Journal of Engineering Design and Analysis
Volume 8, Issue 1 - 2025, Pg. No. 1-8
Peer Reviewed Journal

Review Article

Evolutionary Algorithms for Engineering Problem-Solving: A Review

Priya Sharma

Ph D Scholar, Department of Civil Engineering, National Institute of Technology Kurukshetra (NIT Kurukshetra), Haryana, India

INFO

E-mail Id:
priyasharma12@gmail.ac.in
Orcid Id:
https://orcid.org/0009-0003-2703-9183

How to cite this article:
Sharma P. Evolutionary Algorithms for Engineering Problem-Solving: A Review. *J Engr Desg Anal* 2025*;* 8(1): 1-8.

Date of Submission: 2025-02-11
Date of Acceptance: 2025-03-15

ABSTRACT

Evolutionary algorithms (EAs) have emerged as powerful optimization techniques in engineering problem-solving, offering robust solutions for complex, nonlinear, and multi-objective problems. These nature-inspired algorithms, including Genetic Algorithms (GAs), Evolution Strategies (ES), Differential Evolution (DE), and Genetic Programming (GP), mimic biological evolution to iteratively improve candidate solutions through selection, crossover, and mutation operations. Due to their adaptability and global search capabilities, EAs have been extensively applied in various engineering domains, such as structural optimization, mechanical design, robotics, and industrial automation.

This review provides a comprehensive overview of the fundamental principles of EAs, their key variations, and their role in solving real-world engineering challenges. The study highlights the strengths and limitations of different evolutionary techniques and their performance in handling constrained, dynamic, and multi-objective optimization problems. Furthermore, the integration of evolutionary algorithms with machine learning, swarm intelligence, and metaheuristic hybridization is discussed, demonstrating their enhanced efficiency in tackling complex engineering tasks.

The article also explores recent advancements in evolutionary computation, including hybridization with artificial intelligence (AI), quantum-inspired evolutionary computing, and adaptive parameter control. These emerging trends aim to improve convergence speed, solution accuracy, and computational efficiency. Future research directions focus on developing more intelligent, scalable, and domain-specific evolutionary techniques that can address the increasing complexity of modern engineering problems.

By summarizing the evolution, applications, and future scope of EAs, this review provides valuable insights into their continued relevance and potential impact on engineering optimization and automation.

Keywords: Evolutionary Algorithms (EAs), Nature-Inspired Algorithms, Genetic Programming (GP)

Journal of Engineering Design and Analysis (ISSN: 2582-5607)

Introduction

Optimization is a fundamental challenge in engineering, where finding optimal solutions for design, scheduling, control, and decision-making processes is crucial. Traditional optimization techniques, such as gradient-based and mathematical programming methods, often struggle with highly complex, nonlinear, and multi-objective problems, especially when dealing with discontinuous search spaces, non-differentiable functions, and a large number of constraints. These limitations have led to the increasing popularity of Evolutionary Algorithms (EAs) as a robust alternative approach.

EAs are stochastic, population-based optimization techniques inspired by natural selection and evolution. By mimicking the principles of survival of the fittest, genetic inheritance, mutation, and recombination, EAs iteratively improve candidate solutions through evolutionary operations. Unlike traditional optimization methods, EAs do not require prior knowledge of the problem's gradient or mathematical structure, making them well-suited for solving highly complex and dynamic optimization problems. Their ability to explore large search spaces while avoiding local optima has made them highly effective in engineering applications.[1]

Over the past few decades, EAs have been widely applied in various engineering disciplines, including structural optimization, mechanical design, control system tuning, robotics, and artificial intelligence. Their ability to handle multi-objective and constrained optimization problems has further increased their relevance in fields such as aerospace engineering, industrial automation, and energy systems. Moreover, hybrid approaches that combine EAs with machine learning, swarm intelligence, and metaheuristic techniques have further enhanced their efficiency and effectiveness in solving real-world engineering problems.[2]

This article provides a comprehensive review of evolutionary algorithms, covering their fundamental principles, key variations, and notable applications in engineering. The review also explores the latest advancements in EA-based optimization, including hybridization with artificial intelligence, quantum-inspired evolutionary computing, and adaptive parameter control. By analyzing these emerging trends, this paper aims to highlight the potential future developments in evolutionary computation and its impact on engineering problem-solving.[3]

