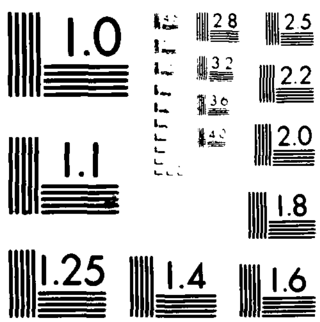
AD-A115 357 GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBU--ETC F/G 12/1
AN O(NLOGN) PLANAR TRAVELLING SALESMAN HEURISTIC BASED ON SPACE--ETC(U)
APR 82 L K PLATZMAN, J J BARTHOLDI N00014-80-K-0709
UNCLASSIFIED PDRC-82-07 NL

1 OF 1
AD A
115357

END
DATE
FILMED
07-82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD A115357

6

PDRC Report Series 82-07
April 1982

Submitted for Publication in <u>Operations Research Letters</u>

An O(NlogN) Planar Travelling Salesman Heuristic Based on Spacefilling Curves

by

L. K. Platzman
J. J. Bartholdi III

PDRC 82-07

An O(NlogN) Planar Travelling Salesman

Heuristic Based On Spacefilling Curves

L. K. Platzman
J. J. Bartholdi III

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

Abstract

2 square root of (NA).

N points in a square of area A may be sorted according to their images under a spacefilling mapping to give a tour of length at most $2\sqrt{NA}$. If the points are statistically independent under a smooth distribution, with N large, then the tour will be roughly 25% longer than optimum (and a simple enhancement reduces this to 15%). The algorithm is easily coded: a complete BASIC program is included in the appendix. Since the algorithm consists essentially of sorting, points are easily added or removed. Our method may also be used with simple dynamic programming to solve TSP path problems.

Key Words: travelling salesman problem, heuristic, routing, spacefilling curve.

The authors were supported by the New Faculty Research Development Program of the Georgia Institute of Technology and the Office of Naval Research under Contract No. N00014-80-k-0709. Reproduction in whole or in part is permitted for any purpose of the U.S. Goverment.

1. Introduction

The travelling salesman problem (TSP) is to construct a circuit of minimum total length that visits each of N given points. Even in the plane, this problem NP-complete [5]. Karp [4] has given an asymptotically efficient heuristic, but it is difficult to code and its effort has a large constant factor. Bentley and Saxe [2] gave an efficient implementation of the nearest neighbor heuristic, but it requires a special data structure. We give a faster, simpler heuristic that performs comparably.

2. Overview of the Method

Let

$$C = \{\theta \mid 0 \leq \theta < 1\} \tag{1}$$

denote the unit circle, so that $\theta \in C$ represents a point on the circle θ revolutions removed, clockwise, from a fixed reference point. Also let

$$S = \{(x,y) \mid 0 \leq x \leq 1,\ 0 \leq y \leq 1\} \tag{2}$$

denote the unit square, and suppose that we are given a continuous mapping ψ from C onto S. Such mappings were first constructed by Peano and Hilbert in the 1890's and are known as "spacefilling curves." (See, e.g., Hobson [3, pp. 451-458].

Suppose, moreover, that $\lim_{\theta \to 1} \psi(\theta) = \psi(0)$. Then, as θ ranges from 0 to 1, $\psi(\theta)$ traces out a "tour" of all the points in S. Given N points in S to be visited, a reasonable strategy is to sequence them as they appear along the spacefilling curve. In short, we sequence them according to their inverse image under ψ.

The specific form of ψ is not essential to our presentation at this time, and is deferred to Section 3. We need only two properties:

(P1) An inverse image of ψ can be easily computed. Specifically, if x and y have k-bit binary representations, then a θ satisfying $(x,y) = \psi(\theta)$ may be computed in $O(k)$ operations. Note that although many such θ may exist, we must compute only one.

(P2) There is a concave function $f(\cdot)$ on $[0,1]$, with $f(0) = 0$ and $f(\Delta) = f(1-\Delta)$, such that

$$|| \psi(\theta) - \psi(\theta') || \leq f(|\theta-\theta'|) \tag{3}$$

where $||\cdot||$ denotes the metric on S with respect to which the tour is to be minimized.

