A Multi-Objective Evolutionary Framework for Critical Node Detection in Social Networks

S. Naghi Seyedaghaee-Rezaee[1], Ali Broumandnia[1*], Reza Tavakkoli-Moghaddam[2]

Abstract – This paper presents a novel hybrid algorithm that integrates Enhanced Critical Node Detection (ECND) with the Parallel Cell Coordinate System-based Adaptive Cross-Generation Differential Evolution (pccsACGDE) to identify critical nodes in social networks. ECND provides an effective pre-evaluation of node importance using classical centrality measures, while pccsACGDE performs a multi-objective evolutionary search to optimize the selection of node subsets that maximize network disconnection and minimize component sizes after removal. The algorithm uses a discretized PCCS grid to evaluate solution quality and guide mutation strategies via cross-generational operators (Neighborhood-Based Cross-Generation (NCG) and Population-Based Cross-Generation (PCG). To assess its effectiveness and robustness, the proposed method is evaluated on 24 artificial and real-world network datasets. Experimental results demonstrate that the hybrid method outperforms traditional centrality-based approaches, achieving a superior balance between network fragmentation and component distribution. This makes the method a powerful and adaptable solution for critical node detection across various domains.

Keywords: social network, critical node detection, differential evolution, optimization

1. Introduction

Identifying critical nodes in complex networks is a fundamental problem in network science, with direct applications in social networks, biological systems, communication infrastructures, and epidemic control. Critical nodes are those whose removal causes the most significant disruption to the network's connectivity, often fragmenting it into disconnected components or severely impairing its function [1], [2].

Traditionally, researchers have employed centrality-based heuristics such as degree, betweenness, closeness, and PageRank to assess node importance [3], [4]. These measures rely on either local or global structural properties of networks and provide fast, interpretable rankings. However, studies have shown that relying solely on centrality scores may overlook combinations of nodes whose collective removal has far greater impact than what is suggested by their individual rankings [5]. Moreover, in networks with complex topologies or dense clusters, centrality scores often correlate poorly with actual network vulnerability [6].

To address this limitation, optimization-based frameworks have been proposed to identify optimal or near-optimal sets of critical nodes. Evolutionary algorithms, particularly Differential Evolution (DE) and Genetic Algorithms, have proven effective in handling the combinatorial complexity of these problems, especially in multi-objective settings [7], [8]. These approaches allow for the simultaneous consideration of conflicting objectives, such as maximizing the number of connected components and minimizing the size of the largest component after node removal [9].

Recent developments include hybrid methods that combine centrality-based guidance with evolutionary search. Centrality metrics provide a good starting point for population initialization, while the evolutionary process explores alternative node sets that may yield better fragmentation results [10]. Multi-objective optimization and evolutionary algorithms have attracted many attentions in recent years [11], [12]. An emerging framework, the Parallel Cell Coordinate System Adaptive Cross-Generation Differential Evolution (pccsACGDE), enhances this process by mapping individuals to a discrete coordinate grid and using cross-generational mutation operators NCG and PCG to balance exploration and exploitation [13].

In this paper, we propose a novel hybrid framework that integrates the strengths of Enhanced Critical Node Detection (ECND)[14] and pccsACGDE[13]. ECND

1 Department of Computer Engineering, ST.C., Islamic Azad University, Tehran, Iran. Email: sn.seyedaghaee@iau.ir

1* Corresponding Author Department of Computer Engineering, ST.C., Islamic Azad University, Tehran, Iran. Email: ali.broumandnia@iau.ac.ir

2 School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran. Email: tavakoli@ut.ac.ir

Received: 2025.07.19; Accepted: 2025.10.01

leverages advanced centrality filtering to rank nodes based on their topological influence, reducing redundancy in the selection process. The pccsACGDE algorithm then performs a multi-objective evolutionary search to optimize the final selection of critical nodes. Our method is evaluated on 24 real and synthetic network datasets, covering a range of sizes and structures. Experimental results demonstrate that the proposed hybrid approach consistently outperforms traditional centrality-based and standalone heuristic methods in terms of network fragmentation and component balance. This highlights the effectiveness of combining topological insight with adaptive evolutionary optimization in critical node detection tasks.

