

一种新的群体智能算法——狼群算法

吴虎胜^{1,2}, 张凤鸣¹, 吴庐山³

(1. 空军工程大学装备管理与安全工程学院, 陕西 西安 710051;

2. 武警工程大学装备工程学院, 陕西 西安 710086;

3. 上海工程技术大学电子电气工程, 上海 201620)

摘要: 基于狼群群体智能, 模拟狼群捕食行为及其猎物分配方式, 抽象出游走、召唤、围攻3种智能行为以及“胜者为王”的头狼产生规则和“强者生存”的狼群更新机制, 提出一种新的群体智能算法——狼群算法(wolf pack algorithm, WPA), 并基于马尔科夫链理论证明了算法的收敛性。将算法应用于15个典型复杂函数优化问题, 并同经典的粒子群算法、鱼群算法和遗传算法进行比较。仿真结果表明, 该算法具有较好的全局收敛性和计算鲁棒性, 尤其适合高维、多峰的复杂函数求解。

关键词: 进化计算; 群体智能; 狼群算法; 函数优化

中图分类号: TP 18

文献标志码: A

DOI:10.3969/j.issn.1001-506X.2013.11.33

New swarm intelligence algorithm—wolf pack algorithm

WU Hu-sheng^{1,2}, ZHANG Feng-ming¹, WU Lu-shan³

(1. Materiel Management and Safety Engineering College, Air Force Engineering University, Xi'an 710051, China;

2. Materiel Engineering College, Armed Police Force Engineering University, Xi'an 710086, China;

3. College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: Based on swarm intelligence of a pack of wolves, inspiring by their hunting behaviors and distributive mode for prey, this paper abstracts three intelligent behaviors, scouting, summoning and beleaguering, and deduces a productive rule for leading wolf that the winner can dominate its all and a renewable mechanism, named survival of the stronger, for a pack of wolves. From the above, a new heuristic swarm intelligent method, wolf pack algorithm (WPA) is proposed. Moreover, this algorithm's global probability convergence is proved based on the theory of Markov chain. Then the proposed algorithm is applied to 15 typical complex function optimization problems, and compared with three classical intelligent algorithms, particle swarm optimization algorithm, artificial fish swarm algorithm and genetic algorithm. Simulation results show that WPA has better global convergence and computational robustness, and is especially suitable for solving high-dimension and multimodal function optimization problems.

Keywords: evolutionary computation; swarm intelligence; wolf pack algorithm(WPA); function optimization

0 引言

经历漫长的生物进化与自然选择, 造就了自然界很多奇妙的群体现象, 让人叹为观止的同时也给我们无尽的启示。文献[1]通过对鸟群扑食行为的观察研究, 提出了粒子群优化(particle swarm optimization, PSO)算法。文献[2]通过模拟蚂蚁社会分工与协作觅食的原理进行寻优, 提出了蚁群优化(ant colony optimization, ACO)算法。文献[3]

模拟鱼群扑食中的觅食、聚群等行为提出了鱼群算法(fish swarm algorithm, FSA)。文献[4]模仿大肠杆菌在人体食道内的觅食行为, 提出了细菌觅食优化(bacterium foraging optimization, BFO)算法。文献[5]模拟青蛙觅食过程中群体信息共享和交流机制提出了人工混合蛙跳算法(shuffled frog leaping algorithm, SFLA)。文献[6]基于蜂后概念、蜂群繁殖及其觅食方式提出了人工蜂群算法(artificial bee colony algorithm, ABCA); 甚至有学者大胆地提出了海豚群算法、

鼠群算法、猴群算法等群体智能算法的思想^[7]。群体智能算法的发展可谓方兴未艾,这为大量复杂优化问题的求解提供了诸多选择。

鸟、鱼、青蛙,甚至蚂蚁、细菌,它们都并不具有人类的复杂逻辑推理、综合判断等高级智能,但它们在相同目标,即食物的激励下,通过对环境的不断适应和群体协作、自组织却突现出强大的群体智能,为人类解决复杂问题提供了许多新的思路。本文在分析狼群的协作捕猎活动特点的基础上,提出一种新的群体智能算法——狼群算法(wolf pack algorithm, WPA),并通过大量仿真对比实验验证了算法的有效性。

1 狼群系统分析

严酷的生活环境和千百年的进化,造就了狼群严密的组织系统及其精妙的协作捕猎方式。从成吉思汗时期蒙古铁骑的“狼群战法”到纳粹德军将领邓尼茨猖狂一时的“狼群潜艇战术”,再到美军研究的电子对抗利器——“狼群攻击系统”,都无不彰显出狼群群体智慧的巨大魅力。狼过着群居生活且都有其明确的社会分工,它们团结协作为狼群的生存与发展承担着各自的责任。

头狼:头狼始终是狼群中最具智慧和最凶猛的,是在“弱肉强食、胜者为王”式的残酷竞争中产生的首领。它不断地根据狼群所感知到的信息进行决策,负责整个狼群的指挥和把关维护,既要避免狼群陷入危险境地又要指挥狼群以期尽快地捕获猎物。

探狼:寻找猎物时,狼群不会全体出动而是派出少数精锐的探狼在猎物的可能活动范围内游猎,根据空气中猎物留下的气味进行自主决策,气味越浓表明狼离猎物越近,探狼始终就朝着气味最浓的方向搜寻。

猛狼:一旦探狼发现猎物踪迹,就会立即向头狼报告,头狼视情通过嚎叫召唤周围的猛狼来对猎物进行围攻。周围的猛狼闻声则会自发地朝着该探狼的方向奔袭,向猎物进一步逼近。

猎物分配规则:捕获猎物后,狼群并不是平均分配猎物,而是按“论功行赏、由强到弱”的方式分配,即先将猎物分配给最先发现、捕到猎物的强壮的狼,而后再分配给弱小的狼。尽管这种近乎残酷的食物分配方式会使得少数弱狼由于食物缺乏而饿死,但此规则可保证有能力捕到猎物的狼获得充足的食物进而保持其强健的体质,在下次捕猎时仍可顺利地捕到猎物,从而维持着狼群主体的延续和发展。

