GA 与 PSO 解 DE 与 LP 问题的效率比较①

庄思发

(韶关学院 数学与统计学院, 韶关 512005)

摘 要: 遗传算法和粒子群算法都具有很强的搜索能力, 在最优化问题中有着极其广泛的应用. 文章针对常微分方程(DE)近似解和一般线性规划(LP)问题的解利用遗传算法和粒子群算法求解, 深入的比较和分析了 GA 与 PSO 在这两种优化问题中的效率. 在固定其他参数而调整群体数量的基础上比较了 GA 与 PSO 在微分方程近似解和 LP 问题解的优化能力.

关键词: 遗传算法; 粒子群算法; 常微分方程; 线性规划; 优化问题

Efficiency Comparison of GA and PSO on DE and LP Problem

ZHUANG Si-Fa

(School of Math & Statistics, Shaoguan University, Shaoguan 512005, China)

Abstract: Genetic algorithm and Particle Swarm Optimization algorithm with strong search capability have a very wide range of applications in the optimization problem. This paper focuses on approximate solutions of ordinary differential equations and LP solutions, based on genetic algorithm and particle swarm algorithms, a comparison and analysis of the efficiency of two kinds of optimization problems is made. We then fix other parameters but adjust the particle population, in the purpose to compare optimization capability of GA and PSO in approximate solutions of differential equation and the LP problem.

Key words: genetic algorithm; particle swarm optimization; differential equation; liear program; optimization problem

1 引言

遗传算法(Genetic Algorithm,简写为 GA)是由美国 Michigan 大学的 Holland 教授于 1969 年提出, 后经 DeJong、Goldberg 等人归纳总结所形成的一类模拟进化算法. 它来源于达尔文的进化论、魏茨曼的物种选择学说和孟德尔的群体遗传学说. 遗传算法是模拟自然界生物进化过程与机制求解极值问题的一类自组织、自适应人工智能技术, 其基本思想是模拟自然界遗传机制和生物进化论而形成的一种过程搜索最优解的算法. [1]遗传算法的主要特点是直接对结构对象进行操作, 不要求函数可导及连续; 具有内在的隐并行性和极好的全局寻优能力; 遗传算法采用概率化的寻优方法, 能自动优化搜索空间, 自适应地调整搜索方向, 从而快速搜索到最优解.

粒子群算法(Particle Swarm Optimization, 简写为 PSO)是模拟鸟群捕食行为的一种智能算法. 粒子群算法具有较强的搜索能力, 根据模型信息实现全局优化得出最优解. [2] PSO 同遗传算法类似, 是一种基于迭代的优化算法. 系统在初始随机解的基础上, 通过迭代搜寻最优值. 由于没有像遗传算法一样采用交叉(crossover)以及变异(mutation), 只在解空间根据最优的粒子调整自己的搜索路线, 因此, PSO 的优势在于容易实现.

GA 与 PSO 在优化问题上都具有很强的搜索能力, 且算法比较相似, 有非常多的文章介绍了他们在各种优化问题中的应用. 而微分方程(简称 DE)和线性规划(简称 LP)是两种常见的解决日常经济问题的数学模型, 对于简单的微分方程或线性规划模型已经有非常成熟的且高效的算法. 然而我们实际中所面临的微分方程模型通常没有精确解或求解过程极其困难, 故而求其数值解为之首选. LP 问题也往往规模较大导致其精确解的计算代价太大. 在这样的情况下, 利用优化算法

① 收稿时间:2015-09-19;收到修改稿时间:2015-11-13 [doi: 10.15888/j.cnki.csa.005183]

寻求次优解就十分必要了. 秦俭修[3]、高雷阜等[4]成功地将 GA 或 PSO 算法应用于常微分方程和偏微分方程数值解, 而李妮等[5]、米永强等[6]则分别介绍了改进粒子群算法在优化问题中的应用, 其中李妮[5]提出一种基于佳点集理论的协同进化粒子群优化算法, 提高了 PSO 在非线性规划中的计算效率. 而隋允康等[7]通过增加一个关于设计变量的非线性等式约束代替设计变量的 0-1 离散限制, 成功地应用遗传算法解决 0-1 整数规划问题. 本文在他们研究成果的基础上, 利用 GA 和 PSO 的高效寻优能力求解 DE 和 LP 问题, 而求解效率是我们关注的重点. 就这两类问题, 对 GA 和 PSO 的算法效率进行深入的比较分析, 并给出使用 PSO 和 GA 求解此类问题时的最优参数.

