

包装物回收物流中的车辆路径优化问题

张 异

(重庆工商职业学院, 重庆 401520)

摘要: **目的** 提高遗传算法 (GA) 求解包装物回收车辆路径优化问题的性能。**方法** 通过对传统 GA 算法的改进, 提出混合蜂群遗传算法 (HBGA)。首先改进传统 GA 算法的初始种群生成方式, 设计初始种群混合生成算子; 其次, 提出最大保留交叉算子, 对优秀子路径进行保护; 然后, 在上述改进的基础上引入蜜蜂进化机制, 用以保证种群多样性和优秀个体特征信息的利用程度; 最后, 对标准算例集进行仿真测试。**结果** 与传统 GA 算法相比, HBGA 算法在全局寻优能力、算法稳定性和运行速度方面均有所改善。HBGA 算法的全局寻优能力和算法稳定性均优于粒子群算法 (PSO)、蚁群算法 (ACO) 和禁忌搜索算法 (TS), 但运行速度稍慢于 TS 算法。**结论** 对传统 GA 算法的改进是合理的, 且 HBGA 算法整体求解性能优于 PSO 算法、ACO 算法和 TS 算法。

关键词: 包装物回收; 路径优化; 遗传算法; 蜜蜂进化机制

中图分类号: TB485.3 **文献标识码:** A **文章编号:** 1001-3563(2017)17-0233-06

Vehicle Routing Optimization Problem in Package Recycling Logistics

ZHANG Yi

(Chongqing Technology and Business Institute, Chongqing 401520, China)

ABSTRACT: The work aims to improve the performance of genetic algorithm (GA) to solve the vehicle routing optimization problem in the package recycling. Based on the improvement of traditional genetic algorithm, the hybrid bee genetic algorithm (HBGA) was put forward. Firstly, the initial population generation method of the traditional GA was improved, and the mixed generation operator of the initial population was designed; secondly, the maximum reservation crossover operator was proposed to protect the excellent sub-path; then, on the basis of the above-mentioned improvement, the bee evolutionary mechanism was introduced to ensure the population diversity and the utilization of the excellent individual characteristic information. Finally, the simulation tests were carried out on a standard example set. Compared with the traditional GA, the HBGA was improved regarding its global optimization ability, algorithm stability and running speed. In addition, the global optimization ability and stability of the HBGA were superior to the particle swarm optimization (PSO) algorithm, ant colony optimization (ACO) algorithm and tabu search (TS) algorithm, but its running speed was slightly slower than the TS algorithm. The improvement of traditional GA is reasonable, and the overall solution performance of HBGA is better than the PSO algorithm, ACO algorithm and TS algorithm.

KEY WORDS: package recycling; routing optimization; genetic algorithm; bee evolutionary mechanism

近年来, 我国经济建设取得了快速发展, 但同时也面临着资源被过度消耗的严峻挑战, 发展循环经济成为了社会各界的共识。在此背景下, 一些新的物流管理方式和运作模式应运而生, 包装物回收物流就是其中的重要组成部分^[1]。为了推动包装物回收

物流业的发展, 我国政府相继出台了《包装废弃物的处理与利用通则》等法律法规, 明确了包装物回收对国家经济未来发展的重大战略意义。由此可见, 回收体系的建立、回收中心的选址、回收路径的优化等包装物回收物流方面的问题必然会成为理论界和实践

收稿日期: 2016-12-20

基金项目: 重庆市教委人文社科项目 (16SKGH209); 重庆工商职业学院重点项目 (ZD2014-03)

