



基于平衡鲸鱼优化算法的无人车路径规划

蔡雨岑, 杜鹏桢

引用本文:

蔡雨岑, 杜鹏桢. 基于平衡鲸鱼优化算法的无人车路径规划[J]. 控制与决策, 2021, 36(11): 2647–2655.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0416>

您可能感兴趣的其他文章

Articles you may be interested in

嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法

Whale optimization algorithm for embedded Circle mapping and one-dimensional oppositional learning based small hole imaging
控制与决策. 2021, 36(5): 1173–1180 <https://doi.org/10.13195/j.kzyjc.2019.1362>

基于改进蚁群算法的水面无人艇路径规划

Path planning for unmanned surface vehicle based on improved ant colony algorithm
控制与决策. 2021, 36(4): 847–856 <https://doi.org/10.13195/j.kzyjc.2019.0839>

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>

一种基于免疫机理的确定性移动机器人路径规划算法

A path planning algorithm of deterministic mobile robot based on immune mechanism
控制与决策. 2021, 36(10): 2418–2426 <https://doi.org/10.13195/j.kzyjc.2020.0059>

凸优化与A*算法结合的路径避障算法

Convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm
控制与决策. 2020, 35(12): 2907–2914 <https://doi.org/10.13195/j.kzyjc.2019.0351>

基于平衡鲸鱼优化算法的无人车路径规划

蔡雨岑, 杜鹏桢[†]

(南京理工大学 计算机科学与工程学院, 南京 210094)

摘要: 针对地面无人车辆路径规划问题的特点, 提出基于和声二次优化的平衡鲸鱼算法求解最优路径. 首先, 利用和声搜索算法二次优化来改善种群质量和全局探索能力, 依据解的适应性进行微调, 提高求解精度; 然后, 引入动态平衡策略和种群重构机制, 跟踪种群最优解状态以协调全局探索与局部开发能力, 出现优化停滞时重构种群以增加多样性, 避免陷入局部最优; 最后, 基于不同环境进行仿真实验, 与多种算法进行对比分析, 研究结果表明了所提出算法在复杂环境下的可行性和有效性, 为鲸鱼优化算法在路径规划问题中的应用提供一种新思路.

关键词: 鲸鱼优化算法; 路径规划; 和声搜索; 动态平衡; 地面无人车辆

中图分类号: TP242

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0416

开放科学(资源服务)标识码(OSID):



引用格式: 蔡雨岑, 杜鹏桢. 基于平衡鲸鱼优化算法的无人车路径规划[J]. 控制与决策, 2021, 36(11): 2647-2655.

Path planning of unmanned ground vehicle based on balanced whale optimization algorithm

CAI Yu-cen, DU Peng-zhen[†]

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: According to the characteristics of the path planning problem of ground unmanned vehicles, a balanced whale algorithm based on harmonic quadratic optimization is proposed. Firstly, the optimization of harmony search algorithm is used to improve the population quality and global exploration ability, and fine-tuning is carried out according to the adaptability of the solution to improve the accuracy. Then, the dynamic balance strategy and the population reconstruction mechanism are introduced to track the optimal solution state of the population to coordinate the global exploration and local development ability, and when the optimization stagnates, the population is reconstructed to increase diversity and avoid falling into the local optimization. Finally, simulation experiments are carried out based on different environments, and compared with a variety of algorithms, which proves the feasibility and effectiveness of the proposed algorithm, providing a new idea for the application of whale optimization algorithm in path planning.

Keywords: whale optimization algorithm; path planning; harmony search; dynamic equilibrium; unmanned ground vehicle

0 引言

路径规划是无人驾驶技术研究领域的一个基本问题,其目的是在部分已知环境中找到一条满足一定约束的可行路径,属于NP (non-deterministic polynomial)完全问题^[1]. 无人车辆的路径规划任务通常涉及更大的任务空间和更复杂的实际环境,车体自由度限制以及环境不确定性等影响因素往往会导致问题的计算复杂度呈指数级增长. 通常,无人车路径规划会对路径有更多的要求,而不仅仅以路径长度作为评判标准,例如路径拐点、转弯角度等,都会对无

人车的行驶造成一定影响. 目前,已有诸多算法,如人工势场法、A*算法、概率路线图法^[2]和快速扩展随机树法^[3-5]等,已被应用于路径规划研究. 文献[2]通过生成路径样本来增强概率路线图法应对复杂环境的能力,但抽样过程的随机性导致算法的完备性较弱,稳定性低. 文献[3-5]改造的快速扩展随机树法能够快速完成高维空间中的路径规划,但在低维空间中结果不唯一且效果不佳. 由于复杂环境下求解路径规划问题的高计算复杂度以及智能优化算法在解决NP问题中表现出的计算高效性,许多智能优化算法,

收稿日期: 2020-04-13; 修回日期: 2020-09-08.

基金项目: 国家自然科学基金项目(71971116); 国家重点研发计划项目(2018YFB1601100, 2018YFB1601101).

责任编辑: 张文安.

[†]通讯作者. E-mail: h.k@foxmail.com.

如蚁群算法^[6]、粒子群算法^[7]等,也被广泛应用于路径规划方面的研究并取得了显著效果. 蚁群算法具有较好的鲁棒性和适应性,但搜索效率较低,容易陷入停滞和局部最优;粒子群算法收敛速度快,但容易产生早熟,局部寻优能力较差.

鲸鱼优化算法 (whale optimization algorithm, WOA) 是 Seyedali 等^[8]提出的一种新型智能优化算法,具有易于实现、控制参数少、鲁棒性强等优点. 目前,鲸鱼优化算法已被广泛应用于连续域优化问题,但在路径规划方面的改进和应用还有待完善. 文献[9]使用余弦曲线优化参数并同步加入惯性权重,提高了鲸鱼算法的前期探索能力和后期收敛速度. 文献[10]采用混沌映射优化参数,增强了算法的全局寻优能力,但算法性能与所使用的混沌映射强相关,不具备通用性. 文献[11]引入莱维飞行和差异进化提升算法整体性能,但算法的计算复杂度较高,在大规模优化问题上的表现欠佳.