2 算法简介

遗传算法中, 通过随机方式产生若干个所求问题的数字编码, 即染色体, 形成初始种群. 再根据优胜劣汰的原则产生下一代种群. 其基本步骤如下:

(1) 初始化种群;

(2) 计算每个个体的适应度值;

(3) 由个体适应度值决定进入下一代的个体;

(4) 按一定的概率进行交叉操作;

(5) 按一定的概率进行变异操作;

(6) 输出最优解.

通过编码组成初始群体后, 遗传操作的任务就是对群体的个体按照它们对环境适应度施加一定的操作, 从而实现优胜劣汰的进化过程. 遗传操作[8]使问题的解逐代进行优化, 并最终逼近最优解. 遗传操作包括三个基本遗传算子(genetic operator): 选择(selection)、交叉(crossover)和变异(mutation).

选择的目的是把较优的个体(解)遗传到下一代或通过配对交叉产生新的个体再遗传到下一代. 选择操作以适应度值为依据, 常用的选择算子有适应度比例方法、随机遍历抽样法和局部选择法.

交叉是指把两个父代个体的部分结构(编码)加以替换而生成新个体的操作, 交叉操作在遗传算法中占有非常重要的作用, 通过交叉, 遗传算法的搜索能力得以迅速提高. 交叉算子按一定的机率将种群中的两个个体随机地交换某些基因, 产生新的基因组合, 实现将有益基因组合在一起, 实现生成更优异的个体. 常用的交叉方法有单点交叉、多点交叉和均匀交叉等.

变异算子是以一定的概率对群体中的某些个体的某些基因作变动. 交叉算子作为主要算子拥有全局搜索能力, 而变异算子却有较强的局部搜索能力,作为遗传算法的辅助算子. 遗传算法通过交叉和变异使其具备兼顾全局和局部的均衡搜索能力.

以上可以看出, 遗传算法相对较复杂, 不易实现. 各个算子以有初始种群对算法的性能影响较大, 对特定问题通常需要不断重复实验以寻找最优参数. 其次, 遗传算法容易出现“早熟”, 即出现局部最优解. 于是出现了各种改进的遗传算法或混合遗传算法.

粒子群算法根据自身所找的最优解(个体极值)和群体所找到的最优解(作全局极值)来调整自己的速度和更新位置. 其基本步骤如下[8]:

(1) 初始化种群;

(2) 计算每个个体的适应度值;

(3) 更新个体极值和全局极值;

(4) 根据个体极值和全局极值更新自身飞行速度和位置;

(5) 输出最优解;

在步骤(4)中, 第 i 个粒子根据下面的公式更新自身速度和位置[9]

$$v_{id} = K\cdot [v_{id} + {}_1\cdot \text{rand}()\cdot(p_{id}-x_{id}) + {}_2\cdot \text{rand}()\cdot(p_{gd}-x_{id})]$$

$$x_{id} = x_{id} + v_{id}$$

其中 d=1,2, …, D, D 为粒子空间的维数. Maurice Clerc 为了提高 PSO 的收缩和控制速度的能力引入了收缩因子 K[10], 由下式计算:

$$K = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$$

其中 $\varphi=\varphi_1+\varphi_2$, $\varphi>4$, φ_1, φ_2 为学习因子.

