21

Points from 116 to 135 do not have population. They correspond to parks or other public facilities which are not concerned in this paper.

1

OINTS OF DEMAND ON Y AXIS	1	10	19	28	39	50	60	70	81	92	132	134
POINT NUMBERS	1	2	3	4	5	6	7	8	9	10	11	12
'EIGHTS ASSOCIATED ITH EACH POINT	84	84	84	84	84	84	63	84	84	84	0	0

$\sum_{i=1}^{6} w_i = 504$

$\sum_{i=7}^{12} w_i = 315$

$\left(\sum_{i=1}^{6} w_i - \sum_{i=7}^{12} w_i\right)\Delta$

$+189\Delta$

$\sum_{i=1}^{5} w_i = 420$

$\sum_{i=6}^{12} w_i = 399$

$+21\Delta$

$\sum_{i=1}^{4} w_i = 336$

$\sum_{i=5}^{12} w_i = 483$

-147Δ

2

POINTS OF DEMAND ON THE x AXIS	134	110	111	112	113	114	115	130	129	135
POINT NUMBERS	1	2	3	4	5	6	7	8	9	10
WEIGHTS ASSOCIATED WITH EACH POINT	0	84	84	84	63	42	21	0	0	0

$\sum_{i=1}^{4} w_i = 252$

$\sum_{i=5}^{10} w_i = 126$

$+126\Delta$

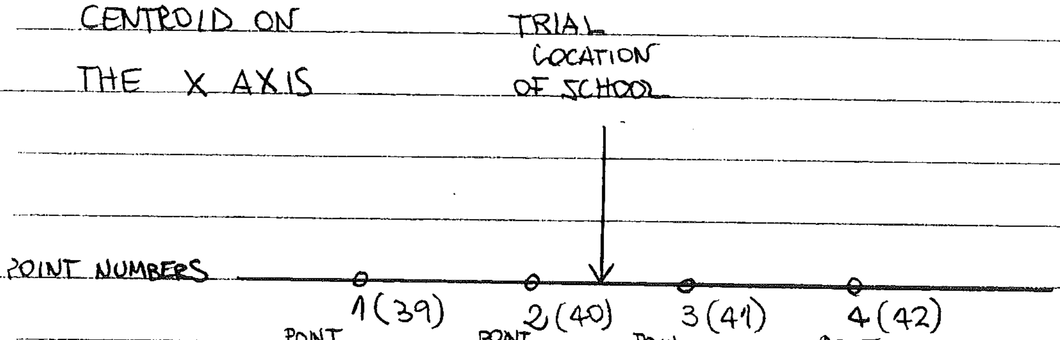
$\sum_{i=1}^{3} w_i = 168$

$\sum_{i=4}^{10} w_i = 210$

-42Δ

AREA A

WEIGHTED CENTROID ON THE X AXIS



WEIGHTS ASSOCIATED WITH EACH POINT

POINT #	WEIGHT	POINT #	WEIGHT	POINT #	WEIGHT	POINT #	WEIGHT
1 –	84	2 –	84	3 –	84	4 –	84
10 –	84	11 –	84	12 –	84	13 –	84
19 –	84	20 –	84	21 –	84	22 –	84
28 –	84	29 –	84	30 –	84	31 –	84
39 –	84	40 –	84	41 –	84	42 –	84
TOTAL	420		420		420		420

$$\sum_{i=1}^{2} w_i = 840$$

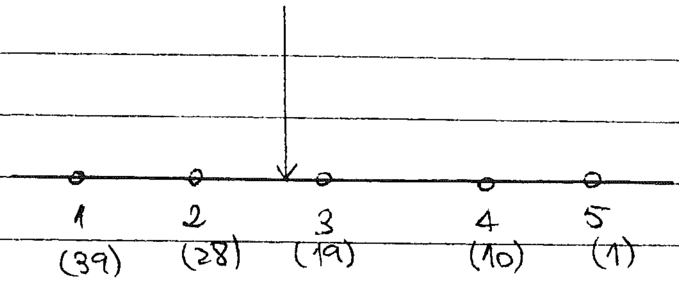
$$\sum_{i=3}^{4} w_i = 840$$

∘ Δ CANDIDATE POINTS TO BE THE CENTROID FOR THE X AXIS: 2(40) or 3(41)

AREA A

WEIGHTED CENTROID ON THE Y AXIS

POINT NUMBERS



WEIGHTS ASSOCIATED WITH EACH POINT

POINT #	W	POINT #	W	POINT #	W
39	84	28	84	19	84
40	84	29	84	20	84
41	84	30	84	21	84
42	84	31	84	22	84
	336		336		336

POINT #	W	POINT #	W
10	84	1	84
11	84	2	84
12	84	3	84
13	84	4	84
	336		336

$$\sum_{i=1}^{2} w_i = 672$$

-336Δ

$$\sum_{i=3}^{5} w_i = 1008$$

CANDIDATE POINT TO BE THE CENTROID FOR THE Y AXIS:
3 (19)

$$\sum_{i=1}^{3} w_i = 1008$$

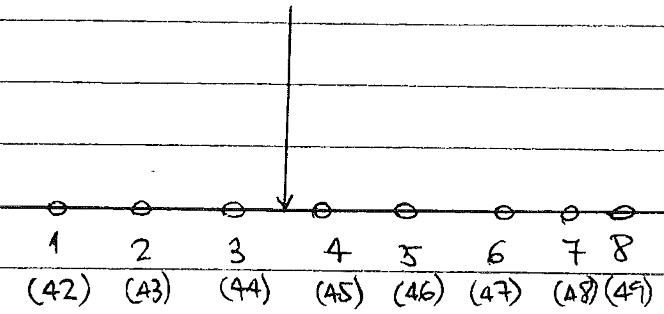
336Δ

$$\sum_{i=4}^{6} w_i = 672$$

AREA B

WEIGHTED CENTROID ON THE X AXIS

POINT NUMBERS



WEIGHT ASSOCIATED WITH EACH POINT

POINT #	WEIGHT	POINT #	W	POINT #	W	POINT #	W
42	84	43	84	44	84	45	84
31	84	32	84	33	84	34	84
22	84	23	84	24	84	–	–
13	84	14	84	15	84	–	–
4	84	5	84	6	84	7	84
	420		420		420		252

$\sum_{i=1}^{[illegible]} w_i = 1260$

-756Δ

$\sum_{:4} w_i = 2016$

POINT #	WEIGHT	POINT #	W	POINT #	W	POINT #	W
46	84	47	84	48	84	49	84
35	84	36	84	37	84	38	84
–	–	25	84	26	84	27	84
–	–	16	84	17	84	18	84
8	84	9	84	–	–	–	–
	252		420		336		336

$\sum_{i=1}^{[illegible]} w_i = 1680$

$+84\Delta$

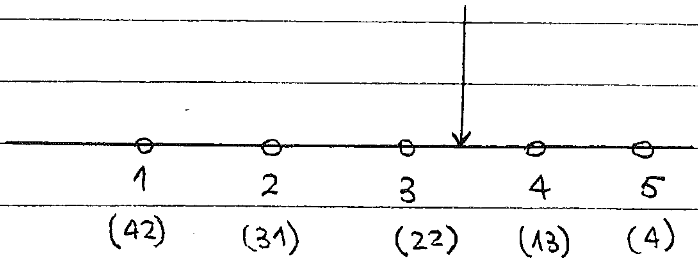
$\sum_{:5} w_i = 1596$

CANDIDATE POINT TO BE THE CENTROID OR THE X AXIS : 4 (45)

AREA B

WEIGHTED CENTROID ON THE Y AXIS

POINT NUMBERS



WEIGHTS ASSOCIATED WITH EACH POINT

POINT #(1)	W	POINT #(2)	W	POINT #(3)	W	POINT #(4)	W
42	84	31	84	22	84	13	84
43	84	32	84	23	84	14	84
44	84	33	84	24	84	15	84
45	84	34	84	-	-	-	-
46	84	35	84	-	-	-	-
47	84	36	84	25	84	16	84
48	84	37	84	26	84	17	84
49	84	38	84	27	84	18	84
	672		672		504		504

POINT #(5)	W
4	84
5	84
6	84
7	84
8	84
9	84
	504

CANDIDATE POINT TO BE THE CENTROID FOR THE Y AXIS : 3 (22)

