

Key words: capacitated arc routing problem (CARP); heterogeneous vehicle; memetic algorithm; exchanging vehicles among same group routes

0 引言

限量弧路由问题 (capacitated arc routing problem, CARP) 是典型的组合优化问题^[1], 应用广泛, 可定义为在某给定图下寻求经过图中某些边或弧并满足约束条件的最小费用消耗路径. CARP 属于 NP-hard 问题, 学者们围绕其求解方法提出相应的算法, 如路径扫描法 Path-Scanning 方法与扩张合并法 Augment-Merge 方法^[2]、Ulusoy-Split 算法^[3]以及 Cycle-Assignment 方法等^[4]. 这些方法对于规模小、结构简单的问题能快速地取得质量较好的解, 但难以应对规模较大、结构较复杂的问题, 于是学者提出了求解较大规模 CARP 的元启发式方法, 如禁忌搜索算法^[5-7]、局部搜索算法^[8-9]、memetic 算法 (memetic algorithm, MA)^[10-15]、演化算法^[16]等. 近年来, 针对 CARP 的特点, 国内外学者在局部搜索上做了大量的工作, 如加强的局域搜索^[17]、改进和条件型局域搜索^[18]、多邻域结构^[19]以及嵌入加强的局部搜索算子的分层遗传算法^[20], 针对 CARP 的求解算法研究已取得了一系列的成果.

实际情况下, 扩展 CARP 模型拓宽其应用领域, 成为当前 CARP 的研究趋势. 现有典型的扩展模型有周期性 CARP、多中心 CARP 以及多车型 CARP 等, 本文研究了多车型 CARP (heterogeneous vehicle CARP, HVCARP) 模型, 旨在服务车辆容量存在差异的前提下, 如何规划路径, 满足基本 CARP 约束条件下, 使得路径费用消耗最小. 该问题的难点在于多样化的车型与服务路径的优化匹配. 目前, 关于 HVCARP 的研究较少, 朱征宇设计了框架结构类似 MA 的简单而运行高效的遗传算法 (high efficient genetic algorithm, HEGA) 对 HVCARP 进行求解, 刘天堂则将路径和车型通过费用计算紧密地结合, 使得多车型有了实际意义^[17]. 文献^[17, 21]中问题求解算法均是基于基本 CARP 对应算法的改进, 势必影响 HVCARP 的求解效果. 充分考虑车型和路径的优化匹配, 降低车辆的费用消耗, 是解决 HVCARP 的关键问题.

本文综合考虑 HVCARP 的多种因素, 包括容量限制、车型及车辆数, 提出了一种同档路径交换车辆的算子, 并将该算子嵌入 MA 中, 对 HVCARP 进

行求解. 为了验证算法的有效性, 本文在 3 组测试集共 81 个测试案例上对算法进行测试, 与其他算法实验结果相比, 本文所提算法明显地提高了最优解的质量, 对平均解质量的提高更加显著.

1 多车型限量弧路由问题

1.1 CARP 问题模型描述

CARP 定义在一个带权重的连通图 $G(V, E)$ 上, 其中 V 代表顶点集, E 代表边集. 有 m 辆容量为 Q 的车辆在停车中心处 (记为 $v_0 \in V$). 图 G 中任意边 $e \in E$ 均有属性: 经过费用 $\text{trav_cost}(e)$, 服务费用 $\text{serve_cost}(e)$ 和需求量 $\text{demand}(e)$. 其中 $\text{serve_cost}(e)$, $\text{demand}(e)$ 为非负值, $\text{trav_cost}(e)$ 为正值. 需求量为正值的边称为“任务”, 由所有任务构成的集合为 T . 同时, 若 $e = (v_i, v_j)$, 则定义 $\text{head}(e) = v_j$, $\text{tail}(e) = v_i$, $\text{inv}(e) = (v_j, v_i)$. 显然, e 和 $\text{inv}(e)$ 具有相同的属性值, 但方向相反, 所以给每个任务分配两个非负整数 ID 号. 基于上面的说明, CARP 的任务是在满足特定的约束条件下, 寻求路径集合, 以使得经过费用和服务费用最少. 约束条件有: ① 每条路径的出发点和终点都为车库 v_0 ; ② 每个任务必须出现在一条路径且仅一条路径中, 即被某条路径服务一次; ③ 每条路径服务的任务总需求量要小于等于容量 Q .

对于 CARP, 车辆 k 对应的路径通常表示为所服务的任务序列, 即

$$R_k = (0, t_{k1}, t_{k2}, \dots, t_{lk}, 0) \quad (1)$$

式中, l_k 为路径长度, 为了确保路径满足约束条件①, 设置虚拟任务 (v_0, v_0) , 并为其分配 ID 为 0. 路径 R_k 的费用消耗为:

$$\text{cost}(R_k) = \sum_{i=1}^{l_k} \text{serve_cost}(t_{ki}) + \sum_{i=0}^{l_k} \delta(\text{tail}(t_{ki}), \text{head}(t_{k(i+1)})) \quad (2)$$

式中, $\delta(v_i, v_j)$ 表示两顶点间最短路径费用消耗, 可由最短路径算法 Dijkstra 算法进行计算. 路径 R_k 的负载量为该路径所服务的所有任务需求量总和, 即

$$\text{load}(R_k) = \sum_{i=1}^{l_k} \text{demand}(t_i) \quad (3)$$

满足负载容量等约束条件的路径集合构成 CARP 的一个可行解 S , 可表示为

$$S = \{R_k \mid k = 1, 2, \dots, m\} \quad (4)$$

式中, m 为服务车辆数, 也就是路径数. 对于解 S , 由式(2)可计算其总费用消耗, 即所有路径的费用之和.

$$t_c(S) = \sum_{k=1}^m \text{cost}(R_k) \quad (5)$$

CARP 的目标是在满足约束条件(①、②、③)的前提下, 所有路径费用消耗 $t_c(S)$ 的最小化.

