



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

ADA081700

A Heuristic Method of Optimal Generalized Hyperscale Encoding for Pictorial Databases

C. C. Yang

Systems Research Branch
Space Systems Division

S. E. Chao and K. K. Suen

Department of Information Engineering
University of Ill.
Chicago, Illinois

January 15, 1980



NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

DTIC
ELECTE
MAR 12 1980
S D
B

80 2 8 014

14) NRL-8382

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8382	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 0
4. TITLE (and Subtitle) A HEURISTIC METHOD OF OPTIMAL GENERALIZED HYPERCUBE ENCODING FOR PICTORIAL DATABASES	5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem number	
7. AUTHOR(s) C.C. Yang, S.K. Chang, K.K. Singh	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375	8. CONTRACT OR GRANT NUMBER(s) 1023	
11. CONTROLLING OFFICE NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem 79-0721-0-0	12. REPORT DATE January 25, 1980
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) SBIE	13. NUMBER OF PAGES 33	15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. AD-EDU		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) AD-EDU-41		
18. SUPPLEMENTARY NOTES *Present address: Department of Information Engineering, University of Illinois, Chicago, Illinois.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Generalized hypercube encoding Similarity retrieval Updating a pictorial database Handle set Threshold density Optimal GH encoding		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Similarity retrieval from a large pictorial database can be much more efficient by encoding the the original database into certain convenient formats. Generalized hypercube (GH) encoding is one such technique. To optimize GH encoding, a heuristic approach has been formulated. Two optimization problems have been considered here: First, given the handle length m, find the optimal GH encoding. Second, given the threshold density, find the optimal GH encoding such that each generated hypercube has a density no less than a threshold density.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251100 40

CONTENTS

INTRODUCTION	1
GENERALIZED HYPERCUBE ENCODING	1
PROBLEM 1	2
ALGORITHM	2
Step 1	2
Step 2	3
Step 3	3
EXPERIMENTAL RESULTS	3
PROBLEM 2	4
ALGORITHM	7
Step 1	7
Step 2	7
Step 3	7
Step 4	7
Step 5	7
Step 6	8
Step 7	8
EXPERIMENTAL RESULTS	8
DISCUSSION	8
REFERENCES	11
APPENDIX A	13
APPENDIX B	23

ON for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION _____	
BY _____	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

A HEURISTIC METHOD OF OPTIMAL GENERALIZED HYPERCUBE ENCODING FOR PICTORIAL DATABASES

INTRODUCTION

In similarity, retrieval from a pictorial database [1], pattern recognition [2,4], and clustering analysis, it is often desired to find the set of database records (or set of patterns, n-dimensional feature vectors, etc.) that are most similar to a test record (or test pattern, test feature vector, etc.). In the case of large databases (or patterns, clusters), it is very useful to encode the original database into certain convenient format in order to facilitate similarity retrieval and updating.

In this report, we shall consider one such technique called "Generalized Hypercube" encoding [3] and describe a heuristic method of generating minimum number of GH encoded tuples.

GENERALIZED HYPERCUBE ENCODING

Given a set of points $S = \{ p_1, p_2, \dots, p_k \}$ in n-dimensional space, where $p_i = (x_1, x_2, \dots, x_n)$ denotes a point in S, we can have a family of n-dimensional GH codes. For each m , $1 \leq m \leq n + 1$, the GH_m codes are:

$$(x_1, \dots, x_{m-1}; a_m, \dots, a_n; b_m, \dots, b_n),$$

where $(x_1, x_2, \dots, x_{m-1}, z_m, \dots, z_n)$ is in S for some coordinates, z_m, \dots, z_n ; and
 $a_j = \min \{ y_j; \text{ for some coordinates, } z_k, (x_1, \dots, x_{m-1}, z_m, \dots, y_j, \dots, z_n) \text{ in } S \}$,
 $b_j = \max \{ y_j; \text{ for some coordinates, } z_k, (x_1, \dots, x_{m-1}, z_m, \dots, y_j, \dots, z_n) \text{ in } S \}$.

When $m = 1$, we are simply using the smallest n-dimensional hypercube containing S as the GH code. When $m = n+1$, the original point set S is used as the GH code. Other in-between values of m give GH codes of various levels of details.

For example, if point set $S = \{ (1,1,3), (1,1,5), (1,2,1), (1,2,4), (2,3,6) \}$, GH_1 is $\{ (1,1,1;2,3,6) \}$. GH_2 is $\{ (1;1,1;2,5), (2;3,6;3,6) \}$. GH_3 is $\{ (1,1;3;5), (1,2;1;4), (2,3;6;6) \}$. GH_4 is S itself. It should be noted that other GH codes can be generated, if we permute the coordinates. The previous example we have $\{ (3;1,1;1,1), (5;1,1;1,1), (1;1,2;1,2), (6;2,3;2,3) \}$.

Therefore, it is important to select carefully the coordinates on which the GH encoding technique can be applied, i.e. to generate a minimum number of GH encoded tuples the problem reduces to select a special "handle" set. It is easy to see that once a handle vector has been selected, the GH encoding is unique.

In this report we shall describe the following heuristic methods for problems:

- Select a handle set $\{i_1, i_2, \dots, i_{m-1}\}$ from $\{1, 2, \dots, n\}$ so that the least number of GH encoded tuples are generated.

- Suppose we define the density of a hypercube H as the number of data points in H, divided by total number of lattice integers. For example the hypercube (1,1,1;2,3,6) previously described has a density of $5/2 \cdot 3 \cdot 6 = 0.1388$. Intuitively, we would like to generate hypercubes having high densities. Therefore Problem 2 can be stated as follows:

Given a threshold density t , $0 \leq t \leq 1.0$, find optimal GH encoding such that each hypercube has density no less than t .

PROBLEM 1

Given m , the method of choosing $m-1$ handles which will generate optimal GHm codes for a point set S is based on the following ideas:

- For each handle (column) i , ($i = 1, 2, \dots, n$) consider vector $V_i = (x_{1i}, x_{2i}, \dots, x_{ki})$ where k is the number of points in the set S. Among the elements $x_{i1}, x_{i2}, \dots, x_{ik}$ those having the same value are grouped together and their associated points (in S) will consist of disjoint groups, say, $G_{1i}, G_{2i}, \dots, G_{\ell i}$. Define the count (or cardinality) of $G_{ji} = C_{ji}(i)$. We next compute a measure of the priority of the i th column p_i .

- $$p_i = \sum_{j=1}^{\ell} [C_{ji}(i)-1]^2.$$

Intuitively, we ignore those groups having only one element and put heavier weight to those groups having more than one element. Obviously, there are many other ways to define a priority measurement for each handle.

- Based on the measure P_i , ($i = 1, 2, \dots, n$) choose the best $m-1$ handles, i.e., i 's which have larger values of P_i .

Following is the algorithm to implement the above ideas:

ALGORITHM

Step 1.

To compute P_i , consider vector $V_i = (x_{1i}, x_{2i}, \dots, x_{ki})$,

a. Make a stack of all k elements of V_i .

NRL REPORT 8382

- b. Set a test element, TELMNT = top element of stack and set COUNT = 1.
- c. Replace the top element of the stack by the last element and, set $k = k - 1$; if $k < 1$, go to substep e.
- d. Restack the k elements to check that the top element = TELMNT.
 - If it does, COUNT = COUNT + 1; then go to substep c.
 - If it does not, then
{ If COUNT > 1, then $P_i = P_i + \text{COUNT} * \text{COUNT}$ and COUNT = 1. TELMNT = top element of the stack; go to substep c }.
- e. Include the last group's population count in P_i .

If COUNT > 1, then $P_i = P_i + \text{COUNT} * \text{COUNT}$.

Step 2.

Using STEP 1, compute all P_i 's for $i = 1, 2, \dots, n$. Store handle numbers in an array, such as $[c(i) = i, i = 1, \dots, n]$.

Step 3.

Sort array P in descending order, and while moving the elements of P during sort, move elements of c also in the same fashion.

For generating GH_m codes, choose handles in the first $(m - 1)$ positions of array c, which correspond to the best $(m - 1)$ values of $P_i, i = 1, 2, \dots, n$.

Once the set of handles has been selected, the GH encoding is unique.

For example, let point set $S = \{(3,1,1), (5,1,1), (1,1,2), (4,1,2), (6,2,3)\}$. Using the previous algorithm, the measure of priority, P_i for $i = 1, 2, 3$ are: $P_1 = 0, P_2 = 16, P_3 = 8$. Based on the best values of P_i , we choose handle set (x_2, x_3, x_1) and with respect to this set of handles (GH_1) is $\{(1,1,1;2,3,6)\}$. GH_2 is $\{(1;1,1;2,5), (2;3,6;6)\}$. GH_3 is $\{(1,1;3;5), (1,2;1;4), (2,3;6;6)\}$. GH_4 is $\{(1,1,3), (1,1,5), (1,2,1), (1,2,4), (2,3,6)\}$.

EXPERIMENTAL RESULTS

The previous algorithm was implemented on an IBM/370 in FORTRAN; a listing of the program is given in Appendix A. The data for the program were random numbers with normal distribution and standard deviation, a number for each column vector V_i was also a random number with uniform distribution.

YANG, CHANG AND SINGH

The selected set of handles by the previous method was used to generate GH_m encoded tuples, and their number was counted for a different number of point set and dimensions of space. To find the optimal GH_m encoding, great effort was taken, and the GH_m codes with minimum number of GH_m tuples was taken as the optimal GH_m encoding. (See Tables 1, 2, and 3.)

