

求解约束优化问题的文化算法研究

黄海燕¹ 顾幸生¹ 刘漫丹¹

摘要 文化算法的主要思想是明确地从进化种群中获得求解问题的知识(即信念)并用于指导搜索过程. 本文提出了一种基于多层信念空间的文化算法, 该算法通过对多层信念空间的择优选用将提取的知识用于提高进化计算性能来解决约束优化问题. 应用实例表明该算法具有较好的结果和较少的计算量.

关键词 文化算法, 约束优化, 多层信念空间
中图分类号 TP18

Research on Cultural Algorithm for Solving Nonlinear Constrained Optimization

HUANG Hai-Yan¹ GU Xin-Sheng¹ LIU Man-Dan¹

Abstract The key idea behind cultural algorithm (CA) is to explicitly acquire problem-solving knowledge (beliefs) from the evolving population and in return apply that knowledge to guide the search. In this paper, we propose a CA based on multilayer belief spaces that selects the best belief space from the multilayer belief spaces so as to apply the extracted knowledge to improve the performance of evolutionary algorithm used for constrained optimization. Examples show that the algorithm produces highly competitive results at a relatively low computational cost.

Key words Cultural algorithm, constrained optimization, multilayer belief spaces

1 引言

文化算法作为一种新的进化算法是由 Reynolds 于 1994 年提出的^[1]. 社会研究人员认为文化是一种将个人以往的经验保存于其中的知识库, 新的个人可以在知识库中学到他没有直接经历的经验知识. 虽然以往的进化计算在许多问题上取得了较好的成绩, 但其研究还只是集中在生物自然选择这一层面上, 而文化算法则提供了一种显性的机制来获取、保存和整合问题求解的知识使种群进化速度超越单纯依靠生物基因遗传的进化速度^[2]. 目前对文化算法的研究已成为计算智能研究的热点, 如文献 [3] 提出了一种局部搜索和文化算法相结合来解决优化问题, 文献 [4] 对文化算法中的知识学习和社会群体进行了研究, 另外还有采用传统进化计算结合文化算法的文献 [5, 6], 并在一些问题上取得了比传统进化算法更好的结果^[7~9].

一般的约束优化问题如式 (1) 所示

$$\begin{aligned} \min f(\mathbf{X}), \mathbf{X} = (x_1, x_2, \dots, x_m) \in \mathbf{R}^m \\ \text{s.t. } h(\mathbf{X}) = 0, g(\mathbf{X}) \leq 0 \end{aligned} \quad (1)$$

上式中, $h(\mathbf{X})$ 代表一组等式约束, $g(\mathbf{X})$ 代表一组不等式约束, 由于约束条件的限制, 这类问题尽管在生产中经常存在, 但却难以求解. 本文深入分析了约束优化问题的特点并与文化算法的优点相结合, 提出了基于多层信念空间的文化算法流程, 经过函数测试比较和丁烯烷化模型的实际应用结果, 显示本文的方法具有较少的计算量及较高的搜索效率.

2 文化算法原理简介

文化算法的框架是由种群空间 (Population space) 和信念空间 (Belief space) 两部分组成的, 如图 1 所示, 前者是基于传统种群的进化; 后者是基于信念文化的进化, 用于知识经验的形成、储存和传播. 两者相对独立但又相互联系. 其中, 函数 $\text{accept}()$ 用于搜集优秀个体的经验知识; $\text{influence}()$ 利用解决问题的知识指导种群空间的进化; $\text{update}()$ 用于更新信念空间; $\text{generate}()$ 是群体操作函数使个体空间得到进化; 函数 $\text{select}()$ 根据规则从新生成个体中选择一部分个体作为下代个体的父辈. 文化算法的基本伪代码如下^[10]:

```
Begin
  t = 0 ;
  Initialize Population POP(t);
  Initialize Belief Space BLF(t);
Repeat
  Evaluate Population POP(t);
  Update(BLF(t), Accept(POP(t)));
  Variation(POP(t), Influence(BLF(t)));
  t=t+1;
  Select(POP(t)from POP(t-1));
Until termination condition achieved
End
```