1.2 HVCARP 问题模型描述

HVCARP 为 CARP 的扩展, 主要区别在于前者有 K 种车辆在停车中心处, 第 k 种车容量为 Q_k , 数量为 m_k , 车辆总数为

$$m = \sum_{k=1}^K m_k \quad (6)$$

每条路径对应一种车型, 所有路径数要小于等于车的总数量 m .

对于 HVCARP, 路径的费用消耗分为两部分: 固定成本和可变成本, 固定成本由所服务的车辆能力决定; 可变成本则是对式(2)计算的路径费用进行一定系数的惩罚所得值. 设车辆 v 为 k 型车, 则其服务的路径 $R_{k,v}$ 费用消耗计算如下^[17]:

$$\text{hv_cost}(R_{k,v}) = \theta_k + \rho_k \text{cost}(R_{k,v}) \quad (7)$$

式中, θ_k 是 k 型车的固定成本, $\text{cost}(R_{k,v})$ 是由式(2)计算的基本路径费用消耗, ρ_k 是 k 型车的惩罚系数. 所以解 S 的总费用消耗为

$$\text{hvtc}(S) = \sum_{k=1}^K \sum_{v=1}^{m_k} \text{hv_cost}(R_{k,v}) \quad (8)$$

此外, 基本 CARP 的所有约束条件在 HVCARP 中均需满足, 其中约束条件①可由式(1)满足, 即车辆的出发点和终点都为中心 v_0 . 为了满足容量约束条件②, 对于任意的任务 $t \in T$, 有如下统计定义:

$$\text{count}(R_{k,v}, t) + \text{count}(R_{k,v}, \text{inv}(t)) \leq 1 \quad (9)$$

式中, $\text{count}(R_{k,v}, t)$ 是用来统计任务 t 被路径 $R_{k,v}$ 的服务次数, 式(9)确保任务 t 或 $\text{inv}(t)$ 至多被路径 $R_{k,v}$ 服务一次. 为使得任务 t 或 $\text{inv}(t)$ 被服务, 则

$$\sum_{k=1}^K \sum_{v=1}^{m_k} (\text{count}(R_{k,v}, t) + \text{count}(R_{k,v}, \text{inv}(t))) = 1 \quad (10)$$

对于路径 $R_{k,v}$, 其负载量 $\text{load}(R_{k,v})$ 可由式(3)进行计算. 为了满足约束③, 即路径任务总需求量要

小于等于容量 Q_k , 所以

$$\text{load}(R_{k,v}) \leq Q_k \quad (11)$$

HVCARP 的目标是在满足约束条件(①、②、③)的前提下, 所有路径费用消耗的最小化, 即

$$\min f(S) = \text{hvtc}(S) \quad (12)$$

图 1 所示路径中的任务数为 5, 其基本费用为 $7+7+0+5+0+18+0+9+0+17+0=63$, 服务车辆的固定费用设定为其容量 6, 惩罚系数为 1.6, 则该路径的实际费用消耗为 $6+1.6 \times 63=106.8$.

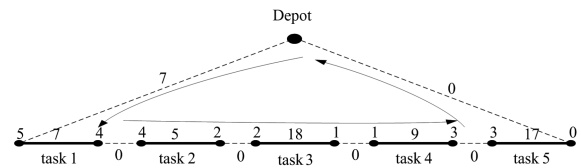


图 1 一条路径费用计算过程

Fig.1 A route cost calculation

式(7)中, 由于不同车型的惩罚系数 ρ_k 存在差异, 所以 $\text{cost}(R_{k,v})$ 一定的情况下, 更改服务车型, 则会导致 $\text{hv_cost}(R_{k,v})$ 的变化. 本文正是基于该原理, 提出了同档路径交换车型的算子, 即在满足一定条件的路径之间, 交换服务车辆, 从而降低路径的费用消耗.

2 基于同档路径交换车辆的 MA

作为相对较新的元启发式搜索方法, MA 是由 Moscato 于 1989 年提出^[22]. MA 可以认为是一种混合的遗传算法 (genetic algorithm, GA), 因为 GA 的变异算子被特定的局部搜索算子所取代, 从而 MA 既继承了 GA 良好的全局搜索能力, 同时弥补了 GA 的局部搜索能力的不足. MA 广泛应用于工程领域的优化问题, 包括 CARP 优化模型的求解^[1,10-15]. 鉴于 HVCARP 的复杂性, 本文设计了一种同档路径交换车辆的局部搜索算子, 并将该算子嵌入 MA 中, 作为 HVCARP 优化模型的求解算法.

2.1 个体编码和适应度函数

CARP 中路径通常表示为任务的有序序列, 如 $(0, 1, 5, 6, 0)$ 表示路径中先后服务了任务 1, 5, 6. 个体则可表示为路径的集合, 如 $(0, 1, 3, 4, 0, 6, 5, 2, 0)$ 表示 6 个任务、两条路径的计划安排. 对于 HVCARP, 由于车型的不一致, 路径中除了任务序列, 还需要包含服务车辆信息, 因此本文提出了一种符合 HVCARP 特性的编码方案, 每条路径都附以车型信息. 图 2 所示为 12 个任务, 3 条路径的个体编码.

0	2	4	7	1	0	vt1
0	3	11	9	8	0	vt2
0	5	12	6	10	0	vt3

图 2 HVCARP 个体编码

Fig.2 Individual coding for HVCARP

HVCARP 是极小化问题, 适应度函数定义为

$$F(S) = M - f(S) \quad (13)$$

式中, $f(S)$ 为目标函数, 由式(12)计算得到, M 是充分大的数, 以确保 $M - f(S)$ 非负.