Evolutionary algorithms operate on a population of candidate solutions, iteratively evolving through selection, recombination, and mutation. These algorithms are particularly effective in exploring large and complex search spaces, allowing for robust optimization in engineering applications. The key components of EAs include:

- **Population Initialization:** The process begins with the generation of an initial population of candidate solutions. This can be done randomly or through heuristic-based methods to improve convergence speed. A well-diversified initial population helps prevent premature convergence and ensures a broader exploration of the solution space.[4]
- **Fitness Evaluation:** Each candidate solution is assigned a fitness score based on an objective function, which quantifies the quality of the solution. The fitness function may involve single or multiple objectives, such as minimizing cost, maximizing performance, or balancing trade-offs in engineering design.
- **Selection Mechanism:** The selection process determines which individuals from the population will contribute to the next generation. Common selection methods include:
- **Tournament Selection:** A subset of individuals is randomly chosen, and the best among them is selected.
- Roulette Wheel Selection: Individuals are chosen probabilistically based on their relative fitness values.[5] Rank-Based Selection: Individuals are ranked according to fitness, and selection probability is assigned accordingly.
- **Elitism:** A mechanism that ensures the best solutions are directly carried over to the next generation to maintain progress.
- **Genetic Operators:** These are used to generate new solutions by modifying existing ones.
- **Crossover (Recombination):** This operator combines genetic material from two parent solutions to produce offspring. Common crossover methods include one-point crossover, two-point crossover, and uniform crossover, each affecting the diversity and convergence properties of the algorithm.
- **Mutation:** Introduces small random variations in individuals to maintain genetic diversity and explore new areas of the search space. Mutation rates must be carefully controlled to balance exploration and exploitation.[6]
- **Survivor Selection:** After generating new offspring, a survivor selection mechanism determines which individuals will advance to the next generation. This can be based on elitism, steady-state selection (replacing only a portion of the population), or generational replacement (where the entire population is replaced).

Evolutionary algorithms come in different variations, each tailored to specific types of optimization problems. These algorithms differ in their representation of solutions, selection mechanisms, and genetic operations, making them suitable for a wide range of engineering applications.[7]

Genetic Algorithms (GA)

Genetic Algorithms (GAs) are the most widely used type of evolutionary algorithm, inspired by the principles of natural selection and genetics. Candidate solutions, known as chromosomes, are typically represented as binary strings, real-valued vectors, or permutation sequences. GAs operate through selection, crossover (recombination), and mutation to iteratively improve solutions over generations. They are particularly effective in solving combinatorial optimization problems, engineering design challenges, and complex scheduling tasks.[8]

Key Features:

- Chromosome-based encoding for solution representation
- Versatile crossover techniques, including single-point, two-point, and uniform crossover
- Adaptive mutation strategies to maintain diversity in the population
- Suitable for both constrained and unconstrained optimization problems

Applications:

- Structural design optimization in civil and mechanical engineering, improving material efficiency and structural integrity
- Feature selection in machine learning, reducing computational complexity while improving model accuracy
- Supply chain and logistics optimization, enhancing routing efficiency and resource allocation
- Design automation in automotive and aerospace industries for lightweight material structures

Evolution Strategies (ES)

Evolution Strategies (ES) differ from GAs in their focus on real-valued parameter optimization and adaptive mutation strategies. Unlike GAs, ES often employs self-adaptive control parameters, where mutation rates evolve over time to refine solutions dynamically. ES is particularly well-suited for high-dimensional and continuous optimization problems.

Key Features:

- Real-valued solution representation
- Self-adaptive mutation rates, improving convergence in complex landscapes
- Parent-offspring competition, often using ($\mu+\lambda$) or (μ,λ) selection schemes
- Robust in handling dynamic and noisy optimization environments

Applications:

- Aerodynamic shape optimization in aerospace engineering, refining airfoil designs for improved lift-to-drag ratios
- Control system tuning in robotics and automation, optimizing controller parameters for precise motion and stability
- Biomedical image analysis, improving segmentation accuracy and feature extraction in medical imaging
- Computational fluid dynamics (CFD) optimization for heat exchangers and turbine blade designs.[9]

Differential Evolution (DE)

Differential Evolution (DE) is a powerful optimization technique specifically designed for continuous-valued functions. Unlike traditional EAs, DE employs vector-based mutation strategies with differential perturbations to enhance the exploration and exploitation balance. DE is widely recognized for its efficiency in handling large-scale and complex optimization problems.

