doi:10.11835/j.issn.1000.582X.2023.12.008

基于自适应步长和莱维飞行策略的改进狼群算法

李彦苍,徐培东
(河北工程大学 土木工程学院,河北 邯郸 056038)

摘要:群智能启发算法在解决大规模分布式问题方面有许多优势。针对传统狼群算法易陷入局部最优和精度不高等缺陷,笔者在分析狼群特点的基础上,提出一种基于自适应性步长和莱维飞行搜索策略的改进狼群算法。首先,通过自适应步长的合理变化,提高搜索精度;其次,采用莱维飞行的搜索策略,在算法后期扩大搜索范围,提高算法的全局搜索能力。最后,为了验证该算法性能,通过仿真实验和实际案例进行了测试,与其他改进方法进行比较。测试结果表明,所提出的改进狼群算法在收敛速度、精度及稳定性方面都有明显优势。

关键词:狼群算法;改进;自适应;莱维飞行;组合优化
中图分类号:TP18 **文献标志码**:A **文章编号**:1000-582X(2023)12-080-16

Improved wolf pack algorithm based on adaptive step size and Levy flight strategy

LI Yancang, *XU Peidong*
(College of Civil Engineering, Hebei University of Engineering, Handan, Hebei 056038, P. R. China;)

Abstract: Swarm intelligence heuristic algorithms offer several advantages in solving large-scale distributed problems. This paper addresses the shortcomings of the traditional wolf pack algorithm which is prone to fall into local optimal and low precision. The paper proposes an improved wolf pack algorithm incorporating adaptive step size and levy flight search strategy after analyzing the characteristics of the wolf pack. Firstly, optimizing the adaptive step size improves search precision, effectively accelerates the convergence speed. Secondly, the incorporation of the levy flight search strategy of expands the search scope, improving the global search capability of the algorithm. Finally, to verify the algorithm's performance, simulations and real-world cases were conducted, comparing it with other improved algorithms. The test results show that the improved wolf pack algorithm has obvious advantages in convergence speed, accuracy and stability.

Keywords: wolf pack algorithm; improvement; adaptive; Levy flight; combinational optimum

群居是一种常见的自然现象,在群居中,社会群体能够适应自然选择原则,在物种内部竞争中生存下来。大雁向南迁移,鱼成群结队游荡在水中搜索食物和蚁群在信息素浓度的帮助下选择最短路径,它们通过减少

收稿日期:2020-10-11 **网络出版日期**:2021-06-08
基金项目:国家自然科学基金资助项目(11202062);河北省高等学校科学技术研究资助项目(ZD2019114)。
Supported by National Natural Science Foundation of China(11202062), Science and Technology Research Project of Hebei Province Colleges and Universities(ZD2019114).
作者简介:李彦苍(1974-),男,博士,教授,博士生导师,主要从事计算智能及其应用方向研究,(E-mail)liyancang@hebeu.edu.cn。

自身能量消耗增加集体利益,是长期自然选择的结果[1-4]。这些集群行为表现为群体的自组织、自适应和团队协作,能够增强群体对环境的整体适应性。群智能算法是一种模拟自然生物进化或觅食行为的一种算法[5]。受动物群体现象启发,人们开发了许多优化计算方法解决复杂问题。常见的群智能算法主要有粒子群优化算法[6]、蚁群优化算法[7]、人工鱼群算法[8]、人工蜂群算法[9]。蝴蝶算法是O'Neil等[10]模仿蝴蝶觅食行为提出的自然启发式优化算法;果蝇算法是Pan等[11]基于果蝇觅食行为提出的全局优化方法;花粉算法是Yang等[12]模拟自然开花植物自授粉和异花授粉生物学特性提出的随机全局优化算法;鸡群算法是Meng等[13]通过模拟鸡的等级和行为提出的优化算法。鸟类、鱼类、蚂蚁和蜜蜂等,它们通过不断适应环境和相互合作,表现出强大的群体智能,给人类提供许多解决复杂问题的新思路,提高处理优化问题的能力,有效推动计算智能的发展,但在计算精度方面仍需进一步研究。

狼群算法(wolf pack algorithm,WPA)是吴胜虎[14]于2013年提出的一种新的群智能算法。该算法在进行优化问题求解时,具有较好的寻优性能,但算法也存在一些不足之处:收敛速度慢、收敛精度不高、鲁棒性低等[15]。文献[16]针对后期收敛速度慢的问题,引入交互式步行运动,提出具有领导策略的狼群搜索算法。文献[17]为解决高维函数优化问题,提出非人工狼群算法(uncultivated wolf pack algorithm,UWPA)。文献[18]采用Tent混沌序列来启动个体位置,提出结合粒子群的狼群优化算法。文献[19]引入差分进化策略,提出基于差分进化的改进狼群算法(improved wolf pack algorithm,IWPA)。文献[20]引入控制自适应参数a和混沌思想,提出自适应调整的混沌灰狼算法。文献[21]提出改进的灰狼算法,通过添加可调参数,提高算法鲁棒性,同时对几种结构进行分析,获得更好解决方案。改进后的算法在一定程度上提高了精确度和收敛精度。

研究在狼群算法的基础上,对狼的游走行为提出了基于莱维飞行的搜索策略,对召唤、围攻行为时的移动步长提出自适应性改进,使每匹狼每次移动的步长由该狼当前位置和当前头狼位置决定。经过测试,提出的自适应步长和莱维飞行策略的改进狼群算法(levy flight and adaptive step size strategy improved wolf pack algorithm,LWPA),收敛速度加快,收敛精度提高,增强了算法的寻优性能和鲁棒性。最后,使用LWPA对桁架结构进行优化设计,并与其他算法进行比较。

1　莱维飞行策略和自适应步长的狼群算法

1.1　智能行为和规则的描述

1.1.1　狼群的初始化

设狼群规模为N,搜索空间的维数为D,第i只人工狼的位置可表示为

$$X_i=(x_i^1,\cdots x_i^d), \tag{1}$$

$$x_{id}=x_{\min}+\mathrm{rand}(x_{\max}-x_{\min}), \tag{2}$$

式中:$x_{\max}$和$x_{\min}$分别是搜索空间的最大范围和最小范围;$\mathrm{rand}\in(0,1)$的一个随机数。

1.1.2　头狼产生规则

在初始解空间中,目标函数值最优的人工狼被选为头狼,每次迭代后更新人工狼的位置。此时如有多个最优人工狼情况,则随机选一个成为头狼。头狼不执行以下智能行为,直接进入迭代,直到被其他更强的人工狼替代。

1.1.3　基于莱维飞行的游走行为

除头狼外选取最佳的S_sum匹人工狼视为探狼,S_sum随机取$\left[n/\alpha+1,n/\alpha\right]$之间的整数,$\alpha$为探狼比例因子。在实际情况中发现,游走过程中探狼只会盲目跟随头狼并逼近头狼位置的猎物气息浓度,不会关心自己身边是否有更优猎物气息浓度,在算法后期,导致种群丧失多样性,易陷入局部收敛,出现早熟收敛现象。针对这种缺陷,利用莱维飞行对群体中的探狼进行全局搜索。莱维飞行属于随机游动,是一种很好的搜索策略,能扩大搜索范围[22]。新一代的探狼i的计算公式如下

$$x_{id}(t+1)=x_{id}(t)-\oplus\mathrm{Levy}(\delta), \tag{3}$$

式中:$x_{id}(t)$表示探狼i在t次迭代第d维的位置;$\oplus$为点对点乘法;c为探狼i位置的随机数,由式(4)决定,$\mathrm{Levy}(\delta)$代表随机搜索路径,由式(5)决定。

$$c=\mathrm{rand}(\mathrm{size}(i\_\mathrm{position}))\,, \tag{4}$$

$$\mathrm{Levy}(\delta)\sim 0.01\frac{u}{|v|^{\frac{1}{\delta}}}(X_i-X_{i\mathrm{best}})\,, \tag{5}$$

式中：δ的取值范围一般为$1<\delta<3$，δ取1.5，X_{best}表示历史最优探狼位置，u和v服从式(6)所示的正态分布

$$u\sim N(0,\sigma_u^2),v\sim N(0,\sigma_v^2)\,。 \tag{6}$$

σ_u和σ_v取值为

$$\sigma_u=\left\{\frac{\Gamma(1+\beta)\sin(\frac{pi\beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right)\beta 2^{\frac{(\beta-1)}{2}}}\right\}^{\frac{1}{\beta}},\sigma_v=1\,。 \tag{7}$$

