CONVERGENCE PROPERTIES OF TWO $(\mu + \lambda)$ EVOLUTIONARY ALGORITHMS ON ONEMAX AND ROYAL ROADS TEST FUNCTIONS

Aram Ter-Sarkisov[1], Stephen Marsland[2]

[1]*School of Computer Science, Massey University, Wellington, New Zealand*
[2]*School of Computer Science, Massey University, Palmerston North, New Zealand*
{a.ter-sarkisov, s.r.marsland}@massey.ac.nz

Keywords: Evolutionary Algorithms, Computational Complexity, Recombination Operators

Abstract: We present a number of bounds on convergence time for two elitist population-based Evolutionary Algorithms using a recombination operator k-Bit-Swap and a mainstream Randomized Local Search algorithm. We study the effect of distribution of elite species and population size.

1 INTRODUCTION

The main objective of this article is to derive convergence properties of two elitist Evolutionary Algorithms (EAs) on OneMax and Royal Roads test functions. One of the analyzed algorithms uses the k-Bit-Swap (kBS) operator introduced in (Ter-Sarkisov et al., 2010). We compare our results to computational findings and other research.

A population consists of a set of solution strings. We split them into two groups: there are elite strings, which have the same highest fitness value and the remaining non-elite strings. We use the standard notation for the population μ, recombination pool λ, the elite species α, the non-elite β.

1.1 Past work

Recently $(1+1)$EA with $\frac{1}{n}$ flip probability (*n* being the length of a chromosome) became a matter of extensive investigation. Sharp lower and upper bounds for One-Max and general linear functions were found in (Doerr et al., 2010c; Doerr et al., 2010a; Doerr et al., 2011; Droste et al., 2002) applying drift analysis and potential functions. Specifically, in (Doerr et al., 2011) the upper bound for $(1+1)$EA solving OneMax was derived to be $(1 + o(1))1.39en \log n$ and in (Doerr et al., 2010a) the lower bound for the same setting was found to be $(1 - o(1))en \log n$. Drift (a form of super martingale) was introduced in (Hajek, 1982; He and Yao, 2003; He and Yao, 2004).

1.2 Definitions and Assumptions

We analyze two fitness functions here, OneMax (simple counting 1's test function) and Royal Roads (see Section 4 for additional definitions for it). The fitness of a population is defined as the fitness of an elite string. Since both functions have global solution at *n*, we are interested in the following time parameter:

$$\tau_A = \min\{t \geq 0 : f(\alpha) = n\} \tag{1}$$

that is, the minimum time when (for algorithm A) the best species in the population reaches the highest fitness value. Since the analysis is probabilistic, we need the expectation of this parameter: $\mathbf{E}\tau_A$.

We assume that we do not need a large number of species for evolution. Though this sounds a bit vague, this justifies the choice of distributions with respective parameters. The expectation of Poisson random variable used here is 1, for Uniform it is $\frac{\mu+1}{2}$. We use the latter due to its simplicity.

We restrict our attention only to elite pairs (kBS) or parents (RLS), to simplify the analysis, since otherwise we would have to make more assumptions about the fitness of non-elite parents β.

1.3 k-Bit-Swap Operator

This genetic recombination operator (see Figure 1) was introduced in (Ter-Sarkisov et al., 2010) and proved to work efficiently both alone and together with mainstream operators (crossovers and mutation). Its efficiency was mostly visible on functions like Rosenbrock, Ackley,
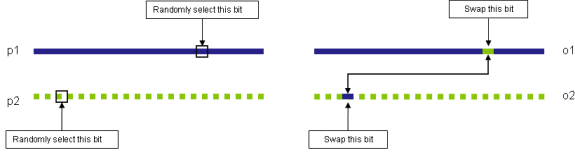
Figure 1: 1-Bit-Swap Operator

Algorithm 1:$(\mu+\lambda)\text{EA}_{1BS}$	
1	Initialize population size μ
	repeat until condition fulfilled:
2a	select λ species from the population using Tournament selection
2b	apply 1BS operator to each pair in the recombination pool
2c	keep α best species in the population, replace the rest with the best species from the pool
Algorithm 2:$(\mu+\lambda)\text{RLS}$	
1	Initialize population size μ
	repeat until condition fulfilled:
2a	select λ species from the population using Tournament selection
2b	flip exactly one bit per chromosome
2c	keep α best species in the population, replace the rest with the best species from the pool

Table 1: Pseudocode for the algorithms analyzed in this article

Rastrigin and Royal Roads. We also tested its performance on OneMax specifically for this article.

