This is an author produced version of *Speeding Up Evolutionary Multi-objective Optimisation Through Diversity-Based Parent Selection*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/115996/

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

Edgar Covantes Osuna
Department of Computer Science
University of Sheffield
Sheffield S1 4DP, 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 S1 4DP, 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 analyses 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 its 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 analyses, theory

1 INTRODUCTION

Evolutionary algorithms have been shown to achieve high performing results for problems from multi-objective optimization. The area of evolutionary multi-objective optimization (EMO) design population-based evolutionary algorithms where the population is used to approximate the so-called Pareto front. Given that evolutionary algorithms use a population which is a set of solutions for a given problem, evolutionary algorithms are suited in a natural way

for computing trade-offs with respect to two (or more) conflicting objective functions.

Multi-objective evolutionary algorithms (MOEAs) 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 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. 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.

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. The parent selection mechanisms include ignoring individuals with a low (or 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 where the outcome of the tournament is defined according to the diversity score and not in the fitness values.

The main assumption is that individuals with a high diversity score (high hypervolume indicator or crowded-comparison method) are located in poorly explored or a less crowded areas of the search space, so the chances of creating new non-dominated individuals are better than 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 analyses that the use of the mentioned parent selection methods taking into account the diversity of the different search points can significantly improve the performance of MOEAs. The area of runtime analysis has contributed significantly to the theoretical understanding of EMO

algorithms [5, 7, 8] and allows to study different components of EMO methods from a rigorous perspective.

In order to gain insights into the potential benefits of the parent selection mechanisms, we study the classical functions OneMin-Max and Lotz problem introduced in [7] and [9], 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 Simple Evolutionary Multi-Objective (SEMO) algorithm. Other studies in the area of runtime analysis of MOEAs consider hypervolume-based algorithms [10] and MOEAs incorporating other diversity mechanisms for survival selection [8].

We show that the use of various parent selection mechanisms speeds up SEMO by a factor of 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 effects, 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.

GSEMO needs time $\Theta(n^3)$ on Lotz. Can we do better? Check work on FEMO [9]

2 PRELIMINARIES

We focus our analysis on two simple MOEAs, SEMO and its variant called Global SEMO (GSEMO) which uses a more general mutation operator. Our aim is to develop rigorous runtime bounds of SEMO and GSEMO introducing different parent selection mechanisms taking into account the diversity contribution of each individual in the population. We want to study how these diversity-parent selection mechanisms help to improve the performance of the MOEAs. So, we first define the diversity contribution metrics and then the diversity-parent selection mechanisms.

Diversity metrics used Hypervolume indicator, how the hypervolume indicator helps to the spread of the population?

How about if we use HYP for parent selection?

Runtime bound for $(\mu + 1)$SIBEA when $\mu \geq n + 1$ is $O(\mu n^2)$.

Drawbacks: Finding the individual with the approximately least contribution to hypervolume indicator is NP-hard [1, 2]. But we can define the contribution in a way that does not need the calculation of exact hypervolume indicator.

There is a paper [11] introducing an MOEA with parent selector using prospect indicator which is similar to hypervolume indicator. And for the survivor selection they use the hypervolume indicator. But they only focus on the experimental results.

If we give a biased possibility for the individual to be selected as parent to produce new offspring based on their contribution to hypervolume indicator, the possibility of getting new individual in the unexplored area should be higher(?). Larger contribution to the hypervolume indicator indicates that the neighbourhood is not fully covered.

Change the hypervolume contribution definition into algorithm format maybe?

Let the population be sorted according to the value of $f_1(x)$ such that $f_1(x_0) < f_1(x_1) < f_1(x_2) < \cdots < f_1(x_\mu)$.

Definition 2.1. The contribution of an individual s_i to the population diversity in objective space is defined as

$$c(s_i) = (f_1(s_i) - f_1(s_{i-1})) \cdot (f_2(s_{i+1}) - f_2(s_i)).$$

x_0 donotes the reference point and how to select a proper reference point is important as it influences the decision involving the extreme points. For the initial experiment, my suggestion is to use $(-1, -1)$ as the reference point.

