

混合细菌觅食算法求解整数规划问题*

Research of Hybrid Bacterial Foraging Algorithm for Integer Programming

麦雄发^{1,2}, 李玲³

MAI Xiong-fa^{1,2}, LI Ling³

(1. 广西师范学院数学科学学院, 广西南宁 530023; 2. 广西师范学院科学计算与智能信息处理广西高校重点实验室, 广西南宁 530023; 3. 广西师范学院继续教育学院, 广西南宁 530001)

(1. School of Mathematical Sciences, Guangxi Teachers Education University, Nanning, Guangxi, 530023, China; 2. Key Lab of Scientific Computing & Intelligent Information Processing in Universities of Guangxi, Nanning, Guangxi, 530023, China; 3. School of Continuing Education, Guangxi Teachers Education University, Nanning, Guangxi, 530001, China)

摘要: 利用混合细菌觅食算法(PO-BFA)求解整数规划问题, 并与量子粒子群算法(QPSO)的求解结果进行对比。经过适当的参数设置混合细菌觅食算法可以有效地解决整数规划问题, 在搜索过程中没有出现早熟现象, 而且 PO-BFA 在求解整数规划问题上的整体性能比 QPSO 更优。

关键词: 细菌觅食算法 量子粒子群算法 整数规划

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1002-7378(2012)03-0187-03

Abstract: The performance of the bacterial foraging algorithm combined with particle swarm optimization and opposition-based learning(PO-BFA) for integer programming was investigated. With the proper setting, the experimental results indicated that PO-BFA efficiently solved the problems of integer programming and converged faster than the QPSO algorithm in most cases.

Key words: bacterial foraging algorithm, quantum-behaved particle swarm optimization, integer programming

整数规划(Integer Programming, IP)是要求决策变量取整数值的优化问题, 是组合最优化理论中的一个核心问题, 在解决任务调度、资金分配、股票分析、网络设计等问题中具有重要的应用。整数规划的一般形式为

$$\min f(x)$$

$$\text{s. t. } g_k(x) \leq 0, k = 1, 2, \dots, p,$$

式中 $x = [x_1, x_2, \dots, x_n]^T$ 为整型变量。

求解整数规划的传统方法主要有分枝限界法、

割平面法、完全枚举法等。由于传统方法在求解大规模整数规划问题时计算时间极大甚至无法求解, 故基于智能优化算法的求解整数规划引起了很多人的关注。文献[1]提出用遗传算法求解整数规划问题的方法, 而文献[2]较早地应用粒子群算法求解整数规划问题, 文献[3]应用蚁群算法解决典型的无约束整数规划问题, 文献[4]和文献[5]提出用量子粒子群算法解决整数规划问题的方法, 均取得了较好的效果。

2002年 K. M. Passino 基于细菌觅食行为过程提出了一种新型仿生随机搜索算法——细菌觅食算法(BFA)^[6]。随后文献[7]结合量子粒子群算法(PSO)^[8]和对立学习^[9,10]提出混合细菌觅食算法(PO-BFA)。本文应用该混合细菌觅食算法求解整数规划问题, 并与量子粒子群算法的求解结果进行

收稿日期: 2012-02-27

修回日期: 2012-07-20

作者简介: 麦雄发(1974-), 男, 副教授、硕士, 主要从事智能计算及其应用研究。

* 科学计算与智能信息处理广西高校重点实验室项目(GXSCIP201204); 广西教育厅科研基金项目(201106LX310, 201204LX216)资助。

对比。

1 混合细菌觅食算法

BFA 模拟细菌群体的行为,包括趋化、繁殖、驱散 3 个步骤,核心是趋化性操作,一般包括翻转和前进两个过程。算法详情可参考文献[6]。虽然 BFA 在不同的应用中取得了一些应用,但是由于 BFA 中采用任意方向进行翻转,收敛较慢。而 PSO 中利用了个体极值和全局极值来更新位置,收敛速度较快。混合细菌觅食算法(PO-BFA)用 PSO 中粒子移动的方法代替细菌的趋化步骤,即在细菌趋化中采用如下公式(1),(2)进行细菌位置的更新:

$$v_i(t) = \omega v_i(t-1) + c_1 r_1(t)(x_{pbest_i} - x_i(t)) + c_2 r_2(t)(x_{gbest} - x_i(t)), \quad (1)$$

$$x_i(t) = x_i(t-1) + v_i(t), \quad (2)$$

其中, $x_i(t)$ 表示细菌的位置, $v_i(t)$ 表示细菌的速度, x_{pbest_i} 为细菌个体的历史最优位置, x_{gbest} 为细菌群体的历史最优位置, ω 为非负惯性权重; c_1, c_2 为非负学习因子; r_1, r_2 为 $[0, 1]$ 之间的随机数。