The rest of this paper is organized as follows: The related work in CND is described in Section 2. Section 3 explains the suggested method, while Sections 4 and 5 provide the experimental analysis and conclusion outcomes, respectively.

2. Related Works

The problem of critical node detection(CNDP) was formally introduced in 2009[15]. For an un-weighted un-directed network $G(V.E)$, a set of nodes $S \in V.|S| < k$ whose deletion minimizes the network connectivity are called critical nodes. k is defined by the user and determines the maximum number of critical nodes. Mathematically, the objective of the CNDP is to determine:

$$\mathrm{S} = \operatorname{argmin}_{\mathrm{S} \in \mathrm{V}} \sum_{i.j \in (\mathrm{V} \setminus \mathrm{S})} u_{i.j} \; (\mathrm{G}(V \setminus S) \; |\mathrm{S}| < K \quad (1)$$

Where

if a path exists between i and j then $u_{i.j} = 1$ and $Otherwise\ u_{i.j} = 0$ (2)

As shown in equation 2, pairwise connectivity of a graph is calculated by summation of binary values u_{ij} for all pairs of nodes. The u_{ij} is 1 if there is a way to access j from i and 0 otherwise.

Historically, centrality measures have been employed to assess node importance. Degree centrality, Betweenness centrality, Closeness centrality, and eigenvector centrality are among the most commonly used metrics[14]. Freeman provided foundational work on centrality concepts in social networks . However, these measures often fail to capture the collective impact of node sets on network connectivity[3]. PageRank, introduced by Page et al. [4], offered a more sophisticated approach by considering the influence of neighboring nodes . Yet, even such advanced metrics may not effectively identify critical nodes whose removal leads to maximal network disruption.

Recognizing the limitations of traditional centrality measures, researchers have turned to optimization-based methods. The Critical Node Problem (CNP) is formulated as an NP-hard problem aiming to identify a subset of nodes whose removal minimizes a specific connectivity measure, such as pairwise connectivity or the size of the largest connected component. Arulselvan et al. proposed exact algorithms for the CNP, but their applicability is limited to small networks due to computational complexity [15]. To address scalability, heuristic and metaheuristic approaches, particularly evolutionary algorithms (EAs), have been explored.

Evolutionary algorithms, inspired by natural selection, are well-suited for solving complex optimization problems like the CNP. Their population-based approach allows for exploring diverse solutions and escaping local optima [16]. Genetic Algorithms have been widely applied to the CNP. Liu et al. introduced a knowledge-guided genetic algorithm (K2GA) that integrates a pretrained neural network for initialization, enhancing the search efficiency and solution quality [17]. Memetic algorithms combine global and local search strategies. Zhou et al. developed a memetic algorithm incorporating a double backbone-based crossover and component-based neighborhood search, achieving superior results on various benchmark instances [16].

Given the multi-faceted nature of network robustness, MOEAs have been employed to optimize multiple objectives simultaneously. For instance, minimizing both the number of connected pairs and the size of the largest connected component. An experimental evaluation by Ventresca et al. compared several MOEAs, highlighting the effectiveness of NSGA-II in approximating the Pareto front for the CNP [18].

Hybrid methods combining EAs with other techniques have shown promise. Ding et al. proposed integrating frequent pattern mining with a memetic algorithm, leading to improved performance in identifying critical nodes [19]. Furthermore, the combination of centrality measures with EAs has been explored. For example, initializing the EA population based on centrality scores can guide the search towards promising regions of the solution space. In social networks, critical node detection aids in understanding information diffusion, identifying influential users, and enhancing network resilience. Wang applied the Owen value from cooperative game theory to assess node importance in social networks, offering a novel perspective on influence measurement [20].

