

Markov Chain Analyses of Random Local Search and Evolutionary Algorithm

Hiroshi Furutani*, Hiroki Tagami and Makoto Sakamoto

*Faculty of Engineering, University of Miyazaki, Gakuen Kibanadai-nishi 1-1
Miyazaki City, Miyazaki Prefecture, 889-2192, Japan*

Yifei Du

*Graduate School of Engineering, University of Miyazaki, Gakuen Kibanadai-nishi 1-1
Miyazaki City, Miyazaki Prefecture, 889-2192, Japan
E-mail: furutani@cs.miyazaki-u.ac.jp*

Abstract

Theoretical studies of evolutionary algorithms (EAs) have been developed by researchers whose main interests are convergence properties of algorithms. In this paper, we report the computational complexity of an algorithm that is a variant of $(1 + 1)$ EA, called Random Local Search (RLS). While a standard EA uses a mutation of flipping each bit in a parent string, RLS flips exactly one bit at each step. It has been noted the close resemblance of RLS with the coupon collector problem (CCP). CCP has a long history of probabilistic research, and many interesting results are obtained. This study makes use of such results with some modifications. We also show some useful results representing the evolution process of $(1 + 1)$ EA.

Keywords: Evolutionary algorithm, Random Local Search, Coupon collector's problem, Markov chain.

1. Introduction

Theoretical studies of evolutionary algorithms (EAs) have been developed by researchers whose main interests are convergence properties of algorithms [1,2]. Especially, the time complexity of algorithm is a most attracting topic for them. For the runtime analysis of EAs, one usually chooses $(1 + 1)$ EA as the first trial. A detailed overview of these studies are presented in [3].

In this paper, we report the computational complexity of an algorithm called Random Local Search (RLS), which is a variation of $(1 + 1)$ EA. While a standard EA uses a mutation of flipping each bit in a parent string, RLS flips exactly one bit at each step. It has been noted the close resemblance of RLS with the coupon collector's problem (CCP). The CCP has a long history of probabilistic research, and many interesting results are obtained [4]. This study makes use of such results with some modifications. We show some useful theorems representing the evolutionary process of RLS.

A general framework for analyzing the average hitting times of EAs by applying Markov chain model is

studied by many researchers. The state space S of the chain is given by the number of bit ones, $\{0, 1, \dots, l\}$, where l is the length of bit string. The behavior of this Markov chain is completely determined by the transition matrix \mathbf{P} of an absorbing Markov chain [5]. The explicit form of absorbing Markov chain can be obtained. Using this expression of \mathbf{P} , we show that the expected runtime of RLS algorithm is lH_1 steps, where H_1 is the first Harmonic number. The runtime of this chain is approximately given by $l \log(l)$. Furthermore, the distribution of runtime T is also obtained, and can be expressed in the closed form by using Stirling number of the second kind. There are many such interesting results for the RLS algorithm. The paper reports some of them.

2. Evolutionary Algorithms

Evolutionary Algorithm $(\mu + \lambda)$ came from Evolutionary Strategy developed by Rechenberg and Schwefel, where μ and λ are numbers of parent solutions and offspring solutions, respectively [6]. We choose the parameters $\mu = 1$ and $\lambda = 1$ for simplicity.

Though $(1 + 1)$ EA seems too simple for an analysis, it has attracting properties. As Wegener states that [2],

- (1). it is efficient for many problems,
- (2). it can not get stuck in a local optimum, and
- (3). the analysis of it reveals many tools that can be used in more practical algorithms.

Algorithm 1 $(1 + 1)$ EA

- 1: Initialize $x \in \{0,1\}^l$ uniformly at random.
 - 2: Create x' by flipping each bit in x with probability p_m .
 - 3: Select if $f(x') \geq f(x)$ then $x := x'$.
 - 4: Go to 2 until a termination condition is fulfilled.
-

As a test function, we adopt OneMax function $f(x)$

$$f(x) = \sum_{i=1}^l x_i, \quad x_i \in \{0,1\},$$

where x is a binary string of length l . We consider the maximization problem of OneMax function. The optimum solution is $x_{opt} = \{1\}^l$, and $f(x_{opt}) = l$. Since many studies suggested that the mutation probability of $p_m = 1/l$ may be the best choice, we carried out our analysis using this value.

