Speeding Up Evolutionary Multi-objective Optimisation Through Diversity-Based Parent Selection

Edgar Covantes Osuna
Department of Computer Science
University of Sheffield, Sheffield, United Kingdom

Wanru Gao
Optimisation and Logistics, School of Computer Science,
The University of Adelaide, Adelaide, Australia

Frank Neumann
Optimisation and Logistics, School of Computer Science,
The University of Adelaide, Adelaide, Australia

Dirk Sudholt
Department of Computer Science
University of Sheffield, Sheffield, United Kingdom

ABSTRACT

Parent selection in evolutionary algorithms for multi-objective optimization is usually performed by dominance mechanisms or indicator functions that prefer non-dominated points, while the reproduction phase involves the application of diversity mechanisms or other methods to achieve a good spread of the population along the Pareto front. We propose to refine the parent selection on evolutionary multi-objective optimization with diversity-based metrics. The aim is to focus on individuals with a high diversity contribution located in poorly explored areas of the search space, so the chances of creating new non-dominated individuals are better than in highly populated areas. We show by means of rigorous runtime analysis that the use of diversity-based parent selection mechanisms in the Simple Evolutionary Multi-objective Optimiser (SEMO) and Global SEMO for the well known bi-objective functions OneMinMax and Lotz can significantly improve their performance. Our theoretical results are accompanied by additional experiments that show a correspondence between theory and empirical results.

KEYWORDS

Parent selection, evolutionary algorithms, multi-objective optimization, diversity mechanisms, runtime analysis, theory

1 INTRODUCTION

The area of evolutionary multi-objective optimization (EMO) designs population-based evolutionary algorithms (EAs) where the population is used to approximate the so-called Pareto front. Given that EAs use a population which is a set of solutions for a given problem, EAs are suited in a natural way for computing trade-offs with respect to two (or more) conflicting objective functions.

Well established multi-objective evolutionary algorithms (MOEAs) such as NSGA-II [6], SPEA2 [2], IBEA [16] have two basic principles. First of all, the goal is to push the current population close to the "true" Pareto front. The second goal is to "spread" the population along the front such that it is well covered. The first goal is usually achieved by dominance mechanisms between the search points or indicator functions that prefer non-dominated points. The second goal involves the use of diversity mechanisms. Alternatively, indicators such as the hypervolume indicator play a crucial role to obtain a good spread of the different solutions of the population along the Pareto front.

In the context of EMO, parent selection is usually uniform whereas offspring selection is based on dominance and the contribution of an individual to the diversity of the population. In this paper, we explore the use of different parent selection mechanisms in EMO. The goal is to speed up the optimization process of an EMO algorithm by selecting individuals that have a high chance of producing beneficial offspring. To our knowledge this is a novel approach; the only previous work we are aware of is [14] where a MOEA with parent selection using a so-called prospect indicator is used to improve SMS-EMOA. The prospect indicator evaluates the potential (or prospect) of an individual to reproduce offspring that dominate itself. Their experimental results show improvement over classical MOEAs.

The parent selection mechanisms studied in this paper use the diversity contribution of an individual in the parent population to select promising individuals for reproduction. These mechanisms include ignoring individuals with a minimum diversity score, rank of individuals in the parent population where the rank is given based on the dominance relation and its contribution to diversity, and the classical tournament selection (TS) where the outcome is defined according to the diversity score and not in the fitness values.

The main assumption is that individuals with a high diversity score are located in poorly explored or a less dense areas of the search space, so the chances of creating new non-dominated individuals are better than in areas where there are several individuals. In this sense we have designed a MOEA that focused on individuals where the neighbourhood is not fully covered and in consequence, force the reproduction in those areas and to the spread of the population along the search space.

We show by means of rigorous runtime analysis that the use of diversity-based parent selection mechanisms can significantly improve the performance of MOEAs. The area of runtime analysis has

contributed significantly to the theoretical understanding of EMO algorithms [8, 10, 11] and allows to study different components of EMO methods from a rigorous perspective.

In order to gain insights into the potential benefits of the diversity-based parent selection mechanisms, we study the classical functions OneMinMax and Lotz (Leading Ones, Trailing Zeroes) problems introduced in [10] and [12], respectively. OneMinMax generalizes the classical OneMax function and Lotz generalizes the well-known LeadingOnes problem to the multi-objective case. Both functions have been examined in a wide range of theoretical studies for variants of the SEMO algorithm. Other studies in the area of runtime analysis of MOEAs consider hypervolume-based algorithms [7, 13], namely a variant of IBEA, and MOEAs incorporating other diversity mechanisms for survival selection [11].

We show that the use of various diversity-based parent selection mechanisms speeds up SEMO by a factor of order n for OneMinMax and Lotz. For Lotz the use of rank-based parent selection can reduce the runtime to compute the whole Pareto front from $\Theta(n^3)$ to $O(n^2)$. Studying OneMinMax, we show a similar effect, i. e. that the runtime reduces from $\Theta(n^2 \log n)$ to $O(n \log n)$ for our best performing rank-based parent selection methods.

The outline of the paper is as follows. In Section 2, we introduce the algorithms and problems that are subject to our investigations. Section 3 establishes the algorithmic framework used in the theoretical and experimental analysis. Section 4 establishes some general properties that enable speed-ups through diversity-based parent selection. Our rigorous runtime results for OneMinMax and Lotz are presented in Section 5 and 6, respectively. An experimental study complementing the theoretical results is presented in Section 7. Finally, we finish with some concluding remarks.

2 PRELIMINARIES

In our investigations we consider problems $f = (f_1, \ldots, f_m) : \{0, 1\}^n \to \mathbb{R}^m$. Throughout this paper, we assume that each function $f_i, 1 \le i \le m$ should be maximized. As there is no single point that maximizes all functions simultaneously, the goal is to find a set of so-called Pareto-optimal solutions.