The calculation of $c(x_i)$ is then $O(\mu \log \mu)$ since only two objectives here. (the definition is similar to the crowding distance in NSGA2)

One possible mutator can select one of the individuals with the least contribution $c(x_i)$ and flip the last 1-bit or the first 0-bit. Before the population covers the whole Pareto Front, there should be at least one of the mutations that can improve the coverage. (not sure whether we can improve the runtime bound or not...)

Other diversity metric applied to our framework is the *crowding distance* used by the Nondominated Sorting Genetic Algorithm II (NSGA-II) defined in [4]. 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. So, a solution with a lower crowding distance value, imply 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.

In the case of the NSGA-II, individuals are chosen for replacement by means of a binary tournament selection operator but the selection criterion is according to the crowded-comparison operator. Assuming that every individual s_i in the population has two attributes, the nondomination rank (lower the rank, the better) and the crowding distance, two solutions with different nondomination ranks, the solution with the lower rank is selected. Otherwise, if both solutions have the same rank, then the solution that is located in a lesser crowded region is chosen.

Now, since both SEMO and GSEMO use a population of nondominated individuals, i.e., all individual have a nondomination rank of 0, we ignore the nondomination rank decision step and we directly apply the crowding distance as our diversity metric. In the same way that in [4], we apply the crowding distance operator as shown in Algorithm 1.

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:         $P[i]._{distance} := P[i]._{distance} + (P[i + 1]._m - P[i - 1]._m)/(f_m^{max} - f_m^{min})$
10:     **end for**
11: **end for**

Although both diversity metrics follow different ideas about how to provide define areas that have not being fully explored in the search space, both promote the spread of the population toward the Pareto front in the same way.

Let us focus on the hypervolume contribution metric, according to the definition provided previously, 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 operator 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 push the spread of the population toward the extreme areas of the search space, and once the extreme points on the Pareto front have been found, be flexible enough to ignore the extreme 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. In first place, we have defined a modified version of the uniform random selection used by SEMO and GSEMO, we have called it *non-minimum uniform at random selection (NMUARS)*, where the individuals with the minimum diversity score are ignored and one individual is selected uniformly at random from the population with high diversity score, in this sense individuals with high diversity score have better chances to be selected and the approach is flexible enough to choose between extreme and intermediate individuals.

In second place, we have defined different ranking-based selection schemes in which the probability of selecting individuals with a high diversity score is higher than individuals with lower diversity score. For this selection mechanisms we have defined 3 selective pressures that provide different degrees of pressure to the best individuals. The first called *exponential*, it strongly favours the best-ranked individuals with a very small tail. The second called *inverse quadratic*, since $\sum_{j=1}^{\mu} \leq \sum_{j=1}^{\infty} = \pi^2/6$, there is a fat tail and yet a constant probability of selecting the first constant ranks. And finally, the thirds ranking scheme called *harmonic*, with a fat tail and only a probability of $O(1/(\log \mu))$ for selecting the best few individuals.

Definition 2.2 (Selective pressures). The probability of selecting the i-th ranked individual is

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

for the exponential, inverse quadratic and harmonic scheme, respectively.

And in third place, we use the classical *tournament selection* of size μ, we choose μ number of individuals uniformly at random from the population, then select the individual with the highest diversity contribution from this group.

Now that we have define the diversity contribution metrics, the parent selection methods, we can incorporate all these mechanisms

into the basic SEMO and GSEMO. In the case of SEMO (see Algorithm 2), it starts with a initial solution $s \in \{0, 1\}^n$ chosen uniformly at random. All non-dominated solutions are stored in the population P. 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 a new search point s' its created due to the mutation step. The new population contains for each non-dominated fitness vector $f(s)$, $s \in P \cup \{s'\}$, one corresponding search point, and in the case where $f(s')$ is not dominated s' is chosen.

