AI-Based Sequence Determination in Manufacturing Systems through MST Graph Optimization

P-ISSN: 1680-9300
E-ISSN: 2790-2129
Vol. (26), No. (2)
pp. 113-124

[1] Soud M. Amen, [2] Salim A. Saleh, [3] Harith Hamoodat

[1] Institute of Technical Management – Nineveh, Northern Technical University, Mosul, Iraq.

[2] Tikrit University, Tikrit, Iraq.

[3] Technical College of Management, Northern Technical University, Mosul, Iraq.

Abstract:

In the realm of computational optimization, the Minimum Spanning Tree (MST) remains a cornerstone algorithm for solving complex networking and sequence-related problems. This research explores a novel application of MST within the field of Automated Process Planning (APP). By leveraging the "Delta Volume" decomposition method, we represent the material removal process as a structured graph of interdependent data nodes. The core contribution of this work lies in the development of a specialized algorithmic framework, implemented via MATLAB, which treats machining volumes as a discrete search space. Our approach utilizes MST-based logic to navigate this space and determine the most efficient execution sequence, effectively transforming a traditional manufacturing challenge into a graph-theory optimization problem. By integrating these computational techniques, the study provides a robust automated solution that bridges the gap between geometric modeling and intelligent process sequencing, ensuring both precision and algorithmic efficiency.

Keywords: MST Algorithm, Delta Volume, Machining Operations Sequence, AI-Enhanced Graph Optimization, Artificial Intelligence in Combinatorial Optimization, Automated Process Planning (APP).

1. Introduction

Process planning is essential for the design and operation of manufacturing systems. It defines the necessary steps and resources to machine a part. A good plan should increase production rates and quality while reducing costs and delays (Babic et al., 2008). During this process, raw materials are converted into finished parts by removing specific material volumes, often called the "delta volume." This delta volume represents the difference between the initial stock and the final part (Lee et al., 2022; Leng et al., 2019).

In recent decades, many efforts have been made to automate process planning within integrated manufacturing systems. However, establishing operational procedures still relies heavily on manual experience (Tao et al., 2018; Zhong et al., 2017). These manual activities require skilled individuals known as process planners (Lu et al., 2020). Since experienced designers are few, different planners often produce different, and sometimes inconsistent, production plans.

Journal of Prospective Researches
Vol. (26), No. (2)
The paper was received on March 16, 2026; accepted on April 22, 2026; and published on May 7, 2026.
Corresponding author's e-mail: s23@ntu.edu.iq

10.61704/pr.615

Computer-Aided Process Planning (CAPP) acts as a bridge between CAD and CAM. To work effectively, CAPP must interpret the part through its features. Therefore, feature recognition is a vital part of integrating CAD and CAM systems (Monostori, 2014). Deep neural networks have recently been utilized for the recognition of features in manufacturing operations to provide for an increased linkage between 3D CAD applications and an improved automated understanding of manufacturing (Yeo et al., 2021). This would reduce their dependence on hand-coded definitions of features, as well as increase the efficiency of extracting features from intelligent manufacturing environments. These approaches help automate the identification of machining features directly from CAD models, reducing dependence on manually defined rules and improving the efficiency of feature extraction in intelligent manufacturing environments. It transforms CAD data into a manufacturing model. Several techniques exist for this, such as hint-based, graph-based, knowledge-based, and volumetric decomposition methods.

Hint-based methods are effective at identifying interacting features (Tao et al., 2019; Wang et al., 2024). However, they have drawbacks (Kausik et al., 2025). These algorithms match predetermined features to tool motion traces. If a complex feature is not in the library, the algorithm may fail to recognize it (Tao et al., 2018).

Graph matching is often used in manufacturing feature recognition and CAD/CAM integration (Xiao et al., 2023). A major criticism of these techniques is that they are typically not very good at identifying overlapping features due to difficulties establishing the relationship between the face patterns and geometry. Recent research has used deep learning techniques to detect machining features from B-rep models to enhance feature extraction and advance the intelligence of manufacturing processes. Researchers explored using deep learning to improve safety, assurance, and intelligent decision-making through digital twin frameworks in sophisticated manufacturing (Wang et al, 2024).

Recent deep learning techniques, specifically BrepMFR, have been shown to improve machining feature recognition in B-rep models by using domain adaptation techniques to enhance recognition accuracy for parts that have complex geometries (Zheng et al., 2024). These methods enable improved extraction of machining features from highly detailed CAD representations and help to overcome problems associated with combining, overlapping or otherwise interacting machining features. As a result, deep learning-based identification techniques will assist with smarter CAD / CAM integration within modern manufacturing systems. Two primary approaches are commonly used in volumetric decomposition: cell-based and convex hull methods (Afif & Sarhan, 2025; Babic et al., 2008; Monostori, 2014).

These cells are then combined to form volumes. Previous studies used this approach to generate multiple feature interpretations (Leng et al., 2019). They suggested that sequencing these volumes correctly can lead to a nearly optimal plan. However, combining these cells is computationally complex (Babic et al., 2008), and the method can suffer from "combinatorial explosion" when the number of cells is large.

According to Afif and Sarhan (2025), Kusiak first used volumetric decomposition in 1985. He represented removed volumes based on their geometric shapes, allowing for multiple ways to represent the total volume. This technique divides the material into sub-volumes (cells) that can be removed in different contexts. In this approach, a specific volume is defined as a set of sub-volumes, like $V1= \{v1, v2, v3\}$. A relational matrix is then used to show the connection between these sub-volumes.

This paper proposes an approach using graph theory to show the relationships between these sub-volumes (cells).

