Runtime Analysis for Maximizing Population Diversity in Single-Objective Optimization

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

ABSTRACT

Recently Ulrich and Thiele [14] have introduced evolutionary algorithms for the mixed multi-objective problem of maximizing fitness as well as diversity in the decision space. Such an approach allows to generate a diverse set of solutions which are all of good quality. With this paper, we contribute to the theoretical understanding of evolutionary algorithms for maximizing the diversity in a population that contains several solutions of high quality. We study how evolutionary algorithms maximize the diversity of a population where each individual has to have fitness beyond a given threshold value. We present a first runtime analysis in this area and study the classical problems called *OneMax* and *LeadingOnes*. Our results give first rigorous insights on how evolutionary algorithms can be used to produce a maximal diverse set of solutions in which all solutions have quality above a certain threshold value.

1. INTRODUCTION

Evolutionary algorithms have been widely used for complex optimization problems. Most evolutionary algorithms incorporate certain diversity mechanisms which ensure that the population consists of a diverse set of individuals [3, 15]. From an optimization point of view this is often beneficial to prevent premature convergence to a locally optimal solution [9]. From a design point of view, this is interesting as a diverse set of solutions gives different design choices of high quality. This is especially important as practitioners in the areas of engineering and manufacturing may have a clear preference for certain solutions even if they have similar quality according to the used fitness functions which evaluate the quality of a given solution.

With this paper, we contribute to the theoretical understanding of diversity mechanisms used in evolutionary algorithms. We study evolutionary algorithms in a rigorous way using runtime analysis [1, 8, 10]. Previous studies in the field of runtime analysis in the context of diversity have examined how different diversity mechanisms influence the ability of an algorithm to obtain an optimal solution [5, 6].

In this paper, we consider diversity from a different perspective. We are interested in how evolutionary algorithms can achieve a diverse set of solutions that all have acceptable quality. Evolutionary algorithms for the problem of maximizing the diversity of a set of solutions where each of these solution has fitness above a given threshold value v have been introduced by Ulrich and Thiele [14]. Their algorithm called NOAH iteratively improves solutions according to quality and diversity of the population. Furthermore, decision space diversity has been examined for hypervolume-based search in the context of multi-objective optimization [13].

For our theoretical investigations, we work with a fixed threshold v. Our goal is to study population-based algorithms until they have obtained a population of maximum diversity where all solutions $x \in P$ have fitness at least v. The subject of our investigations is a classical $(\mu + 1)$-EA that starts maximizing the diversity of the population after all the solutions have reached fitness v. The plain version of the $(\mu + 1)$-EA has already been studied by Witt [16] for classical problems such as *OneMax* and *LeadingOnes*. We will study these problems in the context of diversity optimization and show in a rigorous way how evolutionary algorithms are able to maximize the diversity of a population for *OneMax* and *LeadingOnes*.

The paper is organized as follows. In Section 2, we introduce the algorithm that is subject of our investigations when considering diversity maximization. Our analysis for the classical *OneMax* problem is presented in Section 3 and Section 4 shows our results for the *LeadingOnes* problem. Finally, we finish with some concluding remarks to topics for future work.

2. DIVERSITY MAXIMIZATION

In this section we introduce the basic ideas of the diversity optimization of a simple single-objective problem. We are interested in pseudo-Boolean functions $f: X \to \mathbb{R}$ that map elements of the search space $X = \{0, 1\}^n$ to real values.

There are many ways to measure the difference between different individuals. Since pseudo-Boolean functions are defined on bit-strings, we use Hamming distance

$$H(x, y) = \sum_{i=1}^{n} |x_i - y_i|,$$

where $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \{0, 1\}^n$, to evaluate the difference between two individuals.

According to [12, 13], a diversity measurement should have the following properties:

Algorithm 1: (μ+1)-EA$_\mathrm{D}$

1 Initialize P with μ n-bit binary strings which are chosen uniformly at random from $\{0,1\}^n$.
2 Choose $s \in P$ uniformly at random.
3 Produce s' by flipping each bit of s with probability $1/n$ independently from each other. Add $s'$to P.
4 Let $z \in P$ be a randomly chosen individual with the worst fitness value.
5 If $f(z) \geq v$, then execute OptDiv(P), otherwise eliminate z from P.
6 Repeat step 2 to 5 until termination criterion is reached.

1. Twinning: Duplicate solutions in a population should not change the diversity.
2. Monotonicity in Varieties: Adding a new solution which is not in a population should increase the set diversity.
3. Monotonicity in Distance: $D(P') \geq D(P)$ with $|P| = |P'|$ holds, if all pairs of P' are at least as dissimilar as all pairs of P (according to some given distance function).

To fulfil the required features, the diversity of a set of solutions P is defined as the sum of Hamming distance between each pair of individuals in P. Note that in general P can be a multi-set which may include duplicates. In order to meet the twinning property, duplicates are removed when computing the diversity of a (multi-)set P based on the Hamming distance.

DEFINITION 1. *For a given population P, the population diversity is defined as $D(P) = \sum_{\{x,y\} \in \hat{P} \times \hat{P}} H(x,y)$, where $\hat{P}$ is the set with all distinct solutions in P .*

Since our aim is to find a set of solutions of different structures, we combine the classical (μ+1)-EA with diversity optimization process. The threshold v of fitness value is predefined by the decision maker. The (μ+1)-EA with solution diversity optimization is defined as (μ+1)-EA$_\mathrm{D}$. The whole process of (μ+1)-EA$_\mathrm{D}$ is given in Algorithm 1.

