

一种改进的约束优化粒子群算法 *

吴华伟^{1a}, 陈特放^{1a}, 胡春凯^{1b}, 许 炳²

(1. 中南大学 a. 信息科学与工程学院; b. 航空航天学院, 长沙 410083; 2. 空军驻长沙地区军事代表室, 长沙 410205)

摘 要: 提出一种新的约束优化粒子群算法。该算法采用非固定多段映射罚函数法处理约束条件。在进化过程中, 利用混沌序列初始化种群, 选取最优粒子进行局部一维搜索, 增强了在最优点附近的局部搜索能力, 以加快算法的收敛速度; 引入维变异方法保持种群的多样性。数值实验结果表明了该算法的有效性。

关键词: 约束优化问题; 粒子群算法; 非固定多段映射罚函数; 维变异

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)03-0859-03

doi:10.3969/j.issn.1001-3695.2012.03.016

Improved constrained optimization particle swarm optimization algorithm

WU Hua-wei^{1a}, CHEN Te-fang^{1a}, HU Chun-kai^{1b}, XU Bing²

(1. a. School of Information Science & Engineering, b. School of Aeronautics & Astronautics, Central South University, Changsha 410083, China; 2. Military Representative Office of Air Force in Changsha, Changsha 410205, China)

Abstract: Using non-stationary multi-stage assignment penalty function method to deal with the constraint conditions, this paper proposed a novel constrained optimization particle swarm optimization algorithm. It used chaotic sequences in the initialization of the evolutionary population. In the process of population evolution, the proposed algorithm selected the best population individual for local search to speed up the convergence rate of the algorithm. It maintained the population diversity through dimension mutation method. Numerical experiment results show that it is an effective algorithm.

Key words: constrained optimization problem; particle swarm optimization algorithm; non-stationary multi-stage assignment penalty function; dimension mutation

0 引言

约束优化问题是广泛存在于工程应用和科学研究中的一类优化问题。一个非线性约束优化问题可描述为

$$\begin{aligned} \min f(\mathbf{x}) \quad & \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{R}^n \\ \text{s. t. } & g_j(\mathbf{x}) \leq 0 \quad j=1, 2, \dots, l \\ & h_j(\mathbf{x}) = 0 \quad j=l+1, l+2, \dots, p \\ & l_i \leq x_i \leq u_i \quad i=1, 2, \dots, n \end{aligned} \quad (1)$$

其中: $\mathbf{x} \in \Omega \subseteq S$ 为决策变量; Ω 为可行域; S 为决策空间。一般地, S 为 $\mathbf{R}^n$ 中的 n 维长方体。变量 x_i 的取值范围为 $[l_i, u_i]$ 。 $f(\mathbf{x})$ 为目标函数, $g_j(\mathbf{x}) \leq 0$ 为第 j 个不等式约束条件, $h_j(\mathbf{x}) = 0$ 为第 j 个等式约束条件。通常将等式约束转换为两个不等式约束来处理:

$$\begin{cases} h_j(\mathbf{x}) - \sigma \leq 0 \\ -h_j(\mathbf{x}) - \sigma \leq 0 \end{cases} \quad (2)$$

其中: σ 为容忍值, 一般取一个很小的正数。

由于约束条件通常都是非线性的, 这使得约束优化问题的求解要比无约束优化问题复杂、困难得多。传统的求解约束优化问题的方法如可行方向法、惩罚函数法、梯度投影法等, 都需要借助问题的梯度信息, 要求目标函数或约束条件连续可微, 或对初始点要求较高, 且求得的大多为局部最优解^[1]。与传

统的方法相比, 进化算法如遗传算法、粒子群优化算法等是一种基于种群迭代的随机全局优化方法, 其求解过程中不依赖于目标函数的解析性质, 无须借助问题的梯度信息, 能以较大概率收敛到全局最优解, 更适用于求解约束优化问题, 但它需要结合一种合适的约束处理技术。利用进化算法求解约束优化问题已成为进化计算领域的研究热点之一, 并涌现出大量的约束优化进化算法^[1~3]。