2.2 同档路径交换车辆算子

HVCARP 较 CARP 具有更大的搜索空间, 两者最根本的区别在于车型约束. 由式(7)可知, 车型对路径费用的计算有很大的影响, 因此在路径不变, 改变其服务车型时, 费用消耗可能会产生变化. 本文对路径进行归类, 若某两条或多条路径的负载量皆不超过服务这些路径最小容量车型, 称这些路径为同档路径 (same group routes, SGR). 对于同档路径, 互换它们之间的服务车型, 仍满足容量约束条件. 从而, 本文提出同档路径交换车辆 (exchanging vehicles among SGR, EVSGR), 其处理对象是路径而非传统局部算子对应的任务. EVSGR 通过车辆交换, 从而实现车型和路径的最优化配对关系, 以降低费用消耗. 容量为 Q_k 车型 k 的同档路径集合构造步骤如下:

- Step 1 置同档路径集合 $SGR = \emptyset$;
- Step 2 置路径序号 $i = 1$;
- Step 3 对于负载量不超过 Q_k 的路径 R_i , 若 SGR 为空集, 则将 R_i 添加到 SGR 集合中, 即 $SGR =$

$SGR \cup \{R_i\}$; 否则, 检查 SGR 的每条路径; 若均与 R_i 属同档路径, 则亦将 R_i 添加到 SGR 中;

- Step 4 $i = i + 1$;
- Step 5 若 i 不超过个体中的路径数, 则转步骤 3) 继续执行, 否则结束同档路径构造.

对于每种车型, 按上述步骤, 构造其同档路径集合. 从而, EVSGR 的大致步骤描述如下:

- Step 1 置车型变量 $k = 1$;
- Step 2 构造车型 k 对应的同档路径集合 SGR_k ;
- Step 3 考虑同档路径集合 SGR_k 中的每对二元组合 (R_1, R_2) :
 - Step 3.1 记 R_1 和 R_2 的服务车辆为 V_1 和 V_2 , 计算两条路径的费用消耗为 $cost$;
 - Step 3.2 计算路径 R_1 被 V_2 服务、 R_2 被 V_1 服务对应的总费用消耗 $cost'$;
 - Step 3.3 若 $cost' < cost$, 则交换 R_1 和 R_2 的服务车辆 V_1, V_2 ;
 - Step 4 $k = k + 1$;
 - Step 5 若 $k \leq K$, 即尚存在未考虑的车型, 则转 Step 2 继续执行; 否则结束.

由于考虑同档路径的所有二元组合, 进行多次同档路径车辆交换, 从而个体邻域有较充分的搜索.

图 3 为两条同档路径及服务车辆信息, 其中 R_1 的费用为 $93 \times 1.4 + 6 = 136.2$, R_2 的费用为 $63 \times 1.2 + 5 = 80.6$, 两条路径的总费用消耗为 216.8, R_1 和 R_2 为同档路径, 互换车型后得到路径 R_3 和 R_4 , 如图 4 所示.

0	t_1	t_2	t_3	t_4	t_5	0	路径容量:5, 可变费用:93, 车型:3, 车容量:6, 惩罚系数:1.4
(a) R_1							
0	t_6	t_7	t_8	t_9	t_{10}	0	路径容量:5, 可变费用:63, 车型:2, 车容量:5, 惩罚系数:1.2
(b) R_2							

图 3 两条同档路径

Fig.3 Two same group routes

0	t_1	t_2	t_3	t_4	t_5	0	路径容量:5, 可变费用:93, 车型:2, 车容量:5, 惩罚系数:1.2
(a) R_3							
0	t_6	t_7	t_8	t_9	t_{10}	0	路径容量:5, 可变费用:63, 车型:3, 车容量:6, 惩罚系数:1.4
(b) R_4							

图 4 两条交换后的同档路径

Fig.4 Two same routes after EVSGR

经过 EVSGR 后, R_3 的费用为 $93 \times 1.2 + 5 = 116.6$, R_4 的费用为 $63 \times 1.4 + 6 = 94.2$, 两条路径总费用消耗为 210.8 相对于交换之前的费用(216.8), 两条路径的费用消耗降低了 6. 该例说明了 EVSGR 的有效性, 本文将其嵌入 MA 中, 作为算子对个体邻域进行局部搜索.

2.3 MA 其他算子

(I) 种群初始化

初始种群的质量对于进化的收敛性有一定的影响, 包含部分质量较好的个体将有助于提高算法的收敛性和最优解的质量. 所以, 本文采用三种生成初始解的策略, 尽可能地提高初始解的质量, 其中两种常用的路径生成策略分别是路径扫描法 Path-Scanning 和最优插入法 (beat insertion heuristic, BIH), 它们均能生成较好的个体. 为了保证群体的多样性, 第三种个体生成策略为随机法, 即个体路径中的任务排列是随机安排的.

(II) 交叉算子

任务的排列次序对 CARP 十分重要, 合理的任务排列次序是个体的优良基因组织, 在进化过程中要尽可能地保留. 本文采用基于路径的交叉策略, 即将两个参与交叉的双亲个体的公共基因分别按对方的顺序进行排列, 从而生成两个子个体. 例如, 双亲为 P_1, P_2 , 在 P_1 中随机地选取一条路径, 记为 R , 根据 R 的长度选取交叉区间为 $[\text{pos}_1, \text{pos}_2]$, R 中该区间的任务序列记为 genetic_pub . 将 P_1 中不属于 genetic_pub 的部分按原来顺序复制到 C_1 中, genetic_pub 中的基因按在 P_2 中的顺序复制到 C_1 中; 交换 P_1 和 P_2 的角色, 同法可生成子个体 C_2 .

(III) 局部搜索算子

本文所设计的 MA 算法中, 除了 EVSGR 局部搜索算子, 亦包含了传统的局部搜索算子, 它们的任务是在邻域中寻找更优个体, 若找到, 则替换个体本身. 具体局部搜索算子如下:

Single insertion(SI): 遍历所有任务, 将当前任务插入到另一个位置; Double insertion(DI): 与 SI 相似, 每次操作的对象是相邻的两个任务; Swap: 遍历所有任务, 交换两个任务; 2-opt: 在一条路径中选择一个区间, 逆序区间内的任务. 对于 SI、DI 以及 Swap 涉及的任务, 两个方向都要考虑.

(IV) 选择算子和其他参数

选择算子决定了参与生成下一代的个体, 本文先通过按照个体适应度函数值降序排序, 然后按顺

序无克隆地选择 popsize 个体进入下一代参与进化过程; 交叉概率采用固定常数形式.