步骤(4)中粒子也可以根据下面的公式更新自身速度和位置:

$$v_{id} = w\cdot v_{id} + y_1\cdot \text{rand}()\cdot(p_{id}-x_{id}) + y_2\cdot \text{rand}()\cdot(p_{gd}-x_{id})$$

其中的 w 称为惯性权重, Shi 和 Eberhart [11]指出, 当 Vmax(最大速度)较小时, 使用接近于 1 的惯性权重较合适; 当 Vmax 不是很小时, 使用权重 0.8 较好. 如果没有 Vmax 的信息, 使用 0.8 作为权重也是不错的选择. 当惯性权重较小时粒子群算法偏重于发挥局部搜索能力, 而当惯性权重较大时将偏重于发挥算法的全局搜索能力.

粒子群算法的优点是算法简单, 易于实现, 收敛能力强. 基本 PSO 算法针对单目标优化问题, 这导致它在现实中的应用受到限制. 因此, 不少学者提出了多目标粒子群算法[12-14]. 也有学者将粒子群算法推广至离散

优化的情形, 也取得了不错的进展, 如高海兵等[15].

3 GA与PSO在解DE问题中的效率比较

标准的微分方程数值解通常使用迭代解法[16], 如Euler 法、Runge-Kutta 法等, 然而这些方法不属于极小化问题, 不能直接适用于 GA 或 PSO 算法. 欲使微分方程数值解适于 GA 或 PSO, 必须将它们转化成极小化问题. 泰勒定理告诉我们, 若函数 y 在 $x=x_0$ 处存在直至 $n+1$ 阶导数, 则函数 y 可在 $x=x_0$ 处展开成其 n 阶泰勒公式[17]. 不带余项的泰勒多项式可看作是函数 y 的逼近. 因此, 可以考虑使用合适的多项式去近似微分方程的解 y, 通过最小化它们之间的误差来优化多项式的系数. 这样微分方程数值解的问题便可转化为极小化问题. 庄思发[18]、钟敏玲[19]详细介绍了使用遗传算法求解微分方程的近似解的方法. PSO 算法的应用上与 GA 用法完全类似, 在此不再赘述. 本文也采用文献[18]中的方法和算例, 为便于比较, 将区间调整至[0, 2].

例 1: 考虑如下微分方程

$$y' = -\frac{y}{5} + \exp(-\frac{x}{5})\cos x, y(0) = 0, x \in [0,2]$$

该微分方程的精确解为

$$y(x) = \exp(-\frac{x}{5})\sin(x)$$

根据文献[18]所述的方法利用多项式

$$P(x)=a_n x^n+\cdots+a_1 x+a_0$$

逼近上述微分方程的解, 再通过遗传算法和粒子群算法优化多项式的系数, 输出与真实解误差最小的多项式系数.

遗传算法和粒子群算法对各种参数如群体数量、遗传(迭代)代数等都比较敏感, 合适的参数意味着更高的搜索效率. 但在不同的优化问题之间一般又不存在最优的参数设置, 只有最合适的参数设置. Gerry Dozier[20]中对参数φ_1、φ_2 及群体数量对粒子群算法效率的影响进行了深入研究, 建议大多数情况下采用群体数量为 30 及 φ_1 = 2.8, φ_2 = 1.3. 实验中, 我们选择多个不同的群体数值, 以便更直观比较 GA 与 PSO 的搜索效率. 遗传算法的交叉比例设置为 0.7, 变异比例为 0.2, 其余参数使用 MATLAB 遗传算法工具箱默认.

实验时, 遗传算法和粒子群算法按以上设置的参数运行, 鉴于实验对象规划较小, 很快可收敛到满意解, 为便于比较分析, 将遗传算法的遗传代数和粒子群的飞行次数均设置为 100. 分别使用 2、3、4 和 5 次多项式逼近 y, 并循环运行 100 次, 计算平均搜索时间, 并选择最优结果进行比较, 3 次多项式的详细结果如表 1 所示, 其余结果见表 2.

