ENHANCING EVOLUTIONARY CONTROLLERS WITH A REJECTION-BASED GENETIC ALGORITHM

Ali Deeb
Higher School of Cyberphysical
Systems and Control
Peter the Great St. Petersburg
Polytechnic University
Saint Petersburg, Russian Federation
dib_a@spbstu.ru

Vladimir Khokhlovskiy
Higher School of Cyberphysical
Systems and Control
Peter the Great St. Petersburg
Polytechnic University
Saint Petersburg, Russian Federation
hohlovskij_vn@spbstu.ru

Viacheslav Shkodyrev
Higher School of Cyberphysical
Systems and Control
Peter the Great St. Petersburg
Polytechnic University
Saint Petersburg, Russian Federation
shkodyrev@spbstu.ru

Article history:
Received 05.10.2025, Accepted 20.12.2025

Abstract

Genetic Algorithms (GAs) are promising for receding-horizon control in nonlinear, constrained systems, but their wall-clock cost is dominated by objective evaluations. We propose a rejection-based GA tailored to horizon-structured genomes that performs exactly one plant-evaluation batch per generation and filters unevaluated offspring through a lightweight, calibrated classifier. On a standard nonisothermal Continuous Stirred Tank Reactor (CSTR) benchmark, we compare four optimizers under identical settings: a baseline GA, our committee-gated RB-GA, the baseline with Simple Variable Population Sizing (SVPS), and the recently proposed Two-Level Adaptive Simulated Binary Crossover (TLASBX). RB-GA consistently improves tracking accuracy over the baseline at lower generation budgets. Baseline+SVPS and TLASBX closely match the baseline. We attribute this outcome to the limited number of generations, which constrains their learning effects. Overall, the results indicate that the proposed mechanism enhances optimization convergence in expensive-evaluation regimes.

Key words

genetic algorithms, model predictive control, rejection-based evaluation, process control, AI-driven control systems.

1 Introduction

Recent advances in control and optimization drove Genetic Algorithms (GAs) [Deb, 1999] back to the surface, fortified by their capabilities to optimize throughout complex application scenarios, where non-convex objectives, hard constraints, hybrid dynamics, and multiobjective trade-offs are routine.

In diverse engineering applications, embedding GA-search into a receding horizon framework has delivered consistent improvements in control and optimization compared to baseline (linearized) MPC and GA-only strategies in microgrid energy management [Cavus and Allahham, 2024], chemical reactors [Deeb et al., 2024b], PID control designs [Ramos et al., 2024, Khodadoost et al., 2024], load frequency control under nonlinearities [Saini and Ohri, 2023], and robotics [Deeb et al., 2024a, Deeb et al., 2025b, Liu et al., 2024a, Duan et al., 2024, Wu et al., 2025].

This paper develops a rejection-based genetic algorithm tailored to horizon-structured decision vectors for receding-horizon control. The proposed method uses a lightweight learned filter to prioritize expensive objective evaluations while retaining deterministic elite carry-over and diversity maintenance. We demonstrate that, under identical plant models, objectives, and bounds, the filter preserves solution quality and clarifies when its overhead is justified-namely, in costly-evaluation regimes. From a broader perspective, the proposed methodology aligns with the principles of cybernetical neuroscience, which studies learning, adaptation, and control in complex systems using cybernetic and computational methods [Babich et al., 2025]. In this sense, rejection-based evolutionary control can be viewed as a lightweight mechanism for regulating computational effort in adaptive decision-making loops.

The remainder of this paper is organized as follows. Section 2 reviews related work on surrogate-assisted and learning-enhanced genetic algorithms. Section 3 formulates the optimization problem, details the proposed rejection-based genetic algorithm, and describes its extension to multi-objective optimization and receding-horizon control. Section 4 presents the experimental evaluation on standard benchmark functions and the nonlinear CSTR control case study. Section 5 concludes the paper. Finally, Section 6 discusses the main limitations and outlines future research directions.

2 Related Work

Research in evolutionary computation has increasingly incorporated machine learning to enhance search efficiency and reduce evaluation cost. Two primary avenues have emerged: (i) *learning to adapt variation operators*, and (ii) *learning to approximate fitness evaluation*. These approaches collectively aim to optimize resource allocation during search, and drive a better convergence towards the solution.

On the operator side, adaptive operator selection (AOS) leverages online learning to dynamically choose among variation operators based on performance feedback and problem-specific landscape features [Fialho, 2010, Consoli et al., 2016, Geng et al., 2023]. Policy-based methods, including tree-based and reinforcement learning controllers, extend this adaptivity to more complex decision spaces.

Simultaneously, surrogate-assisted evolutionary algorithms (SAEAs) have emerged as a means to reduce computational cost via predictive modeling. Recent surveys provide comprehensive taxonomies of surrogate types (e.g., regression, classification), infill criteria, and model management strategies in black-box optimization [He et al., 2023]. These surrogates serve as stand-ins for expensive fitness evaluations, supporting faster convergence with fewer objective function calls.

Within SAEAs, classification models are increasingly used to prioritize candidate solutions before evaluation. Techniques include ensemble-based surrogates [Wang et al., 2018] and integrated predictors such as Random Forests in deep evolutionary architectures [Sun et al., 2019]. In many-objective and constrained settings, classifiers estimate dominance or feasibility to filter infeasible or suboptimal offspring [Pan et al., 2018, Li et al., 2022], while active learning and committee models inform sample selection [Wang et al., 2017].

Various learners-including k-NN and gradient-boosted trees-have demonstrated strong ranking performance in constrained evaluation environments [Jiang et al., 2024, Mao et al., 2024]. Hybrid metaheuristics, such as GA-PSO variants, further enhance performance by blending global and local search behaviors [Aivaliotis-Apostolopoulos and Loukidis, 2022, Ali and Tawhid, 2017]. The overarching objective remains consistent: to predict and allocate evaluation effort judiciously.

Structural innovations within the GA family provide a robust foundation for integrating learning-based components. The Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) and its variants—e.g., clustered (CGOMEA) and real-valued (RV-GOMEA) forms—learn linkage information online, offering scalability and theoretical guarantees [Dushatskiy et al., 2024, Andreadis et al., 2024, Qiao and Gallagher, 2024].

Learned variation operators also continue to advance the field. For instance, Deep Neural Crossover (DNC) has demonstrated superior performance over traditional crossover operators in combinatorial domains [Shem-Tov and Elyasaf, 2024], while dueling-DQN agents guide operator selection [Yin and Xiang, 2024]. Classical operators like SBX have also been enhanced via Bayesian hyperparameter tuning [Deeb et al., 2025a]. Theoretical analyses further support these mechanisms, showing provable gains in scenarios such as $(\mu+1)$-GAs on JUMP functions [Opris et al., 2025, Doerr et al., 2024] and EMO tasks requiring crossover for efficient convergence [Dang et al., 2024, Opris, 2025].

