

doi: 10.7690/bgzdh.2025.07.007

基于多策略融合的改进灰狼算法

文 竹¹, 韦杏琼², 刘静怡²

(1. 广西警察学院信息技术学院, 南宁 530028; 2. 广西民族大学人工智能学院, 南宁 530006)

摘要: 针对当前各类路径优化算法搜索规模较小、收敛速度较慢、全局搜索与局部搜索不平衡等问题, 提出一种多策略融合的改进灰狼优化算法(multi-strategy fusion of grey wolf optimization algorithm, MGWO)。通过引入精英反向优化策略对种群进行初始化, 提高初始解的质量。采用自适应权重机制, 动态调整最优狼的领导能力。通过分段搜索方法, 提升平衡局部搜索与全局探索的能力。仿真实验结果表明: 该算法表现出色, 能快速找到最优路径, 提高算法的整体性能, 具有一定借鉴作用。

关键词: 改进灰狼算法; 精英反向策略; 自适应权重; 分段策略; 路径优化

中图分类号: O221; TP183 文献标志码: A

Improved Grey Wolf Algorithm Based on Multi-strategy Fusion

Wen Zhu¹, Wei Xingqiong², Liu Jingyi²

(1. School of Information Technology, Guangxi Police College, Nanning 530028, China;

2. School of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China)

Abstract: In order to solve the problems of small search scale, slow convergence speed and imbalance between global search and local search in current path optimization algorithms, a multi-strategy fusion of grey wolf optimization algorithm (MGWO) is proposed. The quality of the initial solution is improved by introducing the elite reverse optimization strategy to initialize the population. An adaptive weight mechanism is used to dynamically adjust the leadership of the optimal wolf. The ability of balancing local search and global exploration is improved through the piecewise search method. The simulation results show that the algorithm performs well, can quickly find the optimal path, and improve the overall performance of the algorithm, which has a certain reference.

Keywords: improved grey wolf algorithm; elite reverse strategy; adaptive weight; segmentation strategy; path optimization

0 引言

针对各种路径问题的优化策略, 当前主要分为: 精确算法、经典启发式算法和元启发式算法。在精确算法中, 包括分支-约束算法^[1]、切平面算法^[2]和动态编程算法^[3]等, 但受计算复杂度的限制, 更适合解决较小规模的路径优化问题。另一方面, 传统的启发式算法包括邻域搜索算法^[4]和插入算法^[5]它们也受限于解的大小, 在小规模路径问题中表示出一定的实用性。相比之下, 元启发式算法凭借强大的全局搜索能力和灵活性, 在处理多目标和复杂的路径问题时显示出显著优势。这类算法包括模拟退火算法、帝企鹅优化算法^[6]、蝴蝶优化算法^[7]等, 通过模拟自然界中的过程或物理现象来探索问题的解空间。例如, Marinakis 等^[8]提出了一种改进粒子群算法的自适应策略, 并成功地应用于有时间窗的路径问题; Yu 等^[9]设计了一种新颖的信息素更新机

制, 以优化蚁群算法(ant colony optimizing, ACO), 从而有效解决了路径问题。这些研究凸显了元启发式算法在解决复杂 VRP 问题上的巨大潜力。

上述改进策略在一定程度上提高了算法的优化能力, 但仍存在局限性。如改进后的粒子群算法虽然提高了优化能力, 但收敛速度相对较慢; 改进后的蚁群算法虽然拓展了搜索空间, 但在全局搜索能力和局部搜索能力之间还没有达到理想的平衡。笔者提出一种多策略融合的改进灰狼算法(MGWO), 旨在进一步提高算法的收敛速度和寻优性能。MGWO 是基于灰狼优化器(grey wolf optimization, GWO)的进一步优化版本, GWO 由 Mirjalili 等^[10]于 2014 年提出的元启发式算法, 其灵感来源于自然界中狼的狩猎行为, 具有结构简单、易于实现、控制参数少等优点, 已被广泛应用于特征选择^[11]、图像分割、分类任务^[12]、故障诊断^[13]和网络覆盖优

收稿日期: 2024-09-20; 修回日期: 2024-10-20

基金项目: 广西哲学社会科学研究课题(24KSB008); 广西高等教育本科教学改革工程 A 类项目(2024JGA395)

