



中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION



基于近邻牵引算子的离散黑猩猩优化算法

沈孝凯, 张纪会, 郭乙运, 张保华

引用本文:

沈孝凯,张纪会,郭乙运,张保华. 基于近邻牵引算子的离散黑猩猩优化算法[J]. 控制与决策, 2024, 39(4): 1133–1141.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.1417>

您可能感兴趣的其他文章

Articles you may be interested in

基于混沌“微变异”自适应遗传算法

Adaptive genetic algorithm based on chaos “micro variation”

控制与决策. 2021, 36(8): 2042–2048 <https://doi.org/10.13195/j.kzyjc.2021.0319>

一种自适应拟牛顿-状态转移混合智能优化算法及应用

A hybrid state transition optimization algorithm based on adaptive quasi-newton method and its application

控制与决策. 2021, 36(10): 2451–2458 <https://doi.org/10.13195/j.kzyjc.2020.0214>

区间数可重入混合流水车间调度与预维护协同优化

Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance

控制与决策. 2021, 36(11): 2599–2608 <https://doi.org/10.13195/j.kzyjc.2020.0973>

求解约束优化问题的改进果蝇优化算法及其工程应用

Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications

控制与决策. 2021, 36(2): 314–324 <https://doi.org/10.13195/j.kzyjc.2019.0557>

双重驱动的果蝇优化算法及其在PID控制器中的应用

Double drive fruit fly optimization algorithm and its application in PID controller

控制与决策. 2021, 36(9): 2225–2233 <https://doi.org/10.13195/j.kzyjc.2020.0046>

基于近邻牵引算子的离散黑猩猩优化算法

沈孝凯^{1,2}, 张纪会^{1,2†}, 郭乙运³, 张保华³

(1. 青岛大学自动化学院, 山东 青岛 266071; 2. 山东省工业控制技术重点实验室, 山东 青岛 266071; 3. 青岛港国际股份有限公司, 山东 青岛 266011)

摘要: 针对旅行商问题的特点, 提出基于近邻牵引算子的离散黑猩猩优化算法. 首先, 引入优质片段的概念, 并结合每个群组的最优个体设计其检索方法, 以提高组内学习策略的效果, 根据组合优化问题特点对黑猩猩群体的狩猎过程进行离散化表示; 其次, 通过组间交流机制消除部分个体路径交叉; 最后, 为了克服传统的邻域搜索算子收敛慢和搜索效率低的缺点, 提出一种新的邻域搜索方式——近邻牵引算子, 其搜索目的更加明确、收敛更高效, 并设计自适应概率扰动调控策略, 以有效平衡算法的探索与开发. 对30个TSP标准数据集进行实验, 结果表明, 所设计的离散黑猩猩优化算法求解质量高、收敛速度快, 可以应用于组合优化问题求解.

关键词: 离散黑猩猩优化算法; 优质片段; 近邻牵引算子; 自适应概率扰动调控; 旅行商问题

中图分类号: TP301 文献标志码: A

DOI: 10.13195/j.kzyj.2022.1417

引用格式: 沈孝凯, 张纪会, 郭乙运, 等. 基于近邻牵引算子的离散黑猩猩优化算法[J]. 控制与决策, 2024, 39(4): 1133-1141.

Discrete chimp optimization algorithm based on neighbour traction operator

SHEN Xiao-kai^{1,2}, ZHANG Ji-hui^{1,2†}, GUO Yi-yun³, ZHANG Bao-hua³

(1. School of Automation, Qingdao University, Qingdao 266071, China; 2. Shandong Key Laboratory of Industrial Control Technology, Qingdao 266071, China; 3. Qingdao Port International Company Co., Ltd, Qingdao 266011, China)

Abstract: According to the characteristics of the traveling salesman problem, a discrete chimp optimization algorithm based on a nearest neighbour traction operator is proposed. Firstly, we introduce the concept of high-quality fragment, and design a retrieval method combining with the optimal individuals of each group to improve the efficiency of the learning strategy with intra-group, and discretize the hunting process of chimpanzee groups in combination with the characteristics of combinatorial optimization problems. Then, through an inter-group communication mechanism to eliminate path crossing of some individuals. Finally, in order to overcome the shortcomings of slow convergence and low search efficiency of the traditional neighborhood search operators, a new neighborhood search method, the neighbour traction operator, is proposed, which has clearer search purpose and more efficient convergence, and an adaptive probabilistic disturbance control strategy is designed to effectively balance the exploitation and exploration of the algorithm. Experiments are conducted on 30 instances from the TSP standard datasets. The experimental results show that the designed discrete chimp optimization algorithm has high solution quality and fast convergence speed, and can be applied to the solution of combinatorial optimization problems.

Keywords: discrete chimp optimization algorithm; high-quality fragment; neighbour traction operator; adaptive probabilistic disturbance control; traveling salesman problem

0 引言

旅行商问题(the traveling salesman problem, TSP)是经典的组合优化问题, 该问题在车辆路径规划、作

业排序、物流配送等许多领域均有广泛的应用, 因此, 对TSP的有效求解方法的研究有着深远的理论意义和实践价值.

收稿日期: 2022-08-06; 录用日期: 2022-12-20.

基金项目: 国家自然科学基金项目(61673228, 62072260); 青岛市科技计划项目(21-1-2-16-zh).

责任编委: 陈家伟.

†通讯作者. E-mail: zhangjihui@qdu.edu.cn.

*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

TSP是典型的NP-hard问题^[1],随着问题规模的扩大,求解复杂度快速增加. TSP求解方法主要分为精确算法和近似算法两大类. 传统的精确算法,如分支定界法^[2]和动态规划法^[3],可以求得问题的精确解,但是,对于大规模问题,精确算法往往无能为力. 近似算法包括启发式算法和元启发式算法,可以在较短时间内求得大规模问题的满意解,因此,受到众多学者的青睐,但是,这类算法的一个显著缺点是无法保证解的质量. 在文献中有很多典型的近似算法应用于TSP的求解,如模拟退火算法^[4]、蚁群算法^[5]、遗传算法^[6]和粒子群算法^[2]. 同时,许多新型的智能算法在求解TSP方面也具有很大潜力,包括灰狼算法^[8]、蝙蝠算法^[9]、帝国竞争算法^[10]等.

