动态设置自适应参数下界

张柯远，11812714

摘要——自适应遗传算法 EP 中η的下界和优化的结果有着密切的联系。然而η的下界和优化的问题有着高度的相关性[1]，因此自适应地设置η的下界可以使 EP 算法具有更强的适应性。本文讨论了[2]中提出的固定间隔更新η下界的算法 DLB2 可能会出现η无法收敛的问题，并根据该算法设计了一个能够动态调节更新间隔的算法DLB2'。论文中还将展示FEP[1]和 DLB2'算法在多个目标函数上的比较，并且具体分析 DLB2'在单峰函数，上位性函数，多峰函数的不同表现，以及η的变化情况。实验结果表明 DLB2'在单峰函数上能够有效地设置η的下界并且取得比 FLB[2]更好的结果。但是在上位性函数，多峰函数中仍存在一些问题。

关键词—*遗传算法，自适应，动态更改更新间隔*

I. 引言

传统的自适应遗传算法 evolutionary programming (EP)利用参数η来控制种群中每个个体每个分量的突变步长[1]。但是随着迭代次数的增加，η突变成一个非常小的数的概率也会增加，这被称为步长控制丢失[2]。一些研究证明设置一个合适的η下界可以有效地改善 EP 的性能[3]。因此，本文基于一种固定迭代间隔更新η下界的算法 IFEP_DLB2[2]，提出能够动态地改变更新下界的时刻的算法 IFEP_DLB3。

论文之后的内容如下：第二部分会讨论固定迭代间隔可能会带来的问题。第三部分会提出修改算法的思路以及算法的伪代码。第四部分会将第三部分提出的算法和固定η下界的 FEP[1]算法比较并展示实验数据。基于第四部分的实验，第五部分会分析 IFEP_DLB3 对于η下界在不同目标函数上的影响。最后第六部分会总结上文中的重点以及可改进的方向。

II. 固定间隔的更新下界

在论文[2]中，η的第κ下界$\eta_^{\kappa+1}$是每隔5代更新成当前被接受的子代的η每个分量的中位数，原论文中的公式如下：

$$\bar{\delta}(j) = \frac{1}{m}\sum_{1}^{m}\delta_v(j),\ \ j = 1,\dots,n$$

$$\eta\_^{\kappa+1} = median\{\bar{\delta}(j), \quad j = 1,2,\dots,n\} \qquad (1)$$

其中j代表第j个分量，m代表被接受的子代的个数，n表示的是每个个体的维度。显然，如果更新下界的间隔设置的过大，那么会出现 loss of step size control 的情况。不过，当间隔设置的太小，则η有可能向另一个极端发展，即变得过大。可以证明，如果不考虑选择（selection）对η的影响，随着迭代次数的增加，η中出现过大的值的概率也会增大。证明的思路和[2]中有关步长控制丢失概率的证明非常相似，只需将小于号换成大于号，便可对称地得到：

$$\forall A, A > 0, \frac{\partial}{\partial\kappa}P(\eta^{\kappa}(j) > A) > 0, 当\ A > \eta^{0}(j) \qquad (2)$$