1.4 Our findings

The models we derive are complete, i.e. they are functions of just population size μ, recombination pool λ and length of the chromosome n, i.e. the actual parameters of EAs, though we make some weak assumptions about the pairing of parents.

We derive the expectation of convergence time for the population-based elitist EA with a recombination operator (1-Bit-Swap Operator) and mutation-based RLS. Our theoretical and computational findings confirm that for OneMax the benefit of population is unclear, i.e. its effect is not always positive. For Royal Roads it is always positive. This problem-specific issue was noticed before (see e.g. Figures 4 and 5 in (He and Yao, 2002)).

We use two distributions of elite species in the population: Uniform($\frac{1}{\mu}$) and Poisson(1). Since the expressions for the expected first hitting time of algorithms $\mathbf{E}\tau$ we have found are quite complicated, we do the computational estimation and find some asymptotic results as well.

Tournament Selection Procedure We use this selection because it is fairly straightforward in implementation and analysis.

- Select two species x_i, x_j uniformly at random

- if $f(x_i) = f(x_j)$, either x_i or x_j enters the pool at random

- else the species with better fitness enters the pool

2 ANALYSIS OF ALGORITHM 1 ON ONEMAX PROBLEM

We start with Uniform distribution of elite species with parameter $\frac{1}{\mu}$, which gives the lower bound on convergence time, which is due to the assumption on the number of elite species needed for the evolution.

The probability of selecting an elite pair is

$$P_{sel} = \left(\frac{\alpha}{\mu}\right)^4 + 4\left(\frac{\alpha}{\mu}\right)^3\frac{\beta}{\mu} + 4\left(\frac{\alpha}{\mu}\right)^2\left(\frac{\beta}{\mu}\right)^2$$
$$= \frac{\alpha^2(\alpha+2\beta)^2}{\mu^4} = \frac{\alpha^2(\alpha+2(\mu-\alpha))^2}{\mu^4} \quad (2)$$

Since we restrict the analysis only to elite pairs, the probability of evolution (generation of a better offspring as a result of 1-Bit-Swap) is

$$P_{swap} = \frac{1}{2} - 2\left(\frac{k}{n}\right)^2 \quad (3)$$

where $k = 0 : \frac{n}{2} - 1$, which is due to the assumption that at the start of the algorithm $f(\alpha) = \frac{n}{2}$.

2.1 Uniform distribution of elite species

We are deriving an upper bound on the probability (and, therefore, lower bound on the expectation of the first hitting time). We are interested in the probability of evolving at least 1 new elite species next generation, i.e. of at least 1 successful swap.

$P($ at least 1 new elite species in the population at $t+1)$
$= 1 - P($no new elite species in the population at $t+1)$

We define G_0 to be the event that no new species evolves over 1 particular generation. The number of elite pairs H_j in the population varies from 0 to $\frac{\lambda}{2}$, and elite species α in the population from 1 to μ. In this regard, probability to select a number of pairs given α elite species in the population is $P_{sel\alpha}$. By the Law of total probability,

$$P(G_0) = \sum_{j=0}^{\frac{\lambda}{2}} P(G_0|H_j)P(H_j)$$
$$= \sum_{j=0}^{\frac{\lambda}{2}} P(G_0|H_j)\sum_{\alpha=1}^{\mu} P(H_j|\alpha)P(\alpha) \quad (4)$$

We assume Uniform probability of each number of elite species in the population: $P(\alpha) = \frac{1}{\mu}$.

