CHAPTER - 3

CHAPTER-3

ANALYSIS OF SIMPLE JOINTED PLANAR KINEMATIC CHAINS : PART-I

Introduction

The need for reliable and efficient algebraic test for isomorphism is always felt particularly when ever structural synthesis procedure is to be implemented on computer. Most of the available methods have drawbacks. A complete survey of the reported method is presented in the literature review. Most of these demerits are overcome by Hamming code approach, which gives a unique string for chains as well as for identifying inversions. A computerized methodology based on Hamming number technique [25] is formulated and the same is implemented to check isomorphism among kinematic chains and inversions up to 13-link, 2-d.o.f. A computerized methodology to identify best inversion for path and function generation is also formulated based on [36,37].

Literature Review

We now consider the traditional methods of detecting ismorphism and their weaknesses admitting that, in the general case, no efficient solution of the graph isomorphism problem has yet been found.

The bodies in a Kinematic chain may be relabeled through an orthogonal transformation of the adjacency matrix. Woo [27] uses an algorithm based on this approach to identify isomorphisms in a list of contracted mappings. As the number of bodies n in the Kinematic chain increases the number of possible orthogonal transformations of the adjacency matrix increases as n-factorial visual inspection of Kinematic chains was the only available means for detecting isomorphism's before Woo's algorithm, Visual inspection is not always simple; often diagrams of identical Kinematic chains can, at first sight, appear to be different. Woo's algorithm compares orthogonally transformed adjacency matrices with the matrices that are already been proved to represent unique forms. A form that cannot be transformed to produced a matrix already in the list of unique forms is itself unique Since there are many possible orthogonal transformations their number increases rapidly as the number of bodies increases, it is evident that Woo's algorithm quickly becomes more and more laborious, despite some shortcuts which may be taken, such as swapping only those rows and columns that represent bodies of the same degree

More recently, Tang and Liu [31] improve on Woo's method. They encode the adjacency matrix to produce a degree-code and then exhaustively permute the adjacency matrix, swapping the number labels on bodies with the same degree, then nothing the permutation with the maximum degree code. As Tang and Liu point out, this method works well for graphs of epicyclic gear trains and for other mechanisms with few bodies. However, by way of example, there are 614!(= 17280) permutations for each of the 50 planar ten-bar with mobility one that consist of six ternary links and four binary links. In this case the computational overhead associated with the number of permutations is significant.

Some properties of an adjacency matrix are preserved under an orthogonal transformation, including its determinant and its characteristic polynomial. Uicker and Raicu [32] suggested that the characteristic polynomial could be used to test for isomorphism. However, if two Kinematic chains are isomorphic, it is necessary, but not sufficient, that their characteristic polynomials are identical [30,32]. This is because both similarity and congruence transformations leave the characteristic polynomial invariant but do not preserve the kinematic structure represented by the original adjacency matrix. When it was proposed that isomorphism's might be detected using their characteristic polynomials, graph theorists noted the existence of counter-examples, although more of these had meaningful representations as kinematic chains. Nevertheless, several attempts at isomorphic-detection were made using characteristic polynomials. Moreover, at that time the uniqueness of the characteristic polynomial was thought to guarantee the uniqueness of closed, proper Kinematic chains. However, more recently, Mruthynjaya and Balsubramanian [14] report an example of two distinct Kinematic chains that fail this test. There is two such pair' amounts the no planar, mobility one, ten-bars In addition, Fig -5 indicates two pairs of distinct b=3 M=3, ten-bars with identical characteristic polynomials.

Mruthyunjaya and Raghvan [34] recognized that this method of identifying isomorphisms was faulty, and they put forward an alternative method that uses the powers of the adjacency matrix.

1	6	1	1	5	5	1	1	1	1
6	1	1	1	1	1	5	5	1	1
1	1	1	6	5	5	1	1	1	1
1	1	1	6	5	5	1	1	1	1
5	1	5	1	1	1	1	1	1	1
5	1	5	1	1	1	1	1	1	1
1	5	1	5	1	1	1	1	1	1
1	5	1	1	1	1	1	1	1	4
1	1	1	5	1	1	1	1	1	4
1	1	1	1	1	1	1	4	4	1

(a)

1	6	6	1	5	1	1	1	1	1
6	1	1	1	1	5	5	1	1	1
6	1	1	1	1	5	1	5	1	1
1	1	1	1	5	1	1	1	5	5
5	1	1	5	1	1	1	1	1	1
1	5	5	1	1	1	1	1	1	1
1	5	1	1	1	1	1	1	4	1
1	1	5	1	1	1	1	1	1	4
1	1	1	5	1	1	4	1	1	1
1	1	1	5	1	1	1	4	1	1

(b)

1	6	6	1	5	1	1	1	1	1
6	1	1	6	1	5	1	1	1	1
6	1	1	6	1	1	5	1	1	1
1	6	6	1	1	1	1	5	1	1
5	1	1	1	1	1	1	1	4	1
1	5	1	1	1	1	1	1	1	4
1	1	5	1	1	1	1	1	1	4
1	1	1	5	1	1	1	1	4	1
1	1	1	1	4	1	1	4	1	1
1	1	1	1	1	4	4	1	1	1

