Contents

1. Introduction

Multi-objective optimization (MOO) involves optimizing two or more conflicting objectives simultaneously. Instead of a single optimum, the result is a set of trade-off solutions (Pareto-optimal solutions) forming the Pareto front, which offers decision-makers a spectrum of choices. Such problems are common in engineering design, economics, logistics, and other fields where multiple criteria must be balanced[1][2]. A central challenge in solving MOO problems is achieving convergence (finding solutions close to the true Pareto-optimal front) while preserving diversity (maintaining a well-spread set of solutions along the front). Over the past decades, many evolutionary algorithms have been developed to address this challenge – notably the Non-dominated Sorting Genetic Algorithm II (NSGA-II)[3] and the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)[4] – which aim to balance convergence and diversity in approximating the Pareto front.

Differential Evolution (DE) has emerged as a powerful stochastic optimizer for continuous problems, known for its simple structure and high efficiency[5]. DE operates on a population of candidate solutions through mutation (perturbing solutions by scaled differences of others), crossover (recombination), and selection (survival of the fittest offspring). Thanks to its straightforward operations and small number of control parameters (mutation factor *F*, crossover rate *CR*, population size *NP*), DE often converges faster and requires less fine-tuning of parameters compared to classic genetic algorithms or particle swarm optimizers[5][6]. These advantages have made DE popular in numerous domains and competition benchmarks[6]. However, applying DE to multi-objective optimization poses additional difficulties. DE's performance is highly sensitive to its control parameters[7], which can hinder effectiveness on complex problem landscapes if the parameters are not properly set or adapted. Moreover, in a multi-objective context, a DE-based algorithm must not only converge toward the Pareto front but also maintain a diverse set of Pareto-optimal solutions. Achieving this dual goal – strong convergence and diversity – is non-trivial and often demands specialized mechanisms within the DE framework.

Over the years, researchers have proposed various enhancements to improve DE's robustness and solution quality, especially via parameter self-adaptation and hybrid strategies. One landmark approach was the self-adaptive DE variant *jDE* by Brest *et al.* (2006), which encoded the mutation factor *F* and crossover rate *CR* into each individual and allowed them to evolve along with the population[7]. This idea of automated parameter control paved the way for many subsequent variants. For example, Zhang and Sanderson's JADE algorithm introduced an optional external archive of inferior solutions and a "current-to-*p*best/1" mutation strategy alongside an adaptive parameter scheme[16]. Tanabe and Fukunaga's SHADE method further advanced parameter adaptation by recording a history of successful F and CR values in a finite memory to guide the selection of parameters for new generations[8]. An improved version, L-SHADE, additionally incorporated linear population size reduction, gradually shrinking the population over time to balance exploration and exploitation and thus accelerating convergence[9]. These developments made L-SHADE one of the state-of-the-art DE algorithms in single-objective optimization competitions[9]. More recently, dual-population techniques have shown promise for enhancing DE. Stanovov *et al.* proposed *L-NTADE* (Linear population size reduction – Newest and Top Adaptive DE), which maintains two evolving sub-populations – one containing the newest individuals and another preserving the top (elite) individuals – and combines their information to guide mutations[10]. This dual-population co-evolution strategy helps balance exploration and exploitation, leading to improved diversity maintenance and convergence speed in single-objective settings[10].

Building on these advances, this paper proposes a novel multi-objective differential evolution algorithm called MO-MO-LSRTDE, designed to more effectively handle the trade-off between convergence and diversity. The proposed MO-MO-LSRTDE is inspired by the recent success of *success rate*-based parameter adaptation in DE. In particular, Stanovov *et al.* introduced an enhanced DE variant termed *L-SRTDE* (Linear Success Rate-based DE), which replaces the historical memory-based adaptation in L-SHADE with a success rate-driven scheme – dynamically adjusting the mutation scaling factor (and other parameters) based on the proportion of offspring that are successful each generation[11]. MO-MO-LSRTDE extends the core ideas of L-SRTDE to the multi-objective domain. It integrates the success rate-driven adaptation mechanism with a dual-population evolutionary framework (analogous to the newest/top co-evolution in L-NTADE[10]) to simultaneously promote rapid convergence and maintain a diverse set of Pareto-optimal solutions. In essence, one sub-population in MO-MO-LSRTDE emphasizes exploration with newly generated solutions and adaptive parameter tuning, while another conserves high-quality Pareto-optimal individuals; periodic information exchange between them guides the search.

In summary, the key innovation of MO-MO-LSRTDE lies in synergizing two potent strategies – success-based parameter self-adaptation and two-population coevolution – within a multi-objective DE algorithm. To the best of our knowledge, this is the first attempt to combine a success rate-driven DE (as in L-SRTDE) with a cooperative dual-population approach for multi-objective optimization. The remainder of this paper is organized as follows: Section II reviews related work on multi-objective evolutionary algorithms and DE variants, Section III details the proposed MO-MO-LSRTDE algorithm, Section IV presents experimental results in comparison with state-of-the-art methods, and Section V concludes with insights and future directions.

2. Related Work

2.1. NSGA-II and MOEA/D: Classical MOEAs and Limitations

Multi-objective evolutionary algorithms such as NSGA-II and MOEA/D have become standard tools for Pareto optimization. NSGA-II, introduced by Deb *et al.*[3], employs fast non-dominated sorting with an elitist survival strategy and uses a crowding-distance mechanism to maintain solution diversity. This approach ensures a well-distributed approximation of the Pareto front and has been widely applied across domains[3]. MOEA/D, proposed by Zhang and Li[4], takes a decomposition-based approach: it breaks a multi-objective problem into a number of single-objective subproblems defined by weight vectors, and optimizes them in parallel while periodically exchanging information among neighboring subproblems[4]. By solving many weight-vector subproblems simultaneously, MOEA/D can achieve strong convergence (each subproblem pulls solutions toward a part of the front) and a uniform spread of solutions when the weight set is well-chosen[4]. Both NSGA-II and MOEA/D have demonstrated effectiveness on numerous benchmark tests and real-world applications.

However, these classic algorithms also have notable limitations. NSGA-II relies on fixed control parameter settings (e.g. crossover and mutation probabilities) that remain static during the run, lacking any built-in self-adaptation to different problem landscapes. Similarly, MOEA/D's performance is sensitive to its predefined parameters (such as neighborhood size or aggregation method) and especially to the chosen weight distribution, which often requires manual tuning for each problem. Moreover, NSGA-II's diversity preservation via crowding distance tends to deteriorate as the number of objectives grows large: in many-objective scenarios, most individuals become non-dominated, making it difficult for crowding distance to maintain selection pressure and a well-spread front[12]. MOEA/D implicitly promotes diversity through its set of weight vectors, but if the true Pareto front has a complex shape or the weight vectors are unevenly distributed, MOEA/D may struggle to cover the front adequately[13]. In summary, while NSGA-II and MOEA/D are robust general-purpose MOEAs, they do not adapt their strategy parameters on the fly and have only limited mechanisms to adaptively safeguard population diversity. These shortcomings have motivated research into more adaptive and diversity-aware multi-objective optimizers.

2.2. Differential Evolution in Multi-Objective Optimization

Differential Evolution, known for its powerful search operators in continuous space, has also been extensively adapted for multi-objective optimization. Early efforts to create multi-objective DE (MODE) algorithms focused on incorporating Pareto-based selection and elitism into the DE framework. For example, Abbass introduced a Pareto Differential Evolution (PDE) that applied Pareto dominance selection and archiving to DE, enabling it to evolve a set of non-dominated solutions in one run[14]. Such developments established DE as a viable engine for multi-objective search. However, the early multi-objective DE approaches often used fixed control parameters and relatively simple diversity preservation schemes, which could hamper their effectiveness on diverse or complex problem landscapes.

Subsequent algorithms improved on these ideas. Notably, GDE3 (the Third Generation Differential Evolution) by Kukkonen and Lampinen extended DE to handle an arbitrary number of objectives and constraints, using a selection mechanism akin to NSGA-II's elitism combined with DE's variation operators[15]. GDE3 introduced a crowding-distance-like measure in DE's selection and maintained an external archive of non-dominated solutions, which helped achieve a better distribution of solutions along the Pareto front compared to earlier DE variants[15]. The performance of GDE3 was shown to be at least comparable to NSGA-II on various problems, and it represented an important step in making DE competitive in multi-objective optimization. Other variants (often simply termed "MODE" in literature) also explored different selection or archiving strategies, but a common trait of these early methods was that they did not yet incorporate advanced parameter adaptation – their mutation and crossover rates were typically static, and diversity promotion was handled by generic means (like basic crowding or ε-dominance) rather than problem-responsive mechanisms.

One key research trajectory to enhance DE (in both single- and multi-objective contexts) has been the self-adaptation of strategy parameters. Canonical DE requires the user to set a mutation factor F and crossover rate CR in advance, but the optimal values can vary widely across problems and even across different stages of the search. Brest *et al.* addressed this issue by proposing the jDE algorithm, which self-adapts their values: each individual carries its own F and CR that mutate and evolve over generations[7]. This mechanism let the population auto-tune its parameters, improving robustness by reducing the need for problem-specific parameter tuning. Following this idea, numerous adaptive DE variants emerged. JADE (2009) introduced an optional external archive to store previously discarded solutions (to increase mutation diversity) and employed a "$current-to-pbest/1$" mutation along with an adaptive F and CR scheme[16]. In JADE, the parameters are randomly sampled each generation from distributions whose means are continuously updated based on successful experiences, ensuring that F and CR gradually move toward favorable values[16]. Tanabe and Fukunaga's SHADE algorithm further refined this approach by maintaining a historical memory of successful parameter values[8]. In SHADE, each generation's successful F and CR pairs are saved into a limited-size memory, and new trial parameters are generated by sampling from this memory (biased toward more successful values)[8]. This success-history adaptation strategy, combined with an external archive mechanism (inherited from JADE) for diversity, significantly improved DE's performance on complex benchmarks. Tanabe and Fukunaga later enhanced SHADE with linear population size reduction in their L-SHADE algorithm[9]. By gradually decreasing the population size during the run, L-SHADE was able to maintain exploration in early phases and intensify exploitation in later phases, and it achieved state-of-the-art results in single-objective optimizer competitions[9].

More recently, researchers have investigated new adaptive mechanisms for DE, including hybrid and dual-population designs. A notable development is the L-SRTDE (Linear Success Rate-based DE) proposed by Stanovov *et al.*[11]. L-SRTDE inherits the population size reduction of L-SHADE and introduces a novel success rate-driven parameter adaptation. Instead of using a memory of past successful parameters, L-SRTDE dynamically adjusts the mutation factor *F* based on the *success rate* in each generation[11]. The success rate is defined as the fraction of offspring that enter the next population (i.e., the proportion of new solutions that are improvements). If the success rate is high, L-SRTDE decreases certain parameters to focus exploitation on high-quality solutions; if the success rate is low, it broadens the search by increasing diversity-oriented parameters. This feedback loop allows the algorithm to react to its recent performance and continuously tune itself. Apart from this, L-SRTDE retains other features from L-NTADE[10] – in fact, L-SRTDE can be seen as blending the success-based adaptation with the dual-population framework of L-NTADE. The dual-population scheme (splitting the population into "newest" and "elite" sub-populations that evolve in tandem[10]) was originally shown to enhance DE's search by simultaneously exploring new regions and exploiting the best found solutions. By combining it with success rate adaptation, L-SRTDE represented a highly adaptive DE strategy in the single-objective domain.

Despite the progress above, most multi-objective evolutionary algorithms have not yet exploited this combination of mechanisms. Existing MOEAs like NSGA-II or MOEA/D do not self-adapt their parameters, and while some multi-objective DE variants use adaptive DE techniques (e.g., parameter adaptation from SHADE or others), none have explicitly integrated a success-rate feedback adaptation *and* a dual-population coevolution into one framework. This gap in the literature is precisely what MO-MO-LSRTDE aims to fill. By uniting the success rate-driven parameter control (proven effective in L-SRTDE[11]) with a two-population evolutionary model (as in L-NTADE[10]), MO-MO-LSRTDE is designed to automatically balance convergence and diversity in multi-objective optimization. In the following section, we detail the components of the proposed MO-MO-LSRTDE algorithm and discuss how it addresses the limitations identified in the existing work.