第一作者: 文 竹(1982—), 女, 贵州人, 硕士。

化^[14]等领域。在 GWO 的基础上, 研究人员为进一步提高其性能做了许多有益的尝试。Yue 等^[15]通过引入平衡系数, 将焰火算法与 GWO 相结合; Tu 等^[16]通过采用全局最佳引导策略、自适应合作策略和分散觅食策略改进了 GWO, 并成功地将其应用于实际应用。Pradhan 等^[17]优化了灰狼觅食行为中的位置更新公式, 并将其应用于经济负荷调度问题。

鉴于 GWO 在初始阶段面临种群多样性不足、后期容易陷入局部最优的局限性, 通过多种策略的融合, 提出 MGWO, 以期保持 GWO 算法原有优势的同时, 进一步克服其局限性, 提高算法的整体性能。主要从 3 方面对 GWO 进行了优化:

1) 为了提高初始种群的多样性, MGWO 采用了精英反向策略进行种群初始化。通过生成与精英个体方向相反的个体, 有效扩大了搜索范围, 增加了种群的多样性, 从而帮助算法在全局范围内更有效地探索解空间。

2) 为了提高算法的收敛速度和精度, MGWO 引入了动态适应因子。可以根据算法的迭代过程动态调整搜索步长, 既能保证算法在初始阶段快速收敛, 又能避免后期因步长过大而错失最优解。通过这种策略, MGWO 在保持算法稳定性的同时, 显著提高了收敛性能。

3) 为了平衡算法的全局和局部搜索能力, MGWO 采用了分段搜索策略。可根据算法的搜索阶段和当前解的质量动态调整搜索方向和范围。在算法的早期阶段, 重点放在全局搜索上, 以探索更多潜在的解决方案; 而在算法的后期阶段, 重点放在局部搜索上, 以细化搜索并接近最优解。通过这种分段搜索方法, MGWO 在全局搜索和局部搜索能力之间实现了有效平衡。

1 路径优化模型

仿真实验以路径优化问题为模型, 其详细情况如下: K 辆车有效地抵达 n 个目的地, 确保从起点出发并最终返回起点的 n 个目的地全覆盖, 同时使这一系列路径活动的总成本最优化。细化如下:

1) 位置信息明确: 起点和 n 个目的地的地理位置信息是已知的, 为规划最优路径提供了基础数据支持。

2) 即时服务假设: 忽略在各目的地停留所需的时间, 并假定其为瞬时停留, 简化了时间的计算。

3) 一对一和一对多模式: 每个目的地只需一辆车抵达即可, 而同辆车可抵达多个目的地。

4) 统一燃料消耗标准: 所有车辆在行驶过程中每公里消耗的燃料量被视为相同, 从而便于计算和优化整体成本。

5) 匀速行驶条件: 假设所有车辆在整个过程中保持匀速。这一假设简化了速度变化对时间和能耗的影响, 使路径规划更加直观和高效。

该路径模型是一个复杂的组合优化问题, 需要在满足上述约束条件的前提下, 探索成本最低、效率最高的路径方案。由此可得该研究涉及的已知参数如表 1 所示。

表 1 已知参数

parameters	describe	value
n	目的地数量	-
K	车辆数量	-
V/h	车辆速度	40.00
C_z/km	车辆单位里程折旧费	100.00
C_w/km	车辆单位里程维护费	100.00
C_r/h	每辆车单位时间内的人工成本	100.00
C_f/km	车辆的单位里程燃油价格	100.00
T	车辆一次通行最大行驶时间	3.00
mu	车辆折旧率	0.08
Q_k/km	空载时单位距离燃油消耗量	8.18
d_{ij}/km	目的地 i 到 j 的距离	-

根据表 1 所述, 本实验的总成本由车辆折旧 C_1 、车辆维修 C_2 、人工成本 C_3 、车辆燃料成本 C_4 和车辆损耗成本 C_5 组成, 并将其作为目标函数建模, 其目标函数模型 Z 为:

$$Z=\min\{1+2+3+4+5\}。 \quad (1)$$

2 基本灰狼算法

灰狼算法是一种智能优化算法, 其灵感来源于自然界中灰狼的捕食行为。灰狼在捕食活动中遵循严格的等级制度: α 狼作为等级最高的个体, 象征着最优解, 即最接近理想猎物(即最优目标)的解; β 狼紧随其后, 代表次优个体; δ 狼被排在第三等级。在捕食过程中, α 狼扮演领导者的角色, 引领狼群的方向; β 狼和 δ 狼作为辅助决策者, 协助 α 狼制定捕食策略; 其余的狼群成员, 即 ω 狼, 负责执行领头狼制定的决策。GWO 算法建模如下:

1) 搜索猎物。

$$D=|C \cdot X_p(t)-X(t)|; \quad (2)$$

$$X(t+1)=X_p(t)-A \cdot D. \quad (3)$$

式中: D 为灰狼与目标猎物之间的距离; t 为当前迭代次数; $X_p(t)$ 为当前猎物的位置; $X(t)$ 为当前狼的个体位置; $X(t+1)$ 为下一代狼的更新位置; A 为系数向量。

$$A=2ar_1-a; \quad (4)$$

$$C=2r_2; \quad (5)$$

$$a=2(1-t/T_{\max}) \quad (6)$$

式中: A 随 a 值变化; a 值从 2 到 0 线性递减; C 为系数向量; T_{\max} 为最大迭代次数。

2) 捕获猎物。

在狩猎过程中, 新一代个体的位置由适应性最好的前三名个体 α 狼、 β 狼和 δ 狼共同决定; α 狼指挥狼群中的其他个体攻击猎物, 其数学模型如下:

$$D_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}(t)|; \quad (7)$$

$$D_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}(t)|; \quad (8)$$

$$D_\delta = |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}(t)|; \quad (9)$$

$$X_1 = \mathbf{X}_\alpha - A_1 \cdot D_\alpha; \quad (10)$$

$$X_2 = \mathbf{X}_\beta - A_2 \cdot D_\beta; \quad (11)$$

$$X_3 = \mathbf{X}_\delta - A_3 \cdot D_\delta; \quad (12)$$

$$\mathbf{X}(t+1) = (X_1 + X_2 + X_3)/3. \quad (13)$$

式中: D_i ($i=\alpha, \beta, \delta$) 为灰狼 i 与目标猎物之间的距离; X_i ($i=\alpha, \beta, \delta$) 为灰狼 i 的位置; $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ 为随机向量。

3 改进灰狼算法

3.1 算法思想

灰狼算法作为一种高效的智能优化算法, 凭借其卓越的鲁棒性和简洁的结构设计, 在解决各类优化问题中展现出了广泛的应用潜力。然而, 该算法在初期阶段收敛速度相对缓慢, 且在后期局部搜索能力较差, 一定程度上限制了其性能发挥。针对灰狼算法存在的这些缺陷, 提出一系列针对性的改进措施, 旨在提升其收敛速度和强化其局部搜索能力。

1) 引入精英反向初始化策略。

精英反向策略自 2005 年问世以来, 其核心思想在于通过引入反向解来拓宽算法的搜索范围, 从而增强解的多样性。为了进一步提高解的质量, 笔者将精英反向策略融入灰狼算法中。灰狼算法在初始化阶段具有一定的随机性, 导致灰狼初始个体的分布不够均匀, 进而引发个体间的重复性。为解决这一问题, 通过引入精英反向策略以提高初始化种群质量。在具体实施时, 笔者比较了初始化个体与其对应的反向解个体的适应度值。若初始化个体的适应度值优于其反向解个体, 则保留该初始化个体; 反之, 若其适应度值不及反向解个体, 则选择保留反向解个体。这一策略确保了种群中保留的个体均为较优解, 从而有助于算法在后续的迭代过程中更快地收敛至全局最优解。精英反向策略建模如下:

$$\overrightarrow{X} = r(lb + ub) - X_i. \quad (14)$$

式中: \overrightarrow{X} 为精英反向学习产生的个体; r 为介于 $(0, 1)$ 之间的随机数; lb 为可行解的上限; ub 为可行解的下限; X_i 为当前个体。

$$X_i = \begin{cases} \overrightarrow{X}_i, & f(\overrightarrow{X}_i) < f(X_i) \\ X_i, & f(\overrightarrow{X}_i) \geq f(X_i) \end{cases}. \quad (15)$$

通过式(15)优化后, 将随机初始化的个体适应度值与精英逆向策略初始化的个体适应度值进行比较, 选出更优秀的个体初始化值。

2) 引入动态权重因子。

根据式(14)可知: 在灰狼算法中更新最优位置时, 3 头领导狼 α, β, δ 对最终狼群个体位置更新的贡献被视为平等的, 但在实际应用过程中, 尤其在算法初期, α 狼的位置往往明显优于 β 狼和 δ 狼, 因此 α 狼应当占据主导地位, 发挥更为关键的领导作用。随着迭代次数的不断增加, α 狼的领导影响力逐渐减弱, 而 β 狼和 δ 狼的领导作用则逐渐增强。

