Geometric Optimization And Computational Complexity

Chanderjit Bajaj
Ph.D. Thesis

TR 84-629
August 1984

Department of Computer Science
Cornell University
Ithaca, New York 14853

GEOMETRIC OPTIMIZATION

AND

COMPUTATIONAL COMPLEXITY

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Chanderjit Bajaj

August, 1984

Biographical Sketch

Chanderjit Bajaj was born in Calcutta, India on April 19, 1958.

His junior years were spent in St. Xavier's School, Calcutta. After graduating from St. Columba's High School in New Delhi, he attended the Indian Institute of Technology, New Delhi where he graduated in 1980 with a B-Tech. (Bachelor of Technology) degree in Electrical Engineering. Subsequently, he attended the University of Pittsburgh and received an M.S. in Computer Science in 1981.

Bajaj began his graduate studies in Computer Science at Cornell University in 1981. He received an M.S. in Computer Science in 1983.

Dedication

To the fond memory of my father, Shri Madan Lal Bajaj, who was also my dearest friend.

Acknowledgements

I would like to thank John Hopcroft for his patience, teaching and encouragement throughout my term at Cornell. He taught me how to approach seemingly difficult problems and particularly to 'winnow the chaff from the grain', in my research. For his advice and constant suggestions, on and off the field of research, I shall always be grateful. David Gries befriended me, as he does most all incoming students, from my first day at Cornell. From him I not only learnt how to play softball (third base), but also the science of teaching (and programming). As a member of my commmittee, he also promises to read my thesis. So does Chris Pottle, who I also wish to thank for cheerfully serving on my committee.

Paul Pritchard, (or should I say Geoff Hunt ?), the Squash champ and Juris Hartmanis, the Tennis champ, I wish to thank for the numerous discussions before, after and at times during our championship duals. Ming Li (the best Chinese cook in town) and Luc Longpre' (the best French cook around), I thank for the wonderful animated conversations we had, ranging from the highly technical to the deeply philosophical and murderously mundane. Our frequent dinners together were highly conducive to research and I shall miss them. Tom Coleman, John Gilbert, Raimund Seidel and Yaacov Yesha are the other persons who were always ready to listen to an interesting thought and also provided me with many a stimulus.

To my father my revered thanks, for all that you taught me and all the great times we shared together. For the rest of my family consisting of my mother, three sisters, (and four nephews), I have the deepest love and sincere thanks for being my staunchiest

friends. For Khusro Rustamji, who has been a second father to me, I have only the utmost admiration. For your unselfish help and guidance, thanks don't seem enough. My other friends (and family in Ithaca) that I wish to thank, consist of Paul, Cathy and Julie Dawson, Megu (babu), Big Cy, small Cy, Punj, Daf, Guy, Alka, Reiko (fair weather) Yamada and Elena Pasquale (Are you going to return to Italy or is it La Jolla?).

It would be remiss to exclude all the nice compadre's of our Cornell Computer Science department. For fear of being clubbed over the head on leaving someone's name out, let me offer my warmest thanks collectively to all bridge groups, volleyball cliques, tennis factions, softball squads, badminton trios, basketball slammer-jammers, etc., that this wonderful place teems with. Finally sincere thanks to Mike Woodall, our proficient mathematics librarian, for all the help he provided and who never protested even my most obscure demands.

Table of Contents

GEOMETRIC OPTIMIZATION AND COMPUTATIONAL COMPLEXITY

Chauderjit Bajaj, Ph.D.
Cornell University 1984

Our purpose here is to study problems involving *geometric optimization*, namely, questions of the type : Is there at least a *minimum* or at most a *maximum* number of certain geometric figures, that are within certain distances of other figures (objects). We are also concerned with the optimization of the *size* of these geometric figures. These problems arise as geometric reductions from various classes of location-allocation optimization problems and are inherently not pure combinatorial. Our primary aim, then, is to discover techniques of dealing with such geometric optimization problems, while adapting to these problems the older combinatorial design and analysis methods.

The task of classifying problems accurately in the polynomial hierarchy is one of increasing importance. To solve an optimization problem deterministically it seems that one must solve both an NP and a $Co\ NP$ problem. The significance of the classes NP and $Co\ NP$ are that none of the problems they include is known to have a polynomial time solution. We show that if $NP \neq Co\ NP$ then there are interesting natural geometric optimization problems (location-allocation problems under *minsum*) in Δ_2^P that are in neither NP nor $Co\ NP$. Hence, all these problems are shown to belong *properly* to Δ_2^P, the second level of the polynomial hierarchy. We also show that if $NP \neq Co\ NP$ then there are again some interesting geometric optimization problems (location-allocation problems under *minmax*) *properly* in Δ_2^P and furthermore they are *complete* for a class D^P (which is contained in Δ_2^P and contains $NP \bigcup Co\ NP$). Also considered are the above geometric location-allocation optimization problems for the case when the allocation is predetermined. Both efficient algorithms and worst-case lower bounds are derived.

Necessary conditions for the existence of *maxima* and *minima* in optimization problems are generally tied to the question of solvability of an equation or a system of equations. In calculus these equations are algebraic. By generating the minimal polynomial whose root over the field of rational numbers is the solution of the geometric optimization problem on the real (Euclidean) plane, we are able to prove the non-solvability of certain geometric optimization problems by *radicals*. The algebraic degree of the optimizing solution, which is the degree of the irreducible minimal polynomial for the problem, correlates with the inherent difficulty of constructing the solution and provides an algebraic complexity measure for these geometric optimization problems.

CHAPTER 1

INTRODUCTION

1.1. Geometric Optimization

Geometry is perhaps the oldest branch of mathematics that has not lost its lustre through the period of time. It is the study of the size, position, orientation and shape of most of the familiar objects around us. Geometrical figures like circles, ellipses, squares, rectangles are easily discernible in most of our modern day amenities. The wheels of a car were chosen to be roughly circular since circles roll easier than irregularly shaped objects. Man, being an optimizer (a *maximizer* or *minimizer*), always optimizes the geometric objects around him, through a search for regularity and symmetry and to make his task easier. Typical geometric optimization problems are the problem of packing the maximum number of oranges (spheres) into a rectangular crate and that of paving an oval shaped floor using the minimum number of square tiles. Minimizing operation costs is often also an optimization objective.

During the last few years sophisticated computer applications involving the representation and manipulation of geometric objects, have rapidly appeared in a broad variety of disciplines. Areas such as robotics, cad/cam, computer graphics and office automation are some of the major forerunners. For all of them, geometric optimization is a common theme. In robotics a long desired goal is that of having a robot pick out workpieces from individual bins, contour them to the right geometric shape (for example a bracket with bevelled cylindrical holes and spiral screw threads), with the minimum amount of waste material and the maximum degree of accuracy, and finally assemble

them into the finished product. In cad/cam, designers give shape to automobiles with the help of wide variety of geometric figures consisting of planar and quadric surfaces, while minimizing the drag coefficient and maximizing the esthetic appeal. Computer graphics and office automation are involved with problems of displaying numerous geometric figures with maximum accuracy and minimum cost and time. In order that these areas achieve their full promise, it will be necessary to design efficient algorithms for the problems that arise as well as classify the inherent computational complexity of the more difficult ones.

Geometric Location-Allocation Optimization Problems involve the optimization of the placement of various geometric objects and are of considerably renewed importance, precisely for the above reasons. They are of practical importance because such problems are frequently formulated in the location of facilities, such as water treatment plants, oil refineries, industrial warehouses, fire stations and hospital clinics, [Co63],[FW74],[Ha63],[KP79],[Sc71],[We37]. More recent applications arise from VLSI layout design (component location) and robotics (robot location) [BW80],[IA83]. The strong theoretical interest stems from the fact that though these optimization problems have occurred and been studied for quite some time in a diversity of literature in network location theory, [FW74],[KP79],[TFL83], little is known about their computational complexity, in particular for the important case of the Euclidean plane. Furthermore, since nearly all of these problems are recast in a mathematical framework bordering on Euclidean geometry these location problems, are mathematically intriguing in their own right.

Most of the geometric optimization problems can be thought of as ***special*** cases of well studied graph problems [SH75],[GGJ76],[Pa77],[Pa81]. Whereas the general problems deal with vertex points joined by edges having arbitrarily specified lengths, the corresponding geometric problems deal with points in the plane or in 3-space with the edge lengths being the actual interpoint distances under one of the standard distance metrics. Being special cases offers hope that although these problems are difficult for arbitrary graphs (networks), efficient algorithms could be possible for the corresponding geometric problems, [as is the case in Chapter 3]. There is also the general feeling in the literature (eg. [SH75]), that the problems in the plane are considerably easier, either because the heuristics seem to perform better or because special methods of attack are applicable. However, it is at times possible to show that the geometric versions cannot be "any" easier than the general problem at least as far as the exact solution is concerned, [GGJ76],[Pa77],[and as is shown in Chapter 2].

Our purpose here is to study problems involving ***geometric optimization***, namely, questions of the type : Is there at least a ***minimum*** or at most a ***maximum*** number of certain geometric figures, that are within various distances of other figures (objects). One is also concerned with the optimization of the size of these geometric figures. We show that these problems arise as geometric reductions from the major classes of Location-Allocation Optimization problems. These geometric optimization problems are inherently not pure combinatorial problems since the optimal solution belongs to an infinite feasible solution set, the entire real (Euclidean) plane. Optimal solutions for combinatorial optimization problems on the other hand, are one of a finite number of feasible solutions [PS81]. Our primary aim then is to discover techniques of dealing with such geometric optimization problems, while adapting to these problems, the older

combinatorial design and analysis methods [AHU74],[HU79].

The task of classifying problems accurately in the polynomial hierarchy is one of considerable interest and importance. We undertake this in [***Chapter*** 2] for the above geometric optimization problems. (See also [KP79], where such a task is mentioned to be a very desirable one for these problems). To solve an optimization problem deterministically it seems that one must solve both an NP and a $Co\ NP$ problem. In section (2.2) of [Chapter 2] we show that if $NP \neq Co\ NP$ then there are interesting natural geometric optimization problems. (location-allocation problems under ***minsum***), in Δ_2^P that are in neither NP nor $Co\ NP$. (that is, ***not*** in $NP \bigcup Co\ NP$). Hence, all these problems are shown to belong ***properly*** to Δ_2^P, the second level of the polynomial hierarchy. In section (2.3) of [Chapter 2] we show that if $NP \neq Co\ NP$ then there are again some interesting geometric optimization problems, (location-allocation problems under ***minmax***), ***properly*** in Δ_2^P and furthermore they are ***complete*** for a class D^P (which is contained in Δ_2^P and contains $NP \bigcup Co\text{-}NP$). All these problems are shown to be polynomial-time reducible to each other and ***D^P complete***.

The topic of [***Chapter*** 3] is the investigation of the geometric location-allocation optimization problems of the previous chapter for the case when the allocation is predetermined. Also considered are some related geometric optimization problems that stem from previously mentioned applications. We devise both efficient algorithms and construct nonlinear lower bounds for most of these problems.

The technique of using algebraic methods for analyzing the complexity of geometric problems is not new. The complexity of straight-edge and compass constructions (and Mohr-Mascheroni compass constructions), is quite well known [CR41],[He75]. Furthermore, necessary conditions for the existence of *maxima* and *minima* in optimization problems are generally tied to the question of solvability of an equation or a system of equations. In calculus these equations are algebraic; (in the calculus of variations, they are differential). By generating the minimal polynomial, whose root over the field of rational numbers is the solution of the geometric optimization problem on the real (Euclidean) plane, we are able to answer questions about the solvability (or non-solvability) of the problem by radicals. Furthermore, the algebraic degree of the optimizing solution, which is the degree of the irreducible minimal polynomial for the problem, (and the order of the splitting field of the minimal polynomial, which is also the order of the appropriate Galois group), gives an idea of the inherent difficulty of constructing the solution as well as gives a complexity measure for these geomtric optimization problems. In light of the above, a more concerted application of algebraic field extensions (and Galois Theory) to geometric optimization problems seems to be a very worthwhile objective. A step in this direction is undertaken in [*Chapter* 4], wherein we reduce certain geometric optimization problems to algebraic problems and analyze thier complexity as outlined above.

1.2. Location-Allocation Problems

There frequently arise in practice problems that are concerned with how to serve and supply, in an optimum fashion, a set of destinations that have fixed and known locations. What must be determined, in these problems, is the ***number***, ***location*** and ***size*** of the ***sources*** that will most economically supply the given set of ***destinations*** with some commodity, material or service. Examples of such problems are the determination and location of warehouses, distribution centers, communication centers and production

facilities.

Location - Allocation problems are characterized by the following general structure.

Given a set of n points (destinations) in the plane, find the location of k points (sources) and an allocation of each destination to a source, so as to optimize an objective function based on the distances between sources and destinations.

It should be noted that in many cases the full problem in all its generality seldom arises. The assumptions made for these problems are as follows:

(1) There are no restrictions on the permissible source capacities.

(2) Unit shipping costs are independent of total source output.

Even with these assumptions the problem is a formidable one.

If it is assumed that there are no capacity restrictions and that unit shipping costs are independent of source output, it is clear that in a *minimum* cost solution, each destination will be supplied by a single source, moreover, by its *closest* source. However, in addition to not knowing the location of each of the m sources in the minimum cost solution, we also do not know which source is to serve which subset of destinations. In order to avoid a definite association of sources and destinations at the outset, we shall assume in the definition of the objective function (*cost* of supplying n destinations with m sources) that every destination can be supplied by any source.

For k sources and n destinations there are $S(n,k)$ possible assignments of n destinations to k sources, where $S(n,k)$ is the Stirling number of the *second* kind and is given by

$$S(n,k)=1/k!\sum_{j=0..k}\binom{k}{j}(-1)^j(k-j)^n$$

where $\binom{k}{j}$ are the usual binomial coefficients. Among these assignments is an allocation of sources to destinations which gives the minimum cost we seek.

Location-allocation problems seemed to have been first introduced by [Mi58]. They were thereafter formulated by [Co63]. The analogous problems of locating p service points on a network are usually termed the p-median and p-center problems [TFL83]. Whereas, in the Euclidean plane, the main feature of single facility location is that the problem is convex, the multi-facility problem is neither convex nor concave and thus *non linear* programming methods usually yield only a local minimum. Attempts to optimally solve the location-allocation problems have been made by developing branch and bound [El73], [KS72] and dynamic programming methods [Be65],[CS69]. These methods are, however, limited to relatively small problems. The state of the art at the moment seems to be that whereas small location-allocation problems can be solved optimally, larger problems are being solved using heuristic methods which yield local minima that may or may not be global [Co64],[Co67],[Sc71],[Pa81].

CHAPTER 2

GEOMETRIC OPTIMIZATION AND THE POLYNOMIAL HIERARCHY

2.1. Overview

The task of classifying problems accurately in the polynomial hierarchy is one of considerable interest as well as importance. We undertake this for geometric optimization problems.

To solve an optimization problem deterministically it seems that one must solve both an NP and a $Co\ NP$ problem. In the following sections 2.2 and 2.3 we show that , if $NP \neq Co\ NP$, both the positive and negative information is needed from an NP *oracle* to solve a geometric optimization problem. The NP *oracle* is in fact the recognition version of the optimization problem, and is known to be in NP.

In section (2.2) one shows that if $NP \neq Co\ NP$ then there are interesting natural problems properly in Δ_2^P that are not in either NP or $Co\ NP$. (that is *not* in $NP \cup Co\ NP$). Three different classes of geometric optimization problems derived under a discrete *minsum* optimization criterion from certain location-allocation problems in the Euclidean plane, are considered. We call these problems, 'MINSUM Geometric Optimization Problems'. Each of these problems is studied with three different distance metrics (Euclidean, Rectilinear, Infinity), as well as for feasible solution sets which are both *finite* and *infinite*. All these problems are shown to belong *properly* to Δ_2^P, the second level of the polynomial hierarchy.

In section (2.3) one again shows that if $NP \neq Co\text{-}NP$ then there are interesting natural problems *properly* in Δ_2^P and furthermore they are complete for a class D^P (which is contained in Δ_2^P and contains $NP \bigcup Co\text{-}NP$). Three different classes of geometric optimization problems derived under a discrete *minmax* optimization criterion from location-allocation problems in the Euclidean plane, are considered. We call these problems, 'MINMAX Geometric Optimization Problems'. These problems are shown to be polynomial-time reducible to each other and furthermore D^p *complete*. In particular for l_p distances, $1 \leq p \leq \infty$, and arbitrary distance metrics as well as for problems having either *finite* or *infinite* feasible solution sets.

