Drift Conditions for Time Complexity of Evolutionary Algorithms*

HE Jun[1], YAO Xin[1], KANG Li-shan[2]

[1](School of Computer Science, The University of Birmingham, Birmingham B15 2TT, UK);
[2](State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)
E-mail: kang@whu.edu.cn
http://www.whu.edu.cn

Received September 11, 2000; accepted April 17, 2001

Abstract: The computational time complexity is an important topic in the theory of evolutionary algorithms. This paper introduces drift analysis into analysing the average time complexity of evolutionary algorithms, which are applicable to a wide range of evolutionary algorithms and many problems. Based on the drift analysis, some useful drift conditions to determine the time complexity of evolutionary algorithms are studied. These conditions are applied into the fully deceptive problem to verify their efficiency.

Key words: evolutionary algorithms; time complexity; Markov chain; combinatorial optimisation

Evolutionary algorithms (EAs) are a powerful class of adaptive search algorithms. They have been used to solve many combinatorial problems with success in recent years. However, theories on explaining why and how EAs work are still relatively few[1,2]. The computational time complexity of EAs is largely unknown, except for a few simple cases[2~6]. Back[7] and Mühlenbein[8] made the first step in this direction. Ambati *et al.*[3] and Fogel[4] discussed the time complexity of travelling salesman problem by simulated evolution but without theoretical analysis. Aytug and Koehler[9], Hulin[10] estimated the computation time by studying stopping criterion. Rudolph[5,11] proved that (1+1) EAs with mutation probability $p_m=1/n$, where n is the number of bits in a binary string (i.e., individual), converge in average time $O(n\log n)$ for the ONE-MAX function, but didn't analyse other linear functions. Droste and others[6] made a rigorous complexity analysis of EAs for linear functions with Boolean inputs. It has been shown that GAs may take an exponential average time for some deceptive problems[12]. Rudolph[2] made a survey on this topic and pointed out that present theoretical studies are restricted to certain problems classes and simple EAs yet.

This paper presents a more general theory about the average time complexity of EAs. The motivation of this study is to establish a general theory for a class of EAs, rather than a particular EA. The theory should then be used to derive specific complexity results for different EAs on different problems. The theory given in this paper uses drift analysis[13,14], which can estimate the first hitting time by analysing the drift of a Markov Chain. Sasaki

* Supported by the National Natural Science Foundation of China under Grant Nos. 60133010, 60073043 and 70071042 (国家自然科学基金); the Research Foundation of the State Key Laboratory of Software Engineering at Wuhan University of China (武汉大学软件工程国家重点实验室基金)

HE Jun was born in 1967. His research interests are parallel computation and evolutionary computation. **YAO Xin** was born in 1962. His research interests are parallel computation and evolutionary computation. **KANG Li-shan** was born in 1934. He is a professor and doctoral supervisor of the Wuhan University. His current reserach areas are paralle computation and evolutionary computation.

and Hajek once used this method to analyse the time complexity of maximum matching problem by simulated annealing[15]. The paper shows that it is possible to provide general framework for the finite time behaviour of EAs, not as the statement in Rudolph[2]: the examination of the finite time behaviour of EAs cannot be treated in the same general manner as it is possible for the limit behaviour.

The state evolution of an EA population can be modelled by a Markov Chain[16]. By analysing its drift, it is possible to estimate the first hitting time to the optimal solution. In this way, some drift conditions will be useful to estimate the average first hitting time to the optimal solution and then the average time complexity of EAs is obtained. In this paper we apply such drift conditions to the fully deceptive problem and verify these conditions. More examples, including revisited discussion on problems and results appeared in Rudolph's survey[2], can be found in our recently paper[17].

1 Evolutionary Algorithms and Drift Analysis

1.1 Evolutionary algorithms

The combinatorial problem considered in this paper can be formalised as follows: Given a finite state space S and a bounded function $f(x)$, $x\in S$, find

$$\max\{f(x);\ x\in S\}. \tag{1}$$

Let $f_{\max}=\max\{f(x);\ x\in S\}$ be the maximum value and x^* be any optimal point with $f(x^*)=f_{\max}$.

In this paper, the space S is called the individual space where each $x\in S$ is called the individual, and the product space $S^{2N}=S\times\ldots\times S$ is called the population space, where $\eta\in S^{2N}$ is called the population, denoted by $\eta=\{x_1,\ldots,x_{2N}\}$. The fitness of an individual x is $f(x)$, and the fitness of a population η is $f(\eta)=\max\{f(x_i);\ x_i\in\eta\}$.