At the system level, recent developments aim to reduce evaluation overhead and stabilize evolutionary progress. Techniques such as modern niching scale effectively to high-dimensional multimodal landscapes [Matanga et al., 2024], while scheduled population control manages evaluation cost without degrading solution quality [Jiménez Laredo et al., 2024]. Constraint-handling strategies like Boundary Updating have shown to expedite feasibility discovery with minimal bias [Rahimi et al., 2024].

Comprehensive reviews have consolidated best practices for SAEA in both continuous and combinatorial domains [Liang et al., 2024, Liu et al., 2024b]. Notably, large language models (LLMs) are now being integrated into multiobjective SAEA frameworks to further reduce evaluation requirements [Wang et al., 2024]. Automated and parallelized EA systems also support deployment at scale by minimizing human-in-the-loop tuning [Al-Terkawi and Migliavacca, 2025].

Despite this breadth of progress, one strategy remains notably underexplored: the use of a *calibrated* machine learning classifier as a *hard rejection mechanism* during generation assembly. Existing work typically employs classification for ranking, prioritization, or survival prediction [He et al., 2023, Pan et al., 2018, Chen et al., 2021]. In contrast, the proposed method enforces a strict accept/reject gate—applied prior to evaluation—by filtering offspring through a committee-based classifier calibrated to manage false acceptance rates. This approach reconceptualizes classification from a soft heuristic into a formal gatekeeping mechanism, enabling evaluation-aware control without compromising genetic diversity or convergence behavior.

3 Problem Formulation

Consider the continuous optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{s.t.} \quad \ell_i \le x_i \le u_i, \;\; i = 1, \ldots, n, \qquad (1)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective and $[\ell_i, u_i]$ are box constraints for an input vector $\mathbf{x} = [x_1, x_2, \cdots, x_n]$. A population-based search maintains N candidate solutions $\mathcal{P}^{(g)} = \{\mathbf{x}_1^{(g)}, \ldots, \mathbf{x}_N^{(g)}\}$ at generation g. At every generation, each individual $x_j^{(g)}$ is scored by its objective value $f(\mathbf{x}_j^{(g)})$. Afterwise, parents are chosen by binary tournament: sample two individuals uniformly without replacement, and select the one with lower objective value. Repeating this N times yields a mating pool of parent pairs:

$$\mathcal{M}^{(g)} = \{(\mathbf{x}_{p_1}^{(g)}, \mathbf{x}_{p_2}^{(g)})\}_{k=1}^{N/2}. \tag{2}$$

Using Simulated Binary Crossover (SBX), for each parent pair $(\mathbf{x}_{p_1}^{(g)}, \mathbf{x}_{p_2}^{(g)})$ and for each decision variable $i \in \{1, \dots, n\}$, with crossover probability p_c, we generate two children $c_i^{(1)}, c_i^{(2)}$ as:

$$\begin{aligned}
x_i^L &= \min\{x_{p_1,i}^{(g)},\, x_{p_2,i}^{(g)}\}, \\
x_i^U &= \max\{x_{p_1,i}^{(g)},\, x_{p_2,i}^{(g)}\}, \\
\beta_q &= \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & u \le \frac{1}{2}, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & u > \frac{1}{2}, \end{cases} \\
& u \sim \mathcal{U}(0,1),\ \eta_c > 0, \\
c_i^{(1)} &= \tfrac{1}{2}\Big[(1+\beta_q)\, x_{p_1,i}^{(g)} + (1-\beta_q)\, x_{p_2,i}^{(g)}\Big], \\
c_i^{(2)} &= \tfrac{1}{2}\Big[(1-\beta_q)\, x_{p_1,i}^{(g)} + (1+\beta_q)\, x_{p_2,i}^{(g)}\Big].
\end{aligned} \tag{3}$$

In case crossover is not applied (with probability $1 - p_c$), the children inherit $x_{p_1,i}^{(g)}$ and $x_{p_2,i}^{(g)}$ unchanged. SBX is symmetric in the parents and preserves the parental mean. Bound feasibility is enforced by simple repair, for example:

$$c_i^{(m)} \leftarrow \min\big\{u_i,\ \max\{\ell_i,\, c_i^{(m)}\}\big\}, \quad m \in \{1,2\}. \tag{4}$$

For mutation, each child undergoes independent polynomial mutation per variable with probability p_m:

$$\begin{aligned}
\delta_i^- &= \frac{x_i - \ell_i}{u_i - \ell_i}, \qquad \delta_i^+ = \frac{u_i - x_i}{u_i - \ell_i}, \\
& r \sim \mathcal{U}(0,1),\ \eta_m > 0,
\end{aligned} \tag{5}$$

with

$$\Delta_i = \begin{cases} \Delta_{i,1}, & r < \frac{1}{2}, \\ \Delta_{i,2}, & r \ge \frac{1}{2}, \end{cases} \tag{6}$$

where

$$\begin{aligned}
\Delta_{i,1} &= \big[\, 2r + (1-2r)(1-\delta_i^-)^{\eta_m+1} \big]^{\frac{1}{\eta_m+1}} - 1, \\
\Delta_{i,2} &= 1 - \big[\, 2(1-r) + 2(r-\tfrac{1}{2})(1-\delta_i^+)^{\eta_m+1} \big]^{\frac{1}{\eta_m+1}}
\end{aligned} \tag{7}$$

then x_i is updated as $x_i \leftarrow x_i + \Delta_i\,(u_i - \ell_i)$, and followed by the same bound repair. This operator adapts the mutation step to the available interval and is controlled by the mutation distribution index η_m.

The last step is replacement and termination, where the next generation $\mathcal{P}^{(g+1)}$ is formed by the offspring, optionally using $(\mu{+}\lambda)$ elitism to preserve the best μ individuals:

$$\mathcal{P}^{(g+1)} \leftarrow \mathrm{Best}_N\big(\mathcal{P}^{(g)} \cup \mathrm{Offspring}^{(g)}\big). \tag{8}$$

The algorithm iterates evaluation, selection, SBX, and mutation until a stopping criterion holds.