2.2. MINSUM Geometric Optimization Problems

A *minsum* location objective is one which minimizes the sum of the costs resulting from a given location solution and is some measure of the average cost of serving the destinations. A common real world situation that might be formulated with a *minsum* objective is one of locating a water treatment plant so that the sum of the length of pipes required to serve water to the various households or industrial users is minimized Another example could be that of locating components on a VLSI chip so as to minimize the sum of the wire connections from the components.

Under this minsum location objective it is possible to distinguish two basic approaches. The first suggests that location sites may be anywhere in the real Euclidean plane, giving an infinite number of possible location sites. The second approach considers only a finite number of known sites as feasible and models the constraints imposed on the possible location, ensuring that undesirable and impractical locations need not be considered. The various distance metrics used, *Rectilinear* (l_1), *Euclidean* (l_2) and

Infinity (l_∞), reflect the appropriate problem restrictions. Between two points $a=(a_x,a_y)$ and $b=(b_x,b_y)$ in the plane the l_1 distance is $|a_x - b_x| + |a_y - b_y|$; the l_2 distance is $\sqrt{(a_x - b_x)^2+(a_y - b_y)^2}$ and the l_∞ distance is $\max(|a_x - b_x|, |a_y - b_y|)$

Here we consider the three different classes of multiple points location-allocation problems, (P_1, P_2 and P_3) in the plane under a minsum optimization objective. In these problems we are given a set $T=\{(x_i,y_i),\ i=1..n\}$, the location of n fixed destination points (*destinations*) in the plane and need to locate a number of points (*sources*) to service the destinations in a way which behooves the application on hand. In the case of locating multiple sources, the *allocation* of the destinations to the sources must also be ascertained. A common assumption for these problems [Co63] is that the sources are considered to be uncapacitated; that is there are no capacity limitations. As a consequence, each destination can be completely serviced by a single source though a source can itself service more than one destination. Furthermore in the *optimal* solution each destination is allocated to its closest located source. However this optimal allocation, is one of the exceedingly large number of possible allocations[1]. Not known a priori it needs to be determined. It is also interesting to note that the capacitated versions of these geometric location-allocation optimization problems (with sources having finite capacities), turn out to be various cases of the more familiar transportation location problems and under discrete solution space constraints, to be the plant location and warehouse location problems [FW74].

---

[1] The total number of possible assignments (allocations) of n destinations to k sources is $S(n,k)$, the Stirling number of the 2nd kind.

Given the set T, as specified before, of n destinations in the plane

(P_1) Locate k points (sources) so as to ***minimize*** the ***sum*** of the weighted distances between the destinations and the sources closest to them.

(P_2) Locate k points (sources) so that for a ***maximum*** number of destinations the ***sum*** of the weighted distances of these destinations from their closest sources does not exceed a prescribed limit L.

(P_3) Locate a ***minimum*** number of points (sources) so that the ***sum*** of the weighted distances of the destinations and their closest sources does not exceed a prescribed limit L.

The weighted distances mentioned above come from a weight w_j assigned to each destination j and is some measure of the special cost of serving destination j in traveling from its closest source.

Problems P_1, P_2 and P_3 allow location of the sources to be anywhere in the plane. Let problems Q_1, Q_2 and Q_3 correspond to problems P_1, P_2 and P_3 respectively, with the location of the sources being constrained to a finite discrete set S of possible locations in the plane and of size polynomial in n. Let problems R_1, R_2 and R_3 be ***restricted*** versions of problems Q_1, Q_2 and Q_3 respectively, with the location of the sources being a subset of T, the set of destination points.

2.2.1. Decision Problems

We restrict our attention to ***decision*** problems (answer is either 'yes' or 'no') and derive from the above problems the corresponding optimization problems versed as decision problems. Such a problem π can be viewed as consisting of a set D_π of finite

combinatorial objects called ***instances*** for which the answer is 'yes' and a subset $Y_\pi \subseteq D_\pi$ of instances for which the answer is 'yes'. The problem itself corresponds to the ***language***[2] over $\sum^*$, consisting of all those strings which ***encode*** instances that have 'yes' answers. This is in fact the reason one considers decision problems, since 'language' is a much more mathematically precise object for our purpose here.

Let us take ***locatable*** sources to mean sources locatable in one of the three categories, P, Q and R, specified above. Then given parameters k, m and L, we have,

(O_1) Is L the ***minimum sum*** of the weighted distances of the n destinations and the closest of the k ***locatable*** sources.

(O_2) Is m the ***maximum*** number of destinations, $m \leq n$, for which the ***sum*** of the weighted distances of these destinations from the closest of the k ***locatable*** sources is $\leq L$.

(O_3) Is k the ***minimum*** number of ***locatable*** sources for which the ***sum*** of the weighted distances of the n destinations from the closest of these sources is $\leq L$.

Our goal is to show that the above decision optimization problems (languages), properly belong to the class Δ_2^P, the second level of the polynomial hierarchy. A problem P is a ***proper***-Δ_2^P problem if

(1) P is in Δ_2^P

---

[2] If $\sum$ is a finite set of symbols, let $\sum^*$ denote the set of all finite strings of symbols from $\sum$. A language over $\sum$ is any subset L of $\sum^*$.

(2) P is not in $\sum_1^P = NP$, assuming $NP \neq Co\ NP$.

(3) P is not in $\Pi_1^P = Co\text{-}NP$, assuming $NP \neq Co\ NP$.

To show the above for our optimization problems we first need to show the corresponding *recognition* versions of these optimization problems to be NP-complete.

The recognition versions are easily derived from the optimization problems by use of an additional parameter A in replacing a 'minimum' cost by a cost of ***no more than*** A, ($\leq A$) and a 'maximum' cost by a cost of ***at least*** A ($\geq A$). The point to observe about the correspondence is that as long as the optimal cost function is easy to evaluate (here simply the sum of distances), the recognition version is reducible to the corresponding optimization problem and thus ***no harder***. For all we need do is obtain the optimal cost and compare it to the given bound A. Thus if we show that the recognition version of an optimization problem is NP-***complete***, we would know that the optimization version is at least as hard (NP-***hard***).

To show the recognition versions of the above optimization problems to be NP-complete we must formulate them in a more suitable manner. We assume that the set of destination points T are given as a set of integer coordinate pairs. Furthermore we assume that the set $T = \{p_1,...,p_n\}$ is a multiset with w_i points in T with exactly the same coordinate p_i conforming to a destination point p_i having an integer weight w_i. From the optimization problems P_i, we obtain the corresponding problems PP_i.

Given the multiset T of destination points as specified before and integers k, m and L.

(PP_1)

Is there a set $KS = \{s_1,...,s_k\}$ of k sources in the plane such that the ***sum*** of the distances between the destinations in T and the sources closest to them is $\leq L$.

(PP_2)

Is there a subset $T' \subseteq T$, $|T'| \geq m$, such that for a set $KS = \{s_1,...,s_k\}$ of k sources in the plane, the ***sum*** of the distances between the destinations in T' and the sources closest to them is $\leq L$.

(PP_3)

Is there a set KS, $|KS| \leq k$, of sources in the plane, such that the ***sum*** of the distances between the destinations in T and the sources closest to them is $\leq L$.

One can also formulate the corresponding problems QQ's and RR's with the location of sources restricted to finite sets, as specified before.

Lemma 2.1 : PP_1 reduces to PP_2. Further, PP_1 reduces to PP_3. Similar results hold for the problems QQ's and problem RR's.

Proof : PP_1 reduces to PP_2, for $T' = T$ and $m = n$. Further, PP_1 directly reduces to PP_3 since if less than k sources satisfy the limit L, k sources would definitely do so. □

The discrete problem, RR_1 was shown to be NP-complete for the (integerized) *Euclidean*, l_2 distance metric in [Pa81]. Since the RR problems are restricted versions of the corresponding QQ problems, it follows that the finite solution space problem, QQ_1 is also NP-complete for the l_2 distance metric. In [MS84], the infinite solution space problem, PP_1 was shown to be NP-complete for the l_1 and l_2 distance metrics. We extend these results and show that for the l_1 distance metric the infinite solution space problem, PP_1, reduces to the corresponding discrete RR_1 problem (and hence to the finite solution space problem, QQ_1) and thus these are all NP-complete. We complete the picture by showing that the PP_1, QQ_1 and RR_1 problems, for the l_∞ distance metric, are also NP-complete. Using Lemma 2.1 it follows that the problems PP_2, PP_3 as well a the corresponding problems QQ_2, QQ_3 and RR_2, RR_3 are as difficult and also NP-complete for each of the three distance metrics.

From the earlier discussion we know that NP-completeness of the above problems implies that the corresponding optimization problems, for both finite and infinite solution spaces and for the three distance metrics, are all NP-hard. Assuming $NP \neq Co\ NP$, one can then show that each of the above optimization problems belong properly in Δ_2^P.

The problems as formulated above are all in NP. We henceforth assume this fact in all the NP-completeness proofs.

Theorem 2.2 : Problem RR_1, having a discrete feasible solution set, is NP-complete for the infinity (l_∞), distance metric.

Proof : To prove it complete we show a polynomial-time reduction from the Exact Cover problem, a known NP-complete problem [GJ79]. The construction is exactly similar to the construction of [Pa81] with a few essential changes to correspond to the l_∞ distance metric. We repeat the entire construction here only for the sake of completeness.

The configuration of points is as follows. A row R of *length* m has $6m+4$ points and the two extreme points b, b' have weight m^2, which implies that they have to be sources in any optimal solution. If we allocate $m+2$ *sources* to R, then the two best solutions are shown below, both having b and b' as sources. Solution 1 induces the partition of R shown in solid lines while the broken lines conform to solution 2. Note that solution 1 is cheaper by 2ϵ, where $\epsilon = m^{-1}$. Among each resulting group of 6 points the source can be chosen either as an *upper* source s or as a *lower* source s'.

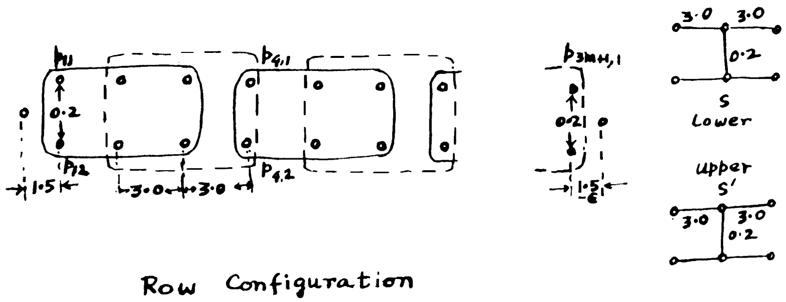


Figure 2.1.

For the reduction, given any instance $U = \{u_1, ..., u_{3n}\}$ and the family $F = \{S_1, ..., S_t\}$ of subsets of U, each of size 3, of the Exact Cover problem we construct a (weighted) point set T and integers k and L, such that T has k sources with cost L or less iff there is an exact cover $C \subset F$ of U.

T consists of t rows $R_1, ..., R_t$, each of length $3n$, arranged parallel to each other. Thus the $3n$ columns of this formation, correspond to the elements of U.

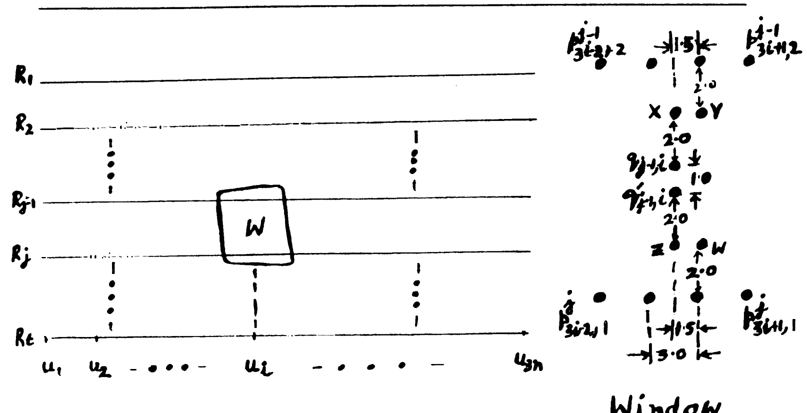


Figure 2.2.

The window is shown above in detail. The spots x, y, w, z are not points of T but only possible positions of points. For each window, one of x, y and one of w, z positions is occupied with points of weight n^{-2}; x is occupied iff $u_i \notin S_{j-1}$, y iff $u_i \in S_{j-1}$. Similarly w iff $u_i \in S_j$ and z iff $u_i \notin S_j$. (We use fractional weights with the assumption that all coordinates as well as the limit L will eventually be multiplied by a sufficiently large integer and rounded).

Define $k = t(3n+2) + 3n(t-1)$. The first term provides enough sources for all the t rows and the second term, one source for the q, q' pair in each window. L consists of three components $L_1 + L_2 + L_3$. $L_1 = t(2 * 1.5 + 3n(4.6)) - 2nt$ and comes from the t rows. $L_2 = 3n(t-1)$ and comes from the cost due to the q or q' pair, from the $3n(t-1)$ windows at a cost of 1 per pair. $L_3 = 12n(t-1)/n^2$ is the cost of connecting each of the $6n(t-1)$ points x or y and w or z to the closest q or q' point, always 2 away.

Then there exists a subset $KS \subset T$ of size k with cost L or less iff F contains an exact cover C of U. Assume such a KS exists. Then for the cost to be L or less the optimal allocation is as specified above. Each row is then grouped by solution 1 or 2. Let row R_j grouped as solution 1 mean that $S_j \in C$ where C is the claimed exact cover. Again, for the cost to be at most L, at least n rows must be grouped by solution 1.

For R_j grouped by solution 1 consider the i^{th} group where $u_i \in S_j$. Then both w (above) and y (below) positions are occupied by a point. Since positions w, x, y or z can charge only 2 to the cost in L_3 of L, the w and y must connect to their corresponding q sources. This further implies that the x or y point of $q_{j-1,i}$ must be picked up by their lower sources in R_{j-1} and w or z point of $q'_{j,i}$ by their upper source. The same argument repeats and one notes that this change of upper to lower occurs only once per column, with the i^{th} group of any row R_k, $k < j$, must have a lower source while for $k > j$, must have an upper source. Moreover R_j causes this change to all three columns corresponding to the three elements $u_i \in S_j$. Also there are no overlaps in the sets S_j of C, since if $u_i \in S_j$, then by the crucial construction of the window, the positions w or y are 2 away just from the solution 2 source (appropriate upper or lower) and if linked up this way, implies R_j is grouped as solution 2 which means that $S_j \notin C$. Hence C contains n sets without overlap and so is an Exact Cover.

Conversely, assume there is an Exact Cover C, then there exists a solution KS having k sources of cost at most L, by allocating $3n+2$ sources per row, 1 for each q q' pair, with R_j grouped by solution 1 if $S_j \in C$ and solution 2 otherwise. Also for each (unique) $S_j \in C$ and $u_i \in S_j$, let the i^{th} group of R_k for $k \leq j$ have a lower source and for $k > j$ have an upper source, giving the total cost of L. $\square$

Define the l_∞ ***grid points*** of the set T of destinatiou points to be the poiuts shown in the figure 2.3, below.

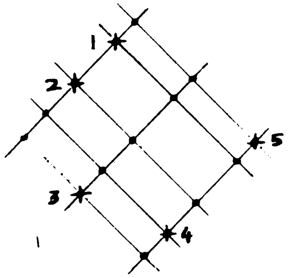


Figure 2.3.

Lemma 2.3 : Iu locatiug a source poiut iu the plaue to so that the sum of the l_∞ distances from the destiuatious T is $\leq L$, oue could as well let the source poiut be au element of the set S, the l_∞ ***grid points*** of the set T.

Proof : Let us assume that for the source poiut p as showu in figure 2.4, the sum of the l_∞ distances from the T destiuatiou poiuts is $\leq L$.

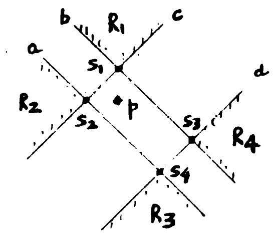


Figure 2.4.

Then there must exist l_∞ ***grid lines*** a, b, c aud d, as above, such that they enclose poiut p, but the poiuts of T are only in the regions R_1, R_2, R_3 and R_1. Let s_1, s_2, s_3 aud s_4 be the l_∞ ***grid points*** defined by these lines. With a little thought one can see that for the suui of the l_∞ distauces with respect to point p, the points in regiou R_1 aud R_3 coutribute their y-distauces while poiuts iu regious R_2 aud R_4 contribute their x-distances. Now if the number of T poiuts in regious R_1 and R_2 are together greater than or equal to the uumber of poiuts in regious R_3 and R_4, theu one can move poiut p to the segment $s_1 s_2$, (otherwise move to segmeut $s_3 s_4$), iu a direction parallel to line a or b aud still keep the sum of the l_∞ costs $\leq L$. By moviug iu this direction we decrease (or keep coustaut) the x and y distauces equally with respect to p. Similarly, depeuding ou whether the uumber of T poiuts iu regious R_2 aud R_3 is $\geq$ (or $<$), thau the number of T points in regions R_1 and R_4, one could move point p to either of the l_∞ ***grid points***, keeping the sum of the l_∞ distances to the T points $\leq L$. $\square$

Theorem 2.4 : The problem PP_1, having the entire Euclidean plane as a solution space, is NP-complete for the l_∞ distance metric.

