

考虑废物包装时间的车辆回收路径规划方法

doi:10.3969/j.issn.1674-7100.2021.05.006

邢立宁 吴 健

国防科技大学

系统工程学院

湖南 长沙 410000

摘 要: 针对考虑废物包装时间的车辆回收路径规划问题, 建立问题数学模型, 提出禁忌搜索算法与模因算法求解该问题, 并与爬山算法、遗传算法进行对比。模因算法是爬山算法和遗传算法的结合。实验结果表明: 在解的质量方面, 禁忌搜索算法与模因算法所求出的解的质量要远远好于另外两种算法, 但在运行时间上, 禁忌搜索、爬山算法与遗传算法要远优于模因算法。

关键词: 包装时间; 车辆路径规划; 禁忌搜索; 模因算法

中图分类号: U492; TP18

文献标志码: A

文章编号: 1674-7100(2021)05-0042-07

引文格式: 邢立宁, 吴 健. 考虑废物包装时间的车辆回收路径规划方法[J]. 包装学报, 2021, 13(5): 42-48.

0 引言

随着人们环保意识的增强和一系列新的环境保护法律法规的出台, 绿色物流成为现代物流可持续发展的必然之路。合理规划车辆路径是实现绿色物流的关键环节^[1-3]。

车辆路径规划问题 (vehicle routing problem, VRP) 是指: 合理调度一定数量的车辆通过一系列的收货点和发货点, 在满足相应约束条件 (车辆容量、时间窗口等) 下, 达到运输总费用最低、路程最短等目标^[4-5]。在此基础上, 多个学者延伸出多个问题的变种, 主要包括: 带时间窗的车辆路径规划问题 (vehicle routing problem with time window, VRPTW), 加入服务客户的时窗约束, 即服务客户的时间在一定范围内; 与车型相关的车辆路径规划问题 (mixed fleet vehicle routing problem, MFVRP), 考虑负责运输的车辆属性 (容量、速度) 不同; 带多个车场的车辆路径规划问题 (multi-depot vehicle routing

problem, MDVRP), 是指有多个车场同时为多个客户提供服务, 在满足客户需求的前提下, 使总运输成本最小; 随机车辆路径规划问题 (stochastic vehicle routing problem, SVRP), 涉及运输中遇到的不确定信息, 比如订单到达随机、服务时间不确定等。这些问题多数从收发货点数量、车辆数量、收发货方式、用户需求等角度研究车辆路径规划问题, 较少从物品本身性质来规划车辆路径。根据《中华人民共和国固体废物污染环境防治法》第八十三条之规定“运输危险废物, 应当采取防止污染环境的措施, 并遵守国家有关危险货物运输管理的规定”, 液体、半固体等危险废物需进行适当的包装并贴有危险废物标签后, 才能进行收集、贮存、运输。因此研究废物运输路径规划问题时, 可将包装时间考虑进来。

基本的车辆路径规划问题已经被证明是 NP-Hard 问题, 即不存在多项式时间内求得最优解的算法。因此更复杂的车辆路径规划问题同样不存在多项式时间内可求得精确解的算法^[6]。国内外学者求解车辆

收稿日期: 2021-08-10

作者简介: 邢立宁 (1980-), 男, 陕西临潼人, 国防科技大学研究员, 博士生导师, 主要从事智能优化, 资源调度及任务规划等研究, E-mail: xinglining@gmail.com

通信作者: 吴 健 (1994-), 男, 江苏盐城人, 国防科技大学博士生, 主要研究方向为智能优化算法及其应用, E-mail: 1551699723@qq.com

路径规划问题的方法大致分为精确算法、启发式算法以及元启发式算法3类。精确算法是以分支定界、动态规划为典型的算法^[7-8]。此类算法能从理论上求得最优解,但局限性在于只适用于小规模场景,扩展性不足,并且简化了问题约束,无法满足实际工程需要。启发式算法有距离越近越优先、冲突消解和任务分配等算法,此类算法能够在短时间内生成可行的路径方案,但求解策略较为简单,解质量较低,无法提升资源的使用效率^[9]。元启发式算法是以遗传算法、蚁群算法、粒子群算法为典型的算法^[10-11],此类算法是通过模拟自然界生物种群演化机理和群体行为,对问题进行迭代寻优,能在一定时间内生成较优解,因而在车辆路径规划问题中应用较为广泛。基于此,本文研究废物回收的车辆路径规划问题时,考虑废物包装时间,并构建问题数学模型,设计两种算法即禁忌搜索算法和模因算法进行求解。

