

带粒子释放和速度限制的粒子群算法

吴正科, 杨青真, 施永强, 李岳锋
(西北工业大学 动力与能源学院, 西安 710072)

摘要: 针对粒子群算法存在的收敛性与多样性之间的矛盾,在惯性权重线性递减的基础上,提出了带粒子释放和速度限制的算法。用粒子释放来增加种群多样性,引导算法跳出局部最优;用速度限制来促使释放后的粒子快速收敛到全局最优,进而平衡粒子的释放速度和收敛速度,保证算法的收敛。通过五个测试函数的实验,证实了粒子释放和速度限制的综合应用显著提高了算法的寻优能力和收敛精度。

关键词: 粒子群优化;线性递减;速度限制;粒子释放

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2013)03-682-02

doi:10.3969/j.issn.1001-3695.2013.03.009

Particle swarm optimization with particle release and speed limit

WU Zheng-ke, YANG Qing-zhen, SHI Yong-qiang, LI Yue-feng
(School of Power & Energy, Northwest Polytechnical University, Xi'an 710072, China)

Abstract: To resolve the conflict between the convergence and the diversity of particle swarm optimization(PSO), this paper proposed an algorithm with particle release and speed limit based on linearly decreasing of the inertia weight. It used particle release to increase swarm diversity, leading the algorithm to leap out of local optimums. It inducted speed limit to make released particles converge to global optimum quickly, and thus to ensure the convergence of the algorithm. The experiments to optimize 5 benchmark functions show that the possibility of finding a best solution is improved by introducing the method of particle release and speed limit, so as to the precision.

Key words: particle swarm optimization(PSO); linearly decreasing; particle release; speed limit

粒子群优化算法(PSO)是通过模拟鸟群觅食行为而发展起来的一种基于群体协作的随机搜索算法,由 Kennedy 等人^[1]于 1995 年提出。由于算法原理简单,所需调节的参数少,且易于程序实现,PSO 算法已经在电力系统优化^[2]、神经网络训练^[3]、函数优化、多目标优化、参数辨识等诸多领域得到了广泛应用。然而粒子群算法在处理复杂的多峰值问题时容易陷入局部最优解,且进化后期收敛速度慢。为此,众多学者针对 PSO 算法提出了改进方案,用于提高算法的性能。总体来说,改进的方法可以分为三类:a)对 PSO 本身的参数因子进行改进,如文献[4~6];b)对粒子邻域拓扑结构的改进,如文献[7~9];c)将其他进化算法与 PSO 相结合,如遗传^[8]、混沌^[10]、禁忌^[11]和量子^[12]等。

总的来说,算法的收敛性与粒子的多样性之间存在着矛盾的关系。因为快速收敛必然导致算法易陷入局部最优,而增加粒子的多样性虽然减少了陷入局部最优的可能,但同时也会降低算法的收敛速度。通常的解决措施是在算法的前期提高粒子的多样性,在算法的后期提高算法的收敛性。

本文提出了一种新的改进思路,即保持算法始终具有很高的收敛性,当出现寻优停滞时,通过粒子释放来引导算法跳出局部最优,同时引入速度限制,保证释放后的粒子能在有限步数内重新收敛到种群最优附近。

1 PSO 算法原理

PSO 从鸟群觅食模型中得到启示,并用于解决优化问题。

粒子群算法寻优的过程是粒子个体根据自己到达过的最佳位置和其他个体到达过的最佳位置来调整自己的速度和位置,从而整体趋向全局最优。

假设在一个 D 维求解空间中,有 N 个粒子组成一个群体,其中第 t 代第 i 个粒子的位置坐标表示为向量:

$$\mathbf{X}_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t, \dots, x_{iD}^t)$$

其速度可以表示为向量:

$$\mathbf{V}_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t, \dots, v_{iD}^t)$$

对第 $t+1$ 代第 i 个粒子的速度和位置的第 d 维,按式(1)和(2)进行迭代更新。

$$v_{id}^{t+1} = wv_{id}^t + c_1r_1(p_{id}^t - x_{id}^t) + c_2r_2(p_{gd}^t - x_{id}^t) \quad (1)$$

