QUANTUM-INSPIRED GENOTYPES FOR THE EVOLUTION OF GRAPH DYNAMICAL SYSTEMS

Peter David Fagan
School of Informatics
University of Edinburgh
United Kingdom
p.d.fagan@ed.ac.uk

ABSTRACT

We introduce a conceptual framework for evolvable dynamical systems based on quantum-inspired genotypes (QIGs): continuous latent representations that collapse into structured binary encodings of graph topology and node-level dynamics. Unlike traditional genetic encodings, QIGs enable smooth crossover, structured mutation, and entropy-aware exploration by operating over a superposition-inspired latent space. We define a collapse operator that maps these latent vectors into executable system representations, and outline how this encoding can be optimized via classical evolutionary search or quadratic unconstrained binary optimization (QUBO) for potential quantum annealing. The current version of this work is presented as an architectural proposal without experimental validation, with the aim of inspiring further research on structure-evolving systems and hybrid continuous–discrete optimization frameworks.

Note: This is a conceptual preprint intended to share an early-stage/evolving idea. It has not undergone peer review and does not include experimental validation. The framework described herein is theoretical and presented to encourage discussion, feedback, and potential future development.

1 Introduction

Over the past decade, neural networks trained via backpropagation have demonstrated remarkable capabilities across perception, control, and reasoning tasks. Their success has redefined the frontier of artificial intelligence. Yet despite these achievements, such systems diverge sharply from biological intelligence in key respects. Unlike their natural counterparts, artificial neural networks rely on supervised training, external error signals, and finely tuned hyperparameters. They are not truly autonomous learners, nor do they readily adapt their internal structure or dynamics over time. This disconnect has sparked growing interest in alternative paradigms—systems that learn not through static architectures and gradient flow, but through open-ended adaptation of their very form.

Historically, evolutionary approaches have offered one such paradigm. Genetic encodings, neuroevolution, and topology optimization techniques have been used to design agents with adaptive, task-specific behaviors. However, these methods often suffer from brittle discrete representations, limited crossover mechanisms, and a lack of structured exploration in high-dimensional genotype spaces. Efforts to encode both the topology and dynamics of evolving systems frequently run into the curse of dimensionality and search-space fragmentation. As a result, the potential of evolvable dynamical systems remains under-explored.

In this work, we introduce Quantum-Inspired Genotypes (QIGs) — continuous latent representations that collapse into structured binary encodings of graph dynamical systems. By operating in a latent superposition space, QIGs enable smooth crossover, structured mutation, and entropy-aware exploration — all of which are difficult to achieve with traditional discrete encodings. We define a collapse operator that maps latent genotypes to concrete systems, encoding both graph topology and node-level dynamics. This approach opens the door to scalable, flexible, and potentially quantum-native evolutionary systems that can learn by evolving both their structure and behavior.

Our contributions are as follows:

- We introduce a novel representation — Quantum-Inspired Genotypes — for encoding evolvable dynamical systems in continuous latent space.
- We define a deterministic collapse operator that maps these genotypes to structured binary codes representing both topology and dynamics.
- We discuss integration with quantum-inspired optimization techniques, such as QUBO solvers, and outline potential extensions to quantum-native implementations.
- We position this work as a foundation for future research on structure-evolving intelligent systems capable of autonomous adaptation.

By bridging ideas from evolutionary computation, dynamical systems, and quantum representation theory, this work contributes a new building block to the literature on adaptive learning systems. It offers a representation that is not only expressive and structured, but inherently compatible with both classical and emerging quantum optimization methods — charting a path toward systems that learn by evolving themselves.

2 Background and Related Work

2.1 Genetic Encodings for Evolutionary Algorithms

Evolutionary algorithms (EAs) are optimization frameworks inspired by biological evolution, where candidate solutions — often referred to as genotypes — are iteratively evolved through variation and selection. The efficacy of an EA is heavily influenced by the structure of its genetic encoding: how solutions are represented, mutated, and recombined.Traditional genetic encodings fall into three primary categories: binary encodings, real-valued encodings, and structured encodings (e.g. trees or graphs).