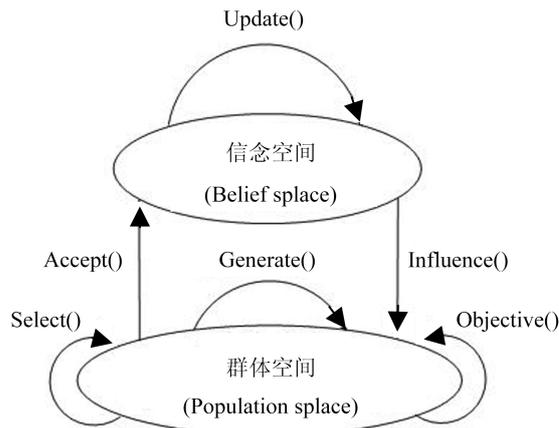


图 1 文化算法框架

Fig. 1 The framework of culture algorithm

3 基于多层信念空间的文化算法设计

Reynolds, Chung 和 Jin 对全局优化问题已经采用文化算法并取得较好的结果^[4, 11, 12]. 他们先后提出了区间模式 (Interval schemata) 和区域模式 (Regional-based schemata) 来表达问题的约束知识, 如图 2 所示, 以两个自变量的约束优化问题为例, 解的域是二维平面内的某个区域, 在搜索过程中提取一个最有可能产生优秀个体的区域, 区域内的曲线

收稿日期 2006-5-12 收修改稿日期 2007-1-15
Received May 12, 2006; in revised form January 15, 2007
1. 华东理工大学自动化研究所 上海 200237
1. Research Institute of Automation, East China University of Science and Technology, Shanghai 200237
DOI: 10.1360/aas-007-1115

表示约束边界, 内部为不可行区域, 外部为可行区域, 把整个区域划分为许多子域称为信念细胞, 其中白色和黑色的信念细胞分别属于可行域和不可行域, 灰色信念细胞属于半可行域, 如图 2(b) 所示. 该区域的边界和内部的约束知识被作为信念保存下来指导以后的搜索. 我们希望在可行域和半可行域内产生更多的个体而不可行域内则抑制产生. 目前许多传统进化计算结合文化算法来求解约束优化问题的文献相继出现^[5, 6, 10], 但都是基于单层信念空间的叠代, 且前后信念空间没有优劣之分, 这样当更新后的信念空间比原来差时就容易陷入局部极值. 因此本文提出在文化算法中建立多层信念空间, 并根据一定的择优原则选出最优的信念空间来指导以后的搜索, 这样做的优点在于优秀的信念知识总是被择优保存下来, 避免进入过早收敛. 此外在接受函数和影响函数的设计上也在前人的基础上作了改进.

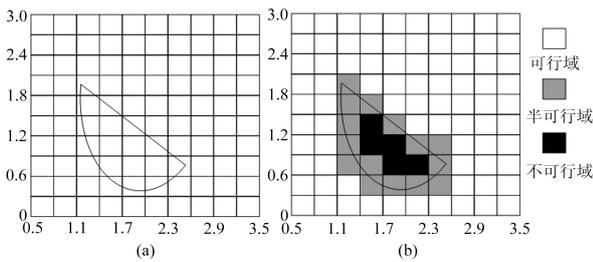


图 2 区域内约束知识的表达

Fig. 2 The expression of constraint knowledge in a region

3.1 多层信念空间的初始化

本文将多层信念空间定义为 $\langle N^H[j], C^H[q], B^H[k] \rangle$, H 表示第 H 层信念空间 (以下类同), $H = 1, 2, \dots, Z$, Z 为信念空间的层数, $N^H[j]$ 用来储存最能产生优秀个体的区域 (称为优胜区域) 的信息, $N^H[j]$ 中包含 $\langle I_j^H, L_j^{t,H}, U_j^{t,H} \rangle$ 三个参数, $j = 1, 2, \dots, n$, n 表示自变量的个数, t 表示第 t 代, 其中参数 I_j^H 表示第 j 个自变量的优胜区间, $I_j^H = [l_j^{t,H}, u_j^{t,H}] = \{x_{i,j}^{t,H} | l_j^{t,H} \leq x_{i,j}^{t,H} \leq u_j^{t,H}, x_{i,j}^{t,H} \in \mathbf{R}\}$, $x_{i,j}^{t,H}$ 表示第 t 代从接受函数选出的第 i 个优秀个体第 j 个自变量, $i = 1, \dots, n_{\text{accepted}}$, n_{accepted} 是优秀个体的数目, I_j^H 初始时设置为变量的定义域, 以后在迭代中不断调整, $L_j^{t,H}$ 、 $U_j^{t,H}$ 分别表示第 t 代第 j 个自变量的下限、上限对应的适应值; $C^H[q]$ 中包含 $Class_q^H, Cnt1_q^H, Cnt2_q^H, P_q^H, Pos_q^H, Csize_q^H$ 六个参数, $q = 1, 2, \dots, m$, m 表示信念细胞的个数, $Class_q^H$ 表示第 H 层信念空间第 q 个信念细胞的类型, $Cnt1_q^H$ 、 $Cnt2_q^H$ 分别用于统计信念细胞内满足和不满足约束条件的个体数目, 初始化为 0. P_q^H 表示信念细胞产生优秀个体的概率, 不同类型的信念细胞有不同的概率, Pos_q^H 表示信念细胞坐标的起始位置, $Csize_q^H$ 表示信念细胞的大小; $B^H[k]$ 中包含 $obj^H(\mathbf{X}_k^H), \mathbf{X}_k^H$ 两个参数, $obj^H(\mathbf{X}_k^H)$ 、 \mathbf{X}_k^H 分别记录种群经 k 代后产生的最佳适应值及其对应的个体.