Key Features:

- Simple yet effective mutation strategies, such as DE/rand/1/bin and DE/best/2/bin
- Self-organizing population evolution, leading to faster convergence
- High efficiency in multimodal function optimization, avoiding premature convergence
- Robust parameter tuning capabilities for real-world engineering applications

Applications:

- Electrical power system optimization, including load flow analysis and voltage stability improvement
- Parameter estimation in mechanical system modeling, enhancing predictive accuracy in dynamic system behavior
- Image processing and pattern recognition, refining object detection and segmentation methods
- Chemical process optimization, improving reaction efficiency and minimizing waste production.[10,11]

Genetic Programming (GP)

Genetic Programming (GP) extends the principles of genetic algorithms to evolve executable programs, symbolic expressions, and mathematical models. Unlike fixed-length chromosome representations in GAs, GP employs tree-based structures that dynamically grow and evolve to represent increasingly complex solutions. GP has proven to be a valuable tool in areas requiring automated decision-making, data-driven modeling, and symbolic regression.

Key Features:

- Tree-based representation of solutions, enabling dynamic program evolution
- Automatically generated algorithms and predictive models without explicit programming
- Crossover and mutation applied to tree structures to refine solutions iteratively
- High flexibility in adapting to different problem domains

Applications:

- Automated design of control algorithms, generating adaptive controllers for robotics and industrial automation
- Symbolic regression for predictive modeling, discovering mathematical relationships in experimental data
- Financial market prediction, evolving trading strategies based on historical data patterns
- Fault diagnosis in mechanical and electrical systems, detecting anomalies through learned rule sets

These major evolutionary algorithms provide a diverse set of tools for solving engineering problems, each with distinct advantages depending on the nature of the optimization task. The continuous evolution of these algorithms, along with hybridization techniques, further enhances their applicability across various industries.

Evolutionary algorithms have demonstrated their effectiveness in various engineering fields by solving complex optimization problems that traditional methods struggle to address. Their ability to handle nonlinear, multi-objective, and large-scale problems makes them valuable tools for improving efficiency, reducing costs, and enhancing performance across multiple industries.[12]

Structural Engineering and Design Optimization

In structural engineering, evolutionary algorithms (EAs) are used to enhance the performance of buildings, bridges, and other infrastructure by optimizing designs for material efficiency, durability, and resilience. These algorithms help engineers explore numerous design possibilities while considering constraints such as load-bearing capacity, seismic resistance, and sustainability.[13]

Key Applications:

- **Topology optimization of load-bearing structures:** EAs assist in generating lightweight yet strong structures by distributing materials effectively.
- **Earthquake-resistant building design:** Optimizing damping and structural reinforcements to minimize damage during seismic activities.
- **Material selection for lightweight construction:** Identifying the best material compositions for durability, cost-effectiveness, and environmental impact.
- **Optimization of trusses and frame structures:** Improving stiffness and load distribution while minimizing material usage.
- **Bridge design improvements:** Enhancing structural stability and lifespan by optimizing cross-sectional designs and support placements.

Mechanical and Aerospace Engineering

Mechanical and aerospace engineers use EAs to improve design efficiency, aerodynamic performance, and mechanical stability. These algorithms help optimize aircraft and automotive structures, leading to reduced fuel consumption and enhanced durability.[14]

Key Applications:

- **Automated aerodynamic shape optimization for aircraft and automotive components:** Refining airfoil and vehicle body shapes to minimize drag and improve fuel efficiency.
- **Engine parameter tuning for fuel efficiency:** Optimizing combustion processes, injection timing, and air-fuel ratios for maximum energy output.
- **Vibration control and noise reduction in mechanical systems:** Adjusting damping mechanisms and material properties to minimize unwanted vibrations and acoustic emissions.
- **Fatigue and durability analysis:** Enhancing the longevity of mechanical components by optimizing stress distribution.
- **Propulsion system optimization:** Improving efficiency in jet engines, gas turbines, and hybrid propulsion systems.

