

文章编号:1006-2467(2013)04-0619-07

## 加强的混合遗传算法求解能力约束弧路径问题

刘天堂, 江志斌, 胡鸿韬, 刘冉

(上海交通大学 机械与动力工程学院, 上海 200240)

**摘要:** 为了在可接受的时间里求解具有 NP-hard 性质的能力约束弧路径问题(CARP),提出了加强的混合遗传算法(EHGA). 该算法是在遗传算法框架里嵌入加强的局域搜索算子来强化搜索,充分发挥了遗传算法的全局搜索能力和加强的局域搜索算子的局域搜索能力. 同时,在进行种群替代时,二元锦标赛替代被提出,并使用了种群管理来保持种群的多样性. 测试了标准 CARP 算例,并给出了算法效果比较. 结果表明,加强的混合遗传算法胜出一般的 Memetic 算法,是有效的求解 CARP 的方法.

**关键词:** 能力约束弧路径问题; 元启发式算法; 混合遗传算法; 二元锦标赛替代

**中图分类号:** TP 301 **文献标志码:** A

## An Enhanced Hybrid Genetic Algorithm for the Capacitated Arc Routing Problem

LIU Tian-tang, JIANG Zhi-bin, HU Hong-tao, LIU Ran

(School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

**Abstract:** In order to solve the NP-hard capacitated arc routing problem (CARP) in acceptable time, an enhance hybrid genetic algorithm (EHGA) was proposed. The enhanced local search (ELS) was used within a genetic algorithm (GA) framework to intensify the search. Binary tournament replacement and population management strategy were proposed when population was updated. The CARP benchmark instances were tested. The results show that EHGA outperforms memetic algorithm (MA), and is effective to solve the CARP.

**Key words:** capacitated arc routing problem (CARP); metaheuristic; hybrid genetic algorithm; binary tournament replacement

关于车辆路径的问题大致可以分为点路径和弧路径. 车辆路径优化问题 (Vehicle Routing Problem, VRP) 是典型的点路径问题, 由 Golden 等<sup>[1]</sup>提出的能力约束弧路径问题 (Capacitated Arc Routing Problem, CARP) 是其对应的弧路径问题. CARP 在现实生活有广泛的应用, 如城市废物收集、冬季道

路撒盐除雪、输电线路检测和道路洒水等.

CARP 描述: 给定图  $G' = (V, E)$ , 其中:  $V$  为  $n$  个点组成的点集合;  $E$  为无向边集合. 每个边有非负的需求、服务成本和空跑成本,  $E_R \subseteq E$  是需求边集合, 也叫任务集合. 一个单一车型组成的固定车队停在一个车场  $D \in V$ , 车辆数是决策变量. CARP

收稿日期: 2011-12-07

基金项目: 国家自然科学基金资助项目 (70872077), 国家自然科学基金国际 (地区) 合作交流项目 (70831160527)

作者简介: 刘天堂 (1981-), 男, 河南省信阳市人, 博士生, 主要研究方向为物流系统优化.

江志斌 (联系人), 男, 教授, 博士生导师, 电话 (Tel.): 021-34206065; E-mail: zbjiang@sjtu.edu.cn.

的总成本包括服务成本和经过成本,其目标是确定总成本最小的优化路线,且满足约束:① 每个车辆从车场出发并回到车场;② 每个需求边或弧(任务)仅被服务 1 次;③ 每个路线的总需求不超过车辆能力  $Q$ .

