

doi: 10.3969/j.issn.1007-7375.2021.05.009

带时间窗的同时取送货车辆路径问题求解算法

闫 军¹, 常 乐², 王璐璐², 赵 彤³

(1. 兰州交通大学 甘肃省物流与信息技术研究院, 甘肃 兰州 730070; 2. 兰州交通大学 机电技术研究所, 甘肃 兰州 730070; 3. 呼和浩特铁路局集团公司包头货运中心, 内蒙古 包头 014000)

摘要: 为了整合物流配送过程的退货与送货服务, 依据实际情况建立带时间窗的同时取送货车辆路径规划模型, 设计一种基于K-means聚类处理的Q-Leaning自启发式蚁群算法解决此类问题。根据配送服务的特性, 在基本的K-means算法上作相应的改进, 同时提高蚁群算法的局部搜索能力, 完成两算法的合理衔接。选用相关文献数据和标准算例进行实验, 验证所提算法具有较好的性能, 可以解决所描述的此类问题。

关键词: 车辆路径问题; 取送货问题; 时间窗; 蚁群算法

中图分类号: TP301.6

文献标志码: A

文章编号: 1007-7375(2021)05-0072-05

A Solution Algorithm for the Problem of Taking Delivery Vehicle Path at the Same Time with Time Window

YAN Jun¹, CHANG Le², WANG Lulu², ZHAO Tong³

(1. Gansu Institute of Logistics and Information Technology;

2. Mechatronics T&R-Institute, Lanzhou Jiaotong University, Lanzhou 730070, China;

3. Baotou Freight Center, Hohhot Railway Bureau Group Company, Baotou 014000, China)

Abstract: In order to integrate the return and delivery service in the process of logistics and distribution, a vehicle routing model with time window is established according to the actual situation, and a Q-learning self-heuristic ant colony algorithm based on K-means clustering processing is designed to solve such problems. According to the characteristics of distribution service, by improving the basic k-means algorithm, the local search ability of the ant colony algorithm is improved, and the reasonable connection of the two algorithms completed. Experiments are carried out according to the relevant literature data and standard examples to verify that the proposed algorithm has good performance and can solve the described problems.

Key words: vehicle routing problem; pickup and delivery problem; time windows; ant colony algorithm

车辆路径问题 (vehicle routing problem, VRP) 是物流生产活动中的核心环节, 也是组合优化和运筹学领域的研究热点^[1]。在实际的物流服务中, 时间窗和取送货服务是需要满足的现实要求, 所以带时间窗的同时取送货的车辆路径问题 (pickup and delivery problem with time windows, PDPTW) 得到越来越多的关注^[2]。

起初解决多车辆的PDPTW问题主要是依靠精确算法, 从1991年Dumas等^[3]的列生成算法到Furtado等^[4]设计基于特殊有效不等式的分支切线方法。精确算法虽然无法求解大规模客户的实际案例, 但丰

富了PDPTW的数学模型, 为目前的启发式算法求解问题提供基础。本文也以此为基础建立优化模型。Nanry等^[5]最早使用响应式禁忌启发搜索算法对PDPTW进行求解, 并进行25、50和100个客户规模的数据实验。Li等^[6]在禁忌搜索算法的基础上混合模拟退火算法进行研究, 而且将其研究的数据集进行公布作为PDPTW的研究基础。吴璟莉^[7]使用遗传算法对拥有10个客户规模的多车、多货物仓库进行PDPTW问题的求解。为减少运行的车辆数目, Nagata等^[8]运用引导式弹射搜索算法来对问题进行求解, 实验证明可以有效减少车辆数。Zou等^[9]使用混合粒

收稿日期: 2020-04-08

基金项目: 甘肃省自然科学基金资助项目(148RJZA049); 甘肃省教育厅战略研究资助项目(2018F-08)