作者简介: 张异 (1973—), 女, 重庆工商职业学院副教授, 主要研究方向为供应链管理、物流管理。

界的研究热点。

遗传算法 (GA) 由 Holland 教授于 1975 年首次提出, 是一种由进化论和遗传学说演化而来的智能算法^[2]。该算法自提出以来就受到学者们的重点关注, 被广泛应用到组合优化、机器学习、信号处理、人工智能等领域^[3]。GA 算法由于拥有良好的并发性和局部搜索能力, 成为了车辆路径优化问题的主要求解算法之一^[4-6]。然而, 根据仿真实验可知, 传统 GA 算法^[2]在求解包装物回收等规模较大的车辆路径优化问题时, 比较容易陷入局部最优解且运行速度较慢, 算法性能有待进一步提高。经过分析, 导致 GA 算法出现上述问题的因素主要有 3 个: 初始种群质量, 根据文献^[7], 高质量的初始种群能够提高 GA 算法的运行速度及解的质量, 尤其对于大规模问题, 由于搜索空间非常之大, 好的初始种群尤为重要; 种群多样性, 与人类进化机制相同, 相似个体进行交叉, 其子个体的性能往往比父辈差, 因此在算法进化过程中, 随着种群多样性的降低, 算法求解性能也受到很大的影响; 优秀个体特征信息的利用程度, GA 算法的求解目标是在最短的时间内求得最优解, 因此算法结束时人们通常只关注种群中最优个体的性能, 从概率上讲, 种群中优秀个体与最优解之间的亲和度要大于其他个体与最优解之间的亲和度, 而且与优秀个体亲和度较大的个体, 其适应度也普遍较高^[8], 因此, 优秀个体的特征信息能否被充分利用, 也是决定算法求解性能的一个关键性因素。

基于上述分析, 这里对包装物回收物流中的车辆路径优化问题进行研究, 提出包装物回收物流中的车辆路径优化模型及求解模型的混合蜂群遗传算法 (HBGA)。首先, 设计初始种群混合生成算子, 提高初始种群质量; 其次, 提出最大保留交叉算子, 最大程度地保护优秀子路径, 进而加快算法运行速度; 然后, 引入蜜蜂进化机制, 利用蜜蜂繁殖的“外来种群”来保证种群多样性, 并通过最优个体辅助种群进化的方式提高优秀个体特征信息的利用程度。

1 数学模型的建立

1.1 问题描述

包装物回收物流中的路径优化问题可描述为^[9]: 回收中心派遣一定数量的车辆去回收客户的包装物, 车辆从回收中心出发, 完成任务后返回回收中心, 目标是在车辆总行驶距离最短的情况下, 寻找能够回收所有客户包装物的行驶路线。主要约束如下所述。

1) 回收中心约束, 所有车辆均从同一回收中心出发, 任务完成后返回回收中心。

2) 访问唯一性约束, 每个客户均被访问且只能被访问 1 次。

3) 车辆能力约束, 每条行驶路径上的包装物回收量之和不超过车辆的承载质量。

1.2 符号说明

V 为节点集, $V=\{0, 1 \dots n\}$, 其中 0 表示回收中心, 其余节点表示客户; d_{ij} 为客户 i 到客户 j 的距离 (在此为 2 点之间的直线距离); K 为车辆集合, $K=\{1, 2 \dots m\}$, m 为完成回收任务所需车辆数; m_Q 为车辆的最大承载质量; m_i 为客户 i 需要回收的包装物质量。

1.3 模型建立

根据该问题的实际情况, 特设定决策变量如下:

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 完成客户 } i \text{ 的回收任务后前往客户 } j \\ 0, & \text{其他情况} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{车辆 } k \text{ 执行客户 } i \text{ 的回收任务} \\ 0, & \text{其他情况} \end{cases}$$

$$f_{\min} = \sum_{k \in K} \sum_{i \in V'} \sum_{j \in V'} d_{ij} x_{ijk} \quad (1)$$

约束条件:

$$\sum_{i \in V' \setminus \{0\}} m_i \sum_{j \in V'} x_{ijk} \leq m_Q, \forall k \in K \quad (2)$$

$$\sum_{k \in K} \sum_{i \in V'} x_{ijk} = 1, \forall j \in V' \setminus \{0\} \quad (3)$$

$$\sum_{k \in K} \sum_{j \in V'} x_{ijk} = 1, \forall i \in V' \setminus \{0\} \quad (4)$$

$$\sum_{i \in V' \setminus \{0\}} x_{i0k} = \sum_{j \in V' \setminus \{0\}} x_{0jk} = 1, \forall k \in K \quad (5)$$

$$\sum_{k \in K} \sum_{i \in V' \setminus \{0\}} x_{i0k} = \sum_{k \in K} \sum_{j \in V' \setminus \{0\}} x_{0jk} = m \quad (6)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S} x_{ijk} \geq 1, \forall S \subseteq V' \setminus \{0\}, |S| \geq 2 \quad (7)$$