另外,在初始化和进化过程中,利用对立学习,可以加快种群的收敛速度^[9,10]。PO-BFA 在初始化和趋化结束后,对每个细菌个体 x ,先用公式(3)产生其近似对立数:

$$x_q = \text{rand}(c, x), \quad (3)$$

其中, c 是细菌群体的的中心; $\text{rand}(c, x)$ 是均匀分布于 $[c, x]$ 的随机数。然后对比 x 和 x_q 的适应度,适应度更优的细菌个体保留在种群中,另一个则丢弃。

2 求解整数规划问题的结果及分析

2.1 测试函数及算法初始化

测试函数 $F_1 \sim F_7$ 及其性能^[5]如下。

$$F_1(x) = \|x\|_1, \text{最优解 } x=0, \text{最优值 } 0,$$

$$F_2(x) = x^T x, \text{最优解 } x=0, \text{最优值 } 0,$$

$$F_3(x) = -(15 \ 27 \ 36 \ 18 \ 12)x +$$

$$x^T \begin{bmatrix} 35 & -20 & -10 & 32 & -10 \\ -20 & 40 & -6 & -31 & 32 \\ -10 & -6 & 11 & -6 & -10 \\ 32 & -31 & -6 & 38 & -20 \\ -10 & 32 & -10 & -20 & 31 \end{bmatrix} x,$$

最优解分别为 $x=(0 \ 11 \ 22 \ 16 \ 6)^T, x=(0 \ 11 \ 23 \ 17 \ 6)^T$,最优值分别为 -737 ,

$$F_4(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2, \text{最优解 } x=(3 \ 2)^T, \text{最优值 } 0,$$

$$F_5(x) = (9x_1^2 + 2x_2^2 - 11)^2 + (3x_1 + 4x_2^2 - 7)^2, \text{最优解 } x=(1 \ -1)^T, \text{最优值 } 0,$$

$$F_6(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \text{最优解 } x=(1 \ 1)^T, \text{最优值 } 0,$$

$$F_7(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \text{最优解 } x=(0 \ 0 \ 0 \ 0)^T, \text{最优值 } 0.$$

各函数所对应的维数、种群数目和最大迭代次数如文献[5]中表 2 所示。PO-BFA 中设定最大驱散代数 $Ned=2$,最大繁殖代数 $Nre=20$,最大趋化代数 $Nc=20$,驱散概率 $Ped=0.25$;粒子群参数 $C1=C2=2, \omega$ 从 1.0 线性递减到 0.4。

算法的初始化参数:初始种群均匀分布在 $[-100, 100]^D$ 。其中 D 是对应测试函数的维数。为了相互对比。PO-BFA 算法和 QPSO 算法每执行 S 次(S 为种群大小)适应度计算函数算一代。每个测试函数将被重复运行 50 次,并记录所得正确解的成功率和每个测试函数的平均迭代次数。细菌更新位置后截至整数。

2.2 结果及分析

用 PO-BFA 算法对测试函数 $F_1 \sim F_7$ 进行试验,并与文献[5]中 QPSO 的相应测试函数中成功率为 100%的最好成绩进行对比(表 1)。

表 1 PO-BFA 与 QPSO 测试函数对应的成功率和平均迭代次数

测试函数	D	QPSO		PO-BFA	
		成功率(%)	迭代次数	成功率	迭代次数
F_1	5	100	20.27	100	5.7
	10	100	24.77	100	3.14
	15	100	25.8	100	2.92
	20	100	28.97	100	2.16
	25	100	31.23	100	2.06
F_2	30	100	34.2	100	2
F_3	5	100	20.22	100	4.8
F_4	5	100	125.03	100	374.90
F_5	2	100	55.98	100	28.66
F_6	2	100	8.95	100	7.94
F_7	2	100	40.1	100	6.9447
	4	100	65.79	100	2.8094

表 1 结果显示,PO-BFA 算法在所有 7 个测试函数中,成功率都达到了 100%,并且在 $F_1, F_2, F_4 \sim F_7$ 6 个测试函数上,平均迭代次数都小于 QPSO 算法。在函数 F_1 上对不同维度的问题进行测试的结果显示,维数越高,种群大小越大,PO-BFA 的优势越明显,特别在 30 维时,由于种群大小达到了 300,PO-BFA 在第 1 代就找到了最优解。但是在函