3.1 Rejection-based gate between reproduction and evaluation

We place a surrogate gate after reproduction (selection+SBX+mutation) and before any new plant evaluations. The current population $\mathcal{P}^{(g)}$ is the only set evaluated at generation g. Offspring $\mathcal{O}^{(g)}$ are screened by the gate to decide admission into the next population, and are evaluated only at generation $g{+}1$.

After evaluating $\mathcal{P}^{(g)}$, assign within-generation labels by a quantile rule with parameter $\beta \in (0,1)$:

$$y_i = \mathbf{1}\big\{f(\mathbf{x}_i) \le \hat{q}_\beta\big(\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{P}^{(g)}}\big)\big\}\,, \qquad \mathbf{x}_i \in \mathcal{P}^{(g)}. \tag{9}$$

Append $(\mathbf{x}_i, y_i)$ to a replay buffer $\mathcal{D}$ and train a calibrated classifier $h : \mathbb{R}^n \to [0,1]$ on the most recent window $\mathcal{D}_{\mathrm{rec}}$ to predict

$$\hat{p}(\mathbf{x}) = \mathbb{P}(y{=}1 \mid \mathbf{x}). \tag{10}$$

For thresholding and committee rule, we fix a target false-reject rate $\epsilon_{\mathrm{fr}} \in (0,1)$, then compute per-model conformal thresholds on recent positives:

$$\begin{aligned}
\tau_m &= \hat{q}_{\epsilon_{\mathrm{fr}}}\Big(\big\{\hat{p}^{(m)}(\mathbf{x}_i) : y_i{=}1,\ (\mathbf{x}_i, y_i) \in \mathcal{D}_{\mathrm{rec}}\big\}\Big)\,, \\
\tau &= \min(\tau_1, \tau_2).
\end{aligned} \tag{11}$$

Given an offspring $\mathbf{x}' \in \mathcal{O}^{(g)}$, accept iff

$$\begin{aligned}
\min\{\hat{p}^{(1)}(\mathbf{x}'), \hat{p}^{(2)}(\mathbf{x}')\} \ge \tau, \wedge \\
\big|\hat{p}^{(1)}(\mathbf{x}') - \hat{p}^{(2)}(\mathbf{x}')\big| \le \delta,
\end{aligned} \tag{12}$$

where $\delta \in [0,1]$ limits committee disagreement. Let $\mathcal{A}^{(g)}$ be the accepted offspring.

Finally, we assemble the next population without evaluating offspring in g:

$$\begin{aligned}
\mathcal{P}^{(g+1)} = & \underbrace{\mathrm{top}_\eta(\mathcal{P}^{(g)})}_{\text{elite}} \ \cup \ \underbrace{\mathcal{A}^{(g)}}_{\text{admitted offspring}} \\
& \cup \ \underbrace{\mathrm{backfill}\Big(N - |\mathrm{top}_\eta| - |\mathcal{A}^{(g)}|\Big)}_{\text{diversity}}.
\end{aligned} \tag{13}$$

If the acceptance ratio $\rho^{(g)} = \frac{|\mathcal{A}^{(g)}|}{|\mathcal{O}^{(g)}|}$ falls below $\rho_{\min}$, backfill by highest mean confidence $\bar{p}(\mathbf{x}) = \frac{1}{2}\big(\hat{p}^{(1)}(\mathbf{x}) + \hat{p}^{(2)}(\mathbf{x})\big)$. This enforces three invariants per generation: one evaluation pass on $\mathcal{P}^{(g)}$; no evaluation of $\mathcal{O}^{(g)}$; surrogates trained only from evaluated individuals in $\mathcal{D}$.

Complexity (per generation): one plant batch of size N, $O(|\mathcal{O}^{(g)}|)$ surrogate queries, and infrequent recalibration every T_{clf} on $\mathcal{D}_{\mathrm{rec}}$.

Algorithm 1 Gate placement between reproduction and evaluation (generation g)

```
Require: Population $\mathcal{P}^{(g)}$ (size $N$); offspring size $M$;
    elite fraction $\eta$; gate params $(\beta, \epsilon_{\mathrm{fr}}, \rho_{\min}, \delta)$; window $W$
Evaluate $f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{P}^{(g)}$
Label via (9); update replay $\mathcal{D}$; set $\mathcal{D}_{\mathrm{rec}}$
if $g \bmod T_{\mathrm{clf}} = 0$ then
    Train/calibrate $h^{(1)}, h^{(2)}$ on $\mathcal{D}_{\mathrm{rec}}$
end if
Reproduce: build $\mathcal{O}^{(g)}$ by tournament, SBX, and polynomial mutation
Gate: compute $\hat{p}^{(1)}, \hat{p}^{(2)}$; accept $\mathcal{A}^{(g)} = \{\mathbf{x}' \in \mathcal{O}^{(g)} : \text{(12) holds}\}$
if $|\mathcal{A}^{(g)}|/M < \rho_{\min}$ then
    Backfill by descending $\bar{p}$ until quota
end if
Assemble $\mathcal{P}^{(g+1)}$ via (13); do not evaluate $\mathcal{O}^{(g)}$ in generation $g$
```

3.2 Extension to Multi-Objective Optimization

We generalize the single-objective tracking problem to the K-objective setting:

$$\begin{aligned} &\min_{\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n} \ \mathbf{F}(\mathbf{x}) = \big[f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})\big]^\top \\ &\text{s.t.} \quad \ell_i \le x_i \le u_i, \end{aligned} \tag{14}$$

with the standard Pareto dominance relation: $\mathbf{u} \prec \mathbf{v}$ iff $f_k(\mathbf{u}) \le f_k(\mathbf{v})$ for all k and $f_j(\mathbf{u}) < f_j(\mathbf{v})$ for at least one j. The optimal solutions form the (unknown) Pareto set $\mathcal{X}^\star$ and Pareto front $\mathcal{F}^\star = \{\mathbf{F}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}^\star\}$.

Within each generation, parents are chosen by NSGA-II criteria: fast non-dominated sorting. Resulting in front ranks $r(\cdot)$ and crowding distance $\mathrm{cd}(\cdot)$-that maintain spread. Tournaments minimize $(r, -\mathrm{cd})$ and elites are retained using the same ordering. Operators (SBX crossover and polynomial mutation) and box projection are unchanged.

The rejection-based gate still guarantees one plant evaluation per generation: offspring are filtered by a calibrated-conformal classifier trained on a replay buffer. For multi-objective data we define a scalar Pareto score

$$s(\mathbf{x}) = r(\mathbf{x}) - \varepsilon\, \widetilde{\mathrm{cd}}(\mathbf{x}), \quad \varepsilon > 0, \tag{15}$$

where r is the front rank and $\widetilde{\mathrm{cd}}$ is the per-front normalized crowding distance. Samples with low scores (better rank, larger crowding) are labeled positive using a percentile threshold, probabilities are calibrated, and a conformal threshold on the positive class controls false rejects. If the accepted set is too small, a soft top-K fallback (by mean calibrated probability) guarantees a minimum offspring ratio, while the remaining slots are filled without additional evaluations (mix of elites and random mutations).

