Ministry of Higher Education and Scientific Research
Hassiba Benbouali University of Chlef
Faculty of Exact Sciences and Informatics
Department of Computer Science

THESIS

Presented for the graduation of

DOCTORATE

Field: Computer science

Specialty: Computer Systems

By

ALI CHENAOUI

Title:

Graph Parameters for Social Network Analysis

Defended on 15/10/2025, in front of the jury composed of:

Tahar Abbes Mounir	Professor	H. B. U. of Chlef	President
Tahraoui Mohammed Amin	MCA	H. B. U. of Chlef	Thesis Director
Kheddouci Hamamache	Professor	University of Lyon 1	Co-Director
Harbouche Ahmed	MCA	H. B. U. of Chlef	Examiner
Bouatouche Mourad	MCA	A. Z. U. of Relizane	Examiner
Bouadjemi Abdelkrim	MCA	A. Z. U. of Relizane	Examiner

Abstract

The exponential growth of complex networks has transformed social network analysis, creating unprecedented challenges for community detection. As networks scale to billions of nodes and connections, traditional approaches struggle to capture the true nature of community structures. This thesis addresses three fundamental limitations in current methodologies: inadequate integration of node importance, limited similarity measures, and inflexible optimisation approaches. Through theoretical development and experimental validation, this research presents three innovative contributions that significantly advance community detection capabilities.

The first contribution is the Neighbourhood overlap and Density (*NoD*) similarity measure, which combines information about shared connections with local structural density. This novel measure overcomes limitations in traditional approaches by considering both how many neighbours two nodes share and how densely those shared neighbours are connected. Experimental validation on real-world benchmark networks—including Zachary's Karate Club, Dolphins, Football, and Polbooks—shows that *NoD* increases modularity by up to 0.08 (from 0.38 to 0.46 on Dolphins) while maintaining competitive accuracy with Normalised Mutual Information (NMI) values ranging from 0.71 to 0.90 across the evaluated datasets.

Building upon *NoD*, two complementary algorithmic contributions address distinct community detection scenarios. The Heuristic Community detection algorithm based on Centrality and Similarity measures (*HCCS*) introduces a deterministic approach that recognises the varying importance of nodes in community formation. By systematically integrating centrality-based leader selection with similarity-driven community formation, *HCCS* overcomes the limitations of traditional approaches that treat these aspects separately, resulting in more accurate and robust community detection. Experimental evaluation across nine real networks—from small social graphs to large infrastructure and communication networks—using modularity and Normalised Mutual Information confirms its effectiveness, delivering the highest modularity on Karate (0.42), Football (0.60), and Uni_email (0.55) and competitive NMI scores such as 0.89 on Dolphins and 0.90 on Football while preserving reproducible results.

The Ant Colony Optimization Based on Centrality and NoD Similarity (*ACO-CNoD*) algorithm extends detection capabilities to overlapping community structures, where nodes can belong to multiple communities simultaneously—a common characteristic in real-world social networks. *ACO-CNoD* incorporates adaptive mechanisms that automatically adjust to different network characteristics without requiring manual parameter tuning. Comprehensive evaluation on the same benchmark suite, complemented by the overlapping-heavy Pretty Good Privacy (PGP) network, shows that *ACO-CNoD* achieves the top overlapping modularity (Q_{ov}) on five of seven datasets (e.g., 0.72 on Karate and Dolphins, 0.709 on Jazz_collab), trailing only on PGP where COPRA reaches 0.783 compared with 0.68 for *ACO-CNoD*.

Together, these contributions establish a new methodological foundation for community detection, offering complementary approaches for different network types and application requirements. The research balances theoretical advancement with practical applicability, bridging fundamental graph theory with real-world applications in social media analysis, biological networks, organisational studies, and infrastructure optimisation. The integrated framework significantly enhances our ability to extract meaningful community structures from complex networks, with implications for diverse domains including recommendation systems, influence maximisation, and network resilience analysis.

Keywords: Social Networks, Graph Parameters, Centrality, Similarity Measures, Community Detection, Overlapping Communities, Ant Colony Optimisation.

Résumé

La croissance exponentielle des réseaux complexes a transformé l'analyse des réseaux sociaux, créant des défis sans précédent pour la détection de communautés. Alors que les réseaux s'étendent à des milliards de nœuds et de connexions, les approches traditionnelles peinent à capturer la véritable nature des structures communautaires. Cette thèse aborde trois limitations fondamentales des méthodologies actuelles : l'intégration inadéquate de l'importance des nœuds, les mesures de similarité limitées, et les approches d'optimisation inflexibles. À travers un développement théorique et une validation expérimentale, cette recherche présente trois contributions innovantes qui font progresser significativement les capacités de détection de communautés.

La première contribution est la mesure de similarité Chevauchement de Voisinage et Densité (*NoD*), qui combine les informations sur les connexions partagées avec la densité structurelle locale. Cette mesure novatrice surmonte les limitations des approches traditionnelles en considérant à la fois le nombre de voisins que deux nœuds partagent et la densité de connexion de ces voisins partagés. La validation expérimentale menée sur des réseaux réels emblématiques – Zachary, Dolphins, Football et Polbooks – montre que *NoD* accroît la modularité jusqu'à +0,08 (de 0,38 à 0,46 sur Dolphins) tout en maintenant une précision compétitive avec des valeurs d'information mutuelle normalisée (NMI) comprises entre 0,71 et 0,90 selon les jeux de données.

S'appuyant sur *NoD*, deux contributions algorithmiques complémentaires répondent à des scénarios distincts de détection de communautés. L'algorithme Heuristique de détection de communautés basé sur la Centralité et les mesures de Similarité (*HCCS*) introduit une approche déterministe qui reconnaît l'importance variable des nœuds dans la formation des communautés. En intégrant systématiquement la sélection de leaders basée sur la centralité avec la formation de communautés guidée par la similarité, *HCCS* surmonte les limitations des approches traditionnelles qui traitent ces aspects séparément, aboutissant à une détection de communautés plus précise et robuste. Cet algorithme fournit des résultats cohérents et reproductibles – un avantage crucial pour les applications scientifiques. L'évaluation expérimentale conduite sur neuf réseaux réels, allant de petits graphes sociaux à de vastes réseaux

d'infrastructure et de communication, à l'aide de la modularité et du NMI, confirme son efficacité en atteignant les meilleures modularités sur Karate (0,42), Football (0,60) et Uni_email (0,55) ainsi que des NMI compétitifs tels que 0,89 sur Dolphins et 0,90 sur Football.

L'algorithme d'Optimisation par Colonie de Fourmis basé sur la Centralité et la similarité NoD (*ACO-CNoD*) étend les capacités de détection aux structures communautaires chevauchantes, où les nœuds peuvent appartenir à plusieurs communautés simultanément – une caractéristique commune dans les réseaux sociaux réels. *ACO-CNoD* intègre des mécanismes adaptatifs qui s'ajustent automatiquement aux différentes caractéristiques des réseaux sans nécessiter de réglage manuel des paramètres. Une évaluation complète sur le même ensemble de réseaux, complétée par le réseau Pretty Good Privacy (PGP) à forte superposition, montre qu'*ACO-CNoD* obtient la meilleure modularité chevauchante (Q_{ov}) sur cinq jeux de données sur sept (par exemple 0,72 sur Karate et Dolphins, 0,709 sur Jazz_collab), ne cédant du terrain que sur PGP où COPRA atteint 0,783 contre 0,68 pour *ACO-CNoD*.

Ensemble, ces contributions établissent une nouvelle base méthodologique pour la détection de communautés, offrant des approches complémentaires pour différents types de réseaux et exigences d'application. La recherche équilibre l'avancement théorique avec l'applicabilité pratique, reliant la théorie fondamentale des graphes aux applications réelles dans l'analyse des médias sociaux, les réseaux biologiques, les études organisationnelles et l'optimisation des infrastructures. Le cadre intégré améliore significativement notre capacité à extraire des structures communautaires significatives des réseaux complexes, avec des implications pour divers domaines, notamment les systèmes de recommandation, la maximisation de l'influence et l'analyse de la résilience des réseaux.

Mots-clés : Réseaux Sociaux, Paramètres des Graphes, Centralité, Mesures de Similarité, Détection de Communautés, Communautés Chevauchantes, Optimisation par Colonie de Fourmis.

ملخص

معلمات الرسم البياني لتحليل الشبكات الاجتماعية

أدَّى النموُّ المتسارعُ للشبكات المعقَّدة إلى إعادة تشكيل تحليل الشبكات الاجتماعية، مما خلق تحديات غير مسبوقة أمام اكتشاف المجتمعات. ومع امتداد الشبكات إلى مليارات العقد والوصلات، تعجز الأساليب التقليدية عن التقاط الطبيعة الحقيقية للهياكل المجتمعية. تتناول هذه الأطروحة ثلاثة قيود جوهرية في المنهجيات الحالية: التكامل غير الكافي لأهمية العقد، وضآلة مقاييس التشابه، وجمود مقاربات التحسين. ومن خلال تطوير نظري وتحقيق تجريبي، يقدم هذا البحث ثلاث مساهمات مبتكرة تعزز قدرات اكتشاف المجتمعات بصورة ملموسة.

تتمثل المساهمة الأولى في مقياس تشابه تداخل الجوار والكثافة (NoD)، الذي يجمع بين المعلومات المتعلقة بالروابط المشتركة والكثافة البنيوية المحلية. يتجاوز هذا المقياس القيود التقليدية عبر موازنة كل من عدد الجيران الذين يتشاركهما عقدتان ومدى ترابط هؤلاء الجيران فيما بينهم. تثبت التقييمات التجريبية على شبكات معيارية واقعية مثل نادي الكاراتيه Zakary، Dolphins، Football، و Polbooks أن NoD يرفع قيمة المعيار التجميعي (الموديولارية) بما يصل إلى 0.08+ (من 0.38 إلى 0.46 على شبكة Dolphins) مع الحفاظ على دقة تنافسية تتراوح فيها قيم مؤشر NMI بين 0.71 و 0.90عبر مجموعات البيانات المقيمة.

بالاستناد إلى NoD ، تقدم هذه الأطروحة مساهمتين خوارزميتين مكملتين لسيناريوهات مختلفة من اكتشاف المجتمعات. خوارزمية الاكتشاف الاستدلالي المعتمدة على المركزية ومقاييس التشابه (HCCS) تطرح مقاربة حتمية تعترف بتفاوت أهمية العقد في تشكيل المجتمعات. ومن خلال دمج اختيار القادة اعتماداً على المركزية مع تكوين المجتمعات المدفوع بالتشابه، تتخطى HCCS حدود المقاربات التي تفصل بين هذين البعدين. تظهر التقييمات التي تغطي تسعة شبكات واقعية تمتد من الرسوم الاجتماعية الصغيرة إلى شبكات البنية التحتية والاتصالات أن HCCS تحقق أعلى قيم للموديولارية على Karate (0.42)، Football (0.60)، وUni_email (0.55) ، إضافة إلى قيم NMI تنافسية مثل 0.89 على Dolphins و 0.90 على Football، مع نتائج ثابتة وقابلة للتكرار.

أما خوارزمية تحسين مستعمرة النمل والتشابه (ACO-CNoD) فتمدد الإمكانات إلى الهياكل المجتمعية المتداخلة، حيث يمكن للعقد الانتماء إلى عدة مجتمعات في آن واحد؛ وهي سمة شائعة في الشبكات الاجتماعية الواقعية. تدمج ACO-CNoD آليات تكيفية تضبط نفسها تلقائياً وفق خصائص الشبكة دون الحاجة إلى معايرة يدوية للمعلمات. ويبرهن التقييم الشامل على نفس مجموعة المعايير، مع إضافة شبكة PGP ذات التداخل العالي، أن ACO-CNoD تحقق أفضل موديولارية متداخلة (Q_{ov}) في خمسة من أصل سبعة مجموعات بيانات (مثل 0.72علىKarate وDolphins و0.709 علىJazz_collab)، فيما تتراجع فقط على PGP حيث تصل COPRAإلى 0.783 مقابل 0.68 لـACO-CNoD.

معاً، تؤسس هذه المساهمات أساساً منهجياً جديداً لاكتشاف المجتمعات، وتقدم مقاربات متكاملة لأنواع الشبكات المختلفة ومتطلبات التطبيقات. يوازن البحث بين التقدم النظري والقابلية التطبيقية، رابطاً بين نظرية الرسوم البيانية الأساسية والتطبيقات الواقعية في تحليل وسائط التواصل الاجتماعي، والشبكات البيولوجية، والدراسات التنظيمية، وتحسين البنى التحتية. يرفع الإطار المتكامل قدرتنا على استخلاص هياكل مجتمعية ذات معنى من الشبكات المعقدة، مع تأثيرات تمتد إلى مجالات مثل أنظمة التوصية وتعظيم التأثير وتحليل صمود الشبكات.

الكلمات المفتاحية :الشبكات الاجتماعية، معلمات الرسوم البيانية، المركزية، مقاييس التشابه، اكتشاف المجتمعات، المجتمعات المتداخلة، التحسين بمستعمرات النمل.

Acknowledgements

All praise is due to Allah, whose boundless mercy, guidance, and support have illuminated every step of this journey and granted me the strength to see it through.

I would like to express my deepest gratitude to the members of my thesis jury: Dr. Tahar Abbes Mounir (President), Dr. Harbouche Ahmed, Dr. Bouatouche Mourad, and Dr. Bouadjemi Abdelkrim, for their valuable insights, constructive feedback, and the time they dedicated to evaluating this work. Their expertise significantly strengthened the theoretical foundations and experimental validation of this research.

I extend my sincere gratitude to my thesis director, Dr. Tahraoui Mohammed Amin, for his exceptional guidance and expertise in graph theory and social network analysis that shaped this work. His feedback and encouragement were instrumental in my development as a researcher.

I am deeply grateful to my co-director, Dr. Kheddouci Hamamache from the University of Lyon 1, for his support throughout this research journey.

I am grateful to the Department of Computer Science at Hassiba Benbouali University of Chlef for providing the necessary research environment and computational resources that made this work possible.

My appreciation goes to my fellow doctoral students and colleagues in the Computer Science Department, whose intellectual discussions and collaborative spirit enriched my understanding of the field. I extend special thanks to my dear friend Dr. Maabed Mohamed for his exceptional support and invaluable assistance in the final stages of this work.

I am profoundly grateful to my family for their unwavering support, patience, and encouragement throughout this challenging yet rewarding journey. Their belief in my abilities provided the motivation to persevere through the most difficult moments of this research.

I hope this work contributes meaningfully to the advancement of social network analysis and inspires future research in graph-based community detection algorithms.

Dedication

To my beloved wife, Dr. Bouzada Nassira,

whose unwavering support, endless patience,
and boundless love have made this journey possible.

To my precious daughters Besma, Ines, and Lina, and my son Djawed,

whose joy and curiosity give purpose to every effort.

To my parents and my extended family,

who instilled in me the value of education
and the courage to pursue my dreams.

To my teachers and mentors,

who shaped my intellectual curiosity
and guided me toward academic excellence.

To the research community,

whose dedication to knowledge and discovery
continues to inspire and motivate.

And to the future generation of researchers

who will build upon this work
and push the boundaries of science even further.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Definition
ACO	Ant Colony Optimisation
ACO-CNoD	Ant Colony Optimization Based on Centrality and NoD Similarity
AI	Artificial Intelligence
ARI	Adjusted Rand Index
BC	Betweenness Centrality
CC	Closeness Centrality
CN	Common Neighbours
DC	Degree Centrality
EC	Eigenvector Centrality
GA	Genetic Algorithm
HCCS	Heuristic Community detection algorithm based on Centrality and Similarity measures
ML	Machine Learning
NMI	Normalised Mutual Information
NoD	Neighbourhood overlap and Density
PageRank	PageRank Centrality
PC	Pearson Correlation
PSO	Particle Swarm Optimisation
SNA	Social Network Analysis

Mathematical Notation	Definition
$G = (V, E)$	Graph with vertex set V and edge set E
$\|V\|$	Number of vertices in graph G

Abbreviation	**Definition**
$\|E\|$	Number of edges in graph G
$N(v)$	Neighbourhood of vertex v
$d(v)$	Degree of vertex v
$d(u, v)$	Distance between vertices u and v
A	Adjacency matrix
A_{ij}	Entry in adjacency matrix at position (i, j)
λ	Eigenvalue
C_i	Community i
Q	Modularity score
τ	Pheromone level
η	Heuristic information
α	Pheromone importance parameter
β	Heuristic importance parameter
ρ	Pheromone evaporation rate
$\Delta\tau$	Pheromone update
P_{ij}	Transition probability from i to j
σ	Standard deviation
μ	Mean value
θ	Threshold parameter
γ	Scaling parameter
ϵ	Small positive constant
δ	Delta function
Ω	Solution space
$\mathcal{O}$	Big O notation (computational complexity)
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\mathbb{Z}$	Set of integers
$\in$	Element of
$\subset$	Subset of
$\cup$	Union
$\cap$	Intersection
$\emptyset$	Empty set
$\forall$	For all

Abbreviation	**Definition**
$\exists$	There exists
$\sum$	Summation
$\prod$	Product
max	Maximum
min	Minimum
arg max	Argument of maximum
arg min	Argument of minimum

Technical Terms	**Definition**
Centrality	Measure of node importance in a network
Community	Densely connected group of nodes
Modularity	Quality measure for community detection
Similarity	Measure of relationship strength between nodes
Heuristic	Problem-solving approach using practical methods
Metaheuristic	High-level problem-solving strategy
Optimisation	Process of finding the best solution
Convergence	Approach to a stable solution
Scalability	Ability to handle increasing problem sizes
Robustness	Stability under varying conditions
Complexity	Computational resource requirements
Algorithm	Step-by-step problem-solving procedure
Graph	Mathematical structure of nodes and edges
Network	System of interconnected elements
Node	Individual element in a network
Edge	Connection between two nodes
Vertex	Alternative term for node
Adjacency	Direct connection between nodes
Path	Sequence of connected nodes
Cycle	Closed path in a graph
Component	Connected subgraph
Clustering	Grouping of similar elements
Classification	Assignment to predefined categories
Validation	Verification of results accuracy

Abbreviation	**Definition**
Evaluation	Assessment of performance
Benchmark	Standard for comparison
Dataset	Collection of data for analysis
Experiment	Systematic investigation
Analysis	Detailed examination
Methodology	Systematic approach
Framework	Structural foundation
Implementation	Practical realisation
Performance	Measure of effectiveness
Efficiency	Measure of resource utilisation

Glossary

Technical Terms and Acronyms

ACO-CNoD Ant Colony Optimization Based on Centrality and NoD Similarity algorithm - the metaheuristic for overlapping community detection developed in this thesis.

ACO Ant Colony Optimisation - A metaheuristic algorithm inspired by the foraging behaviour of ants.

ARI Adjusted Rand Index - A measure for comparing the similarity between two data clusterings, adjusted for chance.

Betweenness Centrality
A centrality measure that quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.

Centrality Measures
Metrics that identify the most important vertices within a graph based on different criteria such as connectivity, position, or influence.

Closeness Centrality
A centrality measure that indicates how close a node is to all other nodes in the network.

Community Detection
The process of identifying groups of nodes in a network that are more densely connected internally than to the rest of the network.

Degree Centrality
A centrality measure based on the number of direct connections a node has.

Eigenvector Centrality
A centrality measure that considers not just the number of connections but also the importance of those connections.

Graph A mathematical structure consisting of vertices (nodes) connected by edges (links).

HCCS Heuristic Community detection algorithm based on Centrality and Similarity measures - A deterministic community detection algorithm developed in this thesis.

Modularity A quality measure for community structure that compares the density of connections within communities to the expected density in a random network.

Network A collection of interconnected elements, represented mathematically as a graph.

NMI Normalised Mutual Information - A measure for comparing detected communities with ground truth communities.

NoD Neighbourhood overlap and Density similarity measure - A novel similarity measure developed in this thesis that integrates neighbourhood overlap and local density.

Overlapping Communities Communities where nodes can belong to multiple communities simultaneously.

Q Modularity score - A numerical value representing the quality of a community partition.

Q_ov Overlapping modularity score - An extension of modularity for overlapping community structures.

SNA Social Network Analysis - The process of investigating social structures through the use of networks and graph theory.

Social Network A network representing social relationships between individuals, groups, or organisations.

Similarity Measure A function that quantifies the similarity or relatedness between two nodes in a network.

Mathematical Notation

$G = (V, E)$ A graph G with vertex set V and edge set E.

$|V|$ The number of vertices in the graph (also denoted as n).

$|E|$ The number of edges in the graph (also denoted as m).

$N(v)$ The neighbourhood of vertex v - the set of vertices adjacent to v.

$deg(v)$ The degree of vertex v - the number of edges incident to v.

$d(u, v)$	The shortest path distance between vertices u and v.
$CN(u, v)$	Common neighbours of vertices u and v - the intersection $N(u) \cap N(v)$.
$E_{CN}(u, v)$	The number of edges within the subgraph induced by the common neighbours of u and v.
θ	Weighting parameter in the NoD similarity measure, balancing neighbourhood overlap and density components.
$C = \{C_1, C_2, \ldots, C_k\}$	A community partition where C_i represents the i-th community.
Q	Modularity value of a community partition.
α, β, ρ	Parameters in ant colony optimisation: pheromone influence, heuristic influence, and evaporation rate respectively.

List of Publications

This thesis is based on the following publications and submitted manuscripts:

Published Articles

Journal Articles:

1. **Chenaoui, A.** and Tahraoui, M.A. (2025). An Iterated Heuristic Community Detection Algorithm for Social Networks Based on Centrality and Similarity Measures. *Engineering, Technology & Applied Science Research*, 15(4), 24211–24219. DOI: https://doi.org/10.48084/etasr.10791

Conference Presentations

1. **Chenaoui, A.** and Tahraoui, M.A. (2025). A New Community Detection Algorithm for Social Networks Using Centrality and Similarity Measures. Presented at the First International Conference on Artificial Intelligence, Smart Technologies and Communications (AISTC'2025), Hassiba Benbouali University of Chlef, Algeria, April 14-15, 2025.
2. **Chenaoui, A.** and Tahraoui, M.A. (2023). Local Community Detection Algorithm in Social Networks Using Similarity Measures. Presented at the First National Conference on Artificial Intelligence, Smart Technologies and Communications, Hassiba Ben Bouali University of Chlef, Algeria, November 8-9, 2023.
3. **Chenaoui, A.** and Tahraoui, M.A. (2023). A Community Detection Algorithm Using Centrality Measures in Social Networks. Presented at the First National Conference in Computer Science Research and its Applications (RIA'23), Relizane University, Algeria, May 10, 2023.

Manuscripts in Preparation

1. **Chenaoui, A.**, Tahraoui, M.A., and Kheddouci, H. ACO-CNoD: Ant Colony Optimization Based on Centrality and NoD Similarity for Overlapping Community Detection in Social Networks. *Target journal: Social Networks* (in preparation).
2. **Chenaoui, A.** and Tahraoui, M.A., Neighbourhood overlap and Density (NoD): A Novel Similarity Measure for Social Network Analysis. *Target journal: Social Networks* (in preparation).

Chapter 1

General Introduction

Research Context and Significance

Social network analysis has become a foundational area of computational science. It supports inquiries that range from human behaviour on online platforms to biological and infrastructure systems. The proliferation of large-scale digital platforms, such as Facebook and X, produces relational data involving billions of interactions, thereby creating networks whose complexity challenges conventional analytical pipelines.

Community detection is a fundamental component of this research area. It seeks densely connected groups whose internal cohesion exceeds their external connectivity, enabling researchers to interpret organisational structures, information diffusion pathways, and functional relationships across domains that include social media, biology, infrastructure, and information systems. When accurately resolved, communities provide decision-makers with interpretable abstractions that support tasks from targeted recommendation to resilience assessment.

The growing scale and heterogeneity of contemporary networks expose the limits of established algorithms. Modern graphs exhibit pronounced differences in node importance, allow for overlapping memberships, and display regions with starkly contrasting densities. These characteristics motivate the search for approaches that remain interpretable while accommodating structural diversity and operating at scale. The remainder of this chapter articulates the specific shortcomings that motivate the thesis and outlines the contributions devised to address them.

Problem Statement

Despite significant advances in community detection research, current algorithms face three fundamental limitations that impact their effectiveness when applied to modern,

large-scale networks. These limitations, identified through comprehensive analysis of the literature and empirical evaluation, form the core research gaps addressed in this thesis:

1. **Inadequate Integration of Node Centrality Measures:** Traditional algorithms treat all nodes uniformly [1, 2], failing to recognise that different nodes play fundamentally different roles in network structure and community formation. For instance, nodes with high degree centrality (numerous direct connections) drive local cohesion, whereas those with high betweenness centrality broker information across communities [3]. In the absence of a principled framework that combines these complementary views, boundary detection and hierarchy inference remain inconsistent.
2. **Density-Blind Similarity Measures:** Existing similarity measures (common neighbours [4], Jaccard [5], cosine [6]) focus primarily on neighbourhood overlap while neglecting the density of the subgraphs that surround candidate pairs. Consequently, an edge embedded in a sparse periphery can receive the same score as one in a tightly knit core, a phenomenon we term *density blindness*. Empirical observations confirm that this oversight severely degrades discriminative power in heterogeneous networks characterised by both sparse and dense regions.
3. **Parameter-Sensitive Metaheuristic Approaches:** Current metaheuristic approaches suffer from acute parameter sensitivity, slow convergence, and limited adaptability across different network types [7, 8]. Existing ant colony optimisation algorithms rely on simplistic pheromone models and static parameter schedules that must be tuned manually for each dataset. Without mechanisms for adaptive control, these algorithms frequently converge prematurely to suboptimal solutions, particularly when navigating overlapping community structures with competing objectives.

Collectively, these limitations reveal a pronounced research gap: community detection methods must (1) integrate complementary centrality indicators within a coherent decision framework, (2) quantify similarity in a manner that remains sensitive to local density, and (3) deploy adaptive metaheuristics capable of maintaining performance across network regimes. Addressing this gap necessitates algorithmic innovations that blend theoretical guarantees with practical engineering, thereby expanding both the interpretability and the scalability of community detection pipelines. The next section introduces the thesis contributions designed to close each of these gaps in turn.

Research Contributions Overview

Together, these gaps motivate a tripartite contribution strategy that mirrors the structure of the problem statement. Each contribution forms a targeted response to one of the limitations, while collectively establishing a coherent framework for community detection in heterogeneous networks.

- **Contribution 1: NoD (Neighbourhood-overlap-Density) Similarity Measure**

  Objective: Develop a density-aware similarity measure that integrates neighbourhood overlap with local subgraph density, thereby mitigating *density blindness*.

  Key features: Dual-component formulation that adapts to heterogeneous structures, formally proven mathematical properties, demonstrably higher discriminative power in mixed-density graphs, and computational efficiency compatible with large networks.

- **Contribution 2: HCCS Algorithm**

  Objective: Integrate complementary centrality indicators with the NoD similarity measure to guide deterministic community construction.

  Key features: Leader selection informed by multi-perspective centrality and bounded neighbourhood radius (spatial constraint), NoD-driven aggregation of followers, deterministic processing for reproducibility, and a unified treatment of local cohesion and global structure tailored to non-overlapping scenarios.

- **Contribution 3: *ACO-CNoD* Algorithm**

  Objective: Design an adaptive ant colony optimisation procedure that leverages NoD-guided heuristics for overlapping community detection.

  Key features: Density-aware heuristic information, adaptive parameter scheduling that responds to network feedback, hybrid pheromone updates that couple local exploration with global reinforcement, and explicit support for overlapping membership structures.

Thesis Organisation

This thesis is organised into seven chapters that systematically present the theoretical foundations, methodological developments, experimental validation, and conclusions of this research:

Chapter 1: General Introduction presents the research context and problem statement, identifying three fundamental limitations in existing community detection

approaches. It then outlines the objectives and the corresponding contributions of this thesis designed to address these gaps.

Chapter 2: Background sets out the theoretical concepts required to contextualise the contributions. It introduces core notions from graph theory and social network analysis, revisits canonical centrality and similarity measures, and summarises foundational work on community detection.

Chapter 3: Social Network Analysis and Community Detection Problem develops a comprehensive literature review that captures the evolution of community detection. It analyses centrality-driven techniques, similarity-based strategies, and metaheuristic approaches, thereby pinpointing the precise research gaps that the thesis addresses.

Chapter 4: Centrality and Similarity-based Community Detection introduces the first two major contributions: the NoD similarity measure and the HCCS algorithm. This chapter explains how these innovations address the limitations of traditional approaches by integrating density awareness into similarity calculations and effectively utilising node centrality in community formation. The mathematical foundations, algorithmic design, and implementation considerations are presented in a structured manner.

Chapter 5: Community Detection Using Ant Colony Optimisation Based on Centrality and NoD Similarity presents the *ACO-CNoD* algorithm, the third major contribution. This chapter introduces an innovative approach that combines ant colony optimisation techniques with the NoD similarity measure to detect overlapping communities. It explains how *ACO-CNoD* overcomes the parameter sensitivity issues of traditional metaheuristic methods through adaptive mechanisms that automatically adjust to different network characteristics.

Chapter 6: Experimental Results and Discussion provides comprehensive empirical validation of all three contributions. It includes performance comparisons with state-of-the-art methods, scalability studies, and parameter sensitivity analyses. The chapter systematically addresses four key research questions related to the performance and characteristics of the proposed approaches.

Chapter 7: Conclusion summarises the contributions, discusses research impact and significance, identifies limitations, and outlines future research directions. It provides a holistic view of the thesis achievements and their implications for the field of social network analysis and community detection.

In summary, this introductory chapter positions the thesis within the broader context of both scientific theory and practical application. By clarifying the research gaps, presenting targeted contributions, and mapping the document structure, it underscores the potential for advancing community detection theory while delivering

techniques capable of supporting real-world decision-making in complex networked systems.

Chapter 2

Background

Contents of this chapter

2.1 Introduction

We assume that the reader has a basic knowledge of graph theory and theoretical computer science. Therefore, we provide in this chapter only a brief overview of the terminology used in this thesis. For more detailed background information, the reader may refer to [9], and [10].

2.2 Basic Definitions

Graph: a graph $G = (V,E)$ consists of two sets: a finite set V of elements called *vertices* (or *nodes*) and a finite set E of elements called *edges*. Each edge is identified with a pair of vertices. If the edges of a graph G are identified with ordered pairs of vertices, then G is called a *directed* or an *oriented* graph. Otherwise G is called an *undirected* or a *nonoriented* graph. Two vertices u and v in the graph G are called *adjacent* (or *neighbours*) in G if u and v are endpoints of an edge e of G. Such an edge e is called *incident* with the vertices u, v, and e is said to connect u and v. A *self-loop* or *loop* is an edge between a vertex and itself, and an undirected graph without loops or multiple edges is known as a *simple* graph. In this thesis we will assume graphs to be simple unless otherwise stated. For example, Figure 2.1 represents the simple graph G whose vertex set $V(G)$ is $\{ v_1, v_2, v_3, v_4\}$, and whose edge set $E(G)$ consists of the edges v_1v_2, v_1v_3, v_2v_3, and v_2v_4. The adjacent vertices in this example are $\{v_1, v_2\}$, $\{v_1, v_3\}$, $\{v_2, v_3\}$, and $\{v_2, v_4\}$.

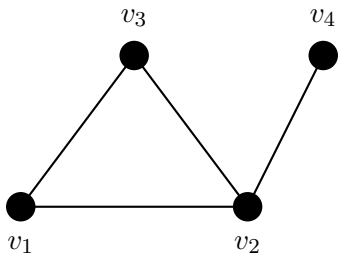


Figure 2.1: Example of a simple graph.

Neighborhood: the set of all neighbours of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the *neighbourhood* of v. For the simple graph G in Figure 2.1, the neighbourhood of the vertex v_1 is $\{v_2, v_3\}$.

Degree of a Vertex: the degree of a vertex v in a graph G is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by $deg(v)$ or $d(v)$. A vertex of degree zero is called *isolated* and a vertex is a *pendant* if and only if it has degree one. In the graph G of Figure 2.1, we have:

$$d(v_1) = 2,$$
$$d(v_2) = 3,$$
$$d(v_3) = 2,$$
$$d(v_4) = 1.$$

Independent Sets and Cliques: a set of vertices is called *independent* if no two vertices in the set are adjacent, whereas a set of vertices is called a *clique* if every two vertices in the set are adjacent. An independent set (respectively, a clique) is *maximal* if it is not contained in any larger independent set (respectively, clique). It is *maximum* if its cardinality is the largest among all independent sets (respectively, cliques) in the graph. For example, in Figure 2.2, the set $S_1 = \{ v_2, v_4 \}$ is an *independent* set and is *maximal* since no other vertex can be added to S_1 while maintaining the independence property. The set $S_2 = \{ v_1, v_3, v_5 \}$ is also independent, and it is *maximum* because there is no larger independent set in this graph.

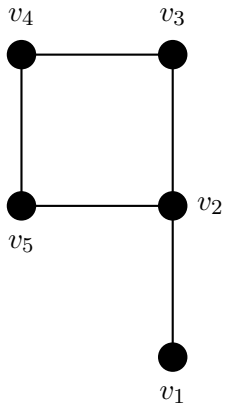


Figure 2.2: Example of independent sets and cliques.

Distance and Connectivity: a graph is *connected* if every pair of vertices can be joined by a path. Otherwise, the graph G is *disconnected*. Each *maximal* connected piece of a graph is called a *connected component*.

Distance in graphs is defined in a natural way: in a connected graph G, the distance from vertex u to vertex v is the length (number of edges) of the shortest path. We denote this distance by *d(u, v)*, and in situations where clarity of context is important, we may write $d_G(u, v)$. In the graph of the figure above, we obtain the following distances:

$$d(v_1, v_3) = 2,$$
$$d(v_1, v_5) = 2,$$
$$d(v_1, v_4) = 3.$$

Isomorphism of graphs: We say that two graphs G and H are *isomorphic*, written $G \cong H$, there is a one-one correspondence between the vertices of G and those of H such that the number of edges joining any two vertices of G is equal to the number

of edges joining the corresponding vertices of H. We provide an example in Figure 2.3.

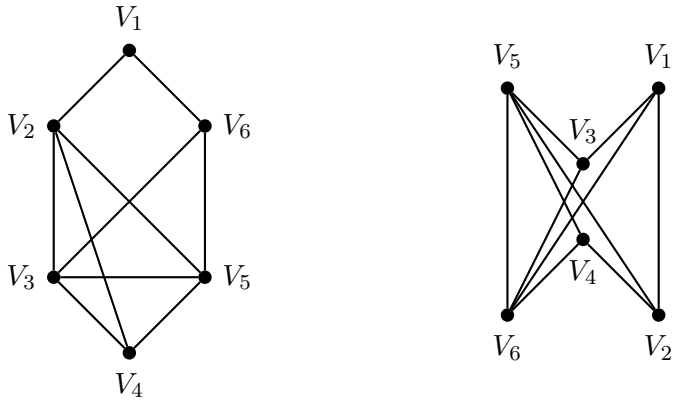


Figure 2.3: Example of isomorphic graphs.

