Application of Graph Theory, Critical Path & Machine Learning to Find Inversions of a Kinematic Chain

R Giri Prasad[1], G S N Murthy[2], Ch S V V S N Murthy[3], A Ramesh[4], M Sreenivasa Reddy[5] and V.V. Kamesh[6*]
{geogiriprasad@gmail.com[1], murthygsnm@yahoo.com[2], chsatyamurty@gmail.com[3], provc_es@adityauniversity.in[4], srmedapati@gmail.com[5], kameshvv@gmail.com[6]}

PG Student, Department of Computer Science and Engineering, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh 533437, India[1]
Professor, Department of Computer Science and Engineering, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh 533437, India[2]
Associate Professor, Department of Computer Science and Engineering, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh 533437, India[3]
Professor, Department of Electrical and Electronics Engineering, Aditya University, Surampalem, Andhra Pradesh 533437, India[4]
Professor, Department of Mechanical Engineering, Aditya University, Surampalem, Andhra Pradesh 533437, India[5, 6]

Abstract. In the analysis of mechanisms, Graph theory is being applied since many years. Representing a k-chain as a planar graph and analyzing by various numerical methods is possible by representing link as node and joint as path in the graph network. In the analysis of Networks i.e., computer networks, project networks and transportation networks, the largest time duration is taken as one of the key elements based on which Critical path and Critical activities are found. K-nearest neighbours is one of the Machine learning technique generally used for the classification of the elements. In the present paper, a new method is proposed in which all the above techniques are combined to find the inversions of a k-chain. Results for 8-link 1-dof k-chains are analyzed and presented.

Keywords: Graph conversion, Longest Path, k-NN(R)

1 Introduction

The identification of isomorphism and inversions in kinematic chains has long been a foundational challenge in mechanism design, with implications for mobility analysis, synthesis of mechanisms, and robotic structure optimization. Early approaches focused on numerical invariants and adjacency-based representations to distinguish structurally similar chains. Rao and Varada Raju [1] pioneered the use of Hamming number techniques to detect isomorphism and inversions, which was later extended through loop-based pseudo Hamming values to incorporate preferred frames and actuator positions [2]. Adjacent-chain tables [3] and fuzzy logic frameworks [4] further enriched the structural analysis by capturing symmetry, parallelism, and mobility. Subsequent efforts expanded the scope to larger chains and more complex configurations. Hasan et al. [5] and Mohammad et al. [6] introduced extended adjacency matrices and link-based invariants to identify distinct mechanisms up to 10-link chains. Parallelism estimation in robotic structures [7] and structural identification of planar inversions [8] added geometric and topological depth to the problem. Bal et al. [9] and Rizvi et al. [10] proposed efficient algorithms leveraging link invariant functions and adjacency-based heuristics, while Rai and Punjabi [11]

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358086

explored elusive methods for inversion detection using novel structural metrics. Recent advancements have emphasized algorithmic innovation and symbolic representations. Kamesh [12] and Kamesh et al. [13] proposed polynomial-based algorithms and rigidity approaches [14] to systematically enumerate distinct mechanisms. Gradient-based methods [15] and link adjacency value techniques [16] further refined the detection process, offering improved computational efficiency and accuracy. Parallel to these developments, the rise of machine learning has opened new avenues for structural pattern recognition in mechanical systems. Amini et al. [17] and Wang et al. [18] surveyed ML-based fault diagnosis and data gathering in mechanical engineering, underscoring the potential of intelligent models in structural analysis. Most notably, Kamesh and Rajesh [19] introduced a hybrid machine learning framework to identify inversions in planar kinematic chains, demonstrating the feasibility of combining symbolic and data-driven approaches.

2 Working Procedure of the Proposed Work

In this proposed work, there will be three stages. Initially k-chain is to be represented as a graph. In the second stage, critical path is to be found for each node to every other node, all the data is to be stored. In the third stage, k-NN machine learning technique is to be applied to analyses the data so that the distinct mechanisms could be found very easily.

