

1-4-2019

Predator-prey Dynamics-based Optimization

Qiuqin Lu

School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Guangqiu Huang

School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Predator-prey Dynamics-based Optimization

Abstract

Abstract: To solve the complex function optimization problem, a predator-prey dynamics optimization algorithm is proposed. *The algorithm assumes that there are two species of predator and prey in an ecosystem. There is competition among the same species, the higher the population density, the more intense the competition, and then characteristics of populations are more likely to produce changes; strong populations are more likely to defeat weak populations when competition, and thus to obtain significant changes in their own characteristics. When predator populations eat prey populations, their characteristics also change. The competition operator, the predator-prey operator and the growth operator are constructed by use of the above relationship of ecosystem phenomena, these operators are helpful to make the search jump out of the local trap. The algorithm is globally convergent and can be used to solve a class of extremely complex optimization problems.*

Keywords

evolution algorithm, intelligent optimization algorithm, meta-heuristic search, population dynamics, predator-prey dynamics model

Recommended Citation

Lu Qiuqin, Huang Guangqiu. Predator-prey Dynamics-based Optimization[J]. Journal of System Simulation, 2018, 30(10): 3975-3984.

捕食-被食动力学优化算法

陆秋琴, 黄光球

(西安建筑科技大学管理学院, 陕西 西安, 710055)

摘要: 为了解决复杂函数优化问题, 提出了捕食-被食动力学优化算法。该算法假设在某生态系统中有捕食者和被食者两个种群。两类种群内部有竞争, 种群内部密度越高竞争越激烈, 种群特征更容易产生变化; 强壮种群更容易在竞争中战胜弱势种群, 从而使得自身特征发生较大改变。捕食者种群食用被食者种群后, 其特征也发生变化。根据上述特点构造出了竞争算子、捕食-被食算子和生长算子, 这些算子有利于使搜索跳出局部陷阱。本算法具有全局收敛性的特点, 可求解一类极复杂优化问题。

关键词: 进化算法; 智能优化计算; 启发式搜索; 种群动力学; 捕食-被食动力学模型

中图分类号: TP301.6 文献标识码: A 文章编号: 1004-731X (2018) 10-3975-10

DOI: 10.16182/j.issn1004731x.joss.201810045

Predator-prey Dynamics-based Optimization

Lu Qiuqin, Huang Guangqiu

(School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: To solve the complex function optimization problem, a predator-prey dynamics optimization algorithm is proposed. The algorithm assumes that there are two species of predator and prey in an ecosystem. There is competition among the same species, the higher the population density, the more intense the competition, and then characteristics of populations are more likely to produce changes; strong populations are more likely to defeat weak populations when competition, and thus to obtain significant changes in their own characteristics. When predator populations eat prey populations, their characteristics also change. The competition operator, the predator-prey operator and the growth operator are constructed by use of the above relationship of ecosystem phenomena, these operators are helpful to make the search jump out of the local trap. The algorithm is globally convergent and can be used to solve a class of extremely complex optimization problems.

Keywords: evolution algorithm; intelligent optimization algorithm; meta-heuristic search; population dynamics; predator-prey dynamics model

引言

工程中存在大量优化问题, 其目标函数和约



收稿日期: 2016-09-14 修回日期: 2017-01-19;
基金项目: 陕西省自然科学基金基础研究计划-重点项目(2015JZ010), 陕西省教育厅服务地方专项计划(16JF015), 教育部人文社会科学研究规划基金一般项目(15YJA910002), 陕西省社会科学基金(2014P07);
作者简介: 陆秋琴(1966-), 女, 广西武鸣, 博士, 教授, 研究方向为计算机仿真、计算智能。

束条件不需要特殊的限制条件, 传统的基于函数连续性和可导性的数学优化方法无法获得这类问题的全局最优解^[1]。求解此类优化问题的常用方法是群智能优化算法, 这类算法具有较广泛的适用性。典型的群智能算法有 GA 算法、ACA 算法、PSO 算法、DE 算法、BBO 算法和 ABC 算法等^[1], 而实码遗传算法(RC-GA)^[2], 差分类蚁群算法

<http://www.china-simulation.com>

• 3975 •

(DASA)^[3], 无参数粒子群算法(NP-PSO)^[4], 修正差分算法(MDE)^[5]、自适应差分算法(SaDE)^[6]、修正生物地理学算法(MBBO)^[7]和基因重组人工蜂群算法(GRABC)^[8]是这些算法的最新进展。文献[9]依据精英协同进化算法结合精英策略、协同进化思想和模拟退火机制, 提出了一种基于模拟退火机制的精英协同进化算法; 文献[10-11]构造了一种基于子任务划分方法的合作型协同进化模型, 并将该模型引入人工免疫系统中, 提出了一种基于合作模型的协同免疫多目标优化算法; 文献[12]提出了一种基于博弈模型的合作式粒子群优化算法, 算法基于重复博弈模型, 使得每次博弈都能够产生最大效益; 文献[13]提出了一种链式多种群多智能体进化算法, 其中的各种群通过与其动态邻域智能体的竞争、合作及自学习操作来增加自身的能量; 动态邻域的链式结构提高了算法的效率、降低了计算复杂度; 多个种群之间的信息定期以一定的方式进行交互, 增强了种群的多样性, 减小了算法陷入局部最优的机率; 文献[14]提出了一种基于生态种群捕获竞争模型的多目标 Memetic 优化算法, 该算法设计并运用了捕获竞争、强化学习算子进行全局搜索, 在种群进化过程中结合了 Memetic 搜索算子进行局部搜索; 文献[15]设计了利用信息共享进行子群体迁徙策略和信息浓度更新规则, 实现了各子群体之间的协同, 加快了各子群体的收敛速度, 使得在较好信息的利用和全局性的搜索两个方面达到最佳平衡。