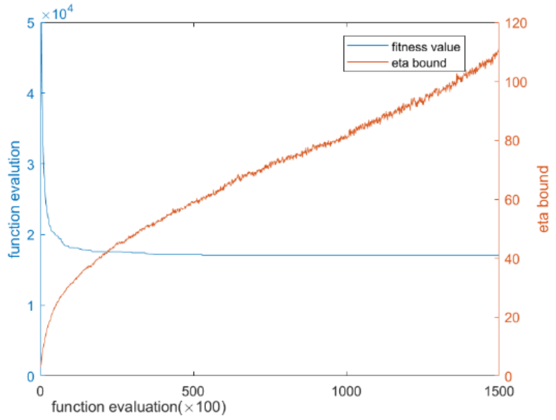

图 1 间隔为5时每代最优目标函数值以及η下界的变化。结果为30次的平均值。

其中j代表第j个分量，η^0(j)是η第j个分量的初始值。从这个结论可以看出，随着迭代次数的增加，会出现一些非常大的η值。实际上，根据η突变的公式[3]：

$$\eta^{\kappa+1}(j) = \eta^{\kappa}(j)\exp(\tau N(0,1))$$

η的取值符合对数正太分布（logarithmic normal distribution）。对数正态分布的期望等于$e^{\mu+\sigma^2/2}$，当迭代次数κ增加时，参数σ也会增加，整体的期望会上升。因此，如果过于频繁地将下界设置为η各部分分量均值的中位数，会导致整体η的值不断上升。而η的值过大会导致算法找到更优解的概率降低[1]，从而导致算法无法继续。

我在30维的球函数$f(x) = \sum_{1}^{39} x^2$用(10+10)EP 算法进行了实验。最大评估次数为十五万次。初始的η都设为3。图一展示了当固定更新间隔为 5 时η下界随迭代次数的变化情况。图中的结果是 30 次运行的平均值。可以看到η的下界随着迭代次数的增加不断地上升，同时最优目标函数值也很快不再变化。

III. 动态更新间隔

根据部分 II 中，过大的η会使得找到更优解的概率降低，因此只要足够长的迭代次数，survival selection 便能够将带有过大η的个体剔除。因此，整个动态更新间隔的算法 DLB2' 只需根据η的变化，调整更新下界的间隔，从而控制η的下界。

下界的数值参考了论文[2]中 DLB2 的公式，并在其基础上做了滑动平均的处理，使得下界的变化更加平滑，公式如下，其中α的值为0.875：

$$\eta\_^{\kappa+1} = \alpha * \eta\_^{\kappa} + (1-\alpha) * median\{\bar{\delta}(j), \quad j = 1,2,\dots,n\}(3)$$

Algorithm 1 IEFP_Mutation

Input: parents
Output: offspring

$\tau = 1/\sqrt{2\sqrt{\mu}}$
$\tau' = 1/\sqrt{2\mu}$
for all individual in parents **do**
  for j = 1 to n **do**
    $\eta(i,j) = \eta(i,j) * exp(\tau * rand + \tau' * rand_j)$
    $child_1 = parents(i,j) * cauchy(0,1)$
    $child_2 = parents(i,j) * gauss(0,1)$
    if eval($child_1$) $\le$ eval($child_2$) **then**
      offspring(i, j) = $child_1$
    else
      offspring(i, j) = $child_2$
    end if
  end for
end for

Algorithm 2 IEFP_Selection

Input: parents, offspring
Output: acc_offspring

tournament_size = 10
total = [parents, offspring]
tournament_score = []
for i in 2*μ **do**
  opponent = select tournament_size from parents and offspringtournament[i] = # of opponent has smaller fitness value than total(i)
end for
index = argsort(tournament_score)(1:μ)
acc_offspring = total[index]

Algorithm 3 IEFP_DLB2'

Input: func
Output: bestx, besty

Initialization
count = 1
while stop criteria is not true **do**
  offspring = mutate(μ)
  μ=tournament_selection(μ, offspring)
  if count is equal to α **then**
    δ = median of mean of μ's component
    if $\delta \le \mu b$ **then**
      $\alpha = \alpha + 10$
    else
      for all μ_i in μ **do**
        $\mu_i = \max\{\mu b, \mu_i\}$
        $\alpha = \lfloor \alpha/2 \rfloor$
      end for
    end if
    $\mu b = 0.875 * \mu b + 0.125 * \delta$
    count = 1
  end if
  count += 1
end while

当到达下一次更新下界时，首先判断当前$median\{\delta(j)\}$是否小于上一次的下界。如果小于，则将更新的间隔设置成原来的一半并且把所有小于下界η_的η改为η_。否则，将更新的间隔增大十代，不更新η。每次到达更新间隔时都会根据(3) 更新η_的值。

算法的其他部分同$(\mu + \lambda)$IFEP[1]。每一次迭代，每一个个体都会用柯西随机变量和高斯随机变量突变生成两个子代，然后选择适应度函数更高的子代作为其真正的子代。在选择部分，采用锦标赛选择的算子，将 tournament size 设为 10，从父代和其真正的子代中选出下一次迭代的子代。