(c)

1	1	6	6	6	6	1	1	1	1
1	1	1	1	1	1	6	6	6	6
6	1	1	1	1	1	4	1	1	1
6	1	1	1	1	1	1	4	1	1
6	1	1	1	1	1	1	1	4	1
6	1	1	1	1	1	1	1	1	4
1	6	4	1	1	1	1	1	1	1
1	6	1	4	1	1	1	1	1	1
1	6	1	1	4	1	1	1	1	1
1	6	1	1	1	4	1	1	1	1

d)

to provide information about the size and number of loops to which a given body belongs The proposed that, by matching bodies based on these properties, *isomorphs could be detected.*

Mruthyunjaya and Balasubramaniam [33] formed a new matrix they called the degree matrix by setting

$\alpha_{ij} = g_i + g_j$ if body i and j are directly connected.

= 1, otherwise

Isomorphs were detected by comparing characteristic polynomials of the degree matrix, fig.-5 shows four Kinematic chains and their degree matrices. However, it is not yet established whether this is a general property and whether it can always be used safely. Whilst Mruthyunjaya and Balasubramanian noted the existence of counter-examples none of the counter-examples represented closed, proper Kinematic chains.

Rao and Varda Raju [25] present a method for detecting isomorphisms based on Hamming numbers of the adjacency matrix. They successfully distinguish between all the b = 3 Kinematic chains with six, eight and ten-bars with mobility. M=1 and ten-bars with mobility M=3. Although no counter-examples are known, when the algorithm was applied to the detection of isomorphisms among the number of inversions of the planar, mobility M=1, ten-bars,

The problem of finding solutions to the more general graph *isomorphism problem* has been debated in computer science circles. Kobler et al. [29] have examined the structural complexity of the graph isomorphism problem and state that there is strong evidence to suggest that no efficiency algorithms exist for the problem (although there are few special cases for which efficient graph isomorphism algorithm exist). This implies that any method of isomorphic detection, which is not exhaustive in its approach, can, at best, detect isomorphic graphs with only a high *probability of being correct.*

Definitions and Terminology's

Various definitions and their abbreviations are given below, which are to be clearly understood before applying the method

Consider a four bar chain (Fig 1); it can be represented by a matrix called connectivity or adjacency matrix (C), whose elements are either 0 or 1. An element C_{ij} of the matrix C assumes the following values-

C_{ij}=0, if links i and j are not directly connected,

C_{ij}=1, if link i and j are in direct contact,

C_{ii}=0, since no link can connect to itself.

Following the above, the connectivity matrix for four bar chain, (Fig.1) will be

$$C_{ij} = \begin{matrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{matrix}$$

It may be noted that the matrix C is square and symmetric and in general C is of size n x n for a chain of n links. The element of each row of C can be considered to constitute a binary code representing the concerned link, e.g., the binary code for link 1,2 3, and4 of the four bar chain are given below,

Link	binary code			
1	0	1	0	1
2	1	0	1	0
3	0	1	0	1
4	1	0	1	0

Hamming number

Hamming number h_{ij} related to links i and j is defined as number of digits at which code of link i and j differ For e.g., h_{12} =4, since code for link 1 and 2 differ at all the 4 digits and h_{11} =0 since codes are identical Following the above concept, a new matrix H- called the Hamming matrix – can be formed for every chain. The Hamming matrix is also square, symmetric and contains zero all along the leading diagonal. However, unlike the connectivity matrix it contains digits, which could be larger than unity.

For the chain of Fig.1

$$h_{ij} = \begin{matrix} 0 & 4 & 0 & 4 \\ 4 & 0 & 4 & 0 \\ 0 & 4 & 0 & 4 \\ 4 & 0 & 4 & 0 \end{matrix}$$

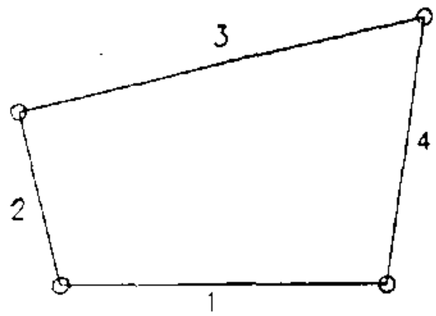
- Link Hamming number of any link i is the sum of all the elements in the I st row of Hamming matrix. Thus, the link Hamming number for link 1 is 8

Chain Hamming number for any chain is the sum of the entire link Hamming numbers of that chain. It also works out to be the sum of all the elements of the Hamming matrix for that chain. The chain Hamming number for the four-bar chain is 32 (=8+8+8+8)

Link Hamming string for any link i is the string obtained by concatenating (a) the link Hamming number of i with (b) the frequency of occurrence, of all the integer elements of I st row). For example for link 1, the Hamming string is 8,20002, implying that the link Hamming number is 8 which comprises of two 4s, no 3s, n0 2s, no 1s and 2 0s.