Binary encodings, popularized by early genetic algorithms [1], represent solutions as fixed-length bitstrings and apply simple crossover and mutation operators. While computationally efficient, these encodings can be brittle in high-dimensional or structured problem spaces, where small mutations often lead to large semantic changes in the decoded phenotype.

Real-valued encodings [2], commonly used in evolution strategies and continuous optimization, enable smooth search in continuous domains. However, without additional structure or constraints, they often struggle to represent or evolve complex, discrete entities such as network topologies or program syntax — where relational or combinatorial structures are key.

Structured encodings address the limits of flat genotypes by representing solutions as graphs, grammars, or modular constructs. Approaches like Cartesian Genetic Programming (CGP) [3], Grammatical Evolution [4], and NEAT [5] have successfully evolved neural networks, control policies, and circuits. While expressive, these methods typically rely on handcrafted variation operators and yield non-reversible, tree-like search trajectories — making it difficult to recombine or revisit past configurations. This brittleness hinders long-term exploration: structural mutations often destabilize behavior or destroy functional components, especially in complex, tightly coupled systems.

A central challenge across all encodings is balancing explorability (the capacity to discover novel solutions) with structural integrity (the preservation of useful components). Poorly calibrated representations create rugged or degenerate search landscapes, impeding convergence. In this work, we introduce a new class of encoding — *quantum-inspired genotypes* (QIGs) — that operates in a continuous latent space and deterministically collapses into discrete representations of graph dynamical systems. This representation offers a novel trade-off between smooth exploration and precise structural expressiveness, opening new directions for evolvable computation.

2.2 Evolution of Structured / Graph Systems

Many real-world systems — from neural networks and control circuits to social and biological networks — exhibit structure that is naturally modeled as graphs. As such, the evolution of structured or graph-based systems has become a key area of focus in evolutionary computation. Unlike vector-based encodings, graph representations allow the modeling of both topology (how components are connected) and structured dynamics, making them particularly suited to domains requiring spatial, modular, or causal structure.

Early efforts in this direction include approaches like NeuroEvolution of Augmenting Topologies (NEAT) [5], which evolve both the weights and the architecture of neural networks by incrementally complexifying topology. Other work, such as Cartesian Genetic Programming (CGP) [3], encodes graphs in a fixed grid, using a linear genome to represent node connections and functions. These methods demonstrated that evolving graph topology, not just parameters, can lead to significantly more expressive and adaptive systems.

More recent techniques have explored graph grammar-based encodings, modular neural networks, and indirect encodings such as HyperNEAT [6], where the genotype encodes a function that generates the phenotype network. These approaches aim to capture regularities and reuse within the structure — a key feature of biological systems — while improving scalability. However, such systems often require handcrafted genotype-to-phenotype mappings and struggle to balance exploration of novel topologies with preservation of functional substructures.

A persistent challenge in this field is the representation of graph structure in a form amenable to evolutionary operations. Crossover and mutation on raw adjacency matrices or connection lists can be destructive or semantically meaningless. Researchers have proposed various solutions, including encoding schemes with locality preservation, hierarchical composition, or constraint enforcement, yet these typically come with trade-offs in generality or flexibility.

Our proposed framework contributes to this area by introducing a continuous latent representation that can be smoothly explored, while collapsing into a discrete graph structure with associated dynamical properties. This allows for structured exploration of graph space without sacrificing the precision needed to define valid and meaningful systems — a key step toward more robust evolution of complex, structured dynamical systems.

2.3 Quantum-Inspired Representations and Evolutionary Algorithms

Quantum superposition enables the encoding of a system's state as a linear combination of multiple basis states, allowing for compact representation of exponentially many configurations. In classical computing, a bit exists in a definite state: either 0 or 1. A quantum bit (qubit), by contrast, exists in a superposed state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where α and β are complex amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. This formulation enables parallel evaluation over state space and supports probabilistic reasoning in optimization contexts.

