

er)、策略 (strategy) 和收益 (payoffs)。为了融合博弈思想实现混合算法中不同算法的调用,将混合算法中的要素与博弈论中的要素进行一一映射,如表 1 所示。

表 1 博弈论和混合优化算法的要素映射

	博弈论	混合算法
参与人	指一个博弈中的决策主体,它的目的是通过选择行动来最大化自己的效用水平	指混合优化算法中参与独立寻优的子群,子群中的所有个体作为一个主体进行策略选择
策略	博弈中,每个参与人在给定的信息的情况下可选的行动方案	指子群寻优时可供选择的单种优化算法,每种算法即代表一种策略
收益	博弈的结果称为收益。每个参与人在一局博弈结束时的收益不仅与该参与人自身所选择的策略有关,而且与所有参与人选定的一组策略有关	指某一策略组合下各子群中解的优劣程度,用适应值来评价

1.2 混合算法的博弈模型

定义两个参与独立寻优的子群 subswarm₁ 和 subswarm₂ 及两个策略 scheme₁ 和 scheme₂,则参与人集合 $N = \{1, 2\}$, 每一个子群 $i \in N$ 是博弈问题中的决策主体,且假设两个子群都具有集体理性,博弈的目的是全局收益的最大化;子群拥有相同的策略集合,subswarm₁ 的策略集合 $S_i = \{scheme_1, scheme_2\}$; 博弈结果用 (q_1, q_2) 来表示, q_1, q_2 相应为 subswarm₁ 和 subswarm₂ 的适应值,采用式(1)计算。

$$q_i = \frac{avg_{ij}}{\sum_{i=1}^2 avg_{ij}} best_j + SC_{ij} \quad (1)$$

其中: SC_{ij} 是子群 i 采用策略 j 迭代获得的最优适应值; avg_{ij} 是子群 i 采用策略 j 迭代获得的平均适应值; $best_j$ 是策略 j 目前最优适应值,采用式(2)计算; $avg_{ij} / \sum_{i=1}^2 avg_{ij}$ 表示当两个子群竞争同一策略时,平均适应值高的子群从该策略中获得更高的收益。

$$best_j = \max(SC_{1j}, SC_{2j}) \quad (2)$$

1.3 博弈模型的求解

两个子群的收益记为 $P_k = (P_{sub1}, P_{sub2})$, 子群 i 的收益 P_{subi} 不仅取决于自己的策略而且与另一子群的策略有关,它是策略组合的函数,策略组合的优劣用支付效用矩阵表示,如表 2 所示。 $P_{sub1} = q_1 (scheme_i, scheme_j)$, $P_{sub2} = q_2 (scheme_i, scheme_j)$, $(scheme_i, scheme_j)$ 是策略组合。

表 2 支付效用矩阵

子群	策略	subswarm ₂	
		scheme ₁	scheme ₂
subswarm ₁	scheme ₁	P_1	P_2
	scheme ₂	P_3	P_4

其中: $P_1 = (\frac{avg_{11}}{avg_{11} + avg_{21}} \times best_1 + SC_{11}, \frac{avg_{21}}{avg_{11} + avg_{21}} \times best_1 + SC_{21})$
 $P_2 = (best_1 + SC_{11}, best_2 + SC_{22})$ $P_3 = (best_2 + SC_{12}, best_1 + SC_{21})$
 $P_4 = (\frac{avg_{12}}{avg_{12} + avg_{22}} \times best_2 + SC_{12}, \frac{avg_{22}}{avg_{12} + avg_{22}} \times best_2 + SC_{22})$ (3)

经典的博弈论以个人理性为前提,每个参与人的目的是最大化自己的收益函数。本文假设两个子群是强调集体理性的参与人,双方达成合作协议,追求整个种群效益 P^* 的最大化。

$$P_k^* = P_{sub1} + P_{sub2} \quad k = 1, 2, 3, 4 \quad (4)$$

在这些假设下,子群个体的位置可由最优策略组合所对应的策略获得

$$X_i^{t+1} = X_i^t + \delta x_i^{t+1}(\bar{P}) \quad (5)$$

其中: X_i^t 表示子群 i 在第 t 次迭代中的位置, \bar{P} 表示最优策略组合, $\delta x_i^{t+1}(\bar{P})$ 表示子群 i 选择最优策略组合所对应的策略迭代的步长。