2.1 Critical Path Finding & Machine learning technique application

Critical path method is generally preferred find the longest path or travelling path from one station to another station in any network i.e., computer network, graph network or transportation network. At first, the kinematic chain is to be transformed as a graph. The longest path or distance between point or station to another point or station can be known by applying the Critical path method.

In the earlier works [1 - 19], different method is being developed by many researchers relating to kinematic chains, graph theory, critical path and application of machine learning techniques in various engineering and societal issues. But there is no single method combining all of these important concepts in a single domain. Being motivated to combine all the above key concepts in the analysis of kinematic chains, new method is proposed, which will be explained in the later sections.

The various stages and steps are explained as follows.

Stage 1:

In this stage, kinematic chain is transformed as a undirected weighted graph. Each link and joint are to transformed as Node and Path.

Stage 2:

The procedure to get Critical path i.e., longest path from each node to every other node is to be applied by online tools. (www.graphonline.ru). From this tool we can extract 'longest path' in any network or graph. All the longest path values are to be stored in the form of a table.

Stage 3:

The longest path values from each to node every other node is to be sorted in the ascending order. After that, k-NN machine learning technique is to be applied to find the value of k-NN. In this technique, the parameter ‘k’ is taken an odd value i.e., 5 in this case. In the present paper, a new parameter is proposed namely “k-NN (R)” in which the ‘k’ values from other side of the sorting array are to be summed to generate a new value which will be designated as ‘k-NN(R)’. Now, k-NN and k-NN(R) are added to give a unique value for any link. Each link’s unique value along with its adjacent links are summated up to evolve a decision-making numerical value which can be defined as ‘Link Characteristic Value’ (LCV). Based on this, clustering will be done based on similar LCV values. After clustering process is completed, the nodes behaving similar are to be formed as ‘group’ which are to be identified as ‘Distinct mechanisms.

All the 3-stages are shown in next-section by taking one example from the Appendix.

3 Worked Out Example for the Proposed Method

One kinematic chain from Appendix (chain no. 4) is taken for calculation. All the 3 stages will be processed to find the distinct mechanisms.

Stage 1:

In this stage, k-chain is to be transformed as graph. In the example k-chain is shown in Fig 1, link 1 is taken as Node 1. As link-1 is connected to two links 2 and 6, two paths are to be made ready for Node 1. Similarly, link-2 is connected to three links 1, 3 and 7. So, three paths are to be made ready for Node 2. In the same fashion, all the links are to be transformed as Nodes and all the joints are to be represented as paths or connections between the nodes. The graph is shown in Fig 2.

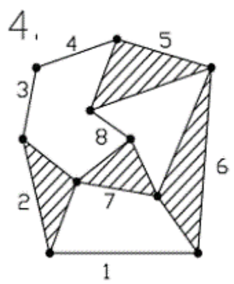

Fig. 1. Primary k-chain (chain 4)

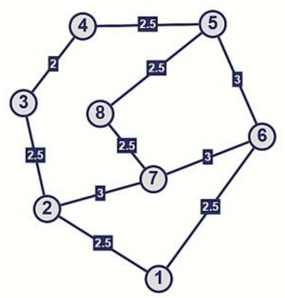

Fig. 2. Graph representation.

Stage 2:

The procedure to find the Critical path or longest path from each node to every other node is to be applied by online tools. (www.graphonline.ru). In this stage, graph is to be drawn in graphonline.ru. Here, the weightage for each path is to be taken as numerical average of both vertices’ incident value. In the Fig. 2, the weightage for the path between Node 6 and Node 7 is taken as average of 3 and 3 i.e., 3. The weightages of all the paths are shown in the table 1 shown below. The path is shown in the format ‘i-j’ where ‘i’ is the starting node and ‘j’ is the ending node.

Table 1. Weightages for paths between the nodes in Fig 2.

