第35卷 第23期 计 算 机 工 程 2009年12月
Vol.35 No.23 Computer Engineering December 2009
·博士论文· 文章编号：1000—3428(2009)23—0024—03 文献标识码：A 中图分类号：TP391

解0-1背包问题的混合编码贪婪DE算法

邓长寿[1,2]，梁昌勇[1]
(1. 合肥工业大学网络系统研究所，合肥 230009；2. 九江学院信息科学与技术学院，九江 332005)

摘　要：提出一种混合编码差异演化算法来求解0-1背包问题。通过增加边界约束处理算子和编码映射函数，构建混合编码差异演化算法，求解离散优化问题，并利用贪婪变换方法对演化过程中的不可行解进行修复。仿真实验结果表明了该算法求解0-1背包问题的有效性与适用性。

关键词：0-1背包问题；边界约束处理算子；混合编码贪婪差异演化

Mixed Coding Greedy Differential Evolution Algorithm for 0-1 Knapsack Problem

DENG Chang-shou[1,2], LIANG Chang-yong[1]
(1. Institue of Network System, Hefei University of Technology, Hefei 230009;
2. School of Information Science and Technology, Jiujiang University, Jiujiang 332005)

【Abstract】Hybrid Coding Greedy Differential Evolution(HCGDE) algorithm is proposed for 0-1 knapsack problem. A new operator, boundary-constraint handling operator, and a coding mapping function are embedded into the original Differential Evolution(DE) to construct a hybrid coding DE algorithm, which expands the continuous domain of DE to the discrete domain. During the evolution process, it uses the greedy transform algorithm to fix the infeasible solutions. Results of the numerical experiment show it is effective and useful in solving 0-1 knapsack problem.

【Key words】0-1 knapsack problem; boundary constraint handling operator; Hybrid Coding Greedy Differential Evolution(HCGDE)

1　概述

0-1背包问题(0-1 knapsack problem)是一个典型的组合优化问题[1]，有着广泛的应用背景，如货物装载、选址问题、材料切割等，而且常作为其他问题的子问题加以研究。背包问题有多种形式，其中，一般的整数背包问题可在有界的前提条件下转化为等价的0-1背包问题。0-1背包问题是指给定n种物品和一个背包，w_i与Val_i $(i=1,2,\cdots,n)$分别为物品i的重量和价值，C为背包总重量限制。如何选择装入背包的物品，使得装入背包中的物品总价值最大。在选择装入背包的物品时，对物品i只有2种选择，即装入或不装入背包，不能将物品i装入背包多次，也不能只装入部分物品i。0-1背包问题可归结为如下的约束优化问题：

$$\begin{aligned}&\text{Maximize}\quad f=\sum_{i=1}^{n}x_i\times Val_i\\&\text{s.t.}\quad \sum_{i=1}^{n}w_i\times x_i\leqslant C\\&\qquad x_i\in\{0,1\}\quad(i=1,2,\cdots,n)\end{aligned}\tag{1}$$

0-1背包问题是属于带约束条件的离散型NP-完全问题，常见的求解方法有精确方法和进化算法。虽然精确求解方法可以得到准确解，但是其时间复杂性与物品数目呈指数关系，因而时间复杂度高。进化算法尽管不一定得到准确解，但可以得到比较有效解，而且时间复杂度比较低，在求解背包问题中得到了广泛使用。例如文献[2]利用一种新的蚁群优化算法求解0-1背包问题，多次运行求解后可以得到最优结果，然而蚁群优化算法求解速度慢且运行结果不稳定。文献[3]利用改进的微粒群优化算法求解背包问题，虽然收敛速度快，但其最好的实验结果成功率为44%。文献[4]利用混合差异演化算法求解背包问题，虽然稳定性好且收敛速度较快，但求解精度仍有待提高。

差异演化(Differential Evolution, DE)是一种实数编码的基于种群演化的全局优化算法[5]。DE原理简单，控制参数少，实施随机、并行、直接的全局搜索，易于理解和实现。DE算法在连续问题的优化中获得了广泛的应用。由于DE算法中的遗传算子针对实数域设计，求解离散域上的优化问题时，会出现计算不封闭问题。虽然DE算法已成功应用于连续域上的最优化问题，但对于离散域上的最优化问题，特别是在NP-完全问题中的应用研究比较少见。本文提出一种混合编码贪婪差异演化(Hybrid Coding Greedy Differential Evolution, HCGDE)算法来求解0-1背包问题。

2　差异演化算法

DE算法在问题的可行解空间随机产生初始种群，然后对当前种群进行变异和交叉操作，产生一个过渡种群；利用基于贪婪的思想，对当前种群和过渡种群中的对应个体进行一对一的选择，产生新一代群体。DE算法根据变异和交叉操作的不同，形成了不同的模式，常见的有DE/rand/1/bin和DE/best/1/bin[5]。

