

改进混合萤火虫算法求解 CVRP

白雪媛¹, 张磊¹, 李琳¹, 武文喆²

(1. 沈阳航空航天大学理学院, 辽宁沈阳 110136;

2. 沈阳航空航天大学电子信息工程学院, 辽宁沈阳 110136)

摘要:提出一种改进混合萤火虫算法(KM-HFA)来解决带容量约束的车辆路径问题。该算法利用K-Means聚类方法将客户集先进行分类,再构建初始解,以较好的初始解开始萤火虫算法的寻优过程,减少了算法的计算量。在萤火虫算法中引入部分匹配交叉算子,2H-opt交换算子,局部搜索算子和变异算子,这些方法加快了算法的收敛速度,提高了萤火虫算法跳出局部最优的能力。选取小规模及中规模数据集进行仿真实验,共94组标准算例。对于79组实例,KM-HFA得到的解优于对照的混合萤火虫算法和CC-CVRP所得的求解方案,KM-HFA所求方案的车辆行驶总距离更小。KM-HFA计算了5组小规模实例,即A-n33-k6, A-n37-k6, P-n16-k8, P-n19-k2和P-n20-k2,在不增加车辆配送路径数目的情况下,得到比经典解更好的配送方案。对于实例P-n22-k8和P-n23-k8,文中算法在比经典解路径数增加了一条的前提下,找到了车辆行驶总距离更小的解。仿真实验结果表明KM-HFA具有较好的稳定性和有效性。

关键词:带容量约束车辆路径问题;改进混合萤火虫算法;K-Means聚类;局部搜索算子;交叉和变异算子

中图分类号:TP18;U116.2

文献标识码:A

文章编号:1673-629X(2023)12-0207-08

doi:10.3969/j.issn.1673-629X.2023.12.029

Improved Hybrid Firefly Algorithm for Solving CVRP

BAI Xue-yuan¹, ZHANG Lei¹, LI Lin¹, WU Wen-zhe²

(1. School of Science, Shenyang Aerospace University, Shenyang 110136, China;

2. School of Electronic Information Engineering, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: An improved hybrid firefly algorithm (KM-HFA) is proposed to solve the capacitated vehicle routing problem. It uses K-Means clustering to classify the customers set first, and then constructs the initial solutions, which starts the optimization process of the firefly with a better initial solution, reducing the calculation of the algorithm. In the firefly algorithm, partial matching crossover operator, 2H-opt exchange operator, local search operator and mutation operators are introduced to accelerate the convergence speed and improve the ability of the algorithm to jump out of the local optimum. In this paper, 94 instances of small-scale and medium-scale data sets are selected for simulation experiments. For 79 instances, the solution obtained by KM-HFA was significantly superior over the hybrid firefly algorithm and CC-CVRP, and the total distance of vehicles driving obtained by KM-HFA is smaller. When KM-HFA calculating 5 groups of small-scale instances: A-n33-k6, A-n37-k6, P-n16-k8, P-n19-k2 and P-n20-k2, KM-HFA gets a better delivery solution than the best-known solution without increasing the number of delivery routes. When KM-HFA calculating the instances P-n22-k8 and P-n23-k8, under the premise that the number of routes is increased by one more than the best-known solution, a solution with a smaller total distance of vehicles is found. Simulation experimental results show that the KM-HFA has good stability and effectiveness.

Key words: capacitated vehicle routing problem; improved hybrid firefly algorithm; K-Means clustering; local search operator; crossover and mutation operators

0 引言

有容量约束车辆路径问题(CVRP)是NP-hard问题,其目标函数为车辆服务完全部客户后,所有车辆所走路程最短或所用费用最少^[1]。目前,该问题求解方法有精确算法、启发式算法和元启发式算法。

因使用精确算法在可接受时间内无法求得大规模CVRP最优解,国内外学者多采用启发式或元启发式算法进行求解^[2]。黄戈文等^[3]利用整数编码和先路由后分组实现解的转换,提出自适应遗传灰狼优化算法(AGGWOA)以研究CVRP。邵可南等^[4]提出模拟

收稿日期:2023-02-25

修回日期:2023-06-27

基金项目:国家自然科学基金项目(61403260);辽宁省自然科学基金项目(2020-MS-233);辽宁省兴辽英才计划项目(XLYC2002017)

作者简介:白雪媛(1999-),女,硕士研究生,研究方向为车辆路径规划、智能优化算法;通信作者:李琳(1978-),女,博士,教授,研究方向为系统建模、智能优化算法。

退火-遗传混合算法 (GA-SA) 解决带硬时间窗和容量约束的单中心车辆路径问题。Francesco 等^[5]用 K-Means 聚类提出 CC-CVRP, 将客户点分为不同集群, 将 CVRP 转化为多个 TSP。

