International Journal of Innovative Computing, Information and Control
Volume **22**, Number **2**, April **2026**

ICIC International ©2026 ISSN 1349-4198
pp. **445–455**

ENHANCING DIFFUSION EVOLUTION WITH SELECTION MECHANISMS

Jun-ichi Kushida

Computer Science Course
Department of Engineering
Faculty of Engineering
Nihon Fukushi University
26-2, Higashi Uzumi-cho, Handa, Aichi 470-3295, Japan
kushida@n-fukushi.ac.jp

Received May 2025; revised September 2025

Abstract. *Recently, Diffusion Evolution has emerged as a novel optimization algorithm that heuristically simulates the reverse diffusion process and shares key similarities with evolutionary computation. It progressively refines a population of candidate solutions by reversing stochastic noise injection inspired by diffusion models. However, the original Diffusion Evolution lacks explicit selection mechanisms, which are essential for guiding the search and maintaining population diversity. In this study, we incorporate three classical selection mechanisms – One-to-One Replacement, Tournament Selection, and Survival Selection – into the Diffusion Evolution framework. Experiments on standard test functions reveal that each strategy offers distinct advantages: OneToOne preserves diversity and mitigates premature convergence in multimodal landscapes; Tournament excels in rapid convergence on unimodal problems; and Survival provides a balance between convergence speed and diversity retention. In particular, on the 30-dimensional Sphere function, the Tournament variant achieved more than four orders of magnitude improvement in final fitness compared to the standard Diffusion Evolution, demonstrating substantially faster convergence. These findings demonstrate that integrating selection mechanisms significantly enhances Diffusion Evolution's performance and that the choice of strategy should align with the problem landscape's modality.*

Keywords: Diffusion Evolution, Evolutionary computation, Selection mechanism, Exploration-exploitation balance

1. **Introduction.** In recent years, one of the major advances in the field of machine learning has been the development of Diffusion Models [1]. Diffusion Models are generative models composed of a forward diffusion process that gradually adds noise to data, and a reverse diffusion process that incrementally removes this noise to recover the original data distribution. The mathematical foundation of these models was initially established by [2] and further developed through score-based generative approaches [3]. This framework has achieved remarkable success in various fields, including image generation, video synthesis, and speech synthesis [4].

Meanwhile, in the fields of biology and computational science, Evolutionary Algorithms (EAs) have long been studied as important methods for solving complex optimization problems. These algorithms mimic the principles of biological evolution – selection, reproduction, and mutation – and are capable of efficiently exploring complex solution spaces. Zhang et al. [5] discovered a deep mathematical equivalence between these two seemingly distinct frameworks: Diffusion Models and EAs. They demonstrated that Di-

DOI: 10.24507/ijicic.22.02.445

ffusion Models can be fundamentally interpreted as EAs and proposed, based on this insight, a new optimization algorithm called Diffusion Evolution.

The core idea of the Diffusion Evolution algorithm is to interpret the fitness landscape as a probability density function, enabling the search process to be framed as probabilistic sampling. Specifically, parameter regions associated with higher (better) fitness values are assigned greater probability density. This effectively reshapes the search space into a distribution where desirable solutions are statistically more likely to be sampled. The algorithm begins with random noise and iteratively applies a reverse diffusion process, progressively reducing stochasticity while guiding the population toward regions of higher fitness. Through this denoising mechanism, the algorithm gradually refines the population, concentrating search efforts on promising areas of the solution space.

However, despite its innovative approach, the standard Diffusion Evolution algorithm lacks explicit selection mechanisms, which are essential in traditional evolutionary algorithms for maintaining selection pressure, guiding the search, and accelerating convergence. Several studies in evolutionary computation have highlighted the critical role of selection mechanisms in balancing exploration and exploitation, improving convergence stability, and enhancing solution diversity [6, 7].

Zhang et al. [5] interpreted that, in Diffusion Evolution, the noise removal process of the diffusion model itself implicitly fulfills the roles of selection and mutation. Therefore, they did not explore how conventional evolutionary operators could be effectively integrated into the Diffusion Evolution framework. As a result, the potential effects of integrating explicit selection mechanisms from traditional evolutionary algorithms into the Diffusion Evolution framework on the search dynamics have not yet been thoroughly investigated. Therefore, in this study, we aim to address this issue through the following contributions.

- We propose the integration of explicit selection mechanisms into the Diffusion Evolution framework.
- We analyze how different selection mechanisms affect convergence behavior and solution diversity through comprehensive experiments.
- We provide insights into the balance between exploration and exploitation in Diffusion Evolution.