Algorithm 2 SEMO

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 parent selection mechanism.
6:    Choose $i \in \{1, \ldots, n\}$ randomly.
7:    Define s' by flipping the ith bit of s.
8:    Determine $f(s')$.
9:    Let P be unchanged, if there is an $s'' \in P$ such that $f(s'') > f(s')$.
10:   Otherwise, exclude all s'' where $f(s') > f(s'')$ from P and add s' to P.
11: **end loop**

As mentioned before, in the case of GSEMO (see Algorithm 3) the steps 6 and 7 from SEMO are replaced with a more general mutation operator. A new solution s' is created by flipping each bit from a solution s independently with probability $1/n$ and as in Algorithm 2 we estimate the diversity contribution for all individuals in the population, and replace the uniform random selection approach with our parent selection scheme.

Algorithm 3 Global SEMO (GSEMO)

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 parent selection mechanism.
6:    Define s' by flipping each bit of s independently of the other bits with probability $1/n$.
7:    Determine $f(s')$.
8:    Let P be unchanged, if there is an $s'' \in P$ such that $f(s'') > f(s')$.
9:    Otherwise, exclude all s'' where $f(s') > f(s'')$ from P and add s' to P.
10: **end loop**

For our test functions we have considered the classical functions OneMinMax and Lotz because both algorithms and functions facilitate the theoretical analysis, also its choice allows comparisons with previous approaches such as [6, 7, 9]. Both consist in the maximization of a 2-dimensional vector valued function. Introduced in [7], OneMinMax (see Definition 2.3) is the multi-objective

version of the popular ONEMAX problem. ONEMINMAX has the particularity that every single solution represents a point in the Pareto front, no search point is strictly dominated by another search point. The objective is to find all individuals in the search space $\{0, 1\}^n$.

Definition 2.3 (ONEMINMAX). The ONEMINMAX problem is defined as 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 both objectives (the maximum number of zeroes and number of ones) at the same time.

In the case of LOTZ (see Definition 2.4), is the multi-objective version of the well-known LEADINGONES function and was introduced in [9]. One particular feature of LOTZ is that all non-Pareto optimal decision vectors only have 1-bit 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.

Definition 2.4 (Leading Ones, Trailing Zeroes, LOTZ). The pseudo-Boolean function LOTZ : $\{0, 1\}^n \to \mathbb{N}^2$ is 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.

3 FAST SPREAD ON PARETO FRONTS

We show that diversity-based parent selection mechanisms can achieve a fast spread on the Pareto front. The following arguments and analyses consider the situation where the population is located on the Pareto front. This is trivially the case for ONEMINMAX as all search points are on the Pareto front. 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 on the are those that have a Hamming neighbour that is on the Pareto front, but not yet contained in the population. We call these search points *good*:

Definition 3.1. With reference to a population P we call a search point $x \in P$ *good* if there is a Hamming neighbour of x that is on the Pareto front, but not contained in P. Otherwise, x is called *bad*.

A diversity measure should encourage the selection of such individuals.

Definition 3.2. We call a measure $C(x, P)$ *diversity-favouring* on $S \subseteq \{0, 1\}^n$ if for all populations P and all $x, y \in P \cap S$ we have $C(x, P) < C(y, P)$ if and only if x is x is bad and y is good.

Note that the definition is restricted to a subset S of the search space. The reason is to allow to exclude 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 hypervolume contribution and crowding distance contribution are both diversity-favouring.

LEMMA 3.3. *The hypervolume contribution $HVC(x, P)$ is diversity-favouring on $\{0, 1\}^n \setminus \{0^n, 1^n\}$ if the reference point is dominated by $(-1, -1)$.*

PROOF. To do   □

LEMMA 3.4. *The crowding distance contribution $CDC(x, P)$ is diversity-favouring on $\{0, 1\}^n \setminus \{0^n, 1^n\}$.*

PROOF. To do   □

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.

With this in mind, the probability of selecting a good individual can be bounded from below as follows.

LEMMA 3.5. *Let $C(x, P)$ be a diversity-favouring measure on $\{0, 1\}^n \setminus \{0^n, 1^n\}$. 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. To do   □

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

LEMMA 3.6. *In the setting described in Lemma 3.5, the probability of selecting a good individual is*

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

PROOF. To do
(we can be quite brief here, e. g. for exponential just write that the probability follows from Lemma 3.5 and

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

)   □

4 ONEMINMAX PROBLEM

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 hypervolume contribution or crowding distance, having a high diversity contribution means that, the area around an individual is empty or it is in a less populated area of the search space, in this sense defining a selection mechanism that favours those individuals to populate that area, and select the next less populated area seems to be a good approach to find all individuals in the Pareto front which is the case for ONEMINMAX.

