Swarm and Evolutionary Computation 100 (2026) 102240

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Survey Paper

A survey on Quality-Diversity optimization: Approaches, applications, and challenges

Haoxiang Qin [a], Yi Xiang [a,*], Hainan Zhang [a], Yuyan Han [b], Yuting Wang [b], Xinrui Tao [c,d], Yiping Liu [e]

[a] School of Software Engineering, South China University of Technology, Guangzhou 510006, China
[b] School of Computer Science, Liaocheng University, Liaocheng 252059, China
[c] School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, 250101, China
[d] Shandong Key Laboratory of Smart Buildings and Energy Efficiency, Shandong Jianzhu University, Jinan, 250101, China
[e] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Keywords:
Quality-Diversity optimization
Evolutionary computation
Feature space
MAP-Elites

ABSTRACT

Quality-Diversity (QD) optimization is a paradigm of evolutionary computation (EC) that extends the classic approaches, aiming to generate a collection of solutions that are both diverse and high-performing. Unlike traditional evolutionary algorithms (EAs), QD methods emphasize the illumination (or coverage) of a user-defined feature space, while simultaneously aiming for local optimization within each discovered region of the feature space. Over the past decade, QD has rapidly developed and proven effective in areas such as evolutionary robotics and video games. However, a systematic review of this growing field remains lacking. To date, the most recent review article on QD was published in 2021. Therefore, to offer a more comprehensive overview of the latest QD research, this paper provides a thorough survey of QD optimization, covering its foundational principles and representative algorithmic frameworks such as Novelty Search with Local Competition (NSLC), MAP-Elites, the unified modular QD framework, and RIBS. In addition, we divide the algorithm improvement part into three modules for discussion: containers, selection, and mutation. Then, the evaluation metrics widely used in QD optimization are listed for researchers. We further explore its diverse applications across domains such as evolutionary robotics, video games, scheduling, software testing, and engineering design. Finally, we discuss the current challenges in the field and outline promising directions for future research.

1. Introduction

Optimization plays a fundamental role in various real-world applications, including electricity [1], part manufacturing [2], transportation [3], and scheduling [4–6]. It typically involves multiple interrelated tasks, and the challenge lies in finding the most appropriate solution without violating the problem constraints [7]. In recent decades, researchers have designed numerous automated programs to generate high-quality solutions, commonly referred to as optimization algorithms [8] which are widely used in a variety of tasks [9–14].

Evolutionary Computation (EC) is a widely used paradigm in optimization. Inspired by the principles of natural evolution, evolutionary algorithms (EAs) have been extensively employed as stochastic optimization techniques over the past decades [15]. The core idea behind EAs is to simulate evolutionary processes — selection, mutation, recombination — within a population of candidate solutions. In nature, evolution promotes not only the survival of the fittest but also the preservation of diversity, which enables populations to adapt to changing environments and avoid premature convergence. However, despite this biological inspiration, most EA methods in practice focus primarily on optimizing a single objective or converging to a global optimum, often neglecting the value of maintaining feature diversity within the population [8]. In particular, diversity in EAs is typically treated as a means to avoid local optima or to maintain search efficiency, rather than as an objective in its own right. As a result, limited attention has been paid to exploring diversity in terms of meaningful structural features of the solutions.

Quality-Diversity (QD) optimization [16], as a paradigm of EA, addresses the aforementioned limitations by leveraging the diversity of

* Corresponding author.
E-mail addresses: 987352978@qq.com (H. Qin), xiangyi@scut.edu.cn (Y. Xiang), zhnzhanghainan@126.com (H. Zhang), hanyuyan@lcu-cs.com (Y. Han), wangyuting@lcu-cs.com (Y. Wang), taoxinrui@shu.edu.cn (X. Tao), yiping0liu@gmail.com (Y. Liu).

https://doi.org/10.1016/j.swevo.2025.102240
Received 17 March 2025; Received in revised form 29 October 2025; Accepted 26 November 2025

problem features (Prior works have used the terms "features", "descriptors", "behavior descriptors", "behavior characteristics", "outcomes", and "measures". For clarity, this paper consistently uses the term "feature" throughout.) to solve the problem in complex scenarios. Unlike global optimization, which seeks only the globally optimal solution, the basic principle of QD is to search within a feature space to identify and maintain a diverse set of high-quality solutions [17]. In QD optimization, each solution is evaluated not only for its quality but also for its contribution to population diversity, with the collective goal of covering diverse regions of the feature space. This differs fundamentally from Multi-Modal Optimization (MMO) [18], which focuses on locating multiple optima in the genotype or decision space, and from Evolutionary Diversity Optimization (EDO) [19], which promotes diversity among feasible solutions but does not operate over an explicit feature space. In contrast, QD simultaneously emphasizes both solution quality and feature diversity in a structured feature space, enabling more interpretable and controllable exploration. Thanks to these advantages, this paradigm has led to significant breakthroughs in solving complex practical problems and has been featured twice in *Nature* [13,20]. To date, QD has been successfully applied across multiple domains, including evolutionary robotics [21–38], video games [39–44], software testing [11], engineering design [45–49], and scheduling and route planning [12,50–53].

The development related to QD can be traced back to the emergence of MMO. As illustrated in Fig. 1 (this figure lists QD-related works with relatively high citation counts in chronological order.), since Goldberg et al. [54] introduced MMO in 1987, the idea of maintaining solution diversity has significantly influenced the evolution of EAs, paving the way for the emergence of QD. Later, a major conceptual shift occurred in 2011 when Lehman and Stanley introduced Novelty Search (NS) [55] and Novelty Search with Local Competition (NSLC) [56], emphasizing feature diversity as a central optimization goal. In 2012, Ulrich et al. [19] proposed EDO, which focused on generating solution sets that are diverse. In 2013, Cully et al. [57] introduced Behavioral Repertoire (BR) Evolution, demonstrating the practical application of diversity-driven methods in robotics. A landmark contribution came in 2015 when Mouret et al. [58] proposed the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm, which discretizes the feature space into grid to jointly promote quality and diversity. That year also saw the formal introduction of the term "Quality-Diversity" by Pugh et al. [59], marking QD as a new paradigm.

Subsequent advances focused on improving efficiency and scalability. In 2017, Gaier et al. [60] introduced Surrogate-Assisted Illumination (SAIL), which employed surrogate models to reduce evaluation costs. In 2018, Cully et al. [8] proposed a modular QD framework, while Vassiliades et al. [61] developed Centroidal Voronoi Tessellation MAP-Elites (CVT-ME) to improve feature space coverage. Cazenille et al. [62] also released QDpy, which is a reusable QD framework. In 2020, Fontaine et al. [63] integrated Covariance Matrix Adaptation Evolution Strategy into MAP-Elites, forming CMA-ME, which enhanced local search abilities, while Mouret et al. [64] extended QD to multi-task settings. The year 2021 brought innovations such as Policy Gradient Assisted MAP-Elites (PGA-ME) [65], Differentiable QD [24], and a comprehensive QD survey [17], which outlined the methods and challenges of this field. Pyribs, one of the widely adopted libraries dedicated to QD algorithms, was initially released in February 2021, and the paper about it was published later at GECCO 2023 [66]. In 2022, QD was extended to multi-objective optimization [67]. QDax was released in 2022 and initially published as a poster paper at GECCO 2022 [68]. In 2023, a longer version of the paper was published in TMLR [37]. Yet another followup was also published in JMLR in 2024 (https://jmlr.org/papers/v25/23-1027.html). Research in 2023 addressed runtime behavior and transferability, including runtime analysis [69], the modular RIBS library [66], accelerated QD via parallelism [70], and QD transfer learning (QDTL) [71]. In 2024, Qian et al. [72] provided theoretical guarantees for QD optimization, while Grillotti et al. [73] proposed the QD Actor–Critic framework, blending reinforcement learning (RL) with QD.

As illustrated in Fig. 2, research in QD has grown rapidly over the past decade, leading to an increasing number of publications in the field. Since the publication of the QD survey by Chatzilygeroudis et al. [17] in 2021, research in this field has grown explosively. The number of papers published from 2022 to 2024 has already surpassed the total output from the preceding 7 years. This rapid development highlights the need for a comprehensive review and further discussion of QD optimization research.

Since the introduction of QD optimization, the field has experienced a rapid surge in research activity, with numerous influential works emerging in both theoretical and applied domains. Yet, to date, no survey has systematically integrated and synthesized these recent developments. Such an effort is both timely and essential for capturing the evolving landscape of QD. Furthermore, prior surveys primarily concentrated on traditional application areas such as evolutionary robotics and video games, but their broader adoption has been hindered by several key challenges. For example, in high-dimensional and constrained environments — such as those involving precedence relations, resource limits, or combinatorial structures — constructing and preserving meaningful feature diversity becomes significantly more difficult, especially when well-defined features are unavailable [74–79]. Scalability is another concern, as maintaining large archives in expansive search and feature spaces imposes substantial computational costs [17,80,81]. Furthermore, most QD methods assume deterministic settings, limiting their robustness under real-world uncertainty [32,82,83]. Maintaining diversity and solution quality also remains difficult in practice due to imperfect feature definitions, ineffective variation operators, and poorly aligned search guidance mechanisms [55,72,84]. Finally, integrating QD with paradigms such as deep reinforcement learning (DRL) or multi-objective optimization introduces additional challenges, including handling dynamic objectives, computational overhead, and maintaining consistent exploration–exploitation trade-offs [67,73,85–87]. In addition, it is worth noting that in recent years, the QD method has been extended to a wider range of fields, including software testing, scheduling, molecular/drug design, and urban planning. This diversification further highlights the pressing need for an up-to-date and comprehensive review to consolidate recent progress and inform future research directions.

The paper is organized as follows. Section 2 provides an overview of the background knowledge related to QD optimization. Section 3 explores the framework, theoretical studies, algorithmic improvements, evaluation metrics, and applications of QD. In Section 4, we discuss the main challenges and outline promising research directions. Finally, Section 5 provides concluding remarks.

2. Preliminaries

Before introducing the formal definition of QD optimization, it is important to acknowledge a closely related line of research—EDO. EDO focuses on generating sets of solutions that satisfy minimum quality requirements while maximizing diversity, particularly in the decision space. Ulrich et al. [88] first emphasized the role of structural diversity in multi-objective optimization, and later proposed NOAH [89], an evolutionary algorithm that alternates between maximizing diversity and adjusting the quality thresholds. Doerr et al. [90,91] demonstrated that incorporating structural diversity accelerates re-optimization and improves search efficiency. In combinatorial optimization, Gao et al. [92] and Bossek et al. [93] evolved TSP instances with varying structural difficulty using creative variation strategies. Some approaches integrated quality indicators such as hypervolume and entropy [94–96] to guide the generation of diverse and high-performing solutions. While both EDO and QD aim to promote diversity, they differ in quality handling. EDO usually works with a quality threshold (constraint) while QD stores the best solution into each cell that has been found without any

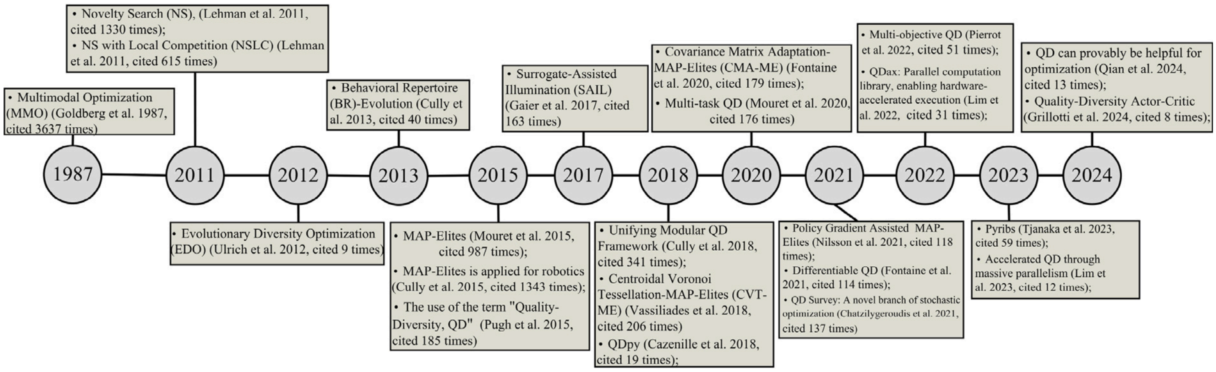


Fig. 1. Evolution of QD optimization: a timeline of developments with relatively higher citation counts (recorded from Google Scholar, August 31, 2025).

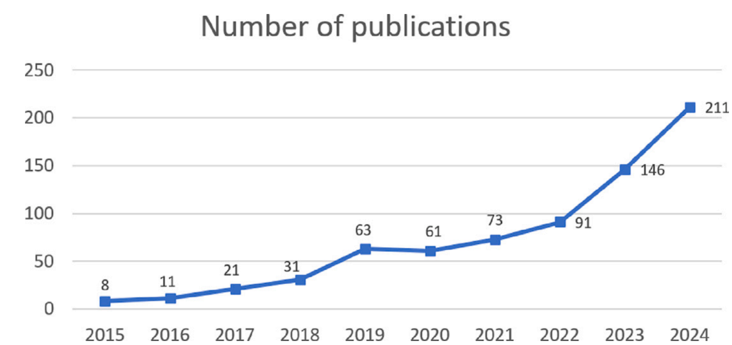


Fig. 2. Trends in the number of publications in the last decade. The data was collected from the Web of Science and the webpage https://quality-diversity.github.io/papers on May 14, 2025, using "Quality-Diversity" as the topic. A total of 716 publications were recorded from 2015 to 2024.

quality constraint (note: this statement of QD is true for MAP-Elites variants, not for NSLC variants). QD also leverages features to maintain an archive that reflects both performance and feature diversity.

We next provide the mathematical definitions of global optimization, MMO, and QD to further distinguish these concepts. Following [17], in traditional global optimization (Fig. 3.A), the objective is to find a single global optimum of a given function. However, due to the nonlinear and nonconvex nature of many problems, methods like EAs often converge to local optima [97]. In contrast, MMO algorithms [18,98–100] aim to discover multiple distinct optima across the search space (Fig. 3.B), which is particularly valuable in applications such as design, scheduling, and planning [12,50,101,102]. QD optimization builds on this concept by identifying multiple high-quality solutions and structuring them according to features. As illustrated in Fig. 3.C, this approach enables QD to maintain structured archives of diverse and high-quality solutions, offering both insight and utility across the feature space. Compared to global optimization and MMO, the key advantage of QD lies in its explicit focus on maintaining *feature diversity* alongside solution quality. Rather than simply identifying multiple optima, QD algorithms seek to illuminate the feature space by filling a structured archive with high-performing solutions that differ meaningfully in their features. This diversity is crucial in complex real-world problems where multiple valid solutions may exist, each with unique features.

We denote the decision space as $\mathcal{D}$, with a mapping $\mathcal{D} \to \mathbb{R}$. Let x_G^*, x_M^*, and x_{QD}^* represent the solutions obtained by global optimization, MMO, and QD optimization algorithms, respectively. The objective function is denoted by $f(x)$. Assuming a maximum problem, we follow the formal definitions outlined in [17], which clearly distinguish between global, multimodal, and QD optimization in terms of their search objectives and coverage features. These definitions are detailed in the following contents.

Global optimization is formally defined as follows:

$$\{x_G^* \in \mathcal{D} : f(x_G^*) > f(x), \forall x \in \mathcal{D}\} \tag{1}$$

where the goal of global optimization is to find the set containing a single globally optimal solution in decision space.

The definition of MMO is given as follows:

$$\{x_M^* \in \mathcal{D} : f(x_M^*) > f(x), \forall x \in \mathcal{D}, d(x, x_M^*) < \epsilon, \epsilon > 0\} \tag{2}$$

where d is a distance function. $\epsilon > 0$ is a predefined radius that determines the neighborhood within which a solution must be better than or equal to all others. The goal of MMO is to find the set containing all local optimal solutions in decision space.

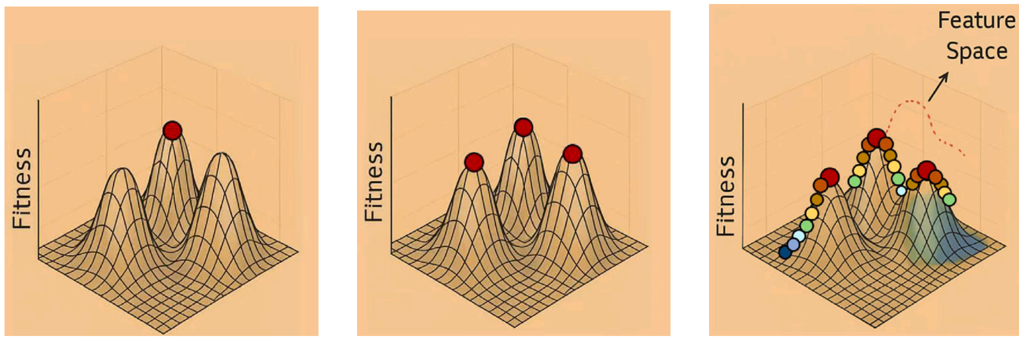
QD optimization aims to discover a diverse set of high-quality solutions within the feature space. See [103,104], the QD optimization model can be defined as follows:

Let $\Theta \subseteq \mathbb{R}^{n_\theta}$ denote the decision space, and let $\mathcal{B} \subseteq \mathbb{R}^{n_b}$ represent the feature space. Each candidate solution $\theta \in \Theta$ is evaluated through a feature function $\phi : \Theta \to \mathcal{B}$, which maps solutions to their corresponding feature representation $b_\theta = \phi_{\mathcal{B}}(\theta)$. The quality or performance of a solution is measured by an objective function $f : \Theta \to \mathbb{R}$.

The goal of QD optimization is to construct an archive $\mathcal{A} \subseteq \Theta$ that simultaneously maximizes performance and maintains diversity across the feature space. Formally, QD aims to identify a set $\mathcal{A}$ satisfying:

$$\begin{cases} \forall b \in \mathcal{B}_{\text{reach}},\ \exists \theta \in A,\ d_{\mathcal{B}}(\phi_{\mathcal{B}}(\theta), b) < \varepsilon, \\ \forall \theta' \in A,\ \theta' = \arg\max_{\theta \in N(b_{\theta'})} f(\theta) \end{cases} \tag{3}$$

where $\mathcal{B}_{\text{reach}} \subseteq \mathcal{B}$ denotes the space of reachable features, and $\varepsilon \in \mathbb{R}^{+*}$ is a small tolerance parameter defining the resolution of the feature space partition. The function $\phi_{\mathcal{B}} : \Theta \to \mathcal{B}$ maps each vector $\theta \in \Theta$ to its feature $b_\theta = \phi_{\mathcal{B}}(\theta)$, while $f : \Theta \to \mathbb{R}$ evaluates the performance of θ. The distance function $d_{\mathcal{B}} : \mathcal{B}^2 \to \mathbb{R}$ measures dissimilarity between features, and $N(b_{\theta'}) = \{\theta \mid \text{neighbor}_{d_{\mathcal{B}}}(b_\theta, b_{\theta'})\}$ defines the neighborhood of $b_{\theta'}$ in $\mathcal{B}$, typically determined using a k-nearest neighbors approach. Accordingly, the first condition guarantees that the archive A densely covers the reachable feature space, while the second ensures that each stored solution is locally optimal within its corresponding feature neighborhood.



(a) Global optimization. (b) Multimodal optimization. (c) QD optimization.

Fig. 3. Comparison among global, multimodal, and QD optimization. Figure adapted from [17]. (A) Global optimization algorithms systematically explore a single best solution that meets the objective function. (B) MMO algorithms focus on identifying multiple solutions that correspond to different local optima, ensuring solution diversity while optimizing the objective.(C) QD optimization algorithms aim to explore diverse features in feature space, generating a broad set of high-performing solutions.

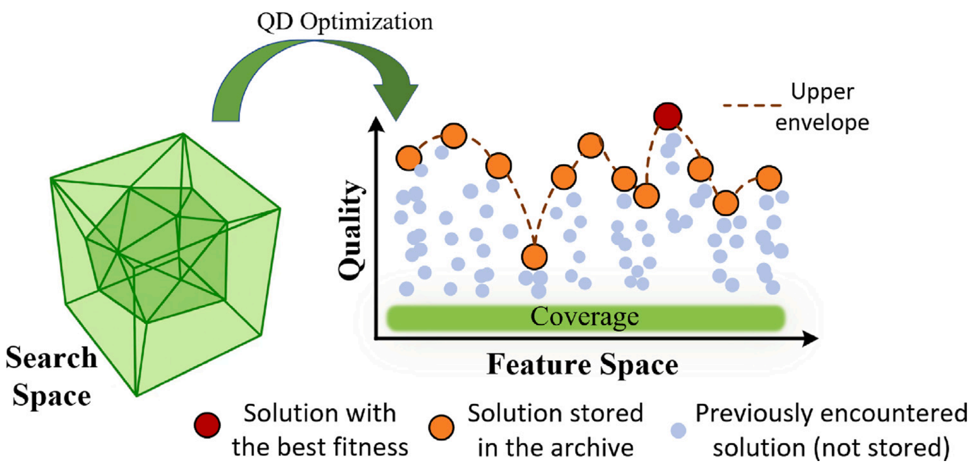


Fig. 4. The search process of QD explores the high-dimensional search space to discover a diverse set of solutions that map into the feature space. Each point in this space represents a solution characterized by its behavior feature (horizontal axis) and quality (vertical axis). Through iterative illumination, QD algorithms aim to uniformly cover the feature space (coverage) while simultaneously identifying high-performing solutions along the upper envelope of achievable performance.