为了更真实地反映这一动态变化过程, 引入了动态权重因子 w 。在狼群寻找最优解的过程中, 通过添加这个动态因子, 可以根据迭代次数的变化, 动态地调整 3 头领导狼 α, β, δ 的领导作用。这样不仅能够更准确地模拟自然界中灰狼种群的领导机制, 而且能够有效提升算法的性能, 使其更快地收敛到全局最优解。数学建模如下:

$$w = \sin(\pi * (t/T_{\max})) + 0.5; \quad (16)$$

$$X_{11} = w \mathbf{X}_\alpha - A_1 \cdot D_\alpha; \quad (17)$$

$$X_{22} = w \mathbf{X}_\beta - A_2 \cdot D_\beta; \quad (18)$$

$$X_{33} = w \mathbf{X}_\delta - A_3 \cdot D_\delta; \quad (19)$$

$$\mathbf{X}(t+1) = (X_{11} + X_{22} + X_{33})/3. \quad (20)$$

3) 引入分阶段搜索。

智能优化算法在迭代后期一般都存在易陷入局部最优的缺点, 为了克服这一局限性, 同时加速算法在初期的收敛速度, 并更有效地平衡全局探索与局部挖掘的能力, 采用了分阶段搜索策略。其数学建模如下:

$$X(t+1) = \begin{cases} (X_1 + X_2 + X_3)/3, & p \geq 0.5 \\ (X_{11} + X_{22} + X_{33})/3, & p < 0.5 \end{cases}. \quad (21)$$

式中 p 为 $(0, 1)$ 之间的随机数。

MGWO 算法如下:

通过式(15)初始化种群

初始化参数 a, A, C

while $t < T_{\max}$

```

for each wolf
    更新参数  $a$ 、 $A$ 、 $C$ 
    if  $p \geq 0.5$ 
        通过式 13)更新灰狼位置
    else
        通过式 20)更新灰狼位置
    end if
end for
检查更新灰狼位置可行性
 $t+1$ 

```

3.2 时间复杂度分析

假设种群规模为 N , 搜索空间维数为 D , 最大迭代次数为 T 。GWO 算法的复杂度包括: 种群初始化复杂度为 $O(ND)$, 迭代优化复杂度为 $O(NT)$, 那么 GWO 的时间复杂度如下:

$$O(\text{GWO})=O(ND)+O(NT)。 \quad (22)$$

即 MGWO 时间复杂度为:

$$O(\text{MGWO})=O(ND)+O(NT)。 \quad (23)$$

综上所述, 提出的 MGWO 算法并没有增加算法的时间复杂度。

4 仿真实验和结果分析

4.1 实验环境

仿真实验在 Matlab 2024a 编程环境下进行, 硬件平台配备 Intel(R) Core(TM) i7-10700 CPU(主频 2.90 GHz)、8 GB 内存和 Windows 11 64 位操作系统。为了简化研究的复杂性, 2 个目的地之间的距

离根据直线距离计算。设定种群规模为 30, 最大迭代次数为 500, 并独立重复实验 20 次以确保实验结果的稳定性和可靠性。为了全面评估算法性能, 设计了 4 个不同的测试场景:

- 场景 1: 部署 3 台车抵达 40 个目的地;
- 场景 2: 部署 4 台车抵达 50 个目的地;
- 场景 3: 部署 5 台车抵达 60 个目的地;
- 场景 4: 部署 6 台车抵达 60 个目的地。

在这些场景下进行仿真实验, 并将所提出的 MGWO 算法与多种先进的优化算法进行对比, 包括 GWO、融合固定点分布的灰狼优化算法 (FPD-GWO)^[18]、蝴蝶优化算法 (butterfly optimization algorithm, BOA)、粒子群优化算法 (particle swarm optimization, PSO) 以及蚁群优化算法 (ant colony optimization, ACO)。通过一系列对比, 旨在深入探究 MGWO 算法在解决路径优化问题上的优越性。实验中各算法的参数配置详见表 2。

表 2 算法参数设置表

methods	parameter
MGWO	$T_{\max}=500$
GWO	$T_{\max}=500$
FPD_GWO	$T_{\max}=500$
BOA	$P=0.8$, power_exponent=0.1, sensory_modality=0.01, $T_{\max}=1500$
PSO	$T_{\max}=500$, $c_1=1.41$, $c_2=1.41$
ACO	$T_{\max}=500$, $\rho=0.5$, $\alpha=1$, $\beta=1$