近似算法在求解TSP时虽然具有一定的全局寻优能力,但存在收敛速度慢且易陷入局部最优等缺点. 为此,许多学者提出了混合算法,加入扰动机制或局部搜索算子,以跳出局部最优并加速算法收敛. 在混合算法方面,文献[11]提出了一种基于模拟退火的自适应离散型布谷鸟算法,在离散算法中引入模拟退火机制,依概率接受差解,防止算法陷入局部最优;文献[12]提出了一种新型的混合群体智能优化算法,利用遗传算法和粒子群优化的随机性、快速性和整体性,获得一系列次优解,以调整蚁群算法中信息素浓度,提高计算效率. 在使用扰动机制和局部搜索算子方面,文献[13]设计了结合高斯变异和交换算子的全局扰动机制来平衡算法的探索与利用能力;文献[14]引入交换、移位和对称变换算子,对经典的灰狼优化算法进行了重新设计,以提高算法的局部搜索能力;文献[15]提出了多组离散共生生物搜索算法,使用转移变异算子优化群体之间的学习能力,提高算法的收敛能力;文献[16]提出了一种求解旅行商问题的离散狼群算法,设计召唤行为时,从头狼染色体中随机选取一定长度的位置编码基因,替换猛狼个体中相同开头或结尾的一段等长染色体片段,以此加速算法收敛. 虽然文献[11-12]融合了多种算法,但仍存在收敛较慢的缺点,求解TSP效果有限;文献[13]采取的扰动机制和两交换算子解决小规模TSP时效果显著,但求解大规模问题时花费时间过长;文献[14]采用的3种邻域搜索算子存在过多无效扰动,求解大规模TSP时效果有限;文献[15-16]采用了随机选取优秀个体某一部分与其他个体进行结合的方式,该方法存在较大的盲目性,影响算法收敛速度.

扰动机制和邻域搜索算子在求解TSP中扮演着重要角色,由此,本文提出一种基于近邻牵引算

子的离散黑猩猩优化算法. 黑猩猩优化算法(chimp optimization algorithm, ChOA)是由Khishe等^[17]于2020年提出的一种新型元启发式优化算法,其灵感来自黑猩猩群体狩猎中的个体智力和社会激励行为,该算法的设计初衷是求解连续空间函数优化问题,目前尚未发现有人将其应用于TSP求解中. 针对智能算法求解TSP时离散设计存在扰动的盲目性和随机性的缺点,本文设计一种结合优秀个体检索优质片段的方法,以及组内学习的扰动机制,实现个体的有效扰动;针对现有邻域搜索算子存在过多无效扰动的问题,设计高效扰动近邻牵引搜索算子,能在短时间内逼近最优解,极大提高算法收敛能力;同时,提出自适应概率扰动调控策略,有效防止陷入局部最优,提高求解精度;采用群体组间交流的方式加速算法收敛,提高求解效率.

1 黑猩猩优化算法

黑猩猩优化算法是根据黑猩猩群体狩猎行为而提出的一种元启发式算法,黑猩猩群体通过分工,各司其职地寻找猎物. 在群体中有4种类型的黑猩猩,分别是:攻击者(attackers)、阻碍者(barriers)、追赶者(chasers)和驱逐者(drivers). 攻击者黑猩猩占据领导者地位,社会地位最高,其他3类主要协助攻击者,社会地位依次递减. 黑猩猩优化算法的基本流程如下.

在 D 维搜索空间中,假设 n 只黑猩猩个体组成的种群, $X = \{X_1, X_2, \dots, X_n\}$,定义第 i 只黑猩猩的位置为 $X_i = (X_i^1, X_i^2, \dots, X_i^D)^T$, X_i^j 为第 i 只黑猩猩第 j 维的分量($j = 1, 2, \dots, D$). 在驱赶和追逐猎物的过程中,黑猩猩 i 接近猎物的行为可以用下式描述:

$$d_i^j = |c_i^j \cdot X_p^j(t) - m_i^j \cdot X_i^j(t)|, \quad (1)$$

$$X_i^j(t+1) = X_p^j(t) - a_i^j \cdot d_i^j. \quad (2)$$

其中: t 为当前的迭代次数; d_i^j 表示第 i 只黑猩猩在第 j 维上与猎物之间的距离; X_p^j 为第 j 维上猎物的位置; c_i^j 、 a_i^j 为系数, m_i^j 为混沌数,可由下式计算得到:

$$c_i^j = 2 \cdot r_2, \quad (3)$$

$$m_i^j = \text{Chaotic_value}, \quad (4)$$

$$a_i^j = 2 \cdot f \cdot r_1 - f, \quad (5)$$

$$f = 2.5 - \frac{2.5 \cdot t}{T}. \quad (6)$$

r_1 和 r_2 是区间 $[0, 1]$ 上均匀分布的随机数; f 为衰减因子,随着迭代次数的增加从2.5非线性递减至0;随机变量 $a_i^j \in [-f, f]$, $|a_i^j| \leq 1$ 表示黑猩猩个体攻击猎物 X_p^j , $|a_i^j| > 1$ 表示黑猩猩个体偏离猎物位置,向更广的范围内搜索; T 为最大迭代次数;Chaotic_value为

通过某种混沌映射得到的数值,表示黑猩猩个体受到的社会激励对狩猎行为的影响; c_i^j 为控制猎物与黑猩猩个体距离的系数,是区间 $[0, 2]$ 上均匀分布的随机数。

狩猎过程通过更新攻击者、阻碍者、追赶者和驱逐者4个最优个体来侦察猎物的位置,并由攻击者攻击猎物,其他黑猩猩根据4个个体位置对其位置进行更新以参与围捕,此过程由下列各式描述:

$$X_{i,1}^j = X_A^j - a_{i,1}^j \cdot |c_{i,1}^j \cdot X_A^j - m_{i,1}^j \cdot X_i^j|, \quad (7)$$

$$X_{i,2}^j = X_B^j - a_{i,2}^j \cdot |c_{i,2}^j \cdot X_B^j - m_{i,2}^j \cdot X_i^j|, \quad (8)$$

$$X_{i,3}^j = X_C^j - a_{i,3}^j \cdot |c_{i,3}^j \cdot X_C^j - m_{i,3}^j \cdot X_i^j|, \quad (9)$$

$$X_{i,4}^j = X_D^j - a_{i,4}^j \cdot |c_{i,4}^j \cdot X_D^j - m_{i,4}^j \cdot X_i^j|, \quad (10)$$

$$X_i^j(t+1) = \frac{(X_{i,1}^j + X_{i,2}^j + X_{i,3}^j + X_{i,4}^j)}{4}. \quad (11)$$