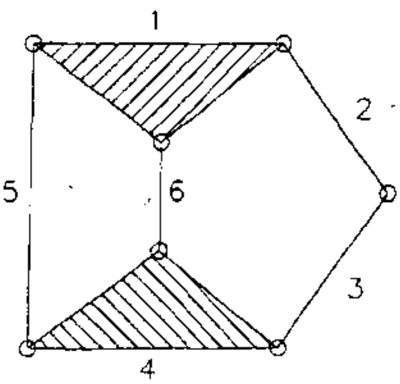
Chain Hamming string is defined as concatenation of (1) the chain Hamming number and (2) the link Hamming strings placed in descending order of magnitude. For the four bar chain it is written as 32, -8,20002-8,20002-8,20002-8,20002. The dashes and commas in the above string have been placed to improve readability, but are not essential.

Chapter - 3



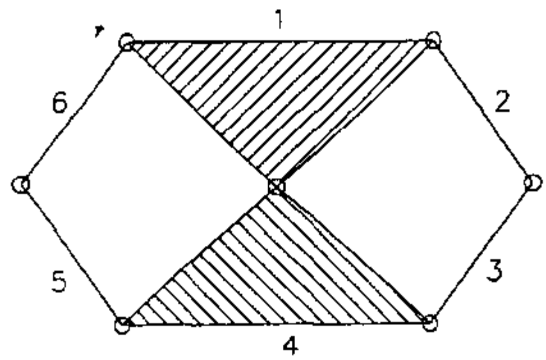
4-link,2-d.o.f.

Fig-1



6-link,1-d.o.f.

Fig-3



6-link,1-d.o.f.

Fig-2

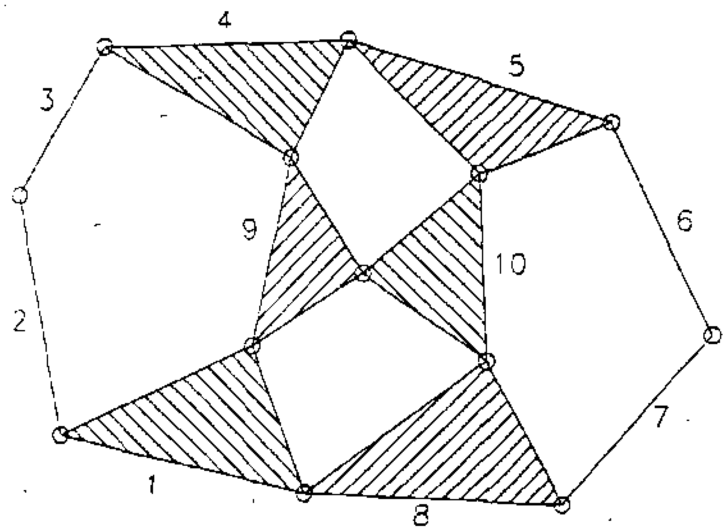


Fig-5

Method

Chain Hamming string is considered as definitive test for isomorphism [25,35] among chains. That is, two chains are said to be isomorphic if their chain Hamming strings is identical. Further to add strength to the methodology second and third order chain Hamming strings are compared wherever necessary

Illustration

Let us compare 6-link, 1-d.o.f Watt chain (Fig.2) and Stephenson chain (Fig.3) for isomorphism. Following the above procedure, the chain Hamming strings for Watt chain is 100,18-1200021,18-120001,16-0120111,16-0120111,16-0120111, 16-0120111. The chain Hamming strings for Stephenson chain (Fig.3) is

100,20,0301101,20,0301101,16,0111201,16,0111201,14,0200202,140200202.

Comparing their Hamming string it can be seen that they are different. Hence the two chains are non-isomorphic. Primary and Secondary Hamming strings discard all isomorphic chains up to 10-links, 1-d.o.f. For higher link chains tertiary Hamming strings are used as isomorphism index.

Inversions

Link neighboring string is considered as definitive test of isomorphism among inversions of simple jointed planar kinematic chains [25,35].

Illustration

Link neighboring strings are obtained for he four bar chain as shown below, Link 1 8, 20002, 8, 20002, 8, 20002

Link 2. 8, 20002, 8, 20002, 8, 20002.

Link 3: 8, 20002, 8, 20002, 8, 20002.

Link 4. 8, 20002, 8, 20002, 8, 20002.

Since all the above four strings are identical, the four inversions are all identical and hence there exists only one inversion for the four-bar

As a second example let us consider Watt chain (Fig.2). The link neighboring strings are,

Link 1:18, 1200021, 18, 1200021, 16, 0120111, 16, 0120111

Link 2: 18, 1200021, 18, 1200021, 16, 0120111,

Link 3: 18, 1200021, 18, 1200021, 16, 0120111,

Link 4: 18, 1200021, 18, 1200021, 16, 0120111, 16, 0120111

Link 5: 18, 1200021, 18, 1200021, 16, 0120111,

Link 6: 18, 1200021, 18, 1200021, 16, 0120111,

It may be seen that link 1 and 4 are identical as well as link 2,3,5,6 are identical. Hence two inversions are possible – one by fixing any one of links (1,4) and another by fixing any one of links (2,3,5,6).

