第 11 卷　第 34 期　2011 年 12 月
1671—1815(2011)34-8405-04

科　学　技　术　与　工　程
Science Technology and Engineering

Vol. 11　No. 34　Dec. 2011
© 2011　Sci. Tech. Engrg.

差异演化算法求解多选择背包问题

王　研　王志刚*
(南京师范大学泰州学院数学科学与应用学院,泰州 225300)

摘　要　多选择背包问题是典型的 NP 难题。建立了多选择背包问题的数学模型。设计了差异演化算法对其进行求解。通过对其它文献中实例的仿真试验和结果对比,表明了算法求解多选择背包问题的可行性和有效性。

关键词　差异演化算法　　多选择背包问题　　优化
中图法分类号　O232　TP183;　　　　**文献标志码**　A

0-1 背包问题属于经典的 NP-hard 问题。对该问题的研究有较大的理论意义和重要的应用价值。随着该问题的发展,产生了许多该问题的变形,例如多选择背包问题、多约束背包问题、有界背包问题、无界背包问题。其中多选择背包问题涉及的约束条件种类最多,在各种变形中最为复杂,目前研究的文献相对较少[1—3]。

差异演化算法(Differential Evolution, DE)是 Rainer Storn 和 Kenneth Prince 提出的一种基于群体差异的演化算法[4—6],属于直接搜索算法。差异演化算法具有收敛速度快、操作简单、易编程实现、极强的稳健性等优点,在首届 IEEE 演化计算大赛中,差异演化算法表现超群,随后在各领域得到了广泛的应用。但这些应用主要集中在连续优化领域[7,8],在离散优化领域上的应用相对较少。本文利用差异演化算法的优点,设计了一种求解多选择背包问题的差异演化算法。仿真实验表明本文提出的算法在多选择背包问题上是可行的和有效的。

1　数学描述

多选择背包问题定义为有附加约束的背包问题,该问题带有互不相关的多选择约束。多选择背包问题的一般性描述如下:有一个载重有限的背包和一批具有装载费用和重量的物品,先将要放入背包的物品分为互相排斥的若干类,每一类中有若干个不同的物品。问题就是从每类中选择且必须选择一个物品放进背包,使得在物品总重量不超过背包承重的前提下,总费用最小化。

问题的数学描述如下:

$$\min f(X) = \sum_{i=1}^{m}\sum_{j=1}^{n_i} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{m}\sum_{j=1}^{n_i} w_{ij}x_{ij} \leqslant W;$$

$$\sum_{j=1}^{n_i} x_{ij} = 1, i = 1,2,\cdots,m;$$

$$x_{ij} \in \{0,1\}, \forall i,j \tag{2}$$

式(2)中 i 是不同分类的下标,j 是每一类中不同物品的下标,n_i 是第 i 类中物品的数量,c_{ij}是第 i 类中第 j 个物品的装载费用,m 是不同分类的数量,w_{ij}是第 i 类中第 j 个物品的重量,W 是背包的载重,当 $x_{ij}=1$ 时,表示第 i 类中的第 j 个物品放入背包,当 $x_{ij}=0$ 时,表示第 i 类中的第 j 个物品不放入背包。

2　差异演化算法

差异演化算法是基于实数编码的进化演化算法,它的整体结构类似于遗传算法, 与遗传算法的主要区别在变异操作上,差异演化算法的变异操作是基于染色体的差异向量进行的,其余操作和遗传算法类似。下面通过求解非线性函数 $f(x_1,x_2,\cdots,$

2011 年 9 月 13 日收到
* 通信作者简介:王志刚。E-mail:wzg19. scut@163. com。

x_n)的最小值问题,x_j 满足 $x_j^L \leqslant x_j \leqslant x_j^U$,$j=1,2,\cdots,n$ 来介绍差异演化算法的操作过程:

令 $x_i(t)$ 是第 t 代的第 i 个染色体则 $x_i(t)=(x_{i1}(t),x_{i2}(t),\cdots,x_{in}(t))$,$i=1,2,\cdots,M$;

$t=1,2,\cdots,t_{\max}$。其中,n 是染色体的长度,即变量的个数,M 为群体规模,$t_{\max}$是最大的进化代数。

2.1 生成初始种群

在 n 维空间里随机产生满足约束条件的 M 个染色体,实施如下措施:

$$x_{ij}(0)=\mathrm{rand}_{ij}(0,1)(x_{ij}^U-x_{ij}^L)+x_{ij}^L;\ i=1,2,\cdots,M; j=1,2,\cdots,n \tag{3}$$

