

一种自适应的微粒群算法^{*}

童 璧¹ 吴智铭¹ 童争雄²

(上海交通大学自动化系 CIMS 实验室 上海 200030)¹ (上海行知学院计算机系 上海 200940)²

摘要 研究发现,种群中个体间交换信息的方式对微粒群算法的性能影响很大。我们定义种群拓扑结构(population topology)为种群内部不同个体之间交流信息的网络。不同的种群拓扑结构有着各自的特点,有些利于加速收敛,有些利于扩展搜索空间。在分析种群拓扑结构变化特点的基础上,提出了一种新的自适应的微粒群算法。和通过调节惯性权重的自适应微粒群算法不同,本算法是通过改变种群拓扑结构来达到自适应优化目的的。

关键词 微粒群算法,函数优化,种群拓扑结构

A New Modified Adaptive Particle Swarm Optimization

TONG Yi¹ WU Zhi-Ming¹ TONG Zheng-Xiong²

(Department of Electronic Engineering for Automation, Shanghai Jiaotong University, Shanghai 200030)¹

(Department of Computer Science and Technology, Shanghai Xingzhi College, Shanghai 200940)²

Abstract In our study, the way of communication among individual and its neighbors plays a key role in the performance of particle swarm optimization. Now we give a clear definition for population topology that a communication network among individuals of population. These different population topologies have their own characters, some ones provide a faster search efficiency and the others offer the advantage that subpopulation could search diverse regions of problem spaces. On the basis of these features, we propose a new modified adaptive particle swarm optimization. Different from the traditional APSO, a new method using dynamically changing population topology is presented.

Keywords Particle swarm optimization, Function optimization, Population topology

1 引言

1995年, Kennedy 和 Eberhart 等^[1,2]在对一个简化社会模型模拟的基础上提出了一种新型的演化计算技术。“群”一字来源于 Millonas^[3]在开发人工生命模型时提出的群体智能应有的5个基本原则。而“微粒”一词源于群中的成员,是没有质量、没有体积的抽象粒子,同时这些粒子又具有速度和加速度状态。

经典的粒子流算法与其他的演化算法相似,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。最初,粒子流算法是通过模拟鸟群的捕食行为来达到优化问题的求解。首先在解空间内随机初始化鸟群,鸟群中的每只鸟被称为“粒子”,这些粒子在解空间里以一定的速度飞行,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即时调整。在每一次迭代中,粒子通过跟踪两个“最优值”来更新自己:第一个是粒子本身的最优解(P_{best}),另一个是整个微粒群至今能找到的最优解(G_{best})。找到这两个极值后,每个粒子根据这些参数决定自己的飞行速度、走向和飞行距离。在D维搜索空间中,第*i*个微粒位置表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,它经历过的最好位置(有最好的)记为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$,也称为 P_{best} 。微粒*i*的速度用 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 表示。对于每一代,它*d*维($1 \leq d \leq D$)微粒*i*的速度和位置根据如下方程变化^[4]:

$$v_{id} = \omega v_{id} + c_1 \text{rand}() (p_{id} - x_{id}) + c_2 \text{rang}() (p_{gd} - x_{id}) \quad (1a)$$

$$x_{id} = x_{id} + v_{id} \quad (1b)$$

式中, c_1, c_2 ——加速常数(acceleration constants),取正常数; $\text{rand}()$ —— $[0, 1]$ 范围内变化的随机数; ω ——惯性权重(inertia weight) P_{id} ——粒子本身的最优解(P_{best})在第*d*维下的值; P_{gd} ——整个微粒词至今能找到的最优解(G_{best})在第*d*维上的值。

此外,粒子流飞行速度还有一个最大值 V_{max} ,用于限制粒子的飞行速度。通过调整 ω, V_{max} 可以到达PSO的最佳寻优能力。

2 种群拓扑结构

微粒群算法认为微粒的轨迹由三个部分组成,分别是“惯性”部分、“认知”部分和“社会”部分。“惯性”部分是微粒先前的速度;“认知”部分是本身最优解对它的影响,代表微粒本身的思考;“社会”部分为全局最优对它的影响,代表微粒之间的信息共享与相互合作^[5]。采用何种方法表示“社会”部分的影响?这一问题现已成为当今研究的热点,其中最为著名的两种方法是gbest和lbest(见图1)^[6,7]。在gbest的种群拓扑结构下,微粒受到种群中任何一个其它微粒的影响,由于群中所有的微粒都互相连通,因此(1)式中的全局最优解(G_{best})是由种群中具有最优值的粒子决定,这和经典的粒子群算法相同。在lbest的种群拓扑结构下,微粒只受和它相邻的粒子的影响,这时(1)式中的全局最优解(fitness)是个体的两个邻居中具有最优的个体。典型的情况为每个粒子只有两个相邻粒子,相互之间首尾相连成一个环形。