3. Routing Problems in the Circle

The basic idea in our approach is to solve routing problems in C rather than S, taking as the distance between two points in C the upper bound $f(|\theta-\theta'|)$ on the distance between their images under ψ. The problem in C has a great deal of structure, attributable to the concavity of f:

Proposition 1: (Triangle Inequality) Let $\theta_1 \leq \theta_2 \leq \theta_3$. Then $f(\theta_2-\theta_1) + f(\theta_3-\theta_2) \geq f(\theta_3-\theta_1)$.

Proposition 2: (Crossing Elimination) Let $\theta_1 \leq \theta_2 \leq \theta_3 \leq \theta_4$. Then

(i) $f(\theta_3-\theta_1) + f(\theta_4-\theta_2) \geq f(\theta_2-\theta_1) + f(\theta_4-\theta_3)$

(ii) $f(\theta_3-\theta_1) + f(\theta_4-\theta_2) \geq f(\theta_3-\theta_2) + f(\theta_4-\theta_1)$

Proofs: We appeal to the following standard inequality for concave functions:

$$f(a) + f(d) \leq f(b) + f(c) \text{ if } a \leq b,c;\ b,c \leq d;\ a + d = b + c.$$

To prove Proposition 1, let $a = 0$, $b = \theta_2 - \theta_1$, $c = \theta_3 - \theta_2$, and $d = \theta_3 - \theta_1$.

To prove Proposition 2 (i), let $a = \theta_2 - \theta_1$, $b = \theta_3 - \theta_1$, $c = 1 - (\theta_4 - \theta_2)$, $d = 1 - (\theta_4 - \theta_3)$, and recall that $f(\Delta) = f(1-\Delta)$. The proof of part (ii) is similar. □

<u>Remark:</u> Our metric is more general than $|\theta-\theta'|$, and the triangle inequality will usually be a strict inequality!

It follows from these propositions that an optimal tour on C, under metric f, is obtained by visiting the points in a sequence from smallest θ to largest θ.

Variations of the TSP may also be easily solved on C. For example, if all points are to be visited starting at θ^* and ending at θ^{**}, $\theta^* < \theta^{**}$, then we know that all points between θ^* and θ^{**} must be visited in an increasing sequence, the remaining points must be visited in decreasing sequence from θ^* to 0 and then from 1 to θ^{**}. The optimal interleafing of these two sequences is obtained in $O(N^2)$ operations, by dynamic programming. The TSP path problem with one free endpoint is similarly solvable.

Note that, once the optimal tour has been obtained for any set of points on C, it has been obtained for any subset (not necessarily consecutive) of those points.

4. The Spacefilling Curve

The approach outlined in Section 2 is valid for any spacefilling curve ψ satisfying (P1) and (P2). We now describe one such curve for which the bound (3) is particularly tight. This curve is recursively defined by dividing S into four identical subsquares, constructing a curve that fills each subsquare, and joining them at the center of S (see Figure 1). The intuition behind this definition is to join a point with its immediate neighbors before proceeding to a new region. In this respect it resembles the partitioning algorithm of Karp [4].

In order to translate this idea into a mathematical equation, we first number the vertices of S:

$$q_0 = (0,0),\ q_1 = (0,1),\ q_2 = (1,1),\ q_3 = (1,0) \tag{4}$$

and adopt the convention:

$$\psi(i/4) = q_i \qquad i = 0,1,2,3 \tag{5}$$

Accordingly, $\psi(\theta)$ will cover the subsquare of S whose outside vertex is q_0 when $|\theta - i/4| \leq 1/8$ (or $\geq 7/8$ when $i = 0$). Each subsquare covering must be rotated so that it begins and ends at the center of S. Consequently we may express ψ recursively as the solution of

$$\psi(\theta) = \tfrac{1}{2}[\psi(\mathrm{fract}(4\theta + (6-i)/4)) + q_i]\Big|_{i = \mathrm{int}(4\theta + \frac{1}{2}) \bmod 4} \tag{6}$$

where the term $\frac{1}{2}$ scales each subsquares, the argument of fract$(\cdot)$ reorients it, and $+q_i$ translates it to the desired position in S.

