

基于免疫进化的粒子群混洗蛙跳算法

李祚泳,张正健,余春雪

(成都信息工程学院 资源环境学院,成都 610041)

(lizuyong@cuit.edu.cn)

摘要:为了避免混洗蛙跳算法易于出现不成熟收敛,提高求解质量,提出了基于免疫进化的粒子群混洗蛙跳算法。该算法将粒子群算法中粒子追踪全局极值的思想融入混洗蛙跳算法中,对族群内的最差个体同时跟踪族群内和全局两个最优个体的信息,进行深度搜索;并引入免疫进化算法对群体中的最优个体进行免疫进化迭代计算,以达到充分利用最优个体的信息的目的。该算法不仅避免了陷入局部极值的局限,以更高的精度逼近全局最优解,而且能加速收敛。对多个典型测试函数的计算表明:基于免疫进化的粒子群混洗蛙跳算法比传统的混洗蛙跳算法具有更好的寻优能力、稳定效果和更快的收敛速度。

关键词:混洗蛙跳算法;免疫进化算法;粒子群算法;函数测试

中图分类号: TP18 **文献标志码:** A

Shuffled frog leaping algorithm based on immune evolutionary particle swarm optimization

LI Zuo-yong, ZHANG Zheng-jian, YU Chun-xue

(College of Resources and Environment, Chengdu University of Information Technology, Chengdu Sichuan 610041, China)

Abstract: A new shuffled frog leaping algorithm based on immune evolutionary particle swarm optimization was proposed in order to avoid premature convergence and to improve the precision of solution by using basic Shuffled Frog Leaping Algorithm (SFLA). The proposed algorithm integrated the global searching idea in the Particle Swarm Optimization (PSO) into SFLA, to pursue the information of two optimal solutions in the sub-swarm and the whole-swarm simultaneously, so as to search thoroughly near by the space gap of the worst solution, and also integrated the immune evolutionary algorithm into SFLA making immune evolutionary iterative computation to the optimal solution in the whole-swarm, so as to use the information of optimal solution fully. This algorithm can not only get free from trapping into local optimum and be close to the global optimal solution with higher precision, but also speeds up the convergence. Calculation results show that the Immune Evolutionary Particle Swarm Optimization-Shuffled Frog Leaping Algorithm (IEPSO-SFLA) has better optimal searching ability and stability as well as faster convergence than those of basic SFLA.

Key words: Shuffled Frog Leaping Algorithm (SFLA); Immune Evolutionary Algorithm (IEA); Particle Swarm Optimization (PSO); function test

传统的优化算法,比如最小二乘法、最速下降法、线性规划、单纯形法和混合型算法等适用于较简单的问题求解,但对于具有多约束的非线性、多极值等复杂问题,不仅求解过程复杂、收敛速度慢,而且对一些问题很难求得全局最优解。近20年来,人们从自然界中一些生物比如蚂蚁、鸟类、鱼群和蜜蜂等群体的社会行为中获得启迪和思路,提出了多种用于求解复杂问题的仿生智能优化算法^[1-9]。混洗蛙跳算法(Shuffled Frog Leaping Algorithm, SFLA)就是一种基于群体智能的具有全局协同的后启发式计算技术。由于它具有原理简单、概念清晰、运算简便、易于实现和全局收敛的特点,因而一经提出,已在若干领域得到应用^[10-11]。但混洗蛙跳算法的迭代过程中,最差个体的空间位置在更新前后发生了较大变化。这种更新策略虽然能扩大解空间的搜索范围,但却易于跳过全局最优解,陷入局部最优,不利于在可行域内有效搜索。因而优化效果不够理想,且收敛速度较慢。

为了加快混洗蛙跳算法的收敛速度和提高求解精度,本

文将免疫进化算法(Immune Evolutionary Algorithm, IEA)^[12]和粒子群算法(Particle Swarm Optimization, PSO)思想引入混洗蛙跳算法中,提出了基于免疫进化的粒子群混洗蛙跳算法(IEPSO-SFLA)。该算法将粒子群算法中的粒子追踪全局极值的思路融入蛙跳算法中,对族群内最差个体,既要通过追寻本族群内的“局域”最优个体,同时亦要追踪全域最优个体来更新自己的位置;此外群体中最优个体也要充分利用自身信息,运用免疫进化迭代算法促使其不断“可持续进化”。这种所谓“劣者追优,优者更优”的更新策略不仅可以避免不成熟收敛,以更高的精度逼近全局最优解,而且能达到快速收敛的目的。该算法还用于多个函数的测试,并与混合蛙跳算法的优化结果进行了比较。