其中: X_A^j 、 X_B^j 、 X_C^j 、 X_D^j 分别为第 t 代攻击者、阻碍者、追赶者和驱逐者的第 j 维的分量, $X_i^j(t+1)$ 是 $t+1$ 代个体的第 j 维分量。

在ChOA迭代过程中,首先根据式(3)和(4)计算系数 c_i^j 和混沌系数 m_i^j ,按照式(5)和(6)更新 a_i^j ;然后通过式(7)~(11)更新每个个体的位置;最后计算种群适应度值,输出本次迭代后的前4个最优个体attacker、barrier、chaser和driver,并利用它们的位置来引导整个黑猩猩种群的下一次迭代位置的更新.重复以上过程直到满足终止条件,输出最优解。

2 求解TSP的离散黑猩猩优化算法设计

原始的黑猩猩优化算法适用于求解连续优化问题,而TSP是一个组合优化问题,因此,为了设计离散黑猩猩优化算法(DChOA)求解TSP,需要合适的离散策略对原始算法进行转换。

2.1 表示方式

解的表示方法是求解TSP的关键问题之一,常见的表示形式包括:二进制表示、路径表示、邻接关系表示、序数表示和矩阵表示等^[18].本文采用整数编码路径表示方法。

2.2 群组初始化

ChOA求解连续优化问题时,将黑猩猩分成独立的组以保证对搜索空间的探索.基于此,DChOA随机初始化种群后,根据个体编号,依顺序划分为4个独立群组,划分的标准为前3个子群中个体数量均为 $\text{int}(n/4)$,剩余个体放入最后一个子群, n 为种群规模.群组初始化是用来划定攻击者、阻碍者、追逐者和驱赶者四类个体所在区域,有利于其他黑猩猩个体

依照四类领导者进行自我更新。

2.3 组内学习

本阶段模拟黑猩猩个体在狩猎过程中服从领导的学习行为以及受到社会激励后出现的混乱行为.首先,选取每个群组内适应度值最大的个体作为领导者,采用片段替换机制实现跟随领导者的学习行为,体现了优秀个体对其他个体的指导作用;然后,为防止领导者的交叉较多的劣质片段替换给学习个体,设计了优质基因片段检索方式,可有效提高算法的收敛速度;最后,针对在群体狩猎过程中某些个体在捕获到食物时受到社会激励而忘记狩猎责任的混乱行为,提出了基于个体路径上某一片段的乱序扰动方式,有利于维持种群多样性,防止算法陷入局部最优。

2.3.1 优质片段检索

为了更好地对该过程的离散策略进行描述,先给出一些定义,从技术角度阐述检索方法。

定义1(片段) 染色体 X 上的一个长度为 L 的一段,记为 $W(L)$,如 $X = [3, 6, 2, 1, 4, 5]$,则 $W(3) = [6, 2, 1]$ 表示长度为3的一个片段。

定义2(优质片段) 种群进化过程中,适应度高的优秀个体之间存在的长度为 L 的不计顺逆排序的相同片段.例如,对于两个优秀个体 $X_A = [3, 6, 7, 2, 5, 4, 1]$, $X_B = [6, 7, 2, 3, 1, 4, 5]$,通过某种方式检索出的长度为3的相同片段有 $W_1(3) = [6, 7, 2]$, $W_2(3) = [5, 4, 1]$, $W_3(3) = [1, 4, 5]$ 。

定义3(片段的距离) 指片段 $W(L)$ 从其起始城市到结尾城市的路径长度 $f_L(W)$ 。

定义4(边的标度) $\phi(V_i) = C_\alpha^2 + C_\beta^2$,其中 $V_i = (C_\alpha, C_\beta)$, C_α 、 C_β 分别为该边所连接的左右两城市.边的标度具有唯一性,若以直角三角形两直角边分别表示路径上某个边的左右两个城市 C_1 和 C_2 ,则由三角形勾股定理知 $\phi(V_1)$ 具有唯一性,改变任何一个直角边都会直接改变斜边 $\sqrt{\phi(V_1)}$ 的数值,即 $\phi(V_1)$ 仅对应于边 $[C_1, C_2]$ 或 $[C_2, C_1]$ 。

定义5(片段的标度) $\psi(W_i(L)) = \phi^2(V_i) + \phi^2(V_{i+1}) + \dots + \phi^2(V_{i+L-1})$,其中 $W_i(L)$ 为路径 X 上长度为 L 的第 i 个片段。

定义6(同度检验法) 对群体中的多个适应度较高的优秀个体,通过计算每个个体中长度为 L 的片段的标度,运用片段的标度找出优质片段的方法。

定义7(多片段比较法) 在一条路径上随机选取若干等长度片段,通过比较,保留 $f_L(W)$ 小的片段的方法。

综上,优质片段检索就是指通过同度检验法检索

出优秀个体之间存在一定长度的相同片段,如图1所示,(a)和(b)为可行解中的两个优秀个体,如果指定检索片段长度为4,则通过同度检验法即可检索出它们中的优质片段[1,7,6,3].在算法前期,优秀个体之间可能存在无相同片段情况,则对每个群组的最优个体分别采用多片段比较法选取优质片段,目的是防止存在交叉过多的劣质片段被选中用于组内学习.

