

基于深度搜索的改进角蜥蜴优化算法求解 TSP 问题

魏杰¹, 习毅聪², 赵璇¹

(1 华中科技大学 管理学院, 武汉 430074; 2 国家电网有限公司 国网北京市电力公司物资分公司, 北京 100054)

摘要:针对角蜥蜴优化算法(horned lizard optimization algorithm, HLOA)无法求解旅行商问题(traveling salesman problem, TSP), 并且改进的连续型元启发式算法求解 TSP 问题存在寻优能力不足等问题, 提出一种基于深度搜索的改进角蜥蜴优化算法(improved horned lizard optimization algorithm with depth search, DSIHLOA). 该算法通过双重编码方式将 HLOA 离散化; 通过提出的信息共享策略提高算法前期的收敛速度; 设计深度搜索过程提高算法的搜索能力, 并提出 Greedy-Insert、Greedy-Swap 和 2-Opt-2 算子应用到深度搜索过程中. 通过多个 TSP 标准算例对 DSIHLOA 的性能进行测试, 并与传统元启发式算法对比. 结果表明: DSIHLOA 拥有较好的求解精度和收敛速度; 与其他文献的改进连续型元启发式算法进行对比, DSIHLOA 具有较好的寻优能力和稳定性.

关键词:旅行商问题; 角蜥蜴优化算法; 信息共享策略; 深度搜索

中图分类号: TP18

文献标志码: A

文章编号: 1000-2367(2026)01-0083-09

旅行商问题(traveling salesman problem, TSP)是组合优化中的一个 NP-hard 问题^[1], 其目标是寻找一条最优路径使旅行商能够以最小的总距离或总成本访问所有城市, 被广泛应用于物流配送、电子通信、项目调度、城市规划等领域, 具有重要的应用和研究价值^[2].

目前求解 TSP 问题的常用算法主要包括精确算法和(元)启发式算法. 以分支定界法、动态规划法为代表的精确算法求解问题的时间复杂度随问题规模的增加呈指数级增长, 只适用于求解小规模算例^[3]. 元启发式算法作为传统人工智能算法, 通常具有较低的时间复杂度, 并能够在合理的时间范围内找到接近最优解的求解方案, 更适用于解决大规模现实问题, 尤其是遗传算法(genetic algorithm, GA)、模拟退火算法(simulated annealing algorithm, SA)、蚁群算法(ant colony optimization, ACO)等元启发式算法^[4]. 为实现高精度、快速求解 TSP 问题的目标, 许多学者对 TSP 问题的求解算法进行了深入研究. 黄涛等^[5]对 GA 进行改进, 用复制算子替换选择算子, 并利用迭代次数和个体适应度自适应调节交叉和变异概率, 提出了一种改进自适应遗传算法, 通过 TSP 标准算例的实验证明了提出的改进算法有较高的收敛精度和收敛速度; 李泓波等^[6]将差分算法嵌入到 ACO 中, 并改进了交叉和变异操作, 通过在中小规模的 TSP 问题标准算例进行仿真实验, 证明了算法的有效性.

随着问题种类不断增加, 问题规模不断扩大, 越来越多的元启发式算法被设计出来, 以寻求更快的速度寻找到最优解, 如灰狼优化算法(wolf optimization algorithm, GWO)^[7]、鲸鱼优化算法(whale optimization algorithm, WOA)^[8]、野马优化算法(wild horse optimizer, WHO)^[9]等, 但这些算法在提出时只适用于求解

收稿日期: 2024-12-12; **修回日期:** 2025-01-16.

基金项目: 国家电网有限公司科技项目(5108-202218280A-2-187-XG).

作者简介: 魏杰(2000—), 男, 河北唐山人, 华中科技大学博士研究生, 研究方向为物流与供应链管理、车辆路径问题、元启发式算法, E-mail: jiew@hust.edu.cn.

通信作者: 赵璇(1995—), 女, 山东潍坊人, 华中科技大学博士研究生, 研究方向为物流与供应链管理、随机服务系统, E-mail: zhao1995@hust.edu.cn.

引用本文: 魏杰, 习毅聪, 赵璇. 基于深度搜索的改进角蜥蜴优化算法求解 TSP 问题[J]. 河南师范大学学报(自然科学版), 2026, 54(1): 83-91. (Wei Jie, Xi Yicong, Zhao Xuan. Improved horned lizard optimization algorithm with depth search for solving traveling salesman problem[J]. Journal of Henan Normal University(Natural Science Edition), 2026, 54(1): 83-91. DOI: 10.16366/j.cnki.1000-2367.2024.12.12.0004.)

连续型问题.近年来许多研究人员针对 TSP 问题的特点在这些算法的基础上进行改进并取得一些成果.杨进等^[10]提出改进的猫群算法(cat swarm optimization,CSO),引入交换子和交换序的概念,对基本猫群算法进行改进以求解 TSP 问题.ZHANG 等^[11]提出改进的变邻域离散鲸鱼优化算法(discrete whale optimization algorithm with variable neighborhood search,VDWOA),通过 TSP 标准算例的实验证明了算法的优化性能和全局搜索能力有所提高.刘海龙等^[12]结合 GA 和 GWO,提出改进的融合遗传灰狼优化算法(GA-IGWO),在筛选全局优秀个体后引入距离启发因子,更全面地探索解空间,避免陷入局部最优解,与其他改进 GWO 算法相比,在求解 TSP 问题的性能上有了大幅提高.虽然已有相关研究对连续型元启发式算法进行改进以求解 TSP 问题,但求解能力不足、寻优能力不强,以及一些改进算法只适用于小规模算例.因此,如何改进现有连续型元启发式算法能够更高效更快速地求解 TSP 问题仍是亟待研究的热点问题.

