

# Expected Running Time Analysis of a Multiobjective Evolutionary Algorithm on Pseudo-boolean Functions

Nilanjan Banerjee and Rajeev Kumar

Department of Computer Science and Engineering  
Indian Institute of Technology Kharagpur  
Kharagpur, WB 721 302, India  
rkumar@cse.iitkgp.ernet.in

**Abstract.** In this paper we suggest a multiobjective evolutionary algorithm based on a restricted mating pool (REMO) with a separate archive for storing the remaining population. Such archive based algorithms have been used for solving real-world applications, however, no theoretical results are available. In this paper, we present a rigorous expected running time complexity analysis for the algorithm on two discrete pseudo boolean functions. We use the well known linear function LOTZ (Leading Zeros : Trailing Ones) and a continuous multiobjective quadratic function which is adapted to the discrete boolean space, for the analysis. The analysis shows that the algorithm runs with an expected time of  $O(n^2)$  on LOTZ. Moreover, we prove that the bound holds with an overwhelming probability. For an unary encoding of the multiobjective quadratic function  $((x - a)^2, (x - b)^2)$  in the boolean space, the expected running time of REMO is found to be  $O(n \log n)$ . A simple strategy based on partitioning of the decision space into fitness layers is used for the analysis.

## 1 Introduction

Evolutionary Algorithms (EAs) are randomized search heuristics that try to imitate the process of natural evolution. They are a broad class of heuristics that are applied to optimization problems. There are countless reports on the successful application of EAs with a huge number of empirical results but theoretical results are very few.

In case of single objective optimization theoretical analyses on running time complexity for few functions are available. Droste et al. [1] provided a rigorous theoretical analysis of the so called  $(1 + 1)$  EA on a wide variety of functions, especially linear functions. ONE-MAX function [2] has also been studied on the  $(1+1)$  EA. Results on the time bounds of algorithms in the discrete search space is available [3]. Rigorous proof on analysis of runtime in the continuous search space has been obtained only recently [4]. An analysis of Evolutionary Algorithms on whether crossover is essential is shown in [5]. However, work on theoretical analysis of multiobjective optimization is rare. Deriving sharp asymptotic bounds for multiobjective optimizers was started by Laumanns et al. [6] and is further extended by them in [7]. They defined a group of multiobjective pseudo-boolean functions which they analyze on their MOEAs.

The algorithms used in most of the work above use a single member population or an unbounded population. There is another group of genetic algorithms which use

an archive. Such algorithms have been empirically demonstrated to work efficiently but no theoretical analysis of such algorithms is available. In this work we define a simple archive based algorithm and perform an expected running time analysis of the algorithm on a linear and a quadratic function. We prove that for the linear function the expected running time is better than those obtained for the same function in [6, 7]. We also show that the algorithm performs efficiently for the well known bi-objective quadratic function. Thus, the work aims at the study of the behavior of an archive based algorithm on a few simple multiobjective functions.

The rest of the paper is organized as follows. Section 2 discusses the new algorithm REMO. Section 3 and 4 analyses the algorithm on the LOTZ and the quadratic multiobjective function, respectively. Section 5 concludes the paper.

## 2 Algorithm

---

### Restricted Evolutionary Multiobjective Optimizer (REMO)

---

1. Initialize two sets  $P = \phi$  and  $A = \phi$ , where  $P$  is the mating pool and  $A$  is an archive.
2. Choose an individual  $x$  uniformly at random from  $Y = \{0, 1\}^n$ .
3.  $P = \{x\}$ .
4. **loop**
5. Select an individual  $y$  from  $P$  at random.
6. Apply mutation operator on  $y$  by flipping a single randomly chosen bit and create  $y'$ .
7.  $P = P \setminus \{l \in P \mid l \prec y'\}$ .
8.  $A = A \setminus \{l \in A \mid l \prec y'\}$ .
9. **if** there does not exist  $z \in P \cup A$  such that  $z \succ y'$  or  $f(z) = f(y')$  **then**  $P = P \cup \{y'\}$
10. **end if.**
11. **if** cardinality of  $P$  is greater than 2 **then**
12. **Handler Function.**
13. **end if.**
14. **end loop.**