在下文中,将对上述狼群行为和规则进行具体描述和实现。

2 狼群算法

算法采用基于人工狼主体的自下而上的设计方法和基于职责分工的协作式搜索路径结构。如图 1 所示,通过狼群个体对猎物气味、环境信息的探知、人工狼相互间信息的共享和交互以及人工狼基于自身职责的个体行为决策最终实现了狼群捕猎的全过程。

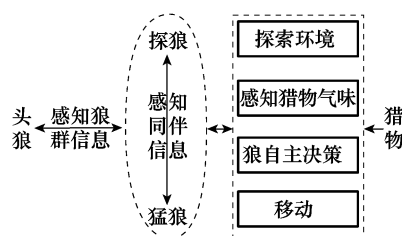


图 1 狼群捕猎模型

2.1 一些定义

设狼群的猎场空间为一个 $N \times D$ 的欧式空间,其中 N 为狼群中人工狼总数, D 为待寻优的变量数。某一人工狼 i 的状态可表示为 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,其中 x_{id} 为第 i 匹人工狼在欲寻优的第 d ($d=1, 2, \dots, D$) 维变量空间中所处位置;人工狼所感知到的猎物气味浓度为 $Y = f(\mathbf{X})$,其中 Y 就是目标函数值;人工狼 p 和 q 之间的距离定义为其状态向量间的 Manhattan 距离 $L(p, q) = \sum_{d=1}^D |x_{pd} - x_{qd}|$,当然也可依据具体问题选用其他的距离度量。另外,由于实际中极大与极小值问题之间可相互转换,为论述方便以下皆以极大值问题进行讨论。

2.2 智能行为和规则的描述

头狼、探狼和猛狼间的默契配合成就了狼群近乎完美的捕猎行动,而“由强到弱”的猎物分配又促使狼群向最有可能再次捕获到猎物的方向繁衍发展。将狼群的整个捕猎活动抽象为 3 种智能行为(即游走行为、召唤行为、围攻行为)以及“胜者为王”的头狼产生规则和“强者生存”的狼群更新机制。

(1) 头狼产生规则。初始解空间中,具有最优目标函数值的人工狼即为头狼;在迭代过程中,将每次迭代后最优狼的目标函数值与前一代中头狼的值进行比较,若更优则对头狼位置进行更新,若此时存在多匹的情况,则随机选一匹成为头狼。头狼不执行 3 种智能行为而直接进入下次迭代,直到它被其他更强的人工狼所替代。

(2) 游走行为。将解空间中除头狼外最佳的 S_num 匹人工狼视为探狼,在解空间中搜索猎物, S_num 随机取 $[n/(\alpha+1), n/\alpha]$ 之间的整数, α 为探狼比例因子。探狼 i 首先感知空气中的猎物气味,即计算该探狼当前位置的猎物气味浓度 Y_i 。若 Y_i 大于头狼所感知的猎物气味浓度 Y_{lead} ,表明猎物离探狼 i 已相对较近且该探狼最有可能捕获猎物。于是 $Y_{lead} = Y_i$,探狼 i 替代头狼并发起召唤行为;若 $Y_i < Y_{lead}$,则探狼先自主决策,即探狼向 h 个方向分别前进一步(此时的步长称为游走步长 $step_p$)并记录每前进一步后所感知的猎物气味浓度后退回原位置,则向第 p ($p=1, 2, \dots, h$) 个方向前进后探狼 i 在第 d 维空间中所处的位置为

$$x_{id}^p = x_{id} + \sin(2\pi \times p/h) \times step_p^d \quad (1)$$

此时,探狼所感知的猎物气味浓度为 Y_{ip} ,选择气味最

浓的且大于当前位置气味浓度 Y_{i0} 的方向前进一步,更新探狼的状态 \mathbf{X}_i ,重复以上的游走行为直到某匹探狼感知到的猎物气味浓度 $Y_i > Y_{\text{lead}}$ 或游走次数 T 达到最大游走次数 T_{max} 。

需要说明的是由于每匹探狼的猎物搜寻方式存在差异, h 的取值是不同的,实际中可依据情况取 $[h_{\min}, h_{\max}]$ 间的随机整数, h 越大探狼搜寻得越精细但同时速度也相对较慢。

(3) 召唤行为。头狼通过嚎叫发起召唤行为,召集周围的 M_{num} 匹猛狼向头狼所在位置迅速靠拢,其中 $M_{\text{num}} = n \cdot S_{\text{num}} - 1$; 听到嚎叫的猛狼都以相对较大的奔袭步长 $step_b$ 快速逼近头狼所在的位置。则猛狼 i 第 $k+1$ 次迭代时,在第 d 维变量空间中所处的位置为

$$x_{id}^{k+1} = x_{id}^k + step_b^d \cdot (g_d^k - x_{id}^k) / |g_d^k - x_{id}^k| \quad (2)$$

式中, g_d^k 为第 k 代群体头狼在第 d 维空间中的位置。式(2)由 2 部分组成,前者为人工狼当前位置,体现狼的围猎基础;后者表示人工狼逐渐向头狼位置聚集的趋势,体现头狼对狼群的指挥。

奔袭途中,若猛狼 i 感知到的猎物气味浓度 $Y_i > Y_{\text{lead}}$, 则 $Y_{\text{lead}} = Y_i$, 该猛狼转化为头狼并发起召唤行为;若 $Y_i < Y_{\text{lead}}$, 则猛狼 i 继续奔袭直到其与头狼 s 之间的距离 d_{is} 小于 d_{near} 时加入到对猎物的攻击行列,即转入围攻行为。设待寻优的第 d 个变量的取值范围为 $[\min_d, \max_d]$, 则判定距离 d_{near} 可由式(3)估算得到

$$d_{\text{near}} = \frac{1}{D \cdot \omega} \cdot \sum_{d=1}^D |\max_d - \min_d| \quad (3)$$

