引用:韩玉艳,安俊宇,杨晓宇,等. 多代理强化学习驱动遗传算法求解软时间窗电动车路径规划问题[J]. 聊城大学学报(自然科学版),2026,39(1):32-43.

Citation: HAN Yuyan, AN Junyu, YANG Xiaoyu, WANG Yuting, LI Huan, TIAN Xinru. Multi-agent reinforcement learning-driven genetic algorithm for solving the soft time window electric vehicle routing problem[J]. Journal of Liaocheng University (Natural Science Edition), 2026, 39(1): 32-43.

文章编号 1672-6634(2026)01-0032-12 **DOI** 10.19728/j.issn1672-6634.2025040010

多代理强化学习驱动遗传算法求解软时间窗电动车路径规划问题

韩玉艳,安俊宇,杨晓宇,王玉亭,李寰,田鑫茹

(聊城大学 计算机学院,山东 聊城 252059)

摘要 针对考虑电动车能耗、订单时间窗和车辆负载等约束条件的路径优化问题,首先,构建了以路径成本最小化为核心优化目标的混合整数规划模型;然后,根据问题特性,提出多代理强化学习驱动的遗传算法(Multi-Agent Reinforcement Learning-Driven Genetic Algorithm,MRLGA)。在 MRLGA 算法中,运用多代理智能体动态调整遗传算法的变异交叉概率以及选择交叉操作,以提升算法的搜索效率;通过引入香农多样性指数衡量种群多样性,避免算法早熟收敛,有效维护种群的多样性;采用 2-opt 搜索算法增强局部搜索能力,同时引入离散莱维飞行策略提高全局搜索能力,实现对车辆路径的高效规划。最后,通过 120 个测试算例展开实验,结果表明所提 MRLGA 算法在满足复杂约束条件下,能有效降低路径成本,验证了算法的有效性和可行性。

关键词 车辆路径问题;遗传算法;香农多样性指数;强化学习

中图分类号 U492.22;TP18 **文献标志码** A **开放科学(资源服务)标识码(OSID)**

本文电子版文后含有增强出版内容。

Multi-agent reinforcement learning-driven genetic algorithm for solving the soft time window electric vehicle routing problem

HAN Yuyan, AN Junyu, YANG Xiaoyu, WANG Yuting, LI Huan, TIAN Xinru

(School of Computer Science, Liaocheng University, Liaocheng 252059, China)

Abstract For the path optimization problem considering constraints such as the energy consumption of electric vehicles, order time windows, and vehicle loads, a mixed-integer programming model with the minimization of path cost as the core optimization objective is first constructed. Then, based on the problem characteristics, a multi-agent reinforcement learning-driven genetic algorithm (MRLGA) is proposed for solving it. In the MRLGA, the multi-agent reinforcement learning method is used to dynamically optimize the mutation and crossover probabilities in the genetic algorithm and intelligently proxy the selection and

收稿日期:2025-04-15

基金项目:国家自然科学基金项目(61973203, 61803192, 62106073, 61966012);山东省自然科学基金项目(ZR2023MF022, ZR2024MF112);聊城大学光岳青年创新团队项目(LCUGYTD2022-03)资助

通信作者:韩玉艳,女,汉族,博士,副教授,研究方向:智能优化调度,E-mail:hanyuyan@lcu-cs.com。

中国知网 https://www.cnki.net

crossover operations to enhance the algorithm's search efficiency. By introducing the Shannon diversity index to measure population diversity, premature convergence is avoided, and effective maintenance of population diversity is achieved. The 2-opt search algorithm is adopted to enhance local search capabilities, and the discrete Lévy flight strategy is introduced to improve global search capabilities, achieving efficient vehicle path planning. Finally, experiments are conducted with 120 test cases, and the results show that the proposed MRLGA algorithm can effectively reduce path costs under complex constraints, verifying the effectiveness and feasibility of the algorithm.

Keywords　vehicle routing problem; genetic algorithm; shannon diversity index; reinforcement learning

0　引言

车辆路径问题(Vehicle Routing Problem, VRP)作为物流管理领域的核心研究内容，其本质在于在满足一系列约束条件的前提下，通过对车辆的合理调度，实现服务成本的最小化。该问题由 G. Dantzig 和 J. Ramser 于 1959 年首次提出，历经多年发展，已成为运筹学、管理学、计算机科学、组合数学以及图论等多学科交叉的研究热点[1]。同时随着全球气候变化问题日益突出，由化石燃料燃烧所产生的大量温室气体排放被公认为主要驱动因素[2]。在此背景下，新能源交通工具不断涌现，其中电动汽车凭借其能够显著降低碳排放、改善空气质量的优势，成为新能源交通工具中的重要类别[3]，因此现在众多学者展开了对于电动车车辆路径问题(Electric Vehicle-Routing Problem, EVRP)的研究。

当前，众多学者针对 VRP 及相关拓展问题展开了深入研究[4]，一系列高性能的启发式算法不断涌现，包括遗传算法(Genetic Algorithm, GA)[5]、粒子群优化算法(Particle Swarm Optimization, PSO)[6]、蚁群优化算法(Ant Colony Optimization, ACO)[7]以及禁忌搜索算法(Tabu Search, TS)[8]等。例如，Liu 等人提出利用 ACO 解决冷链电动汽车路径问题，并结合四种局部优化算子构建 HACO 算法，有效提升了算法的优化效率[9]。Wang 等人运用深度学习方法，借助动态图转换模型(Dynamic Graph Transformer Model, DGTM)和动态位置编码(Dynamic Positional Encodin, DPE)对模型进行训练求解[10]。He 等人在 Liu 等人的研究基础上，针对考虑软时间窗下冷链物流多温共配的电动汽车路径优化问题展开研究，结合蚁群算法与 2-opt 优化方法，显著增强了局部搜索能力[11]。

然而，现有方法在实际应用中仍然面临诸多挑战。一方面，部分算法在处理大规模问题时，计算复杂度较高，导致运行时间过长，难以满足实时或大规模场景的应用需求；另一方面，一些方法过度依赖大量训练数据和计算资源，且在不同环境或条件变化下的泛化能力欠佳，难以适应动态变化的运输需求。特别在参数选择方面缺乏自适应性和动态调整的能力，从而限制了算法的有效性。

上述方法中，遗传算法作为一种模拟生物进化过程的随机搜索算法，由于其全局搜索能力和对复杂问题的适应性，被广泛应用于各类组合优化问题，包含 VRP 的求解，如 Bai 等人将遗传算法与退火模拟算法结合求解可容忍时间窗的 VRP 问题[12]。Chen 等人基于强化学习提出了一种改进的遗传算法对于柔性作业车间调度问题进行了求解[13]。但是，传统遗传算法在求解 VRP 时存在一定局限性，其中较为突出的问题是算法参数难以动态调整。如，交叉概率(P_c)和变异概率(P_m)通常在算法运行前就被固定设定，或者在整个进化过程中仅进行全局统一的调整，无法根据种群进化的实时状态和不同子群体的特性进行动态变化。这导致算法在面对复杂多变的 VRP 实例时，容易陷入局部最优解，难以在全局搜索和局部搜索之间找到良好的平衡。

为应对上述问题，引入强化学习(Reinforcement Learning, RL)为遗传算法参数的动态调整提供了新的策略[14]。强化学习是一种通过智能体与环境进行交互并根据奖励信号来学习最优行为策略的机器学习方法。受 Chen 等人利用强化学习代理交叉概率和变异概率等参数的启发[13]，同时考虑到生物种间竞争所具有的多样性特点，即同一生物种群在不同环境下进化速率和方向存在差异，本文在此基础上设计多种交叉和选择方案，并通过强化学习进行决策控制代理，以提升局部搜索和全局搜索能力。本文贡献主要如下：(1)传统的 GA-RL 方法通常在全局层面统一调整 GA 的关键参数，缺乏对种群内部不同子群体特性的细致响应。本文提出为每个子群体独立调整选择策略和交叉方法的机制，旨在根据子群体的特定需求和进化状态，

中国知网　https://www.cnki.net

灵活地优化各自的遗传操作,从而提高算法的整体性能和对复杂问题的适应性;(2) 提出使用汉明距离和层次聚类对物种进行划分,然后通过香农多样性指数描述种群的多样性。这种方法能够更准确地刻画种群在进化过程中的多样性变化,为遗传算法的参数调整和进化策略优化提供更有效的指导;(3) 通过实验验证MRLGA 方法在 EVRP 问题上的可行性,并通过对照实验验证其在解决该问题时相较于传统方法的性能优势,为该方法在实际物流配送中的应用提供理论支持和实践依据。

1 带时间窗的电动车路径问题

1.1 问题描述

带时间窗的电动车路径问题的场景为:给定 m 辆额定载重为 Q 的电动车辆,其初始位置均为 v_0(通常为货仓)。这些车辆的任务是向一系列客户提供货物运输服务,完成任务后需返回起始点,目标是在满足所有约束条件的前提下,使整个服务过程的总成本即电力消耗成本,因时间服务有误产生的惩罚成本以及换电站的换电成本达到最低。本文考虑的约束如下:(1) 针对每一个客户,均明确了专属时间窗,仅当车辆在该时间窗区间内抵达客户位置并完成服务行为,方可认定为正常服务,如果超出范围服务则需要付出一定的惩罚成本。(2) 电动车辆在执行运输任务的行驶进程中,可依据自身电量消耗情况,自主选择在换电站实施电池更换作业,每辆车通过每一个换电站的次数也被严格限定为一次。(3) 车辆行驶的过程中不可以超出行驶距离,也不可以超出服务载重,并且车辆始终匀速行驶。(4) 每个客户只能服务一次。(5) 不考虑车辆行驶过程中出现车辆故障以及发生交通事故。(6) 规定所有车辆均投入使用并均进行服务。

1.2 数学模型

本文涉及到的所有符号,变量及相应解释见表 1。

表 1 符号,决策变量及相应解释

Tab. 1 Symbols, decision variables and their corresponding descriptions

变量	含义	变量	含义	变量	含义
e_i	客户最早到达时间	m	车辆个数	c_{dis}	单位路径消耗
l_i	客户最晚到达时间	n	客户的数量	g_i	客户 i 的需求
c_{epu}	早到惩罚	Q	车辆的承重	V	用户的集合
c_{lpu}	晚到惩罚	$n_{k,i}^{\mathrm{ele}}$	换电站 i 给 k 车换电次数	K_{Car}	车辆集合
P	换电站的个数	$d_{i,j}$	从客户 i 到 j 的距离	h_i	i 客户的服务时间
F	续航里程	x_i	客户所在横坐标	E	换电站集合
v_c	车辆行驶速度	y_i	客户所在纵坐标	c_i^{ele}	换电站 i 的换电成本
$t_{i,k}$	车辆 k 到达客户 i 的时间	C_{DIS}	路径消耗总成本	$y_{i,k}$	二进制决策变量,客户 i 由 k 服务为 1 否则为 0
C_{T}	总成本	C_{SC}	换电站换电总成本	$z_{i,k}$	二进制决策变量,车辆 k 使用换电站 i
C_{TP}	时间窗惩罚成本	$x_{i,j,k}$	二进制决策变量,k 车从客户 i 出从客户 j 点进入为 1,否则为 0		