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

其中: w 为惯性权重; c_1 和 c_2 为学习因子; r_1 和 r_2 为(0,1)内均匀分布的随机数; p_{id}^t 和 p_{gd}^t 分别表示第 t 代第 d 维的个体最优位置和种群全局最优位置。

2 惯性权重线性递减的 PSO(LDPSO)

在 PSO 的研究中,惯性权重的改进是 PSO 参数因子改进中非常重要的一个分支。惯性权重 w 作为 PSO 算法中一个重要的参数,衡量着前一时刻的速度对下次移动的影响。 w 的取值对于保证收敛和在更大的范围内进行搜索有着重要作用。研究指出,较大的 w 有利于全局搜索,并增加种群的多样性,而较小值的 w 能提高算法的局部开发能力。

收稿日期:2012-06-24;修回日期:2012-08-07

作者简介: 吴正科(1988-),男,湖北人,硕士研究生,主要研究方向为进化算法、航空宇航推进系统优化(wuzk@yahoo.cn);杨青真(1964-),男,教授,博士,主要研究方向为飞行器气动外形设计与优化。

对于惯性权重已有诸多改进,但通常采用线性递减权重策略,即从算法前期的较大值(通常为0.9)线性递减至一个较小值(通常为0.4),以保证算法前期的全局搜索能力和后期的局部搜索能力。递减公式为

$$w^t = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \times t \quad (3)$$

其中: t 为当前迭代数; T 为最大迭代数; $w_{\max}$ 为初始惯性权重; $w_{\min}$ 为进化到最大代数时的惯性权重。

3 带粒子释放和速度限制的粒子群算法(PSO-PR&SL)

为进一步提高 PSO 算法的性能,本文增加了粒子释放和速度限制的策略,以增加粒子的多样性和算法的收敛性。

对粒子进行释放的方法是:当检测到算法的种群最优在一定步数内没有更新时,则认为算法陷入寻优停滞,此时当前最佳粒子的全局搜索能力最差。为跳出停滞状态,算法将当前最佳粒子释放到边界附近,以增加其搜索潜力,即

$$\begin{cases} x_d = u_d + v_d & v_d \leq 0 \\ x_d = l_d + v_d & v_d > 0 \end{cases} \quad (4)$$

其中: x_d 和 v_d 分别为粒子第 d 维的位置和速度; u_d 和 l_d 为第 d 维空间的上、下限。

粒子释放后,如果一定步数内种群最优仍没有得到更新,算法会再次将当前最佳粒子进行释放,继续增加种群多样性。而为保证算法最终能够收敛,必须避免出现粒子的释放速度快于收敛速度的情况,于是算法中又引入了速度限制策略,让释放后的粒子能快速地再次收敛到种群最优附近。

对粒子的速度进行限制时,第 d 维粒子的最大速度为

$$v_d^{\max} = \rho \times (u_d - l_d) \quad \rho \in (0, 1]$$

引入速度限制的目的是:当粒子的当前位置与个体最优位置、种群最优位置之间相差很大时,尤其是在粒子释放后,由式(1)可知再次更新后粒子的速度必然很大,容易出现粒子跳出求解区域的情况,同时过大的速度也会导致粒子难以快速收敛到全局最优附近。而通过速度限制则可以避免此类情况,增加收敛速度,同时也保留了释放粒子的搜索能力,使其被再次释放前能收敛到全局最优附近,并有机会更新种群最优。