If we view (6) as a fixed point identity, $\psi = T\psi$, then T is seen to be a contraction operator, and so there is a unique function ψ satisfying (6). Moreover $T^m\tilde{\psi} \to \psi$ for any initial approximation $\tilde{\psi}$ of ψ. A sequence of approximations starting from (5) is shown in Figure 2.

The same argument may be carried out in reverse to show that there is a function ϕ: $S \to C$ such that $\psi(\phi(x,y)) = (x,y)$. This function satisfies a contracting recursive identity similar to (6), and is evaluated iteratively in the same manner as ψ. Since this function is given explicitly as part of the algorithm in the appendix, we avoid repeating it here.

This spacefilling curve satisfies (P1) and (P2) as we now show. Note that (6) breaks the range of θ into quarters to determine the first bit of the binary expansion of x and y. Each successive pair of bits in θ corresponds to a new bit for x and a new bit for y. Thus, it is clear that ψ satisfies (P1). For any θ, θ', we may construct a square of side at most $4\sqrt{|\theta-\theta'|}$ containing $\psi(\theta)$ and $\psi(\theta')$, so $f(\Delta) = 4\sqrt{2\Delta}$, $0 \le \Delta \le \frac{1}{2}$, in (P2), where $||\cdot||$ in (3) is understood to be the Euclidean distance. We may improve this bound to

$$f(\Delta) = 2\sqrt{\Delta} \qquad 0 \le \Delta \le \tfrac{1}{2} \tag{7}$$

The justification for (7) is too long to be given here.

This definition of ψ is readily extended to d-dimensional space. Each d bits of θ then determine a single bit for each coordinate, and $f(\Delta) = 4 \cdot \sqrt{d} \cdot \sqrt[d]{\Delta}$. If the rectilinear metric is to be considered, $f(\Delta) = 4 \cdot d \cdot \sqrt[d]{\Delta}$; for the sup norm metric, $f(\Delta) = 4 \cdot \sqrt[d]{\Delta}$.

5. Performance Analysis

<u>Computation Effort.</u> Since our algorithm projects the given points onto C in $O(kN)$ operations (in view of (P1)), and sorts them in $O(N\log N)$ operations, it requires $O(N\log N)$ operations to obtain the heuristic tour.

<u>Coding.</u> A short BASIC code, given in the appendix, demonstrates the ease with which this algorithm may be implemented.

<u>Worst-Case Analysis</u>. Given N sorted points $\theta_1, \ldots, \theta_N$, the tour length is bounded above by $f(\theta_N - \theta_1) + \sum_{i=1}^{N-1} f(\theta_{i+1} - \theta_i)$. Since this expression is concave in $\theta_1, \ldots, \theta_N$, it achieves a maximum of $Nf(1/N)$. In view of (7), the heuristic tour cannot exceed $2\sqrt{N}$ in length. Projected onto a square of area A, this implies

$$\text{heuristic tour} \le 2\sqrt{NA} \tag{8}$$

An interesting corollary is: optimal tour $\leq 2\sqrt{NA}$.

Probabilistic Analysis. If the points are uniformly distributed in S, then they will be uniformly distributed on C as well, and so are approximated by a Poisson process. Consequently, the expected tour length is bounded above by $N \int Ne^{-Nx} f(x)dx = \sqrt{\pi N}$. Because of the recursive nature of the algorithm, it is easy to show that, in a square of area A,

$$\frac{\text{heuristic tour}}{\sqrt{NA}} \rightarrow \text{a constant}, \quad \text{as } N \rightarrow \infty \tag{9}$$

in much the same sense that (optimal tour / NA) → .765; see Beardwood, Halton and Hammersley[1]. We have estimated the parameter in (9) be .956, and so the heuristic tour will be roughly 25% over optimum when N is large. Interestingly the ratio of heuristic tour to optimal tour does not depend on the points' distribution - so long as it has bounded density. We have also shown that the longest distance between successive points in the heuristic tour is bounded above by $2\sqrt{(A/N)\ \log N}$, almost surely, as $N \rightarrow \infty$; it may generally be found between $1.1\ \sqrt{(A/N)\log N}$ and $1.3\sqrt{(A/N)\ \log N}$.

