2003 年 3 月
第 5 卷第 3 期

中国工程科学
Engineering Science

Mar. 2003
Vol.5 No.3

学术论文

基于空间收缩的并行演化算法

王　涛，李歧强
（山东大学控制科学与工程学院，济南　250061）

[摘要]　提出了一种基于空间收缩的求解 MINLP 问题的新算法。算法应用了快速有效的不完全演化搜索较优解的分布信息，通过分布信息定位最优解的可能分布，再由精英个体信息决定下次搜索空间。仿真结果表明该算法在搜索效率、应用范围、解的精确性和鲁棒性上都优于其他现存演化算法。

[关键词]　空间收缩；演化算法；MINLP
[中图分类号]　TP301　　[文献标识码]　A　　[文章编号]　1009－1742（2003）03－0057－05

1　引言

演化算法(EA，evolutionary algorithms)是近年来兴起的一类基于生物界的自然选择和自然遗传机制的计算方法[1~3]，如遗传算法（GA，genetic algorithms)、演化策略(ES，evolution strategy)[4，5]等方法，它是一类内在并行(inherent parallelism)的优化方法。这类方法的主要优点在于其本质上的并行性、广泛的可应用性和算法的高度鲁棒性、简明性与全局优化性，其特性决定了它特别适合于解大规模并行模型。比较优秀的演化计算方法有 DE 算法(differential evolution)[6]、基于 DE 和 HDE 的 MIHDE 算法（mixed integer hybrid differential evolution)[7]、Guo Tao 的 GT 算法[8]以及 ACSSOS 算法[9]等。

大多数的演化算法在求解 MINLP 问题时，由于问题的规模过大和算法本身的缺陷，导致算法收敛速度过慢，或者得到的解不准确，甚至是不可行解。在算法中引入空间划分或空间收缩的思想，可以快速的缩小寻优空间，极大的提高收敛速度。现在大多数的空间搜索算法，都是通过空间的划分和淘汰实现搜索，本质上也是一类分支定界算法。它首先将解空间划分为若干子空间，然后在每个子空间内分别选取若干个测试点，通过测试点的好坏来确定保留与淘汰的空间。在局域性不好的搜索空间中，如果这种算法保留的空间和测试点数不足够大，很难找到全局最优点。MIHDE 算法已经成功解决了很多 MINLP 问题，但对有些问题得到的解的质量不高。GT 算法是一种快速有效的空间搜索算法，但它没有考虑求解非线性混合整数规划问题。另外，ACSSOS 也是一种快速的空间收缩的算法，但它要求适应值函数的搜索空间具有连续性和凸性，对于不满足这些条件的问题便无能为力。

笔者研究的一种空间收缩算法应用了快速有效的不完全演化搜索较优解的分布信息[9]，通过分布信息定位最优解的可能分布，逐步缩小求解空间。收缩空间时由于采用了由精英个体决定下次搜索空间的方法，因此使算法比其他同类空间收缩算法（如 ACSSOS）具有更高的搜索效率，应用范围也更广，可以求解搜索空间为连续或离散的

[收稿日期]　2002－09－05；修回日期　2002－11－20
[基金项目]　“八六三”高技术资助项目（2001AA413420），山东省优秀中青年科学家奖励基金资助项目（9901）
[作者简介]　王　涛（1978－），男，山东济南市人，山东大学硕士研究生

MINLP 问题。演化策略采用了 $\mu+1$ 选择策略。

2 混合整数优化问题的表示

为不失一般性，MINLP 问题可描述为：

$$\min_{x,y} f(x,y),$$

$$\text{s.t. } h_i(x,y)=0,\ i=1,\cdots,m_i,$$

$$g_j(x,y)\leqslant 0,\ j=1,\cdots,m_j,$$

$$x^l\leqslant x\leqslant x^u,$$

$$y^l\leqslant y\leqslant y^u。$$

其中：$x=(x_1,x_2,\cdots,x_{n_r})$ 表示 n_r 维实型变量；

$x^l=(x_1^l,x_2^l,\cdots,x_{n_r}^l)$，$x^u=(x_1^u,x_2^u,\cdots,x_{n_r}^u)$ 分别表示决策变量 x 的下、上界；