$\sum_{i=1}^{3} = 1848$

840Δ

$\sum_{i=4}^{5} = 1008$

$\sum_{i=1}^{2} = 1344$

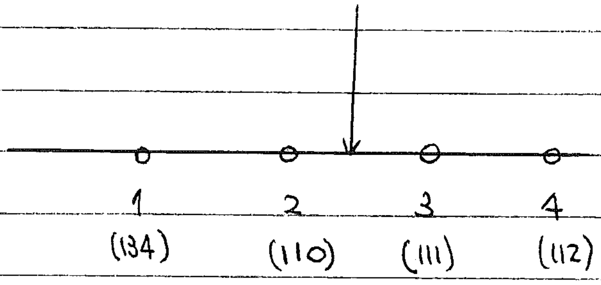
−168Δ

$\sum_{i=3}^{5} = 1512$

AREA C

WEIGTHED CENTROID ON THE X AXIS

POINT NUMBERS



WEIGHTS ASSOCIATED WITH EACH POINT

POINT #(1)	W	POINT #(2)	W	POINT #(3)	W
134	0	110	84	111	84
132	0	103	84	104	84
92	84	93	84	94	84
81	84	82	84	83	84
70	84	71	84	72	42
60	63	-	-	61	63
50	84	51	84	52	84
39	84	40	84	41	84
	483		588		609

POINT #(4)	W
112	84
105	84
95	84
84	84
73	42
62	84
-	-
42	84
	546

CANDIDATE POINT TO BE THE CENTROID FOR THE X AXIS : 3(111).

$\sum_{i=1}^{2} w_i = 1071$

$-84\,\Delta$

$\sum_{i=3}^{4} w_i = 1155$

$\sum_{i=1}^{3} w_i = 1680$

$+1134\,\Delta$

$\sum_{i=4}^{4} w_i = 546$

52

AREA C

WEIGHTED
CENTROID
ON THE Y AXIS

POINT NUMBERS

1	2	3	4	5	6	7	8
(134)	(132)	(92)	(81)	(70)	(60)	(50)	(39)

WEIGHTS ASSOCIATED
WITH EACH POINT

$\sum_{i=1}^{4} w_i = 1176$

126Δ

$\sum_{i=5}^{8} w_i = 1050$

POINT #(1)	W	POINT #(2)	W	POINT #(3)	W
134	-	132	-	92	84
110	84	103	84	93	84
111	84	104	84	94	84
112	84	105	84	95	84
	252		252		336

$\sum_{i=1}^{3} w_i = 840$

−546Δ

$\sum_{i=4}^{8} w_i = 1386$

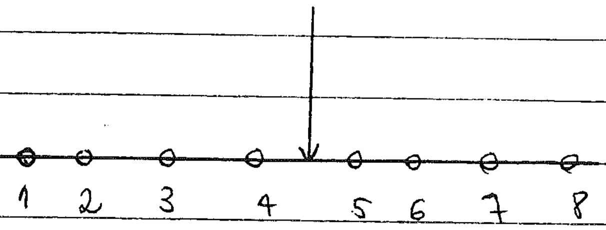
POINT #(4)	W	POINT #(5)	W	POINT #(6)	W	POINT #(7)	W	POINT #(8)	W
81	84	70	84	60	63	50	84	39	84
82	84	71	84	-	-	51	84	40	84
83	84	72	42	61	63	52	84	41	84
84	84	73	42	62	84	-	-	42	84
	336		252		210		252		336

NDIDATE POINT
TO BE THE CENTROID
OR THE Y AXIS:
4(81)

AREA D

WEIGHTED CENTROID ON THE X AXIS

POINT NUMBERS



WEIGHTS ASSOCIATED WITH EACH POINT

$\sum_{i=1}^{4} w_i = 2310$

252Δ

$\sum_{i=5}^{8} w_i = 2058$

$\sum_{i=1}^{3} w_i = 1743$

$\sum_{i=4}^{8} w_i = 2625$

−882Δ

POINT #(1)	W	POINT #(2)	W	POINT #(3)	W	POINT #(4)	W
112	84	113	63	114	42	115	21
105	84	106	84	107	84	108	84
95	84	96	84	97	84	98	42
84	84	85	42	86	42	87	84
73	42	74	84	75	84	76	84
62	84	63	84	64	84	65	84
–	–	53	84	54	84	55	84
42	84	43	84	44	84	45	84
TOTAL	546		609		588		567

POINT #(5)	W	POINT #(6)	W	POINT #(7)	W	POINT #(8)	W
–	–	–	–	–	–	–	–
109	84	–	–	–	–	–	–
99	42	100	84	101	84	102	84
88	84	89	84	90	84	91	84
77	84	78	84	79	84	80	84
66	84	67	84	68	84	69	84
56	84	57	84	58	84	59	84
46	84	47	84	48	84	49	84
	546		504		504		504

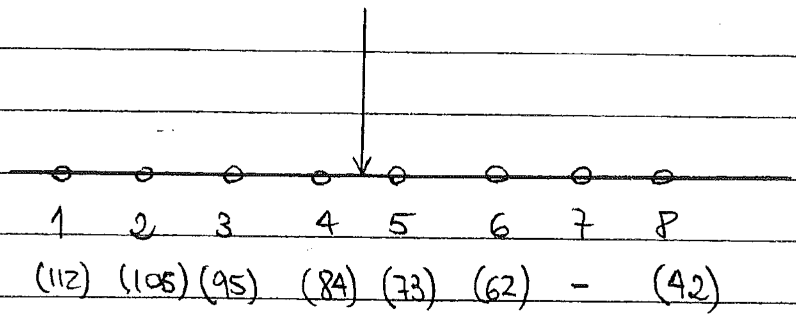
CANDIDATE POINT TO BE THE CENTROID FOR THE X AXIS:

4(115)

AREA D

WEIGHTED
CENTROID ON THE
Y AXIS

POINT NUMBERS



WEIGHTS ASSOCIATED
WITH EACH POINT

POINT #(1)	W	POINT #(2)	W	POINT #(3)	W	POINT #(4)	W
112	84	105	84	95	84	84	84
113	63	106	84	96	84	85	42
114	42	107	84	97	84	86	42
115	21	108	84	98	42	87	84
-		109	84	99	42	88	84
-		-		100	84	89	84
-		-		101	84	90	84
-		-		102	84	91	84
	210		420		588		588

$\sum_{i=1}^{4} w_i = 1806$

−756Δ

$\sum_{i=5}^{8} w_i = 2562$

$\sum_{i=1}^{5} w_i = 2436$

$\sum_{i=6}^{8} w_i = 1932$

504Δ

POINT#	W	POINT#	W	POINT#	W	POINT#	W
73	42	62	84	-	-	42	84
74	84	63	84	53	84	43	84
75	84	64	84	54	84	44	84
76	84	65	84	55	84	45	84
77	84	66	84	56	84	46	84
78	84	67	84	57	84	47	84
79	84	68	84	58	84	48	84
80	84	69	84	59	84	49	84
	630		672		588		67[illegible]

CANDIDATE POINT
TO BE THE CENTROID
FOR THE Y AXIS:
5(73)

