

粒子群优化算法综述

杨 维, 李歧强

(山东大学控制科学与工程学院, 济南 250061)

[摘要] 粒子群优化 (PSO) 算法是一种新兴的优化技术, 其思想来源于人工生命和演化计算理论。PSO 通过粒子追随自己找到的最好解和整个群的最好解来完成优化。该算法简单易实现, 可调参数少, 已得到广泛研究和应用。详细介绍了 PSO 的基本原理、各种改进技术及其应用等, 并对其未来的研究提出了一些建议。

[关键词] 群体智能; 演化算法; 粒子群优化

[中图分类号] TP301.6 **[文献标识码]** A **[文章编号]** 1009-1742(2004)05-0087-08

1 前言

从 20 世纪 90 年代初, 就产生了模拟自然生物群体 (swarm) 行为的优化技术。Dorigo 等从生物进化的机理中受到启发, 通过模拟蚂蚁的寻径行为, 提出了蚁群优化方法; Eberhart 和 Kennedy 于 1995 年提出的粒子群优化算法是基于对鸟群、鱼群的模拟。这些研究可以称为群体智能 (swarm intelligence)。通常单个自然生物并不是智能的, 但是整个生物群体却表现出处理复杂问题的能力, 群体智能就是这些团体行为在人工智能问题中的应用。粒子群优化 (PSO) 最初是处理连续优化问题的, 目前其应用已扩展到组合优化问题^[1]。由于其简单、有效的特点, PSO 已经得到了众多学者的重视和研究。

2 PSO 基本原理

2.1 基本 PSO 原理

粒子群优化算法是基于群体的演化算法, 其思想来源于人工生命和演化计算理论。Reynolds 对鸟群飞行的研究发现, 鸟仅仅是追踪它有限数量的邻居, 但最终的整体结果是整个鸟群好像在一个中心

的控制之下, 即复杂的全局行为是由简单规则的相互作用引起的。PSO 即源于对鸟群捕食行为的研究, 一群鸟在随机搜寻食物, 如果这个区域里只有一块食物, 那么找到食物的最简单有效的策略就是搜寻目前离食物最近的鸟的周围区域。PSO 算法就是从这种模型中得到启示而产生的, 并用于解决优化问题。另外, 人们通常是以他们自己及他人的经验来作为决策的依据, 这就构成了 PSO 的一个基本概念。

PSO 求解优化问题时, 问题的解对应于搜索空间中一只鸟的位置, 称这些鸟为“粒子” (particle) 或“主体” (agent)。每个粒子都有自己的位置和速度 (决定飞行的方向和距离), 还有一个由被优化函数决定的适应值。各个粒子记忆、追随当前的最优粒子, 在解空间中搜索。每次迭代的过程不是完全随机的, 如果找到较好解, 将会以此为依据来寻找下一个解。

令 PSO 初始化为一群随机粒子 (随机解), 在每一次迭代中, 粒子通过跟踪两个“极值”来更新自己: 第一个就是粒子本身所找到的最好解, 叫做个体极值点 (用 pbest 表示其位置), 全局版 PSO 中的另一个极值点是整个种群目前找到的最好解,

[收稿日期] 2003-08-05; **[修回日期]** 2003-09-08

[基金项目] “八六三”高技术资助项目 (2001AA413420), 山东省自然科学基金资助项目 (2003G01)

[作者简介] 杨 维 (1978-), 女 (满族), 山东济南市人, 山东大学控制科学与工程学院硕士研究生

称为全局极值点 (用 $gbest$ 表示其位置), 而局部版 PSO 不用整个种群而是用其中一部分作为粒子的邻居, 所有邻居中的最好解就是局部极值点 (用 $lbest$ 表示其位置)。在找到这两个最好解后, 粒子根据如下的式 (1) 和式 (2) 来更新自己的速度和位置。粒子 i 的信息可以用 D 维向量表示, 位置表示为 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$, 速度为 $\mathbf{V}_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$, 其他向量类似。则速度和位置更新方程为

$$v_{id}^{k+1} = v_{id}^k + c_1 \text{rand}_1^k (\text{pbest}_{id}^k - x_{id}^k) + c_2 \text{rand}_2^k (gbest_d^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