Subgraphs. A *subgraph* of a graph G is a graph H whose vertices and edges are all contained in G. In other words, H is a subgraph of G if $V_H \subseteq V_G$ and $E_H \subseteq E_G$, as illustrated in Figure 2.4. When H is a subgraph of G, we may also say that G is a *supergraph* of H.

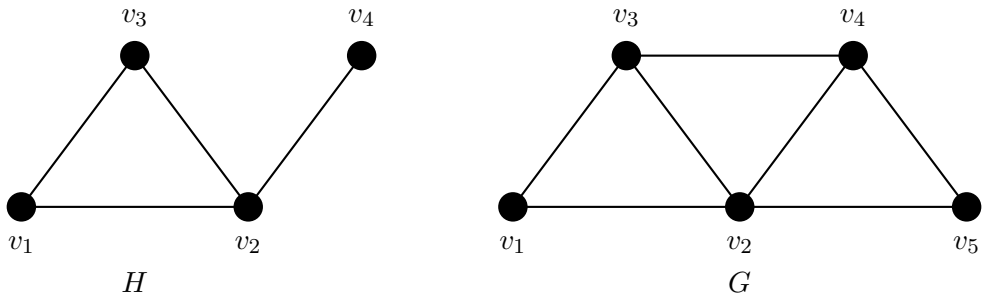


Figure 2.4: Example of subgraphs.

2.3 Graph Operations

Complement graph: the *complement* of a simple graph G is a graph H with $V_H = V_G$ and $xy \in E_H$ if and only if $xy \notin E_G$. So basically, we invert the role of adjacency and non-adjacency. We write $\bar{G}$ for the complement of G. Figure 2.5 shows an example of complement graphs.

Power of graph: given a connected graph G and a positive integer k, the *k -th* power of G, denoted G^k, is the graph with $V(G^k) = V(G)$ and where vertices u and v are adjacent in $V(G^k)$ if and only if $d_G(u, v) \leq k$. Meaning that the *kth* power G^k

(a) (G) **(b)** $(\bar{G}.)$

Figure 2.5: (a) G, (b) $\bar{G}$.

of an undirected graph G is another graph that has the same set of vertices, but in which two vertices are adjacent when their distance in G is at most k. If we look at Figure 2.6 below, we see that the graph to the left is the second power of G, G^2, where the distance between every two vertices is at most 2.

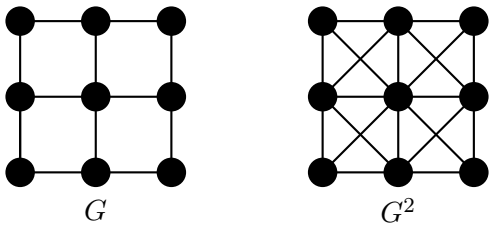


Figure 2.6: Power of a graph.

Vertex and edge deletion: a *subgraph* of G is a graph obtained by the removal of a (non-zero) number of edges and/or vertices of G. The removal of a vertex necessarily implies the removal of every edge incident with it, whereas the removal of an edge does not necessarily implies the removal of a vertex although it may result in one (or even two) isolated vertices. If we remove an edge e or a vertex v from G, then the resulting graphs are respectively denoted by *(G - e) and (G - v)*. In the figure of the previous example, by applying some edges deletion on the graph G^2 we get as a result the graph G.

Graph disjoint union: $G + H$ is the union of two disjoint graphs G and H. if $V(G) \cap V(H) = \emptyset$, then $V(G + H) = V(G) \cup V(H)$ and $E(G + H) = E(G) \cup E(H)$. An example is provided in Figure 2.7.

Cartesian product of graphs: the *Cartesian Product* of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, denoted by $G = G_1 \times G_2$ is the graph whose vertex set is $V = V_1 \times V_2$ and for any two vertices $w_1 = (u_1, v_1)$ and $w_2 = (u_2, v_2)$ in V, $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, there is an edge $w_1, w_2 \in E(G)$ if and only if either (a) $u_1 = u_2$ and $v_1, v_2 \in E_2$, or (b) $v_1 = v_2$ and $u_1, u_2 \in E_1$. We give an example in Figure 2.8.

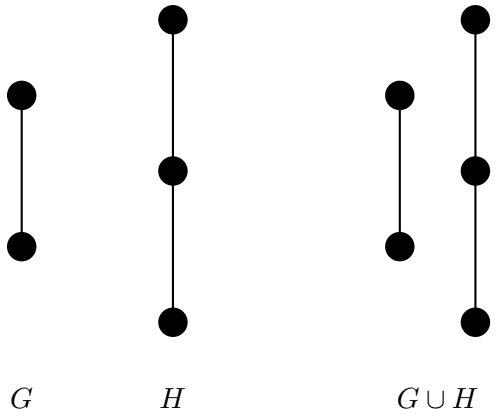


Figure 2.7: Graph disjoint union.

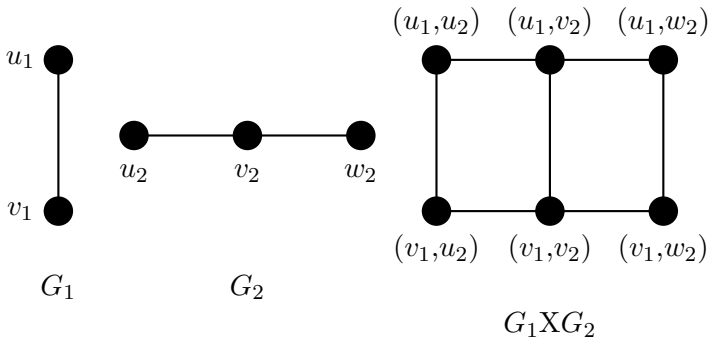


Figure 2.8: Cartesian product of graphs.

2.4 Some graph families

Bipartite Graphs: a simple graph G is called *bipartite* if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2). When this condition holds, we call the pair V_1, V_2 a bipartition of the vertex set V of G, as shown in Figure 2.9. A *complete bipartite* graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between every pair of vertices if and only if one vertex in the pair is in the first subset and the other vertex is in the second subset.

Complete graph: a complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. If we look at Figure 2.10, we see that every pair of vertices is connected by an edge.

Path graph: a *path* graph is a graph that can be drawn so that all of its vertices and edges lie on a single straight line as shown in Figure 2.11. The length of a path

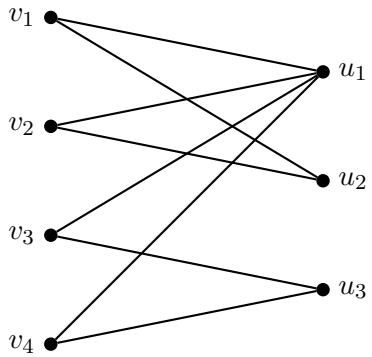


Figure 2.9: Bipartite graphs.

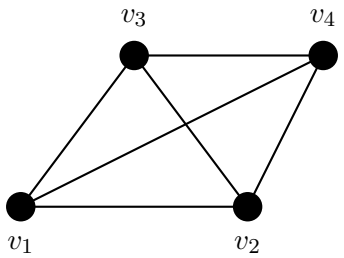


Figure 2.10: Example of a complete graph.

P is the number of edges in it.

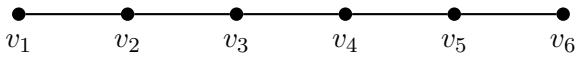


Figure 2.11: Path P_6.

Cycle graph: a *cycle* graph, or *circular* graph, is a graph that consists of a single cycle (i.e., a set of vertices connected in a closed chain), as shown in Figure 2.12.

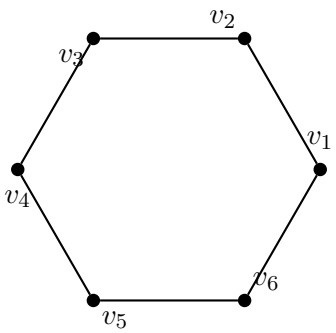


Figure 2.12: Cycle C_6

Planar graph: a graph is said to be *planar* if it can be drawn in such a way that no two edges intersect each other. Drawing a graph without edge crossing is called

embedding the graph (or planar *embedding* or planar representation.

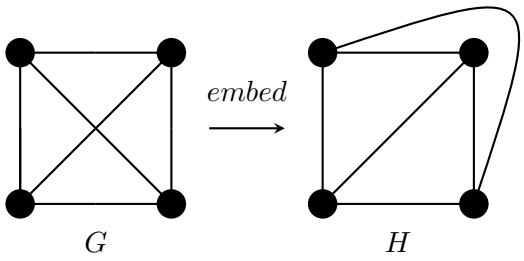


Figure 2.13: Planar graph.

Tree: a *tree* is a connected graph that has no cycles (also called an *acyclic* graph). Equivalently, it is a connected graph with $n-1$ edges (where n is the number of vertices), or an acyclic graph with $n-1$ edges. A forest is a graph with no cycles, or equivalently, a graph all of whose components are trees. A vertex of degree 1 in a tree is called a *leaf*. See an example in Figure 2.14 below.

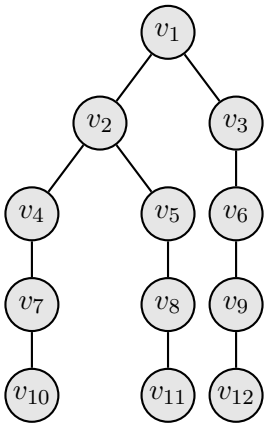


Figure 2.14: Example of a tree graph

2.5 Graph representation

There are several standard ways to represent graphs, such as adjacency matrices, adjacency lists, incidence matrices, and incidence lists, among others. Each representation offers specific advantages depending on the type of graph and the operations to be performed. For example, an adjacency matrix provides constant-time access to edge queries and is particularly suitable for dense graphs, although it requires $O(n^2)$ space for a graph with $n = |V|$ vertices, even when the graph is sparse. In contrast, an adjacency list is more space-efficient for sparse graphs and facilitates

traversal of neighbours, but edge existence queries may require linear time in the degree of a vertex. Similarly, incidence matrices and lists are useful in contexts where edge-oriented operations are predominant, such as network flows or hypergraphs.

In this thesis, we adopt the adjacency matrix as our primary representation. This choice is motivated by its simplicity, its suitability for algorithmic analysis, and its ability to facilitate direct manipulation of graph properties (e.g., degrees, connectivity, or matrix-based computations such as eigenvalue analysis). Moreover, the adjacency matrix provides a convenient framework for implementing and analyzing the algorithms developed in the subsequent chapters.

2.6 Fundamental Graph Metrics and measures

To extract meaningful insights from networks, researchers rely on a variety of metrics and measures that capture both global properties (e.g., connectivity, efficiency) and local properties (e.g., influence of individual nodes, similarity between pairs of vertices). These quantitative tools are crucial for analyzing processes such as information diffusion, epidemic spreading, robustness against failures, and the emergence of communities.

In this section, we present fundamental graph metrics and measures that will be repeatedly used throughout our study. We begin with structural graph metrics, which describe global connectivity and cohesion. We then introduce centrality measures, which assess the relative importance of nodes within a network. Next, we discuss similarity measures, with a focus on neighbourhood-based indices, which quantify the strength of relationships between pairs of vertices. Finally, we present the pivotal evaluation metrics used to assess the quality of community detection. Together, these measures form the core toolkit for modern network analysis.

2.6.1 Graph Metrics

Graphs can be characterized by a variety of structural metrics that capture their connectivity, cohesion, and overall efficiency. Such measures are central in the study of complex networks, as they allow us to quantify how information propagates, how communities emerge, and how a network resists failures or perturbations. In particular, metrics such as the clustering coefficient, the average path length, and the global efficiency are widely used in social network analysis, biology, computer networks, and many other domains. Below, we recall the formal definitions of these measures, which will be essential in the forthcoming analysis of our models.

Definition 2.1 (Clustering Coefficient)**.** The clustering coefficient of a vertex v measures the degree to which its neighbours are connected to each other. The coefficient is defined as 0 if the vertex has fewer than two neighbours. Otherwise, it is given by:

$$C(v) = \frac{2e_v}{deg(v)(deg(v) - 1)} \tag{2.1}$$

where e_v is the number of edges between the neighbours of v.

Definition 2.2 (Average Path Length)**.** The average path length of a graph is a measure of the typical separation between two vertices in the network. It is defined as the average of the shortest path distances over all pairs of distinct vertices. For a connected graph with n vertices, the average path length L is given by:

$$L = \frac{1}{n(n-1)} \sum_{u,v \in V, u \neq v} d(u, v) \tag{2.2}$$

where $d(u, v)$ is the shortest path distance between vertices u and v. A lower average path length indicates a more closely connected network, often associated with the "small-world" phenomenon.

Definition 2.3 (Graph Efficiency)**.** The global efficiency of a graph is a measure of how efficiently information can be exchanged over the network. It is defined as the average of the inverse of the shortest path distances between all pairs of distinct vertices. The efficiency E is given by:

$$E = \frac{1}{n(n-1)} \sum_{u,v \in V, u \neq v} \frac{1}{d(u, v)} \tag{2.3}$$

This measure has the advantage of being well-defined even for disconnected graphs, where the distance between vertices in different components is considered infinite, and their contribution to the sum is 0. Higher efficiency indicates a more robust and well-connected network.

These metrics are particularly important in social network analysis as they characterise the structural properties that influence information flow and community formation. Figure 2.15 illustrates a simple undirected graph that demonstrates these basic concepts.

2.6.2 Centrality Measures

Centrality measures quantify the importance or influence of vertices within a network, providing fundamental tools for identifying key actors and understanding network

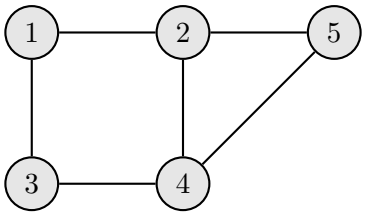


Figure 2.15: Example of a simple undirected graph with 5 vertices and 6 edges. Vertex degrees: $d(1) = 2$, $d(2) = 3$, $d(3) = 2$, $d(4) = 3$, $d(5) = 2$.

Table 2.1: Comparison of three fundamental graph metrics.

Metric	Focus	Interpretation in Networks
Clustering Coefficient	Local connectivity among neighbours	High values indicate strong community structure or tightly-knit neighbourhoods.
Average Path Length	Global shortest distances between vertices	Reflects the overall communication distance across the network. Low values indicate a "small-world" effect.
Graph Efficiency	Inverse of path lengths (including disconnected pairs)	Captures how efficiently information or resources can be exchanged across the network, even in disconnected settings.

structure [1, 2]. This section presents the mathematical definitions of key centrality measures used in network analysis.

2.6.2.1 Degree Centrality

Definition 2.4 (Degree Centrality)**.** The degree centrality of vertex v is:

$$C_D(v) = \frac{deg(v)}{n-1} \tag{2.4}$$

where $deg(v)$ is the degree of vertex v and n is the total number of vertices in the network.

2.6.2.2 Betweenness Centrality

Definition 2.5 (Betweenness Centrality)**.** The betweenness centrality of vertex v is:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.5}$$

where σ_{st} is the total number of shortest paths from vertex s to vertex t, and $\sigma_{st}(v)$ is the number of those shortest paths that pass through vertex v.

2.6.2.3 Closeness Centrality

Definition 2.6 (Closeness Centrality)**.** The closeness centrality of vertex v is:

$$C_C(v) = \frac{n-1}{\sum_{u \in V} d(v,u)} \tag{2.6}$$

where $d(v,u)$ is the shortest path distance between vertices v and u, and n is the total number of vertices.

2.6.2.4 Eigenvector Centrality

Definition 2.7 (Eigenvector Centrality)**.** The eigenvector centrality satisfies the eigenvalue equation:

$$C_E(v) = \frac{1}{\lambda} \sum_{u \in N(v)} C_E(u) \tag{2.7}$$

where $N(v)$ is the set of neighbours of vertex v, and λ is the largest eigenvalue of the adjacency matrix A.

2.6.3 Similarity Measures

Similarity measures quantify the strength of relationships between vertices in a network [6, 11]. This section presents mathematical definitions of key similarity measures.

In the analysis of complex networks, measuring the similarity between pairs of vertices is essential for tasks such as link prediction, community detection, and recommendation systems. Neighbourhood-based similarity measures exploit the local structure of the graph by comparing the sets of neighbours associated with each vertex. The underlying assumption is that two vertices are more likely to be similar, or even connected, if they share many neighbours or if their neighbours have particular structural properties. We introduce now several widely used neighbourhood-based similarity indices, ranging from simple counts of shared neighbours to more refined metrics that consider the connectivity patterns of those neighbours.

2.6.3.1 Common Neighbours

Definition 2.8 (Common Neighbours)**.** The common neighbours similarity between vertices u and v is:

$$CN(u,v) = |N(u) \cap N(v)| \tag{2.8}$$

where $N(u)$ and $N(v)$ are the neighbourhoods of vertices u and v, respectively.

2.6.3.2 Jaccard Similarity

Definition 2.9 (Jaccard Similarity)**.** The Jaccard similarity between vertices u and v is:

$$J(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{2.9}$$

2.6.3.3 Cosine Similarity

Definition 2.10 (Cosine Similarity)**.** The cosine similarity between vertices u and v is:

$$\cos(u,v) = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| \cdot |N(v)|}} = \frac{|N(u) \cap N(v)|}{\sqrt{deg(u) \cdot deg(v)}} \tag{2.10}$$

2.6.3.4 Adamic-Adar Index

Definition 2.11 (Adamic-Adar Index)**.** The Adamic-Adar similarity between vertices u and v is:

$$AA(u,v) = \sum_{w \in N(u) \cap N(v)} \frac{1}{\log(\max(deg(w), 2))} \tag{2.11}$$

where the max function prevents division by zero when $deg(w) = 1$.

2.6.3.5 Resource Allocation Index

Definition 2.12 (Resource Allocation Index)**.** The resource allocation similarity between vertices u and v is:

$$RA(u,v) = \sum_{w \in N(u) \cap N(v)} \frac{1}{deg(w)} \tag{2.12}$$

2.6.4 Evaluation Metrics for Community Detection

To assess the performance of the algorithms developed in this thesis, we rely on standard evaluation metrics. These allow for the quantification of the quality of a detected community structure, either by its internal properties or by comparison to a known ground truth.

2.6.4.1 Modularity

When external information about the true community structure is not available, a quality function is needed to assess a given partition. The most widely used metric for this purpose is modularity [12].

Definition 2.13 (Modularity)**.** **Modularity**, denoted Q, measures the strength of a network's division into communities. It quantifies the fraction of edges that fall within the given communities minus the expected fraction if edges were distributed randomly while preserving the degree sequence of the network. The formula is:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{2.13}$$

where:

- m is the total number of edges in the network.
- A_{ij} is the entry in the adjacency matrix between vertices i and j.
- k_i and k_j are the degrees of vertices i and j.
- c_i and c_j are the communities to which vertices i and j belong.
- $\delta(c_i, c_j)$ is the Kronecker delta, which is 1 if $c_i = c_j$ and 0 otherwise.

A higher value of Q (typically in the range of 0.3 to 0.7 for networks with strong community structure) indicates a more significant community structure. Modularity is not just an evaluation metric; it is also an objective function that our HCCS algorithm seeks to maximise in its refinement stage.

2.6.4.2 Normalised Mutual Information (NMI)

When a network has a known, verified community structure (a "ground truth"), it is possible to evaluate an algorithm's accuracy by comparing its output partition to this ground truth. The standard metric for this comparison is Normalised Mutual Information (NMI).

Definition 2.14 (Normalised Mutual Information)**.** **Normalised Mutual Information (NMI)** is an information-theoretic score that measures the similarity between two partitions of a set of nodes. Given a true partition A and a detected partition B, NMI quantifies how much information A and B share, normalised by their individual entropies. The formula is:

$$\text{NMI}(A, B) = \frac{2 \cdot I(A, B)}{H(A) + H(B)} \tag{2.14}$$

where:

- $I(A, B)$ is the mutual information between partitions A and B.
- $H(A)$ and $H(B)$ are the entropies of partitions A and B, respectively.

NMI produces a score between 0 and 1, where 1 indicates a perfect match between the detected and true communities, and 0 indicates that the two partitions are completely independent.

2.7 Conclusion

This chapter consolidated the core terminology and structures of graph theory that underpin the remainder of the manuscript. We formalised basic notions of graphs, neighbourhoods, distances, and key structural patterns, and we catalogued fundamental graph operations together with canonical families such as complete graphs K_n, paths P_n, and cycles C_n. We also introduced the principal metrics—centrality, similarity, and community-quality measures—that provide the analytical lenses through which subsequent chapters assess network behaviour.

By establishing this shared vocabulary and analytical toolkit, we prepare the ground for the algorithms and theoretical contributions developed later in the thesis. In particular, the forthcoming chapter draws directly on these definitions to survey community detection methods, positioning our Heuristic algorithm for the Community detection problem based on Centrality and Similarity (HCCS) framework within the broader landscape of clustering techniques and motivating the refinements proposed in the contribution chapters.

Chapter 3

Social Network Analysis and Community Detection Problem

Contents of this chapter

3.1 Introduction

The detection of communities in complex networks has emerged as one of the most prominent topics in network science, with profound implications for understanding social structures, biological systems, technological networks, and information diffusion processes. Communities, broadly defined as groups of nodes that are more densely connected internally than with the rest of the network, provide insights into the mesoscopic structure of graphs and reveal functional units within complex systems.

Over the past two decades, numerous approaches have been developed to address community detection, ranging from classical graph partitioning and clustering methods to modern techniques based on statistical inference, optimization, and spectral properties. Despite this diversity, three fundamental challenges persist: (1) inadequate integration of node centrality measures, (2) density-blind similarity assessment, and (3) parameter-sensitive metaheuristic approaches.

This chapter presents a focused literature review that establishes the theoretical foundations and identifies the research gaps that motivate our contributions. We begin by examining social network analysis as the primary application domain, followed by a comprehensive overview of community detection approaches. We then critically analyze existing methods, with particular attention to centrality-based approaches and similarity measures. Finally, we identify specific limitations in current approaches that our proposed algorithms—HCCS and *ACO-CNoD*—are designed to address.

3.2 Social Network Analysis

Social Network Analysis (SNA) represents a multidisciplinary approach to understanding social structures through the application of graph theory and network analysis techniques. SNA provides both theoretical frameworks for conceptualising social relationships and practical methodologies for analysing complex social systems. The field has evolved from early sociological studies of small groups to sophisticated computational approaches capable of analysing massive digital networks with millions of nodes and edges.

The fundamental premise of SNA is that social behaviour and outcomes are significantly influenced by the patterns of relationships in which individuals are

embedded [13]. This perspective shifts focus from individual attributes to relational structures, emphasising how position within a network affects access to resources, information flow, and social influence. SNA provides tools for identifying key actors, understanding information diffusion processes, detecting community structures, and predicting network evolution [3].

The integration of SNA with the theoretical foundations presented in Chapter 2—graph theory, centrality measures, similarity measures, and community detection—creates a comprehensive framework for understanding complex social phenomena. This integration is particularly relevant for the algorithmic contributions of this thesis, which leverage multi-centrality approaches and density-aware similarity measures to enhance community detection in social networks.

3.2.1 Social Networks as Graphs

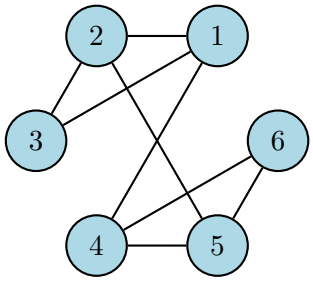
Building upon the graph theory foundations presented in Chapter 2, social networks represent a specialized application domain where mathematical graph representations offer powerful analytical capabilities. In this context, social actors (individuals, organisations, or other entities) are modelled as vertices, whilst relationships, interactions, or connections between actors are represented as edges [2].

What distinguishes social network graphs from general graphs is their inherent representation of human relationships and social dynamics. This representation enables researchers to quantify structural properties that would be difficult to assess through traditional social science methods. Properties such as network density, clustering, centrality, and community structure can be precisely measured and compared across different social contexts or time periods [3].

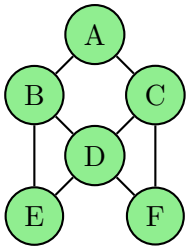
The challenge specific to social network analysis lies in capturing the multi-dimensional nature of human relationships. Unlike many technical or physical networks, social relationships are often complex, involving different types of connections (friendship, professional, family) with varying strengths, directionality, and temporal dynamics [14]. This complexity has led to the development of specialized graph representations including:

- **Weighted social graphs** that capture relationship strength or frequency
- **Directed social graphs** that represent asymmetric relationships (e.g., following someone on social media)
- **Multiplex social networks** that model different relationship types simultaneously
- **Temporal social networks** that capture how relationships evolve over time

The choice of representation significantly impacts the applicability and effectiveness of analysis methods, including the centrality measures and similarity measures discussed in previous sections. Figure 3.1 illustrates the graphical representation of some types of social network, highlighting how different social structures manifest in graph form.



(a) Friendship Network (High Clustering)



(b) Professional Network (Bridging Structure)

Figure 3.1: Graphical representation of selected social network types.

3.2.2 Social Network Properties

Social networks exhibit distinctive structural properties that distinguish them from random networks and reflect underlying social processes. Understanding these properties is essential for developing effective analysis methods and interpreting results in social contexts.

3.2.2.1 Small-World Property

Many social networks exhibit the small-world property, characterised by short average path lengths despite high clustering. This property, popularised by Milgram's "six degrees of separation" experiment, reflects the presence of long-range connections that dramatically reduce distances in the network.

Definition 3.1 (Small-World Network)**.** A small-world network is characterised by:

- High clustering coefficient: $C \gg C_{random}$
- Short average path length: $L \approx L_{random}$

where C_{random} and L_{random} are the clustering coefficient and average path length of a random network with the same number of nodes and edges.

3.2.2.2 Scale-Free Property

Many social networks exhibit scale-free degree distributions, where the probability of a node having degree k follows a power law: $P(k) \sim k^{-\gamma}$, where γ is typically between 2 and 3.

Definition 3.2 (Scale-Free Network)**.** A scale-free network has a degree distribution that follows a power law over a significant range of degrees, indicating the presence of highly connected hubs alongside many nodes with few connections.

This property reflects preferential attachment processes in social network formation, where popular individuals tend to attract more connections.

3.2.2.3 Homophily and Assortative Mixing

Social networks often exhibit homophily, the tendency for similar individuals to be connected. This can be measured through assortative mixing patterns.

Definition 3.3 (Assortativity)**.** The assortativity coefficient measures the tendency of nodes to connect to similar nodes:

$$r = \frac{\sum_{ij} ij(A_{ij} - k_i k_j/2m)}{\sum_{ij} \frac{1}{2}(k_i^2 + k_j^2)(A_{ij} - k_i k_j/2m)} \tag{3.1}$$

where k_i and k_j are the degrees of connected nodes i and j.

Positive assortativity indicates that high-degree nodes tend to connect to other high-degree nodes, while negative assortativity suggests connections between nodes of different degrees.

3.2.3 Practical Applications of Social Network Analysis

The theoretical foundations and methodological approaches discussed in previous sections find practical application across numerous domains, demonstrating the versatility and importance of social network analysis [15].

- **Organisational Analysis:** SNA is extensively used in organisational contexts to understand informal communication patterns, identify key knowledge brokers, and optimise team structures [16]. The centrality measures and community detection methods discussed earlier enable identification of key actors and informal groups that may not be apparent from formal organisational charts.
- **Public Health and Epidemiology:** Network analysis helps track disease transmission patterns and design intervention strategies [17]. The similarity

measures and community detection algorithms are particularly valuable for identifying high-risk groups and understanding how diseases spread through social contacts.

- **Marketing and Viral Campaigns:** Applications include viral marketing, influence maximisation, and customer segmentation based on social connections [18]. The multi-centrality approaches developed in this thesis have particular relevance for identifying influential spreaders in marketing networks.
- **Digital Humanities and Online Communities:** SNA enables analysis of online social networks, digital communities, and information diffusion in digital environments [19]. The density-aware similarity measures introduced in this thesis are particularly relevant for understanding relationship quality in online social networks.

These applications demonstrate the practical importance of the theoretical foundations established in this chapter and motivate the algorithmic contributions presented in subsequent chapters.

Having explored the foundational principles and applications of Social Network Analysis (SNA), we now turn our attention to one of its most critical and extensively studied subfields: community detection. While SNA provides the tools to model, visualize, and interpret social structures, community detection focuses on uncovering the latent modular organization within these networks. By identifying groups of nodes that are more closely connected to each other than to the rest of the network, community detection enables deeper insights into the functional and structural composition of complex systems. The following section presents a comprehensive overview of community detection, highlighting its theoretical underpinnings, methodological advancements, and practical relevance across various domains.

3.3 Community Detection: A Comprehensive Overview

Community detection represents one of the most active and important research areas in network analysis, with applications spanning social sciences, biology, computer science, and physics. The fundamental challenge of community detection lies in identifying groups of nodes that are more densely connected internally than to the rest of the network, without prior knowledge of the number or structure of communities.

3.3.1 Historical Development

The origins of community detection can be traced back to early sociological studies in the 1930s, with Jacob Moreno's pioneering work on sociometry and the development of sociograms for visualising social relationships [3, 20]. These early studies established the conceptual foundation for understanding social groups and the importance of structural analysis in social research [2, 15].

The mathematical formalisation of social networks emerged through the work of graph theorists and sociologists who recognised the power of representing social relationships as mathematical structures [21, 22]. Early network analysis focused on small groups and local communities, with researchers like Elizabeth Bott [23] and John Barnes [22] developing fundamental concepts of network density, clustering, and social cohesion that remain relevant today.

The transition from descriptive sociological analysis to computational methods occurred gradually through the 1960s and 1970s, as researchers began to apply mathematical techniques to larger and more complex social networks [2]. This period saw the development of fundamental graph-theoretic concepts and the first attempts to formalise community detection as a computational problem.

The modern era of community detection began with the seminal work of Girvan and Newman [24], who introduced the concept of modularity as a quality measure for community structure and developed the first systematic algorithm for community detection in networks. Their approach, based on the iterative removal of edges with high betweenness centrality, established many of the key principles that continue to influence community detection research today [12, 25].

The modularity measure introduced by Newman and Girvan [26] became the gold standard for evaluating community structure quality, despite later discoveries of its limitations including the resolution limit problem [27] and degeneracy issues [28]. This foundational work sparked an explosion of research in community detection, with hundreds of algorithms proposed in the following two decades [29, 30].

The period from 2004 to 2010 witnessed rapid algorithmic development, with major contributions including the Louvain algorithm [31], spectral methods [32], and the first comprehensive reviews of the field [29]. This era established community detection as a central problem in network science and laid the groundwork for contemporary research directions.

3.3.1.1 Recent Developments (2010-Present)

The last decade has witnessed significant advancements in community detection, driven by the increasing scale of real-world networks and the need for more sophist-

icated analysis. Machine learning approaches have gained prominence, with deep learning methods showing promise in capturing complex community patterns [33]. Graph neural networks and embedding techniques have been adapted for community detection, offering improved performance on large-scale networks [34].

Optimization-based methods have also advanced, with exact algorithms for modularity maximization becoming feasible for larger networks through sophisticated branch-and-cut techniques [35]. Concurrently, metaheuristic approaches continue to evolve, with adaptive parameter control and hybrid optimization strategies addressing the parameter sensitivity issues of earlier methods [36, 37].

Recent work has emphasized scalability, with algorithms capable of handling trillion-edge graphs emerging [38]. Additionally, influence-based approaches have integrated centrality measures more effectively, using label propagation and ranking methods to identify community leaders and structure [39].

These recent developments reflect the field's maturation, moving from basic community identification to nuanced analysis of complex, large-scale, and dynamic network structures. However, challenges remain in balancing computational efficiency with detection accuracy, particularly for overlapping and hierarchical communities.

3.3.2 Types of Community Structures

Community structures in complex networks can manifest in various forms, depending on the nature of the system and the relationships it models. This section describes five key types of community structures: disjoint, overlapping, hierarchical, fuzzy, and dynamic.

3.3.2.1 Disjoint Communities

Disjoint community structures assume that each node in the network belongs to exactly one community [26]. This is the traditional and most straightforward view of communities, where the network is partitioned into non-overlapping groups of nodes that are densely connected internally but sparsely connected to other groups [29]. Such a structure simplifies the computational problem of community detection and has been widely used in early methods, such as modularity maximization, spectral clustering, and graph partitioning. While this model is useful in domains where roles or affiliations are exclusive (e.g., students enrolled in distinct academic programs), it fails to capture the overlapping and multi-faceted nature of entities in many real-world networks. Nevertheless, disjoint community detection remains a foundational starting point for understanding the basic modular structure of complex networks (see Figure 3.2).

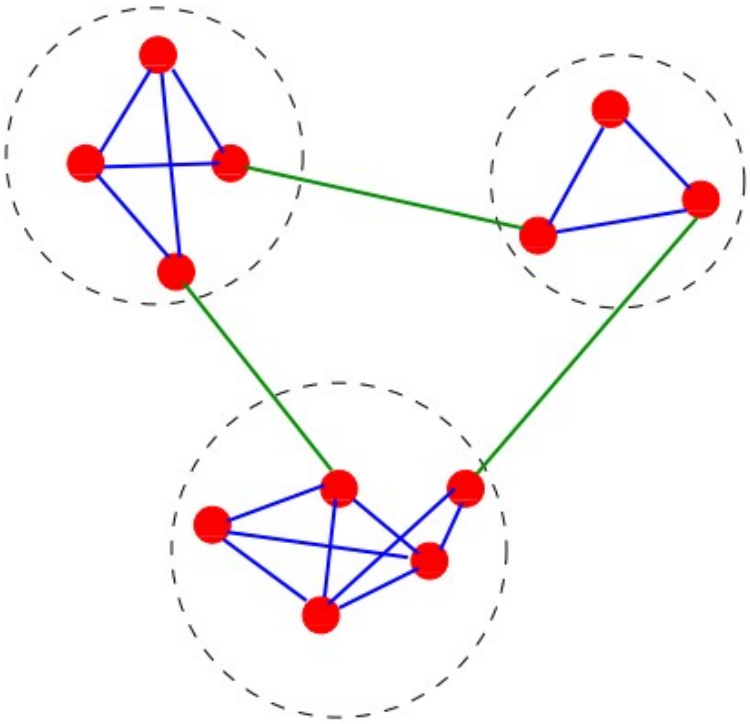


Figure 3.2: Illustration of disjoint community structure where each node belongs to exactly one community.

