Provable Speedups From Dynamic Population Sizes in Evolutionary Algorithms for Multiobjective Optimization

Andre Opris
Chair of Algorithms for Intelligent Systems
University of Passau
Innstraße 43, 94032 Passau
andre.opris@uni-passau.de

Abstract

This paper investigates the role of dynamic population sizes in evolutionary multi-objective optimization. Although such approaches are widely used in practice, their benefits remain poorly understood, and rigorous runtime analyses explaining when and why they help are still scarce. To address this, we introduce the bi-objective problem class CLIMB_ℓ, where the parameter ℓ determines the size of the Pareto front. On this problem class, we analyze the runtime of GSEMO and the widely used NSGA-II. Our results show that allowing a dynamic population size for NSGA-II can lead to a moderate improvement, yielding a super-constant speedup for $\ell \in \omega(\log(n)) \cap o(\sqrt{n\log(n)})$ in terms of fitness evaluations. In particular, we prove that GSEMO and NSGA-II-DYN, a version of NSGA-II with dynamic population sizes that we propose in this paper, can cover the Pareto front of CLIMB_ℓ in expected $O(n\log(n) + \ell^2\log(en/(n-2\ell+3)))$ evaluations, and that this bound is tight for $\ell = O(\sqrt{n\log(n)})$. In contrast, NSGA-II with a fixed population size of at least 4ℓ requires expected $O(n\ell + n\log(n) + \ell^2\log(en/(n-2\ell+3)))$ evaluations to cover the entire Pareto front of CLIMB_ℓ, while still requiring expected $\Omega(\ell n)$ evaluations to even find a Pareto-optimal search point. To the best of our knowledge, this is the first rigorous runtime analysis in bi-objective optimization that demonstrates a super-constant speedup of GSEMO over NSGA-II. Our analysis builds on concepts from single-objective optimization, such as the evolution of population diversity over time, and employs the well-known family-tree method to prove the lower bound. [1]

1 Introduction

Many real-world optimization problems are characterized by several, often conflicting objectives. These can be tackled by evolutionary multi-objective algorithms (MOEAs) which mimic principles from natural evolution to evolve a population of solutions to find a Pareto optimal set. A human decision maker may then select a solution that best matches their needs. Due to their population-based nature, MOEAs are among the most prominent approaches to such problems and have found applications in numerous practical domains like vehicle design [Xingtao et al., 2008], scheduling problems [Ishibuchi and Murata, 1998], or research fields like artificial intelligence [Luukkonen et al., 2023], and especially machine learning [Zhang et al., 2025, Mehri et al., 2025, Chowdhury et al., 2026] and neural networks [Alizadeh et al., 2025, Martínez-Comesaña et al., 2023]. The most prominent MOEA is the NSGA-II [Deb et al., 2002] (with around 65,000 citations). In each iteration

[1] A long version of that paper can be found on archive Opris [2026b]

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

(or generation), NSGA-II forms a combined parent–offspring population, which is then partitioned into layers of non-dominated fitness values for survival selection. The first layer contains only non-dominated individuals, while the i-th layer consists only of those individuals which are dominated by an individual with rank in $\{1, \dots, i-1\}$. Individuals are first selected according to their layer rank and ties are broken using crowding distance as a secondary criterion. In a nutshell, the algorithm favors solutions in less crowded regions of the objective space among individuals of equal rank. It has been shown that this algorithm optimizes bi-objective problems very efficiently (see [Köppen and Yoshida, 2007] for empirical results or [Zheng et al., 2022, Doerr and Qu, 2022, Dang et al., 2024] for rigorous runtime analyses). Despite these extensive studies, a complete theoretical understanding of when and why the algorithm performs well is still lacking, particularly from the perspective of runtime analysis. This is somewhat surprising, given that the mathematical analysis of MOEAs began more than 20 years ago with results on the (global) simple evolutionary multiobjective optimizer ((G)SEMO) [Laumanns et al., 2002, Giel, 2003, Thierens, 2003]. GSEMO is particularly simple, as it relies solely on Pareto dominance for selection and generates only a single offspring per generation. Due to its simplicity, it was the first MOEA for which rigorous runtime analyses were carried out, and it remains a central algorithm in which new phenomena are first observed (see, e.g., [Dinot et al., 2023, Dang et al., 2023, Doerr et al., 2025b], as well as applications to submodular optimization [Qian et al., 2015, 2018]). Building on these insights, similar analyses have recently been extended to other widely used MOEAs, including NSGA-II, NSGA-III, SMS-EMOA, and SPEA2 (see [Zheng et al., 2022, Wietheger and Doerr, 2023, Opris et al., 2024, Opris, 2025a, Zheng and Doerr, 2024, Ren et al., 2024, Opris, 2026d] for breakthroughs and recent analyses). A key difference between GSEMO and these algorithms is that the latter use a fixed population size. In contrast, a dynamic population size has the potential to avoid unnecessary fitness evaluations on low-quality individuals and thereby accelerate the optimization process. While empirical studies already indicate advantages of dynamic population sizes in evolutionary multi-objective optimization [Wang et al., 2022, Tian et al., 2025, Liang et al., 2023], to the best of our knowledge, there is no corresponding result from the perspective of runtime analysis apart from [Doerr et al., 2026b]. In particular, we are not aware of any benchmark problem on which GSEMO provably outperforms these popular MOEAs. Instead, most existing results suggest that, in terms of fitness evaluations, GSEMO and these algorithms achieve the same runtime guarantees [Laumanns et al., 2004, Zheng et al., 2022]. These insights may guide practitioners in designing improved variants of NSGA-II, NSGA-III, SMS-EMOA, and SPEA2 with enhanced performance, particularly for complex problems featuring rugged fitness landscapes.

Our contribution: We propose a pseudo-Boolean function class CLIMB_ℓ that serves as an example where using dynamic population sizes in NSGA-II leads to a super-constant speedup for some values of ℓ. Here, ℓ is the size of the Pareto front. In essence, reaching the Pareto front of CLIMB_ℓ for small values of ℓ is more difficult than covering it. With a static population size, NSGA-II requires at least $\Omega(\ell n)$ fitness evaluations just to find a single Pareto optimal point. In this setting, an upper bound on the number of fitness evaluations for NSGA-II for covering the front is $O(n\ell + n\log(n) + \ell^2 \log(en/(n-2\ell+3)))$. We also prove that this upper runtime bound for NSGA-II is tight for $\ell = O(\sqrt{n\log(n)})$. In contrast, GSEMO and NSGA-II-DYN, the latter being a variant of NSGA-II with a dynamic population size which we will introduce in this paper (see Algorithm 1 below), can cover the Pareto front of CLIMB_ℓ in an expected $O(n\log(n) + \ell^2 \log(en/(n-2\ell+3)))$ number of evaluations. Hence, those algorithms achieve a super-constant speedup in runtime for $\ell \in \omega(\log(n)) \cap o(\sqrt{n\log(n)})$. For example, if $\ell = \sqrt{n}$, the speedup is of order $\Omega(\sqrt{n}/\log(n))$. For NSGA-II-DYN, we adapt the population size $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$ in each generation, where S_t is a maximum set of mutually incomparable non-dominated solutions of R_t in generation t and λ_t is the offspring population size (see Section 2 for details). For the upper bound, a challenge is to bound the maximum population size in expectation during the hill climbing process, depending on the progress of the algorithm already made towards the Pareto front. One also needs arguments about the population dynamics in the rare case of an unsuccessful initialization. To this end, we use the total sum of Hamming distances of the entire population from given search points, with respect to a specified set of genes, to demonstrate a positive drift towards promising regions of the search space. This is used in [Osuna and Sudholt, 2019] to describe the *clearing* mechanism to preserve population diversity (see also [Lengler et al., 2024] for similar results). For the lower bound, we apply the well-known family tree argument from [Witt, 2006]. For our purposes, we also adapt the general arguments of [Zheng et al., 2022] showing that NSGA-II preserves high-quality solutions to this setting, including the case of dynamic population sizes. We are also confident that we

obtain very similar results when introducing and analyzing dynamic-population variants of NSGA-III, SMS-EMOA, or SPEA2.

Related work: The theoretical runtime analysis of MOEAs initially focused on relatively simple algorithms such as (G)SEMO [Laumanns et al., 2002, 2004, Giel, 2003, Thierens, 2003], and was later extended to NSGA-II, the most widely used MOEA, for the problem to cover the Pareto front [Zheng et al., 2022, Doerr and Qu, 2022, 2023, Dang et al., 2025, 2024, Bian et al., 2023], including combinatorial optimization problems [Cerf et al., 2023, Deng et al., 2024]. Runtime analyses of other popular MOEAs on simple benchmark functions have only emerged in recent years. These include SMS-EMOA [Zheng and Doerr, 2024, Li et al., 2025b], SPEA2 [Ren et al., 2024, Doerr et al., 2026a], variants of NSGA-II [Doerr et al., 2025a], and NSGA-III [Wietheger and Doerr, 2023, Opris et al., 2024, Wietheger and Doerr, 2024, Opris, 2026a, 2025c, 2026c]. It has further been shown that these algorithms can outperform GSEMO exponentially [Dang et al., 2026], including in terms of approximating the Pareto front Li et al. [2025a], particularly when the Pareto front grows exponentially in the problem size n. Even for simpler problems, there are results on approximating the Pareto front when its size exceeds the population size. This includes studies on NSGA-II [Zheng and Doerr, 2022], NSGA-III [Deng et al., 2025], and SPEA2 [Alghouass et al., 2025], with the latter two also demonstrating advantages over NSGA-II. Overall, the theoretical analysis of MOEAs remains an active area of research, with even the relatively simple GSEMO algorithm being the subject of recent work [Doerr et al., 2025b]. Doerr et al. [2025b] established tight runtime bounds for GSEMO on the well-known COCZ benchmark, but did not analyze population growth during hill climbing towards the Pareto front. Thus, it remains open whether the population becomes linear or stays sublinear. In contrast, hill climbing on CLIMB_ℓ is more aggressive: every improvement in hill climbing dominates all previous solutions, causing GSEMO's population to immediately shrink to a single point. This property enables a precise analysis of the population size of NSGA-II-DYN during this phase. Moreover, their work does not establish lower runtime bounds for GSEMO on pseudo-Boolean multi-objective problems for reaching the Pareto front, nor does it consider NSGA-II.

However, how dynamic population sizes can accelerate multi-objective optimization remains largely open. Beyond [Doerr et al., 2026b], which analyzes an NSGA-II variant with dynamic population size on the bi-objective OneMinMax and OneJumpZeroJump functions, we are not aware of runtime analyses of dynamic population sizes in multi-objective optimization. Their analysis focuses primarily on the behavior of Pareto-optimal solutions. In contrast, we analyze NSGA-II-DYN both during hill climbing towards the Pareto front and after reaching it. Controlling the maximum population size during hill climbing is a central technical contribution. We further show that NSGA-II-DYN can match GSEMO's runtime guarantees while maintaining a small population, whereas NSGA-II may exhibit worse runtime. To the best of our knowledge, this paper is the first to theoretically demonstrate such advantages of GSEMO. Further, there is a substantial difference between their NSGA-II variant with dynamic population size and ours. Their variant uses a fixed time window τ, after which the population size is doubled, independently of the current population and, in particular, of the number of mutually incomparable rank-one solutions in the population. Thus, the appropriate choice of τ may depend on the underlying problem, motivating their additional variant that runs several phase lengths in parallel. However, this approach may incur substantial computational overhead, as multiple runs have to be maintained simultaneously. In contrast, our NSGA-II-DYN adapts its population size based on the observed number of mutually incomparable rank-one solutions, which makes it inherently adaptive without tuning an additional parameter.

2 Preliminaries

Notation: Denote by $\mathbb{N}$ the set of natural numbers and by $\mathbb{N}_0$ the set of natural numbers including the zero. For a set B, denote by $|B|$ its cardinality, and by $\ln$ the logarithm to base e. Denote by $\vec{1}_m := (1, \dots, 1)$ the unit vector of dimension m. For two random variables Y and Z on $\mathbb{N}_0$ we say that Z *stochastically dominates* Y if $\Pr(Z \leq c) \leq \Pr(Y \leq c)$ for every $c \geq 0$. For $n \in \mathbb{N}$ let $[n] := \{1, \dots, n\}$ and we say that $\mu = O(\mathrm{poly}(n))$ if μ does not grow asymptotically faster than a polynomial in n. The number of ones in a bit string $x \in \{0,1\}^n$ is denoted by $|x|_1$ and the number of zeros by $|x|_0$, respectively. The *Hamming distance* of two bit strings $x, y \in \{0,1\}^n$ is defined as $H(x,y) = \sum_{j=1}^{n} |x_j - y_j|$. For a d-objective function $f : \{0,1\}^n \to \mathbb{N}_0^d, x \mapsto (f_1(x), \dots, f_d(x))$, and two search points $x, y \in \{0,1\}^n$, x *weakly dominates* y, written as $x \succeq y$, if $f_i(x) \geq f_i(y)$ for

Algorithm 1: NSGA-II-DYN algorithm for maximizing a given d-objective function $f\colon \{0,1\}^n \to \mathbb{R}^d$. The population size is updated by a function $g_t : \bigcup_{\ell=1}^{r}(\{0,1\}^n)^\ell \to \mathbb{N}$, and the number of offspring created by a function $h_t : \bigcup_{\ell=1}^{r}(\{0,1\}^n)^\ell \to \mathbb{N}$ in every generation t, where $r > 0$ is a positive threshold.