$y=(y_1,y_2,\cdots,y_{n_i})$ 表示 n_i 维整型变量；

$y^l=(y_1^l,y_2^l,\cdots,y_{n_i}^l)$，$y^u=(y_1^u,y_2^u,\cdots,y_{n_i}^u)$ 分别表示决策变量 y 的下、上界。

引入下列记号：

解空间 $S=\{(x,y)\mid x^l\leqslant x\leqslant x^u,\ y^l\leqslant y\leqslant y^u\}$；

空间的边长 $(\omega_x,\omega_y)=(x^u-x^l,y^u-y^l)$；

空间的维数 $n=n_r+n_i$；

空间的中心 $C_x=(C_{x_1},C_{x_2},\cdots,C_{x_{n_r}})=\left(\frac{x_1^l+x_1^u}{2},\frac{x_2^l+x_2^u}{2},\cdots,\frac{x_{n_r}^l+x_{n_r}^u}{2}\right)$；

$c_y=(C_{y_1},C_{y_2},\cdots,C_{y_{n_r}})=\left(\frac{y_1^l+y_1^u}{2},\frac{y_2^l+y_2^u}{2},\cdots\frac{y_{n_r}^l+y_{n_r}^u}{2}。\right)$

点到中心距离 $R=\sqrt{R_x^2+R_y^2}$，

$$R_x^2=(x_1-C_{x_1})^2+\cdots+(x_{n_r}-C_{x_{n_r}})^2,$$

$$R_y^2=(y_1-C_{y_1})^2+\cdots+(y_{n_i}-C_{y_{n_i}})^2;$$

空间的半径 $R=\max(R_l,R_u)$，

$$R_l=\sqrt{(R_x^l)^2+(R_y^l)^2},$$

$$R_u=\sqrt{(R_x^u)^2+(R_y^u)^2}。$$

这里 x 表示 n_r 维实型变量（$x\in R^{n_r}$），y 表示 n_i 维整型变量（$y\in I^{n_i}$），上述问题的目标函数、等式约束、不等式约束都是非线性的，称为带约束的非线性混合整数规划。求解这类问题的一般方法是通过惩罚函数将其变成无约束问题，惩罚适应函数为

$$\Phi(x,y)=f(x,y)+\sum_{k=1}^{m_i}\alpha_k h_k^2(x,y)+\sum_{k=1}^{m_j}\beta_k g_k^2(x,y),$$

其中 $g_k(x,y)=\max(0,g_k)$。

3 算法的主要思想

3.1 不完全演化

大多数的演化算法在运行初期，群体中个体的适应值改进较快，执行到一定代数后，群体中的个体分散在各个峰值附近，此后用遗传操作很难再改进后代适应值，从而导致收敛速度变慢。如果当个体分散在各个峰值附近时停止演化，便称之为不完全演化算法。它的优点是可以快速获取较优点信息。重复进行不完全演化，便可获得足够多的较优点信息，为定位最优解、缩小解空间提供依据。

3.2 $\mu+1$ 选择策略

$\mu+1$ 是演化策略中一种有效的选择策略，它利用群体中 μ 个个体的信息产生一个新个体，若此新个体优于群体中的最差个体，则将其取代。

3.3 空间收缩的思想

通过在解空间内进行不完全演化，快速获得足够数目的较优点信息。将传统的根据空间来决定测试点的方法，改为由测试点来决定空间，把保留空间改为保留测试点。测试点选取不完全演化后的精英解，由精英解决定的空间作为下一代的解空间。这样做可以充分利用不完全演化得到的分布信息，快速高效的对空间进行收缩。

3.4 半限定数学方法

若一个变量 x_i 的实际值是未知的，但包含该变量实际值的一个取值区间是已知的，那么这个变量就叫做半限定变量（subdefinite variables），这个区间就是半限定变量真值的一个估计。半限定变量用符号 X 表示。当这个区间收缩为一个点时，半限定变量转变为确定变量。为了衡量空间压缩的程度，定义了限定度这个概念。

设有变量 $x=(x_1,x_2,\cdots,x_n)\in R^n$，其对应的半限定变量为 $X=(X_1,X_2,\cdots,X_n)$，则各半限定变量的限定度定义为