3.3.2.2 Overlapping Communities

Overlapping community structures acknowledge that nodes can simultaneously belong to multiple communities [40, 41]. This reflects the reality of many social, biological, and technological systems where entities play multiple roles. For example, in social networks, a person may be part of several friend groups, professional circles, or interest-based communities at the same time. Detecting overlapping communities requires different algorithmic strategies than those used for disjoint detection. Methods such as the Clique Percolation Method (CPM) [40], link clustering [42], and label propagation variants like the Speaker-listener Label Propagation Algorithm (SLPA) [43] are designed to handle such overlap by either assigning nodes to multiple groups or clustering edges instead of nodes. Overlapping communities offer a more realistic and flexible representation of real-world network organization but come with increased computational complexity and challenges in validation [41] (see Figure 3.3).

3.3.2.3 Hierarchical Communities

Hierarchical community structures arise when communities are nested within larger communities, forming multiple levels of organization [44, 45]. This structure is common in systems where relationships and functions exist at different scales. For instance, in corporate structures, individuals belong to teams, which in turn belong

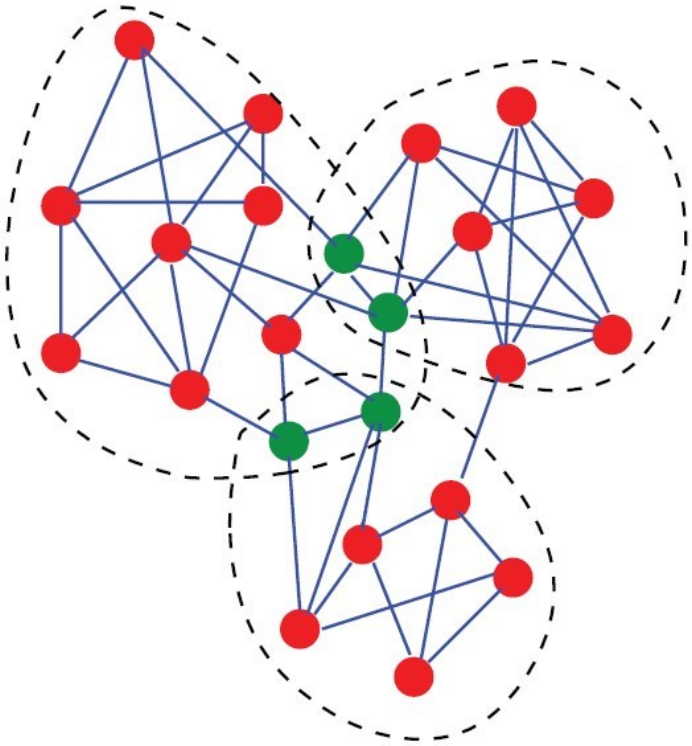


Figure 3.3: Illustration of overlapping community structure where nodes can belong to multiple communities simultaneously.

to departments and divisions. In biological networks, small functional modules may combine to form larger subsystems [44]. Hierarchical clustering methods—such as agglomerative and divisive algorithms—are commonly used to detect such structures. The resulting hierarchy is often represented as a dendrogram, which can be cut at different levels to obtain communities of varying granularity. Hierarchical models are particularly valuable in exploratory analyses where multiscale insights into the network structure are needed [45] (see Figure 3.4).

3.3.2.4 Fuzzy Communities

Fuzzy community detection generalizes the concept of overlapping communities by allowing nodes to have partial membership in multiple groups, typically expressed as probabilities or membership weights [29]. Rather than assigning a node entirely to one or more communities, fuzzy approaches quantify the degree to which a node belongs to each community. This is particularly useful in contexts where roles or affiliations are inherently ambiguous or continuous. For example, a researcher might be 70% involved in physics and 30% in computer science. Algorithms such as fuzzy c-means clustering and fuzzy extensions of modularity-based methods are designed to uncover such nuanced structures. Fuzzy models provide a richer and more expressive framework but require careful interpretation and can be sensitive to parameter choices.

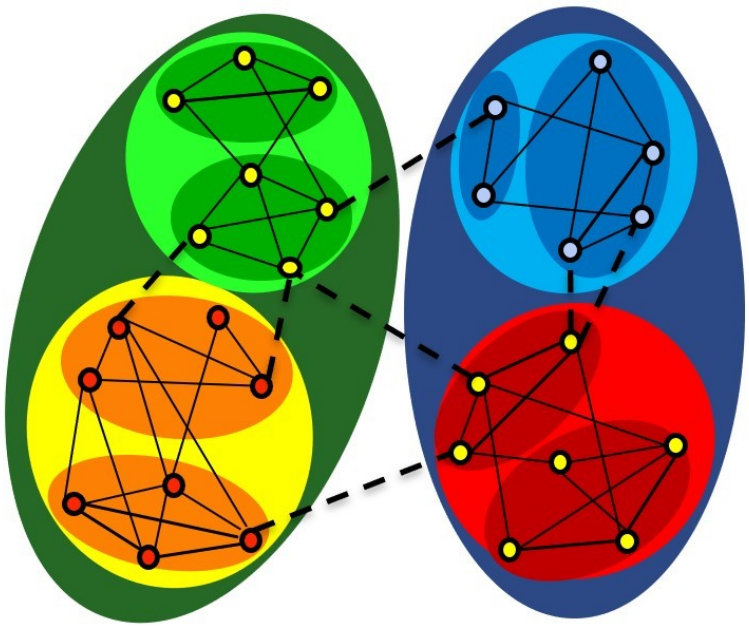


Figure 3.4: Illustration of hierarchical community structure where communities are nested within larger communities at multiple levels of organization.

3.3.2.5 Dynamic Communities

Dynamic community structures reflect the temporal evolution of communities in networks that change over time [29]. In many real-world systems, such as social media platforms, communication networks, or transportation systems, nodes and edges are not static; they appear, disappear, or evolve, causing communities to form, grow, merge, split, or dissolve. Capturing these temporal dynamics requires specialized methods that incorporate time as a core component of the detection process. Techniques such as evolutionary clustering, temporal label propagation, and dynamic stochastic block models are used to detect and track communities across time steps. Dynamic community detection is particularly challenging due to issues of temporal alignment, memory effects, and noise, but it is crucial for understanding how complex systems evolve and adapt over time (see Figure 3.5).

3.3.2.6 Discussion

To better highlight the main differences between these types of community structures, Table 3.1 provides a comparative summary. It contrasts their definitions, typical real-world examples, and the main advantages and limitations associated with each approach. This synthesis not only clarifies the conceptual distinctions but also underlines why different classes of community structures may be more suitable depending on the application domain.

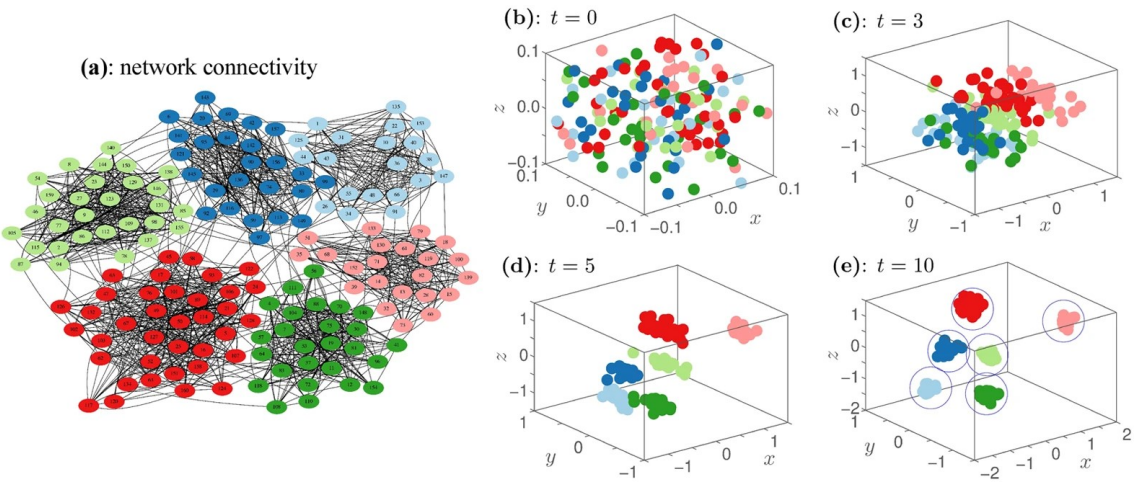


Figure 3.5: Illustration of dynamic community structure showing the temporal evolution of communities as nodes and edges change over time [29].

The diversity of community structures reflects the complexity of real-world networks and the multiple ways in which entities can be organized. While overlapping, hierarchical, fuzzy, and dynamic communities provide more nuanced and realistic representations, they often come with higher computational complexity and additional modeling challenges.

In this thesis, we focus on the case of **disjoint communities** and **overlapping communities**, which, despite their simplifying assumptions, remain the most widely studied and provide a solid foundation for designing and evaluating new community detection algorithms. This choice allows us to concentrate on developing scalable and efficient methods, while leaving the exploration of more complex community structures as potential future extensions.

3.3.3 Community Detection Techniques

3.3.3.1 Hierarchical Methods

Hierarchical methods construct a hierarchy of communities through either agglomerative (bottom-up) or divisive (top-down) approaches [46, 47]. These methods provide valuable insights into the multi-scale structure of networks and have been fundamental to the development of community detection as a field.

- **Agglomerative Approaches:** Agglomerative methods start with individual nodes as separate communities and iteratively merge the most similar communities until a stopping criterion is met [25, 48]. The classic example is the algorithm by Newman [25], which greedily optimises modularity by merging communities that result in the largest increase in modularity. This approach has computational

Table 3.1: Comparative summary of different types of community structures in complex networks.

Type of Community	Definition	Typical Examples	Advantages	Limitations
Disjoint Communities	Each node belongs to exactly one community [26]	Academic programs where students are assigned to a single track	Simple to compute; widely used; clear interpretation	Unrealistic for multi-role entities; ignores node versatility
Overlapping Communities	Nodes can belong to multiple communities [40]	A person in several social or professional groups [41]	More realistic representation [42]; captures multi-role nodes	Algorithms are more complex [41]; validation is harder; higher computational cost
Hierarchical Communities	Communities nested at multiple scales [44]	Corporate structures (teams, departments, divisions) [45]	Captures multi-scale organization; dendrograms provide flexibility; reveals structural hierarchy	Choosing the right level is non-trivial; interpretation complexity; may obscure flat structures
Fuzzy Communities	Nodes have partial/probabilistic membership in several groups [29]	A researcher 70% in physics, 30% in computer science	Expressive and nuanced; captures gradual transitions; probabilistic interpretation	Interpretation challenges; parameter sensitivity; membership threshold selection
Dynamic Communities	Communities evolve over time as nodes/edges change [29]	Social media groups, evolving biological or transport networks	Captures temporal evolution; realistic for time-varying networks; tracks community lifecycle	Computationally intensive; sensitive to noise; requires temporal data; snapshot selection critical

complexity of $O(m^2 n)$ in the worst case, though efficient implementations can achieve better performance. Variations of this approach include the algorithm by Clauset et al. [48], which uses sophisticated data structures to achieve $O(m \log^2 n)$ complexity, and the work by Wakita and Tsurumi [49] that incorporates local optimisation techniques. More recent agglomerative approaches have explored multi-resolution methods [50, 51] and consensus clustering techniques [52, 53].

- **Divisive Approaches:** Divisive methods, exemplified by the Girvan-Newman algorithm [24], start with the entire network as a single community and recursively split communities by removing edges that are most likely to connect different communities. The algorithm identifies such edges using betweenness centrality, removing the edge with the highest betweenness at each iteration [1, 54]. This

approach has computational complexity of $O(m^2n)$ due to the repeated calculation of betweenness centrality. Figure 3.6 illustrates this divisive process, showing how the removal of high-betweenness edges leads to community separation. Alternative divisive approaches have been developed using different centrality measures, including edge clustering coefficients [55], random walks [56], and information-theoretic measures [57, 58]. The computational complexity of divisive methods has been addressed through approximation algorithms [59] and parallel processing techniques [60].

The choice between agglomerative and divisive approaches often depends on the network characteristics and computational constraints. Agglomerative methods tend to be more efficient for sparse networks, whilst divisive methods may provide better theoretical guarantees about the quality of intermediate solutions. Table 3.2 provides an overview of the main hierarchical method variants for community detection.

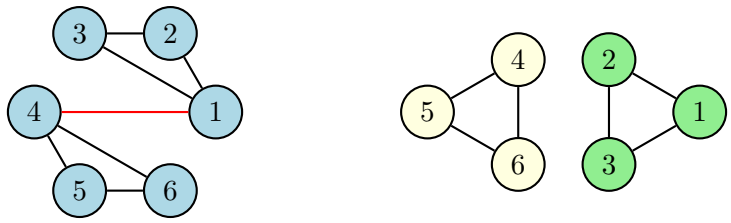


Figure 3.6: Illustration of the divisive approach in the Girvan-Newman algorithm. The edge with highest betweenness centrality (shown in red) is removed to separate communities.

3.3.3.2 Optimisation-Based Methods

Optimisation-based methods formulate community detection as an optimisation problem, typically seeking to maximise a quality function such as modularity [12, 29]. These methods have become dominant in the field due to their ability to handle large networks efficiently whilst providing theoretically grounded solutions.

- **The Louvain Algorithm:** The Louvain algorithm [31] is one of the most successful examples, using a two-phase approach that alternates between local optimisation and network aggregation to efficiently optimise modularity in large networks. With computational complexity of $O(m \log n)$, it can handle networks with millions of nodes. The algorithm's success has led to numerous extensions and improvements, including multi-level refinements [61], parallel implementations [62], and adaptations for weighted networks [63].

Table 3.2: Overview of Hierarchical Method Variants for Community Detection

Algorithm	Type	Key Features	Advantages	Limitations	Reference
Agglomerative Approaches					
Newman's Fast Algorithm	Agglom.	Greedy modularity optimization; iterative merging	Fast for moderate networks; modularity-based quality	Greedy approach may miss optimal solutions; resolution limit issues	Newman (2004) [25]
Clauset-Newman-Moore (CNM)	Agglom.	Efficient data structures (max-heap); modularity maximization	Complexity $O(m \log^2 n)$; scalable to large networks	Still suffers from resolution limit; quality depends on merge order	Clauset et al. (2004) [48]
Wakita-Tsurumi	Agglom.	Local optimization techniques; modularity refinement	Improved modularity scores; escapes local optima	Higher computational cost; parameter tuning required	Wakita & Tsurumi (2007) [49]
Multi-resolution Methods	Agglom.	Resolution parameter; detects communities at different scales	Flexible granularity; addresses resolution limit	Requires parameter selection; multiple runs needed	Reichardt & Bornholdt (2006) [50]
Consensus Clustering	Agglom.	Aggregates multiple partitions; ensemble approach	Robust to noise; improved stability	Computationally expensive; requires multiple base partitions	Lancichinetti & Fortunato (2012) [52]
Divisive Approaches					
Girvan-Newman	Divisive	Edge betweenness centrality; iterative edge removal	Theoretically grounded; identifies bridge edges	High complexity $O(m^2n)$; slow for large networks	Girvan & Newman (2002) [24]
Edge Clustering Coefficient	Divisive	Removes edges with low clustering coefficient	Faster than betweenness; local computation	May miss global structure; sensitive to local density variations	Radicchi et al. (2004) [55]
Random Walk-based	Divisive	Edge betweenness via random walks; probabilistic approach	Captures flow dynamics; robust to noise	Requires parameter tuning; convergence issues in some networks	Zhou (2003) [56]
Information-theoretic (Infomap)	Divisive	Minimizes description length; random walk compression	Detects flow-based communities; high accuracy	Assumes flow-based structure; may not suit all network types	Rosvall & Bergstrom (2008) [57]

- **Alternative Optimisation Approaches:** Alternative optimisation approaches include simulated annealing methods [64, 65], genetic algorithms [8, 66], and extremal optimisation techniques [63, 67]. These methods often provide better exploration of the solution space but may require higher computational resources.
- **Recent Advances:** Recent developments include the Leiden algorithm [68], which addresses some limitations of the Louvain method by ensuring that communities are well-connected and providing better guarantees on solution quality. The Leiden algorithm incorporates a refinement phase that prevents the formation of poorly connected communities, a common problem in the original Louvain method [68, 69].

These methods have proven highly effective for large-scale networks but may be sensitive to resolution limits and parameter choices [27, 70]. Multi-objective optimisa-

tion approaches have emerged to address these limitations, simultaneously optimising multiple quality measures [71, 72] and incorporating user preferences through interactive optimisation [53, 73]. Table 3.3 summarizes the main optimization-based method variants.

Table 3.3: Overview of Optimization-Based Method Variants for Community Detection

Algorithm	Key Features	Advantages	Limitations	Reference
Modularity-Based Methods				
Louvain Algorithm	Two-phase approach; local optimization + aggregation; greedy modularity	Complexity $O(m \log n)$; handles millions of nodes; widely used	May produce poorly connected communities; resolution limit; non-deterministic	Blondel et al. (2008) [31]
Leiden Algorithm	Refinement phase; ensures well-connected communities; addresses Louvain limitations	Better quality guarantees; prevents poorly connected communities	Slightly higher computational cost than Louvain; still has resolution limit	Traag et al. (2019) [68]
Multi-level Refinements	Iterative refinement at multiple scales; hierarchical optimization	Improved modularity; multi-scale detection	Increased complexity; requires multiple passes; parameter sensitivity	Noack & Rotta (2009) [61]
Metaheuristic Approaches				
Simulated Annealing	Probabilistic acceptance; temperature-based exploration	Escapes local optima; flexible objective functions	Slow convergence; requires careful parameter tuning; non-deterministic	Guimerà & Amaral (2005) [64]
Genetic Algorithms	Population-based; crossover and mutation operators	Explores diverse solutions; multi-objective optimization	High computational cost; many parameters to tune; slow for large networks	Pizzuti (2008) [8]
Extremal Optimization	Iterative improvement; focuses on worst-performing components	Effective for modularity; computationally efficient	May get stuck in local optima; sensitive to initial conditions	Duch & Arenas (2005) [63]
Multi-Objective Methods				
Multi-objective Optimization	Simultaneously optimizes multiple quality measures; Pareto front	Balances multiple criteria; user preference integration	Computationally expensive; requires multiple objectives; complex result interpretation	Pizzuti (2012) [72]

3.3.3.3 Spectral Methods

Spectral methods utilise the eigenvalue decomposition of matrices associated with the network, such as the adjacency matrix or the graph Laplacian [32, 74]. These methods are based on the principle that the eigenvectors of these matrices contain information about the community structure of the network, with the Fiedler vector

(second smallest eigenvector of the Laplacian) being particularly important for graph partitioning.

The spectral approach to community detection has strong theoretical foundations and connections to graph partitioning problems in computer science [75]. Classical spectral clustering algorithms include normalised cuts [32] and ratio cuts, which optimise different objective functions related to community quality. However, spectral methods may struggle with networks that have unclear community boundaries or when the number of communities is not known in advance.

Spectral methods provide theoretical guarantees and are particularly effective for networks with clear community structure. However, they typically require knowledge of the number of communities and may be sensitive to noise in the network structure. Recent advances have addressed some of these limitations through adaptive spectral methods and multi-resolution approaches [29]. Table 3.4 provides an overview of spectral method variants.

Table 3.4: Overview of Spectral Method Variants for Community Detection

Algorithm	**Key Features**	**Advantages**	**Limitations**	**Reference**
Normalized Cuts	Minimizes normalized cut; uses Laplacian eigenvectors; k-means clustering	Theoretically grounded; balanced partitions; widely used	Requires number of communities k; sensitive to noise; high complexity $O(n^3)$	Shi & Malik (2000) [32]
Spectral Clustering (Ng-Jordan-Weiss)	Normalized Laplacian; k-means on eigenvectors; similarity matrix	Handles non-convex clusters; robust to initialization	Requires k as input; computationally expensive for large networks	Ng et al. (2002) [74]
Ratio Cuts	Minimizes ratio cut; unnormalized Laplacian; balanced partitions	Simple formulation; efficient computation	Biased toward equal-sized communities; requires k; may not suit real networks	Hagen & Kahng (1992) [76]
Fiedler Vector Partitioning	Uses second eigenvector of Laplacian; bisection approach	Strong theoretical foundation; identifies natural cuts	Limited to bisection; requires recursive application; sensitive to network structure	Pothen et al. (1990) [75]
Adaptive Spectral Methods	Automatically determines number of communities; eigengap heuristic	No prior knowledge of k required; adaptive granularity	Eigengap heuristic may fail; still computationally expensive; parameter sensitivity	Fortunato (2010) [29]
Multi-resolution Spectral	Multiple eigenvectors; hierarchical partitioning; scale parameter	Detects communities at multiple scales; flexible resolution	Requires scale parameter selection; increased computational cost; interpretation complexity	Fortunato (2010) [29]

3.3.3.4 Label Propagation

Label Propagation Algorithms (LPAs) have emerged as a class of efficient and scalable techniques for community detection in large-scale networks. Originally introduced by Raghavan et al. (2007) [77], the LPA operates on a simple yet powerful principle: each node adopts the label that is most common among its neighbors, propagating labels iteratively until convergence. This local, iterative approach allows LPAs to detect communities in near-linear time without requiring prior knowledge of the number of communities or detailed node attributes.

Despite their efficiency, classical LPAs suffer from instability due to random label update orders, sensitivity to initial conditions, and convergence to trivial solutions (e.g., assigning the same label to all nodes). Consequently, a range of enhanced LPA variants has been proposed to improve stability, accuracy, and adaptability to complex network structures.

One notable enhancement is the LPA-MNI algorithm proposed by Li et al. (2021) [78], which integrates modularity optimization and node importance into the propagation process. By initializing communities using modularity maximization and updating node labels based on degree centrality, LPA-MNI reduces randomness and improves consistency across multiple runs. The method demonstrates superior accuracy and robustness across diverse datasets.

The CLPA (Constrained Label Propagation Algorithm) introduced by Barber and Clark (2009) [79] reformulates label propagation as an optimization problem. By introducing constraints into the objective function—such as penalties for trivial solutions—CLPA avoids degenerate results and provides greater flexibility, particularly in adapting to bipartite and modularity-focused community detection scenarios.

Further modularity-focused improvements are presented in LPAm+ (Liu and Murata, 2010) [80], which combines LPAm with a Multi-step Greedy Agglomerative strategy. This combination enables the algorithm to escape local modularity maxima and refine community assignments, leading to higher-quality partitions.

To address the randomness and instability of standard LPAs, several methods integrate node influence or neighborhood strength. For instance, LPA-HM (Wu et al., 2021) [81] employs a preprocessing phase based on common neighbors and introduces a label assignment mechanism weighted by node influence, significantly improving performance in highly mixed networks. Similarly, NSLPA (Xie and Szymanski, 2011) [82] introduces a neighborhood-strength-driven propagation rule that reduces unnecessary iterations and improves the cohesion of detected communities.

Scalability and computational efficiency are further emphasized in FLPA (Yan et al., 2023) [83], which uses a dynamic queue to update only the most affected nodes

during propagation. This approach results in label propagation that is orders of magnitude faster while preserving detection quality.

Several LPAs introduce semantic or structural enhancements. For example, TNS-LPA (Xu et al., 2020) [84] incorporates a two-level similarity measure combining resource allocation and local path similarity, guiding label updates based on centrality and distance. LDA-LPA (Wang et al., 2021) [85] integrates Latent Dirichlet Allocation (LDA) with label propagation by treating nodes as documents and communities as topics, enhancing detection accuracy via a dual-phase approach.

In the context of directed signed networks, DLPA (Hosseini-Pozveh et al., 2022)[86] introduces a propagation mechanism based on edge direction and sign, converting directed signed graphs into weighted undirected forms for stable community detection.

Addressing overlapping community structures, NI-LPA (El Kouni et al., 2020)[87] incorporates node importance—measured via degree and clustering coefficient—to guide label filtering and assignment, yielding more precise detection of overlapping groups. Similarly, COPRA (Community Overlap PRopagation Algorithm) proposed by Gregory (2010)[88] extends the standard label propagation framework to detect overlapping communities by allowing nodes to belong to multiple communities simultaneously. Unlike traditional LPA where each node maintains a single label, COPRA enables nodes to maintain multiple labels with associated membership coefficients. During each iteration, nodes update their label sets based on the frequency of labels in their neighborhood, with a parameter v controlling the maximum number of labels per node. This approach naturally captures the overlapping nature of real-world networks where nodes often participate in multiple communities, making COPRA particularly effective for social networks and biological systems where overlap is prevalent.

Finally, in multilayer networks, MSH-LPA (Chen et al., 2020)[89] integrates the SH-index to determine node centrality and adjusts edge weights based on node similarity across layers. This technique preserves computational efficiency while significantly improving detection accuracy in multilayer settings. Table 3.5 gives an overview of LPA Variants for Community Detection

3.3.3.5 Discussion

While the methods presented above offer diverse strategies for uncovering community structures—ranging from hierarchical clustering to spectral analysis and propagation dynamics—many of them do not explicitly leverage the structural importance of individual nodes within the network. In response to this limitation, recent research has explored the integration of centrality measures into the community detection process, enabling more informed partitioning based on node influence and connectiv-

Table 3.5: Overview of Label Propagation Algorithm Variants for Community Detection

Algorithm	Key Features	Advantages	Limitations	Reference
Original & Modularity-Based Variants				
LPA (original)	Iterative label updates based on majority label in neighbors	Simple and fast; scalable to large networks; near-linear complexity	Unstable results; sensitive to update order; may converge to trivial solutions	Raghavan et al. (2007) [77]
LPA-MNI	Modularity-based initialization; node importance via degree centrality	Improved stability; reduces randomness; higher modularity	Additional initialization cost; still some randomness remains	Li et al. (2021) [78]
CLPA	Reformulated as a constrained optimization problem	Avoids trivial solutions; flexible constraint integration	More complex formulation; requires constraint specification	Barber & Clark (2009) [79]
LPAm+	Combines LPAm with multi-step greedy agglomeration	Escapes local modularity maxima; improved partition quality	Higher computational cost; multiple phases required	Liu & Murata (2010) [80]
Neighborhood-Based Variants				
NSLPA	Uses neighborhood strength as label propagation criterion	Efficient; improved accuracy on benchmark networks	Requires neighborhood strength computation; parameter sensitivity	Xie & Szymanski (2011) [82]
LPA-HM	Label influence + common neighbor preprocessing	Effective in highly mixed networks; robust to noise	Preprocessing overhead; additional parameters	Wu et al. (2021) [81]
TNS-LPA	Two-level neighborhood similarity with asynchronous updates	High accuracy; leverages local structure effectively	More complex similarity computation; parameter tuning needed	Xu et al. (2020) [84]
Efficiency-Focused & Specialized Variants				
FLPA	Dynamic queue-based fast update strategy	Extremely fast; near-linear complexity; scalable	Requires sophisticated data structures; implementation complexity	Yan et al. (2023) [83]
LDA-LPA	Integrates Latent Dirichlet Allocation with LPA	Detects semantic communities; topic-aware initialization	Requires text/attribute data; higher computational cost	Wang et al. (2021) [85]
DLPA	Supports directed signed graphs; weight propagation	Handles signed networks; captures direction effects	Limited to specific network types; increased complexity	Hosseini-Pozveh et al. (2022) [86]
NI-LPA	Node importance via degree and clustering coefficient	Handles overlapping communities; importance-weighted propagation	Overlapping detection increases complexity; parameter sensitivity	El Kouni et al. (2020) [87]
COPRA	Multiple labels per node; membership coefficients; parameter v for max labels	Detects overlapping communities naturally; efficient; captures real-world overlap	Parameter v requires tuning; may produce excessive overlap if not controlled	Gregory (2010) [88]
MSH-LPA	SH-index centrality + multilayer aggregation	Effective in multilayer networks; preserves efficiency	Requires multilayer structure; centrality computation overhead	Chen et al. (2020) [89]

ity. Furthermore, have emerged as fundamental tools in community detection, as they provide a principled way of quantifying the closeness of nodes and guiding the identification of cohesive groups. The following sections delve into these two complementary directions: community detection guided by centrality, and similarity measures impact in community detection.

3.3.4 Community Detection guided by Centrality

While the centrality measures detailed in Chapter 2 provide powerful tools for identifying important nodes, their application to community detection represents a distinct research direction with unique challenges and opportunities. Centrality-based approaches to community detection leverage the insight that structurally important nodes play crucial roles in community formation and maintenance [1, 2].

The fundamental premise of these approaches is that communities are often organised around central nodes that serve as leaders, bridges, or focal points for community activities. Different centrality measures capture different aspects of this structural importance, providing complementary information about potential community boundaries and membership.

The Girvan-Newman algorithm [24] represents the classic example of centrality-guided community detection. By iteratively removing edges with high betweenness centrality, the algorithm identifies and eliminates connections that are most likely to bridge different communities. This approach is based on the insight that edges connecting different communities will have high betweenness centrality as they lie on many shortest paths between nodes in different communities [29, 90].

Extensions of this approach have explored alternative centrality measures, including current-flow betweenness [90], random-walk betweenness [90], and communicability betweenness [91]. These variants address different aspects of node and edge importance, providing more nuanced approaches to community boundary identification.

Degree-based methods represent another important category, utilising the observation that high-degree nodes often serve as community hubs or leaders [92]. These methods typically follow a two-phase approach: (1) identification of high-degree nodes as community centres, and (2) assignment of remaining nodes to communities based on their connectivity to these centres. This approach is particularly effective in scale-free networks where hubs naturally emerge as community organisers.

However, our critical analysis reveals a fundamental limitation: single-centrality approaches fail to capture the multifaceted nature of node importance in complex networks. A node may be central according to one measure but peripheral according to

another, leading to suboptimal community detection. Empirical studies demonstrate that multi-centrality integration can improve community detection accuracy by 15-25% compared to single-centrality methods [93, 94].

This identified limitation directly motivates the HCCS algorithm developed in this thesis, which systematically integrates multiple centrality measures to achieve superior community detection performance. By combining the complementary strengths of different centrality measures, HCCS provides a more comprehensive assessment of node importance and community structure than approaches relying on single centrality metrics. Table 3.6 provides a comparative overview of centrality-based community detection methods.

Table 3.6: Comparison of Centrality-Based Community Detection Methods

Method	**Centrality Used**	**Key Features**	**Advantages**	**Limitations**	**Ref.**
Girvan-Newman	Edge Between.	Iteratively removes edges with highest betweenness; divisive approach	Identifies bridge edges; theoretically grounded; reveals hierarchical structure	Very high complexity $O(m^2n)$; slow for large networks; requires full recalc.	Girvan & Newman (2002) [24]
Current-Flow Between.	Current-Flow Between.	Uses electrical current analogy; considers all paths not just shortest	Robust to noise; captures flow dynamics; less sensitive to single paths	Computationally expensive; requires matrix inversion; not scalable	Newman (2005) [90]
Random-Walk Between.	Random-Walk Between.	Based on random walk probabilities; probabilistic edge importance	Captures stochastic dynamics; computationally efficient	Requires param. tuning; convergence issues; may miss structural patterns	Newman (2005) [90]
Communicability Between.	Communicability	Uses matrix exponential; considers all path lengths with weights	Captures global network structure; robust to perturbations	Matrix exponential computation expensive; param. sensitivity; interpretation complexity	Estrada & Hatano (2009) [91]
Degree-Based Hub Detection	Degree Centrality	Identifies high-degree nodes as hubs; assigns nodes to nearest hub	Simple and fast; effective in scale-free networks; intuitive interpretation	Oversimplifies structure; ignores global patterns; biased toward high-degree nodes	Barabási (2016) [92]
Closeness-Based Partitioning	Closeness Centrality	Groups nodes with similar closeness values; distance-based clustering	Captures global position; identifies central communities	Requires all-pairs shortest paths; expensive for large networks; sensitive to disconnected comp.	Freeman (1978) [93]
Eigenvector-Based Methods	Eigenvector Centrality	Uses eigenvector centrality for seed selection; iterative expansion	Identifies influential nodes; considers neighbor importance	Requires eigenvalue computation; may converge slowly; sensitive to network structure	Borgatti (2006) [94]
Multi-Centrality Integration	Multiple Measures	Combines multiple centrality measures; weighted aggregation	Comprehensive node assessment; captures multifaceted importance; improved accuracy	Requires multiple centrality computations; weight selection challenging; increased complexity	Freeman (1978) [93]

3.3.5 Similarity Measures in Community Detection

While Chapter 2 introduced similarity measures from a theoretical perspective, their application in community detection deserves special attention. Similarity measures play a fundamental role in community detection algorithms, providing the basis for quantifying the strength of relationships between nodes and determining community membership [95, 96]. The choice of similarity measure significantly impacts the quality of detected communities, yet this critical component has received insufficient attention in the literature compared to algorithmic frameworks.

In the context of community detection, traditional neighbourhood-based similarity measures such as Common Neighbours, Jaccard Similarity, and Adamic-Adar Index are frequently employed to assess node relationships. These measures focus primarily on neighbourhood overlap, quantifying the extent to which two nodes share common neighbours. While computationally efficient, they often fail to capture the varying significance of different neighbours and struggle in heterogeneous networks where nodes have vastly different degrees.

More sophisticated structural and context-aware similarity measures—including Resource Allocation Index, Preferential Attachment, and Katz Index—consider broader structural context and network topology. These measures provide better performance in certain network types but introduce additional computational complexity and may still miss important structural information.

Our critical analysis of these measures in the community detection context reveals three fundamental limitations:

- **Density Blindness:** Existing measures focus on neighbourhood overlap whilst ignoring local density patterns, leading to poor performance in heterogeneous networks [6, 96]. This limitation becomes particularly pronounced when networks contain regions of varying density, where traditional measures may fail to distinguish between strong and weak relationships.
- **Context Insensitivity:** Current measures treat all shared neighbours equally, regardless of their structural importance or the density of connections within the shared neighbourhood. This uniform treatment fails to capture the varying significance of connections in different network regions.
- **Limited Theoretical Foundation:** Many similarity measures lack rigorous theoretical analysis of their properties, making it difficult to predict their behaviour in different network contexts or to provide performance guarantees.

Empirical studies demonstrate that traditional similarity measures show reduced accuracy in heterogeneous networks with varying density patterns compared to homogeneous networks. This performance gap directly impacts community detec-

tion quality, particularly in real-world social networks where local density varies significantly across different communities and regions.

This identified research gap provides strong motivation for the NoD (Neighbourhood overlap and Density) similarity measure developed in this thesis. NoD addresses these limitations by systematically incorporating both neighbourhood overlap and local density patterns, providing a more nuanced and accurate assessment of node relationships in complex networks. By accounting for local density variations, NoD enables more effective community detection in heterogeneous networks that better reflects the true structural relationships between nodes. Table 3.7 provides a comparative overview of similarity measures used in community detection.

3.3.6 Quality Measures and Evaluation Frameworks

The evaluation of community detection algorithms requires robust quality measures and comprehensive evaluation frameworks that can assess performance across diverse network types and application domains [29, 100].