The EA for solving the combinatorial optimisation problem can be described as follows:

(1) Initialisation: generate, either randomly or heuristically, an initial population of $2N$ individuals $\{x_1,\ldots,x_{2N}\}$, denote the population by ξ_0, and let $k\leftarrow 0$ (where k represents the time step).

(2) Generation: generate a new (intermediate) population by a crossover and then by mutation (or only by mutation or only by crossover), denote this population by $\xi_{k+1/2}$.

(3) Selection: select $2N$ individuals from population $\xi_{k+1/2}$ (or and ξ_k) and obtain the new (intermediate) population and denote it by ξ_{k+S}.

(4) Let $\xi_{k+1}=\xi_{k+S}$ and increase the time step k by 1, $k\leftarrow k+1$, and go to Step 2.

Obviously the above description of algorithms includes a wide range of EAs using crossover, mutation and selection. The description does not set any restrictions on the type of crossover, mutation or selection schemes used. It includes EAs which do not use crossover or mutation.

No stopping rule appears in the description because in some case the first time when an EA finds an optimal solution may be infinitely long.

Many EAs can be modelled by a Markov chain $\{\xi_k;\ k=0,1,2,\ldots\}$ defined on the state space S^{2N} if the population state at time $k+1$ only depend on the state at time k[2]. And the chain will be a homogeneous Markov chain if no self-adaptation is used. In the rest of paper, EAs are assumed to be convergent under certain conditions[2,16].

1.2 Drift analysis

For the space S, we can introduce a distance function $d(x,x^*)$ to measure the distance between a point x and an optimal point x^*. As an distance function, $d(x,x^*)$ usually satisfies the following properties: $d(x,x^*)=0$ and $d(x,x^*)>0$ for any non-optimal point x. For convenience, we denote it in short $d(x)$. If the optimal points is a set, for example S^*, then $d(x)$ means $d(x,S^*)=\min\{d(x,x^*);\ x\in S^*\}$. Since in combinatorial optimisation, S is always finite, without any question, we assume $d(x)\geq 0$ and bounded.

Given a population $\eta=\{x_1,\ldots,x_{2N}\}$, denote

$$d(\eta)=\min\{d(x);x\in\eta\}, \tag{2}$$

which is used to measure the distance between a population η and the optimal point. When $d(\eta)=0$, it means that the population η has already included an optimal point. The drift of Markov chain $\{\xi_k, k=0,1,\ldots\}$ at time k is defined by

$$\Delta(d(\xi_k))=d(\xi_{k+1})-d(\xi_k)$$

which represents one-step drift at time k. This definition is a special case of the drift defined in the general space[14]. If $E[\Delta]>0$, the average drift of ξ_k will be away from the optimal point, and if $E[\Delta]<0$, the average drift of ξ_k will be close to the optimal point.

Define the first hitting time of an EA as

$$\tau=\min\{k;d(\xi_k)=0\}$$

which is the first hitting time of an EA on the optimal point. And $E[\tau]$ is the mean computational time when the EA find the optimal point. The task of time complexity of EAs now becomes to investigate the relationship between the expect first hitting time $E[\tau]$ and the space size of S, which is often represented by an integer n in combinatorial optimisation[18]. In this paper, we focus on the following question: from which kind of conditions about the drift $\Delta(d(\xi_k))$, can we estimate the expect first hitting time $E[\tau]$? In particular, we study the conditions under which an EA is guaranteed to find the optimal solution in polynomial time on average and conditions under which an EA takes at least exponential time on average to find the optimal solution.

2 Conditions for Polynomial Average Computation Time

2.1 Drift conditions

This section studies drift conditions under which an EA can solve an optimisation problem in polynomial average time.

Condition 1. There exists a polynomial of problem size n, $h_0(n)>0$, such that

$$d(\eta)\leq h_0(n)$$

for any given population η.

This condition says that the distance from any population η to the optimal solution is bounded by a polynomial of the problem size n.

Condition 2. There exists a polynomial of problem size n, $h_1(n)>0$, such that

$$E[\Delta(d(\xi_k)\mid\xi_k=\eta]\leq-\frac{1}{h_1(n)}$$

for any time k and population η with $d(\eta)>0$, where $\Delta(d(\xi_k))=d(\xi_{k+1})-d(\xi_k)$.

This condition indicates that the one-step mean drift of Markov chain $\{\xi_k;k=0,1,2,\ldots\}$ is toward the optimal point, not away from the point.

If an EA is convergent, we can assume that the expect first hitting time τ is finite, i.e. $E[\tau\mid\xi_0]<+\infty$. Now we give the our first main result.

Theorem 1. If Markov chain $\{\xi_k;\ k=0,1,2,\ldots\}$ satisfies Conditions 1 and 2, then starting from any initial population η with $d(\eta)>0$,

$$E[\tau\mid\xi_0=X]\leq h(n),$$

where $h(n)$ is a polynomial of problem size n.