Few other heuristic approaches were tested to select the proper set of handles to generate optimal GH encoding, but the previous method has shown the best results and will be used in the next optimization problem.

PROBLEM 2

From the experimental results we know that the smaller the value of m (length of handles) the smaller the number of GH_m encoded tuples for a point set S , but at the same time such hypercubes tend to be sparse. Therefore, to optimize GH codes for a given

Table 1 — The number of optimal GH_m encoded tuples

No. of Points (M)	GH_2			GH_3		
	H	O	Error (%)	H	O	Error (%)
10	8	8	0.0	10	10	0.0
20	15	15	0.0	20	20	0.0
30	22	22	0.0	30	30	0.0
40	12	12	0.0	33	33	0.0
50	34	34	0.0	50	50	0.0
60	16	16	0.0	56	55	1.02
70	24	24	0.0	68	68	0.0
80	30	30	0.0	75	75	0.0
90	27	27	0.0	88	87	1.16
100	20	20	0.0	86	86	0.0

Dimension N = 3
H = heuristic
O = optimum

NRL REPORT 8382

Table 2 — The number of optimal GH_m encoded tuples

No. of Points (M)	GH_2			GH_3			GH_4		
	H	O	Error (%)	H	O	Error (%)	H	O	Error (%)
10	8	8	0.0	10	10	0.0	10	10	0.0
20	16	16	0.0	20	20	0.0	20	20	0.0
30	10	10	0.0	29	28	3.57	30	30	0.0
40	21	21	0.0	40	39	2.56	40	40	0.0
50	16	16	0.0	44	44	0.0	50	50	0.0
60	25	25	0.0	59	59	0.0	60	60	0.0
70	28	28	0.0	68	68	0.0	70	70	0.0
80	21	21	0.0	74	71	4.23	80	80	0.0
90	18	18	0.0	75	75	0.0	90	90	0.0
100	22	22	0.0	93	93	0.0	100	100	0.0

Dimension N = 4
H = heuristic
O = optimum

Table 3 — The number of optimal GH_m encoded tuples

No. of Points (M)	GH_2			GH_3			GH_4			GH_5		
	H	O	Error (%)	H	O	Error (%)	H	O	Error (%)	H	O	Error (%)
10	8	8	0.0	10	10	0.0	10	10	0.0	10	10	0.0
20	13	13	0.0	20	19	5.26	20	20	0.0	20	20	0.0
30	10	10	0.0	29	29	0.0	30	30	0.0	30	30	0.0
40	16	16	0.0	38	38	0.0	40	40	0.0	40	40	0.0
50	21	21	0.0	49	49	0.0	50	50	0.0	50	50	0.0
60	18	18	0.0	55	55	0.0	59	59	0.0	60	60	0.0
70	18	18	0.0	64	63	1.58	70	70	0.0	70	70	0.0
80	21	21	0.0	75	75	0.0	80	79	1.26	80	80	0.0
90	26	26	0.0	84	84	0.0	90	90	0.0	90	90	0.0
100	22	22	0.0	96	92	4.35	100	100	0.0	100	100	0.0

Dimension $N = 5$
 H = heuristic
 O = optimum

threshold density t , find a set of handles with minimum length m , so that the resultant hypercubes have densities no less than threshold density, t .

Two special cases can be solved immediately. If $t = 0$, then we can select $m = 1$, so that a single n -dimensional hypercube is generated, and GH_1 becomes the optimum GH encoding. If $t = 1.0$, then all the hypercubes must have unit density, and we can select $m = n + 1$ so that the original point set S becomes the optimum GH encoding. For other values of t , m should fall between 1 and $n + 1$.

ALGORITHM

The algorithm optimize GH encoding for a given threshold density.

Step 1.

Start with $m = 1$;

generate GH_1 codes, which is one hypercube. If the density of this hypercube is \geq threshold density, then this is the optimum GH encoding, and go to Step 7,

otherwise, set $m = m + 1$.

Step 2.

By using the previous heuristic method discussed in Problem 1, choose a set of handles which will generate the optimal GH_m encoding for a given m .

Set a flag = 0; and

$\text{NUM} = 0$, where NUM will be the number of GH_m encoded tuples with respect to the chosen handle vector of length m .

Step 3.

a. Generate the next hypercube with respect to the above chosen handle vector and compute its density.

b. If density of this hypercube \geq threshold density, and

if the number of points in this hypercube > 1 , then

{ include this hypercube in optimal GH encoding.

$\text{NUM} = \text{NUM} + 1$; and

delete the points included in this hypercube from the original point set. }

c. If the density of this hypercube $<$ threshold density, then set the flag = 1

Step 4.

Repeat Step 3 until all the points in point set have been considered once.

Step 5.

If the number of points in remaining point set = 0 then go to Step 7.

If (number of points in point set = 1 or

YANG, CHANG AND SINGH

NUM \leq 0 and flag = 0), then go to Step 6.

If (NUM \leq 0 and flag = 1), then $m = m + 1$.

Go to Step 2.

Step 6.

Generate GH_m encoded tuples with respect to the last handle vector of length m . Each GH_m encoded tuple is included in the optimal GH encoding.

Step 7.

Stop.

As an illustration, if point set $S = \{ (3,1,1), (5,1,1), (1,1,2), (4,1,2), (6,2,3) \}$ and threshold density = 50% (0.5), then with the previous algorithm, optimal encoding is attended as $\{ (1;1,2;2,5), (2;3,6;3,6) \}$ and the density of each hypercube is \geq 50%.

EXPERIMENTAL RESULTS

The previous algorithm also was implemented on an IBM/370 in FORTRAN. A listing of the program is given in Appendix B. The test data were again random numbers.

The results are in Tables 4, 5, and 6.

DISCUSSION

In this section, some interesting applications of the generalized GH encoding technique are discussed.

Reference 1 suggests that a relational file R can be characterized by a logical expression E , so that every n -tuple (x_1, x_2, \dots, x_n) in R satisfies this logical expression.

The generalized GH encoding technique can be used to find a suitable logical expression characterizing a relational file.

For example, if S is as given in the Generalized Hypercube Encoding section of this report and $m = 1$, then $GH_1 = \{ (1,1,1;2,3,6) \}$ and the corresponding logical expression is $(1 \leq x_1) \& (1 \leq x_2) \& (1 \leq x_3) \& (x_1 \leq 2) \& (x_2 \leq 3) \& (x_3 \leq 6)$;

when $m = 2$, then $GH_2 = \{ (1;1,1;2,5), (2;3,6;3,6) \}$ and the corresponding logical expression is $(x_1 = 1) \& (1 \leq x_2) \& (1 \leq x_3) \& (x_3 \leq 5) \vee ((x_1 = 2) \& (x_2 = 3) \& (x_3 = 6))$, thus, each GH encoding corresponds to a logical characteristic expression for the relational file.

NRL REPORT 8382

Table 4 — The number of optimal GH encoded tuples

(M)	t=10	t=20	t=30	t=40	t=50	t=60	t=70	t=80	t=90
10	6	7	7	7	9	9	10	10	10
20	11	14	18	18	18	18	18	18	18
30	9	23	24	24	24	26	28	28	28
40	18	36	36	36	36	37	38	38	38
50	23	36	39	40	42	48	50	50	50
60	12	43	43	48	53	55	58	58	58
70	21	40	46	46	49	52	64	64	64
80	1	52	53	54	57	57	60	60	60
90	1	45	50	52	56	77	80	84	84
100	1	40	52	58	60	81	82	91	91

M = number of points in point set
t = percent threshold density
Dimension N = 3

YANG, CHANG AND SINGH

Table 5 — The number of optimal GH encoded tuples

(M)	t=10	t=20	t=30	t=40	t=50	t=60	t=70	t=80	t=90
10	7	7	9	9	9	9	10	10	10
20	14	17	17	18	18	18	18	18	18
30	25	28	28	29	29	29	29	29	29
40	32	36	39	39	39	39	39	39	39
50	38	45	46	49	50	50	50	50	50
60	46	52	56	57	57	57	59	59	59
70	42	57	60	68	68	68	70	70	70
80	45	72	77	78	79	79	79	79	79
90	58	72	78	79	82	87	88	88	88
100	66	78	81	90	91	92	92	93	93

M = number of points in point set
t = percent threshold density
Dimension N = 4

Table 6 — The number of optimal GH encoded tuples

(M)	t=10	t=20	t=30	t=40	t=50	t=60	t=70	t=80	t=90
10	10	10	10	10	10	10	10	10	10
20	20	20	20	20	20	20	20	20	20
30	28	29	30	30	30	30	30	30	30
40	40	40	40	40	40	40	40	40	40
50	47	50	50	50	50	50	50	50	50
60	57	58	58	58	58	59	59	59	59
70	68	68	69	69	69	70	70	70	70
80	76	77	77	77	77	79	79	79	79
90	79	86	87	87	88	89	90	90	90
100	95	100	100	100	100	100	100	100	100

M = number of points in point
t = percent threshold density
Dimension N = 5

In addition to applications in database characterization, the previously described technique also can be applied in clustering analysis where clusters are described by hypercubes with handles. Thus optimization technique will generate optimal encoding for the clusters.

A third application can be the description of n-dimensional pictures, which will have a practical significance in computer graphics and image processing applications.