Kennedy 和 Eberhart 的研究表明,gbest收敛速度较快但是易于陷入局部最优,lbest因为有着子种群的存在,能成功

^{*} 本文得到自然科学基金 65074049 支持。童 璧 硕士研究生。

地越过一些局部最优但是它的收敛却很慢^[8]。Kennedy 和 Mendes 曾做过一系列试验,希望通过测试一些标准测试函数找到一种比较好的种群拓扑结构(population neighborhood topologies)。最后得出的结论是:低连通度的种群通常在多峰函数问题上表现较好,而高连通度的种群则更适合于解决那些单峰函数问题^[9]。

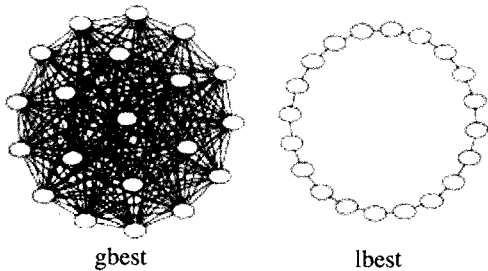


图1 gbest 和 lbest 的种群拓扑结构图

Watts 就网络拓扑结构对信息在网络内的传播的影响作了一些分析^[10,11],这些方法也可以被用于分析 PSO 中的种群拓扑结构。他的评价标准有二:一是网络中所有顶点(vertex)度的平均值 k ;二是网络中所有节点簇(xcluster)的平均值 C 。在粒子流算法中,每个粒子都受它的邻居们(邻居数的大小对应于网络中节点的度)所发现的最优值的影响, k 代表群中粒子在每次迭代过程中交信息的能力。 k 值的大小正是区分 gbest 和 lbest 的标准,同时它也影响着 PSO 的性能。另一个标准是簇,记为 C ,定义如下:如果网络中两个节点互为邻居,那么它们被称为簇。同样地,一个网络的 C 也是由各个节点的平均值计算出来的。在一个网络中,信息从一个节点传递到另一个节点的速度是由两个节点之间的最短距离所决定的,在 PSO 中这等价于两个相邻的粒子要交换信息所要经历的迭代次数,显然这个值是由 k 和 C 这两个参数所决定的。

3 自适应微粒群算法

经典的自适应粒子群算法是 Y Shi 和 Eberhart 在 1998 年提出的,他们在基本 PSO 算法的基础上对参数 w 进行调整,使 w 随着迭代代数的增加逐渐减小^[12]。在 PSO 算法的搜索过程中,随着迭代代数的增加,搜索区域会越来越小,所以自适应的 PSO 算法比基本 PSO 收敛速度更快,结果更好。对于搜索区域的减少是非线性的,表现为数量级上的减少,所以 w 采用一个同样是以数量级减少的分段函数来表述^[13]。如果最大迭代次数为 t ,那么 w 的取值函数如下:

$$w = \begin{cases} 1, & 1 \leq \text{iter} \leq 0.4t \\ 0.1, & 0.4t \leq \text{iter} \leq 0.6t \\ 0.001, & 0.6t \leq \text{iter} \leq t \end{cases} \quad (2)$$

本文提出的自适应粒子群算法与经典方法不同,算法通过改变种群的拓扑结构来达到更加好的搜索能力。在算法初期,因为需要在整个解空间内搜索,所以倾向于使用连通度的拓扑结构;而在搜索的后期,需要集中力量对最优解所在的空间进行深度搜索,所以采用高连通度的拓扑结构。

Kennedy 和 Mendes 测试了一系列的拓扑结构,最后发现“von Neumann”结构在大多数测试函数中都有较好的表现^[8]。“von Neumann”结构因为它的发明人而得名,一个典型的结构如下:种群中的所有粒子呈正方形排列,以一个 5×4 的排列为例,每个粒子都在上下左右与其它的粒子相连,从