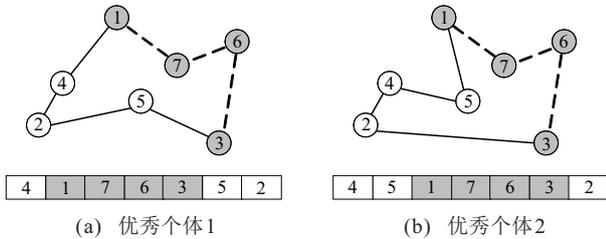


图1 7个城市节点的TSP可行解路径示例

2.3.2 学习机制

结合每个群组的最优个体,通过使用同度检验法找出它们路径上共有的一条长度为 L 的优质片段,或针对每个群组的最优个体采用多片段比较法找出相对较好的优质片段,用此片段替换掉组内其他黑猩猩个体的相同城市为起点的等长片段,对非替换片段出现重复的基因做映射,使得路径上的所有城市仅被访问一次.例如, $X_1 = [3, 5, 1, 4, 6, 8, 2, 7]$, $X_2 = [6, 7, 2, 1, 5, 8, 3, 4]$, $L = 3$,从 X_1 中随机选取一段 $[1, 4, 6]$ 替换 X_2 中的 $[1, 5, 8]$,得到 $X'_2 = [6, 7, 2, 1, 4, 6, 3, 4]$,通过映射关系6-8、4-5调整后得 $X_2^{new} = [8, 7, 2, 1, 4, 6, 3, 5]$.该学习机制既保留了学习者的大部分基因,又融合了领导者的优质基因片段.

2.3.3 乱序扰动

以概率选取个体 X 为受到社会激励的黑猩猩,在其路径上随机选取一条长度为 L 的片段 X_L ,同时生成一个区间 $[1, L]$ 上的非重复随机整数序列 X_S ,并用序列 X_S 调整片段 X_L 上的基因位置,得到一个新的片段 X'_L ,最后,返回 X'_L 并更新 X ,得到新解 X' ,如图2所示.

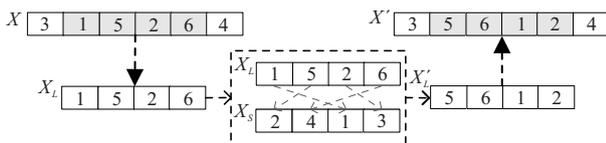


图2 乱序扰动策略

2.4 组间交流

为了提高算法效率,本文提出了群组间个体信息交流策略,以检查并消除个体的部分路径交叉问题,具体操作步骤如下:

1) 依据每个群组的个体适应度值,采用轮盘赌策

略选出每个组要参与交流的个体,这里的适应度为个体路径长度的倒数,每个群组的个体被选择进行交流的概率为

$$P_i = \frac{f(x_i)}{\sum_{j=1}^{\mu} f(x_j)} \quad (12)$$

其中: P_i 为群组内第 i 个个体被选中的概率, x_i 为第 i 个个体, $f(x_i)$ 为第 i 个个体的适应度值, μ 为群组中个体的数量.

2) 对所有群组选出的个体进行适应度比较,用适应度最大的个体替换其他个体,并将它们分别返回到自己群组中.

2.5 局部扰动

作为局部扰动机制,本文设计了新的搜索方法,称为近邻牵引算子(neighbour traction operator, traction),并结合交换、插入和反转这3种经典的邻域算子进行邻域搜索.同时,为了使组间交流策略与局部扰动能够协调使用,有效平衡算法的全局与局部搜索,本文提出了自适应概率扰动调控策略,以概率进行策略和扰动方式的选择,使得算法前期扰动幅度大,种群多样性得以保持,防止算法陷入局部最优;在算法中后期,向局部搜索倾斜,使依据调控策略赋予的组间交流策略使用概率得到提高,增强算法的开发能力,以提高算法收敛速度.

2.5.1 近邻牵引算子

定义8(近邻牵引算子) 在一条 N 个城市的路径中,随机选择一个城市 A ,将其他城市按照与 A 的距离由小到大排列,截取前 S 个城市作为 A 的近邻备选城市集合 P , $P = \{C_1, C_2, \dots, C_S\}$, $S \leq 20$,这是因为目前已知所有TSP实例中最优路径的任意两个相邻城市都在彼此较近的前20个城市之内^[19].最后,从集合 P 中随机选择两个城市 C_L 和 C_R 分别插入到

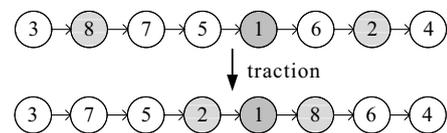


图3 近邻牵引算子搜索策略

表1 8个城市之间距离

城市	1	2	3	4	5	6	7	8
1	∞	4	5	7	11	5	4	1
2	4	∞	6	5	7	2	2	3
3	5	6	∞	3	12	5	3	5
4	7	5	3	∞	10	4	3	6
5	11	7	12	10	∞	7	8	10
6	5	2	5	4	7	∞	1	4
7	4	2	3	3	8	1	∞	3
8	1	3	5	6	10	4	3	∞

城市A的左右两端. 如图3所示, 从可行解中随机选择城市1, 然后在表1中找出城市1对应的近邻城市备选集合8, 7, 2, 3, 6, 从中随机选取城市2和城市8, 并将其分别插入到城市1的左右两端后得一个新解.

2.5.2 3种邻域搜索算子

交换算子 (swap): 从路径上随机选择两个城市, 将两城市进行位置互换.

插入算子 (insertion): 从路径上随机选择两个城市, 将第1个城市插入到第2个城市位置后面.

反转算子 (reversion): 从路径上随机选择两个城市, 将两个城市之间的城市元素 (包含该两个) 进行逆序排列.

2.5.3 自适应概率扰动调控策略

为了更好地平衡算法的探索与开发, 根据邻域算子对解的不同扰动程度, 自适应地对其选择使用: 在算法前期, 多采取较大幅度的高效扰动以加速收敛; 针对后期多优质个体, 则以进行微调的精确搜索为主, 提高求解精度. 因此, 依概率选择两种扰动方式产生新解, 即当 $\text{rand} < p$ 时, 使用 traction 算子进行邻域搜索; 反之, 则依次进行 swap、insertion 和 reversion 等扰动贪婪选择来产生新解. 其中: rand 为区间 $[0, 1]$ 上均匀分布的随机数; p 为收敛因子, 平衡两种扰动方式的选取概率, 由下式计算所得:

$$p = 1 - \left(\tan \left(\frac{\pi \cdot \text{iter}}{4 \cdot \text{max_iter}} \right) \right)^U \quad (13)$$

iter 为当前迭代次数; max_iter 为最大迭代次数; U 为控制因子, 取值区间为 $(0, 0.5]$, 控制 p 的衰减幅度, U 越大 p 衰减越慢, 对应算法前期新解由第1类扰动方式产生的比重越大, 反之越小, 如图4所示.