萤火虫算法 (Firefly Algorithm, FA) 是 Yang^[6] 开发的启发式算法。FA 和不同方法结合, 衍生出许多解决 NP-Hard 问题的算法^[7]。标准萤火虫算法因每只萤火虫迭代寻优时都会随机移动, 寻优搜索过程复杂, 算法收敛速度较慢, 易陷入局部最优。为提高算法性能, 许多学者对 FA 做出多方面改进。

Mohammadi 等^[8]用 FA 在考虑订单的交货时间和成本基础上最小化总完成时间。李媛媛等^[9]以 K-Means 聚类中心作为寻优萤火虫, 结合邻域与随机吸引 (N-R 吸引) 提出了 KNRFA 模型。闫军等^[10]将 K-Means 聚类和 Q-learning 自启发式引入蚁群算法, 求解带时间窗同时取送货车辆路径问题。

Li 等^[11]在 FA 中加入自适应对数惯性权重, 用步长减小因子动态调整随机步长, 提出 LWFA。Mohsen 等^[12]为解决 TSP 将 FA 与局部搜索算法 (2-opt 和 3-opt), 遗传算法中的变异交叉因子相结合, 提出离散混合萤火虫算法 (HDFA)。Matthopoulos 等^[13]结合贪婪算法提出混合萤火虫算法, 解决异构固定车队 VRP。Jaradat 等^[14]首先利用 K-Means 聚类将 TSP 分为子问题, 再利用 FA 在每一类中寻找最优路径。

Altabeeb 等^[15-16]将 FA 与局部搜索 (改进的 2-opt 和 2-h-opt)、变异交叉算子结合提出 CVRP-FA, 用田口方法统计确定参数的最佳值。但因 CVRP-FA 在 2-h-opt 处理后只接受有改进方案, 算法易陷入局部最优, 无法产生更好的解, 故将 2-h-opt 作为一种变异, 而非局部搜索, 提出 CVRP-CHFA。该方式使算法可接受 2-h-opt 产生的所有解。该算法还引入合作岛模型概念, 保持解的多样性, 避免算法陷入局部最小值, 达到加快算法收敛速度的目的。

该文提出一新的混合萤火虫算法 (KM-HFA) 求解 CVRP。引入有约束条件的 K-Means 聚类, 以聚类结果为基础构建较好可行解作为萤火虫寻优初始解。结合部分匹配交叉和 Intra-Swap, Inter-Swap 变异算子, 防止算法陷入局部最优。引入 2-opt 和 2H-opt 局部搜索算法提高算法收敛速度, 将算法用于求解标准算例。

对 A, B, P, E 组数据集 72 组算例, 算法 KM-HFA 所得可行解比文献^[5]提出的 CC-CVRP、文献^[13]提出的 HFA 结果好。当客户点逐渐增多时, KM-HFA 的计算稳定性显著优于 HFA, 最优解与经典解的百分比差异要比另两种算法求解的百分比差异小得多, 算例验证了文中算法的有效性。

1 CVRP 描述和数学模型

CVRP 是经典 VRP, 其定义在图 $G(N, E)$ 上, 节点集为 $N = \{0, 1, \dots, n\}$ 。节点 0 为仓库 (配送中心), 节点 $\{1, 2, \dots, n\}$ 表示需服务客户点, $i \in N \setminus \{0\}$ 的需求为 q_i , 从节点 i 到 j 的运输成本为 C_{ij} 。K 为仓库所拥有的车辆集合, Q 为车辆最大载重。

CVRP 数学模型描述如下:

$$\text{Min} \sum_{i \in N} \sum_{j \in N} C_{ij} x_{ij} \quad (1)$$

s. t.

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

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \quad i \in \{1, 2, \dots, n\}, i \neq j \quad (3)$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij}^k q_i \leq Q \quad k \in K \quad (4)$$

$$\sum_{j \in N} x_{ij}^k = \sum_{j \in N} x_{ji}^k \quad i = 0, k \in K \quad (5)$$

$$x_{ij}^k = \begin{cases} 1 & \text{车辆 } k \text{ 从客户 } i \text{ 直接到 } j \\ 0 & \text{其它} \end{cases} \quad (6)$$

目标函数 (1) 为最小化车辆总运输成本, 约束 (2) 和 (3) 保证节点仅被访问一次, 约束 (4) 保证车辆运输过程中载重不超过车辆最大载重, 约束 (5) 保证车辆从仓库出发并最终回到仓库。