So, for ONEMINMAX, it is just necessary that the selection mechanism provides the enough pressure to select a parent neighboured by an empty area, we define this empty area as a gap (see Definition 4.1).

Definition 4.1 (Gap). In the objective space, we refer to a missing point in a population on the Pareto front as a *gap*. In the following, we will denote p_{gap} as (a lower bound on) the probability of selecting a parent x that is neighboured to a gap: formally, there is a Hamming neighbour y of x such that y is Pareto-optimal, and the population does not include a search point with the same objective values as y.

Any gap can on any side of an individual in the Pareto front also, this gap increase the probabilities of selecting an individual as a parent.

LEMMA 4.2. *Suppose that the probability of selecting a parent on the Pareto front that is neighboured to a gap is at least p. Then the expected runtime for SEMO or GSEMO to find all solutions in the Pareto front on ONEMINMAX is bounded above by $O((n \log n)/p)$.*

PROOF. We call a step a *relevant step* if the algorithm selects a parent on the Pareto front such that it is neighbored to a gap 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 $1/p$.

We use the *accounting method* (see, e. g. Section 17.2 in [3]) to bound the number of relevant steps. Specifically, we count the number of relevant steps spent selecting a parent with i ones. Summing up (upper bounds on) all these times across all $0 \leq i \leq n$ will imply the claim.

Note that, once potential gaps at $i - 1$ and $i + 1$ (if existent) 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 $i/n \cdot (1 - 1/n)^{n-1} \geq i/(en)$, and the probability of mutation creating an individual with $i + 1$ ones is at least $(n-i)/n \cdot (1-1/n)^{n-1} \geq (n-i)/(en)$ (this holds both for SEMO and GSEMO; for SEMO the factor $1/e$ can be removed). The time for filling both gaps (if existent) is at most $en/i + en/(n - i)$. Hence the 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

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

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

Now we define the expected runtime for all the parent selection mechanisms defined before.

4.1 Non-minimum uniformly at random selection

LEMMA 4.3. *The probability of selecting an i good individual with non-minimum diversity score uniformly at random is*

$$\text{Prob}(i = 1) = 1.$$
$$\text{Prob}(i = 2) = \frac{1}{2}.$$
$$\text{Prob}(i = 3) = \frac{1}{2}.$$
$$\text{Prob}(i \geq 4) \geq \frac{i - 2}{i} \geq \frac{1}{2}.$$

PROOF. To proof this Lemma, first it is necessary to define the selection probability p. Due to the nature of the diversity mechanisms used, extreme points are always included in the selection pool since they always are going to have a non-minimum diversity score. This behaviour do not present a problem since choosing these points will contribute to the spread of the population in the Pareto front.

However, once we have found the extreme points in the Pareto front and if there still exists gaps in intermediate zones of the Pareto front, the algorithms will need to select between the intermediate and extreme individuals, once we have reached this point we can distinguish between 2 types of individuals, good and bad individuals. Good individuals are the intermediate individuals with a non-minimum diversity score and its selection will allow us to close a gap (make an improvement and continue with the remaining ones if there are more gaps) and the extreme points became bad individuals since they have a non-minimum diversity score but its selection do not contribute any more to the expansion of the Pareto front.

Now, since we start with only $i = 1$ good individual in the search space, this individual is selected with probability 1, after we create another search point as a result of mutation $i = 2$, both individuals became the extreme points, from this several scenarios may arise according to the location of those points in the Pareto front, the worst case scenario is when both points are located in the furthest area of the Pareto front making 1 search point bad, and the other good for selection. In this case we have defined pessimistically the probability of selecting the good individual at least $\geq 1/2$. The same scenario applies for $i = 3$, the intermediate individual is ignored by the selection scheme and the extreme point are considered for selection but only one is considered as good and the other bad so the probability of selecting the good individual again is at least $\geq 1/2$.