$$P(G_0|H_0)P(H_0) = P(G_0|H_0)\sum_{\alpha=1}^{\mu} P(H_0|\alpha)P(\alpha)$$

$$= \sum_{\alpha=1}^{\mu} P(H_0|\alpha)P(\alpha)$$

$$= \left(1 - P_{sel1}\right)^{\frac{\lambda}{2}} + \left(1 - P_{sel2}\right)^{\frac{\lambda}{2}} + \ldots + \left(1 - P_{sel\mu}\right)^{\frac{\lambda}{2}}$$

$$= \sum_{\alpha=1}^{\mu} \left(1 - P_{sel\alpha}\right)^{\frac{\lambda}{2}} \qquad (5)$$

to get 1 elite pair:

$$P(G_0|H_1)P(H_1) = \left(\frac{1}{2} + 2\left(\frac{k}{n}\right)^2\right) \frac{1}{\mu}\sum_{\alpha=1}^{\mu} P(H_1|\alpha)$$

and

$$\sum_{\alpha=1}^{\mu} P(H_1|\alpha) = \binom{\frac{\lambda}{2}}{1}\sum_{\alpha=1}^{\mu} P_{sel\alpha}\left(1 - P_{sel\alpha}\right)^{\frac{\lambda}{2} - 1}$$

therefore, the probability of failure given 1 elite pair given k improvements so far is

$$P(G_{0k}|H_1)P(H_1) = \left(\frac{1}{2} + 2\left(\frac{k}{n}\right)^2\right)\binom{\frac{\lambda}{2}}{1}\frac{1}{\mu}\sum_{\alpha=1}^{\mu} P_{sel\alpha}\left(1 - P_{sel\alpha}\right)^{\frac{\lambda}{2} - 1}$$

$$\qquad (6)$$

For cases $\{H_j : 2 \le j \le \frac{\lambda}{2}\}$ the logic is similar, so the full expression for the probability of failure is

$$P(G_{0k}) = \sum_{j=0}^{\frac{\lambda}{2}} P(G_0|H_j)P(H_j)$$

$$= \frac{1}{\mu}\sum_{j=0}^{\frac{\lambda}{2}} \left(\frac{1}{2} + 2\left(\frac{k}{n}\right)^2\right)^j \binom{\frac{\lambda}{2}}{j}\sum_{\alpha=1}^{\mu} P_{sel\alpha}^j\left(1 - P_{sel\alpha}\right)^{\frac{\lambda}{2} - j}$$

$$= \frac{1}{\mu}\sum_{j=0}^{\frac{\lambda}{2}} \left(\frac{1}{2} + 2\left(\frac{k}{n}\right)^2\right)^j \binom{\frac{\lambda}{2}}{j} \cdot$$

$$\cdot \sum_{\alpha=1}^{\mu} \left(\frac{\alpha^2(\alpha + 2(\mu - \alpha))^2}{\mu^4}\right)^j \left(1 - \frac{\alpha^2(\alpha + 2(\mu - \alpha))^2}{\mu^4}\right)^{\frac{\lambda}{2} - j}$$

Interchanging the sums and using the standard binomial expansion $(s + t)^n = \sum_{k=0}^{n}\binom{n}{k}s^k t^{n-k}$

$$P(G_{0k}) = \frac{1}{\mu}\sum_{\alpha=1}^{\mu}\sum_{j=0}^{\frac{\lambda}{2}}\binom{\frac{\lambda}{2}}{j}\left\{\left(\frac{1}{2} + 2\left(\frac{k}{n}\right)^2\right)\left(\frac{\alpha^2(\alpha + 2(\mu - \alpha))^2}{\mu^4}\right)\right\}^j$$

$$\left\{\left(1 - \frac{\alpha^2(\alpha + 2(\mu - \alpha))^2}{\mu^4}\right)\right\}^{\frac{\lambda}{2} - j}$$

$$= \frac{1}{\mu^{2\lambda+1}}\sum_{\alpha=1}^{\mu}\left\{\mu^4 - \left(\frac{1}{2} - 2\left(\frac{k}{n}\right)^2\right)(\alpha(\alpha + 2(\mu - \alpha)))^2\right\}^{\frac{\lambda}{2}}$$