2 改进混合萤火虫算法

FA 作为元启发式算法, 随机生成初始解, 但寻优结果在很大程度上依赖初始解, 初始解较好时, 其寻优结果会更好。该文提出的 KM-HFA 先引入带约束条件 K-Means 聚类, 以聚类结果为基础, 构建较好可行解作为萤火虫寻优初始解, 克服寻优结果依赖初始解的问题。为增加萤火虫种群的多样性, 将 GA 的部分匹配交叉和 2H-opt 交换算子与 FA 结合, 进行可行解间操作。最后, 引入局部搜索算法 2-opt 和两种变异算子提高算法收敛速度, 防止其陷入局部最优。KM-HFA 的伪代码如下, 各步骤将在后续部分中具体说明。

KM-HFA 算法

Begin

Input: 目标函数 $f(S)$;

Output: 全局最优解 S^* ;

Initialize: MI, C-R, M-R;

$t = 0$; /* t 代表迭代次数 */

初始种群 $P: S = (S_1, S_2, \dots, S_n)$;

For $i = 1 - P - S$ do

计算萤火虫 S_i 的目标函数 $f(S_i)$;

找出初始种群内最优解 S^* ;

End for

计算汉明距离和亮度

while $t \leq MI$ do

```

for  $i = 1$  to  $P - S$  do
    for  $j = 1$  to  $i$  do
        If  $I_i > I_j$  then
            2H-opt procedure  $S_i$ ;
                If rand <  $C - R$  then
                    PMX procedure  $S_i$ ;
                End if
            2-opt procedure  $S_i$ ;
            mutation procedure  $S_i$ ;
            End if
        End for  $j$ 
    End for  $i$ 
    排序,找出当前最优萤火虫;
     $t = t + 1$ ;
End while
返回全局最优解  $S^*$ ;
End

```

2.1 路径编码转换

在计算过程中,可行解由 K 条 (K 为所用车辆数目) 路径组成,每条路都由 n 个点构成,表示该路径所分配到的客户点个数。该文在计算萤火虫亮度和可行解之间的操作时,需将多路径 CVRP 转换为单路径 TSP,进行路径编码转换(见图 1)。在完成可行解间操作后,编码方式转换回多路径编码,再对各可行解内的路径进行局部搜索和变异操作。

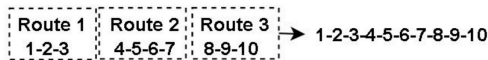


图 1 路径编码转换

2.2 有约束的 K-Means 聚类构建初始解

标准 FA 的寻优随机生成初始解,寻优结果对初始解有较强依赖性,当初始解较好时,最终可找到更好解。该文先引入 K-Means 聚类,聚类后的客户点间距离较接近,再进行路线构建。聚类使距离较接近的客户点归为一类,利用该特性进行路线构建得到的初始解比随机生成的初始解好,将聚类后构建的解为启动萤火虫寻优算法的初始解。

2.2.1 K-Means 聚类

K-Means 聚类采用距离作为对象相似性的评价指标,两个点离得越近就越相似,相似度和距离成反比。两点间距离采用欧氏距离来计算:

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (X_{i,k} - X_{j,k})^2} \quad (7)$$

每类有且只有一个聚类中心,该中心是类内所有数据的平均值,计算点与聚类中心的距离,判断出该点属于哪一类。最终所有数据点被分为独立的,由距离相近的点所组成的 K 类。K-Means 聚类的 K 值需预先设定。为满足车辆载重约束, K 值由客户点总需求

和车辆最大载重计算,结果向上取整,即:

$$K = \lceil \sum_i^N q_i / Q \rceil \quad (8)$$

K-Means 聚类的步骤为:

- (1) 计算两点间欧氏距离;
- (2) 由公式 8 设定聚类数 K ,并随机选取 K 个客户点作为聚类中心 C_i ;
- (3) 将客户点分配给距离其最近的聚类中心,直至所有客户被分为 K 类;
- (4) 更新类的中心点,并重复步骤 3 直至簇不再发生变化。

2.2.2 构建初始种群

初始种群由多个可行解构成,每个可行解都按如下步骤进行构建:

- (1) 派出一辆车从仓库出发随机选取一类 CL_i 进行服务;
- (2) 在该类内随机选择一个客户点并将该客户点从列表删除,以免重复服务;
- (3) 将选中的客户点加入到该路径;
- (4) 判断加入该客户点后的车辆是否超载,若车上的载重未超过最大载货量则重复步骤 2 和步骤 3;否则将车辆返回仓库,并派出下一辆车继续服务;
- (5) 当该类客户都服务完,找到和该类中心最近的下一类中心 C_j ,对 CL_j 类内的客户进行服务;
- (6) 重复操作至数据集内客户点都服务完。

上述步骤每进行一次则得到一个可行解,多次重复后得到初始种群,即后续萤火虫寻优算法的初始解。对每个可行解计算目标函数,即车辆行驶的总距离。初始种群中,萤火虫的亮度和目标函数值成反比,目标函数值越大亮度越小,目标函数值较低的可行解就是更具吸引力的萤火虫。根据亮度对初始种群内的可行解进行排序,目标函数值最小的萤火虫作为当前的最优解 S_{best} 。