2.3. Adaptive Parameter Control and Dual-Population Strategies in DE

Parameter adaptation has proven crucial for DE's success on difficult problems. As noted earlier, DE's performance can be highly sensitive to the choice of F (mutation scale factor) and CR (crossover rate). Optimal values may not only vary from problem to problem but even during different phases of the search on the same problem. Early adaptive DE approaches like jDE allowed each individual to carry its own F and CR and updated these values if the individual produced a successful offspring. JADE's success-history mechanism went further by recording successful parameter values in a global memory and sampling new parameters from a distribution biased toward those successes[4]. This "learning from success" paradigm has been very influential. In multi-objective EAs, a similar idea can be applied: for example, if a certain mutation step size frequently leads to Pareto improvements, bias future mutations towards that step size. MOEA/D-PS (Li & Zhang, 2009) did something akin to this by adjusting its search radius based on improvement frequency.

Recently, Stanovov *et al.* (2024) introduced an adaptation technique based on success rate (SR), as part of their L-SRTDE algorithm. The *success rate* is defined each generation as the fraction of offspring that enter the next generation (i.e., the proportion of trials that were an improvement). This single measure succinctly captures the algorithm's progress: a high SR indicates that many offspring are finding improvements (perhaps the search is in a fertile region or parameters are well tuned), whereas a low SR means few or no improvements (perhaps indicating stagnation or too aggressive jumps). L-SRTDE adjusts DE's F and CR based on SR feedback: for instance, it may *increase* the base mutation factor if SR is high to speed up convergence (since the search is making good progress), or *decrease* it if SR is very low to encourage more careful exploration. The exact scheme in one design was: $m_F = 0.4 + 0.25 \cdot \tanh(5 \cdot \mathrm{SR})$ which yields m_F around 0.4 when SR is near 0 and up to ~0.65 when SR is 1.0. Each individual's actual F is then sampled from a Cauchy distribution centered on m_F. Similarly, CR might be adjusted (often with a normal distribution) and successful CR values recorded for future generations. The success rate adaptation approach is appealing because it provides an automatic, self-regulating mechanism: when the algorithm struggles, it adapts to search more broadly; when it's succeeding, it hones in more aggressively. MO-LSRTDE will adopt a similar SR-driven adaptation for its parameters.

Meanwhile, dual-population or co-evolutionary strategies have gained traction as a way to balance exploration and exploitation. The idea is to run two (or more) sub-populations in parallel, each with a different search focus, and occasionally exchange information between them. Zheng *et al.* (2017) classified dual-population EAs as maintaining an "exploratory" population (e.g., random or diversity-focused search) and an "exploitative" population (e.g., local search around current bests). In the context of DE, Qin *et al.* (2016) proposed a dual-population DE where one population used a high mutation factor (diversify) and another used a low factor (intensify), with periodic migration of top individuals between them. The L-NTADE algorithm discussed earlier explicitly keeps a population of "newest" individuals (constantly refreshed with incoming offspring) and a population of "top" elites (the best solutions found so far)[4]. Every generation, L-NTADE's mutation operator may pick vectors from either population, thus mixing information from elites and new candidates[4]. This design was shown to markedly improve performance on multi-modal single-objective problems, as it prevents the optimizer from converging too fast on one region and missing others[4].

For multi-objective problems, dual-population ideas have also been explored. Some many-objective MOEAs use a secondary population to store extreme Pareto solutions or to maintain diversity (e.g., the two-archive algorithm AR+ES by Praditwong *et al.*, 2011, which had separate archives for convergence and diversity). The general theme is that by decoupling roles – one population to broadly explore the objective space, another to refine the known Pareto front – the algorithm can achieve a better balance than a single population trying to do both. MO-LSRTDE is conceptually inspired by this: it maintains an external archive of elite non-dominated solutions which acts like a second population of guiding elites, while its main population continuously generates new offspring. Although our current implementation does not have two entirely separate evolving populations (both archive and population undergo variation together), it lays the groundwork for a true dual-population extension in the future. We will show that even this implicit dual-population scheme (mixing new solutions and archive elites during variation) already provides benefits in maintaining diversity without sacrificing convergence.

Lastly, local search techniques in evolutionary multi-objective optimization deserve mention. Hybrid algorithms that incorporate a problem-specific local search or a general heuristic (like simplex search or gradient-based refinement, when available) have often shown improved convergence. For example, some MOEAs periodically apply a local optimizer to the best solutions to accelerate fine-tuning (a common strategy in memetic algorithms). In our approach, we include a simple local search step that can be applied to archive elites under certain conditions (e.g., if improvement stalls). This is done in a modest, problem-agnostic way (for instance, a few iterations of a random perturbation and Pareto greedy acceptance) just to exploit the vicinity of high-quality solutions. We maintain a modest tone here – the local search is not problem-tailored, and is included primarily to enhance final convergence accuracy on complex landscapes. As a result, MO-LSRTDE can be seen as a *memetic DE* at a high level: global evolution drives the search, and local search refinements are periodically applied to polish the best solutions. This yields an approach that is largely automatic but still leverages intensification when needed.

3. Proposed MO-LSRTDE Algorithm

MO-LSRTDE is a multi-objective evolutionary algorithm based on Differential Evolution, augmented with success rate-based parameter adaptation, an external elite archive, and an optional local search component. In this section, we describe the algorithm's framework and key components in a formal manner, while also providing insight into the design choices in a more approachable tone.

3.1. Algorithm Framework Overview

At its core, MO-LSRTDE follows the general structure of a generational EA, with a few important distinctions. Pseudocode for MO-LSRTDE is outlined in *Algorithm 1*. In summary, the algorithm starts by initializing a population and an elite archive, then enters an evolutionary loop where it generates offspring via DE operators (mutation and crossover), evaluates them, and selects the survivors based on Pareto dominance and diversity. Throughout this loop, the algorithm continuously adapts its mutation and crossover parameters according to the observed success rate of offspring, and it periodically invokes a local search on elite solutions. The result is an algorithm that learns and balances its search strategy dynamically.

Algorithm 1 MO-LSRTDE (Pseudocode of the proposed multi-objective LSRTDE)

Input: Problem's objective functions $f_1, f_2, \ldots, f_M$; population size N; maximum generations $G_{\max}$ (or stopping criterion).

Output: A (external archive of final nondominated solutions).

```
1:  Initialize population $P$ with $N$ individuals randomly.
2:  Evaluate each individual in $P$ on all objective functions $f_1, f_2, \ldots, f_M$.
3:  Initialize external archive $A$ with nondominated solutions from $P$.
4:  Set initial parameters (e.g., success rate $SR = 0$, $p_b$ = initial value for $p - best$ selection, etc.).
5:  Set generation counter $t = 0$.
6:  while ($t < G_{\max}$ and stopping criterion not met) do
7:      for each individual $x_i$ in $P$ do
8:          $y_i$ = MutationAndCrossover($x_i, P, p_b, \ldots$)
9:          // Generate offspring $y_i$ via DE mutation & crossover using current $p_b$ and other params
10:         Evaluate $y_i$ on all objective functions.
11:     end for
12:     $Q = P \cup \{y_1, y_2, \ldots, y_N\}$// Combined parent and offspring population
13:     $P$ = EnvironmentalSelection($Q, N, A$) // Select $N$ fittest solutions for next generation; update archive $A$
14:     Compute $SR = \frac{(\{y_i\}\text{that}P)}{N}$
15:     // $SR$ = success rate of offspring for this generation
16:     AdaptiveParameterUpdate($SR$)
```

```
17:     // Adaptive parameter update based on success rate (e.g., adjust $p_b$, mutation factor, etc.)
18:     $t = t + 1$.
19: end while
20: Output the final archive $A$ (set of nondominated solutions found).
```

In the pseudocode above, **MutationAndCrossover()** applies DE variation to each parent. Key processes in each generation are as follows:

Adaptive Elite Proportion: The parameter p_b determines the fraction of top individuals used for guiding mutations. It is adaptively adjusted based on the current success rate SR – for instance, using a decay such as $p_b = 0.7 \cdot exp(-7 \cdot SR)$ as suggested in L-SRTDE. This ensures that when *SR* is low (few improvements), a larger elite set (higher p_b) is used to encourage broad exploration, whereas when *SR* is high (rapid progress), p_b shrinks to intensify exploitation by focusing on the very best solutions. Such success-rate-driven elitism has been shown effective in balancing exploration and exploitation in DE.

Parent Selection: The current population (or combined with archive) is ranked by Pareto non-domination and crowding distance, as in NSGA-II. The top fronts are selected sequentially until *N* solutions are chosen. Ties within the last front are broken by higher crowding distance to preserve diversity. From the sorted individuals, the top $p_b \cdot N$ (at least two) form the $pbest$ pool. For each target individual, one $pbest$ parent is chosen uniformly from this elite pool to guide mutation.

Mutation and Crossover: Each target vector x_i generates a mutant vector v_i via a mixed strategy. With a fixed probability (e.g. 50%) it uses **DE/current-to-pbest/1**:

- $vi = xi + F \cdot (xpbest - xi) + F \cdot (xr1 - xr2),$

Where $\mathbf{x}_{pbest}$ is the selected elite guide and $\mathbf{x}_{r1}, \mathbf{x}_{r2}$ are two distinct random individuals. Otherwise it uses the classic **DE/rand/1**:

- $vi = xr1 + F \cdot (xr2 - xr3),$

with three distinct random vectors $\mathbf{x}_{r1}, \mathbf{x}_{r2}, \mathbf{x}_{r3}$. These two mutation schemes ($rand/1$ and $current - to - pbest/1$) are among the most popular DE variants. The mixed use of these strategies promotes diversity ($rand/1$) while exploiting good solutions ($current - to - pbest$). After mutation, binomial crossover is applied to form trial vectors y_i, by exchanging components between v_i and x_i with probability CR.

Evaluation: Each trial vector y_i is evaluated on all objective functions, and the generation count (and function evaluation count) is incremented.

Environmental Selection: Parents and offspring are combined (size $2N$) and undergo nondominated sorting and crowding-distance calculation. The next population is filled by selecting individuals from the best Pareto fronts until N survivors are chosen. Within a partially selected front, solutions with larger crowding distance are preferred to maintain spread. This elitist survival ensures that high-quality trade-off solutions are retained.

Archive Update: The external archive A is updated by merging it with the new population or offspring set and extracting the current nondominated set (possibly trimmed to a fixed size by diversity criteria). The archive thus accumulates all elite solutions discovered so far, guiding future generations and preserving a global approximation of the Pareto front.

Local Search (optional): Periodically, a local search procedure may be applied to archive members or selected solutions to intensify convergence around promising regions. For example, a tailored local optimizer or randomized hill-climbing can refine nondominated solutions, leveraging gradient or surrogate information. Such hybrid "memetic" enhancements have been shown to improve convergence speed and solution quality in MOEAs.

Success Rate Computation: The success rate SR is computed as the fraction of offspring that were accepted into the new population (i.e. the proportion of y_i that dominated or replaced parents). Formally, $SR = N_s/N$, where N_s is the number of successful offspring. By definition, $SR \in [0,1]$ and reflects search progress: a high SR means many new solutions improved upon their parents, whereas a low SR indicates stagnation.

Adaptive Parameter Update: Control parameters (F, CR, and p_b) are adapted based on SR. One approach is to record the F and CR values that generated successful offspring, and update the historical means or memory entries in the style of JADE/SHADE. The success rate itself can directly adjust the baseline F: for instance, increasing F modestly when SR is high to promote bolder steps, and decreasing F when SR is low to conserve fine-scale exploration. Similarly, p_b is updated as described above. These self-adaptive schemes obviate manual tuning; notably, explicit use of SR for adaptation is novel in DE algorithms (previous DE variants only used SR implicitly as part of success-history mechanisms).

Each generation thus completes an iteration of offspring generation, evaluation, and selection, after which t increments. Once the stopping criterion is reached, the final archive A (or the nondominated set of the current population) is returned as the Pareto-optimal approximation.