REFERENCES

1. S.K. Chang et al. "Design Considerations of a Pictorial Database System," *International Journal of Policy Analysis and Information Systems* 1 (No. 2) (Jan. 1978).
2. K.S. Fu, *Syntectic Methods in Pattern Recognition*, Academic Press, 1973, pp 69-82.
3. C.C. Yang and S.K. Chang, "Encoding Techniques for Efficient Similarity Retrieval From Pictorial Databases", in *Pictorial Information Systems*, Springer-Verlag, 1979.
4. T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, 1977.

APPENDIX A

```

1. //JUNK JOB
2. /*JOBPARM T=(8,0),L=2,R=256
3. // EXEC FORTGCLG
4. //FORT.SYSIN DD *
5. CCCCCCCCC
6. C
7. C MAIN PROGRAM TO GENERATE OPTIMAL GHM ENCODING FOR A POINTSET BY
8. C A HEURISTIC METHOD & TO CHECK THE RESULTS. NUMBER OF THESE GHM
9. C ENCODED TUPLES IS COMPARED WITH THE MINIMUM (REAL OPTIMUM) NO.
10. C OF GHM ENCODED TUPLES FOUND IN EXHAUSTIVE APPROACH.
11. C
12. C CALLS:
13. C (1) GENPTS: TO GENERATE TEST DATA I.E. A POINT SET.
14. C (2) HEURIS: TO GENERATE OPTIMAL GHM ENCODED TUPLES BY A
15. C HEURISTIC METHOD.
16. C (3) EXHAUS: TO GENERATE OPTIMAL GHM ENCODING BY EXHAUSTIVE
17. C APPROACH.
18. C
19. CCCCCCCCC
20. C INTEGER PTSET(110,6),P(6),NCODE(5,3)
21. C IFL=1
22. C N = 4
23. C M = 50
24. C CALL GENPTS(PTSET,M,N)
25. C IF(IFL .EQ. 1) GO TO 440
26. C PRINT 1
27. C 1 FORMAT('1',10X, 'INITIAL POINT SET IS:')
28. C DO 100 I = 1,M
29. C PRINT 2, (PTSET(I,J),J=1,N)
30. C 100 CONTINUE
31. C 2 FORMAT('0',10X, '( '.5I3, ' )' )
32. C PRINT 3
33. C 3 FORMAT(////,100(' '))
34. C PRINT 4
35. C 4 FORMAT(//,25X, 'HEURISTIC APPROACH')
36. C PRINT 5
37. C 5 FORMAT(25X, '-----')
38. C PRINT 6
39. C 6 FORMAT(//,100(' '))
40. C 440 CALL HEURIS(PTSET,M,N,NCODE,IFL)
41. C IF(IFL .EQ. 1) GO TO 442
42. C PRINT 11
43. C 11 FORMAT(////,100(' '))
44. C PRINT 12
45. C 12 FORMAT(//,21X, 'EXHAUSTIC APPROACH')
46. C PRINT 13
47. C 13 FORMAT(21X, '-----')
48. C PRINT 14
49. C 14 FORMAT(//,100(' '))
50. C 442 CALL EXHAUS(PTSET,M,N,NCODE,IFL)
51. C N1 = N-1
52. C PRINT 20,M
53. C 20 FORMAT(//,10X, 'NO. OF POINTS IS = ', I4)
54. C PRINT 22,N
55. C 22 FORMAT(10X, 'NO. OF DIMENSINS = ', I3)
56. C PRINT 15
57. C 15 FORMAT(//,20X, 'HEURISTIC',10X, 'OPTIMUM',4X, 'WORST',4X, 'ERR')
58. C DO 200 I = 1,N1
59. C ERR = (NCODE(I,1) - NCODE(I,2))*100./NCODE(I,2)
60. C 200 PRINT 16, I, (NCODE(I,J),J= 1,3),ERR

```

YANG, CHANG AND SINGH

```

61.      16      FFORMAT('0'.10X,'GH'.12,3(10X.14),F16.3)
62.      IF(IFL .EQ. 1) PRINT 66
63.      66      FFORMAT(//.100('.'))
64.      500     CCONTINUE
65.      600     CCONTINUE
66.      STOP
67.      END
68.      CCCCCCCC
69.      C
70.      C      RANDJM(X): GENERATES A RANDOM NUMBER X . 0<= X <=1.0 WITH
71.      C      UNIFORM DISTRIBUTION.
72.      C
73.      CCCCCCCCCC
74.      C      SUBROUTINE RANDOM(X)
75.      C      REAL X
76.      C      INTEGER A/19727/, MULT/25211/, BASE/32768/
77.      C      A = MOD(MULT*A,BASE)
78.      C      X = FLOAT(A)/FLOAT(BASE)
79.      C      RETURN
80.      C      END
81.      CCCCCCCCCC
82.      C
83.      C      GENPTS(PTSET,M,N):- TO GENERATE TEST DATA SET IN ARRAY PTSET.
84.      C      WHICH CONTAINS M NUMBER OF POINTS IN N DIMENSIONAL SPACE.
85.      C      CALLS:
86.      C      (1) FA03A: A LIBRARY SUBROUTINE TO GENERATE RANDOM NUMBERS OF
87.      C      NORMAL DISTRIBUTION WITH A GIVEN STANDARD
88.      C      DEVIATION.
89.      C      (2) RANDOM: TO GET ANOTHER RANDOM NUMBER TO BE USED AS FIX
90.      C      STANDARD DEVIATION FOR ONE COLUMN.
91.      C
92.      CCCCCCCCCC
93.      C      SUBROUTINE GENPTS(PTSET, M,N)
94.      C      INTEGER PTSET(110,6)
95.      C      DO 100 J = 1, N
96.      C      CALL RANDOM(U)
97.      C      STD = 19.0*U + 3
98.      C      DO 100 I = 1, M
99.      C      CALL FA03A(STD,X)
100.     C      PTSET(I,J) = INT(X + .5)
101.     100 CONTINUE
102.     RETURN
103.     END
104.     CCCCCCCCCC
105.     C
106.     C      PROCES(PTSET,P,M,N):- COMPUTES THE MEASURE OF PRIORITY P(I)
107.     C      FOR EACH COLUMN I WHERE I = 1,...,N FOR A GIVEN POINT SET
108.     C      IN ARRAY PTSET CONTAINING M POINTS IN N DIMENSIONAL SPACE.
109.     C      CALLS:
110.     C      (1) HPODER: TO COMPUTE P(I) FOR ONE PARTICULAR VALUE OF I.
111.     C
112.     CCCCCCCCCC
113.     C      SUBROUTINE PROCES(PTSET,P,M,N)
114.     C      INTEGER PTSET(110,6),TEMPR(110),P(6)
115.     C      DO 100 I = 1,N
116.     C      DO 200 J = 1,M
117.     200 TEMPR(J) = PTSET(J,I)
118.     C      CALL HPODER(I,TEMPR,M,P)
119.     100 CONTINUE
120.     RETURN

```

NRL REPORT 8382

```

121.      END
122.      CCCCCCCCCC
123.      C
124.      C  HPODER(I,TEMPR,M,P):- COMPUTES P(I) FOR A COLUMN VECTOR TEMPR
125.      C      AND USES HEAP STRUCTURE.
126.      C  CALLS:
127.      C      (1) REHEAP: TO MAKE HEAP OF ELEMENTS OF ARRAY TEMPR.
128.      C
129.      CCCCCCCCCC
130.      SUBROUTINE HPODER(I,TEMPR,M,P)
131.      INTEGER TEMPR(110),P(6)
132.      INTEGER TEMP,ELEMENT,PSUM,SUM
133.      MI = M
134.      L = M/2 + 1
135.      10  L = L-1
136.      CALL REHEAP(L,MI,TEMPR)
137.      IF (L .GT. 1) GOTO 10
138.      ELEMENT = 0
139.      PSUM = 0
140.      SUM = 0
141.      20  TEMP = TEMPR(I)
142.      TEMPR(I) = TEMPR(MI)
143.      IF (ELEMENT .EQ. TEMPIGO TO 30
144.      IF (SUM .GT. 1) PSUM = PSUM + SUM * SUM
145.      ELEMENT = TEMP
146.      SUM = 1
147.      GO TO 40
148.      30  SUM = SUM + 1
149.      MI = MI - 1
150.      40  IF (MI .GT. 0) GO TO 50
151.      IF (SUM .GT. 1) PSUM = PSUM + SUM * SUM
152.      P(I) = PSUM
153.      RETURN
154.      50  CALL REHEAP(I,MI,TEMPR)
155.      GOTO 20
156.      END
157.      CCCCCCCCCC
158.      C
159.      C  REHEAP(L,MI,TEMPR):- MAKES HEAP OF ELEMENTS OF TEMPR STARTING
160.      C      FROM TEMPR(L) TO TEMPR(MI).
161.      C
162.      CCCCCCCCCC
163.      SUBROUTINE REHEAP(L,MI,TEMPR)
164.      INTEGER TEMPR(110)
165.      INTEGER FLAG , X
166.      II = L
167.      JJ = 2*II
168.      X = TEMPR(II)
169.      FLAG = 1
170.      IF (JJ .GT. MI) RETURN
171.      40  IF (JJ .GE. MI) GOTO 10
172.      IF (TEMPR(JJ) .LT. TEMPR(JJ+1)) JJ = JJ+1
173.      10  IF (X .GE. TEMPR(JJ)) GOTO 20
174.      TEMPR(II) = TEMPR(JJ)
175.      II = JJ
176.      JJ = 2*II
177.      TEMPR(II) = X
178.      GOTO 30
179.      20  FLAG = 0
180.      30  IF (JJ .LE. MI .AND. FLAG .EQ. 1) GOTO 40