1 问题描述与数学模型

考虑废物包装时间的车辆路径规划问题描述如下:1个回收中心点和 N 个回收站点,回收中心点拟派 K 辆车回收废物,所有车辆均从回收中心点出发,最后到达回收中心点。该问题可用 $G=(V, E)$ 的有向连通图表示(见图1), V 表示回收中心点和回收站点的集合, E 表示图中所有边的集合。图1中有3条车辆行驶路径,分别为 $0 \rightarrow 2 \rightarrow 7 \rightarrow 0$, $0 \rightarrow 6 \rightarrow 1 \rightarrow 0$, $0 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 0$ 。

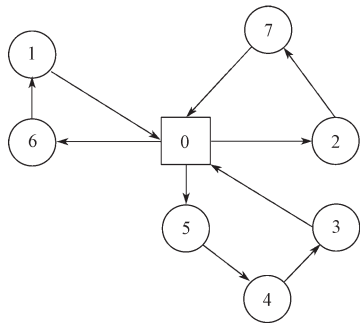


图1 车辆路径规划示意图

Fig. 1 Vehicle route diagram

本车辆路径规划问题的数学模型如下:

1) 目标函数即车辆行驶总路程最短为

$$\min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}, \quad (1)$$

式中: c_{ij} 为站点 i 与 j 之间的距离; x_{ijk} 为模型的决策变量,取值为0或1, $x_{ijk}=1$ 表示第 k 辆车服务完站

点 i 之后立马服务站点 j ,否则为0。为了简化计算,假设车辆行驶速度为1,则车辆从站点 i 到站点 j 的时间 t_{ij} 与距离 c_{ij} 相等。

2) 每个回收站点只被服务一次,

$$\sum_{k \in K} \sum_{i \in N} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in N. \quad (2)$$

3) 车辆均从回收中心点出发,最后到达回收中心点,即

$$\sum_{(0,j) \in E} x_{0jk} = 1, \quad \forall k \in K, \quad (3)$$

$$\sum_{(i,0) \in E} x_{i0k} = 1, \quad \forall k \in K. \quad (4)$$

4) 时间约束为

$$w_{ik} + s_i + t_{ij} \leq b_j, \quad \forall k \in K, x_{ijk} = 1; \quad (5)$$

$$w_{jk} = \max\{w_{ik} + s_i + t_{ij}, a_j\}, \quad \forall k \in K, x_{ijk} = 1. \quad (6)$$

式中: s_i 为废物包装的时间; w_{ik} 、 w_{jk} 分别为第 k 辆车在第 i 个和第 j 个站点的开始包装时间; a_j 、 b_j 分别为站点 j 接受废物回收的最早、最晚时间。当车辆到达站点 j 的时间小于 a_j ,则必须等待;当车辆到达站点 j 的时间超过 b_j ,则需放弃该站点。

5) 车辆容量约束为

$$\sum_{i \in N} d_i \sum_{j \in N} x_{ijk} \leq C_k, \quad \forall k \in K. \quad (7)$$

式中: d_i 为站点 i 的废物容量; C_k 为第 k 辆车容量。

2 算法设计

2.1 禁忌搜索算法

禁忌搜索(tabu search, TS)算法是由Glover于1986年提出的一种带有记忆策略的局部搜索算法。禁忌算法以传统爬山算法为基础开展搜索优化,并通过禁忌表记录优化过程中的局部最优解或产生局部最优解的操作,以避免对局部最优空间的重复搜索,达到跳出局部最优、开辟优质解空间的效果。