Definition 2.1 (Pareto optimality). Let $f : X \to F$, where $X \subseteq \{0, 1\}^n$ is called decision space and $F \subseteq \mathbb{R}^m$ objective space. The elements of X are called decision vectors and the elements of F objective vectors. A decision vector $x \in X$ is Pareto optimal if there is no other $y \in X$ that dominates x. y dominates x, denoted as $y \succ x$, if $f_i(y) \ge f_i(x)$ for all $i = 1, \ldots, m$ and $f_i(y) > f_i(x)$ for at least one index i. A decision vector y weakly dominates x, denoted by $y \succeq x$, if $f_i(y) \ge f_i(x)$, for all i. If neither $y \preceq x$ nor $y \succeq x$, both decision vectors are incomparable, denoted by $y||x$. The set of all Pareto-optimal decision vectors X^* is called Pareto set. $F^* = f(X^*)$ is the set of all Pareto-optimal objective vectors and denoted as Pareto front.

OneMinMax and Lotz are ideal benchmark functions since both facilitate the theoretical analysis. This choice also allows comparisons with previous approaches such as [9, 10, 12]. OneMinMax (see Definition 2.2) has the particularity that every single solution represents a point in the Pareto front, no search point is strictly dominated by another. The goal is to cover the Pareto front, i. e. finding individuals with i ones, for all $0 \le i \le n$.

Definition 2.2 (OneMinMax). A pseudo-Boolean function with the objective functions

$$\text{OneMinMax}(x_1, \ldots, x_n) := \left(n - \sum_{i=1}^n x_i, \sum_{i=1}^n x_i \right),$$

where the aim is to maximize the number of zeroes and ones at the same time.

In the case of Lotz (see Definition 2.3), all non-Pareto optimal decision vectors only have Hamming neighbours that are either better or worse, but never incomparable to it. This fact facilitates the analysis of the population-based algorithms, which certainly cannot be expected from other multi-objective optimisation problems. Note that the Pareto front for Lotz is given by the set of $n + 1$ search points $\{1^i 0^{n-i} \mid 0 \le i \le n\}$.

Definition 2.3 (Leading Ones, Trailing Zeroes, Lotz). A pseudo-Boolean function $\{0, 1\}^n \to \mathbb{N}^2$ defined as

$$\text{Lotz}(x_1, \ldots, x_n) = \left(\sum_{i=1}^n \prod_{j=1}^i x_j, \sum_{i=1}^n \prod_{j=i}^n (1 - x_j) \right),$$

where the goal is to simultaneously maximize the number of leading ones and trailing zeroes in a bit-string.

We focus our analysis on two simple MOEAs, SEMO and its variant called Global SEMO (GSEMO) because of their simplicity and suitability for a rigorous theoretical analysis. SEMO starts with an initial solution $s \in \{0, 1\}^n$ chosen uniformly at random. All non-dominated solutions are stored in the population P. Then, it selects a solution s uniformly at random from P, and a new search point s' its created due to the mutation step by flipping the i-th bit (chosen uniformly at random from $i \in \{1, \ldots, n\}$) of s. The new population contains for each non- dominated fitness vector $f(s)$, $s \in P \cup \{s'\}$, one corresponding search point (dominated individuals are removed from the population), and in the case where $f(s')$ is not dominated, s' is added to P.

In the case of GSEMO, a new solution s' is created by flipping each bit from a solution s independently with probability $1/n$ (for a formal definition of both algorithms see [10] or Algorithm 2 where the uniform selection has been replaced with the diversity-parent selection scheme). For SEMO, we know that the expected running time on OneMinMax is at most $O(n^2 \log n)$ [10]. We prove that this upper bound is asymptotically tight.

THEOREM 2.4. *The expected time for SEMO to cover the whole Pareto front on OneMinMax is $\Theta(n^2 \log n)$.*

PROOF. The upper bound was shown in [10]. For the lower bound, let $|x|_1$ denote the number of 1-bits and $|x|_0$ denotes the number of 0-bits in x. Define $X_t := \min_{x \in P_t} \{|x|_1\}$ if for the initial search point x_0 we have $|x_0|_1 \ge n/2$, and $X_t := \min_{x \in P_t} \{|x|_0\}$ otherwise. Note that, by definition, $X_0 \ge n/2$. Now, $X_t = 0$ is a necessary requirement for covering the whole Pareto front at time t. Hence we lower-bound the sought time by the expected time for X_t to reach value 0.

Since only local mutations are used, X_t can only decrease by 1. In order to decrease X_t we have to select a parent with Hamming distance X_t to 0^n or 1^n, respectively, which happens with probability $1/|P_t|$. Note that $|P_t| \ge n/2 - X_t$ as the population contains

individuals with $X_t, X_t + 1, \ldots, \lceil n/2 \rceil$ ones. Moreover, mutation needs to flip one of the X_t bits differing to 0^n or 1^n, respectively. Hence

$$\text{Prob}(X_{t+1} = X_t - 1 \mid X_t) \le \frac{1}{n/2 - X_t} \cdot \frac{X_t}{n}.$$

The total expected time to decrease X_t to 0 is thus at least

$$\sum_{j=1}^{n/2} \left(\frac{n}{2} - j \right) \frac{n}{j} = \sum_{j=1}^{n/2} \frac{n^2}{2j} - \sum_{j=1}^{n/2} n = \frac{n^2 \ln n}{2} - O(n^2)$$

as $\sum_{j=1}^{n/2} 1/j \ge \ln n/2 = \ln n - \ln 2$. $\qquad\qquad\square$

The reason for the relatively high running time is that the growing population slows down exploration. The population can only expand on the Pareto front in case search points with the current highest or lowest number of ones is chosen (corresponding to a minimum X_t-value in the proof of Theorem 2.5). Once the population has grown to a size of $\mu = \Theta(n)$, the probability that this happens has decreased to $\Theta(1/n)$. This means that only a $\sim 1/n$-th fraction of the time the algorithm has a chance to expand on the Pareto front! Uniform parent selection means that most steps are spent idling. The same effect occurs for SEMO on Lotz as proved in [12].