式(3)中,x_{ij}^U,x_{ij}^L 分别是第 j 个变量上界和下界,$\mathrm{rand}_{ij}(0,1)$是[0, 1]之间的随机小数。

2.2 变异操作

从群体中随机选择 3 个染色体 x_{p1},x_{p2},x_{p3} 且($i \neq p1 \neq p2 \neq p3$),则

$$h_{ij}(t+1)=x_{p1j}(t)+\lambda(x_{p2j}(t)-x_{p3j}(t)) \tag{4}$$

$x_{p2j}(t+1)-x_{p3j}(t+1)$ 为差异化向量,λ 为缩放因子。

2.3 交叉操作

交叉操作是为了增加群体的多样性,具体操作如下:

$$v_{ij}(t+1)=\begin{cases} h_{ij}(t+1), & \mathrm{rand}_{ij}(0,1)\leqslant pc \text{ 或 } j=\mathrm{rand}(i) \\ x_{ij}(t), & \mathrm{rand}_{ij}(0,1)>pc \text{ 且 } j\neq \mathrm{rand}(i) \end{cases} \tag{5}$$

$\mathrm{rand}_{ij}(0,1)$是在[0,1]之间的随机小数,pc 为交叉概率,$pc \in [0,\ 1]$, $\mathrm{rand}(i)$为在[1,n]之间的随机整数,这种交叉策略可确保 $x_i(t+1)$至少有一分量由 $h_i(t+1)$的相应分量贡献。

2.4 选择操作

为了决定 $x_i(t)$是否成为下一代的成员,向量 $v_i(t+1)$和目标向量 $x_i(t)$对评价函数进行比较:

$$x_i(t+1)=\begin{cases} v_i(t+1), & f(v_{i1}(t+1),\cdots,v_{in}(t+1)) < f(x_{i1}(t+1),\cdots,x_{in}(t+1)) \\ x_i(t), & f(v_{i1}(t+1),\cdots,v_{in}(t+1)) \geqslant f(x_{i1}(t+1),\cdots,x_{in}(t+1)) \end{cases} \tag{6}$$

反复执行 2.2 至 2.4 操作,直至达到最大的进化代数 $t_{\max}$。

3 求解多选择背包问题的差异演化算法

3.1 个体编码方法

本文采用正整数编码方法表示个体,在正整数编码方法中,种群中每一个个体由 m 个基因组成,分别对应着 m 个不同的分类,其中第 i 个基因是从相互排斥集合 N_i 中选取的一个正整数,N_i 为第 i 类中所包含物品的编号构成的集合($\{1,2,\cdots,n_i\}$),这样,基因的位置用来表示物品所在类的编号,而基因值则用来表示物品在该类中的编号。对于个体 $X=(t_1,t_2,\cdots,t_m)^{\mathrm{T}}$,令 t_i 是个体中第 i 个基因,当 $t_i=j$ 时,对应着(1)、(2)式中的 $x_{ij}=1$,$j\in N_i$,$i=1,2,\cdots,m$,表示第 i 类中的第 j 个物品装入了背包。

3.2 求解多选择背包问题的差异演化算法

差异演化算法虽然成功应用于连续域上的数值优化问题,但对于离散域上组合优化问题却不能直接应用,主要是由于变异操作的计算公式(4)是基于实数域上的四则运算,而离散域上组合优化问题的解通常由自然数表示,公式(4)对自然数不满足封闭性。如何在保留变异操作高效计算的基础上使自然数的四则运算封闭是把差异演化算法用于求解组合优化问题的关键。本文通过借鉴文献[9]的思想,提出了一种新颖的差异演化算法,使其可以对采用上述编码方式的多选择背包问题进行求解。新算法中把变异操作和交叉操作合在一起,方式如下:

$$v_{ij}(t+1)=\begin{cases} x_{ij}(t) & \mathrm{rand}_{ij}(0,1)\leqslant pc_1 \text{ 或 } j=\mathrm{rand}(i) \\ x_{d_1j}(t) & pc_1<\mathrm{rand}_{ij}(0,1)\leqslant \sum_{i=1}^{2} pc_i \\ & \text{且 } j\neq \mathrm{rand}(i) \\ x_{d_2j}(t) & \sum_{i=1}^{2} pc_i<\mathrm{rand}_{ij}(0,1)\leqslant \sum_{i=1}^{3} pc_i \\ & \text{且 } j\neq \mathrm{rand}(i) \\ x_{d_3j}(t) & \sum_{i=1}^{3} pc_i<\mathrm{rand}_{ij}(0,1)\leqslant \sum_{i=1}^{4} pc_i \\ & \text{且 } j\neq \mathrm{rand}(i) \end{cases} \tag{7}$$

