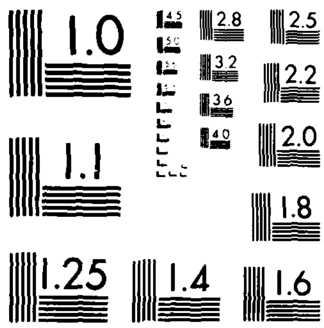
AD-A149 320 HEURISTICS BASED ON SPACEFILLING CURVES FOR COMBINATORIAL PROBLEMS IN THE..(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION RESE.. 1/1

UNCLASSIFIED J J BARTHOLDI ET AL. SEP 84 PDRC-84-08 F/G 12/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

(12)

Georgia Institute of Technology

AD-A149 320

PRODUCTION and DISTRIBUTION RESEARCH CENTER

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

PDRC Report Series 84-08
September 1984

12

HEURISTICS BASED ON SPACEFILLING CURVES FOR COMBINATORIAL PROBLEMS IN THE PLANE

by

John J. Bartholdi, III
Loren K. Platzman

PDRC 84-08

A

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This work was supported by the Office of Naval Research under Contract No. N00014-83-K-0147 and the National Science Foundation under Grant ECS-8351313.

Submitted to <u>Operations Research</u>

A

Letter on file

Abstract

A1

We introduce a family of heuristics, based on spacefilling curves, to solve general combinatorial problems in the plane, such as routing, location, and clustering. These remarkably simple and fast heuristics are nonetheless fairly accurate and so seem well-suited to operational problems where time or computing resources are limited. They ignore many details of the problem, yet generate solutions that are good simultaneously with respect to a variety of measures. (This may be useful when the problem specification is incomplete or cannot be agreed upon.) Furthermore they are extremely simple to code, and in some cases may even be implemented without a computer.

AMS 1980 Subject Classification: Primary: 90C42. Secondary: 90B35.

OR/MS Index, 1980 Subject Classification:

Primary: 632, programming/integer/algorithms/heuristic.

Secondary:491, networks/graphs/travelling salesman.

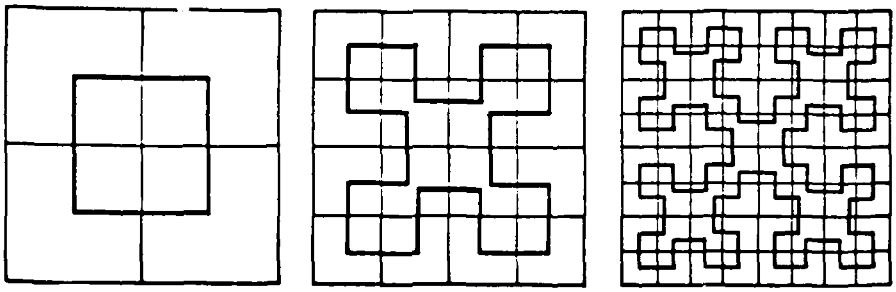
Key words: spacefilling curves, geometrical combinatorial optimization, heuristic, routing, clustering, location.

0. Introduction

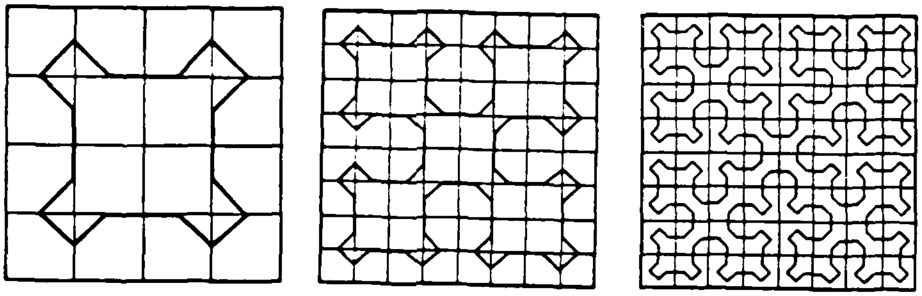
In Bartholdi and Platzman (1982), we introduced spacefilling curves as the basis for an extremely fast heuristic to solve the travelling salesman problem in the plane. The usefulness of this heuristic is demonstrated in Bartholdi *et al*. (1983), wherein is described the implementation of a commercial routing system so simple that it requires no computer. (It consists of two Rolodex™ card files, and is being used for the daily routing of four vehicles to 200-300 locations.) Here we provide a detailed discussion of the principles underlying spacefilling methods, and we extend our earlier work in two ways. First we suggest a more general application of spacefilling curves to the solution of a variety of combinatorial problems in the plane. Then we discuss the relative merits of different spacefilling curves, and show how to design a "best" one.

Consider a combinatorial problem in which are given n points in the unit square together with a specified metric. (The coordinates of each point are assumed to be given to fixed, prespecified accuracy.) The problem asks for some combinatorial structure of maximal or minimal cost. Examples include the travelling salesman problem, the matching problem, the K-median problem, etc. Many such problems are inherently difficult; for example, the Euclidean travelling salesman problem is NP-complete (Garey and Johnson (1980) and Papadimitriou (1977)). Other planar problems such as matching may have formally efficient solution techniques that are nevertheless unsuited for some real-time operational environments (Avis (1983) and Bartholdi and Platzman (1983)). We suggest a family of fast heuristics, based on spacefilling curves, for these problems.

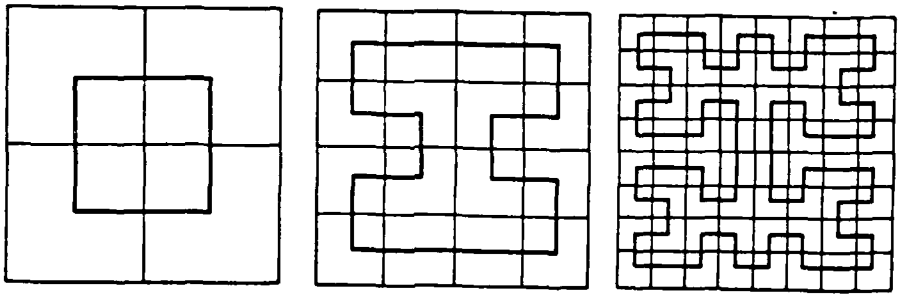
A spacefilling curve is a continuous mapping of the unit interval *onto* the n-dimensional unit hypercube. (See Figures 1 and 2 for examples



(A)

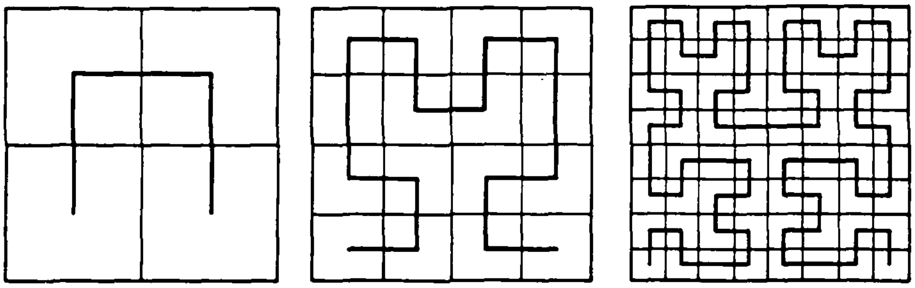


(B)