Finally, the probability of selecting a good individual may change according the position of the gaps in the search space, we have bounded pessimistically $p \geq \frac{i-2}{i} \geq 1/2$ for $i \geq 4$ where i represents the number of individuals with a non-minimum diversity score, excluding the two extreme points. Again, the worst case scenario is the same as the the the events mentioned above. □

4.2 Exponential scheme

LEMMA 4.4. *The expected runtime for finding all solutions on the Pareto front with SEMO or GSEMO with ranking-based exponential selection scheme on ONEMINMAX is bounded below by* $\Omega(1)$.

PROOF. As in the previous Lemma, it is necessary to define the selection probability p. And as mentioned before, we still to deal with the extreme points during all the evolution process. For this case it is just enough to define the probability of selecting the i-th ranked individual for all $i \in \{1, 2, 3\}$. So the probability of selecting the i-th ranked individual with the exponential scheme is defined as

$$\text{Prob(select } i\text{-th ranked individual)} = \frac{2^{-i}}{\sum_{j=1}^{\mu} 2^{-j}} = \frac{2^{-i}}{1 - 2^{-\mu}}.$$

The probability of selecting i-th ranked individual is at least 2^{-i}. So the probability of selecting of selecting the $i \in \{1, 2, 3\}$ ranked individual is

$$\text{Prob}(i = 1) \geq 2^{-1} = \frac{1}{2} = \Omega(1).$$

$$\text{Prob}(i = 2) \geq 2^{-2} = \frac{1}{4} = \Omega(1).$$

$$\text{Prob}(i = 3) \geq 2^{-3} = \frac{1}{8} = \Omega(1).$$

□

4.3 Harmonic scheme

LEMMA 4.5. *The expected runtime for finding all solutions on the Pareto front with SEMO or GSEMO with ranking-based harmonic selection scheme on ONEMINMAX is bounded above by* $O(1/\log \mu)$.

PROOF. As on the previous Lemma, we define the selection probability p as $1/(i \cdot H_\mu) \geq 1/(i \cdot (\ln(\mu) + 1))$. So the probability of selecting the i-th ranked individual with the exponential scheme is defined as

$$\text{Prob(select } i\text{-th ranked individual)} = \frac{1}{i \cdot \sum_{j=1}^{\mu} \frac{1}{j}} = \frac{1}{i \cdot H_\mu} \geq \frac{1}{i \cdot (\ln \mu + 1)}.$$

So the probability of selecting of selecting the $i \in \{1, 2, 3\}$ ranked individual is

$$\text{Prob}(i = 1) = \frac{1}{1 \cdot (\ln \mu + 1)} = O\left(\frac{1}{\log \mu}\right).$$

$$\text{Prob}(i = 2) = \frac{1}{2 \cdot (\ln \mu + 1)} = O\left(\frac{1}{\log \mu}\right).$$

$$\text{Prob}(i = 3) = \frac{1}{3 \cdot (\ln \mu + 1)} = O\left(\frac{1}{\log \mu}\right).$$

□

4.4 Inverse Quadratic scheme

LEMMA 4.6. *The expected runtime for finding all solutions on the Pareto front with SEMO or GSEMO with ranking-based inverse quadratic selection scheme on ONEMINMAX is bounded below by* $\Omega(1)$.

PROOF. As on the previous Lemma, we define the selection probability p as $6/(i^2 \cdot \pi^2)$. So the probability of selecting the i-th ranked individual with the inverse quadratic scheme is defined as

$$\text{Prob(select } i\text{-th ranked individual)} = \frac{6}{i^2 \pi^2}.$$

So the probability of selecting of selecting the $i \in \{1, 2, 3\}$ ranked individual is

$$\text{Prob}(i = 1) \geq \frac{6}{\pi^2} = \Omega(1).$$

$$\text{Prob}(i = 2) \geq \frac{3}{2\pi^2} = \Omega(1).$$

$$\text{Prob}(i = 3) \geq \frac{2}{3\pi^2} = \Omega(1).$$

□

4.5 Tournament selection

Tournament selection size μ

5 LOTZ PROBLEM

Definition 5.1. Define

$$L(x) = LO(x) + TZ(x),$$

which denotes the total number of leading ones and trailing zeros of a certain individual x.

