计算机研究与发展 ISSN 1000-1239/CN 11-1777/TP
Journal of Computer Research and Development 43(5)：841～850, 2006

一种求解约束优化问题的演化规划算法

董红斌[1,2] 黄厚宽[1] 何 军[1] 侯 薇[3]
[1](北京交通大学计算机与信息技术学院 北京 100044)
[2](哈尔滨师范大学计算机科学系 哈尔滨 150080)
[3](东北农业大学计算机科学系 哈尔滨 150030)
(donghongbin@263.net)

An Evolutionary Programming to Solve Constrained Optimization Problems

Dong Hongbin[1,2], Huang Houkuan[1], He Jun[1], and Hou Wei[3]
[1](*School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044)
[2](*Department of Computer Science, Harbin Normal University, Harbin* 150080)
[3](*Department of Computer Science, Northeast Agricultural University, Harbin* 150030)

Abstract A mixed strategies evolutionary programming to solve constrained optimization problems is presented in this paper. The approach does not require the use of a penalty function. Instead, it uses a diversity conservation mechanism based on allowing infeasible solutions to remain in the population. A mixed mutation strategy and feasibility-based comparison mechanism is used to guide the process fast toward the feasible region of the search space. This new evolutionary programming has been tested on 13 benchmark functions. The results obtained show that the new approach is a general and effective method.

Key words constrained optimization; mixed strategy; diversity conservation mechanism; evolutionary programming

摘 要 提出了一种新的求解约束优化问题的演化算法——基于混合策略求解约束优化问题的演化规划算法(CMSEP). 借鉴了Mezura-Montes的算法中直接比较的约束处理方法, 为求解位于边界附近的全局最优解采用多样性保护机制, 允许一定比例最好不可行解进入下一代种群, 混合策略变异机制用于指导算法快速搜索过程. 标准测试函数的实验结果验证了算法的通用性和有效性

关键词 约束优化; 混合策略; 多样性保护机制; 演化规划

中图法分类号 TP18

1 引 言

演化算法(EA)求解约束优化问题时, 使用最广泛的处理约束条件的方法是罚函数法, 罚函数方法存在一个缺点, 即加上罚系数和罚函数项后生成的新目标函数 $F(x)$ 的最优解依赖于罚系数的选择. 对于要解决的约束优化问题, 事先确定适当的罚系数是很困难的, 往往需要多次实验来调整罚系数.

近年来, 已提出多种求解约束优化问题的演化算法[1~7]. Michalewicz等人在文献[1]中将现有的约束优化算法分成4类: 保存可行解法、罚函数法、可行解优于不可行解方法及其他混合方法. Riche等人采用了两个不同罚系数的方法[2], 即将群体中的个体分成两组, 每组对应不同的罚系数, 然后将两组个体混合一起进行演化. 这一算法的优点是由于

收稿日期: 2005-08-06; 修回日期: 2005-11-21
基金项目: 国家自然科学基金项目(60443003); 北京交通大学科技基金项目(2003SZ003)

两组个体的罚系数不同, 演化的轨迹也不同, 而且个体之间共同进化, 使算法跳出局部最优解的机会多, 对罚系数也不太敏感; 两个罚系数一个选得较大, 另一个较小, 使算法能从可行域和不可行域两侧同时收敛, 这样两组个体可以更快地收敛到全局最优解. Powell 和 Skolnick 提出一种特殊的罚函数法, 目的是使可行解永远优于不可行解[3]. Deb 等人引入了不需要罚因子而直接比较个体优劣的方法[4]. 其个体评价函数如下:

$$fit_i(\boldsymbol{x})=\begin{cases} f_i(\boldsymbol{x}), & \text{if feasible,} \\ f_{\text{worst}}+\sum_{j=1}^{n} g_j(\boldsymbol{x}), & \text{otherwise.}\end{cases} \quad (1)$$

周育人等人提出了 Pareto 强度值演化算法求解约束优化问题的方法[5]. Sarimveis 提出了一种称做排列微分演化(LUDE)求解约束优化问题的新算法[6], 算法基于增广拉格朗日函数的罚函数方法处理约束, 罚参数和拉格朗日乘子随着算法的不断进化而被修改, 算法中解排列的位置很重要, 它决定着交叉和排列算子的选择范围, 数值实验结果证明了这种方法的有效性. Mezura-Montes 提出了一种简单多成员演化策略(SMES)求解约束优化问题的方法[7]. 这种方法不使用罚函数, 它采用具有自适应机制的多成员演化策略进行约束搜索, 为求解位于边界附近的全局最优解, 允许一定比例的最好的不可行解进入下一代种群.

借鉴 Mezura-Montes 算法中直接比较的约束处理方法, 本文提出了一种新的求解约束优化问题的演化算法——基于混合变异策略求解约束优化问题的演化规划算法(CMSEP). CMSEP 不使用罚函数而采用一种个体比较准则, 允许部分最好的不可行解进入下一代种群的方法处理约束, 混合策略的自适应变异机制用于指导搜索过程. 测试函数的数值实验结果验证了算法的通用性和有效性.

2 约束优化问题

约束优化问题可描述为如下(最小化问题)形式:

$$\min f(x) \quad (2)$$

$$\text{s.t.}\begin{cases} g_i(\boldsymbol{x})\leqslant 0, & i=1,\cdots,n, \\ h_j(\boldsymbol{x})=0, & j=1,\cdots,p,\end{cases} \quad (3)$$

这里 $\boldsymbol{x}$ 是解向量, $\boldsymbol{x}=(x_1, x_2, \cdots, x_r)^{\mathrm{T}}$, n 是不等式约束数, p 是等式约束数(约束可以是线性的或非线性的). 如果用 F 表示可行区域, S 表示整个搜索空间, 则 $F\subseteq S$.