To provide a more intuitive understanding of this formulation, see [8], Fig. 4 illustrates how QD optimization operates.

3. Quality-diversity optimization

In this section, we discuss the overall framework of QD optimization alongside its theories and core components. Regarding algorithmic improvements, we categorize them according to the container structure, selection mechanism, and variation strategy of QD. After that, we discuss the evaluation metrics and applications of QD. A taxonomy of QD is provided in Fig. 5.

3.1. Algorithm frameworks

We introduce representative frameworks that have shaped the development of QD methods, including NSLC and MAP-Elites. While both aim to discover a diverse set of high-performing solutions by leveraging features, they differ fundamentally in their storage and archiving strategies. NSLC uses an unstructured container, which maintains solutions using a distance threshold in feature space, allowing for flexible storage. In contrast, MAP-Elites employs a structured grid-based container that discretizes the feature space into predefined cells, with each cell typically storing only the best-performing solution within its region. We then present a unified QD framework that abstracts these methods into a modular structure and we highlight recent enhancements. Finally, another QD framework, RIBS, is introduced. Unlike the above frameworks, RIBS integrates various advanced optimization strategies.

3.1.1. Novelty search with local competition

Before introducing NSLC [56], it is important to briefly explain NS [55], as it forms the basis for NSLC. NS is an algorithm that ignores objective performance. Instead, it encourages exploration by rewarding feature diversity. Each solution is evaluated using a novelty score. This score is calculated as the average distance from its k nearest neighbors in a novelty archive. The distance is measured based on user-defined features. If the novelty score exceeds a certain threshold, the solution is added to the archive. This helps promote further exploration. NS is especially useful in deceptive tasks like the maze navigation problem. In such cases, following the objective can lead to local optima. In contrast, focusing on diversity helps uncover better solutions. NSLC builds upon NS by introducing local objective-based selection to complement diverse exploration. While NS focuses solely on novelty and entirely ignores the objective function, NSLC preserves this emphasis on feature diversity while also incorporating a performance criterion to guide the search toward high-quality solutions within similar feature regions. In NSLC, the distance measure captures how novel a solution is in the feature space by comparing it to existing solutions, using user-defined features [105]. By combining novelty and local performance, NSLC encourages the evolution of solutions that are not only diverse but also locally optimal with respect to others in their neighborhood. This

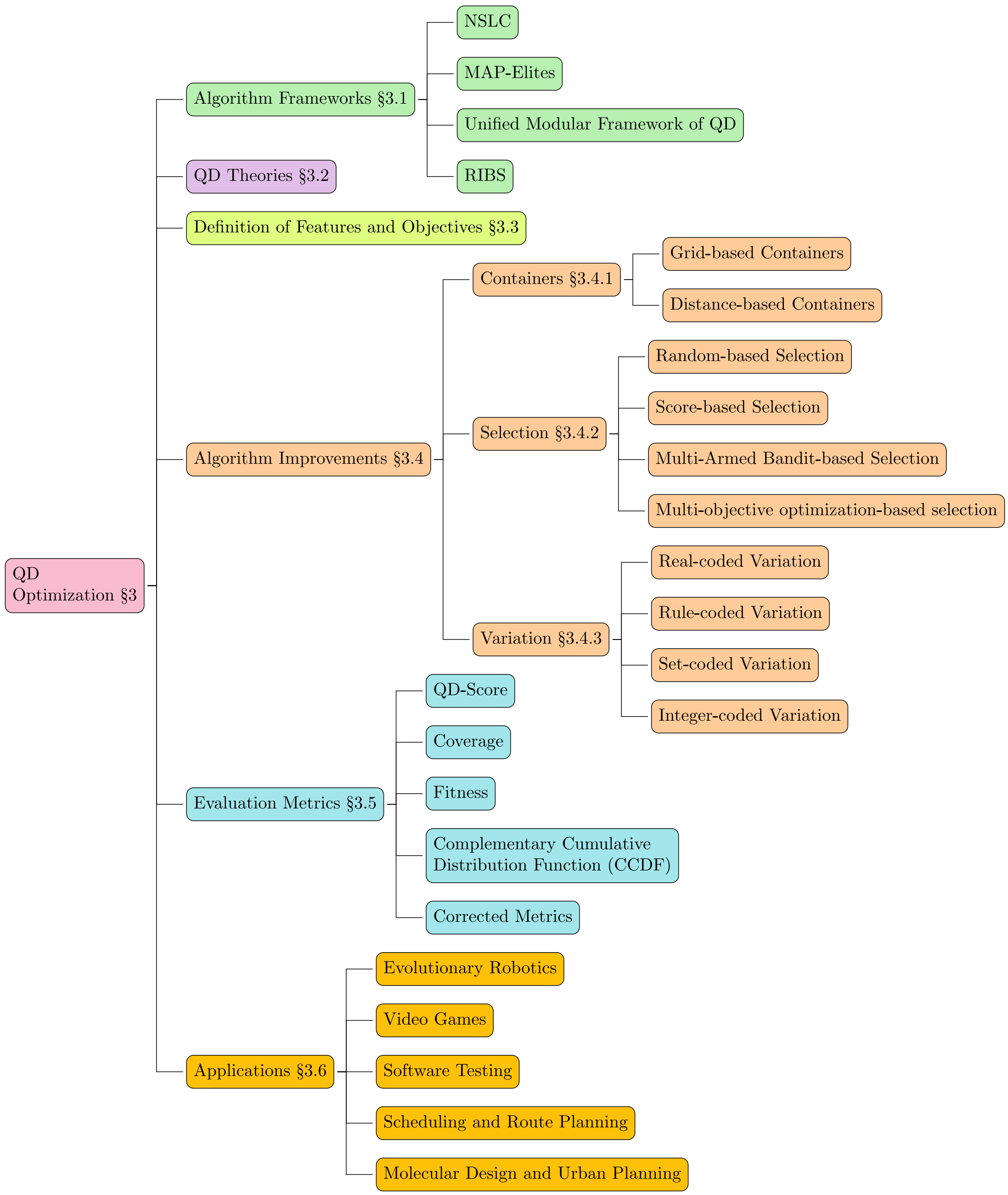


Fig. 5. Taxonomy of QD optimization.

balance helps avoid premature convergence and supports the discovery of a wide range of high-performing solutions across the feature space. NSLC paved the way for later QD algorithms such as MAP-Elites, and remains a foundational approach for promoting both exploration and competitiveness in feature-driven evolutionary search.

NSLC was first applied to species evolution [56], where traits such as height and the number of joints were used as solution features, while walking speed served as the feature of solution quality (performance). By evaluating individuals based on both novelty and local performance, NSLC effectively guided the search toward behaviorally diverse and locally optimal solutions. Consequently, the algorithm produced thousands of unique species, demonstrating its ability to evolve a population that balances diversity and high performance. This dual-objective mechanism allowed NSLC to avoid premature convergence and maintain exploration in deceptive or sparse-reward environments. It facilitated a more structured exploration of the solution space and enabled the discovery of multiple high-quality solutions that differ meaningfully in features. In addition, it also optimized computational resources by focusing on promising regions of the search space, especially when full exploration was impractical due to its vast size [106]. To date, NSLC has been effectively employed to maintain an archive of diverse solutions, and in many cases, it has enhanced goal-oriented evolutionary searches [107], serving as a foundational approach for many modern QD algorithms.

3.1.2. MAP-Elites

NSLC has shown that searching for novel solutions through novelty scoring is highly effective and helps avoid local optima to some extent. Following the idea of the above algorithms, the current successors of NS are called the “QD optimization algorithms [16]” and “illumination algorithms [58]”. It unifies diversity-driven exploration and objective-based optimization to explore the entire feature space, aiming to find

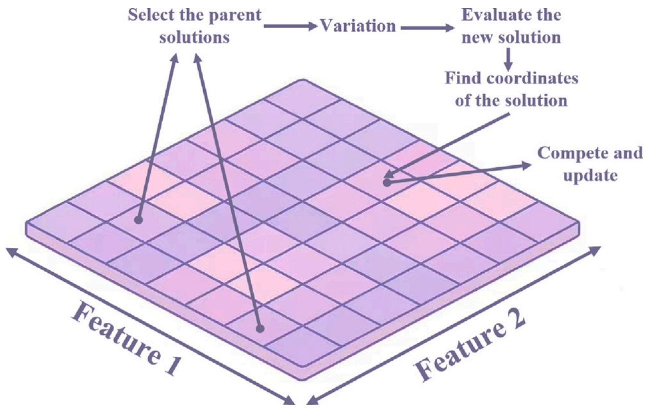


Fig. 6. Visualization of the MAP-Elites update process. Figure is adapted from [109]. Parent solutions are selected from the archive and undergo variation to produce new offspring. These offspring are evaluated and mapped to a specific location in the feature space based on their features. If the new solution outperforms the existing one in its corresponding cell, it replaces the old one in the grid. This iterative process gradually fills the feature space with diverse and high-performing solutions.

a set of high-performing and diverse solutions. MAP-Elites [58] is one of the most representative and well-known QD optimization algorithms, and is originally designed to "illuminate" the landscape of the objective function in a way that is easy to implement and improve [108]. The key distinction between MAP-Elites and NSLC is that the evolving population is represented by the set of solutions stored within the feature space. Operations such as selection and archiving are then carried out directly in feature space.

Algorithm 1 MAP-Elites algorithm

```
Input: batch
Output: Grid 𝒜
procedure MAP-Elites(N)   %N denotes the dimension
  Create empty Grid 𝒜
  for i ← 1 to batch do   %Initialization
      x = random_solution()
      Add_to_Grid(x, 𝒜)
  end for
  while termination is not met do
      x = selection(𝒜)
      x′ = variation(x)
      Add_to_Grid(x′, 𝒜)
  end while
  return 𝒜
procedure Add_to_Grid(x, 𝒜)
  (p, b) ← evaluate(x)   % b is the feature vector
  c ← get_cell_index(b)
  if 𝒜(c) = ∅ or 𝒜(c).p > p then
      𝒜(c) ← p, x
  end if
```

MAP-Elites discretizes the feature space into a grid, which is initially empty. At the beginning, a batch of solutions is generated. Each solution is then evaluated — typically by computing its objective function value — and its features are recorded. Each solution is assigned to a specific grid cell. If the cell is unoccupied, the solution is stored. If it is already occupied, the new solution is evaluated against the existing one, and only the higher-performing solution is kept. Once the initialization is complete, the evolution begins. In this phase, evolution typically starts by randomly selecting a solution from the grid, followed by a variation operation. The new solution, resulting from variation (mutation), is then evaluated and updated in the same way as the previous solutions [8]. The specific visualization process is illustrated in Fig. 6. Moreover, Algorithm 1 outlines the detailed steps of MAP-Elites.

Table 1
The definitions of selection criteria.

Selection criteria	Definition
Novelty Score	The Novelty Score of an individual x is defined as the average distance to its k nearest neighbors in the feature space.
Curiosity Score	The Curiosity Score represents the propensity of an individual to generate offspring that are added to the collection.
Contribution Score	The Contribution Score measures how much a test input contributes to covering new areas in the feature space.

MAP-Elites was first demonstrated across diverse domains, including the evolution of modular neural networks and the design of both simulated and real soft robots [58]. Later, it played a key role in enabling robots to quickly adapt to unexpected damage [13]. Additionally, MAP-Elites was employed to create feature libraries containing various steering gaits and was shown to generate better feature ensembles more efficiently than the BR-Evolution algorithm [105]. In recent years, most real-world problems have been found to exhibit multiple conflicting objectives that need to be optimized. To address this, the MAP-Elites algorithm has been extended to multi-objective environments, leading to the development of Multi-Objective MAP-Elites (MOME). As shown in Fig. 7, the schematic diagram illustrates the storage and update process of MOME. The top of Fig. 7 illustrates the iterative process of MOME, where each cell in the feature space stores a local Pareto front by maximizing its hypervolume. The bottom of Fig. 7 visualizes the feature and objective space coverage, showing that numerous local fronts collectively form a high-performance global Pareto front. This highlights the advantage of feature space coverage in multi-objective optimization.

3.1.3. Unified modular framework of QD

Cully et al. [8] introduced a unified modular framework (UMF) of QD, presenting all QD algorithms as variations of a single high-level approach. This framework highlights how NSLC and MAP-Elites can be formulated as the same QD optimization algorithm by employing different operator combinations. Moreover, it provides a new perspective by combining MAP-Elites and NSLC in different ways, allowing them to complement each other and leverage their respective strengths. Specifically, the framework consists of two key components. The first is a container that collected and stored solutions. The second is a selection operator that chose solutions from the container for modification in the next generation. Unlike traditional EAs, which select only from the current population, this operator considered all stored solutions. This approach allows for broader exploration and a more diverse set of potential solutions.

After randomly generating a set of solutions and storing them in the container, the QD optimization framework repeatedly executes the following four steps:

- The selection operator identifies a set of solutions from the container to undergo variation before the next evaluation.
- The modified solutions are assessed, with their features and performance metrics recorded.
- Each newly evaluated solution is compared against the existing one in the same position to determine if it should be added to the container.
- Various selection criteria, such as Novelty Score [55], Curiosity Score [8] and Contribution Score [110], are updated accordingly. The definitions of these criteria are listed in Table 1.

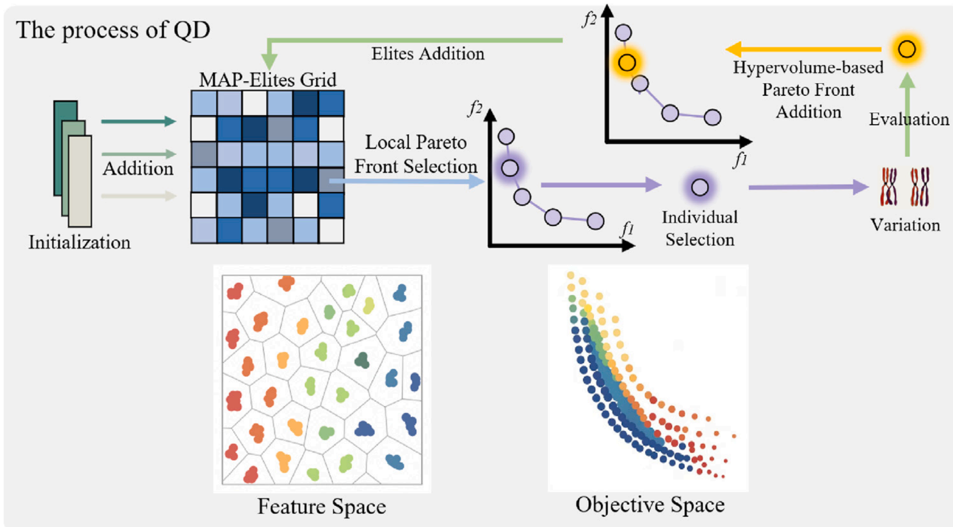


Fig. 7. The schematic diagram of the storage and update process of MOME. Figure is adapted from [67]. The upper part illustrates the full MOME optimization pipeline. At each generation, candidate solutions are evaluated and assigned to cells in the feature space based on their features. Within each cell, the evaluation function seeks to maximize the hypervolume of the local Pareto front, i.e., the set of non-dominated solutions with respect to multiple objectives within that cell. As a result, instead of storing a single best individual, each cell maintains a diverse set of trade-off solutions (a local Pareto front). The bottom figure presents a visualization of the feature space and objective space coverage achieved by MOME in a specific task. Many cells in the feature space store local Pareto fronts, which can be combined to obtain a single high-performance global Pareto front. An interesting phenomenon observed in the original paper is that numerous local cells contribute to the global Pareto front, indicating the benefits of covering the feature space for multi-objective optimization.

Algorithm 2 Unified Modular Framework of QD

Input: Maximum number of iterations I, the number of selected parents N

Output: Container $\mathcal{A}$

1: Initialize an empty Container $\mathcal{A}$
2: **for** $i \leftarrow 1$ **to** I **do**
3:   **if** i = 1 **then**
4:     $S_{parents} \leftarrow$ RandomSolution()
5:     $S_{offspring} \leftarrow$ RandomSolution()
6:   **else**
7:     $S_{parents} \leftarrow$ Selection(N, $\mathcal{A}$)
8:     $S_{offspring} \leftarrow$ Variation($S_{parents}$)
9:   **end if**
10:   **for each** $x \in S_{offspring}$ **do**
11:     $\{f_x, \boldsymbol{b}_x\} \leftarrow$ Evaluation(x)
12:     **if** InsertionCriterion(x, $\mathcal{A}$) **then**
13:       Add_to_Container (x, $\mathcal{A}$)
14:       **if** CreditAssignment is defined **then**
15:         CreditAssignment($S_{parents}$, Success = True) % Reward or penalize parent based on offspring
16:       **end if**
17:     **else**
18:       **if** CreditAssignment is defined **then**
19:         CreditAssignment($S_{parents}$, Success = False)
20:       **end if**
21:     **end if**
22:     UpdateContainer ($\mathcal{A}$)
23:   **end for**
24: **end for**
25: **return** $\mathcal{A}$

However, in the UMF, Cully et al. [8] used a scoring mechanism based on "Curiosity Score", which was not abstracted or parameterized. In this paper, the framework was further abstracted to make it more inclusive. For further details, refer to the pseudo-code in Algorithm 2. In lines 4–5, a batch of parent and offspring solutions is randomly generated. Lines 7–8 describe the selection of individuals from the container as parents, followed by generating offspring through crossover or mutation. Here, mutation is explicitly identified as a core building block of the framework, highlighting its essential role in driving exploration, which was not clearly emphasized in earlier frameworks. Subsequently, each offspring undergoes evaluation to determine its performance and features, which are recorded (line 11). Lines 12–21 indicate that if the user defines a "contribution evaluation mechanism" (which can be any user-defined scoring mechanism), then the parent individual that produces an accepted solution is rewarded. If x is not accepted, then the parent individual is penalized to steer the search away from invalid regions. The function "Add_to_Container" (line 13) determines whether a solution should be added to the container. Finally, the container is updated (line 22).

3.1.4. RIBS

To accommodate the growing diversity of QD algorithms, recent work has proposed a modular framework called RIBS (stems from the title of Fontaine et al. [63], Covariance Matrix Adaptation for the **R**apid **I**llumination of **B**ehavior **S**pace). Unlike the earlier UMF that mainly connects MAP-Elites and NS through selection and variation, RIBS is designed to support a broader class of QD algorithms that incorporate advanced optimization techniques such as evolution strategies [111,112], gradient-based methods [24,84], and Bayesian Optimization [109].

The RIBS framework abstracts QD into three components: (1) an archive for storing high-performing and diverse solutions, (2) one or more emitters responsible for generating new candidate solutions, and (3) a scheduler that orchestrates interactions between the archive and emitters. This architecture not only unifies existing QD algorithms but also enables the development of new ones by mixing and matching modular components. Importantly, the ability of RIBS to support multiple emitter populations is critical for methods like CMA-ME [63], which rely on running multiple CMA-ES instances in parallel.

3.2. QD theories

While experimental performance has driven most advances in QD optimization, theoretical understanding remains limited but increasingly important. A small but growing body of work has begun to

Table 2
Summary of representative theoretical studies on QD algorithms.

Reference	Focus/Problem domain	Main contributions	Key theoretical results
Nikfarjam et al. (2022) [113]	Knapsack problem	First theoretical analysis of MAP-Elites on knapsack; feature space mimics dynamic programming	MAP-Elites finds optimal solutions in pseudo-polynomial expected time
Bossek et al. (2023) [69]	Pseudo-Boolean and combinatorial optimization (e.g., OneMax, spanning trees)	Runtime analysis of QD over granular feature spaces comparison with GSEMO	Tight upper bounds on feature space coverage time; Efficient $(1-1/e)$-approximation for monotone submodular functions; MST discovery in $O(n^2\ m)$ time
Qian et al. (2024) [72]	NP-hard problems: submodular maximization, set coverage	Comparison of QD (MAP-Elites) vs. traditional evolutionary algorithms	MAP-Elites can effectively avoid local optima. MAP-Elites achieves polynomial-time approximation, while $(\mu+1)$-EA may require exponential time

formalize QD features, convergence properties, and comparative advantages over traditional evolutionary approaches. Nikfarjam et al. [113] were the first to analyze the performance of MAP-Elites on the knapsack problem. They demonstrated that, with a well-constructed two-dimensional feature space, MAP-Elites could emulate dynamic programming and reach an optimal solution in pseudo-polynomial expected time. Bossek et al. [69] studied the runtime of MAP-Elites on pseudo-Boolean and combinatorial optimization problems, providing tight upper bounds for the expected time to cover the feature space. In the analysis, they examined a simple QD algorithm operating over a feature space defined by unitation (i.e., the number of 1 s in the binary string) and analyzed the impact of the granularity parameter k on coverage time. Here, k defines the granularity of the feature space partitioning, i.e., the number of intervals into which the range of feature values is divided. For OneMax and other monotone functions, they proved that QD could achieve complete feature space coverage efficiently, and they also derived runtime bounds for $(1-1/e)$-approximation on monotone submodular maximization with uniform cardinality constraints. Additionally, by redefining the feature space in terms of the number of connected components in edge-weighted graphs, they showed that QD is capable of finding minimum spanning trees in $O(n^2m)$ time—mimicking Kruskal's algorithm. In this setting, n refers to the number of nodes in the graph and m to the number of edges. Notably, they observed that while QD shares similarities with the Generalized Simple Evolutionary Multi-Objective Optimizer (GSEMO) in principle, it requires no Pareto dominance checks, making implementation simpler. Qian et al. [72] further established the theoretical advantages of QD over traditional methods. For two NP-hard problems — monotone submodular maximization under cardinality constraints and the set coverage problem — they proved that MAP-Elites could achieve (asymptotically) optimal polynomial-time approximation ratios, while the $(\mu + 1)$-EA may require exponential time. Their work suggests that simultaneously optimizing for performance and diversity helps escape local optima and leads to more globally effective solutions. The above work has been summarised in Table 2.