Proof. According to Condition 2, we know that $\{d(\xi_k);\ k=0,1,2,\ldots\}$ in fact is a super-martingale. Since

$d(\xi_k)\leqslant h_0(n)$, it converges almost everywhere, and

$$\lim_{k\to\infty}E[d(\xi_k)\mid\xi_0]=0.$$

According to the definition of τ, $d(\xi_\tau)=0$. Hence,

$$E[d(\xi_\tau)\mid\xi_0=\eta]=0$$

for any initial population η.

For any time $k\geqslant 1$,

$$E[d(\xi_k)\mid\xi_0=\eta]=E[E[d(\xi_{k-1})+\Delta(d(\xi_{k-1}))\mid\xi_{k-1}]\mid\xi_0=\eta].$$

According to Condition 2, we have for any $k-1<\tau$,

$$E[d(\xi_{k-1})+\Delta(d(\xi_{k-1}))\mid\xi_{k-1}]\leqslant d(\xi_{k-1})-\frac{1}{h_1(n)}.$$

Therefore

$$E[d(\xi_k)\mid\xi_0=\eta]\leqslant E\left[d(\xi_{k-1})-\frac{1}{h_1(n)}\mid\xi_0=\eta\right].$$

By induction on k, we can get

$$E[d(\xi_k)\mid\xi_0=\eta]\leqslant E\left[d(\xi_0)-\frac{k}{h_1(n)}\mid\xi_0=\eta\right].$$

Hence we have

$$0=E[d(\xi_\tau)\mid\xi_0]\leqslant E\left[d(\xi_0)-\frac{\tau}{h_1(n)}\mid\xi_0=\eta\right]\leqslant E[d(\xi_0)\mid\xi_0=\eta]-\frac{1}{h_1(n)}E[\tau\mid\xi_0=\eta].$$

According to the above inequality and Condition 1,

$$E[\tau\mid\xi_0=\eta]\leqslant E[d(\eta)]h_1(n)\leqslant h_0(n)h_1(n).$$

Let $h(n)=h_0(n)h_1(n)$. We arrive at

$$E[\tau\mid\xi_0=X]\leqslant h(n).$$

where $h(n)$ is a polynomial of n.

2.2 Application in fully deceptive problem

Now we apply the above results to a genetic algorithm with multiple structures for solving the fully deceptive problem[12]. The detail description of the problem and algorithm is referred to Ref. [12], here is the algorithm:

(1) Recoding: for any individual $x=(s_1\ldots s_n)$, the new code of x is $z=(x-1)\bmod(n)$, where its fitness is $f(z)=f(x)$. Then we form a new population state space.

(2) Execute the following procedure on the new state space:

(3) Initialisation: choose $2N$ individuals as the initial population ξ_0, let time $k=0$.

(4) Crossover: using one point crossover, generate an intermediate population, denote it by ξ_{k+C}, where the subscript $k+C$ represents the crossover at time k.

(5) Selection A: select $2N$ individuals with the highest fitness from the population ξ_k and ξ_{k+C}, and form an intermediate population, denote it by $\xi_{k+1/2}$.

(6) Mutation: let $z=(s_1\ldots s_n)$ be any individual in $\xi_{k+1/2}$, choose one bit in z randomly, let it flip. Each individual mutates in this way, then generate an intermediate population, denote it by ξ_{k+M}, where the subscript $k+M$ represents the mutation at time k.

(7) Selection B: select $2N$ individuals with the highest fitness from the population $\xi_{k+1/2}$ and ξ_{k+M}, and form the next generation population, denoted by ξ_{k+1}.

(8) Increase time step k by 1: $k\leftarrow k+1$, and return to Step 4.

In the new coding structure, the optimal solution is $(1\ldots1)$. Define the distance function $d(x)$ as

$$d(x)=\sum_{i=1}^{n}|s_i-1|, \tag{3}$$

which represents the distance between x and the optimal point. It is obvious that the smaller the distance is, the higher the fitness of individual is.

Theorem 2. For any population η with $d(\eta)>0$, the average computation time $E[\tau]$ satisfies

$$E[\tau \mid \xi_0=\eta]\leqslant h(n)$$

where $h(n)$ is a polynomial of n.

Proof. According to Theorem 1, we should verify $\{d(\xi_k),k=0,1,\dots\}$ to satisfy Conditions 1 and 2.

From (3) and (2), we know for any population η:

$$d(\eta)\leqslant n.$$

Then $\{d(\xi_k);k=0,1,\dots\}$ satisfies Condition 1.