1 混洗蛙跳算法的基本思想及算法实现

由Eusuff等人提出的混洗蛙跳算法(SFLA)是一种基于群体智能的后启发式全新进化算法^[10]。其基本思想为:解群

收稿日期:2011-06-01;修回日期:2011-07-15。 基金项目:国家自然科学基金资助项目(50739002)。

作者简介:李祚泳(1944-),男,四川宜宾人,教授,博士生导师,主要研究方向:优化算法、人工智能、BP神经网络;张正健(1986-),男,重庆璧山人,硕士研究生,主要研究方向:环境评价、优化算法;余春雪(1987-),女,安徽合肥人,硕士研究生,主要研究方向:人工智能、环境信息分析。

体由一群具有相同结构的解(青蛙)构成,整个群体又被分为若干个族群。族群中的个体按照一定策略进行解空间中的局部深度搜索。在规定的局部搜索迭代次数结束后,再将各族群的解进行混合。此局部搜索和混合过程一直进行到满足收敛条件结束为止。全局信息交换和局部深度搜索策略能较有效地避免陷入局部极值,从而实现全局最优。

SFLA 算法实现过程为:随机生成 $P = N \times M$ 只青蛙的初始解群体,第 i 只青蛙代表问题的第 i 个解 $x_i = (x_{i1}, x_{i2}, \dots, x_{is})$, 其中 s 为解空间维数, M 为族群数, N 为族群内青蛙只数。然后计算每个解的适应度值 $f(x_i)$, 将适应度值从大到小排序后的解,依序逐一循环分配给 M 个族群,直到全部 P 个解分配完毕为止。对每个族群中的适应度值最差的解 x_w 按式(1)进行进化迭代计算。

$$x_{w,new}^{(k)} = x_{w,old}^{(k)} + \text{rand}(0,1) \times (x_b^{(k)} - x_{w,old}^{(k)}) \quad (1)$$

式中, $x_{w,new}^{(k)}$ 和 $x_{w,old}^{(k)}$ 分别表示族群 k 中最差解的新、旧值; $x_b^{(k)}$ 表示族群 k 中的最好解; $\text{rand}(0,1)$ 为 $(0,1)$ 内的随机数。 $\text{rand}(0,1) \times (x_b^{(k)} - x_{w,old}^{(k)})$ 表示族群 k 中最差解移动的距离,令 $d(k) = \text{rand}(0,1) \times (x_b^{(k)} - x_{w,old}^{(k)})$ 要求满足 $d_{\max} \geq d(k) \geq -d_{\max}$, d_{\max} 表示允许最差解改变的最大值。再进行判断:若 $f(x_{w,new}^{(k)}) \geq f(x_{w,old}^{(k)})$, 则用新解 $x_{w,new}^{(k)}$ 取代旧解 $x_{w,old}^{(k)}$, 并重新确定 $x_b^{(k)}$ 、 $x_w^{(k)}$ 及 $x_b^{(g)}$, 其中 $x_b^{(g)}$ 为全局最优解;若 $f(x_{w,new}^{(k)}) < f(x_{w,old}^{(k)})$, 则用 $x_b^{(g)}$ 取代 $x_{w,old}^{(k)}$ 重复式(1)的更新策略,若仍无改进,则随机产生一个新解 x_{new}' 取代旧的最差解 $x_{w,old}^{(k)}$ 重复此过程,直到设定的族群 k 内的更新迭代次数 T' 。当所有 M 个族群的局部深度搜索完成后,将所有 N 个族群的 P 个解重新混合,并按适应度值大小重新排序和划分族群,然后再对每个族群进行局部深度搜索。如此反复迭代操作,直至满足问题的终止条件为止。SFLA 的原理及算法描述详见文献[10-11]。