Quantum-inspired evolutionary algorithms (QIEAs) [7] adopt this principle by encoding candidate solutions as probabilistic genotypes. Each bit is represented by a qubit-like pair of amplitudes:

$$Q = \left\{ \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \right\}_{i=1}^{n}, \tag{2}$$

where each (α_i, β_i) pair defines a probability distribution over binary values. During evolution, variation operators analogous to quantum rotation gates are used to update these amplitudes based on fitness feedback. Sampling from the resulting distributions generates candidate solutions through a stochastic collapse process. QIEAs have been successfully applied to a variety of combinatorial problems, including knapsack optimization, scheduling, and satisfiability. Their strength lies in balancing early-stage exploration (via high entropy distributions) with late-stage convergence through adaptive amplitude updates. However, existing QIEA methods typically operate on flat binary strings with shallow, task-specific genotype–phenotype mappings. These encodings often lack the expressive capacity to generate structured systems such as graphs or coupled dynamical models. In contrast, our approach retains a flat binary genotype but introduces a flexible collapse mechanism — deterministic or stochastic — that maps to expressive, structured phenotypes. This enables the evolution of rich, modular systems while preserving compatibility with QUBO-based optimization and other binary search strategies.

In this work, we extend the principles of quantum-inspired encoding to develop *quantum-inspired genotypes* (QIGs): continuous latent representations that deterministically collapse into discrete graph dynamical systems. Each QIG encodes a distribution over system topologies and node-level parameters, which are resolved through a structured collapse operator. This framework preserves the smooth search benefits of QIEAs while enabling the generation of coherent, executable systems with non-trivial structure and dynamics. By bridging latent probabilistic search with deterministic structural instantiation, QIGs offer a new foundation for evolving modular, adaptable, and interpretable computational systems.

3 Quantum-Inspired Genotypes for the Evolution of Graph Dynamical Systems

3.1 Graph Dynamical System

We define a *graph dynamical system* (GDS) as a discrete-time system whose evolution is governed by both the topology of an underlying graph and the dynamics associated with its nodes. Formally, a GDS is specified by a tuple:

$$\mathcal{G} = (V, E, \mathcal{F}, \Theta), \tag{3}$$

where:

- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes.
- $E \subseteq V \times V$ is the set of directed or undirected edges defining the system's topology.
- $f_i \in \mathcal{F}$ is a function drawn from a predefined set of candidate node dynamics.
- $\theta_i \in \mathbb{R}^k$ is a parameter vector specific to f_i, determining the node's dynamical response.

Let $\mathcal{N}_i = \{v_j \mid (v_j, v_i) \in E\}$ denote the set of in-neighbors of node v_i. At each discrete timestep t, the state of the system is described by a vector $\mathbf{x}^{(t)} = [x_1^{(t)}, \ldots, x_n^{(t)}]$. The update dynamics proceed according to:

$$x_i^{(t+1)} = f_i\left(x_i^{(t)}, \{x_j^{(t)} \mid v_j \in \mathcal{N}_i\}, \theta_i\right). \tag{4}$$

That is, the next state of each node is determined by its current state, the states of its neighbors, and its internal parameters.

This formulation is flexible enough to describe a broad class of discrete-time dynamical systems defined over networked structures. Importantly, both the topology E and the set of update rules F can be learned or evolved — enabling expressive and adaptive models of computation and behavior.

In the context of this work, we restrict our attention to graph dynamical systems where f_i is selected from a parametric function class. This aligns with the goal of evolving structured systems whose connectivity and local dynamics jointly define global behavior.

3.2 Graph Dynamical System Encoding

To represent a graph dynamical system (GDS), we define a compact phenotype composed of three components:

- A binary adjacency matrix $A \in \{0, 1\}^{n \times n}$, which specifies the presence of directed edges between nodes.
- A coupling matrix $W \in \mathbb{R}^{n \times n}$ or a quantized variant, assigning interaction strengths to each edge. These values modulate the influence of neighboring nodes during state updates and are masked by A.
- A binary parameter matrix $P \in \{0, 1\}^{n \times k}$, where each row defines the configuration of the local update function f_i, including discrete function types and associated parameters.

We denote the full system encoding as:

$$\Gamma = (A, W, P),$$

where Γ defines the complete structure and local behavior of a graph dynamical system. In our framework, this phenotype is generated by collapsing a latent continuous vector $\mathbf{z} \in \mathbb{R}^{d \times 2}$, referred to as a *quantum-inspired genotype* (QIG), via a structured decoding process.