Initialize $\mu_0 \in \mathbb{N}$ with $\mu_0 \leq r$
Initialize $P_0 \sim \mathrm{Unif}((\{0,1\}^n)^{\mu_0})$
for $t := 0 \to \infty$ **do**
  Initialize $Q_t := \emptyset$
  Compute $\lambda_t = h_t(P_t)$
  for $i := 1 \to \lambda_t$ **do**
    Sample s from P_t uniformly at random
    Create s' by bitwise mutation on s with rate $1/n$
    Update $Q_t := Q_t \cup \{s'\}$
  Set $R_t := P_t \cup Q_t$, and $\mu_{t+1} = g_t(R_t)$
  Partition R_t into layers $F^1_{t+1}, F^2_{t+1}, \ldots$ of non-dominated solutions
  Find $i^* \geq 1$ such that $\sum_{i=1}^{i^*-1}|F^i_{t+1}| < \mu_{t+1}$ and $\sum_{i=1}^{i^*}|F^i_{t+1}| \geq \mu_{t+1}$
  For each $x \in F^{i^*}_{t+1}$ compute $\textsc{cDist}(x, F^{i^*}_{t+1})$ and let Y_t be the set of $\mu_{t+1} - \sum_{i=1}^{i^*-1}|F^i_{t+1}|$ individuals from $F^{i^*}_{t+1}$ with highest crowding distance with respect to $F^{i^*}_{t+1}$ where ties are broken uniformly at random
  Create the next population $P_{t+1} := \bigcup_{i=1}^{i-1} F^{i^*}_{t+1} \cup Y_t$

all $i \in [d]$ and x *(strictly) dominates* y, written as $x \succ y$, if one inequality is strict. We call x and y *incomparable* if neither $x \succeq y$ nor $y \succeq x$. Each solution x not dominated by any other in $\{0,1\}^n$ is called *Pareto optimal* and we call $f(x)$ *non-dominated fitness value*. The set of all non-dominated fitness values is called *Pareto front*.

Algorithms: NSGA-II with dynamic population sizes, called NSGA-II-DYN, is summarized in Algorithm 1 for bitwise mutation. Its vanilla version, where $g_t = h_t = \mu_0 = \mu$ for a given population size $\mu \in \mathbb{N}$ and all generations t, was introduced in [Deb et al., 2002, Deb, 2011]. The initial population size is set to $\mu_0 \in \mathbb{N}$, and μ_0 individuals are sampled uniformly at random from $\{0,1\}^n$. In each generation, the new offspring population size $\lambda_t = h_t(P_t)$ is computed, and a population Q_t of λ_t offspring is created by repeatedly selecting a parent s from P_t uniformly at random and applying bitwise mutation. That is, each offspring is obtained by flipping each bit of s independently with probability $1/n$. Then the joint population $R_t = P_t \cup Q_t$ of size $\mu_t + \lambda_t$ is computed, and the population size μ_{t+1} for the next generation is adapted via the function g_t where $1 \leq g_t(R_t) \leq \mu_t + \lambda_t$. The functions g_t and h_t are deterministic and chosen by the user. To avoid unnecessarily large population sizes, we assume that g_t and h_t are bounded from above by a positive threshold r, which is also set by the user. To ensure that the Pareto front of a given function f can be covered, we require $r \geq 4|S|$, where S is a maximum set of mutually incomparable solutions with respect to f. This condition guarantees that first ranked solutions are protected between generations (see Lemma 2 below). Note that NSGA-II-DYN may still cover the Pareto front of f even if $\mu_t < 4|S|$ throughout the entire run, since the size of the Pareto front can be much smaller than $|S|$, as it is the case for CLIMB_ℓ for $\ell \in o(n)$ (see Definition 1 below). The vanilla NSGA-II from [Deb et al., 2002] works with $r = \mu$. During survival selection, the parent and offspring populations P_t and Q_t are joined into R_t, and then partitioned into layers $F^1_{t+1}, F^2_{t+1}, \ldots$ by the *non-dominated sorting algorithm* [Deb et al., 2002]. The layer F^1_{t+1} consists of all non-dominated search points, and F^i_{t+1} for $i > 1$ only contains points that are dominated by those from $F^1_{t+1}, \ldots, F^{i-1}_{t+1}$. Then the critical rank i^* with $\sum_{i=1}^{i^*-1}|F^i_{t+1}| < \mu_{t+1}$ and $\sum_{i=1}^{i^*}|F^i_{t+1}| \geq \mu_{t+1}$ is determined (i.e. there are fewer than μ_{t+1} search points in R_t with a lower rank than i^*, but at least μ_{t+1} search points with rank at most i^*). For P_{t+1}, all individuals with a rank lower than i^* are selected. The remaining individuals are all taken from $F^{i^*}_{t+1}$, particularly those with highest crowding distance with respect to $M := F^{i^*}_{t+1}$ where ties are broken uniformly at random. For $M = (x_1, x_2, \ldots, x_{|M|})$ the crowding distance is computed as follows. First, sort the elements of M as $M = (x_{k_1}, \ldots, x_{k_{|M|}})$ with respect to each

Algorithm 2: The GSEMO algorithm [Laumanns et al., 2002, Giel, 2003] for maximizing a given bi-objective function $f\colon \{0,1\}^n \to \mathbb{R}^d$.

```
Initialize P_0 := {s} where s ~ Unif({0,1}^n)
for t := 0 → ∞ do
    Sample s ~ Unif(P_t)
    Create s' by bitwise mutation on s with rate 1/n
    if s' is not dominated by any individual in P_t then
        Create the next population P_{t+1} := P_t ∪ {s'}
        Remove all x ∈ P_{t+1} weakly dominated by s'
    else
        P_{t+1} = P_t
```

objective $k \in [d]$ separately in descending order.

$$\textsc{cDist}(x_i, M) := \sum_{k=1}^{d} \textsc{cDist}_k(x_i, M), \text{ where} \tag{1}$$

$$\textsc{cDist}_k(x_{k_i}, M) := \begin{cases} \infty & \text{if } i \in \{1, |M|\}, \\ 0 & \text{if } i \notin \{1, |M|\} \text{ and } f_k(x_{k_1}) = f_k(x_{k_M}), \\ \frac{f_k(x_{k_{i-1}}) - f_k(x_{k_{i+1}})}{f_k(x_{k_1}) - f_k(x_{k_M})} & \text{otherwise.} \end{cases} \tag{2}$$

Then the first- and last-ranked individuals are always assigned an infinite crowding distance with respect to objective k. If the first- and last-ranked individuals have identical values in objective k, all remaining individuals are assigned a crowding distance of zero with respect to objective k. Otherwise, the crowding distance is computed as the difference between the f_k values of the neighboring individuals ordered immediately above and below. This value is then normalized by the difference between the f_k values of the first and last ordered individuals. We propose a $(\mu + \lambda)$ version of NSGA-II-DYN to capture steady state variants on the one hand, when only a few offspring are created per generation (see for example [Mishra et al., 2016]), and the vanilla version on the other hand, when $\mu = \lambda$ remains constant over time.

The GSEMO algorithm is shown in Algorithm 2. Starting from one randomly generated solution, in each generation a new search point s' is created by bitwise mutation with rate $1/n$ on a parent selected uniformly at random from P_t. If s' is not dominated by any solutions of the current population P_t then it is added to P_t, and those weakly dominated by s' are removed from P_t. Note that the population of GSEMO contains only non-dominated solutions with different fitness vectors, and the population size may vary.

Runtime: We measure runtime as the number of fitness evaluations needed to cover the entire Pareto front, rather than generations, since in NSGA-II-DYN the number of evaluations per generation varies due to its dynamic offspring population size.

Problem class: In this paper, we introduce the bi-objective CLIMB_ℓ function class to illustrate the benefits of using a dynamic population size in NSGA-II, where the parameter ℓ determines the size of the Pareto front. The bit string is split into two halves of length $n/2$. In the first half, both objectives align and simply count ones, creating a strong hill climbing signal: solutions with more ones dominate those with fewer, provided that they have a fitness distinct from zero. In the second half, we simultaneously maximize ones and zeros in an equal way, making the objectives conflicting. As the number of ones in the first half increases, the number of mutually incomparable solutions decreases moderately until it reaches ℓ, the size of the Pareto front. This property ensures that hill climbing remains easy for both GSEMO and NSGA-II-DYN. However, for a broad range of ℓ, reaching the Pareto front via hill climbing is intended to be more difficult than covering it once a Pareto-optimal solution has been found.

Definition 1. *Let n be a natural number divisible by 2 and let $b(n) \in \{1, 2\}$ with $b(n) = 1$ if n is divisible by four, and $b(n) = 2$ otherwise. Then for $\ell \in \{b(n), b(n) + 2, \ldots, n/2 - 1, n/2 + 1\}$ the bi-objective function $\text{CLIMB}_\ell : \{0,1\}^n \to \mathbb{N}_0^2$ is defined as $\text{CLIMB}_\ell(x) = (f_1(x), f_2(x))$ with*

$(f_1(x), f_2(x)) =$

$$\begin{cases} n \cdot |x^1|_1 \vec{1}_2 + (|x^2|_1, |x^2|_0), & \text{if } (\frac{1}{2} - \frac{\ell-1}{n})|x^1|_1 \leq |x^2|_1 \leq \frac{n}{2} - (\frac{1}{2} - \frac{\ell-1}{n})|x^1|_1, \\ (0,0), & \text{else,} \end{cases}$$

for all $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, where $x^1 := (x_1, \ldots, x_{n/2})$ denotes the first half, and $x^2 := (x_{n/2+1}, \ldots, x_n)$ the second half of x.

To simplify the notation, particularly those involving rounding issues, we assume throughout the whole paper that n is divisible by four implying that ℓ is an odd number. All the results below can be easily adapted to the general case where n is not divisible by four, since this only affects constants in the analysis.

Note that all Pareto optimal search points x of CLIMB_ℓ satisfy $|x^1|_1 = n/2$ and $n/4 - (\ell-1)/2 \leq |x^2|_1 \leq n/4 + (\ell-1)/2$. Hence, the Pareto front $\mathcal{F}$ of CLIMB_ℓ is

$$\mathcal{F} = \left\{ \left(\frac{n^2}{2} + k, \frac{n^2}{2} + \frac{n}{2} - k \right) \;\middle|\; k \in \left\{ \frac{n}{4} - \frac{\ell-1}{2}, \ldots, \frac{n}{4} + \frac{\ell-1}{2} \right\} \right\}$$

which has cardinality $n/4 + (\ell-1)/2 - (n/4 - (\ell-1)/2) + 1 = \ell$. So the vanilla NSGA-II needs a population size of at least ℓ to optimize CLIMB_ℓ, particularly to cover its Pareto front, which means that for every $v \in \mathcal{F}$ there is an $x \in P_t$ with $f(x) = v$. The condition ensuring $f(x) \neq 0$ in the first line of the definition of CLIMB_ℓ guarantees that, with overwhelming probability, the initial search point has nonzero fitness. The typical optimization process is that individuals perform hill climbing by increasing the number of ones in their first half, while keeping a fitness distinct from zero. Furthermore, a maximum set of mutually incomparable solutions S with respect to CLIMB_ℓ has size at most $n/2 + 1$ for all $\ell \in \{1, 3, \ldots, n/2 + 1\}$.

Assumptions on g_t, h_t and μ_0: In our main theorem (Theorem 9), we assume that $\mu_0 = 1$ and $\mu_{t+1} = \min\{4|S_t|, \mu_t + \lambda_t\}$, where S_t denotes a maximum set of mutually incomparable non-dominated (first-ranked) solutions in the joint population R_t. Of course, $|S_t| \leq |S|$ where S is a maximum set of mutually incomparable solutions with respect to CLIMB_ℓ. So we let μ_t depend only on the objective vectors attained by P_t, not on their genotype. We consider the choice of μ_{t+1} appropriate, since non-dominated solutions are preserved across generations or even better solutions are created (see Lemma 2), and also dominated solutions in lower layers have a chance to survive. Thus, even such poor solutions can contribute to maintaining a well-spread population, as is the case for vanilla NSGA-II. In our main theorem, we assume that $1 \leq \lambda_t \leq \mu_t$.

Key structural lemma: The following key lemma generalizes Lemmas 1 and 7 from [Zheng et al., 2022] to arbitrary bi-objective functions within the NSGA-II-DYN framework and, in a nutshell, states that promising solutions cannot be lost between generations. The proof idea is the same as in [Zheng et al., 2022], as for every fitness vector v there are at most four individuals in R_t covering v with crowding distance larger zero.

Lemma 2. *Consider NSGA-II-DYN optimizing a bi-objective function $f : \{0,1\}^n \to \mathbb{R}^2$ with any function h_t, any positive threshold $r \geq 4|S|$ for g_t and population size $\mu_{t+1} \geq \min\{4|S_t|, \mu_t + \lambda_t\}$. Let v be covered by a first ranked individual $x \in F^1_{t+1}$. Then there is $y \in P_{t+1}$ with $f(y) = v$.*

3 Runtime Analysis of GSEMO and NSGA-II-DYN on CLIMB_ℓ

In this section, we derive upper bounds on the expected runtime of GSEMO and NSGA-II-DYN when optimizing CLIMB_ℓ. Unlike vanilla NSGA-II, these algorithms allow for a variable population size. One can show that, as long as the Pareto front has not yet been reached, the population size of NSGA-II-DYN remains small enough for hill climbing to be effective. Since GSEMO creates only one offspring per generation and discards dominated solutions, the number of fitness evaluations required to make progress towards the Pareto front is independent of the population size in our analysis. Thus, the exact population size of GSEMO does not need to be determined during the hill climbing phase. However, before hill climbing begins, an individual with nonzero fitness must be created, which occurs with overwhelming probability during initialization by restricting the number of ones in each half on a sufficiently small interval around $n/4$. The main analytical tool for this result is a Chernoff bound, which is applicable because initialization is performed uniformly at random. From Lemma 3 to Lemma 6, we do not assume that $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$, as these results hold in a more general setting.

