HSEA-hw3

201300086史浩男

1 Problem 1: 求解LeadingOnes问题(20)

请用适应层分析法来分析(1+1)-EA找到LeadingOnes问题最优解的期望运行时间上界。

定义 1 (LeadingOnes). 一个规模为n的*LeadingOnes*问题旨在找到一个n位的*01*串，以最大化

$$f(s)=\sum_{i=1}^{n}\prod_{j=1}^{i}s_j,$$

这里s_j指$s\in\{0,1\}^n$的第j位。

Algorithm 1 (1+1)-EA

Input: 伪布尔函数$f:\{0,1\}^n\to\mathbb{R}$
Output: $\{0,1\}^n$中的一个解

```
1: 随机均匀地从{0,1}^n中选择一个解s作为初始解
2: while 算法终止条件不满足 do
3:     s' ←将s的每一位独立地以1/n的概率翻转
4:     if f(s') ≥ f(s) then
5:         s ← s'
6:     end if
7: end while
8: return s
```

第一步

适应层分析法，先划分解空间：

根据从左开始连续1的个数，划分为n+1个子空间，其中$S_i=\{s\in\{0,1\}^n|f(s)=i\}$

第二步

计算从较低层S_j `jump` 到较高层S_i的概率：

保持从左开始连续的1不变，翻转遇到的第一个0

$$P(\xi_{t+1}\in\cup_{j=i+1}^{m}S_j\mid\xi_t\in S_i)\geq\frac{1}{n}(1-\frac{1}{n})^i \tag{1}$$

第三步

运行时间公式：

$$\begin{aligned}&\sum_{i=0}^{n-1}\pi_0\left(S_i\right)\sum_{j=i}^{n-1}\frac{1}{v_j}\leq\sum_{j=0}^{n-1}\frac{1}{v_j}\\=&\sum_{j=0}^{n-1}n\frac{1}{\left(1-\frac{1}{n}\right)^i}\leq\sum_{j=0}^{n-1}n\frac{1}{\left(1-\frac{1}{n}\right)^{n-1}}\leq en^2\in O\left(n^2\right)\end{aligned} \tag{2}$$

2 Problem 2: 求解OneMax问题(40)

(1) 请用乘性漂移分析法求解(1+1)-EA算法找到OneMax问题最优解的期望运行时间上界(20)。

(2) 请用加性漂移分析法求解(1+1)-EA算法找到OneMax问题最优解的期望运行时间上界(20)。

定义 2 (OneMax)**.** 一个规模为n的*OneMax*问题旨在找到一个n位的*01*串，以最大化

$$f(\boldsymbol{s})=\sum_{i=1}^{n} s_i,$$

这里s_i指$\boldsymbol{s}\in\{0,1\}^n$的第$i$位。

(1)乘性漂移分析

第一步：设计距离函数

$V(x)=n-f(x)$，其中$f(x)$表示x中总共有多少位是1

第二步：计算期望单位漂移距离的下界

在现有$n-i$个0中，翻转一个，其他位不变

$$E\left[V\left(\xi_t\right)-V\left(\xi_{t+1}\right) \mid \xi_t=x\right] \geq (n-i)\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}=\delta V(x) \tag{3}$$

第三步：计算期望运行时间上界

$$\sum_{x\in\chi}\pi_0(x)\frac{1+\ln\left(V(x)/V_{min}\right)}{\delta}=\sum_{x\in\chi}\pi_0(x)\frac{1+\ln(n-f(x))}{\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}}=\frac{1+\ln n}{\frac{e}{n}}\in O(n\log n) \tag{4}$$

(2)加性漂移分析

第一步：设计距离函数

$V(x)=n-f(x)$，其中$f(x)$表示x中总共有多少位是1

第二步：计算期望单位漂移距离的下界

在现有$n-i$个0中，翻转一个，其他位不变

$$E\left[V\left(\xi_t\right)-V\left(\xi_{t+1}\right) \mid \xi_t=x\right] \geq (n-i)\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}=c_l \tag{5}$$

第三步：计算期望运行时间上界

$$\sum_{x\in\chi}\pi_0(x)\frac{V(x)}{c_l}\leq\frac{n}{c_l}=en^2\in O(n^2) \tag{6}$$

3 Problem 3: 求解COCZ问题(40)

试求解GSEMO算法找到COCZ问题的帕累托前沿的期望运行时间上界。

定义 3 (COCZ). 一个规模为n的$COCZ$: $\{0,1\}^n \to \mathbb{N}^2$问题旨在找到一个$n$位的$01$串，以最大化

$$COCZ(s) = \left(\sum_{i=1}^{n} s_i, \sum_{i=1}^{n/2} s_i + \sum_{i=n/2+1}^{n} (1 - s_i) \right) \tag{3}$$

这里n为偶数，且s_i指$\boldsymbol{s} \in \{0,1\}^n$的第$i$位。

对于二目标优化问题COCZ，可通过占优规则来比较两个解的优劣，具体定义如下：

定义 4 (占优规则). 对于具有两目标(f_1, f_2)的解$\boldsymbol{s}$和$\boldsymbol{s}'$来说，

1. 若 $\forall i: f_i(\boldsymbol{s}) \geq f_i(\boldsymbol{s}')$，则$\boldsymbol{s}$ 弱占优 $\boldsymbol{s}'$，即$\boldsymbol{s}$好于$\boldsymbol{s}'$，表示为$\boldsymbol{s} \succeq \boldsymbol{s}'$;

2. 若 $\boldsymbol{s} \succeq \boldsymbol{s}' \wedge \exists i: f_i(\boldsymbol{s}) > f_i(\boldsymbol{s}')$，则$\boldsymbol{s}$ 占优 $\boldsymbol{s}'$，即$\boldsymbol{s}$严格好于$\boldsymbol{s}'$，表示为$\boldsymbol{s} \succ \boldsymbol{s}'$;

3. 若既不满足$\boldsymbol{s} \succeq \boldsymbol{s}'$又不满足$\boldsymbol{s}' \succeq \boldsymbol{s}$，则$\boldsymbol{s}$ 和 $\boldsymbol{s}'$ 二者不可比.