Theorem 2.5 (Lemma 2 in [12]). *The expected time for SEMO to cover the whole Pareto front on Lotz is $\Theta(n^3)$.*

For GSEMO we have upper bounds of the same order, $O(n^2 \log n)$ for OneMinMax [10] and $O(n^3)$ for Lotz [9], though no lower matching bounds are available in the literature.

We remark that Lotz can also be optimised more efficiently, in time $O(n^2)$, by a tailored algorithm that uses local search along individual objectives during initialisation to locate both extreme points of the Pareto front, 0^n and 1^n, and then uses crossover to produce the whole Pareto front from these points [15]. Incorporating a fairness mechanism which makes sure that each individual produces roughly the same number of offspring into SEMO leads to the algorithm FEMO. For FEMO a runtime bound of $\Theta(n^2 \log n)$ has been given in [12]. The runtime analysis provided for IBEA in [13] gives an upper bound of $O(n^2 \log n)$ and $O(n^3)$ for OneMinMax and Lotz, respectively, if the population size is set to $n + 1$ and therefore does not improve on the results for SEMO given in [12].

Our aim is to develop rigorous runtime bounds of SEMO and GSEMO introducing different diversity-based parent selection. We want to study how these mechanisms help to improve the performance of the MOEAs.

3 DIVERSITY-BASED PARENT SELECTION

Hypervolume-based EAs have become very popular in recent years for multi-objective optimization where the hypervolume indicator is used as a measurement of the coverage of the population [1, 16]. The hypervolume indicator measures a set of elements corresponding to images of the individuals with the volume of the dominated portion of the objective space. It is calculated based on the selection of a reference point. In particular, given a reference point $r \in \mathbb{R}^m$,

the hypervolume indicator is defined on a set $P \subset S$ as

$$I_H(P) = \lambda \left(\bigcup_{x \in P} [f_1(x), r_1] \times [f_2(x), r_2] \times \cdots \times [f_m(x), r_m] \right)$$

where $\lambda(S)$ denotes the Lebesgue measure of a set S and $[f_1(a), r_1] \times [f_2(a), r_2] \times \cdots \times [f_m(a), r_m]$ is the orthotope with $f(a)$ and r in opposite corners. We define the contribution of an element $x \in P$ to the hypervolume of a set of elements P as

$$c(x, P) = I_H(P) - I_H(P \setminus \{x\}).$$

The calculation of hypervolume indicator and the calculation of the contribution are both NP-hard when the number of objectives m is a parameter [3, 4]. However, both can be computed in polynomial time if m is fixed. In the following, for bi-objective problems like OneMinMax and Lotz, we can directly calculate the contribution of an element by taking into account the two direct neighbours in the objective space as follows.

Definition 3.1 (Hypervolume contribution). For a given reference point $r = (r_1, r_2)$, we set $f_1(x_0) = r_1$ and $f_2(x_{\mu+1}) = r_2$ where x_0 and $x_{\mu+1}$ are individuals used to estimate the hypervolume contribution, and hereinafter μ denotes the size of the current population in SEMO/GSEMO. Furthermore, we assume that $r_1 = f_1(x_0) < f_1(x_1)$, $r_2 = f_2(x_{\mu+1}) < f_2(x_\mu)$. Let the population be sorted according to the value of $f_1(x_i)$ such that

$$f_1(x_0) < f_1(x_1) < f_1(x_2) < \cdots < f_1(x_\mu).$$

The contribution of an individual x_i to the hypervolume of a population P is then given by

$$HVC(x_i, P) = (f_1(x_i) - f_1(x_{i-1})) \cdot (f_2(x_i) - f_2(x_{i+1})).$$

Another diversity metric applied to our framework is the *crowding distance* used by the NSGA-II defined in [6]. The crowding distance operator is a density metric of solutions surrounding a particular solution in the population used to determine their extent proximity with other solutions. A solution with a lower crowding distance value implies that the region occupied by this solution is crowded by other solutions. The solutions with a higher crowding distance value are chosen/preferred for reproduction.

Now, since both SEMO and GSEMO use a population of non-dominated individuals, i. e. all individual have the minimum non-domination rank possible, we can directly apply the crowding distance as our diversity metric (Algorithm 1). The population is sorted according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference of the function values of two adjacent solutions (see Line 9 of Algorithm 1, $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of the m-th objective function).

As in previous theoretical studies, we measure the running time as the number of function evaluations needed to fully cover the Pareto front. This common practice is motivated by the fact that function evaluations are often the most time-consuming operations. Note that for SEMO and GSEMO the number of function evaluations coincides with the number of generations needed as each generation only creates one new offspring whose fitness is evaluated.

Algorithm 1 Crowding Distance Operator

1: Let $l := |P|$.
2: **for all** i individuals $\in P$ **do**
3:   Set $P[i]._{distance} := 0$
4: **end for**
5: **for all** m objectives **do**
6:   Sort P according to m objective function value in ascending order.
7:   $P[1]._{distance} := P[l]._{distance} := \infty$.
8:   **for** $i = 2$ to $l - 1$ **do**
9:     $P[i]._{distance} := P[i]._{distance} + \frac{P[i+1]._m - P[i-1]._m}{f_m^{max} - f_m^{min}}$
10:   **end for**
11: **end for**

For the hypervolume contribution (HVC), according to Definition 3.1, the reference point can be defined so that the current extreme individuals in the population and individuals in intermediate empty areas have a high diversity score, and a strong influence for the algorithm. In the case of the crowding distance contribution (CDC) the same behaviour applies, extreme points in the search space receive a high distance while intermediate individuals surrounded by empty areas receive a higher distance than the ones where the area is more crowded.