5.1 SEMO with local search

Starting with a single initial solution, before reaching the Pareto front, SEMO keeps only one individual in the population.

LEMMA 5.2. *If a solution is not on the Pareto front, a 1-bit flip can only generate an offspring which either dominates the parent or is dominated by the parent.*

PROOF. In order to generate a non-dominating or non-dominated solution, both objectives need to be changed by the 1-bit flip. In LOTZ problem, the objective values is decided by position of the first 0-bit after the leading ones and the first 1-bit before the trailing zeros. However, through a 1-bit flip, the mutator can only change one of these 2 bits, which means it can only change one of the objectives. 1-bit flip leads to an offspring which has at least one objective that is the same as its parents, which implies the offspring either dominates the parent or is dominated by the parent.   □

The population size remains unchanged before there is a solution on the Pareto front. For a solution on the Pareto front, SEMO with local search only accepts its offspring which is also on the Pareto front since the possible offspring of such a solution from local search is dominated by it if not on the Pareto front.

LEMMA 5.3. *Assume the probability of selecting a solution on the Pareto front with a gap on either side in the population is at least p. The expected runtime for SEMO with local search to reach a population covering the whole Pareto front is bounded above by $O(n^2/p)$.*

PROOF. Starting from a popualtion with single individual, the population size remains 1 before achieving at least one solution on the Pareto front. Let $L_{max} = \max_{x \in P}(L(x))$. An individual x' with

$L(x') = n$ implies this individual is on the Pareto front. Then when $L_{max} = n$, there is at least one solution in the current population that is on the Pareto front. Flipping the first 0-bit or the last 1-bit of an individual x results in an offspring x' with $L(x') \geq L(x) + 1$. Therefore, the probability of increasing L_{max} when $L_{max} < n - 1$ is at least $2/n$. The expected runtime for obtaining a solution on the Pareto front is at most

$$\sum_{L_{max}=0}^{n-2} \frac{n}{2} = O(n^2).$$

When there exists at least one solution on the Pareto front, among all possible offspring from a solution on the Pareto front with a gap next to it, only the offspring that are also on the Pareto front will be accepted according to the algorithm. Let μ represent the current population size. Since there are only solutions on the Pareto front in the population once reaching the front, when $\mu = n + 1$, the population covers the whole Pareto front. The probability of increasing μ is at least $p \cdot \frac{1}{n}$. The expected runtime for covering the whole Pareto front is bounded above by

$$\sum_{\mu=1}^{n} \frac{n}{p} = O(n^2/p).$$

Hence, the overall expected runtime for SEMO with local search to achieve a population covering the whole Pareto front of Lotz is upper bounded by $O(n^2/p)$. □

If the parent selected to generate offspring is the one with $\max_{x \in P}(c(x))$, the probability p of selecting a solution on the Pareto front with a gap on either side equals to 1, since an individual with no gaps on both sides has $c(x) = 1$ which is the smallest value that $c(x)$ can get. Then the expected runtime for *SEMO* to reach a population covering the whole Pareto front of Lotz is bounded above by $O(n^2)$.

5.2 Global SEMO

For GSEMO, we consider the parent selection scheme that considers the hypervolume indicator together with the value of L.

Define an indicator r of individual $x \in P$ as

$$r(x) = (L(x), c(x)).$$

For two individuals $x, y \in P$, we define $r(x) < r(y)$ if $L(x) < L(y)$ or $L(x) = L(y) \wedge c(x) < c(y)$. The parent to be selected is a random individual from $\arg\max_{x \in P}(r(x))$.

LEMMA 5.4. *The expected runtime for GSEMO with this selection scheme to reach the Pareto front is bounded above by $O(n^2)$.*

PROOF. According to the definition, before reaching the Pareto front, the solution with $\max_{x \in P}(L(x))$ is selected to generate an offspring. Consider the event of only flipping the first 0-bit or the last 1-bit of the selected individual. Since the offspring from this event has higher value of one of the objectives than its parent which is of the maximum $L(x)$ in the population, the offspring is non-dominated by any individuals in the population and is accepted by the algorithm. Hence, the probability of increasing $\max_{x \in P}(L(x))$ is at least

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

Throughout the process, the value of $\max_{x \in P}(L(x))$ in the population never goes down.