2 免疫进化粒子群混洗蛙跳算法

2.1 免疫进化粒子群混洗蛙跳算法的基本思想

为了使混洗蛙跳算法能最大限度地避免不成熟收敛和具有更佳的优化效果,提出了将免疫进化算法^[12]和粒子群算法的思想引入蛙跳算法中的免疫进化粒子群混洗蛙跳算法(IEPSO-SFLA)。该算法的进化策略为:一方面,从概率意义上讲,群体中的优秀个体与全局最优解之间的空间距离要小于群体中其他个体与全局最优解之间的空间距离;而且与全局最优解空间距离较小的个体也有较高的适应值。因此,为了充分利用群体中最优个体的信息在进化过程中的重要作用,引入免疫进化算法对群体中的最优个体进行如式(2)所示的迭代进化计算:

$$x_{b,new}^{(g)} = x_{b,old}^{(g)} + N(0,1)\sigma_0 e^{-\frac{At}{T}} \quad (2)$$

其中: $x_{b,new}^{(g)}$ 和 $x_{b,old}^{(g)}$ 分别表示群体中最优个体的新、旧值; $N(0,1)$ 为服从标准正态分布的随机数; σ_0 为对应于初始解群体的标准差; A 为标准差动态调整系数; t 为当前进化代数; T 为总进化代数。

另一方面,在蛙跳算法的族群 k 内的最差个体进行解空间的局部深度搜索时,融入 PSO 算法的粒子坐标更新思路,对最差个体 $x_{w,old}^{(k)}$ 不仅要追踪本族群内最优个体 $x_{b,old}^{(k)}$, 同时亦要追踪整个群体中最优个体 $x_{b,new}^{(g)}$ 进行位置更新,即用式(3)替代式(1)对族群内的最差个体进行解空间的既全局又局部的深度搜索。

$$x_{w,new}^{(k)} = x_{w,old}^{(k)} + \text{rand}(0,1) \times (x_b^{(k)} - x_{w,old}^{(k)}) +$$

$$\text{rand}(0,1) \times (x_{b,new}^{(g)} - x_{w,old}^{(k)}) \quad (3)$$

式中, $\text{rand}(0,1) \times (x_b^{(k)} - x_{w,old}^{(k)})$ 即是族群内的最差个体 $x_{w,old}^{(k)}$ 跟踪免疫进化迭代后新的全局最优个体 $x_{b,new}^{(g)}$ 的位置更新。式(2)和式(3)就是 IEPSO-SFLA 算法的核心,它们一起构成了对混洗蛙跳算法的改进。其中式(2)在进化初期,能兼顾群体最优个体附近解空间以外区域的搜索,使群体保持较好的多样性,增强“勘探能力”,可有效避免不成熟收敛;而在进化后期,随着局部搜索能力的不断增强,算法能以更高的精度逼近全局最优解。式(3)则能使族群内最差个体在解空间内进行既全局又局部的深度搜索,增强“开采能力”。

2.2 免疫进化粒子群混洗蛙跳算法的实现步骤

根据简单混洗蛙跳算法的思想和改进策略,免疫进化粒子群混洗蛙跳算法的具体实现步骤如下。

步骤1 程序初始化。设置算法所需的蛙群规模 P 、族群数 M 、族群内青蛙数 N 、族群内最大迭代次数 T' 、个体随机移动最大步长 D 、全局最大迭代次数 T 、IEA 动态调整系数 A 等参数。

步骤2 在解空间内随机产生规模为 P 的初始蛙群,并计算每个个体的适应度值。

步骤3 按照适应度值大小对解群体划分族群,记录每个族群内的最差解、最优解以及全局最优解。

步骤4 对群体的最优解 $x_{b,old}^{(g)}$ 按式(2)进行免疫进化更新,若免疫进化操作后新解的适应度值大于旧解,则用新解 $x_{b,new}^{(g)}$ 代替旧解 $x_{b,old}^{(g)}$, 否则保持旧解不变。

步骤5 对每个族群的最差解 $x_{w,old}^{(k)}$ 用式(3)进行既局部又全局的深度搜索,若新解 $x_{w,new}^{(k)}$ 的适应度值大于旧解 $x_{w,old}^{(k)}$, 则用新解代替旧解,否则保持旧解不变;再对该族群的 N 个个体按适应度值大小排序,并重新确定 $x_b^{(k)}$ 、 $x_w^{(k)}$ 及 $x_b^{(g)}$ 。重复步骤5,直至设定的族群内最大迭代次数 T' 。

