

# 一种改进的狼群算法

惠晓滨<sup>1</sup>, 郭 庆<sup>1†</sup>, 吴婷婷<sup>2</sup>, 赵 昱<sup>1</sup>

(1. 空军工程大学 装备管理与安全工程学院, 西安 710051; 2. 94188 部队, 西安 710077)

**摘 要:** 基于狼群算法的基本思想提出探狼更新规则, 并引入相位因子, 以改善探狼的搜索灵活性; 为了提高猛狼的围攻能力, 提出围攻半径的概念, 并给出其计算方法, 其变化可有效体现算法跳出局部最优的能力; 优化了传统狼群算法步长的种类, 并设计新的猛狼位置更新公式. 基于以上改进, 并结合混沌优化思想构建改进狼群算法的计算框架. 通过对测试函数和路径规划问题的仿真模拟, 并与传统狼群算法进行对比, 验证了所提出算法的有效性.

**关键词:** 狼群算法; 探狼更新规则; 围攻半径; 混沌优化

**中图分类号:** TP18      **文献标志码:** A

## An improved wolf pack algorithm

HUI Xiao-bin<sup>1</sup>, GUO Qing<sup>1†</sup>, WU Ping-ping<sup>2</sup>, ZHAO Yu<sup>1</sup>

(1. College of Equipment and Safety Engineering, Air Force Engineering University, Xi'an 710051, China; 2. 94188 Forces, Xi'an 710077, China)

**Abstract:** Based on the wolf pack algorithm, the update rule of scout wolf is proposed, and a phase factor is added to its formula in order to improve the flexibility of scout wolf. To enhance the circumvent ability of ferocious wolf, the concept of siege radius is defined, and its calculation method is given. The change of siege radius can evaluate the ability of escaping from the local optima. The attack step kind is optimized, and a formula new way of position update is designed. On the basis of the above improvements and chaos optimization theory, a framework of the wolf pack algorithm is constructed. The comparing experiment with the traditional wolf pack algorithm on the benchmark functions and path planning problems verifies the effectiveness of the improved algorithm.

**Keywords:** wolf pack algorithm; update rule of scout wolf; siege radius; chaotic optimization

## 0 引 言

Liu 等<sup>[1]</sup> 仿生狼群捕猎活动抽象出搜索、围攻、更新 3 种行为, 提出了狼群算法(WCA); 吴虎胜等<sup>[2]</sup> 分析狼群的协作捕猎活动及猎物分配等特点, 抽象出 3 种人工狼、3 种智能行为(探狼游走行为、头狼召唤行为、猛狼围攻行为)及 2 种智能规则(“胜者为王”的头狼角逐规则、“优胜劣汰”的狼群更新规则), 提出了与 WCA 算法寻优策略不同的狼群算法(WPA), 并基于马尔可夫链理论证明了算法收敛性. WPA 算法是基于狼群集群智慧提出的一种新的智能群优化算法<sup>[2]</sup>. 由于该算法在全局搜索和局部开发能力上具有良好的表现, 自提出就不断地引起国内外学者的关注, 并被迅速应用到实际工程当中. 伊廷华等<sup>[3]</sup> 利用狼群算法解决传感器优化布置问题; 吴虎胜等<sup>[4-5]</sup> 利

用狼群算法解决二进制背包问题和 TSP 问题; 方严军等<sup>[6]</sup> 利用狼群算法解决小车存取系统三维空间路径优化问题; 王建群等<sup>[7]</sup> 利用狼群算法解决水电站水库优化调度问题; 刘永兰等<sup>[8]</sup> 利用狼群算法解决无人机路径规划问题.

WPA 算法以系统思想、职责分工协作思想、“由粗到细”渐进思想和导向随机思想为基本框架, 这与粒子群优化算法(PSO)、蚁群优化算法(ACO)、遗传算法(GA)、人工蜂群算法(ABC)、人工鱼群算法(AFSA)等智能仿生优化算法的寻优机制有所不同. 文献[2]表明该算法对多峰、高维复杂函数的处理具有显著的效果. 然而, 该算法仍存在一些不足, 诸如探狼的过度贪婪及“死板”的游走方式易使探狼陷入局部最优的尴尬位置、猛狼的后期围攻效果不理想等.

收稿日期: 2016-06-01; 修回日期: 2016-09-05.

基金项目: 国家自然科学基金青年基金项目(71501184).

作者简介: 惠晓滨(1976—), 男, 教授, 博士生导师, 从事复杂系统建模与仿真、数据分析与智能决策等研究; 郭庆(1994—), 男, 硕士生, 从事信息系统工程与智能计算的研究.

†通讯作者. E-mail: 781106222@qq.com

本文在 WPA 算法的基础上进行改进,在游走行为中提出探狼更新规则,设计“由粗到细”的游走机制,并向游走公式中引入随机的相位因子,提高探狼的游走灵活性,改善算法的全局搜索能力;在围攻行为中取消判定距离参数,提出围攻半径的概念,给出基于部分人工狼位置信息的计算方法,并依此判定人工狼是否转入围攻行为.最后,结合反向学习<sup>[9]</sup>和混沌优化<sup>[10]</sup>思想系统构建了一种改进的狼群算法(GWPA)计算框架.仿真计算结果表明了本文改进狼群算法的有效性.

## 1 狼群算法的基本思想及优缺点

狼群算法的主体是由探狼游走、头狼召唤、猛狼围攻3种智能行为和“胜者为王”的头狼角逐规则以及“优胜劣汰”的狼群更新规则构成.以求解函数最大值为例,简要叙述狼群算法的求解过程.

**Step 1:** 在解空间中随机初始化狼群的空间坐标,依据目标函数值的大小角逐出人工头狼.

**Step 2:** 探狼开始随机游走搜索猎物,若发现某个位置的目标函数值大于头狼的目标函数值,将更新头狼位置,同时头狼发出召唤行为;若没有发现,探狼将继续游走直到达到最大游走次数,头狼在原本的位置发出召唤行为.

**Step 3:** 听到头狼召唤的猛狼以较大的步长快速向头狼奔袭,若奔袭途中猛狼的目标函数值大于头狼的目标函数值,则将对头狼位置进行更新;否则,猛狼将继续奔袭直到进入围攻范围.

**Step 4:** 靠近头狼的猛狼将联合探狼对猎物(把头狼位置视为猎物)进行围捕,围捕过程中若其他人工狼的目标函数值大于头狼的目标函数值,则对头狼位置进行更新,直到捕获猎物.

**Step 5:** 淘汰狼群中目标函数值较小的人工狼,并在解空间中随机生成新的人工狼,实现狼群的更新.

**Step 6:** 最后判断头狼的目标函数值是否达到精度要求或算法是否达到最大迭代次数,倘若没有达到,则从探狼游走开始继续迭代直到达到为止;若达到,则输出头狼的空间坐标以及目标函数值,即为被求函数的最优位置和最优解.

狼群算法中涉及到游走步长  $\text{step}_a$ , 奔袭步长  $\text{step}_b$ , 攻击步长  $\text{step}_c$ , 在第  $d$  维度计算采用如下公式:

$$\text{step}_a^d = \text{step}_b^d / 2 = 2 \times \text{step}_c^d = |\max_d - \min_d| / S. \quad (1)$$

其中:  $\max_d$ 、 $\min_d$  分别表示狼群在第  $d$  维坐标的最大值和最小值;  $S$  表示步长因子.

