

DOI:10.3969/j.issn.1003-5060.2026.04.005

求解大规模车辆路径问题的超启发式算法

王子源¹, 夏维^{1,2}, 伍康¹

(1. 合肥工业大学 管理学院, 安徽 合肥 230009; 2. 过程优化与智能决策教育部重点实验室, 安徽 合肥 230009)

摘要:针对大规模车辆路径规划问题,文章提出一种基于密度空间聚类和超启发式(hyper-heuristic, HH)算法的两阶段路径规划框架。第 1 阶段根据客户点的空间位置特征,将大规模车辆路径客户点集合分解为若干个车辆路径优化子簇;第 2 阶段设计一种基于进化机制的 HH 算法进行求解。该算法采用底层的 4 种构造型算子和 6 种扰动型算子生成各簇的初始种群,再运用进化策略和模拟退火方案接收准则进行各子簇方案的全局融合优化,最后通过仿真实验在 Uchoa 标准数据集上验证了算法的有效性。

关键词:车辆路径问题;超启发式(HH)算法;进化算法;基于密度的空间聚类算法;模拟退火

中图分类号:TP18 **文献标志码:**A **文章编号:**1003-5060(2026)04-0464-09

A hyper-heuristic algorithm for solving large-scale vehicle routing problems

WANG Ziyuan¹, XIA Wei^{1,2}, WU Kang¹

(1. School of Management, Hefei University of Technology, Hefei 230009, China; 2. Key Laboratory of Process Optimization and Intelligent Decision Making, Ministry of Education, Hefei 230009, China)

Abstract: A two-stage route planning framework based on density-based spatial clustering of applications with noise(DBSCAN) and hyper-heuristic(HH) algorithm is suggested for the large-scale vehicle routing planning. The set of large-scale vehicle routing customer points is decomposed into several vehicle routing optimization sub-clusters based on the spatial location characteristics of the customer points in the first stage, and the problem is solved in the second stage using an HH algorithm based on evolutionary mechanism. The algorithm first generates the initial populations of each cluster using the underlying four conformal operators and six perturbation-type operators, and then uses the evolutionary strategy and the simulated annealing(SA) reception criterion to perform global fusion optimization of each sub-cluster solution. Finally, simulation tests on the Uchoa standard dataset verify the efficacy of the algorithm.

Key words: vehicle routing problem (VRP); hyper-heuristic (HH) algorithm; evolution algorithm; density-based spatial clustering of applications with noise(DBSCAN); simulated annealing(SA)

近年来,随着城市物流配送需求的快速增加,车辆配送路径规划问题的规模也显著增长。国家邮政局发布的《2021 年邮政行业发展统计公报》显示,全行业营业网点平均服务人口已达 0.3 万人^[1]。在此背景下,大规模车辆路径规划问题愈发引起学者关注,该问题以在可接受的计算时间内给出大规模车辆路径规划问题的满意解为目

标,而设计高效求解算法是解决大规模车辆路径问题的关键所在。

车辆路径问题(vehicle routing problem, VRP)是一类典型的 NP-hard 组合优化问题,其求解算法主要包括精确算法、启发式算法和元启发式算法等。精确算法包括分支定界法^[2]、动态规划法^[3]和列生成法^[4]等,其时间复杂度较高,难

收稿日期:2023-09-04;修回日期:2024-01-02

基金项目:国家自然科学基金资助项目(72271074)

作者简介:王子源(1999—),男,河北廊坊人,合肥工业大学硕士生;

夏维(1983—),男,安徽巢湖人,博士,合肥工业大学副教授,硕士生导师,通信作者,E-mail:xiawei@hfut.edu.cn.

以解决大规模问题;启发式算法如节约插入法^[5]、改进-交换法^[6]等,仅能够在中小规模问题实例中给出满意解,通常不具备全局搜索能力;元启发式算法如遗传算法(genetic algorithm,GA)^[7]、粒子群算法^[8]和模拟退火算法^[9]等,能够进行全局搜索跳出局部最优解,但局部搜索能力较差,且在大规模优化问题中性能会明显下降^[10]。

为了设计一种能够在较短时间内获得大规模VRP问题满意解的算法,本文设计一种基于聚类方法和超启发式(hyper-heuristic,HH)算法的求解框架。首先使用基于密度的空间聚类算法(density-based spatial clustering of applications with noise,DBSCAN)对问题进行分簇;在每个簇中利用多种启发式算子形成各簇的部分方案,将其融合并构建总问题初始解种群;最后使用GA算法进行全局搜索获得满意解。

HH算法于2000年首次应用于人员调度问题^[11],被描述为“搜索启发式算法的启发式算法”,该方法通过高层启发式(high-level heuristic,HLH)策略构建低级启发式(low-level heuristic,LLH)执行序列^[12]。近年来,HH算法逐渐成为组合优化领域的研究热点,并已在时间安排表问题^[13]、工程调度问题^[14]、车辆路径问题^[15]等领域表现了出色的性能。文献^[16]研究了将强化学习作为HH高层策略求解小规模VRP问题;文献^[15]研究了一种基于K均值聚类算法求解大规模VRP问题的HH算法,实验结果表明聚类技术能够提高HH在大规模VRP问题中的性能。在现有的文献中,HH算法尚未被广泛应用于大规模VRP问题的求解。

基于分治融合思想,本文提出一种基于密度空间聚类的超启发式遗传算法(hyper-heuristic genetic algorithms based on density-based spatial clustering of applications with noise,DBSCAN-HHGA)求解大规模VRP问题。第1阶段,利用DBSCAN聚类算法将客户点划分成空间距离较近的多个簇;第2阶段,在LLH中实现并构建LLH算子库,在HLH中将GA算法作为高层选择策略搜索底层启发式空间,构建各簇的搜索启发式,并采用模拟退火接收准则决定是否接受新邻域解,每次迭代选取最佳启发式算子搜索解空间。最后通过仿真实验证明DBSCAN-HHGA算法在大规模VRP问题上的有效性。