Therefore, the overall expected runtime for GSEMO with this selection scheme to reach the Pareto front is at most

$$\sum_{L_{max}=0}^{n-2} \frac{en}{2} = O(n^2).$$

□

LEMMA 5.5. *After there is at least one solution on the Pareto front, the parent to be selected is always on the Pareto front and has a gap on either side.*

PROOF. According to the selection scheme, the parent to be selected has the maximum $L(x)$ value in the population. Since there is at least one solution on the Pareto front, the parent is selected from the individuals on the Pareto front which have the maximum $L(x) = n$.

If an individual on the Pareto front does not have gaps on either side, its contribution to the hypervolume indicator $c(x)$ equals to 1 which is the minimum value. Before the whole Pareto front is covered, there should exist at least a gap on the front. The individuals next to the gap have greater contribution to the hypervolume indicator than 1 from which a parent will be selected to generate the offspring. □

LEMMA 5.6. *The expected runtime of GSEMO with this selection scheme achieving a population covering the whole Pareto front of Lotz is bounded above by $O(n^2)$.*

PROOF. Before the population covers the whole Pareto front, the optimization process of GSEMO can be divided into two stages. The focuses of the two stages are obtaining one individual on the Pareto front and covering the Pareto front.

As proved in Lemma 5.4, the expected runtime for GSEMO with the certain selection scheme to reach the Pareto front is at most $O(n^2)$.

In the second stage, the parent to be selected is on the Pareto front and has a gap on either side. Consider the event of generating an offspring in the gap. Since the offspring is on the Pareto front and different from the individuals existing in the population, it is accepted by the algorithm. The probability of such an event to happen is at least

$$p \cdot \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{p}{en},$$

where p denotes the probablity of selecting the individual on the Pareto front and with a gap on either of its two sides. According to the selection scheme, the parent to be selected is in $\arg\max_{x \in P}(r(x))$. Since there is at least one individual on the Pareto front, the solution with $\max_{x \in P}(r(x))$ should have $L(x) = n$ and gap on either side, which implies $p = 1$.

There are $(n + 1)$ distinct points on the Pareto front of Lotz problem. In order to cover the whole Pareto front, at least one individual corresponding to each points on the Pareto front should be included in the population. Since the individuals on the Pareto front are not dominated by other individuals, they will not be removed from the population, which indicates that the number of

covered points on the Pareto front never decrease. Therefore, the expected runtime of the second stage is at most

$$\sum_{\mu=1}^{n} en = O(n^2).$$

To sum up, the overall runtime for GSEMO with the selection scheme choosing parent solution from $\arg\max_{x \in P}(r(x))$ on Lotz to reach a population covering the whole Pareto front is bounded above by $O(n^2)$.   □

6 EXPERIMENTS

Aside of the theoretical results, we rely on experiments to test our theoretical affirmations. The experimental approach is focused on the analysis of the Algorithms 2 and 3 and its performance with and without the diversity-based parent selection mechanisms. We are interested in observing if we can speed up in the performance from the classical approaches.

This also allows a more detailed comparison of the hypevolume contribution (HVC), the crowding distance contribution (CDC), and the parent selection methods. In the case of the HVC, we have defined two reference points, $HVC(-1, -1)$ and $HVC(-n, -n)$. For the first reference point, we have provide a slightly preference to the extreme points while with the second reference point, 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 assign infinity value to the current 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 requires to found the Pareto front, we have defined that outcome as a stopping criterion, and we repeat the experimental framework for 100 runs with $n = 100$ for all algorithmic approaches and report the mean as our metric 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 of generations required to found the Pareto front for the classic SEMO and GSEMO that use uniform random selection for both test functions. The second section (lower part), refers to the mean of generations required to found 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 point can be potentially dangerous, and to exemplify this, we have introduced a deterministic selection mechanism which we have named High Diversity Contribution (HDC), select the individual with the highest diversity contribution.