At time $k\geqslant 0$, assume $d(\xi_k)>0$, which means the population doesn't include an optimal individual yet. Let's investigate the effect of crossover on the drift. After the crossover, one of the following three events will happen: (a1) event $I\{d(\xi_{k+C})<d(\xi_k)\}$, (a2) event $I\{d(\xi_{k+C})=d(\xi_k)\}$ and (a3) event $I\{d(\xi_{k+C})>d(\xi_k)\}$.

First we prove event (a1) cannot happen. Let x_1 and x_2 be two individuals in ξ_k, y_1 and y_2 their offspring after crossover, then

$$d(y_1)+d(y_2)=d(x_1)+d(x_2).$$

This means that the increase of one individual's drift lead to the decrease of another individual's drift. Then event (a1) cannot happen.

According to Selection A, we have $d(\xi_{k+1/2})=d(\xi_{k+C})\leqslant d(\xi_k)$.

In mutation, one of the following three events may happen subsequently: (b1) event $I\{d(\xi_{k+M})<d(\xi_{k+1/2})\}$, (b2) event $I\{d(\xi_{k+M})=d(\xi_{k+1/2})\}$, and (b3) event $I\{d(\xi_{k+M})>d(\xi_{k+1/2})\}$.

Since the mutation is a simple bit-flipping, it is easy to show that the probability of event (b1) is not less than $1/n$ (if $d(\xi_{k+1/2})>0$). The probability of event (b3) is not greater than $(n-1)/n$. If $d(\xi_{k+1/2})=0$, then the population $\xi_{k+1/2}$ has included one optimal solution.

In Selection B, one of the following events may happen: (c1) event $I\{d(\xi_{k+1})<d(\xi_{k+1/2})\}$ and (c2) event $I\{d(\xi_{k+1})=d(\xi_{k+1/2})\}$. If event $I\{d(\xi_{k+M})<d(\xi_{k+1/2})\}$ happens, then event (c1) happens; if event $I\{d(\xi_{k+M})\geqslant d(\xi_{k+1/2})\}$ happens, then event (c2) will follow.

Considering all the cases discussed together, we have

$$\begin{aligned}
E[d(\xi_{k+1})-d(\xi_k)\mid d(\xi_k)>0]\leqslant{}& E[(d(\xi_{k+1})-d(\xi_k))I\{d(\xi_{k+1/2})>0\}\mid d(\xi_k)]+\\
& E[(d(\xi_{k+1})-d(\xi_k))I\{d(\xi_{k+1/2})=0\}\mid d(\xi_k)]\\
\leqslant{}& E[(d(\xi_{k+1})-d(\xi_k))I\{d(\xi_{k+1/2})>0,d(\xi_{k+M})\\
&<d(\xi_{k+1/2}),d(\xi_{k+1})<d(\xi_{k+1/2})\}\mid d(\xi_k)>0]+\\
& E[(d(\xi_{k+1})-d(\xi_k))I\{d(\xi_{k+1/2})>0,d(\xi_{k+M})\\
&\geqslant d(\xi_{k+1/2}),d(\xi_{k+1})=d(\xi_{k+1/2})\}\mid d(\xi_k)>0]+\\
& E[(d(\xi_{k+1})-d(\xi_k))I\{d(\xi_{k+1/2})=0,d(\xi_{k+1})=0\}\mid d(\xi_k)>0]\leqslant -1/n.
\end{aligned}$$

Then we prove $\{d(\xi_k),k=0,1,\dots\}$ satisfies Condition 2. According to Theorem 1, we know

$$E[\tau \mid \xi_0=\eta]\leqslant h(n)$$

where $h(n)$ is a polynomial of n. □

But this estimation is worse than the result given in Ref.[12]. So further strict drift conditions should be studied for obtaining tighter bounds.

3 Drift Conditions for Exponential Average Computation Time

3.1 Drift conditions

In this subsection, we investigate the drift conditions under which EAs will take the average time exponential in the problem size n to find the optimal solution. Our analysis is based on Hajek's earlier work[13].

For a distance function $d(x)$, we define two first hitting times τ and τ': $\tau=\inf\{k: d(\xi_k)=0\}$ and $\tau'=\inf\{k: d(\xi_k)\leqslant d_b\}$ where $d_b\geqslant 0$ and define $\tau'=\inf\{\varnothing\}=+\infty$. It is obvious that $E[\tau]\geqslant E[\tau']$. Then if we want to prove that $E[\tau]$ is an exponential function of n, we only need to prove that $E[\tau']$ is for some $d_b>0$.