PSO-PR&SL 算法步骤如下(其中 S 用来监测是否进行粒子释放):

a) 确定粒子群规模,随机初始化粒子的位置和速度,其中 $x_d \in [l_d, u_d]$, $|v_d| \leq v_d^{\max}$ 。

b) 计算每个粒子的适应度,并更新每个粒子的个体最优和种群全局最优,令 $S=0$ 。

c) 按式(3)计算 w ,按式(1)更新粒子的速度。若 $v_d > v_d^{\max}$,则 $v_d = v_d^{\max}$;若 $v_d < -v_d^{\max}$,则 $v_d = -v_d^{\max}$ 。再按式(2)更新粒子的位置;若 $x_d < l_d$,则 $x_d = l_d$;若 $x_d > u_d$,则 $x_d = u_d$ 。

d) 计算每个粒子的适应度,并更新每个粒子的个体最优和种群全局最优。

e) 若全局最优有更新,则 $S=0$;否则 $S=S+1$ 。

f) 若 $S=m$ 时,令 $S=-n$,并按式(4)释放最优粒子。

g) 若没有到达最大迭代数,返回步骤c);否则,运算停止。

从算法的步骤可以看出,当种群最佳粒子 m 步内没有更新全局最优,则释放最佳粒子,余下的粒子有 $m+n$ 步的机会再次更新全局最优。若 $m+n$ 步内全局最优仍没有更新,则继

续释放当前最佳粒子;若 $m+n$ 步内全局最优有更新,则粒子释放的无更新步数又变为 m 。这样,通过适度改变 m 和 n 的大小,可以在粒子释放速度与收敛速度之间取得平衡,进一步避免因粒子释放过快而难以收敛。

4 测试函数

1) Sphere 函数

$$f(x) = \sum_{i=0}^{d-1} x_i^2$$

其中: $x_i \in [-100, 100]$ 。

2) Rastrigin 函数

$$f(x) = \sum_{i=0}^{d-1} (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

其中: $x_i \in [-5.12, 5.12]$ 。

3) Griewanks 函数

$$f(x) = \sum_{i=0}^{d-1} \frac{x_i^2}{4000} - \prod_{i=0}^{d-1} \cos\left(\frac{x_i}{\sqrt{i+1}}\right) + 1$$

其中: $x_i \in [-600, 600]$ 。

4) Ackley 函数

$$f(x) = 20 - 20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=0}^{d-1} x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=0}^{d-1} \cos(2\pi x_i)\right) + e$$

其中: $x_i \in [-32, 32]$ 。

5) Rosenbrock 函数

$$f(x) = \sum_{i=0}^{d-1} (100(x_{i+1}^2 - x_i)^2 + (1 - x_i)^2)$$

其中: $x_i \in [-100, 100]$ 。

Sphere 函数是简单的单峰值函数;Rastrigin 函数是一个多峰值函数,具有大量的按正弦拐点排列的、很深的局部最优值;Griewanks 和 Ackley 函数都是激烈的多峰值函数;Rosenbrock 函数的解空间有大量狭窄的通道,导致很难获得全局最优值。这些函数的全局最优值都是 $f_{\min}(x) = 0$ 。

5 仿真实验与性能分析

在性能对比的实验中,测试函数的维数为10维,粒子数为14个,最大迭代步数为2000步。LDPSO中 $w_{\max} = 0.9$, $w_{\min} = 0.4$, $c_1 = c_2 = 1.8$ 。LDPSO-PR表示带粒子释放的LDPSO,粒子释放中 $m=5$, $n=15$ 。LDPSO-SL表示带速度限制的LDPSO,速度限制中 $\rho=0.1$ 。PSO-PR&SL表示既有粒子释放又有速度限制的LDPSO。分别对五个测试函数实验50次,测试结果经处理后如表1所示(保留3位有效数字)。

表1 测试结果比较

函数/可接受误差	Sphere/ <1	Rastrigin/ <10	Griewanks/ <1	Ackley/ <1	Rosenbrock/ <100	
LDPSO	最优值	8.10e-72	0.995	3.45e-02	7.69e-15	3.41e-04
	平均值	200	8.44	1.92	2.32	1.81e05
	成功率	98%	84%	98%	86%	46%
LDPSO-PR	最优值	5.03e-67	1.99	1.72e-02	4.14e-15	0.223
	平均值	200	7.20	1.91	1.94	2.01e5
	成功率	98%	90%	98%	90%	54%
LDPSO-SL	最优值	5.95e-73	2.98	3.94e-02	4.14e-15	0.817
	平均值	1.79e-45	8.86	8.82e-02	1.04e-14	2.02e04
	成功率	100%	72%	100%	100%	94%
PSO-PR&SL	最优值	0	0	0	4.14e-15	2.55
	平均值	0	0	0	7.34e-15	3.86
	成功率	100%	100%	100%	100%	100%