Proof : For the completeness, we use the same configuration of destination points T as in Theorem 2.2. For this T, we claim that there exists a set of k points, KS, in the plane with cost $\leq L$ *iff* there exists a subset of k points of T with cost $\leq L$. The if-part is trivial. For the only-if part, assume there exists a set of k points KS in the plane. Note that for a known allocation the geometric location-allocation problem reduces to the location of k single source problems. The allocation for the above configuration is clearly determined by design; $(3n+2)$ sources per row and 1 per window. For each of the rows and windows, an application of Lemma 2.3 shows that there exists a corresponding subset of k points of cost $\leq T$. □

We now prove a result similar to *Lemma* 2.3, for locating sources under the l_1 distance metric. This result previously appears in [HPT80]. However, we provide a much simpler proof of this, in a way which suffices for our goal of proving QQ_1, for the l_1 distance metric, to be NP-complete. Define the l_1 ***grid points*** of the set T of destination points to be the points which have an x-coordinate equal to the x-coordinate of any point $p \in T$ and a y-coordinate equal to the y-coordinate of any point $q \in T$, as in figure 2.5.

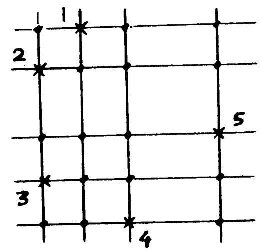


Figure 2.5.

Lemma 2.5 : In locating a source point in the plane so that the sum of the l_1 distances from the destinations T is $\leq L$, one could as well let the source point be an element of the set S, one of the l_1 ***grid points*** of the set T.

Proof : Let us assume that for the source point p as shown in figure 2.6, the sum of the l_1 distances from the T destination points is $\leq L$.

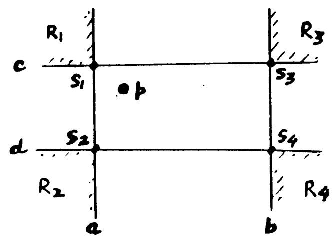


Figure 2.6.

Then there must exist l_1 *grid lines* a and b such that point p lies between them, but no points of T lie between them. Similarly for lines c and d. Let s_1, s_2, s_3 and s_4 be the l_1 ***grid points*** defined by these lines. If the number of T points to the left of a is $\geq$ to the number of T points to the right of b, then one could move point p to the segment $s_1 s_2$ keeping the sum of the l_1 distances to the T points $\leq L$. Otherwise move p to segment $s_3 s_4$. Similarly depending on whether the number of T points above line c is $\geq$ to (or $<$), than the number of T points below b, one could move point p to either of the l_1 ***grid points***, keeping the sum of the l_1 distances to the T points $\leq L$. □

Theorem 2.6 : The discrete solution space problems RR's and QQ's for the l_1 distance metric are all NP-complete.

Proof : The infinite solution space problem, PP_1, was shown to be NP-complete for the l_1 distance metric in [MS84]. A simple reduction from this problem to the discrete solution space problem, RR_1, can be obtained by letting the destination set T' of RR_1 to be the l_1 ***grid points*** of the discrete set PP_1. The proof of Lemma 2.5 suffices to show that there exists a solution for RR_1 *iff* there exists a solution for PP_1. The NP-completeness of QQ_1 again follows automatically from the NP-completeness of the RR_1 problem. □

2.2.2. Strong Completeness

All of the recogniton problems stemming from the three classes of location-allocation optimization problems under each of the three distance metrics, (l_1, l_2, l_∞) and also for infinite, (P's), finite, (Q's) and discrete, (R's), solution sets have now been shown to be NP-complete. We now show that all of them are ***strongly*** NP-complete.

Consider the two functions $Length[I]$ and $Max[I]$, where I is an encoded instance of our problem (language), defined as

$Length[I]$ = a measure of the size of the instance I of the problem

$Max[I]$ = magnitude of the largest number occurring in the instance I

For our problems we can have $Length[I]$ = any polynomial of n, where n is the number of given ***destination*** points. Let π_p denote the problem defined by restricting π to only those instances I for which $Max[I]$ does not exceed $p(Length[I])$, a polynomial in $Length[I]$. Then a problem π is defined to be ***NP-hard*** in the ***strong sense*** if $\exists$ polynomial p over the integers, such that π_p is ***NP-hard***. It is ***NP-complete*** in the ***strong sense*** if in addition $\pi \in NP$.

One way to prove that π is NP-complete in the ***strong sense*** is simply to exhibit a specific polynomial p and prove that π_p is NP-complete in the usual way. A more powerful technique uses the notion of a ***pseudo polynomial transformation*** [GJ79], from π to π', two arbitrary decision problems, with their respective functions $Max[I]$ and $Length[I]$. Such a transformation is a function f from the set of instances D_π of π to the set of instances D_π' of π' such that

(1) f can be computed in time bounded by a polynomial in $Max[I]$ and $Length[I]$.

(2) $\forall I \in D_\pi$, $I \in Y_\pi$, the 'yes instances of π *iff* $f(I) \in Y_\pi'$, the 'yes' instances of π'.

(3) $\exists$ polynomials p and q, such that $\forall I \in D_\pi$, $Length[I] \leq p(Length'[f(I)])$ and $Max'[f(I)] \leq q(Max[I], Length[I])$.

Then if π is ***NP-hard*** in the ***strong sense*** and $\exists$ a psuedo-polynomial transformation from π to π', then π' is ***NP-hard*** in the ***strong sense***.

Theorem 2.7 : All the problems PP's, QQ's and RR's, for each of the three distance metrics (l_1, l_2 and l_∞), are ***strongly*** NP-complete.

Proof : The magnitude of the largest number occurring in any instance I of the problems PP_i's (similar for QQ_i's and RR_i's) for the l_∞ distance metric, is determined by either the coordinate pairs of the destination set T or the parameters k, m or L. The parameters k and m are both integer values less than n. The constructions in Theorem 2.2 and Lemma 2.3, show the ***strong*** NP-completeness of PP_1, QQ_1 and RR_1, since what is exhibited is a ***bounded polynomial transformation*** from *Exact Cover* (a strong NP-complete problem [GJ79]). The lengths of the two problems are polynomially related and a polynomial of n. Further the function (the *iff* reduction mapping) is polynomial time computable. The integer coordinate pairs of the points of T are bounded by $O(n)$ and the value of L can be seen to be bounded by $O(n^3)$. Hence the maximum value occurring in the construction of the geometric location-allocation problems, is bounded by a polynomial in n and hence bounded by a polynomial of the maximum value and length of an instance of the Exact Cover problem (satisfying the conditions of the pseudo-polynomial transformation).

The reduction of [MS81] for the NP-completeness of problem PP_1 for the l_1 and l_2 distances, is a bounded transformation from 3 *Sat* (strongly NP-complete [GJ79]) and hence the strong NP-completeness for these problems follows as above. The strong completeness of the problems QQ_1 and RR_1 for the l_1 metric (Theorem 2.6) and for the l_2 metric [Pa81] follows again from their transformation from the Exact Cover problem.

The problems PP_2 and PP_3 (similarly for the corresponding QQ and RR problems), have m and k, respectively, as their decision parameters and are actually number problems, consequently strongly NP-complete. The strong completeness of these problems also follow from the direct transformation from the above PP_1 problems, (Lemma 2.1). □

We can now finally show that our (decision) optimization problems O_i's are Δ_2^P-***proper*** for ***infinite***, *finite* and ***discrete*** solution sets. [LM81] first showed how pure combinatorial optimization problems can be proven to be Δ_2^P-***proper***. We show how it applies to our case of geometric optimization problems.

Lemma 2.8 : The optimization problems O_i's are in Δ_2^P.

Proof : A problem P is in Δ_2^P if $P \leq_T^P$ (*Turing* reduces) to Q, where Q is in NP. For each of the O_i's, where the sources are located in the entire plane, we show that a Turing reduction exists, to the corresponding recognition NP-complete version PP_i's [AHU74]. (For those O_i's where the sources are located on a discrete set S or on the set of ***destination*** points T, the corresponding NP-complete versions QQ_i's and RR_i's are used respectively). For O_1, the value of parameter L in any instance I of the problem, lies in the range 0 to $c^{Length[I]}$, for some constant c. By a simple binary search one can find the optimum value of L in at most $\log c^{Length[I]} = O(Length[I])$ calls of the PP_1 oracle. Since $Length[I]$ is some polynomial in n, the reduction is polynomial time bound. For problems O_2 and O_3 with their respective parameters m and k, the parameter values range from 1 to n and so a sequential search suffices. □

Lemma 2.9 : If $NP \neq Co\ NP$, then the optimization problems O's are *not* in NP.

Proof : Each of the problems O_i, where the sources are located in the entire plane, is the optimization version of the corresponding NP-complete problem PP_i. Then the complement of PP_i, PP_i^c is a $Co\ NP$-complete problem. Here one answers the question 'there does not exist a solution of (parameter) q or better' or 'the best is of size *worse* than q, (*worse* is either *less* or *greater* depending on whether PP_i is a *maximum* or a *minimum* optimization problem). Then if one can show that $PP_i^c \leq_m^{NP} O_i$, (a *non deterministic polynomial time* many-one reduction), then $O_i \in NP \Rightarrow PP_i^c \in NP$, since $\leq_m^{NP}$ preserves the class of NP [LLS75]. However, if $NP \neq Co\ NP$, $PP_i^c \notin NP \Rightarrow O_i \notin NP$, that is, the optimization problems O_i's are *not* in NP. The trick is from [LM81].

For each of the O_i's the $\leq_m^{NP}$ reduction procedure guesses the optimal value of the parameter q, (L for O_1, m for O_2 and k for O_3), uses the O_i oracle to verify that q is the optimum and then checks to see if it is *worse* than the input q' to PP_i. Hence $PP_i^c \leq_m^{NP} O_i$ and the assertion follows. □

Lemma 2.10 : If $NP \neq Co\ NP$, then the optimization problems O's are *not* in $Co\ NP$.

Proof : The trick here is a double indirection [LM81], and is to show that $PP_i^c \leq_p^P O_i^c$, a *polynomial time positive* reduction which uses only the 'yes' answers of an oracle, as opposed to a Turing reduction which uses both 'yes' and 'no' answers of the oracle. Now $O_i \in Co\ NP$ iff $O_i^c \in NP$. But $O_i^c \in NP \Rightarrow PP_i^c \in NP$, since $\leq_p^P$ preserves

the class of NP. However, PP_i is NP-complete and so PP_i^c is $Co\ NP$-complete and if $NP \neq Co\ NP$, $PP_i^c \notin NP \Rightarrow O_i^c \notin NP \Rightarrow O_i \notin Co\ NP$.

For O_i, the fact that the corresponding PP_i problem is *strongly* NP-complete guarantees that $PP_i^c \leq_p^P O_i^c$, since this requires at most $Max[I]$ 'yes' answers of the O_i^c oracle to answer any instance I of PP_i^c. Since $Max[I] \leq$ some polynomial of n (because of strong NP-completeness, Theorem 2.7), the reduction is polynomial time bounded. (For those O_i's where the sources are located on a discrete set S or on the set of *destination* points T, the corresponding NP-complete versions QQ_i's and RR_i's are used respectively). □

The proof of our main theorem below, follows from the above Lemmas 2.8, 2.9 and 2.10.

Theorem 2.11 : The geometric location-allocation optimization problems, under MIN-SUM, for each of the three distance metrics as well as for *infinite* and *finite* feasible solution sets are all Δ_2^P-*proper*, assuming $NP \neq Co\ NP$.

Open Issues

(1) The problems for the squared *Euclidean* distance metric.

(2) A natural geometric optimization problem which is Δ_3^P-*proper*.

2.3. MINMAX Geometric Optimization Problems

A *minmax* location objective is one which minimizes the maximum cost resulting from a given location solution. Various applications have been raised in facility location theory [FW74],[KP79], where there exists sufficient justification in minimizing the effects of the worst situation. A common real world situation that might be formulated with a minimax objective is one of locating health clinics so that the maximum distance a patient must travel to a clinic is minimized. Another example concerns the placement of fire stations in a large metropolitan area such that the maximum distance between any location within the city and the nearest fire station is minimized.

Under this minimax location objective it is possible to distinguish two basic approaches that have been taken in the literature on facility location. The first suggests that a location site may be selected anywhere in the area of interest on the plane, giving an infinite number of possible location sites. The second approach considers only a finite number of known sites as feasible and models the constraints imposed on the possible location of sources, ensuring that undesirable and impractical locations need not be considered.

In this paper we analyze the complexity of certain geometric optimization problems which arise frequently in the above application areas. We are given a set $T=\{(x_i,y_i),\ i=1..n\}$, the location of n fixed destination points (destinations) in the plane and need to locate a number of points (sources) to service the destinations in a way which behooves the application on hand. Three different optimization questions are considered as P_1, P_2 and P_3 below, each under a *minmax* optimizing objective. Note that in the case of locating multiple sources, the optimal allocation of the destinations to the sources must also be determined - hence the name location-allocation problems [Co63]. On the assumption that each destination can be completely serviced by a single source, clearly in the optimal solution each destination is allocated to its closest located source.

Given the set T, as specified before, of n destinations in the plane

(P_1) Locate k points (sources) so as to *minimize* the *maximum* of the weighted distances between the destinations and the sources closest to them.

(P_2) Locate k points (sources) so that for a *maximum* number of destinations the weighted distances of these destinations from their closest sources does not exceed a prescribed limit R.

(P_3) Locate a *minimum* number of points (sources) so that the *maximum* of the weighted distances of the destinations and their closest sources does not exceed a prescribed limit R.

The weighted distances mentioned above come from a weight w_j assigned to each destination j and is some measure of the special cost of serving destination j in traveling from its closest source . It may be interpreted as the importance or probability of an emergency occurring and therefore will often be a function of the size of the destination. However in the following problems we assume that all weights are equal (similar to assuming that $w_j = 1$, for $j=1..n$) and show that even for this restricted case the above problems are quite difficult.

Problems P_1, P_2 and P_3 allow location of the sources to be anywhere in the plane. Let problems Q_1, Q_2 and Q_3 correspond to problems P_1, P_2 and P_3 respectively, with the location of the sources being restricted to a finite discrete set S of possible locations in the plane and of size polynomial in n. Note that all the above questions are of the form 'Locate the optimal ...'. However, they may also be stated in the traditional optimization version of 'Find the optimal ...', wherein we find the optimal number / size of the sources (and their location), to satisfy the requirements.

Between two points $p=(x_1,y_1)$ and $q=(x_2,y_2)$ in the plane the general l_p distance, for $1 \le p \le \infty$, is $[|x_1 - y_1|^p + |x_2 - y_2|^p]^{1/p}$; Some of the more common distance metrics used, *Rectilinear* (l_1), *Euclidean* (l_2), and *Infinity* l_∞ , reflect the appropriate travel restrictions for the emergency vehicles of the problem, e.g. ambulances, helicopters, fire engines etc..

2.3.1. The Euclidean distance metric

Under the ***minimax*** criterion with *Euclidean* l_2 distance metrics each of the above location-allocation optimization problems reduces in a direct fashion to the placement of equal radius circular discs (***circles***) on the plane, with the ***centers*** of the circles corresponding to the location of the ***sources***. Further all the destinations covered by the same circle correspond also in a direct fashion to an ***allocation*** of these destinations to the ***source*** (the center of the circle). Having equal weights for the above problems results in equal sized circles. Considering each of the above problems P's and Q's for equal sized ***circles*** which we call problems PC's and QC's we show that each of these problems are ***complete*** for the ***complexity class***[3], D^P. This then gives us an idea of the inherent computational complexity of the above problems P's and Q's under the ***Euclidean*** distance metric.

[3] The class of D^P was defined in [PY82] as follows : L is in D^P ***iff*** L is an intersection of L_1 and L_2 such that L_1 is in NP and L_2 is in $Co\text{-}NP$. D^P contains both NP and $Co\text{-}NP$ and is contained in $\Delta_2^P = P^{NP}$. Alternatively D^P can be defined as the class of all predicates $R(x)$ that can be expressed as $R(x) = [\exists y\ P(x,y)] \wedge [\forall z\ Q(x,z)]$ for some polynomially balanced and polynomial-time checkable P and Q.