v_{id}^k 是粒子 i 在第 k 次迭代中第 d 维的速度; c_1, c_2 是加速系数 (或称学习因子), 分别调节向全局最好粒子和个体最好粒子方向飞行的最大步长, 若太小, 则粒子可能远离目标区域, 若太大则会导致突然向目标区域飞去, 或飞过目标区域^[2]。合适的 c_1, c_2 可以加快收敛且不易陷入局部最优, 通常令 $c_1 = c_2 = 2$; $\text{rand}_{1,2}$ 是 $[0, 1]$ 之间的随机数; x_{id}^k 是粒子 i 在第 k 次迭代中第 d 维的当前位置; pbest_{id} 是粒子 i 在第 d 维的个体极值点的位置 (即坐标); $gbest_d$ 是整个群在第 d 维的全局极值点的位置。为防止粒子远离搜索空间, 粒子的每一维速度 v_d 都会被钳位在 $[-v_{d \max}, +v_{d \max}]$ 之间, $v_{d \max}$ 太大, 粒子将飞离最好解, 太小将会陷入局部最优^[2]。假设将搜索空间的第 d 维定义为区间 $[-x_{d \max}, +x_{d \max}]$, 则通常 $v_{d \max} = kx_{d \max}$, $0.1 \leq k \leq 1.0$, 每一维都用相同的设置方法。基本 PSO 的流程可以描述为:

Step 1 初始化 初始搜索点的位置 X_i^0 及其速度 V_i^0 通常是在允许的范围内随机产生的, 每个粒子的 pbest 坐标设置为其当前位置, 且计算出其相应的个体极值 (即个体极值点的适应度值), 而全局极值 (即全局极值点的适应度值) 就是个体极值中最好的, 记录该最好值的粒子序号, 并将 $gbest$ 设置为该最好粒子的当前位置。

Step 2 评价每一个粒子 计算粒子的适应度值, 如果好于该粒子当前的个体极值, 则将 pbest 设置为该粒子的位置, 且更新个体极值。如果所有粒子的个体极值中最好的好于当前的全局极值, 则将 $gbest$ 设置为该粒子的位置, 记录该粒子的序号, 且更新全局极值。

Step 3 粒子的更新 用式 (1) 和式 (2) 对每一个粒子的速度和位置进行更新。

Step 4 检验是否符合结束条件 如果当前的迭代次数达到了预先设定的最大次数 (或达到最小错误要求), 则停止迭代, 输出最优解, 否则转到 Step 2。

PSO 的一些特点:

- 1) 基本 PSO 算法最初是处理连续优化问题的。
- 2) 类似于遗传算法 (GA) PSO 也是多点搜索。
- 3) 式 (1) 的第一项对应多样化 (diversification) 的特点, 第二项、第三项对应于搜索过程的集中化 (intensification) 特点, 因此这个方法在多样化和集中化之间建立均衡^[1]。

PSO 有全局版和局部版两种, 全局版收敛快, 但有时会陷入局部最优。局部版 PSO 通过保持多个吸引子来避免早熟, 假设每一个粒子在大小 l 的邻域内定义为一个集合

$$N_i = \{\text{pbest}_{i-l}, \text{pbest}_{i-l+1}, \dots, \text{pbest}_{i-1}, \text{pbest}_i, \text{pbest}_{i+1}, \dots, \text{pbest}_{i+l-1}, \text{pbest}_{i+l}\} \quad (3)$$

从 N_i 中选出最好的, 将其位置作为 $lbest$ 代替公式中的 $gbest$, 其他与全局版 PSO 相同。实验表明, 局部版比全局版收敛慢, 但不容易陷入局部最优^[3]。在实际应用中, 可以先用全局 PSO 找到大致的结果, 再用局部 PSO 进行搜索。

2.2 二进制 PSO

最初的 PSO 是从解决连续优化问题发展起来的, Eberhart 等又提出了 PSO 的离散二进制版, 用来解决工程实际中的组合优化问题^[4]。他们在提出的模型中将每一维 x_{id} 和 pbest_{id} 限制为 1 或者为 0, 而速度 v_{id} 不作这种限制。用速度来更新位置时, 如果 v_{id} 高一些, 粒子的位置 x_{id} 更有可能选 1, v_{id} 低一点则选 0, 阈值在 $[0, 1]$ 之间, 而有这种特点的函数就是 Sigmoid 函数:

$$\text{sig}(v_{id}^k) = 1 / (1 + \exp(-v_{id}^k)) \quad (4)$$

二进制版本 PSO 的公式为

$$\text{if } \rho_{id}^{k+1} < \text{sig}(v_{id}^{k+1}) \text{ then } x_{id}^{k+1} = 1; \\ \text{else } x_{id}^{k+1} = 0 \quad (5)$$

向量 ρ_{id}^{k+1} 的各个分量是 $[0, 1]$ 之间的随机数^[1]。为了防止 Sigmoid 函数饱和, 可以将 v_{id}^k 钳位在 $[-4.0, +4.0]$ 之间。二进制 PSO 其他部分与基本连续 PSO 类似。实验结果显示, 在大多