作者简介: 闫军(1971—), 男, 甘肃省人, 高级工程师, 硕士, 主要研究方向为协同物流及动态规划。

子群算法对PDPTW进行求解, 目的是为优化多目标的PDPTW问题。蚁群算法被用来进行PDPTW问题的求解, 首次出现于2017年。Tchoupo等^[10]强化蚁群算法局部优化能力后, 与以往实验结果进行对比, 发现其方法在98.2%的数据实验上取得相同或更好的结果, 证明了蚁群算法在此类问题上的有效性。

本文在考虑客户规定时间窗和退送货需求的条件下, 研究企业物流配送的最小化成本, 建立车辆路径问题优化数学模型。首先运用K-means聚类算法将具有相同属性的客户划分成不同服务区域, 综合考虑时间窗、与仓库的相对位置及与其他客户的距离因素; 然后在每个服务区域内运用蚁群算法对单个车辆进行PDPTW问题求解, 为提高蚂蚁的搜索能力, 建立Q-leaning机制来改进蚁群算法内蚂蚁的学习能力。

1 问题描述及模型构建

1.1 问题描述与符号说明

PDPTW定义在图 $G(N, A)$ 上, 节点集为 $N = \{0, 1, \dots, 2n+1\}$, 其中, 0和 $2n+1$ 分别表示起点和终点, 它们可以表示同一位置。子集 $P = \{1, \dots, n\} \subset N$ 和 $D = \{n+1, \dots, 2n\} \subset N$ 将一节点划分为取货点和送货点两个节点集。节点 $i \in N$ 的需求量为 q_i , m 辆容量为Cap的运输车, 从仓库 $N=0$ 出发, 按一定顺序通过所有节点, 然后返回仓库, 每一辆车只能服务一条线路, 当到达节点 i 时车的负载和时间分别为 Q_i 和 B_i , 其中, B_i 要满足时间窗 $[e_i, l_i]$ 的要求。边集为 $A = \{(i, j) : i, j \in N, i \neq j\}$, 当且仅当车辆从节点 i 到节点 j 时, 令 x_{ij} 为等于1的二进制变量, C_{ij} 为运输的成本, t_{ij} 为 i 节点到 j 节点的车辆行驶时间^[11]。

本文研究的PDPTW可描述为在不超过客户规定的时间窗和运输车辆的载货量的约束下, 找到不超过 m 条路径使运输成本最小, 并且尽可能地平衡各路径的总长度, 通过这样的路径规划为客户提供同时取送货的服务, 并且车辆最终返回仓库。

1.2 模型建立

基于以上问题描述与符号, 建立以下PDPTW两指标数学模型。

$$\min \sum_{i \in N} \sum_{j \in N} C_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in N} x_{ij} = 1; \quad (2)$$

$$\sum_{j \in N} x_{ij} = 1; \quad (3)$$

$$B_j \geq B_i + t_{ij} - M(1 - x_{ij}); \quad (4)$$

$$Q_j \geq Q_i + q_j - M(1 - x_{ij}); \quad (5)$$

$$e_i \leq B_i \leq l_i; \quad (6)$$

$$\max\{0, q_i\} \leq Q_i \leq \min\{\text{Cap}, \text{Cap} + q_i\}; \quad (7)$$

$$B_{n+i} \geq B_i + t_{i, n+i}; \quad (8)$$

$$v_{n+i} = v_i; \quad (9)$$

$$v_j \geq jx_{0j}; \quad (10)$$

$$v_j \leq jx_{0j} - n(x_{0j} - 1); \quad (11)$$

$$v_j \geq v_i + n(x_{ij} - 1); \quad (12)$$

$$v_j \leq v_i + n(1 - x_{ij}); \quad (13)$$

$$x_{ij} \in \{0, 1\}. \quad (14)$$

目标函数(1)使总的路径成本最小化。式(2)和(3)保证每个节点仅被访问一次, 时间变量和负载变量的一致性分别由式(4)和(5)确保, 其中, 常数 M 被定义为足够大的数字, 时间窗和车辆容量分别由式(6)和(7)保证。此外, 式(4)和(6)消除子路径。式(8)满足取货点和送货点的优先级关系。约束(9)~(13)进行路径信息的索引, 保证车辆与路径的配对关系。其中, v 表示储存路径信息, 当节点 i 属于第一条路径时 $v_i = 1$ 。最后, 约束条件(14)使两指标变量变得完整。

