

微粒群优化算法的研究

谭伟,李向

(中国地质大学 计算机学院,湖北 武汉 430074)

摘要:粒子群优化(PSO)算法是一种启发式全局优化技术,一种基于群智能的演化计算方法,其源于鸟群群体运动行为的研究,算法通过粒子间的相互作用发现复杂搜索空间中的最优区域。PSO的优势在于简单容易实现而又功能强大。分析了粒子群优化算法的基本原理,给出了多种改进形式以及研究现状和发展情况,并提出了未来可能的研究方向。

关键词:粒子群优化算法;演化算法;优化;群体智能

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2009)03-0087-04

Research Status and Development of Particle Swarm Optimization

TAN Wei, LI Xiang

(Computer College of China University of Geosciences, Wuhan 430074, China)

Abstract: Particle swarm optimization algorithm is a heuristic global optimization technology and also a kind of evolutionary computation technology based on group of intelligence. It comes from the research on the bird flock movement behavior. The algorithm finds optimal regions of complex problem spaces through the pheromone interaction of particles. The predominance of particle swarm optimization is its excellent performance and simple for implementation. Reviews the basic theory, presents some kinds of improved versions of PSO and research situation, and then future research issues are also given.

Key words: particle swarm optimization; evolutionary computation; optimization; swarm intelligence

0 引言

粒子群优化(Particle Swarm Optimization, PSO)是由 Kennedy 和 Eberhart 等于 1995 年提出的一种演化计算技术,其基本思想源于他们早期参照生物学家的群体模型,对鸟类群体行为的规律性研究。PSO 中,每个优化问题的潜在解都是搜索空间中的一只鸟,称之为“粒子”。所有的粒子都有一个由被优化的函数决定的适应值,每个粒子还有一个速度决定它们飞翔的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索。

微粒群算法概念简明,参数设置少,很好地采用简单速度-位移模型,并能根据当前的搜索情况动态调整搜索策略,对解决复杂环境中的优化问题非常有效。由于认识到 PSO 在函数优化等领域所蕴含的广阔的应用前景,在 Kennedy 和 Eberhart 之后很多学者都进行了这方面的研究。目前,已提出了多种 PSO 改进算

法,并且 PSO 已广泛应用于函数优化,神经网络训练,模式分类、模糊系统控制以及其他的应用领域。

1 微粒群优化算法原理

1.1 基本 PSO 算法及其特点

PSO 算法与其他演化算法相似,也是基于群体的,根据对环境的适应度将群体中的个体移动到好的区域,然而它不像其他演化算法那样对个体使用演化算子,而是将每个个体看作 D 维搜索空间中的一个没有体积的微粒(点),在搜索空间中以一定的速度飞行。这个速度根据它本身的飞行经验以及同伴的飞行经验进行动态调整。第 i 个微粒表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$,它经历过的最好位置(有最好的适应值)记为 $P_i = (P_{i1}, P_{i2}, \dots, P_{id})$,也称为 P_{best} 。在群体所有微粒经历过的最好位置的索引号用符号 g 表示,即 P_g ,也称为 g_{best} 。微粒 i 的速度用 $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 表示。对每一代,其第 d 维($1 \leq d \leq D$)根据如下方程变化:

$$v_{id} = \omega v_{id} + c_1 \text{rand}_1() (P_{id} - x_{id}) + c_2 \text{rand}_2() (P_{gd} - x_{id}) \quad (1)$$

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

其中: ω 为惯性权重, c_1 和 c_2 为加速常数(acceleration

收稿日期:2008-06-16

基金项目:国家自然科学基金资助项目(60643037)

作者简介:谭伟(1982-),男,湖北宜昌人,硕士研究生,主要研究方向为智能计算及其应用;李向,副教授,博士,主要研究方向为智能计算、计算机仿真等。

constants), $\text{rand}()$ 和 $\text{rand}()$ 为两个在 $[0, 1]$ 范围内变化的随机函数。

此外,微粒的速度 V_i 被一个最大速度 V_{\max} 所限制。如果当前对微粒的加速导致它在某维的速度 V_{id} 超过该维的最大速度 $V_{\max d}$, 则该维的速度被限制为该维最大速度 $V_{\max d}$ 。