式中, ω 为距离判定因子,其不同取值将影响算法的收敛速度,一般而言 ω 增大会加速算法收敛,但 ω 过大会使得人工狼很难进入围攻行为,缺乏对猎物的精细搜索。

召唤行为体现了狼群的信息传递与共享机制,并融入了社会认知观点,通过狼群中其他个体对群体优秀者的“追随”与“响应”,充分显示出算法的社会性和智能性。

(4) 围攻行为。经过奔袭的猛狼已离猎物较近,这时猛狼要联合探狼对猎物进行紧密地围攻以期将其捕获。这里将离猎物最近的狼,即头狼的位置视为猎物的移动位置。具体地,对于第 k 代狼群,设猎物在第 d 维空间中的位置为 G_d^k , 则狼群的围攻行为可用方程(4)表示

$$x_{id}^{k+1} = x_{id}^k + \lambda \cdot step_c^d \cdot |G_d^k - x_{id}^k| \quad (4)$$

式中, λ 为 $[-1, 1]$ 间均匀分布的随机数; $step_c$ 为人工狼 i 执行围攻行为时的攻击步长。若实施围攻行为后人工狼感知到的猎物气味浓度大于其原位置状态所感知的猎物气味浓度,则更新此人工狼的位置,若不然,人工狼位置不变。

设待寻优第 d 个变量的取值范围为 $[\min_d, \max_d]$, 则 3 种智能行为中所涉及到游走步长 $step_a$ 、奔袭步长 $step_b$ 、攻击步长 $step_c$ 在第 d 维空间中的步长存在如下关系:

$$step_a^d = step_b^d / 2 = 2 \cdot step_c^d = |\max_d - \min_d| / S \quad (5)$$

式中, S 为步长因子,表示人工狼在解空间中搜寻最优解的精细程度。

(5) “强者生存”的狼群更新机制。猎物按照“由强到弱”的原则进行分配,导致弱小的狼会被饿死。即在算法中去目标函数值最差的 R 匹人工狼,同时随机产生 R 匹人工狼。 R 越大则新产生的人工狼越多,有利于维护狼群个体的多样性,但若 R 过大算法就趋近于随机搜索;若 R 过小,则不利于维护狼群的个体多样性,算法开辟新的解空间的能力减弱。由于实际捕猎中捕获猎物的大小、数量是有差别的,进而导致了不等数量的弱狼饿死。因此,这里 R 取 $[n/(2 \times \beta), n/\beta]$ 之间的随机整数, β 为群体更新比例因子。

2.3 算法描述

狼群算法的具体步骤如下。

步骤 1 数值初始化。初始化狼群中人工狼位置 \mathbf{X}_i 及其数目 N , 最大迭代次数 k_{max} , 探狼比例因子 α , 最大游走次数 T_{max} , 距离判定因子 ω , 步长因子 S , 更新比例因子 β 。

步骤 2 选取最优人工狼为头狼,除头狼外最佳的 S_{num} 匹人工狼为探狼并执行游走行为,直到某只探狼 i 侦察到的猎物气味浓度 Y_i 大于头狼所感知的猎物气味浓度 Y_{lead} 或达到最大游走次数 T_{max} , 则转步骤 3。

步骤 3 人工猛狼据式(2)向猎物奔袭,若途中猛狼感知的猎物气味浓度 $Y_i > Y_{\text{lead}}$ 则 $Y_{\text{lead}} = Y_i$, 替代头狼并发起召唤行为;若 $Y_i < Y_{\text{lead}}$, 则人工猛狼继续奔袭直到 $d_{is} \leq d_{\text{near}}$, 转步骤 4。

步骤 4 按式(4)对参与围攻行为的人工狼的位置进行更新,执行围攻行为。

步骤 5 按“胜者为王”的头狼产生规则对头狼位置进行更新;再按照“强者生存”的狼群更新机制进行群体更新。

步骤 6 判断是否达到优化精度要求或最大迭代次数 k_{max} , 若达到则输出头狼的位置,即所求问题的最优解,否则转步骤 2。

综上, WPA 的流程图如图 2 所示。

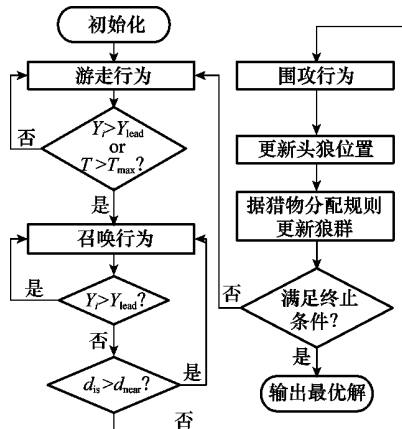


图 2 WPA 流程图

3 算法收敛性分析

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^[8]。WPA 是不断重复游走、召唤和围攻行为的过程,每种行为都只与当前的群体状态有关,而与以前的无关。显然,其种群序列为 Markov 链。

设搜索空间为 H ,游走、召唤和围攻行为会引起状态空间中的状态转移,所以可用转移矩阵 S, M 和 W 来分别表示它们的影响,定义算法 Markov 链的转移矩阵如下:

$$P = S \times M \times W \quad (6)$$

定义 1 设 P_{ij} 为 Markov 链的转移概率矩阵,若 $\forall i, j \in H$, 存在 $k \geq 1$, 使得 $P_{ij}^k > 0$, 则称此 Markov 链是不可约的。

定义 2 设非空集合 $U = \{k | k \geq 1, P_{ij}^k > 0, \forall i, j \in H\}$ 且 U 的最大公约数为 1, 则称此 Markov 链为非周期的。

定义 3 设 $u_i = \sum_{k=1}^{\infty} k P_{ij}^k$, 对于常返状态 i , 若 $u_i < +\infty$, 称 u_i 为正常返的。特别当 i 为正常返且非周期, 则此 Markov 链是遍历的。

引理 1 WPA 的优化解序列是一个有限齐次 Markov 链。

证明 设 $Q_k = \{X_1, X_2, \dots, X_N\}$ 为 WPA 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。一方面, 所有群体 Q_k 是有限的, 遂构成有限的 Markov 链; 另一方面, 由于 WPA 算法中人工狼的游走、召唤、围攻以及群体更新都是在独立随机过程中进行的, 每次人工狼的更新具有优胜选择的继承性。第 $k+1$ 代狼群的产生仅依赖于第 k 代狼群, 而与各代狼群之间的转移概率和代数 k 无关。同时, 经过人工头狼的状态更新, 可得到一组 WPA 的优化解序列。因此, 经过 WPA 一系列独立随机的变换处理得到的优化解序列是一个有限齐次 Markov 链。 证毕