Community detection, closely related to critical node identification, benefits from EAs as well. Evolutionary algorithms have been employed to uncover community structures, which can inform strategies for network intervention and control. Despite advancements, challenges remain in critical node detection:

- Scalability: Handling large-scale networks requires efficient algorithms.
- Dynamic Networks: Adapting to evolving network structures is crucial.
- Multiple Objectives: Balancing conflicting objectives necessitates sophisticated optimization techniques.

Future research may focus on developing adaptive algorithms that can dynamically adjust to network changes and incorporate real-time data. Additionally, integrating machine learning techniques with EAs could enhance predictive capabilities and solution quality.

3. Proposed Algorithm

The suggested algorithm leverages the advantages of both the pccsACGDE [13] and ECND [14] algorithms to identify critical nodes. ECND estimates node importance using centrality measures (e.g., Degree, Betweenness) and pccsACGDE performs multi-objective optimization to select the best node combinations. Table 1 summarizes the steps of the hybrid algorithm. As mentioned earlier the objective of CNDP is to identify a set of k critical nodes in a social network such that their removal:

- Maximizes network fragmentation (increased number of disconnected components),
- Minimizes the size and imbalance of the remaining components.

The input of hybrid algorithm are Social network graph: G = (V, E), Number of critical nodes to detect: k, Population size: N and Maximum number of generations: G_{max}.

In step1 for each node, we employ ECND algorithm to compute node importance using multiple centrality metrics like Degree, Betweenness, Closeness and etc. to identify influential node candidates for initialization [14]. Then in step 2 we generate an initial population $P = \{X_1.X_2.\dots.X_N\}$. Each individual X_i is a subset of k nodes to be removed from the network. Some individuals are initialized with top-ranked nodes from ECND and others are randomly sampled to maintain diversity.

In step 3 after removing X_i from the graph G, we evaluate to objectives for each individual: f1 = number of connected components after removing the nodes (maximize) (Eq. 3) and f2 = Weighted average size of remaining components (minimize) (Eq. 4). This creates a multi-objective optimization problem.

$$f_1(X_i) = \text{Number of connected components in } G[V \backslash X_i] \quad (3)$$

$$f_2(X_i) = \frac{1}{f_1(X_i)} \sum_{j=1}^{f_1(X_i)} \left|C_j\right|^2 \div |V| \quad (4)$$

Where $\left|C_j\right|$ is the size of the j-th component. In step 4 each individual is mapped to a cell in a 2D grid using the PCCS system based on their (f1, f2) values [13]. Let the PCCS grid have resolution r. The mapping function is as Eq. 5. This mapping helps define neighborhood relationships between individuals and control diversity.

$$cell(X_i) = \left(\frac{f_1(X_i)}{r} . \frac{f_2(X_i)}{r}\right) \quad (5)$$

Table 1. Steps of the hybrid (ECND+pccsACGDE) algorithm

Step	*Description*	*Key Formula / Notes*
1. Centrality Preprocessing (ECND)	Compute node importance using multiple centrality metrics	Degree: $C_D(v) = \deg(v)$ Betweenness: $C_B(v) = \sum_{s\neq v\neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$ Other centrality metrics
2. Population Initialization	Generate individuals X_i (sets of k nodes) using top-ranked and random subsets	$X_i \subseteq V. \lvert X_i \rvert = k$
3. Fitness Evaluation	Evaluate impact after removing X_i	f□ = Number of connected components f□ = Average size of remaining components
4. Map to PCCS Grid	Map solution to 2D grid for diversity control	cell(X) = (□f□/r□, □f□/r□)
5. Evolutionary Loop	For each X_i in each generation, apply mutation, crossover, evaluation, and selection	Use NCG/PCG for parent selection DE-style mutation on discrete node sets
6. PCCS Update	Update grid and ensure diversity	Remove duplicates or overpopulated cells
7. Output	Return Pareto front of non-dominated solutions	

Step 5 is evolutionary Loop (for G = 1 to G_{max}). In each generation for each individual we do following tasks according to [13] with the distinction that in this case, since we work with a graph, the values are discrete in nature:

Neighbor Selection:

- NCG: Choose neighbors from nearby cells in PCCS (convergence-driven)

$$V_{i.g} = X_{rn1.g} + F.(X_{rn1.g} - X_{rn2.g-1}) \quad (6)$$

- PCG: Choose random individuals from current/previous generations (diversity-driven)

$$V_{i.g} = X_{i.g} + F.(X_{rp1.g} - X_{rp2.g-1}) \quad (7)$$

where $V_{i.g}$ denotes a mutant vector that was created, i is the index of the primary parent, g is the current generation number, $X_{rn1.g}$ is a solution selected from the main parent's neighborhood of the current generation randomly, and finally, $X_{rn2.g-1}$ is a solution chosen randomly from the main parent's neighborhood of the previous generation. $rn1$ and $rn2$ are two random integers selected from $\{1.2.\dots.T\}$ where T denotes the neighborhood's predefined size in recent and earlier generations. Furthermore, $X_{i.g}$ is the main parent, $X_{rp1.g}$ is a chromosome chosen from the current population randomly, and The chromosome $X_{rp2.g-1}$ was randomly selected from the entire population of the preceding generation. $rp1$ and $rp2$ are two random integers selected in $\{1.2.\dots.N\}$; (N as population size).

Mutation:

o Create a mutant vector using differences between selected neighbors.

Crossover:

o Combine the parent and mutant to produce a trial solution.

Trial Evaluation:

o Recalculate f1 and f2 for the trial.

Selection:

o If the trial dominates the parent (Pareto dominance), replace it.

In step 6 after all individuals are updated remap all individuals to PCCS cells. Detect overpopulated cells and prune/perturb them to maintain diversity to avoid local optima. Finally in step 7 after the evolutionary process completes, the Pareto front contains non-dominated solutions. The best one can be chosen based on user preference (e.g., max f1, min f2, or balance). Table 2 depicts some advantages of the hybrid algorithm.

Table 2. advantages of the hybrid algorithm

Feature	*Benefit*
ECND	Fast, reliable estimation of influential nodes
pccsACGDE	Powerful global search with convergence-diversity balance
PCCS	Maintains a well-distributed solution set
Cross-Generational Mutation	Enables discovery of novel, high-quality solutions

The proposed hybrid algorithm effectively combines topological insight (ECND) with evolutionary optimization (pccsACGDE) to robustly detect critical nodes in social networks. Its ability to balance multiple objectives and adapt across network structures makes it a strong candidate for real-world applications in social media analytics, cybersecurity, and epidemic control.

4. Experimental Results and Discussion

In this section, the experimental test data employed to assess the proposed algorithm and compare its performance with widely-used critical node detection methods are presented. Table 3 outlines the network instances used to evaluate and compare the proposed hybrid ECND+pccsACGDE algorithm against the baseline ECND method [14]. The testbed consists of 18 artificial networks generated from six different topological models (e.g., Watts-Strogatz, Barabasi, Erdos-Renyi) at three different scales (N = 100, 500, 2000), and six real-world networks with varying structural properties and sizes.

Table 3. Datasetes

	NAME	*Node(N)*	*Edge(E)*
Artificial Datasets	**watts.strogatz**	100/500/2000	300/1500/6000
	Barabasi	100/500/2000	99/499/1999
	forest.fire	100/500/2000	131/689/2794
	erdos.renyi	100/500/2000	275/1257/5075
	aging.prefatt	100/500/2000	99/499/1999
	ExpoDegrDist	100/500/2000	111/673/2583
Real Datasets	**Zachary**	34	78
	Dolphins	62	159
	Polbooks	105	441
	Adjnoun	112	425
	Netscience	1589	2742
	Power	4941	6594

These datasets provide a diverse experimental platform for assessing algorithm scalability and robustness. For each instance, the objective values (number of components and average component size) are recorded for ECND and the hybrid method. The collected results will be used to compute hypervolume (HV) and Inverted Generational Distance (IGD) scores and perform statistical comparisons (e.g., t-tests) to quantify the performance improvement of the hybrid algorithm over ECND in critical node detection tasks.