表 1　微分方程三次多项式近似解结果比较

算法	群体数	平均耗时(秒)	最佳适应度值	多项式系数(a_3,a_2,a_1,a_0)	误差ε
GA	10	0.3008	0.0126	(0.0017, -0.3756, 1.0456, 0)	0.0022
	20	0.5893	0.0117	(-0.0013, -0.3697, 1.0495, 0)	6.6750e-4
	30	0.8020	0.0117	(-0.0010, -0.3706, 1.0502, 0)	6.5434e-4
	40	1.1500	0.0117	(-0.0013, -0.3695, 1.0494, 0)	6.6736e-4
	50	1.4114	0.0117	(-0.0007, -0.3714, 1.0509, 0)	6.4467e-4
PSO	10	0.1788	0.0117	(-0.0014, -0.3692, 1.0491, 0)	6.7501e-4
	20	0.3768	0.0117	(-0.0014, -0.3692, 1.0491, 0)	6.7488e-4
	30	0.5290	0.0117	(-0.0014, -0.3692, 1.0491, 0)	6.7500e-4
	40	0.7436	0.0117	(-0.0014, -0.3692, 1.0491, 0)	6.7497e-4
	50	0.9666	0.0117	(-0.0014, -0.3692, 1.0491, 0)	6.7499e-4

表 2　微分方程其它多项式近似解结果比较

算法	$P(x)$次数	平均耗时(秒)	最佳适应度值	多项式系数	误差ε
GA	2	0.6528	0.0117	(-0.3734, 1.0518, 0)	6.8351e-4
	4	0.8304	0.0014	(0.0343,-0.1457,-0.1838,0.9811,0)	1.9404e-4
	5	0.8025	0.0089	(-0.0253, 0.1607, -0.3366, -0.1278, 1.0274, 0)	0.0016
PSO	2	0.5315	0.0117	(-0.3734, 1.0518, 0)	6.8335e-4
	4	0.5432	3.3973e-5	(0.0342,-0.1380,-0.2092,1.0018,0)	1.1125e-6
	5	0.5473	3.8370e-5	(-0.0004, 0.0357, -0.1393, -0.2098, 1.0016, 0)	1.6603e-5

注: xe-n 表示 $x\times10^{-n}$

从结果可以看出, 遗传算法与粒子群算法对微分方程近似解都有非常好的逼近效果. 从总体效果上看, PSO 表现不但计算准确而且相当稳定. 除了群体数量为 10 时, 最优结果 0.0117 出现的次数为 89 外, 其余群体数量的 100 次的运行结果中, PSO 都得到 100 次相同的结果. 而 GA 在不同群体数量的 100 次的运行结果中最优结果 0.0117 出现的次数非常少. 群体数量为 10 时没有出现此结果, 而其余群体数的 100 次运行结果中最优结果出现的次数均少于 10 次. 误差表现方面, GA 与 PSO 两者表现较为接近. GA 的计算精确度与群体数量密切相关, 群体数量越多, 搜索精度越高. PSO 没有在群体数量方面表现出敏感的特性, 5 种群体数量对应的搜索精度几乎维持在相同水平. 在搜索时间上, GA 与 PSO 两者以不同群体数量运行时的时间消耗比较接近, 均随着群体数量增加而呈线性增长. 在相同群体数量时, PSO 的平均耗时均低于 GA, 说明其在解此类问题时的运行效率较 GA 更优.

就该微分方程的近似解而言, 由表 2 的数据可以看到, 当 $x \in [0, 2]$时, 使用 4 次多项式的逼迫效果最佳.

4 GA与PSO在解LP问题中的效率比较

LP 问题拥有非常成熟的单纯形法[21], 算法虽简单高效, 但其计算量大是致命弱点. 即使使用计算机辅助求解, 对于大规模的 LP 问题, 也不可避免地要面临大量的数据计算和存储. 遗传算法和粒子群算法在全局优化方面的出色表现, 使得它们在LP问题方面也有一席用武之地. 且它们的优势在于算法简单, 计算量小, 速度快. 与单纯形法相比, 遗传算法和粒子群算法的不足之处在于所求得的优化结果并非最优解, 而在实际应用中, 次优解也往往能完全满足要求.