3.3 Preceding-horizon Control with RB-GA

We address online horizon optimization for output tracking using a single–batch, rejection–based genetic algorithm (RBGA). The plant is treated as a black box that can be rolled out from the current state; all plant specifics are deferred to the experiments.

At control step k, given state $\mathbf{x}_k$ and a reference trajectory $r_{k+1:k+H}$, we choose a horizon-structured genome

$$\mathbf{g} \ \equiv \ u_{k:k+H-1} \in [u_{\min}, u_{\max}]^H, \tag{16}$$

and define the discounted, normalized tracking loss

$$\begin{aligned} J_k(\mathbf{g} \mid \mathbf{x}_k) &= \frac{\sum_{\tau=1}^{H} w_\tau \big(y_{k+\tau}(\mathbf{g}; \mathbf{x}_k) - r_{k+\tau}\big)^2}{\sum_{\tau=1}^{H} w_\tau}, \\ w_\tau &= \gamma^{\tau-1}, \ \ \gamma \in (0, 1], \end{aligned} \tag{17}$$

where $y_{k+\tau}(\mathbf{g}; \mathbf{x}_k)$ is the plant output at step $k+\tau$ when rolling out from $\mathbf{x}_k$ under $\mathbf{g}$. We solve

$$\min_{\mathbf{g} \in [u_{\min}, u_{\max}]^H} \ J_k(\mathbf{g} \mid \mathbf{x}_k), \tag{18}$$

applying only $u_k^\star$ then shifting the horizon.

4 Experiments and Evaluations

4.1 Benchmarking RBGA on Standard Test Functions- Single Objective Optimization

We compare five GA variants on standard continuous benchmarks at dimension D=30:

1. plain GA (baseline).
2. RBGA with a calibrated Decision Tree gate (RBGA-DT).
3. RBGA with a calibrated XGBoost gate (RBGA-XGB).
4. RBGA committee gate requiring DT and XGB agreement (RBGA-committee).
5. Shrinking viable population schedule, following the 2024 SVPS schedule [Jiménez Laredo et al., 2024].

Each run uses population 120 for 150 generations and performs exactly one expensive objective evaluation per generation. We run 20 repeats per method and function. Objectives are Sphere, Rastrigin, and Ackley with standard box bounds. Metrics per method: final best objective, AUC of the best-so-far curve over generations (lower is better) and mean wall time.

Results show that RBGA with calibrated RBGA-XGB gate delivers the best final values and AUC on all three functions tested. On Sphere it reaches 0.65 ± 0.16 with AUC $3.65{\times}10^3$, improving over RBGA-DT (0.91 ± 0.21) and far outperforming baseline (14.05 ± 6.23). On Rastrigin RBGA-XGB achieves 88.26 ± 16.50 and AUC $2.54{\times}10^4$, beating RBGA-committee (101.35 ± 14.73) and RBGA-DT (126.77 ± 18.27). On Ackley RBGA-XGB yields 5.25 ± 0.54 and AUC $1.89{\times}10^3$, ahead of

RBGA-DT $(5.50{\pm}0.71)$ and baseline $(9.40{\pm}1.73)$. The committee gate is competitive on Rastrigin but slower. Wall time reflects gating overhead: baseline/`svps` are sub-second per run, RBGA-DT/RBGA-XGB/RBGA-committee are slower (about 9.7, 14.0, and 20.6 s respectively). Table 1 lists means ± std across runs for the final objective, AUC, and time, and Figure 1 shows the mean and std so-far convergence during the run. Calibrated gating with RBGA-XGB reliably accelerates convergence in objective space at the cost of higher CPU time. The committee gate is conservative and slower but competitive on structured landscapes. Plain GA and SVPS are fastest but less effective under the same evaluation budget.

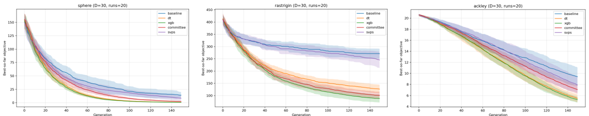

Figure 1. Convergence curves on $D = 30$ benchmarks. Mean best-so-far with $\pm 1\sigma$ band over 20 runs.

Table 1. RBGA variants on $D = 30$ benchmarks (20 runs, 150 generations, pop 120).

Function	Method	Final	AUC	Time [s]
Sphere	baseline	1.41×10^{1} $\pm\,6.23 \times 10^{0}$	6.47×10^{3} $\pm\,1.16 \times 10^{3}$	9.40×10^{-1} $\pm\,2.00 \times 10^{-2}$
	dt	9.13×10^{-1} $\pm\,2.05 \times 10^{-1}$	3.83×10^{3} $\pm\,4.52 \times 10^{2}$	9.69×10^{0} $\pm\,5.70 \times 10^{-1}$
	xgb	$\mathbf{6.45 \times 10^{-1}}$ $\mathbf{\pm\,1.56 \times 10^{-1}}$	$\mathbf{3.65 \times 10^{3}}$ $\mathbf{\pm\,3.79 \times 10^{2}}$	1.40×10^{1} $\pm\,4.26 \times 10^{0}$
	committee	2.34×10^{0} $\pm\,7.20 \times 10^{-1}$	4.78×10^{3} $\pm\,5.80 \times 10^{2}$	2.07×10^{1} $\pm\,8.10 \times 10^{-1}$
	svps	8.28×10^{0} $\pm\,5.22 \times 10^{0}$	5.74×10^{3} $\pm\,7.51 \times 10^{2}$	7.20×10^{-1} $\pm\,1.00 \times 10^{-2}$
Rastrigin	baseline	2.71×10^{2} $\pm\,2.05 \times 10^{1}$	4.47×10^{4} $\pm\,2.08 \times 10^{3}$	1.05×10^{0} $\pm\,1.00 \times 10^{-2}$
	dt	1.27×10^{2} $\pm\,1.83 \times 10^{1}$	2.95×10^{4} $\pm\,1.91 \times 10^{3}$	9.01×10^{0} $\pm\,4.40 \times 10^{-1}$
	xgb	$\mathbf{8.83 \times 10^{1}}$ $\mathbf{\pm\,1.65 \times 10^{1}}$	$\mathbf{2.54 \times 10^{4}}$ $\mathbf{\pm\,2.27 \times 10^{3}}$	1.28×10^{1} $\pm\,5.80 \times 10^{-1}$
	committee	1.01×10^{2} $\pm\,1.47 \times 10^{1}$	2.68×10^{4} $\pm\,1.89 \times 10^{3}$	2.07×10^{1} $\pm\,7.20 \times 10^{-1}$
	svps	2.46×10^{2} $\pm\,3.33 \times 10^{1}$	4.36×10^{4} $\pm\,2.12 \times 10^{3}$	7.70×10^{-1} $\pm\,1.00 \times 10^{-2}$
Ackley	baseline	9.40×10^{0} $\pm\,1.73 \times 10^{0}$	2.21×10^{3} $\pm\,1.64 \times 10^{2}$	1.10×10^{0} $\pm\,1.00 \times 10^{-2}$
	dt	5.50×10^{0} $\pm\,7.10 \times 10^{-1}$	1.91×10^{3} $\pm\,1.14 \times 10^{2}$	9.70×10^{0} $\pm\,3.90 \times 10^{-1}$
	xgb	$\mathbf{5.25 \times 10^{0}}$ $\mathbf{\pm\,5.40 \times 10^{-1}}$	$\mathbf{1.89 \times 10^{3}}$ $\mathbf{\pm\,9.60 \times 10^{1}}$	1.25×10^{1} $\pm\,4.50 \times 10^{-1}$
	committee	7.07×10^{0} $\pm\,8.00 \times 10^{-1}$	2.06×10^{3} $\pm\,7.40 \times 10^{1}$	2.09×10^{1} $\pm\,7.70 \times 10^{-1}$
	svps	7.91×10^{0} $\pm\,1.45 \times 10^{0}$	2.16×10^{3} $\pm\,1.65 \times 10^{2}$	8.40×10^{-1} $\pm\,1.00 \times 10^{-2}$