The probability of evolution (obtaining a better species) is therefore

$$P(G_k) = 1 - P(G_{0k})$$

$$= 1 - \frac{1}{\mu^{2\lambda+1}}\sum_{\alpha=1}^{\mu}\left\{\mu^4 - \left(\frac{1}{2} - 2\left(\frac{k}{n}\right)^2\right)(\alpha(\alpha + 2(\mu - \alpha)))^2\right\}^{\frac{\lambda}{2}} \qquad (7)$$

for each $0 \le k \le \frac{n}{2} - 1$ we have

$$\mathbf{E}T_k = \frac{1}{1 - \frac{1}{\mu^{2\lambda+1}}\sum_{\alpha=1}^{\mu}\left\{\mu^4 - \left(\frac{1}{2} - 2\left(\frac{k}{n}\right)^2\right)(\alpha(\alpha + 2(\mu - \alpha)))^2\right\}^{\frac{\lambda}{2}}}$$

$$= \frac{\mu^{2\lambda+1}}{\mu^{2\lambda+1} - \sum_{\alpha=1}^{\mu}\left\{\mu^4 - \left(\frac{1}{2} - 2\left(\frac{k}{n}\right)^2\right)(\alpha(\alpha + 2(\mu - \alpha)))^2\right\}^{\frac{\lambda}{2}}}$$

and therefore the expected first hitting time for the algorithm is

$$\mathbf{E}\tau_{(\mu+\lambda)EA_{1BS}} = \sum_{k=0}^{\frac{n}{2}-1} \mathbf{E}T_k$$

$$= \mu^{2\lambda+1}\sum_{k=0}^{\frac{n}{2}-1}\frac{1}{\mu^{2\lambda+1} - \sum_{\alpha=1}^{\mu}\left\{\mu^4 - \left(\frac{1}{2} - 2\left(\frac{k}{n}\right)^2\right)(\alpha(\alpha + 2(\mu - \alpha)))^2\right\}^{\frac{\lambda}{2}}}$$

$$= \mu^{2\lambda+1}\phi(\lambda, \mu, n) \qquad (8)$$

Despite having two sums without closed forms, the convergence rate of this algorithms depends only on the size of the population, recombination pool and the length of the string, that is, the real-life parameters of EA. Therefore, the model is complete.

3 ANALYSIS OF ALGORITHM 2 SOLVING ONEMAX

For comparison, we derive $\mathbf{E}\tau$ for $(\mu + \lambda)\text{EA}_{RLS}$ using similar approach (Law of total probability + sum of Geometric RVs). Changes apply mostly to the selection probability, as we have no pairs to form:

$$P_{sel} = \left(\frac{\alpha}{\mu}\right)^2 + \frac{2\alpha\beta}{\mu^2} = \frac{\alpha(2\mu - \alpha)}{\mu^2} \qquad (9)$$

3.1 Uniform distribution of elite species

We use the same assumptions of uniform distribution of elite species in the population as with the $(\mu + \lambda)\text{EA}_{1BS}$.

Failure event G_0 is defined in the same way: no successful flips in the recombination pool, so the probability thereof is defined in a similar way to the one in Equation 4.

$$P(G_0) = \sum_{j=0}^{\lambda} P(G_0|H_j)\sum_{\alpha=1}^{\mu} P(H_j|\alpha)P(\alpha) \qquad (10)$$

Only in this case j is the number of elite parents in the pool and goes from 0 to λ.

$$P(H_j|\alpha)P(\alpha) = \binom{\lambda}{j}p_{sel}^j(1 - p_{sel})^{\lambda - j}$$

and therefore (using the same idea with the binomial expansion)

$$P(G_{0k}) = \frac{1}{\mu}\sum_{j=0}^{\lambda}\left(\frac{k}{n}\right)^j\binom{\lambda}{j}\sum_{\alpha=1}^{\mu} P_{sel}^j(1 - P_{sel})^{\lambda - j}$$

$$= \frac{1}{\mu}\sum_{\alpha=1}^{\mu}\sum_{j=0}^{\lambda}\binom{\lambda}{j}\left(\frac{k}{n}P_{sel}\right)^j(1 - P_{sel})^{\lambda - j}$$

$$= \frac{1}{\mu^{2\lambda+1}}\sum_{\alpha=1}^{\mu}[\mu^2 - \alpha(2\mu - \alpha)(1 - \frac{k}{n})]^{\lambda} \qquad (11)$$

Unfortunately, the closed expression for this sum exists only for specific values of λ, so we have to keep it this way and later obtain the results computationally.

$$P(G_k) = 1 - P(G_{0k}) = 1 - \frac{1}{\mu^{2\lambda+1}} \sum_{\alpha=1}^{\mu} [\mu^2 - \alpha(2\mu - \alpha)(1 - \frac{k}{n})]^{\lambda}$$