针对鲸鱼优化算法易出现局部最优和全局搜索能力弱的问题,本文提出一种基于和声二次优化的平衡鲸鱼算法,通过和声搜索优化种群来增强算法的全局探索能力;依据解的质量动态平衡全局探索与局部开发过程,避免陷入局部最优解. 本文针对两种改进分别进行实验验证,并在3种模拟环境中与传统A*算法及多种不同智能优化算法进行路径规划的对比,实验结果充分表明了本文改进算法的有效性和可行性.

1 基本鲸鱼优化算法

鲸鱼优化算法模拟座头鲸的狩猎行为,将问题的每个方案都视为一条鲸鱼,每条鲸鱼在捕猎时采用随机探索机制搜寻猎物,在发现猎物后使用收缩包围和螺旋气泡网攻击两种方式发起攻击. 假设种群中共有 n 条鲸鱼在 d 维空间中觅食,第 i 条鲸鱼的位置可用 d 维向量 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 表示,初始化的鲸鱼群随机且均匀分布在搜索空间中. 鲸鱼优化算法的3种位置更新机制描述如下.

1.1 收缩包围机制

座头鲸在发现猎物后以收缩包围的方式靠近猎物,位置更新公式如下:

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A} \times \mathbf{D}, \quad (1)$$

$$\mathbf{D} = |\mathbf{C} \times \mathbf{X}^*(t) - \mathbf{X}(t)|. \quad (2)$$

其中: t 表示当前迭代次数; $\mathbf{A}$ 和 $\mathbf{C}$ 是系数向量; $\mathbf{X}^*$ 是当前种群中最优解的位置向量; $\mathbf{X}$ 是当前个体的位置向量,其维度由搜索空间的维度决定; $\mathbf{D}$ 和 $\mathbf{A}$ 控

制包围步长. 系数 $\mathbf{A}$ 和 $\mathbf{C}$ 由下式计算:

$$\mathbf{A} = 2a\mathbf{r} - a, \quad (3)$$

$$\mathbf{C} = 2\mathbf{r}. \quad (4)$$

其中: a 为收敛因子,随着迭代的进行从2线性减小到0,以 $a = 2 - \frac{2 \times t}{t_{\max}}$ 计算; $\mathbf{r}$ 是在 $[0,1]$ 内均匀分布的随机向量. 在收缩包围机制下,每条鲸鱼根据种群当前最优位置更新自身位置,调整系数向量 $\mathbf{A}$ 和 $\mathbf{C}$ 的值可以控制鲸鱼在猎物附近搜索,通过减小 a 值可以实现收缩包围行为.

1.2 螺旋更新机制

螺旋更新机制使用对数螺旋方程来更新位置,公式如下:

$$\mathbf{X}(t+1) = \mathbf{D}' \times e^{bl} \times \cos(2\pi l) + \mathbf{X}^*(t). \quad (5)$$

其中: $\mathbf{D}' = |\mathbf{X}^*(t) - \mathbf{X}(t)|$ 表示当前鲸鱼与猎物之间的距离, b 是定义对数螺旋形状的常数,随机数 $l \in [-1, 1]$. 座头鲸在捕猎时会不断收缩包围圈并同时沿着螺旋路径在猎物周围游动,假设座头鲸选择收缩包围机制或螺旋更新机制来靠近猎物的几率各为50%,则捕猎行为可由下式表示:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}^*(t) - \mathbf{A} \times \mathbf{D}, & p < 0.5; \\ \mathbf{D}' \times e^{bl} \times \cos(2\pi l) + \mathbf{X}^*(t), & p \geq 0.5. \end{cases} \quad (6)$$

其中 p 为属于 $[0,1]$ 的随机数.

1.3 搜索猎物机制

在尚未确定猎物的大致位置之前,为了增强对捕猎空间的探索,座头鲸将使用彼此之间的位置作为参考,搜索猎物机制的位置更新公式如下:

$$\mathbf{X}(t+1) = \mathbf{X}_{\text{rand}} - \mathbf{A} \times \mathbf{D}. \quad (7)$$

其中: $\mathbf{X}_{\text{rand}}$ 表示鲸鱼群中随机个体的位置, $\mathbf{D} = |\mathbf{C} \times \mathbf{X}_{\text{rand}} - \mathbf{X}(t)|$ 表示当前个体与随机鲸鱼个体之间的距离,系数向量 $\mathbf{A}$ 和 $\mathbf{C}$ 的定义与式(3)和(4)相同.

鲸鱼优化算法从一组均匀分布的随机解开始,当 $|\mathbf{A}| > 1$ 时,算法处于探索模式,鲸鱼个体将以种群内的随机个体位置为指导探索解空间,或以50%的概率采用螺旋更新机制探索;当 $|\mathbf{A}| \leq 1$ 时,算法处于开发模式,鲸鱼个体将以当前种群内最优个体位置为指导,并以各50%的概率选择收缩包围机制或螺旋更新机制更新位置.

2 基于和声二次优化的平衡鲸鱼算法

2.1 基于和声搜索的二次优化

和声搜索算法是由 Geem 等^[12]提出的智能优化算法,被应用于多种优化问题并取得了较好的效果. 算法中所有解的集合称为和声记忆库(HM),库中的每个和声代表问题的一个解 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{id})$. 进行优化时,生成两个属于 $[0, 1]$ 的随机数 r_1 、 r_2 ,若 r_1 小于记忆库取值概率(HMCR),则从记忆库中随机选择变量作为新变量,此时如果 r_2 小于微调概率(PAR),则以带宽(BW)对其进行微调;若 r_1 大于 HMCR,则从解空间随机生成一个新的变量. 最后,在新和声与当前 HM 中最差的和声之中选取较优者加入和声库.