3.2 接受函数 (Acceptance function)

信念空间中参数的形成和更新是对从种群中选出的适应值优秀的个体群进行统计而得到的, 本文认为随着叠代次数

的增加, 如果每代产生的适应值比前面的更好则减少选取优秀个体的数目, 如果不变则保持原数目, 结果更差则增加数目, 该方法是在 Saleem 等人^[13] 方法的基础上作了改进, 本文中接受函数选出的优秀个体数目由式 (2) 得到

$$n_{\text{accepted}} = \%p * \text{popsize} + \frac{h * (1 - \%p) * \text{popsize}}{g} \quad (2)$$

这里, popsize 表示种群内个体数目, $\%p$ 是根据需要而定的概率, 一般建议选 0.2, g 是代数 (Generation counter), h 是扩大倍数, 当适应值结果在最近的 p 代中变好或没有变化时 $h = 1$; 当结果变差时 $h = 2$, 即增加选取个体数目. 这样使得随着代数的增加适应值越来越好时就逐渐减少计算量, 减少搜索时间, 当适应值不理想时又能够增加个体数目来弥补.

3.3 多层信念空间的更新

多层信念空间的主要思想是在进化过程中同时保留具有最佳适应值记录的前 Z 层信念空间, 实行分层管理, 而每次通过影响函数指导进化的只有一层信念空间, 该层信念空间是在这 Z 层当中通过一定的评价函数来择优选出, 这种“多层空间、择优选用”的策略比以往只有单层信念空间的叠代更趋完善. 因为优秀信念知识不会被“劣质”的信念叠代, 当新的信念空间产生时须通过严格的筛选才能启用, 这样做能够更加准确地接近最优解, 而流程上只是增加了评价择优这个环节, 因此计算量增加很少. 图 3 为多层信念空间的文化算法示意图.

多层信念空间的更新包括两部分:

- 当前的信念空间根据更新规则产生新的信念空间;
- 新的信念空间与已有的 Z 层信念空间经评价函数选出最佳的信念空间来指导种群进化, 同时去掉最差的, 使信念空间仍保持 Z 层.

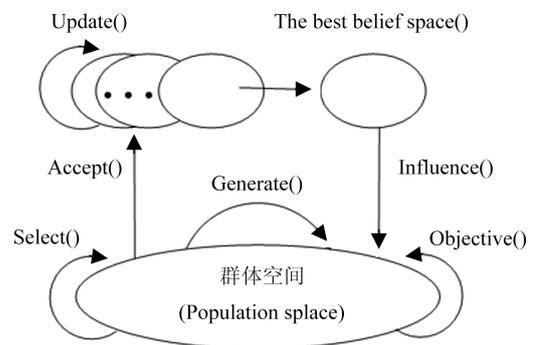


图 3 基于多层信念空间的文化算法框架

Fig. 3 The framework of culture algorithm based on multilayer belief spaces

3.3.1 信念空间中优胜区域的更新规则

假设第 H 层信念空间中第 i 个优秀个体决定优胜区域中第 j 个自变量的下限, 而第 i' 个优秀个体则决定其上限.

更新规则如式 (3)~(6) 所示

$$l_j^{t+1,H} = \begin{cases} x_{i,j}^{t,H}, & \text{若 } x_{i,j}^{t,H} \leq l_j^{t,H} \text{ 或 } \text{obj}(x_{i,j}^{t,H}) < L_j^{t,H} \\ l_j^{t,H}, & \text{否则} \end{cases} \quad (3)$$

$$L_j^{t+1,H} = \begin{cases} \text{obj}(x_{i,j}^{t,H}), & \text{若 } x_{i,j}^{t,H} \leq l_j^{t,H} \text{ 或 } \text{obj}(x_{i,j}^{t,H}) < L_j^{t,H} \\ L_j^{t,H}, & \text{否则} \end{cases} \quad (4)$$

$$u_j^{t+1,H} = \begin{cases} x_{i',j}^{t,H}, & \text{若 } x_{i',j}^{t,H} \geq u_j^{t,H} \text{ 或 } \text{obj}(x_{i',j}^{t,H}) < U_j^{t,H} \\ u_j^{t,H}, & \text{否则} \end{cases} \quad (5)$$

$$U_j^{t+1,H} = \begin{cases} \text{obj}(x_{i',j}^{t,H}), & \text{若 } x_{i',j}^{t,H} \geq u_j^{t,H} \text{ 或 } \text{obj}(x_{i',j}^{t,H}) < U_j^{t,H} \\ U_j^{t,H}, & \text{否则} \end{cases} \quad (6)$$