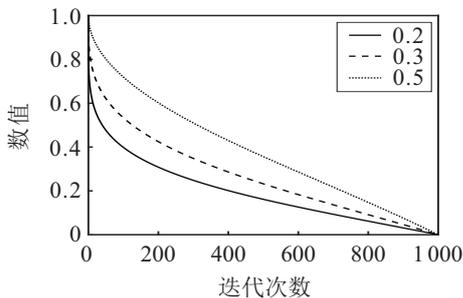


图4 U 的不同取值下 p 值的收敛曲线对比

2.6 算法步骤

综上所述, 本文提出的离散黑猩猩优化算法具体步骤如下, 设 n 为黑猩猩种群规模, max_iter 为算法运行的最大迭代次数, stop_iter 为适应度连续保持不变的迭代次数, inter_iter 为局部扰动迭代次数, δ 为组内学习率, BL 为优质片段长度, Q 为近邻备选城市个数, 算法终止条件为攻击者黑猩猩个体适应度值连续

stop_iter 代保持不变或达到最大迭代次数 max_iter.

step 1: 初始化 max_iter、inter_iter、 δ 、BL 和 Q 等参数, 初始化种群以及初始化群组.

step 2: 计算种群中所有个体适应度, 确定群组中的攻击者、阻碍者、追赶者和驱逐者.

step 3: 对4个领导者个体按2.3.1节的方法进行优质片段检索.

step 4: 按2.3.2节学习机制设置一定的学习率 δ , 对每个个体都产生一个区间 $[0, 1]$ 上均匀分布的随机数 r_1 , 若 $\delta > r_1$, 则对个体进行优质片段替换; 否则, 采用2.3.3节方式进行乱序扰动.

step 5: 产生一个区间 $[0, 1]$ 上均匀分布的随机数 r_2 , 若 $r_2 > p$, 则采用式(12)轮盘赌方法在每个群组中选取一个个体, 对选中的个体执行2.4节的组间交流操作.

step 6: 根据2.5.3节自适应概率扰动调控策略进行邻域搜索.

step 7: 如果算法达到终止条件, 则输出全局最优解; 否则, 重复 step 2 ~ step 6.

3 实验及结果分析

为了验证所提出离散黑猩猩优化算法的有效性, 使用标准 TSPLIB 数据集进行仿真实验. 实验环境为: Matlab2018b, Win10操作系统, Intel (R) Core (TM) i5-11300H CPU, 16 GB 内存. 实验参数设置如下: $n = 20$, $\text{max_iter} = 1000$, $\text{stop_iter} = 50$, $\text{inter_iter} = 300$, $\delta = 0.5$, BL、近邻备选城市个数 Q 以及控制因子 U 等参数会因城市规模不同而取值不一, 其设置如表2所示.

表2 DChOA 部分参数设置

城市规模 N	$N \leq 200$	$200 < N \leq 700$	$N > 700$
优质片段长度 BL	5	7	10
近邻备选城市个数 Q	3	3	5
控制因子 U	0.2	0.3	0.5

3.1 近邻牵引算子的有效性

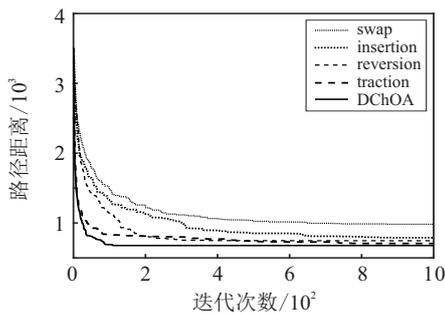
为了验证所提出的近邻牵引算子和自适应概率扰动调控策略的有效性, 采用控制变量法在 DChOA 基础上测试每个算子求解 TSP 实例效果, 同时为了直观地显示每次迭代的适应度值变化, 将局部扰动内循环的次数设置为10, 选择3个不同规模的实例进行测试, 各进行20次实验, 取最优值、平均值和标准差进行比较, 实验结果如表3所示.

由表3结果可以看出: 单独使用4种算子时, 求解效果较好的是反转算子和近邻牵引算子; 而近邻牵

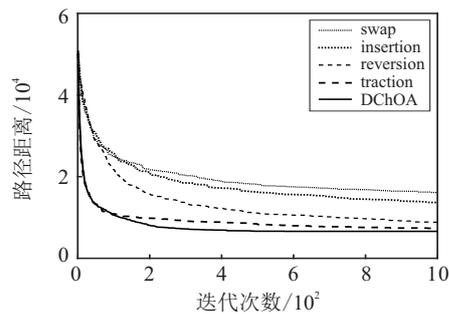
表3 DChOA使用不同的算子求解TSP实例的比较

实例	准则	搜索算子				DChOA
		swap	insertion	reversion	traction	
st 70	最优值	867	734	687	682	675
	平均值	982.15	827.50	719.53	713.62	686.55
	标准差	63.09	47.94	19.93	17.50	5.11
ch 150	最优值	13722	11392	8636	7339	6715
	平均值	15210.94	12607.97	9147.79	8529.37	6825.30
	标准差	575.16	537.60	225.99	430.26	106.23
tsp 225	最优值	10534	9388	6864	4920	3997
	平均值	11962.02	10315.89	7239.22	5457.05	4033.34
	标准差	518.99	438.35	147.81	139.19	40.60

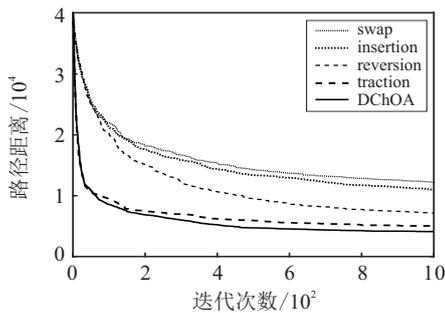
引算子无论最优值还是平均值均优于反转算子,且随着问题规模的扩大,近邻牵引算子求解效果越加显著.依据3个实例的求解结果的平均值和标准差来看,使用的自适应概率扰动调控策略使得算法在求解的精度和稳定性方面更具优势.