PERAZA-VÁZQUEZ 等^[13]在 2024 年基于角蜥蜴如何隐藏和保护自己不受捕食者伤害的防御机制提出了一种新型连续型元启发式算法——角蜥蜴优化算法(HLOA),该算法在寻找全局最优解的策略上表现出良好的性能,同时提高了求解速度和优化能力,兼顾了全局优化和避免局部最优的特点,但目前只适用于求解连续型问题,不适合求解 TSP 等离散优化问题.鉴于 HLOA 最新被提出,对其研究尚处于起步阶段,国内外尚无对 HLOA 求解 TSP 问题的研究,对 HLOA 的实际应用潜力和效能探索还存在研究空白,亟待学者挖掘与探究.为提高对现有改进连续型元启发式算法求解 TSP 问题的性能,同时拓展 HLOA 的应用范围,本文首次提出基于深度搜索的改进角蜥蜴优化算法(Improved horned lizard optimization algorithm with depth search,DSIHLOA),以求解 TSP 等离散优化问题.

1 TSP 及其数学模型

TSP 一般可描述为:现有一个旅行商需要访问 n 个城市,城市集合为 N ,随机从某一个城市出发,依次访问每一个城市且只能访问一次,最终返回起始城市,为该旅行商设计访问路线,使得旅行商的总行驶路径最短.给定城市 i 到城市 j 之间的距离为 $d(i, j)$,在对称 TSP 中, $d(i, j) = d(j, i)$.决策变量 $x_{ij} = 1$ 时,旅行商先访问城市 i 再访问城市 j ,否则 $x_{ij} = 0$.用数学模型表示 TSP 的目标函数如式(1)所示:

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}. \quad (1)$$

TSP 的约束条件为:

$$\sum_{i=1}^n x_{ij} = 1, \forall j \in N, \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \forall i \in N, \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset N, 2 \leq |S| \leq n - 1, \quad (4)$$

其中,约束(2)和(3)表示每个城市只能访问一次,约束(4)为消除子回路.

2 角蜥蜴优化算法

HLOA 是受美国中南部到墨西哥东北部的爬行动物角蜥蜴的生活习性启发而设计的元启发式算法^[14].该算法通过将角蜥蜴的多种防御机制抽象为数学模型,构建了包含两种被动与一种主动防御策略的优化算法,其中被动防御方法包括:①隐藏,角蜥蜴通过改变自身皮肤的颜色、图案、形状等,与环境融为一体,混淆敌人视线^[15];②移动,也称为逃跑策略,以远离敌人.主动防御方法也被称为侵略性策略,角蜥蜴会从眼睛里喷出一股短血,驱赶敌人^[16].同时,角蜥蜴皮肤颜色变化的快慢与 α -黑色素细胞刺激激素(α -melanophore stimulating hormone, α -MSH)水平有关^[17].因此,角蜥蜴优化算法将以上防御策略总结成算法的 5 个策略,分别为隐藏行为、皮肤暗化或亮化、喷血、逃跑、 α -MSH 比率.以下将对每个策略进行详细介绍.

2.1 策略一:隐藏

角蜥蜴通过改变自身皮肤的颜色、图案和形状来躲避敌人的追捕,提高自身生存能力,这是一种适应性

行为,角蜥蜴颜色变化理论已实现由数学形式来表示.国际照明委员会已对颜色评价系统进行了定义^[18],通过笛卡尔坐标系统和极坐标系统计算颜色空间中的颜色,对计算公式进行抽象,将角蜥蜴隐藏行为的变化抽象成可行解的更新公式^[16],如等式(5)所示.

$$X_i(t+1) = X_{\text{best}}(t) + (\partial - \frac{\partial t}{\text{Max } G}) [c_1(\sin(X_{r_1}(t)) - \cos(X_{r_2}(t))) - (-1)^\sigma c_2(\cos(X_{r_3}(t)) - \sin(X_{r_4}(t)))], \quad (5)$$

其中 $X_i(t+1)$ 表示第 $t+1$ 代的第 i 个个体; $X_{\text{best}}(t)$ 为第 t 代最优个体; r_1, r_2, r_3, r_4 是种群中随机选择的互不相同的个体编号; $X_{r_1}(t), X_{r_2}(t), X_{r_3}(t), X_{r_4}(t)$ 是第 t 代的 r_1, r_2, r_3, r_4 对应的个体代表的可行解; $\text{Max } G$ 是算法的最大迭代次数; σ 是 0-1 变量,取值规则为每次使用时在 $[0, 1]$ 范围内生成一个随机数,若大于 0.5,则 $\sigma = 1$,否则 $\sigma = 0$; c_1 和 c_2 是标准化调色盘里随机选择的数值,且 $c_1 \neq c_2$; ∂ 为一个定值.

2.2 策略二:皮肤暗化或亮化

角蜥蜴的皮肤可以根据外界太阳能的吸收量改变自身的颜色,式(6)表示角蜥蜴皮肤颜色变浅策略,式(7)表示角蜥蜴皮肤颜色变深策略.

$$X_{\text{worst}}(t) = X_{\text{best}}(t) + \frac{1}{2} \cdot L_1 \cdot \sin(X_{r_1}(t) - X_{r_2}(t)) - (-1)^\sigma \frac{1}{2} \cdot L_2 \cdot \sin(X_{r_3}(t) - X_{r_4}(t)), \quad (6)$$

$$X_{\text{worst}}(t) = X_{\text{best}}(t) + \frac{1}{2} \cdot D_1 \cdot \sin(X_{r_1}(t) - X_{r_2}(t)) - (-1)^\sigma \frac{1}{2} \cdot D_2 \cdot \sin(X_{r_3}(t) - X_{r_4}(t)). \quad (7)$$

其中 $X_{\text{worst}}(t)$ 是第 t 代最差的个体, L_1 和 L_2 是 $[\text{Lightening}_1, \text{Lightening}_2]$ 范围的随机数, D_1 和 D_2 是 $[\text{Darkening}_1, \text{Darkening}_2]$ 范围的随机数,具体调色范围数值参考 PERAZA-VÁZQUEZ 等^[13]的研究.

2.3 策略三:喷血

角蜥蜴可通过从眼睛中喷射血液抵挡敌人,该过程可用抛射运动表示,将血液运动分解成垂直运动和水平运动,将运动方程拟化为解向量更新公式如式(8)所示.

$$X_i(t+1) = [v_0 \cos(\alpha - \frac{1}{\text{Max } G}) + \epsilon] X_{\text{best}}(t) + [v_0 \sin(\alpha - \frac{1}{\text{Max } G}) - g + \epsilon] X_i(t), \quad (8)$$

其中 v_0 为血液喷射的初速度; α 为血液喷射运动方向与水平方向的夹角,在算法中取为定值; ϵ 为一个定值参数; g 表示重力加速度.

