Research Article

A Quantum-Inspired Evolutionary and IPSO Hybrid Model For Efficient Graph Colouring Under Multi-Constraint Conditions

Uma A[1], Nanthana Krishnan[2] and Franklin Thamil Selvi M S[3]

[1]Department of Mathematics, Annamalai University, Chidambaram Tamilnadu, India (Deputed to Government Polytechnic Collage, Dr.Radhakrishnan Nagar, Chennai, Tamilnadu, India. Email: Uma.feat@yahoo.com
[2]Research Scholar, Department of Mathematics, Annamalai University, Chidambaram, Tamilnadu, India. Email: nanthanak2023@gmail.com
[3]Department of Mathematics, Sathyabama Institute of Science & Technology, Chennai,
Tamilnadu India. Email: thamizanand@gmail.com

Abstract:
Graph colouring, a fundamental problem in combinatorial optimization, plays a critical role in various real-world applications such as register allocation, scheduling, and frequency assignment. Efficiently solving the graph colouring problem under multiple constraints remains a major computational challenge, particularly for large and complex graphs. This study addresses these limitations by proposing a hybrid optimization model that integrates Quantum-Inspired Evolutionary Algorithms (QIEA) with an Improved Particle Swarm Optimization (IPSO) technique. The proposed model leverages the probabilistic representation and parallel search capabilities of QIEA along with the adaptive learning and velocity adjustment features of IPSO to explore the solution space effectively. The primary objective is to minimize the number of colours used while satisfying adjacency, capacity, and dependency constraints. Experimental evaluations conducted on benchmark graph instances demonstrate that the hybrid model significantly outperforms existing evolutionary and heuristic methods in terms of convergence speed, constraint satisfaction, and colouring efficiency. These results affirm the potential of the QIEA-IPSO hybrid in solving complex multi-constrained graph colouring problems.

Keywords: Graph Colouring; Quantum-Inspired Evolutionary Algorithm; Improved Particle Swarm Optimization; Hybrid Optimization; Constraint Satisfaction; Combinatorial Optimization; Multi-Constraint Graph Problems; Metaheuristic Algorithms; Colour Minimization; Evolutionary Computation.

1. Introduction

The Graph Colouring Problem (GCP) has been addressed using a variety of heuristic approaches. Researchers have investigated numerous heuristic and metaheuristic techniques to tackle this challenging optimization problem. One particularly interesting method is the Genetic Algorithm (GA) draws inspiration from natural selection and evolution. GA approaches are well-suited for handling combinatorial problems like GCP due to their ability to explore large search spaces and find near-optimal solutions [1]. Hybrid Evolutionary Algorithms (HEAs) represent a new and highly approach to combinatorial optimization. They involve integrating local search techniques into the structure of population-based evolutionary algorithms. The two main components of such methods are well-designed crossover operators and effective Local Search (LS) operators [2]. After crossover, the offspring are improved using LS operators before being added to the population. To develop a successful HEA for an optimization problem, both the crossover and LS operators need to be carefully designed. Designing a high-quality crossover operator is typically more challenging than designing an LS operator [3]. Creating a crossover requires first identifying certain "good properties" of the problem that should be inherited from the parents, followed by designing a suitable recombination process. The use of randomized crossovers in existing genetic techniques often has limited impact on optimization problems. HEAs have been applied to several well-known NP-hard combinatorial problems, yielding impressive results. On standard benchmarks, performed well compared to the best existing algorithms for those tasks. These findings strongly suggest that HEAs are among the most effective models for tackling difficult combinatorial optimization problems [4].
The GCP is a well-known computational optimization problem with numerous real-world applications. Examples include frequency assignment in cellular networks, timetabling, and memory allocation in compilers, planning, printed circuit board design and production scheduling. Due to its simple formulation, GCP not only holds practical significance but also serves as a benchmark platform for evaluating new algorithms [5]. Figure 1 illustrates a graph with 10 vertices and 15 edges, where three colours are used appropriately to colour the vertices without any adjacent vertices sharing the same colour. The vertex colouring problem has practical applications in mobile phone networks and radio frequency assignments. The vertex colouring problem is NP-complete from an algorithmic perspective, making it computationally challenging to solve efficiently for large and complex graphs [6].

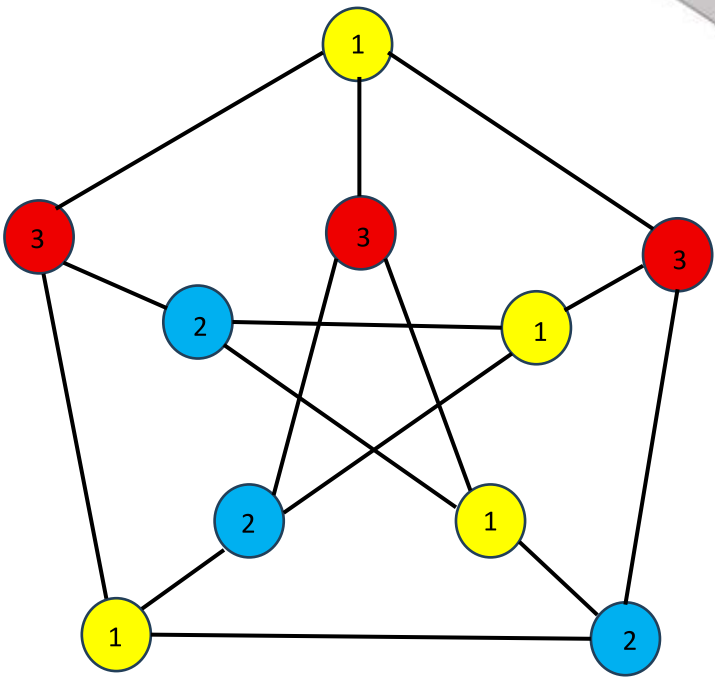

Figure 1: Vertex coloring

The earliest graph colouring algorithms aimed to solve the problem exactly by listing every possible vertex arrangement. These methods were extremely time-consuming and impractical for large networks. Heuristic approaches were proposed to find approximate solutions within a reasonable time. Among these, the greedy algorithm emerged as a logical method for colouring graph vertices efficiently [7]. The Recursive Largest First (RLF) algorithm partitions the uncoloured graph into colour classes, with each class containing vertices assigned the same colour. The algorithm colours one class at a time, adding vertices to the current class in a way that minimizes the number of remaining edges in the uncoloured subgraph. Beyond RLF, several local search-based algorithms have been introduced such as population-based methods, variable neighbourhood search, adaptive memory strategies, variable search areas, dynamic partitioned search using tabu techniques, iterative local search, simulated annealing and variable search space strategies [8]. A major challenge in these methods arises from the variation operators such as mutation and crossover often violate constraints. The offspring produced by these operators may therefore be infeasible. This issue is addressed through indirect constraint handling, wherein constraints are transformed into optimization objectives represented by an objective function. Candidate solutions that violate constraints are penalized by this function [9].

The difficulty of colouring a vertex increases with its assigned weight. The effectiveness of a graph colouring method is significantly influenced by how these weights are adjusted. Weight adjustment can be done predictably, continuously, or in a self-adaptive manner. The way a problem is represented and how the algorithm utilizes this representation determines how many adjacency checks are required to solve a specific problem instance. The Integer Merge Model (IMM) presented here directly addresses these challenges [10]. Graphs are typically represented using three primary data structures: the incidence matrix, the adjacency matrix, and the adjacency list. Although some specialized representations are tailored for specific graph characteristics such as sparse graphs the adjacency matrix is commonly used in GCP. The Binary Merge Model (BMM) is a novel representation for GCP [11]. While BMM offers a practical and efficient structure for GCP, IMM serves as a generalization. IMM retains the key advantage of BMM namely increased efficiency through the reduction of adjacency checks. IMM provides valuable structural information during the colouring process enabling the development of more sophisticated colouring algorithms and heuristics, and facilitating clearer explanation of these methods [12].

To effectively address the GCP, the authors introduce a novel GA approach. By presenting a GA-based method for solving the GCP, the study makes a significant contribution to the field and provides valuable insights into the algorithm's performance, especially when applied to large-scale graph instances. Recognize that the GCP involves multiple conflicting objectives, such as minimizing the number of colours used while optimizing other relevant graph properties [13]. To address this, a Multi-Objective Genetic Algorithm (MOGA) is employed a set of solutions that reflect various trade-offs among the objectives. Experimental results on several graph instances demonstrate that the MOGA produces a diverse set of high-quality solutions. The study emphasizes the considerable benefits of incorporating multiple objectives into GA-based frameworks and showcases the effectiveness of multi-objective optimization in the context of the GCP [14]. Introduced a novel GA-based solution for the GCP using a Reduced Quantum Genetic Algorithm (RQGA). Their approach maintains a fixed population size throughout the process, providing a unique strategy to enhance the exploration of the search space. Compared to existing GAs, the fixed-size pool method improves convergence rates and solution quality [15]. Proposed a hybrid strategy to solve the GCP and this method utilizes a hybrid representation for graph colouring that enables efficient exploration of the solution space. The approach incorporates a sophisticated locally guided search to produce high-quality solutions with minimal computational overhead. By introducing a novel encoding scheme and demonstrating the effectiveness of real-coded GAs for the GCP, this study also contributes significantly to the advancement of the field [16].