Specifically, we propose three enhanced versions of Diffusion Evolution: 1) with One-to-One Replacement, 2) with Tournament Selection, and 3) with Survival Selection. These algorithms combine the mathematical framework of diffusion models with traditional selection pressures from evolutionary algorithms, potentially leading to more efficient optimization methods. By comparing their performance on various benchmark functions with different dimensionalities, we analyze how different selection mechanisms affect optimization performance, convergence speed, and diversity maintenance. Through these contributions, we aim to provide deeper theoretical insights into Diffusion Evolution algorithms and improve their practical performance across a wide range of optimization problems.

The remainder of this paper is organized as follows. Section 2 introduces the fundamentals of the Diffusion Evolution algorithm. Section 3 presents the proposed integration of selection mechanisms into Diffusion Evolution. Section 4 describes the experimental setup and results. Finally, Section 5 concludes this paper and discusses future research directions.

2. Diffusion Evolution Algorithm.

2.1. Reverse diffusion-based optimization framework. The Diffusion Evolution algorithm, proposed by [5], is a new optimization method that applies the mathematical framework of diffusion models to evolutionary computation. As shown in Algorithm 1, this approach views the fitness landscape as a probability density function and performs sampling from high-fitness regions through the reverse process of diffusion models.

The fundamental idea of Diffusion Evolution is to transform the fitness function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ into a probability density function $p(\boldsymbol{x}_0 = \boldsymbol{x}) = g[f(\boldsymbol{x})]$ using a transformation function g, ensuring that parameter regions with higher fitness have higher probability density. Sampling from this distribution is achieved through the reverse process of diffusion models.

Algorithm 1 Diffusion Evolution Algorithm

Require: Population size NP, parameter dimension D, fitness function f, density mapping function g, total evolution steps T, diffusion schedule $\boldsymbol{\alpha}$ and noise schedule $\boldsymbol{\sigma}$.
Ensure: $\alpha_0 \sim 1,\ \alpha_T \sim 0,\ \alpha_i > \alpha_{i+1},\ 0 < \sigma_i < \sqrt{1-\alpha_{i-1}}$
1: $\left[\boldsymbol{x}_T^{(1)}, \dots, \boldsymbol{x}_T^{(NP)}\right] \leftarrow \mathcal{N}\left(0, I^{NP\times D}\right)$ ▷ Initialize population
2: **for** $t = T, T-1, \dots, 2$ **do**
3: $\forall i \in \{1, \dots, NP\} : Q_i \leftarrow g\left[f\left(\boldsymbol{x}_t^{(i)}\right)\right]$ ▷ Fitness values are cached to avoid repeated evaluations
4: **for** $i = 1$ **to** NP **do**
5: $Z \leftarrow \sum_{j=1}^{NP} Q_j \mathcal{N}\left(\boldsymbol{x}_t^{(i)}; \sqrt{\alpha_t}\boldsymbol{x}_t^{(j)}, 1-\alpha_t\right)$
6: $\hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{Z}\sum_{j=1}^{NP} Q_j \mathcal{N}\left(\boldsymbol{x}_t^{(i)}; \sqrt{\alpha_t}\boldsymbol{x}_t^{(j)}, 1-\alpha_t\right)\boldsymbol{x}_t^{(j)}$
7: $\boldsymbol{w} \leftarrow \mathcal{N}\left(0, I^D\right)$
8: $\boldsymbol{x}_{t-1}^{(i)} \leftarrow \sqrt{\alpha_{t-1}}\hat{\boldsymbol{x}}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\frac{\boldsymbol{x}_t^{(i)} - \sqrt{\alpha_t}\hat{\boldsymbol{x}}_0}{\sqrt{1-\alpha_t}} + \sigma_t\boldsymbol{w}$
9: **end for**
10: **end for**

As evident from Algorithm 1, Diffusion Evolution refines solutions through T reverse diffusion steps. The algorithm generates an initial population from a standard normal distribution (Line 1) and proceeds backward in time from step $t = T$ (Line 2). It is important to note that this backward time progression (t decreasing from T to 2) is a distinctive feature of Diffusion Evolution that contrasts with traditional evolutionary computation. While conventional evolutionary algorithms advance forward through generations (from generation 1 to 2, 3, and so on), Diffusion Evolution follows the reverse process of diffusion models, starting with noise and gradually refining it.