目标:

$$\mathrm{Min}C_{\mathrm{T}}, \tag{1}$$

$$C_{\mathrm{T}}=C_{\mathrm{TP}}+C_{\mathrm{DIS}}+C_{\mathrm{SC}}, \tag{2}$$

$$C_{\mathrm{DIS}}=\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m}(c_{\mathrm{dis}}\times d_{i,j}\times x_{i,j,k}), \tag{3}$$

$$C_{\mathrm{TP}}=\sum_{i=0}^{n}\sum_{k=1}^{m}(c_{\mathrm{lpu}}\times\max(t_{i,k}-l_i,0)+c_{\mathrm{epu}}\times\max(e_i-t_{i,k},0))\times y_{i,k}, \tag{4}$$

$$C_{\mathrm{SC}}=\sum_{i=1}^{P}\sum_{k=1}^{m}(z_{i,k}\times c_i^{\mathrm{ele}})。 \tag{5}$$

中国知网 https://www.cnki.net

约束:

$$d_{i,j}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2},\forall i,j\in V, \tag{6}$$

$$\sum_{i=0}^{n} g_i\times y_{i,k}\leqslant Q,\forall k\in K_{\text{car}}, \tag{7}$$

$$\sum_{i=0}^{n} x_{i,j,k}=y_{i,k},\forall k\in K_{\text{car}}\ \forall j\in V, \tag{8}$$

$$\sum_{j=0}^{n} x_{i,j,k}=y_{i,k},\forall i\in V, \tag{9}$$

$$\sum_{k=1}^{m} y_{i,k}=1,\forall i\in V, \tag{10}$$

$$x_{i,j,k}\in\{0,1\},\forall k\in K_{\text{car}},\forall i,j\in V, \tag{11}$$

$$t_{i,k}+h_i+\frac{d_{i,j}}{v_c}-t_{j,k}\leqslant M-M\times x_{i,j,k},\forall k\in K_{\text{car}}\ \forall j,i\in V,M\to\infty, \tag{12}$$

$$t_{i,k}-t_{j,k}+M\times x_{i,j,k}\leqslant M-1,\forall k\in K_{\text{car}},\forall i,j\in V,i\neq j,M\to\infty, \tag{13}$$

$$\sum_{j=0}^{n} x_{i,j,k}\geqslant z_{i,k},\forall i\in E,\forall k\in K_{\text{car}}, \tag{14}$$

$$n_{k,i}^{\text{ele}}\in\{0,1\},\forall k\in K_{\text{car}},\forall i\in E, \tag{15}$$

$$\sum_{i=0}^{n} y_{i,k}\geqslant 1,\forall k\in K_{\text{car}}。 \tag{16}$$

式(1)考虑了因服务过早或过晚产生的惩罚成本以及换电站操作所产生的换电成本。式(2)~(5)分别为总成本、路径消耗成本、时间窗惩罚成本、换电站使用成本。式(6)描述了直角坐标系下的距离计算方法,式(7)约束了每辆车的载重限制,确保不超过承重上限 Q。式(8)、(9)确保每辆车的路径上车辆数量为 1,式(10)规定每个客户的需求只能由一辆车完成,并且只能被服务一次,式(11)限定了车辆的取值范围。式(12)确保了时间传播的正确性,考虑了车辆在完成服务和行驶时间后到达下一个节点,使用大常数 M 作为松弛变量,避免未选择路径的干扰。式(13)通过路径顺序约束防止次环的出现,确保车辆按顺序依次访问节点。式(14)规定,如果车辆在换电站进行换电操作,必须经过该换电站,确保换电操作的顺序和路径一致性。式(15)保证每个换电站仅能使用一次,且如果车辆未在换电站充电,则设为 1,否则设为 0。式(16)保证了所有的车辆均投入使用并且至少服务一个客户。

2　多代理强化学习改进遗传算法

传统遗传算法在求解软时间窗的 EVRP 问题时,参数通常在算法运行前就被固定设定,无法根据种群进化的实时状态和不同子群体的特性进行动态变化。这导致算法在面对复杂多变的 VRP 实例时,容易陷入局部最优解,难以在全局搜索和局部搜索之间找到良好的平衡。为了解决上述问题,本研究融入了多代理强化学习方法,提出了 MRLGA,对参数值进行动态的调整以及交叉策略的动态选择,以提升遗传算法的局部搜索和全局搜索能力。如算法 1 所示,第 1 行至第 3 行主要完成基本参数的设定以及代理的初始化工作。从第 7 行至第 11 行,算法开展选择交叉操作,此过程根据特定的选择策略挑选个体,并通过交叉操作生成新的个体,以增加种群的多样性和探索更优解的可能性。在完成选择交叉操作之后,算法进入变异操作阶段。本研究采用了三种不同的变异方式,分别为交换变异(Swap Mutation,详见支撑材料图 4)、反转变异(Reverse Mutation,详见支撑材料图 5)和插入变异(Insert Mutation,详见支撑材料图 6)。交换变异通过随机交换染色体上两个基因的位置来产生新个体;反转变异则是随机选取染色体上的一段基因序列并将其反转;插入变异是随机选择一个基因并将其插入到染色体的另一个位置。变异操作完成后,算法采用离散莱维飞行策略和 2-opt 策略对现有解进行局部调整来优化路径,减少路径的总长度或成本。最后,第 15 行至第 21 行根据当前种群的状态和优化结果,更新强化学习代理的策略,调整算法的参数,从而不断提升算法的性能和求解质量(程序流程图详见支撑材料图 7)。

2.1　编码

为有效求解带时间窗的电动车路径问题,本文采用离散整数编码方式来表征个体,以构建遗传算法的基

中国知网　https://www.cnki.net

础框架[15]。染色体长度设定为 $n+P+m+1$，其中，n 代表客户数量，P 表示换电站数量，m 为车辆数量，而最后的“1”则对应起始点(通常为货仓)。这一长度设定涵盖了问题中的所有关键元素。具体编码过程如下：首先在区间[1，$n+P$]内生成一个全排列序列。涵盖所有客户和换电站的编号，为搜索全局最优解提供了广阔的解空间。随后，在生成的全排列序列中插入 $m+1$ 个“0”。这些‘0’用于表示每辆车从起始点出发并返回起始点，其位置分布在染色体的开头、结尾以及中间部分。这种分布方式能够准确模拟车辆从起始点出发，依次访问客户和换电站，最后返回起始点的实际路径规划过程。通过这种编码方式，染色体与问题解之间建立了一一对应的有效映射关系，使得遗传算法能够在染色体空间内进行搜索，从而找到对应问题的最优解。图 1 直观展示了一条染色体结构(可行解)。从图 1 中可以清晰地看出，一共有三辆车参与运输任务，其各自的行驶路径通过染色体上的数字序列得到了有效表示：0→3→5→0，0→1→4→2→0 和 0→7→6→8→9→0。

算法 1 MRLGA(P_S，G)

Input：种群大小(P_S)，总迭代次数(G)

01：P←Initialpopulation()//全随机的方式生成若干客户序列并通过插 0 的方式构成初代种群

02：P_c&P_mAgent←创建交叉概率调整代理 Selection&CrossoverAgent ←创建选择和交叉代理

03：初始化强化学习代理的 Q 表、alpha、gamma、epsilon 等参数

04：基于极大惩罚的方法计算种群每个个体 g 的适应度 $f(g)$ //详见 2.2 节描述

05：**for** generation←1 to G **do**

06： P_h←GetState (P_h，f_b，f_a，D) // 详见 2.3.1 节描述

07： SelStrategy←Selection&CrossoverAgent. choose_action(P_h) // 详见 2.3.2 节描述

08： Selection P←Select (P，S_{sele}) //根据选择策略 SelStrategy 执行选择操作

09： CroMethod←Selection&CrossoverAgent. crossover_action(P_h)// 详见 2.3.2 节描述

10： Child P←Select(Selection P，S_{cros}) //根据交叉策略 CroStrategy 执行交叉操作 1

11： P←mergeGenes(P, Child P), MutationMethod←$P_c$choose$_{action}$ //通过随机概率选择变异方法

12： P←Mutation(P，MutationMethod)，P←LevyFlight(P) //使用 Levy Flight 对基因进行全局优化，详见 2.4 节描述

13： P←TwoOpt(P) //使用 2-opt 算法进行局部搜索优化，详见 2.5 节描述

14： 基于极大惩罚的方法计算种群每个个体 g 的适应度 $f(g)$ 以及f_a，f_b

15： Diversity←ShonnoCalculate$P$ //计算香农多样性指数 详见 2.3.3 节描述

16： $R=0.5\times\Delta f_b+0.3\times\Delta f_a+0.2\times\Delta D-P_{pun}$ //计算奖励

17： P_h←getNextStage(ΔD，Δf_a，Δf_b，P) //确定下一个阶段

18： Selection&CrossoverAgent. Q←Selection&CrossoverAgent. learn(P_h，R')

19： P_c&P_mAgent. Q←P_c&P_mAgent. learn(P_h, R') //更新Q表 详见 2.3 节描述

20： P_c←P_c&P_mAgent. P_c_action(P_h)//详见 2.3.2 节描述

21： $P_m$ut←P_c&P_mAgent. $P_m$ut_action (P_h)，ArchiveSet←UpdateArchiveSet(ArchiveSet，P)//存档集更新

22：Endfor

Output：存档集(Archive Set)

2.2 基于极大惩罚的适应度计算

从图 1 所示的已生成的多个解的情况分析可知，当前解空间中对于载重 Q 和续航 F 尚未施加有效的约束机制。为保障算法迭代生成的解能够契合实际应用场景中的载重与续航限制条件，本研究引入一个极大值 M。当某一解所对应的车辆载重超过额定载重 Q，或者车辆的续航里程超出最大续航 F 时，在计算该解的适应度值 f 过程中(如式(17)所示)，将对其赋予一个与 M 相等的惩罚值。鉴于 M 取值极大，在遗传算法

中国知网 https://www.cnki.net

基于适应度值进行的选择、交叉和变异等操作过程中，携带此类违规情况的解会因其极低的适应度值，在自然选择的优胜劣汰机制下被直接排除，从而保证算法搜索空间始终聚焦于满足载重和续航限制的可行解区域，促使算法在迭代过程中逐步逼近问题的全局最优解。由式(17)可知，目标函数与适应度函数呈反比关系，所以对于适应度函数而言，函数值越大，说明路径的总成本越低，从而适应度越大，在选择阶段越容易被选中，

$$f=\frac{1}{C_{\mathrm{T}}+M\times\max\left(\sum_{i=0}^{n}(g_i\times y_{i,k})-Q,0\right)+M\times\max\left(0,\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m}(d_{i,j}\times x_{i,j,k})-F\right)}。\tag{17}$$