2.4 策略四:逃跑

角蜥蜴会以一定概率逃离敌人的追捕,基于该种防御机制,产生了一个包含局部和全局运动函数以更新可行解,如式(9)所示.

$$X_i(t+1) = X_{\text{best}}(t) + W_a (\frac{1}{2} - \zeta) X_i(t), \quad (9)$$

其中, W_a 为一个随机数且 $W_a \in [-1, 1]$, $\zeta \in N(0, 1)$.

2.5 策略五: α -MSH 比率

角蜥蜴颜色变化的快慢由自身的 α -MSH 激素水平决定,算法中定义每个个体的 α -MSH 激素水平为:

$$m(i) = \frac{F_{\text{max}} - F(i)}{F_{\text{max}} - F_{\text{min}}}, \quad (10)$$

其中, F_{max} 为当前种群中适应度最高的个体, F_{min} 为当前种群中适应度最低的个体, $F(i)$ 为第 i 个个体的适应度, $m(i) \in [0, 1]$.当 $m(i) < 0.3$ 时,对第 i 个个体进行更新,更新公式为:

$$X_i(t) = X_{\text{best}}(t) + \frac{1}{2} [X_{r_1}(t) - (-1)^\sigma X_{r_2}(t)]. \quad (11)$$

3 基于深度搜索的改进角蜥蜴优化算法

3.1 解编码策略

本文采用自然数和实数的双重编码方式.第一重为自然数编码,表示旅行商访问城市的顺序;第二重为

实数编码,并实现实数编码向自然数编码的转换.ZHANG 等^[19]曾采用将连续值转换成索引值的方法,对连续值向量的每一维按升序排列获得其对应的索引序号,实现离散化编码,但该方法并没有限制范围,容易降低搜索能力.本文对该种方法进行改进,限制连续值的变化范围,进而提高搜索能力.假设个体 i 的第一重编码为 $X_{i1} = (1, 2, 3, 4, 5)$,第二重编码为 $X_{i2} = (1.1, 1.5, 1.9, 3.7, 4.8)$,个体 j 的第一重编码为 $X_{j1} = (2, 5, 4, 1, 3)$,第二重编码为 $X_{j2} = (2.3, 4.5, 3.4, 1.3, 2.9)$,解更新过程为每个个体的第二重编码进行运算得到个体 k 的第二重编码 $X_{k2} = X_{j2} - X_{i2} = (1.2, 3.0, 1.5, -2.4, -1.9)$,运算后对解进行修复使得每一维的值始终在 $[1, n]$ 内(n 为城市数量),修复方法为将超出边界值的值更改为其所靠近的边界值,即修复后的个体 k 的第二重编码为 $X_{k2} = (1.2, 3.0, 1.5, 1.1)$,对修复后的结果重新排序,得到个体 k 的第一重编码 $X_{k1} = (3, 5, 4, 1, 2)$,即个体 k 代表的可行解表示的旅行商访问城市顺序为 $3 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 2$.

3.2 初始解集生成

本文采用贪婪算法生成和随机生成两种方式生成初始解集,假设种群规模为 N_{pop} ,城市数量为 n .① 通过贪婪算法生成一个可行解;② $N_{\text{pop}} - 1$ 个可行解的生成方式:对每个个体 i 随机生成 $X_{i2} = (x_1, x_2, \dots, x_n)$ 的第二重编码向量,每个元素的取值范围为 $[1, n]$,通过解修复方式生成个体 i 的第一重编码向量,完成初始解集生成.

3.3 信息共享策略

元启发式算法普遍存在收敛速度较慢的缺点,为提高算法的收敛速度,本文提出了信息共享策略.对于每个种群中的每个个体 i ,在完成 HLOA 的每个策略后,以一定的概率 P_I 全局最优解的信息片段.在算法初期, P_I 的值应偏高,促进种群个体获得全局最优解的优秀片段,继而起到提高收敛速度的作用;但在算法后期,过高的 P_I 值会使得算法陷入局部最优,个体无法继续进化.因此,需采取自适应性信息共享策略,自动调整信息共享概率,概率 P_I 的计算公式为:

$$P_I = \omega e^{-\theta t}, \quad (13)$$

其中, $\omega, \theta \in (0, 1)$ 为定值, t 为当前迭代次数.操作过程为:在全局最优解中随机选择两点 p_1 和 p_2 及其之间的城市序号,将个体 i 的对应位置以该片段替换,并对新的解向量进行修复,使个体 i 代表的解为可行解.

3.4 深度搜索

为使算法有更好的局部搜索能力,本文采用 2 种常用操作算子,分别为 Insert 和 Swap 算子,并设计 3 种深度搜索操作算子,分别为 Greedy-Insert、Greedy-Swap 和 2-Opt-2 算子,提升算法寻优性能.

(1) Insert 算子.以 8 个城市的路径为例,假设当前访问路线为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$,随机选择一个城市,如选择城市 4,随机插入到另一个城市之后,如选择插入到城市 6 之后,则操作后的访问路线为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 7 \rightarrow 8$.

(2) Swap 算子.以 8 个城市的路径为例,假设当前访问路线为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$,随机选择两个城市,如选择城市 4 和 6,交换两个城市的位置,操作后的访问路线为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 8$.

(3) Greedy-Insert 算子.基于贪婪算法思想对 Insert 算子进行改进.随机选择一个城市,将该城市插入到总行驶距离减少最多的位置.

(4) Greedy-Swap 算子.基于贪婪思想对 Swap 算子进行改进.随机选择一个城市,将该城市尝试与其他所有城市位置交换,选择总行驶距离减少最多的城市位置完成最终交换.