基本鲸鱼算法在优化初期使用式(7)更新鲸鱼位置,通过系数向量 $\mathbf{A}$ 强制鲸鱼远离当前最优值以扩大搜索范围,但参考方向仍在原始种群中选择,全局搜索能力受限于种群质量,降低了算法的稳定性. 本文针对该问题,将和声搜索思想融入鲸鱼优化算法中,将每次迭代后的鲸鱼种群进行二次优化,对种群最优个体按概率 PAR 进行幅度为 BW 的微调,而对于从原始种群中随机选择的个体,通过评估其适应度水平采用不同的微调方式. 适应度优于平均水平的个体参考自身位置与最优位置进行小范围抖动,以增强局部优化能力;适应度在平均水平之下的个体参考自身位置与其他个体位置进行微调,扩大搜索范围,进一步提升种群范围内的探索能力. 微调方式的数学表述如下:

$$\mathbf{X}_i = \begin{cases} \mathbf{X}_i + \text{rand}(0, 1) \times (\mathbf{X}^* - \mathbf{X}_i), & f_i < f_{\text{avg}}; \\ \mathbf{X}_i + \text{rand}(0, 1) \times (\mathbf{X}_{\text{rand}} - \mathbf{X}_i), & f_i \geq f_{\text{avg}}. \end{cases} \quad (8)$$

每代鲸鱼中的精英个体将会以一定的比例得以保留,在改善种群质量的同时,通过调整记忆库取值概率和微调概率能够在原始种群内外空间采样生成新个体,在很大程度上丰富了种群多样性,增强了鲸鱼算法的全局搜索能力.

2.2 动态平衡策略

群智能优化算法的性能与探索、开发过程的平衡密切相关,鲸鱼优化算法也由探索和开发两种模式构成,模式间的切换时机和持续时间对算法的性能有着很大的影响. 基本鲸鱼算法通过系数向量 $\mathbf{A}$ 控制全局探索与局部开发过程的切换,其变化不能反映优化进行的程度和最优解的状态. 当 $|\mathbf{A}| \leq 1$ 时,所有个体以当前最优个体位置为参考进行位置更新,加快收敛的同时也急剧降低了种群多样性;如果在 $|\mathbf{A}| > 1$

的探索过程中没有覆盖到全局最优解,则会极大增加算法最终陷入局部最优的可能性.

鉴于上述问题,本文引入平衡因子 balance 代替收敛因子 a ,其计算公式为

$$\text{balance} = \begin{cases} 1 - \text{balance}, & \text{count} > \text{thr}; \\ \text{balance}, & \text{count} \leq \text{thr}. \end{cases} \quad (9)$$

其中:计数器 count 记录最优解的状态;thr 为切换阈值,控制模式切换的时机. 在算法的初始化阶段,设 $\text{balance} = 0.8$,代表全局探索行为在鲸鱼的行为选择中占有更大比重,算法处于探索模式. 每轮迭代结束时,若最优解发生变化则重置计数器,否则计数器加 1. 当计数器累计超过切换阈值时进行搜索模式的切换,同时重置计数器,此时局部开发行为在鲸鱼的行为选择中占优,算法进入开发模式.

若算法在开发阶段仍触发了搜索模式切换,则表明可能出现了优化停滞. 为了保留历史优化结果并增加种群多样性以跳出局部最优,此时以锦标赛选择方法从原种群中选择包含最优解在内的 20% 个体保留,而剩余 80% 的个体将被舍弃并替换成随机生成的新个体. 种群重构之后,平衡因子和计数器等相关参数同样被还原.

2.3 改进算法具体步骤

基于和声二次优化的平衡鲸鱼算法(harmony search-based balanced whale optimization algorithm, HS-WOA)流程如图 1 所示,具体步骤如下.

step 1: 设置种群大小、最大迭代次数等相关参数,初始化鲸鱼种群.

atep 2: 计算所有个体适应度,更新最优个体 $\mathbf{X}^*$.

step 3: 计算并更新参数 $\{a, \mathbf{A}, \mathbf{C}, p, f_{\text{avg}}\}$.

step 4: 生成随机数 $R = \text{rand}(0, 1)$. 若 $R \geq \text{balance}$,则根据式(6)对鲸鱼个体进行位置更新;若 $R < \text{balance}$,则根据式(7)对鲸鱼个体进行位置更新.

step 5: 使用改造的和声搜索算法对鲸鱼种群进行二次优化,若新生成的鲸鱼个体适应度优于原种群中最差鲸鱼个体,则使用新生成的鲸鱼位置替代最差鲸鱼的位置,直到整个鲸鱼种群完成位置更新为止.

step 6: 若最优鲸鱼个体的位置 $\mathbf{X}^*$ 发生改变,则重置计数器 $\text{count} = 0$;若未发生改变,则计数器加 1.

step 7: 计数器超过切换阈值时进行搜索模式切换,令 $\text{balance} = 1 - \text{balance}$,同时重置计数器. 若在开发模式下发生切换,则用锦标赛方法选择包含当前最优的 20% 个体保留,随机生成 80% 个体进行种群重构.