In each generation, the fitness of all individuals is first evaluated and transformed using the function g to obtain weight values Q_i (Line 3), where $Q_i \leftarrow g\left[f\left(\boldsymbol{x}_t^{(i)}\right)\right]$. In this implementation, unlike the function g defined in [5] which requires the theoretical optimal value f^*, we adopt an approach that does not require prior knowledge of the optimum. Assuming a minimization problem where lower fitness values $y = f\left(\boldsymbol{x}_t^{(i)}\right)$ are better, the function g applied in Line 3 computes an unnormalized weight based on the fitness:

$$g(y) = \exp\left(-\frac{y}{\tau}\right) \tag{1}$$

where $\tau > 0$ is the temperature parameter controlling the selection pressure. A smaller τ increases the contrast between individuals' fitness, thereby strengthening the selection pressure. In our experiments, we set $\tau = 0.1$ by default, balancing the need for exploration and exploitation throughout the optimization.

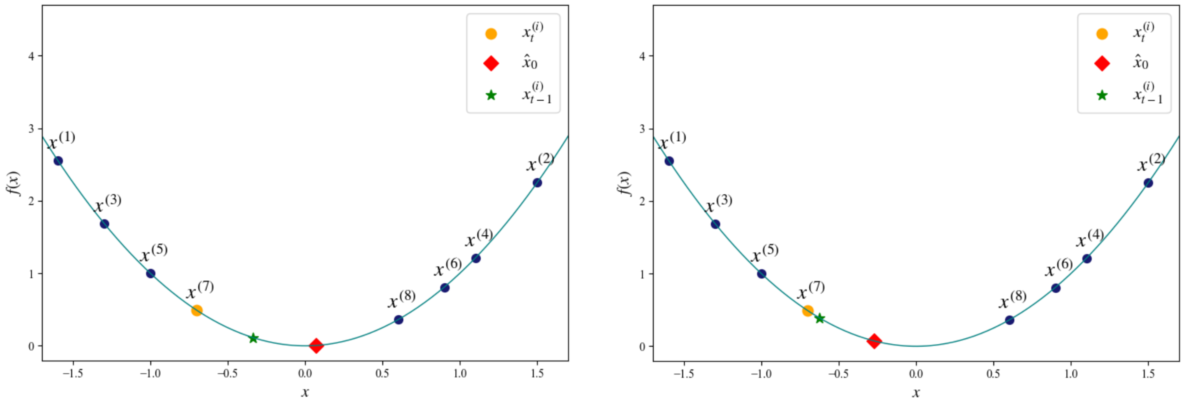
After computing these weight values Q_i, the algorithm proceeds to process each individual (Lines 4-8). For each individual $\boldsymbol{x}_t^{(i)}$, the algorithm computes the normalization constant Z and the high-fitness target point $\hat{\boldsymbol{x}}_0$ (Lines 5-6). To compute these, each fitness-based weight Q_j is multiplied by the Gaussian term $\mathcal{N}\left(\boldsymbol{x}_t^{(i)}; \sqrt{\alpha_t}\boldsymbol{x}_t^{(j)}, 1 - \alpha_t\right)$ for all individuals j. This Gaussian term measures the proximity between the current individual $\boldsymbol{x}_t^{(i)}$ and each population member $\boldsymbol{x}_t^{(j)}$, with its variance $1 - \alpha_t$ changing over time as determined by the diffusion schedule. The normalization constant Z is calculated by summing these weighted proximity values over the entire population. Then, $\hat{\boldsymbol{x}}_0$ is determined as a weighted average of the individuals' positions, using the same fitness-weighted Gaussian terms as in the calculation of Z. Thus, $\hat{\boldsymbol{x}}_0$ can be interpreted as an estimate of where high-fitness solutions are likely to be found from the perspective of the current individual $\boldsymbol{x}_t^{(i)}$.

Here, α_t is a time-dependent parameter that controls the variance of the Gaussian term and thereby regulates the balance between global and local information aggregation throughout the optimization process. When α_t is small (early stages), the variance is large, which allows distant individuals to contribute significantly to the estimation of the target $\hat{\boldsymbol{x}}_0$. This encourages the aggregation of information from the broader population and tends to draw individuals toward a common region, resembling the behavior observed when the social (gBest) coefficient is strong in Particle Swarm Optimization (PSO) [8]. As α_t increases toward 1 in later stages, the variance decreases, restricting the influence to only individuals located near the current solution $\boldsymbol{x}_t^{(i)}$. Consequently, each individual predominantly relies on its own search trajectory and its immediate neighborhood, facilitating local exploitation. This behavior resembles the case where the personal best (pBest) coefficient is set to a large value in PSO.