As a third illustration let us consider again the ten-bar chain (Fig.-5). The link neighborhood string for each of the links is

Link 1: 40, 00003212101,	40, 00003212101,	38, 00003202201,	36, 0000333001
Link 2: 36, 0000333001,	40, 00003212101,	36, 0000333001	
Link 3: 36, 0000333001,	40, 00003212101,	36, 0000333001	
Link 4: 40, 00003212101,	40, 00003212101,	38, 00003202201,	36, 0000333001
Link 5: 40, 00003212101,	40, 00003212101,	38, 00003202201,	36, 0000333001
Link 6: 36, 0000333001,	40, 00003212101,	36, 0000333001	
Link 7: 36, 0000333001,	40, 00003212101,	36, 0000333001	
Link 8: 40, 00003212101,	40, 00003212101,	38, 00003202201,	36, 0000333001
Link 9: 38, 00003202201,	40, 00003212101,	40, 00003212101,	38, 00003202201
Link10. 38, 00003202201,	40, 00003212101,	40, 00003212101.	38, 00003202201

On comparing the chains the conclusion is that totally three inversions are possible – one by fixing any one of links (1,4,5,8) another by fixing any one of links (2,3,6,7) and the third by fixing either of links (9,10).

Best of the Inversions

Once the inversions are identified the obvious question that arises is which nversion is better for path generation and function generation respectively.

Assort all links in a chain in ascending order of link connectivity and link Hamming numbers.

1) for function generation arrange the highest connectivity link having lowest Hamming value followed by other links in the descending order[35,36].
2) For path generation arrange the lowest connectivity link having lowest Hamming value first followed by other links in the ascending order [35,36].

Illustration

As an example on best of function and path generation let us consider chain ig.1. The connectivity matrix of Fig.1 is as below,

0	1	0	0	1	1
1	0	1	0	0	0
0	1	0	0	0	0
0	0	1	0	1	1
1	0	0	1	0	0
1	0	0	1	0	0

When link 1 is considered to be fixed all the elements of the first row in the connectivity matrix are made 0 The modified matrix is called transmission matrix, which is as below,

0	0	0	0	0	0
1	0	1	0	0	0
0	1	0	1	0	0
0	0	1	0	1	1
1	0	0	1	0	0
1	0	0	1	0	0

The Hamming matrix is obtained from the modified connectivity matrix as is obtained from the connectivity matrix. The same is shown below,

0	2	2	3	2	2
2	0	4	3	2	2
2	4	0	5	2	2
3	3	5	0	5	5
2	2	2	5	0	0
2	2	2	5	0	0

The details of the Hamming matrix obtained above are summarized. Link 2 is a binary link having Link Hamming number 13, Link 3 is binary link having link Hamming number 5, Link 4 is ternary link having link Hamming number 21, Link 5 and 6 are binary link having link Hamming number 11 each.

For chain (Fig.1) having link 1 fixed the links are arranged in (i) better function generation (ii) better path generation as per method reported.

i) 4, (5,6),2,3 i.e. ternary link, binary link having lower Hamming number

ii) (5,6), 2,3,4 i.e. binary link having lower Hamming number followed by ternary link so on.

The above results confirm the result reported in [36,37] based on flow path The following explanation will support the above statement When one of the links of a chain is fixed, motion transfer (flow) from one link to other can not take place through the fixed link. Thus, the type and location of the fixed link in the chain influence the total number of flow paths of an inversion. The number of flow paths from one link to another in a chain are equal to the minimum number of joints between the links under consideration In the joints the path in which the fixed link falls should be avoided. It is easy to conceive that among chains with equal number of link and joints *the chains with lesser number of total flow paths are better.*

Computer program

The methodology uses matrix notation for mathematical representation of chains. Entire algorithm is based on manipulation of these matrices. Further, number 0 and 1 are particularly suited for manipulation and storage on a computer. These make the algorithm computationally efficient. The input to the software is kinematic chains in its representation form.

Applications

The computerized methodology formulated is implemented up to 12-links, 2-d.o.f chains to identify inversions and best inversions (for path and function generation). Results obtained are tabulated below.

Results

Count of inversions obtained up to 12-link, 1-d.o.f is tabulated with there processing time. New result is of 12-link, 1-d.o.f inversions. All other results match with the previously reported results in their particular category.

Number of *distinct* mechanism *obtained*

Category of SJKC	**No. of Distinct Inversions**	**CPU time for complete analysis**
6 Link, 1-d.o.f.	5	2 sec
8 Link, 1-d o f.	71	4 sec
9 Link, 2-d.o.f	254	10 sec
10 Link, 1-d.o.f.	1834	1 min2 sec
11 Link, 2-d.o.f.	7898	2 min 20 sec
12 Link, 1-d.o.f.	75594	5 min 04 sec

Conclusion

As the genesis of a chain is clearly defined, it can realize that sketching becomes easier. Primary Hamming number with least modifications is used to arrive at the number of inversions possible in a particular category of kinematic chains. Further, best inversions for path generation and function generation is also predicted.

References

27.L.S. WOO, J.*Engng Ind. Trans.* ASME 89,159-172 (February 1967)
28.T.S. Mruthyunjaya and M. R Raghavan , Mech. Mach. Theory 19, 357-368 (1984)
29.J.Kobler, U. Schoning and J. Toran , Burkhauser , (1993)

30 T S Mruthyunjaya and H R Balasubramaniyan, J Mech Design, Trans ASME101, 488-494 July 1979)

31C S Tang and T Liu, J. Mech. Design Trans. ASME, 115, 627-630, (September 1993).

