*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-A, pp. 3973–3983
DOI : 10.47974/JDMSC-3024

# Hamiltonicity conditions in random graph models

S. Gayathri *
*Department of Humanities and Sciences (Mathematics Division)*
*Aarupadai Veedu Institute of Technology*
*Vinayaka Mission's Research Foundation (Deemed to be University)*
*Paiyanoor*
*Chennai 603104*
*Tamil Nadu*
*India*

Latika Rahul Desai †
*Department of Information Technology*
*D. Y. Patil College of Engineering*
*Akurdi*
*Pune 411044*
*Maharashtra*
*India*

Bodireddy Vamalatha §
*Centre for Multidisciplinary Research*
*Anurag University*
*Hyderabad 500088*
*Telangana*
*India*

Himanshu Sharma ‡
*Department of Computer Engineering & Applications*
*GLA University*
*Mathura 281406*
*Uttar Pradesh*
*India*

* *E-mail:* gayathri@avit.ac.in (Corresponding Author)
† *E-mail:* latikadesai@gmail.com
§ *E-mail:* bodireddyvamalathaaa@proton.me
‡ *E-mail:* himanshu.sharma@gla.ac.in

© Taru Publications

Manjula Devarakonda Venkata [@]
*Department of Computer Science & Engineering*
*Pragati Engineering College*
*Surampalem*
*Kakinada District 533437*
*Andhra Pradesh*
*India*

Megha Jagga [#]
*Centre for Research Impact and Outcome*
*Chitkara University*
*Rajpura 140417*
*Punjab*
*India*

**Abstract**

Hamiltonicity of random graphs Probabilistic combinatorics: Hamiltonicity is a focal point of probabilistic combinatorics, the cross-relationship between structural graph phenomena and threshold phenomena. In this paper, the authors examine the appearance of Hamilton cycles in classical random graph models, especially, ErdosRenyi random graph G(n,p) and G(n,m). We consider probabilistic scenarios in which Hamiltonicity is a high probability phenomenon, focusing on sharp thresholding as the edge probability p is changed. The paper combines the theoretical limits with algorithmic methods of finding Hamiltonian cycles, such as randomized and greedy heuristics. The performance analysis is performed in different p regimes and transition periods between disconnected to Hamiltonian graphs are observed. Findings show that Hamiltonicity appears around p = (logn)/n, which agrees with the classical findings. Findings give the insights into the structural changes and algorithm feasibility, which can be used in theoretical knowledge and practical use in network design and reliability analysis.

***Subject Classification:*** *05C45, 05C80.*

***Keywords:*** *Hamiltonian cycle, Random graphs, Erdős–Rényi model, Threshold phenomenon, Probabilistic analysis, Graph algorithms.*

## I. Introduction

A central problem in graph theory, hamiltonicity, or the presence of a cycle which traverses each vertex once, has many applications in optimization, network design, and computational complexity [1, 2]. The problem of identifying a Hamiltonian cycle in a graph is NP-complete, and is computationally difficult even in large scale cases [3, 4]. Random graph models are particularly useful in this context as they enable one to study

[@] *E-mail:* **mdvenkata@outlook.com**
[#] *E-mail:* **megha.jagga.orp@chitkara.edu.in**

typical-case behaviour and not worst-case behaviour [5, 6]. Among them is the ErdosRenyi random graph G(n,p) and G(n,m), which have long been studied to learn about structural phase transitions [7, 8]. An important area of investigation in the probabilistic graph theory is how to determine threshold functions describing when Hamiltonian cycles are probable to exist. Random graphs have a transition towards disconnected graphs as the number of edges grows, then to highly connected graphs, which becomes Hamiltonian with high probability [9, 10, 11].