The diversity optimization is conducted until all individuals in the solution set reach the fitness requirement. Moreover, once entering the diversity optimization process, the algorithm will reject the offspring with fitness below threshold. The contribution of solution x is defined as

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

If an offspring of acceptable quality is produced, the individual with least contribution of diversity is eliminated from the solution set. If this solution is not unique, a solution is chosen uniformly at random among the solutions with the smallest diversity contribution. Algorithm 2 defines the *OptDiv(P)* component.

We study our algorithm in terms of the number of fitness evaluations until it has produced a population P with $f(x) \geq v, \forall x \in P$ that has the maximal diversity $D(P)$. We call this the *optimization time* of the algorithm. The *expected optimization time* refers to the expected number of fitness evaluations to reach this goal.

We first analyze the time until all individuals have fitness of at least v after having achieved such an individual for the first time. The process is similar to the take-over effect in a population and we show an upper bound of $O(\mu \log \mu)$ for a population of size μ in the following lemma. It will serve later on throughout our analysis.

Algorithm 2: Diversity optimization component OptDiv(P)

1 Choose a solution $z \in \{x \in P \mid c(x) = \min_{y \in P} c(y)\}$ uniformly at random.
2 Set $P := P \setminus \{z\}$.

LEMMA 1. *Having obtained a population with at least one individual of fitness at least v, the expected runtime until all individuals have fitness at least v is upper bounded by $O(\mu \log \mu)$.*

PROOF. Since there is already one individual which has fitness value at least v, then one possible method is making duplicates of the best solution until all μ solutions are replaced by the replicas. The probability of making a duplicate of the acceptable solution when there already exists i individuals with fitness value above the threshold in the population is

$$\frac{i}{\mu} \cdot \left(1 - \frac{1}{n}\right)^n = \frac{i}{\mu} \cdot \frac{n-1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geqslant \frac{i(n-1)}{e\mu n} \geqslant \frac{i}{2e\mu}.$$

Before entering the diversity optimization process, we need all of the μ individuals in the population set to have acceptable fitness value. The expected waiting time for this process is at most

$$\sum_{i=1}^{\mu-1} \frac{2e\mu}{i} = 2e\mu \sum_{i=1}^{\mu-1} \frac{1}{i} = O(\mu \log \mu).$$

□

3. OneMax

In this section, we investigate the classical *OneMax* problem which has been subject to numerous studies in the area of runtime analysis of evolutionary algorithms [4, 16]. Our goal is to understand how simple evolutionary algorithms can maximize the diversity of its population for this simple benchmark problem. The problem is defined as

$$OneMax(x) = \sum_{i=1}^{n} x_i.$$

We first analyze until one solution has fitness at least v. To do this, we follow the ideas of Witt [16] for the analysis of the classical $(\mu+1)$-EA.

Let v be the threshold of the fitness value, hence, the acceptable solution should have at least v 1-bits. The diversity optimization process will not begin until all of the solutions in the population have the fitness above the threshold. We denote by $L = \max_{x \in P} OneMax(x)$ the maximal fitness value of the current population and upper bound the time to achieve for the first time a solution of fitness at least v.

LEMMA 2. *The expected time until $(\mu+1)-EA$ has obtained a solution x with $OneMax(x) \geq v$ is $O(\mu v + n \log \frac{n}{n-v})$.*

PROOF. For a certain L value, duplicates will be made from the individuals with fitness value L before L improves. Following Witt [16], we assume that L remains the same before there are $\min\{\frac{n}{n-L}, \mu\}$ duplicates of the individual with fitness L.The expected time for the population to have at least $\frac{n}{n-L}$ duplicates of one of these individuals with fitness value L is at most

$$\sum_{i=1}^{\min\{n/(n-L),\mu\}} \frac{e\mu n}{i(n-1)} = \frac{e\mu n}{n-1} \sum_{i=1}^{\min\{n/(n-L),\mu\}} \frac{1}{i} \leqslant \frac{e\mu n}{n-1} \ln \frac{en}{n-L}$$

For a population set which has i individuals with fitness value L, improvement can be made by selecting one of these individuals and flipping one of its 0-bits. The considered probability is

$$\frac{i}{\mu} \cdot \frac{(n-L)}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geqslant \frac{i(n-L)}{e\mu n}$$

Therefore the expected time for the fitness value to increase is at most $\frac{e\mu n}{i(n-L)}$.

The waiting time of the (μ+1)-EA achieving the first satisfactory solution equals to the sum of expected time for each L value which includes the time for increasing L and time for duplicating individuals. The expected waiting time for the (μ+1)-EA getting the first individual with fitness v is at most

$$\sum_{L=0}^{v-1} \frac{e\mu n}{\min\{\mu, n/(n-L)\} \cdot (n-L)} + \frac{e\mu n}{n-1} \sum_{L=0}^{v-1} \ln \frac{en}{n-L}.$$