CARP 被 Golden 等<sup>[1]</sup>证明是 NP-hard 问题,精确方法只限于求解小规模算例. 因此,在 20 世纪 80 年代到 90 年代早期,各种可以求解大规模算例的启发式算法被先后提出. 例如:简单启发式算法有 Augment-Merge<sup>[1]</sup>、Path-Scanning<sup>[2]</sup>、Parallel-Insert<sup>[3]</sup>、Construct-Strike<sup>[4]</sup> 和 Augment-Insert<sup>[5]</sup>; 两阶段启发式算法有 Route-First、Cluster-Second<sup>[6]</sup>、Cluster-First 和 Route-Second<sup>[7]</sup>. 近些年来,许多元启发式算法已经被提出用于求解 CARP,如 Tabu Search<sup>[8-9]</sup>、Variable Neighborhood Descent<sup>[10]</sup>、Guided Local Search<sup>[11]</sup>、Memetic Algorithm(MA)<sup>[12]</sup>和 Ant Colony Optimization<sup>[13]</sup>等.

上述中的 MA 本质上是一种混合遗传算法,即联合使用遗传算法(Genetic Algorithm, GA)和局域搜索(Local Search, LS),这种算法充分发挥了 GA 的全局搜索能力和 LS 的局域搜索能力,具有很好的搜索效果. 本文设计的加强的混合遗传算法(Enhanced Hybrid Genetic Algorithm, EHGA)就是基于此,但与文献[12]中的 MA 相比,主要有 3 点不同:① 使用了种群管理;② 提出了加强的局域搜索(Enhanced Local Search, ELS);③ 提出了二元锦标赛替代.

## 1 算法设计

为了更清晰地描述需求边(任务),本文把图  $G' = (V, E)$  转化为一个有向图  $G = (V, A)$ ,每一个边用 2 个方向相反的弧代替. 对每个需求弧进行编号,不再用点对  $(i, j)$  来表示. 每个弧  $u \in A$  有一个弧尾(起点)  $a(u)$ ,一个弧头(终点)  $b(u)$  和经过成本(车辆空跑成本)  $c_t(u)$ .  $A_R \subseteq A$  是  $G$  中的需求弧集合,其中每个弧  $u \in A_R$  有一个需求  $d(u)$ ,一个服务成本  $c_s(u)$  和一个反向弧标志  $inv(u)$ . 原始图  $G'$  中的一个边任务  $u \in E_R$  能从 2 个方向中的任意一个去服务,即在  $G$  中任务  $u$  和  $inv(u)$  只能一个被服务.  $G'$  中的每个边任务会在  $G$  中成为 2 个相反方向的弧  $u$  和  $v$ ,且  $inv(u) = v$ ,  $inv(v) = u$ ,  $c_t(u) = c_t(v)$ ,  $d(u) = d(v)$ ,  $c_s(u) = c_s(v)$ .

如上所述,本文设计了 EHGA,该算法具有与 GA 相似的组成部分——染色体编码、初始种群、选择、交叉,只不过变异被 ELS 代替.

### 1.1 染色体编码和评估

在传统染色体编码中,一个染色体就是一个解,即用车场作为路线分割界限;而在本文的 EHGA 中,采用了与文献[12]中相同的表示方式,即每个染色体是一个任务序列,在 2 个连续的任务之间隐含着最短路. 这个染色体可以看作是 1 个乡村邮递员(Rural Postman Problem, RPP)或者 1 个能力足够大的车辆走过的大巡回(Tour). 这样的染色体结构需要通过 1 个分割(Partition)程序<sup>[6, 12]</sup>将 1 个大的巡回分割成 1 个 CARP 解,即具体的车辆路线. 此程序相当于染色体解码,并用于评估染色体的好坏,而染色体的适应值就是对应 CARP 解的成本.

给定一个 CARP 问题,有 4 个任务,需求分别是  $c_1(8)$ 、 $c_2(14)$ 、 $c_3(8)$  和  $c_4(9)$ . 车辆能力  $Q=30$ . 图 1(a)所示巡回  $T=(c_1, c_2, c_3, c_4)$ ,任务下面表示的是服务成本,虚线表示任意两点间的最短路. 图 1(b)所示为辅助图  $G_a$ . 从点 0 到点 4 的最短路(粗线)表示的是巡回的最优分割:2 个旅行,使用 2 辆车,总成本为 89. 具体来说,旅行路线 1 是 1 辆车服务任务  $c_1$  和  $c_2$ ,成本为  $15+8+4+8+7=42$ ;旅行路线 2 是 1 辆车服务任务  $c_3$  和  $c_4$ ,成本为  $9+8+6+8+16=47$ . 图 1(c)所示为对应的 CARP 解.