此时，探狼感知的猎物气息浓度函数值为Y_{ip}，选择最大的猎物气息浓度函数值，若大于当前函数值Y_i，则向Y_{ip}的方向前进一步，同时更新探狼状态，重复以上游走行为，直到某匹探狼i的函数值$Y_i>Y_{\mathrm{lead}}$或游走次数达到最大游走次数$T_{1\max}$。

1.1.4 奔袭行为

在基本的狼群算法中，狼群位置变动是由步长step决定的，对于每一个固定的D维空间，相应的$[d_{\min},d_{\max}]$是固定的，因而每一次迭代对应的步长step是固定的。如果step过大，会影响算法优化的准确度；如果step过小，影响算法的收敛速度，当达到最大迭代次数时，最优解还未被找到。借鉴文献[23]，狼i每一次移动的步长由该狼当前位置和当前头狼位置决定，因此在奔袭和围攻行为时采用自适应步长

$$\mathrm{step}=\mathrm{rand}\times\left\|x_i-X_{\mathrm{leadd}}\right\|_2,\ d=1,2,\cdots,D, \tag{8}$$

式(8)中rand表示[0,1]间的随机数，当狼离头狼距离远时，以较大步长逼近头狼，加快收敛速度，避免不必要的搜索；当离头狼距离近时，以较小步长逼近头狼，提高搜索精细程度。

与以往狼群算法不同，随机选取除头狼外的全部狼群中M_num只猛狼参与召唤，而不仅是头狼附近的人工狼。在猛狼奔袭过程中，当某只猛狼感知到其所在位置猎物气息浓度更高时，则替代头狼，重新选取猛狼，进行召唤，直到其所在位置的猎物气息浓度低于头狼位置气息浓度。同时，召唤行为的步长取式(8)，猛狼根据式(9)更新当前位置

$$x_{id}^{\ *}=x_{id}+\mathrm{rand}\times\left\|x_{id}-x_{\mathrm{leadd}}\right\|_2\times\frac{x_{\mathrm{leadd}}-x_{id}}{\left|x_{\mathrm{leadd}}-x_{id}\right|},\ d=1,2,\ \cdots,\ D, \tag{9}$$

式中：x_{id}^{*}表示更新后猛狼的位置；x_{id}为当前猛狼的位置；x_{leadd}为头狼的位置。

1.1.5 围攻行为

同时，猛狼联合探狼对猎物进行围捕。移动步长采用式(8)，狼群围攻行为由式(10)表示

$$x_{id\,1}\ =x_{id}+\lambda\times\mathrm{rand}\times\left\|x_{id}-G_d\right\|_2\times\left|G_d-x_{id}\right|,\ d=1,2,\cdots,D, \tag{10}$$

式中：G_d为猎物在D维空间的位置，λ为区间[-1,1]均匀分布的随机数。

1.1.6 “强者生存”的狼群更新机制

猎物的分配遵循“由强到弱”的原则，即在算法中去除目标函数值最差的R匹人工狼，同时随机产生R匹人工狼，在实际捕猎过程中，每次捕猎的数量都是随机的，这也导致不同数量的弱狼被淘汰。基于此，β取$\left[n/(2\times\beta),n/\beta\right]$之间的随机整数，$\beta$为更新比例因子。

1.2 改进狼群算法描述

Step1 初始化狼群中人工狼的数目N和其所在位置X_i，最大迭代次数$K_{\max}$，探狼比例因子α，更新比例因子β，最大游走次数$T_{1\max}$，最大奔袭次数$T_{2\max}$。

Step2 根据头狼产生规则确定头狼。

Step3 探狼按照莱维飞行策略公式(3)～(7)执行游走行为，直到某匹探狼i的函数值$Y_i>Y_{\mathrm{lead}}$或游走次数达到最大游走次数$T_{1\max}$，转Step4。

Step4　猛狼执行奔袭行为,并按照公式(9)向猎物奔袭。在奔袭过程中,若猛狼感知的猎物气息浓度的函数值 $Y_i > Y_{\text{lead}}$,则令 $Y_i = Y_{\text{lead}}$,该猛狼转化为头狼并发起召唤行为;若 $Y_i < Y_{\text{lead}}$,则继续奔袭直到某匹猛狼的函数值小于头狼函数值或奔袭达到最大奔袭次数 $T_{2\max}$,转 Step5。

Step5　按照公式(10),更新参与围攻的人工狼位置,进行围攻。

Step6　执行狼群的更新机制。

Step7　判断算法是否满足优化精度要求或最大迭代次数 $K_{\max}$,若满足要求,输出头狼位置,即所求问题的最优解,否则转 Step2。

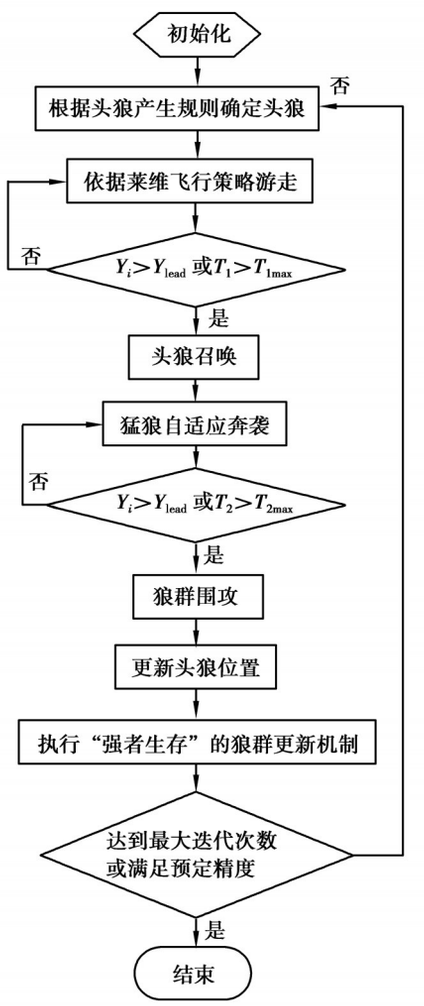


图 1　LWPA 算法的基本流程图

Fig. 1　LWPA algorithm iteration diagram

2　实验仿真与分析

2.1　基本测试函数与参数设置

表 1 中的“*U*”表示函数为单模态函数,“*M*”为多模态函数,“*S*”为可分离函数,“*N*”为不可分离函数。多模态函数比单模态函数复杂,一般算法难以找到具有多个局部极值的全局最优值,容易陷入局部极值或局部极值之间的振荡[24]。因此,多模态常被用来测试算法的全局搜索性和避免早熟收敛能力[25-26]。由于不可分函数变量之间的关系较复杂,很难找到不可分函数的全局最优值[27-28]。WPA、UWPA 以及 IWPA 算法所涉及到的参数分别参考文献[14-19]进行设置。N 取 50,$K_{\max}$ 取 1 000,β 取 4,α 取 4,$T_{1\max}$($T_{2\max}$)取 10。