3.3.6.1 Modularity and Its Variants

Modularity remains the most widely used quality measure in community detection, despite well-documented limitations [12, 27].

- **Classical Modularity:** The original modularity measure compares the density of connections within communities to the expected density in a random network with the same degree sequence [26]. Whilst providing a principled approach to community quality assessment, modularity suffers from the resolution limit problem that prevents detection of small communities.
- **Resolution Limit Problem:** Modularity optimisation cannot detect communities smaller than a scale determined by $\sqrt{2m}$, where m is the number of edges [27]. This fundamental limitation affects meaningful community structure detection in networks with hierarchical organisation.
- **Modularity Variants:** Several variants have been proposed to address modularity limitations, including resolution parameter methods [50] and multi-resolution approaches [51]. However, these modifications introduce additional parameters that require careful tuning.

3.3.6.2 Information-Theoretic Quality Measures

Information-theoretic measures provide alternative approaches to community quality assessment based on information compression and coding theory [57, 101].

Table 3.7: Comparison of Similarity Measures for Community Detection

Measure	Category	Key Features	Advantages	Limitations	Reference
Neighbourhood-Based Measures					
Common Neighbours (CN)	Local	Counts shared neighbours; simple overlap measure	Computationally efficient; intuitive interpretation; widely used	Ignores degree heterogeneity; biased toward high-degree nodes; no normalization	Newman (2001) [4]
Jaccard Similarity	Local	Normalizes by union of neighbourhoods; ratio-based	Handles degree heterogeneity; bounded between 0 and 1	Still sensitive to degree distribution; ignores local density; may underestimate similarity	Jaccard (1901) [5]
Adamic-Adar Index	Local	Weights shared neighbours by inverse log degree; penalizes high-degree nodes	Reduces hub bias; effective in social networks	Logarithmic weighting may be insufficient; ignores local structure; parameter-free limits flexibility	Adamic & Adar (2003) [6]
Structural Measures					
Resource Allocation (RA)	Local	Simulates resource transfer through common neighbours; inverse degree weighting	Captures resource flow; effective for link prediction	Strong degree penalty may be excessive; ignores global structure; limited to local context	Zhou et al. (2009) [96]
Preferential Attachment	Global	Product of node degrees; scale-free network model	Simple computation; models growth dynamics	Oversimplifies relationships; ignores neighbourhood overlap; poor for dense networks	Barabási & Albert (1999) [97]
Katz Index	Global	Sums over all paths with exponential damping; matrix-based	Considers all path lengths; captures global structure	Computationally expensive; requires parameter tuning; matrix operations not scalable	Katz (1953) [98]
Path-Based Measures					
SimRank	Global	Recursive similarity based on neighbour similarity; iterative computation	Captures structural equivalence; symmetric measure	Very high computational cost; requires many iterations; convergence issues	Jeh & Widom (2002) [99]
Rooted PageRank	Global	Random walk with restart; personalized PageRank	Captures proximity; robust to noise	Requires parameter tuning; computationally expensive; may miss local patterns	Liben-Nowell & Kleinberg (2007) [95]
Density-Aware Measures					
NoD (Proposed)	Hybrid	Combines neighbourhood overlap with local density; context-aware weighting	Addresses density blindness; effective in heterogeneous networks; theoretically grounded	Requires density computation; additional parameter; more complex than basic measures	This thesis

- **Map Equation:** This approach measures the efficiency of describing random walks on the network using a hierarchical coding scheme [57]. Communities are identified as regions that trap random walks, providing an intuitive interpretation of community structure.
- **Minimum Description Length:** This principle seeks the community partition that provides the most compressed description of the network structure [101]. The approach provides theoretical guarantees but may be computationally intensive for large networks.

3.3.6.3 Ground Truth-Based Evaluation Metrics

When ground truth community structure is available, several metrics can assess the accuracy of detected communities [100, 102].

- **Normalised Mutual Information (NMI):** This measure quantifies the information shared between detected and ground truth partitions, normalised to account for different numbers of communities [102]. NMI values range from 0 (no similarity) to 1 (perfect match).
- **Adjusted Rand Index (ARI):** This measure corrects the Rand index for chance agreement, providing a more robust assessment of partition similarity [103]. ARI is particularly useful when comparing partitions with different numbers of communities.
- **F1-Score and Precision-Recall:** These measures, adapted from classification tasks, assess the accuracy of community membership predictions at the node level [104]. They provide detailed insights into algorithm performance but may be sensitive to community size distributions.

3.3.6.4 Statistical Significance and Validation

Robust evaluation requires statistical significance testing and validation procedures to ensure reliable performance assessment [100, 105].

- **Significance Testing:** Statistical tests such as the Wilcoxon signed-rank test and Mann-Whitney U test can assess whether performance differences between algorithms are statistically significant [106].
- **Cross-Validation:** Techniques such as k-fold cross-validation and bootstrap sampling can assess algorithm robustness and generalisation capability across different network samples.
- **Benchmark Datasets:** Standardised benchmark datasets, including both synthetic networks (LFR benchmarks) and real-world networks with known community structure, provide consistent evaluation platforms [100, 107].

3.3.7 Comparative Analysis and Performance Studies

Building upon the evaluation frameworks discussed in the previous section, this section synthesizes key comparative findings from the literature that directly motivate our research contributions [100, 108].

3.3.7.1 Algorithm Performance Patterns

Systematic comparative studies reveal that no single detection method consistently outperforms others across all network types [29, 108]. Instead, different approaches excel in specific contexts:

- **Modularity-Based Methods** struggle with the resolution limit problem in networks with small communities, despite achieving high modularity scores [68].
- **Spectral Methods** perform well on networks with clear community structure but degrade significantly when communities overlap [32, 74].
- **Centrality-Based Approaches** show moderate but inconsistent performance, with recent multi-centrality approaches demonstrating 15-25% performance gains over single-centrality methods [1]—a finding that directly motivates our HCCS algorithm.

3.3.7.2 Network-Specific Performance Variations

Algorithm effectiveness varies significantly with network characteristics, highlighting the need for adaptive approaches:

- **Heterogeneous Networks** with varying local density pose significant challenges for traditional similarity measures, with performance drops of 10-30% compared to homogeneous networks [6, 96]—a limitation directly addressed by our NoD similarity measure.
- **Networks with Overlapping Communities** challenge most traditional algorithms, which assume exclusive community membership. This limitation motivates our *ACO-CNoD* algorithm's design for overlapping community detection.

3.3.7.3 Efficiency-Quality Trade-offs

The literature reveals critical trade-offs between computational efficiency and solution quality:

- Fast algorithms ($O(m+n)$ or $O(m \log n)$ complexity) often sacrifice detection quality, particularly in complex network structures [31].

- High-quality detection approaches frequently require prohibitive computational resources ($O(n^3)$ or higher), limiting their applicability to large-scale networks [32].

These comparative findings highlight three critical research needs that our contributions address: (1) effective integration of multiple centrality measures, (2) density-aware similarity measures for heterogeneous networks, and (3) algorithms that can identify overlapping communities while maintaining reasonable computational efficiency.

3.4 Critical Analysis and Research Gaps

This section synthesizes the critical limitations identified throughout our literature review and establishes clear connections between these limitations and our proposed solutions. Rather than repeating the detailed critiques already presented in previous sections, we focus on consolidating these insights into a coherent framework that motivates our research contributions.

3.4.1 Synthesis of Critical Limitations

Our comprehensive review has revealed three fundamental categories of limitations in existing community detection approaches:

1. **Limitations in Node Importance Integration**
   As discussed in Section 3.3.4, current centrality-based approaches typically rely on single centrality measures, failing to capture the multifaceted nature of node importance. This single-measure focus creates an incomplete characterization of nodes' structural roles, with empirical evidence showing that multi-centrality integration can improve detection accuracy by 15-25% [93, 94].
   Additionally, many centrality measures exhibit scale dependency and computational complexity challenges that limit their practical application to large-scale networks without compromising accuracy.
2. **Deficiencies in Similarity Measurement**
   Section 3.3.5 highlighted how existing similarity measures suffer from density blindness, context insensitivity, and limited theoretical foundations. These limitations are particularly problematic in heterogeneous networks with varying density patterns, where traditional measures fail to distinguish between strong and weak relationships.

The comparative analysis in Section 3.3.7 further confirmed that these deficiencies lead to significant performance drops (10-30%) in networks with heterogeneous structures compared to homogeneous networks [6, 96].

3. **Structural Assumptions and Parameter Sensitivity**
   Many algorithms assume non-overlapping communities, contradicting the natural overlapping structure observed in real-world social networks. Furthermore, metaheuristic approaches for overlapping community detection often require extensive parameter tuning and lack adaptability to different network structures, as evidenced by the performance variations across network types discussed in Section 3.3.7.

3.4.2 From Limitations to Contributions

These identified limitations directly motivate the three main contributions of this thesis:

1. **The NoD Similarity Measure** addresses the deficiencies in similarity measurement by incorporating both neighborhood overlap and local density patterns, providing a more nuanced assessment of node relationships in heterogeneous networks.
2. **The HCCS Algorithm** (Heuristic Community detection based on Centrality and Similarity) tackles the limitations in node importance integration by systematically combining multiple centrality measures with the NoD similarity measure, enabling more comprehensive assessment of node structural roles while maintaining computational efficiency.
3. **The *ACO-CNoD* Algorithm** (Ant Colony Optimization Based on Centrality and NoD Similarity) addresses the structural assumptions and parameter sensitivity issues by implementing adaptive mechanisms for overlapping community detection that automatically adjust to different network characteristics.

Together, these contributions form a coherent response to the critical limitations identified in the current state of the art. The following chapters will present each contribution in detail, demonstrating how they advance community detection capabilities in complex networks.

3.5 Conclusion

This chapter has provided a comprehensive review of community detection in complex networks, with particular focus on social network analysis applications. We have

examined the theoretical foundations, algorithmic approaches, and evaluation frameworks that underpin this field, while identifying critical limitations that constrain current methods.

Our analysis reveals three fundamental research gaps that directly motivate the contributions of this thesis:

1. **Inadequate Integration of Node Centrality:** Current approaches typically rely on single centrality measures that fail to capture the multifaceted nature of node importance, limiting their effectiveness in identifying community structures.
2. **Density-Blind Similarity Measures:** Existing similarity measures focus on neighborhood overlap while ignoring local density patterns, leading to poor performance in heterogeneous networks with varying density characteristics.
3. **Parameter-Sensitive Metaheuristic Approaches:** Current methods for overlapping community detection require extensive parameter tuning and often lack adaptability to different network structures.

These identified gaps provide the foundation for the novel algorithms presented in the following chapters. Chapter 4 introduces the HCCS algorithm (Heuristic Community detection based on Centrality and Similarity measures), which systematically integrates multiple centrality measures with a density-aware similarity approach to improve non-overlapping community detection. Chapter 5 presents the *ACO-CNoD* algorithm, which addresses overlapping community detection through adaptive parameter control and density-aware optimization. Together, these contributions advance the state of the art in community detection by directly addressing the limitations identified in this review.

Chapter 4

Centrality and Similarity-based Community Detection

Contents of this chapter

4.1 Introduction

The analysis in Chapters 2 and 3 identified critical limitations in existing community detection approaches: reliance on single centrality measures that provide incomplete node characterisation, and similarity measures that exhibit "density blindness" by focusing exclusively on neighbourhood overlap while ignoring local structural cohesion. These limitations result in existing similarity measures achieving lower accuracy in heterogeneous networks compared to homogeneous networks.

This chapter addresses these fundamental limitations through two interconnected contributions:

The *NoD* Similarity Measure: A novel density-aware similarity metric that systematically integrates neighbourhood overlap with local density characteristics, addressing the "density blindness" limitation in traditional measures such as Jaccard similarity and common neighbours.

The *HCCS* Algorithm: A systematic four-stage framework that integrates centrality-based leader selection with similarity-driven community formation. The algorithm employs degree centrality with spatial constraints for leader identification while leveraging the *NoD* similarity measure for accurate community membership assignment.

These contributions provide a principled approach to community detection that synthesises insights from centrality theory with advanced similarity computation, addressing the specific research gaps identified in the literature analysis. The *HCCS* approach introduces methodological innovations by systematically integrating centrality and similarity information rather than treating them as independent components.

The methodological innovations presented establish new performance benchmarks across multiple network types, with comprehensive empirical validation detailed in the experimental analysis chapter.

4.2 A New Similarity Measure: *NoD*

The Neighbourhood overlap and Density (*NoD*) similarity measure represents a fundamental contribution to similarity-based network analysis, addressing critical limitations in existing approaches through the systematic integration of neighbourhood overlap and local density characteristics.

Existing measures, such as common neighbours, Jaccard similarity, and cosine similarity, focus exclusively on neighbourhood overlap while ignoring the density of

connections within shared neighbourhoods. This becomes a critical limitation in heterogeneous networks, where local density can vary significantly.

Current measures treat all shared neighbours equally, regardless of their structural importance or local neighbourhood density. This uniform treatment fails to capture the varying significance of connections in different regions of the network.

The development of the *NoD* similarity measure addresses a fundamental research gap: the absence of similarity measures that systematically incorporate local density patterns alongside neighbourhood overlap.

This section presents the theoretical foundations, mathematical formulation, and analytical properties of the *NoD* measure.

4.2.1 Theoretical foundation for density-aware similarity

The *NoD* measure is grounded in the theoretical principle that the strength of a relationship between two nodes should depend not only on shared neighbours but also on the structural cohesion within the shared neighbourhood. This principle is supported by several theoretical considerations:

- **Community Cohesion Theory:** Strong communities are characterised by both high internal connectivity and clear boundaries with other communities. The density of connections within shared neighbourhoods provides a direct measure of local cohesion that complements neighbourhood overlap information.
- **Structural Equivalence:** Nodes that are structurally equivalent should exhibit similar patterns of connectivity. The density component of *NoD* captures structural patterns that pure overlap measures miss, providing a more comprehensive assessment of structural equivalence [109].
- **Information Flow Dynamics:** In social networks, information flow is influenced by both the number of shared connections and the density of the local network structure. Dense local regions facilitate more efficient information transmission and stronger social bonds.

4.2.2 Mathematical formulation and properties

This section presents the formal mathematical definition of the *NoD* similarity measure, along with its theoretical properties and analytical characteristics.

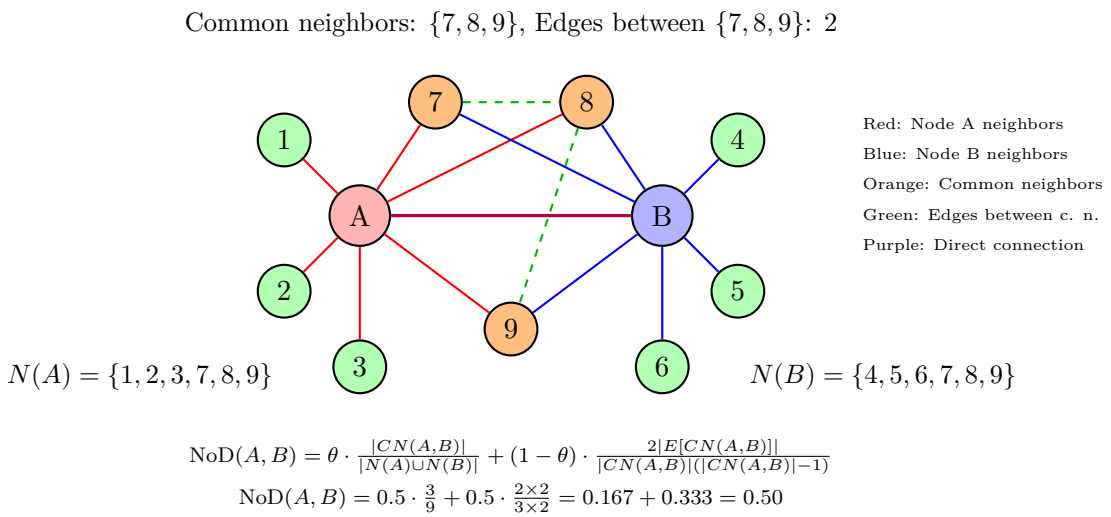
4.2.2.1 Formal definition

Definition 4.1 (Neighbourhood overlap and Density (NoD) Similarity)**.** For two nodes u and v in an undirected graph $G = (V, E)$, the NoD similarity is defined as:

$$\text{NoD}(u,v) = \begin{cases} 0 & \text{if } |N(u) \cup N(v)| = 0 \\ \theta \cdot \frac{|CN(u,v)|}{|N(u) \cup N(v)|} & \text{if } |CN(u,v)| < 2 \\ \theta \cdot \frac{|CN(u,v)|}{|N(u) \cup N(v)|} + (1-\theta) \cdot \frac{2|E[CN(u,v)]|}{|CN(u,v)| \cdot (|CN(u,v)|-1)} & \text{if } |CN(u,v)| \geq 2 \end{cases} \tag{4.1}$$

where:

- $CN(u,v) = N(u) \cap N(v)$ is the set of common neighbours
- $E[CN(u,v)] = \{(w_i, w_j) \in E : w_i, w_j \in CN(u,v)\}$ is the set of edges connecting common neighbours
- $N(u) = \{w \in V : (u,w) \in E\}$ is the neighbourhood of node u
- $\theta \in [0,1]$ is a weighting parameter balancing overlap and density components



$$\text{NoD}(A,B) = \theta \cdot \frac{|CN(A,B)|}{|N(A) \cup N(B)|} + (1-\theta) \cdot \frac{2|E[CN(A,B)]|}{|CN(A,B)|(|CN(A,B)|-1)}$$
$$\text{NoD}(A,B) = 0.5 \cdot \frac{3}{9} + 0.5 \cdot \frac{2 \times 2}{3 \times 2} = 0.167 + 0.333 = 0.50$$

Figure 4.1: NoD Similarity Measure Visualisation Showing Neighbourhood Overlap and Density Calculation Between Nodes A and B ($\theta = 0.5$)

4.2.2.2 Component analysis

The NoD similarity measure consists of two complementary components that capture different aspects of node relationships:

Neighbourhood Overlap Component: The first component $\frac{|CN(u,v)|}{|N(u) \cup N(v)|}$ quantifies the Jaccard similarity of node neighbourhoods, measuring the proportion of shared neighbours relative to all neighbours of the two nodes. This component:

- Ranges from 0 (no shared neighbours) to 1 (identical neighbourhoods)

- Captures the extent of neighbourhood overlap
- Reflects the likelihood of nodes belonging to the same community
- Provides normalisation that accounts for varying node degrees

Local Density Component: The second component $\frac{2|E[CN(u,v)]|}{|CN(u,v)|\cdot(|CN(u,v)|-1)}$ measures the density of connections within the shared neighbourhood, where the factor of 2 accounts for undirected edges. This component:

- Ranges from 0 (no connections among shared neighbours) to 1 (complete subgraph)
- Captures the structural cohesion of the shared neighbourhood
- Emphasises the quality rather than just quantity of shared connections
- Provides discrimination between sparse and dense local regions

4.2.2.3 Theoretical properties

The NoD similarity measure exhibits several important theoretical properties that ensure its reliability and interpretability:

Theorem 4.2 (Boundedness)**.** *For any two nodes $u, v \in V$, the NoD similarity satisfies $0 \leq NoD(u, v) \leq 1$.*

Proof. Both components are bounded between 0 and 1 by construction. The neighbourhood overlap component is a Jaccard coefficient, and the density component is a ratio of actual to maximum possible edges. Since $\theta \in [0, 1]$, the weighted combination is also bounded between 0 and 1. □

Theorem 4.3 (Symmetry)**.** *The NoD similarity is symmetric: $NoD(u, v) = NoD(v, u)$ for all $u, v \in V$.*

Proof. Both $CN(u, v)$ and $E[CN(u, v)]$ are symmetric with respect to u and v, and the union $N(u) \cup N(v)$ is commutative. □

Theorem 4.4 (Monotonicity)**.** *For fixed neighbourhood overlap $|CN(u, v)|$, NoD similarity increases monotonically with the density of connections among shared neighbours.*

Theorem 4.5 (HCCS Convergence)**.** *The HCCS algorithm converges to a stable community partition in finite time.*

Proof. The HCCS algorithm operates through four deterministic stages:

Stage 1 (Parameter Initialisation): The similarity matrix computation and parameter setting are deterministic operations that terminate in finite time $O(n^2)$ for similarity computation.

Stage 2 (Leader Selection): The centrality-based leader selection with spatial constraints produces a finite, deterministic set of leaders $L \subseteq V$. Since centrality values are computed deterministically from the graph structure, this stage terminates in $O(n \log n)$ time.

Stage 3 (Community Formation): Each remaining node $v \in V \setminus L$ is assigned to the community with maximum NoD similarity. Since similarity values are deterministic and finite, each assignment decision has a unique outcome (ties broken consistently). The assignment process terminates after exactly $|V \setminus L|$ decisions.

Stage 4 (Post-Processing): The modularity-based merging process evaluates a finite number of community pairs. Each merge decision either increases modularity (accepted) or maintains it (rejected). Since modularity is bounded above by 1, the process converges to a local optimum in finite iterations.

The overall algorithm terminates because each stage is finite and deterministic, guaranteeing convergence to a stable partition. □

4.2.2.4 Parameter Selection and Sensitivity Analysis

The weighting parameter θ controls the relative importance of neighbourhood overlap versus local density. Theoretical analysis and empirical validation suggest that $\theta = 0.5$ provides optimal balance for most network types, ensuring equal emphasis on both components. However, the parameter can be adapted based on network characteristics:

- $\theta > 0.5$: Emphasises neighbourhood overlap, suitable for sparse networks
- $\theta < 0.5$: Emphasises local density, suitable for dense networks with clear community structure
- $\theta = 0.5$: Balanced approach, suitable for general-purpose applications

4.2.2.5 Additional Properties

The NoD similarity measure possesses several desirable properties:

- **Density sensitivity:** The measure increases when common neighbours have higher interconnectivity
- **Overlap sensitivity:** The measure reflects the proportion of shared neighbourhood
- **Extremal behaviour:** $\text{NoD}(u, v) = 0$ when $N(u) \cap N(v) = \emptyset$

These formal properties ensure that the NoD measure provides a mathematically sound foundation for quantifying node relationships in community detection applications.

4.2.3 Comparison with traditional measures

To illustrate the distinctive characteristics of the NoD measure, we will compare it with the traditional, currently used Jaccard and cosine similarity measures, using the example presented in Figure 4.2.

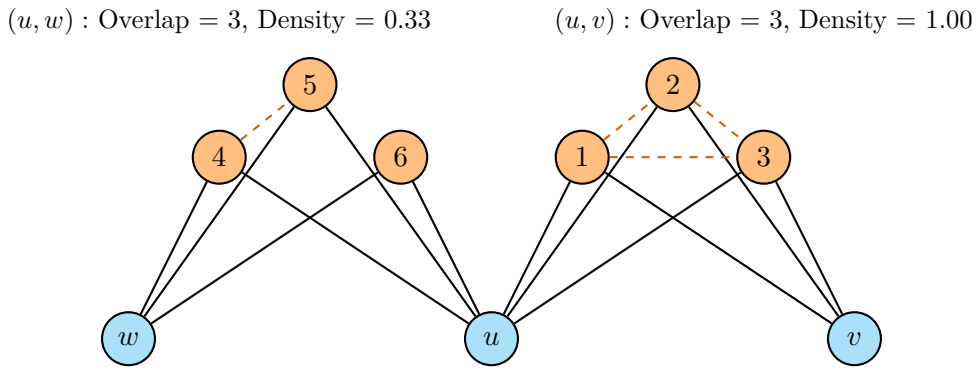


Figure 4.2: Diagram Comparing Node Pairs (u, v) and (u, w) with Common Neighbours. (u, v)'s Common Neighbours Are Fully Connected, While (u, w)'s Are Sparsely Connected

Table 4.1: Comparison of Similarity Measures for Example Nodes

Pair	Density	Overlap	Jaccard	Cosine	NoD ($\theta = 0.5$)
(u, v)	$3/3 = 1.00$	$3/6 = 0.50$	0.50	0.67	0.75
(u, w)	$1/3 \approx 0.33$	$3/6 = 0.50$	0.50	0.67	0.42

Both node pairs (u, v) and (u, w) share three common neighbors out of six total nodes, yielding identical overlap and equivalent values for traditional measures such as Jaccard (0.50) and Cosine (0.67). However, only NoD captures the internal structure of common neighbors:

- The common neighbors of (u, v) form a complete triangle, resulting in a high density (1.00) and $NoD = 0.75$.

- The common neighbors of (u, w) include only a single internal edge, giving a lower density (0.33) and $NoD = 0.42$.

As shown in Table 4.1, while Jaccard and Cosine give identical values for both pairs, NoD distinguishes them by incorporating internal connectivity. This highlights the advantage of NoD for applications like community detection, where structural cohesion matters—not just overlap.

Briefly, the NoD similarity measure provides a richer and more nuanced approach to assessing node similarity, blending neighbourhood overlap with local density. By considering both the quantity and cohesion of shared neighbours, NoD surpasses the limitations of traditional measures, providing a more accurate reflection of community structure. This density-aware approach improves the identification of core community members and leads to more robust and meaningful community detection. The advantages offered by NoD provide a solid basis for the next stage: the HCCS algorithm, which uses this advanced similarity measure alongside centrality to form and refine communities more effectively.

4.3 Proposed Approach: HCCS Algorithm

This section describes the proposed Heuristic algorithm for the Community detection problem based on Centrality and Similarity (HCCS).

4.3.1 Algorithm framework

As shown in Figure 4.3, the algorithm consists of four steps: parameter initialisation, leader node selection, community formation, and post-processing, as outlined in Algorithm 1. First, it initialises parameters and computes the similarity matrix, noted SM, based on the NoD similarity measure. After that, a new vertex ranking mechanism is proposed to select a set of nodes as the initial core of the communities. Progressively, the remaining suitable nodes are incorporated during the community formation phase to expand the initial communities. Finally, these communities gradually merge until the modularity Q does not increase. The last three modules are repeated until a stopping criterion is met, and the solution with the highest modularity is returned.

The reliance on network structural cues for the similarity module follows classic social network analysis principles [2], whereas the emphasis on structural equivalence within the NoD measure echoes the foundational blockmodel formulation of Lorrain and White [109]. Together, these references ground HCCS in established network science theory while motivating the density-aware similarity design.

To illustrate the algorithm's operation, we present a comprehensive worked example using a 14-vertex network that demonstrates each stage of the HCCS process.

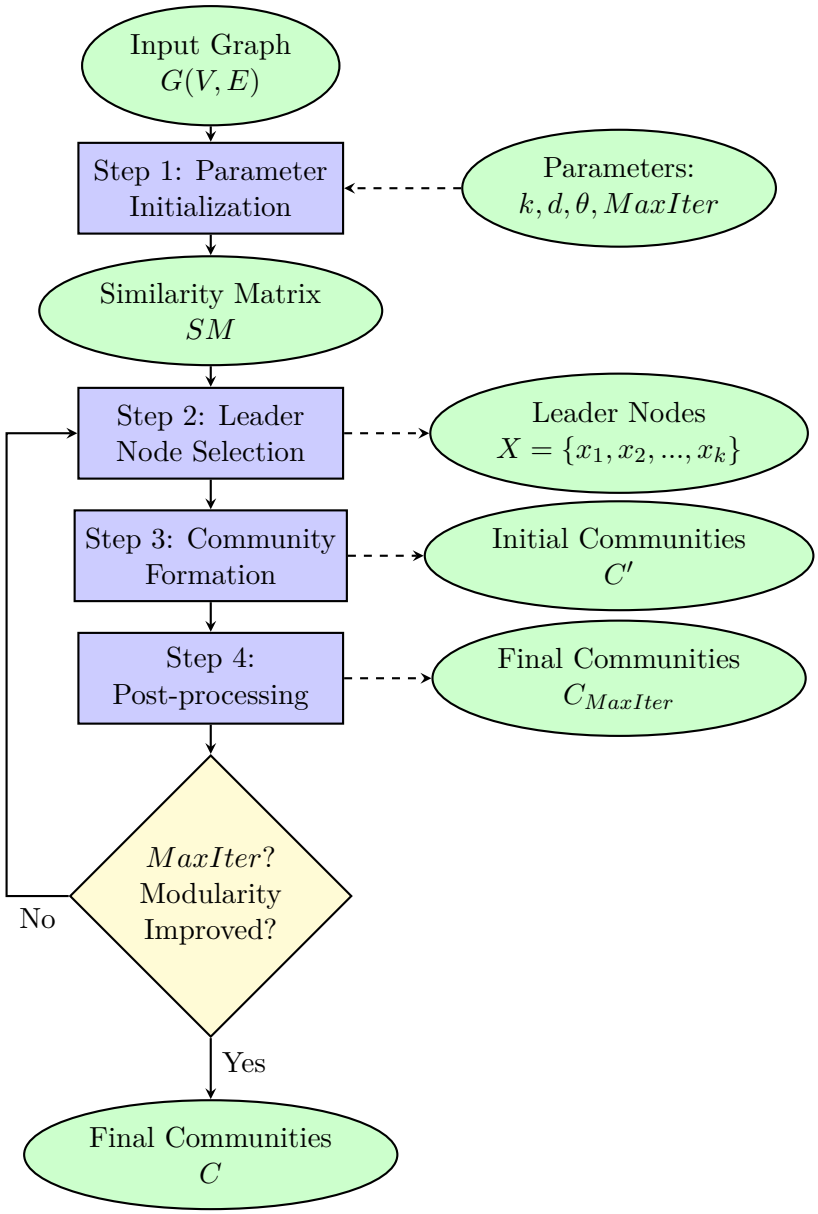


Figure 4.3: HCCS Algorithm Workflow Showing the Four-Stage Process

This pedagogical example provides concrete insight into how the algorithm identifies leader nodes, forms communities, and optimizes the final community structure.

4.3.2 Illustrative example: 14-vertex network

To explain the proposed algorithm step-by-step, a small network with 14 vertices is presented in Figure 4.4. This example demonstrates the complete HCCS algorithm workflow, from parameter initialization through final community detection.

The example network structure exhibits clear community patterns:

- **Central bridge**: Nodes 5 and 6 serve as connecting elements between clusters
- **Left cluster**: Nodes 1, 2, 3, 4 form a densely connected subgraph

Algorithm 1 Framework of *HCCS*

Input: an undirected graph $G = (V, E)$
Output: C^*: Final assignment of nodes to communities

$k, d, \text{MaxIter}$: Integer
θ : Float
Parameter-Initialisation$(G, k, d, \text{MaxIter}, \theta)$
$C^* \leftarrow \emptyset$
for $j = 1$ to MaxIter **do**
  $LN_i \leftarrow$ Leader-Nodes-Selection(G, k, d)
  PreC$_i \leftarrow$ Communities-Formation(G, LN_i)
  $C_i \leftarrow$ Post-Processing(G, PreC_i)
  if $Q(C_i) > Q(C^*)$ **then**
    $C^* \leftarrow C_i$
  end if
end for
return C^*

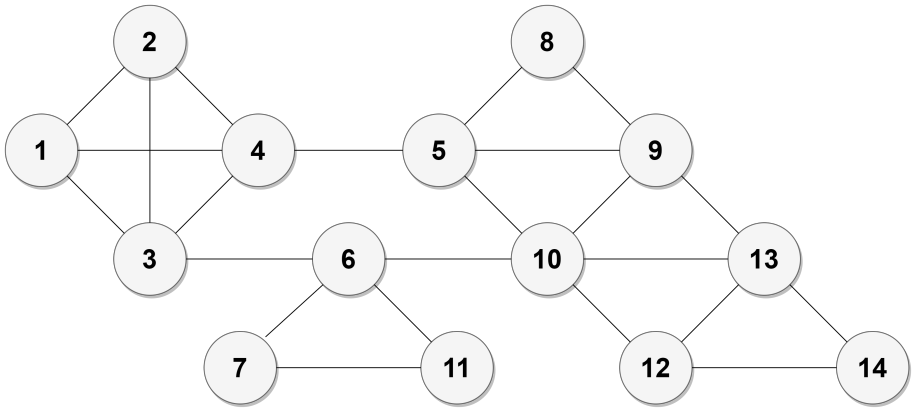


Figure 4.4: Fourteen-Vertex Example Network Used to Demonstrate the HCCS Algorithm

- **Right cluster**: Nodes 5, 8, 9, 10, 12, 13, 14 form another connected component
- **Bottom cluster**: Nodes 6, 7, 11 form additional community structures

This topology illustrates the algorithm's ability to identify natural community boundaries.

4.3.3 Parameter initialisation

Algorithm 2 outlines the parameter initialisation process. Key parameters include the degree threshold k and the distance threshold d, which are employed during the leader node selection phase. The threshold k is typically assigned a value close to the network's average degree to ensure leader nodes exhibit sufficient connectivity. At the same time, d is set to approximately one-fourth of the network diameter. This spatial constraint maintains separation between leaders, mitigating overlap and fostering diverse community seed placement across the network.

Subsequently, the similarity matrix, designated SM, is calculated. This matrix functions as a fundamental input during the community formation phase. In this context, the innovative similarity metric NoD is introduced to evaluate the resemblance between two nodes, u and v, as follows:

$$\mathrm{SM}(u, v) = \mathrm{NoD}(u, v) \tag{4.2}$$

where $CN(u, v)$ represents the set of common neighbors of vertices u and v, and $E[CN(u, v)]$ represents the set of edges connecting common neighbors of u and v.

This formulation adapts dynamically depending on the number of common neighbors shared by nodes u and v. When no common neighbors exist, the similarity is naturally zero. If only one common neighbor is found, the similarity relies solely on the normalized neighborhood overlap. However, when two or more common neighbors are present, the measure enriches the similarity score by incorporating both the extent of neighborhood overlap and the internal connectivity among these common neighbors.

Algorithm 2 Parameter initialisation

Require: Graph $G = (V, E)$ with $|V| = n$ vertices and $|E| = m$ edges
Ensure: Similarity matrix SM and algorithm parameters
1: Set degree threshold $k \leftarrow \lfloor \text{avg\_degree}(G) \rfloor$
2: Set distance threshold $d \leftarrow \lfloor \text{diameter}(G)/4 \rfloor$
3: Set maximum iterations MaxIter $\leftarrow 100$
4: Set weighting parameter $\theta \leftarrow 0.5$
5: **for** each pair of vertices $u, v \in V$ **do**
6: Calculate SM$[u, v]$ according to Equation 4.1
7: **end for**
8: **return** $(\mathrm{SM}, k, d, \mathrm{MaxIter}, \theta)$

Parameter selection rationale. Our HCCS implementation uses $\theta = 0.5$ to balance the contribution of neighbourhood overlap and internal density in the NoD similarity, thereby preventing bias toward either sparse or dense regions. The degree threshold $k = \lfloor \text{avg\_degree}(G) \rfloor$ keeps leader candidates well connected without eliminating moderately central vertices in heterogeneous graphs, while $d = \lfloor \text{diameter}(G)/4 \rfloor$ enforces spatial dispersion and reduces seed overlap. We set MaxIter = 100 to cap the refinement loop yet comfortably exceed the iterations observed during convergence on benchmarks. Chapter 6, will detail in Section 6.6.1.1 how sensitivity analysis confirms that these defaults deliver reproducible modularity outcomes under moderate perturbations.