2 聚类处理的蚁群算法设计

算法分为两步: 1) 对配送点进行区域划分, 根据所划分区域情况明确每一区域具体的取送货量以及整个配送活动的运行车辆数目; 2) 根据第1)步的数据进行具体的单车辆线路规划。具体流程如下。

2.1 基于K-means算法的区域分割聚类处理的蚁群算法设计

本文对于PDPTW问题进行客户节点聚类, 与传统的K-means聚类方法不同的是要考虑配送车辆的容量限制、仓库与客户点的位置关系(根据仓库的位置进行客户点的聚类)的影响。因此本文为适应PDPTW的实际需求, 在K-means划分区域的基础上进行修正。具体流程如下。

步骤1 本文借鉴文献[12]的类间类内划分(between-within proportion, BWP)指标确定最佳聚类数来评价聚类的有效性。假设当前节点被聚集成

c 类所求得的BWP值越大表示聚类效果越好, 式(15)中, $y_s^{(a)}$ 表示第 a 类的第 s 个节点, $y_r^{(k)}$ 表示第 k 类的第 r 个节点, $y_b^{(a)}$ 表示第 a 类的第 b 个节点, 其中, 第 k 类拥有 n_k 个节点以及第 a 类拥有 n_a 个节点。

$$\text{BWP}(a, b) = \frac{\min_{1 \leq k \leq c, k \neq a} \left(\frac{1}{n_k} \sum_{r=1}^{n_k} \|y_r^{(k)} - y_b^{(a)}\|^2 \right) - \frac{1}{n_a - 1} \sum_{s=1, s \neq b}^{n_a} \|y_s^{(a)} - y_b^{(a)}\|^2}{\min_{1 \leq k \leq c, k \neq a} \left(\frac{1}{n_k} \sum_{r=1}^{n_k} \|y_r^{(k)} - y_b^{(a)}\|^2 \right) + \frac{1}{n_a - 1} \sum_{s=1, s \neq b}^{n_a} \|y_s^{(a)} - y_b^{(a)}\|^2} \quad (15)$$

根据拥有的运输汽车数量设定聚类数范围, 计算平均BWP也就是avgBWP值来判断整体的聚类有效性。当平均BWP值最大时, 对应的聚类数便是最佳聚类数 k_{opt} 。

$$\text{avg BWP}(k) = \frac{1}{n} \sum_{a=1}^k \sum_{b=1}^{n_a} \text{BWP}(a, b); \quad (16)$$

$$k_{\text{opt}} = \arg \max \{ \text{avgBWP}(k) \}. \quad (17)$$

步骤2 根据步骤1进行聚类, 将所有节点划分到不同区域 $d = \{d_1, d_2, \dots, d_n\}$ 。令 $Q = \{Q_1, Q_2, \dots, Q_n\}$ 为 D 区域的负载并计算方差 d_Q , 令 $M_d = \{m_{d_1}, m_{d_2}, \dots, m_{d_n}\}$ 为 d 的聚类中心, d_i ($Q_i = \max \{Q_1, Q_2, \dots, Q_n\}$ 对应区域)中 j 点到区域聚类中心 m_k ($k = 1, 2, \dots, n$)的距离为 d_{jk} 。计算 $d^{jk} = \min |d_{jk} - d_{ji}|$, $i \neq k$, 将 d_i 中的 j 点划分到 d_k 中并更新 Q 和 d_Q , 当 d_Q 不再减小, 则停止步骤2。

2.2 基于Q-learning的蚁群算法

初始的多车辆复杂VRP经过聚类处理后变成简单的单车辆VRP, 故求解的计算量减少, 所以此阶段需要算法有很强的局部搜索能力。蚁群算法中蚂蚁的转移概率引入强化学习机制, 使蚁群算法的收敛能力得到提高^[13]。