2.3 亮度计算

该文使用汉明距离 (Hamming Distance, HD) 计算寻优过程中的萤火虫亮度。HD 为两相等长度字符串对应位元素不同的数量,即比较两种可行解对应位置的客户是否相同,若客户相同则两者间的汉明距离不增加,若不同则汉明距离加 1。

图 2 展示了计算汉明距离的例子, S_i 和 S_{best} 都有 12 个客户点,在客户点中,对应位不同的客户共 7 个,两者之间的汉明距离为 7。

设萤火虫 i 的亮度为 1 到 HD 之间的随机数:

$$I_i = \text{Random}(1, \text{HD}_{i, \text{best}}) \quad (9)$$

比较两只萤火虫的亮度,对亮度大的萤火虫进行后续操作。

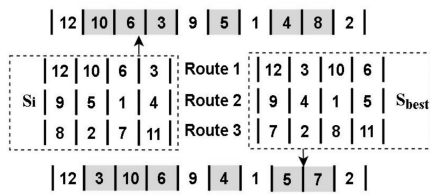


图 2 汉明距离计算

2.4 可行解间的交换

2H-opt 交换和部分匹配交叉都是针对整个可行解进行处理的变换,涉及到路径转换,所以将 2H-opt 和部分匹配交叉放在一起进行操作,可避免多次路径编码的转换增加算法复杂度。在可行解间进行 2H-opt 交换和部分匹配交叉可增加萤火虫种群的多样性,扩大寻优范围,增加找到更优解的可能性。

2.4.1 2H-opt 交换

2H-opt 交换^[17]随机选择可行解 S_i 中两点插入到新方案 S'_i 的头部位置,其余客户点按原路线顺序放入。将 S'_i 和 S_i 的目标函数值进行比较,若新方案 S'_i 的目标函数值小于 S_i ,则更新方案 $S_i = S'_i$ 。

2.4.2 部分匹配交叉

交叉是遗传算法中的操作算子,它可以挖掘种群的多样性,克服启发式算法容易陷入局部解的问题。该文将 S_i 和 S_{best} 作为父代,对两者进行部分匹配交叉 (Partially Matching Crossover, PMX) 操作,具体步骤通过图 3 中的例子展示。

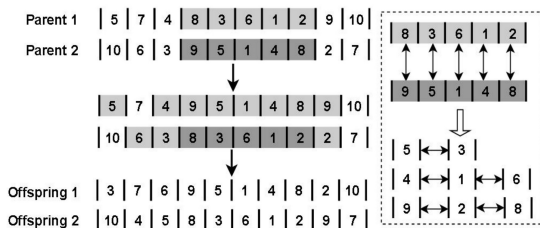


图 3 部分匹配交叉

在两个父代中随机选取两者中相同位置的基因片段进行交换,图中灰色片段被交换;如图 3 所示,由于交换后染色体中可能会出现如基因 5,8,3 等与交换片段所重复的基因,所以需要染色体进行重复检测并进行去重处理;交换的片段建立映射关系,如基因 5 的映射点为基因 3;重复的点将会由其映射点替换,注意此时替换的是未被交换的基因,被交换的基因不做任何处理;将重复的点全部替换完成后,最终形成两个子代。交叉后得到两子代,计算两子代目标函数值,与 S_i 目标值比较,若子代中有更小的解则更新 S_i ,否则不更新。

2.5 可行解内部交换

可行解内变换利用局部搜索和变异算子对可行解内路径进行处理,不仅可以提高算法的收敛速度,维持

萤火虫种群的多样性,从而获得更优可行解,还能避免萤火虫算法寻优过程陷入局部最优。

2.5.1 局部搜索

用局部搜索 2-opt 算子在种群内产生新可行解。在可行解内随机选取某一条路径,在该路径中随机选取两个客户点,将这两点间的子片段进行反转操作后重新插入到该片段中,即可产生新的可行解。

2.5.2 变异算子

变异能维持萤火虫种群的多样性,一定程度克服 FA 易陷入局部最优的缺陷。给定变异率,在区间 (0, 1) 内产生随机数,当随机数小于设定变异率,利用变异算子产生新的可行解。该文采用两种变异算子: Intra-Swap 是在可行解内随机选取一条路径 r_1 , 随机选取该路径的两个点 c_1, c_2 进行交换得到新的可行解; Inter-Swap 是在可行解内选取两条路径 r_1, r_2 , 在 r_1 中随机选择一客户 c_1 , 在 r_2 中随机选择一客户 c_2 , 将 c_1, c_2 进行交换,得到新的可行解。