例 2: 考虑如下线性规划问题[22]

$$\min f = -2x_1 - x_2 + 3x_3 - 5x_4$$

$$s.t.\begin{cases} x_1 + 2x_2 + 4x_3 - x_4 \le 6 \\ 2x_1 + 3x_2 - x_3 + x_4 \le 12 \\ x_1 + x_3 + x_4 \le 4 \\ x_i \ge 0, i = 1, ..., 4. \end{cases}$$

该问题的最优解为 X=(0, 8/3, 0, 4)$^{\mathrm{T}}$, 最优目标值为 -68/3.

这里我们分别采用遗传算法和粒子群算法对上述 LP 问题进行求解, 以 LP 问题的目标函数为适应度值, 其余参数及迭代次数和飞行次数等设置同第 3 节. 满足所有约束条件的解为可行解, 算法只有搜索到可行解才认为是成功的, 计算结果见表 3.

表 3 GA 与 PSO 在 LP 问题中的结果比较

算法	群体数	平均耗时(秒)	最佳目标值	决策变量	绝对误差
GA	10	0.1007	-21.5218	(0.2693, 2.5073, 0.0007, 3.6956)$^{\mathrm{T}}$	-1.1449
	20	0.1769	-22.5544	(0.0312, 2.6557, 0.0002, 3.9674)$^{\mathrm{T}}$	-0.1123
	30	0.2477	-22.6320	(0.0027, 2.6666, 0.0003, 3.9922)$^{\mathrm{T}}$	-0.0347
	40	0.3179	-22.6162	(0.0086, 2.6653, 0.0019, 3.9879)$^{\mathrm{T}}$	-0.0505
	50	0.3930	-22.6352	(0.0071, 2.6648, 0.0010, 3.9918)$^{\mathrm{T}}$	-0.0315
PSO	10	0.0210	-22.6662	(0, 2.6663, 0, 4.0000)$^{\mathrm{T}}$	-4.4042e-004
	20	0.0417	-22.6666	(0, 2.6666, 0, 4.0000)$^{\mathrm{T}}$	-5.6548e-005
	30	0.0606	-22.6666	(0, 2.6667, 0, 4.0000)$^{\mathrm{T}}$	-1.6668e-005
	40	0.0807	-22.6667	(0, 2.6667, 0, 4.0000)$^{\mathrm{T}}$	-6.2152e-006
	50	0.0995	-22.6667	(0, 2.6667, 0, 4.0000)$^{\mathrm{T}}$	-2.2061e-007

注: 绝对误差为 GA 或 PSO 运算所得目标值与真实目标值之差.

由上表可见, 遗传算法和粒子群算法在LP问题的应用中表现也相当出色, 但从与最优解的逼迫程度来看, PSO 算法明显占据上风, 得出的结果几乎与最优解完全重合. 若从可行解的搜索成功率来看, 遗传算法则优于粒子群算法. 100 次的结果中, PSO 的成功率分别为 46%、69%、83%、90%和 95%. 粒子群算法对 LP 问题的可行解的搜索成功率依赖于群体数量, 尽管如此, 在单次的搜索中, 一旦搜索成功, 其逼迫最优解的能力却是相当出色的. 5 种群体数的遗传算法对可行解的搜索成功率均为 100%. 这显然是遗传算法的特点所致, 每次迭代计算“遗传基因”发挥了关键的作用, 保持了解的可行性. GA 算法虽然没有失败的搜索,

但因每次运算得到的解之间有较大差距, 导致在单次搜索的误差也较大. 若从运算速度上进行比较, 则PSO 明显占优. 相同群体数量的平均搜索速度 PSO 远高于GA, 而PSO的误差也远小于GA. 综合来看, PSO在 LP 问题的表现可用“快、准、稳”三个字形容.