```

YANG, CHANG AND SINGH

```

181.          RETURN
182.          END
183.          CCCCCCCCCC
184.          C
185.          C   SORT(P,O,N):- SORTS P ARRAY CONTAINING N ELEMENTS AND DURING
186.          C   THIS SORT ALSO MOVES THE ELEMENTS OF ARRAY O IN THE SAME
187.          C   FASHION.
188.          C
189.          CCCCCCCCCC
190.          SUBROUTINE SORT(P,O,N)
191.          INTEGER O(6),P(6)
192.          INTEGER FLAG, TOP, TEMP, BOTTOM
193.          FLAG = 1
194.          TOP = 1
195.          BOTTOM = N
196.          DO 100 I = 1,11
197.          IF(FLAG .NE. 0 .AND. TOP .LT. BOTTOM) GO TO 20
198.          RETURN
199.          20  FLAG = 0
200.             I1 = BOTTOM - 1
201.             DO 100 I = 1,11
202.             IF (P(I) .GE. P(I+1)) GO TO 100
203.             X = P(I)
204.             P(I) = P(I+1)
205.             P(I+1) = X
206.             TEMP = O(I)
207.             O(I) = O(I+1)
208.             O(I+1) = TEMP
209.             FLAG = 1
210.          100 CONTINUE
211.             BOTTOM = BOTTOM - 1
212.             GO TO 10
213.          END
214.          CCCCCCCCCC
215.          C
216.          C   SORTPT(P,SET,O,M,N):- TO SORT POINTS OF POINTSET IN PTSET W.R.T.
217.          C   HAVDLE VECTOR IN ARRAY O. HEAP SORT IS USED FOR THIS.
218.          C   CALLS:
219.          C   (1) REHIP: TO MAKE HEAP.
220.          C   (2) COMPAR: TO COMPARE ORDER OF TWO POINTS.
221.          C   (3) EXCHAN: TO EXCHANGE POSITIONS OF TWO POINTS IN PTSET.
222.          C
223.          CCCCCCCCCC
224.          SUBROUTINE SORTPT(P,SET,O,M,N)
225.          INTEGER PTSET(110,6),O(6)
226.          M1 = M
227.          L = M1/2 + 1
228.          L = L - 1
229.          10  CALL REHIP(L,M1,M,N,O,PTSET)
230.             IF(L .GT. 1) GOTO 10
231.             CALL EXCHAN(1,M1,N,PTSET)
232.             M1 = M1 - 1
233.             CALL REHIP(1,M1,M,N,O,PTSET)
234.             IF(M1 .GT. 1) GO TO 20
235.             RETURN
236.          END
237.          CCCCCCCCCC
238.          C
239.          C   REHIP(L,M1,M,N,O,PTSET):- MAKES A HEAP OF ELEMENTS OF PTSET
240.          C   STARTING FROM L TO M1. PTSET CONTAINS M POINTS IN N
          C   DIMENSIONAL SPACE.

```

NRL REPORT 8382

```

241. C
242. CCCCCCCCCC
243. SUBROUTINE REHIP(L,M,N,O,PTSET)
244. INTEGER PTSET(110,6),D(6)
245. INTEGER FLAG ,X
246. I = L
247. J = 2*I
248. X = M+1
249. DO 5 LL=1,N
250. 5 PTSET(X,LL) = PTSET(I,LL)
251. FLAG = 1
252. IF(J .GT. M) RETURN
253. 40 IF(J .GE. M) GO TO 10
254. CALL COMPAR(J,J+1,N,O,PTSET,ICOMP)
255. IF (ICOMP .LT.0) J = J+1
256. 10 CALL COMPAP(X,J,N,O,PTSET,ICOMP)
257. IF (ICOMP .GE.0) GO TO 20
258. DO 100 LL = 1,N
259. 100 PTSET(I,LL) = PTSET(J,LL)
260. I = J
261. J = 2*I
262. DO 200 LL = 1,N
263. 200 PTSET(I,LL) = PTSET(X,LL)
264. GO TO 30
265. 20 FLAG = 0
266. 30 IF (J .LE. M) .AND. FLAG .EQ.1) GO TO 40
267. RETURN
268. END
269. CCCCCCCCCC
270. C
271. C COMPAR(I,J,N,O,PTSET,ICOMP):- COMPARES TWO POINTS I & J W.R.T.
272. C 4 HANDLE VECTOR IN O, AND RETURNS IN ICOMP:
273. C 1 IF ITH POINT > JTH POINT.
274. C 0 IF ITH POINT = JTH POINT.
275. C -1 IF ITH POINT < JTH POINT.
276. C
277. CCCCCCCCCC
278. SUBROUTINE COMPAR(I,J,N,O,PTSET,ICOMP)
279. INTEGER PTSET(110,6),D(6)
280. II = I
281. 100 III = D(III)
282. IF (PTSET(I,III) .NE. PTSET(J,III)) GO TO 10
283. IF (II .LT. N) GO TO 5
284. ICOMP = 0
285. RETURN
286. 5 II = II+1
287. GO TO 100
288. 10 IF(PTSET(I,III) .LT. PTSET(J,III)) ICOMP = -1
289. IF (PTSET(I,III).GT.PTSET(J,III)) ICOMP = 1
290. RETURN
291. END
292. CCCCCCCCCC
293. C
294. C EXCHAN(I,J,N,PTSET):-SWITCHES THE POSITIONS OF TWO POINTS I &
295. C J IN ARRAY PTSET WHICH CONTAINS A POINT SET IN N DIMENSIONAL
296. C SPACE.
297. C
298. CCCCCCCCCC
299. SUBROUTINE EXCHAN(I,J,N,PTSET)
300. INTEGER PTSET(110,6)

```

YANG, CHANG AND SINGH

```

301.      DO 100 II = 1,N
302.      ITEMP = PTSET(I,II)
303.      PTSET(I,II) = PTSET(J,II)
304.      PTSET(J,II) = ITEMP
305. 100    CONTINUE
306.      RETURN
307.      END
308. CCCCCCCCC
309. C
310. C ENCODE(MM,M,N,O,PTSET,NUM,IFL):-GENERATES GHMM CODES FOR A
311. C SET PTSET, CONTAINING M POINTS IN N DIMENSIONAL SPACE.
312. C NUM WILL CONTAIN THE NUMBER OF SUCH GHMM ENCODED TUPLES IN IT.
313. C IFL IS A FLAG TO TURN OFF AND ON THE PRINTING OF INTERMEDIATE
314. C RESULTS.
315. C CALLS:
316. C (1) MINMAX: TO FIND THE MINIMUM & MAXIMUM ELEMENT IN A
317. C PARTICULAR COLUMN FOR SOME POINTS CONSIDERED TO BE IN
318. C CURRENT HYPERCUBE.
319. C
320. CCCCCCCCC
321. C SUBROUTINE ENCODE(MM,M,N,O,PTSET,NUM,IFL)
322. C INTEGER PTSET(110,6),O(6),CODE(12)
323. C IF(IFL .EQ. 1) GO TO 44
324. C PRINT 1
325. C   1  FORMAT(//)
326. C   5  PRINT 5,MM
327. C   5  FORMAT(11X,'GH',12,' CODES ARE:')
328. C 44  IF (MM .EQ. 1) GO TO 10
329. C     IF(MM .EQ. N+1) GO TO 20
330. C     NUM = 0
331. C     NI = MM-1
332. C     NI = M + 1
333. C     I = 1
334. C     J = 1
335. C     DO 100 II = 2,MI
336. C       IF (II .GT. M) GO TO 40
337. C       FLAG = 0
338. C       DO 200 JJ = 1,NI
339. C         JJ = O(JJ)
340. C         IF(PTSET(II-I,JJ) .EQ. PTSET(II,JJ)) GO TO 200
341. C         FLAG = 1
342. C         GO TO 30
343. C 200  CONTINUE
344. C 30  IF (FLAG .EQ. 1) GO TO 40
345. C     J = J + 1
346. C     GO TO 100
347. C 40  DO 300 JJ = 1,NI
348. C     JJ = O(JJ)
349. C     CODE(JJ) = PTSET(I,JJ)
350. C 300  CONTINUE
351. C     L = NI
352. C     NN = N - NI
353. C     DO 400 JJ = MM,N
354. C     JJ = O(JJ)
355. C     CALL MINMAX(PTSET,M,JJ,I,J,MIN,MAX)
356. C     L = L + 1
357. C     CODE(L) = MIN
358. C 400  CODE(L+NN) = MAX
359. C     N2 = L
360. C     N3 = L + NN