上式中, $x_{i,j}^{t,H}$ 和 $x_{i',j}^{t,H}$ 分别表示第 H 层第 t 代第 i 和 i' 个优秀个体的第 j 个自变量, $l_j^{t+1,H}$ 和 $L_j^{t+1,H}$ 表示第 H 层第 t 代优秀个体中第 j 个自变量的下限和其适应值, 类似地, $u_j^{t+1,H}$ 和 $U_j^{t+1,H}$ 表示第 j 个自变量的上限和其适应值.

3.3.2 信念空间中约束知识的更新规则

当优胜区域被界定之后, 将其划分为众多规则的子域即信念细胞. 这些信念细胞有的满足约束条件, 有的不满足. 种群中的个体分布在各个细胞内, 若第 H 层信念空间中第 q 个信念细胞内可行解和不可行解的统计数目由计数器 $Cnt1_q^H, Cnt2_q^H$ 来记录, 则细胞类型 $Class_q^H$ 和计数器之间的关系如式 (7) 所示

$$Class_q^H = \begin{cases} \text{unknown}, & \text{若 } Cnt1_q^H = 0 \text{ 且 } Cnt2_q^H = 0 \\ \text{feasible}, & \text{若 } Cnt1_q^H > 0 \text{ 且 } Cnt2_q^H = 0 \\ \text{unfeasible}, & \text{若 } Cnt1_q^H = 0 \text{ 且 } Cnt2_q^H > 0 \\ \text{semifeasible}, & \text{若 } Cnt1_q^H > 0 \text{ 且 } Cnt2_q^H > 0 \end{cases} \quad (7)$$

细胞产生优秀个体的概率 P_q^H 随着细胞类型的改变而更新. $Pos_q^H, Csize_q^H$ 由划分信念细胞的规则或者优胜区域的变化来更新; $B^H[k]$ 随着每一代优秀个体的产生自动更新.

3.3.3 多层信念空间的评价和选择

本文首先从多层信念空间每层的最佳记录中进行适应值比较, 假设经比较得到最优、次优和最差的适应值记录的信念空间分别为第 H 、 V 、 W 层, 则各层的最佳适应值和对应的个体分别为 $\text{obj}^H(\mathbf{X}_{k_h}^H)$ 、 $\mathbf{X}_{k_h}^H$; $\text{obj}^V(\mathbf{X}_{k_v}^V)$ 、 $\mathbf{X}_{k_v}^V$; $\text{obj}^W(\mathbf{X}_{k_w}^W)$ 、 $\mathbf{X}_{k_w}^W$. 则将最佳的第 H 层信念空间中相关信息经 (8), (9) 两式调整作为当前的信念空间来指导以后的搜索, 式中 α 、 β 是两个调整参数, 本文中均选为 0.5, 也可以根据情况调整, 同时将最差的第 W 层信念空间从多层空间中除去. $l_j^{t,H}$ 、 $u_j^{t,H}$ 分别表示第 H 层信念空间中最终保留的经 t 代后第 j 个自变

量对应的下界和上界, 其余符号类同.

$$l_j^{t,H} = \begin{cases} l_j^{t,H}, & \text{若 } l_j^{t,H} \leq l_j^{t,V} \\ l_j^{t,H} + \alpha * (l_j^{t,H} - l_j^{t,V}), & \text{若 } l_j^{t,H} > l_j^{t,V} \end{cases} \quad (8)$$

$$u_j^{t,H} = \begin{cases} u_j^{t,H}, & \text{若 } u_j^{t,H} \geq u_j^{t,V} \\ u_j^{t,H} + \beta * (u_j^{t,H} - u_j^{t,V}), & \text{若 } u_j^{t,H} < u_j^{t,V} \end{cases} \quad (9)$$

3.4 影响函数 (Influence function)

本文中采用基于多层信念空间的文化算法结合进化规划 (CAEP), 采用影响函数来指导进化规划的变异操作主要有两种情况.