$$\sum_{j \in V'} x_{ijk} = \sum_{j \in V'} x_{jik}, \forall k \in K, \forall i \in V' \setminus \{0\} \quad (8)$$

$$x_{ijk} (x_{ijk} - 1) = 0, \forall i, j \in V', \forall k \in K \quad (9)$$

$$y_{ik} (y_{ik} - 1) = 0, \forall i \in V', \forall k \in K \quad (10)$$

其中, 式 (1) 表示行驶路径最短的目标函数; 式 (2) 表示车辆能力约束; 式 (3—4) 表示客户访问唯一性约束; 式 (5—6) 表示配送中心约束; 式 (7) 表示子回路约束; 式 (8) 表示客户车辆流守恒约束; 式 (9—10) 表示 0~1 变量约束。

2 混合蜂群遗传算法

根据前述研究可知, 由于受初始种群质量、种群多样性和优秀个体特征信息利用程度的限制^[7-8], 传统 GA 算法较容易出现“早熟”的问题, 而且运行速度较慢。对此, 这里对传统 GA 算法进行了对应的改进, 主要包括: 利用基于扫描算法和节约算法的混合算法生成初始种群, 提高初始种群的质量; 提出最大保留交叉算子, 保护优秀子路径; 引入蜜蜂进化机制,

提出 HBGA 算法。

2.1 编码方式

采用文献[10]的自然数编码, 将每个回收方案表达为 $(0, i_1, i_2, \dots, i_e, 0, i_f, \dots, i_k, 0, \dots, 0, i_p, \dots, i_n, 0)$ 的形式。其中, 0 表示回收中心, i_j 表示该路径上的第 j 个客户。

2.2 初始种群生成

目前, 很少有学者专门对 GA 算法的初始种群生成方式进行改进, 导致 GA 算法的初始种群中普遍含有大量不可行个体和重复个体, 严重影响了算法的求解性能。针对该问题, 这里提出初始种群混合生成算子。

1) 以回收中心为极点, 回收中心与任一客户的连线为极轴建立极坐标体系。

2) 逆时针扫描客户点, 按车辆承载质量对客户分群。

3) 在各客户群中, 利用客户点与回收中心构建相应的回收子网络。

4) 在行驶距离最短的目标下, 采用节约算法^[11]对回收子网络进行优化, 形成以回收中心为起始点的回收子路径。

5) 不同子路径组合成一个染色体。

6) 极轴随机偏转一个角度, 重复步骤 2 至 4, 直到生成种群规模一半数量的个体。

7) 顺时针扫描客户点进行分群, 重复步骤 3 至

4, 直到生成另一半数量的个体。

2.3 适应度函数

适应度值是衡量个体优劣的重要标准, 这里选取式 (1) 的倒数作为适应度函数。

2.4 选择算子

采用精英保留和轮盘赌选择的混合算子方法。首先, 将所有个体按适应度值从小到大排序, 选择最优秀个体直接进入算法下一步骤; 其次, 对剩余的个体进行轮盘赌选择, 按个体的适应度划分赌盘, 使优秀个体进入下一代的可能性更高。

2.5 最大保留交叉算子

传统交叉算子不但会对优秀子路径造成破坏, 影响算法的运行速度, 而且当 2 个父个体相同时, 还不利于新个体的产生, 进而降低种群的多样性^[10, 12]。对此, 这里提出一种最大保留交叉算子, 见图 1。

1) 最优子路径复制。计算父个体 A 中各子路径的车载率, 将车载率最高的子路径复制到父个体 B 对应的临时串 string_ B 的首部。

2) 剩余子路径确定。删除 B 中与 string_ B 相同的客户点, 按照初始种群生成算法中的步骤 1 至 4 确定剩余子路径, 将其与 string_ B 合并, 新的 string_ B 即为交叉后的子个体 B' 。同样的方法可以得到交叉后的子个体 A' 。