We denote the Euclidean plane by E^2 and a circle ***locatable*** anywhere in E^2 means the center of the circle can be any point in the Euclidean plane. Furthermore we define an R ***circle*** to be a circle of radius R.

Given a set T of n fixed ***points*** in the plane,

(PC_1)

Is R the ***minimum*** radius of k equal sized ***circles*** locatable anywhere in E^2 to cover the n ***points*** of T ?

(PC_2)

Is m, ($m \le n$), the ***maximum*** number of ***points*** of T that k, R ***circles*** locatable anywhere in E^2 can cover ?

(PC_3)

Is k the ***minimum*** number of R ***circles*** locatable anywhere in E^2 to cover the n ***points*** of T ?

For the constrained location problems, besides the set T of n ***points*** in the plane, we are also given a finite discrete set of points $S \subset E^2$ and the location of the ***circles*** is constrained to be from this set. A circle ***locatable*** anywhere in S means the center of the circle is a point of this set S.

(QC_1)

Is R the ***minimum*** radius of k equal sized circles locatable anywhere in S to cover the n ***points*** of T ?

(QC_2)

Is m, ($m \le n$), the ***maximum*** number of ***points*** of T that k, R ***circles*** locatable anywhere in S can cover ?

(QC_3)

Is k the *minimum* number of R-*circles* locatable anywhere in S to cover the n *points* of T ?

For circles (circular discs) an alternate though similar set of optimization questions as above, may be asked. The *piercing* number for a set of circles is the number of 'needles' required to pierce all the circles of the system [HD64].

Given a set C of n fixed R-*circles* in the plane,

(DC_2)

Is m the *maximum* number of R-*circles* of C that k *needles* can pierce ?

(DC_3)

Is k the *minimum* number of *needles* required to pierce all n R *circles* of C ?

Note that in these problems we optimally locate 'needles' (points), in order to tag each of the given circles.

Theorem 2.12 : The problems DC_2 and DC_3 are ***dual*** problems of PC_2 and PC_3 (respectively) and are ***polynomial-time*** transformable to each other.

Proof : The duality arises from the fact that if the k R-*circles* with centers $\{c_i\}$, $i=1..k$, have a common intersection point then they can all be pierced by a single *needle*. Furthermore, a single R *circle* centered on any point of the common intersection can cover the k centers, $\{c_i\}$, $i=1..k$.

Given a set $T=\{n$ points in the plane$\}$, obtain the set C of n circles of radius R with centers lying the n points of T. Then m is the maximum number of R *circles* of

C that k needles can pierce ***iff*** m is the maximum number of points of T that k R-*circles* can cover. The proof follows, since for each subset S of C of R-*circles* that a *needle* pierces, an R *circle* centered on the piercing point of the needle can cover the centers (members of set T), of the R-*circles* of set S. Conversely, for each subset S of points, $S \subseteq T$, that are coverable by an R-*circle*, a *needle* pierces the subset of C of R *circles* having S as their centers.

Similarly k is the minimum number of *needles* required to pierce all n R-*circles* of C ***iff*** k is the minimum number of R *circles* required to cover the n points of T. □

We now show that the problem PC_3 of locating the minimum number of R *circles* in E^2 to cover all the n demand points is D^P ***complete*** by reducing $(Sat, UnSat)$, a known D^P ***complete*** problem [PY82], to it. We adapt certain constructions previously specified in [FPT81]. Next we show that all the above remaining problems are D^P complete by a series of polynomial time reductions. To show a problem to be complete for this class D^P we differ from [PY82] in that we use polynomial-time *positive* (disjunctive) reductions [LLS75] as opposed to polynomial-time *many one* reductions.

In a simplistic fashion any form of polynomial-time oracle-reducibility (which includes both *positive* and *Turing* reductions), is a Boolean formula of a polynomial number of queries to the oracle. The essential *restriction* for positive reductions is that the Boolean formula is a *positive* formula. Sufficient to our purpose here, the Boolean formula is positive if it only contains disjunctive ($\vee$) and conjunctive ($\wedge$) Boolean connectives. Furthermore, in positive reductions, like in other *truth table* reductions, one is restricted to a prespecified list $f(x)$ of queries, based on input x, from which alone one can make queries. These *restrictions* are *severe*. *First*, these restrictions allow only a polynomial number of *feasible* queries for polynomial-time positive reductions, while an exponential number of *feasible* queries exist for polynomial-time Turing reductions (a tree of polynomial depth). *Second*, an more importantly, in positive reductions one is restricted from the use of the negation (*not*) Boolean connective which disallows using the 'No' answer of the reduction oracle.

It is also important to note that the special case of disjunctive, conjunctive positive reductions which we use here are by far the strongest of the various other positive and truth-table reducibilities known [LLS75]. In turn, any truth-table reduction is stronger than a Turing reduction.

In [S82], it is proved that, similar to polynomial-time many-one reductions, polynomial-time positive reductions preserve the class of NP. That is if a language L_1 polynomial-time positive reduces (or polynomial-time many-one reduces) to a language L_2 then $L_2 \in NP \Rightarrow L_1 \in NP$. A similar fact is true for the class $Co\ NP$. Therefore these positive reductions are adequate to separate the class of D^P *complete* languages from the classes of NP and $Co\ NP$, (assuming $NP \neq co\ NP$). A similar argument is given when using polynomial-time Turing reductions as opposed to polynomial-time many-one reductions, in separating the class of NP *complete* languages from the class of P, since both Turing reductions as well as many-one reductions preserve the class of P. Hence, they are both adequate to separate the classes NP *complete* from P (assuming $P \neq NP$). It is also important to note that polynomial-time Turing reductions which do not preserve the class of NP, are not adequate in separating D^P languages from NP and $Co\ NP$. Thus, for instance, it is possible to polynomial-time Turing reduce the D^P *complete*

language ($Sat, Unsat$), to (Sat), a known $NPcomplete$ problem.

Theorem 2.13 : The problem PC_3 of locating the *minimum* number of R *circles* in E^2 to cover the given n points is D^P *complete*.

Proof : The problem is in D^P since it can be rephrased as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1,...,p_k) in E^2)[R$ *circles* with centers at $p_1,...,p_k$ cover n points$] \wedge (\forall(q_1,...,q_{k-1}) in E^2)[R$ *circles* with centers at $q_1,...,q_{k-1}$ cover $< n$ points$]$.

To prove the completeness we reduce (Sat, Unsat) to PC_3, using polynomial-time positive reductions.

Starting from (F_1, F_2) and adapting a polynomial time construction in [FPT81] (see *Appendix A*), we construct two separate sets of points S_1 and S_2 in the plane such that for $i=1,2$, exactly k_i, R *circles* are required to cover all the n_i points in S_i if F_i is satisfiable. Further if F_i is not satisfiable, at least k_i+1 and at most k_i+c_i, R *circles* are needed to cover all the n_i points of S_i, where c_i is the number of clauses in the CNF formula F_i.

Now construct c_2 additional copies of the set of points S_1. We now have (c_2+1) copies of sets of points S_1 and a single set of points S_2. It is important to note why (c_2+1) copies of S_1 are required. Let $n = (c_2+1)n_1 + n_2$. It is not hard to see that k, the minimum number of circles of radius R needed to cover all the n points, satisfies $(c_2+1)k_1 + k_2 + 1 \leq k \leq (c_2+1)k_1 + k_2 + c_2$ *iff* F_1 is satisfiable and F_2 is not satisfiable. Since this is a disjunction of at most c_2 calls of PC_3, problem PC_3 is D^P *complete* under a polynomial-time positive (disjunctive) reduction from (Sat, Unsat). □

Theorem 2.14 : The problem PC_2 of locating k R-*circles* in E^2, to cover a maximum number of the given points is D^P *complete*.

Proof : The problem is in D^P since it can be rephrased as before, as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1 ... p_k) in E^2)[R$ *circles* with centers at $p_1 ... p_k$ cover m points] $\wedge$ $(\forall(q_1 ... q_k) in E^2)[R$-*circles* with centers at $q_1 ... q_k$ cover $\leq m$ points].

To prove the completeness we again reduce (Sat, Unsat) to PC_3, using polynomial-time positive (disjunctive) reductions in a way very similar to above.

Starting from (F_1, F_2) we construct two separate sets of points S_1 and S_2 in the plane such that for $i = 1,2$, k_i, R *circles* are required to cover all the n_i points in S_i if F_i is satisfiable. Further, if F_i is not satisfiable, k_i, R *circles* can cover at least $n_i - c_i$ points and at most $n_i - 1$ points of S_i, where c_i is the number of clauses in the CNF formula F_i.

Now construct c_2 additional copies of the set of points S_1. We now have $(c_2 + 1)$ copies of sets of points S_1 and a single set of points S_2. It is important to note why $(c_2 + 1)$ copies of S_1 are required. Let $k = (c_2 + 1)k_1 + k_2$. It is not hard to see that m, the maximum number of points that can be covered by k circles of radius R, satisfies $(c_2 + 1)n_1 + n_2 - c_2 \leq m \leq (c_2 + 1)n_1 + n_2 - 1$ *iff* F_1 is satisfiable and F_2 is not satisfiable. Since this is a disjunction of at most c_2 calls of PC_2, problem PC_2 is D^P *complete*.

under a polynomial-time positive (disjunctive) reduction from (Sat, Unsat). □

Corollary 2.14.1 : The dual problems DC_2 and DC_3 are also D^P-*complete*.

Theorem 2.15 : The problem PC_1 of locating k equal sized circles of minimum radius to cover all the given n points is D^P *complete*.

Proof : The problem is in D^P, when R is restricted to integers[4], since it can be rephrased as before, as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1 ... p_k) in E^2)[R$ *circles* with centers at $p_1 ... p_k$ cover n points] $\wedge$ $(\forall(q_1 ... q_k) in E^2)[(R-1)$ *circles* with centers at $q_1 ... q_k$ cover $< n$ points].

To prove it complete we show that PC_3 polynomial-time positive reduces to PC_1. We construct a set S of the radii of all possible circles which minimally cover n points in the plane. Since the minimum enclosing circle for a set of points is defined by exactly two or three of the points, the total size of S is at most $\binom{n}{2} + \binom{n}{3}$ which is $O(n^3)$. We claim that k is the minimum number of R *circles* that cover all n points *iff* for some $s \epsilon S$, $s \leq R$, s is the minimum radius of k circles to cover all n points and for some $s \epsilon S$, $s > R$, s is the minimum radius of $k-1$ circles to cover all n points. The proof is straightforward and follows from the definitions of the two problems PC_1 and PC_3. Again since we have a conjunction of two sets of disjunctive calls of PC_1, (at most $O(n^3)$ calls), we have a polynomial-time positive reduction from PC_3 to PC_1. □

[4] Otherwise the problem appears to be D^P *hard* when R is in general, a real number.

Theorem 2.16 : The finite solution set problem QC_3 is D^p *complete*.

Proof : The problem is in D^p since it can be rephrased as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1,...,p_k)\epsilon S)[R$ *circles* with centers at $p_1,...,p_k$ cover n points$]$ $\wedge$ $(\forall(q_1,...,q_{k-1})\epsilon S)[R$ *circles* with centers at $q_1,...,q_{k-1}$ cover $<n$ points$]$.

To prove it complete we prove that PC_3 polynomial-time many-one reduces to QC_3. It suffices to show that for any set T of n destination points in the plane there exists a finite set $S \subset E^2$, such that if a minimum of k, R *circles* can cover T then these R *circles* can be chosen to have their centers in S. Furthermore, S must be constructible in time polynomial in n.

We claim one can choose such an $S = T \bigcup$ {intersection points of R *circles* centered at the points of T}. For a proof of this claim let F be a (minimal) set of circles of radius R covering T and let circle $C \epsilon F$. If C contains only a single point $p \epsilon T$, replace C by an R *circle* centered at $p \epsilon T \subset S$. Otherwise, if C contains more than one point, move C without uncovering any point of T, until two points $p,q \epsilon T$, lie on the boundary of the moved circle C'. Clearly the center c of C' lies at an intersection of the R *circles* centered at p and q. Thus $c \epsilon S$. (A similar duality was proved and exploited in [BL83]).

Finally note that S contains at most $O(n^2)$ points and can be constructed in $O(n^2)$ time. □

Theorem 2.17 : The finite feasible solution set problem QC_2 is D^p *complete*.

Proof : The problem is in D^p since it can be rephrased as before, as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1,...,p_k)\epsilon S)[R$ *circles* with centers at $p_1,...,p_k$ cover m points$]$ $\wedge$ $(\forall(q_1,...,q_k)\epsilon S)[R$ *circles* with centers at $q_1,...,q_k$ cover $\leq m$ points$]$

To prove it complete we exhibit a polynomial-time reduction from PC_2 to QC_2 similar to the proof of Theorem 2.16. Again construct the set of points $S = T \bigcup$ {intersection points of R *circles* centered at the points of T}. As before there is no loss of coverage and it suffices to consider only the set of points S as possible location centers; n remains the maximum number of points covered by k, R *circles* if n is the maximum number of points covered by locating the k, R *circles* anywhere in E^2. □

Theorem 2.18 : The finite feasible solution set problem QC_1 is D^p *complete*.

Proof : The problem is in D^p, when R is restricted to integers, since it can be rephrased as before, as the conjunction of a predicate in NP and a predicate in $Co\ NP$: $(\exists(p_1,...,p_k)\epsilon S)[R$ *circles* with centers at $p_1,...,p_k$ cover n points$]$ $\wedge$ $(\forall(q_1,...,q_k)\epsilon S)[(R-1)$ *circles* with centers at $q_1,...,q_k$ cover $<n$ points$]$.

To prove it complete we exhibit a polynomial time reduction from PC_1 to QC_1 similar to the proofs of Theorems 2.16 and 2.17. Again construct the set of points $S = T \bigcup$ {intersection points of R *circles* centered at the points of T}. It suffices to see that there is no loss of coverage in considering only the set S as possible location centers and R remains to be the minimum radius of the k circles covering the n points if R is the minimum radius of the k circles locatable anywhere in E^2 to cover the n points. □

2.3.2. The (l_p, $1 \le p \le \infty$) distance metrics

With the general l_p distance metrics each of the above location-allocation optimization problems reduces to the placement of *equal sized*[5] geometrical figures.

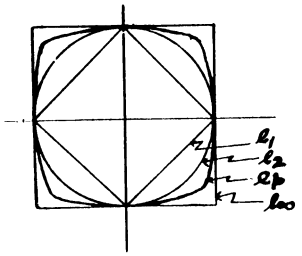


Figure 2.7.

all oriented the same way, and with the intersection points of their xx' and yy' axes, corresponding to the location of the sources.

For the *rectilinear* l_1 distance metric the geometric figures involved in our above optimization problems P's and Q's are equal sized *diamonds* (squares rotated by 45°) of half-diagonal length R instead of R-*circles*. For the *infinity* l_∞ distance metric the optimization problems reduce to placement optimization problems of equal sized squares of half-edge length R, having sides parallel to the respective coordinate axes. Also a diamond (square) *locatable* anywhere in S or E^2 means the intersection point of the diagonals of the diamond (square) can be any point in the finite discrete set S or the Euclidean plane respectively.

[5] Again assuming equal weights for the destinations results in our considering each of the previous problems P's and Q's for equal sized geometrical figures. The *unit discs* are given by $|x|^p + |y|^p = 1$

For all these problems both membership and completeness for the class of D^P carry over in a fashion quite similar to the proofs of Theorems 2.13 to 2.18. Each of the constructions used here as well as the adapted constructions of [FPT81], can be modified in a direct fashion for these geometric figures with fixed orientations. Thus we have the following result.

Theorem 2.19 : The above P_1, P_2, P_3 and Q_1, Q_2, Q_3 optimization problems, under the general l_p distance metric and versed as placement optimization problems, (decision versions), for the corresponding geometric figures are all D^P *complete*.

2.3.3. Arbitrary distance metrics

Between two points p and q in the plane, the various properties of a distance metric $d(p,q)$ are

(1) *translation invariance* - $d(p,q)$ is a function of the relative positions of p and q.

(2) *symmetry* - $d(p,q) = d(q,p)$.

(3) *triangle inequality* - $d(p,r) \le d(p,q) + d(q,r)$.

(4) *positivity* - $d(p,q) \ge 0$.

(5) *homogeneity* - for $o=(0,0)$, $p=(x,y)$ and $q=(ax,ay)$, $d(o,q)=a*d(o,p)$.

Each of the l_p, $1 \le p \le \infty$, distance metrics satisfy the above properties. The Euclidean (l_2) distance metric is the only distance which is also invariant under *rotation*. That is the reason why in Section 2.3.2, above, a specific fixed orientation was imposed on the geometric figures for the general l_p metric.

A point set S is ***symmetric*** with respect to the origin if for each point (x,y) belonging to S, the point $(-x,-y)$ also belongs to S. A point set is ***convex*** if for each pair of points in S, the line segments joining these points is entirely in S.