Algorithm 1 和 2 展示了突变和选择部分的伪代码。Algorithm 3 主要展示了如何更新η和间隔的部分，其余部分同 IFEP[1]。

IV. 实验结果

A. 实验参数设置

为了验证DLB2’的性能，本次实验还实现了FEP算法[1]与其进行对比。实验测试的目标函数为$f1-f12, f13$[1]。对于 FEP，种群的数量设为 100，η的固定下界是0.0001，初始值为3。DLB2’的种群数量为3，η的初始值也为3。种群的初始化是在定义域范围内的随机分布。

B. 实验结果

表一展示了实验的结果。其中Mean Best表示函数最后一次迭代产生的最优解的目标函数值的平均值，Std Dev 表示从表中可以看出。DLB2’算法在$f1-f4, f10, f11$中都取得了更好的结果。但是在$f5, f7, f8, f9, f13$中的效果都要差于固定间隔的FEP算法。再下一部分会针对单峰函数$f1$，$f5$（$rosenbrock$函数）以及多峰函数$f8, f9, f13$具体分析DLB2’的运行情况。

V. 实验结果分析

A. 单峰函数

在很多单峰函数上，DLB2’都比固定间隔的IFEP_FLB取得了更好的结果。如图2中所示，DLB2’目标函数的收敛速度明显快于FLB。

为了验证在$f1$上DLB2’真的能够合理地设置η，我又在$f1$进行了一次相同的实验。这次实验统计了每一次迭代 selection 后的子代中最小的η值以及η的下界变化。从图 3 可以看出，η的最小值随着η下界不断地下降，最后会到$1e^{-8}$ 。这是明显低于固定下界的0.0001的。第二点，η的最小值会不停地抖动，这也验证了随着迭代次数的增加, 的确会突变出现过小的η值。这说明 DLB2’能够根据η的变化，改变η的下界，使得寻找最优解的速度得到了一定的加快。