3 实验及结果分析

文中算法用 Python 在 Intel i5 1.8 GHz 和 4 GB 内存计算机上实现。测试实例包括以下在 <http://vrp.galcos.inf.puc-rio.br> 上公开提供的实例:客户数量在 100 以下的小规模数据集 A, B, P, E, 共 79 组实例;客户数量为 101-1 001 的中规模数据集 X, 选取其中 15 组实例。KM-HFA 对每个实例独立运行 10 次,最大迭代次数为 1 000。该算法参考了 Altabeeb 等^[17]通过田口统计实验所提出的参数值,种群数为 20,变异率为 0.1,交叉率为 0.95。

表 1 列举了文中所用数据的相关信息,其中 Set 为数据集名称,Dimension 为数据维度,Q 表示该数据所用车辆的容量,Instance Number 为从数据集中选取实验实例的个数,Time 为该数据集创建时间。

表 1 数据相关信息

Set	Dimension	Q	Instance Number	Time
Set A	32-80	100	27	1995
Set B	31-78	100	23	1995
Set P	16-101	35-400	23	1995
Set E	13-101	112-6 000	6	1969
Set X	101-1 001	3-1 816	15	2014

文献[5]利用 K-Means 聚类将客户分为不同集群,提出 CC-CVRP 算法,把 CVRP 转为 TSP,对 Set A, X 进行求解;文献[13]结合贪婪算法提出 HFA 对 Set A, B, P, E 进行求解。该文将 KM-HFA 同上述两种算法进行比较,比较各算法得出的最优解所需的

路径距离 (Best) 及与网站公布的已知最优解 (B-K) 之间的百分比差异 (Gap/%)。

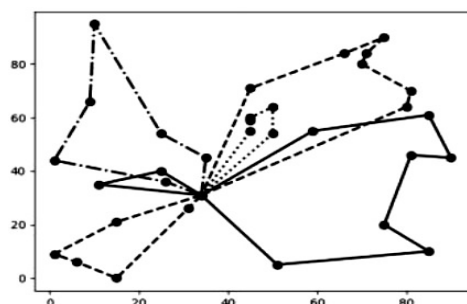
表 2 数据集 A 的 KM-HFA 与其他算法结果比较

Instance	B-K	HFA		CC-CVRP		KM-HFA	
		Best	Gap/%	Best	Gap/%	Best	Gap/%
A-n32-k5	784	831	6.0	832	6.1	810	3.3
A-n33-k5	661	711	7.6	691	4.5	663	0.3
A-n33-k6	742	783	5.5	812	9.4	737 *	-0.7
A-n34-k5	778	827	6.3	796	2.3	793	1.9
A-n36-k5	799	870	8.9	849	6.3	824	3.1
A-n37-k5	669	669	0.0	717	7.2	682	1.9
A-n37-k6	949	1 021	7.6	1 061	11.8	946 **	-0.3
A-n38-k5	730	787	7.8	827	13.3	740	1.4
A-n39-k5	822	898	9.2	895	8.9	860	4.6
A-n39-k6	831	868	4.5	907	9.1	837	0.7
A-n44-k6	937	1 051	12.2	1 034	10.4	966	3.1
A-n45-k6	944	1 108	17.4	1 208	28.0	999	5.8
A-n45-k7	1 146	1 281	11.8	—	—	1 184	3.3
A-n46-k7	914	1 049	14.8	1 029	12.6	945	3.4
A-n48-k7	1 073	1 218	13.5	1 158	7.9	1 130	5.3
A-n53-k7	1 010	1 200	18.8	1 204	19.2	1 098	8.7
A-n54-k7	1 167	1 374	17.7	1 254	7.5	1 226	5.1
A-n55-k9	1 073	1 324	23.4	1 205	12.3	1 123	4.7
A-n60-k9	1 354	1 650	21.9	1 504	11.1	1 411	4.2
A-n61-k9	1 034	1 362	31.7	1 301	25.8	1 117	8.0
A-n62-k8	1 288	1 566	21.6	1 402	8.9	1 356	5.3
A-n63-k10	1 314	1 606	22.2	1 420	8.1	1 373	4.5
A-n63-k9	1 616	1 976	22.3	1 827	13.1	1 670	3.3
A-n64-k9	1 401	1 684	20.2	1 545	10.3	1 484	5.9
A-n65-k9	1 174	1 522	29.6	1 358	15.7	1 191	1.4
A-n69-k9	1 159	1 534	32.4	1 241	7.1	1 228	6.0
A-n80-k10	1 763	2 235	26.8	1 910	8.3	1 969	11.7