帕累托前沿的定义如下：

定义 5 (帕累托前沿). 令$\mathcal{X}$代表问题的解空间。若解空间中不存在解能优于$\boldsymbol{s}$，则称$\boldsymbol{s}$为帕累托最优解。所有帕累托最优解的目标向量集合称为帕累托前沿。

Algorithm 2 GSEMO

```
随机均匀地从{0,1}^n中选择一个解s作为初始解
将初始解放入种群P ← {s}
while 算法终止条件不满足 do
    随机均匀地从种群P中挑选出解s
    s' ←将s的每一位独立地以1/n的概率翻转
    if ∄z ∈ P 使得z ≻ s' then
        P = (P − {z ∈ P | s' ⪰ z}) ∪ {s'}
    end if
end while
```

1、问题分析：

分析COCZ问题的解，发现所有帕累托最优解恰好是**所有前$n/2$位**都是1的解

观察GSEMO算法，发现只有可比的解之间才会发生替换，也就是说随着算法的运行，解空间中解的个数**严格递增**；且已达到帕累托最优解的个体不会被替换掉，所以解空间中帕累托最优解的个数也是**严格递增**、

2、解决思路：

因此可以把解决问题拆成两部分：

1. 算法找到互相之间不可比的n/2个解组成的解空间，即种群P中解的个数达到最大值。

   这一部分，可以看成同时处理n/2个类似`OneMax`的问题，其中每个类似`OneMax`的问题都在优化n位的01串的**后n/2位**，优化目标为后n/2位中共有i个1。

2. 种群P中解的个数不变，只对每个解进行可能的优化，直到所有解都成为帕累托最优解。

   这一部分，可以看成同时处理n/2个`OneMax`问题，其中每个`OneMax`问题都在优化n位的01串的**前n/2位**，优化目标为前n/2位全是1

3、引理--OneMax的推广：

定义一类与和`OneMax`相似的问题，我称之为`类OneMax`问题。`类OneMax`与`OneMax`的唯一区别在于，`OneMax`的优化目标是`111...111`，而`类OneMax`的优化目标是同样长的任意01串

下面证明，`类OneMax`的解决的时间上界都和`OneMax`相同，是$O(n\log n)$

第一步，划分解空间：

根据从左开始连续满足优化目标的位数个数，划分为n+1个子空间，其中$S_i=\{s\in\{0,1\}^n|s\text{中的前}i\text{位满足优化目标}\}$

第二步：

计算从较低层S_j `jump`到较高层S_i的概率：

保持其他位数不变，翻转从左开始遇到的第一个不满足优化目标的位

$$P(\xi_{t+1}\in\cup_{j=i+1}^{m}S_j\mid\xi_t\in S_i)\geq\frac{n-i}{n}(1-\frac{1}{n})^{n-1} \tag{7}$$

第三步

运行时间公式：

$$\begin{aligned}&\sum_{i=0}^{n-1}\pi_0\left(S_i\right)\sum_{j=i}^{n-1}\frac{1}{v_j}\leq\sum_{j=0}^{n-1}\frac{1}{v_j}\\ \leq&\sum_{j=0}^{n-1}\frac{n}{n-j}\frac{1}{\left(1-\frac{1}{n}\right)^{n-1}}\leq en\sum_{j=1}^{n}\frac{1}{j}\in O(n\log n)\end{aligned} \tag{8}$$

4、第一部分解决方法：

把“找到互相之间不可比的n/2个解”分解成“找到特定的n/2个解”：

就像`OneMax`问题的目标是找到`111...111`一样，我把问题分解成找到最后n/2位分别为`000..000`,`1000...000`,`1100...000`,`1110...000`,......,`111...100`,`111...110`,`111...111`的这n/2个`类OneMax`子问题，这些子问题代表了所有互相之间不可比的解。由于解空间中解的个数**严格递增**，因此解出的每个子问题都不会被算法后续步骤破坏掉。解出这些子问题，也就找到了互相之间不可比的n/2个解。

根据引理，可以在$n*O(n\log n)=O(n^2\log n)$时间内解决这n个子问题，也就解决了第一部分，得到了包含互相之间不可比的n/2个解组成的解空间

5、第二部分解决方法：

此时已得到互相之间不可比的n/2个解，只需对每个解进行优化，直到所有解都成为帕累托最优解。

其中每个解的优化都是优化目标为前n/2全为1的`类OneMax`问题，一共有n/2个`类OneMax`问题，一共需要$n * O(n \log n) = O(n^2 \log n)$的时间上界

6、结论

因此，总时间上界为$O(n^2 \log n) + O(n^2 \log n) = O(n^2 \log n)$