Optional Enhancement. An $O(kN\log N)$ additional step can reduce the expected ratio of heuristic tour to optimal tour to an estimated 1.15. Briefly, the enhancement projects each point onto the boundary of its minimal containing region and attempts to interchange its position on C with that of the inverse image of the point on the opposite side of this boundary.

6. Conclusions

Our algorithm should prove useful in large applications because of its speed, only $O(N \log N)$, an order of magnitude faster than any other TSP heuristic commonly considered. It achieves this by the surprising tactic of ignoring interpoint distances (there are $O(N^2)$). Its simplicity should make it attractive for small problems as well. The heuristic tour it produces has nice properties: points may be easily inserted or deleted (in $O(\log N)$ operations) without re-solving the entire problem, and the longest distance between successive points on the tour tends to be small. (This is not true, in general, for tours that are nearest neighbor, optimal, etc.) Moreover the tour tends to be good with respect to a variety of metrics and for points drawn from general (possibly unknown) distributions. The algorithm requires no real multiplications or square roots, and so should execute quickly on microprocessor-based systems. The authors have even become relatively proficient at solving small (up to 100 points) problems by hand, graphically. Alternatively, the points and their images θ may be marked on index cards and placed in a shoebox. To produce a heuristic tour of any subset of the points, locate their cards and perform a manual sort.

APPENDIX

The following BASIC program will compute a heuristic tour of N points in a square of side S. The following remarks will help to interpret it. Lines 50-100 compute the inverse image TH(I) of the point (X(I)/S,Y(I)/S) in the unit square, taking into account only the first K=10 bits of the coordinates' binary expansion. In the J-th iteration of line 70, the IQ(J)-th quadrant of the present subsquare is selected as the next subsquare to be examined. Lines 110-220 perform a sort to identify the I-th smallest value in TH as the ID(I)-th element of the array. Lines 150-180 will be executed N $\lceil \log N \rceil$ times. Variables whose names start with I-N will contain integer values. This program may require simple modifications to run under some versions of BASIC (e.g., eliminate variable dimensions in 30).

```
10   INPUT "NUMBER OF POINTS";N : K=10 : KP=2↑(K-1)
20   INPUT "SIDE OF SQUARE";S : PRINT "NAME, X, Y:"
30   DIM A$(N),X(N),Y(N),TH(N),IQ(K),ID(N),IC(N)
40   FOR I=1 TO N : INPUT A$(I),X(I),Y(I) : NEXT I
50   FOR I=1 TO N : KR=2*KP-.1
60      KX=INT(X(I)*KR/S) : KY=INT(Y(I)*KR/S)
70      FOR J=1 TO K : JX=INT(KX/KP) : JY=INT(KY/KP)
           : KX=2*(KX-KP*JX) : KY=2*(KY-KP*JY)
           : IQ(J)=JY+3*JX-2*JX*JY : NEXT J
80      T=IQ(K)/4
90      FOR J=K-1 TO 1 STEP -1 : T=T+(6-IQ(J))/4
           : T=T-INT(T) : T=(3.5+T+IQ(J))/4 : NEXT J
100     TH(I)=T-INT(T) : ID(I)=I : NEXT I
110  M2=2 : TH(0)=-1
120  M=M2 : M2=M2*2 : IF M2<N THEN GOTO 120
130  FOR I=N TO N-M+1 STEP -1
140     J1=I : J2=I-M : J3=0 : J4=I
150     J3=J3+1
160     IF J1<0 THEN J1=0 ELSE IF J2<0 THEN J2=0
170     IF TH(ID(J1))>TH(ID(J2)) THEN IC(J3)=ID(J1)
          : J1=J1-M2 ELSE IC(J3)=ID(J2) : J2=J2-M2
180     IF J1>0 OR J2>0 THEN GOTO 150
190     IF J3<=1 THEN GOTO 220
200     FOR J=1 TO J3 : ID(J4)=IC(J) : J4=J4-M : NEXT J
210     NEXT I
220  M2=M : M=M/2 : IF M>=1 THEN GOTO 130
230  PRINT : PRINT "RANK","NAME","X","Y","THETA"
240  FOR I=1 TO N : J=ID(I)
       : PRINT I,A$(J),X(J),Y(J),TH(J) : NEXT I
250  END
```

REFERENCES

[1] J. Beardwood, J.H. Halton, and J.M. Hammersley, "The shortest path through many points", Proceedings of the Cambridge Philosophical Society 55, 299-327 (1959).