粒子群优化 (PSO) 算法是由 Kennedy 等人^[4]提出的一种模拟群体智能搜索的优化算法, 其具有参数设置少、容易实现、能有效地解决复杂优化问题的特点, 在函数优化、模式识别、组合优化以及一些工程领域得到了广泛的应用。本文采用非固定多段罚函数法^[5,6]处理约束条件, 将非线性约束优化问题转换为一系列无约束优化问题, 然后利用改进的粒子群算法进行优化。为保证种群个体均匀分布在搜索空间中, 利用混沌序列初始化种群。对种群中的最优粒子进行局部搜索, 以加快算法的收敛速度。几个典型的 Benchmark 问题的实验结果表明了该算法的有效性。

1 非固定多段罚函数法

在求解约束优化问题中, 罚函数法是最常用的约束处理技术之一, 其基本思想是在目标函数中加上一个能反映是否满足

收稿日期: 2011-08-08; 修回日期: 2011-09-15 基金项目: 国家“863”计划资助项目 (2009AA034302)

作者简介: 吴华伟 (1979-), 男, 湖北襄阳人, 博士研究生, 主要研究方向为智能控制、飞机着陆系统 (csuw@2008@163.com); 陈特放 (1957-), 湖南涟源人, 教授, 博导, 主要研究方向为机车故障诊断、智能交通系统; 胡春凯 (1965-), 男, 湖北黄梅人, 高级工程师, 主要研究方向为飞机着陆系统; 许炳 (1980-), 男, 湖南浏阳人, 工程师, 主要研究方向为航空自动控制。

约束的惩罚项,从而构成一个无约束的广义目标函数,然后利用优化算法对该广义目标函数进行寻优,使得算法在惩罚项的作用下找到问题的最优解^[5]。

对于如式(1)所示的约束优化问题,等式约束按式(2)进行转换。通常所构造的广义目标函数具有如下形式:

$$F(x) = f(x) + \delta(t)H(x) \quad (3)$$

其中: $f(x)$ 为原目标函数; $\delta(t)H(x)$ 为惩罚项; $\delta(t)$ 表示惩罚力度; $H(x)$ 为惩罚因子。

在求解约束优化问题的整个过程中,如果式(3)中的 $\delta(t)$ 固定不变,则称为固定罚函数法,否则称为非固定罚函数法。固定罚函数法也叫精确罚函数法,在整个求解过程中只需计算一个目标函数,但惩罚力度 δ 难以选择。如果 δ 太大,则会在可行域边界使目标函数产生病态,给计算带来困难;如果太小,则收敛减慢,甚至出现不收敛的情况^[5,7]。而非固定罚函数法则在求解问题的过程中, $\delta(t)$ 是随迭代次数 t 不断变化的,将约束优化问题转换为一系列无约束优化问题来求解。Paropoulos 等人^[6]提出了一种基于非固定多段映射罚函数法的约束优化粒子群算法。本文也采用非固定多段映射罚函数法处理约束条件,在文献[7]的基础上,对其进行适当修正。设

$$H(x) = \sum_{j=1}^{2m-p} \theta(q_j(x)) q_j(x)^{r(q_j(x))} \quad (4)$$

$$q_j(x) = \max\{0, p_j(x)\} \quad (5)$$

$$p_i(x) = \begin{cases} g_j(x) & j = 1, 2, \dots, p \\ h_{j-p}(x) & j = p+1, p+2, \dots, 2m-p \end{cases} \quad (6)$$

$$\theta(q_j(x)) = \begin{cases} 6 & q_j(x) < 0.001 \\ 12 & 0.001 \leq q_j(x) \leq 0.1 \\ 18 & 0.1 < q_j(x) \leq 1 \\ 36 & \text{其他} \end{cases} \quad (7)$$

$$\delta(t) = \sqrt{t} \quad (8)$$

其中,式(5)表示对约束的违反程度;式(6)表示惩罚函数的强度;式(7)为分段映射函数;式(8)为随迭代次数变化的惩罚力度。这样可根据约束违反程度的大小,自适应选取不同的惩罚力度,将约束优化问题转换为一系列无约束优化问题来处理,可以避免采用精确罚函数法中惩罚因子难以选取的问题^[5-7]。