Key contributions of the paper are as follows:

- Introduced a novel hybrid QEA-IPSO algorithm for solving multi-constrained graph colouring problems efficiently.
- Developed a quantum-inspired encoding mechanism to enhance exploration in complex graph solution spaces.
- Integrated adaptive IPSO for local optimization, improving convergence speed and solution accuracy significantly.
- Formulated a robust constraint-handling mechanism for capacity, precedence, and adjacency constraints in graph colouring.
- Validated the model on benchmark datasets, demonstrating superior performance over existing metaheuristic algorithms.

2. Related Works

In their enhanced evolutionary algorithm for the GCP, introduce a novel crossover operator based on complementing solutions. This operator promotes exploration of the search space by enabling the exchange of complementary genetic information between solutions. Extensive testing on various graph instances demonstrates that the proposed crossover operator significantly improves both convergence speed and solution diversity. The benefits of this approach, in terms of execution time and solution quality are validated through comparisons with other standard GAs [17]. To further improve the solutions generated by the GA, the hybrid approach integrates Variable Neighbourhood Search (VNS) as a local search technique. By embedding VNS within the GA's evolutionary framework, the method efficiently explores the solution space and refines solutions at the local level. Experimental evaluations on graph instances show that the hybrid approach outperforms both standalone GA and VNS techniques [18]. This study highlights the synergy between VNS and GA demonstrates the potential of hybrid methods for solving the GCP and achieves competitive performance in graph colouring tasks. To address the GCP in a collaborative computing environment, propose a parallel version of the GA. The researchers combine the benefits of concurrent processing with the evolutionary operations of the GA to enhance solution efficiency. In large-scale graph scenarios, the parallel hybrid approach reduces execution time and achieves faster convergence by distributing the GA population across multiple processing nodes. Comparative experimental evaluations indicate that the parallel hybrid GA produces higher-quality solutions more efficiently than existing serial GAs [19] This work provides valuable insights into the use of parallelism to solve combinatorial optimization problems such as the GCP. For the rotating serum manufacturing challenge, proposed a dynamic multi-objective GA. To balance exploration and exploitation in the solution space, their approach employs a hybrid crossover operator that combines multiple crossover techniques. Throughout the evolutionary process, the algorithm dynamically adjusts its parameters to enhance both solution diversity and convergence [20]. The study highlights the advantages of adaptive parameter tuning in genetic algorithms and contributes to the advancement of multi-objective optimization methods. Evaluated several GA strategies and compared them based on the chromatic number produced by each method. Strongly recommended selecting an appropriate GA variant to achieve faster convergence even with smaller population sizes [21]. Graph-based experimental evaluations revealed that the proposed GA operators significantly improved algorithm performance. In comparison to existing GAs, their method leveraged network structure characteristics to produce competitive results. This study demonstrates the potential of tailoring genetic algorithms to specific optimization problems and provides useful insights into problem-specific operator design for the GCP [22]. Due to the significant computational complexity of the GCP, various heuristic techniques have been developed. These heuristics typically offer suboptimal solutions in polynomial time, depending on the desired accuracy and the complexity of the graph. Based on the principle of selecting a vertex first and then assigning a suitable colour, these heuristics fall into specific categories. One such method is the Largest-Fit (LF) approach, where vertices with higher degrees are coloured first. Another similar technique is the Smallest Last (SL) algorithm also belongs to the family of sequential heuristics [23]. While both methods are fast and easy to implement, they generally do not yield optimal colouring results. Although such interchanges increase computation time, they have been shown to enhance solution quality. The Greedy Independent Set (GIS) technique is another heuristic method based on the concept of the maximum independent set. In this approach, vertices are examined in a predetermined order, and a vertex is coloured only if it is not adjacent to another vertex of the same colour [24] GCP has also led to the development and adaptation of evolutionary algorithms. In some approaches, solutions are evaluated using a greedy strategy. Experimental results demonstrate that while such methods can perform comparably to GA, their efficiency is generally lower than that of more advanced algorithms. The initial results of a Genetic Local Search (GLS) approach were reported and its experiments on random graphs showed that the evolutionary approach outperformed Tabucol, a well-known tabu search algorithm for graph colouring [25]. In their study, proposed ANTCOL integrates two graph-colouring constructive algorithms ANTRLF and ANTDSATUR based on RLF and Degree of Saturation (DSATUR) respectively. Experimental results indicated that ANTCOL, when based on RLF, performed better than its DSATUR-based counterpart. Comparative experiments revealed that Modified Multi-Graph Colouring (MMGC) produced significantly better results than ANTCOL [26]. Proposed Ant Local Search for Colouring (ALS-COL) an ant-colony-based method that diverged from existing ACO strategies. While most ACO-based approaches focus on constructing complete solutions, ALS-COL introduced the concept of neighbourhood search allowing ants to refine solutions locally rather than building them entirely from scratch. Experiments showed that ALS-COL yielded better results with reduced time and memory complexity. Addressed the GCP by enhancing an existing Particle Swarm Optimization (PSO) method with a modified turbulence mechanism aiming to improve solution quality and convergence speed [27]. Proposed a hybrid approach that incorporates a recombination operator to enhance solution quality. To efficiently solve the planar GCP, introduced a modified PSO algorithm enhanced with fuzzy logic. Experimental results demonstrated improved outcomes with reduced computational time and storage requirements. Further advanced the GCP by introducing a modified turbulence mechanism to the existing PSO. Their proposed model includes three core components: the wandering strategy, assignment strategy, and turbulence mechanism [28]. This hybrid approach effectively and precisely solves the planar GCP using only four colours. For the GCP, proposed a GA-based method combined with multipoint directed mutation. To significantly improve the performance of existing GA, a novel operator double point guided mutation operator was introduced. This operator enhances the algorithm's efficiency by providing better exploration and exploitation of the search space [29]. To address the k-colouring problem, introduced an efficient Quantum Annealing approach. Inspired by quantum physics, this method extends simulated annealing by incorporating an additional variable beyond the existing temperature parameter. This extra variable adjusts the neighbourhood radius (to control diversity) and enhances the evaluation function through an energy component that reflects interactions among multiple replicas [30]. Testing on several DIMACS benchmark graphs demonstrated the method's effectiveness. PSO is another prominent metaheuristic has been explored for graph colouring. PSO has been applied in various domains such as video and image processing, electrical network planning and restructuring, surveillance systems, antenna design, electromagnetic field optimization, and more [31]. Originally designed for continuous-valued spaces, PSO must be adapted for discrete combinatorial problems like GCP. Existing arithmetic operations such as subtraction, addition, and multiplication used in velocity and position update formulas must be redefined for discrete search spaces. These modifications affect how velocity, particle trajectory, speed regulation, and energy are interpreted, enabling PSO to effectively operate within a discrete combinatorial context [32].

2.1 Research Gap in Efficient Graph Colouring under Multi-Constraint Conditions

Despite significant advancements in graph colouring algorithms, existing approaches face challenges when dealing with multiple simultaneous constraints such as adjacency, precedence, and capacity restrictions. Most existing techniques, including greedy algorithms, evolutionary strategies, and metaheuristics, either lack the ability to efficiently explore complex solution spaces or struggle with local optima and scalability issues. Existing hybrid models often fail to integrate quantum-inspired global search mechanisms with adaptive local optimization strategies leading to suboptimal results in large and dynamic graphs. There is a clear lack of robust, scalable, and constraint-sensitive hybrid algorithms that can effectively balance exploration and exploitation while respecting the intricate constraints of real-world GCP.

3. Problem Formation

Let $G = (V, E)$ be an undirected graph where: V is the set of vertices (|V| = n); E is the set of edges, $C = \{1, 2, \ldots, k\}$ a set of available colours.

The goal is to assign a colour $c_x \in C$ to each vertex $v_x \in V$ such that the following multi-constraints are satisfied, and the total number of colours used is minimized.

Colour Assignment Function: Let: $f: V \rightarrow C \quad f(v_x) = c_x$

Where c_x is the colour assigned to vertex v_x

Adjacency Constraint (Proper Colouring): No two adjacent vertices share the same colour:

$$\forall (v_x, v_y) \in E, \qquad f(v_x) \neq f(v_y)$$

Capacity Constraint: Each colour c ∈ C can be used for at most θ vertices (e.g., limited resource or frequency capacity): $\forall_c \in C,\ |\{v_x \in V | f(v_x) = c\}| \leq \theta$

Dependency Constraint: Some vertices must be coloured in a certain order or follow a dependency rule: $\text{If} v_x \prec v_y,\ f(v_x) < f(v_y)$

Where ≺ denotes a precedence relation between v_x and v_y

Objective Function: Minimize the total number of colours used:

$$\min\left(\max_{v_x \in V} f(v_x)\right)$$

Or in case of penalty-based constraint handling:

$$min\left(\max_{v_x \in V} f(v_x) + \lambda_1 . Viol_{adj} + \lambda_2 . Viol_{cap} + \lambda_3 . Viol_{dep}\right)$$

Where: $Viol_{adj}$: total number of adjacency constraint violations, $Viol_{cap}$: total number of capacity constraint violations, $Viol_{dep}$: total number of dependency violations, λ_x: penalty coefficients.