3.3. Definition of features and objectives

As defined in [8], QD is designed to generate a large set of solutions that are both diverse and high-performing, with coverage spanning a specified region of the feature space. Before illustrating the QD algorithm, it is critical to define both the objective space and the feature space relevant to a specific problem. However, selecting appropriate feature descriptors and fitness functions remains a central open question in QD research.

In traditional optimization problems, the objective function is typically well-defined, leaving little ambiguity about what constitutes a "better" solution. In contrast, QD often operates in domains where performance is only one aspect of interest, and the definition of diversity — encoded through feature descriptors — plays an equally important role. For instance, in a robotic locomotion task, the fitness function (e.g., walking speed or energy efficiency) may be straightforward to specify, yet the choice of behavioral features (e.g., gait symmetry, joint usage, or body posture) is largely arbitrary and heavily influences the diversity of the resulting solutions. Conversely, in a creative design or procedural generation task, the behavioral space may be naturally defined by the structure or composition of the artifacts, while the performance objective — such as "aesthetic quality" or "user satisfaction"— is more abstract and difficult to quantify.

Highlighting this ambiguity clarifies one of the defining distinctions between QD and traditional EAs: rather than optimizing a single scalar objective, QD simultaneously explores and organizes a set of diverse solutions. Next, we present an overview of the definitions of objectives and feature spaces in several representative application domains.

(1) Evolutionary Robotics: In evolutionary robotics, objectives were typically designed to promote adaptability, resilience, and generalization. For instance, Cully et al. [13] emphasized post-damage adaptability, guiding robots to select compensatory features via a precomputed behavioral repertoire. Features were often derived from motion characteristics — such as duty factors for legged locomotion or end-effector trajectories in robotic arms — to capture feature diversity independent of specific task rewards. Additionally, optimizing grasping features was a common goal, particularly in improving grasp success rates across different robot-gripper configurations and standard objects while maintaining both diversity and performance in grasping trajectories [104]. Nilsson et al. [65] introduced PGA-MAP-Elites, integrating policy gradient-based variation into MAP-Elites to improve early-stage search efficiency. In terms of feature design, their work continued to rely on commonly used descriptors in evolutionary robotics. At the objective level, they extended QD optimization to handle more complex robotics goals that combined speed maximization with energy minimization. Subsequently, Flageat et al. [35] conducted an empirical study of PGA-MAP-Elites in uncertain environments, specifically examining how these objectives and features influence policy performance and search efficiency under stochastic conditions.

In addition, unsupervised QD has been developed. Unlike traditional QD methods that rely on manually defined features, this approach enables robots to autonomously construct features by interacting with the environment and applying unsupervised dimensionality reduction techniques (e.g., autoencoders) [114]. Representative of this line, AUtonomous RObots that Realize their Abilities (AURORA) [114] learns features directly from sensory data through autoencoders, and has shown promising results across various robotic tasks by significantly improving autonomous exploration and skill acquisition in high-dimensional and unstructured domains. Another unsupervised QD approach, Task Agnostic Exploration of Outcome Space through Novelty and Surprise (TAXONS), aims to discover a repertoire of diverse policies in sparse-reward environments without any task-specific prior knowledge [115]. Its optimization objective is to uniformly explore the outcome space using novelty and surprise as intrinsic rewards. Features are learned directly from high-dimensional visual observations via an autoencoder, and both the latent representation and reconstruction error are used to measure policy diversity. Building on this direction, Relevance-guided Unsupervised Discovery of Abilities (RUDA) [116] enhances unsupervised QD by aligning feature diversity with task relevance. The optimization objective is to generate behavior repertoires that concentrate solutions in regions of the feature space most useful for downstream tasks. To achieve this, RUDA automatically learns the features from sensory data and guides the search using a relevance-weighted diversity metric. QD through Human Feedback (QDHF) [117] optimizes for diverse and high-quality solutions by learning diversity metrics directly from human feedback. Its core idea is to align latent representations with human-perceived differences using contrastive learning, enabling QD algorithms to explore semantically meaningful variations without manually defined features.

(2) Video Games: In procedural content generation, objective functions were crafted to balance novelty and functionality, often penalizing oscillatory features, inefficiency, or structural infeasibility.

For example, in *Minecraft*, Medina et al. [118] introduced objectives that promoted sustained flight while discouraging excessive block usage. Feature spaces were constructed from structural features such as block distributions, symmetry, and path complexity, allowing the algorithm to maintain creative diversity while meeting playability constraints. In another *Minecraft* application [119], objectives were similarly defined to encourage creative yet feasible content generation under gameplay constraints, with feature spaces designed around structural and behavioral metrics.

Beyond Minecraft, other domains have applied QD to procedural content generation. In real-time level generation, features such as geometric layout and challenge difficulty were employed to guide the production of diverse, functional content [120]. Zhang et al. [42] addressed automated Hearthstone deckbuilding, where deck strategy style, card composition diversity, and game tempo were explicitly used as features to structure the MAP-Elites archive and promote diversity. The optimization aimed to maximize deck win rate while maintaining diversity across these strategic dimensions.

(3) Software Testing: In software testing, objectives focused on maximizing coverage and minimizing test suite cost and redundancy. Xiang et al. [11,121] optimized for t-wise coverage and cost-effectiveness, with features constructed from test execution paths, case diversity, and suite compactness. Akbarova et al. [122] proposed Software Edge-case Testing via Boundary Value Exploration (SETBVE), where input/output features were used to structure the archive and guide exploration of underexplored boundary regions. Additionally, Mazouni et al. [123] reformulated policy testing in RL as a QD optimization problem, with the objective of minimizing the cumulative reward achieved by the policy under test. Features were derived from execution traces under different initial states, capturing diverse failure-inducing behaviors.

(4) Scheduling and Route Planning: For scheduling tasks like flexible job-shop scheduling (FJSP), objective functions typically minimized makespan, flow time, or energy consumption. Qin et al. [12] proposed features such as the number of job transports and machine idle times, which correlated strongly with system throughput and resource usage. These features provided meaningful dimensions for structuring the search space and supported diverse yet efficient schedule generation. In broader routing applications, metrics such as employee cost, emissions, and car usage were used to construct feature spaces reflecting operational and environmental diversity [124]. Urquhart et al. [51] also introduced the micro-depot Vehicle Routing Problem (VRP), in which a central supply vehicle delivers to distributed micro-depots, and last-mile deliveries are fulfilled by couriers using electric vehicles, bicycles, or on foot. To characterize solution diversity, they defined features such as total distance traveled, total emissions, and the number of couriers used. Marrero et al. [53] applied QD to generate benchmark instances for the Knapsack Problem, aiming to improve diversity and solver discrimination. The optimization goals were to maximize profit, minimize weights, and optimize the profit-to-weight ratio, while feature selection focused on profits, program efficiency, and weight distribution.

(5) Molecular Design and Urban Planning: In molecular discovery, QD methods aimed to explore chemically meaningful regions of the solution space by optimizing across multiple material properties. Janmohamed et al. [125] proposed a multi-objective QD framework for crystal structure prediction, using features such as toughness, magnetism, and thermoelectric efficiency. Verhellen et al. [46] introduced Bayesian Illumination to improve generative model diversity by incorporating structure-based features into the search process. In discrete molecular representation tasks, Boige et al. [47] designed features that reflected structural and functional properties, enabling QD to operate effectively in non-differentiable, combinatorial spaces. Galanos et al. [49] introduced ARCH-Elites, a MAP-Elites–based framework for urban design that optimized Floor Space Index (FSI) — a feature of built volume per land area — while navigating trade-offs between pedestrian comfort and safety under extreme wind conditions. The feature dimensions were (1) the percentage of open space deemed suitable for comfortable sitting, and (2) the area exposed to dangerously high winds, both predicted via a surrogate wind-flow model. Mosphilis et al. [48] optimized placement of surveillance cameras in commercial chicken farms using CMA-ES, MAP-Elites, and CMA-ME. The optimization aimed to maximize coverage for effective monitoring, with features including coverage area and solution diversity of camera configurations.

Summary: Table 3 summarizes the optimization objectives and key feature representations of QD across various domains. For further details, readers are encouraged to refer to the original papers. Since each study defines its own optimization objectives and features, it is rare to find identical implementations of QD. These differences arise from the distinct practical requirements of various fields, leading to diverse implementations of QD. For example, in evolutionary robotics, optimization often prioritizes task success rates and diversity, whereas in neural network optimization, the emphasis often lies in balancing exploration efficiency with computational cost. This flexibility enables QD to be effectively applied across diverse research domains, demonstrating strong adaptability and generalization in addressing complex and practical optimization problems. As QD advances, its potential applications in emerging fields are expected to grow even further.

3.4. Algorithm improvements

This section presents improvement strategies, organized into three subsections: container design, selection mechanisms, and variation operators.

3.4.1. Containers

The container is responsible for storing all discovered solutions, retaining only those that are both diverse and high-performing. In QD optimization, two primary types of containers are commonly used: grid-based and distance-based. MAP-Elites employs an N-dimensional grid to discretize the feature space, while NSLC (Novelty Search with Local Competition) uses an unstructured archive that relies on Euclidean distance between features. NSLC has exerted a far-reaching impact on the field. Its archive mechanism — independent of any predefined grid structure — provides a valuable alternative approach to organizing diverse solutions. Importantly, the choice between grid-based and distance-based containers is not tied to specific problem domains; both structured and unstructured containers can be effectively applied to a wide range of optimization tasks. Therefore, this subsection presents containers by type rather than by application domain. The following discussions describe the implementation details of both grid-based and distance-based approaches.

(1) Grid-based Container: MAP-Elites stores solutions within an N-dimensional grid, where each dimension represents a feature of solutions [58]. This structure partitions the feature space into discrete cells, with each cell storing a solution corresponding to a specific feature. Some implementations add “depth” to grid containers [32], enabling greater flexibility for complex requirements. In multi-objective optimization, a depth-based grid allows each cell to maintain multiple

Table 3
Summary of definitions in objectives and features.

Applications	Objectives	Features	Optimization tasks
Evolutionary Robotics	Optimize the variance of the joint angles	End-effector position of the redundant arm	Redundant arm [126]
	Maximize the throughput of the warehouse layout	Number of connected shelf components and average length of tasks	Workstation [127]
	Maximize grasp success rate	Grasp trajectories and positions of end effector	3 different arms and grippers [128]
	Maximize grasp success rate	Grasp poses, prior-based grasp features	Four grippers with 2-to-5 fingers [26]
	Optimize the locomotion speed and robot-specific energy penalty	Successor features, skill-conditioned policy, and locomotion patterns	Six challenging continuous-control locomotion tasks [129]
	Optimize the efficiency of policy and sample	Duty factor features for robotic tasks	Robot motion features [35,65]
Video Games	Optimize flying capability, stability, and diverse structures	Flight height and duration, movement pattern, and structural diversity	Flying machines in Minecraft [119]
	Maximize level generation speed, enhance level diversity and quality	Geometric layout of levels, playability, and level challenge difficulty	Video game levels [120]
	Improve sample efficiency and maximize deck win rate	Deck strategy style, card composition, and game tempo	Automated Hearthstone deckbuilding [42]
	Maximize solution quality and match user preferences	Feature diversity and target feature matching	Construction of Gaussian-process surrogate models [43]
Software Testing	Maximize T-wise coverage and novelty	Test suite size	Automatic test case generation [10]
	Test suite diversity and dissimilarity	Test suite size	Automatic test case generation [11]
	Improve the diversity of boundary samples	Input/Output features	Software Testing via Boundary Value Exploration [122]
	Minimize the cumulative reward	Behavioral trajectories	Automatic test case generation [123]
Scheduling and Route Planning	Minimize the total distance traveled	Emissions, employee cost, travel cost, and car travel	Workforce scheduling and routing [124]
	Minimize total cost	Total distance traveled, emissions, and couriers	Vehicle routing problems [51]
	Optimize profit-to-weight ratio	Profits, program efficiency, and weight distribution	Knapsack problems [53]
	Minimize the energy consumption	Machine idle and vehicle times	Flexible job-shop scheduling problem [12,74]
Molecular Design and Urban Planning	Optimize structural and thermoelectric efficiency	Crystalline materials, toughness, and hardness	Crystal structure prediction [125]
	Maximize the performance of molecules	Structural diversity of molecules	Molecular design [46,47]
	Optimize Floor Space Index (FSI)	Open-space ratio and wind-exposure area	Urban design [49]
	Maximize coverage for effective monitoring	Coverage area and camera configurations	Placement of cameras in chicken farms [48]

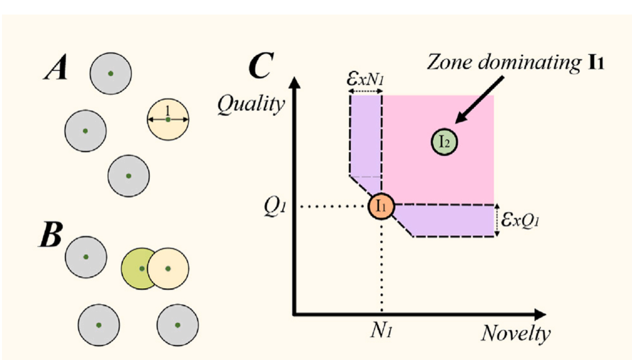


Fig. 8. Management of unstructured archiving solutions. This figure is adapted from [8]. (A) If the new solution is farther than l from its nearest neighbor, it is directly added to the container. (B) If the distance is less than l, meaning the solutions overlap, the new solution competes with its neighbor. If it performs better, it replaces the existing one. (C) Under strict ϵ-domination, a solution is considered superior if its improvement in the target objective exceeds the decline in other objectives within a predefined threshold ϵ.

Pareto non-dominated solutions, acknowledging that no single optimal solution exists [130]. Another approach retains the original grid structure while storing only one solution per cell, using the knee point method to better meet practical needs [131]. For high-dimensional feature spaces, centroidal Voronoi tessellation is used to create equal-volume cells, regardless of dimensionality [61,132]. The following section explains how solutions are stored in the grid and how novelty scores are computed.

If a grid cell is empty, the new solution is directly added. If the cell is occupied, the performance of the new solution (typically measured by its objective value) is compared with the existing one, and the better-performing solution is retained. In multi-objective problems, when using a depth-based grid to store the Pareto set, a solution is added if it improves a performance metric such as hypervolume (HV). This mechanism aligns with the single-objective case, where selection is based on the objective function value. When applying the knee point method [131], a hyperplane is constructed using extreme points, and the distance of each solution from this hyperplane is calculated. Solutions farther from the hyperplane are more likely to be identified as knee points, especially in minimization problems. If the user provides weights for different objectives, a weighted sum approach can be used to select the most satisfactory solution. If no weights are specified, a balanced solution that performs well across all objectives is chosen. Despite these variations, the grid structure remains consistent with that used in single-objective optimization.

(2) Distance-based Container: The distance archive stores solutions in an unstructured set using Euclidean distance and features. Unlike the grid-based storage in MAP-Elites, which discretizes the feature space, the distance archive forms its structure dynamically based on the distribution of solutions.

The archiving process begins with a user-defined distance threshold. If a new solution is sufficiently distant from all existing solutions — exceeding the threshold — it is added to the archive. Alternatively, if a new solution outperforms its closest neighbor, it replaces the existing solution. While traditional NS archive management methods are straightforward, they have notable drawbacks [133]. One issue is uneven solution distribution, where certain regions become overly dense due to the k-nearest neighbor scoring method [105]. Another issue

Table 4
Comparison between grid-based and distance-based containers in QD optimization.

Aspect	Grid-based container (e.g., MAP-Elites)	Distance-based container (e.g., NSLC)
Structure	A discrete N-dimensional grid partitions the feature space into fixed cells, with each cell typically storing one or more solutions based on a predefined grid structure.	An unstructured, distance-based container organizes solutions dynamically using Euclidean distance in the feature space.
Insertion rule	If the corresponding cell is empty, insert directly; if occupied, compare objective values (or multi-objective indicators) to decide replacement.	Add if the solution is farther than a distance threshold from all others, or replace nearest if superior in quality or novelty.
Behavior coverage strategy	It ensures uniform coverage of the feature space through discretized cells and can be extended to high-dimensional cases using centroidal Voronoi tessellation.	It adapts the archive structure to the empirical distribution of solutions and aims to maintain coverage by dynamically controlling the spread based on distance metrics.
Common issues	It requires predefined discretization of the feature space, which may restrict adaptability.	It suffers from poor exploitation, misleading diversity in sparse or high-dimensional spaces, and increased archive maintenance overhead.

is boundary erosion, where the archive contracts inward as higher-quality solutions replace lower-quality ones, reducing overall diversity [133]. To address these limitations, Cully et al. [8] proposed a new archive management strategy that regulated solution density based on proximity to neighbors. Their method incorporated Pareto dominance principles and a relaxed dominance variant to refine replacement decisions. Specifically, if a new solution is close to an existing one but exhibits significantly better quality or novelty, it may replace the current solution, ensuring a balance between archive coverage and solution quality. Following [8], the process is illustrated in Fig. 8. The results demonstrate that this approach effectively mitigates traditional limitations, preserving high-quality solutions while maintaining a more evenly distributed archive. Additionally, a recent study introduced Dominated Novelty Search (DNS) [134], which replaces explicit container structures with dynamic fitness transformations, removing the need for predefined grids or archives while maintaining diversity and performance.

Table 4 shows a comparison of the above two containers. Each container type has its own strengths and limitations. Grid-based containers provide a structured and efficient storage method but require predefined discretization of the feature space, which may restrict adaptability. In contrast, archive-based containers offer greater flexibility by relying solely on pairwise distances between individuals without imposing a fixed space structure. However, this adaptability also introduces complexity, as the algorithm must dynamically determine how to organize and maintain the archive effectively.

3.4.2. Selection

Selection plays a pivotal role in QD optimization, as it determines which solutions from the archive are chosen to undergo variation and generate new offspring. Different selection strategies influence the balance between exploration and exploitation, ultimately affecting the diversity and quality. Over the years, a variety of selection mechanisms have been proposed, ranging from simple random sampling to more sophisticated score-based, adaptive, and multi-objective methods. Table 5 provides a concise comparison of these approaches, highlighting their underlying principles, advantages, and representative implementations.

(1) Random-based selection: Random-based selection is the most fundamental and widely adopted selection strategy in QD algorithms, serving as the default mechanism in many canonical frameworks. One of the most representative algorithms, MAP-Elites [58], typically employs uniform random sampling from the current archive to select parents for variation. This approach assumes no explicit preference among archived solutions and treats each with equal opportunity, regardless of their fitness or location in the feature space.

The appeal of random selection lies not only in its simplicity, low computational cost, and domain-agnostic applicability, but more importantly in its non-biased nature. In MAP-Elites, random selection was purposefully adopted to avoid premature convergence and to promote the emergence of stepping stones—intermediate solutions that may lead to high-performing regions of the feature space. It requires no additional bookkeeping or estimation of utility metrics such as fitness trends, novelty gradients, and reward expectations. As a result, it became a standard choice in early QD implementations and remains effective, especially in settings with limited prior knowledge or constrained computational budgets. Moreover, random-based selection introduces an implicit uniform exploration pressure across the archive. This property is particularly beneficial in the early stages of QD search, where maintaining coverage and avoiding premature convergence is critical. Despite its simplicity, random selection remains surprisingly robust and often competitive, especially when paired with variation operators that implicitly exploit structure (e.g., directional mutation or learned encodings [74]). In practice, random selection is also commonly used as a baseline for evaluating more sophisticated selection policies, such as score-based selection [8,55,110], novelty-guided mechanisms, or adaptive strategies like Multi-Armed Bandit-based selection [23]. Its ubiquity and neutrality make it a foundational component of QD.

(2) Score-based selection: Score-based selection evaluates the potential of individuals or regions in the archive using explicitly defined metrics and chooses parents accordingly, rather than sampling uniformly at random. This approach introduces a more informed selection process that can improve search efficiency by favoring promising areas of the search or feature space.

Several metrics have been proposed to quantify individual or cell potential, including the *novelty score* [55], *contribution score* [110], and *curiosity score* [8]. These metrics can be used to guide parent selection either through score-proportionate sampling (e.g., roulette wheel selection) or adaptive resource allocation strategies [136]. For instance, in NSLC, novelty and local competition scores are used to bias selection toward behaviorally novel and locally dominant individuals within a neighborhood. Similarly, in MAP-Elites variants that incorporate score-based mechanisms, cells or individuals with higher historical contribution or curiosity scores are prioritized.