According to the Harmonic sum,

$$\begin{aligned}\sum_{L=0}^{v-1} \frac{e\mu n}{\min\{\mu(n-L), n\}} &\leqslant \sum_{L=0}^{v-1} \frac{en}{n-L} + \sum_{L=0}^{v-1} e\mu \\ &\leqslant en(\ln en - \ln(n-v)) + e\mu v \\ &= en \ln \left(\frac{en}{n-v}\right) + e\mu v\end{aligned}$$

$$\begin{aligned}\frac{e\mu n}{n-1} \sum_{L=0}^{v} \ln \frac{en}{n-L} &= \frac{e\mu n}{n-1} \ln \frac{e^v n^v}{n(n-1)(n-2)\cdots(n-v)} \\ &= \frac{e\mu n}{n-1} \ln \frac{e^v n^v (n-v-1)!}{n!}\end{aligned}$$

As stated in Stirling's Formula, $e^v n^v < \frac{e^{2v} n^v v!}{v^v \sqrt{2\pi v}}$. Hence,

$$\ln \frac{e^v n^v (n-v-1)!}{n!} < \ln\left(\frac{e^{2v} n^v}{v^v \sqrt{2\pi v} \cdot (n-v)} \cdot \frac{v!(n-v)!}{n!}\right).$$

The Binomial coefficients $\binom{n}{k}$ has the property that

$$\left(\frac{n}{k}\right)^k \leqslant \binom{n}{k}.$$

Hence, we get,

$$\begin{aligned}\frac{e\mu n}{n-1} \sum_{L=0}^{v-1} \ln \frac{en}{n-L} &< \frac{e\mu n}{n-1} \ln\left(\frac{e^{2v} n^v}{v^v \sqrt{2\pi v} \cdot (n-v)} \cdot \frac{v!(n-v)!}{n!}\right) \\ &< \frac{e\mu n}{n-1} \ln\left(\frac{e^{2v} n^v}{v^v \sqrt{2\pi v} \cdot (n-v)} \cdot \left(\frac{v}{n}\right)^v\right) \\ &= \frac{e\mu n}{n-1} \ln \frac{e^{2v}}{\sqrt{2\pi v} \cdot (n-v)} \\ &< \frac{2e\mu n v}{n-1}\end{aligned}$$

Thus, the expected waiting time of (μ+1)EA with threshold v is $O(n \log \frac{n}{n-v} + \mu v)$. □

Due to Lemma 1, we already know that after an additional phase of $O(\mu \log \mu)$ all individuals in the population have fitness at least v.

3.1 Large threshold

Firstly, we begin with a simple case where the threshold $v = n-1$. There are $(n+1)$ possible solutions which have fitness value above the threshold. The composition of optimal solution set depends on the population size μ.

THEOREM 1. *Let $v = n-1$ and $\mu \geqslant n+1$, then the expected optimization time of (μ+1)-EA$_D$ on* OneMax *is upper bounded by $O(\mu n + \mu \log \mu + n^2 \log n)$.*

PROOF. There are $(n+1)$ different individuals that have fitness value above the threshold. When $\mu \geqslant n+1$, the optimal solution set should contain all of the $(n+1)$ different individuals. According to our definition of diversity, duplicates will not affect the diversity. Then the $(\mu - n - 1)$ other individuals have no contribution to the diversity.

As stated in Lemma 2, the expected waiting time until (μ+1)-EA has obtained a solution with fitness value above the threshold when $v = n-1$ is bounded above by $O(\mu n + n \log n)$.

After the first solution with fitness value above the threshold is produced, the algorithm will focus on producing other individuals with acceptable quality. According to Lemma 1, the expected runtime of this procedure is bounded above by $O(\mu \log \mu)$.

We now work under the assumption that all individuals have fitness at least v. Note, that (μ+1)-EA$_D$ will not accept any solution of fitness below v. In the worst case, these μ solutions are replicas, so the population diversity equals to 0 at the beginning. The diversity can be improved by producing new solutions from the replicas. Since the duplicates in the population have no contribution to the diversity, they will be replaced by the new individual which has a higher contribution to the diversity. It does not matter which individual is selected from the population to produce a new solution, since the individual with the least contribution will always be the one to be replaced. If the current population has i different individuals, the probability of creating a new solution with fitness value v is at least

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

The 1^n solution can be produced in any stage and will stay in the population. The probability of producing the 1^n solution is

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

Since the duplicated individuals in the population will not affect the diversity measurement, the diversity optimization process will not stop until the optimal diversity is reached, which means the population set contains all possible solutions that fulfil the requirement in fitness value.

The expected time for optimizing the diversity is

$$\begin{aligned} en + \sum_{i=1}^{n-1} \frac{en(n-1)}{n-i} &= en + en(n-1)\sum_{i=1}^{n-1}\frac{1}{i} \\ &\leqslant en + en(n-1)\ln(en) \end{aligned}$$

Hence, the expected optimization of the (μ+1)-EA on *OneMax* with diversity optimization for threshold $(n-1)$ is bounded above by

$$\begin{aligned} & O(\mu n + n\log n) + O(\mu\log\mu) + O(n + n^2\log n) \\ = \ & O(\mu n + \mu\log\mu + n^2\log n). \end{aligned}$$

□

We now study smaller population sizes such that not all different solutions of fitness at least v can be included in the population. Here the (μ+1)-EA$_\text{D}$ has to obtain of subset of μ solutions maximizing the diversity.

THEOREM 2. *Let $v = n-1$ and $\mu < n+1$, then the expected optimization time of (μ+1)-EA$_\text{D}$ on* OneMax *is upper bounded by $O(\mu n\log(\frac{n}{n-\mu}) + n\log n)$.*

PROOF. When $\mu < n+1$, the population set can not contains all possible solutions with fitness value above the threshold. Since the all 1-bit solution only has 1 bit different to other acceptable individuals which have 2 bits different to each other, it will not be in the optimal solution set. Moreover, every individual with fitness $n-1$ has the same Hamming distance to each other, therefore, it does not matter which individual is contained in the population set.