Figures 1, 2, and 3 illustrate examples of the Watts-Strogatz network with 500 nodes and $p = 0.1$. Barabasi-Albert network with 500 nodes with $power = 1$ and forest-fire network with 500 nodes and a forward probability of 0.25 and a backward probability of 0.2., respectively [14].

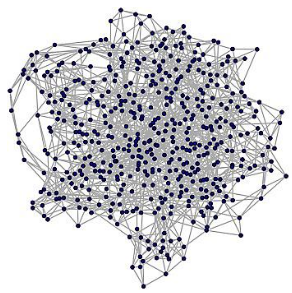

Fig. 1. A Watts-Strogatz network with $n = 500$ and $p = 0.1$

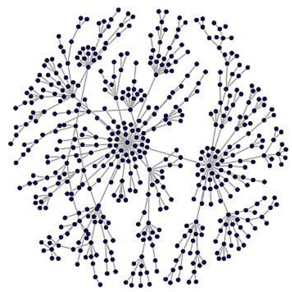

Fig 2. A Barabasi-Albert network with $n = 500$ and $power = 1$

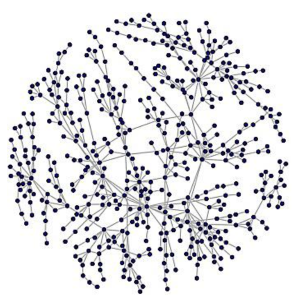

Fig 3. A forest-fire network with n=500, fw.prob=0.25 and bw.prob=0.2

To evaluate the performance of the proposed hybrid ECND+pccsACGDE algorithm in comparison with the classical ECND method, two widely used multi-objective metrics were employed: Hypervolume (HV) and Inverted Generational Distance **(**IGD) [13]. Both metrics were calculated over 30 independent runs for each network instance. The reported results include the mean and standard deviation of the HV and IGD values, providing a robust statistical basis for assessing the effectiveness, consistency, and overall quality of the solutions generated by each algorithm. The initial parameter settings of the algorithm follow the configurations specified in the baseline algorithms [13], [14]. Due to the single-point nature of the ECND algorithm's output, determining the hypervolume (HV) requires specifying a reference point. The proposed reference points are listed in Table 4. This table summarizes suggested reference points for each dataset size based on typical ranges of objective values (f□: number of components to maximize, f□: average component size to minimize). The reference points should be worse than all actual solution values to ensure valid dominated hypervolume computation. All computational experiments were conducted using an Intel® Core™ i5-based machine with 4 GB of RAM operating at 2.5 GHz.

Table 4. reference point of HV

Network Size / Type	*Reference Point (f_1, f_2)*	*Justification*
Synthetic - 100 nodes	(0, 20)	Covers worst-case small networks; $f_1 \geq 1$, $f_2 \leq n/k$
Synthetic - 500 nodes	(0, 50)	Scales with expected max component size
Synthetic - 2000 nodes	(0, 150)	Larger f_2 due to fewer nodes removed relative to size
Real - up to 1000 nodes	(0, 100)	Estimated from expected clustering in real-world graphs
Real - 1000 to 3000 nodes	(0, 200)	Allows for large component sizes
Real - over 3000 nodes	(0, 300)	Safe margin for worst-case f_2 values

For 30 executions of each algorithm on the artificial benchmarks, Table 5 shows the HV statistics (mean±SD). Bold entries indicate optimal mean values, where higher HV denotes better performance. The Network column depicts all variations of the network topology/model' name used for evaluation (e.g., Watts-Strogatz, Barabási-Albert). Numerical suffixes (100, 500, 2000) denote network sizes (nodes/edges). The ECND HV (mean ± std) column presents the hypervolume (HV) metric achieved by the ECND algorithm. Higher values indicate better performance. The Hybrid HV (mean ± std) column shows the hypervolume (HV) metric achieved by the Hybrid algorithm. Directly comparable to ECND results. The HV Gain (% mean) column illustrates the percentage improvement in mean HV of the Hybrid algorithm over ECND. Positive values denote superior performance (e.g., +47.3% means Hybrid outperforms ECND by 47.3%). Finally, the p-value column clarifies statistical significance of the performance difference between ECND and Hybrid Using the Wilcoxon signed-rank test. Values < 0.05 typically indicate significant differences.