Condition 3. At time $k\geqslant 0$, for any population η with $d_b<d(\eta)<d_a$, where $d_b\geqslant 0$ and $d_a>0$,

$$E[e^{-(d(\xi_{k+1})-d(\xi_k))}\mid\xi_k=\eta, d_b<d(\xi_k)<d_a]\leqslant\rho<1, \tag{4}$$

where $\rho>0$ is a constant.

Condition 4. At time $k\geqslant 0$, for any population η with $d(\eta)\geqslant d_a, d_a>0$,

$$E[e^{-(d(\xi_{k+1})-d_a)}\mid\xi_k=\eta, d(\xi_k)\geqslant d_a]\leqslant D, \tag{5}$$

where $D\geqslant 1$ is a constant.

Under these two conditions, the following two theorems can be shown by following Hajek's work on drift analysis[13].

Theorem 3. If Markov chain $\{\xi_k; k=0,1,\ldots\}$ satisfies Conditions 3 and 4, then for any initial population ξ_0,

$$E[e^{-d(\xi_k)}\mid\tau'>k-1, d(\xi_0)]\leqslant\rho^k e^{-d(\xi_0)}+\frac{1-\rho^k}{1-\rho}De^{-d_a}, \tag{6}$$

and

$$P[d(\xi_k)\leqslant d_b\mid\tau'>k-1, d(\xi_0)]\leqslant\rho^k e^{-(d(\xi_0)-d_b)}+\frac{1-\rho^k}{1-\rho}De^{-(d_a-d_b)}. \tag{7}$$

Proof. Inequality (6) is clearly true for $k=0$.

For $k>0$ and $\tau'>k$,

$$E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_0)]=E[E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_k)]\mid d(\xi_0)].$$

Now

$$E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_k)]=E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_k)\geqslant d_a]+E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_k)<d_a]. \tag{8}$$

The first term on the right-hand side of Inequality (8) is upper bounded by De^{-d_a} according to Condition 4, and the second term is upper-bounded by $\rho e^{-d(\xi_k)}$ according to Condition 3. Using these bounds we can arrive at

$$E[e^{-d(\xi_{k+1})}\mid\tau'>k, d(\xi_0)]\leqslant\rho E[e^{-d(\xi_k)}\mid\tau'>k-1, d(\xi_0)]+De^{-d_a}.$$

By induction on k, it is easy to show that the above inequality implies Inequality (6) for all $k\geqslant 0$. Inequality (7) follows from Inequality (6) by Chebyshev's inequality. In other words,

$$\begin{aligned}E[e^{-(d(\xi_k)-d_b)}\mid\tau'>k-1, d(\xi_0)]&=E[E[e^{-(d(\xi_k)-d_b)}\mid\tau'>k-1, d(\xi_k)\leqslant d_b]\mid d(\xi_0)]+\\&\quad E[E[e^{-(d(\xi_k)-d_b)}\mid\tau'>k-1, d(\xi_k)>d_b]\mid d(\xi_0)]\\&\geqslant E[E[e^{-(d(\xi_k)-d_b)}\mid\tau'>k-1, d(\xi_k)\leqslant d_b]\mid d(\xi_0)]\\&\geqslant e^0 P(d(\xi_k)\leqslant d_b\mid\tau'>k-1, d(\xi_0)).\end{aligned}$$

Hence

$$P(d(\xi_k)\leqslant d_b\mid\tau'>k-1, d(\xi_0))\leqslant E[e^{-(d(\xi_k)-d_b)}\mid\tau'>k-1, d(\xi_0)].$$

Then from the inequality (6) we have

$$P(d(\xi_k)\leqslant d_b\mid d(\xi_0)\geqslant d_a)\leqslant\rho^k e^{-(d(\xi_0)-d_b)}+\frac{1-\rho^k}{1-\rho}De^{-(d_a-d_b)}.$$ □

Theorem 4. Assume Conditions 3 and 4 hold. If $d(\xi_0)\geqslant d_a$, $D\geqslant 1$ and $\rho<1$, then there exist some $\delta_1>0$ and $\delta_2>0$ such that

$$E[\tau'\mid d(\xi_0)\geqslant d_a]\geqslant\delta_1 e^{\delta_2(d_a-d_b)} \tag{9}$$

Proof. Because $d(\xi_0)\geqslant d_a$, we have

$$e^{-(d(\xi_0)-d_b)}\leqslant e^{-(d_a-d_b)}.$$

Since $D\geqslant 1$ and $\rho<1$, we can obtain

$$\rho^k e^{-(d(\xi_0)-d_b)}\leqslant\frac{\rho^k}{1-\rho}De^{-(d_a-d_b)}.$$

According to Inequality (7) and the above inequality,

$$P(d(\xi_k)\leqslant d_b \mid \tau'>k-1, d(\xi_0))\leqslant \frac{1}{1-\rho}De^{-(d_a-d_b)}.$$