在狼群算法中,探狼的游走行为集中体现了算法

的全局搜索能力,猛狼的围攻行为集中体现了算法的局部开发能力,而“优胜劣汰”的狼群更新机制保证了人工狼的多样性,从而避免算法陷入局部最优解,“胜者为王”的头狼产生机制则是引导着整个狼群算法的求解过程向着有利的方向进行.由于狼群算法在求解策略上能够较好地兼顾对解空间的全局搜索和局部开发,因而狼群算法具有优良的寻优性能.然而,算法在游走行为和围攻行为中仍存在一些不足,需要进一步改进.

不足1: 在游走行为中,探狼采取贪婪式游走策略,即探狼始终朝着比当前更优的位置前进.当探狼的最大游走次数  $T_{\max}$  设定的较大时,易使探狼因贪婪过度而陷入局部最优的位置,从而无法继续开辟探索空间,降低算法的全局搜索能力.图1所示是应用 WPA 求解多峰函数 Eggcrate 的最小值时,在第30次迭代后的探狼连续3次游走的分布图.

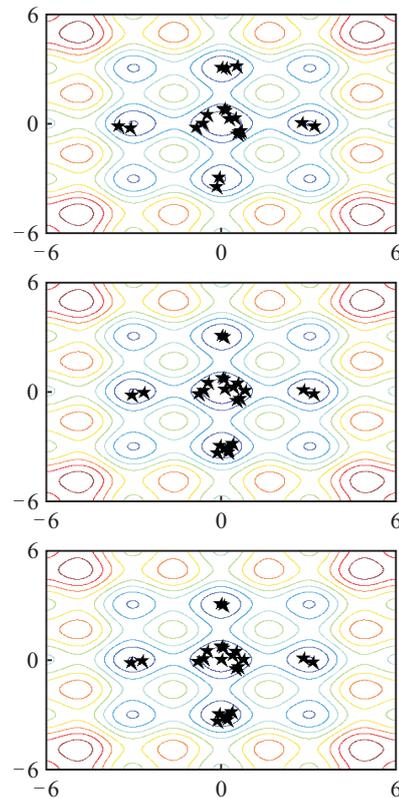


图1 第30次迭代后的探狼游走分布图

从图1可以看出,已有半数的探狼陷入到全局次优、局部最优的尴尬位置.另外,从对探狼游走(见下式)的分析可以看出,一旦参数  $h$  相同,探狼的搜索方向就相同,这削弱了算法的随机性.

$$x_{id}^p = x_{id} + \sin(2\pi \times p/h) \cdot \text{step}_a^d. \quad (2)$$

其中:  $p$  为方向,  $p = 1, 2, \dots, h$ ;  $d$  为维度,  $d = 1, 2, \dots, D$ ;  $i = 1, 2, \dots, N - 1$ ;  $x_{id}$  为探狼  $i$  在第  $d$  维的坐标;  $x_{id}^p$  为探狼  $i$  探索第  $p$  个方向在第  $d$  维的坐标.

不足2:狼群算法在运行前期收敛速度较快,而算法后期收敛速度有明显的下降.这是由猛狼的后期围攻效果欠佳所导致.猛狼的围攻行为是在围攻区域内进行的,围攻区域是由判定距离 $d_{near}$ 限定的,而 $d_{near}$ 随着猛狼围攻的进行不断变化,其大小间接反映了猛狼的围攻强度.图2所示是应用WPA在求解多峰函数Eggcrate的全局最小值的过程中 $d_{near}$ 的变化曲线.

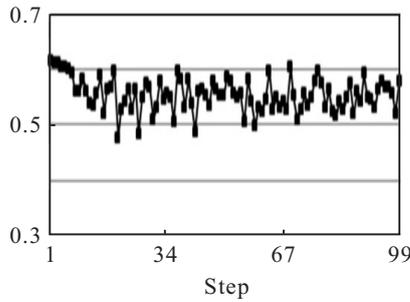


图2 判定距离 $d_{near}$ 的过程曲线

从图2可以看出,计算前期 $d_{near}$ 在不断减小,而后期 $d_{near}$ 一直在0.5~0.6之间波动,整体上没有减小的趋势.

不足3:狼群算法中,游走步长 $step_a$ 的大小反映算法的全局搜索细度,攻击步长 $step_c$ 的大小反映算法的局部开发细度.在实际的计算中,需要针对不同的求解对象调整算法的全局搜索能力与局部开发能力,使得算法具有较优的运算效率.然而,从式(1)可以看出,游走步长与攻击步长之间存在一定的耦合关系,不能分别独立设定,这削弱了算法对全局搜索能力与局部开发能力的统筹.

## 2 算法的改进

### 2.1 改进策略

针对狼群算法的3点不足,提出如下几点改进(以求解最大值为例).

改进1.在游走行为中,引入探狼更新规则并设计“由粗到细”的游走机制.其基本思想如下:在探狼 $i$ 探索周围 $h$ 个方向中,若 $h$ 个方向的函数值都小于探狼 $i$ 的目标函数值,则在探狼 $i$ 周围很可能存在极大值,这时缩短搜索距离(游走步长),再次搜索周围 $h$ 个方向,若 $h$ 个方向的函数值仍都小于探狼 $i$ 的目标函数值,则继续缩短搜索距离;假如经过多次缩短搜索距离后,探狼 $i$ 周围 $h$ 个方向的函数值仍都小于探狼 $i$ 的目标函数值,则可以认为探狼 $i$ 陷入局部最优位置,此时的探狼 $i$ 需要重新初始位置,再执行游走行为.该设计思路进一步体现了狼群算法“由粗到细”的渐近思想.另外,向探狼游走(式(2))中引入相位因子,得到

新公式如下:

$$x_{id}^p = x_{id} + (1 - k/k_{max} + \eta) \times \sin(2\pi \times p/h + \theta) \cdot step_a^d. \quad (3)$$

其中: $k$ 为迭代次数, $k = 1, 2, \dots, k_{max}$ ;  $\eta \in (-0.1, 0.1)$ ;  $\theta \in (0, p/h)$ ;  $h \in [4, 7]$ 为整数.  $(1 - k/k_{max} + \eta)$ 的设计同样体现了探狼游走“由粗到细”的渐近思想,而相位因子的引入改善了探狼搜索的灵活性、随机性和遍历性.

游走行为改进后的伪代码如下.

Code:改进的游走行为, floatArtificial\_Wolf : AW\_ScoutWolf().

Input:  $X, Y$ ;

参数:  $h, D, N, T_{max}$ .