4.2 实验结果

经过 20 次重复实验, MGWO 与多种算法的各场景实验结果如表 3 和图 1 所示。

表 3 适应度实验结果表

scenario	methods	min	max	average	SD
场景 1	MGWO	1.533 047e+07	1.829 395e+07	1.666 923e+07	1.502 291e+06
	GWO	1.616 940e+07	1.663 755e+07	1.632 608e+07	2.697 430e+05
	FPD-GWO	1.946 014e+07	2.007 295e+07	1.974 259e+07	3.092 029e+05
	BOA	1.760 707e+07	1.823 950e+07	1.801 875e+07	3.568 358e+05
	PSO	1.788 897e+07	1.803 185e+07	1.797 833e+07	7.789 549e+04
	ACO	1.702 542e+07	1.793 193e+07	1.748 168e+07	4.532 846e+05
场景 2	MGWO	1.179 380e+07	1.689 788e+07	1.359 557e+07	2.863 846e+06
	GWO	1.308 710e+07	1.395 604e+07	1.361 096e+07	4.612 272e+05
	FPD-GWO	1.674 832e+07	2.104 922e+07	1.914 861e+07	2.193 556e+06
	BOA	1.515 490e+07	1.619 109e+07	1.564 994e+07	5.196 352e+05
	PSO	1.371 105e+07	1.451 037e+07	1.417 900e+07	4.167 971e+05
	ACO	1.458 372e+07	1.586 562e+07	1.504 177e+07	7.149 597e+05
场景 3	MGWO	1.468 555e+07	2.202 093e+07	1.879 219e+07	3.745 656e+06
	GWO	1.603 782e+07	1.669 984e+07	1.632 158e+07	3.409 743e+05
	FPD-GWO	2.388 155e+07	2.651 970e+07	2.536 545e+07	1.349 614e+06
	BOA	1.907 092e+07	2.096 899e+07	1.984 068e+07	9.985 480e+05
	PSO	1.723 572e+07	2.032 659e+07	1.880 964e+07	1.546 225e+06
	ACO	1.752 414e+07	1.973 297e+07	1.853 167e+07	1.117 093e+06
场景 4	MGWO	1.210 407e+07	1.313 322e+07	1.257 451e+07	5.202 252e+05
	GWO	1.277 337e+07	1.313 517e+07	1.295 765e+07	1.809 929e+05
	FPD-GWO	2.016 315e+07	2.299 148e+07	2.180 395e+07	1.467 632e+06
	BOA	1.405 098e+07	1.672 211e+07	1.578 670e+07	1.504 693e+06
	PSO	1.354 030e+07	1.532 316e+07	1.469 223e+07	9.991 203e+05
	ACO	1.380 367e+07	1.560 427e+07	1.499 739e+07	1.033 842e+06

