



论文

粒子群优化算法位置更新方程的统一及简化

胡建^{①*}, 胡旺^②, 殷锋^③

① 西南民族大学西南民族研究院, 成都 610041

② 电子科技大学计算机科学与工程学院, 成都 610054

③ 西南民族大学计算机科学与技术学院, 成都 610041

* 通信作者. E-mail: hujian@swun.edu.cn

收稿日期: 2016-08-19; 接受日期: 2016-09-09; 网络出版日期: 2016-11-09

教育部人文社会科学研究青年基金项目 (批准号: 15YJC870010)、四川省科技厅科技支撑项目 (批准号: 2014SZ0104)、四川省教育厅资助科研项目重点项目 (批准号: 16ZA0012) 和西南民族大学研究生学位点建设项目 (批准号: 2016-XWD-B0304) 资助

摘要 粒子群优化算法的理论分析和避免早熟一直是被重点研究的两个问题, 但前者因复杂的动态性而不得不在简化的系统条件下进行, 后者因不可避免地引入形式多样的操作算子而增加了算法复杂性, 进而使得理论分析更加困难. 对此, 本文整理归纳出了大多数现有改进算法的位置更新方程之间的共性规律, 给出了统一形式, 并由多阶随机差分方程简化为一阶随机差分方程, 使得粒子搜索行为控制和收敛性分析更为容易. 实验在具有代表性的算法上进行, 验证了对位置更新方程的统一和简化过程的合理性, 并表明本文方法性能更具有竞争力.

关键词 粒子群优化 群体智能 人工智能 进化计算 收敛性

1 引言

群体智能 (swarm intelligence)^[1] 是人工智能中近年来发展迅速的研究方向, 其受大自然, 尤其受生物系统的启发, 通过构建低层次的、分散的个体之间的简单交互行为而涌现出高层次的、集体的复杂智能行为, 具有高度的自组织、自适应性、非线性和涌现的系统特征, 在没有集中控制且不提供全局模型的前提下较其他方法具有明显的优势, 已被用于解决原来非常棘手的复杂问题, 包括许多 NP 难问题. 粒子群优化 (particle swarm optimization, 简称 PSO) 算法是群体智能领域中最受瞩目的成果之一, 源于对鸟群觅食行为的模拟, 由社会心理学家 Kennedy 和电气工程师 Eberhart 于 1995 年提出^[2], 以形式简单和性能高效的特点, 赢得了研究者的广泛关注, 涉及到数学、计算机科学、通信、自动控制等多个领域. 其中, 有如下两方面问题被重点研究.

一方面问题是, 对 PSO 的理论分析, 如收敛性分析和机理研究. 对此, 自 PSO 被发明之后不久, 就有不少研究者试图用传统数学工具定量分析粒子的运动轨迹以解释和了解粒子的搜索行为, 但是, PSO 因存在很多变化成分以及各成员之间的相互影响而具有复杂的动态性, 使得研究者遇到了困

引用格式: 胡建, 胡旺, 殷锋. 粒子群优化算法位置更新方程的统一及简化. 中国科学: 信息科学, 2016, 46: 1676–1692, doi: 10.1360/N112016-00096

难, 不得不在简化的系统条件下进行理论分析, 如: 不考虑位置更新方程中的随机数^[3,4]; 粒子历史最优位置和群体历史最优位置保持不变^[5~7].

另一方面问题是, PSO 容易陷入局部最优 (或称为早熟), 尤其是对高维多极值的复杂问题. 对此, 已经有大量的研究者提出了成百上千种改进算法, 如改变参数取值^[8,9], 改变拓扑结构^[10], 引入选择、变异和交叉操作算子^[11], 与其他进化计算杂交^[12], 采用概率或物理现象模拟位置更新的方法^[13,14]. 虽然, 其中几乎所有文献都声称其提高了 PSO 的性能, 但是, 他们不可避免地引入了新的数学或逻辑操作算子, 增加了算法复杂性, 这使得对 PSO 的理论分析更加困难.

正如 PSO 发明人之一 Kennedy 所说的“在粒子群研究中, 简约是一种美. 使粒子群回到其本质并以此为出发点, 这也许将有助于我们更好地理解和改进之”^[15], 本文将整理归纳出第二方面问题中已有的大多数改进算法位置更新方程之间的共性规律, 给出其统一形式, 并简化之. 这种统一且简化的形式较这些改进算法不仅具有可媲美的性能表现, 而且把在数学上令人费解的多阶随机差分方程简化为一阶随机差分方程, 从而降低了第一方面问题的解决难度, 更为重要的是, 其清楚地描述了粒子的搜索行为, 可为 PSO 进一步改进提供新的思路.

本文其余部分组织如下: 第 2 节介绍相关工作, 主要涉及到标准 PSO 和已有的改进算法的位置更新方程, 第 3 节给出已有的大多数改进算法位置更新方程的统一形式, 第 4 节对位置更新方程的统一形式进行了简化, 第 5 节是收敛性分析, 第 6 节是实验和讨论, 第 7 节是总结.

2 相关工作

2.1 标准 PSO 的位置更新方程

为了建立一个性能测试的基准, 文献 [16] 把带压缩因子 PSO^[17] 规定为标准 PSO (Standard PSO, 由于其是在 Proceedings of the 2007 IEEE Swarm Intelligence Symposium 上被提出来的, 本文简称之为 SPSO-SIS), 其速度和位置更新规则分别为式 (1) 和 (2):

$$V_{i,d} = \chi \cdot \left[V_{i,d} + \frac{\varphi}{2} \cdot \text{rand}_1 \cdot (P_{i,d} - X_{i,d}) + \frac{\varphi}{2} \cdot \text{rand}_2 \cdot (P_{n,d} - X_{i,d}) \right], \quad (1)$$

$$X_{i,d} = X_{i,d} + V_{i,d}, \quad (2)$$

其中, 下标 i 和 d 分别表示粒子和维度的序号; $V_{i,d}$ 和 $X_{i,d}$ 分别表示粒子 i 在 d 维上的速度和位置; $P_{i,d}$ 表示粒子 i 的历史最优位置的第 d 维; n 表示邻居中最优粒子的序号; $P_{n,d}$ 表示邻居的历史最优位置的第 d 维; χ 是压缩因子, 取近似值 0.72984; φ 是常数, 取值 4.1; rand 是服从均匀分布的范围为 [0,1] 的随机数, 不同的下标表示随机数相互独立. 为了方便起见, 把式 (1) 和 (2) 合并为一个位置更新方程, 并去掉部分下标 i 和 d , 标上循环次数 t , 即有

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \chi \cdot \left\{ \mathbf{V}(t-1) + \frac{\varphi}{2} \cdot \text{rand}_1 \cdot [\mathbf{P}_i - \mathbf{X}(t-1)] + \frac{\varphi}{2} \cdot \text{rand}_2 \cdot [\mathbf{P}_n - \mathbf{X}(t-1)] \right\}. \quad (3)$$

由于 PSO 隐喻着社会心理现象, 研究者们^[18,19] 通常把压缩因子称为惯性因子 (记为 ω), 把式 (3) 展开后的 $\mathbf{P}_i - \mathbf{X}(t-1)$ 的系数 $\chi \cdot \frac{\varphi}{2}$ 称为认知因子 (记为 C_1), $\mathbf{P}_n - \mathbf{X}(t-1)$ 的系数 $\chi \cdot \frac{\varphi}{2}$ 称为社会因子 (记为 C_2), 于是, 式 (3) 可改写为

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \omega \cdot \mathbf{V}(t-1) + C_1 \cdot \text{rand}_1 \cdot [\mathbf{P}_i - \mathbf{X}(t-1)] + C_2 \cdot \text{rand}_2 \cdot [\mathbf{P}_n - \mathbf{X}(t-1)]. \quad (4)$$

基于同样的目的, “Particle Swarm Central”¹⁾ 也在 2006 年, 2007 年和 2011 年分别提供了不同版本的标准 PSO (分别简称为 SPSO-2006, SPSO-2007 和 SPSO-2011), 由于其采用源码公开的形式, 这为研究者们建立了更为直接和具体的测试基准. 这几种标准 PSO 各有优劣, 如 SPSO-SIS 比较简洁, SPSO-2011 虽然在位置更新方程中处理了“偏好”^[20] 问题, 但增加了计算复杂性. 所以, 它们均被广泛地用于 PSO 研究的对比算法中.