While ( $T < T_{max}$ )

for  $i = 1, 2, \dots, N - 1$

if  $Y_i > Y_{lead}$  then

$X_{lead} = X_i, Y_{lead} = Y_i$ ;

AW\_Summon(); %进入召唤行为

break;

else

for  $p = 1, 2, \dots, h$

for  $d = 1, 2, \dots, D$

$x_{id}^p = x_{id} + (1 - k/k_{max} + \eta) \times \sin(2\pi \times p/h + \theta) \cdot step_a^d$ ;

end for

$Y_{ip} = f(x_i^p)$

end for

$Y_{ip*} = \max\{Y_{ip} | p = 1, 2, \dots, h\}$ ;

if  $Y_{ip*} > Y_i$  then

$X_i = X_i^{p*}, Y_i = Y_{ip*}$ ;

$T = T + 1, failure_i = 0$ ;

else

$step_a = step_a / 2$ ; %步长减半

$failure_i = failure_i + 1$  %计算探狼连续被困次数

困次数

end if

end if

if  $failure_i > T_{max} / 10$  then

初始化头狼 $i$ 的位置; %探狼死亡

end if

end for

end While

定义1 在解空间中,可执行围攻行为的猛狼到猎物(头狼)的最远距离称为围攻半径,用 $L_{near}$ 表示.

定义2 在解空间中,以猎物(头狼)的位置为中心点,以 $L_{near}$ 为邻域半径所构成的去心邻域称为围

攻区域.

改进2. 取消判定距离, 提出围攻半径的概念, 并给出利用部分人工狼的位置信息计算围攻半径  $L_{near}$  的方法. 在狼群算法中, 头狼即视为猎物, 召唤行为与围攻行为促使部分人工狼会不断地向头狼靠近, 而“优胜劣汰”的狼群更新规则又使得另一部分人工狼在远离头狼的位置开辟新的探索空间. 假设人工狼的总数为  $N$ , 每次更新人工狼的数量为  $R$ , 那么在算法中至少会有  $(N - R - 1)$  匹人工狼是不断向头狼靠近的. 因此, 设计如下公式计算围攻半径  $L_{near}$ :

$$L_{near} = \sum_{i=1}^{NL} \sum_{d=1}^D |x_i^d - x_{lead}^d| / NL \times \omega, \quad (4)$$

$$NL = \lfloor N \times (1 - 1/\beta) \rfloor. \quad (5)$$

其中:  $\beta$  为狼群更新比例因子,  $x_i$  为距离头狼最近的人工狼的空间位置,  $\omega$  为距离判定因子. 距离判定因子  $\omega$  的大小影响算法的前期收敛速度, 通常  $\omega$  越大, 算法前期收敛速度越快. 需要注意的是, 当  $\omega$  过大时, 会导致起始的  $L_{near}$  过小, 参与围捕的人工狼会很难进入围攻圈, 反而降低算法的前期收敛速度.

改进3. 为了消除游走步长与攻击步长的耦合关系, 同时保证不增加额外的设置参数, 本文取消了围攻行为中的攻击步长  $step_c$ , 设计如下公式对参与围攻的人工狼进行位置更新:

$$x_{id}^{k+1} = x_{id}^k + \lambda \times (g_d^k - x_{id}^k). \quad (6)$$

其中:  $g_d^k$  为第  $k$  代头狼在第  $d$  个维度的坐标位置;  $\lambda$  为  $(0, 2)$  之间的随机数, 该选择可以保证猛狼的围攻行为在围攻区域内进行, 并且可以攻击到猎物的后方.

最后, 结合反向学习和混沌优化思想设计反向双混沌更新策略, 其具体步骤如下.

**Step 1:** 用 Sinusoidal iterator 和 Gauss map 两个一维混沌映射方程按等概率产生  $D$  个长度为  $M$  的混沌序列  $CX = \{cx_1, cx_2, \dots, cx_D\}$ ,  $cx_d = \{cx_{1d}, cx_{2d}, \dots, cx_{Md}\}^T$ ,  $d = 1, 2, \dots, D$ ,  $M$  为需要更新/初始化人工狼的数量. 下式分别为 Sinusoidal iterator 和 Gauss map 的一维映射表达式:

$$x_{k+1} = a \cdot x_k \sin(\pi x_k), \quad x_k \in (0, 1); \quad (7)$$

$$x_{k+1} = \begin{cases} 0, & x_k = 0; \\ 1/x_k - \lfloor 1/x_k \rfloor, & x_k \in (0, 1). \end{cases} \quad (8)$$

**Step 2:** 依据下式将混沌序列映射到解空间中, 获得初始化/更新狼群  $X = \{x_1, x_2, \dots, x_M\}^T$ :

$$x_{id} = L_d + cx_{id}(U_d - L_d). \quad (9)$$

**Step 3:** 依据下式算出狼群  $X$  的反向狼群  $OX = \{ox_1, ox_2, \dots, ox_M\}^T$ .

$$ox_{id} = L_d + U_d - x_{id}. \quad (10)$$

其中:  $L_d$  为解空间  $d$  维的下限,  $U_d$  为解空间第  $d$  维的上限.

**Step 4:** 计算  $X$  和  $OX$  的目标函数值, 选取其中目标函数值最大的  $M$  个人工狼作为初始化/更新的人工狼.

## 2.2 基本流程

改进狼群算法(GWPA)的基本流程如下(以求解最大值为例).

**Step 1:** 初始化. 初始化人工狼的数目  $N$ , 最大迭代次数  $k_{max}$ , 最大游走次数  $T_{max}$ , 距离判定因子  $\omega$ , 步长因子  $S$ , 更新比例因子  $\beta$ , 并按反向双混沌更新策略初始化狼群的空间位置  $X_i$ .

**Step 2:** 游走行为. 选取目标函数值最大的人工狼为头狼, 其余的人工狼均视为探狼并按式(3)进行游走(探狼  $i$  游走期间, 若探狼  $i$  周围  $h$  个方向的函数值都小于探狼  $i$  的目标函数值, 则游走步长  $step_a$  减半, 若游走步长连续经过多次减半后, 探狼仍无法前进, 则认为探狼  $i$  陷入局部最优, 应以反向双混沌更新策略更新探狼  $i$  的空间位置), 直到其中某只探狼  $i$  的目标函数值  $Y_i$  大于头狼的目标函数值  $Y_{lead}$  或探狼的游走次数达到最大游走次数  $T_{max}$ , 转入 Step 3.

**Step 3:** 召唤行为. 头狼发出召唤, 猛狼(除头狼外的所有人工狼)以下式快速向头狼奔袭:

$$x_{id}^{k+1} = x_{id}^k + step_b^d \cdot (g_d^k - x_{id}^k) / |g_d^k - x_{id}^k|. \quad (11)$$

在奔袭的途中, 若猛狼  $i$  的目标函数值  $Y_i > Y_{lead}$ , 则转到 Step 2 的末尾; 若  $Y_i \leq Y_{lead}$ , 则猛狼  $i$  继续奔袭直到进入围攻区域, 即猛狼  $i$  与头狼的距离  $L_i \leq L_{near}$ , 转入 Step 4.