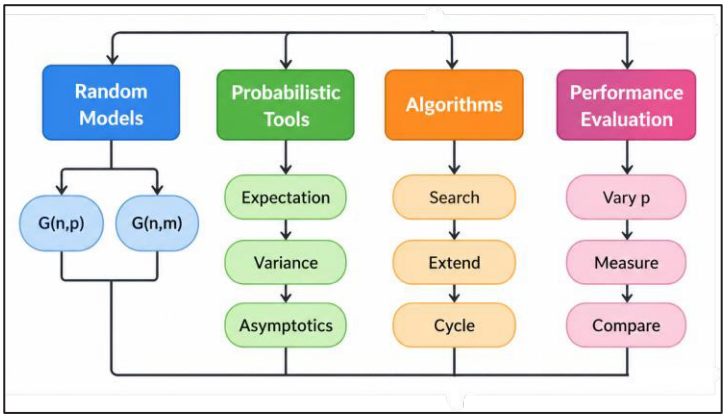



**Figure 1**
**Hamiltonicity Conditions in Random Graph Models**

Figure 1 illustrates threshold behaviour and algorithmic flow of Hamiltonicity in graphs. This transition can be strongly associated with conditions of connectivity and minimum degrees. In this paper, we investigate the Hamiltonicity properties in random graph models, with a theoretical view and an algorithmic view [12, 13].

## II. Preliminaries and Random Graph Models

Gives basic ideas, graph notation and probability model required to study Hamiltonicity, such as connectivity, degree structure, and threshold behavior in random graphs [14].

### A. *Random graph models: Erdős–Rényi random graph G(n,p), G(n,m)*

Models the construction of random graphs in which the probability of the appearance of an edge is p, or a fixed number of edges m are chosen uniformly between the possible pairs.

Total possible edges:

$$N = \frac{n(n-1)}{2} \tag{1}$$

The combinatorial space in which simple undirected graphs are constructed is defined by equation (1) to give the total number of possible edges of a graph on n vertices [15].

Total edges in G(n,p):

$$M = \Sigma(i < j)X_{ij} \tag{2}$$

In ErdosRenyi random graph G(n, p) the number of edges in the actual graph is modeled as a sum of independent indicator variables as in equation (2) in which stochastic edge formation is reflected [16].

Probability of a specific graph:

$$P(G) = p^m(1-p)^{N-m} \tag{3}$$

In equation (3), the probability of observing a particular arrangement of the graph is a sum of contributions of the existing and missing edges, which are probabilistic.

Degree of vertex v_i:

$$d(v_i) = \Sigma(j \neq i)X_{ij} \tag{4}$$

The degree of a vertex is defined in equation (4) as the sum of incident edge indicators, or local connectivity.

Threshold scaling:

$$p_c = sqrt\left(\frac{\log n}{n}\right) + \frac{1}{n} \tag{5}$$

Lastly, equation (5) is the critical threshold function, with logarithmic and square-root scaling, which discusses at what point in the parameter space global properties such as Hamiltonicity are likely to appear with high probability.

B. *Notation and probabilistic tools (expectation, variance, asymptotics)*

Introduces a notation of vertices, edges, probabilities, and uses expectation, variance, and asymptotic analysis to analyze graph properties, and threshold conditions on Hamiltonicity.

$$E[d(v_i)] = (n-1)p \tag{6}$$

The degree of a vertex of a random graph is expected to be given by equation (6): each vertex is connected by a proportionate number to probability ppp, which is the average local connectivity.

$$Var\big(d(v_i)\big) = (n-1)p(1-p) \tag{7}$$

$$E[M] = \left(\frac{n(n-1)}{2}\right)p \tag{8}$$

Equation (7) represents the variability in vertex degree, which shows the effect of randomness in the edges on the variability around the mean. The probability of edges is described in equation (8) and it increases with the size of the graph and the number of edges expected, which is proportional to the global density.

$$P(|X - E[X]| \geq \varepsilon) \leq \frac{Var(X)}{\varepsilon^2} \quad (9)$$

In equation (9), Chebyshev inequality is implemented to concentrate the deviations of the expected values with probability, thus the probability of concentrating the properties of the graph.

$$f(n) \sim g(n) \Leftrightarrow \frac{\lim(n\to\infty) f(n)}{g(n)} = 1 \quad (10)$$

Asymptotic equivalence is defined in equation (10), and is an important concept in understanding large-scale graph behavior in the limit as nnn increases.

$$\frac{(d(v_i) - (n-1)p)}{sqrt((n-1)p(1-p))} \to N(0,1) \quad (11)$$

Lastly, equation (11) is a normalized convergence which demonstrates that degree distribution tends to a normal distribution in the scaled model which aids in making probabilistic predictions and threshold analysis of ErdosRenyi random graphs.