2.2 PSO 改进算法的位置更新方程

目前已有成百上千种 PSO 改进算法, 虽然策略多样, 但都是直接或间接地改变粒子的运动规则, 均离不开位置更新方程. 若以是否保留 SPSO-SIS 中的位置更新方程 (3) 或 (4) 的形式为判断标准, 这些改进算法可以分为两类: 第 1 类为保持位置更新方程形式不变的算法, 第 2 类为改变位置更新方程形式的算法.

2.2.1 保持位置更新方程形式不变

这一类算法通过改变位置更新方程中部分项的取值而改变粒子的运动, 该类算法又可分为如下两子类:

一是改变位置更新方程 (4) 的参数 ω , C_1 和 C_2 的取值, 例如: 采用新的参数值, 并在整个计算过程中固定不变, 如 SPSO-2006 和 SPSO-2007; 把参数替换为随机数^[21]; 参数值随进化代数 (或适应度评价次数) 而变化^[22]; 采用确定性判断准则或模糊逻辑, 以适应度或位置等反馈信息为条件, 适应性地调节参数^[8, 23]; 采用优化算法对参数进行动态优化^[9].

二是在位置更新方程 (4) 执行之前或之后, 通过增加其他策略而影响式 (4) 中的 $\mathbf{V}(t-1)$, $\mathbf{P}_i$, $\mathbf{P}_n$, $\mathbf{X}(t)$, $\mathbf{X}(t-1)$, rand_1 和 rand_2 的取值, 例如: 修改通信拓扑, 研究者们早期采用的是固定的单群体拓扑^[24], 后来吸纳了社会群体分工的思想, 提出了多子群的通信拓扑^[25], 并进一步采用了动态拓扑思想, 使群体通信范围和方式动态性地变化^[10]; 采用其他进化计算中的选择、变异和交叉操作算子^[11], 甚至采用其他优化算法对粒子位置进行直接优化^[12]; 增加适应度评价策略^[26]; 改变速度或位置的边界条件^[27]; 位置按维更新^[28].

2.2.2 改变位置更新方程形式

对很多研究者来说, 只改变 SPSO-SIS 位置更新方程中部分项的取值在大多数时候已经不能满足他们的需求, 而且 PSO 模拟了鸟觅食行为, 具有社会心理学渊源和物理运动背景, 这也为他们改变位置更新方程的形式提供了丰富的思想. 该类算法又可分为如下两子类, 其中的符号若未特别说明, 均与 2.1 小节中含义相同.

第 1 子类是, 隐喻社会心理学中的学习思想, 增、减或修改位置更新方程中的部分项. 如: 文献 [29] 提出了 FIPS (fully informed particle swarm) 算法, 认为粒子可以向更多的榜样学习, 位置更新方程为

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \chi \cdot \left\{ \mathbf{V}(t-1) + \left(\sum_{k=1}^N W_k \cdot \varphi_k \cdot \text{rand}_k \right) \cdot [\mathbf{P}_m - \mathbf{X}(t-1)] \right\}, \quad (5)$$

其中, $\mathbf{P}_m = \frac{\sum_{k=1}^N W_k \cdot \varphi_k \cdot \text{rand}_k \cdot \mathbf{P}_k}{\sum_{k=1}^N W_k \cdot \varphi_k \cdot \text{rand}_k}$, N 是当前粒子的所有邻居数, 且需要满足 $\sum_{k=1}^N W_k \cdot \varphi_k = \varphi$, W_k 是其中第 k 个粒子的权重系数.

1) Particle swarm central [2016-03-14]. <http://www.particleswarm.info>.

文献 [30, 31] 分别提出了 CLPSO (comprehensive learning particle swarm optimizer) 和 OLPSO (orthogonal learning particle swarm optimization) 算法, 认为把邻居粒子最优位置和自身历史最优位置按维进行组合, 可以形成更优秀的榜样 (标识为 $\mathbf{P}$), 位置更新方程为

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \omega \cdot \mathbf{V}(t-1) + C \cdot \text{rand} \cdot [\mathbf{P} - \mathbf{X}(t-1)], \quad (6)$$

其中, C 是系数. 在 $\mathbf{P}$ 的确定过程中, CLPSO 采用的是在邻域中随机择优的方式, OLPSO 采用的是正交实验设计的方式.

文献 [32] 提出了 MLPSO (multi-layer particle swarm optimization) 算法, 认为粒子应该向各层次粒子的历史最优学习, 位置更新方程为

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \omega \cdot \mathbf{V}(t-1) + \sum_{j=1}^{m-1} \sum_{k=1}^{N_j} \varphi_{jk} \cdot \text{rand}_{jk} \cdot [\mathbf{P}_{jk} - \mathbf{X}(t-1)], \quad (7)$$

其中, m 为群体的层数, N_j 为在第 j 层次中的榜样数, $\mathbf{P}_{jk}$ 为在第 j 层次中的第 k 个榜样, 且满足 $\varphi_{jk} = \frac{\varphi}{\sum_{j=1}^{m-1} N_j}$.

第 2 子类是, 抛弃或部分抛弃社会心理学中的学习思想, 采用概率或物理现象模拟位置更新的方法. 如: 文献 [33] 提出了 BBPSO (bare bones particle swarm optimization) 算法, 采用概率模拟的方法, 位置更新方程为

$$\mathbf{X}(t) = \text{randn} \left(\frac{\mathbf{P}_i + \mathbf{P}_n}{2}, |\mathbf{P}_i - \mathbf{P}_n| \right), \quad (8)$$

其中, 式子右边表示满足 Gauss 分布的均值为 $\frac{\mathbf{P}_i + \mathbf{P}_n}{2}$ 、标准差为 $|\mathbf{P}_i - \mathbf{P}_n|$ 的随机数.

文献 [13, 14] 提出了 QPSO (quantum-behaved particle swarm optimization), 采用量子物理模型, 位置更新方程为

$$\mathbf{X}(t) = \begin{cases} \mathbf{L} + \alpha \cdot |\mathbf{M} - \mathbf{X}(t-1)| \cdot \ln\left(\frac{1}{u}\right), & k \leq 0.5, \\ \mathbf{L} - \alpha \cdot |\mathbf{M} - \mathbf{X}(t-1)| \cdot \ln\left(\frac{1}{u}\right), & k > 0.5, \end{cases} \quad (9)$$

其中, $\mathbf{L} = \phi \cdot \mathbf{P}_i + (1 - \phi) \cdot \mathbf{P}_g$, g 是具有群体历史最优位置的粒子的序号, $\mathbf{P}_g$ 是群体历史最优位置; α 为系数; $\mathbf{M} = \frac{1}{N} \sum_{j=1}^N \mathbf{P}_j$, N 是群体中粒子个数, $\mathbf{P}_j$ 是第 j 个粒子的历史最优位置; ϕ , k 和 u 表示满足均匀分布、范围为 $[0, 1]$ 的随机数.

SPSO-2011 为了克服原算法存在的坐标“偏好”^[20] 问题, 通过在超球体中选取随机点的方式修改位置更新方程为

$$\mathbf{X}(t) = \omega \cdot \mathbf{V}(t) + \mathbf{G} + \|\mathbf{G} - \mathbf{X}(t-1)\| \cdot \mathbf{R}, \quad (10)$$

其中, $\mathbf{G} = \mathbf{X}(t-1) + \frac{C_1 \cdot [\mathbf{P}_i - \mathbf{X}(t-1)] + C_2 \cdot [\mathbf{P}_n - \mathbf{X}(t-1)]}{3}$, $\|\bullet\|$ 表示 Euclid 范数, $\mathbf{R}$ 表示以坐标原点为中心的单位超球体中的随机点.

3 位置更新方程的统一形式

改进方法的千变万化说明了 PSO 具有丰富的源动力, 但这也给算法的理论分析带来了一定的难度. 因此, 用一个统一形式描述这些算法的位置更新方程将是有益的.

第 1 类算法虽然改变了 SPSO-SIS 位置更新方程中部分项的值, 但保留了位置更新方程的形式, 即式 (4), 展开该式右边, 可得

$$\mathbf{X}(t) = \sum_{j=1,2} a_j \cdot \mathbf{X}(t-j) + \sum_{k=i,n} b_k \cdot \mathbf{P}_k, \quad (11)$$

其中, $a_1 = 1 + \omega - C_1 \cdot \text{rand}_1 - C_2 \cdot \text{rand}_2$, $a_2 = -\omega$, $b_i = C_1 \cdot \text{rand}_1$, $b_n = C_2 \cdot \text{rand}_2$.