1) 父个体在信念空间所界定的优胜区域之外. 我们希望能够向优胜区域靠拢, 变异操作如式 (10), (11) 所示

$$x_{i'',j}^{t+1,H} = x_{i'',j}^{t,H} + |(u_j^{t,H} - l_j^{t,H}) \cdot \text{rand}(1)|, \quad \text{若 } x_{i'',j}^{t,H} < l_j^{t,H} \quad (10)$$

$$x_{i'',j}^{t+1,H} = x_{i'',j}^{t,H} - |(u_j^{t,H} - l_j^{t,H}) \cdot \text{rand}(1)|, \quad \text{若 } x_{i'',j}^{t,H} > u_j^{t,H} \quad (11)$$

其中, $x_{i'',j}^{t,H}$ 表示被选中的第 H 层信念空间第 t 代种群中第 i 个个体的第 j 个自变量, $i'' = 1, \dots, \text{popsize}$, popsize 表示种群内个体数目, $l_j^{t,H}$ 和 $u_j^{t,H}$ 代表第 j 个自变量的下限和上限, $\text{rand}()$ 表示 0~1 之间的随机函数.

2) 父个体在优胜区域之内也分为两种情况, 对于可行域和半可行域内的个体让其继续在区域内或附近变异; 对于不可行域内则将其按一定的概率向四种类型细胞内迁移, 概率大小根据四种类型细胞产生优秀个体的可能性来确定. 具体过程如式 (12) 所示

$$x_{i,j}^{t+1,H} = \begin{cases} \text{moveTo}(\text{choose}(\text{Cell}[m])), & \text{若 } x_{i,j}^{t,H} \in \text{unfeasible} \\ x_{i'',j}^{t,H} + \frac{b}{m_j} \cdot (u_j^{t,H} - l_j^{t,H}) \cdot \text{rand}(1), & \text{否则} \end{cases} \quad (12)$$

式中, m_j 是对应第 j 个自变量的信念细胞个数, b 是位置参数, 根据信念细胞的大小而定, $\text{choose}()$ 是根据不同细胞的概率采用轮盘赌方法选择目标细胞. $\text{moveTo}()$ 是一个迁移函数, 即将属于不可行域的父个体迁往被选出的目标细胞产生后代.

综上, 本文中提出的基于多层信念空间求解约束优化问题的文化算法流程如下:

- Step 1.** $k = 0$;
- Step 2.** 初始化种群空间和相关参数;
- Step 3.** 初始化多层信念空间 $\langle N^H, C^H, B^H \rangle$;
- Step 4.** 评价个体适应值, 符合要求结束, 否则往下执行;
- Step 5.** 更新多层信念空间
 - a) 当前信念空间的更新;
 - b) 多层信念空间评价、选优产生最佳的信念空间;
- Step 6.** 根据信念知识父个体通过影响函数产生后代;
- Step 7.** $k = k + 1$;

Step 8. 通过接受函数从种群空间选出优秀个体;

Step 9. 转到 Step 4, 直到满足终止条件.

4 性能测试

为了验证算法的有效性, 我们选取几个具有代表性的例子^[14] 来进行验证.

$$\begin{aligned} \text{Min } f_1(\mathbf{X}) &= \frac{\sin^3(2\pi x_1)\sin^3(2\pi x_2)}{x_1^3(x_1 + x_2)} \\ \text{s.t. : } g_1 &= x_1^2 - x_2 + 1 \leq 0 \\ g_2 &= 1 - x_1 + (x_2 - 4)^2 \leq 0 \\ 0 &\leq x_2 \leq 10, 0 \leq x_1 \leq 10 \end{aligned} \quad (13)$$

$$\begin{aligned} \text{Min } f_2(\mathbf{X}) &= x_1^2 + (x_2 - 1)^2 \\ \text{s.t. : } h(\mathbf{X}) &= x_2 - x_1^2 = 0, \\ &-1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1 \end{aligned} \quad (14)$$

$$\begin{aligned} \text{Min } f_3(\mathbf{X}) &= 5.3578547x_3^2 + 0.8356891x_1x_5 + \\ &37.293239x_1 - 40792.141 \\ \text{s.t. : } g_1 &= 85.334407 + 0.0056858x_2x_5 + \\ &0.0006262x_1x_4 - 0.0022053x_3x_5 - \\ &92 \leq 0 \\ g_2 &= -85.334407 - 0.0056858x_2x_5 - \\ &0.0006260x_1x_4 + 0.0022053x_3x_5 \leq 0 \\ g_3 &= 80.51249 + 0.0071317x_2x_5 + \\ &0.0029955x_1x_2 + 0.0021813x_3^2 - \\ &110 \leq 0 \\ g_4 &= -80.51249 - 0.0071317x_2x_5 - \\ &0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\ g_5 &= 9.300961 + 0.0047026x_3x_5 + \\ &0.0012547x_1x_3 + 0.0019085x_3x_4 - \\ &25 \leq 0 \\ g_6 &= -9.300961 - 0.0047026x_3x_5 - \\ &0.0012547x_1x_3 - 0.0019085x_3x_4 + \\ &20 \leq 0 \end{aligned}$$

$$78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_i \leq 45 \quad (i = 3, 4, 5) \quad (15)$$