```

NRL REPORT 8382

```

361.          IF (IFL .EQ. 1) GO TO 85
362.          PRINT 11, (CODE(I), I=1, N)
363.          11      FORMAT ('0', 10X, '( ', 3I3)
364.          PRINT 13, (CODE(I), I=MM, N2)
365.          13      FORMAT (' ', 22X, '( ', 3I3)
366.          N2 = N2 + 1
367.          PRINT 15, (CODE(I), I= N22, N3)
368.          15      FORMAT (' ', 33X, '( ', 3I3)
369.          PRINT 16
370.          16      FORMAT (' ', 44X, ' ')
371.          85      I = 11
372.          J = 11
373.          NUM = NUM + 1
374.          100     CONTINUE
375.          RETURN
376.          10      DO 111 I = 1, N
377.          JJ = 0(I)
378.          CALL MINMAX(PYSET, M, JJ, 1, M, MIN, MAX)
379.          CODE(I) = MIN
380.          CODE(I+N) = MAX
381.          111     CONTINUE
382.          IF (IFL .EQ. 1) RETURN
383.          PRINT 8, (CODE(I), I=1, N)
384.          8        FORMAT ('0', 10X, '( ', 4I3)
385.          NN = N + 1
386.          N2 = 2*N
387.          PRINT 90, (CODE(I), I=NN, N2)
388.          90       FORMAT (' ', 24X, '( ', 4I3)
389.          PRINT 93
390.          93       FORMAT (' ', 38X, ' ')
391.          RETURN
392.          20      DO 222 I = 1, M
393.          DO 220 J = 1, N
394.          JJ = 0(J)
395.          220     CODE(J) = PYSET(I, JJ)
396.          IF (IFL .EQ. 1) GO TO 222
397.          PRINT 22, (CODE(J), J = 1, N)
398.          22      FORMAT ('0', 10X, '( ', 4I3)
399.          PRINT 23
400.          23      FORMAT (' ', 24X, ' ')
401.          222     CONTINUE
402.          RETURN
403.          END
404.          CCCCCCCCC
405.          C
406.          C      MINMAX(PYSET, M, ICOL, I, J, MIN, MAX) :- TO FIND THE MINIMUM &
407.          C      MAXIMUM ELEMENTS OF COLUMN ICOL, STARTING FROM I TO J IN ARRAY
408.          C      PYSET.
409.          C
410.          CCCCCCCCC
411.          C      SUBROUTINE MINMAX(PYSET, M, ICOL, I, J, MIN, MAX)
412.          C      INTEGER PYSET(110, 6)
413.          C      MIN = PYSET(I, ICOL)
414.          C      MAX = PYSET(I, ICOL)
415.          C      DO 100 II = 1, J
416.          C      IF (PYSET(II, ICOL) .LT. MIN) MIN = PYSET(II, ICOL)
417.          C      IF (PYSET(II, ICOL) .GT. MAX) MAX = PYSET(II, ICOL)
418.          C      100 CONTINUE
419.          C      RETURN
420.          C      END

```

YANG, CHANG AND SINGH

```

421. CCCCCCCCC
422. C
423. C HEURIS(PTSET,M,N,NCODE,IFL):- TO STORE THE NUMBER OF GHI ENCODED
424. C TUPLES FOR I=1,2,...,N+1 IN ARRAY NCODE. THE SET OF HANDLES
425. C IS CHOSEN BY HEURISTIC METHOD.
426. C CALLS:
427. C (1) PROCES: TO COMPUTE THE MEASURE OF PRIORITY P(I) FOR EACH
428. C COLUMN I.
429. C (2) SORT: TO GET THE PROPER SET OF HANDLES CORRESPONDING TO
430. C TO BEST VALUES OF THEIR PRIORITY P(I).
431. C (3) SORTPT: TO SORT POINT SET PTSET W.R.T. CHOSEN SET OF
432. C HANDLES.
433. C (4) ENCODE: TO GENERATE GH ENCODED TUPLES.
434. C
435. CCCCCCCCC
436. C SUBROUTINE HEURIS(PTSET,M,N,NCODE,IFL)
437. C INTEGER PTSET(110,6),O(6),P(6),NCODE(5,3)
438. C CALL PROCES(PTSET,P,M,N)
439. C PRINT 1
440. C 1 FORMAT(///,11X,'PRINTOUT OF ARRAY P IS:')
441. C PRINT 2 ,(P(I),I=1,N)
442. C 2 FORMAT(10X,5I10)
443. C DO 200 I = 1,N
444. C 200 O(I) = I
445. C CALL SORT(P,O,N)
446. C PRINT 3
447. C 3 FORMAT('0',10X,'HANDLES CHOSEN ARE:')
448. C PRINT 4 ,(O(I),I=1,N)
449. C 4 FORMAT(' ',10X,4I5)
450. C CALL SORTPT(PTSET,O,M,N)
451. C IF(IFL.EQ.1) GO TO 44
452. C PRINT 5
453. C 5 FORMAT('0',10X,'SORTED POINT SET IS:')
454. C DO 300 I = 1,M
455. C 300 PRINT 10,(PTSET(I,J),J=1,N)
456. C 10 FORMAT(11X,'(' ,4I3,')')
457. C PRINT 11
458. C 11 FORMAT(///,20X,'FOLLOWING ARE GH CODES W.R.T.ABOVE HANDLES:')
459. C 44 VI = N + 1
460. C DO 400 I = 1,N1
461. C CALL ENCODE(I,M,N,O,PTSET,NUM,IFL)
462. C IF( I .LE.1 .OR. I .GE. N1) GO TO 400
463. C NCODE(I-1,1) = NUM
464. C NCODE(I-1,2) = NUM
465. C NCODE(I-1,3) = NUM
466. C 400 CONTINUE
467. C RETURN
468. C END
469. CCCCCCCCC
470. C
471. C EXHAUS(PTSET,M,N,NCODE,IFL):- TO STORE THE MINIMUM NUMBER OF GHI
472. C ENCODED TUPLES FOR I=1,2,...,N+1 IN ARRAY NCODE. ALL THE
473. C POSSIBLE SET OF HANDLES ARE USED TO GENERATE GH CODES & THE
474. C ONE WITH MINIMUM NUMBER OF TUPLE IS TAKEN AS OPTIMAL GH
475. C CODES.
476. C CALLS:
477. C (1) RCJMR: TO FIN ALL POSSIBLE SETS OF HANDLES.
478. C (2) SORTPT: TO SORT THE POINTSET W.R.T. CURRENT SET OF HANDLES
479. C (3) ENCODE: TO GENERATE GH ENCODED TUPLES.
480. C

```

NRL REPORT 8382

```

481.      20      RETURN
482.      10      DO 222 I = 1,N
483.              JJ = O(I)
484.              CALL MINMAX(PTSET,ORDER,JJ,1,NPTS,MIN,MAX)
485.              CODE(I) = MIN
486.              CODE(I+N) = MAX
487.      222      CONTINUE
488.              DEN = DENSTY(CODE,MM,N,1,NPTS)
489.              IF(DEN .GE. THRESH) GOTO 444
490.              NUM = 0
491.              RETURN
492.      444      IF (FLG .EQ. 1) GO TO 666
493.              PRINT 21,(CODE(I),I=1,N)
494.      21      FORMAT('0',10X,'(',.6I3)
495.              N1 = N+1
496.              N2 = N+N
497.              PRINT 22,(CODE(I),I=N1,N2)
498.      22      FORMAT('+',.30X,';',.6I3)
499.              PRINT 23
500.      23      FORMAT('+',.50X,')')
501.      666      NUM = 1
502.              RETURN
503.              END
504.      CCCCCCCCC
505.      C
506.      C   MINMAX(PTSET,ORDER,JJ,I,J,MIN,MAX):- FIND THE MINIMUM & MAXIMUM
507.      C   OF ELEMENTS OF COLUMN JJ ,STORED IN PTSET,THEIR POINTERS ARE
508.      C   STORED IN ARRAY ORDER STARTING FROM I TO J.
509.      C
510.      CCCCCCCCC
511.      C   SUBROUTINE MINMAX(PTSET,ORDER,JJ,I,J,MIN,MAX)
512.      C   INTEGER PTSET(110,6),ORDER(110)
513.      C   I1 = ORDER(I)
514.      C   MIN = PTSET(I1,JJ)
515.      C   MAX = PTSET(I1,JJ)
516.      C   DO 100 K = I,J
517.      C     I1 = ORDER(K)
518.      C     IF(PTSET(I1,JJ) .LT. MIN) MIN = PTSET(I1,JJ)
519.      C     IF( PTSET(I1,JJ) .GT. MAX) MAX= PTSET(I1,JJ)
520.      C  100  CONTINUE
521.      C   RETURN
522.      C   END
523.      CCCCCCCCC
524.      C
525.      C   DENSTY(CODE(MM,I,J):- COMPUTES THE DENSITY OF HYPERCUBE
526.      C   GENERATED BY GHM ENCODED TUPLE STORED IN CODE WHICH INCLUDES
527.      C   POINTS POINTED TO BY POINTERS IN ORDER STARTING FROM I TO J.
528.      C
529.      CCCCCCCCC
530.      C   FUNCTION DENSTY(CODE,MM,N,I,J)
531.      C   INTEGER CODE(12)
532.      C   NUMPTS = J-1 +1
533.      C   VOL = 1
534.      C   M1 = MM + 1
535.      C   DO 100 K = M1 ,N
536.      C     VOL = VOL*(CODE(K+N-MM)-CODE(K)+1)
537.      C  100  CONTINUE
538.      C   DENSTY = NUMPTS / VOL
539.      C   RETURN
540.      C   END