数测试函数中, 二进制 PSO 都比遗传算法速度快, 尤其在问题的维数增加时。

2.3 解决混和整数非线性规划 (MINLP) 问题的 PSO

如果用离散数字来表达当前的位置和速度, 或者在速度更新完成后对整数变量进行取整运算, 则可以在算法中同时处理连续和离散变量。这就为解决 MINLP 等问题提供了广阔天地, 文献 [5] 中已将解决 MINLP 的 PSO 用于无功功率和电压的控制问题。

3 PSO 的变化和改进

PSO 与 GA 有很多共同之处, 两者都随机初始化种群, 使用适应值来评价系统和进行一定的随机搜索。但 PSO 是根据自己的速度来决定搜索, 没有 GA 的明显的交叉和变异。如果将 pbest 也看作是种群的成员, 那么 PSO 也有一种很弱形式的选择^[6]。速度更新方程中如果没有 v_{id}^k 项, 可以解释为一种代数交叉, 而对 v_{id}^k 更好的理解是把每次迭代看成是一种不断适应的过程。GA 中染色体共享信息, 故整个种群较均匀地向最优区域移动。在 PSO 中 gbest (或者 lbest) 将信息传给其他粒子, 是单向的信息流动, 多数情况下, 所有的粒子可能更快地收敛于最优解。由于每个粒子在算法结束时仍然保持着其个体极值, 因此, 如将 PSO 用于调度和决策问题时, 可以给出多种有意义的选择方案。而基本遗传算法在结束时, 只能得到最后一代个体的信息, 前面迭代的信息没有保留。另外, PSO 算法对种群大小不十分敏感, 即种群数目下降时性能下降不是很大。

PSO 收敛快, 特别是在算法的早期, 但也存在着精度较低, 易发散等缺点。若加速系数、最大速度等参数太大, 粒子群可能错过最优解, 算法不收敛; 而在收敛的情况下, 由于所有的粒子都向最优解的方向飞去, 所以粒子趋向同一化 (失去了多样性), 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, 所能达到的精度也比 GA 低^[7], 因此很多学者都致力于提高 PSO 算法的性能。

3.1 收敛速度的改进

3.1.1 惯性权重 (inertia weight) 法 Shi 等提出了惯性权重的方法^[8]。惯性权重 w 是与前一次速度有关的一个比例因子, 而速度更新方程为

$$v_{id}^{k+1} = wv_{id}^k + c_1 \text{rand}_1^k (\text{pbest}_{id}^k - x_{id}^k) + c_2 \text{rand}_2^k (\text{gbest}_{id}^k - x_{id}^k) \quad (6)$$

用惯性权重来控制前面的速度对当前速度的影响, 较大的 w 可以加强 PSO 的全局搜索能力, 而较小的 w 能加强局部搜索能力^[9]。基本的 PSO 可以看作 $w = 1$, 因此在迭代后期缺少局部搜索能力。实验结果表明, w 在 $[0.8, 1.2]$ 之间 PSO 有更快的收敛速度。文献 [10] 中试验了将 w 设置为从 0.9 到 0.4 的线性下降, 使得 PSO 在开始时探索较大的区域, 较快地定位最优解的大致位置, 随着 w 逐渐减小, 粒子速度减慢, 开始精细的局部搜索 (这里 w 类似于模拟退火中的温度参数)。该方法加快了收敛速度, 提高了 PSO 算法的性能。当待解问题很复杂时, 该法使得 PSO 在迭代后期全局搜索能力不足, 导致不能找到要求的最优解, 这可以用自适应改变惯性权重来克服。

3.1.2 模糊惯性权重 (fuzzy inertia weight) 法 Shi 等提出用模糊控制器来动态自适应地改变惯性权重的技术^[11]。控制器的输入是当前惯性权重 w 和当前最好性能评价 (CBPE), CBPE 衡量 PSO 目前找到的最好候选解的性能; 输出是 w 的改变量。由于不同的问题有不同范围的性能评价, 因此需要对 CBPE 进行如下的规范化

$$\text{NCBPE} = \frac{\text{CBPE} - \text{CBPE}_{\min}}{\text{CBPE}_{\max} - \text{CBPE}_{\min}} \quad (7)$$

NCBPE 是规范化后的评价, CBPE_{\min} 和 CBPE_{\max} 依问题而定, 且需事先得知或者可估计。

模糊 w 法与线性下降 w 方法的比较结果显示, 后者不知道应该降低 w 的合适时机, 而自适应模糊控制器能预测使用什么样的 w 更合适, 可以动态地平衡全局和局部搜索能力。但是由于需知道 CBPE_{\min} 和 CBPE_{\max} 等, 使得模糊权重法的实现较为困难, 因而无法广泛使用^[3]。