1 问题模型

VRP问题包括带容量约束的车辆路径问题

(capacitated vehicle routing problem,CVRP)、带时间窗的车辆路径问题和多车场的车辆路径问题等多种变种问题。CVRP作为VRP的一个基本问题,针对其开发的算法可推广到其他类型的VRP问题求解中,因此为了方便介绍所提出的方法及其在大规模问题中的性能,本文基于CVRP问题进行建模和求解。

1.1 问题描述

CVRP问题可描述为:在一个无向图 $G=(V,A)$ 中, V 为图中所有节点集合, B 为顾客节点集合, $V-B=\{v_0|v_0\in V\text{且}v_0\notin B\}$, v_0 表示配送中心; A 为任意两节点的边, $A=\{(i,j)|i,j\in V,i\neq j\}$ 表示边集合,每条边的权重 $c_{ij}(i,j\in V,i\neq j)$ 表示从顾客节点 i 到顾客节点 j 的配送距离。 $K=\{k_1,\dots,k_m\}$ 表示车辆集合,每辆车的最大容量约束为 Q ,每个顾客节点 i 的需求为 d_i 且 $d_i<Q$ 。CVRP问题具有以下假设:

- 1) CVRP问题中所有的信息均是确定的,顾客的服务类型仅为取送货,提供服务的车辆类型与容量约束相同;
- 2) 每个顾客节点只能被访问1次,且需求量均不大于车辆的负载量;
- 3) 车辆的起点和终点均为配送中心,并在行驶距离最小化的前提下经过所有顾客节点。

1.2 数学模型

CVRP问题数学模型如下:

$$\min F = \sum_{i\in V} \sum_{j\in V} \sum_{k\in K} c_{ij} x_{ijk} \quad (1)$$

$$\sum_{i\in V} x_{0ik} = 1, \quad k \in K \quad (2)$$

$$\sum_{i\in V} x_{ilk} = \sum_{j\in V} x_{ljk}, \quad \forall l \in V, k \in K \quad (3)$$

$$\sum_{j\in V} x_{j0k} = 1, \quad k \in K \quad (4)$$

$$\sum_{k\in K} \sum_{i\in V} x_{ijk} = 1, \quad \forall j \in K \quad (5)$$

$$\sum_{i\in V} \sum_{j\in B} x_{ijk} d_i \leq Q, \quad k \in K \quad (6)$$

$$\mu_i - \mu_j + N x_{ij} \leq N - 1, \quad \forall i, j \in V, \quad i, j \neq 0, \quad i \neq j \quad (7)$$

$$x_{ijk} \in \{0, 1\}, \mu_i \geq 0, \mu_i \in \mathbf{R}, \quad \forall i, j \in V, \quad k \in K \quad (8)$$

式(1)表示最小化配送的总距离的目标函数;式(2)~(4)确保每辆车从配送中心出发,服务顾客后必须离开,并最终返回配送中心;式(5)确保每个顾客只能被1辆车服务1次;式(6)确保每一条配送路径上顾客的总需求不超过车辆的最大负载量;式(7)消除子回路;式(8)为决策变量。

2 DBSCAN-HHGA 算法

本文所提出的 DBSCAN-HHGA 算法将从以下方面进行设计:① DBSCAN 算法设计;② 高层策略域构造;③ 底层问题域设计;④ 编码设计;⑤ 算法流程。

DBSCAN-HHGA 算法架构图如图 1 所示。

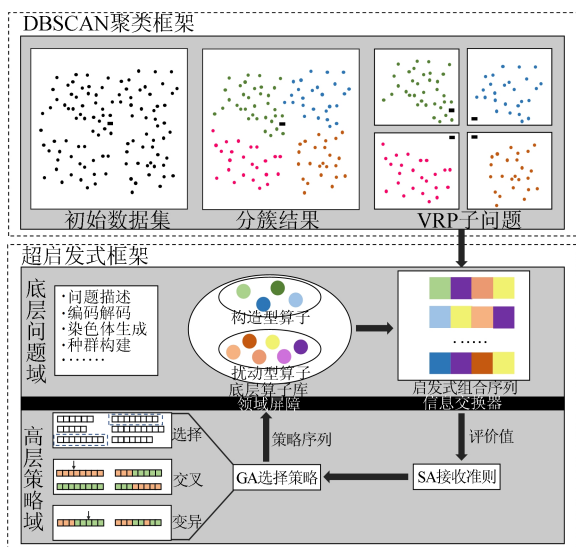


图 1 DBSCAN-HHGA 算法架构图

图 1 中:上半部分为 DBSCAN 聚类框架,显示了将初始数据集划分为 4 个子簇的可视化过程;下半部分为超启发式框架,其内部分为 HLH 和 LLH。HLH 将底层算子库的启发式组合序列通过信息交换器发送给 LLH,LLH 按照策略序列优化子路径解,并将其适应度函数值作为 GA 算法中染色体的评价价值传给 HLH,以此判断底层启发式组合序列的优劣。

2.1 自适应 DBSCAN 算法

在求解大规模 VRP 问题中,聚类能够有效地将数据划分为若干个数据子集,进而提高后续 HHGA 的求解速度和解的质量。城市人口在地理分布上具有高度的集中性,因此 VRP 问题顾客节点天然可以根据坐标点进行聚类。DBSCAN 需要确定邻域半径 ϵ 和邻域内最少点数目 MinPts 这 2 个参数,在一个邻域半径 ϵ 内包含 MinPts 个数的邻域可认为是一个簇。相较于常用的 K 均值聚类算法,DBSCAN 可同时适用于任何形状分布的数据且不需要事先指定聚类簇数。