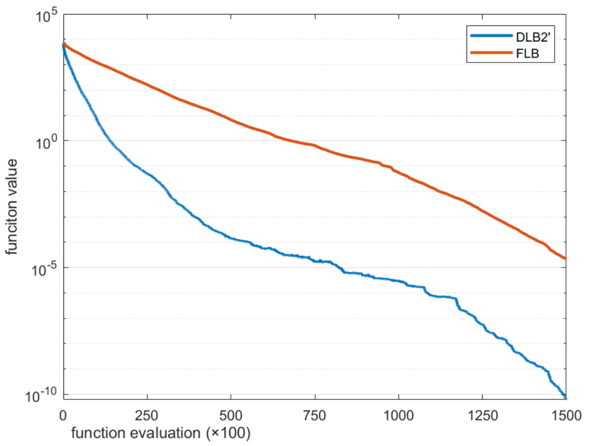

图 2 对比 IFEP_DLB2’ 与 FEP_FLB 在$f1$上目标函数收敛的情况。横坐标为函数评估次数，纵坐标为目标函数值。结果是 30 次的平均值。

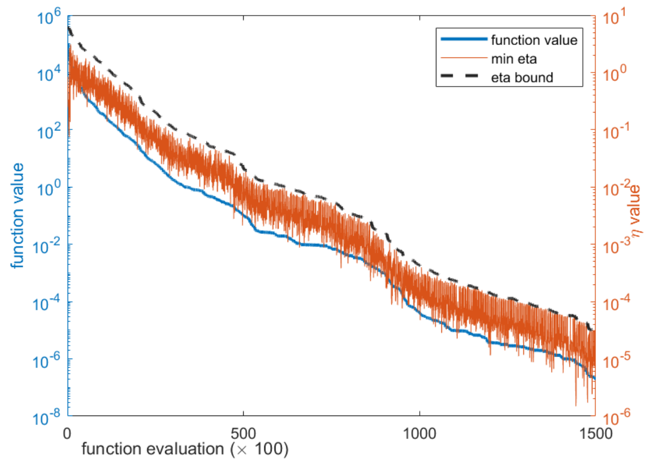

图 3 η下界和目标函数值得的变化曲线。蓝色是目标函数的收敛曲线，橙色是每代η值的最小值，黑色虚线是算法中设置的η下界。实验结果是 30 次的平均值。

表格 1 在$f1-f11,f13$上比较 IFEP_DLBB2’ 和 FEP_FLB。其中 Mean Best 表示

Function	# of Evaluation (× 100)	IFEP_DLB2’		FEP_FLB		DLB2’- FLB
		Mean Best	Std Dev	Mean Best	Std Dev	Signrank
$f1$	1500	2.89E-10	1.51E-9	1.16E-4	3.29E-4	-4.78
$f2$	2000	1.75E-4	4.16E-4	1.81E-2	6.77E-4	-4.78
$f3$	5000	1.28E-16	2.69E-16	0.88	1	-4.78
$f4$	5000	1.48E-15	3.73E-15	0.02	0.04	-4.78
$f5$	20000	12.69	7.23	5.31	4.43	3.34
$f6$	1500	0	0	0	0	0
$f7$	3000	3.54E-2	1.37E-2	1.43E-2	7.4E-3	4.78
$f8$	9000	-10899	358.25	-11409	365.96	3.69
$f9$	5000	31.24	9.23	0.69	0.90	4.78
$f10$	1500	2.40E-5	5.86E-5	5.60E-3	6.1E-3	-4.78
$f11$	2000	1.86E-2	1.56E-2	9.75E-2	0.21	-2.25
$f13$	100	-5.17	3.92	-9.00	3.83	4.29

B. f5 函数

$f5$在某些方向上的下降速度非常缓慢，因此将整体η所有分量的中位数作为每个分量的下界有可能会导致一些分量上的η太早地收敛，从而导致优化的速度变慢。从图 4 中可以看出，DLB2’目标函数值很快就停止了下降，而 FLB 却可以持续缓慢地下降。

为了验证 DLB2’的η在某些分量上被其他分量影响，我选择两者第 30 个维度上的η值进行比较，因为第 30 维向函数全局最优解下降的梯度最小。从图 5 中的对比可以看出 DLB2’的η要明显小于 FLB 的η值，最开始下降的速度也是 DLB2’的更快。而 DLB2’ η值尽管会因为不断设置下界而上下波动，但是因为下界明显低于 FLB 固定的下界，因此在第 30 维方向上突变的步长也会小于 FLB。

C. 多峰函数

DLB2’在多峰函数上的效果都不如 FLB。在分析了多个多峰函数上 DLB2’的η的均值变化的情况后，我总结出了可能的原因：尽管动态的调整更新下界的迭代间隔可以减少η过小或者过大带来的影响，但是并不能调整至当前状态合适的步长。换言之，有时候η收