**Step 4:** 围攻行为. 把头狼位置视为猎物的移动位置, 参与围攻的猛狼按式(6)对猎物进行围攻.

**Step 5:** 狼群更新. 将本次迭代产生最优狼的目标函数与上次迭代头狼的目标函数值进行对比, 若更大, 则更新头狼位置; 反之, 则记录头狼未更新次数  $t$ . 依据更新比例因子  $\beta$ , 确定淘汰目标函数值较小的人工狼的数量  $R$ ,  $R \in (N/2 \cdot \beta, N/\beta)$ . 若  $t$  小于限值  $t_{max}$ , 则按照下式并结合反向学习更新狼群; 反之, 将按照反向双混沌策略更新狼群:

$$x_{id} = g_d \cdot [\sin(\gamma) + 1], \quad (12)$$

其中  $\gamma \in (-0.1, 0.1)$ .

**Step 6:** 判断终止. 判断头狼的目标函数值是否达到计算精度要求, 或算法是否达到最大迭代次数  $k_{max}$ , 若达到, 则输出头狼位置及目标函数值, 否则转入 Step 2.

改进狼群算法的基本流程见图3.

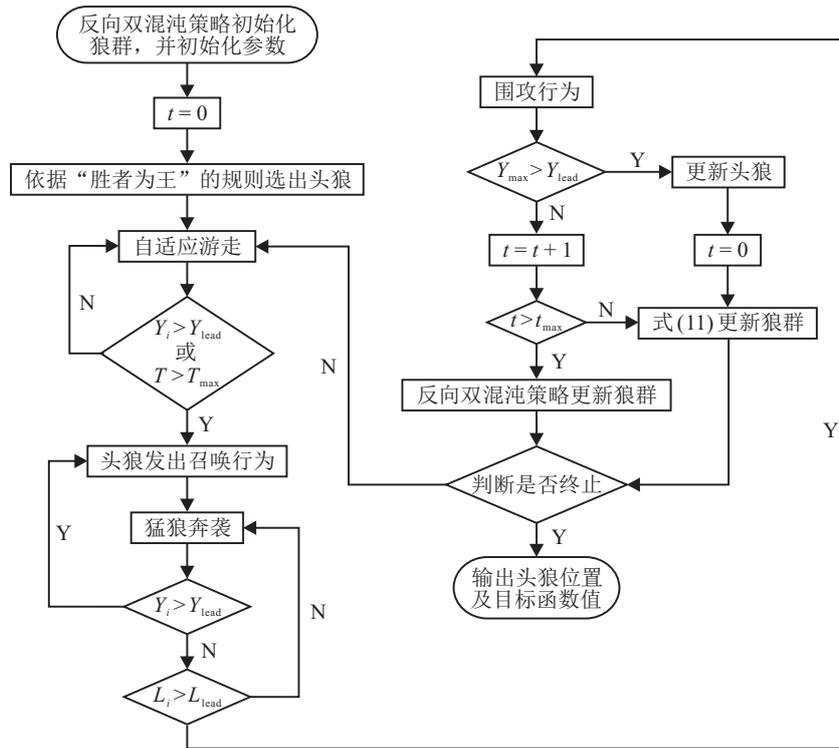


图3 改进狼群算法的基本流程图

### 3 实验与分析

为了验证改进狼群算法的有效性,选取14个标准测试函数测试改进算法的性能,并对机器人路径规划问题进行仿真实验,测试改进算法的工程实用性.

#### 3.1 函数优化仿真

选取的14种标准测试函数<sup>[1,2,11-14]</sup>的基本信息如表1所示.其中:U为单峰,M为多峰,S为可分,N为不可分.

表1 14个标准测试函数

函数	函数名	函数表达式	特征	解空间	全局极值
$F_1$	Easom	$f(X) = -\cos(x) \cos(y) \cdot e^{-(x-\pi)^2 - (y-\pi)^2}$	UN	$[-10, 10]^2$	min = -1
$F_2$	Matyas	$f(X) = 0.26(x^2 + y^2) - 0.48xy$	UN	$[-10, 10]^2$	min = 0
$F_3$	Rosenbrock	$f(X) = 100(y - x^2)^2 + (1 - x)^2$	UN	$[-2.048, 2.048]^2$	min = 0
$F_4$	Rastrigin	$f(X) = 20 + [x^2 - 10 \cos(2\pi x)] + [y^2 - 10 \cos(2\pi y)]$	MN	$[-5.12, 5.12]^2$	min = 0
$F_5$	Sumsquares	$f(X) = \sum_{i=1}^D ix_i^2$	US	$[-10, 10]^D$	min = 0
$F_6$	Sphere	$f(X) = \sum_{i=1}^D x_i^2$	US	$[-10, 10]^D$	min = 0
$F_7$	Bohachevsky1	$f(X) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7$	MS	$[-10, 10]^2$	min = 0
$F_8$	Quadric	$f(X) = \sum_{i=1}^D \left( \sum_{k=1}^i x_k \right)^2$	MS	$[-30, 30]^D$	min = 0
$F_9$	Eggcrate	$f(X) = x^2 + y^2 + 25(\sin^2 x + \sin^2 y)$	MN	$[-2\pi, 2\pi]^2$	min = 0
$F_{10}$	SixHump-CamelBack	$f(X) = 4x^2 - 2.1x^4 + x^6/3 + xy - 4y^2 + 4y^4$	MN	$[-5, 5]^2$	min = -1.03 163
$F_{11}$	Bohachevsky3	$f(X) = x^2 + 2y^2 - 0.3 \cos(3\pi x + 4\pi y) + 0.3$	MN	$[-100, 100]^2$	min = 0
$F_{12}$	Bridge	$f(X) = \sin \sqrt{x^2 + y^2} / \sqrt{x^2 + y^2} + e^{(\cos(2\pi x) + \cos(3\pi y))/2} - 0.7129$	MN	$[-1.5, 1.5]^2$	max = 3.0054
$F_{13}$	Griewank	$f(X) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i / \sqrt{i})$	MN	$[-600, 600]^D$	min = 0
$F_{14}$	Ackley	$f(X) = -20e^{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}} - e \sum_{i=1}^D \cos(2\pi x_i) / D + 20 + e$	MN	$[-32, 32]^D$	min = 0

为了体现仿真实验的有效性,对测试函数分别进行50次仿真计算实验,对计算后的数据进行统计分析,从最优值、平均值、标准差、平均消耗时间(默认单位为s)等统计特性对算法效能进行评估.根据文献[2, 15-16]对算法的参数进行设定:狼群规模 $N$ 取

100;探狼最大游走次数 $T_{max}$ 取20;狼群更新比例因子 $\beta$ 取5;距离判定因子 $\omega$ 取20;其余参数的设定如表2所示(考虑到多维函数的计算复杂度要远大于2维函数,因此其最大迭代次数选取的大一些).

表2 参数设定

算法	参数设定
WPA	$S = 1\ 000$ ;对于2维函数, $k_{max} = 100$ ,对于多维函数, $k_{max} = 500$ .
IWPA/CWPA/OWPA	$S = 0.008$ ;对于2维函数, $k_{max} = 100$ ,对于多维函数, $k_{max} = 500$ .
OCWPA/GWPA	$S = 0.008$ ; $k_{max} = 100$ .