2 改进的粒子群优化算法

2.1 混沌序列产生初始种群

粒子群算法一般是随机产生初始种群,这样可能会导致初始个体在解空间中分布不均匀,从而影响算法的性能。为了提高算法的全局搜索能力,初始种群个体应尽可能均匀分布在解空间中。本文利用混沌序列来初始化种群个体。混沌是一种非线性现象,具有遍历性和随机性等特性,可在一定范围内按自身规律不重复地遍历所有状态,因此,将其应用到优化算法中来提高算法的全局搜索能力。

本文将引入常用的一维 Logistic 映射混沌模型^[8,9]来进行种群的初始化,其表达式为

$$Z_i^{k+1} = \mu Z_i^k (1 - Z_i^k) \quad (9)$$

其中: $Z_i \in (0, 1)$ 表示混沌变量, $i(i = 1, 2, \dots, n)$ 为混沌变量的序号; $k(k = 1, 2, \dots, N)$ 表示种群序号, N 为种群规模; μ 为控制参数。当 $\mu = 4$ 时,式(9)完全处于混沌状态。 μ 的取值不同,

混沌序列的搜索范围也不同^[9]。

利用混沌对初值敏感的特点,给式(9)赋 n 个微小差异的初值,即可得到 n 个混沌变量 Z_i^k ,依次取 $k = 1, 2, \dots, N$,便可得到 N 个初始种群粒子。将 n 个混沌变量 Z_i^k 作逆映射得到相应优化空间中的变量 x_i^k :

$$x_i^k = l_i + (u_i - l_i) Z_i^k \quad (10)$$

式中: $i = 1, 2, \dots, n$; $[l_i, u_i]$ 为变量 x_i 的变化区间。

2.2 嵌入局部一维搜索的粒子群算法

在 PSO 算法中, $u_i = (u_{i1}, u_{i2}, \dots, u_{im})$ 表示粒子 i 的位置向量, $p_i = (p_{i1}, p_{i2}, \dots, p_{im})$ 表示第 i 个粒子所经历过的最好位置,每个粒子的速度向量表示为 $v_i = (v_{i1}, v_{i2}, \dots, v_{im})$,种群中所有粒子经历过的最好位置为 $p_g = (p_{g1}, p_{g2}, \dots, p_{gm})$,粒子 i 的速度、位置更新公式如下:

$$\begin{cases} v_{im}^{k+1} = wv_{im}^k + c_1 r_1 (p_{im} - u_{im}^k) + c_2 r_2 (p_{gm} - u_{im}^k) \\ u_{im}^{k+1} = u_{im}^k + v_{im}^{k+1} \end{cases} \quad (11)$$

式中: r_1 和 r_2 为 $[0, 1]$ 之间的随机数, w 为惯性权重。 w 是粒子群算法的一个关键参数,它可以平衡算法的全局搜索能力和局部搜索能力。在进化初期,希望粒子具有较好的探索能力,随着迭代次数的增加,在进化后期,希望粒子具有较好的开发能力。所以在进化过程中需要动态调整惯性权重。因此,本文采用惯性权重 w 随进化代数线性递减^[10]:

$$w = w_{\max} - (w_{\max} - w_{\min}) \frac{\text{iter}}{\text{iter}_{\max}} \quad (12)$$