3 基于混合策略求解约束优化问题的基本思想

工程应用领域中存在许多约束优化问题, 其最优解位于约束边界上或附近, 即在最优点处不等式约束的全部或大部分取为等号. 因此, 在 CMSEP 算法中, 为了求解位于边界附近的全局最优解, 种群中允许部分最好的不可行解个体进入下一代种群.

3.1 约束比较规则

为避免罚函数法选择因子存在的缺陷, 用如下的方法进行约束处理[4, 7], 规则如下:

(1) 当两个解个体都是可行解时, 适应值 $f(x)$ 高的解为优;

(2) 当两个解之中有一个可行而另一个不可行时, 则可行解个体为优;

(3) 如两个解个体都是不可行解时, 则违反约束程度小的个体为优.

3.2 混合变异策略

演化规划中已提出了许多变异策略来产生新个体[8~11], 如 Gaussian, Cauchy, Levy 和单点等. Gaussian 变异是传统演化规划(CEP)的标准变异, 1999 年 Yao 等人利用 Cauchy 分布代替演化规划中的正态分布[8], 提出了快速演化规划(FEP). Iwamatsu 利用 Levy 分布代替正态分布提出了推广演化规划(GEP)[9]. Lee 等人将 Levy 分布应用在演化规划中[10]. Ji 等人提出了单点变异演化规划[11]. 上述实验研究表明, 对典型的多峰函数优化问题, FEP 比通常的 CEP 有更好的演化特性; Levy 变异与 Cauchy 变异算子相比, 更具通用性和灵活性; 单点变异中由于每次对当前解的一个分量进行变异, 因此算法时间消耗相对较少稳定性强, 比其他 EP 更易收敛到全局最优解. 混合变异策略是基于博弈论的思想, 组合 4 种变异策略的优势, 通过调整演化过程中个体变异策略的概率分布产生子代, 使算法的性能更通用、有效和稳定[12].

4 约束优化算法

算法 1. 基于混合策略求解约束问题的演化规划算法

(1) 纯策略集合记为{1, 2, 3, 4}, 分别代表

Gaussian, Cauchy, Levy, 单点变异.

(2) 初始化:

① 产生由μ个个体组成的初始种群, 每个个体代表一个实值向量集合$(\boldsymbol{x}_i, \sigma_i)$, $i=1, \cdots, \mu$, $\boldsymbol{x}_i$ 和 σ_i 有m 个独立分量:

$$\begin{matrix} \boldsymbol{x}_i = \{x_i(1), x_i(2), \cdots, x_i(m)\}, \\ \sigma_i = \{\sigma_i(1), \sigma_i(2), \cdots, \sigma_i(m)\}, \end{matrix} \quad i = 1, \cdots, \mu.$$

② 对每个个体 i, 为混合策略向量分配初始的概率分布:

$$\rho_i = (\rho_i(1), \rho_i(2), \rho_i(3), \rho_i(4)),\ i = 1, \cdots, \mu.$$

(3) 变异:对每个个体 i, 根据混合策略向量 ρ 中的概率选择一种变异策略h, 根据选择的策略进行变异, 产生子代. 相关变异策略如下:

Gaussian 变异: 每个父代$(\boldsymbol{x}_i^{(k)}, \sigma_i^{(k)})$按如下方法产生一个子代$(\boldsymbol{x}_i^{(k+1)}, \sigma_i^{(k+1)})$:

$$\begin{cases} \sigma_i^{(k+1)}(j) = \sigma_i^k(j)\exp\{\tau N(0,1) + \tau' N_j(0,1)\}, \\ \quad j = 1, 2, \cdots, m, \\ x_i^{(k+1)}(j) = x_i^k(j) + \sigma_i^{(k+1)}(j) N_j(0,1), \\ \quad j = 1, 2, \cdots, m, \end{cases} \tag{4}$$

$N(0, 1)$代表标准 Gaussian 随机变量, 对于一个给定的个体 i, 它是固定的; $N_j(0, 1)$是每个 j 分量的标准 Gaussian 随机变量; 参数 τ 和 τ'定义如下(μ为种群数):

$$\tau = 1/\sqrt{2\mu},\ \tau' = 1/\sqrt{2\sqrt{\mu}}.$$

Cauchy 变异: 每个父代$(\boldsymbol{x}_i^{(k)}, \sigma_i^{(k)})$按如下方法产生一个子代$(\boldsymbol{x}_i^{(k+1)}, \sigma_i^{(k+1)})$:

$$\begin{cases} \sigma_i^{(k+1)}(j) = \sigma_i^k(j)\exp\{\tau N(0,1) + \tau' N_j(0,1)\}, \\ \quad j = 1, 2, \cdots, m, \\ x_i^{(k+1)}(j) = x_i^k(j) + \sigma_i^{(k+1)}(j)\,\delta_j, \\ \quad j = 1, 2, \cdots, m, \end{cases} \tag{5}$$

δ_j代表标准Cauchy 随机变量, 不同的分量j 产生不同的 δ_j.

Levy 变异: 每个父代$(\boldsymbol{x}_i^{(k)}, \sigma_i^{(k)})$按如下方法产生一个子代$(\boldsymbol{x}_i^{(k+1)}, \sigma_i^{(k+1)})$:

$$\begin{cases} \sigma_i^{k+1}(j) = \sigma_i^k(j)\exp\{\tau N(0,1) + \tau' N_j(0,1)\}, \\ \quad j = 1, 2, \cdots, m, \\ x_i^{(k+1)}(j) = x_i^k(j) + \sigma_i^{(k+1)}(j) L_j(\beta), \\ \quad j = 1, 2, \cdots, m, \end{cases} \tag{6}$$

$L_j(\beta)$是遵循 Levy 分布的随机数, $\beta=0.8$.