自适应 DBSCAN 算法通过对数据集统计特性的分析,自适应确定 ϵ 和 MinPts 这 2 个参数。

2.2 高层策略域构造

2.2.1 选择机制

在主流的超启发式框架中^[17],选择 LLH 和移动接收准则作为连续的过程迭代执行,直到满足终止条件。DBSCAN-HHGA 将 GA 算法作为底层启发式的选择器。以下介绍 GA 算法的初始化种群过程、选择算子、交叉算子以及变异算子。

1) 初始化种群。GA 算法将底层启发式策略序列求解 VRP 子问题所得的 N_H 个子路径解作为染色体,并使用随机的方法产生 N_R 个染色体以初始化种群,种群大小 $N_{POP} = N_H + N_R$ 。

2) 选择。所谓选择,即基于种群中染色体的适应度评估,选取适应度值高的染色体直接遗传至下一代,或选择染色体通过交叉和变异产生新的染色体遗传至下一代。本文设计了基于精英策略与轮盘赌相结合的选择策略,该方法将种群中固定比例的最优染色体直接复制到下一代,余下的染色体通过轮盘赌法进行选择,染色体的适应值越高,在轮盘中的比例越大,选择的概率也就越高。每条染色体被选择的概率为:

$$P_i = F_i / \sum_{i=1}^N F_i \quad (9)$$

其中: P_i 为选择第 i 条染色体的概率; F_i 为第 i 条染色体的适应度值; N 为种群大小。

3) 交叉。本文设计了简单随机交叉和最大覆盖交叉 2 种交叉算子。简单随机交叉将父代 P_A 和 P_B 中的一段基因位互换,从而形成 2 个新的染色体;最大覆盖交叉将基因位的相对位置和载重约束考虑在内,将父代 P_A 中的随机子路径插入到另一父代 P_B 中形成子代染色体,该交叉的最优决策通过边界框重叠计算。

4) 变异。变异算子有助于保证种群多样性,提升算法全局搜索能力。本文采用自适应变异算子控制随机变异^[18],其过程为首先生成一个随机数 $r \in (0, 1)$,若 $r < P_{MO}$,则随机改变该染色体上某个基因位上的值,形成新的子代染色体。自适应变异算子 P_{MO} 如下所示:

$$P_{MO} = 0.1 \times \left(\frac{\alpha I}{I_{max}} \right) \quad (10)$$

其中: α 为变化速率; I_{max} 为总的迭代次数; I 为当前迭代次数。随着迭代次数的增加,变异的概率也会随之增大。这一设计可以防止在算法后期,种群内部过度相似而陷入局部最优。

2.2.2 接收机制

DBSCAN-HHGA 将模拟退火 (simulated

annealing, SA)作为接收准则,对算法收敛速度及寻优性能起重要作用^[19]。该接收准则将解的相对质量和温度作为选择较差解的概率,以改善对搜索空间的探索,是避免算法陷入局部最优停滞的关键。若子代解优于父本,则接收子代解替换父本进入下一次迭代,否则以概率接收子代解,即

$$P = \begin{cases} e^{-\frac{\Delta F}{T}}, & F_{\text{new}} < F_{\text{old}}; \\ 1, & F_{\text{new}} \geq F_{\text{old}} \end{cases} \quad (11)$$

其中: F_{new} 为子代解的适应度值; F_{old} 为父代解的适应度值; ΔF 为子代解与父本的适应度值之差; T 为当前温度。

2.3 底层问题域设计

底层问题域中的 LLH 算子需要根据实际问题进行设计,本文依据文献[20]的分类标准,将 LLH 算子划分为构造型 LLH 和扰动型 LLH。底层 LLH 算子列表见表 1 所列。

表 1 底层 LLH 算子列表

算子编号	类型	简介
LLH-C-1	构造型算子	并行节约算法 ^[21] ,其核心思想是基于节约值构造路径
LLH-C-2	构造型算子	Cawlip 节约算法 ^[22] ,即在并行节约算法后增加一个后优化步骤,以此来提高所生成解的质量
LLH-C-3	构造型算子	插入算法 ^[23] ,将非路由客户插入新路由上的任意 2 个连续节点之间
LLH-C-4	构造型算子	最小代价插入法 ^[23] ,该方法指定起点后寻找代价最小节点插入路径,直至形成闭合的初始路径
LLH-P-1	线路内扰动型算子	2-opt,选择单条路由,随机删除内部 2 条边后依次连接两端节点,2 条边内的中间节点顺序反转
LLH-P-2	线路内扰动型算子	Relocate,选择单条路由,随机选择一个移出和插入节点,将移除节点插入到插入节点所在位置
LLH-P-3	线路内扰动型算子	Exchange,选择单条路由,随机选择任意 2 个节点进行交换
LLH-P-4	线路间扰动型算子	2-opt*,选择 2 条路由,各随机删除 1 条边后依次连接两端节点
LLH-P-5	线路间扰动型算子	Relocate,选择 2 条路由,随机选择其中一条路线上的节点插入另一条路线的各个位置
LLH-P-6	线路间扰动型算子	Exchange,选择 2 条路由,各随机选择 1 个节点进行交换

算子示意图以 LLH-P-3 和 LLH-P-6 为例,如图 2 所示。图 2a 为路线内 Exchange 算子示意图,在原路由中选择节点 1 和节点 4 交换位置,形成 1 条新路由;图 2b 为路线间 Exchange 算子示意图,在 2 条原路由中选择节点 2 和节点 7 交换位置,形成 2 条新路由。