With this information we can define selection mechanisms capable of selecting those extreme points and pushing the spread of the population toward the outer areas of the search space. However, as our theoretical analysis will show, in case the population already contains the extreme points of the Pareto front (0^n and 1^n for One-MinMax and Lotz), we need to be flexible enough to ignore those points and select intermediate individuals surrounded by empty areas in the search space to fully populate the Pareto front.

The selection mechanisms defined in this paper use the previous diversity contribution metrics but any other metric can be easily applied that follows the behaviour mentioned before. Firstly, we define 3 different rank-based selection schemes in which the probability of selecting individuals with a high diversity score is higher than individuals with a lower diversity score (see Definition 3.2). The first is called *exponential*; it is a rather aggressive scheme that strongly favours the best-ranked individuals and has a very small tail. The second called *inverse quadratic*; it is much less aggressive with a fat tail and yet a constant probability of selecting the first constant ranks. And finally, the third ranking scheme is called *Harmonic*; it is the least aggressive scheme with a fat tail and only a probability of $O(1/(\log \mu))$ for selecting the best few individuals.

Definition 3.2 (Rank-based selection schemes). The probability of selecting the i-th ranked individual is

$$\frac{2^{-i}}{\sum_{j=1}^{\mu} 2^{-j}}, \quad \frac{1}{i^2 \cdot \sum_{j=1}^{\mu} \frac{1}{j^2}}, \quad \frac{1}{i \cdot \sum_{j=1}^{\mu} \frac{1}{j}}$$

for the exponential, inverse quadratic, and Harmonic ranking scheme, respectively.

Secondly, we use the classical TS, but with a specific tournament size of μ, the current size of the population. This means we choose μ individuals uniformly at random with replacement from the population and then select the individual with the highest diversity

contribution from this multi-set. Selection with replacement implies that there is a chance of not selecting particular individuals, while other individuals might be picked multiple times.

Now we introduce the diversity-based parent selection into SEMO (see Algorithm 2) and GSEMO. We remove the uniform parent selection from both algorithms and instead, we estimate the diversity contribution for all the individuals in the population, and a new individual is selected according to the diversity-based parent selection method, and continues as the original algorithms. Our parent selection mechanisms are not limited to these algorithms and may prove useful on a much broader class of MOEAs.

Algorithm 2 SEMO with diversity-based parent selection

1: Choose an initial solution $s \in \{0, 1\}^n$ uniformly at random.
2: Determine $f(s)$ and initialize $P := \{s\}$.
3: **loop**
4:   Estimate diversity contribution $\forall s \in P$.
5:   Choose $s \in P$ according to the parent selection mechanism.
6:   Choose $i \in \{1, \ldots, n\}$ uniformly at random.
7:   Define s' by flipping the i-th bit of s.
8:   **if** s' is not dominated by any individual in P **then**
9:     Add s' to P, and remove all individuals weakly dominated by s' from P.
10:   **end if**
11: **end loop**

---

4 ON DIVERSITY-BASED PROGRESS

We show that diversity-based parent selection mechanisms can achieve a fast spread on the Pareto front F^*. The following arguments and analyses consider the situation where the population is located on the Pareto front. This is trivially the case for One-MinMax as all search points are Pareto-optimal. For Lotz we later supply a separate analysis that covers the process of reaching the Pareto front.

For OneMinMax and Lotz the most promising parents are those that have a Hamming neighbour that is on the Pareto set, but not yet contained in the population. We call these search points *good*:

Definition 4.1. With reference to a population P and a fitness function with Pareto front F^*, we call a search point $x \in P \cap X^*$ *good* if there is a Hamming neighbour y of x such that $y \in X^*$ but $y \notin P$. Otherwise, x is called *bad*.

A diversity measure should encourage the selection of such good individuals.

Definition 4.2 (diversity-favouring). We call a measure $C(x, P)$ *diversity-favouring* on $S \subseteq \{0, 1\}^n$ with respect to a fitness function with Pareto front F^* if for all populations P and all $x, y \in P \cap X^* \cap S$ we have the following: if x is bad and y is good then $C(x, P) < C(y, P)$.

Note that the definition is restricted to a subset S of the search space. The reason is to allow the exclusion of certain search points for which the property is not true. For OneMinMax and Lotz, the property does not hold for the extreme points on the Pareto front, 0^n and 1^n. We show that both HVC and CDC are both diversity-favouring on all other search points.

LEMMA 4.3. *The hypervolume contribution $HVC(x, P)$ is diversity-favouring on $\{0, 1\}^n \setminus \{0^n, 1^n\}$ for both OneMinMax and Lotz if the reference point is dominated by $(-1, -1)$.*

PROOF. Let us consider an individual $x_i \notin \{0^n, 1^n\}$ of the sorted population according to f_1, using the notation from Definition 3.1. If x is bad, then there are Hamming neighbours x_{i-1} and x_{i+1} of x_i in P, the $HVC(x_i, P)$ is the minimum possible, since $f_1(x_i) - f_1(x_{i-1}) = 1$ and $f_2(x_i) - f_2(x_{i+1}) = 1$ yielding $HVC(x_i, P) = (f_1(x_i) - f_1(x_{i-1})) \cdot (f_2(x_i) - f_2(x_{i+1})) = 1$.