The proof for expected time of (μ+1)-EA achieving the population set with duplicates of an individual with fitness value above the threshold is the same as that in Theorem 1. The expected time is at most $O(\mu n + n\log n + \mu\log\mu)$.

It is of great possibility that the solution with all 1-bits is introduced in some stage of the diversity optimization process. When the population size is small, the probability to select the all 1-bit solution to produce the new solution is large. Since all the individuals in the population have reached the threshold, the probability of getting the 1^n solution is

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

Then the expected time taken to produce the 1^n solution is less than $en = O(n)$.

After the 1^n solution is introduced, it will remain in the population until the other individuals all have different patterns. The probability of getting a new solution by flipping one 1-bit of the 1^n individual when there are i different solutions with fitness value above the threshold is

$$\frac{1}{\mu}\cdot\frac{n-i}{n}\cdot\left(1-\frac{1}{n}\right)^{n-1} \geqslant \frac{n-i}{e\mu n}.$$

For the $\mu < n+1$ situation, all of the individuals in the expected solution population should be of different structures. Since the contribution of 1^n in diversity is smaller comparing to those of the individuals with fitness v, the 1^n solution will be replaced by a solution with fitness value v after the other $(\mu-1)$ individuals are different from each other.

Then the waiting time for achieving a population of μ different solutions with fitness $(n-1)$ from the intermediate step is

$$\sum_{i=1}^{\mu-1}\frac{e\mu n}{n-i} = e\mu n\sum_{i=1}^{\mu-1}\frac{1}{n-i} \leqslant e\mu n(\ln n - \ln(n-\mu))$$

Having obtained μ individuals of fitness $v = n-1$, the solution 1^n is removed from the population as it has the smallest diversity contribution, and then the optimal population is achieved.

Summing up, the expected optimization time is

$$\begin{aligned} & O(\mu n + n\log n + \mu\log\mu) + O(n) + O(\mu n\log(\frac{n}{n-\mu})) \\ = \ & O(n\log n + \mu n\log\frac{n}{n-\mu}). \end{aligned}$$

□

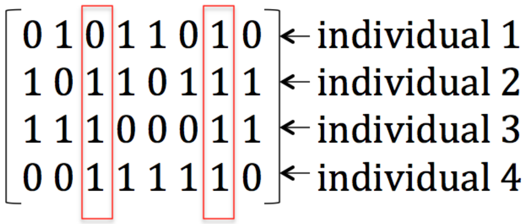

Figure 1: The $\mu \times n$ matrix represents the individuals in a population. In the example, it is a matrix for a population with 4 individuals which are all 8 bits in length. The 7th column is all-1-bit column and the 3rd column is 0-bit column as defined.

3.2 Smaller thresholds

We now consider the case where $n/2 \leq v < n-1$ holds. For convenience, we store the population in a $\mu \times n$ matrix where each individual as a row and define the column where there is no 0-bit as all-1-bit column and the column where there is only one 0-bit as 0-bit column. An example is shown in Figure 1.

The following lemma shows crucial properties of a population maximizing diversity.

LEMMA 3. *Let $\mu \leqslant \binom{n}{v}$. The matrix of a population P represents an optimal population, if the whole matrix contains $\mu(n-v)$ 0-bits and each column contains $\mu(n-v)/n$ 0-bits.*

PROOF. There are $\binom{n}{v}$ possible solutions for the *OneMax* problem with threshold v. Assuming $v \geq n/2$ implies that there have to be at least as many 1-bits as 0-bits in each individual. For $\mu \leqslant \binom{n}{v}$, we show that the optimal population should contains only individuals with fitness value v, since these individuals can make a higher contribution to the overall diversity. Then the total number of 0-bits in the population can be represented as $\mu\cdot(n-v)$, and w.l.o.g , we assume that $\mu\cdot(n-v)/n$ is an integer.

From the perspective of matrix representing the optimal population, the contribution of each column has no influence on those of other columns, so the population diversity equals to the sum of contribution of every column in the matrix. The contribution of each column should be maximized so that the population diversity is maximized. If there are m 0-bits in a column, the contribution of this column will be

$$m(\mu - m).$$

The population diversity can be calculated as

$$\sum_{i=1}^{n} m_i(\mu - m_i),$$

where m_i represents the number of 0-bits in the ith column. The constraint is that the total number of 0-bits in the population is at most $\mu(n-v)$, which can be represented as

$$\sum_{i=1}^{n} m_i \leqslant \mu(n-v).$$

When $\mu \leqslant \binom{n}{v}$, the constraint is

$$\sum_{i=1}^{n} m_i = \mu(n-v).$$

Before all columns are balanced in the number of 0-bits, there exist at least two columns that one has more 0-bits than average number and the other has less 0-bits than average number. Let i, j, k represent the number of 0-bits in columns which has 0-bits above, below and equal to the average number separately. Their relationship can be interpreted as $j < k < i$, where $i, j, k \in \mathbb{N}$. Reducing the unbalance rate by flipping a 1-bit and a 0-bit of column with i and j. Increasing j by 1 causes the diversity changed by

$$(j+1)(\mu - j - 1) - j(\mu - j) = \mu - 2j - 1.$$

Decreasing i by 1 causes the diversity changed by

$$(i-1)(\mu - i + 1) - i(\mu - i) = -\mu - 2i - 1.$$

Therefore, the overall change to diversity is

$$(\mu - 2j - 1) + (-\mu - 2i - 1) = 2(i - j - 1).$$

Since i, j and k are all natural numbers and none of them are equal as defined, $2(i-j-1)$ should be at least 2. Hence, whenever there is unbalance in the number of 0-bits in each column, there exist some columns which can be changed to gain balance and increase diversity, which implies that the population diversity is optimized only when the 0-bits are evenly distributed in each column. The number of each column is then $\mu \cdot (n-v)/n$. $\square$

We now consider the case of a small population where $\mu \leqslant n/(n-v)$ holds. In this case the optimal population contains only individuals which have 0-bits in different positions.