Robotics and Automation

EAs play a vital role in robotics, enhancing adaptability, autonomy, and efficiency in motion planning, control system optimization, and intelligent decision-making. These algorithms allow robotic systems to learn, adapt, and perform complex tasks in dynamic environments.[15]

Key Applications:

- **Gait optimization in legged robots:** Refining walking patterns to achieve energy-efficient and stable movement.
- **Swarm robotics for autonomous search-and-rescue missions:** Coordinating multiple robots to navigate complex terrains in disaster response scenarios.
- **Gripper design and force control for industrial robots:** Enhancing robotic manipulators for precise handling of delicate and variable-sized objects.
- **Autonomous navigation in dynamic environments:** Optimizing path planning algorithms for drones, self-driving cars, and industrial automation systems.
- **Collaborative robotics (cobots) for smart factories:** Enhancing human-robot collaboration through adaptive motion planning.

Industrial Process and Manufacturing Optimization

EAs have revolutionized industrial processes by optimizing manufacturing strategies, improving product quality, and reducing operational costs. These algorithms enable real-time decision-making and enhance process efficiency.

Key Applications:

- **Assembly line balancing and job-shop scheduling:** Allocating resources efficiently to minimize bottlenecks and increase throughput.

- **Predictive maintenance and fault diagnosis:** Using evolutionary approaches to detect equipment failures before they occur, reducing downtime and maintenance costs.
- **Additive manufacturing process parameter optimization:** Enhancing 3D printing processes by optimizing material deposition rates, laser power, and cooling rates.
- **Welding process optimization:** Improving welding strength and reducing defects through fine-tuned parameter adjustments.
- **Supply chain logistics and inventory management:** Minimizing costs while ensuring timely delivery of raw materials and finished goods.

Electrical and Energy Systems

The energy sector benefits significantly from evolutionary algorithms, which aid in optimizing power generation, distribution, and renewable energy integration. EAs are increasingly applied to smart grid management and energy efficiency enhancement.

Key Applications:

- **Optimal placement of distributed energy resources:** Determining the best locations for renewable energy sources such as solar panels and wind turbines.
- **Load forecasting using hybrid evolutionary models:** Enhancing demand prediction accuracy to balance energy supply and consumption.
- **Fault detection in electrical grids:** Identifying system failures early to prevent large-scale outages and improve grid reliability.
- **Smart grid optimization:** Managing real-time power distribution to improve grid stability and efficiency.
- Electric vehicle (EV) charging station network design – Optimizing locations and charging infrastructure to support widespread EV adoption.

The field of evolutionary computation is rapidly evolving, with new trends enhancing the efficiency, accuracy, and applicability of Evolutionary Algorithms (EAs) across engineering disciplines. Recent advancements in artificial intelligence, quantum computing, and adaptive optimization techniques are shaping the future of EA-based problem-solving. These innovations enable more sophisticated and efficient solutions for complex engineering challenges[16].

Hybrid Evolutionary Algorithms

Modern engineering problems often involve highly complex, high-dimensional, and multi-modal optimization landscapes that traditional EAs struggle to handle efficiently. Hybrid optimization approaches, which combine EAs with artificial intelligence (AI), deep learning, fuzzy logic, or swarm intelligence, are increasingly being explored to enhance performance.

Key Advancements:

- **Faster convergence in large-scale optimization problems:** Hybrid models leverage machine learning to guide evolutionary search, reducing computational effort.
- **Improved generalization in data-driven design automation:** AI-driven hybrid approaches enhance predictive modeling and automated design generation.
- **Integration with reinforcement learning (RL):** Reinforcement learning techniques enhance evolutionary search strategies, improving decision-making in dynamic environments.
- **Neuroevolution techniques:** The application of EAs to optimize neural network architectures results in more efficient AI models for engineering applications.
- **Combination with swarm intelligence:** Hybridization with algorithms like Particle Swarm Optimization (PSO) or Ant Colony Optimization (ACO) enhances exploration and exploitation balance.