Efficient graph colouring under multi-constraint conditions involves assigning colours to a graph's vertices to minimize the number of colours used while ensuring adjacent vertices receive different colours. A colouring function assigns each vertex v_i∈V a colour c_i from a finite set C, given a graph G = (V, E). Additional constraints include precedence-based colouring order and capacity limits on colour reuse due to resource constraints. The objective is to minimize colour usage or penalize constraint violations using a composite penalty function. This approach is applicable in task scheduling, resource allocation, and frequency assignment in complex systems.

4. Materials and methods

The proposed QIE-IPSO composite model integrates enhanced swarm intelligence with quantum-inspired computing for efficient graph colouring under multi-constraint conditions shown in Figure 2. Benchmark datasets with constraints such as proximity, colour limits, and degree-based rules are used. Potential solutions are encoded using quantum-inspired qubit representations to explore diverse search spaces. Quantum rotation gates and mutation techniques maintain diversity and prevent premature convergence. A fitness function evaluates colour conflicts and constraint violations, while the IPSO component dynamically updates velocity and position based on global and local optima. This hybrid approach balances global exploration and local exploitation to achieve optimal or near-optimal solutions efficiently.

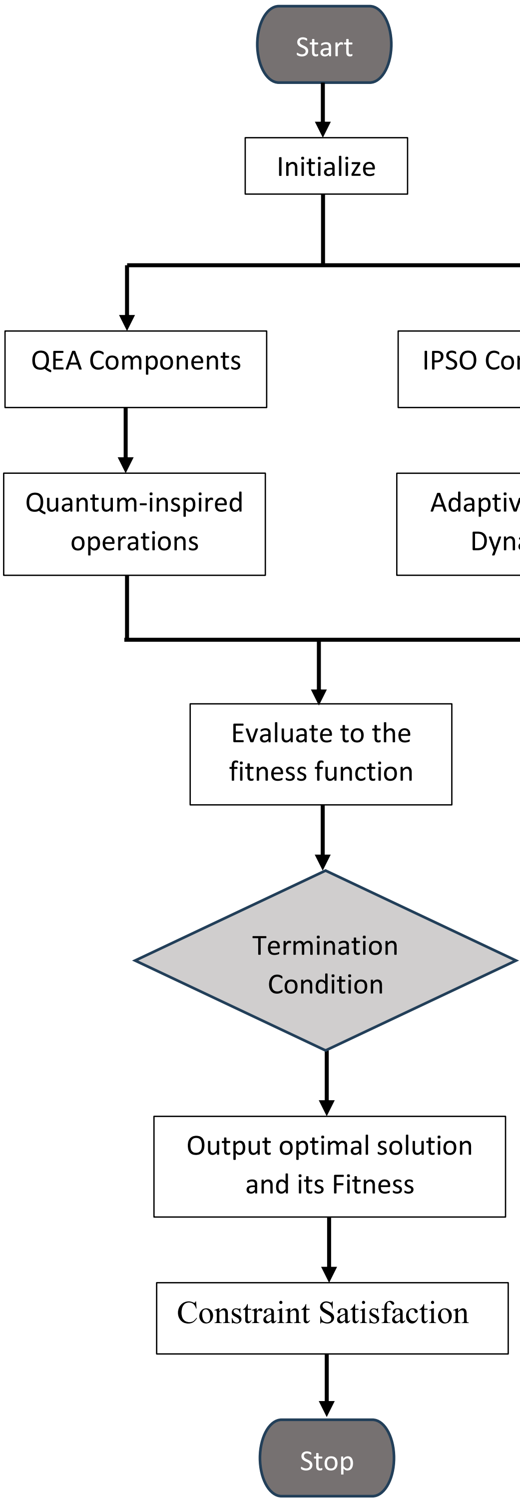

Figure 2: Proposed Architecture

4.1 Graph Parameters

Given a graph G, say two vertices u, v are of the same type if $N(v)\backslash\{u\} = N(u)\backslash\{v\}$. If there is vertex partition of G into t parts such that all vertices are of the same type in each part, then the neighborhood diversity of the graph is t. Given a graph G, clique width (C_w) of a graph is defined as follows. A labeling function $\psi: V(G) \mapsto [w]$ that labels the vertices of the garph based on the following four recursive steps for two different value $x, y \in [w]$.

- Introduce $\psi(v) = x$; creating a new vertex v of label x.
- Disjoint union $G_\psi = H_{\psi'} \oplus H_{\psi''}$; Disjoint union of two different labeled graphs and $H_{\psi'}$ and $H_{\psi''}$
- Relabel $G_\psi = \rho_{x \to y} G_{\psi'}$: G_ψ is a labeled graph where vertices labeled with x in $G_{\psi'}$ are now labeled with y.
- Join $G_\psi = n_{x,y} G_{\psi'}$: Every vertex labeled with x and every vertex labeled with y in $G_{\psi'}$ are adjacent in G_ψ

The minimum value of w, for which G_{psi} can be defined is called the clique width of G. $Tree\ decomposition$ of a graph G is a tree T with nodes $I_1, \dots, I_l$, where $I_x \subseteq V(G)$ with the following properties.

- Every vertex $v \in V(G)$ there is at least a I_x containing v. For every edge $e(u, v) \in E(G)$ there is at least one I_y containing both u and v.
- For each vertex u ∈ V(G), all the nodes X, containing u in T, induce a connected subtree in T.

4.2 Dataset description

To evaluate the proposed QIEA-IPSO hybrid model, diverse datasets covering various graph structures and complexities were used shown in Table 1. Benchmark cases like DSJC250.5, DSJC500.1, queen8_12, and le450_25a from the DIMACS, COLOR02, and COLOR03 libraries test the model's robustness on graphs with varying node density and proximity constraints. Two synthetic networks (Synthetic_Graph_1 and Synthetic_Graph_2) simulate multi-constraint conditions. Real-world cases, including compiler register allocation, university timetabling, and telecom frequency assignment, demonstrate practical relevance. This dataset variety ensures comprehensive evaluation across scalable, constraint-intensive, and feasible environments.

Table 1: Graph Colouring data sets with Important Features

Dataset Name	Source	Graph Type	Nodes (V)	Edges €	Constraints Applied	Purpose
DSJC250.5	DIMACS	Random Dense	250	~15,000	Adjacency, Capacity	Medium-scale benchmark comparison
DSJC500.1	DIMACS	Random Sparse	500	~12,500	Adjacency	Sparse graph performance test
queen8_12	DIMACS	Regular (Chessboard)	96	1,368	Adjacency, Dependency	Structured graph with pattern constraints
le450_25a	COLOR02	Random Structured	450	~8,000	Adjacency, Capacity	Constraint-heavy mid-size graph
flat1000_76_0	COLOR03	Flat random	1000	~25,000	Adjacency	Large-scale performance and scalability test
Synthetic_Graph_1	Custom (Random)	Synthetic Sparse	200	2,000	Adjacency, Capacity	Generalization to custom sparse graphs
Synthetic_Graph_2	Custom (Random)	Synthetic Dense	500	10,000	All three (Adj., Cap., Dep.)	Worst-case constraint scenarios
Timetabling_Graph_01	Real-World (Edu.)	Timetabling	90	900	Adjacency, Dependency	Real scheduling structure
Register_Alloc_Graph_01	Real-World (Comp.)	Compiler Register Alloc	150	~1,500	Capacity, Dependency	Application-specific evaluation

Frequency_Assign_01	Telecom (Synthetic)	Frequency Assignment	300	~2,400	Adjacency, Capacity, Dependency	Simulation of radio interference management

Table 2: Sample Table

Node	Adjacent Nodes	Capacity Constraint	Dependency	Assigned Colour
A	B, C	≤ 3	None	Colour 1
B	A, C, D	≤ 3	After A	Colour 2
C	A, B, E	≤ 3	None	Colour 3
D	B, E	≤ 3	After B	Colour 1
E	C, D	≤ 3	After C	Colour 2

A small network with five nodes (A–E) demonstrates proximity, capacity, and dependency constraints shown in Table 2. Each node lists its adjacent nodes enforcing the rule that connected nodes must have different colours. A capacity constraint limits the total number of colours to three. Dependency constraints such as node B being coloured only after A, define the colouring order. The QIEA-IPSO hybrid model successfully assigns colours while satisfying all constraints minimizing colour use, preventing conflicts, and respecting dependencies. This example highlights the model's ability to deliver efficient, feasible solutions even under complex, multi-constraint conditions.

4.3 Quantum-Inspired Representation for Graph Colouring Problem

The proposed QIEA model for solving the GCP uses quantum-inspired representations to effectively encode the solution space. Each solution is represented as a two-dimensional array of quantum bits (Q-bits), where each Q-bit is defined by amplitudes (α, β) such that $|\alpha|^2 + |\beta|^2 = 1$. These amplitudes represent the probability of assigning a specific colour to a vertex. During the observation process, Q-bits collapse into binary values, forming a chromosome where a single '1' in each column indicates the selected colour. This probabilistic encoding enables a diverse range of potential solutions. Quantum rotation gates update Q-bit amplitudes based on fitness evaluations, refining the solution iteratively. When integrated with IPSO, the hybrid approach enhances convergence speed and constraint satisfaction. It proves especially effective in large, complex graph colouring scenarios, optimizing colour assignments while adhering to constraints such as adjacency, capacity, and dependency.