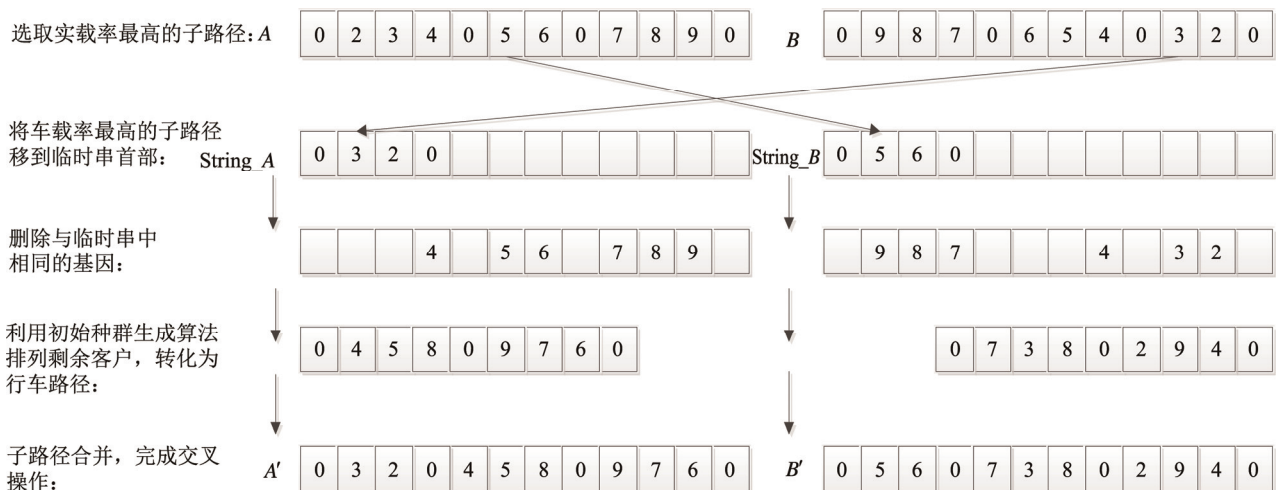


图 1 最大保留交叉算子

Fig.1 The maximum reservation crossover operator

2.6 变异算子

随机选取 2 个点作为变异点, 若 2 个变异点均不是 0 基因 (回收中心), 则交换其位置, 完成变异操作。反之, 则重新选择变异点。变异算子的选取见图 2。

2.7 混合蜂群遗传算法

根据蜜蜂的繁殖机制可知^[8], 通过选择遗传种群

中最优个体作为蜂后, 将一定数量的优秀个体作为雄蜂, 通过促进蜂后与雄蜂进行交叉的方式来提高优秀个体特征信息的利用程度。此外, 还可以随机生成种群, 用以模拟外来蜂群, 保证算法进化过程中的种群多样性。基于此, 这里提出 HBGA 算法, 具体步骤如下。

1) 设定初始化种群规模为 N , 最大迭代次数为



图 2 变异算子

Fig.2 The mutation operator

$g_{\max}$, 交叉概率 P_c 和变异概率 P_m 等算法参数, 并设定当前迭代次数 $g=0$ 。

2) 利用 2.2 节的初始种群生成算子生成规模为 N 的种群 A 。

3) 计算种群 A 中所有个体的适应度值, 选择最优个体作为蜂后 Q_A 。

4) 利用 2.2 节的初始种群生成算子生成外来种群 B (规模为 N), 并选出蜂后 Q_B 。

5) 生成 $(0, 1)$ 之间的随机数 r_A 和 r_B , $r_A + r_B = 1$;

6) 利用 2.4 节的选择算子从种群 A 和 B 中选出 $\frac{N}{2}r_A$ 和 $\frac{N}{2}r_B$ 个雄蜂个体。

7) 利用 2.5 节的交叉算子将 Q_A 和 Q_B 与对应的雄蜂种群进行交叉, 组成新的种群 C (规模为 N)。

8) 利用 2.6 节的变异算子对种群 C 进行变异, 生成新种群 D , 选出对应的蜂后 Q_D 。

9) Q_A , Q_B 和 Q_D 进行竞争, 选择最优个体作为新一代蜂后, 种群 D 为新的种群。

10) $g=g+1$, 比较 g 与 $g_{\max}$ 的大小, 若 $g \geq g_{\max}$ 时, 算法停止, 输出蜂后, 即为最优调度方案, 若 $g < g_{\max}$, 则继续迭代, 算法转入步骤 4。