$\gamma(x_i)=\exp(-V_i/V'_i)$，$i=1,2,\cdots,n$。

其中 V_i 表示收缩后变量 x_i 的取值范围，V'_i 表示收缩前变量 x_i 的取值范围，即

$$V_i=\begin{cases}|V_i^{\max}-V_i^{\min}|, & x_i\text{ 为连续变量},\\ N_i, & x_i\text{ 为离散变量};\end{cases}$$

$$V'_i = \begin{cases} |V'^{\max}_i - V'^{\min}_i|, & x_i \text{ 为连续变量}, \\ N'_i, & x_i \text{ 为离散变量}。 \end{cases}$$

$V_i^{\max}$和 $V_i^{\min}$ 为 X_i 的区间端点，N_i 为 X_i 的元素个数。

半限定变量的限定度越大，原组合优化问题的解空间就被压缩的越大，搜寻最优解所花的时间就会越短。

3.5　空间收缩停止条件

根据解空间的规模，变量的限定度到达适当范围时停止收缩，改为完全演化或其他快速算法。目前比较实用的算法如遗传算法、禁忌搜索、模拟退火等。

3.6　遗传操作

在不完全演化过程中，遗传操作用来产生下一代的个体，遗传操作中用到了交叉和变异算子。交叉操作要求交叉产生的子代能够比较均匀地分布在当前解空间内。群体中 m 个个体张成仿射空间 $S = \{X \mid X = \sum_{i=1}^{m} \omega_i x_i\}$，在满足 $\sum_{i=1}^{m} \omega_i = 1$ 的情况下，适当地随机选取权值，可使生成点比较均匀地分布在 m 个个体形成的最小凸集内。由 m 个父体产生一个个体 (x', y') 的交叉算子定义为：

$$x' = \sum_{i=1}^{m} \alpha_i x_i,$$

$$y' = \text{ent}\left(\sum_{i=1}^{m} \alpha_i y_i\right)。$$

随机产生 α_i 满足：

$$-0.5 \leqslant \alpha_i \leqslant 1.5, \text{ 且 } \sum_{i=1}^{m} \alpha_i = 1。$$

变异算子定义为：

$$x' = x + \tau N(0, \sigma'),$$

$$y' = y + \tau N(0, \sigma'),$$

$$\sigma' = \sigma \exp[\tau N(0, 1)]。$$

式中，$N(0, 1)$是均值为 0、方差为 1 的正态分布随机变量，τ 是算子集参数，表示变异运算时的个体步长，σ 用来调整个体进行变异操作时变异量的大小。遗传操作的具体方法是在种群中随机选取一个个体进行变异，然后随机选取两个个体进行交叉，将得到的新个体取代种群中的最差个体。

3.7　算法流程

Step 1　在解空间内随机选取种群，初始化优解池，计算解空间参数。

Step 2　解空间内进行不完全演化 r 次，更新优解池。

Step 3　选取适应值较好的个体和小概率的少量较差个体进入精英池。

Step 4　由精英池内个体确定下一代空间范围，并计算其半径。若限定度满足要求范围，转 Step 5，否则转 Step 1。

Step 5　以收缩后的空间为解空间，再应用完全演化算法求解。

4　仿真

为了检验算法的性能，对下面两个优化问题进行了求解：

例 1

$$\min_{x,y} f(x, y) = 0.6224(0.0625y_1)x_1x_2 + 1.7781(0.0625y_2)x_1^2 + 3.1661(0.0625y_1)^2x_2 + 19.84(0.0625y_1)^2x_1$$

$$\text{s.t. } g_1(x, y) = 0.0193x_1 - 0.0625y_1 \leqslant 0,$$

$$g_2(x, y) = 0.00954x_1 - 0.0625y_2 \leqslant 0,$$

$$g_3(x, y) = 750 \times 1728 - \pi x_1^2 x_2 - 4\pi x_1^3/3 \leqslant 0,$$

$$g_4(x, y) = x_2 - 240 \leqslant 0。$$

这是一个压力容器设计问题，由 Sandgren[10] 提供，设计参数是容器的各种尺寸，目标函数 $f(x, y)$ 是熔制压力容器的总成本，约束条件保证容器符合美国机械工程师学会标准，这是一个混合整数优化问题。下面是各种算法对上例的计算结果比较，仿真程序参数表如表 1 和表 2 所示。