微粒自身位置的调整是通过将当前位置向量与更新后速度向量进行叠加来实现的,该运算是一种数值关系的叠加。微粒的运动速度增量与其历史飞行经验和群体飞行经验相关,并受最大飞行速度的限制。因此,这样的运动模式可被用于各类寻优问题的求解。

从社会学的角度来看,式(1)中的第 1 部分为微粒先前的速度乘一个权值进行加速,表示微粒对当前自身运动状态的信任,依据自身的速度进行惯性运动,因此称这个权值为“惯性权重”;第 2 部分(微粒当前位置与自身最优位置之间的距离)为“认知(cognition)”部分,表示微粒本身的思考,即微粒的运动来源于自己经验的部分,它使微粒有了足够强的全局搜索能力,避免局部极小;第 3 部分(微粒当前位置与群体最优位置之间的距离)为“社会(social)”部分,表示微粒间的信息共享与相互合作,即微粒的运动来源于群体中其它微粒经验的部分,通过认知模仿较好同伴的运动^[1]。在这三部分的共同作用下,微粒根据历史经验并利用信息共享机制,不断调整自己的位置,以期望找到问题的最优解。

1.2 算法流程

基本 PSO 算法流程如下:

(1) 初始化所有微粒(群体规模为 N), 在允许范围内随机设置微粒的初始位置和速度,并将各微粒的 P_{id} 设为初始位置,取 P_{gd} 为 P_{id} 中的最优值。

(2) 评价每个微粒的适应值,即分别计算每个微粒的目标函数值。

(3) 对于每个微粒,将其适应值与所经历过的最好位置 P_{id} 的适应值进行比较,若较好,则将其作为当前的最优位置。

(4) 对于每个微粒,将其适应值与群体所经历过的最好位置 P_{gd} 的适应值进行比较,若较好,则将其作为当前的全局最优位置。

(5) 根据速度和位置更新方程对微粒的速度和位置进行更新。

(6) 如未达到结束条件,通常为足够好的适应值或是达到一个预设的最大迭代代数,则返回第(2)步。

1.3 参数分析

1.3.1 惯性权重 ω

ω 对 PSO 能否收敛起重要作用,它使微粒保持运

动惯性,使其有扩展搜索空间的趋势,有能力探索新的区域。 ω 值大些有利于全局搜索,收敛速度快,但不易得到精确解; ω 值小些有利于局部搜索和得到更为精确的解,但收敛速度慢且有时会陷入局部极值。合适的 ω 值在搜索精度和搜索速度方面起协调作用。

1.3.2 加速常数 c_1 和 c_2

在公式(1)中,若 $c_1 = c_2 = 0$,微粒将一直以当前的速度惯性飞行,直到到达边界为止,由于它只能搜索有限的区域,所以很难找到最好解。若 $c_1 = 0$,则微粒没有认知能力,只有社会部分,在微粒的相互作用下,有能力达到新的搜索空间,此时收敛速度比基本 PSO 快,但对复杂问题,则比基本 PSO 容易陷入局部极值;若 $c_2 = 0$,则微粒之间没有社会信息共享,只有认知部分,此时个体间没有交互,因而得到最优解的机率非常小。

1.3.3 最大速度 V_{\max}

微粒群算法是通过调整每一次迭代时每一个微粒在每一维上移动的距离来进行。速度的改变是随机的,但不希望不受控制的微粒轨道扩展到越来越广阔的空间,并最终达到无穷。所以当最大速度超过 V_{\max} 值时就要用 V_{\max} 替代。对 V_{\max} 值的选择需对问题有一定的先验知识。

2 微粒群算法的发展

微粒群算法自提出 10 多年以来,得到了国际上相关领域众多学者的关注和研究,成为国际进化计算界研究的热点,现在已经应用于许多科学和工程领域。目前,PSO 出现了多种改进算法:一种是直接在微粒群算法上进行改进;另一种是与其它智能优化算法进行有机结合,形成混合微粒群算法。下面将对这两种改进方法分别进行阐述。