表3 测试结果对比

函数	维数	特征	全局极值	评价标准	WPA	IWPA	CWPA	OWPA	OCWPA	GWPA
$F_1$	$D = 2$	UN	min = -1	最优值	-0.999 994	-1	-1.000 000	-1	-1	-1
				平均值	-0.994 029	-1	-0.999 997	-0.999 996	-1	-1
				平均耗时	0.46	0.52	0.54	0.45	0.61	0.55
$F_2$	$D = 2$	UN	min = 0	最优值	3.56e-08	5.71e-20	2.14e-18	5.03e-21	1.90e-189	1.20e-261
				平均值	2.90e-04	4.19e-13	4.27e-12	4.87e-17	1.16e-76	9.74e-213
				平均耗时	0.37	0.51	0.51	0.51	0.59	0.57
$F_3$	$D = 2$	UN	min = 0	最优值	2.50e-06	1.87e-09	4.00e-09	4.10e-13	7.93e-12	1.08e-14
				平均值	2.14e-04	5.56e-08	5.73e-08	5.71e-11	1.64e-08	3.15e-11
				平均耗时	0.48	0.46	0.46	0.41	0.56	0.55
$F_4$	$D = 2$	MN	min = 0	最优值	2.23e-02	4.22e-12	3.32e-13	1.07e-14	0.00e+00	0.00e+00
				平均值	6.56e-01	1.19e-11	1.49e-12	1.89e-13	0.00e+00	0.00e+00
				平均耗时	0.38	0.50	0.58	0.52	0.65	0.71
$F_5$	$D = 20$	US	min = 0	最优值	1.35e-03	6.52e-10	2.40e-10	1.19e-13	1.34e-57	6.28e-135
				平均值	9.23e-03	1.58e-09	1.71e-09	5.91e-13	1.50e-48	6.26e-113
				平均耗时	12.31	5.71	8.07	5.73	1.69	1.60
$F_6$	$D = 20$	US	min = 0	最优值	3.38e-03	2.25e-09	9.85e-10	8.63e-14	5.93e-64	7.70e-128
				平均值	6.03e-03	7.80e-09	1.00e-08	4.64e-13	1.21e-49	2.76e-114
				平均耗时	15.97	5.50	7.74	5.57	1.65	1.43
$F_7$	$D = 2$	MS	min = 0	最优值	2.24e-04	4.13e-10	2.11e-10	4.13e-13	0.00e+00	0.00e+00
				平均值	5.65e-04	3.12e-09	1.39e-09	1.20e-12	0.00e+00	0.00e+00
				平均耗时	0.41	0.52	0.63	0.55	0.61	0.66
$F_8$	$D = 20$	MS	min = 0	最优值	1.40e-03	6.25e-07	2.70e-07	9.95e-10	5.53e-35	9.11e-110
				平均值	2.28e-03	2.40e-06	1.49e-06	4.51e-09	1.25e-20	9.16e-90
				平均耗时	11.88	5.74	8.10	5.83	1.75	1.54
$F_9$	$D = 2$	MN	min = 0	最优值	4.27e-04	1.63e-14	2.26e-15	1.83e-17	1.45e-195	5.61e-294
				平均值	6.16e-02	8.73e-11	2.15e-11	3.37e-15	5.56e-151	7.17e-256
				平均耗时	0.42	0.54	0.68	0.54	0.68	0.65
$F_{10}$	$D = 2$	MN	min = -1.031 63	最优值	-1.031 627	-1.031 628	-1.0316 28	-1.031 628	-1.031 628	-1.031 628
				平均值	-1.031 379	-0.990 820	-1.031 628	-0.990 820	-1.031 628	-1.031 628
				平均耗时	0.44	0.60	0.74	0.54	0.68	0.84
$F_{11}$	$D = 2$	MN	min = 0	最优值	1.14e-02	2.76e-13	1.78e-11	1.61e-15	0.00e+00	0.00e+00
				平均值	2.09e-01	6.79e-12	7.92e-10	3.17e-14	0.00e+00	0.00e+00
				平均耗时	0.40	0.57	0.72	0.54	0.58	0.61
$F_{12}$	$D = 2$	MN	max = 3.005 4	最优值	3.005 381	3.005 382	3.005 382	3.005 382	3.005 382	3.005 382
				平均值	2.996 757	2.981 729	2.981 729	2.982 489	3.005 382	3.005 382
				平均耗时	0.46	0.73	0.74	0.70	0.74	0.81
$F_{13}$	$D = 20$	MN	min = 0	最优值	5.24e-04	2.02e-08	1.90e-08	4.75e-13	0.00e+00	0.00e+00
				平均值	3.13e-02	4.41e-07	2.00e-07	3.86e-12	0.00e+00	0.00e+00
				平均耗时	11.40	7.19	9.51	7.45	1.89	1.56
$F_{14}$	$D = 100$	MN	min = 0	最优值	7.92e-01	1.85e-05	1.30e-06	2.57e-09	3.55e-15	0.00e+00
				平均值	8.54e-01	1.90e-05	1.86e-05	3.78e-09	2.18e-14	0.00e+00
				平均耗时	57.75	20.12	27.73	21.62	6.62	3.29

3.1.1 算法对比分析

表3所示是GWPA与几种传统的狼群算法对14个测试函数仿真的计算对比结果.图4所示是求解Eggcrate( $F_9$ )函数的进化曲线.从仿真结果可以看出,狼群算法能够搜寻到全局最优值,并且具有较高的计算精度.这也间接说明狼群算法具有较好的全局搜索能力和局部开发能力.

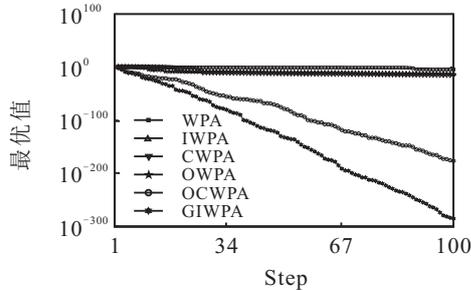


图4 求解  $F_9$  函数的收敛图

表3中, GWPA与传统的狼群算法相比具有以下特点:在2维函数求解中,改进的狼群算法的收敛速度(图4)和计算精度得以明显提高,计算时间的缩短却不是很明显;在多维复杂函数求解中, GWPA的收敛速度、计算精度的改善更加明显,并且其运算时间最少也缩短了50%.

对于高维函数,随着维度的增加,问题求解复杂度会呈指数式增长,即计算中的“维数灾难”.许多智能算法对2维、低维函数的计算效果很好,而对高维函数的计算却表现不佳.而狼群算法不仅对低维函数的计算表现良好,对高维复杂函数的计算效果更佳.为了进一步测试狼群算法对高维函数的计算优势,以Sumsquares( $F_5$ )、Sphere( $F_6$ )、Quadric( $F_8$ )、Griewank( $F_{13}$ )4个函数为例,分别取函数维度  $D$  为50、100、200进行仿真计算.仿真结果如表4所示.