式(7)中 $pc_1+pc_2+pc_3+pc_4=1$。为防止当种群趋于一致时,执行上述的操作很难产生新的个体而导致算法停止进化,因此以一定的概率 pc 对执行完上述操作的第 i 个粒子($i=1,2,\cdots,N$)的分量进行重新初始化,即产生一个 1 到 n_i 之间的随机整数。

由于多选择背包问题是约束优化问题,因此,在求解过程中个体编码对应的解还必须满足不等式(2),为了解决这个问题,本文引入罚函数,其操作如下:当个体编码对应的解满足不等式(2)时,$f(X)=\sum_{i=1}^{m}\sum_{j=1}^{n_i}c_{ij}x_{ij}$;否则,$f(X)=\sum_{i=1}^{m}\sum_{j=1}^{n_i}c_{ij}x_{ij}+M(\sum_{i=1}^{m}\sum_{j=1}^{n_i}w_{ij}x_{ij}-W)$。其中 M 是大的正惩罚值。

4　仿真实验

为了说明本文提出的求解多选择背包问题的差分演化算法的可行性与有效性,对于文献[10]中 Gen 等人提出的一个含有 8 类物品的多选择背包问题典型实例进行仿真实验,文献[10]中提到,Gen 设置种群规模为 40,迭代次数为 100,杂交率为0.2,变异率为 0.1。进行 40 次运算,找到最优解 34 的概率为 67.5%。我们设置种群大小和迭代次数与 Gen 的相同,参数 $pc=0.15$,$pc_1=0.1$,$pc_2=0.3$,$pc_3=0.3$,$pc_4=0.3$。

$$\begin{aligned}\min f(X)=&3x_{11}+4x_{12}+5x_{13}+4x_{14}+8x_{15}+5x_{16}+4x_{17}+\\&4x_{18}+7x_{21}+9x_{22}+8x_{23}+4x_{24}+5x_{25}+5x_{31}+\\&6x_{32}+9x_{33}+8x_{34}+4x_{35}+6x_{36}+2x_{41}+8x_{42}+\\&7x_{43}+5x_{44}+4x_{45}+3x_{46}+7x_{47}+8x_{48}+8x_{51}+\\&5x_{52}+7x_{53}+9x_{54}+7x_{55}+4x_{56}+7x_{57}+6x_{61}+\\&8x_{62}+7x_{63}+9x_{64}+5x_{65}+6x_{71}+9x_{72}+7x_{73}+\\&4x_{74}+3x_{75}+7x_{76}+9x_{77}+8x_{78}+9x_{81}+2x_{82}+\\&5x_{83}+5x_{84}+8x_{85}+6x_{86}。\end{aligned}$$

$$\begin{aligned}\text{s.t.}\ 6\,&x_{11}+8x_{12}+5x_{13}+4x_{14}+9x_{15}+5x_{16}+9x_{17}+3x_{18}+\\&6x_{21}+5x_{22}+3x_{23}+5x_{24}+9x_{25}+8x_{31}+6x_{32}+4x_{33}+\\&8x_{34}+4x_{35}+3x_{36}+9x_{41}+4x_{42}+8x_{43}+9x_{44}+4x_{45}+\\&9x_{46}+4x_{47}+5x_{48}+3x_{51}+7x_{52}+4x_{53}+2x_{54}+3x_{55}+\\&8x_{56}+5x_{57}+8x_{61}+3x_{62}+2x_{63}+8x_{64}+6x_{65}+5x_{71}+\\&4x_{72}+3x_{73}+8x_{74}+9x_{75}+2x_{76}+4x_{77}+4x_{78}+2x_{81}+\\&9x_{82}+2x_{83}+6x_{84}+4x_{85}+3x_{86}\leqslant 40。\\&x_{11}+x_{12}+x_{13}+x_{14}+x_{15}+x_{16}+x_{17}+x_{18}=1,\\&x_{21}+x_{22}+x_{23}+x_{24}+x_{25}=1,\\&x_{31}+x_{32}+x_{33}+x_{34}+x_{35}+x_{36}=1,\\&x_{41}+x_{42}+x_{43}+x_{44}+x_{45}+x_{46}+x_{47}+x_{48}=1,\\&x_{51}+x_{52}+x_{53}+x_{54}+x_{55}+x_{56}+x_{57}=1,\\&x_{61}+x_{62}+x_{63}+x_{64}+x_{65}=1,\\&x_{71}+x_{72}+x_{73}+x_{74}+x_{75}+x_{76}+x_{77}+x_{78}=1,\\&x_{81}+x_{82}+x_{83}+x_{84}+x_{85}+x_{86}=1。\end{aligned}$$