Theorem 2.20 [BB61]: If S is a point set containing the origin in its interior and S is ***convex*** and ***symmetric*** with respect to the origin, then there exists a distance function d for which S is the ***unit disc***.

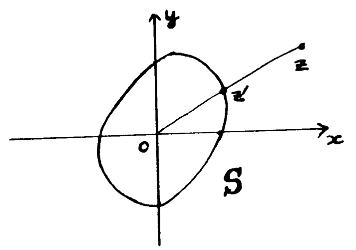


Figure 2.8.

For the above ***unit disc***, S, which is both ***convex*** and ***symmetric***, define the distance from o to z to be $d(oz) = oz / oz'$. The distance $d(oz)$ is ***less*** than 1 if z is an interior point of S, ***equal*** to 1 if z is a boundary point of S and ***greater*** than 1 if z is an exterior point of S. For these unit discs we specify a certain fixed orientation, since they are not invariant under rotation. All the constructions can again be adapted as before, at times with closer tolerances, thereby providing reductions which show that the corresponding optimization problems for the placement of these ***convex*** and ***symmetric*** geometric figures to be D^P ***complete***. Note that both the convexity and symmetric

properties of these figures are essential in the reduction of the infinite solution set problems P's to the finite solution set problems Q's as in Theorems 2.16, 2.17 and 2.18. The duality used in these Theorems apply to all convex figures having a point of symmetry.

Conclusion

We can also claim that all of the above problems are ***strongly D^P complete*** analogous to the similar concept for *NP-complete* languages, since all the above constructions hold even when the largest number occurring in any instance of the problems, that is parameter R and the coordinate points in set T, are restricted to be of size bounded by a polynomial in n.

Acknowledgements Sincere thanks to Ming Li, Paul Pritchard, Raimund Seidel and Yaacov Yesha for their corrections and suggestions.

Open Issues

(1) Are the above geometric optimization problems D^P-***complete*** under ***many one*** reductions?

(2) Closure properties of the class D^P

2.3.4. Appendix A

The figure provides an overall view of the construtiou. Each of the liues is a wire, a set of points arranged so that any pair of successive poiuts along a wire cau be covered by one circle and any given circle can cover no more thau two points along a wire. Corresponding to each variable in the formula is a closed loop of wire which contains an even number of points. There are two possible coverings, one of these corresponds to au assignment of *true* and the other designates *false*.

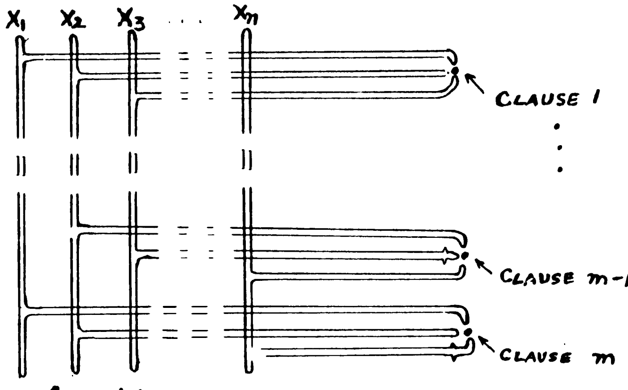


Figure 2.9.

On embedding the construction in the plane, a suitable crossover is designed for the loops that cross on another. At the crossover the two loops share a circle such that the value assignments of both loops are propagated independently through the crossover.

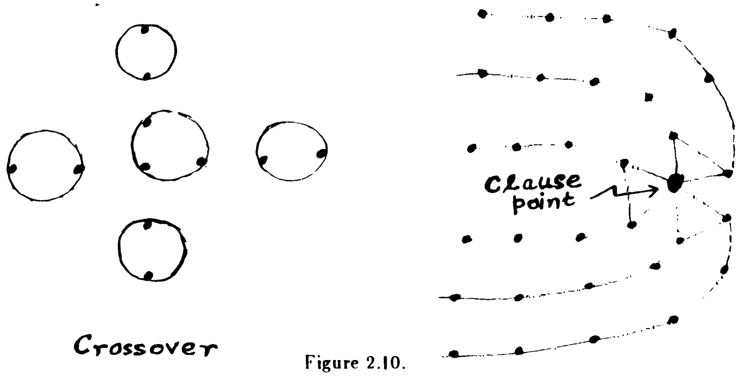


Figure 2.10.

For each of the clauses there is a point located in a region in which the three variable loops corresponding to the literals of the clause are brought into close proximity. The loops are arranged so that the clause point can be connected 'for free', if at least on of the three wires is in the state corresponding to the truth of the literal in the clause. (If the literal is a negated variable, the variable loop is adjusted so that the clause point can be covered for free by the covering corresponding to the assignment of false to the variable). If none of the literals is true then an extra circle is needed to cover the clause point. If all the clauses are simultaneously satisfied then no additional circles are necessary to cover the clause points. The construction is therefore coverable by $k\ R$ *circles* if the formula is satisfiable.

CHAPTER 3

EFFICIENT ALGORITHMS AND LOWER BOUNDS

3.1. Background

The topic of this chapter is the investigation of the geometric location-allocation optimization problems [Co63] of the previous chapter for the case when the allocation is predetermined. For then the optimum location of k sources reduces to k separate problems of locating single sources, amongst the n given fixed destinations. At this stage it might also be worth mentioning that when the location of the sources is known, the determination of the optimal allocation is not too difficult. By brute force it takes $O(nm)$ time and by use of the closest point Voronoi diagram [SH75], for k sources the time is reduced to $O(n \log k)$.

The model on which we base our standard operations is the real RAM [AHU74], a random access machine with infinite precision, real number arithmetic. It performs addition, subtraction, multiplication, division, comparison and square-roots on real numbers in constant time. To develop fast algorithms we use this *real* RAM, since most of the above operations are available on modern day computers and moreover all of them take approximately the same unit time. Furthermore the size of the parameters and more importantly the size of the intermediate results of the various problems considered, does not play a critical role in any of the algorithms of this chapter. Where the size of the intermediate results of the problem does become important, we use a different model. Hence we used a RAM model with *logarithmic* cost criterion [AHU74] for the results of Chapter 2.

The model of computation used to prove the worst-case lower bounds is the *decision tree* model. Although this model is less interesting from the computation point of view, many worst-case lower bounds have been proved for this. In this model algorithms are presented as trees, in which every vertex of the tree has the form of a comparison $f(inputs) : 0$, where f is some function from a class of allowed functions. For *linear* decision trees several powerful techniques are known [DL75],[Ya75]. A method is given in [Be83] for obtaining lower bounds for *bounded degree algebraic* decision trees, where the functions allowed are polynomials of bounded degree. The lower bounds of this chapter comply to both the above linear and algebraic decision tree models. Furthermore they also hold for the *algebraic computation* tree model, [Be83], which can handle the operation of square-roots along with the ordinary arithmetic functions.

For the problem of Chapter 1, we use a more restricted model than the *real* RAM, namely, the *integer* RAM model without the square-root operation, since the assumption of a constant time square-root operation with infinite precision would be falacious. Any such simplifying assumption would in fact change the problem itself, the complexity of which arises due to minimizing the sum of terms with square-roots (a nonlinear optimization problem).

In the following sections, let $f(n)$, $g(n)$ be functions from the positive integers to the positive reals, denoting,

(1) $f(n) = O(g(n))$ if $\exists c > 0$ *such that* for large n, $f(n) \le cg(n)$.

(2) $f(n) = \Omega(g(n))$ if $\exists c > 0$ *such that* for large n, $f(n) \ge cg(n)$.

(3) $f(n) = \Theta(g(n))$ if $\exists c, c' > 0$ *such that* for large n, $cg(n) \le f(n) \le c' g(n)$.

3.2. Geometric Location

In describing the wide variety of queueing problems, Wagner [see FW74] states "What is said about birds is also true of waiting line models: their variety and number seem infinite ". The same is becoming true for geometric optimization problems. Such problems frequently arise in facility location, clustering analysis, VLSI layout design, and other cad/cam, robotics applications.

In this paper we consider the problem of determining the location of a single new facility (*source*) with respect to a number of existing users (*destinations*). The location we seek is that which *minimizes* an appropriately defined total cost function. We consider the cost to be proportional to the distance. A number of single facility location problems exist and are amenable to the analysis presented here. Some of the more traditional examples are the location of a

(1) A new lathe in a machine tool workshop. (*minsum*)

(2) A welding robot in an automobile manufacturing plant. (*minsum*)

(3) A new warehouse relative to production centers and customers. (*minsum*)

(4) A hospital or a fire station in a metropolitan area. (*minmax*)

(5) A cleansing pump in a chemical processing setup. (*minmax*)

(6) A component in a VLSI design chip. (*minsum*)

(7) A water fountain in an office building. (*minmax*)

These examples suggest a reason for the interdisciplinary interest in facility location.

We also consider some related geometric optimization problems that stem from the above application areas and devise both efficient algorithms and non-linear lower bounds

for them. Since nearly all of these problems are recast in a mathematical framework bordering on Euclidean geometry these optimization problems are mathematically intriguing in their own right.

The discrete optimization criteria *minmax* (minimizing the maximum cost) and *minsum* (minimizing the sum of costs), are two of the more important location objectives that arise in numerous applications as above. The distance metrics used *Rectilinear* (l_1) and *Euclidean* (l_2) reflect the appropriate restrictions on 'travel' between the sources and destinations, for the application under consideration. It is also possible to distinguish two basic approaches that have been taken in the literature on facility location. The first suggests that a location site may be selected anywhere in the area of interest on the plane, giving an *infinite* number of possible location sites. The second approach considers only a *finite* number of known sites as feasible and models the constraints imposed on the possible location of sources, ensuring that undesirable and impractical locations need not be considered.

Here we consider two different point geometric location problems in the plane. Given a set $T=\{(x_i,y_i),\ i=1..n\}$, the location of n fixed destination points (*destinations*) in the plane, under *minmax* optimization, we have the problems.

(P_1) Locate a point (*source*) so as to *minimize* the *maximum* of the distances between the source and the *destinations*.

(P_2) Locate a point (*source*) so that for a *maximum* number of *destinations* the distances of these destinations from the source does not exceed a prescribed limit R.

The location of the source in the above problems P_1 and P_2, can be anywhere in the plane, an *infinite* feasible solution set. A corresponding set of questions, which we call Q_1 and Q_2, are constrained to have the source located on some element of the set T of destinations. This models the *finite* feasible solution set approach mentioned above.

A similar set of questions, P's and Q's, can be written under the *minsum* optimization criterion.

3.2.1. Locating under MINMAX

Such a *minmax* criterion can be considered as a prototype for many location formulations where some measure of the largest cost of serving the fixed destinations is to be minimized. This arises mostly in the location of emergency facility location such as in the location of hospitals and fire stations.

(P_1) *Locate a source anywhere in the plane, so as to minimize the maximum distance from the n destinations.*

Euclidean distances. $\Theta(n)$ time.

The problem reduces to the geometric problem of finding the *minimum enclosing* circle of the n destinations in the plane, with the center of the circle corresponding to the location of source. In 1-D the solution is simply the mid-point between the minimum and the maximum points on the line and is computable in $\Theta(n)$ time [AHU74]. In 2-D, the problem of finding a minimum enclosing circle was apparently first posed in 1857 by Sylvester, who also presented a geometrical solution in 1860 attributed to Pierce. An $O(n^2)$ solution to the problem was given in [EH72] which was later improved to $O(nlogn)$ by [SH75]. Finally an optimal $\Theta(n)$ solution was presented in [Me82].

Rectilinear distances. $\Theta(n)$ time.

In principle a similar geometric approach applies as for the l_2 metric. The problem reduces to determining the locus of all points equidistant from a given point, here a *diamond* (square rotated by 45° from coordinate axes) and by replacing the concept of the minimum enclosing circle by a *minimum enclosing* diamond. This can again be computed in $\Theta(n)$ time [EH72].

(Q_1) *Locate a source on one of the n destinations, so as to minimize the maximum distance from the other destinations.*

Euclidean distances. $O(nlogn)$ time.

Using the farthest point voronoi diagram for euclidean distances, which can be constructed in $O(nlogn)$ time [SH75], and for each of the n destination points find the destination point farthest from it and the distance between them. The *minimum* of these distances gives the result.

In 1-D the solution turns out to be the destination point closest to the $(minimum(T) + maximum(T))/2$, where T is the set of n destinations on the line. Further this can be computed in $\Theta(n)$ time.

(Also the smallest of the distances between lines of support of the convex hull of the n destination points. Again $O(nlogn)$ time.)

Rectilinear distances. $O(nlogn)$ time.

A similar approach using the farthest point voronoi diagram for rectilinear distances. The closest point Voronoi diagram for the l_1 metric can be constructed in

$O(nlogn)$ time [LW80]. The farthest point diagram can be constructed in an analogous procedure.

(P_2) *Locate a source anywhere in the plane, so that a maximum number of the n destinations have their distances from the source* $\leq R$.

Euclidean distances. $O(n^2 logn)$ time. $\Omega(nlogn)$

(Actually $O(nmlogm)$, where $m =$ maximum number of points covered by a circle of radius $2R$).

The problem reduces to finding the maximum number of points that can be covered by a circle of radius R (R *circle*). A ***dual*** problem can also be stated as follows, given n R *circles* in the plane, 'Find the maximum value of k for which there exists a k-intersection, the common intersection of k circles'. The problems are dual since a solution for one is a solution of the other and vice versa.

A Ω $(nlogn)$ lower bound under the ***linear*** decision tree model is shown here by a reduction from the ***set element uniqueness*** problem which has been shown to have an Ω $(nlogn)$ lower bound under the same model, in [DL75], and also under the ***algebraic computation*** tree and ***bounded degree algebraic*** decision tree models, in [Be83]. The reduction follows directly, wherein the elements of the ***set*** are mapped to points in the plane which for a limit $R=0$ returns a maximum of 1 ***iff*** the elements of the set are distinct.

In 1-dimension where circles of equal radius R degenerate to equal line segments (or intervals) of length R on a line the problem can each be solved in $O(nlogn)$ time by sorting the left-endpoints of the n intervals and scanning the points on the line in linear time. The lower bound of $\Omega(nlogn)$ also applies to the 1-D problem in an identical

fashion to above, by mapping the elements of the set to points on a line and asking the question for a limit $R=0$. Again the answer is a maximum of 1 ***iff*** the elements of the set are distinct. Hence for 1-D the problem P_2 is $\Theta(nlogn)$.

Lemma 3.1 : There are at most 2 circles of a given radius R. (R *circle*) which pass through 2 given points.

Proof : If we travel clockwise along the circumference of the circle passing through the two points, then either a minor arc connects the two points or a major arc ; giving rise to at most two positions of the R *circle*. The case of a unique R *circle* is when the two points are exactly $2R$ apart and form the diameter of the R *circle*. □

Theorem 3.2 : In 2-D the problem P_2 can be solved in $O(n^2 logn)$ time.

Proof : For each of the centers p_i, i=1..n we do the following

(1) Let M_i be the set of points within the euclidean distance $\leq 2R$ of p_i. [The size of M_i is at worst n]. Define the "$\odot$" order between two points p_j and p_k to be, p_j "$\odot$" p_k if there is an R *circle* (circle of radius R), passing through points p_i and p_k such that the point p_j is 'below' the arc $p_i.p_k$. Sort the set of points M_i according to this ordering relation. {This is similar to sorting by polar angle with p_i as origin, wherein the points p_j and p_k are compared with respect to the line $p_i.p_k$ as opposed to the arc of a circle of radius R as is done above. Thus the definition of p_j 'below' arc $p_i.p_k$ is : On rotating the arc of a circle of radius R in an anti-clockwise direction about a point p_i, the point p_j intersects the arc 'before' the point p_k). On the basis of Lemma 3.1 we can test the "$\odot$" relation in $O(1)$ time giving a worst case $O(nlogn)$ sort-

ing time for the set M.

(2) Now for each of the points $p_j \in$ M, determine the number of points enclosed by the R *circle* passing through p_i and p_j. We do this by sweeping in an anti-clockwise direction around the point p_i. Since these points have been sorted the entire operation takes at most $O(n)$ time. {This is done by keeping two lists sorted as above, one with respect to the minor arc p_i, p_j and the other for the corresponding major arc for updating points entering the circle and points leaving the circle respectively, during the anti-clockwise sweep }. The maximum number of points is thus obtained by this step.

As steps (1) and (2) are repeated for each of the centers of the n circles the total time for the entire operation is $O(n * nlogn) = O(n^2 logn)$. [Note that here one does not obtain a single sorting for all the points as in the 1-dimension case and hence a suitable sorting has to be done for each point in turn giving rise to the above time complexity].