So the expected optimization time of the algorithm is

$$\mathbf{E}\tau_{(\mu+\lambda)RLS} = \sum_{k=\frac{n}{2}}^{n-1} \mathbf{E}T_k = \sum_{k=\frac{n}{2}}^{n-1} \frac{1}{P(G_k)}$$

$$= \mu^{2\lambda+1} \sum_{k=\frac{n}{2}}^{n-1} \frac{1}{\mu^{2\lambda+1} - \sum_{\alpha=1}^{\mu}(\mu^2 - \alpha(2\mu - \alpha)(1 - \frac{k}{n}))^{\lambda}} \quad (12)$$

As the case is with $(\mu+\lambda)EA_{1BS}$, this is a somewhat optimistic estimate, since it assigns fairly high probabilities to high proportions of elite species in the population. This is confirmed by numerical estimates.

4 ANALYSIS OF ALGORITHM 1 ON ROYAL ROADS FUNCTION

We use the setup for RR problem along the lines of (Mitchell, 1996) (referred to as R_1 in the book).The chromosome of length n is split into K blocks, each of length M. The fitness of each block is 0 if there are any 0s in the block, and M if all of the bits in it have value 1. The fitness of the chromosome is the sum of the value of the blocks, so it can take values $0, K, 2K, \ldots MK$. We index the blocks using index κ. Originally this problem was designed to test EA's capacity for recombining building blocks compared to other heuristics (for details see (Mitchell, 1996)).

Additionally, we introduce an auxiliary function used to measure progress between improvements in the fitness (the idea is similar to that in, e.g. (Doerr et al., 2010b; He and Yao, 2004)), which in our case is $V_{\kappa} = \mathrm{OneMax}(s_{\kappa})$ since both functions achieve the global optimum at $s_k = M$ and $\max f(s) = V_s = \sum V_{\kappa} = n$.

There is an important difference from the standard OneMax problem: unlike it, when parents exchange genetic information, it doesn't matter where the information comes from (which segment of the parent), but it matters where it is inserted, because it may mean that the fitness of the recipient segment has reached M, and therefore the fitness of the whole parent increased by the same value.

The second important observation is that of all segments in the chromosome there is one, which evolves first, denote it κ_1 (this is only possible due to the parameters of the EA in discussion). This means that segments evolve in a sequence: $\kappa_1, \kappa_2, \ldots, \kappa_k$.

We pessimistically assume that the best auxiliary function value in the first generation is $\frac{n}{2}$ and fitness function is 0. We also assume that the starting value in each bin κ is $\frac{M}{2}$. In the same way as with OneMax, we make assumptions about the distribution of elite species

in the population, rather than their exact or approximate number.

We start with introducing the probability of failure:

$$P(G_0) = \sum_{j=0}^{\frac{\lambda}{2}} P(G_0|H_j) \sum_{\alpha=1}^{\mu} P(H_j|\alpha)P(\alpha)$$

where all variables are the same as in $(\mu+\lambda)EA_{1BS}$ solving OneMax: H_j is j'th elite pair in the recombination pool λ, α is the number of elite species in the population μ with both highest fitness and auxiliary function values. The selection function is the same as Equation 2:

$$P_{sel}(\alpha) = \frac{\alpha^2(\alpha + 2(\mu - \alpha))^2}{\mu^4}$$

The successful event is defined as evolution of at least one more elite species in the population. The number of bits equal to 0 left to swap/flip in a segment we use l. So now the probability of successful swap in Equation 2 becomes

$$P_{swap} = \frac{2\left(\frac{M}{2} - l\right)\left(\frac{n}{2} + \frac{kM}{2} + l\right)}{n^2} = \frac{(M - 2l)(n + kM + 2l)}{2n^2} \quad (13)$$