算法结构如图1。其中, \mathbf{AQ} 表示蚂蚁的学习经验矩阵, 初始值 \mathbf{AQ}_0 设为零矩阵。在循环结构中, 首先在每个节点随机放置蚂蚁, 蚂蚁根据式(18)选择 $J_k(r)$ 中的节点进行移动^[13], $J_k(r)$ 表示蚂蚁尚未到达的节点集合。

$$s = \begin{cases} \max_{u \in J_k(r)} \{ [\mathbf{AQ}(r, u)]^\delta \cdot [\text{HE}(r, u)]^\beta \}, & \text{若 } f \leq f_0; \\ S, & \text{否则。} \end{cases} \quad (18)$$

$$\text{Pk}(r, s) = \begin{cases} \frac{[\mathbf{AQ}(r, s)]^\delta \cdot [\text{HE}(r, s)]^\beta}{\sum_{z \in J_k(r)} [\mathbf{AQ}(r, z)]^\delta \cdot [\text{HE}(r, z)]^\beta}, & \text{若 } s \in J_k(r); \\ 0, & \text{否则。} \end{cases} \quad (19)$$

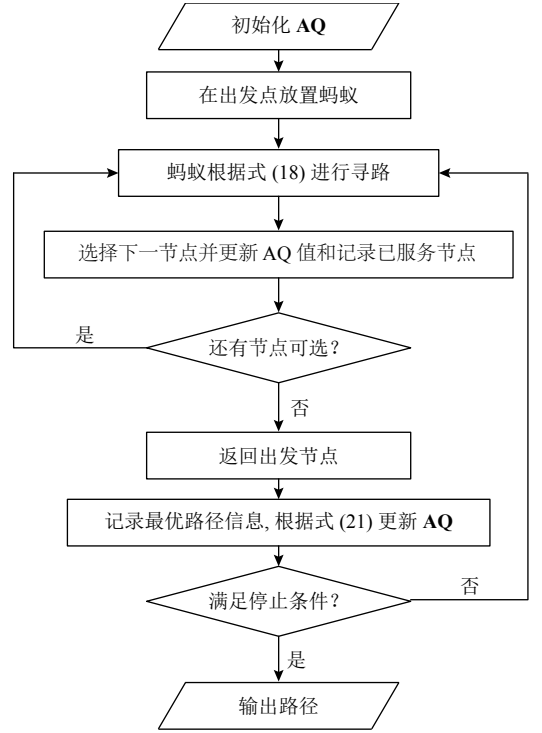


图 1 基于Q-learning的蚁群算法结构

Figure 1 Ant colony algorithm structure based on Q-learning

式中, $\text{HE}(r, u)$ 是一种启发式函数, 在本文中用来评价节点 r 移动到 s 的优劣。 δ 、 β 是权衡学习经验矩阵和启发式函数相对重要性的参数。 f 是属于 $[0, 1]$ 的随机值, 而 f_0 是大于0小于1的一个参数, 当 $f \leq f_0$ 时选择下一个节点 s , 否则随机产生属于 $[0, 1]$ 的数 s 。通过轮盘赌的方式与式(19)所计算的概率值 $\text{Pk}(r, s)$ 进行比较选择下一节点。路径上的信息素依靠学习经验矩阵的循环迭代进行更新, \mathbf{AQ} 根据式(20)进行更新。

$$\mathbf{AQ}(r, s) \leftarrow (1 - \alpha) \cdot \mathbf{AQ}(r, s) + \alpha \cdot \left\{ \Delta \mathbf{AQ}(r, s) + \gamma \cdot \max_{z \in J(s)} \mathbf{AQ}(s, z) \right\}. \quad (20)$$

$$\Delta \mathbf{AQ}(r, s) = \begin{cases} \frac{W}{L_{\text{Best}}}, & \text{若 } (r, s) \in \text{当前最优路径;} \\ 0, & \text{否则。} \end{cases} \quad (21)$$