单点变异: 每个父代$(\boldsymbol{x}_i^{(k)}, \sigma_i^{(k)})$按如下方法产生一个子代$(\boldsymbol{x}_i^{(k+1)}, \sigma_i^{(k+1)})$:

$$\begin{cases} \sigma_i^{(k+1)}(j) = \sigma_i^k(j)\exp(-\alpha), \\ x_i^{(k+1)}(j) = x_i^k(j) + \sigma_i^{(k+1)}(j) N_j(0,1), \end{cases} \tag{7}$$

其中, j 是从集合$\{1, \cdots, m\}$中随机选取的. 这里$\alpha=1.01$, $\sigma_i(j)$的初值等于$0.5(b_j - a_j)$. 如果 $\sigma_i(j) < 10^{-4}$, 那么 $\sigma_i(j) = 0.5(b_j - a_j)$.

(4) μ个父代产生μ个子代, 计算它们的适应度值.

(5) 选择下一代种群:

if 每代 parents, offspring 联合种群中可行个体比例超过 97% then

基于比较准则, 从 parents, offspring 联合种群中选择最好的个体(97 个),

同时从 parents, offsprings 联合种群中选择最好的不可行个体(3 个)将它们添加到下一代种群.

else

从 parents, offspring 联合种群中选择可行的个体, 同时从联合种群中选择最好的不可行个体直至达到种群数, 将它们添加到下一代种群.

end if

(6) 按如下方法更新子代混合策略概率(最小化):

根据约束比较规则, 并且如果个体来自于子代种群, 采用的变异纯策略为 h, $h\in\{1, 2, 3, 4\}$, 要加强这个纯策略:

$$\begin{cases} \rho_{ih}^{(k+1)} = \rho_{ih}^{(k)} + (1 - \rho_{ih}^{(k)}) \times \gamma, \\ \rho_{il}^{(k+1)} = \rho_{il}^{(k)} - \rho_{il}^{(k)} \times \gamma,\ \forall l \neq h, \end{cases} \tag{8}$$

这里, $0<\gamma<1$ 用来调整混合策略的概率分布, $\gamma=1/3$.

如果个体来自于父代种群, 采用的变异纯策略为 h, $h\in\{1, 2, 3, 4\}$, 要减弱这个纯策略:

$$\begin{cases} \rho_{ih}^{(k+1)} = \rho_{ih}^{(k)} - \rho_{ih}^{(k)} \times \gamma, \\ \rho_{il}^{(k+1)} = \rho_{il}^{(k)} + \dfrac{1}{m-1} \times \gamma \times \rho_{il}^{(k)},\ \forall l \neq h. \end{cases} \tag{9}$$

(7) 重复步骤(3)~(6), 直至满足终止条件.

5 实验结果分析

5.1 实验结果

为测试 CMSEP 算法性能, 选取 13 个标准测试函数进行对比实验[7]. 这些测试函数包括多种复杂性质的约束优化问题, 约束条件有等式约束、不等式约束以及两者的混合. 详细的函数描述见附录.

对于等式约束优化问题, 采用一种动态处理机制, 这种动态处理机制首先由 Hamida 等人在适应

分割约束处理算法(ASCHEA)中提出[13],并应用于SMES[7]. 偏差值 ε 随着种群代数的递增逐渐下降,其表达式如下:

$$\varepsilon_j(t+1)=\varepsilon_j(t)/C, \qquad (10)$$

其中 C 表示常数,在函数 $g03$, $g05$ 和 $g11$ 中 C 的值为 1.00195,在函数 $g13$ 中 C 的值为 1.6. 算法运行种群数是 100、运行代数为 1000,CMSEP 算法的其他相关参数如表 1 所示. 表 2 列出了新算法 CMSEP 对 13 个测试函数实验的统计结果.

对每个问题在相同条件下独立运行 30 次,记录其最好结果、平均结果和最差结果. 为了对比实验结果,将 CMSEP 运算结果与其他 4 种较新算法——SMES[9],ASCHEA[13]、同态映射(HM)[14]以及随机排序(SR)[15]做了比较. 表 3 至表 5 分别列出了 5 种算法在 13 个测试函数实验中运行得最好、均值和最差结果比较,这些实验结果均来自相关算法的参考文献[7, 13~15].

Table 1 Experiment Parameter of CMSEP

表 1 CMSEP 算法运行过程中的相关参数

Problem	α	σ	ε(0)
g01	1.01	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g02	0.008	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g03	2.01	$0.05\times(x_i^u-x_i^l)/\sqrt{n}$	0.001
g04	1.01	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g05	0.001	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	0.001
g06	0.01	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g07	0.005	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g08	1.01	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g09	0.001	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g10	0.015	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g11	0.09	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	0.001
g12	1.01	$0.4\times(x_i^u-x_i^l)/\sqrt{n}$	
g13	1.01	$0.025\times(x_i^u-x_i^l)/\sqrt{n}$	3.5

Table 2 Statistical Results Obtained by Our CMSEP for the 13 Test Functions over 30 Independent Runs