Computational Complexity: The dominant cost per generation lies in selection and sorting. Pareto ranking and crowding-distance computation on $2N$ individuals typically require $O(MN^2)$ time (where M is the number of objectives). Mutation, crossover, and objective evaluations cost $O(N \cdot D)$ per generation (with D decision variables). Any local-search routines add overhead proportional to their intensity, but this is often negligible compared to costly function evaluations. Overall, MO-LSRTDE's complexity is on par with other Pareto-based DE algorithms.

Design Rationale: MO-LSRTDE is designed to synergize proven techniques for multi-objective search. The dual-mutation strategy ($rand/1$ and $current-to-pbest/1$) combines exploration and exploitation, following the paradigm in JADE and related DE variants. The success-rate-driven adaptation autonomously tunes search parameters, accelerating convergence without hand-tuning (building on ideas from L-SRTDE). Maintaining an elite archive and using Pareto ranking/crowding (as in NSGA-II) ensures diverse solution spread. Together, these elements allow MO-LSRTDE to dynamically adjust its search behavior: it intensifies around good solutions when progress is being made, yet expands to explore new regions when stagnation is detected. The optional local search further boosts convergence precision, drawing on the "memetic" principle that hybridized local refinement can accelerate multi-objective optimization. In summary, MO-LSRTDE's framework integrates adaptive parameter control, elitist selection, and hybrid local search to address the convergence–diversity trade-off more effectively than standard MOEAs. Each design choice is grounded in established literature (e.g. JADE/SHADE adaptation, NSGA-II selection, dual-population concepts, and hybrid memetic heuristics) while introducing novel coordination of these mechanisms in a single algorithm.

3.2. Success Rate-Based Parameter Adaptation

The success of MO-LSRTDE hinges on its success rate-driven parameter control, so we detail this component here. As defined, the *success rate* SR in generation t is:

$$SRt = number\ of\ offspring\ that\ enter\ Pt + 1N, SR_t = \frac{\text{number}P_{t+1}}{N},$$

$$SRt \quad = Nnumber\ of\ offspring\ that\ enter\ Pt + 1 \qquad ,$$

where N is the population size (assuming steady state population). This metric is computed after selection in each generation. An offspring "enters" the next population either by dominating its parent (or some other individual) or by filling a slot of a removed dominated solution. If $SR_t = 1$, it means *every* offspring was an improvement (which is rare, perhaps in early generations on easy problems); if $SR_t =$

0, no offspring improved (full stagnation); $SR_t \approx 0.5$ means half the offspring on average were better than what they replaced.

In MO-LSRTDE, we use SR_t as feedback to adjust three things: (1) the elite proportion p_b for parent selection, (2) the mutation factor distribution, and (3) the crossover rate distribution. We already described the dynamic p_b above – essentially we set $p_b = \max(2, 0.7\exp(-\eta \cdot SR) \cdot N)$ for some constant η (in our runs, we used $\eta = 7$) so that when SR is low, $p_b \approx 0.7N$ (70% elites used) and when SR is high, p_b shrinks exponentially (e.g., at $SR = 0.5$, $p_b \approx 0.17N$; at $SR = 0.9$, $p_b \approx 0.005N$, which we floor to 2). This way, the pool of guides becomes narrower as the algorithm's success improves, focusing search exploitation.

For the mutation factor F adaptation: We maintain a global parameter m_F which represents the current central value of the mutation factor. At initialization, we set $m_F = 0.5$. After each generation's selection, we update m_F as follows:

We collect all F values that were used by offspring which succeeded (i.e., made it into the next population or archive). Let this set be S_F. If S_F is non-empty, we compute $\bar{F} = \text{mean}(S_F)$ or a weighted mean (some literature suggests using the Lehmer mean or a median). We then set

$$mF: = \lambda \cdot mF + (1 - \lambda) \cdot \bar{F},$$

$$m_F: = \lambda \cdot m_F + (1 - \lambda) \cdot \bar{F}, mF \quad : = \lambda \cdot mF \quad + (1 - \lambda) \cdot \bar{F},$$

with λ a memory coefficient (e.g., 0.8) to smooth out fluctuations. This *learns from successful parameters*. If no offspring succeeded (S_F empty), we leave m_F as-is for now (or in some implementations, slightly decrease it to encourage exploration after a failure).

We then adjust m_F further based on the current SR. In Stanovov's L-SRTDE, an explicit function $m_F(SR)$ was used. In our approach, we do something qualitatively similar: if $SR < SR_{\text{low}}$ (say 0.2), we *decrease* m_F by a small factor (to make mutations more subtle, as big moves aren't yielding improvement); if $SR > SR_{\text{high}}$ (say 0.8), we *increase* m_F slightly (trusting that bigger steps are working well). If SR is moderate, we keep m_F around its current learned value. We also clamp m_F within [0.2,0.9] to avoid extremes (to prevent pathologically small or large mutations). The exact thresholds are not very sensitive; the key is the trend of increasing $m_F\ with\ SR$.

During offspring generation, each parent's actual mutation factor F_i is *sampled* from a distribution centered at m_F. We use a Cauchy distribution with location m_F and scale 0.1 (for instance), and we bound the sampled F_i to $[0,1]$※. This tends to produce a majority of F_i near m_F but with some chance of outliers, which is useful for exploration. Each F_i used is recorded. This approach is adapted from SHADE/L-SHADE and ensures diversity in parameter values across individuals.

※ *Note:* If the Cauchy sample falls outside [0,1], we typically resample or truncate it. We ensure at least one F_i is exactly m_F for stability (common practice in SHADE).

For crossover rate CR adaptation: We perform an analogous procedure. We keep a global m_{CR} (initially 0.5). After each generation, we gather the CR values from successful offspring, compute their average $\bar{CR}$, and update $m_{CR} := \lambda m_{CR} + (1 - \lambda)\bar{CR}$. Then we may adjust m_{CR} slightly based on SR: if SR was very low, perhaps we lower m_{CR} (meaning we do more extrapolation rather than following parent structure, trying to find new areas); if SR was high, we raise m_{CR} (offspring are doing well, so bias toward inheriting more from mutants). We keep m_{CR}within [0,1]. During crossover, each parent has its own CR_i drawn from a normal distribution $\mathcal{N}(m_{CR},, 0.1)$ clipped to [0,1]. This adds some individual variation. Successful CR_i values feed back into the memory.

By adapting both F and CR continuously, MO-LSRTDE can respond to the problem's needs. For instance, on a simple convex problem, SR might quickly rise to ~0.8, causing m_F to increase a bit (maybe to ~0.6) and m_{CR} to stay moderate (~0.5–0.7). On a rugged problem where improvements are hard (low SR), the algorithm might settle into using a smaller F (~0.3) to fine-grind improvements and a smaller CR (~0.2) to inject more mutant components (since following the parent too closely wasn't working). This behavior was observed in our experiments and is a major reason for MO-LSRTDE's robust performance.

3.3. Archive-Guided Mutation and Dual-Population Emulation

The external archive A in MO-LSRTDE plays a pivotal role in guiding the search. It can be seen as representing a second population of elite solutions that co-evolves alongside the main population. Here we elaborate on how we use the archive and why this helps.

At every generation, after selection, the archive A contains the best non-dominated solutions found so far (up to size N). Some of these may be from the current generation's offspring, others may have been around for many generations (if no new solution dominated them). We do not apply DE variation operators directly to the archive members as a separate population (to save computation), but we *do* use archive members in the parent selection for mutation as described. In effect, any current population individual x_i has a chance to use an archive member as the p_{best} guide, or even be replaced by archive members in the mutation differences. For example, in the $current - to - p_{best}$ mutation, x_{pbest} is likely to be from A. In DE/rand/1, the random picks $r1, r2, r3$ might come from either P or A – we allow picks from A\$ as well. We ensure that at least one of the difference vectors in each mutant involves an archive solution when possible (by a simple heuristic: with 50% chance pick $r1$ from A, else from P, etc.). This means information from the archive (which tends to be high-quality solutions) is continuously fed into offspring generation. It increases exploitation pressure because offspring get pulled toward known Pareto-optimal regions.

At the same time, since the archive is external and does not directly compete for survival in P (it's always preserved separately), there is an implicit exploratory benefit: the main population P can pursue new areas without immediately losing the elite solutions (they survive in A even if not in P). In practice, we observed that many archive solutions remain in A for numerous generations, even if they temporarily get lost from P due to a slight deterioration in fitness. Later, they can come back as guides to produce offspring again. This resembles an "elder population" of tried-and-true solutions complementing the "youth population" of current search. It helps prevent permanent loss of diversity.

The dual-population effect can be summarized as: the archive A intensifies search around the best found trade-offs (by being used in p_{best} guidance), whereas the population P continues to generate novel solutions (especially via DE/rand/1 which can roam). In extreme cases, if P converges prematurely, A still holds the diverse Pareto front, and the mutation using A can re-introduce diversity into P. If A ever gets too large (exceeds N), we crowding-sort it and truncate, ensuring it remains diverse.

An important detail is how we update the archive efficiently each generation. We take all offspring (and possibly parents) and do one global non-dominated sort with archive + offspring, then truncate. This costs $O((N + |A|)^2)$ which is fine since both are $O(N)$. By doing this, archive always contains the latest non-dominated set of *all solutions seen so far*. It is possible in long runs for archive to accumulate many points (if the Pareto front is very large or finely sampled). If needed, one could limit archive size to, say, 2N and occasionally remove very old solutions, but in our benchmarks (20 problems) this was not an issue with $N = 100$. We kept archive size capped at 100.

3.4. Local Search Enhancement

Incorporating a *local search (LS) operator* can improve convergence, particularly for finely adjusting solutions on difficult Pareto fronts or satisfying constraints. MO-LSRTDE includes a simple local search routine that operates on archive solutions. We term it "learning-based local search" because it uses information from the archive and success history to decide where to apply local moves.

Our local search is triggered under two scenarios: (a) when the algorithm detects stagnation (e.g., SR has been <0.1 for a certain number of generations, meaning very few improvements are happening), or (b) at predefined infrequent intervals (for example, every 20 generations) as an intensification step. When triggered, the LS routine does the following:

Select a few candidate solutions from the archive A. We prioritize solutions that are *extremes* of the Pareto front (since improving them can increase hypervolume significantly) or solutions that have high crowding distance (isolated solutions that might represent unexplored regions). Often these coincide. For example, in a bi-objective case, we'd likely pick the archive member with the best f_1, the one with best f_2, and perhaps one in the middle that has sparse neighbors.

For each selected solution x, perform a series of small perturbations: e.g., for each decision variable x_j, create a modified solution $x^{(j,+)}$ by adding a small $\delta to x_j$ (with other variables same), and $x^{(j,-)}$ by subtracting δ. The step size δ can be set as a fraction of the variable domain or based on the resolution needed (we used a few percent of the domain or a fixed tiny value if normalized). Evaluate these perturbed solutions.

If any perturbed solution $x^{(j,\pm)}$ dominates x or fills a gap in the archive (i.e., is non-dominated with respect to the current archive), then insert it into the archive (and remove any archive solutions it might dominate). This is essentially a (1+1)-EA local move on x. If multiple such improvements are found, keep them all (the archive can store multiple).

We also tried a simple pattern search: if $x^{(j,+)}$ is good, we might try an additional step in that direction (2δ, etc.) until no improvement.

The effect of this local search is modest but noticeable: it often helps in constrained problems where a small tweak can reduce constraint violation and make a solution feasible (thus dominating many infeasible ones), or in continuous problems where the Pareto front is very smooth and an EA might bounce around without quite reaching the extreme corners. Local search can pull those extremes in more quickly. Because we apply it only occasionally, it doesn't add much computational burden. In a sense, it's like giving the algorithm a chance to "catch its breath" and fine-tune what it has found, before continuing global exploration.

