INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.195

Critical component detection in assemblies: a graph centrality approach

Robert Ballantyne ✉, Adam McClenaghan, Oliver Schiffmann and Chris Snider

University of Bristol, United Kingdom

✉ robert.ballantyne@bristol.ac.uk

Abstract

This study examines the use of graph centrality to identify critical components in assembly models, a method typically dominated by computationally intense analyses. By applying centrality measures to simulated assembly graphs, components were ranked to assess their criticality. These rankings were compared against Monte Carlo sensitivity analysis results. Preliminary findings indicate a promising correlation, suggesting graph centrality as a valuable tool in assembly analysis, enhancing efficiency and insight in critical component identification.

Keywords: *network analysis, digital manufacturing, modelling, tolerance representation and management*

1. Introduction

Real-time monitoring and analysis of assemblies is an increasingly critical concern in many industrial settings (Sun et al., 2020). This is particularly true for 'High Precision Products' like those in aviation and aerospace. In such environments, each assembly is typically unique, leading to higher costs and narrower profit margins (Bakker et al., 2017). Having knowledge of how the tolerances of these components interact to influence the Key Characteristics is critical to this challenge.

Graph theory is versatile branch of mathematics that deals with mapping the interactions of complex systems. It has demonstrated its use in wide-ranging applications across many disciplines such as neuroscience, biology, computer science, and social network analysis (Koutrouli et al., 2020; Farahani et al., 2019; Majeed and Rauf, 2020). In these fields, its ability to model complex relationships in our brains, the environment and the internet, through nodes and edges has provided profound insights.

Due to the complexity and size of many modern assemblies, and the growing need for faster, more efficient analysis, graphs are an ideal candidate to aid in understanding the relationships between components. In recent times, the scope of graph theory in assembly analysis has expanded. Studies such as: (Gunji et al., 2022), (Watson and Hermans, 2019) and (Kosec et al., 2020) have utilised graphs to model assembly sequences and their assemblability, with the latter integrating tolerance analysis via error propagation. Innovative applications such as using knowledge graphs for process planning (Zhou et al., 2022), kinematic modelling of a configurable assembly cell (Zhang et al., 2022) and the generation of graphs from boundary representation models that incorporate error transfer and tolerances (Li and Hou, 2021), exemplify the evolving role of graph theory in this field. Furthermore, the concept of knowledge graphs (see Akroyd et al., 2021) has also been used to build a semantic understanding of systems, aiming to develop comprehensive Digital Twins. This reflects a shift towards a more dynamic application of graph theory, extending beyond traditional data representation.

Previous research has introduced the use of unique properties of graph theory in assembly analysis (Ballantyne et al., 2023). We then proposed that centrality could indicate the importance or influence of

a component within the assembly, akin to how it identifies key nodes in social networks or critical pathways in transportation networks. This research delves deeper into this concept, particularly focusing on the alignment between centrality measures and tolerance analysis.
The objective of this study is to investigate the potential of graph centrality measures to identify critical components within assembly models. By comparing graph-based findings with Monte Carlo sensitivity analysis, this work aims to establish the potential contributions of graph theory to assembly analysis.
This paper is driven by two core aims:

1. **Investigate the Alignment of Graph Analysis with Monte Carlo (MC) Results:** This investigation seeks to begin the exploration into applying graph-based analysis to assemblies. As such it presents a cross-comparison between an existing method (MC) and the novel analysis technique, graph centrality. The primary objective is to assess how closely the results obtained from graph-based analysis correlate with those derived from MC sensitivity analysis.
2. **Assess Scalability of the Approach for Larger Assemblies:** Tolerance analysis, in particular MC, quickly becomes computationally intensive as assembly size or complexity grows. An essential aspect of this research is to evaluate how well this graph-based approach scales when applied to larger assembly models. This is important to predict how well the model will cope with real assembly models.

The structure of the paper is as follows: The methodology will outline the processes and criteria for graph generation and analysis, as well as the methodologies for both MC sensitivity analysis and graph centrality measures. The results section will then present a comparative analysis of these methods across different scales of assemblies. The discussion will interpret these results, their real-world implications, and acknowledge the study's limitations. Finally, the paper will conclude by summarising the key findings and suggesting directions for future research in this evolving field.