3.1.3 压缩因子 (constriction factor) 法 Clerc 得出结论: 压缩因子有助于确保 PSO 算法收敛^[12]。这种方法的速度更新方程为

$$v_{id}^{k+1} = \chi (v_{id}^k + c_1 \text{rand}_1^k (\text{pbest}_{id}^k - x_{id}^k) + c_2 \text{rand}_2^k (\text{gbest}_{id}^k - x_{id}^k)) \quad (8)$$

其中 $\chi = 2 / |2 - \varphi - (\varphi^2 - 4\varphi)^{1/2}|$ 为压缩因子, $\varphi = c_1 + c_2$, 且 $\varphi > 4$ 。

约束因子法控制系统行为最终收敛, 且可以有效搜索不同的区域, 该法能得到高质量的解, 如果与此同时将每维 $v_{d \max}$ 设置为一维搜索空间的大小

$x_{d \max}$, 则可以得到更好的效果。

3.1.4 选择 (selection) 法 Angeline 提出的混和 PSO 主要应用 PSO 的基本机制以及演化计算所采用的自然选择机制^[6]。由于 PSO 搜索过程依赖 pbest 和 gbest, 所以搜索区域有可能被他们限制住了。自然选择机制的引入将会逐渐减弱其影响^[1]。测试结果显示, 虽然在大多数测试函数中选择法取得了比基本 PSO 更好的效果, 却在 Griewank 函数上得到了较差的结果。因此该法提高了 PSO 的局部搜索能力, 但同时削弱了全局搜索能力^[3]。

3.1.5 繁殖 (Breeding) 法 Løvbjerg 等人将遗传算法中的复制和重组这些称为繁殖的操作加入到全局版 PSO 中, 该方法是对按概率 P_i 选出的粒子进行如下式

$$\text{child}_1(X_i) = p_i \text{parent}_1(X_i) + (1.0 - p_i) \text{parent}_2(X_i) \quad (9)$$

$$\text{child}_2(X_i) = p_i \text{parent}_2(X_i) + (1.0 - p_i) \text{parent}_1(X_i) \quad (10)$$

$$\text{child}_1(V_i) = \frac{\text{parent}_1(V_i) + \text{parent}_2(V_i)}{|\text{parent}_1(V_i) + \text{parent}_2(V_i)|} \cdot |\text{parent}_1(V_i)| \quad (11)$$

$$\text{child}_2(V_i) = \frac{\text{parent}_1(V_i) + \text{parent}_2(V_i)}{|\text{parent}_1(V_i) + \text{parent}_2(V_i)|} \cdot |\text{parent}_2(V_i)| \quad (12)$$

的代数杂交操作, 产生子代的粒子取代父代。选择父代没有基于适应值, 防止了基于适应值的选择对那些多局部极值的函数带来潜在问题^[3]。 p_i 是 (0, 1) 间的随机数 (经验值约为 0.2)。理论上讲繁殖法可以更好地搜索粒子间的空间, 2 个在不同次优峰处的粒子经繁殖后, 可以从局部最优逃离。结果显示, 对单峰函数, 繁殖法虽略加快了收敛速度, 却不如基本 PSO 和 GA 找到的解好, 而对于多局部极值的函数, 繁殖 PSO 不仅加快了收敛速度, 而且找到了同样好或更好的解^[13]。

3.2 增加多样性的改进

3.2.1 空间邻域 (spatial neighborhoods) 法 基本的局部版 PSO 中 (见式 (3)) 粒子的邻域是基于粒子序号划分的。Suganthan 提出了基于粒子的空间位置划分的方案^[14]。在迭代中, 计算每一个粒子与群中其他粒子的距离, 记录任何 2 个粒子间的最大距离为 d_{\max} 。对每一粒子按照

$$\|X_a - X_b\| / d_{\max} \quad (13)$$

计算一个比值, 其中 $\|X_a - X_b\|$ 是当前粒子 a 到

粒子 b 的距离。而选择阈值 frac 根据迭代次数而变化。当另一粒子 b 满足 $\|X_a - X_b\| / d_{\max} < \text{frac}$ 时, 则 b 成为当前粒子的邻域。所有满足该条件的粒子组成 N'_i , 其他同原局部版 PSO。应用改进的邻域规则, 且采用时变 ω 的 PSO 在绝大多数测试中都取得了比全局版 PSO 更优良的性能。

3.2.2 邻域拓扑 (neighborhood topologies) 法 基本局部版 PSO (见式 (3)) 每个粒子左右两边的粒子都是它的邻居, 实际上这是应用了环形拓扑。Kennedy 测试了如图 2 所示的几种邻域拓扑结构^[15]。

