

一种求解多车型 CARP 问题的高效进化算法

朱征宇, 杨永, 邓欣, 谢志华, 夏梦霜, 李小花

ZHU Zheng-yu, YANG Yong, DENG Xin, XIE Zhi-hua, XIA Meng-shuang, LI Xiao-hua

重庆大学 计算机学院, 重庆 400044

School of Computer Science, Chongqing University, Chongqing 400044, China

E-mail: yanyou_99@yahoo.com.cn

ZHU Zheng-yu, YANG Yong, DENG Xin, et al. High efficient evolutionary computing method for solving multi-vehicle CARP. Computer Engineering and Applications, 2008, 44(8): 212-216.

Abstract: This paper produces a new algorithm called HEGA for solving the CARP (Capacitated Arc Routing Problem). The new algorithm improves the method of coding chromosomes and structure of population in traditional genetic algorithm, and some of its ideas derive from Partheno-Genetic Algorithm (PGA) and memetic algorithm. HEGA can solve the multi-vehicle CARP which is hardly to be solved by existing algorithms, in solving general CARP, its efficiency and accuracy are more excellent as well. Related with a typical case of optimizing the routings of sprinkler cars, the new method is evaluated with a real set of data.

Key words: multi-vehicle; CARP; HEGA; sprinkler car

摘要: 对传统遗传算法的染色体编码机制和种群结构进行了改进, 并借鉴单亲遗传算法和 Memetic Algorithm (MA) 算法的优秀思想, 设计了一种解决 CARP (Capacitated Arc Routing Problem) 问题的高效算法 HEGA。新算法不但有效解决了使用现有算法无力解决的多车型 CARP 问题, 并且应用于一般的单车型 CARP 问题在求解效率和求解精度上也比现有 MA 算法效果更好。结合洒水车路径优化问题, 通过一组真实的数据集合对文中算法在该问题上的求解能力做出评测。

关键词: 多车型; CARP; HEGA; 洒水车路线优化

文章编号: 1002-8331(2008)08-0212-05 **文献标识码:** A **中图分类号:** TP301.6; N945.15

1 引言

车辆路径优化问题的研究大体上可以分为两类: 对点服务的路径优化问题 (VRP) 和对边服务的路径优化问题 (ARP), 而在 ARP 问题中应用最为广泛的要属带容量约束的弧路径优化问题 (CARP), 比如: 道路清洁车路线优化、冬季道路除冰车路线优化、输电线路检测等等^[1], 本文研究所针对的洒水车路线优化问题同样属于 CARP 中的一个典型应用实例。

同众所周知的 VRP 相比, CARP 的研究在相当长的时间内都被研究者们所忽视了, 国内对 CARP 的研究目前仅仅处于起步探索阶段。然而随着 CARP 在日常生活中的应用越来越广泛, 从上世纪 80 年代起此研究领域已经在国际上引起了相当数量的专家学者们的重视, 许多研究者从不同角度对此问题作了大量的分析建模, 并提出了很多优秀的解决方案。CARP 是一个 NP-hard 问题, 使用精确的数学方法求解仅仅局限于在 20 条服务弧以内的数据规模下进行^[2], 因此在解决实际问题中通常是通过构建优质的启发式算法来进行求解, 比如: Hertz^[3] 所提出的基于禁忌搜索的 CARPET 算法和 Beullens^[4] 的带引导的局部搜索算法都是目前解决 CARP 问题的经典算法。

Philippe Lacomme^[5,6]通过对传统遗传算法改进并结合局部搜索 (Local Search) 技术提出了一种 Memetic Algorithm (MA) 算法, 在求解城市垃圾回收车辆路线优化问题的过程中取得了良好的效果。

尽管如此, CARP 目前在国际上的研究还远未成熟, 大多数研究仍然局限于解决基本 CARP 或稍复杂一些的混合 CARP (MCARP)。当问题的数据规模很大 (100 条以上服务弧) 时, 现有算法在求解效率上均不够理想, 主要表现为运行时间长和求解精度下降, 同时现有算法无力解决应用中常见的由于车辆的容量不统一而带来的多车型 CARP 问题。因此本文提出一种新的进化算法 HEGA (High Efficient Genetic Algorithm), 结合洒水车路线优化这样典型的 CARP 应用实例进行研究, 成功解决了以上两个难题, 并在第 7 章通过一组由有关环卫部门提供的真实数据对本文算法的优化能力做出测试。