步骤6 计算群体中每个个体的适应度值,并根据适应度值大小对蛙群进行混合;若此次迭代的最优解达到设定精度要求或全局最大迭代次数 T , 则算法停止;否则,转入步骤3。

3 函数测试实验及比较

3.1 函数测试实验

本文选取了8个典型测试函数,分别用 IEPSO-SFLA 算法和传统 SFLA 算法优化计算,各函数的函数名、表达式、约束区间及函数理论最优值如表1所示。其中 DeJong F1 函数和 Schwefel's 函数为典型的单峰函数; Camel 函数、Schaffer F6 函数、Schaffer F7 函数和 Shubert 函数为复杂的多峰函数; Rosenbrock 函数是一个单模态的病态函数,它的全局最优解位于一个平滑、狭长的抛物线形山谷内,很难找到全局最优解,是一个复杂的经典优化函数,因此该函数常用来评价优化算法的效率。为了保证两种优化算法对比测试的可比性和有效性,用 IEPSO-SFLA 和 SFLA 求解上述8个函数的全局最优解的过程中,对所有测试函数两种优化算法的参数设置均相同(SFLA 不需要设定动态调整系数 A), 见表2。此外,两种优化算法对各测试函数的全局最大迭代次数 T 亦均设置为相同(见表3)。

为了在相同条件下将 IEPSO-SFLA 与 SFLA 对8个测试函数的计算结果相比较,本文对每个测试函数皆分别独立运行50次,记录其最优值、最劣值、50次计算的平均值和标准偏差,其统计结果见表3。此外,文中还给出了两种优化算法对8个测试函数的适应度值随总进化代数 T 变化的进化曲

线,如图 1 所示。其中函数的适应度表达式为 $G_1 = 1/(1 + f_1)$, $G_2 = 1/(2 + f_2)$, $G_3 = 1/(1 + f_3)$, $G_4 = 1/(1 + f_4)$, $G_5 = 1/(188 + f_5)$, $G_6 = 1/(1 + f_6)$, $G_7 = 1/(1 + f_7)$, $G_8 = 1/(1 + f_8)$ 。

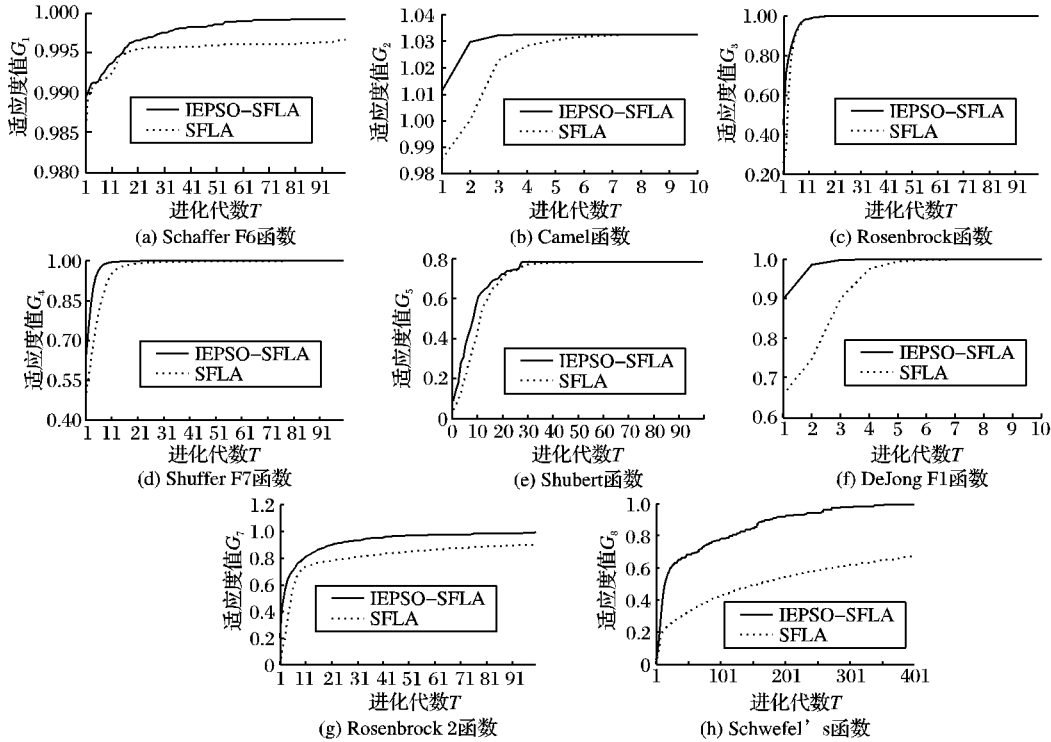


图 1 两种优化算法对 8 个测试函数的进化曲线(适应度值 G_j —全局进化代数 T)