3.3 Collapse Operator

The *collapse operator* $\mathcal{C} : \mathbb{R}^{d \times 2} \to \Gamma$ is the core mechanism that maps a latent quantum-inspired genotype (QIG) $\mathbf{z} \in \mathbb{R}^{d \times 2}$ into a valid graph dynamical system encoding $\Gamma = (A, W, P)$. Each of the d components in the binary system representation is associated with a latent pair (α_i, β_i), representing a continuous superposition over binary outcomes. This process draws inspiration from wavefunction collapse in quantum mechanics, transforming a smooth latent code into a discrete, executable structure.

Latent Partitioning. The latent vector is partitioned into subcomponents:

- $\mathbf{z}^{(e)} \in \mathbb{R}^{n \times n}$: edge presence scores.
- $\mathbf{z}^{(w)} \in \mathbb{R}^{n \times n}$: raw edge weights.
- $\mathbf{z}^{(p)} \in \mathbb{R}^{n \times k}$: node-level parameter logits.

Collapse Variants. The collapse may proceed via either deterministic thresholding or stochastic sampling:

- *Deterministic:*

$$A_{ij} = \mathbb{I}[z_{ij}^{(e)} > \tau_e], \quad W_{ij} = \text{quantize}(z_{ij}^{(w)}) \cdot A_{ij}, \quad P_{i,l} = \mathbb{I}[z_{i,l}^{(p)} > \tau_p]$$

- *Stochastic:*

$$A_{ij} \sim \text{Bernoulli}(\sigma(\beta z_{ij}^{(e)})), \quad P_{i,l} \sim \text{Bernoulli}(\sigma(\beta z_{i,l}^{(p)}))$$

  where $\sigma(\cdot)$ is the sigmoid function and β controls sampling temperature. Weights W_{ij} may also be sampled from a discrete codebook using softmax over latent logits.

In both cases, the adjacency matrix A acts as a structural mask over W, ensuring that only active edges contribute to system dynamics.

3.4 Genetic Operators

To explore and optimize the space of latent genotypes, we employ a set of genetic operators inspired by classical evolutionary algorithms. These operators act directly on the latent quantum-inspired genotype (QIG) $\mathbf{z} \in \mathbb{R}^{d\times 2}$, which encodes a superposition-like representation of a discrete graph dynamical system.

Mutation. Mutation introduces local variability into the genotype by perturbing individual components of $\mathbf{z}$. For a given QIG, each latent pair (α_i, β_i) is modified according to a noise distribution, such as:

$$\alpha_i \leftarrow \alpha_i + \epsilon_1, \quad \beta_i \leftarrow \beta_i + \epsilon_2, \quad \epsilon_1, \epsilon_2 \sim \mathcal{N}(0, \sigma^2)$$

This form of Gaussian mutation preserves the continuous nature of the latent space and enables fine-grained exploration. More structured alternatives (e.g., directionally biased noise or coordinate-wise dropout) can be applied depending on the desired variation profile.

Crossover. Crossover enables recombination of genetic material from two or more parent genotypes. Given two parent QIGs $\mathbf{z}^{(a)}$ and $\mathbf{z}^{(b)}$, we produce an offspring via either:

- *Uniform crossover*: randomly selecting each latent pair from one of the parents:

$$z_i^{(\text{child})} = \begin{cases} z_i^{(a)} & \text{with probability } 0.5 \\ z_i^{(b)} & \text{otherwise} \end{cases}$$

- *Interpolated crossover*: blending latent values using a convex combination:

$$z_i^{(\text{child})} = (1-\lambda) \cdot z_i^{(a)} + \lambda \cdot z_i^{(b)}, \quad \lambda \in [0, 1]$$

Uniform crossover promotes structural diversity by recombining discrete traits, while interpolated crossover enables smoother inheritance of shared traits and latent regularities.

Collapse-aware Variation. The structure of the collapse operator introduces a natural nonlinearity between the latent space and the resulting phenotype. In particular, mutations in the latent vector $\mathbf{z}$ may yield subtle or substantial changes in the binary system representation $\Gamma = (A, W, P)$, depending on proximity to collapse thresholds. This may allow for a range of variation scales — from local edge flips to larger topological shifts — although we leave a formal characterization of this behavior to experiments in future revisions.