The novelty score was originally defined by Lehman and Stanley [55] as the average distance to an individual's k-nearest neighbors in feature space. This concept can also be adapted for archive grids, allowing novelty to be featured based on the spatial structure of the grid. Cully et al. [8] introduced a method for computing the Novelty Score based on the inherent structure of the grid. Unlike [55], which defined as the average distance to the k-nearest neighbors, this approach determines novelty by assessing the density of occupied cells within a subgrid surrounding the solution. The size of this subgrid is controlled by a parameter specifying a range of $\pm k$ cells. In this formulation, the novelty score is minimized to encourage diversity. Taking the curiosity score as an example, if a selected cell produces offspring that successfully enter the grid, its score increases by 1; otherwise, it decreases by 0.5. A structured selection strategy is then applied based on cell potential: (1) If one cell represents a newly discovered region and the other contains improved solutions, the newly discovered cell is prioritized; (2) If both are newly discovered, the one with higher-quality individuals is chosen; (3) If both contain improved solutions,

Table 5
Comparison of selection strategies in QD optimization.

Selection strategy	Core principle	Advantages	Representative methods
Random-based selection	The algorithm uniformly samples individuals or cells from the archive without applying any preference.	This strategy is simple, computationally efficient, and domain-independent, which encourages unbiased exploration.	MAP-Elites [58]
Score-based selection	The selection process prioritizes individuals based on utility scores such as novelty, curiosity, or contribution.	By leveraging historical feedback, this method improves search efficiency and better balances diversity with quality.	NSLC [56], Curiosity Score [8], Contribution Score [110]
Multi-Armed Bandit-based selection	The selection strategy models each region or cell as a bandit arm and applies upper-confidence bound or Thompson sampling.	This approach adaptively balances exploration and exploitation, making it suitable for sparse or high-dimensional archives.	UCB-MAP-Elites [23], Dobby Slot Machine [135]
Multi-Objective Optimization-based selection	The algorithm samples cells uniformly and selects individuals from the local Pareto front maintained within each cell.	This method is designed for cases where other algorithms are not directly applicable, standing out as a general approach that can handle scenarios with conflicting goals.	MOME [67]

the cell with the higher improvement rate is selected. By integrating scoring mechanisms into the selection process, QD algorithms can balance diversity and quality more effectively.

(3) Multi-Armed Bandit-based selection: In QD optimization, the challenge of balancing exploration and exploitation is particularly prominent during the parent selection phase. A promising direction to address this challenge is to model selection as a multi-armed bandit (MAB) problem. The MAB, commonly used in reinforcement learning, is designed for sequential decision-making under uncertainty. In this setting, each "arm" (i.e., choice) is associated with an unknown reward distribution. This formulation naturally aligns with the need to dynamically prioritize regions of the feature space that may yield novel or high-quality solutions.

Recent work has explored this idea explicitly. Sfikas et al. [23] integrated MAB principles into MAP-Elites by treating each cell as an arm and applying upper-confidence bound (UCB) strategies to guide parent selection. Instead of relying solely on fitness or novelty, the selection policy incorporates indirect features such as offspring survival rate, balancing risk and reward across the archive. Empirical results across multiple domains demonstrate that this approach accelerates the discovery of valuable regions and enhances the overall quality-diversity trade-off in the solution set. For example, Yang et al. [135] provided a review of MAB-inspired strategies, which are often referred to as Dobby Slot Machine algorithms and have been applied in online gaming. These studies highlight the value of MAB in dynamically steering attention toward promising areas without requiring complete environmental knowledge. When applied to QD, MAB-based selection mechanisms can more efficiently allocate search resources, particularly in high-dimensional or constrained feature spaces.

Overall, MAB offers adaptive selection in QD, supporting more intelligent exploration of the archive and potentially improving convergence speed, archive coverage, and solution quality.

(4) Multi-Objective Optimization-based Selection: Multi-objective MAP-Elites (MOME) [67] extends the classical QD framework by integrating Pareto-based selection principles. The core idea of MOME is to simultaneously promote feature diversity and optimize multiple objectives by maintaining a Pareto front within each cell of the feature space. According to [67], MOME adopts a selection strategy where cells are uniformly sampled from the grid to ensure feature diversity, and individuals are then sampled from the Pareto front within each cell. This approach emphasizes diversity across the feature space while maintaining multiple trade-offs within each region.

A more advanced variant, MOME with Policy-Gradient Assistance and Crowding-based Exploration (MOME-PGX), proposed in [137], improves over MOME by employing crowding-distance-based selection within the Pareto front. This mechanism increases exploration pressure and helps distribute solutions more evenly along the Pareto front. MOME-PGX also incorporates policy gradient-based mutation to boost sample efficiency and optimization performance in high-dimensional tasks.

Table 6
Overview of variation methods in QD algorithms by encoding type.

Encoding type	Variation methods
Real-coded	This encoding is common in continuous domains. Variation methods include Gaussian mutation [17], CMA-ES [80], directional mutation [138], and adaptive mutation [139]. Latent-space sampling with variational autoencoders [81] and emitter-based exploration in CMA-ME [63] further improve search efficiency in high-dimensional control tasks [13,140]. In addition, recent approaches like CMA-MEGA [24] leverage the gradient of the objective and features to guide variation, offering enhanced performance in differentiable settings.
Rule-coded	This encoding is used in procedural content generation, where solutions are represented by rule sets or grammar-based generators [141,142]. Mutation operates by adding or removing constraints from the rule set [118], and is often paired with novelty-driven strategies to enhance both structural and feature diversity.
Set-coded	This encoding applies to tasks like test suite generation, where each solution is a set of test cases. Variation includes addition, removal, and substitution of test cases [11], with operator selection adaptively controlled based on the current suite size to maintain diversity across the archive.
Integer-coded	This encoding suits discrete and combinatorial problems such as scheduling, routing, and hybrid optimization. Common operators include gene-level swaps, sequence perturbations, and segment relocations [51,124,143], along with grammar-based tree edits [52]. In hybrid tasks like the TTP, routing crossovers are combined with dynamic programming for joint optimization [144]. Projection-based variation is also used to promote instance-level diversity [53].

By combining Pareto-based selection with QD principles, and further enhancing selection via crowding distance as in [137], these multi-objective QD algorithms provide an effective framework for discovering diverse and competitive solutions in problems involving conflicting objectives.

3.4.3. Variation

Variation operators play a critical role in QD optimization by generating new solutions that expand the coverage of the feature space. Their design is typically problem-specific and depends on the solution representation or encoding. This section categorizes variation strategies into four major encoding types: real-coded, rule-coded, set-coded, and integer-coded to highlight how operator design is tailored to each representation.

(1) Real-coded Variation: These strategies are commonly adopted in continuous control and robotic domains, where solution representations consist of real-valued parameters. In such domains, variation operators must handle high-dimensional, continuous search spaces while preserving feature feasibility. Distribution-based techniques like CMA-ES [80] are widely used for their ability to adapt search distributions around elite solutions [17]. Directional variation [138] enhances exploration by generating offspring solutions that combine isotropic noise with directional perturbations based on the relative position of elite pairs. Given two elites $\theta_i^{(t)}$ and $\theta_j^{(t)}$, a new candidate $\theta_i^{(t+1)}$ is generated as:

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \sigma_1 \mathcal{N}(\mathbf{0}, \mathbf{I}) + \sigma_2(\theta_j^{(t)} - \theta_i^{(t)})\mathcal{N}(0, 1) \tag{4}$$

where σ_1 controls isotropic Gaussian noise and σ_2 scales directional perturbation aligned with $\theta_j^{(t)} - \theta_i^{(t)}$. This operator has shown strong empirical performance across various QD domains, offering a computationally simple yet effective way to introduce correlation-aware variation.

Adaptive mutation [139] dynamically adjusts mutation parameters based on search progress. Building on black-box strategies, Fontaine et al. [63] proposed CMA-ME, which integrated CMA-ES with MAP-Elites via emitters for optimization (local refinement), improvement (boundary pushing), and random exploration. In parallel, rather than extending CMA-ME, another line of work has explored the use of gradient information in white-box domains, where both objectives and features are differentiable. Specifically, Fontaine et al. [24] introduced MAP-Elites via a Gradient Arborescence (MEGA), which leveraged gradients of the objective and feature functions to guide variation steps. Unlike CMA-ME, where CMA-ES directly serves as a mutation operator, CMA-MEGA employs ES to optimize the weighting of multiple gradients, representing a distinct intuition about how to drive variation.

Within the subfield of QD Reinforcement Learning (QD-RL), both PGA-MAP-Elites [65] and QD-PG [145] adopt RL-based policy-gradient methods as mutation operators. Specifically, PGA-MAP-Elites introduced a hybrid variation strategy that combined traditional genetic mutation with policy gradient updates derived from a critic network. In this approach, controller parameters were optimized via several steps of deterministic policy gradient ascent to efficiently improve high-dimensional neural network policies. Similarly, QD-PG applied two gradient-based mutation operators and introduced a Diversity Policy Gradient that used time-step-level information to guide exploration. These RL-based mutation strategies proved especially effective in tasks such as evolving locomotion features, robot gait adaptation [13], and simulating virtual agents [56,140]. More recently, QD Imitation Learning (QD-IL) frameworks [146] have been proposed, which integrate adversarial imitation learning with QD principles. These methods encourage feature-level exploration via reward bonuses and generalized expert knowledge through feature-conditioned discriminators, enabling agents to learn a wide range of skills from limited demonstrations.

(2) Rule-coded Variation: These strategies are prevalent in procedural content generation and video game design, where solutions represent generative rules rather than raw outputs. Variation is applied to abstract rule systems, grammar trees, or constraint-based configurations that define game levels or mechanics [141,142]. This enables search within the generative design space rather than on the final product, facilitating creativity while maintaining playability. Medina et al. [118] proposed mutation operators that modify procedural rules by adjusting constraints or toggling components within rule genomes, producing diverse layouts with varying complexity and gameplay. In this domain, maintaining novelty is crucial to avoid repetitive or trivial outputs, so variation operators are often paired with novelty-driven scoring or diversity filters to prioritize structurally unique yet functionally coherent designs. Rule-coded variation promotes scalable, interpretable, and efficient exploration in content generation pipelines.

(3) Set-coded Variation: These variation schemes are typically employed in software engineering tasks like test suite generation, where candidate solutions consist of unordered or partially ordered sets. In Software Product Lines (SPLs), each solution represents a set of test cases that must collectively satisfy coverage and size constraints [11]. The MAP-Elites-based approach uses mutation operators, including addition, deletion, and substitution. The operators are selected adaptively depending on the current size of the test suite: when near the lower bound, addition and substitution dominate; when near the upper bound, deletion and substitution are preferred. This ensures balanced diversity across all feasible test suite sizes. These operators allow fine-grained control over structural diversity and facilitate the exchange of information between adjacent solutions. Furthermore, prioritizing test reuse and suite diversity enhances robustness and generalization across multiple product configurations.

(4) Integer-coded Variation: Integer-coded variation dominates in discrete optimization domains such as scheduling, routing, and hybrid combinatorial problems. In these settings, solutions are typically represented as sequences, sets, or integer vectors, requiring specialized variation operators that respect problem-specific constraints. In workforce scheduling and routing problems, operators manipulate job sequences through gene swaps, segment moves, or random insertions [124,143]. Crossover combines parent segments while preserving feasibility, and mutation perturbs segments to enable new routing patterns [51]. In decision tree optimization, Ferigo et al. [52] applied Grammatical Evolution to mutate tree structures by editing production rules, which enables both syntactic correctness and feature diversity. In hybrid optimization problems like the Traveling Thief Problem (TTP), Nikfarjam et al. [144] proposed a dual-layered strategy combining Edge Assembly Crossover for TSP and dynamic programming for Knapsack components. Further, Marrero et al. [53] demonstrated that low-dimensional projection-based variation can evolve structurally diverse Knapsack instances. These discrete mutation and crossover strategies are tailored to the structural and combinatorial complexity of the domain, allowing for effective exploration without violating feasibility constraints.

The examples above represent only a portion of ongoing research into variation strategies in QD optimization. Table 6 briefly categorizes and summarizes them. As QD algorithms continue to be applied across an expanding range of problem domains, the landscape of variation techniques grows correspondingly in complexity and breadth. For further exploration, readers are encouraged to consult https://quality-diversity.github.io/papers for a comprehensive collection of related works.

3.5. Evaluation metrics

To evaluate the quality of the solution set produced by QD (as summarized in Table 7), four primary metrics are commonly used.

(1) QD-Score [17] measures both the quality and diversity of solutions by summing the objective values of all solutions in non-empty cells.

(2) Coverage [58] reflects the diversity of the solution set in feature space by calculating the ratio of filled cells to the total number of cells.

(3) Archive Fitness indicates the performance value of the best-performing solution in the archive.

(4) Complementary Cumulative Distribution Function (CCDF) [61] reflects the distribution of solution quality across the archive by showing the proportion of solutions whose performance exceeds a given threshold.

(5) Corrected Metrics [35,147] are designed to account for the effects of uncertainty. Each solution preserved by the algorithm is re-evaluated a fixed number of times N, and the average fitness and behavior descriptors of these N re-evaluations are used as an approximation of the "ground-truth" values.

To concretely illustrate how these metrics are computed and interpreted, consider a simple example where the feature space is discretized into a 5 × 5 grid, yielding 25 cells in total. After running the QD algorithm, 16 cells are filled with solutions. For each filled cell, the performance of the best solution (in a maximization problem) is recorded, yielding the following values: $\{92, 85, 77, 88, 90, 83, 95, 80, 74, 91, 87, 79, 70, 89, 93, 86\}$. Based on this data, the *Coverage* can be calculated as the ratio of non-empty cells to total cells, i.e., $16/25 = 0.64$, reflecting

Table 7
Comparison of common evaluation metrics in QD optimization.

Metric	Description and focus
QD-Score	This metric measures the overall quality of the archive by summing the objective values of all solutions in non-empty cells. It captures both solution *performance* and *diversity*.
Coverage	This metric quantifies the *diversity* of the container by calculating the ratio of filled cells to the total number of cells in the feature space.
Archive Fitness	This metric evaluates the best-performing solution stored in the archive, providing an indication of the highest achievable *performance*.
CCDF	This metric analyzes the distribution of solution quality by showing the proportion of solutions in the container whose performance exceeds a given threshold, thereby reflecting *quality spread and robustness.*
Corrected Metrics	Re-evaluate each solution multiple times to account for uncertainty. The re-evaluated solutions are placed in a "Corrected" archive, following the same archive addition rules as the original one but using the re-evaluated results. The QD-score and Max-fitness of this archive are then computed and referred to as *Corrected QD-score* and *Corrected Max-fitness*.

the feature diversity across the feature space. The *QD-Score*, defined as the sum of performance values across all occupied cells, is 1359, representing the overall quality-diversity. The *Archive Fitness*, which captures the performance of the best solution in the archive, is 95. Finally, to compute the CCDF at a performance threshold of 85, we count the number of solutions whose performance exceeds or equals this threshold (10 out of 16), resulting in a CCDF value of $10/16 = 0.625$. This example highlights how these metrics jointly characterize the coverage of QD archive, quality, and distribution of performance.

In uncertain environments, where solution performance or features may vary across evaluations, traditional metrics like QD-score or coverage can be unreliable if based on single trials. To address this, recent work [32] proposes shifting from fixed batch-size comparisons to a sampling-size paradigm, i.e., comparing algorithms based on the total number of evaluations used per generation. This allows a fair assessment of sample-heavy strategies that leverage high-parallelism platforms.

3.6. Applications of QD

QD algorithms have achieved great success across various application domains, including evolutionary robotics, video games, software testing, molecular/drug design, urban planning, and scheduling & route planning (as shown in Fig. 9). This section organizes the applications based on several representative areas, highlighting important applied cases and developments in each domain.

(1) Evolutionary Robotics. In evolutionary robotics, QD algorithms demonstrated strong capabilities in evolving diverse and high-performing solutions. These included evolving virtual creatures capable of adaptive motion [56], and constructing feature-performance maps that enabled resilient robotics [13]. In the latter, robots adapted to unexpected damage in under two minutes by using a precomputed repertoire of high-performing features, without relying on explicit self-diagnosis or predefined contingency plans. QD was also used to generate diverse feature repertoires for complex tasks such as walking and obstacle avoidance [140]. Additionally, QD was applied to the design of controllers that guided robots through maze-like environments [55]. Recently, QD has also been extended to swarm robotics, where maintaining both morphological and behavioral diversity improves collective adaptability and robustness in dynamic environments [148]. Recently, QD has also been extended to swarm robotics, where maintaining both morphological and behavioral diversity improves collective adaptability and robustness in dynamic environments [148]. Similarly, multi-level QD optimization has been applied to co-evolve body, brain, and behavior of robotics, highlighting the importance of diversity across multiple levels for individual adaptability [149]. In robotic manipulation, grasping remained a challenging task due to issues such as sparse rewards, sparse feature feedback, and misaligned features [104]. Traditional approaches often relied on expensive human demonstrations or constrained operational settings, which limited the generalization and adaptability of grasping policies. Therefore, some training and learning-based methods have been incorporated into QD to improve optimization performance.

In [104], QD was applied to robotic grasping, where it enabled the automatic generation of diverse grasping trajectories across different robot-gripper configurations and object types. To support reproducibility in this application domain, the Evolved Grasping Analysis Dataset (EGAD) [150] provides over 2000 geometrically diverse 3D-printable objects for training and evaluating robotic grasp detection algorithms. Beyond grasping, QD has also been explored in robotic perception tasks such as pose estimation [151], where it helped robots infer multiple plausible object poses from sparse tactile feedback. These applications highlight QD's ability to provide both diversity and reliability in real-world robotic manipulation and perception tasks.

Beyond these applications, a growing body of research has focused on integrating QD with RL. The lineage of QD-RL algorithms can be roughly divided into two main threads.

The first thread begins around 2020, when Colas et al. [152] proposed ME-ES (MAP-Elites with Evolutionary Strategies) and Pierrot et al. introduced QD-PG (later published at GECCO 2022) [145]. These works were among the earliest attempts to combine QD with RL methods. Building on this line, Nilsson et al. proposed PGA-MAP-Elites [35], which integrated policy gradient-based variation into MAP-Elites, especially improving early-stage search. PGA-MAP-Elites later evolved into DCG-MAP-Elites [29], which enhanced policy gradient variation with a feature-conditioned critic and distilled archived knowledge into a single versatile policy, significantly improving QD performance. QD Actor–Critic (QDAC) [129] can be seen as a further evolution of this thread, eliminating the explicit archive and using off-policy actor–critic training with value and successor feature critics to jointly optimize for performance and diversity. The second thread of QD-RL research was built on DQD algorithm, CMA-MEGA. The key idea was first implemented by Tjanaka et al. [111]. It was to approximate gradients in non-differentiable environments using RL techniques. This approach was extended by Batra et al. [153], who incorporated PPO into the CMA-MEGA framework to estimate gradients more effectively in high-dimensional and stochastic tasks. Other efficiency-oriented approaches within this lineage include RefQD [154], which decomposed neural networks into shared representation and decision components to reduce training overhead, and CCQD [155], which further applied cooperative coevolution to optimize these components in separate but interacting subpopulations.

Through extensive evaluation across ten grasping domains, Huber et al. [104] demonstrated that QD methods could overcome the challenges of reward and feature sparsity, outperforming alternative approaches by prioritizing successful solutions. Notably, it also revealed that naive novelty search could be misled in sparse-interaction environments, highlighting the importance of informed selection strategies within QD. This indicated a significant advance in applying QD to high-dimensional robotic tasks beyond locomotion, especially in generating adaptable, generalizable grasping features.

(2) Video Games. In the field of video games, QD algorithms were predominantly used for procedural content generation. Applications included the automatic generation of diverse game scenes and levels [141,142], evolving gameplay elements that foster player creativity, and mixed-initiative design systems that support human-AI co-creation [118]. These use cases leveraged the diversity-seeking nature of QD to enhance content richness while maintaining functional constraints such as playability. Recent advances further demonstrated

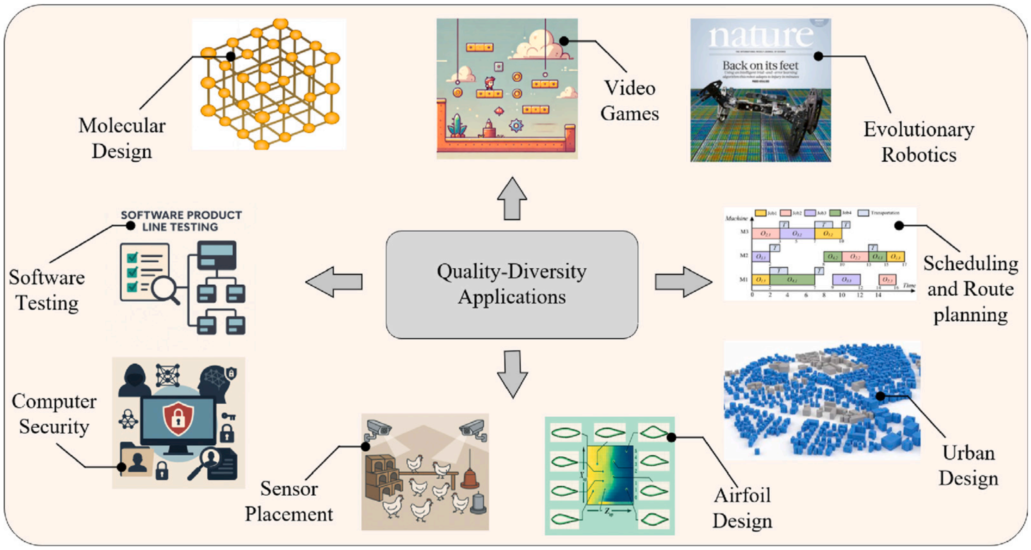


Fig. 9. Representative applications of QD optimization. Some of the above figures are adapted from: evolutionary robots [13], scheduling and route planning [74], urban design [49], airfoil design [60], and molecular design [125].

the versatility of QD in this domain. In Hearthstone deckbuilding, a Deep-Surrogate-Assisted MAP-Elites framework was proposed to improve sample efficiency by combining online-learned outcome prediction with diversity-driven exploration, enabling faster discovery of high-performing and strategically varied decks [42]. In level generation, CMA-ME was applied to evolve Neural Cellular Automata (NCA), producing generators capable of creating diverse, solvable levels across maze, Sokoban, and Zelda domains [156]. Other studies explored tree-structured representations for dungeon layout design, integrating QD to populate maps with locked-door missions and enemies in a structured and player-approved manner [41]. Recently, ML has been used to imitate the exploratory behavior of QD in game content generation. The ML of QD (MLQD) framework [157] integrated QD evolution with Transformer-based modeling to generate diverse and high-quality game content, reproducing the search characteristics while reducing computational costs. These efforts collectively highlighted the strength of QD in fostering both structural variety and functional integrity in procedural game content generation.