Path (i-j)	Weightage
1-2	2.5
1-6	2.5
2-3	2.5
2-7	3
3-4	2
4-5	2.5
5-6	3
5-8	2.5
6-7	3
7-8	2.5

Now the longest paths from each node to every other node are to be found using online tool option as shown below in fig 3.

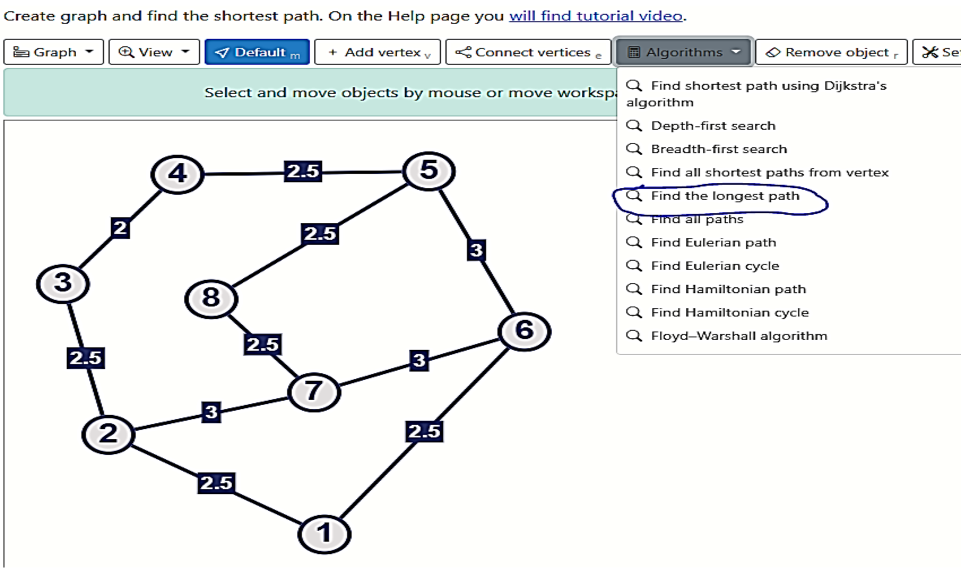

Fig. 3. Interface.

After finding all the longest path values, the results are shown in Table 2.

Table 2. Longest paths from Node-1 in Fig 2.

Node	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
1	0.0	17.5	16.0	18.0	15.5	17.5	15.5	18.0

Now all the longest paths are to be stored in the form of a matrix which is shown in Table 3.

Table 3. Longest paths from all nodes to all other nodes -Chain – 4.

	Node 1	**Node** 2	**Node** 3	**Node** 4	**Node** 5	**Node** 6	**Node** 7	**Node** 8
Node 1	0	17.5	16	18	15.5	17.5	15.5	18
Node 2	17.5	0	17.5	15.5	13	15	13	15.5
Node 3	16	17.5	0	18	15.5	17.5	15.5	18
Node 4	18	15.5	18	0	17.5	15.5	17.5	16
Node 5	15.5	13	15.5	17.5	0	13	15	17.5
Node 6	17.5	15	17.5	15.5	13	0	17	15.5
Node 7	15.5	13	15.5	17.5	15	17	0	17.5
Node 8	18	15.5	18	16	17.5	15.5	17.5	0

Stage 3: Application of Machine learning technique

To apply machine learning technique for the data in Table 3 above, we must sort the longest path values from lower value to higher value (ascending order). As k-Nearest Neighbor technique is to be applied to find the sum of the first ‘k’ nearer values, it should be in ascending order. Now the sorted longest distances are shown in Table 4.