表1 用于测试算法性能的15个函数

Table 1 15 functions for testing algorithm performance

编号	函数	表达式	维数	特征	取值范围	理论最优解
1	Eason	$-\cos x_1 \cos x_2 \times \exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	2	UN	[-100,100]	$\min f=-1$
2	Matyas	$f(X)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	2	UN	[-10,10]	$\min f=0$
3	Booth	$f(X)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	2	MS	[-10,10]	$\min f=0$
4	Bohachevs1	$f(X)=x_1^2+2x_2^2-0.3\cos(3pix_1)-0.4\cos(4pix_2)+0.7$	2	MS	[-100,100]	$\min f=0$
5	Eggcrate	$f(X)=x_1^2+x_2^2+25(\sin^2 x_1+\sin^2 x_2)$	2	MS	[-pi,pi]	$\min f=0$
6	Schaffer	$f(X)=0.5+\frac{\left(\sin\sqrt{(x_1^2+x_2^2)}\right)^2-0.5}{\left(1+0.001(x_1^2+x_2^2)\right)^2}$	2	MN	[-100,100]	$\min f=0$
7	Six Hump Camel Back	$f(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	MN	[-5,5]	$\min f=-1.0136$
8	Bohachevs3	$f(X)=x_1^2+2x_2^2-0.3\cos(3pix_1+4pix_2)+0.3$	2	MN	[-100,100]	$\min f=0$
9	Bridge	$f(X)=\frac{\sin\sqrt{(x_1^2+x_2^2)}}{\sqrt{(x_1^2+x_2^2)}}+\exp\left(\frac{\cos 2pix_1+\cos 2pix_2}{2}\right)-0.7129$	2	MN	[-1.5,1.5]	$\min f=0$
10	Trid6	$\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D}x_ix_{i-1}$	6	UN	[-36,36]	$\min f=-50$
11	Sumsquares	$\sum_{i=1}^{D}ix_i^2$	10	US	[-10,10]	$\min f=0$
12	Sphere	$\sum_{i=1}^{D}x_i^2$	30	US	[-1.5,1.5]	$\min f=0$
13	Rastrigin	$\sum_{i=1}^{D}[x_i^2-10\cos 2\pi x_i+10]$	60	MS	[-10,10]	$\min f=0$
14	Quadric	$\sum_{i=1}^{D}(\sum_{k=1}^{i}x_k)^2$	120	MS	[-30,30]	$\min f=0$
15	Ackley	$-20\exp(-0.2\sqrt{(\frac{1}{D}\sum_{i=1}^{D}x_i^2)})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i)+20+e$	200	MN	[-32,32]	$\min f=0$

2.2　算法对比验证

为充分计算算法的性能,使用LWPA、UWPA、WPA以及IWPA分别对15个复杂函数进行了100次连续优化计算[29]。基于表2的6种指标对该算法进行评估。当不同的计算结果与最优值之间的误差超过e-3,被认为是一种失败。结果见表2。

表2　4种算法在15个测试函数中的结果比较

Table 2　Comparison of results of four algorithms in 15 test functions

编号	函数	算法	最优值	最坏值	平均值	标准差	成功率	耗时/s
1	Eason	LWPA	-1	-1	-1	1e-7	100	1.384 6
		UWPA	-1	-0.999 985	-0.999 999	0	100	2.750 3
		WPA	-0.900 99	-0.900 990	-0.260 600	0.160 1	12	33.112 0
		IWPA	-0.999 94	-0.952 511	-0.985 008	1.59e-4	34	3.987 8
2	Matyas	LWPA	1.43e-8	1.59e-5	4.16e-6	2.03e-11	100	1.497 1
		UWPA	8.22e-07	0.027 27	0.004 330	4.24e-05	86	2.913 6
		WPA	-127.83	-6 897.11	-2 474.76	1 426.25	21	35.119 0
		IWPA	-41.57	-2.680 50	-19.575 200	105.045	46	7.349 5
3	Booth	LWPA	1.96e-9	5.10e-6	5.76e-6	1.50e-10	100	1.554 6
		UWPA	3.41e-8	6.06e-8	1.24e-8	1.81e-8	100	1.999 3
		WPA	2	2.001 01	2.000 010	1.080 80	32	29.547 0
		IWPA	1.44e-9	0.114 56	0.018 520	6.64e-4	47	3.453 9
4	Bohachevs1	LWPA	1.09e-7	2.37e-6	8.20e-7	3.34e-13	100	3.096 9
		UWPA	6.79e-3	0.595 89	0.166 070	0.026 67	98	4.405 1
		WPA	7.69e-10	171.208 00	14.062 100	915.964	11	34.021 0
		IWPA	2.59e-13	7.58e-3	1.67e-4	6.63e-7	91	3.420 7
5	Eggcrate	LWPA	2.50e-8	2.94e-6	9.21e-7	6.54e-13	100	0.982 0
		UWPA	2.34e-5	6.81e-4	1.75e-4	2.15e-8	100	7.993 9
		WPA	6.07e-13	9.21e-7	6.68e-8	1.69e-14	100	43.983 0
		IWPA	9.56e-17	5.24e-6	1.59e-7	4.13e-13	100	3.679 8
6	Schaffer	LWPA	2.95e-8	0.000 98	0.000 820	1.03e-5	100	0.885 2
		UWPA	4.76e-4	0.037 27	0.010 270	3.47e-5	48	1.502 8
		WPA	3.02e-11	0.126 99	0.019 260	7.72e-3	12	63.947 0
		IWPA	1.478e-9	0.037 22	0.013 100	8.97e-5	24	6.818 5
7	Six Hump Camel Back	LWPA	-1.031 6	-1.031 60	-1.031 600	0	100	1.045 1
		UWPA	-1.031 6	-1.031 20	-1.031 500	8.14e-9	100	5.358 3
		WPA	-1.85e-8	-4.47e-13	-2.27e-900	1.49e-17	8	59.445 0
		IWPA	-1.031 6	-0.532 90	-0.973 800	7.50e-3	12	3.563 9
8	Bohachevs3	LWPA	1.28e-7	8.74e-4	3.72e-4	4.50e-7	100	2.502 0
		UWPA	2.17e-5	0.269 21	0.050 280	2.72e-3	13	3.403 3
		WPA	6.14e-7	175.978	9.333 130	616.702 00	14	33.866 0
		IWPA	2.66e-7	0.230 80	4.16e-2	2.89e-2	26	3.556 7
9	Bridge	LWPA	-3.005 40	-3.005 40	-3.005 400	0	100	1.138 0
		UWPA	-3.005 38	-3.005 30	-3.005 300	1.36e-9	80	15.332 0
		WPA	-3.005 40	-2.705 12	-2.939 900	0.015 22	6	61.280 0
		IWPA	-3.005 30	-2.624 07	-2.973 420	4.48e-3	8	5.607 4
10	Trid6	LWPA	-50	-48.495 00	-49.780 800	0.206 95	95	7.129 4
		UWPA	-127.830 00	-6 897.110 00	-2 474.760000	1 426.25	95	35.119 0
		WPA	-33.500 00	-33.500 00	-33.500 000	1.14e-16	0	7.591 8
		IWPA	-41.570 00	-2.680 50	-19.575 200	105.045 00	0	7.349 5

续表2

编号	函数	算法	最优值	最坏值	平均值	标准差	成功率	耗时/s
11	Sumsquares	LWPA	1.08e-6	4.71e-6	2.49e-6	1.69e-12	100	2.668 2
		UWPA	1.02e-5	3.41e-4	9.99e-5	3.88e-9	100	1.603 4
		WPA	3.005 38	6 897.11	2 326.73	1.88e+6	0	20.908 0
		IWPA	3.34e-6	7.96e-2	2.91e-3	8.99e-5	70	11.585 0
12	Sphere	LWPA	2.27e-7	4.56e-7	3.66e-7	5.30e-15	100	1.874 1
		UWPA	0.001 02	0.005 71	0.002 3	6.09e-7	80	1.554 0
		WPA	3.005 38	6 897.11	1 790.03	2.94e+4	0	67.006 0
		IWPA	6.42e-4	0.281 89	0.013 68	1.17e-3	1	26.663 0
13	Rastrigin	LWPA	2.11e-10	3.68e-8	1.43e-8	1.16e-8	100	8.563 2
		UWPA	52.788 6	157.196	100.765	240.213	0	6.987 2
		WPA	2 082.85	3 105.82	2 497.08	523.516	0	207.870 0
		IWPA	2.29e+2	4.85e+2	3.43e+2	4.23e+3	0	56.761 0
14	Quadric	LWPA	6.85e-16	1.50e-5	1.61e-5	1.42e-11	100	29.845 0
		UWPA	3 474.1	500 129	128 464	5.36e+9	0	27.512 0
		WPA	7.9e+10	5.20e+10	1.24e+10	1.1e+10	0	377.980 0
		IWPA	7.61e+7	3.03e+8	1.60e+8	1.17e+5	0	140.550 0
15	Ackley	LWPA	1.155 00	2.81	2.25	0.513 6	100	37.706 0
		UWPA	2.211 22	3.907 41	3.073 91	0.114 95	0	66.859 0
		WPA	20.529 10	21.480 4	21.195 3	0.187 6	0	367.290 0
		IWPA	19.380 00	20.107 1	19.804 3	2.41e-2	0	232.910 0