2.4 基于同档路径交换车辆的 MA 步骤描述

根据上述各算子的描述, 本文所提基于同档路径交换车辆 MA(MA with EVSGR, MAEVSGR) 具体描述如下:

Step 1 采用 2.3 节所述三种策略生成初始群体 pop, 规模为 popsize, 计算种群中个体的适应值. 设置最大进化代数 G_{\max} 、交叉概率 P_c 、局部搜索概率 P_s ; 令代数 $\text{ngen}=1$;

Step 2 令个体生成次数 $\text{IT}=1$;

Step 3 从 pop 选取个体 P_1, P_2 , 按照概率 P_c 进行交叉操作, 生成子个体 C_1 和 C_2 ;

Step 4 对 C_1 和 C_2 分别施以算子 EVSGR 操作, 令得到的个体分别为 C_1' 和 C_2' ;

Step 5 对 C_1', C_2' 分别进行局部搜索操作, 包括 SI、DI、Swap 以及 2-opt, 从而得到邻域个体 C_1'' 和 C_2'' ;

Step 6 若 C_1'' 相对于种群 pop 个体非克隆, 则将其插入到种群中, 否则若 C_1' 相对于种群 pop 个体非克隆, 则将其插入到种群中; 同法对 C_2'' 和 C_2' 进行判断和处理;

Step 7 $\text{IT}=\text{IT}+1$;

Step 8 若 $\text{IT} \leq \text{popsize}/2$, 转 Step 3 继续生成新个体;

Step 9 对种群 pop 中个体按 2.3 所述选择策略进行个体选择, 并更新种群 pop;

Step 10 $\text{ngen}=\text{ngen}+1$;

Step 11 若 $\text{ngen} \leq G_{\max}$, 则转 Step 2 继续执行; 否则结束算法.

3 实验结果及分析

因为 HVCARP 是基本 CARP 的新扩展, 相关文献较少, 本文根据文献[17]中算例的生成策略对基本测试数据进行修改, 得到新的测试集. 为了验证本文算法的有效性, 以最好性能以及平均性能评价标准, 对算法 MAEVSGR 与文献[21]提出的高效遗传算法 HEGA 进行比较. 此外, 为了验证算子 EVSGR 的有效性, 对 MAEVSGR 以及移去该算子对应的算法 MA 进行了比较. 算法在多个标准数据测试集上运行、测试, 并对实验结果的最好解和平均值进行比较. 同时, 由于所涉及的算法均是基于概率的, 为了进一步说明算法的有效性和稳定性, 本文采用秩和检验法对实验结果进行了检验.

3.1 测试数据及算法参数设置

本文从 CARP 的 Benchmark 测试数据库中选取三种规模的测试集进行实验测试,分别为小规模测试集 gdb、中等规模测试集 val 和大规模测试集 egl,共 81 个测试算例.由于 HVCARP 异于 CARP,故对上述三个测试集中的算例进行修改,以适应 HVCARP,分别记为 hv gdb、hv val 和 hv egl.本文将服务车辆分为三种类型,并确保修改后的车辆总容量保持不变,车型的容量是对原车容量增加绝对值相同的量.每种车型的车型数可通过计算得到,满足总容量不变,固定费用为车容量.车辆惩罚系数 ρ 设置为 1.0,1.2,1.4.表 1 给出原测试案例 gdb2 改变的 HVCARP 算例,表中将车容量为 5 的一种车型改为车容量分别为 4,5,6 的三种车型,车辆数分别是 2,2,2.本文的其他测试案例亦按此规则进行修改.

表 1 一个 HVCARP 案例

Tab.1 An example of HVCARP instances

名称	顶点数	边数	总容量	容量 Q_k	车辆数 m_k	惩罚系数 ρ_k
hv gdb2	12	26	6 * 5	4	2	1.0
				5	2	1.2
				6	2	1.4

对于 MAEVSGR 和 MA,种群规模 popsize 设置为 30,最大迭代次数 G_{max} 为 5,交叉概率 $P_c = 0.8$,式(13)中充分大数 $M = 30000$.对于 HEGA,本文按照文献[21]的算法设计思想进行了代码实现,最大迭代次数 G_{max} 为 30 000,最大算法重启次数为

15,其余参数设置详见文献[21].本文所涉及算法均采用 C 语言实现,运行环境为 VC 6.0,硬件配置为 Intel(R) Core(TM) i3-41603.60 GHz,每个算例独立运行 30 次.

3.2 结果与分析

实验结果如表 2~4 所示,其中 $|V|$ 为顶点数, $|E|$ 为边数,平均值为算例 30 次独立运行结果的均值±标准差,时间为算例运行 30 次的平均时间,单位为秒,最好值为 30 次独立运行的最优解对应的费用,最好解数为测试集中算法得到好于其他算法的最优解的算例数目,最优解以粗体标出.在某算例上算法的 30 次结果通过秩和检验法检验,若明显(95%的概率)好于其他算法,则标以“* ”.

(I) 小规模测试集 hv gdb 实验结果

hv gdb 共有 23 个算例,实验结果如表 2 所示.由表 2 知,相对于 HEGA 和 MA,MAEVSGR 在测试集 hv gdb 上明显显示出较好的求解性能.MAEVSGR 在小规模测试集 hv gdb 所有算例上,其平均值和最优值都好于 HEGA,说明其在该测试集上的寻优能力好于后者.由于结合了算子 EVSGR,MAEVSGR 问题求解的平均性能和最好性能均明显好于 MA,前者在在所有算例上平均值均好于后者,并且在 20 个算例上取得了好于后者的最好值.由秩和检验结果知,MAEVSGR 在多数算例(14 个算例)上同时明显好于其余算法,而 HEGA 和 MA 却无明显好的算例,所以 MAEVSGR 对于小规模 HVCARP 是有效的.