其中:iter为当前迭代次数;iter_{max}为最大迭代次数; $w_{\max}$ 和 $w_{\min}$ 为惯性权重的最大值和最小值。

在粒子群算法中, c_1 和 c_2 分别为自我学习因子和社会学习因子,一般都取为2。很少有人研究学习因子 c_1 和 c_2 对粒子群算法性能的影响。在算法初期,要求粒子具有较大的自我学习能力和较小的社会学习能力,这样使粒子可以在整个搜索空间飞行;而在算法后期,则要求粒子具有较小的自我学习能力和较大的社会学习能力,使粒子飞向全局最优解^[11]。为了提高算法的收敛性能,本文采用动态调整自我学习因子 c_1 和社会学习因子 c_2 。具体调整策略^[11]为

$$\begin{cases} c_1 = (c_{1e} - c_{1s}) \frac{\text{iter}}{\text{iter}_{\max}} + c_{1s} \\ c_2 = (c_{2e} - c_{2s}) \frac{\text{iter}}{\text{iter}_{\max}} + c_{2s} \end{cases} \quad (13)$$

其中: c_{1s} 、 c_{1e} 、 c_{2s} 和 c_{2e} 分别为 c_1 和 c_2 的初始值和最终值。一般取 $c_{1s} = 2.5$, $c_{1e} = 0.5$, $c_{2s} = 0.5$, $c_{2e} = 2.5$ 时算法效果较好^[11]。

粒子群算法虽然具有较强的全局搜索能力,但其局部搜索能力较弱。由于其迭代过程的随机性,并不能完全保证迭代后的下一代粒子比上代粒子好,也就是说,粒子不一定沿着最优的方向进行迭代,从而影响了算法的性能^[12]。本文采用嵌入局部一维搜索来加强算法的局部搜索能力,其思路为:在每一次粒子群算法迭代以后,每个粒子都有其对应的适应度值,比较后找出最优粒子,对最优粒子直接进行一次以其梯度信息确定搜索方向的 Wolfe-Powell 一维搜索^[12],保证搜索以后得到的后代粒子的适应度比上一代更优,从而增强了算法的局部搜索能力,同时也加快了算法的收敛速度^[12]。

2.3 种群多样性保持

在算法迭代后期和收敛的情况下,由于所有粒子都向最优解的方向飞去,导致粒子不可避免地趋向同一化(多样性损

失),使得后期收敛速度明显变慢,同时算法收敛到一定精度时,无法继续优化,进而陷入局部最优解^[13]。因此,为了提高解的质量,当粒子群收敛到一定程度时就要进行变异,即以一定的几率让所有粒子在这一维上的值均匀分布,在以后的迭代过程中可能找到更好的解,增加跳出局部最优或得到更高精度解的机会^[13]。

很多参数都可以表示粒子群的收敛程度,它们的大小和变化情况都可以作为确定变异的依据。在此,选用式(14)来衡量种群的多样性^[14]:

$$\text{div}(P) = \frac{2}{|P| \times (|P| - 1) \times |L|} \times \sum_{i=1}^{|P|-1} \sum_{j=i+1}^{|P|} \sqrt{\sum_{k=1}^N (p_{ik} - p_{jk})^2} \quad (14)$$

其中: P 为种群; $|P|$ 为种群个体数目; n 为问题的维数; $|L|$ 为搜索空间 S 的最长半径; p_{ik} 和 p_{jk} 分别为种群中第 i 个和第 j 个个体的第 k 个基因。在对种群的多样性进行衡量后,在这里预先定义一个阈值 div (不同的问题其值不同),当种群多样性值 $\text{div}(P) < \text{div}$ 时,进行维变异,即随机对种群中的某一维重新初始化^[14]。

2.4 算法步骤

综上所述,本文提出的结合非固定多段映射罚函数的约束优化粒子群算法的步骤如下:

- a) 设置算法参数。初始化种群规模,一般为参数维数的10倍。在每个变量的定义域内,利用混沌序列产生初始种群。
- b) 按式(5)~(7)分别计算各粒子每个约束条件的惩罚因子。
- c) 按式(4)分别计算每个粒子的所有约束条件的惩罚因子 $H(x)$ 。
- d) 根据式(3)计算出每个粒子的适应度值,求出最优适应度值及最优个体。
- e) 判断惩罚因子 $H(x)$ 是否达到精度要求或是否达到最大迭代次数,若是则退出;否则执行步骤f)。
- f) 粒子按照式(11)~(13)进行迭代,产生新的粒子;对最优粒子进行局部一维搜索和进行维变异后产生新的粒子,合并所有粒子后返回步骤b)。