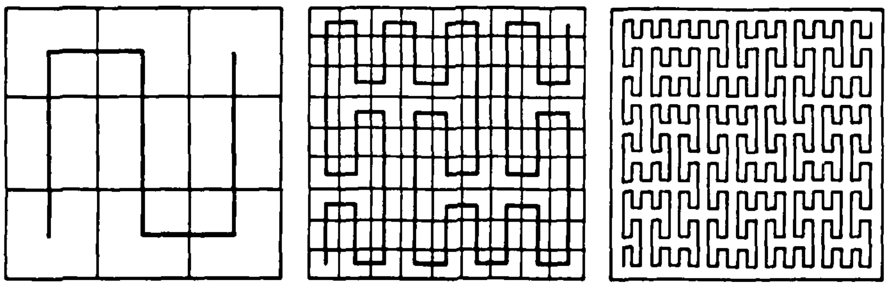


(C)

Figure 1: Geometric construction of some spacefilling curves in the unit square. Each curve is the limit of a sequence of recursive constructions.



(A)



(B)

<u>Figure 2</u>: These spacefilling curves are paths, not circuits, through the unit square.

in two dimensions.) Such curves were first introduced by the mathematicians Peano (1890), Hilbert (1891), and Sierpinski (1912) as "topological monsters," since it seems contrary to intuition that a lower-dimensional space can be mapped continuously onto a higher-dimensional space. Since then, spacefilling curves have continued to interest mathematicians and computer scientists for their elegant recursive structure, and for the surprise and visual delight they afford. They are part of the family of fractal curves discussed in detail by Mandelbrot (1983), who has done much to stimulate interest in them.

Spacefilling curves may be defined in any dimension. However, for ease of exposition, we shall discuss them as continuous mappings from the unit interval onto the unit square. All of the ideas we present are easily generalized to n dimensions.

A property of spacefilling curves that is crucial to our purpose is that they tend to preserve "nearness" among points. If two points are close on the curve, then they are close in the plane. Conversely, if two points are close in the plane, then they are likely - note the qualifier! - to be close on the curve. This tendency to preserve nearness is due to the highly convoluted shape of a spacefilling curve; it tends to visit all the points in one region of the plane before travelling to a new region.

These properties of spacefilling curves suggest the following idea: transform the problem in the unit square, via a spacefilling curve, to an easier problem on the unit interval; then solve the easier problem and take that solution as a heuristic solution to the original problem. Combinatorial problems are generally easier when posed on the unit interval than when posed in the unit square. The spacefilling curve enables us to model the unit square in a simple way while tending to preserve nearness among points. Since the common combinatorial problems have objective functions that depend on nearness, the problem on the unit interval will tend to be faithful to the original problem in the most important way. Hence the following.

GENERIC HEURISTIC

Step 1: Transform the problem in the unit square, via a spacefilling curve, to a problem on the unit interval.

Step 2: Solve the (easier) problem on the unit interval.

This is actually a whole family of heuristics, depending on the combinatorial optimization problem, the particular spacefilling curve, and the implementation of Step 2.

For this heuristic to be useful, the transformation via a spacefilling curve must be easily computable. In fact the transformation is quick and straightforward for each of the spacefilling curves we studied. If the coordinates of each point are given to k-digit accuracy, only O(kn) elementary steps (+,-,*,/) are needed to accomplish Step 1. (And, in fact, the multiplication and division are exclusively by a constant which depends only on the spacefilling curve and not on the problem instance. For the curves we studied, this constant is 2 or 3.) Table 1 gives a pseudo-Pascal program to compute, for any point in the unit square, a corresponding point on the unit interval determined by the spacefilling curve of Figure 1A. Figure 3 shows a point set transformed via this curve. Descriptions of how to compute other curves and their inverses may be found in Bially (1969), Butz (1971), and Patrick et al. (1968).

Let (X,Y) be a point in the unit square; POSITION(X,Y) is a corresponding point on the unit interval.

<u>Function POSITION(X,Y)</u>

```
if X = 1 and Y = 1 then RETURN(0.5)
Q = NV(MIN(INT(2*X), 1),MIN(INT(2*Y), 1))
                                        {Q identifies the quadrant
                                         containing (X,Y)}
T = POSITION(2*ABS(X - 0.5),2*ABS(Y - 0.5)
                                        {T is the position along the
                                         subcurve in quadrant Q}
if MOD(Q,2) = 1 then T = 1 - T
                                        {Visit the vertices of a
                                         quadrant clockwise}
RETURN(FRACT((Q + T)/4 + 7/8))
```

where

```
ABS(A) = A if A > 0,    -A if A < 0,
INT(A) = the largest integer not larger than A,
FRACT(A) = A - INT(A),
MIN(A,B) = A if A < B,   B if A > B,
MOD(A,B) = B*FRACT(A/B),
NV(X,Y) = the 'number' of vertex (X,Y) of the unit square, counting
          clockwise from the origin, i.e., Nv(0,0) = 0, Nv(0,1) = 1,
          Nv(1,1) = 2, Nv(1,0) = 3.
```

<u>Table 1</u>: An algorithm to compute a position on the unit interval that corresponds (under the curve of Figure 1A) to a given point on the unit square.