(5) 2-Opt-2 算子.本文对 2-Opt 算子进行改进,设计 2-Opt-2 算子:随机选择可行解中的一个城市,以该城市在可行解中的位置为起点,向其中一个方向随机选择连续的几个城市形成的片段进行翻转,翻转完成后,继续以该位置为起点,向另一个方向选择相同数量的连续城市形成的片段继续翻转,得到操作后的可行解.以 8 个城市的路径为例,假设当前访问路线为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$,随机选择一个城市,如选择城市 4(路径的第 4 个位置),向左随机选择连续的几个城市,如选择城市 2 和 3,则对 $2 \rightarrow 3 \rightarrow 4$ 路径进行翻转,得到路径 $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$,继续以路径的第 4 个位置为起点,向右选择等数量的连续城市(选择城市 5 和 6),则对路径 $2 \rightarrow 5 \rightarrow 6$ 进行翻转,得到最终变化后的路径为 $1 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 2 \rightarrow 7 \rightarrow 8$.

深度搜索过程应用于原始 HLOA 与信息共享策略过程之后,通过对 Insert、Swap、Greedy-Insert、Greedy-Swap 和 2-Opt-2 算子的循环使用以达到深度搜索的效果,提高算法寻优能力.具体深度搜索伪代码

如算法 1 所示.

算法 1 深度搜索

输入: x (角蜥蜴个体代表的可行解), β (深度搜索最大次数)

```

1:  $i \leftarrow 1, k \leftarrow 1$ 
2: While  $i \leq \beta$ 
3:   Switch  $k$ 
4:     Case 1
5:       If  $r > 0.5$  // 以 0.5 的概率实施 Insert 和 Greedy-Insert 算子
6:          $x' \leftarrow \text{Insert}(x)$  // 对  $x$  采用 Insert 算子
7:       Else
8:          $x' \leftarrow \text{GreedyInsert}(x)$ 
9:       End If
10:    Case 2
11:      If  $r > 0.5$  // 以 0.5 的概率实施 Swap 和 Greedy-Swap 算子
12:         $x' \leftarrow \text{Swap}(x)$ 
13:      Else
14:         $x' \leftarrow \text{GreedySwap}(x)$ 

```

```

15:    End If
16:    Case 3
17:       $x' \leftarrow 2\text{Opt2}(x)$ 
18:    End Switch
19:    If  $f(x') < f(x)$  // 如果所得的解有更好的适应度值,则替换原来的解
20:       $x \leftarrow x', i \leftarrow i + 1$ 
21:    Else
22:       $k \leftarrow k + 1$  // 若  $x$  没有更新则采用下一个算子再次进行搜索
23:    End If
24:    If  $k \geq 3$  // 若经过所有算子的操作,  $x$  都没有进化,进入下一次深度搜索
25:       $i \leftarrow i + 1, k \leftarrow 1$ 
26:    End If
27:  End While
28: Return  $x$ 

```

3.5 算法流程及复杂度计算

根据 HLOA 的算法原理和对 HLOA 的改进策略,DSIHLOA 流程图如图 1 所示. HLOA 的算法复杂度在文献[13]中已得到精确的计算,为 $O(I \times P \times D \times O(T))$,其中 I 为最大迭代次数, P 为种群规模, D 为问题维度, $O(T)$ 为目标函数的计算复杂度. 根据图 1 展示的 DSIHLOA 算法流程,相较于原始的 HLOA,解修复过程、信息共享策略和深度搜索过程被加入,在每个个体的每次迭代过程中,每个过程的算法复杂度分别为 $O(1)$ 、 $O(1)$ 和 $O(\beta \times O(L))$,其中 β 为深度搜索的最大次数, $O(L)$ 为复杂度最高的搜索算子在每次计算时的计算复杂度. 因此,DSIHLOA 的算法复杂度为 $O(I \times P \times D \times O(T) \times \beta \times O(L))$. 与 HLOA 相比,DSIHLOA 的计算量与计算时间会与 β 和 $O(L)$ 的值密切相关.

4 仿真实验与结果分析

为测试本文设计的 DSIHLOA 的有效性,本章基于 TSP 对其做仿真实验分析. 实验环境为 Intel(R) Core(TM) i5-8300H

CPU @2.3GHz,运行内存 16 GB,Windows10 操作系统以及 MATLAB 2019b 版本软件.选取 TSPLIB 数据集中的标准对称 TSP 算例进行实验.算法参数设置为: $d=2, v_0=1, \epsilon=10^{-6}, \alpha=\pi/2, g=0.009\ 807, \omega=0.9,$

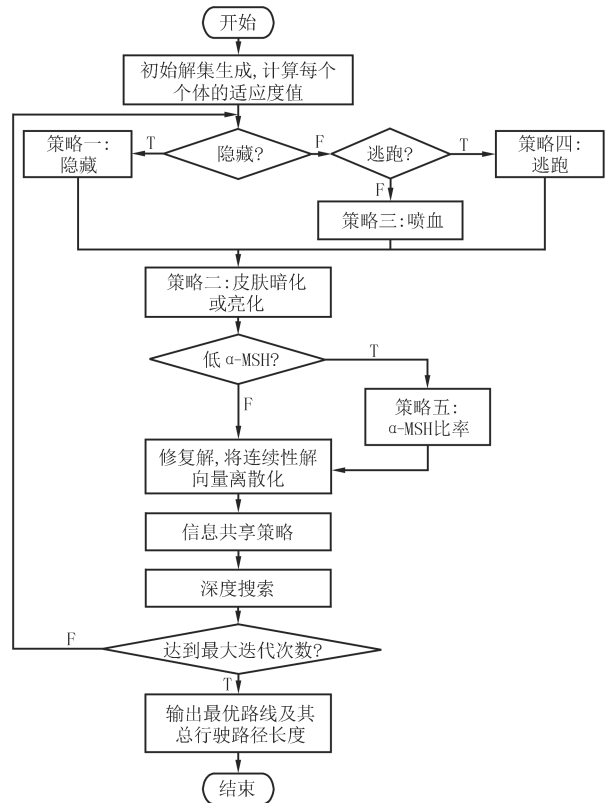


图1 DSIHLOA流程图

Fig. 1 DSIHLOA flowchart

$\theta=0.7,\beta=80.$

4.1 DSIHLOA 求解能力

下面选取 TSPLIB 数据集中的 9 个算例验证 DSIHLOA 求解能力,求解结果如表 1 所示.对每个算例进行 10 次独立实验,初始种群规模均设为 20,最大迭代次数为 300.