上述算法主要存在如下问题: (1)在种群进化过程中未考虑种群内部的竞争, 在竞争过程中未考虑数量的随机变化; (2)竞争过程单一, 未体现出常见的不同类型的种群间的复杂捕食-被食关系; (3)竞争过程未体现不同类型的种群的特征的随机变化。

为了解决上述研究存在的问题, 本文提出一种具有同类种群间竞争的捕食-被食动力学优化算法(Optimization Algorithm Based on Predator-prey Dynamics with Competition among the Same Species, 简称 OA-PPD 算法), 该算法具有搜索能

力强和全局收敛性的特点, 为复杂优化问题的求解提供了一种解决方案。

1 优化算法设计方法

考虑优化问题

$$\begin{aligned} \min f(\mathbf{X}) \\ \text{s.t.} \quad \begin{cases} g_i(\mathbf{X}) \geq 0, i=1, 2, \dots, I \\ \mathbf{X} \in H \subset R^n, \mathbf{X} \geq 0 \end{cases} \end{aligned} \quad (1)$$

式(1)中, R^n 是 n 维欧氏空间; $\mathbf{X}=(x_1, x_2, \dots, x_n)$ 是一个 n 维向量, 变量 $x_i(i=1, 2, \dots, n)$ 为实数; H 为搜索空间, 又称解空间; $f(\mathbf{X})$ 为目标函数; $g_i(\mathbf{X}) \geq 0$ 为第 i 个约束条件, $i=1, 2, \dots, I$, I 为不等式约束条件个数。目标函数 $f(\mathbf{X})$ 和约束条件 $g_i(\mathbf{X})$ 不需要特殊的限制条件, 传统的基于函数连续性和可导性的数学优化方法无法获得该问题的全局最优解^[1]。

1.1 生物捕食-被食场景

假设在某个生态系统中生长有两个生物种群, 第一个是捕食者种群, 该种群消耗被食者种群的全部或部分身体, 直接获得营养以维持自己的生命; 第二个是被食者种群, 该种群以生态系统中的一些资源为食。假定同类种群间为争夺资源而存在竞争, 高密度种群容易在竞争中获得较大的优势, 从而使其生长特征发生较大的变化; 强势种群能够战胜弱势种群, 也容易在竞争中获得较大的优势。捕食者种群食用被食者种群后, 其自身生长特征也发生改变。被食者种群被过度捕食, 反而会给捕食者种群带来生存危机, 从而导致其数量减少。描述上述关系的捕食-被食系统动力学模型^[16]为:

$$\begin{cases} \frac{dS_i(t)}{dt} = S_i(t)(b_i - \varepsilon \sum_{j=1, j \neq i}^N S_j(t) - \beta_i E_i(t)) \\ \frac{dE_i(t)}{dt} = E_i(t)(-g_i + \gamma_i S_i(t) - \varepsilon \sum_{j=1, j \neq i}^N E_j(t)) \end{cases}, \quad i=1, 2, \dots, N \quad (2)$$

式(2)中, $S_i(t)$, $E_i(t)$ 分别表示被食者种群 i 和捕食者种群 i 的密度大小; b_i , g_i , β_i , γ_i , ε 均为正常数, 分别表示出身率、死亡率、捕食率、食饵消化吸收

后转化的增长率、同类种群间的竞争系数。

1.2 生物捕食-被食场景与优化问题求解过程的影射关系

为了计算方便, 将优化问题式(1)的目标函数改写成式(3):

$$F(\mathbf{X}) = \begin{cases} f(\mathbf{X}) & \forall i \in \{1, 2, \dots, I\}, g_i(\mathbf{X}) \geq 0; \\ & \forall j \in \{1, 2, \dots, E\}; \mathbf{X} \in H; \mathbf{X} \geq 0 \\ F_{\max} & \text{其它} \end{cases} \quad (3)$$

式(3)中, F_{\max} 为非常大的实数。

将1.1节描述的生物捕食场景影射到复杂优化问题(1)的解空间, 其含义为: 优化问题的搜索空间与生态系统相对应, 时期 t , 捕食者和被食者种群分别有 N 个, 生态系统共有 $2N$ 个种群在活动, N 个捕食者种群和 N 个被食者种群分别对应于搜索空间中的 N 个试探解, 即 $\{X_1^u(t), X_2^u(t), \dots, X_N^u(t)\}$, 其中, $X_i^u(t) = (x_{i1}^u(t), x_{i2}^u(t), \dots, x_{in}^u(t))$, $i=1, 2, \dots, N$; $u \in \{P, D\}$, P 、 D 分别表示捕食者和被食者种群。种群 i 中的一个特征属性 j 对应于优化问题试探解 $X_i^u(t)$ 中的一个变量 $x_{ij}^u(t)$ 。捕食者和被食者种群在同类中的竞争和捕食者种群食用被捕食者种群过程中, 其生长特征发生变化, 该变化相当于搜索空间的试探解从一个位置转移到另外一个位置。