AREA A

WEIGHTED DISTANCES

FACILITY \ DEMAND	1	2	3	4	10	11	12	13	19	20	21	22	28	29	30	31	39	40	41	42	TOTAL
1	0	420	924	1176	588	1008	1512	1764	1092	1512	2016	2268	1596	2016	2520	2772	2016	2436	2940	3192	33768
2	420	0	504	756	1008	588	1092	1344	1512	1092	1596	1848	2016	1596	2100	2352	2436	2016	2520	2772	29568
3	924	504	0	252	1512	1092	588	840	2016	1596	1092	1344	2520	2100	1596	1848	2940	2520	2016	2268	29568
4	1176	756	252	0	1764	1344	840	588	2268	1848	1344	1092	2772	2352	1848	1596	3192	2772	2268	2016	32088
10	588	1008	1512	1764	0	420	924	1176	504	924	1428	1680	1008	1428	1932	2184	1428	1848	2352	2604	27712
11	1008	588	1092	1344	420	0	504	756	924	504	1008	1260	1428	1008	1512	1764	1848	1428	1932	2184	22512
12	1512	1092	588	840	924	504	0	252	1428	1008	504	756	1932	1512	1008	1260	2352	1932	1428	1680	22512
13	1764	1344	840	588	1176	756	252	0	1680	1260	756	504	2184	1764	1260	1008	2604	2184	1680	1428	25032
19	1092	1512	2016	2268	504	924	1428	1680	0	420	924	1176	504	924	1428	1680	924	1344	1848	2100	24696
20	1512	1092	1596	1848	924	504	1008	1260	420	0	504	756	924	504	1008	1260	1344	924	1428	1680	20496 ⇐
21	2016	1596	1092	1344	1428	1008	504	756	924	504	0	252	1428	1008	504	756	1848	1428	924	1176	20496 ⇐
22	2268	1848	1344	1092	1680	1260	756	504	1176	756	252	0	1680	1260	756	504	2100	1680	1176	924	23016
28	1596	2016	2520	2772	1008	1428	1932	2184	504	924	1428	1680	0	420	924	1176	420	840	1344	1596	26712
29	2016	1596	2100	2352	1428	1008	1512	1764	924	504	1008	1260	420	0	504	756	840	420	924	1176	22512
30	2520	2100	1596	1848	1932	1512	1008	1260	1428	1008	504	756	924	504	0	252	1344	924	420	672	22512
31	2772	2352	1848	1596	2184	1764	1260	1008	1680	1260	756	504	1176	756	252	0	1596	1176	672	420	25032
39	2016	2436	2940	3192	1428	1848	2352	2604	924	1344	1848	2100	420	840	1344	1596	0	420	924	1176	31752
40	2436	2016	2520	2772	1848	1428	1932	2184	1344	924	1428	1680	840	420	924	1176	420	0	504	756	27552
41	2940	2520	2016	2268	2352	1932	1428	1680	1848	1428	924	1176	1344	924	420	672	924	504	0	252	27552
42	3192	2772	2268	2016	2604	2184	1680	1428	2100	1680	1176	924	1596	1176	672	420	1176	756	252	0	30072

AREA A

pascal 0000/2.0 aaec (27jul80)

```
0680 --     program rdat01 (input,output);
06a8 --     const RECS = 20;        (*  number of records in data *)
06a8 --           LIST = 0;         (*  0 for a long list or 1 to supress it *)
06a8 --     var n,i,j,k,abs1,min : integer;
06c0 --         minx1,miny1,minx2,miny2 : integer;
06d0 --         sum,x,y,w : array (.1..RECS.) of integer;
0810 -- a   function abs (x:integer) : integer;
!
:
0018 --        if x<0 then abs:=-x
002a --               else abs:= x;
004c -0 a   end;
0000 0-     begin
0062 --        min:=32676;
006a --        for i:=1 to RECS do
0088 --          readln (n,x(.i.),y(.i.),w(.i.));
011c --        for i:=1 to RECS do
013a 1-          begin
013a --          write (' Record#',i:1,'  point1= (');
0176 --          writeln (x(.i.):1,',',y(.i.):1,')');
023a --          if LIST = 0
023a 2-            then begin
0244 --                 write   ('      Point2');
025c --                 write   ('   Population');
0274 --                 writeln ('        Result');
0292 -2                 end;
0292 --          sum(.i.) := 0;
02aa --          for j:=1 to RECS do
02c8 2-            begin
02c8 --            abs1:=abs(x(.i.)-x(.j.));
0312 --            abs1:=abs1+abs(y(.i.)-y(.j.));
0368 --            abs1:=abs1*w(.j.);
039a --            if (LIST = 0)
03aa 3-              then begin
03a4 --                   write ('      ');
03bc --                   write ('(',x(.j.):2,',',y(.j.):2,')     ');
048c --                   writeln (w(.j.):6,'          ',abs1:6);
04ec -3                   end;
04ec --            sum(.i.) := sum(.i.) + abs1;
052a -2            end;
054c --          writeln ('                            sum = ',sum(.i.):7);
0594 --          if min > sum(.i.)
05b0 2-            then begin
05c0 --                 min:=sum(.i.);
05e0 --                 minx1:=x(.i.);
0600 --                 miny1:=y(.i.);
0620 -2                 end;
0620 --          writeln;
062c -1          end;
064e --        writeln;
065a --        writeln (' The Minimum is ',min);
0690 --        writeln ('  At Point1 = (',minx1:1,',',miny1:1,')');
0742 -0     end.
```

aaec pascal 2.0 compilation concluded

no errors detected in pascal program

loader

program size - 58e8 (hex), 22,760 (decimal).
Record#1 point1= (5,69)

Point2	Population	Result
(5,69)	84	0
(10,69)	84	420
(16,69)	84	924
(19,69)	84	1176
(5,62)	84	588
(10,62)	84	1008
(16,62)	84	1512
(19,62)	84	1764
(5,56)	84	1092
(10,56)	84	1512
(16,56)	84	2016
(19,56)	84	2268
(5,50)	84	1596
(10,50)	84	2016
(16,50)	84	2520
(19,50)	84	2772
(5,45)	84	2016
(10,45)	84	2436
(16,45)	84	2940
(19,45)	84	3192
	sum =	33768

AREA A

Record#2 point1= (10,69)

Point2	Population	Result
(5,69)	84	420
(10,69)	84	0
(16,69)	84	504
(19,69)	84	756
(5,62)	84	1008
(10,62)	84	588
(16,62)	84	1092
(19,62)	84	1344
(5,56)	84	1512
(10,56)	84	1092
(16,56)	84	1596
(19,56)	84	1848
(5,50)	84	2016
(10,50)	84	1596
(16,50)	84	2100
(19,50)	84	2352
(5,45)	84	2436
(10,45)	84	2016
(16,45)	84	2520
(19,45)	84	2772
	sum =	29568

Record#3 point1= (16,69)

Point2	Population	Result
(5,69)	84	924
(10,69)	84	504
(16,69)	84	0
(19,69)	84	252
(5,62)	84	1512
(10,62)	84	1092
(16,62)	84	588
(19,62)	84	840
(5,56)	84	2016
(10,56)	84	1596
(16,56)	84	1092
(19,56)	84	1344
(5,50)	84	2520
(10,50)	84	2100
(16,50)	84	1596
(19,50)	84	1848
(5,45)	84	2940

AREA A

(10,45)	84	2520
(16,45)	84	2016
(19,45)	84	2268
	sum =	29568

Record#4 point1= (19,69)

Point2	Population	Result
(5,69)	84	1176
(10,69)	84	756
(16,69)	84	252
(19,69)	84	0
(5,62)	84	1764
(10,62)	84	1344
(16,62)	84	840
(19,62)	84	588
(5,56)	84	2268
(10,56)	84	1848
(16,56)	84	1344
(19,56)	84	1092
(5,50)	84	2772
(10,50)	84	2352
(16,50)	84	1848
(19,50)	84	1596
(5,45)	84	3192
(10,45)	84	2772
(16,45)	84	2268
(19,45)	84	2016
	sum =	32088

Record#5 point1= (5,62)

Point2	Population	Result
(5,69)	84	588
(10,69)	84	1008
(16,69)	84	1512
(19,69)	84	1764
(5,62)	84	0
(10,62)	84	420
(16,62)	84	924
(19,62)	84	1176
(5,56)	84	504
(10,56)	84	924
(16,56)	84	1428
(19,56)	84	1680
(5,50)	84	1008
(10,50)	84	1428
(16,50)	84	1932
(19,50)	84	2184
(5,45)	84	1428
(10,45)	84	1848
(16,45)	84	2352
(19,45)	84	2604
	sum =	26712

Record#6 point1= (10,62)

Point2	Population	Result
(5,69)	84	1008
(10,69)	84	588
(16,69)	84	1092
(19,69)	84	1344
(5,62)	84	420
(10,62)	84	0
(16,62)	84	504
(19,62)	84	756
(5,56)	84	924
(10,56)	84	504
(16,56)	84	1008

AREA A

(19,56)	84	1260
(5,50)	84	1428
(10,50)	84	1008
(16,50)	84	1512
(19,50)	84	1764
(5,45)	84	1848
(10,45)	84	1428
(16,45)	84	1932
(19,45)	84	2184
	sum =	22512

Record#7 point1= (16,62)