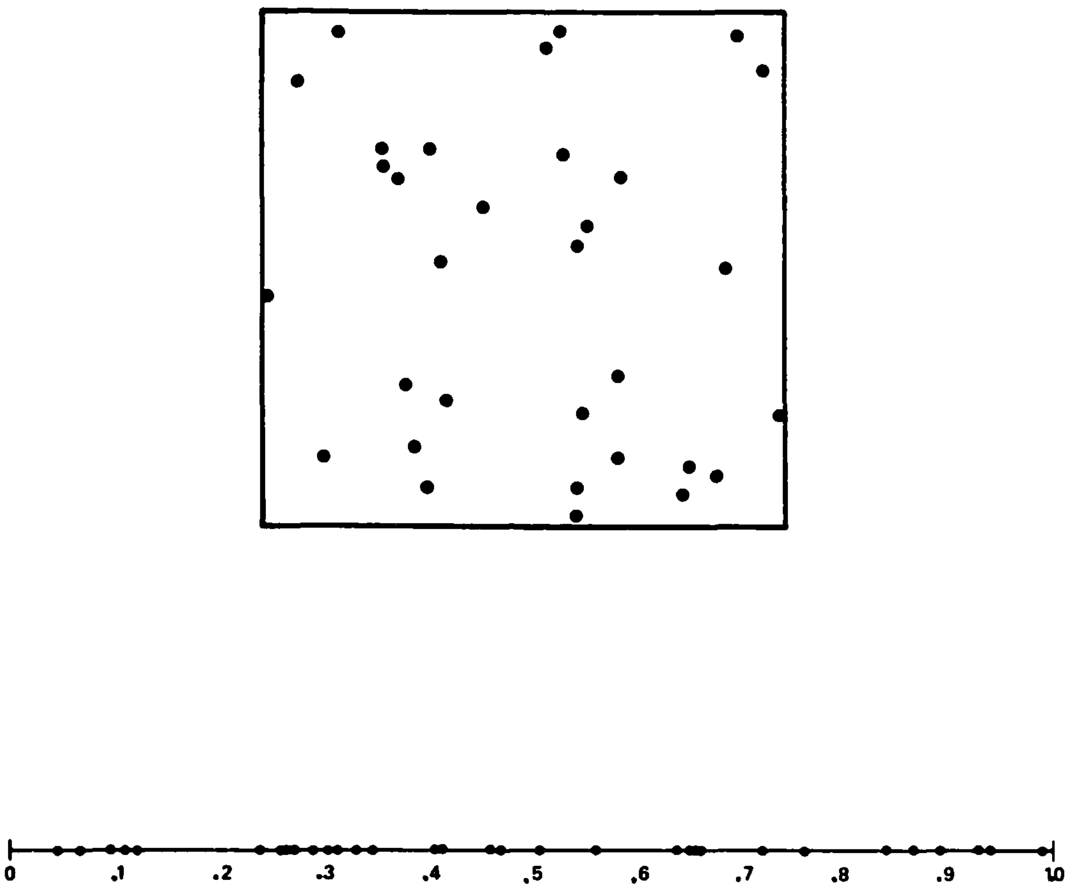


Figure 3: A point set in the plane transformed to a point set on the line via the spacefilling curve of Figure 1A. Clusters of points tend to be preserved since the curve is continuous.

In Sections 1 and 2 we illustrate implementations of the generic heuristic for several problems. Section 3 gives a very general performance analysis. In Section 4 we consider the question of finding the "best" spacefilling curve, and provide a method to compute customized curves for specific applications. Concluding remarks are given in Section 5.

1. Routing problems and spacefilling curves

Implicit in most routing problems is the planar travelling salesman problem: given n points, which we take to be in the unit [illegible] find the shortest circuit connecting all the points. Bartholdi a[illegible] Platzman (1982) suggested a heuristic for the planar travelling sale[illegible] problem, of which this work is a generalization. That heuristic was based on a specific curve (Figure 1A). A more general statement of that algorithm is

ALGORITHM TSP

Step 1: For each point calculate, via a spacefilling curve, a corresponding position on the unit interval.

Step 2: Sort the points according to their corresponding positions on the unit interval.

This heuristic simply visits the points in the same order as does the spacefilling curve, and so may be implemented by straightforward sorting. The spacefilling curve may be thought of as the route of an <u>obsessive</u> salesman who visits <u>every</u> point in the unit square. The heuristic route visits only the required points, but in the same sequence as they appear along the spacefilling curve. (See Figure 4.)

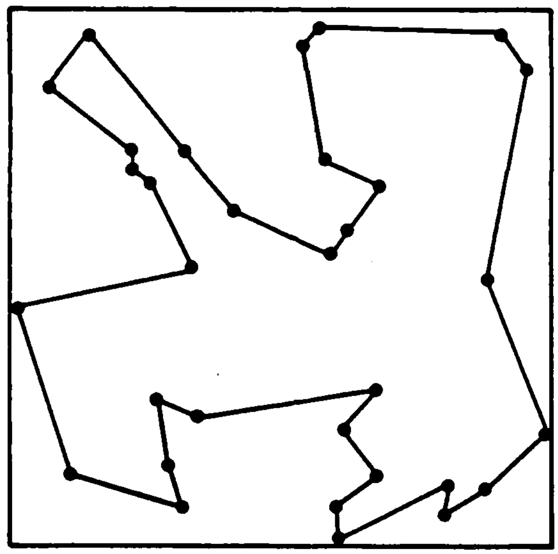


Figure 4: A heuristic travelling salesman's tour based on the spacefilling curve of Figure 1A.

The performance of this heuristic for the specific curve of Figure 1A is analyzed in detail in Platzman and Bartholdi (1983). We summarize the attractive features of that algorithm since they are typical of the generic algorithm. First the heuristic is abstemious in its data requirements: only the $O(n)$ coordinates of the points to be visited are necessary. In fact the $O(n^2)$ distances between points are ignored! By ignoring so much of the problem data, such as the metric and the distribution from which the points are drawn, the user is freed from the expense of collecting that information. The algorithm is extremely fast: it consists essentially of sorting, and so can be implemented to run in $O(n \log n)$ steps (worst-case), and $O(n)$ steps (expected case). The algorithm is agile in that it can quickly update solutions in response to small changes in the problem: points may be inserted into or deleted from the heuristic tour within $O(\log n)$ steps. (By constrast, solutions generated by other methods may need to be entirely re-solved when the problem changes.) Finally, the heuristic is trivial to code, requiring only about 20 lines of BASIC.

Of course an algorithm that ignores so much of the problem cannot hope to be exceptionally accurate and, indeed, this heuristic is only fairly accurate. For uniformly distributed points it produces tours that are 25% beyond optimum when measured by the Euclidean metric (almost surely, as n gets large). The worst-case ratio (heuristic tour length/optimum tour length) is no more than $O(\log n)$, and we suspect that this can be improved to $O(1)$.

In general the algorithm seems well-suited to operational problems in which time and computing resources are limited. For example, a version of this heuristic might be used to "route" naval gunfire among targets. The ability to quickly update solutions could be critical in such an application.

Another possible application would be to assign zip codes according to a (quantized) spacefilling curve. Then not only would locations with similar zip codes be close, but also close locations would tend to have similar zip codes. This could be useful in, say, parcel delivery, for a good route could be constructed by simply visiting the locations from smallest to largest zip code.

Another use is suggested in Bartholdi and Platzman (1983). A heuristic for matching is to simply choose every other edge of the heuristic TSP tour. This gives good solutions quickly, and so may be useful in controlling the movement of a mechanical plotter pen in real-time. (The fastest known optimum-finding algorithm can require $O(n^3)$ steps, which may be too time-consuming.)

2. Location/Clustering problems and spacefilling curves

The planar K-median problem is to choose, from among n given points, K of those points to be "medians", so as to minimize the sum of distances from each point to its closest median. This problem arises, for example, in choosing locations for distribution or service centers in a geographical region. It has been studied by Fisher and Hochbaum (1980) and by Papadimitriou (1981), who established the NP-completeness of the Euclidean problem.