Quantum-Inspired Evolutionary Computation

With the rise of quantum computing, researchers are exploring quantum-inspired evolutionary computation to improve problem-solving capabilities. By incorporating quantum principles such as superposition and entanglement, these methods offer significant advantages over classical evolutionary algorithms.

Key Advancements:

- **Faster exploration of solution spaces:** Quantum-based operators allow more efficient parallel searches, accelerating convergence in high-dimensional problems.
- **Enhanced performance in multi-objective optimization:** Quantum algorithms improve the handling of conflicting objectives in engineering design.
- **Quantum-enhanced mutation and crossover mechanisms:** These mechanisms help escape local optima more effectively, improving global search efficiency.
- **Application in cryptographic engineering and cybersecurity:** Quantum-inspired optimization is being used to enhance data security and encryption algorithms.
- **Energy-efficient computation:** Quantum computing has the potential to significantly reduce computational energy costs for complex simulations.

Adaptive and Self-Learning Evolutionary Algorithms

Traditional EAs operate with fixed parameters, which can limit their effectiveness in dynamic environments. Adaptive and self-learning evolutionary algorithms modify key parameters in real time, improving optimization efficiency and robustness.

Key Advancements:

- **Greater robustness in dynamic environments:** Self-adjusting algorithms can respond to changes in constraints, objectives, and system conditions.
- **Improved search efficiency in complex engineering problems:** Dynamic adaptation of mutation rates and crossover probabilities enhances solution diversity.
- **Automated parameter tuning:** Evolutionary strategies can self-optimize parameters based on real-time feedback from the optimization process.
- **Self-adaptive multi-population strategies:** These strategies enable simultaneous optimization of multiple conflicting objectives with improved convergence rates.
- **Applications in autonomous systems:** Adaptive EAs are used in self-learning robotic systems, smart grids, and real-time industrial control processes.

Evolutionary Multi-Objective Optimization (EMO)

Engineering optimization problems often involve multiple conflicting objectives, such as minimizing cost while maximizing performance. Evolutionary Multi-Objective Optimization (EMO) techniques, particularly those based on Pareto dominance, enable engineers to explore trade-offs efficiently.

Key Advancements:

- **Better trade-off solutions for engineering design:** Advanced Pareto-based algorithms generate a diverse set of optimal solutions, allowing decision-makers to select the most suitable trade-offs.
- **Higher efficiency in resource allocation problems:** EMO enhances decision-making in energy management, supply chain optimization, and infrastructure planning.
- **Many-objective optimization (MaOEA) methods:** These methods extend traditional multi-objective EAs to handle optimization problems with more than three conflicting objectives.
- **Machine learning-assisted EMO:** AI-driven techniques improve the scalability and decision-making capabilities of multi-objective optimization frameworks.
- **Real-time multi-objective decision support systems:** EMO is increasingly used in adaptive engineering environments, where real-time decisions must be made under uncertainty.

Evolutionary Computation in Digital Twin Technology

Digital twins—virtual representations of physical systems—are becoming a cornerstone of engineering innovation. The integration of evolutionary computation with digital twins enables real-time optimization and predictive analytics.

Key Advancements:

- **Real-time synchronization with physical systems:** EAs continuously update digital twin models based on sensor data and real-world observations.
- **Predictive maintenance and failure prevention:** Evolutionary algorithms optimize maintenance schedules and detect anomalies before system failures occur.
- **Optimization of cyber-physical systems:** EA-driven simulations enhance the efficiency of interconnected smart systems, such as autonomous vehicles and smart manufacturing.
- **Application in healthcare engineering:** Digital twins of biological systems are being optimized using EAs for personalized medicine and medical device design.
- **Integration with IoT and Industry 4.0:** Evolutionary computing is playing a crucial role in optimizing industrial automation and intelligent monitoring systems.

Evolutionary Deep Learning and AutoML

Automated Machine Learning (AutoML) and evolutionary deep learning are emerging as powerful approaches for engineering applications. By leveraging evolutionary search strategies, these methods optimize neural network architectures and hyperparameters.