基金项目：国家自然科学基金资助项目(70771037)；江西省教育厅科学技术基金资助项目(GG09347)

作者简介：邓长寿(1972－)，男，副教授、博士，主研方向：计算智能，智能决策；梁昌勇，教授、博士、博士生导师

收稿日期：2009-05-16 **E-mail：**csdeng@jju.edu.cn

DE/rand/1/bin 的详细描述如下：设 DE 算法种群规模为 NP，每个个体是 n 维向量，则第 G 代中的个体可表示为 $X_{i,G}, i=1,2,\cdots,NP$。DE 算法的主要操作包括产生初始种群、变异算子、交叉算子和选择算子。

产生初始种群比较简单，即在 n 维空间中，随机产生均匀分布的满足上下界约束的 NP 个个体的种群 $X_{i,0}, i=1,2,\cdots,NP$，如式(2)所示：

$$X_{ij,0}=X_j^l+rand(0,1)(X_j^u-X_j^l) \tag{2}$$

其中，$i=1,2,\cdots,NP$；$j=1,2,\cdots,n$；X_j^l 和 X_j^u 分别表示第 j 个变量的上界和下界。

DE 算法的变异算子利用随机 2 个个体间的偏差扰动产生新个体。变异算子得到中间个体记为 $v_{i,G+1}$，即

$$v_{i,G+1}=X_{r1,G}+F\times(X_{r2,G}-X_{r3,G}) \tag{3}$$

交叉算子将当前所得到的中间个体 $v_{i,G+1}$ 与个体 $X_{i,G}$ 进行杂交，如式(4)所示。交叉操作得到当前个体的候选个体 $u_{i,G+1}=(u_{1i,G+1},u_{2i,G+1},\cdots,u_{Di,G+1})$。

$$u_{ij,G+1}=\begin{cases} v_{ij,G+1} & \text{if} \quad (rand<CR) \quad \text{or} \quad j=R(i) \\ X_{ij,G} & \text{else} \end{cases} \tag{4}$$

其中，$i=1,2,\cdots,NP$; $j=1,2,\cdots,D$; $rand\in[0,1]$ 是一个均匀分布的随机数；$R(i)\in[1,D]$ 之间的随机整数；$CR\in[0,1]$ 为交叉概率。采用这种交叉策略，可以确保下一代个体中至少有一个染色体来源于中间个体。

选择算子根据候选个体的函数适应值，按照式(5)选择相应的个体进入下一代种群。

$$X_{i,G+1}=\begin{cases} u_{i,G+1} & \text{if } f(u_{i,G+1})\leqslant f(X_{i,G}) \\ X_{i,G} & \text{else} \end{cases} \tag{5}$$

DE 算法的控制参数有 3 个 NP, F 和 CR。种群规模 NP 取值一般为变量维数的 2 倍~20 倍，且至少应大于 4。$F\in[0,2]$ 的实数，$CR\in[0,1]$ 的实数。

3 解 0-1 背包问题的混合编码贪婪差异演化算法

DE 演化算法在连续优化领域取得了成功的应用，然而由式(1)所示的数学模型可知，0-1 背包问题是带约束的离散优化问题。由式(3)可知，在整个实数域该算子的计算是封闭的，对于变量的取值是离散型的或是在一定区间范围内的实数，计算结果往往是不封闭的。为了求解 0-1 背包问题，本文在 DE 算法中增加违反边界约束处理算子，提出一种新的区间编码映射，确保变异算子的计算封闭性，使得基于浮点数编码的 DE 算法可以有效地求解属于离散优化的 0-1 背包问题；在演化求解过程中，利用贪心变化法对违反约束条件的个体进行修正。本文将包含贪心变换方法处理约束条件的混合编码差异演化算法称为混合编码贪婪差异演化算法。

3.1 编码映射

由于基本 DE 算法适合处理连续变量优化问题，求解 0-1 背包问题时，需要将连续变量映射转化为离散变量。HCGDE 算法中，选择初始种群的取值在区间[0, 1]。为了便于 HCGDE 算法处理离散约束优化问题，将区间[0, 1]分成 2 个子区间[0, 0.5]和[0.5, 1]。当连续变量的取值属于区间[0, 0.5]时，其映射值为 0；当连续变量的取值属于区间[0.5, 1]时，其映射值为 1。映射函数 $f(x)$ 定义如下：

$$f(x)=\begin{cases} 0 & x\in[0,0.5) \\ 1 & x\in[0.5,1] \end{cases} \tag{6}$$

3.2 违反边界约束的处理算子

在 DE 算法中，通过变异算子与交叉算子所产生的新个体中的某分量可能超出了区间[0, 1]。在 HCGDE 算法中，采用重新生成新的分量方法解决违反边界约束条件问题。即利用式(7)所定义边界约束处理操作，生成一个新的分量来代替因变异和交叉操作所产生的不良分量。

$$u_{ij}(t+1)=\begin{cases} u_{ij}(t+1) & \text{if } u_{ij}(t+1)\in[0,1] \\ rand(0,1) & \text{else} \end{cases} \tag{7}$$