2. Methodology

As previously mentioned, this study tests the hypothesis that graph centrality, a measure of node importance, can align well with the key components identified in traditional tolerance sensitivity analysis. To explore this, we generated our own graphs that represent real-life assemblies, as suitable datasets were not available. Graph centrality algorithms were then applied to these graphs to determine nodes of highest influence on the assembly as a whole and compared to ground truth of the MC analysis. For the baseline tolerance analysis, we chose MC for its ease of implementation, despite some known limitations compared to other tolerance analysis methods (Kosec et al., 2020). These revolve around sensitivity issues with more complex tolerances, and as such will not affect the outcome of this work. In the graph analysis, we focused on common centrality measures applicable to undirected graphs, such as degree, betweenness and closeness centralities, as well as other less common ones. These metrics offer varied perspectives on the significance of nodes within the graph.

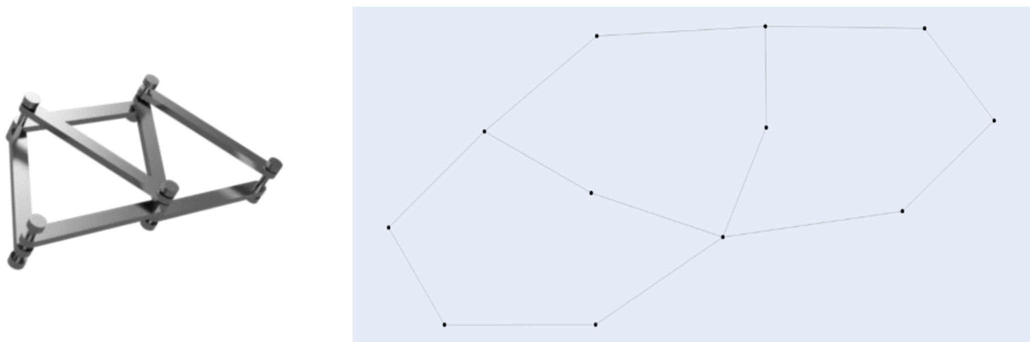

Figure 1. Truss Render with graph representation

The methodology culminates in a comparative analysis, where we employ two ranking systems to assess and compare the criticality of nodes as identified by both graph centrality measures and MC tolerance

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

analysis. Enhancing our understanding of how graph analysis can be used to assess component criticality in assembly modelling.

2.1. Method for graph generation

As information on aerospace assemblies is commercially sensitive it was essential to generate the models for analysis. This section introduces the method for graph generation, tailored for 2D assembly modelling as this simplifies the setup while retaining the analytical depth essential for thorough analysis. By employing graphs, which are non-dimensional and emphasise the relationships between the nodes and edges, we streamline the representation of assemblies. This allows us to concentrate on the interactions and dependencies within the assembly. The focus on 2D modelling lays the groundwork for future exploration into real 3D assemblies, where factors such as model accuracy and detailed tolerance chain nuances become crucial. An example of a truss represented by a graph is given in Figure 1, where each component and join piece on the graph has its own node, although this truss is not one of those used in this study.

2.1.1. Random generation with constraints

To produce realistic yet varied graphs it was essential to constrain the graph generation algorithm. This gives the graphs a sense of realism whilst also providing a variety of structures.
As such, the generation of these graphs was subject to these three constraints:

- **Node Connectivity**: Each node, representing a component in the assembly, was required to have between 2 and 5 edges, roughly approximating the average number of connections in a plane wingbox structure (Sarh, 1998). The exact number of edges for each node was randomly determined during the graph generation process. This randomness ensures a variety of graph configurations, simulating different potential assembly scenarios.
- **Planarity**: For a graph to be planar, there must be no overlapping edges. This proxy forces the model to be 2D, one of the model assumptions. Whenever an edge was added, the graph was checked for planarity and if this was broken then the edge would be added elsewhere.
- **Feature Nodes on Edges**: Recognising that components in assemblies are interconnected through assembly features, the model needed to capture this aspect. Consequently, for every edge in the component graph (which represents the interface between two components), two additional feature nodes were introduced. These feature nodes, one for each component at the edge, are connected to each other and to their respective component nodes. This structure captures the influence of assembly features on the overall assembly, something which will be especially useful in future models, where components may connect through multiple features with potentially different tolerances.

Once this graph was created, each edge that joined a component to an assembly feature was assigned a uniform tolerance band of ±0.2mm in both X and Y directions. This value was chosen to approximate real-world tolerances and is close to the B9 tolerance defined in ISO286 ("ISO 286" 2010, p. 2), used in large scale machinery and manufacturing settings.