表 2 hv gdb 测试集结果

Tab.2 The results of hv gdb set

算例	$ V $	$ E $	HEGA			MA			MAEVSGR		
			平均值	时间	最好解	平均值	时间	最好解	平均值	时间	最好解
hv gdb1	12	22	434.54±9.46	<1.00	413.80	400.97±3.77	<1.00	392.40	392.89±3.16*	<1.00	386.80
hv gdb2	12	26	475.48±13.73	1.04	446.80	439.58±3.06	<1.00	432.20	436.20±3.38*	<1.00	428.80
hv gdb3	12	22	388.85±10.54	<1.00	366.80	355.65±5.05	<1.00	343.20	348.53±4.02*	<1.00	342.60
hv gdb4	11	19	377.28±12.72	<1.00	353.20	336.87±1.81	<1.00	333.00	336.30±1.75	<1.00	332.60
hv gdb5	13	26	536.28±14.01	1.05	507.00	487.23±4.95	<1.00	478.60	481.24±5.64*	<1.00	470.60
hv gdb6	12	22	429.31±13.90	<1.00	400.00	390.81±4.69	<1.00	378.80	386.21±5.32*	<1.00	376.00
hv gdb7	12	22	443.18±13.57	<1.00	421.20	409.19±3.32	<1.00	402.80	405.39±2.63*	<1.00	400.20
hv gdb8	27	46	774.37±12.86	1.31	740.80	709.65±4.09	<1.00	696.80	699.47±3.89*	<1.00	685.80
hv gdb9	27	51	750.53±10.87	1.36	729.20	651.49±3.31	<1.00	650.20	647.76±1.60*	<1.00	643.40
hv gdb10	12	25	390.27±10.90	1.30	360.60	347.25±5.43	<1.00	338.40	344.66±5.67	<1.00	335.60

续表 2

算例	V	E	HEGA			MA			MAEVSGR		
			平均值	时间	最好解	平均值	时间	最好解	平均值	时间	最好解
hvgdb11	22	45	832.19±16.80	4.16	795.00	729.97±10.07	<1.00	706.60	719.69±5.57*	<1.00	706.20
hvgdb12	13	23	879.57±23.87	<1.00	839.20	794.75±17.51	<1.00	766.60	782.29±12.25*	<1.00	761.40
hvgdb13	10	28	997.57±18.93	<1.00	941.40	910.49±12.52	<1.00	889.20	905.05±13.50	<1.00	877.60
hvgdb14	7	21	231.93±4.32	<1.00	224.40	220.72±0.71	<1.00	219.00	219.17±1.11*	<1.00	215.60
hvgdb15	7	21	214.72±1.46	1.15	211.60	209.77±0.59	<1.00	208.60	209.23±0.34*	<1.00	208.40
hvgdb16	8	28	284.42±3.73	<1.00	275.60	268.08±1.23	<1.00	264.80	267.51±1.12	<1.00	265.40
hvgdb17	8	28	312.98±2.44	1.35	307.80	306.91±0.49	<1.00	305.60	306.54±0.66*	<1.00	304.80
hvgdb18	9	36	393.92±5.47	2.88	383.20	372.31±1.89	<1.00	368.40	371.51±1.36	<1.00	368.60
hvgdb19	8	11	153.56±3.14	<1.00	147.20	146.29±0.47	<1.00	145.80	145.93±0.19*	<1.00	145.80
hvgdb20	11	22	265.15±3.44	<1.00	258.00	246.32±2.18	<1.00	242.20	246.05±2.22	<1.00	241.20
hvgdb21	11	33	368.81±3.58	1.12	363.40	342.9±1.95	<1.00	340.00	341.72±2.25	<1.00	336.80
hvgdb22	11	44	465.97±3.01	1.33	458.60	443.35±0.66	<1.00	441.20	443.16±0.97	<1.00	440.80
hvgdb23	11	55	585.48±3.52	1.21	577.20	547.93±5.53	<1.00	536.60	547.31±4.72	<1.00	537.80
测试集均值			477.67		457.48	437.76		429.61	434.08		426.64
最好解数					0			3			20

(II) 中等规模测试集 hvval

该测试集包含 34 个测试算例, 实验结果如表 3 所示. 由表 3 可以看出, MAEVSGR 在测试集 hvval 上较测试集 hvgdb 取得了更加明显的整体优化效果, 不仅平均值在所有算例上都好于 HEGA 和 MA, 且在 33 个算例上取得了好于其余算法的最好解, MA 仅在一个算例上取得好于其他算法的最好

值, 秩和法检验结果亦好于 hvgdb, MAEVSGR 在 23 个算例上取得了同时明显好于 HEGA 和 MA 的结果, 明显好比例为 23 / 34, 该比例高于在测试集 hvgdb 上明显好的比例为 14 / 23, 显示了 MAEVSGR 在测试集 hvval 的有效性. 所以, MAEVSGR 对于中等规模的 HVCARP 问题较小规模的具有更强的优化效果.

表 3 hvval 测试集结果

Tab.3 The results of hvval set

算例	V	E	HEGA			MA			MAEVSGR		
			平均值	时间	最好解	平均值	时间	最好解	平均值	时间	最好解
hv1a	24	39	676.83±13.50	4.17	647.60	614.62±3.22	<1.00	610.60	610.06±3.68*	<1.00	607.60
hv1b	24	39	734.09±22.22	1.14	698.80	608.28±7.96	<1.00	594.20	608.09±8.15	<1.00	583.40
hv1c	24	39	791.64±18.25	<1.00	756.40	717.43±17.50	<1.00	670.20	714.24±17.35	<1.00	665.60
hv2a	24	34	771.51±25.59	3.49	720.40	681.12±3.59	<1.00	675.80	669.25±0.72*	<1.00	665.40
hv2b	24	34	754.41±20.86	3.59	728.00	680.15±5.39	<1.00	668.80	669.31±0.50*	<1.00	666.60
hv2c	24	34	992.09±21.91	<1.00	950.20	921.39±14.58	<1.00	887.60	905.01±14.23*	<1.00	869.00
hv3a	24	35	295.70±7.41	3.95	279.20	263.05±0.48	<1.00	261.20	262.98±0.83	<1.00	259.40
hv3b	24	35	284.73±8.14	2.88	271.40	259.59±1.96	<1.00	255.40	256.72±0.84*	<1.00	254.20
hv3c	24	35	348.22±7.37	<1.00	335.60	313.25±0.72	<1.00	309.40	310.36±0.22*	<1.00	309.20