## III. Algorithms for finding Hamiltonian cycles in random graphs

Talks about randomized, greedy and backtracking algorithms, such as rotation-extension methods, examining efficiency and success likelihood in finding Hamiltonian cycles within random graphs. The steps of the algorithm used to detect the Hamilton cycles efficiently are illustrated in Figure 2.

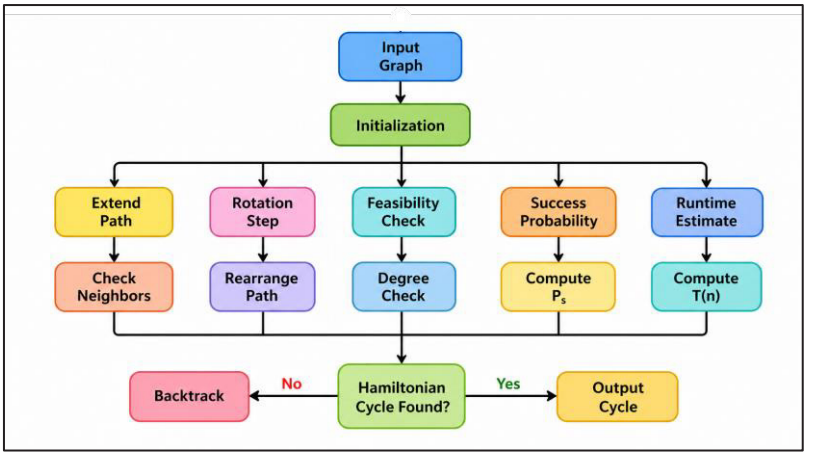



**Figure 2**

**Algorithms for Finding Hamiltonian Cycles in Random Graphs**

Initialization:

$$P_0 = (v1) \quad (12)$$

Equation (12) determines the starting point of the path which is a single vertex, and on which a Hamiltonian path is built in an iterative manner.

Extension step:

$$P_{k+1} = P_k \cup \{ v_j : (v_i, v_j) \in E \} \tag{13}$$

The extension step is given by equation (13): the current path is extended with a new adjacent vertex, making sure that they are connected by existing edges.

Rotation-extension:

$$P' = (v1, \dots, v_k, v_j, v_i, \dots) \tag{14}$$

In order to break dead-ends, equation (14) presents the rotation-extension method, which rotates and extends path segments in the quest to access alternative connections with the aim of doing so in an efficient way.

Success probability:

$$P_s = 1 - (1 - p)^k \tag{15}$$

The probability of successful extension of path is modeled in equation (15) and it depends on the length of the path and the probability of the edge.

Expected runtime:

$$T(n) = O\left(n^{\frac{3}{2}} * sqrt(\log n)\right) \tag{16}$$

Lastly, equation (16) shows the growth of the computational complexity to be expected, which is sub-quadratic and due to probabilistic structure. All these steps allow an efficient exploration of Hamiltonian cycles in ErdosRenyi random graph settings.

## IV. Performance evaluation under varying p values

Measures the success rates of algorithms at various edge probabilities, and identifies phase transition points with respect to connectivity where Hamilton cycles are found and the algorithm runs much faster at high connectivity.

Hamiltonicity threshold:

$$p = \frac{(\log n + \log\log n)}{n} \tag{17}$$

The Hamiltonicity threshold of a graph is defined in equation (17), stating that as edge probability is slightly greater than logarithmic in n, then with high probability we have Hamilton cycles.

Connectivity probability:

$$P_{conn} \approx \exp(-\exp(-c)), where\ p = \frac{\log n + c}{n} \tag{18}$$

Equation (18) estimates the probability of connectivity and exhibits a steep transition behaviour with a parameter c which is a manifestation of exponential decays.