2.2. Monte Carlo sensitivity analysis

In this study, the MC sensitivity analysis is tailored to determine how the various tolerances influence the uncertainty of a specific Key Characteristic (KC). After generating the planar graphs, we concentrate on identifying those that contribute more, or less, to the uncertainty. This approach allows us to determine which tolerances have the largest impact, and acts as a ground truth for comparison against network-derived measures. The MC method was chosen due to its straightforward implementation compared to alternatives like T-Maps or kinematic chains (Shen et al., 2005), as other methods require significantly more model setup. This strategic focus is instrumental in understanding the broader implications of tolerance variations on assembly performance. The following describes our procedure for conducting the MC analysis:

1. **Identification of Critical Paths**: Determined the paths through the assembly graph along which tolerance chains would be evaluated.

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

The Key Characteristics (KCs) were defined as the X and Y dimensional tolerances between the initial (0th) and terminal (N-1th) nodes of each assembly graph. To determine the critical paths, we utilised a Depth-First Search (DFS) technique. This approach involved incrementally increasing the depth cut-off in our DFS until at least 15 distinct paths were identified between the chosen start and end points. This methodology was chosen to ensure a comprehensive capture of various paths that contribute to the overall uncertainty in the assembly. By not restricting our analysis to the shortest path or paths, we are able to include a broader spectrum of pathways, encompassing those that play the most significant role in the dimensional accuracy and variability of the assembly as a whole. Investigation of impact of this choice would be beneficial for future work.

2. **Sampling Method**: Determined the way in which the tolerances were calculated for each path.
   a. For every identified path, the tolerance distribution stored on that edge was sampled. The sum of these tolerances gave the total variation for that path.
   b. For each iteration each of the path lengths was stored and added to a list.
3. **Convergence**: To ensure that the MC dataset was large enough to produce reliable results, it was essential to run the simulation until it converged. This was achieved as follows:
   a. The simulation was run in batches of 1000 samples, meaning that each of the critical paths was sampled 1000 times.
   b. After each batch, the standard deviation of all path lengths was taken (including those of previous batches).
   c. This was repeated until adding another batch changed the standard deviation by less than (0.01%, $1\text{x}10^{-4}$), corresponding approximately to an 8-sigma standard deviation.
4. **Baseline Determination**: Once convergence is achieved, the standard deviation of the X and Y tolerance sums is recorded. These values constitute the baseline.
5. **Sensitivity Analysis per Edge**: The analysis then proceeds to assess the sensitivity of the assembly to changes in tolerances for each edge:
   a. The tolerance distribution width for each edge is altered to 10% above and below the baseline value. Providing a perturbation from the normal, representative of a change in manufacturing quality for example.
   b. The MC analysis is re-run with this new distribution.
   c. The new standard deviation of X and Y positions are compared to the baseline values.
   d. The absolute difference between the baseline and current standard deviation was taken to be the sensitivity of that edge.
6. **Component Sensitivity Calculation:** Finally, for each component node in the graph, the sensitivities of all its connecting edges are summed. This total sensitivity score for each component indicates how changes in tolerances this component affects the assembly at large. The more an edge changes the standard deviation or mean, the more influential it is.

This MC sensitivity analysis method provides a baseline understanding of how tolerance variations in different parts of the assembly influence the overall assembly's dimensional accuracy and variability. Allowing for comparison with the graph-based metrics.

2.3. Graph centralities

In this study, centrality metrics needed to be suitable for undirected graphs while providing meaningful insights into the assembly's structure and the roles and influence of its components. A further constraint was that the metrics had to work with un-weighted graphs. This approach gives a clearer understanding of the assembly's inherent structural relationships, although modifying the weights is something we wish to modify alongside the centrality metrics in future work.

2.3.1. Centrality metrics

Centrality measures the importance of a given node in a graph, derived from its relationship to the other nodes. It stands to reason that more central nodes should have more influence on the KCs of an assembly, the hypothesis of this work. However, there are subtle differences between the metrics which invites research into if there are notable differences in outcome, and if they are predictable.