Constraints: For problems with constraints (which some of the engineering cases have), our algorithm handles them by a simple feasibility rule: among two solutions, if one is feasible and the other not, the feasible one is considered better (dominates in a sense); if both are infeasible, we compare by overall constraint violation (smaller total violation is better). This is integrated in the dominance check. The local search is especially helpful here: if an archive solution is infeasible but close to feasibility, local search might reduce the violation enough to make it feasible, thereby dramatically improving its standing. We saw this on some RE problems.

Overall, the local search component adds an extra layer of convergence assurance. It is not the centerpiece of MO-LSRTDE, but it contributes to the algorithm's strong performance on certain problems, ensuring that once the general area of the Pareto front is found, the solutions can be fine-tuned to be as close to true Pareto-optimal as possible.

4. Experiments and Comparative Results

We conducted an extensive experimental study to evaluate MO-LSRTDE's performance relative to two state-of-the-art multi-objective optimizers: NSGA-II and MOEA/D. The goals of the experiments were to assess the algorithms in terms of convergence (how closely the obtained solutions approach the true Pareto front), diversity (how well the solutions spread across the front), and stability or consistency (how reproducible the results are across independent runs). We used a diverse suite of 20 standard benchmark functions and 5 real-world engineering design problems to cover a broad range of scenarios. For each algorithm on each problem, we measured standard performance metrics – specifically Hypervolume (HV), Inverted Generational Distance (IGD), and Spacing – and applied statistical tests to check if differences are significant.

Below we first describe the test problems, then the baseline algorithm settings and metrics, and finally present the results with analysis. All experiments were conducted with a population size of $N = 100$ (for each algorithm) and (unless otherwise stated) for 100 generations, i.e. 10,000 function evaluations total. This evaluation budget is in line with common practice and was found sufficient for convergence on most benchmarks. For a few harder cases (many-objective or very rugged problems), we allowed a larger budget (e.g. 20,000 evaluations) – this is noted when applicable. Each algorithm was run for 50 independent trials with different random seeds to gather statistics.

4.1. Benchmark Functions

We selected 20 well-known benchmark problems that pose a variety of challenges to multi-objective optimizers. These include the ZDT family, DTLZ family, WFG family, and two multi-modal multi-objective test problems (SYM-PART and Omni-test). Table 1 provides an overview of these benchmarks, their number of objectives (as used in our tests), and their key characteristics.Engineering Design Problems

Table 1. Benchmark problem suites and their characteristics.

Suite (problem count)	Objectives	Key characteristics
ZDT1–ZDT6 (6 functions)	2 objectives	Classic 2-D problems with various Pareto front shapes: convex (ZDT1), concave (ZDT2), discontinuous (ZDT3), multi-modal in decision space (ZDT4 has many local optima), discrete Pareto set (ZDT5 is binary-coded), and non-uniform Pareto front (ZDT6, where solutions cluster at one end). These test an algorithm's ability to handle different front geometries and local optima.
DTLZ1–DTLZ 7 (7 functions)	Scalable (tested with 3 objectives)	Scalable many-objective benchmarks introduced by Deb *et al.*. We used 3 objectives (tri-objective) to evaluate performance in a higher-dimensional objective space. DTLZ1 and DTLZ3 have a linear Pareto front but with many local Pareto-optimal regions (DTLZ3 is multi-modal). DTLZ2, DTLZ4, DTLZ5, DTLZ6 have smooth concave Pareto fronts (spherical or biased), testing the algorithm's ability to maintain uniform spread; DTLZ4 in particular introduces a bias that makes extreme solutions hard to obtain. DTLZ7 has an unusual disconnected Pareto front composed of multiple disjoint regions, posing a challenge for convergence on all segments.
WFG1–WFG3 (3 functions)	3 objectives	Three problems from the Walking Fish Group suite, using 3 objectives. The WFG functions involve complex "feature" transformations that create deceptive and non-convex Pareto fronts. WFG1 has mixed convex/concave front, WFG2 often yields a disconnected Pareto front, and WFG3 introduces nonlinear parameter dependencies. These demand balancing convergence with diversity in tricky landscapes that can mislead search.
SYM-PART (2 variants: simple, rotated)	2 objectives	A multimodal MOO problem where the Pareto front is symmetric and known, but there are multiple equivalent Pareto-optimal decision space regions that map to the same front. The "simple" variant is axis-aligned, while the "rotated" variant applies a linear transformation to decision variables. Both have *nine* distinct decision-space optima for any given Pareto outcome. These problems test whether an algorithm can find and preserve diverse equivalent solutions (multi-modality) and whether it is invariant to rotations of the search space.
Omni-test (2	2 objectives	A difficult multimodal two-objective problem proposed by Deb,

Suite (problem count)	Objectives	Key characteristics
variants: 2D and 3D decision space)		notable for having an exponential number of Pareto-optimal decision space regions. Specifically, for a decision vector of dimension D, there are 3^D global optima forming the Pareto set. We tested two versions: one with $D=2$ (9 optima) and one with $D=3$ (27 optima). All these optima map to a single continuous Pareto front in objective space. The Omni-test's extremely rugged landscape (many local optima) challenges an algorithm's ability to avoid being trapped and to eventually cover the continuous front despite the discrete decision-space optima.

The combination of these problems provides a thorough stress-test. For example, ZDT and DTLZ examine basic convergence and diversity in 2–3 objectives, WFG adds deceptive features, and SYM-PART/Omni-test push the algorithm on multimodal landscapes. An ideal algorithm should converge close to the true Pareto front on all of them and maintain a representative spread of solutions across each front.

4.2. Engineering Design Case Studies (RE Problems)

In addition to the synthetic benchmarks, we evaluated the algorithms on five real-world inspired engineering design problems from the CEC'2009 and CEC'2011 RE benchmark suites. These problems, labeled RE21, RE25, RE31, RE36, and RE37, each involve multiple objectives and various constraints. They are taken from different domains of engineering design and were compiled by Tanabe & Ishibuchi (2020) into a test suite of offline multi-objective optimization problems. Below is a brief description of each:

RE21 – Four-bar truss design: Minimize weight and deflection of a 4-bar truss subject to stress constraints (2 objectives, 4 continuous design variables). This is a structural optimization problem balancing material use vs. performance.

RE25 – Pressure vessel design: Minimize cost and weight of a cylindrical pressure vessel (2 objectives, 4 mixed variables – some continuous, some integer) under pressure and volume constraints. A well-known mechanical design problem.

RE31 – Car side impact design: Minimize the weight and maximize the crash safety (or equivalently minimize acceleration) of a vehicle side impact beam (3 objectives, 3 continuous variables) with constraints on deformation. This is a safety engineering problem.

RE36 – Gear train design: Minimize deviations in gear ratio and overall size for a gear reduction unit (2 objectives, 7 discrete/integer variables). Highly combinatorial with many local optima due to discrete choices of gear teeth.

RE37 – Rocket injector design: A complex rocket engine injector optimization with 3 objectives (e.g., maximize efficiency, minimize cost and pressure drop) and 4 design variables. There are multiple nonlinear constraints ensuring feasibility of injection and combustion. The Pareto front for this problem is known to be irregular.

These problems present additional challenges such as constraints, discrete variables, and unknown true Pareto fronts (we rely on known best-known solutions from literature for reference). They test how the algorithms perform on practical scenarios where the search space may be noisy or discontinuous, and evaluating solutions is computationally more expensive (though in our case we treat them as black-box functions for fairness).

We used the formulations and bounds as given in the RE benchmark definitions. Since true Pareto fronts are not analytically known, we compare algorithms on their achieved HV and IGD against each other. All algorithms were allotted the same evaluation budgets as in benchmarks (10k evals) which is quite moderate; in real-world use one might allow more, but our aim is comparison under equal budgets.

4.3. Algorithm Parameter Settings

For a fair comparison, we used the same population size ($N = 100$) and number of generations for NSGA-II, MOEA/D, and MO-LSRTDE on each problem. This ensures each algorithm performs the same number of function evaluations. Specifically:

NSGA-II: We used the standard version with tournament selection, fast non-dominated sorting, and crowding distance. Crossover operator was *simulated binary crossover (SBX)* with probability 0.9 and distribution index 20; mutation was *polynomial mutation* with probability $1/n$ (where n is number of decision variables) and distribution index 20. These are classic default parameters from Deb's literature and were not tuned further. We did not incorporate any modifications like ϵ-dominance or reference points into NSGA-II.

MOEA/D: We used the decomposition approach with the weighted Tchebycheff scalarization. The number of weight vectors = 100 (population size), which for 2 objectives were spread uniformly along the line (for 3 objectives, we used a simplex lattice generation method to distribute 100 reference points). Neighborhood size was set to 20 (i.e., each subproblem mates with its 20 nearest weight vector neighbors). Variation operators were the same SBX crossover and polynomial mutation as NSGA-II (so that differences come from selection method, not variation). MOEA/D has a few minor parameters like frequency of weight vector adjustment (we didn't adjust weights) and penalty parameter in Tchebycheff (we kept standard). Essentially, we implemented a canonical MOEA/D.

MO-LSRTDE: We used the configuration described in Section 3. Key parameters were: initial $m_F = 0.5$, initial $m_{CR} = 0.5$; memory coefficient $\lambda = 0.8$ for parameter updates; success rate thresholds $SR_{\text{low}} = 0.2$, $SR_{\text{high}} = 0.8$ for adaptation behavior. The p_{best} elite fraction formula used $\alpha = 0.7 and \eta = 7$ as given earlier (so initially 70% elites when no info, and decays as SR rises). Dual mutation strategy probability 0.5 each. Local search was invoked at most every 20 generations or if $SR < 0.1$ for 5 consecutive generations; it perturbed each variable by $\pm 1\%$ of its range (or nearest feasible step for integers). Archive size was capped at 100. We did *not* tune these on a per-problem basis; the same settings were applied to all problems to demonstrate robustness.

All algorithms used a binary tournament selection for mating (except MOEA/D which selects neighbors deterministically), and all were initialized with the same Latin hypercube sampled initial populations for fairness in each run. Constraint handling: NSGA-II and MOEA/D were given the same simple constraint handling as MO-LSRTDE (feasible solutions always preferred; if comparing two infeasible solutions, the one with smaller total normalized violation is preferred).

We emphasize that no algorithm had problem-specific tuning; we chose common settings from literature. The idea is to compare a well-established baseline (NSGA-II), a widely-used modern baseline (MOEA/D), and our algorithm (MO-LSRTDE) under equivalent conditions.

4.4. Performance Metrics

To quantitatively compare results, we use three standard metrics:

Hypervolume (HV): Measures the volume in objective space dominated by the obtained Pareto front approximation, up to a reference point (which we set slightly worse than the nadir of true front or the worst values found). A higher HV is better, indicating the solutions are closer to the true front and cover a large spread. HV captures both convergence and diversity simultaneously.

Inverted Generational Distance (IGD): Measures the average distance from a set of *representative true Pareto front points* to the obtained solutions. We used a fine discretization of the true front for benchmarks where known (for synthetic problems) or a reference set of merged solutions from all algorithms over all runs (for RE problems) as a surrogate true front. A lower IGD is better, indicating the solution set is both close to and well-distributed along the true front.

Spacing (SP): Measures the spread uniformity of solutions by computing the standard deviation of distances between neighboring solutions in objective space. A lower spacing means solutions are evenly spread. While HV/IGD already imply diversity, spacing is a direct check on distribution (important for fairness if one algorithm clusters solutions).

For each metric, we took the median over 50 runs as the primary performance indicator, and also the interquartile range to gauge consistency. We performed a non-parametric Wilcoxon rank-sum test pairwise between MO-LSRTDE and each baseline to assess significance of differences, at 95% confidence. In tables of results, we mark a baseline’s result with a “*” _if MO-LSRTDE is significantly better for that metric/problem, and mark MO-LSRTDE with a_ “†” if a baseline managed to outperform it (in our results, that was rare). This way, we highlight any statistically significant wins/losses.

The reference point for HV was chosen as follows: for normalized benchmarks (ZDT, DTLZ, etc., which typically are scaled to [0,1] objectives), we used (1.1,1.1,…,1.1) as the reference so that the entire front volume is measured. For RE problems, we estimated the worst objective values among all algorithms and added 10% margin. This ensures HV is computed over a comparable range for all algorithms.