We suggest two versions of the generic heuristic. Both are stated in their simplest form; they can be made more accurate, at the cost of extra computation, by including more powerful subroutines.

The first is a fixed partition scheme.

ALGORITHM K-MEDIAN 1

Step 1: For each point calculate, via a spacefilling curve, a corresponding position on the unit interval.

Step 2: Solve the K-median problem on the unit interval;

> Divide the interval into K identical subintervals;
> Choose the medians to be those points closest to the centers of the subintervals.

The second version is a variable partition scheme. It consists of replacing Step 2 with the following.

Step 2': Solve the K-median problem on the unit interval;

> Choose the medians to be the K-dian points (i.e. every n/Kth point)

Both of these heuristics require only $O(n)$ data. A straightforward implementation of the first heuristic requires $O(n)$ steps in the worst-case. The second heuristic consists essentially of sorting, and so may be implemented to require $O(n \log K)$ computational steps (worst-case) and $O(n)$ steps (expected case). Again solutions produced by either heuristic may be updated quickly.

Figure 5 shows the solution produced by algorithm K-median 1 on a set of random points. Typically, it is fairly good at identifying clusters of points; it is less good at choosing the best median for a cluster.

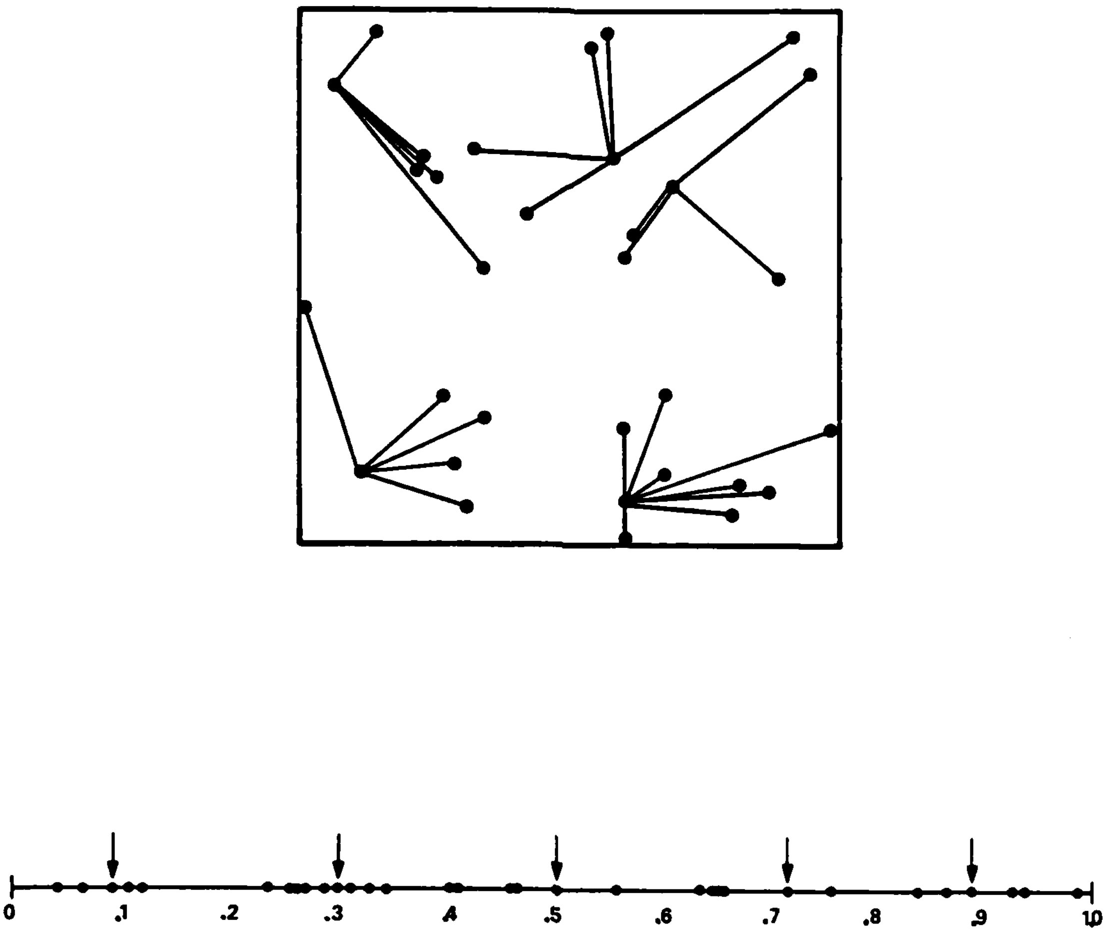


Figure 5: Heuristic solution of a 5-median problem on the unit interval and the corresponding heuristic solution in the unit square.

The K-center problem is to choose K of the n points so that the maximum distance between any point and its closest center is minimized. This problem is not known to be NP-complete when restricted to the plane, but has been conjectured to be so by Papadimitriou (1981). The K-median algorithms given above may be used, unchanged, for this problem too.

Alternative versions of the K-median and the K-center problems have been studied by Megiddo and Supowit (1984). They show that when the medians/centers are allowed to be arbitrary points, not necessarily among the original point set, then the problems are NP-hard under either the Euclidean or rectilinear metric. The analogous problems on the unit interval are easier however: the K-median problem on the interval is solvable in $O(n^2K)$ time (Megiddo <u>et al</u>. (1983)) and the K-center problem is solvable in $O(n \log n)$ time (Megiddo <u>et al</u>. (1981)). The exact solution procedures for the interval may be used as the implementation of step 2 in the generic heuristic, with presumed consequent improvement in accuracy.

Spacefilling curve techniques might be useful in very general problems that require the identification of clusters of points. Duran and Odell (1974) give a survey of cluster analysis, in which is discussed a variety of measures of "nearness" for points in space of arbitrary dimension. These are used to formalize notions of "similarity" of data points. Since the K-median/center heuristics ignore the actual metric, but nevertheless captures "nearness", they may be expected to produce solutions that are reasonably good with respect to many of these measures simultaneously. This could be of special use to statisticians, who may not agree on the appropriate measure of "similarity" ("nearness").

3. Performance Analysis

In this section, we give a very general performance analysis of the generic spacefilling heuristic which suggests that it may be effective in solving a wide variety of combinatorial problems in the plane.

Consider the problem of selecting, from a given complete graph, a subgraph of given structure and minimal total weight. An instance of the problem is specified by a set of points P and a metric (distance measure) D. The nodes of the graph correspond to the points in P and the edges are labeled with distances determined by D. We denote by $V^*(P,D)$ the value of the optimal subgraph, that is, the sum of its edge weights. To each problem type (TSP, matching, spanning tree, etc.), there corresponds a particular function V^*. The heuristic selects a subgraph of proper structure whose total weight is small but not necesarily minimal. Its value is denoted by $V(P,D)$.