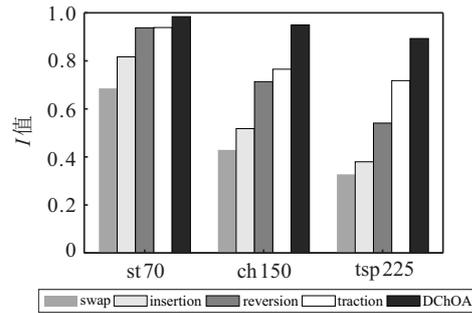
(a) st 70实例的收敛效果对比



(b) ch 50实例的收敛效果对比



(c) tsp 225实例的收敛效果对比



(d) 算子求解的I值对比

图5 不同算子求解TSP实例的收敛效果和I值对比

图5(a)、(b)、(c)是各个算子求解3个实例的迭代过程中目标函数变化情况对比,图5(d)显示了各个算子在求解TSP实例时的平均值与已知最优值的近似程度I的对比.其中

$$I = \frac{\text{optimal}}{\text{average}} \times 100\%, \quad (14)$$

optimal为TSPLIB库中已知的TSP实例最优值,average为算法求得平均值.

由图5(a)、(b)、(c)可以看出:近邻牵引算子在前期能使算法快速收敛,且近邻牵引算子与DChOA收敛曲线几乎重合,说明算法求解过程中,发挥主要作用的是近邻牵引算子;自适应概率扰动调控策略在算法中后期可以进一步提高算法求解精度,一定程度上有利于防止算法陷入局部最优.由图5(d)中I值的对比可以看出,自适应概率调控扰动设置求解的均值与最优值的近似程度最高,近邻牵引算子次之.通过本实验验证了所提出的近邻牵引算子和自适应概率扰动调控策略的有效性.

3.2 近邻牵引算子的通用性

将上述实验中邻域算子加入到SA中做实验,SA的外循环设置为1000,内循环为500,降温系数为0.99;测试数据采用上述实验的TSP实例,图6是各算子求解3个实例的目标函数变化情况对比.

由图6可以明显看出,近邻牵引算子相比其他搜索算子,不仅求解精度高,而且收敛速度极快,在算法前期就可以迅速逼近最优值.结果说明近邻牵引算

子不仅在DChOA上具有优势,与其他智能优化算法(如SA)相比,求解TSP时也具有显著效果.

3.3 验证所提出的策略

在实验设计上:一是采取逐步增加策略对比的方式,即DA 1为只有局部扰动的DChOA,DA 2为具有局部扰动和组内学习机制的DChOA,DA 3为具有组间交流机制的DChOA;二是为了更好地做对比,最大迭代次数取500,局部扰动迭代次数为100.选取4

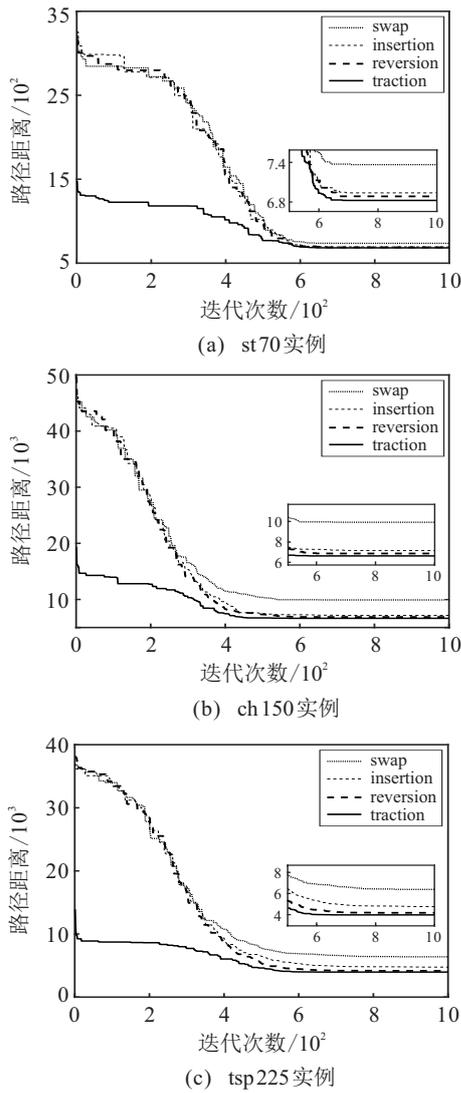


图6 不同算子在SA上求解TSP实例的收敛效果对比

个较大规模的TSP实例,观察算法在500次迭代中的收敛状态,并避免因城市个数少而过早求得最优解不利于实验的对比,每个策略的实验独立运行20次,取最优值、平均值和平均值偏差率(式(15))进行比较,实验结果如表4所示。

表4 DChOA使用不同的策略求解TSP实例的比较

策略	实例	pr 264	lin 318	att 532	rat 783
	理论最优值	49 135	42 029	86 729	8 806
DA 1	平均值	54 404.05	47 496.10	112 365.20	13 451.30
	最优值	53 298	46 513	109 583	13 198
	均值偏差率/%	10.73	13.01	29.56	52.76
	平均值	54 278.45	47 370.50	110 946.20	13 163.10
DA 2	最优值	52 870	46 130	108 023	12 770
	均值偏差率/%	10.47	12.71	27.92	49.48
	平均值	53 685.23	44 908.36	92 214.70	11 652.10
DA 3	最优值	51 556	43 913	95 969	11 208
	均值偏差率/%	9.26	6.85	6.33	32.32

依据表4的实验结果,从整体上看,三者实例

pr 264 的求解结果无明显差距,但随着问题规模的扩大,无论是最优值还是平均值偏差率,DA 3 都表现出明显的优势.对比 DA 1 和 DA 2 的实验结果可知,组内学习策略在一定程度上可以加速算法的收敛. DA 3 的实验效果优于 DA 2,说明组间交流方式通过消除部分个体的路径交叉可以加速算法收敛并提高求解精度,从而进一步验证了所提出的组内学习和组间交流策略的有效性。