第 2 类算法对 SPSO-SIS 的位置更新方程作了不同程度的修改. 从式 (5)~(10) 中可以看出, 大多数改进算法中新位置为 $\mathbf{X}(t-j)$ 和 $\mathbf{P}_k$ 的线性组合, 其中 $\mathbf{X}(t-j)$ 是当前粒子在第 $t-j$ 次循环时的位置, $\mathbf{P}_k$ 为粒子 k 的历史最优位置, 故可以改写为类似于式 (11) 的形式, 即

$$\mathbf{X}(t) = \sum_j a_j \cdot \mathbf{X}(t-j) + \sum_k b_k \cdot \mathbf{P}_k, \quad (12)$$

其中, a_j 和 b_k 分别是 $\mathbf{X}(t-j)$ 和 $\mathbf{P}_k$ 的系数. 例如: FIPS 的位置更新方程式 (5) 可以改写为

$$\mathbf{X}(t) = a_1 \cdot \mathbf{X}(t-1) + a_2 \cdot \mathbf{X}(t-2) + \sum_{k=1}^N b_k \cdot \mathbf{P}_k, \quad (13)$$

其中, $a_1 = 1 + \chi - \chi \cdot (\sum_{i=1}^N W_k \cdot \varphi_k \cdot \text{rand}_k)$, $a_2 = -\chi$, $b_k = -\chi \cdot W_k \cdot \varphi_k \cdot \text{rand}_k$.

BBPSO 的位置更新方程式 (8) 可改写为

$$\mathbf{X}(t) = b_i \cdot \mathbf{P}_i + b_n \cdot \mathbf{P}_n, \quad (14)$$

其中, $b_i = \frac{1}{2} + \text{randn}$, $b_n = \frac{1}{2} - \text{randn}$, randn 表示满足标准 Gauss 分布的随机数.

但是, 第 2 类算法中也有少数因位置更新方程右边不是 $\mathbf{X}(t-j)$ 和 $\mathbf{P}_k$ 的线性组合而无法写成式 (12) 的, 例如 SPSO-2011.

式 (11) 与 (12) 的区别在于后者的 j 和 k 可以取更少或更多的值, 前者是后者的一个特例, 因此本文把式 (12) 作为大多数改进算法的位置更新方程的统一形式. 该统一形式表示了粒子的新位置是该粒子历史位置和榜样历史最优位置的线性组合, 蕴含了众 PSO 的位置更新方程之间的共性规律. 这不仅为变化多样的 PSO 改进算法的理论分析和简化提供了一致的数学基础, 而且有利于促进对算法的改进和对算法本质的研究.

4 位置更新方程的简化

虽然式 (12) 统一了大多数 PSO 改进算法的位置更新方程, 但是等式右边依然存在一些复杂因素 (如“随机性的存在”与“粒子有记忆和智能”所造成的复杂动态性), 对算法的理论分析仍然困难, 因此有必要对式 (12) 进行简化.

从第 1 和 2 类算法的位置更新方程可以看出所有方程的右边系数之和均为 1, 即

$$\sum_j a_j + \sum_k b_k = 1. \quad (15)$$

式 (15) 的成立有如下原因:

如果要使 PSO 具有较好的性能, 必须使得算法在运行末期具有如下条件: 第一, 各粒子的轨迹收敛; 第二, 粒子群聚集; 第三, 各粒子的收敛位置是历史最优的. 前两个条件保证粒子群具有在某处集

中搜索的能力(即开采能力), 第 3 个条件保证粒子群是在历史最优位置开采的. 如果这 3 个条件不能满足, 那么算法就会因无法在历史最优位置处精细搜索而不能获得较好的精度. 从上面 3 个条件容易得到: 从轨迹收敛后的第 j 进化代数起, 总满足

$$\mathbf{X}(t) = \mathbf{X}(t-j) = \mathbf{P}_k = \text{常数}. \quad (16)$$

把式 (16) 带入式 (12) 即可以得到式 (15). 虽然式 (15) 是在粒子轨迹收敛后的情形得出的结果, 但是, 由于 a_j 和 b_k 与粒子的位置无关, 即它们与粒子轨迹是否已经达到收敛状态无关, 因此式 (15) 在算法的整个运行过程中均成立. 相反地, 如果式 (15) 得不到满足, 那么算法的性能就得不到保证, 例如, 文献 [34, 35] 因位置更新方程右边系数之和的绝对值小于 1 而具有原点“偏好”的缺陷.

若令

$$\mathbf{Q} = \frac{\sum_k b_k \cdot \mathbf{P}_k}{\sum_k b_k}, \quad (17)$$

即 $\mathbf{Q}$ 为所有 $\mathbf{P}_k$ 的加权平均, 则式 (12) 可变为

$$\mathbf{X}(t) = \sum_j a_j \cdot \mathbf{X}(t-j) + \left(\sum_k b_k \right) \cdot \mathbf{Q}.$$

根据式 (15), 上式进一步变为

$$\mathbf{X}(t) = \sum_j a_j \cdot \mathbf{X}(t-j) + \left(1 - \sum_j a_j \right) \cdot \mathbf{Q} = \mathbf{Q} + \sum_j a_j \cdot [\mathbf{X}(t-j) - \mathbf{Q}]. \quad (18)$$

上式表明, 我们可以这样得到粒子的新位置 $\mathbf{X}(t)$: 粒子运动到 $\mathbf{Q}$ 处, 然后再从 $\mathbf{Q}$ 处运动 $\sum_j a_j \cdot [\mathbf{X}(t-j) - \mathbf{Q}]$, 其中, $\mathbf{X}(t-j) - \mathbf{Q}$ 表示 $\mathbf{Q}$ 到 $\mathbf{X}(t-j)$ 的矢量.

由于在粒子的搜索过程中, 尤其是在搜索初期, 粒子的行为主要表现为探索, 其位置变化具有较大的随机性, 因此 $\sum_j a_j \cdot [\mathbf{X}(t-j) - \mathbf{Q}]$ 也具有较大的随机性, 其时而为正时而为负, 可以简单地认为, $\sum_j a_j \cdot [\mathbf{X}(t-j) - \mathbf{Q}]$ 就是起一个扰动的作用. 因此, 我们可以假设粒子的新位置是以 $\mathbf{Q}$ 为中心的, 即, 可把式 (18) 改写为

$$\mathbf{X}(t) = \mathbf{Q} + \text{rand}(0), \quad (19)$$

其中, $\text{rand}(0)$ 表示均值为 0 的随机数. 因此, 粒子的搜索行为可以这样描述: 粒子以加权经验 $\mathbf{Q}$ 为中心, 在其附近以一定概率随机搜索. 各 PSO 之所以不同, 就是因为它们具有不同的 $\mathbf{Q}$ 和概率分布. 下面分析如何确定各 PSO 算法的 $\mathbf{Q}$ 和概率分布.

由于式 (17) 是 $\mathbf{Q}$ 的定义, 因此, $\mathbf{Q}$ 可以根据各 PSO 的位置更新方程来确定, 例如: 在 SPSO-SIS 中, 可根据式 (4) 或式 (11) 得到

$$\mathbf{Q} = \frac{C_1 \cdot \text{rand}_1 \cdot \mathbf{P}_i + C_2 \cdot \text{rand}_2 \cdot \mathbf{P}_n}{C_1 \cdot \text{rand}_1 + C_2 \cdot \text{rand}_2};$$

在 FIPS 中, 可根据式 (5) 或式 (13) 得到

$$\mathbf{Q} = \frac{\sum_{k=1}^N \chi \cdot W_k \cdot \varphi_k \cdot \text{rand}_k \cdot \mathbf{P}_k}{\sum_{k=1}^N \chi \cdot W_k \cdot \varphi_k \cdot \text{rand}_k};$$

在 BBPSO 中, 可根据式 (8) 或式 (14) 得到 $\mathbf{Q} = (\frac{1}{2} + \text{randn}) \cdot \mathbf{P}_i + (\frac{1}{2} - \text{randn}) \cdot \mathbf{P}_n$.

下面进一步讨论该概率分布.

由于文献 [33, 36, 37] 表明 Gauss 分布在 PSO 中具有较强的寻优效果, 而且 Gauss 分布也是进化计算中常用的变异方式, 因此, 可以把式 (19) 简化为

$$\mathbf{X}(t) = \mathbf{Q} + \text{randn}(0), \quad (20)$$

其中, $\text{randn}(0)$ 表示满足 Gauss 分布且均值为 0 的随机数.

