OneMax Helps Optimizing XdivK: Theoretical Runtime Analysis for RMHC and EA+RL

Proofs and Tables

Maxim Buzdalov
ITMO University
49 Kronverkskiy prosp.
Saint-Petersburg, Russia
mbuzdalov@gmail.com

Arina Buzdalova
ITMO University
49 Kronverkskiy prosp.
Saint-Petersburg, Russia
abuzdalova@gmail.com

1. DEFINITIONS

The OneMax problem is often used in theoretical research on evolutionary algorithms. Given n, the size of the problem. The search space consists of all bit vectors of length n. The fitness function of a bit vector is the number of bits set to one. One needs to maximize the fitness function.

The XdivK problem is defined in this paper in the similar way. The only difference to OneMax is that the fitness function is the number of bits set to one divided evenly by the parameter k. In other words, $\text{XdivK}(x) = \lfloor \text{OneMax}(x)/k \rfloor$. The parameter k is a divisor of n. We also define the problem XdivK+OneMax, which has the target objective XdivK and a single extra objective OneMax.

We consider a simplistic optimization algorithm called "random mutation hill climbing algorithm", or simply RMHC, which is also called the "randomized local search", or RLS. It stores a bit vector x — the current candidate solution. In the beginning of the algorithm, x is initialized in some way — this paper assumes that the initial vector contains only zero bits. An iteration of this algorithm works this way:

- $y \leftarrow x$ with exactly one random bit flipped;
- if $f(y) \geq f(x)$ then $x \leftarrow y$;
- if $f(x)$ is the maximum, terminate.

Note that if $f(x) = f(y)$, the previous solution is discarded. While this does not change the current fitness, it makes exploration of plateaus possible, and, in fact, XdivK cannot be solved by RMHC without this assumption.

2. EXPECTED RUNNING TIME

In this section, we derive the exact expressions for the expected running time for both XdivK problem solved by RMHC and XdivK+OneMax problem solved by RMHC under the control of reinforcement learning. We assume that the RL state is determined solely by the value of XdivK fitness function, and the reward is equal to the difference of XdivK fitness values in consecutive optimization states. We also assume that there are n bits in the bit vector, and the division factor is k, such that $n \bmod k = 0$.

The expressions for the expected running time are, in fact, quite similar. Due to this fact, in the rest of this section the similar formulae and proofs will be grouped together. The similarity follows from the following fact:

Lemma 1. *The EA+RL algorithm never returns to any state where some reward has been obtained.*

Proof. The experience (i.e. non-zero reward) can be gained only by escaping the current RL state, which can happen only if the XdivK fitness increases or decreases. However, the latter cannot happen: assume that the XdivK fitness decreases, then the OneMax fitness decreases as well, so regardless the choice of the fitness function, RMHC algorithm allowed the current fitness function to decrease, which cannot happen.

The last fact also ensures that once the algorithm escapes a certain RL state, it can not eventually return to it. □

As a corollary to Lemma 1, in the case of XdivK+OneMax both the XdivK and the OneMax fitness functions are always chosen with the probability of $1/2$.

2.1 Markov Chains

To compute the expected running time of the algorithms, we construct Markov chains for them. Unlike the definition of RL state, the state of a Markov chain is determined in this paper by the OneMax fitness value. The states are clustered into consecutive groups of n/k states, except for the terminal state n, which is on its own (Fig. 1). The clusters correspond to the states with the same value of XdivK fitness.

The Markov chains look similar for both algorithms, however, for different algorithms the transitions have different probabilities. For the XdivK problem, the probability of going up from the state x is always $(n-x)/n$. The probability of going down is zero for states with $x \bmod k = 0$ (colored blue in the picture), and is equal to x/n for all other node. For the states where $x \bmod k = 0$, the probability of remaining in the same state is x/n (Fig. 2).

In the case of the XdivK+OneMax problem, the probability of going up is $(n-x)/n$, which is composed from $(n-x)/(2n)$ for the XdivK fitness and $(n-x)/(2n)$ for the OneMax fitness. For the states where $x \bmod k = 0$, the probability of remaining in the same state is x/n, which is

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.