续表 3

算例	V	E	HEGA			MA			MAEVSGR		
			平均值	时间	最好解	平均值	时间	最好解	平均值	时间	最好解
hv4a	41	69	1393.44±45.55	14.56	1284.60	1178.99±6.26	<1.00	1162.00	1164.59±4.12*	<1.00	1153.60
hv4b	41	69	1390.04±34.74	7.92	1320.20	1189.21±2.23	<1.00	1188.80	1188.75±0.25	<1.00	1187.40
hv4c	41	69	1440.06±40.30	3.03	1360.60	1212.35±2.84	<1.00	1206.00	1192.69±4.53*	<1.00	1187.00
hv4d	41	69	1537.19±31.29	1.54	1482.00	1367.16±8.95	<1.00	1353.20	1347.09±8.88*	<1.00	1332.20
hv5a	34	65	1396.22±30.20	12.80	1335.00	1191.28±6.27	<1.00	1179.20	1182.34±8.71*	<1.00	1167.60
hv5b	34	65	1404.94±32.98	6.44	1349.20	1195.91±4.26	<1.00	1194.00	1193.2±1.19*	<1.00	1186.80
hv5c	34	65	1433.53±34.71	3.34	1361.60	1247.55±2.01	<1.00	1239.60	1243.88±5.21*	<1.00	1228.60
hv5d	34	65	1562.46±34.29	1.44	1471.80	1441.34±10.79	<1.00	1416.80	1425.43±9.26*	<1.00	1408.00
hv6a	31	50	884.56±21.88	9.06	839.20	785.80±4.10	<1.00	779.40	783.93±5.08	<1.00	773.00
hv6b	31	50	871.73±21.65	2.80	813.20	776.36±6.93	<1.00	755.00	775.37±3.00	<1.00	768.20
hv6c	31	50	969.61±14.34	<1.00	945.40	906.29±6.11	<1.00	896.20	891.49±5.19*	<1.00	880.00
hv7a	40	66	1130.08±29.14	12.39	1079.20	947.89±3.16	<1.00	938.80	947.63±3.43	<1.00	936.40
hv7b	40	66	1124.45±27.12	6.64	1083.00	967.25±6.96	<1.00	954.40	963.23±9.16	<1.00	944.40
hv7c	40	66	1175.37±25.55	1.16	1132.80	1024.79±8.2	<1.00	1008.20	1015.67±7.53*	<1.00	1003.20
hv8a	30	63	1256.53±23.77	9.39	1223.60	1075.68±3.17	<1.00	1064.40	1074.01±5.44	<1.00	1056.40
hv8b	30	63	1263.53±23.40	5.02	1211.40	1098.97±7.99	<1.00	1081.00	1088.84±4.26*	<1.00	1078.40
hv8c	30	63	1435.22±27.84	<1.00	1379.80	1273.13±3.22	<1.00	1270.80	1267.86±0.75*	<1.00	1263.80
hv9a	50	92	1323.59±27.07	34.95	1276.80	1104.64±4.55	<1.00	1094.20	1099.05±5.13*	<1.00	1086.20
hv9b	50	92	1314.89±25.72	16.33	1275.80	1109.93±6.17	<1.00	1096.80	1104.19±6.56*	<1.00	1092.40
hv9c	50	92	1324.14±26.48	9.53	1278.40	1119.57±4.64	<1.00	1105.20	1112.97±4.70*	<1.00	1102.40
hv9d	50	92	1377.31±20.54	2.26	1313.00	1211±7.44	<1.00	1196.40	1203.03±6.85*	<1.00	1191.20
hv10a	50	97	1560.36±31.31	33.66	1503.60	1270.57±0.98	<1.00	1270.20	1270.19±1.51	<1.00	1266.20
hv10b	50	97	1535.41±33.03	20.79	1461.80	1300.54±1.82	<1.00	1293.40	1298.60±7.24	<1.00	1275.00
hv10c	50	97	1544.15±37.42	10.12	1466.00	1309.06±3.81	<1.00	1303.20	1304.50±7.56*	<1.00	1293.20
hv10d	50	97	1627.84±33.63	2.49	1558.60	1421.64±4.49	<1.00	1415.80	1394.90±8.87*	<1.00	1390.60
测试集均值			1144.88		1093.95	993.70		982.24	986.45		974.49
最好解数					0			1			33

(III) 大规模测试集 hvegl

该测试集中的算例数为 24, 算法实验结果如表 4 所示. 由表 4 所示可知, MAEVSGR 在大规模测试集 hvegl 上的优化效果同样明显, 其在算例上的平均值和最好值均有良好的整体优化效果. 30 次独立运行结果对应的秩和检验结果, MAEVSGR 明显好于 HEGA 和 MA 的算例数为 19, 其比例 (19 / 24)

远高于其对应的小规模测试集 hv gdb (14 / 23) 和中等规模测试集 hv val (23 / 34). 这说明面对复杂的大规模 HVCARP, MAEVSGR 显示了很强的问题求解性能, 主要是因为所提算子 EVSGR 的牵引, 算法 MAEVSGR 能够在解空间里朝最优解的方向进行搜索, 从而提高了最好解的质量以及平均优化效果.