“对应的策略迭代的步长”改成“对应策略的迭代步长”

$$\bar{P} = \max(P_1^*, P_2^*, P_3^*, P_4^*) \quad (6)$$

2 基于博弈论的混合优化算法

“上面的”改成“以上”

针对上面的分析,给出基于博弈论的混合优化算法流程,步骤如下:

- 在搜索空间内随机产生两个个体数量相同的子群,设置其搜索边界;初始化两个策略;初始化最大迭代次数 MaxDT 和独立寻优代数 MaxJT;初始化全局最优解 fbest。
- 对于每一次迭代,分别计算子群 i 采用策略 j 的最优适应值 SC_{ij} , 平均适应值 avg_{ij} , 记录策略 j 的最优适应值 $best_j$ 。
- 对于每一个子群 i , 根据支付效用矩阵计算其收益 P_{subi} , 计算种群收益 P_k^* 。
- 计算最优策略组合 \bar{P} 。
- 对于每一个子群 i , 依据式(5)按照选定的最优策略 j 独立寻优 MaxJT 代, 计算最优解。
- 对于每一个子群 i , 判断其最优解是否小于全局最优解 fbest, 是则更新 fbest。
- 判断是否达到最大迭代次数 MaxDT, 是则输出全局最优解 fbest, 否则转步骤 b)。

3 仿真实验

典型的群智能优化算法有很多,考虑到 CS、PSO 和 DE 原理简单,控制参数较少,且在搜索机制上存在较大差异,本文分别选取策略集合 $\{scheme_1, scheme_2\}$ 等于 $\{CS, PSO\}$ 、 $\{DE, PSO\}$ 和 $\{DE, CS\}$, 提出 CS-PSO、DE-PSO 和 DE-CS 算法来测试基于博弈论的混合优化算法的性能。

3.1 测试函数

使用本文提出的 CS-PSO、DE-PSO 和 DE-CS 对四个标准测试函数进行搜索运算,并与 CS、PSO 和 DE 进行比较,四个测试函数如表 3 所示。

表 3 四个标准测试函数

函数名	表达式	搜索范围	理论全局最优解
sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	$f(0) = 0$
griewank	$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n (\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	$f(0) = 0$
	$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	$f(0) = 0$
ack1	$f(x) = 20 + e - 20e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - \frac{1}{e} \sum_{i=1}^n \cos 2\pi x_i$	$[-32, 32]$	$f(0) = 0$

其中: f_1 是 sphere 函数,此函数为非线性的对称单峰函数,不同维之间是可分离的,此函数相对比较简单,大多数算法都能够达到优化效果,主要用于测试算法的寻优精度; f_2 是 griewank 函数,此函数为旋转、不可分离的、可变维数的多峰函数,随着维数的增加,局部最优的范围越来越窄,从而使得找寻全局最优值就会变得相对容易; f_3 是 rastrigin 函数,此函数使用了余弦函数产生大量的局部最小值,是一个典型具有大量局部最优点的复杂多峰函数,此函数很容易使算法陷入局部最优,而不

“就会”去掉

“典型”改成“典型的”

“它的”改成“其”

“给定的信息”改成“给定信息”

“、”改成“、”

“从”改成“将从”

“P3=(best2+SC12, best1+SC21)”应另起一行

空两格

能得到全局最优解; f_4 是 ackley 函数, 它的拓扑结构的特征是外部区域几乎平坦(由于主导函数是指数函数), 中间出现一个孔或者峰(由于余弦波形的调整), 从而变得不平滑, 具有大量的局部最优解。