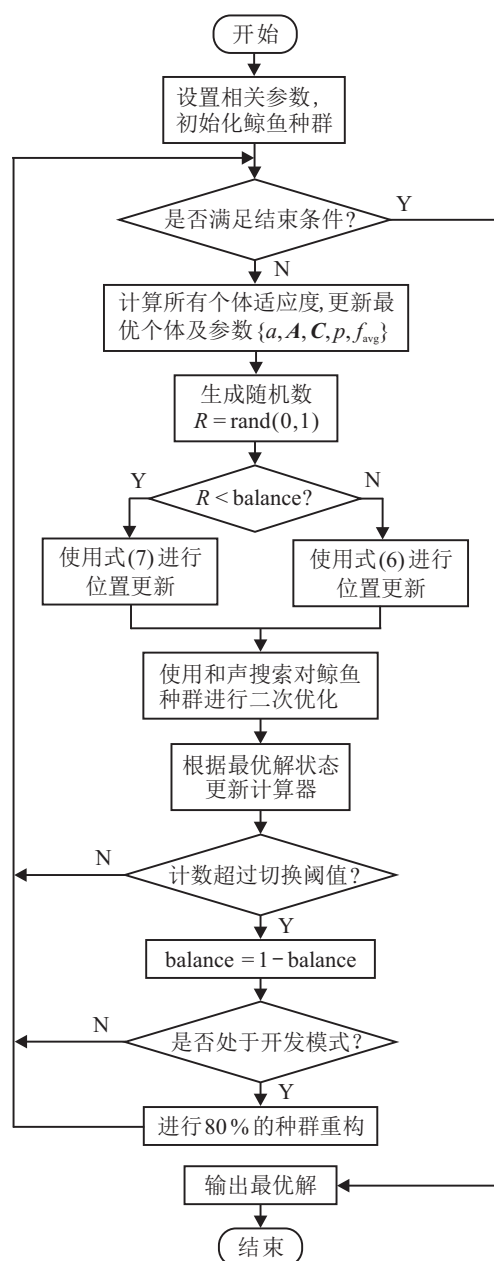


图1 基于和声二次优化的平衡鲸鱼算法流程

step 8: 判断是否到达最大迭代次数, 若达到最大迭代次数, 则执行 step 9; 否则, 转至 step 2.

step 9: 输出当前最优个体及其适应度值.

3 基于平衡鲸鱼算法的路径规划

3.1 环境建模

本文采用栅格法对广域任务空间进行环境建模, 白色栅格代表可通行区域, 黑色栅格代表障碍物区域, 栅格尺度取决于任务空间和障碍物大小^[13]. 为保障无人车的通行安全, 对所有障碍栅格按无人车尺寸的一半进行膨胀, 所占面积不满一个栅格的障碍物按一个栅格计算. 在栅格地图中, 将无人车视为直径远小于栅格单位尺寸的圆, 其每一步的可通行区域由无

人车坐标为中心的八邻域内的白色栅格组成.

3.2 随机路径生成与优化

不同于鲸鱼优化算法在连续域优化方面的应用, 每条鲸鱼对应路径规划任务中的一条可行路径, 路径维度可能不同, 也无法从连续区间直接采样随机生成, 因此在使用鲸鱼算法进行优化之前, 需要随机生成一组可行路径. 本文模仿基本鲸鱼优化过程, 采用启发式随机探索与回溯法相结合的方式实现. 相关定义与随机路径生成步骤如下.

定义1 在栅格地图中, 对于栅格 $n = (i, j)$, $\text{gridmap}(n) = 0$ 表示障碍栅格, $\text{gridmap}(n) = 1$ 表示自由栅格. 栅格 n 的可行域由其八邻域内所有未访问的自由栅格构成, 记为 $\text{accessible}(n)$, $A(n)$ 为栅格 n 可行域包含的栅格数量.

定义2 记起点栅格为 start , 终点栅格为 target , 当前栅格为 current . D 表示起点到终点的欧氏距离, d 表示当前点到终点的欧氏距离, 参数 $R = 2 - 2 \times \frac{D-d}{D}$ 用于控制生成路径的随机化程度.

step 1: 将起点设为当前栅格, 并将当前栅格加入 path 中.

step 2: 设 current 为已访问栅格, 计算 current 的可行域, 若 $\text{target} \in \text{accessible}(\text{current})$, 则将其加入 path , 算法结束.

step 3: 若 $A(n) = 0$, 则转至 step 6 进行回溯.

step 4: 如果参数 $R > 1$, 则对 $\text{accessible}(\text{current})$ 中的栅格 x 按下式计算适应度:

$$\text{fitness}(x) = D - \sqrt{(i_t - i_x)^2 + (j_t - j_x)^2}. \quad (10)$$

其中: (i_t, j_t) 表示终点栅格坐标, (i_x, j_x) 表示可行域内栅格 x 坐标. 用轮盘赌选择方式生成下一栅格, 将其加入 path 中并设为当前栅格, 转至 step 2.

step 5: 若参数 $R \leq 1$, 则选择 $\text{accessible}(\text{current})$ 中最接近终点的栅格作为下一栅格, 将其加入 path 中并设为当前栅格, 转至 step 2.

step 6: 若 $A(n) = 0$ 且 path 中包含的栅格总数大于 1, 则从 path 中删除最后一个栅格, 重复 step 6; 若 $A(n) = 0$ 且 path 包含栅格总数等于 1, 则表明不存在可行路径, 算法结束; 若 $A(n) \neq 0$, 则将 path 中的尾部栅格设为当前栅格, 转至 step 2.

由于路径生成过程的随机化和可行域的设定, 随机路径中将不可避免地出现“回环”或者“凹陷”, 如图 2(a) 所示, 需要对其进一步优化. 本文采用从起点向终点方向扫描, 将路径上相隔最远且中间不存在障碍的栅格直连的方法对随机路径做简单优化, 图 2(a) 路径优化后如图 2(b) 所示.

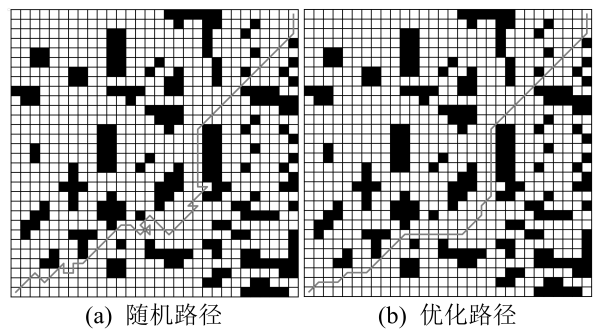


图 2 随机路径与优化路径

3.3 路径移动与微调

基于和声二次优化的平衡鲸鱼算法中涉及到路径的移动与微调操作,处理连续域优化问题时,使用欧氏距离衡量数据间的差异并对数据进行位置更新操作,但这种位置更新方法并不适用于不同维度的路径.因此,本文采用限定区域内重新生成的方式实现路径的移动与微调.当路径A向路径B移动时,以路径A和路径B作为边界约束,在路径A与路径B之间的区域重新生成可行路径代替A向B的移动操作,限定空间如图3(a)中的白色区域所示.当对路径A进行带宽为BW的微调时,以路径A为中线向左右两侧扩张宽度为BW的空间,如图3(b)中的白色区域所示,在该区域内重新生成可行路径代替对路径A的微调操作.