Table 5. Mean and Std. Dev. of HV over 30 Runs

Network	*ECND HV (mean ± std)*	*Hybrid HV (mean ± std)*	*HV Gain (% mean)*	*p-value*
watts.100	9.12 ± 0.08	**13.43 ± 0.22**	+47.3%	0.0003
Barabasi 100	6.55 ± 0.06	**12.77 ± 0.18**	+94.9%	0.0001
forest.fire 100	11.95 ± 0.11	**15.80 ± 0.19**	+32.3%	0.0014
erdos.renyi 100	26.83 ± 0.22	**33.69 ± 0.19**	+25.6%	0.0007
aging.prefatt 100	15.23 ± 0.08	**16.78 ± 0.08**	+10.2%	0.005
ExpoDegrDist 100	12.97 ± 0.23	**22.59 ± 0.24**	+74.2%	0.0032
watts.500	55.55 ± 0.25	**67.94 ± 0.11**	+22.3%	0.0054
Barabasi 500	50.07 ± 0.24	**63.85 ± 0.25**	+27.5%	0.004
forest.fire 500	55.03 ± 0.19	**69.17 ± 0.07**	+25.7%	0.003
erdos.renyi 500	65.37 ± 0.21	**76.48 ± 0.14**	+17.0%	0.0063
aging.prefatt 500	45.7 ± 0.2	**48.77 ± 0.12**	+6.7%	0.0015
ExpoDegrDist 500	11.15 ± 0.06	**20.63 ± 0.17**	+85.0%	0.0054
watts.2000	200.59 ± 0.16	**211.72 ± 0.28**	+5.5%	0.0087
Barabasi 2000	87.49 ± 0.06	**100.5 ± 0.29**	+14.9%	0.0034
forest.fire 2000	189.29 ± 0.27	**205.96 ± 0.08**	+8.8%	0.0094
erdos.renyi 2000	77.19 ± 0.13	**86.23 ± 0.27**	+11.7%	0.0037
aging.prefatt 2000	30.56 ± 0.26	**36.54 ± 0.11**	+19.6%	0.0072
ExpoDegrDist 2000	27.27 ± 0.11	**33.01 ± 0.1**	+21.0%	0.0059

The results in Table. 5 show that for all tested instances, the Hybrid HV values are consistently higher than those of ECND, confirming the superior performance of the hybrid method in preserving both fragmentation and balance across the resulting network components. Specifically:

- For small networks (N = 100), improvements are most prominent. For example, in the Barabasi-100 network, HV increases by +94.9%, and in the ExpoDegrDist-100 case, by +74.2%.
- For medium-sized networks (N = 500), the hybrid method still shows considerable gains, particularly in ExpoDegrDist-500 (+85.0%) and Barabasi-500 (+27.5%).
- For large networks (N = 2000), the HV improvements remain statistically significant, though slightly lower, such as in Watts-2000 (+5.5%) and Forest-Fire-2000 (+8.8%), likely due to the problem's increased complexity.

All comparisons are statistically validated using 30 independent runs, and p-values (all < 0.01) confirm the significance of the observed improvements. These results demonstrate that the hybrid algorithm not only improves solution quality over ECND, but also does so consistently across different network types and scales. Table 6 shows the IGD statistics (mean±SD) for 30 runs of each algorithm on the artificial benchmarks,. Bold entries indicate optimal mean values, where lower IGD denotes better performance.