3.2 参数设置

本次仿真实验硬件环境为 Intel® Core™ i3-3240 CPU/3.40GHz/4.00GB RAM; 软件平台为 Windows7 操作系统; 计算软件为 MATLAB 2014a。CS 的参数设置参考文献[11]: $\beta = 1.00, pa = 0.25$; DE 的参数设置参考文献[12], $F = 0.50, Cr = 0.5$; PSO 采用惯性权重线性减小的速度更新公式^[13], $\max\omega = 0.9, \min\omega = 0.4$, 学习因子 $c_1 = c_2 = 2$ 。CS-PSO、DE-PSO 和 DE-CE 中策略的参数设置与 CS、DE 和 PSO 相同。单种优化算法的种群规模 $N = 50$, CS-PSO、DE-PSO 和 DE-CE 产生的是数目相等的个体, 即 $\text{subswarm}_1 = \text{subswarm}_2 = 25$ 。实验以函数的总计算次数 (FEs = 5000) 为终止条件, 其中 CS-PSO、DE-PSO 和 DE-CE 的迭代次数 $\text{MaxDT} = 100$, 独立寻优代数 $\text{MaxJT} = 50$ 。

3.3 仿真结果与分析

针对每一种算法, 程序分别独立运行 30 次, 记录 30 次结果的最优适应值 (best)、最差适应值 (worst)、平均适应值 (mean) 以及均方差 (std), 测试函数的维度 D 取 30, 仿真结果如表 4 所示。

表 4 四个标准测试函数的仿真结果比较 ($D = 30$)

函数	实验结果	DE	CS	PSO	DE-CS	PSO-CS	DE-PSO
sphere	best	4.98E-77	1.86E-29	5.43E-04	4.22E+04	0.00E+00	0.00E+00
	worst	2.83E-74	4.47E-28	1.00E+04	8.15E+04	0.00E+00	0.00E+00
	mean	4.00E-75	1.19E-28	3.33E+03	6.39E+04	0.00E+00	0.00E+00
	std	6.78E-75	9.56E-29	4.79E+03	7.95E+03	0.00E+00	0.00E+00
griewank	best	0.00E+00	0.00E+00	6.63E-04	4.10E+02	0.00E+00	0.00E+00
	worst	1.48E-02	7.11E-15	2.70E+02	6.91E+02	0.00E+00	0.00E+00
	mean	1.64E-03	4.70E-16	3.02E+01	5.86E+02	0.00E+00	0.00E+00
	std	3.90E-03	1.44E-15	5.97E+01	6.16E+01	0.00E+00	0.00E+00
rastrigin	best	1.09E+01	2.33E+01	2.61E-03	3.45E+02	0.00E+00	0.00E+00
	worst	4.66E+01	4.38E+01	2.31E+02	4.73E+02	0.00E+00	5.38E+01
	mean	1.67E+01	3.35E+01	6.66E+01	4.29E+02	0.00E+00	5.51E+00
	std	7.30E+00	5.07E+00	5.32E+01	3.15E+01	0.00E+00	1.04E+01
ackley	best	2.93E-78	1.47E-30	9.42E-06	5.31E+03	0.00E+00	0.00E+00
	worst	5.26E-75	5.65E-29	2.05E+03	8.24E+03	0.00E+00	0.00E+00
	mean	8.55E-76	1.21E-29	4.44E+02	6.75E+03	0.00E+00	0.00E+00
	std	1.39E-75	1.13E-29	5.82E+02	6.42E+02	0.00E+00	0.00E+00

从表 4 可以看出, 同原单种优化算法相比, PSO-CS 在四个基准测试函数上都能找到全局最优解, 且表现出良好的稳定性, 明显优于 PSO 和 CS; DE-PSO 也表现出比原单种优化算法更好的搜索性能, 除了函数 f_3 均能搜索到全局最优解, 搜索精度和稳定性大幅提高; 而 DE-CS 则表现出比原单种优化算法更差的搜索效果, 所记录的最优适应值、最差适应值、平均适应值以及均方差均不及 DE 和 CS。