A norm $\|\cdot\|$ is a measure of a vector's magnitude. It satisfying $\| ap \| = a \|p\|$ for all scalars a and vectors p. A metric may be induced by a norm via $D(p,p') = \|p-p'\|$; it is then unaffected by shifts, $D(p+q,p'+q) = D(p,p')$, and it responds linearly to changes of scale, $D(ap,ap') = a\, D(p,p')$. (However, it may be affected by rotation.) Euclidean, rectilinear, and Chebychev distances are all examples of normed metrics.

A spacefilling curve ψ, mapping the unit interval $C = [0,1]$ onto the unit square $S = [0,1]^2$, is said to be recursively defined if, for some m, its path over each subsquare of side $1/m$ is similar to its path over S (but scaled by a factor of m and possibly rotated). At the k-th level of recursion, subsquares have sides of length m^{-k}. Although the curve may enter and leave a subsquare more than once, its path during any particular visit to a subsquare must span a region whose area is at least a fraction α of the area of the subsquare. (The curve in Figure 1A visits subsquares at most twice, and covers at least 1/2 the area of a subsquare on each visit.)

The notion that a spacefilling curve preserves nearness may now be formalized.

<u>LEMMA 1</u>. If ψ is a recursively-defined spacefilling curve and D is a norm-induced metric, then there is a constant c such that

$$D(\psi(\theta),\psi(\theta')) \leq c\sqrt{|\theta-\theta'|}.$$

<u>Proof</u>. Let k be the largest integer such that $\alpha\, m^{-2k} \geq |\theta-\theta'|$. Then $|\theta-\theta'| \geq \alpha\, m^{-2(k+1)}$, or equivalently,

$$m^{-k} \leq m\, \alpha^{-1/2}\sqrt{|\theta-\theta'|}. \quad (1)$$

If S is partitioned into subsquares of side m^{-k}, then C may be divided into subintervals of length at least $\alpha\, m^{-2k}$ such that the image under ψ of any subinterval lies entirely within a subsquare. Since $|\theta-\theta'|$ is bounded above by the shortest subinterval length, θ and θ' must lie within adjacent subsquares. So $D(\psi(\theta),\psi(\theta')) \leq 2m^{-k}W$, where W is the largest distance between two points in S. With (1), this completes the proof.

[]

For the particular case of ψ as in Figure 1A and D = Euclidean distance, Platzman and Bartholdi (1983) showed that c = 2.

It is easy to show that the heuristic TSP tour is $O(\sqrt{n})$. Let $\Delta_1,\ldots,\Delta_N$ be the distances (in C) between consecutive θ's in the sorted list. Since C has length one,

$$\Sigma \Delta_i = 1$$

and by Proposition 1,

$$\text{heuristic tour length} \leq \Sigma\, c\sqrt{\Delta_i}.$$

This upper bound achieves its maximum at $\Delta_i \equiv 1/N$, so

$$\text{heuristic tour length} \leq c\sqrt{N}.$$

We now show that, for a much more general class of problems, the spacefilling heuristic produces $O(\sqrt{n})$ solutions.

A problem is called subadditive if, for any partition Σ of S (the cells $\sigma\epsilon\Sigma$ need not be identical nor even have equal areas) there is a γ such that

$$V^*(P,D) \leq \sum_{\sigma\epsilon\Sigma} [V^*(P \cap \sigma, D) + \gamma \max \{D(p,p'): p,p' \epsilon \sigma\}].$$

This says that the problem may be partitioned in any way, solved locally, and patched together with a penalty that depends only on the partition Σ. Examples of subadditive problems include TSP, matching, minimum spanning tree, and k-median for $k = \alpha n$.

If the problem to be solved on the line (in Step 2 of the spacefilling heuristic) is not a travelling salesman problem then we must take explicit note of the metric (on C) to be used. The following assumes that this metric is $\sqrt{\Theta - \Theta'}$ and that the problem on the line is solved exactly.

PROPOSITION 1. The spacefilling heuristic provides $O(\sqrt{n})$ solutions to subadditive problems with norm-induced metrics.

Proof. Let $\Delta(p,p') = \sqrt{|\theta-\theta'|}$ where $p = \psi(\theta)$ and $p' = \psi(\theta')$. Δ is a metric (but not a normed metric). By Lemma 1, $D(p,p') < c\Delta(p,p')$, so $V^*(P,D) < V(P,D) < cV^*(P,\Delta)$. We show that $V^*(P,\Delta) = O(\sqrt{n})$ so that the same is true of $V(P,D)$. Partition C into N subintervals, each containing only one Θ value. Let Δ_i be the subinterval lengths. Project the subintervals onto S via ψ to obtain a partition of S. By subadditivity,

$$V^*(P,\Delta) < \Sigma\ \gamma c\sqrt{\Delta_i}$$

since V^* (a single point, $\cdot$) = 0 and $\max \{\Delta(p,p') \colon p,p' \in \psi(I)\} < c\sqrt{\Delta}$ when I is a subinterval (in C) of length Δ. By concavity of $\sqrt{\cdot}$,

$$V^*(P,\Delta) < \gamma c\sqrt{N}.$$

[]

Stochastic analysis shows that the generic spacefilling heuristic produces solutions that are close to optimal in a certain sense. Suppose that $p_1, p_2, \ldots$ is a sequence of independent uniformly distributed points on S. Let $P_N = \{p_1, \ldots, p_N\}$. For a general class of subadditive problems, Steele (1981) proves that there is a β^* such that

$$\frac{V^*(P_N, \text{Euclidean metric})}{\sqrt{N}} \to \beta^* \quad \text{a.s.}$$

Steele's proof is readily extended to general normed metrics. In any case, if

$$\frac{V^*(P_N,D)}{\sqrt{N}} \to \beta^* \quad \text{a.s.}$$

then, by Proposition 1, there is an R such that

$$\limsup_{N \to \infty} \frac{V(P_N,D)}{V^*(P_N,D)} \leq R \quad \text{a.s.}$$

Thus the generic spacefilling heuristic produces solutions which are likely to be within a given constant factor of optimal when N is large. (For the Euclidean TSP, we have estimated this factor to be 1.25.)