3 仿真分析

采用车辆调度问题的标准算例来验证文中对传统 GA 算法的改进是否合理。首先, 在“CPU i5-2440, RAM 8G ddr3 1600, MATLAB 2012a”的运行环境下, 分别采用传统 GA 算法^[2]、“改进 1”算法、“改进 2”算法和“改进 3”算法对测试算例进行求解。其中, “改进 1”算法指将传统 GA 算法的初始种群生成算子改进为文中提出的混合生成算子, “改进 2”算法指将“改进 1”算法的交叉算子改进为文中提出的最大保留交叉算子, “改进 3”算法指在“改进 2”算法中引入蜜蜂进化机制, 即文中提出的 HBGA 算法。各算法的参数设置均为: 初始化种群规模 $N=50$, 最大迭代次数 $g_{\max}=120$, 交叉概率 $P_c=0.85$, 变异概率 $P_m=0.05$ 。

对每个算例运行 60 次, 统计各算例求得的最好解 (x_b)、最好解与已知最优解 (x_{bk}) 的误差、平均值 ($\bar{x}$)、平均值与已知最优解的误差和平均运行时间等指标。由于文章篇幅有限, 这里只选择了最好解与已知最优解的误差 (e_1)、平均值与已知最优解的误差 (e_2) 和平均运行时间 (t) 进行展示, 结果见表 1。表 1 中, $e_1=(x_b-x_{bk})/x_{bk} \times 100\%$; $e_2=(\bar{x}-x_{bk})/x_{bk} \times 100\%$; t 为算法进行 60 次求解的平均运行时间, 单位为 s。

表 1 各改进算法仿真结果对比

Tab.1 The simulation results comparison of different improved algorithms

算例	客户数量	x_{bk}	GA算法			“改进1”算法			“改进2”算法			“改进3”算法		
			$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s
C_1	50	524.6	10.3	21.7	15.3	0.0	11.2	10.2	0.0	6.6	9.4	0.0	1.9	7.1
C_2	75	835.3	14.0	25.6	16.7	5.3	11.9	11.7	0.0	8.4	10.8	0.0	3.4	8.8
C_3	100	826.1	24.6	31.4	19.9	3.0	10.8	12.1	1.2	7.7	11.5	0.0	4.9	10.0
C_4	150	1028.4	32.3	32.9	25.6	6.0	15.7	14.5	2.1	9.2	16.3	0.0	4.0	12.4
C_5	199	1291.3	46.8	40.8	24.3	7.1	16.3	20.0	2.9	10.1	19.5	0.6	6.5	14.3
平均值			25.6	30.5	20.4	4.3	13.2	13.7	1.2	8.4	13.5	0.1	4.1	10.5

根据表 1, 在全局寻优能力上, 传统 GA 算法不能求得 C_1 至 C_5 算例集中任一算例的已知最优解 (e_1 不为 0), 且与已知最优解之间的误差较大 (平均值高达 25.6%), 说明在大规模车辆调度问题的求解上, 传统 GA 算法的全局优化能力较差, 非常容易陷入局部最优解。“改进 1”算法、“改进 2”算法和“改进 3”算法能够求得已知最优解的算例个数分别为 1, 2, 4 个, 对应的“ e_1 ”平均值依次为 4.3%, 1.2%, 0.1%, 说明相对于传统 GA 算法, “改进 1”算法、“改进 2”算法和“改进 3”算法的全局寻优能力均得到了大幅度改

善, 且呈现依次增强的趋势。在算法稳定性方面, 传统 GA 算法对应的“ e_2 ”平均值高达 30.5%, 而“改进 1”算法、“改进 2 算法”和“改进 3 算法”对应的平均值均大幅降低, 特别是“改进 3”算法更是降低到了 4.1%, 说明传统 GA 算法的稳定性最差, “改进 1”算法、“改进 2”算法和“改进 3”算法的稳定性均有所提高, 且“改进 3”算法的稳定性最好。在运行速度上, 传统 GA 算法的运行时间较长, “改进 1”算法、“改进 2”算法和“改进 3”算法的运行时间得以大幅减少, 说明传统 GA 算法的运行速度较慢, 而后三者算法的运行速度都有所

提高,且“改进 3”算法速度最快。综上分析,文中改进的初始种群生成算子、交叉算子以及引入的蜂群进化机制都能够提高 GA 算法的求解性能,且同时实施 3 个改进的效果最佳。由此可见,文中对传统 GA 算法的改进是有效的。