表 2 CMSEP 算法对 13 个标准测试函数实验的统计结果

Problem	Statistical Results of the Constrained Mixed Strategies Evolutionary Programming					
	Optimal	Best	Mean	Median	Worst	St. Dev.
g01	−15.000	**−15.000**	**−15.000**	**−15.000**	**−15.000**	0
g02	0.803619	0.769563	0.753077	0.753502	0.730705	1.10E-2
g03	1.000	**1.000**	**1.000**	**1.000**	**1.000**	1.77E-4
g04	−30665.539	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	0
g05	5126.498	5126.509	5141.957	5142.506	5152.217	6.351
g06	−6961.814	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	1.12E-4
g07	24.30621	24.817	25.44257	25.45584	25.90297	3.08E-01
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**	0
g09	680.63	680.6341	680.648	680.6493	680.656	7.16E-03
g10	7049.25	7051.585	7233.773	7252.034	7352.635	74.185
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	1.28E-4
g12	1.000	**1.000**	**1.000**	**1.000**	**1.000**	0
g13	0.05395	0.0566	0.0772	0.073	0.154004	2.25E-02

Note: A result in boldface indicates that the global optimum (or best known solution) has been reached.

Koziel 等人提出了同态映射(HM)解约束优化算法[14],HM 在 n 维立方体空间和可行搜索空间执行一个同态映射,这种方法的主要思想是将原问题转换为一个容易被 EA 优化的拓扑等价函数. HM 处理两种情况:凸的可行空间和非凸可行空间. HM 创新之处在于采用遗传算法的二进制 gray 编码、无杰出者的比例选择机制、传统的交叉和变异操作. Runarsson 等人基于(μ, λ)演化策略提出了随机排序约束优化算法(SR)[15]. SR 不要求定义罚系数,通过一个随机冒泡排序过程实现选择过程. SR 的目标是平衡目标函数和罚函数对解的适应度的影响. Hamida 等人提出了适应分割约束处理算法(ASCHEA)[13],ASCHEA 主要贡献包括 3 部分内容:①一个适应罚函数;②一个约束驱动重组;③基于可行的分离选择机制. ASCHEA 算法通过一个(100+300)演化策略实现.

Table 3 Comparison of the Best Solutions Found by Our CMSEP Against the Homomorphous Maps (HM), Stochastic Ranking (SR), ASCHEA and SMES

表 3 新算法 CMSEP 与算法 HM, SR, ASCHEA 和 SMES 最好解的实验结果比较

Problem	Comparison of the Best Solution Obtained					
	Optimal	HM	SR	ASCHEA	SMES	CMSEP
g01	−15.000	−14.7886	**−15.000**	**−15.000**	**−15.000**	**−15.000**
g02	0.803619	0.79953	0.803515	0.785	0.803601	0.769563
g03	1.000	0.9997	**1.000**	**1.000**	**1.000**	**1.000**
g04	−30665.539	−30664.5	**−30665.539**	−30665.5	**−30665.539**	**−30665.539**
g05	5126.498	—	5126.497	5126.5	5126.599	5126.509
g06	−6961.814	−6952.1	**−6961.814**	−6961.81	**−6961.814**	**−6961.814**
g07	24.306	24.620	24.307	24.3323	24.327	24.817
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.91	**680.630**	**680.630**	680.632	680.6341
g10	7049.25	7147.9	7054.316	7061.13	7051.903	7051.585
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	1.000	0.999999857	**1.000**	NA	**1.000**	**1.000**
g13	0.053950	NA	0.053957	NA	0.053986	0.0566

Note: A result in boldface indicates a better result or that the global optimum (or best known solution) has been reached. "—" means that no feasible solutions have been found. NA= not available.

Table 4 Comparison of the Mean Solutions Found by Our CMSEP Against the Homomorphous Maps (HM), Stochastic Ranking (SR), ASCHEA and SMES

表 4 新算法 CMSEP 与算法 HM, SR, ASCHEA 和 SMES 均值解的实验结果比较

Problem	Comparison of the Mean Solution Obtained					
	Optimal	HM	SR	ASCHEA	SMES	CMSEP
g01	−15.000	−14.7082	**−15.000**	−14.84	**−15.000**	**−15.000**
g02	0.803619	0.79671	0.781975	0.59	0.785238	0.753077
g03	1.000	0.9989	**1.000**	0.99989	**1.000**	**1.000**
g04	−30665.539	−30655.3	**−30665.539**	−30665.5	**−30665.539**	**−30665.539**
g05	5126.498	—	5128.881	5141.65	5174.492	5141.957
g06	−6961.814	−6342.6	−6875.940	−6961.81	−6961.284	−6961.814
g07	24.306	24.826	24.374	24.66	24.475	25.44257
g08	0.095825	0.0891568	**0.095825**	**0.095825**	**0.095825**	**0.095825**
g09	680.63	681.16	680.656	680.641	680.643	680.648
g10	7049.25	8163.6	7559.192	7193.11	7253.047	7233.773
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	1.000	0.999134613	**1.000**	NA	**1.000**	**1.000**
g13	0.053950	NA	0.057006	NA	0.166385	0.0772

Note: A result in boldface indicates a better result or that the global optimum (or best known solution) has been reached. "—" means that no feasible solutions have been found. NA= not available.