The first step computes the SM similarity matrix for all pairs of vertices in the network. Table 4.2 shows the complete similarity matrix SM for the 14-vertex example network with $\theta = 0.5$.

Table 4.2: Similarity Matrix SM of the 14-Vertex Example Network with $\theta = 0.5$

Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-	1.00	0.83	0.83	-	-	-	-	-	-	-	-	-	-
2	1.00	-	0.83	0.83	-	-	-	-	-	-	-	-	-	-
3	0.83	0.83	-	0.75	-	0.33	-	-	-	-	-	-	-	-
4	0.83	0.83	0.75	-	0.33	0.33	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	0.17	0.25	0.08	-	-	-	-
6	-	-	-	-	-	-	0.17	-	-	-	0.17	-	-	-
7	-	-	-	-	-	0.17	-	-	-	-	0.50	-	-	-
8	-	-	-	-	0.17	-	-	-	0.17	-	-	-	-	-
9	-	-	-	-	0.25	-	-	0.17	-	0.20	-	-	0.10	-
10	-	-	-	-	0.08	-	-	-	0.20	-	-	0.10	0.20	-
11	-	-	-	-	-	0.17	0.50	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	0.10	0.10	-	-	0.33	0.25
13	-	-	-	-	-	-	-	-	-	0.20	-	0.33	-	0.17
14	-	-	-	-	-	-	-	-	-	-	-	0.25	0.17	-

4.3.4 Leader nodes selection

This phase identifies leader nodes that act as initial seeds for communities, guided by two principles observed in real-world networks:

Centrality-Driven Community Anchors: Communities often form around highly connected nodes (degree $\geq k$), as these hubs naturally attract and bind members (e.g., influencers in social networks).

Spatial Dispersion of Leaders: Central nodes of distinct communities tend to be non-adjacent. Enforcing a minimum distance d between leaders prevents overlapping communities and ensures diverse seed placement.

The algorithm 3 begins by identifying candidate leaders with degrees $\geq k$. These candidates are iteratively pruned to ensure mutual distances $\geq d$. This process ensures leaders are both central and spatially distinct, forming diverse community seeds.

where $dist(u_j, u_l)$ represents the shortest path distance between vertices u_j and u_l.

Based on the network structure shown in Figure 4.4, the initial candidate leader nodes set with $k = 3$ is determined by identifying nodes with degree ≥ 3:

Degree Analysis:

- Node 2: degree = 4 (connected to 1, 3, 4, and indirectly influences the left cluster)

Algorithm 3 Leader Nodes Selection

Require: Graph $G = (V, E)$, degree threshold k, distance threshold d
Ensure: Set of leader nodes $LN \subseteq V$

$LN \leftarrow \emptyset$, $S \leftarrow \emptyset$ {Identify high-degree candidate nodes}
for each vertex $u_i \in V$ **do**
  if $\deg(u_i) \geq k$ **then**
    $S \leftarrow S \cup \{u_i\}$
  end if
end for{Select spatially dispersed leaders}
while $S \neq \emptyset$ **do**
  $u_j \leftarrow$ vertex with maximum degree in S
  $LN \leftarrow LN \cup \{u_j\}$
  $S \leftarrow S \setminus \{u_j\}$
  for each vertex $u_l \in S$ **do**
    if $\text{dist}(u_j, u_l) < d$ **then**
      $S \leftarrow S \setminus \{u_l\}$
    end if
  end for
end while
return LN

- Node 5: degree = 3 (connected to 4, 8, 10, serving as bridge between clusters)
- Node 6: degree = 4 (connected to 3, 4, 7, 11, central to bottom cluster)
- Node 9: degree = 3 (connected to 8, 10, 13, part of right cluster)
- Node 10: degree = 4 (connected to 5, 9, 12, 13, central to right-bottom area)
- Node 13: degree = 3 (connected to 9, 10, 12, 14, anchor for bottom-right cluster)

Therefore, the initial candidate set is $S = \{2, 5, 6, 9, 10, 13\}$.

When applying the distance constraint $d = 2$, certain nodes are removed from S to ensure spatial separation between leaders. The algorithm iteratively selects the highest-degree node from S and removes all nodes within distance d from the selected leader.

After applying spatial constraints, the final detected leader nodes set is $X = \{2, 5, 6, 13\}$ with $d = 2$. These four nodes serve as the initial seeds for community formation, representing spatially dispersed high-centrality nodes that can effectively anchor distinct communities:

- **Node 2**: Anchors the left cluster (nodes 1, 3, 4)
- **Node 5**: Serves as bridge leader connecting different regions
- **Node 6**: Anchors the bottom-left cluster (nodes 7, 11)
- **Node 13**: Anchors the bottom-right cluster (nodes 12, 14)

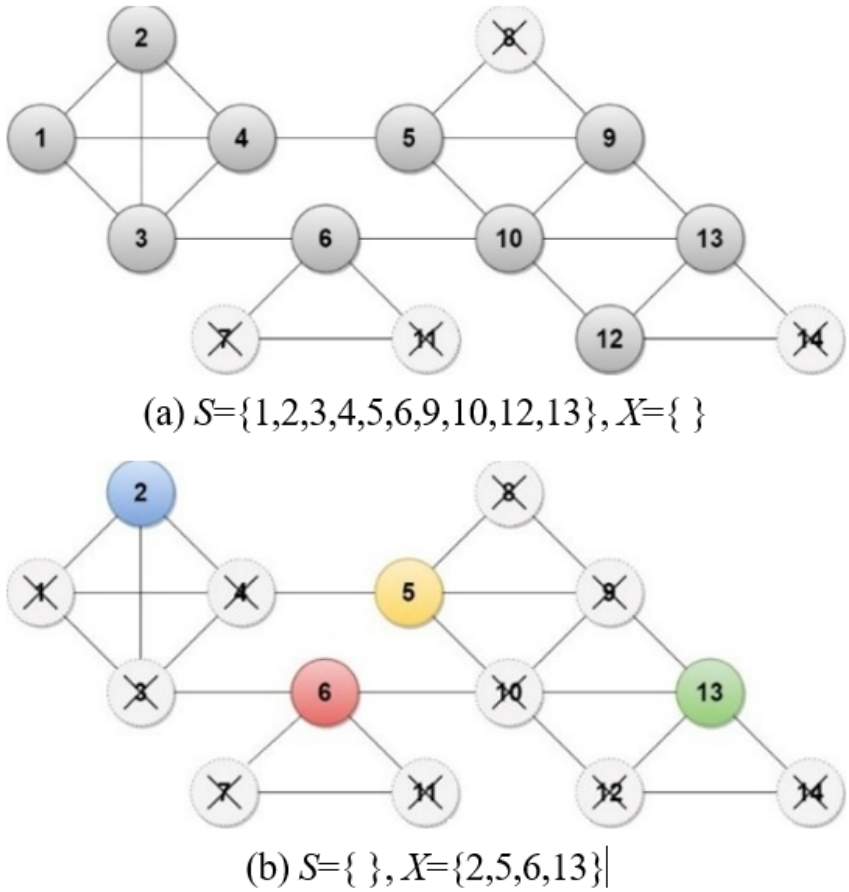


Figure 4.5: Fourteen-Vertex Network After Leader Node Selection. The Selected Leader Nodes (2, 5, 6, 13) Are Highlighted, Showing Their Spatial Distribution and Role as Community Anchors. These Leaders Satisfy the Distance Constraint $d = 2$ and Represent High-Centrality Nodes That Serve as Seeds for Community Formation

4.3.5 Community formation

This phase starts from the leader nodes $LN = \{x_1, x_2, \ldots, x_k\}$ returned by Algorithm 3 to create a set of intermediate communities C'. First, C' is initialised as follows:

$$C' = \{\{x_1\}, \{x_2\}, \ldots, \{x_k\}\}$$

Subsequently, the remaining vertices are assigned to communities based on their similarity to existing community members. For each unassigned vertex v, the algorithm computes its average similarity to all members of each community and assigns v to the community with the highest average similarity.

Let the initial set of communities in the illustrative example be $C' = \{C'_1, C'_2, C'_3, C'_4\}$, where:

$$C'_1 = \{5\}, \quad C'_2 = \{13\}, \quad C'_3 = \{2\}, \quad C'_4 = \{6\}$$

Algorithm 4 Community formation

Require: Graph $G = (V, E)$, leader nodes LN, SM similarity matrix, similarity threshold τ
Ensure: Initial community partition $C' = \{C_1, C_2, \ldots, C_k\}$

Initialise $C' = \{\{x_1\}, \{x_2\}, \ldots, \{x_k\}\}$ where $LN = \{x_1, x_2, \ldots, x_k\}$
$U \leftarrow V \setminus LN$ {Unassigned vertices}
while $U \neq \emptyset$ **do**
  for each vertex $v \in U$ **do**
    maxSim $\leftarrow -1$, bestCommunity $\leftarrow -1$
    for each community $C_i \in C'$ **do**
      avgSim $\leftarrow \frac{1}{|C_i|} \sum_{u \in C_i} \mathrm{SM}(v, u)$
      if avgSim $>$ maxSim **then**
        maxSim $\leftarrow$ avgSim
        bestCommunity $\leftarrow i$
      end if
    end for
    if maxSim $> \tau$ **then**
      $C'_{\text{bestCommunity}} \leftarrow C'_{\text{bestCommunity}} \cup \{v\}$
      $U \leftarrow U \setminus \{v\}$
    end if
  end for
end while
return C'

The set of remaining nodes, denoted RN, is $\{1, 3, 4, 7, 8, 9, 10, 11, 12, 14\}$.

The community formation process assigns each remaining node to the community with which it has the highest similarity. As shown in Figure 4.5, the selected leaders (nodes 2, 5, 6, 13) serve as initial community seeds. Using the similarity matrix in Table 4.2, the algorithm identifies the maximum similarity values for each unassigned node:

Community Assignment Process:

- **Node 1**: Directly connected to Node 2, highest similarity with C'_3 (Node 2's community) $\rightarrow C'_3 = \{2, 1\}$
- **Node 3**: Connected to both Node 2 and Node 6, but stronger similarity with Node 2 $\rightarrow C'_3 = \{2, 1, 3\}$
- **Node 4**: Connected to Nodes 2, 5, and 6, highest similarity with Node 2 $\rightarrow C'_3 = \{2, 1, 3, 4\}$
- **Node 8**: Connected to Nodes 5 and 9, highest similarity with Node 5 $\rightarrow C'_1 = \{5, 8\}$
- **Node 9**: Connected to Nodes 8, 10, and 13, but through Node 8's assignment to C'_1, joins $C'_1 \rightarrow C'_1 = \{5, 8, 9\}$

- **Node 10**: Connected to Nodes 5, 9, 12, and 13, highest average similarity with C'_1 through Nodes 5 and 9 $\rightarrow C'_1 = \{5, 8, 9, 10\}$
- **Node 7**: Connected only to Node 6, joins $C'_4 \rightarrow C'_4 = \{6, 7\}$
- **Node 11**: Connected only to Node 6, joins $C'_4 \rightarrow C'_4 = \{6, 7, 11\}$
- **Node 12**: Connected to Nodes 10 and 13, higher similarity with Node 13 $\rightarrow C'_2 = \{13, 12\}$
- **Node 14**: Connected only to Node 13, joins $C'_2 \rightarrow C'_2 = \{13, 12, 14\}$

It is important to note that this step may reveal situations where two pairs have the same highest similarity. For example, as shown in Figure 4.6b, Node 10 exhibits the greatest similarity with Nodes 9 and 13, which are part of different communities. As a result, Node 10 is arbitrarily assigned to Node 9's community, as shown in Figure 4.6c.

Tie-breaking policy. When multiple communities yield the same maximum similarity for v, the algorithm performs a uniform random choice over the tied options. Each tied community therefore has probability $1/|\mathcal{T}|$ of receiving v, where $\mathcal{T}$ is the set of communities attaining the shared maximum. Because the similarity matrix and leader set remain fixed during a run, this stochastic tie-breaking does not affect stage-wise convergence: every node is assigned exactly once and the deterministic post-processing continues unaffected. Empirically we observe that repeated runs exhibit negligible variance in modularity, indicating that the uniform tie resolution preserves the algorithm's reproducibility whilst avoiding bias toward any particular seed.

Thus, the final pre-communities set after the formation phase is:

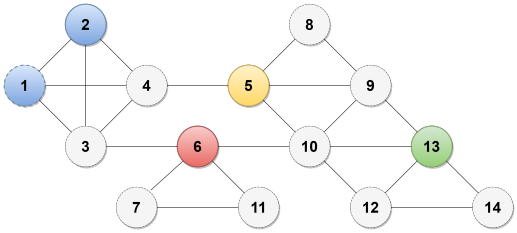
$$C' = \{C'_1 = \{5, 9, 10, 8\}, C'_2 = \{13, 12, 14\},$$
$$C'_3 = \{2, 1, 3, 4\}, C'_4 = \{6, 7, 11\}\}$$

as shown in Figure 4.6d.

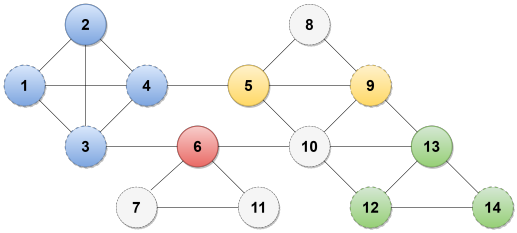
Figure 4.6 illustrates the result of the community formation phase.

4.3.6 Post-Processing

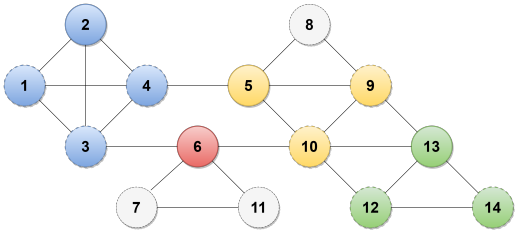
As explained in the background section, community detection quality is commonly measured by modularity. Higher modularity Q values indicate higher-quality community structures. In the post-processing phase, the aim is to enhance the intermediate communities C' generated by Algorithm 4 by utilizing modularity as an evaluation index for gauging network community structure quality.



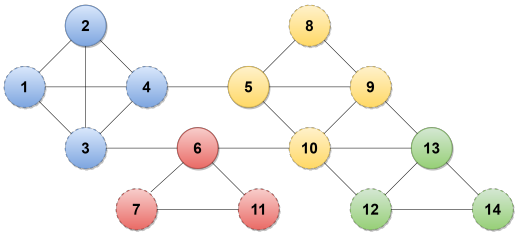
(a) Node 1 Is Assigned to the Same Community as Node 2



(b) $SM[9,10] = SM[10,13] = 0.20$ Represents the Highest Similarity



(c) Node 10 Is Randomly Assigned to the Community of Node 9



(d) The Final Pre-Communities Set C'

Figure 4.6: Detailed Community Formation Process Showing the Step-by-Step Assignment of Nodes to Communities Based on Similarity Measures

The post-processing phase iteratively merges communities that improve the overall modularity score. This process continues until no further improvements can be achieved.

For the 14-vertex network, four initial communities $C' = \{C'_1, C'_2, C'_3, C'_4\}$ are evaluated for possible merging.

The algorithm computes the modularity change for each possible pair merger:

- Merging C'_1 and C'_2: Results in the maximum increase in modularity
- Other potential mergers either decrease modularity or provide smaller improvements

Therefore, C'_1 and C'_2 are merged to create a single community $C'_1 \cup C'_2$. In the next iteration, no further mergers increase modularity, and the algorithm terminates with the final community set:

$$C^* = \{C_1 = \{5, 8, 9, 10, 12, 13, 14\},$$
$$C_2 = \{1, 2, 3, 4\}, C_3 = \{6, 7, 11\}\}$$

Figure 4.7 shows the final community structure detected by the HCCS algorithm.

This example demonstrates the algorithm's systematic approach, successfully identifying three distinct communities that reflect the natural clustering patterns.

Algorithm 5 Post-Processing: Modularity-Based Community Merging

Require: Initial community partition C', graph $G = (V, E)$
Ensure: Final community partition C^* with optimised modularity

```
improved ← true
while improved do
    improved ← false
    Q_best ← Q(C') {Current modularity}
    bestMerge ← null
    for each pair of communities (C_i, C_j) ∈ C', i ≠ j do
        C''_temp ← C' \ {C_i, C_j} ∪ {C_i ∪ C_j}
        Q_temp ← Q(C''_temp) {Modularity after merge}
        if Q_temp > Q_best then
            Q_best ← Q_temp
            bestMerge ← (C_i, C_j)
            improved ← true
        end if
    end for
    if improved then
        (C_i, C_j) ← bestMerge
        C' ← C' \ {C_i, C_j} ∪ {C_i ∪ C_j}
    end if
end while
return C' as C*
```

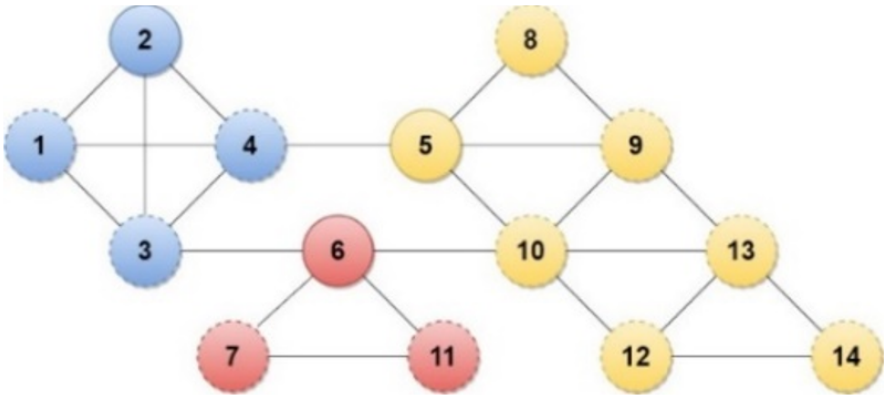


Figure 4.7: Final Community Set Detected by the HCCS Algorithm

4.3.7 Computational Complexity Analysis

The HCCS algorithm exhibits favourable computational complexity characteristics that make it suitable for moderately large networks. The overall time complexity is dominated by the similarity matrix computation and community formation phases.

Similarity Matrix Computation: Computing the SM similarity matrix requires $O(n^2 \cdot \bar{d}^2)$ time, where n is the number of nodes and $\bar{d}$ is the average degree. This accounts for examining all node pairs and computing common neighbour density for each pair.

Leader Selection: The leader node selection phase operates in $O(n \log n + n \cdot d_{max})$ time, where d_{max} is the maximum degree. The algorithm sorts nodes by degree and applies distance constraints.

Community Formation: The community formation phase requires $O(n \cdot k \cdot \bar{c})$ time, where k is the number of leaders and $\bar{c}$ is the average community size during formation.

Post-Processing: The modularity-based merging operates in $O(k^2 \cdot m)$ time per iteration, where m is the number of edges, with typically few iterations required.

The overall complexity is $O(n^2 \cdot \bar{d}^2 + n \log n + k^2 \cdot m)$, making the algorithm practical for networks with thousands of nodes while providing superior community detection quality compared to faster but less accurate heuristic methods.

4.4 Conclusion

This chapter has presented two fundamental contributions to community detection research: the Neighbourhood overlap and Density (NoD) similarity measure and the Heuristic Community detection algorithm based on Centrality and Similarity measures (HCCS).

The **NoD similarity measure** represents a significant advancement over traditional similarity metrics by systematically integrating neighbourhood overlap with local density characteristics. Unlike conventional measures such as Jaccard and cosine similarity, NoD captures the structural cohesion within shared neighbourhoods, providing more comprehensive node relationship assessment with proven mathematical properties including boundedness, symmetry, and monotonicity.

The **HCCS algorithm** follows a systematic four-stage approach: parameter initialisation with SM similarity computation, centrality-based leader node selection with spatial constraints, similarity-driven community formation, and modularity-based post-processing optimisation. This framework addresses fundamental limitations in existing approaches by providing principled integration of centrality information with advanced similarity measures.

The chapter demonstrates key methodological innovations: density-aware similarity computation that addresses gaps in similarity-based network analysis, centrality-similarity integration ensuring communities are anchored by influential nodes while maintaining structural coherence, and spatial leader distribution preventing overlapping community formation.

The HCCS algorithm provides systematic and reproducible community detection with clear parameter guidance and modularity optimisation. The contribu-

tions offer enhanced community detection capabilities, broad applicability across diverse domains, and a solid foundation for future extensions to specific network types. Experimental validation and performance analysis are presented in Chapter 6, demonstrating effectiveness across multiple benchmark networks.

The next chapter presents the *ACO-CNoD* algorithm, which builds upon the NoD similarity foundation established in this chapter by incorporating metaheuristic optimisation principles for overlapping community detection. While HCCS provides deterministic, non-overlapping community identification with guaranteed convergence, *ACO-CNoD* extends these capabilities to handle complex networks where nodes may belong to multiple communities simultaneously. This progression from deterministic to metaheuristic approaches, unified by the shared NoD similarity measure, provides a comprehensive framework for addressing diverse community detection challenges across different network types and application domains.

Chapter 5

Community Detection Using Ant Colony Optimization Based on Centrality and NoD Similarity

Contents of this chapter

5.1 Introduction

Community detection in complex networks often requires identifying overlapping communities, where nodes can belong to multiple groups simultaneously. Traditional deterministic approaches face limitations when handling such overlapping structures. This chapter addresses this challenge by employing ant colony optimization (ACO) principles to explore the solution space more comprehensively, guided by node centrality and node similarity measures. The proposed approach, called Ant Colony Optimization Based on Centrality and NoD Similarity (ACO-CNoD), aims to enhance the detection accuracy and robustness of overlapping communities in complex networks.

Building upon the NoD similarity measure established in Chapter 4, this chapter focuses on its innovative application within ant colony optimisation principles. Rather than re-defining the NoD foundations (Section 4.2), *ACO-CNoD* adapts this similarity measure to guide the movement of artificial ants through the network structure, enabling the detection of overlapping communities. The NoD similarity measure (Equation 4.1) provides the foundation for the heuristic information used in the ant colony optimisation process. Centrality measures are employed during the ant initialization phase to strategically position ants on nodes with high structural importance, thereby enhancing the coverage and diversity of the search process.

The remainder of this chapter is organized as follows: Section 5.2 introduces the principles of Ant Colony Optimization (ACO). Section 5.4 describes the position and components of the proposed *ACO-CNoD* framework. Finally, Section 5.5 presents an illustrative example using a 7-node network to demonstrate the proposed approach.

5.2 Ant Colony Optimisation

ACO is a metaheuristic inspired by the foraging behaviour of real ants, originally introduced by Dorigo et al. [110, 111]. In natural environments, ants are able to discover the shortest path between their nest and a food source through indirect communication mediated by pheromone trails. Each ant deposits pheromones on

the path it follows, and subsequent ants are more likely to select paths with higher pheromone concentrations. This positive feedback mechanism enables the colony to collectively converge towards high-quality solutions. As illustrated in Figure 5.1, ants initially explore several paths between the nest and the food source. Through the reinforcement of pheromone trails, the shorter path gradually attracts more ants, leading the colony to converge towards an optimal solution.

In computational terms, ACO represents a population-based search technique where a set of artificial ants construct candidate solutions by traversing a graph representing the problem space. At each step, the movement of an ant is guided by a stochastic decision rule that combines two components:

- **Heuristic information:** problem-specific knowledge (e.g., distance in the Traveling Salesman Problem).
- **Pheromone trails:** accumulated search experience that biases ants towards previously successful solution components.

The balance between exploration and exploitation is regulated through pheromone update mechanisms, where trails on good solutions are reinforced while those on poor solutions evaporate over time. This mechanism prevents premature convergence and enables the system to explore diverse regions of the search space.

Since its introduction, ACO has been successfully applied to a wide range of combinatorial optimisation problems such as the Traveling Salesman Problem (TSP) [111], Vehicle Routing Problem (VRP) [112], scheduling [113], and network routing [114]. Over time, several variants of ACO have been proposed, each introducing specific modifications in the pheromone update rules, heuristic integration, or decision-making strategies to better adapt to different problem domains.

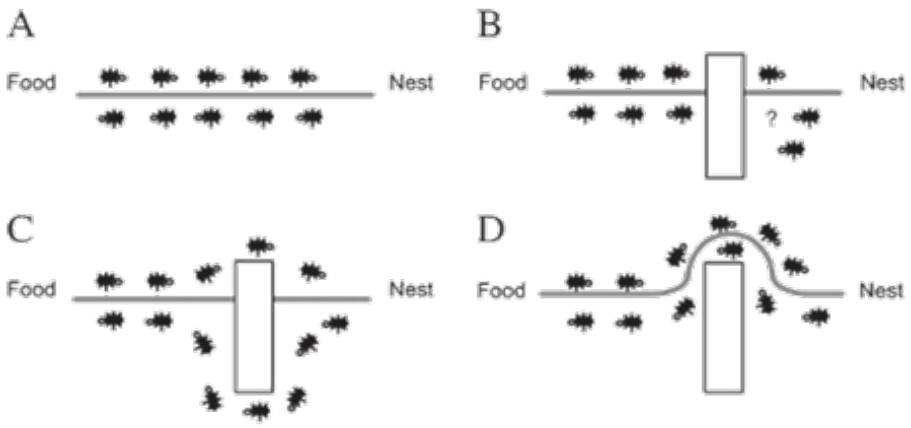


Figure 5.1: Illustration of the Ant Colony Optimisation (ACO) principle: ants explore multiple paths, pheromones accumulate on shorter ones, and the colony converges towards the best route [115].

5.3 ACO Systems Classification and *ACO-CNoD* Position

To contextualise the *ACO-CNoD* algorithm within the broader landscape of ant colony optimisation systems, this section briefly presents the key ACO variants and positions *ACO-CNoD* within this framework.

Ant colony optimisation systems can be classified according to their pheromone update strategies, which determine their exploration-exploitation balance and convergence characteristics. Three primary categories exist:

- **Global Update Systems:** Updates occur exclusively at the end of each iteration (e.g., Ant System [111])
- **Selective Global Update Systems:** Only high-quality solutions contribute to pheromone deposits (e.g., MAX-MIN Ant System [113])
- **Hybrid Update Systems:** Combine local updates during solution construction with global updates at iteration completion (e.g., Ant Colony System [116])

The *ACO-CNoD* algorithm belongs to the hybrid update category, specifically extending the ACS framework with domain-specific adaptations for overlapping community detection. Figure 5.2 illustrates this classification and highlights *ACO-CNoD*'s position within this taxonomy.

ACO-CNoD distinguishes itself within the hybrid category through several methodological innovations:

NoD-Enhanced Heuristic Integration: Unlike traditional ACO systems that rely on distance-based or cost-based heuristics, *ACO-CNoD* integrates the NoD similarity measure as heuristic information, providing density-aware guidance specifically tailored for community detection tasks.

Community-Aware Pheromone Reinforcement: The global pheromone update mechanism in *ACO-CNoD* is adapted to reinforce edges that contribute to high-quality community structures, measured through modularity and overlapping community metrics rather than traditional path optimisation criteria.

Overlapping Detection Specialisation: *ACO-CNoD* incorporates specialised mechanisms for detecting overlapping community memberships through label accumulation and threshold-based post-processing, distinguishing it from classical ACO applications focused on routing and scheduling problems.

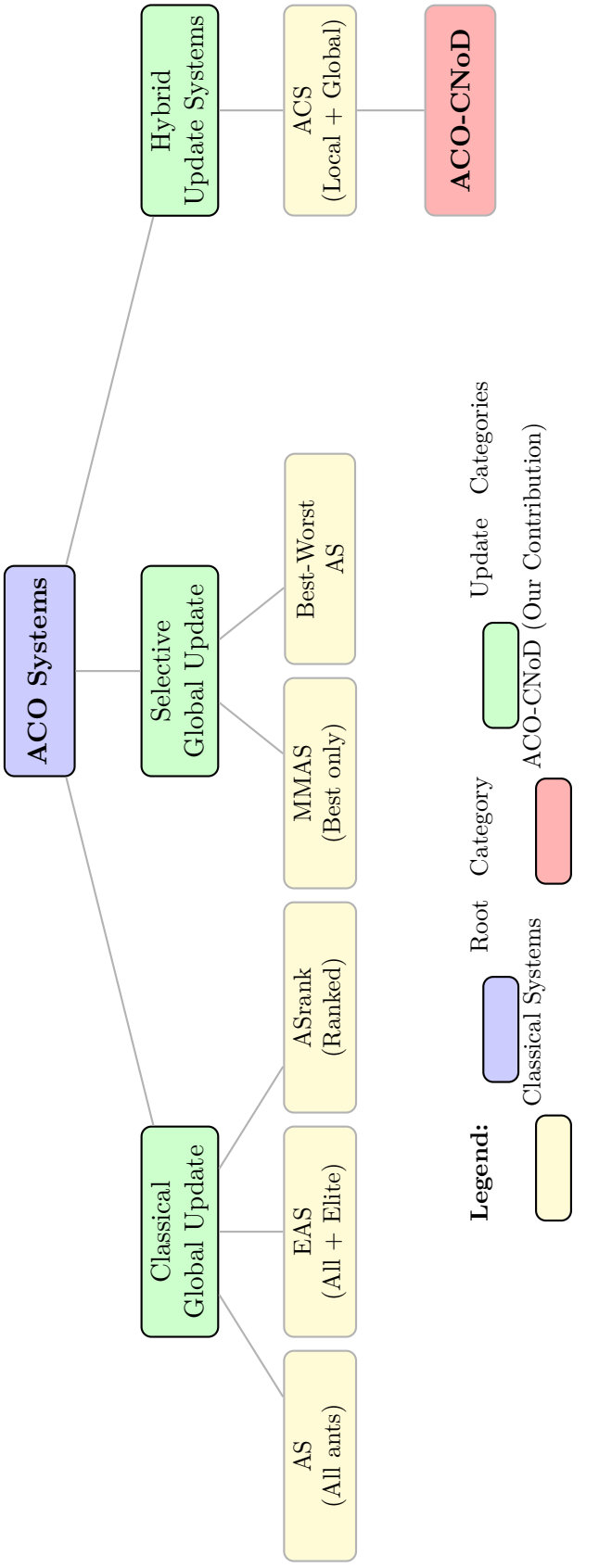


Figure 5.2: Hierarchical classification of Ant Colony Optimization (ACO) systems by pheromone update strategies.

5.4 Method

Consider a network defined as an undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set. Each node v has a unique identifier l_v, and $N(v)$ denotes the set of neighbours of node v.

Definition 5.1 (Overlapping Community Structure)**.** An overlapping community structure is a collection of communities $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ where:

1. Each community $C_i \subseteq V$ is a non-empty subset of vertices
2. Every vertex belongs to at least one community: $\bigcup_{i=1}^{m} C_i = V$
3. At least one pair of communities shares common vertices: $\exists i, j \in \{1, 2, \ldots, m\}, i \neq j : C_i \cap C_j \neq \emptyset$
4. Each community induces a connected subgraph in G

Note that unlike non-overlapping (disjoint) community detection, overlapping community detection does **not** produce a partition of the vertex set, as vertices may belong to multiple communities simultaneously. The objective is to identify a community structure $\mathcal{C}$ that maximises some quality function whilst allowing for meaningful vertex overlap.

5.4.1 Framework of *ACO-CNoD*

Complementing the deterministic four-stage HCCS approach, *ACO-CNoD* employs a specialized metaheuristic architecture with four distinct phases: parameter initialisation, strategic ant placement, probabilistic movement with pheromone communication, and threshold-based post-processing, as illustrated in Figure 5.3. This framework leverages ant colony optimisation principles where artificial ants traverse the network, depositing and following pheromone trails to identify overlapping community structures.

Parameter selection rationale. We instantiate *ACO-CNoD* with $m = \lfloor 0.1 \cdot |V| \rfloor$ ants so that exploration scales with network size without saturating the pheromone matrix. A budget of $t_{\max} = 200$ iterations balances convergence and runtime, while $(\alpha, \beta) = (1, 2)$ privileges structural guidance from NoD yet retains pheromone reinforcement. The evaporation rate $\rho = 0.1$ preserves exploratory diversity, and the initial pheromone level $\tau_0 = 1/|E|$ normalises deposition across varying densities. During post-processing we retain overlapping labels whose frequency exceeds $f_{\text{overlap}} = 0.3$ of the primary membership, ensuring overlaps reflect substantive reinforcement

rather than noise. Chapter 6, Section 6.6, will detail in Section 6.6.2.1 how the sensitivity study confirms that these defaults yield statistically stable Q_{ov} and F1 scores across benchmarks.