数 F_3 上,PO-BFA 的收敛速度没有 QPSO 快。

为了进一步分析结果,给出函数的寻优进化曲线。由于篇幅关系,只选取函数 F_1, F_2 和 F_7 进行说明。其中函数 F_1 取的是维数等于 5 时的数据。从图 1~3 可以看出:(1)应用了对立学习初始化方法的 PO-BFA 获得了更好的初始种群;(2)PO-BFA 收敛更快。函数 F_1 中,PO-BFA 最晚在第 8 代繁殖后已经达到最优值,而 QPSO 最晚在第 33 代繁殖时才达到最优值;对于函数 F_2 , PO-BFA 在第 18 代繁殖后已经达到最优值,而 QPSO 在第 35 代繁殖时才达到最优值;对于函数 F_7 , PO-BFA 在第 10 代已经达到最优值,而 QPSO 在第 104 代才达到最优值。

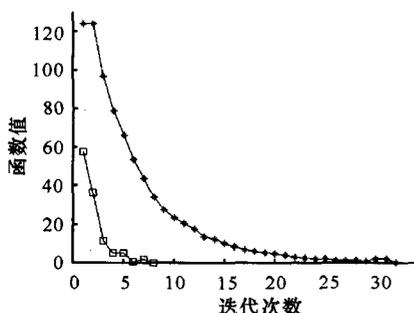


图 1 函数 F_1 的平均适应度的进化曲线
* :QPSO, □:PO-BFA。

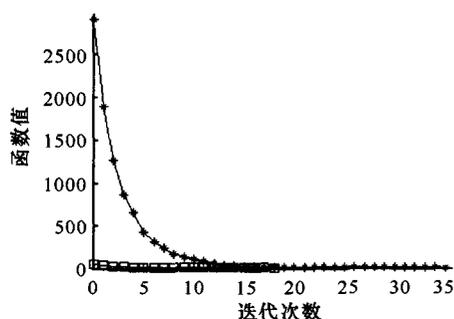


图 2 函数 F_2 的平均适应度的进化曲线
* :QPSO, □:PO-BFA。

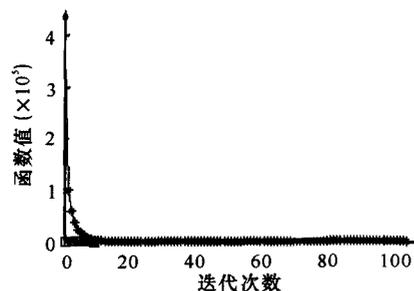


图 3 函数 F_7 的平均适应度的进化曲线
* :QPSO, □:PO-BFA。

3 结束语

通过 7 个测试函数的仿真实验可知,PO-BFA 算法和 QPSO 算法经过适当的参数设置均可以有效地解决整数规划问题,并在搜索过程中没有出现早熟现象,而 PO-BFA 算法在求解整数规划问题上的整体性能比 QPSO 更优。

参考文献:

- [1] Malasri S, Martin JR, Medina R A, et al. Solving mathematical programming problems using genetic algorithms[C] //Proc of the 3rd Conference on Computing in Civil Engineering, 1996:233-239.
- [2] Laskarie, Parsopoulouk, Vrahatism. Particle swarm optimization for integer programming[C] //Proc of Congress on Evolutionary Computation Washington DC: IEEE Computer Society, 2002:1582-1587.
- [3] 高尚, 杨静宇. 非线性整数规划的蚁群算法[J]. 南京理工大学学报, 2005, 29(增刊):120-123.
- [4] 杨荣华, 刘建华. 量子粒子群算法求解整数规划的方法[J]. 科学技术与工程, 2011, 11(33):8195-8198, 8202.
- [5] 刘静, 须文波, 孙俊. 基于量子粒子群算法求解整数规划[J]. 计算机应用研究, 2007, 24(3):79-81, 105.
- [6] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, 22(3):52-67.
- [7] 麦雄发, 李玲, 彭显忠. 基于 PSO 与对立学习的细菌觅食算法[J]. 计算机工程, 2011, 37(23):171-173.
- [8] Kennedy J, Eberhart R C. Particle swarm optimization [C]//Proc of the IEEE Int Conf on Neural Networks. Piscataway, NJ:IEEE Service Center, 1995:1942-1948.
- [9] Simon D. Biogeography-based optimization[J]. IEEE Trans on Evolutionary Computation, 2008, 12(6):702-713.
- [10] Tizhoosh H. Opposition - based learning: A new scheme for machine intelligence [C]//Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation, vol, 1:695-701, 2005.

(责任编辑:尹 闯)