After estimating $\hat{\boldsymbol{x}}_0$, the algorithm generates a random noise vector $\boldsymbol{w}$ (Line 7) to introduce stochasticity into the update process, serving a role similar to mutation in traditional EAs. In Line 8, the position of the current individual is updated based on the estimated target $\hat{\boldsymbol{x}}_0$, the residual term representing the deviation from $\hat{\boldsymbol{x}}_0$, and a stochastic noise term $\boldsymbol{w}$. This update mechanism balances three forces: convergence to promising regions ($\hat{\boldsymbol{x}}_0$), retention of diversity, and stochastic exploration. As the diffusion process progresses (i.e., as t decreases from T to 2) and α_t increases while σ_t decreases, the influence of noise and the individual's previous position gradually diminishes. This leads to refined, exploitative search behavior in later stages.

2.2. **Effect of $\boldsymbol{\alpha_t}$ on the update behavior.** To visualize how α_t influences the update behavior, Figure 1 illustrates the update process described above using a simple one-dimensional unimodal function, where lower values of $f(\boldsymbol{x})$ indicate better solutions. In both subfigures, eight candidate solutions $\boldsymbol{x}_t^{(1)}$ to $\boldsymbol{x}_t^{(8)}$ are shown, with $\boldsymbol{x}_t^{(7)}$ selected as the focal individual. The figure demonstrates how the estimation of $\hat{\boldsymbol{x}}_0$ and the generation of the new candidate $\boldsymbol{x}_{t-1}^{(i)}$ vary depending on the value of α_t. Note that in this illustrative example, the stochastic noise term $\boldsymbol{w}$ is omitted for simplicity.

Figure 1(a) depicts the case with a small α_t, where the Gaussian variance is large. This allows distant individuals to contribute substantially to the estimation of $\hat{\boldsymbol{x}}_0$, resulting in a target that aggregates information from the broader population. As a result, the new candidate $\boldsymbol{x}_{t-1}^{(i)}$ tends to move toward a region reflecting global trends, promoting exploration. Figure 1(b) shows the case with a large α_t, where the Gaussian variance is small. Here, the estimation of $\hat{\boldsymbol{x}}_0$ is predominantly influenced by nearby individuals, while the



(a) Small α_t (b) Large α_t

FIGURE 1. Generation of $\boldsymbol{x}_{t-1}^{(i)}$ from $\boldsymbol{x}_t^{(i)}$ via estimated target $\hat{\boldsymbol{x}}_0$

influence of distant individuals becomes negligible. As a result, the new candidate $\boldsymbol{x}_{t-1}^{(i)}$ tends to stay close to the focal individual $\boldsymbol{x}_t^{(i)}$, leading to smaller incremental improvements. This behavior facilitates local exploitation, refining solutions based on information from the immediate neighborhood. Such a mechanism is particularly advantageous in multimodal landscapes, as it enables different individuals to independently explore separate valleys (local basins of attraction) during the exploitation phase, thereby increasing the likelihood of locating multiple optima.

3. **Incorporating Selection Mechanisms into Diffusion Evolution.** While the standard Diffusion Evolution algorithm shows promising results, it lacks an explicit selection mechanism – a fundamental component of traditional EAs. In Algorithm 1, all individuals unconditionally progress to the next generation without fitness-based selection. Our research aims to address this limitation by integrating established selection mechanisms into the Diffusion Evolution framework.

3.1. **Selection mechanisms in evolutionary computation.** Selection mechanisms in evolutionary computation create selective pressure that drives populations toward higher fitness regions. We investigate three widely used selection paradigms and their potential integration with Diffusion Evolution:

- **One-to-One Replacement** (from Differential Evolution [9]): Compares each offspring with its corresponding parent, retaining only the better individual. This maintains diversity while guaranteeing that solution quality does not deteriorate.
- **Tournament Selection** (common in Genetic Algorithms [10]): Randomly selects small subsets of individuals and propagates only the best from each group. Tournament size controls selection pressure – larger tournaments increase pressure toward higher fitness individuals.
- **Survival Selection** (exemplified by Evolution Strategies' $(\mu + \lambda)$ approach [11]): Combines parents and offspring into a single pool, selecting the best individuals for the next generation. This elitist approach ensures rapid convergence but may reduce diversity.

By integrating these mechanisms into Diffusion Evolution, we create enhanced variants that benefit from both the mathematical foundation of diffusion models and the ability to focus the search on better solutions, similar to traditional EAs.