Point2	Population	Result
(5,69)	84	1512
(10,69)	84	1092
(16,69)	84	588
(19,69)	84	840
(5,62)	84	924
(10,62)	84	504
(16,62)	84	0
(19,62)	84	252
(5,56)	84	1428
(10,56)	84	1008
(16,56)	84	504
(19,56)	84	756
(5,50)	84	1932
(10,50)	84	1512
(16,50)	84	1008
(19,50)	84	1260
(5,45)	84	2352
(10,45)	84	1932
(16,45)	84	1428
(19,45)	84	1680
	sum =	22512

Record#8 point1= (19,62)

Point2	Population	Result
(5,69)	84	1764
(10,69)	84	1344
(16,69)	84	840
(19,69)	84	588
(5,62)	84	1176
(10,62)	84	756
(16,62)	84	252
(19,62)	84	0
(5,56)	84	1680
(10,56)	84	1260
(16,56)	84	756
(19,56)	84	504
(5,50)	84	2184
(10,50)	84	1764
(16,50)	84	1260
(19,50)	84	1008
(5,45)	84	2604
(10,45)	84	2184
(16,45)	84	1680
(19,45)	84	1428
	sum =	25032

Record#9 point1= (5,56)

Point2	Population	Result
(5,69)	84	1092
(10,69)	84	1512
(16,69)	84	2016
(19,69)	84	2268
(5,62)	84	504

AREA A

(10,62)	84	924
(16,62)	84	1428
(19,62)	84	1680
(5,56)	84	0
(10,56)	84	420
(16,56)	84	924
(19,56)	84	1176
(5,50)	84	504
(10,50)	84	924
(16,50)	84	1428
(19,50)	84	1680
(5,45)	84	924
(10,45)	84	1344
(16,45)	84	1848
(19,45)	84	2100
	sum =	24696

Record#10 point1= (10,56)

Point2	Population	Result
(5,69)	84	1512
(10,69)	84	1092
(16,69)	84	1596
(19,69)	84	1848
(5,62)	84	924
(10,62)	84	504
(16,62)	84	1008
(19,62)	84	1260
(5,56)	84	420
(10,56)	84	0
(16,56)	84	504
(19,56)	84	756
(5,50)	84	924
(10,50)	84	504
(16,50)	84	1008
(19,50)	84	1260
(5,45)	84	1344
(10,45)	84	924
(16,45)	84	1428
(19,45)	84	1680
	sum =	20496

Record#11 point1= (16,56)

Point2	Population	Result
(5,69)	84	2016
(10,69)	84	1596
(16,69)	84	1092
(19,69)	84	1344
(5,62)	84	1428
(10,62)	84	1008
(16,62)	84	504
(19,62)	84	756
(5,56)	84	924
(10,56)	84	504
(16,56)	84	0
(19,56)	84	252
(5,50)	84	1428
(10,50)	84	1008
(16,50)	84	504
(19,50)	84	756
(5,45)	84	1848
(10,45)	84	1428
(16,45)	84	924
(19,45)	84	1176
	sum =	20496

Record#12 point1= (19,56)

AREA A

Point2	Population	Result
(5,69)	84	2268
(10,69)	84	1848
(16,69)	84	1344
(19,69)	84	1092
(5,62)	84	1680
(10,62)	84	1260
(16,62)	84	756
(19,62)	84	504
(5,56)	84	1176
(10,56)	84	756
(16,56)	84	252
(19,56)	84	0
(5,50)	84	1680
(10,50)	84	1260
(16,50)	84	756
(19,50)	84	504
(5,45)	84	2100
(10,45)	84	1680
(16,45)	84	1176
(19,45)	84	924
	sum =	23016

Record#13 point1= (5,50)

Point2	Population	Result
(5,69)	84	1596
(10,69)	84	2016
(16,69)	84	2520
(19,69)	84	2772
(5,62)	84	1008
(10,62)	84	1428
(16,62)	84	1932
(19,62)	84	2184
(5,56)	84	504
(10,56)	84	924
(16,56)	84	1428
(19,56)	84	1680
(5,50)	84	0
(10,50)	84	420
(16,50)	84	924
(19,50)	84	1176
(5,45)	84	420
(10,45)	84	840
(16,45)	84	1344
(19,45)	84	1596
	sum =	26712

Record#14 point1= (10,50)

Point2	Population	Result
(5,69)	84	2016
(10,69)	84	1596
(16,69)	84	2100
(19,69)	84	2352
(5,62)	84	1428
(10,62)	84	1008
(16,62)	84	1512
(19,62)	84	1764
(5,56)	84	924
(10,56)	84	504
(16,56)	84	1008
(19,56)	84	1260
(5,50)	84	420
(10,50)	84	0
(16,50)	84	504
(19,50)	84	756
(5,45)	84	840

AREA A

[illegible]	[illegible]	[illegible]
(16,45)	84	924
(19,45)	84	1176
	sum =	22512

Record#15 point1= (16,50)

Point2	Population	Result
(5,69)	84	2520
(10,69)	84	2100
(16,69)	84	1596
(19,69)	84	1848
(5,62)	84	1932
(10,62)	84	1512
(16,62)	84	1008
(19,62)	84	1260
(5,56)	84	1428
(10,56)	84	1008
(16,56)	84	504
(19,56)	84	756
(5,50)	84	924
(10,50)	84	504
(16,50)	84	0
(19,50)	84	252
(5,45)	84	1344
(10,45)	84	924
(16,45)	84	420
(19,45)	84	672
	sum =	22512

Record#16 point1= (19,50)

Point2	Population	Result
(5,69)	84	2772
(10,69)	84	2352
(16,69)	84	1848
(19,69)	84	1596
(5,62)	84	2184
(10,62)	84	1764
(16,62)	84	1260
(19,62)	84	1008
(5,56)	84	1680
(10,56)	84	1260
(16,56)	84	756
(19,56)	84	504
(5,50)	84	1176
(10,50)	84	756
(16,50)	84	252
(19,50)	84	0
(5,45)	84	1596
(10,45)	84	1176
(16,45)	84	672
(19,45)	84	420
	sum =	25032

Record#17 point1= (5,45)

Point2	Population	Result
(5,69)	84	2016
(10,69)	84	2436
(16,69)	84	2940
(19,69)	84	3192
(5,62)	84	1428
(10,62)	84	1848
(16,62)	84	2352
(19,62)	84	2604
(5,56)	84	924
(10,56)	84	1344
(16,56)	84	1848

AREA A

[illegible]	84	[illegible]
(5,50)	84	420
(10,50)	84	840
(16,50)	84	1344
(19,50)	84	1596
(5,45)	84	0
(10,45)	84	420
(16,45)	84	924
(19,45)	84	1176
	sum =	31752

Record#18 point1= (10,45)

Point2	Population	Result
(5,69)	84	2436
(10,69)	84	2016
(16,69)	84	2520
(19,69)	84	2772
(5,62)	84	1848
(10,62)	84	1428
(16,62)	84	1932
(19,62)	84	2184
(5,56)	84	1344
(10,56)	84	924
(16,56)	84	1428
(19,56)	84	1680
(5,50)	84	840
(10,50)	84	420
(16,50)	84	924
(19,50)	84	1176
(5,45)	84	420
(10,45)	84	0
(16,45)	84	504
(19,45)	84	756
	sum =	27552

Record#19 point1= (16,45)

Point2	Population	Result
(5,69)	84	2940
(10,69)	84	2520
(16,69)	84	2016
(19,69)	84	2268
(5,62)	84	2352
(10,62)	84	1932
(16,62)	84	1428
(19,62)	84	1680
(5,56)	84	1848
(10,56)	84	1428
(16,56)	84	924
(19,56)	84	1176
(5,50)	84	1344
(10,50)	84	924
(16,50)	84	420
(19,50)	84	672
(5,45)	84	924
(10,45)	84	504
(16,45)	84	0
(19,45)	84	252
	sum =	27552

Record#20 point1= (19,45)

Point2	Population	Result
(5,69)	84	3192
(10,69)	84	2772
(16,69)	84	2268
(19,69)	84	2016
(5,62)	84	2604