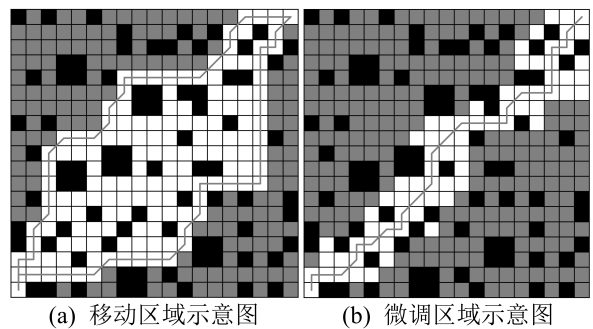


图 3 移动与微调区域示意

4 仿真实验与分析

首先对所提出的两种改进策略进行消融实验,分别验证和声二次优化策略和动态平衡策略的实际性能和有效性;然后,在3种环境下与传统A*算法、改进量子行为粒子群算法(QPSO)^[14]、改进势场蚁群算法^[15]和改进烟花-蚁群混合算法(APFWA-ACA)^[16]进行对比,进一步验证本文算法的性能优势.实验运行环境为:Windows10 64 bit,PyCharm 2018,处理器AMD Ryzen7 2700 U,主频2.2 GHz,内存8 GB.

4.1 参数设定

切换阈值thr控制搜索模式切换的时机,其取值对于动态平衡策略的有效性至关重要.阈值太小将

导致搜索模式切换过快,优化过程不能充分进行并且影响最终解的质量;而阈值太大则会使得算法迟钝,容易陷入局部最优解.两种情况都会明显降低算法的收敛速度.由于在简单环境下动态平衡策略的效果并不明显,本节采用障碍覆盖率为20%的50×50栅格地图对thr参数和算法的其他参数取值进行控制变量实验.为避免偶然性带来的结果偏差,对每种取值进行30次实验并取平均值,结果如表1所示.

表 1 参数优化实验结果

切换阈值	1	2	3	4	5
平均路径长度	78.075	77.431	78.837	78.485	78.368
平均收敛次数	13.3	8.9	13.1	13.2	15.2
平衡参数	0.6	0.7	0.8	0.9	1.0
平均路径长度	82.053	80.318	78.998	82.648	82.654
平均收敛次数	18.7	15.3	9.4	18.2	20.4
种群规模	5	10	15	20	25
平均路径长度	82.246	80.305	79.771	77.665	77.521
平均收敛次数	15.6	12.0	11.6	9.2	9.2

从表1数据可以看出:不同的切换阈值对最终得到的路径长度的影响不大,但平均收敛次数则有较大差异,当切换阈值取值为2时,算法性能最优,增大或减小收敛阈值都会不同程度地增加平均收敛次数;平衡参数balance的值取0.8时能够较好地配合切换阈值平衡探索与开发过程所占比重;种群规模为20时算法性能达到饱和,继续增大种群规模将不会带来性能上的提升.

参数HMCR和PAR参照文献[12]通过一般经验选取,HMCR值为0.8,PAR的值为0.3.微调幅度BW的值不宜过大,否则将对最优个体造成较大改变,本文取BW=2.算法参数设定如表2所示.本文中涉及的对比较算法,其运行参数都按照相关文献中的说明设定.

表 2 实验参数设置

max_iteration	种群规模	thr	balance	HMCR	PAR	BW
50	20	2	0.8	0.8	0.3	2

4.2 消融实验

为了验证和声二次优化和动态平衡策略改进算法的有效性,分别将两种改进方法单独作用于鲸鱼优化算法,并在图4所示环境中进行实验验证.

和声二次优化策略以一定的概率在原始种群外部空间进行采样,并采用精英策略按一定比例保留每代中的优质个体,提升种群的整体质量.图4(a)为使用和声二次优化策略改进的鲸鱼算法在寻优过程

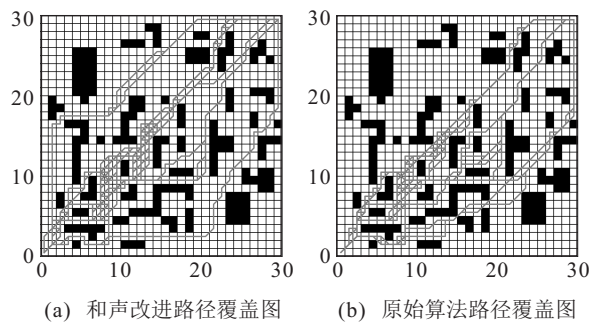


图4 全局搜索性能比较

中的路径覆盖图;图4(b)为原始鲸鱼算法的路径覆盖图,可以看出,原始鲸鱼算法的搜索区域被限制在初始种群所决定的范围内,寻优过程在该区域中重复进行,搜索路径分布较为均匀,对区域边缘的次优区域也进行了多次搜索.相比于原始鲸鱼算法,经过和声二次优化的鲸鱼算法寻优过程中的搜索路径覆盖面更广,具有更好的全局搜索性能,并且种群中的大部分个体在寻优过程中逐渐向最优区域靠拢,避免了对边缘次优区域的过度搜索,在一定程度上加快了收敛速度.

表3数据为图4环境下,和声二次优化策略和动态平衡策略单独以及混合作用于鲸鱼算法的实验结果,取30次实验数据的平均数和最小值.从数据中可以看出,和声二次优化策略算法的平均路径长度小于原始鲸鱼算法,并且收敛更快.路径最优率指在30次实验中,算法得到最优路径的次数所占的比例,原始鲸鱼算法与和声二次优化改进算法虽然能得到相同的路径最小值,但和声二次优化策略的路径最优率高出13.4%,可见该策略能在一定程度上帮助鲸鱼优化算法避开局部最优值.