---

### Handler Function

---

1. For all the members  $x$  of  $P \cup A$  calculate a fitness function  $F(x, P \cup A) = H(x)$  where  $H(x)$  denotes the number of Hamming neighbors of  $x$  in  $P \cup A$ .
  2. Select the two individuals with the minimum  $F(x, P \cup A)$  values into  $P$  and put the rest of the individuals in the archive  $A$ . In case of equal  $F(x, P \cup A)$  values the selection is made at random.
-

### 3 LOTZ : Leading Ones Trailing Zeros

The LOTZ is a bi-objective linear function that was formulated and analyzed by Laumanns et al. [6, 7]. The authors proved expected runtimes of  $\theta(n^3)$  and  $\theta(n^2 \log n)$  for their algorithms SEMO (Simple Evolutionary Multiobjective Optimizer) and FEMO (Fair Evolutionary Multiobjective Optimizer) respectively on LOTZ. We show that REMO (algorithm above) has an expected running time of  $O(n^2)$  for LOTZ and moreover prove that the above bound holds with an overwhelming probability.

#### 3.1 Problem Definition

The Leading Ones (LO), Trailing Zeros (TZ) and the LOTZ problems can be defined as follows where the aim is to maximize both the objectives:

$$\begin{aligned} \text{LO}(\mathbf{x}) &= \sum_{i=1}^n \prod_{j=1}^i x_j, & \text{TZ}(\mathbf{x}) &= \sum_{i=1}^n \prod_{j=i}^n (1 - x_j) \\ \text{LOTZ}(\mathbf{x}) &= (\text{LO}(\mathbf{x}), \text{TZ}(\mathbf{x})). \end{aligned}$$

**Proposition 1.** The Pareto-optimal front for LOTZ can be represented as a set  $S = \{(i, n - i) \mid 0 \leq i \leq n\}$  and the Pareto set consists of all bit vectors belonging to the set  $P = \{1^i 0^{n-i} \mid 0 \leq i \leq n\}$ .

**Proof.** In the first part of the proof we aim to show that corresponding to any arbitrary non-Pareto optimal bit vector in the decision space we can always find an individual in  $P$  which dominates it. Let us consider any arbitrary individual  $Y = 1^i 0^{n-(i+j+2)} 10^j$  and an optimal bit string  $X = 1^k 0^{n-k}$  where  $0 \leq k \leq n$ . It is clear that if  $k = i + 1$  then  $\text{LO}(Y) < \text{LO}(X)$ . In the string  $Y$ , there is a 0 following the  $i$  Leading-ones and a 1 preceding the  $j$  Trailing-zeros. This implies that  $j \leq n - (i + 2)$  therefore  $\text{TZ}(Y) < \text{TZ}(X)$ , thus proving that  $X$  dominates  $Y$ .

Now we need to show that any bit string of the form  $X = 1^k 0^{n-k}$  where  $0 \leq k \leq n$  cannot be dominated by any other individual in the decision space. It is clear that if  $\text{LO}(Y)$  or  $\text{TZ}(Y)$  (where  $Y$  is defined in the above paragraph) is greater than that of  $X$  for a certain choices of  $i$  and  $j$ , the other objective is bound to be less for  $Y$ , thus implying that in the worst case the two strings are incomparable. The same argument holds for any string  $Z = 1^l 0^{n-l}$  where  $l \neq k$ ; thus proving the proposition.

#### 3.2 Analysis

The expected running time analysis of the function above is divided into two distinct phases. Phase 1 ends with the first Pareto-optimal point in the population  $P$ , and Phase 2 ends with the entire Pareto-optimal set in  $P \cup A$ .

**Theorem 1.** The expected running time of REMO on LOTZ is  $O(n^2)$ . The above bound holds with a probability of  $1 - e^{-\Omega(n)}$ .