3 数值实验及分析

为了测试本文算法的性能,从文献[1]的13个标准测试问题中选取了6个优化问题,即g01、g02、g04、g06、g08和g11进行测试,各测试问题的具体表达式见文献[1]。将本文算法记为ICOPSO,并与CPSO^[15]、CHMP SO^[16]、PES O^[17]和RVPSO算法^[17]得到的结果进行比较。在比较实验中,ICOPSO算法的参数设置为:种群规模 N 取为每个问题参数变量个数的10倍, $w_{\max} = 0.9$, $w_{\min} = 0.2$,容忍值 $\sigma = 0.0001$,种群多样性值 div 根据不同问题具体设定。对所有的测试问题,适应值评价次数均设置为350 000次。其他算法的参数设置分别见其各自的文献。每个测试问题在相同条件下独立运行30次实验,记录其最好结果、平均结果和最差结果。

表1给出了在上述参数设置下,五种算法对6个标准测试问题的寻优结果比较。

从表1可知,本文所提出的ICOPSO算法对6个问题30次实验中一致地找到全局最优解,且具有较强的稳定性。CPSO算法除了问题g01和g08找到最优解,在其他4个问题上得到的解离理论最优解相差较远。与ICOPSO算法相比,CPSO算

法除问题g08取得相似的结果外,其余5个问题得到的解要明显劣于ICOPSO算法。CHMP SO算法在问题g01和g08上找到了最优解,其他4个问题得到的结果要明显劣于ICOPSO算法。除了高维多峰测试问题g02和含等式约束的g11外,PE-SO算法对其他4个测试问题也一致地找到全局最优解,问题g11得到的结果也接近理论最优解。与PESO算法相比,ICOP-SO算法在测试问题g02和g11上得到的结果要优,而在测试问题g01、g04、g06和g08上取得了相似的结果。VRPSO算法除了问题g02和g11没有找到最优解外,在其他4个问题上全部找到了最优解。基于以上比较和分析,ICOPSO算法要优于或相似于其他算法。

表 1 五种算法对 6 个问题的寻优结果比较

问题	最优值	状态	方法				
			CP SO ^[15]	CHMP SO ^[16]	PES O ^[17]	RVPSO ^[17]	ICOPSO
g01	-15.000	最好结果	-15.000	-15.000	-15	-15.000	-15.0000
		平均结果	-15.001	-15.000	-15	-14.4187	-15.0000
		最差结果	-134.219	-15.000	-15	-12.4531	-15.0000
g02	-0.803619	最好结果	-0.801388	-0.803432	-0.792608	-0.664028	-0.803619
		平均结果	-0.76530	-0.790406	-0.721749	-0.413257	-0.792163
		最差结果	-0.09170	-0.750393	-0.614135	-0.259980	-0.759012
g04	-30665.538	最好结果	-30665.659	-30665.500	-30665.538	-30665.538	-30665.538
		平均结果	-30665.656	-30665.500	-30665.538	-30665.538	-30665.538
		最差结果	-30665.626	-30665.500	-30665.538	-30665.538	-30665.538
g06	-6961.814	最好结果	-6961.825	-6961.810	-6961.814	-6961.814	-6961.814
		平均结果	-6875.940	-6961.810	-6961.814	-6961.814	-6961.814
		最差结果	-6482.755	-6952.810	-6961.814	-6961.814	-6961.814
g08	-0.095825	最好结果	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
		平均结果	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
		最差结果	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
g11	0.7499	最好结果	0.7490	0.7499	0.749	0.749	0.7499
		平均结果	0.7505	0.7501	0.749	0.749	0.7499
		最差结果	0.4466	0.7528	0.749	0.749	0.7499

4 结束语

本文采用非固定多段映射罚函数法对约束条件进行处理,提出一种改进的约束优化粒子群优化算法。根据约束条件的违背程度,自适应地选择不同的惩罚力度,将约束优化问题转换为一系列的无约束优化问题,然后利用改进的粒子群算法进行优化。嵌入局部一维搜索技术增加算法的局部搜索能力。几个标准测试问题的实验结果表明,该算法具有较强的通用性和稳定性。

参考文献:

[1] RUNARSSON T P, YAO Xin. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Trans on Evolutionary Computation, 2000, 4(3): 284-294.