表3 消融实验数据

算法	原始鲸鱼 算法	和声策略 策略	动态平衡 改进	混合改进
平均路径长度	46.215 6	45.602 8	45.254 1	45.112 7
最短路径长度	45.112 6	45.112 6	45.112 6	45.112 6
平均迭代次数	12.3	8.3	20.4	9.2
路径最优率/%	23.3	36.7	93.3	96.7

采用动态平衡策略改进鲸鱼优化算法在避免局部最优方面有很大改善,从表3中可以看出,使用动态平衡策略改进鲸鱼算法的路径最优率为93.3%.动态平衡策略通过跟踪种群最优解的状态来决定切换搜索模式的时机,从而在算法陷入停滞时切换搜索模式以避免局部最优.

融合了和声二次优化策略和动态平衡策略的改进算法与仅采用动态平衡策略的改进相比,路径最优

率提高了3.4%,表明混合改进算法在避开局部最优与稳定性方面有了进一步提升;与仅采用和声二次优化策略的改进相比,解的整体质量有所提升,平均迭代次数略微增加;与原始鲸鱼算法相比,混合改进算法在平均路径长度、平均迭代次数和路径最优率方面都有较大改善.可见,融合和声二次优化策略与动态平衡策略的改进增加了种群多样性,提高了算法的寻优能力和整体求解质量.

4.3 对比实验分析

本节通过仿真实验比较了基于和声二次优化的平衡鲸鱼算法(HS-WOA)与基本鲸鱼算法(WOA)、传统A*算法、QPSO算法^[14]、改进势场蚁群算法^[15]和APFWA-ACA算法^[16]在路径规划中的性能差异.由于无人车辆的实际行驶环境更为复杂,过多的转向会极大降低行车速度并带来一定的安全隐患.考虑到无人车路径规划对行车速度和安全性的特殊要求,采用路径长度和路径拐点数作为多重指标计算路径适应度,计算公式为 $f(x) = L(x) + \theta \cdot G(x)$.其中: $L(x)$ 表示路径 x 的长度; $G(x)$ 表示路径 x 的拐点数;参数 θ 表示拐点数对路径评价的影响程度,取值范围为 $[0.5, 1.5]$,可根据实际需要取值.当车速较快时应尽量减少路径拐点数以保证车辆平稳行驶, θ 应取较大值;而车速较慢时 θ 的取值可以偏小.本文基于模拟地图进行仿真实验,无实际车速,为了体现算法在多重指标下的优化性能,取 $\theta = 1$.

图5为A*算法、改进QPSO算法、基本WOA算法和HS-WOA算法在仿真环境1中分别进行50次迭代的最优结果,蓝色实线为HS-WOA算法规划路径,橙色三角实线为A*算法规划路径,虚线为基本WOA算法规划路径,点划线为改进QPSO算法规划路径.从图5中可以看出:改进QPSO算法最终没有收敛到最优路径,且存在较多拐点,路径平滑性差,收敛速度慢;A*算法、基本WOA算法和HS-WOA算法在该环境下得到了相同长度的最优路径,但相比于基本WOA,HS-WOA算法得出的规划路径拐点更少,平滑性相对更好,收敛更快.

表4数据由A*算法、改进QPSO、WOA和HS-WOA在环境1中进行50次实验得到.由表4可以看出:在较为简单的环境中,传统A*算法表现稳定,耗时较少;HS-WOA和WOA的最短路径长度、平均路径长度均优于改进QPSO,且改进QPSO的平均规划耗时最长;HS-WOA虽然在平均耗时方面略逊色于WOA,但整体表现稳定,在每次实验中都能得到最优路径.

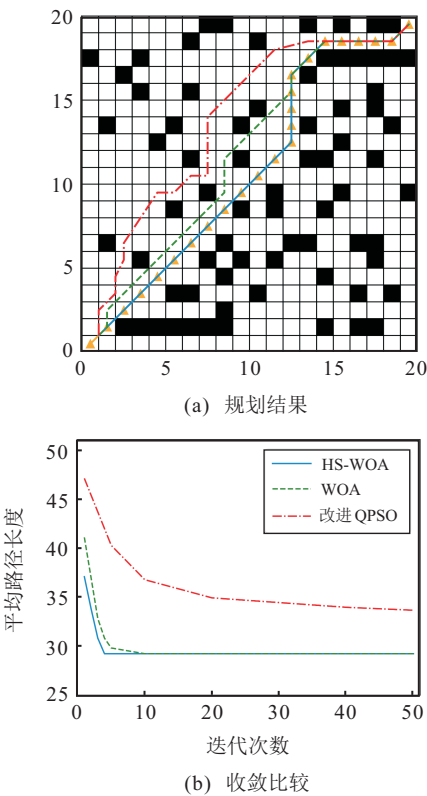


图 5 环境 1 仿真结果

表 4 环境 1 仿真结果对比数据

算法	A* 算法	改进 QPSO ^[14]	WOA	HS-WOA
最短路径长度算法	29.213 2	30.608 8	29.213 2	29.213 2
平均路径长度	29.213 2	32.056 4	29.413 7	29.213 2
平均耗时/s	0.298 7	2.046 1	0.375 2	0.903 7
拐点数	4	13	8	4

图 6 为 A* 算法、改进势场蚁群算法、基本 WOA 算法和 HS-WOA 在环境 2 中的仿真结果,改进势场蚁群算法路径为红色,其余算法路径表示与环境 1 相同. 相比于环境 1,环境 2 的障碍物更多且可通行区域更为狭窄. 从结果中可以看出:在相同尺寸的地图上增加障碍物覆盖率,A* 算法虽然仍能得到与 HS-WOA 算法长度相当的路径,但在路径拐点数方面已显现出差异,A* 算法在寻路时仅考虑长度指标,并不能满足路径在其他方面的特殊要求;相比于文献[15]中的改进势场蚁群算法,WOA 和 HS-WOA 算法得到的路径更短,拐点数更少;HS-WOA 算法在该环境中得到的路径尽可能地减少了无人车的转向操作,在路径长度相同的条件下,提高了行车速度和安全性.