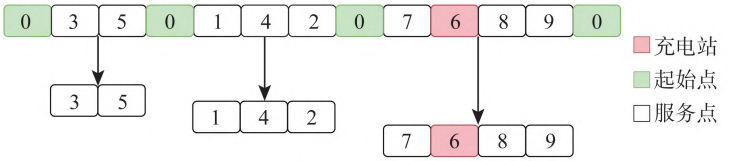

图 1　路径编码构成图

Fig. 1　Path encoding composition diagram

2.3　多代理强化学习驱动的交叉和变异操作

多代理强化学习驱动的交叉和变异操作是一种将多代理强化学习技术融入遗传算法的创新方法，核心原理在于通过智能体与环境的持续交互，依据环境反馈的奖励信号，不断学习并优化自身行为策略，以实现累积奖励的最大化。在本研究中，为了实现对遗传算法交叉和变异操作的精细化控制，引入了两个关键代理，分别为引入了 P_{c}&P_{m}Agent 和 Selection&CrossoverAgent，其中前者负责动态调整遗传算法的交叉概率和变异概率参数，后者则承担决定遗传算法交叉操作和选择操作具体策略的职责。整体系统架构涵盖以下四个关键环节：(1) 定义状态，精确确定当前遗传算法运行过程中的状态信息，包括进化代数、种群最佳适应度值、平均适应度值以及种群多样性指标等；(2) 定义动作，基于所确定的状态，Selection&CrossoverAgent 依据特定决策规则从预设策略集合中选择具体交叉和选择方法，同时 P_{c}&P_{m}Agent 动态调整交叉概率和变异概率参数值；(3) 奖励机制，根据遗传算法在当前迭代过程中的性能变化，给予一定的奖励与惩罚，指导强化学习代理优化动作选择；(4) 学习与更新，Q-Learning 算法对强化学习代理的策略进行迭代更新，代理根据当前状态和执行动作所获奖励，更新对不同状态-动作对的价值估计(Q 值)，逐步学习在不同状态下的最优动作选择策略。更新公式为

$$Q(s,a)\leftarrow Q(s,a)+\alpha\left[r+\gamma\max_{a'}Q(s',a')-Q(s,a)\right],\tag{18}$$

式中 $Q(s,a)$是在状态 s 下采取动作 a 的价值，α 是学习率即控制新信息对旧信息的影响程度。γ 是折扣因子用于衡量未来奖励的当前价值。r 为即时奖励。s' 为执行动作后的下一个状态。$\max_{a'}Q(s',a')$ 为下一个状态 s'下所有可能动作的最大 Q 值。

2.3.1　四元组状态集定义。状态是强化学习代理决策基础，反应遗传算法当前的状态，状态集定义为 $S=(P_h,f_b,f_a,D)$，其中 P_h 即代数阶段，可以根据当前代数 g 确定为前中后期，公式如

$$P_h=\begin{cases}\text{early},\text{if } g<0.33\times G,\\ \text{middle},\text{if } 0.33\times G\leqslant g\leqslant 0.7\times G,\\ \text{late},\text{if } g\geqslant 0.7\times G,\end{cases}\tag{19}$$

式中 G 为总迭代次数。f_b 为最佳适应度，表示的当前种群中适应度最高的个体的适应度值，f_a 为平均适应度，公式如

$$f_b=\max_{i=1}^{N}f_i,\tag{20}$$

$$f_a=\frac{1}{N}\sum_{i=1}^{N}f_i,\tag{21}$$

式中 f_i 为第 $i(i=1,2,\cdots,N)$ 个个体的适应度值。D 表示种群的多样性，这里引入香农多样性指数 H'[16] 对于种群的多样性进行衡量，具体内容详见 2.3.3。

2.3.2　动作空间的设计和动作选择的决策。动作定义了强化学习代理可以执行的操作，动作集定义为

中国知网　https://www.cnki.net

$A_{\text{action}}=(\Delta P_{\text{c}},\Delta P_{\text{m}})$,$B_{\text{action}}=(S_{\text{sele}},S_{\text{cros}})$。$A_{\text{action}}$ 动作集中包含 2 个参数,ΔP_{c} 是交叉概率的调整操作,可以是增加、保持或减少。ΔP_{m} 是变异概率的调整操作,同样可以是增加、保持或减少。基于调整操作,交叉和变异概率的动态设计如

$$P_{\text{c}}^{\text{new}}=\begin{cases}\min(P_{\text{c}}^{\text{initial}}\times 1.1,0.9), & \text{if } \Delta P_{\text{c}}=\text{increase},\\ \max(P_{\text{c}}^{\text{initial}}\times 0.9,0.4), & \text{if } \Delta P_{\text{c}}=\text{decrease},\\ P_{\text{c}}^{\text{initial}}, & \text{if } \Delta P_{\text{c}}=\text{keep},\end{cases}\tag{22}$$

$$P_{\text{m}}^{\text{new}}=\begin{cases}\min(P_{\text{m}}^{\text{initial}}\times 1.1,0.21), & \text{if } \Delta P_{\text{m}}=\text{increase},\\ \max(P_{\text{m}}^{\text{initial}}\times 0.9,0.01), & \text{if } \Delta P_{\text{m}}=\text{decrease},\\ P_{\text{m}}^{\text{initial}}, & \text{if } \Delta P_{\text{m}}=\text{keep}。\end{cases}\tag{23}$$

B_{action} 包含 2 个参数,S_{sele}(Selection Strategy)为选择策略的调整操作,S_{cros}(Crossover Method)为交叉方法的调整操作。S_{sele} 中包括排名选择(Rank Selection),轮盘赌选择(Roulette Wheel Selection)以及锦标赛选择(Tournament Selection)。锦标赛选择与排名选择在遗传算法的初期和中期阶段,可借助适度的选择压力保障种群的多样性,避免算法过早收敛。轮盘赌选择则能在后期阶段凭借较高的选择压力,促使算法快速收敛至最优解,有效平衡了种群多样性维持与加速收敛之间的矛盾。当适应度分布不均时,轮盘赌选择可能致使部分个体被过度选择,进而降低种群的多样性,此时可运用排名选择更精准地调控选择压力,避免个体适应度差异过大。通过集成锦标赛选择、排名选择和轮盘赌选择这三种方法,能够充分契合不同的优化需求与种群状态,同时利用强化学习代理依据当前状态做出决策,显著增强了选择操作环节的适应性[17]。此外,为解决最优解丢失问题,引入精英策略[18],选取种群中的部分优秀个体组成精英种群并保存备份,防止其在自适应交叉变异操作中流失。精英种群中的个体不参与交叉、变异等遗传操作,但为缓解精英个体占比过高问题,保留 5%的精英个体参与后续操作。S_{cros} 中包括顺序交叉(Order Crossover, OX)(如支撑材料图 1 所示)、部分映射交叉(Partially Mapped Crossover, PMX)[19](如支撑材料图 2 所示)和循环交叉(Cycle Crossover, CX)(如支撑材料图 3 所示)。

上述 2 个动作集共有 4 个参数,每个参数都有三种调整动作,因此整个动作集共有 81 种不同的动作组合,为了提高强化学习代理的学习效率,把单代理拆分,分成多代理的形式,通过这种方案每个代理只需要有 9 种不同的动作组合极大提高了强化学习代理的学习效率。本文采用 ε -贪婪策略(如式(24)所示)动作的选择,以使智能体在训练初期进行更多随机探索,并逐渐减少探索以便于专注最优策略,算法在利用阶段优先选择 Q 值最高的动作,从而加速收敛。

$$\text{Action}=\begin{cases}\text{Randomly select action}, & \text{with probability } \varepsilon,\\ \text{Select the action with the highest } Q\text{-value}, & \text{with probability } 1-\varepsilon。\end{cases}\tag{24}$$

2.3.3 融入香农多样性指数的奖励机制。奖励机制是整个过程的关键,将直接影响代理的学习效果。Chen 等人通过最佳适应度和平均适应度来决定 Agent 的奖赏[13]。鉴于本文将交叉和选择操作的决策权同样交由强化学习代理,若仅以最佳适应度与平均适应度作为奖励依据,难以充分刻画种群基因的多样性。香农多样性指数 H' 是生物统计学中用于衡量群落中物种多样性的一种指数,其充分考虑了物种的丰富度即物种的数量和均匀度即物种的相对丰度,所有物种在整个群落中的个体数量分布越均匀,那么香农指数越高。其中个体与个体的划分是通过汉明距离 $d_H(A,B)$ 的计算并经过形成层次树,从而进行物种划分进而计算所有个体的总数 N 和第 i 种物种的个体数 N_i。基于香农多样性指数变化 ΔD,最佳适应度变化 Δf_{b} 以及平均适应度变化 Δf_{a} 得出奖励 R,并通过约束惩罚 P_{pun} 进行约束,其中 f_b^{cur} 代表当前种群最佳适应度,f_b^{pre} 代表迭代前种群最佳适度,f_a^{cur},H'_{cur} 与 f_a^{pre},H'_{cur} 脚标含义同上。具体奖励机制计算公式如

$$H'=-\sum_{i=1}^{S}\frac{N_i}{N}\ln\left(\frac{N_i}{N}\right),\tag{25}$$

$$d_H(A,B)=\sum_{i=1}^{n}\delta(a_i,b_i)\ \text{其中}\ \delta(a_i,b_i)=\begin{cases}1,\text{if } a_i\neq b_i,\\ 0,\text{else},\end{cases}\tag{26}$$

$$R=0.5\times\Delta f_b+0.3\times\Delta f_a+0.2\times\Delta D-P_{\text{pun}},\tag{27}$$

$$R'=\max(\min(R,1.0),-1.0),\tag{28}$$

中国知网 https://www.cnki.net

$$\Delta f_b = \frac{f_b^{\text{cur}} - f_b^{\text{pre}}}{f_b^{\text{pre}}}, \tag{29}$$

$$\Delta f_a = \frac{f_a^{\text{cur}} - f_a^{\text{pre}}}{f_a^{\text{pre}}}, \tag{30}$$

$$\Delta D = \frac{H'_{\text{cur}} - H'_{\text{cur}}}{H'_{\text{pre}}}, \tag{31}$$

$$P_{\text{pun}} = \begin{cases} 1.0, \text{ if any Gene. total_cost} > \text{constraint_threshold}, \\ 0, \text{ otherwise}, \end{cases} \tag{32}$$

式(32)中 P_{pun} 表示如果种群中存在任何违反约束条件即超载，路径不合法等问题的个体，那么会给予负奖励。式(28)确保奖励值 R 限制在[−1.0，1.0]范围内，以确保学习过程中的数值稳定性。针对香农多样性指数的计算式(25)，其具体步骤如下：(1) 据式(26)计算个体与个体之间的汉明距离。(2) 根据距离合并最近的个体或者簇，从而形成树形结构。(3) 通过距离阈值对于物种进行划分即通过阈值分割层次树。式(29)～(31)分别计算了最佳与平均适应度变化及香农多样性指数的变化以用于引导自适应操作。

2.4　离散莱维飞行

交叉变异之后，不难发现早期出现了适应度远超群体平均适应度的超级个体，导致该个体迅速在种群中占据主导地位，从而使得种群的多样性迅速下降，失去了继续寻优的能力。为解决这个问题，引入了离散莱维飞行策略[20]，显著提高了全局搜索的能力。具体公式如

$$\alpha \oplus \text{Levi}(\beta) \sim 0.01 \frac{u}{|v|^{-\beta}}(X(t) - X_\alpha(t)) \ , u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2), \tag{33}$$

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta \cdot 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}}, \sigma_v = 1, \tag{34}$$

$$\lambda = \frac{u}{|v|^{1/\beta}}。 \tag{35}$$

式(33)定义了更新规则，结合了莱维飞行的长尾分布特性，通过随机变量 u 和 v 控制步长与方向。u 和 v 的分布均服从正态分布，其中 v 的方差设为 1。式(34)计算了步长的尺度因子 σ_u，其形式依赖稳定性指数 β，用于保持莱维分布的数学特性。式(35)定义了步长 λ，用以生成符合莱维分布的随机跳跃步长。其具体操作为：当步长 $\lambda \leqslant 1$ 时，根据一个个体，随机选择其一段基因序列进行逆序，模拟局部搜索。当 $\lambda > 1$ 时，随机选择两个个体，进行交叉操作进行重组，生成全新个体，模拟全局搜索，对整体进行操作之后，以新的种群与原种群对比，按比例保留适应度较高的个体，并进入下一次的遗传迭代。