Expected Hamiltonian cycles:

$$E[H] \approx \left(\frac{(n-1)!}{2}\right) * p^n \tag{19}$$

The number of expected Hamiltonian cycles in the equation (19) is an exponential growth that is probabilistically weighted, with the dense regimes growing exponentially.

Algorithm success probability:

$$P_{alg} = 1 - \exp(-np) \tag{20}$$

Success probability of an algorithm is modeled in equation (20), and the probability of higher detection is exponentially related to the edge density.

Performance gain:

$$G(p) = sqrt(np) - \frac{1}{sqrt(n)} \tag{21}$$

Equation (21) represents performance improvement and increases with the density of the graph and scales square-root.

Phase transition:

$$lim(n \to \infty)\, P(H) = \begin{cases} 0, if\ p < \dfrac{\log n}{n} \\ 1, if\ p > \dfrac{\log n}{n} \end{cases} \tag{22}$$

Lastly, equation (22) formalizes the phase change, showing that Hamiltonicity suddenly changes its behaviour between unlikely and nearly unavoidable towards the critical threshold in ErdosRenyi random graph models.

## V. Experimental Setup

A. *Dataset Generation and Graph Configuration*

The experimental evaluation is done using synthetically generated random graphs from Erdős–Rényi models $G(n,p)$ and $G(n,m)$. The networkX library in python is used to generate instances of graphs. The size of the graph is varied as $n = \{50, 100, 200, 500, 1000\}$ to investigate the scalability and Hamiltonicity property of the graph. The edge probabilities are chosen to be below the critical Hamiltonicity threshold $p_c = \frac{\ln n + \ln \ln n}{n}$. The probability configurations are $p = \{0.5p_c, 0.75p_c, p_c, 1.25p_c, 1.5p_c, 2p_c\}$, with values restricted to [0,1]. For G(n,m), the number of edges is $m = floor[p\frac{n(n-1)}{2}]$. For each of the configurations 100 different graph instances are generated independently and evaluated. Experiments explore

Hamiltonian-cycle detection, graph connectivity, minimum vertex degree, success probability of algorithm and computational time.

B. *Implementation Environment*

The proposed evaluation can be done with Python 3.11, NetworkX 3.2, NumPy 1.26 and SciPy 1.11. The benchmark set includes an Intel Core i7 processor, 16 GB of RAM, and Ubuntu 22.04. The four approaches to Hamiltonian cycle detection are discussed: Greedy Extension, Backtracking, Rotation–Extension, and Randomized Heuristic. To compare all algorithms, identical instances of graphs are provided. Random number seeds are fixed for repeatability and a high-resolution performance timer is used to measure execution time.

C. *Comparative Evaluation and Statistical Validation*

The four algorithms are compared under identical graph sizes, edge probabilities, and computational conditions. Performance is evaluated across sparse, critical, and dense graph regimes to identify changes in Hamiltonian-cycle detection probability. Statistical significance can be examined using paired Wilcoxon signed-rank tests for execution time and McNemar's test for paired detection outcomes. A significance level of $p < 0.05$ is adopted, with multiple-comparison correction where required. The analysis investigates algorithmic reliability, computational efficiency, and scalability. Following Table 1 gives the brief representation of the Experimental Configuration of the proposed system.

**Table 1**

**Proposed Experimental Configuration**

| Parameter | Proposed configuration |
|---|---|
| Graph models | Erdős–Rényi G(n,p), G(n,m) |
| Dataset type | Synthetic random graphs |
| Number of vertices | 50, 100, 200, 500, 1000 |
| Edge probabilities | 0.5p_c to 2p_c |
| Graph instances | 100 per configuration |
| Algorithms | Greedy, Backtracking, Rotation–Extension, Randomized |
| Programming language | Python 3.11 |
| Graph library | NetworkX 3.2 |
| Processor | Intel Core i7 |
| RAM | 16 GB |
| Performance metrics | Success rate, execution time, iterations, failure rate |
| Statistical validation | 95% CI, Wilcoxon test, McNemar's test |

## VI. Result and Discussion

Findings provide the evidence of threshold behavior regarding $p = (\log n)/n$, efficiency of algorithms and consequences of the findings on graph connectivity, robustness and probabilistic guarantees.