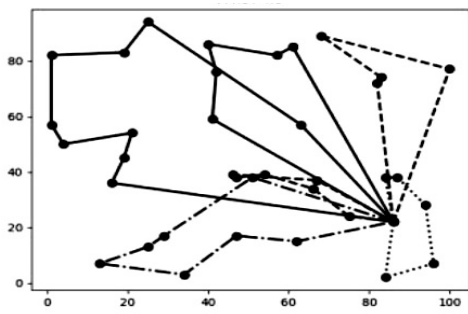
表 2 ~ 表 5 为 KM-HFA 同 CC-CVRP 和 HFA 计算 79 组小规模实例的结果比较,表 6 为 KM-HFA 与 CC-CVRP 计算中规模数据集 X 中 15 组实例结果比较。其中,第一列为 CVRP 实例数据,第二列为公布的已知最优解,第三、四列为 HFA 最佳结果和它与最优解的百分比差异,最后两列为 KM-HFA 求得的最好解和它与最优解的百分比差异。表中加粗部分为与其他算法相比该算法结果更好,“*”代表和经典解有相同路径数的更优新解,“**”代表比经典解增加一条路径,车辆行驶总距离更小的新解。

由表 2 知,对 A 组 27 个实例,KM-HFA 的结果 26 个优于 HFA 和 CC-CVRP 的解,且 A-n33-k5 的解几

乎接近经典解。值得注意的是,对实例 A-n33-k6 和 A-n37-k6,在同路径数条件下,KM-HFA 找到了比经典解更好的新可行解。路径展示如图 4 所示。



(a) A-n33-k6



(b) A-n37-k6

图 4 KM-HFA 得到的 A 组新可行解

表 3 数据集 B 的 KM-HFA 与混合萤火虫结果比较

Instance	B-K	HFA		KM-HFA	
		Best	Gap/%	Best	Gap/%
B-n31-k5	672	672	0	684	1.8
B-n34-k5	788	788	0	810	2.8
B-n35-k5	955	986	3.2	1 008	5.5
B-n38-k6	805	834	3.6	818	1.6
B-n39-k5	549	605	10.2	576	4.9
B-n41-k6	829	919	10.9	866	4.5
B-n43-k6	742	793	6.9	748	0.8
B-n44-k7	909	1 003	10.3	932	2.5
B-n45-k5	751	843	12.3	760	1.2
B-n45-k6	678	751	10.8	730	7.7
B-n50-k7	741	838	13.1	779	5.1
B-n50-k8	1 312	1 414	7.8	1 313	0.1
B-n51-k7	1 032	1 230	19.2	1 080	4.7
B-n52-k7	747	889	19.0	762	2.0
B-n56-k7	707	825	16.7	748	5.8
B-n57-k7	1 153	1 469	27.4	1 167	1.2
B-n57-k9	1 598	1 802	12.8	1 657	3.7
B-n63-k10	1 496	1 811	21.1	1 598	6.8
B-n64-k9	861	1 162	35.0	985	14.4
B-n66-k9	1 316	1 560	18.5	1 339	1.7
B-n67-k10	1 032	1 318	27.7	1 135	10.0
B-n68-k9	1 272	1 512	18.9	1 328	4.4
B-n78-k10	1 221	1 625	33.1	1 296	6.1

对表 3 展示的 B 组 23 个实例, KM-HFA 可行解有 21 个优于 HFA 得到的可行解, 且计算 B-n50-k8 的解几乎接近经典解。对 B-n31-k5 和 B-n34-k5 两个实例, 文中算法得到了与经典解相同的结果。

KM-HFA 和 HFA 对 P 组 23 个实例计算结果如表 4 所示, KM-HFA 可行解有 21 个显著优于 HFA 的可行解。对实例 P-n21-k2 和 P-n22-k2, 两种算法都搜寻到经典解。KM-HFA 算法在计算 P-n16-k8, P-n19-k2, P-n20-k2 时, 在不增加路径数的同时得到比

经典解更好的方案; 在计算 P-n22-k8, P-n23-k8 两组实例时, KM-HFA 算法在比经典解增加一条路径的前提下, 找到了车辆总行驶距离更小的解。

