A Many-Objective Problem Where Crossover is Provably Indispensable

Andre Opris

Chair of Algorithms for Intelligent Systems, University of Passau, Passau, Germany
andre.opris@uni-passau.de

Abstract

This paper addresses theory in evolutionary multiobjective optimisation (EMO) and focuses on the role of crossover operators in many-objective optimisation. The advantages of using crossover are hardly understood and rigorous runtime analyses with crossover are lagging far behind its use in practice, specifically in the case of more than two objectives. We present a many-objective problem class together with a theoretical runtime analysis of the widely used NSGA-III to demonstrate that crossover can yield an exponential speedup on the runtime. In particular, this algorithm can find the Pareto set in expected polynomial time when using crossover while without crossover it requires exponential time to even find a single Pareto-optimal point. To our knowledge, this is the first rigorous runtime analysis in many-objective optimisation demonstrating an exponential performance gap when using crossover for more than two objectives.

Introduction

Evolutionary multi-objective algorithms (EMOAs) mimic principles from natural evolution as mutation, crossover (recombination) and selection to evolve a population of solutions dealing with multiple conflicting objectives to explore a Pareto optimal set. Those have been frequently applied to a variety of multi-objective optimisation problems and also have several applications in practice (Deb 2001; Coello, Veldhuizen, and Lamont 2013) such as scheduling problems (Ishibuchi and Murata 1998), vehicle design (Xingtao et al. 2008) or practical combinatorial optimisation problems (Li et al. 2024). They are also widely used in machine learning, artificial intelligence, and various fields of engineering (Qu et al. 2021; Luukkonen et al. 2023; Sharma and Chahar 2022). Particularly, in real world scenarios, there exist many problems with four or more objectives (Chikumbo, Goodman, and Deb 2012; Coello and Lamont 2004). Thus, it is not unexpected that the study of EMOAs became a very important area of research in the last decades, especially for many objectives. However, when the number of objectives increases, the size of the Pareto front and the number of incomparable solutions can grow exponentially and therefore, covering a high dimensional front, is a difficult task. There are already strong differences between two and more objectives. NSGA-II (Deb et al. 2002), the most used EMOA, optimises bi-objective problems efficiently (see (Köppen and Yoshida 2007) for empirical results or (Zheng, Liu, and Doerr 2022; Dang et al. 2023b; Doerr and Qu 2022; Dang et al. 2023a; Dang, Opris, and Sudholt 2024) for rigorous runtime analyses) while it perform less when dealing with three or more objectives (see (Chaudhari et al. 2022) for empirical results or (Zheng and Doerr 2024a) for rigorous negative results). The reason is that the so-called *crowding distance*, the tie breaker in NSGA-II, induces a sorting only for two objectives and therefore, Pareto-optimal search points can be lost between generations. Hence, Deb and Jain (2014) proposed NSGA-III, a refinement of the very popular NSGA-II, designed to handle more than two objectives, and instead of the crowding distance, uses reference points (previously set by the user) to guarantee that the solution set is well-distributed across the objective space. In particular, Deb and Jain (2014) empirically showed that NSGA-III can solve problems between 3 and 15 objectives efficiently. Due to its versatileness, it gained significant traction ($\sim$5500 citations) and now has sereval applications (Tang et al. 2024; Bhesdadiya et al. 2016; Gu, Xu, and Li 2022). However, theoretical breakthroughs on its success have only occurred recently. The first rigorous runtime analyses of the state of the art NSGA-III were only published at IJCAI 2023 (Wietheger and Doerr 2023) and GECCO 2024 (Opris et al. 2024) and hence, its theoretical understanding is still substantially behind its achievements in practice. For example, there are several empirical results on the usefulness of crossover in many objectives, particularly for NSGA-III (Yi et al. 2020; Sekine and Tatsukawa 2018), but we are not aware of any such theoretical result addressing rigorous runtime analysis in more than two objectives. This is remarkable, because crossover is a very useful operator in evolutionary computation. In the bi-objective setting, particularly for NSGA-II, Doerr and Qu (2023) proved that crossover guarantees a speedup of $O(n)$ on a certain class of functions or Dang et al. (2023b) showed that this speedup is even exponential on a more artificial benchmark. The latter is based on a RealRoyalRoad function constructed by Jansen and Wegener (2005) for single objective optimisation, where NSGA-II can optimise this benchmark in expected $O(n^3 + \mu/n)$ generations. But particularly rigorous mathematical proofs on NSGA-III provide restrictions and capabilities on how NSGA-III really works and are able to guide practitioners.

Our contribution: We build on the considerations of (Dang et al. 2023b) to more than two objectives, and investigate an example of a pseudo-Boolean function m-$\mathrm{RR}_{\mathrm{MO}}$ for a constant number m of objectives serving as a "royal road" where the use of crossover significantly improves performance. When crossover is turned off, NSGA-III requires expected exponential time to find a single Pareto-optimal point. In sharp contrast, NSGA-III using crossover can find the Pareto set of m-$\mathrm{RR}_{\mathrm{MO}}$ in expected $O(n^3)$ generations. This runtime does not asymptotically depend on the number m of objectives, and even not on the population size μ in contrast to (Dang et al. 2023b) for the bi-objective case. For large population sizes (i.e. $\mu = \Omega(n^4)$), this is an improvement by a factor of $\Omega(\mu/(n^4))$ compared to (Dang et al. 2023b). If μ grows exponentially in the number of objectives m, this factor also becomes exponentially large in m. This is typical for many-objective optimization to ensure adequate coverage of the Pareto front. For our purposes we also have to adapt the general arguments from (Opris et al. 2024) about the protection of good solutions of NSGA-III to this situation.

Related work: As already mentioned in (Dang et al. 2023b), there are several rigorous results on the usefulness of crossover on pseudo-boolean functions in single objective optimisation. An exponential performance gap in the runtime was proven by Jansen and Wegener (2005). They constructed a function REALROYALROAD where EAs without crossover need exponential time with overwhelming probability while an easy designed EA with 1-point crossover optimises REALROYALROAD in polynomial time. The reason is that REALROYALROAD yields EAs to evolve strings with all 1-bits cumulated in a single block, and then with 1-point crossover the optimal string can easily be generated. For JUMP_k, where a fitness valley of size k has to be traversed, it has been shown rigorously that *uniform* crossover gives a polynomial or superpolynomial speedup, depending on the parameter k (Jansen and Wegener 2002; Kötzing, Sudholt, and Theile 2011; Dang et al. 2017; Opris, Lengler, and Sudholt 2024). Advantages through crossover were also proven for the easy $\mathrm{ONEMAX}(x)$ problem (Sudholt 2017; Corus and Oliveto 2018; Doerr, Doerr, and Ebel 2015; Doerr and Doerr 2018) where just the number of ones in the bit string x is counted, combinatorial problems like closest string problem (Sutton 2021) or shortest paths (Doerr, Happ, and Klein 2012; Doerr et al. 2013). Also special NP hard graph problems like the k-vertex cover or k-vertex cluster problem can be optimised efficiently with variants of crossover (Sutton and Lee 2024).

There are only a few papers about EMOAs which gave a rigorous runtime analysis about this topic. A few variants of GSEMO with crossover have been studied (Qian, Yu, and Zhou 2013; Qian, Bian, and Feng 2020; Doerr, Hadri, and Pinard 2022) and the first improvement of NSGA-II with crossover on the runtime on the classical LOTZ, OMM and COCZ problems to $O(n^2)$ was provided by Bian and Qian (2022). However, they used stochastic tournament selection, a special parent selection strategy, and could not outperform Covantes Osuna et al. (2020) which used SEMO with diversity-based parent selection schemes, but without crossover. Later, in parallel independent work, Doerr and Qu (2023) and Dang et al. (2023b) showed the first improvements on the runtime of NSGA-II with crossover for classical parent selection mechanisms in the bi-objective setting. The former studied the OJZJ, a variant of the JUMP_k-benchmark for two objectives, and showed that crossover speeds up the expected runtime by a factor of n. The latter constructed a REALROYALROADMO-function, similar to Jansen and Wegener (2005), to show that crossover can give an exponential speedup on the runtime.