表 1 DSHLOA 算法求解 TSP 实验结果

Tab. 1 Experimental results of DSHLOA algorithm for solving TSP

算例	Opt	Best	Ave	BER	MER	算例	Opt	Best	Ave	BER	MER
dantzig42	699	688.3	692.1	-0.015 3	-0.009 9	pr76	108 159	110 656.0	112 783.4	0.023 1	0.042 8
att48	33 522	33 523.7	33 772.2	0.000 1	0.007 5	kroa100	21 282	23 511.9	24 235.5	0.104 8	0.138 8
eil51	426	429.5	433.5	0.008 3	0.017 6	tsp225	3 916	4 222.1	4 526.9	0.078 2	0.156 0
berlin52	7 542	7 544.4	7 645.6	0.000 3	0.013 7	a280	2 579	2 763.9	2 773.1	0.071 7	0.075 2
st70	675	699.7	713.2	0.036 6	0.056 6						

从表 1 可以看出,DSIHLOA 对于 dantzig42 算例求得的结果优于已知最优解(Opt),对其他算例求得的结果非常接近最优解,特别是对 att48 和 berlin52 算例的求解结果与最优解相差无几,其中最优解误差率(BER)最大为 0.104 8,BER 均值为 0.034 0,平均解误差率(MER)最大为 0.156 0,MER 均值为 0.055 4.从总体上看 DSIHLOA 对于不同规模城市都具有较强的全局搜索能力,无论对于 dantzig42、att48 等小规模算例,还是 tsp225、a280 等中大规模算例都能获得比较理想的结果.附录 S1 为求解 dantzig42、att48 和 berlin52 算例所获得的路径图.

4.2 DSIHLOA 与经典元启发式算法对比

为验证算法有效性,本节将 DSIHLOA 与 HLOA、GA、SA 和 PSO 进行对比.其中 HLOA 是对 HLOA 只进行离散化的算法,GA 和 SA 是经典的离散型元启发式算法,PSO 是经典的连续型元启发式算法且已完成离散化.选取 TSPLIB 数据集中的 13 个算例进行实验,每种算法对每个算例进行 10 次实验,初始种群规模均设为 20,最大迭代次数均为 300(SA 的内循环最大次数为 50),实验结果如附录表 S1 所示.

从表 S1 可以看到,DSIHLOA 在 13 个测试算例中所求得的最优解“Best”和平均值“Ave”均优于其他 4 个算法,说明 DSHLOA 的寻优能力要优于其他用于对比的经典元启发式算法;DSIHLOA 在 10 个算例中的标准差“Std”统计量优于其他算法,说明 DSIHLOA 的稳定性要优于其他算法.

4.3 DSIHLOA 与最新改进元启发式算法对比

为进一步验证 DSHILOA 的有效性,本节将它与其他文献中最新提出的求解 TSP 问题的改进连续型元启发式算法进行对比,包括改进融合遗传灰狼算法(GA-IGWO)、改进猫群算法(CSO)和改进鲸鱼优化算法(VDWOA),分析结果如表 2 所示.

在表 2 中,DSIHLOA 求得的最优解在 11 个算例中都优于其他算法求得的结果,说明本文算法有着较强的全局搜索能力;而 DSIHLOA 求得的平均值结果在 11 个算例中优于其他算法,有 8 个算例的标准差优于其他算法,验证了本文算法有较好的鲁棒性和稳定性.从总体上来说,DSIHLOA 的寻优能力和稳定性都具有优势.

本节也从算法的复杂度上对 DSIHLOA 与 GA-IGWO、CSO 和 VDWOA 进行比较分析,GA-IGWO、CSO 和 VDWOA 的算法复杂度分别为 $O(I \times P \times D \times O(T) \times O(L))$ 、 $O(I \times P \times D \times O(T)) + O(L)$ 和 $O(I \times P \times D \times O(T) \times O(L))$.从对比中可以看出,除了算法复杂度最低的 CSO 以外,DSIHLOA 与其他两种算法的算法复杂度相比,增加了 $O(\beta)$,这也就意味着,与其他算法相比, β 的大小将直接影响 DSIHLOA 在计算复杂度上的优劣程度.而由于 DSIHLOA 的种群质量较高,且深度搜索过程提高了算法的搜索能力,因此,DSIHLOA 可以以较小的迭代次数和较短的运行时间获得高质量的解.在 4.6 节中,本文对 DSIHLOA 的时间性能进行了更详细的实验分析.

4.4 消融实验

为验证本文提出的信息共享策略对 HLOA 算法的改进机制的有效性,本节进行消融实验.首先定义:DSIHLOA-I 为缺少信息共享策略的 DSIHLOA.选取 att48、kroa100 和 gr137 算例进行收敛性分析,最大迭代次数为 300,实验结果如图 2 所示.