(3) Software Testing. In software testing, particularly for SPLs, QD introduced a novel optimization paradigm that simultaneously considered test suite quality (e.g., t-wise coverage) and feature attributes (e.g., test suite size) within a unified framework [11]. Unlike traditional single- or multi-objective optimization approaches, QD facilitated the generation of a large set of high-quality, behaviorally diverse test suites, thereby supporting more flexible and user-preference-driven selection. MAP-Elites was employed to solve this problem by discretizing the feature space into a grid and optimizing local subproblems in each cell, ensuring broad feature coverage and strong performance across the archive. Additionally, MAP-Elites has been shown to produce more fault-revealing test suites than a recent novelty-search-based method, with diversity in test suites positively correlated with fault detection rates. This study not only validated the practicality of QD optimization for SPL testing but also opened promising avenues for future research. In addition, SETBVE [122] extended QD to boundary value exploration (BVE), aiming to uncover diverse failure-inducing features in black-box software systems. By maintaining an archive structured by features and applying local refinement, it consistently outperformed traditional methods in both quality and feature diversity. This further demonstrated the potential of QD in exposing edge-case software features beyond standard optimization techniques. Recent advances in software robustness testing have recast RL policy evaluation as a QD optimization problem, extending the applicability of QD to fault discovery in learned decision models [123]. Rather than maximizing the quantity of failures (e.g., similar collisions), the goal of this work is to uncover semantically diverse fault cases that better characterize the vulnerabilities of the learned policies. The authors tackled challenges specific to action-policy spaces and implemented two illumination-based QD algorithms to search for failure-inducing test cases. The experiments demonstrated that QD-based policy testing uncovers richer feature failure modes than prior methods, all without increasing testing budget—highlighting the unique strength of QD in capturing structural diversity in fault discovery.

(4) Scheduling and Route Planning. QD algorithms have shown growing potential in solving discrete combinatorial problems such as scheduling and routing. In workforce scheduling and routing problems, MAP-Elites outperformed evolutionary algorithms in providing diverse, high-quality schedules when computational budgets were sufficient [50]. Urquhart et al. [51] further proposed a micro-depot vehicle routing model and demonstrated how QD improved emissions, cost, and courier utilization. Decision tree learning models had been hybridized with RL and diversified using QD, showing promising results in dynamic control tasks [52]. Marrero et al. [53] employed QD algorithms to evolve diverse and discriminatory instances of Knapsack Problems in a low-dimensional projection, demonstrating complementary coverage when combined with traditional feature-space search. In the field of job shop scheduling, MAP-Elites combined with RL effectively addressed FJSP with transportation constraints, outperforming traditional metaheuristic methods [12].

(5) Molecular Design and Urban Planning. QD optimization was recently adopted in molecular design tasks to generate structurally diverse and high-performing solutions in complex search spaces. Unlike traditional approaches that focused on a single global optimum, which was typically the most stable molecule or crystal structure, QD aimed to discover a diverse set of candidates across multiple trade-offs and feature dimensions. For example, Janmohamed et al. [125] applied Multi-Objective QD to crystal structure prediction, moving beyond stability-only objectives by incorporating other material properties like magnetism and thermoelectric efficiency. Their method successfully rediscovered known crystal configurations and identified new ones with valuable trade-offs, while also proposing a visualization scheme to illuminate the multi-objective landscape. To further enhance search efficiency and diversity in small molecule generation, Verhellen et al. [46] proposed a Bayesian Illumination framework. This method integrates bespoke molecular kernels into a QD framework via Bayesian optimization, enabling more effective exploration of the chemical space. Bayesian Illumination outperformed both traditional QD and deep generative models, yielding a broader spectrum of high-quality molecules. In discrete molecular design tasks, such as protein generation or discrete latent space illumination, gradient-based guidance is often infeasible due to the combinatorial nature of the domain. To address this, Boige et al. [47] proposed a Gradient-Informed Discrete Emitter (ME-GIDE) that adapts gradient information to guide QD search over discrete spaces. ME-GIDE demonstrated superior performance across challenging benchmarks by leveraging differentiable surrogate functions for both objective and feature evaluations. These developments collectively highlight the expanding role of QD in molecular and material discovery, where the ability to generate diverse, interpretable, and high-quality solutions provides a significant advantage over conventional optimization techniques.

In addition to the applications mentioned above, QD has also been applied to design domains. For instance, Gaier et al. [158] proposed a surrogate-assisted neuroevolution approach for design tasks involving variable-topology neural networks. By leveraging the compatibility distance from NEAT within a kernel-based surrogate model, they significantly improved data efficiency in classic control problems like cart–pole swing-up and half-cheetah. This work highlighted the value of QD in evolving high-performing and structurally diverse neural architectures under limited evaluations. More recently, Gaier et al. [159] applied QD to generative architectural design. Their method first used QD to produce a diverse, high-performing dataset, which was then used to fine-tune a language model for generating high-level layouts. These were subsequently refined into constraint-satisfying architectural plans, demonstrating that QD-synthesized data enhances generative model performance and fidelity to user intent.

(6) Public Libraries for QD Research: While not an application domain in itself, the increasing availability of open-source software libraries has played an important role in expanding the reach and adoption of QD algorithms. To complement our discussion of application areas, we provide several widely used and impactful QD libraries that facilitate experimentation and deployment for both newcomers and experienced researchers.

Among the earliest contributions, Sferes*v*2 [160] was one of the first general evolutionary computation libraries to include support for QD algorithms. Developed in C++ with template meta-programming, it emphasized performance and extensibility, though this design also made it less accessible to non-expert users.

QDpy [62] was one of the first dedicated QD libraries in Python. It provided feature-rich and modular tools, including built-in containers, visualizations, and distributed evaluation capabilities. It supported standard algorithms such as MAP-Elites and CMA-ME through intuitive abstractions, and played an important role in lowering the barrier to entry for Python users.

Building on these foundations, Pyribs [66] was introduced in 2021 as the software implementation of RIBS, and formally published in 2023. Pyribs adhered to the principles of simplicity, flexibility, and accessibility. It provided a minimal set of essential components, supported easy integration with other frameworks, and was beginner-friendly with rich documentation and tutorials. Since its release, Pyribs has been applied across a variety of domains, including image and environment generation, RL, and hyperparameter optimization, demonstrating its broad utility and adaptability in the QD community.

QDax [37], published shortly after in 2022, leveraged JAX [161] for just-in-time compilation and hardware acceleration (e.g., GPU/TPU), making it suitable for large-scale or real-time QD applications. It supported MAP-Elites and PGA-MAP-Elites, offering efficient batched evaluation as well as evolution pipelines, which were particularly useful in high-dimensional continuous control tasks.

Other libraries also contributed significantly to the QD ecosystem. For instance, pymap_elites [162] served as a lightweight reference implementation for several MAP-Elites variants, such as CVT-MAP-Elites [61], Iso+LineDD [138], and Multi-task MAP-Elites [64].

4. Challenges and future directions in QD optimization

QD optimization has become a powerful paradigm in evolutionary computing, aimed at generating a set of high-performing solutions across different domains. However, several challenges persist that hinder its full potential. Addressing these challenges is essential for further advancing QD optimization and expanding its practical applications.

4.1. Challenges

(1) Challenges in Applying QD to Discrete Optimization: Despite the widespread success of QD algorithms in continuous optimization, their application to discrete domains remains relatively underexplored. A central challenge arises from the incompatibility between standard variation operators (e.g., Gaussian mutation or real-valued crossover) and the structural nature of discrete solution representations. In discrete problems, solutions are often represented as integer vectors, permutations, or symbolic structures, where small changes in encoding can lead to non-local and unpredictable changes in features. This disrupts the feature-wise continuity needed for QD algorithms to perform effective gradual exploration. Although specialized variation operators have been proposed for discrete settings, such as gene swaps, segment insertions, or grammar-based tree editing [51–53,124,143,144], their integration into QD remains limited in scope. In FJSP, beyond these structural and representational issues, another critical challenge is how to leverage QD to explore and optimize discrete solutions in a way that maximizes solution quality without compromising feature diversity. Maintaining this balance is still challenging, since pursuing optimization too aggressively tends to reduce diversity, while excessive exploration can lead to suboptimal overall performance. Furthermore, integrating QD with domain-specific search heuristics and machine learning–based strategies remains a challenging [74]. While such hybrid frameworks have the potential to enhance exploration efficiency and adaptability, their practical realization is still limited by the difficulty of balancing heuristic guidance with the preservation of generative diversity—a defining property of QD. Moreover, when applying QD to multi-objective FJSP [163] with complex constraints, effectively exploiting diverse solutions across different features to enable directional search and informed archive updates poses an important yet unresolved challenge.

Furthermore, discrete optimization problems often involve vast and rugged search spaces that lack gradient information and involve complex feasibility constraints. Although such constraints are also present in continuous domains, they are particularly problematic in discrete contexts due to the combinatorial explosion of valid configurations and the difficulty of designing constraint-preserving variation operators. This complicates both the exploration of novel features and the maintenance of a feasible and diverse archive. Another difficulty lies in the design and utilization of features. Many QD approaches organize solution diversity using containers such as MAP-Elites grids or centroid-based decompositions. However, in discrete problems, the combinatorial explosion of possible features often results in sparse and unevenly populated feature spaces, making these traditional structures inefficient or even ineffective. In addition, small changes in discrete structures may not induce significant variation in feature space, weakening the core mechanism of feature-driven search. Lastly, while unsupervised or learned feature construction methods (e.g., autoencoders) have demonstrated promise in continuous QD [24], they often struggle to capture semantic similarity in symbolic or non-differentiable spaces typical of discrete problems. This limits the adaptability of QD in domains where manual feature engineering is infeasible or costly.

(2) Scalability and Computational Complexity: As the dimensionality of the search and feature space increases, the computational resources required for QD algorithms grow exponentially [80]. This issue of dimensionality limits the practical application of QD in complex, real-world problems, as higher-dimensional spaces require significantly more evaluations to maintain diversity and quality [17]. In robotic applications, evolving a repertoire of features for high-degree-of-freedom robots becomes computationally intensive. To manage this complexity, simplifications are often introduced, but they can come at the cost of solution quality [13]. For instance, optimizing locomotion strategies in legged robots requires thousands of evaluations in simulation, making direct deployment infeasible due to time limitations and complex constraints. Moreover, the management of large archives in QD poses memory and processing challenges as high-dimensional solutions require increased storage and computational overhead [81]. Traditional QD methods rely on exhaustive or random exploration of the search space, which becomes impractical as dimensionality increases.

(3) Uncertainty and Robustness: QD algorithms often assume deterministic environments, but real-world applications introduce uncertainties due to noise, emergencies, hardware failure, environmental variability, and model inaccuracies. These factors can significantly impact the reliability of evolved solutions, making robustness a big challenge [32,82]. For instance, robotic features optimized in simulation often fail in real-world deployment due to unmodeled dynamics, leading to the well-known sim-to-real gap [83]. Addressing this requires QD algorithms capable of handling stochastic evaluations and adapting to uncertain conditions. One promising approach to enhancing robustness is uncertain QD, which models both fitness and features as probability distributions rather than fixed values [32]. This formulation better captures variability in noisy or stochastic environments and enables more informed decision-making. However, it also increases computational cost due to the need for repeated sampling and can complicate archive management, especially in high-dimensional or resource-constrained settings. Additionally, in FJSP, QD optimization algorithms have been employed to enhance scheduling efficiency [74]. However, real-world manufacturing environments often present uncertainties such as machine breakdowns [164], fuzzy processing times [165], and emergent job arrivals [78]. Addressing these uncertainties within the framework of QD optimization poses significant challenges.

(4) The Maintenance of Diversity and Quality: Maintaining diversity and quality simultaneously remains a big challenge in QD optimization. While promoting diversity encourages broad exploration of the feature space, it often leads to the retention of low-performing solutions that add little practical value. Conversely, focusing primarily on quality can reduce feature diversity and cause premature convergence to suboptimal regions [72]. Most QD algorithms, such as MAP-Elites, often operate with fixed selection pressure and lack adaptivity over time, limiting their ability to adjust the trade-off between exploration and exploitation as the search progresses. Furthermore, the effectiveness of diversity preservation depends heavily on the feature space—if it fails to capture meaningful distinctions, the search may become biased or redundant [84]. In sparse or deceptive environments, excessive novelty pressure can further undermine search efficiency by favoring uniqueness over task relevance, leading to suboptimal solution archives [55]. Although recent efforts have proposed adaptive selection mechanisms and learned diversity metrics to address these problems, most approaches remain heuristic and lack generalizability across domains, highlighting the need for more principled and robust strategies in managing diversity and quality.

(5) Feature Design: One of the fundamental challenges in QD optimization is the design of features. The choice of features significantly impacts the effectiveness of QD algorithms, as they define the diversity dimensions that guide the search process. A poorly designed feature may fail to capture meaningful distinctions between solutions, leading to redundant exploration or missing valuable regions of the search space [16]. In high-dimensional problems, manually specifying features may become infeasible due to the complexity and variability of the solution space [166]. For instance, in robotics, defining features for locomotion strategies requires careful consideration of movement patterns, stability, and energy efficiency. A simplistic feature based on step length or velocity might fail to differentiate functionally diverse features, whereas a high-dimensional feature may introduce unnecessary complexity and computational overhead [167]. Additionally, the optimal features often vary across tasks and environments, making it difficult to design a universally applicable representation. To address the above challenges, recent work has explored automatic feature design using unsupervised learning techniques. These methods aim to extract compact and informative features directly from raw sensor data or state trajectories, thereby reducing the need for manual approaches. However, despite their promise, such automatically learned features may not always align with the task-relevant aspects of features [168]. In some cases, the learned representations may emphasize superficial variations or noise, while ignoring subtle but functionally critical distinctions. Moreover, the lack of interpretability in these latent features makes it difficult to understand or debug the search dynamics. As a result, ensuring that automatically derived features are both meaningful and aligned with the optimization goals remains an open problem in QD research.

(6) Integration with Machine Learning: A major challenge is the difficulty of integrating QD optimization with modern machine learning techniques such as DRL. QD and DRL have fundamentally different optimization dynamics and goals: QD emphasizes exploring a broad variety of solutions to maintain diversity, whereas DRL mainly focuses on maximizing specific performance metrics for a single optimal policy. This divergence can lead to conflicts during training, where aggressively pursuing feature diversity may impede the improvement of reward-based performance, and vice versa [169]. While some studies have demonstrated that incorporating diversity objectives into RL can help overcome issues like deceptive or sparse rewards, achieving a seamless integration of both objectives remains an open challenge [85]. In practice, maintaining the twin aims of reward and diversity requires careful coordination; if not managed properly, one objective can dominate, resulting in suboptimal policies that either excel in diversity at the cost of performance or achieve high reward but collapse into a narrow set of features [169].

Several hybrid approaches were proposed to integrate QD optimization with DRL, aiming to combine the strengths of both paradigms. For example, PGA-MAP-Elites [65] combined genetic mutations with policy gradient updates, while CMA-ME [63] integrated CMA-ES with MAP-Elites using multiple emitters. Despite these advances, effectively integrating QD with DRL remained challenging. A key limitation was the conflict between reward maximization and diversity preservation: aggressively pursuing one often compromised the other, leading to premature convergence or stagnation [73]. Among existing attempts, QDAC [73] introduced Lagrangian constraints to explicitly balance reward and diversity. However, this formulation was unique to that framework and differs from conventional QD methods: rather than maintaining a population of solutions, QDAC embeds diverse behaviors within a single constrained policy. In fact, this distinctive design even prompt discussions about the extent to which QDAC should be considered a QD approach, given its departure from the population-based paradigm that underpins most QD methods. Such differences highlight the difficulty of transferring DRL mechanisms directly into QD settings. In many cases, RL components become under-trained or inefficient due to interference from diversity-seeking operations. These challenges underscore the need for adaptive, scalable strategies that can robustly coordinate exploration and exploitation in high-dimensional environments.

Another major limitation of combining QD with ML is the increased computational burden and reduced sample efficiency. QD algorithms maintain and evaluate a large archive of diverse solutions, which is inherently resource-intensive. In complex domains, QD may spawn hundreds or thousands of candidate policies to cover the feature space [170]. This brute-force exploration becomes especially costly when merged with gradient-based optimization, which requires additional updates per individual. In high-dimensional settings, the number of evaluations and gradient steps can grow prohibitively, making QD–ML hybrids significantly less efficient than standalone DRL or evolutionary methods. As a result, practical deployment of such systems remains constrained by computational costs and training time, highlighting the need for more scalable and sample-efficient integration strategies.

(7) Multi-objective QD optimization: Multi-objective optimization in QD algorithms is particularly challenging due to the need to simultaneously preserve feature diversity and balance trade-offs among conflicting objectives [67]. While traditional QD frameworks like MAP-Elites rely on a single quality metric, many real-world tasks — such as robotics, neural architecture search, and automated design — require optimizing multiple objectives, such as performance, energy efficiency, cost, or safety [171,172]. To address this, MOME extends the QD paradigm by maintaining Pareto fronts within each niche of the feature space, offering a principled way to explore both functional diversity and Pareto-optimal trade-offs [67]. However, this approach introduces a number of challenges that limit its scalability and effectiveness in practice.

First, the archive size increases rapidly, as each cell may contain multiple non-dominated solutions. This leads to significant computational overhead in maintaining local Pareto fronts and updating dominance relations—particularly problematic in high-resolution grids or continuous feature spaces. Second, selection mechanisms in MOME often struggle to promote sufficient pressure toward unexplored niches while preserving objective-level trade-offs, especially when one objective dominates the others numerically, leading to biased exploration. Third, conflicting objectives often cause solutions to cluster near extremes of the Pareto front, resulting in sparse or unevenly populated archives, which reduces the overall diversity that QD aims to achieve. Lastly, due to the inherent complexity of managing both behavioral and objective diversity, many MOME implementations are computationally expensive and do not scale well to large or real-time environments.

4.2. Future directions

(1) Enhancing QD for combinatorial Optimization Problems: Theoretical insights have laid the groundwork for understanding QD optimization, but much work remains to be done for multi-constraint, large-scale, high-dimensional, and multi-objective combinatorial optimization problems. Future research can aim to (1) extend theoretical analysis to high-dimensional environments; (2) validate the feature design for different types of QD variants; and (3) investigate theoretical models for multi-objective QD, particularly under uncertainty or dynamic environments. Future research can focus on developing specialized variation operators tailored for combinatorial optimization problems. Rather than relying solely on generic mutation and crossover techniques, incorporating problem-specific knowledge through adaptive or collaborative search operators, such as heuristic search [173,174], variable neighborhood search [175–177], and local intensification [178–181], can significantly enhance search efficiency. Another promising direction is to explore surrogate models and reinforcement learning techniques [182,183] to guide QD exploration in combinatorial settings, ensuring that generated solutions remain feasible while maintaining diversity. Additionally, efforts should be made to define features for combinatorial optimization problems. Leveraging problem-specific characteristics could help capture meaningful feature diversity. Finally, integrating QD with other combinatorial optimization paradigms — such as genetic programming [184,185], differential evolution (DE) [186], or micro search [187] — could enhance its applicability to real-world discrete optimization tasks, making QD a more versatile approach beyond continuous domains.