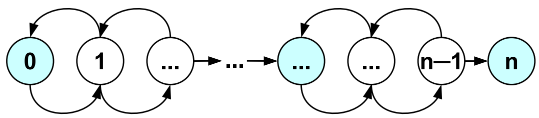

Figure 1: An overview of the Markov chain. The states correspond to OneMax fitness value, the clusters of states correspond to XdivK fitness value. Loops are not shown.

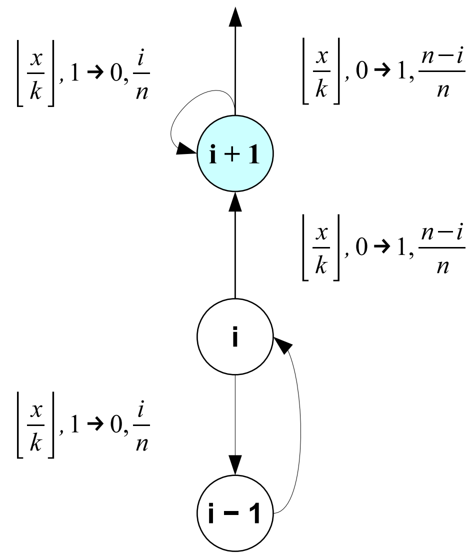

Figure 2: Markov chain fragment for the analysis of expected running time for XdivK. The labels on the transitions have the format F,M,P where F is the chosen fitness function, M is the effect of mutation, P is the probability of the combination.

$x/(2n)$ for XdivK plus $x/(2n)$ for OneMax. For all other states, the probability of remaining in the same state is $x/(2n)$ (in the case OneMax is chosen), and the probability of going down is $x/(2n)$ (Fig. 3).

2.2 Expected Number of Steps Between States

Let $Z_X(x)$ be the expected number of steps for RMHC solving XdivK problem to reach the state $x+1$ from the state x.

If $x \bmod k = 0$, we can only go up or stay at the same state (see Fig. 2):

$$Z_X(x) = \frac{n-x}{n} + \frac{x}{n}(1 + Z_X(x)) = \frac{n}{n-x}. \tag{1}$$

If $x \bmod k \neq 0$, we can go either up or down. In the latter case, we know the expected number of steps to return back, so we can express $Z_X(x)$ in terms of $Z_X(x-1)$:

$$\begin{aligned} Z_X(x) &= \frac{n-x}{n} + \frac{x}{n}(1 + Z_X(x-1) + Z_X(x)) = \\ &= \frac{n}{n-x} + Z_X(x-1)\frac{x}{n-x}. \end{aligned} \tag{2}$$

The equivalent value $Z_R(x)$ for RMHC under EA+RL solving XdivK+OneMax can be computed for $x \bmod k = 0$

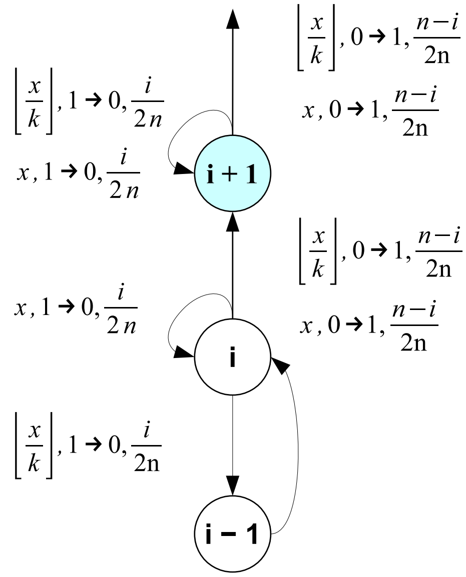

Figure 3: Markov chain fragment for the analysis of expected running time for XdivK+OneMax. The labels on the transitions have the format F,M,P where F is the chosen fitness function, M is the effect of mutation, P is the probability of the combination.