从式 (18) 的物理意义可以看出, 该 Gauss 分布的标准差与 $\sum_j a_j \cdot [\mathbf{X}(t-j) - \mathbf{Q}]$ 成正相关的关系. 由于在粒子收敛之前, 其所有位置均是随机的, 因此可粗略地认为 Gauss 分布的标准差只与 $\mathbf{Q}$ 到粒子上一次循环的位置 $\mathbf{X}(t-1)$ 之差成正比, 即标准差为 $C \cdot |\mathbf{Q} - \mathbf{X}(t-1)|$, 其中 C 为系数. 故, 式 (20) 可进一步明确为

$$\mathbf{X}(t) = \mathbf{Q} + \text{randn}(0, C \cdot |\mathbf{Q} - \mathbf{X}(t-1)|), \quad (21)$$

其中, $\text{randn}(0, C \cdot |\mathbf{Q} - \mathbf{X}(t-1)|)$ 表示满足 Gauss 分布的均值为 0、标准差为 $C \cdot |\mathbf{Q} - \mathbf{X}(t-1)|$ 的随机数. 由于在部分 PSO 算法中, 若式 (21) 中的 $\mathbf{Q}$ 与 $\mathbf{X}(t-1)$ 相等, 则 $\mathbf{X}(t)$ 将直接取值为 $\mathbf{Q}$, 这会对粒子的搜索行为不利, 例如会使得 SPSO-SIS 中群体最优粒子一直停止于其历史最优位置, 直到有其他粒子更新为群体最优粒子为止, 所以, 为了处理这种不期望的现象, 本文再作如下简单处理: 如果 $\mathbf{Q}$ 与 $\mathbf{X}(t-1)$ 相等, 则在邻域内随机选取一个粒子的位置代入式 (21) 中的 $\mathbf{X}(t-1)$.

可见, 式 (21) 较式 (12): 使粒子的运动行为更为容易理解, 即粒子以加权经验 $\mathbf{Q}$ 为中心、以满足标准差正比于 $C \cdot |\mathbf{Q} - \mathbf{X}(t-1)|$ 的 Gauss 分布进行随机搜索, 通过调节参数 C 可以方便地控制粒子的搜索范围; 把多阶随机差分方程简化为一阶随机差分方程, 使得收敛性分析更为容易.

5 收敛性分析

已有一些文章分析了 SPSO-SIS 的收敛性^[3~7], 但是由于相互作用的粒子众多和随机性强等原因, 这些文章大多数只研究了一个粒子、一维、历史最优位置不变并且没有随机数的简化系统. 而且, 虽然 SPSO-SIS 已经有很多改进版本, 但是几乎所有的改进版本都不可避免地增加了新的策略而使算法更为复杂, 也正是这个原因, 它们大多数没有给出算法的收敛性分析.

本文不仅就大多数 PSO 算法给出了作为进化核心规则的位置更新方程的统一形式, 而且对其进行了简化. 因此, 本文不仅能够在上述简化系统中分析算法的收敛性, 而且将考虑随机性的影响, 即把系统条件扩展为一个粒子、一维、加权经验 $\mathbf{Q}$ 不变和包含随机数. 为了讨论方便, 下面仍将去掉所有的下标, 即粒子位置表示为 $\mathbf{X}(t)$.

设粒子的初始化位置为 $\mathbf{X}(0)$, 由式 (21) 可以递推得到粒子在第 t 代时的位置为

$$\mathbf{X}(t) = \mathbf{Q} - (-1)^t C^t \cdot [\mathbf{Q} - \mathbf{X}(0)] \cdot \prod_{n=1}^t \text{randn}(n), \quad (22)$$

其中, $\text{randn}(n)$, $n = 1, 2, 3, \dots, t$, 表示相互独立的满足标准 Gauss 分布的随机数. 由式 (22) 可知, $\mathbf{X}(t)$ 收敛的充分必要条件为

$$\lim_{t \rightarrow +\infty} (-1)^t C^t \cdot [\mathbf{Q} - \mathbf{X}(0)] \cdot \prod_{n=1}^t \text{randn}(n) = 0,$$

即

$$\lim_{t \rightarrow +\infty} C^t \cdot \prod_{n=1}^t \text{randn}(n) = 0.$$

若记

$$Y = C^t \cdot \prod_{n=1}^t \text{randn}(n),$$

则, Y 的期望值为 $E(Y) = 0$, Y 的方差为

$$D(Y) = E([Y - E(Y)]^2) = E(Y^2) = E\left(C^{2t} \cdot \prod_{n=1}^t \text{randn}(n)^2\right) = C^{2t} \cdot \prod_{n=1}^t E(\text{randn}(n)^2). \quad (23)$$

因 $\text{randn}(n)^2$ 的概率密度函数为 $f(x) = \frac{1}{\sqrt{2\pi}} x^{-\frac{1}{2}} \cdot e^{-\frac{x}{2}}$, 故有

$$E(\text{randn}(n)^2) = \int_0^{+\infty} x \cdot f(x) dx = \int_0^{+\infty} x \cdot \frac{1}{\sqrt{2\pi}} x^{-\frac{1}{2}} \cdot e^{-\frac{x}{2}} dx = 1.$$

把上式代入式 (23), 得到 Y 的方差为 $D(Y) = C^{2t}$. 则, 根据 Chebyshev 不等式有: 对任意一个 $\varepsilon > 0$, 均满足

$$\Pr(|Y| \geq \varepsilon) \leq \frac{C^{2t}}{\varepsilon^2}.$$

因此, 要使 Y 依概率收敛于 0, 需要有 $\lim_{t \rightarrow +\infty} \frac{C^{2t}}{\varepsilon^2} = 0$, 即 $|C| < 1$.

又由于 Y 收敛是 $\mathbf{X}(t)$ 收敛的充分必要条件, 故有, 当 $|C| < 1$ 时, 粒子位置 $\mathbf{X}(t)$ 依概率收敛到稳定点 $\mathbf{Q}$.

结合式 (21), 很容易理解粒子的轨迹为: 当满足 $|C| < 1$ 时, 粒子会以 $\mathbf{Q}$ 为中心振荡, 并且其振幅会越来越小, 也就是粒子在前期会探索一个较大的区域, 然后会慢慢地在 $\mathbf{Q}$ 周围的一个较小的区域开采, 并最终依概率收敛到 $\mathbf{Q}$. 值得注意的是, 由于 Chebyshev 不等式自身是比较粗糙的, 所以 $|C| < 1$ 只是粒子位置 $\mathbf{X}(t)$ 依概率收敛到 $\mathbf{Q}$ 的充分条件, 但这已能够确保采用统一及简化的位置更新方程的算法的收敛性.

上述收敛性分析的目的在于发现式 (21) 中参数 C 的边界, 即如何使得参数 C 保证粒子的轨迹收敛, 而不发散. 这种收敛性分析在部分文献中也被称为稳定性分析^[5]. 除此之外, 局部收敛性和全局收敛性也是算法的一个重要性质, 分别描述了算法收敛到局部最优和全局最优的能力. PSO 算法除了位置更新方程外, 还可以有很多其他策略, 如重新初始化、扰动群体最优粒子等, 由于本文所研究的位置更新方程的统一和简化, 并不涉及和限制这些策略的使用, 所以算法的局部收敛性或全局收敛性的分析需要结合具体算法来进行. 例如, 文献 [38] 提出了一种随机 PSO 算法, 其为了保证全局收敛, 采用了重新初始化群体历史最优粒子的策略, 并基于 Solis 和 Wets 对随机优化算法提出的全局收敛条件^[39]进行了理论证明; 若该随机 PSO 算法采用统一及简化的位置更新方程, 由于其已有的重新初始化群体历史最优粒子的策略并不会受到影响, 所以该策略依然会使得该算法满足文献 [39] 中的全局收敛条件, 确保该算法全局收敛.

6 实验及讨论

本文设计了两个实验, 实验一旨在通过测试统一及简化的位置更新方程与原算法的位置更新方程的相似性来验证本文统一和简化过程的合理性, 实验二旨在通过比较采用统一及简化位置更新方程的算法与原算法的性能表现来验证本文方法的实用性.

6.1 实验一: 测试统一及简化过程的合理性

6.1.1 实验设计

在保持算法其他条件完全一样的条件下, 统一及简化的位置更新方程与原算法的位置更新方程的相似性可以通过粒子群的搜索行为的相似性来准确判断, 但是, 粒子群中复杂的动态性会使得对粒子群搜索行为的相似性的判断非常困难. 因此, 本文将用一种简化的判断方式, 即, 把粒子群的搜索行为近似地表示为粒子群轨迹的收敛过程, 用粒子群轨迹收敛过程的相似性来近似地判断位置更新方程的相似性. 其中, 粒子群轨迹收敛的计算将借鉴文献 [40] 中的方法, 即在进化代数 $t+1$ 时, 其值为

$$\text{Conv}(t) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{X}_i(t) - \mathbf{X}_i(t-1)\|, \quad (24)$$

其中, N 为粒子数, $\mathbf{X}_i(t)$ 和 $\mathbf{X}_i(t-1)$ 分别为粒子 i 在进化代数 t 和 $t-1$ 时的位置, $\|\bullet\|$ 表示 Euclid 范数.