5 结束语

遗传算法和粒子群算法都是当今在优化问题中最好的算法之一, 两者在微分方程近似解和LP问题的表现上各有千秋, 总体上PSO稍优于GA. 在实际应用中, GA与PSO的群体数量设置在20以上会有较好的表现. 在微分方程近似解中, 应多次试验确定最优多项式的次数后再微调算法的参数搜索最优系数. 在LP问题中, 若使用 PSO, 为提高可行解的搜索成功率, 建议群体数量至少设为 50.

虽然 GA 与 PSO 在解微分方程近似解时都有很好逼迫效果, 但算法精度受群体数量、迭代次数等参数限制. 近似多项式的次数也影响计算精度, 因此计算前需先进行预估或经多次实验以确定最优参数. LP 问题的近似解法也受其规模的限制, 遗憾的是本文没有给出 GA 与 PSO 在不同规模的 LP 问题间的求解精度比较. 另外, 遗传算法和粒子群算法虽然都适用于连续优化问题, 但遗传算法处理高维优化问题时收敛速度则比较慢. 无论是 GA 还是 PSO 或是其他智能算法, 将他们应用于离散优化问题中去还是非常值得大家深入去研究的.

参考文献

1 葛继科,邱玉辉,吴春明,蒲国林.遗传算法研究综述.计算机应用研究, 2008,25(10):2911−2919.

2 王娟勤,李书琴.粒子群优化算法在求解线性规划方程中的效率研究.制造业自动化,2011,33(3):88−91.

3 秦俭修.遗传算法求解一阶常微分方程的数值解.科技传播,2010(16):118−119.

4 高雷阜,齐微.改进的粒子群算法在偏微分方程中的仿真应用.计算机仿真,2011,28(12):208−210.

5 李妮,欧阳艾嘉,李肯立.求解约束优化的改进粒子群优化算法.计算机应用,2012, 32(12):3319−3321.

6 米永强,高岳林.求解约束优化问题的改进粒子群优化算法.江西师范大学学报(自然科学版),2015,39(1):59−63.

7 隋允康,贾志超.0-1 线性规划的连续化及其遗传算法解法.数学的实践与认识,2010,40(6):119−127.

8 蔡自兴,徐光佑.人工智能及其应用.第三版.北京:清华大学出版社,2004:177−178.

9 Eberhart RC, Shi YH. Computational Intelligence-Concepts to Implementations. Morgan Kaufmann, 2007: 87−92.

10 Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. Proc. ICEC. Washington, DC. 1999. 1951−1957.

11 Shi YH, Eberhart R. A modified particle swarm optimizer. Evolutionary Computation Proceedings. IEEE. Anchorage, AK. 1998. 69−73.

12 沈佳杰,江红,王肃.基于多点速度向量的多目标粒子群算法改进.计算机工程与应用,2015,51(2):46−56.

13 李宁,邹彤,孙德宝等.基于粒子群的多目标优化算法. 计算机工程与应用,2005,41(23):43−46.

14 樊纪山,刘冠蓉,王鲁等.一种改进快速稳定的多目标优化算法.计算机应用研究,2007,24(4):452−454.

15 高海兵,周驰,高亮.广义粒子群优化模型.计算机学报, 2005,28(12):1980−1987.

16 李庆杨,王能超,易大义.数值分析.第 4 版.北京:清华大学出版社,2001:336−351.

17 华东师范大学数学系.数学分析.第三版.北京:清华大学出版社,2001:134−138.

18 庄思发.常微分方程近似解的遗传算法研究.韶关学院学报, 2012,33(10):10−14.

19 钟敏玲.遗传算法求解常微分方程应用实证分析.渤海大学学报(自然科学版),2009,30(2):158−161.

20 Carlisle A, Dozier G. An off-the-shelf PSO. Proc. of the Workshop on Particle Swarm Optimization. Indianapolis, IN. 2001. 1−6.

21 Luenberger DG, Ye YY. Linear and Nonlinear Programming. Springer, 2008: 46−50.

22 傅家良.运筹学方法与模型.上海:复旦大学出版社, 2006:14−24.