4.2 Multi-Objective Benchmark: ZDT1/ZDT2/ZDT3 with Pareto-Front Tracking

We evaluate RBGA on the ZDT suite ($K{=}2$ objectives; decision domain $\mathcal{X} = [0,1]^D$), comparing baseline GA, RBGA-DT, RBGA-XGB, RBGA-Committee, and SVPS. Per generation only the current population is evaluated; offspring are filtered by the calibrated–conformal gate (committee agreement when applicable) and the remainder of the next population is filled without additional evaluations, preserving the one-evaluation-per-generation invariant. For each generation we log the objective cloud, extract the non-dominated set, and compute IGD/GD (lower is better) and HV (higher is better; reference $\mathbf{r} = (1.2, 1.2)$). We show the per-step best convergence in Figures 2,3 and 4, and the final-generation metrics in Tables 2, 3, and 4.

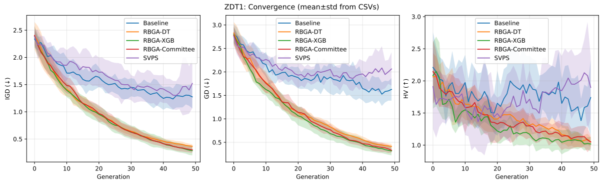

Figure 2. ZDT1: convergence from per-generation CSVs — (a) IGD, (b) GD, (c) HV.

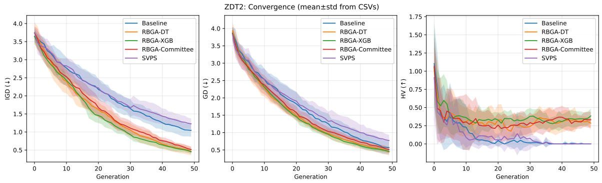

Figure 3. ZDT2: convergence from per-generation CSVs — (a) IGD, (b) GD, (c) HV.

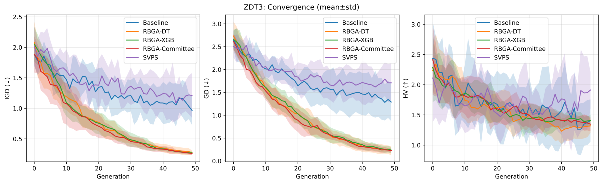

Figure 4. ZDT3: convergence from per-generation CSVs — (a) IGD, (b) GD (c) HV.

Table 2. ZDT1: final-generation metrics. Lower IGD/GD, higher HV are better.

method	IGD ↓	GD ↓	HV ↑
baseline	1.27×10^{0} $\pm 2.20 \times 10^{-1}$	1.63×10^{0} $\pm 2.50 \times 10^{-1}$	1.74×10^{0} $\pm 6.20 \times 10^{-1}$
dt	3.67×10^{-1} $\pm 3.80 \times 10^{-2}$	4.15×10^{-1} $\pm 5.40 \times 10^{-2}$	1.05×10^{0} $\pm 8.10 \times 10^{-2}$
xgb	2.86×10^{-1} $\pm 9.00 \times 10^{-2}$	3.16×10^{-1} $\pm 9.80 \times 10^{-2}$	1.02×10^{0} $\pm 1.20 \times 10^{-1}$
committee	3.03×10^{-1} $\pm 6.00 \times 10^{-2}$	3.32×10^{-1} $\pm 5.60 \times 10^{-2}$	1.05×10^{0} $\pm 1.10 \times 10^{-1}$
svps	1.52×10^{0} $\pm 4.80 \times 10^{-1}$	2.06×10^{0} $\pm 3.70 \times 10^{-1}$	1.90×10^{0} $\pm 5.10 \times 10^{-1}$

Table 3. ZDT2: final-generation metrics. Lower IGD/GD, higher HV are better.

method	IGD ↓	GD ↓	HV ↑
baseline	1.05×10^{0} $\pm 1.90 \times 10^{-1}$	5.69×10^{-1} $\pm 2.10 \times 10^{-1}$	0.00×10^{0} $\pm 0.00 \times 10^{0}$
dt	4.67×10^{-1} $\pm 1.20 \times 10^{-1}$	4.43×10^{-1} $\pm 1.00 \times 10^{-1}$	2.85×10^{-1} $\pm 9.50 \times 10^{-2}$
xgb	4.52×10^{-1} $\pm 6.80 \times 10^{-2}$	4.58×10^{-1} $\pm 5.40 \times 10^{-2}$	3.85×10^{-1} $\pm 1.10 \times 10^{-1}$
committee	5.17×10^{-1} $\pm 8.90 \times 10^{-2}$	5.12×10^{-1} $\pm 9.30 \times 10^{-2}$	3.33×10^{-1} $\pm 1.20 \times 10^{-1}$
svps	1.23×10^{0} $\pm 1.60 \times 10^{-1}$	7.72×10^{-1} $\pm 1.70 \times 10^{-1}$	0.00×10^{0} $\pm 0.00 \times 10^{0}$