利用本文提出的算法运行 100 次,找到最优解的概率为 99%,其中找到的最差解为 35,仅比最优解 34 差 1 个单位,这表明本文算法对于求解多选择背包问题,不仅是可行的,而且是高效的。

5　结论

针对多选择背包问题的特点,本文设计了一种差异演化算法对其进行求解。测试结果表明,本文算法求解多选择背包问题是高效的,虽然不能 100% 的找到最优解,但考虑到多选择背包问题约束条件的类型多而特殊,这已是比较好的结果了。

差异演化算法的研究虽然还处于初期,仍还有许多问题需要进一步的研究,但从当前的应用效果来看,差异演化算法无疑具有十分光明的前景,进一步拓展差异演化算法新的应用领域仍是一项非常有意义的工作。

参 考 文 献

1　叶宇风. 多重群体遗传算法在多选择背包问题中的应用. 计算机工程与设计, 2005;26(12):3442—3444

2　贺毅朝, 寇应展, 陈致明. 求解多选择背包问题的改进差分演化算法. 小型微型计算机系统, 2007;28(9):1682—1685

3　鲍江宏. 用遗传算法实现罚函数法解多选择背包问题. 计算机工程与设计, 2008;29(17):4518—4524

4　Storn R, Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012. Berkeley: Intern-ational Computer Science Institute,1995

5　Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global

Optimization, 1997;11:341—359

6 Price K. Differential evolution *vs*. the funct-ions of the 2 *nd* ICEO. Proceeding of 1997 IEEE International Conference on Evolutionary Computation, 1997

7 Pahner U, Hameyer K. A daptive coupling of differential evolution and multiquadricsapprox-imation for the tuning of the optimization process. IEEE Transactions Magnetics,2000;36(4):1047—1051

8 Cheng S L , Hwang C. Optimal approximation of linear systems by a differential evolution algorithm. IEEE Transactions Systems, Man and Cybernetics —Part A , 2001;31(6) : 698—707

9 谭 皓,王金岩,何亦征,等.一种基于子群杂交机制的粒子群算法求解旅行商问题.系统工程,2005;23(4):83—87

10 玄光男,程润伟,于歆杰,等.遗传算法与工程优化.北京:清华大学出版社, 2004: 56—59

Differential Evolution for Mutilple-choice Knapsack Problem

WANG Yan,WANG Zhi-gang*

(School of Mathematics, Nanjing Normal University Taizhou College, Taizhou 225300, P. R. China)

[**Abstract**] Multiple-choice knapsack problem is a typical NP problem. The model of multiple-choice knapsack problem was formulated and differential evolution was designed to solve it. Example in other references is recomputed and both simulation results are compared. It can be found that the possibility and the efficiency of the algorithm in solving multiple-choice knapsack problem.

[**Key words**] differential evolution multiple-choice knapsack problem optimization

(上接第 8404 页)

Mond-Weir Duality for Multiobjective Programming under the $(p,r)_{h,\varphi}$- Invex

GAO Ying,ZHANG Qing-xiang*

(College of Mathematics and Computer Science,Yan'an University, Yan'an 716000,P. R. China)

[**Abstract**] $(p,r)_{h,\varphi}$-invex and strictly $(p,r)_{h,\varphi}$-invex function were defined. So,the Mond-Weir duality of multiobjective programming are obtained by applying (p,r)-η Invex.

[**Key words**] $(p,r)_{h,\varphi}$-invex function efficientsolution multiobjective programming Mond-Weir duality