基于上述场景, 可以构造出用于求解复杂优化问题式(3)的优化算法。为方便计算, 将式(2)改为离散递推形式, 即

$$\begin{cases} S_i(t+1) = S_i(t) + S_i(t)(b_i - \varepsilon \sum_{j=1, j \neq i}^N S_j(t) - \beta_i E_i(t)) \\ E_i(t+1) = E_i(t) + E_i(t)(-g_i + \gamma_i S_i(t) - \varepsilon \sum_{j=1, j \neq i}^N E_j(t)) \end{cases}, \quad i=1, 2, \dots, N \quad (4)$$

在时期 t , 类型为 u 的种群在其所在的种群中所占的比例为 r_i^u , 即

$$r_i^P(t) = E_i(t) / \sum_{k=1}^N E_k(t), \quad i=1, 2, \dots, N \quad (5)$$

$$r_i^D(t) = S_i(t) / \sum_{k=1}^N S_k(t), \quad i=1, 2, \dots, N \quad (6)$$

时期 t , 在类型为 u 的种群中, $u \in \{P, D\}$,

其第 i 个种群的生长能力强弱用种群生长指数 PGI 来表示, 其计算方法为:

$$PGI(X_i^u(t)) = \begin{cases} (1 + F(X_i^u(t)))^{-1}, & \text{若 } F(X_i^u(t)) > 0 \\ 1 + |F(X_i^u(t))|, & \text{若 } F(X_i^u(t)) \leq 0 \end{cases}, \quad i=1, 2, \dots, N \quad (7)$$

式中, $X_i^u(t)$ 为时期 t 类型为 u 的第 i 个种群所对应的试探解。

1.3 特征种群集合生成方法

时期 t , 假设当前种群的类型为 u , $u \in \{P, D\}$, 种群编号为 i , 特征种群集合生成方法如下:

(1) 高密度种群集合 HM^u : 从类型为 u 的种群中随机挑选出 L 个种群, 由其编号形成集合为 $HM^u = \{s_1^u, s_2^u, \dots, s_L^u\}$, 使得对于所有 $s \in \{s_1^u, s_2^u, \dots, s_L^u\}$, 满足 $r_s^u(t) > r_i^u(t)$ 。 L 又称为施加影响的种群数。密度越高的种群内部的竞争越激烈。

(2) 优势种群集合 GM^u : 从类型为 u 的种群中随机挑选出 L 个种群, 这些种群的 PGI 指数比当前种群的 PGI 指数高, 由其编号形成的集合为 $GM^u = \{q_1^u, q_2^u, \dots, q_L^u\}$ 。优势种群有某些优良特征, 这些优良特征有利于其在同类种群的竞争中获胜。

(3) 强势种群集合 OM^u : 从类型为 u 的种群中随机挑选出 L 个种群, 这些种群的 PGI 指数和占比要比当前种群的 PGI 指数和占比高, 形成强势种群集合 $OM^u = \{w_1^u, w_2^u, \dots, w_L^u\}$, 即对于所有 $w \in \{w_1^u, w_2^u, \dots, w_L^u\}$, 有 $PGI(X_w^u(t)) > PGI(X_i^u(t))$, 且占比 $r_w^u(t) > r_i^u(t)$ 。强势种群不仅特征强, 而且密度大, 强势种群间的竞争更加激烈。

1.4 演化算子设计

(1) 竞争算子。该算子描述的是同类种群之间相互竞争后种群的变化。在这种情况下, 当前种群的特征变化计算方法为

$$v_{i,j}^u(t+1) = \alpha x_{i,j}^u(t) + \lambda_1 \sum_{k \in HM^u, k \neq i} x_{k,j}^u(t) + \lambda_2 \sum_{m \in GM^u, m \neq i} x_{m,j}^u(t) + \lambda_3 \sum_{o \in OM^u, o \neq i} x_{o,j}^u(t) \quad (8)$$

式中, $v_{i,j}^u(t+1)$ 为时期 $t+1$ 类型为 u 、编号为 i 的种群

的特征 j 的状态值; $x_{k,j}^u(t)$ 、 $x_{m,j}^u(t)$ 和 $x_{o,j}^u(t)$ 分别为时期 t 类型为 u 类的高密度、优势和强势种群的特征 j 的状态值; $\alpha=Rand(0.85, 0.95)$, $Rand(a, b)$ 为在 $[a, b]$ 产生均匀分布随机数; $\lambda_1, \lambda_2, \lambda_3$ 的取值方法为

a) 若当前种群只与高密度种群竞争, 则 $\lambda_2=0$, $\lambda_3=0$, $\lambda_1=Rand(0.6, 0.7)$ 。

b) 若当前种群只与优势种群竞争, 则 $\lambda_1=0$, $\lambda_3=0$, $\lambda_2=Rand(0.6, 0.7)$ 。

c) 若当前种群只与强势种群竞争, 则 $\lambda_1=0$, $\lambda_2=0$, $\lambda_3=Rand(0.6, 0.7)$ 。

d) 若当前种群与所有种群都竞争, 则 $\lambda_1=Rand(0.6, 0.7)$, $\lambda_2=Rand(0.6, 0.7)$, $\lambda_3=Rand(0.6, 0.7)$ 。