2.5　2-opt 局部搜索策略

上述全局搜索后，为了提高算法的局部搜能力，本文采用 2-opt 局部搜索算法[11]。该算法的基本原理在于对现有解实施局部操作，识别并移除其中表现欠佳的部分，进而生成更优的解。具体而言，2-opt 算法通过调整路径中两条边的连接方式，有针对性地消除路径中存在的交叉状况或不合理布局，以此降低路径的总长度或成本。其核心思想聚焦于在路径中精准定位交叉边，随后执行非交叉交换操作，也就是改变点对之间的连接方式以消除交叉部分。这一过程在确保路径连通性不受影响的同时，实现了对路径的有效优化。该算法的伪代码如算法 2 所示。

3　仿真实验

本文所提出的算法以及对比算法均基于 Python 3.12 编程环境，在 Pycharm 集成开发环境中完成代码实现。实验运行环境部署于 16 台 ubuntu 服务器，每台服务器配备 32 v CPU 的 AMD EPYC 9654 96-Core Processor 处理器，内存配置为 60 GB，以此保障实验运行具备充足的计算资源和稳定的运行环境。

在测试数据集选取方面，由于当前针对本研究模型的公开基准实例集较难获取，仅检索到 Gao[5] 提出的算例。因此，以 Gao 的算例为基础构建一个小型算例集，详见链接：https://data.mendeley.com/datasets/

中国知网　https://www.cnki.net

h3mrm5dhxw/1,算例的组成详见支撑材料表1。对Goeke[21]提出的基准实例集进行适应性修改后用于测试。修改后的测试数据集涵盖不同规模的实例,其中小型规模实例12个,中等规模实例24个,大型规模实例56个。为确保实验结果的可靠性与稳定性,每个实例均独立运行16次。

算法2 2-optOptimization(max _iterations,invalid_iterations,invalid_count)

Input: 最大迭代次数 max _iterations=100,无效反转次数 invalid_iterations=0,用于提前终止,invalid_count=0

01: 初始化路径数组 path //由一系列节点组成的路径

02: initial_cost=计算路径总成本(path)

03: 获取当前最佳路径 best_path 和最佳路径长度 best_cost

04: **while** invalid_count<max _iterations **do**

05: 随机选择两个不同的节点索引 i , j , 且 $i<j$

06: new_path=反转路径(path,i,j) // 获取新路径 new_cost=计算路径总长度(new_path) // 计算新路径的总长度

07: **if** new_cost<best_cost **then** best_path=new_path, best_cost=new_cost, invalid_count=0

08: **else** invalid_count+=1 **end if**

09: **if** invalid_count>=10 **then** break //如果连续 10 次反转无效,提前退出优化循环 **end if**

10: **if** new_cost>best_cost×1.01 then break //如果新路径长度变化很小(比如大于原路径的 1%),则提前退出优化循环

11: **end if**

12: **End while**

Output : best_path, best_cost

3.1 消融实验

为全面验证本文所提出的多动作集设计的合理性、局部搜索策略的有效性以及香农多样性指数的作用效能,本节开展消融实验。具体而言,将去除局部搜索策略后的算法标记为MRLGA_noopt;未进行动作集拆分且同时去掉局部搜索策略的方法标记为RLGA;去除香农多样性指数和局部搜索策略的算法标记为MRLGA_noShanno。在实验对比设置中,安排MRLGA_noopt与MRLGA进行直接对比,以直观展现局部搜索策略对算法性能的影响;同时,使MRLGA_noopt、RLGA、MRLGA_noShanno进行横向比较,通过这种方式间接实现与MRLGA的对比,可以从多维度深入剖析各因素对算法性能的影响。所有对比算法的参数设置相同,均为:种群数量设置为800,迭代次数2 000。学习率为0.1,折扣因子为0.9,探索率为1,最小探索率为0.05。

由表2可知,本文所提多代理强化遗传算法得出的最佳成本为6 498元,相较于Gao研究结果中的6 543元[5],成本降低约0.7%。图2所示是MRLGA在Gao算例[5]上的结果,分别代表适应度和总成本随代数变化,香农多样性指数随代数变化和最优路径图。表3给出了92个算例按换电站个数分组后的结果,92个算例全部结果详见支撑材料中表2。由表3和支撑材料中表2可知,MRLGA的性能更优,清晰展现出多代理强化遗传算法在解决此类问题时具备显著优势与良好效果。对比MRLGA_noopt与MRLGA可知,二者性能差异显著。在小型算例中,两者结果一致,但在其余算例里,MRLGA的性能均优于未采用局部搜索策略的MRLGA_noopt,证实了局部搜索策略在提升算法性能方面的有效性。从横向比较来看,RLGA的性能远逊于MRLGA_noopt,且MRLGA_noopt在所有算例中的结果均优于RLGA。这一系列对比结果有效验证了动作集设计的合理性,凸显出合理设计动作集能够显著提升算法性能。当比较RLGA_noopt与MRLGA_noShanno时,二者所得的平均最低成本极为接近,波动约为1%~3%。然而,在大部分算例中,MRLGA_noopt的结果更优。这一现象表明,香农多样性指数在提升解的质量方面具备一定的有效性。此外,通过图2可知,香农多样性指数在每次出现下降趋势后,均能迅速回升。这一现象有力验证了强化学习代理所提供的决策策略以及多样性调整方案的有效性,表明强化学习代理能够根据算法运行状态,动态且有效地调整策略,维持种群多样性,进而推动算法不断向更优解逼近。但是可以看出在计算香农多样性指数时

中国知网 https://www.cnki.net

时间开销巨大,导致了算法求解性能有所下降,但是在解的质量上有所提升。

表 2　4 个算法在 Gao[5] 测试算例上的平均成本值

Tab. 2　The average cost value of 4 algorithms in Gao[5] test instances

Gao	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA
6 543	6 498	6 498	6 498	6 498

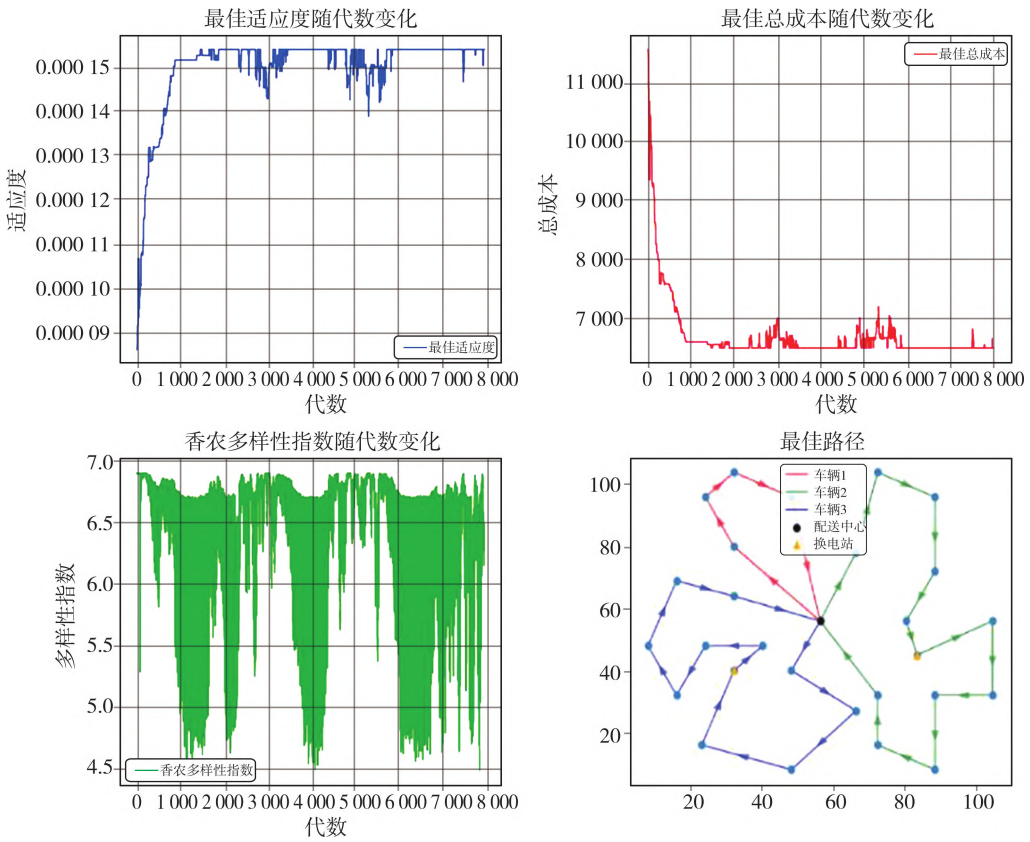

图 2　所提算法在 Gao[5] 算例过程分析图和最佳路径

Fig. 2　process analysis diagram

表 3　消融实验在 Goeke[21] 测试用算例上的结果(将 92 个算例按换电站个数分组)

Tab. 3　Results of the 4 ablation experiment on the Goeke[21] test case group by the number of change stations

测试	算例	C_5	C10	C15	C21
c	RLGA	3 790	6 575	9 107	50 131
	MRLGA_noopt	3 790	6 559	8 911	44 100
	MRLGA_noShanno	3 790	6 557	8 858	44 548
	MRLGA	3 790	6 557	8 832	20 574
t	RLGA	288.85	393.26	507.52	1 956.30
	MRLGA_noopt	464.62	607.02	745.90	2 486.68
	MRLGA_noShanno	360.68	452.01	538.72	1 600.64
	MRLGA	561.68	683.22	782.43	2 038.72

注:c 表示 Average cost,单位为元; t 表示 Time,单位为秒。

3.2　整体算法的性能比较

为了验证算法的可行性与性能,总共选取 4 种算法进行实验操作,种群数量设置为 800,迭代次数 2 000,其它其参数选择如:GA[5]:选取的交叉和选择决策与本文的相同,变异概率 0.05,交叉概率 0.85;RLIGA[13]:使用强化学习代理 P_c 和 P_m,并且引入离散莱维飞行策略,学习率为 0.1,折扣因子为 0.9,探索

中国知网　https://www.cnki.net

率为1,最小探索率为0.05;INACS[7]:蚂蚁数量 $m=200$,参数 $q=0.6$,$\beta=4$,信息素挥发率 $\rho=0.7$。移位算子的长度=交换算子的次数=逆算子字符串长度=$n/10$;MRLGA:学习率为0.1,折扣因子为0.9,探索率为1,最小探索率为0.05。