4. What is the best spacefilling curve?

The generic heuristic may be implemented with any spacefilling curve. However, the quality of solutions may be different for different curves. For example, we tested the travelling salesman heuristic on random point sets for each of the spacefilling curves of Figures 1 and 2. For each of these curves it can be proven that, for random point sets drawn from a sufficiently smooth distribution over the unit square, the variance of (heuristic tour length/$\sqrt{n}$ vanishes almost surely as n gets large. (See Platzman and Bartholdi (1983) for a proof for the curve of Figure 1A.) Note that, unlike the optimal tour length, expectation of this ratio need not converge. However our tests for n up to 1,000 suggest that their first several decimal digits are nearly equal. In any case, by results of the previous section, they are bounded by a constant. Consequently, for the purpose of comparing the performances of various curves, we shall speak of the ratios (heuristic tour length /$\sqrt{n}$) for each curve as if they converged rapidly to some β. We estimated β for the curves of Figure 1 to be 0.96, 0.98, and 1.12 respectively. For the curves of Figure 2, we estimated β to be $1.10 + 1/\sqrt{n}$, and $1.12 + 1/\sqrt{n}$ respectively, where the latter terms represent the distance necessary to close the path to form a circuit. (This additional distance makes these curves unsuitable for the travelling salesman problem when n is small.) This may be compared to the result of Beardwood, Halton, Hammersley (1959), who prove that the ratio (optimum tour length/$\sqrt{n}$) approaches β^* almost surely as n gets large, where β^* has been estimated to be 0.765.

A curve with small β is to be preferred, since it tends to produce shorter tours. Accordingly, for the travelling salesman problem we may consider the curve of Figure 1A best. It is more "homogeneous", and therefore performs well for homogeneously random point sets. The curve of Figure 1B is the same curve in the limit, but the finite version performs slightly less well than 1A. The curve of Figure 1C has fewer axes of symmetry and so performs still less well.

The curves of Figure 2 are paths rather than circuits, so that, for small problems they tend to link the first and last points inefficiently, and so produce less accurate solutions. However, for sufficiently large n, these differences tend to disappear. The asymptotic performances of all the curves are similar because locally they tend to resemble each other.

We also tested the algorithm using spacefilling curves which are not, strictly speaking, curves. They are not curves because they are not continuous. (See Figure 6.) However, because of their recursive structure, they tend to enjoy, although to a lesser extent, the "nearness-preserving" of spacefilling curves. For the two curves of Figure 6 we estimated β to be $1.12 + \sqrt{2/n}$ and $1.37 + \sqrt{2/n}$, respectively.

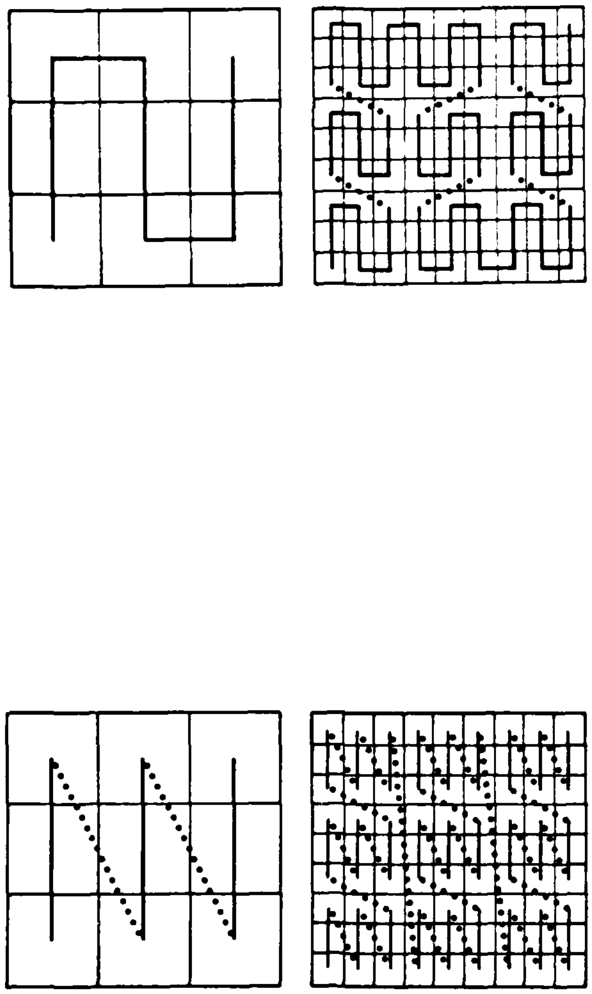


Figure 6: These recursive structures are not, strictly speaking, curves, since they are not continuous.

Their relatively poor performance confirms the intuition that continuity is essential to effectively model nearness.

The heuristic can even be implemented with "curves" that are discontinuous everywhere, such as the one illustrated in Figure 7, for which β was estimated to be $1.99 + (1/3)\sqrt{2/n}$. It is surprising that any such recursively-defined "curve" enables the algorithm to perform well (in the sense that the average heuristic solution grows at the same rate as the optimum as n gets large).

What is the best spacefilling curve? For the aforementioned reasons of symmetry and homogeneity, we think the curve of Figure 1A is best for combinatorial problems on *uniformly distributed* points. Even for smooth distributions the curve tends to perform fairly well since, within small regions, the distribution tends to appear uniform. However, in general, other curves may give better performance. To say more than this we must reconsider the role played by the spacefilling curve.

The essential contribution of the spacefilling curve is simply to provide a linear ordering of all the points in the plane. The generic algorithm could be implemented with <u>any</u> linear ordering that could be computed or looked up quickly. But, to be most effective, the linear ordering should be tailored to the distribution from which the problem instances are drawn.

In general we might be willing to spend considerable effort to design an effective spacefilling curve for a particular problem, since this is a design problem and so needs to be solved only once. Afterwards, our operational problems will be solved quickly and accurately by the generic heuristic, and this good performance will amortize the design costs.

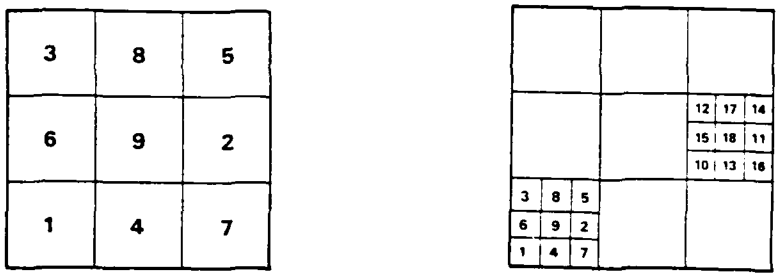
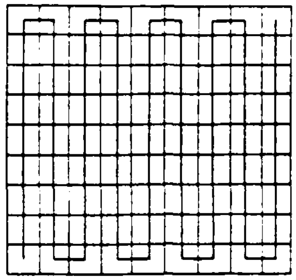


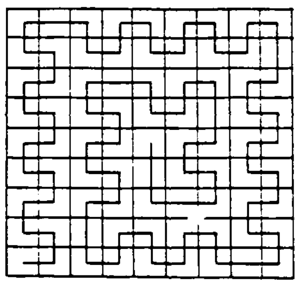
Figure 7: Recursive construction of a "curve" that is discontinuous everywhere. For clarity, the structure of the "curve" is indicated by numbers instead of lines.