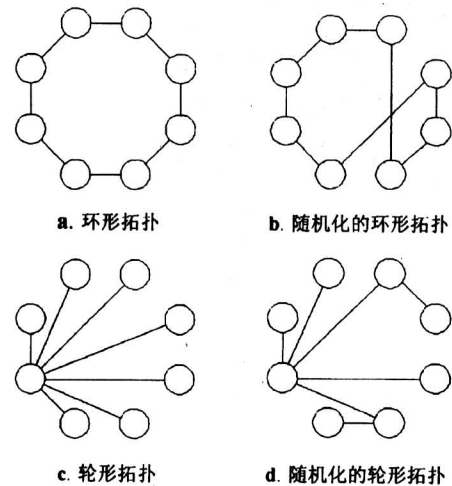


图 1 两种可能的邻域拓扑在有些边被随机改变前后的图形

Fig.1 A diagrammatic representation of two possible neighbourhood topologies, before and after edges have been randomly exchanged

结果表明, 拓扑非常影响算法的性能, 且最佳的拓扑形式因问题而定。如对有很多局部最优的函数, 轮形拓扑邻域算法能得到最好的结果, Kennedy 推测, 这是由于较慢的信息流动; 而对于单峰函数, 星形拓扑邻域算法可以产生较好的结果, 因为它有较快的信息流动 (这里的拓扑都是在粒子序号空间下的拓扑)。

3.2.3 社会趋同 (social stereotyping) 法 Kenney 提出了混和空间邻域和环形拓扑方法的另一个局部 PSO 版本, 称为社会趋同法^[16]。因为生活中人们往往是试图追随一个群体的共同观点, 而不是群体中某一个人的立场。将该思想应用到 PSO 中, 即不用每个粒子的经验而是用它所属空间聚类的共同经验来更新自己。

3.3 全局方法

3.3.1 序列生境技术 (SNT, sequential niche technique) 法 Beasley 等提出的最初使用在遗传算法中的序列生境技术可以系统地访问每一个全局极值^[17]。其思想是在找到每一个极值后, 都用下降函数来自适应地改变适应值函数, 如此算法就不会再回到该极值。虽然将 SNT 引入 PSO 会带来诸如参数选择、引入更多局部极值等问题, 但是该法能够枚举所有全局极值, 在多目标优化问题上还是很有意义的^[3]。

3.3.2 函数延伸 (function stretching) 法 为了减小定位所有全局极值的花费, Parsopoulos 等将自适应改变目标函数方法应用到了 PSO 中^[9]。他提出一个两步的转化过程来改变目标函数, 以防止 PSO 返回已经找到的局部极值。该方法能跳出局部最优, 有效定位全局最优点, 有助于 PSO 稳定地收敛, 虽然增加了计算量, PSO 的成功率却有明显的提高。但该方法不能枚举全部最优。

3.4 动态目标函数

很多现实优化问题的目标函数是随时间变化的, 因此已经有很多学者开始致力于这方面的研究。Parsopoulos 等的试验表明, 基本 PSO 就可以有效而稳定地在噪声环境中工作, 且在很多情况下, 噪声的存在还可以帮助 PSO 避免陷入局部最优^[18]。Carlisle 等的研究得出 PSO 不能跟随一个非静态的目标函数。Eberhart 等提出一种动态惯性权重法以试图解决这一问题^[2], 该方法令 $w = 0.5 + r(t)/2.0$, $r(t)$ 是 $(0,1)$ 间的随机数 (w 的平均值为 0.75), 常数 $c_1 = c_2 = 1.494$, 这是受到 Clerc 压缩因子方法的启发而得出的。结果显示 PSO 能跟随非静态目标函数, 比进化规划和进化策略得到的结果精度更高^[3]。

3.5 其他改进

3.5.1 保收敛 PSO (guaranteed convergence PSO)

若粒子的当前位置恰是全局最好位置, 那么速度更新方程式 (6) 就只剩下 v_{id}^k , 这将会导致早熟。文献 [19] 中提出一种确保 PSO 收敛到局部最优的改进方法 (GCP SO), 其策略是对全局最好粒子用一个新的更新方程, 方程将该粒子的位置复位到全局极值点, 并且使其在全局最好位置附近产生一个随机搜索, 而其他粒子仍用原方程更新。该方法较基本 PSO 的在收敛速度上有很大提高, 尤其是当粒子数较少时, 在单峰函数中性能提高明显。但