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

It is important to note that in our analysis, centrality measures were applied to the graph containing all components, rather than being limited to the subsections along the identified critical paths. This approach ensures a holistic view of the assembly's structure and the roles of all components within it. Future iterations of this study may include a comparative analysis between centrality metrics applied to the entire graph versus those applied only to critical paths, to explore any differences in identifying key components. The selected centrality metrics were from Networkx , a Python based graphing library, ("Networkx" 2023), mathematical descriptions of which are described by (Das et al., 2018):

1. **Degree Centrality**: This metric counts the number of direct connections a node has. In our assembly graphs, it reflects the number of direct relationships of each component.
2. **Betweenness Centrality**: It measures the frequency of a node appearing on the shortest paths between other nodes. This is useful for identifying components that act as bridges within the assembly.
3. **Closeness Centrality**: This centrality evaluates how close a node is to all other nodes, pinpointing components pivotal for efficient communication, be it of information or variations, within the assembly.
4. **Harmonic Centrality**: Like closeness, it sums the reciprocal of the shortest path distances to all nodes, offering a broad view of a component's interconnectedness.
5. **Load Centrality**: Load centrality focuses on the fraction of all shortest paths passing through a node. This metric helps identify nodes that bear a significant load in terms of the assembly's overall connectivity and flow.

Combining these metrics provides a comprehensive view of the assembly's structure, highlighting both the most connected components and those strategically positioned or critical to the assembly process. By applying these metrics to our non-directional, uniformly weighted graphs, we aim to gain insights into key components and their roles within the overall assembly structure. Future work will explore the potential for applying edge weights, but their role and the form those weights take is yet undecided.

2.4. Comparison of metrics

This study employs a comparative analysis methodology to align and assess the results of graph centrality measures against those derived from the MC sensitivity analysis. Two distinct methods were used to rank the centrality measures: the Sum of Ranks and a Voting System. This was as no one method is expected to be perfect, they measure different forms of centrality. As components can influence the assembly around them in multiple ways it was therefore essential to find a way to account for these different influences.

Sum of Ranks: The centrality results for each component were normalised between 0 and 1. This was to maintain homogeneity, as the centrality metrics measure different aspects of the nodes they work on vastly different scales. For example, degree centrality is the number of edges that a node has, therefore can be theoretically infinite. Whereas load centrality is the fraction of shortest paths that pass through a node and is therefore already limited to between 0 and 1. Aggregating each of these normalised centrality scores would then help identify the components which ranked highly across the different approaches.

Voting System: The second method implemented a voting mechanism, where each centrality metric 'voted' for its top 5 components. This number was chosen based on the dataset size, ensuring a focused yet comprehensive representation of key components. The voting system highlights components that are repeatedly recognised as critical across different centrality measures. The number of votes allocated to each metric is something that could be explored, as the ability of the specific centrality measures to identify critical nodes is still unknown.

Comparison of Results: In this study, Cohen's Kappa, a measure of inter-rater agreement, is used to assess the alignment between the graph centrality measures and MC sensitivity analysis. Cohen's Kappa evaluates the degree of agreement between two sets of categorical data, in this case, the component rankings, while accounting for agreement occurring by chance (Cohen, 1960).

Focusing on the top 5 components provides a concentrated analysis, facilitating more actionable insights in an assembly setting. This approach gauges the alignment between the two methods.

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

3. Results: Analysis and scalability

This section displays the outcomes of comparing graph centrality with MC sensitivity analysis, emphasis being the identification of critical assembly components. Additionally, it assesses the methodology's scalability and repeatability across different graph sizes, highlighting the robustness and applicability of our approach in diverse assembly contexts and industrial settings. Table 1 shows the mean Cohens Kappa scores across 3 runs for each of the 4 graph sizes.

Table 1. Cohen's Kappa results across graphs

	Metric as Compared to Cohens Kappa						
Component Number	Betweenness	Closeness	Degree	Harmonic	Load	Rank	Votes
10	0.33333	0.33333	0.60000	0.33333	0.33333	0.82222	0.64444
25	0.64127	0.64127	0.55079	0.46032	0.64127	0.81325	0.71880
100	0.59406	0.59406	0.59254	0.59254	0.59406	0.58519	0.67037
200	0.43906	0.43906	0.43906	0.19005	0.43906	0.42986	0.34444

As can be seen, for a given graph size no single metric is good enough, as none have full alignment (a Cohen's Kappa of 1), the focus of section 3.1. Equally, there are significant differences between the graph sizes. Determining how this method is affected by the graph size is explored in section 3.2.