式中, z 节点是蚂蚁到达 s 节点后所面临选择的下一节点; α 是常数; $\Delta \mathbf{AQ}(r, s)$ 是 $\mathbf{AQ}(r, s)$ 的信息素增强机制; W 是固定参数; L_{Best} 是当前循环最优蚂蚁走的最优路径的长度。经算法复杂度分析理论计算, 此算法的复杂度为 $O(\text{RN} \cdot K \cdot (n/K)^2)$, 其中, RN 为迭代次数, K 为节点集分类数。

3 算例实验与结果分析

为检验设计算法的有效性, 本文设计两个实验。实验1为验证加入聚类分析的作用, 将普通多车量VRP问题作为模型, 运用本文方法和文献中的其他方法分别求解, 得出结果进行对比。实验2为验证本文提出的模型和算法在大规模车辆路径问题中的适用性, 采取300个节点规模PDPTW问题的算例进行实验分析。

本文算法在Matlab 2018a上进行实验, 并在一台搭载1.9GHz的AMD四核A10处理器和8GB内存的计算机上实现, 蚁群数设为100, 进行50次迭代。算法涉及参数根据控制变量法实验得出, 每次只变动某一参数进行实验, 确定合适的参数值, 具体参数设置如表1所示。

实验1运用Wang等^[14]提出的关于VRPSPDTW的标准算例, 并且将本文提出的KQ-SA (结合K-means的Q-learning蚁群算法)与Wang等^[14]提出的并行模拟退火算法 (parallel-simulated annealing, p-SA)和王超

表 1 参数设置

Table 1 parameter settings

算法参数	δ	β	f_0	α	γ	W
取值	1	2	0.9	0.1	0.2/0.6	10

等^[15]的离散布谷鸟算法 (discrete cuckoo search, DCS) 进行对比实验。本文应用环境为大规模车辆路径问题, 所以采用的算例节点数为25、50、100的各3组算例进行实验。节点数为100的算例路径规划如图2和图3所示, 图中五角星为仓库位置。

每个算例进行10次, 取最优结果所需车辆数目 (number vehicle, NV)和行驶距离(travel distanced, TD)记录, 如表2所示。其中行驶距离单位为km。由表2可知, KQ-AG可以的中小规模中的算例中可以得到与当前国际最优解相近似的解, 并且在100个节点的算例中能够得出优于文献中的计算结果。

实验2 收集文献中PDPTW问题算例实验结果最优的数据, 并将其作为对照组与本文的所建模型和算法进行对比, 表3展示了15组300客户规模的实验对比结果。

从表3中数据可以看出, 本文模型和求解算法在计算时长上有很突出的进步, 这主要是使用了聚类方法简化计算的复杂度。表中标粗数据为总行驶距离最优解, 本文提出的方法在15组实验中有6组