表4给出了92个算例按换电站个数分组后的结果,92个算例全部结果详见支撑材料中表3。由表5和支撑材料中表3可知,MRLGA(绿色曲线)和IMACS(红色曲线)的求解效果表现更优。具体而言,MRLGA在90个算例中取得了最低成本解,展现出较强的全局寻优能力。相比之下,IMACS有17个最优解,RLIGA(黄色曲线)有38个。图3、图4分别展示了四种算法的最终成本箱线图和收敛过程曲线图。从收敛曲线来看MRLGA(绿色曲线)具有更快的收敛速度,能够更早地达到较低的目标值(总成本);而箱线图显示,MRLGA的成本最小且分布范围最窄,说明其在多次运行中的稳定性更强、波动性更小在算法运行时间方面,MRLGA的平均运行时间为552.34 s,虽然明显慢于GA(96.31 s)和与RLIGA(388.47 s),但与IMACS比较下在中小型算例下计算效率较低,但在大型算例中,其运算时间与IMACS相当。通过表3中的消融实验可见,MRLGA主要的时间开销来源于计算香农多样性指数时,构建距离矩阵和进行层次聚类的时间复杂度约为 $O(n^2)$ 和 $O(n^3)$(时间复杂度分析详见支撑材料第3节),当种群规模较大时会形成显著的计算瓶颈。综上所述,这些图表与数据充分表明,MRLGA在求解基于换电站模式的EVRPTW问题时具备巨大潜力与优势,同时也依然存在进一步优化的空间,特别是在计算效率与时间复杂度方面,有望通过近似聚类、矩阵稀疏化等手段进行改进。

表4 4个算法在Goeke[21]测试用算例上的结果(将92个算例按换电站个数分组)

Tab. 4 Results of the 4 algorithms on the Goeke test case group by the number of change stations

测试	算例	C_5	C10	C15	C21
c	GA	3 790	6 573	9 086	56 727
	RLIGA	3 790	6 558	8 860	43 851
	IMACS	3 892	7 034	9 743	38 390
	MRLGA	3 790	6 557	8 832	20 574
t	GA	103.30	146.36	192.43	832.34
	RLIGA	406.23	541.30	676.49	2 522.45
	IMACS	25.57	48.02	79.60	3 447.15
	MRLGA	561.68	683.22	782.43	2 038.72

注:c 表示 Average cost,单位为元;t 表示 Time,单位为秒。

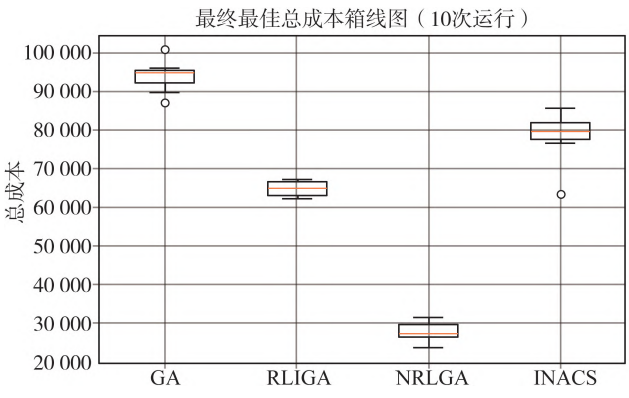

图3 最佳成本的箱线图

Fig. 3 Box plot of optimal cost

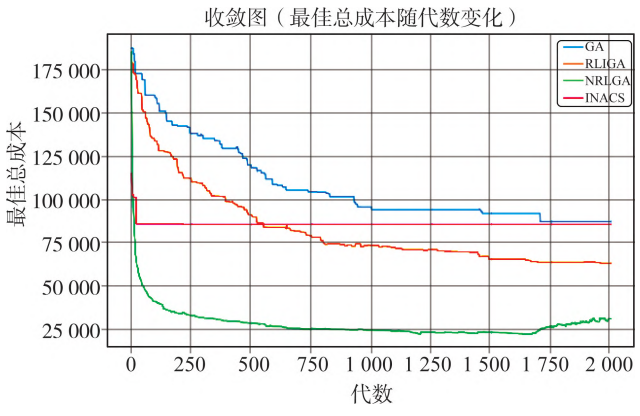

图4 所有算法的收敛过程图

Fig. 4 Convergence process diagram for all algorithms

4 结论

本文针对具有软时间窗约束的电动车路径问题,构建了相应的数学模型,并提出多代理强化遗传算法求

中国知网 https://www.cnki.net

解该问题。该算法结合多代理方法与遗传算法的优势，借助智能体间的协作与竞争，提升算法的搜索效率与求解精度。同时，引入莱维飞行策略强化全局搜索能力以及运用 2-opt 局部搜索算法对局部搜索进行优化，实现全局和局部搜索的有效平衡。此外，提出利用香农多样性指数结合汉明距离来精准衡量种群的多样性，有效避免过早收敛。通过设计的 93 个测试算例进行实验验证，结果充分表明了所提出的多代理强化遗传算法（MRLGA）在求解该问题时具有显著的优越性。然而，本研究仍存在一定的局限性。从模型构建角度分析，在实际日常生活场景中，车辆数量通常并非固定不变，而是会根据需求动态变化；同时，大多数换电站在实际运营中可被多次访问，且电动车电池的耗电过程并非呈现简单的线性关系，这些复杂特性在当前模型中尚未得到充分考虑。此外，诸如车辆故障、交通事故以及订单动态变化等不确定因素也未纳入模型范畴，这在一定程度上限制了模型对现实场景的刻画能力。从算法实现层面来看，在多代理模式下，动作集拆分后可能产生数量众多的组合，若在探索过程中搜索深度和广度不足，将导致选择交叉策略的学习能力受限，进而使得算法的优化效果无法达到预期。综上所述，未来的研究可着重从以下两个方向展开：一是深入探讨非线性电池特性以及各类不确定因素对换电模式下电动车路径规划的影响，通过改进模型以更准确地描述现实场景；二是针对当前算法存在的问题，深入研究更具针对性的算法，改进算法结构和参数设置，提升学习能力，增强算法的稳定性与优化效果。

参　考　文　献

[1] ZHANG X, HAO Y C, ZHANG L Y, et al. Application of improved genetic algorithm to vehicle routing problem considering the environmental self-regulation of the freight companies[J]. Expert Systems with Applications, 2025, 274: 127010.

[2] 杨涛，孙晓莉. 气候暖化和“双碳”目标约束下的大城市交通发展策略[J]. 交通与港航，2021，8(6)：3-7.

[3] 狄卫民，杜慧莉，张鹏阁. 考虑动态拥堵的多车型绿色车辆路径问题优化[J]. 计算机工程与设计，2021，42(9)：2614-2620.

[4] ZHOU Y M, LIU L H, BENLIC U, et al. Solving soft and hard-clustered vehicle routing problems: a bi-population collaborative memetic search approach[J]. European Journal of Operational Research, 2025, 324(3): 825-838.

[5] 高升. 基于电动汽车的带时间窗的路径优化问题研究[D]. 大连：大连海事大学，2015.

[6] PÓVOA C L R, ROBOREDO M C, VELASCO A S, et al. A hybrid GRASP and tabu-search heuristic and an exact method for a variant of the multi-compartment vehicle routing problem[J]. Expert Systems with Applications, 2025, 259: 125319.

[7] 童瑞，吕明，张捷. 改进的多蚁群系统算法解决具有时间窗约束的车辆路径问题[J]. 工业控制计算机，2025，38(2)：86-87.

[8] 何小年. 带软时间窗约束的多车场车辆路径问题及其禁忌搜索算法研究[J]. 信息记录材料，2023，24(1)：97-99.

[9] 刘志硕，刘若思，陈哲. 基于混合蚁群算法的冷链电动汽车车辆路径问题[J]. 计算机应用，2022，42(10)：3244-3251.

[10] 王扬，陈智斌. 一种求解 CVRP 的动态图转换模型[J]. 计算机工程与科学，2023，45(5)：859-868.

[11] 何美玲，付文青，韩珣，等. 软时间窗下考虑冷链物流多温共配的电动汽车路径优化[J]. 江苏大学学报（自然科学版），2024，45(6)：629-635.

[12] 白雪媛，张磊，李琳. 基于 GA-ALNS 算法的带可容忍时间窗的 VRP 求解[J]. 沈阳师范大学学报（自然科学版），2024，42(3)：273-279.

[13] 陈祉烨，胡毅，刘俊，等. 利用强化学习的改进遗传算法求解柔性作业车间调度问题[J]. 科学技术与工程，2024，24(25)：10848-10856.

[14] MESA J P, MONTOYA A, RAMOS-POLLAN R, et al. Machine-learning component for multi-start metaheuristics to solve the capacitated vehicle routing problem [J]. Applied Soft Computing, 2025, 173: 112916.

[15] 刘超，韩玉艳，杨晓宇，等. 多目标协同进化算法求解具有可变加工速度和自动导引车的柔性作业车间调度问题[J]. 聊城大学学报（自然科学版），2025，38(3)：317-332.

[16] 唐子燕，刘济明，陈敬忠，等. 主要产地米槁根际土壤可培养细菌多样性特征及差异[J]. 分子植物育种，2025，23(14)：4765-4773.

[17] 张玉昊，纪洪泉. 基于改进即时学习策略的非线性多模态过程故障检测方法[J]. 山东科技大学学报（自然科学版），2024，43(6)：124-134.

[18] 王储，南英，许航. 一种新的精英遗传算法及在多弹拦截分配策略的应用[J]. 航天控制，2021，39(4)：59-66.

[19] 姜薇. 遗传算法中交叉算法的改进[D]. 长春：吉林大学，2009.

[20] 陶琬丛，宋睿，田汉. 物料配送路径离散莱维飞行改进遗传算法规划[J]. 机械设计与制造，2025(7)：316-318.

[21] GOEKE DOMINIK. “E-VRPTW instances” [DS/OL]. V1, Mendeley Data, 2019. doi: 10.17632/h3mrm5dhxw.1.

（责任编辑：赵海军）

中国知网　https://www.cnki.net

文章编号 1672-6634(2026)01-0032-12 **DOI** 10.19728/j.issn1672-6634.2025040010

支撑材料

Supporting Information

多代理强化学习驱动遗传算法求解软时间窗电动车路径规划问题

Multi-agent reinforcement learning-driven genetic algorithm for solving the soft time window electric vehicle routing problem

韩玉艳,安俊宇,杨晓宇,王玉亭,李寰,田鑫茹

HAN Yuyan, AN Junyu, YANG Xiaoyu, WANG Yuting, LI Huan, TIAN Xinru

(聊城大学 计算机学院,山东 聊城 252059)

(School of Computer Science, Liaocheng University, Liaocheng 252059, China)

1 基因交叉方法和变异方法示意图及程序流程图

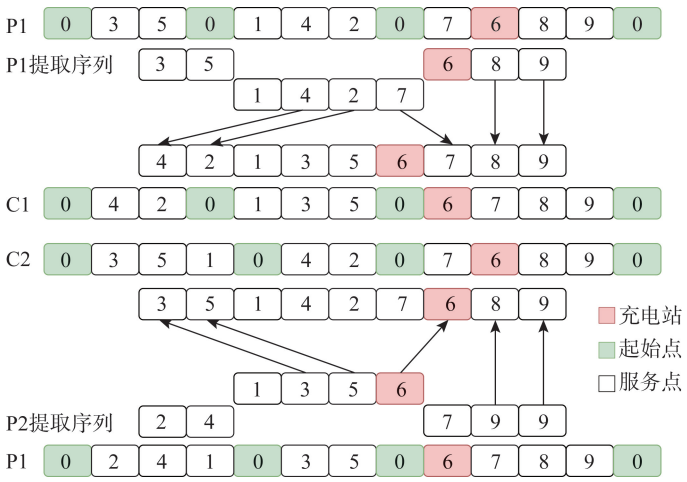

图 1 OX 操作

Fig. 1 Order Crossover diagram

收稿日期:2025-04-15

基金项目:国家自然科学基金项目(61973203, 61803192, 62106073, 61966012);山东省自然科学基金项目(ZR2023MF022, ZR2024MF112);聊城大学光岳青年创新团队项目(LCUGYTD2022-03)资助