1) 禁忌长度

在禁忌搜索算法中,禁忌长度即禁忌解集的大小是影响算法性能的主要因素。为增强算法在不同问题规模的适应能力,设禁忌长度为任务集 T 规模的某一比例。禁忌对象直接为解,即回收路径总长度。

$$|U_T| = a_T \times |T|, \quad (8)$$

式中 a_T 为比例系数,取值为0.1~0.3。

2) 编码方式

编码采取实数编码方式。每辆车的行驶路径

单独编码,该编码方式更加简洁直观。两辆车的行驶路径编码如图2所示,车辆1的行驶路径为 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 0$, 车辆2的行驶路径为 $0 \rightarrow 4 \rightarrow 6 \rightarrow 0$ 。

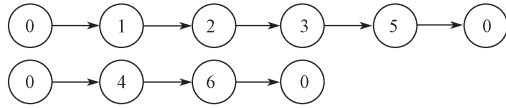


图2 解的编码

Fig. 2 The coding of the solution

3) 插入算子

采用插入算子产生新解,如图3所示。图中,选择路径1中回收站点3插入到路径2中回收站点6的后面,以产生新解。插入算子完全遍历当前解得到的解集即为邻域。选取邻域最好解或者非禁忌最好解作为下一迭代的当前解。

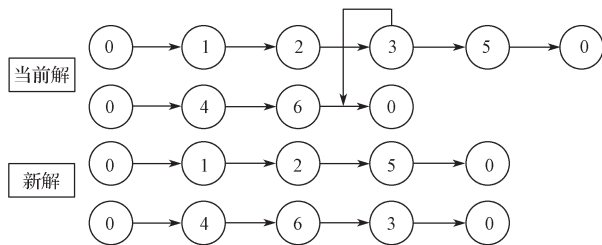


图3 插入算子

Fig. 3 Insertion operator

禁忌搜索算法流程如图4所示。

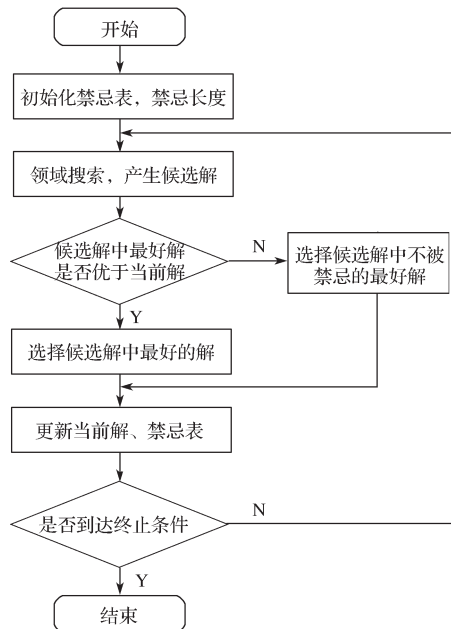


图4 禁忌搜索算法流程图

Fig. 4 The flow chart of tabu search algorithm

2.2 模因算法

采用遗传算法与局部搜索算法相结合的模因算法解决废物回收车辆路径规划问题。遗传算法能够保证解的多样性,但不能保证求得一个较优解,而局部搜索算法即爬山算法可在遗传算法的基础上,对解进行邻域搜索,实现解的多样性与集中性。算法流程如图5所示。

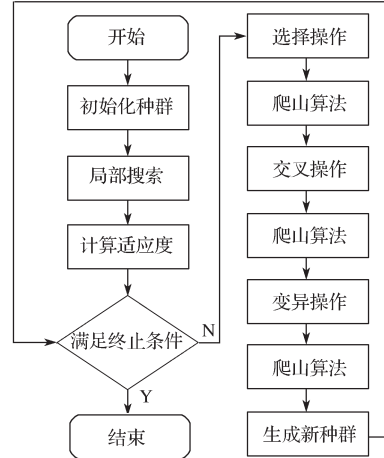


图5 模因算法流程图

Fig. 5 The flow chart of memetic algorithm

2.2.1 遗传算法

1) 编码方式

个体编码采取实数编码方式。为了方便遗传算法的交叉变异操作,采取与图3不同的的编码方式,将所有车辆的行驶路径进行统一编码,当作一个个体,如图6所示。图中,节点2,3为一条路径,节点4为一条路径,节点5,1,6为一条路径,其中0表示路径的起点和终点。