(10,62)	84	2184
(16,62)	84	1680
(19,62)	84	1423
(5,56)	84	2100
(10,56)	84	1680
(16,56)	84	1176
(19,56)	84	924
(5,50)	84	1596
(10,50)	84	1176
(16,50)	84	672
(19,50)	84	420
(5,45)	84	1176
(10,45)	34	756
(16,45)	84	252
(19,45)	84	0
	sum =	30072

AREA A

```
The Minimum is          20496
  At Point1 = (10,56)
*go
```

ENG FCZ C
4549

CPU MODEL	SERIAL	3090	170596
VM/SP SERVICE LEVEL		0316	
USERID	ORIGIN	ENGFCZC	ENGFCZC
DISTRIBUTION CODE		ENGFCZC	
SPOOL FILE NAME TYPE		ENGFCZC	TERM
CREATION DATE		04/11/86 18:06:58	
PRINT DATE		04/11/86 18:06:59	
SPOOL FILE ID		4029	
RECORD COUNT		0091	
DEVICE	CLASS	0E1E	A
FORMS		STANDARD	

Boston University
Academic Computing Center
111 Cummington Street
Boston, Massachusetts 02215
(617) 353-2780

AREA B

```
 LISTING OF .MPI004B  FRI APR 11, 1986 18:06:59   PAGE 1
/LOAD PASCAL
PROGRAM RDAT01 (INPUT,OUTPUT);
CONST RECS = 33;        (*  NUMBER OF RECORDS IN DATA *)
      LIST = 1;         (*  0 FOR A LONG LIST OR 1 TO SUPRESS IT *)
VAR N,I,J,K,ABSL,MIN : INTEGER;
    MINX1,MINY1,MINX2,MINY2 : INTEGER;
    SUM,X,Y,W : ARRAY (.1..RECS.) OF INTEGER;
FUNCTION ABS (X:INTEGER) : INTEGER;
BEGIN
  IF X<0 THEN ABS:=-X
         ELSE ABS:= X;
END;
BEGIN
   MIN:=99999999;
   FOR I:=1 TO RECS DO
     READLN (N,X(.I.),Y(.I.),W(.I.));
   FOR I:=1 TO RECS DO
     BEGIN
     WRITE (' RECORD#',I:1,'  POINT1= (');
     WRITELN (X(.I.):1,',',Y(.I.):1,')');
     IF LIST = 0
       THEN BEGIN
            WRITE   ('        POINT2');
            WRITE   ('    POPULATION');
            WRITELN ('          RESULT');
            END;
     SUM(.I.) := 0;
     FOR J:=1 TO RECS DO
       BEGIN
       ABSL:=ABS(X(.I.)-X(.J.));
       ABSL:=ABSL+ABS(Y(.I.)-Y(.J.));
       ABSL:=ABSL*W(.J.);
       IF (LIST = 0)
         THEN BEGIN
              WRITE ('        ');
              WRITE ('(',X(.J.):2,',',Y(.J.):2,')      ');
              WRITELN (W(.J.):6,'              ',ABSL:6);
              END;
       SUM(.I.) := SUM(.I.) + ABSL;
       END;
     WRITELN ('                              SUM = ',SUM(.I.):7);
     IF MIN > SUM(.I.)
       THEN BEGIN
            MIN:=SUM(.I.);
            MINX1:=X(.I.);
            MINY1:=Y(.I.);
            END;
     WRITELN;
     END;
   WRITELN;
   WRITELN (' THE MINIMUM IS ',MIN);
   WRITELN ('    AT POINT1 = (',MINX1:1,',',MINY1:1,')');
END.
/DATA
  4      19          69          84
```

AREA B

5	27	69	84
6	32	69	84
7	37	69	84
8	41	69	84
9	46	69	84
13	19	62	84
14	27	62	84
15	32	62	84
16	46	62	84
17	52	62	84
18	58	62	84
22	19	56	84
23	27	56	84
24	32	56	84
25	46	56	84
26	52	56	84
27	58	56	84
31	19	50	84
32	27	50	84
33	32	50	84
34	37	50	84
35	41	50	84
36	46	50	84
37	52	50	84
38	58	50	84
43	27	45	84
44	32	45	84
45	37	45	84
46	41	45	84
47	46	45	84
48	52	45	84
49	58	45	84

/END DATA

END OF FILE .MPIOO4B

RECORD COUNT: 88

AREA B

5	27	69	84
6	32	69	84
7	37	69	84
8	41	69	84
9	46	69	84
13	19	62	84
14	27	62	84
15	32	62	84
16	46	62	84
17	52	62	84
18	58	62	84
22	19	56	84
23	27	56	84
24	32	56	84
25	46	56	84
26	52	56	84
27	58	56	84
31	19	50	84
32	27	50	84
33	32	50	84
34	37	50	84
35	41	50	84
36	46	50	84
37	52	50	84
38	58	50	84
43	27	45	84
44	32	45	84
45	37	45	84
46	41	45	84
47	46	45	84
48	52	45	84
49	58	45	84

/END DATA

END OF FILE .MPIOO4B RECORD COUNT: 88

AREA B


```
0680 --     PROGRAM RDATO1 (INPUT,OUTPUT);
06A8 --     CONST RECS = 33;        (*  NUMBER OF RECORDS IN DATA *)
06A8 --           LIST = 1;         (*  O FOR A LONG LIST OR 1 TO SUPRESS IT *)
06A8 --     VAR N,I,J,K,ABSL,MIN : INTEGER;
06C0 --         MINX1,MINY1,MINX2,MINY2 : INTEGER;
06D0 --         SUM,X,Y,W : ARRAY (.1..RECS.) OF INTEGER;
08E0 -- A   FUNCTION ABS (X:INTEGER) : INTEGER;
0000 O- A   BEGIN
0018 --       IF X<O THEN ABS:=-X
002A --              ELSE ABS:= X;
004C -O A   END;
0000 O-     BEGIN
0062 --        MIN:=99999999;
006A --        FOR I:=1 TO RECS DO
0088 --          READLN (N,X(.I.),Y(.I.),W(.I.));
011C --        FOR I:=1 TO RECS DO
013A 1-          BEGIN
013A --          WRITE (' RECORD#',I:1,'  POINT1= (');
0176 --          WRITELN (X(.I.):1,',',Y(.I.):1,')');
023A --          IF LIST = O
023A 2-            THEN BEGIN
0246 --                 WRITE   ('       POINT2');
025E --                 WRITE   ('   POPULATION');
0276 --                 WRITELN ('          RESULT');
0294 -2                 END;
0294 --          SUM(.I.) := O;
02AC --          FOR J:=1 TO RECS DO
02CA 2-            BEGIN
02CA --            ABSL:=ABS(X(.I.)-X(.J.));
0314 --            ABSL:=ABSL+ABS(Y(.I.)-Y(.J.));
036A --            ABSL:=ABSL*W(.J.);
039C --            IF (LIST = O)
03AE 3-              THEN BEGIN
03A8 --                   WRITE ('       ');
03C0 --                   WRITE ('(',X(.J.):2,',',Y(.J.):2,')     ');
0490 --                   WRITELN (W(.J.):6,'             ',ABSL:6);
04F0 -3                   END;
04F0 --            SUM(.I.) := SUM(.I.) + ABSL;
052E -2            END;
0550 --          WRITELN ('                                  SUM = ',SUM(.I.):7);
0598 --          IF MIN > SUM(.I.)
05B4 2-            THEN BEGIN
05C4 --                 MIN:=SUM(.I.);
05E4 --                 MINX1:=X(.I.);
0604 --                 MINY1:=Y(.I.);
0624 -2                 END;
0624 --          WRITELN;
0630 -1          END;
0652 --        WRITELN;
065E --        WRITELN (' THE MINIMUM IS ',MIN);
0694 --        WRITELN ('    AT POINT1 = (',MINX1:1,',',MINY1:1,')');
0746 -O     END.
```