通信作者:韩玉艳,女,汉族,博士,副教授,研究方向:智能优化调度,E-mail:hanyuyan@lcu-cs.com。

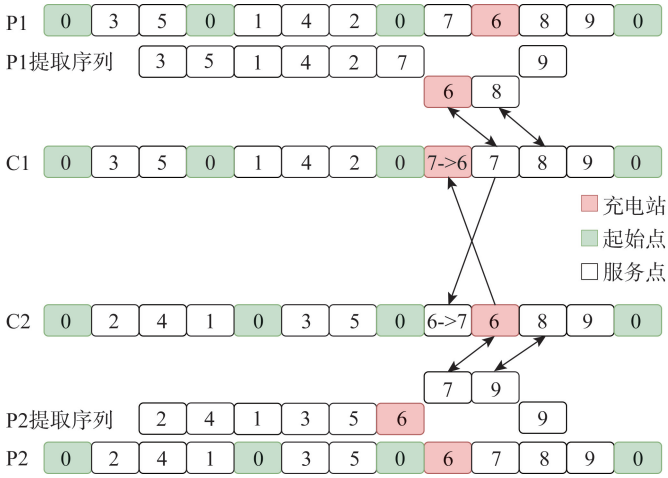

图 2　PMX 操作

Fig. 2　Partially Mapped Crossover diagram

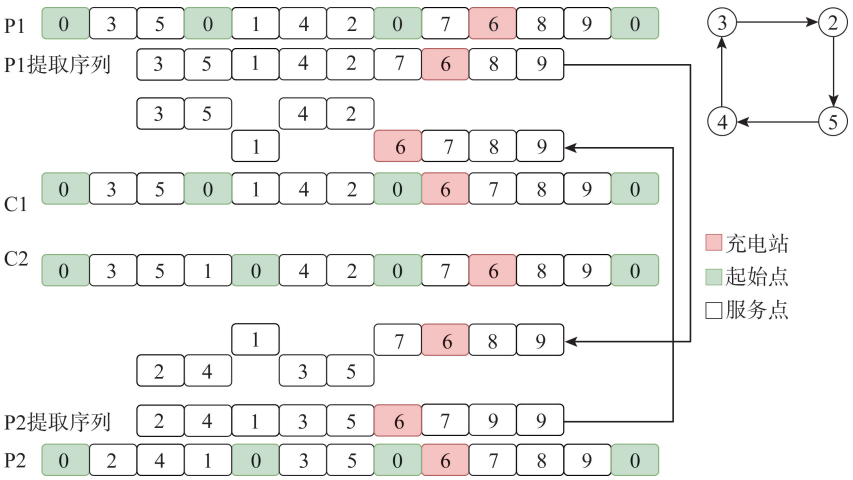

图 3　CX 操作

Fig. 3　Cycle Crossover diagram

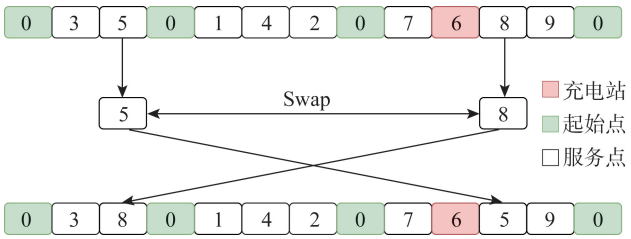

图 4　交换变异操作

Fig. 4　Swap Mutation diagram

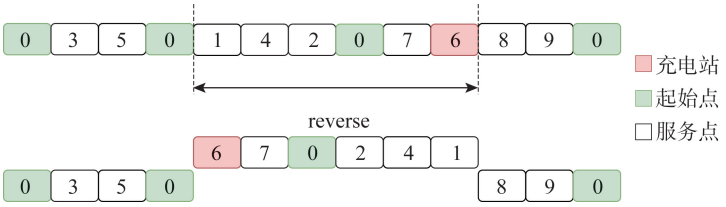

图 5　反转变异操作

Fig. 5　Reverse Mutation diagram

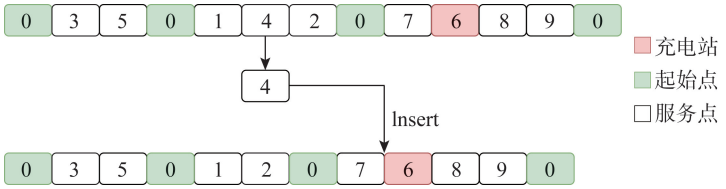

图 6　插入变异操作

Fig. 6　Insert Mutation diagram

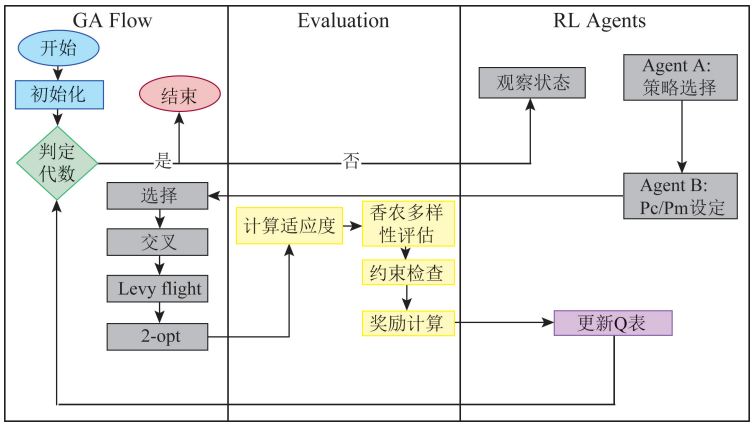

图 7　算法程序流程图

Fig. 7　Algorithm program flow chart

2　实验数据结果

2.1　实验数据

表 1　Gao 提出的测试算例

Tab. 1　Test instances proposed by Gao

编号	*X*	*Y*	需求量	最早到达	最晚到达	服务时间	类型
0	56	56	0	0	100	0	起始点
1	66	78	0.2	0	1	0.2	客户
2	66	27	0.3	1	2	0.3	客户
3	88	72	0.3	2	4	0.3	客户
4	88	32	0.3	7	8	0.3	客户
5	24	48	0.3	5	6	0.3	客户
6	40	48	0.5	3	5	0.5	客户
7	32	80	0.8	0	2	0.8	客户
8	16	69	0.4	7	8	0.4	客户
9	88	96	0.5	1	3	0.5	客户
10	48	96	0.7	4	5	0.7	客户
11	32	104	0.7	1	2	0.7	客户
12	80	56	0.6	3	4	0.6	客户
13	48	40	0.2	0	1	0.2	客户
14	23	16	0.2	2	4	0.2	客户
15	48	8	0.4	1	2	0.4	客户
16	16	32	0.1	7	8	0.1	客户
17	8	48	0.1	6	8	0.1	客户

(续)表 1 Gao 提出的测试算例

(Continued)Tab. 1 Test instances proposed by Gao

编号	X	Y	需求量	最早到达	最晚到达	服务时间	类型
18	32	64	0.2	7	9	0.2	客户
19	24	96	0.5	1	3	0.5	客户
20	72	104	0.2	1	3	0.2	客户
21	72	32	0.7	8	10	0.7	客户
22	72	16	0.2	6	10	0.2	客户
23	88	8	0.7	7	8	0.7	客户
24	104	56	0.1	6	7	0.1	客户
25	104	32	0.4	4	6	0.5	客户
26	83	45	0.5	0	100	0.4	换点站
27	32	40	0.4	0	100	0.4	换点站

Goeke 的测试算例详见链接:https://data.mendeley.com/datasets/h3mrm5dhxw/1。算例的名称型如[c/rc/r][序列号][c5/c10/c15/_21]可以分成三个部分:第一个部分代表着节点的分布特征,在 R 类数据集中,节点的坐标位置随机生成,较为分散;在 C 类数据集中,节点的坐标位置有明显的聚簇,较为集中,而 RC 类数据集则混合了随机和聚簇的特点。第三部分代表换点站个数或者客户个数即问题的规模大小。

2.2 实验结果

表 2 4 个消融实验算法在 Goeke 测试用例上的结果

Tab. 2 Results of the 4 ablation experiment algorithms on the Goeke test case

测试	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA
c101C5	4 193	4 193	4 193	**4 193**	**266.59**	448.40	300.57	552.34
c103C5	3 012	3 012	3 012	**3 012**	**247.95**	423.12	282.697 5	532.88
c206C5	7 214	7 214	7 214	**7 214**	**290.88**	480.53	320.04	526.63
c208C5	5 510	5 510	5 510	**5 510**	**269.81**	453.09	299.1	546.34
c101C10	10 618	10 552	10 552	**10 552**	**406.83**	630.68	422.07	712.86
c104C10	4 477	4 475	4 475	**4 475**	**376.37**	594.25	398.85	644.64
c202C10	9 693	9 678	9 678	**9 678**	**393.34**	622.73	405.8175	678.92
c205C10	10 080	10 080	10 080	**10 080**	368.00	590.88	392.025	667.93
c103C15	8 828	8 596	8 596	**8 596**	483.13	734.10	**433.2375**	786.43
c106C15	6 611	5 949	5 949	**5 949**	417.22	651.10	**411.195**	683.33
c202C15	13 352	13 104	13 104	**13 104**	500.63	747.99	**493.372 5**	789.03
c208C15	14 471	14 343	14 343	**14 343**	493.16	736.99	**488.617 5**	763.47
c101_21	77 756	64 874	66 190	**24 563**	1 866.33	2 570.77	**1 665.1575**	2 039.95
c102_21	63 247	55 281	56 965	**22 138**	1 868.79	2 565.36	**1 688.1075**	2 065.30
c103_21	52 040	44 009	45 127	**19 539**	1 844.11	2 547.39	**1 675.65**	1 983.10
c104_21	41 259	37 112	36 627	**16 845**	1 825.28	2 520.66	**1 664.385**	1 945.56
c105_21	70 299	61 157	61 090	**22 521**	1 851.32	2 581.96	**1 663.5225**	1 999.25
c106_21	63 562	51 482	56 629	**21 163**	1 867.52	2 556.46	**1 674.4575**	2 017.29
c107_21	63 889	56 833	55 504	**20 829**	1 846.62	2 568.62	**1 670.8575**	2 029.88
c108_21	57 783	49 570	50 650	**19 466**	1 847.82	2 576.85	**1 659.7575**	2 005.91
c109_21	50 823	45 238	46 932	**17 670**	1 847.64	2 559.33	**1 654.4175**	1 977.98
c201_21	109 452	90 494	91 447	**29 027**	1 858.53	2 572.36	**1 665.195**	2 006.14
c202_21	84 931	75 413	74 387	**23 226**	1 851.41	2 537.78	**1 791.18**	1 992.58

(续)表 2　4 个消融实验算法在 Goeke 测试用例上的结果

(Continued)Tab. 2　Results of the 4 ablation experiment algorithms on the Goeke test case