表 1　例 1 各种算法计算结果比较

Table 1　Comparison among the results of various algorithms in example 1

项	IDCNLP	SA	MVEP	MIHDE	空间收缩
x_1	48.380 7	58.290 0	51.195 8	48.576 5	42.006 0
x_2	111.744 9	43.693 0	90.782 1	110.055 9	183.047 3
y_1	18	18	16	15	13
y_2	10	10	10	8	7
g_1	−0.191 3	−0.025 0	−0.011 9	0.000 0	−0.001 8
g_2	−0.163 4	−0.068 9	−0.136 6	−0.036 6	−0.036 8
g_3	−75.875 0	6.549 6	−13 584.563 1	0.000 0	−29 169.080 1
g_4	−128.255 1	−196.307 0	−147.217 9	−129.944 1	−56.952 7
f	8 048.619 0	7 197.7	7 108.616 0	6 370.703 5	6 193.778 8

从表 1 和表 2 可以看出，搜索空间收缩的幅度很大。限定度越大的变量，其搜索空间的收缩幅度

越大。求解例1的模型，空间收缩算法耗时20 ms，收敛速度大大快于其他几种算法。

表2 例1空间收缩算法仿真程序参数

Table 2 Parameters of simulated procedure in example 1

	压缩前	压缩后	限定度
种群规模	100	100	
空间半径	162.018 5	19.365 7	
x_1 范围	0～100	40.554 7～45.209 7	0.954 4
x_2 范围	0～200	173.487 7～211.804 1	0.825 7
y_1 范围	0～50	13～15	0.960 8
y_2 范围	0～50	7～9	0.960 8

例2

$$\min f(x) = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

这个实例来自 Schaffer[11]，已知上式的全局最优点是(0,0)，目标函数 $f(x)$ 最优值为 -0.5。函数的三维图如图1所示，在距全局最优点附近有无限多个局部最优点。由于其强烈的振荡性以及全局最优点被局部最优点包围的特性，一般优化算法很难找到它的全局最优值。

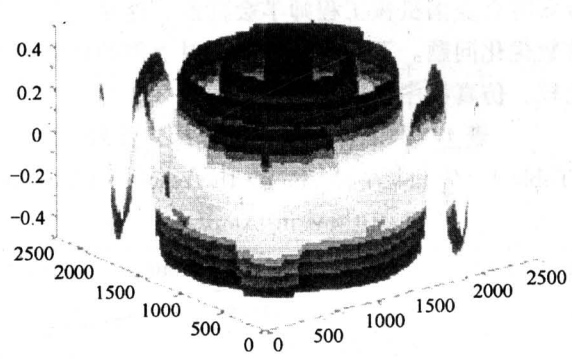

图1 例2目标函数三维图

Fig.1 Three dimensional graph of the objective function in example 2

算法的运行结果如下：最佳个体为 $x_1 = 0.001\ 377\ 55$，$x_2 = -0.000\ 618\ 82$；适应值为 $-0.499\ 997\ 71$。仿真中各个参数如表3所示。

用空间收缩算法求解实例2的目标函数，平均5次就有1次能找到全局最优解，所用时间为10 ms，即使在找不到最优解的情况里，收缩后的空间也全都包含最优解。这说明空间收缩在定位最优解范围上还是非常快速和准确的，之所以某些时候找不到全局最优解，是因为空间收缩后的完全演化过程导致算法收敛到了局部最优点，这时可以考虑增大空间收缩的限定度，也就是使空间的收缩幅度更大。

表3 例2仿真程序参数

Table 3 Parameters of simulated procedure in example 2

	压缩前	压缩后	限定度
种群规模	100	100	
空间半径	14.142 1	3.620 3	
x_1 范围	−10～10	−2.889 6～3.166 7	0.738 7
x_2 范围	−10～10	−1.367 3～2.60 2	0.820 0