为了较全面地对本文所提出的思想进行验证, 本实验选择了下面 10 种被广泛使用的或较为新颖的算法: (1) Standard PSO from IEEE Swarm Intelligence Symposium (SPSO-SIS) [16], (2) Standard PSO 2007 in Particle Swarm Central (SPSO-2007), (3) Cooperative PSO (CPSO) [28], (4) PSO with general fitness evaluation strategy (PSO-GFES) [26], (5) Fully informed particle swarm (FIPS) [29], (6) Comprehensive learning PSO (CLPSO) [30], (7) Orthogonal learning PSO (OLPSO) [31], (8) Multi-layer particle swarm optimization (MLPSO) [32], (9) Bare bones PSO (BBPSO) [33], (10) Quantum-behaved particle swarm optimization (QPSO) [13, 14]. 其不仅具有代表性, 而且包含了已有的改进算法的全部分类: 算法 (1) 和 (2) 属于保持位置更新方程形式不变类的第 1 子类, 算法 (3) 和 (4) 属于保持位置更新方程形式不变类的第 2 子类, 算法 (5)~(8) 属于改变位置更新方程形式类的第 1 子类, 算法 (9) 和 (10) 属于改变位置更新方程形式类的第 2 子类.

为了测试的公正性, 所有算法的参数设置均同对应的参考文献, 除了需要进一步明确的如下设置外: FIPS 的粒子数依据文献 [16] 设置为 50; MLPSO 和 QPSO 的速度限制设置为不超过 $1/2$ 空间可行域; SPSO-SIS, PSO-GFES, FIPS, CLPSO, OLPSO, MLPSO, BBPSO 和 QPSO 允许粒子越界, 但按维进行越界判断, 若越界则反复更新该维, 直到该维不越界为止; 因对位置越界的考虑而一般无法预设进化代数, OLPSO 和 QPSO 中的原有按进化代数递减的参数修改为按适应度评价次数递减; CPSO 选取文献 [28] 中表现最好的 CPSO-H; PSO-GFES 选取文献 [26] 中表现最好的 PSO-P5 算法; FIPS 选取文献 [29] 中获得最高成功率的采用 U-Ring 拓扑的 FIPS; OLPSO 选取文献 [31] 中表现最好的采用 Ring 拓扑的 OLPSO-L; MLPSO 选取文献 [32] 中表现最好的采用 U-Ring 拓扑的 MLPSO-6L; BBPSO 选取文献 [33] 中表现最好的具有交互概率和随机邻居的算法; QPSO 依据文献 [14] 采用线性递减的收缩膨胀系数; 采用 Matlab 双精度浮点数运算, 计算结果若小于 $1\text{E}-16$, 均按 “0” 处理; 依据文献 [16], 算法停止条件设置为适应度评价 300000 次, 最终实验结果采用独立运行 30 次后的平均值.

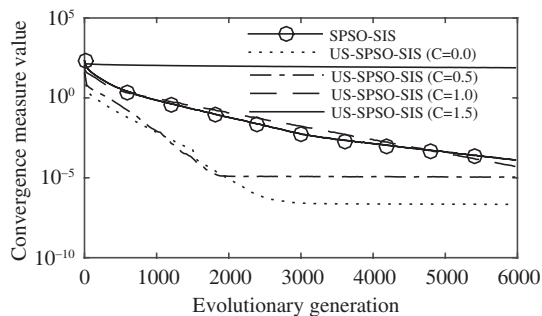
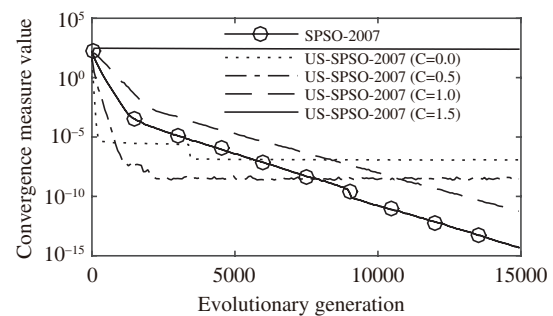
上述算法各自对应着式 (21) 的一个具体形式. 为了方便起见, 本文在上述算法的缩写符号前加上前缀 “US-” (即 Unified and Simplified 的缩写) 用以表示上述算法在采用式 (21) 后的算法, 如 US-SPSO-SIS 表示采用式 (21) 的 SPSO-SIS. 为了充分观察式 (21) 中参数 C 的影响, 且考虑到参数 C 的符号对式 (21) 的结果无影响, 本实验对参数 C 的如下 4 种取值进行了测试: 0.0, 0.5, 1.0 和 1.5. 为了测试的公平性, 所有这些算法的其他参数设置均同对应原算法的设置.

实验采用 10 个常用于优化算法比较的基准函数 [16, 30], 其中有 2 个单峰函数、4 个多峰函数和 4

表 1 实验中的基准函数

Table 1 Benchmark functions in the experiments

Function	x^*	$f(x^*)$	Feasible bound	Initialization range
Sphere (unimodal): $f_1(x) = \sum_{i=1}^D x_i^2$	0^D	0	$[-100, 100]^D$	$[50, 100]^D$
Rosenbrock's (unimodal): $f_2(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	1^D	0	$[-30, 30]^D$	$[15, 30]^D$
Ackley's (multimodal): $f_3(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D X_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	0^D	0	$[-32, 32]^D$	$[16, 32]^D$
Griewank's (multimodal): $f_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	0^D	0	$[-600, 600]^D$	$[300, 600]^D$
Rastrigin's (multimodal): $f_5(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	0^D	0	$[-5.12, 5.12]^D$	$[2.56, 5.12]^D$
Schwefel's (multimodal): $f_6(x) = 418.9829D - \sum_{i=1}^D x_i \sin(x_i ^{\frac{1}{2}})$	0^D	0	$[-500, 500]^D$	$[-500, -250]^D$
f_7, f_8, f_9 and f_{10} : f_7, f_8 and f_9 are f_3, f_4 and f_5 under coordinate rotation using Salomon's algorithm ^[37] , respectively; f_{10} is f_6 rotated by the methods in Ref. [30].				

图 1 SPSO-SIS 与 US-SPSO-SIS 轨迹收敛过程
Figure 1 Convergence in SPSO-SIS and US-SPSO-SIS图 2 SPSO-2007 与 US-SPSO-2007 轨迹收敛过程
Figure 2 Convergence in SPSO-2007 and US-SPSO-2007

个进行坐标旋转后的函数, 函数的名称、形式、理论全局最优位置 x^* 、理论极值 $f(x^*)$ 、搜索范围以及粒子位置的初始化范围见表 1, 其中的 D 表示基准函数的维数, 在实验中取值 30.

6.1.2 实验结果及讨论

实验结果如图 1~10 所示, 其中, 为了方便显示, 各算法的粒子群轨迹收敛过程在所有基准函数上的测试结果按如下计算方式被综合在了一张图上: $MConv = \{\prod_{f=1}^F [\frac{1}{R} \sum_{r=1}^R Conv_{f,r}(t)]\}^{\frac{1}{F}}$, 其中, $Conv_{f,r}(t)$ 表示按式 (24) 计算得到的在第 r 次独立测试第 f 个基准函数时的第 t 进化代数时的粒子群轨迹收敛值, R 为独立测试总次数 (即 30), F 为基准函数总数 (即 10). 下面总结该实验结果所显示的几个现象, 并逐一讨论之.