The second choice of Evolutionary Algorithm is the Random Local Search (RLS). We define RLS as

Algorithm 2 Random Local Search

- 1: Initialize $x \in \{0,1\}^l$ uniformly at random.
 - 2: Create x' by flipping one bit in x which is selected at random.
 - 3: Select if $f(x') \geq f(x)$ then $x := x'$.
 - 4: Go to 2 until a termination condition is fulfilled.
-

The difference of two algorithms are in the step 2.

3. Coupon Collector's Problem

Consider the case that there are c types of coupons, and each day a collector randomly gets one coupon with the equal probability $1/c$. How many days has the collector to wait for collecting all types of coupons.

As an elementary problem, we consider the number of days n until both of the 2 types of coupons, A and B,

are collected. First, we calculate the expectation value $E[n]$. If two types of coupons are collected at the n th day, there are two cases,

$$\underbrace{AAA \cdots A}_{n-1} B, \quad \underbrace{BBB \cdots B}_{n-1} A.$$

Since there are n^2 cases, the probability of collecting one set of coupons is

$$p^{(2)}(n) = \frac{2}{2^n} = \frac{1}{2^{n-1}}.$$

The expectation of n is given by

$$E[n] = \sum_{n=2}^{\infty} n \left(\frac{1}{2}\right)^{n-1} = \frac{1}{(1 - 1/2)^2} - 1 = 3.$$

Next, we try to solve this problem in another approach. First, we calculate the probability $q^{(2)}(n)$ of collecting one set of coupons until n days. Among all events in n days, there are two events which do not collect at least one of two coupons,

$$\underbrace{AAA \cdots A}_n, \quad \underbrace{BBB \cdots B}_n.$$

Hence, we have

$$q^{(2)}(n) = 1 - \frac{2}{2^n} = 1 - \frac{1}{2^{n-1}}.$$

Using this result, we have

$$p^{(2)}(n) = q^{(2)}(n) - q^{(2)}(n-1) = \frac{1}{2^{n-1}}.$$

Then, we consider the case of c coupons. Let A_i be the event that the i th coupon is still not collected at the n th day. Probabilities $P(A_i)$ and $P(A_i \cap A_j)$ with $(i \neq j)$ are given by

$$P(A_i) = \left(1 - \frac{1}{c}\right)^n, \quad P(A_i \cap A_j) = \left(1 - \frac{2}{c}\right)^n.$$

In general, the probability of the event that the coupons i_1, i_2, \dots, i_k are still not collected at the n th day is

$$P(A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}) = \left(1 - \frac{k}{c}\right)^n.$$

We define events F and G . Event F is the case of failure that at least one coupon is still not collected at the n th day, while G is the case of success that a collector gets all coupons until the n th day. Since the relation

$$P(G) = 1 - P(F),$$

holds, we calculate the failure probability $P(F)$ instead of of the success probability $P(G)$. Event F is given by

$$F = A_1 \cup A_2 \cup \dots \cup A_c = \bigcup_{i=1}^c A_i.$$

Using the principle of inclusion-exclusion

$$P\left(\bigcup_{i=1}^c A_i\right) = \sum_{i=1}^c P(A_i) - \sum_{i_1 < i_2} P(A_{i_1} \cap A_{i_2}) + \dots + (-1)^{c+1} P(A_1 \cap \dots \cap A_c).$$

Thus we have the probability of collecting all types of coupons until n th day

$$q^{(c)}(n) = 1 - P(F) = \sum_{i=0}^c \binom{c}{i} (-1)^i \left(1 - \frac{i}{c}\right)^n. \quad (1)$$

The final result of the probability of collecting c types of coupons at the n th day is

$$\begin{aligned} p^{(c)}(n) &= q^{(c)}(n) - q^{(c)}(n-1) \\ &= \sum_{i=0}^{c-1} \binom{c-1}{i} (-1)^i \left(1 - \frac{i}{c}\right)^{n-1}. \end{aligned} \quad (2)$$

4. Markov Chain Model

This section presents the Markov chain approaches to the EAs.