究其原因, CS 在寻优过程中使用 Levy 飞行机制进行解的更新, 很容易从一个区域跳到另一个区域, 使解的随机性大, 另外, 算法还采用随机淘汰机制, 又能有效避免陷入局部最优解, 但正是由于 Levy 飞行的高随机性, 使得算法搜索过程盲目性很大, 收敛速度变得很慢, 尤其是在最优解附近, 搜索效率非常低^[14]; DE 完全平行地以随机性的概率转换机制代替序贯地运用确定性的机理转换机制, 采用群体协同搜索, 具有利用个体局部信息和群体全局信息指导算法进一步搜索的能力, 但很易造成早熟或求解时间过长, 最后难以达到全局最优^[15]; PSO 中全局最优粒子给其他粒子共享信息, 群体粒子跟随当前群体最

“结构的特征”改成“结构特征”

优粒子向问题最优解的方向移动, 因此收敛速度较快。但是, 单向的信息流动使得 PSO 在应用中易陷入局部极值点, 进化后期收敛慢导致精度较差

分析算法的搜索机制, 可以看到: CS 和 DE 的搜索过程随机性较大, 而 PSO 的搜索则表现出极强的导向性, 将 PSO 同 CS 和 DE 混合, 有效平衡了搜索中的随机性和导向性, 使混合策略能优势互补, 扬长避短, 因此 PSO-CS 和 DE-PSO 都表现出良好的搜索性能; 相反, DE-CS 混合了两种随机性强的搜索策略, 使得搜索过程盲目性很大, 收敛速度变慢, 因此寻优结果不佳。

为了反映 PSO-CS 和 DE-PSO 的收敛速度, 图 1 和 2 分别给出了两种混合优化算法在维数 $D = 30$ 时迭代 500 次的函数收敛曲线。从图 1 和 2 可以看出, PSO-CS 和 DE-PSO 的收敛速度明显优于单种优化算法, 寻优能力相对较好。仿真结果表明, 本文的 PSO-CS 和 DE-PSO 混合优化算法不仅提高了搜索精度, 而且加快了算法的收敛速度。