By using the Minimum Spanning Tree (MST) algorithm, we can determine the best sequence for machining stages to minimize production costs. In this study, "AI" refers to structured decision support using algorithmic logic, rather than machine learning. The main contribution is applying MST to machining sequence planning using the delta volume representation. This method remains simple and direct, reducing unnecessary transitions between steps and providing a structured solution for process planning.

10.61704/pr.615

2. Problem Definition

Depending on the specific domain, the word "feature" might denote a variety of meanings. For instance, in manufacturing it refers to notch section, slots, holes, and pockets in design, while in inspection as reference or a datum point on item to be inspected.

Workable processing plan, requires evaluation of part design data, machining features, decomposition of the volume of material to be removed (delta volume), create precedence restrictions, and sequence machining features, choose manufacturing procedures, equipment, and tools. Process planning involves identifying production attributes, which is a crucial step. The technique employed in the references was based on defining a particular volume that stood in for a collection of sub-volumes (cells), let V1=v1, v2, v3 be that volume. As a result, the displacement of the set of vi volumes is included in each processing technique. A relational matrix between the sub-volumes and their constituents represents the deleted volume.

Previous studies took the model and basic manufacturing features of product (such as elementary volumes or cells) into account. To obtain the chosen final shape of the part, (often by the design engineer), the delta volume V (see Fig. 1) must be removed from stock. By drawing planes next to each surface of the part, delta volume can be divided into volumes vi, {where i= 1, 2, ...,5}, (see Fig. 2). Multi pass machining be done in one or more tool passes. It might also be seen as creating extra planes, like p1 (see Fig. 3), to remove such volume. The extra plane p1 divides the volumes of v1, ..., and v5 into additional, smaller volumes called e1, e2, ..., and e8 (see Fig. 3). In order to develop machinable volumes, elementary volumes that can generate different interpretations must be combined.

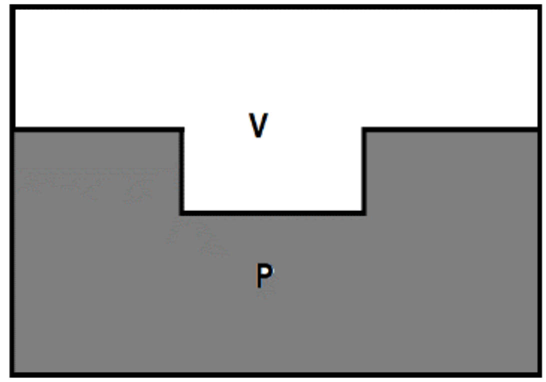


Figure 1. Volume V To Be Removed and Part P

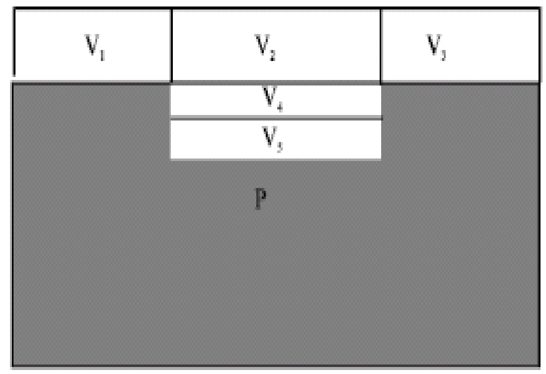


Figure 2. Delta Volume V Divided Into (V1, ..., & V5)

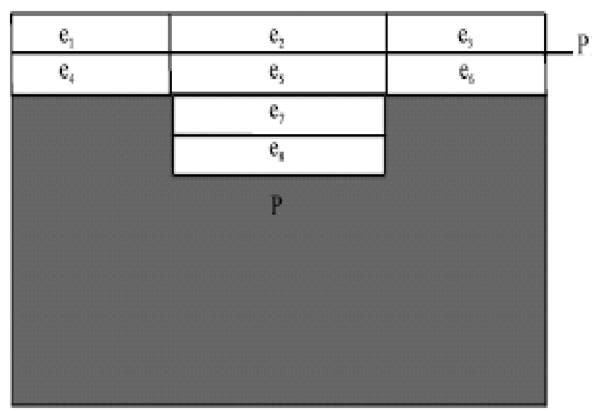


Figure 3. Division Of Volumes (Vi) Into Elementary Volumes (E1, E2, ..., & E8) To Achieved Multi-Pass Requirements

2.1 Volume Representation by Graphs

The majority of expert process planning systems were built using a collection of predefined components (part) features. A collected group of geometric objects, such as surfaces, curves, and points, known as component (part) features, have some value for process planning. Consider a few of the part's geometrical features using the attributed adjacency graph (AAG), as displayed in fig. (4). Each node and each arc in an AAG represent a face and an edge of the part, respectively. Additionally, if the angle formed at the edge by two faces is

10.61704/pr.615

concave (convex) an attribute value 0 (1) is assigned to the arc. Fig. (4) illustrates these geometrical features of the part and its associated AAG.

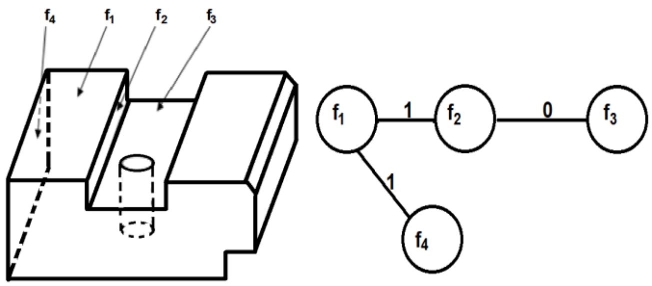


Figure 4. Sample Part with Geometric Features and AAG Representation of The Step and Slot

However, the relationship between the faces has evolved into the relationship between the volumes. Kusiak (Kusiak, 1990) in 1987 established a relationship between the main operating volumes (Machinable Volumes Vj's), based on the principle of dividing them into groups of partial volumes (Elementary Volumes vi's), as shown in (Fig. 5). Division process will convert the Machinable Volumes Vj's into groups of Elementary Volumes vi's, i.e., machinable volumes V1= {v2, v7, v8, v10}, V3= (v7), V4= {v8, v10}, and so on. He represented it with the Incidence Matrix shown in (Fig. 6) in order to solve it by Integer Programming method and find the optimal cost the produced part.