Now, let us consider a good search point y_i, that is, y_{i-1} or y_{i+1} is not a Hamming neighbour of y_i. Then we have $f_1(y_i) - f_1(y_{i-1}) > 1$ or $f_2(y_i) - f_2(y_{i+1}) > 1$ and in any case $HVC(y_i, P) = (f_1(y_i) - f_1(y_{i-1})) \cdot (f_2(y_i) - f_2(y_{i+1})) > 1$. Thus $HVC(y_i, P) > HVC(x_i, P)$, which completes the proof. □

LEMMA 4.4. *The crowding distance contribution $CDC(x, P)$ is diversity-favouring on $\{0, 1\}^n \setminus \{0^n, 1^n\}$ for both OneMinMax and Lotz.*

PROOF. By Algorithm 1 the search points with the minimum and maximum f_1 score in the population are going to have infinite diversity score, regardless of the objective chosen to sort the population.

Let us say that there is a bad individual x_i with a Hamming neighbours x_{i-1} and/or x_{i+1} contained in P. According to the numerator of Line 9 of Algorithm 1, the difference between the $f_1(x_{i-1})$ (or $f_2(x_{i-1})$) and $f_1(x_{i+1})$, is the minimum possible, which means the minimum $CDC(x_i, P)$ is assigned to the individual x_i.

In the case of a good search point y_i, that is, y_{i-1} or y_{i+1} are not Hamming neighbours of y_i, the difference between the next contained search points in P is higher, if the difference between $f_1(y_i)$ (or $f_2(y_i)$) is higher than the minimum possible, this means $CDC(x_i, P) < CDC(y_i, P)$ which completes the proof. □

Note that in both above measures 0^n and 1^n, if contained in the population, will always receive a high score, regardless of whether they are good or bad. If they are bad, there is a high chance that a bad individual will be selected as parent in a diversity-based parent selection mechanisms. With this in mind, the probability of selecting a good individual can be bounded from below as follows.

LEMMA 4.5. *Let $C(x, P)$ be a diversity-favouring measure on $\{0, 1\}^n \setminus \{0^n, 1^n\}$. Consider either OneMinMax or Lotz and assume the population P is a subset of the Pareto set, $P \subseteq X^*$. Imagine P being sorted according to non-increasing $C(x, P)$ values. Consider a parent selection mechanism based on $C(x, P)$ such that r_i is the probability of selecting the i-th element of P in the sorted sequence. Then the probability of selecting a good individual is at least $\min\{r_1, r_2, r_3\}$ unless P already covers the Pareto front.*

PROOF. Before the whole Pareto front is covered by the population P, there exists at least one good individual x in population P with no corresponding Hamming neighbour s in the Pareto set X^*. Then the individuals which correspond to the Hamming neighbours of the missing point s are good search points.

Since $C(x, P)$ is defined as a diversity-favouring measure on $\{0, 1\}^n \setminus \{0^n, 1^n\}$, the good search points have higher contribution than bad search points that are neither 0^n nor 1^n. Therefore, among

the top three ranked elements in P, there exists at least one good individual. The probability of selecting this good individual is at least $\min\{r_1, r_2, r_3\}$. □

The parent selection mechanisms thus have the following probability of selecting good individuals.

LEMMA 4.6. *In the setting described in Lemma 4.5, the probability p_{good} of selecting a good individual is*

(1) $\Omega(1)$ *for the exponential and inverse quadratic ranking schemes,*
(2) $\Omega(1/\log \mu)$ *for the Harmonic ranking scheme,*
(3) $\Omega(1)$ *for TS with tournament size μ.*

PROOF. For the parent selection with the exponential ranking scheme, the probability follows from Lemma 4.5, which fulfills

$$r_1 \geq r_2 \geq r_3 = \frac{2^{-3}}{\sum_{j=1}^{\mu} 2^{-j}} \geq 2^{-3} = \Omega(1).$$

For the inverse quadratic ranking scheme, since $\sum_{j=1}^{\mu} \frac{1}{j^2} \leq \sum_{j=1}^{\infty} \frac{1}{j^2} = \pi^2/6$, the probability fulfils

$$r_1 \geq r_2 \geq r_3 = \frac{1}{3^2 \cdot \sum_{j=1}^{\mu} \frac{1}{j^2}} \geq \frac{2}{3 \cdot \pi^2} = \Omega(1).$$

In the case of Harmonic ranking scheme, since $\sum_{j=1}^{\mu} \frac{1}{j} \leq \ln \mu + 1$, the probability fulfils

$$r_1 \geq r_2 \geq r_3 = \frac{1}{3 \cdot \sum_{j=1}^{\mu} \frac{1}{j}} \geq \frac{1}{3 \cdot (\ln \mu + 1)} = \Omega(1/\log \mu).$$

For the TS, the probability of selecting a good individual is at least $\min\{r_1, r_2, r_3\}$ and $r_1 \geq r_2 \geq r_3$. In order for the individual with the 3rd maximum contribution to be selected in the TS, the individuals with the 1st and 2nd maximum contribution should never be selected in the μ times (probability of $(1 - 2/\mu)^\mu$). And, conditional on this happening, the individual with the 3rd maximum contribution has to be chosen at least once amongst the other $\mu - 2$ individuals in the μ times with probability $1 - \left(1 - \frac{1}{\mu-2}\right)^\mu$. Hence, the probability of selecting a good individual is at least

$$p_{\text{good}} \geq \left(1 - \left(1 - \frac{1}{\mu - 2}\right)^\mu\right) \cdot \left(1 - \frac{2}{\mu}\right)^\mu \geq \left(1 - \frac{1}{e}\right) \cdot \left(1 - \frac{2}{\mu}\right)^\mu$$

using $\left(1 - \frac{1}{x}\right)^x \leq 1/e$ for $x > 1$. Since $f(x) = \left(1 - \frac{1}{x}\right)^x$ is non-decreasing when $x \geq 1$, with $\mu \geq 3$, $\left(1 - \frac{2}{\mu}\right)^{\frac{\mu}{2}} \geq \left(1 - \frac{2}{3}\right)^{\frac{3}{2}} \geq 0.19$. Therefore, $p_{\text{good}} \geq \left(1 - \frac{1}{e}\right) \cdot 0.19^2 = \Omega(1)$. □

5 SPEEDUPS ON ONEMINMAX

For any parent selection mechanism defined before, the parent selection is focused on selecting an individual with a high diversity score. In the case of HVC or CDC, having a high diversity contribution means that, apart from the possible exceptions of 0^n and 1^n, the parent will be good, i. e. located in a less populated area of the Pareto front. We show that by preferring good individuals in the parent selection, SEMO and GSEMO can quickly find the whole Pareto front for OneMinMax.