THEOREM 3. *Let $\mu \leqslant \frac{n}{n-v}$, then the expected optimization time of (μ+1)-EA$_D$ on OneMax is upper bounded by $O(\mu n^2 \log \mu)$.*

PROOF. According to Lemma 1 and 2, it takes $O(\mu v + n \log \frac{n}{n-v})$ time to achieve a population with all individuals above the threshold.

Since the population size is $\mu \leqslant \frac{n}{n-v}$, the population set with optimal diversity value should contains only individuals which have 0-bits in different position with other individuals. The matrix for the population with optimal diversity value should only have all-1-bit columns and 0-bit columns.

In the worst case, there are $\mu(n-v)$ 0-bits that has duplicates in the same column. In order to achieve the optimal population, the number of columns with more than one 0-bits should be decreased to 0.

At the beginning of the diversity optimization process, the population diversity is 0 as in the worst case where there are only duplicates. The number of all-1-bit columns is v. Before the population diversity reaches the optimal value, there should exist at least one column that has more than one 0-bits. Hence, one way of improving the diversity is selecting an individual with 0-bit not in the 0-bit column and increasing its contribution to diversity. Let the number of 0-bits in ith column be represented by m_i. Then flipping one 0-bit of an individual will cause the contribution change by

$$(m_i - 1)(\mu - m_i + 1) - m_i(\mu - m_i) = -\mu + 2m_i - 1.$$

Flipping one 1-bit in the all-1-bit column will increase the contribution by $(\mu - 1)$. Therefore, flipping a pair of 1-bit and 0-bit as restricted above will change its contribution by

$$(-\mu + 2m_i - 1) + (\mu - 1) = 2(m_i - 1).$$

In order to increase the diversity, the 0-bit chosen should fulfil the condition of $m_i > 1$, which means the 0-bit to be flipped should have duplicates in the same column.

Before the diversity is optimized, there should always exist a column which has more than one 0-bit. We consider the event that selecting an individual which has an 0-bit in the column with $m_i > 1$ and flipping the certain 0-bit together with a 1-bit in one of its all-1-bit columns. According to our analysis above, this event will produce an individual that increases the diversity by at least $2(m_i - 1)$. Let the number of 0-bits with duplicates in the same column be represented by k. Then the probability for such an event described above to happen equals to

$$\frac{k}{\mu} \cdot \frac{1}{n} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-2} \geqslant \frac{k}{e\mu n^2}.$$

The event described in the last paragraph decreases the number k by 1. When there are 2 0-bits in a column, these 2 duplicated 0-bits can be split into two 0-bit columns in one iteration. Hence, it takes $\mu(n-v) - (n-v)$ steps to get the optimal population.

Therefore the overall waiting time is

$$\begin{aligned}\sum_{k=\mu(n-v)}^{(n-v)} \frac{e\mu n^2}{k} &= e\mu n^2 \sum_{k=\mu(n-v)}^{n-v} \frac{1}{k} \\ &\leqslant e\mu n^2 (\ln e\mu(n-v) - \ln(n-v)) \\ &\leqslant e\mu n^2 \ln e\mu\end{aligned}$$

Hence, the expected optimization time is

$$O(\mu v + n \log \frac{n}{n-v}) + O(\mu n^2 \log \mu) = O(\mu n^2 \log \mu).$$

$\square$

In the following, we study how (μ+1)-EA$_D$ is able to achieve an optimal population if μ is larger.

THEOREM 4. *Let $\mu \leqslant \binom{n}{v}$, then the expected optimization time of (μ+1)-EA$_D$ on OneMax is upper bounded by $O(\mu n^2 (n-v) \log(\frac{n}{n-v}))$.*

PROOF. According to Lemma 2, the waiting time of (μ+1)-EA on *OneMax* to achieve the first individual with fitness value above the threshold v is $O(\mu v + n \log \frac{n}{n-v})$. After that, it takes $O(\mu \log \mu)$ time to obtain a solution set that contains only individuals with fitness value above the threshold as proved in the previous section.

As proved in Lemma 3, the 0-bits should be equally distributed in all columns so that the population diversity is optimized. The total number of 0-bits in the population is $\mu(n-v)$. The optimal solution set should make sure that in the corresponding matrix there are either

$$\left\lceil \frac{\mu(n-v)}{n} \right\rceil \text{ or } \left\lfloor \frac{\mu(n-v)}{n} \right\rfloor$$

0-bits in each column. Let i represent the largest number of 0-bits in a column and j represent the smallest number of 0-bits in a column. If $j \leqslant i \leqslant j+1$ is fulfilled, the population diversity is optimized. When $\mu(n-v)\%n = 0$, then the equality stands.