2.3 LWPA主要参数分析

LWPA虽然具有一定优越性,但所涉及的参数也众多,主要参数对算法性能的影响也不尽相同。$T_{1\max}/T_{2\max}$分别是狼群在游走/奔袭过程中的最大次数,β是狼群的更新比例系数。根据15个函数的特性将其分成7大类,分别改变$T_{\max}$和β的大小对这7种函数进行50次寻优计算,$T_{\max}$和β对算法性能的影响如表3、4所示。

2.4 LWPA收敛性分析

Markov链是一种无后效性的随机过程,常被应用于分析收敛性问题。由于LWPA是基于游走、召唤和围攻3种智能行为的不断重复,每种行为都与当前的群体状态有关,而与之前无关,因此LWPA的种群序列为Markov链。设$Q_k=\{X_1,X_2,\cdots,X_N\}$为LWPA的第$k$代种群,其中$N$为人工狼总数,$X_i$为第$i$匹人工狼的状态。

定理1 文献[30]已经证明若一个进化算法满足:1)对可行解空间中任意2点x_1和x_2,x_2是x_1由算法中的各种算子产生且是可达的;2)若种群序列$Q_1,Q_2,\cdots,Q_N$是单调的,则此进化算法是以概率1收敛于问题的全局最优解。

定理2 LWPA算法以概率1收敛于问题的全局最优解。

证明:由文献[14]的推理可知LWPA种群序列的Markov链也是遍历链,且LWPA优化序列是一个有限齐次Markov链,每次迭代狼群个体位置状态只有遇到更优解时才会更新。因此,LWPA产生的子代Q_{k+1}中的任意解都不差于Q_k中的任意解。由此可知种群序列$Q_1,Q_2,\cdots,Q_N$是单调的,于是由定理1得证,LWPA以概率1收敛于问题的全局最优解。

2.5 结果分析

由表2各种算法的对比结果可知:

1)对于单峰、低维的不可分函数Eason、Matyas, LWPA与UWPA都寻优成功且具有较好性能,接近于最优值,就耗时而言,UWPA的消耗时间是LWPA的2倍,IWPA和WPA的精度较差。

2)对于多峰、低维的可分函数Booth、Bohachevs1、Eggcrate,LWPA的收敛精度明显高于其他3种算法,达到1e-6以上,耗时方面,LWPA和UWPA的耗时最短,IWPA次之,WPA耗时最长;

表 3　T_{max}对算法的影响

Table 3　Effect of T_{max} on the algorithm

函数	标准差					
	6	8	10	12	14	16
Eason	2.4e-8	8.4e-9	9.6e-9	1.4e-9	2.6e-8	2.1e-8
Booth	3.2e-11	4.2e-11	7.5e-11	4.1e-11	3.2e-11	2.6e-11
Bridge	5.6e-15	5.6e-15	5.6e-15	5.6e-15	5.6e-15	5.6e-15
Sumsquares	3.2e-7	3.4e-7	5.5e-7	3.8e-7	3.6e-7	3.0e-7
Sphere	7.1e-14	4.5e-15	7.8e-16	1.9e-15	8.7e-14	6.9e-14
Quadric	4.1e-16	2.4e-19	4.2e-19	8.3e-18	8.5e-17	5.4e-16
Ackley	5.3e-17	5.3e-17	5.3e-17	5.3e-17	5.3e-17	5.3e-17

表 4　β对算法的影响

Table 4　Effect of β on the algorithm

函数	标准差					
	2	3	4	5	6	7
Eason	4.0e-8	8.7e-8	1.2e-9	8.6e-8	6.4e-8	5.9e-8
Booth	6.1e-11	9.5e-11	1.9e-12	8.5e-11	7.5e-11	6.3e-11
Bridge	5.6e-15	5.6e-15	5.6e-15	5.6e-15	5.6e-15	5.6e-15
Sumsquares	3.2e-7	4.4e-7	1.2e-8	5.4e-7	3.9e-7	3.4e-7
Sphere	1.6e-14	5.2e-15	2.9e-16	4.3e-15	3.9e-15	2.3e-14
Quadric	2.8e-17	1.4e-18	2.8e-19	8.3e-16	6.9e-15	5.3e-15
Ackley	5.3e-17	5.3e-17	5.3e-17	5.3e-17	5.3e-17	5.3e-17

3)对于多峰、低维的不可分函数 Schaffer、Six Hump Camel Back、Bohachevs3、Bridge,WPA 与 IWPA 由于陷入局部最优而导致寻优失败,LWPA 与 UWPA 寻优成功且 LWPA 具有更好的寻优性能,同时,LWPA 的耗时最短,明显少于其他 3 种算法;

4)对于单峰、高维的不可分函数 Trid6,LWPA 与 UWPA 寻优成功且性能较好。耗时方面,除 WPA 耗时较长外,其他 3 种算法耗时相当;

5)对于单峰、高维的可分函数 Sumsquares、Sphere,LWPA 与 UWPA 寻优成功,LWPA 的寻优精度明显优于其他 3 种算法,达到 1e-7 以上,但在耗时方面,UWPA 略优于 LWPA;

6)对于多峰、高维的可分函数 Rastrigin、Quadric,随着维数的递增,只有 LWPA 寻优成功且精度较高,LWPA 与 UWPA 的耗时都较短;

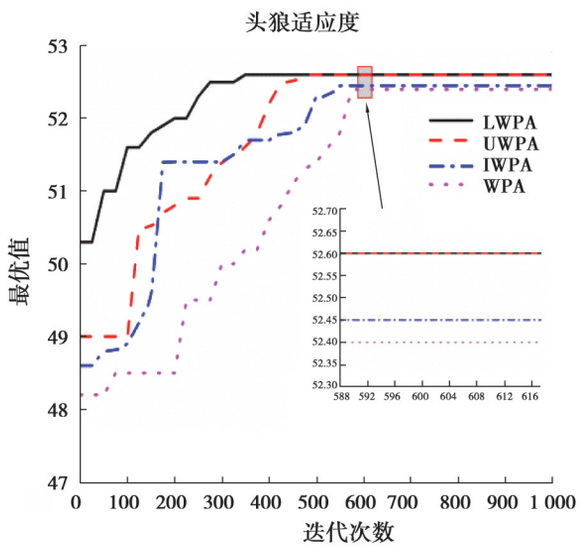
7)对于多峰、高维的不可分函数 Ackley,只有 LWPA 寻优成功,耗时也最短。

就时间复杂度而言,由于探狼在游走过程中,探狼采取莱维飞行的随机搜索策略,扩大搜索范围,增加了运行时间。但在对召唤、围攻行为的移动步长进行自适应性改进,当狼群离头狼较远时,以较大步长逼近头狼;当离头狼距离较近时,以较小步长逼近头狼,实现自适应性的灵活调节,使算法更具灵活性,极大缩短运行时间,搜索效率进一步提高。

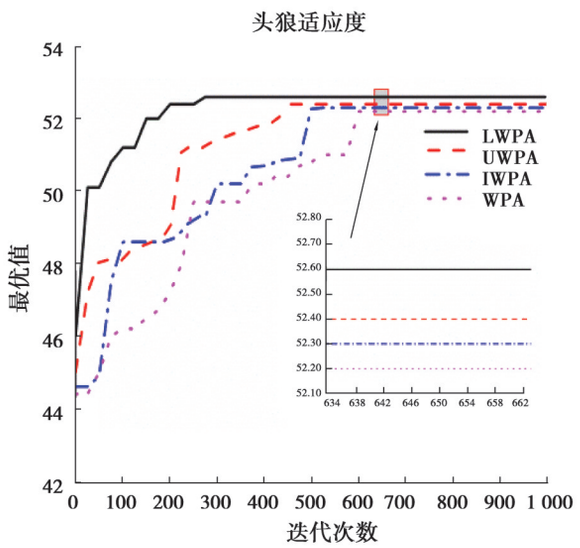
综上可知,无论是从精度方面还是耗时方面,基于自适应步长和莱维飞行策略的改进狼群算法在处理函数问题时比其他改进的狼群算法更精确、效果更好,尤其是对多峰、高维的复杂函数,效果更佳。UWPA 的效果次之,IWPA 和 WPA 较差。由表 3 可知,随着 T_{max} 增大,标准差呈现出先减小后增大的趋势。若 T_{max} 值太小,会导致狼群搜索效率降低,耗时较长,找不到最优解;若 T_{max} 值太大,以至于忽略最优解,达不到所需精度。综上,算法中 T_{max} 取 10。由表 4 可知,随着 β 增大,标准差呈现先减小后增大趋势。若 β 太小,狼群更新数量太多,狼群难以聚集,导致算法优化效果降低;但若 β 太大时,狼群更新的数量太少,导致狼群多样性的急