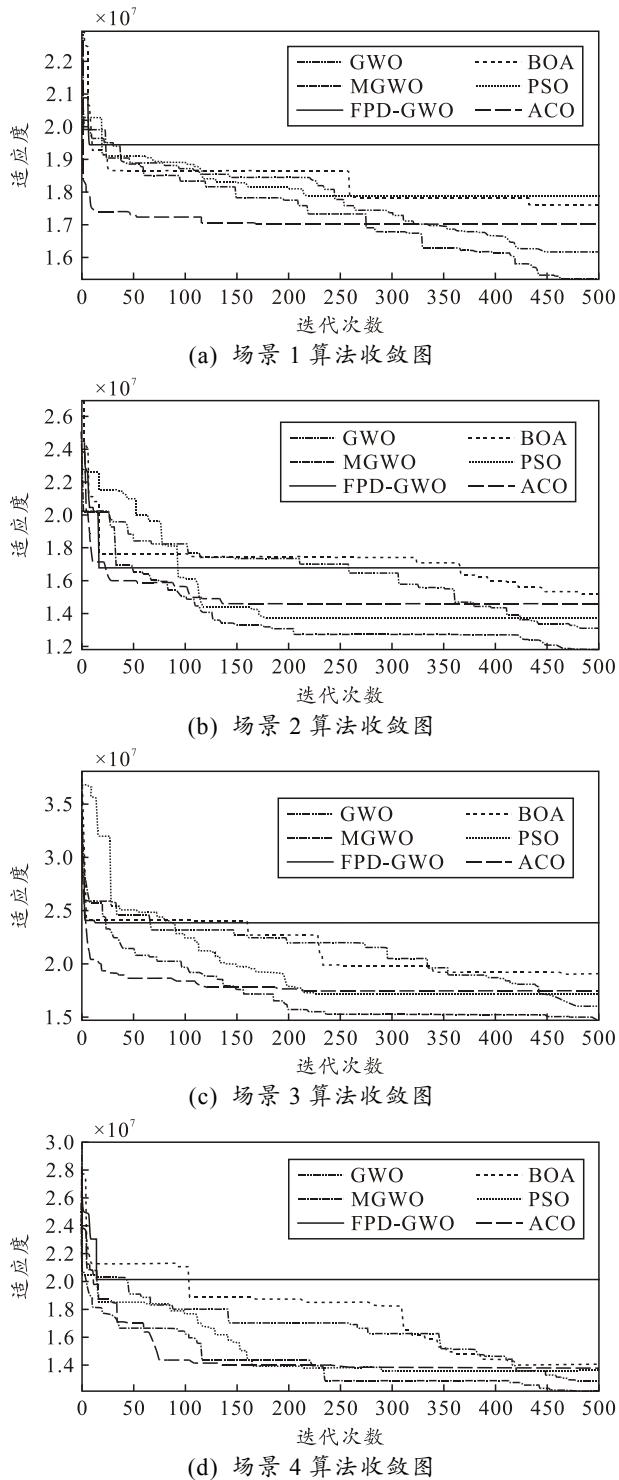


图 1 各场景算法分布收敛图

如表 3 和图 1 所示, 场景 1 在 3 辆车和 40 个目的地的情况下, MGWO 算法成功地将总成本的适应度降低至约 1.53×10^7 , 相较于 GWO 算法的 1.61×10^7 , 节省成本约 8×10^5 。更重要的是, MGWO 算法不仅在最小值上表现优异, 其最大值和平均值也均低于其他对比算法。场景 2 车辆增加至 4 辆, 目的地达到 50 个, MGWO 算法找到的最低总成本

适应度约为 1.18×10^7 , 相较于 GWO 算法的 1.31×10^7 , 节省成本约 1.3×10^6 , 且其平均值依然领先于其他算法。场景 3 车辆增至 5 辆, 目的地增加到 60 个。MGWO 算法以约 1.47×10^7 的总成本适应度最小值脱颖而出, 比 GWO 算法的 1.60×10^7 节省成本约 1.3×10^6 , 在所有对比算法中, MGWO 算法的最小值依然是最小的, 再次体现了其高效性和准确性。最后在场景 4, 车辆增加到 6 辆, 目的地数量仍为 60 个。MGWO 算法以约 1.21×10^7 的总成本适应度最小值领先, 相较于 GWO 算法的 1.27×10^7 , 节省成本约 6×10^5 。此外, MGWO 算法的平均值也低于其他对比算法, 进一步彰显了其稳定性和可靠性。

从实验结果得知: 无论在哪个场景, MGWO 算法都能迅速找到成本最低的路径适应度, 相较于 GWO、FPD-GWO、BOA、PSO 和 ACO 等算法, 其性能均表现出色, 表明 MGWO 算法在最优路径问题上的高效性和实用性, 并能应用到精准打击、物流配送等应用场景的最优路径决策上。

5 结论

该研究旨在通过优化策略, 降低路径成本, 提出了 MGWO 算法。针对传统灰狼算法存在的收敛速度慢、收敛精度不足等局限性实施改进与提升, 融合精英反向策略以初始化种群、引入动态因子与分段搜索策略, 更精准地更新个体位置, 从而加速收敛过程并提高解的质量。仿真实验结果表明, 在复杂问题中 MGWO 算法展现出了卓越的性能, 取得了令人满意的实验结果。在未来的工作中, 笔者将继续深化 MGWO 算法的研究与应用, 探索其在多目标、多载体等更复杂场景下的适用性与效能。

参考文献:

- [1] THEURICH F, FISCHER A, SCHEITHAUER G. A branch-and-bound approach for a vehicle routing problem with customer costs[J]. EURO Journal on Computational Optimization, 2021, 9: 100003.
- [2] BRUGLIERI M, MANCINI S, PISACANE O. A more efficient cutting planes approach for the green vehicle routing problem with capacitated alternative fuel stations[J]. Optimization Letters, 2021, 15(8): 2813–2829.
- [3] ULMER M W, GOODSON J C, MATTFELD D C, et al. On modeling stochastic dynamic vehicle routing problems[J]. EURO Journal on Transportation and Logistics, 2020, 9(2): 100008.