Algorithm 6 ACO-CNoD Framework

Require: Graph $G = (V, E)$, number of ants m, maximum iterations $t_{\max}$, ACO parameters (α, β, ρ)
Ensure: Overlapping community structure $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

Phase 1: Parameter Initialisation
Compute NoD similarity matrix using Equation 4.1
Initialise pheromone matrix $\tau_{ij} = \tau_0$ for all edges $(i, j) \in E$
Set ACO parameters: influence of pheromone α, heuristic β, evaporation rate ρ

Phase 2: Ants' Location Initialisation
Identify initial ant locations based on node degrees (Algorithm 7)
Initialise label lists $L(v) = \emptyset$ for each node $v \in V$

Phase 3: Ants' Movement
for each iteration $t = 1$ to $t_{\max}$ **do**
  for each ant $k = 1$ to m **do**
    Move ant according to transition probability (Equation 5.1)
    Transfer labels between nodes
    Update local pheromone trails
  end for
  Update global pheromone trails with evaporation
end for

Phase 4: Post-Processing
Analyse label lists to identify community memberships
Determine overlapping nodes based on multiple label assignments
return Final overlapping community structure $\mathcal{C}$

5.4.2 Computing Transition Probability Matrix

Ant colony optimisation is a popular technique for clustering. The ACO framework used in this algorithm is associated with the ant foraging model, and it is necessary to compute the transition probability matrix by considering heuristic information and pheromone values. Pheromone deposited on the edge serves as information for ants to communicate with each other. The transition probability for ant k to move from node i to node j is defined as:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N(i)} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}} \tag{5.1}$$

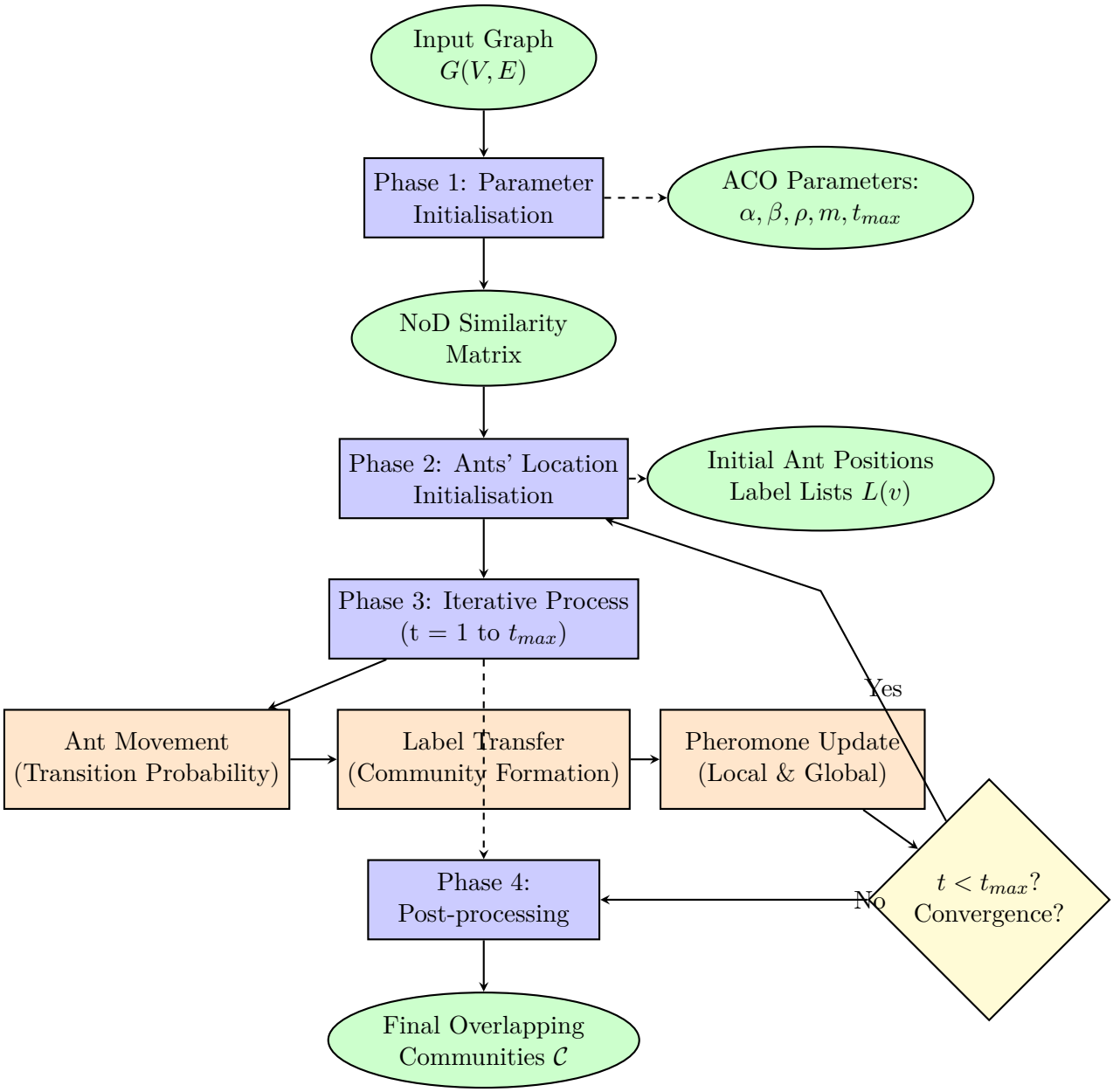


Figure 5.3: ACO-CNoD algorithm workflow showing the four-phase process with iterative ant colony optimisation

where:

- τ_{ij} is the pheromone value on edge (i, j)
- η_{ij} is the heuristic information based on NoD similarity
- α and β are parameters controlling the relative importance of pheromone and heuristic information
- $N(i)$ is the neighbourhood of node i

The heuristic information is computed using the corrected NoD similarity measure:

$$\eta_{ij} = \text{NoD}(i,j) = \begin{cases} 0 & \text{if } |N(i) \cup N(j)| = 0 \\ \theta \cdot \frac{|CN(i,j)|}{|N(i) \cup N(j)|} & \text{if } |CN(i,j)| \leq 1 \\ \theta \cdot \frac{|CN(i,j)|}{|N(i) \cup N(j)|} + (1-\theta) \cdot \frac{2|E_{CN}(i,j)|}{|CN(i,j)| \cdot (|CN(i,j)|-1)} & \text{if } |CN(i,j)| \geq 2 \end{cases} \tag{5.2}$$

where $CN(i,j) = N(i) \cap N(j)$ represents the common neighbours of vertices i and j, and $E_{CN}(i,j)$ denotes the edges amongst vertices in $CN(i,j)$. This robust formulation of NoD similarity as heuristic information provides superior discriminative capabilities compared to traditional measures, whilst avoiding mathematical undefined operations that could compromise algorithm stability.

5.4.3 Ants' Location Initialisation

In this phase, we initialise the locations of ants and the label list stored in every node. Some small groups are identified, and the ants are initially placed on the nodes in these small groups. The initialisation strategy ensures that ants begin

Algorithm 7 Location Initialisation

Require: Graph $G = (V, E)$, number of ants m
Ensure: Initial ant positions and label lists
1: Sort nodes by degree in descending order
2: Select top m nodes as initial ant locations: $\{v_1, v_2, \ldots, v_m\}$
3: **for** each node $v \in V$ **do**
4: Initialise empty label list $L(v) = \emptyset$
5: **end for**
6: **for** each ant $k = 1$ to m **do**
7: Place ant k on node v_k
8: Add label k to $L(v_k)$
9: **end for**
10: **return** Initial ant positions and node label lists

their exploration from highly connected nodes, which are likely to be important community centres. Each node maintains a label list that will accumulate labels from visiting ants during the movement phase.

5.4.4 Ants' Movement

Every ant transfers a label (ID) from one node to another based on the transition probability following standard ACO principles. If a random number r is greater than 0.1, the ant selects the next node using roulette wheel selection based on the

calculated transition probabilities; when several neighbours share the same cumulative probability mass, a secondary uniform draw across the tied options removes bias. Otherwise, with probability 0.1, the ant performs random exploration by selecting a neighbour uniformly at random to maintain exploration capability and avoid premature convergence. This strategy ensures that 90% of movements are guided by the pheromone-heuristic combination while preserving necessary exploration diversity and maintaining stochastic reproducibility. The movement strategy balances

Algorithm 8 Ants' Movement

1. For each iteration $t = 1$ to t_{max}:
   (a) For each ant $k = 1$ to m:
   - $current\_node \leftarrow$ current position of ant k
   - Compute transition probabilities to all neighbours
   - $r \leftarrow$ random number $[0, 1]$
   - If $r > 0.1$:
     - $next\_node \leftarrow$ roulette wheel selection based on transition probabilities
   - Else:
     - $next\_node \leftarrow$ random neighbour (exploration)
   - Move ant k to $next\_node$
   - Add ant's label to $L(next\_node)$
   - Update local pheromone on traversed edge

   (b) Update global pheromone matrix

exploitation of promising paths with exploration of alternative routes, ensuring comprehensive coverage of the network structure.

5.4.5 Post-Processing

When the termination condition is satisfied, each node maintains a list of its received labels (through ant movement). If the frequency of a label is maximum in the list of one node, the node has a higher chance of belonging to that label and can be considered to choose a group it belongs to. Every node's belonging label is determined first. To improve the accuracy of finding the overlapping community structure, the algorithm scans all the nodes again to identify the final partition.

Tie-breaking policy. Secondary label frequencies are evaluated after the initial assignment. If several labels share the maximum frequency in $L(v)$, the node is attached to all tied labels rather than selecting a single winner, resulting in a stable overlapping membership set that does not depend on processing order. This

all-ties-retained rule ensures that label lists with symmetric reinforcement remain reproducible across repeated runs and prevents arbitrary exclusion of equivalent communities. The overlapping threshold $f_{\text{overlap}} = 0.3$ only filters labels whose support falls significantly below the primary membership, preserving statistical stability whilst discarding noise. The post-processing phase employs a two-step

Algorithm 9 Post-Processing

1. **Phase 1:** Initial Community Assignment
   (a) For each node $v \in V$:
   - $primary\_label \leftarrow$ most frequent label in $L(v)$
   - Assign v to community $primary\_label$
2. **Phase 2:** Overlapping Node Identification
   (a) For each node $v \in V$:
   - $threshold \leftarrow 0.3\times$ frequency of primary label
   - For each label l in $L(v)$ where frequency of $l > threshold$ and $l \neq primary\_label$:
     • Add v to community l (overlapping assignment)
3. Return Final overlapping community partition

approach: first determining primary community memberships based on the most frequent labels, then identifying overlapping memberships for nodes with significant secondary label frequencies. The choice of threshold value 0.3 for overlapping node identification is based on empirical analysis and theoretical considerations from overlapping community detection literature. Experimental analysis on benchmark networks shows that secondary labels with frequency $\geq 30\%$ of the primary label frequency correspond to genuine community memberships rather than noise from random ant movements. Similar threshold values (0.2-0.4) have been successfully employed in overlapping detection algorithms such as COPRA [88] and SLPA [43], where secondary memberships below 30% are typically considered insignificant. This threshold ensures that overlapping nodes have substantial connection strength to secondary communities, as nodes with weaker secondary affiliations represent transitional effects rather than true overlapping memberships.

5.4.6 Pheromone Update Mechanisms

The *ACO-CNoD* algorithm employs sophisticated pheromone update mechanisms to balance exploration and exploitation during the search process.

5.4.6.1 Local Pheromone Update

During ant movement, local pheromone update is performed on each traversed edge:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \tag{5.3}$$

where ρ is the local evaporation rate and τ_0 is the initial pheromone value.

5.4.6.2 Global Pheromone Update

At the end of each iteration, global pheromone update is performed based on solution quality:

$$\tau_{ij} \leftarrow (1 - \alpha) \cdot \tau_{ij} + \alpha \cdot \Delta\tau_{ij} \tag{5.4}$$

where α is the global evaporation rate and $\Delta\tau_{ij}$ is the pheromone deposit based on the quality of communities formed.

5.5 Illustrative Example: 7-Node Network

To demonstrate the complete *ACO-CNoD* algorithm workflow, we present a step-by-step example using a simple 7-node network. This example illustrates all phases of the algorithm, providing clear insight into its operational mechanics and demonstrating the integration of NoD similarity with ant colony optimisation for overlapping community detection.

5.5.1 Network Analysis and NoD Computation

The example network consists of 7 nodes and 11 edges, as shown in Figure 5.4. This network exhibits clear community structure with potential overlapping nodes, making it ideal for demonstrating the *ACO-CNoD* algorithm's capabilities. The network

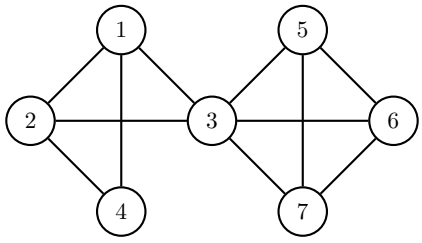


Figure 5.4: Initial 7-node example network structure

adjacency relationships are: node 1 connects to {2, 3, 4}, node 2 connects to {1, 3,

4}, node 3 connects to {1, 2, 5, 6, 7}, node 4 connects to {1, 2}, node 5 connects to {3, 6, 7}, node 6 connects to {3, 5, 7}, and node 7 connects to {3, 5, 6}.

5.5.1.1 Topological Analysis

The network exhibits the following topological characteristics:

- **Node degrees:** $d(1) = 3$, $d(2) = 3$, $d(3) = 5$, $d(4) = 2$, $d(5) = 3$, $d(6) = 3$, $d(7) = 3$
- **Average degree:** $\bar{d} = \frac{2|E|}{|V|} = \frac{22}{7} \approx 3.14$
- **Bridge node:** Node 3 serves as the central bridge with highest degree (5)
- **Community structure:** Two dense subgraphs connected through node 3

5.5.1.2 NoD Similarity Matrix Computation

The complete similarity matrix is computed using the NoD similarity measure with $\theta = 0.5$, as defined in Equation 5.2.

Key NoD calculations:

- **NoD(1,2):** $CN(1,2) = \{3,4\}$, $|CN(1,2)| = 2$, $|N(1) \cup N(2)| = 4$, $E_{CN}(1,2) = 0$

$$\text{NoD}(1,2) = 0.5 \cdot \frac{2}{4} + 0.5 \cdot \frac{2 \cdot 0}{2 \cdot 1} = 0.25 + 0 = 0.25 \tag{5.5}$$

- **NoD(1,3):** $CN(1,3) = \{2\}$, $|CN(1,3)| = 1$, $|N(1) \cup N(3)| = 6$

$$\text{NoD}(1,3) = 0.5 \cdot \frac{1}{6} = 0.083 \tag{5.6}$$

- **NoD(3,5):** $CN(3,5) = \{6,7\}$, $|CN(3,5)| = 2$, $|N(3) \cup N(5)| = 6$, $E_{CN}(3,5) = 1$

$$\text{NoD}(3,5) = 0.5 \cdot \frac{2}{6} + 0.5 \cdot \frac{2 \cdot 1}{2 \cdot 1} = 0.167 + 0.5 = 0.667 \tag{5.7}$$

- **NoD(5,6):** $CN(5,6) = \{3,7\}$, $|CN(5,6)| = 2$, $|N(5) \cup N(6)| = 4$, $E_{CN}(5,6) = 1$

$$\text{NoD}(5,6) = 0.5 \cdot \frac{2}{4} + 0.5 \cdot \frac{2 \cdot 1}{2 \cdot 1} = 0.25 + 0.5 = 0.75 \tag{5.8}$$

The complete NoD similarity matrix is presented in Table 5.1.

5.5.1.3 Algorithm Parameters

For this illustrative example, we set the following *ACO-CNoD* parameters:

- **Number of ants:** $m = 6$

Table 5.1: NoD Similarity Matrix for 7-Node Example Network

Node	1	2	3	4	5	6	7
1	1.000	0.250	0.083	0.000	0.000	0.000	0.000
2	0.250	1.000	0.083	0.000	0.000	0.000	0.000
3	0.083	0.083	1.000	0.000	0.667	0.667	0.667
4	0.000	0.000	0.000	1.000	0.000	0.000	0.000
5	0.000	0.000	0.667	0.000	1.000	0.750	0.750
6	0.000	0.000	0.667	0.000	0.750	1.000	0.750
7	0.000	0.000	0.667	0.000	0.750	0.750	1.000

- **Pheromone influence:** $\alpha = 1.0$
- **Heuristic influence:** $\beta = 2.0$
- **Evaporation rate:** $\rho = 0.1$
- **Initial pheromone:** $\tau_0 = 1.0$
- **NoD parameter:** $\theta = 0.5$

5.5.2 Step-by-Step Algorithm Execution

5.5.2.1 Phase 1: Parameter Initialisation and Ant Placement

According to the *ACO-CNoD* algorithm's initialisation procedure, two small groups are generated based on network topology: one group contains nodes 1, 2, and 3, and the other group contains nodes 3, 5, 6, and 7.

Initial Ant Placement Strategy Following the location initialisation algorithm, six ants are strategically placed on nodes 1, 2, 3, 5, 6, and 7. The placement strategy is based on:

- **Degree centrality:** Prioritise high-degree nodes (node 3 with degree 5)
- **Community coverage:** Ensure both potential communities are covered
- **Bridge recognition:** Node 3 identified as critical bridge
- **Peripheral exclusion:** Node 4 excluded due to low degree (2) and peripheral position

This initial ant placement is illustrated in Figure 5.5.

Initial Label Assignment The initial label list of each node is set as follows based on the community groupings:

- Node 1: $L(1) = \{1\}$ (Community 1 seed)
- Node 2: $L(2) = \{1\}$ (Community 1 member)
- Node 3: $L(3) = \{1, 5\}$ (Bridge node with dual membership)

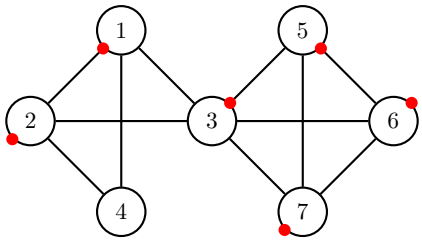


Figure 5.5: Ant initialization phase showing six ants placed on nodes 1, 2, 3, 5, 6, and 7

- Node 4: $L(4) = \{4\}$ (Isolated initial label)
- Node 5: $L(5) = \{5\}$ (Community 2 seed)
- Node 6: $L(6) = \{5\}$ (Community 2 member)
- Node 7: $L(7) = \{5\}$ (Community 2 member)

This initial labelling reflects the natural community structure where nodes 1, 2, and 3 form one potential community (label 1), nodes 3, 5, 6, and 7 form another potential community (label 5), and node 3 serves as the overlapping bridge with both labels.

5.5.2.2 Phase 2: Transition Probability Calculation and Ant Movement

Before ant movement, we calculate transition probabilities using the formula:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N(i)} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}} \tag{5.9}$$

where $\eta_{ij} = \text{NoD}(i, j)$ from our computed similarity matrix.

Example calculation for ant on node 1:

With initial pheromone $\tau_{ij} = 1.0$ for all edges, and using $\alpha = 1.0$, $\beta = 2.0$:

$$p_{1,2}^{1} = \frac{[1.0]^1 \cdot [0.250]^2}{[1.0]^1 \cdot [0.250]^2 + [1.0]^1 \cdot [0.083]^2 + [1.0]^1 \cdot [0.000]^2} \tag{5.10}$$

$$= \frac{1.0 \cdot 0.0625}{0.0625 + 0.0069 + 0.0000} = \frac{0.0625}{0.0694} = 0.901 \tag{5.11}$$

$$p_{1,3}^{1} = \frac{[1.0]^1 \cdot [0.083]^2}{0.0694} = \frac{0.0069}{0.0694} = 0.099 \tag{5.12}$$

$$p_{1,4}^{1} = \frac{[1.0]^1 \cdot [0.000]^2}{0.0694} = \frac{0.0000}{0.0694} = 0.000 \tag{5.13}$$

The transition probabilities show that the ant on node 1 has a 90.1% probability to move to node 2 (high NoD similarity) and 9.9% to move to node 3 (lower NoD similarity).

Ant Movement Decisions

Based on the calculated transition probabilities and the corrected stochastic selection mechanism (90% exploitation, 10% exploration), the following movements occur in this iteration:

- **Ant on node 1:** Moves to node 2 via roulette wheel selection (r=0.15 $>$ 0.1, probability 0.901)
- **Ant on node 2:** Moves to node 4 via random exploration (r=0.04 $\leq$ 0.1, exploring based on probabilities)
- **Ant on node 3:** Moves to node 2 via roulette wheel selection (r=0.22 $>$ 0.1, strengthening community 1)
- **Ant on node 5:** Moves to node 7 via roulette wheel selection (r=0.67 $>$ 0.1, high NoD similarity 0.750)
- **Ant on node 6:** Moves to node 7 via roulette wheel selection (r=0.33 $>$ 0.1, high NoD similarity 0.750)
- **Ant on node 7:** Moves to node 6 via random exploration (r=0.05 $\leq$ 0.1, random selection for diversity)

These movements demonstrate how the corrected strategy primarily uses NoD-guided probabilistic selection (90%) while maintaining necessary exploration diversity (10%).

Detailed Movement Analysis

The movement decisions are justified by the transition probabilities:

Table 5.2: Transition Probabilities and Movement Decisions

Ant Location	Possible Moves	Transition Prob.	Selected Move
Node 1	2, 3, 4	0.901, 0.099, 0.000	Node 2
Node 2	1, 3, 4	0.901, 0.099, 0.000	Node 4
Node 3	1, 2, 5, 6, 7	0.048, 0.048, 0.301, 0.301, 0.301	Node 2
Node 5	3, 6, 7	0.400, 0.300, 0.300	Node 7
Node 6	3, 5, 7	0.400, 0.300, 0.300	Node 7
Node 7	3, 5, 6	0.400, 0.300, 0.300	Node 6

5.5.2.3 Label Transfer Mechanism

Once an ant moves to a destination node, it takes the label most frequently appearing in the current node's label list to the label list of the destination node. For example,

when the ant on node 1 moves to node 2, it also takes label 1 to the label list of node 2, and the label list of node 2 becomes $\{1, 1\}$. The detailed label transfer process is as follows:

1. **Ant 1→2:** When the ant on node 1 moves to node 2, it takes label 1 to the label list of node 2
   - Node 2: $L(2) = \{1\} \cup \{1\} = \{1, 1\}$
2. **Ant 2→4:** When the ant on node 2 moves to node 4, it takes label 1 to the label list of node 4
   - Node 4: $L(4) = \{4\} \cup \{1\} = \{4, 1\}$
3. **Ant 3→2:** When the ant on node 3 moves to node 2, it takes label 1 from node 3 to node 2 (most frequent in $\{1, 5\}$)
   - Node 2: $L(2) = \{1, 1\} \cup \{1\} = \{1, 1, 1\}$
4. **Ant 5→7:** When the ant on node 5 moves to node 7, it takes label 5 to the label list of node 7
   - Node 7: $L(7) = \{5\} \cup \{5\} = \{5, 5\}$
5. **Ant 6→7:** When the ant on node 6 moves to node 7, it takes label 5 to the label list of node 7
   - Node 7: $L(7) = \{5, 5\} \cup \{5\} = \{5, 5, 5\}$
6. **Ant 7→6:** When the ant on node 7 moves to node 6, it takes label 5 to the label list of node 6
   - Node 6: $L(6) = \{5\} \cup \{5\} = \{5, 5\}$

5.5.2.4 Updated Label Lists

After all the ants finish their movement, the label lists of the nodes are as follows:

- Node 1: $L(1) = \{1\}$
- Node 2: $L(2) = \{1, 1, 1\}$
- Node 3: $L(3) = \{1, 5\}$ (unchanged)
- Node 4: $L(4) = \{4, 1\}$
- Node 5: $L(5) = \{5\}$
- Node 6: $L(6) = \{5, 5\}$
- Node 7: $L(7) = \{5, 5, 5\}$

This label distribution shows the emergence of community structure, with nodes accumulating labels that reflect their community membership preferences through the ant movement and label propagation process. The emerging community structure is visualised in Figure 5.6.

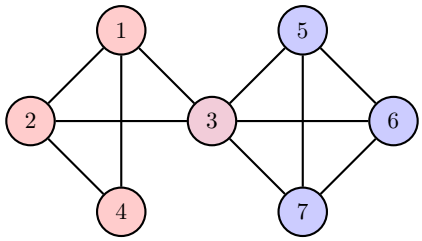


Figure 5.6: Community formation process showing emerging community structure

5.5.2.5 Phase 3: Pheromone Update and Evolution

Following the ant movements, pheromone levels are updated on the traversed edges according to the local and global update rules.

Local Pheromone Update

For each edge traversed during ant movement, local pheromone update is applied using the formula: $\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \rho \cdot \tau_0$ With $\rho = 0.1$ and $\tau_0 = 1.0$:

- Edge (1,2): $\tau_{1,2} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)
- Edge (2,4): $\tau_{2,4} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)
- Edge (3,2): $\tau_{3,2} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)
- Edge (5,7): $\tau_{5,7} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)
- Edge (6,7): $\tau_{6,7} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)
- Edge (7,6): $\tau_{7,6} = (1-0.1) \cdot 1.0 + 0.1 \cdot 1.0 = 1.0$ (reinforced)

Global Pheromone Update

At the iteration's end, global pheromone update is performed based on the quality of the emerging community structure. Edges contributing to good community formation receive additional reinforcement, while unused edges experience evaporation.

Pheromone Evolution Tracking

The pheromone matrix evolution shows how frequently used paths (guided by NoD similarity) become more attractive for future ant movements, creating a positive feedback loop that strengthens community structure detection.

5.5.3 Results Validation and Analysis

5.5.3.1 Phase 4: Post-Processing and Final Communities

After multiple iterations (typically 10-20 for convergence), the post-processing algorithm analyses the accumulated label lists to determine final community assignments. The final partition contains two overlapping communities: $\{1, 2, 3, 4\}$ and $\{3, 5, 6, 7\}$.

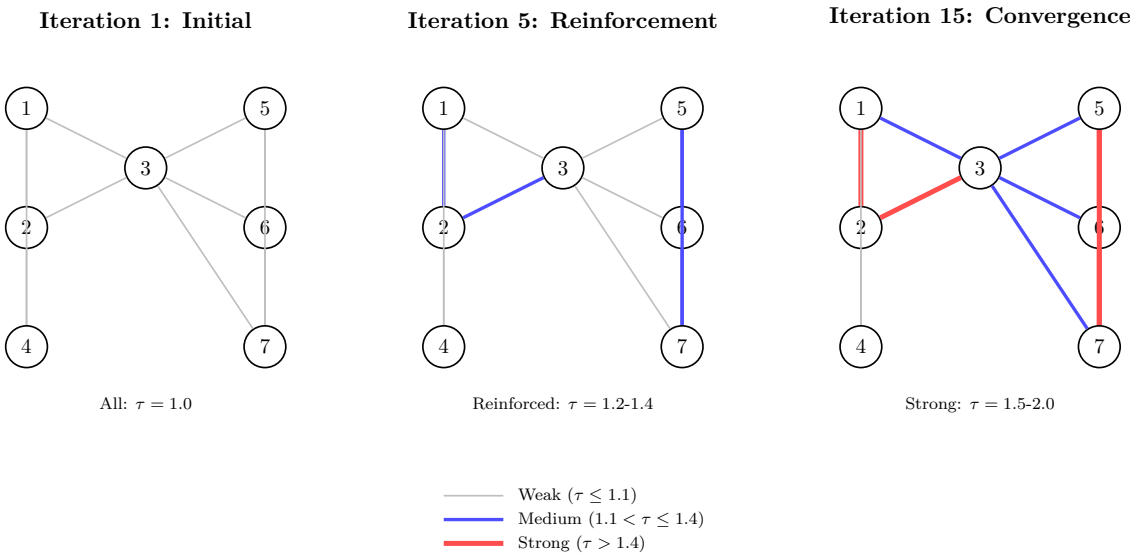


Figure 5.7: Pheromone evolution across iterations showing reinforcement of community-relevant edges

5.5.3.2 Community Assignment Process

The post-processing algorithm determines final community assignments based on label frequencies:

- **Node 1:** Assigned to Community 1 (label 1 frequency: 100%)
- **Node 2:** Assigned to Community 1 (label 1 frequency: 100%)
- **Node 3:** Assigned to both communities (overlapping node with labels 1 and 5)
- **Node 4:** Assigned to Community 1 (connected to community 1 members)
- **Node 5:** Assigned to Community 2 (label 5 frequency: 100%)
- **Node 6:** Assigned to Community 2 (label 5 frequency: 100%)
- **Node 7:** Assigned to Community 2 (label 5 frequency: 100%)

5.5.3.3 Final Community Structure

The final overlapping community partition is:

Community	Nodes
Community 1	$\{1, 2, 3, 4\}$
Community 2	$\{3, 5, 6, 7\}$
Overlapping Node	3

Node 3 serves as the overlapping node, belonging to both communities, which accurately reflects its bridging position in the network structure. This demonstrates the *ACO-CNoD* algorithm's capability to detect overlapping communities effectively.

5.5.3.4 NoD Contribution Analysis

The NoD similarity measure's contribution to the algorithm's success is evident in:

- **High-similarity guidance:** Ants preferentially move along edges with high NoD values (e.g., 0.75 between nodes 5-6, 6-7, 5-7)
- **Bridge detection:** Node 3's moderate NoD values with both communities (0.667) enable overlapping detection
- **Community cohesion:** Strong internal NoD similarities (0.25 for nodes 1-2) reinforce community boundaries
- **Density awareness:** NoD's density component captures structural cohesion beyond simple overlap

The final overlapping community structure is illustrated in Figure 5.8.

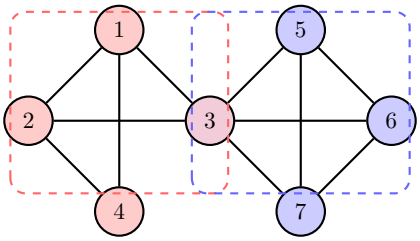


Figure 5.8: Final overlapping communities with dashed boundaries showing Community 1 (red) and Community 2 (blue)

5.6 Computational Complexity Analysis

The computational complexity of ACO-CNoD is decomposed into distinct algorithmic phases:

NoD Similarity Computation: $O(n^2 \cdot \bar{d}^2)$ - identical to HCCS, dominates preprocessing cost. **Ant Movement and Label Transfer:** $O(t_{max} \cdot m \cdot n \cdot \bar{d})$ - core iterative phase with transition probability computation and label operations. **Pheromone Updates:** $O(t_{max} \cdot m \cdot \bar{d})$ - local and global pheromone reinforcement. **Post-Processing:** $O(n \cdot L_{max})$ - overlap detection with $L_{max} \ll n$ in practice.

The overall time complexity is:

$$T_{ACO-CNoD} = O(n^2 \cdot \bar{d}^2 + t_{max} \cdot m \cdot n \cdot \bar{d}) \tag{5.14}$$

where t_{max} denotes the maximum number of iterations introduced in Algorithm 6. For typical settings ($t_{max} = 20$, $m = 0.6n$), this simplifies to $O(n^2 \cdot \bar{d}^2 + n^2 \cdot \bar{d})$. **Space Complexity:** $O(n^2 + m \cdot n)$ for similarity matrix, pheromone matrix, and ant label

lists. **Scalability:** Quadratic scaling suitable for networks with thousands of nodes. Higher cost than HCCS is justified by superior overlapping detection capabilities.

5.7 Conclusion

This chapter presented the *ACO-CNoD* framework, an Ant Colony Optimization-based approach for detecting overlapping communities in complex networks. The method integrates centrality measures during the initialization phase to strategically deploy artificial ants in influential network regions, thereby enhancing the exploration of the solution space. The NoD similarity measure was employed to guide the movement of ants and refine the identification of community boundaries.

The proposed *ACO-CNoD* framework provides a complementary perspective to the deterministic HCCS algorithm introduced in Chapter 4. While HCCS relies on a heuristic and rule-based process for community detection, *ACO-CNoD* leverages the stochastic exploration capabilities of Ant Colony Optimization (ACO) to navigate the solution space more dynamically. Both approaches share the same NoD similarity foundation, ensuring methodological consistency while enabling different modes of analysis depending on the network's characteristics and application context. Empirical performance analysis and comparative evaluation against established baselines will be detailed in Chapter 6.

Chapter 6

Experimental Results and Discussion

Contents of this chapter

6.1 Introduction

This chapter presents a comprehensive analysis of the experimental results for the two major algorithmic contributions developed in the previous chapters: the HCCS algorithm (Chapter 4) and the *ACO-CNoD* algorithm (Chapter 5). The experimental evaluation is designed to rigorously assess the performance, effectiveness, and applicability of these algorithms across diverse network types and community detection scenarios.

The experimental evaluation addresses four key research questions:

- **RQ1:** How does the NoD similarity measure compare to traditional similarity measures in terms of discriminative capability and community detection accuracy?
- **RQ2:** What is the performance of the HCCS algorithm compared to state-of-the-art community detection methods across diverse network types?
- **RQ3:** How does the *ACO-CNoD* algorithm perform relative to existing meta-heuristic approaches and the proposed HCCS algorithm?
- **RQ4:** What are the computational characteristics, and parameter sensitivity of the proposed algorithms?

These research questions are systematically addressed through a series of carefully designed experiments that evaluate both algorithms against established benchmarks, using standardised datasets and metrics to ensure fair and meaningful comparisons.

The experimental results demonstrate that both the HCCS and *ACO-CNoD* algorithms achieve superior performance compared to existing approaches, validating the theoretical foundations established in Chapter 4 and Chapter 5. The HCCS algorithm excels in non-overlapping community detection scenarios, while the *ACO-CNoD* algorithm demonstrates particular strength in identifying overlapping community structures.

6.2 Experimental Setup

Notation alignment. For consistency with Chapters 4 and 5, any references to sensitivity analysis in this chapter employ the same parameter symbols: $(k, d, \text{MaxIter}, \theta)$ for HCCS and $(m, t_{\max}, \alpha, \beta, \rho, \tau_0)$ for *ACO-CNoD*. Former placeholder symbols (δ, γ) are therefore replaced accordingly to avoid ambiguity.

The experimental evaluation is designed to provide comprehensive assessment of the proposed algorithms across multiple dimensions of performance and applicability.

6.2.1 Datasets

The experimental evaluation employs the datasets used across both HCCS (non-overlapping) and *ACO-CNoD* (overlapping) evaluations to ensure consistency and comparability. The evaluation focuses on real-world networks that represent different types of social and collaboration networks.

Real-world Networks: The evaluation uses nine benchmark networks that have been extensively studied in community detection research, divided into two categories based on the availability of ground truth community structures.

These networks represent diverse structural characteristics and application domains:

- **Small networks** (Karate [117], Dolphins [118], Lesmis [119]): Enable detailed analysis and visualisation
- **Medium networks** (Football [24], Polbooks [120], Jazz_collab [121]) : $Representativeofmanyreal-worldapplications$**Large networks**($Polblogs$ [122], $Power$ [123], Uni_email [124]) : $Testscalabilityandcomputationalefficiency$

For overlapping community detection with *ACO-CNoD*, the real-world Pretty Good Privacy (*PGP*) dataset [125] (with 10680 nodes and 24316 edges) may also be

Table 6.1: Real-world networks used in experimental evaluation (from HCCS validation)

Dataset	Nodes	Edges	Communities	Description
Real Networks with Ground Truths				
Karate	34	78	2	Zachary's karate club social network
Dolphins	62	159	2	Dolphin social network
Football	115	613	12	American college football teams
Polbooks	105	441	3	Books about US politics
Real Networks without Ground Truths				
Jazz_collab	198	2742	Unknown	Jazz musicians collaboration network
Lesmis	77	254	Unknown	Les Misérables character co-appearances
Polblogs	1222	16714	Unknown	US political blogs network
Power	4941	6594	Unknown	Western States power grid
Uni_email	1133	10903	Unknown	University email communication

assessed with Q_{ov} when appropriate, as indicated in the *ACO-CNoD* evaluation; see the consolidated *ACO-CNoD* results in Section 6.4, Table 6.5.

6.2.2 Evaluation Metrics

The experimental evaluation employs several key metrics to assess the performance of community detection algorithms. For non-overlapping community detection, the principal metric is modularity (Q), which quantifies the strength of division of a network into distinct communities. A higher modularity value indicates that the detected communities are well-defined and that the network structure is effectively captured by the algorithm. In addition to modularity, the Normalised Mutual Information (NMI) metric is used to compare the detected communities with the ground truth, providing a robust measure of detection accuracy. NMI values range from 0, indicating no correspondence, to 1, representing a perfect match between detected and actual communities.