剧下降，并且容易陷入局部最优。综上，算法中β取4。

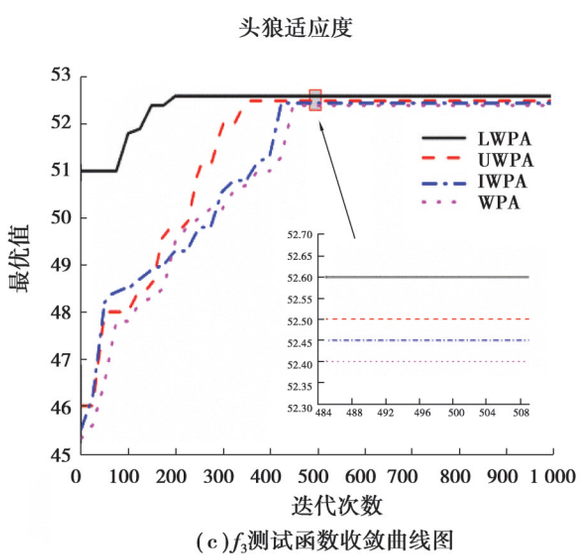
为进一步直观说明LWPA的优越性，图2给出了LWPA与UWPA、IWPA、WPA在各测试函数中的收敛曲线图。从图中看出，对于单峰、低维的不可分复杂函数，当算法迭代到300次时，LWPA已经找到最优值且趋于稳定，而其他3种算法迭代到400次时虽也趋于稳定，但精度较差；对于简单的低维函数，UWPA在前期搜索中效果最好，但随着后期搜索效率不高，导致耗时长且易陷入局部最优；对于复杂函数，LWPA的收敛精度明显高于其他3种算法，当维数增加到30维、60维、120维，甚至200维时，UWPA、IWPA、WPA 3种算法寻优效果明显较差，出现前期搜索时间较长，后期陷入局部最优的情况。因此，针对算法在后期容易陷入局部最优问题，LWPA已经有很好改进。比较可知，与其他3种改进狼群算法相比，在收敛速度还是收敛精度方面，LWPA的优化性能都有明显提升，说明改进方向的正确性。