**Proof.** We partition the decision space into fitness layers defined as  $(i, j)$ ,  $(0 \leq i, j \leq n)$  where  $i$  refers to the number of Leading-ones and  $j$  is the number of Trailing-zeros in a chromosome.

For LOTZ, in phase 1 the population cannot contain more than one individual for REMO because a single bit flip will create a child that is either dominating or is dominated by its parent. Phase 1 begins with an initial random bit vector in  $P$ . An individual can climb up a fitness layer  $(i, j)$  by a single bit mutation if it produces the child  $(i+1, j)$  or  $(i, j+1)$ . The probability of flipping any particular bit in the parent is  $\frac{1}{n}$ , thus the probability associated with such a transition is  $\frac{2}{n}$ . The factor of 2 is multiplied because we could either flip the leftmost 0 or the rightmost 1 for a success. Therefore, the expected waiting time for such a successful bit flip is at most  $\frac{n}{2}$ . If we pessimistically assume that Phase 1 begins with a random individual in the population then algorithm would require at most  $n$  successful mutation steps till the first Pareto-optimal point is found. Thus, it takes  $\frac{n^2}{2}$  steps for the completion of Phase 1. To prove that the above bound holds with an overwhelming probability let us consider that the algorithm is run for  $n^2$  steps. The expected number of successes for these  $n^2$  steps is at least  $2n$ . If  $S$  denotes the number of successes, then by Chernoff's bounds :

$$P[S \leq (1 - \frac{1}{2}) \cdot 2n] = P[S \leq n] \leq e^{-\frac{n}{4}} = e^{-\Omega(n)}.$$

Phase 2 begins with an individual of the form  $I = (i, n - i)$  in  $P$ . A success in Phase 2 is defined as the production of another Pareto-optimal individual. The first successful mutation in Phase 2 leads to production of the individual  $I_{+1} = (i + 1, n - i - 1)$  or  $I_{-1} = (i - 1, (n - i + 1))$  in the population  $P$ . The probability of such a step is given by  $\frac{2}{n}$ . Thus, the waiting time till the first success occurs is  $\frac{n}{2}$ . If we assume that after the first success  $I$  and  $I_{-1}$  are in  $P$  (without loss of generality), then the Pareto-optimal front can be described as two paths from  $1^{i-1}0^{n-i+1}$  to  $0^n$  and  $1^i0^{n-i}$  to  $1^n$ . At any instance of time  $T$ , let the individuals in  $P$  be represented by  $L = (l, n - l)$  and  $K = (k, n - k)$  where  $0 \leq k < l \leq n$ . As the algorithm would have followed the path from  $(i - 1, n - i + 1)$  to  $(k, n - k)$  and  $(i, n - i)$  to  $(l, n - l)$  to reach the points  $L$  and  $K$ , it is clear that at time  $T$  all the individuals of the form  $S = (j, n - j)$  with  $l < j < k$  have already been found and form a part of the archive  $A$ . Moreover, the handler function, assures that  $L$  and  $K$  are farthest in Hamming distance . At time  $T$  the probability of choosing any one individual for mutation is  $\frac{1}{2}$ . Let us assume, without loss of generality, that the individual selected is  $(k, n - k)$ . The flipping of the left most 0 produces the individual  $K_{+1} = (k + 1, n - k - 1)$  and the flipping of the rightmost 1 produces the individual  $K_{-1} = (k - 1, n - k + 1)$ . Since, the algorithm does not accept weakly dominated individuals and  $K_{+1}$  is already in  $A$ , the production of  $K_{-1}$  can only be considered as a success. Thus, the probability of producing another Pareto-optimal individual at time  $T$  is  $\frac{1}{4n}$ . The expected waiting time of producing another Pareto-optimal individual is at most  $4n$ . Since, no solutions on the Pareto-optimal front is revisited in Phase 2, it takes a maximum of  $n + 1$  steps for its completion. Therefore, REMO takes  $O(n^2)$  for Phase 2. By arguments similar to Phase 1, it can be shown that the bound in phase 2 holds with a probability  $1 - e^{-\Omega(n)}$ .