2 问题描述与数学模型

对于标准 CARP 已经由 Golden 和 Wong^[7] 在 1981 年给出了定义, 然而不同应用场合下的 CARP 会有各自不同的约束条

基金项目: 高等院校博士学科点专项科研基金 (the China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No.20030611016)。

作者简介: 朱征宇 (1959-), 男, 副教授, 硕士生导师, 主要研究方向: 智能优化管理、Web 智能检索、Web 数据挖掘; 杨永 (1982-), 男, 硕士研究生, 主要研究方向: 智能优化管理、电子商务。

收稿日期: 2007-07-05 **修回日期:** 2007-09-17

件。对于多车型洒水车路径优化问题, 有两个方面特殊于标准 CARP: (1) 洒水车行驶和喷洒的道路大多是城市中的主干道, 对一条道路喷洒通常要对道路两边分别进行喷洒, 因此处理问题时应将每条道路转化为方向相反的两条单行道; (2) 执行喷洒作业的洒水车吨位不统一。

洒水车每次从车场出发沿着某条路径对该路径中需要喷洒的道路进行服务并返回车场加水。问题的求解要求获得一个车辆作业路径的集合, 按照这些路径进行作业可以使总行驶花费 $cost$ 最小(即成本最低), 并且满足两个约束条件: (1) 每条路径中总共需要喷洒的水量 $load(T_i)$ 不超过服务该路径的洒水车的容量 Q_i ; (2) 所有需要喷洒的街道都只被一辆洒水车服务一次。即有如下的目标函数和约束条件:

$$\text{目标函数: } \text{Min} \sum_{i=1}^K cost(T_i) \quad (1)$$

约束条件:

$$load(T_i) = \sum_{j=1}^{|T_i|} q(T_{ij}) \leq Q_i, 1 \leq i \leq K \quad (2)$$

$$cost(T_i) = D(\sigma, T_{i1}) + \sum_{j=1}^{|T_i|-1} (w(T_{ij}) + D(T_{ij}, T_{i,j+1})) + w(T_{i,|T_i|}) + D(T_{i,|T_i|}, \sigma), 0 < |T_i| < \varepsilon \quad (3)$$

$$\sum_{i=1}^K |T_i| = \varepsilon \quad (4)$$

$$T_i \cap T_j = \Phi \quad (5)$$

在上述各等式中 T_i 表示第 i 条路径, $|T_i|$ 表示该路径中需要服务的道路数目, K 为解中所含路径数目, 其值取决于所求得具体解, T_{ij} 表示第 i 条路径中第 j 条需要服务的道路, $q(T_{ij})$ 表示 T_{ij} 需要喷洒的水量。式(3)为第 i 条路径的行驶花费计算函数, 其中 $w(T_{ij})$ 表示 T_{ij} 的长度, $D(T_{ij}, T_{i,j+1})$ 表示 T_{ij} 的终点到 $T_{i,j+1}$ 的起点之间的最短距离, ε 表示所有需要服务的道路数目。式(5)表示一条道路只能在一条路径中被服务。

3 进化算法设计

3.1 距离矩阵

算法要求输入的数据有: (1) 城市道路交通图 $G(V, E)$, E 中包括所有需要服务的主干街道(服务边)和不需要服务但车辆可以行驶通过的非主干街道(非服务边), V 为 G 中所有顶点的集合; (2) 各种吨位车辆的数目。在算法初始化时将无向图 $G(V, E)$ 转换成有向图 $G(V, A)$, 将 E 中的每条边表示成方向相反、需求量和长度相等的两条弧, 并对所有服务弧顺序编号。通过 Floyd 算法计算出图 G 中任意两点之间的最短距离矩阵 D 和记录任意两点间最短路径的矩阵 P 。