Before the diversity is maximized, it should be true that $i > j$. Therefore, there exists an individual that has a 0-bit in the column with i 0-bits and a 1-bit in the column with j 0-bits. Consider the event that selecting the certain individual and flipping those 0-bit and 1-bit in the same iteration. The overall change to contribution is

$$(-\mu + 2i - 1) + (\mu - 2j - 1) = 2(i - j - 1),$$

which is the same as that in Lemma 3. Since $i > j + 1$ before the diversity is optimized, such an event should lead to improvement in the diversity. The probability for such an event happen is

$$\frac{1}{\mu} \cdot \frac{1}{n} \cdot \frac{1}{n} \cdot (1 - \frac{1}{n})^{n-2} \geqslant \frac{1}{e\mu n(n-1)}.$$

Hence, the expected runtime for the improvement is bounded above by $O(\mu n^2)$.

In the beginning of diversity optimization process, as the worst case, the row matrix storing the numbers of 0-bits of each column contains $(n-v)$ elements with value μ and v elements with value 0. The event mentioned above decreases the number of 0-bits in the certain column and increases the number of 0-bits in the corresponding column. As proved in Lemma 3, when $v \geqslant \mu/2$, the contribution to diversity of a column is monotonously increasing with the number of 0-bits.

In the worst case, there are $(n-v)$ columns which have μ 0-bits at the beginning. Then it takes $(n-v)$ iterations to decrease the largest number of 0-bits in a column ,which is defined as i, by 1. For a certain i, there are at least i individuals that can be chosen to produce a child by flipping a pair of 0-bit and 1-bit as stated in the previous paragraph.

Therefore the overall waiting time of (μ+1)-EA$_\text{D}$ on *OneMax* for a population with less than $\binom{n}{v}$ individuals is at most

$$\begin{aligned} &\sum_{i=\mu}^{\frac{\mu(n-v)}{n}+1} (n-v) \cdot en(n-1) \cdot \frac{\mu}{i} \\ \leqslant\ & e\mu n^2(n-v) \sum_{i=\mu}^{\frac{\mu(n-v)}{n}+1} \frac{1}{i} \\ \leqslant\ & e\mu n^2(n-v) \ln(\frac{en}{n-v}) \end{aligned}$$

Hence, the expected optimization time is bounded above by

$$O(\mu n^2(n-v) \log \frac{n}{n-v}).$$

□

4. LeadingOnes

In this section we will discuss the expected runtime for (μ+1)-EA$_\text{D}$ on the classical *LeadingOnes* problem which has been subject to several investigations in the area of runtime analysis [2, 16]. *LeadingOnes* is defined as

$$LeadingOnes(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j.$$

Similar to that of *OneMax* problem, the diversity optimization on *LeadingOnes* problem can also be divided into two stages. The first one is obtaining a population of all individuals with acceptable fitness value and the second one is maximizing the population diversity.

LEMMA 4. *The expected runtime until* $(\mu{+}1)-EA$ *on* LeadingOnes *problem has obtained a solution of fitness value above the threshold* v *is* $O(nv + \mu v \log n)$.

PROOF. Assume L represents the largest number of leading ones among all individuals in the current population and i represents the number of individuals with the fitness value L. For a certain L value, we assume it will not change until there are $\min\{n/\ln(en), \mu\}$ duplicates of the individual with fitness L as stated in Witt [16].

The probability for making a duplicate of the individual with fitness L is at least

$$\frac{i}{\mu} \cdot (1 - \frac{1}{n})^n \geqslant \frac{i(n-1)}{e\mu n}.$$

The expected runtime for making $\min\{n/\ln(en), \mu\}$ duplicates is at most

$$\begin{aligned} \sum_{i=1}^{\min\{n/\ln(en),\mu\}-1} \frac{e\mu n}{i(n-1)} &= \frac{e\mu n}{n-1} \sum_{i=1}^{\min\{n/\ln(en),\mu\}-1} \frac{1}{i} \\ &\leqslant \frac{e\mu n}{n-1} \ln \frac{en}{\ln(en)} \\ &\leqslant 2e\mu \ln(en) \end{aligned}$$

After there exist at least $\min\{n/\ln(en), \mu\}$ duplicates, L will be improved by selecting an individual with fitness L and flipping its leftmost 0-bit. The probability for this event is

$$\frac{i}{\mu} \cdot \frac{1}{n} \cdot (1 - \frac{1}{n})^{n-1} \geqslant \frac{i}{e\mu n}.$$

Since before the improvement is made, there are already

$$\min\{n/\ln(en), \mu\}$$

replicas, i is equal to $\min\{n/\ln(en), \mu\}$, which makes the expected runtime be

$$\frac{e\mu n}{\min\{n/\ln(en), \mu\}} \leqslant e\mu \ln(en) + en$$

The expected runtime of the (μ+1)-EA obtaining the first individual with fitness value above the threshold v equals to the sum of waiting time for each L value. Therefore, the overall waiting time is

$$v \cdot \left(2e\mu \ln(en) + \frac{e\mu n}{\min\{n/\ln(en), \mu\}}\right) \leqslant v(3e\mu \ln(en) + en)$$

In conclusion, the overall waiting time for $(\mu+1) - EA$ on *LeadingOnes* problem has obtained a solution of fitness value above the threshold v is $O(nv + \mu v \log n)$.