```

YANG, CHANG AND SINGH

```

541.          L = L + ITERM
542.          I1 = I1 + 1
543.          L1 = L - 1
544.          IF (I1 .GT. L1) GO TO 20
545.          DO 500 I2 = I1, L1
546.          TABLE(I2, COL) = TABLE(I1, COL)
547.          CONTINUE
548.          500  IF (I .GT. N1) GO TO 400
549.          CALL FILL(TABLE, COL, N, I1, L)
550.          400  CONTINUE
551.          300  CONTINUE
552.          K = NUM*K
553.          NUM = NUM - 1
554.          COL = COL + 1
555.          IF (NUM .GE. 2) GO TO 10
556.          RETURN
557.          END
558.          CCCCCCCCCC
559.          C
560.          C   FILL(TABLE, COL, N, I1, L):- TO STORE SOME ELEMENTS IN LTH ROW OF
561.          C   ARRAY TABLE.
562.          C
563.          C   CCCCCCCCCC
564.          C   SUBROUTINE FILL(TABLE, COL, N, I1, L)
565.          C   INTEGER TABLE(720,6)
566.          C   INTEGER COL, COL1
567.          C   COL1 = COL + 1
568.          C   IX = TABLE(I1, COL)
569.          C   DO 100 I = COL1, N
570.          C   TABLE(L, I-1) = TABLE(I1, I)
571.          C   TABLE(L, N) = IX
572.          C   RETURN
573.          C   END

```

APPENDIX B

```

1.  /*JDHPARA T=(2.0).L=4
2.  // EXEC FORTGCLG
3.  //FORT.SYSIN DD *
4.  CCCCCCCCCC
5.  C
6.  C      MAIN PROGRAM TO OPTIMIZE GH ENCODING FOR A POINTSET.
7.  C      SUCH THAT EACH HYPERCUBE HAS DENSITY NO LESS THAN A GIVEN
8.  C      THRESHOLD DENSITY.
9.  C      PTSET:      ARRAY CONTAINING POINTSET.
10. C      THRESH:     THRESHOLD DENSITY.
11. C      M:          NO.OF POINTS IN POINT SET.
12. C      N:          DIMENSIONAL SPACE.
13. C      FLG:       FLAG USED TO TURN OFF C ON PRINTING OF INTERMEDIATE
14. C               RESULTS.
15. C      CALLS :-
16. C      (1) GENPTS: TO GENERATE M POINTS IN N DIMENSIONAL SPACE C
17. C                STORE THEM IN ARRAY PTSET.
18. C      (2) OPTMUM: TO GENERATE OPTIMAL GH ENCODED TUPLES. AND
19. C                RETURNS THEIR NUMBER IN NCODE. IF FLG IS ON ,PRINTS
20. C                THOSE GH ENCODED TUPLES ALSO. THERSH IS THE THRESHOLD
21. C                DENSITY SUCH THAT EACH HYPERCUBE HAS DENSITY NO LESS
22. C                THAN THRESH.
23. C
24. CCCCCCCCCC
25. C      INTEGER PTSET(110.6),TABLE(11)
26. C      INTEGER FLG
27. C      N = 3
28. C      FLG = 1
29. C      DJ 300 M = 10.100.10
30. C      CALL GENPTS(PTSET,M,N)
31. C      DJ 400 ITH = 10.110.10
32.1 C      NJEN = ITH - 10
32.2 C      THRESH = NDEN/100.0
32.3 C      IF (FLG .EQ. 1) GO TO 10
33. C      PRINT 1
34. C
35. C      1      FJRMAT('1'.10X, 'INITIAL POINT SET IS:')
36. C      DJ 100 I = 1.M
37. C      PRINT 2 ,(PTSET(I,J),J=1.N)
38. C      100   CUNTINUE
39. C      2      FJRMAT('0'.10X,('0.613.0'))
40. C      PRINT 3
41. C      3      FJRMAT(////.100('*'))
42. C      PRINT 4
43. C      4      FJRMAT(//.25X,'HEURISTIC APPROACH')
44. C      PRINT 5
45. C      5      FJRMAT(25X,'-----')
46. C      10     CALL OPTMUM(PTSET,M,N,THRESH,NCODE,FLG)
47. C      400   TABLE(ITH/10) = NCODE
48. C      PRINT 6,N
49. C      6      FJRMAT(//.10X,'DIMENSION = ',15)
50. C      PRINT 7,M
51. C      7      FJRMAT(10X,'NO. OF POINTS IN SET= ',15)
52. C      PRINT 8
53. C      8      FJRMAT('0'.10X,'THRESHOLD ',10X,'% OF CODES GENERATED')
54. C      DJ 200 I = 1.11
55. C      ITH = I * 10 - 10
56. C      PRINT 9, ITH, TABLE(I)
57. C
58. C      200   CUNTINUE
59. C      300   CUNTINUE
60. C      9      FJRMAT(10X.110.10X.110)

```

YANG, CHANG AND SINGH

```

        STOP
        END
CCCCCCCCC
C
C      RANDOM(X): GENERATE A RANDOM NUMBER X,  $0 \leq X \leq 1.0$ , WITH
C      UNIFORM DISTRIBUTION.
C
CCCCCCCCC
SUBROUTINE RANDOM(X)
REAL X
INTEGER A/19727/, MULT/25211/, BASE/32768/
A = MOD(MULT*A,BASE)
X = FLOAT(A)/FLOAT(BASE)
RETURN
END
CCCCCCCCC
C
C      GENPTS(PTSET,M,N):- TO GENERATE TEST DATA SET IN ARRAY
C      PTSET, WHICH CONTAINS M NUMBERS IN N DIMENSIONAL
C      SPACE.
C
C      CALLS:
C      (1) RANDOM: TO GET A RANDOM NUMBER.
C
CCCCCCCCC
SUBROUTINE GENPTS(PTSET, M,N)
INTEGER PTSET(110,6)
DO 100 J = 1, N
  DO 100 I = 1, M
    CALL RANDOM(U)
    PTSET(I,J) = INT(9 * U) + 1
100 CONTINUE
RETURN
END
CCCCCCCCC
C
C      OPTMUM(PTSET,M,N,THRESH,NCODE,FLG):-
C      TO GENERATE OPTIMAL GH ENCODING FOR A GIVEN THRESHOLD
C      DENSITY, THRESH; AND RETURNS THE NUMBER OF SUCH TUPLES
C      IN NCODE.
C
C      CALLS:-
C      (1) PRDCE: TO COMPUTE THE MEASURE OF PRIORITY P(I) FOR
C      EACH COLUMN I.
C      (2) SORT: TO GET THE PROPER SET OF HANDLES CORRESPONDING
C      TO BEST VALUES OF THEIR PRIORITIES, P(I).
C      (3) SORT1: TO SORT POINTSET IN PTSET A.R.T. NEW SET OF
C      HANDLES.
C      (4) ENCODE: TO GENERATE GHM ENCODED TUPLES WITH DENSITY
C      NO LESS THAN THRESHOLD DENSITY.
C
CCCCCCCCC
SUBROUTINE OPTMUM(PTSET,M,N,THRESH,NCODE,FLG)
INTEGER PTSET(110,6),P(6),O(6),ORDER(110)
INTEGER FLG
NCODE = 0
DO 100 I = 1,M
  ORDER(I) = I
100 NPTS = M
  MM = 1
110 JFL = 0
  DO 500 I = 1,N

```

NRL REPORT 8382

```

121.      500      O(I) = 1
122.      IF (MM .GT. 1) GOTO 80
123.      CALL ENCODE(MM,N,PTSET,ORDER,O,NPTS,IFL,NUM,THRESH,FLG)
124.      IF (NUM .EQ. 0) GOTO 70
125.      NCODE = NUM
126.      RETURN
127.      70      MM = MM + 1
128.      80      CALL PROCES(PTSET,ORDER,P,NPTS,N)
129.      IF (FLG .EQ. 1) GO TO 30
130.      PRINT 1,(P(I),I=1,N)
131.      1      FORMAT(//,10X,'PRINTOUT OF P ARRAY IS:',.6I5)
132.      30      CALL SORT(P,O,N)
133.      IF (FLG .EQ. 1) GO TO 40
134.      PRINT 2,(O(I),I=1,N)
135.      2      FORMAT(//,10X,'ORDER OF VARIOUS HANLES IS:',.6I5)
136.      40      CALL SURTPT(PTSET,O,ORDER,NPTS,N,M)
137.      CALL ENCODE(MM,N,PTSET,ORDER,O,NPTS,IFL,NUM,THRESH,FLG)
138.      NP = 0
139.      DO 200 I = 1,NPTS
140.      IF(ORDER(I) .EQ. 0) GO TO 200
141.      NP = NP + 1
142.      ORDER(NP) = ORDER(I)
143.      200     CONTINUE
144.      NPTS = NP
145.      NCODE = NCODE + NUM
146.      IF (NPTS .EQ. 1) GO TO 20
147.      IF (NPTS .LE.0) RETURN
148.      IF (NUM .LT.1 .AND. IFL .EQ. 1) MM = MM + 1
149.      IF (NUM .LT. 1 .AND. IFL .EQ. 0) GO TO 20
150.      GO TO 110
151.      20      IF (FLG .EQ. 1) GO TO 50
152.      PRINT 5, MM
153.      5      FORMAT (//,10X,'HANDLE LENGTH = ',I5)
154.      PRINT 6 ,(O(I),I= 1,MM)
155.      6      FURMAT(/,10X,'HANDLES ARE :',.6I5)
156.      PRINT 7
157.      7      FORMAT(/,10X,' CORRESPONDING GH CODES ARE:')
158.      50     NCODE = NCODE + NPTS
159.      IF (FLG .EQ. 1) RETURN
160.      DO 300 I = 1,NPTS
161.      II = ORDER(I)
162.      DO 333 J=1,N
163.      JJ = O(J)
164.      333     P(J) = PTSET(II,JJ)
165.      PRINT 8,(P(K),K=1,MM)
166.      8      FORMAT(10X,'('',.6I5,'')')
167.      IF (N-MM .EQ.0) GO TO 300
168.      NI = MM+1
169.      PRINT 9,(P(K),K=NI,N)
170.      9      FORMAT('',.42X,';',.5I5)
171.      PRINT 10,(P(K),K=NI,N)
172.      10     FORMAT('',.65X,';',.5I5)
173.      PRINT 11
174.      11     FORMAT('',.85X,'')
175.      300     CONTINUE
176.      RETURN
177.      END
178.      CCCCCCCCC
179.      C
180.      C PROCES(PTSET,ORDER,NPTS,N):- COMPUTES THE MEASURE OF PRICTY