3.1. Comparison of graph centrality and Monte Carlo sensitivity outcomes

In this section, we delve into the examination of a 100-component graph, selected for its illustrative value in demonstrating the application of graph theory to assembly analysis. This example serves as a representative case study, highlighting the potential of our methodology in identifying critical components within complex assembly structures.

3.1.1. Baseline Monte Carlo analysis

The baseline analysis was run on the subgraph containing all critical paths, which are distinctly highlighted in red linking Component 0 (marked in green) and Component 99 (shown in red), Figure 2. This subgraph, comprising 27 out of the total 100 nodes in the graph, was central to the scope of our analysis. It is important to note that assembly features are not depicted in the figure. The results taken for this paper converged after 21 batches of 1000 samples.

Table 2. Top 5 sensitivity scores for a 100 component graph

Component	32	36	37	3	0
Sensitivity	0.00404	0.00330	0.00297	0.00295	0.00295

3.1.2. Sensitivity analysis

In the subsequent phase of our study, a sensitivity analysis was conducted to delve deeper into the effects of tolerance variations within the assembly's critical paths.

The results of this sensitivity analysis are shown in Figure 3, with the top 5 results shown in Table 2. This figure illustrates the sensitivity of each node, with the colour intensity correlating to the maximum sensitivity observed in the standard deviations of the X and Y dimensions. It is important to note that despite these variations in sensitivity, the mean values for these dimensions remained approximately zero, consistent with the expected outcomes of our model. These results imply that Component 32 has the highest potential to drive the variation of the chosen KCs. Even though the tolerances of each component was increased by the same amount, they had different effects on the KC, resulting in a difference in criticality.

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

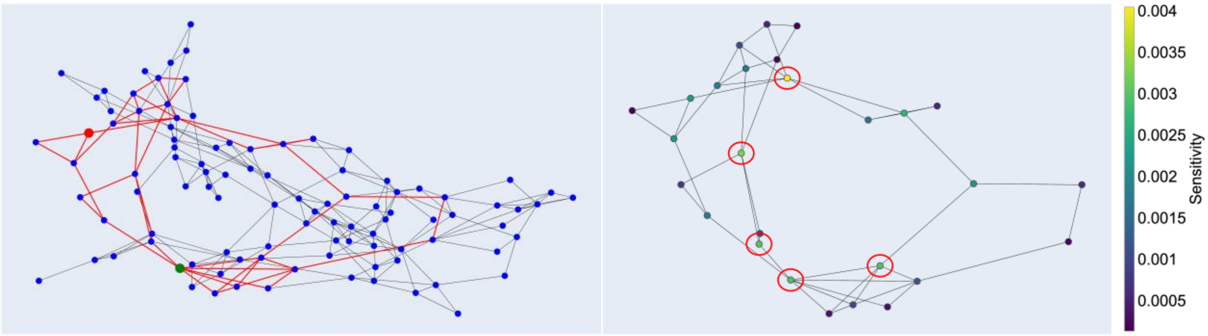

Figure 2. Full assembly graph highlighting critical paths; Critical component subgraph, coloured by sensitivity

3.1.3. Centrality metrics and consolidated ranking results

In this section, we compare the results from two distinct centrality ranking methods applied to our 100-component graph: the Sum of Ranks and the Voting System. These provide different perspectives on identifying critical nodes, and a comparative analysis offers an overview of the graph's structure.
Figures 4 and 5 show the results of the graph centrality metrics, with the top 5 nodes for each aggregate circled in red, with the top 5 for sensitivity circled in green. Figure 4 displays the results of the Sum of Ranks method, aggregating normalised scores from various centrality measures, while Figure 5 shows the results of the Voting System, where each centrality metric put forward its top-ranked nodes. As these results are aggregate scores, the absolute values are not insightful. However, the relative scores between nodes may prove more useful in future research, once the aggregation methods have been revised.
It was found that the original hypothesis that the aggregate measures would provide an improvement over individual centralities was also dependant on the topology. For example, for the first 100 component graph, highlighted above, each centrality had a Cohens Kappa of 0.75 when compared to the sensitivity analysis. Which, as a result, was the same for the aggregate scores. Whereas for the 2nd generated graph of 100 components, the aggregate scores outperformed the individual metrics, Table 3.