3.2. **Proposed selection-enhanced Diffusion Evolution variants.**

3.2.1. *Diffusion Evolution with One-to-One Replacement (OneToOne).* Compared with Algorithm 1, the OneToOne variant keeps Lines 1-7 unchanged and modifies Line 8 as follows. After calculating the potential offspring $\boldsymbol{y}_{t-1}^{(i)}$ using the standard diffusion process (as defined in Line 8 of Algorithm 1), the algorithm compares its fitness with that of the parent $\boldsymbol{x}_t^{(i)}$. Only if the offspring has better fitness is it accepted as $\boldsymbol{x}_{t-1}^{(i)}$; otherwise, the parent is carried forward. This modification introduces an explicit parent-offspring replacement step, ensuring monotonic improvement while maintaining the stochastic exploration of diffusion-based updates.

$$\boldsymbol{y}_{t-1}^{(i)} \leftarrow \sqrt{\alpha_{t-1}}\,\hat{\boldsymbol{x}}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \left(\frac{\boldsymbol{x}_t^{(i)} - \sqrt{\alpha_t}\,\hat{\boldsymbol{x}}_0}{\sqrt{1-\alpha_t}} \right) + \sigma_t\,\boldsymbol{w} \tag{2}$$

Then, we apply the following selection rule:

$$\boldsymbol{x}_{t-1}^{(i)} = \begin{cases} \boldsymbol{y}_{t-1}^{(i)} & \text{if } f\left(\boldsymbol{y}_{t-1}^{(i)}\right) < f\left(\boldsymbol{x}_t^{(i)}\right) \\ \boldsymbol{x}_t^{(i)} & \text{otherwise} \end{cases} \tag{3}$$

This rule retains the offspring only if it improves upon the parent's fitness, ensuring monotonic improvement while preserving the exploratory nature of diffusion-based updates.

3.2.2. *Diffusion Evolution with Tournament Selection (Tournament).* Compared with Algorithm 1, the Tournament variant keeps Lines 1-8 unchanged and adds a tournament-based selection step after generating the entire offspring population. At step t, each parent individual $\boldsymbol{x}_t^{(i)}$ and its offspring $\boldsymbol{y}_t^{(i)}$ form a combined pool. For each position in the new population at step $t-1$, a tournament of size k is conducted by randomly selecting k individuals from this combined pool. The individual with the best fitness among the selected candidates is chosen as the winner and assigned to the corresponding position in the next generation. This addition introduces explicit selection pressure, controlled by the tournament size parameter k, while preserving diversity. Because tournaments for each position are conducted independently, high-fitness individuals are more likely to win multiple tournaments and be copied into several positions in the next generation.

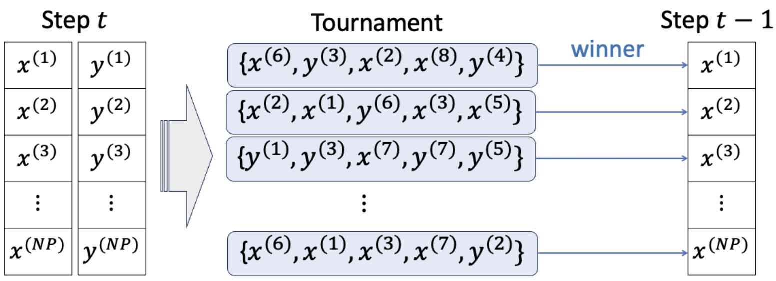


FIGURE 2. Tournament Selection process in Diffusion Evolution

3.2.3. *Diffusion Evolution with Survival Selection (Survival).* Compared with Algorithm 1, the Survival variant keeps Lines 1-8 unchanged and adds a Survival Selection step after generating the entire offspring population. This variant implements a $(\mu+\lambda)$-style selection mechanism, where μ corresponds to the parent population size and λ denotes the number of offspring generated, which is set equal to μ. After generating λ offspring using the diffusion-based update rule, the parent and offspring populations (totaling 2μ individuals) are combined. The top μ individuals with the best fitness are then selected to form the next generation. This modification introduces strong selection pressure by consistently retaining the highest-fitness individuals, ensuring rapid convergence while reducing population diversity compared with the other variants.

4. Experiment.

4.1. Setup.
To evaluate the performance of our proposed selection-enhanced Diffusion Evolution algorithms, we conducted extensive experiments on a set of standard test functions. In this study, we used four test functions commonly employed in evolutionary computation, each representing a distinct optimization challenge, as shown in Table 1. These functions were evaluated in three different dimensions ($D = 5$, $D = 10$, and $D = 30$) to analyze algorithm performance across varying problem complexities.