Assume $P(\tau'<+\infty \mid d(\xi_0))=1$. By using the fact that $P(\tau'=k \mid d(\xi_0))=P(d(\xi_k)\leqslant d_b \mid \tau'>k-1, d(\xi_0))$, we have

$$P(\tau'>k \mid d(\xi_0))=1-\sum_{j=1}^{k}P(\tau'=j \mid d(\xi_0))\geqslant \max\left(0, 1-k\frac{De^{-(d_a-d_b)}}{1-\rho}\right).$$

Therefore

$$E[\tau' \mid d(\xi_0)]=\sum_{j=1}^{+\infty}P(\tau'>j \mid d(\xi_0))\geqslant \sum_{k=0}^{+\infty}\max\left(0, 1-k\frac{De^{-(d_a-d_b)}}{1-\rho}\right)\geqslant \frac{1-\rho}{2D}e^{(d_a-d_b)}.$$

Let $\delta_1=\frac{1-\rho}{2D}$ and $\delta_2=1$ then

$$E[\tau' \mid d(\xi_0)\geqslant d_a]\geqslant \delta_1 e^{\delta_2(d_a-d_b)}$$

and then $E[\tau \mid d(\xi_0)\geqslant d_a]\geqslant \delta_1 e^{\delta_2(d_a-d_b)}$. □

3.2 Application in fully deceptive problem

This subsection will discuss another genetic algorithm given in Ref. [12] for fully deceptive problem. The algorithm is:

(1) Initialisation: choose $2N$ individuals as the initial population ξ_0, let time step $k=0$.

(2) Crossover: using one-point crossover, generate an intermediate population, and denote it by ξ_{k+C}.

(3) Mutation: let x be an individual in ξ_{k+C}, then each bit in x flip with probability p (where $0<p<e^{-(1+\varepsilon)}$, ε is a small positive). Each individual will mutate in the same way, then generate an intermediate population, and denote it by ξ_{k+M}.

(4) Selection: select $2N$ individuals with the highest fitness from the population ξ_k and ξ_{k+M}, and form the next generation population, denote it by ξ_{k+1}.

(5) Increase time step k by 1: $k\leftarrow k+1$, and return to Step 2.

Define the distance function by

$$d(x)=\left|\sum_{i=1}^{n}s_i-0\right|.$$

If two individuals x_1 and x_2 satisfies $d(x_1)>d(x_2)>0$, then the fitness of x_1 is higher than that of x_2[12].

Define $\tau=\min\{k; d(\xi_k)=0\}$, let $d_a=3n/4$, $d_b=5n/8$ and define $\tau'=\min\{k; d(\xi_k)\leqslant d_b\}$.

Theorem 5. For Markov chain $\{\xi_k; k=0,1,\dots\}$, if $d(\xi_0)\geqslant d_a$, then there are $\delta_1>0$ and $\delta_2>0$:

$$E[\tau' \mid d(\xi_0)\geqslant d_a]\geqslant \delta_1\exp(\delta_2 n)$$

for enough large n.

Proof. According to Theorem 4, it is needed to verify that Conditions 3 and 4 hold.

Let's verify Condition 3 first. Assume $d_b\leqslant d(\xi_k)<d_a$ at time k.

After crossover, three events may happen: (a1) event $I\{d(\xi_{k+C})<d(\xi_k)\}$, (a2) event $I\{d(\xi_{k+C})=d(\xi_k)\}$ or (a3) event $I\{d(\xi_{k+C})>d(\xi_k)\}$. Like the analysis in Theorem 2, event (a3) never happens. After crossover, we have $d(\xi_{k+C})\geqslant n/8$ since $d(\xi_k)\geqslant 5n/8$.

After mutation, two events may happen: (b1) event $I\{d(\xi_{k+M})=0\}$ or (b2) event $I\{d(\xi_{k+M})>0\}$, where the probability of event (b1) is not greater than $p^{n/8}$, and that of event (b2) is not less than $1-p^{n/8}$. Furthermore, let y be the offspring of x after mutation, then the probability of event $I\{d(y)=0 \mid d(x)>0\}$ is $p^{d(x)}(1-p)^{n-d(x)}$. Then

$$e^{d(x)}P(d(y)=0 \mid d(x)>0)=e^{d(x)}p^{d(x)}(1-p)^{n-d(x)}\leqslant O(e^{-\varepsilon n/8}).$$

Here we use assumption $p<e^{-(1+\varepsilon)}$ where $\varepsilon>0$.