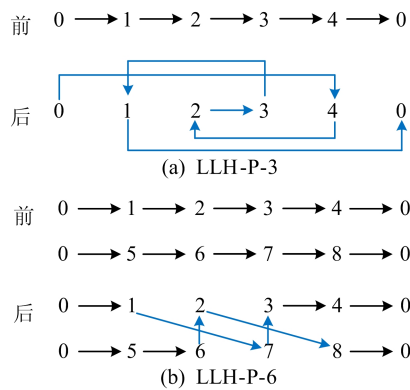


图 2 LLH 算子示意图

2.4 编码设计

在高层控制域中,由底层问题域 LLH 算子构成染色体的每个基因^[24],同一染色体中可包含相同的 LLH 算子,染色体长度为最大长度 4 的非定长自然数列。高层控制域染色体示意图如图 3 所示。在解码高层控制域染色体时,将编码转换为相对应的 LLH 算子并从左向右依次执行。

编码	2	7	10	8
对应算子	LLH-C-2	LLH-P-3	LLH-P-6	LLH-P-4

图 3 高层控制域染色体编码结构示意图

在底层问题域中,染色体编码分为 2 个部分,其结构如图 4 所示。

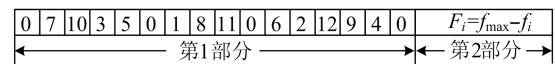


图 4 底层问题域染色体编码结构示意图

第 1 部分采用基于客户顺序的编码方式,一个完整的解可表示为所有路线的集合 $R = \{r_1, r_2, \dots, r_K\}$,每条车辆行驶路径 r_i 两端均为配送中心,中间段为客户节点,编码顺序即为访问顺序。在车辆规模 K 和客户点规模 N 条件下,染色体长度为 $K + N + 1$ 。以一条可能的染色体 $\{0, 7, 10, 3, 5, 0, 2, 8, 11, 0, 6, 2, 12, 9, 4, 0\}$ 为例,其中客户节点集合以配送中心 0 为分隔符,该染色体表示车辆 1 由配送中心 0 出发,经由客户节点 7、10、3、5 后回到配送中心 0,车辆 2、3 同理。第 2 部分记录该条染色体的适应度函数值,适应度值越大越容易保留,由于本文的目标函数是配送总距离最小,适应度函数可表示为:

$$F_i = f_{\text{max}} - f_i \quad (12)$$

其中: F_i 表示第 i 条染色体的适应度计算函数; f_{\max} 表示一个很大的正数, 可以取目标函数的一个上界或取当前种群中的最大值; f_i 为第 i 条染色体的目标函数值。

2.5 算法流程

本文所提出的 DBSCAN-HHGA 算法伪代码描述如下。

输入: 聚类簇划分 $C = \{C_1, C_2, \dots, C_k\}$, LLH 算子集合 $L = \{L_1, L_2, \dots, L_i\}$

输出: 种群

```

初始化种群
设定初始温度
for  $C_i \in C$  do
  while 迭代次数 < 2 000 次 do
    从  $L$  中选择构造型启发式  $L_i$  生成初始解  $s_i$ 
    从  $L$  中选择构造型启发式  $L_i$  生成初始解  $s_i'$ 
    对解  $s_i'$  进行选择、交叉、变异构造解  $s_i''$ 
    应用模拟退火接受准则决定是否接受解  $s_i''$ 
    if 解  $s_i''$  优于解  $s_i'$  or 大于接收概率  $P$  then
       $s_i = s_i''$ 
    end if
    降温
  end while
end for
  
```

3 实验结果与结果分析

3.1 测试算例

算法采用 Python3.7 编程实现, 在 i7-10700F CPU、16 GiB RAM、Windows10 操作系统的计算机上运行。本文选取 Uchoa^[25] 提出的 20 个算例进行测试, 算例数据可在 CVRPLIB 实例库 (<http://vrp.atd-lab.inf.puc-rio.br/index.php/en>) 中查看。算法具体参数设置见表 2 所列。

表 2 参数设置

参数	含义	取值
I_{\max}	迭代次数	2 000
N_{POP}	种群规模	200, 400, 600
P_{CO}	交叉概率	0.6
α	变异变化速率	15
t	初始温度	50
β	降温速率	0.9

3.2 仿真实验与数值分析

3.2.1 算法优化性能对比分析

为了验证 DBSCAN-HHGA 算法的有效性, 分别采用 DBSCAN-HHGA、HHGA 和 GA 算法重复求解 Uchoa 标准算例 20 次, 实验结果见表 3 所列。