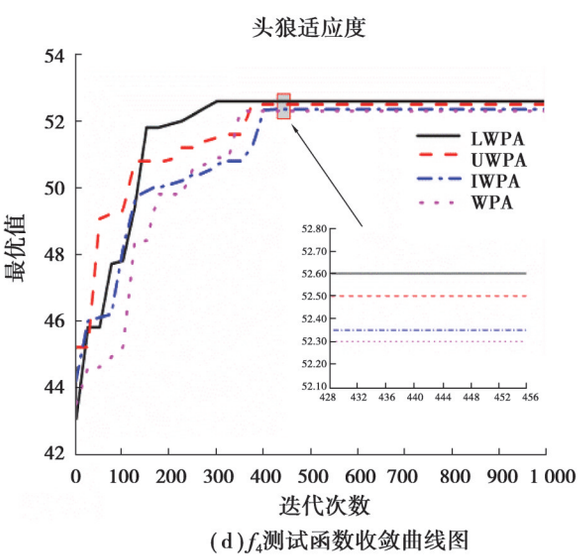
(a)f_1测试函数收敛曲线图



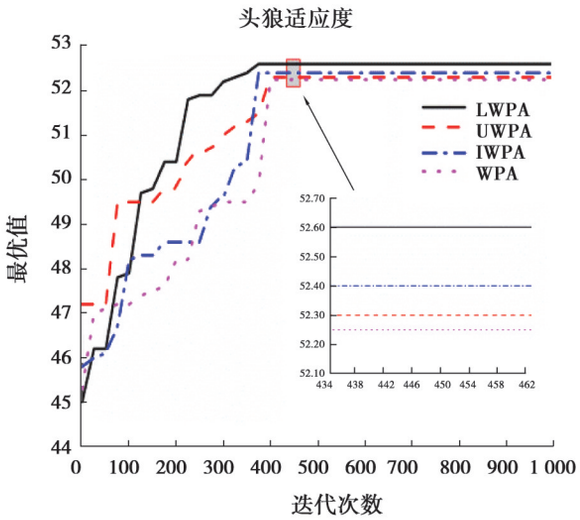
(b)f_2测试函数收敛曲线图



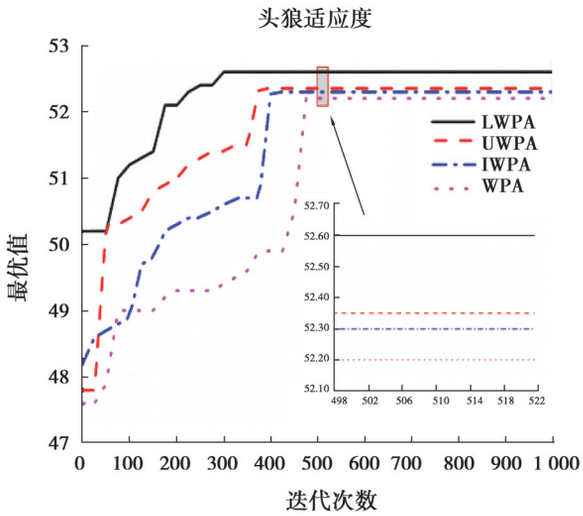
(c)f_3测试函数收敛曲线图



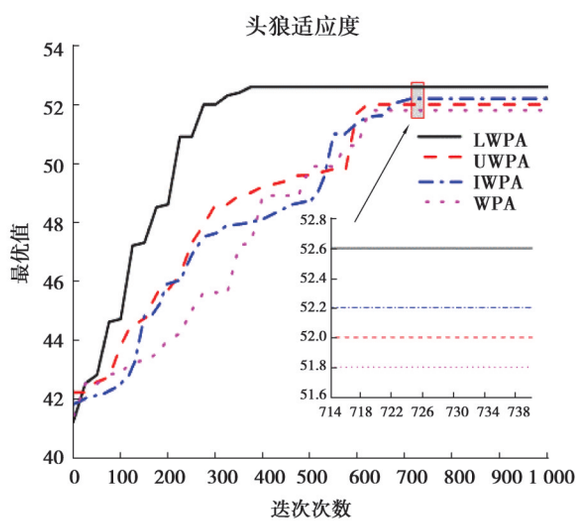
(d)f_4测试函数收敛曲线图



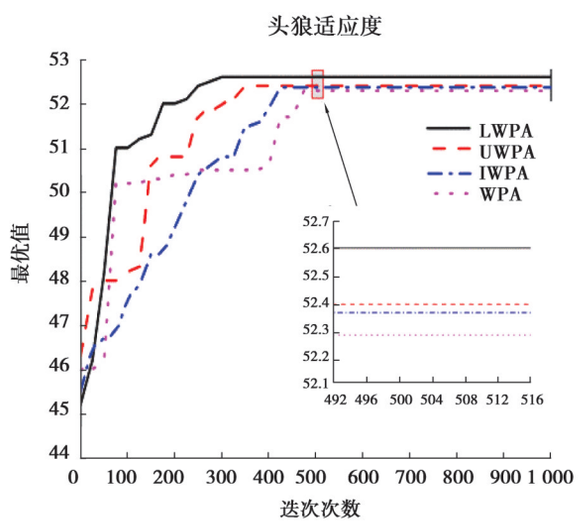
(e)f_5测试函数收敛曲线图



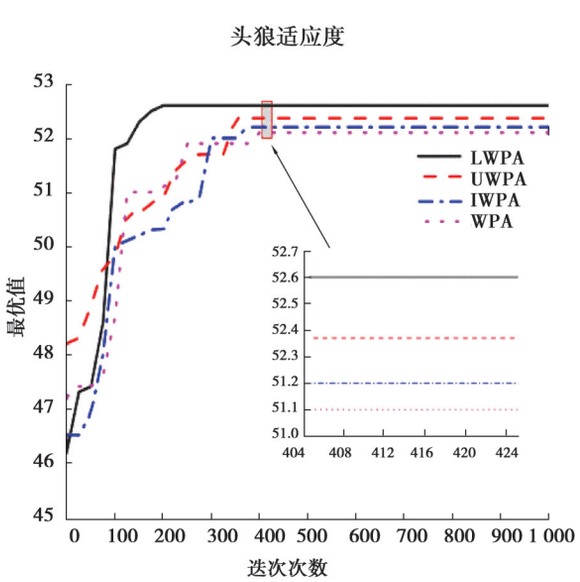
(f)f_6测试函数收敛曲线图



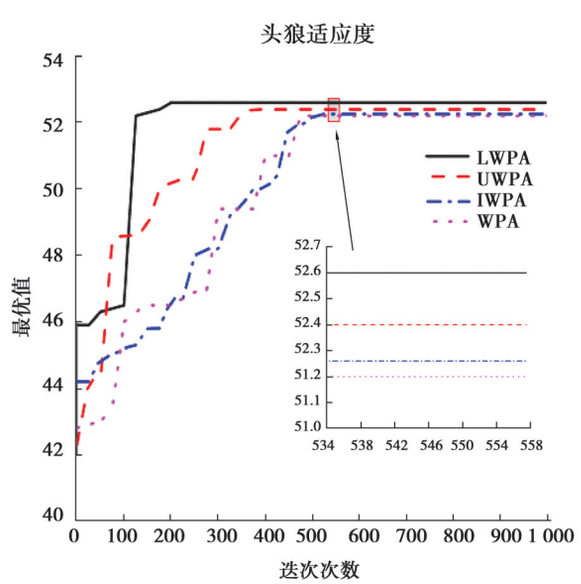
(g)f_7测试函数收敛曲线图



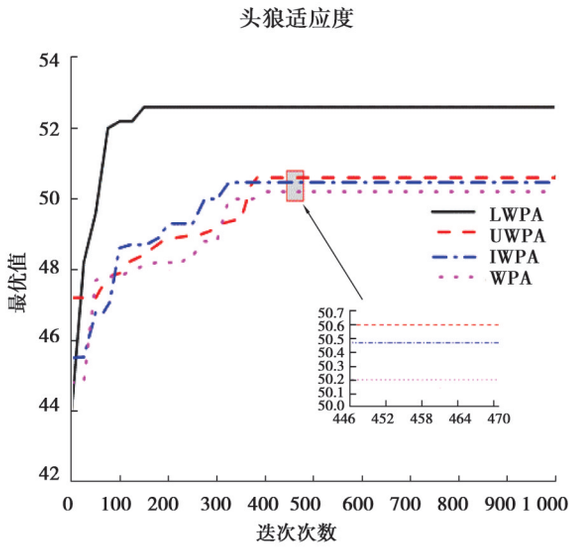
(h)f_8测试函数收敛曲线图



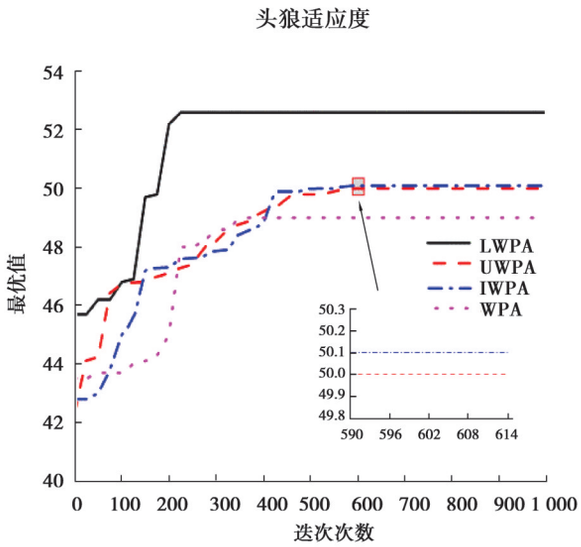
(i)f_9测试函数收敛曲线图



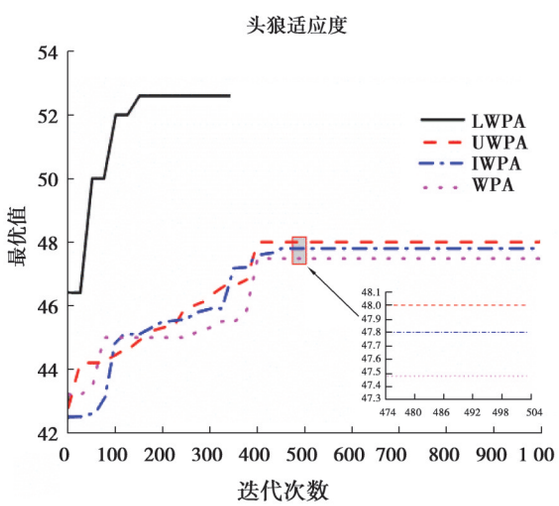
(j)f_{10}测试函数收敛曲线图



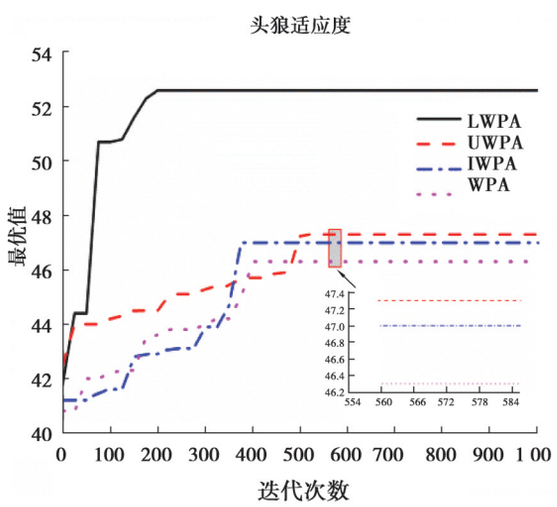
(k)f_{11}测试函数收敛曲线图



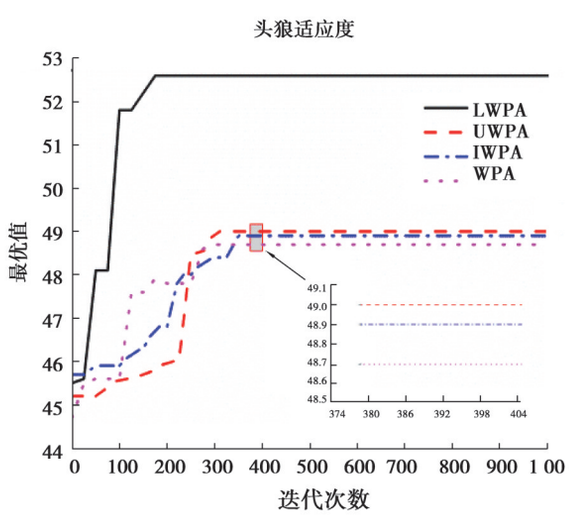
(l)f_{12}测试函数收敛曲线图



(m)f_{13}测试函数收敛曲线图



(n)f_{14}测试函数收敛曲线图



(o)f_{15}测试函数收敛曲线图

图2 15个测试函数的收敛曲线图

Fig. 2 Convergence curves of 15 test functions

3　LWPA对桁架结构优化实例

3.1　桁架结构优化模型

1)优化模型

以截面积为设计变量的桁架优化模型问题可描述为

$$\min F = W(x), \tag{11}$$

$$\text{s.t.} g_i(x) \leqslant 0, i = 1,2,\cdots,p\ , \tag{12}$$

式中:$g_i(x)$为约束函数;p为约束个数。

2)目标函数

$$W(A) = \sum_{i=1}^{n} \gamma A_i L_i\ , \tag{13}$$

式中:$W(A)$为结构的重量;A_i为第i根杆件的截面积;L_i为第i根杆件的长度;γ为材料密度;n为设计变量个数。

3)约束条件

各杆必须满足对强度、刚度、稳定性和截面尺寸要求,约束条件如下

$$\frac{\sigma_i}{[\sigma]} - 1 \leqslant 0, \tag{14}$$

$$\frac{\mu_j}{\mu_{\max}}, \tag{15}$$

$$A_{\min} \leqslant A_i \leqslant A_{\max}。 \tag{16}$$

式中:σ_i为第i杆的轴向正应力;$[\sigma]$为材料的许用应力;μ_j为节点j的位移;$\mu_{\max}$为节点j的许用位移;$A_{\min}$、$A_{\max}$分别为杆件截面的上限、下限。