2.1 粒子群优化算法的改进

2.1.1 惯性权重的增加与改进

文献[2]研究了惯性因子 ω 对优化性能的影响,发现较大的值有利于跳出局部极小点,而较小的 ω 值有利于算法收敛,因此提出了自适应调整 ω 的策略,即随着迭代的进行,线性地减小 ω 的值。

然而,PSO 搜索过程是一个非线性的复杂过程,让 ω 线性过渡的方法并不能正确地反映真实的搜索过程。因而,文献[3]提出了一种用模糊规则动态调整 ω 的方法,通过对当前最好性能评价(CBPE)和当前惯性权重制定相应的隶属度函数和模糊推理规则,确定惯性权重 ω 的增量。CBPE 测量的是 PSO 找到的最好候选解的性能。由于不同的优化问题有不同的性能评价范围,所以,为了让该模糊系统有广泛的适用性,通

常使用标准化的 CBPE (NCB2PE)。假定优化问题为最小化问题,则:

$$\text{NCBPE} = (\text{CBPE} - \text{CBPEmin}) / (\text{CBPEmax} - \text{CBPEmin}) \quad (3)$$

其中, CBPEmin 为估计的(或实际的)最小值,而 CBPEmax 为非优 CBPE,任何 CBPE 值大于或等于 CBPEmax 的解都是最小化问题所不能接受的解。实验结果表明,与 ω 线性减小的 PSO 相比,模糊自适应 PSO 有类似或更好的结果。

2.1.2 增加收敛因子

文献[4]中介绍了一种带收敛因子的粒子群优化算法,其位置和速度更新公式如下:

$$v_{id} = \chi \{ \omega \times v_{id} + c_1 \times \text{rand}() \times (P_{id} - x_{id}) + c_2 \times \text{rand}() \times (P_{gd} - x_{id}) \} \quad (4)$$

其中, $\chi = 2 / (2 - \varphi - \varphi^2 - 4\varphi)$ 为收敛因子, $\varphi = c_1 + c_2 > 4$ 时,在假设条件下粒子运动轨迹稳定。在此基础上, Bergh^[5]做了进一步分析,研究随机性对粒子轨迹的影响,并在测度空间上对收敛性做了分析。

2.1.3 协同 PSO 算法

协同 PSO 算法的基本思想是:用 N 个相互独立的微粒群分别在 D 维的目标搜索空间中的不同维度方向上进行搜索^[6]。其具体做法是:选定划分因子 N 和微粒群的微粒数 M ,将输入的 D 维向量(微粒的速度和位置向量)划分到 N 个微粒群。前 $D \bmod N$ 个微粒群,其微粒的位置和速度都是 D/N 维的;后 $K(D \bmod N)$ 个微粒群,其微粒的位置和速度向量也是 D/N 维的。在每一步迭代中,这 N 个的微粒群相互独立地进行状态更新,微粒群之间不共享信息。计算适应值时,将每个微粒群中最优微粒的位置向量拼接起来,组成 D 维向量并代入适应函数计算适应值。此算法在迭代初期,适应值下降缓慢(收敛速度缓慢),且其收敛速度与种群所含微粒数目成反比。但由于协同 PSO 算法采用的是局部学习策略,因此比基本 PSO 算法更容易跳出局部极值,从而达到较高的收敛精度。

2.1.4 使用新的位置和速度更新公式

文献[7]提出一种基于 Multi-start PSO 改进的带变异算子的粒子群优化算法,其基本思想是引入变异算子:与遗传算法类似,在子群的历史最优粒子位置 p_l 连续无变化或变化极小时,若粒子群出现较严重聚集情况,则保留历史最优粒子位置 p_l ,将粒子中少部分重新随机初始化,以此增强全局搜索能力,克服收敛到局部最优点的缺点,又不降低收敛速度和搜索精度。