TABLE 1. Test functions used in the experimental evaluation

Function	Definition	Domain	Modality
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	$-5.12 \le x_i \le 5.12$	Unimodal
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$-2.048 \le x_i \le 2.048$	Unimodal
Rastrigin	$f(\mathbf{x}) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$-5.12 \le x_i \le 5.12$	Multimodal
Ackley	$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$-32.768 \le x_i \le 32.768$	Multimodal

We compared the four variants of the Diffusion Evolution algorithm described in Section 3: the standard Diffusion Evolution (**Standard**), Diffusion Evolution with One-to-One Replacement (**OneToOne**), Diffusion Evolution with Tournament Selection (**Tournament**, tournament size $k = 10$), and Diffusion Evolution with Survival Selection (**Survival**). All algorithms were evaluated using the following common parameters: population size $NP = 512$. The diffusion schedule α_t follows the cosine schedule proposed in [5] and is defined as

$$\alpha_t = 0.5 \times \cos\left(\frac{\pi t}{T}\right) + 0.5 \tag{4}$$

where t is the current reverse diffusion step, decreasing from T to 2, and T is the total number of evolution steps. This schedule ensures that exploration is emphasized in early stages, while exploitation is enhanced in later stages. The noise schedule σ_t is dynamically determined at each step according to the following formula:

$$\sigma_t = \sigma_m \sqrt{\frac{1 - \alpha_{t-1}}{1 - \alpha_t}} \sqrt{1 - \frac{\alpha_t}{\alpha_{t-1}}} \tag{5}$$

where σ_m is a noise magnitude control parameter. The cosine schedule was selected based on previous research showing its effectiveness with fewer iterations [12]. In our implementation, we set $\sigma_m = 1.0$ by default, consistent with the original Diffusion Evolution formulation [5].

The maximum number of generations (steps) was set according to the dimensionality: 100, 500, and 1000 generations for $D = 5$, $D = 10$, and $D = 30$, respectively. For statistical validity, all experiments were conducted with 20 independent runs, with results reported as means and standard deviations.

4.2. **Results.** Table 2 presents the mean and standard deviation of the fitness values obtained by each algorithm at the final generation for each dimensional setting, averaged over 20 independent runs.

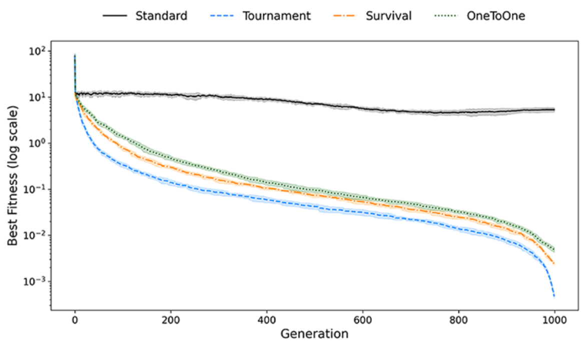
TABLE 2. Comparison of algorithm performance (mean ± std)

Algorithm	$D = 5$	$D = 10$	$D = 30$
Sphere function			
Standard	7.69E-03 ± 4.21E-03	2.36E-01 ± 5.13E-02	5.32E+00 ± 5.91E-01
Tournament	5.37E-05 ± 2.02E-05	**2.65E-05 ± 9.11E-06**	**4.71E-04 ± 3.80E-05**
Survival	**2.63E-05 ± 9.33E-06**	5.88E-05 ± 1.65E-05	2.49E-03 ± 1.74E-04
OneToOne	1.05E-04 ± 4.69E-05	1.14E-04 ± 3.93E-05	4.89E-03 ± 5.93E-04
Rosenbrock function			
Algorithm	$D = 5$	$D = 10$	$D = 30$
Standard	1.83E+00 ± 4.63E-01	1.64E+02 ± 5.34E+01	5.56E+03 ± 1.17E+03
Tournament	**1.63E-01 ± 1.17E-01**	3.03E+00 ± 1.36E+00	2.64E+01 ± 1.90E+00
Survival	3.30E-01 ± 1.40E-01	2.83E+00 ± 4.59E-01	2.49E+01 ± 8.71E-01
OneToOne	2.85E-01 ± 1.43E-01	**2.10E+00 ± 1.00E+00**	**2.45E+01 ± 9.09E-01**
Ackley function			
Algorithm	$D = 5$	$D = 10$	$D = 30$
Standard	3.49E-02 ± 9.60E-03	5.04E-01 ± 1.36E-01	4.46E+00 ± 1.39E-01
Tournament	1.42E-02 ± 3.40E-03	**6.29E-03 ± 9.36E-04**	**1.65E-02 ± 1.24E-03**
Survival	**7.78E-03 ± 1.94E-03**	9.93E-03 ± 1.51E-03	4.09E-02 ± 1.98E-03
OneToOne	1.86E-02 ± 5.97E-03	1.66E-02 ± 2.57E-03	7.69E-02 ± 5.40E-03
Rastrigin function			
Algorithm	$D = 5$	$D = 10$	$D = 30$
Standard	1.70E-02 ± 7.75E-03	**2.74E-01 ± 3.15E-01**	2.39E+02 ± 1.07E+01
Tournament	2.15E+00 ± 1.58E+00	5.68E+00 ± 2.40E+00	2.34E+01 ± 6.46E+00
Survival	**1.58E-03 ± 4.53E-04**	1.35E+00 ± 5.70E-01	**1.07E+01 ± 1.88E+00**
OneToOne	2.26E-01 ± 2.26E-01	1.68E+00 ± 4.73E-01	1.47E+01 ± 1.75E+00