Lemma 3. *Let $\ell \in \{1, 3, \ldots, n/2 + 1\}$. Consider GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *with $\mu_0 = \text{poly}(n)$, any g_t and any h_t, and any threshold $r \geq 4|S|$. Then all initialized individuals x satisfy $19n/80 \leq |x^1|_1 \leq 21n/80$ and $3n/16 \leq |x^2|_1 \leq 5n/16$ with probability $1 - e^{-\Omega(n)}$. In this case, $f(x) \neq 0$.*

However, to prove the upper bound, we must also handle the rare case in which all initial individuals have fitness zero. In this setting, the population exhibits a positive drift towards the region

$$W := \{x \in \{0,1\}^n \mid n/4 - \sqrt{n} \leq |x^2|_1 \leq n/4 + \sqrt{n}\}.$$

After entering W, one can create an individual y with $|y^2|_1 = n/4$ by flipping $\sqrt{n}$ specific bits at once (see Theorem 9 below). Such a y has nonzero fitness. In what follows, we investigate the drift towards W. Individuals may start on either side of W, so we partition them accordingly. At initialization, we call an individual x *left* if $|x^2|_1 < n/4 - \sqrt{n}$ and *right* if $|x^2|_1 > n/4 + \sqrt{n}$. Then, in generation $t \geq t_0$, we call an individual in R_t *left* if it is the offspring of a left individual and we call it *right* if it is an offspring of a right individual. Note that if all individuals have fitness zero in P_0, the population consists entirely of left and right individuals. We measure progress towards W using the potential

$$\Phi(P_t) := \sum_{x \in P_t} H(x^2, x^*) = \sum_{x \in P_t} \sum_{i=1}^{n/2} |(x^2)_i - x^*_i|,$$

where $x^* = 0^{n/2}$ for left individuals and $x^* = 1^{n/2}$ for right ones. First, we analyze the expected change in this potential over one generation. Note that this analysis takes place on a completely flat fitness landscape, where there is no preference for which individuals are selected or retained.

Lemma 4. *Consider a generation t of GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *with $\ell \in \{1, 3, \ldots, n/2 + 1\}$, population size $\mu := \mu_t$, any offspring population size $\lambda := \lambda_t$, and $\mu_{t+1} = \mu_t = \mu$. Suppose that all individuals in P_t have fitness zero. Then*

$$\mathrm{E}\left[\Phi(P_{t+1}) \mid \Phi(P_t)\right] = \Phi(P_t)\left(1 - \frac{2\lambda}{n(\mu+\lambda)}\right) + \frac{\lambda\mu}{2(\mu+\lambda)},$$

where $\lambda = \mu = 1$ in the case of GSEMO.

Surprisingly, $\Phi(P_{t+1})$ depends only on $\Phi(P_t)$, the population size μ, and the offspring population size λ, and not on the exact genotypes of the μ individuals in P_t, in particular, not on their distribution in the search space. We can interpret the change $\Phi(P_{t+1}) - \Phi(P_t)$ as genetic drift as long as no individual has entered the region W, and use the additive drift theorem [He and Yao, 2004] to bound the expected number of fitness evaluations until an individual enters W.

Lemma 5. *Consider GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *for $\ell \in \{1, 3, \ldots, n/2 + 1\}$, with population size $\mu := \mu_t$ and any offspring population size λ_t bounded by $\lambda \in \text{poly}(n)$ for all $t \geq 1$. Further, let $\mu, \lambda \in \text{poly}(n)$. Suppose that all individuals in P_t have fitness zero. Then the expected number of fitness evaluations required to generate an individual in W is $O(\lambda\mu n\sqrt{n})$ for NSGA-II-DYN and $O(n\sqrt{n})$ for GSEMO.*

After initialization, and after possibly adjusting the population to ensure that at least one individual has nonzero fitness, we aim to reach a Pareto optimal solution via hill climbing. To achieve this, we exploit the strong fitness signal provided by the number of ones in the first half of the bit string, gradually increasing this value while avoiding the creation of individuals with zero fitness. In the following lemma, we derive a lower bound on the probability of increasing, through mutation, the maximum number of ones in the first half without generating an individual with zero fitness.

Lemma 6. *Consider a generation t of GSEMO or NSGA-II-DYN for any choice of g and h when optimizing* CLIMB_ℓ *for $\ell \in \{1, 3, \ldots, n/2 + 1\}$, and any threshold $r \geq 1$. Suppose that $\delta_t := \max\{i \in \{0, \ldots, n/2\} \mid$ there is $x \in P_t$ with $|x^1|_1 = i$ and $f(x) \neq 0\} \in \{0, \ldots, n/2 - 1\}$ which means that there exists $x \in P_t$ with nonzero fitness, but no Pareto optimal search point has been found yet. If an individual $x \in P_t$ with $|x^1|_1 = \delta_t$ is chosen as a parent, then x mutates to a y with $|y^1|_1 > |x^1|_1$ and $f(y) \neq 0$ with probability at least $(n/2 - \delta_t)/(4en)$. This increases δ_t.*

However, GSEMO and NSGA-II-DYN behave quite differently during the hill climbing phase. In GSEMO, the population contains only non-dominated solutions, so some parent with $|x^1|_1 = \delta_t$ is always selected, and in each generation a single offspring is created. In contrast, NSGA-II-DYN

generates λ_t offspring per generation and may also retain dominated solutions. This can result in $\Omega(n)$ non-dominated individuals among the first-ranked solutions, and thus a population of the same order. Such growth slows down hill climbing, since a single generation then already requires $\Omega(n)$ fitness evaluations. Controlling the population size is therefore crucial in the analysis of NSGA-II-DYN on CLIMB_ℓ, as shown in the following lemma. Recall that S_t is a maximum set of mutually incomparable, non-dominated solutions from R_t with respect to CLIMB_ℓ, and S is a maximum set of such solutions in $\{0,1\}^n$.

Lemma 7. *Consider NSGA-II-DYN optimizing* CLIMB_ℓ *for* $\ell \in \{1,3,\dots,n/2+1\}$ *with population sizes* $\mu_0 = 1$ *and* $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$*, where* $1 \le \lambda_t \le \mu_t$ *is the offspring population size, and any threshold* $r \ge 4|S|$*. For* $i \in \{0,\dots,n/2-1\}$ *define* $p_i := (n/2-i)/(4en)$*. Suppose that all individuals* $x \in P_0$ *have nonzero fitness and satisfy* $|x^1|_1 \ge 19n/80$*. Let* $\mu_t^{(i)}$ *denote the maximum population size occurring in a generation* t *when* $\delta_t = i$ *holds. Then* $\mathrm{E}\left[\mu_t^{(i)}\right] \le 320/p_i$*.*

Once the Pareto front is reached, the runtime to cover it can be easily computed using standard arguments from [Zheng et al., 2022, Opris, 2025a].

Lemma 8. *Consider GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *for* $\ell \in \{1,3,\dots,n/2+1\}$ *and suppose there is a generation* t_0 *such that* P_{t_0} *contains at least one Pareto optimal solution. Then the following statements hold.*

(1) GSEMO needs $O(\ell^2 \ln(en/(n-2\ell+3)))$ *fitness evaluations in expectation to cover the Pareto front of* CLIMB_ℓ*.*

(2) For NSGA-II-DYN, suppose that there are $\mu, \lambda \in \mathbb{N}$ *with* $1 \le \mu_t \le \mu$ *and* $1 \le \lambda_t \le \lambda$ *for all* $t \ge t_0$*. Further, suppose that* $\mu_{t+1} \ge \min\{\mu_t + \lambda_t, 4|S_t|\}$ *for all* $t \ge 0$*. Then NSGA-II-DYN needs* $O(\lambda\ell + \mu\ell \ln(en/(n-2\ell+3)))$ *fitness evaluations in expectation to cover the Pareto front of* CLIMB_ℓ*.*

We are now ready to prove the main result of this section: an upper bound on the time required by GSEMO and NSGA-II-DYN to cover the entire Pareto front of CLIMB_ℓ, starting from initialization. The result follows by combining all the preceding lemmas.

Theorem 9. *Consider GSEMO or NSGA-II-DYN optimizing* $f := \text{CLIMB}_\ell$ *for* $\ell \in \{1,3,\dots,n/2+1\}$*. For NSGA-II-DYN, let* $\mu_0 = 1$*,* $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$*, where* $1 \le \lambda_t \le \mu_t$*, and suppose that the positive threshold* r *satisfies* $r \ge 4|S|$*. Then both algorithms need at most* $O(n\ln(n) + \ell^2 \ln(en/(n-2\ell+3)))$ *fitness evaluations in expectation to cover the Pareto front of* CLIMB_ℓ*.*

Proof. We call an individual $x \in P_t$ *good* if $f(x) \neq 0$, and $|x^1|_1 \ge 19n/80$. We consider two cases.

Case 1: Suppose that the initialized individual is good. By Lemma 3, the probability that x is initialized as a good individual equals $e^{-\Omega(n)}$. We apply the method of typical runs [Jansen, 2013, Section 5.6] and split the optimization process into two phases. Some phases may be skipped if their goals have already been achieved. In the first phase, we estimate the expected number of fitness evaluations required to obtain a Pareto optimal solution. The second phase then focuses on covering the entire Pareto front.

Phase 1: Create a Pareto optimal search point. We show that both algorithms need $O(n\ln(n))$ fitness evaluations in expectation to finish this phase. Let δ_t be defined as in Subphase 2. Then $\delta_t \ge 19n/80$, and a Pareto optimal solution exists if $\delta_t = n/2$. Further, δ_t cannot decrease by Lemma 2 for NSGA-II-DYN, and likewise cannot decrease for GSEMO, since the latter retains only non-dominated solutions. By Lemma 6, δ_t can be increased with probability at least $(n/2-\delta_t)/(4en)$ if an individual $x \in P_t$ with $|x^1|_1 = \delta_t$ is chosen as a parent.

GSEMO: Note that P_t consists only of good individuals x with $|x^1|_1 = \delta_t$ and hence such an individual is chosen with probability one as a parent. Thus, δ_t increases with probability at least $(n/2-\delta_t)/(4en)$ in each generation. Thus, setting $p_i := (n/2-i)/(4en)$, the expected number of generations (which is one less than the number of fitness evaluations) required to complete this phase in total is at most

$$\sum_{i=\lceil 19n/80\rceil}^{n/2-1} \frac{1}{p_i} \le \sum_{i=1}^{n/2-1} \frac{4en}{n/2-i} \le \sum_{i=1}^{n/2} \frac{4en}{i} \le 4en(\ln(n/2)+1) = O(n\ln(n))$$

where we use the harmonic-sum bound $\sum_{i=1}^{k} 1/i \leq \ln(k) + 1$ for all $k \in \mathbb{N}$.

NSGA-II-DYN: Note also that $\mu_{t+1} \leq 4|S_t|$ and hence at least $\mu_{t+1}/4$ individuals from P_{t+1} are non-dominated, since by the non-dominated sorting procedure at least $\min\{\mu_{t+1}, |S_t|\} = \min\{\mu_t + \lambda_t, 4|S_t|, |S_t|\} = |S_t|$ non-dominated individuals are taken into P_{t+1}. Therefore, an individual x with $|x^1|_1 = i = \delta_t$ is chosen as a parent with probability at least $1/4$. So we need $4/p_i$ fitness evaluations in expectation to create an individual y with $|y^1|_1 > \delta_t$ and nonzero fitness. Note that this number is independent of the current population size μ_t, since at least $\mu_t/4$ individuals from P_t are always non-dominated. Denote by t_i^* the generation in which δ_t increases from i. It remains to estimate the expected number of fitness evaluations until generation $t_i^* + 1$ is reached (in which $\delta_{t_i^*+1} > i$). Note that all $x \in P_t$ have fitness distinct from zero and satisfy $|x^1|_1 \geq 19n/80$. Note also that $\mu_0 = 1$, since we initialize a good individual. So we can apply Lemma 7 and obtain that the maximum population size $\mu_t^{(i)}$ over all generations t for which $\delta_t = i$ satisfies $\mathrm{E}\left[\mu_t^{(i)}\right] \leq 320/p_i$. Thus, after also accounting for the at most $\lambda_t \leq \mu_t$ additional offspring created in generation $t = t_i^*$, we need at most $4/p_i + 320/p_i = 324/p_i$ fitness evaluations in expectation to start in the subsequent generation $t_i^* + 1$. Hence, after at most $324/p_i$ fitness evaluations in expectation, δ_t has been increased. Since $\delta_t \geq \lceil 19n/80 \rceil$, the total expected number of fitness evaluations required to reach $\delta_t = n/2$, or equivalently the Pareto front, is at most

$$\sum_{i=\lceil 19n/80 \rceil}^{n/2-1} \frac{324}{p_i} \leq \sum_{i=1}^{n/2-1} \frac{324}{p_i} = O(n \ln(n)).$$

Phase 2: Cover the whole Pareto front: We show that both algorithms need $O(\ell^2 \ln(en/(n - 2\ell + 3)))$ fitness evaluations in expectation to cover the Pareto front. For GSEMO, this follows from Lemma 8(1). For NSGA-II-DYN, we have that $\mu_0 = 1$ and $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$ for $t \geq 0$. Note that $|S_t| \leq \ell$ and hence $\mu_t \leq 4\ell$. Thus, we can also apply Lemma 8(2) with $\mu = \lambda = 4\ell$ to see that NSGA-II-DYN needs $O(\ell^2 \ln(en/(n - 2\ell + 3)))$ fitness evaluations in expectation to cover the Pareto front of CLIMB_ℓ.

Case 2: Suppose that the initialized individual is not good. We then divide the optimization process into three phases, where the last two correspond to the two phases considered in the first case. In the first phase, we estimate the expected number of fitness evaluations needed to obtain a Pareto-optimal solution.