2.2 PSO 混合算法

2.2.1 混合 PSO (HPSO) 算法

Angeline 于 1998 年提出采用进化计算中的选择

操作的改进型 PSO 模型,称为混合 PSO (HPSO)^[8]。在 Angeline 的 HPSO 模型中,将每次迭代产生的新的微粒群根据适应函数进行选择,用适应度高的一半微粒的位置和速度取代适应度低的一半微粒的位置和速度,并保持后者个体极值不变。HPSO 提高了收敛速度并保持了一定的全局收敛能力,在大多数的函数优化结果上比 PSO 更好。不过,它在解决超高维、多局部极值的复杂性优化问题时有些力不从心。

2.2.2 杂交 PSO 算法

借鉴遗传算法的思想, Lovbjerg M^[9]提出了基于杂交的具有繁殖和子群的 PSO 算法。微粒群中的微粒被赋予一个杂交概率,这个杂交概率是用户自己确定的,与微粒的适应值无关。在每次迭代中,依据杂交概率选取指定数量的微粒放入一个池中。池中的微粒随机地两两杂交,产生同样数目的子代微粒,并用子代微粒代替父代微粒,从而保持种群的微粒数目不变。子代微粒的位置由父代微粒的位置决定。对于局部版的 PSO 算法而言,相当于在一个种群中划分了若干个子群。因此,杂交操作既可以在同一子群内部进行,也可以选择在不同的子群之间进行。

实验结果显示, PSO 杂交算法的收敛速度比较快,搜索精度也相对比较高,对一些非线性优化问题可以很好的解决。但因为引用的待调整参数较多,导致仿真的次数增多,工作量加大,所以对使用者的经验有一定要求。

2.2.3 基于模拟退火的 PSO 算法

文献[10]把模拟退火算法思想引入到 PSO 中,将 PSO 算法的全局寻优能力,以及计算速度快、实现简单等优点与模拟退火算法的较强跳出局部最优解能力相结合。从而避免了 PSO 容易陷入局部极值点的缺点,提高了 PSO 进化后期的收敛速度。

2.2.4 免疫 PSO 算法

受生物免疫机制的启发,文献[11]把免疫系统的免疫信息处理机制引入到 PSO 中。此算法结合了 PSO 算法的全局寻优能力和免疫系统的免疫信息处理机制,改善了 PSO 摆脱局部极值点的能力,提高了算法优化过程中的收敛速度和精度。

2.2.5 自适应变异的 PSO (AMPSO)

文献[12]中提出了自适应变异的 PSO 算法,即在进化过程中增加了随机变异算子,变异操作增强了 PSO 跳出局部最优解的能力,从而提高了全局搜索能力,并且能够有效避免早熟收敛的问题。

以上改进算法各有优缺点,它们引入了一些新的参数,在改进算法性能的同时也一定程度上增加了算法的复杂性。

3 PSO 算法的应用

目前, PSO 应用领域主要有: 函数优化、神经网络训练及其它进化算法常用的生产调度等应用领域。

3.1 用于函数优化领域

PSO 最直接的应用是函数优化问题, 包括多元函数优化、带约束优化问题。Angeline 经过大量的实验研究发现^[13], 微粒群优化算法在解决一些典型函数优化问题时, 能够取得比遗传算法 (GA) 更好的优化效果。这就说明微粒群优化算法在解决很多实际问题时具有很好的应用前景, 因为许多实际问题都可以归结为函数优化问题。

3.2 用于神经网络训练

PSO 是一种非常有潜力的神经网络 (NN) 训练算法, 它操作简单, 可以方便地调节神经网络的连接权值和阈值, 已被成功地用来解决许多实际问题。如基于 PSO 的神经网络应用于医疗诊断, 分析人的颤抖; 通过大量的分类问题和函数逼近问题对神经网络进行了详细的性能分析, 证明了使用 PSO 算法训练网络具有较快的速度, 网络达到的精度和泛化能力也优于 BP 算法和 GA 算法。

3.3 用于工程的生产调度领域

一直以来, 生产调度问题是一种未能很好解决的理论难题, 已被证明是属于 NP 完全问题。微粒群算法是一种通过模拟自然界自然现象发展起来的仿生型智能优化算法, 它可以用来求解生产调度问题, 而且经过实验表明已经达到了比较好的效果。