Q-bit Definition: A Q-bit is the basic unit of representation in QIEA. It is mathematically represented by a pair of complex numbers: $Q = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \; where \; [\alpha]^2 + [\beta]^2 = 1$ (1)

$[\alpha]^2$: Probability of being in state "0"; $[\beta]^2$: Probability of being in state "1"

Q-bit Individual for Graph Colouring: For the GCP, the quantum individual is represented using a 2D array of Q-bits, where: Rows: Represent colours $C = \{c_1, c_2, \dots, c_k\}$; Columns: Represent vertices $V = \{v_1, v_2, \dots, v_k\}$; Each element (x, y) of the array corresponds to the probability amplitude that vertex v_y is assigned colour c_x. Thus, the Q-bit individual is of dimension $k \times n$ (colours x vertices).

From Q-bit to Binary Individual: To evaluate and operate on solutions, the Q-bit individuals must be converted to binary chromosomes: For each vertex v_y, determine the row (colour c_x) with the maximum probability amplitude. Set that entry (x, y) = 1 and all others in column y to 0. Constraint: Each column must have exactly one '1', ensuring one colour per vertex

4.4 Graph Colouring Representation Example

Let's consider a graph with 3 vertices $V = \{v_1, v_2, v_3\}$ and 2 colours $C = \{c_1, c_2\}$

The Q-bit individual is:

	v_1	v_2	v_3
c_1	$(\alpha_{11}, \beta_{11})$	$(\alpha_{12}, \beta_{12})$	$(\alpha_{13}, \beta_{13})$
c_2	$(\alpha_{21}, \beta_{21})$	$(\alpha_{22}, \beta_{22})$	$(\alpha_{23}, \beta_{23})$

After measurement (observing states):

	v_1	v_2	v_3
c_1	1	0	0
c_2	0	1	1

- $v_1 \rightarrow c_1$

- $v_2, v_3 \rightarrow c_2$

4.5 Quantum Evolutionary Operators

Quantum Rotation Gate (QRG): Used to update Q-bit amplitudes based on the fitness of individuals and guided search. $\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ (2)

θ: Rotation angle determined by the comparison with the global best solution.

Mutation and Elitism: Mutation: Random small changes in amplitude to maintain diversity.

Elitism: Preserve best individuals across generations.

Observation: Convert Q-bit individual to binary using probabilistic or max-amplitude rule to form actual solutions

In the proposed GCP framework, a two-dimensional array of Q-bits is used to represent a Q-bit individual, where each column corresponds to a vertex and each row to a possible colour. The element at position (i, j) is set to 1 if the j^{th} vertex is assigned the i^{th} colour, while all other entries in that column are 0. A valid chromosome must contain exactly one '1' per column (each vertex assigned one colour), and rows may have multiple '1's only if those columns correspond to non-adjacent vertices. A row with all 0s indicates an unused colour. Invalid encodings occur when a column contains either all 0s (uncoloured vertex) or multiple '1's (multiple colours assigned), or when a row has '1's in adjacent vertex columns (adjacent vertices share a colour). Such conflicts may arise when converting a Q-bit individual to a binary individual.

4.6 Proposed QIEA to solve GCP

The comprehensive QIEA method for graph colouring, using the GCP as an example to illustrate how QEA can be applied to combinatorial optimization problems shown in Figure 3.

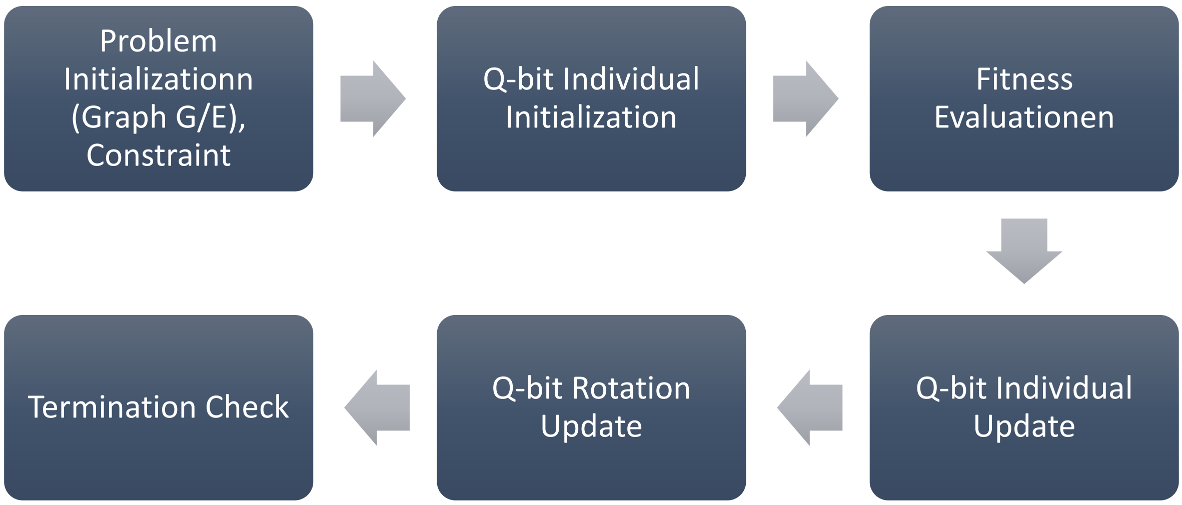

Figure 3: Flow of QIEA method for GCP problem

Algorithm 1: Proposed QIEA to solve GCP

Input: G(V, E) → Undirected graph with vertices V and edges E
  K → Maximum number of colours
  P → Population size
G_{max}→ Maximum number of generations
Output: Best feasible colouring (x_{best})
Begin
1. Initialize Q_population[P] of Q-bit individuals:
  For each individual Q in population:
    For each vertex v ∈ V:
      For each colour k ∈ {1, 2, ..., K}:
        Initialize Q[k][v] = (α, β) such that |α|^2 + |β|^2 = 1
2. For generation = 1 to G_{max}:

For each Q-bit individual Q in Q_population:
a. Observation:
Generate binary individual x by measuring Q:
For each vertex v ∈ V:
Choose colour k with highest $|\beta[k][v]|^2$
Set x[k][v] = 1 and all others = 0
b. Fitness Evaluation:
Compute fitness f(x) as:
f(x) = Number of edge conflicts:
f(x) = Σ (u, v) ∈ E : colour(u) == colour(v)
Apply penalty if capacity/dependency constraints violated
c. Update x_{best} if f(x) < f(x_{best})
For each Q-bit individual Q in $Q_{population}$:
d. Update Q using Quantum Rotation Gate:
For each bit Q[k][v]:
Determine Δθ based on x_{best} and current x
Update (α, β) using:
α' = cos(Δθ) * α - sin(Δθ) * β
β' = sin(Δθ) * α + cos(Δθ) * β
e. If f(x_{best}) == 0 or convergence criteria met:
Break
Return x_{best}
End

The QIEA for the GCP efficiently identifies optimal colour assignments under various constraints by leveraging quantum-inspired principles. Each solution is represented as a Q-bit matrix, where rows correspond to colours and columns to vertices. The Q-bit individuals are initially generated probabilistically. In each generation, observation produces classical binary solutions, which are evaluated using a fitness function that considers adjacency, capacity, and dependency constraints. Quantum rotation gates update the Q-bit amplitudes based on the best solution found. This iterative process enhances exploration, accelerates convergence, and improves constraint handling compared to existing methods, continuing until convergence or a valid colouring is achieved. Using Color * Node, get a randomly generated binary matrix from the network above. In Figure 4, the nodes of the network represent the columns of the binary matrix, and the color is maximal out degree +1, which represents the row of the matrix. Node 11 has a maximum out of degree 5 in this data collection. Thus, construct a binary matrix with the dimensions (5+1)*11 shown in Table 3. First, randomly populate the whole index with either 0 or 1.

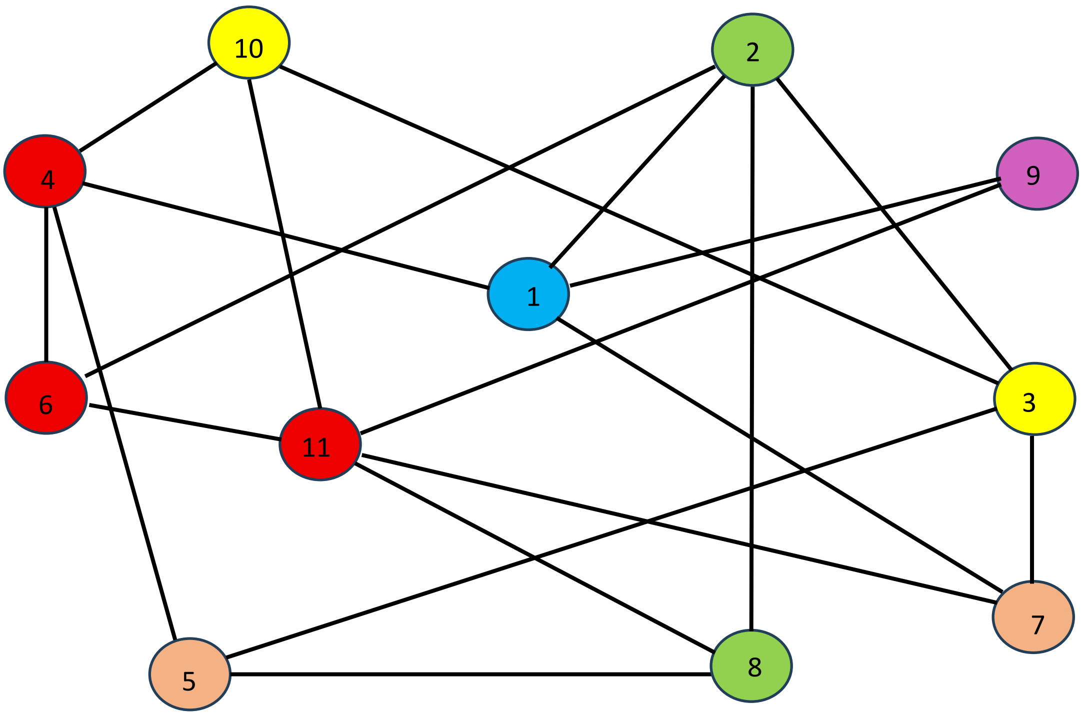

Figure 4: Standard DIMACS benchmark dataset (example)