5.2 结果分析

从表 2 可以看出,新算法在 7 个测试函数($g1$, $g3$, $g4$, $g6$, $g8$, $g11$ 和 $g12$)中达到了全局最优解,其他 6 个函数接近全局最优解. 新算法与其他算法的实验结果比较如下:

(1) 新算法 CMSEP 与算法 HM 的比较

CMSEP 在 8 个函数($g1$, $g3$, $g4$, $g5$, $g6$, $g9$, $g10$ 和 $g12$)中找到较好的最优解, 在 2 个函数中

($g8$ 和 $g11$)找到相近的最优解, HM 算法在 2 个函数($g2$ 和 $g7$)中找到较好的最优解. 此外, 新算法在 9 个函数($g1$, $g3$, $g4$, $g5$, $g6$, $g8$, $g9$, $g10$ 和 $g12$)中找到较好的均值解和最差解, 在 1 个函数中($g11$)找到与 HM 算法相近的均值和最差解, HM 算法在 2 个函数($g2$ 和 $g7$)中找到较好的均值和最差解. 函数 $g13$ 没有比较, 因为 $g13$ 没有在 HM 算法的计算数据.

(2) 新算法 CMSEP 与算法 SR 的比较

新算法在函数 $g10$ 中找到较好的最优解, 在 7 个函数($g1$, $g3$, $g4$, $g6$, $g8$, $g11$ 和 $g12$)中找到与 SR 相近的最优解, SR 算法在 5 个函数($g2$, $g5$, $g7$, $g9$ 和 $g13$)中找到较好的最优解. 新算法在 3 个函数($g6$, $g9$ 和 $g10$)中找到较好的均值, 在 6 个函数($g1$, $g3$, $g4$, $g8$, $g11$ 和 $g12$)中找到相近的均值解, SR 算法在 4 个函数($g2$, $g5$, $g7$ 和 $g13$)中找到较好的均值. 新算法在 5 个函数($g2$, $g6$, $g9$, $g10$ 和 $g13$)中找到较好的最差解, 在 6 个函数($g1$, $g3$, $g4$, $g8$, $g11$ 和 $g12$)中找到相近的最差解, SR 算法在 2 个函数($g5$ 和 $g7$)中找到较好的最差解.

Table 5 Comparison of the Worst Solutions Found by Our CMSEP Against the Homomorphous Maps (HM), Stochastic Ranking (SR), ASCHEA and SMES

表 5 新算法 CMSEP 与算法 HM, SR, ASCHEA 和 SMES 最差解的实验结果比较

Problem	Comparison of the Worst Solution Obtained					
	Optimal	HM	SR	ASCHEA	SMES	CMSEP
$g01$	−15.000	−14.6154	**−15.000**	NA	**−15.000**	**−15.000**
$g02$	0.803619	0.79119	0.726288	NA	0.751322	0.730705
$g03$	1.000	0.9978	**1.000**	NA	**1.000**	**1.000**
$g04$	−30665.539	−30645.9	**−30665.539**	NA	**−30665.539**	**−30665.539**
$g05$	5126.498	−	5142.472	NA	5304.167	5152.217
$g06$	−6961.814	−5473.9	−6350.262	NA	−6952.482	−6961.814
$g07$	24.306	25.069	24.642	NA	24.843	25.90297
$g08$	0.095825	0.0291438	**0.095825**	NA	**0.095825**	**0.095825**
$g09$	680.63	683.18	680.763	NA	680.719	680.656
$g10$	7049.25	9659.3	8835.655	NA	7638.366	7352.635
$g11$	0.75	**0.75**	**0.75**	NA	**0.75**	**0.75**
$g12$	1.000	0.991950498	**1.000**	NA	**1.000**	**1.000**
$g13$	0.053950	NA	0.216915	NA	0.468294	0.154004

Note: A result in boldface indicates a better result or that the global optimum (or best known solution) has been reached. "−" means that no feasible solutions have been found. NA= not available.

(3) 新算法 CMSEP 与算法 ASCHEA 的比较

新算法在 3 个函数($g4$, $g6$ 和 $g10$)中找到较好的最优解, 在 4 个函数($g1$, $g3$, $g8$ 和 $g11$)中找到相近的最优解, ASCHEA 算法在 4 个函数($g2$, $g5$, $g7$ 和 $g9$)中找到较好的最优解. ASCHEA 算法在 2 个函数($g12$ 和 $g13$)中没有比较数据. 新算法在 5 个函数($g1$, $g2$, $g3$, $g4$ 和 $g6$)中找到较好的均值, 在 2 个函数($g8$ 和 $g11$)中找到相近的均值解, ASCHEA 算法在 4 个函数($g5$, $g7$, $g9$ 和 $g10$)中找到较好的均值解. ASCHEA 算法在 2 个函数($g12$ 和 $g13$)中没有比较数据. ASCHEA 算法没有最差解的数据

(4) 新算法 CMSEP 与算法 SMES 的比较

新算法在 2 个函数($g5$ 和 $g10$)中找到较好的最优解, 在 7 个函数($g1$, $g3$, $g4$, $g6$, $g8$, $g11$ 和 $g12$)中找到相近的最优解, SMES 算法在 4 个函数($g2$, $g7$, $g9$ 和 $g13$)中找到较好的最优解. 新算法在 4 个函数($g5$, $g6$, $g10$ 和 $g13$)中找到较好的均值解, 在 6 个函数($g1$, $g3$, $g4$, $g8$, $g11$ 和 $g12$)中找到相近的均值解, SMES 算法在 3 个函数($g2$, $g7$ 和 $g9$)中找到较好的均值解. 新算法在 5 个函数($g5$, $g6$, $g9$, $g10$ 和 $g13$)中找到较好的最差解, 在 6 个函数($g1$, $g3$, $g4$, $g8$, $g11$ 和 $g12$)中找到相近的最差解, SMES 算法在 2 个函数($g2$ 和 $g7$)中找到较好的最差解.