表4 高维复杂函数计算对比

维数	函数	$F_5$					$F_6$				
	算法	IWPA	CWPA	OWPA	OCWPA	GWPA	IWPA	CWPA	OWPA	OCWPA	GWPA
$D = 50$	最优值	5.51e-05	5.86e-05	2.93e-08	1.89e-40	6.11e-104	8.14e-05	1.04e-04	1.46e-08	4.13e-41	7.62e-95
	平均值	1.09e-04	9.78e-05	7.25e-08	1.39e-35	1.23e-94	2.06e-04	1.99e-04	2.61e-08	2.82e-34	3.96e-83
	平均耗时	11.81	11.73	11.70	4.11	2.03	10.91	10.91	11.54	4.03	1.81
$D = 100$	最优值	3.10e-02	3.30e-02	2.40e-05	3.08e-30	1.92e-90	2.01e-03	1.56e-02	8.93e-06	3.51e-32	1.02e-81
	平均值	6.00e-02	5.94e-02	4.33e-05	3.56e-25	1.09e-77	4.58e-03	4.25e-02	1.53e-05	1.41e-24	9.52e-72
	平均耗时	18.89	18.87	19.98	6.98	3.25	18.31	18.27	19.69	7.4	2.68
$D = 200$	最优值	3.03e-03	4.41e-03	2.60e-06	7.32e-30	4.97e-190	1.33e-03	2.07e-03	5.56e-07	1.65e-29	7.29e-179
	平均值	5.55e-03	5.96e-03	4.13e-06	1.32e-23	1.83e-173	2.71e-03	2.91e-03	7.90e-07	2.36e-24	8.71e-166
	平均耗时	71.53	70.36	74.33	30.01	10.84	68.49	68.34	73.24	29.87	9.83
维数	函数	$F_8$					$F_{13}$				
	算法	IWPA	CWPA	OWPA	OCWPA	GWPA	IWPA	CWPA	OWPA	OCWPA	GWPA
$D = 50$	最优值	1.07e-09	3.33e-10	1.42e-13	5.84e-55	6.72e-179	7.15e-12	7.50e-12	4.22e-15	0.00e+00	0.00e+00
	平均值	4.39e-09	2.06e-09	1.52e-12	4.54e-10	4.13e-135	9.10e-12	1.26e-11	6.61e-15	0.00e+00	0.00e+00
	平均耗时	35.81	33.28	34.84	5.6	4.16	27.78	24.45	24.24	3.76	2.2
$D = 100$	最优值	1.58e-04	4.92e-04	1.18e-06	2.48e-13	2.95e-149	1.37e-07	1.66e-07	1.00e-11	0.00e+00	0.00e+00
	平均值	4.45e-04	1.21e-04	4.62e-06	7.41e-07	2.06e-103	2.07e-07	2.50e-07	4.06e-11	0.00e+00	0.00e+00
	平均耗时	61.85	59.69	61.00	10.19	5.10	48.56	41.50	41.15	6.81	3.05
$D = 200$	最优值	4.33e-01	5.84e-02	1.66e-03	1.65e-06	6.55e-65	5.04e-04	3.49e-04	6.29e-08	0.00e+00	0.00e+00
	平均值	6.50e-01	9.91e-01	4.95e-02	2.97e-05	1.74e-44	7.69e-04	7.97e-04	6.55e-08	0.00e+00	0.00e+00
	平均耗时	106.69	107.8	12178.64	20.17	7.4	89.86	75.54	74.08	12.85	5.15

从表4的对比结果来看,与低维函数仿真实验的计算结果相比,改进狼群算法(GWPA)在针对高维函数的仿真计算上性能更佳.算法计算平均耗时缩短了50%以上,同时计算结果精度也提升了至少50%.这进一步证实了改进狼群算法对高维复杂函数的求解优势.

为了证实围攻半径的引入价值,分别以单峰函数Easom( $F_1$ )、多峰函数Eggcrate( $F_9$ )为例,对GWPA的

围攻半径  $L_{near}$  和WPA的判定距离  $d_{near}$  的变化进行展示,其仿真过程变化曲线如图5和图6所示.

图5是求解单峰函数  $F_1$  的变化曲线,从图中可以看出:随着算法的运行,改进狼群算法的围攻半径在不断地减小,最终趋向于零;而基本狼群算法的判定距离在0.8 ~ 1.0之间波动,减小趋势不明显.图6是求解多峰复杂函数  $F_9$  的变化曲线,从中也可以看出改进狼群算法的围攻半径不断减小并趋向于零.与

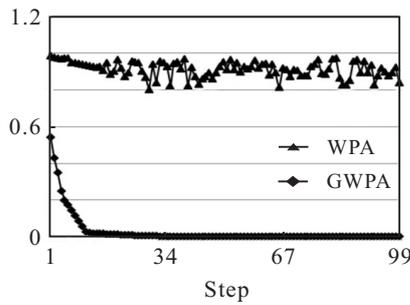


图5  $F_1$  函数的围攻半径过程曲线

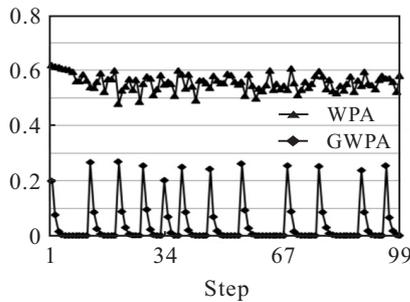


图6  $F_9$  函数的围攻半径过程曲线

图5不同的是,图6的变化曲线似乎存在某种周期特性,其原因在于 $F_9$ 是多峰函数,在寻优过程中需要多次跳出局部最优解才能找到全局最优值.图6中极值的数量也就是改进狼群算法在计算中跳出局部最优的次数,这不仅说明改进狼群算法能够顺利、多次跳出局部最优,具有较强的围攻特性(局部开发能力),而且还可以用围攻半径 $L_{near}$ 的变化曲线来量化算法跳出局部最优的能力.在图4中,可以看出改进狼群算法在运算后期的计算精度仍在继续攀升,其原因在于围攻半径 $L_{near}$ 随着算法的运行在不断地减小.这印证了算法引入围攻半径的价值所在.

### 3.1.2 参数分析

尽管GWPA具有优秀的寻优性能,但也可以看出算法具有较多的参数.通过大量的仿真实验发现GWPA的参数可以在限定的范围内随机选取,具有较好的鲁棒性.为此,以求解多峰函数Bridge(如图7)的最大值为例,分别讨论探狼的搜索方向 $h$ 、狼群更新比例因子 $\beta$ 、距离判定因子 $\omega$ 对GWPA性能的影响.

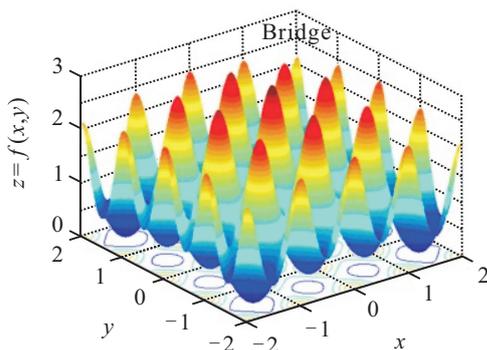


图7 Bridge函数及等高线

首先,讨论探狼的搜索方向 $h$ 对GWPA性能的影响.实验时,狼群规模 $N$ 取20,其余参数不改变.在设定不同的 $h$ 后,分别对Bridge函数进行50次寻优计算,统计结果如表5所示.