(下转第687页)

$O(2^n)$,还简化了搜索过程,蜘蛛死亡后得到的搜索辐即为极小碰集,避免了计算集合是否为集合簇的极小碰集的过程,节约了计算时间。

4 结束语

近年来国内外许多学者对极小碰集的求解方法进行了研究,主要集中在三个方面:a)基于树的数据结构求解;b)应用数学知识进行求解;c)近似求解。基于树的数据结构求解不易编程实现,且时间复杂度与空间复杂度较低。利用数学知识求解,如布尔代数 Boolean 方法,不需要建立树的结构,其空间复杂度和时间复杂度比基于树的方法都有较大降低。近似求解算法,如遗传算法和粒子群算法,以牺牲算法的完备性来换取时间效率,这类方法只能较快地计算出 95% 以上的碰集,而所有的极小碰集在很长的时间内都不能计算得到,不能保证其正确性。

本文提出的蛛网算法——Cobweb 算法,仿照生物学中的蜘蛛捕食,把集合元素之间的相关性作为蜘蛛进行搜索捕食的信息,有选择地生成访问蜘蛛,并按照捕食信息进行衍生和死亡,不但可以在较短的时间内得出全部正确解,且易于编程实现。由图 2 和表 1 可以看出,该方法要比空间复杂度和时间复杂度较好的 Boolean 算法好很多,更进一步说明了该算法的优越性。同时,不用像近似算法那样通过牺牲正确性来提高时间效率。换句话说,该算法在保证正确性的前提下,具有较高的时间效率和较低的空间复杂度,并节约了一定的内存占用。

本文给出的测试集合按照一定规律生成。下一步的工作可以探讨随机生成的集合之间的相关程度对空间以及时间复杂度的影响,同时在实际应用中检验该方法,建立复杂的基于模型诊断的配电网系统,利用该方法求取其极小碰集,以期在较短的时间内得出正确的故障诊断,满足实时性要求。

参考文献:

[1] 欧阳丹彤,欧阳继红,刘大有. 基于模型诊断的研究与新进展[J].

吉林大学自然科学学报,2001(2):38-45.

- [2] REITER R. A theory of diagnosis from first principles[J]. *Artificial Intelligence*,1987,32(1):57-96.
- [3] 李林,卢显良. 一种基于位向量交集运算的规则冲突检测算法[J]. *计算机研究与发展*,2008,45(2):237-245.
- [4] GREINER R. A correction to the algorithm in Reiter's theory of diagnosis[J]. *Artificial Intelligence*,1989,41(1):79-88.
- [5] WOTAWA F. A variant of Reiter's hitting-set algorithm[J]. *Information Processing Letters*,2001,79(1):45-51.
- [6] 姜云飞,林笠. 用对分-HS 树计算最小碰集[J]. *软件学报*,2002,13(12):2267-2274.
- [7] 姜云飞,林笠. 用布尔代数求解最小碰集[J]. *计算机学报*,2003,26(8):920-924.
- [8] 林笠. 基于模型诊断中的用逻辑数组计算最小碰集[J]. *暨南大学学报*,2002,23(1):24-27.
- [9] 黄杰,陈琳,邹鹏. 一种求解极小碰集的遗传模拟退火算法[J]. *软件学报*,2004,15(9):1345-1350.
- [10] LIN Li, JIANG Yun-fei. Computing minimal hitting sets with genetic algorithm[C]//Proc of the 13th International Workshop on Principles of Diagnosis. 2002:77-80.
- [11] 张楠,孙吉贵,赵祥福,等. 求极小碰集的遗传算法[J]. *广西师范大学学报*,2006,24(4):62-65.
- [12] 蒋荣华,天书林,龙兵. 基于 DPSO 最小碰集算法的掩盖故障识别[J]. *系统工程与电子技术*,2009,31(4):997-1001.
- [13] 张立明,欧阳丹彤,曾海林. 基于动态极大度的极小碰集求解方法[J]. *计算机研究与发展*,2011,48(2):209-215.
- [14] 杨新宇,王鹏,曾明. MPLS 自治系统中基于人工蜘蛛的重路由研究[J]. *自然科学进展*,2004,14(7):85-90.
- [15] 蒋亚静,李远杰. 一种基于人工逻辑蛛网的路由算法[J]. *微机发展*,2004,14(11):21-24.
- [16] ZSCHOKKE S. Form and function of the orb-web[M]. Aarhus: Aarhus University Press,2000:99-106.