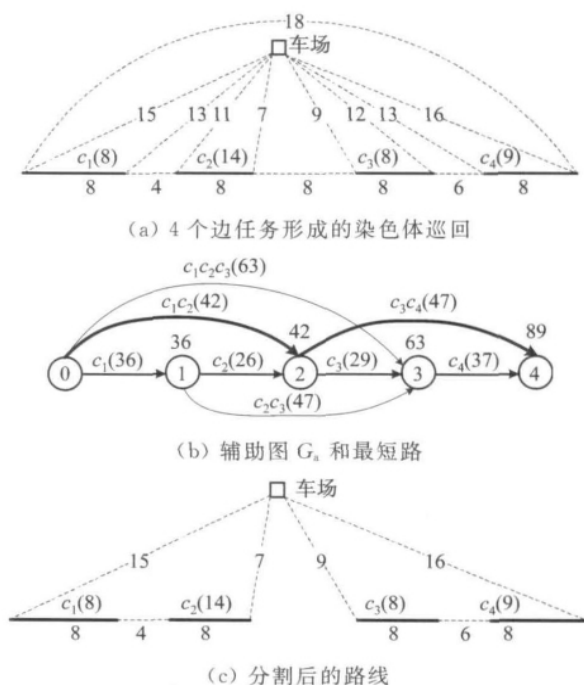


图 1 分割程序示例

Fig. 1 Example of partition

### 1.2 初始种群

种群  $P$  由  $psize$  个染色体组成. 初始种群包括 3 个好(低成本)的染色体以及  $psize-3$  个随机染色

体. 3 个好的染色体分别用 Frederickson<sup>[14]</sup>、路径扫描(Path-Scanning, PS)<sup>[2]</sup>和随机的路径扫描(Random PS, RPS)算法.

RPS 是基于 PS 算法的思想,但是在构建路线时不使用 PS 算法中的 5 个规则,而是从接近目前任务的候选任务中随机选择下一个任务.

PS 和 RPS 算法分别产生一个 CARP 解. 通过将 CARP 解(路线)中的任务从左到右排成一个序列,并排除车场,则这 2 个 CARP 解可以转化为 2 个染色体. Frederickson 算法和随机产生的方法直接产生染色体. 然后,所有 psize 个染色体分别被分割(Partition)程序转化成 CARP 解,同时获得每个染色体对应的解成本. 最后,染色体以成本增序存储.

### 1.3 选择和交叉

本文的 EHGA 采用二元锦标赛(Binary Tournament)选择 2 个父亲  $P_1$  和  $P_2$ , 然后从顺序交叉(Order Crossover, OX)和线性顺序交叉(Linear Order Crossover, LOX)中随机选择一个作为交叉算子,交叉后得到 2 个孩子  $C_1$  和  $C_2$ , 随机选择一个作为孩子  $C$ . 即: OX(LOX)沿着染色体长度随机选择 2 个交叉点  $i$  和  $j$  ( $1 \leq i \leq j \leq t$ ), 子序列  $P_1(i) \sim P_1(j)$  被复制到  $C_1(i) \sim C_1(j)$ , 从  $j+1$  到  $i-1$  循环扫描  $P_2$  (LOX 是从左到右扫描  $P_2$ ), 发现未在  $C_1$  中的任务, 将其填入  $C_1$ . 交换  $P_1$  和  $P_2$  的角色同理可以获得  $C_2$ .