Key Advancements:

- **Automated neural architecture search (NAS):** EAs optimize deep learning models for better accuracy, efficiency, and interpretability.
- **Hyperparameter tuning for AI models:** Evolutionary strategies enhance training efficiency and generalization capabilities.
- **Applications in image processing and pattern recognition:** Evolutionary deep learning is being used for defect detection in manufacturing and predictive maintenance.
- **Optimization of reinforcement learning policies:** Evolutionary algorithms help train AI agents for robotics, autonomous systems, and smart decision-making.
- **Scalable AI model development:** EA-driven AutoML accelerates the deployment of AI models in engineering applications.

The field of evolutionary computation is rapidly evolving, with new trends enhancing the efficiency, accuracy, and applicability of Evolutionary Algorithms (EAs) across engineering disciplines. Recent advancements in artificial intelligence, quantum computing, and adaptive optimization techniques are shaping the future of EA-based problem-solving. These innovations enable more sophisticated and efficient solutions for complex engineering challenges.

Hybrid Evolutionary Algorithms

Modern engineering problems often involve highly complex, high-dimensional, and multi-modal optimization landscapes that traditional EAs struggle to handle efficiently. Hybrid optimization approaches, which combine EAs with artificial intelligence (AI), deep learning, fuzzy logic, or swarm intelligence, are increasingly being explored to enhance performance.

Key Advancements:

- **Faster convergence in large-scale optimization problems:** Hybrid models leverage machine learning to guide evolutionary search, reducing computational effort.
- **Improved generalization in data-driven design automation:** AI-driven hybrid approaches enhance predictive modeling and automated design generation.
- **Integration with reinforcement learning (RL):** Reinforcement learning techniques enhance evolutionary search strategies, improving decision-making in dynamic environments.
- **Neuroevolution techniques:** The application of EAs to optimize neural network architectures results in more efficient AI models for engineering applications.
- **Combination with swarm intelligence:** Hybridization with algorithms like Particle Swarm Optimization (PSO) or Ant Colony Optimization (ACO) enhances exploration and exploitation balance.

Quantum-Inspired Evolutionary Computation

With the rise of quantum computing, researchers are exploring quantum-inspired evolutionary computation to improve problem-solving capabilities. By incorporating quantum principles such as superposition and entanglement, these methods offer significant advantages over classical evolutionary algorithms.

Key Advancements:

- **Faster exploration of solution spaces:** Quantum-based operators allow more efficient parallel searches, accelerating convergence in high-dimensional problems.
- **Enhanced performance in multi-objective optimization:** Quantum algorithms improve the handling of conflicting objectives in engineering design.
- **Quantum-enhanced mutation and crossover mechanisms:** These mechanisms help escape local optima more effectively, improving global search efficiency.
- **Application in cryptographic engineering and cybersecurity:** Quantum-inspired optimization is being used to enhance data security and encryption algorithms.
- **Energy-efficient computation:** Quantum computing has the potential to significantly reduce computational energy costs for complex simulations.

Adaptive and Self-Learning Evolutionary Algorithms

Traditional EAs operate with fixed parameters, which can limit their effectiveness in dynamic environments. Adaptive and self-learning evolutionary algorithms modify key parameters in real time, improving optimization efficiency and robustness.

Key Advancements:

- **Greater robustness in dynamic environments:** Self-adjusting algorithms can respond to changes in constraints, objectives, and system conditions.
- **Improved search efficiency in complex engineering problem:** Dynamic adaptation of mutation rates and crossover probabilities enhances solution diversity.
- **Automated parameter tuning:** Evolutionary strategies can self-optimize parameters based on real-time feedback from the optimization process.
- **Self-adaptive multi-population strategies:** These strategies enable simultaneous optimization of multiple conflicting objectives with improved convergence rates.
- **Applications in autonomous systems:** Adaptive EAs are used in self-learning robotic systems, smart grids, and real-time industrial control processes.

Evolutionary Multi-Objective Optimization (EMO)

Engineering optimization problems often involve multiple conflicting objectives, such as minimizing cost while maximizing performance. Evolutionary Multi-Objective Optimization (EMO) techniques, particularly those based on Pareto dominance, enable engineers to explore trade-offs efficiently.