Let us consider a finitized version of the design problem. Suppose that a finite set of points in the unit square is distinguished. Over subsets of the distinguished points is defined some distribution from which instances of a combinatorial problem are drawn. The generic algorithm may be implemented via any linear ordering of the distinguished points, given, for example, by a simple list. To emphasize the application to finite point sets we refer to a linear ordering used to implement step 1 of the generic heuristic as a "presequence". The effectiveness of any particular presequence is measured by the expected value of the objective function, over all instances of the problem, when the generic algorithm is implemented with that presequence. We want a presequence for which the generic algorithm produces the best solutions (on the average).

An idea similar to presequencing has been used by Iri, Murota, and Matsui (1983) in a heuristic for planar matching. However, the presequences they studied (Figure 8) are not spacefilling curves, and in particular are not circuits, and so do not model the plane as well as they might. Indeed, the spacefilling curve of Figure 1A, when used in their algorithm, performed better than either of the presequences of Figure 8. An additional disadvantage of the presequences of Figure 8 is that they are not recursively constructed. Thus they are likely to perform poorly (relative to the optimum) for non-uniform distributions of points. Finally, since these presequences depend on the number of points in the problem instance, heuristic solutions are not so easy to modify as those based on spacefilling curves (whose structure is independent of the problem instance).



(A)



(B)

Figure 8: Two presequences that have been used in related work. They do not work as well as they might since they are not circuits, and they are not recursively constructed.

Unfortunately, it can be hard to determine the best presequence. In the worst case, if all of the points occur in a problem instance with probability 1, then finding the best presequence is equivalent to solving an instance of the combinatorial problem. Nevertheless, it is possible to construct presequences that are effective, if not optimal. Given a class of problem instances and a specified combinatorial problem, one can design a good presequence by an interchange heuristic. (For a general discussion of interchange methods, see Papadimitriou and Steiglitz (1982).)

ALGORITHM K-INTERCHANGE

Step 0: Begin with an initial presequence, and designate it the current best. Set $M = 0$.

Step 1: Interchange a random selection of k precedences of the presequence to form a new presequence.

Step 2: Estimate its performance by solving a sufficiently large random sample of problems with the generic heuristic. If the new presequence gives improved performance, then choose it as the current best and set $M = 0$.

Step 3: Set $M = M + 1$. If $M < M_{max}$ then return to Step 1.

We tested an implementation of this heuristic to design effective spacefilling curves for several different travelling salesman problems. We chose problems defined as follows: suppose that a finite grid of points in the plane is distinguished, and that to each point j there corresponds a probability p(j). Each point j occurs in a problem instance independently with probability p(j). (Note that this independence assumption is not necessary to apply the method; this was simply a convenient way of generating sample problems.) We implemented the design heuristic as a 3-interchange.

In analyzing the presequences produced, we noticed an interesting phenomenon. In regions where many points had a p(j) near 1, the presequence tended to have many straight segments. This makes sense since, because of the large p(j), the design problem became almost a travelling salesman problem. (If all p(j) = 1, the design problem is exactly a travelling salesman problem on the grid of points.) On the other hand, in any region with many points with small p(j), the presequence tended to be highly convoluted because it was "hedging". (See Figure 8.) The presequence was uncertain which, if any, of the points in that region would be next in a random problem. The smaller the p(j)'s within a region, the more the presequence hedged, and the more convoluted it became.

The phenomenon of hedging relates to the structure of spacefilling curves in an interesting way. Let all of the distinguished points have the same probability p(j) = p. Then as the number of points gets large and p gets small (with np constant), the optimal presequence becomes extraordinarily convoluted as it hedges among many points with tiny probabilities. In fact, the hedging of the presequence becomes the non-differentiability of a spacefilling curve.

A possible use for this might be in the area of warehouse operations. It is common for retrieval to be sequenced by simply visiting storage bins according to their bin number. If the bins were numbered according to the best presequence, the performance of this retrieval strategy could be enhanced. Figure 9 shows a wall of bins along a warehouse aisle with an idealized bin-numbering sequence suggested by our computer simulations. Notice that near the front of the warehouse aisle, where are stored the most frequently requested items, the presequence tends locally to be a travelling salesman tour. Farther along the aisle, where are stored the progressively less-often-requested items, the presequence hedges increasingly. Finally, at the end of the aisle, the presequence clearly resembles a (quantized) spacefilling curve. (Note: Due to edge effects and alternative optima, our simulation did not produce the exact curve as Figure 9. However, the increased hedging among low probability locations was clearly recognizeable, and Figure 9 idealizes that.)

5. Concluding Remarks

We have observed that, for many combinatorial problems in the plane, the generic heuristic tends to produce solutions that grow at the same rate as the optimum solution. If we consider this property "decent", then we can say that the generic heuristic gives decent solutions for many different problems. Moreover, a specific implementation of the generic heuristic may give solutions which are decent simultaneously for many different objective functions and for many different metrics. This is because the heuristic tends to produce solutions based on "nearness" and not on a specific metric. In fact, because of the simplicity of the problem on the line, the implementation of step 2 is frequently independent of the metric in the plane and sometimes even independent of the precise form of objective function. This robustness of solution may be especially useful for ill-defined problems. Thus, one might say, if decisions must be made quickly, but one is not sure of the objective or the data, then use a spacefilling curve-based heuristic.

An interesting multiple application of spacefilling curves concerns a hierarchical routing system, that uses the generic heuristic to first recognize clusters of locations, and then to route a vehicle through each