图 2 给出了一个顺序交叉示例. 其中, 2 个交叉点  $i=4, j=7$ . 10 个边任务, 每个边任务  $f$  被转化成 2 个相反方向的弧任务, 且  $\text{inv}(f)=10+f$ , 即:  $\text{inv}(1)=11, \text{inv}(2)=12$  等. 因此, 当用  $P_2(j+1)$  到  $P_2(i-1)$  的任务填入  $C_1$  时,  $P_2$  中的任务 2 应该被排除, 因为它的反向弧任务 12 是同一个任务且已经在  $C_1$  里了.

				$i=4$			$j=7$			
$P_1$ :	1	4	5		7	9	10	12		3 6 8
$P_2$ :	13	2	10		8	6	15	7		9 1 4
$C_1$ :	8	6	15		7	9	10	12		1 4 13
$C_2$ :	9	10	12		8	6	15	7		3 1 4

图 2 顺序交叉示例

Fig. 2 Example of OX crossover

### 1.4 加强的局域搜索

在执行交叉算子后, 得到的孩子  $C$  以概率  $p_m$  执行加强的局域搜索(Enhanced Local Search, ELS).

ELS 扫描每对任务  $(f, g)$ , 执行下面  $M_1 \sim M_6$

个移动算子:

$M_1$ —移动任务  $f$  到任务  $g$  后;

$M_2$ —移动 2 个连续任务  $(f, i)$  到任务  $g$  后;

$M_3$ —交换任务  $f$  和  $g$ ;

$M_4$ —交换任务  $f$  和 2 个连续任务  $(g, j)$ ;

$M_5$ —交换 2 个连续任务  $(f, i)$  和  $(g, j)$ ;

$M_6$ —2-opt.

其中, 任务  $i$  和  $j$  是在各自的路线中在任务  $f$  和  $g$  后被服务.

每个移动算子可能在同一个路线或不同路线中被执行, 在  $M_1 \sim M_5$  中, 假如任务  $f$  被移动到另一个位置, 它可能以  $f$  和  $\text{inv}(f)$  的 2 种方式出现. 在  $M_6$  中, 可能需要将 1 个任务子序列反向.

ELS 和 MA 中的 LS<sup>[12]</sup> 的主要区别在于, LS 执行  $M_1 \sim M_5$  和  $M_6$  的 4 个算子, 而 ELS 执行上述的 6 个算子, 提供了比 MA 的 LS 更有力的搜索能力.

### 1.5 算法框架和种群管理

EHGA 包括 2 个阶段: 主阶段和重启阶段. 主阶段的框架如图 3 所示. 在初始种群构建阶段随机产生染色体时, 为了防止克隆(相同的染色体)出现, 本文试验 try\_max 次来产生随机染色体.

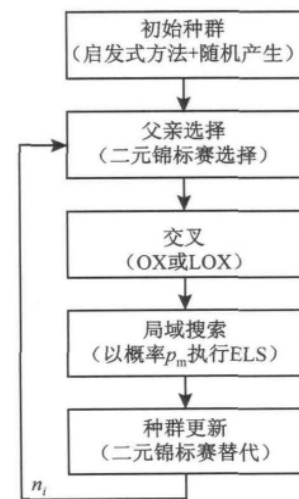


图 3 EHGA 主阶段框架

Fig. 3 Main phase framework of EHGA

在种群更新阶段, 本文提出了新的染色体替代方法(称为二元锦标赛替代). 即: 从父亲种群中随机选择 2 个染色体, 其中成本较大的 1 个染色体作为将被孩子  $C$  替代的染色体  $P_r$ . 种群管理策略可用来动态控制种群的多样性, 避免早熟<sup>[15]</sup>. 若  $C.\text{cost} < P_r.\text{cost}$  或  $\forall P_j \in P \setminus \{P_r\}, \text{Dist}(C, P_j) \geq \epsilon$  成立, 则用  $C$  替代  $P_r$ . 其中:  $\epsilon$  为为了保持种群多样性而设置的临界值;  $\text{Dist}(C, P_j)$  为两个染色体之间的距离, 该

距离以一个染色体的每 2 个相邻任务在另 1 个染色体中是否相邻来确定,若不相邻,则距离 Dist 加 1. 设  $\epsilon_{\min}=1, \epsilon_{\max}=0.25(t-1)$ , 其中  $t$  为任务数. 第 1 次迭代时,  $\epsilon=\epsilon_{\max}=0.25(t-1)$ , 最后一次迭代时,  $\epsilon=\epsilon_{\min}=1$ . 中间每次迭代按照等差递减的方法动态控制  $\epsilon$  的变化.