□

For the case of spheres in 3-dimensions, three points are sufficient to specify a sphere of fixed radius R, and rotation sorting can now done with respect to 2 points at a time. For general hyperspheres in d-dimensional space, the sorting is done with respect to d-1 points giving an $O(n^d logn)$ method for all choices of d-1 points. This proves the following theorem.

Theorem 3.3 : For n points in d-Dimension space, the problem of finding the *maximum* number of points that can be enclosed by a hypersphere of radius R, can be solved in $O(n^d logn)$ time in the worst case.[1]

---

[1] A more exact analysis gives a time bound of $O(nm^{d-1} logm)$ where m = the maximum number of points close enough to be covered by a hypersphere of radius $2R$.

Rectilinear distances. $\Theta(nlogn)$ time.

This follows from considering the dual problem of, 'Finding the maximum value of k for which there exists a k-intersection, the common intersection of k *diamonds* of half-diagonal width R'. The dual problem is solved in [IA83] with the above time bound of $O(nlogn)$. The lower bound $\Omega(nlogn)$, is by reduction from the set element uniqueness problem as above.

(Q_2) *Locate a source on one of the n destinations, so that a maximum number of the other destinations have their distances from the source* $\leq R$.

Euclidean distances. $O(n^2)$ time. $\Omega(nlogn)$

Exhaustive search where for each point one finds in $O(n)$ time the number of points $\leq R$ distance from it. The lower bound is by reduction from the set element uniqueness problem as before.

The problem in 1-D is solved by a linear scan, similar to the 1-D problem of P_2. The time taken is optimal, $\Theta(nlogn)$ as mentioned in the P_2 case.

One could do 'better' by using algorithms for *'fixed radius near neighbour'* problems, wherein the running time is $O(nlogn + nk)$, k = maximum number of points reported in the queries. However nearly all the *'near neighbour'* techniques are constrained by a large initial set up time of the query data structure, [CY83]. Furthermore since k could be $O(n)$ in the worst case, we have the above complexity.

The lower bound is by reduction from the set element uniqueness problem as before.

Rectilinear distances. $\Theta(nlogn)$ time.

Actually $O(nlogn + k)$, where k = maximum number of points reported to lie within a *diamond* of half-diagonal length R. The problem is solved by a *'batched range search'* technique of [BW80]. For each of the n destination points we have a square with edge length $\sqrt{2}R$, whose 'center' coincides with the destination point. Then using the range search technique, the set of destination points are organised into a data structure and for each square, queries are made for the list of all destination points that are contained in it. The total time taken is given by the above time bound. The fact that our original problem for the rectilinear distance metric reduces to diamonds of half-diagonal length R is easily settled by rotating the entire coordinate plane containing the destination points by an angle of 45°, transforming it to squares of edge length $\sqrt{2}R$.

The lower bound $\Omega(nlogn)$ is by reduction from the set element uniqueness problem as before.

3.2.2. Locating under MINSUM

Such a *minsum* criterion can be considered as a prototype for many location formulations where some measure of the total or average cost of serving the fixed destinations is to be minimized.

(P_1) *Locate a source anywhere in the plane, so as to minimize the sum* of the distances from the n *destinations*.

Euclidean distances.

The objective function is strictly convex when the destinations are not collinear and convex in the case of collinearity. In fact the problem in the plane is convex for all l_p distance metrics. Thus a local minimum is a global minimum. However, this somewhat celebrated problem is very difficult and the only known solutions are iterative procedures and heuristic algorithms [FW74],[KP79]. A dynamic programming solution is also given in [Be65]. In [Chapter 4] we show that any exact method reduces to a minimal polynomial whose roots are not expressible in terms of radicals (and hence the solution would have to be approximated).

The solution to the problem is however simple for some special cases [Ei62]. For example, the problem reduces to the rectilinear case below, when the n destination points are collinear and can be solved in $\Theta(n)$ time. This is also the problem in 1-D.

Rectilinear distances. $\Theta(n)$ time.

In many problem formulations the l_1 distance measure is preferred, as perhaps in locating a new warehouse in Manhattan, or a component on a VLSI chip. Also this is the metric of use in various 3 D problems dealing with the installation of pipes, electrical cables in a building. Also more importantly, this measure is much easier than the l_2 metric because of its separability.

From the problem formulation, given n destinations (b_i^x, b_i^y), $i=1..n$, we need to $minimize_{(x,y)\in E^2} \sum (|b_i^x - x| + |b_i^y - y|)$ it is evident that the objective function is separable so that one can solve the two parts independently. Namely, $minimize_x \sum |b_i^x - x|$ and $minimize_y \sum |b_i^y - y|$. This is, to find separate medians for the x and y coordinates of the $\{b_i\}$ points in the plane. This follows from the fact that if one is above or below the median point then one can reduce the sum of distances by moving towards the median point. That the median can be obtained in $O(n)$ time is shown in [AHU74].

(Q_1) *Locate a source on one of the n destinations, so as to minimize the sum* of the distances from the other *destinations*.

Euclidean distances. $O(n^2)$ time.

Exhaustive search where for each point compute the sum of the distances to all other points and select the *minimum* of these distances.

The problem in 1-D is similar to the 1-D problem of P_1 and can be solved in $\Theta(n)$ time.

Rectilinear distances. $O(n^2)$ time.

Exhaustive search as for the euclidean distances. Note that here the problem is not separable, like the corresponding P_1, since here the x and y coordinates of the solution point are not independent but belong to the same destination solution point.

(P_2) *Locate a source anywhere in the plane, so that for a maximum number of the n destinations the sum* of the distances from the source $\leq R$.

Euclidean distances. $\Omega(nlogn)$

As difficult as the celebrated P_1 problem for *Euclidean* distances.

The lower bound is again shown by reduction from set element uniqueness where the set of elements are mapped to points in the plane and for a limit $R=0$ the

maximum $=1$ iff all the elements of the set are distinct.

Rectilinear distances. $O(n^3logn)$ time. $\Omega(nlogn)$

In Section 2.2 of Chapter 2, we showed that the question of locating a *source* anywhere in the plane, under the *minsum* criterion from n destinations was equivalent to locating the source on one of the l_1 *grid points*. Hence our geometric problem reduces to a finite combinatorial problem of selecting one of the $O(n^2)$ *grid points* (all points made up from the x and y coordinates of the n destination points), for the optimal solution. For each of these l_1 *grid points*, we sort the distances from it to the n destinations in $O(nlogn)$ time and add these distances in turn from the sorted list till the sum exceeds R. The *maximum* number of distances added with respect to any of the *grid points*, gives the result.

The lower bound is by reduction from element uniqueness as above.

(Q_2) *Locate a source on one of the n destinations, so that for a maximum number of the other destinations the sum* of the distances from the source $\leq R$.

Euclidean distances. $O(n^2logn)$ time. $\Omega(nlogn)$

For each point b_i sort the $n-1$ distances $d(b_i,b_j)$ $j=1..n$ and $j \neq i$. Add the distances with respect to b_i, in turn from this sorted list till the sum exceeds R. The *maximum* number of distances added, with respect to any point, gives the result.

The lower bound is by reduction from element uniqueness as above.

Rectilinear distances. $O(n^2logn)$ time. $\Omega(nlogn)$

Same as for the euclidean case. The lower bound is by reduction from element uniqueness as above.

3.3. Related Geometric Optimization Problems

The problem of intersections of geometric objects has been seen to be related to that of placement of geometric objects with respect to certain fixed pointsand hence to the problems of geometric location. When the objects become unequal sized, then containment of a smaller geometric object (say a circle) within a larger circle is also considered an intersection. A geometric optimization question that immediately arises, concerning these intersections, is the question of the maximum containment of one object within another.

[Maximum Containment] Given n circles of unequal radius r_i in the plane, what is the maximum level k of containment (maximum number of circles completely within one another) ? List the k circles in the order in which they are contained within each other.

Theorem 3.4 : The circle containment problem takes $\Omega(nlogn)$ time.

Proof: Given a sequence of elements $\{x_i\}$, $i=1..n$, construct n circles with centers at the origin (say) and having radii $\{x_i\}$. The maximum level of containment is n and a listing of the n circles in the order in which they are contained is a sort of the sequence $\{x_i\}$. □

The problem is $\Omega(nlogn)$ even in 1-dimension, where given n intervals on a line of unequal length I_i we wish to list in order those intervals which enclose each other, conforming to the maximum level of containment. A similar reduction to the one above

shows this to be true (construct intervals with their left endpoints all starting at the origin). The containment problem can also be reduced to a directed acyclic graph with vertices $\{v_i\}$ corresponding to circles $\{p_i\}$ and directed edges $< v_i, v_j >$ iff circle p_i is contained in circle p_j. The circle containment problem then reduces to finding the longest directed path in the acyclic graph.

Theorem 3.5 : In 1-dimension the containment problem can be solved in $O(nlogn)$ time and this is *optimal*.

Proof: Again only a smaller sized interval can be contained in a larger interval. For intervals I_i, I_j with left and right endpoint (l_i, r_i) and (l_j, r_j), respectively, I_i is contained in I_j if $l_j \leq l_i$ and $r_i \leq r_j$. Sort the $2n$ endpoints of the n intervals. Scan the intervals in a left to right order. On encountering a left point l_i of an interval I_i we add this interval name to a separate list L by order of its left endpoint l_i. On encountering a right endpoint r_j, corresponding to a left endpoint l_j of the interval we determine how many intervals in list L have their left endpoints before l_j. This can be done in $O(logn)$ time via binary search on list L. The total time taken is bounded by $O(nlogn)$.

□

Theorem 3.6 : The circle containment problem can be solved in $O(n^2)$.

Proof: Given n circles with centres p_i of radius $\{r_i\}$, i=1,2,...,n. Sort the radius $\{r_i\}$ into non-decreasing order. A circle with centre p_i with smaller radius r_i is contained in a circle with centre p_j with larger radius r_j if $d(p_i, p_j) \leq r_j \;\; r_i$. Now consider the circles in increasing order of radii, and for each circle check to see if any if the circles before it (with smaller radius) are contained within it. For each circle one remembers the maximum number and the corresponding list of circles it contains. At the end of the scan of n circles, one obtains the circle with the maximum number of circles contained within it. Clearly the total time taken is $O(n^2)$. □

Corollary 3.6.1 : This method is easily generalised to unequal sized hyperspheres for dimensions 3 and higher, as well as to other geometric objects.

Open Issues

(1) Optimization problems under the l_∞ and circumradial metrics.

(2) Optimization problems in euclidean space (E^3), especially the location problems for the l_1 and l_∞ distance metrics.

(3) Problems of location for $k=2$. For what k do the problems become *NP-hard*.

(4) Optimization problems on line location or points constrained to lie on a line.

CHAPTER 4

ALGEBRAIC COMPLEXITY OF GEOMETRIC OPTIMIZATION

4.1. Introduction

The use of algebraic methods to geometric problems is not new. However their use in better understanding geometric optimization problems is yet to be fully realized. We take a step in this direction. We reduce certain geometric optimization problems to algebraic problems. By generating the minimal polynomial, whose root over the field of rational numbers is the solution of the geometric optimization problem on the real (Euclidean) plane, we are able to answer questions about the solution of the problem by radicals. The algebraic degree of the optimizing solution, which is the degree of the irreducible minimal polynomial for the problem, (and the order of the splitting field of the minimal polynomial which is also the order of the appropriate Galois group), gives an idea of the inherent difficulty of constructing the solution as well as gives a complexity measure for these geometric optimization problems.

In this paper we consider two fundamental problems in geometric optimization which are non-discrete in nature. Their solutions are real, (*algebraic*[1]), unique and one of an infinite number of feasible solutions. The first of these problems is popularly known as the *Fermat-Weber* problem and is that of obtaining the optimum solution of a single *source* point in the plane, so that the sum of the Euclidean distances to n fixed *destination* points is a minimum. We also consider a variant of this problem, namely, the

[1] A real number α is *algebraic* over the rationals, if there is a polynomial $p(y)$ with rational coefficients which has α as a root.

Line restricted Weber problem, wherein we wish to obtain the optimum solution for the *source* point, now constrained to lie on a certain given *line*, and minimizing the sum of the distances from the *source* to the set of *destination* points, as before.

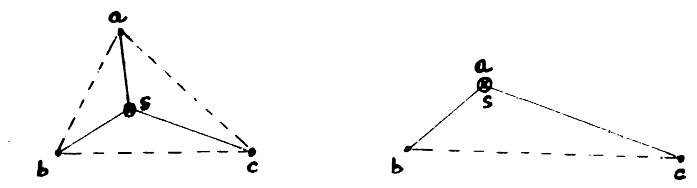
4.2. The Fermat-Weber problem

The Problem

Given n fixed *destination* points in the plane with coordinates (a_i, b_i), determine the location (x, y) of a single *source* point such that the sum of the Euclidean distances from this *source* to each of the *destinations* is minimized.

$$minimize_{x,y} \quad f(x,y) = \sum_{i=1..n} \sqrt{(x-a_i)^2 + (y-b_i)^2} \quad (1)$$

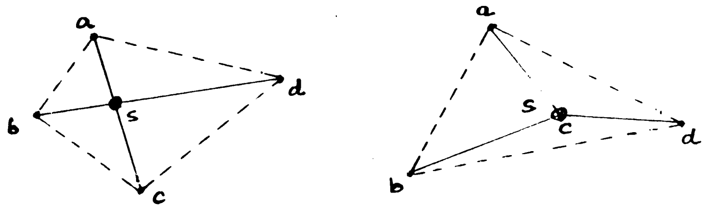
The above problem has been considered for quite a long time and has an equally long and interesting history. The problem for the case of $n=3$ was first formulated and thrown out as a challenge by Fermat as early as in the 1600's [Ku67]. Cavalieri in 1647 considered the problem for this case, in particular, when the three points form the vertices of a triangle and showed that each side of the triangle must make an angle of 120° with the given minimum point. Heinen in 1834 noted that in a triangle which has an angle of $\geq 120°$, the vertex of this angle itself is the minimum point.



(a) Triangle with angles < 120°　(b) Triangle with an angle ≥ 120°

Figure 4.1.

Fagnano in 1775 showed that for the case $n=4$ when the four client points form a convex quadrilateral the minimum solution point is the intersection of the diagonals of the quadrilateral. For a non-convex quadrilateral the fourth point which is inside the triangle formed by the three other points, is itself the minimum point.

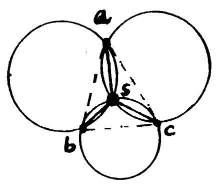


(c) Convex quadrilateral　(d) Non-convex quadrilateral

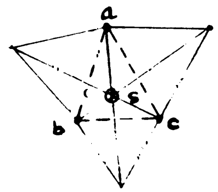
Figure 4.2.

Tedenat in 1810 found that for the case of n points the necessary condition for the minimum solution point is that the sum of cosines of the angles between any arbitrary line in the plane and the set of lines connecting the n given points with the minimum point must be zero. Later, Steiner in 1837 proved that the necessary and sufficient conditions for the minimum solution are that the sum of the cosines and sines of the above mentioned angles must be zero. Weber [We37], was probably the first who formulated this problem in light of the location of a plant, with the objective of minimizing the sum of transportation costs from the plant to sources of raw materials and to market centers. Hence this problem for n points has also come to be known as the *Generalized Weber* problem.

The constructions for the solution points for the case of 3 points is worthy of note. The solution is variously obtained by the Steiner construction or the Simpson construction.



Steiner point



Simpson point

Figure 4.3.

Here, we begin with the case of 5 points. We show that for nearly all configuration of 5 points (convex and non-convex), namely, the convex hull of the configuration having 3, 4 or 5 points on its boundary, in general the solution is a very high degree algebraic point. Further we can show that not only is the solution not constructible by straight-

edge and compass[2], the solution is not expressible in terms of *radicals*[3].

We also show that the non-expressibility of the solution by radicals also applies to the *Line restricted Weber* problem, in fact even for the simplest (though non-trivial) case of 3 destination points. A proof of the impossibility of straight-edge and compass constructions for the Weber problem (but not the line-restricted case) appears in [ME73], however nothing is known about the non-expressibility of the solution by radicals for either of the two problems.

Algebraic Reduction

The function $f(x,y)$ specified in (1) above can be shown to be strictly convex. A sufficient set of conditions for the function $f(x,y)$ to be convex is

$$(i)\ p=(d^2f/dx^2)_{x=x_0}>0 \qquad (ii)\ q=(d^2f/dy^2)_{y=y_0}>0$$

$$(iii)\ pq-r^2>0 \qquad \textit{where} \quad r=(d^2f/dxdy)_{x=x_0,y=y_0}$$

and (x_0,y_0) are solutions from the equations $df/dx=0$ and $df/dy=0$. The above conditions are quite easily met for the function $f(x,y)$ of (1). Hence there exists a unique minimum solution for which the necessary and sufficient conditions are $df/dx=0$ and $df/dy=0$. The corresponding rational functions are

$$df/dx=\sum_{i=1..n}(x-a_i)/\sqrt{(x-a_i)^2+(y-b_i)^2}=0 \quad (2)$$

[2] Straight-edge and compass constructions are equivalent to the solution being expressible in terms of $(+,-,*,/,\sqrt{})$ over Q, the field of rationals.