(2) Tackling Scalability and Computational Complexity: To tackle the scalability challenges of QD algorithms, prior research focused on developing more efficient data structures and search strategies. One promising approach was to employ adaptive dimensionality reduction techniques, where high-dimensional search and feature spaces were projected into a lower-dimensional latent representation. Techniques such as VAE-like models or contrastive learning were leveraged to learn compact and informative features, reducing the computational burden of QD optimization while preserving diversity [188]. In the context of QD-RL, Policy Manifold Search [189] proposed searching for diverse policies within a low-dimensional latent manifold learned from policy parameters. By using an autoencoder-like structure and guiding exploration with the Jacobian of the inverse mapping, it ensured that generated policies remained valid and high-performing. This approach improved scalability in high-dimensional spaces while preserving the core diversity of QD. Additionally, mixed-precision strategies improved efficiency in high-dimensional domains. They not only sped up the computation but also ensured robust convergence, and were beneficial for a variety of applications [190]. Another avenue for improvement involved the application of parallel and distributed computing to accelerate QD evaluations [70]. Furthermore, to address scenarios where objective and feature evaluations were computationally expensive, several approaches employed surrogate models to approximate these functions. By replacing direct evaluations with predictions from a learned model, these methods significantly reduced computational overhead. These strategies enhanced sample efficiency by decreasing the number of costly evaluations required to effectively solve a QD problem. For example, BOP-Elites [191] leveraged Gaussian processes as surrogate models to efficiently approximate the objective landscape, thereby informing the selection of the most promising candidates for evaluation. Surrogate-Assisted Illumination (SAIL) [158] was an efficient QD algorithm that used surrogate models to generate diverse and high-performing solutions while minimizing the number of fitness evaluations. To enable the discovery of diverse features based on costly phenotypic features, Hagg et al. [192] proposed surrogate-assisted phenotypic niching QD that combined GPU-based simulation with surrogate modeling. This method was applied to 2D fluid dynamics, where it reduced the number of required simulations. Other approaches that leveraged surrogate models included using deep models to assist MAP-Elites in automated Hearthstone deckbuilding [42]. By training a surrogate model online to predict game outcomes, the method reduced expensive evaluations while guiding MAP-Elites toward high-quality, strategically diverse decks.

(3) Tackling Uncertainty and Robustness: Considering uncertainty in QD optimization opens promising directions for future research, particularly as QD algorithms are increasingly deployed in stochastic real-world environments. In such settings, individuals may yield variable outcomes due to noise or partial observability. Recent advances suggest that treating features as probabilistic distributions — rather than fixed values — can significantly enhance the reliability of QD archives under uncertainty. For instance, Deep-Grid MAP-Elites [82] introduced a depth dimension to record multiple evaluations per cell, enabling better estimates of expected performance and feature diversity. Building on this idea, parallelized variants [32] leverage modern hardware to scale up evaluations and improve the statistical robustness of search dynamics. Future work could further investigate scalable sampling methods and memory-efficient mechanisms for integrating distributional estimates into QD pipelines. These advances would be crucial for enabling real-time deployment in noisy or dynamic environments. In addition, some efforts have introduced modular frameworks that unify uncertain QD approaches and facilitate the design of more reliable methods across stochastic domains [193]. More recently, balancing performance and reproducibility has been identified as a promising direction to enhance robustness in uncertain QD [194].

As QD algorithms are increasingly deployed in complex and stochastic real-world environments, addressing uncertainty remains a promising direction for future research. Rather than assuming deterministic evaluations and stable features, future QD frameworks could integrate probabilistic modeling to capture variability in fitness and feature measurements [32,82]. This may involve treating both objective and feature values as distributions rather than fixed points, enabling more robust archive updates under noisy conditions. In robotics, bridging the sim-to-real gap — where discrepancies between simulation and real-world dynamics degrade policy performance — calls for QD algorithms that can generalize across domains, potentially by incorporating domain adaptation, meta-learning, or uncertainty-aware simulation techniques [83]. In manufacturing applications like the FJSP, dynamic events such as machine breakdowns or emergency job arrivals suggest the need for online or reactive QD methods that can adjust to changing constraints in real time [74,78,164,165].

Additionally, a promising research avenue lies in learning features that are robust to noise, either through smoothing, probabilistic embedding, or latent representation learning. Coupling this with sample-efficient uncertainty estimation — e.g., via Bayesian optimization, active sampling, or ensemble modeling — may significantly enhance QD performance in high-stakes domains where repeated evaluations are costly. Ultimately, exploring the intersection of robust optimization, uncertainty quantification, and QD opens exciting possibilities. Future work could not only discover diverse solutions but also adaptively maintain reliability in unpredictable environments.

In addition, future research should move beyond passive uncertainty modeling and explore proactive mechanisms for uncertainty mitigation. Several works have already introduced adaptive sampling strategies for QD. For example, Justesen et al. [147] extended MAP-Elites with adaptive sampling and drifting-elites to improve robustness in noisy domains, while more recent work on Uncertain QD [32] proposed archive-sampling, parallel-adaptive-sampling, and deep-grid-sampling to better handle stochastic evaluations. Moreover, integrating adaptive sampling strategies such as active learning [195] could prioritize evaluations based on uncertainty estimates or informativeness, thus reducing redundant computations. There is also a need to develop uncertainty-aware selection and variation operators that maintain archive stability and feasibility under real-time constraints, especially in discrete and dynamic domains like manufacturing. Additionally, more data-efficient methods — such as transfer learning or Bayesian meta-learning — could improve generalization across environments and tasks with minimal evaluations. In sim-to-real scenarios, research may explore the coupling of QD with domain randomization and fine-tuning mechanisms to reduce the reality gap [83]. Finally, the design of probabilistic archives that not only store performance and novelty but also encode confidence intervals or uncertainty bounds remains an open and impactful direction for enabling robust decision-making under uncertainty.

(4) Tackling Maintenance of Diversity and Quality: To effectively maintain diversity and quality in QD optimization, future research can explore adaptive diversity regulation mechanisms. These mechanisms dynamically adjust selection pressure based on the evolving state of the population, preventing excessive novelty pressure while still maintaining feature and structural diversity. For example, multi-stage QD algorithms can be designed to transition between exploration and exploitation phases. Early search phases may emphasize the discovery of novel solutions across the behavior space, while later stages shift toward refining high-performing individuals, thereby reducing premature convergence and improving solution quality [84]. Another promising direction involves designing variation operators with strong local exploitation capability, especially in structured or high-dimensional domains. Such operators can deeply explore neighborhoods of elite solutions to improve convergence speed without compromising diversity [74]. For instance, future work could extend DE-based strategies by dynamically adapting the use of difference vectors depending on the diversity level of the current population [1]. When diversity is high, using multiple difference vectors may promote exploration, while in low-diversity regimes, using best-based base vectors can intensify local search.

In addition, learning-based or self-adaptive variation strategies could be explored. These approaches adjust mutation strengths, crossover rates, or recombination templates in response to feedback from the population (e.g., changes in archive density or feature entropy). RL or Bayesian optimization might also be employed to optimize variation hyperparameters online, guiding the search adaptively in response to current search progress. Furthermore, Templier et al. [27] proposed JEDi (Quality with Just Enough Diversity), which learns the relationship between behavior descriptors and fitness to focus evaluations on promising regions while maintaining sufficient diversity to avoid deception. This framework offers valuable guidance for designing future QD algorithms that aim to achieve higher efficiency in complex and deceptive domains. Finally, future work may consider integrating diversity metrics directly into parent or emitter selection mechanisms, enabling the algorithm to favor lineages that contribute most to underrepresented regions of the feature space. Such diversity-aware selection can help maintain coverage even in deceptive or redundant environments.

(5) Tackling Feature Design: Automatic feature learning can help define features in QD algorithms. Dimensionality reduction techniques like auto-encoders and PCA extract meaningful features from datasets or generated solutions, reducing manual effort [17]. Adaptive methods iteratively refine features based on newly discovered solutions. Techniques like AURORA [114], TAXONS [115], and RUDA [116] represent advances in unsupervised QD, enabling algorithms to autonomously discover features tailored to the task. RUDA, in particular, guides exploration toward task-relevant regions of the learned feature space by incorporating a relevance-driven diversity metric. Self-supervised learning, including contrastive learning and autoencoder-based representations, can further enhance feature quality by automatically identifying informative features [166]. Task-adaptive feature learning enables QD algorithms to adjust feature representations based on specific optimization goals, thereby improving relevance and effectiveness in different problem settings. To achieve this adaptability, meta-learning approaches [196] can be employed to generalize feature selection strategies across tasks, while reinforcement learning [74] can dynamically refine features during the optimization process based on feedback signals. Recently, there has also been work that tries to use human feedback to define the features, called Quality Diversity through Human Feedback (QDHF) [117]. Instead of relying on predefined diversity metrics, QDHF infers them from human similarity judgments, enabling QD algorithms to operate effectively in complex, open-ended tasks.

(6) Tackling Integration with Machine Learning: The integration of QD with machine learning can be enhanced through hybrid learning frameworks that leverage both evolutionary and gradient-based optimization techniques. One representative direction is the integration of QD principles into reinforcement learning algorithms [168,197]. For example, QD-AC [73] samples skill variables and leverages two critics — a value function and a successor feature critic — to jointly optimize for both high return and feature diversity via constrained policy updates. This unified approach allows agents to learn diverse and high-performing policies during training without the need for post hoc fine-tuning. It is also a promising research direction to introduce RL or DRL [6,198] to dynamically adjust the strategy of using variation operators in QD optimization. By training an intelligent agent to learn and select the most suitable variation operator at different stages of the search process and in varying environments, the local search abilities of QD algorithms can be effectively enhanced. For instance, the agent can adaptively choose between local search-oriented or global search-oriented mutation operators based on the current population diversity and fitness distribution, thereby achieving a better balance between exploration and exploitation. This approach not only enhances the flexibility of the algorithm but also improves optimization efficiency, facilitating the discovery of higher-quality and more diverse solutions within complex search spaces. Techniques such as population-based training and evolutionary hyperparameter optimization can be integrated with QD to improve its adaptability in complex learning environments [85]. Additionally, the application of transfer learning in QD can significantly reduce computational costs and improve generalization [71,199]. Ultimately, maintaining methodological diversity remains crucial for scientific progress—although QD may not yet be mainstream, its principles of fostering both excellence and variation could form an essential foundation for future breakthroughs in ML [200].

(7) Tackling Multi-objective QD Optimization: While several approaches have extended QD to handle multiple objectives [67,171,172], developing flexible, scalable, and generalizable multi-objective QD frameworks remains a promising direction for future research. Unlike traditional QD, which typically optimizes a single performance criterion alongside feature diversity, multi-objective QD must simultaneously balance trade-offs across multiple, often conflicting, objectives (e.g., speed vs. energy efficiency, accuracy vs. interpretability), while still maintaining a diverse archive of solutions. This introduces fundamental challenges in terms of archive management, selection pressure, and feature space representation under multi-objective settings. One open question is how to effectively maintain both Pareto-optimality and feature diversity without bloating the archive or sacrificing computational efficiency. Traditional Pareto-based approaches may suffer from loss of diversity, while grid-based QD methods may struggle to represent complex trade-off surfaces. Therefore, future research could explore hybrid archive structures that jointly organize solutions by feature similarity and Pareto dominance. Another way forward is suggested on unstructured and unbounded repertoire formulations for MOQD [201], which provide a promising starting point for extending such hybrid approaches to adaptive and learned feature spaces in complex domains. Another promising direction lies in adaptive objective weighting and preference modeling. In many real-world applications, the importance of objectives may shift over time or depend on the deployment context. RL, meta-optimization, or user-in-the-loop feedback mechanisms could be employed to dynamically adjust optimization focus during the search, enabling better responsiveness to task-specific constraints and user-defined priorities.

Moreover, visualization and interpretability in multi-objective QD remain underexplored. As archives grow more complex, there is a growing need for methods to analyze, summarize, and interact with high-dimensional Pareto-diverse archives. Leveraging techniques from dimensionality reduction, explainable AI, or interactive visualization could support more effective deployment of QD in human-centric design and decision-making processes. Finally, future work could investigate domain-informed feature construction that reflects trade-offs more explicitly—e.g., decomposing features into sub-skills aligned with different objectives, or using learned embeddings to structure the archive in a task-aware manner. These lines of research could greatly enhance the applicability of QD in multi-criteria optimization scenarios, such as autonomous systems, multi-objective controller synthesis, and co-optimization in hardware/software co-design.

5. Conclusion

This paper provides a comprehensive survey of Quality-Diversity (QD) optimization, a paradigm that extends traditional optimization by balancing performance and diversity. We explore recent advances in QD and examine its applications in evolutionary robotics, video games, neural networks, software testing, scheduling & route planning and engineering design. By reviewing state-of-the-art methods, we identify the key components of QD algorithms, including archive structures, selection mechanisms, variation operators, and features. These elements play a crucial role in shaping the effectiveness of QD across different domains.

QD optimization has achieved significant success in continuous optimization, where solution representations and variation operators align naturally with evolutionary computation techniques. However, its application in discrete optimization remains relatively underexplored. While recent studies have introduced QD into software testing, job shop scheduling, and workforce routing, fundamental challenges persist. Discrete problems lack gradient information, making efficient exploration strategies difficult to design. Additionally, defining feature spaces in discrete search domains is complex. It often requires problem-specific knowledge or learning-based approaches. Addressing these issues is crucial for advancing QD in discrete optimization. Beyond these established applications, future research could explore the use of QD in emerging domains. One promising area is model rocket design [202], where structural and aerodynamic trade-offs yield rich design spaces. Another is optical system design [203], where diverse high-quality lens configurations can improve robustness and tolerance to manufacturing variations. QD may also advance financial portfolio optimization [204], which naturally aligns with its principles by balancing performance and diversification across multiple assets.

We hope this survey serves as a valuable resource for researchers and practitioners interested in QD optimization. By highlighting its strengths and limitations, we hope to inspire further studies and innovative applications in both continuous and discrete problems. Addressing current challenges and expanding QD will pave the way for novel solutions in complex optimization domains.

CRediT authorship contribution statement

Haoxiang Qin: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Yi Xiang:** Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Hainan Zhang:** Writing – review & editing, Supervision. **Yuyan Han:** Writing – review & editing, Supervision, Formal analysis. **Yuting Wang:** Writing – review & editing, Supervision. **Xinrui Tao:** Writing – review & editing, Supervision. **Yiping Liu:** Writing – review & editing, Supervision.

Declaration of competing interest

We declare that we have no personal relationships with other people or organizations that can inappropriately influence our work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This work is supported by Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515030022); Natural Science Foundation of Guangdong Province, China (No. 2022A1515110058); the Fundamental Research Funds for the Central Universities, China (No. 2024ZYGXZR097, 93K172024K03); National Natural Science Foundation of China (No. 61906069, 62566012); the Natural Science Research Project of Education Department of Guizhou Province (No. QJJ2023061).

Data availability

No data was used for the research described in the article.

References

[1] H. Qin, W. Bai, Y. Xiang, F. Liu, Y. Han, L. Wang, A self-adaptive collaborative differential evolution algorithm for solving energy resource management problems in smart grids, IEEE Trans. Evol. Comput. 28 (5) (2024) 1427–1441, http://dx.doi.org/10.1109/TEVC.2023.3312769.

[2] O. Sigmund, A 99-line topology optimization code written in Matlab, Struct. Multidiscip. Optim. 21 (2001) 120–127.

[3] Z. Lin, K. Gao, N. Wu, P. Nagaratnam Suganthan, Problem-specific knowledge based multi-objective meta-heuristics combined Q-learning for scheduling urban traffic lights with carbon emissions, IEEE Trans. Intell. Transp. Syst. 25 (10) (2024) 15053–15064, http://dx.doi.org/10.1109/TITS.2024.3397077.

[4] X. Tao, Q. Pan, L. Gao, An iterated greedy algorithm with reinforcement learning for distributed hybrid FlowShop problems with job merging, IEEE Trans. Evol. Comput. (2024) 1, http://dx.doi.org/10.1109/TEVC.2024.3443874.

[5] Y. Wang, Y. Han, Y. Wang, Q.-K. Pan, L. Wang, Sustainable scheduling of distributed flow shop group: A collaborative multi-objective evolutionary algorithm driven by indicators, IEEE Trans. Evol. Comput. 28 (6) (2024) 1794–1808, http://dx.doi.org/10.1109/TEVC.2023.3339558.
[6] H. Qin, Y. Han, Q. Chen, L. Wang, Y. Wang, J. Li, Y. Liu, Energy-efficient iterative greedy algorithm for the distributed hybrid flow shop scheduling with blocking constraints, IEEE Trans. Emerg. Top. Comput. Intell. 7 (5) (2023) 1442–1457, http://dx.doi.org/10.1109/TETCI.2023.3271331.
[7] F. D'Errico, Multi-Objective Optimization in Engineering Design, Springer International Publishing, 2015.
[8] A. Cully, Y. Demiris, Quality and diversity optimization: A unifying modular framework, IEEE Trans. Evol. Comput. 22 (2) (2018) 245–259, http://dx.doi.org/10.1109/TEVC.2017.2704781.
[9] A. Antoniou, W.-S. Lu, Practical Optimization: Algorithms and Engineering Applications, Springer Science & Business Media, 2007.
[10] Y. Xiang, H. Huang, M. Li, S. Li, X. Yang, Looking for novelty in search-based software product line testing, IEEE Trans. Softw. Eng. 48 (7) (2022) 2317–2338, http://dx.doi.org/10.1109/TSE.2021.3057853.
[11] Y. Xiang, H. Huang, S. Li, M. Li, C. Luo, X. Yang, Automated test suite generation for software product lines based on quality-diversity optimization, ACM Trans. Softw. Eng. Methodol. 33 (2) (2023) http://dx.doi.org/10.1145/3628158.
[12] H. Qin, Y. Xiang, Y. Han, X. Yan, Optimizing energy-efficient flexible job shop scheduling with transportation constraints: A Q-learning enhanced quality-diversity algorithm, in: 2024 6th International Conference on Data-Driven Optimization of Complex Systems, DOCS, 2024, pp. 373–378, http://dx.doi.org/10.1109/DOCS63458.2024.10704469.
[13] A. Cully, J. Clune, D. Tarapore, J.-B. Mouret, Robots that can adapt like animals, Nature 521 (7553) (2015) 503–507, http://dx.doi.org/10.1038/nature14422.
[14] J. Kober, J. Peters, Reinforcement learning in robotics: A survey, in: M. Wiering, M. van Otterlo (Eds.), Reinforcement Learning. Adaptation, Learning, and Optimization, Vol. 12, Springer, Berlin, Heidelberg, 2012, http://dx.doi.org/10.1007/978-3-642-27645-3_18.
[15] W. Nowak, Introduction to stochastic search and optimization. Estimation, simulation, and control (Spall, J.C.; 2003) [book review], IEEE Trans. Neural Netw. 18 (3) (2007) 964–965, http://dx.doi.org/10.1109/TNN.2007.897481.
[16] J.K. Pugh, L.B. Soros, K.O. Stanley, Quality diversity: A new frontier for evolutionary computation, Front. Robotics Ai 3 (2016).
[17] K. Chatzilygeroudis, A. Cully, V. Vassiliades, J.-B. Mouret, Quality-diversity optimization: A novel branch of stochastic optimization, in: Black Box Optimization, Machine Learning, and No-Free Lunch Theorems, Springer International Publishing, Cham, 2021, pp. 109–135, http://dx.doi.org/10.1007/978-3-030-66515-9_4.
[18] A. Pétrowski, A clearing procedure as a niching method for genetic algorithms, in: Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 798–803.
[19] T. Ulrich, Exploring Structural Diversity in Evolutionary Algorithms (Ph.D. thesis), ETH Zurich, 2012, http://dx.doi.org/10.3929/ethz-a-007562769, Supervisors: Lothar Thiele, Kalyanmoy Deb. URL http://hdl.handle.net/20.500.11850/56473.
[20] A. Ecoffet, J. Huizinga, J. Lehman, et al., First return, then explore, Nature 590 (2021) 580–586, http://dx.doi.org/10.1038/s41586-020-03157-9.
[21] T. Gangwani, J. Peng, Y. Zhou, Harnessing distribution ratio estimators for learning agents with quality and diversity, in: Proceedings of the 2020 Conference on Robot Learning, in: Proceedings of Machine Learning Research, vol. 155, PMLR, 2021, pp. 2206–2215, URL https://proceedings.mlr.press/v155/gangwani21a.html.
[22] D.M. Bossens, D. Tarapore, QED: Using quality-environment-diversity to evolve resilient robot swarms, IEEE Trans. Evol. Comput. 25 (2) (2021) 346–357, http://dx.doi.org/10.1109/TEVC.2020.3036578.
[23] K. Sfikas, A. Liapis, G.N. Yannakakis, Monte Carlo elites: quality-diversity selection as a multi-armed bandit problem, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '21, ACM, 2021, pp. 180–188, http://dx.doi.org/10.1145/3449639.3459321.
[24] M.C. Fontaine, S. Nikolaidis, Differentiable quality diversity, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Curran Associates Inc., Red Hook, NY, USA, 2021.
[25] S.C. Smith, B. Lim, H. Janmohamed, A. Cully, Quality-diversity optimisation on a physical robot through dynamics-aware and reset-free learning, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, GECCO '23 Companion, 2023, pp. 171–174, http://dx.doi.org/10.1145/3583133.3590625.
[26] J. Huber, F. Hélénon, M. Kappel, E. Chelly, M. Khoramshahi, F.B. Amar, S. Doncieux, Speeding up 6-dof grasp sampling with quality-diversity, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2024, pp. 2985–2991, http://dx.doi.org/10.1109/IROS58592.2024.10801391.
[27] P. Templier, L. Grillotti, E. Rachelson, D. Wilson, A. Cully, Quality with just enough diversity in evolutionary policy search, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'24, ACM, New York, NY, USA, 2024, pp. 105–113, http://dx.doi.org/10.1145/3638529.3654047, GECCO '24.
[28] A. Salehi, A. Coninx, S. Doncieux, Few-shot quality-diversity optimization, IEEE Robotics Autom. Lett. 7 (2) (2022) 4424–4431, http://dx.doi.org/10.1109/LRA.2022.3148438.
[29] M. Faldor, F. Chalumeau, M. Flageat, A. Cully, MAP-elites with descriptor-conditioned gradients and archive distillation into a single policy, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 138–146, http://dx.doi.org/10.1145/3583131.3590503.
[30] A. Feiden, J. Garcke, Overcoming deceptive rewards with quality-diversity, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, in: GECCO '23 Companion, Association for Computing Machinery, New York, NY, USA, 2023, pp. 279–282, http://dx.doi.org/10.1145/3583133.3590741.
[31] R. Boldi, L. Spector, Can the problem-solving benefits of quality diversity be obtained without explicit diversity maintenance? in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, in: GECCO '23 Companion, Association for Computing Machinery, New York, NY, USA, 2023, pp. 2152–2156, http://dx.doi.org/10.1145/3583133.3596336.
[32] M. Flageat, A. Cully, Uncertain quality-diversity: Evaluation methodology and new methods for quality-diversity in uncertain domains, IEEE Trans. Evol. Comput. 28 (4) (2024) 891–902, http://dx.doi.org/10.1109/TEVC.2023.3273560.
[33] J. Bongard, et al., Resilient machines through continuous self-modeling, Science 314 (2006) 1118–1121, http://dx.doi.org/10.1126/science.1133687.
[34] A. Eiben, J. Smith, From evolutionary computation to the evolution of things, Nature 521 (2015) 476–482, http://dx.doi.org/10.1038/nature14544.
[35] M. Flageat, F. Chalumeau, A. Cully, Empirical analysis of PGA-MAP-elites for neuroevolution in uncertain domains, ACM Trans. Evol. Learn. Optim. 3 (1) (2023) http://dx.doi.org/10.1145/3577203.
[36] F.D. Rold, O. Witkovski, N. Aubert-Kato, Synaptic pruning with MAP-elites, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 639–642, http://dx.doi.org/10.1145/3520304.3528813.
[37] B. Lim, M. Flageat, A. Cully, Efficient exploration using model-based quality-diversity with gradients, in: Deep Reinforcement Learning Workshop NeurIPS 2022, 2022, URL https://openreview.net/forum?id=Dqw6SJq18He.
[38] V. Macé, R. Boige, F. Chalumeau, T. Pierrot, G. Richard, N. Perrin-Gilbert, The quality-diversity transformer: Generating behavior-conditioned trajectories with decision transformers, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1221–1229, http://dx.doi.org/10.1145/3583131.3590433.
[39] M. González-Duque, R.B. Palm, D. Ha, S. Risi, Finding game levels with the right difficulty in a few trials through intelligent trial-and-error, in: 2020 IEEE Conference on Games, CoG, 2020, pp. 503–510, http://dx.doi.org/10.1109/CoG47356.2020.9231548.
[40] D. Perez-Liebana, C. Guerrero-Romero, A. Dockhorn, L. Xu, J. Hurtado, D. Jeurissen, Generating diverse and competitive play-styles for strategy games, in: 2021 IEEE Conference on Games, CoG, IEEE Press, 2021, pp. 1–8, http://dx.doi.org/10.1109/CoG52621.2021.9619094.
[41] B.M.F. Viana, L.T. Pereira, C.F.M. Toledo, Illuminating the space of dungeon maps, locked-door missions and enemy placement through MAP-elites, 2022, ArXiv abs/2202.09301. URL https://api.semanticscholar.org/CorpusID:246996951.
[42] Y. Zhang, M.C. Fontaine, A.K. Hoover, S. Nikolaidis, Deep surrogate assisted MAP-elites for automated hearthstone deckbuilding, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 158–167, http://dx.doi.org/10.1145/3512290.3528718.
[43] P. Kent, J. Branke, Bayesian quality diversity search with interactive illumination, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, 2023, pp. 1019–1026, http://dx.doi.org/10.1145/3583131.3590486.
[44] K. Xue, Y. Wang, C. Guan, L. Yuan, H. Fu, Q. Fu, C. Qian, Y. Yu, Heterogeneous multi-agent zero-shot coordination by coevolution, IEEE Trans. Evol. Comput. (2024) 1, http://dx.doi.org/10.1109/TEVC.2024.3485177.
[45] J. Verhellen, J.V. den Abeele, Illuminating elite patches of chemical space, Chem. Sci. 11 (42) (2020) 11485–11491, http://dx.doi.org/10.1039/d0sc03544k, This journal is ©The Royal Society of Chemistry.
[46] J. Verhellen, Bayesian illumination: Inference and quality-diversity accelerate generative molecular models, ChemRxiv (2024) http://dx.doi.org/10.26434/chemrxiv-2024-tqf0x, This content is a preprint and has not been peer-reviewed.
[47] R. Boige, G. Richard, J. Dona, T. Pierrot, A. Cully, Gradient-informed quality diversity for the illumination of discrete spaces, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 119–128, http://dx.doi.org/10.1145/3583131.3590407.
[48] K. Mosphilis, V. Vassiliades, Optimizing camera placement for chicken farm monitoring, in: P. García-Sánchez, E. Hart, S.L. Thomson (Eds.), Applications of Evolutionary Computation, Springer Nature Switzerland, Cham, 2025, pp. 354–369.