主阶段执行  $n_i$  次迭代结束;程序重启执行  $n_r$  次,每次保留最好的 3 个染色体,其余的被随机产生的染色体代替,然后重复主阶段过程,只不过局域搜索概率  $p_r$  比主阶段的  $p_m$  大一些.

2 算例试验与分析

2.1 算例与参数设置

计算试验在 3 个 CARP 标杆数据集<sup>[12]</sup>上进

行. 3 个数据集分别为 gdb、val 和 egl, 共有 81 个算例. EHGA 算法使用 C 语言编写, 运行于 Intel Pentium Dual 1.8 GHz CPU, 1 GB 内存 PC.

经过初步试验,主要参数设置如下: 种群尺寸 psize=30, 初始染色体随机产生次数 try\_max=10, 防止克隆的临界值  $\delta=1$ , 主阶段和重启阶段的局域搜索概率  $p_m$  和  $p_r$  分别为 0.1 和 0.2. 停止准则是主阶段的最大迭代次数  $n_i=10\ 000$  和重启最大次数  $n_r=10$ .

2.2 算例结果

算例结果如表 1~3 所示. 表中:第 1~3 列分别表示算例名称、顶点数( $|V|$ )、需求边数( $|E_R|$ );第 4、5 列分别表示 CARP 的下界(Lower Bound, LB)、目前最好解(Best Known Solution, BKS)<sup>[13]</sup>;

表 1 gdb 数据集的结果  
Tab. 1 Results of the gdb set

算例	$ V $	$ E_R $	LB	BKS	MA		EHGA		
					最好解	$t/s$	中值解	最好解	$t/s$
gdb1	12	22	316	316	316	0.00	316	<b>316</b>	0.00
gdb2	12	26	339	339	339	0.24	339	<b>339</b>	0.25
gdb3	12	22	275	275	275	0.03	275	<b>275</b>	0.07
gdb4	11	19	287	287	287	0.00	287	<b>287</b>	0.00
gdb5	13	26	377	377	377	0.06	377	<b>377</b>	0.08
gdb6	12	22	298	298	298	0.94	298	<b>298</b>	1.28
gdb7	12	22	325	325	325	0.03	325	<b>325</b>	0.05
gdb8	27	46	348	348	350	22.12	350	<b>348</b>	27.95
gdb9	27	51	303	303	303	3.94	303	<b>303</b>	4.11
gdb10	12	25	275	275	275	0.03	275	<b>275</b>	0.04
gdb11	22	45	395	395	395	0.70	395	<b>395</b>	0.88
gdb12	13	23	458	458	458	5.43	458	<b>458</b>	7.26
gdb13	10	28	536	536	536	4.12	544	<b>536</b>	6.45
gdb14	7	21	100	100	100	0.03	100	<b>100</b>	0.13
gdb15	7	21	58	58	58	0.00	58	<b>58</b>	0.00
gdb16	8	28	127	127	127	0.03	127	<b>127</b>	0.05
gdb17	8	28	91	91	91	0.03	91	<b>91</b>	0.03
gdb18	9	36	164	164	164	0.06	164	<b>164</b>	0.07
gdb19	8	11	55	55	55	0.00	55	<b>55</b>	0.00
gdb20	11	22	121	121	121	0.18	121	<b>121</b>	0.28
gdb21	11	33	156	156	156	0.09	156	<b>156</b>	0.14
gdb22	11	44	200	200	200	1.86	200	<b>200</b>	2.69
gdb23	11	55	233	233	233	25.60	235	<b>233</b>	34.67
平均值				253.8	253.9	2.8	254.3	253.8	3.8
最优解数量/个				23	22	—	21	23	—
与 BKS 的平均偏差 /%					0.03	—	0.21	0	—