表5 搜索方向 $h$ 对GWPA性能的影响

$h$	最差值	平均值	标准差	成功率/%	平均耗时/s
2	3.005 372	3.005 382	9.66e-07	90	0.078
3	3.000 634	3.005 290	6.21e-04	87	0.092
4	3.005 381	3.005 382	1.07e-07	90	0.106
5	3.005 382	3.005 382	2.16e-10	94	0.116
6	3.005 382	3.005 382	1.41e-11	93	0.125
7	3.005 382	3.005 382	4.23e-13	97	0.139
8	3.005 346	3.005 381	3.63e-06	95	0.154
9	3.004 160	3.005 369	1.24e-04	96	0.175
10	3.005 382	3.005 382	3.12e-12	92	0.190

从表5可以看出,随着 $h$ 加大,算法的寻优效果不断改善,主要表现在算法的计算成功率、最差值、平均值不断增加.标准差的不断减小体现了算法的敏感性不断降低.同时, $h$ 加大也导致了消耗时间的延长,考虑到时间与寻优效果,算法中 $h$ 取 $[4, 7]$ 的随机整数是较为合理的.

其次,讨论更新比例因子 $\beta$ 对GWPA性能的影响.同样,狼群规模 $N$ 取20,其余参数不改变.设定不同的 $\beta$ 后,分别对Bridge函数进行50次寻优计算,统计结果如表6所示.

表6 更新比例因子 $\beta$ 对GWPA性能的影响

$\beta$	最差值	平均值	标准差	成功率/%	平均耗时/s
1	2.923 786	2.978 422	3.45e-02	10	0.130
2	2.996 636	3.005 038	1.61e-03	60	0.137
3	3.004 941	3.005 370	7.06e-05	80	0.137
4	3.005 382	3.005 382	2.36e-09	90	0.131
5	3.005 382	3.005 382	6.01e-14	90	0.133
6	3.005 382	3.005 382	4.93e-09	96	0.127
7	3.005 382	3.005 382	2.52e-11	88	0.126
8	3.005 382	3.005 382	4.26e-08	96	0.121
9	3.005 382	3.005 382	4.60e-08	96	0.125
10	3.004 713	3.005 354	1.35e-04	90	0.122

从表6可以看出,随着 $\beta$ 加大,最差值、平均值、成功率基本呈先增加后减小的趋势变化,而标准差是呈先减小后增加的趋势变化,平均耗时变化不大.分析认为: $\beta$ 过小时,狼群更新数量过多,狼群很难聚集,致使算法寻优效果下降;而 $\beta$ 过大时,狼群的更新数量过少,致使狼群的多样性锐减,算法很容易陷入局部最优.为此,建议 $\beta$ 在 $[3, 9]$ 区间中选取.

最后,讨论距离判定因子 $\omega$ 对GWPA性能的影响.同样,狼群规模 $N$ 取20,其余参数不改变.设定不同的 $\omega$ 后,分别对Bridge函数进行50次寻优计算,统计结果如表7所示.

表7 距离判定因子 $\omega$ 对GWPA性能的影响

$\omega$	最差值	平均值	标准差	成功率/%	平均耗时 /s
5	3.004 777	3.005 358	1.20e-04	88	0.133
10	3.004 413	3.005 340	1.92e-04	92	0.137
20	3.005 382	3.005 382	1.03e-13	92	0.134
50	3.005 382	3.005 382	3.31e-14	90	0.133
100	3.005 382	3.005 382	2.10e-12	94	0.140
200	3.004 987	3.005 370	5.84e-05	86	0.130
500	3.004 549	3.005 348	1.65e-04	74	0.131
800	3.003 192	3.005 220	4.47e-04	76	0.129
1 000	3.002 264	3.005 207	5.25e-04	74	0.132
2 000	3.000 906	3.004 780	1.15e-03	60	0.129

从表7可以看出,随着 $\omega$ 的增加,最差值、平均值、成功率也基本呈先增加后减小的趋势变化,标准差呈先减小后增加的趋势变化,平均耗时变化不大.分析认为:由式(4)和(5)可以看出, $\omega$ 的大小直接影响到围攻半径 $L_{near}$ 的大小. $\omega$ 过小时,围攻半径会很大,致使猛狼的围攻效果降低;而 $\omega$ 过大时,围攻半径过小,猛狼很难进入到围攻区域,这就削弱了算法的局部开发能力.为此,建议 $\omega$ 在[10, 800]区间中选取.

### 3.2 工程实例仿真

为了验证GWPA算法的工程实用性,本文选择机器人最短路径规划问题<sup>[17-18]</sup>进行实例仿真.图8所示是机器人在二维平面中由起点 $B$ 到终点 $T$ 绕过障碍物的行走路径示意图.图中黑色物体表示障碍物, $p_1$ 、 $p_2$ 、 $p_3$ 表示中转点.

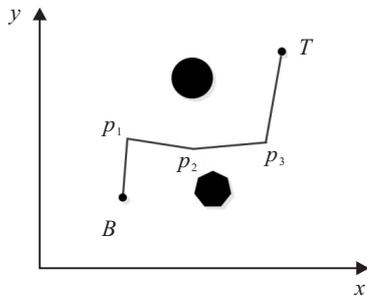


图8 机器人行走路径示意图

假定机器人从起点 $B$ 经过 $n$ 个中转点到终点 $T$ ,那么机器人的行走路径Path可以表示为

$$Path = \{B, p_1, p_2, \dots, p_n, T\}. \quad (13)$$

路径Path的长度用 $L_{path}$ 表示,采取欧氏距离表征为

$$L_{path} = L_{B,p_1} + \sum_{i=2}^n L_{p_{i-1},p_i} + L_{p_n,T}. \quad (14)$$

仿真实验中,起点为 $B(100, 100)$ ,终点为 $T(430, 430)$ ,并设定两个直径分别为50和70的圆形障碍物.分别采用WPA、GWPA计算最短路径,算法中狼群规模 $N$ 取100,最大迭代次数 $k_{max}$ 取1000,其余参数与函数优化的设定一致.仿真结果如表8所示.

表8 WPA与GWPA的仿真结果对比

$n$	$L_{near}$		$t/s$	
	WPA	GWPA	WPA	GWPA
10	467.690 5	466.692 0	21.91	13.04
20	469.249 8	468.114 7	42.09	16.72
30	489.954 4	480.449 1	62.20	20.27

从对比结果看,对多种中转点数的路径寻优,GWPA的路径都比WPA的短,并且时间消耗也比WPA少.这证实了改进算法的有效性,同时也说明改进的狼群算法可以被用来求解路径规划问题.

## 4 结 论

仿真实验表明,改进后的狼群算法具有优良的全局搜索机制,算法能够顺利跳出局部最优值,全局搜索能力得到大幅度提升;另外,围攻半径 $L_{near}$ 的提出不仅增强了算法的局部开发能力,还可以量化算法跳出局部最优的能力.通过仿真实验对比,证实了改进狼群算法的合理性、有效性和实用性.不足之处在于步长因子 $S$ 的设定还没有形成一定的标准,需要依据经验设定,算法的工程实用仍需要进一步的推广,这都是今后研究的重点.