[49] T. Galanos, A. Liapis, G.N. Yannakakis, R. Koenig, ARCH-elites: Quality-diversity for urban design, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 313–314, http://dx.doi.org/10.1145/3449726.3459490.

[50] N. Urquhart, E. Hart, Optimisation and illumination of a real-world workforce scheduling and routing application (WSRP) via map-elites, in: A. Auger, C.M. Fonseca, N. Lourenço, P. Machado, L. Paquete, D. Whitley (Eds.), Parallel Problem Solving from Nature – PPSN XV, in: Lecture Notes in Computer Science, vol. 11101, Springer, Cham, 2018, pp. 536–547, http://dx.doi.org/10.1007/978-3-319-99253-2_39.

[51] N. Urquhart, S. Höhl, E. Hart, An illumination algorithm approach to solving the micro-depot routing problem, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1347–1355, http://dx.doi.org/10.1145/3321707.3321767.

[52] A. Ferigo, L.L. Custode, G. Iacca, Quality diversity evolutionary learning of decision trees, in: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 425–432, http://dx.doi.org/10.1145/3555776.3577591.

[53] A. Marrero, E. Segredo, E. Hart, J. Bossek, A. Neumann, Generating diverse and discriminatory knapsack instances by searching for novelty in variable dimensions of feature-space, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 312–320, http://dx.doi.org/10.1145/3583131.3590504.

[54] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application, L. Erlbaum Associates Inc., USA, 1987, pp. 41–49.

[55] J. Lehman, K.O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, Evol. Comput. 19 (2) (2011) 189–223.

[56] J. Lehman, K.O. Stanley, Evolving a diversity of virtual creatures through novelty search and local competition, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ACM, 2011, pp. 211–218.

[57] A. Cully, Y. Demiris, Hierarchical behavioral repertoires with unsupervised descriptors, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 69–76, http://dx.doi.org/10.1145/3205455.3205571.

[58] J.-B. Mouret, J. Clune, Illuminating search spaces by mapping elites, 2015, arXiv:1504.04909. URL https://arxiv.org/abs/1504.04909.

[59] J.K. Pugh, L.B. Soros, P.A. Szerlip, K.O. Stanley, Confronting the challenge of quality diversity, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 967–974, http://dx.doi.org/10.1145/2739480.2754664.

[60] A. Gaier, A. Asteroth, J.-B. Mouret, Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 99–106, http://dx.doi.org/10.1145/3071178.3071282.

[61] V. Vassiliades, K. Chatzilygeroudis, J.-B. Mouret, Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm, IEEE Trans. Evol. Comput. 22 (4) (2018) 623–630, http://dx.doi.org/10.1109/TEVC.2017.2735550.

[62] L. Cazenille, Qdpy: A python framework for quality-diversity, 2018, URL https://gitlab.com/leo.cazenille/qdpy.

[63] M.C. Fontaine, J. Togelius, S. Nikolaidis, A.K. Hoover, Covariance matrix adaptation for the rapid illumination of behavior space, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO, 2020.

[64] J.-B. Mouret, G. Maguire, Quality diversity for multi-task optimization, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO '20, ACM, 2020, http://dx.doi.org/10.1145/3377930.3390203.

[65] O. Nilsson, A. Cully, Policy gradient assisted MAP-elites, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 866–875, http://dx.doi.org/10.1145/3449639.3459304.

[66] B. Tjanaka, M.C. Fontaine, D.H. Lee, Y. Zhang, N.R. Balam, N. Dennler, S.S. Garlanka, N.D. Klapsis, S. Nikolaidis, Pyribs: A bare-bones python library for quality diversity optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 220–229, http://dx.doi.org/10.1145/3583131.3590374.

[67] T. Pierrot, G. Richard, K. Beguir, A. Cully, Multi-objective quality diversity optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'22, ACM, 2022, pp. 139–147.

[68] F. Chalumeau, B. Lim, R. Boige, M. Allard, L. Grillotti, M. Flageat, V. Macé, G. Richard, A. Flajolet, T. Pierrot, A. Cully, QDax: A library for quality-diversity and population-based algorithms with hardware acceleration, J. Mach. Learn. Res. 25 (1) (2024) http://dx.doi.org/10.48550/arXiv.2308.03665.

[69] J. Bossek, D. Sudholt, Runtime analysis of quality diversity algorithms, in: Proceedings of the 25th ACM Conference on Genetic and Evolutionary Computation, GECCO, ACM, Lisbon, Portugal, 2023, pp. 1546–1554.

[70] B. Lim, M. Allard, L. Grillotti, A. Cully, Accelerated quality-diversity through massive parallelism, Trans. Mach. Learn. Research (2023) URL https://openreview.net/forum?id=znNITCJyTI.

[71] S. Vaid, S. Urschel, D.M. Olson, M. Guzdial, Quality-diversity transfer learning (QDTL), 2023, ICLR 2023. URL https://openreview.net/forum?id=5921.

[72] C. Qian, K. Xue, R.-J. Wang, Quality-diversity algorithms can provably be helpful for optimization, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24, 2024, http://dx.doi.org/10.24963/ijcai.2024/773.

[73] L. Grillotti, M. Faldor, B.G. León, A. Cully, Quality-diversity actor-critic: Learning high-performing and diverse behaviors via value and successor features critics, in: Forty-First International Conference on Machine Learning, ICML, 2024, URL https://openreview.net/forum?id=ISG3l8nXrI.

[74] H. Qin, Y. Xiang, F. Liu, Y. Han, Y. Wang, Enhancing quality-diversity algorithm by reinforcement learning for flexible job shop scheduling with transportation constraints, Swarm Evol. Comput. 93 (2025) 101849, http://dx.doi.org/10.1016/j.swevo.2025.101849, URL https://www.sciencedirect.com/science/article/pii/S2210650225000070.

[75] X. He, Q.-K. Pan, L. Gao, L. Wang, P.N. Suganthan, A greedy cooperative co-evolutionary algorithm with problem-specific knowledge for multiobjective flowshop group scheduling problems, IEEE Trans. Evol. Comput. 27 (3) (2023) 430–444.

[76] C. Lu, L. Gao, J. Yi, X. Li, Energy-efficient scheduling of distributed flow shop with heterogeneous factories: A real-world case from automobile industry in China, IEEE Trans. Ind. Inform. 17 (10) (2021) 6687–6696, http://dx.doi.org/10.1109/TII.2020.3043734.

[77] Y. Wang, Y. Han, Y. Wang, X. Wang, Y. Liu, K. Gao, Reinforcement learning-assisted memetic algorithm for sustainability-oriented multiobjective distributed flow shop group scheduling, IEEE Trans. Syst. Man Cybern. Syst. (2025) 1–15, http://dx.doi.org/10.1109/TSMC.2024.3518625.

[78] J.-P. Huang, L. Gao, X.-Y. Li, A hierarchical multi-action deep reinforcement learning method for dynamic distributed job-shop scheduling problem with job arrivals, IEEE Trans. Autom. Sci. Eng. 22 (2025) 2501–2513, http://dx.doi.org/10.1109/TASE.2024.3380644.

[79] H. Qin, Y. Han, Y. Wang, Y. Liu, J. Li, Q. Pan, Intelligent optimization under blocking constraints: A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem, Knowl.-Based Syst. 258 (2022) 109962, http://dx.doi.org/10.1016/j.knosys.2022.109962, URL https://www.sciencedirect.com/science/article/pii/S0950705122010553.

[80] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.

[81] A. Gaier, A. Asteroth, J.-B. Mouret, Discovering representations for black-box optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, Vol. 11, 2020.

[82] M. Flageat, A. Cully, Fast and stable MAP-elites in noisy domains using deep grids, in: The 2020 Conference on Artificial Life, in: ALIFE 2020, MIT Press, 2020, http://dx.doi.org/10.1162/isal_a_00316.

[83] H. He, P. Wu, C. Bai, H. Lai, L. Wang, L. Pan, X. Hu, W. Zhang, Bridging the sim-to-real gap from the information bottleneck perspective, in: Annual Conference on Robot Learning, CORL, 2024, Oral presentation. URL https://arxiv.org/abs/2305.18464.

[84] M. Fontaine, S. Nikolaidis, Covariance matrix adaptation MAP-annealing, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 456–465, http://dx.doi.org/10.1145/3583131.3590389.

[85] E. Conti, V. Madhavan, F.P. Such, J. Lehman, K.O. Stanley, J. Clune, Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS '18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 5032–5043.

[86] B. Lim, M. Flageat, A. Cully, Understanding the synergies between quality-diversity and deep reinforcement learning, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1212–1220, http://dx.doi.org/10.1145/3583131.3590388.

[87] A. Gaier, J. Stoddart, L. Villaggi, P.J. Bentley, T-DominO - exploring multiple criteria with quality-diversity and the tournament dominance objective, in: PPSN (2), 2022, pp. 263–277, URL https://doi.org/10.1007/978-3-031-14721-0_19.

[88] T. Ulrich, J. Bader, E. Zitzler, Integrating decision space diversity into hypervolume-based multiobjective search, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, 2010, pp. 455–462.

[89] T. Ulrich, L. Thiele, Maximizing population diversity in single-objective optimization, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, 2011, pp. 641–648.

[90] B. Doerr, W. Gao, F. Neumann, Runtime analysis of evolutionary diversity maximization for oneminmax, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, 2016, pp. 557–564.

[91] B. Doerr, C. Doerr, F. Neumann, Fast re-optimization via structural diversity, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 233–241.
[92] W. Gao, S. Nallaperuma, F. Neumann, Feature-based diversity optimization for problem instance classification, Evol. Comput. 29 (1) (2021) 107–128.
[93] J. Bossek, P. Kerschke, A. Neumann, M. Wagner, F. Neumann, H. Trautmann, Evolving diverse TSP instances by means of novel and creative mutation operators, in: Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, 2019, pp. 58–71.
[94] A. Neumann, W. Gao, M. Wagner, F. Neumann, Evolutionary diversity optimization using multi-objective indicators, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 837–845.
[95] A. Nikfarjam, J. Bossek, A. Neumann, F. Neumann, Entropy-based evolutionary diversity optimisation for the traveling salesperson problem, in: Genetic and Evolutionary Computation Conference, 2021.
[96] A. Neumann, W. Gao, C. Doerr, F. Neumann, M. Wagner, Discrepancy-based evolutionary diversity optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2018, pp. 991–998.
[97] G. Rudolph, Self-adaptive mutations may lead to premature convergence, IEEE Trans. Evol. Comput. 5 (4) (2001) 410–414.
[98] B. Sareni, L. Krahenbuhl, Fitness sharing and niching methods revisited, IEEE Trans. Evol. Comput. 2 (1998) 97–106.
[99] G. Singh, K. Deb, Comparison of multi-modal optimization algorithms based on evolutionary algorithms, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 1305–1312.
[100] D.E. Goldberg, J. Richardson, et al., Genetic algorithms with sharing for multimodal function optimization, in: Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, NJ, 1987, pp. 41–49.
[101] C.-G. Lee, D.-H. Cho, H.-K. Jung, Niching genetic algorithm with restricted competition selection for multimodal function optimization, IEEE Trans. Magn. 35 (3) (1999) 1722–1725.
[102] O. Shir, M. Emmerich, T. Bäck, M. Vrakking, Conceptual designs in laser pulse shaping obtained by niching in evolution strategies, in: EUROGEN 2007, 2007.
[103] A. Cully, J.-B. Mouret, S. Doncieux, Quality-diversity optimisation, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 864–889, http://dx.doi.org/10.1145/3520304.3533637.
[104] J. Huber, F. Helenon, M. Coninx, F.B. Amar, S. Doncieux, Quality diversity under sparse interaction and sparse reward: Application to grasping in robotics, Evol. Comput. (2025) http://dx.doi.org/10.1162/evco_a_00363.
[105] A. Cully, J.-B. Mouret, Behavioral repertoire learning in robotics, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, ACM, 2013, pp. 175–182.
[106] S. Doncieux, J.-B. Mouret, Behavioral diversity measures for evolutionary robotics, in: IEEE Congress on Evolutionary Computation, IEEE, 2010, pp. 1–8.
[107] J.-B. Mouret, Novelty-based multiobjectivization, in: New Horizons in Evolutionary Robotics, Springer, 2011, pp. 139–154.
[108] J. Clune, J.-B. Mouret, H. Lipson, The evolutionary origins of modularity, Proc. R. Soc. Lond. B Biol. Sci. 280 (1755) (2013).
[109] P. Kent, J. Branke, 2020, arXiv:2005.04320.[link]. URL https://arxiv.org/abs/2005.04320.
[110] T. Zohdinasab, V. Riccio, A. Gambi, P. Tonella, Efficient and effective feature space exploration for testing deep learning systems, ACM Trans. Softw. Eng. Methodol. 32 (2) (2023) http://dx.doi.org/10.1145/3544792.
[111] B. Tjanaka, M.C. Fontaine, J. Togelius, S. Nikolaidis, Approximating gradients for differentiable quality diversity in reinforcement learning, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1102–1111, http://dx.doi.org/10.1145/3512290.3528705.
[112] B. Tjanaka, M.C. Fontaine, D.H. Lee, A. Kalkar, S. Nikolaidis, Training diverse high-dimensional controllers by scaling covariance matrix adaptation MAP-annealing, IEEE Robotics Autom. Lett. 8 (10) (2023) 6771–6778, http://dx.doi.org/10.1109/LRA.2023.3313012.
[113] A. Nikfarjam, A.V. Do, F. Neumann, Analysis of quality diversity algorithms for the knapsack problem, in: Proceedings of the 17th International Conference on Parallel Problem Solving from Nature, PPSN, Dortmund, Germany, 2022, pp. 413–427.
[114] A. Cully, Autonomous skill discovery with quality-diversity and unsupervised descriptors, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 81–89, http://dx.doi.org/10.1145/3321707.3321804.
[115] G. Paolo, A. Laflaquière, A. Coninx, S. Doncieux, Unsupervised learning and exploration of reachable outcome space, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, 2020, pp. 2379–2385, http://dx.doi.org/10.1109/ICRA40945.2020.9196819.
[116] L. Grillotti, A. Cully, Relevance-guided unsupervised discovery of abilities with quality-diversity algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, ACM, 2022, pp. 77–85, http://dx.doi.org/10.1145/3512290.3528837.
[117] L. Ding, J. Zhang, J. Clune, L. Spector, J. Lehman, Quality diversity through human feedback: towards open-ended diversity-driven optimization, in: Proceedings of the 41st International Conference on Machine Learning, ICML '24, JMLR.org, 2024.
[118] A. Medina, M. Richey, M. Mueller, J. Schrum, Evolving flying machines in minecraft using quality diversity, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, ACM, 2023, pp. 1418–1426, http://dx.doi.org/10.1145/3583131.3590352.
[119] A. Medina, M. Richey, M. Mueller, J. Schrum, Evolving flying machines in minecraft using quality diversity, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1418–1426, http://dx.doi.org/10.1145/3583131.3590352.
[120] M. Beukman, C.W. Cleghorn, S. James, Procedural content generation using neuroevolution and novelty search for diverse video game levels, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1028–1037, http://dx.doi.org/10.1145/3512290.3528701.
[121] R.M. Hierons, M. Li, X. Liu, J.A. Parejo, S. Segura, X. Yao, Many-objective test suite generation for software product lines, ACM Trans. Softw. Eng. Methodol. 29 (1) (2020) http://dx.doi.org/10.1145/3361146.
[122] S. Akbarova, F. Dobslaw, F.G. de Oliveira Neto, R. Feldt, SETBVE: Quality-diversity driven exploration of software boundary behaviors, 2025, arXiv:2505.19736. URL https://arxiv.org/abs/2505.19736.
[123] Q. Mazouni, H. Spieker, A. Gotlieb, M. Acher, Testing for fault diversity in reinforcement learning, in: Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test (AST 2024), AST '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 136–146, http://dx.doi.org/10.1145/3644032.3644458.
[124] N. Urquhart, E. Hart, W. Hutcheson, Using MAP-elites to support policy making around workforce scheduling and routing, At - Autom. 68 (2) (2020) 110–117, http://dx.doi.org/10.1515/auto-2019-0107, [cited 2025-02-21].
[125] H. Janmohamed, M. Wolinska, S. Surana, T. Pierrot, A. Walsh, A. Cully, Multi-objective quality-diversity for crystal structure prediction, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'24, ACM, 2024, pp. 1273–1281.
[126] M. Flageat, L. Grillotti, A. Cully, Benchmark tasks for quality-diversity applied to uncertain domains, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, in: GECCO '23 Companion, Association for Computing Machinery, New York, NY, USA, 2023, pp. 2157–2162, http://dx.doi.org/10.1145/3583133.3596326.
[127] Y. Zhang, M.C. Fontaine, V. Bhatt, S. Nikolaidis, J. Li, Multi-robot coordination and layout design for automated warehousing, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23, 2023, http://dx.doi.org/10.24963/ijcai.2023/611.
[128] J. Huber, F. Hélénon, H. Watrelot, F.B. Amar, S. Doncieux, Domain randomization for sim2real transfer of automatically generated grasping datasets, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 4112–4118, http://dx.doi.org/10.1109/ICRA57147.2024.10610677.
[129] L. Grillotti, M. Faldor, B.G. León, A. Cully, Quality-diversity actor-critic: learning high-performing and diverse behaviors via value and successor features critics, in: Proceedings of the 41st International Conference on Machine Learning, ICML '24, JMLR.org, 2024.
[130] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, IEEE Trans. Evol. Comput. 21 (1) (2017) 131–152, http://dx.doi.org/10.1109/TEVC.2016.2587808.
[131] Y. Xiang, Y. Zhou, X. Yang, H. Huang, A many-objective evolutionary algorithm with Pareto-adaptive reference points, IEEE Trans. Evol. Comput. 24 (1) (2020) 99–113, http://dx.doi.org/10.1109/TEVC.2019.2909636.
[132] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: Applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[133] A. Cully, J.-B. Mouret, Evolving a behavioral repertoire for a walking robot, Evol. Comput. 24 (1) (2016) 59–88, http://dx.doi.org/10.1162/EVCO_a_00143.
[134] R. Bahlous-Boldi, M. Faldor, L. Grillotti, H. Janmohamed, L. Coiffard, L. Spector, A. Cully, Dominated novelty search: Rethinking local competition in quality-diversity, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 104–112, http://dx.doi.org/10.1145/3712256.3726310.
[135] L. Yang, Innovative applications and challenges of dobby slot machine algorithms in online gaming, ITM Web Conf. 73 (2025) 01023, http://dx.doi.org/10.1051/itmconf/20257301023.
[136] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: G.J. Rawlins (Ed.), in: Foundations of Genetic Algorithms, vol. 1, Elsevier, 1991, pp. 69–93, http://dx.doi.org/10.1016/B978-0-08-050684-5.50008-2, URL https://www.sciencedirect.com/science/article/pii/B9780080506845500082.
[137] H. Janmohamed, T. Pierrot, A. Cully, Improving the data efficiency of multi-objective quality-diversity through gradient assistance and crowding exploration, in: Proceedings of Genetic and Evolutionary Computation Conference, GECCO'23, ACM, 2023, pp. 165–173.