Altogether considering both the phases, REMO takes  $n^2$  steps to find the entire Pareto-optimal set for LOTZ. For the bound on the expected time we have not assumed anything about the initial population. Thus, the above bound on the probability holds for the next  $n^2$  steps. Since the lower bound on the probability that the algorithm will find the entire Pareto set is more than  $\frac{1}{2}$  (in fact exponentially close to 1) the expected

number of times the algorithm has to run is bounded by 2. Combining the results of both the phases 1 and 2 yields the bounds in the theorem.

## 4 Quadratic Function

We use a continuous bi-objective minimization function  $((x - a)^2, (x - b)^2)$  and adapt it to the discrete boolean decision space in the following manner:

$$\text{if } \|x\| = \sum_{i=1}^n x_i, \text{ QF} : ((\|x\| - a)^2, (\|x\| - b)^2)$$

**Proposition 2.** *The Pareto-optimal front of QF is the set  $F = \{(i^2, (i - (b - a))^2) \mid a \leq i \leq b\}$  and consists of individuals where  $a \leq \|x\| \leq b$ .*

**Proof.** First, we aim to prove that corresponding to any arbitrary individual in the decision space (which is not in  $F$ ) we can always find a bit vector in  $F$  which dominates it. We represent any arbitrary individual with  $i$  number of ones as  $X_i$ . If  $i < a$  then the objective value for  $X_i = ((i - a)^2, (i - b)^2)$ . In the best case  $i = a - 1$ . The value of  $\text{QF}(X_i) = (1, (b - a + 1)^2)$ . We can find the individual (in  $F$ ) with  $\|x\| = a$  to have both the objectives less than  $X_i$ , thus proving that  $X_i$  is dominated. A similar proof can be proposed for any bit vector in which  $\|x\| > b$ . Now we need to prove that there does not exist any bit vector that can dominate an individual in  $F$ . Let us represent an individual in  $F$  as  $X_f$ . It is obvious to see that any other bit vector in  $F$  is incomparable to  $X_f$ . An individual with  $\|x\| > b$  or  $\|x\| < a$  in the best case can have any one objective value lower than that of  $X_f$ . Therefore, either the individual is dominated or is incomparable to  $X_f$ ; thus proving the proposition.

### 4.1 Analysis

**Theorem 2.** *The expected running time of REMO on QF is  $O(n \log n)$  for any value of  $a$  and  $b$ .*

**Proof.** We partition the analysis into two phases. Phase 1 ends with the first Pareto-optimal point in  $P$  and the second phase continues till all the Pareto-optimal bit vectors are in  $P \cup A$ . In phase 1 there can be a maximum of 2 individuals in  $P \cup A$ . Thus, the archive  $A$  is empty. This is because a single bit mutation of a parent with  $\|x\| < a$  or  $\|x\| > b$  will produce an individual which is dominated by or dominates its parent. We partition the decision space into sets with individuals having the same number of ones. Let us consider a bit vector represented as  $I_d$  where  $d$  represents the number of ones in the individual. A single bit mutation of  $I_d$  is considered to be a success if the number of ones increases (decreases) when  $d < a$  ( $d > b$ ). Therefore a success  $S$  requires the flipping of any one of the  $d$  1-bits ( $n - d$  0-bits) when  $d < a$  ( $d > b$ ). The probability of a successful mutation  $P(S) = \frac{d}{2^n}$  (or  $\frac{n-d}{2^n}$ ). The expected waiting time of  $S$  is given by  $E(S) \leq \frac{2^n}{d}$  (or  $\frac{2^n}{n-d}$ ). The total expected time till the first Pareto optimal individual arrives in the population is at most  $\sum_{i=1}^n \frac{2^n}{d} = 2nH_n = 2n \log n + \theta(2n) = O(n \log n)$ , where  $H_n$  stands for the  $n^{\text{th}}$  harmonic number, by the linearity of expectations.