3.4 优质片段检索的有效性

为了验证所设计的检索优质片段用以组内学习的离散化方法的有效性,与随机选取等长度片段进行组内学习的方法做对比,即 RDA 2 为在每个群组中当前最优个体染色体上随机截取一片段进行组内学习的方式.为了更好地将两者进行对比,将实验参数中的最大迭代次数设置为500,局部扰动迭代次数设置为10,选取3个不同规模的TSP实例,并记录两种方法在每一代求得的路径距离.通过 RDA 2 和 DA 2 求解 TSP 实例的收敛效果对比可以看出,DA 2 的整体收敛速度快于 RDA 2,能够在相同代数中较早地寻找到质量较优解,即采取优质片段检索方法的 DA 2 可以有效地减少盲目性的片段替换扰动,提高算法的求解速度.通过对 kroA 100 和 att 532 实例求解可以发现,DA 2 在收敛前期的求解速度也略快于 RDA 2,说明作为检索优质片段备用方法的多片段比较法在一定程度上也可以提高算法的收敛速度.由此可见,所设计的优质片段检索方法是有效的。

3.5 DChOA 整体性能

为了全面验证 DChOA 的性能,选取30个不同城市规模的实例进行测试,每个实例实验独立运行30次,记录最优值、最差值、平均值、标准差、最优值偏差率(PDbest)和平均求解时间.其中:平均值偏差率和最优值偏差率的计算公式为

$$PD_{avg} = \frac{\text{average} - \text{optimal}}{\text{optimal}} \times 100\%, \quad (15)$$

$$PD_{best} = \frac{\text{best} - \text{optimal}}{\text{optimal}} \times 100\%. \quad (16)$$

average 为实验的平均值, optimal 为 TSPLIB 库中已知的 TSP 实例最优值, best 为实验最优值。

实验结果表明 DChOA 求解 TSP 具有良好的性能:从求解精度来看,有 33% 的实例已求到理论最优值, 56% 的实例最优偏差率都在 1% 以内,超过 76% 的实例求得结果最优偏差率都在 2% 以内;从求解的标准差来看,整体标准差较小,算法具有较好的稳定性;从算法执行时间来看,大多数实例求解均可在短时间内逼近最优值。

3.6 与经典智能优化算法比较

为了检验所提出算法的性能,将DChOA算法与改进循环交叉算子的遗传算法(GA-CX2)^[20]、蚁群算法(ACO)、基于汉明距离改进的粒子群算法(IMRGHPSO)^[21]、自适应升温模拟退火算法(SA)^[22]等进行对比,算法参数设置均取自原始文献,每个算法在5个不同城市规模的数据集上独立运行20次,结果如表5所示(本文算法的求解数据均加粗表示)。

表5 算法测试结果对比

实例	算法	最优值	平均值
st 70	GA-CX2	695	717.8
	ACO	705	719.0
	IMRGHPSO	679	686.2
	SA	677	681.0
	DChOA	675	680.3
kroA 150	GA-CX2	27 861	28 564.4
	ACO	28 717	29 448.6
	IMRGHPSO	28 024	28 231.2
	SA	27 063	27 284.5
	DChOA	26 979	27 277.3
pr 264	GA-CX2	53 613	56 419.8
	ACO	53 833	54 161.5
	IMRGHPSO	55 162	55 440.6
	SA	51 107	52 839.3
	DChOA	50 357	52 238.8
lin 318	GA-CX2	60 643	62 345.5
	ACO	47 229	47 701.0
	IMRGHPSO	47 834	48 042.8
	SA	44 395	44 786.5
	DChOA	44 136	44 276.5
att 532	GA-CX2	155 135	168 264.0
	ACO	99 269	99 901.2
	IMRGHPSO	106 212	107 023.8
	SA	92 540	94 251.0
	DChOA	91 502	92 712.8

表5实验数据表明,无论是最优值还是平均值,DChOA均优于其他4种算法,具有较好的求解精度

和稳定性.从收敛情况可以看出,在相同的迭代时间内,相对于GA-CX2和SA而言,得益于近邻牵引算子的作用,DChOA算法大约30代以内就能收敛到最优值附近;而GA-CX2中使用的CX2交叉算子搜索存在一定的盲目性;SA在全局寻优方面虽具有一定的优势,但存在过多随机扰动,无法对解空间进行高效探索,故需要大量迭代才能收敛.IMRGHPSO算法性能表现良好,主要是因为算法中加入了2-opt和3-opt扰动,使得算法具备较强的探索能力,但因后期开发能力不足,导致无法找到最优解.在pr 264和att 532实例中,ACO与ChOA算法虽然收敛时间相近,但DChOA求解的精度要高,且ACO明显陷入局部最优解,求解结果偏差较大.整体来看,DChOA求解TSP具有较强的收敛速度和寻优能力.

3.7 与新型群智能优化算法比较

除了与经典算法对比外,本文算法还与其他几种新型群智能算法(包括离散蜘蛛猴算法DSMO^[23]、离散社会蜘蛛算法^[24]、新型帝国竞争算法ICA^[10]和改进获益点的蜜蜂算法VPBA-II^[25])做了对比实验,结果见表6(本文算法的求解数据均加粗表示).结果显示:对于eil 76和rat 99实例而言,只有DChOA找到了最优解;对于其他实例,DChOA求解精度也优于其他大多数算法.通过对比可以看出,DChOA求解TSP的效果优于现有大多数文献所提出的算法,说明离散黑猩猩优化算法求解旅行商问题是可行的.