Table 3. Cohens Kappa scores comparing graph metrics to sensitivity top 5

Metric	Degree	Closeness	Betweenness	Harmonic	Load	Ranking	Voting
Cohens Kappa	0.5	0.5	0.66	0.5	0.66	0.82	0.82

3.1.4. Comparison of methods

The top 5 results from each method (sensitivity analysis, sum of ranks, and voting system) are shown in Figures 4 and 5. This shows a high level of agreement, with 4 of the 5 being presented by all methods (a Cohens Kappa of 0.75 in both cases). The graph methods favour node 7 whereas the sensitivity analysis favours node 37. This may be explained by the fact that node 37 is directly connected to node 0 and is more influential for sensitivity. Node 7 is generally favoured by the graph metrics, possibly because it is central in the full graph.

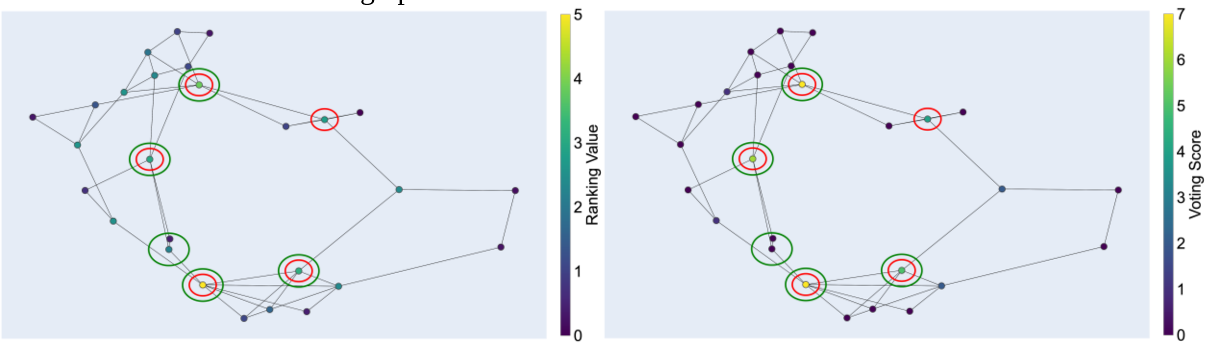

Figure 3. Graph highlighting sum of ranks score; Graph highlighting voting system score

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

3.2. Efficiency and consistency across graph sizes

Understanding the consistency and computational efficiency of graph analysis methods as they scale up in size forms the crux of this comparative analysis. We focus on evaluating Cohen's Kappa scores for different graph sizes to assess consistency and examine the run times to gauge computational efficiency.

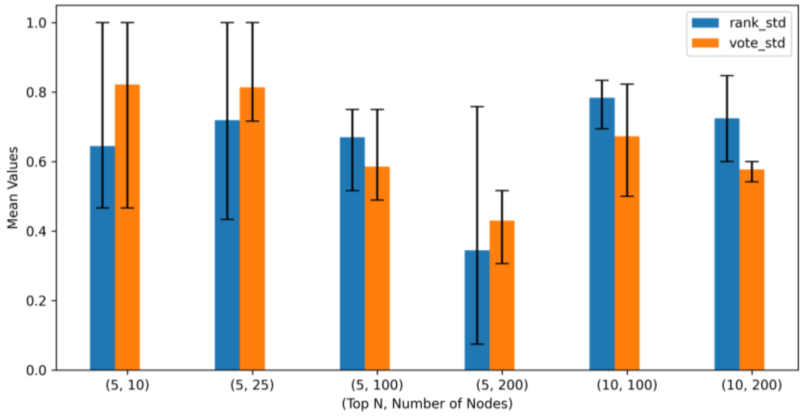

Figure 4. Comparing the graph metrics to the MC results for various; Graph sizes and top 'N's

Cohen's Kappa scores, computed for the different graph sizes, provide crucial insights into how reliably the centrality measures perform across the scales. These scores are shown in Figure 6, the bars show the Cohen's Kappa between the rank and sensitivity methods (rank_std) and the voting and sensitivity methods (vote_std), the bar shows the mean result with the error bars representing the maximum and minimum over 3 runs. This offers a clear comparison of agreement levels between different graph sizes. Although not originally planned for in the study, when the number of components got very large, the results became significantly worse. This could be caused by many factors, possibly that there are more nodes that are equally central. When this constraint is loosened, e.g. to the top 10, this problem is somewhat alleviated. Predicting the alignment between top N and graph size or topology would be an interesting piece of future work, although is outwith the scope of this study.