表 4 数据集 P 的 KM-HFA 与混合萤火虫结果比较

Instance	B-K	HFA		KM-HFA	
		Best	Gap/%	Best	Gap/%
P-n16-k8	450	450	0	421 *	-6.4
P-n19-k2	212	212	0	209 *	-1.4
P-n20-k2	216	216	0	209 *	-3.2
P-n21-k2	211	211	0	211	0
P-n22-k2	216	216	0	216	0
P-n22-k8	603	603	0	570 **	-5.5
P-n23-k8	529	529	0	502 **	-5.1
P-n40-k5	458	508	10.9	483	5.5
P-n45-k5	510	595	16.7	516	1.2
P-n50-k10	696	843	21.1	707	1.6
P-n50-k7	554	689	24.4	581	4.9
P-n50-k8	631	828	31.2	677	7.3
P-n51-k10	741	895	20.8	771	4.0
P-n55-k10	694	853	22.9	749	7.9
P-n55-k7	568	670	18.0	604	6.3
P-n55-k8	588	741	26.0	593	0.9
P-n60-k10	744	940	26.3	757	1.7
P-n60-k15	968	1 195	23.5	998	3.1
P-n65-k10	792	1 020	28.8	898	13.4
P-n70-k10	827	1 119	35.3	864	4.5
P-n76-k4	593	733	23.6	637	7.4
P-n76-k5	627	832	32.7	657	4.8
P-n101-4	681	874	28.3	791	16.2

各实例对应解为: P-n16-k8 共 8 条路线: 0-5-14-0; 0-1-0; 0-4-11-0; 0-3-8-0; 0-10-12-15-0; 0-7-9-13-0; 0-2-0; 0-6-0; P-n19-k2 共 2 条路线: 0-18-5-13-15-9-7-2-6-1-0; 0-4-11-14-12-3-17-16-8-10-0; P-n20-k2 共 2 条路线: 0-19-5-14-16-9-13-2-7-6-0; 0-1-10-8-17-18-3-12-15-11-4-0; P-n22-k8 共 9 条路线 0-9-2-1-10-0; 0-18-17-0; 0-14-0; 0-16-0; 0-5-7-0; 0-19-21-20-0; 0-4-3-6-8-0; 0-11-13-0; 0-12-15-0; P-n23-k8 共 9 条路线: 0-5-14-22-0; 0-6-16-0; 0-7-20-0; 0-8-2-0; 0-4-11-0; 0-21-0; 0-3-19-18-0; 0-17-9-13-0; 0-1-10-12-15-0。

表 5 呈现 E 组 6 个实例结果, 其中 KM-HFA 的可行解有 4 个明显优于 HFA 的解。但在计算 E-n33-k4 实例时, HFA 搜寻到经典解, 但是 KM-HFA 得到的最优可行解也与经典解非常接近。

表 5 数据集 E 的 KM-HFA 与混合萤火虫结果比较

Instance	B-K	HFA		KM-HFA	
		Best	Gap/%	Best	Gap/%
E-n22-k4	375	375	0	375	0
E-n33-k4	835	835	0	840	0.6
E-n51-k5	521	626	20.2	522	0.2
E-n76-k10	830	1 186	42.9	970	16.9
E-n76-k14	1 021	1 406	37.7	1 144	12.0
E-n76-k7	682	922	35.2	758	11.1

表 6 数据集 X 的 KM-HFA 与 CC-CVRP 结果比较

Instance	B-K	CC-CVRP		KM-HFA	
		Best	Gap/%	Best	Gap/%
X-n106-k14	26 362	27 515	4.4	27 343	3.7
X-n110-k13	14 971	16 885	12.8	16 735	11.8
X-n120-k6	13 332	14 279	7.1	15 235	14.3
X-n125-k30	55 539	58 535	5.4	62 528	12.6
X-n129-k18	28 940	32 101	10.9	30 662	6.0
X-n134-k13	10 916	12 319	12.9	12 318	12.8
X-n139-k10	13 590	15 973	17.5	14 562	7.2
X-n157-k13	16 876	17 751	5.2	17 508	3.7
X-n162-k11	14 138	15 354	8.6	15 828	12.0
X-167-k10	20 557	22 891	11.4	24 141	17.4
X-n190-k8	16 980	18 168	7.0	18 939	11.5
X-n204-k19	19 565	22 157	13.2	21 781	11.3
X-n209-k16	30 656	33 128	8.1	34 370	12.1
X-n214-k11	10 856	13 163	21.3	13 339	22.9
X-n219-k73	117 595	118 305	0.6	119 693	1.8

表 6 是 X 中 15 个实例结果,其中 KM-HFA 的可行解有 7 个都优于 CC-CVRP 的解;但 CC-CVRP 有 8 组实例结果更好;表明 KM-HFA 在计算中规模数据实例时,计算质量有所降低。分析数据发现,是因为中规模数据相比于小规模数据,客户数量较多、位置较分散。客户点较分散时,K-Means 聚类不能将客户进行更有效的分类,导致萤火虫算法不能以较好的初始解开始寻优,算法计算性能下降。

4 结束语

该文提出 KM-HFA 求解 CVRP。通过引入 K-Means 聚类算法获得更好初始解,将所得解作为萤火虫寻优初始解。结合遗传算法中的 PMX 交叉在种群内产生新的子代,以及两种变异算子 Intra-Swap、Inter-Swap,防止算法陷入局部最优。为提升萤火虫种群的多样性,增强算法的局部搜索能力,加入 2H-opt 交换算子和 2-opt 局部搜索算子。