表2 val数据集合的结果  
Tab. 2 Results of the val set

算例	V	E <sub>R</sub>	LB	BKS	MA		EHGA		
					最好解	t/s	中值解	最好解	t/s
val1A	24	39	173	173	173	0.00	173	<b>173</b>	0.09
val1B	24	39	173	173	173	4.56	173	<b>173</b>	4.91
val1C	24	39	245	245	245	15.93	245	<b>245</b>	21.08
val2A	24	34	227	227	227	0.03	227	<b>227</b>	0.06
val2B	24	34	259	259	259	0.12	259	<b>259</b>	0.10
val2C	24	34	457	457	457	12.09	457	<b>457</b>	12.16
val3A	24	35	81	81	81	0.03	81	<b>81</b>	0.03
val3B	24	35	87	87	87	0.00	87	<b>87</b>	0.01
val3C	24	35	138	138	138	18.82	138	<b>138</b>	26.79
val4A	41	69	400	400	400	0.40	400	<b>400</b>	0.44
val4B	41	69	412	412	412	0.67	412	<b>412</b>	0.76
val4C	41	69	428	428	428	10.62	431	<b>428</b>	14.18
val4D	41	69	526	530	541	57.37	533	<b>530</b>	71.19
val5A	34	65	423	423	423	1.03	423	<b>423</b>	1.09
val5B	34	65	446	446	446	0.58	446	<b>446</b>	0.63
val5C	34	65	473	474	474	56.17	474	<b>474</b>	80.12
val5D	34	65	573	575	581	50.41	583	<u>579</u>	99.19
val6A	31	50	223	223	223	0.94	223	<b>223</b>	1.01
val6B	31	50	233	233	233	37.41	233	<b>233</b>	55.23
val6C	31	50	317	317	317	29.02	317	<b>317</b>	42.91
val7A	40	66	279	279	279	1.09	279	<b>279</b>	1.94
val7B	40	66	283	283	283	0.24	283	<b>283</b>	0.33
val7C	40	66	334	334	334	56.21	334	<b>334</b>	75.10
val8A	30	63	386	386	386	0.37	386	<b>386</b>	0.54
val8B	30	63	395	395	395	5.53	395	<b>395</b>	8.75
val8C	30	63	518	521	527	39.70	525	<u>523</u>	70.69
val9A	50	92	323	323	323	10.16	323	<b>323</b>	13.12
val9B	50	92	326	326	326	16.34	326	<b>326</b>	19.45
val9C	50	92	332	332	332	39.55	332	<b>332</b>	58.94
val9D	50	92	385	389	391	117.29	391	391	170.03
val10A	50	97	428	428	428	14.16	428	<b>428</b>	25.39
val10B	50	97	436	436	436	2.59	436	<b>436</b>	3.02
val10C	50	97	446	446	446	9.61	446	<b>446</b>	15.78
val10D	50	97	525	526	530	119.47	533	530	169.00
平均值			343.8	344.3	345.1	21.4	345.1	344.6	31.3
最优解数量/个				28	28	—	28	28	—
最好解数量/个				34	29	—	28	30	—
与BKS的平均偏差/%					0.25	—	0.23	0.10	—