3.2　算例1

10杆平面桁架结构见图3,此桁架有6个节点,10个设计变量,如表5所示。优化目标是获得最小的结构总重量。E=68 950 Mpa,ρ =2 768 kg/m^3,全部杆件的许用应力为±172.4 MPa,各杆件截面积的下限为0.645 cm^2,上限为258 cm^2,工况只有一个,在5、6号节点作用向下载荷P=4.445 KN的集中力,可动节点向下的位移约束均5.08 cm,图中l=914.4 cm。

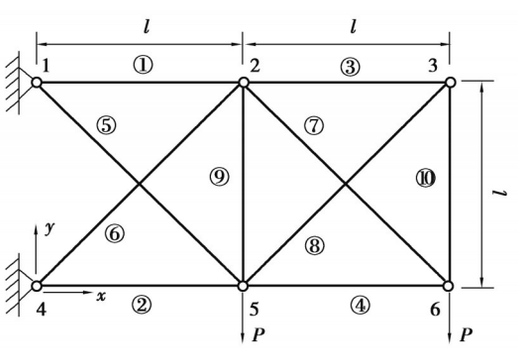


图3　10杆平面桁架结构图

Fig. 3　Plane truss structure diagram of 10-bar

表5　10杆桁架结构优化结果对比

Table 5　Optimal results of the 10-bar truss structure

杆件编号	杆件截面面积/cm^2			
	遗传算法(GA)	萤火虫算法(FA)	狼群算法(WPA)	改进狼群算法(LWPA)
1	32.557	37.813	36.299	35.1465
2	16.678	9.9691	14.131	14.6658

续表 5

杆件编号	杆件截面面积/cm²			
	遗传算法（GA）	萤火虫算法（FA）	狼群算法（WPA）	改进狼群算法（LWPA）
3	32.557 0	40.366 0	34.855 0	35.687 30
4	16.678 0	16.889 0	14.911 0	15.091 90
5	2.215 2	2.167 8	0.664 4	0.645 04
6	4.567 5	3.965 2	4.872 3	4.622 12
7	22.911 0	25.409 0	23.568 0	23.554 80
8	22.911 0	21.714 0	25.609 0	24.467 80
9	17.591 0	11.678 0	12.808 0	12.718 70
10	17.591 0	11.287 0	12.452 0	12.684 10
结构总重量/kg	526.550 0	514.580 0	513.350 0	509.620 00

3.3 算例 2

25 杆空间桁架结构如图 4 所示，该结构有 10 个节点，25 根杆件。应力约束为[-275.8,275.8]Mpa，材料的密度 ρ =2 768 kg/m³，弹性模量 E=68 950 Mpa，1、2 节点的最大竖向位移不能超过 d_{max}=0.889 cm，L=63.5 cm。根据对称性，将 25 根杆件分成 8 组，即设计变量为 8 个，如表 6-7 所示。

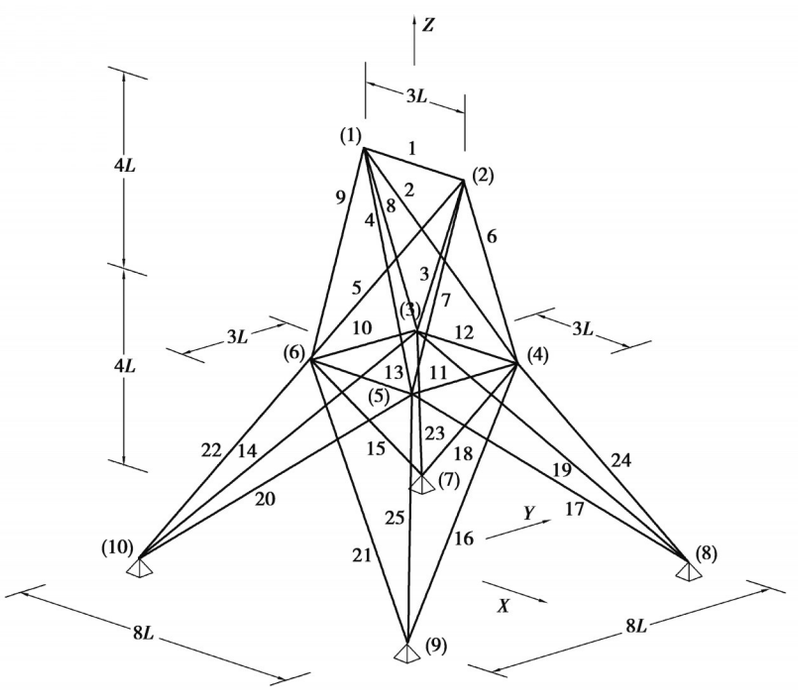


图 4 25杆空间桁架结构图

Fig. 4 The 25-bar spatial truss structure

表 6 25杆空间桁架工况荷载

Table 6 Load cases of the 25-bar spatial truss structure

节点编号	F_x	F_y	F_z
1	4.448	44.482	-22.241
2	0	44.482	-22.241
3	22.241	0	0
6	22.241	0	0

表 7　25杆空间桁架结构优化结果对比

Table 7　Comparison of optimal results for the 25-bar spatial truss structure

编号	杆件截面面积/cm^2			
	遗传算法（GA）	萤火虫算法（FA）	狼群算法（WPA）	改进狼群算法（LWPA）
A_1	0.645 0	14.245 0	0.645 0	0.645 0
A_2-A_5	8.881 0	17.438 0	0.645 0	0.645 0
A_6-A_9	8.721 0	14.231 0	26.248 0	29.489 0
A_{10}-A_{11}	11.451 0	15.331 0	0.917 1	0.645 0
A_{12}-A_{13}	3.004 0	12.824 0	13.900 0	17.267 0
A_{14}-A_{17}	7.200 0	6.856 0	8.056 9	6.417 5
A_{18}-A_{21}	0.645 0	7.078 0	1.926 9	0.891 9
A_{22}-A_{25}	48.679 0	18.676 0	35.292 0	32.519 0
结构总重量（kg）	297.020 0	284.490 0	280.490 0	269.460 0

3.4　算法迭代曲线对比

算例 1 为无约束优化问题,算例 2 为含约束优化问题,通过以上 2 种经典算例模型进行优化对比,由表 5 和表 7 的优化结果可知改进后算法 LWPA 在优化程度和精度方面表现更加良好,达到减轻重量目的;通过图 5 的 4 种算法迭代曲线,在初始条件相同情况下, LWPA 算法相较于其它算法在迭代初期的迭代速度更快、全局搜索能力更强,以较快迭代速度寻找质量较好的全局最优解,在寻优迭代过程中算法表现稳定,验证了 LWPA 有较高收敛速度和精度,具有其独特优越性。