上述例子采用本文提出的 CAEP 对其进行求解, 采用 3 层信念空间、种群规模 50、最大代数 1000、 α 和 β 均取 0.5、 $\%p$ 取 0.2, 现将我们的结果与 Koziel 和 Michalewicz 提出的方法^[15] 进行比较, 后者是进化算法中处理约束优化问题较为有效和前沿的一种方法. 如下页表 1 所示, 采用 Koziel 和 Michalewicz 的方法需要对适应值函数进行 1 400 000 次

计算, 而本文的方法只需计算 50 050 次即可, 相比之下计算量大大减少, 且所得结果相对较好, 主要原因在于我们在进化过程中提取了求解问题的约束知识, 找到了能产生优秀个体的区域来指导进化的过程, 从而避免不满足约束条件和适应值情况差的区域的搜索, 将搜索非常迅速地集中到满足约束条件的优胜区域上.

5 CAEP 用于丁烯烷化过程的约束优化

将 CAEP 应用于丁烯烷化过程的约束优化. 过程流程如图 4 所示. 先将含 100% 丁烯的烯烃进料、循环纯异丁烷、100% 的异丁烷补充流和新鲜的酸催化剂引入反应器, 再将废酸从反应器内移走, 反应产物还将通过分离器使异丁烷与烷基化物分离^[16]. Adjiman^[16] 等为该过程建立了以收益最大为目标的优化模型, 如式 (16) 所示, x_1 为烯烃流速 (桶/天), x_2 为酸添加速率 (千磅/天), x_3 为烷基化物产率 (桶/天), x_4 为酸强度 (质量分数), x_5 为马达法辛烷值, x_6 为异丁烷补充流与烯烃的比率, x_7 为 F-4 性能值, *Profit* 为收益最大值. 文献 [16] 采用基于 α 的分支定界法 (α -based branch and bound, α BB) 对其求解, α BB 是一种确定性的全局优化算法. 但参数较难确定, 且搜索时间很长. 此后文献 [17] 采用连续约束蚁群优化算法 (Constrained ant colony system, CACS) 又对该模型进行了求解.

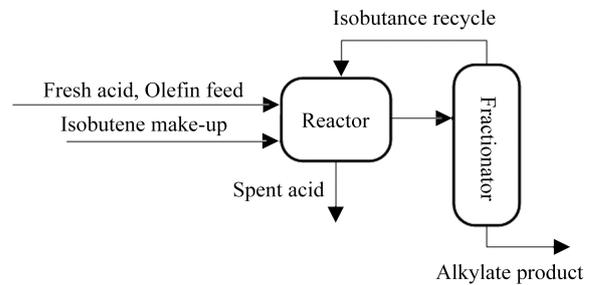


图 4 丁烯烷化流程图

Fig. 4 Alkylation process flowsheet

本文应用 CAEP 对其进行求解, 采用 4 层信念空间、种群规模 50、最大代数 1000、 α 和 β 均取 0.5、 $\%p$ 取 0.2, 图 5(见下页) 为式 (16) 对应目标函数的收敛曲线. 表 2(见下页) 列出了三种方法的运行结果 (另两种方法数据分别取自文献 [16]、[17]), 包括所得的最优解, 相应的对关键约束条件 g_1 、 g_3 、 g_6 的违反情况, 以及最优值的统计结果. 由表 2 可见, 采用 CAEP 得到的最优解求得的收益最大值比采用 α BB 和 CACS 所得的收益最大值 (pb) 和其统计结果即收益最大值的平均值 (pa) 更好, 对于关键约束条件 g_1 、 g_3 、 g_6 采用本文提出的 CAEP 能够全部满足, 采用 CACS 不满足约束条件 g_3 , 而 α BB 则严重违反了约束条件 g_6 ; 表 2 中, CAEP 采用的最大代数为 1000, 而 CACS 采用的最大代数为 2000, 因此计算量大大减少. CAEP 对约束优化问题的实用性、高效性和全局优化性能均令人满意.