Exploration via Stochastic Collapse. Stochastic variants of the collapse operator can be combined with genetic operators to encourage exploration in high-entropy regions of latent space. For instance, latent logits near 0 may lead to more frequent sampling variability, while increasing the temperature parameter β may bias sampling toward more deterministic outcomes. This introduces a tunable tradeoff between exploration and convergence, though we do not empirically evaluate this mechanism here.

Together, these genetic operations enable robust exploration of the QIG latent space while preserving the semantic integrity and executability of the resulting systems.

3.5 Evolutionary Algorithm - Classical Computing

To evolve quantum-inspired genotypes (QIGs) into high-performing graph dynamical systems, we implement a population-based evolutionary algorithm operating directly in the continuous latent space $\mathbf{z} \in \mathbb{R}^{d\times 2}$. The algorithm iteratively selects, perturbs, and evaluates QIGs based on the quality of the systems produced via the collapse operator.

Overview. Each individual in the population is a latent vector $\mathbf{z}_i$, which is collapsed into a phenotype $\Gamma_i = (A_i, W_i, P_i)$ using the deterministic collapse operator $\mathcal{C}$. This phenotype defines a specific graph dynamical system whose behavior is evaluated on a given task. Fitness is assigned based on the emergent dynamics or performance of the resulting system, depending on the evaluation context.

Algorithm. The evolutionary loop proceeds as follows:

1. **Initialization:** Sample an initial population $\{\mathbf{z}_i\}_{i=1}^N$, where each latent vector $\mathbf{z}_i \in \mathbb{R}^{d\times 2}$ is composed of d latent pairs $z_{i,j} = (\alpha_{i,j}, \beta_{i,j})$ sampled from the unit circle:
$$\theta_{i,j} \sim \text{Uniform}[0, 2\pi], \quad z_{i,j} = (\cos(\theta_{i,j}), \sin(\theta_{i,j}))$$
This ensures each latent component lies on the 2D unit circle, consistent with a superposition-style encoding.
2. **Collapse:** For each genotype $\mathbf{z}_i$, compute $\Gamma_i = \mathcal{C}(\mathbf{z}_i)$ and instantiate the corresponding GDS.
3. **Evaluation:** Simulate each system and compute fitness $f_i = \mathcal{R}(\Gamma_i)$ based on task-specific performance metrics.
4. **Selection:** Select a subset of genotypes using tournament selection or rank-based sampling.
5. **Variation:** Apply mutation and crossover operators in latent space:
   - **Mutation:** Add Gaussian noise: $\mathbf{z}' = \mathbf{z} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I)$
   - **Crossover:** Perform blend crossover: $\mathbf{z}' = \lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2$, with $\lambda \in [0, 1]$
6. **Replacement:** Update the population with the offspring and optionally retain elites.

Unlike classical encodings, QIGs enable smooth exploration via continuous latent vectors. This evolutionary process holds the potential to enable the emergence of structured, dynamical systems with meaningful topologies and behavior — all driven by optimization over a latent continuous genotype space.

3.6 Quadratic Unconstrained Binary Optimization — Quantum Computer

Quadratic Unconstrained Binary Optimization (QUBO) is a canonical formulation for combinatorial optimization problems, expressed as the minimization of a quadratic objective over binary variables:

$$\min_{x\in\{0,1\}^n} \quad x^\top Q x + c^\top x,$$

where $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix capturing pairwise interactions between variables, and $c \in \mathbb{R}^n$ encodes individual biases. QUBO is especially well-suited for encoding structural design problems, constraint satisfaction, and logic synthesis.

In our context, we propose formulating the direct optimization of the graph dynamical system (GDS) encoding $\Gamma = (A, W, P)$ as a QUBO problem. To enable this, we discretize the system representation into a binary genotype composed of:

- Flattened edge indicators $A_{ij} \in \{0, 1\}$,
- Quantized coupling weights, represented as binary encodings drawn from a predefined codebook,
- Binary node-level parameters $P_{i,l} \in \{0, 1\}$.