采用小规模数据 79 组,中规模数据 15 组,共 94 组数据检验 KM-HFA 的计算效果。实验结果表明:在 72 组小规模实例中,KM-HFA 得到的可行解比文献[5]提出的 CC-CVRP、文献[13]提出的 HFA 得到的结果好。且当客户点增多时,KM-HFA 计算稳定性显著优于 HFA 的,得到的最优解与经典解的百分比差异要比另外两种算法求得的解的百分比差异小得多。在 15 组中规模实例中,由于客户数量较多、位置较分散,K-Means 聚类不能将客户进行更有效的分类,导致萤火虫算法不能以较好的初始解开始寻优,最终使 KM-HFA 的计算质量有所降低。

KM-HFA 在计算 5 组小规模实例 A-n33-k6, A-n37-k6, P-n16-k8, P-n19-k2, P-n20-k2 时,不增加路径数的同时得到比经典解更好的方案。在计算实例 P-n22-k8 和 P-n23-k8 时,比经典解路径数增加了一条的前提下,找到了车辆行驶总距离更小的解。

提出的 KM-HFA 还可应用于解决其他 VRP,例

如 VRPTW、绿色车辆路径问题等。未来将研究如何把该算法与其它精确求解算法相结合,得到运算效果更好的混合算法。

参考文献:

- [1] 黄 遥. 混合蛙跳算法及其在带容量约束的车辆路径问题中的应用研究[D]. 南京:南京信息工程大学,2020.
- [2] 靳康飞,闫 军,梁云涛. 容量约束的车辆路径问题研究现状综述[J]. 甘肃科技纵横,2022,51(10):52-56.
- [3] 黄戈文,蔡延光,戚远航,等. 自适应遗传灰狼优化算法求解带容量约束的车辆路径问题[J]. 电子学报,2019,47(12):2602-2610.
- [4] 邵可南,吕成瑶,张帅帅,等. 一种基于冷链低碳物流路径的混合优化算法[J]. 计算机技术与发展,2021,31(2):27-32.
- [5] ALESIANI F, ERMIS G, GKIOTSALITIS K. Constrained clustering for the capacitated vehicle routing problem[J]. Applied Artificial Intelligence,2022,36:1,1995658.
- [6] YANG X S. Firefly algorithms for multimodal optimization[J]. Mathematics,2009,5792:169-178.
- [7] 宋甫祺. 改进的萤火虫算法及其在车辆路径规划中的应用与研究[D]. 南昌:江西财经大学,2021.
- [8] MUSTAFA M, AHMAD S S, MOHAMADREZA L, et al. Modeling and solving the flow-shop scheduling problem with sequence-dependent setup times by firefly algorithm (case study: automotive industry)[J]. Discrete Dynamics in Nature and Society,2022,2022:8962052.
- [9] 李媛媛,魏 延,张文泷,等. 基于 K-means 的邻域结合随机吸引的萤火虫算法[J]. 重庆师范大学学报:自然科学版,2021,38(6):114-121.
- [10] 闫 军,常 乐,王璐璐,等. 带时间窗的同时取送货车辆路径问题求解算法[J]. 工业工程,2021,24(5):72-76.
- [11] LI Y, ZHAO Y, SHANG Y, et al. An improved firefly algorithm with dynamic selfadaptive adjustment[J]. PLoS ONE, 2021,16(10):e0255951.
- [12] MOHSEN A M, AL-SORORI W. A new hybrid discrete firefly algorithm for solving the traveling salesman problem[J]. Applied Computing and Information Technology,2017,695:169-180.
- [13] MATTHOPOULOS P P, SOFIANOPOULOU S. A firefly algorithm for the heterogeneous fixed fleet vehicle routing problem[J]. Int. J. Industrial and Systems Engineering, 2019,33(2):204-224.
- [14] JARADAT A, MATALKEH B, DIABAT W. Solving traveling salesman problem using firefly algorithm and k-means clustering[C]//IEEE Jordan international joint conference on electrical engineering and information technology. Amman; IEEE,2019:586-589.
- [15] ALTABEEB A M, MOHSEN A M, ABUALIGAH L, et al. An improved hybrid firefly algorithm for capacitated vehicle routing problem[J]. Applied Soft Computing Journal,2019,84:105728.
- [16] ALTABEEB A M, MOHSEN A M, ABUALIGAH L, et al. Solving capacitated vehicle routing problem using cooperative firefly algorithm[J]. Applied Soft Computing Journal, 2021,108:107403.
- [17] BENTLEY J J. Fast algorithms for geometric traveling salesman problems[J]. ORSA Journal on Computing, 1992,4(4):387-411.