(2) 捕食-被食算子。该算子描述的是捕食者种群捕食被食者种群后, 其生长特征发生变化的情况。该算子分为4种情况: 捕食者种群吸收高密度被食者种群、捕食者种群吸收优势被食者种群和捕食者种群吸收强势被食者种群和捕食者种群随机挑选某类被食者种群。其算子为:

$$v_{i,j}^p(t+1) = \sum_{k \in HM^p \cup GM^p \cup OM^p} \gamma_k x_{k,j}^D(t) \quad (9)$$

(3) 生长算子。该算子描述的是捕食者种群、被食者种群的生长, 即

$$\begin{cases} \mathbf{X}_i^u(t+1) = \begin{cases} \mathbf{V}_i^u(t+1) & \text{若 } PGI(\mathbf{V}_i^u(t+1)) > PGI(\mathbf{X}_i^u(t)), \\ \mathbf{X}_i^u(t) & \text{其它} \end{cases} \\ i=1, 2, \dots, N \end{cases} \quad (10)$$

式中: $u \in \{P, D\}$; 函数 $PGI()$ 按式(7)计算;

$$\mathbf{V}_i^u(t+1) = (v_{i,1}^u(t+1), v_{i,2}^u(t+1), \dots, v_{i,n}^u(t+1))。$$

1.5 OA-PPD 算法设计

OA-PPD 算法如下:

步骤 1: a) 令时期 $t=0$; 初始化如下参数: 演化时期数 $G=10^8$; 最优解误差 $\varepsilon=10^{-8} \sim 10^{-15}$; 捕食、被食种群数 $N=100 \sim 200$; 施加影响的种群数 $L \geq 3$; 种群生长特征受到影响的概率 $E_0=1/1000 \sim 1/100$; b) 分别随机确定 N 个捕食、被食种群的初始密度 $S_1(0)$,

$S_2(0), \dots, S_N(0)$; $E_1(0), E_2(0), \dots, E_N(0)$ 。c) 随机确定 $2N$ 个试探解 $\mathbf{X}_1^u(0), \mathbf{X}_2^u(0), \dots, \mathbf{X}_N^u(0)$, $u \in \{P, D\}$ 。

步骤 2: 执行下列步骤:

FOR $t=0$ TO G

按式(5)和式(6)计算 $r_i^p(t)$ 和 $r_i^D(t)$, $i=1, 2, \dots, N$;

FOR $i=1$ TO N

设置模型参数: $b_i, g_i, \beta_i, \gamma_i, \varepsilon=Rand(0.1, 0.3)$;

生成特征种群集合 HM^u, GM^u, OM^u , $u \in \{P, D\}$;

按式(4)计算 $S_i(t+1), E_i(t+1)$;

FOR $u=P$ TO D

FOR $j=1$ TO n

$p=Rand(0, 1)$;

IF $p \leq E_0$ THEN

$m_0=Rand(0, 1)$;

IF $u=P$ THEN

$q_0=Rand(0, 1)$;

IF $q_0 \leq 1/5$ THEN

令 $\lambda_2=0, \lambda_3=0, \lambda_1=Rand(0.6, 0.7)$, 按式(8)执行竞争算子, 得到 $v_{i,j}^u(t+1)$;

ELSE IF $1/5 < q_0 \leq 2/5$ THEN

$\lambda_1=0, \lambda_3=0, \lambda_2=Rand(0.6, 0.7)$, 按式(8)执行竞争算子, 得到 $v_{i,j}^u(t+1)$;

ELSE IF $2/5 < q_0 \leq 3/5$ THEN

$\lambda_1=0, \lambda_2=0, \lambda_3=Rand(0.6, 0.7)$, 按式(8)执行竞争算子, 得到 $v_{i,j}^u(t+1)$;

ELSE IF $3/5 < q_0 \leq 4/5$ THEN

$\lambda_1=Rand(0.6, 0.7), \lambda_2=Rand(0.6, 0.7), \lambda_3=Rand(0.6, 0.7)$, 按式(8)执行竞争算子, 得到 $v_{i,j}^u(t+1)$;

ELSE IF $4/5 < q_0 \leq 1$ THEN

按式(9)执行捕食-被食算子, 得到 $v_{i,j}^u(t+1)$;