Depending on the juxtaposition relationship between the volumes that must be removed from the raw piece during the machining process necessary to produce the piece shown in (Fig. 4) above. For the part shown in (Fig. 5), an adjacency relationship between the Elementary Volumes to be removed (v1, …, …, v21) can be employed to show their relationship. After that, it is converted into an adjacency matrix between volumes called (nodes) with different arcs values, as shown in (Fig. 7), which is easy to process mathematically and programmatically. If there is no direct adjacent relationship between any two nodes, the arc takes a very large value (∞), while the arc between the two directly adjacent nodes takes the value (1). In this study, equal weights are used to simplify the model and focus on the structure of the solution. In real applications, these weights can represent factors such as machining time, cost, or tool changes. But in the node, itself, the arc takes the value (0), which means that the operating tools did not move from its position and the part state did not change.

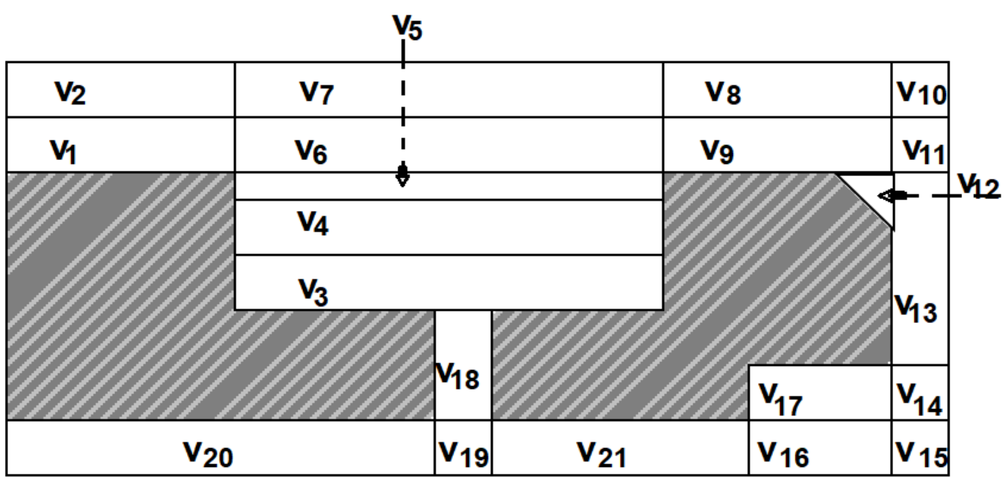


Figure 5. Two-Dimensional View of The Part in Fig. 4 With Volumes V1 To V21 To Be Removed

10.61704/pr.615

		Machinable Volume																					
		V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}
Elementary volume	v_1					1						1												
	v_2	1	1																					
	v_3								1															
	v_4							1																
	v_5						1																	
	v_6						1					1												
	v_7	1		1																				
	v_8	1			1																			
	v_9											1												1
	v_{10}	1			1										1	1								
	v_{11}											1			1	1								1
	v_{12}																		1					
	v_{13}														1			1						
	v_{14}														1		1				1			
	v_{15}												1	1	1		1			1				
	v_{16}												1	1						1				
	v_{17}																				1			
	v_{18}									1	1													
	v_{19}										1		1									1		
	v_{20}												1	1								1	1	
	v_{21}												1	1								1	1	

Figure 6. Kusiak; Incidence Matrix

i/j	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}
v_0	0	∞	1	∞	∞	∞	∞	∞	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	∞	1	∞
v_1	∞	0	∞	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_2	∞	1	0	∞	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_3	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	∞	∞	∞
v_4	∞	∞	∞	1	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_5	∞	∞	∞	∞	1	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_6	∞	1	∞	∞	∞	1	0	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_7	∞	∞	1	∞	∞	∞	1	0	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_8	∞	∞	∞	∞	∞	∞	∞	1	0	1	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_9	∞	∞	∞	∞	∞	∞	1	∞	∞	0	∞	1	1	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_{10}	∞	∞	∞	∞	∞	∞	∞	∞	1	∞	0	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_{11}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	1	1	∞	∞	∞	∞	∞	∞	∞	∞
v_{12}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞
v_{13}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	1	0	1	∞	∞	∞	∞	∞	∞	∞
v_{14}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	1	∞	1	∞	∞	∞	∞
v_{15}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	1	1	∞	∞	∞	∞
v_{16}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	1	∞	∞	∞	1
v_{17}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
v_{18}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
v_{19}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	1	1
v_{20}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	0	∞
v_{21}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	∞	∞	1	∞	0

Figure 7. Adjacency Matrix

3. Minimum Spanning Tree (MST)

MST is used in this study because it provides a direct solution with minimum total cost without the need for repeated iterations or parameter tuning. Compared to methods like Genetic Algorithms, it is simpler and faster, which makes it suitable for practical use in machining sequence planning.