The auxiliary function for each bin κ, V_{κ} lies between 0 and $\frac{M}{2} - 1$ and k between 0 and $K - 1$, where K is the total number of bins κ to fill. The probability of failure is

$$P_F = 1 - P_{swap} = 1 - \frac{(M - 2l)(n + kM + 2l)}{2n^2}$$

$$= \frac{2n^2 - (M - 2l)(n + kM + 2l)}{2n^2}$$

The probability to fail to improve a bit in a bin given l improvements so far is

$$P(G_{0l}) = \frac{1}{\mu} \sum_{j=0}^{\frac{\lambda}{2}} \left(\frac{2n^2 - (M - 2l)(n + kM + 2l)}{2n^2}\right)^j \binom{\frac{\lambda}{2}}{j}$$

$$\sum_{\alpha=1}^{\mu} \left(\frac{(\alpha(\alpha + 2\mu(\mu - \alpha)))}{\mu^2}\right)^j \left(1 - \frac{(\alpha(\alpha + 2\mu(\mu - \alpha)))}{\mu^2}\right)^{\frac{\lambda}{2} - j}$$

$$= \frac{1}{\mu} \sum_{j=0}^{\frac{\lambda}{2}} P_F^j \binom{\frac{\lambda}{2}}{j} \sum_{\alpha=1}^{\mu} (P_{sel}(\alpha))^j (1 - P_{sel}(\alpha))^{\frac{\lambda}{2} - j}$$

$$= \frac{1}{\mu} \sum_{\alpha=1}^{\mu} (1 - P_{sel}(\alpha)P_{swap})^{\frac{\lambda}{2}} \quad (14)$$

Therefore,

$$P(G_l) = 1 - P(G_{0l}) = 1 - \frac{1}{\mu} \sum_{\alpha=1}^{\mu} (1 - P_{sel}(\alpha)P_{swap})^{\frac{\lambda}{2}}$$

Expected time until improving the auxiliary function of a bin κ is

$$\mathbf{E}T_{\kappa} = \sum_{l=0}^{\frac{M}{2} - 1} \frac{1}{P(G_l)} \quad (15)$$

and, finally,summing over all k from 0 to $K - 1$ we obtain (since G depends on both l and k)

$$\mathbf{E}\tau_{(\mu+\lambda)EA_{1BS}} = \sum_{k=0}^{K-1} \sum_{l=0}^{\frac{M}{2} - 1} \frac{1}{P(G_{l,k})}$$

$$= \mu^{2\lambda+1} \sum_{k=0}^{K-1} \sum_{l=0}^{\frac{M}{2} - 1} \frac{1}{\mu^{2\lambda+1} - \sum_{\alpha=1}^{\mu}(\mu^4 - (\alpha(\alpha + 2(\mu - \alpha)))^2 P_{swap})^{\frac{\lambda}{2}}}$$

$$(16)$$

μ	$c(\mu)$
4	$\frac{24e}{41}$
10	≈ 1.58198
20	≈ 1.58198
30	≈ 1.58198

Table 2: Values of the normalizing constant, Equation 17

5 ANALYSIS OF ALGORITHM 2 ON ROYAL ROADS FUNCTION

Just as is the case with OneMax, we present the results for population-based RLS on Royal Roads. This model is a bit simpler since we do not have to pair the parents, and the selection is just

$$P(\alpha)_{sel} = \left(\frac{\alpha}{\mu}\right)^2 + \frac{2\alpha\beta}{\mu^2} = \frac{\alpha(2\mu-\alpha)}{\mu^2}$$

Instead of Uniform distribution of elite species in the population, we try Poisson distribution with parameter 1, and normalizing constant

$$c(\mu) = \frac{e}{\sum_{\alpha=1}^{\mu} \frac{1}{\alpha!}} = \frac{e\Gamma(\mu+1)}{e\Gamma(\mu+1,1) - \Gamma(\mu+1)} \quad (17)$$

since $\sum_{k=0}^{n} \frac{x^k}{k!} = \frac{e^\lambda \Gamma(n+1,\lambda)}{\Gamma(n+1)}$ where $\Gamma(n+1,\lambda)$ is incomplete Gamma function. The sizes of populations used in the computational experiments, the values of the normalizing constant are set in Table 2.