[2] J.L. Bentley and J.B. Saxe, "An analysis of two heuristics for the Euclidean travelling salesman problem", 18th Annual Allerton Conference on Communication, Control, and Computing, 41-49, October 1980.

[3] E.W. Hobson, The Theory of Functions of a Real Variable and the Theory of Fourier's Series, Vol. I, 3rd edition, Cambridge University Press and Harren Press, Washington, D.C. (1950).

[4] R.M. Karp, "Probabilistic analysis of partitioning algorithms for the travelling salesman problem in the plane", Mathematics of Operations Research 2, 209-224 (1977).

[5] C.H. Papadimitriou, "The Euclidean travelling salesman problem is NP-complete", Theoretical Computer Science 4, 237-244, (1977).

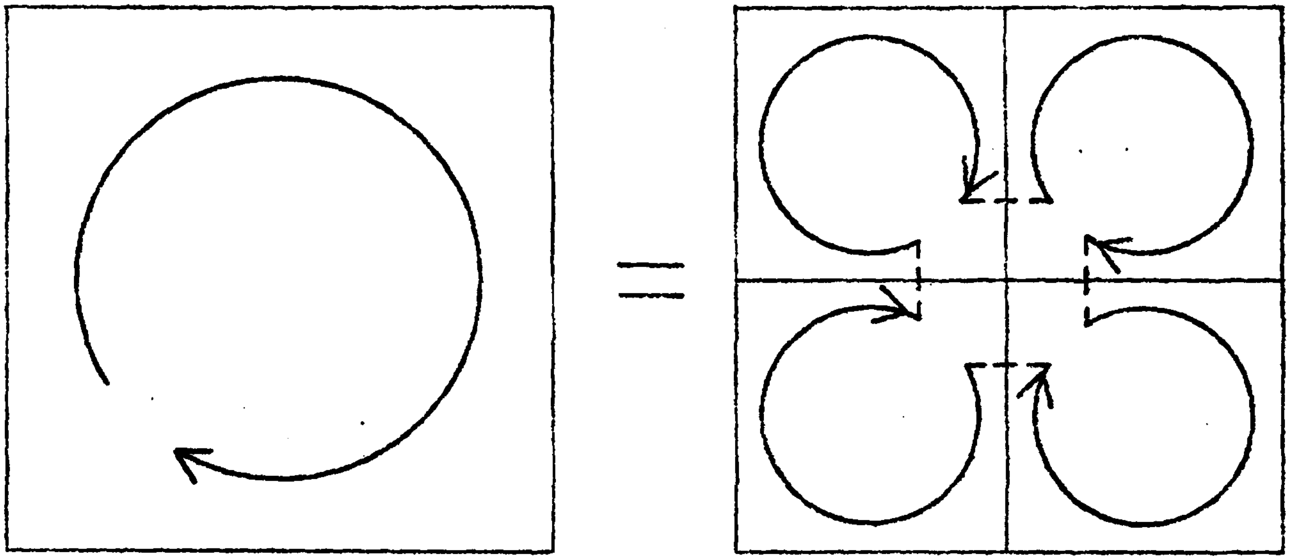
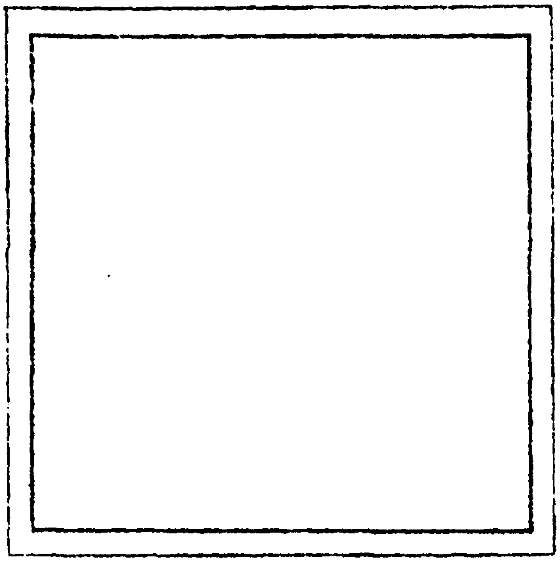
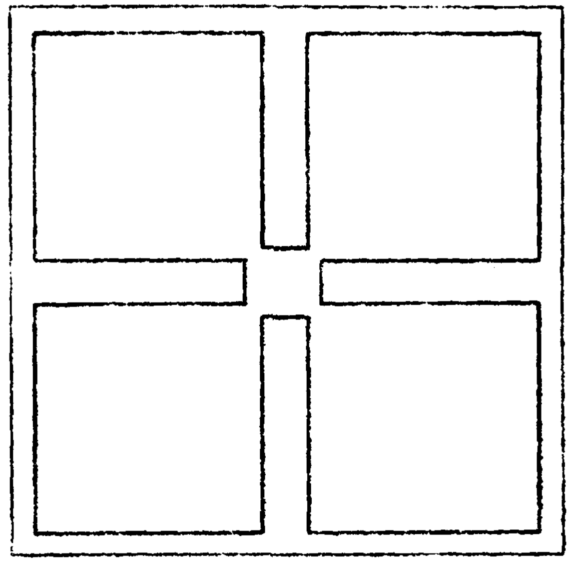