从比较结果来看, 新算法对于 $g1$, $g3$, $g4$, $g6$,

$g8$, $g11$ 和 $g12$ 等函数的运算结果在最优解、均值解和最差解诸方面都达到了全局最优解，其他 6 个函数也得到了较好的最优解、均值解和最差解，上述比较结果充分显示了新算法是一种便于实现、通用性强、具有较好稳定性的算法.

6 结 论

约束条件处理是使用 EA 求解约束优化问题所面临的一个关键问题. 提出了基于混合策略求解约束优化问题的演化规划算法. 算法采用一种个体比较准则，允许部分最好的不可行解进入下一代种群，混合 4 种变异策略指导演化规划快速搜索. 数值实验显示新算法是一种便于实现、通用性强、性能稳定的算法. 进一步的工作是继续改进提出的算法，提高其性能并将算法推广到其他约束优化问题.

参 考 文 献

1 Z. Michalewicz, M. Schoenauer. Evolutionary algorithms for constraint parameter optimization problems. Evolutionary Computation, 1996, 4(1): 1~32

2 R. Le Riche, C. K. Lenoir, R. T. Haftka. A segregated genetic algorithm for constrained structural optimization. In: Proc. 6th Int'l Conf. Genetic Algorithms. San Francisco: Morgan Kaufmann, 1995

3 D. Powell, M. M. Skolnick. Using genetic algorithms in engineering design optimization with nonlinear constraints. The 5th Int'l Conf. Genetic Algorithms (ICGA-93), SanMateo, CA, 1993

4 K. Deb, S. Agrawal. A niched-penalty approach for constraint handling in genetic algorithms. The ICANNGA-99, Portoroz, Slovenia, 1999

5 Zhou Yuren, Li Yuanxiang, Wang Yong, *et al*. A Pareto strength evolutionary algorithm for constrained optimization. Journal of Software, 2003, 14(7): 1243~1249 (in Chinese)
(周育人，李元香，王勇，等. Pareto 强度值演化算法求解约束优化问题. 软件学报，2003, 14(7): 1243~1249)

6 Haralambos Sarimveis. A line up evolutionary algorithm for solving nonlinear constrained optimization problems. Computers and Operations Research, 2005, 32(6): 1499~1514

7 Efrén Mezura-Montes, A. Carlos. Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans. Evolutionary Computation, 2005, 9(1): 1~17

8 Xin Yao, Yong Liu, Guangming Lin. Evolutionary programming made faster. IEEE Trans. Evolutionary Computation, 1999, 3(2): 82~102

9 Masao Iwamatsu. Generalized evolutionary programming with Lévy-type mutation. Computer Physics Communications, 2002, 147(8): 729~732

10 C-Y. Lee, X. Yao. Evolutionary programming using mutations based on the levy probability distribution. IEEE Trans. Evolutionary Computation, 2004, 8(1): 1~13

11 Mingjun Ji, Huanwen Tang, Juan Guo. A single-point mutation evolutionary programming. Information Processing Letters, 2004, 90(2): 293~299

12 Hongbin Dong, Houkuan Huang, Jun He, *et al*. A mixed mutation strategies evolutionary programming based species conservation for function optimization. The 4th Mexican Int'l Conf. Artificial Intelligence, Mexican, 2005

13 S. B. Hamida, M. Schoenauer. ASCHEA: New results using adaptive segregational constraint handling. Congress on Evolutionary Computation, Hawaii, USA, 2002

14 S. Koziel, Z. Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evolutionary Computation, 1999, 7(1): 19~44

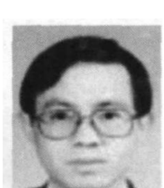
15 T. P. Runarsson, X. Yao. Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evolutionary Computation, 2000, 4(3): 284~294

Dong Hongbin, born in 1963. Professor. Currently Ph. D. candidate of Beijing Jiaotong University, Beijing, China, majoring in computer application technology. His main research interests are artificial intelligence, multi-agent system, evolutionary computation, etc.

董红斌，1963 年生，博士研究生，教授，主要研究方向为人工智能、多智能体系统和演化计算等.

Huang Houkuan, born in 1940. Currently professor and Ph. D. supervisor of Beijing Jiaotong University. His main research interests are artificial intelligence, machine learning, data warehousing, data mining, multi-agent system, etc.

黄厚宽，1940 年生，教授，博士生导师，主要研究方向为人工智能、机器学习、数据仓库、数据挖掘和多智能体系统等.



He Jun, born in 1967. Ph. D. He is currently a research fellow with the School of Computer Science, University of Birmingham, Birmingham, U. K. His main research interests include evolutionary computation, network security and parallel algorithms.

何军，1967 年生，博士，现英国 Birmingham 大学计算机科学学院 Research Fellow. 主要研究方向为演化计算、网络安全和并行算法等.

Hou Wei, born in 1973. Received her M.S.' degree from Harbin Normal University in 2005. Her main research interests include evolutionary computation.
侯薇, 1973 年生, 硕士, 助教, 主要研究方向为演化计算

Research Background

Evolutionary algorithms (EAs) have been widely used to solve several types of optimization problems. Nevertheless, they are unconstrained search techniques and lack an explicit mechanism to bias the search in constrained search spaces. This has motivated the development of a considerable number of approaches to incorporate constraints into the fitness function of an EA. This paper presents a mixed strategies evolutionary programming to solve constrained optimization problems. The approach does not require the use of a penalty function. Instead, it uses a diversity conservation mechanism based on allowing infeasible solutions to remain in the population. A mixed mutation strategy and feasibility-based comparison mechanism is used to guide the process fast toward the feasible region of the search space. This mixed strategy distribution is dynamically adjusted based on the performance of mutation strategies. This new evolutionary programming has been tested on 13 benchmark functions. The results obtained show that the new approach is a general and effective method. Its performance outperforms some other techniques. Our work is supported by the National Natural Science Foundation of China (60443003).

