第3章　粒子群优化算法

粒子群优化算法是模拟鸟类觅食行为的群智能优化算法。鸟类在飞行过程中,当一只鸟飞离鸟群而飞向栖息地时,将影响其他鸟也飞向栖息地。鸟类寻找栖息地的过程与对一个特定问题寻找解的过程相似。鸟在搜索空间中以一定的速度飞行,要根据自身的飞行经历和周围同伴的飞行经历比较,模仿其他优秀个体的行为,不断修正速度的大小和方向。鸟在粒子群算法中被视为一个粒子,粒子们追随当前的最优粒子在解空间搜索最优解。本章介绍粒子群优化算法的基本原理、描述、实现步骤、流程,以及粒子群优化算法的特点及其改进。

3.1　粒子群优化算法的提出

粒子群优化(Particle Swarm Optimization,PSO)算法是在1995年由美国社会心理学家Kennedy和电气工程师Eberhart共同提出的,又称为粒群算法、微粒群算法。

最初PSO算法模拟鸟群捕食的群体智能行为,它是以研究连续变量最优化问题为背景提出的。虽然PSO算法是针对连续优化问题而提出的,但通过二进制编码可以得到离散变量的PSO形式。因此,它也可以用于离散系统的组合优化问题求解,如用于求解TSP问题等。PSO还可以用于求解多目标优化、带约束优化、多峰函数优化、聚类、调度与规划、控制器参数优化等问题。

3.2　粒子群优化算法的基本原理

PSO算法的基本思想是利用生物学家Heppner的生物群体模型,模拟鸟类觅食等群体智能行为的进化算法。鸟类在飞行过程中是相互影响的,当一只鸟飞离鸟群而飞向栖息地时,将影响其他鸟也飞向栖息地。鸟类寻找栖息地的过程与对一个特定问题寻找解的过程相似。鸟的个体要与周围同类比较,模仿优秀个体的行为,因此可利用其解决优化问题,而人类的决策过程使用了两种重要的知识:一类是自己的经验;二是他人的经验。这有助于提高决策的科学性。

鸟在飞行过程中要具有个性,鸟不能互相碰撞,又要求鸟的个体要向寻找到好解的其他鸟学习。因此,通过仿真研究鸟类群体行为时,要考虑以下3条基本规则。

(1) 飞离最近的个体,以避免碰撞。

(2) 飞向目标(食物源、栖息地、巢穴等)。

(3) 飞向群体的中心,以避免离群。

PSO 算法模拟鸟类捕食行为。假设一群鸟在只有一块食物的区域内,随机搜索食物。所有鸟都不知道食物的位置,但它们知道当前位置与食物的距离,最为简单而有效的方法是搜寻目前离食物最近的鸟的区域。PSO 算法从这种思想得到启发,将其用于解决优化问题。

设每个优化问题的解是搜索空间中的一只鸟,把鸟视为空间中的一个没有重量和体积的理想化“质点”,称为“粒子”或“微粒”,每个粒子都有一个由被优化函数所决定的适应度值,还有一个速度决定它们的飞行方向和距离。然后粒子通过追随当前的最优粒子在解空间中搜索最优解。

3.3 粒子群优化算法的描述

设 n 维搜索空间中,粒子 i 的当前位置 X_i、当前飞行速度 V_i 及所经历的最好位置 P_i(即具有最好适应度值的位置)分别表示为

$$X_i=(x_{i1},x_{i2},\cdots,x_{in}) \tag{3.1}$$

$$V_i=(v_{i1},v_{i2},\cdots,v_{in}) \tag{3.2}$$

$$P_i=(p_{i1},p_{i2},\cdots,p_{in}) \tag{3.3}$$

对于最小化问题,若 $f(X)$ 为最小化的目标函数,则微粒 i 的当前最好位置由下式确定

$$P_i(t+1)=\begin{cases}P_i(t), & f(X_i(t+1))\geqslant f(P_i(t))\\ X_i(t+1), & f(X_i(t+1))<f(P_i(t))\end{cases} \tag{3.4}$$