Table 4. ZDT3: final-generation metrics. Lower IGD/GD, higher HV are better.

method	IGD ↓	GD ↓	HV ↑
baseline	9.66×10^{-1} $\pm 2.50 \times 10^{-1}$	1.29×10^{0} $\pm 4.40 \times 10^{-1}$	1.41×10^{0} $\pm 3.70 \times 10^{-1}$
dt	2.71×10^{-1} $\pm 3.90 \times 10^{-2}$	2.26×10^{-1} $\pm 1.00 \times 10^{-1}$	1.32×10^{0} $\pm 1.30 \times 10^{-1}$
xgb	2.70×10^{-1} $\pm 2.40 \times 10^{-2}$	2.38×10^{-1} $\pm 6.90 \times 10^{-2}$	1.41×10^{0} $\pm 9.70 \times 10^{-2}$
committee	2.62×10^{-1} $\pm 2.70 \times 10^{-2}$	2.26×10^{-1} $\pm 4.60 \times 10^{-2}$	1.35×10^{0} $\pm 1.20 \times 10^{-1}$
svps	1.21×10^{0} $\pm 3.90 \times 10^{-1}$	1.71×10^{0} $\pm 4.50 \times 10^{-1}$	1.91×10^{0} $\pm 3.20 \times 10^{-1}$

4.3 Nonlinear Control of CSTR system

We consider the standard nonisothermal CSTR [Berberich et al., 2022] with states x_1 (concentration) and x_2 (temperature), sampled with period $T_s > 0$. To avoid overloading the time index k, we denote the reaction-rate constant by $\kappa > 0$. The discrete-time dynamics are

$$x_{1,k+1} = x_{1,k} + \frac{T_s}{\theta}\big(1 - x_{1,k}\big) - T_s\,\kappa\,x_{1,k}\,e^{-\frac{M}{x_{2,k}}}, \quad (19)$$

$$\begin{aligned} x_{2,k+1} = x_{2,k} &+ \frac{T_s}{\theta}\big(x_f - x_{2,k}\big) \\ &+ T_s\,\kappa\,x_{1,k}\,e^{-\frac{M}{x_{2,k}}} \\ &- T_s\,\alpha\,u_k\,(x_{2,k} - x_c), \end{aligned} \quad (20)$$

where $\theta > 0$ is the residence-time constant, $M > 0$ the scalar activation parameter, x_f the feed temperature, x_c the coolant temperature, and $\alpha > 0$ the heat-transfer coefficient. We use the output $y_k := x_{1,k}$.

At each control step k, given the current state $\mathbf{x}_k = [x_{1,k},\, x_{2,k}]^\top$, we optimize an H-step input sequence $u_{k:k+H-1} := \{u_k, \dots, u_{k+H-1}\}$ to minimize a discounted, weighted MSE:

$$\begin{aligned} \min_{u_{k:k+H-1}} \; & J_k\big(u_{k:k+H-1} \mid \mathbf{x}_k\big) = \\ & \frac{\sum_{\tau=1}^{H} w_\tau \big(y_{k+\tau} - r_{k+\tau}\big)^2}{\sum_{\tau=1}^{H} w_\tau}, \end{aligned} \quad (21)$$

$$\text{s.t.} \qquad u_{\min} \;\le\; u_{k+\tau-1} \;\le\; u_{\max}, \quad \tau = 1, \dots, H,$$

with designer-chosen positive weights w_τ (we use $w_\tau = \gamma^{\tau-1}$ with $\gamma \in (0, 1]$) and box constraints $(u_{\min}, u_{\max})$. Only the first action $u_k^\star$ is applied; the horizon then shifts (apply-and-shift MPC). The decision vector (genome) is

$$\mathbf{g} \;\equiv\; u_{k:k+H-1} \;\in\; [u_{\min}, u_{\max}]^H. \quad (22)$$

Unless stated otherwise, we follow [Deeb et al., 2024b]. In our experiments we use $T_s = 0.2$, $\theta = 20$, $\kappa = 300$, $M = 5$, $x_f = 0.3947$, $x_c = 0.3816$, $\alpha = 0.117$, a concentration reference $y_{\text{ref}} = 0.6519$, horizon $H = 15$, and input bounds $(u_{\min}, u_{\max}) = (0.1,\, 2.0)$.

We solve (21) using a rejection-based GA tailored to horizon-structured genomes as explained in Algorithm 1.

We compare *four* optimizers that share exactly the same interface, objective (21), plant model (19)–(20), bounds, horizon H, and target sequence. Any performance difference thus stems from the search mechanism. Wee add to the methods used in this experiment, the recently proposed TLASBX-η; baseline GA using an SBX distribution index η learned on a source task by Bayesian Optimization [Deeb et al., 2025a].

Evaluation protocol.

1. Same receding-horizon loop of length $T = 750$ with $T_s = 0.2$; at each step we plan an $H = 15$-step sequence and apply the first input (apply-and-shift).
2. GA hyperparameters: population $P = 100$, extension factor $\lambda = 100$ for offspring proposals, identical seeds across methods.

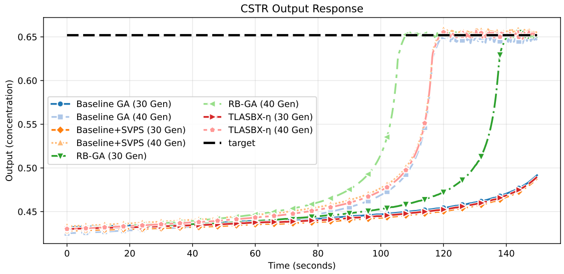

Figure 5. CSTR output overlays for all methods (physical time axis). Curves largely coincide for TLASBX-η, Baseline, and Baseline+SVPS; RB–GA reduces tracking error at the expense of run time and—at lower budgets—higher control effort.

Table 5. Baseline GA, RB–GA (Decision-Tree gate), Baseline+SVPS, and TLASBX-η under $T = 750$, $H = 15$, population $P = 100$, and extension factor $\lambda = 100$. Times are end-to-end wall clock for the full T-step run on the same machine.