表 3 3 种算法求解 Uchoa 标准算例的实验结果

算例	BKS	T/min	DBSCAN-HHGA			HHGA			GA		
			BF	GAP/%	T/min	BF	GAP/%	T/min	BF	GAP/%	T/min
X-n153-k22	21 220	37.3	22 644	6.71	13.1	22 335	5.25	12.6	22 562	6.32	10.8
X-n200-k36	58 578	901.7	62 579	6.83	15.8	61 689	5.31	15.1	63 091	7.70	13.4
X-n251-k28	38 684	4 767.0	40 986	5.95	18.1	40 941	5.83	17.3	41 631	7.62	15.4
X-n322-k28	29 834		31 591	5.89	24.2	31 749	6.42	23.4	32 325	8.35	21.2
X-n376-k94	147 713	3.3	154 471	4.58	8.4	156 999	6.29	7.8	161 427	9.28	5.9
X-n401-k29	66 154		69 576	5.17	23.6	70 532	6.62	22.3	72 102	8.99	18.9
X-n449-k29	55 233		58 284	5.52	26.7	58 918	6.67	25.2	59 929	8.50	20.8
X-n502-k39	69 226		73 062	5.54	28.1	74 059	6.98	26.2	75 102	8.49	21.7
X-n524-k153	154 593	212.1	161 525	4.48	21.6	165 293	6.92	19.8	169 396	9.56	15.2
X-n573-k30	50 673		53 037	4.67	34.8	54 256	7.07	31.7	55 283	9.10	27.1
X-n613-k62	59 535		62 450	4.90	32.9	64 120	7.70	30.9	65 152	9.43	25.8
X-n641-k35	63 682		66 580	4.55	37.4	68 514	7.59	35.2	69 625	9.33	32.8
X-n670-k130	146 332		152 577	4.27	32.2	158 656	8.42	29.4	160 553	9.72	24.4
X-n701-k44	81 923		85 708	4.62	42.8	88 598	8.15	38.9	89 623	9.40	33.7
X-n749-k98	77 269		80 604	4.32	39.7	83 263	7.76	36.4	84 772	9.71	31.9
X-n801-k40	73 305		76 370	4.18	49.5	79 037	7.82	46.2	80 841	10.28	39.5
X-n856-k95	88 965		92 397	3.86	52.4	96 002	7.91	48.7	98 342	10.54	41.6
X-n916-k207	329 179		342 965	4.19	63.5	355 908	8.12	58.1	366 179	11.24	51.4
X-n957-k87	85 465		88 552	3.61	72.8	92 045	7.70	67.4	94 627	10.72	60.7
X-n1001-k43	72 355		75 482	4.32	93.9	77 987	7.78	89.1	80 683	11.51	79.5

表 3 中: n 为客户节点数目, k 为车辆数目; T 为各算法运行时间; BKS 表示已知最优解; BF 表

示各算法搜索到的最优解; GAP 表示 BF 与 BKS 的差距。

由表 3 可知:DBSCAN-HHGA 算法在 20 个算例中的 17 个中取得 3 个算法中的最好表现,由此可以验证本文所提出的 DBSCAN-HHGA 算法的有效性;同时在全部的 20 个算例中,DBSCAN-HHGA 与 HHGA 算法均优于 GA 算法,可以推断本文所提出的 HH 框架在求解大规模 CVRP 问题中的有效性。在前 3 组算例中 DBSCAN-HHGA 算法所得解劣于 HHGA 算法,原因在于其数据集规模较小,通过 DBSCAN 将数据集强制分簇可能会使部分最优路径被破坏,影响解的质量。

在算法运行时间上,Uchoa^[25]运用精确算法中的分支定价法对算例进行求解,求解时间参见表 3。从表 3 可以看出:当节点规模 $n < 200$ 时,分支定价法能够在可接受的时间内计算出精确

解;当节点规模 $200 \leq n \leq 250$ 时,分支定价法虽仍能计算出精确解,但计算时间十分庞大;当节点规模 $n > 250$ 时,分支定价法在绝大多数算例中已经无法在可计算的时间内得出精确解。

由此,对于大规模 CVRP 问题的求解,本文所提出的 DBSCAN-HHGA 算法能够在可观的时间内计算出节点规模 $n > 250$ 算例的相对满意解,具有可行性。