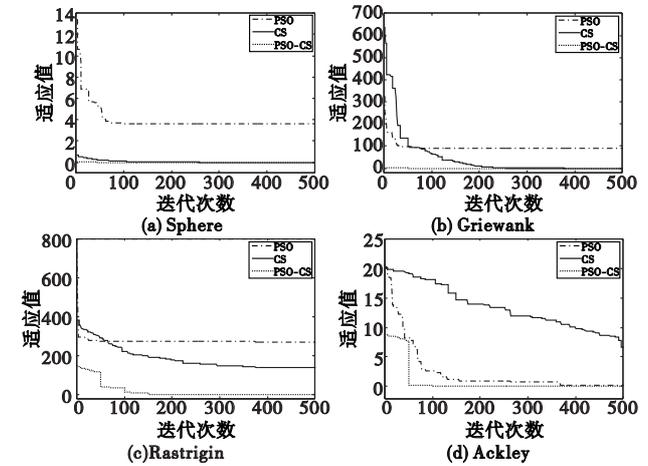


图 1 PSO-CS 与原单种优化算法的收敛曲线对比

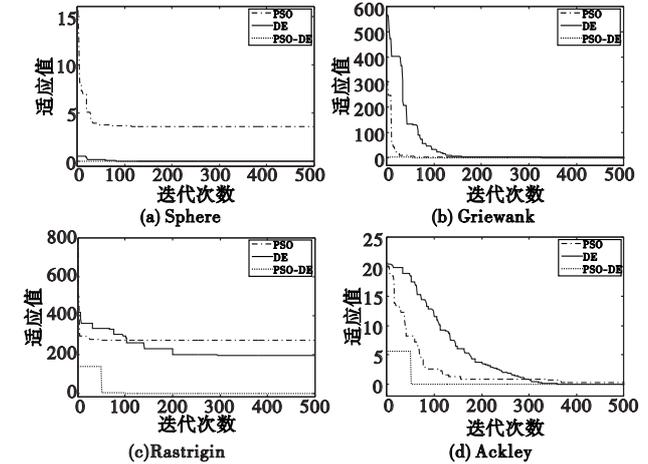


图 2 DE-PSO 与原单种优化算法的收敛曲线对比

4 结束语

本文借鉴博弈论在策略优化中的相关思想, 提出多子群——多策略的混合优化机制, 通过构建支付效用矩阵为参与寻优的子群选择最优策略, 并控制混合算法中单种算法的调用顺序。为了验证混合机制的有效性, 选定 CS、PSO 和 DE 作为子群寻优的策略, 提出 CS-PSO 算法、DE-PSO 算法和 DE-CS 算法, 并基于四个标准测试函数仿真。仿真结果表明: 当混合策略中的单种优化算法具有不同的搜索特性时, 所提出的混合算法是行之有效的, 且在搜索精度和收敛速度等方面均表现出良好的优化性能。

(下转第 页)

“以函数”改成“以达函数”

“产生的是数目相等”改成“产生数目相等”

去掉“能”

“CS和DE”改成“CS或DE”

“很大”改成“增大”

“随机性大”改成“随机性增大”

“又能”改成“可”

“过长”改成“过长的问題”

“仿真”改成“进行仿真”

(上接第 页)

参考文献:

- [1] Ho Y C, Pepyne D L. Simple explanation of the no-free-lunch theorem and its implications[J]. *Journal of Optimization Theory and Applications*, 2002, 115(3):549-570.
- [2] 暴励, 曾建潮. 一种双种群差分蜂群算法[J]. *控制理论与应用*, 2011, 28(2):266-272.
- [3] Rao A R M, Lakshmi K. Discrete hybrid PSO algorithm for design of laminate composites with multiple objectives[J]. *Journal of Reinforced Plastics and Composites*, 2011, 30(20):1703-1727.
- [4] 梁旭, 黄明. 现代智能优化混合算法及其应用[M]. 北京: 电子工业出版社, 2011:32.
- [5] Huang Zhimin, Li S X. Co-op advertising models in manufacturer-retailer supply chains: a game theory approach[J]. *European Journal of Operational Research*, 2001, 135(3):527-524.
- [6] Scharlemann J P W, Eckel C C, Kacelnik A, *et al.* The value of a smile: game theory with a human face[J]. *Journal of Economic Psychology*, 2001, 22(5):617-640.
- [7] Pan Feng, Xi Bao, Wang Lin. Environmental regulation strategy analysis of local government based on evolutionary game theory[C]//Proc of International Conference on Management Science & Engineering. [S. l.]:IEEE Press, 2014:17-19.
- [8] Saad W, Han Z, Debbah M, *et al.* Coalitional game theory for communication networks [J]. *IEEE Signal Processing Magazine*, 2009, 26(5):77-97.
- [9] Li Xinyu, Gao Liang, Li Weidong. Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling[J]. *Expert Systems with Applications*, 2012, 39(1):288-297.
- [10] 马胜蓝, 叶东毅. 一种基于博弈策略的群智能属性约简算法[J]. *计算机工程与应用*, 2012, 48(1):145-149.
- [11] Yang Xinshe, Deb S. Engineering optimisation by cuckoo search[J]. *International Journal of Math Modelling Numerical Optimisation*, 2010, 1(4):330-343.
- [12] Das S, Suganthan P N. Differential evolution: a survey of the state-of-the-art[J]. *IEEE Trans on Evolutionary Computation*, 2011, 15(1):4-31.
- [13] Feng Yong, Teng Guifa, Wang Aixin, *et al.* Chaotic inertia weight in particle swarm optimization[C]//Proc of the 2nd International Conference on Innovative Computing, Information and Control. Washington DC:IEEE Computer Society, 2007:1899-1902.
- [14] 钱伟懿, 侯慧超, 姜守勇. 一种新的自适应布谷鸟搜索算法[J]. *计算机科学*, 2014, 41(7):279-282.
- [15] 戈剑武, 祁荣宾, 钱锋, 等. 一种改进的自适应差分进化算法[J]. *华东理工大学学报:自然科学版*, 2009, 35(4):600-605.