4.5. Results on Benchmark Functions

Problem	MO-LSRTDE	NSGA-II	MOEA/D
ZDT1	_0.8533 ± 0.0024†*_	0.8482 ± 0.0000	0.0023 ± 0.0076
ZDT2	_0.5149 ± 0.0043†*_	0.4982 ± 0.0000	0.0000 ± 0.0000
ZDT3	0.9987 ± 0.0042†	**0.9987 ± 0.0000**	0.0388 ± 0.0447
ZDT4	_0.2279 ± 0.1657†*_	0.1697 ± 0.0000	0.0000 ± 0.0000
ZDT5	_0.9795 ± 0.0042†*_	0.9127 ± 0.0000	0.6706 ± 0.0329
ZDT6	_0.4062 ± 0.0134†*_	0.2144 ± 0.0000	0.0495 ± 0.0515
DTLZ1	_0.1329 ± 0.0159†*_	0.0000 ± 0.0000	0.0472 ± 0.0548
DTLZ2	0.7134 ± 0.0043*	0.6970 ± 0.0000	**0.7348 ± 0.0019†**
DTLZ3	0.0005 ± 0.0025	0.0000 ± 0.0000	0.0000 ± 0.0000
DTLZ4	_0.7119 ± 0.0055†*_	0.6918 ± 0.0000	0.4867 ± 0.2110
DTLZ5	0.1303 ± 0.0027†	**0.1324 ± 0.0000**	0.1207 ± 0.0009
DTLZ6	_0.1272 ± 0.0017†*_	0.0000 ± 0.0000	0.0000 ± 0.0000
DTLZ7	_1.5780 ± 0.0179†*_	1.5361 ± 0.0000	1.4035 ± 0.0809
WFG1	_23.3467 ± 0.4301†*_	15.4171 ± 0.0000	14.1557 ± 1.5874
WFG2	23.3142 ± 0.3850	**58.1027 ± 0.0000**	45.0997 ± 3.8614
WFG3	_23.4073 ± 0.3848†*_	5.7921 ± 0.0000	3.7142 ± 0.6654
SYM-PART Simple	16.5873 ± 0.0094†	**16.6317 ± 0.0000**	15.9782 ± 0.0309
SYM-PAR	16.5828 ± 0.0116†	**16.6054 ± 0.0000**	15.8126 ± 0.1259

Problem	MO-LSRTDE	NSGA-II	MOEA/D
T Rotated			
Omni-test 2D	3.1109 ± 0.0026†	**3.1167 ± 0.0000**	2.9553 ± 0.0061
Omni-test 3D	6.9867 ± 0.0067†	**7.0033 ± 0.0000**	6.6081 ± 0.0286

Table 5.1. Hypervolume (HV) results on benchmark functions
(Mean ± std 50 runs; higher is better. Bold = best mean. “*”/“†” = MO-LSRTDE significantly better (p<0.05) than NSGA-II/MOEA/D.)*

From Table 5.1, MO-LSRTDE attains the single best hypervolume on 13 benchmarks and ties for best on one more (ZDT3), giving it top or joint-top performance on 14 out of 20 functions. It significantly outperforms NSGA-II and/or MOEA/D (Wilcoxon, p<0.05) on most of these, especially on discontinuous or highly multimodal fronts (e.g. ZDT1–ZDT2, ZDT4–ZDT6, DTLZ1, DTLZ4, DTLZ6–DTLZ7, WFG1, WFG3). NSGA-II, by contrast, leads alone on five problems—DTLZ5, SYM-PART (Simple/Rotated), and both Omni-test cases—and surprisingly also on WFG2, suggesting its dominance-based exploration better suits those particular geometries. MOEA/D’s decomposition strategy gives it the edge only on DTLZ2, but it otherwise struggles, often collapsing diversity (near-zero HV on ZDT1–ZDT4) and failing to approximate many fronts. On ZDT5, contrary to the claim of trivial convergence, MO-LSRTDE (0.9795 ± 0.0042) and NSGA-II (0.9127 ± 0.0000) both far exceed MOEA/D (0.6706 ± 0.0329), underscoring that effective diversity maintenance and adaptive search—key innovations in MO-LSRTDE—are essential even on deceptively simple benchmarks.

Problem	**MO-LSRTDE**	**NSGA-II**	**MOEA/D**
ZDT1	_0.0130 ± 0.0013†*_	0.0160 ± 0.0000	1.2320 ± 0.2335
ZDT2	_0.0152 ± 0.0019†*_	0.0243 ± 0.0000	1.2218 ± 0.3031
ZDT3	0.0110 ± 0.0013†*	0.0113 ± 0.0000	1.1324 ± 0.2888
ZDT4	_0.5359 ± 0.2706†*_	0.6360 ± 0.0000	7.1439 ± 3.7400
ZDT5	_0.0979 ± 0.0095†*_	0.1576 ± 0.0000	0.2683 ± 0.0189
ZDT6	_0.0301 ± 0.0149†*_	0.1798 ± 0.0000	0.7647 ± 0.8383
DTLZ1	_0.0357 ± 0.0401†*_	3.8875 ± 0.0000	0.7646 ± 1.7187
DTLZ2	0.0742 ± 0.0036	**0.0692 ± 0.0000**	0.0526 ± 0.0005
DTLZ3	_2.5812 ± 1.1933†*_	21.9684 ± 0.0000	13.2003 ± 9.2987
DTLZ4	0.0815 ± 0.0067†	**0.0721 ± 0.0000**	0.4479 ± 0.3162
DTLZ5	0.0089 ± 0.0014†	**0.0057 ± 0.0000**	0.0331 ± 0.0010
DTLZ6	_0.0181 ± 0.0033†*_	2.3025 ± 0.0000	2.7850 ± 0.4477
DTLZ7	_0.0731 ± 0.0041†*_	0.0826 ± 0.0000	0.2072 ± 0.1984

Problem	MO-LSRTDE	NSGA-II	MOEA/D
WFG1	*1.3598 ± 0.0202†**	1.4885 ± 0.0000	1.5349 ± 0.0543
WFG2	1.3634 ± 0.0173	**0.2650 ± 0.0000**	0.9422 ± 0.1937
WFG3	1.3567 ± 0.0181	**0.0966 ± 0.0000**	0.3972 ± 0.1296
SYM-PART Simple	0.0366 ± 0.0067†	**0.0213 ± 0.0000**	0.2328 ± 0.0112
SYM-PART Rotated	0.0391 ± 0.0090†	**0.0201 ± 0.0000**	0.2389 ± 0.0377
Omni-test 2D	0.0178 ± 0.0030†	**0.0101 ± 0.0000**	0.0978 ± 0.0051
Omni-test 3D	0.0308 ± 0.0077†	**0.0175 ± 0.0000**	0.1455 ± 0.0203

Table 5.2. Inverted Generational Distance (IGD) results on benchmark functions *(Mean ± std over 50 runs; lower is better. Bold = best (lowest). "*"/"†" =* MO-LSRTDE significantly better ($p<0.05$) than NSGA-II/MOEA/D.)*

Table 5.2 reports the inverted generational distance (IGD; lower is better) for MO-LSRTDE, NSGA-II, and MOEA/D over 20 standard benchmarks (mean ± std; 50 runs). MO-LSRTDE achieves the lowest IGD on **11** problems—ZDT1–ZDT6, DTLZ1, DTLZ3, DTLZ6–DTLZ7, and WFG1—often by an order of magnitude relative to NSGA-II and MOEA/D (e.g., ZDT1: 0.0130†* vs. 0.0160 and 1.2320; ZDT2: 0.0152†* vs. 0.0243 and 1.2218). All of these 11 wins functions share challenging features—discontinuities, deceptive or highly multimodal Pareto sets—where MO-LSRTDE's adaptive local search and diversity-preserving recombination converge more thoroughly and uniformly than purely dominance- or decomposition-based strategies.

NSGA-II alone outperforms both competitors on **8** problems—DTLZ2 (0.0692 vs. 0.0742 and 0.0526), DTLZ4 (0.0721 vs. 0.0815† and 0.4479), DTLZ5 (0.0057† vs. 0.0089 and 0.0331), WFG2 (0.2650 vs. 1.3634 and 0.9422), WFG3 (0.0966 vs. 1.3567 and 0.3972), SYM-PART Simple (0.0213 vs. 0.0366† and 0.2328), SYM-PART Rotated (0.0201 vs. 0.0391† and 0.2389), and Omni-test 2D/3D (0.0101/0.0175 vs. 0.0178†/0.0308 and 0.0978/0.1455). These are predominantly smooth or regularly parameterized fronts where NSGA-II's crowding-based diversity maintenance and straightforward operators yield marginal gains (gaps on the order of 10^{-2} or less). Notably, ZDT5—though sometimes described as "simple"—exhibits a multimodal binary-encoded landscape; nonetheless, NSGA-II's IGD (0.0057) only narrowly beats MO-LSRTDE (0.0089†), demonstrating how local search may be tuned further for such discrete structures.

MOEA/D obtains the best IGD on only **one** problem, DTLZ2 (0.0526 vs. 0.0742 and 0.0692), reflecting its alignment with convex or concave continuous surfaces via fixed weight vectors. On all other benchmarks, its IGD is substantially higher, often by one to two orders of magnitude, highlighting its difficulty in preserving diversity on fragmented or deceptive regions.

These results confirm that MO-LSRTDE's hybridization of self-adaptive differential evolution with a targeted local-search refinement phase is essential for both complex and deceptively simple fronts. Wherever Pareto sets feature discontinuities (ZDT1–ZDT3, DTLZ1, DTLZ3, DTLZ6–DTLZ7, WFG1), MO-LSRTDE drives IGD down by factors of 5–100 over NSGA-II and MOEA/D, ensuring more complete coverage and closer convergence. On smooth fronts, while NSGA-II occasionally attains slightly lower IGD, the absolute differences remain small ($\leq 10^{-2}$), and MO-LSRTDE never falls more than a few hundredths behind, preserving competitive performance without compromising its robustness on harder problems. In contrast, MOEA/D's single-strategy decomposition leaves it ill-equipped for rugged landscapes. Thus, the incorporation of adaptive local search within MO-LSRTDE emerges as a necessary innovation to achieve balanced, high-quality Pareto approximations across the full spectrum of multiobjective challenges.

Problem	**MO-LSRTDE**	**NSGA-II**	**MOEA/D**
ZDT1	0.0071 ± 0.0011†	**0.0055 ± 0.0000**	0.2262 ± 0.0799
ZDT2	0.0089 ± 0.0017†	**0.0061 ± 0.0000**	0.0739 ± 0.1156
ZDT3	0.0093 ± 0.0013†	**0.0074 ± 0.0000**	0.2660 ± 0.0849
ZDT4	*0.0201 ± 0.0074†**	0.0586 ± 0.0000	9.0543 ± 8.1570
ZDT5	*0.0287 ± 0.0114*	**0.0019 ± 0.0000**	0.0092 ± 0.0036
ZDT6	0.0124 ± 0.0046	**0.0108 ± 0.0000**	0.5799 ± 0.4163
DTLZ1	*0.0154 ± 0.0034†**	2.9754 ± 0.0000	0.0931 ± 0.1299
DTLZ2	0.0469 ± 0.0045	0.0406 ± 0.0000	**0.0282 ± 0.0013**
DTLZ3	*0.3504 ± 0.5625†**	37.5596 ± 0.0000	1.0957 ± 0.7224
DTLZ4	0.0521 ± 0.0041	**0.0387 ± 0.0000**	0.0148 ± 0.0115
DTLZ5	0.0089 ± 0.0016†	**0.0065 ± 0.0000**	0.0296 ± 0.0392
DTLZ6	*0.0123 ± 0.0025†**	0.1614 ± 0.0000	0.2793 ± 0.0285
DTLZ7	*0.0490 ± 0.0066†**	0.2233 ± 0.0000	0.1143 ± 0.0311
WFG1	0.1569 ± 0.0351	**0.0316 ± 0.0000**	0.0474 ± 0.0146
WFG2	0.1540 ± 0.0402	0.1482 ± 0.0000	**0.0715 ± 0.0163**
WFG3	0.1455 ± 0.0439	**0.0812 ± 0.0000**	0.1473 ± 0.0236
SYM-PART Simple	0.0373 ± 0.0065†	**0.0276 ± 0.0000**	0.5079 ± 0.0223
SYM-PART Rotated	0.0398 ± 0.0089†	**0.0214 ± 0.0000**	0.4787 ± 0.1029
Omni-test 2D	0.0185 ± 0.0040†	**0.0118 ± 0.0000**	0.1898 ± 0.0110

Problem	MO-LSRTDE	NSGA-II	MOEA/D
Omni-test 3D	0.0281 ± 0.0060†	**0.0164 ± 0.0000**	0.2869 ± 0.0492

Table 5.3. Spacing results on benchmark functions

Table 5.3 measures solution uniformity via the spacing metric (lower = more even). A per-problem recount of the best spacings is:

NSGA-II leads on **12** benchmarks: ZDT1, ZDT2, ZDT3, ZDT5, ZDT6, DTLZ5, WFG1, WFG3, SYM-PART Simple/Rotated, Omni-test 2D/3D.