4 结论

本文提出了一种基于近邻牵引算法求解TSP的离散黑猩猩优化算法,拓展了黑猩猩优化算法在离散组合优化问题上的应用.通过设计优质片段检索方法,并引入组内学习策略来实现算法离散化,增强算

表6 不同算法求解结果对比

实例(已知最优解)	算法	所得最优解	实例(已知最优解)	算法	所得最优解
eil 76 (538)	DSMO	558.68	kroA 200 (29 368)	DSMO	30 481.35
	DSSA	559.31		DSSA	29 666
	ICA	544.37		ICA	29 430.88
	VPBA-II	552		VPBA-II	31 473
	DChOA	538		DChOA	29 565
rat 99 (1 211)	DSMO	1 225.56	lin 318 (42 029)	DSMO	45 460.25
	DSSA	1 221		DSSA	42 686
	ICA	1 219.24		ICA	42 307.53
	VPBA-II	1 243		VPBA-II	44 980
	DChOA	1 211		DChOA	43 913
kroA 150 (26 524)	DSMO	27 591.44	pr 439 (107 217)	DSMO	112 105.2
	DSSA	27 027		DSSA	111 450
	ICA	—		ICA	107 917.10
	VPBA-II	27 336		VPBA-II	—
	DChOA	27 231		DChOA	111 262

法扰动效果;通过设计组间交流策略和近邻牵引算子,保证算法快速收敛的同时有效提高了全局搜索能力,并通过设计自适应概率扰动调控策略平衡了算法的全局与局部搜索.实验结果表明:近邻牵引算子比传统搜索算子扰动更高效;DChOA在求解质量和收敛速度方面都明显较优;将黑猩猩优化算法应用于求解TSP是可行且有效的.下一步的研究方向是将其应用于更多组合优化问题.

参考文献(References)

- [1] Arora S. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems[J]. *Journal of the ACM: JACM*, 1998, 45(5): 753-782.
- [2] Volgenant T, Jonker R. A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation[J]. *European Journal of Operational Research*, 1982, 9(1): 83-89.
- [3] Ergun Ö, Orlin J B. A dynamic programming methodology in very large scale neighborhood search applied to the traveling salesman problem[J]. *Discrete Optimization*, 2006, 3(1): 78-85.
- [4] İlhan İ, Gökmen G. A list-based simulated annealing algorithm with crossover operator for the traveling salesman problem[J]. *Neural Computing and Applications*, 2022, 34(10): 7627-7652.
- [5] Dahan F, El Hindi K, Mathkour H, et al. Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem[J]. *Sensors*, 2019, 19(8): 1837.
- [6] Zhang P, Wang J, Tian Z, et al. A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem[J]. *Applied Soft Computing*, 2022, 127: 109339.
- [7] Shi X H, Liang Y C, Lee H P, et al. Particle swarm optimization-based algorithms for TSP and generalized TSP[J]. *Information Processing Letters*, 2007, 103(5): 169-176.
- [8] Panwar K, Deep K. Discrete grey wolf optimizer for symmetric travelling salesman problem[J]. *Applied Soft Computing*, 2021, 105: 107298.
- [9] Saji Y, Riffi M E. A novel discrete bat algorithm for solving the travelling salesman problem[J]. *Neural Computing and Applications*, 2016, 27(7): 1853-1866.
- [10] Zhang X L, Chen X W, Xiao H, et al. A new imperialist competitive algorithm for solving TSP problem[J]. *Control and Decision*, 2016, 31(4): 586-592.
- [11] Zhang Z C, Han W, Mao B. Adaptive discrete cuckoo algorithm based on simulated annealing for solving TSP[J]. *Acta Electronica Sinica*, 2018, 46(8): 1849-1857.
- [12] Deng W, Chen R, He B, et al. A novel two-stage hybrid swarm intelligence optimization algorithm and application[J]. *Soft Computing*, 2012, 16(10): 1707-1722.
- [13] Zhang Z, Han Y. Discrete sparrow search algorithm for symmetric traveling salesman problem[J]. *Applied Soft Computing*, 2022, 118: 108469.
- [14] Panwar K, Deep K. Transformation operators based grey wolf optimizer for travelling salesman problem[J]. *Journal of Computational Science*, 2021, 55: 101454.
- [15] Du Z G, Pan J S, Chu S C, et al. Multi-group discrete symbiotic organisms search applied in traveling salesman problems[J]. *Soft Computing*, 2022, 26(9): 4363-4373.
- [16] 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.
- [17] Khishe M, Mosavi M R. Chimp optimization algorithm[J]. *Expert Systems with Applications*, 2020, 149: 113338.
- [18] Larrañaga P, Kuijpers C M H, Murga R H, et al. Genetic algorithms for the travelling salesman problem: A review of representations and operators[J]. *Artificial Intelligence Review*, 1999, 13(2): 129-170.
- [19] Wang Y P, Li Y H. A novel quantum genetic algorithm for TSP[J]. *Chinese Journal of Computers*, 2007, 30(5): 5748-5755.
- [20] Hussain A, Muhammad Y S, Nauman Sajid M, et al. Genetic algorithm for traveling salesman problem with modified cycle crossover operator[J]. *Computational Intelligence and Neuroscience*, 2017, 2017: 7430125.
- [21] Qiao S, Lv Z M, Zhang N. Improved particle swarm optimization algorithm based on Hamming distance for traveling salesman problem[J]. *Journal of Computer Applications*, 2017, 37(10): 2767-2772.
- [22] Chen K S, Xian S D, Guo P. Adaptive temperature rising simulated annealing algorithm for traveling salesman problem[J]. *Control Theory & Applications*, 2021, 38(2): 245-254.
- [23] Akhand M A H, Ayon S I, Shahriyar S A, et al. Discrete spider monkey optimization for travelling salesman problem[J]. *Applied Soft Computing*, 2020, 86: 105887.
- [24] Baş E, Ülker E. Discrete social spider algorithm for the traveling salesman problem[J]. *Artificial Intelligence Review*, 2021, 54(2): 1063-1085.
- [25] Sultan Z, Harun I A, Natalia H, et al. An improved vantage point bees algorithm to solve combinatorial optimization problems from TSPLIB[J]. *Macromolecular Symposia*, 2021, 396(1): 2000299.

作者简介

沈孝凯(1997—),男,硕士生,从事智能优化算法的研究, E-mail: shenxiaokai321@126.com;

张纪会(1969—),男,教授,博士生导师,从事复杂系统建模、智能优化算法、物流系统工程等研究, E-mail: zhangjihui@qdu.edu.cn;

郭乙运(1979—),男,工程师,硕士,从事智慧物流、企业安全与信息化等研究, E-mail: guoyy.ywb@qdport.com;

张保华(1969—),男,工程师,硕士,从事智慧物流、智能系统等研究, E-mail: zbh@qdport.com.