Table 4. Sorted longest paths.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	0	0	0	0	0
15.50	13	15.50	15.50	13	13	13	15.50
15.50	13	15.50	15.50	13	15.00	15.00	15.50
16	15	16	16	15	15.50	15.50	16
17.50	15.50	17.50	17.50	15.50	15.50	15.50	17.50
17.50	15.50	17.50	17.50	15.50	17	17	17.50
18	17.50	18	18	17.50	17.50	17.50	18
18	17.50	18	18	17.50	17.50	17.50	18

For Node 1 : k-NN = 15.5+15.5+16+17.5+17.5 = 82

For Node 2: k-NN = 13+13+15+15.5+15.5 = 72

3.1 Motivation behind the k-NN(R)

In the k-nearest neighbor method, it is customary to calculate the sum of first ‘k’ nearest values to classify or cluster the nodes. But, when this technique is applied for a close network or graph, analysis for the remaining nodes will be left over, which leads to un-precise conclusions and results. Hence, it is necessary to involve all the nodes or elements in the analysis process. For this purpose, another parameter k-NN (R) is proposed in this paper, for which summation will be done for the ‘k’ elements taken from the other side. As the sorted list is in the ascending order, k-NN(R) is the summation of ‘k’ farthest longest paths.

For Node 1: k-NN (R) = 18+18+17.5+17.5+16 = 87

For Node 2: k-NN (R) = 17.5+17.5+15.5+15.5+15= 81

After finding of k-NN and k-NN (R) values, we need to find the ‘LCV” value to analyze the node characteristics.

3.2 Finding LCV (Link Characteristic Value)

All the k-links with similar LCV (Link Characteristic Value) are supposed to be UNIQUE. For the given case, the values of k-NN & k-NN(R) are shown in Table 5.

Table 5. Finding k-NN&k-NN(R).

Node 1	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
0	0	0	0	0	0	0	0
15.5	13	15.5	15.5	13	13	13	15.5
15.5	13	15.5	15.5	13	15	15	15.5
16	15	16	16	15	15.5	15.5	16
17.5	15.5	17.5	17.5	15.5	15.5	15.5	17.5
17.5	15.5	17.5	17.5	15.5	17	17	17.5
18	17.5	18	18	17.5	17.5	17.5	18
18	17.5	18	18	17.5	17.5	17.5	18
82	**72**	**82**	**82**	**72**	**76**	**76**	**82**
Node 1	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
87	**81**	**87**	**87**	**81**	**83**	**83**	**87**

In the above table, Row 9 and Row 11 represent k-NN and k-NN(R) values respectively.

Now, according to adjacency of each link (node), k-NN & k-NN(R) values are summated. These values are shown as.

L1	169
L2	153
L3	169
L4	169

L5	153
L6	159
L7	159
L8	169

Now, the Link Characteristic Value (LCV) is computed by adding the numerical values of each link with its adjacent link values. For example, Link 1 is having adjacency with Link 2 and Link 6. Hence, the LCV value for Link 1 will be calculated as the summation of 169 + 153 + 159 = 481. The LCV values of each link are calculated and shown in Table 6.

Table 6. LCV values of links of Chain no. 4.

	k-NN + k-NN(R)	Link1	Link2	Link3	Link4	Link5	Link6	Link7	Link8	LCV
Link 1	169	0	153	0	0	0	159	0	0	481
Link 2	153	169	0	169	0	0	0	159	0	650
Link 3	169	0	153	0	169	0	0	0	0	491
Link 4	169	0	0	169	0	153	0	0	0	491
Link 5	153	0	0	0	169	0	159	0	169	650
Link 6	159	169	0	0	0	153	0	159	0	640
Link 7	159	0	153	0	0	0	159	0	169	640
Link 8	169	0	0	0	0	153	0	159	0	481

From the above table, it is clear that Link 1 and Link 8 are having same LCV ➔ both are having similar behavior. Similarly, other links are also compared and distinct mechanisms are found and they are (1,8), (2,5), (3,4), (6,7) which leads to Four (4) inversions.

4 Results and Discussion

All the stages of the proposed method are implemented on all the kinematic chains of 8-link 1-dof, and the results are shown in Table 7.

Table 7. Results of proposed method for k-chains of 8-link 1-dof.