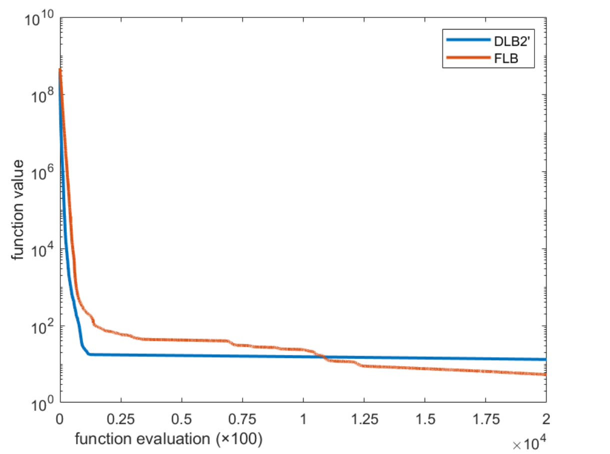

图 4 在$f5$上比较 FLB 和 DLB2’的目标函数收敛曲线。蓝色表示 DLB2’，橙色表示 FLB。实验的结果是 30 次的平均值。

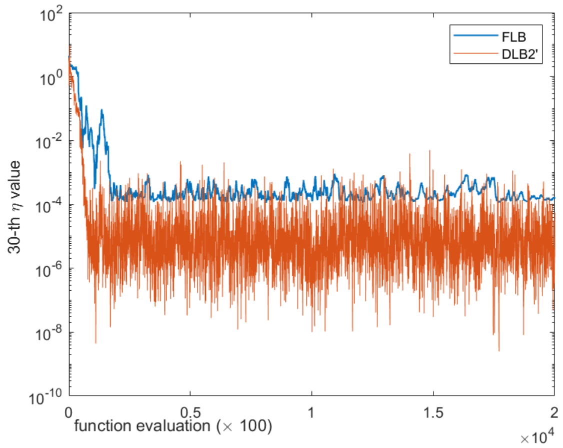

图 5 f5上 FLB 和 DLB2’第 30 维η的均值的变化情况。蓝色代表 FLB，橙色代表 DLB2’

敛的速度很快，因为 DLB2’贪心地选取子代的η的均值作为下界，所以下界收敛的速度也会很快，最终导致突变步长过短无法继续优化；也有可能当η过大，收敛的速度较慢时，η的均值下降缓慢。但是 DLB2’持续地设置下界会渐慢η下降的速度，使得η不能或需要很长时间下降到合适的大小，因此优化也很难进行下去。

图 6 展示了收敛速度过快的情况。图中可以看到当 DLB2’很快地收敛到一定程度时便停在了局部最优解无法继续优化。图 8 展示了 DLB2’ η下界和 FLB η均值的对比。从图 8 可以看出，DLB2’的η在迅速下降，并一直低于 FLB 的η值。这和图 6 的目标函数收敛曲线是一致的。

图 7 和图 9 展示了f9上 DLB2’和 FLB 的情况。除了f8中出现的η收敛速度过快导致突变步长过小的情况，还在迭代的初期出现η收敛过慢，优化暂时停止的情况。从图 7 可以看出，在 100000 到 200000 次评估之间，当前最优的目标函数值都没有下降，明显小于慢于 FLB。而从图 9 中可以看到，在这个区间里η的均值也没有下降，而 FLB 有明显的下降。这和图 7 中的现象是符合的。而从 250000 次评估之后，DLB2’的η收敛速度过快，导致步长提前过小。

总之，在多峰函数上，DLB2’表现出η下界无法合理地控制步长。在η下降速度过快或者过慢导致无法继续优化时，都不能有效地减缓或者加速η收敛速度。

VI. 总结

论文中讨论分析了在 EP 算法中固定间隔更新η下界可能会导致的问题，并由此问题出发提出依据η均值变化趋势动态更改时间间隔的算法 DLB2’。实验的结果表明，DLB2’可以有效地避免丢失步长控制以及步长不断增大的问题，并且在一些单峰函数上取得了比固定η下界更可观的结果。但是 DLB2’依然存在很多问题。在f5函数上表现出某些η的某些分量会受到其他分量的影响导致过小。另外，动态更新间隔不能反映出合适的突变步长。因此在多峰函数中，η有时过早地收敛至很小的数，有时又在较大时卡住。这些都会导致找不到最优解或者目标函数收敛缓慢。