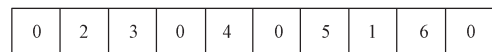


图6 解的编码

Fig. 6 The coding of the solution

2) 选择策略

选择策略采用轮盘赌策略。假设共有 N 个个体,第 i 个个体的适应度为 $fits(i)$, 第 i 个个体被选择的概率为

$$p(i) = \frac{fits(i)}{\sum_{i=1}^N fits(i)} \quad (9)$$

由式(9)可知,适应度越高的个体被选中进入下一

代的概率越大。

3) 交叉策略

交叉策略采取部分交叉映射 (partially mapped crossover, PMX)。种群中的个体随机进行两两配对, 配对成功的两个个体作为父代 1 和父代 2 进行交叉操作。首先选两个交叉点, 交换中间部分, 确定映射关系, 最后将未换部分按映射关系恢复合法性, 具体操作如图 7 所示。

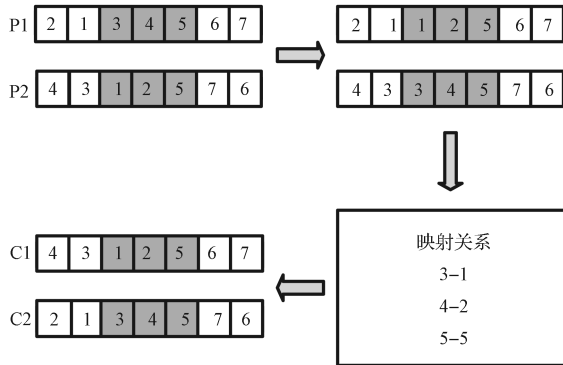


图 7 交叉策略

Fig. 7 Crossover strategy

完成交叉策略后, 随机选择几个位置, 将表示起点和终点的 0 随机插入, 以完成了两个完整解的交叉操作。本实验中, 交叉率设置为 95%。

4) 变异策略

变异策略采用两点互换。随机生成两个基因位, 并交换两个基因位上的基因, 如图 8 所示。本实验中, 变异率设置为 10%。

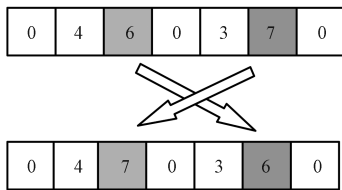


图 8 变异策略

Fig. 8 Mutation strategy

2.2.2 爬山算法

爬山算法是从当前的节点开始, 和周围邻居节点的值进行比较, 然后不断向有提升的方向前进。本文采取首次改进 (first-improvement) 策略的爬山算法。首次改进策略是接受搜索过程中出现的第一个改进解。如当前解为 $[0, 2, 3, 0, 4, 0, 5, 1, 6, 0]$, 由于开始位置与结束位置都必须为 0, 因此从第二个节点开始尝试两两节点依次进行交换, 即 $(2, 3), (2, 0), \dots, (2, 6)$,

$(3, 0), \dots, (1, 6)$, 交换后计算解的提升情况, 如果有提升, 则接受此解。如 2 和 6 交换可得到有提升的解, 则当前解变成 $[0, 6, 3, 0, 4, 0, 5, 1, 2, 0]$ 。此时下一轮搜索序列变更为 $(6, 3), (6, 0), \dots, (6, 1), (3, 0), \dots, (1, 2)$, 重新开始搜索并接受第一个改进解。爬山算法的终止条件设置为当某次提升后在所有邻域中都找不到改进解。

3 案例分析

某制造企业为提高资源利用率, 给社会环境和企业带来可观的经济效益, 需对某产品的废物进行回收。根据前期市场售卖产品情况, 100 个站点有产品废物, 为此该企业建立了 1 个回收中心点。各站点的地理位置、废物质量以及所需包装时间见附表 1, 其中节点 0 为回收中心点, 剩余 100 个节点为回收站点。

3.1 禁忌搜索算法结果

禁忌搜索算法参数设置如下: 总迭代次数为 2000, 禁忌长度分别为 10, 20, 40。禁忌搜索算法运行 10 次, 最好解如下:

回收路径 1: $0 \rightarrow 45 \rightarrow 83 \rightarrow 99 \rightarrow 94 \rightarrow 96 \rightarrow 0$;

回收路径 2: $0 \rightarrow 27 \rightarrow 31 \rightarrow 88 \rightarrow 7 \rightarrow 0$;

回收路径 3: $0 \rightarrow 40 \rightarrow 53 \rightarrow 26 \rightarrow 0$;

回收路径 4: $0 \rightarrow 62 \rightarrow 11 \rightarrow 90 \rightarrow 10 \rightarrow 0$;

回收路径 5: $0 \rightarrow 2 \rightarrow 21 \rightarrow 73 \rightarrow 41 \rightarrow 56 \rightarrow 4 \rightarrow 0$;

回收路径 6: $0 \rightarrow 14 \rightarrow 44 \rightarrow 38 \rightarrow 43 \rightarrow 58 \rightarrow 0$;

回收路径 7: $0 \rightarrow 36 \rightarrow 47 \rightarrow 19 \rightarrow 8 \rightarrow 46 \rightarrow 17 \rightarrow 0$;

回收路径 8: $0 \rightarrow 12 \rightarrow 76 \rightarrow 78 \rightarrow 34 \rightarrow 35 \rightarrow 77 \rightarrow 0$;

回收路径 9: $0 \rightarrow 65 \rightarrow 71 \rightarrow 9 \rightarrow 66 \rightarrow 1 \rightarrow 0$;

回收路径 10: $0 \rightarrow 63 \rightarrow 64 \rightarrow 59 \rightarrow 48 \rightarrow 0$;

回收路径 11: $0 \rightarrow 28 \rightarrow 29 \rightarrow 79 \rightarrow 50 \rightarrow 68 \rightarrow 0$;

回收路径 12: $0 \rightarrow 39 \rightarrow 23 \rightarrow 67 \rightarrow 55 \rightarrow 25 \rightarrow 0$;

回收路径 13: $0 \rightarrow 33 \rightarrow 81 \rightarrow 3 \rightarrow 54 \rightarrow 24 \rightarrow 80 \rightarrow 0$;

回收路径 14: $0 \rightarrow 69 \rightarrow 30 \rightarrow 51 \rightarrow 20 \rightarrow 32 \rightarrow 70 \rightarrow 0$;

回收路径 15: $0 \rightarrow 95 \rightarrow 98 \rightarrow 61 \rightarrow 86 \rightarrow 91 \rightarrow 100 \rightarrow 0$;

回收路径 16: $0 \rightarrow 82 \rightarrow 18 \rightarrow 84 \rightarrow 60 \rightarrow 89 \rightarrow 0$;

回收路径 17: 0 → 72 → 75 → 22 → 74 → 0;

回收路径 18: 0 → 52 → 6 → 0;

回收路径 19: 0 → 92 → 42 → 15 → 87 → 57 → 97 → 0;

回收路径 20: 0 → 59 → 5 → 16 → 85 → 37 → 93 → 0。

3.2 模因算法结果

模因算法、遗传算法、爬山算法的迭代次数设置为 100 代，模因算法、遗传算法的种群规模设置为 100，交叉率为 95%，变异率为 10%。通过先验实验，发现在 100 次迭代过程中加入爬山算法与在最后迭代的 10 代中加入爬山算法的结果相差不大，但在最后迭代的 10 代中加入爬山算法时，算法时间有了巨大的提升。因此，本文选择在最后迭代的 10 代中加入爬山算法。模因算法运行 10 次，其中一个解的表达方式如下：

0 → 96 → 13 → 92 → 76 → 80 → 54 → 0 → 21 → 81 → 78 → 33 → 24 → 55 → 93 → 0 → 28 → 4 → 20 → 66 → 65 → 9 → 68 → 75 → 72 → 37 → 16 → 84 → 85 → 2 → 26 → 8 → 49 → 70 → 6 → 59 → 7 → 18 → 77 → 58 → 97 → 60 → 32 → 35 → 12 → 71 → 79 → 34 → 29 → 30 → 63 → 62 → 43 → 57 → 23 → 74 → 0 → 52 → 22 → 73 → 67 → 39 → 46 → 86 → 38 → 98 → 61 → 95 → 88 → 11 → 64 → 90 → 17 → 47 → 19 → 69 → 87 → 56 → 40 → 89 → 83 → 27 → 45 → 36 → 3 → 0 → 100 → 44 → 42 → 5 → 1 → 51 → 50 → 91 → 15 → 94 → 99 → 14 → 82 → 31 → 25 → 41 → 10 → 48 → 53 → 0。