4.1. (1 + 1)Evolutionary algorithm

The search space of OneMax function is $\Omega = \{0,1\}^l$, and we divide Ω into $(l+1)$ subsets $\Omega = S_0 \cup S_1 \cup \dots \cup S_l$, where $f(S_i) = i$. The transition matrix $P_{i,j} = P(j|i)$ represents the evolution of (1 + 1) EA,

$$\begin{aligned} (1). \text{ for } i > j & \\ P_{i,j} &= 0. \\ (2). \text{ for } i < j & \\ P_{i,j} &= \sum_{r=0}^k \binom{i}{r} p_m^r (1-p_m)^{i-r} \binom{l-i}{j-i+r} \\ &\quad \times p_m^{j-i+r} (1-p_m)^{l-i-(j-i+r)} \\ &= \sum_{r=0}^k \binom{i}{r} \binom{l-i}{s} p_m^{r+s} (1-p_m)^{l-(r+s)}, \end{aligned} \quad (3)$$

where

$$s = j - i + r, \quad k = \min(l-j, i).$$

$$(3). \text{ for } i = j \\ P_{i,i} = 1 - \sum_{j>i} P_{i,j}.$$

In a matrix form, these equations are given by

$$P = \begin{pmatrix} P_{0,0} & P_{0,1} & \dots & \dots & P_{0,l} \\ 0 & P_{1,1} & \dots & \dots & P_{1,l} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad (4)$$

The form of P suggests that this Markov chain is absorbing, and there are l transient and one absorbing states, respectively. The canonical form of the transition matrix is given by

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix} \quad (5)$$

The $l \times l$ submatrix Q represents the transitions among transient states S_0, S_1, \dots, S_{l-1} . The unit matrix I in this case is a scalar 1. The $l \times 1$ submatrix R is the probability of the transitions from transient states to the absorbing state S_l ,

$$R = (p_{0,l}, p_{1,l}, \dots, p_{l-1,l})^T.$$

For the calculation of the hitting time of the optimum solution, we use the fundamental matrix

$$N = (I - Q)^{-1}, \quad (6)$$

where the size of I is the same as Q . The Markov chain theory tells that the expected steps to enter into the absorbing state(s) are given by [5]

$$m = N\mathbf{1},$$

where m_i is the expected absorbing steps started from S_i , and $\mathbf{1}$ is a column vector whose all entries are 1. By solving a set of linear equations

$$(I - Q)m = \mathbf{1}, \quad (7)$$

we obtain the solutions

$$m_{l-1} = 1/p_{l-1,l},$$

and for $(0 \leq i \leq l-1)$

$$m_i = (1 + \sum_{j=i+1}^{l-1} P_{i,j} m_j) / (\sum_{j=i+1}^l P_{i,j}).$$

4.2. RLS algorithm

The transition matrix for the RLS process is given by

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1/l & (l-1)/l & 0 & \dots & 0 \\ 0 & 0 & 2/l & (l-2)/l & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & (l-1)/l & 1/l \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad (8)$$

From eq.(7)

$$\frac{1-i}{l}(m_i - m_{i+1}) = 1, \quad (0 \leq i \leq l-1)$$

$$\frac{1}{l}m_{l-1} = 1,$$

we have a set of solutions

$$m_i = l \left(1 + \frac{1}{2} + \dots + \frac{1}{l-i} \right). \quad (9)$$

Thus the expected first hitting time of the optimum solution in RLS is given by eq.(9) for i , the number of bit ones in the initial state. The distribution of the first hitting time n is given by $p^{(l)}(n)$ for $i = 0$ using eq. (2).

5. Numerical Experiment

In this section, we compare results of the theoretical predictions with experiments of RLS and $(1+1)$ EA. We used the mutation rate $p_m = 1/l$ for $(1+1)$ EA. The length of string is $l = 100$, and we performed 10000 runs for each parameter set, and averaged over them.

Figure 1 shows the time dependence of probability $p^{(l)}(n)$ of the first hitting time of optimum solution in RLS. The solid line is the result of numerical calculation.

The result is moving averaged with window size of 21. The dotted line is the theoretical prediction obtained by eq.(2), which well reproduces the result of the numerical calculation.