AAEC PASCAL 2.0 COMPILATION CONCLUDED

NO ERRORS DETECTED IN PASCAL PROGRAM

```
LOADER

PROGRAM SIZE - 58F0 (HEX), 22,768 (DECIMAL).
RECORD#1   POINT1= (19,69)
                   SUM =    91392

RECORD#2   POINT1= (27,69)
                   SUM =    74592

RECORD#3   POINT1= (32,69)
                   SUM =    68292

RECORD#4   POINT1= (37,69)
                   SUM =    66192

RECORD#5   POINT1= (41,69)
                   SUM =    66528

RECORD#6   POINT1= (46,69)
                   SUM =    69468

RECORD#7   POINT1= (19,62)
                   SUM =    79044

RECORD#8   POINT1= (27,62)
                   SUM =    62244

RECORD#9   POINT1= (32,62)
                   SUM =    55944

RECORD#10   POINT1= (46,62)
                   SUM =    57120

RECORD#11   POINT1= (52,62)
                   SUM =    65688

RECORD#12   POINT1= (58,62)
                   SUM =    78288

RECORD#13   POINT1= (19,56)
                   SUM =    74508

RECORD#14   POINT1= (27,56)
                   SUM =    57708

RECORD#15   POINT1= (32,56)
                   SUM =    51408

RECORD#16   POINT1= (46,56)
                   SUM =    52584

RECORD#17   POINT1= (52,56)
                   SUM =    61152

RECORD#18   POINT1= (58,56)
                   SUM =    73752

RECORD#19   POINT1= (19,50)
                   SUM =    76020

RECORD#20   POINT1= (27,50)
```

ENG FCZ C
4548

CPU MODEL	SERIAL	3090	170596
VM/SP SERVICE LEVEL		0316	
USERID	ORIGIN	ENGFCZC	BATCH004
DISTRIBUTION CODE		BATCH004	
SPOOL FILE NAME TYPE		ENGFCZC	J0017
CREATION DATE		04/11/86	18:06:46
PRINT DATE		04/11/86	18:06:49
SPOOL FILE ID		4028	
RECORD COUNT		0167	
DEVICE	CLASS	0E1E	A
FORMS		STANDARD	

Boston University
Academic Computing Center
111 Cummington Street
Boston, Massachusetts 02215
(617) 353-2780


```
START: JOB 0017 18:06:45 FRI APR 11, 1986   BATCH004
 /ID ENGFCZC   RMT Q J0017    30S
 /LOAD PASCAL
```

AREA B

```
                         SUM =    59220
RECORD#21   POINT1= (32,50)
                         SUM =    52920
RECORD#22   POINT1= (37,50)
                         SUM =    50820
RECORD#23   POINT1= (41,50)
                         SUM =    51156
RECORD#24   POINT1= (46,50)
                         SUM =    54096
RECORD#25   POINT1= (52,50)
                         SUM =    62664
RECORD#26   POINT1= (58,50)
                         SUM =    75264
RECORD#27   POINT1= (27,45)
                         SUM =    67200
RECORD#28   POINT1= (32,45)
                         SUM =    60900
RECORD#29   POINT1= (37,45)
                         SUM =    58800
RECORD#30   POINT1= (41,45)
                         SUM =    59136
RECORD#31   POINT1= (46,45)
                         SUM =    62076
RECORD#32   POINT1= (52,45)
                         SUM =    70644
RECORD#33   POINT1= (58,45)
                         SUM =    83244

THE MINIMUM IS          50820
  AT POINT1 = (37,50)


END: JOB 0017 18:08:49 FRI APR 11, 1986 COST=$0.26/$0.00 BALANCE=$219.11/$0.00
     CPU=0.12 IO=154 CDS_IN=0 CDS_OUT=0 PAGES=2 TAPE_MTS=0 TAPE_RES=0:00:00
```

ENG FCZ C
4555

CPU MODEL	SERIAL	3090	170596
VM/SP SERVICE LEVEL		0316	
USERID	ORIGIN	ENGFCZC	ENGFCZC
DISTRIBUTION CODE		ENGFCZC	
SPOOL FILE NAME TYPE		ENGFCZC	TERM
CREATION DATE		04/11/86 18:08:20	
PRINT DATE		04/11/86 18:08:24	
SPOOL FILE ID		4039	
RECORD COUNT		0087	
DEVICE	CLASS	OE1E	A
FORMS		STANDARD	

Boston University
Academic Computing Center
111 Cummington Street
Boston, Massachusetts 02215
(617) 353-2780

AREA C

```
 LISTING OF .MPIOO4C  FRI APR 11, 1986 18:08:20   PAGE 1
/LOAD PASCAL
PROGRAM RDATO1 (INPUT,OUTPUT);
CONST RECS = 28;        (*  NUMBER OF RECORDS IN DATA *)
      LIST = 1;         (*  O FOR A LONG LIST OR 1 TO SUPRESS IT *)
VAR N,I,J,K,ABSL,MIN : INTEGER;
    MINX1,MINY1,MINX2,MINY2 : INTEGER;
    SUM,X,Y,W : ARRAY (.1..RECS.) OF INTEGER;
FUNCTION ABS (X:INTEGER) : INTEGER;
BEGIN
  IF X<O THEN ABS:=-X
         ELSE ABS:= X;
END;
BEGIN
   MIN:=999999;
   FOR I:=1 TO RECS DO
     READLN (N,X(.I.),Y(.I.),W(.I.));
   FOR I:=1 TO RECS DO
     BEGIN
     WRITE (' RECORD#',I:1,'  POINT1= (');
     WRITELN (X(.I.):1,',',Y(.I.):1,')');
     IF LIST = O
       THEN BEGIN
            WRITE    ('       POINT2');
            WRITE    ('   POPULATION');
            WRITELN ('        RESULT');
            END;
     SUM(.I.) := O;
     FOR J:=1 TO RECS DO
       BEGIN
       ABSL:=ABS(X(.I.)-X(.J.));
       ABSL:=ABSL+ABS(Y(.I.)-Y(.J.));
       ABSL:=ABSL*W(.J.);
       IF (LIST = O)
         THEN BEGIN
              WRITE ('       ');
              WRITE ('(',X(.J.):2,',',Y(.J.):2,')     ');
              WRITELN (W(.J.):6,'          ',ABSL:6);
              END;
       SUM(.I.) := SUM(.I.) + ABSL;
       END;
     WRITELN ('                               SUM = ',SUM(.I.):7);
     IF MIN > SUM(.I.)
       THEN BEGIN
            MIN:=SUM(.I.);
            MINX1:=X(.I.);
            MINY1:=Y(.I.);
            END;
     WRITELN;
     END;
   WRITELN;
   WRITELN (' THE MINIMUM IS ',MIN);
   WRITELN ('   AT POINT1 = (',MINX1:1,',',MINY1:1,')');
END.
/DATA
 39      5          45          84
```

AREA C

40	10	45	84
41	16	45	84
42	19	45	84
50	5	39	84
51	10	39	84
52	16	39	84
60	5	33	63
61	16	33	63
62	19	33	84
70	5	27	84
71	10	27	84
72	16	27	42
73	19	27	42
81	5	21	84
82	10	21	84
83	16	21	84
84	19	21	84
92	5	16	84
93	10	16	84
94	16	16	84
95	19	16	84
103	10	10	84
104	16	10	84
105	19	10	84
106	27	10	84
110	10	4	84
111	16	4	84
112	19	4	84

/END DATA

END OF FILE .MPIOO4C RECORD COUNT: 84

AREA C

40	10	45	84
41	16	45	84
42	19	45	84
50	5	39	84
51	10	39	84
52	16	39	84
60	5	33	63
61	16	33	63
62	19	33	84
70	5	27	84
71	10	27	84
72	16	27	42
73	19	27	42
81	5	21	84
82	10	21	84
83	16	21	84
84	19	21	84
92	5	16	84
93	10	16	84
94	16	16	84
95	19	16	84
103	10	10	84
104	16	10	84
105	19	10	84
106	27	10	84
110	10	4	84
111	16	4	84
112	19	4	84