3.2 种群结构和染色体的编码机制

传统 GA 中, 通常每条染色体都表示一个问题解, 种群中有多条染色体, 在种群初始化时要确定种群中染色体的数目。Lacomme 的 MA 算法中每条新的染色体都必须通过一个时间复杂度为 $O(\varepsilon^2)$ 的 split 过程来计算其适应值并将其转化为一个可行解, 显然当服务弧数目 ε 很大时, 算法中对 split 过程的大量调用使整个算法运行效率急剧下降。

在本文的 HEGA 中, 每条染色体仅代表一条路径, 即问题解的一部分, 而整个种群代表问题解。种群中染色体的数目不固定, 其值取决于当前解中路径的数目, 并在种群中增加一条

空染色体来实现种群中染色体的数目在进化过程中可以随时增加或减少, 即: 若当前问题解中有 k 条路径, 则当前种群中有 $k+1$ 条染色体。

染色体采用整数编码机制, 染色体中的基因位用它所代表的那条路径要服务的所有服务弧的编号来表示。比如: 若当前种群中有一条染色体为 $chrom1: (4, 5, 8)$, 就表示当前问题解中有一条路径依次服务到第 4、5、8 条服务弧。显然, 染色体的长度取决于该染色体对应的路径中服务弧的数目, 并且任意两条染色体之间不会有重复的服务弧出现。每条路径的总行驶花费 $cost(T_i)$ 可以通过式(3)来求得, 算法中采用路径行驶花费的倒数来做为与该路径对应的染色体的适应值, 即: $chrom[i].fitness = 1/cost(T_i)$ 。

根据式(2), 每条路径的总洒水量不得超过服务该路径的车辆的容量, 因此算法中每条染色体用一个固定的 $load$ ($load =$ 某辆车的容量) 属性来判断该染色体是否可行, 即: 若所有服务弧的洒水量之和不超过该染色体的 $load$ 值 ($chrom[i].demand \leq chrom[i].load$), 则该染色体可行。

初始种群采用随机生成的方式。随机生成一条新染色体并将 A 中的服务弧随机地向新染色体中填充, 同时保证染色体可行, 并删除 A 中已经填充过的服务弧, 反复执行直到 A 中所有服务弧被全部删除。

3.3 染色体的变异和局部搜索

在本文的 HEGA 算法中借鉴了 MA 的局部搜索思想, 染色体的变异由一个局部搜索过程所替代, 通过局部搜索来实现在一定空间内为一条染色体找到所有基因位的相对最优排列顺序。在局部搜索中对染色体中的每一对服务弧 (u, v) 依次执行四步操作:

LSN1: 如果 u 和 v 是端点相同但方向相反的两条弧, 则检查交换 u 和 v 的位置后染色体的适应值是否增加, 若增加则交换, 否则退出 LSN1;

LSN2: 检查将 u 移到 v 后面是否会增加染色体的适应值, 若增加则交换, 否则退出 LSN2;

LSN3: 若 u 与 v 不相邻, 则检查把一系列边 $(u \cdots x)$ (x 为 u 和 v 之间的一条弧) 移到 v 的后面是否会增加染色体的适应值, 若增加则交换, 否则退出 LSN3;

LSN4: 若 u 与 v 之间的弧多于 2 条, 则检查将 u 与 v 之间的所有基因位的顺序逆转以后是否能增加染色体适应值, 若增加则逆转, 否则退出 LSN4。

3.4 进化算子

传统遗传算法中, 一般主要是通过一个复杂的交叉算子来产生子代染色体, 然而单一而复杂的交叉限制了算法的灵活性且不利于算法的全局收敛性。在 HEGA 算法中以高效、灵活为目标设计了三个简单的进化算子: 移位、换位、简单交叉, 来代替传统 GA 中的交叉算子, 在进化阶段按相同的概率随机选择其中一个算子对双亲染色体执行进化操作。

(1) 选择: 由于在算法中, 每条染色体都只能代表问题解的一部分, 因此任何两条染色体的适应值之间没有可比性, 所以本算法在选择双亲染色体时采用最简单的方法, 即随机选两条不同的染色体 $P1, P2$ 作为双亲染色体。