Method	Gens	RMSE	MAE
Baseline GA	40	0.17745	0.15528
Baseline GA	30	0.20862	0.20822
RB-GA (DT gate)	40	**0.16782**	**0.13972**
RB-GA (DT gate)	30	0.19446	0.18500
Baseline + SVPS	40	0.17745	0.15528
Baseline + SVPS	30	0.20862	0.20822
TLASBX-η	40	0.17745	0.15528
TLASBX-η	30	0.20862	0.20822

3. Metrics: tracking RMSE and MAE between y and r over T. Hardware: Intel Core i9, 64 GB RAM, Ubuntu 24.04.

Results. Table 5 summarizes performance at two generation budgets $(30, 40)$ under the large-horizon, extended-population regime. RB–GA consistently improves tracking accuracy over the Baseline GA, with the clearest gains at 40 generations (RMSE 0.1678 vs. 0.1774, MAE 0.1397 vs. 0.1553). Baseline+SVPS tracks the Baseline identically in this configuration, confirming that deterministic shrinkage alone does not alter accuracy. TLASBX, despite introducing Bayesian optimization of the SBX parameter on a source task, converges to numerically identical trajectories to the Baseline under our reported setup, suggesting that in this regime the learned η does not change closed-loop behavior. The closeness of results across Baseline, Baseline+SVPS, and TLASBX can be attributed to the relatively small number of generations allocated to the GA, which limits the opportunity for their respective learning mechanisms to manifest. Output overlays in Fig. 5 illustrate that all methods track the target closely; only RB–GA exhibits a visible reduction in error envelopes, while SVPS and TLASBX coincide with the Baseline.

5 Conclusion

Genetic Algorithms (GAs) show potential for use in receding-horizon control of nonlinear and constrained systems, although their computational expense primarily stems from numerous objective function evaluations. This study introduces a specialized rejection-based GA designed specifically for genome structures related to horizons. It ensures only one batch of plant evaluations per generation and utilizes an efficient, calibrated classifier to screen out untested offspring. In comparison experiments on a typical non-isothermal CSTR scenario, four different optimizers were evaluated under uniform conditions: a basic GA, the newly developed committee-gated RB-GA, a baseline augmented with simple variable population sizing (SVPS), and Two-Level Adaptive Simulated Binary Crossover (TLASBX). Results demonstrated that RB-GA consistently enhanced tracking precision compared to the baseline while using fewer generational resources. Both Baseline+SVPS and TLASBX showed performance similar to the baseline due to the restricted number of iterations limiting their ability to learn effectively. These findings suggest that the proposed methodology enhances optimization efficiency particularly in computationally intensive environments that seems to be very important and promising for use in open real-time automation systems like cyber-physical ones. Moreover, the presented approach can be considered as a basis for implementation of specific GA for computationally-consuming real-time tasks.

6 Limitations and Future Work

While the proposed RB-GA achieves substantial evaluation savings and improved convergence, several limitations remain. First, the gating mechanism depends on classifier calibration quality; inaccurate calibration or limited replay data can lead to premature rejection of promising candidates. Second, the approach assumes deterministic evaluations and noise-free feedback, which may not hold in practical control settings. Third, while we enforced one evaluation batch per generation, this constraint may slow convergence for low-cost objectives where more evaluations are affordable. Finally, the current implementation uses fixed hyperparameters for quantile labeling, false-reject tolerance, and committee disagreement.

An important direction for future research is the application of RB-GA to partial differential equations (PDE)-constrained optimal control problems, such as subsurface reservoir optimization, where objective evaluations depend on computationally intensive simulators [Sirota and Ovsyannikov, 2025]. While recent work has demonstrated the effectiveness of neural operator surrogates for enabling gradient-based optimization in such settings, rejection-based evolutionary controllers provide a complementary, gradient-free alternative that may offer improved robustness to model mismatch, nonconvex constraints, and limited differentiability.

Future work will focus on adaptive schemes that learn these parameters online, probabilistic committees that quantify uncertainty, and extensions to constrained and stochastic optimization. We also plan to explore differentiable surrogate gates embedded within gradient-based meta-optimization, and real-time deployment on cyber-physical testbeds for process control and microgrid energy management.

Acknowledgment

The research is funded by the Ministry of Science and Higher Education of the Russian Federation under Agreement No. 075-15-2025-674 dated 08/20/2025.

References

Aivaliotis-Apostolopoulos, P. and Loukidis, D. (2022). Swarming genetic algorithm: A nested fully coupled hybrid of genetic algorithm and particle swarm optimization. *PLoS One*, **17** (9), pp. e0275094.

Al-Terkawi, L. and Migliavacca, M. (2025). An automated parallel genetic algorithm with parametric adaptation for distributed data analysis. *Scientific Reports*, **15** (1), pp. 10836.

Ali, A. F. and Tawhid, M. A. (2017). A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Engineering Journal*, **8** (2), pp. 191–206.

Andreadis, G., Alderliesten, T., and Bosman, P. A. (2024). Fitness-based linkage learning and maximum-clique conditional linkage modelling for gray-box optimization with rv-gomea. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 647–655.

Babich, N., Marzel, E., Rybalko, A., Chen, O., Chulkin, V., and Fradkov, A. (2025). Outline of cybernetical neuroscience. *Cybernetics and Physics*, **14** (1), pp. 13–18.

Berberich, J., Köhler, J., Müller, M. A., and Allgöwer, F. (2022). Linear tracking mpc for nonlinear systems—part i: The model-based case. *IEEE transactions on automatic control*, **67** (9), pp. 4390–4405.

Cavus, M. and Allahham, A. (2024). Enhanced microgrid control through genetic predictive control: integrating genetic algorithms with model predictive control for improved non-linearity and non-convexity handling. *Energies*, **17** (17), pp. 4458.

Chen, G., Li, Y., Zhang, K., Xue, X., Wang, J., Luo, Q., Yao, C., and Yao, J. (2021). Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization. *Information Sciences*, **542**, pp. 228–246.

Consoli, P. A., Mei, Y., Minku, L. L., and Yao, X. (2016). Dynamic selection of evolutionary operators based on online learning and fitness landscape analysis. *Soft Computing*, **20** (10), pp. 3889–3914.

Dang, D.-C., Opris, A., and Sudholt, D. (2024). Crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. *Artificial Intelligence*, **330**, pp. 104098.

Deb, K. (1999). An introduction to genetic algorithms. *Sadhana*, **24** (4), pp. 293–315.

Deeb, A., Aakel, A., Khokhlovskiy, V., and Shkodyrev, V. (2024a). Optimal path-following control for redundant manipulators: A multi-objective optimization approach. In *Mechatronics and Automation Technology*, pp. 52–59. IOS Press.