是该改进同基本 PSO 一样不能保证算法收敛到全局最优。

3.5.2 协同 PSO (cooperative PSO) 文献 [20] 提出了一种协同 PSO, 该方法是将 n 维向量分到 n 个粒子群中, 每个粒子群优化一维向量, 评价适应值时将分量合成一个完整向量。例如针对第 j 个粒子群, 除第 j 个分量外, 其他 $n-1$ 个分量都设为最好值, 不断用第 j 个粒子群中的粒子替换第 j 个分量, 得到第 j 维的最好值, 其他维相同。为将有联系的分量划分在一个群, 可将 n 维向量分到 k 个粒子群来优化, 则前 $n \bmod k$ 个粒子群是 $\lceil n/k \rceil$ 维的, 后 $k - (n \bmod k)$ 个粒子群是 $\lfloor n/k \rfloor$ 维的。称这种算法为 CPSO - S_k , 它在某些问题上有更快的收敛速度, 但该算法容易被欺骗, 而 GCPSO 可以保证收敛到局部最优解, 因此作者又将这两种方法结合, 结果显示, 结合版拥有两者的优点, 取得了较好的效果^[3]。

4 PSO 的应用

PSO 的优势在于算法的简洁性, 易于实现, 没有很多参数需要调整, 且不需要梯度信息。PSO 是非线性连续优化问题、组合优化问题和混合整数非线性优化问题的有效优化工具^[1]。目前已经广泛应用于函数优化、神经网络训练、模糊系统控制以及其他遗传算法的应用领域。PSO 最初应用到神经网络训练上, 在随后的应用中, PSO 又可以用来确定神经网络的结构。作为演化神经网络的例子, Eberhart 等已经成功用 PSO 来分析人类的帕金森综合症等颤抖类疾病^[21]; Tandon 用 PSO 训练控制铣削过程的神经网络^[22]; 文献 [23] 中用改进的速度更新方程训练模糊神经网络。Parsopoulos 等将 PSO 用于解决多目标优化问题、最小最大化问题、整数规划问题和定位所有全局极值等问题。

一般说来, PSO 比较有潜力的应用包括系统设计、多目标优化、分类、模式识别、调度、信号处理、决策、机器人应用等。其中具体应用实例有: 模糊控制器设计、车间作业调度、机器人实时路径规划、自动目标检测、时频分析等^[2]。

5 仿真

为了检验算法的性能, 笔者按照图 2 的流程框图编写了仿真程序, 对两例优化问题进行求解。

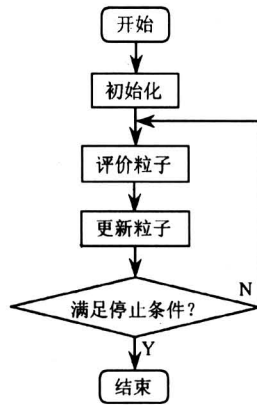


图 2 PSO 流程图

Fig.2 PSO flow chart

例 1 混合整数优化问题^[24]

$$\min_{x,y} f(x,y) = 0.6224(0.0625y_1)x_1x_2 + 1.7781(0.0625y_2)x_1^2 + 3.1661(0.0625y_1)^2x_2 + 19.84(0.0625y_1)^2x_1,$$

$$\text{s. t. } g_1(x,y) = 0.0193x_1 - 0.0625y_1 \leq 0,$$

$$g_2(x,y) = 0.00954x_1 - 0.0625y_2 \leq 0,$$

$$g_3(x,y) = 750 \times 1728 - \pi x_1^2x_2 - 4\pi x_1^3/3 \leq 0,$$

$$g_4(x,y) = x_2 - 240 \leq 0.$$

表 1 是各种算法对例 1 的仿真计算结果。采用线性下降惯性权重从 0.9 到 0.4 的 PSO 和压缩因子 $c_1 = c_2 = 2.05$ 的 PSO 两种方法, 得到了同样好的结果, 惯性权重法得到该解的概率更高。

表 1 仿真例 1 的各种算法计算结果比较

Table 1 Comparison among the results of various algorithms in example 1

项	IDCNLP	SA	MVEP	MIHDE	空间收缩	PSO
x_1	48.380 7	58.290 0	51.195 8	48.576 5	42.006 0	42.098 44
x_2	111.744 9	43.693 0	90.782 1	110.055 9	183.047 3	176.636 66
y_1	18	18	16	15	13	13
y_2	10	10	10	8	7	7
g_1	-0.191 3	-0.025 0	-0.011 9	0.000 0	-0.001 8	-6.790×10^{-8}
g_2	-0.163 4	-0.068 9	-0.136 6	-0.036 6	-0.036 8	-0.035 88
g_3	-75.8750	6.5496	-13584.5631	0.0000	-29169.0801	-0.07965
g_4	-128.255 1	-196.307 0	-147.217 9	-129.944 1	-56.952 7	-63.363 34
f	8048.6190	7197.7	7108.6160	6370.7035	6193.7788	6059.71533