Now let's investigate the effect of selection on the drift. Since the selection is a $(2N+2N)$ elite strategy, then there will be two possible events: (c1) if event $I\{d(\xi_{k+M})=0\}$ happens, then event $\{d(\xi_{k+1})=0\}$ follows; (c2) if event $I\{d(\xi_{k+M})>0\}$ happens, then event $\{d(\xi_{k+1})>d(\xi_k)\}$ follows.

Summarizing all the above events, we have

$$\begin{aligned}E[e^{-(d(\xi_{k+1})-d(\xi_k))}\mid d(\xi_k)>d_b] = {} & E[e^{-(d(\xi_{k+1})-d(\xi_k))}I\{d(\xi_{k+C})<d(\xi_k),d(\xi_{k+M})=0,d(\xi_{k+1})=0\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-d(\xi_k))}I\{d(\xi_{k+C})<d(\xi_k),d(\xi_{k+M})>0,d(\xi_{k+1})>d(\xi_k)\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-d(\xi_k))}I\{d(\xi_{k+C})=d(\xi_k),d(\xi_{k+M})=0,d(\xi_{k+1})=0\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-d(\xi_k))}I\{d(\xi_{k+C})=d(\xi_k),d(\xi_{k+M})>0,d(\xi_{k+1})=d(\xi_k)\}\mid d(\xi_k)>d_b]\\ \leqslant {} & O(e^{-cn/8})+e^{-1}.\end{aligned}$$

Since $\lim_{n\to+\infty}O(e^{-cn/8})+\exp(-1)=\exp(-1)<1$, then for enough large n, there is a positive $\rho<1$:

$$E[e^{-(d(\xi_{k+1})-d(\xi_k))}\mid d(\xi_k)>d_b]<\rho<1.$$

This means $\{\xi_k;k=0,1,\ldots\}$ satisfies Condition 3.

Next we prove Condition 4 holds. Assume $d(\xi_k)\geqslant d_a$ at time k.

After crossover, one of the two event will happens: (A1) event $I\{d(\xi_{k+C})<d(\xi_k)\}$ and event (A2) $I\{d(\xi_{k+C})=d(\xi_k)\}$. After crossover, $d(\xi_{k+C})\geqslant n/4$ since $d(\xi_k)\geqslant 3n/4$.

After mutation, one of two events will happens: (B1) event $I\{d(\xi_{k+M})=0\}$ and (B2) event $I\{d(\xi_{k+M})>0\}$, where the probability of event (B1) is not greater than $p^{n/4}$, and that of (B2) is not less than $1-p^{n/4}$. Let y be the offspring of x after mutation, like the previous analysis, we have $e^{d(x)}P(d(y)=0\mid d(x)>0)\leqslant O(e^{-cn/4})$.

Let's consider the effect of selection on the drift: (C1) if event $I\{d(\xi_{k+M})=0\}$ happens, then event $\{d(\xi_{k+1})=0\}$ follows; (C2) if event $I\{d(\xi_{k+M})>0\}$ happens, then event $\{d(\xi_{k+1})\geqslant d(\xi_k)\}$ follows.

Summarising all the above events, we have

$$\begin{aligned}E[e^{-(d(\xi_{k+1})-3n/4)}\mid d(\xi_k)>d_b] = {} & E[e^{-(d(\xi_{k+1})-3n/4)}I\{d(\xi_{k+C})<d(\xi_k),d(\xi_{k+M})=0,d(\xi_{k+1})=0\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-3n/4)}I\{d(\xi_{k+C})<d(\xi_k),d(\xi_{k+M})>0,d(\xi_{k+1})\geqslant d(\xi_k)\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-3n/4)}I\{d(\xi_{k+C})=d(\xi_k),d(\xi_{k+M})=0,d(\xi_{k+1})=0\}\mid d(\xi_k)>d_b]+\\ & E[e^{-(d(\xi_{k+1})-3n/4)}I\{d(\xi_{k+C})=d(\xi_k),d(\xi_{k+M})>0,d(\xi_{k+1})\geqslant d(\xi_k)\}\mid d(\xi_k)>d_b]\\ \leqslant {} & O(e^{-cn/4})+2.\end{aligned}$$

Let $D=3$, then Conditions 4 holds.

From Theorem 4 and the fact $d_a-d_b=3n/4-5n/8=n/8$, we have

$$E[\tau\mid d(\xi_0)\geqslant d_a]\geqslant E[\tau'\mid d(\xi_0)\geqslant d_a]\geqslant\delta_1 e^{\delta_2 n},$$

where δ_1 and δ_2 are positive constants. □

4 Conclusions and Further Work

In this paper we have introduced drift analysis into analysing the average time complexity of EAs. Drift analysis is a useful technique for estimating the bounds of EA's average computation time. It does not estimate the first hitting time directly, but estimate the drift, which is easier to implement in some case.