The theoretical analysis of NSGA-III only succeeded recently: Wietheger and Doerr (2023) conducted the first runtime analysis of NSGA-III on the 3-ONEMINMAX problem and showed that for $p \geq 21n$ divisions along each objective for defining the set of reference points, NSGA-III finds the complete Pareto front of 3-OMM in expected $O(\mu n \log(n))$ evaluations where the population size μ coincides with the size of the Pareto front of 3-OMM. Opris et al. (2024) generalised this result on more than three objectives and gave also a runtime analysis for the classical m-COUNTINGONESCOUNTINGZEROES and m-LEADINGONESTRAILINGZEROES benchmarks (Laumanns, Thiele, and Zitzler 2004) for any constant number m of objectives: NSGA-III with uniform parent selection and standard bit mutation optimises m-LOTZ in expected $O(n^{m+1})$ evaluations with a population size of $\mu = O(n^{m-1})$ and m-OMM, m-COCZ in expected $O(n^{m/2+1}\log(n))$ fitness evaluations where $\mu = O(n^{m/2})$ (coinciding with the size of the Pareto front of m-OMM and m-COCZ, respectively). They could also reduce the number of required divisions by more than a factor of 2. However, all these results do not take crossover operators into account.

Preliminaries

Let $\ln$ be the logarithm to base e and $[m] := \{1, \ldots, m\}$ for $m \in \mathbb{N}$. For a finite set A we denote by $|A|$ its cardinality. For two random variables X and Y on $\mathbb{N}_0$ we say that Y stochastically dominates X if $P(Y \leq c) \leq P(X \leq c)$ for every $c \geq 0$. The number of ones in a bit string x is denoted by $|x|_1$. The number of leading zeros in x, denoted by $\mathrm{LZ}(x)$, is the length of the longest prefix of x which contains only zeros, and the number of trailing zeros in x, denoted by $\mathrm{TZ}(x)$, the length of the longest suffix of x containing only zeros respectively. For example, if $x = 00110110110000$, then $\mathrm{LZ}(x) = 2$ and $\mathrm{TZ}(x) = 4$.

This paper is about many-objective optimisation, particularly the maximisation of a discrete m-objective function $f(x) := (f_1(x), \ldots, f_m(x))$ where $f_i : \{0,1\}^n \to \mathbb{N}_0$ for each $i \in [m]$. When $m = 2$, the function is also called *bi-objective*. Let $f_{\max}$ be the maximum possible value of f in one objective, i.e. $f_{\max} := \max\{f_j(x) \mid x \in \{0,1\}^n, j \in [m]\}$. Denote by $\vec{1} := (1, \ldots, 1) \in \mathbb{N}^m$ the unit vector. For $N \subseteq \{0,1\}^n$ let $f(N) := \{f(x) \mid x \in N\}$.

Definition 1. *Consider an m-objective function f.*

1) *Given two search points $x, y \in \{0,1\}^n$, x weakly dominates y, denoted by $x \succeq y$, if $f_i(x) \geq f_i(y)$ for all*

Algorithm 1: NSGA-III on an m-objective function f with population size μ and crossover probability p_c (Deb and Jain 2014)

Initialise $P_0 \sim \text{Unif}((\{0,1\}^n)^\mu)$
for $t := 0$ *to* ∞ **do**
  Initialise $Q_t := \emptyset$
  for $i = 1$ *to* $\mu/2$ **do**
    Sample p_1, p_2 from P_t uniformly at random
    Sample $r \sim \text{Unif}([0,1])$
    if $r \leq p_c$ **then**
      Create c_1 by 1-point crossover on p_1, p_2
      Create c_2 by 1-point crossover on p_1, p_2
    else
      Create c_1, c_2 as copies from p_1, p_2
    Create s_1, s_2 by standard bit mutation on c_1, c_2 with mutation probability $1/n$
    Update $Q_t := Q_t \cup \{s_1, s_2\}$
  Set $R_t := P_t \cup Q_t$
  Partition R_t into layers $F_t^1, F_t^2, \ldots, F_t^k$ of non-dominated fitness vectors
  Find $i^* \geq 1$ such that $\sum_{i=1}^{i^*-1} |F_t^i| < \mu$ and $\sum_{i=1}^{i^*} |F_t^i| \geq \mu$
  Compute $Y_t = \bigcup_{i=1}^{i^*-1} F_t^i$
  Choose $\tilde{F}_t^{i^*} \subset F_t^{i^*}$ such that $|Y_t \cup \tilde{F}_t^{i^*}| = \mu$ with Algorithm 2
  Create the next population $P_{t+1} := Y_t \cup \tilde{F}_{i^*}^t$

$i \in [m]$ and x (strictly) dominates *y, denoted by $x \succ y$, if one inequality is strict; if neither $x \succeq y$ nor $y \succeq x$ then x and y are* incomparable.

2) *A set $S \subseteq \{0,1\}^n$ is a* set of mutually incomparable solutions *with respect to f if all search points in S are incomparable.*
3) *Each solution not dominated by any other in $\{0,1\}^n$ is called* Pareto-optimal. *A mutually incomparable set of these solutions that covers all possible non-dominated fitness values is called a* Pareto(-optimal) set *of f.*

The NSGA-III algorithm (Deb and Jain 2014) is shown in Algorithm 1 (compare also with (Wietheger and Doerr 2023) or (Opris et al. 2024)). At first, a population of size μ is generated by initialising μ individuals uniformly at random. Then in each generation, a population Q_t of μ new offspring is created by conducting the following operations $\mu/2$ times. At first two parents p_1 and p_2 are chosen uniformly at random. Then 1-point crossover will be applied on (p_1, p_2) with some probability $p_c \in [0,1)$ to produce two solutions c_1, c_2. If 1-point crossover is not executed (with probability $1 - p_c$), c_1, c_2 are exact copies of p_1, p_2. Finally, two offspring s_1 and s_2 are created with *standard bit mutation* on c_1 and c_2, i.e. by flipping each bit independently with probability $1/n$.
During the survival selection, the parent and offspring populations P_t and Q_t are merged into R_t, and then partitioned into layers $F_{t+1}^1, F_{t+1}^2, \ldots$ using the *non-dominated sorting*

Algorithm 2: Selection procedure based on a set $\mathcal{R}_p$ of reference points for maximising a function

Compute the normalisation f^n of f
Associate each $x \in Y_t \cup F_t^{i^*}$ with its reference point $\text{rp}(x)$ such that the distance between $f^n(x)$ and the line through the origin and $\text{rp}(x)$ is minimised
For each $r \in \mathcal{R}_p$, set $\rho_r := |\{x \in Y_t \mid \text{rp}(x) = r\}|$
Initialise $\tilde{F}_t^{i^*} = \emptyset$ and $R' := \mathcal{R}_p$
while *true* **do**
  Determine $r_{\min} \in R'$ such that $\rho_{r_{\min}}$ is minimal (where ties are broken randomly) Determine $x_{r_{\min}} \in F_t^{i^*} \setminus \tilde{F}_t^{i^*}$ which is associated with $r_{\min}$ and minimises the distance between the vectors $f^n(x_{r_{\min}})$ and $r_{\min}$ (where ties are broken randomly)
  if $x_{r_{\min}}$ *exists* **then**
    $\tilde{F}_t^{i^*} = \tilde{F}_t^{i^*} \cup \{x_{r_{\min}}\}$
    $\rho_{r_{\min}} = \rho_{r_{\min}} + 1$
    if $|Y_t| + |\tilde{F}_t^{i^*}| = \mu$ **then return** $\tilde{F}_t^{i^*}$;
  else $R' = R' \setminus \{r_{\min}\}$;

algorithm (Deb et al. 2002). The layer F_{t+1}^1 consists of all non-dominated points, and F_{t+1}^i for $i > 1$ consists of points that are only dominated by those from $F_{t+1}^1, \ldots, F_{t+1}^{i-1}$. Then the critical and unique index i^* with $\sum_{i=1}^{i^*-1} |F_i^t| < \mu$ and $\sum_{i=1}^{i^*} |F_i^t| \geq \mu$ is determined (i.e. there are fewer than μ search points in R_t with a lower rank than i^*, but at least μ search points with rank at most i^*). All individuals with a smaller rank than i^* are taken into P_{t+1} and the remaining points are chosen from $F_{i^*}^t$ with Algorithm 2.