设群体中的粒子数为 S,群体中所有粒子所经历过的最好位置为 $P_{\mathrm{g}}(t)$,称为全局最好位置,即

$$\begin{aligned}&f(P_{\mathrm{g}}(t))=\min\{f(P_1(t)),f(P_2(t)),\cdots,f(P_s(t))\}\\&P_{\mathrm{g}}(t)\in\{P_1(t),P_2(t),\cdots,P_s(t)\}\end{aligned} \tag{3.5}$$

基本粒子群算法粒子 i 的进化方程可描述为

$$v_{ij}(t+1)=v_{ij}(t)+C_1r_{1j}(t)(P_{ij}(t)-x_{ij}(t))+C_2r_{2j}(t)(P_{gj}(t)-x_{ij}(t)) \tag{3.6}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \tag{3.7}$$

其中,$v_{ij}(t)$为粒子 i 第 j 维第 t 代的运动速度;C_1、C_2 为加速度常数;r_{1j}、r_{2j} 分别为两个相互独立的随机数;$P_{\mathrm{g}}(t)$为全局最好粒子的位置。

式(3.6)描述了粒子 i 在搜索空间中以一定的速度飞行,这个速度要根据自身的飞行经历(式(3.6)中右第 2 项)和同伴的飞行经历(式(3.6)中右第 3 项)进行动态调整。

PSO 算法中粒子 i 飞行方向的校正示意如图 3.1 所示,图中 $P_i(t)$是粒子 i 当前所处位置,$P_{i\mathrm{b}}(t)$是粒子 i 到目前为止找到的最好位置,$P_{\mathrm{gb}}(t)$是当前种群 $X(t)$到目前为止找到的最好位置;$v_i(t)$是粒子 i 的当前飞行速度。$v_i(t+1)$是粒子 i 的$(t+1)$时刻根据它自身到目前为止找到的最好位置,以及当前种群到目前为止找到的最好位置来调整后的运动速度。

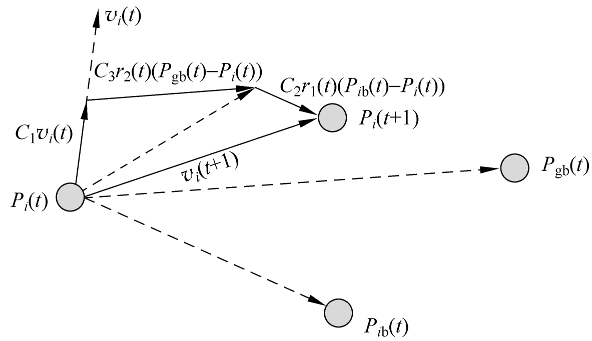

图 3.1　PSO 算法中粒子 i 飞行方向校正图

3.4　粒子群优化算法的实现步骤及流程

在问题求解中，每个粒子以其几何位置与速度向量表示，每个粒子参考自身所经历的最优方向和整个鸟群所公共认识的最优方向来决定自己的飞行方向。

每个粒子 X 可标识为

$$X=\langle p,\boldsymbol{v}\rangle=\langle \text{几何位置,速度向量}\rangle \tag{3.8}$$

PSO 算法的实现步骤如下。

(1) 构造初始粒子群体，随机产生 n 个粒子 $X_i=\langle p_i,v_i\rangle(i=1,2,\cdots,n)$。

$$\begin{aligned}X(0)&=(X_1(0),X_2(0),\cdots,X_n(0))\\&=(\langle p_1(0),v_1(0)\rangle,\langle p_2(0),v_2(0)\rangle,\cdots,\langle p_n(0),v_n(0)\rangle)\end{aligned} \tag{3.9}$$

置 $t:=0$。

(2) 选择。

① 假定以概率 1 选择 $X(t)$每一个体。

② 求出每个粒子 i 到目前为止所找到的最优粒子 $X_{ib}(t)=\langle P_{ib}(t),v_{ib}(t)\rangle$。

③ 求出当前种群 $X(t)$到目前为止所找到的最优粒子 $X_{gb}(t)=\langle P_{gb}(t),v_{gb}(t)\rangle$。

(3) 繁殖，对每个粒子 $X_i(t)=\langle p_i(t),v_i(t)\rangle$，令

$$p_i(t+1)=p_i(t)+\alpha v_i(t+1) \tag{3.10}$$

$$\begin{aligned}v_i(t+1)=&C_1v_i(t)+C_2r_1(0,1)[P_{ib}(t)-P_i(t)]+\\&C_3r_2(0,1)[P_{gb}(t)-P_i(t)]\end{aligned} \tag{3.11}$$