Using drift analysis, we have shown a couple of important theorems. Theorem 1 gives some general conditions under which an EA can solve a problem in polynomial time on average. Theorem 4 gives some general conditions under which an EA needs at least exponential computation time on average to solve a problem.

In the paper, we have also applied these theorems to fully deceptive problem and obtained similar results on time bound of EAs as in paper [12].

The future work of this study includes: to discuss more EAs for other combinatorial optimisation problems by drift analysis; to study more strict drift conditions in order to obtain tighter upper-bounds of the average computation time.

References:

[1] Eiben, A.E., Rudolph, G. Theory of evolutionary algorithms: a bird eye view. Theoretical Computer Science, 1999,229(1/2):3~9.

[2] Rudolph, G. Finite Markov chain results in evolutionary computation: A tour d'Horizon. Fundamenta Informaticae, 1998, 35(1~4):67~89.

[3] Ambati, B.K., Ambati, J., Mokhtar, M.M. Heuristic combinatorial optimization by simulated Darwinian evolution: a polynomial time algorithm for the travelling salesman problem. Biological Cybernetics, 1991,65(1):31~35.

[4] Fogel, D.B. Empirical estimation of the computation required to discover approximate solutions to the travelling salesman problem using evolutionary programming. In: Fogel, D.B., Atmar, A., eds. Proceedings of the Second Ann. Conference on Evolutionary Programming. La Jolla, CA: Evolutionary Programming Society. 1993. 56~61.

[5] Rudolph, G. Convergence Properties of Evolutionary Algorithms. Kovač, Hamburg, Germany, 1997.

[6] Droste, S., Jansen, T., Wegener, I. A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with boolean inputs. Evolutionary Computation, 1998,6(2):185~196.

[7] Bäck, T. The interaction of mutation rate, selection and self adaptation within a genetic algorithm. In: Männer, R., Manderick, B., eds. Parallel Problem Solving from Nature 2. Amsterdam, North Holland. 1992. 85~94.

[8] Mühlenbein, H. How genetic algorithms really works I: Mutation and hill-climbing. In: Männer, R., Manderick, B., eds. Parallel Problem Solving from Nature 2. Amsterdam, North Holland. 1992. 15~25.

[9] Aytug, H., Koehler, J.C. Stopping criteria for finite length genetic algorithms. Informs Journals on Computing, 1996,8(2):183~191.

[10] Hulin, M. An optimal stopping criteria for genetic algorithms: A Bayesian approach. In: Bäck, T., ed. Proceedings of the 7th Conference on Genetic Algorithms. San Francisco: Morgan Kaufmann. 1997. 135~142.

[11] Rudolph, G. How mutation and selection solve long path problems in polynomial expected time. Evolutionary Computation, 1996,4(2):207~211.

[12] He, J., Huang, H., Kang, L. The computational time of genetic algorithms for fully deceptive problem. Chinese Journal of Computers, 1999,21(9):999~1003 (in Chinese).

[13] Hajek, B. Hitting time and occupation time bounds implied by drift analysis with applications. Adv. Appl. Probab., 1982, 14:502~525.

[14] Meyn, S.P., Tweedie, R.L. Markov chains and Stochastic Stability. New York: Springer Verlag, 1993.

[15] Sasaki, G.H., Hajek, B. The time complexity of maximum matching by simulated annealing. Journal of the ACM, 1988, 35(2):387~403.

[16] He, J., Kang, L. On the convergence rate of genetic algorithms. Theoretical Computer Science, 1999,229(1/2):23~39.

[17] He, J., Yao, X. Drift analysis and average time complexity of evolutionary algorithms. Artificial Intelligence, 2001,127(1):57~85.

[18] Papadimitriou, C.H., Steiglitz, K. Combinatorial Optimization: Algorithms and Complexity. Dover, Mineola, New York, 1998.

附中文参考文献:

[12] 何军,黄厚宽,康立山.演化算法求解完全欺骗性问题的平均计算时间.计算机学报,1999,21(9):999~1003.

演化算法时间复杂性的趋势条件

何 军, 姚 新, 康立山

(伯明翰大学 计算机学院,伯明翰 B15 2TT,英国);
(武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

摘要:计算时间复杂性是演化理论中的一个重大课题.将趋势分析引入演化算法的平均时间复杂性分析,可用于很广一类演化算法及许多问题.基于趋势分析,研究了确定演化算法时间复杂性的一些有用的趋势条件.这些条件应用于完全欺骗问题以验证其有效性.

关键词:演华算法;时间复杂性;Markov 链;组合优化

中图法分类号:TP18 **文献标识码**:A