At first in Algorithm 2, a normalised objective function f^n is computed and then each individual with rank at most i^* is associated with reference points. We use the same set of structured reference points $\mathcal{R}_p$ as proposed in the original paper (Deb and Jain 2014), originated in (Das and Dennis 1998). The points are defined on the simplex of the unit vectors $(1,0,\ldots,0)^\intercal, (0,1,\ldots,0)^\intercal, \ldots, (0,0,\ldots,1)^\intercal$ as:

$$\left\{ \left(\frac{a_1}{p}, \ldots, \frac{a_m}{p} \right) \mid (a_1, \ldots, a_m) \in \mathbb{N}_0^m, \sum_{i=1}^{m} a_i = p \right\}$$

where $p \in \mathbb{N}$ is a parameter one can choose according to the fitness function f.

Now each individual x is associated with the reference point $\text{rp}(x)$ such that the distance between $f^n(x)$ and the line through the origin and $\text{rp}(x)$ is minimal. Then, one iterates through all the reference points where the reference point with the fewest associated individuals that are already selected for the next generation P_{t+1} is chosen. Ties are broken uniformly at random. A reference point is omitted if it only has associated individuals that are already selected for P_{t+1}. Then, among the not yet selected individuals of that reference point, the one nearest to the chosen reference point is taken for the next generation where ties are again broken uniformly at random. If the required number of individuals is reached (i.e. if $|Y_t| + |\tilde{F}_t^{i^*}| = \mu$) the selection ends. In Line 6 of Algorithm 2 one could use any other diversity-perserving mechanism if $\rho_{r_{\min}} \geq 1$. Note that NSGA-II follows the same scheme as Algorithm 1 with the difference that $\tilde{F}_t^{i^*} \subset F_t^{i^*}$ is chosen based on sorting according to the crowding distance, rather than using Algorithm 2 (Deb et al. 2002).

Further, we use the normalisation from (Blank, Deb, and Roy 2019). The detailed procedure is provided in that paper. However, for our purposes, the following description is sufficient. For an m-objective function $f\colon \{0,1\}^n \to \mathbb{N}_0^m$, the normalised fitness vector $f^n(x) := (f_1^n(x), \ldots, f_m^n(x))$ of a search point x is given by

$$f_j^n(x) = \frac{f_j(x) - y_j^{\min}}{y_j^{\text{nad}} - y_j^{\min}} \quad (1)$$

for each $j \in [m]$. The points $y^{\text{nad}} := (y_1^{\text{nad}}, \ldots, y_m^{\text{nad}})$ and $y^{\min} := (y_1^{\min}, \ldots, y_m^{\min})$ from the objective space are denoted by *nadir* and *ideal* points, respectively. In particular, $y_j^{\min}$ is set to the minimum value in objective j from all search points seen so far (i.e. from $R_0, \ldots, R_t$). Computing the nadir point is non-trivial, but the procedure described in (Blank, Deb, and Roy 2019) ensures for each $j \in [m]$ that $y_j^{\text{nad}} \geq \varepsilon_{\text{nad}}$, and $y_j^{\min} \leq y_j^{\text{nad}} \leq y_j^{\max}$ where $y_j^{\max}$ is the maximum value in objective j from all search points seen so far and ε_{nad} is a positive threshold. The following crucial result from (Opris et al. 2024) shows that sufficiently many reference points protect good solutions. In other words, if a population covers a fitness vector v with a first-ranked individual x, i.e. there is $x \in F_t^1$ with $f(x) = v$, then it is covered for all future generations as long as $x \in F_t^1$. (Compare also with (Wietheger and Doerr 2023) for a similar result, but limited to the 3-objective m-OMM problem for a higher number p of divisions.)

Lemma 2 (Opris et al. (2024), Lemma 3.4)**.** *Consider NSGA-III optimising an m-objective function f with $\varepsilon_{nad} \geq f_{\max}$ and a set $\mathcal{R}_p$ of reference points for $p \in \mathbb{N}$ with $p \geq 2m^{3/2} f_{\max}$. Let P_t be its current population and F_t^1 be the multiset describing the first layer of the merged population of parent and offspring. Assume the population size μ fulfills the condition $\mu \geq |S|$ where S is a maximum set of mutually incomparable solutions. Then for every $x \in F_t^1$ there is a $x' \in P_{t+1}$ with $f(x') = f(x)$.*

The Many-Objective Royal-Road Function

In this section, we define the many-objective REALROYALROAD function which we denote by m-RR_{MO}. Fix $m \in \mathbb{N}$ divisible by 2 and let n be divisible by $5m/2$. For a bit string x let $x := (x^1, \ldots, x^{m/2})$ where all x^j are of equal length $2n/m$. Let $B := \{y \in \{0,1\}^{2n/m} \mid |y|_1 = 6n/(5m), \text{LZ}(y) + \text{TZ}(y) = 4n/(5m)\}$ and $A := \{y \in \{0,1\}^{2n/m} \mid |y|_1 = 8n/(5m), \text{LZ}(y) + \text{TZ}(y) = 2n/(5m)\}$ refer to the substring x^j. The following sets refer to the whole bit string, and are needed to partition the search space accordingly.

- $L := \{x \mid 0 < |x^j|_1 \leq 6n/(5m) \text{ for all } j \in [m/2], |x^i|_1 < 6n/(5m) \text{ for an } i \in [m/2]\}$,
- $M := \{x \mid |x^j|_1 = 6n/(5m) \text{ for all } j \in [m/2] \text{ and } x^i \notin B \text{ for an } i \in [m/2]\}$,
- $N := \{x \mid x^j \in A \cup B \text{ for all } j \in [m/2]\}$.

Definition 3. *The function class m-$\text{RR}_{\text{MO}} : \{0,1\}^n \to \mathbb{N}_0^m$, is defined as*

$$m\text{-RR}_{\text{MO}}(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$

with

$$f_k(x) = g_k(x) := \begin{cases} |x^{1+(k-1)/2}|_1 & \text{if } k \text{ is odd,} \\ |x^{1+(k-2)/2}|_1 & \text{if } k \text{ is even,} \end{cases}$$

if $x \in L$,

$$f_k(x) = h_k(x) := g_k(x) + \begin{cases} \text{LZ}(x^{1+(k-1)/2}) & \text{if } k \text{ is odd,} \\ \text{TZ}(x^{1+(k-2)/2}) & \text{if } k \text{ is even,} \end{cases}$$

if $x \in M$,

$$f_k(x) = 4n|K(x)|/(5m) + h_k(x)$$

if $x \in N$ where $K(x) := \{j \in [m/2] \mid x^j \in A\}$, and $f_k(x) = 0$ otherwise.

In the m-objective REALROYALROAD function the bit string is divided into $m/2$ blocks of equal length $2n/m$. Algorithms initialising their population uniformly at random typically begin with search points x such that $0 < |x^j|_1 \leq 3/5(2n/m) = 6n/(5m)$ for each $j \in [m/2]$. Then we give a fitness signal to increase the number of ones to $3/5(2n/m) = 6n/(5m)$ in each x^j. After that, we aim to store all these ones in a cumulative block which is achieved by increasing the sum of leading and trailing zeros in each block j to obtain $x^j \in B$. Finally, if $x^j \in A \cup B$ for each block $j \in [m/2]$, there is a strong fitness signal *equally* to each objective according to $|K(x)|$, the number of blocks $j \in [m/2]$ in x such that $x^j \in A$. In the following we summarise important properties of m-RR_{MO}.