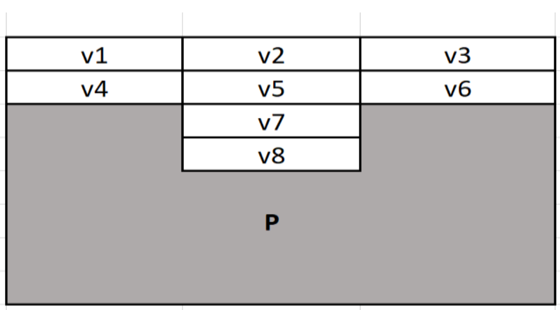
Consider a mixed connected network (G= {V, E, A} given in Fig. 8) consists of nodes (V = 8 = {v1, …, …, v8}), undirected edges (E = 4= {e1, e2, e3, e4}), and directed arcs (A = 5= {a1, a2, a3, a4, a5}). Keeping only nodes and deleting all edges and arcs in order to creating MST gradually. Now begin to construct or "grow" the "tree" by gradually adding one edge (ei) or one arc (ai) from the original network. Putting the first edge (e1) at the junction between the nodes (v1 & v2). Then, adding new edge (a1 or e2) to link one of the linked nodes (v1 or v2) with a new node of (v3 or v4) that has not yet been linked with any of others. By adding an edge in this manner, a cycle is prevented and it is made sure that there are 1 more linked node than edges. Each additional edge (or "arc") creates a larger tree, which is a linked network devoid of undirected cycles. Such process comes to an end Once the edge or "arc" number "n – 1" is added because the resulting tree spans and links all n nodes. A linked network devoid of undirected cycles and joined all n nodes is known as a spanning tree, and this tree falls under that definition. Each spanning tree contains exactly "n – 1" edges being the minimum needed to create a linked network, and the maximum to prevent undirected cycles. Thus, the MST with its possible forms will appear in (Fig. 9).

The shortest path problem and the MST problem share some characteristics. In all situations, a connected network is taken into account, and the data given contains some established indicator non-negative values, as (km, dollar, hr., etc.), related to each edge. A set of edges with the shortest overall length among all edges that meet a specific requirement must be chosen for both problems. These characteristic states that the chosen edges for the shortest path problem must offer a route between the origin and the destination. The necessary characteristic for the MST problem is the chosen edges has to offer a route between any node and another.

MST issue is summarized as follows:

- Starting with nodes without edges, and knowing the links and their positive lengths if they are included in the network.
- The desire here is to secure the network design requirements by introducing enough links to satisfy the path requirements between any two nodes.
- The goal is to fulfill this need while minimizing the overall length of edges entering the network.

A network of n nodes needs only (n - 1) edges to establish a route between any two nodes. Using extra edges will unnecessarily increase the length of selected links. The network of n nodes with (n-1) edges selected in specified way will form a spanning tree with minimum total length of its links.



	v1	v2	v3	v4	v5	v6	v7	v8
v1	0	1	∞	1	∞	∞	∞	∞
v2	1	0	1	∞	1	∞	∞	∞
v3	∞	1	0	∞	∞	1	∞	∞
v4	∞	∞	∞	0	1	∞	∞	∞
v5	∞	∞	∞	1	0	1	1	∞
v6	∞	∞	∞	∞	1	0	∞	∞
v7	∞	∞	∞	∞	∞	∞	0	1
v8	∞	∞	∞	∞	∞	∞	∞	0

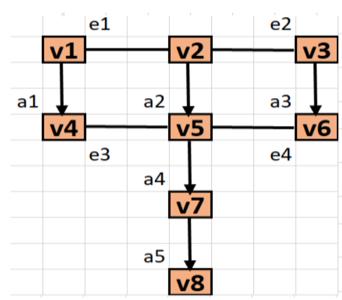


Figure 8. Part P with Eight Elementary Volumes and its Dependency Matrix and Mixed Graph G= (V, E, A)

10.61704/pr.615

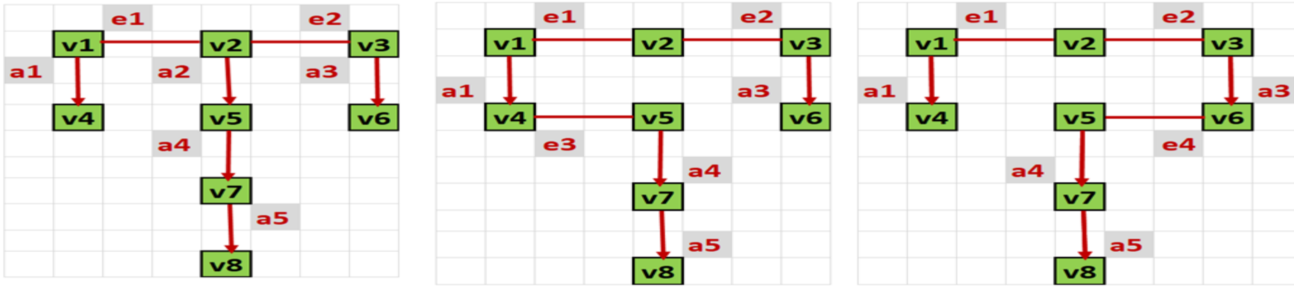


Figure 9. Most Possible Spanning Trees

4. Some Minimal Spanning Tree Applications

The applications of the minimum spanning tree can be summarized in the following areas:

- Design of communication networks between nodes of (optical fibers, computers, leased phones, television wire, etc).
- Transport network design with minimum overall cost of providing links (railways, roads, etc.).
- The layout of the high-voltage power transmission network.
- To reduce overall length of the wire, create a network of a computer systems and other electrical equipment;
- Create pipeline networks to link various locations.
- This paper adds a new application, namely: using it to determine the optimal sequence of machining processes needed to produce a specific piece, followed by the lowest possible production cost for manufacturing.

With the rapid development of information transmission methods, applications of communication network design have received special attention. The classic application of the MST is widely used with the introduction of sufficient links to provide the paths for each pair of nodes involved in the design of the telecommunications network. Finding the MST for each communications network is critical to optimizing its infrastructure and reducing its multi-million-dollar costs today.