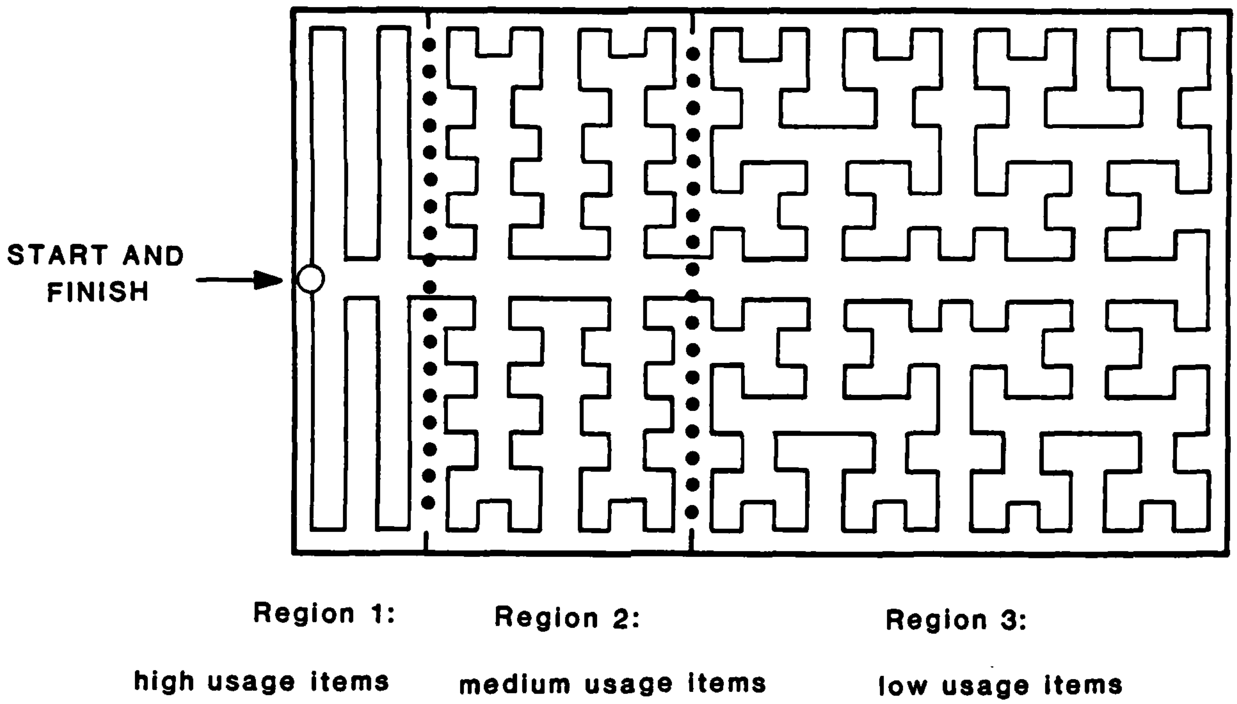


Figure 9: A wall of bins along a warehouse aisle numbered in an effective sequence. The curve becomes increasing convoluted as it "hedges" among the low probability items.

cluster. We have used this idea to quickly analyze and improve the routing system of a commercial package courier service in Atlanta, Georgia. The true metric of the problem, travel times, varied with the time of day and the day of the week, and so was too complex to be useful (or even knowable). The generic heuristic ignored this metric and yet tended to do well (we think!). At least it was a clear improvement over what had been done previously.

Some researchers consider the ultimate test of a method to be its ability to catch a lion (Stewart and Jaworski (1981)). For example, to catch a lion by binary search, start with all of Africa and bisect, retaining the half that contains a lion, until the remaining area is the size of a cage; it will contain a lion. To satisfy these readers we offer two ways to catch a lion by spacefilling methods. First, grab a spear (or a net) and run through Africa along the path of a spacefilling curve; you will catch at least a lion. Alternatively, map Africa onto the interval, and stand at $\theta = 0$ facing $\theta = 1$; you will see the lion directly ahead of you, no more than one (theta) unit away.

Acknowledgements

The authors thank James Orlin, Robin Roundy, Craig Tovey, and Zhang Jixian for helpful comments, and Zhang Jixian for programming and testing many of the heuristics.

REFERENCES

D. Avis (1983). "A Survey of Heuristics for the Weighted Matching Problem," Networks 4(13), pp. 475-493.

J. J. Bartholdi and L. K. Platzman (1982). "An O(n log n) Planar Travelling Salesman Heuristic Based on Spacefilling Curves," Operations Research Letters 1(4), pp. 121-125.

J. J. Bartholdi and L. K. Platzman (1983). "A Fast Heuristic Based on Spacefilling Curves for Minimum-Weight Matching in the Plane," Information Processing Letters 17, pp. 177-180.

J. J. Bartholdi, L. K. Platzman, R. L. Collins, and W. H. Warden (1983). "A Minimal Technology Routing System for Meals-on-Wheels," Interfaces 13(3), pp. 1-8.

J. Beardwood, J. H. Halton, and J. Hammersley (1959). "The Shortest Path Through Many Points," Proc. Cambridge Philosophical Society 55, pp. 299-327.

T. Bially (1969). "Spacefilling Curves: Their Generation and Their Application to Bandwidth Reduction," IEEE Trans. Inform. Theory II-15, pp. 658-664.

A. R. Butz (1969). "Convergence with Hilbert's Spacefilng Curve," J. Comput. Sys. Sci. 3, pp. 128-146.

A. R. Butz (1971). "Alternative Algorithm for Hilbert's Spacefilling Curve," IEEE Trans. Comput., April, pp. 424-426.

B. S. Duran and P. L. Odell (1974). Cluster Analysis: A Survey, Lecture Notes in Economics and Mathematical Systems, Vol. 100, Springer-Verlag.

M. Fisher and D. Hochbaum (1980). "Probabilistic Analysis of the Planar K-Median Problem," Mathematics of Operations Research 5(1), pp. 27-34.

D. Hilbert (1891). "Uber die Stetige Abbildung einer Lunie auf ein Flachenstuck,' Math. Ann. 38.

M. Iri, K. Murota, and S. Matsui (1983). "Heuristics for Planar Minimum Weight Perfect Matchings," Networks 13, pp. 67-92.

B. B. Mandelbrot (1983). The Fractal Geometry of Nature, W. H. Freeman and Co.

N. Megiddo and K. J. Supowit (1984). "On the Complexity of Some Common Geometric Location Problems," SIAM J. Comput. 13(1), pp. 182-196.

N. Megiddo, A. Tamir, E. Zemel, and R. Chandrasekaran (1981). "An $O(n \log^2 n)$ Algorithm for the kth Longest Path in a Tree with Applications to Location Problems," SIAM J. Comput. 10(2), pp. 328-337.

N. Megiddo, E. Zemel, and S. L. Hakimi (1983). "The Maximum Coverage Location Problem," SIAM J. Alg. Disc. Meth. 4(2), pp. 253-261.

C. H. Papadimitriou (1977). "The Euclidean Travelling Salesman Problem Is NP-Complete," Theoretical Computer Science 4, pp. 237-244.

C. H. Papadimitriou (1981). "Worst-Case and Probabilistic Analysis of a Geometric Location Problem," SIAM Journal of Computing 10(3), pp. 542-557.

C. H. Papadimitriou and K. Steiglitz (1982). Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall.

G. Peano (1890). "Sur une Courbe qui Remplit Toute en Aire Plaine," Math. Ann. 36.

E. A. Patrick, D. R. Anderson, and F. K. Bechtel (1968). "Mapping Multidimensional Space to One Dimension for Computer Output Display," IEEE Trans. Comput. C-17(10), pp. 949-953.

L. K. Platzman and J. J. Bartholdi (1983). "Spacefilling Curves and Routing Problems in the Plane," PDRC technical report 83-02, Georgia Institute of Technology, submitted to for publication.

J. M. Steele (1981). "Subadditive Euclidean Functionals and Non-linear Growth in Geometric Probability," Annals of Probability 9(3), pp. 365-376.

W. Sierpinski (1912). "Sur une Nouvelle Courbe qui Remplit Toute une Aire Plaine," Bull. Acad. Sci. Cracovie, Serie A, pp. 462-478.

I. Stewart and J. Jaworski, editors (1981). Seven Years of Manifold (1968-1980), Shiva Publishing Company, Cheshire, England.

END

FILMED

2-85

DTIC