Lemma 4. *The following properties hold.*

(1) Let x, y with $x \in L$, $y \in M \cup N$. Then y dominates x.
(2) Let $\mathcal{Q} := \{x \in \{0,1\}^n \mid x^j \in B \text{ for all } j \in [m/2]\}$. Then for every $x \in M$ there is $y \in \mathcal{Q}$ dominating x.
(3) Let $x, y \in N$ be with $|K(x)| < |K(y)|$. Then y dominates x.
(4) The Pareto set $\mathcal{P}$ of RR_{MO} is

$$\mathcal{P} := \{x \in \{0,1\}^n \mid x^j \in A \text{ for all } j \in [m/2]\}.$$

Proof. (1): Note that $f_k(y) \geq 6n/(5m)$ for every $k \in [m]$ since each block j contains at least $6n/(5m)$ ones. On the other hand, $f_k(x) \leq 6n/(5m)$ and there is a block $i \in [m/2]$ with $|x^i|_1 < 6n/(5m)$, i.e. $f_{2i}(x) < 6n/(5m)$.
(2): Let $i \in [m/2]$ such that $\text{LZ}(x^i) + \text{TZ}(x^i)$ is not maximum (i.e. $\text{LZ}(x^i) + \text{TZ}(x^i) < 4n/(5m)$). Then there is a zero in x^i not contributing to $\text{LZ}(x^i) + \text{TZ}(x^i)$ (i.e. between the leftmost and rightmost one in x^i). Hence, exchanging that zero with the leftmost one creates a search point w with $w^j = x^j$ for $j \in [m/2] \setminus \{i\}$, $|w^i|_1 = |x^i|_1$, $\text{LZ}(w^i) = \text{LZ}(x^i) + 1$ and $\text{TZ}(w^i) = \text{TZ}(x^i)$. Hence, w dominates x. Repeating this operation in w^i until there is no

such zero left gives the desired search point y by the transitivity of dominance.
(3): Note that $h_k(x) \leq 2n/m$ for all $k \in [m]$ since in every block the sum of the number of ones and leading (trailing) zeros does not exceed $2n/m$. Since each block contains at least $6n/(5m)$ ones, $h_k(x) \geq 6n/(5m)$. Putting this together gives $f_k(x) = 4n|K(x)|/(5m) + h_k(x) \leq 4n|K(x)|/(5m) + 2n/m = 4n(|K(x)| + 1)/(5m) - 4n/(5m) + 2n/m \leq 4n|K(y)|/(5m) + 6n/(5m) \leq 4n|K(y)|/(5m) + h_k(y)$. Since either $h_1(x) < 2n/m$ or $h_2(x) < 2n/m$ (the leading and trailing zeros in block 1 are not $4n/(5m)$ at the same time since $|x^1|_1 = 6n/(5m)$), one of the inequalities above is strict for $k = 1$ or $k = 2$.
(4): Note that $|K(z)| = m/2$ for $z \in \mathcal{P}$ and $|K(w)| < m/2$ for every $w \notin \mathcal{P}$. Hence, by (2) every point $z \in \mathcal{P}$ dominates every point $w \notin \mathcal{P}$. Let $x, y \in \mathcal{P}$ be two search points with $x \neq y$. Then we find $j \in [m/2]$ with $x^j \neq y^j$. Since $|x^j|_1 = |y^j|_1 = 8n/(5m)$ which are stored in a cumulative block, we see either $\mathrm{LZ}(x^j) > \mathrm{LZ}(y^j)$ (i.e. $\mathrm{TZ}(x^j) < \mathrm{TZ}(y^j)$) or $\mathrm{LZ}(y^j) > \mathrm{LZ}(x^j)$ (i.e. $\mathrm{TZ}(y^j) < \mathrm{TZ}(x^j)$). In both cases, x and y are incomparable. □

We also bound the number of mutually incomparable solutions contained in any population as follows.

Lemma 5. *Let m be a constant and S be a set of mutually incomparable solutions of* $\mathrm{RR}_{\mathrm{MO}}$*. Then $|S| \leq c(4n/(5m)+1)^{m-1}$ for a constant $c \in \mathbb{N}$ with $c = 1$ if $m = 2$.*

Proof. Suppose that $|S| \geq 2$. Hence, by Lemma 4(1), we either have $S \subset L$ or $S \subset N \cup M$.
If $S \subset L$ then every $v \in f(S)$ fulfills $v_k \in [5n/(6m)]$, and $v_{k-1} = v_k$ if $k \in [m]$ is even. Note also that $(u_1, \ldots, u_{m-2}) \neq (v_1, \ldots, v_{m-2})$ for two different $u, v \in f(S)$. Otherwise, x, y with $f(x) = u$ and $f(y) = v$ are comparable (due to $f(S) \subset f(L)$ and $v_{m-1} = v_m$ for every $v \in f(L)$). Therefore, $|S| = |f(S)| \leq (6n/(5m) + 1)^{m/2-1} \leq (\sqrt{6n/(5m)+1})^{m-1} \leq (4n/(5m)+1)^{m-1}$ (due to $\sqrt{x+1} \leq 2x/3 + 1$ for every $x \geq 0$).
Assume $S \subset N \cup M$ and let $x \in S$. Then by Lemma 4(3) $|K(y)| = |K(x)|$ for every $y \in S$. If $|K(x)| = 0$ then $f(S) \subset V := 6n/(5m)\vec{1} + \{(v_1, \ldots, v_m) \mid v_i \in \{0\} \cup [4n/(5m)]$ for all $i \in [m]\}$ since $\mathrm{LZ}(x_j), \mathrm{TZ}(x_j) \leq 4n/(5m)$ for every $j \in [m/2]$. Since $(u_1, \ldots, u_{m-1}) \neq (v_1, \ldots, v_{m-1})$ for two different $u, v \in f(S)$, we see that $|S| = |f(S)| \leq (4n/(5m)+1)^{m-1}$ in a similar way as above. If $|K(x)| > 0$ we just estimate S by the number of search points y with $\ell := |K(y)| = |K(x)|$ and $y^j \in A \cup B$ which is $\binom{m/2}{\ell}(4n/(5m) + 1)^{m/2-\ell}(2n/(5m) + 1)^{\ell} \leq c_\ell(4n/(5m)+1)^{m/2}$ for $c_\ell := \binom{m/2}{\ell}$. Taking c as the maximum on $c_1, \ldots, c_{m/2}$ gives the result. □

We will see that a Pareto optimal search point can be explored easily using recombination, in particular 1-point crossover, by successively recombining individuals with $x^j \in B$ in order to create individuals y with $y^j \in A$. A minor modification to the bi-objective $\mathrm{RR}_{\mathrm{MO}}$ function from (Dang et al. 2023b) is that we count the leading ones and trailing zeros in each block only if the number of ones in every block reaches $6n/(5m)$, instead of scaling the number of ones by a factor of n. The reason is that, when scaling, the maximum possible fitness value in one objective becomes $\Theta(n^2)$, which implies, according to Lemma 2, that NSGA-III requires a significantly higher number of reference points to protect good solutions.

Crossover Guarantees Polynomial Time

Now we show that for NSGA-III can find the whole Pareto set of $\mathrm{RR}_{\mathrm{MO}}$ in expected polynomial time.