Phase 1: All search points $x \in P_t$ have fitness distinct from zero.

Phase 2: Create a Pareto optimal search point.

Phase 3: Cover the Pareto front.

Lemma 10. *GSEMO and NSGA-II-DYN complete Phase 1 in expected $O(n\sqrt{n}e^{\sqrt{n}\ln(n)})$ fitness evaluations, Phase 2 in expected $O(n^2)$ fitness evaluations, and Phase 3 in expected $O(\ell^2 \ln(en/(n - 2\ell + 3)))$ fitness evaluations.*

By summing the runtimes of both phases from Case 1 and the three phases from Case 2, we conclude with Lemma 10 that the total expected number of fitness evaluations required for both algorithms to cover the Pareto front is at most

$$\begin{aligned} &O\Big(e^{-\Omega(n)}\Big(n\sqrt{n}e^{\sqrt{n}\ln(n)} + n^2\Big) + (1 - e^{-\Omega(n)})n\ln(n) + \ell^2 \ln\Big(\frac{en}{n - 2\ell + 3}\Big)\Big) \\ &\qquad = O\Big(n\ln(n) + \ell^2 \ln\Big(\frac{en}{n - 2\ell + 3}\Big)\Big) \end{aligned}$$

which proves the theorem. □

To complement the established upper runtime bounds for GSEMO and NSGA-II-DYN, we also provide lower runtime bounds for reaching the Pareto front of CLIMB_ℓ. Note that this process relies solely on hill climbing. Consequently, for $\ell = O(\sqrt{n \ln n})$, the runtime bounds for GSEMO and NSGA-II-DYN on CLIMB_ℓ are tight.

Theorem 11. *Both GSEMO and NSGA-II-DYN, with $\mu_0 = 1$, any choice of g_t and h_t, and any threshold $r \geq 1$, require $\Omega(n \ln(n))$ fitness evaluations in expectation to reach the Pareto front when optimizing $f := \mathrm{CLIMB}_\ell$ for $\ell \in \{1, 3, \ldots, n/2 + 1\}$.*

4 Runtime Analysis of the Vanilla NSGA-II on CLIMB_ℓ

In this section, we first provide an upper bound on the runtime of NSGA-II-DYN with fixed population size $\mu \geq \ell$ and offspring population size λ for the optimization of CLIMB_ℓ. The proof of this result combines arguments from [Antipov and Doerr, 2021] for analyzing the hill climbing process with those from [Zheng et al., 2022] for spreading solutions across the Pareto front and covering it.

Theorem 12. *Consider $f := \text{CLIMB}_\ell$ for $\ell \in \{1, 3, \ldots, n/2\}$. Let $\mu, \lambda \in \mathbb{N}$ with $\mu \geq 4\ell$. Then, NSGA-II-DYN with $\mu_t = \mu$, $\lambda_t = \lambda$, and any threshold $r \geq \mu$, requires $O((\mu + \lambda)n + n\ln(n) + \mu\ell \ln(en/(n - 2\ell + 3)))$ fitness evaluations in expectation to cover the Pareto front of f. For $\mu = 4\ell$ and $1 \leq \lambda \leq \mu$, the stated runtime bound becomes $O(\ell n + n\ln(n) + \ell^2 \ln(en/(n - 2\ell + 3)))$.*

However, the main result of this section is a lower bound on the number of fitness evaluations this version of NSGA-II-DYN with fixed population size μ and offspring population size λ requires to optimize CLIMB_ℓ, in particular, to find a Pareto-optimal search point. This fixed population size may hinder efficient optimization: μ must be at least as large as the size of the Pareto front, forcing the algorithm to maintain a population of size at least ℓ from the start. Consequently, many fitness evaluations are spent on individuals with low fitness, which slows down progress towards the Pareto front. To formalize this, we adapt the family-tree argument from [Witt, 2006], similar to its use as in [Sudholt, 2009, Antipov and Doerr, 2021]. We define a family forest consisting of μ family trees. Each node stores a triple: the individual, the generation t in which it was created, and an index $i \in \{0, 1, \ldots, \lambda\}$ indicating that the individual was created during the i-th iteration of the offspring loop (Line 5 of Algorithm 1), where $i = 0$ indicates an initialized individual. Each node is linked to its parent. The forest is defined inductively. In generation 0, it consists of μ single-node trees, one for each individual in P_0, with all components of the triple equal to 0. Suppose the forest for P_t is given. During generation t, whenever a parent x creates an offspring y during iteration i of the offspring loop, we add a new node $(y, t + 1, i)$ to the same tree as x and connect it to x. After all offspring have been created, this gives the forest for P_{t+1}. The survival selection in generation $t + 1$ does not alter the forest.

Theorem 13. *For $1 \leq \mu = O(\text{poly}(n))$ and $1 \leq \lambda = O(\text{poly}(n))$ the NSGA-II-DYN with fixed population size μ and offspring population size λ (if $g = \mu$ and $h = \lambda$) needs at least $\Omega(\mu n)$ fitness evaluations in expectation to find a Pareto optimal search point of $f := \text{CLIMB}_\ell$ for every $\ell \in \{1, 3, 5, \ldots, n/2 + 1\}$. For $\mu = \ell$, the expected number of fitness evaluations becomes $\Omega(\ell n)$.*

5 Conclusions and Discussions

To explore the benefits of variable population sizes, we developed NSGA-II-DYN, a variant of NSGA-II that dynamically adjusts the population size. Our results show that this flexibility can improve performance, especially during the hill-climbing phase. Specifically, we showed that both GSEMO and NSGA-II-DYN achieve superconstant speedups over the vanilla NSGA-II on CLIMB_ℓ across a broad range of values of ℓ. This contrasts sharply with classical pseudo-Boolean benchmarks such as LOTZ, OMM, and COCZ, for which only matching upper bounds on the expected running times of GSEMO and NSGA-II are known. To the best of our knowledge, this is the first result showing a superconstant speedup of GSEMO over vanilla NSGA-II on a bi-objective pseudo-Boolean problem. Similar advantages may hold for other MOEAs with fixed population sizes, such as SPEA-2, SMS-EMOA, or NSGA-III. However, our work has several limitations that point to interesting directions for future research. First, our analysis is limited to CLIMB_ℓ. It remains unclear how MOEAs with dynamic populations behave on more complex benchmark problems and whether dynamic population sizes can affect not only hill climbing but also the exploration of different regions of the search space, potentially leading to exponential speedups. Second, we only considered specific update rules for adjusting the population size in NSGA-II-DYN. One could also investigate alternative choices of g and h and how they may accelerate the optimization process. Another direction for future work is to consider different MOEAs, such as PAES25, and dynamize the size of their archives [Opris, 2025b]. Overall, alongside [Doerr et al., 2026b], this work represents an initial step towards understanding the role of dynamic population sizes in evolutionary multi-objective optimization. We hope that the insights gained will be valuable for both theoretical analysis and practical applications. In particular, they may guide the development of improved variants of our proposed NSGA-II-DYN, further enhancing the performance of MOEAs on complex real-world problems.

References

Yasser Alghouass, Benjamin Doerr, Martin S. Krejca, and Mohammed Lagmah. Proven approximation guarantees in multi-objective optimization: SPEA2 beats NSGA-II. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, pages 8833–8841. ijcai.org, 2025.

Shahla Alizadeh, Souvik Ta, and Lakshminarayanan Samavedham. Physics-informed neural network with NSGA II and levenberg–marquardt method for kinetic modeling in heavy oil hydrocracking. *Industrial and Engineering Chemistry Research*, 64(40):19624 – 19640, 2025.

Denis Antipov and Benjamin Doerr. A tight runtime analysis for the $(\mu + \lambda)$ EA. *Algorithmica*, 83: 1054–1095, 2021.

Chao Bian, Yawen Zhou, Miqing Li, and Chao Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI 2023, pages 5513–5521. ijcai.org, 2023.

Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Yakob Kahane, and Simon Wietheger. The first proven performance guarantees for the non-dominated sorting genetic algorithm II (NSGA-II) on a combinatorial optimization problem. abs/2305.13459, 2023.

Fahmida Khatun Chowdhury, Md Sadman Sami, Md. Jahedul Alam, Ahmed Sayem, and Mohammad Muhshin Aziz Khan. Machine learning-assisted NSGA-II-TOPSIS optimization of weld strength and processing time in fused deposition modeling. *Results in Engineering*, 29:109365, 2026.

Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. Analysing the robustness of NSGA-II under noise. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'23)*, pages 642–651. ACM Press, 2023.

Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. *Artificial Intelligence*, 330:104098, 2024.

Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Runtime analysis of functions where widely used evolutionary multi-objective algorithms beat simple ones. *ACM Transactions on Evolutionary Learning and Optimization*, 6(2):1–33, 2025.

Duc-Cuong Dang, Adrian Opris, and Dirk Sudholt. Why dominance is not enough: Lessons from practical evolutionary multi-objective algorithms. *Algorithmica*, 88:60, 2026.

Kalyanmoy Deb. NSGA-II source code in C, version 1.1.6. `https://www.egr.msu.edu/~kdeb/codes/nsga2/nsga2-gnuplot-v1.1.6.tar.gz`, 2011. Accessed: 2022-08-15.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

Renzhong Deng, Weijie Zheng, Mingfeng Li, Jie Liu, and Benjamin Doerr. Runtime analysis for state-of-the-art multi-objective evolutionary algorithms on the subset selection problem. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*, PPSN XVIII, page 264–279. Springer, 2024.

Renzhong Deng, Weijie Zheng, and Benjamin Doerr. The first theoretical approximation guarantees for the non-dominated sorting genetic algorithm III (NSGA-III). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, pages 8867–8875. ijcai.org, 2025.

Matthieu Dinot, Benjamin Doerr, Ulysse Hennebelle, and Sebastian Will. Runtime analyses of multi-objective evolutionary algorithms in the presence of noise. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 5549–5557. ijcai.org, 2023.

Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*, PPSN XVII, pages 399–412. Springer, 2022.

Benjamin Doerr and Zhongdi Qu. Runtime analysis for the NSGA-II: provable speed-ups from crossover. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2023, pages 12399–12407. AAAI Press, 2023.

Benjamin Doerr, Tudor Ivan, and Martin S. Krejca. Speeding up the NSGA-II with a simple tie-breaking rule. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2025, pages 26964–26972. AAAI Press, 2025a.

Benjamin Doerr, Martin S. Krejca, and Andre Opris. Tight runtime guarantees from understanding the population dynamics of the GSEMO multi-objective evolutionary algorithm. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, pages 8876–8884. ijcai.org, 2025b.

Benjamin Doerr, Martin S. Krejca, and Milan Stankovic. Improved runtime guarantees for the SPEA2 multi-objective optimizer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2026, pages 36855–36863. AAAI Press, 2026a.

Benjamin Doerr, Martin S. Krejca, and Simon Wietheger. Speeding up the NSGA-II via dynamic population sizes. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI 2026, pages 6273–6281. ijcai.org, 2026b.

Oliver Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*, CEC 2003, pages 1918–1925. IEEE Press, 2003.

Jun He and Xin Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3:21–35, 2004.

H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3):392–403, 1998.

Thomas Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Natural Computing Series. Springer, 2013.

Mario Köppen and Kaori Yoshida. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization*, pages 727–741. Springer, 2007.

Marco Laumanns, Lothar Thiele, Eckart Zitzler, Emo Welzl, and Kalyanmoy Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature, PPSN 2002*, pages 44–53. Springer, 2002.

Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of multiobjective evolutionary algorithms on Pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2):170–182, 2004.

Johannes Lengler, Andre Opris, and Dirk Sudholt. Analysing equilibrium states for population diversity. *Algorithmica*, 27(3):2317–2351, 2024.

Mingfeng Li, Qiang Zhang, Weijie Zheng, and Benjamin Doerr. Why popular MOEAs are popular: Proven advantages in approximating the pareto front. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL `https://openreview.net/forum?id=8OvST1bejm`.

Mingfeng Li, Weijie Zheng, and Benjamin Doerr. Scalable speed-ups for the SMS-EMOA from a simple aging strategy. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, pages 8885–8893. ijcai.org, 2025b.

Jing Liang, Zhaolin Chen, Yaonan Wang, Xuanxuan Ban, Kangjia Qiao, and Kunjie Yu. A dual-population constrained multi-objective evolutionary algorithm with variable auxiliary population size. *Complex & Intelligent Systems*, 9(5):5907–5922, 2023.

Sohvi Luukkonen, Helle W. van den Maagdenberg, Michael T.M. Emmerich, and Gerard J.P. van Westen. Artificial intelligence in multi-objective drug design. *Current Opinion in Structural Biology*, 79:102537, 2023.

Miguel Martínez-Comesaña, Javier Martínez-Torres, and Pablo Eguía-Oller. Optimisation of LSTM neural networks with NSGA-II and FDA for PV installations characterisation. *Engineering Applications of Artificial Intelligence*, 126:106770, 2023.

Izanloo Mehri, Shahravi Majid, and Khalkhali Abolfazl. Hybrid machine learning and NSGA-II optimization of rail anti-climbing systems: a multi-criteria design framework. *Structural and Multidisciplinary Optimization*, 69(17), 2025.

Sumit Mishra, Samrat Mondal, and Sriparna Saha. Fast implementation of steady-state NSGA-II. In *IEEE Congress on Evolutionary Computation*, CEC 2016, pages 3777–3784, 2016.

Andre Opris. A many objective problem where crossover is provably indispensable. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2025, pages 27108–27116. AAAI Press, 2025a.

Andre Opris. A first runtime analysis of the PAES-25: An enhanced variant of the Pareto archived evolution strategy. In *Proceedings of the Foundations of Genetic Algorithms*, FOGA 2025, page 202–213. ACM, 2025b.