环境 2 所比较的算法均在前 30 次迭代中收敛,故仅显示前 30 次迭代的结果. 从图 6(b)中可以看出:改进势场蚁群算法收敛最快,但得到的结果并不是全局最优解;HS-WOA 与 WOA 最终得到的路径长度相等,但 HS-WOA 的收敛速度比 WOA 更快.

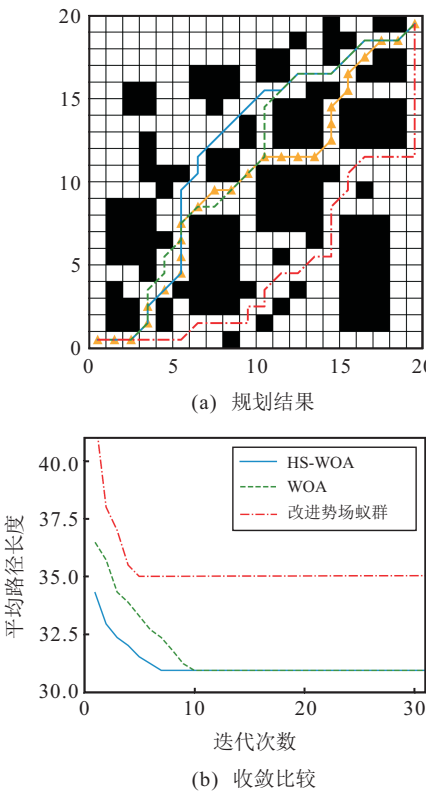


图 6 环境 2 仿真结果

表 5 给出了各算法在环境 2 中的仿真结果.

表 5 环境 2 仿真结果对比数据

算法	A* 算法	改进势场蚁群 ^[15]	WOA	HS-WOA
最短路径长度	30.970 6	35.071 1	30.970 6	30.970 6
拐点数	15	15	15	13
平均迭代次数	—	7.7	10.7	7.9

图 7 中红色路径为 APFWA-ACA 算法的规划结果,橙色路径为 A* 算法规划结果,其余算法路径表示与环境 1 相同. APFWA-ACA 算法在 50 次迭代内的平均路径长度远高于 WOA 和 HS-WOA,且算法后期出现了轻微的震荡现象,在环境 3 中得到的最短路径长度为 77.15,与平均路径长度相差较大,这也从侧面反映出 APFWA-ACA 算法在该环境中缺乏稳定性,最终结果的最优覆盖率较低. 基本 WOA 算法和 HS-WOA 算法与 APFWA-ACA 相比都具有更好的稳定性和收敛速度,在环境 3 中 HS-WOA 相对于 WOA 的优势也更加明显. 由于 HS-WOA 的二次优化机制,算法得到的平均路径长度优于 WOA,收敛速度更快,同时,动态平衡策略能使 HS-WOA 有效避开局部最优解.

从表 6 数据可以进一步看出,在扩大地图尺寸和障碍覆盖率之后,A* 算法得到的路径在拐点数方面与 HS-WOA 算法相差较大,不能满足无人车路径规划的特殊需求. 对比其他 3 种算法,HS-WOA 表

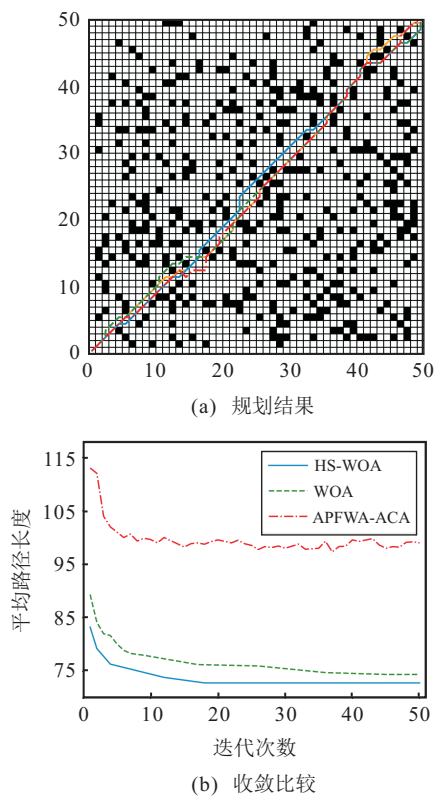


图7 环境3仿真结果

表6 环境3仿真结果对比数据

算法	A*算法	APFWA-ACA ^[16]	WOA	HS-WOA
最短路径长度	71.973 1	77.150 0	73.325 9	72.104 1
平均路径长度	71.973 1	99.051 3	74.396 9	72.811 2
平均耗时/s	0.767	364.320	1.561	2.727
拐点数	24	22	24	18

现最为稳定,且规划出的路径在长度和拐点数方面都有明显优势.在算法平均耗时方面:A*算法仅进行单次规划,速度最快;而APFWA-ACA算法在环境3中50次迭代的平均耗时远远高于HS-WOA算法,实用性较差;因为引入了二次优化和动态平衡策略,HS-WOA耗时略高于基本WOA,但解的整体质量更好.除A*算法外的其他3种智能优化算法平均耗时均以50次迭代的总时间进行计算,考虑到收敛速度的差异,HS-WOA在迭代到收敛时花费的时间更少,因此,HS-WOA在整体迭代耗时中表现的劣势是可以接受的.

在上述3种模拟环境中,传统A*算法在路径长度和平均耗时方面都表现出了较好的性能.但事实上,A*算法所表现的优势只适用于二维栅格地图且仅考虑路径长度的情况.当地图的维度增加(如考虑道路的高度信息),或者对路径的评判指标增加(如考虑拐点数),都会在很大程度上影响A*算法的寻路性能,特别是维度的增加会使A*算法的计算消耗呈指

数级增长.