例 2 Schaffe 提出的实例^[25]

$$\min f(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, |x_i| \leq 100.$$

已知全局最优点是 (0, 0), $f(x)$ 为 -0.5。由于其强烈的振荡性以及全局最优点被无限个局部最优点包围的特性, 一般优化算法很难找到它的全局最优值。PSO 算法的运行结果如表 2。两种方法都只需要迭代 200 次, 即可找到以上最优值, 且耗时只有十几毫秒。

表 2 仿真例 2 的 PSO 算法计算结果

Table 2 Results of the PSO simulated procedure in example 2

项	惯性权重 0.9~0.4 线性下降的 PSO	使用约束因子的 PSO
x_1	$-3.416\ 583 \times 10^{-5}$	$-5.412\ 430 \times 10^{-5}$
x_2	$-2.828\ 398 \times 10^{-6}$	$-2.138\ 258 \times 10^{-7}$
$f(x)$	-0.5	-0.5

2 个实例的仿真表明, 用基本 PSO 很难得到满意的结果, 而这里采用的两种改进方案都十分简单、有效。

5 结语

粒子群优化算法是一种新兴的有潜力的演化算法, 但还存在一些问题, 在这一点上它与遗传算法很类似。

1) PSO 在实际应用中被证明是有效的, 并没有给出收敛性、收敛速度估计等方面的数学证明。文献 [12, 26] 中对收敛性等做了一些研究, 但是目前其理论和数学基础的研究还很不够。

2) 利用不同问题的特点设计出相应有效的算法, 是非常有意义的工作。因此当前针对具体问题深化研究 PSO 算法是值得提倡的工作。

3) 应该致力于补充和扩展 PSO 与其他算法或技术的结合, 解决 PSO 易陷入局部最优的问题。

目前我国已有学者开始了对 PSO 的研究^[7,27], 希望 PSO 可以为优化研究工作带来更多的新思路。

参考文献

[1] Fukuyama Y. Fundamentals of particle swarm techniques [A]. Lee K Y, El-Sharkawi M A. Modern Heuristic Optimization Techniques With Applications to Power Systems [M]. IEEE Power Engineering Society, 2002. 45~51

[2] Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources [A]. Proceedings of the IEEE Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 2001. 81~86

- [3] van den Bergh F. An analysis of particle swarm optimizers [D]. South Africa: Department of Computer Science, University of Pretoria, 2002
- [4] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm [A]. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics [C]. Piscataway, NJ: IEEE Service Center, 1997. 4104~4109
- [5] Yoshida H, Kawata K, Fukuyama Y, et al. A particle swarm optimization for reactive power and voltage control considering voltage stability [A]. Proceedings of the International Conference on Intelligent System Application to Power System [C]. Rio de Janeiro, Brazil, 1999. 117~121
- [6] Angeline P. Using selection to improve particle swarm optimization [A]. Proceedings of IJCNN'99 [C]. Washington, USA, 1999. 84~89
- [7] 徐海, 刘石, 马勇, 等. 基于改进粒子群游优化的模糊逻辑系统自学习算法 [J]. 计算机工程与应用, 2000, (7): 62~63
- [8] Shi Y, Eberhart R C. A modified particle swarm optimizer [R]. IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, May 1998
- [9] Parsopoulos K E, Plagianakos V P, Magoulas G D, et al. Improving particle swarm optimizer by function "stretching" [A]. Hadjisavvas N, Pardalos P. Advances in Convex Analysis and Global Optimization [M]. The Netherlands: Kluwer Academic Publishers, 2001. 445~457
- [10] Shi Y, Eberhart R C. Empirical study of particle swarm optimization [A]. Proceeding of Congress on Evolutionary Computation [C]. : Piscataway, NJ: IEEE Service Center, 1999. 1945~1949
- [11] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization [A]. Proceedings of the Congress on Evolutionary Computation [C]. Seoul, Korea, 2001
- [12] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization [A]. Proceedings of the Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 1999. 1951~1957
- [13] Løvbjerg M, Rasmussen T K, Krink T. Hybrid particle swarm optimiser with breeding and subpopulations [A]. Proceedings of the Genetic and Evolutionary Computation Conference [C]. San Francisco, USA, July 2001
- [14] Suganthan P N. Particle swarm optimizer with neighbourhood operator [A]. Proceedings of the Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 1999. 1958~1961
- [15] Kennedy J. Small worlds and mega - minds: effects of neighborhood topology on particle swarm performance [A]. Proceedings of IEEE Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 1999. 1931~1938
- [16] Kennedy J. Stereotyping: Improving particle swarm performance with cluster analysis [A]. Proceedings of the Congress on Evolutionary Computing [C]. Piscataway, NJ: IEEE Service Center, 2000. 1507~1512
- [17] Beasley D, Bull D, Martin R. A sequential niche technique for multimodal function optimization [J]. Evolutionary Computation, 1993, 1(2): 101~125
- [18] Parsopoulos K E, Vrahatis M N. Particle swarm optimizer in noisy and continuously changing environments [A]. Hamza M H. Proceeding of the IASTED International Conference on Artificial Intelligence and Soft Computing [C]. Cancun, Mexico: IASTED/ACTA Press, 2001. 289~294
- [19] van den Bergh F, Engelbrecht A P. A new locally convergent particle swarm optimizer [A]. Proceedings of IEEE Conference on Systems, Man, and Cybernetics [C]. Hammamet, Tunisia, 2002. 96~101
- [20] van den Bergh F, Engelbrecht A P. Using cooperative particle swarm optimization to train product unit neural networks [R]. IEEE International Joint Conference on Neural Networks, Washington D C, USA, 2001
- [21] Eberhart R C, Hu X. Human tremor analysis using particle swarm optimization [A]. Proceedings of the IEEE Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Service Center, 1999. 1927~1930
- [22] Tandon V. Closing the gap between CAD/CAM and optimized CNC end milling [D]. Indianapolis: (Master's thesis) Purdue School of Engineering and Technology, Indiana University Purdue University, 2001
- [23] He Z, Wei C, Yang L, et al. Extracting rules from fuzzy neural network by particle swarm optimization [A]. Proceedings of IEEE Congress on Evolutionary Computation [C]. Anchorage, Alaska, USA, 1998. 74~77