5. The MST Algorithm

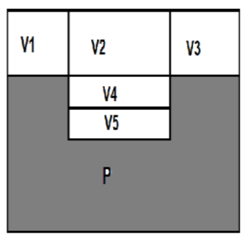
1) Start with any arbitrarily defined node and link it to the closest node.
2) Locate the nearest disconnected node and link it to a linked node. Up till every node has been joined, repeat this process.
3) Relations may be broken arbitrarily, but the algorithm has to still produce an optimum result when determining the closest distinct node (step 1) or the closest unconnected node (step 2). Such relations are an indication that there can, but is not necessary, more than one optimum solution. All these optimum solutions can be determined by following all the ways to break relations with their conclusion.

6. Machining Examples

For the purposes of this paper, a MATLAB R2020a program was built to implement the proposed algorithm on several machining cases needed to produce products of different shapes. Compared to other approaches such as Genetic Algorithms and heuristic methods, the MST-based approach provides a direct and structured solution without the need for repeated iterations or parameter tuning. This makes it easier to implement and faster in generating a valid machining sequence, although other methods may provide more flexibility in more complex cases. The proposed method does not require iterative optimization or parameter tuning, unlike Genetic Algorithms, which may need multiple runs to reach a stable solution. The program was implemented to calculate the MST of the removed volumes in order to manufacture the required parts. Determining the MST for manufacturing processes indicates to the optimal sequence of machining processes required to produce the required parts:

- **First Example**

In order to evaluate the performance of the proposed method, the problem shown in (Fig. 10) is the part P with decomposing the magnitude of delta V into sub-volumes (v1, v2, v3, v4 and v5) and its input data matrix for MATLAB program. The program processed the input data and resolved it to produce the output matrix and the represented MST with sum of 4 units of arcs (n-1 = 5-1), which is shown in (Fig. 11). The obtained sequence minimizes the number of required connections to the theoretical minimum (n−1), which reduces unnecessary transitions compared to non-optimized machining sequences. The structure of the MST ensures that all nodes are connected using (n−1) edges, which reduces unnecessary transitions between machining steps. This helps in organizing the sequence in a more efficient way and avoids redundant operations. Whereas, The MST represents the path of the machining sequence to produce the desired part (P) in (Fig. 10). This example is simplified to clearly demonstrate the proposed method. However, it reflects real machining logic and can be extended to real industrial applications by including additional factors such as machining time, cost, tool changes, and production constraints. The same approach can be applied to real parts with more complex geometries and larger sets of machining features.



#	1	2	3	4	5
S	V1	V2	V2	V3	V4
T	V2	V3	V4	V2	V5
Wt	1	1	1	1	1

Figure 10. Decomposition of Delta Volume V into V1, V2, V3, V4 and V5 with their Input Representation into MATLAB Program Table

T.Edges

1 EndNodes		2 Weight
'V1'	'V2'	1
'V2'	'V3'	1
'V2'	'V4'	1
'V4'	'V5'	1

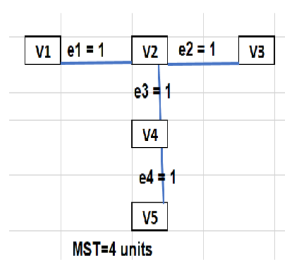
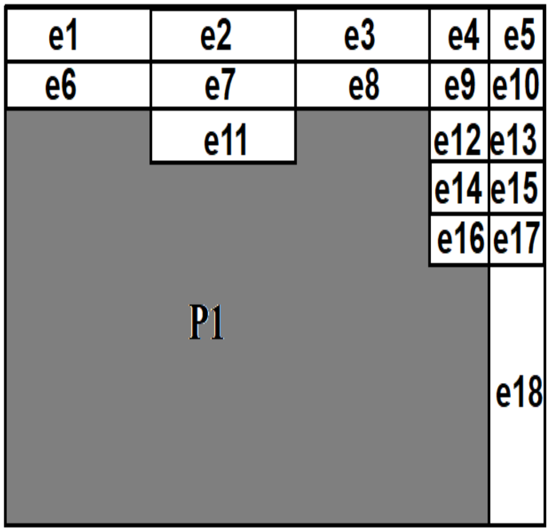