表 2 DSHLOA 与其他文献算法实验结果对比

Tab. 2 Comparison results of DSHLOA and other algorithms

算例	统计量	DSIHLOA	GA-IGWO	CSO	VDWOA	算例	统计量	DSIHLOA	GA-IGWO	CSO	VDWOA
dantzig42	Best	688.3	—	689.0	688.3	kroa100	Best	23 511.9	—	—	123 749.0
	Ave	692.1	—	—	688.3		Ave	24 235.5	—	—	134 691.7
	Std	7.5	—	—	0.0		Std	376.6	—	—	4 871.7
att48	Best	33 523.7	33 444.0	—	92 224.0	gr137	Best	742.3	956.7	—	900.1
	Ave	33 772.2	33 956.9	—	105 251.4		Ave	753.7	962.5	—	970.1
	Std	122.9	217.1	—	4 998.6		Std	6.7	2.4	—	25.8
eil51	Best	429.5	—	447.0	1 018.5	kroa200	Best	37 854.4	—	—	283 713.0
	Ave	433.5	—	—	1 139.9		Ave	38 303.2	—	—	291565.1
	Std	1.8	—	—	48.8		Std	287.1	—	—	4 084.0
berlin52	Best	7 544.4	—	—	18 197.5	tsp225	Best	4 222.1	—	—	9 938.8
	Ave	7 645.6	—	—	19 431.8		Ave	4 526.9	—	—	10 025.5
	Std	61.9	—	—	697.4		Std	193.6	—	—	69.8
st70	Best	699.7	—	694.0	2 510.4	a280	Best	2 763.9	—	—	2 815.3
	Ave	713.2	—	—	2 706.1		Ave	2 773.1	—	—	2 818.3
	Std	6.4	—	—	98.5		Std	11.6	—	—	1.0
pr76	Best	110 656.0	117 823.3	—	149 212.9	lin318	Best	50 178.8	—	—	114 819.2
	Ave	112 783.4	120 624.0	—	150 455.4		Ave	52 195.2	—	—	118 513.7
	Std	1 430.3	1 304.6	—	530.2		Std	1 499.3	—	—	1 399.6
gr96	Best	540.9	698.0	—	639.8	rd400	Best	21 220.6	—	—	188 651.4
	Ave	548.0	711.9	—	705.2		Ave	21 553.8	—	—	192 709.0
	Std	4.4	6.9	—	37.4		Std	196.3	—	—	2 242.2

通过消融实验的收敛曲线可以看出,在 3 种规模算例的实验下,DSIHLOA 在拥有信息共享策略时,算法前期的收敛速度大幅提升,这也保证了为算法后期提供更优秀的种群个体,以提高算法的寻优能力。

4.5 DSIHLOA 的收敛性

为进一步分析 DSIHLOA 的收敛速度和寻优能力,本文选取 GA、SA、HLOA 和 VDWOA 4 种算法进行对比,分别选取小、中、大规模算例 att48、gr137 和 rd400 进行收敛性分析.各算法收敛曲线如图 3 所示。

根据图 3 可以看出,对于算例 att48 和 rd400,GA、HLOA 和 VDWOA 的求解精度较低,且收敛速度很慢,而 DSIHLOA 在迭代开始时就开始收敛,SA 的收敛速度较快,但与 DSIHLOA 相比仍较慢;对于算例 gr137,HLOA 的求解能力较高于 SA,且与 VDWOA 相当,但 DSIHLOA 的求解能力仍有明显的优势.从收敛图可以看出,在小、中、大规模算例中,DSIHLOA 均可在 20 代以内收敛到较好的结果.由于设计了初始解的生成策略,可以在算法初期以一个较好的解开始进行搜索过程,从而提高算法的全局搜索能力。

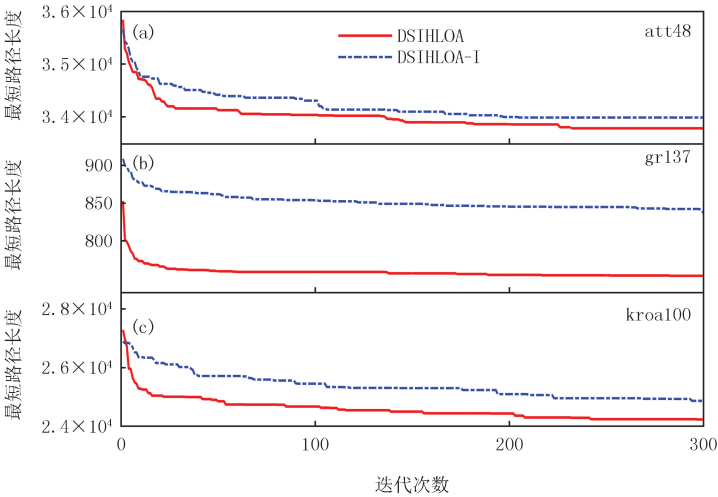


图2 消融实验收敛曲线

Fig.2 Convergence curves of ablation experiment

4.6 DSIHLOA 的时间性能分析

为进一步分析 DSIHLOA 与其他改进算法在时间上的性能,本节通过 pr76 和 gr137 算例从时间和搜索能力方面,综合评估算法性能.GA-IGWO 是在求解精度和求解时间上综合考量下较为优秀的算法,因此,本文以 DSIHLOA 和 GA-IGWO 对比为例.对于 pr76 算例,DSIHLOA 用时 6.67 s 找到的解的路径长度为 117 586.0,而 GA-IGWO 用时 7.00 s 找到的解的路径长度为 117 823.3;而对于 gr137,DSIHLOA 用时 5.02 s 找到的解的路径长度为 804.426,而 GA-IGWO 用时 13.55 s 找到的解的路径长度为 956.7,可以认为 DSIHLOA 的求解性能较好,深度搜索过程提高了算法的搜索能力,使得 DSIHLOA 能够在更短时间内搜索到更好的解.

5 结 论

本文对 HLOA 算法进行改进,设计了 DSIHLOA 并在 TSP 问题上进行了求解验证.设计了双重编码方式将 HLOA 离散化;设计了一种信息共享策略提高算法的收敛速度;设计了深度搜索过程,通过采用 Insert、Swap 算子并设计了 Greedy-Insert、Greedy-Swap 和 2-Opt-2 算子,提高算法的搜索能力.选取多个 TSP 标准算例进行实验,与只进行离散化的 HLOA,传统的 GA、SA、PSO 和其他文献算法 GA-IGWO、CSO、VDWOA 进行对比,验证提出的 DSIHLOA 的有效性;通过 DSIHLOA 的收敛性和时间性能综合分析,表明了 DSIHLOA 在求解 TSP 问题上有较好的性能.

目前对 HLOA 的研究刚刚起步,未来研究将进一步对 DSIHLOA 进行改进,尝试使用其他搜索算子并设计新的搜索算子,提出搜索算子的自适应选择策略,以提高算法的求解速度和在大规模算例上的寻优能力;通过设计提高种群多样性的策略,提高算法跳出局部最优的能力.此外,研究将该算法用于其他离散型问题上,以拓展 HLOA 的应用领域.

附录见电子版(DOI:10.16366/j.cnki.1000-2367.2024.12.12.0004).