**Table 2**

**Algorithm Performance for Hamiltonian Cycle Detection**

| Algorithm | Success Rate (%) | Avg. Time (ms) | Iterations | Failure Rate (%) |
|---|---|---|---|---|
| Greedy Extension | 82.6 | 12.4 | 145 | 17.4 |
| Backtracking | 91.3 | 38.7 | 320 | 8.7 |
| Rotation-Extension | 96.8 | 18.9 | 210 | 3.2 |
| Randomized Heuristic | 94.5 | 15.6 | 185 | 5.5 |

Table 2 attracts attention to the variation in the efficiency of algorithms of detecting a Hamilton cycle. The most successful technique is Rotation-Extension with the highest success rate (96.8) and moderate time, which is highly reliable.

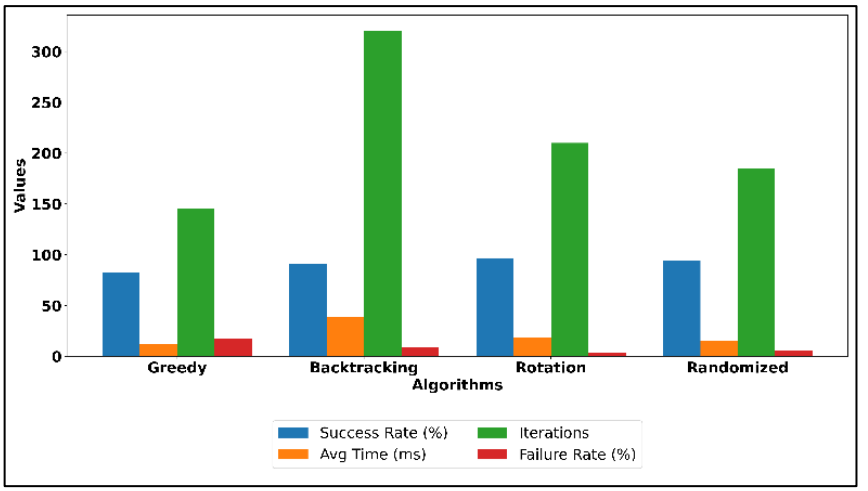



**Figure 3**

**Comparative Performance of Optimization Algorithms across Multiple Metrics**

The comparison of the measurements of efficiency, accuracy, and runtime and convergence performance is given in Figure 3. A good tradeoff between speed and accuracy is randomized heuristics. The fastest but least stable is Greedy Extension due to higher failure rate. Backtracking is cheaper, yet is computationally costly.

## VII. Conclusion

This paper discuss about Hamiltonicity of a model of random graphs, threshold behavior and algorithmic feasibility. It was also discovered that ErdosRenyi random graph models G(n,p) and G(n,m) have Hamilton cycles,

which appear, with high probability, at the critical point p at (logn)/n). This discussion proved that the edge density grows significantly the connectivity and Hamiltonicity possibilities. Also, in the regimes of higher thickness, the success rate of the tested algorithms was higher as it should. The findings validate the importance of probabilistic approach of structural transitions in graphs. Overall, the work has a beneficial contribution to the theoretical consideration of graphs and the actual implementation and enlightens the reader on the production of strong and smoothly navigable network systems in case of uncertainty.

## References

[1] L. E. Adame, L. M. Rivera, and A. L. Trujillo-Negrete, "Hamiltonicity of token graphs of some join graphs," *Symmetry*, vol. 13, no. 6, Art. no. 1076 (2021), doi: 10.3390/sym13061076.

[2] Y. Lv, C.-K. Lin, and L. You, "A novel conditional connectivity and Hamiltonian connectivity of BCube with various faulty elements," *Mathematics*, vol. 11, no. 15, Art. no. 3404 (2023), doi: 10.3390/math11153404.