The flip probability is just

$$P_{flip} = \frac{M-2l}{n} \quad (18)$$

Probability of failure given l successful flips so far is

$$P(G_{0l}) = \frac{c(\mu)}{e} \sum_{j=0}^{\lambda} \left(1 - \frac{M-2l}{2n}\right)^j \binom{\lambda}{j}$$

$$\cdot \sum_{\alpha=1}^{\mu} \left(\frac{\alpha(2\mu-\alpha)}{\mu^2}\right)^j \left(1 - \frac{\alpha(2\mu-\alpha)}{\mu^2}\right)^{\lambda-j} \frac{1}{\alpha!}$$

$$= \frac{c(\mu)}{e} \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[1 - \frac{\alpha(2\mu-\alpha)}{\mu^2}\left(\frac{M-2l}{2n}\right)\right]^\lambda$$

$$= \frac{c(\mu)}{e\mu^{2\lambda}} \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[\mu^2 - \alpha(2\mu-\alpha)\left(\frac{M-2l}{2n}\right)\right]^\lambda$$

Therefore, the probability of success is

$$P(G_l) = 1 - P(G_{0l})$$

and the expected time to fill the first bin κ_1 is therefore

$$\mathbf{E}T_{\kappa 1} = \sum_{l=0}^{\frac{M}{2}-1} \frac{1}{1 - \frac{c(\mu)}{e\mu^{2\lambda}} \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[\mu^2 - \alpha(2\mu-\alpha)\left(\frac{M-2l}{2n}\right)\right]^\lambda}$$

$$= e\mu^{2\lambda} \sum_{l=0}^{\frac{M}{2}-1} \frac{1}{e\mu^{2\lambda} - c(\mu) \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[\mu^2 - \alpha(2\mu-\alpha)\left(\frac{M-2l}{2n}\right)\right]^\lambda}$$

Since we have k such bins and the probability of successful sampling does not depend on the number of 1's in the parent (unlike $(\mu+\lambda)EA_{1BS}$), we obtain the expected first

hitting time for the algorithm on RR:

$$\mathbf{E}\tau_{(\mu+\lambda)RLS} = e\mu^{2\lambda}$$

$$\sum_{k=0}^{K-1} \sum_{l=0}^{\frac{M}{2}-1} \frac{1}{e\mu^{2\lambda} - c(\mu) \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[\mu^2 - \alpha(2\mu-\alpha)\left(\frac{M-2l}{2n}\right)\right]^\lambda}$$

$$= Ke\mu^{2\lambda} \sum_{l=0}^{\frac{M}{2}-1} \frac{1}{e\mu^{2\lambda} - c(\mu) \sum_{\alpha=1}^{\mu} \frac{1}{\alpha!} \left[\mu^2 - \alpha(2\mu-\alpha)\left(\frac{M-2l}{2n}\right)\right]^\lambda} \quad (19)$$

6 COMPUTATIONAL RESULTS

Since the expressions derived in this article do not have a closed form, we find them computationally. To test our results, we run each algorithm with parameter set (μ, λ, n) with $\mu = \lambda$ almost always for 50 independent runs, each run was 2000 generations long. The average of optimization time is denoted $\bar{\tau}$. Probability distribution used for each model follow standard notation in Probability theory: R for Uniform and P for Poisson.

In general, results for Algorithm 1 tend to be better than for Algorithm 2 and for OneMax sharper than for RR. As we mentioned already, this is due to different patterns of dynamics of elite species and has to be investigated further. Apparently for both algorithms solving RR both Uniform($\frac{1}{\mu}$) and Poisson(1) distributions give a fairly rough approximation that we can improve both statically (using other parameters) and dynamically (modeling change in the number of elite species).

The other important result is that the increase of population size for both algorithms solving OneMax problem does not necessarily result in the improvement in performance, which we showed both theoretically and numerically. For RR the situation is much more clear: increase in the population always brings about the improvement in performance. Both models confirm this quite consistently.

7 CONCLUSIONS AND FUTURE WORK

We have derived expected running time for EAs based on two different genetic operators on two relatively simple fitness functions. In the future there are two extensions that we particularly plan to focus on:

Approximate results for Equations 8, 12, 16, 19   This is the most obvious of developments. Although these equations give good estimates for optimization time, and we have found some asymptotic lower bounds, it is desirable to find sharper bounds in the form $O(g(\mu, \lambda, n))$. A big problem here are the complicated expressions involving sums.