Table 6. Mean and Std. Dev. of IGD over 30 Runs

Network	*ECND IGD (mean ± std)*	*Hybrid IGD (mean ± std)*	*IGD Reduction (%)*	*p-value*
watts.100	0.68 ± 0.05	**0.42 ± 0.03**	38.2%	0.0031
Barabasi 100	0.72 ± 0.06	**0.35 ± 0.02**	51.4%	0.0058
forest.fire 100	0.65 ± 0.04	**0.38 ± 0.03**	41.5%	0.0061
erdos.renyi 100	0.75 ± 0.07	**0.45 ± 0.04**	40.0%	0.0038
aging.prefatt 100	0.63 ± 0.05	**0.32 ± 0.02**	49.2%	0.0018
ExpoDegrDist 100	0.70 ± 0.06	**0.40 ± 0.03**	42.9%	0.0032
watts.500	0.55 ± 0.04	**0.28 ± 0.02**	49.1%	0.0032
Barabasi 500	0.62 ± 0.05	**0.25 ± 0.02**	59.7%	0.0022
forest.fire	0.58 ± 0.04	**0.30 ± 0.02**	48.3%	0.001

500				
erdos.renyi 500	0.67 ± 0.05	**0.35 ± 0.03**	47.8%	0.0075
aging.prefatt 500	0.60 ± 0.04	**0.27 ± 0.02**	55.0%	0.0058
ExpoDegrDist 500	0.64 ± 0.05	**0.33 ± 0.03**	48.4%	0.007
watts.2000	0.48 ± 0.03	**0.22 ± 0.01**	54.2%	0.0036
Barabasi 2000	0.53 ± 0.04	**0.18 ± 0.01**	66.0%	0.0015
forest.fire 2000	0.50 ± 0.03	**0.20 ± 0.01**	60.0%	0.0053
erdos.renyi 2000	0.57 ± 0.04	**0.25 ± 0.02**	56.1%	0.0029
aging.prefatt 2000	0.52 ± 0.04	**0.21 ± 0.01**	59.6%	0.0029
ExpoDegrDist 2000	0.55 ± 0.04	**0.23 ± 0.02**	58.2%	0.0089

The results of Table. 6 clearly show the hybrid algorithm outperforms ECND in all test cases: Hybrid achieved 38% to 66% lower IGD scores (lower is better), Performance gains increased with network size (best for 2000-node networks), All improvements are statistically significant ($p < 0.01$), Works especially well on scale-free networks (66% improvement for Barabasi). The hybrid algorithm is consistently better, particularly for large, complex networks.

Table 7. Mean and Std. Dev. of HV over 30 Runs

Network	*ECND HV (mean ± std)*	*Hybrid HV (mean ± std)*	*HV Gain (% mean)*	*p-value*
Zachary	9.12 ± 0.08	**13.43 ± 0.22**	+47.3%	0.0003
Dolphins	6.55 ± 0.06	**10.77 ± 0.18**	+64.4%	0.0001
Polbooks	11.95 ± 0.11	**15.80 ± 0.19**	+32.3%	0.0014
Adjnoun	12.32 ± 0.10	**16.45 ± 0.22**	+33.5%	0.0018
Netscience	51.55 ± 0.19	**63.70 ± 0.11**	+23.6%	0.0034
Power	76.43 ± 0.20	**97.33 ± 0.09**	+27.3%	0.0032

Table 7 presents the HV metrics (mean ± standard deviation) from 30 independent runs on real-world benchmarks, with boldface indicating superior mean values (higher HV = better performance).

As expected and demonstrated in Table 7, the hybrid algorithm consistently outperforms ECND across all real-world networks, with: 23.6% to 64.4% higher HV scores (higher is better), Most significant gains in smaller networks (e.g., +64.4% for Dolphins), All improvements statistically significant ($p < 0.01$), Stable performance (low std. deviations). The hybrid approach reliably improves results on real networks, especially smaller ones. (Based on 30 independent runs)

5. Conclusion and Future Works

In this study, we proposed a hybrid framework that integrates the centrality-based ECND method with the pccsACGDE evolutionary algorithm to improve the identification of critical nodes in complex networks. By combining structural centrality measures with multi-objective evolutionary search, the method effectively balances network fragmentation and component size. Experimental results on both artificial and real-world networks demonstrated that the hybrid approach achieves significantly lower IGD and higher HV values compared to the baseline ECND, reflecting its superior convergence toward the ideal solution set and enhanced solution diversity. Future work will explore extending the method to dynamic and weighted networks, as well as incorporating learning-based strategies to guide the evolution process and further enhance performance in large-scale or real-time scenarios.