引理 2 WPA 算法种群序列的 Markov 链是遍历链。

证明 首先证明 WPA 算法的种群序列是不可约的 Markov 链。由于群体转移概率矩阵 $P_{ij} = P\{Q_{k+1} = j | Q_k = i, k \geq 1\}$ 只与状态 i, j 有关, 且 $Q_k > 0$, 则群体转移概率矩阵 P 为正定矩阵。因此, 据定义 1 可知, WPA 算法的种群序列为不可约的 Markov 链。

其次证明 WPA 是非周期的不可约的 Markov 链。对于给定的 $k > 0$, 由 WPA 是不可约的 Markov 链可知, $\exists j \in H$, 使得 $P_{ij} > 0$, 又由定义 2 知 $k=1$ 。因此有集合 U 的最大公约数为 1, 则 WPA 为非周期不可约的 Markov 链。

最后证明 WPA 是遍历链。 H 为人工狼的搜索空间, P_{ij} 为状态 i 经各行为达状态 j 的转移概率。由于转移矩阵 S, M 和 W 都处于 $(0, 1)$ 之间, 则 $0 < P_{ij} < 1$, 令 $\epsilon = \max\{P_{ij} : \forall i, j \in H\}$, 则由 Cauchy-Riemann 方程和定义 3 可知

$$u_i = \sum_{k=1}^{\infty} k P_{ij}^k \leq \sum_{k=1}^{\infty} k \epsilon^k < \infty \quad (7)$$

综上所述, WPA 算法的 Markov 链是遍历链。 证毕

引理 3 文献[9]已经证明若一个进化算法满足以下 2 个条件。

(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是可达的。

(2) 若种群序列 Q_1, Q_2, \dots, Q_N 是单调的, 则此进化算法以概率 1 收敛于问题的全局最优解。

引理 4 WPA 算法以概率 1 收敛于问题的全局最优解。

证明 由于 WPA 算法种群序列的 Markov 链是遍历链, 条件(1)显然是成立的。由于 WPA 算法中的游走行为、围攻行为、“强者生存”的群体更新以及头狼的位置状态更新都体现出优秀解的保持策略。特别是依据引理 1 可知 WPA 优化解序列是一个有限齐次 Markov 链, 并且每次迭代狼群个体的位置状态只有遇到更优的解时才会更新。因此, WPA 算法产生的子代 Q_{k+1} 中的任意解都非劣于 Q_k 中的解或者说至少不差于 Q_k 中的任意解。由此很显然可知种群序列 Q_1, Q_2, \dots, Q_N 是单调的, 于是据引理 3 得证, WPA 算法以概率 1 收敛于问题的全局最优解。 证毕

4 实验与分析

为充分测试算法的性能和特点, 通常要引入较多具有不同特征的标准函数^[10]。这里采用如表 1 所示的 15 个复杂函数^[11-13]作为实验对象。表中特征“U”表示此函数为单峰(Unimodal)函数, “M”表示多峰(Multimodal), “S”表示可分(Separable), “N”表示不可分(Non-Separable)。

上述所选函数涉及单峰、多峰、可分、不可分等多种特征的复杂函数。所谓的单峰函数, 即指函数在定义域内只有全局最优值, 没有局部极值; 而多峰函数是指拥有多个局部极值的函数, 相对单峰函数更加复杂, 一般算法都较难找到全局最优值, 很容易陷入局部极值或在各局部极值间震荡^[14], 常用来检验算法的全局搜索性能和避免早熟收敛的能力^[15]。另外, 若一个拥有 N 个变量的函数能用 N 个单变量函数之和表示则此函数为可分的, 反之为不可分函数。由于不可分函数变量间关系复杂, 因此对此类函数寻优相对更加困难^[16]。此外, 搜索空间的维数也是个重要因素, 很多对于低维函数效果很好的算法面对高维复杂函数其寻优效果却较差^[17]。表 1 中函数的变量维数从 2 维直至 200 维, 都是难度较大的复杂优化问题, 具有很好的测试性, 可较为全面的反应各算法的性能。

利用表 1 中 15 个复杂函数对 WPA 算法以及经典的 PSO、FSA 和遗传算法(genetic algorithm, GA)进行比较分析。其中 PSO 算法采用美国北卡罗莱纳州大学的 Brian 教授开发的工具箱, GA 算法采用由英国谢菲尔德大学开发的算法工具箱, FSA 算法据文献[3]编程实现。

表 1 基准函数

编号	函数	表达式	维数	特征	取值范围	理论最优解
1	Easom	$f(X)=-\cos(x_1)\cos(x_2)\times\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	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	Trid6	$f(X)=\sum_{i=1}^D(x_i-1)^2-\sum_{i=2}^Dx_ix_{i-1}$	6	UN	$[-36,36]$	$\min f=-50$
4	Sumsquares	$f(X)=\sum_{i=1}^Dix_i^2$	10	US	$[-10,10]$	$\min f=0$
5	Sphere	$f(X)=\sum_{i=1}^Dx_i^2$	30	US	$[-1.5,1.5]$	$\min f=0$
6	Booth	$f(X)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	2	MS	$[-10,10]$	$\min f=0$
7	Bohachevsky1	$f(X)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	2	MS	$[-100,100]$	$\min f=0$
8	Eggcrate	$f(X)=x_1^2+x_2^2+25(\sin^2x_1+\sin^2x_2)$	2	MN	$[-\pi,\pi]$	$\min f=0$
9	Schaffer	$f(X)=0.5+\frac{(\sin\sqrt{x_1^2+x_1^2})^2-0.5}{(1+0.001(x_1^2+x_1^2))^2}$	2	MN	$[-100,100]$	$\min f=0$
10	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	$[-5,5]$	$\min f=-1.031\ 6$
11	Bohachevsky3	$f(X)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$	2	MN	$[-100,100]$	$\min f=0$
12	Bridge	$f(X)=\frac{\sin\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2}}+\exp(\frac{\cos2\pi x_1+\cos2\pi x_2}{2})-0.712\ 9$	2	MN	$[-1.5,1.5]$	$\max f=3.005\ 4$
13	Rastrigin	$f(X)=\sum_{i=1}^D[x_i^2-10\cos(2\pi x_i)+10]$	60	MS	$[-10,10]$	$\min f=0$
14	Quadric	$f(X)=\sum_{i=1}^D(\sum_{k=1}^ix_k)^2$	120	MS	$[-30,30]$	$\min f=0$
15	Ackley	$f(X)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^Dx_i^2})-\exp(\frac{1}{D}\sum_{i=1}^D\cos2\pi x_i)+20+e$	200	MN	$[-32,32]$	$\min f=0$