(2) 换位: 分别在双亲染色体 $P1, P2$ 中任选连续 n 位做交换, n 不大于双亲染色体中任意一条的基因位数目。如: $P1: (2, 4, 6, 7, 15, 17); P2: (1, 8, 13, 20); C1: (2, 4, 6, 8, 13, 17);$

C2:(1,7,15,20)。

(3)移位:对于双亲染色体 $P1, P2$, 在 $P1$ 中任选连续的 m 位插入 $P2$ 中某一基因位之前。当 $P1, P2$ 中有一条染色体为空染色体时, 从非空染色体中选择连续 m 位放入空染色体中, 此时两条子染色体都为非空。若 m 等于 $P1$ 中基因位的数目时, 此时子染色体中将有一条为空, 如: $P1:(2,4,6,7,15,17); P2:(1,8,13,20); C1:(2,4,6); C2:(7,15,17,1,8,13,20)$ 。

(4)交叉:单点交叉, 即:对于双亲染色体 $P1, P2$, 分别随机选择其中的一位作为交叉点, 然后从各自交叉点开始将 $P1, P2$ 中后半部分对调, 当 $P1, P2$ 中有一条染色体为空时, 此时会产生一条新的非空染色体, 如: $P1:(2,4,6,7,15,17); P2:(1,8,13,20); C1:(2,4,6,13,20); C2:(1,8,7,15,17)$ 。

(5)染色体替换:进化算子执行完毕以后将在种群中对双亲染色体作替换操作, 从而完成种群的一次进化, 染色体替换操作由 4 步完成:

步骤 1 判断两条子染色体是否可行, 若可行则进入下一步, 否则抛弃两条子染色体, 退出染色体替换操作。

步骤 2 对两条子染色体分别调用局部搜索变异操作。

步骤 3 判断两条子染色体是否比双亲染色体更优化, 即若 $C1.fitness+C2.fitness>P1.fitness+P2.fitness$, 则用两条子染色体替换双亲染色体并进入下一步, 否则抛弃两条子染色体, 退出染色体替换操作。

步骤 4 由于移位和交叉算子可能会使种群中空染色体的数目发生改变, 因此当双亲染色体被替换后, 应判断种群空染色体的数目是否发生变化, 并通过删除或增加一条空染色体来保证种群中始终只有一条空染色体。

4 算法主要执行过程

若不考虑第 5 章的优化减速后重优化措施, 则算法的终止条件有两个: (1)进化代数达到所设的最大值 $maxln$; (2)连续无优化进化次数超过所设的最大值 $maxnon$ 。整个算法的执行流程如图 1 所示。

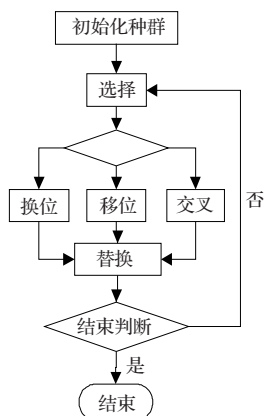


图 1 算法执行流程

5 优化减速后的重优化措施

实验发现, 初始种群进化次数大约超过 9 000 以后, 算法会出现优化减速现象。随着进化次数的增多这种减速现象会更加严重, 而此时收敛的解可能并未达到最优。因此 HEGA 算法采用了一种优化减速后的重优化措施, 即: restart, 实验表明使用 restart 会使算法全局收敛性和解的稳定性更好。具体措施如下:

算法采用两个种群来求解问题: $bestPopu$ 用来保存当前最优解, $computePopu$ 作为进化种群, $computePopu$ 种群第一次初始化后, 将其拷贝入 $bestPopu$ 保存。对 $computePopu$ 执行第 4 章的算法流程, 当结束条件满足时判断 restart 次数是否已达到所设的最大次数 mrn , 若没有达到就执行 restart 操作, 否则整个算法停止并输出 $bestPopu$ 中所存最优解。restart 操作由以下步骤构成:

步骤 1 若 $computePopu.totalCost < bestPopu.totalCost$ ($totalCost$ 为种群中所有染色体的 $cost$ 值之和), 则将 $computePopu$ 拷贝到 $bestPopu$ 中保存, 否则不保存直接进入步骤 2。

步骤 2 根据不同的概率选择三种操作的一种: (a) 将 $bestPopu$ 中所有染色体的服务弧序列连接在一起对 $computePopu$ 重新初始化(概率:0.6); (b) 使用所有服务弧的随机序列来对 $computePopu$ 重新初始化(概率:0.3); (c) 将 $bestPopu$ 拷贝入 $computePopu$ (概率:0.1)。

步骤 3 重新对 $computePopu$ 执行第 4 章的算法过程。

6 实验分析

本文算法实验采用的数据来自重庆市某区的实际主要街道交通图和该区日常洒水降尘作业的一些实际数据, 所有数据由该区环卫部门提供。完整数据规模为 37 个顶点、51 条服务边和 8 条非服务边(拓扑图如图 2 所示), 该单位实际拥有车辆情况为: 5 吨车 2 辆, 8 吨车 3 辆, 10 吨车 3 辆。表 1 中给出了每条街道的实际长度($cost$)和每条街道的单边需水量($demand$), 其中街道长度的单位为百米, 需水量以 1/10 吨为单位。

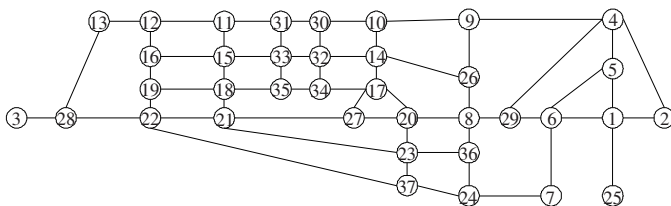


图 2 数据集拓扑图(顶点 1 为加水点)

本文 HEGA 算法的实验除了在图 2 中的完整数据规模下进行以外, 另取图 2 的三个子图形成三个不同规模的数据进行实验(实验数据包含在表 2 中)。实验分两部分进行: (1)求解多车型洒水车路线优化问题的实验; (2)在解决单车型 CARP 时本文算法与 Memetic Algorithm 的对比实验。所有实验对算法中关键的三个参数采用相同的设置: $maxln: 30\ 000, maxnon: 6\ 000, mrn: 15$ 。实验环境: Intel Pentium 1 300 MHz, 256 M 内存, 操作系统: Microsoft Windows XP Professional, 程序开发环境: Delphi 7.0。

6.1 多车型实验

在这四个不同规模的数据下本文算法的实验结果如表 2 所示。

以下数据是算法在数据集 data4 下的一次执行结果(路径中“=”连接的是在路径中需要喷洒的边, “-”连接的是不需要喷洒的边):

第 1 条路径: 1=6=7=6-1, 车辆: 5 吨, 路程: 1.89, 需求: 4.88

第 2 条路径: 1-6-29-8-26=9-10=30-32=33=15=16=12=11=31=30=10=14-26-8-29-6-1, 车辆: 5 吨, 路程: 4.586, 需求: 4.867