References

[1] Albert, Réka, and Albert-László Barabási. "Statistical mechanics of complex networks." *Reviews of modern physics* 74.1 (2002): 47.

[2] Crucitti, Paolo, Vito Latora, and Massimo Marchiori. "Model for cascading failures in complex networks." *Physical Review E* 69.4 (2004): 045104.

[3] Freeman, Linton. C. *Centrality in social networks: Conceptual clarification*. Social Networks, 1979. 1(3), 215–239.

[4] Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web*. Stanford infolab, 1999.

[5] Ugurlu, Onur. "Comparative analysis of centrality measures for identifying critical nodes in complex networks." *Journal of Computational Science* 62 (2022): 101738.

[6] Saxena, Akrati, and Sudarshan Iyengar. "Centrality measures in complex networks: A survey." *arXiv preprint arXiv:2011.07190* (2020).

[7] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2 (2002): 182-197.

[8] Zhang, Hai-Fei, et al. "Three-stage multi-modal multi-objective differential evolution algorithm for vehicle routing problem with time windows." *Intelligent Data Analysis* 28.2 (2024): 485-506.

[9] Béczi, Eliézer, and Noémi Gaskó. "Approaching the bi-objective critical node detection problem with a smart initialization-based evolutionary algorithm." *PeerJ Computer Science* 7 (2021): e750.

[10] Zhang, Lei, et al. "An interactive co-evolutionary framework for multi-objective critical node detection on large-scale complex networks." *IEEE Transactions on Network Science and Engineering* 10.3 (2023): 1722-1735.

[11] Nematalla Mottaki, Homayun Motameni, and Hosein Mohamadi. "Multi-objective optimization for coverage aware sensor node scheduling in directional sensor networks." *Journal of Applied Dynamic Systems and Control* 4.1 (2021): 43-52.

[12] Ajam, Leila, Ali Nodehi, and Hosein Mohamadi. "A Genetic-based algorithm to solve priority-based target coverage problem in directional sensor networks." *Journal of Applied Dynamic Systems and Control* 4.1 (2021): 89-96.

[13] Seyed Naghi Seyedaghaee, Ali Broumandnia, Reza Tavakkoli-Moghadam. Adaptive Multiobjective Differential Evolution Based on Parallel Cell Coordinate System. *Journal of Cluster computing, Springer*. (2025).

[14] Ajam, Leila, and Seyed Naghi Seyedaghaee. "Enhanced Critical Node Detection in Social Networks." *Computing and Informatics* 40.6 (2021): 1422-1443.

[15] Arulselvan, Ashwin, et al. "Detecting critical nodes in sparse graphs." *Computers & Operations Research* 36.7 (2009): 2193-2200.

[16] Zhou, Yangming, Jin-Kao Hao, and Fred Glover. "Memetic search for identifying critical nodes in sparse graphs." *IEEE transactions on cybernetics* 49.10 (2018): 3699-3712.

[17] Liu, Chanjuan, et al. "Improving Critical Node Detection Using Neural Network-based Initialization in a Genetic Algorithm." *arXiv preprint arXiv:2402.00404* (2024).

[18] Ventresca, Mario, Kyle Robert Harrison, and Beatrice M. Ombuki-Berman. "An experimental evaluation of multi-objective evolutionary algorithms for detecting critical nodes in complex networks." *European Conference on the Applications of Evolutionary Computation*. Cham: Springer International Publishing, 2015.

[19] Ding, Hongyuan, et al. "Research on Combining Pattern Mining and Evolutionary Algorithm for Critical Node Detection Problems." *International Journal of Frontiers in Engineering Technology* 4.4 (2022).

[20] Wang, Xue-Guang. "An algorithm for critical nodes problem in social networks based on owen value." *The Scientific World Journal* 2014.1 (2014): 414717.