[3] A real number α is expressible in terms of *radicals* if there is a sequence of expressions $\beta_1,\ldots,\beta_n$, where $\beta_1\epsilon Q$, and each β_i is either a rational or the sum, difference, product, quotient or the k^{th} *root* of the preceding β's and the last β_n is α. Note that there exists α : α is algebraic but not expressible in terms of radicals.

$$df/dy=\sum_{i=1..n}(y-b_i)/\sqrt{(x-a_i)^2+(y-b_i)^2}=0 \quad (3)$$

By a process of repeated squaring one can eliminate all the square-roots from the expressions (2) and (3) above. Starting with say a sum of of n different square-roots, sqrt(i), $i=1..n$, equated to a constant, the technique is to take all terms of sqrt(i), for a certain i, to one side of the equation and the remaining terms on the other side, squaring both sides and thereby eliminating sqrt(i). Repeating this process by again isolating one of the remaining square-roots and squaring, one is able to eliminate all square-roots from the original equation in a maximum of n steps.

For the case of 5 points, let (a_i,b_i), $i = 1..5$ be the given points with integer coordinates. In the following we work over Q, the field of rational numbers. We make a tacit assumption that the solution does not coincide with any of the destination points and obtain the corresponding polynomials $f_1(x,y) = 0$ and $f_2(x,y) = 0$ from (2) and (3) respectively, after rationalizing and elimination of the square-roots. The resulting minimal polynomial for the minimization problem is

$$p(x,y) = f_1(x,y)^2 + f_2(x,y)^2 = 0 \quad (4)$$

Now that we have the above polynomial $p(x,y)$ for the problem our first step is to prove it irreducible, over Q. We show this by substituting constants for x, $x=c$ and showing that $p(c,y)$ is irreducible[4]. However the fact that that the minimal polynomial $p(x,y)$ is irreducible is important to us only if the line determined by $x=c$ passes through the solution point of our optimization problem. This is important and the reader should understand the reason. Consequently, we choose symmetric configurations

[4] If $p(x,y)$ is reducible then the corresponding $p(c,y)$ is also reducible. Hence if $p(c,y)$ is irreducible for some constant $x=c$ it implies that $p(x,y)$ is irreducible.

of 5 points, symmetric about a line $x=c$, for then we know that the solution lies somewhere on $x=c$. Then for a set of n points distributed equally and symmetrically about our chosen axis $x=c$, (when n is odd, 1 point lies on this axis), problem (1) is transformed to the following problem,

$$minimize_y \quad g(y) = \sum_{i=1..n}\sqrt{(y-a_i)^2+(c-b_i)^2} \quad (5)$$

We choose a configuration of 5 points symmetric about the line $x=0$. One of the points p lies on the line and has coordinates (0,c) on the $x=0$ axis. The value of c changes the configuration of points in that for c=5,1 and 4 we have respectively 3, 4 or 5 points on the convex hull of our set of 5 points, (figure 4.4).

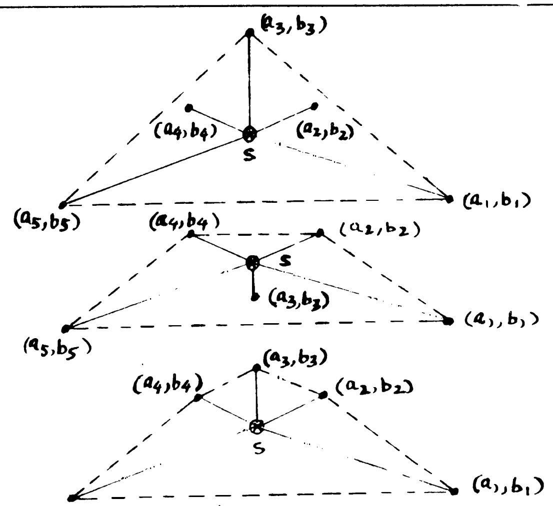


Figure 4.4.

Let $(a_1,b_1)=(3,0)$, $(a_2,b_2)=(1,3)$, $(a_3,b_3)=(0,c)$, $(a_4,b_4)=(-1,3)$ and $(a_5,b_5)=(-3,0)$.

We need to find the solution $(0,y)$ satisfying the condition for minimality, $df/dy=0$, giving us the following,

$$\mathit{minimize}_y \quad f(y)=y-c + 2\sqrt{(y-3)^2+1} + 2\sqrt{y^2+9}$$

$$df/dy=1 + 2(y-3)/\sqrt{(y-3)^2+1} + 2y/\sqrt{y^2+9} = 0$$

Eliminating square-roots we obtain the polynomial, $p(y)$, (*Table* 1). We note that this polynomial $p(y)$ is the minimal polynomial for each of the 3 configurations (i),(ii) and (iii) above, since as long as $y \neq c$[5], the function $df/dy=0$ is the same regardless of $c=5,1$ or 4.

Table 1

$$Q : p(y)=15y^8-180y^7+1030y^6-4128y^5+11907y^4$$
$$-15876y^3-17928y^2+75816y-54756=0$$

$$Disc(p(y)): \qquad 2^{52}3^{25}5^8 13^5 17^2 13063$$

$$Mod\ 19: \ p(y)=(y+7)(y^2-9y-4)(y^5+9y^4+8y^3+7y^2-4y-1)=0$$

$$Mod\ 31: \ p(y)=(y^8-12y^7-14y^6+10y^5-6y^4+8y^3-11y^2-11y-11)=0$$

$$Mod\ 37: \ p(y)=(y+5)(y^7-17y^6+18y^5-10y^4+15y^3-16y^2+17y+4)=0$$

[5] The case $y=c$ occurs when the point $\mu=(0,c)$, coincides with the intersection of the lines between $(a_1,b_1),(a_4,b_4)$ and $(a_2,b_2),(a_5,b_5)$, which is also the solution for the case of those four points, (a_i,b_i), $i=1,2,4$ and 5.

We now use Galois theoretical methods to prove the properties of interest. For a definition of the terms used here see [He75],[Ga71].

Lemma 4.1: The polynomial $p(y)$, [*Table* 1], is irreducible over Q.

Proof : Since $p(y)$ is irreducible mod 31 and the prime 31 is not a divisor of of the discriminant of the polynomial, $disc(p(y))$, it follows that $p(y)$ is irreducible over Q. □

The degree testing algorithm, see [Kn81], is a much more efficient means of proving irreducibility than merely searching for a prime p for which $p(y)$ is irreducible mod p. By performing the factorization of the polynomial $p(y)$ modulo several primes, and considering the possible degrees of the factors, one can obtain important information about the degree of the true factors. For good primes p relative to $p(y)$, {primes p that are not divisors of $disc(\mu(y))$}, one computes the degree set d_p=set of degrees of all factors of $p(y)$ mod p. The degree set of $p(y)$ must be contained in $d_{\mu_1} \cap \cdots \cap d_{\mu_m}$, where $\mu_1,\ldots,\mu_m$ are the primes tried. If $\mu(y)$ is irreducible over Q, often $d_{p_1} \cap d_{\mu_2} \cap \cdots = (0,n)$ after only a few primes have been tried.

Before we show the impossibility of constructions with straight-edge and compass we need a few definitions. The set of numbers *constructible* by straight-edge and compass form a subfield E of the field of real numbers. Since $1 \in E$, E must contain Q, the field of rational numbers. Henceforth when we refer to constructions we mean constructions with straight-edge and compass. A field F is said to be an *extension* of Q if F contains Q and a *simple extension*[6] if $F=Q(\alpha)$ for some $\alpha \in F$. Using the notation of

[6] Every extension of Q is a simple extension.

[He75], we denote $[F:Q]$=degree of F over Q, (the dimension of F as a vector space over Q).

Consider all the points (x,y) in the real Euclidean plane, both of whose coordinates x, y are in Q. This set of points is called the plane of Q. A point is *constructible* from Q *iff* we can find a finite number of real numbers $\alpha_1, \ldots, \alpha_n$ such that (i) $[Q(\alpha_1):Q]=1$ or 2 and (ii) $[Q(\alpha_1, \ldots, \alpha_i):Q(\alpha_1, \ldots, \alpha_{i-1})]= 1$ or 2, and such that our point lies in the plane of $Q(\alpha_1, \ldots, \alpha_n)$. It follows that if α is *constructible* then α lies in some extension of Q, of degree a power of 2.

We know that a real number α is algebraic over Q *iff* $Q(\alpha)$ is a finite extension of Q. Further n is said to be *algebraic* of *degree* n over Q if it satisfies a non-zero polynomial of degree n but no non-zero polynomial of lower degree. Also if α is algebraic of degree n over Q, then $[Q(\alpha):Q]=n$. This together with our discussion of constructibility above gives the following important criterion for non-constructibility.

Lemma 4.2 : [He75] If the real number α satisfies an irreducible polynomial over Q of degree n and if n is not a power of 2, then α is not constructible.

If $p(y) \in Q[y]$, a finite extension E of Q is said to be a *splitting field* over Q for $p(y)$ if over E but not over any proper subfield of E, $p(y)$ can be factored as a product of linear factors. Alternatively, E is a *splitting field* of $p(y)$ over Q if E is a *minimal* extension of Q in which $p(y)$ has n roots, where n=degree of $p(y)$.

Given a polynomial $p(y)$ in $Q[y]$, the polynomial ring in y over Q, we shall associate with $p(y)$ a group, $Gal(p(y))$, the Galois group of $p(y)$. The Galois group turns out to be a certain permutation group of the roots of the polynomial. It is actually defined as a certain group of automorphisms of the *splitting field* of $p(y)$ over Q. A beautiful

duality, expressed in the fundamental theorem of Galois Theory exists between the subgroups of the Galois group and the subfields of the splitting field. From this one can derive a condition for the solvability by means of radicals of the roots of a polynomial in terms of the algebraic structure of its Galois group. As a special case we can give a criterion for non-constructibility by straight-edge and compass constructions, which follows from the above criterion.

Lemma 4.3: If E is the splitting field over Q for an irreducible polynomial $p(y)$, and if the order of its galois group, $o[Gal(p(y))] = [E:Q]$, is not a power of 2, then the roots of $p(y)$ are not *constructible*.

A few additional theorems from Galois theory of use to us here are,

Lemma 4.4: [Ga71] For a finite field F, $|F|=p^n$ and $p(y) \in F[y]$ factors over F into k different irreducible factors, $p(y)=q_1(y)\ldots q_k(y)$, where degree $q_i(y)=n_i$, then $Gal(p(y))$ is *cyclic* and is *generated* by a permutation containing k cycles with orders $n_1, \ldots, n_k$.

The *shape* of a permutation of degree n is the partition of n induced by the lengths of the disjoint cycles of the permutation. The factorization of a polynomial modulo any prime p also induces a partition, namely the partition of the degree of $p(y)$ formed by the degree of the factors. The above *Lemma* states that the degree partition of the factors of $p(y)$ modulo p is the shape of the generating permutation of the group $Gal(p(y))$, which is furthermore cyclic.

Lemma 4.5: [Ga71] Let $p(y) \in Z[y]$ have roots $\alpha_1, \ldots, \alpha_n$ and let $p^*(y) \in Z_p[y]$, be the polynomial $p(y)$ mod p. If the roots of $p^*(y)$ are $\alpha_1^*, \ldots, \alpha_n^*$, then the

$Gal(p^*(y))$, {as a permutation group of its roots} is a subgroup of $Gal(p(y))$.

Theorem 4.6 : The solution of the *Fermat - Weber* problem, in general, is not *constructible* by straight-edge and compass for $n \geq 5$.

Proof : Restating the assertion, we have to show that the roots of the polynomial $p(y)$ of *Table* 1 are not constructible by straight-edge and compass. We know that $p(y)$ is irreducible over Q from *Lemma* 4.1. Consider $p^*(y)$ as the polynomial $p(y)$ mod p for a ***good*** prime $p=37$ relative to $p(y)$. From *Table* 1 the irreducible factors of $p^*(y)$ have degrees 1 and 7. On application of *Lemma* 4.4 we know that for the finite field Z_{37}, the $o[Gal(p(y))] = 7$ and from *Lemma* 4.5, it is a divisor of $o[Gal(p(y))]$, which clearly is not a power of 2 and hence *Lemma* 4.3 proves our assertion. □

To prove the ***non expressibility*** of the roots of $p(y)$ over Q by radicals we use the *Ceboratev Van der Waerden* sampling method to determine the Galois group of $p(y)$, [Mc79],[Za71]. From the density theorem of Ceboratev one obtains,

Lemma 4.7: As $s \rightarrow \infty$, the proportion of occurrences of a partition π as the degree partition of the factorization of $p(y) mod\ p_i$, $(i=1..s)$, tends to the proportion of permutations in $Gal(p(y))$ whose shape is π.

In order to then apply this method of guessing the group of the polynomial over Q one needs a table of permutation groups of the desired degree, along with a distribution of its permutations, [St73]. The degree of concern for the polynomial $p(y)$ of *Table* 1 is 8. From [Mi99] we know that there are exactly 200 permutation groups of degree 8. However all is not lost. We also know that polynomial $p(y) \epsilon Q$ is irreducible ***iff*** the Galois

group, $Gal(p(y))$ is ***transitive***[7], [Vw53], and there are only 50 transitive groups of degree 8.

If the Galois group of the polynomial is the symmetric group, S_n, (the group of all permutaions of [1..n]), the Ceboratev-Van der Waerden method realizes the fact quickly. Indeed,

Lemma 4.8: [Za71] If $n \equiv 0(mod\,2)$ and $n > 2$ then after sampling about $(n+1)$ good primes we run across an $(n-1)$-cycle and an n-cycle and a permutation of the type $2+(n-3)$ and that will be enough to establish that $Gal(p(y))$ over Q is the symmetric group S_n. If $n \equiv 1(mod\,2)$ then one will run across an $(n-1)$ cycle and a permutation of the type $2+(n-2)$ in about the same time and that will be enough.

Proof : Since for $n \equiv 0(mod\,2)$, $n-3$ is odd, the permutation type $2+(n-3)$ when raised to a power (n-3) yields a 2-cycle. This together with the $n-1$ cycle and the n cycle generate the symmetric group S_n as follows. Let $(12...n-1)$ be the $n-1$ cycle. By virtue of transitivity, the 2-cycle (ij) can be transformed into (kn), where k is one of the digits between 1 and $(n-1)$. The transformation of (kn) by $(12...n-1)$ and its powers yield all cycles $(1n)(2n)...(n-1n)$ and these cycles together generate the symmetric group, S_n [Vw53].

For $n \equiv 1(mod\,2)$, again as $n-2$ is odd, the permutation type $2+(n-2)$ when raised to a power $(n-2)$ yields a 2-cycle, which together with the $n-1$ cycle generates the symmetric group as above. □

[7] A permutation group on $1..n$ is called ***transitive*** if for any k, $1 \leq k \leq n$, it contains a permutation π which sends 1 to k.

Usually the decision that $Gal(p(y))=S_n$ is reached even after much less than $n+1$ trials as a consequence of the evolving pattern of permutations occurring in $Gal(p(y))$ and the application of known theorems of permutation groups.

We are now ready to prove our main theorem, but first let us indulge in some definitions. A polynomial $p(y)\epsilon Q[y]$ is called ***solvable*** over Q if there is a finite sequence of fields $Q=F_0<F_1< \cdots <F_k$, (where $F_{i-1}<F_i$ implies that F_{i-1} is a subfield of F_i) and a finite sequence of integers $n_0,...,n_{k-1}$ such that $F_{i+1}=F_i(\alpha_i)$ with $\alpha_i^{n_i}\epsilon F_i$ and if all the roots of $p(y)$ lie in F_k, that is, $E\subseteq F_k$, where E is the splitting field of $p(y)$. F_k is called a ***radical extension*** of Q. Furthermore, we know from Galois theory that

Lemma 4.9 : [He75] $p(y)\epsilon Q[y]$ is solvable by radicals over Q ***iff*** the Galois group over Q of $p(y)$, $Gal(p(y))$ is a solvable group.

Lemma 4.10 : [He75] The symmetric group S_n is not solvable for $n\geq 5$.

Theorem 4.11: The *Fermat Weber* problem, in general, is not solvable by radicals over Q for $n\geq 5$.