LEMMA 5.1. *Suppose that the probability of selecting a good individual is at least p_{good}. Then the expected runtime for SEMO or GSEMO to find all solutions in the Pareto front on ONEMINMAX is bounded above by $O((n \log n)/p_{\text{good}})$.*

PROOF. We call a step a *relevant step* if the algorithm selects a good parent on the Pareto front. We show in the following that $O(n \log n)$ relevant steps are sufficient for covering the whole Pareto front of ONEMINMAX, regardless of irrelevant steps performed. This shows the claim as the expected time for a relevant step is $1/p_{\text{good}}$.

We use the *accounting method* (see, e. g. Section 17.2 in [5]) to bound the number of relevant steps. Specifically, we count the number of relevant steps spent selecting a good parent with i ones. Summing up (upper bounds on) all these times across all $0 \le i \le n$ will imply the claim.

Note that, once potential gaps at $i-1$ and $i+1$ are filled, there can be no more relevant steps at i ones, due to the definition of a relevant step. Hence the expected number of relevant steps at i ones is bounded by the expected number of mutations from i needed to fill both these gaps. If an individual with i ones, $0 < i < n$, is selected as parent, the probability of mutation creating an individual with $i-1$ ones is at least $i/n \cdot (1-1/n)^{n-1} \ge i/(en)$, and the probability of mutation creating an individual with $i+1$ ones is at least $(n-i)/n \cdot (1-1/n)^{n-1} \ge (n-i)/(en)$ (this holds both for SEMO and GSEMO; for SEMO the factor $1/e$ can be removed). The time for filling both gaps is at most $en/i + en/(n-i)$. Hence there are at most $en/i + en/(n-i)$ relevant steps selecting a parent with i ones. In the special cases of $i = 0$ or $i = n$ the time to fill the neighbouring gaps simplifies to $en/n = e$.

Summing over all i, the expected total number of relevant steps is hence at most

$$2e + \sum_{i=1}^{n-1} \left(\frac{en}{i} + \frac{en}{n-i} \right) = 2e + 2 \sum_{i=1}^{n-1} \frac{en}{i} = 2 \sum_{i=1}^{n} \frac{en}{i} \le 2enH(n).$$

As $H(n) = O(\log n)$ this completes the proof. □

Combining Lemma 4.6 and Lemma 5.1, we have proved the following results. Note that the population size μ is always at most $n + 1$ on ONEMINMAX and LOTZ, hence for the Harmonic ranking scheme, $p_{\text{good}} = \Omega(1/\log \mu) = \Omega(1/\log n)$.

THEOREM 5.2. *The expected time for SEMO and GSEMO to find the whole Pareto front on ONEMINMAX is bounded by $O(n \log n)$ for the exponential and inverse quadratic ranking schemes, and for TS with tournament size μ. It is bounded by $O(n \log^2 n)$ for the Harmonic ranking scheme.*

As both SEMO and GSEMO with the classical uniform parent selection need time $\Theta(n^2 \log n)$ on ONEMINMAX, our parent selection schemes lead to speedups of order $\Theta(n)$ and $\Theta(n/\log n)$, respectively.

6 SPEEDUPS ON LOTZ

We now turn to the function LOTZ. In contrast to ONEMINMAX, where all individuals are Pareto optimal, for LOTZ we have to estimate the time for the population to reach the Pareto front. For SEMO the approach to the Pareto front can be estimated easily as before reaching the Pareto front, SEMO keeps only one individual

in the population. For local mutations as used in SEMO, whenever an offspring is created, either the offspring dominates the parent, or the parent dominates the offspring (or both, if they have the same function values). The population size remains unchanged before there is a solution on the Pareto front. For any parent on the Pareto front, SEMO only accepts its offspring if it is also on the Pareto front, otherwise the offspring is dominated by the parent.

LEMMA 6.1. *The expected time for SEMO to reach the Pareto front is $O(n^2)$. Assume that afterwards the probability of selecting a good individual in the population is at least p_{good}. The expected runtime for SEMO to reach a population covering the whole Pareto front on LOTZ is bounded above by $O(n^2/p_{\text{good}})$.*

PROOF. The time for the population to find the first Pareto-optimal point is $O(n^2)$ and has already been proved in Lemma 1 in [12]. So we can focus on the time required to find the whole Pareto front. When there exists at least one good solution, among all possible offspring from a good solution, only the offspring that is also on the Pareto front will be accepted according to Algorithm 2. Using the *accounting method* used to prove Lemma 5.1, we count the number of relevant steps spent selecting a good parent with i leading ones, $1^i 0^{n-i}$, and sum up all these times across all $0 \le i \le n$ to prove the claim.

The potential gaps consist of non-existing non-dominated individuals at $i-1$ and $i+1$ ($1^{i-1}0^{n-i+1}$ and $1^{i+1}0^{n-i-1}$, respectively), it is necessary to fill those gaps, hence there can be no more relevant steps at i leading ones. So the expected number of mutations at i leading ones is bounded by the expected number of mutations from i needed to fill $i-1$ and $i+1$. If $1^i 0^{n-i}$ is selected as parent, the probability of mutation creating $1^{i-1}0^{n-i+1}$ or $1^{i+1}0^{n-i-1}$ is $1/n$, respectively. The time for filling both gaps (if existent) is at most $n + n$. Hence there are at most $2n$ relevant steps selecting a parent with i leading ones.