[2] 王翔,董晓马,阎瑞霞,等.改进DE/EDA算法在求解难约束优化问题中的应用研究[J]. 计算机应用研究, 2010, 27(11): 4114-4117.

[3] MICHALEWICZ Z, SCHOENAUER M. Evolutionary algorithm for constraint parameter optimization problems[J]. Evolutionary Computation, 1996, 4(1): 1-32.

[4] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proc of International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.

[5] 吴亮红,王耀南,周少武,等.采用非固定多段映射罚函数的非线性约束优化差分进化算法[J]. 系统工程理论与实践, 2007, 27(3): 128-133.

[6] PARSOPOULOS K E, VRAHATIS M N. Particle swarm optimization method for constrained optimization problems[J]. Computer and Information Science, 2011, 181(1): 1153-1163. (下转第864页)

$$\begin{aligned}x_{21} + x_{22} + x_{23} + x_{24} + x_{25} &= 1 \\x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} &= 1 \\x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} &= 1 \\x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} &= 1 \\x_{61} + x_{62} + x_{63} + x_{64} + x_{65} &= 1 \\x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} + x_{77} + x_{78} &= 1 \\x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} &= 1\end{aligned}$$

由于多选择背包问题通常是以费用最小化为目标函数,即求 $f(x)_{\min}$ 的最小值。上式中,不等式约束表示背包的重量约束条件,等式约束表示每类物品中只能选一件。在计算求解中,种群数量 $S=40$,最大迭代次数 $G=100$, $\text{limit}=5$,跟随蜂个数 $\text{No}=5$,初始化种群 $\text{colony}(:,j)=\text{unidrnd}(\text{num}(j),S,1),1\leq j\leq 8$ 的整数。可求得该问题的最优值为34,相应的最优解为 $[1,4,5,5,6,3,5,3],[8,4,5,5,2,5,5,3],[8,4,5,5,6,5,4,3]$ 。

根据优化变量代表的含义,上述解可以写为
 $(x_{11},x_{24},x_{35},x_{45},x_{56},x_{63},x_{75},x_{83})$ 约束条件: $6+5+4+4+8+2+9+2=40\leq 40$
 $(x_{18},x_{24},x_{35},x_{45},x_{52},x_{65},x_{75},x_{83})$ 约束条件: $3+5+4+4+7+6+9+2=40\leq 40$
 $(x_{18},x_{24},x_{35},x_{45},x_{56},x_{65},x_{74},x_{83})$ 约束条件: $3+5+4+4+8+6+8+2=40\leq 40$

目标函数值随迭代次数的变化曲线如图3所示。

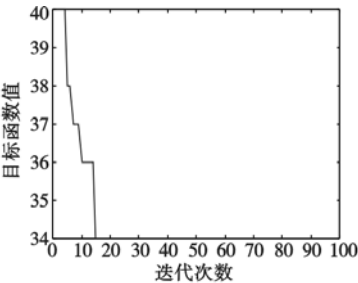


图3 目标函数值随迭代次数的变化曲线

以下是遗传算法、改进的差分演化算法(MDEA)^[3]与ABC算法就该问题的对比结果,如表1所示。
由表1可以看出,三种方法都可以求得该实例的最优解,但遗传算法的频率(求得最优解次数/运行总次数)比较低。在文献[3]中,改进的差分演化算法平均需9.3次迭代即可求得最优值34,即 $f(x)_{\min}=34$ 。由图3可知,人工蜂群算法需15次迭代即可求得最优值34,取得了比较满意的效果。经大量计算实验,算法体现了令人满意的性能,但由于这类算法思想出