第 3 条路径: 1-6=5=1, 车辆: 5 吨, 路程: 1.702, 需求: 4.678

表 1 图 2 中每条街道的长度和需水量

edges	cost	demand	edges	cost	demand	edges	cost	demand	edges	cost	demand
(1,5)	7.57	30.28	(8,20)	6.17	7.16	(13,28)	3.10	13.39	(18,21)	0.71	2.84
(1,6)	3.95	15.80	(8,26)	2.08	8.32	(14,17)	1.10	4.40	(18,35)	1.13	4.50
(1,2)	12.45	75.00	(8,36)	3.86	4.47	(14,26)	4.23	16.92	(19,22)	0.95	3.80
(1,25)	4.73	21.28	(9,10)	2.90	12.53	(14,32)	0.42	1.66	(20,27)	1.88	2.18
(2,4)	16.00	69.12	(9,26)	2.25	9.00	(15,16)	2.03	8.12	(21,22)	1.86	2.16
(4,5)	2.25	9.00	(10,14)	0.86	3.44	(15,18)	1.89	7.56	(21,23)	18.33	73.32
(5,6)	5.50	16.50	(10,30)	0.70	3.00	(15,33)	0.31	1.25	(21,27)	1.88	2.18
(6,7)	5.50	16.50	(11,12)	1.30	5.62	(16,19)	1.66	6.64	(22,28)	1.60	1.86
(6,29)	3.35	3.89	(11,15)	2.48	9.92	(17,20)	3.13	12.52	(22,37)	7.14	28.56
(8,29)	3.36	3.89	(11,31)	0.52	2.26	(17,27)	0.37	1.48	(23,36)	9.16	36.64
(3,28)	1.26	2.46	(12,13)	2.90	12.53	(17,34)	1.50	6.00	(24,37)	2.38	9.52
(4,9)	6.75	29.16	(12,16)	2.37	9.48	(18,19)	1.95	7.80	(24,36)	1.54	1.79
(30,31)	0.52	2.25	(32,33)	0.31	1.25	(34,35)	1.13	4.50	(30,32)	1.38	0.00
(32,34)	1.38	0.00	(31,33)	1.78	0.00	(33,35)	1.48	0.00	(4,29)	11.26	0.00
(7,24)	8.72	0.00	(20,23)	6.20	0.00	(23,37)	4.13	0.00			

表 2 多车型 CARP 实验数据

数据集	数据规模	测试数据 包含的结点(图 2 中)	实验车辆			实验结果/km	运行时间/s
			5吨	8吨	10吨		
data1	9 个顶点 20 条服务弧	1,2,4,5,6,7,8,25,29	1	1	0	20.216	2
data2	16 个顶点 36 条服务弧	1,2,4,5,6,7,8,9,20,23,24,25,26,29,36,37	1	1	1	32.712	8
data3	21 个顶点 54 条服务弧	1,2,4,5,6,7,8,9,10,14,17,27,21,20,23,24,25,26,29,36,37	2	2	2	51.310	19
data4	37 个顶点 102 条服务弧	全图	2	3	3	73.325	43

第 4 条路径:1-2=1, 车辆:8 吨, 路程:2.49, 需求:7.5

第 5 条路径:1-6-29-8-26=14-10-30=31=11=12=13=28-22=21=27-17=14-26=8=29-6-1, 车辆:8 吨, 路程:5.065, 需求:7.392

第 6 条路径:1-6=29-8-26-14-17-27=21=23-37=24=36-8-29-6-1, 车辆:10 吨, 路程:6.122, 需求:9.07

第 7 条路径:1-6-29-8=36=24-37=22=28=13=12=16=15=33-32-14-26-8-29-6=1, 车辆:10 吨, 路程:5.559, 需求:9.725

第 8 条路径:1=5=6-1, 车辆:5 吨, 路程:1.702, 需求:4.678

第 9 条路径:1-6-29=8=26-14=32-34=35=18=19=18=15=11=15-33-32=14=17=27=20-8-29=6-1, 车辆:8 吨, 路程:5.3, 需求:7.948

第 10 条路径:1=2-1, 车辆:8 吨, 路程:2.49, 需求:7.5

第 11 条路径:1-6-29-8-36=23=36=8-29-6-1, 车辆:10 吨, 路程:4.736, 需求:7.775

第 12 条路径:1-5-4=9=10-14-32-33-15=18=21=18=35=34-32-33=32-14=10=9=26-8-29-6-1, 车辆:10 吨, 路程:4.811, 需求:9.015

第 13 条路径:1-6-29-8=20=17=34=17-14=26-8-29-6-1, 车辆:5 吨, 路程:4.103, 需求:4.86

第 14 条路径:1=25=1, 车辆:5 吨, 路程:0.946, 需求:4.256

第 15 条路径:1-2=4=5-1, 车辆:8 吨, 路程:3.827, 需求:7.812

第 16 条路径:1-5=4=2-1, 车辆:8 吨, 路程:3.827, 需求:7.812

第 17 条路径:1-6-29-8-36-24-37-23=21-27=17=20=8-29-6-1, 车辆:10 吨, 路程:6.311, 需求:9.448

第 18 条路径:1-6-29-8-20=27-21=22-28=3=28=22=19=16=19=22=37-24=37-24-36-8-26-9=4-5-1, 车辆:10 吨, 路程:7.397, 需求:9.924