The full binary encoding can thus be represented as a vector $x \in \{0, 1\}^d$, where each bit corresponds to a design decision in the graph's topology, dynamics, or coupling structure. A task-specific QUBO matrix Q is then constructed to assign energy values to candidate solutions. The objective can incorporate terms that reward:

- Topologies that meet global or local constraints (e.g., connectivity, sparsity),
- Coupling patterns that promote desirable behaviors (e.g., synchrony, stability),

- Node configurations that lead to higher performance on a simulated or approximated task.

This formulation enables the use of quantum annealing hardware — such as D-Wave's quantum annealers — to explore the binary design space via physical quantum processes. These devices operate by initializing a system of qubits in a superposed low-energy state and gradually evolving it to minimize a QUBO energy landscape, ideally converging to the global optimum.

While the expressive mapping from high-level system behavior to the QUBO energy landscape remains a non-trivial modeling challenge, this approach opens the door to scalable hardware-accelerated search over complex, structured dynamical systems. In future work, we anticipate extending this formulation to support surrogate-based learning of QUBO coefficients from task-level rewards, and leveraging hybrid quantum-classical solvers to co-optimize latent QIGs and discrete system encodings.

4 Discussion

This work introduces a novel framework for the representation and evolution of graph dynamical systems, grounded in the concept of quantum-inspired genotypes (QIGs). By combining continuous latent representations with a structured collapse operator, we enable smooth optimization over discrete, executable systems that integrate topology, coupling, and local dynamics.

Our encoding is both expressive and compositional: the binary system representation $\Gamma = (A, W, P)$ captures the structural and functional properties of a broad class of discrete-time dynamical systems, while the latent genotype $\mathbf{z} \in \mathbb{R}^{d\times 2}$ allows for variation through well-defined genetic operators such as crossover and mutation. The collapse mechanism translates soft latent codes into interpretable and executable systems, supporting both deterministic and stochastic instantiations. We further outline how direct optimization of the binary encoding can be approached via QUBO, enabling potential deployment on quantum annealing hardware or classical solvers.

This work is presented as a conceptual and architectural contribution. We do not yet provide empirical evidence validating the evolutionary effectiveness of the framework or the behavioral properties of collapsed systems. In particular, the relationship between latent perturbations and phenotypic diversity, as well as the convergence characteristics of QIG-based optimization, remain to be demonstrated. Similarly, while QUBO offers a pathway to hardware-accelerated search, constructing Q matrices that faithfully encode task-level objectives is a non-trivial challenge.

In summary, this work proposes a new approach to encoding, evolving, and optimizing graph dynamical systems by uniting ideas from quantum-inspired computation, generative decoding, and combinatorial optimization. We hope that this foundation will stimulate further research into evolutionary design of structured systems, and provide a bridge between continuous search methods and discrete, interpretable computational architectures.

5 Conclusion

We have proposed a new framework for representing and evolving graph dynamical systems using quantum-inspired genotypes (QIGs). This approach combines continuous latent encodings with a structured collapse operator that maps into discrete system configurations defined by topology, coupling, and node-level dynamics. By enabling both genetic search over latent representations and direct QUBO-based optimization of binary system encodings, this framework offers a flexible foundation for exploring complex dynamical systems through evolutionary computation.

This work is presented as a conceptual contribution. No experimental results are provided, and the behavioral properties of the proposed encoding and variation mechanisms remain to be validated. Nonetheless, we believe the ideas introduced here — in particular, the structured collapse from continuous genotypes to discrete systems and the integration of QUBO as an optimization layer — offer a promising direction for future research. We encourage the community to explore, critique, and build upon this formulation.

6 Acknowledgements

The author gratefully acknowledges their UKRI PhD studentship at the University of Edinburgh, which supported prior research training and early exploration of related ideas. This work was completed independently during a period of academic leave. The author also acknowledges the use of large language models (LLMs) to support drafting, editing, and idea development during the research and writing process.

References

[1] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[2] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[3] Julian Francis Miller. *Cartesian Genetic Programming*. Springer, Berlin, Heidelberg, 2011.

[4] Conor Ryan, JJ Collins, and Michael O'Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Proceedings of the First European Workshop on Genetic Programming*, pages 83–96. Springer, 1998.

[5] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[6] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[7] Kwang Y Han and Jong H Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC)*, pages 3180–3185. IEEE, 2002.