其中，$r_1(0,1)$、$r_2(0,1)$分别为(0，1)中的随机数；C_1 为惯性系数；C_2 为自身认知系数；C_3 为社会学习系数；一般 C_2、C_3 取值为 0～2，C_1 取值为 0～1。

由此形成第 $t+1$ 代粒子群。

$$\begin{aligned}X(t+1)&=(X_1(t+1),X_2(t+1),\cdots,X_n(t+1))\\&=(\langle p_1(t+1),v_1(t+1)\rangle,\langle p_2(t+1),v_2(t+1)\rangle,\cdots,\langle p_n(t+1),v_n(t+1)\rangle)\end{aligned} \tag{3.12}$$

(4) 终止检验，如果 $X(t+1)$已产生满足精度的近似解或达到进化代数要求，则停止计算并输出 $X(t+1)$最佳个体为近似解。

否则置 $t:=t+1$ 转入步骤(2)。

一个基本微粒群算法流程如图 3.2 所示。

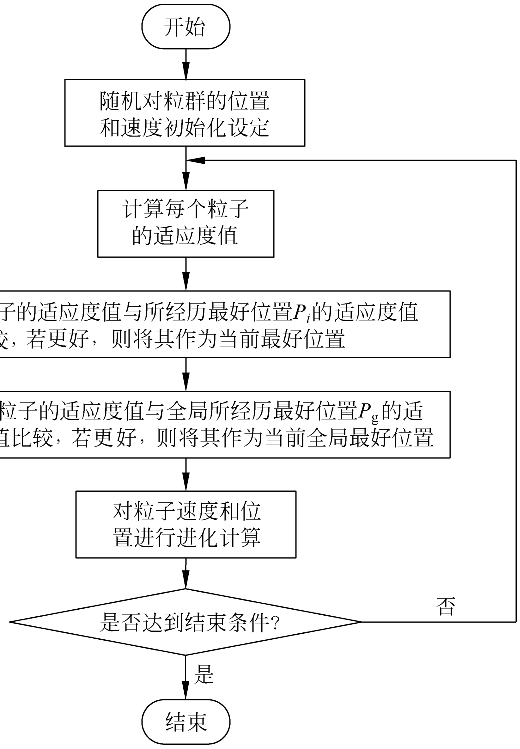

图 3.2　基本微粒群算法流程图

3.5　粒子群优化算法的特点及其改进

PSO 算法具有的特点是：设计模型简单，无需梯度信息，控制参数较少，易于实现，运行速度快；但存在收敛过程易出现停滞及收敛精度较低的缺点。

为了提高基本 PSO 算法的局部搜索能力和全局搜索能力以加快搜索速度，下面提出一些改进方法。

1. 带有惯性因子的 PSO 算法

对于式(3.11)中 $v_i(t)$项前加以惯性权重 ω，一般选取

$$\omega(t)=(0.9\sim 0.5)t/[\text{最大截止代数}] \tag{3.13}$$

此外，对惯性因子可以在线动态调整，如采用模糊逻辑将 $v_i(t)$表示成[低]、[中]、[高] 3 个模糊语言变量，通过模糊推理决定相应的加权大小。

2. 带有收缩因子的 PSO 算法

$$\begin{aligned} v_{ij}(t+1)=\mu[v_{ij}(t)+C_1 r_{1j}(t)[P_{ij}(t)-x_{ij}(t)]+ \\ C_2 r_{2j}(t)[P_{gj}(t)-x_{ij}(t)]] \end{aligned} \tag{3.14}$$

$$\mu=\frac{2}{|\,2-l-\sqrt{l^2-4}\,|} \tag{3.15}$$

其中，μ 为收缩因子；$l=C_1+C_2$，$l>4$。

此外，通过与其他智能优化算法，如遗传算法、差分进化、量子优化等相融合，以及基于动态邻域(小生境)等方法加以改进。