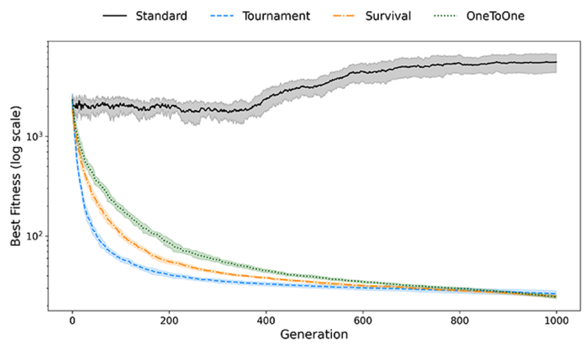
Based on the experimental results, Diffusion Evolution algorithms incorporating selection mechanisms generally exhibit improved performance compared to the standard variant. In particular, for the Sphere function, the Survival variant achieves a notable improvement in the 5-dimensional setting, while the Tournament variant yields even greater improvements in 10 and 30 dimensions. For the Rosenbrock function, Tournament achieves the best performance in the low-dimensional case ($D = 5$), although the differences among the selection-enhanced variants remain relatively small. As the dimensionality increases ($D = 10$ and $D = 30$), the OneToOne variant produces the best final fitness values. However, across all high-dimensional settings, the final fitness values remain relatively large, indicating that none of the methods successfully converged to solutions near the global optimum.

For multimodal functions, different patterns were observed. In the case of the weakly multimodal Ackley function, the algorithms exhibited similar behavior to that observed on the unimodal Sphere function, with Tournament Selection outperforming the others. For the more rugged Rastrigin function, Tournament exhibited lower performance, likely due to premature convergence caused by its strong selection pressure. In particular, it performed worse than the Standard variant in the low-dimensional cases ($D = 5$ and $D = 10$). Instead, in higher dimensions, the Survival variant achieved the best performance.

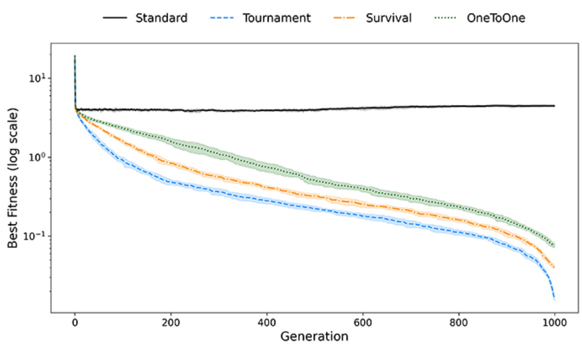
Figure 3 traces the best-fitness trajectories of the four algorithms over 1000 generations on the 30-dimensional test set, each curve representing the mean of 20 runs. It should be noted that while the generation axis in the performance plots starts from 0 (representing the beginning of the evolution), the reverse diffusion step index t used within the algorithm itself decreases from T to 2. On the unimodal Sphere problem, Tournament exhibits both the fastest initial descent and the lowest final fitness, demonstrating that strong selection pressure is advantageous when a single global optimum dominates the landscape. Survival shows the second fastest convergence, followed by OneToOne, while the Standard variant exhibits minimal improvement throughout the run. On the Rosenbrock function, the initial convergence speed follows a similar trend to the Sphere function: Tournament achieves the fastest early descent, followed by Survival and OneToOne. However, in the later stages, as the population spreads along the Rosenbrock valley's curved slope, the convergence rate of all three selection-enhanced methods (Tournament, Survival, and OneToOne) slows and becomes comparable. This behavior reflects the difficulty of navigating the narrow, curved valley toward the global optimum. Meanwhile, the Standard variant exhibits worsening fitness values in the later stages, which is likely due to individuals drifting away from the valley floor and into less promising regions of the search space.