3.2.2 算法计算时间对比分析

为了验证 DBSCAN-HHGA 算法的可行性,将其与 HHGA、GA 算法进行迭代速率和运行时间的对比实验,实验在具有代表性的 5 组算例中进行。

3 种算法的最小行驶距离适应度曲线如图 5a~图 5e 所示,平均运行时间如图 5f 所示。

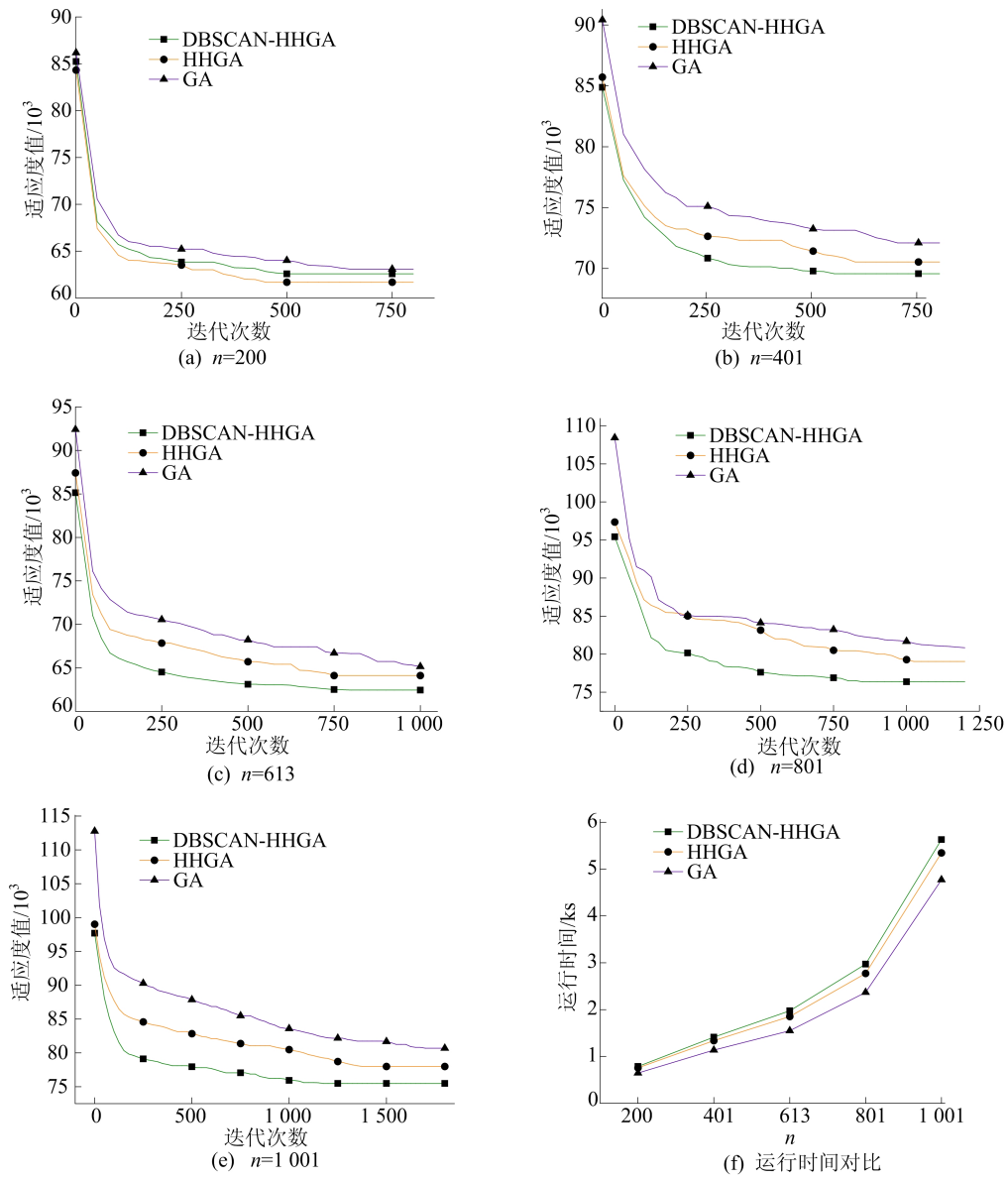


图 5 3 种算法迭代速率与求解时间的对比

由图 5a~图 5e 可知, DBSCAN-HHGA 算法迭代速率及收敛时间快于 HHGA 和 GA 算法, 且优化目标值更小。由图 5f 可知, DBSCAN-HHGA 算法求解时间高于其他对比算法, 原因在于对输入数据集进行聚类分簇需要耗费一定的时间。但车辆路径规划在时间线上先于配送, 时效性相对较弱, 因此可接受该运行时间。

3.2.3 底层算子效用分析

为了进行验证, 对 5 组代表性算例重复实验

20 次。在底层算子效用分析中, 采用平均优化提升和平均运行时间这 2 个维度。平均优化提升为仅调用构造型算子与同时调用单一扰动型算子的适应度值的 GAP 值, 总运行时间为构造型算子与扰动型算子运行时间之和, 扰动型算子为空白表示仅调用构造型算子。实验结果见表 4 所列。

底层算子评分表见表 5 所列, 其中运行时间得分按正序进行赋分, 优化提升得分按倒序进行赋分。

表 4 底层算子平均优化提升率和总平均运行时间

构造型 算子	扰动型 算子	平均优化提升率/%					总平均运行时间/s				
		200	401	613	801	1 001	200	401	613	801	1 001
LLH-C-1	LLH-P-1	0.098	0.238	0.344	0.564	0.497	0.087	0.547	1.341	2.269	3.631
	LLH-P-2	0.098	0.212	0.381	0.656	0.523	0.089	0.569	1.357	2.322	3.687
	LLH-P-3	0.098	0.208	0.318	0.393	0.389	0.089	0.576	1.344	2.325	3.753
	LLH-P-4	0	0.136	0.043	0.036	0.022	0.176	1.371	1.672	4.716	7.607
	LLH-P-5	0.240	0.704	0.083	0	0.115	0.198	1.009	1.595	2.276	4.230
	LLH-P-6	0.151	0.323	0.065	0.321	0.045	0.106	0.733	3.039	6.698	4.362
LLH-C-2	LLH-P-1	0.098	0.873	0.370	0.602	0.540	0.180	1.366	3.039	4.652	7.544
	LLH-P-2	0.098	0.647	0.431	0.727	0.566	0.194	1.752	3.796	8.206	13.676
	LLH-P-3	0.098	0.559	0.344	1.744	0.406	0.196	1.712	4.142	8.368	13.976
	LLH-P-4	0	0	0	0	0	0.193	1.646	3.749	7.650	13.558
	LLH-P-5	0.071	0.160	0.078	0	0.204	0.286	1.984	3.928	5.671	8.648
	LLH-P-6	0.151	0.319	0.03	0.326	0.035	0.255	1.972	3.844	4.726	10.502
LLH-C-3	LLH-P-1	-0.144	-0.041	0.434	0.301	0.638	10.131	30.703	58.457	155.852	277.696
	LLH-P-2	-0.144	0	0.618	0.777	0.852	10.273	33.085	59.311	157.738	279.169
	LLH-P-3	0	0	0.545	0.148	0.788	10.158	33.738	60.402	156.901	281.423
	LLH-P-4	0.995	0.042	0.305	0.085	0.899	10.269	31.413	59.437	157.218	277.734
	LLH-P-5	-0.144	-0.055	-1.003	0.193	0.250	10.272	35.660	61.345	160.541	278.713
	LLH-P-6	-0.113	0.045	-0.795	0.712	1.190	10.223	34.113	59.574	156.823	278.185
LLH-C-4	LLH-P-1	16.327	36.082	31.348	17.161	22.557	10.214	33.077	59.205	162.520	278.015
	LLH-P-2	17.410	34.292	29.132	16.162	22.424	0.176	1.521	3.374	7.767	11.712
	LLH-P-3	13.457	21.887	24.407	10.009	12.620	0.198	1.622	3.473	8.055	12.158
	LLH-P-4	27.964	27.217	9.910	0.102	1.423	0.214	1.727	3.485	8.320	12.286
	LLH-P-5	28.970	34.814	19.238	0	1.795	0.197	1.612	3.415	8.041	12.167
	LLH-P-6	8.918	3.081	3.731	0.549	0.235	0.542	21.125	15.271	11.479	21.509
											14.334
											12.221