### 参考文献(References)

- [1] Liu C A, Yan X H, Liu C Y, et al. The wolf colony algorithm and applications[J]. Chinese J of Electronics, 2011, 20(2): 212-216.
- [2] 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.)
- [3] 伊廷华, 王传伟, 李宏男. 基于等级划分狼群算法的三维传感器优化布置方法研究[J]. 建筑结构学报, 2014, 35(4): 223-229. (Yi T H, Wang C W, Li H N. Hierarchic wolf algorithm for optimal triaxial sensor placement[J]. J of Building Structures, 2014, 35(4): 223-229.)
- [4] 吴虎胜, 张凤鸣, 战仁军, 等. 求解0-1背包问题的二进制狼群算法[J]. 系统工程与电子技术, 2014, 36(8): 1660-1667. (Wu H S, Zhang F M, Zhan R J, et al. A binary wolf pack algorithm for solving 0-1 knapsack problem[J]. Systems Engineering and Electronics, 2014, 36(8): 1660-1667.)
- [5] 吴虎胜, 张凤鸣, 李浩, 等. 求解TSP问题的离散狼群算法[J]. 控制与决策, 2015, 30(10): 1861-1867. (Wu H S, Zhang F M, Li H, et al. Discrete wolf pack algorithm for traveling salesman problem[J]. Control and Decision, 2015, 30(10): 1861-1867.)
- [6] 方彦军, 唐猛. 自动小车存取系统复合作业三维空间路径优化[J]. 计算机集成制造系统, 2015, 21(3):

- 702-708.  
(Fang Y J, Tang M. Three-dimensional routing optimization for AVS/RS' s composite operation[J]. Computer Integrated Manufacturing Systems, 2015, 21(3): 702-708.)
- [7] 王建群, 贾洋洋, 肖庆元. 狼群算法在水电站水库优化调度中的应用[J]. 水利水电科技进展, 2015, 35(3): 1-4.  
(Wang J Q, Jia Y Y, Xiao Q Y, Application of wolf pack search algorithm to optimal operation of hydropower station[J]. Advances in Science and Technology of Water Resources, 2015, 35(3): 1-4.)
- [8] 刘永兰, 李为民, 吴虎胜, 等. 基于狼群算法的无人机航迹规划[J]. 系统仿真学报, 2015, 27(8): 1838-1844.  
(Liu Y L, Li W M, Wu H S, et al. Track Planning for Unmanned Aerial Vehicles Based on Wolf Pack Algorithm[J]. J of System Simulation, 2015, 27(8): 1838-1844.)
- [9] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence[C]. Proc of Int Conf on Computing Intelligence Modeling Control and Automatic. Vienna: IEEE, 2005: 695-701.
- [10] Gandomi A H, Yang X S, Talatahari S, et al. Firefly algorithm with chaos[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(1): 89-98.
- [11] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.
- [12] Tang R, Fong S, Yang X S, et al. Wolf search algorithm with ephemeral memory[C]. Seventh Int Conf on Digital Information Management. IEEE, 2012: 165-172.
- [13] Yamany W, Emary E, Hassanien A E. Wolf search algorithm for attribute reduction in classification[C]. IEEE Symposium on Computational Intelligence & Data Mining. IEEE, 2014: 351-358.
- [14] Fong S, Deb S, Hanne T, et al. Eidetic wolf search algorithm with a global memory structure[J]. European J of Operational Research, 2016, 254(1): 19-28.
- [15] Wu H S, Zhang F M. Wolf pack algorithm for unconstrained global optimization[J]. Mathematic Problems in Engineering, 2014, 2014(1): 1-17.
- [16] Wu H, Zhang F. A uncultivated wolf pack algorithm for high-dimensional functions and its application in parameters optimization of PID controller[C]. IEEE Congress on Evolutionary Computation. IEEE, 2014: 1477-1482.
- [17] Zhou Q, Zhou Y, Chen X. A wolf colony search algorithm based on the complex method for uninhabited combat air vehicle path planning[C]. Int J of Hybrid Information Technology, 2014, 7(1): 183-200.
- [18] Ali A F, Tawhid M A. A hybrid cuckoo search algorithm with Nelder Mead method for solving global optimization problems[J]. Springerplus, 2016, 5(1): 473.

(责任编辑: 齐 霁)

## 第30届中国控制与决策会议(2018CCDC)征文通知

第30届中国控制与决策会议(2018CCDC)将于2018年6月9日~11日在中国沈阳举行. 会议由东北大学和中国自动化学会信息物理系统控制与决策专业委员会主办.

第30届中国控制与决策会议论文集集中的英文论文将进入IEEE Xplore Data Base, 被EI检索.

第30届中国控制与决策会议将涉及理论与应用两方面, 主要涵盖系统、控制与决策相关课题. 征文范围如下:

**控制与决策:** 自适应控制; 复杂系统与复杂网络; 控制系统应用; 控制工程教育; 协同控制; 信号处理; 数据处理; 数据驱动控制; 决策理论与方法; 决策支持系统; 管控一体化; 时滞系统; 离散事件系统; 分布控制系统; 分布参数系统; 故障诊断与容错控制; 模糊系统; 对策论; 混杂系统; 系统辨识与参数估计; 智能系统; 知识管理与知识工程; 管理信息系统与企业信息化; 供应链与物流管理; 电子商务建模与优化; 运动控制; 网络控制系统; 神经网络; 非线性系统; 优化控制; 过程控制; 生产计划与调度; 鲁棒控制; 传感器网络; 社会经济系统; 随机系统; 变结构控制; 控制与仿真.

**自动化:** CIMS与制造系统; 工厂建模与仿真; 家庭、实

验室及服务自动化; 仪器仪表系统; 智能自动化; 人机交互; 纳米自动化与装配; 基于网络的系统; 计划、调度与协调; 过程自动化.

**机器人:** 以人为中心的人机系统; 医用机器人与生物机器人学; 微机器人与微操作; 移动机器人学; 移动传感器网络; 感知系统; 机器人控制; 机器人传感与数据融合; 搜寻、援救与野外机器人学; 人机交互; 空间与水下机器人; 遥控机器人; 视觉伺服; 多足机器人系统.

**新兴技术领域:** 信息物理系统; 智能电网; 再生能源; 能源管理系统; 集成系统与过程; 微机电系统; 电动车辆与智能交通; 交通控制; 暖通系统优化和控制; 生物系统建模.

除分组报告外, 会议另设有邀请专题, 特别专题, 特邀大会报告和杰出讲座.

会议投稿者请于2017年10月31日前提交全文. 录用文章的作者需要注册并到会宣讲论文. 请登陆<http://www.ccdc.neu.edu.cn>了解具体事宜并投稿. 同时也可通过E-mail向大会秘书([secretary\\_ccdc@ise.neu.edu.cn](mailto:secretary_ccdc@ise.neu.edu.cn))咨询.

中国控制与决策会议秘书处