表 4 hvegl 测试集结果
Tab.4 The results of hvegl set

算例	V	E	HEGA			MA			MAEVSGR		
			平均值	时间	最好解	平均值	时间	最好解	平均值	时间	最好解
hvegl-e1-A	77	49	7208.82±186.47	1.66	6819.60	6212.69±83.97	<1.00	5981.60	6079.06±76.78*	<1.00	5922.60
hvegl-e1-B	77	49	7982.42±206.94	1.18	7392.40	7307.23±113.21	<1.00	6977.40	7094.72±80.21*	<1.00	6913.20
hvegl-e1-C	77	49	9353.20±202.84	<1.00	8983.00	8796.07±112.05	<1.00	8577.40	8568.65±76.83*	<1.00	8436.80
hvegl-e2-A	77	49	9948.25±311.63	2.19	9296.00	8547.52±78.28	<1.00	8368.60	8280.48±81.62*	<1.00	8158.40
hvegl-e2-B	77	49	11119.10±228.62	1.42	10693.60	10198.80±112.39	<1.00	9978.20	9869.94±123.30*	<1.00	9682.40
hvegl-e2-C	77	49	13507.50±215.12	<1.00	12890.20	12727.30±170.78	<1.00	12310.40	12385.70±87.74*	<1.00	12166.60
hvegl-e3-A	77	49	12312.80±385.41	1.94	11548.60	10054.6±170.08	<1.00	9643.60	9877.71±114.34*	<1.00	9652.20
hvegl-e3-B	77	49	13888.90±364.57	1.44	13191.60	12455.00±147.85	<1.00	12073.00	12205.70±118.78*	<1.00	11848.80
hvegl-e3-C	77	49	14453.50±434.05	1.20	13507.60	12607.10±114.24	<1.00	12360.20	12564.80±179.61	<1.00	12296.20
hvegl-e4-A	77	49	13522.90±373.39	2.32	12722.00	11080.70±111.35	<1.00	10884.60	10882.80±105.86*	<1.00	10657.40
hvegl-e4-B	77	49	16105.80±393.61	1.32	15175.60	14198.50±162.39	<1.00	13779.60	13978.80±170.10*	<1.00	13646.60
hvegl-e4-C	77	49	20821.70±563.12	<1.00	19783.40	19780.40±384.26	<1.00	18826.80	19678.00±460.13	<1.00	18458.80
hvegl-s1-A	140	95	9559.39±248.73	2.58	8938.60	8287.26±116.99	<1.00	8036.60	8106.29±119.85*	<1.00	7870.80
hvegl-s1-B	140	95	11044.40±218.57	1.69	10642.20	9985.10±137.29	<1.00	9729.80	9634.96±90.19*	<1.00	9488.60
hvegl-s1-C	140	95	13533.90±248.03	<1.00	13146.60	12721.20±145.65	<1.00	12427.60	12066.50±131.87*	<1.00	11827.20
hvegl-s2-A	140	95	20567.00±482.48	2.88	19514.40	16608.90±163.01	<1.00	16196.60	16046.10±176.90*	<1.00	15691.20
hvegl-s2-B	140	95	27794.80±955.14	1.42	26650.80	24369.70±664.69	<1.00	22010.40	24338.80±722.18	<1.00	22140.00
hvegl-s2-C	140	95	28879.40±635.91	1.36	27508.80	26069.20±527.27	<1.00	25298.60	25919.50±694.95	<1.00	24950.40
hvegl-s3-A	140	95	21012.60±488.73	3.86	19950.20	17323.00±128.04	<1.00	17202.80	16824.80±105.24*	<1.00	16552.00
hvegl-s3-B	140	95	25243.60±467.35	1.94	24450.60	21937.80±156.96	<1.00	21676.60	21225.10±207.16*	<1.00	20800.80
hvegl-s3-C	140	95	29907.50±576.97	1.45	28999.80	26537.80±306.18	<1.00	26105.00	26009.10±186.44*	<1.00	25585.40
hvegl-s4-A	140	95	26786.20±619.77	3.17	25755.40	21182.00±218.71	<1.00	20798.00	20666.00±248.31*	<1.00	20230.20
hvegl-s4-B	140	95	31348.10±614.95	1.99	30275.80	26322.40±176.10	<1.00	25889.80	25626.60±202.60*	<1.00	25115.80
hvegl-s4-C	140	95	35845.80±1378.78	2.22	33307.00	24922.50±2587.07	<1.00	20762.20	24916.20±2752.24	<1.00	20862.40
测试集均值			17989.48		17130.99	15426.37		14828.98	15118.60		14539.78
最好解数					0			3			21

(IV) 运行时间

如表 2~4 所示, HEGA 仅在少数算例上的平均运行时间小于 1 秒, 其余算例上耗费了较长的时间, 而 MAEVSGR 在所有算例上的平均时间消耗均小于 1 秒. 算法耗时说明 MAEVSGR 能够在较短时间内取得好于 HEGA 的最好值和平均值. 由于在规定代数下, 解的质量若无提高, 则 HEGA 的内嵌进化部分提前结束, 所以出现大规模算例运行时间

少于小规模算例的现象, 如算例 hvegl-s3-C (1.42 秒) 和 hvqdb11 (4.16 秒). 由于 MA 相对于 MAEVSGR, 没有嵌入算子 EVSGR, 所以复杂性低于后者, 其在算例上的时间消耗亦小于 1 秒, 但是 MA 的求解能力明显弱于 MAEVSGR.

(V) 秩和检验统计

为了进一步验证算法的有效性, 本文对 MAEVSGR、HEGA 以及 MAEVSGR 和 MA 在各

测试集上的算例 30 次独立运行结果进行了秩和检验法检验(95%的概率显著性),统计结果如表 5 所示,其中 W 表示 95%的概率明显好, D 表示无明显区分, L 代表明显差。

由表 5 可知,在 3 个测试集的共 81 个算例上,MAEVSGR 在全部算例上的结果明显好于 HEGA。该统计结果进一步说明了 MAEVSGR 较 HEGA 具有很强的优化性能。对于 MAEVSGR 与 MA,MAEVSGR 明显好于 MA 的结果对应算例数为 57,即在大部分算例上,MAEVSGR 的结果都是明显好的。在 24 个算例上,MAEVSGR 与 MA 无明显区别,且 MAEVSGR 无明显差于 MA 的结果。该统计结果验证了算子 EVSGR 对算法 MAEVSGR 是非常有效的。

表 5 秩和检验
Tab.5 Rank sum test

测试集	MAEVSGR 与 HEGA			MAEVSGR 与 MA		
	W	D	L	W	D	L
hvgdb	23	0	1	15	8	0
hvval	34	0	0	23	11	0
hvegl	24	0	0	19	5	0
算例统计	81	0	0	57	24	0