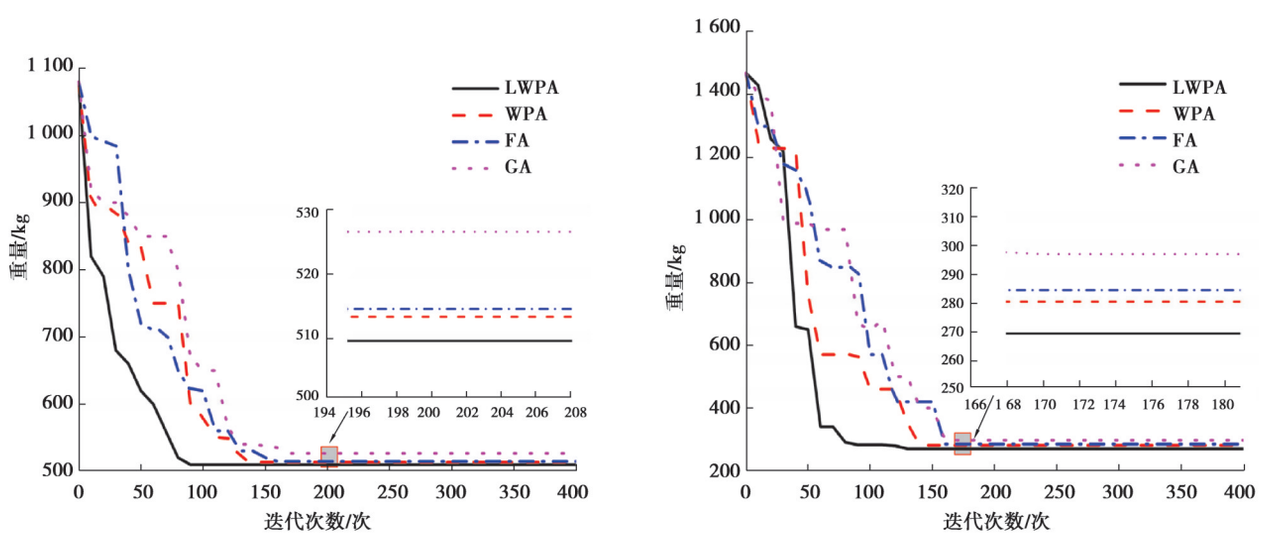


图 5　2个算例的寻优迭代曲线示意图

Fig. 5　The optimization iteration curve of two examples

4　结　论

笔者在基本狼群算法上引入自适应性步长和莱维飞行搜索策略,避免探狼游走过于盲目,使算法能够在搜索后期扩大搜索范围,避免陷入局部收敛,在提高收敛精度的同时能较好提高收敛速度,达到改进目的。

通过仿真实验和方法对比，验证了改进狼群算法的可行性、有效性，并将其运用于桁架结构的优化中，优化结果表明改进后的狼群算法达到了预期重量最轻的目的。该方法可用于求解组合优化问题。虽然对算法的改进有一定成效，但实际工程中的问题复杂多变，今后的研究重点是如何将改进的狼群算法解决更加复杂的工程优化问题。

参考文献

[1] Beheshti Z, Shamsuddin S M. Non-parametric particle swarm optimization for global optimization[J]. Applied Soft Computing, 2015, 28(1): 345-359.

[2] Sudholt D. Theory of swarm intelligence[C]//Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. July 7 - 11, 2012, Philadelphia, Pennsylvania, USA. New York: ACM, 2012: 1215-1238.

[3] Li W J, Bi Y Z, Zhu X F, et al. Hybrid swarm intelligent parallel algorithm research based on multi-core clusters[J]. Microprocessors and Microsystems, 2016, 47(3): 151-160.

[4] Ma L B, Zhu Y L, Zhang D Y, et al. A hybrid approach to artificial bee colony algorithm[J]. Neural Computing and Applications, 2016, 27(2): 387-409.

[5] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Networks. November 27 - December 1, 1995, Perth, WA, Australia: IEEE, 2002: 1942-1948.

[6] Jahangiri M, Ali Hadianfard M, Najafgholipour M A, et al. Interactive autodidactic school: a new metaheuristic optimization algorithm for solving mathematical and structural design optimization problems[J]. Computers & Structures, 2020, 235(1): 106268.

[7] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[8] Xiao L M. An optimizing method based on autonomous animals: fish-swarm algorithm [J].Systems Engineering Theory and practice, 2002,22(11): 32-38.

[9] Krarboga D.An idea based on honey bee swarm for numerical optimization[R]. Kayseri:Erciyes University,2005.

[10] O'Neil M, Woolfe F, Rokhlin V. An algorithm for the rapid evaluation of special function transforms[J]. Applied and Computational Harmonic Analysis, 2010, 28(2): 203-226.

[11] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26(2): 69-74.

[12] Yang X S. Flower pollination algorithm for global optimization[C]//International Conference on Unconventional Computing and Natural Computation. Berlin, Heidelberg: Springer, 2012: 240-249.

[13] Meng X B, Liu Y, Gao X Z, et al. A new bio-inspired algorithm: chicken swarm optimization[C]//International Conference in Swarm Intelligence. Cham: Springer, 2014: 86-94.

[14] 吴虎胜, 张凤鸣, 吴庐山 . 一种新的群体智能算法:狼群算法[J]. 系统工程与电子技术, 2013, 35(11): 2430-2438.
Wu H S, Zhang F M, Wu L S. New swarm intelligence algorithm—wolf pack algorithm[J]. Systems Engineering and Electronics, 2013, 35(11): 2430-2438.(in Chinese)

[15] Yang C G, Tu X Y, Chen J. Algorithm of marriage in honey bees optimization based on the wolf pack search[C]//The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007). October 11-13, 2007. Jeju, Korea (South): IEEE, 2008: 462-467.

[16] Wu C H, Qin K Y, He P H, et al. An improved wolf colony search algorithm based on mutual communication by a sensor perception of wireless networking[J]. EURASIP Journal on Wireless Communications and Networking, 2018(1): 1-10.

[17] Wu H S, Zhang F M. A uncultivated wolf pack algorithm for high-dimensional functions and its application in parameters optimization of PID controller[C]//2014 IEEE Congress on Evolutionary Computation (CEC). July 6-11, 2014. Beijing, China: IEEE, 2014: 1477-1482.

[18] Teng Z J, Lv J L, Guo L W. An improved hybrid grey wolf optimization algorithm[J]. Soft Computing, 2019, 23(15): 6617-6631.

[19] 王盈祥, 陈民铀, 程庭莉, 等 . 基于差分进化的改进狼群算法研究[J]. 计算机应用研究, 2019, 36(8): 2305-2310.

Wang Y X, Chen M Y, Cheng T L, et al. Research of improved wolf pack algorithm based on differential evolution[J]. Application Research of Computers, 2019, 36(8): 2305-2310.(in Chinese)

［20］张悦, 孙惠香, 魏政磊, 等 . 具有自适应调整策略的混沌灰狼优化算法[J]. 计算机科学, 2017, 44(B11): 119-122, 159.
Zhang Y, Sun H X, Wei Z L, et al. Chaotic gray wolf optimization algorithm with adaptive adjustment strategy[J]. Computer Science, 2017, 44(B11): 119-122, 159.(in Chinese)

［21］Kaveh A, Zakian P. Improved GWO algorithm for optimal design of truss structures[J]. Engineering With Computers, 2018, 34(4): 685-707.

［22］莫艳红, 聂慧, 刘振丙, 等 . 基于莱维飞行的灰狼优化算法[J]. 微电子学与计算机, 2019, 36(4): 78-83.
Mo Y H, Nie H, Liu Z B, et al. Grey wolf optimization algorithm based on levy flight[J]. Microelectronics & Computer, 2019, 36(4): 78-83.(in Chinese)

［23］郭立婷 . 基于自适应和变游走方向的改进狼群算法[J]. 浙江大学学报(理学版), 2018, 45(3): 284-293.
Guo L T. Improved wolf pack algorithm based on adaptive step length and adjustable scouting direction[J]. Journal of Zhejiang University (Science Edition), 2018, 45(3): 284-293.(in Chinese)

［24］Tang Q, Shen Y, Hu C Y, et al. Swarm intelligence: based cooperation optimization of multi-modal functions[J]. Cognitive Computation, 2013, 5(1): 48-55.

［25］Parpinelli R S, Teodoro F R, Lopes H S. A comparison of swarm intelligence algorithms for structural engineering optimization [J]. International Journal for Numerical Methods in Engineering, 2012, 91(6): 666-684.

［26］Caamaño P, Bellas F, Becerra J A, et al. Evolutionary algorithm characterization in real parameter optimization problems[J]. Applied Soft Computing, 2013, 13(4): 1902-1921.

［27］Wu J, Jing Z, Li R, et al. A multi-subpopulation PSO immune algorithm and its application on function optimization [J]. Journal of Compute, 2012, 49(9):1883-1898.

［28］Wu H S, Zhang F M. Wolf pack algorithm for unconstrained global optimization[J]. Mathematical Problems in Engineering, 2014, 2014: 1-17.

［29］李彦苍, 王旭 . 基于信息熵的改进海豚群算法及其桁架优化[J]. 重庆大学学报, 2019, 42(5): 76-85.
Li Y C, Wang X. Improved dolphin swarm algorithm based on information entropy and its truss optimization[J]. Journal of Chongqing University, 2019, 42(5): 76-85.(in Chinese)

［30］Bäck T. Introduction[M]//Evolutionary Algorithms in Theory and Practice. Oxford: Oxford University Press, 1996.

（编辑　侯　湘）