/END DATA

END OF FILE .MPIOO4C RECORD COUNT: 84

AREA C

PASCAL 8000/2.0 AAEC (27JUL80) 11 APR 86 AT 18:08:10 PAGE 1

```
0680 --      PROGRAM RDAT01 (INPUT,OUTPUT);
06A8 --      CONST RECS = 28;        (*  NUMBER OF RECORDS IN DATA *)
06A8 --            LIST = 1;         (*  0 FOR A LONG LIST OR 1 TO SUPRESS IT *)
06A8 --      VAR N,I,J,K,ABSL,MIN : INTEGER;
06C0 --          MINX1,MINY1,MINX2,MINY2 : INTEGER;
06D0 --          SUM,X,Y,W : ARRAY (.1..RECS.) OF INTEGER;
0890 -- A    FUNCTION ABS (X:INTEGER) : INTEGER;
0000 0- A    BEGIN
0018 --        IF X<0 THEN ABS:=-X
002A --               ELSE ABS:= X;
004C -0 A    END;
0000 0-      BEGIN
0062 --         MIN:=999999;
006A --         FOR I:=1 TO RECS DO
0088 --           READLN (N,X(.I.),Y(.I.),W(.I.));
011C --         FOR I:=1 TO RECS DO
013A 1-           BEGIN
013A --           WRITE (' RECORD#',I:1,'  POINT1= (');
0176 --           WRITELN (X(.I.):1,',',Y(.I.):1,')');
023A --           IF LIST = 0
023A 2-             THEN BEGIN
0246 --                    WRITE    ('        POINT2');
025E --                    WRITE    ('   POPULATION');
0276 --                    WRITELN ('          RESULT');
0294 -2                    END;
0294 --           SUM(.I.) := 0;
02AC --           FOR J:=1 TO RECS DO
02CA 2-             BEGIN
02CA --             ABSL:=ABS(X(.I.)-X(.J.));
0314 --             ABSL:=ABSL+ABS(Y(.I.)-Y(.J.));
036A --             ABSL:=ABSL*W(.J.);
039C --             IF (LIST = 0)
03AE 3-               THEN BEGIN
03A8 --                      WRITE ('      ');
03C0 --                      WRITE ('(',X(.J.):2,',',Y(.J.):2,')      ');
0490 --                      WRITELN (W(.J.):6,'          ',ABSL:6);
04F0 -3                      END;
04F0 --             SUM(.I.) := SUM(.I.) + ABSL;
052E -2             END;
0550 --           WRITELN ('                                   SUM = ',SUM(.I.):7);
0598 --           IF MIN > SUM(.I.)
05B4 2-             THEN BEGIN
05C4 --                    MIN:=SUM(.I.);
05E4 --                    MINX1:=X(.I.);
0604 --                    MINY1:=Y(.I.);
0624 -2                    END;
0624 --           WRITELN;
0630 -1           END;
0652 --         WRITELN;
065E --         WRITELN (' THE MINIMUM IS ',MIN);
0694 --         WRITELN ('   AT POINT1 = (',MINX1:1,',',MINY1:1,')');
0746 -0      END.
```

AAEC PASCAL 2.0 COMPILATION CONCLUDED

NO ERRORS DETECTED IN PASCAL PROGRAM

LOADER

PROGRAM SIZE - 58F0 (HEX), 22,768 (DECIMAL).

```
RECORD#1  POINT1= (5,45)
                    SUM =    62979
RECORD#2  POINT1= (10,45)
                    SUM =    56679
RECORD#3  POINT1= (16,45)
                    SUM =    56175
RECORD#4  POINT1= (19,45)
                    SUM =    59577
RECORD#5  POINT1= (5,39)
                    SUM =    53655
RECORD#6  POINT1= (10,39)
                    SUM =    47355
RECORD#7  POINT1= (16,39)
                    SUM =    46851
RECORD#8  POINT1= (5,33)
                    SUM =    47355
RECORD#9  POINT1= (16,33)
                    SUM =    40551
RECORD#10  POINT1= (19,33)
                    SUM =    43953
RECORD#11  POINT1= (5,27)
                    SUM =    43575
RECORD#12  POINT1= (10,27)
                    SUM =    37275
RECORD#13  POINT1= (16,27)
                    SUM =    36771
RECORD#14  POINT1= (19,27)
                    SUM =    40173
RECORD#15  POINT1= (5,21)
                    SUM =    42819
RECORD#16  POINT1= (10,21)
                    SUM =    36519
RECORD#17  POINT1= (16,21)
                    SUM =    36015
RECORD#18  POINT1= (19,21)
                    SUM =    39417
RECORD#19  POINT1= (5,16)
                    SUM =    45549
RECORD#20  POINT1= (10,16)
```

ENG FCZ C
4554

CPU MODEL SERIAL	3090 170596
VM/SP SERVICE LEVEL	0316
USERID ORIGIN	ENGFCZC BATCH004
DISTRIBUTION CODE	BATCH004
SPOOL FILE NAME TYPE	ENGFCZC J0019
CREATION DATE	04/11/86 18:08:10
PRINT DATE	04/11/86 18:08:19
SPOOL FILE ID	4038
RECORD COUNT	0152
DEVICE CLASS	0E1E A
FORMS	STANDARD

Boston University
Academic Computing Center
111 Cummington Street
Boston, Massachusetts 02215
(617) 353-2780

```
START: JOB 0019 18:08:09 FRI APR 11, 1986   BATCH004
 /ID ENGFCZC    RMT Q J0019    30S
 /LOAD PASCAL
```

AREA C

```
                          SUM =    39249
RECORD#21   POINT1= (16,16)
                          SUM =    38745
RECORD#22   POINT1= (19,16)
                          SUM =    42147
RECORD#23   POINT1= (10,10)
                          SUM =    46557
RECORD#24   POINT1= (16,10)
                          SUM =    46053
RECORD#25   POINT1= (19,10)
                          SUM =    49455
RECORD#26   POINT1= (27,10)
                          SUM =    65919
RECORD#27   POINT1= (10,4)
                          SUM =    57897
RECORD#28   POINT1= (16,4)
                          SUM =    57393

THE MINIMUM IS           36015
  AT POINT1 = (16,21)


END: JOB 0019 18:08:12 FRI APR 11, 1986 COST=$0.26/$0.00 BALANCE=$218.79/$0.00
     CPU=0.12 IO=154 CDS_IN=0 CDS_OUT=0 PAGES=2 TAPE_MTS=0 TAPE_RES=0:00:00
```

ENG FCZ C
4557

CPU MODEL SERIAL	3090 170596
VM/SP SERVICE LEVEL	0316
USERID ORIGIN	ENGFCZC ENGFCZC
DISTRIBUTION CODE	ENGFCZC
SPOOL FILE NAME TYPE	ENGFCZC TERM
CREATION DATE	04/11/86 18:09:37
PRINT DATE	04/11/86 18:09:38
SPOOL FILE ID	4047
RECORD COUNT	0114
DEVICE CLASS	OE1E A
FORMS	STANDARD

Boston University
Academic Computing Center
111 Cummington Street
Boston, Massachusetts 02215
(617) 353-2780

AREA D

LISTING OF .MPIOO4D FRI APR 11, 1986 18:09:37 PAGE 1

```
/LOAD PASCAL
PROGRAM RDATO1 (INPUT,OUTPUT);
CONST RECS = 55;        (*  NUMBER OF RECORDS IN DATA *)
      LIST = 1;         (*  O FOR A LONG LIST OR 1 TO SUPRESS IT *)
VAR N,I,J,K,ABSL,MIN : INTEGER;
    MINX1,MINY1,MINX2,MINY2 : INTEGER;
    SUM,X,Y,W : ARRAY (.1..RECS.) OF INTEGER;
FUNCTION ABS (X:INTEGER) : INTEGER;
BEGIN
  IF X<O THEN ABS:=-X
         ELSE ABS:= X;
END;
BEGIN
   MIN:=999999;
   FOR I:=1 TO RECS DO
     READLN (N,X(.I.),Y(.I.),W(.I.));
   FOR I:=1 TO RECS DO
     BEGIN
     WRITE (' RECORD#',I:1,'  POINT1= (');
     WRITELN (X(.I.):1,',',Y(.I.):1,')');
     IF LIST = O
       THEN BEGIN
            WRITE   ('      POINT2');
            WRITE   ('  POPULATION');
            WRITELN ('        RESULT');
            END;
     SUM(.I.) := O;
     FOR J:=1 TO RECS DO
       BEGIN
       ABSL:=ABS(X(.I.)-X(.J.));
       ABSL:=ABSL+ABS(Y(.I.)-Y(.J.));
       ABSL:=ABSL*W(.J.);
       IF (LIST = O)
         THEN BEGIN
              WRITE ('       ');
              WRITE ('(',X(.J.):2,',',Y(.J.):2,')     ');
              WRITELN (W(.J.):6,'           ',ABSL:6);
              END;
       SUM(.I.) := SUM(.I.) + ABSL;
       END;
    WRITELN ('                                    SUM = ',SUM(.I.):7);
    IF MIN > SUM(.I.)
      THEN BEGIN
           MIN:=SUM(.I.);
           MINX1:=X(.I.);
           MINY1:=Y(.I.);
           END;
    WRITELN;
    END;
  WRITELN;
  WRITELN (' THE MINIMUM IS ',MIN);
  WRITELN ('    AT POINT1 = (',MINX1:1,',',MINY1:1,')');
END.
/DATA
 43      27          45           84
```