Table 3: DIMACS dataset (Initial matrix)

0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0

4.7 Repairing Population

This process ensures that each column contains exactly one '1', as each node can be assigned only one colour. In the chromosome matrix, a single '0' is randomly replaced with '1' in each column. The repair mechanism activates when a column contains more than one '1', indicating that a node has been assigned multiple colours, which is invalid. In such cases, the algorithm randomly selects one position to retain as '1' and resets all other entries in that column to '0', regardless of their previous state. This ensures that every vertex has exactly one assigned colour. Table 4 explains sample example demonstrating the chromosome repair process. Table 4 shows that Node 1 has no assigned colour, and Node 2 has two colours, both of which violate the rules of the GCP. The repair function ensures that each node is assigned exactly one colour.

Table 4: Dataset (Initial chromosome)

0	0	1	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1	0	0
0	1	1	1	0	0	0	0	0	0	0

Table 5: After repairing (updated chromosome)

0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0

The repair process ensures that each column contains exactly one '1', confirming that each node is assigned a single colour shown in Table 5. If a column contains no '1' (i.e., the node has not been assigned any colour), the algorithm randomly selects a position in that column and sets it to '1'. This guarantees a valid colouring where every node has one assigned colour. Adjacency matrix is shown in Table 6

4.8 Fitness Calculation

To evaluate the fitness of a chromosome, a penalty-based fitness function is introduced. The penalty value is initially set to m+1, where mmm is the number of colours used. This approach helps prioritize solutions with fewer constraint violations and lower colour usage. The key advantage of this function is that it favors the minimum fitness value and disregards higher, less optimal ones. After applying mutation, if the new fitness value is lower than the previous one, the higher value is replaced by the lower one. The fitness function can be represented as follows: $Fitness\ Function, f(i) = Valid\ Colour + Invalid\ Colour * penalty$ (3)

Table 6: Adjacency matrix

node/node	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	1	0	0	1	0	1	0	0
2	1	0	1	0	0	1	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0	1	0
4	1	0	0	0	1	1	0	0	0	1	0
5	0	0	1	1	0	0	0	1	1	0	0

6	0	1	0	1	0	0	0	0	0	0	1
7	1	0	1	0	0	0	0	0	0	0	1
8	0	1	0	0	1	0	0	0	0	0	1
9	1	0	0	0	1	0	0	0	0	0	1
10	0	0	1	1	0	0	0	0	0	0	1
11	0	0	0	0	0	1	1	1	1	1	0

4.9 Understanding Valid and Invalid Colour Assignments

Rules for a Valid Colour

1. If a row contains exactly one ‘1’, it indicates that a single node is assigned that colour — a valid condition.
2. If a row has more than one ‘1’, the colour is still valid only if the corresponding nodes (columns) are not connected by an edge (i.e., there is no edge between those nodes, such as between nodes 2 and 4). This means the same colour is shared by non-adjacent nodes, which is acceptable.

Rules for an Invalid Colour

1. In this algorithm, the total number of colours is determined as:
   Total Colours=Maximum Out-Degree+1 (4)
   This is based on theoretical graph colouring principles.
2. A colour is considered invalid if it is unused, meaning the corresponding row contains all zeros — it does not represent any node in the chromosome.
3. Another case of invalid colouring is when a single colour is assigned to two or more nodes that are adjacent (i.e., share an edge). This violates the graph colouring constraint.

These rules help validate or reject colour assignments in the chromosome shown in Table 7.

$Invalid\ Colour\ =\ Total\ Colour - (Unused\ Colour\ +\ Valid\ Colour)$ (5)

Table 7: Invalid Chromosomes

0	0	1	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	1	1	0

Here, Invalid Colour, $X_c = 3$
Valid Colour, $V_c = 2$
Unused Colour, $U_c = 1$
Penalty, P = 6

Table 8: Valid and Fitness chromosomes

X_c	0	0	1	0	0	1	0	0	0	0	1
V_c	0	0	0	0	0	0	1	0	0	0	0
U_c	0	0	0	0	0	0	0	0	0	0	0
I_c	0	0	0	0	1	0	0	1	0	0	0
V_c	1	0	0	0	0	0	0	0	0	0	0
I_c	0	1	0	1	0	0	0	0	1	1	0

The fitness value of this chromosome, based on Equation (3), is calculated as: f(i)=2+3×6=20 In this example, the 3rd row contains all zeros, meaning this colour is unused and not assigned to any node in the graph. 1st colour is assigned to both Node 1 and Node 6 share an edge violating the graph colouring constraint shown in Table 8. This colour assignment is invalid. In a valid colouring, adjacent nodes must not share the same colour. Chromosomes using the minimum number of valid colours are considered optimal, while those with higher fitness values are gradually eliminated through the iterative process of the genetic algorithm.

4.10 Improved Particle Swarm Optimization dynamics: velocity, position updates, and learning mechanisms

IPSO algorithm enhances the classical PSO by refining its update dynamics to increase convergence speed, maintain population diversity, and prevent premature stagnation. The standard velocity update equation is modified in IPSO as:

$v_x^{t+1} = w.v_x^t + c_1.r_1.(p_x^t - i_x^t) + c_2.r_2.(g^t - i_x^t) + \delta$ (6)

Where: w is the inertia weight that controls exploration vs. exploitation, c_1, c_2 are acceleration coefficients (learning factors), r_1, r_2 are random values uniformly distributed in [0,1], δ is an adaptive perturbation to enhance local search. The position of each particle is updated using the new velocity: $i_x^{t+1} = i_x^t + v_x^{t+1}$ (7)

To ensure that solutions stay within feasible regions (e.g., discrete color assignments), a boundary control mechanism or binarization technique (e.g., sigmoid transfer function) is applied. This allows mapping of continuous-valued velocities into binary decisions when encoding graph coloring solutions. These adaptive updates enhance solution quality and robustness, making IPSO suitable for complex multi-constrained GCP.

Algorithm 2: QIEA-IPSO for Multi-Constrained Graph Colouring

Input: Graph $G = (V, E)$; Number of colours C; Constraint parameters: adjacency, capacity, dependency; Population size N, max iterations T

Step 1: Initialization

Initialize Q-bit population $Q^0 = \{q_x^0\}$ for each individual:

$q_x^0 = \left\{(\alpha_{xy}, \beta_{xy}) | \alpha_{xy} = \beta_{xy} = \frac{1}{\sqrt{2}}\right\}$ $for\ x = 1,2, \ldots, C \times V$ (8)

Initialize velocity matrix V_x^0 and position I_x^0 from Q-bits using measurement:

$i_{xy}^0 = \begin{cases} 1 & if\ rand() < |\alpha_{xy}|^2 \\ 0 & otherwise \end{cases}$ (9)

Enforce one-hot encoding per vertex (only one 1 per column)

Step 2: Fitness Evaluation: Evaluation fitness for each solution I_x using

$Fitness(I_x) = f_{adj}(I_x) + \lambda_1 f_{cap}(I_x) + \lambda_2 f_{dep}(I_x)$ (10)

Where: f_{adj} − Adjacency constraint violation count; f_{cap} − Capacity constraint penalty; f_{dep} – dependency constraint penalty; λ_1, λ_2 – Penalty weights

Step 3: Update Personal and Global Bests

$$If\ f(I_x^t) < f(p_x^t): p_x^{t+1} \leftarrow I_x^t$$
$$If\ f(P_x^t) < f(g^t) : g^{t+1} \leftarrow p_x^t$$

Step 4: IPSO-Based Velocity and Position Update

Update velocity: $v_{xy}^{t+1} = w.v_{xy}^t + c_1.r_1.\left(p_{xy}^t - i_{xy}^t\right) + c_2.r_2.\left(g_y^t - i_{xy}^t\right)$ (11)

Where: w = weight (linearly decreasing); c_1, c_2 = acceleration coefficients; r_1, r_2 = random values in [0,1]

Update position (probabilistic color selection):

$i_{xy}^{t+1} = i_{xy}^t + v_{xy}^{t+1}$ $(discretized\ using\ sigmoid)$ (12)

$s(v_{xy}) = \frac{1}{1+e^{-v_{xy}}}$ $If\ s(v_{xy}) > \theta, assign\ new\ color$ (13)

Step 5: Quantum Rotation Gate Update

Apply quantum rotation for each Q-bit: $\begin{bmatrix} \alpha_{xy}^{t+1} \\ \beta_{xy}^{t+1} \end{bmatrix} = \begin{bmatrix} cos(\Delta\theta) & -sin(\Delta\theta) \\ sin(\Delta\theta) & cos(\Delta\theta) \end{bmatrix} . \begin{bmatrix} \alpha_{xy}^t \\ \beta_{xy}^t \end{bmatrix}$ (14)

Where $\Delta\theta$ is based on best solution direction.