Deeb, A., Khokhlovskiy, V., and Shkodyrev, V. (2025a). Adaptive simulated binary crossover with bayesian optimization for industrial applications. In *2025 International Russian Smart Industry Conference (SmartIndustryCon)*, IEEE, pp. 608–613.

Deeb, A., Khokhlovskiy, V., and Shkodyrev, V. (2025b). Hierarchical multi-objective control of nonlinear systems with dynamical input constraints. In *Artificial Intelligence and Applications*.

Deeb, A., Khokhlovskiy, V. N., and Shkodyrev, V. P. (2024b). Model predictive control and genetic algorithms for optimization of continuous stirred tank reactors. In *Smart Electromechanical Systems: Mathematical and Software Engineering*, pp. 185–191. Springer.

Doerr, B., Echarghaoui, A., Jamal, M., and Krejca, M. S. (2024). Runtime analysis of the (μ+ 1) ga: provable speed-ups from strong drift towards diverse populations. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 35–36.

Duan, P., Yu, Z., Gao, K., Meng, L., Han, Y., and Ye, F. (2024). Solving the multi-objective path planning problem for mobile robot using an improved nsga-ii algorithm. *Swarm and Evolutionary Computation*, **87**, pp. 101576.

Dushatskiy, A., Virgolin, M., Bouter, A., Thierens, D., and Bosman, P. A. (2024). Parameterless gene-pool optimal mixing evolutionary algorithms. *Evolutionary computation*, **32** (4), pp. 371–397.

Fialho, Á. (2010). *Adaptive operator selection for optimization*. PhD thesis, Université Paris Sud-Paris XI.

Geng, H., Xu, K., Zhang, Y., and Zhou, Z. (2023). A classification tree and decomposition based multi-objective evolutionary algorithm with adaptive operator selection. *Complex & Intelligent Systems*, **9** (1), pp. 579–596.

He, C., Zhang, Y., Gong, D., and Ji, X. (2023). A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Systems with Applications*, **217**, pp. 119495.

Jiang, H., Wang, G., Liu, Q., and Cao, J. (2024). A multi-population surrogate-assisted algorithm based on granular-ball k-nearest neighbors classification for expensive optimization. In *2024 IEEE/WIC Interna-*

tional Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE, pp. 411–415.

Jiménez Laredo, J. L., Fernandes, C., Julián Merelo, J., and Gagné, C. (2024). Improving genetic algorithms performance via deterministic population shrinkage. *arXiv e-prints*, pp. arXiv–2401.

Khodadoost, S., Saraee, M., Talatahari, S., and Sareh, P. (2024). Optimal design of fractional-order proportional integral derivative controllers for structural vibration suppression. *Scientific Reports*, **14** (1), pp. 17207.

Li, J., Wang, P., Dong, H., Shen, J., and Chen, C. (2022). A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization. *Knowledge-Based Systems*, **242**, pp. 108416.

Liang, J., Lou, Y., Yu, M., Bi, Y., and Yu, K. (2024). A survey of surrogate-assisted evolutionary algorithms for expensive optimization. *Journal of Membrane Computing*, pp. 1–20.

Liu, S., Tian, Q., and Tang, C. (2024a). Mobile robot path planning algorithm based on nsga-ii. *Applied Sciences*, **14** (10), pp. 4305.

Liu, S., Wang, H., Peng, W., and Yao, W. (2024b). Surrogate-assisted evolutionary algorithms for expensive combinatorial optimization: a survey. *Complex & Intelligent Systems*, **10** (4), pp. 5933–5949.

Mao, F., Chen, M., Zhong, K., Zeng, J., and Liang, Z. (2024). An xgboost-assisted evolutionary algorithm for expensive multiobjective optimization problems. *Information Sciences*, **666**, pp. 120449.

Matanga, Y., Owolawi, P., Du, C., and van Wyk, E. (2024). Niching global optimisation: Systematic literature review. *Algorithms*, **17** (10), pp. 448.

Opris, A. (2025). A many-objective problem where crossover is provably indispensable. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 27108–27116.

Opris, A., Lengler, J., and Sudholt, D. (2025). Achieving tight runtime bounds on jump k by proving that genetic algorithms evolve near-maximal population diversity. *Algorithmica*, pp. 1–56.

Pan, L., He, C., Tian, Y., Wang, H., Zhang, X., and Jin, Y. (2018). A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **23** (1), pp. 74–88.

Qiao, Y. and Gallagher, M. (2024). Analyzing the runtime of the gene-pool optimal mixing evolutionary algorithm (gomea) on the concatenated trap function. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1520–1526.

Rahimi, I., Gandomi, A. H., Nikoo, M. R., Mousavi, M., and Chen, F. (2024). Efficient implicit constraint handling approaches for constrained optimization problems. *Scientific Reports*, **14** (1), pp. 4816.

Ramos, M. P., Busboom, A., and Slobodyan, A. (2024). Genetic pi controller tuning to emulate a pole assignment design. *IFAC-PapersOnLine*, **58** (7), pp. 138–143.

Saini, N. and Ohri, J. (2023). Load frequency control in three-area single unit power system considering non-linearities effect. *Cybernetics and Physics*, **12** (1), pp. 60–69.

Shem-Tov, E. and Elyasaf, A. (2024). Deep neural crossover. *arXiv preprint arXiv:2403.11159*.

Sirota, D. and Ovsyannikov, D. (2025). Pde-constrained optimal control of subsurface reservoir systems via neural operator surrogates. *Cybernetics and Physics*, **14** (3), pp. 285–291.

Sun, Y., Wang, H., Xue, B., Jin, Y., Yen, G. G., and Zhang, M. (2019). Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, **24** (2), pp. 350–364.

Wang, H., Jin, Y., and Doherty, J. (2017). Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE transactions on cybernetics*, **47** (9), pp. 2664–2677.

Wang, H., Jin, Y., Sun, C., and Doherty, J. (2018). Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation*, **23** (2), pp. 203–216.

Wang, Z., Liu, S., Chen, J., and Tan, K. C. (2024). Large language model-aided evolutionary search for constrained multiobjective optimization. In *International Conference on Intelligent Computing*, Springer, pp. 218–230.

Wu, W., Zhang, L., Le, J., and Lu, Z. (2025). Integrated method for multi-uav task assignment and trajectory planning with deadlock based on three-dimensional dubins path. *Scientific Reports*, **15** (1), pp. 24152.

Yin, S. and Xiang, Z. (2024). Adaptive operator selection with dueling deep q-network for evolutionary multi-objective optimization. *Neurocomputing*, **581**, pp. 127491.