AREA D

44	32	45	84
45	37	45	84
46	41	45	84
47	46	45	84
48	52	45	84
49	58	45	84
53	27	39	84
54	32	39	84
55	37	39	84
56	41	39	84
57	46	39	84
58	52	39	84
59	58	39	84
62	19	33	84
63	27	33	84
64	32	33	84
65	37	33	84
66	41	33	84
67	46	33	84
68	52	33	84
69	58	33	84
73	19	27	42
74	27	27	84
75	32	27	84
76	32	27	84
77	41	27	84
78	46	27	84
79	52	27	84
80	58	27	84
84	19	21	84
85	27	21	42
86	32	21	42
87	37	21	84
88	41	21	84
89	46	21	84
90	52	21	84
91	58	21	84
95	19	16	84
96	27	16	84
97	32	16	84
98	37	16	42
99	41	16	42
100	46	16	84
101	52	16	84
102	58	16	84
105	19	10	84
106	27	10	84
107	32	10	84
108	37	10	84
109	41	10	84
112	19	4	84
113	27	4	63
114	32	4	42
115	37	4	21

/END DATA

END OF FILE .MPIOO4D RECORD COUNT: 110

AREA D

44	32	45	84
45	37	45	84
46	41	45	84
47	46	45	84
48	52	45	84
49	58	45	84
53	27	39	84
54	32	39	84
55	37	39	84
56	41	39	84
57	46	39	84
58	52	39	84
59	58	39	84
62	19	33	84
63	27	33	84
64	32	33	84
65	37	33	84
66	41	33	84
67	46	33	84
68	52	33	84
69	58	33	84
73	19	27	42
74	27	27	84
75	32	27	84
76	32	27	84
77	41	27	84
78	46	27	84
79	52	27	84
80	58	27	84
84	19	21	84
85	27	21	42
86	32	21	42
87	37	21	84
88	41	21	84
89	46	21	84
90	52	21	84
91	58	21	84
95	19	16	84
96	27	16	84
97	32	16	84
98	37	16	42
99	41	16	42
100	46	16	84
101	52	16	84
102	58	16	84
105	19	10	84
106	27	10	84
107	32	10	84
108	37	10	84
109	41	10	84
112	19	4	84
113	27	4	63
114	32	4	42
115	37	4	21

/END DATA

END OF FILE .MPIOO4D RECORD COUNT: 110

```
0680 --    program rdat01 (input,output);
06a8 --    const RECS = 55;      (*  number of records in data *)
06a8 --         LIST = 1;        (*  0 for a long list or 1 to supress it *)
06a8 --    var n,i,j,k,abs1,min : integer;
06c0 --        minx1,miny1,minx2,miny2 : integer;
06d0 --        sum,x,y,w : array (.1..RECS.) of integer;
0a40 -- a  function abs (x:integer) : integer;
0000 0- a  begin
0018 --      if x<0 then abs:=-x
002a --             else abs:= x;
004c -0 a  end;
0000 0-    begin
0062 --      min:=999999;
006a --      for i:=1 to RECS do
0088 --        readln (n,x(.i.),y(.i.),w(.i.));
011c --      for i:=1 to RECS do
013a 1-        begin
013a --        write (' Record#',i:1,'  point1= (');
0176 --        writeln (x(.i.):1,',',y(.i.):1,')');
023a --        if LIST = 0
023a 2-          then begin
0246 --               write    ('      Point2');
025e --               write    ('  Population');
0276 --               writeln  ('       Result');
0294 -2               end;
0294 --        sum(.i.) := 0;
02ac --        for j:=1 to RECS do
02ca 2-          begin
02ca --          abs1:=abs(x(.i.)-x(.j.));
0314 --          abs1:=abs1+abs(y(.i.)-y(.j.));
036a --          abs1:=abs1*w(.j.);
039c --          if (LIST = 0)
03ae 3-            then begin
03a8 --                 write ('      ');
03c0 --                 write ('(',x(.j.):2,',',y(.j.):2,')    ');
0490 --                 writeln (w(.j.):6,'          ',abs1:6);
04f0 -3                 end;
04f0 --          sum(.i.) := sum(.i.) + abs1;
052e -2          end;
0550 --        writeln ('                        sum = ',sum(.i.):7);
0598 --        if min > sum(.i.)
05b4 2-          then begin
05c4 --               min:=sum(.i.);
05e4 --               minx1:=x(.i.);
0604 --               miny1:=y(.i.);
0624 -2               end;
0624 --        writeln;
0630 -1        end;
0652 --      writeln;
065e --      writeln (' The Minimum is ',min);
0694 --      writeln ('   At Point1 = (',minx1:1,',',miny1:1,')');
0746 -0    end.
```

aaec pascal 2.0 compilation concluded

no errors detected in pascal program

loader

program size - 58f0 (hex), 22,768 (decimal).

Record#1 point1= (27,45)
sum = 135156

Record#2 point1= (32,45)
sum = 124446

Record#3 point1= (37,45)
sum = 119616

Record#4 point1= (41,45)
sum = 120288

Record#5 point1= (46,45)
sum = 126588

Record#6 point1= (52,45)
sum = 140196

Record#7 point1= (58,45)
sum = 159852

Record#8 point1= (27,39)
sum = 116508

Record#9 point1= (32,39)
sum = 105798

Record#10 point1= (37,39)
sum = 100968

Record#11 point1= (41,39)
sum = 101640

Record#12 point1= (46,39)
sum = 107940

Record#13 point1= (52,39)
sum = 121548

Record#14 point1= (58,39)
sum = 141204

Record#15 point1= (19,33)
sum = 131796

Record#16 point1= (27,33)
sum = 104916

Record#17 point1= (32,33)
sum = 94206

Record#18 point1= (37,33)
sum = 89376

Record#19 point1= (41,33)
sum = 90048

Record#20 point1= (46,33)
sum = 96348

Record#21 point1= (52,33)
sum = 109956

Record#22 point1= (58,33)
sum = 129612

Record#23 point1= (19,27)
sum = 128268

Record#24 point1= (27,27)
sum = 101388

Record#25 point1= (32,27)
sum = 90678

Record#26 point1= (37,27)
sum = 85848

Record#27 point1= (41,27)
sum = 86520

Record#28 point1= (46,27)
sum = 92820

Record#29 point1= (52,27)
sum = 106428

Record#30 point1= (58,27)
sum = 126084

Record#31 point1= (19,21)
sum = 132300

Record#32 point1= (27,21)
sum = 105420

Record#33 point1= (32,21)
sum = 94710

Record#34 point1= (37,21)
sum = 89880

Record#35 point1= (41,21)
sum = 90552

Record#36 point1= (46,21)
sum = 96852

Record#37 point1= (52,21)
sum = 110460

Record#38 point1= (58,21)
sum = 130116

Record#39 point1= (19,16)
sum = 141540

Record#40 point1= (27,16)
sum = 114660

Record#41 point1= (32,16)
sum = 103950

Record#42 point1= (37,16)
sum = 99120

Record#43 point1= (41,16)
sum = 99792

Record#44 point1= (46,16)
sum = 106092

```
Record#45  point1= (52,16)
                        sum =  119700

Record#46  point1= (58,16)
                        sum =  139356

Record#47  point1= (19,10)
                        sum =  159684

Record#48  point1= (27,10)
                        sum =  132804

Record#49  point1= (32,10)
                        sum =  122094

Record#50  point1= (37,10)
                        sum =  117264

Record#51  point1= (41,10)
                        sum =  117936

Record#52  point1= (19,4)
                        sum =  182868

Record#53  point1= (27,4)
                        sum =  155988

Record#54  point1= (32,4)
                        sum =  145278

Record#55  point1= (37,4)
                        sum =  140448


The Minimum is        85848
  At Point1 = (37,27)
*go
.
```