as follows:

$$Z_R(x) = \frac{n-x}{n} + \frac{x}{n}(1 + Z_R(x)) = \frac{n}{n-x}, \tag{3}$$

and for $x \bmod k \neq 0$ as follows:

$$\begin{aligned} Z_R(x) \quad &= \frac{n-x}{n} + \frac{x}{2n}(1 + Z_R(x)) + \\ &\quad + \frac{x}{2n}(1 + Z_R(x-1) + Z_R(x)) = \\ &= \frac{n}{n-x} + Z_R(x-1)\frac{x}{2(n-x)}. \end{aligned} \tag{4}$$

2.3 Recursion Elimination

We need to eliminate recursion in the expressions for Z_X and Z_R.

Theorem 1. *The following holds for all x:*

$$Z_X(x) = \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}. \tag{5}$$

Proof. We prove the theorem by induction. For all x such that $x \bmod k = 0$ the induction statement follows directly from (1):

$$Z_X(x) = \frac{n}{n-x} = \frac{\binom{n}{x}}{\binom{n-1}{x}} = \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}.$$

Assume that $v > 0$ and the theorem is proven for all x' such that $x' \bmod k = v-1$. Then for all x with $x \bmod k = v$

the following can be deduced using (2):

$$
\begin{aligned}
Z_X(x) &= \frac{n}{n-x} + Z_X(x-1)\frac{x}{n-x} = \\
&= \frac{n}{n-x} + \frac{x}{n-x}\left(\sum_{i=0}^{(x-1) \bmod k} \frac{\binom{n}{x-1-i}}{\binom{n-1}{x-1}}\right) = \\
&= \frac{n}{n-x} + \frac{x}{n-x}\left(\sum_{i=1}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x-1}}\right) = \\
&= \frac{n}{n-x} + \sum_{i=1}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}} = \\
&= \frac{\binom{n}{x}}{\binom{n-1}{x}} + \sum_{i=1}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}} = \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}. \quad \square
\end{aligned}
$$

THEOREM 2. *The following holds for all x:*

$$
Z_R(x) = \sum_{i=0}^{x \bmod k} 2^{-i} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}. \tag{6}
$$

PROOF. The theorem can be proven by induction using the same algorithm as above atop (3) and (4). $\square$

2.4 Total Expected Running Time

The total running time for RMHC solving XDIVK problem starting from all-zero bit vector is:

$$
T_X(n,k) = \sum_{x=0}^{n-1} Z_X(x) = \sum_{x=0}^{n-1} \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}. \tag{7}
$$

Similarly, for RMHC using EA+RL solving XDIVK+ONEMAX problem the total running time is:

$$
T_R(n,k) = \sum_{x=0}^{n-1} Z_R(x) = \sum_{x=0}^{n-1} \sum_{i=0}^{x \bmod k} 2^{-i} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}. \tag{8}
$$

These expressions are rather hard to analyse, especially if it is needed to compare T_X and T_R in terms of asymptotic behavior. We rewrite $T_X(n,k)$ as follows:

$$
\begin{aligned}
T_X(n,k) &= \sum_{x=0}^{n-1} \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}} = \\
&= \sum_{m=0}^{\frac{n}{k}-1} \sum_{j=0}^{k-1} \sum_{i=0}^{j} \frac{\binom{n}{mk+j-i}}{\binom{n-1}{mk+j}} = \\
&= \sum_{m=0}^{\frac{n}{k}-1} \sum_{i=0}^{k-1} \sum_{j=i}^{k-1} \frac{\binom{n}{mk+j-i}}{\binom{n-1}{mk+j}} = \\
&= \sum_{i=0}^{k-1} \sum_{j=i}^{k-1} \sum_{m=0}^{\frac{n}{k}-1} \frac{\binom{n}{mk+j-i}}{\binom{n-1}{mk+j}}.
\end{aligned} \tag{9}
$$

We ge the following by rewriting $T_R(n,k)$ in exactly the same way:

$$
\begin{aligned}
T_R(n,k) &= \sum_{x=0}^{n-1} \sum_{i=0}^{x \bmod k} 2^{-i} \frac{\binom{n}{x-i}}{\binom{n-1}{x}} = \\
&= \sum_{i=0}^{k-1} 2^{-i} \sum_{j=i}^{k-1} \sum_{m=0}^{\frac{n}{k}-1} \frac{\binom{n}{mk+j-i}}{\binom{n-1}{mk+j}}.
\end{aligned} \tag{10}
$$

The idea of these transformations is to promote 2^{-i} as much outwards as possible to simplify estimating the complexity of T_X and T_R.