Theorem 6. *Let $m \in \mathbb{N}$ be any constant divisible by 2. Then the algorithm NSGA-III (Algorithm 1) with $p_c \in (0,1)$, $\varepsilon_{nad} \geq 2n/5+2n/m$, a set $\mathcal{R}_p$ of reference points as defined above for $p \in \mathbb{N}$ with $p \geq 2m^{3/2}(2n/5 + 2n/m)$, and a population size $\mu \geq c(4n/(5m) + 1)^{m-1}$ for a constant $c \in \mathbb{N}$ becoming 1 if $m = 2$, $\mu \in 2^{O(n)}$, finds the Pareto set of $f := m$-$\mathrm{RR}_{\mathrm{MO}}$ in expected $O(n^3/(1-p_c) + p_c)$ generations and $O(\mu n^3/(1-p_c) + \mu p_c)$ fitness evaluations.*

Proof. Note that $f_{\max} = 2n/5 + 2n/m$ by noticing that $|K_t(x)| \leq m/2$ and $|x^j|_1 + \mathrm{LZ}(x^j), |x^j|_1 + \mathrm{TZ}(x^j) \leq 2n/m$. So during the whole optimisation procedure we may apply Lemma 2. Further, we use the method of typical runs (Wegener 2002, Section 11) and divide a run into several phases. For every phase we compute the expected waiting time to reach one of the next phases. A phase can be skipped if the goal of a later phase is achieved.
Phase 1: Create x with $f(x) \neq 0$.
Let x be initialised uniformly at random. By a classical Chernoff bound the probability that $0 < |x^j|_1 \leq 6n/(5m)$ for every $j \in \{1, \ldots, m/2\}$ is $1 - e^{-\Omega(n)}$ since the expected value of ones in one block of a search point is $1/2 \cdot 2n/m = n/m$ after initialisation. Hence, the probability that every individual has fitness zero after initialisation is $e^{-\Omega(\mu n)}$. If this event occurs, the probability is at least n^{-n} to create any individual with mutation (no matter if crossover is executed) and hence, one with fitness distinct from 0. So the expected number of generations to finish this phase is at most $(1 - e^{-\Omega(\mu n)}) + e^{-\Omega(\mu n)} n^n = 1 + o(1)$.
Phase 2: Create x with $|x^j|_1 = 6n/(5m)$ for all $j \in [m/2]$.
Let $O_t := \max\{|x|_1 \mid x \in P_t, f(x) \neq 0\}$. Since $0 < |x|_1^j \leq 6n/(5m)$ for every $j \in [m/2]$ if $f(x) \neq 0$, we have $m/2 \leq O_t < m/2 \cdot 6n/(5m) = 3n/5$ and O_t cannot decrease by Lemma 2 since a solution x with $|x|_1 = O_t$ is non-dominated. Note that $|x^j|_1 < 6n/(5m)$ for a $j \in [m/2]$. To increase O_t in one trial it suffices to choose a parent $z \in P_t$ with $|z|_1 = O_t$ (prob. at least $1 - (1 - 1/\mu)^2 \geq 1/\mu$), omit crossover (prob. $(1 - p_c)$) and flip one of $2n/m - |x^j|_1 \geq 4n/(5m)$ zero bits to one (prob. $4(1 - 1/n)^{n-1}/(5m) \geq 4/(5me)$). Hence, in one generation, the probability to increase O_t is at least $1 - (1 - r_t)^{\mu/2} \geq \frac{\mu r_t}{2}/(1 + \frac{\mu r_t}{2})$ for $r_t := 4(1 - p_c)/(5\mu m e)$ (for this inequality see, for example, Lemma 10 in (Badkobeh, Lehre, and Sudholt 2015)) since in each generation $\mu/2$ many pairs of two individuals are generated independently of each other. Hence, the expected number of generations to complete this phase is at most $(3n/5 - m/2)(1 + 2/(\mu r_t)) = (3n/5 - m/2)(1 + 10me/(4(1 - p_c))) = O(n/(1 - p_c))$.

Phase 3: Create x with $x^j \in B$, i.e. $\mathrm{LZ}(x^j) + \mathrm{TZ}(x^j) = 4n/(5m)$, for every $j \in [m/2]$.
Let $W_t := \{x \in P_t \mid |x^j|_1 = 6n/(5m) \text{ for all } j \in [m/2]\}$ and for $k \in [m]$, $x \in \{0,1\}^n$ let $T_k(x) = \mathrm{LZ}(x^{1+(k-1)/2})$ if k is odd and $T_k(x) = \mathrm{TZ}(x^{1+(k-2)/2})$ otherwise. Note that $f_k(x) = 6n/(5m) + T_k(x)$ for $x \in W_t$. Set $\alpha_t = \max\{\sum_{i=1}^{m/2}(T_{2i-1}(x) + T_{2i}(x)) \mid x \in W_t\}$. Since $0 \le \mathrm{LZ}(x^j) + \mathrm{TZ}(x^j) \le 4n/(5m)$ for every $j \in [m/2]$, we obtain $\alpha_t \in \{0, \dots, 2n/5 - 1\}$. Note that this phase is finished if α_t becomes $2n/5$. According to Lemma 2, α_t cannot decrease since a corresponding solution w with value α_t is non-dominated. In w the total number of zeros not contributing to any $\mathrm{LZ}(w^j) + \mathrm{TZ}(w^j)$ for $j \in [m/2]$ is $2n/5 - \alpha_t := \sigma_t$. To increase α_t in a trial is suffices to choose such a solution w from P_t, omit crossover and execute mutation as follows: Flip one of the σ_t zeros to one and the leftmost one bit in the same block i to zero to increase $\mathrm{LZ}(w^i) + \mathrm{TZ}(w^i)$ (prob. $\sigma_t/(n^2) \cdot (1-1/n)^{n-2} \ge \sigma_t/(en^2)$) while $\mathrm{LZ}(w^j) + \mathrm{TZ}(w^j)$ remains unchanged for every $j \in [m/2] \setminus \{i\}$. Let $r_t := (1-p_c)\sigma_t/(en^2\mu)$. Then the probability is at least $1 - (1-r_t)^{\mu/2} \ge \frac{r_t\mu}{2}/(1 + \frac{r_t\mu}{2})$ to increase α_t in one generation. Since $\sigma_t \in [2n/5]$, the expected number of generations to obtain $\alpha_t = 2n/5$ is at most $\sum_{j=1}^{2n/5}(1 + 2en^2/((1-p_c)j) \le 2n/5 + 2en^2(\ln(2n/5) + 1)/(1-p_c)) = O(n^2 \ln(n)/(1-p_c))$.
For defining the next phases let $\ell_t := \max\{|K(x)| \mid x \in P_t \text{ with } x^j \in A \cup B \text{ for every } j \in [m/2]\}$. Suppose that $\ell_t \in [0] \cup [m/2-1]$, i.e. there is a individual $z \in P_t$ with $z^j \in A \cup B$ and $|K(z)| = \ell_t$, but no corresponding w with $|K(w)| > \ell_t$. By Lemma 4(3), ℓ_t cannot decrease.
Phase ℓ_t+4: Create an individual x with $|K(x)| = \ell_t + 1$.
Note that Phase 4 starts when $\ell_t = 0$. If Phase $m/2 - 1 + 4 = m/2 + 3$ is finished a Pareto optimal search point is found since $\ell_t + 1 = m/2$. We consider several subphases.
Subphase A: Let $I := K(z)$. Then cover $S_I := \{x \mid x^j \in B \text{ for all } j \in [m/2] \setminus I \text{ and } x^j \in A \text{ for all } j \in I\}$.
For a specific search point $w \in S_I$ not already covered we first upper bound the probability by $e^{-\Omega(n)}$ that a solution x with $x = w$ has not been created after $8en^3/(1-p_c)$ generations. Let $D_t := \{x \in P_t \mid x^j \in B \text{ for every } j \in [m/2] \setminus I \text{ and } x^j \in A \text{ else}\}$. We consider $d_t := \min_{x \in D_t} \sum_{i \in [m/2]} H(x^i, w^i)/2$ where $H(x^i, w^i)$ denotes the Hamming distance between x^i and w^i. For $x \in D_t$ we have that $H(x^i, w^i)$ is even, $H(x^i, w^i) \le 8n/(5m)$ if $i \notin I$, and $H(x^i, w^i) \le 4n/(5m)$ if $i \in I$ (since $|x^i|_1 = |y^i|_1 = 6n/(5m)$ if $i \notin I$, $|x^i|_1 = |y^i|_1 = 8n/(5m)$ if $i \in I$ and every x^i has length $2n/m$). This implies $0 < d_t \le 2n|I|/(5m) + 4n(m/2 - |I|)/(5m) = 2n/5 - 2n|I|/(5m) := s(n)$. Since a solution $x \in D_t$ is non-dominated (compare with the proof of Lemma 4(2)), d_t cannot increase (by Lemma 2). Note that we created w if $d_t = 0$. For $1 \le \beta \le s(n)$, define the random variable X_β as the number of generations t with $d_t = \beta$. Then the total number of generations required to find a solution x with $x = w$ is at most $X = \sum_{\beta=1}^{s(n)} X_\beta$. Fix $y \in D_t \cup P_t$ with $\sum_{i \in [m/2]} H(w^i, y^i)/2 = d_t \neq 0$. To decrease d_t, it suffices to choose y as a parent, omit crossover and flip two specific bits during mutation in order to shift a block of ones in y in that direction of the corresponding block of w (prob. $1/n^2 \cdot (1-1/n)^{n-2} \ge 1/(en^2)$). Hence, for $a_t := (1-p_c)/(\mu en^2)$, the probability to decrease d_t in one generation is at least $1 - (1-a_t)^{\mu/2} \ge \frac{a_t\mu/2}{1 + a_t\mu/2} \ge a_t\mu/4 = (1-p_c)/(4en^2)$. Thus, for every $\beta \in \{1, \dots, s(n)\}$, the random variable X_β is stochastically dominated by a geometrically distributed random variable Z_β with success probability $q := q_\beta := (1-p_c)/(4en^2)$. Note that the Z_β can be also considered as independent. Let $Z := \sum_{\beta=1}^{s(n)} Z_\beta$. Then $E[Z] = 4en^2 s(n)/(1-p_c)$. Now we use Theorem 15 in (Doerr 2019): For $d := \sum_{\beta=1}^{s(n)} 1/q_\beta^2 = 16e^2n^4 s(n)/(1-p_c)^2$ and $\lambda \ge 0$ we obtain