3.3 混合编码

在 HCGDE 算法中，每个个体的编码由维数相同的 2 个子向量构成，第 1 子向量为实数向量 $\boldsymbol{X}$，第 2 个子向量为二进制向量 $\boldsymbol{B}$。其中，子向量 $\boldsymbol{X}$ 为 DE 算法中的向量，二进制向量 $\boldsymbol{B}$ 由实数向量 $\boldsymbol{X}$ 利用式(6)所表示的函数映射而得到的离散向量。个体混合编码向量 $\boldsymbol{Y}$ 如下：

$$\boldsymbol{Y}=(\boldsymbol{X},\boldsymbol{B}) \tag{8}$$

其中，$\boldsymbol{B}=f(\boldsymbol{X})$。本文将由实数域的编码 $\boldsymbol{X}$ 和由离散域的编码 $\boldsymbol{B}$ 所共同构成的混合向量编码 $\boldsymbol{Y}$ 称为混合编码。

采用上述混合编码的 HCGDE 算法，不仅保留了 DE 算法的优点，而且可以求解离散域的优化问题。在变异和交叉运算时，利用实数域编码子向量 $\boldsymbol{X}$，在选择操作和计算目标函数时，利用离散域编码子向量 $\boldsymbol{B}$。因此，混合编码的 DE 算法能够有效解决离散优化计算中的不封闭问题。

3.4 贪心变化算法GTA

在演化求解 0-1 背包问题的过程中，可能出现违反总重量 C 的个体。为了保证顺利求解问题，在迭代的每一过程中，利用贪心变换算法将违反约束条件的个体转换为满足重量约束条件的个体。设个体 Y 为不满足式(1)中的重量约束的一个解，将所有物品按照价值密度 Val_i/w_i 降序排列得到一个队列 $q[i](i=1,2,\cdots,n)$。贪心变换算法如下。

算法 $GTA(\boldsymbol{Y}, C, q)$

输入 违反重量约束的个体 $\boldsymbol{Y}$，总重量 C 和队列 q

输出 修正后的个体 $\boldsymbol{Y}$

```
Set k=1, Temp=wq(1)
While (Temp<=C) do
Y.Bq(1)=1, k=k+1, Temp= Temp+wq(k)
End while
For j=k to n do Y.Bq(j)=0;
For i=1 to n do
if (Y.B(i)=0) Y.X(i)=rand*0.5
else Y.X(i)=0.5+rand*0.5
endif
Return(Y)
```

3.5 HCGDE算法描述

求解 0-1 背包问题的 HCGDE 算法流程如下：

(1)初始化。包括设置最大迭代代数、种群中个体数目、变异概率和交叉概率，并生成混合编码的初始种群。

(2)不满足结束条件时，执行步骤(3)~步骤(8)，否则转步骤(9)。

(3)执行变异算子。

(4)执行交叉算子。

(5)对于违反重量约束的个体，利用贪心变化算法进行修正。

(6)按照式(8)定义的区间编码映射，对种群中的向量重新编码。

(7)按照式(5)定义的选择算子，确定下一代个体。

(8)计算并保留最优值。

(9)输出最优值，算法结束。

4 仿真实例

用一个有 50 件物品的背包问题进验证本文算法的性能，改进的微粒群算法(PSO)[3]和混合差异演化算法[4]的最优解均为 3 103。

实例 物品种类 $n=50$, $C=1000$,

$w=\{80,82,85,70,72,70,66,50,55,25,50,55,40,48,50,32,$
$22,60,30,32,40,38,35,32,25,28,30,22,25,30,45,30,$
$60,50,20,65,20,25,30,10,20,25,15,10,10,10,4,4,2,1\}$

$Val=\{220,208,198,192,180,180,165,162,160,158,155,$
$130,125,122,120,118,115,110,105,101,100,100,$
$98,96,95,90,88,82,80,77,75,73,72,70,69,66,65,63,$
$60,58,56,50,30,20,15,10,8,5,3,1\}$