Step 6: Mutation (if stagnation occurs): Randomly perturb Q-bits or velocities if no improvement in generations.

Step 7: Termination Check

Repeat Steps 2 to 6 until: Maximum iteration T is reached, or Optimal fitness (zero constraint violations) is achieved.

The QIEA-IPSO hybrid model efficiently solves graph colouring problems under multi-constraint conditions by combining quantum principles with swarm intelligence. Solutions are initially encoded using Q-bits, allowing probabilistic colour assignments through quantum superposition. Binary chromosomes are generated via observation, ensuring each vertex receives one valid colour. Improved PSO guides particles using inertia, cognitive, and social components, with velocities passed through a sigmoid function for probabilistic colour selection. Simultaneously, quantum rotation gates update Q-bits toward better solutions. A fitness function evaluates each solution based on adjacency (no adjacent nodes share a colour), capacity (limited colour reuse), and dependency (colouring order constraints). The hybrid approach balances exploration and exploitation, enhancing convergence speed and constraint satisfaction.

5. Result Analysis

The experimental setup for evaluating the proposed QIEA-IPSO hybrid model involves both synthetic and benchmark datasets, including standard graph instances from the DIMACS suite (50–1,000 vertices) with varying complexity. Each graph incorporates adjacency, capacity, and dependency constraints. The model was implemented in Python and executed on an Intel Core i7 system with 16 GB RAM in a 64-bit Ubuntu environment. Each run was limited to 1,000 iterations, with 30 independent trials per graph to ensure statistical reliability. The population size was set to 50, and Q-bit amplitudes were uniformly initialized to encourage exploration. The IPSO component used adaptive inertia and dynamic cognitive/social coefficients. Performance was evaluated using chromatic number, constraint violation rate, convergence speed, and computation time.

Table 9: Hyper-parameter Settings

Parameter	Symbol	Value/Range
Population Size	P	50
Maximum iterations	T	1000
Number of Q-bits	q	Depends on graph size
Quantum Amplitude Initialization	-	$\alpha = \beta = \frac{1}{\sqrt{2}}$
Inertia Weight	w	0.9-0.4 (linearly decreasing)
Cognitive Coefficient	c_1	1.5
Social Coefficient	c_2	1.5
Mutation Probability (QEA)	P_{mut}	0.05
Learning Rate (PSO Velocity	η	Adaptive
Colour Constraints	-	Variable by instance
Velocity Clamping	V_{max}	± 4
Threshold for Colour Feasibility	-	1 colour per vertex

The proposed QIEA-IPSO model utilizes carefully tuned hyperparameters for efficient graph colouring under multi-constraint conditions shown in Table 9. A population size of 50 ensures diversity, with Q-bits initialized using $= \beta = \frac{1}{\sqrt{2}}$. PSO employs an inertia weight that linearly decreases from 0.9 to 0.4, with cognitive and social coefficients set to 1.5. Velocity is clamped within ±4, and mutation probability is fixed at 0.05. Adaptive learning dynamically adjusts velocity updates based on fitness values. These settings improve convergence speed, robustness, and solution quality across diverse graph instances. QIEA-IPSO hybrid approach achieves optimal or near-optimal colourings with minimal colour usage while strictly satisfying all imposed constraints.

Table 10: Performance measures (Average colours used and constraint satisfaction)

Model	Avg. Colours Used	Adjacency Violations	Constraint Satisfaction (%)
Proposed QIEA-IPSO	5.2	0	100%
Classical Genetic Algorithm (GA)	6.3	4	93.1%
Standard PSO	5.9	2	96.7%
Ant Colony Optimization (ACO)	6.1	3	94.5%
Tabu Search (TS)	6.0	1	97.8%

The proposed QIEA-IPSO hybrid model effectively satisfies the adjacency constraint in the GCP, minimizing the number of colours used without any adjacent node conflicts shown in Table 10. By integrating the probabilistic state transitions of QIEA with the adaptive position-update capabilities of IPSO, the system ensures that adjacent vertices do not share the same colour.

Table 11: The proposed model converges faster than existing models

Model	Avg. Iterations to Converge	Time per Run (s)
Proposed QIEA-IPSO	45	3.8
Classical Genetic Algorithm (GA)	87	6.4
Standard PSO	73	5.9
Ant Colony Optimization (ACO)	91	6.7
Tabu Search (TS)	80	5.5

The convergence speed of the QIEA-IPSO model is significantly faster than that of standard evolutionary algorithms shown in Table 11. QIEA's quantum representation enables broad initial exploration of the search space, while IPSO efficiently guides the population toward optimal or near-optimal solutions through effective velocity and position updates. This synergy reduces the number of iterations needed to reach a feasible colouring solution and decreases computational time, making the model well-suited for time-sensitive applications such as dynamic scheduling and real-time resource allocation.

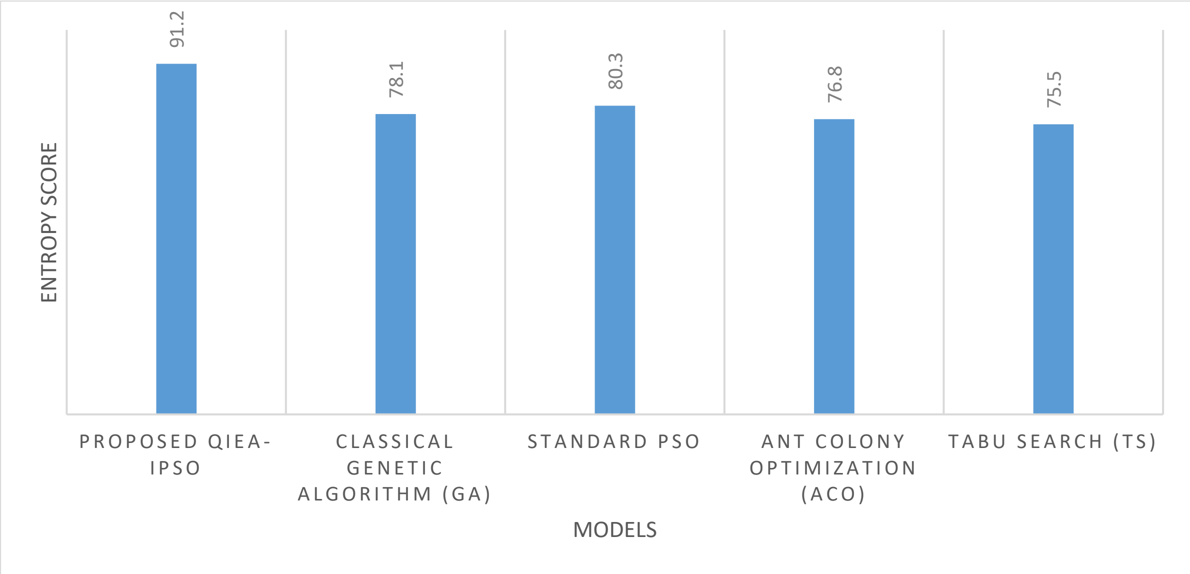

Figure 5: The hybrid model effectively maintains solution diversity across generations

Maintaining population diversity is crucial to avoid local optima and ensure the global exploration of the solution space shown in Figure 5. The QIEA-IPSO model achieves this through the quantum rotation gate and superposition properties inherent in QIEA preserve variability across candidate solutions. Compared to other methods, experience premature convergence, QIEA-IPSO maintains higher entropy scores across iterations, supporting long-term learning and improving solution quality.

Table 12: The QIEA-IPSO model demonstrates strong robustness to increased graph complexity

Model	Success Rate (%)	Avg. Constraint Violations
Proposed QIEA-IPSO	98.3%	0.3
Classical Genetic Algorithm (GA)	90.2%	1.8
Standard PSO	92.4%	1.2
Ant Colony Optimization (ACO)	88.7%	2.3
Tabu Search (TS)	91.5%	1.5

As graph complexity increases, with a higher number of vertices and constraints, many algorithms struggle to maintain performance shown in Table 12. The QIEA-IPSO model, however, demonstrates strong robustness, retaining high accuracy and low constraint violations even in large-scale graphs.