附录 A 标准测试函数

(1) g01

Maximize $f(\boldsymbol{x})=5\sum_{i=1}^{4}x_i-5\sum_{i=1}^{4}x_i^2-\sum_{i=5}^{13}x_i$.

Subject to

$g_1(\boldsymbol{x})=2x_1+2x_2+x_{10}+x_{11}-10\leqslant 0$,

$g_2(\boldsymbol{x})=2x_1+2x_3+x_{10}+x_{12}-10\leqslant 0$,

$g_3(\boldsymbol{x})=2x_2+2x_3+x_{11}+x_{12}-10\leqslant 0$,

$g_4(\boldsymbol{x})=-8x_1+x_{10}\leqslant 0$,

$g_5(\boldsymbol{x})=-8x_2+x_{11}\leqslant 0$,

$g_6(\boldsymbol{x})=-8x_3+x_{12}\leqslant 0$,

$g_7(\boldsymbol{x})=-2x_4-x_5+x_{10}\leqslant 0$,

$g_8(\boldsymbol{x})=-2x_6-x_7+x_{11}\leqslant 0$,

$g_9(\boldsymbol{x})=-2x_8-x_9+x_{12}\leqslant 0$,

这里, $0\leqslant x_i\leqslant 1(i=1,\cdots,9)$, $0\leqslant x_i\leqslant 100(i=10,11,12)$, 和 $0\leqslant x_{13}\leqslant 1$. 全局最优解为

$x^*=(1,1,1,1,1,1,1,1,1,3,3,3,1)$,

$f(x^*)=-15$.

(2) g02

Maximize $f(\boldsymbol{x})=$

$$\left|\frac{\sum_{i=1}^{n}\cos^4(x_i)-2\prod_{i=1}^{n}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{n}ix_i^2}}\right|.$$

Subject to

$$g_1(\boldsymbol{x})=0.75-\prod_{i=1}^{n}x_i\leqslant 0,$$

$$g_2(\boldsymbol{x})=\sum_{i=1}^{n}x_i-7.5n\leqslant 0,$$

这里, $n=20$ 和 $0\leqslant x_i\leqslant 10(i=1,\cdots,10)$, 已知最优解为 $f(x^*)=0.803619$, $g_1=-10^{-8}$.

(3) g03

Maximize $f(\boldsymbol{x})=(\sqrt{n})^n\prod_{i=1}^{n}x_i$.

Subject to $h(\boldsymbol{x})=\sum_{i=1}^{n}x_i^2-1=0$,

这里, $n=10$ 且 $0\leqslant x_i\leqslant 1(i=1,\cdots,n)$. 全局最优解是 $x_i^*=1/\sqrt{n}(i=1,\cdots,n)$, $f(x^*)=1$.

(4) g04

Minimize $f(x)=$

$5.3578547x_3^2+0.8356891x_1x_5+37.293239x_1-40792.141$.

Subject to

$g_1(\boldsymbol{x})=85.334407+0.0056858x_2x_5+0.0006262x_1x_4-0.0022053x_3x_5-92\leqslant 0$,

$g_2(\boldsymbol{x})=-85.334407-0.0056858x_2x_5-0.0006262x_1x_4+0.0022053x_3x_5\leqslant 0$,

$g_3(\boldsymbol{x})=80.51249+0.0071317x_2x_5+0.0029955x_1x_2+0.0021813x_3^2-110\leqslant 0$,

$g_4(\boldsymbol{x})=-80.51249-0.0071317x_2x_5-0.0029955x_1x_2-0.0021813x_3^2+90\leqslant 0$,

$g_5(\boldsymbol{x})=9.300961+0.0047026x_3x_5+0.0012547x_1x_3-0.0019085x_3x_4-25\leqslant 0$,

$g_6(\boldsymbol{x})=-9.300961-0.0047026x_3x_5-$

$0.0012547x_1x_3-0.0019085x_3x_4+20\leqslant 0$,

这里, $78\leqslant x_1\leqslant 102$, $33\leqslant x_2\leqslant 45$, $27\leqslant x_i\leqslant 45$ $(i=3,4,5)$,

最优解为 $x^*=(78, 33, 29.995256025682, 45, 36.775812905788)$,

$f(x^*)=-30665.539$.

(5) g05

Maximize $f(x)=3x_1+0.000001x_1^3+2x_2+\left(\frac{0.000002}{3}x_2^3\right)$.

Subject to

$g_1(x)=-x_4+x_3-0.55\leqslant 0$,

$g_2(x)=-x_3+x_4-0.55\leqslant 0$,

$h_3(x)=1000\sin(-x_3-0.25)+1000\sin(-x_4-0.25)+894.8-x_1=0$,

$h_4(x)=1000\sin(x_3-0.25)+1000\sin(x_3-x_4-0.25)+894.8-x_2=0$,

$h_5(x)=1000\sin(x_4-0.25)+1000\sin(x_4-x_3-0.25)+1294.8=0$,

这里, $0\leqslant x_1\leqslant 1200$, $0\leqslant x_2\leqslant 1200$, $-0.55\leqslant x_3\leqslant 0.55$, $-0.55\leqslant x_4\leqslant 0.55$. 已知最优解为

$x^*=(679.9453, 1026.067, 0.1188764, -0.3962336)$,

$f(x^*)=5126.4981$.

(6) g06

Maximize $f(x)=(x_1-10)^3+(x_2-20)^3$.