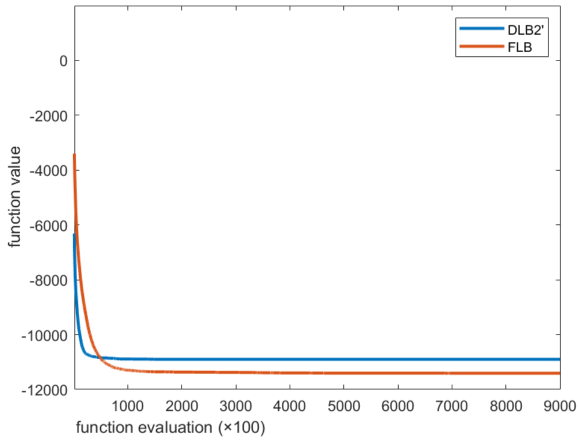

图 6 在f8上比较 FLB 和 DLB2’的目标函数收敛曲线。蓝色表示 DLB2’，橙色表示 FLB。实验的结果是 30 次的平均值。

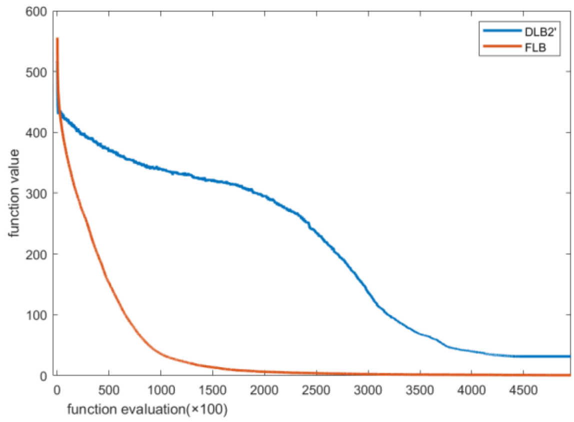

图 7 在f9上比较 FLB 和 DLB2’的目标函数收敛曲线。蓝色表示 DLB2’，橙色表示 FLB。实验的结果是 30 次的平均值。

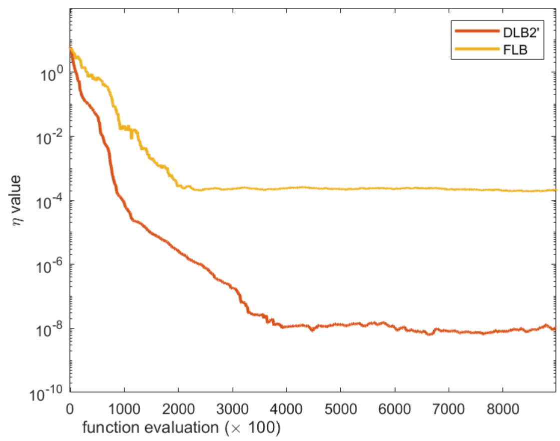

图 8 f8上比较 FLB 和 DLB2’的η变化。黄色是 FLB，橙色是 DLB。

基于上述的种种问题，该实验未来可以从以下两个角度研究：一是如何定义η的下界。可以考虑将η整体的均值和η各个分量结合在一起，使得分量之间不会因为互相影响而导致在某一方向上过大或过下。二是除了η均值的变化外，寻找其他的先验信息来决定更新

下界的时间间隔。使得下界的更新能反映出当前种群合适的突变步长。

REFERENCES

[1] X. Yao, Y. Liu and G.M. Lin, "Evolutionary programming made faster," in *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, July 1999.

[2] Liang, Ko Hsin, X. Yao and C. S. Newton, "Adapting Self-Adaptive Parameters in Evolutionary Algorithms." in *Applied Intelligence*, vol. 15 no. 3 pp.171-180, 2001.

[3] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," Evol. Comput., vol. 1, no. 1, pp. 1–23, 1993.

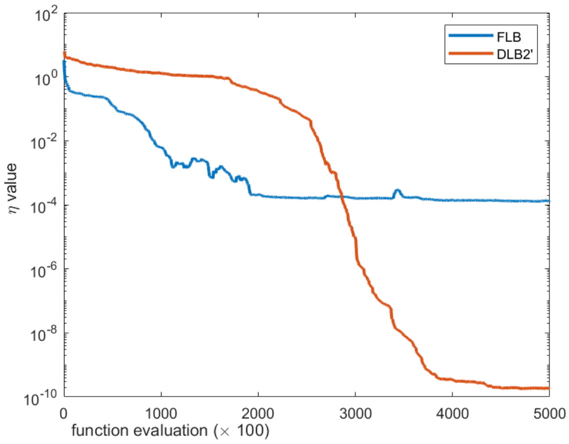

图 9 $f9$上比较 FLB 和 DLB2' η的均值变化。黄色是 FLB，橙色是 DLB。