As can be seen in Table 1, HDC fails to found the Pareto front for OneMinMax and Lotz in the case of GSEMO. Due to the mutation mechanism, once we start expanding the Pareto front, the algorithm may create an offspring far from the parent, leaving unexplored areas (or gaps) between them, and since the parent selection is only focused on the current extreme individuals, it will expand the Pareto front until it reaches the most extreme individuals, and it

will continue selecting those individuals ignoring the intermediate individuals, leaving the population in a stagnation state.

Finally, for any other parent selection defined in this paper, we have achieved an speeding up in the performance of SEMO and GSEMO. 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., significantly less generations are required for both test functions.

7 CONCLUSIONS

Future work: analyse Minimum Spanning Trees, SetCover, VertexCover

REFERENCES

[1] Karl Bringmann and Tobias Friedrich. 2010. An Efficient Algorithm for Computing Hypervolume Contributions. *Evol. Comput.* 18, 3 (Sept. 2010), 383–402. DOI: http://dx.doi.org/10.1162/EVCO_a_00012

[2] 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. DOI: http://dx.doi.org/10.1016/j.tcs.2010.09.026

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

[4] 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. DOI: http://dx.doi.org/10.1109/4235.996017

[5] Tobias Friedrich, Christian Horoba, and Frank Neumann. 2011. Illustration of fairness in evolutionary multi-objective optimization. *Theor. Comput. Sci.* 412, 17 (2011), 1546–1556. DOI: http://dx.doi.org/10.1016/j.tcs.2010.09.023

[6] Oliver Giel. 2003. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, Vol. 3. 1918–1925 Vol.3. DOI: http://dx.doi.org/10.1109/CEC.2003.1299908

[7] Oliver Giel and Per Kristian Lehre. 2010. On the Effect of Populations in Evolutionary Multi-Objective Optimisation. *Evolutionary Computation* 18, 3 (2010), 335–356. DOI: http://dx.doi.org/10.1162/EVCO_a_00013

[8] 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*, Carlos A. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan (Eds.). Studies in Computational Intelligence, Vol. 272. Springer, 23–44. DOI: http://dx.doi.org/10.1007/978-3-642-11218-8_2

[9] 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. DOI: http://dx.doi.org/10.1109/TEVC.2004.823470

[10] 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. DOI: http://dx.doi.org/10.1016/j.tcs.2014.06.023

[11] 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. DOI: http://dx.doi.org/10.1109/ICTAI.2011.47

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

	OneMinMax			Lotz		
SEMO	4.16E+04			3.17E+05		
GSEMO	1.06E+05			6.58E+05		
Selection Mechanism	$HVC(-1,-1)$	$HVC(-n,-n)$	CDC	$HVC(-1,-1)$	$HVC(-n,-n)$	CDC
HDC	9.14E+02	8.90E+02	1.05E+03	1.24E+04	1.25E+04	1.41E+04
	2.12E+03	0	0	3.06E+04	0	0
NMUAR	8.92E+02	1.05E+03	1.03E+03	1.25E+04	1.38E+04	1.41E+04
	2.14E+03	2.54E+03	2.58E+03	3.17E+04	3.50E+04	3.58E+04
Exponential	1.28E+03	1.27E+03	1.36E+03	1.57E+04	1.58E+04	1.78E+04
	3.21E+03	3.18E+03	3.24E+03	3.45E+04	4.00E+04	5.87E+04
Harmonic	3.05E+03	3.24E+03	3.28E+03	3.14E+04	3.08E+04	3.53E+04
	7.89E+03	7.26E+03	8.03E+03	6.69E+04	6.33E+04	6.73E+04
Inverse Quadratic	1.15E+03	1.24E+03	1.34E+03	1.54E+04	1.51E+04	1.69E+04
	2.87E+03	2.85E+03	3.32E+03	3.40E+04	5.03E+04	5.73E+04
Tournament Selection (μ)	1.05E+03	1.08E+03	1.21E+03	1.38E+04	1.41E+04	1.55E+04
	2.58E+03	2.60E+03	2.81E+03	3.16E+04	6.53E+04	7.87E+04