$$P(Z \ge E[Z] + \lambda) \le \exp\left(-\frac{1}{4}\min\left\{\frac{\lambda^2}{d}, \lambda q\right\}\right).$$

For $\lambda = 4en^3/(1-p_c)$ we obtain $P(X \ge 8en^3/(1-p_c)) \le P(Z \ge 8en^3/(1-p_c)) \le e^{-\Omega(n)}$.
By a union bound over all possible w, the probability is at most $|S_I| \cdot e^{-\Omega(n)} = ((4n/(5m)+1)^{m/2-|I|} + (2n/(5m)+1)^{|I|}) \cdot e^{-\Omega(n)} = e^{-\Omega(n)}$ that S_I is completely covered by individuals after $8en^3/(1-p_c)$ generations. If this does not happen, we can repeat the argument. Thus the expected number of generations to finish this phase is at most $(1+o(1))(8en^3/(1-p_c)) = O(n^3/(1-p_c))$.
Subphase B: Every $x \in P_t$ fulfills $|K(x)| = \ell_t$ and $x^j \in A \cup B$ for every $j \in [m/2]$.
Let N_t be the number of such individuals. By Lemma 4(2) the non-dominated individuals x are precisely those and hence, N_t cannot decrease. Note also that $N_t = \Omega(n)$ (since we got through Subphase A). Denote by X_t the number of new created individuals of this kind in $1/(1-p_c)$ generations. Then $E[X_t] \ge N_t/4$ since in one trial such an individual is cloned with probability at least $N_t(1-p_c)/(4\mu)$ (with prob. at least N_t/μ one such individual is selected as parent, with prob. $(1-p_c)$ crossover is omitted and no bit is flipped with prob. $(1-1/n)^n \ge 1/4$ during mutation) and by a classical Chernoff bound $P(X_t \le 0.5E[X_t]) \le e^{-\Omega(N_t)} = e^{-\Omega(n)}$. Hence, with probability $1 - e^{-\Omega(n)}$ we have that $N_{t+1} \ge \min\{N_t + N_t/8, \mu\} = \min\{9N_t/8, \mu\}$ and by a union bound, we obtain with probability at least $1 - e^{-\Omega(n)}$ that Subphase B is finished in at most $\lceil \log_{9/8}(\mu/N_t) \rceil/(1 - p_c) = O(\ln(\mu)/(1-p_c)) = O(n/(1-p_c))$ many generations since $\mu \in O(2^n)$. If this does not happen, we repeat the argument and obtain an expected number of $(1+o(1))O((1-p_c)n) = O((1-p_c)n)$ generations.
Subphase C: Create an individual y with $|K(y)| \ge \ell_t + 1$.
To create such an individual y in one generation one has to choose two individuals y_1, y_2 with $K(y_1) = K(y_2)$ as parents such that $y_1^i = 0^a 1^{6n/(5m)} 0^{4n/(5m)-a}$ and $y_2^i = 0^{2n/(5m)+a} 1^{6n/(5m)} 0^{2n/(5m)-a}$ for an $a \in \{0, \dots, 2n/(5m)\}$, performing one-point crossover with cutting point $k \in \{(i-1)(2n)/m + 2n/(5m) + a, \dots, (i-1)(2n)/m + 6n/(5m) + a\}$ (i.e. in block i at position $b \in \{2n/(5m) + a, \dots, 6n/(5m) + a\}$), and then omit-

ting mutation. Note that $K(y_1) \cup \{i\} \subset K(y)$ (since either $y^j = y_1^j$ or $y^j = y_2^j$ for every $j \in K(y_1)$) and hence, $|K(y)| \geq \ell_t + 1$. We estimate the probability that this sequence of events occurs for *good* generations, defined as follows. A generation is called *good* if for every $z \in P_t$ the corresponding $S_{K(z)}$ (defined in Subphase A) is completely covered by P_t. Generations which are not good are called *bad*. Since every $x \in P_t$ satisfies $|K(x)| = \ell_t$, and there are $\binom{m/2}{\ell_t}$ different possibilities for $K(x)$ (since $K(x) \subset [m/2]$ and $|K(x)| = \ell_t$ there are $\binom{m/2}{\ell_t}$ subsets of $[m/2]$), there are at least $\mu/\binom{m/2}{\ell_t}$ individuals x with the same $K(x)$-value in good generations by the pigeonhole principle. Since the generation is good, an arbitrary individual z with $K(z) = K(x)$ can be either used as first or second parent in the recombination step (since either $z^i = 0^a 1^{6n/(5m)} 0^{4n/(5m)-a}$ or $z^i = 0^{2n/(5m)+a} 1^{6n/(5m)} 0^{2n/(5m)-a}$ for an $a \in \{0, \dots, 2n/(5m)\}$). Hence, in a good generation, suitable parents for one-point crossover are chosen with probability at least $r := (\mu/\binom{m/2}{\ell_t})/\mu^2 = 1/(\binom{m/2}{\ell_t}\mu) \in \Omega(1/\mu)$, and a correct cutting point without mutation afterwards is found with probability at least $s := (2n/(5m)+1)/(4(n+1)) \in \Theta(1)$. Hence, the probability to create such an y in a good generation is at least $1 - (1 - p_c rs)^{\mu/2} \geq \frac{p_c rs\mu/2}{1+p_c rs\mu/2} \geq p_c rs\mu/4 = \Omega(p_c)$. The expected number of bad generations can be estimated by $O(n^3/(1-p_c))$ since there are $\binom{m/2}{\ell_t}$ possible $K(x)$, and one has to run at most $\binom{m/2}{\ell_t}$ times through Subphase A again to cover all occuring S_I for $I := K(x)$. In total, this phase is finished in $O(n^3/(1-p_c) + p_c)$ generations in expectation.