Figure 11. Output of MATLAB Program and MST Representation

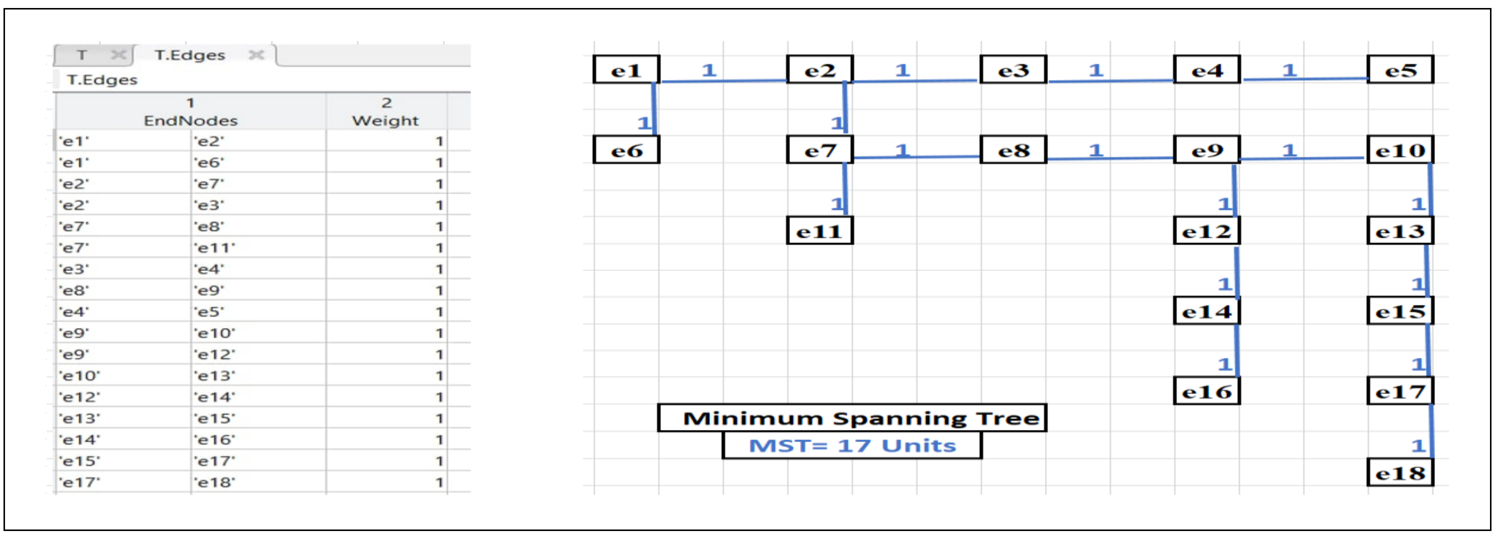
- **Second Example**

The problem shown in (Fig. 12) representing Part P1 with related Elementary volumes and its input data matrix for the MATLAB program. The program processed the input data and resolved it to produce the output matrix and the represented MST with sum of 17 units of edges (n-1 = 18-1), which is shown in (Fig. 13). Whereas, the MST represents the path of the sequence of machining operations to produce the part (P1) required in (Fig. 12).



#	S	T	Wt	#	S	T	Wt	#	S	T	Wt	#	S	T	Wt
1	e1	e2	1	13	e5	e10	1	25	e9	e10	1	37	e14	e12	1
2	e1	e6	1	14	e6	e1	1	26	e9	e12	1	38	e14	e15	1
3	e2	e1	1	15	e6	e7	1	27	e10	e5	1	39	e14	e16	1
4	e2	e7	1	16	e7	e2	1	28	e10	e9	1	40	e15	e13	1
5	e2	e3	1	17	e7	e6	1	29	e10	e13	1	41	e15	e14	1
6	e3	e2	1	18	e7	e8	1	30	e11	e7	1	42	e15	e17	1
7	e3	e8	1	19	e7	e11	1	31	e12	e9	1	43	e16	e14	1
8	e3	e4	1	20	e8	e3	1	32	e12	e13	1	44	e16	e17	1
9	e4	e3	1	21	e8	e7	1	33	e12	e14	1	45	e17	e15	1
10	e4	e9	1	22	e8	e9	1	34	e13	e10	1	46	e17	e16	1
11	e4	e5	1	23	e9	e4	1	35	e13	e12	1	47	e17	e18	1
12	e5	e4	1	24	e9	e8	1	36	e13	e15	1	48	e18	e17	1

Figure 12. Part P1 With Eighteen Elementary Volumes and Table of Their Input Representation into MATLAB Program


T.Edges

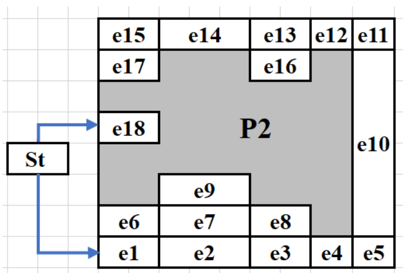
1 EndNodes		2 Weight
'e1'	'e2'	1
'e1'	'e6'	1
'e2'	'e7'	1
'e2'	'e3'	1
'e7'	'e8'	1
'e7'	'e11'	1
'e3'	'e4'	1
'e8'	'e9'	1
'e4'	'e5'	1
'e9'	'e10'	1
'e9'	'e12'	1
'e10'	'e13'	1
'e12'	'e14'	1
'e13'	'e15'	1
'e14'	'e16'	1
'e15'	'e17'	1
'e17'	'e18'	1

Figure 13. Output of MATLAB Program and MST Representation

- **Third Example**

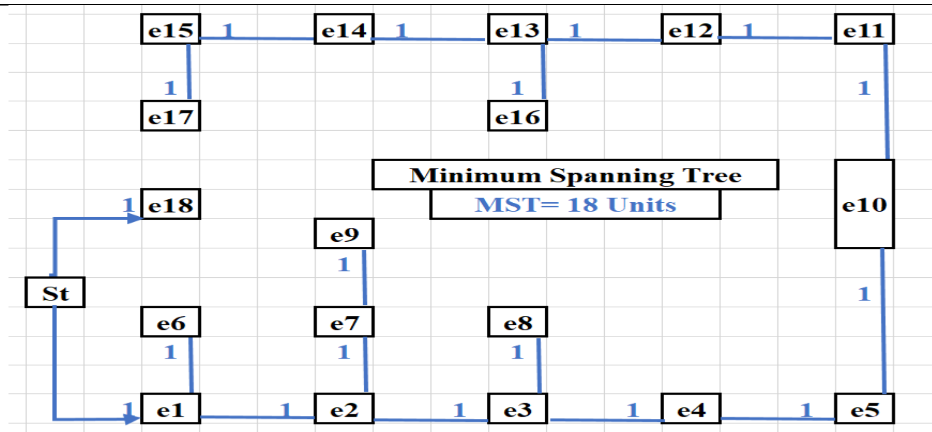
The problem shown in (Fig. 14) representing Part P2 with Elementary volumes and its input data matrix for the MATLAB program. Since the node (e18) is isolated and not connected or adjacent to any of the other nodes, we have to use an added proposed start node (St) to ensure that communication between all nodes goes smoothly. The program processed the input data and resolved it to produce the output matrix and the represented MST with sum of 18 units of edges (n-1 = 19-1), which is shown in (Fig. 15). Whereas, the MST represents the path of the sequence of machining operations to produce the part (P2) required in (Fig. 14).