总路程:72.864 公里。2 辆 5 吨车服务 6 条路径(一辆服务第 1、2 和 14 条路径, 总路程 7.422 公里; 另一辆服务第 3、8 和 13 条路径, 总路程 7.507 公里), 3 辆 8 吨车服务 6 条路径(一辆服务第 4 和 5 条路径, 总路程 7.555 公里; 另一辆服务第 9 和 10 条路径, 总路程 7.79 公里; 第三辆服务第 15 和 16 路径, 总路程 7.654 公里), 3 辆 10 吨车服务 6 条路径(一辆服务第 6、7 路径, 总路程 11.681 公里; 另一辆服务第 12 和 17 路径, 总路程 11.122 公里, 第三辆服务第 11、18 条路径, 总路程 12.133 公里)。

当城市的交通道路图比较大时, 以手工的方式为洒水车安排一个合理的作业路线是一件很费时费力的工作, 而要找到一种理想节约的方案更加困难。以上实验结果显示, 尽管随着数据规模的增大算法运行时间有所增加, 但通过本算法仍然可以在很短的时间内寻找到有效理想的安排方案。

根据试点单位提供的经验数据, 其日常洒水降尘作业路线是根据长期作业经验人工安排形成, 每完成一次对所有需要服务道路的喷洒, 车辆总行驶里程在 95 公里以上。而采用本文的进化方法所求得方案可以使这个值下降到 73 公里以下, 按该单位每天对道路喷洒两次计算, 每天可节省 44 公里(节约 23% 里程)的行驶成本。此外使用本算法可提高管理效率, 用户可以根据工作需要随时调整道路洒水要求和安排行驶路线。在更大规模的城市道路图下, 用手工方式寻找优化方案将更加困难, 使用本算法求解将更加有效。

6.2 单车型对比实验

显然, 当各车型的容量值设为一致时就是一个单车型 CARP 问题, 因此本文的 HEGA 算法用来求解普通的单车型 CARP 问题同样适用。为了进一步验证本文算法的先进性, 将国际上求解 CARP 的典型算法 MA 用程序实现, 并对其算法中 local search 部分在不影响算法整体性能的前提下稍作修改, 以适应在有向图下求解 CARP 来与新算法 HEGA 做对比实

验。对比实验仍然在以上的4个数据集上进行,每个数据集上用两种不同吨位对两个算法分别执行5次,取5次的平均测试结果以及平均运行时间。实验结果如表3所示(由于每个数据集中都包含需求量大于5吨的服务弧,因此单车型试验无法用5吨车来进行)。

表3 新算法与 Memetic Algorithm 的对比实验结果

数据规模	实验车辆/吨	实验结果/km		运行时间/s	
		MA	HEGA	MA	HEGA
data1	8	20.216	20.216	6	3
data2	8	32.540	33.330	31	9
data3	8	52.258	51.016	63	18
data4	8	75.365	73.129	197	48
data1	10	18.702	19.152	9	4
data2	10	30.408	30.408	37	13
data3	10	45.830	45.073	103	24
data4	10	65.270	64.350	452	63

通过表3中的实验数据可以看出,在求解单车型 CARP 问题时,新算法 HEGA 在运行效率上比 MA 提高了2到3倍。MA 算法在求解中小规模问题时可以获得比较好的实验结果,然而随着问题规模的增大,使用 MA 求解所需时间急剧增加,而 HEGA 则时间涨幅平稳,且可以获得比 MA 更优的解。总之,新算法 HEGA 不但可以有效地解决多车型 CARP 问题,同时在求解普通的单车型 CARP 问题中同样有着出色的效果,而且无论是在求解精度还是在求解效率上都已经达到甚至超越了 MA 算法。