Key Advancements:

- **Better trade-off solutions for engineering design:** Advanced Pareto-based algorithms generate a diverse set of optimal solutions, allowing decision-makers to select the most suitable trade-offs.
- **Higher efficiency in resource allocation problems:** EMO enhances decision-making in energy management, supply chain optimization, and infrastructure planning.
- **Many-objective optimization (MaOEA) methods:** These methods extend traditional multi-objective EAs to handle optimization problems with more than three conflicting objectives.
- **Machine learning-assisted EMO:** AI-driven techniques improve the scalability and decision-making capabilities of multi-objective optimization frameworks.
- **Real-time multi-objective decision support systems:** EMO is increasingly used in adaptive engineering environments, where real-time decisions must be made under uncertainty.

Evolutionary Computation in Digital Twin Technology

Digital twins—virtual representations of physical systems—are becoming a cornerstone of engineering innovation. The integration of evolutionary computation with digital twins enables real-time optimization and predictive analytics.

Key Advancements:

- **Real-time synchronization with physical systems:** EAs continuously update digital twin models based on sensor data and real-world observations.
- **Predictive maintenance and failure prevention:** Evolutionary algorithms optimize maintenance schedules and detect anomalies before system failures occur.
- **Optimization of cyber-physical systems:** EA-driven simulations enhance the efficiency of interconnected smart systems, such as autonomous vehicles and smart manufacturing.
- **Application in healthcare engineering:** Digital twins of biological systems are being optimized using EAs for personalized medicine and medical device design.
- **Integration with IoT and Industry 4.0:** Evolutionary computing is playing a crucial role in optimizing industrial automation and intelligent monitoring systems.

Evolutionary Deep Learning and AutoML

Automated Machine Learning (AutoML) and evolutionary deep learning are emerging as powerful approaches for engineering applications. By leveraging evolutionary search strategies, these methods optimize neural network architectures and hyperparameters.

Key Advancements:

- **Automated neural architecture search (NAS):** EAs optimize deep learning models for better accuracy, efficiency, and interpretability.
- **Hyperparameter tuning for AI models:** Evolutionary strategies enhance training efficiency and generalization capabilities.
- **Applications in image processing and pattern recognition:** Evolutionary deep learning is being used for defect detection in manufacturing and predictive maintenance.
- **Optimization of reinforcement learning policies:** Evolutionary algorithms help train AI agents for robotics, autonomous systems, and smart decision-making.
- **Scalable AI model development:** EA-driven AutoML accelerates the deployment of AI models in engineering applications.

References

1. Holland J. Adaptation in artificial and natural systems. Ann Arbor: The University of Michigan Press. 1975;232.
2. Goldberg D. Genetic algorithms in search, optimization and machine learning. boston. usa.
3. Yao X, Liu Y. Fast Evolutionary Programming. Evolutionary programming. 1996 Feb 29;3:451-60.
4. Storn R. Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute. 1995;11.
5. Koza JR. Genetic programming: on the programming of computers by means of natural selection Cambridge. MA: MIT Press.[Google Scholar]. 1992.
6. Deb K. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons Ltd. Baffins Lane, Chichester, West Sussex, PO19 1UD. 2001.
7. Coello CA. Evolutionary algorithms for solving multi-objective problems. springer. com; 2007.
8. Eiben AE, Smith JE. Introduction to evolutionary computing. springer; 2015.
9. Haupt RL, Haupt SE. Practical genetic algorithms. John Wiley & Sons; 2004 Jul 16.
10. Gandomi AH, Yang XS, Alavi AH. Mixed variable structural optimization using firefly algorithm. Computers & Structures. 2011 Dec 1;89(23-24):2325-36.
11. Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Transactions on Evolutionary computation. 1999 Jul;3(2):82-102.
12. Bäck T, Fogel DB, Michalewicz Z. Handbook of evolutionary computation. Release. 1997;97(1):B1.
13. Beyer HG, Schwefel HP. Evolution strategies–a comprehensive introduction. Natural computing. 2002 Mar;1:3-52.
14. Fogel LJ, Owens AJ, Walsh MJ. Artificial Intelligence Through.
15. Deb K, Pratap A, Agarwal S, Meyarivan TA. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation. 2002 Apr;6(2):182-97.
16. EX L, INNO N. International Journal of Engineering and Advanced Technology... International Journal of Engineering and Advanced Technology.