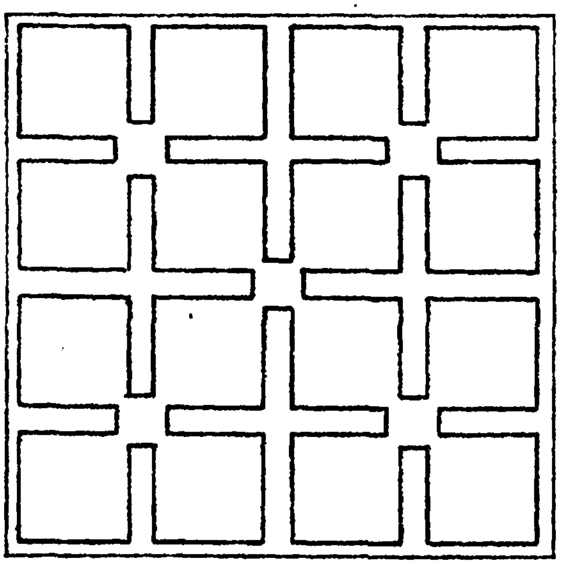
Figure 1: Recursive Structure of the Spacefilling Curve Given by (6)



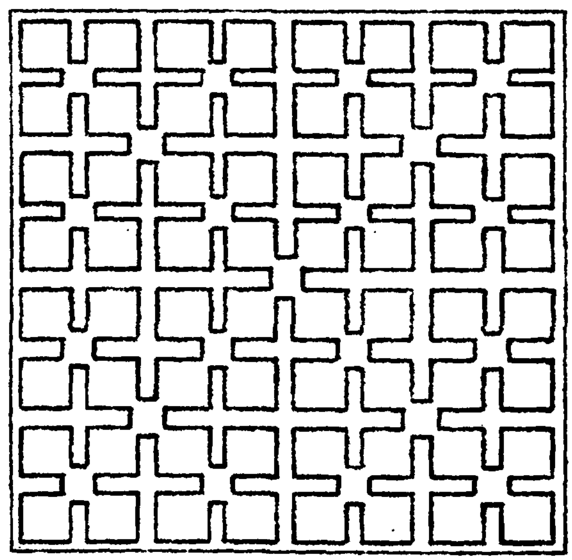
a. Initial approximation $\tilde{\psi}$



b. $T\tilde{\psi}$



c. $T^2\tilde{\psi}$



d. $T^3\tilde{\psi}$

Figure 2: Successive Approximations to ψ