为了进一步验证文中提出的 HBGA 算法的求解

性能,分别采用禁忌搜索算法 (TS)^[13]、蚁群算法 (ACO)^[14]和粒子群算法 (PSO)^[15]对 C_1 至 C_5 算例集进行求解。对每个算例运行 60 次,统计相应数据见表 2。其中,各算法的参数设置分别采取相应参考文献的参数。

根据表 2,HBGA 算法、PSO 算法、ACO 算法

表 2 不同算法仿真结果对比
Tab.2 The simulation results comparison of different algorithms

算例	客户数量	x_{bk}	HBGA			TS			ACO			PSO		
			$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s	$e_1/\%$	$e_2/\%$	t/s
C_1	50	524.6	0.0	1.9	7.1	8.2	14.6	6.6	0.0	12.7	13.3	0.0	11.5	8.3
C_2	75	835.3	0.0	3.4	8.8	11.0	16.7	6.9	5.5	13.9	15.4	0.0	12.8	8.7
C_3	100	826.1	0.0	4.9	10.0	11.8	17.7	7.8	6.4	15.6	19.7	5.2	14.3	9.0
C_4	150	1028.4	0.0	4.0	12.4	14.9	17.0	9.1	6.9	14.8	20.6	4.9	14.0	12.9
C_5	199	1291.3	0.6	6.5	14.3	15.7	25.7	10.4	8.7	19.1	25.1	8.5	16.5	15.1
平均值			0.1	4.1	10.5	12.3	18.3	8.2	5.5	15.2	18.8	3.7	13.8	10.8

分别能够求得 4, 2, 1 个算例的最优解, TS 算法则不能求得任一算例的最优解。同时,各算法对应的“ e_1 ”平均值分别为 0.1%, 3.7%, 5.5%, 12.3%, 说明全局寻优能力由强到弱依次为 HBGA 算法、PSO 算法、ACO 算法和 TS 算法。各算法的“ e_2 ”平均值由小到大分别为 HBGA 算法、PSO 算法、ACO 算法和 TS 算法,说明 HBGA 算法的稳定性最好, PSO 算法和 ACO 算法次之, TS 算法最弱。HBGA 算法和 TS 算法的运行时间均小于 ACO 算法和 PSO 算法,且 TS 算法的运行时间最短,说明 TS 算法的运行速度最快, HBGA 算法次之, PSO 算法和 ACO 算法最慢。综上分析, HBGA 算法在全局优化能力和稳定性方面均优于 PSO 算法、ACO 算法和 TS 算法,虽然其运行速度比 TS 算法稍慢,但以此为代价来换取解质量的大幅度提高也是值得的。HBGA 算法的求解性能整体上优于 PSO 算法、ACO 算法和 TS 算法。

4 结语

主要对包装物回收物流中的车辆调度问题进行了研究。针对传统 GA 算法“早熟”、运行速度较慢等不足,对传统 GA 算法的初始种群生成方式和交叉算子进行了改进,并在此基础上引入了混合蜜蜂进化机制,提出了 HBGA 算法。仿真结果表明,文中对传统 GA 算法的一系列改进是合理的,提高了算法的求解性能,且 HBGA 算法的整体求解性能优于其他智能算法。

参考文献:

[1] 刘国秋, 黄小勇, 贾扬蕾. 循环共生经济下包装物逆

向物流回收模式研究[J]. 企业经济, 2014(4): 23—27.
LIU Guo-qiu, HUANG Xiao-yong, JIA Yang-lei. Study on the Recycling Mode of Packaging Reverse Logistics under the Circular Symbiotic Economy[J]. Enterprise Economy, 2014(4): 23—27.
[2] 邢文训, 谢金星. 现代优化计算方法[M]. 北京: 清华大学出版社, 2005.
XING Wen-xun, XIE Jin-xing. Modern Optimization Calculation Method[M]. Beijing: Tsinghua University Press, 2005.
[3] VIDAL T, CRAINIC T G, GENDREAU M, et al. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems[J]. Operations Research, 2012, 60(3): 611—624.
[4] VIDAL T, CRAINIC T G, GENDREAU M, et al. A Hybrid Genetic Algorithm with Adaptive Diversity Management for a Large Class of Vehicle Routing Problems with Time-windows[J]. Computers & Operations Research, 2013, 40(4): 475—489.
[5] NAZIF H, LAI S L. Optimised Crossover Genetic Algorithm for Capacitated Vehicle Routing Problem[J]. Applied Mathematical Modelling, 2012, 36(5): 2110—2117.
[6] CHENG A, YU D. Genetic Algorithm for Vehicle Routing Problem[C]// International Conference on Transportation Engineering, 2013: 2876—2881.
[7] 娄山佐, 史忠科. 有限车辆调度问题的模型和改进遗传算法[J]. 计算机应用研究, 2006, 23(4): 60—62.
LOU Shan-zuo, SHI Zhong-ke. Model and Improved Genetic Algorithm for Vehicle Dispatching Problem with Limited Vehicle Number[J]. Application Research of Computers, 2006, 23(4): 60—62.
[8] 卢雪燕, 周永权. 蜜蜂双种群进化型遗传算法[J]. 计算机工程与设计, 2008, 29(13): 3422—3424.
LU Xue-yan, ZHOU Yong-quan. Double Bee Population Evolutionary Genetic Algorithm[J]. Computer En-

- gineering and Design, 2008, 29(13): 3422—3424.
- [9] 樊贵香. 混合模拟植物生长算法在包装件配送中的应用[J]. 包装工程, 2016, 37(13): 43—49.
FAN Gui-xiang. Application of Hybrid Plant Growth Simulation Algorithm in Packaged Products Distribution[J]. Packaging Engineering, 2016, 37(13): 43—49.
- [10] 蹇洁, 王旭, 葛显龙. 云自适应遗传算法有能力约束的车辆调度优化[J]. 重庆大学学报, 2013, 36(8): 40—45.
JIAN Jie, WANG Xu, GE Xian-long. Research on Capacitated Vehicle Routing Problem with Cloud Adaptive Genetic Algorithm[J]. Journal of Chongqing University, 2013, 36(8): 40—45.
- [11] 李军, 郭耀煌. 物流配送车辆优化调度理论与方法[M]. 北京: 中国物资出版社, 2001.
LI Jun, GUO Yao-huang. Theory and Method of Logistics Distribution Vehicle Scheduling[M]. Beijing: China Goods and Materials Press, 2001.
- [12] 王振锋, 王旭, 葛显龙. 基于遗传算法的不同约束条件车辆调度问题研究[J]. 计算机应用研究, 2010, 27(10): 3673—3675.
WANG Zhen-feng, WANG Xu, GE Xian-long. Study on VSP with Different Constraints Based on Genetic Algorithm[J]. Application Research of Computers, 2010, 27(10): 3673—3675.
- [13] REN C Y. Tabu Search Algorithm for Capacitated Vehicle Routing Problem[J]. Advanced Materials Research, 2013(3): 3060—3063.
- [14] GIANLUCA U. Ant Colony Optimization for Capacitated Vehicle Routing Problem[J]. Journal of Computer Science, 2012, 8(6): 846—852.
- [15] AI T J, KACHITVICHYANUKUL V. Particle Swarm Optimization and Two Solution Representations for Solving the Capacitated Vehicle Routing Problem[J]. Computers & Industrial Engineering, 2009, 56(1): 380—387.

《智能包装与活性包装》特色栏目征稿函

智能包装与活性包装是包装工程技术领域的发展趋势,也是全球包装行业研发和应用的焦点。智能包装新技术与活性包装新材料的应用,能改善包装物条件的体系(通过释放物质、排除或抑制SU),延长包装物使用寿命;提高卫生安全性;改善气味和口感特性的同时保证品质不变。利用新型的包装材料、结构与形式对商品的质量和流通安全性进行积极干预与保障,通过信息收集、管理、控制与处理技术完成对运输包装系统的优化管理等。

鉴于此,本刊拟围绕“智能包装与活性包装”这一主线,作系列专项报道。本刊编辑部特邀请相关专家为该栏目撰写稿件,以期进一步提升本刊的学术质量和影响力。稿件以研究论文为主,也可为综述性研究,请通过网站投稿,编辑部将快速处理并优先发表。

编辑部电话: 023-68792294 网址: www.packjour.com

《包装工程》编辑部