表 1 KM 和 CAEP 的结果比较
Table 1 Comparison of results by KM and CAEP

Function	Exact optimum	Best (CAEP)	Best (KM)	Average (CAEP)	Average (KM)	Worst (CAEP)	Worst (KM)
f_1	-0.095 825	-0.095 825	-0.095 345 6	-0.095 345 6	-0.089 156 8	-0.091 240 1	-0.029 143 8
f_2	0.75	0.75	0.75	0.763 846 6	0.75	0.821 033 1	0.75
f_3	-30 665.539	-30 665.1	-30 664.5	-30 657.1	-30 655.3	-30 607.8	-30 645.9

表 2 α BB, CACS 和 CAEP 的结果比较
Table 2 Comparison of results by α BB, CACS and CAEP

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	g_1	g_3	g_6	pb	pa
CAEP	1 699.8	53.321	3 033.1	90.225	95.00	10.485	154.53	-1.216 4	-0.985 3	$-2.241 5 \cdot 10^5$	1 777.1	1 776.3
CACS	1 700.6	54.346	3 033.2	90.183	94.999	10.51	153.53	-0.644	4.588 7	-2 080	1 776.6	1 775.1
α BB	1 698.2	53.66	3 031.3	90.11	95.00	10.50	153.53	0.016 5	4.752 1	1 727.9	1 772.8	-

$$\begin{aligned}
 \text{Profit} &= -\min(1.715x_1 + 0.035x_1x_6 + 4.0565x_3 - 10.0x_2 - 0.063x_3x_5) \\
 \text{s.t. : } g_1 &= 0.0059553571x_6^2x_1 + 0.88392857x_3 - 0.1175625x_6x_1 - x_1 < 0 \\
 g_2 &= 1.1088x_1 + 0.1303533x_1x_6 - 0.0066033x_1x_6^2 - x_3 < 0 \\
 g_3 &= 6.66173269x_6^2 + 172.39878x_5 - 56.596669x_4 - 191.20592x_6 - 10^4 < 0 \\
 g_4 &= 1.08702x_6 + 0.32175x_4 - 0.03762x_6^2 - x_5 + 56.85075 < 0 \\
 g_5 &= 0.006198x_7x_4x_3 + 2462.3121x_2 - 25.125634x_2x_4 - x_3x_4 < 0 \\
 g_6 &= 161.18996x_3x_4 + 5000.0x_2x_4 - 489510.0x_2 - x_3x_4x_7 < 0 \\
 g_7 &= 0.33x_7 - x_5 + 44.333333 < 0 \\
 g_8 &= 0.0225565x_5 - 0.007595x_7 < 1 \\
 g_9 &= 0.00061x_3 - 0.0005x_1 < 1 \\
 g_{10} &= 0.819672x_1 - x_3 + 0.819672 < 0 \\
 g_{11} &= 24500.0x_2 - 250.0x_2x_4 - x_3x_4 < 0 \\
 g_{12} &= 1020.4082x_4x_2 + 1.2244898x_3x_4 - 10^5x_2 < 0 \\
 g_{13} &= 6.25x_1x_6 + 6.25x_1 - 7.625x_3 - 10^5 < 0 \\
 g_{14} &= 1.22x_3 - x_6x_1 - x_1 + 1 < 0
 \end{aligned}$$

$$\begin{aligned}
 1500 < x_1 < 2000, 1 < x_2 < 120, 3000 < x_3 < 3500 \\
 85 < x_4 < 93, 90 < x_5 < 95, 3 < x_6 < 12, 145 < x_7 < 1620 \\
 (16)
 \end{aligned}$$

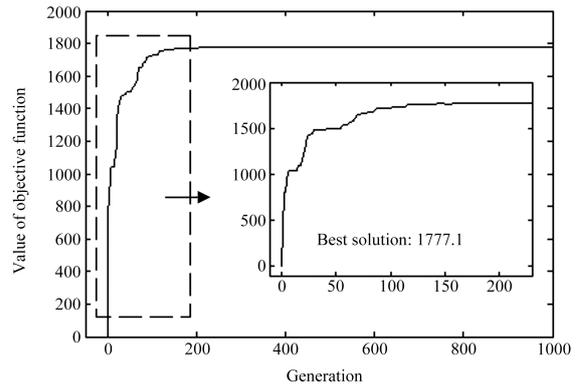


图 5 目标函数的收敛曲线
Fig. 5 Curve of convergence for objective function

6 结论

本文提出了一种基于多层信念空间的文化算法,并结合进化规划的算法流程来解决约束优化问题.将 CAEP 应用于 3 个典型的约束优化测试问题,结果表明这种算法具有较好的结果和相对少的计算量;将 CAEP 应用于丁烯烷化过程取得了令人满意的结果,进一步表明了该算法对约束优化问题的适用性和全局优化性能.