由上述分析知,MAEVSGR 在 3 个标准测试集所有算例上,在极短的时间内取得了好于 HEGA 的优化效果,并且秩和检验法的检验结果表明了 MAEVSGR 在所有算例上的结果明显好于 HEGA。另外,MAEVSGR 在所有算例上,取得了好于 MA 的平均值,并且在多数算例上取得了好于后者的最优解,说明了算子 EVSGR 能够很好地处理多车型的车辆分配问题,因为其对于路径的合适车型进行了搜索,从而降低了车辆的费用消耗。综上所述,本文所提算法 MAEVSGR 对于 HVCARP 是非常有效的。

4 结论

CARP 是众多问题的原型,本文研究了其更实际的扩展问题模型 HVCARP。根据 HVCARP 的特点,提出了同档路径互换车辆的局部搜索算子 EVSGR,并将其嵌入到 MA 中,形成了 HVCARP 的求解算法 MAEVSGR。为了验证所提算子和算法的有效性,本文将三种规模的标准测试集 gdb、val 和 egl 进行了更改,从而得到 HVCARP 的测试集

hvgdb、hvval 和 hvegl,共包含 81 个算例,并以它们为对象进行实验。实验结果表明,本文所提算法 MAEVSGR 对 HVCARP 问题的最优解和平均解的提升有显著的效果,从而说明了其是有效的。

参考文献(References)

- [1] 梅一. 基于元启发式方法对限量弧路由问题的求解[D]. 中国科学技术大学, 2010.
- [2] GOLDEN B L, DEARMON J S, BAKER E K. Computational experiments with algorithms for a class of routing problems [J]. *Computers & Operations Research*, 1983, 10(1): 47-59.
- [3] ULUSOY G. The fleet size and mix problem for capacitated arc routing [J]. *European Journal of Operational Research*, 1985, 22(3): 329-337.
- [4] LÓPEZ E B, AUCEJO V C, SALVADOR A C, et al. The capacitated arc routing problem: A heuristic algorithm[J]. 1990, 14(1): 107-122.
- [5] HERTZ A, LAPORTE G, MITTAZ M. A tabu search heuristic for the capacitated arc routing problem [J]. *Operations Research*, 2000, 48(1): 129-135.
- [6] GREISTORFER P. A tabu scatter search metaheuristic for the arc routing problem [J]. *Computers & Industrial Engineering*, 2003, 44(2): 249-266.
- [7] BRANDÃO J, EGGLESE R. A deterministic tabu search algorithm for the capacitated arc routing problem [J]. *Computers & Operations Research*, 2008, 35(4): 1112-1126.
- [8] HERTZ A, MITTAZ M. A variable neighborhood descent algorithm for the undirected capacitated arc routing problem[J]. *Transportation Science*, 2001, 35(4): 425-434.
- [9] BEULLENS P, MUYLDERMANS L, CATTRYSSE D, et al. A guided local search heuristic for the capacitated arc routing problem[J]. *European Journal of Operational Research*, 2003, 147(3): 629-643.
- [10] LACOMME P. Competitive memetic algorithms for arc routing problems [J]. *Annals of Operations Research*, 2004, 131(1): 159-185.
- [11] ZHANG Y Z, MEI Y, TANG K, et al. Memetic algorithm with route decomposing for periodic capacitated arc routing problem [J]. *Applied Soft Computing*, 2017, 52: 1130-1142.
- [12] TANG K, MEI Y, YAO X. Memetic algorithm with extended neighborhood search for capacitated arc routing problems [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 1151-1166.
- [13] MEI Y, TANG K, YAO X. A memetic algorithm for periodic capacitated arc routing problem [J]. *IEEE*

- Transactions on Systems Man & Cybernetics, Part B: Cybernetics, 2011, 41(6): 1654-1667.
- [14] MEI Y, LI X D, YAO X. Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 435-449.
- [15] MEI Y, TANG K, YAO X. A global repair operator for capacitated arc routing problem [J]. IEEE Transactions on Systems Man & Cybernetics, Part B: Cybernetics, 2009, 39(3): 723-734.
- [16] HANDA H, CHAPMAN L, YAO X. Robust route optimization for gritting/salting trucks: A CERCIA experience [J]. IEEE Computational Intelligence Magazine, 2006, 1(1): 6-9.
- [17] 刘天堂, 江志斌, 耿娜, 等. 带有异质固定车队的能力约束弧路径问题[J]. 上海交通大学学报, 2012, 46(11): 1759-1763.
- LIU T T, JIANG Z B, GENG N, et al. The heterogeneous fixed fleet capacitated arc routing problem[J]. Journal of Shanghai Jiaotong University, 2012, 46(11): 1759-1763.
- [18] 胡珊, 林丹. 求解 CARP-RP-ML 问题的改进算法[J]. 计算机工程, 2012, 38(7): 168-170.
- HU Shan, LIN D. Algorithms for solving CARP-RP-ML problem[J]. Computer Engineering, 2012, 38(7): 168-170.
- [19] 金倩倩, 林丹. 求解 UCARPP 问题的变邻域搜索算法[J]. 计算机工程, 2012, 38(21): 290-292.
- JIN Q Q, LIN D. Variable neighborhood search algorithm for solving UCARPP problem[J]. Computer Engineering, 2012, 38(21): 290-292.
- [20] 徐凯, 朱征宇. 改进遗传算法对带服务时间约束的弧路径问题的求解[J]. 微机处理, 2010, 31(5): 58-62.
- [21] 朱征宇, 杨永, 邓欣. 一种求解多车型 CARP 问题的高效进化算法[J]. 计算机工程与应用, 2008, 44(8): 212-216.
- ZHU Z Y, YANG Y, DENG X, et al. High efficient evolutionary computing method for solving multi-vehicle CARP [J]. Computer Engineering & Applications, 2008, 44(8): 212-216.
- [22] MOSCATO P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms[R]. Pasadena, CA, CalTech Concurrent Computation Program, 1989: 158-179.