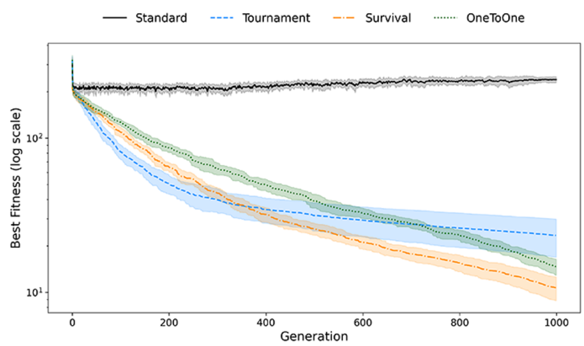
(a) Sphere function



(b) Rosenbrock function



(c) Ackley function



(d) Rastrigin function

FIGURE 3. Comparison of best fitness values on test functions ($D = 30$)

On the Ackley function, all algorithms show similar trends to the Sphere function. In contrast, on the Rastrigin function, Tournament shows rapid initial improvement, but the search efficiency declines in the later stages. This is likely because, as the search progresses and the population encounters the multimodal landscape, the strong selection pressure causes individuals to become trapped in local basins of attraction. Meanwhile, OneToOne achieves good final fitness by effectively preserving diversity and maintaining steady improvement throughout the run, as the scope of Survival competition is limited to each parent-offspring pair. Survival achieves the best performance, balancing selection pressure and diversity retention.

5. **Conclusion.** In this study, we demonstrated that integrating selection mechanisms into the Diffusion Evolution framework improves optimization performance across both unimodal and multimodal problems. Each selection strategy exhibited distinct traits, offering different advantages depending on the landscape's features. The OneToOne variant maintained steady, though relatively slow, evolutionary progress. Because Survival Selection was limited to parent-offspring pairs, the population avoided rapid diversity loss and premature convergence, thereby enabling stable exploration over time. In contrast, the Tournament variant emphasized fast convergence by allowing top-performing individuals to proliferate throughout the population. However, with the Tournament size adopted in this study, its strong selection pressure sometimes caused premature convergence, particularly in rugged, multimodal landscapes. The Survival Selection strategy demonstrated an intermediate convergence speed between OneToOne and Tournament. This balance of moderate selection pressure and diversity retention led to consistent performance across various problem landscapes.

Based on these observations, the choice of selection mechanism should align with the problem's modality. In particular, in evolutionary computation, several methods have been proposed to estimate the modality of the problem landscape based on the distribution of the population during the search [13]. Therefore, selecting an appropriate survival selection method informed by such estimations could represent an effective and adaptive strategy. Building upon this perspective, future research will explore adaptive selection strategies that dynamically adjust selection pressure during the search process and will investigate the application of these methods to more complex, real-world optimization problems.

REFERENCES

[1] J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, *Advances in Neural Information Processing Systems (NeurIPS)*, vol.33, pp.6840-6851, 2020.

[2] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, *International Conference on Machine Learning*, pp.2256-2265, 2015.

[3] Y. Song and S. Ermon, Generative modeling by estimating gradients of the data distribution, *Advances in Neural Information Processing Systems*, vol.32, 2019.

[4] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui and M.-H. Yang, Diffusion models: A comprehensive survey of methods and applications, *ACM Computing Surveys*, vol.56, no.4, pp.1-39, 2023.

[5] Y. Zhang, B. Hartl, H. Hazan and M. Levin, Diffusion models are evolutionary algorithms, *The 13th International Conference on Learning Representations*, 2025.

[6] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2015.

[7] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2010.

[8] D. Wang, D. Tan and L. Liu, Particle swarm optimization algorithm: An overview, *Soft Computing*, vol.22, no.2, pp.387-408, 2018.

[9] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.

[10] J. Zhong, X. Hu, J. Zhang and M. Gu, Comparison of performance between different selection strategies on simple genetic algorithms, *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol.2, pp.1115-1121, 2005.

[11] H. G. Beyer and H. P. Schwefel, Evolution strategies – A comprehensive introduction, *Natural Computing*, vol.1, pp.3-52, 2002.

[12] A. Q. Nichol and P. Dhariwal, Improved denoising diffusion probabilistic models, *Proceedings of the 38th International Conference on Machine Learning*, vol.139, pp.8162-8171, 2021.
[13] T. Nishio, J. Kushida, A. Hara and T. Takahama, Particle swarm optimisation with dynamic search strategies based on rank correlation, *International Journal of Computational Intelligence Studies*, vol.6, no.4, pp.311-332, 2017.

Author Biography

Jun-ichi Kushida obtained his Ph.D. degree in Engineering from Ritsumeikan University, Japan, in 2009. From 2012 to 2021, he served at Hiroshima City University, Japan, as an Assistant Professor, Lecturer, and subsequently as an Associate Professor in the Faculty of Information Sciences. He has been a Professor at Nihon Fukushi University, Japan, since 2021, belonging to the Department of Health Sciences until 2024, and to the Department of Engineering from 2025.

Dr. Kushida has published over 60 papers in academic journals and conference proceedings. His research interests include soft computing, evolutionary computation, and optimization algorithms, with a focus on constrained and multimodal optimization.