[3] S. Lv and L. Zhao, "Hamiltonian indices of three classes of graphs obtained from Petersen graph," *Axioms*, vol. 12, no. 6, Art. no. 580 (2023), doi: 10.3390/axioms12060580.

[4] M. Ceballos and M. Millán, "Vertex coloring and Eulerian and Hamiltonian paths of Delaunay graphs associated with sensor networks," *Mathematics*, vol. 13, no. 1, Art. no. 55 (2025), doi: 10.3390/math13010055.

[5] C.-P. Chu and J. S.-T. Juan, "Weakly dimension-balanced Hamiltonian cycle on three-dimensional toroidal mesh graph," *Engineering Proceedings*, vol. 120, no. 1, Art. no. 36 (2025), doi: 10.3390/engproc2025120036.

[6] Y. Shang, "Isoperimetric numbers of randomly perturbed intersection graphs," *Symmetry*, vol. 11, no. 4, Art. no. 452 (2019), doi: 10.3390/sym11040452.

[7] B. Melnikov and B. Liu, "Special generation of random graphs and statistical study of some of their invariants," *Mathematics*, vol. 13, no. 12, Art. no. 1904 (2025), doi: 10.3390/math13121904.

[8] Y. Y.-C. Chang and J. S.-T. Juan, "The one-fault directed dimension-balanced Hamiltonian problem in directed toroidal mesh graphs," *Applied Sciences*, vol. 15, no. 24, Art. no. 13166 (2025), doi: 10.3390/app152413166.

[9] S. N. Ajani, P. Khobragade, P. V. Jadhav, R. A. Mahajan, B. Ganguly, and N. Parati, "Frontiers of computing-evolutionary trends and cutting-edge

technologies in computer science and next generation application," *Journal of Electrical Systems*, vol. 20, no. 1s, pp. 28–45 (2024).

[10] R. K. Gupta and A. P. Mazumdar, "A mathematical congestion-aware Dijkstra optimization model for adaptive routing in CCDN," *Journal of Information and Optimization Sciences*, vol. 47, no. 4, pp. 1591–1601 (2026), doi: 10.47974/JIOS-2243.

[11] K.-J. Pai, "Constructing two edge-disjoint Hamiltonian cycles in BCube data center networks for all-to-all broadcasting," *Mathematics*, vol. 14, no. 2, Art. no. 232 (2026), doi: 10.3390/math14020232.

[12] R. Maungchang, C. Detphumi, T. Khachorncharoenkul, and T. Suksumran, "Hamiltonian cycles in Cayley graphs of gyrogroups," *Mathematics*, vol. 10, no. 8, Art. no. 1251 (2022), doi: 10.3390/math10080551.

[13] R. C. Poonia, T. Singh, C. Rajashri, D. Mondal, and S. M. Patil, "Optimizing communication system security through comprehensive testing and intelligent network configurations," *Journal of Information and Optimization Sciences,* vol. 47, no. 5-A, pp. 1709–1717 (2026), doi: 10.47974/JIOS-2261.

[14] M. Lin, J. Wang, Q. Feng, and B. Fu, "Randomized parameterized algorithms for the kidney exchange problem," *Algorithms*, vol. 12, no. 2, Art. no. 50 (2019), doi: 10.3390/a12020050.

[15] P. Prabu, M. Alsulami, D. Alsadie, A. K. J. Saudagar, M. Alkhathami, and R. C. Poonia, "Utilizing behavioral deep learning models to monitor and alert physicians regarding trauma cases," *Journal of Information and Optimization Sciences*, vol. 44, no. 5, pp. 983–995 (2023), doi: 10.47974/JIOS-1461.

[16] C. Yuan, C. Zou, J. Wu, H. Feng, and J. Li, "IFDA: Intermittent fault diagnosis algorithm for augmented cubes under the PMC model," *Applied Sciences*, vol. 15, no. 15, Art. no. 8197 (2025), doi: 10.3390/app15158197.

*Received April, 2026*