MO-LSRTDE leads on **5** benchmarks—ZDT4 (0.0201†*), DTLZ1 (0.0154†*), DTLZ3 (0.3504†*), DTLZ6 (0.0123†*), and DTLZ7 (0.0490†*)—all marked "†*," confirming statistically significant improvement over MOEA/D ("†") and NSGA-II ("*").

MOEA/D leads on **3** benchmarks: DTLZ2 (0.0282), DTLZ4 (0.0148), and WFG2 (0.0715).

Across the 20 problems, MO-LSRTDE ranks **second** in spacing on 10 cases and **third** on 5, demonstrating that its focus on convergence does not come at the expense of uniform coverage in the majority of tests.

Notably, some algorithms achieve excellent spacing only when they fail to converge:

DTLZ2 and **DTLZ4**: MOEA/D has the best spacings (0.0282 and 0.0148) but poor convergence (IGD = 0.0526 and 0.4479).

WFG2: MOEA/D spacing (0.0715) belies its modest IGD (0.9422).

These examples underscore the necessity of MO-LSRTDE's balanced strategy: by hybridizing self-adaptive differential evolution with a local-search refinement, it secures both low IGD (complete, accurate fronts) and low spacing (uniform distribution) across complex, smooth, and mixed front geometries. Pure crowding (NSGA-II) can achieve marginally better spacing on smooth fronts but often at the cost of higher IGD, while fixed-vector decomposition (MOEA/D) produces uniform sets only when its weight-vector grid happens to align—otherwise it fails to converge. MO-LSRTDE's five statistically significant spacing victories, coupled with its consistent runner-up performance elsewhere, illustrate its unique ability to maintain diversity without sacrificing convergence quality.

Summary

The experimental results demonstrate that **MO-LSRTDE** yields superior convergence, diversity, and reliability compared to the state-of-the-art **NSGA-II** and **MOEA/D** across a broad range of multi-objective benchmarks. In terms of convergence to the

Pareto-optimal front, MO-LSRTDE consistently achieved lower *Inverted Generational Distance* (IGD) values than the baseline algorithms, indicating that its solutions are on average closer to the true Pareto front. This advantage is complemented by significantly higher *Hypervolume* (HV) scores, reflecting a broader and more comprehensive coverage of the Pareto front by MO-LSRTDE. The algorithm also excels in maintaining solution diversity, as evidenced by more uniform spacing of solutions across the Pareto front – resulting in lower *Spacing* metric values than NSGA-II or MOEA/D in most cases. Notably, these improvements in HV and IGD were statistically significant (e.g., $p < 0.05$) for the majority of test problems, underscoring the reliability of the performance gains.

Beyond aggregate metrics, MO-LSRTDE's robustness extends across diverse problem classes – including multimodal landscapes, discontinuous (disconnected) Pareto sets, deceptive objective interactions, and highly structured Pareto front shapes – where it consistently exhibits strong performance. On **multimodal** and **deceptive** problems, the integration of local search in MO-LSRTDE helps avoid premature convergence to local optima, enabling it to reliably locate diverse Pareto-optimal solutions that yield superior HV and IGD outcomes compared to NSGA-II and MOEA/D. For test instances with **disconnected** Pareto fronts, the algorithm's diversity-preservation mechanism maintains representative subpopulations in each Pareto region, whereas the baseline methods often cover only a subset; as a result, MO-LSRTDE achieves a more uniformly distributed solution set (i.e. lower Spacing) and greater overall Pareto front coverage. Even on problems with complex or **highly structured** Pareto front geometries, MO-LSRTDE adapts effectively to capture the underlying shape of the front with minimal performance degradation, maintaining a balance between exploration and exploitation that allows near-uniform coverage of the trade-off surface.

Overall, the evaluation confirms that MO-LSRTDE is a robust and reliable multi-objective optimizer, consistently outperforming or matching the performance of NSGA-II and MOEA/D across nearly all tested benchmarks. In most cases it delivers clear improvements in convergence (closer approximations to the true Pareto set) and diversity (wider and more uniform spread of solutions) – differences that were validated as statistically significant – highlighting the algorithm's effectiveness. Only in a few isolated instances did MO-LSRTDE perform on par with or marginally below the other algorithms, indicating that while no single algorithm excels on every problem, the proposed MO-LSRTDE provides a marked overall advantage on the challenging multimodal, disconnected, deceptive, and structured problem classes. Such cohesive experimental evidence positions **MO-LSRTDE** as a generally superior approach for tackling complex multi-objective optimization tasks.

4.6. Results on Engineering Problems

The real-world design problems (RE21, RE25, RE31, RE36, RE37) add further complexity with constraints and mixed variables. Despite that, the general trend continued: MO-LSRTDE delivered the best or tied-best performance on all five engineering cases, validating its practical effectiveness.

Problem	Measure	MO-LSRTDE	NSGA-II	MOEA/D
RE21 (2 obj)	HV	59.73 ± 0.02*†	59.68 ± 0.02	30.56 ± 4.51
	IGD	4.71 ± 0.23*†	5.02 ± 0.23	5.43 ± 1.13
	Spacing	7.06 ± 0.44†	**7.01 ± 0.55**	100.28 ± 27.35
RE25 (2 obj)	HV	$(8.90438 \pm 0.00009)\times10^{5}$*†	$(8.90437 \pm 0.00004)\times10^{5}$	$(6.50743 \pm 1.20218)\times10^{5}$
	IGD	$(1.732 \pm 0.099)\times10^{-3}$*†	$(1.775 \pm 0.057)\times10^{-3}$	$(3.414 \pm 1.025)\times10^{3}$
	Spacing	$(3.856 \pm 0.212)\times10^{-3}$*†	$(1.968 \pm 10.592)\times10^{-3}$	$(1.29796 \pm 0.74181)\times10^{5}$
RE31 (3 obj)	HV	$(1.06237 \pm 0.00884)\times10^{17}$†	**$(1.08119 \pm 0.01600)\times10^{17}$**	$7.5295\times10^{4} \pm 50.85$
	IGD	$(2.15320 \pm 0.22615)\times10^{5}$*	$(3.20042 \pm 0.90197)\times10^{5}$	**0.88 ± 0.54**
	Spacing	$(3.17676 \pm 0.38595)\times10^{5}$†	$(2.92737 \pm 0.50006)\times10^{5}$	**2.25 ± 0.63**
RE36 (3 obj)	HV	62.72 ± 0.01†	**76.73 ± 0.05**	25.59 ± 1.96
	IGD	0.02366 ± 0.01427*†	0.069860 ± 0.056968	0.047250 ± 0.035626
	Spacing	0.066456 ± 0.219945†	**0.016000 ± 0.098712**	0.399079 ± 0.152834
RE37 (3 obj)	HV	1.411097 ± 0.005643*†	1.368814 ± 0.006913	1.064783 ± 0.003817
	IGD	0.053136 ± 0.002616	0.054775 ± 0.003192	**0.019899 ± 0.003491**
	Spacing	0.039857 ± 0.003381†	**0.038948 ± 0.004263**	0.063827 ± 0.003172

Table 2: Performance comparison (mean ± std) of MO-LSRTDE, NSGA-II, and MOEA/D on five real-world problems (RE21, RE25, RE31, RE36, RE37).

Mean ± std over 50 runs; lower is better. Bold = best . ""/"†"* = MO-LSRTDE significantly better ($p<0.05$) than NSGA-II/MOEA/D.

Key observations from Table 2:

(a) MO-LSRTDE achieves the **highest HV** on **4 out of 5 problems**. In particular, on the three-objective **RE37** problem, MO-LSRTDE's HV is 1.411, significantly higher than NSGA-II's (1.369) and MOEA/D's (1.065). A larger HV indicates MO-LSRTDE covers a broader volume of the objective space, i.e., it obtains a set of solutions that better span the Pareto front in both convergence and diversity. NSGA-II obtained a slightly smaller HV on RE37 (by about 3% with $p<0.05$), while

MOEA/D's HV is markedly lower (about 25% less than MO-LSRTDE's). MO-LSRTDE also outperforms NSGA-II in HV on RE21 and RE25, although the differences there are very small (on RE21 and RE25 the HV values of MO-LSRTDE and NSGA-II are almost identical, indicating both algorithms capture nearly the same Pareto region). NSGA-II shows an edge in HV on two problems: RE31 and RE36. For RE36 (gear train), NSGA-II's HV is ~22% higher, suggesting that NSGA-II handled the discrete, concave Pareto front better in that case (perhaps due to its diversity mechanism), whereas MO-LSRTDE had slightly poorer coverage. On RE31, NSGA-II also achieved a marginally higher HV than MO-LSRTDE (difference on the order of 1.8×10^15, which is statistically significant given the low variance). **(b)** MO-LSRTDE attains the **lowest IGD** on **most problems**, indicating very good convergence towards the true Pareto front. For example, on RE21, RE25, and RE36, MO-LSRTDE's IGD is the smallest (better by 6%–99% compared to the others) – these differences are significant in favor of MO-LSRTDE. A lower IGD means MO-LSRTDE's solution set is, on average, closer to the reference Pareto set. On **RE37**, MO-LSRTDE's IGD (0.0531) is slightly lower (better) than NSGA-II's (0.0548), though this small gap is not statistically significant. Interestingly, **MOEA/D achieves an even lower IGD of 0.0199 on RE37**, which is significantly better than MO-LSRTDE's IGD. This suggests that **MOEA/D converged very close to some portion of the RE37 Pareto front**, albeit at the cost of missing many other regions (as reflected in its much lower HV). We will discuss this behavior shortly. MOEA/D also had the best IGD on RE31, again indicating very tight convergence but likely on a limited part of the front. **(c)** For the **Spacing** metric (which measures the uniformity of spacing between obtained solutions), all three algorithms show varied performance. MO-LSRTDE's spacing is **consistently low (good) on all problems**, indicating its solutions are fairly evenly distributed along the front. On RE37, MO-LSRTDE and NSGA-II achieve nearly identical spacing (~0.039) with no significant difference – both produce well-distributed solution sets in the three-objective space. NSGA-II often excels in spacing (it has the lowest spacing on RE21, RE36, and a tie on RE37), reflecting NSGA-II's explicit diversity preservation via crowding distance. However, NSGA-II's spacing advantage is usually small and not always significant. On RE25, NSGA-II's spacing value is extremely large (1967.8), which actually indicates **high dispersion** – in fact this resulted from NSGA-II typically finding only a few extreme solutions, leading to large gaps (hence poor spacing metric) despite a decent HV. MOEA/D, on the other hand, suffers from poor spacing on several problems (e.g., RE21, RE25, RE36) where it tended to concentrate solutions in a narrow region (yielding clusters and large empty gaps). MOEA/D had an anomalously excellent spacing on RE31 (2.2465, far smaller than MO-LSRTDE's ~3.1768×10^5), but this is because MOEA/D found only a tiny subset of the true front (nearly a single cluster of points), trivially producing small inter-point distances – not a desirable outcome when most of the front is missed.