10.61704/pr.615



#	S	T	Wt	#	S	T	Wt	#	S	T	Wt	#	S	T	Wt
1	St	e_1	1	9	e_4	e_5	1	16	e_8	e_7	1	23	e_{13}	e_{12}	1
2	St	e_{18}	1	10	e_5	e_4	1	17	e_{10}	e_5	1	24	e_{13}	e_{16}	1
3	e_1	e_2	1	11	e_5	e_{10}	1	18	e_{10}	e_{11}	1	25	e_{13}	e_{14}	1
4	e_1	e_6	1	12	e_6	e_7	1	19	e_{11}	e_{10}	1	26	e_{14}	e_{13}	1
5	e_2	e_3	1	13	e_7	e_6	1	20	e_{11}	e_{12}	1	27	e_{14}	e_{15}	1
6	e_2	e_7	1	14	e_7	e_8	1	21	e_{12}	e_{11}	1	28	e_{15}	e_{14}	1
7	e_3	e_4	1	15	e_7	e_9	1	22	e_{12}	e_{13}	1	29	e_{15}	e_{17}	1
8	e_3	e_8	1												

Figure 14. Part P2 With Start Node (St) Beside Eighteen Elementary Volumes and Table of Their Input Representation into MATLAB Program



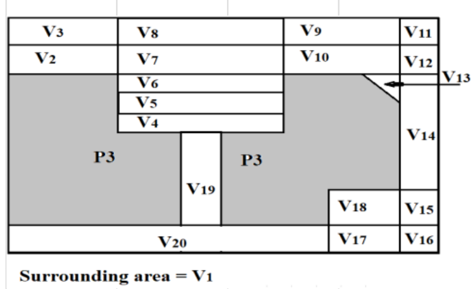
T.Edges

	1 EndNodes		2 Weight
1	'St'	'e1'	1
2	'St'	'e18'	1
3	'e1'	'e2'	1
4	'e1'	'e6'	1
5	'e2'	'e3'	1
6	'e2'	'e7'	1
7	'e3'	'e4'	1
8	'e3'	'e8'	1
9	'e7'	'e9'	1
10	'e4'	'e5'	1
11	'e5'	'e10'	1
12	'e10'	'e11'	1
13	'e11'	'e12'	1
14	'e12'	'e13'	1
15	'e13'	'e16'	1
16	'e13'	'e14'	1
17	'e14'	'e15'	1
18	'e15'	'e17'	1

Figure 15. Output of MATLAB Program MST Representation

- **Fourth Example**

The problem shown in (Fig. 16) representing Part P3 with Elementary volumes, added surrounding Node (v1) and its input data matrix for the MATLAB program. Kuziak (Kusiak, 1990) solved the problem but with different technique. The program processed the input data and resolved it to produce the output matrix and the represented MST with sum of 19 units of edges (n-1 = 20-1), which is shown in (Fig. 17). Whereas, the MST represents the path of the sequence of machining operations to produce the part (P3) required in (Fig. 16).



#	S	T	Wt	#	S	T	Wt	#	S	T	Wt
1	1	3	1	13	7	10	1	25	12	14	1
2	1	11	1	14	8	3	1	26	14	12	1
3	1	16	1	15	8	7	1	27	14	13	1
4	1	20	1	16	8	9	1	28	14	15	1
5	2	7	1	17	9	8	1	29	15	14	1
6	3	2	1	18	9	10	1	30	15	16	1
7	3	8	1	19	9	11	1	31	16	15	1
8	4	19	1	20	10	7	1	32	16	17	1
9	5	4	1	21	10	12	1	33	17	16	1
10	6	5	1	22	11	9	1	34	17	18	1
11	7	2	1	23	11	12	1	35	17	20	1
12	7	6	1	24	12	11	1				

Figure 16. Part P3 With Surrounding Node (V1) Beside Nineteen Elementary Volumes and Table of Their Input Representation into MATLAB Program