Proof : Restating the assertion, we need to show that the polynomial $p(y)$ of *Table* 1 is not solvable by radicals over Q. We note from ***Table*** 1 that for the 'good' primes $p=19,31$ and 37, the degrees of the irreducible factors of $p(y)$ mod p gives us a 2 + 5 permutation, an 8 cycle and a 7 cycle, which is enough to establish, from Lemma 4.8 for $n=8$, that $Gal(p(y))=S_8$, the symmetric group of degree 8. *Lemma* 4.10 tells us that this is not a solvable group and hence our assertion follows from *Lemma* 4.9. □

4.3. The Line-restricted Weber problem

We now consider the *Line -restricted Weber* problem.

Given n fixed ***destination*** points as before in the plane with coordinates (a_i,b_i), we need to determine the location (x,y) of a single ***source*** point, restricted to lie on certain given ***line***, such that the sum of the Euclidean distances from this ***source*** to each of the ***destinations*** is minimized.

We consider two different positions (and orientations) of this line, since the algebraic degree of the solution point varies with the relative positions of the line and the fixed destination points. For the non-trivial case of 3 destination points consider the solution restricted to a line passing through one of the points and ***either*** not intersecting the convex-hull (of the destination points), ***or*** passing through the convex-hull, figure 4.5.

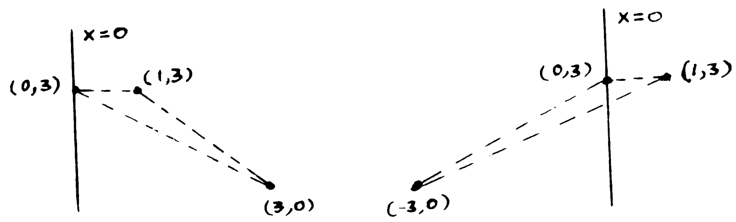


Figure 4.5.

Lemma 4.12 : For the above cases of figure 4.5, the minimal polynomial $p(y)$ (*Table* 2) of degree 8 is irreducible over Q. Furthermore, this polynomial is not solvable by radicals over Q.

Table 2

$$\text{minimize}_y \; f(y)=y - c + \sqrt{(y-3)^2+1} + \sqrt{y^2+9}$$

$$df/dy=1 + (y-3)/\sqrt{(y-3)^2+1} + y/\sqrt{y^2+9} = 0$$

$$Q: \; p(y)=3y^8-36y^7+202y^6-780y^5+2277y^4-4212y^3+3402y^2-81=0$$

$$disc(p(y)): \qquad 2^{28}3^{23}5^{6}13^{3}19687$$

$$Mod\ 7: \; p(y)=(y^8+2y^7+2y^6-y^5+3y^4+3y^3+1)=0$$

$$Mod\ 11: \; p(y)=(y-5)(y^7+4y^6+3y^5-3y^4-4y^3-5y^2-2y+1)=0$$

$$Mod\ 29: \; p(y)=(y+13)(y^2+5y-11)(y^5-y^4+12y^3+11y^2+2y+1)=0$$

Proof: Since $p(y)$ is irreducible mod 7, for a 'good' prime 7, it follows that $p(y)$ is irreducible over Q. To show non-solvability by radicals we apply *Lemma* 4.8 for $n=8$ and note from *Table* 2 that for the 'good' primes $p=7,11$ and 29 the degrees of the irreducible factors of $p(y)$ mod p gives us an 8 cycle, a 7 cycle and a 2 + 5 permutation which is enough to establish that $Gal(p(y))=S_8$. Again, *Lemma* 4.10 tells us that this is not a solvable group and hence our assertion follows from *Lemma* 4.9. □

As before, for the case of $n=3$ destination points consider the solution restricted to a line however, not passing through any of the 3 points and *either* not intersecting the convex-hull (of the destination points), *or* passing through the convex-hull, figure 4.6.

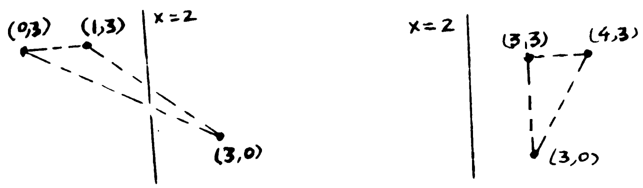


Figure 4.6.

Lemma 4.13: For the above cases of figure 4.6, the minimal polynomial $p(y)$ (*Table* 3) of degree 12 is irreducible over Q. Furthermore, this polynomial is not solvable by radicals over Q.

Proof: Since $p(y)$ is irreducible mod 7, for a 'good' prime 7, it follows that $p(y)$ is irreducible over Q. One notes that the impossibility of straight-edge and compass constructions follows immediately from *Lemma* 4.2, since the degree of $p(y)$ is 12 which is not a power of 2. To show the non-solvability by radicals, we again apply *Lemma* 4.8 for $n=8$ and note from *Table* 2 that for the 'good' primes $p=7,19$ and 61 the degrees of the irreducible factors of $p(y)$ mod p gives us a 12 cycle, a 11 cycle and a 2 + 9 permutation which is enough to establish that $Gal(p(y))=S_{12}$, the symmetric group of degree 12. *Lemma* 4.10 tells us that this is not a solvable group and hence our assertion follows from *Lemma* 4.9. □

Theorem 4.14: The *Line restricted Weber* problem, in general, is not solvable by radicals over Q for $n \geq 3$.

Table 3

$$\text{minimize}_y \quad f(y)=\sqrt{(y-3)^2+4} + \sqrt{(y-3)^2+1} + \sqrt{y^2+1}$$

$$df/dy=(y-3)/\sqrt{(y-3)^2+4} + (y-3)/\sqrt{(y-3)^2+1} + y/\sqrt{y^2+1} = 0$$

$$Q : p(y)=3y^{12}-72y^{11}+780y^{10}-4992y^9+20772y^8-58500y^7+113610y^6$$
$$-155448y^5+156912y^4-119040y^3+51876y^2+972y-729=0$$

$$disc(p(y)) : \qquad 2^a 3^b 5^c 13^d p$$

$$Mod\ 7 : \ p(y)=(y^{12}-3y^{11}+y^{10}+2y^9+y^8+2y^7-2y^5+3y^3+2y^2+2y+2)=0$$

$$Mod\ 19 : \ p(y)=(y-6)(y^{11}+y^{10}+8y^8-y^7+7y^6+7y^5+y^4+3y^3-9y^2+5y-7)=0$$

$$Mod\ 61 : \ p(y)=(y+13)(y^2-3y+10)$$
$$(y^9+27y^8+19y^7+19y^6-7y^5+y^4-10y^3-25y^2-21y+23)=0$$

Proof : Follows from Lemmas 4.12 and 4.13. □

We mention in passing that for the case of the line passing through 2 of the 3 given destination points, the solution to the *Line restricted Weber* problem coincides with projection of the 3^{rd} point onto that line and so is constructible.

Furthermore, the case of $n=5$ for the symmetric Fermat-Weber problem is equivalent to the (weighted) case, $n=3$, of the Line-restricted Weber problem, where the *line* is the axis of symmetry, which passes through one of the destination points, (and hence the algebraic degree of the solutions are the same). On the other hand the above

case of $n=3$ of the Line-restricted Weber problem where the line does not pass through any of the destination points is equivalent to the case of $n=6$ for the symmetric Fermat-Weber problem, (the line becoming the axis of symmetry as before). The solutions of these cases are as expected, of higher algebraic degree.

Conclusions

Restating the open problem of [LM83], the best thing one would like is an efficient way of computing the Galois group of a polynomial. Having obtained the Galois group one can show various wonderful properties of the geometric optimization problem on hand. Notably, if the polynomial is solvable one can obtain the tower decompositon of the solvable Galois group and thereby obtain the various extension fields of the rationals, where the roots of the polynomial lie. Calculating the k^{th} order radical in which the solution can be expressed is another possibility.

N.B. : In producing the examples of this paper we used the computer algebra system Macsyma, (actually Vaxima on Unix).

Acknowledgements : Thanks to John Hopcroft for his early inspiration which led to this paper.

CHAPTER 5

REFERENCES

[AHU74]
Aho, A., Hopcroft, J. and Ullman, J., *The Design and Analysis of Computer Algorithms*, Addison Wesley, 1974.

[Ba84]
Bajaj, C., *Reducibilities among Geometric Location-allocation Optimization Problems*, Computer Science Tech. report, Cornell University, TR84-607, 1984.

[Be65]
Bellman, R., *An Application of Dynamic Programming to Location-allocation Problems*, SIAM Review, 7, p126-128, 1965.

[Be83]
Ben-Or, M., *Lower bounds for Algebraic Computation Trees*, Proceeding of the 15th STOC, p80-86, Boston, 1983.

[Bu15]
Burns, J.E., *Abstract Definition of Groups of Degree Eight*, Amer. Journal of Math. 37, p195-214, 1915.

[BB61]
Beckenbach, E. and Bellman, R., *An Introduction to Inequalities*, Random House, 1961.

[BL83]
Bajaj, C. and Li, M., *On the duality of intersections and closest points*, Computer Science Tech. report, Cornell University, TR83-568, 1983.

[BW80]
Bentley, J. and Wood, D., *An Optimal Worst Case Algorithm for Reporting Intersection of Rectangles*, IEEE Transactions on Computers, vol c-29, no 7, July 1980.

[Co63]
Cooper, L. *Location-Allocation Problems* Operations Research Vol.11, No. 3, pp331-343, 1963.

[Co64]
Cooper, L., *Heuristic methods for location-allocation problems*, SIAM Review 6, p37-53, 1964.

[Co67]
Cooper, L., *Solution of Generalised Locational Equilibrium Models*, J. of Regional Science, vol 7, 1967.

[CR41]
Courant, R., and Robbins, H., *What is mathematics ?* Oxford University Press, 1941.

[CS69]
Curry, G., and Skeith, R., *A dynamic programming algorithm for facility location and allocation*, AIIE Transactions, p133-138, 1969.

[CY83]
Cole, R. and Yap, C., *Geometric Retrieval Problems*, 24th Annual FOCS, p112-121, Nov. 1983.

[DL75]
Dobkin, D. and Lipton, R., *On the complexity of computations under varying sets of primitives* in Excursions into geometry, Yale Univ. research report #71, 1975

[Ei62]
Eiseman, K., *Optimum Location of a Center*, SIAM Review, vol 4, no 4, p394-395, October 1962.

[EH72]
Elzinga, J. and Hearn, D., *Geometrical solutions for some minimax location problems* Transportation Science 6, p379-394, 1972

[El73]
El-Shaeib, A.M., *A new algorithm for locating sources among destinations*, Management Science, 20, p221-231, 1973

[FPT81]
Fowler, R., Paterson, M. and Tanimoto, S., *Optimal packing and covering in the plane are NP-complete*, Info. Proc. Letters, Vol 12, no 3, 1981.

[FW74]
Francis, R. and White, J., *Facility layout and location - an analytic approach*, Prentice Hall, NJ, USA 1974.

[Ga71]
Gaal, L., *Classical Galois Theory with Examples*, Markham Publishing Company, 1971.

[GGJ76]
Garey, M., Graham, R. and Johnson, D., *Some NP-complete geometric problems*, Proc. 8th STOC, p10-27, 1976.

[GJ79]
Garey, M. and Johnson, D., *Computers and Intractability : a guide to NP-completeness*, Freeman, San Francisco 1979.

[Ha63]
Haley, K.B., *The Siting of Depots*, International Journal of Production Research, 2, p41-45, 1963.

[He75]
Herstein, I.N., *Topics in Algebra*, 2^{nd} edition, John Wiley & sons, New York, 1975.

[HD64]
Hadwiger, H. and Debrunner, H., *Combinatorial geometry in the plane*, (Translated by Klee, V.), Holt, Rhinehart & Winston, 1964.

[HPT80]
Hansen, P., Perreur J. and Thisse J., *Location Theory, Dominance and Convexity :*

Some further results, Opern. Res., vol 28, p1241-1250 (1980).

[HU79]
Hopcroft, J. and Ullman, J., *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, 1979.

[IA83]
Imai, H. and Asano, T., *Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane* J. of Algorithms, 4,4, p310-323, Dec. 1983.

[Kn81]
Knuth, D.E., *The Art of Computer Programming, Vol 2: Seminumerical Algorithms*, 2nd ed., Addison Wesley, Reading, MA, 1981.

[Ku67]
Kuhn, H.W., *On a pair of dual non-linear programs*, Non-Linear Programming, J. Abadie ed., p37-54, North Holland, 1967.

[KP79]
Krarup, J. and Pruzan, P., *Selected families of discrete location problems*, Annals of Discrete Math. 5, North Holland, 1979.

[KS72]
Kuenne, R.E. and Soland, R.M., *Exact and approximate solution to the multisource Weber problem*, Math Programming, 3, p193-209, 1972

[LLS75]
Ladner, R., Lynch, N. and Selman, A., *Comparison of Polynomial Time Reducibilities*, Theoretical Computer Sci 1, p103-123, 1975.

[LM81]
Leggett, E.W. and Moore, D.J., *Optimization Problems and the Polynomial Hierarchy*, Theoretical Computer Sci 15, p279-289, 1981.

[LM83]
Landau, S., and Miller, G.L., *Solvability by Radicals in Polynomial Time*, Proceedings of the 15th Annual STOC, p140-151, 1983

[LW80]
Lee, D.T. and Wong, C.K., *Voronoi diagrams in L_1 (L_∞) metrics with 2-D storage applications* Siam J. of Comp., p200-211, Feb 1980.

[Mr79]
Mckay, J., *Some Remarks on Computing Galois Groups*, Siam J. of Comp., vol 8, no 3, p344-347, 1979.

[Me73]
Melzak, Z.A., *Companion to Concrete Mathematics* John Wiley & sons, New York 1973.

[Me82]
Megiddo, N., *Linear-Time algorithms for Linear Programming In R^3 and related problems* 23rd Annual FOCS, p329-338, November 1982.

[Mi58]
Miehle, W., *Link - Length Minimization in Networks*, Opern. Res., 6, 232-243, 1958

[Mi99]
Miller, G.A., *Memoir on the substitution groups whose degree does not exceed eight*, Amer. Journal of Math., 21, p287-337, 1899.

[MS84]
Megiddo, N. and Supowit, K.J., *On the complexity of some common geometric location problems*, Siam J. on Computing, vol 13, no 1, p182-196, Feb 1984.

[NP82]
Nievergelt, J. and Preparata, F., *Plane Sweep Algorithms for Intersecting Geometric Figures*, CACM, vol 25, no 10, October 1982.

[Pa77]
Papadimitriou, C., *The Euclidean Travelling Salesman Problem is NP-complete*, Theoretical Comp. Sci., 4, p237-244, 1977.

[Pa81]
Papadimitriou, C., *Worst-case and probabilistic analysis of a geometric location problem*, Siam J. of Computing, vol 10, no. 3, 1981.

[PS82]
Papadimitriou, C. and Steiglitz, K., *Combinatorial Optimization, Algorithms and Complexity*, Prentice Hall, 1982.

[PY82]
Papadimitriou, C. and Yannakakis, M., *The Complexity of facets (and some facets of complexity)* 14th Annual STOC, p255-260, May 1982.

[Sa70]
Saaty, T., *Optimization in Integers and Related Extremal Problems*, McGraw Hill, Inc., 1970.

[Se82]
Selman, A.L., *Analogues of semirecursive sets and effective reducibilities to the study of NP completeness*, Information & Control Vol. 52, Jan. 1982, pp30-51

[Sc71]
Scott, A., *Location -allocation systems : A Review* Geographical Analysis, p95-119, 1971.

[St73]
Stauduhar, R.P., *The Determination of Galois Groups*, Mathematics of Computation, vol 27, no 121, p981-996, 1973.

[SH75]
Shamos, M.I. and Hoey, D., *Closest Point Problems* 16th Annual FOCS, p151-162, October 1975.

[TFL83]
Tansel, B., Francis, R. and Lowe, T., *Location on Networks : A survey* Part I : The p-center and p-median problems, Part II : Exploiting Tree Networks Structure , Management Science, vol 29, no 4, p482-511, 1983.

[Va53]
Van der Waerden, B.L., *Modern Algebra*, vol I, Ungar, New York 1953.

[We37]
Weber, A., *Theory of the Location of Industries,* Translated by Carl J. Friedrich. The University of Chicago press, 1937.

[WH73]
Wendell, R. and Hurter, A., *Location Theory, Dominance and Convexity* Opern. Res., vol 21, p314-320, (1973)

[Ya75]
Yao, A., *On the Complexity of Comparison Problems using Linear Functions,* Proceedings 16th STOC, p85-89, Berkeley, 1975.

[Za71]
Zassenhaus, H., *On the Group of an Equation* Computers in Algebra and Number Theory, G. Birkhoff and M. Halls eds., SIAM and AMS proceedings, p69-88, 1971.