测试	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA
c203_21	64 473	56 133	57 344	**19 296**	**1 834.51**	2 544.13	2 044.695	2 019.67
c204_21	43 741	37 373	39 396	**16 258**	**1 792.95**	2 501.69	2 001.825	1 988.02
c205_21	93 977	83 790	83 862	**23 831**	**1 859.93**	2 569.09	1 658.475	1 998.58
c206_21	80 240	65 970	71 101	**20 389**	1 835.96	2 548.59	**1 867.8225**	1 969.58
c207_21	74 311	64 824	63 082	**19 063**	1 851.52	2 558.19	**1 651.2075**	1 982.86
c208_21	76 546	66 467	64 576	**18 718**	1 844.67	2 549.59	**1 641.3675**	1 969.96
r104C5	2 018	2 018	2 018	**2 018**	280.46	452.39	**275.132 5**	521.74
r105C5	2 637	2 637	2 637	**2 637**	287.04	461.70	**268.42**	577.23
r202C5	2 491	2 491	2 491	**2 491**	290.62	475.37	**284.185**	551.31
r203C5	3 852	3 852	3 852	**3 852**	324.72	485.92	**306.577 5**	590.03
r102C10	4 361	4 335	4 335	**4 335**	**378.38**	607.89	**381.817 5**	661.51
r103C10	2 240	2 225	2 225	**2 225**	**362.74**	547.82	**519.9**	636.87
r201C10	4 419	4 368	4 368	**4 368**	**382.39**	626.13	392.817 5	698.61
r203C10	6 102	6 104	6 078	**6 078**	**444.20**	671.16	580.007 5	684.85
r102C15	7 701	7 345	7 477	**7 289**	**549.48**	811.32	578.492 5	811.00
r105C15	7 220	7 238	7 017	**7 002**	**525.70**	811.90	584.912 5	790.54
r202C15	9 349	9 243	9 220	**9 220**	575.06	819.14	**505.652 5**	810.88
r209C15	8 430	8 179	8 179	**8 179**	**539.98**	747.73	568.822 5	784.19
r101_21	44 970	40 464	41 308	**26 910**	2 251.14	3 017.34	**1 674.0375**	2 238.65
r102_21	40 874	38 523	38 839	**24 427**	2 289.79	2 938.23	**1 485.547 5**	2 209.89
r103_21	37 762	35 113	35 272	**22 182**	2 272.64	2 929.46	**500.272 5**	2 192.41
r104_21	33 145	29 712	30 705	**19 219**	2 207.56	2 828.44	**1 921.77**	2 137.20
r105_21	42 316	39 825	39 271	**25 546**	2 320.50	2 921.83	**1 820.37**	2 205.54
r106_21	39 331	35 823	37 132	**23 324**	2 342.44	2 903.02	**1 502.017 5**	2 208.97
r107_21	36 146	33 432	33 130	**20 909**	**2 248.04**	2 852.31	2 400.252 5	2 172.80
r108_21	32 860	30 392	29 946	**19 424**	**2 301.15**	2 831.74	2 513.607 5	2 130.23
r109_21	38 687	35 335	35 898	**23 092**	**2 311.18**	2 861.92	2 325.8	2 196.95
r110_21	35 330	32 647	32 078	**21 179**	2 297.91	2 819.85	**1 620.697 5**	2 166.56
r111_21	35 502	32 487	32 833	**21 557**	2 330.80	2 844.16	**1 569.87**	2 162.23
r112_21	32 571	29 101	29 655	**20 357**	2 300.54	2 840.06	**2 330.447 5**	2 166.51
r201_21	52 768	47 344	47 125	**22 276**	**1 900.14**	2 375.06	1 939.522 5	2 030.57
r202_21	44 154	39 514	38 963	**18 267**	**1 825.60**	2 285.85	1 926.7575	1 950.99
r203_21	37 687	33 648	35 066	**16 002**	1 808.01	2 242.69	**1 521.37**	1 931.89
r204_21	30 630	27 312	26 260	**12 145**	**1 736.68**	2 214.55	1 933.612 5	1 880.32
r205_21	44 572	38 946	39 580	**19 282**	**1 803.59**	2 246.12	1 983.952 5	1 956.59
r206_21	40 721	36 387	36 002	**17 153**	1 807.63	2 267.61	299.527 5	1 957.63
r207_21	34 122	30 343	30 300	**13 714**	**1 743.47**	2 207.21	1 978.3425	1 871.04
r208_21	29 047	25 061	24 777	**11 918**	**1 752.47**	2 188.97	1 932.015	1 902.42
r209_21	39 716	35 992	34 753	**16 933**	1 813.97	2 260.43	**1 472.507 5**	2 002.91
r210_21	38 529	33 293	34 569	**16 242**	1 811.51	2 264.95	**1 664.047 5**	1 952.64
r211_21	33 572	29 903	29 495	**14 067**	1 746.91	2 207.45	468.81	1 909.71

(续)表 2 4 个消融实验算法在 Goeke 测试用例上的结果

(Continued) Tab. 2 Results of the 4 ablation experiment algorithms on the Goeke test case

测试	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA	RLGA	MRLGA_noopt	MRLGA_noShanno	MRLGA
rc105C5	2 866	2 866	2 866	**2 866**	**298.69**	478.29	484.52	580.39
rc108C5	5 155	5 155	5 155	**5 155**	**319.00**	503.26	468.905	581.88
rc204C5	3 447	3 447	3 447	**3 447**	**282.43**	451.72	470.127 5	587.22
rc208C5	3 488	3 488	3 488	**3 488**	**285.79**	**445.48**	507.732 5	582.87
rc102C10	5 545	5 534	5 534	**5 534**	**408.10**	597.41	502.94	671.18
rc108C10	5 319	5 319	5 319	**5 319**	**402.19**	610.31	504.372 5	716.82
rc201C10	7 348	7 341	7 341	**7 341**	**392.89**	587.49	515.667 5	719.27
rc205C10	8 696	8 694	8 694	**8 694**	**403.67**	597.54	407.865	705.16
rc103C15	7 286	7 219	7 162	**7 162**	**535.51**	745.70	668.96	800.68
rc108C15	5 952	5 850	5 693	**5 663**	**463.61**	697.82	695.622 5	772.97
rc202C15	11 964	11 859	11 749	**11 749**	**510.79**	734.88	**339.465**	817.66
rc204C15	8 122	8 001	7 807	**7 723**	**495.99**	712.11	696.23	779.02
rc101_21	51 140	46 086	46 751	**25 940**	2 148.00	2 552.61	**1 333.62**	2 186.60
rc102_21	47 636	42 010	42 732	**24 312**	2 146.79	2 545.42	**1 348.5**	2 124.46
rc103_21	44 363	39 610	39 986	**22 881**	2 128.36	2 552.24	**1 381.92**	2 112.09
rc104_21	40 509	34 684	36 275	**20 866**	2 107.54	2 533.36	**1 562.932 5**	2 081.24
rc105_21	47 372	42 259	43 183	**24 298**	2 113.87	2 484.06	**1 341.12**	2 145.01
rc106_21	46 831	40 435	42 929	**23 904**	2 110.58	2 466.78	**1 329.577 5**	2 134.96
rc107_21	43 164	38 598	38 003	**22 205**	2 098.40	2 436.01	**1 351.005**	2 102.92
rc108_21	41 014	35 381	36 483	**21 520**	2 091.24	2 378.02	**1 341.99**	2 129.85
rc201_21	64 288	57 512	54 553	**26 884**	1 773.81	2 101.75	**1 597.9425**	1 997.94
rc202_21	54 401	47 408	46 572	**21 836**	1 708.58	2 057.84	**399.465**	1 919.32
rc203_21	46 479	41 730	42 152	**20 000**	1 739.16	2 065.27	**1 499.86**	1 981.45
rc204_21	39 146	34 814	35 148	**16 829**	1 765.46	2 080.85	**1 297.105**	1 919.90
rc205_21	54 539	48 984	48 706	**21 919**	1 768.97	2 072.02	**1 398.97**	1 935.35
rc206_21	53 968	45 717	47 025	**20 997**	1 739.07	2 046.46	**1 622.52**	1 962.63
rc207_21	46 768	40 815	42 838	**18 972**	1 732.32	2 057.12	**1 300.225**	1 980.79
rc208_21	42 354	36 933	37 492	**18 103**	1 761.93	2 046.42	**1 405.86**	1 927.60

注:第 1 栏表示 Average cost,单位为元; 第 2 栏表示 Time,单位为秒。

表 3 4 个算法在 Goeke 测试用例上的结果

Tab. 3 Results of the 4 algorithms on the Goeke test case

测试	GA	RLIGA	IMACS	MRLGA	GA	RLIGA	IMACS	MRLGA
c101C5	4 193	4 193	4 193	**4 193**	96.31	388.47	**25.94**	552.34
c103C5	3 012	3 012	3 029	**3 012**	87.33	357.74	**21.64**	532.88
c206C5	7 214	7 214	8 272	**7 214**	107.33	421.37	**27.93**	526.63
c208C5	5 510	5 510	5 552	**5 510**	96.79	391.96	**23.79**	546.34
c101C10	10 602	10 557	11 640	**10 552**	156.04	569.45	**51.93**	712.86
c104C10	4 475	4 475	4 475	**4 475**	143.43	530.20	**48.90**	644.64
c202C10	9 707	9 678	12 035	**9 678**	148.92	543.35	**52.30**	678.92
c205C10	10 080	10 080	**10 697**	10 080	139.40	526.03	**42.59**	667.93
c103C15	8 889	8 623	10 038	**8 596**	187.45	667.93	**77.82**	786.43