表 1 测试函数

序号	函数名	表达式	约束区间	理论最优值
1	Schaffer F6	$f_1 = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	$[-10, 10]$	0
2	Camel	$f_2 = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2$	$[-2, 2]$	-1.031628
3	Rosenbrock	$f_3 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$[-10, 10]$	0
4	Schaffer F7	$f_4 = (x_1^2 + x_2^2)^{0.25} [1 + \sin^2(50(x_1^2 + x_2^2)^{0.1})]$	$[-10, 10]$	0
5	Shubert	$f_5 = \sum_{i=1}^5 i \cos((i+1)x_1 + i) \cdot \sum_{i=1}^5 i \cos((i+1)x_2 + i)$	$[-10, 10]$	-186.7309
6	DeJong F1	$f_6 = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]$	0
7	Rosenbrock 2	$f_7 = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-5.12, 5.12]$	0
8	Schwefel's	$f_8 = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $	$[-5, 5]$	0

表 2 免疫进化粒子群混洗蛙跳算法和混洗蛙跳算法的参数设置

参数	值	参数	值
群体规模 P	100	族群内最大迭代次数 T'	10
族群数 M	10	青蛙随机移动最大步长 D	0.2
族群内青蛙数 N	10	IEA 动态调整系数 A	1

3.2 IEPso-SFLA 与 SFLA 两种算法测试结果的比较

从表 3 的统计结果可以看出,在运行次数等参数设置均相同的情况下,对所有 8 个测试函数,两种优化算法皆独立运行 50 次的情况下,IEPso-SFLA 获得的最优解、最劣解、50 次运算的平均值和标准偏差都比 SFLA 优化结果和稳定性有非常明显的提高。尤其对 Rosenbrock、Shuffer F7、DeJong F1 和 Schwefel's 等 4 个测试函数 IEPso-SFLA 的平均值、标准差和最差解的求解精度都提高了几个数量级。从图 1 两种优化算法对不同测试函数的进化曲线也可以清晰地看出,无论是在

收敛精度还是收敛效率上,IEPso-SFLA 相比 SFLA 都有明显的优势。此外,从表 3 还可以看出:除 $f_6 \sim f_8$ 3 个函数外,IEPso-SFLA 算法对 $f_1 \sim f_5$ 均能搜索到理论最优解;而 SFLA 算法只能对 $f_1 \sim f_2$ 搜索到理论最优解。

4 结语

免疫进化粒子群混洗蛙跳算法通过对族群内最劣个体同时跟踪族群内的最优个体和群体最优个体的空间位置,进行既全局又局部的更新操作,提高了收敛速度;通过对群体最优个体引入免疫进化迭代计算,保持了群体多样性,避免了不成熟收敛,最终能以更高的精度逼近全局最优解。该算法亦具有原理简单、意义直观、程序设计易于实现的优点。对多个典型测试函数的实验表明,在参数设置相同的情况下,免疫进化粒子群混洗蛙跳算法的优化效果、稳定性和收敛效率均比 SFLA 有显著提高。