现时间不长,理论基础尚未很好地奠定,渐近收敛性、鲁棒性等性质的严格数学证明有待于以后更深入地工作研究。

表1 算法比较

算法	最优解	最优值	频率
遗传算法	$[1,4,5,5,6,3,5,3]$	34	0.675
	$[8,4,5,5,2,5,5,3]$		
	$[8,4,5,5,6,5,4,3]$		
MEDA	$[1,4,5,5,6,3,5,3]$	34	1
	$[8,4,5,5,2,5,5,3]$		
	$[8,4,5,5,6,5,4,3]$		
ABC 算法	$[8,4,5,5,6,5,4,3]$	34	1
	$[8,4,5,5,2,5,5,3]$		
	$[1,4,5,5,6,3,5,3]$		

4 结束语

ABC算法作为一种新的智能优化算法,具有收敛快、鲁棒性强等特点。多选择背包问题属于组合优化问题,且是NP难题。本文就多选择背包问题的特点设计了基于MATLAB环境的人工蜂群算法,编写了算法程序。测试结果表明,ABC算法对MCKP求解是可行的,为有效求解多选择背包问题提供了一种新的可行方法,同时也拓展了人工蜂群算法的应用领域。

参考文献:

[1] 玄光男,程润伟,于歆杰,等. 遗传算法与工程优化[M]. 北京: 清华大学出版社, 2004: 56-59.
[2] GEN M, CHENG R, SASAKI M. Multiple-choice knapsack problem using genetic algorithms[C]//Advances in Engineering Design and Automation Research II. 1998: 1127-1132.
[3] 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[J]. 小型微型计算机系统, 2007, 28(9): 1682-1685.
[4] KARABOGA D. An idea based on honey bee swarm for numerical optimization, Technical Report TR06[R]. Turkey: Computer Engineering Department, Erciyes University, 2005.
[5] KARABOGA D, BASTURK B. On the performance of artificial bee colony(ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.
[6] 樊小毛,马良. 0-1 背包问题的蜂群优化算法[J]. 数学的实践与认识, 2010, 40(6): 155-160.
[7] SINGH A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. Applied Soft Computing, 2009, 9(2): 625-631.

(上接第 861 页)

[7] 刘俊梅,高岳林. 非线性混合整数规划问题的改进差分进化算法[J]. 工程数学学报, 2010, 27(6): 967-974.
[8] 任子武,伞冶. 实数遗传算法的改进及性能研究[J]. 电子学报, 2007, 35(2): 269-274.
[9] 梁晋明,龙文,龙祖强,等. 自适应梯度指导交叉的进化算法[J]. 小型微型计算机系统, 2011, 39(7): 1331-1335.
[10] SHI Y, EBERHART R C. Empirical study of particle swarm optimization[C]//Proc of Congress on Evolutionary Computation. Piscataway: IEEE Press, 1999: 1945-1950.
[11] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
[12] 龙文,梁晋明,董淑华,等. 嵌入局部一维搜索技术的混合粒子

群优化算法[J]. 计算机应用研究, 2009, 26(9): 3279-3281.
[13] 龙文,梁晋明,董淑华,等. 动态调整惯性权重的粒子群优化算法[J]. 计算机应用, 2009, 29(8): 2240-2242.
[14] LACEVIC B, KONJICIA S, AVDAGIC Z. Population diversity measure based on singular values of the distance matrix[C]//Proc of Congress on Evolutionary Computation. Singapore: IEEE Press, 2007: 1863-1869.
[15] CAGNINA L C, ESQUIVEL S C, COELLO C A. A particle swarm optimization for constrained numerical optimization[C]//Lecture Notes in Computer Science, vol 4193. 2006: 910-919.
[16] PULIDO G T, COELLO C A. A constraint-handling mechanism for particle swarm optimization[C]//Proc of Congress on Evolutionary Computation. [S.l.]: IEEE Press, 2004: 1396-1403.
[17] LU Hai-yan, CHEN Wei-qi. Dynamic-objective particle swarm optimization for constrained optimization problems[J]. Journal of Combinational Optimization, 2006, 12(4): 409-419.