END IF

ELSE IF $u=D$ THEN

```

 $q_0 = \text{Rand}(0, 1);$ 
IF  $q_0 \leq 1/4$  THEN
 $\lambda_2 = 0, \lambda_3 = 0, \lambda_1 = \text{Rand}(0.6, 0.7)$ , 按式(8)执行
竞争算子, 得到  $v_{i,j}^u(t+1)$ ;
ELSE IF  $1/4 < q_0 \leq 2/4$  THEN
 $\lambda_1 = 0, \lambda_3 = 0, \lambda_2 = \text{Rand}(0.6, 0.7)$ , 按式(8)执行
竞争算子, 得到  $v_{i,j}^u(t+1)$ ;
ELSE IF  $2/4 < q_0 \leq 3/4$  THEN
 $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = \text{Rand}(0.6, 0.7)$ , 按式(8)执行
竞争算子, 得到  $v_{i,j}^u(t+1)$ ;
ELSE IF  $3/4 < q_0 \leq 1$  THEN
 $\lambda_1 = \text{Rand}(0.6, 0.7), \lambda_2 = \text{Rand}(0.6, 0.7),$ 
 $\lambda_3 = \text{Rand}(0.6, 0.7)$ , 按式(8)执行竞争算子, 得到
 $v_{i,j}^u(t+1)$ ;
END IF
END IF
ELSE
 $v_{i,j}^u(t+1) = x_{i,j}^u(t+1)$ ;
END IF
END FOR
按式(10)执行生长算子, 得到  $X_i^u(t+1)$ ;
END FOR
END FOR
IF 新得到的全局最优解  $X^{*t+1}$  的误差  $< \varepsilon$  THEN
转步骤3;
保存新得到的全局最优解  $X^{*t+1}$ ;
END FOR
步骤3: 结束。

```

1.6 OA-PPD 算法的特性及收敛性分析

(1) 演化过程具有 Markov 特性。从竞争算子和捕食-被食算子的定义知, 任何一个试探解的新一代的生成只与该试探解的当前状态有关, 而与该试探解以前是如何演变到当前状态的历程无关。

(2) 演化过程具有“步步不差”特性。从生长算子的定义便知。

由于 OA-PPD 算法的演化过程具有 Markov 特

性和“步步不差”特性, 根据文献[17]可得如下定理成立:

定理 1 OA-PPD 算法具有全局收敛性。

定理 1 的证明方法可参见文献[17]。

1.7 时间复杂度

OA-PPD 算法的时间复杂度计算过程如表 1 所示, 其时间复杂度与演化时期 G 、捕食-被食种群数 N 、变量个数 n 以及各算子的时间复杂度以及其他辅助操作相关。

表 1 OA-PPD 算法的时间复杂度计算表

操作	时间复杂度	最多循环次数
初始化	$O(6n+8nN+2n^2N)$	1
计算 $r_i^P(t)$ 和 $r_i^D(t)$	$O(2)$	GN
竞争算子	$O((N+2L+3)nE_0/6)$	GN
捕食-被食算子	$O((N+2L+3)nE_0/8)$	GN
状态保持	$O((2-8E_0/10)n)$	GN
目标函数计算	$O(n) \sim O(n^2)$	GN
生长算子	$O(6n)$	GN
结果输出	$O(n)$	1

2 本算法与其他算法比较

本次实验采用一台 Toshiba 笔记本电脑来测试 OA-PPD 算法, 其 CPU 是 Intel Core™ I5, M520@2.40GHz, 内存是 4GB, 操作系统为 Windows 7。本文使用 CEC2013^[18]所提供的国际上通用的基准函数来测试 OA-PPD 的性能, 本文选择的 6 个基准函数为 F15~F20, 如表 2 所示。这些基准函数更多的信息见参考文献[18]。

表 2 基准函数

基准函数	每个变量的范围	理论全局最优解	理论全局最优目标函数值
F15	[-100,100]	未知	未知
F16	[-100,100]	O	200
F17	[-100,100]	O	300
F18	[-100,100]	O	400
F19	[-100,100]	O	500
F20	[-100,100]	O	600

在表2中, \mathbf{O} 是一个 n 维决策向量。在这里用 OA-PPD 去求解表2所示的基准函数, OA-PPD 的参数是 $n=50$, $\varepsilon=1.0E-10$, $N=200$, $G=1.0E+8$; F15~F20 个函数参数 $M1$ 和 $M2$ 的值根据文献[18]设置, 为了进行收敛性分析, \mathbf{O} 的值随机产生。

选择7种优化算法与 OA-PPD 算法进行比较, 这些算法包括: RC-GA^[2], DASA^[3], NP-PSO^[4], MDE^[5]、SaDE^[6]、MBBO^[7]和 GRABC^[8]。计算时, 这7种被比较优化算法的参数按表3进行初始化。

用这些算法独立运行每个基准函数51次, 表

4列出了平均最优目标函数值和平均耗时, 并对每种算法进行排序。表4的排名1按该算法的平均最优目标函数值排序, 排名2按平均最优目标函数值和对应的耗时排序。表5是 OA-PPD 算法和7种被比较算法所得最优解的非参数 Wilcoxon 秩和检验结果, 表中 $h\text{-value}=1$ 表明 OA-PPD 算法从统计学意义上来说能99%地肯定明显优于被比较算法, $h\text{-value}=-1$ 表示被比较算法明显优于 OA-PPD 算法, 而 $h\text{-value}=0$ 表示两种算法的结果差异不显著。