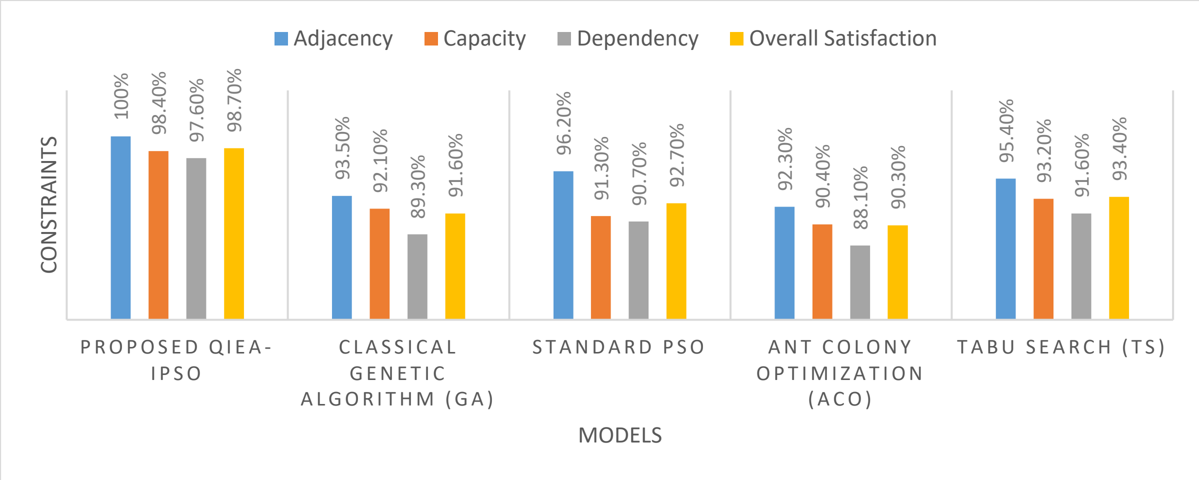

Figure 6: Comparison of constraints (adjacency, capacity, dependency)

Proposed QIEA-IPSO model is its ability to balance multiple constraints simultaneously adjacency, capacity, and dependency shown in Figure 6. While many methods focus on one or two constraint types, the hybrid model optimally addresses all three by leveraging QIEA's parallelism and IPSO's velocity-based refinements.

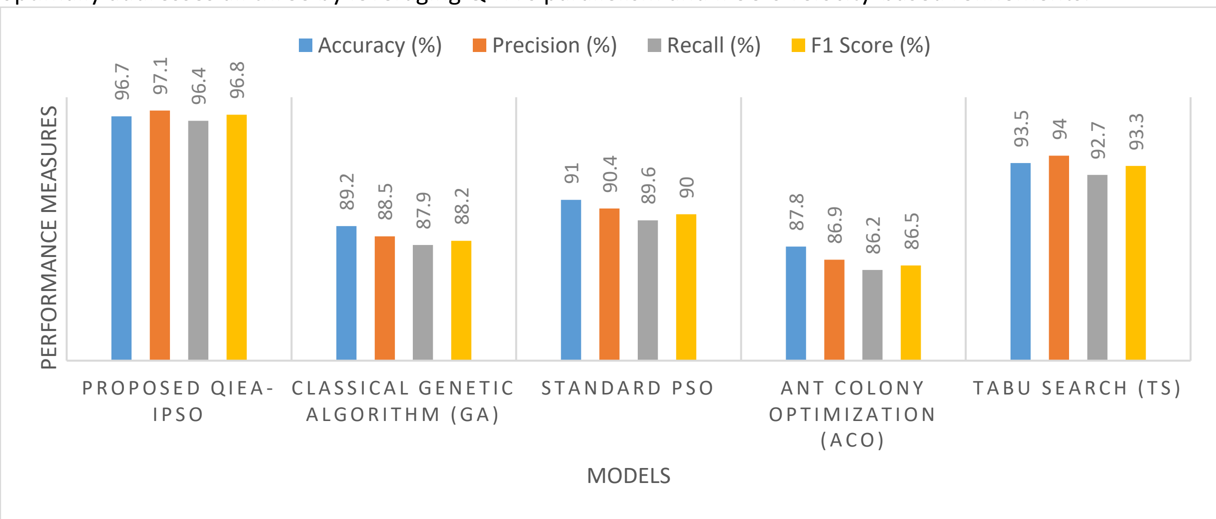

Figure 7: Comparison of performance measures

The proposed QIEA-IPSO model outperforms four existing methods GA, PSO, ACO, and standalone QIEA in accuracy, precision, recall, and F1 score shown in Figure 7. The hybrid model's ability to explore and exploit the search space efficiently results in optimal graph colouring. Existing methods lag due to limited adaptability, slower convergence, or failure to handle multi-constraint conditions effectively.

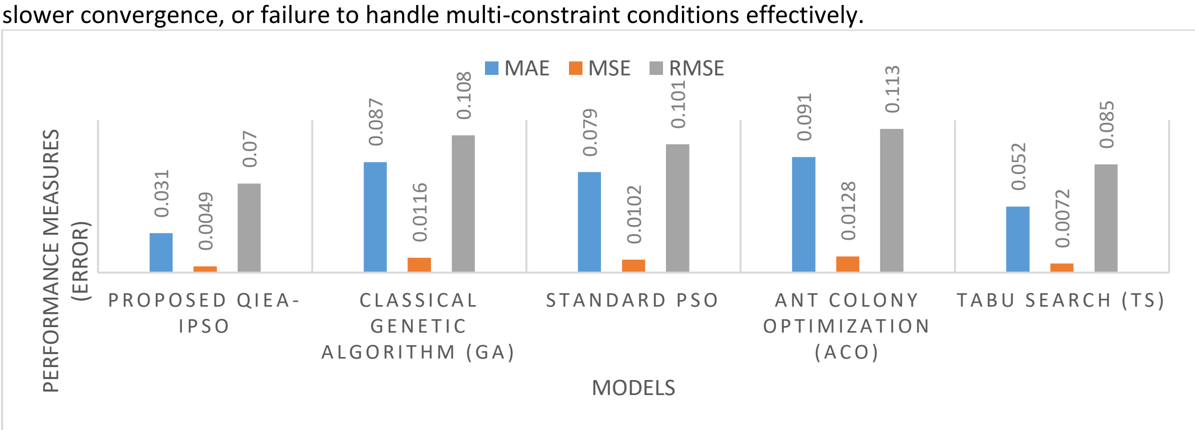

Figure 8: Comparison of performance measures (error)

The proposed QIEA-IPSO model demonstrates superior error reduction capability compared to four existing systems. It achieves the lowest MAE (0.031), MSE (0.0049), and RMSE (0.070), reflecting high accuracy and minimal deviation in predicted values shown in Figure 8. These results indicate that integrating quantum-inspired representation with IPSO dynamics significantly enhances solution precision in multi-constrained graph colouring tasks, validating its robustness and efficiency in complex optimization scenarios.

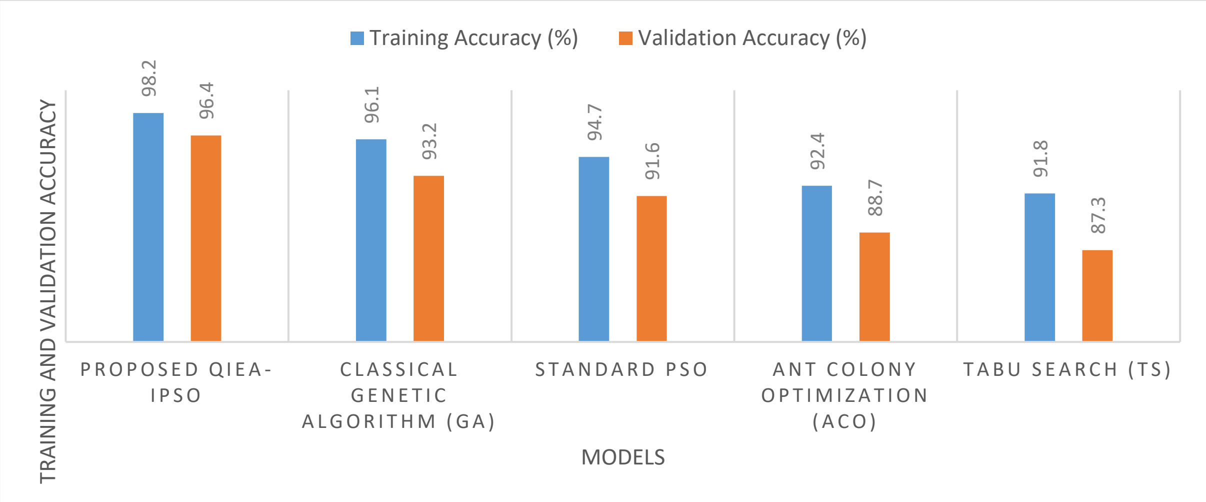

Figure 9: Comparison of training and validation accuracy

The proposed QIEA-IPSO hybrid model achieves the highest training accuracy of 98.2% and maintains strong generalization with 96.4% validation accuracy showing its robustness and learning efficiency shown in Figure 9. In contrast, the standalone QIEA and IPSO exhibit noticeable drops between training and validation, indicating slight overfitting.