Andre Opris. A first runtime analysis of NSGA-III on a many-objective multimodal problem: Provable exponential speedup via stochastic population update. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2025, pages 8903–8911. ijcai.org, 2025c.

Andre Opris. Many-objective problems where crossover is provably essential. *Artificial Intelligence*, 350:104453, 2026a.

Andre Opris. Provable speedups from dynamic population sizes in evolutionary algorithms for multiobjective optimization, 2026b. URL `https://arxiv.org/abs/2607.23800`. to appear at NEURIPS 2026.

Andre Opris. Towards a rigorous understanding of the population dynamics of the NSGA-III: Tight runtime bounds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2026, pages 37125–37133. AAAI Press, 2026c.

Andre Opris. On the impact of crossover in many-objective optimization: A runtime analysis of NSGA-III. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI 2026, pages 6334–6342. ijcai.org, 2026d.

Andre Opris, Duc-Cuong Dang, Frank Neumann, and Dirk Sudholt. Runtime analyses of NSGA-III on many-objective problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2024, page 1596–1604. ACM Press, 2024.

Edgar Covantes Osuna and Dirk Sudholt. On the runtime analysis of the clearing diversity-preserving mechanism. *Evolutionary Computation*, 27(3):403–433, 2019.

Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. In *Neural Information Processing Systems, NIPS 2015*, pages 1774–1782, 2015.

Chao Qian, Yibo Zhang, Ke Tang, and Xin Yao. On multiset selection with size constraints. In *Conference on Artificial Intelligence*, AAAI 2018, pages 1395–1402. AAAI Press, 2018.

Shengjie Ren, Chao Bian, Miqing Li, and Chao Qian. A first running time analysis of the Strength Pareto Evolutionary Algorithm 2 (SPEA2). In *Proceedings of the International Conference of Parallel Problem Solving from Nature*, PPSN XVIII, pages 295–312. Springer, 2024.

Dirk Sudholt. The impact of parametrization in memetic evolutionary algorithms. *Theoretical Computer Science*, 410(26):2511–2528, 2009.

Dirk Thierens. Convergence time analysis for the multi-objective counting ones problem. In *Evolutionary Multi-Criterion Optimization*, EMO 2003, pages 355–364. Springer, 2003.

Ye Tian, Ruiqin Wang, Yajie Zhang, and Xingyi Zhang. Adaptive population sizing for multi-population based constrained multi-objective optimization. *Neurocomputing*, 621:129296, 2025.

Bing-Chuan Wang, Zhong-Yi Shui, Yun Feng, and Zhongwei Ma. Evolutionary algorithm with dynamic population size for constrained multiobjective optimization. *Swarm and Evolutionary Computation*, 73:101104, 2022.

Simon Wietheger and Benjamin Doerr. A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III). In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI 2023, pages 5657–5665. ijcai.org, 2023.

Simon Wietheger and Benjamin Doerr. Near-tight runtime guarantees for many-objective evolutionary algorithms. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*, PPSN XVIII, pages 153–168, 2024.

Carsten Witt. Runtime analysis of the $(\mu + 1)$ EA on simple Pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.

Liao Xingtao, Li Qing, Yang Xujing, and Zhang Weigang. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35:561–569, 2008.

Xinzhe Zhang, Rentao Wang, Chuankun Zang, Kai Song, Xiaolu Wang, and Guoju Li. Machine learning-based multi-objective optimization of ceramic composite armor plates via the NSGA-II genetic algorithm. *Composites Part B: Engineering*, 307:112961, 2025.

Weijie Zheng and Benjamin Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2022, pages 611–619. ACM Press, 2022.

Weijie Zheng and Benjamin Doerr. Runtime analysis of the SMS-EMOA for many-objective optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI 2024, pages 20874–20882. AAAI Press, 2024.

Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 10408–10416. AAAI Press, 2022.

A Technical appendices and supplementary material

This document contains the proofs that we omitted in the main paper in full details, due to space restrictions. Recall that S_t is a maximum set of mutually incomparable, non-dominated solutions from R_t with respect to CLIMB_ℓ, and S is a maximum set of such solutions from $\{0,1\}^n$. Further, recall that R_t denotes the joint population of parents P_t and offspring Q_t in generation t, and F^1_{t+1} denotes the first ranked solutions from R_t according to non-dominated sorting (see Algorithm 1).

Lemma 2. *Consider NSGA-II-DYN optimizing a bi-objective function $f : \{0,1\}^n \to \mathbb{R}^2$ with any function h_t, any positive threshold $r \geq 4|S|$ for g_t and population size $\mu_{t+1} \geq \min\{4|S_t|, \mu_t + \lambda_t\}$. Let v be covered by a first ranked individual $x \in F^1_{t+1}$. Then there is $y \in P_{t+1}$ with $f(y) = v$.*

Proof. If $\mu_{t+1} = \mu_t + \lambda_t$ then $P_{t+1} = R_t$ and the lemma holds. So suppose that $\mu_{t+1} = 4|S_t|$. First, we claim that there are at most $4|S_t|$ individuals with positive crowding distance in F^1_{t+1} as all individuals with positive crowding distance in F^1_{t+1} survive. Let v be a fitness vector covered by at least five first ranked individuals. This means that there are $m \geq 5$ individuals $x_1, \ldots, x_m$ with fitness vector v, sorted with respect to f_1 in descending order, say $(x_1, \ldots, x_m)$. With respect to f_2, assume the descending order is $(x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ for distinct $i_1, \ldots, i_m \in [m]$. Let $j \in \{1, \ldots, m\} \setminus (\{1, m\} \cup \{i_1, i_m\})$. Then $j = i_k$ for some $k \in \{2, \ldots, m-1\}$ and x_j has crowding distance zero, since $f_1(x_{j-1}) - f_1(x_{j+1}) = 0$ and $f_2(x_{i_{k-1}}) - f_2(x_{i_{k+1}}) = 0$, proving the claim that there are at most $4|S_t|$ individuals with positive crowding distance in F^1_{t+1}.

Note that for every vector v covered by an individual $x \in F^1_{t+1}$, there exists an individual $y \in F^1_{t+1}$ with $f(y) = v$ that has positive crowding distance. To see this, let $V := \{v \in \mathbb{N}_0^2 \mid \text{ there is } x \in F^1_{t+1} \text{ with } f(x) = v\}$. If $v_1 = \max\{v_1 \mid v \in V\}$, then there exists an $x \in F^1_{t+1}$ with crowding distance equal to infinity (which is, for instance, x_{k_1} in the above definition of the crowding distance with respect to F^1_{t+1}). Otherwise, consider, for example, the individual x_i covering v that appears first in the descending order with respect to f_1. Its crowding distance with respect to F^1_{t+1} is at least

$$\frac{f_1(x_{i-1}) - f_1(x_{i+1})}{f_1(x_{k_1}) - f_1(x_{k_M})} > 0$$

since $f_1(x_{k_1}) > f_1(x_{k_M})$ (otherwise $v_1 = \max\{v_1 \mid v \in V\}$). This proves the lemma, since such a y survives by the claim above. □

Lemma 3. *Let $\ell \in \{1, 3, \ldots, n/2+1\}$. Consider GSEMO or NSGA-II-DYN optimizing CLIMB_ℓ with $\mu_0 = \text{poly}(n)$, any g_t and any h_t, and any threshold $r \geq 4|S|$. Then all initialized individuals x satisfy $19n/80 \leq |x^1|_1 \leq 21n/80$ and $3n/16 \leq |x^2|_1 \leq 5n/16$ with probability $1 - e^{-\Omega(n)}$. In this case, $f(x) \neq 0$.*

Proof. Using a standard Chernoff bound, the probability that an individual x initializes with $(|x^1|_1, |x^2|_1) \in [\frac{19n}{80}, \frac{21n}{80}] \times [\frac{3n}{16}, \frac{5n}{16}] \subset [\frac{n}{8}, \frac{3n}{8}] \times [\frac{3n}{16}, \frac{5n}{16}]$ is $1 - e^{-\Omega(n)}$, since the expected number of ones in each half is $\frac{n}{4}$. Note also that each search point x with $(|x^1|_1, |x^2|_1) \in [\frac{n}{8}, \frac{3n}{8}] \times [\frac{3n}{16}, \frac{5n}{16}]$ satisfies

$$\begin{aligned}|x^1|_1 \left(\frac{1}{2} - \frac{\ell-1}{2}\right) \leq \frac{3n}{8}\left(\frac{1}{2} - \frac{\ell-1}{2}\right) = \frac{3n}{16} - \frac{3n(\ell-1)}{16} \leq \frac{3n}{16} \leq |x^2|_1 \leq \frac{5n}{16} = \frac{n}{2} - \frac{3n}{16} \\ \leq \frac{n}{2} - \frac{3n}{16} + \frac{3n}{8}\frac{\ell-1}{n} = \frac{n}{2} - \frac{3n}{8}\left(\frac{1}{2} - \frac{\ell-1}{n}\right) \leq \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|x^1|_1\end{aligned}$$

and hence, x has fitness distinct from zero. The statement for the whole population of size μ_0 follows by a union bound, since $\mu_0 = \text{poly}(n)$. □

Lemma 4. *Consider a generation t of GSEMO or NSGA-II-DYN optimizing CLIMB_ℓ with $\ell \in \{1, 3, \ldots, n/2+1\}$, population size $\mu := \mu_t$, any offspring population size $\lambda := \lambda_t$, and $\mu_{t+1} = \mu_t = \mu$. Suppose that all individuals in P_t have fitness zero. Then*

$$\mathrm{E}\left[\Phi(P_{t+1}) \mid \Phi(P_t)\right] = \Phi(P_t)\left(1 - \frac{2\lambda}{n(\mu+\lambda)}\right) + \frac{\lambda\mu}{2(\mu+\lambda)},$$

where $\lambda = \mu = 1$ in the case of GSEMO.

Proof. Let y be an individual generated by mutation on $x \in P_t$. We obtain under the condition that we select x as a parent, denoted by event B_x,

$$\begin{aligned}\mathrm{E}\left[H(y^2, x^*) \mid B_x\right] &= H(x^2, x^*) + \frac{n/2 - H(x^2, x^*)}{n} - \frac{H(x^2, x^*)}{n} \\ &= H(x^2, x^*) - \frac{2H(x^2, x^*)}{n} + \frac{1}{2}.\end{aligned}$$

Note that the expected number of flipped bits increasing the Hamming distance between x^2 and x^* is $(n/2 - H(x^2, x^*))/n$ (i.e. the number of positions in the second half of x where x, x^* coincide divided by n), while the expected number of bits decreasing the Hamming distance between x^2 and x^* is $H(x^2, x^*)/n$. Hence, if x among μ parents is chosen uniformly at random, then by the law of total probability

$$\mathrm{E}\left[H(y^2, x^*)\right] = \frac{1}{\mu} \sum_{x \in P_t} \left(H(x^2, x^*) - \frac{2H(x^2, x^*)}{n} + \frac{1}{2} \right) = \frac{\Phi(P_t)}{\mu} - \frac{2\Phi(P_t)}{\mu n} + \frac{1}{2}$$

and therefore, if λ offspring $y_1, \ldots, y_\lambda$ are generated,

$$\mathrm{E}\left[\Phi(Q_t) \mid \Phi(P_t)\right] = \lambda \left(\frac{\Phi(P_t)}{\mu} - \frac{2\Phi(P_t)}{\mu n} + \frac{1}{2} \right) = \frac{\lambda\Phi(P_t)}{\mu} - \frac{2\lambda\Phi(P_t)}{\mu n} + \frac{\lambda}{2}$$

where $Q_t = \{y_1, \ldots, y_\lambda\}$. This implies for the joint population $R_t = P_t \cup Q_t$ that

$$\begin{aligned}\mathrm{E}\left[\Phi(R_t) \mid \Phi(P_t)\right] &= \mathrm{E}\left[\Phi(P_t) \mid \Phi(P_t)\right] + \mathrm{E}\left[\Phi(Q_t) \mid \Phi(P_t)\right] \\ &= \left(1 + \frac{\lambda}{\mu}\right) \Phi(P_t) - \frac{2\lambda\Phi(P_t)}{\mu n} + \frac{\lambda}{2}.\end{aligned}$$

To form P_{t+1}, λ individuals chosen uniformly at random are removed from R_t (since $\mu_{t+1} = \mu_t = \mu$). Note that the probability to remove a certain individual $x \in R_t$ is $\lambda/(\mu + \lambda)$, and hence

$$\mathrm{E}\left[\Phi(P_{t+1}) \mid \Phi(R_t)\right] = \Phi(R_t) - \sum_{x \in R_t} \frac{\lambda H(x^2, x^*)}{\mu + \lambda} = \Phi(R_t) - \frac{\lambda}{\mu + \lambda}\Phi(R_t) = \frac{\mu\Phi(R_t)}{\mu + \lambda}$$

which implies

$$\begin{aligned}\mathrm{E}\left[\Phi(P_{t+1}) \mid \Phi(P_t)\right] &= \frac{\mu}{\mu + \lambda} \left(\left(1 + \frac{\lambda}{\mu}\right) \Phi(P_t) - \frac{2\lambda\Phi(P_t)}{\mu n} + \frac{\lambda}{2} \right) \\ &= \Phi(P_t) - \frac{2\lambda\Phi(P_t)}{n(\mu + \lambda)} + \frac{\lambda\mu}{2(\mu + \lambda)} \\ &= \Phi(P_t) \left(1 - \frac{2\lambda}{n(\mu + \lambda)}\right) + \frac{\lambda\mu}{2(\mu + \lambda)}.\end{aligned}$$

This proves the lemma by noting that GSEMO only creates one offspring in each iteration, and satisfies $|P_t| = 1$. □