- [4] MASUDIN I, SA'DIYAH R F, UTAMA D M, et al. Capacitated vehicle routing problems: nearest neighbour vs tabu search[J]. International Journal of Computer Theory and Engineering, 2019, 11(4): 76–79.
- [5] BREUNIG U, BALDACCI R, HARTL R F, et al. The electric two-echelon vehicle routing problem[J]. Computers & Operations Research, 2019, 103: 198–210.
- [6] YANG Z, DENG L B, WANG Y, et al. Apenodytes forsteri optimization: algorithm and applications[J]. Knowledge-based Systems, 2021, 232: 107483.
- [7] 浦徐进, 李丽超, 付亚平. 改进蝴蝶优化算法在护理人员路径规划与调度中的应用[J]. 控制工程, 2024, 31(2): 281–287.
- [8] MARINAKIS Y, MARINAKI M, MIGDALAS A. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows[J]. Information Sciences, 2019, 481: 311–329.
- [9] YU B, YANG Z Z, YAO B. An improved ant colony optimization for vehicle routing problem[J]. European journal of operational research, 2009, 196(1): 171–176.
- [10] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46–61.
- [11] AL-TASHI Q, MD RAIS H, ABDULKADIR S J, et al. A review of grey wolf optimizer-based feature selection methods for classification[J]. Evolutionary Machine Learning Techniques, 2020, 1: 273–286.
- [12] RUBINI L J, PERUMAL E. Hybrid kernel support vector machine classifier and grey wolf optimization algorithm based intelligent classification algorithm for chronic kidney disease[J]. Journal of Medical Imaging and Health Informatics, 2020, 10(10): 2297–2307.
- [13] 曾耀传. 基于 GWO-VMD-LSSVM 的滚动轴承故障诊断[J]. 自动化技术与应用, 2024, 43(11): 24–28.
- [14] 荀平章, 郭保永, 郭苗. 基于改进 GWO 和贪婪算法的覆盖优化方法 [J]. 传感技术学报, 2024, 37(9): 1602–1612.
- [15] YUE Z, ZHANG S, XIAO W. A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm[J]. Sensors, 2020, 20(7): 2147.
- [16] TU Q, CHEN X, LIU X. Multi-strategy ensemble grey wolf optimizer and its application to feature selection[J]. Applied Soft Computing, 2019, 76: 16–30.
- [17] PRADHAN M, ROY P K, PAL T. Grey wolf optimization applied to economic load dispatch problems[J]. International Journal of Electrical Power & Energy Systems, 2016, 83: 325–334.
- [18] DEREKI S. A new modified grey wolf optimization algorithm proposal for a fundamental engineering problem in robotics[J]. Neural Computing and Applications, 2021, 33(21): 14119–14131.
-

(上接第 10 页)

- [8] 韩俊男. 浅析微服务架构在企业信息系统构建中的应用价值[J]. 中国管理信息化, 2018, 21(2): 53–54.
- [9] WASEEM M, LIANG P, SHAHIN M. A systematic mapping study on microservices architecture in devops[J]. Journal of Systems and Software, 2020, 170: 110798.
- [10] 吴斌峰. 基于微服务架构的物联网中间件设计[J]. 计算机科学, 2019, 46(S1): 580–584.
- [11] 蒋勇. 基于微服务架构的基础设施设计[J]. 软件, 2016, 37(5): 93–97.
- [12] ELKHOLY M, EIFATATRY A. Framework for interaction between databases and microservice architecture[J]. IT Professional, 2019, 21(5): 57–63.

- [13] 刘从军, 刘毅. 基于微服务的维修资金管理系统[J]. 计算机系统应用, 2019, 28(4): 52–60.
- [14] YOON K S, KIM Y H. A Design and Implementation of Integrated Content Management System Based on Microservices Architecture[J]. KIPS Transactions on Software and Data Engineering, 2019, 8(3): 97–108.
- [15] 吴磊, 湛健, 宋丽华. 微服务架构在智能家居网关系统中的应用研究[J]. 计算机技术与发展, 2019, 29(11): 200–205.