```

YANG, CHANG AND SINGH

```

181. C          ,P(I) FOR EACH COLUMN I WHERE I=1.....N. FOR A GIVEN
182. C          POINT SET IN ARRAY PTSET CONTAINING NPTS NO. OF POINTS
183. C          IN N DIMENSIONAL SPACE.
184. C          CALLS:
185. C          (1) HPODER: TO COMPUTE P(I) FOR ONE PARTICULAR COLUMN I.
186. C
187. CCCCCCCCCC
188. C          SUBROUTINE PROCES(PTSET,ORDER,P,NPTS,N)
189. C          INTEGER PTSET(110,6),ORDER(110),TEMPR(110),P(6)
190. C          DO 100 I = 1,N
191. C          DO 200 J = 1,NPTS
192. C             JJ = ORDER(J)
193. C          200 TEMPR(J) = PTSET(JJ,I)
194. C             CALL HPODER(I,TEMPR,NPTS,P)
195. C          100 CONTINUE
196. C             RETURN
197. C             END
198. CCCCCCCCCC
199. C
200. C          HPODER(I,TEMPR,M,P):- COMPUTES P(I) FOR A COLUMN VECTOR TEMPR
201. C          .IT USES HEAP STRUCTURE FOR THAT AND HENCE,
202. C          CALLS:
203. C          (1) REHEAP: TO MAKE HEAP OF ELEMENTS OF COLUMN VECTOR TEMPR.
204. C
205. CCCCCCCCCC
206. C          SUBROUTINE HPODER(I,TEMPR,M,P)
207. C          INTEGER TEMPR(110),P(6)
208. C          INTEGER TEMP,ELMENT,PSUM,SUM
209. C          M1 = M
210. C          L = M/2 + 1
211. C          L = L-1
212. C          10 CALL REHEAP(L,M1,TEMPR)
213. C             IF (L .GT. 1) GOTO 10
214. C             ELMENT = 0
215. C             PSUM = 0
216. C             SUM = 0
217. C          20 TEMP = TEMPR(1)
218. C             TEMPR(1) = TEMPR(M1)
219. C             IF (ELMENT .EQ. TEMP)GO TO 30
220. C             IF (SUM .GT. 1) PSUM = PSUM + SUM * SUM
221. C             ELMENT = TEMP
222. C             SUM = 1
223. C             DO TO 40
224. C             SUM = SUM + 1
225. C             M1 = M1 - 1
226. C             IF (M1 .GT. 0) GO TO 50
227. C             IF (SUM .GT. 1) PSUM = PSUM + SUM * SUM
228. C             P(I) = PSUM
229. C             RETURN
230. C          50 CALL REHEAP(1,M1,TEMPR)
231. C             GOTO 20
232. C             END
233. CCCCCCCCCC
234. C          REHEAP(L,M1,TEMPR):- MAKES A HEAP OF THE ELEMENTS OF ARRAY TEMPR
235. C          STARTING FROM TEMPR(L) TO TEMPR(M1).
236. C
237. CCCCCCCCCC
238. C          SUBROUTINE REHEAP(L,M1,TEMPR)
239. C          INTEGER TEMPR(110)
240. C          INTEGER FLAG , X

```

NRL REPORT 8382

```

241.      II = L
242.      JJ = 2*II
243.      X = TEMPR(II)
244.      FLAG = 1
245.      IF (JJ .GT. M1) RETURN
246. 40     IF (JJ .GE. M1) GOTO 10
247.      IF (TEMPR(JJ) .LT. TEMPR(JJ+1)) JJ = JJ+1
248. 10     IF (X .GE. TEMPR(JJ)) GOTO 20
249.      TEMPR(II) = TEMPR(JJ)
250.      II = JJ
251.      JJ = 2*II
252.      TEMPR(II) = X
253.      GOTO 30
254. 20     FLAG = 0
255. 30     IF (JJ .LE. M1 .AND. FLAG .EQ. 1) GOTO 40
256.      RETURN
257.      ENJ
258.      CCCCCCCCCC
259.      C
260.      C SORT(P,O,N):- SORTS P ARRAY CONTAINING N ELEMENTS AND DURING
261.      C THIS SORT ALSO MOVES THE ELEMENTS OF ARRAY O IN THE SAME FASH-
262.      C ION SO THAT INORDER TO CHOOSE A HANDLE SET OF M LENGTH, WE CAN
263.      C CHOOSE FIRST M-1 HANDLES STORED IN FIRST M-1 POSITION OF O.
264.      C
265.      CCCCCCCCCC
266.      C SUBROUTINE SORT(P,O,N)
267.      C INTEGER O(6),P(6)
268.      C INTEGER FLAG, TOP, TEMP, BOTTOM
269.      C FLAG = 1
270.      C BOTTOM = N
271.      C TOP = 1
272. 10     IF (FLAG .NE. 0 .AND. TOP .LT. BOTTOM) GO TO 20
273.      RETURN
274. 20     FLAG = 0
275.      II = BOTTOM - 1
276.      DO 100 I = 1,II
277.      IF (P(I) .GE. P(I+1)) GO TO 100
278.      X = P(I)
279.      P(I) = P(I+1)
280.      P(I+1) = X
281.      TEMP = O(I)
282.      O(I) = O(I+1)
283.      O(I+1) = TEMP
284.      FLAG = 1
285. 100    CONTINUE
286.      BOTTOM = BOTTOM - 1
287.      GO TO 10
288.      END
289.      CCCCCCCCCC
290.      C
291.      C SDRTPT(P,SET,O,ORDER,M,N,M2):- TO SORT POINT IN ARRAY PTSET
292.      C W.R.T. NEW HANDLE VECTOR IN O. ORDER CONTAINS POINTERS TO
293.      C POINTS IN PTSET WHICH HAVE NOT BEEN DELETED YET. M IS THE
294.      C NUMBER OF SUCH POINTS WHILE THERE WERE M2 POINTS IN THE
295.      C ORIGINAL POINT SET. N IS THE DIMENSION OF SPACE.
296.      C
297.      C HEAPSORT IS USED HERE, THEREFORE:
298.      C CALLS:
299.      C (1) REHIP: TO MAKE HEAP.
300.      C (2) COMPAR: TO COMPARE ORDER OF TWO POINTS.

```

YANG, CHANG AND SINGH

```

301. C      (3) EXCHANG: TO EXCHANG POSITION OF TWO PCINTS IN PTSET.
302. C
303. CCCCCCCCC
304. SUBROUTINE SORTPT(PTSET,O,ORDER,M,N,M2)
305. INTEGER PTSET(110,6),O(6),ORDER(110)
306. M1 = M
307. L = M1/2 + 1
308. 10  L = L-1
309. CALL REHIP(L,M1,M2,N,O,PTSET,ORDER)
310. IF(L .GT. 1) GO TO 10
311. 20  CALL EXCHAN(ORDER(1),ORDER(M1),N,PTSET)
312. M1 = M1-1
313. CALL REHIP(1,M1,M2,N,O,PTSET,ORDER)
314. IF(M1 .GT. 1) GO TO 20
315. RETURN
316. END
317. CCCCCCCCC
318. C
319. C REHIP(L,M1,M2,N,O,PTSET,ORDER):- MAKES A HEAP STRUCTURE
320. C OF THOSE POINTS IN PTSET WHOSE POINTER ARE IN ARRAY ORDER.
321. C STARTING FROM ORDER(L) TO ORDER(M1).
322. C M2 IS THE NUMBER OF POINTS IN ORIGINAL PCINTSET
323. C
324. CCCCCCCCC
325. SUBROUTINE REHIP(L,M1,M2,N,O,PTSET,ORDER)
326. INTEGER PTSET(110,6),O(6),ORDER(110)
327. INTEGER FLAG ,X
328. II = L
329. JJ = 2*II
330. I = ORDER(II)
331. X = M2 + 1
332. DO 5 LL=1,N
333. 5  PTSET(X,LL) = PTSET(I,LL)
334. FLAG = 1
335. IF(JJ .GT.M1) RETURN
336. 40  IF(JJ .GE. M1) GO TO 10
337. CALL COMPAR(ORDER(JJ),ORDER(JJ+1),N,O,PTSET,ICOMP)
338. IF (ICOMP .LT.0) JJ = JJ+1
339. 10  CALL COMPAR(X,ORDER(JJ),N,O,PTSET,ICOMP)
340. IF (ICOMP .GE.0) GO TO 20
341. I = ORDER(II)
342. J = ORDER(JJ)
343. DO 100 LL = 1,N
344. 100 PTSET(I,LL) = PTSET(J,LL)
345. II = JJ
346. JJ = 2*II
347. I = ORDER(II)
348. DO 200 LL = 1,N
349. 200 PTSET(I,LL) = PTSET(X,LL)
350. GO TO 30
351. 20  FLAG = 0
352. 30  IF (JJ .LE. M1 .AND. FLAG .EQ.1) GO TO 40
353. RETURN
354. END
355. CCCCCCCCC
356. C
357. C COMPAR(I,J,N,O,PTSET,ICOMP):- COMPARES TWO POINTS I & J W.R.T. A
358. C HANDLE VECTOR IN O, AND RETURNS IN ICOMP:
359. C 1 IF ITH POINT > JTH PCINT.
360. C J IF ITH POINT = JTH POINT.