实验环境为:Windows XP 系统,2 G 内存,CPU 2.00 GHz,算法基于 Matlab 2008a 用 M 语言实现。

4.1 算法对比验证

为充分比较各算法的性能,利用 WPA、PSO、FSA、GA 4 种智能算法分别对 15 个复杂函数重复进行 100 次寻优计算。从最优值、最差值、平均值、计算成功率、平均耗时等多方面对算法进行评估。这里,设 F^* 为每次寻优计算所得最优函数值, F 为函数的理想最优值,当两者满足式(8)时,即视此次寻优计算成功。将寻优计算成功的次数除以总计算次数即为计算成功率,可较好地反应算法运算的稳定性和对抗局部极值的能力。

$$\begin{cases} |F-F^*|/F\leq 1e-3, F\neq 0 \\ |F-F^*|< 1e-3, F=0 \end{cases} \quad (8)$$

实验设置最大迭代次数为 2 000、初始狼群、粒子群、鱼群和染色体的规模皆为 50,PSO、FSA 和 GA 算法所涉及到

的其他参数分别遵照文献[1,3,18]的基本原则,具体设置如表 2 所示。

表 2 算法参数表

算法	主要参数
PSO	惯性权重 $w=0.729\ 8$,学习因子 $c_1=c_2=2$,个体速度限制为 $[-0.5,0.5]$
FSA	最多试探次数 $try_num=100$,感知距离 $v=1$,拥挤度因子 $de=0.618$,步长 $step=0.1$
GA	交叉概率 $p_c=0.8$,变异概率 $p_m=0.01$,代沟 $G_p=0.95$,采用轮盘赌法的选择操作,交叉和变异操作位置为随机多点
WPA	探狼比例因子 $\alpha=4$,最大游走次数 $T_{\max}=20$,距离判定因子 $w=500$,步长因子 $S=1\ 000$,更新比例因子 $\beta=6$

表 3 给出了 4 种算法对 15 个复杂函数寻优计算的统计结果,其中若计算结果小于 1e-16,则视为 0。

表 3 函数优化结果对比

编号	函数	方法	最佳值	最差值	平均值	标准差	成功率/%	平均耗时/s
1	Easom	WPA	−1	−0.996 2	−0.999 1	9.01e-4	70	5.874 2
		PSO	−1	−1	−1	0	100	1.546 6
		FSA	−1	−0.825 9	−0.994 8	0.027 5	83	133.110 2
		GA	−1	−1	−1	0	100	3.7878
2	Matyas	WPA	0	0	0	0	100	0.140 8
		PSO	3.07e-14	3.34e-10	4.70e-11	6.53e-11	100	1.498 6
		FSA	2.35e-14	3.12e-11	1.13e-11	9.02e-12	100	41.002 0
		GA	3.73e-11	9.31e-10	1.03e-10	1.78e-10	100	3.814 6
3	Trid6	WPA	−49.999 9	−49.996 1	−49.999 5	4.23e-4	95	7.919 9
		PSO	−50	−50	−50	0	100	1.591 8
		FSA	−49.969 7	−49.777 0	−49.904 7	0.046 4	0	35.112 9
		GA	−50	−49.998 5	−49.999 6	2.61e-4	97	7.349 5
4	Sumsquares	WPA	0	0	0	0	100	0.659 4
		PSO	6.67e-6	3.95e-4	7.67e-5	6.07e-5	100	1.603 4
		FSA	2.03e-12	1.03e-7	1.12e-8	1.88e-8	100	20.904 8
		GA	0.025 3	0.133 4	0.075 7	0.020 1	0	11.581 5
5	Sphere	WPA	0	0	0	0	100	0.749 8
		PSO	9.14e-4	5.71e-3	2.48e-3	9.12e-4	2	2.544 0
		FSA	5.334 7	10.552 1	8.902 0	0.789 3	0	67.006 6
		GA	0.201 5	0.452 0	0.338 7	4.38 e-2	0	26.664 3
6	Booth	WPA	9.21e-11	8.34e-6	2.78e-6	2.55e-6	100	2.768 1
		PSO	2.05e-11	1.69e-8	1.85e-9	3.06e-9	100	1.499 3
		FSA	5.75e-11	1.38e-8	2.54e-9	2.97e-9	100	29.548 7
		GA	4.55e-11	4.55e-11	4.55e-11	0	100	3.453 9
7	Bohachevsky1	WPA	0	0	0	0	100	0.057 5
		PSO	2.86e-11	1.12e-7	1.85e-8	2.29e-8	100	1.502 8
		FSA	4.18e-13	1.20e-9	2.34e-10	2.53e-10	100	34.025 1
		GA	4.36e-7	4.36e-7	4.36e-7	0	100	3.420 7
8	Eggcrate	WPA	0	0	0	0	100	0.0469
		PSO	9.46e-12	3.56e-7	2.78e-8	5.33e-8	100	1.824 9
		FSA	2.88e-7	0.002 3	3.21e-4	5.37 e-4	88	43.985 3
		GA	1.87e-9	1.87e-9	1.87e-9	2.91e-24	100	3.679 8
9	Schaffer	WPA	0	0	0	0	100	0.025 0
		PSO	3.21e-11	4.49e-8	5.63e-9	6.54e-9	100	2.183 0
		FSA	5.98e-9	0.246 9	0.061 0	0.053 5	7	63.945 7
		GA	1.82e-11	0.044 0	0.005 6	0.006 3	20	6.818 5
10	Six Hump Camel Back	WPA	−1.031 6	−1.031 6	−1.031 6	0	100	1.534 8
		PSO	−1.031 6	−1.031 6	−1.031 6	0	100	1.592 9
		FSA	−1.031 6	−1.031 6	−1.031 6	0	100	59.447 5
		GA	−1.031 6	−1.031 6	−1.031 6	0	100	3.563 9
11	Bohachevsky3	WPA	0	0	0	0	100	0.054 5
		PSO	1.51e-11	4.37e-8	4.67e-9	7.04e-9	100	1.493 4
		FSA	1.43e-11	2.17e-9	4.15e-10	4.84e-10	100	33.862 6
		GA	4.07e-8	5.12e-7	4.55e-8	4.72e-8	100	3.556 7
12	Bridge	WPA	3.005 4	3.005 4	3.005 4	5.54e-6	100	0.868 0
		PSO	3.005 4	2.847 7	3.002 2	0.022 2	98	2.857 9
		FSA	3.005 4	3.003 3	3.005 0	3.77e-4	93	61.283 0
		GA	3.005 4	2.847 7	2.987 5	0.049 3	88	5.607 4
13	Rastrigin	WPA	0	0	0	0	100	1.362 7
		PSO	22.311 4	73.704 1	40.873 8	9.263 8	0	2.987 2
		FSA	9.79 e+2	1.22 e+3	1.12 e+2	58.440 5	0	49.623 2
		GA	4.79e+2	5.78e+2	5.30e+2	19.998 7	0	56.763 1
14	Quadric	WPA	0	0	0	0	100	2.032 6
		PSO	11.078 8	59.243 0	22.055 6	6.890 0	0	29.514 2
		FSA	3.18e+3	1.12e+4	6.37e+3	1.65e+3	0	133.596 6
		GA	2.45e+3	3.89e+3	3.33e+3	2.66e+2	0	140.585 4
15	Ackley	WPA	8.88e-16	8.88e-16	8.88e-16	0	100	0.707 6
		PSO	1.689 4	2.477 4	2.078 2	0.153 2	0	6.850 9
		FSA	20.825 5	20.972 1	20.901 5	0.035 2	0	101.846 1
		GA	18.597 6	18.937 1	18.807 4	0.075 8	0	232.950 4