图可以看见它们在空间中的结构和平面化的效果(图 2)。定义每个节点的度都为 k 的“von Neumann”结构为 k 阶 von Neumann 结构。当 $k=2$ 时,该结构就成了 lbest;当 $k=D$ 时,该结构就成了 gbest。我们希望随着迭代次数调整 k 的值,来实现自适应地优化过程。

k 的值由如下公式确定:

$$k = 2 + \frac{R(D-2)}{R_{\max}} \quad (3)$$

式中, R —— 当前迭代次数; R_{\max} —— 最大迭代次数。

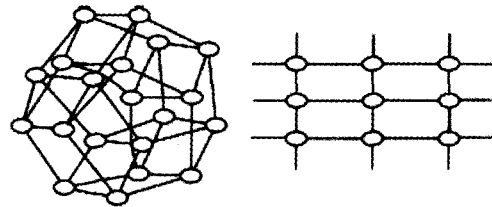


图2 “von Neumann”结构

自适应 PSO 算法的步骤如下:

- (1) 初始化一群粒子(群体大小为 m),包括随机位置和速度;
- (2) 评价每个粒子适应度;
- (3) 对于每个粒子,将其适应值与其自身经历过的最好位置作比较,如果较好,则将其作为最好位置 P_{best} ;
- (4) 对于每个粒子,将其适应值与邻居所经历过的最好位置作比较,如果较好,则将其作为最好位置 P_{best} ;
- (5) 根据(1)式(其中 w 由(2)式决定,随代数数的增加而逐渐减小)改变粒子的速度和位置;
- (6) 根据(3)式刷新 k 的值,改变种群的拓扑结构;
- (7) 如没有达到终止条件(比如为足够好的或是预定的迭代代数),则返回(2)。

4 优化实例

下面给出几个测试函数,它们在文献中常反复被提到。前两个是单峰函数,另外几个为多峰函数,其最优值均为或接近原点^[14]。

球面模型: $f_0(x) = \sum_{i=1}^n x_i^2$ (单峰函数,用于测试优化算法局部寻优能力)

Rosenbrock 函数,亦称为香蕉函数: $f_1(x) = \sum_{i=1}^n (100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2)$ (经典优化函数)

Schaffer 函数: $f_3(x) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$

函数 3: $f_4(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2, -100 < x_i < 100$

函数 4: $f_5(x) = (x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0], -100 < x_i < 100$

所有的函数都求全局最小值,函数 0 到函数 3 取 $n=2$ 。在 MATLAB 上对着六个函数进行优化计算,求解结果见表 1。由表 1 可以看出,改进的优化算法与传统的自适应 PSO 算法相比,在迭代代数相同的情况下,可以得到更加优质的解。

(下转第 252 页)

动性,资源的受限性和高度的动态性等问题。结合基于 Kerberos 认证机制保证了普适计算的安全性。该软件架构在普适计算智能办公系统的原型实现说明了本文提出的普适计算软件体系结构适应普适计算环境的应用需求,也展示了该软件体系结构的可行性,为普适计算软件架构的设计提供了一个参考模型。

参考文献

- 1 Weiser M. The Computer for the Twenty-First Century. Scientific American, 1991, 265(3): 43~50
- 2 Agoston T C, Ueda T, Nishimura Y. Pervasive Computing in a Networked World. In: Global Distributed Intelligence for Everyone, INET2000; 10th Annual Internet Society Conference in Yokohama, Japan, 2000. 213~215
- 3 Banavar G, Beck J, Gluzberg E, et al. Challenges: An Application Model for Pervasive Computing. In: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, 2000. 266~274
- 4 Grimm R, Anderson T, Bershad B, et al. A System Architecture for Pervasive Computing. In: Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, Denmark, 2000. 177~182
- 5 Kindberg T, Fox A. System Software for Ubiquitous Computing Magazine. IEEE Pervasive Computing, 2002, 1(1): 70~81
- 6 Mokhtar S B, Georgantas N, Issarny V, et al. Software Architectures for Pervasive Computing Systems. <http://www.inria.fr/rapportsactivite/RA2005/arles/uid41.html>, updated: 11/24/2005
- 7 Augustin I, Yamin A C, Barbosa J L V, et al. ISAM, A Software Architecture for Adaptive and Distributed Mobile Applications. In: Proceedings of the Seventh International Symposium on Computers and Communications, 2002. 333~338