Parallel to the consistency analysis, our study examined the computational demands across different graph sizes. We observed that run times increase with larger graph sizes. However as can be seen in Table 4 this is imperceptible, without the aid of the timing function of Jupyter. Run times for the Monte Carlo were not recorded accurately, but the times scaled exponentially with graph size. Going from approximately 30 minutes for the 10-component example, to over 12 hours for the 200-component example. Although absolute runtimes will vary greatly between systems, the relative difference between these two results emphasises the ability of the graph-based methods to perform rapid, useful, analysis.

Table 4. Mean centrality run times

	10 Components	25 Components	100 Components	200 Components
Mean run time	23ms	25ms	47ms	51ms

4. Discussion

In this study, we delve into the application of graph theory in assembly analysis, assessing its impact on and contributions to the field of tolerance analysis. Our investigation not only bridges graph-based methods with traditional approaches but also provides insights into how these methodologies might coexist and complement each other in industrial applications.

Agreement Between Graph Analysis and Sensitivity: The good agreement between graph and sensitivity analyses, as indicated by the Cohen's Kappa scores, bridges traditional and graph-based approaches in assembly modelling. This demonstrates the effectiveness of graph theory in such applications. Of particular value here is the relative run-time and computational cost of graph methods compared to more traditional approaches. With a reduction in computation of over 99% over the MC analysis, graph methods show substantial potential for efficiency gain.

https://doi.org/10.1017/pds.2024.195 Published online by Cambridge University Press

Performance of Combined Metrics: For each scenario, the aggregate scores equalled or outperformed, the individual metrics. As mentioned previously, this can be attributed to the perspective these combined metrics offer. Different centralities favour different types of 'central' components within the assembly graph - for example, degree centrality identifies well-connected components, for example a chassis, while betweenness centrality is effective in recognising components crucial for through connections.
Importantly, the application of these combined centrality metrics to the entire network suggests their broader applicability in analysing any tolerance chain within an assembly. The aggregate approach holds has the potential to identify key components across diverse assembly configurations. However, to fully harness this potential, further refinement of the is necessary. Future work will focus on fine-tuning this methodology, ensuring it can accurately capture and reflect the nuanced interplay of different tolerance chains in various assembly models. Such advancements would substantiate the broader applicability and utility of our approach, particularly in complex, real-world assembly scenarios.
Sensitivity to Graph Topology: Our findings also reveal that all aspects of the method are sensitivity to graph topology, indicating that the structure of the graph significantly influences the results. This sensitivity underscores the importance of considering different assembly configurations and sizes, as they are likely to yield varying levels of agreement with traditional methods. The potential generalisability of the use of centrality metrics, as mentioned earlier, is a promising area for further investigation. Understanding the relationship between graph topology and tolerance analysis could lead to more tailored and accurate assembly analyses in future work.
Computational Considerations: Our analysis on computational efficiency is particularly relevant for industrial-scale assemblies, especially evident in our 100-200 node graph examples. These larger graphs are representative of the complexity found in industrial settings, underscoring the practical applicability of our approach (Sarh, 1998). Despite the slight increase in computational demand with larger graph sizes, our method remains significantly more computationally efficient compared to traditional simulations, as well as those employed by industry. 3DCS, a cutting edge tolerance analysis software, can take hours and days to perform its analysis (3DCS, 2017). This efficiency, offering rapid analysis in complex assembly scenarios, is crucial for industrial applications where time is often at a premium.

4.1. Limitations of current approach

This study, while insightful, has several limitations that pave the way for future research. Several are mentioned throughout yet one key limitation is the need to enhance the realism of our model for industrial applications. Understanding how graph-based analysis would integrate with existing industrial methods could offer a complementary tool, enhancing current practices in assembly analysis. As part of our ongoing collaboration with GKN Aerospace we hope to apply these methods in a this industrially significant case study. By addressing these limitations, future iterations of this study could not only refine our understanding of assembly models using graph theory but also bridge the gap between theoretical research and practical, industrial applications.

5. Conclusion

Our study has demonstrated the viability of graph theory as a potent tool in assembly modelling, showcasing its capability to align with Monte Carlo sensitivity analysis. We successfully applied centrality measures to graphs, representing 2D assemblies, to identify critical components. This approach, while vastly more computationally efficient, also revealed a sensitivity to graph topology. This alongside the possibilities mentioned throughout, will be interesting avenues for future exploration. The findings reinforce the potential of graph-based methods to augment traditional assembly analysis techniques, promising a more efficient, accurate, and insightful approach to understanding and managing complex assemblies in critical industrial sectors.