表3 7种优化算法的参数
Tab. 3 Parameters of the 7 compared algorithms

优化算法	参数
RC-GA	染色体数目 $N=100$, 变异率 $=0.01$, 父个体数量 $=0.5N$, $G=1.0E+8$
DASA	蚂蚁的数量 $m=37$, 离散基 $b=10$, 信息素衰减率 $\rho=0.2$, 全局规模增长因素 $s_1=0.02$, 全局规模递减因素 $s_2=0.01$, 变量的最大精度 $\varepsilon=1.0e-15$, $G=1.0E+8$
NP-PSO	$N=100$, $G=1.0E+8$
MBBO	生境修正概率 $=1$, 基因转移概率 $=[0,1]$, 概率的数值积分步长 $=1$, 每个岛屿的最大迁移和迁移率 $=1$, 变异率 $=0.02$, $N=100$, 精英 $=2$, $G=1.0E+8$
MDE	权重因子 $F=0.5$, 交叉常数 $CR=0.9$, $N=100$, $G=1.0E+8$
SaDE	加权因子区间 $=[0.45,0.55]$, 交叉常数区间 $=[0.85,0.95]$, $N=100$, $G=1.0E+8$
GRABC	工蜂或侦察蜂 $=100$, 测试次数 $=100n$, $G=1.0E+8$

表4 OA-PPD 算法和7种被比较算法求解基准函数 F15~F20 的结果
Tab. 4 Comparison results of OA-PPD and the 7 compared algorithms when they solve benchmark functions F15~F20

基准函数	OA-PPD	RC-GA	DSDA	NP-PSO	MBBO	MDE	SaDE	GRABC
F15								
平均值	-3 521.476 4	-3 521.476 4	-3 521.476 3	-3 521.475 9	-3 521.475 0	-3 521.476 1	-3 521.476 4	-3 521.475 9
耗时/s	31	9	327	445	121	444	504	93
排名1	1	1	1	5	8	7	1	5
排名2	2	1	3	6	8	7	4	5
F16								
平均值	200.000 0	200.000 0	200.000 1	200.000 1	200.000 3	200.000 1	200.000 0	200.000 1
耗时/s	29	6	1946	1844	141	394	552	479
排名1	1	1	1	5	8	5	1	5
排名2	2	1	4	7	8	5	3	6
F17								
平均值	300.000 0	300.005 1	300.003 8	300.001 8	302.024 7	300.043 2	300.009 6	300.000 0
耗时/s	2	7	653	552	489	239	625	0
排名1	1	5	4	3	8	7	6	1
排名2	2	5	4	3	8	7	6	1
F18								
平均值	400.000 0	400.000 0	400.000 2	400.000 0	682.5197	400.121 4	400.000 0	400.001 3
耗时/s	27	3	1	570	188	18	647	26

排名 1	1	1	5	1	8	7	1	6
排名 2	3	1	5	2	8	7	4	6
F19								
平均值	512.251 7	918.407 0	553.082 3	11 327.083 4	538.540 3	502.154 8	502.082 4	528.232 6
耗时/s	19 464	10 150	3 123	10 328	2 351	199	1 296	720
排名 1	3	7	6	8	5	2	1	4
排名 2	3	7	6	8	5	2	1	4
F20								
平均值	601.489 7	608.581 8	603.695 4	601.902 6	605.115 0	613.027 8	613.953 0	611.739 4
耗时/s	813	3701	716	499	4 546	400	99	553
排名 1	1	5	3	2	4	7	8	6
排名 2	1	5	3	2	4	7	8	6
排名总分 1	8	20	20	24	41	35	18	27
排名总分 2	13	20	25	28	41	35	26	28
最终排名 1	1	3	3	5	8	7	2	6
最终排名 2	1	2	3	5	8	7	4	5

表 5 Wilcoxon 秩和检验结果比较 ($\alpha=0.01$)
Tab. 5 Results comparison of Wilcoxon rank sum test ($\alpha=0.01$)

基准函数	检验结果	OA-PPD vs.						
		RC-GA	DSDA	NP-PSO	MBBO	MDE	SaDE	GRABC
F15	<i>p</i> -value	3.702 5E-01	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	7.794 8E-01	8.854 9E-05
	<i>h</i> -value	0	1	1	1	1	0	1
	<i>z</i> -val	-0.896	-3.920	-3.920	-3.920	-3.920	-0.280	-3.920
F16	<i>p</i> -value	1	1.401 0E-04	8.854 9E-05	8.854 9E-05	8.854 9E-05	1	8.854 9E-05
	<i>h</i> -value	0	1	1	1	1	0	1
	<i>z</i> -val	0.000	-3.808	-3.920	-3.920	-3.920	0	-3.920
F17	<i>p</i> -value	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	1
	<i>h</i> -value	1	1	1	1	1	1	0
	<i>z</i> -val	-3.920	-3.920	-3.920	-3.920	-3.920	-3.920	0.000
F18	<i>p</i> -value	1	8.854 9E-05	1	8.854 9E-05	8.854 9E-05	1	8.854 9E-05
	<i>h</i> -value	0	1	0	1	1	0	1
	<i>z</i> -val	0	-3.920	-3.920	-3.920	-3.920	0	-3.920
F19	<i>p</i> -value	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05
	<i>h</i> -value	1	1	1	1	-1	-1	1
	<i>z</i> -val	-3.920	-3.920	-3.920	-3.920	-3.920	-3.920	-3.920
F20	<i>p</i> -value	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05	8.854 9E-05
	<i>h</i> -value	1	1	1	1	1	1	1
	<i>z</i> -val	-3.920	-3.920	-3.920	-3.920	-3.920	-3.920	-3.920
1(优于)		3	6	5	6	5	2	5
0(相同)		3	0	1	0	0	3	1
-1(劣于)		0	0	0	0	-1	1	0