文化算法作为一种新的进化算法仿佛为传统的进化计算提供了指路的明灯,它可以从进化的过程中提取相关的知识信念来指导搜索,避免多走“弯路”,从而大大提高搜索效率.文化算法到目前为止还未形成完整的体系结构,尚需要不断完善.特别当自变量的维数不断增多,约束条件更加复杂时,知识的统计、提取和储存方法等均是急需解决的问题.

References

- 1 Robert R G. An introduction to cultural algorithms. In: Proceedings of the 3rd Annual Conference Evolution Programming, Singapore: World Scientific Publishing, 1994. 131~136
- 2 Renfrew A C. *Dynamic Modeling in Archaeology: What, When, and Where? Dynamical Modeling and the Study of*

- Chang in Archaeology*. Edinburgh Scotland: Edinburgh University Press, 1994
- 3 Trung T N, Xin Y. Hybridizing cultural algorithms and local search. *Lecture Notes in Computer Science*. Springer, 2006, **4224**: 586~594
 - 4 Reynolds R G, Peng B. Knowledge learning and social swarms in culture algorithms. *The Journal of Mathematic Sociology*, 2005, **29**(2): 115~132
 - 5 Ricardo L B, Carlos A, Coello C. A cultural algorithm with differential evolution to solve constrained optimization problems. *Lecture Notes in Computer Science*. Springer, 2004, **3315**: 881~890
 - 6 Gao F, Cui G, Liu H W. Integration of genetic algorithm and cultural algorithms for constrained optimization. *Lecture Notes in Computer Science*. Springer, 2006, **4234**: 817~825
 - 7 Yuan X H, Yuan Y B. Application of cultural algorithm to generation scheduling of hydrothermal systems. *Energy Conversion and Management*, 2006, **47**: 2192~2201
 - 8 Reynolds R G, Saleem S. Culture algorithms in dynamic environments. In: Proceedings of Congress on Evolutionary Computation. San Diego, California, 2000. **2**: 1513~1520
 - 9 Coello C A, Becerra R I. Evolutionary multiobjective optimization using a cultural algorithm. In: Proceedings of 2003 IEEE Swarm Intelligence Symposium. Indianapolis, Indiana, IEEE Service Center, 2003. 6~13
 - 10 Reynolds R G, Zhu S. Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2001, **31**(1): 1~18
 - 11 Chung C J, Robert G R. A testbed for solving optimization problems using cultural algorithms. In: Proceedings of the Fifth Annual Conference on Evolutionary Programming. Cambridge, Massachusetts: MIT Press, 1996
 - 12 Jin X, Reynolds R G. Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. In: Proceedings of 1999 Congress on Evolutionary Computation. Washington: IEEE, 1999. 1672~1678
 - 13 Saleem S M. Knowledge-based Solution to Dynamic Optimization Problems Using Cultural Algorithms [Ph. D. dissertation], Wayne State University, 2001
 - 14 Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 1996, **4**(1): 1~2
 - 15 Slawomir K, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation*, 1999, **7**(1): 19~44
 - 16 Adjiman C S, Dallwig S, Floudas C A, Neumaier A. A global optimization method, alpha BB, for general twice-differentiable constrained NLPs-I. *Theoretical Advances*, 1998, **22**(9): 1137~1158
 - 17 He Yi-Jun, Chen De-Zhao. Constrained ant colony system and its application in process optimization of butene alkylation. *Journal of Chemical Industry and Engineering*, 2005, **56**(9): 1708~1713
(贺益君, 陈德钊. 连续约束蚁群优化算法的构建及其在丁烯烷化过程中的应用. 化工学报, 2005, **56**(9): 1708~1713)
- 黄海燕 讲师, 博士研究生. 主要研究方向为控制理论与控制工程.
E-mail: huanghong@ecust.edu.cn
(HUANG Hai-Yan Ph. D. candidate, lecturer. Her research interest covers control theory and control engineering.)
- 顾幸生 教授. 主要研究方向为控制理论与控制工程. 本文通信作者.
E-mail: xsgu@ecust.edu.cn
(GU Xin-Sheng Professor. His research interest covers control theory and control engineering. Corresponding author of this paper.)
- 刘漫丹 副教授. 主要研究方向为智能控制与模式识别.
E-mail: liumandan@ecust.edu.cn
(LIU Man-Dan Associate professor. Her research interest covers intelligent control and pattern recognition.)