(续)表 3　4 个算法在 Goeke 测试用例上的结果

(Continued) Tab. 3　Results of the 4 algorithms on the Goeke test case

测试	GA	RLIGA	IMACS	MRLGA	GA	RLIGA	IMACS	MRLGA
c106C15	6 487	6 016	6 245	**5949**	154.52	585.96	**75.18**	683.33
c202C15	13 334	13 123	16 322	**13 104**	194.22	685.41	**75.32**	789.03
c208C15	14 523	14 343	16 527	**14 343**	193.41	669.38	**69.37**	763.47
c101_21	95 403	66 242	71 570	**24 563**	**789.38**	2 584.48	3 007.00	2 039.95
c102_21	74 241	55 573	56 383	**22 138**	**769.63**	2 568.31	2 953.15	2 065.30
c103_21	60 445	46 108	41 595	**19 539**	**782.40**	2 518.37	2 971.26	1 983.10
c104_21	46 921	36 587	25 723	**16 845**	**772.25**	2 527.59	2 980.18	1 945.56
c105_21	86 149	60 718	63 182	**22 521**	**765.56**	2 547.43	2 980.86	1 999.25
c106_21	78 174	54 483	55 253	**21 163**	**787.20**	2 569.29	2 985.14	2 017.29
c107_21	77 264	54 983	55 173	**20 829**	**782.89**	2 550.89	2 958.20	2 029.88
c108_21	70 079	49 868	48 885	**19 466**	**772.20**	2 566.13	2 945.33	2 005.91
c109_21	60 762	45 161	39 028	**17 670**	**768.38**	2 521.42	2 947.12	1 977.98
c201_21	130 015	92 458	71 118	**29 027**	**770.93**	2 516.10	3 124.67	2 006.14
c202_21	96 964	72 273	57 181	**23 226**	**768.66**	2 508.92	3 128.92	1 992.58
c203_21	72 256	55 328	46 439	**19 296**	**756.80**	2 489.93	3 095.76	2 019.67
c204_21	45 717	36 076	26 619	**16 258**	**745.14**	2 455.83	3 027.76	1 988.02
c205_21	115 392	79 644	59 512	**23 831**	**752.29**	2 508.07	3 123.04	1 998.58
c206_21	98 674	68 865	51 561	**20 389**	**755.99**	2 504.28	3 116.78	1 969.58
c207_21	87 688	62 258	46 684	**19 063**	**771.33**	2 537.26	3 113.32	1 982.86
c208_21	88 977	63 493	45 242	**18 718**	**772.94**	2 493.28	3 115.36	1 969.96
r104C5	2 018	2 018	2 018	**2 018**	97.66	385.31	**23.34**	521.74
r105C5	2 637	2 637	2 637	**2 637**	102.79	396.55	**25.03**	577.23
r202C5	2 491	2 491	2 491	**2 491**	103.45	411.40	**24.29**	551.31
r203C5	3 852	3 852	3 852	**3 852**	116.93	449.99	**26.14**	590.03
r102C10	4 339	4 335	4 335	**4 335**	141.11	529.27	**46.15**	661.51
r103C10	2 225	2 225	2 348	**2 225**	129.56	495.88	**44.53**	636.87
r201C10	4 452	4 382	5 155	**4 368**	139.66	547.74	**50.31**	698.61
r203C10	6 101	6 078	6 100	**6 078**	164.45	598.14	**49.06**	684.85
r102C15	7 643	7 383	7 596	**7 289**	211.16	743.50	**92.38**	811.00
r105C15	7 167	7 036	7 004	**7 002**	199.00	699.67	**83.50**	790.54
r202C15	9 357	9 226	10 343	**9 220**	220.97	748.24	**80.30**	810.88
r209C15	8 477	8 180	9 306	**8 179**	204.80	677.37	**73.40**	784.19
r101_21	47 362	41 648	36 615	**26 910**	**1 064.28**	3 042.71	1 419.67	2 238.65
r102_21	43 405	37 781	32 654	**24 427**	**1 108.89**	3 005.83	1 415.85	2 209.89
r103_21	39 131	34 476	28 821	**22 182**	**1 105.61**	2 966.49	1 419.69	2 192.41
r104_21	35 064	30 070	23 935	**19 219**	**1 052.74**	2 894.12	1 491.76	2 137.20
r105_21	44 730	38 994	33 570	**25 546**	**1 108.49**	3 032.45	1 416.54	2 205.54
r106_21	41 095	36 122	30 450	**23 324**	**1 095.30**	2 976.97	1 416.93	2 208.97
r107_21	38 282	33 126	27 297	**20 909**	**1 060.36**	2 922.41	1 464.40	2 172.80
r108_21	34 168	29 258	24 193	**19 424**	**1 063.70**	2 918.11	1 457.66	2 130.23
r109_21	40 521	35 279	29 267	**23 092**	**1 059.55**	2 933.92	1 461.02	2 196.95

(续)表 3　4 个算法在 Goeke 测试用例上的结果

(Continued) Tab. 3　Results of the 4 algorithms on the Goeke test case

测试	GA	RLIGA	IMACS	MRLGA	GA	RLIGA	IMACS	MRLGA
r110_21	37 021	32 006	25 901	**21 179**	**1 042.01**	2 898.65	1 497.87	2 166.56
r111_21	37 128	32 337	26 296	**21 557**	**1 074.98**	2 914.13	1 476.20	2 162.23
r112_21	34 291	29 056	23 937	**20 357**	**1 033.94**	2 883.05	1 500.00	2 166.51
r201_21	62 449	46 716	65 031	**22 276**	**752.32**	2 427.74	4 615.64	2 030.57
r202_21	51 420	38 731	49 405	**18 267**	**727.02**	2 250.42	7 368.36	1 950.99
r203_21	41 599	33 348	29 484	**16 002**	**711.84**	2 285.36	4 539.72	1 931.89
r204_21	33 294	26 227	16 525	**12 145**	**674.97**	2 196.39	8 516.52	1 880.32
r205_21	50 720	39 417	46 843	**19 282**	**710.71**	2 288.31	5 303.77	1 956.59
r206_21	44 831	35 734	38 138	**17 153**	**711.82**	2 283.67	4 639.09	1 957.63
r207_21	38 115	30 000	24 079	**13 714**	**679.32**	2 221.18	8 653.98	1 871.04
r208_21	30 854	24 563	16 546	**11 918**	**688.13**	2 207.23	6 011.87	1 902.42
r209_21	44 240	35 294	33 217	**16 933**	**715.37**	2 274.03	4 578.36	2 002.91
r210_21	42 836	34 322	32 857	**16 242**	**716.26**	2 289.10	4 590.06	1 952.64
r211_21	36 974	29 734	20 219	**14 067**	**674.80**	2 211.94	8 814.96	1 909.71
rc105C5	2 866	2 866	2 866	**2 866**	106.23	416.06	**28.04**	580.39
rc108C5	5 155	5 155	5 155	**5 155**	117.88	446.97	**26.48**	581.88
rc204C5	3 447	3 447	3 447	**3 447**	98.54	395.63	**30.25**	587.22
rc208C5	3 488	3 488	3 488	**3 488**	101.39	395.55	**24.34**	582.87
rc102C10	5 540	5 534	5 562	**5 534**	153.03	536.43	**47.14**	671.18
rc108C10	5 319	5 319	5 319	**5 319**	148.29	541.92	**47.13**	716.82
rc201C10	7 341	7 341	7 924	**7 341**	142.52	533.29	**48.43**	719.27
rc205C10	8 694	8 695	8 822	**8 694**	149.90	543.87	**47.75**	705.16
rc103C15	7 236	7 162	7 173	**7 162**	195.17	673.71	**76.81**	800.68
rc108C15	5 880	5 667	5 715	**5 663**	172.99	629.55	**83.22**	772.97
rc202C15	11 905	11 773	12 740	**11 749**	194.10	677.89	**73.56**	817.66
rc204C15	8 131	7 809	7 911	**7 723**	181.38	659.29	**94.39**	779.02
rc101_21	54 858	45 661	35 097	**25 940**	**929.10**	2 653.60	1 599.31	2 186.60
rc102_21	50 552	42 384	32 171	**24 312**	**923.32**	2 660.59	1 618.95	2 124.46
rc103_21	48 200	39 518	28 949	**22 881**	**922.44**	2 662.68	1 618.05	2 112.09
rc104_21	43 801	34 908	26 120	**20 866**	**921.20**	2 627.33	1 650.42	2 081.24
rc105_21	50 790	42 662	31 715	**24 298**	**931.58**	2 629.23	1 621.70	2 145.01
rc106_21	50 486	41 858	31 140	**23 904**	**924.98**	2 593.73	1 643.29	2 134.96
rc107_21	46 433	37 667	27 568	**22 205**	**926.51**	2 581.08	1 645.18	2 102.92
rc108_21	44 074	35 576	26 249	**21 520**	**905.86**	2 492.64	1 663.59	2 129.85
rc201_21	73 973	56 212	68 387	**26 884**	**694.63**	2 177.75	5 498.72	1 997.94
rc202_21	64 836	46 892	48 348	**21 836**	**661.06**	2 131.26	9 081.71	1 919.32
rc203_21	51 803	41 573	35 168	**20 000**	**689.81**	2 130.52	5 383.41	1 981.45
rc204_21	41 355	33 361	26 059	**16 829**	**702.60**	2 167.20	3 362.34	1 919.90
rc205_21	61 897	47 977	49 420	**21 919**	**705.76**	2 132.34	4 670.46	1 935.35

(续)表 3　4 个算法在 Goeke 测试用例上的结果

(Continued) Tab. 3　Results of the 4 algorithms on the Goeke test case

测试	GA	RLIGA	IMACS	MRLGA	GA	RLIGA	IMACS	MRLGA
rc206_21	61 635	46 456	48 124	**20 997**	**673.16**	2 111.41	6 235.11	1 962.63
rc207_21	52 098	41 726	34 479	**18 972**	**675.68**	2 114.49	5 345.38	1 980.79
rc208_21	45 277	36 848	24 424	**18 103**	**706.86**	2 099.75	3 759.07	1 927.60

注:第 1 栏表示 Average cost,单位为元; 第 2 栏表示 Time,单位为秒。

2.3　统计检验验证算法显著性

表 4　统计检验核心结果

Tab. 4　Core results of statistical tests

指标	Friedman χ^2	Iman-Davenport F	p-value	结论
最优成本	168.82	143.34	8.7×10^{-56}	四算法差异极显著
运行时间 ln(1+t)	110.71	60.95	3.4×10^{-30}	四算法差异显著

表 5　平均秩(成本,数字越小越优)

Tab. 5　Average rank (cost, smaller numbers are better)

算法	平均秩
MRLGA	1.00
IMACS	2.46
RLIGA	2.24
GA	3.29

表 6　事后 Wilcoxon (+ Holm)—运行时间

Tab. 6　Post hoc Wilcoxon (+ Holm)-running time

算法对	pHolm	均值差/s	更快者
GA-RLIGA	4.9×10^{-16}	−1 183	GA
GA-MRLGA	4.9×10^{-16}	−941	GA
IMACS-GA	3.9×10^{-8}	−1 554	IMACS
RLIGA-MRLGA	9.9×10^{-8}	+242	RLIGA(略快)
RLIGA-IMACS	0.77	−371	不显著
IMACS-MRLGA	0.069	+613	边缘差异

从上表可以看出成本上 MRLGA ≪ IMACS<RLIGA<GA,在时间上 IMACS<GA≪(RLIGA ≈MRLGA),可以得到结论,MRLGA 在求解质量上有很显著的效果,但是在时间开销上也很大。

3　算法时间复杂度分析

表 7　变量含义表

Tab. 7　Variable meaning table

记号	含义
G	迭代次数(Gens)
P	每代种群规模(PSize)
L	单条染色体长度≈客户数 n +充电站数 m +1
I	精英局部搜索循环上限(max_iterations)
V	varyOne 中的微扰次数(varyNum=10)

表 8 时间复杂度分析表

Tab. 8 Time complexity analysis table

步骤	操作	复杂度
选择	TournamentO(P) roulette:期望 O(P),最坏 O(P^2) rank:排序 O($P\log P$)	O($P\log P$)
交叉	交叉 $P/2$ 对,每对子序列操作 O(L)	O(PL)
合并与精英保存	对 P 个体排序	O($P\log P$)
多样性计算	计算 $\left(\frac{P}{2}\right)$ 对汉明距离	O(P^2L)
RL-step	Q-table 查更、奖励更新	O(1)
变异	每次遍历 P 条染色体,最坏 O(VL)	O(PVL)
局部搜索	精英比例 $e=0.1$,精英数 eP :	O($ePIL$)≈O(PL)
适应度计算	(eP)×I 次邻域操作,每次 O(L) 遍历 P 条染色体	O(PL)

综上所述每一代的时间复杂度为 O($P^2L+P\log P+PL(V+1+eI)$)≈O(P^2L),所以总的时间复杂度为 O(GP^2L),所以总时间开销基本集中在香农多样性指数的计算,因此这个部分的优化也是未来的方向。

(责任编辑:赵海军)