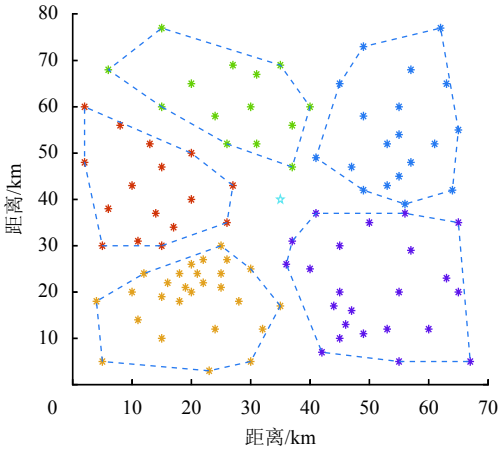


图 2 节点聚类结果

Figure 2 Clustering results of nodes

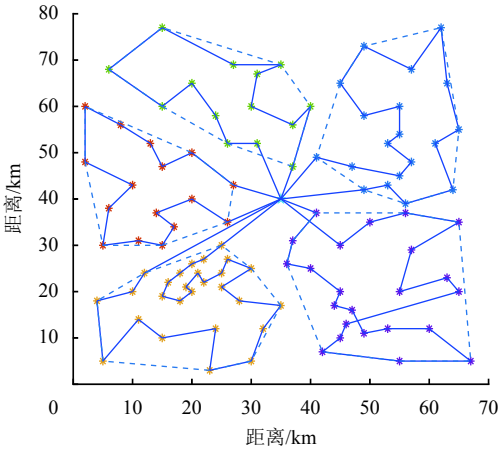


图 3 节点路线规划结果

Figure 3 Node route planning results

表 2 实验1算法结果对比

Table 2 Comparison of algorithm results of Experiment 1

算例/客户数	P-SA		DCS		KQ-AG	
	NV/辆	TD/km	NV/辆	TD/km	NV/辆	TD/km
Rcdp2501/25	5	552.21	5	551.05	5	552.15
Rcdp2504/25	4	473.46	4	473.46	4	473.46
Rcdp2507/25	5	540.87	5	540.87	5	540.87
Rcdp5001/50	9	994.70	9	994.18	9	995.18
Rcdp5004/50	6	725.90	6	725.59	6	727.59
Rcdp5007/50	7	810.04	7	809.72	7	809.72
Rdp101/100	19	1 660.98	19	1 658.65	19	1 653.53
Rdp102/100	17	1 491.75	17	1 490.13	17	1 488.04
Rdp103/100	14	1 226.77	14	1 228.48	14	1 216.16

低于文献的最优解, 有5组优于文献的解, 其余得出相同解。本文方法在减少行驶距离时的优势不明显, 低于文献最优解的实验中发现, 仓库在偏离配送点较远的位置, 聚类的优势效果没有体现出来。

表 3 实验2算法结果对比

Table 3 Comparison of algorithm results of Experiment 2

算例	文献结果		本文最优结果		距离降低 比率/%	计算时长 降低值/s
	总行驶 距离/km	计算 时长/s	总行驶 距离/km	计算 时长/s		
Ir1_6-1	31 578.5	1 020.2	31 146.2	403.6	1.37	616.6
Ir1_6-2	26 572.8	1 125.8	27 432.5	510.1	-3.24	615.7
Ir1_6-3	23 699.6	2 189.2	24 148.3	735.8	-1.89	1 453.4
Ir1_6-4	16 713.2	2 923.3	17 867.4	635.2	-6.91	2 288.1
Ir1_6-5	29 357.2	949.0	24 649.2	501.2	16.04	447.8
Ir1_6-6	26 473.3	1 508.3	23 356.1	618.4	11.77	889.9
Ir1_6-7	20 578.1	1 754.4	19 877.8	598.4	3.40	1 156
Ir1_6-8	17 290.2	1 405.2	17 290.2	720.6	0.00	684.6
Ir1_6-9	27 319.1	1 109.2	27 408.2	466.3	-0.33	642.9
Ir1_6-10	22 814.0	1 689.0	23 128.4	821.6	-1.38	867.4
Irc1_6-1	23 911.5	750.5	21 332.6	321.3	10.79	429.2
Irc1_6-2	20 518.6	875.5	18 745.4	630.2	8.64	245.3
Irc1_6-3	16 722.2	1 151.8	16 722.2	537.4	0.00	614.4
Irc1_6-4	13 314.6	2 526.3	13 475.9	433.9	-1.21	2 092.4
Irc1_6-5	20 375.0	2 684.8	20 375.0	807.9	0.00	1 876.9
平均值	22 482.5	1 577.5	21 797.0	582.0	3.05	994.7

综合15轮实验数据，本问题提出的模型和解决算法可以求解出较优的结果，并且在缩短计算时间方面有较好的表现。

4 结语

本文建立两指标的PDPTW问题模型，较三指标体系更简单计算速度更快。为解决大规模算例计算量大的问题，本文对客户节点首先进行聚类处理，然后进行具体路径求解，并且在使用K-means算法时，引入BWP指标来提升聚类的效果。为了提高蚁群算法的搜索能力，使用Q-leaning自适应蚁群算法对单独车辆的路径进行规划。本文提出的方法具有较强的适用性，在中小规模的案例中均能有较好表现，并且在计算大规模的算例时，缩短计算时间有比较大的突破。本文在缩短路径距离方面能力一般，接下来的研究主要是在聚类方面，如何能够解决偏离客户节点的仓库位置对路径规划的影响。

参考文献:

[1] 江海, 陈峰. 面向快递同城运输的车辆路径问题研究[J]. *工业工程*, 2019, 22(4): 58-63.
JIANG Hai, CHEN Feng. A study of vehicle routing problem for urban transport of express[J]. *Industrial Engineering Journal*, 2019, 22(4): 58-63.