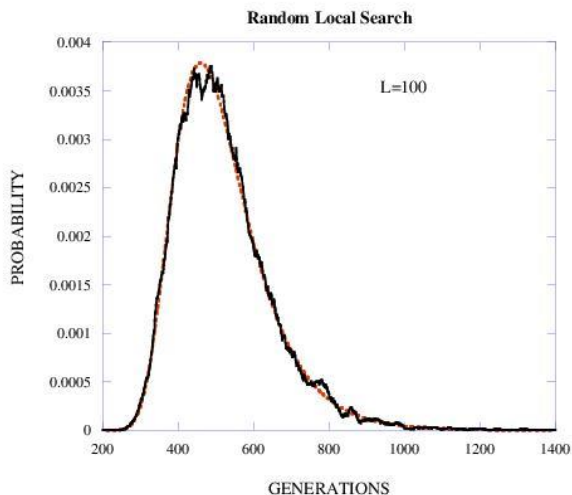


Fig. 1. Distribution of the first hitting time of optimum solution in RLS with $l = 100$. The initial state is a bit string of all zeros. The solid line is the result of RLS calculation. The dotted line is the theoretical prediction.

Figure 2 shows the time dependence of probability $p^{(l)}(n)$ of the first hitting time of optimum solution in $(1+1)$ EA. The solid line is the result of numerical EA calculation. The dotted line is the theoretical prediction obtained by using Markov chain transition matrix. The present model using Markov chain also can reproduce the numerical experiment of EA.

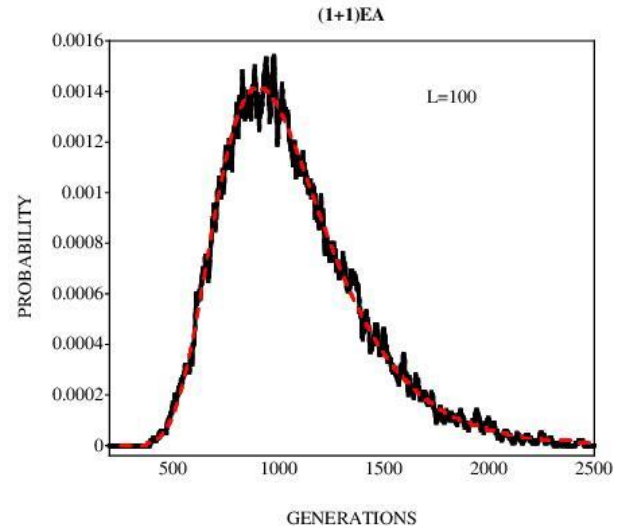


Fig. 2. Distribution of the first hitting time of optimum solution in $(1+1)$ EA with $l = 100$. The initial state is a bit string of $l/2$ zeros and $l/2$ ones. The solid line is the result of $(1+1)$ EA calculation. The dotted line is the theoretical prediction using Markov chain model.

6. Summary

In this paper, we demonstrate that the probabilistic methods including Markov chain model can reproduce the first hitting time of optimum solution in RLS and $(1+1)$ EA for OneMax function. These studies can help to understand the working mechanism of Evolutionary Algorithms, and give some suggestions to design algorithms for other problems. We cannot show results for linear functions obtained by present algorithms and methods of population genetics [7] due to limitation of space. We will report them in other occasions.

Our next aim is to apply drift analysis to optimization problems. Drift analysis was a class of tools developed by Yao and other researchers [3,8], and can give upper and lower bounds to running time of a given algorithm.

We are now trying to combine drift analysis and classical probabilistic methods.

References

1. A. Auger and B. Doerr (eds.) (2011), *Theory of Randomized Search Heuristics*, World Scientific, Singapore.
2. I. Wegener (2001), Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions, in *Evolutionary Optimization*, Kluwer Academic Publishers, Dordrecht, pp.349-369.
3. P. S. Oliveto and X. Yao (2011), Runtime analysis of evolutionary algorithms for discrete optimization, in [1], pp.21-52.
4. T. Nakata and I. Kubo (2006), A coupon collector's problem with bonuses, in *Proceedings of Discrete Mathematics and Theoretical Computer Science*, **AG**, pp.215-224.
5. M. Iosifescu (1980), *Finite Markov chain processes and their applications*, John Wiley & Sons, New York.
6. H.-G. Beyer and H.-P. Schwefel (2002), Evolution Strategies: A comprehensive introduction, *Natural Computing*, **1**, pp.3-52.
7. J. W. J. Ewens (2004), *Mathematical Population Genetics. I. Theoretical Introduction*, Second Edition. Springer-Verlag, New York.
8. B. Hajek (1982), Hitting-time and occupation-time bounds implied by drift analysis with applications, *Advances in Applied Probability* **13**, pp.502-525.