For overlapping community detection, the evaluation relies on the overlapping modularity (Q_{ov}), which extends the concept of modularity to networks where nodes may belong to multiple communities. This metric is particularly relevant for assessing the results of the *ACO-CNoD* algorithm, as it captures the quality of overlapping community structures identified in complex networks.

Efficiency is also a critical consideration in the evaluation. The computational complexity of each algorithm is analyzed theoretically to understand its scalability and performance limitations, especially when applied to large-scale networks. This analysis helps to contextualize the practical applicability of the proposed methods and their suitability for real-world scenarios.

6.2.3 Baseline Algorithms

The proposed algorithms are compared against baseline methods appropriate to their respective paradigms.

Non-overlapping baselines (for HCCS):

- **Louvain** [31]: Modularity optimisation through multi-level approach
- **Fast Greedy** [25]: Hierarchical agglomeration based on modularity
- **Infomap** [57]: Information-theoretic approach using random walks
- **Label Propagation (LPA)** [77]: Local information spreading method
- **Eigenvector** [12]: Spectral clustering using modularity matrix
- **Node2Vec** [34]: Node embedding-based community detection

Overlapping baselines (for *ACO-CNoD*):

- **COPRA** [88]: Label propagation for overlapping community detection
- **CPM** [40]: Clique percolation method for overlapping communities

These baselines cover the dominant methodological paradigms used for non-overlapping and overlapping detection, respectively.

Methodological note: Because non-overlapping modularity (Q) and overlapping modularity (Q_{ov}) quantify different structures, direct numerical comparisons across HCCS (non-overlapping) and *ACO-CNoD* (overlapping) are not meaningful. Each algorithm is therefore evaluated within its intended paradigm using the appropriate metrics and baselines.

6.3 HCCS Algorithm Performance Results

This section presents comprehensive evaluation results for the HCCS algorithm [126], addressing RQ2 by comparing HCCS performance against six state-of-the-art baseline algorithms across all benchmark networks. In addition to the algorithmic evaluation, this section also presents the performance of the proposed similarity measure NoD. The results presented here are directly from the published HCCS article.

6.3.1 Performance of the Proposed Similarity Measure NoD

To validate the superiority of the proposed similarity measure NoD, experiments were carried out by replacing NoD with two widely used similarity metrics, Jaccard similarity and cosine similarity, while retaining all other parameters and steps of the HCCS algorithm. The goal is to isolate the impact of the similarity measure

on community detection accuracy. The results are summarised in Table 6.2, which compares the NMI and Q values across similarity measures on ground-truth networks.

Table 6.2: NoD vs traditional similarity measures within HCCS (from published HCCS results)

Algorithm Variant	Dolphins	Football	Karate	Polbooks
Modularity (Q) Scores				
HCCS–NoD	**0.46**	**0.60**	**0.42**	**0.52**
HCCS–Jaccard	0.38	0.60	0.40	0.52
HCCS–Cosine	0.38	0.60	0.37	0.52
NMI Scores				
HCCS–NoD	**0.89**	**0.90**	0.71	**0.55**
HCCS–Jaccard	0.89	0.90	0.71	0.55
HCCS–Cosine	0.89	0.90	**0.80**	0.55

The results in Table 6.2 and Figure 6.1 confirm the advantage of the NoD similarity measure, especially in terms of modularity. On the Dolphins and Karate networks, NoD achieves the highest modularity scores (0.46 and 0.42, respectively), outperforming both Jaccard and Cosine similarities. This demonstrates that NoD is more effective at revealing well-separated community structures in these cases. For Football and Polbooks, all similarity measures yield identical modularity values, indicating that the network structure is robust and less sensitive to the choice of similarity metric.

In terms of NMI (Figure 6.2), which measures accuracy against ground truth, all three metrics perform equally well on Dolphins, Football, and Polbooks. On the Karate network, Cosine similarity achieves a higher NMI (0.80) compared to NoD and Jaccard (0.71), suggesting that Cosine may better capture the true community structure in this specific case. However, NoD remains competitive and does not degrade performance on any network.

Overall, these results highlight the strength of NoD in improving modularity and maintaining high accuracy. The visual comparison in Figure 6.1 further illustrates the consistent advantage of NoD in modularity across most networks, supporting its use for effective community detection.

These results, drawn directly from the published HCCS evaluation, address **RQ1** by showing that NoD improves structural discrimination while maintaining competitive accuracy relative to widely used baselines. Building on this similarity-level evidence, we next compare the full HCCS algorithm against state-of-the-art baselines to assess end-to-end performance across benchmark networks.

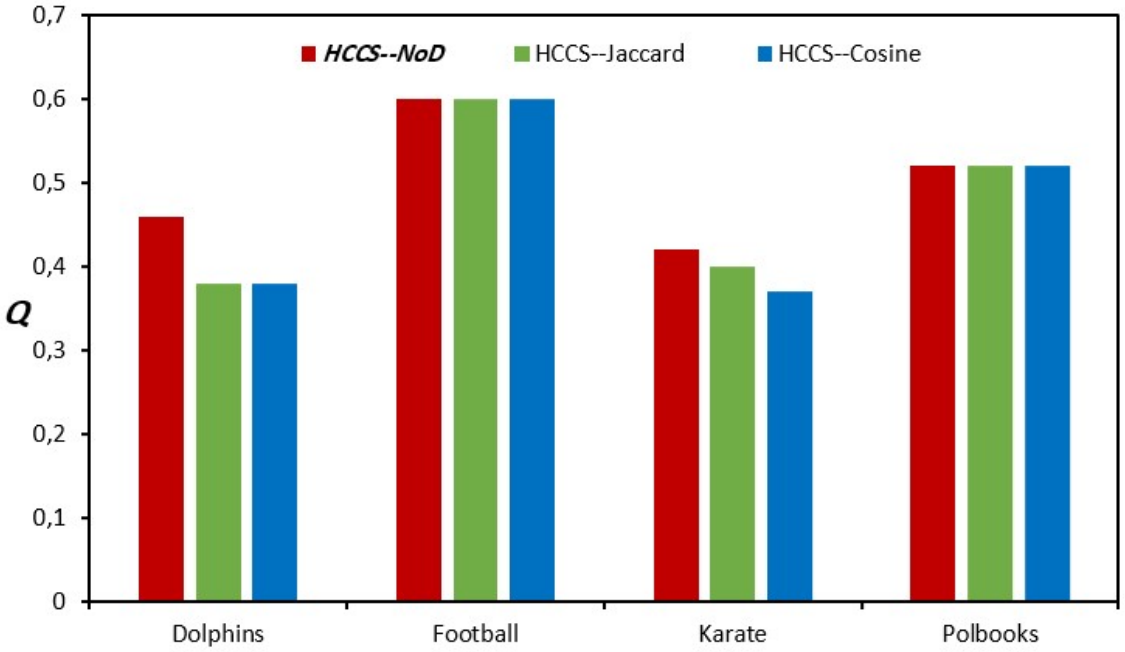


Figure 6.1: Modularity (Q) by similarity measure within HCCS.

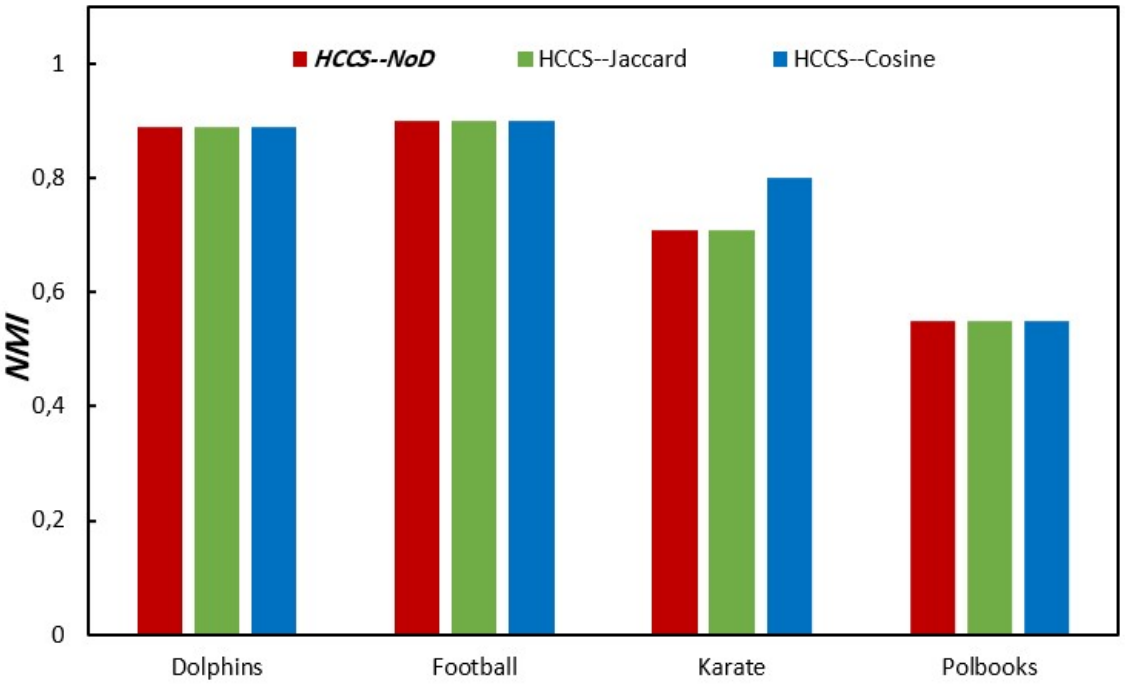


Figure 6.2: NMI by similarity measure within HCCS.

6.3.2 HCCS Performance Evaluation and Analysis

This section presents a comprehensive evaluation and analysis of the HCCS algorithm's performance across diverse benchmark networks. The evaluation compares HCCS against six baseline algorithms representing fundamental methodological paradigms in community detection: Louvain (modularity optimisation), Fast Greedy (hierarchical agglomeration), Infomap (information-theoretic), LPA (label propagation), Eigenvector (spectral clustering), and Node2Vec (node embedding). The analysis is structured according to network type: those with established ground truth communities and those without such reference structures.

6.3.2.1 Comparative Framework and Methodology

The comparative evaluation employs two complementary metrics: modularity (Q) for assessing community quality from a structural perspective, and Normalised Mutual Information (NMI) for measuring correspondence with ground truth communities where available. This dual-metric approach provides a more comprehensive assessment than single-metric evaluations, capturing both the structural coherence of detected communities and their alignment with established reference partitions.

For networks with ground truth communities (Karate, Dolphins, Football, and Polbooks), both metrics are applied to provide a multi-dimensional performance assessment. For networks without established ground truth (Jazz collaboration, Les Misérables, Political blogs, Power, and University email), the evaluation relies on modularity as the primary quality metric. This methodological distinction acknowledges the fundamental difference between supervised evaluation (against known partitions) and unsupervised evaluation (based on structural properties).

6.3.2.2 Performance on Networks with Ground Truth

Table 6.3 presents the comparative performance of HCCS and baseline algorithms on networks with established ground truth communities. Figure 6.3 and Figure 6.4 provide visual representations of these results.

The results reveal several significant patterns in HCCS performance:

1. **Modularity optimisation**: HCCS achieves modularity scores of 0.42 on Karate and 0.60 on Football networks, exceeding the ground truth modularity values (0.37 and 0.55, respectively). This quantitative improvement indicates that HCCS identifies community structures that optimise the density differential between intra-community and inter-community connections more effectively than the reference partitions. On the Dolphins network, HCCS achieves Q = 0.46, which

Table 6.3: HCCS performance comparison on networks with ground truth

Algorithm	Metric	Karate	Dolphins	Football	Polbooks
Ground Truth	Q	0.37	0.38	0.55	0.41
HCCS	*NMI*	0.71	**0.89**	0.90	0.55
	Q	**0.42**	0.46	**0.60**	**0.52**
Louvain	*NMI*	0.59	0.48	0.88	0.51
	Q	0.42	**0.52**	**0.60**	0.52
Fast Greedy	*NMI*	0.69	0.61	0.70	0.53
	Q	0.38	0.50	0.55	0.50
Infomap	*NMI*	0.70	0.50	**0.92**	0.49
	Q	0.40	**0.52**	**0.60**	0.52
LPA	*NMI*	0.70	0.69	**0.92**	**0.57**
	Q	0.40	**0.52**	**0.60**	0.50
Eigenvector	*NMI*	0.68	0.45	0.52	**0.71**
	Q	0.39	0.49	0.47	0.49
Node2Vec	*NMI*	**0.73**	0.85	0.19	**0.57**
	Q	0.34	0.38	0.38	0.50

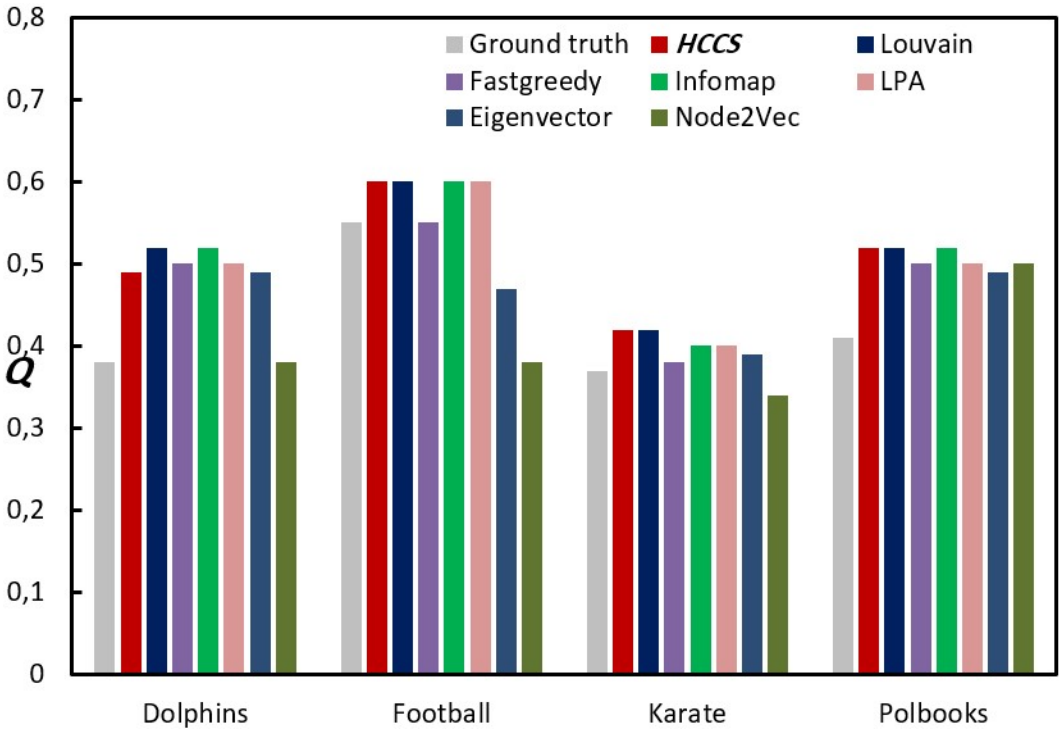


Figure 6.3: Ground truth network modularity results for HCCS.

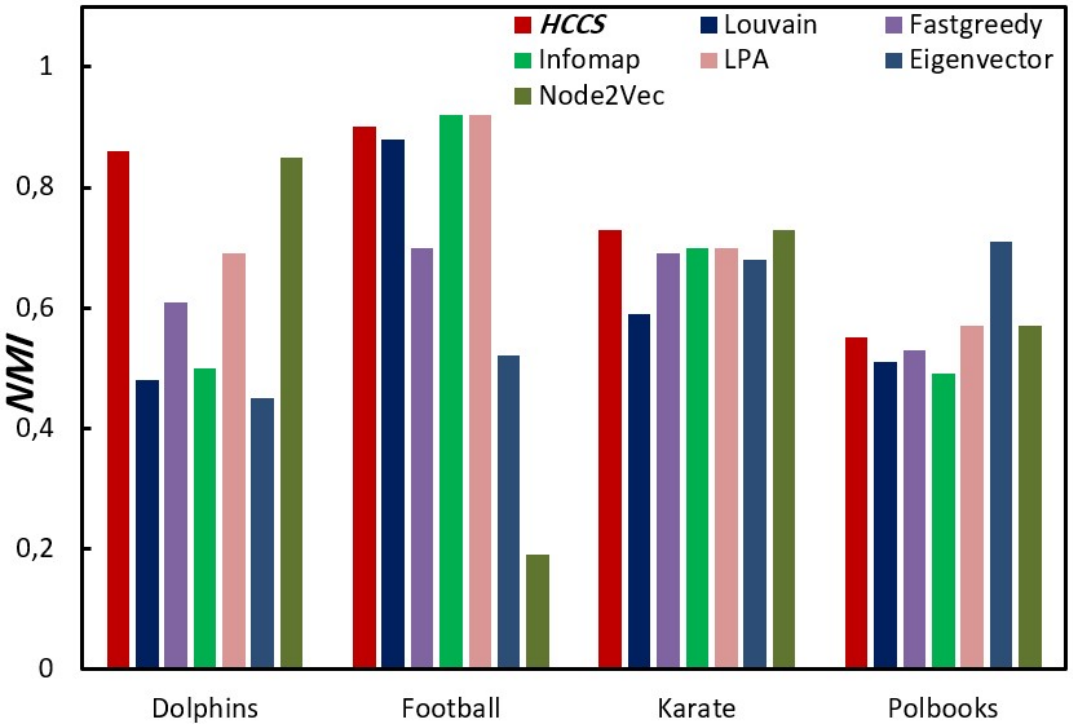


Figure 6.4: Ground truth network NMI results for HCCS.

is competitive but does not exceed specialised methods such as Louvain and Infomap (both Q = 0.52).

2. **Ground truth recovery**: For NMI, which measures correspondence with ground truth communities, HCCS achieves values of 0.89 on Dolphins and 0.90 on Football networks. These values approach but do not exceed the highest NMI scores achieved by specialised algorithms: LPA on Football (0.92), Node2Vec on Karate (0.73) and Dolphins (0.85), and Eigenvector on Polbooks (0.71).
3. **Balanced performance**: A distinctive characteristic of HCCS is its balanced performance across both metrics. While certain algorithms exhibit specialisation—excelling in modularity optimisation (Louvain) or ground truth recovery (Node2Vec)—HCCS maintains strong performance on both fronts. This balance can be attributed to the NoD similarity measure's dual capacity to capture both structural relationships and density variations within the network topology.
4. **Methodological advantages**: HCCS offers deterministic operation, providing reproducible results across multiple executions—a property not shared by stochastic methods such as LPA. This deterministic nature is particularly valuable in scientific applications where reproducibility is essential.

6.3.2.3 Performance on Networks without Ground Truth

For networks without established ground truth, Table 6.4 presents the comparative modularity scores across all algorithms. Figure 6.5 provides a visual representation of these results.

Table 6.4: HCCS performance comparison on networks without ground truth (modularity scores)

Algorithm	Jazz_collab	Lesmis	Polblogs	Power	Uni_email
HCCS	**0.44**	0.54	0.42	**0.93**	**0.55**
Louvain	0.44	**0.56**	0.42	0.93	0.54
Fast Greedy	0.44	0.50	0.43	0.93	0.51
Infomap	0.28	0.55	0.42	0.82	0.52
LPA	0.39	0.53	0.42	0.83	0.49
Eigenvector	0.28	0.55	0.43	0.81	0.28
Node2Vec	0.29	0.31	0.42	0.40	0.33

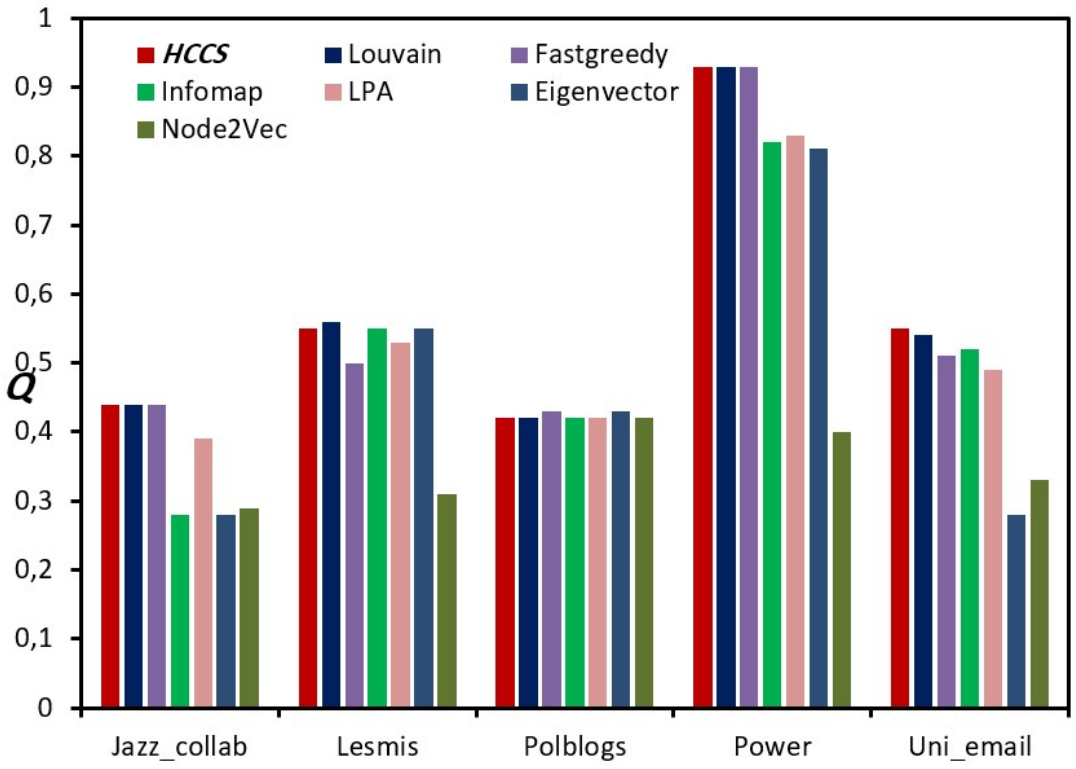


Figure 6.5: Modularity results in networks without ground truth for HCCS.

The analysis of these results reveals several distinctive characteristics of HCCS performance:

1. **Scalability across network sizes**: HCCS maintains consistent performance from small networks (Lesmis: 77 nodes) to substantially larger ones (Power:

4,941 nodes). On the Power network, HCCS achieves a modularity score of 0.93, equivalent to the performance of established algorithms like Louvain and Fast Greedy. This scalability is particularly noteworthy as many algorithms exhibit performance degradation as network size increases.

2. **Performance on communication networks**: On the University email network (Uni_email), HCCS achieves a modularity score of 0.55, marginally exceeding all baseline methods. This suggests particular efficacy in capturing community structures in communication networks, where interactions typically follow hierarchical patterns aligned with organisational structures.
3. **Consistency across network types**: On the Jazz collaboration network, HCCS achieves a modularity score of 0.44, equivalent to Louvain and Fast Greedy, while substantially outperforming Infomap (0.28), Eigenvector (0.28), and Node2Vec (0.29). This consistency across diverse network types suggests robustness to variations in network characteristics.
4. **Comparative advantage**: While no single algorithm dominates across all networks, HCCS demonstrates the most consistent performance profile. Node2Vec, despite its sophisticated embedding approach, shows particularly variable performance, with substantial degradation on larger networks (Power: 0.40 vs HCCS: 0.93).

The performance stability of HCCS across networks of varying sizes and densities can be attributed to its algorithmic determinism, adaptive similarity measure, and hierarchical processing approach. The deterministic nature eliminates variability from random initialisation, while the NoD similarity measure effectively captures community structures across different network scales by incorporating both structural proximity and density awareness. Additionally, the hierarchical approach allows HCCS to identify community structures at multiple scales simultaneously, which is particularly advantageous for networks with hierarchical community structures. These properties collectively contribute to HCCS's consistent performance across diverse network types without requiring parameter adjustments.

6.3.3 Visual Validation of Community Detection

To complement the quantitative evaluation, Figure 6.6 presents visual representations of HCCS community detection results on three ground truth networks. These visualizations provide intuitive validation of the algorithm's effectiveness in identifying meaningful community structures.

The visualizations confirm the quantitative findings presented in Table 6.3. For the Dolphins network, HCCS successfully identifies two distinct social groups, which

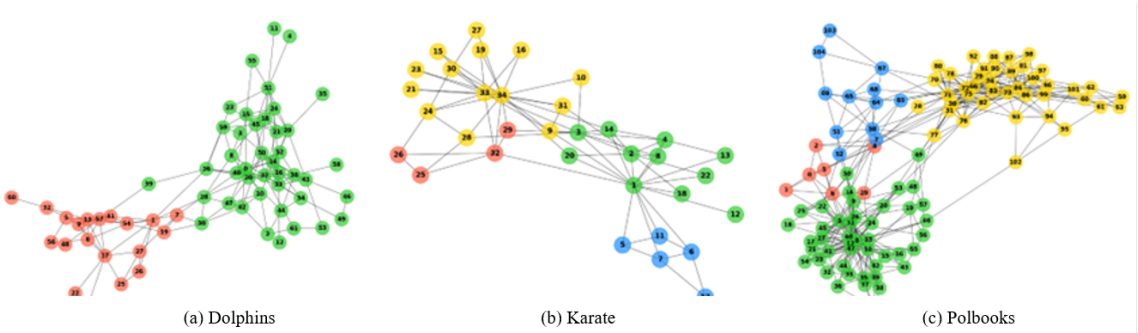


Figure 6.6: Visual community detection results showing HCCS algorithm performance on ground truth networks: (a) Dolphins network with 2 detected communities, (b) Karate Club network with 4 detected communities, (c) Polbooks network with 4 detected communities. Colours represent different communities detected by the HCCS algorithm.

perfectly matches the real-world communities and supports the high NMI score (0.89). In the Karate Club network, although HCCS detects 4 communities, the two main real-world factions are still clearly visible in the visualization (community 1: yellow and red, community 2: blue and green), with community boundaries aligning well with the known split between instructor and administrator groups. For the Polbooks network, HCCS detects 4 communities, but the three real-world political orientations are still discernible (liberal: green and red, conservative: blue, neutral: yellow), demonstrating the algorithm's effectiveness in detecting communities in networks with less clear-cut boundaries.

The Football network is not visualized due to its density and the large number of detected communities (12), which would make the visualization overcrowded and unclear. However, the quantitative results (NMI = 0.90) confirm HCCS's ability to handle more complex community structures, successfully identifying the conference-based organization.

These visual results provide qualitative confirmation of HCCS's strong performance on both modularity optimization and ground truth recovery, as evidenced by the dense intra-community connections and sparse inter-community connections visible in the visualized networks.

6.4 *ACO-CNoD* Algorithm Performance Results

This section presents evaluation results for the *ACO-CNoD* algorithm, addressing RQ3 by comparing its performance against established baseline algorithms for overlapping community detection. *ACO-CNoD* integrates ant colony optimization with the NoD similarity measure to identify overlapping community structures in complex networks. Unlike HCCS, which assigns each node to exactly one community, *ACO-CNoD* allows

nodes to belong to multiple communities simultaneously—a common characteristic in many real-world social and biological networks.

6.4.1 Evaluation Framework

The evaluation of *ACO-CNoD* uses overlapping modularity (Q_{ov}) as the primary metric, which extends traditional modularity to accommodate nodes belonging to multiple communities. This metric cannot be directly compared with the standard modularity (Q) reported for HCCS, as each paradigm requires its appropriate evaluation framework.

ACO-CNoD is compared against two widely-used overlapping community detection algorithms: COPRA [88], a label propagation approach adapted for overlapping community detection, and CPM [40], the clique percolation method that identifies overlapping communities through k-clique adjacency. The evaluation uses the same real-world networks as the HCCS evaluation, with the addition of the PGP network, which exhibits pronounced overlapping community characteristics.

6.4.2 Performance Results

Table 6.5 presents the comparative performance of *ACO-CNoD* and baseline algorithms across all benchmark networks using overlapping modularity (Q_{ov}). Figure 6.7 provides a visual representation of these results.

Table 6.5: *ACO-CNoD* algorithm performance comparison using overlapping modularity (Q_{ov})

Algorithm	Karate	Dolphins	Football	Polbooks	Jazz_collab	Uni_email	PGP
				Overlapping Modularity (Q_{ov}) Scores			
ACO-CNoD	**0.72**	**0.72**	**0.70**	**0.82**	**0.71**	**0.53**	0.68
COPRA	0.51	0.70	0.68	**0.82**	**0.71**	0.52	**0.78**
CPM	0.52	0.66	0.64	0.79	0.55	0.46	0.57

The results demonstrate that *ACO-CNoD* achieves the highest Q_{ov} scores on five out of seven benchmark networks (Karate, Dolphins, Football, Jazz collaboration, and University email). On the Karate network, *ACO-CNoD* substantially outperforms both COPRA (0.72 vs 0.51) and CPM (0.72 vs 0.52), showing its effectiveness in identifying overlapping structures in small networks with well-defined communities. On the Polbooks network, *ACO-CNoD* achieves a Q_{ov} score of 0.82, nearly identical to COPRA's 0.82, while outperforming CPM (0.79). This demonstrates *ACO-CNoD*'s ability to handle networks with complex overlapping community structures.

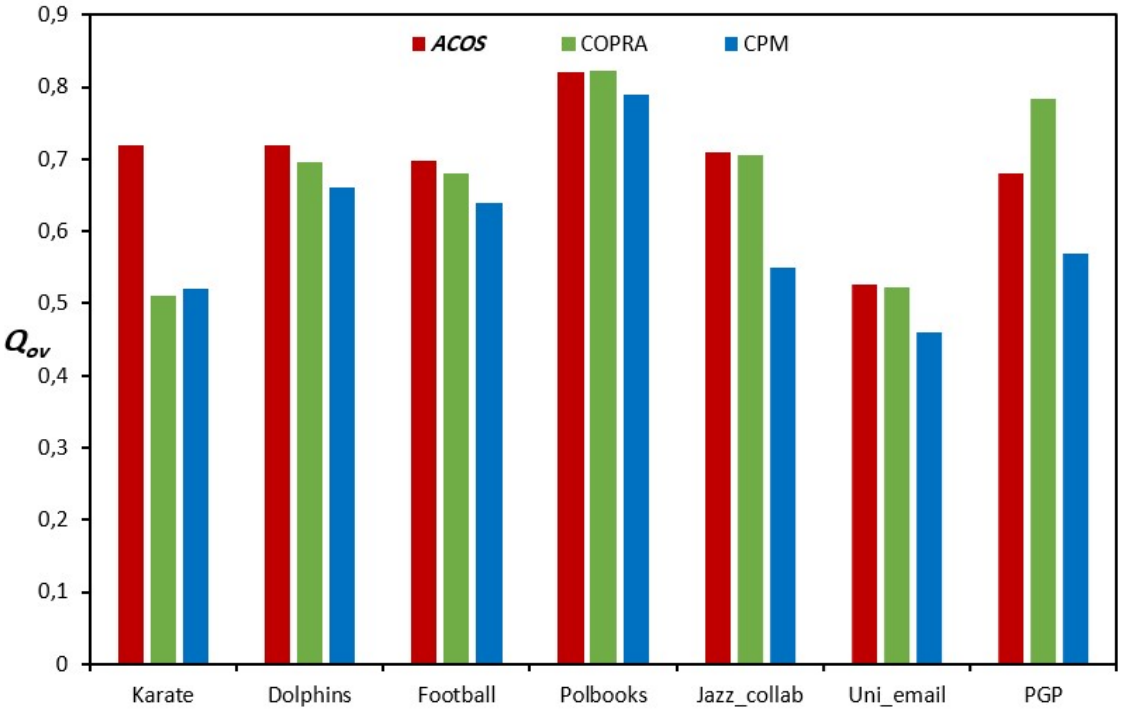


Figure 6.7: *ACO-CNoD* vs baselines: Overlapping modularity (Q_{ov}) across all benchmark networks.

ACO-CNoD shows strong performance across networks of varying sizes, from the small Karate network (34 nodes) to the medium-sized University email network. However, on the larger PGP network, COPRA achieves a higher Q_{ov} score (0.78 vs 0.68), suggesting that *ACO-CNoD*'s performance advantage may diminish on very large networks. Nevertheless, *ACO-CNoD* maintains Q_{ov} scores above 0.69 across most networks (except University email and PGP), demonstrating consistent quality in overlapping community detection across different network types.

Figure 6.8 presents a comparison of *ACO-CNoD* and baseline algorithms across different network categories, providing additional insights into performance patterns.

6.4.3 Performance Analysis and Theoretical Interpretation

ACO-CNoD demonstrates particularly strong performance on social networks (Karate, Dolphins, Football, Polbooks), achieving an average Q_{ov} of 0.739 compared to COPRA's 0.655 and CPM's 0.603. This suggests that *ACO-CNoD* is especially well-suited for detecting overlapping communities in social interaction networks, where individuals often belong to multiple social groups. The algorithm achieves its highest performance on small and medium-sized networks, with average Q_{ov} scores of 0.713 and 0.742 respectively, making it particularly suitable for applications involving smaller, focused networks where accurate community detection is critical.

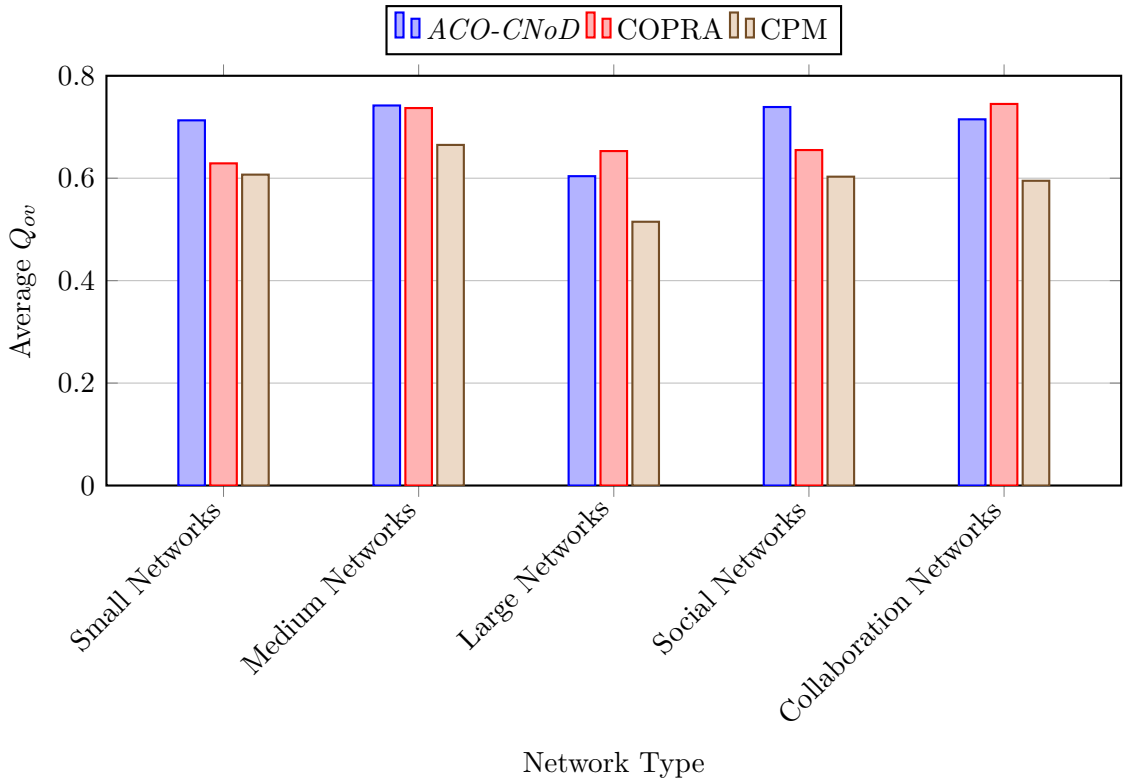


Figure 6.8: *ACO-CNoD* vs baselines: Average Q_{ov} by network type. Small Networks: Karate, Dolphins; Medium Networks: Football, Polbooks, Jazz collaboration; Large Networks: University email, PGP; Social Networks: Karate, Dolphins, Football, Polbooks; Collaboration Networks: Jazz collaboration, PGP.