表 3 IEPSO-SFLA 与 SFLA 优化结果比较

测试函数	统计项目	全局最大迭代次数 T	SFLA	IEPSOSFLA
f_1	最优解	100	0	0
	最坏解		9.71E-3	9.71E-3
	平均值		3.36E-3	6.62E-4
	标准差		4.62E-3	1.93E-3
f_2	最优解	10	-1.031628	-1.031628
	最坏解		-1.031463	-1.031628
	平均值		-1.031614	-1.031628
	标准差		2.98E-5	1.76E-11
f_3	最优解	100	6.66E-15	0
	最坏解		5.98E-6	2.95E-11
	平均值		2.34E-7	1.31E-12
	标准差		8.79E-7	4.34E-12
f_4	最优解	100	1.19E-10	0
	最坏解		6.12E-6	2.22E-16
	平均值		4.34E-7	4.89E-17
	标准差		1.21E-6	9.29E-17
f_5	最优解	100	-186.7308	-186.7309
	最坏解		-186.7170	-186.7232
	平均值		-186.7275	-186.7298
	标准差		3.30E-3	1.48E-3
f_6	最优解	10	1.73E-7	3.29E-12
	最坏解		3.35E-5	1.62E-9
	平均值		4.52E-6	2.18E-10
	标准差		6.25E-6	2.53E-10
f_7	最优解	100	3.12E-6	3.33E-9
	最坏解		4.04E-1	9.78E-3
	平均值		1.14E-1	1.35E-3
	标准差		8.80E-2	1.84E-3
f_8	最优解	400	6.23E-2	1.18E-8
	最坏解		1.65	5.90E-2
	平均值		0.55	3.48E-3
	标准差		0.35	9.94E-3

由于融入的 PSO 算法的粒子坐标更新策略和引入的 IEA

迭代计算及混洗蛙跳算法(SFLA)都只针对连续函数的优化,因而本文提出的 IEPSO-SFLA 亦只适用于连续函数或参数的优化,若要将其应用于离散函数、组合函数的优化还需要进一步的深入研究。

参考文献:

- [1] LEE Z J, LEE C Y. A hybrid search algorithm with heuristics for resource allocation problem[J]. Information Sciences, 2005, 173(1/3): 155-167.
- [2] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005.
- [3] CHAU K W. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun river[J]. Journal of Hydrology, 2006, 329(3/4): 363-367.
- [4] 李荣钧, 常先英. 一种新的混合粒子群优化算法[J]. 计算机应用研究, 2009, 26(5): 1700-1702, 1705.
- [5] 李祚泳, 汪嘉杨, 郭淳. PSO 算法优化 BP 网络的新方法及仿真实验[J]. 电子学报, 2008, 30(11): 2224-2228.
- [6] 王联国, 洪毅, 赵付青, 等. 一种简化的人工鱼群算法[J]. 小型微型计算机系统, 2009, 30(8): 1663-1667.
- [7] 李雪梅, 张素琴. 基于仿生理论的几种优化算法综述[J]. 计算机应用研究, 2009, 26(6): 2032-2034.
- [8] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3): 459-471.
- [9] 孟伟, 韩学东, 洪炳镪. 蜜蜂进化型遗传算法[J]. 电子学报, 2006, 34(7): 1294-1300.
- [10] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [11] 赵鹏军, 刘三阳. 求解复杂函数优化问题的混合蛙跳算法[J]. 计算机应用研究, 2009, 26(7): 2435-2437.
- [12] 李祚泳, 汪嘉杨, 程会珍. 基于免疫进化算法优化的地下水水质评价普适公式[J]. 水科学进展, 2008, 19(5): 707-713.

《计算机应用》征订启事

《计算机应用》月刊于1981年创刊,是中国计算机学会会刊,由中国科学院成都计算机应用研究所和四川计算机学会主办,科学出版社出版。

《计算机应用》系中文核心期刊、中国科技核心期刊。被《中国科学引文数据库》、《中国科技论文统计源数据库》等国家重点检索机构列为引文期刊,并被英国《科学文摘》(SA)、俄罗斯《文摘杂志》(AJ)、日本《日本科学技术振兴机构数据库》(JST)、美国《剑桥科学文摘:材料信息》(CSA: MI)、波兰《哥白尼索引》(IC)、德国《数学文摘》(Zentralblatt MATH)等多种国外重要检索系统列为来源期刊。

本刊紧紧围绕“应用”。主要涉及网络与通信、信息安全、先进计算、人工智能、图形图像技术、数据库技术、计算机软件技术、现代服务业信息技术和典型应用等。

本刊是您学习计算机应用理论,借鉴计算机应用技术,参考计算机应用经验的最佳选择。

中国标准连续出版物号: ISSN 1001-9081

CN 51-1307/TP

2012年定价: 33元/册

国外发行代号: M4616

国内邮发代号: 62-110

联系人: 雍平

通信地址: 四川成都237信箱(武侯区)《计算机应用》

编辑部(610041)

电话: (028) 85224283-803

传真: (028) 85222239-816

电子邮箱: bjb@joca.cn

网址: www.joca.cn