□

For a *LeadingOnes* problem with threshold v, there are 2^{n-v} different possible solutions. When $\mu > 2^{n-v}$, all of the 2^{n-v} different possible solutions should be contained in the optimal population set and there should be duplicates in the population. According to our definition of diversity, the duplicates will not affect the diversity measurement.

LEMMA 5. *The optimal solution of (μ+1)-EA$_D$ on LeadingOnes with threshold v has the population diversity $\mu^2(n-v)/4$.*

PROOF. Assume that there is a matrix which contains all individuals and each individual as a row, which is similar to the matrix for *OneMax* problem. Let m_i equal to the number of 0-bits in ith column. Then the contribution to diversity of each column can be represented as $m_i(\mu - m_i) = \mu m_i - {m_i}^2$. In the left v columns, there are only 1-bits so the contribution to diversity is 0. For the following $(n-v)$ columns, when $m_i = \mu/2$, the quadratic function reaches its maximal. The contribution of each column has no effect on those of other columns. Hence, if there is no duplicate in the population and each of the $(n-v)$ columns has $\mu/2$ 0-bits, the population diversity equals to $(\mu^2/4) \cdot (n-v) = \mu^2(n-v)/4$, which is the maximum value. □

LEMMA 6. *The expected waiting time of (μ+1)-EA$_D$ on LeadingOnes to achieve μ different solutions above the threshold, where $\mu \leqslant 2^{\frac{n-v}{2}-1}$, is bounded above by $O(nv + \mu v \log n + \mu n \log \mu)$.*

PROOF. After the first individual with fitness value above the threshold is achieved in $O(nv + \mu v \log n)$ time, another $(\mu - 1)$ individuals with fitness value above the threshold are produced before the diversity optimization process begins. This process will take $O(\mu \log \mu)$ time as proved in Lemma 1.

Since the duplicates make no contribution to the diversity and may interfere the optimization process, we should get rid of the duplicates at the beginning of the diversity optimization process. When there is duplicates in the population, a new individual will always be accepted and replace one of the duplicates. A new individual can be produced by selecting an individual and flipped one bit to become one of its undiscovered Hamming neighbours. An upper bound for the expected number of undiscovered Hamming neighbours of a set of individuals is given in [7] as at least $n - 2 \cdot r$ where $0 < |P| \leqslant 2^r$ and P is the set of discovered individuals. In the *LeadingOnes* problem, the v leftmost bits should be all $1's$. Only the $(n-v)$ other bits can be either 0-bit or 1-bit so the expected Hamming neighbours are at least $n - v - 2 \cdot r$. Since $\mu \leqslant 2^{\frac{n-v}{2}-1}$, the expected number $n - v - 2 \cdot r \geqslant 2$. Assume the number of non-duplicated individuals in the current population is s. Then the expected number of Hamming neighbours is equal to $(n - v - 2\log s)$. The probability of obtaining an undiscovered Hamming neighbour is at most

$$(n - v - 2\log s) \cdot \frac{s}{\mu} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geqslant \frac{s(n - v - 2\log s)}{e\mu n}.$$

Therefore the total time for obtaining a population with μ different individuals is

$$\begin{aligned}\sum_{s=1}^{\mu-1} \frac{e\mu n}{s(n-v-2\log s)} &= e\mu n \sum_{s=1}^{\mu-1} \frac{1}{s(n-v-2\log s)} \\ &\leqslant e\mu n \sum_{i=1}^{\mu-1} \frac{1}{s} \\ &\leqslant e\mu n \ln(e\mu)\end{aligned}$$

Hence, it takes at most $O(\mu n \log \mu)$ time to get a population set with no duplicates in it. Taken all stages into consideration, the expected runtime of (μ+1)-EA$_D$ on *LeadingOnes* to achieve μ solutions above the threshold is bounded above by

$$\begin{aligned} & O(nv + \mu v \log n) + O(\mu \log \mu) + O(\mu n \log \mu) \\ = \ & O(nv + \mu v \log n + \mu n \log \mu).\end{aligned}$$

□

Now, we show an upper bound for *LeadingOnes* that holds for $\mu \leq 2^{\frac{n-v}{2}-1}$.

THEOREM 5. *Let $\mu \leqslant 2^{\frac{n-v}{2}-1}$, then expected optimization time of (μ+1)-EA$_D$ on LeadingOnes is upper bounded by $O(nv + \mu v \log n + \mu n \log(\mu(n-v)))$.*

PROOF. According to Lemma 4, it takes at most $O(nv + \mu v \log n + \mu n \log \mu)$ time for (μ+1)-EA$_D$ on *LeadingOnes* to get a population of μ different individuals.

After the duplicates are replaced by different solutions, if each column has $\mu/2$ 0-bits, the population diversity should equal to the maximal value $\mu^2 n/4$ as proved in Lemma 5. In the worst case, the initial value of m_i of each column is either μ or 0. In order to increase the diversity of the population, m_i value of each column should either increase or decrease to $\mu/2$. Since the duplicates does no contribution to the population diversity, in this process, it should be guaranteed that the new individual produced is not a replica of any existing individuals.