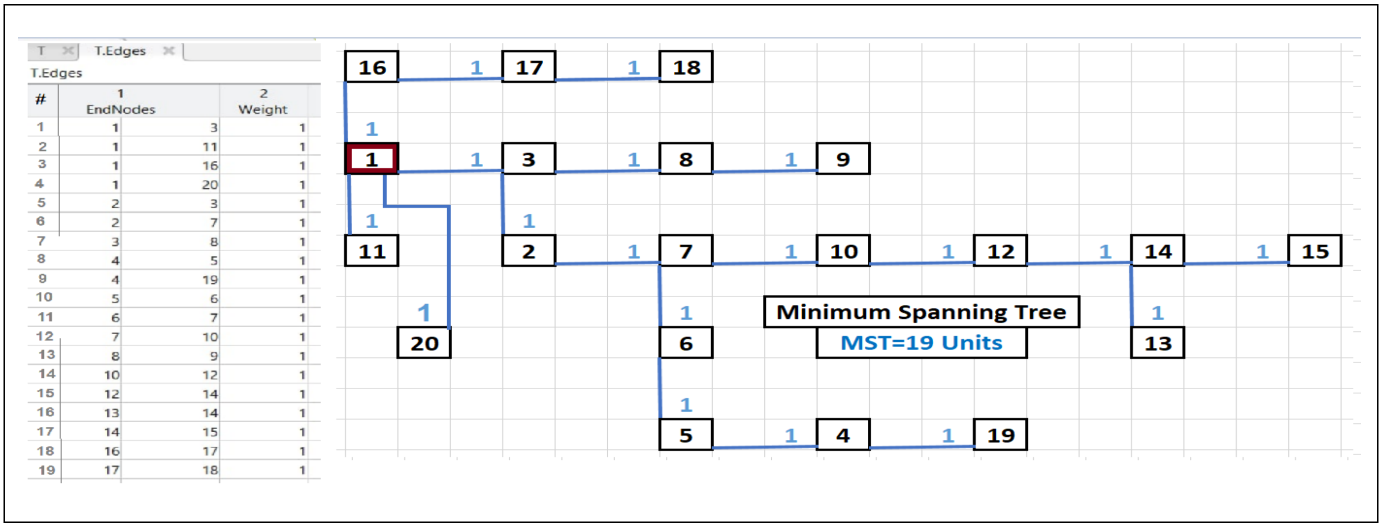


Figure 17. Output of MATLAB Program and MST Representation

7. Results and Discussions

The examples presented in this study show that the proposed MST-based method can generate a clear and valid machining sequence using the minimum number of connections (n−1). In all cases, the method helped in organizing the order of operations in a simple way and reduced unnecessary movements between steps. The results are consistent across the different examples, even though they are simplified cases. This indicates that the method can be applied to more complex parts by including additional factors such as time, cost, and tool changes. Overall, the approach provides a practical and easy-to-use solution for machining sequence planning.

8. Conclusion

The generated sequence reduces redundant movements between operations, which can help reduce processing time and improve efficiency in practical applications. Based on the experimental results and the computational framework developed in this study, the following conclusions can be drawn:

- Graph-Based Data Representation: The study successfully demonstrates the feasibility of modeling the relationships between Machinable Volumes (Vj's) as a complex connected network. By representing these volumes as nodes within a Graph-based dependency matrix, we have transformed a physical manufacturing problem into a structured data environment suitable for advanced computational analysis.
- Algorithmic Optimization using MST: The application of the MST algorithm proved to be a highly efficient AI-driven optimization tool for sequence planning. While MST is traditionally used in network routing and electronic circuit design, its adaptation here provides an optimal, automated path for material removal, ensuring the lowest possible computational and production costs—a novel shift in the field of intelligent manufacturing.
- Automation & Logic Visualization: The research highlights the ease of graphical and logical representation of manufacturing processes. By using automated processing rules, the complex dependencies between volumes become computationally clear, facilitating a more intuitive design for Intelligent Process Planning systems.

- Integration with AI-CAPP Systems: The proposed method serves as a vital algorithmic bridge within Computer-Aided Process Planning (CAPP) software. It enhances the integration between CAD and CAM systems by providing an automated decision-making layer, which is essential for the next generation of smart factories.
- Data Integrity vs. Heuristic Penalty: Unlike many traditional AI models that rely on "penalty factors" or virtual assumptions, the MST-based approach utilizes real, deterministic data. This ensures higher reliability in the decision-support system, as it eliminates the bias often found in methods that depend on subjective heuristic penalties.

Future work will focus on using real industrial data, applying variable weights such as time and cost, and comparing the proposed method with other optimization techniques.

References

Afif, M. F., & Sarhan, A. A. D. (2025). Computer-aided process planning, digital twin, and smart manufacturing: Interconnections and integration in CNC machining processes. International Journal of Production Research, 63(24), 9593–9632. https://doi.org/10.1080/00207543.2025.2521717

Babic, B., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. Computers in Industry, 59(4), 321–337. https://doi.org/10.1016/j.compind.2007.09.001

Kausik, A. K., Bin Rashid, A., Baki, R. F., & Jannat Maktum, M. M. (2025). Machine learning algorithms for manufacturing quality assurance: A systematic review of performance metrics and applications. Array, 26, 100393. https://doi.org/10.1016/j.array.2025.100393

Lee, J., Lee, H., & Mun, D. (2022). 3D convolutional neural network for machining feature recognition with gradient-based visual explanations from 3D CAD models. Scientific Reports, 12(1), 14864. https://doi.org/10.1038/s41598-022-19212-6

Leng, J., Zhang, H., Yan, D., Liu, Q., Chen, X., & Zhang, D. (2019). Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. Journal of Ambient Intelligence and Humanized Computing, 10(3), 1155–1166. https://doi.org/10.1007/s12652-018-0881-5

Lu, Y., Liu, C., Wang, K. I.-K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing, 61, 101837. https://doi.org/10.1016/j.rcim.2019.101837

Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. Procedia CIRP, 17, 9–13. https://doi.org/10.1016/j.procir.2014.03.115

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. Journal of Manufacturing Systems, 48, 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006

Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186

Wang, S., Zhang, J., Wang, P., Law, J., Calinescu, R., & Mihaylova, L. (2024). A deep learning-enhanced digital twin framework for improving safety and reliability in human-robot collaborative manufacturing. Robotics and Computer-Integrated Manufacturing, 85, 102608. https://doi.org/10.1016/j.rcim.2023.102608

Xiao, Y., Zheng, S., Shi, J., Du, X., & Zhao, J. (2023). Knowledge graph-based manufacturing process planning: A state-of-the-art review. Journal of Manufacturing Systems, 70, 417–435. https://doi.org/10.1016/j.jmsy.2023.08.006

Yeo, C., Kim, B. C., Cheon, S., Lee, J., & Mun, D. (2021). Machining feature recognition based on deep neural networks to support tight integration with 3D CAD systems. Scientific Reports, 11, 22147. https://doi.org/10.1038/s41598-021-01313-3

Zheng, S., Shi, J., & Zhao, J. (2024). BrepMFR: Enhancing machining feature recognition in B-rep models through deep learning and domain adaptation. Computer-Aided Geometric Design, 111, 102318. https://doi.org/10.1016/j.cagd.2024.102318

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of Industry 4.0: A review. Engineering, 3(5), 616–630. https://doi.org/10.1016/j.eng.2017.05.015