该解可解码为：

回收路径 1: 0 → 96 → 13 → 92 → 76 → 80 → 54 → 0;

回收路径 2: 0 → 21 → 81 → 78 → 33 → 24 → 55 → 93 → 0;

回收路径 3: 0 → 28 → 4 → 20 → 66 → 65 → 9 → 68 → 75 → 72 → 37 → 16 → 84 → 85 → 2 → 26 → 8 → 49 → 70 → 6 → 59 → 7 → 18 → 77 → 58 → 97 → 60 → 32 → 35 → 12 → 71 → 79 → 34 → 29 → 30 → 63 → 62 → 43 → 57 → 23 → 74 → 0;

回收路径 4: 0 → 52 → 22 → 73 → 67 → 39 → 46 → 86 → 38 → 98 → 61 → 95 → 88 → 11 → 64 → 90 → 17 → 47 → 19 → 69 → 87 → 56 → 40 → 89 → 83 → 27 → 45 → 36 → 3 → 0;

回收路径 5: 0 → 100 → 44 → 42 → 5 → 1 → 51 → 50 → 91 → 15 → 94 → 99 → 14 → 82 → 31 → 25 → 41 → 10 → 48 → 53 → 0。

爬山算法、遗传算法和模因算法的算法运行时间和距离如表 1 所示，距离对比如图 9 所示。由表 1 和图 9 可知，在算法运行时间方面，模因算法相较于另外两种算法表现较差，但在求解效果上，每种算法的表现均较稳定，在解的质量方面，模因算法所求解远远好于遗传算法和爬山算法，普遍提升了 30% 以上。

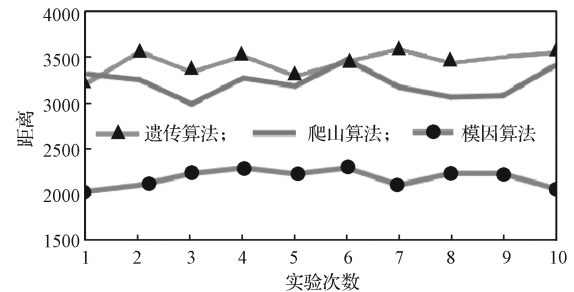


图 9 距离对比

Fig. 9 Comparison of distance

表 1 算法结果对比

Table 1 Comparison of algorithm results

名 称	运行时间 /s										平均值	标准差
	1	2	3	4	5	6	7	8	9	10		
爬山算法	2.46	2.48	2.45	2.47	2.52	2.54	2.58	2.45	2.54	2.53	2.50	0.046
遗传算法	0.24	0.23	0.22	0.24	0.25	0.26	0.23	0.22	0.23	0.22	0.23	0.011
模因算法	131.37	120.00	106.00	159.16	147.29	145.44	145.71	180.52	102.50	166.97	140.49	25.570

名 称	距 离										平均值	标准差
	1	2	3	4	5	6	7	8	9	10		
爬山算法	3310	3248	2983	3267	3179	3469	3167	3056	3071	3399	3214.9	154.43
遗传算法	3216	3556	3331	3510	3287	3445	3581	3430	3497	3544	3439.7	123.78
模因算法	2023	2093	2220	2285	2216	2284	2092	2222	2218	2058	2171.1	95.46

4 结语

本文构建了考虑废物包装时间的车辆回收路径问题的模型,并设计了禁忌搜索算法与模因算法求解该问题。结果表明:在算法运行时间上,模因算法稍逊色一点,但在求解质量上,禁忌搜索算法与模因算法均优于爬山算法、遗传算法。