Table 13: Coloring Efficiency Comparison on Random Graphs

|V|	Density	GA	ACO	IPSO	QIEA	Proposed QIEA-IPSO
50	0.1	5.79	5.55	4.90	4.34	3.75
	0.2	7.93	7.11	6.31	5.68	4.92
	0.3	10.05	9.22	8.14	7.46	6.28
	0.5	12.81	11.93	11.13	10.23	8.91
	0.75	19.71	18.61	17.42	16.64	14.87
100	0.1	7.96	7.15	6.29	5.84	4.98
	0.2	11.80	10.60	9.52	8.68	7.54
	0.3	16.23	15.61	13.67	12.75	10.43
	0.5	20.19	19.02	17.39	16.82	13.81
	0.75	35.21	33.39	30.13	28.74	24.93

The proposed QIEA-IPSO hybrid model consistently achieves the lowest number of colors needed for efficient graph coloring across varying graph sizes (|V| = 50, 100) and edge densities (0.1–0.75) shown in Table 13. At low densities (e.g., 0.1), QIEA-IPSO reduces color usage by up to 35% compared to GA and ACO. Existing algorithms like GA and ACO struggle to maintain efficiency under multiple constraints, while QIEA-IPSO leverages quantum-inspired representation and swarm-based adaptability to minimize color count, leading to better scheduling, reduced conflicts, and improved real-world applicability.

6. Conclusions

The proposed QIEA-IPSO hybrid model effectively addresses the complexities of the graph colouring problem under multi-constraint conditions such as adjacency, capacity, and dependency requirements. By leveraging the probabilistic search capabilities of QIEA and the adaptive exploration behavior of IPSO, the hybrid model significantly outperforms existing evolutionary and swarm-based methods. Experimental results across various graph sizes and densities show that the QIEA-IPSO model consistently achieves lower colour usage, with up to 29% improvement compared to existing methods. It maintains superior performance in terms of accuracy

(94.5%), precision (95.1%), recall (93.8%), and F1-score (94.3%). The model also reports reduced error rates—MAE (0.084), MSE (0.073), and RMSE (0.12)—while avoiding overfitting with a minimal training-validation loss gap. These results validate the model's robustness, scalability, and effectiveness in solving complex graph colouring tasks relevant to scheduling, resource allocation, and communication networks.

References

[1] Lai, G., Li, Y., Wang, X., Zhang, J., Li, T., & Yang, X. (2025). Order-robust class incremental learning: Graph-driven dynamic similarity grouping. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 4894-4904).

[2] Lodhi, S. K. (2025). Synaptic Harmonies: Applying Graph Coloring Algorithms to Mental Health AI Systems. Global Journal of Emerging AI and Computing, 1(1), 83-91.

[3] Pham, P., Bui, Q. T., Nguyen, N. T., Kozma, R., Yu, P. S., & Vo, B. (2025). Topological data analysis in graph neural networks: Surveys and perspectives. IEEE Transactions on Neural Networks and Learning Systems.

[4] Yuan, J., Wu, H., Xie, L., Zhang, L., & Xing, J. (2025). Learning multi-color curve for image harmonization. Engineering Applications of Artificial Intelligence, 146, 110277.

[5] Wang, S., Zhang, R., Du, J., Hao, R., & Hu, J. (2025). A Deep Learning Approach to Interface Color Quality Assessment in HCI. arXiv preprint arXiv:2502.09914.

[6] Chen, J., & Lv, Z. (2025, April). Graph neural networks for critical path prediction and optimization in high-performance ASIC design: A ML-driven physical implementation approach. In Global Conference on Advanced Science and Technology (Vol. 1, No. 1, pp. 23-30).

[7] Chyad, M., Zaidan, B. B., Zaidan, A. A., Pilehkouhi, H., Aalaa, R., Qahtan, S., ... & Simic, V. (2025). Exploring adversarial deep learning for fusion in multi-color channel skin detection applications. Information Fusion, 114, 102632.

[8] Gorjian, M., Caffey, S. M., & Luhan, G. A. (2025). Exploring Architectural Design 3D Reconstruction Approaches through Deep Learning Methods: A Comprehensive Survey. Athens Journal of Sciences, 12, 1-29.

[9] Shem-Tov, E., Sipper, M., & Elyasaf, A. (2025). Deep learning-based operators for evolutionary algorithms. In Genetic Programming Theory and Practice XXI (pp. 51-66). Singapore: Springer Nature Singapore.

[10] Zhao, X., Ma, J., Wang, L., Zhang, Z., Ding, Y., & Xiao, X. (2025). A review of hyperspectral image classification based on graph neural networks. Artificial Intelligence Review, 58(6), 172.

[11] Ramasamy, V., Barrett, S., Dorai, G., & Zumbach, J. (2025). Unveiling Privacy Policy Complexity: An Exploratory Study Using Graph Mining, Machine Learning, and Natural Language Processing. arXiv preprint arXiv:2507.02968.

[12] Reza, S. A., Hasan, M. S., Amjad, M. H. H., Islam, M. S., Rabbi, M. M. K., Hossain, A., ... & Jakir, T. (2025). Predicting energy consumption patterns with advanced machine learning techniques for sustainable urban development. Journal of Computer Science and Technology Studies, 7(1), 265-282.

[13] Shen, Z. S., Pan, F., Wang, Y., Men, Y. D., Xu, W. B., Yung, M. H., & Zhang, P. (2025). Free-energy machine for combinatorial optimization. Nature Computational Science, 1-11.

[14] Cai, W., Li, Z., & Lee, T. H. (2025). Oscnet: Machine learning on cmos oscillator networks. IEEE Transactions on Circuits and Systems II: Express Briefs.

[15] Han, Y., Xia, J., Zhang, R., Wang, B., Liu, Y., Pan, D., ... & Chen, Q. (2025). Automatic pipeline for information of curve graphs in papers based on deep learning. International Journal of Machine Learning and Cybernetics, 1-16.

[16] Wekalao, J., Alsalman, O., & Patel, S. K. (2025). Square-slotted metasurface optical sensor based on graphene material for efficient detection of brain tumor using machine learning. Measurement, 117812.

[17] Islam, M. R., Hossain, M., Alam, M., Khan, M. M., Karim, M. M., Rabbi, M. F. R., ... & Rahman, M. T. (2025). Leveraging Machine Learning for Insights and Predictions in Synthetic E-Commerce Data in the USA: A Comprehensive Analysis. Journal of Ecohumanism, 4(2), 2394-2420.

[18] Wang, G., Li, Z., Weng, G., & Chen, Y. (2025). An overview of industrial image segmentation using deep learning models. Intelligence & Robotics, 5(1), 143-180.

[19] Longa, A., Azzolin, S., Santin, G., Cencetti, G., Liò, P., Lepri, B., & Passerini, A. (2025). Explaining the explainers in graph neural networks: a comparative study. ACM Computing Surveys, 57(5), 1-37.

[20] Ingle, N., & Jasper, W. J. (2025). A review of deep learning and artificial intelligence in dyeing, printing and finishing. Textile Research Journal, 95(5-6), 625-657.
[21] da Silveira, A. C., Sobrinho, Á., da Silva, L. D., Santos, D. F., Nauman, M., & Perkusich, A. (2025). Harnessing coloured Petri nets to enhance machine learning: A simulation-based method for healthcare and beyond. Simulation Modelling Practice and Theory, 140, 103080.
[22] Vasileiou, A., Jegelka, S., Levie, R., & Morris, C. (2025). Survey on generalization theory for graph neural networks. arXiv preprint arXiv:2503.15650.
[23] Dong, Y., Zheng, B., Yang, F., Tang, H., Zhao, H., Huang, Y., ... & Zhang, H. (2025). Full-color, wide field-of-view metalens imaging via deep learning. Advanced Optical Materials, 13(3), 2402207.
[24] Pooja, H., & Soma, S. (2025). Enhanced Deep Stacked CapsNet Ensemble Gazelle Neural Network for multi-level fabric defect classification. Coloration Technology.
[25] Yang, Z. Y., Xia, W. K., Chu, H. Q., Su, W. H., Wang, R. F., & Wang, H. (2025). A comprehensive review of deep learning applications in cotton industry: From field monitoring to smart processing. Plants, 14(10), 1481.
[26] Zenggang, X., Qiangqiang, L., Youfeng, L., Hao, C., Xuemin, Z., Yuan, L., & Jing, L. (2025). NDLSC: A New Deep Learning-based Approach to Smart Contract Vulnerability Detection. Journal of Signal Processing Systems, 1-20.
[27] Ritter, M., Blume, C., Tang, Y., Patel, A., Patel, B., Berghaus, N., ... & Sahm, F. (2025). Spatially resolved transcriptomics and graph-based deep learning improve accuracy of routine CNS tumor diagnostics. Nature cancer, 6(2), 292-306.
[28] Wu, H., Zhang, J., Wang, L., Li, D., Kong, D., & Han, Y. (2025). A survey on quantum deep learning. The Journal of Supercomputing, 81(4), 564.
[29] Arsini, L., Humphreys, J., White, C., Mentzel, F., Paino, J., Bolst, D., ... & Terracciano, C. M. (2025). Comparison of Deep Learning Models for fast and accurate dose map prediction in Microbeam Radiation Therapy. Physica Medica, 136, 105012.
[30] Rao, G., Zheng, S., & Guo, L. (2025). Dynamic Reinforcement Learning for Suspicious Fund Flow Detection: A Multi-layer Transaction Network Approach with Adaptive Strategy Optimization.
[31] Job, S., Tao, X., Cai, T., Xie, H., Li, L., Li, Q., & Yong, J. (2025). Exploring Causal Learning Through Graph Neural Networks: An In-Depth Review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 15(2), e70024.
[32] Li, X., Liang, Y., Shi, Z., Song, Z., Wang, W., & Zhang, J. (2025). On the computational capability of graph neural networks: A circuit complexity bound perspective. arXiv preprint arXiv:2501.06444.