首先,表 3 中的最佳值、平均值可体现算法的收敛精度和寻优能力。对于简单的低维函数如 Matyas, Booth, Bohachevsky1, Six Hump Camel Back, Bohachevsky3, 4 种算法收敛精度都较高,达到 $1e-7$ 以上。对于单峰、不可分的低维复杂函数如 Easom, Matyas 和 Trid6, 特别是对于 Trid6, PSO 算法寻优效果最好,表现出较高的寻优精度和较好的算法执行能力。WPA 算法对此类函数的寻优性能稍逊于 PSO 方法。但随着变量维数的逐渐增加,算法的搜索空间复杂度成指数倍增大,对其寻优模式是个极大的考验。对于函数 Trid6, Sumsquares(维数分别为 6 和 10), PSO、FSA、GA, 3 种算法有不同程度的较好寻优表现。但当搜索空间维数增加到 30 维如 Sphere 函数,就只有 PSO 算法能找到 $1e-4$ 精度的最优解了且并不稳定。当维数增加到 60 维(Rastrigin 函数)、120 维(Quadric 函数),甚至 200 维(Ackley 函数)时,PSO、FSA、GA 这 3 种算法的寻优效果就不理想了,对于 120 维的 Quadric 函数,FSA 和 GA 算法甚至与理想值存在 $1e+3$ 级的误差。可见 WPA 算法在处理多峰、高维复杂函数中的相对优势。

其次,表 3 中的最差值、标准差和成功率体现了算法的鲁棒性和对抗局部极值的能力。除 Rastrigin、Quadric 和 Ackley 这 3 个高维复杂函数外,PSO、FSA 和 GA 这 3 种算法 100 次计算中的最差值接近理想值、标准差也较小,这反应了 3 种算法对于低维函数的寻优计算是具有一定鲁棒性的。而 WPA 算法的标准差始终处于 $1e-3$ 以下的级别,可见其在寻优运算中的良好鲁棒性。

15 个基准函数中有较多的多峰复杂函数,特别是 Rastrigin,它是 Sphere 函数的多峰版本,具有大量按正弦拐点排列的、很深的局部最优点。通常的智能算法易陷入局部最优而使得寻优失败。若成功率低于 60% 则视该算法寻优失败,则由表 3 可知: PSO 算法仅对高维多峰的 Rastrigin、Quadric 和 Ackley 这 3 个函数寻优失败; FSA 算法对 Trid6、Sphere、Schaffer、Rastrigin、Quadric 和 Ackley 共 6 个函数寻优失败; GA 算法对 Sumsquares、Sphere、Schaffer、Rastrigin、Quadric 和 Ackley 共 6 个函数寻优失败。说明它们都不同程度地存在早熟收敛的问题。这主要因为随着迭代次数的增加,3 种算法中个体的多样性逐渐消失,造成早熟收敛。而 WPA 算法对这些函数却都取得了较好的结果,分析认为游走行为使得算法在进化过程中能充分遍历搜索解空间;头狼产生规则和头狼的召唤行为使得狼群向最有可能捕获猎物的优良解区域移动;而围攻行为又使得算法在优良解域中具有较强的精细搜索能力及在优良解域附近开辟新的解空间的能力;“强者生存”的狼群更新机制则在很大程度上保证了狼群的多样性,避免算法陷入局部最优。

再次,从平均耗时来看。由表 3 可知,几乎对于所有的基准函数:FSA 算法的耗时最长,GA 算法次之。这主要因为 FSA 算法中人工鱼的主要行为——觅食行为存在大量的随机状态选择,需要反复试探,因而耗时较长。由表 3 也