Uicker Jr and A. Raicu, Mech Mach. Theory 10, 375-383 (1975)

33 T S. Mruthyunjaya and H. R. Balasubramaniyam, Mech. Mach. Theory 22,131-139 (1987)

34 T.S. Mruthyunjaya and M. R. Raghavan,J. Mech. Design Trans ASME 101,488-4941979)

35 A.C.Rao, Mech.Mach. Theory, 32.No.4.pp 477-488,1997

36 A.C.Rao, Mech, Mach, Theory, vol 28, No 1, pg 113-126, 1993.

37 A.C.Rao, Mech, Mach, Theory, vol 28, No 1, pg 127-143, 1993.

38.A.C Rao and VVNRP Raju, Trans of ASME,Mechanical Design, vol 122. March 2000,31-40.

APPENDIX-3

Chain Number	Representation Code
1	4 6(1) DLP (12)
2.	4 6(1) DLP (13)
3.	4 6(1) DLP (14)
4.	4 6(1) DLP (34)
5.	4 6(1) DLP (36)
6.	4 6(1) DLP (46)
7.	4 6(2) DLP (13)
8.	4 6(2) DLP (15)
9.	4 6(2) DLP (16)
10.	4 6(2) DLP (25)
11.	4 6(2) DLP (25)
12.	4 6(2) DLP (56)
13.	4 6(1) DJ (12) AALP (35)
14	4 6(1) DJ (12) AALP (36)
15.	4 6(1) DJ (15) AALP (15)
16.	4 6(1) DJ (15) AALP (67)

Representation of 8-Link 1 d.o.f. Chains

Primary Hamming	Secondary Hamming	Connectivity	Representation Code
1 208	2288	1,2 1,4 1,5 1,7 2,3 2,6 2,8 3,4 5,6 7,8	4 6(1) DLP(12)
2 212	2508	1,2 1,4 1,5 1,7 2,3 2,6 3,4 3,8 5,6 7,8	4 6(1) DLP(13)
3. 212	2464	1,2 1,4 1,5 1,7 2,3 2,6 3,4 4,8 5,6 7,8	4 6(1) DLP(14)
4. 216	2576	1,2 1,4 1,5 2,3 2,6 3,4 3,7 4,8 5,6 7,8	4 6(1) DLP(34)
5. 216	2456	1,2 1,4 1,5 2,3 2,6 3,4 3,7 5,6 6,8 7,8	4 6(1) DLP(36)
6. 216	2656	1,2 1,4 1,5 2,3 2,6 3,4 4,7 5,6 6,8 7,8	4 6(1) DLP(46)
7. 208	2096	1,2 1,4 1,5 1,7 2,3 3,4 3,6 3,8 5,6 7,8	4 6(2) DLP(13)
8. 212	2488	1,2 1,4 1,5 1,7 2,3 3,4 3,6 5,6 5,8 7,8	4 6(2) DLP(15)
9. 212	2448	1,2 1,4 1,5 1,7 2,3 3,4 3,6 5,6 6,8 7,8	4 6(2) DLP(16)
10. 216	2448	1,2 1,4 1,5 2,3 2,7 3,4 3,6 4,8 5,6 7,8	4 6(2) DLP(24)
11. 216	2612	1,2 1,4 1,5 2,3 2,7 3,4 3,6 5,6 5,8 7,8	4 6(2) DLP(25)
12. 216	2600	1,2 1,4 1,5 2,3 3,4 3,6 5,6 5,7 6,8 7,8	4 6(2) DLP(56)
13. 216	2672	1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 5,6 5,8	4 6(1) DJ(12)AALP(35)
14. 216	2424	1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 5,6 6,8	4 6(1) DJ(12)AALP(36)
15. 212	2072	1,2 1,4 1,7 1,8 2,3 3,4 3,6 5,6 5,7 5,8	4 6(1) DJ(15) AALP(15)
16. 216	2608	1,2 1,4 1,7 2,3 3,4 3,6 5,6 5,7 6,8 7,8	4 6(1) DJ(15) AALP(67)

ANALYSIS OF 9-LINK, 2-D.O.F SIMPLE JOINTED PLANAR KINEMATIC CHAINS

1 2nd HAM=2976 : 1,4 1,5 1,7 1,8 2,3 2,6 2,7 2,9 3,4 5,6 8,9

(a) Inversion = 3 Detail (1 2) (3 4 5 6 8 9) (7)

(b) Best Inversion= (6)

(c) Path & Function Generation : (3 4 5 6 8 9) (7) Binary

(d) (1 2) Quaternary

2. 2nd HAM=3492 : 1,4 1,5 1,7 1,8 2,3 2,6 2,7 3,4 3,9 5,6 8,9

(a) Inversion = 5 Detail (1) (2 3) (4 7) (5 8) (6 9)

(b) Best Inversion= (3 6)