[138] V. Vassiliades, J.-B. Mouret, Discovering the elite hypervolume by leveraging interspecies correlation, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, 2018.

[139] J. Nordmoen, E. Samuelsen, K.O. Ellefsen, K. Glette, Dynamic mutation in MAP-elites for robotic repertoire generation, in: Artificial Life Conference Proceedings, MIT Press, 2018, pp. 598–605.

[140] J. Nordmoen, F. Veenstra, K. Glette, K.O. Ellefsen, MAP-elites enables powerful stepping stones and diversity for modular robotics, Front. Robotics AI 8 (2021) 639173, http://dx.doi.org/10.3389/frobt.2021.639173.

[141] M. Charity, A. Khalifa, J. Togelius, Baba is y'all: Collaborative mixed-initiative level design, in: 2020 IEEE Conference on Games (CoG), 2020, pp. 542–549, http://dx.doi.org/10.1109/CoG47356.2020.9231807.

[142] O. Withington, Illuminating super mario bros: quality-diversity within platformer level generation, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 223–224, http://dx.doi.org/10.1145/3377929.3390043.

[143] N.B. Urquhart, E. Hart, A. Judson, Multi-modal employee routing with time windows in an urban environment, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, in: GECCO Companion '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1503–1504, http://dx.doi.org/10.1145/2739482.2764649.

[144] A. Nikfarjam, A. Neumann, F. Neumann, On the use of quality diversity algorithms for the travelling thief problem, ACM Trans. Evol. Learn. Optim. 4 (2) (2024) http://dx.doi.org/10.1145/3641109.

[145] T. Pierrot, V. Macé, F. Chalumeau, A. Flajolet, G. Cideron, K. Beguir, A. Cully, O. Sigaud, N. Perrin-Gilbert, Diversity policy gradient for sample efficient quality-diversity optimization, in: Proceedings of the 24th ACM Conference on Genetic and Evolutionary Computation, GECCO, 2022, pp. 1075–1083.

[146] Z. Wan, X. Yu, D.M. Bossens, Y. Lyu, Q. Guo, F.X. Fan, I. Tsang, Quality diversity imitation learning, 2025, URL https://openreview.net/forum?id=ozhRaoRGyl.

[147] N. Justesen, S. Risi, J.-B. Mouret, Map-elites for noisy domains by adaptive sampling, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 121–122, http://dx.doi.org/10.1145/3319619.3322060.

[148] S. Mkhatshwa, G. Nitschke, Body and brain quality-diversity in robot swarms, ACM Trans. Evol. Learn. 5 (1) (2025) http://dx.doi.org/10.1145/3664656.

[149] G. Nadizar, E. Medvet, D.G. Wilson, Enhancing adaptability in embodied agents: A multi-quality-diversity approach, IEEE Trans. Evol. Comput. (2025) 1, http://dx.doi.org/10.1109/TEVC.2025.3596746.

[150] D. Morrison, P. Corke, J. Leitner, Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation, IEEE Robotics Autom. Lett. 5 (3) (2020) 4368–4375, http://dx.doi.org/10.1109/LRA.2020.2992195.

[151] S. Zhong, D. Berenson, N. Fazeli, CHSEL: Producing Diverse Plausible Pose Estimates from Contact and Free Space Data, in: Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea, 2023, http://dx.doi.org/10.15607/RSS.2023.XIX.077.

[152] C. Colas, V. Madhavan, J. Huizinga, J. Clune, Scaling MAP-elites to deep neuroevolution, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 67–75, http://dx.doi.org/10.1145/3377930.3390217.

[153] S. Batra, B. Tjanaka, M.C. Fontaine, A. Petrenko, S. Nikolaidis, G.S. Sukhatme, Proximal policy gradient arborescence for quality diversity reinforcement learning, in: The Twelfth International Conference on Learning Representations, 2024, URL https://openreview.net/forum?id=TFKIfhvdmZ.

[154] R.-J. Wang, K. Xue, C. Guan, C. Qian, Quality-diversity with limited resources, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, Vienna, Austria, 2024.

[155] K. Xue, R.-J. Wang, P. Li, D. Li, J. HAO, C. Qian, Sample-efficient quality-diversity by cooperative coevolution, in: The Twelfth International Conference on Learning Representations, 2024, URL https://openreview.net/forum?id=JDud6zbpFv.

[156] S. Earle, J. Snider, M.C. Fontaine, S. Nikolaidis, J. Togelius, Illuminating diverse neural cellular automata for level generation, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 68–76, http://dx.doi.org/10.1145/3512290.3528754.

[157] K. Sfikas, A. Liapis, G.N. Yannakakis, Diverse level generation via machine learning of quality diversity, in: Proceedings of the 20th International Conference on the Foundations of Digital Games, FDG '25, Association for Computing Machinery, New York, NY, USA, 2025, http://dx.doi.org/10.1145/3723498.3723843.

[158] A. Gaier, A. Asteroth, J.-B. Mouret, Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination, 2017, http://dx.doi.org/10.1145/3071178.3071282, arXiv:1702.03713, URL https://arxiv.org/abs/1702.03713.

[159] A. Gaier, J. Stoddart, L. Villaggi, S. Sudhakaran, Generative design through quality-diversity data synthesis and language models, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 823–831, http://dx.doi.org/10.1145/3638529.3654138.

[160] J.-B. Mouret, S. Doncieux, Sferesv2: Evolvin' in the multi-core world, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8, http://dx.doi.org/10.1109/CEC.2010.5586158.

[161] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: Composable transformations of Python+NumPy programs, 2018, http://github.com/google/jax.

[162] J. Mouret, J. Clune, Python3 map-elites, 2019, URL https://github.com/resibots/pymap_elites.

[163] H. Qin, Y. Xiang, Y. Han, Y. Wang, Q. Chen, P. Zhou, Intelligent scheduling approach for distributed flexible job-shop with deep reinforcement learning and quality-diversity optimization, Sci. Sinica Informationis (2025) http://dx.doi.org/10.1360/SSI-2025-0113.

[164] Y. Han, D. Gong, Y. Jin, Q. Pan, Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns, IEEE Trans. Cybern. 49 (1) (2019) 184–197, http://dx.doi.org/10.1109/TCYB.2017.2771213.

[165] R. Li, W. Gong, C. Lu, L. Wang, A learning-based memetic algorithm for energy-efficient flexible job-shop scheduling with type-2 fuzzy processing time, IEEE Trans. Evol. Comput. 27 (3) (2023) 610–620, http://dx.doi.org/10.1109/TEVC.2022.3175832.

[166] S. Silaich, S. Gupta, Feature selection in high dimensional data: A review, in: S. Kumar, H. Sharma, K. Balachandran, J.H. Kim, J.C. Bansal (Eds.), Third Congress on Intelligent Systems, CIS 2022, in: Lecture Notes in Networks and Systems, vol. 608, Springer, Singapore, 2023, http://dx.doi.org/10.1007/978-981-19-9225-4_51.

[167] N. Kashiri, A. Abate, S.J. Abram, A. Albu-Schaffer, P.J. Clary, M. Daley, S. Faraji, R. Furnemont, M. Garabini, H. Geyer, A.M. Grabowski, J. Hurst, J. Malzahn, G. Mathijssen, D. Remy, W. Roozing, M. Shahbazi, S.N. Simha, J.B. Song, N. Smit-Anseeuw, S. Stramigioli, B. Vanderborght, Y. Yesilevskiy, N. Tsagarakis, An overview on principles for energy efficient robot locomotion, Front. Robotics AI 5 (2018) 129, http://dx.doi.org/10.3389/frobt.2018.00129.

[168] R.-J. Wang, K. Xue, Y. Wang, P. Yang, H. Fu, Q. FU, C. Qian, Diversity from human feedback, in: Second Agent Learning in Open-Endedness Workshop, 2023, URL https://openreview.net/forum?id=JK94vOwU29.

[169] J. Jiang, H. Piao, Y. Fu, Y. Hao, C. Jiang, Z. Wei, X. Yang, Phasic diversity optimization for population-based reinforcement learning, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 272–278, http://dx.doi.org/10.1109/ICRA57147.2024.10610814.

[170] S. Batra, B. Tjanaka, S. Nikolaidis, G. Sukhatme, Quality diversity for robot learning: Limitations and future directions, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, in: GECCO '24 Companion, Association for Computing Machinery, New York, NY, USA, 2024, pp. 587–590, http://dx.doi.org/10.1145/3638530.3654431.

[171] M. Rebolledo, D. Zeeuwe, T. Bartz-Beielstein, A.E. Eiben, Impact of energy efficiency on the morphology and behaviour of evolved robots, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 109–110, http://dx.doi.org/10.1145/3449726.3459489.

[172] G. Öcal, A. Özgövde, Network-aware federated neural architecture search, Future Gener. Comput. Syst. 162 (2025) 107475, http://dx.doi.org/10.1016/j.future.2024.07.053, URL https://www.sciencedirect.com/science/article/pii/S0167739X24004205.

[173] H. Qin, Y. Wang, Y. Han, Q. Chen, J. Li, Adapting a reinforcement learning method for the distributed blocking hybrid flow shop scheduling problem, in: 2021 5th Asian Conference on Artificial Intelligence Technology, ACAIT, 2021, pp. 751–757, http://dx.doi.org/10.1109/ACAIT53529.2021.9731228.

[174] N. Kang, Z. Miao, Q.-K. Pan, W. Li, M.F. Tasgetiren, Multi-objective teaching-learning-based optimizer for a multi-weeding robot task assignment problem, Tsinghua Sci. Technol. 29 (5) (2024) 1249–1265, http://dx.doi.org/10.26599/TST.2023.9010075.

[175] X. Han, Y. Han, B. Zhang, H. Qin, J. Li, Y. Liu, D. Gong, An effective iterative greedy algorithm for distributed blocking flowshop scheduling problem with balanced energy costs criterion, Appl. Soft Comput. 129 (2022) 109502, http://dx.doi.org/10.1016/j.asoc.2022.109502, URL https://www.sciencedirect.com/science/article/pii/S1568494622005920.

[176] X. He, Q.-K. Pan, L. Gao, J.S. Neufeld, J.N. Gupta, Historical information based iterated greedy algorithm for distributed flowshop group scheduling problem with sequence-dependent setup times, Omega 123 (2024) 102997, http://dx.doi.org/10.1016/j.omega.2023.102997, URL https://www.sciencedirect.com/science/article/pii/S0305048323001615.

[177] Y. Hou, H. Wang, Y. Fu, K. Gao, H. Zhang, Multi-objective brain storm optimization for integrated scheduling of distributed flow shop and distribution with maximal processing quality and minimal total weighted earliness and tardiness, Comput. Ind. Eng. 179 (2023) 109217, http://dx.doi.org/10.1016/j.cie.2023.109217, URL https://www.sciencedirect.com/science/article/pii/S0360835223002413.

[178] H.-X. Qin, Y.-Y. Han, Y.-P. Liu, J.-Q. Li, Q.-K. Pan, Xue-Han, A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints, Expert Syst. Appl. 201 (2022) 117256, http://dx.doi.org/10.1016/j.eswa.2022.117256, URL https://www.sciencedirect.com/science/article/pii/S0957417422006315.
[179] S. Yang, H. Huang, F. Luo, Y. Xu, Z. Hao, Local-diversity evaluation assignment strategy for decomposition-based multiobjective evolutionary algorithm, IEEE Trans. Syst. Man Cybern. Syst. 53 (3) (2023) 1697–1709, http://dx.doi.org/10.1109/TSMC.2022.3207457.
[180] X.-R. Tao, Q.-K. Pan, L. Gao, An efficient self-adaptive artificial bee colony algorithm for the distributed resource-constrained hybrid flowshop problem, Comput. Ind. Eng. 169 (2022) 108200, http://dx.doi.org/10.1016/j.cie.2022.108200, URL https://www.sciencedirect.com/science/article/pii/S0360835222002704.
[181] H.-X. Qin, Y.-Y. Han, B. Zhang, L.-L. Meng, Y.-P. Liu, Q.-K. Pan, D.-W. Gong, An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem, Swarm Evol. Comput. 69 (2022) 100992, http://dx.doi.org/10.1016/j.swevo.2021.100992, URL https://www.sciencedirect.com/science/article/pii/S2210650221001541.
[182] J.-P. Huang, L. Gao, X.-Y. Li, C.-J. Zhang, A novel priority dispatch rule generation method based on graph neural network and reinforcement learning for distributed job-shop scheduling, J. Manuf. Syst. 69 (2023) 119–134, http://dx.doi.org/10.1016/j.jmsy.2023.06.007, URL https://www.sciencedirect.com/science/article/pii/S0278612523001176.
[183] Y. Hou, H. Wang, X. Huang, A Q-learning-based multi-objective evolutionary algorithm for integrated green production and distribution scheduling problems, Eng. Appl. Artif. Intell. 127 (2024) 107434, http://dx.doi.org/10.1016/j.engappai.2023.107434, URL https://www.sciencedirect.com/science/article/pii/S0952197623016184.
[184] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling, IEEE Trans. Evol. Comput. 28 (1) (2024) 147–167, http://dx.doi.org/10.1109/TEVC.2023.3255246.
[185] X. Xue, Y. Mei, B. Zhao, M. Zhang, Adaptive similarity feature construction for ontology matching via multi-layer hybrid genetic programming, IEEE Trans. Evol. Comput. (2025) 1, http://dx.doi.org/10.1109/TEVC.2025.3547578.
[186] W. Bai, F. Meng, M. Sun, H. Qin, R. Allmendinger, K.Y. Lee, Differential evolutionary particle swarm optimization with orthogonal learning for wind integrated optimal power flow, Appl. Soft Comput. 160 (2024) 111662, http://dx.doi.org/10.1016/j.asoc.2024.111662, URL https://www.sciencedirect.com/science/article/pii/S1568494624004368.
[187] H. Huang, F. Feng, S. Huang, L. Chen, Z. Hao, Microscale searching algorithm for coupling matrix optimization of automated microwave filter tuning, IEEE Trans. Cybern. 53 (5) (2023) 2829–2840, http://dx.doi.org/10.1109/TCYB.2022.3166225.
[188] A. Sarkar, S. Cooper, Generating and blending game levels via quality-diversity in the latent space of a variational autoencoder, in: Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG '21, Association for Computing Machinery, New York, NY, USA, 2021, http://dx.doi.org/10.1145/3472538.3472545.
[189] N. Rakicevic, A. Cully, P. Kormushev, Policy manifold search: Exploring the manifold hypothesis for diversity-based neuroevolution, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 901–909, http://dx.doi.org/10.1145/3449639.3459320.
[190] J. Ge, J. Liu, J. Zhang, Three-precision iterative refinement with parameter regularization and prediction for solving large sparse linear systems, 2025, arXiv:2501.04229. URL https://arxiv.org/abs/2501.04229.
[191] P. Kent, A. Gaier, J.-B. Mouret, J. Branke, Bayesian optimization for quality diversity search with coupled descriptor functions, IEEE Trans. Evol. Comput. 29 (2) (2025) 302–316, http://dx.doi.org/10.1109/TEVC.2024.3376733.
[192] A. Hagg, D. Wilde, A. Asteroth, T. Bäck, Designing air flow with surrogate-assisted phenotypic niching, in: Parallel Problem Solving from Nature – PPSN XVI: 16th International Conference, PPSN 2020, Leiden, the Netherlands, September 5-9, 2020, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2020, pp. 140–153, http://dx.doi.org/10.1007/978-3-030-58112-1_10.
[193] M. Flageat, J. Huber, F. Helenon, S. Doncieux, A. Cully, Extract-QD framework: A generic approach for quality-diversity in noisy, stochastic or uncertain domains, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 140–148, http://dx.doi.org/10.1145/3712256.3726404.
[194] M. Flageat, H. Janmohamed, B. Lim, A. Cully, Exploring the performance-reproducibility trade-off in quality-diversity, IEEE Trans. Evol. Comput. (2025) 1, http://dx.doi.org/10.1109/TEVC.2025.3548438.
[195] W. Tan, L. Du, W. Buntine, Diversity enhanced active learning with strictly proper scoring rules, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Curran Associates Inc., Red Hook, NY, USA, 2021.
[196] D.M. Bossens, D. Tarapore, Quality-diversity meta-evolution: Customizing behavior spaces to a meta-objective, IEEE Trans. Evol. Comput. 26 (5) (2022) 1171–1181, http://dx.doi.org/10.1109/TEVC.2022.3152384.
[197] Y. Wang, K. Xue, C. Qian, Evolutionary diversity optimization with clustering-based selection for reinforcement learning, in: International Conference on Learning Representations, 2022, URL https://openreview.net/forum?id=74x5BXs4bWD.
[198] X.-R. Tao, Q.-K. Pan, H.-Y. Sang, L. Gao, A.-L. Yang, M. Rong, Non-dominated sorting genetic algorithm-II with Q-learning for the distributed permutation flowshop rescheduling problem, Knowl.-Based Syst. 278 (2023) 110880, http://dx.doi.org/10.1016/j.knosys.2023.110880, URL https://www.sciencedirect.com/science/article/pii/S0950705123006305.
[199] R. Chavhan, H. Gouk, D. Li, T. Hospedales, Quality diversity for visual pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 5384–5394.
[200] M. Stock, D. Van Hauwermeiren, B. De Baets, S. Taelman, D. Marzougui, M. Van Haeverbeke, Quality-diversity methods for the modern data scientist, WIREs Comput. Stat. 17 (4) (2025) e70047, http://dx.doi.org/10.1002/wics.70047, e70047 EOCS-766.R2. arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.70047, URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.70047.
[201] H. Janmohamed, A. Cully, Multi-objective quality-diversity in unstructured and unbounded spaces, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 149–157, http://dx.doi.org/10.1145/3712256.3726394.
[202] J. Schrum, C. Crosby, A quality diversity approach to evolving model rockets, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 1471–1479, http://dx.doi.org/10.1145/3712256.3726338.
[203] K. Antonov, T. Tukker, T. Botari, T. Bäck, A.V. Kononova, N. van Stein, Quality-diversity driven robust evolutionary optimization of optical designs, in: Optical Systems Design, 2024, URL https://api.semanticscholar.org/CorpusID:270574186.
[204] B. Gašperov, S. Begušić, T. Bauman, Z. Kostanjčar, Quality-diversity and novelty search for portfolio optimization and beyond, Comput. Econ. (2025) http://dx.doi.org/10.1007/s10614-025-10985-2.