4.4 算法实用性分析

本文改进鲸鱼优化算法主要应用于无人车辆全局路径规划中,相比于A*算法等传统路径规划算法,本文算法在使用适应度对路径进行评价时,可综合考虑无人车路径所需的特殊因素,如拐点数等,而不是以路径长度作为唯一评价指标,最终生成更符合无人车规划需求的路径.从上述仿真实验和算法对比结果中可以看出,本文改进算法能够更好地满足无人车路径规划的特殊要求.

此外,A*算法在寻路时总是进行单次规划并得到唯一的最优结果,若考虑在实时规划过程中遇到动态障碍物或其他因素,如路况因素,则导致原规划路径不可行的情况,A*算法需要以当前坐标为起点,重新规划一条新的可行路径.而本文所提出的算法更为灵活,在寻优过程中能够覆盖多个最优结果或次优结果,将这些结果作为候选路径信息存储,可在环境发生变化时为无人车切换路径提供有效参考,从而避免二次规划带来的额外消耗,提高行车效率.

5 结 论

针对地面无人车辆的全局路径规划问题,本文提出了一种基于和声二次优化的平衡鲸鱼算法.将和声搜索策略和动态平衡策略与基本鲸鱼算法融合,并根据鲸鱼个体的质量自适应地选择优化行为,增强了鲸鱼算法的寻优能力和求解质量;配合动态平衡和种群重构机制避免算法陷入局部最优;充分探索任务空间信息,多样化的改进策略同时增加了算法的稳定性.最后,分别对两种改进策略的有效性进行了消融实验验证,并基于3种不同的栅格环境进行了仿真实验,从最短路径长度、平均路径长度、算法平均耗时和拐点数等方面比较了本文算法与传统A*算法以及其他改进智能优化算法的优劣.结果表明,基于和声二次优化的平衡鲸鱼算法在最优性、稳定性和收敛速度方面均表现出明显优势,并且具有较好的实用性能.本文的模拟环境为静态障碍环境,下一步的工作重心是平衡鲸鱼算法在动态环境下的路径规划应用及相关理论的研究.

参考文献(References)

[1] 李宝磊,吕丹桔,张钦虎,等.基于多元优化算法的路径规划[J].电子学报,2016,44(9):2242-2247.
(Li B L, Lv D J, Zhang Q H, et al. A path planner based on multivariant optimization algorithm[J]. Acta Electronica Sinica, 2016, 44(9): 2242-2247.)

- [2] Shi K, Denny J, Amato N M. Spark PRM: Using RRTs within PRMs to efficiently explore narrow passages[C]. 2014 IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, 2014: 4659-4666.
- [3] Mashayekhi R, Idris M Y I, Anisi M H, et al. Informed RRT*-connect: An asymptotically optimal single-query path planning method[J]. IEEE Access, 2020, 8: 19842-19852.
- [4] Zaid Tahir, Ahmed H Qureshi, Yasar A, et al. Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments[J]. Robotics and Autonomous Systems, 2018, 108: 13-27.
- [5] Paniagua A H, Bandera J P, Quintanilla M R, et al. Quad-RRT: A real-time GPU-based global path planner in large-scale real environments[J]. Expert Systems with Applications, 2018, 99: 141-154.
- [6] Luo Q, Wang H, Zheng Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Neural Computing and Applications, 2019(1): 1-12.
- [7] Fatin H Ajeil, Ibraheem Kasim Ibraheem, Mouayad A Sahib, et al. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm[J]. Applied Soft Computing, 2020, 89: 106076.
- [8] Seyedali Mirjalili, Andrew Lewis. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [9] 黄清宝, 李俊兴, 宋春宁, 等. 基于余弦控制因子和多项式变异的鲸鱼优化算法[J]. 控制与决策, 2020, 35(3): 559-568.
(Huang Q B, Li J X, Song C N, et al. Whale optimization algorithm based on cosine control factor and polynomial mutation[J]. Control and Decision, 2020, 35(3): 559-568.)
- [10] Gaganpreet Kaur, Sankalop Arora. Chaotic whale optimization algorithm[J]. Journal of Computational Design & Engineering, 2018, 5(3): 275-284.
- [11] Liu Min, Yao Xifan, Li Yongxiang. Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems[J]. Applied Soft Computing, 2020, 87: 105954.
- [12] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: Harmony search[J]. Simulation, 2001, 76(2): 60-68.
- [13] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, et al. Heuristic approaches in robot path planning: A survey[J]. Robotics and Autonomous Systems, 2016, 86: 13-28.
- [14] 胡章芳, 孙林, 张毅, 等. 一种基于改进QPSO的机器人路径规划算法[J]. 计算机工程, 2019, 45(4): 281-287.
(Hu Z F, Sun L, Zhang Y, et al. A robot path planning algorithm based on improved QPSO[J]. Computer Engineering, 2019, 45(4): 281-287.)
- [15] 刘建华, 杨建国, 刘华平, 等. 基于势场蚁群算法的移动机器人全局路径规划方法[J]. 农业机械学报, 2015, 46(9): 18-27.
(Liu J H, Yang J G, Liu H P, et al. Robot global path planning based on ant colony optimization with artificial potential field[J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(9): 18-27.)
- [16] 张玮, 马焱, 赵捍东, 等. 基于改进烟花-蚁群混合算法的智能移动体避障路径规划[J]. 控制与决策, 2019, 34(2): 335-343.
(Zhang W, Ma Y, Zhao H D, et al. Obstacle avoidance path planning of intelligent mobile based on improved fireworks-ant colony hybrid algorithm[J]. Control and Decision, 2019, 34(2): 335-343.)

作者简介

蔡雨岑(1996—), 男, 硕士生, 从事智能机器人路径规划和导航的研究, E-mail: solaalus@163.com;

杜鹏桢(1982—), 男, 讲师, 博士, 从事机器人及智能计算等研究, E-mail: h.k@foxmail.com.

(责任编辑: 李君玲)