在未来的研究工作中,将在现有算法框架的基础上,一方面集成更多的进化算法(蚁群算法、粒子群算法)与局部搜索算法(模拟退火算法等),另一方面考虑多种不同性质的物品包装,以丰富车辆回收路径问题的研究方法。

参考文献:

- [1] LIN C H, CHOY K L, HO G T S, et al. Survey of Green Vehicle Routing Problem: Past and Future Trends[J]. *Expert Systems with Applications*, 2014, 41(4): 1118-1138.
- [2] DEVIKA K, JAFARIAN A, NOURBAKHS V. Designing a Sustainable Closed-Loop Supply Chain Network Based on Triple Bottom Line Approach: A Comparison of Metaheuristics Hybridization Techniques[J]. *European Journal of Operational Research*, 2014, 235(3): 594-615.
- [3] RAMOS T R P, GOMES M I, BARBOSA-PÓVOA A P. Planning a Sustainable Reverse Logistics System: Balancing Costs with Environmental and Social Concerns[J]. *Omega*, 2014, 48: 60-74.
- [4] 孙丽君, 胡祥培, 王 征. 车辆路径规划问题及其求解方法研究进展 [J]. *系统工程*, 2006, 24(11): 31-37.
SUN Lijun, HU Xiangpei, WANG Zheng. Reviews on Vehicle Routing Problem and Its Solution Methods[J]. *Systems Engineering*, 2006, 24(11): 31-37.
- [5] 胡运权. 运筹学教程 [M]. 2 版. 北京: 清华大学出版社, 2003: 80-94.
HU Yunquan. *Operations Research Course*[M]. 2nd ed. Beijing: Tsinghua University Press, 2003: 80-94.
- [6] 符 卓. 带装载能力约束的开放式车辆路径问题及其禁忌搜索算法研究 [J]. *系统工程理论与实践*, 2004, 24(3): 123-128.
FU Zhuo. The Capacitated Open Vehicle Routing Problem and Its Tabu Search Algorithm[J]. *Systems Engineering-Theory & Practice*, 2004, 24(3): 123-128.
- [7] 刘 洋, 陈 璐. 养护车辆路径规划的鲁棒性优化方法 [J]. *上海交通大学学报*, 2018, 52(4): 388-394.
LIU Yang, CHEN Lu. A Robust Optimization Approach for the Routing Problem of Road Network Daily Maintenance[J]. *Journal of Shanghai Jiao Tong University*, 2018, 52(4): 388-394.
- [8] PRINS C. A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem[J]. *Computers & Operations Research*, 2004, 31(12): 1985-2002.
- [9] GAUR D R, MUDGAL A, SINGH R R. Improved Approximation Algorithms for Cumulative VRP with Stochastic Demands[J]. *Discrete Applied Mathematics*, 2020, 280: 133-143.
- [10] 郎茂祥, 胡思继. 车辆路径问题的禁忌搜索算法研究 [J]. *管理工程学报*, 2004, 18(1): 81-84.
LANG Maoxiang, HU Siji. Study on the Tabu Search Algorithm for Vehicle Routing Problem[J]. *Journal of Industrial Engineering and Engineering Management*, 2004, 18(1): 81-84.
- [11] ZHENG L, HE Z M, LIANG W. VRP Problem Solving Based on Adaptive Dynamic Search Ant Colony Algorithm[J]. *Journal of Physics: Conference Series*, 2020, 1487: 012030.

(责任编辑: 邓 彬)

The Planning Method of Vehicle Routing Problem Considering Waste Packaging Time

XING Lining, WU Jian

(School of System Engineering, National University of Defense Technology, Changsha 410000, China)

Abstract: To solve the problem of vehicle routing with waste packaging time, firstly the mathematical model of the problem was established, the tabu search algorithm and memetic algorithm were proposed to solve the problem, and the hill climbing algorithm and the genetic algorithm were selected for comparison. The memetic algorithm was the combination of the hill climbing algorithm and the genetic algorithm. The experiment showed that: in terms of the quality of the solution, the tabu search algorithm and the memetic algorithm were much better than the other two algorithms, but in terms of running time, the tabu search algorithm, the hill climbing algorithm and the genetic algorithm were far superior to the memetic algorithm.