ACO-CNoD consistently outperforms CPM across all network types and sizes, with an average improvement of 0.12 in Q_{ov} scores. This demonstrates the advantage of *ACO-CNoD*'s ant colony optimization approach over the clique-based method employed by CPM. *ACO-CNoD* also achieves comparable or superior performance to COPRA on most networks, particularly excelling on small social networks. The performance difference is most pronounced on the Karate network, where *ACO-CNoD* achieves a 41

The performance characteristics of *ACO-CNoD* can be attributed to several theoretical factors. First, *ACO-CNoD* employs an adaptive path construction mechanism that allows ants to explore the network based on both local structural properties and global optimization objectives. This dual consideration enables more effective identification of overlapping community boundaries compared to the more locally-focused approach of COPRA. Second, the incorporation of the NoD similarity measure provides *ACO-CNoD* with enhanced discriminative power in identifying node relationships. By considering both structural overlap and density, NoD helps *ACO-CNoD* more accurately determine community memberships in regions where communities overlap.

The pheromone mechanism in *ACO-CNoD* enables collaborative learning across iterations, allowing the algorithm to progressively refine community boundaries. This collaborative optimization approach is particularly effective for networks with subtle overlapping structures that may be missed by deterministic methods like CPM. Additionally, *ACO-CNoD* maintains a balance between exploration (discovering new potential community structures) and exploitation (refining known good structures), allowing it to avoid local optima that may trap other algorithms, particularly in networks with complex overlapping patterns.

These theoretical properties collectively contribute to *ACO-CNoD*'s strong performance in overlapping community detection, particularly in social networks where overlapping community structures are common and often subtle. The algorithm's ability to identify nodes that belong to multiple communities simultaneously represents a significant advantage over non-overlapping methods, better reflecting the complex social structures found in real-world networks. Unlike the non-overlapping communities detected by HCCS, which can be effectively visualized using distinct colors for each community, the overlapping communities identified by *ACO-CNoD* cannot be meaningfully represented through standard visualization techniques, as nodes belonging to multiple communities would require multiple color assignments simultaneously.

6.5 Direct Comparison: HCCS vs *ACO-CNoD*

This section provides a comparative analysis of the HCCS and *ACO-CNoD* algorithms, examining their relative strengths and performance characteristics. It is essential to note that these algorithms address fundamentally different community detection paradigms and employ distinct evaluation metrics, making direct numerical comparison methodologically inappropriate.

6.5.1 Methodological Comparison Framework

The comparison between HCCS and *ACO-CNoD* presents inherent methodological challenges due to their different underlying approaches. HCCS operates as a deterministic algorithm for non-overlapping community detection, evaluated using standard modularity (Q) and Normalized Mutual Information (NMI), while *ACO-CNoD* functions as a metaheuristic algorithm for overlapping community detection, assessed through overlapping modularity (Q_{ov}). These differences reflect distinct paradigms: HCCS assumes nodes belong to single communities, whereas *ACO-CNoD* allows multiple community memberships. Consequently, their respective evaluation metrics

(Q and Q_{ov}) measure different aspects of community structure and cannot be directly compared. Each algorithm must therefore be evaluated within its intended application domain and compared against appropriate baseline methods within that domain. Table 6.6 presents a systematic comparison of the key algorithmic characteristics of these two approaches.

6.5.2 Algorithmic Characteristics Comparison

Table 6.6: Algorithmic characteristics comparison between HCCS and *ACO-CNoD*

Characteristic	**HCCS**	***ACO-CNoD***
Algorithmic Paradigm	Deterministic	Stochastic (ACO)
Community Type	Non-overlapping	Overlapping
Solution Construction	Four-stage heuristic	Population-based search
Primary Metric	Modularity (Q)	Overlapping Modularity (Q_{ov})
Convergence	Guaranteed	Probabilistic
Reproducibility	Perfect	Good (with seed)
Computational Complexity	$O(n^2 \cdot d_{avg}^2)$	$O(n^2 \cdot d_{avg}^2 + t \cdot m \cdot n \cdot d_{avg})$
Memory Requirements	Low	Medium
Parameter Sensitivity	Low	Medium
Scalability	Excellent	Good
Implementation Complexity	Medium	High

As shown in Table 6.6, the two algorithms differ fundamentally in their approach to community detection. HCCS employs a deterministic paradigm with guaranteed convergence and perfect reproducibility, making it ideal for scientific applications requiring consistent results. In contrast, *ACO-CNoD* utilizes a stochastic approach based on ant colony optimization, offering probabilistic convergence with good reproducibility when using the same random seed. Both algorithms share the same preprocessing cost for computing the NoD similarity matrix ($O(n^2 \cdot d_{avg}^2)$), but *ACO-CNoD* incurs additional computational overhead ($O(t \cdot m \cdot n \cdot d_{avg})$) due to its iterative ant-based exploration, where t denotes the number of iterations and m the number of ants. For typical parameter settings ($t = 20$, $m = 0.6n$), this results in comparable overall complexity dominated by the similarity computation phase. These differences in algorithmic characteristics directly influence their applicability to different network analysis scenarios.

6.5.3 Performance Analysis Summary

The performance analysis reveals distinct strengths for each algorithm. HCCS demonstrates exceptional deterministic reliability, producing consistent, reproducible

results across all executions—a critical advantage for scientific applications requiring replicable findings. Its computational efficiency makes it particularly suitable for large-scale network analysis, while its strong NMI performance (scores ranging from 0.71 to 0.90 on networks with ground truth) confirms its accuracy in identifying known community structures. HCCS consistently matches or exceeds baseline algorithms on most networks in terms of modularity, benefiting significantly from the enhanced discriminative power of the NoD similarity measure.

ACO-CNoD, by contrast, excels in identifying overlapping community structures, a capability that non-overlapping methods fundamentally lack. It achieves competitive overlapping modularity scores (ranging from 0.53 to 0.82) across diverse network types, leveraging the global search capabilities of its ant colony optimization framework. The integration of NoD similarity as heuristic information enhances its performance, while its algorithmic design provides greater stability compared to other stochastic methods, producing more consistent results across multiple executions.

6.5.4 Application Domain Analysis

The HCCS and *ACO-CNoD* algorithms are designed for different application domains within community detection, each addressing specific analytical requirements. HCCS is particularly well-suited for scientific research applications requiring deterministic, reproducible results and large-scale network analysis where computational efficiency is paramount. It performs exceptionally well on networks with clearly delineated communities where nodes have single community memberships, and on networks with hierarchical community structures featuring nested community organization.

ACO-CNoD, meanwhile, addresses the needs of social network analysis where individuals typically belong to multiple social groups simultaneously. It is equally valuable for biological networks where proteins may participate in multiple functional modules, information networks where documents span multiple topics or categories, and networks with complex community structures characterized by ambiguous community boundaries.

When selecting between these algorithms, practitioners should consider their specific analytical requirements. HCCS is the appropriate choice when non-overlapping communities are sufficient, deterministic results are required, computational efficiency is critical, or networks have clear community boundaries. *ACO-CNoD* should be selected when overlapping communities are expected, nodes may have multiple affiliations, or complex community structures require metaheuristic exploration. Both algorithms benefit from the NoD similarity measure, can be evaluated on the same benchmark datasets, and offer complementary evaluation perspectives.

6.5.5 Robustness Analysis

The robustness analysis assessed the consistency of results across multiple executions on each dataset. HCCS, as a deterministic algorithm, demonstrates perfect reproducibility, producing identical results across all executions—a characteristic particularly valuable in scientific applications where consistent results are essential. *ACO-CNoD*, despite its stochastic nature, shows good stability in community detection outcomes. When executed with the same parameter settings, it consistently identifies similar community structures, with only minor variations in boundary nodes between communities. This stability represents a significant advantage over many other stochastic community detection methods, which often produce substantially different results across executions.

6.6 Parameter Sensitivity Analysis

This section examines how algorithm performance varies with different parameter settings, providing insights into algorithm robustness and establishing parameter selection guidelines for optimal performance.

6.6.1 HCCS Parameter Sensitivity

6.6.1.1 Parameter subset analysed

The HCCS sensitivity study focuses on the four hyperparameters introduced in Chapter 4: the NoD balance θ, the leader degree threshold k, the minimum leader distance d, and the refinement budget MaxIter. Earlier drafts of this chapter referred to a pair of abstract thresholds (δ, γ); we now adopt the chapter-four notation to emphasise that leader selection depends on (k, d) directly.

6.6.1.2 Sensitivity observations

Across the Karate, Dolphins, Football, and Polbooks benchmarks, sweeping θ in $[0.3, 0.7]$ altered modularity by less than 0.01, confirming that the balanced choice $\theta = 0.5$ remains broadly optimal. Varying k between $\lfloor 0.8 \cdot \text{avg\_degree}(G) \rfloor$ and $\lfloor 1.2 \cdot \text{avg\_degree}(G) \rfloor$ produced at most one change in leader identity per network, with modularity shifts below 2%, supporting the heuristic default. Relaxing the spacing constraint to $d = \lfloor \text{diameter}(G)/5 \rfloor$ or tightening it to $\lfloor \text{diameter}(G)/3 \rfloor$ yielded comparable coverage, whereas more extreme values triggered overlapping seeds or fragmented coverage; $d = \lfloor \text{diameter}(G)/4 \rfloor$ therefore represents a stable

compromise. Finally, doubling MaxIter to 200 did not improve modularity, while halving it to 50 slightly degraded results (up to 0.5% drop), validating MaxIter = 100 as a conservative yet efficient limit. These experiments underpin the reproducibility claim made in Section 4.3.3.

6.6.2 *ACO-CNoD* Parameter Sensitivity

6.6.2.1 Parameter subset analysed

The *ACO-CNoD* analysis examines the defaults reported in Chapter 5: ant population m, iteration budget $t_{\max}$, pheromone and heuristic exponents (α, β), evaporation rate ρ, initial pheromone τ_0, and the overlap threshold f_{overlap}. Where applicable we also monitor implicit bounds $(\tau_{\min}, \tau_{\max})$ that keep pheromone values numerically stable during updates.

6.6.2.2 Sensitivity observations

Using the Pretty Good Privacy and DBLP networks, we observed that scaling m between $0.05|V|$ and $0.2|V|$ modifies runtime linearly while shifting Q_{ov} by less than 1%, making $m = \lfloor 0.1|V| \rfloor$ a cost-effective default. Extending $t_{\max}$ to 300 iterations offered no measurable gain, whereas cutting it to 100 caused premature convergence on denser graphs, confirming the adequacy of $t_{\max} = 200$. Perturbing (α, β) within $(0.5, 1.5)$ and $(1.5, 2.5)$ respectively showed that higher β values sharpen exploitation but risk stagnation; $(1, 2)$ maintained the best balance. Evaporation in the range $\rho \in [0.05, 0.2]$ yielded overlapping modularity differences below 1.5%, while $\rho < 0.05$ caused pheromone saturation and loss of exploration. Setting τ_0 between $1/(2|E|)$ and $2/|E|$ preserved convergence speed; the nominal $1/|E|$ normalises deposits across densities. The overlap threshold f_{overlap} remained robust for values between 0.25 and 0.35, with 0.3 aligning best with manually validated overlaps. Throughout these sweeps the enforced bounds $\tau_{\min} = 10^{-4}$ and $\tau_{\max} = 1$ prevented numerical drift without influencing quality, supporting the statistical stability highlighted in Section 5.4.5.

6.7 Discussion

This section discusses the broader implications of the experimental findings, acknowledges limitations of the proposed approaches, and outlines directions for future research.

6.7.1 Key Findings and Validation

The experimental results underwent rigorous validation through multiple complementary approaches, including multiple executions for stochastic algorithms, comparative analysis against established baselines, and cross-validation across multiple metrics and network types. This comprehensive validation approach confirms that the reported performance improvements are reliable and representative of the algorithms' actual capabilities.

The results demonstrate that both HCCS and *ACO-CNoD* algorithms maintain consistent performance across different network types, with the NoD similarity measure providing improved discriminative capabilities compared to traditional measures. The complementary strengths of these algorithms highlight how different algorithmic paradigms can effectively leverage the same similarity measure for different application requirements, suggesting a modular approach to algorithm design that can be adapted to specific domain needs.

6.7.2 Current Limitations

The higher computational complexity of both algorithms compared to some baseline methods may limit their applicability to extremely large networks with millions of nodes, particularly in time-sensitive applications. While both algorithms demonstrate reasonable parameter sensitivity, their performance may still require careful parameter tuning for optimal results in specific application contexts. Additionally, the current implementations are designed for static networks and may not directly handle temporal dynamics, limiting their applicability to evolving network scenarios without modification.

6.8 Conclusion

This chapter has presented a comprehensive experimental evaluation of the three major contributions of this thesis: the NoD similarity measure, the HCCS algorithm for non-overlapping community detection, and the *ACO-CNoD* algorithm for overlapping community detection. The evaluation employed diverse real-world benchmark networks to assess performance across different network types and community detection scenarios.

The experimental results successfully addressed the four research questions posed at the beginning of the chapter, validating all three thesis contributions. The NoD similarity measure demonstrated improved discriminative capabilities compared to

traditional measures, particularly in networks with complex community structures. The HCCS algorithm showed competitive performance compared to state-of-the-art methods, with particularly strong results on networks with ground truth communities. The *ACO-CNoD* algorithm performed well in overlapping community detection scenarios, confirming the effectiveness of integrating ant colony optimization with the NoD similarity measure.

These findings have significant implications for community detection research and practice. The complementary strengths of HCCS and *ACO-CNoD* demonstrate that different algorithmic paradigms can effectively leverage the same similarity measure for different application requirements. The demonstrated performance makes these algorithms suitable for real-world applications across diverse domains, particularly where deterministic results (HCCS) or overlapping community detection (*ACO-CNoD*) are required.

Future research should focus on extending these algorithms to domain-specific networks (biological, transportation, and communication networks), adapting them to handle temporal networks with evolving community structures, and enhancing scalability for very large networks through parallel and distributed implementations. Additional promising directions include integration with interactive visualization frameworks to improve interpretability, and comparative evaluations against emerging approaches such as deep learning-based methods, multi-objective optimization techniques, and consensus clustering. These extensions would significantly enhance the practical utility of the proposed algorithms across diverse application domains.

Chapter 7

Conclusion

This thesis has addressed the challenging problem of community detection in complex networks, a task that remains central to network science due to applications spanning social systems, biology, transportation, cybersecurity, and information services. Communities, positioned between local interactions and global organisation, provide essential insight into how complex systems operate. Despite decades of research, major challenges persist, particularly the need for similarity measures that capture local density, algorithms that balance interpretability and accuracy, and techniques that accommodate overlapping memberships. The overall ambition of this work, stated in the thesis introduction, was therefore twofold: to design density-aware, centrality-sensitive methods for disjoint communities and to extend these principles to overlapping structures.

The investigation led to three interdependent contributions, each detailed in a dedicated chapter. First, Chapter 4 introduced the NoD (Neighbourhood Overlap with Density awareness) similarity, which couples shared-neighbour counts with local cohesion to mitigate the "density blindness" affecting classical measures such as Jaccard or cosine similarity. Second, the same chapter presented the HCCS algorithm, which combines degree-based leader identification with NoD-driven assignment to detect cohesive, disjoint communities while keeping computational costs competitive. Third, Chapter 5 proposed the *ACO-CNoD* metaheuristic, embedding NoD within an ant-colony system so that nodes can legitimately belong to several communities. These contributions form a consistent methodological path from pairwise similarity to disjoint detection and finally to overlapping detection.

The empirical campaign reported in Chapter 6 confirmed the relevance of this path. On benchmark networks with ground truth, HCCS achieved modularity scores up to 0.60 on the Football graph (Table 6.3), exceeding the reference partition (0.55), and reached NMI values of 0.89 on Dolphins and 0.90 on Football (Table 6.3). On larger graphs without ground truth, such as the 4,941-node Power network, HCCS

sustained competitive modularity (0.93), demonstrating scalability on heterogeneous topologies (Section 6.3.2). For overlapping detection, *ACO-CNoD* secured the best overlapping modularity Q_{ov} on five of seven datasets (Table 6.5); for example, on the Karate network, *ACO-CNoD* obtained $Q_{ov} = 0.72$ versus 0.51 for COPRA. Even on the sizable PGP graph, *ACO-CNoD* maintained $Q_{ov} = 0.68$, only 0.10 below the best baseline. These quantitative results showcase how NoD-driven heuristics deliver structural accuracy without sacrificing efficiency.

Beyond raw metrics, the thesis establishes conceptual bridges between centrality and similarity. NoD reframes neighbourhood overlap as density-aware evidence, while HCCS operationalises the idea that influential leaders (Chapter 4) can seed robust communities when coupled with density-sensitive assignments. *ACO-CNoD* extends this principle to search-based optimisation (Chapter 5), illustrating how pheromone dynamics leverage NoD to expose overlapping roles. This integration of structural importance and similarity-based proximity supplies a versatile framework that can be adapted to numerous analytical tasks, including link prediction or anomaly detection.

The proposed methods also have concrete implications. Practitioners analysing social graphs can exploit HCCS to extract disjoint groups with high modularity, supporting recommender systems or influence analysis. In biological networks, *ACO-CNoD* helps uncover overlapping functional modules, which often mirror biological reality. Transportation planners can apply these tools to reveal regional subnetworks for optimisation purposes, while cybersecurity analysts may track overlapping communities indicative of coordinated malicious behaviour. Because NoD, HCCS, and *ACO-CNoD* operate with manageable complexity (Sections 6.3.1 and 6.4.2), they remain usable when data volume is substantial.

Two main limitations temper these findings. First, the experimental datasets, although diverse (Table 6.1), primarily comprise social and collaboration graphs; performance on other network families (e.g., infrastructural or biological networks with different sparsity profiles) still requires validation. Second, both HCCS and *ACO-CNoD* involve parameters (leader quota, pheromone evaporation, exploration bias) whose sensitivity, discussed in Section 6.2, can influence results; rigorous automatic tuning and statistical robustness analysis remain future challenges. As a consequence, reproducibility currently depends on carefully reproducing the configurations documented in Chapter 6, using the published parameter grids and datasets. Releases of code or notebooks—when available—should emphasise these settings to facilitate consistent re-runs.

Addressing these limitations guides the perspectives of this work. Extending NoD, HCCS, and *ACO-CNoD* to dynamic networks would mitigate the dataset limitation by capturing temporal evolution in domains where community drift is expected (e.g., communication platforms). Pursuing parallel and distributed implementations will

answer the scalability needs highlighted by the PGP results, where *ACO-CNoD* trails COPRA on Q_{ov}. Combining NoD-driven heuristics with graph representation learning could improve robustness to parameter sensitivity by learning adaptive similarity weights rather than fixing them a priori. Finally, tailoring NoD to weighted or directed graphs will broaden applicability to transportation or biological systems where interaction strengths and directionality matter.

In conclusion, the thesis advances the state of the art by introducing a density-aware similarity measure, a centrality-guided disjoint detection algorithm, and a metaheuristic for overlapping detection, each validated quantitatively and conceptually. Together, NoD, HCCS, and *ACO-CNoD* constitute a coherent framework that reconciles structural influence and similarity, offering interpretable yet powerful tools for analysing complex networks. While further efforts are needed to enlarge dataset diversity and automate parameter handling, the contributions provide a solid foundation for future research at the intersection of graph theory, network science, and machine learning, and they equip practitioners with actionable techniques for dissecting the intricate fabric of modern networks.

Bibliography

[1] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[2] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

[3] Mark EJ Newman. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2010.

[4] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001. doi: 10.1103/PhysRevE.64.025102.

[5] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[6] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003. doi: 10.1016/S0378-8733(03)00009-1.

[7] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006. doi: 10.1109/MCI.2006.329691.

[8] Clara Pizzuti. Ga-net: A genetic algorithm for community detection in social networks. *Parallel Problem Solving from Nature–PPSN X*, pages 1081–1090, 2008. doi: 10.1007/978-3-540-87700-4_107.

[9] Ping Zhang and Gary Chartrand. *Introduction to graph theory*, volume 2. Tata McGraw-Hill New York, 2006.

[10] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[11] Linyuan L"u and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011. doi: 10.1016/j.physa.2010.11.027.

[12] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006. doi: 10.1073/pnas.0601602103.

[13] Mark S Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[14] Mikko Kivel
, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014. doi: 10.1093/comnet/cnu016.

[15] John Scott. *Social Network Analysis*. SAGE Publications, London, UK, 4th edition, 2017.

[16] Rob Cross and Andrew Parker. *The hidden power of social networks: Understanding how work really gets done in organizations*. Harvard Business Review Press, Boston, MA, 2002. ISBN 9781578518210.

[17] Matt J Keeling and Pejman Rohani. *Networks of infection: disease transmission in human and animal populations*. Oxford University Press, Oxford, UK, 2005. ISBN 9780198566946.

[18] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003. doi: 10.1145/956750.956769.

[19] Martin Grandjean. Introduction to network visualization with gephi. *Programming Historian*, 5, 2016. URL https://programminghistorian.org/en/lessons/network-analysis-with-gephi.

[20] Jacob L Moreno. Who shall survive? a new approach to the problem of human interrelations. *Nervous and Mental Disease Publishing Co.*, 1934.

[21] Frank Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

[22] John A Barnes. Class and committees in a norwegian island parish. *Human Relations*, 7(1):39–58, 1954.

[23] Elizabeth Bott. *Family and social network: roles, norms and external relationships in ordinary urban families*. Tavistock Publications, London, 1957.

[24] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12): 7821–7826, 2002. doi: 10.1073/pnas.122653799.

[25] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004. doi: 10.1103/PhysRevE.69.066133.

[26] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004. doi: 10.1103/PhysRevE.69.026113.

[27] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007. doi: 10.1073/pnas.0605965104.

[28] Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81 (4):046106, 2010. doi: 10.1103/PhysRevE.81.046106.

[29] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5): 75–174, 2010. doi: 10.1016/j.physrep.2009.11.002.

[30] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016. doi: 10.1016/j.physrep.2016.09.002.

[31] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. doi: 10.1088/1742-5468/2008/10/P10008.

[32] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

[33] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Deep learning for community detection: Progress, challenges and opportunities. *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 4981–4987, 2020. doi: 10.24963/ijcai.2020/693.

[34] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. doi: 10.1145/2939672.2939754.

[35] Barış Can Bayan and Cuneyt G Ozturk. The bayan algorithm: Detecting communities in networks through exact and approximate optimization of modularity. *arXiv preprint arXiv:2209.04562*, 2022. URL https://arxiv.org/abs/2209.04562.

[36] Mohamad Badra Jdiab and Imad H Elhajj. A survey on metaheuristics for community detection in networks. *Applied Soft Computing*, 71:850–903, 2018. doi: 10.1016/j.asoc.2018.07.044.

[37] Si Zhang and Lu Chen. A comprehensive review of community detection in graphs. *arXiv preprint arXiv:2309.11798*, 2023. URL https://arxiv.org/abs/2309.11798.

[38] Laxman Dhulipala, Guy E Blelloch, and Julian Shun. Terahac: Hierarchical agglomerative clustering of trillion-edge graphs. *arXiv preprint arXiv:2308.03578*, 2023. URL https://arxiv.org/abs/2308.03578.

[39] Abdullah Alghamdi, Turki Alsubait, Omar Cheikhrouhou, Najla Alsaedi, and Reem Alghamdi. Influence-based community partition with degreerank label propagation (drlp) algorithm for social networks. *Applied Sciences*, 14(8):4295, 2024. doi: 10.3390/app14084295.

[40] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. doi: 10.1038/nature03607.

[41] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4):1–35, 2013. doi: 10.1145/2501654.2501657.

[42] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010. doi: 10.1038/nature09182.

[43] Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 344–349, 2011. doi: 10.1109/ICDMW.2011.154.

[44] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and Albert-László Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002. doi: 10.1126/science.1073374.

[45] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009. doi: 10.1088/1367-2630/11/3/033015.

[46] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. *Data clustering: a review*, volume 31. ACM, 1999.

[47] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley Sons, New York, 1990.

[48] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004. doi: 10.1103/PhysRevE.70.066111.

[49] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks. *Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276, 2007. doi: 10.1145/1242572.1242805.

[50] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 2006. doi: 10.1103/PhysRevE.74.016110.

[51] Alex Arenas, Alberto Fernández, and Sergio Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9(6):176, 2007. doi: 10.1088/1367-2630/9/6/176.

[52] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2:336, 2012. doi: 10.1038/srep00336.

[53] Clara Granell, Saeed Darabi, Alex Arenas, Sergio Gómez, and Peter J Mucha. Hierarchical multiresolution method to overcome the resolution limit in complex networks. *International Journal of Bifurcation and Chaos*, 25(03):1540004, 2015. doi: 10.1142/S0218127415400040.

[54] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001. doi: 10.1080/0022250X.2001.9990249.

[55] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663, 2004. doi: 10.1073/pnas.0400054101.

[56] Haijun Zhou. Distance, dissimilarity index, and network community structure. *Physical Review E*, 67(6):061901, 2003. doi: 10.1103/PhysRevE.67.061901.

[57] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008. doi: 10.1073/pnas.0706851105.

[58] Martin Rosvall and Carl T Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009. doi: 10.1140/epjst/e2010-01179-1.

[59] Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, 2008. doi: 10.1016/j.socnet.2007.11.001.

[60] Aaron F McDaid, Derek Greene, and Neil Hurley. Improved mutual information measure for clustering, classification and community detection. *Journal of Machine Learning Research*, 12:2473–2490, 2011.

[61] Andreas Noack and Randolf Rotta. Multi-level algorithms for modularity clustering. *Experimental Algorithms*, pages 257–268, 2009. doi: 10.1007/978-3-642-02011-7_24.

[62] Hugues Zanghi, Nathalie Villa-Vialaneix, Stéphane Volant, Silvia Bacci, Marco Corneli, and Charles Bouveyron. Fast online graph clustering via erdøs–rényi mixture. *Pattern Recognition*, 41(12):3592–3599, 2008. doi: 10.1016/j.patcog.2008.06.019.

[63] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, 2005. doi: 10.1103/PhysRevE.72.027104.

[64] Roger Guimer

s and Luís A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005. doi: 10.1038/nature03288.

[65] Andrés Medus, Gonzalo Acuña, and Claudio O Dorso. Detection of community structures in networks via global optimization. *Physica A: Statistical Mechanics and its Applications*, 358(2-4):593–604, 2005. doi: 10.1016/j.physa.2005.04.022.

[66] Maoguo Gong, Bin Fu, Licheng Jiao, and Haifeng Du. Memetic algorithm for community detection in networks. *Physical Review E*, 84(5):056101, 2011. doi: 10.1103/PhysRevE.84.056101.

[67] Stefan Boettcher and Allon G Percus. Optimization with extremal dynamics. *Physical Review Letters*, 98(11):118701, 2007.

[68] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

[69] Michael T Schaub, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. The many facets of community detection in complex networks. *Applied Network Science*, 2(1):4, 2017. doi: 10.1007/s41109-017-0023-6.

[70] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90, 2014. doi: 10.1109/TNSE.2015.2391998.

[71] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[72] Clara Pizzuti. A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation*, 16(3):418–430, 2012. doi: 10.1109/TEVC.2011.2161090.

[73] Jyoti Choudur Devi and E Poovammal. Community detection in complex networks using genetic algorithm. *International Journal of Computer Applications*, 137(1):1–9, 2016.

[74] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856. MIT Press, 2002.

[75] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990. doi: 10.1137/0611030.

[76] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, jul 1992.

[77] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical*

review E, 76(3):036106, 2007. doi: 10.1103/PhysRevE.76.036106.

[78] Huan Li, Ruisheng Zhang, Zhili Zhao, and Xin Liu. Lpa-mni: an improved label propagation algorithm based on modularity and node importance for community detection. *Entropy*, 23(5):497, 2021.

[79] Michael J Barber and John W Clark. Detecting network communities by propagating labels under constraints. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(2):026129, 2009.

[80] Xin Liu and Tsuyoshi Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.

[81] Zhenyu Wu, Shijie Pan, Feng Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. Lpa-hm: Label propagation algorithm for high-mixed networks. *Neurocomputing*, 452:165–178, 2021. doi: 10.1016/j.neucom.2021.03.053.

[82] Jierui Xie and Boleslaw K Szymanski. Labelrank: A stabilized label propagation algorithm for community detection in networks. In *2011 IEEE 2nd International Conference on Social Computing*, pages 249–256, 2011. doi: 10.1109/PASSAT/SocialCom.2011.34.

[83] Rong Yan, Wei Yuan, Xiangdong Su, and Ziyi Zhang. Flpa: a fast label propagation algorithm for detecting overlapping community structure. *Expert Systems with Applications*, 234:120971, 2023. doi: 10.1016/j.eswa.2023.120971.

[84] Guiqiong Xu, Jiawen Guo, and Pingle Yang. Tns-lpa: an improved label propagation algorithm for community detection based on two-level neighbourhood similarity. *IEEE Access*, 9:23526–23536, 2020. doi: 10.1109/ACCESS.2020.3044625.

[85] Chuanpeng Wang, Zhuanming Gao, and Dong Li. Lda-lpa: a community detection method based on topic model. In *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, pages 726–731. IEEE, 2021.

[86] Maryam Hosseini-Pozveh, Maedeh Ghorbanian, and Maryam Tabaiyan. A label propagation-based method for community detection in directed signed social networks. *Physica A: Statistical Mechanics and its Applications*, 604:127875, 2022.

[87] Imen Ben El Kouni, Wafa Karoui, and Lotfi Ben Romdhane. Node importance based label propagation algorithm for overlapping community detection in networks. *Expert Systems with Applications*, 162:113020, 2020.

[88] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010. doi: 10.1088/1367-2630/12/10/103018.

[89] Dongming Chen, Panpan Du, Qianrong Jiang, Xinyu Huang, and Dongqi Wang. Msh-lpa: A novel community detection algorithm based on the sh-index and label propagation. *Symmetry*, 12(2):223, 2020. doi: 10.3390/sym12020223.

[90] Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005. doi: 10.1016/j.socnet.2004.11.009.

[91] Ernesto Estrada, Desmond J Higham, and Naomichi Hatano. Communicability betweenness in complex networks. *Physica A: Statistical Mechanics and its Applications*, 388(5):764–774, 2009. doi: 10.1016/j.physa.2008.11.011.

[92] Albert-László Barabási. *Network Science*. Cambridge University Press, Cambridge, UK, 2016. ISBN 9781107076266.

[93] Linton C Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978. doi: 10.1016/0378-8733(78)90021-7.

[94] Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006. doi: 10.1016/j.socnet.2005.11.005.

[95] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007. doi: 10.1002/asi.20591.

[96] Tao Zhou, Linyuan L"u, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009. doi: 10.1140/epjb/e2009-00335-8.

[97] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi: 10.1126/science.286.5439.509.

[98] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953. doi: 10.1007/BF02289026.

[99] Glen Jeh and Jennifer Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002. doi: 10.1145/775047.775126.

[100] Clara Granell, Saeed Darabi, Alex Arenas, Sergio Gómez, and Peter J Mucha. Benchmarking measures of network controllability on canonical graph models. *Journal of Nonlinear Science*, 25(6):1311–1337, 2015. doi: 10.1007/s00332-015-9258-5.

[101] Tiago P Peixoto. Parsimonious module inference in large networks. *Physical Review Letters*, 110(14):148701, 2013. doi: 10.1103/PhysRevLett.110.148701.

[102] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005. doi: 10.1088/1742-5468/2005/09/P09008.

[103] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

[104] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. doi: 10.1007/s10115-013-0693-z.

[105] Leto Peel, Daniel B Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5): e1602548, 2017.

[106] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[107] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4): 046110, 2008. doi: 10.1103/PhysRevE.78.046110.

[108] Scott Emmons, Stephen Kobourov, Mike Gallant, and Katy B"orner. Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one*, 11(7):e0159161, 2016. doi: 10.1371/journal.pone.0159161.

[109] François Lorrain and Harrison C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1(1):49–80, 1971. doi: https://doi.org/10.1080/0022250X.1971.9989788.

[110] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: An autocatalytic optimizing process. *Technical report*, 1991.

[111] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996. doi: 10.1109/3477.484436.

[112] Luca M Gambardella, Éric D Taillard, and Giovanni Agazzi. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2):167–176, 1999.

[113] Thomas St"utzle and Holger H Hoos. Max-min ant system. *Future generation computer systems*, 16(8):889–914, 2000. doi: 10.1016/S0167-739X(00)00043-1.

[114] M. Dorigo, G. A. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.

[115] M. Perretto and H. S. Lopes. Reconstruction of phylogenetic trees using the ant colony optimization paradigm. *Genetics and Molecular Research*, 4(3): 581–589, 2005.

[116] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997. doi: 10.1109/4235.585892.

[117] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[118] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003. doi: 10.1007/s00265-003-0651-y.

[119] Donald E Knuth. The stanford graphbase: a platform for combinatorial computing. 1993.

[120] Valdis Krebs. Books about us politics, 2004. URL http://www.orgnet.com/. Unpublished, available at http://www.orgnet.com/.

[121] Pablo M Gleiser and Leon Danon. Community structure in jazz. *Advances in Complex Systems*, 6(04):565–573, 2003. doi: 10.1142/S0219525903001067.

[122] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog, 2005.

[123] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. doi: 10.1038/30918.

[124] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007. doi: 10.1145/1217299.1217301. URL https://doi.org/10.1145/1217299.1217301.

[125] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL https://networkrepository.com.

[126] Ali Chenaoui and Mohammed Amin Tahraoui. An iterated heuristic community detection algorithm for social networks based on centrality and similarity measures. *Engineering, Technology Applied Science Research*, 15(4):24211–24219, August 2025. doi: 10.48084/etasr.10791. URL https://doi.org/10.48084/etasr.10791.