Summing over all i, the expected total number of relevant steps is hence at most

$$\sum_{i=0}^{n} 2n = 2n(n+1) = O(n^2).$$

Noting that the expected waiting time for a relevant step is $1/p_{\text{good}}$, the overall expected runtime for SEMO to achieve a population covering the whole Pareto front of LOTZ is upper bounded by $O(n^2) + O(n^2/p_{\text{good}}) = O(n^2/p_{\text{good}})$. □

Combining Lemma 4.6 and Lemma 6.1, we now have proved the following results.

THEOREM 6.2. *The expected time for SEMO to find the whole Pareto front on LOTZ is bounded by $O(n^2)$ for the exponential and inverse quadratic ranking schemes, and for TS with tournament size μ. It is bounded by $O(n^2 \log \mu)$ for the Harmonic ranking scheme.*

The analysis of GSEMO turns out to be more difficult than the analysis of SEMO. The reason is that the approach to the Pareto front becomes harder to analyse. With global mutations, GSEMO can create incomparable search points while approaching the Pareto front. This means that the population can expand in size while approaching the Pareto front, and even after the whole population has reached the Pareto front, it is possible to create search points off the Pareto front that are accepted in the population.

Experiments in Section 7 indicate that this behaviour does not slow down the algorithm by more than a constant factor. However, proving that the bound $O(n^2)$ for SEMO also holds for GSEMO turns out to be very challenging and is left for future work.

7 EXPERIMENTS

The experimental approach is focused on the analysis of SEMO and GSEMO and its performance with and without the diversity-based parent selection mechanisms. We are interested in observing if we can speed up the performance from the classical approaches.

This also allows a more detailed comparison of the HVC, the CDC, and the parent selection methods. In the case of the HVC, we have defined two reference points, $(-1, -1)$ and $(-n, -n)$. For the first reference point, a slight preference to the extreme points is provided while with the second, the influence of the extreme points become very strong. This particular characteristic became an interesting feature to observe in the case of the ranking-based selection schemes, and expose a potential flaw for the case of HVC with low (or high in the case of minimisation) reference point or CDC (since it assigns infinite value to the extreme points) and the parent selection mechanisms that focus very aggressively toward the extreme points, as we shall see below.

Since we are interested in the time required to find the Pareto front, we have defined that outcome as a stopping criterion, and we repeat the experimental framework for 100 runs with problem size $n = 100$ for all algorithmic approaches and report the mean and standard deviation (STD) as our metrics of interest.

In Table 1 we have summarized and divided the results of our experimental framework into 2 sections. The first section (upper part), refers to the mean and STD of generations required to find the Pareto front for the classic SEMO and GSEMO that use uniform parent selection for both test functions. The second section (lower part), refers to the mean and STD of generations required to find the Pareto front for SEMO and GSEMO with the different diversity-based parent selection schemes.

As we mentioned before, a parent selection mechanisms that is extremely focused on the extreme points can be potentially dangerous, and to exemplify this, we have introduced a deterministic selection mechanism which we have named Highest Diversity Contribution (HDC): always select the individual with the highest diversity contribution. We also have defined a modified version of the uniform random selection used by SEMO and GSEMO, that we call *non-minimum uniform at random selection (NMUARS)*, where the individuals with the minimum diversity score in the population are ignored and one individual is selected uniformly at random from all individuals with a non-minimum diversity score. In this sense individuals with high diversity score have better probabilities to be selected and the approach is flexible enough to choose between extreme and intermediate individuals.

As can be seen in Table 1, HDC fails to find the Pareto front for OneMinMax and Lotz in the case of GSEMO for both diversity-based metrics. Due to the mutation mechanism, once it starts expanding the Pareto set, the algorithm may create an offspring far from the parent, leaving unexplored areas between them, and since the parent selection is only focused on the current extreme individuals, it will expand the Pareto set until it reaches the individuals

$\{0^n, 1^n\}$, and it will continue selecting those individuals ignoring the intermediate ones, leaving the population in a *stagnation* state. This observation also justifies why we introduced parent selection schemes of varying degree of aggressiveness.

Finally, for any other parent selection defined in this paper, we have achieved an significant speed up in the performance of SEMO and GSEMO of around one order of magnitude. As can be observed in Table 1, SEMO and GSEMO with diversity-based parent selection mechanisms are able to find the Pareto front faster than its classical versions, i. e. fewer generations are required for both test functions. Note that the problem size $n = 100$ is relatively moderate; as our theoretical results prove, speedups over the original algorithms will grow further when the problem size is increased.

8 CONCLUSIONS

Diversity plays a crucial role in the area of EMO. So far, diversity-based parent selection has not been the main focus on algorithm design. In this paper, we have shown that diversity-based parent selection can significantly speed up EMO algorithms. Our theoretical results on OneMinMax and Lotz show that a linear factor can be saved for the investigated settings and this is confirmed by our experimental results showing a speedup of one magnitude for problems of size $n = 100$.

For future work, it would be interesting to study the benefit of diversity-based parent selection on more complex problems. From a theoretical perspective, combinatorial optimization problems such as minimum spanning tress and covering problems for which SEMO has already been studied would be natural candidates. On the experimental side, it would be interesting to integrate the presented diversity-based parent selection methods into state-of-art EMO algorithms and evaluate their performance on well-established benchmark sets.

REFERENCES

[1] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. 2012. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science* 425 (2012), 75 – 103.

[2] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler. 2001. Multiobjective Genetic Programming: Reducing Bloat by Using SPEA2. In *Congress on Evolutionary Computation (CEC 2001)*. IEEE, 536–543.