表 5 底层算子评分表

LLH-C	T_C	S_C	LLH-P	O_P	S_{POI}	T_P	S_{PT}	S_P
LLH-C-1	1.575	1	LLH-P-1	6.444	1	0.934	2	3
LLH-C-2	3.356	2	LLH-P-2	6.293	2	1.151	4	6
LLH-C-3	106.568	4	LLH-P-3	4.421	3	0.717	1	4
LLH-C-4	4.910	3	LLH-P-4	3.459	5	3.524	6	11
			LLH-P-5	4.286	4	1.108	3	7
			LLH-P-6	0.966	6	1.760	5	11

表 5 中: T_C 和 T_P 分别表示 LLH-C 和 LLH-P 的平均运行时间; S_C 和 S_{PT} 分别表示 LLH-C 和

LLH-P 的运行时间得分; O_P 表示 LLH-P 平均优化提升; S_{POI} 表示 LLH-P 优化提升得分; S_P 表示

LLH-P 总得分。由表 5 可知:在构造型算子中,表现最好的为 LLH-C-1,LLH-C-3 在大规模问题时构造初始解时间成本较高;在扰动型算子中,线路内算子效用较线路间算子普遍更好,LLH-P-1 表现最为突出。

3.2.4 统计分析

选取具有代表性的 5 组算例重复实验 20 次,通过变异系数验证算法的可靠性,统计分析结果见表 6 所列,变异系数的计算公式如下:

$$V_{\sigma} = \sigma / \bar{x} \quad (13)$$

其中: σ 为 20 次实验结果的标准差; \bar{x} 为 20 次实验结果的平均值。

变异系数能够较好地反映出本文所提算法在求解大规模 CVRP 问题的波动程度。变异系数越小,算法所得解越稳定可靠;变异系数越大,算法可靠性越差。

变异系数计算结果见表 6 所示。由表 6 数据可知,5 组算例之间标准差差异较大,但变异系数相差不大且均小于 3%。这说明使用变异系数衡量算法可靠性,能够消除算例规模对于结果的影响,从而较好地反映出算法的波动性,其结果也说明了本文所提算法具有较好的可靠性。

表 6 变异系数计算结果

算例	\bar{x}	σ	$V_{\sigma}/\%$
X-n200-k36	63 939.40	941.14	1.47
X-n401-k29	72 054.60	1 341.52	1.86
X-n613-k62	64 052.40	1 350.93	2.11
X-n801-k40	78 823.20	1 519.17	1.93
X-n1001-k43	78 262.60	1 808.99	2.31

4 结 论

本文研究了带容量约束的大规模车辆路径问题 CVRP,目标函数为最小化车辆行驶距离。在此基础上,本文提出了一种基于密度空间聚类 and HH 算法的两阶段路径规划框架对上述问题进行求解,最后,基于 Uchoa 的标准数据集对所提 DBSCAN-HHGA 算法的性能进行了验证。结论如下:

1) DBSCAN-HHGA 算法与 HHGA 算法、GA 算法的对比实验结果表明,DBSCAN-HHGA 算法具有更好的优化效果。

2) 底层算子对比的实验结果表明,LLH-C-1 和 LLH-P-1 在底层算子中具有更好的优化效果。

[参 考 文 献]

- [1] 国家邮政局. 2021 年邮政行业发展统计公报[EB/OL]. (2022-06-09)[2023-04-12]. <https://www.spb.gov.cn/gjyzj/c100276/202206/d58b4f4fb1414177a6988b6054e40529.shtml>.
- [2] ALI O, CÔTÉ J F, COELHO L C. Models and algorithms for the delivery and installation routing problem[J]. *European Journal of Operational Research*, 2021, 291(1): 162-177.
- [3] YU G, YANG Y. Dynamic routing with real-time traffic information[J]. *Operational Research*, 2019, 19: 1033-1058.
- [4] COSTA L, CONTARDO C, DESAULNIERS G. Exact branch-price-and-cut algorithms for vehicle routing[J]. *Transportation Science*, 2019, 53(4): 946-985.
- [5] CHRISTIAENS J, VANDEN B G. Slack induction by string removals for vehicle routing problems[J]. *Transportation Science*, 2020, 54(2): 417-433.
- [6] DONDO R G, CERDÁ J. A hybrid local improvement algorithm for large-scale multi-depot vehicle routing problems with time windows[J]. *Computers & Chemical Engineering*, 2009, 33(2): 513-530.
- [7] RABBOUCH B, SAËDAOUI F, MRAIHI R. Efficient implementation of the genetic algorithm to solve rich vehicle routing problems[J]. *Operational Research*, 2021, 21: 1763-1791.
- [8] DRIDI I H, ALAÏA E B, BORNE P, et al. Optimization of m-MDPDPTW using the continuous and discrete PSO[J]. *Studies in Informatics and Control*, 2019, 28(3): 289-298.
- [9] HUO L, ZHU J, WU G, et al. A novel simulated annealing based strategy for balanced UAV task assignment and path planning[J]. *Sensors*, 2020, 20(17): 4769.
- [10] BURKE E K, HYDE M, KENDALL G, et al. A classification of hyper-heuristic approaches[M]//*Handbook of metaheuristics*. Boston: Springer, 2010: 449-468.
- [11] COWLING P, KENDALL G, SOUBEIGA E. A hyperheuristic approach to scheduling a sales summit[C]//*Practice and Theory of Automated Timetabling III: Third International Conference*. Heidelberg: Springer, 2001: 176-190.
- [12] BURKE E K, GENDREAU M, HYDE M, et al. Hyper-heuristics: a survey of the state of the art[J]. *Journal of the Operational Research Society*, 2013, 64(12): 1695-1724.
- [13] BURKE E K, MCCOLLUM B, MEISELS A, et al. A graph-based hyper-heuristic for educational timetabling problems[J]. *European Journal of Operational Research*, 2007, 176(1): 177-192.
- [14] 黄子钊, 庄子龙, 滕浩, 等. 自动化码头出口箱箱位分配优化超启发式算法[J]. *计算机集成制造系统*, 2022, 28(8): 2619-2632.
- [15] COSTA J G C, MEI Y, ZHANG M. Cluster-based hyper-heuristic for large-scale vehicle routing problem[C]//*2020 IEEE Congress on Evolutionary Computation (CEC)*. [S. l.]: IEEE, 2020: 1-8.