- [24] 王 涛, 李歧强. 基于空间收缩的并行演化算法 [J]. 中国工程科学, 2003, 5(3): 57~61
- [25] Schaffer J D, Carnana R A, Eshelman L J. A study of control parameters affecting online performance of genetic algorithm for function optimization [A]. Proc of the 3rd Int'l Conf on Genetic Algorithms [C]. Morgan Kaufmann, Los Altos, 1989. 51~60
- [26] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58~73
- [27] 李爱国, 覃 征, 鲍复民, 等. 粒子群优化算法 [J]. 计算机工程与应用, 2002, 38(21): 1~3

Survey on Particle Swarm Optimization Algorithm

Yang Wei, Li Qiqiang

(School of Control Science and Engineering, Shandong University, Jinan 250061, China)

[Abstract] Particle swarm optimization (PSO) is a new optimization technique originating from artificial life and evolutionary computation. The algorithm completes the optimization through following the personal best solution of each particle and the global best value of the whole swarm. PSO can be implemented with ease and few parameters need to be tuned. It has been successfully applied in many areas. In this paper, the basic principles of PSO are introduced at length, and various improvements and applications of PSO are also presented. Finally, some future research directions about PSO are proposed.

[Key words] swarm intelligence; evolutionary algorithm; particle swarm optimization

(上接第 55 页)

- [4] 鸥进萍, 刘学东, 王光远. 现役结构安全度评估的环境荷载标准研究[J]. 工业建筑, 1995, 25(8): 11~16
- [5] 张俊芝. 服役结构可靠性评估的可变荷载取值研究[J]. 工业建筑, 2000, 30(12): 58~61
- [6] 张建仁, 刘 洋. 结构可靠度理论及其在桥梁工程中的应用[M]. 北京: 人民交通出版社, 2003. 102~110
- [7] 混凝土结构设计规范(GB 50010-2002)[S]. 北京: 建筑工业出版社, 2002
- [8] 牛荻涛, 王庆霖. 服役结构抗力的概率模型及其统计参数[J]. 西安建筑科技大学学报, 1997, 29(4): 355~359
- [9] 李桂青, 李秋胜. 工程结构时变可靠度理论及其应用[M]. 北京: 科学出版社, 2001. 139~142
- [10] Mori Y, Ellingwood R. Time-dependent system reliability analysis adaptive importance sampling[J]. Structural Safety, 1993, 12(1): 59~73

Study of the Variable Load While Estimating Existing Bridge Structure

Suo Qinghui, Qian Yongjiu, Zhang Fang

(Southwest Jiaotong University, Chengdu 610031, China)

[Abstract] On the basis of design load in force, the revised factors of variable load used in estimating existing bridge structures are given according to the theory, which is the probability that the load used exceeds the re-service term should be equal to the probability that the load used in design exceeds the service term. The theory of time-dependent ability is adopted to evaluate the residual service life of structure and a method is adopted to revise the variable load.

[Key words] bridge structure; variable load; equal exceeding probability; time-dependent ability; re-service life