Subject to

$g_1(x_1-5)^2-(x_2-5)^2+100\leqslant 0$,

$g_2(x)=-(x_1-6)^2-(x_2-5)^2-8281\leqslant 0$,

这里, $13\leqslant x_1\leqslant 100$, $0\leqslant x_2\leqslant 100$. 最优解为

$x^*=(14.095, 0.84296)$,

$f(x^*)=-6961.81388$.

(7) g07

Minimize $f(x)=x_1^2+x_2^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+4(x_4-5)^2+(x_5-3)^2+2(x_6-1)^2+5x_7^2+7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45$.

Subject to

$g_1(x)=-105+4x_1+5x_2-3x_7+9x_8\leqslant 0$,

$g_2(x)=10x_1-8x_2-17x_7+2x_8\leqslant 0$,

$g_3(x)=-8x_1+2x_2+5x_9-2x_{10}-12\leqslant 0$,

$g_4(x)=3(x_1-2)^2+4(x_2-3)^2+2x_3^2-7x_4-120\leqslant 0$,

$g_5(x)=5x_1^2+8x_2+(x_3-6)^2-2x_4-40\leqslant 0$,

$g_6(x)=x_1^2+2(x_2-2)^2-2x_1x_2+14x_5-6x_6\leqslant 0$,

$g_7(x)=0.5(x_1-8)^2+2(x_2-4)^2+3x_5^2-x_6-30\leqslant 0$,

$g_8(x)=-3x_1+6x_2+12(x_9-8)^2-7x_{10}\leqslant 0$,

这里, $-10\leqslant x_i\leqslant 10(i=1,\cdots,10)$, 最优解为

$x^*=(2.171996, 2.363683, 8.773926, 5.095984, 0.990548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$,

$f(x^*)=24.3062091$.

(8) g08

Maximize $f(x)=\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$,

Subject to

$g_1(x)=x_1^2-x_2+1\leqslant 0$,

$g_2(x)=1-x_1+(x_2-4)^2\leqslant 0$,

这里, $0\leqslant x_1\leqslant 10$, $0\leqslant x_2\leqslant 10$, 最优解为

$x^*=(1.2279713, 4.2453733)$,

$f(x^*)=0.095825$.

(9) g09

Minimize $f(x)=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x^2x_6+x_7^4-4x_6x_7-10x_6-8x_2$.

Subject to

$g_1(x)=-127+2x_1^2+3x_2^4+x_3+4x_4^2+5x_5\leqslant 0$,

$g_2(x)=-282+7x_1+3x_2+10x_3^2+x_4-x_5\leqslant 0$,

$g_3(x)=-196+23x_1+x_2^2+6x_6^2-8x_7\leqslant 0$,

$g_4(x)=4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7\leqslant 0$,

这里, $-10\leqslant x_i\leqslant 10(i=1,\cdots,7)$. 最优解为

$x^*=(2330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$,

$f(x^*)=680.6300573$.

(10) g10

Maximize $f(x)=x_1+x_2+x_3$.

Subject to

$g_1(x)=-1+0.0025(x_4+x_6)\leqslant 0$,

$g_2(x)=-1+0.0025(x_5+x_7-x_4)\leqslant 0$,

$g_3(x)=-1+0.01(x_8-x_5)\leqslant 0$,

$g_4(x)=-x_1x_6+833.33252x_4+100x_1-83333.333\leqslant 0$,

$g_5(x)=-x_2x_7+1250x_5-x_2x_4-1250x_4\leqslant 0$,

$g_6(\boldsymbol{x})=-x_3x_8+1250000+x_3x_5-2500x_5\leqslant 0$,
这里, $100\leqslant x_1\leqslant 10000$, $1000\leqslant x_i\leqslant 10000(i=2,3)$, $10\leqslant x_i\leqslant 1000(i=4,\cdots,8)$. 最优解为

$x^*=(579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.40, 395.5979)$,

$f(x^*)=7049.25$.

(11) g11

Maximize $f(\boldsymbol{x})=x_1^2+(x_2-1)^2$.

Subject to $h(\boldsymbol{x})=x_2-x_1^2=0$,

这里, $-1\leqslant x_1\leqslant 1$, $-1\leqslant x_2\leqslant 1$, 最优解为

$x^*=(\pm 1/\sqrt{2}, 1/2)$, $f(x^*)=0.75$.

(12) g12

Maximize $f(\boldsymbol{x})=$

$$\frac{100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2}{100}.$$

Subject to

$g_1(\boldsymbol{x})=(x_i-p)^2+(x_2-q)^2-(x_3-r)^2-0.0625\leqslant 0$,

这里, $0\leqslant x_i\leqslant 10$, $(i=1,2,3)$ 且 $p, q, r=1,2,\cdots,9$. 搜索空间的可行区域由 9^3 个不连续的部分组成. 最优解为

$x^*=(5,5,5)$, $f(x^*)=1$.

(13) g13

Maximize $f(\boldsymbol{x})=e^{x_1x_2x_3x_4x_5}$.

Subject to

$h_1(\boldsymbol{x})=x_1^2+x_2^2+x_3^2+x_4^2+x_5^2-10=0$,

$h_2(\boldsymbol{x})=x_2x_3-5x_4x_5=0$,

$h_3(\boldsymbol{x})=x_1^3+x_2^3+1=0$,

这里, $-2.3\leqslant x_i\leqslant 2.3$, $(i=1,2)$, $-3.2\leqslant x_i\leqslant 3.2$, $(i=3,4,5)$, 最优解为

$x^*=(-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$,

$f(x^*)=0.0539498$.