表 3 egl 数据集的结果  
Tab. 3 Results of the egl set

算例	V	E <sub>R</sub>	LB	BKS	MA		EHGA		
					最好解	t/s	中值解	最好解	t/s
egl-e1-A	77	51	3 548	3 548	3 548	41.26	3 548	<b>3 548</b>	52.30
egl-e1-B	77	51	4 498	4 498	4 498	38.60	4 525	<b>4 498</b>	50.55
egl-e1-C	77	51	5 542	5 595	5 595	39.54	5 595	<b>5 595</b>	50.41
egl-e2-A	77	72	5 011	5 018	5 018	84.77	5 018	<b>5 018</b>	93.22
egl-e2-B	77	72	6 280	6 317	6 340	85.23	6 348	6 338	94.78
egl-e2-C	77	72	8 234	8 335	8 415	72.02	8 354	<b>8 335</b>	105.55
egl-e3-A	77	87	5 898	5 898	5 898	134.44	5 902	<b>5 898</b>	160.23
egl-e3-B	77	87	7 697	7 775	7 822	141.86	7 801	7 789	178.42
egl-e3-C	77	87	10 163	10 292	10 433	114.64	10 313	<b>10 292</b>	128.20
egl-e4-A	77	98	6 395	6 444	6 461	162.15	6 475	6 461	177.69
egl-e4-B	77	98	8 884	8 983	9 021	173.81	9 026	9 021	211.53
egl-e4-C	77	98	11 427	11 596	11 779	140.21	11 739	11 668	175.75
egl-s1-A	140	75	5 014	5 018	5 018	115.89	5 019	<b>5 018</b>	126.44
egl-s1-B	140	75	6 379	6 388	6 435	115.98	6 435	<b>6 388</b>	129.06
egl-s1-C	140	75	8 480	8 518	8 518	91.97	8 518	<b>8 518</b>	106.06
egl-s2-A	140	147	9 824	9 884	9 995	485.76	10 002	9 970	548.17
egl-s2-B	140	147	12 968	13 100	13 174	422.50	13 236	13 223	599.91
egl-s2-C	140	147	16 353	16 425	16 795	414.96	16 735	<u>16 711</u>	679.89
egl-s3-A	140	159	10 143	10 220	10 296	594.72	10 333	<u>10 295</u>	748.98
egl-s3-B	140	159	13 616	13 682	14 053	591.12	13 890	<u>13 823</u>	793.40
egl-s3-C	140	159	17 100	17 230	17 297	485.72	17 364	17 297	698.70
egl-s4-A	140	190	12 143	12 268	12 442	854.22	12 490	<u>12 373</u>	1 348.64
egl-s4-B	140	190	16 093	16 321	16 531	794.59	16 523	<u>16 490</u>	1 150.32
egl-s4-C	140	190	20 375	20 517	20 832	830.56	20 824	<u>20 772</u>	1 212.41
平均值			9 669.4	9 744.6	9 842.3	292.8	9 833.9	9 805.8	400.9
最优解数量/个				3	3	—	1	3	3
最好解数量/个					7	—	4	10	7
与 BKS 的平均偏差/%					1.00	—	0.92	0.63	—

第 6、7 列表示文献[12]中的 MA 解和平均计算时间 ( $t$ );由于 MA 是在 1 GHz CPU 上运行,故本文的计算时间被除以 1.8 以便进行近似公平地比较;第 8~10 列表示本文的 EHGA 运行 10 次的中值解、最好解和平均计算时间( $t$ ). 每个表中的最后几行分别表示平均值、最优解数量、最好解数量,以及 MA 与 EHGA 2 个算法所获得的解与目前最好解(BKS)的平均偏差. 表中:黑体表示 EHGA 的最好解达到了目前最好解(BKS);下划线表示 EHGA 的最好解好于 MA 的最好解.

由表 1~3 中可以看出:

(1) EHGA 总体上胜出 MA. 对于小规模算例

gdb 集合, EHGA 比 MA 多获得 1 个更好解(见表 中下划线);对于中等规模算例 val 集合, EHGA 比 MA 多获得 2 个更好解;对于大规模算例 egl 集合, EHGA 比 MA 多获得 24 个算例中的 13 个更好解, 1 个更差解. 对于部分算例, EHGA 的中值解甚至都好于 MA 的最好解,如算例 egl-s4-B 和 egl-s4-C. 此外,对于每个集合, EHGA 的最好解与目前最好解(BKS)的平均偏差均小于 MA 与目前最好解(BKS)的平均偏差,这也反映了 EHGA 比 MA 的效果好.