- 8 Noble B. System Support for Mobile, Adaptive Applications. IEEE Personal Communications (IEEE pers. Commun.), 2000, 7(1): 44~49
- 9 Bolliger J, Gross T. A Framework-based Approach to the Development of Network-Aware Applications. IEEE Transactions on Software Engineering, 1998, 24(5): 376~390
- 10 Chen G, Kotz D. A Survey of Context-Aware Mobile Computing Research. [Technical Report; TR 2000-381]. 2000
- 11 Gray R, Caripe W, Cybenko G. Network-Awareness and Mobile Agent Systems. IEEE Communications Magazine, 1998, 36(7): 44~49
- 12 Kunz T, Black J P. An Architecture for Adaptive Mobile Applications. In: Proceedings of Wireless99, the 11th International Conference on Wireless Communications, Canada, 1999. 27~38
- 13 Wong-Bushby, Egan R I, Isaacson C. A Case Study in SOA and Re-Architecture at Company ABC. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. 8: 4~7
- 14 van Thanh D, Jorstad I. A Service-Oriented Architecture Framework for Mobile Services. In: Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop, Lisbon, Portugal, 2005. 65~70
- 15 Nickull D. Service Oriented Architecture Whitepaper. <http://www.adobe.com/enterprise/pdfs/Services-Oriented-Architecture-from-Adobe.pdf>
- 16 杜攀, 徐进. SOA 体系下细粒度组件服务整合的探讨. 计算机应用, 2006(3): 99~702
- 17 Cervesato, Jaggard A D, Scedrov A, Walstad C. Specifying Kerberos 5 cross-realm authentication. In: Proceedings of the 2005 workshop on Issues in the theory of security (WITS'05), Long Beach, California, 2005. 12~26
- 18 Steiner J G, Neuman B C, Schiller J I. Kerberos: An Authentication Service for Open Network Systems. In: Proceedings of the Winter 1988 Usenix Conference, 1988. 191~201

(上接第 199 页)

表 1 两种自适应 PSO 算法的比较

函数	最优点	最优解	经典自适应 PSO		改进的自适应 PSO	
			最优点	最优解	最优点	最优解
$f_0(x)$	(0, 0, 0)	0	(0.000071, 0.000012)	0.000001	(0.000001, 0.000000)	0.000000
$f_1(x)$	(0, 0, 0)	1	(0.0097, 0.0048)	0.9998	(0.77e-6, -0.12e-6)	1
$f_2(x)$	(0, 0, 0)	1	(-0.0095, 0.0286)	0.9991	(-0.12e-6, 0.17e-6)	1
$f_3(x)$	(-0.0898, 0.7126)	(0.0898, -0.7126)	(0.1120, -0.7150)	-1.029734	(0.0893, -0.7121)	-1.031615
$f_4(x)$	(0, 0, 0)	0	(0.1207e-6, 0.0712e-6)	0.000644	(0.903e-7, 0.324e-8)	0

结论 本文提出的自适应 PSO 算法在传统的自适应 PSO 上更进了一步,在寻优过程中不仅对惯性权重 ω 加以调整,同时又通过调整 k 值改变种群拓扑结构来评价每个粒子。这样使算法在优化过程中不断缩小优化变量的搜索空间,不断加深优化变量的搜索精度,搜索的效率很高,并且能有效地避免陷入局部最优解。

参考文献

- 1 Kennedy J, Eberhart R. Particle swarm optimization[A]. In: IEEE Int Conf. on Neural Networks[C]. Perth, 1995. 1942~1948
- 2 Eberhart R, Kennedy J. A new optimizer using particle swarm theory[A]. In: Proc. 6th Int Symposium on Micro Machine and

- Human Science[C]. Nagoya, 1995. 39~43
- 3 Millonas M M. Swams Phase Transition and Collective Intelligence[M]. MA: Addison Wesley, 1994
- 4 Shi Yuhui, Eberhart R. A modified particle swarm optimizer [A]. In: Proc. IEEE Int Conf. on Evolutionary Computation[C]. Anchorage, 1998. 69~73
- 5 Kennedy J. The particle swarm: Social adaptation of knowledge [A]. In: Proc. IEEE Int Conf. on Evolutionary Computation[C]. Indianapolis, 1997. 303~308
- 6 Kennedy J, Mendes R. Population structure and particle swarm performance [A]. In: 2002 World Congress on Computational Intelligence. Honolulu, HI, May 2002
- 7 Kennedy J, Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms [A]. IEEE Transactions on Systems, Man and Cybernetics, Part c Applications and reviews, 2006, 36(4)
- 8 Kennedy J, Eberhart R. Swarm Intelligence. Morgan Kaufmann Academic Press, 2001
- 9 Kennedy J. Small words and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 Conference on Evolutionary Computation, 1999. 1931~1938
- 10 Watts D J. Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press, 1999
- 11 Watts D J, Strogatz. Collective dynamics of 'small-world' networks. Nature, 1998, 393: 440~442
- 12 Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proc. IEEE World Congress on Computational Intelligence, 1998. 66~73
- 13 Li Junjun, Wang Xihuai. A modified particle swarm optimization algorithm. In: Proceeding of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P R China, June 2004
- 14 谢晓峰, 张文俊, 杨之廉. 微粒群算法综述. 控制与决策, 2003(2): 129~134