Let s_i and t_i represent the number of 0-bits and 1-bits in the ith column separately. Then $|s_i - t_i| = d_i$ can be regarded as the balance rate of 0-bits and 1-bits in the ith column. Consider the event that selecting an individual randomly and flipping one of its 0-bits or 1-bits to decrease the balance rate of the column. When $s_i < t_i$, this event will cause the contribution $s_i \cdot t_i$ change by

$$(s_i + 1)(\mu - s_i - 1) - s_i(\mu - s_i) = \mu - 2s_i - 1,$$

as the contribution to diversity of other columns will not change. Since $s_i + t_i = \mu$ and $s_i < t_i$, $s_i < \mu/2$. Then the contribution change is at least 0. At the beginning of this stage, there is no duplicate in the population and in each iteration, it should be guaranteed that there is no duplicate introduced to the population. Since there is no duplicates in the parent population, there should exists at most $\min\{t_i, s_i\}$ individuals that are only different in the chosen column from any other individuals. Therefore, there exist at least $|s_i - t_i|$ individuals which have no replicas in pattern without considering the selected column , which also means there should be at least $(s_i - t_i)$ 0-bits that can be flipped without making a duplicates. The probability for such an event to happen is

$$\frac{1}{\mu} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geqslant \frac{1}{e\mu n}.$$

In a population which is not optimized, there should be $\sum_{i=0}^{n-v} |s_i - t_i|$ different mutations that can lead to diversity improvement through the event we described in the last paragraph. At first, we assume there are either all 1-bits or all 0-bits in each column which makes the total number of feasible mutation equals to $\mu(n - v)$. After each mutation, the number of feasible mutation is decreased by 2 according to the definition of balance rate. Let $d = \sum_{i=0}^{n-v} |s_i - t_i|$, then the expected time for the improvement is at most $e\mu n/d$. For the optimized population, the balance rate of each column should be 0. Before the population diversity is maximized, one bit flipping as discussed above causes the balance rate of the certain column decreased by 2. Hence, the total waiting time for the population diversity to be maximized is

$$\sum_{d=1}^{\mu(n-v)/2} e\mu n \cdot \frac{1}{d} \leqslant \frac{1}{2} e\mu n \ln(e\mu(n - v))$$

Hence, the overall runtime of (μ+1)-EA$_{\mathrm{D}}$ on *LeadingOnes* is bounded above by $O(n^2 + \mu v \log n + \mu n \log(\mu(n - v)))$. □

Conclusions

The population of an evolutionary algorithm can be used to generate a diverse set of solutions where all solutions are of good quality. We examined such approaches in a rigorous way by a first runtime analysis and studied a (μ+1)-EA$_{\mathrm{D}}$ which maximizes the diversity of the population once all solutions have fitness beyond a given threshold value v. Our results for the classical benchmark problems *OneMax* and *LeadingOnes* show that the algorithm is efficiently maximizing diversity of the population.

Our investigations should set the basis for analysis of diversity maximization for classical combinatorial optimization problems and it would be an interesting topic for future work to study the investigated (μ+1)-EA$_{\mathrm{D}}$ on classical combinatorial optimization problems such as the traveling salesperson problem or the vertex cover problem.

Acknowledgements

Frank Neumann has been supported by ARC grants DP130104395 and DP140103400.

References

[1] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing Co., Inc., 2011.

[2] S. Böttcher, B. Doerr, and F. Neumann. Optimal fixed and adaptive mutation rates for the LeadingOnes problem. In Schaefer et al. [11], pages 1–10.

[3] N. Chaiyaratana, T. Piroonratana, and N. Sangkawelert. Effects of diversity control in single-objective and multi-objective genetic algorithms. *J. Heuristics*, 13(1):1–34, 2007.

[4] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theor. Comput. Sci.*, 276:51–81, 2002.

[5] T. Friedrich, N. Hebbinghaus, and F. Neumann. Comparison of simple diversity mechanisms on plateau functions. *Theor. Comput. Sci.*, 410(26):2455–2462, 2009.

[6] T. Friedrich, P. S. Oliveto, D. Sudholt, and C. Witt. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation*, 17(4):455–476, 2009.

[7] C. Horoba and F. Neumann. Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 641–648, New York, NY, USA, 2008. ACM.

[8] T. Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Natural Computing Series. Springer, 2013.

[9] Y. Leung, Y. Gao, and Z. Xu. Degree of population diversity - a perspective on premature convergence in genetic algorithms and its markov chain analysis. *IEEE Transactions on Neural Networks*, 8(5):1165–1176, 1997.

[10] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization:Algorithms and Their Computational Complexity*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[11] R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph, editors. *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, volume 6238 of *Lecture Notes in Computer Science*. Springer, 2010.

[12] T. Ulrich, J. Bader, and L. Thiele. Defining and optimizing indicator-based diversity measures in multiobjective search. In Schaefer et al. [11], pages 707–717.

[13] T. Ulrich, J. Bader, and E. Zitzler. Integrating decision space diversity into hypervolume-based multiobjective search. In M. Pelikan and J. Branke, editors, *GECCO*, pages 455–462. ACM, 2010.

[14] T. Ulrich and L. Thiele. Maximizing population diversity in single-objective optimization. In N. Krasnogor and P. L. Lanzi, editors, *GECCO*, pages 641–648. ACM, 2011.

[15] R. K. Ursem. Diversity-guided evolutionary algorithms. In J. J. M. Guervós, P. Adamidis, H.-G. Beyer, J. L. F.-V. Martín, and H.-P. Schwefel, editors, *PPSN*, volume 2439 of *Lecture Notes in Computer Science*, pages 462–474. Springer, 2002.

[16] C. Witt. Runtime analysis of the $(\mu + 1)$ EA on simple pseudo-boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.