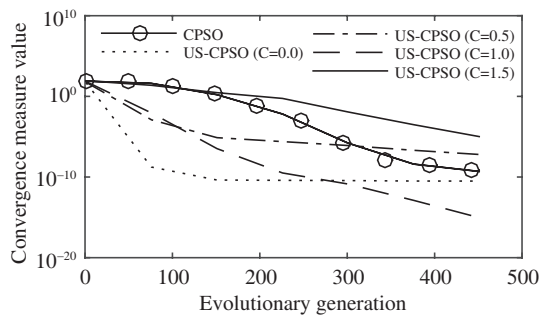


图 3 CPSO 与 US-CPSO 轨迹收敛过程
Figure 3 Convergence in CPSO and US-CPSO

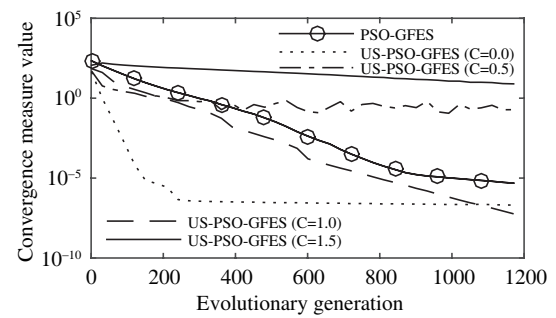


图 4 PSO-GFES与 US-PSO-GFES 轨迹收敛过程
Figure 4 Convergence in PSO-GFES and US-PSO-GFES

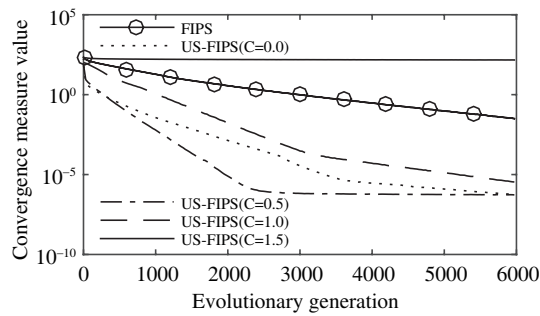


图 5 FIPS 与 US-FIPS 轨迹收敛过程
Figure 5 Convergence in FIPS and US-FIPS

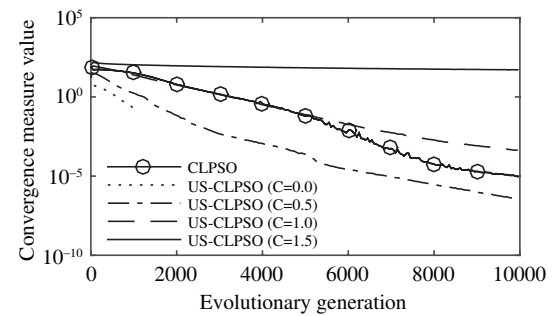


图 6 CLPSO 与 US-CLPSO 轨迹收敛过程
Figure 6 Convergence in CLPSO and US-CLPSO

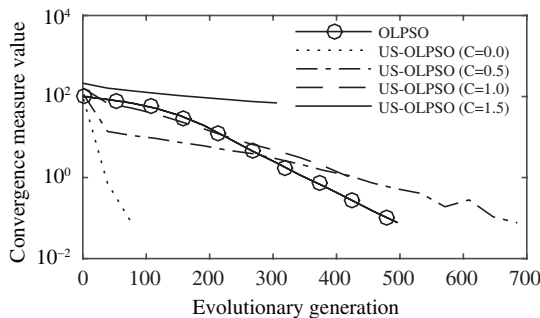


图 7 OLPSO 与 US-OLPSO 轨迹收敛过程
Figure 7 Convergence in OLPSO and US-OLPSO

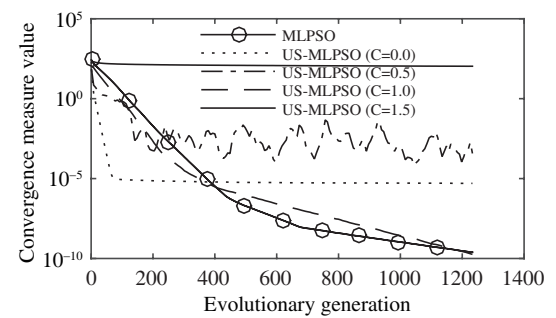


图 8 MLPSO 与 US-MLPSO 轨迹收敛过程
Figure 8 Convergence in MLPSO and US-MLPSO

(1) 在采用统一及简化的位置更新方程的算法中, 当 C 为 0.0, 0.5 和 1.0 时, 所有算法的粒子群轨迹随进化而收敛, 其收敛速度随着 C 值的增加而减小. 这是因为: 正如第 5 节结论一样, $|C| < 1$ 是粒子位置依概率收敛的充分条件; C 值的大小控制着式 (21) 中的 Gauss 分布的标准差, 其值越大, 粒子的新位置远离 Q 的概率越大.

(2) 有 1 对算法的收敛过程基本相同: BBPSO 与 US-BBPSO ($C = 0.0$). 这是因为当 $C = 0.0$ 时,

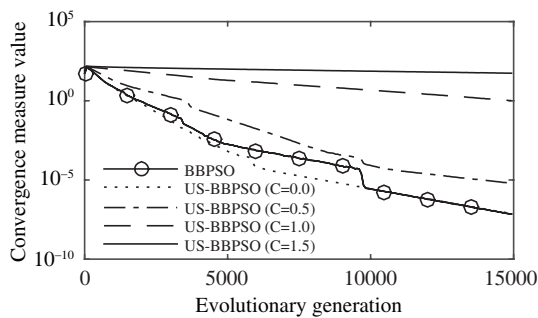


图 9 BBPSO 与 US-BBPSO 轨迹收敛过程
Figure 9 Convergence in BBPSO and US-BBPSO

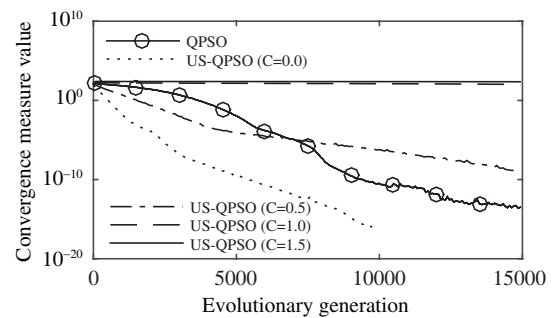


图 10 QPSO 与 US-QPSO 轨迹收敛过程
Figure 10 Convergence in QPSO and US-QPSO

US-BBPSO 与 BBPSO 具有相同的位置更新方程, 即两算法是完全一样的, 图 9 中的收敛过程的细微差异是因为位置更新方程中的随机数在各独立测试中具有不同的值.

(3) 有 5 对算法的收敛过程相似: SPSO-SIS 与 US-SPSO-SIS ($C=1.0$), SPSO-2007 与 US-SPSO-2007 ($C=1.0$), PSO-GFES 与 US-PSO-GFES ($C=1.0$), CLPSO 与 US-CLPSO ($C=1.0$), MLPSO 与 US-MLPSO ($C=1.0$). 这是因为, 在这些算法中, 统一及简化的位置更新方程是由原位置更新方程推理而得, 二者使粒子具有类似的位置更新行为.

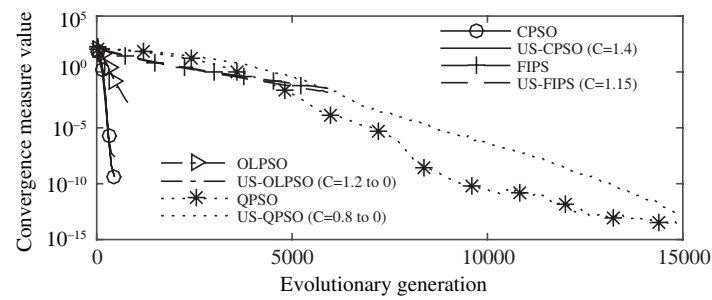
(4) 有 4 对算法的收敛过程有较大差异, 但原算法仍介于两不同 C 值的统一及简化位置更新方程的算法之间: CPSO 介于 US-CPSO 的 $C=1.0$ 时与 $C=1.5$ 时之间; FIPS 介于 US-FIPS 的 $C=1.0$ 时与 $C=1.5$ 时之间; OLPSO 介于 US-OLPSO 的 $C=0.0$ 时与 $C=1.5$ 时之间; QPSO 介于 US-QPSO 的 $C=0.0$ 时与 $C=1.0$ 时之间. 这是因为, 不同的位置更新方程应该具有不同的位置分布规律, 而本文方法均采用 Gauss 分布, 这单一的概率分布不可能正确匹配所有原算法, 而且本文简化了该 Gauss 分布的标准差, 这更是加大了本文方法和原算法之间的差异. 但是, 即便如此, 由于原算法的收敛过程介于两不同 C 值的统一及简化位置更新方程的算法之间, 所以可以通过调节 C 值在一定程度上减少各对算法收敛过程的差异, 例如图 11 显示了 CPSO 与 US-CPSO ($C=1.4$)、FIPS 与 US-FIPS ($C=1.15$)、OLPSO 与 US-OLPSO (C 为 1.2 随适应度评价次数线性递减至 0)、QPSO 与 US-QPSO (C 为 0.8 随适应度评价次数线性递减至 0) 相似, 这种收敛的可调节性正是采用统一及简化的位置更新方程 (21) 所带来的益处.