(上接第 683 页)

从表 1 中可以看出,在只引入粒子释放时,算法的性能并没有显著提高。这是因为 LDPSO 本身已经兼顾了收敛性和多样性,算法收敛速度并不快,出现收敛停滞的时间较晚,所以粒子释放策略作用有限。并且被释放后粒子的位置与个体最优、种群最优之间差距过大,导致粒子速度很大,难以再次快速收敛到种群最优附近,而粒子的释放速度又很快,造成算法的全局最优难以被再次更新。在只引入速度限制时,算法能通过前期的速度限制来加快粒子的收敛速度,但也容易出现因收敛速度过快而使多样性下降,导致算法过早收敛,影响算法的寻优精度。同时引入粒子释放和速度限制时,算法的收敛速度得到很好的保留,在进化后期出现收敛停滞时,算法也能通过粒子释放来增加多样性,并能很好地平衡粒子的释放速度和收敛速度,因而算法的寻优能力和收敛精度都有显著提高。

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of IEEE International Joint Conference on Neural Networks. 1995:1942-1948.
- [2] 林小朗,王磊. 改进粒子群优化算法的电力系统最优潮流计算[J]. *广东电力*,2007,20(3):12-15.
- [3] 唐雪琴,王侃,徐宗昌. 基于 MAPSO 算法的小波神经网络训练方

法研究[J]. *系统仿真学报*,2012,24(3):608-612.

- [4] SHI Yu-hui, EBERHART R. A modified particle swarm optimizer[C]//Proc of IEEE International Conference on Evolutionary Computation. 1998:69-73.
- [5] RATNAWEERA A, HALGAMUGE S, WATSON H. Particle swarm optimization with self-adaptive acceleration coefficients[C]//Proc of the 1st International Conference on Fuzzy Systems and Knowledge Discovery. 2003:246-268.
- [6] 田雨波,朱人杰,薛权祥. 粒子群优化算法中惯性权重的研究进展[J]. *计算机工程与应用*,2008,44(23):39-41.
- [7] 姚灿中,杨建梅. 一种基于有向动态网络拓扑的粒子群优化算法[J]. *计算机工程与应用*,2009,45(27):15-17.
- [8] 陈自郁,何中市,张程. 一种邻居动态调整的粒子群优化算法[J]. *模式识别与人工智能*,2010,23(4):586-592.
- [9] 王维博,冯全源. 基于分层多子群的混沌粒子群优化算法[J]. *控制与决策*,2010,25(11):1663-1668.
- [10] 孙湘,周大为,张希望. 一种混沌粒子群算法[J]. *计算机工程与科学*,2010,32(12):85-88.
- [11] 张玉芳,薛青松,熊忠阳. 基于禁忌搜索的动态粒子群算法[J]. *计算机工程与应用*,2008,44(24):56-58.
- [12] 方伟,孙俊,须文波. 基于微分进化算子的量子粒子群优化算法及应用[J]. *系统仿真学报*,2008,20(24):6740-6744.
- [13] 李季,孙秀霞,李士波,等. 基于遗传交叉因子的改进粒子群优化算法[J]. *计算机工程*,2008,34(2):181-183.