Keywords: packaging time; vehicle routing problem; tabu search algorithm; memetic algorithm

附表 1 回收点的基本数据

Table 1 Basic data of recycling site

序号	横坐标	纵坐标	质量 / kg	开始 时间 /min	结束 时间 /min	包装 时间 /min	序号	横坐标	纵坐标	质量 / kg	开始 时间 /min	结束 时间 /min	包装 时间 /min
0	35	35	0	0	230	0	51	49	58	10	88	98	10
1	41	49	10	161	171	10	52	27	43	9	52	62	10
2	35	17	7	50	60	10	53	37	31	14	95	105	10
3	55	45	13	116	126	10	54	57	29	18	140	150	10
4	55	20	19	149	159	10	55	63	23	2	136	146	10
5	15	30	26	34	44	10	56	53	12	6	130	140	10
6	25	30	3	99	109	10	57	32	12	7	101	111	10
7	20	50	5	81	91	10	58	36	26	18	200	210	10
8	10	43	9	95	105	10	59	21	24	28	18	28	10
9	55	60	16	97	107	10	60	17	34	3	162	172	10
10	30	60	16	124	134	10	61	12	24	13	76	86	10
11	20	65	12	67	77	10	62	24	58	19	58	68	10
12	50	35	19	63	73	10	63	27	69	10	34	44	10
13	30	25	23	159	169	10	64	15	77	9	73	83	10
14	15	10	20	32	42	10	65	62	77	20	51	61	10
15	30	5	8	61	71	10	66	49	73	25	127	137	10
16	10	20	19	75	85	10	67	67	5	25	83	93	10
17	5	30	2	157	167	10	68	56	39	36	142	152	10
18	20	40	12	87	97	10	69	37	47	6	50	60	10
19	15	60	17	76	86	10	70	37	56	5	182	192	10
20	45	65	9	126	136	10	71	57	68	15	77	87	10
21	45	20	11	62	72	10	72	47	16	25	35	45	10
22	45	10	18	97	107	10	73	44	17	9	78	88	10
23	55	5	29	68	78	10	74	46	13	8	149	159	10
24	65	35	3	153	163	10	75	49	11	18	69	79	10
25	65	20	6	172	182	10	76	49	42	13	73	83	10
26	45	30	17	132	142	10	77	53	43	14	179	189	10
27	35	40	16	37	47	10	78	61	52	3	96	106	10
28	41	37	16	39	49	10	79	57	48	23	92	102	10
29	64	42	9	63	73	10	80	56	37	6	182	192	10
30	40	60	21	71	81	10	81	55	54	26	94	104	10
31	31	52	27	50	60	10	82	15	47	16	55	65	10
32	35	69	23	141	151	10	83	14	37	11	44	54	10
33	53	52	11	37	47	10	84	11	31	7	101	111	10
34	65	55	14	117	127	10	85	16	22	41	91	101	10
35	63	65	8	143	153	10	86	4	18	35	94	104	10
36	2	60	5	41	51	10	87	28	18	26	93	103	10
37	20	20	8	134	144	10	88	26	52	9	74	84	10
38	5	5	16	83	93	10	89	26	35	15	176	186	10
39	60	12	31	44	54	10	90	31	67	3	95	105	10
40	40	25	9	85	95	10	91	15	19	1	160	170	10
41	42	7	5	97	107	10	92	22	22	2	18	28	10
42	24	12	5	31	41	10	93	18	24	22	188	198	10
43	23	3	7	132	142	10	94	26	27	27	100	110	10
44	11	14	18	69	79	10	95	25	24	20	39	49	10
45	6	38	16	32	42	10	96	22	27	11	135	145	10
46	2	48	1	117	127	10	97	25	21	12	133	143	10
47	8	56	27	51	61	10	98	19	21	10	58	68	10
48	13	52	36	165	175	10	99	20	26	9	83	93	10
49	6	68	30	108	118	10	100	18	18	17	185	195	10
50	47	47	13	124	134	10							

注：在不影响实验结果的前提下，假设回收站点的坐标都是在二维直角坐标系中。