另外，笔者还用该算法求解了一个调度问题模型，这是一个对多目标、多约束的 MINLP 模型进行求解的组合优化问题。该模型有100个决策变量，120个约束方程，算法完成空间收缩约用300～800 ms，完成整个寻优过程约用600～1 100 ms，而用普通遗传算法求解同样的模型并达到类似的结果大约用时15～25 s。

5 结论

1）根据上述仿真结果，空间收缩能比较准确迅速地缩小解空间，即使初始解空间很大，算法也能在很短时间内完成。计算结果表明，该算法在解的质量及收敛速度上优于一般的演化算法。

2）在不完全演化中，搜索空间越大，函数的峰越多，所需要的较优点信息也就越多，需要进行不完全演化的次数也越多。所以在算法中可以考虑随着空间的收缩，逐步减小不完全演化的次数，从而进一步提高收缩速度。

3）该算法采用了由精英个体决定下一代空间的方法，在用精英解构造新空间后，得到的是一个连续的凸空间，与初始的解空间是否连续或是否凸无关。如果初始解空间是不连续或非凸，新的解空间里可能包含不可行解，由于采用了对不可行解加惩罚的方法，选出的下一代精英解中将都是可行解，经过几次收缩，最终算法将空间收缩到一个连续的凸空间内。所以在解决搜索空间不连续的问题或空间非凸问题时，可以把空间演化的过程看作为空间取舍和空间收缩同步进行的过程。

4) 在算法中，主要的过程是解空间的不断演化，而个体的选择与淘汰只是用来决定下一代的解空间，上一代的超级个体并不带入下一代，这样可以保证算法在运行之初不易陷入局部最优，避免了早熟收敛。

参考文献

[1] Fogel D B. An introduction to simulated evolutionary optimization [J]. IEEE Transactions on Neural Networks, 1994, 5 (1): 3～14

[2] 徐宗本，李　国. 解全局优化问题的仿生类算法 (I)——模拟进化计算 [J]. 运筹学杂志，1995, 14 (2): 1～13

[3] 谢金星. 进化计算简要综述 [J]. 控制与决策，1997, 12 (1): 1～7

[4] Guo Chonghui, Tang Huanwen. Global convergence properties of evolution strategies [J]. Mathematica Numerica Sinica, 2001, 23 (1): 105～110

[5] 李　宏，唐焕文，郭崇慧. 一类进化策略的收敛性分析 [J]. 运筹学学报，1999, 3 (4): 79～83

[6] Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces [J]. Global Optimization, 1997, 11: 431～469

[7] Lin Yunchien, Wang Fengshen, Hwang Kaoshing. A hybrid method of evolutionary algorithms for mixed-integer nonlinear optimization problems [A]. Proceedings of the Congress on Evolutionary Computation [C]. Washington DC USA, IEEE Service Center, 1999, 3 (2): 159～169

[8] Guo Tao, Kang Lishan, Li Yan. A new algorithm for solving function optimization problems with inequality constraints [J]. Wuhan Univ (Nat Sci Ed), 1999, 45 (5B): 771～775

[9] Zeng Sanyou, Kang Lishan, Ding Lixin. A new method of evolutionary algorithm for mixed-integer nonlinear optimization problem [J]. Wuhan Univ (Nat Sci Ed), 2000, 46 (5B): 554～558

[10] Sandgren E. Nonlinear integer and discrete programming in mechanical design [J]. ASME J Mechanical Design, 1990, 112: 223～229

[11] Schaffer J D, Carnana R A, Eshelman L J. A study of control parameters affecting online performance of genetic algorithm for function optimization [A]. Proc of the 3rd Int'l Conf on Genetic Algorithms [C]. Morgan Kaufmann, Los Altos, 1989. 51～60

A Parallel Evolutionary Algorithm Based on Space Contraction

Wang Tao, Li Qiqiang

(*School of Control Science and Engineering, Shandong University, Ji'nan* 250061, *China*)

[**Abstract**] A novel algorithm which is based on space contraction for solving MINLP problems is proposed. The algorithm applies fast and effective non-complete evolution to the search for the information of better solutions, by which locates the possible area of optimal solutions, determines next search space by the information of elite individuals. The result shows that it is better than other existing evolutionary algorithms in search efficiency, range of applications, accuracy and robustness of solutions.

[**Key words**] space contraction; evolutionary algorithms; MINLP