从表 4 我们可以看出 OA-PPD, RC-GA, DSDA, NP-PSO, MBBO, MDE, SaDE 和 GRABC 的排序如下:

(1) 按平均最优目标函数值最终排序如下:
OA-PPD>SaDE>RC-GA=DASA>NP-PSO>GR
ABC>MDE>MBBO

(2) 按平均最优目标函数值和耗时排序如下:

OA-PPD>RC-GA>DASA>SaDE>NP-PSO=GR
ABC>MDE>MBBO

从表5我们可以知道, OA-PPD的性能明显优于7种被比较算法。

图1(a)~(f)说明 OA-PPD, RC-GA, DSDA, NP-PSO, MBBO, MDE, SaDE 和 GRABC 算法求解基准函数 F15~F20 时的样本收敛曲线。为了突出这些样品的收敛曲线的变化, 水平和垂直轴采用对数刻度。

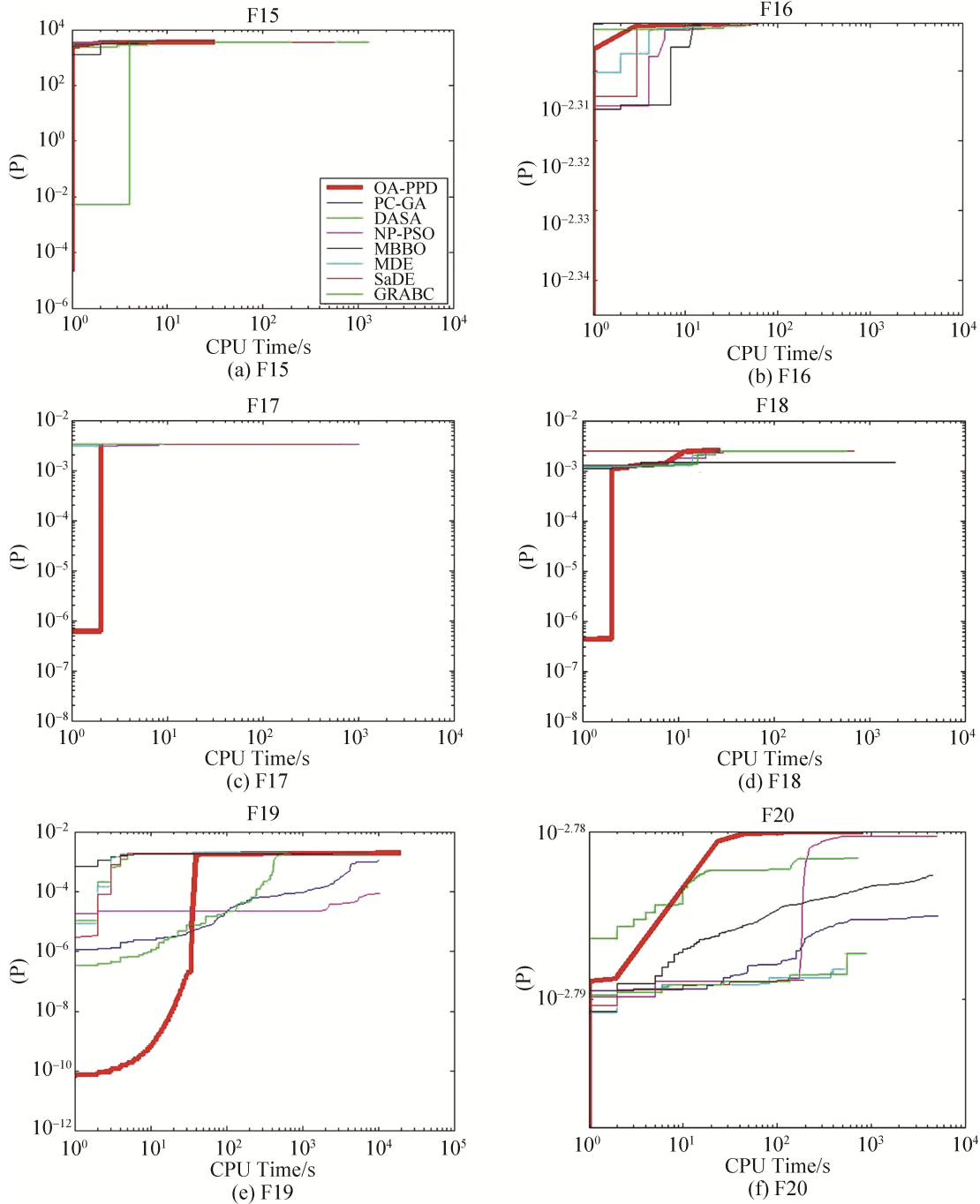


图2 求解 F15~F20 时的收敛曲线

Fig. 2 Convergence curves when solving F15~F20

3 结论

本文和现有算法相比, 具有如下优点:

(1) 本文提出了一种具有全局收敛性的新型优化算法—OA-PPD算法, 该算法采用生态系统动力学中的捕食-被食理论, 该理论与其它算法所采用的理论完全不同。该算法假设在某个生态系统中生长有捕食者种群和被食者种群, 同类种群内部存在相互竞争关系, 不同类种群之间存在捕食-被食关系。利用该生态系统中的同类种群内竞争及捕食-被食关系, 构造出了竞争算子和捕食-被食算子。本算法具有搜索能力强和全局收敛性的特点, 为复杂优化问题的求解提供了一种解决方案。

(2) OA-PPD 算法的搜索能力很强。OA-PPD 算法包括竞争算子和捕食-被食算子, 这些算子增加了其搜索能力。

(3) 模型参数取值简单。采用随机方法确定算法中竞争算子、捕食-被食算子中的相关参数, 既大幅减少了参数输入个数, 又使模型更能表达实际情况。

(4) 捕食种群和被食种群数量的增减相当于搜索空间的试探解从一个位置跳到另外一个位置, 密度高的种群内部的竞争越激烈, 它们的生长特征更容易得到更新, 即更容易逼近最优值。密度高的被食者种群更容易被捕食者种群捕食, 因此, 捕食者种群的生长特征更容易更新, 也即更容易逼近最优值。

(5) OA-PPD 算法体现出了生态系统中常见的不同类型的种群间的复杂种间竞争和捕食-被食关系。

参考文献:

- [1] Y Zhang, D W Gong, Z H Ding. Handling multi-objective optimization problems with a multi-swarm operative particle swarm optimizer [J]. *Expert Systems with Applications* (S0957-4174), 2011, 38(11): 13933-13941.
- [2] Y C Chuang, C T Chen, C Hwang. A simple and efficient real-coded genetic algorithm for constrained optimization [J]. *Applied Soft Computing* (S1687-9724), 2016, 38(3): 87-105.
- [3] P Korošec, Jurij Šilc, Bogdan Filipic. The differential ant-stigmergy algorithm [J]. *Information Sciences* (S0020-0255), 2012, 192(5): 82-97.
- [4] Z Beheshti, S M Shamsuddin. Non-parametric particle swarm optimization for global optimization [J]. *Applied Soft Computing* (S1568-4946), 2015, 28(4): 345-359.
- [5] R Mukherjee, S Debchoudhury, S Das. Modified Differential Evolution with Locality induced Genetic Operators for dynamic optimization [J]. *European Journal of Operational Research* (S0377-2217), 2016, 253(2): 337-355.
- [6] Z W Zhao, J M Yang, Z Y Hu, et al. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems [J]. *European Journal of Operational Research* (S0377-2217), 2016, 250(1): 30-45.
- [7] A R Al-Roomi, M E El-Hawary. Metropolis biogeography-based optimization [J]. *Information Sciences* (S0020-0255), 2016, 360(3): 73-95.
- [8] G H Li, L Z Cui, X H Fu, et al. Artificial bee colony algorithm with gene recombination for numerical function optimization [J]. *Applied Soft Computing* (S1568-4946), 2017, 52(6): 146-159.
- [9] 贺玫瑰, 罗杰. 基于模拟退火机制的精英协同进化算法 [J]. *计算机技术与发展*, 2015, 25(1): 91-95.
He Meilu, Luo Jie. Elite Co-evolutionary Genetic Algorithm Based on Simulated Annealing Mechanism [J]. *Computer Technology and Development*, 2015, 25(1): 91-95.
- [10] 王丽萍, 彭骏, 邱飞岳, 等. 基于多偏好自适应协同的高维目标进化算法 [J]. *小型微型计算机系统*, 2016, 37(6): 1308-1312.
Wang Liping, Peng Jun, Qiu Feiyue, et al. Many-objective Optimization Via Cooperative Coevolution with Adaptive Preferences [J]. *Journal of Chinese Computer Systems*, 2016, 37(6): 1308-1312.
- [11] 戚玉涛, 刘芳, 任元, 等. 基于合作模型的协同免疫多目标优化算法 [J]. *电子学报*, 2014, 42(5): 858-867.
Qi Yutao, Liu Fang, Ren Yuan, et al. A Cooperative Immune Co-evolutionary Algorithm for Multi-Objective Optimization [J]. *Acta Electronica Sinica*, 2014, 42(5): 858-867.
- [12] 张睿哲, 杨照峰. 基于博弈模型的合作式粒子群优化算法 [J]. *计算机系统应用*, 2014, 23(6): 170-174.