Evolution of elite species.   This area has seen little focus in EA community, and we are keen to develop a dynamic model for evolution of species. In this article the model is static, i.e. distribution of elite species in the population is fixed (Uniform or Poisson). As a result, some bounds, especially for RR, seem to be quite loose. If instead of assuming a probability distribution of elite species with fixed parameters we study the convergence of the distribution, we can derive sharper bounds on optimization time.

n	μ	λ	$\mathbf{E}\tau^R_{Algorithm1}$	$\tilde{\tau}_{Algorithm1}$	$\mathbf{E}\tau^R_{Algorithm2}$	$\tilde{\tau}_{Algorithm2}$
	1	2	112.9801	113.14	190.7979	192.12
	2	2	144.6145	218.12	116.2812	184.6
50	4	4	94.7621	145.4	73.124	193.54
	8	8	62.5691	121.86	48.596	184.52
	10	10	55.4784	116.98	43.488	197.92
	1	2	259.8688	265.84	449.9205	418.8
	2	2	332.6321	455.9048	271.56	393.39
100	4	4	215.2445	314.1	168.03	410.39
	8	8	139.2885	267.72	108.826	420.416
	10	10	122.4884	266.9	96.3978	405.22
	1	2	3743.2354	3682.6	6792.8	6715
	2	2	4791.3413	7072.7	4025.9876	6630
1000	4	4	3021.5468	4574.3	2413.0033	6990
	8	8	1872.3595	3866.5	1481.761	7016
	10	10	1616.4433	3807	1283.5502	6834

Table 3: Theoretical and computational bounds for OneMax test function with the assumption of Uniform distributions of elite species in the population (Equations 8 and 12)

n	K	M	μ	λ	$\mathbf{E}\tau^R_{Algorithm1}$	$\tilde{\tau}_{Algorithm1}$	$\mathbf{E}\tau^P_{Algorithm2}$	$\tilde{\tau}_{Algorithm2}$
			4	4	145	315.3077	64.8084	672.25
32	4	8	10	10	72.4	268.2195	58.124	504.625
			20	20	44.2	192.2917	56.175	334.125
			30	30	34.5	173.5625	55.1	221
			4	4	570.625	612.46	249.959	-
64	8	8	10	10	279.88	497.93	222.565	820.6667
			20	20	153.46	454.4681	212.452	715.92
			30	30	112.297	372.04	209.373	663.6744
			4	4	2264.36	1365	1021	-
128	16	8	10	10	1048	1239	940.999	-
			20	20	570.44	1091.5	887.396	1612
			30	30	401.99	949.4	871.1131	1505

Table 4: Theoretical and computational bounds for Royal Roads test function with the assumptions of Uniform and Poisson distributions of elite species in the population (Equations 16 and 19).

REFERENCES

Doerr, B., Fouz, M., and Witt, C. (2010a). Quasirandom Evolutionary Algorithm. In *Genetic and Evolutionary Computing Conference (GECCO) 2010*, pages 1457–1464.

Doerr, B., Johannsen, D., and Winzen, C. (2010b). Drift Analysis and Linear Functions Revisited. In *Genetic and Evolutionary Computing Conference (GECCO) 2010*, pages 1–8.

Doerr, B., Johannsen, D., and Winzen, C. (2010c). Multiplicative Drift Analysis. In *Genetic and Evolutionary Computing Conference (GECCO) 2010*, pages 1449–1456.

Doerr, B., Johannsen, D., and Winzen, C. (2011). Multiplicative Drift Analysis. in press.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81.

Hajek, B. (1982). Hitting-Times and Occupation-Time Bounds Implied by Drift Analysis with Applications. *Advanced Applied Probability*, 14:502–525.

He, J. and Yao, X. (2002). From an Individual to a Population: An Analysis of the First Hitting Time of Population-Based Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 6-5, October 2002:495–511.

He, J. and Yao, X. (2003). Towards an analytic framework for analysing the computational time of evolutionary algorithm. *Artificial Intelligence*, 145(2003):59–97.

He, J. and Yao, X. (2004). A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(2004):21–35.

Mitchell, M. (1996). *Introduction to Genetic Algorithms*. Kluwer Academic Publishers.

Ter-Sarkisov, A., Marsland, S., and Holland, B. (2010). The k-Bit-Swap: A New Genetic Algorithm Operator. In *Genetic and Evolutionary Computing Conference (GECCO) 2010*, pages 815–816.