2.5 Complexity Estimation

In this section we derive the complexity estimation for T_X and T_R.

First, let us define $V(n,k,i)$ as:

$$
V(n,k,i) = \sum_{j=i}^{k-1} \sum_{m=0}^{\frac{n}{k}-1} \frac{\binom{n}{mk+j-i}}{\binom{n-1}{mk+j}}. \tag{11}
$$

Then, from (9) and (11), it follows that:

$$
T_X(n,k) = \sum_{i=0}^{k-1} V(n,k,i) \tag{12}
$$

and equivalently from (10) and (11):

$$
T_R(n,k) = \sum_{i=0}^{k-1} 2^{-i} V(n,k,i). \tag{13}
$$

LEMMA 2. *For all i, x, such that $0 \le i < k$, $0 \le x < n-i-1$, the following holds:*

$$
\frac{\binom{n}{x}}{\binom{n-1}{x+i}} < \frac{\binom{n}{x+1}}{\binom{n-1}{x+i+1}}. \tag{14}
$$

PROOF.

$$
\begin{aligned}
\frac{\frac{\binom{n}{x}}{\binom{n-1}{x+i}}}{\frac{\binom{n}{x+1}}{\binom{n-1}{x+i+1}}} &= \frac{(x+1)(n-x-i-1)}{(n-x)(x+i+1)} = \\
&= \left(1 - \frac{i}{x+i+1}\right)\left(1 - \frac{i+1}{n-x}\right) < 1. \quad \square
\end{aligned}
$$

As a simple corollary to (14), the following is true:

$$
\frac{\binom{n}{x}}{\binom{n-1}{x+i}} \le \binom{n}{n-i-1}. \tag{15}
$$

THEOREM 3. *If k is constant of n, then:*

$$
V(n,k,i) = \Omega(n^{i+1}) = O(n^{i+2}). \tag{16}
$$

PROOF. If we consider in (11) the case when $j = k-1$ and $m = n/k - 1$, then the ratio of binomial quotients is $\binom{n}{n-i-1}$, so:

$$
\binom{n}{n-i-1} \le V(n,k,i).
$$

From (15), it follows that:

$$
V(n,k,i) \le \binom{n}{n-i-1}(k-i)\frac{n}{k}.
$$

If k is constant of n, then it follows from $i < k$ that $\binom{n}{n-i-1} = \Theta(n^{i+1})$. The asymptotic bounds follow from this fact and the previous equation. $\square$

THEOREM 4. *The complexity of both $T_X(n,k)$ and $T_R(n,k)$ for constant k is $\Omega(n^k)$ and $O(n^{k+1})$.*

PROOF. This follows from (12), (13), and (16). $\square$

THEOREM 5. *For sufficiently large n and fixed k*

$$T_X(n,k) \geq 2^{k-2}(1-o(1))T_R(n,k).$$

PROOF. The following is true:

$$T_X(n,k) = V(n,k,k-1) + V(n,k,k-2) + o(n^k)$$

$$T_R(n,k) = \frac{V(n,k,k-1)}{2^{k-1}} + \frac{V(n,k,k-2)}{2^{k-2}} + o(n^k).$$

Consider three cases:

1. $V(n,k,k-1) = \Theta(n^k)$, $V(n,k,k-2) = \Theta(n^k)$. In this case, we estimate the ratio as:

$$\frac{T_X(n,k)}{T_R(n,k)} = \frac{V(n,k,k-1) + V(n,k,k-2) + o(n^k)}{\frac{V(n,k,k-1)}{2^{k-1}} + \frac{V(n,k,k-2)}{2^{k-2}} + o(n^k)} \geq$$
$$\geq \frac{V(n,k,k-1) + V(n,k,k-2) + o(n^k)}{\frac{V(n,k,k-1)+V(n,k,k-2)}{2^{k-2}} + o(n^k)} =$$
$$= \frac{2^{k-2} + \frac{o(n^k)}{V(n,k,k-1)+V(n,k,k-2)}}{1 + \frac{o(n^k)}{V(n,k,k-1)+V(n,k,k-2)}} =$$
$$= \frac{2^{k-2} + o(1)}{1+o(1)} \geq \frac{2^{k-2}}{1+o(1)} = 2^{k-2}(1-o(1)).$$

2. $V(n,k,k-1) = \Theta(n^k)$, $V(n,k,k-2) = o(n^k)$. In this case, we estimate the ratio as:

$$\frac{T_X(n,k)}{T_R(n,k)} = \frac{V(n,k,k-1) + o(n^k)}{\frac{V(n,k,k-1)}{2^{k-1}} + o(n^k)} \geq$$
$$\geq \frac{2^{k-1} + \frac{o(n^k)}{V(n,k,k-1)}}{1 + \frac{o(n^k)}{V(n,k,k-1)}} =$$
$$= \frac{2^{k-1} + o(1)}{1+o(1)} \geq \frac{2^{k-1}}{1+o(1)} = 2^{k-1}(1-o(1)).$$

3. $V(n,k,k-1) = \omega(n^k)$. In this case, we estimate the ratio as:

$$\frac{T_X(n,k)}{T_R(n,k)} = \frac{V(n,k,k-1) + O(n^k)}{\frac{V(n,k,k-1)}{2^{k-1}} + O(n^k)} \geq$$
$$\geq \frac{2^{k-1} + \frac{O(n^k)}{V(n,k,k-1)}}{1 + \frac{O(n^k)}{V(n,k,k-1)}} =$$
$$= \frac{2^{k-1} + o(1)}{1+o(1)} \geq \frac{2^{k-1}}{1+o(1)} = 2^{k-1}(1-o(1)).$$

To sum up, the first case yields the ratio estimation of $2^{k-2}(1-o(1))$, while two latter cases yield $2^{k-1}(1-o(1))$. □

3. ASYMPTOTIC BEHAVIOR EVALUATION

The values of $T_X(n,k)$ and $T_R(n,k)$ for several n and k are presented in Table 1. One can see that the value of T_X/T_R grows slowly with n when k is constant. The ratio is very much similar to 2^{k-1}.

For asymptotic determination, we tabulate the values of $T_X(n,k)$ and $T_X(n/2,k)$ for several n and k. The results are presented in Table 2. One can see that the value of $T_X(n,k)/T_X(n/2,k)$ approaches 2^k as n grows, which suggests that the reality is $T_X(n,k) = \Theta(n^k)$.

From these two results it follows that in Theorem 5, the second case ($V(n,k,k-1) = \Theta(n^k)$, $V(n,k,k-2) = o(n^k)$) reflects the situation in practice.