7 小结

本文提出的 HEGA 算法用一种新颖的思路来对传统 GA 做改进,在求解 CARP 的实验中取得了出色的效果,同时在实践中为解决洒水车路线优化问题提供了很好的解决方案。然

而 CARP 的研究要想进一步符合实际中各种应用场合下的优化需求还必须解决好其它几个关键的扩展问题,比如:多车场 CARP、复杂交通图的 CARP 以及阶段性 CARP (Peoriadic CARP) 等等,这些也是本文作者以后将致力于研究的后续内容。

参考文献:

- [1] Assad A A, Golden B L. Arc routing methods and applications [C]// Balleral M O. Handbooks in OR and MS, Elsevier, Amsterdam, 1995: 375-483.
- [2] Hirabayashi R, Saruwatari Y, Nishida N. Tour construction algorithm for the capacitated arc routing problem [J]. Asia-Pacific Journal of Operational Research, 1992: 155-175.
- [3] Hertz A, Laporte G, Mittaz M. A tabu search heuristic for the capacitated arc routing problem [J]. Operations Research, 2000, 48(1): 129-135.
- [4] 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.
- [5] Lacomme P, Prins C, Ramdane-Cherif W. Competitive memetic algorithms for arc routing problems. Working Paper 1, LOSI, Univ Technologie de Troyes, France, 2001.
- [6] Lacomme P, Prins C, Ramdane-Cherif W. Fast algorithm for general arc routing problems [C]// IFORS 2002 Conference, Edinburgh, UK, 2002.
- [7] Golden B L, Wong R T. Capacitated arc routing problems [J]. Networks, 1981: 305-315.
- [8] Ulusoy G. The fleet size and mix problem for capacitated arc routing [J]. European Journal of Operational Research, 1985, 22: 329-337.
- [9] 玄光男, 程润伟. 遗传算法与工程优化 [M]. 北京: 清华大学出版社, 2004.

(上接 211 页)

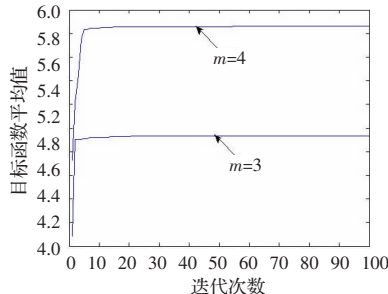


图2 目标函数的平均值随迭代次数变化的过程

4 结论

本文对逢低买入拍卖中的最优定价问题进行定量研究,提出了一种带极值扰动算子的 QPSO 算法对模型进行优化和数值计算。算例表明该算法具有较好的全局收敛能力,可以快速有效地获得模型的满意解。

参考文献:

- [1] 陈剑, 陈熙龙, 宋西平. 拍卖理论与网上拍卖 [M]. 北京: 清华大学出版社, 2005: 125-200.

- [2] Chen Jian, Chen Xi-long, Song Xi-ping. Comparison of the group-buying auction and the fixed pricing mechanism [J]. Decision Support Systems, 2007, 43(2): 445-459.
- [3] 陈剑, 陈熙龙, 宋西平. 规模经济下的逢低买入 [J]. 系统工程理论方法应用, 2003, 12(3): 193-196.
- [4] 陈剑, 陈熙龙, 宋西平. 逢低买入与固定价格机制比较研究 [J]. 管理科学学报, 2003, 6(5): 34-39.
- [5] 毛彦妮, 王刊良. 网上集体议价成交量的实证研究 [J]. 管理学报, 2005, 2(5): 559-563.
- [6] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948.
- [7] Sun Jun, Xu Wen-bo. A global search strategy of quantum-behaved particle swarm optimization [C]// Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, 2004: 111-116.
- [8] 周丹, 须文波, 孙俊, 等. QPSO 算法在非线性的观测器设计中的应用 [J]. 计算机应用研究, 2007, 24(4): 246-248.
- [9] 张春燕, 须文波, 孙俊, 等. MQPSO: 一种具有多群体与多阶段的 QPSO 算法 [J]. 计算机应用研究, 2007, 24(3): 100-102.
- [10] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法 [J]. 软件学报, 2007, 18(4): 861-868.