(c) Path & Function Generation : (4 7) (5 8) (6 9) Binary

(d) (2 3) Ternary

(e) (1) Quaternary

3. 2nd HAM=3508 : 1,4 1,5 1,7 1,8 2,3 2,6 2,7 3,4 4,9 5,6 8,9

(a) Inversion = 8 Detail (1) (2) (3) (4) (5) (6 9) (7) (8)

(b) Best Inversion= (6)

(c) Path & Function Generation : (3) (5) (6 9) (7) (8) Binary

(d) (2) (4) Ternary

(e) (1) Quaternary

4. 2rd HAM=3492 : 1,4 1,5 1,7 1,8 2,3 2,6 2,7 3,4 5,6 7,9 8,9

(a) Inversion = 6 Detail (1) (2) (3 6 9) (4 5) (7) (8)

(b) Best Inversion= (3 4)

(c) Path & Function Generation : (3 6 9) (4 5) (8) Binary

(d) (2) (7) Ternary

(e) (1) Quaternary

5 2nd HAM=3808 . 1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 4,9 5,6 8,9

(a) Inversion = 5 Detail (1 2) (3 4) (5 6) (7) (8 9)

(b) Best Inversion= (6)

(c) Path & Function Generation (5 6) (7) (8 9) Binary

(d) (1 2) (3 4) Ternary

6 2nd HAM=3880 1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 5,6 5,9 8,9

(a) Inversion = 4 Detail (1 2) (3 5) (4 6 7) (8 9)

(b) Best Inversion= (3 5 6)

(c) Path & Function Generation : (4 6 7) (8 9) Binary

(d) (1 2) (3 5) Ternary

7 2nd HAM=3640 : 1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 5,6 6,9 8,9

(a) Inversion = 6 Detail (1) (2) (3 6) (4 5) (7) (8 9)

(b) Best Inversion= (6)

(c) Path & Function Generation : (4 5) (7) (8 9) Binary

(d) (1) (2) (3 6) Ternary

8. 2nd HAM=3696 : 1,4 1,5 1,7 2,3 2,6 2,7 3,4 3,8 5,6 7,9 8,9

(a) Inversion = 5 Detail (1 3) (2 7) (4) (5 8) (6 9)

(b) Best Inversion= (3)

(c) Path & Function Generation . (4) (5 8) (6 9) Binary

(d) (1 3) (2 7) Ternary

9. 2nd HAM=3056 : 1,2 1,5 1,7 1,8 2,3 2,6 2,9 3,4 4,7 5,6 8,9

(a) Inversion = 3 Detail (1 2) (3 5 6 7 8 9) (4)

(b) Best Inversion= (2 4 5 6 7 8)

(c) Path & Function Generation : (3 5 6 7 8 9) (4) Binary

(d) (1 2) Quaternary

10. 2nd HAM=3444 : 1,2 1,5 1,7 1,8 2,3 2,6 3,4 4,7 5,6 5,9 8,9

(a) Inversion = 8 Detail (1) (2) (3 9) (4) (5) (6) (7) (8)

(b) Best Inversion= (5)

(c) Path & Function Generation . (3 9) (4) (6) (7) (8) Binary

(d) (2) (5) Ternary

(e) (1) Quaternary

11 2^{nd} HAM=3424 1,2 1,5 1,7 1,8 2,3 2,6 3,4 4,7 5,6 6,9 8,9

(a) Inversion = 8 Detail (1) (2) (3 9) (4) (5) (6) (7) (8)

(b) Best Inversion= (4)

(c) Path & Function Generation (3 9) (4) (5) (7) (8) Binary

(d) (2) (6) Ternary

(e) (1) Quaternary

12 2^{nd} HAM=3500 1,2 1,5 1,7 1,8 2,3 2,6 3,4 4,7 5,6 7,9 8,9

(a) Inversion = 5 Detail (1) (2 7) (3 4) (5 8) (6 9)

(b) Best Inversion= (4 7)

(c) Path & Function Generation (3 4) (5 8) (6 9) Binary

(d) (2 7) Ternary

(e) (1) Quaternary

13. 2^{nd} HAM=3656 : 1,2 1,5 1,7 2,3 2,6 3,4 3,8 4,7 4,9 5,6 8,9

(a) Inversion = 5 Detail (1 4) (2 3) (5 9) (6 8) (7)

(b) Best Inversion= (6)

(c) Path & Function Generation : (5 9) (6 8) (7) Binary

14. 2^{nd} HAM=3792 : 1,2 1,5 1,7 2,3 2,6 3,4 3,8 4,7 5,6 5,9 8,9

(a) Inversion = 7 Detail (1) (2) (3) (4 8 9) (5) (6) (7)

(b) Best Inversion= (5)

(c) Path & Function Generation : (4 8 9) (6) (7) Binary

(d) (1) (2) (3) (5) Ternary

15. 2^{nd} HAM=3668 : 1,2 1,5 1,7 2,3 2,6 3,4 3,8 4,7 5,6 6,9 8,9

(a) Inversion = 5 Detail (1 6) (2) (3) (4 7 8 9) (5)

(b) Best Inversion= (4)

(c) Path & Function Generation : (4 7 8 9) (5) Binary

(d) (1 6) (2) (3) Ternary

16. 2^{nd} HAM=3748 : 1,2 1,5 1,7 2,3 2,6 3,4 4,7 4,8 5,6 5,9 8,9

(a) Inversion = 9 Detail (1) (2) (3) (4) (5) (6) (7) (8) (9)