```

NRL REPORT 8382

```

361. C      -1 IF ITH POINT < JTH POINT.
362. C
363. CCCCCCCCC
364. SUBROUTINE COMPAR(I,J,N,O,PTSET,ICOMP)
365. INTEGER PTSET(110,6),O(6)
366. II = 1
367. 100  III = O(III)
368.      IF (PTSET(I,III) .NE. PTSET(J,III)) GO TO 10
369.      IF (II .LT. N) GO TO 5
370.      ICOMP = 0
371.      RETURN
372. 5    II = II+1
373.      GO TO 100
374. 10  IF(PTSET(I,III) .LT. PTSET(J,III)) ICOMP = -1
375.      IF (PTSET(I,III).GT.PTSET(J,III)) ICOMP = 1
376.      RETURN
377.      END
378. CCCCCCCCC
379. C
380. C      EXCHAN(I,J,N,PTSET):- SWITCHES THE POSITIONS OF TWO POINTS I & J
381. C      IN ARRAY PTSET WHICH CONTAINS A POINT SET IN N DIMENSIONAL
382. C      SPACE.
383. C
384. CCCCCCCCC
385. SUBROUTINE EXCHAN(I,J,N,PTSET)
386. INTEGER PTSET(110,6)
387. DO 100 II = 1,N
388.   ITEMP = PTSET(I,II)
389.   PTSET(I,II) = PTSET(J,II)
390.   PTSET(J,II) = ITEMP
391. 100 CONTINUE
392. RETURN
393. END
394. CCCCCCCCC
395. C
396. C      ENCODE(M,N,PTSET,ORDER,D,NPTS,IFL,NUM,THRESH,FLG):- GENERATE
397. C      GHM CODES FOR A POINT SET PTSET, CONTAINING NPTS POINTS IN
398. C      N DIMENSIONAL SPACE, ONLY THOSE HYPERCUBES WHICH HAVE DENSITY
399. C      => THAN THRESHOLD DENSITY, THRESH, ARE INCLUDED IN OPTIMAL GH
400. C      ENCODING. FOR A GIVEN M, NUM CONTAINS THE NUMBER OF SUCH
401. C      HYPERCUBE GENERATED W.R.T. A GIVEN HANDLE VECTOR IN O.
402. C      ORDER ARRAY CONTAINS POINTERS TO POINTS IN PTSET WHICH ARE
403. C      STILL IN POINTSET. FLG IS A FLAG USED TO TURN OFF & ON
404. C      PRINTING OF OPTIMAL GH ENCODED TUPLES.
405. C      CALLS:
406. C      (1) MINMAX: TO FIND THE MINIMUM & MAXIMUM ELEMENT IN A
407. C      PARTICULAR COLUMN FOR SOME POINTS CONSIDERED TO BE IN
408. C      IN THE CURRENT HYPERCUBE.
409. C      (2) DENSITY: TO COMPUTE THE DENSITY OF A GENERATED HYPERCUBE.
410. C
411. CCCCCCCCC
412. SUBROUTINE ENCODE(M,N,PTSET,ORDER,D,NPTS,IFL,NUM,THRESH,FLG)
413. INTEGER FLG,FLAG
414. INTEGER PTSET(110,6),O(6),CODE(12),ORDER(110)
415. MM = M - 1
416. IF (FLG .EQ. 1) GO TO 70
417. PRINT I
418. 1  FORMAT(//)
419. PRINT S,M
420. 5  FORMAT(11X,'GH',12,' CODES ARE:')

```

YANG, CHANG AND SINGH

```

421.          PRINT 6,(O(I),I=1,MM)
422.          E          FORMAT('0',10X,'HANDLES ARE:',6I5)
423.          70         IF(MM .EQ. N) GO TO 20
424.          IF (MM .EQ. 0) GO TO 10
425.          NUM = 0
426.          IFL = 0
427.          MI = NPTS + 1
428.          I = 1
429.          J = 1
430.          DO 100 K = 2,MI
431.             IF (K .GT. NPTS) GO TO 40
432.             FLAG = 0
433.             I1 = ORDER(K)
434.             I2 = ORDER(K-1)
435.             DO 200 J1 = 1,MM
436.                JJ = O(J1)
437.                IF (PTSET(I1,JJ) .EQ. PTSET(I2,JJ)) GO TO 200
438.                FLAG = 1
439.                GO TO 330
440.          200         CONTINUE
441.          330         IF (FLAG .EQ. 1) GO TO 40
442.                J = J + 1
443.                GO TO 100
444.          40          I1 = ORDER(I)
445.                DO 300 J1 = 1,MM
446.                   JJ = O(J1)
447.                   CODE(J1) = PTSET(I1,JJ)
448.          300         CONTINUE
449.                NI = MM+1
450.                L = MM
451.                NN = N - MM
452.                DO 400 J1 = NI,N
453.                   JJ = O(J1)
454.                   CALL MINMAX(PTSET,ORDER,JJ,I,J,MIN,MAX)
455.                   L = L + 1
456.                   CODE(L) = MIN
457.          400         CODE(L+NN) = MAX
458.                DEN = DENSTY(CODE,MM,N,I,J)
459.                IF (DEN .GE. THRESH) GO TO 111
460.                IFL = 1
461.                GO TO 555
462.          111         IF (I .EQ. J) GO TO 555
463.                N2 = L
464.                N3 = L + NN
465.                DO 333 IK = I,J
466.                   ORDER(IK) = 0
467.                   IF ( FLG .EQ. 1) GO TO 72
468.                   PRINT 32,(CODE(I),I = 1,MM)
469.                   32          FORMAT('0',10X,'(',6I3)
470.                   PRINT 33,(CODE(I),I=NI,N2)
471.                   33          FORMAT('+',30X,',';6I3)
472.                   N22 = N2+1
473.                   PRINT 34,(CODE(I),I=N22,N3)
474.                   34          FORMAT('+',50X,',';6I3)
475.                   PRINT 35
476.                   35          FORMAT('+',60X,')')
477.                   72          NUM = NUM + 1
478.                   I = K
479.                   J = K
480.          100         CONTINUE

```

NRL REPORT 8382

```

481. CCCCCCCCCC
482. C
483. C
484. SUBROUTINE EXHAUST(PTSET,M,N,NCODE,IFL)
485. INTEGER PTSET(110,6),O(6),TABLE(720,6),NCODE(5,3)
486. CALL PRCOMB(TABLE,N)
487. ITERM = 1
488. DO 100 I = 1,N
489. ITERM = ITERM*I
490. DO 200 J = 1,ITERM
491. DO 300 K = 1,N
492. O(I,J) = TABLE(I,J)
493. IF(IFL .EQ. 1) GO TO 88
494. PRINT 1
495. 1 FORMAT(//,11X,'HANDLES CHOSEN ARE:')
496. PRINT 2,(O(I,J),J=1,N)
497. 2 FORMAT('0',10X,4I5)
498. 88 CALL SORTPT(PTSET,O,M,N)
499. IF(IFL .EQ. 1) GO TO 89
500. PRINT 3
501. 3 FORMAT(//,11X,'FOLLOWING ARE GH CODES W.R.T. ABOVE '
502. 4 'HANDLES:')
503. 89 N1 = N + 1
504. DO 500 J = 1,N1
505. CALL ENCODE(J,M,N,O,PTSET,NUM,IFL)
506. IF(J .LE. 1 .OR. J .GE. N1) GO TO 500
507. IF(NUM .LT. NCODE(J-1,2)) NCODE(J-1,2) = NUM
508. IF(NUM .GT. NCODE(J-1,3)) NCODE(J-1,3) = NUM
509. 500 CONTINUE
510. IF( IFL .EQ. 1) GO TO 200
511. PRINT 4
512. 4 FORMAT(//,100(' '))
513. 200 CONTINUE
514. RETURN
515. END
516. CCCCCCCCCC
517. C
518. C PRCOMB(TABLE,N): TO FIND ALL POSSIBLE SET OF HANDLES, USING
519. C PERMUTATION COMBINATION.EACH ROW OF TABLE CONTAINS ON SET
520. C HANDLES.N IS THE DIMENSION OF SPACE.
521. C CALLS:
522. C (1) FILL: TO STORE PROPER COLUMN NUMBERES IN TABLE.
523. C
524. CCCCCCCCCC
525. SUBROUTINE PRCOMB(TABLE,N)
526. INTEGER TABLE(720,6)
527. INTEGER COL
528. NUM = N
529. DO 100 I = 1,N
530. TABLE(I,I) = I,
531. K = 1
532. COL = 1
533. 10 L = I
534. N1 = NUM - 1
535. ITERM = 1
536. DO 200 J = 1,N1
537. ITERM = ITERM*I
538. DO 300 K1 = 1,K
539. DO 400 I = 1,NUM
540. I1 = L

```