综上, 本文方法可以通过调节 C 值而方便地控制粒子群轨迹的收敛过程, 使之与原算法相似; 同时, 本文方法因采用简化的概率分布而与原算法并不具有完全相同的收敛过程, 但是, 需要注意到, 因为粒子群具有复杂的动态性, 完全去除这种差异是非常困难的, 也是意义不大的. 因此, 本实验验证了本文对位置更新方程的统一和简化过程的合理性, 并表明了其在调节粒子群轨迹收敛上的优势.

6.2 实验二: 性能测试

6.2.1 实验设计

本实验将把式 (21) 中参数 C 设置为较为理想的值, 对采用统一及简化位置更新方程的算法与原算法在性能上进行比较. 本实验仍采用实验一中的算法和基准函数, 并参考实验一的结果, 为了让粒子群在进化初期具有较慢的轨迹收敛性以利于探索, 在后期具有较快的收敛性以利于开采, C 值设置为随适应度评价次数从 C_{Max} 线性递减到 C_{Min} , 其中, C_{Min} 统一设置为 0, C_{Max} 设置为实验一图 1~

图 11 调节 C 值可减少各对算法轨迹收敛过程的差异Figure 11 Reduce the difference of convergence by adjusting the value of C

10 中的收敛值略大于原算法的 C 值, 具体为: US-BBPSO 的 C_{Max} 分别设置为 0.5, US-QPSO 的 C_{Max} 设置为 1.0, 其余算法的 C_{Max} 设置为 1.5.

6.2.2 实验结果及讨论

表 2 显示了各算法在各基准函数上实验结果的收敛精度, 并采用文献 [16] 中建议的 Bonferroni 方法 [41] 就各对算法在所有测试函数上的结果进行了显著性测试, 其显著性水平设置为 0.05, 显著性测试结果表示在 “Significance” 行中, 其中 “1” 和 “-1” 分别表示采用本文方法的结果显著性优于和差于原算法的结果, “0” 表示二者之间不存在显著性差异.

统计表 2 中各对算法的显著性测试结果, 可以得到: US-SPSO-SIS 在 4 个函数上显著地优于 SPSO-SIS; US-SPSO-2007 在 5 个函数上显著地优于 SPSO-2007, 在 1 个函数上显著地差; US-CPSO 在 1 个函数上显著地优于 CPSO, 在 1 个函数上显著地差; US-PSO-GFES 在 1 个函数上显著地优于 PSO-GFES; US-FIPS 在 4 个函数上显著地优于 FIPS, 在 2 个函数上显著地差; US-CLPSO 在 2 个函数上显著地优于 CLPSO, 在 1 个函数上显著地差; US-OLPSO 在 2 个函数上显著地优于 OLPSO; US-MLPSO 在 3 个函数上显著地优于 MLPSO, 在 2 个函数上显著地差; US-BBPSO 在 1 个函数上显著地优于 BBPSO; US-QPSO 与 QPSO 在所有函数上均没有显著性差异. 因此, 在整体上, 本文方法较原算法在性能上有所提高. 原因之一是, 正如实验一所验证的那样, 统一及简化的位置更新方程可以让粒子具有与原位置更新方程相类似的搜索行为, 这为本文方法的性能表现提供了基础. 更为重要的原因是, 本文方法的位置更新方程中只有一个参数 C , 较原算法更为容易控制粒子的搜索行为: C 值随粒子群进化过程从大到小变化, 可使粒子从探索行为逐渐过渡到开采行为, 这既可避免前期的早熟, 又可提高后期的收敛精度.

7 总结

本文整理归纳了在已有的大多数 PSO 改进算法的位置更新方程之间的共性规律, 给出了统一形式, 并对其进行了简化, 认为粒子的搜索行为是: 粒子以加权经验 Q 为中心、以满足标准差正比于 $C \cdot |Q - X(t-1)|$ 的 Gauss 分布进行随机搜索. 这种统一及简化的位置更新方程通过调节参数 C 可以方便地控制粒子的搜索范围, 同时, 把多阶随机差分方程简化为一阶随机差分方程, 使得收敛性分析更为容易. 实验在具有代表性的算法上进行, 结果不仅验证了本文对位置更新方程的统一和简化过程的合理性, 同时也体现了本文方法性能更有竞争力.

表 2 30 次独立实验的平均结果
Table 2 Mean fitness for all problems after 30 trails

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
SPSO-SIS	0.00E+00	9.21E+00	6.99E-15	4.93E-04	5.21E+01	3.68E+03	1.72E-01	4.93E-04	5.85E+01	5.61E+03
SU-SPSO-SIS	0.00E+00	2.49E+01	6.28E-14	5.75E-04	1.27E+01	1.27E+03	4.48E-02	2.47E-04	4.09E+01	3.94E+03
Significance	0	0	0	0	1	1	0	0	1	1
SPSO-2007	0.00E+00	2.60E+00	9.65E-01	1.64E-02	4.61E+01	4.50E+03	1.31E+00	1.77E-02	5.04E+01	6.18E+03
SU-SPSO-2007	0.00E+00	7.93E+01	1.87E+01	5.00E-03	1.10E+01	2.44E+03	5.21E-15	4.51E-03	3.37E+01	3.41E+03
Significance	0	0	-1	0	1	1	1	0	1	1
CPSO	0.00E+00	1.84E+01	4.61E-14	1.67E-02	0.00E+00	3.51E+03	1.96E+01	4.42E-02	2.41E+02	7.82E+03
SU-CPSO	0.00E+00	1.42E+01	2.94E-13	3.58E-02	1.32E+01	3.11E+03	1.97E+01	5.40E-02	2.41E+02	7.08E+03
Significance	0	0	0	0	-1	1	0	0	0	0
PSO-GFES	0.00E+00	3.38E+01	7.34E-15	0.00E+00	6.63E-02	1.59E+03	7.58E-15	0.00E+00	2.99E+01	3.85E+03
SU-PSO-GFES	0.00E+00	5.00E+01	1.09E-14	1.70E-07	6.89E-08	1.70E+02	1.32E+00	7.40E-04	3.33E+01	3.31E+03
Significance	0	0	0	0	0	1	0	0	0	0
FIPS	3.52E-15	2.42E+01	1.11E-08	3.50E-05	5.85E+01	1.40E+03	1.42E-08	2.04E-06	1.18E+02	4.08E+03
SU-FIPS	0.00E+00	2.58E+01	1.69E-13	1.32E-05	4.32E+01	5.61E+03	1.89E-13	1.23E-04	7.66E+01	3.94E+03
Significance	0	-1	1	0	1	-1	1	0	1	0
CLPSO	0.00E+00	7.79E+00	7.70E-15	0.00E+00	0.00E+00	5.49E+02	1.79E-11	4.37E-08	2.92E+01	3.46E+03
SU-CLPSO	0.00E+00	2.30E+01	8.64E-15	0.00E+00	0.00E+00	0.00E+00	9.47E-15	2.16E-09	2.89E+01	2.34E+03
Significance	0	-1	0	0	0	1	0	0	0	1
OLPSO	0.00E+00	7.78E+00	4.72E-11	0.00E+00	1.22E-06	1.63E+03	7.66E-10	1.16E-06	3.45E+01	5.04E+03
SU-OLPSO	0.00E+00	3.76E+00	5.16E-14	0.00E+00	6.63E-02	5.13E+01	1.20E-13	2.24E-08	3.47E+01	3.61E+03
Significance	0	0	0	0	0	1	0	0	0	1
MLPSO	0.00E+00	1.64E-04	3.55E-15	0.00E+00	1.47E+01	3.62E+03	3.55E-15	4.93E-04	1.53E+01	5.06E+03
SU-MLPSO	0.00E+00	2.39E+01	7.82E-15	2.47E-04	1.04E+01	7.45E+03	8.05E-15	7.39E-04	8.76E+00	3.84E+03
Significance	0	-1	0	0	1	-1	0	0	1	1
BBPSO	0.00E+00	3.05E+01	7.11E-15	0.00E+00	1.08E+01	1.12E+03	6.99E-15	1.98E-10	7.63E+01	4.13E+03
SU-BBPSO	0.00E+00	2.99E+01	6.99E-15	0.00E+00	1.12E+01	6.44E+02	7.11E-15	1.32E-09	7.53E+01	3.88E+03
Significance	0	0	0	0	0	1	0	0	0	0
QPSO	0.00E+00	2.76E+01	7.22E-15	2.24E-02	1.83E+01	2.04E+03	7.11E-15	9.60E-03	2.55E+01	3.63E+03
SU-QPSO	0.00E+00	2.89E+01	8.41E-15	1.09E-02	1.92E+01	3.84E+03	7.58E-15	8.36E-03	2.70E+01	3.16E+03
Significance	0	0	0	0	0	0	0	0	0	0

据我们所知, PSO 中位置更新方程的统一及简化在已有文献中尚未研究, 虽然本文提出了一些思路, 但该问题仍然是开放的, 仍然还有很多方面值得研究, 如:

(1) 统一及简化位置更新方程并不影响和限制算法的其他策略的使用, 把其与一些新的智能优化算法中的有效策略 (如文献 [42] 提出的负相关搜索策略) 进行组合研究, 算法有望获得更好的优化性能.