(b) Best Inversion= (6)

(c) Path & Function Generation : (3) (6) (7) (8) (9) Binary

(d) (1) (2) (4) (5) Ternary

17 2nd HAM=3696 1,2 1,5 1,7 2,3 2,6 3,4 4,7 5,6 5,8 6,9 8,9

(a) Inversion = 5 Detail (1 2) (3 7) (4) (5 6) (8 9)

(b) Best Inversion= (2 6)

(c) Path & Function Generation (3 7) (4) (8 9) Binary

(d) (1 2) (5 6) Ternary

18 2nd HAM=3552 : 1,2 1,4 1,7 1,8 2,3 2,9 3,4 3,6 5,6 5,7 8,9

(a) Inversion = 9 Detail (1) (2) (3) (4) (5) (6) (7) (8) (9)

(b) Best Inversion= (3)

(c) Path & Function Generation : (4) (5) (6) (7) (8) (9) Binary

(d) (2) (3) Ternary

(e) (1) Quaternary

19. 2nd HAM=2832 : 1,2 1,4 1,7 1,8 2,3 3,4 3,6 3,9 5,6 5,7 8,9

(a) Inversion = 5 Detail (1 3) (2 4) (5) (6 7) (8 9)

(b) Best Inversion= (1 3)

(c) Path & Function Generation : (2 4) (5) (6 7) (8 9) Binary

(d) (1 3) Quaternary

20. 2nd HAM=3220 : 1,2 1,4 1,7 1,8 2,3 3,4 3,6 5,6 5,7 5,9 8,9

(a) Inversion = 7 Detail (1) (2 4) (3 5) (6) (7) (8) (9)

(b) Best Inversion= (6)

(c) Path & Function Generation : (2 4) (6) (7) (8) (9) Binary

(d) (3 5) Ternary

(e) (1) Quaternary

21. 2nd HAM=3624 : 1,2 1,4 1,7 1,8 2,3 3,4 3,6 5,6 5,7 6,9 8,9

(a) Inversion = 6 Detail (1) (2 4) (3) (5 9) (6) (7 8)

(b) Best Inversion= (1 3)

(c) Path & Function Generation : (2 4) (5 9) (7 8) Binary

(d) (3) (6) Ternary

(e) (1) Quaternary

22 2^{nd} HAM=3624 1,2 1,4 1,7 1,8 2,3 3,4 3,6 5,6 5,7 7,9 8,9

(a) Inversion = 8 Detail (1) (2 4) (3) (5) (6) (7) (8) (9)

(b) Best Inversion= (1 3)

(c) Path & Function Generation (2 4) (5) (6) (8) (9) Binary

(d) (3) (7) Ternary

(e) (1) Quaternary

23 2^{nd} HAM=3672 1,2 1,4 1,7 2,3 2,8 3,4 3,6 4,9 5,6 5,7 8,9

(a) Inversion = 4 Detail (1 2 3 4) (5) (6 7) (8 9)

(b) Best Inversion= (7 8)

(c) Path & Function Generation . (5) (6 7) (8 9) Binary

(d) (1 2 3 4) Ternary

24. 2^{nd} HAM=3592 : 1,2 1,4 1,7 2,3 2,8 3,4 3,6 5,6 5,7 5,9 8,9

(a) Inversion = 7 Detail (1 3) (2) (4) (5) (6 7) (8) (9)

(b) Best Inversion= (5 6)

(c) Path & Function Generation : (4) (6 7) (8) (9) Binary

(d) (1 3) (2) (5) Ternary

25. 2^{nd} HAM=3520 : 1,2 1,4 1,7 2,3 3,4 3,6 5,6 5,7 5,8 6,9 8,9

(a) Inversion = 8 Detail (1) (2 4) (3) (5) (6) (7) (8) (9)

(b) Best Inversion= (6)

(c) Path & Function Generation : (2 4) (7) (8) (9) Binary

(d) (1) (3) (5) (6) Ternary

26. 2^{nd} HAM=3816 : 1,2 1,4 1,7 2,3 3,4 3,6 5,6 5,7 6,8 7,9 8,9

(a) Inversion = 5 Detail (1 3) (2 4) (5) (6 7) (8 9)

(b) Best Inversion= (4)

(c) Path & Function Generation : (2 4) (5) (8 9) Binary

(d) (1 3) (6 7) Ternary

27. 2^{nd} HAM=3448 : 1,2 1,4 1,8 2,3 2,9 3,4 3,5 3,7 5,6 6,7 8,9

(a) Inversion = 8 Detail (1) (2) (3) (4) (5 7) (6) (8) (9)

(b) Best Inversion= (3)

(c) Path & Function Generation : (4) (5 7) (6) (8) (9) Binary

(d) (1) (2) Ternary

(e) (3) Quaternary

28 2nd HAM=3168 1,2 1,4 1,8 2,3 3,4 3,5 3,7 3,9 5,6 6,7 8,9

(a) Inversion = 7 Detail (1) (2 4) (3) (5 7) (6) (8) (9)

(b) Best Inversion= (1 3)