可以看出,相对 GA 算法,PSO 算法的耗时仅为其一半左右,这主要是因为 GA 算法求解到一定范围时往往导致大量无谓的冗余迭代,计算时间相对较长,求解效率降低且易于早熟^[19]。而 PSO 算法没有选择、交叉、变异等操作因而更加简洁,因此 PSO 算法具有相对更快的计算速度。但 PSO 算法中粒子位置的更新主要通过比较自身位置与其周围位置和群粒子中当前最优位置来进化的,模式较为单一,使得其收敛速度在计算初期较快,但在计算的后期效率不高,耗时稍长也易陷入局部极值^[20]。而对于 WPA 算法,围攻行为使得算法具有在优良解域附近开辟新的解空间的能力,且在每次迭代后狼群都经历“强者生存”的群体更新,保持其多样性。加之,每次迭代计算时算法的部分参数是在限定范围内随机选择的,使得 WPA 算法不仅未陷入局部极值且得到的平均寻优结果也最优。因此,除 Easom、Trid6 函数外,对于 Sumsquares、Sphere、Bohachevsky1、Bohachevsky3、Eggcrate、Schaffer、Rastrigin、Quadric 和 Ackley 等多个函数,PSO 的平均耗时都相对 WPA 算法要长。

综上可见 WPA 算法对于绝大部分的函数具有较好的寻优效果,特别是对于高维、多峰的复杂函数,具有较好鲁棒性和全局搜索能力。

4.2 WPA 算法的参数分析

WPA 算法虽然具有一些优秀的性能,但也注意到 WPA 算法涉及到的参数相对较多。笔者经过大量的实例仿真,发现各参数对 WPA 算法的性能均有不同程度的影响。但不同于其他智能算法中的参数是确定的数值,WPA 算法中探狼比例因子 α ,更新比例因子 β 都是在限定范围内随机选取的,因此算法对于此种参数设置不像其他智能算法那样十分敏感,只需限定参数选取范围即可,这一点经过笔者们大量实验已验证。另外,经过多个复杂函数的仿真发现对 WPA 算法性能起主要影响作用的是距离判定因子和步长因子。下面就这 2 个参数对算法性能的影响进行详细讨论。

首先讨论距离判定因子对算法性能的影响。实验时,以 Bridge 函数为例进行说明,如图 3 所示,该函数为典型多峰函数,且极大值与其附近的诸多局部极值在数值上非常相近,普通优化算法很容易陷入局部极值或在各局部极值间震荡。

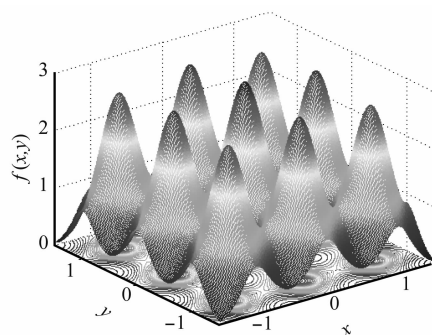


图 3 Bridge 函数及其等高线

实验时,限定最大迭代次数为 300,WPA 算法的其他参数同表 2,改变距离判定因子 w 后分别对 Bridge 函数进行 100 次寻优计算,则 w 对算法性能的影响如表 4 所示(由于以下几种情况寻优最佳值均为 3.005 4 故省去该项)。

表 4 距离判定因子 w 对 WPA 性能的影响

w	最差值	平均值	标准差	成功率/%	平均迭代代数
100	3.002 8	3.004 4	5.65e-4	65	299
200	3.004 5	3.005 2	1.89e-4	100	243
300	3.005 3	3.005 3	3.18e-5	100	215
400	3.005 3	3.005 4	1.92e-5	100	211
500	3.005 4	3.005 4	6.25e-6	100	190
600	3.005 4	3.005 4	3.59e-6	100	231
800	3.005 4	3.005 4	2.33e-6	100	248
1 000	3.005 4	3.005 4	2.28e-6	100	239
1 200	2.445 3	2.991 4	0.0886	97	300

如表 4 所示,随着距离判定因子 w 的增加,WPA 算法寻优的效果有所提高,主要表现在算法寻找到的最差值、平均值和成功率都逐渐增加,方差逐渐减小,平均迭代次数减少,在 $w=500$ 时达到最佳;再增加距离判定因子,算法寻优效果也较好但平均迭代次数逐渐增加。本文分析认为:距离判定因子 w 是控制着人工猛狼由向头狼迅速靠拢的奔袭状态转入向目标猎物围攻行为的重要参数。距离判定因子 w 增加,判定距离 d_{near} 减小,使得人工猛狼在距离头狼较近的位置转入围攻行为,即只需对头狼附近很小的解域进行精细搜索,这样就促进了 WPA 算法的快速收敛,故平均迭代次数减小。但再增加距离判定因子 w 又使得判定距离 d_{near} 过小,使得人工猛狼较难转入围攻行为,迭代次数增加。当距离判定因子 w 增加到一定程度时(如表 4 中 $w=1\ 200$),只有少量的人工猛狼转入围攻行为,由于缺乏对优良解域足够的精细搜索而导致寻优效果下降,致使 100 次寻优计算中偶尔会出现较差的结果,平均迭代次数也大幅增加。但同时,由表 4 也可以看出,距离判定因子 w 取值为 $[200,1\ 000]$ 时,WPA 算法均能得到较好的寻优效果,进而降低了 WPA 参数选取的难度,在实际中更具应用价值。

其次讨论步长因子 S 对算法性能的影响,仍以 Bridge 函数为例。其他参数同表 2,改变 S 后分别对该函数进行 100 次寻优计算,则 S 对 WPA 算法性能的影响如表 5 所示。

表 5 步长因子 S 对 WPA 性能的影响

w	最差值	平均值	标准差	成功率/%	平均迭代代数
100	3.003 7	3.005 2	3.67e-4	90	36
200	3.005 1	3.005 3	6.41e-5	100	63
400	3.005 3	3.005 4	1.28e-5	100	103
600	3.005 4	3.005 4	4.41e-6	100	144
800	3.005 4	3.005 4	5.88e-6	100	169
1 000	3.005 4	3.005 4	6.25e-6	100	190
1 200	3.005 1	3.005 4	4.89e-5	100	217
1 400	2.982 0	3.003 0	0.007 4	93	286
1 600	2.856 2	3.001 2	0.023 6	88	300