现在已经有文献将 PSO 用于解决资源受限制的工程调度问题, 线性和非线性的二层规划问题, Job-shop 调度问题, 柔性工作车间调度问题, 混流装配线优化调度的问题等等, 并取得较好的效果。

4 PSO 算法的展望

粒子群优化算法是一类新兴的基于群智能的启发式优化技术, 同其它的进化算法相比, 其最具吸引人的特征是简单容易实现和更强的全局优化能力。但对 PSO 研究尚处于初期, 还有许多问题有待解决。如:

(1) 算法理论方面。虽然 PSO 在实际应用中被证明是有效的, 但其理论基础较弱, 目前还没有给出收敛性、收敛速度估计等方面的数学证明。

(2) 粒子群的拓扑结构方面。不同的粒子群邻居的拓扑结构是对不同类型社会的模拟, 如何选择拓扑结构以使 PSO 有最佳性能, 也是进一步研究的问题。

(3) 与其他演化计算的融合方面。如何将其它演化的优点和 PSO 的优点相结合, 构造出有特色有实用价值的混合算法是当前算法改进的一个重要方向。

(4) 算法应用方面。算法的有效性必须在实际应用中才能得到体现。虽然 PSO 算法已得到广泛的应用, 但进一步拓展其应用领域, 对深化研究 PSO 算法也是十分有意义的。

参考文献:

- [1] Kennedy J. Particle swarm: social adaptation of knowledge [C]//Proc IEEE Int Conf on Evolutionary Computation. Indianapolis: IEEE, 1997: 303 - 308.
- [2] Shi Y, Eberhart R C. A Modified Particle Swarm Optimizer [C]//Proc of the IEEE World Congress on Computational Intelligence. Anchorage: IEEE, 1998: 69 - 73.
- [3] Shi Y, Eberhart R C. Fuzzy Adaptive Particle Swarm Optimization [C]//Proc of the IEEE Conf on Evolutionary Computation. Seoul, Piscataway: IEEE Service Center, 2001: 101 - 106.
- [4] Eberhart R C, Shi Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization [C]//Proc of the IEEE Conference on Evolutionary Computation. Piscataway: IEEE Press, 2000: 84 - 88.
- [5] Ven den Bergh F, Engelbrecht A P. Using Neighborhoods with the Guaranteed Convergence PSO [C]//Proc of the Swarm Intelligence Symp. Indianapolis: IEEE Service Center, 2003: 235 - 242.
- [6] Van den Bergh F, Engelbrecht A P. Training Product Unit Networks Using Cooperative Particle Swarm Optimization [C]//Proc of the third Genetic and Evolutionary Computation Conference (GECCO). San Francisco, USA: [s. n.], 2001.
- [7] 李 宁, 孙德宝, 岑翼刚, 等. 带变异算子的粒子群优化算法 [J]. 计算机工程与应用, 2004, 40(17): 12 - 14.
- [8] Angeline P J. Using Selection to Improve Particle Swarm Optimization [C]//IEEE International Conference on Evolutionary Computation. Anchorage, Alaska: [s. n.], 1998.
- [9] Lovbjerg M, Rasmussen T K, Krink T. Hybrid Particle Swarm Optimization With Breeding and Subpopulations [C]//Proc. of the 3rd Genetic and Evolutionary Computation Conference. San Francisco, USA: Morgan Kaufmann Publishers, 2001: 469 - 476.
- [10] 高 鹰, 谢胜利. 基于模拟退火的粒子群优化算法 [J]. 计算机工程与应用, 2004, 40(1): 47 - 50.
- [11] 高 鹰, 谢胜利. 免疫粒子群优化算法 [J]. 计算机工程与应用, 2004, 40(6): 416 - 420.
- [12] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报, 2004, 32(3): 416 - 420.
- [13] Angeline P J. Using Selection to Improve Particle Swarm Optimization [C]//Proceedings of the 1998 International Conference on Evolutionary Computation. New York, NY, USA: IEEE, 1998: 84 - 89.