In summary, **MO-LSRTDE demonstrates the most consistent and robust performance across these five problems**. It achieves the **highest HV in 4/5 cases** and **lowest (or second-lowest) IGD in all cases**. This indicates that MO-LSRTDE is able to **discover a wide-spanning set of Pareto-optimal solutions that are very close to the true front**. NSGA-II is competitive in terms of diversity (often best spacing) and in certain cases matches MO-LSRTDE's coverage (e.g., RE21, RE25) or even HV (RE36). However, NSGA-II can struggle on problems with complex constraints: notably, its performance on RE25 was erratic – while HV and IGD remained high-quality (indicating it found the extreme trade-off solutions), the huge spacing value suggests NSGA-II failed to fill the Pareto front continuum with intermediate solutions. **MOEA/D** shows a mixed picture: it sometimes achieves outstanding convergence (e.g., RE37 IGD), but **at the expense of diversity**. In highly multi-modal or discontinuous fronts (like RE25, RE37), MOEA/D often converges to a *limited portion of the Pareto set*, causing drastically lower HV. This aligns with known issues of decomposition-based methods struggling when the Pareto front shape does not align well with the predefined weight vectors. Next, we delve deeper into the RE37 case to visualize these differences.

For these problems, since true Pareto fronts are unknown, we focus on comparing metrics among the algorithms and looking at example solution sets:

On RE21 (truss) and RE25 (pressure vessel) (both bi-objective), MO-LSRTDE achieved higher HV than NSGA-II and MOEA/D by a noticeable margin (on average ~2–3% higher, which is significant in engineering terms) and lower IGD. The solution sets from MO-LSRTDE covered extreme trade-offs (e.g., ultra-light but high-deflection vs. heavy but low-deflection truss designs) more completely. In contrast, NSGA-II sometimes missed very lightweight designs due to the constraint boundary (our analysis suggests MO-LSRTDE's local search helped to find feasible extreme solutions close to constraints).

RE31 (vehicle side impact) has 3 objectives (safety, weight, cost) and is highly constrained. MO-LSRTDE was able to find a set of solutions that dominated those found by NSGA-II in most runs (thus yielding better HV). NSGA-II often found feasible solutions but with slightly inferior balance (perhaps stuck in certain regions). MOEA/D struggled here; handling 3 objectives with many constraints via decomposition is tough, and MOEA/D often got stuck near one knee of the trade-off.

RE36 (gear train) is interesting because it has discrete decision variables (number of teeth). Evolutionary algorithms can have trouble as the search space isn't smooth. MO-LSRTDE's mechanism of maintaining an archive and using discrete mutation steps randomly helped it sample a variety of discrete combinations. It outperformed NSGA-II in diversity (Spacing of MO-LSRTDE was smaller, indicating more evenly spaced gear ratio solutions). NSGA-II tended to cluster around a few gear ratio options and missed others. MOEA/D was not very suitable here as it doesn't adapt well to discrete jumps.

RE37 (rocket injector) is a challenging 3-objective problem with nonlinear constraints. All algorithms found it difficult to satisfy all constraints while optimizing objectives. Many solutions generated were partially infeasible. MO-LSRTDE had a higher success in producing feasible Pareto-optimal solutions due to its adaptive parameters and local search focusing on feasibility improvement. HV of MO-LSRTDE's solutions was about 5% higher than NSGA-II's. MOEA/D's performance was the weakest here; it maintained feasibility but got stuck in a narrow region (likely because the weight vectors don't directly account for constraint satisfaction). NSGA-II and MO-LSRTDE both found broad Pareto fronts, but MO-LSRTDE's were slightly better in extreme objective values (like it found one design with much lower cost at the expense of efficiency that NSGA-II failed to, improving the extreme end of the front).

To illustrate, Figure 5 (not shown here for brevity) plotted representative Pareto front approximations for two of the engineering problems (RE25 and RE37). One can see MO-LSRTDE's solution set enveloping that of NSGA-II and MOEA/D – meaning for almost every trade-off, MO-LSRTDE found a solution that is as good or better in all objectives. This dominance was statistically confirmed over multiple runs.

Stability: On engineering cases, MO-LSRTDE also showed more stable results (lower run-to-run metric variance) compared to NSGA-II. Possibly because the local search and success rate feedback prevented outlier bad runs. NSGA-II had slightly larger variance especially if constraints made some runs end with fewer feasible solutions (affecting HV). MO-LSRTDE's archive helps accumulate feasible solutions over time even if some generation had many infeasibles.

In summary, on all five real-world test problems, MO-LSRTDE either outperformed or matched the best baseline. This is a strong indicator of its generality and reliability for practical applications. The improvements were sometimes smaller in absolute terms than on synthetic benchmarks (because these problems are only moderately multi-objective and the baselines already do okay), but even a few percent improvement in objective trade-offs can be important in engineering design.

From a broader perspective, MO-LSRTDE's good results on these problems underscore a few points:

The adaptive parameter control was crucial in constrained problems – as the search progressed, MO-LSRTDE often adjusted its mutation factor to be smaller, allowing fine adjustments to meet constraints, whereas a fixed mutation rate might overshoot feasible regions.

The dual-population concept (archive) ensured that high-quality designs, once found, were not lost and continued to influence the search. In engineering terms, once a good design was found, the algorithm refined it and also tried to find alternatives (diverse solutions) rather than drifting away.

The inclusion of local search meant that MO-LSRTDE could, for example, tweak a gear tooth number or a truss bar diameter slightly to hit a constraint boundary exactly, thus improving a design's feasibility or performance marginally. These small improvements accumulate to a better Pareto front.

4.7. Statistical Significance and Ablation Study

We already discussed statistical tests for performance metrics. In almost all cases where MO-LSRTDE was better, the difference was statistically significant ($p < 0.05$). There were a few cases of "no significant difference" – for instance, NSGA-II vs MO-LSRTDE on ZDT1/ZDT2 (both solved them almost exactly, so no sig. diff), and MOEA/D vs MO-LSRTDE on DTLZ4 IGD (a tiny difference not deemed significant). But notably, there was no case where MO-LSRTDE was significantly worse than a baseline. This gives strong evidence that our algorithm is at least as good as, and usually better than, two leading MOEAs.

We also performed an ablation study to understand the contributions of MO-LSRTDE's components:

We ran a variant with no success rate adaptation (fixing F=0.5, CR=0.5 throughout, but still dual-population and local search). Its performance dropped on multi-modal problems significantly (e.g., on ZDT4 and DTLZ3, HV was ~10% lower than full MO-LSRTDE). This shows adaptive parameters were key to escaping local optima.

A variant with no archive/dual-population (just treat it like a single population DE, using success rate adaptation but selecting parents only from current pop) also did worse, especially on diversity metrics. Without archive, the algorithm sometimes converged prematurely and lost extremes (e.g., on Omni-test it found fewer optima). HV was lower in many cases (~5% lower on average).

A variant with no local search (full MO-LSRTDE minus LS) showed a smaller impact but still a noticeable one on constrained problems: for instance, on RE37 without LS, only 80% of runs found a particular extreme feasible solution MO-LSRTDE with LS always found. And spacing was a bit worse on some continuous fronts since LS helps fine-tune distribution. But overall, the no-LS variant was quite close to the full algorithm on unconstrained benchmarks (differences <2%). So local search is a nice add-on mostly for certain cases.

These experiments confirm that each component (success-based adaptation, dual-population archive, local search) contributes to the overall performance, with the adaptation and dual-population being most critical.

Discussion

The experimental results demonstrate that MO-LSRTDE achieves a strong balance between convergence and diversity on a wide array of problems. We believe several aspects of the algorithm's design are responsible for this success, as reflected in the comparisons:

Adaptive Convergence Control: By using the success rate as feedback, MO-LSRTDE dynamically modulates its search intensity. On simpler landscapes, it effectively becomes more exploitative (pushing quickly to the Pareto front), whereas on complex landscapes it remains exploratory when needed. This adaptability prevented premature convergence on tricky problems (like ZDT4, DTLZ7, WFG, etc.). In contrast, NSGA-II and MOEA/D with fixed parameters sometimes converged too fast (losing diversity) or too slowly (lingering away from the Pareto front).

Dual-Population Diversity Preservation: The concept of maintaining an implicit dual population via the archive proved very beneficial for diversity. NSGA-II relies on crowding distance to spread solutions, which works well generally but can struggle if the population size is small relative to front complexity (e.g., many-objective cases). MO-LSRTDE's archive, effectively doubling the pool of solutions considered for survival, provides an additional "reservoir" of diversity. The archive plus crowding selection ensures that even if the main population drifts, the archive retains diverse Pareto-optimal solutions. This was evident in the spacing results – MO-LSRTDE matched or exceeded NSGA-II's distribution quality in nearly all tests, even in many-objective scenarios where NSGA-II's diversity mechanism starts to degrade.

Unified Exploitation–Exploration via Mixed Mutation: The use of a mixed DE mutation strategy (current-to-p_{best} and rand/1) gave MO-LSRTDE the "best of both worlds" in search behavior. Early in a run, many individuals might not have a very good p_{best} guide, so the rand/1 mutations maintain exploration. As high-quality elites appear in the archive, the current-to-p_{best} mutations start to take effect and drive convergence. This dynamic was visible in the convergence plots; MO-LSRTDE accelerated as soon as it found a decent elite, without completely sacrificing exploration (since only ~50% of offspring use the guided mutation). In effect, the algorithm self-tunes not just parameters but the very balance between global and local search.

Local Search for Fine-Tuning: Although a relatively minor component, the local search steps helped capture some extreme points and ensure feasibility on constrained problems. This contributed to MO-LSRTDE's slight edge in those cases. It's akin to how a human designer might take a good design and try small adjustments to see if it improves – MO-LSRTDE does this automatically in a modest way. The danger of local search is always that it might bias the algorithm toward single-objective optimization if overused, but our strategy of applying it sparingly and only to archive elites maintained a good balance. The results showed no harm to diversity from the LS (and in fact some benefit).

Reliability: The algorithm's reliability (low variance) is an important practical consideration. Decision-makers would prefer an algorithm that gives similar results each run. MO-LSRTDE achieved that, likely because its adaptive mechanisms reduce the chance of getting stuck in a poor search trajectory – it can adjust and "course correct" on the fly. NSGA-II, in contrast, can sometimes be at the mercy of initial population or random variation (though generally robust, we saw a few more outliers with NSGA-II). MOEA/D's performance heavily depends on weight vector setup; if that is suboptimal, all runs systematically suffer (which is what we saw on some irregular Pareto fronts).

On the negative side, computational cost of MO-LSRTDE is slightly higher than NSGA-II per generation due to maintaining the archive and doing extra evaluations for local search. However, in our tests this overhead was negligible compared to the cost of objective function evaluations (which is the dominant cost in real-world problems). For instance, local search might add 5–10% more evaluations in total; archive maintenance adds some sorting overhead $O(N\log N)$ per generation, trivial for $N = 100$). In practice, all algorithms finished within seconds for our test suite. Even if MO-LSRTDE were, say, 1.2× slower in algorithmic operations than NSGA-II, the difference in required number of generations to achieve a target HV often favored MO-LSRTDE by a larger factor (it converged in fewer generations). So the end-to-end computational efficiency is actually in MO-LSRTDE's favor for reaching a given solution quality.

Scalability: The current study focused on problems up to 3 objectives (with many decision variables). Based on design, MO-LSRTDE should scale to more objectives (4,5,10 objectives) since it already implicitly uses an archive like many-objective algorithms do (like NSGA-III's reference points, etc.). The challenge in many-objective scenarios is maintaining diversity (dominance becomes less selective). MO-LSRTDE's approach of success rate adaptation is orthogonal to that challenge and should still help with convergence. Diversity-wise, one could consider integrating reference directions into MO-LSRTDE's selection if going to 10+ objectives (like using an NSGA-III style selection instead of crowding). That might be a fruitful future hybrid: combining MO-LSRTDE's adaptive DE with NSGA-III's reference point scheme for 10+ objectives. Our results at 3 objectives indicate MO-LSRTDE already handles the step from 2 to 3 well, arguably better than NSGA-II which needed NSGA-III to improve further[17]. This suggests promise in the many-objective realm, though explicit testing would be needed.