参 考 文 献

[1] 郭洪升,李忠伟,罗偲,等.基于混合人工蜂群算法和 A* 算法的求解旅行商问题算法[J].科学技术与工程,2023,23(11):4718-4724.
GUO H S,LI Z W,LUO C,et al.Algorithm for solving travel salesman problems based on hybrid artificial bee colony algorithm and A-star algorithm[J].Science Technology and Engineering,2023,23(11):4718-4724.

[2] 但开,段隆振.基于最优插入子集的动态规划算法求解旅行商问题[J].计算机应用与软件,2022,39(12):260-265.
DAN K,DUAN L Z.Dynamic programming algorithm based on optimal insertion subset for travelling salesman problem[J].Computer Applications and Software,2022,39(12):260-265.

[3] 陈加俊,谭代伦.求解旅行商问题的探索—开发—跳跃策略单亲遗传算法[J].计算机应用研究,2023(5):1375-1380.
CHEN J J,TAN D L.Partheno-genetic algorithm based on explore-develop-jump strategy for solving traveling salesman problem[J].Ap-plication Research of Computers,2023(5):1375-1380.

[4] 袁培燕,刘萍,高宏卿.蚁群算法迭代次数的一种优化策略[J].河南师范大学学报(自然科学版),2010,38(4):48-50.
YUAN P Y,LIU P,GAO H Q.An optimized strategy for iteration numbers in ant colony algorithm[J].Journal of Henan Normal Univer-sity(Natural Science),2010,38(4):48-50.

[5] 黄涛,邓斌,何栋,等.一种改进的自适应遗传算法[J].计算机仿真,2024,41(3):347-351.
HUANG T,DENG B,HE D,et al.Improved adaptive genetic algorithm for solving multi-parameter problems[J].Computer Simulation,2024,41(3):347-351.

[6] 李泓波,张晓霞.基于 DE 算法的改进 ACO 算法求解旅行商问题[J].辽宁科技大学学报,2023,46(2):127-132.

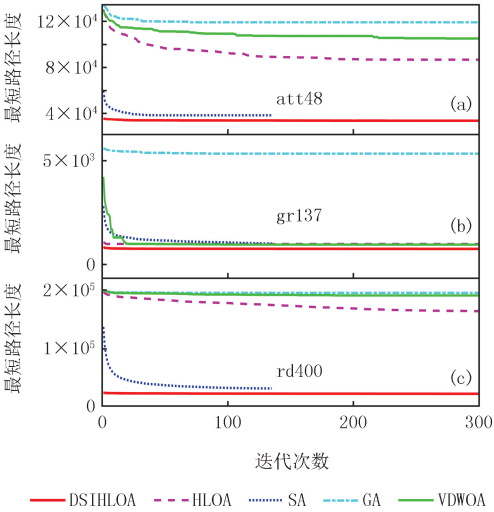


图3 各算法收敛曲线
Fig.3 Convergence curves of each algorithm

LI H B,ZHANG X X.An improved ACO algorithm based on DE for solving traveling salesman problem[J].Journal of University of Science and Technology Liaoning,2023,46(2):127-132.

[7] MIRJALILI S,MIRJALILI S M,LEWIS A.Grey wolf optimizer[J].Advances in Engineering Software,2014,69:46-61.

[8] 夏正龙,陈宇,陆良帅,等.基于多目标鲸鱼算法的配电网动态无功优化研究[J].河南师范大学学报(自然科学版),2025,53(1):116-126.

XIA Z L,CHEN Y,LU L S,et al.Research on dynamic reactive power optimization of distribution network based on multi-objective whale optimization algorithm[J].Journal of Henan Normal University(Natural Science Edition),2025,53(1):116-126.

[9] NARUEI I,KEYNIA F.Wild horse optimizer:A new meta-heuristic algorithm for solving engineering optimization problems[J].Engineering with Computers,2022,38(4):3025-3056.

[10] 杨进,郑允,马良.改进的猫群算法求解 TSP[J].计算机应用研究,2017,34(12):3607-3610.

YANG J,ZHENG Y,MA L.Improved cat swarm optimization for solving traveling salesman problem[J].Application Research of Computers,2017,34(12):3607-3610.

[11] ZHANG J,HONG L,LIU Q.An improved whale optimization algorithm for the traveling salesman problem[J].Symmetry,2021,13(1):48.

[12] 刘海龙,雷斌,王菀莹,等.求解 TSP 问题的改进融合遗传灰狼优化算法[J].计算机仿真,2023,40(9):333-338.

LIU H L,LEI B,WANG W Y,et al.Improved fused genetic grey wolf optimization algorithm for solving TSP[J].Computer Simulation,2023,40(9):333-338.

[13] PERAZA-VÁZQUEZ H,PENÁ-DELGADO A,MERINO-TREVIÑO M,et al.A novel metaheuristic inspired by horned lizard defense tactics[J].Artificial Intelligence Review,2024,57(3):59.

[14] LEACHÉ A D,MCGUIRE J A.Phylogenetic relationships of horned lizards(Phrynosoma)based on nuclear and mitochondrial data:Evidence for a misleading mitochondrial gene tree[J].Molecular Phylogenetics and Evolution,2006,39(3):628-644.

[15] STEVENS M,MERILAITA S.Animal Camouflage:Mechanisms and Function[M].Cambridge:Cambridge University Press,2011.

[16] COOPER W E Jr,SHERBROOKE W C.Plesiomorphic escape decisions in cryptic horned lizards(Phrynosoma)having highly derived anti-predatory defenses[J].Ethology,2010,116(10):920-928.

[17] SHERBROOKE W C.Physiological(rapid)change of color in horned lizards[J].Amphibia-Reptilia,1997,18(2):155-175.

[18] SCHRÖDINGER E.Erwin Schrödinger's Color Theory:Translated with Modern Commentary[M].Berlin:Springer,2017.

[19] ZHANG Z,HAN Y.Discrete sparrow search algorithm for symmetric traveling salesman problem[J].Applied Soft Computing,2022,118:108469.