Acknowledgements

The authors would like to thank GKN Aerospace for their continued support of this project.

References

3DCS, B.R., 2017. URL https://blog.3dcs.com/run-tolerance-analysis-stacks-two-three-four-times-faster (accessed 11.24.23).

Akroyd, J., Mosbach, S., Bhave, A., Kraft, M., 2021. Universal Digital Twin - A Dynamic Knowledge Graph. Data-Centric Eng. 2, e14. https://doi.org/10.1017/dce.2021.10

Bakker, O.J., Popov, A.A., Ratchev, S.M., 2017. Variation Analysis of Automated Wing Box Assembly. Procedia CIRP 63, 406–411. https://doi.org/10.1016/j.procir.2017.02.034

Ballantyne, R., Snider, C., McClenaghan, A., Nassehi, A., 2023. On Applying Network Theory to Assembly Analysis. Procedia CIRP 120, 1552–1557. https://doi.org/10.1016/j.procir.2023.09.213

Cohen, J., 1960. A Coefficient of Agreement for Nominal Scales. Educ. Psychol. Meas. 20, 37–46. https://doi.org/10.1177/001316446002000104

Das, K., Samanta, S., Pal, M., 2018. Study on centrality measures in social networks: a survey. Soc. Netw. Anal. Min. 8, 13. https://doi.org/10.1007/s13278-018-0493-2

Farahani, F.V., Karwowski, W., Lighthall, N.R., 2019. Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review. Front. Neurosci. 13, 585. https://doi.org/10.3389/fnins.2019.00585

Gunji, B.M., Sharma, D., Sathyajith, S., Patil, K., Mutra, R.R., 2022. Optimal assembly sequence generation for flexible and rigid parts product using stress information matrix. (ICCEM 2021), Shah Alam, Selangor, Malaysia, https://doi.org/10.1063/5.0103525

ISO 286 - Geometrical Product Specifications (GPS) 2010. https://www.iso.org/standard/44784.html

Kosec, P., Skec, S., Miler, D., 2020. A comparison of the tolerance analysis methods in the open-loop assembly. Adv. Prod. Eng. Manag. 15, 44–56. https://doi.org/10.14743/apem2020.1.348

Koutrouli, M., Karatzas, E., Paez-Espino, D., Pavlopoulos, G.A., 2020. A Guide to Conquer the Biological Network Era Using Graph Theory. Front. Bioeng. Biotechnol. 8, 34. https://doi.org/10.3389/fbioe.2020.00034

Li, C., Hou, W., 2021. Analysis of Assembly Tolerance Based on Assembly Constraint Information Model. Math. Probl. Eng. 2021, 1–18. https://doi.org/10.1155/2021/7438966

Majeed, A., Rauf, I., 2020. Graph Theory: A Comprehensive Survey about Graph Theory Applications in Computer Science and Social Networks. Inventions 5, 10. https://doi.org/10.3390/inventions5010010

Networkx Documentation 2023.

Sarh, B., 1998. Wing Structural Assembly Methodology p. 982156. https://doi.org/10.4271/982156

Shen, Z., Ameta, G., Shah, J.J., Davidson, J.K., 2005. A Comparative Study Of Tolerance Analysis Methods. J. Comput. Inf. Sci. Eng. 5, 247–256. https://doi.org/10.1115/1.1979509

Sun, X., Bao, J., Li, J., Zhang, Y., Liu, S., Zhou, B., 2020. A digital twin-driven approach for the assembly-commissioning of high precision products. Robot. Comput.-Integr. Manuf. 61, 101839. https://doi.org/10.1016/j.rcim.2019.101839

Watson, J., Hermans, T., 2019. Assembly Planning by Subassembly Decomposition Using Blocking Reduction. IEEE Robot. Autom. Lett. 4, 4054–4061. https://doi.org/10.1109/LRA.2019.2929995

Zhang, T., et al 2022. Assembly Configuration Representation and Kinematic Modeling for Modular Reconfigurable Robots Based on Graph Theory. Symmetry 14, 433. https://doi.org/10.3390/sym14030433

Zhou, B., Bao, J., Chen, Z., Liu, Y., 2022. KGAssembly: Knowledge graph-driven assembly process generation and evaluation for complex components. Int. J. Comput. Integr. Manuf. 35, 1151–1171. https://doi.org/10.1080/0951192X.2021.1891572