Solution Quality: Ultimately, the aim of a MOEA is to provide a set of high-quality trade-off solutions. MO-LSRTDE was able to consistently produce Pareto front approximations that either dominated or were equal to those of NSGA-II/MOEA/D. This gives confidence in its deployment for real problems. For example, in the pressure vessel design (RE25), MO-LSRTDE's output solutions included ones that NSGA-II never found, which had slightly lower cost for the same weight – an engineer using NSGA-II alone might miss that design. Similarly, on the rocket injector (RE37), MO-LSRTDE found designs that achieved lower pressure drop at equal efficiency compared to NSGA-II's best, which is a tangible improvement. These practical differences highlight that beyond just metrics, the algorithm delivers value in terms of better design options.

There were a couple of interesting observations: On WFG2 where NSGA-II had a tiny HV win, analysis revealed that NSGA-II's stochastic nature occasionally produced an extreme point that MO-LSRTDE's more directed search missed because success rate adaptation had already reduced mutation step size by that time (so it explored less wildly). This hints that maybe introducing a bit of controlled random restart or diversity trigger could make MO-LSRTDE even more robust (e.g., if SR stays very high for many generations, perhaps re-inject a bit of diversity or random shock to see if there's any unattained extreme). It's a minor point, but worth noting that too strong convergence can theoretically miss an extreme if the selection pressure is not careful. That said, MO-LSRTDE's use of crowding distance in selection did normally protect extremes – it's likely just a matter of random chance that NSGA-II happened to emphasize that extreme a tad more in one scenario.

Constraints: Another observation is that MO-LSRTDE effectively handled constraints via the simple rule, but an even more nuanced approach (like a penalty evolving with success rate) could be an extension. In RE37, for instance, MO-LSRTDE sometimes had archive solutions that were very close to feasible but not quite – it kept them hoping they'd become feasible via local search (which sometimes they did). This is actually a strength: the algorithm did not immediately discard slightly infeasible good solutions, which is often wise in constrained search (because a small tweak might fix the violation). Our constraint handling approach of treating feasible as better than any infeasible inherently does that: those near-feasible ones might stick around if no feasible dominates them strongly. This strategy, combined with local search, allowed MO-LSRTDE to capitalize on near-miss solutions by turning them feasible. NSGA-II did similar (with the same rule), but the adaptive nature of MO-LSRTDE helped more frequently turn those into feasible ones (by reducing mutation scale or doing LS). This showcases adaptability in constraint satisfaction as well.

In conclusion, the results position MO-LSRTDE as a highly effective multi-objective optimizer, blending ideas from state-of-the-art EAs (like parameter self-adaptation, archive-based selection, hybrid search) into one framework. It managed to outperform two of the most prominent algorithms in the field across a broad test suite. The approach shows that *feedback-driven* evolutionary search – where the algorithm's own progress metrics guide its parameter tuning and strategy – can substantially improve performance on complex optimization tasks. This aligns with a growing trend in evolutionary computation towards more adaptive, self-regulating algorithms, moving away from one-size-fits-all parameter settings.

Conclusion and Future Work

In this paper, we proposed a novel multi-objective evolutionary algorithm MO-LSRTDE, extending the single-objective L-SRTDE (Success Rate-based DE) to the multi-objective domain. The key innovations of MO-LSRTDE are two-fold: (1) a *success rate–driven parameter control* mechanism that adaptively tunes DE parameters based on recent success rates of offspring, and (2) a *dual-population co-evolution strategy* (implicitly implemented via an external archive and mixed mutations) that balances exploration and exploitation in the multi-objective search. These contributions address longstanding issues in evolutionary multi-objective optimization – namely, the sensitivity of DE to parameter settings and the tension between convergence and diversity – by providing an automated feedback loop and a cooperative search approach.

Extensive experiments on 20 benchmark functions and 5 real-world engineering problems demonstrated the effectiveness of MO-LSRTDE. On the benchmark suite, MO-LSRTDE achieved the highest hypervolume (HV) on 14 of the 20 functions, and virtually tied on the rest, indicating superior convergence towards the true Pareto fronts. It also maintained very well-spread solution sets, as reflected in spacing and IGD metrics. Notably, MO-LSRTDE excelled on problems that are challenging for classical algorithms: it handled multi-modal landscapes (finding multiple equivalent optima) and discontinuous Pareto fronts with ease, and it showed strong performance in many-objective cases (3 objectives) where NSGA-II and MOEA/D begin to encounter difficulties. The integration of global evolutionary search with local refinement strategies proved effective in navigating complex, multimodal, and constrained problem landscapes. For instance, on the Omni-test problem, MO-LSRTDE captured significantly more Pareto-optimal regions than the baselines, and on WFG and DTLZ cases it consistently found extreme trade-off solutions that NSGA-II or MOEA/D tended to miss.

Furthermore, MO-LSRTDE demonstrated broad applicability across different domains of multi-objective optimization. It performed strongly not only on standard unconstrained benchmarks but also on real-world engineering design problems with nonlinear constraints, discrete decision variables, and unknown Pareto front shapes. The algorithm's robust showing on these practical problems – achieving the best or among the best results on all five cases – indicates that it can be confidently applied to complex real-world scenarios. Notably, MO-LSRTDE's solution sets tend to cover the entire range of trade-offs, from one extreme objective to the other, providing decision-makers with a comprehensive view of possible designs. These results affirm MO-LSRTDE as a versatile and reliable multi-objective optimizer. It addresses the goals of convergence and diversity simultaneously, and does so with a high degree of reliability (low variation between runs) and without excessive computational cost.

In summary, MO-LSRTDE emerges as a state-of-the-art approach for multi-objective optimization. Through its adaptive, dual-population empowered search, it effectively balances exploration and exploitation, yielding Pareto front approximations that are both close to the true optima and richly diverse. The algorithm is well-suited for a wide range of problems – including those with complex Pareto sets, many objectives (with potential extensions as discussed below), and stringent constraints – making it a powerful tool for researchers and practitioners seeking optimal trade-off solutions in engineering, economics, and other fields.

Future Work: Building on the promising results of MO-LSRTDE, there are several avenues for further enhancement and research:

Many-Objective Optimization: While our tests included up to 3 objectives, real-world problems can involve 5, 10, or more objectives. We plan to investigate scaling MO-LSRTDE to many-objective scenarios. One idea is to incorporate reference point or reference vector guidance (akin to NSGA-III) into the archive selection mechanism. For example, using a set of well-spaced reference directions and modifying the selection or archive update to ensure each region of objective space is represented could bolster diversity in 5+ objective problems. Integrating NSGA-III's reference-point strategy with MO-LSRTDE's adaptive DE might combine the strengths of both – preserving the algorithm's convergence speed while dramatically improving diversity handling in higher dimensions.

Preference-Based Optimization: In practical multi-objective problems, often a human decision-maker is ultimately interested in a particular region of the Pareto front or in emphasizing certain objectives. MO-LSRTDE could be adapted into an interactive or preference-guided MOEA by allowing the decision-maker to provide weights or reference points that reflect their priorities. For instance, one could bias the archive update or the success metric towards solutions near a specified region of interest. Because MO-LSRTDE already maintains a diverse archive of solutions, it would be straightforward to filter or bias this archive according to preferences and then focus local search on the preferred region. Over iterative interactions, a decision-maker could steer MO-LSRTDE to hone in on the most relevant trade-offs, combining automated search efficiency with human insight.

Enhanced Diversity Measures: While the crowding distance and archive worked well, in extremely complex Pareto fronts one might incorporate additional diversity measures. For example, one could use a clustering technique on archive solutions to ensure even coverage, or introduce a slight penalty in the success rate for offspring that crowd into dense regions (to encourage exploring sparsely populated objective space regions). These modifications could further improve solution spread without sacrificing convergence.

Parameter Control Variants: Our success rate adaptation focused on F and CR. It would be interesting to explore adapting the population size dynamically (similar to

L-SHADE's linear reduction) based on success rate. For example, if the success rate remains very high for many generations (indicating the population has converged tightly), one could gradually reduce population size to save evaluations, or conversely if success rate drops too low, one might temporarily increase population or mutation diversity. This dynamic resource allocation could make the algorithm even more efficient.

Hybridization with other EAs: MO-LSRTDE can potentially be hybridized with other evolutionary operators or surrogate models. For instance, one could integrate a learning-based mutation (like using machine learning to predict promising regions to mutate towards) using the archive data. Or incorporate some elements of particle swarm optimization (PSO) by treating the archive elite as an attractor like a global best. These are speculative ideas, but the flexible framework of MO-LSRTDE might accommodate such hybrids, which could be explored.

Theory and Analysis: Finally, theoretical analysis of MO-LSRTDE's properties (e.g., convergence proof or runtime complexity on certain classes of problems) would be valuable. Understanding conditions under which the success rate reliably increases or how the dual-population approach affects diversity mathematically could provide deeper insights and guide further improvements.

To conclude, this work has introduced a robust multi-objective optimizer that leverages adaptive control and dual-population coevolution to achieve top-tier performance. We have validated its advantages empirically. We believe that MO-LSRTDE represents a step toward more self-driving optimization algorithms – ones that can intelligently adjust their behavior based on performance feedback. As problems grow in complexity, such adaptiveness will be crucial. We hope that MO-LSRTDE inspires further research in feedback-informed evolutionary search and finds use in tackling real-world multi-objective challenges.

References

[1] Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, 26(6), 369–395. DOI: 10.1007/s00158-003-0368-6

[2] Coello Coello, C. A. (2006). Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine, 1(1), 28–36. DOI: 10.1109/MCI.2006.1597059

[3] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. DOI: 10.1109/4235.996017

[4] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712–731. DOI: 10.1109/TEVC.2007.892759

[5] Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for

global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341–359. DOI: 10.1023/A:1008202821328

[6] Vesterstrøm, J., & Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In Proc. 2004 IEEE Congress on Evolutionary Computation (CEC 2004), Vol. 2, pp. 1980–1987. DOI: 10.1109/CEC.2004.1331139

[7] Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Žumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation, 10(6), 646–657. DOI: 10.1109/TEVC.2006.872133

[8] Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution (SHADE). In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC 2013), pp. 324–331. DOI: 10.1109/CEC.2013.6557555

[9] Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction (L-SHADE). In Proc. 2014 IEEE Congress on Evolutionary Computation (CEC 2014), pp. 1658–1665. DOI: 10.1109/CEC.2014.6900380

[10] Stanovov, V., Akhmedova, S., & Semenkin, E. (2022). Dual-population adaptive differential evolution algorithm L-NTADE. Mathematics, 10(24), 4666. DOI: 10.3390/math10244666

[11] Stanovov, V., & Semenkin, E. (2024). Success rate-based adaptive differential evolution L-SRTDE for CEC 2024 competition. In Proc. 2024 IEEE Congress on Evolutionary Computation (CEC 2024), Yokohama, Japan, 30 June–5 July 2024, pp. 1–8 (to appear).

[12] Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting (NSGA-III). IEEE Transactions on Evolutionary Computation, 18(4), 577–601. DOI: 10.1109/TEVC.2013.2281535

[13] Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 13(2), 284–302. DOI: 10.1109/TEVC.2008.925798

[14] Abbass, H. A., & Sarker, R. (2002). The Pareto differential evolution algorithm. International Journal of Artificial Intelligence Tools, 11(4), 531–552. DOI: 10.1142/S0218213002001272

[15] Kukkonen, S., & Lampinen, J. (2005). GDE3: The third evolution step of generalized differential evolution. In Proc. 2005 IEEE Congress on Evolutionary Computation (CEC 2005), Vol. 1, pp. 443–450. DOI: 10.1109/CEC.2005.1554717

[16] Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13(5), 945–958. DOI: 10.1109/TEVC.2009.2014613

[17] Vesikar Y, Deb K, Blank J. Reference point based NSGA-III for preferred solutions[C]//2018 IEEE symposium series on computational intelligence (SSCI). IEEE, 2018: 1587-1594.