Chain	Value	L-1	L-2	L-3	L-4	L-5	L-6	L-7	L-8	Inversions	
1	K-NN	71	82	82	71	71	82	82	71	(1,4,5,8) ,(2,3,6,7)	2
	K-NN(R)	83	87	87	83	83	87	87	83		
2	K-NN	75	80	80	75	75	80	80	75	(1,4,5,8), (2,3,6,7)	2
	K-NN(R)	82	87	87	82	82	87	87	82		

3	K-NN	75	78	82	72	86	74	81	78	(1),(2),(3),(4),(5),(6),(7),(8)	8
	K-NN(R)	83	85	87	81	89	81	89	83		
4	K-NN	82	72	82	82	72	76	76	82	(1,8),(2,5),(3,4),(6,7)	4
	K-NN(R)	87	81	87	87	81	83	83	87		
5	K-NN	84	67	65	74	74	65	67	84	(1,8),(2,7),(3,6),(4,5)	4
	K-NN(R)	89	75	79	83	83	79	75	89		
6	K-NN	82	68	78	78	68	82	72	80	(1,6),(2,5),(3,4),(7),(8)	5
	K-NN(R)	87	75	85	85	75	87	77	85		
7	K-NN	78	72	82	68	82	72	74	80	(1),(2,6),(3,5),(4),(7),(8)	6
	K-NN(R)	85	77	87	73	87	77	81	85		
8	K-NN	74	59	59	74	59	59	74	74	(1,4,7,8),(2,3,5,6)	2
	K-NN(R)	83	71	71	83	71	71	83	83		
9	K-NN	74	74	81	74	81	81	74	81	(1,2,4,7),(3,5,6,8)	2
	K-NN(R)	81	81	85	81	85	85	81	85		
10	K-NN	55	72	62	67	62	72	72	72	(1),(2,6,7,8),(3,5),(4)	4
	K-NN(R)	63	78	68	78	68	78	78	78		
11	K-NN	58	68	70	54	74	64	74	70	(1),(2),(3),(4),(5,7),(6),(8)	7
	K-NN(R)	69	76	82	61	84	72	84	84		
12	K-NN	71	68	80	61	74	72	68	80	(1),(2),(3,8),(4),(5),(6),(7)	7
	K-NN(R)	83	74	86	71	83	78	76	86		
13	K-NN	70	74	78	60	78	76	66	78	(1),(2),(3),(4),(5),(6),(7),(8)	8
	K-NN(R)	81	80	86	69	86	84	73	88		
14	K-NN	72	78	82	65	82	78	72	77	(1,7),(2,6),(3,5),(4),(8)	5
	K-NN(R)	81	86	87	77	87	86	81	87		
15	K-NN	70	54	70	70	54	70	70	70	(1,3,4,6,7,8),(2,5)	2
	K-NN(R)	82	61	82	82	61	82	82	82		
16	K-NN	52	68	68	52	68	68	68	68	(1,4),(2,3,5,6),(7,8)	3
	K-NN(R)	59	76	76	59	76	76	78	78		
				TOTAL NO. OF INVERSIONS							**71**

5 Conclusion

The present method is an effective mixing of different methods i.e., graph theory, critical path method and machine learning technique in an effective way to find the inversions of k-chains. This method can be extended to next level of links and degree-of-freedom.

References

[1] A.C.Rao, D. Varada Raju: Application of the Hamming number technique to detect isomorphism among kinematic chains and inversions. Mechanism and machine theory 26(1), 55-75 (1991).

[2] A.C.Rao, C.N.Rao: Loop based pseudo hamming values-II Inversions, preferred frames, and actuators. Mechanism and machine theory 28(1), 129-143 (1993).

[3] C.J.Kui, C.W.Qing: Identification of Isomorphism among kinematic chains and inversions using link's adjacent-chain-table. Mechanism and machine theory 29(1), 53-58 (1994).