(下转第 481 页)

- multisensor multiresolution image fusion[J]. IEEE Transactions on Geoscience and Remote Sensing, 1999, 37(3): 1212-1226.
- [6] WU M Q, HUANG W J, NIU Z, et al. Generating daily synthetic Landsat imagery by combining Landsat and MODIS data[J]. Sensors, 2015, 15(9): 24002-24025.
- [7] AMORÓS-LÓPEZ J, GÓMEZ-CHOVA L, ALONSO L, et al. Multitemporal fusion of Landsat/TM and ENVISAT/MERIS for crop monitoring [J]. International Journal of Applied Earth Observation and Geoinformation, 2013, 23: 132-141.
- [8] SONG H H, LIU Q S, WANG G J, et al. Spatiotemporal satellite image fusion using deep convolutional neural networks[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, 11(3): 821-829.
- [9] TAN Z Y, DI L P, ZHANG M D, et al. An enhanced deep convolutional model for spatiotemporal image fusion[J]. Remote Sensing, 2019, 11(24): 2898.
- [10] TAN Z Y, GAO M L, LI X H, et al. A flexible reference-insensitive spatiotemporal fusion model for remote sensing images using conditional generative adversarial network [J]. IEEE Transactions on Geoscience and Remote Sensing, 2021, 60: 1-13.
- [11] SONG Y Y, ZHANG H Y, HUANG H, et al. Remote sensing image spatiotemporal fusion via a generative adversarial network with one prior image pair[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-17.
- [12] HUANG B, SONG H H. Spatiotemporal reflectance fusion via sparse representation[J]. IEEE Transactions on Geoscience and Remote Sensing, 2012, 50(10): 3707-3716.
- [13] SONG H H, HUANG B. Spatiotemporal satellite image fusion through one-pair image learning[J]. IEEE Transactions on Geoscience and Remote Sensing, 2012, 51(4): 1883-1896.
- [14] WU B, HUANG B, ZHANG L P. An error-bound-regularized sparse coding for spatiotemporal reflectance fusion [J]. IEEE Transactions on Geoscience and Remote Sensing, 2015, 53(12): 6791-6803.
- [15] WEI J B, WANG L Z, LIU P, et al. Spatiotemporal fusion of remote sensing images with structural sparsity and semi-coupled dictionary learning [J]. Remote Sensing, 2016, 9(1): 21.
- [16] LIU X, DENG C W, ZHAO B J. Spatiotemporal reflectance fusion based on location regularized sparse representation [C]//2016 IEEE International Geoscience and Remote Sensing Symposium. [S. l.]: IEEE, 2016: 2562-2565.
- [17] HOU B, ZHOU K, JIAO L C. Adaptive super-resolution for remote sensing images based on sparse representation with global joint dictionary model[J]. IEEE Transactions on Geoscience and Remote Sensing, 2017, 56(4): 2312-2327.
- [18] IRANI M, PELEG S. Improving resolution by image registration[J]. CVGIP: Graphical Models and Image Processing, 1991, 53(3): 231-239.
- [19] ZHU X L, HELMER E H, GAO F, et al. A flexible spatiotemporal method for fusing satellite images with different resolutions [J]. Remote Sensing of Environment, 2016, 172: 165-177.

(责任编辑 胡亚敏)

(上接第 471 页)

- [16] 张景玲, 冯勤炳, 赵燕伟, 等. 基于强化学习的超启发算法求解有容量车辆路径问题[J]. 计算机集成制造系统, 2020, 26(4): 1118-1129.
- [17] ZHANG F, MEI Y, NGUYEN S, et al. Multitask genetic programming-based generative hyperheuristics: a case study in dynamic scheduling[J]. IEEE Transactions on Cybernetics, 2022, 52(10): 10515-10528.
- [18] 李尚函, 胡蓉, 钱斌, 等. 超启发式遗传算法求解模糊柔性作业车间调度 [J]. 控制理论与应用, 2020, 37(2): 316-330.
- [19] 南丽君, 陈彦如, 张宗成. 改进的自适应大规模邻域搜索算法求解动态需求的混合车辆路径问题[J]. 计算机应用研究, 2021, 38(10): 2926-2934.
- [20] PILLAY N, QU R. Hyper-heuristics: theory and applications[M]. Cham, Switzerland: Springer, 2018.
- [21] CLARKE G, WRIGHT J W. Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations research, 1964, 12(4): 568-581.
- [22] ROBBINS J A, TURNER W C. Cawlip Clark and Wright-Lin interchange program for vehicle routing problems[J]. Computers & Industrial Engineering, 1979, 3(1): 89-100.
- [23] MOLE R H, JAMESON S R. A sequential route-building algorithm employing a generalised savings criterion [J]. Journal of the Operational Research Society, 1976, 27(2): 503-511.
- [24] 胡小建, 杨智. 基于混合遗传算法的多拣货小车路径规划研究[J]. 合肥工业大学学报(自然科学版), 2022, 45(12): 1715-1722.
- [25] UCHOA E, PECIN D, PESSOA A, et al. New benchmark instances for the capacitated vehicle routing problem[J]. European Journal of Operational Research, 2017, 257(3): 845-858.

(责任编辑 胡亚敏)