Improved horned lizard optimization algorithm with depth search for solving traveling salesman problem

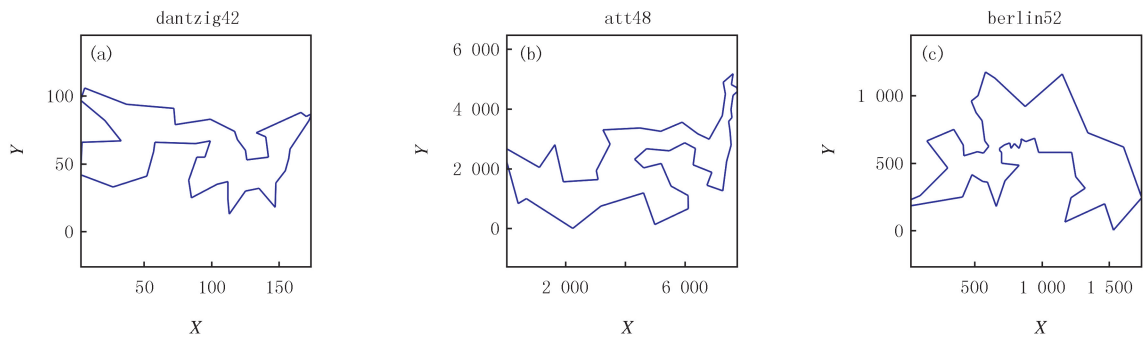
Wei Jie¹, Xi Yicong², Zhao Xuan¹

(1. School of Management, Huazhong University of Science and Technology, Wuhan 430074, China;
2. State Grid Beijing Procurement Company, State Grid Corporation of China, Beijing 100054, China)

Abstract: Aiming at the problems of the horned lizard optimization algorithm(HLOA)cannot solve the discrete problems, for example, the traveling salesman problem(TSP), and the improved continuous metaheuristic algorithm has insufficient optimization capability for solving the TSP, this paper proposed an improved horned lizard optimization algorithm with depth search(DSIHLOA). The algorithm discretizes the HLOA utilizing double coding. The proposed information sharing strategy improves the convergence speed of the algorithm at the early stage. The depth search process improves the search ability of the algorithm. The proposed Greedy-Insert, Greedy-Swap, and 2-Opt-2 operators are applied to the depth search process. Through simulation experiments on several TSP instances and comparing them with traditional metaheuristic algorithms, the results show that DSIHLOA possesses better solution accuracy and convergence speed. Compared with other improved continuous metaheuristic algorithms in the literature, the results show that DSIHLOA possesses better optimization search ability and stability.

Keywords: traveling salesman problem; horned lizard optimization algorithm; information sharing strategy; depth search

附 录



图S1 DSIHLOA求解3个算例路径图
Fig.S1 DSIHLOA solves path diagrams for three examples

表 S1 DSIHLOA 与经典元启发式算法实验结果对比
Tab. S1 Comparison results of DSIHLOA and classical metaheuristic algorithms

算例	统计量	DSIHLOA	HLOA	GA	SA	PSO
dantzig42	Best	688.3	688.3	2 091.5	688.3	1 184.9
	Ave	692.1	714.9	2 216.6	714.1	1 335.4
	Std	7.5	79.6	79.9	18.1	92.5
att48	Best	33 523.7	64 767.4	105 305.8	36 059.7	78 528.2
	Ave	33 772.2	86 719.6	119 134.7	38 485.7	82 990.6
	Std	122.9	16 508.5	8 092.7	2 465.2	2 871.5
eil51	Best	429.5	1 271.8	1 324.2	440.8	975.9
	Ave	433.5	1 343.0	1 386.2	454.6	1 069.3
	Std	1.8	68.6	40.6	10.4	53.2
berlin52	Best	7 544.4	20 751.4	22 870.2	8 139.7	13 781.3
	Ave	7 645.6	22 259.5	24 181.1	8 793.5	15 986.6
	Std	61.9	997.9	1 114.1	374.5	911.9
st70	Best	699.7	2 975.8	2 853.4	737.7	2 200.6
	Ave	713.2	3 165.0	3 106.9	771.5	2 333.7
	Std	6.4	128.1	114.5	27.5	87.6
pr76	Best	110 656.0	150 779.9	470 438.5	134 098.0	209 778.9
	Ave	112 783.4	150 779.9	487 827.0	150 031.0	241 823.3
	Std	1 430.3	0.0	9 263.4	12 390.5	20 042.4
gr96	Best	540.9	742.2	2 779.2	586.1	1 625.7
	Ave	548.0	779.4	2 922.5	641.5	2 026.9
	Std	4.4	88.3	61.7	38.6	225.0

续表						
算例	统计量	DSIHLOA	HLOA	GA	SA	PSO
kroa100	Best	23 511.9	92 873.5	136 969.0	26 812.7	103 789.9
	Ave	24 235.5	10 4051.9	143 414.7	30 040.2	109 401.1
	Std	376.6	6 189.2	3 912.5	1 748.0	2 892.7
gr137	Best	742.3	973.0	5 008.6	974.7	4 152.2
	Ave	753.7	990.9	5 344.4	1 007.9	4 743.2
	Std	6.7	6.3	165.0	29.5	318.1
kroa200	Best	37 854.4	218 725.3	301 026.8	46 795.8	233 154.7
	Ave	38 303.2	237 382.8	306 333.7	51 114.2	237 999.1
	Std	287.1	10 486.8	4 523.0	2 707.3	3 155.3
tsp225	Best	4 222.1	9 949.8	36 709.3	5 416.4	28 895.9
	Ave	4 526.9	10 536.1	38 060.4	5 770.5	30 192.1
	Std	193.6	893.7	728.0	256.6	906.5
a280	Best	2 763.9	2 818.6	30 289.5	4 437.8	29 250.2
	Ave	2 773.1	2 818.6	31 242.4	4 729.1	30 859.7
	Std	11.6	0.0	557.5	140.2	979.9
rd400	Best	21 220.6	15 7240.3	195 355.8	29 406.4	136 076.4
	Ave	21 553.8	166 099.7	198 026.8	30 965.0	138 952.6
	Std	196.3	6 621.3	1 560.9	1 126.1	1 667.2