Lemma 5. *Consider GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *for* $\ell \in \{1, 3, \ldots, n/2 + 1\}$*, with population size* $\mu := \mu_t$ *and any offspring population size* λ_t *bounded by* $\lambda \in \mathrm{poly}(n)$ *for all* $t \geq 1$*. Further, let* $\mu, \lambda \in \mathrm{poly}(n)$*. Suppose that all individuals in* P_t *have fitness zero. Then the expected number of fitness evaluations required to generate an individual in* W *is* $O(\lambda\mu n\sqrt{n})$ *for NSGA-II-DYN and* $O(n\sqrt{n})$ *for GSEMO.*

Proof. Define $\Phi(P_t) := \sum_{x \in P_t} H(x^2, x^*)$. Let T be the time until we created an individual x with $H(x^2, x^*) \geq n/4 - \sqrt{n}$. We obtain $H(x^2, x^*) \geq n/4 - \sqrt{n}$ for an individual $x \in P_t$ if $\Phi(P_t) \geq \mu n/4 - \mu\sqrt{n}$. So suppose $\Phi(P_t) < \mu n/4 - \mu\sqrt{n}$. We see by Lemma 4 that for all $\lambda_t \geq 1$

$$\mathrm{E}\left[\Phi(P_{t+1}) \mid \Phi(P_t)\right] = \Phi(P_t) \left(1 - \frac{2\lambda_t}{n(\mu + \lambda_t)}\right) + \frac{\lambda_t\mu}{2(\mu + \lambda_t)}$$

and therefore,

$$\mathrm{E}\left[\Phi(P_{t+1}) - \Phi(P_t) \mid \Phi(P_t)\right] = \frac{\lambda\mu}{2(\mu + \lambda_t)} - \frac{2\lambda_t}{n(\mu + \lambda_t)}\Phi(P_t)$$

$$> \frac{\lambda_t\mu}{2(\mu+\lambda_t)} - \frac{2\lambda_t\mu}{n(\mu+\lambda_t)}(n/4-\sqrt{n})$$
$$= \frac{2\lambda_t\mu}{\sqrt{n}(\mu+\lambda_t)} \geq \frac{1}{\sqrt{n}} =: \delta.$$

By the additive drift theorem [He and Yao, 2004] we obtain $\mathrm{E}[T] \leq (\mu n/4 - \mu\sqrt{n})/\delta \leq \mu n\sqrt{n}/4$. In other words, after at most $\mu n\sqrt{n}/4$ generations in expectation, we created a search point x with $H_2(x^*, x) \geq n/4 - \sqrt{n}$. In this case, either this search point is in W, or is an offspring of an individual z with the following property. If $|z^2|_1 < n/4 - \sqrt{n}$ then $|x^2|_1 > n/4 + \sqrt{n}$, or if $|z^2|_0 < n/4 - \sqrt{n}$ then $|x^2|_0 > n/4 + \sqrt{n}$. In other words, a mutation on z causes $|z^2|_1$ to cross the interval $[n/4 - \sqrt{n}, n/4 + \sqrt{n}]$ which we denote as a *failure*. However, this requires to flip at least $\lceil 2\sqrt{n} \rceil$ many bits at once. So the probability of a failure is at most

$$\frac{\binom{n/2}{\lceil 2\sqrt{n}\rceil}}{n^{\lceil 2\sqrt{n}\rceil}} \leq \frac{(n/2)^{\lceil 2\sqrt{n}\rceil} \cdot e^{\lceil 2\sqrt{n}\rceil}/(\lceil 2\sqrt{n}\rceil)^{\lceil 2\sqrt{n}\rceil}}{n^{\lceil 2\sqrt{n}\rceil}} = O\left(\frac{1}{(\sqrt{n})^{2\sqrt{n}}}\right) = O\left(\frac{1}{n^{\sqrt{n}}}\right)$$

where we used the standard inequality $\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$ for all $k, n \in \mathbb{N}$ with $k \leq n$. By a union bound, the probability that a failure occurs in some given generation $t \leq T$ is at most $O(\lambda^*/n^{\sqrt{n}})$ where $\lambda^* := \max\{\lambda_t \mid t \leq T\} \leq \lambda = O(\mathrm{poly}(n))$. Since $\mathrm{E}[T] = O(\mu n\sqrt{n})$, again by a union bound, a failure occurs with probability $O(\lambda^*\mu n\sqrt{n}/n^{\sqrt{n}}) = o(1)$ within T generations. If a failure occurs in some generation $t \leq T$, and there is still no search point $x \in P_{t+1} \cap W$, we repeat the above arguments in a new period of generations of length T, where we redefine "left" and "right" individuals with respect to generation $T+1$ as follows: We call an individual y *left* if $|y^2|_1 < n/4 - \sqrt{n}$ in generation $T+1$ and otherwise *right*. Then, in generation $t' > T+1$, we call an individual *left*, if it is the offspring of a left individual and otherwise, if it is the offspring of a right individual, call it *right*. The expected number of periods is $1 + o(1)$, which proves the lemma, since the expected number of fitness evaluations to enter W is at most $O(\mu\lambda n\sqrt{n})$, since $\lambda_t \leq \lambda$ for all $t \geq 0$. □

Lemma 6. *Consider a generation t of GSEMO or NSGA-II-DYN for any choice of g and h when optimizing* CLIMB_ℓ *for $\ell \in \{1, 3, \ldots, n/2+1\}$, and any threshold $r \geq 1$. Suppose that $\delta_t := \max\{i \in \{0, \ldots, n/2\} \mid$ there is $x \in P_t$ with $|x^1|_1 = i$ and $f(x) \neq 0\} \in \{0, \ldots, n/2-1\}$ which means that there exists $x \in P_t$ with nonzero fitness, but no Pareto optimal search point has been found yet. If an individual $x \in P_t$ with $|x^1|_1 = \delta_t$ is chosen as a parent, then x mutates to a y with $|y^1|_1 > |x^1|_1$ and $f(y) \neq 0$ with probability at least $(n/2 - \delta_t)/(4en)$. This increases δ_t.*

Proof. One can create a z with $|z^2|_1 > \delta_t$ by flipping a zero in the first half of x, and flipping a one in the second half of x if $|x^2|_1 > \frac{n}{4}$, a zero there if $|x^2|_1 < \frac{n}{4}$, and no bit there if $|x^2|_1 = \frac{n}{4}$. This happens with probability at least

$$\frac{n/2-\delta_t}{n} \cdot \frac{n/4}{n} \cdot \left(1-\frac{1}{n}\right)^{n-2} \geq \frac{n/2-\delta_t}{4en}.$$

We still have to argue why $y := z$ has fitness distinct from zero or, in other words, why $(\frac{1}{2} - \frac{\ell-1}{n})|y^1|_1 \leq |y^2|_1 \leq \frac{n}{2} - (\frac{1}{2} - \frac{\ell-1}{n})|y^1|_1$ holds. Note that $(\frac{1}{2} - \frac{\ell-1}{n})|x^1|_1 \leq |x^2|_1 \leq \frac{n}{2} - (\frac{1}{2} - \frac{\ell-1}{n})|x^1|_1$. So if $|x^2|_1 < \frac{n}{4}$ then $|y^2|_1 = |x^2|_1 + 1$, and we obtain

$$\left(\frac{1}{2} - \frac{\ell-1}{n}\right)|y^1|_1 = \left(\frac{1}{2} - \frac{\ell-1}{n}\right)(|x^1|_1 + 1) = \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|x^1|_1 + \frac{1}{2} - \frac{\ell-1}{n}$$
$$\leq \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|x^1|_1 + 1 \leq |x^2|_1 + 1 = |y^2|_1,$$

as well as

$$|y^2|_1 \leq \frac{n}{4} \leq \frac{n}{2} - \frac{n}{4} + \frac{\ell-1}{2} = \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)\frac{n}{2} \leq \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|y^1|_1,$$

since $|y^1|_1$ is naturally bounded by $n/2$ from above. In a similar way, if $|x^2|_1 > \frac{n}{4}$ then $|y^2|_1 = |x^2|_1 - 1 \geq \frac{n}{4}$ and therefore

$$\left(\frac{1}{2} - \frac{\ell-1}{n}\right)|y^1|_1 \leq \frac{n}{4} - \frac{\ell-1}{2} \leq \frac{n}{4} \leq |y^2|_1.$$

Further, since $|x^2|_1$ is naturally bounded by $n/2$ from above,

$$|y^2|_1 = |x^2|_1 - 1 \leq \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)\frac{n}{2} - 1 \leq \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|x^1|_1 - 1$$
$$\leq \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)(|x^1|_1 + 1) = \frac{n}{2} - \left(\frac{1}{2} - \frac{\ell-1}{n}\right)|y^1|_1,$$

proving the lemma, since by Lemma 2, an individual y with increased δ_t survives. □

Lemma 7. *Consider NSGA-II-DYN optimizing* CLIMB_ℓ *for* $\ell \in \{1, 3, \ldots, n/2+1\}$ *with population sizes* $\mu_0 = 1$ *and* $\mu_{t+1} = \min\{\mu_t + \lambda_t, 4|S_t|\}$*, where* $1 \leq \lambda_t \leq \mu_t$ *is the offspring population size, and any threshold* $r \geq 4|S|$*. For* $i \in \{0, \ldots, n/2-1\}$ *define* $p_i := (n/2 - i)/(4en)$*. Suppose that all individuals* $x \in P_0$ *have nonzero fitness and satisfy* $|x^1|_1 \geq 19n/80$*. Let* $\mu_t^{(i)}$ *denote the maximum population size occurring in a generation* t *when* $\delta_t = i$ *holds. Then* $\mathrm{E}\left[\mu_t^{(i)}\right] \leq 320/p_i$*.*

Proof. Note that $\mu_0 = 1, \mu_1 = 2$, $\mu_2 = 4$ and $\mu_\ell \geq 4$ for all $\ell \geq 3$. Note also that $\mu_{t+1} \leq 4|S_t|$ where S_t is a maximum set of mutually incomparable non-dominated solutions in R_t and hence, at least $\mu_{t+1}/4 \leq |S_t|$ many individuals from P_{t+1} are non-dominated (since by the non-dominated sorting procedure at least $\min\{\mu_{t+1}, |S_t|\} = \min\{\mu_t + \lambda_t, 4|S_t|, |S_t|\} = |S_t|$ many non-dominated individuals are taken into P_{t+1}). Also recall that S_t is a maximum set of mutually incomparable solutions from R_t. Hence, for each $t \geq 0$, at least $\mu_t/4$ individuals are non-dominated. For generation t define $\delta_t := \max\{i \in [n/2] \mid \text{there is an } x \in P_t \text{ with } |x^1|_1 = i \text{ and } f(x) \neq 0\}$. Let $i^* := \delta_0 \geq 19n/80$. Then, the expected number of fitness evaluations required for creating an individual y with $|y^1|_1 > i^*$ is at most $4/p_{i^*}$, since at least $\mu_t/4$ individuals from P_t are non-dominated and hence, satisfy $|x^1|_1 = i^*$. Note that the probability is at least $1/4$ to choose such an individual as a parent in one trial. So $\mathrm{E}\left[\mu_t^{(i^*)}\right] \leq 1 + 4/p_{i^*} \leq 320/p_{i^*}$, where the additive term 1 accounts for the evaluation of the initial individual. For an induction on $i \in \{i^*, \ldots, n/2-2\}$, suppose that $\mathrm{E}\left[\mu_t^{(i)}\right] \leq 320/p_i$. The goal is to show that $\mathrm{E}\left[\mu_t^{(i+1)}\right] \leq 320/p_{i+1}$. Denote by $\mu_t^{(i,\mathrm{up})}$ the population size in generation t after $\delta_t = i$ increased in generation $t-1$. Denote by $\lambda_{t-1}^{(i,\mathrm{up})}$ the number of offspring created in generation $t-1$. We claim the following.