(2) 从式 (21) 的形式, 可以隐约地看见进化策略^[43]与差分进化^[44]的“影子”, 对其进一步研究可能会在 PSO 与其他进化计算之间搭起一座“桥梁”.

参考文献

- 1 Kennedy J, Eberhart R, Shi Y. *Swarm Intelligence*. Singapore: Elsevier (Singapore) Pte Ltd, 2009
- 2 Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Perth, 1995. 1942–1948
- 3 Bergh F V D, Engelbrecht A P. A study of particle swarm optimization particle trajectories. *Inf Sci*, 2006, 8: 937–971
- 4 Cleghorn C W, Engelbrecht A P. A generalized theoretical deterministic particle swarm model. *Swarm Intell*, 2014, 1: 35–59
- 5 Kadirkamanathan V, Selvarajah K, Fleming P J. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans Evol Comput*, 2006, 3: 245–255
- 6 Poli R. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Trans Evol Comput*, 2009, 4: 712–721
- 7 Cleghorn C W, Engelbrecht A. Fully informed particle swarm optimizer: convergence analysis. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Sendai, 2015. 164–170
- 8 Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization. *IEEE Trans Syst Man Cyber*, 2009, 6: 1362–1381
- 9 Hu M, Wu T F, Weir J D. An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Trans Evol Comput*, 2013, 5: 705–720
- 10 Chu X, Hu M, Wu T, et al. AHPS 2: an optimizer using adaptive heterogeneous particle swarms. *Inf Sci*, 2014, 280: 26–52
- 11 Saxena N, Tripathi A, Mishra K K, et al. Dynamic-PSO: an improved particle swarm optimizer. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Sendai, 2015. 212–219
- 12 Mahmoodabadi M J, Mottaghi Z S, Bagheri A. HEPHO: high exploration particle swarm optimization. *Inf Sci*, 2014, 18: 101–111
- 13 Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Portland, 2004. 1571–1580
- 14 Sun J, Fang W, Palade V, et al. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Appl Math Comput*, 2011, 7: 3763–3775
- 15 Kennedy J. In search of the essential particle swarm. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, 2006. 1694–1701
- 16 Bratton D, Kennedy J. Defining a standard for particle swarm optimization. In: *Proceedings of the IEEE Swarm Intelligence Symposium*, Honolulu, 2007. 120–127
- 17 Clerc M, Kennedy J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput*, 2002, 1: 58–73
- 18 Shi Y, Eberhart R. A modified particle swarm optimizer. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, 1998. 69–73
- 19 Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Trondheim, 1999. 1951–1957
- 20 Spears W M, Green D T, Spears D F. Biases in particle swarm optimization. *Int J Swarm Intell Res*, 2010, 2: 34–57
- 21 Hariya Y, Kurihara T, Shindo T, et al. Lévy flight PSO. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Sendai, 2015. 2678–2684
- 22 Oca M A M D, Thomas S, Ken V D E, et al. Incremental social learning in particle swarms. *IEEE Trans Syst Man Cyber*, 2011, 2: 368–384
- 23 Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization. In: *Proceedings of IEEE Congress on Evolutionary Computation*, Korea, 2001. 101–106
- 24 Mendes R. Population topologies and their influence in particle swarm performance. Dissertation for Ph.D. Degree. Portugal: Minho University, 2004
- 25 Yang S, Li C. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments.

- IEEE Trans Evol Comput, 2010, 6: 959–974
- 26 Hu J, Wang Z, Qiao S, et al. The fitness evaluation strategy in particle swarm optimization. Appl Math Comput, 2011, 21: 8655–8670
 - 27 Helwig S, Branke J, Mostaghim S. Experimental analysis of bound handling techniques in particle swarm optimization. IEEE Trans Evol Comput, 2013, 2: 259–271
 - 28 van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization. IEEE Trans Evol Comput, 2004, 3: 225–239
 - 29 Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput, 2004, 3: 204–210
 - 30 Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput, 2006, 3: 281–295
 - 31 Zhan Z H, Zhang J, Li Y, et al. Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput, 2011, 6: 832–847
 - 32 Wang L, Yang B, Chen Y. Improving particle swarm optimization using multi-layer searching strategy. Inf Sci, 2014, 274: 70–94
 - 33 Kennedy J. Bare bones particle swarms. In: Proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, 2003. 80–87
 - 34 Cui Z, Zeng J, Sun G. Predicted particle swarm optimization. In: Proceedings of IEEE International Conference on Cognitive Informatics, Beijing, 2006. 658–661
 - 35 Cai M, Zhang X, Tian G, et al. Particle swarm optimization system algorithm. Commun Comput Inf Sci, 2007, 3: 388–395
 - 36 Kennedy J. In search of the essential particle swarm. In: Proceedings of IEEE Congress on Evolutionary Computation, Vancouver, 2006. 1679–1686
 - 37 Salomon R. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. Biosystems, 1996, 39: 263–278
 - 38 Zeng J C, Cui Z H. A guaranteed global convergence particle swarm optimizer. J Comput Res Dev, 2004, 8: 1333–1338
[曾建潮, 崔志华. 一种保证全局收敛的 PSO 算法. 计算机研究与发展, 2004, 8: 1333–1338]
 - 39 Solis F J, Wets R J B. Minimization by random search techniques. Math Oper Res, 1981, 1: 19–30
 - 40 Cleghorn C W, Engelbrecht A P. Particle swarm convergence: an empirical investigation. In: Proceedings of IEEE Congress on Evolutionary Computation, Beijing, 2014. 2524–2530
 - 41 Jaccard J, Wan C K. LISREL Approaches to Interaction Effects in Multiple Regression. Calif: Sage Publications, 1996
 - 42 Tang K, Yang P, Yao X. Negatively correlated search. IEEE J Sel Areas Commun, 2016, 3: 542–550
 - 43 Beyer H G, Schwefel H P. Evolution strategies—a comprehensive introduction. Natural Comput, 2002, 1: 3–52
 - 44 Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Opt, 1997, 4: 341–359

Unification and simplification for position updating formulas in particle swarm optimization

Jian HU^{1*}, Wang HU² & Feng YIN³

1 Southwest Nationalities Research Academy, Southwest University for Nationalities, Chengdu 610041, China;

2 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China;

3 School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, China

*E-mail: hujian@swun.edu.cn

Abstract Particle swarm optimization (PSO) has been a popular research area in artificial intelligence technology, where the two issues of theoretical analysis and premature convergence have been the focus of attention. However, due to the complex dynamics of a particle swarm, the former was conducted only for simplified systems. The latter has only been addressed by introducing some additional operations, which inevitably increases the complexity of PSO and complicates the theoretical analysis. This paper proposes a unified and simplified rule for position updating in the existing algorithms as an attempt to solve the above-mentioned problems. This rule simplifies the multiple-order stochastic difference equation to a first-order stochastic difference equation, and facilitates the analysis of the convergence and control of the search behavior of particles. Experiments were conducted on some representative algorithms, and the results verified the correctness of the unification and simplification of position updating formulas, which also performed more competitively.

Keywords particle swarm optimization (PSO), swarm intelligence, artificial intelligence, evolutionary computation, convergence



Jian HU was born in 1980. He received the Ph.D. degree in computer-applied technology from Sichuan University, Chengdu, China, in 2009. Currently, he is a senior engineer at the Southwest Nationalities Research Academy at the Southwest University for Nationalities. His current research interests include evolutionary computation, document image retrieval, and data mining.



Wang HU was born in 1974. He received the Ph.D. degree in computer-applied technology from Sichuan University, Chengdu, China, in 2006. He is currently an associate professor at the School of Computer Science and Engineering of the University of Electronic Science and Technology. His current research interests include computational intelligence, business intelligence, and software engineering.



Feng YIN was born in 1974. He received the Ph.D. degree in computer-applied technology from Sichuan University, Chengdu, China, in 2006. Currently, he is a professor at the School of Computer Science and Technology of the Southwest University for Nationalities. His current research interests include data mining, computer networks, and software testing.