(c) Path & Function Generation : (2 4) (5 7) (6) (8) (9) Binary

(d) (1) Ternary

(e) (3) Pentanary

29 2nd HAM=3396 . 1,2 1,4 1,8 2,3 3,4 3,5 3,7 5,6 6,7 6,9 8,9

(a) Inversion = 4 Detail (1 6) (2 4 5 7) (3) (8 9)

(b) Best Inversion= (1 3 4 6)

(c) Path & Function Generation : (2 4 5 7) (8 9) Binary

(d) (1 6) Ternary

(e) (3) Quaternary

30. 2nd HAM=3256 : 1,2 1,4 2,3 2,8 3,4 3,5 3,7 3,9 5,6 6,7 8,9

(a) Inversion = 6 Detail (1 8) (2) (3) (4 9) (5 7) (6)

(b) Best Inversion= (3 8)

(c) Path & Function Generation : (1 8) (4 9) (5 7) (6) Binary

(d) (2) Ternary

(e) (3) Pentanary

31. 2nd HAM=3404 : 1,2 1,4 2,3 2,8 3,4 3,5 3,7 4,9 5,6 6,7 8,9

i) Inversion = 6 Detail (1) (2 4) (3) (5 7) (6) (8 9)

) Best Inversion= (0)

) Path & Function Generation : (1) (5 7) (6) (8 9) Binary

) (2 4) Ternary

) (3) Quaternary

32. 2nd HAM=3320 : 1,6 1,7 2,3 2,5 3,4 4,5 4,6 4,8 6,9 7,8 7,9

(a) Inversion = 7 Detail (1 9) (2) (3 5) (4) (6) (7) (8)

(b) Best Inversion= (7)

(c) Path & Function Generation : (1 9) (2) (3 5) (8) Binary

(d) (6) (7) Ternary

(e) (4) Quaternary

33 2nd HAM=3544. 1,2 1,4 1,5 2,3 2,6 3,4 3,9 5,6 6,8 7,8 7,9

(a) Inversion = 6 Detail (1) (2) (3 6) (4 5) (7) (8 9)

(b) Best Inversion= (3 4)

(c) Path & Function Generation : (4 5) (7) (8 9) Binary

(d) (1) (2) (3 6) Ternary

34 2rd HAM=3468. 1,4 1,7 1,8 1,9 2,3 2,6 2,7 3,4 3,8 5,6 5,9

(a) Inversion = 7 Detail (1) (2 3) (4 8) (5) (6) (7) (9)

(b) Best Inversion= (6)

(c) Path & Function Generation : (4 8) (5) (6) (7) (9) Binary

(d) (2 3) Ternary

(e) (1) Quaternary

35. 2nd HAM=3368: 1,4 1,5 1,7 2,6 2,7 2,9 3,4 3,8 3,9 5,6 6,8

(a) Inversion = 4 Detail (1 3) (2 6) (4) (5 7 8 9)

(b) Best Inversion (4 6 7 8)

(c) Path & Function Generation : (4) (5 7 8 9) Binary

(d) (1 3) (2 6) Ternary

36. 2nd HAM=3424 : 1,2 1,7 1,8 1,9 2,3 2,6 3,4 4,7 4,8 5,6 5,9

(a) Inversion = 8 Detail (1) (2) (3) (4) (5) (6) (7 8) (9)

(b) Best Inversion= (6 7)

(c) Path & Function Generation : (3) (5) (6) (7 8) (9) Binary

(d) (2) (4) Ternary

(e) (1) Quaternary

37. 2nd HAM=3784 : 1,2 1,5 1,9 2,3 2,6 3,4 3,8 4,7 5,6 5,8 7,9

(a) Inversion = 9 Detail (1) (2) (3) (4) (5) (6) (7) (8) (9)

(b) Best Inversion= (7)

(c) Path & Function Generation . (4) (6) (7) (8) (9) Binary

(d) (1) (2) (3) (5) Ternary

38 2^{nd} HAM=3664 1,2 1,4 1,9 2,3 2,6 3,7 3,8 4,7 4,8 5,6 5,9

(a) Inversion = 5 Detail (1 2) (3 4) (5) (6 9) (7 8)

(b) Best Inversion= (6 7)

(c) Path & Function Generation : (5) (6 9) (7 8) Binary

(d) (1 2) (3 4) Ternary

39 2^{nd} HAM=3752 : 1,2 1,4 1,9 2,3 3,4 3,6 4,8 5,6 5,7 6,8 7,9

(a) Inversion = 5 Detail (1 6) (2 8) (3 4) (5 9) (7)

(b) Best Inversion= (1 7)

(c) Path & Function Generation : (2 8) (5 9) (7) Binary

(d) (1 6) (3 4) Ternary

40. 2^{nd} HAM=3480 : 1,2 1,4 1,9 2,3 3,4 3,6 5,7 5,8 5,9 6,7 6,8

(a) Inversion = 4 Detail (1 5) (2 4 7 8) (3 6) (9)

(b) Best Inversion= (1 3 6 7)

(c) Path & Function Generation : (2 4 7 8) (9) Binary

(d) (1 5) (3 6) Ternary

(e) Total Inversion = 254