Phase 2 works with the assumption that  $b - a > 1$  or else there would be no second phase. The number of individuals in the population is bounded by 2. The selection

mechanism ensures that they are the bit vectors that are most capable of producing new individuals. The Pareto-front can be visualized as a path of individuals with number of ones varying from  $a$  to  $b$  or  $b$  to  $a$ . Let us represent any individual with  $a < \|x\| < b$  as  $I_k$  where  $k$  represents the number of ones in the bit vector. Such a bit vector can be produced either by an individual with  $k + 1$  ones or  $k - 1$  ones. The associated probability for such a successful mutation is at least  $\frac{k+1}{2n}$  and  $\frac{n-k+1}{2n}$  respectively. Hence, the expected waiting time till the  $I_k^{th}$  Pareto optimal point is in the population (assuming that its parent is in the population) is  $E(I_k) \leq \frac{2n}{k+1}$  and  $\frac{2n}{n-k+1}$  for the two cases above. Thus, the total expected time till all the Pareto points are in  $P \cup A$  is at most  $\sum_{k=a}^b E(I_k) \leq \sum_{k=a}^b \frac{2n}{k+1} \leq \sum_{k=0}^{b-a} \frac{2n}{k+1} = 2nH_{b-a}$ . Therefore, the expected time for Phase 2 is at most  $2nelog(b-a) + \theta(2ne) = O(nlog(b-a))$ . Since  $a$  and  $b$  can have a maximum value of  $n$  the expected running time for REMO on QF is  $O(nlogn)$ .

## 5 Discussion and Conclusions

In this paper, an archive based multiobjective evolutionary optimizer (REMO) is presented and a rigorous runtime complexity analysis of the algorithm on a linear multiobjective function and one multiobjective quadratic function is shown. The key feature of REMO is its special restricted population for mating and a separate archive. The idea is to restrict the mating pool to a constant  $c$ . The value of 2 for  $c$  is sufficient for most linear and quadratic functions. In case of certain linear functions a single individual population with a similar selection scheme as REMO may suffice. However, two bit vectors may be required for functions where the Pareto front can be reached via two paths as is the case of the quadratic function. The bounds for REMO presented in the paper are better to those found on the linear function LOTZ earlier in [6, 7].

## References

1. Droste, S., Jansen T., Wegener, I.: On the Analysis of the  $(1 + 1)$  Evolutionary Algorithm. *Theoretical Computer Science*, **276** (2002) 51 - 81.
2. Garnier, J. , Kallel, L., Schoenauer, M. : Rigorous Hitting Times for Binary Mutations. *Evolutionary Computation*, **7(2)** (2002) 167 - 203.
3. Droste, S. , Jansen, T., Wegener, I.: On the Optimization of Unimodal Functions with the  $(1 + 1)$  Evolutionary Algorithm. *Proceedings of the 5<sup>th</sup> Conference of Parallel Problem Solving from Nature (PPSN V)*, LNCS **1498** (1998) 13 - 22.
4. Jagerskupper, J. : Analysis of Simple Evolutionary Algorithm for Minimization in Euclidean Spaces. *Proceedings of the 30<sup>th</sup> International Colloquium on Automata, Languages and Programming*. LNCS **2719** (2003) 1068 - 1079.
5. Jansen,T. ,Wegener, I. : On the Analysis of Evolutionary Algorithms, a Proof that Crossover Really Can Help. *Proceedings of the 7<sup>th</sup> Annual European symposium of Algorithms(ESA 99)*. LNCS **1643** (1999) 184 - 193.
6. Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., Deb, K. : Running Time Analysis of Multiobjective Evolutionary Algorithms on a Discrete Optimization Problem. *Parallel Problem Solving from Nature (PPSN VII)*, LNCS **2439** (2002) 44 - 53.
7. Laumanns, M., Thiele, L., Zitzler, E.: Running Time Analysis of Evolutionary Algorithms on Vector-Valued Pseudo-Boolean Functions. *IEEE Transactions on Evolutionary Computation*, 2004.