[2] WANG L Y, SONG Y B. Multiple charging station location routing problem with time window of electric vehicle[J]. *Journal of Engineering Science and Technology Review*, 2015, 8(5):

190-201.

[3] DUMAS Y, DESROSIERS J, SOUMIS F. The pickup and delivery problem with time windows[J]. *European Journal of Operational Research*, 1991, 54(1): 7-22.

[4] FURTADO M G S, MUNARI P, MORABITO R. Pickup and delivery problem with time windows: a new compact two-index formulation[J]. *Operations Research Letters*, 2017, 45(4): 334-341.

[5] NANRY W P, BARNES J W. Solving the pickup and delivery problem with time windows using reactive tabu search[J]. *Transportation Research Part B: Methodological*, 2000, 34(2): 107-121.

[6] LI H, LIM A. A met heuristic for the pickup and delivery problem with time windows[J]. *International Journal on Artificial Intelligence Tools*, 2003, 12(2): 173-186.

[7] 吴璟莉. 求解复杂有时间窗装卸货问题的遗传算法[J]. *计算机应用*, 2006, 26(6): 1459-1462.
WU Jingli. Genetic algorithm for solving the complex pickup and delivery problem with time windows[J]. *Computer Applications*, 2006, 26(6): 1459-1462.

[8] NAGATA Y, KOBAYASHI S. Guided ejection search for the pickup and delivery problem with time windows[C]// *European Conference on Evolutionary Computation in Combinatorial Optimization*. Berlin, Heidelberg: Springer, 2010: 202-213.

[9] ZOU S, LI J, LI X. A hybrid particle swarm optimization algorithm for multi-objective pickup and delivery problem with time windows[J]. *Journal of Computers*, 2013, 8(10): 2583-2590.

[10] TCHOUP M N, YALAOUI A, AMODEO L, et al. Ant colony optimization algorithm for pickup and delivery problem with time windows[C/OL]// *International Conference on Optimization and Decision Science*. Sorrento: Springer, 2017 [2017-11-05]. https://link.springer.com/chapter/10.1007/978-3-319-67308-0_19.

[11] MARCO D, LUCA M G. A study of some properties of ANT-Q [C]// *The Proceedings of PPSN IV—Fourth International Conference on Parallel Problem Solving from Nature*. Berlin: Springer-Verlag, 1996: 656-665.

[12] 周世兵, 徐振源, 唐旭清. K-means算法最佳聚类数确定方法[J]. *计算机应用*, 2010, 30(8): 1995-1998.
ZHOU Shibing, XU Zhenyuan, TANG Xuqing. Method for determining optimal number of clusters in K-means clustering algorithm[J]. *计算机应用*, 2010, 30(8): 1995-1998.

[13] 孙戡, 曹雷, 陈希亮, 等. 多智能体深度强化学习研究综述[J]. *计算机工程与应用*, 2020, 56(5): 13-24.
SUN Yu, CAO Lei, CHEN Xiliang, et al. Overview of multi-agent deep reinforcement learning[J]. *Computer Engineering and Applications*, 2020, 56(5): 13-24.

[14] WANG H F, CHEN Y Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window[J]. *Computers & Industrial Engineering*, 2012, 62(1): 84-95.

[15] 王超, 刘超, 穆东, 等. 基于离散布谷鸟算法求解带时间窗和同时取送货的车辆路径问题[J]. *计算机集成制造系统*, 2018, 24(3): 570-582.
WANG Chao, LIU Chao, MU Dong, et al. VRPSPTW problem solving by discrete cuckoo search[J]. *Computer Integrated Manufacturing Systems*, 2018, 24(3): 570-582.