步长因子 S 体现人工狼在解空间中搜寻最优解的精细程度。如表 5 所示,随着 S 的增加,搜索精细程度增加,WPA 算法的平均迭代次数也相应增加。但从收敛精度来看,当 S 取值在 $[100,600]$ 时,最差值和平均值增加,标准差由 $3.67\text{e-}4$ 减小到 $4.41\text{e-}6$,说明在一定范围内增加 S 有利于提高 WPA 算法的收敛精度。随着 S 的继续增加,标准差逐渐增大,收敛精度有所下降,至 $S=1\ 400$ 时,由于步长因子 S 过大,造成人工狼在解域中运动时的步长过小,不能尽可能有效地遍及优良解域,造成寻优效果相对不佳,平均迭代次数增加。但同样由表 5 也可看出,WPA 算法对参数 S 的适应取值范围较大,在 $[200,1\ 200]$ 范围内取值皆可得到较好的寻优结果,不会因参数的微小改变而较大程度地影响算法性能,再次说明 WPA 的参数选取难度低,算法鲁棒性较好。

5 结 论

基于对狼群群体智能的系统分析,抽象出游走、召唤、围攻 3 种群体智能行为以及“胜者为王”的头狼产生规则和“强者生存”的狼群更新机制,实现了 WPA。通过 WPA 与 3 种经典的智能算法在解决 15 个复杂函数寻优问题中的比较以及 WPA 的参数分析实验,结果显示 WPA 对不同特征的复杂函数都具有较好的鲁棒性和全局收敛性能,可有效避免常用智能算法所出现的算法早熟收敛问题。特别对于如 Quadric、Ackley 等多峰、高维的复杂函数寻优效果较好,可以给大量非线性、多峰值的复杂优化问题的求解提供新的思路和解决方法。

参考文献:

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proc. of the IEEE International Conference on Neural Networks, 1995: 1942-1948.

[2] Dorigo M, Maniezzo V, Coloni A. Ant system: optimization by a colony of cooperating agent[J]. IEEE Trans. on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.

[3] Li X L, Shao Z J, Qian J X. An optimizing method based on autonomous animats: fish-swarm algorithm[J]. Systems Engineering Theory and Practice, 2002, 22(11): 32-38. (李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.)

[4] Passino K. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, 22(3): 52-67.

[5] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.

[6] 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.

- [7] Jorge A, Ocotlán D P, Felipe C, et al. Meta-heuristics algorithms based on the grouping of animals by social behavior for the traveling salesman problem[J]. *International Journal of Combinatorial Optimization Problems and Informatics*, 2012, 3(3): 104 – 123.
- [8] Li J H, Li M. Convergence analysis and convergence rate estimate of cellular genetic algorithm[J]. *Pattern Recognition and Artificial Intelligence*, 2012, 25(5): 874 – 878. (李军华, 黎明. 元胞遗传算法的收敛性和收敛速度估计[J]. 模式识别与人工智能, 2012, 25(5): 874 – 878.)
- [9] Back T. *Evolutionary algorithms in theory and practice* [M]. New York: Oxford University Press, 1996: 21 – 28.
- [10] Boyer D O, Martínez C H, Pedrajas N G. Crossover operator for evolutionary algorithms based on population features[J]. *Journal of Artificial Intelligence Research*, 2005, 24(2): 1 – 48.
- [11] Dervis K, Bahriye A. A comparative study of artificial bee colony algorithm [J]. *Applied Mathematics and Computation*, 2009, 90(3): 1 – 25.
- [12] Hu M Q, Wu T, Weir J D. An intelligent augmentation of particle swarm optimization with multiple adaptive methods[J]. *Information Sciences*, 2012, 20(5): 68 – 83.
- [13] Saeed M, Hamid S Z. Improved particle swarm optimization and applications to hidden markov model and ackley function[C] // *Proc. of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2011: 146 – 169.
- [14] Tang Q, Shen Y, Hu C Y, et al. Swarm intelligence-based co-operation optimization of multi-modal functions[J]. *Cognitive Computation*, 2013, 5(1): 48 – 55.
- [15] 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.
- [16] Pilar C, Francisco B, Jose A, et al. Evolutionary algorithm characterization in real parameter optimization problems[J]. *Applied Soft Computing*, 2013, 2(1): 1902 – 1921.
- [17] Wu J H, Zhang J, Li R F, et al. A multi-subpopulation PSO immune algorithm and its application on function optimization[J]. *Journal of Computer Research and Development*, 2012, 49(9): 1883 – 1898. (吴建辉, 章兢, 李仁发, 等. 多子种群微粒群免疫算法及其在函数优化中应用[J]. 计算机研究与发展, 2012, 49(9): 1883 – 1898.)
- [18] Sriniva M, Patnaik L M. Genetic algorithms: a survey[J]. *Computer*, 1994, 27(6): 17 – 26.
- [19] Yu S W, Wei Y M, Zhu K J. Hybrid optimization algorithms based on particle swarm optimization and genetic algorithm[J]. *Systems Engineering and Electronics*, 2011, 33(7): 1647 – 1652. (於世为, 魏一鸣, 诸克军. 基于粒子群-遗传的混合优化算法[J]. 系统工程与电子技术, 2011, 33(7): 1647 – 1652.)
- [20] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. *Information Processing Letters*, 2003, 85(6): 317 – 325.

作者简介:

吴虎胜(1986 –), 男, 博士研究生, 主要研究方向为信息系统工程与智能决策、智能数据挖掘。

E-mail: wuhusheng0421@163.com

张凤鸣(1963 –), 男, 教授, 博士, 主要研究方向为系统工程、数据挖掘、故障诊断。

E-mail: zfmwenzhang007@163.com

吴庐山(1988 –), 男, 硕士研究生, 主要研究方向为机电工程、故障诊断。

E-mail: 623827681@qq.com