[3] Karl Bringmann and Tobias Friedrich. 2010. An Efficient Algorithm for Computing Hypervolume Contributions. *Evol. Comput.* 18, 3 (Sept. 2010), 383–402.

[4] Karl Bringmann and Tobias Friedrich. 2012. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science* 425 (2012), 104–116.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (3rd ed.). The MIT Press.

Table 1: First section. Mean (first rows) and STD (second rows) of generations required to find the Pareto front for SEMO and GSEMO on OneMinMax and Lotz with $n = 100$. Second section. Mean (first rows) and STD (second rows) of generations required to find the Pareto front for SEMO and GSEMO with diversity-based parent selection methods on OneMinMax and Lotz with $n = 100$.

	OneMinMax			Lotz		
SEMO	4.16E+04			3.17E+05		
	1.15E+04			5.34E+04		
GSEMO	1.06E+05			6.58E+05		
	3.47E+04			1.12E+05		
Algorithms	$HVC(-1,-1)$	$HVC(-n,-n)$	CDC	$HVC(-1,-1)$	$HVC(-n,-n)$	CDC
SEMO & HDC	9.14E+02	8.90E+02	1.05E+03	1.24E+04	1.25E+04	1.41E+04
	1.76E+02	1.65E+02	2.40E+02	9.79E+02	1.22E+03	1.80E+03
GSEMO & HDC	2.12E+03	Stagnation	Stagnation	3.06E+04	Stagnation	Stagnation
	4.28E+02	Stagnation	Stagnation	2.62E+03	Stagnation	Stagnation
SEMO & NMUAR	8.92E+02	1.05E+03	1.03E+03	1.25E+04	1.38E+04	1.41E+04
	1.81E+02	2.72E+02	2.59E+02	1.10E+03	1.49E+03	1.52E+03
GSEMO & NMUAR	2.14E+03	2.54E+03	2.58E+03	3.17E+04	3.50E+04	3.58E+04
	4.97E+02	6.57E+02	7.86E+02	3.13E+03	3.85E+03	3.75E+03
SEMO & Exponential	1.28E+03	1.27E+03	1.36E+03	1.57E+04	1.58E+04	1.78E+04
	2.72E+02	2.71E+02	3.44E+02	1.31E+03	1.33E+03	2.47E+03
GSEMO & Exponential	3.21E+03	3.18E+03	3.24E+03	3.45E+04	4.00E+04	5.87E+04
	9.35E+02	9.12E+02	7.72E+02	2.87E+03	8.60E+03	1.63E+04
SEMO & Harmonic	3.05E+03	3.24E+03	3.28E+03	3.14E+04	3.08E+04	3.53E+04
	6.97E+02	8.63E+02	8.03E+02	3.60E+03	3.24E+03	5.68E+03
GSEMO & Harmonic	7.89E+03	7.26E+03	8.03E+03	6.69E+04	6.33E+04	6.73E+04
	1.90E+03	1.69E+03	2.09E+03	7.23E+03	7.40E+03	1.02E+04
SEMO & Inverse Quadratic	1.15E+03	1.24E+03	1.34E+03	1.54E+04	1.51E+04	1.69E+04
	2.48E+02	2.89E+02	3.00E+02	1.26E+03	1.36E+03	2.13E+03
GSEMO & Inverse Quadratic	2.87E+03	2.85E+03	3.32E+03	3.40E+04	5.03E+04	5.73E+04
	6.35E+02	6.22E+02	1.07E+03	3.30E+03	1.24E+04	1.43E+04
SEMO & TS (μ)	1.05E+03	1.08E+03	1.21E+03	1.38E+04	1.41E+04	1.55E+04
	2.24E+02	2.18E+02	3.09E+02	1.25E+03	1.12E+03	1.94E+03
GSEMO & TS (μ)	2.58E+03	2.60E+03	2.81E+03	3.16E+04	6.53E+04	7.87E+04
	5.48E+02	7.91E+02	7.34E+02	2.88E+03	2.15E+04	2.57E+04

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

[7] Benjamin Doerr, Wanru Gao, and Frank Neumann. 2016. Runtime Analysis of Evolutionary Diversity Maximization for OneMinMax. In *Proceedings of the 2016 Genetic and Evolutionary Computation Conference.* 557–564.

[8] Tobias Friedrich, Christian Horoba, and Frank Neumann. 2011. Illustration of fairness in evolutionary multi-objective optimization. *Theor. Comput. Sci.* 412, 17 (2011), 1546–1556.

[9] Oliver Giel. 2003. Expected runtimes of a simple multi-objective evolutionary algorithm. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.,* Vol. 3. 1918–1925.

[10] Oliver Giel and Per Kristian Lehre. 2010. On the Effect of Populations in Evolutionary Multi-Objective Optimisation. *Evolutionary Computation* 18, 3 (2010), 335–356.

[11] Christian Horoba and Frank Neumann. 2010. Approximating Pareto-Optimal Sets Using Diversity Strategies in Evolutionary Multi-Objective Optimization. In *Advances in Multi-Objective Nature Inspired Computing.* 23–44.

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

[13] Anh Quang Nguyen, Andrew M. Sutton, and Frank Neumann. 2015. Population size matters: Rigorous runtime results for maximizing the hypervolume indicator. *Theor. Comput. Sci.* 561 (2015), 24–36.

[14] D. H. Phan, J. Suzuki, and P. Boonma. 2011. SMSP-EMOA: Augmenting SMS-EMOA with the Prospect Indicator for Multiobjective Optimization. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence.* 261–268.

[15] Chao Qian, Yang Yu, and Zhi-Hua Zhou. 2013. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence* 204 (2013), 99–119.

[16] Eckart Zitzler and Simon Künzli. 2004. Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference.* 832–842.