[4] A.C.Rao: Application of fuzzy logic for the study of isomorphism, inversions, symmetry, parallelism, and mobility in kinematic chains. Mechanism and machine theory 35, 1103-1116 (2000).

[5] Ali hasan, R.A.Khan, A.K.Dargar: Isomorphism and Inversions of Kinematic Chains Up to 10-Links. In: 13th National Conference on Mechanisms and Machines (NaCoMM07), IISc, Bangalore, India, (2007).

[6] A.Mohammad, R.A.Khan and V.P.Agrawal: Identification of kinematic chains and distinct mechanisms using extended adjacency matrix. Proc. IMechE Part C: J. Mechanical Engineering Science. 221, 81-88 (2007).

[7] A.B.S.Rao, V.V.Kamesh: An Estimate of Parallelism in In-parallel Robotic Structures. Proceedings of the National conference on Recent advances in Design, applications, and manufacturing systems (RADAMS 09), Thanthai periyar Govt. institute of technology, Vellore, 17^{th} April. (2009) 361-366.

[8] S.Sanyal: Structural identification of distinct inversions of planar kinematic chains. IIUM Engineering Journal. 85-92 (2011)

[9] J.S.Bal, P.B.Deshmukh, A.Jagadeesh: Link Invariant Functions and Detection of Isomorphism and Inversions of Kinematic Chains. Proceedings of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM2013), IIT Roorkee, India, Dec 18-20 (2013)

[10] S.S.H.Rizvi, A.Hasan, R.A.Khan: An efficient algorithm for distinct inversions and isomorphism detection in kinematic chains. Perspectives in Science. 8, 251-253 (2016).

[11] R.K.Rai, S.Punjabi: An Elusive Method to Identify Isomorphism and Inversions of Kinematic Chains and Mechanisms. Indian Journal of Science and Technology. 10(18), 1-13 (2017).

[12] V.V.Kamesh: A Novel Algorithm to Find Distinct Mechanisms of a Planar Kinematic Chain. International Journal of Innovative research explorer. 5(2018) 27-31.

[13] V.V.Kamesh, D.V.S.S.S.V.Prasad: A Novel algorithm to find distinct mechanisms of a kinematic chain using link polynomial. Proceedings of the National Conference on Recent Technologies in Engineering and Basic Sciences-2020 (NCRTEBS-2020). BVC Engineering College(A), Odalarevu,AP. (2020)

[14] Vinjamuri Venkata Kamesh, D.V.S.S.S.V.Prasad, P.S.Ranjit, Bh.Varaprasad, V.Srinivasa Rao: A rigidity approach to find distinct mechanisms of a planar kinematic chain. Materials today Proceedings (Elsevier) https://doi.org/10.1016/j.matpr.2020.11.684 (Article in Press)

[15] A.K.Shukla, S.Sanyal: Gradient method for identification of isomorphism of planar kinematic chains Australian Journal of Mechanical Engineering. 18(3). 1-18 (2017). DOI: 10.1080/14484846.2017.1374815

[16] A.Dargar, A.Hasan, R.A.Khan: Identification of isomorphism among kinematic chains and inversions using link adjacency values. IJMME, 4(3). 309-315 (2009)

[17] M.Amini,K.Sharifani,A.Rahmani: Machine learning Model towards evaluating data gathering methods in Manufacturing and Mechanical Engineering. IJASER. 15(4). (2023)

[18] Qi Wang,R.Huang,J.Xiong,X.Dong. A survey on Fault diagnosis of Rotating Machinery based on Machine learning. Doi: 10.1088/1361-6501/ad6203

[19] V.V.Kamesh, M.V.Rajesh. A Hybrid Machine Learning Approach to Find Inversions of Planar Kinematic Chains. Intelligent Computing and Communication, Lecture Notes in Networks and Systems 1241, https://doi.org/10.1007/978-981-96-1267-3_46 pp. 567-576

Appendix I: 8-link 1-dof k-chains