Table 1: Evaluation of T_X/T_R ratio

n	k	$T_X(n,k)$	$T_R(n,k)$	$T_X(n,k)/T_R(n,k)$
40	2	$1.19 \cdot 10^3$	$6.81 \cdot 10^2$	1.75
48	2	$1.70 \cdot 10^3$	$9.56 \cdot 10^2$	1.78
56	2	$2.30 \cdot 10^3$	$1.28 \cdot 10^3$	1.80
64	2	$2.98 \cdot 10^3$	$1.64 \cdot 10^3$	1.81
72	2	$3.76 \cdot 10^3$	$2.05 \cdot 10^3$	1.83
80	2	$4.62 \cdot 10^3$	$2.51 \cdot 10^3$	1.84
60	3	$3.94 \cdot 10^4$	$1.08 \cdot 10^4$	3.66
72	3	$6.79 \cdot 10^4$	$1.83 \cdot 10^4$	3.71
84	3	$1.08 \cdot 10^5$	$2.86 \cdot 10^4$	3.76
96	3	$1.60 \cdot 10^5$	$4.23 \cdot 10^4$	3.79
108	3	$2.28 \cdot 10^5$	$5.97 \cdot 10^4$	3.81
120	3	$3.12 \cdot 10^5$	$8.14 \cdot 10^4$	3.83
80	4	$1.72 \cdot 10^6$	$2.30 \cdot 10^5$	7.45
96	4	$3.57 \cdot 10^6$	$4.72 \cdot 10^5$	7.55
112	4	$6.61 \cdot 10^6$	$8.68 \cdot 10^5$	7.62
128	4	$1.13 \cdot 10^7$	$1.47 \cdot 10^6$	7.66
144	4	$1.81 \cdot 10^7$	$2.35 \cdot 10^6$	7.70
160	4	$2.76 \cdot 10^7$	$3.57 \cdot 10^6$	7.73
100	5	$8.05 \cdot 10^7$	$5.37 \cdot 10^6$	14.98
120	5	$2.02 \cdot 10^8$	$1.33 \cdot 10^7$	15.16
140	5	$4.38 \cdot 10^8$	$2.86 \cdot 10^7$	15.28
160	5	$8.56 \cdot 10^8$	$5.57 \cdot 10^7$	15.38
180	5	$1.55 \cdot 10^9$	$1.00 \cdot 10^8$	15.45
200	5	$2.63 \cdot 10^9$	$1.69 \cdot 10^8$	15.50

Table 2: Evaluation of T_X asymptotic

n	k	$T_X(n,k)$	$T_X(n/2,k)$	$T_X(n,k)/T_X(n/2,k)$
40	2	$1.19 \cdot 10^3$	$3.11 \cdot 10^2$	3.82
48	2	$1.70 \cdot 10^3$	$4.42 \cdot 10^2$	3.84
56	2	$2.30 \cdot 10^3$	$5.95 \cdot 10^2$	3.86
64	2	$2.98 \cdot 10^3$	$7.71 \cdot 10^2$	3.87
72	2	$3.76 \cdot 10^3$	$9.69 \cdot 10^2$	3.88
80	2	$4.62 \cdot 10^3$	$1.19 \cdot 10^3$	3.89
60	3	$3.94 \cdot 10^4$	$5.07 \cdot 10^3$	7.78
72	3	$6.79 \cdot 10^4$	$8.67 \cdot 10^3$	7.83
84	3	$1.08 \cdot 10^5$	$1.37 \cdot 10^4$	7.86
96	3	$1.60 \cdot 10^5$	$2.03 \cdot 10^4$	7.88
108	3	$2.28 \cdot 10^5$	$2.88 \cdot 10^4$	7.90
120	3	$3.12 \cdot 10^5$	$3.94 \cdot 10^4$	7.91
80	4	$1.72 \cdot 10^6$	$1.07 \cdot 10^5$	16.10
96	4	$3.57 \cdot 10^6$	$2.21 \cdot 10^5$	16.10
112	4	$6.61 \cdot 10^6$	$4.11 \cdot 10^5$	16.10
128	4	$1.13 \cdot 10^7$	$7.02 \cdot 10^5$	16.09
144	4	$1.81 \cdot 10^7$	$1.13 \cdot 10^6$	16.09
160	4	$2.76 \cdot 10^7$	$1.72 \cdot 10^6$	16.08
100	5	$8.05 \cdot 10^7$	$2.43 \cdot 10^6$	33.14
120	5	$2.02 \cdot 10^8$	$6.11 \cdot 10^6$	32.98
140	5	$4.38 \cdot 10^8$	$1.33 \cdot 10^7$	32.85
160	5	$8.56 \cdot 10^8$	$2.61 \cdot 10^7$	32.75
180	5	$1.55 \cdot 10^9$	$4.73 \cdot 10^7$	32.68
200	5	$2.63 \cdot 10^9$	$8.05 \cdot 10^7$	32.61