Lemma 14. *In expectation, at most* $19E[\lambda_{t-1}^{(i,up)}]/80 + 1$ *individuals* x *from* P_t *have nonzero fitness and satisfy* $|x^2|_1 > \delta_t$*.*

Proof. We show that an individual $x \in P_t$ with $|x^1|_1 \geq 19n/80$ creates an offspring y with $|y^1|_1 > \delta_t$ with probability at most $19/80$. Then the result follows, since the expected number of such offspring y is at most $19\lambda_{t-1}^{(i,\mathrm{up})}/80$, and there is at least one offspring z with nonzero fitness satisfying $|z^1|_1 > \delta_t$. If $|x^1|_1 \geq 21n/80$, the probability follows by a union bound, since it requires to flip a zero bit in the first half of x, and there are at most $n/2 - 21n/80 = 19n/80$ zero bits to flip there. So suppose that $19n/80 \leq |x^1|_1 < 21n/80$. Fix $L = \lfloor 19n/80 \rfloor$ positions with ones and $L' = \lfloor 19n/80 \rfloor$ positions of zeros in the first half of x. Denote by C the remaining positions in the first half. Then, to increase the number of ones there, one has to flip a bit in the region C, or to create an offspring with more ones than zeros in the region $L \cup L'$. Suppose the latter happens with probability p^{up}. Note that this probability is the same as the probability of creating an offspring with more zeros than ones in $L \cup L'$ by symmetry. Flipping no bits in $L \cup L'$ happens with probability $(1-1/n)^{2|L|} \geq (1-1/n)^{n/2} \geq 1/\sqrt{e} - o(1)$. Since we have $2p^{\mathrm{up}} + (1-1/n)^{2|L|} = 1$, we can estimate $p^{\mathrm{up}} \leq 1/2 - (1-1/n)^{2|L|}/2 \leq 1/2 - 1/(2\sqrt{e}) + o(1) \leq 1/5$ for n sufficiently large. Since $|C| \leq n/40 + 2$, we see by a union bound that a bit in region C is flipped with probability at most $1/40 + o(1)$. Hence, we obtain for the probability to increase $|x^1|_1$ the upper bound $1/2 - 1/(2\sqrt{e}) + 1/40 + o(1) < 1/5 + 1/40 = 9/40 < 19/80$ for n sufficiently large, proving the lemma. □

With Lemma 14, we see that $\mathrm{E}\left[\mu_t^{(i,\mathrm{up})}\right] \leq 4 \cdot 19E[\lambda_{t-1}^{(i,\mathrm{up})}]/80 + 4 = 76\lambda_{t-1}^{(i,\mathrm{up})}/80 + 4$ (since $\mu_t^{(i,\mathrm{up})}$ is bounded by $4|S_t|$ and $\mathrm{E}\left[|S_t|\right] \leq \mathrm{E}\left[|F_{t+1}^1|\right] \leq 19\mathrm{E}\left[\lambda_{t-1}^{(i,\mathrm{up})}\right]/80 + 1$ since S_t is the maximum number

of mutually incomparable non-dominated solutions in R_t which is a subset of F^1_{t+1}). Considering also at most $4/p_{i+1}$ additional offspring creations until we increase $\delta_t = i+1$ again, we see that

$$\begin{aligned}\mathrm{E}\left[\mu_t^{(i+1)}\right] &\leq \mathrm{E}\left[\mu_t^{(i,\mathrm{up})}\right] + \frac{4}{p_{i+1}} \leq \frac{76\mathrm{E}\left[\lambda_{t-1}^{(i,\mathrm{up})}\right]}{80} + 4 + \frac{4}{p_{i+1}} \\ &\leq 76 \cdot \frac{320}{80p_i} + 4 + \frac{4}{p_{i+1}} \leq \frac{304}{p_{i+1}} + 4 + \frac{4}{p_{i+1}} = \frac{308}{p_{i+1}} + 4 \leq \frac{320}{p_{i+1}},\end{aligned}$$

proving the lemma. □

Lemma 8. *Consider GSEMO or NSGA-II-DYN optimizing* CLIMB_ℓ *for* $\ell \in \{1, 3, \ldots, n/2+1\}$ *and suppose there is a generation* t_0 *such that* P_{t_0} *contains at least one Pareto optimal solution. Then the following statements hold.*

(1) GSEMO needs $O(\ell^2 \ln(en/(n-2\ell+3)))$ *fitness evaluations in expectation to cover the Pareto front of* CLIMB_ℓ.

(2) For NSGA-II-DYN, suppose that there are $\mu, \lambda \in \mathbb{N}$ *with* $1 \leq \mu_t \leq \mu$ *and* $1 \leq \lambda_t \leq \lambda$ *for all* $t \geq t_0$. *Further, suppose that* $\mu_{t+1} \geq \min\{\mu_t + \lambda_t, 4|S_t|\}$ *for all* $t \geq 0$. *Then NSGA-II-DYN needs* $O(\lambda\ell + \mu\ell \ln(en/(n-2\ell+3)))$ *fitness evaluations in expectation to cover the Pareto front of* CLIMB_ℓ.

Proof. Without loss of generality, we can assume that $t_0 = 0$. For $t \geq 0$, let $\mathcal{F}_t$ denote the set of all Pareto optimal individuals in the current population P_t. Note that $\mathcal{F}_t \neq \emptyset$. As long as the Pareto front is not yet fully covered, there always exists an uncovered fitness vector v on the Pareto front that has a covered neighbor w. That is, $|v_1 - w_1| = 1$ and there exists $x \in P_t$ such that $f(x) = w = (n^2/2 + |x^2|_1, n^2/2 + n/2 - |x^2|_1)$, while there is no $y \in P_t$ such that $f(y) = v$. Call such a v an *adjoining* fitness vector. All solutions y covering v satisfy $||y^2|_1 - |x^2|_1| = 1$. So in a trial, v can be covered by choosing x as a parent (which happens with probability at least $1/|P_t|$ for both GSEMO and NSGA-II-DYN) and flipping a one bit to zero in the second half of x if $|y^2|_1 = |x^2|_1 - 1$ or flipping a zero bit to one in the second half of x if $|y^2|_1 = |x^2|_1 + 1$, while leaving all other bits unchanged. If $|y^2|_1 \neq n/2$, a successful mutation in the former case happens with probability at least $|x^2|_1/n \cdot (1-1/n)^{n-1} \geq |x^2|_1/(en) = (|y^2|_1 + 1)/(en)$ and if $|y^2|_1 \neq 0$, in the latter with probability at least $(n/2 - |x^2|_1)/n \cdot (1-1/n)^{n-1} \geq (n/2 - |x^2|_1)/(en) = (n/2 - |y^2|_1 + 1)/(en)$. If $|y^2|_1 \in \{0, n/2\}$, the correct bit in x to cover v is flipped with probability at least $1/n \cdot (1-1/n)^{n-1} \geq 1/(en) = \min\{|y^2|_1 + 1, n/2 - |y^2|_1 + 1\}/(en)$. In either case, v is covered in a single trial with probability at least $\min\{|y^2|_1 + 1, n/2 - |y^2|_1 + 1\}/(en|P_t|)$.

GSEMO: Note that $1 \leq |P_t| \leq \ell$. Since only one solution in each generation is created, and at most ℓ distinct fitness vectors on the Pareto front have to be covered when they are adjoining fitness vectors, GSEMO requires

$$\sum_{j=\frac{n}{4}-\frac{\ell-1}{2}}^{\frac{n}{4}+\frac{\ell-1}{2}} \frac{en\ell}{\min\{j+1, n/2-j+1\}} \leq 2\sum_{j=\frac{n}{4}}^{\frac{n}{4}+\frac{\ell-1}{2}} \frac{en\ell}{n/2-j+1} = 2en\ell \sum_{j=\frac{n}{4}-\frac{\ell-1}{2}+1}^{\frac{n}{4}+1} \frac{1}{j}$$

fitness evaluations in expectation to cover the whole Pareto front. We estimate the sum on the right separately for the following two subcases (i) and (ii).

(i): Suppose that $\ell < n/4$. Then $n/4 - (\ell-1)/2 + 1 \geq n/8$ and hence

$$\sum_{j=\frac{n}{4}-\frac{\ell-1}{2}+1}^{\frac{n}{4}+1} \frac{1}{j} \leq \sum_{j=\frac{n}{4}-\frac{\ell-1}{2}+1}^{\frac{n}{4}+1} \frac{8}{n} = \frac{\ell+1}{2} \cdot \frac{8}{n} \leq \frac{8\ell}{n}$$

and the total expected runtime to cover the Pareto front is $O(\ell^2) = O(\ell^2 \ln(en/(n-2\ell+3)))$.

(ii): Suppose that $\ell \geq n/4$. Then for $\alpha := \max\{1, n/4 - (\ell-1)/2\}$

$$\sum_{j=\frac{n}{4}-\frac{\ell-1}{2}+1}^{\frac{n}{4}+1} \frac{1}{j} = \sum_{j=1}^{\frac{n}{4}+1} \frac{1}{j} - \sum_{j=1}^{\frac{n}{4}-\frac{\ell-1}{2}} \frac{1}{j} \leq \ln\Big(\frac{n}{4}+1\Big) + 1 - \ln(\alpha) = \ln\left(\frac{e(n+4)}{4\alpha}\right)$$

$$= O\left(\ln\Big(\frac{en}{n-2\ell+3}\Big)\right),$$

resulting in an expected runtime of $O(n\ell\ln(en/(n-2\ell+3))) = O(\ell^2\ln(en/(n-2\ell+3)))$ for covering the Pareto front of CLIMB_ℓ.

NSGA-II-DYN: Following the same arguments as for GSEMO, we obtain the following bound on the total number of fitness evaluations. The additional term λ accounts for the remaining fitness evaluations needed to complete a generation t in which an adjoining fitness vector is covered. Note also that Lemma 2 applies, since $\mu_t \geq \min\{4|S_t|, \mu_t + \lambda_t\}$. Thus, once vectors on the Pareto front are covered, they cannot become uncovered. Since $\lambda_t \leq \lambda$ and $|P_t| \leq \mu_t \leq \mu$, we obtain the upper bound on the expected number of fitness evaluations

$$\sum_{j=\frac{n}{4}-\frac{\ell-1}{2}}^{\frac{n}{4}+\frac{\ell-1}{2}} \left(\frac{\mu en}{\min\{j+1, n/2-j+1\}} + \lambda\right) \leq \lambda\ell + 2\sum_{j=\frac{n}{4}}^{\frac{n}{4}+\frac{\ell-1}{2}} \frac{\mu en}{n/2-j+1}$$

$$= \lambda\ell + 2\mu en \sum_{j=\frac{n}{4}-\frac{\ell-1}{2}+1}^{\frac{n}{4}+1} \frac{1}{j}.$$

So in a similar way as for GSEMO above, we obtain that NSGA-II-DYN covers the Pareto front in expected $O(\lambda\ell + \mu\ell\ln(\frac{en}{n-2\ell+3}))$ fitness evaluations which completes the proof. □

Lemma 10. *GSEMO and NSGA-II-DYN complete Phase 1 in expected $O(n\sqrt{n}e^{\sqrt{n}\ln(n)})$ fitness evaluations, Phase 2 in expected $O(n^2)$ fitness evaluations, and Phase 3 in expected $O(\ell^2\ln(en/(n-2\ell+3)))$ fitness evaluations.*

Proof. We consider all three subphases separately.

Phase 1: Create an individual with nonzero fitness. During this phase, the maximum number of mutually incomparable non-dominated solutions is one, since all search points have fitness zero. Consequently, the population size is one for GSEMO and at most four for NSGA-II-DYN. In the latter case, from the third generation onward, we have $\mu_t = 4$, so the population size μ is constant as long as no individual with nonzero fitness is created. Thus, Lemma 5 applies with $\mu = 1$ for GSEMO and $\mu = 4$ for NSGA-II-DYN (from the third generation onward). After $O(n\sqrt{n})$ fitness evaluations, an individual x is created with $|x^2|_1 \in W = \{x \in \{0,1\}^n \mid n/4 - \sqrt{n} \leq |x^2|_1 \leq n/4 + \sqrt{n}\}$. After creating x in generation T, the probability that x survives is at least $1/2$. Moreover, with probability at least $n^{-\sqrt{n}}/4 = e^{-\sqrt{n}\ln(n)}/4$, an individual y with $|y^2|_1 = n/4$ is created in generation $T+1$ by choosing an individual $z \in W$ as a parent (probability at least $1/4$) and flipping at most $\lfloor\sqrt{n}\rfloor$ bits (probability at least $n^{-\sqrt{n}} = e^{-\sqrt{n}\ln(n)}$). Such a y has nonzero fitness. If that individual y is not created in $T+1$ generations, we repeat. The expected number of repetitions required is at most $O(e^{\sqrt{n}\ln(n)})$, and each repetition requires $O(n\sqrt{n})$ fitness evaluations in expectation. By Wald's equation, the expected number of fitness evaluations required to finish this subphase is at most $O(n\sqrt{n}e^{\sqrt{n}\ln(n)})$.

Phase 2: Create a Pareto optimal individual. We consider both algorithms separately.
GSEMO: As in the Case 1, where only one good individual is initialized, P_t consists only of good individuals x with $|x^1|_1 = \delta_t$ and hence, such an individual is chosen with probability one as a parent. Thus, $i = \delta_t$ increases with probability at least $p_i = (n/2 - i)/(4en)$ in each generation. So the expected number of generations to complete this phase is at most $\sum_{i=1}^{n/2-1} \frac{1}{p_i} = O(n\ln(n))$.

NSGA-II-DYN: As in Case 1, we have $\mu_{t+1} \leq 4|S_t|$ and hence at least $\mu_{t+1}/4$ individuals from P_{t+1} are non-dominated. Therefore, an individual x with $|x^1|_1 = i = \delta_t$ is chosen as a parent with probability at least $1/4$. So we need $4/p_i$ fitness evaluations in expectation to create an individual y with $|y^1|_1 > \delta_t$ and nonzero fitness. To finish the iteration where such a y is created, we need at most $\lambda_t \leq \mu_t \leq 2n+4$ additional fitness evaluations. So the total expected number of fitness evaluations required to complete this phase is at most

$$\sum_{i=1}^{n/2-1} \Big(\frac{4}{p_i} + 2n + 4\Big) = O(n\ln(n) + n^2) = O(n^2).$$

Phase 3: Cover the Pareto front. Note also, for NSGA-II-DYN, we have that $\mu_t \leq 4\ell$ for all $t \geq 0$. Thus, this phase can be analyzed in the same way as Phase 2 in Case 1. Consequently, both algorithms require an expected number of $O(\ell^2 \ln(en/(n-2\ell+3)))$ fitness evaluations to cover the Pareto front of CLIMB_ℓ. □

Theorem 11. *Both GSEMO and NSGA-II-DYN, with $\mu_0 = 1$, any choice of g_t and h_t, and any threshold $r \geq 1$, require $\Omega(n \ln(n))$ fitness evaluations in expectation to reach the Pareto front when optimizing $f := \text{CLIMB}_\ell$ for $\ell \in \{1, 3, \dots, n/2+1\}$.*

Proof. We first show that each of the following three failure events occurs with probability at most $o(1)$. Conditioned on none of these events occurring, we then show that creating a Pareto-optimal individual requires $\Omega(n \ln n)$ fitness evaluations in expectation. Thus, the overall expected number of fitness evaluations required by either algorithm to optimize CLIMB_ℓ is at least $(1+o(1))\Omega(n \ln n) = \Omega(n \ln n)$.