(2) EHGA 达到了绝大大部分算例的目前最好解(BKS). 对于 gdb 数据集合, EHGA 全部达到

了目前最好解(如表中黑体);对于 val 集合,EHGA 达到了 34 个算例中的 30 个目前最好解;对于 egl 集合,EHGA 达到了 24 个算例中的 10 个目前最好解。

(3) EHGA 的计算时间比 MA 略长。通过 3 个集合的算例测试可以发现,EHGA 比 MA 多获得了更好解是以比 MA 更长的计算时间为代价的。主要原因是种群管理会导致许多次迭代无效;另外,ELS 本身会导致每次迭代的计算时间变长。不过 EHGA 的计算时间仍然是可以接受的。

### 3 结 语

本文针对 CARP 提出了加强的 EHGA,此算法具有如下 3 个特点:种群多样性管理、加强的局域搜索(ELS)和二元锦标赛替代。算例测试表明,EHGA 胜出 MA,是一个有效的求解 CARP 的方法。而且这 3 种策略适用于 GA 求解其他优化问题。同时,从算例结果可以看出,EHGA 未超过目前最好解(BKS)。因此,EHGA 算法还有改进空间,下一步的研究将从新的交叉算子、联合其他局域搜索技术等方面展开。此外,将 EHGA 算法应用于求解其他路径优化问题的研究工作也在进行之中。

#### 参考文献:

- [1] Golden B L, Wong R T. Capacitated arc routing problems[J]. *Networks*, 1981, 11(3): 305-315.
- [2] Golden J, 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] Chapleau L, Ferland J A, Lapalme G, *et al.* A parallel insert method for the capacitated arc routing problem [J]. *Operations Research Letters*, 1984, 3(2): 95-99.
- [4] Pearn W. Approximate solutions for the capacitated arc routing problem [J]. *Computers & Operations Research*, 1989, 16(6): 589-600.
- [5] Pearn W. Augment-insert algorithms for the capacitated arc routing problem [J]. *Computers & Operations Research*, 1991, 18(2): 189-198.
- [6] Ulusoy G. The fleet size and mix problem for capacitated arc routing [J]. *European Journal of Operational Research*, 1985, 22(3): 329-337.
- [7] Benavent E, Campos V, Corberán A, *et al.* The capacitated arc routing problem. A heuristic algorithm [J]. *Qüestió*, 1990, 14(1-3): 107-122.
- [8] Hertz A, Laporte G, Mittaz M. A tabu search heuristic for the capacitated arc routing problem [J]. *Operations Research*, 2000, 48(1): 129-135.
- [9] Brandão J, Eglese R. A deterministic tabu search algorithm for the capacitated arc routing problem [J]. *Computers & Operations Research*, 2008, 35(4): 1112-1126.
- [10] Hertz A, Mittaz M. A variable neighborhood descent algorithm for the undirected capacitated arc routing problem [J]. *Transportation Science*, 2001, 35(4): 425-434.
- [11] 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.
- [12] Lacomme P, Prins C, Ramdane-Cherif W. Competitive memetic algorithms for arc routing problems [J]. *Annals of Operations Research*, 2004, 131(1): 159-185.
- [13] Santos L, Coutinho-Rodrigues J, Current J R. An improved ant colony optimization based algorithm for the capacitated arc routing problem [J]. *Transportation Research Part B: Methodological*, 2010, 44(2): 246-266.
- [14] Frederickson G. Approximation algorithms for some postman problems [J]. *Journal of the ACM (JACM)*, 1979, 26(3): 538-554.
- [15] Sörensen K, Sevaux M. MA | PM: Memetic algorithms with population management [J]. *Computers & Operations Research*, 2006, 33(5): 1214-1225.

Computational experiments with algorithms for a class of routing problems