实验时种群数目为 250，变异参数为 0.5，交叉概率为 0.5。实验时各种最大迭代次数分别运行 50 次，统计最差解、最好解、平均解以及找到(或超过)已知最优解 3 103 的次数。算法运行结果及与改进的微粒群算法的对比如表 1 所示，其中改进的微粒群算法中种群数目为 500，其求解结果引自文献[4]。

表 1 HCGDE 算法与改进的 PSO 算法求解结果对比

最大迭代次数	算法	最差解	最好解	平均解	达到(或超过) 3 103 的次数
100	HCGDE	3 020	3 083	3 050.50	0
100	改进的 PSO	3 009	3 075	3 057.00	0
200	HCGDE	3 071	3 111	3 092.00	6
200	改进的 PSO	3 057	3 095	3 075.90	0
300	HCGDE	3 095	3 115	3 105.90	35
300	改进的 PSO	3 081	3 103	3 091.15	3
400	HCGDE	3 106	3 119	3 113.10	50
400	改进的 PSO	3 086	3 103	3 094.35	5
500	HCGDE	3 111	3 119	3 116.60	50
500	改进的 PSO	3 090	3 103	3 094.35	5
600	HCGDE	3 114	3 119	3 118.10	50
600	改进的 PSO	3 086	3 103	3 095.75	9
700	HCGDE	3 118	3 119	3 118.90	50
700	改进的 PSO	3 089	3 103	3 098.75	20
800	HCGDE	3 118	3 119	3 119.00	50
800	改进的 PSO	3 093	3 103	3 099.60	20
900	HCGDE	3 119	3 119	3 119.00	50
900	改进的 PSO	3 096	3 103	3 099.50	22
1 000	HCGDE	3 119	3 119	3 119.00	50
1 000	改进的 PSO	3 091	3 103	3 098.75	22

表 1 的实验结果表明，本文的 HCGDE 算法求解的最优解为 3 119。此外，HCGDE 算法也找到了多个优于已知最优解 3 103 的多个次优解。在最大迭代次数为 900 时，每次都能找到新的最优解 3 119。

从表 1 中 HCGDE 算法与改进的 PSO 算法的求解结果对比可以看出，HCGDE 算法在求解精度、收敛速度与鲁棒性等 3 个方面均优于改进的 PSO 算法。表 2 给出了 HCGDE 算法找到的最优解和几个次优解。

表 2 HCGDE 算法找到的最优解和次优解

解	*x*	*f*/*C*
最优解	11010101111010011011011111111110000101101100000000010	3 119/1 000
次优解	10011101111011011011001111111110000101101100000000010	3 113/1 000
次优解	10010101111011011011011111111110100101101100000001010	3 114/1 000
次优解	11010101111010011011011111111110000101101100000000001	3 117/999

5 结束语

DE 算法在无约束连续优化问题领域中得到了广泛的应用。本文在 DE 算法中增加违反边界约束处理算子和编码映射方法，构建混合编码 DE 算法。0-1 背包问题实例仿真实验结果表明，HCGDE 算法求解出比文献[3-4]的最优解更好的多个次优解和最新的最优解。HCGDE 算法求解精度高、算法鲁棒性高、收敛速度快，是求解背包问题的有效方法，也可用于求解其他的离散优化问题。

参考文献

[1] SysioM M. Discrete Optimization Algorithms[M]. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.

[2] 秦 玲, 白 云, 章春芳. 解 0-1 背包问题的蚁群算法[J]. 计算机工程, 2006, 32(6): 212-214.

[3] 沈显君, 王伟武, 郑波尽. 基于改进的微粒群优化算法的 0-1 背包问题求解[J]. 计算机工程, 2006, 32(18): 23-24.

[4] 郭广寒, 王志刚, 郝志峰, 等. 混合差异演化算法在背包问题中的应用[J]. 计算机工程与应用, 2008, 44(8): 89-91.

[5] Storn R, Price K. Differential Evolution——A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

编辑 顾姣健

~~~~~~

（上接第 23 页）

参考文献

[1] Web Coverage Service(WCS) Implementation Standard[EB/OL]. [2008-3-19]. http://www.opengeospatial.org/legal/.

[2] 王 晗, 孔令富, 练秋生. 基于图像映射的关联规则数据挖掘方法[J]. 计算机工程, 2008, 34(21): 71-72.

[3] Wang Yunsong, Bollig E F, Benjamin J, et al. Web-IS(Integrated System): An Overall View[J]. Visual Geosciences, 2005, (10): 27-42.

[4] 梅 彪, 姜新文, 吴 恒. WS-BPEL 业务流程与访问控制[J]. 计算机工程, 2008, 34(19): 144-146.

[5] Sceppa D. Programming Microsoft ADO.Net 2.0 Core Reference[M]. 北京: 清华大学出版社, 2007-06.

编辑 顾姣健
~~~~~~