Phase m/2+4: Cover the whole Pareto front.

The treatment of this phase is similar to Subphase A of Phase ℓ_t+4. Let $V := \{x \in P_t \mid x^i \in A \text{ for every } i \in [m/2]\}$. We have $V \neq \emptyset$ and every $x \in V$ is Pareto-optimal (by Lemma 5(4)). Fix a search point $w \in V$ with $w \notin P_t$ and let $d_t := \min_{x \in P_t} H(x, w)/2$. Note that $0 < d_t \leq (2n/5m) \cdot m/2 = n/5$ and d_t cannot increase. Define for $j \in [n/5]$ the random variable X_j as the number of generations t with $d_t = j$. As in Subphase A, with probability at least $(1-p_c)/(4en^2)$, the value d_t decreases in one generation. Hence, X_j is stochastically dominated by a geometrically distributed random variable Z_j with success probability $q := q_j := (1-p_c)/(4en^2)$, and the number of generations until $d_t = 0$ is at most $\sum_{j=1}^{n/5} Z_j$. Note that the Z_j can be seen as independent. Hence, we can apply the remaining arguments from Subphase A adapted to this situation to cover the whole Pareto front in expected $O(n^3/(1-p_c))$ generations.

In total, the complete Pareto front is covered in expected $O(n^3/(1-p_c)+p_c)$ generations and $O(\mu n^3/(1-p_c)+\mu p_c)$ fitness evaluations since Subphases A,B and C are passed at most $m/2$ times. Since m is a constant, the running time for passing through Phase 4 to $m/2+3$ is asymptotically the same as passing through Subphases A,B and C once. $\square$

In Theorem 6, the expected number of generations to optimise m-RR$_{\text{MO}}$ does not asymptotically depend on m and even not on the population size μ. Dang et al. (2023b) showed for their bi-objective version of m-RR$_{\text{MO}}$ a runtime bound of $O(n^3/(1-p_c) + \mu/(p_c n))$ for NSGA-II which is by a factor of μ/n^4 worse for population sizes $\mu \in \Omega(n^4)$ and constant $p_c \in (0,1)$.

Difficulty of NSGA-III Without Crossover

Finally, we point out that NSGA-III without crossover (i.e. when $p_c = 0$) becomes extremely slow. This even holds for finding the first Pareto optimal point.

Theorem 7. *Suppose that m is a constant divisible by 2. NSGA-III (Algorithm 1) on m-RR$_{\text{MO}}$ with $p_c = 0$, any choice of $\mathcal{R}_p$, and μ polynomial in n needs at least $n^{\Omega(n)}$ generations in expectation to create any Pareto-optimal search point of m-RR$_{\text{MO}}$.*

Proof. We see with probability of $2^{-\Omega(n)}$ that an individual x with $0 < |x^i|_1 \leq 6n/(5m)$ for every $i \in [m/2]$ initialises with probability $1 - 2^{-\Omega(n)}$. Hence, by a union bound, with probability $1 - \mu 2^{-\Omega(n)} = 1 - o(1)$ every individual x initialises with $0 < |x^i|_1 \leq 6n/(5m)$ for every $i \in [m/2]$ since $\mu \in \text{poly}(n)$. Suppose that this happens. Then the algorithm will always reject search points with fitness zero. Therefore, it is required to flip $2n/(5m)$ many zeros at once even to create a search point y with $y^i \in A$ for any $i \in [m/2]$. This happens with probability $n^{-\Omega(n)}$. So the expected number of needed generations in total is at least $(1-o(1))(n^{\Omega(n)}/\mu) = n^{\Omega(n)}$. $\square$

As in (Dang et al. 2023b),a similar result can be also formulated for a general class of $(\mu + \lambda)$ elisist black-box algorithms, i.e. unary blackbox algorithms, which use so-called *unary unbiased variation operators* (Doerr and Lengler 2017) which generalise standard bit mutation. For reasons of space, we will not present the results.

Conclusions

We defined m-RR$_{\text{MO}}$, a variant of the bi-objective RR$_{\text{MO}}$-function proposed by (Dang et al. 2023b), for the many objective setting on which the EMO algorithm NSGA-III using crossover for a constant $0 < p_c < 1$ and a constant number of objectives m can find the whole Pareto set in expected $O(n^3)$ generations and $O(\mu n^3)$ fitness evaluations. As for other many-objective function classes like LOTZ, OMM and COCZ, the upper bound on the expected number of generations behaves asymptotically independently of μ and m. On the other hand, if crossover is disabled, NSGA-III requires exponential time to even find a single Pareto-optimal point. This is the first proof for an exponential performance disparity for the use of crossover in the many-objective setting, particularly for NSGA-III. However, we are confident that for the m-ONEJUMPZEROJUMP$_k$ benchmark proposed by Zheng and Doerr (2024b), the many-objective version of the bi-objective ONEJUMPZEROJUMP (Doerr and Qu 2022), crossover provably guarantees a subexponential speedup of order $O(n)$. We hope that our work may serve as a stepping stone towards a better understanding of the advantages of crossover on more complex problem classes, as it has been done in single-objective optimisation.

Acknowledgements

This work benefited from fruitful discussions at Dagstuhl Seminar 24271 ”Theory of Randomized Optimization Heuristics”.

References

Badkobeh, G.; Lehre, P. K.; and Sudholt, D. 2015. Black-box Complexity of Parallel Search with Distributed Populations. In *Proceedings of the Foundations of Genetic Algorithms, FOGA 2015*, 3–15. ACM Press.

Bhesdadiya, R. H.; Trivedi, I. N.; Jangir, P.; Jangir, N.; and Kumar, A. 2016. An NSGA-III Algorithm for Solving Multi-Objective Economic/Environmental Dispatch Problem. *Cogent Engineering*, 3(1): 1269383.

Bian, C.; and Qian, C. 2022. Better Running Time of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) by Using Stochastic Tournament Selection. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN ’22)*, 428–441. Springer.

Blank, J.; Deb, K.; and Roy, P. C. 2019. Investigating the Normalization Procedure of NSGA-III. In Deb, K.; Goodman, E.; Coello Coello, C. A.; Klamroth, K.; Miettinen, K.; Mostaghim, S.; and Reed, P., eds., *Evolutionary Multi-Criterion Optimization*, 229–240. Cham: Springer International Publishing.

Chaudhari, P.; Thakur, A. K.; Kumar, R.; Banerjee, N.; and Kumar, A. 2022. Comparison of NSGA-III with NSGA-II for Multi Objective Optimization of Adiabatic Styrene Reactor. *Materials Today: Proceedings*, 57: 1509–1514.

Chikumbo, O.; Goodman, E. D.; and Deb, K. 2012. Approximating a Multi-Dimensional Pareto Front for a Land Use Management Problem: A Modified MOEA with an Epigenetic Silencing Metaphor. *2012 IEEE Congress on Evolutionary Computation*, 1–9.

Coello, C.; Veldhuizen, D. V.; and Lamont, G. 2013. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic Algorithms and Evolutionary Computation. Springer US.

Coello, C. A. C.; and Lamont, G. B. 2004. *Applications of Multi-Objective Evolutionary Algorithms*. WORLD SCIENTIFIC.

Corus, D.; and Oliveto, P. S. 2018. Standard Steady State Genetic Algorithms Can Hillclimb Faster Than Mutation-Only Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 22(5): 720–732.

Covantes Osuna, E.; Gao, W.; Neumann, F.; and Sudholt, D. 2020. Design and Analysis of Diversity-Based Parent Selection Schemes for Speeding Up Evolutionary Multi-objective Optimisation. *Theoretical Computer Science*, 832: 123–142.