The first failure event is that the initial search point x satisfies $|x^1|_1 \geq n - 3\sqrt{n}$. For both algorithms, this occurs with probability at most $e^{-\Omega(n)}$. The second failure event is that, within $O(n \ln n)$ fitness evaluations, at least $2\sqrt{n}$ bits in the first half of some search point $x \in P_t$ are flipped. By a union bound, this happens with probability at most

$$O(n \ln(n)) \binom{n/2}{\lceil 2\sqrt{n} \rceil} / (n^{\lceil 2\sqrt{n} \rceil}) = O(n \ln(n) / n^{\sqrt{n}}) = o(1).$$

Compare also with the proof of Lemma 5 for a similar argument. A third failure event is that, for $m \in \mathbb{N}$ with $m \leq \lceil n - 3\sqrt{n} \rceil$, at least $3+m$ zero bits are flipped in the first half of a search point x when $n - 3\sqrt{n} - m \leq |x^1|_1$ within $O(n \ln(n))$ fitness evaluations, happening with probability at most

$$O(n \ln(n)) \cdot \binom{\lceil 3\sqrt{n} \rceil + m}{3+m} / n^{3+m} \leq O(n \ln(n)) \cdot \binom{\lceil 3\sqrt{n} \rceil}{3} / n^3 = O(n \ln(n) \cdot n^{-1.5}) = o(1).$$

So, suppose that none of these occur. Let

$$\delta_t := \max\{i \in \{0, \dots, n/2\} \mid \text{there is } x \in P_t \text{ with } |x^1|_1 = i\}.$$

Note that there is a time t_0 where $n - 3\sqrt{n} \leq \delta_{t_0} \leq n - \sqrt{n}$. Then δ_t for $t \geq t_0$ can only be increased by at most two, and that with probability at most δ_t/n: If an individual x with $|x^1|_1 = \delta_t$ is selected as a parent, then one zero bit is flipped in the first half of x with probability at most δ_t/n. But also if $|x^1|_1 < \delta_t$ the probability to flip at least $\delta_t - |x^1|_1 + 1$ zero bits in the first half of x is not larger than δ_t/n. So, in expectation, at least n/δ_t fitness evaluations are necessary to increase δ_t, and if δ_t is increased, then by at most two. Also, it happens that $\delta_t \in \{n/2 - 2i, n/2 - 2i + 1\}$ for each $i \in [\lfloor \sqrt{n}/2 \rfloor]$, and each such δ_t has to be increased to further progress towards the Pareto front. So the expected number of fitness evaluations to increase δ_t to n is at least $\sum_{i=1}^{\lfloor \sqrt{n}/2 \rfloor} n/(2i) \geq n/2 \cdot \ln(\lfloor \sqrt{n}/2 \rfloor) = \Omega(n \ln(n))$. □

Theorem 12. *Consider $f := \text{CLIMB}_\ell$ for $\ell \in \{1, 3, \dots, n/2\}$. Let $\mu, \lambda \in \mathbb{N}$ with $\mu \geq 4\ell$. Then, NSGA-II-DYN with $\mu_t = \mu$, $\lambda_t = \lambda$, and any threshold $r \geq \mu$, requires $O((\mu+\lambda)n + n \ln(n) + \mu\ell \ln(en/(n-2\ell+3)))$ fitness evaluations in expectation to cover the Pareto front of f. For $\mu = 4\ell$ and $1 \leq \lambda \leq \mu$, the stated runtime bound becomes $O(\ell n + n \ln(n) + \ell^2 \ln(en/(n-2\ell+3)))$.*

Proof. We derive the optimization process in two phases. The first phase is about reaching the Pareto front, and the second is about covering it. For reaching the Pareto front, define $\delta_t := \max\{i \in \{0, \dots, n/2-1\} \mid \text{there is } x \in P_t \text{ with } |x^1|_1 = i \text{ and } f(x) \neq 0\}$. To increase δ_t, we first create at least $\min\{\mu, \lceil n/(n/2 - \delta_t) \rceil\}$ solutions x with $|x^1|_1 = \delta_t$ and fitness distinct from zero and then, increase δ_t by applying a mutation. For the first case, suppose there are currently i such individuals x. In one trial, the probability of increasing their number by one is at least $i/\mu \cdot 1/4 = i/(4\mu)$, since we choose one of these individuals with probability at least $1/\mu$, and then clone it with probability at least $(1-1/n)^n \geq 1/4$. Hence, the expected number of trials to double the number of individuals x, or to reach μ such individuals, is at most 8μ. Consequently, the expected number of generations is $\lceil 8\mu/\lambda \rceil \leq 8\mu/\lambda + 1$. Let $\gamma_t := n/2 - \delta_t$. Then the expected number of generations until there are $\min\{\mu, \lceil n/\gamma_t \rceil\}$ such solutions x is $(8\mu/\lambda + 1)(\lceil \log(\min\{\mu, \lceil n/\gamma_t \rceil\}) \rceil) \leq (8\mu/\lambda + 1)(\log(\min\{\mu, \lceil n/\gamma_t \rceil\}) + 1) \leq (8\mu/\lambda + 1)(\log(n/\gamma_t) + 2)$ where log denotes the logarithm to base

2. Then, increasing δ_t happens with probability at least $\min\{\mu, \lceil n/\gamma_t \rceil\}/\mu \cdot \gamma_t/n \geq \min\{\gamma_t/n, 1/\mu\}$. Thus, the expected number of trials required to increase δ_t after creating $\lceil n/\gamma_t \rceil$ individuals is at most $\mu + n/\gamma_t$. This results in at most $\lceil \mu/\lambda + n/(\lambda\gamma_t) \rceil \leq \mu/\lambda + n/(\lambda\gamma_t) + 1$ generations. Therefore, the expected number of generations until the Pareto front is reached is bounded from above by

$$
\begin{aligned}
&\sum_{i=1}^{n/2} \Big(\Big(\frac{8\mu}{\lambda} + 1 \Big) \Big(\log\Big(\frac{n}{i} \Big) + 2 \Big) + \frac{\mu}{\lambda} + \frac{n}{\lambda i} + 1 \Big) \\
&= \Big(\frac{8\mu}{\lambda} + 1 \Big) \Big(n + \sum_{i=1}^{n/2} \log\Big(\frac{n}{i} \Big) \Big) + \frac{n\mu}{2\lambda} + \frac{n}{2} + \sum_{i=1}^{n/2} \frac{n}{\lambda i} \\
&\leq \Big(\frac{8\mu}{\lambda} + 1 \Big) \Big(n + \log\Big(\frac{n^{n/2}}{(n/2)!} \Big) \Big) + \frac{n\mu}{2\lambda} + \frac{n}{2} + \frac{n(\ln(n)+1)}{\lambda} \\
&\leq \Big(\frac{8\mu}{\lambda} + 1 \Big) \Big(n + \log\Big(\frac{(en)^{n/2}}{(n/2)^{n/2}} \Big) \Big) + \frac{n\mu}{2\lambda} + \frac{n}{2} + \frac{n(\ln(n)+1)}{\lambda} \\
&\leq \Big(\frac{8\mu}{\lambda} + 1 \Big) \Big(n + \frac{n}{2} \log(2e) \Big) + \frac{n\mu}{2\lambda} + \frac{n}{2} + \frac{n(\ln(n)+1)}{\lambda} \\
&= O\left(n + \frac{n\mu}{\lambda} + \frac{n \ln(n)}{\lambda} \right).
\end{aligned}
$$

So the expected number of fitness evaluations until the Pareto front is reached in $O(n\lambda + n\mu + n\ln(n))$.

So it remains to cover the Pareto front. To this end, we can apply Lemma 8(2), since $\mu \geq 4\ell \geq \min\{\mu + \lambda, 4|S_t|\}$, and obtain that the Pareto front of CLIMB_ℓ is covered in $O(\lambda\ell + \mu\ell \ln(en/(n - 2\ell + 3)))$ fitness evaluations. This proves the theorem by summing the runtimes for the hill climbing phase and the phase covering the Pareto front. □

Theorem 13. *For $1 \leq \mu = O(\text{poly}(n))$ and $1 \leq \lambda = O(\text{poly}(n))$ the NSGA-II-DYN with fixed population size μ and offspring population size λ (if $g = \mu$ and $h = \lambda$) needs at least $\Omega(\mu n)$ fitness evaluations in expectation to find a Pareto optimal search point of $f := \text{CLIMB}_\ell$ for every $\ell \in \{1, 3, 5, \ldots, n/2 + 1\}$. For $\mu = \ell$, the expected number of fitness evaluations becomes $\Omega(\ell n)$.*

Proof. By Lemma 3 every individual x is initialized with $(|x^1|_1, |x^2|_1) \in [\frac{19n}{80}, \frac{21n}{80}] \times [\frac{n}{8}, \frac{3n}{8}]$ with probability at least $1 - \mu e^{-\Omega(n)} = 1 - e^{-\Omega(n)}$. These individuals have nonzero fitness. Suppose this happens. Let $P_0 := \{x_0^1, \ldots, x_0^\mu\}$. Denote by $\text{tree}_t(x_0^i)$ the family tree with root x_0^i. First, we show that after t generations, we have $\Pr(\text{depth}(\text{tree}_t(x_0^i)) \geq 3t\lambda/\mu) \leq \mu e^{-\Omega(t\lambda/\mu)}$ for all $i \in [\mu]$. To this end, we consider the probability of generating a path of length k consisting of nodes with labels $(t_1, i_1), \ldots, (t_k, i_k)$ as the second and third components for $0 \leq t_1 < \ldots < t_k \leq t$ and $i_1, \ldots, i_k \in [\lambda]$. Note that there are $\binom{t}{k}$ ways to label such a path by the second component, and λ^k ways to label it with respect to the third component. So the number of such paths of length k is at most $\binom{t}{k} \cdot \lambda^k$. For $m > 1$, the probability of extending such a path by a node with label (t_m, i_m) is at most $1/\mu$, since the corresponding individual of the node with label (t_{m-1}, i_{m-1}) must be chosen as a parent in the i_m-th iteration of the for-loop in generation t_m. Thus, the probability of creating such a path is at most $(1/\mu)^{k-1}$. Hence, by a union bound, the probability of creating any path of length k in t generations is at most $\mu \cdot \binom{t}{k} \cdot (\lambda/\mu)^k$. If $k \geq 3t\lambda/\mu$, then, using the standard bound $\binom{t}{k} \leq (et/k)^k$,

$$
\mu \binom{t}{k} \frac{\lambda^k}{\mu^k} \leq \mu \left(\frac{et\lambda}{k\mu} \right)^k \leq \mu \left(\frac{e}{3} \right)^k = \mu e^{-\Omega(t\lambda/\mu)}.
$$

Suppose that this does not happen. In particular, all μ trees in the forest have depth at most $\alpha := \min\{t, 3t\lambda/\mu\}$. So $k \leq \alpha$. We then bound the Hamming distance between $y_0 := x_0^i$ and solutions corresponding to nodes in the tree with root y_0 as follows. We call a path to a leaf at time t *bad* if a solution with Hamming distance of at least $n/5$ to y_0 in the second half of y_0 is created along that path. When adding the m-th node with triple (y_m, t_m, i_m) for some $i_m \in [\lambda]$ to a path, we see that $\text{E}\left[H(y_{m-1}^2, y_m^2)\right] = 1/2$ since the expected number of flipped bits in the second half of y_{m-1} is $1/2$, when applying standard bit mutation. Hence, the expected change in Hamming distance along that path is at most $k/2 \leq 3t\lambda/(2\mu)$. For $t \leq 4n\mu/(54\lambda)$, we obtain that the expected

change in Hamming distance is at most $n/9$. Thus, using a standard Chernoff bound, for $\delta = 4/5$, every $m \in [k]$, and $t \leq 4n\mu/(54\lambda)$, we have that $\Pr(H(y_0^2, y_m^2) \geq n/5) = \Pr(H(y_0^2, y_m^2) \geq (1+\delta)n/9) \leq e^{-\mathrm{E}[X]\delta^2/3} \leq e^{-16n/675}$. By setting $t = \lfloor 4n\mu/(1350\lambda) \rfloor$, we obtain by a union bound over all possible paths of length at most $\alpha \leq 12n/1350$ and all nodes on such a path that the probability of generating a bad path is at most

$$\alpha \sum_{k=1}^{\min\{\alpha,t\}} \mu \binom{t}{k} \frac{\lambda^k}{\mu^k} \cdot e^{-16n/675} \leq \alpha\mu \sum_{k=1}^{\alpha} \left(\frac{4en}{1350k}\right)^k \cdot e^{-16n/675} = \alpha\mu e^{-\Omega(n)} = e^{-\Omega(n)}$$

within t generations. This concludes the proof, since the existence of such a path is necessary for creating a Pareto optimal search point. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the paper's contributions are summarized in the abstract, as well as in a subsection of the introduction called "Our contributions".

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: in the conclusions we discuss some limitations of our approaches, which could be avenues for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This work is purely theoretical, and all statements are rigorously proven in detail. However, due to space restrictions, some of the proofs are shifted to the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: This work is purely theoretical and hence, does not contain experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: This work is purely theoretical and hence, does not contain experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [N/A]

Justification: This work is purely theoretical and hence, does not contain experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: This work is purely theoretical and hence, does not contain experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: This work is purely theoretical and hence, does not contain experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and is therefore not harmful for anyone in the society or for the environment.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not address societal impact, neither in positive nor negative ways. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not address any assets. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not address any assets. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not contain crowdsourcing experiments and research with human subjects. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not contain any type of risks mentioned above, and hence, does not need any approvals. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This paper addresses runtime analysis of multiobjective evolutionary algorithms and it does therefore not require the usage of LLMs as a core method of the research. It instead aims to conduct runtime analysis to understand when and why MOEAs perform well, with the goal of guiding practitioners in designing more advanced versions of these algorithms.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.