Dang, D.-C.; Friedrich, T.; Kötzing, T.; Krejca, M. S.; Lehre, P. K.; Oliveto, P. S.; Sudholt, D.; and Sutton, A. M. 2017. Escaping Local Optima Using Crossover with Emergent Diversity. *IEEE Transactions on Evolutionary Computation*, 22: 484–497.

Dang, D.-C.; Opris, A.; Salehi, B.; and Sudholt, D. 2023a. Analysing the Robustness of NSGA-II under Noise. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’23)*, 642–651. ACM Press.

Dang, D.-C.; Opris, A.; Salehi, B.; and Sudholt, D. 2023b. A Proof That Using Crossover Can Guarantee Exponential Speed-Ups in Evolutionary Multi-Objective Optimisation. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2023*, 12390–12398. AAAI Press.

Dang, D.-C.; Opris, A.; and Sudholt, D. 2024. Crossover can Guarantee Exponential Speed-ups in Evolutionary Multi-Objective Optimisation. *Artificial Intelligence*, 330: 104098.

Das, I.; and Dennis, J. E. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, 8(3): 631–657.

Deb, K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

Deb, K.; and Jain, H. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4): 577–601.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

Doerr, B. 2019. Analyzing Randomized Search Heuristics Via Stochastic Domination. *Theoretical Computer Science*, 773: 115–137.

Doerr, B.; and Doerr, C. 2018. Optimal Static and Self-Adjusting Parameter Choices for the $(1+(\lambda,\lambda))$ Genetic Algorithm. *Algorithmica*, 80(5): 1658–1709.

Doerr, B.; Doerr, C.; and Ebel, F. 2015. From Black-Box Complexity to Designing New Genetic Algorithms. *Theoretical Computer Science*, 567: 87–104.

Doerr, B.; Hadri, O. E.; and Pinard, A. 2022. The $(1 + (\lambda, \lambda))$ Global SEMO Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’22)*, 520–528. ACM Press.

Doerr, B.; Happ, E.; and Klein, C. 2012. Crossover can Provably be Useful in Evolutionary Computation. *Theoretical Computer Science*, 425: 17–33.

Doerr, B.; Johannsen, D.; Kötzing, T.; Neumann, F.; and Theile, M. 2013. More Effective Crossover Operators for the All-Pairs Shortest Path Problem. *Theoretical Computer Science*, 471: 12–26.

Doerr, B.; and Qu, Z. 2022. 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 ’22)*, volume 13399, 399–412. Springer.

Doerr, B.; and Qu, Z. 2023. Runtime Analysis for the NSGA-II: Provable Speed-ups from Crossover. In *Conference on Artificial Intelligence, AAAI 2023*, 12399–12407. AAAI Press.

Doerr, C.; and Lengler, J. 2017. Introducing Elitist Black-Box Models: When Does Elitist Behavior Weaken the Performance of Evolutionary Algorithms? *Evolutionary Computation*, 25(4): 587–606.

Gu, Q.; Xu, Q.; and Li, X. 2022. An Improved NSGA-III Algorithm Based on Distance Dominance Relation for Many-Objective Optimization. *Expert Systems with Applications*, 207: 117738.

Ishibuchi, H.; and Murata, T. 1998. 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.

Jansen, T.; and Wegener, I. 2002. On the Analysis of Evolutionary Algorithms—A Proof That Crossover Really Can Help. *Algorithmica*, 34(1): 47–66.

Jansen, T.; and Wegener, I. 2005. Real Royal Road Functions—Where Crossover Provably is Essential. *Discrete Applied Mathematics*, 149: 111–125.

Köppen, M.; and Yoshida, K. 2007. Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems. In *Evolutionary Multi-Criterion Optimization*, 727–741. Springer Berlin Heidelberg.

Kötzing, T.; Sudholt, D.; and Theile, M. 2011. How Crossover Helps in Pseudo-Boolean Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'11)*, 989–996. ACM Press.

Laumanns, M.; Thiele, L.; and Zitzler, E. 2004. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions. *IEEE Transactions on Evolutionary Computation*, 8(2): 170–182.

Li, M.; Han, X.; Chu, X.; and Liang, Z. 2024. Empirical Comparison between MOEAs and Local Search on Multi-Objective Combinatorial Optimisation Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'24)*, 547–556. ACM Press.

Luukkonen, S.; van den Maagdenberg, H. W.; Emmerich, M. T.; and van Westen, G. J. 2023. Artificial Intelligence in Multi-Objective Drug Design. *Current Opinion in Structural Biology*, 79: 102537.

Opris, A.; Dang, D.-C.; Neumann, F.; and Sudholt, D. 2024. Runtime Analyses of NSGA-III on Many-Objective Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'24)*, 1596–1604. ACM Press.

Opris, A.; Lengler, J.; and Sudholt, D. 2024. A Tight $O(4k/p_c)$ Runtime Bound for a $(\mu + 1)$ GA on $\textsc{Jump}_k$ for Realistic Crossover Probabilities. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'24)*, 1605–1613. ACM Press.

Qian, C.; Bian, C.; and Feng, C. 2020. Subset Selection by Pareto Optimization with Recombination. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2020*, 2408–2415. AAAI Press.

Qian, C.; Yu, Y.; and Zhou, Z. 2013. An Analysis on Recombination in Multi-Objective Evolutionary Optimization. *Artificial Intelligence*, 204: 99–119.

Qu, Q.; Ma, Z.; Clausen, A.; and Jørgensen, B. N. 2021. A Comprehensive Review of Machine Learning in Multi-objective Optimization. In *2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI)*, 7–14.

Sekine, K.; and Tatsukawa, T. 2018. A Parametric Study of Crossover Operators in Multi-objective Evolutionary Algorithm. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1196–1203.

Sharma, S.; and Chahar, V. 2022. A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future. *Archives of Computational Methods in Engineering*, 29: 5605–5633.

Sudholt, D. 2017. How Crossover Speeds Up Building-Block Assembly in Genetic Algorithms. *Evolutionary Computation*, 25(2): 237–274.

Sutton, A.; and Lee, J. 2024. Evolving Populations of Solved Subgraphs with Crossover and Constraint Repair. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN '24)*, 133–148. Springer.

Sutton, A. M. 2021. Fixed-Parameter Tractability of Crossover: Steady-State GAs on the Closest String Problem. *Algorithmica*, 83(4): 1138–1163.

Tang, H.; Xiao, Y.; Zhang, W.; Lei, D.; Wang, J.; and Xu, T. 2024. A DQL-NSGA-III algorithm for solving the flexible job shop dynamic scheduling problem. *Expert Systems with Applications*, 237: 121723.

Wegener, I. 2002. Methods for the Analysis of Evolutionary Algorithms on Pseudo-Boolean Functions. In *Evolutionary Optimization*, 349–369. Kluwer.

Wietheger, S.; and Doerr, B. 2023. 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*, 5657–5665. ijcai.org.

Xingtao, L.; Qing, L.; Xujing, Y.; and Weigang, Z. 2008. Multiobjective Optimization for Crash Safety Design of Vehicles Using Stepwise Regression Model. *Structural and Multidisciplinary Optimization*, 35: 561–569.

Yi, J.-H.; Xing, L.-N.; Wang, G.-G.; Dong, J.; Vasilakos, A. V.; Alavi, A. H.; and Wang, L. 2020. Behavior of Crossover Operators in NSGA-III for Large-Scale Optimization Problems. *Information Sciences*, 509: 470–487.

Zheng, W.; and Doerr, B. 2024a. Runtime Analysis for the NSGA-II: Proving, Quantifying, and Explaining the Inefficiency for Many Objectives. *IEEE Transactions on Evolutionary Computation*, 28(5): 1442–1454.

Zheng, W.; and Doerr, B. 2024b. Runtime Analysis of the SMS-EMOA for Many-Objective Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18): 20874–20882.

Zheng, W.; Liu, Y.; and Doerr, B. 2022. 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*, 10408–10416. AAAI Press.