

唐传茵¹ 章明理¹ 李静红^{2,3} 苑莹⁴ 卫美荣¹

基于改进蚁群算法的外卖配送路径规划研究

摘要

从外卖配送员角度出发提出一种改进蚁群算法(Improved Ant Colony Optimization, IACO),在此基础上进行外卖配送路径规划研究.首先通过蚁群算法(Ant Colony Optimization, ACO)求解得到初始规划路径,然后通过大规模邻域搜索算法(Large Neighborhood Search, LNS)优化初始规划路径,通过将ACO和LNS算法结合,提高求解质量.为了验证方法的有效性,对外卖配送过程进行仿真,并且选用不同订单数量场景进行对照分析.根据最优配送方案路线图和目标罚函数的最优值可以得出,IACO算法是有效的,且可以提高外卖配送员外卖配送的效率.IACO算法不但能够提升配送的智能化水平,还从外卖配送员的角度提出一种更为人性化的配送方法,支持网络互联外卖平台派送系统的可持续发展.

关键词

改进蚁群算法;大规模邻域搜索算法;外卖配送;配送方案

中图分类号 F252;TP18

文献标志码 A

收稿日期 2023-03-11

资助项目 中央高校基本科研业务费战略新兴资助项目(N2103028)

作者简介

唐传茵,博士,副教授,研究方向为系统动力学.monsigor@aliyun.com

0 引言

随着科技的快速发展,人们的生活变得更加便利.外卖行业蓬勃发展,使得人们能够在互联网上购买自己想要和需要的物品.外卖行业在日常生活中显得越来越重要,同时对于外卖配送服务的要求也越来越高.外卖配送服务的流程是客户在第三方平台购物下单,商家接受订单后,平台的外卖配送员根据订单信息进行外卖配送服务.在正常情况下,一位外卖配送员在同一时间段中所需要配送的外卖可能会有多个,在外卖配送过程中会因为配送路线安排不合理导致订单无法及时送达等问题.在外卖平台系统的设置中,配送超时是不被允许的,一旦出现配送超时的问题,客户对外卖配送员的评价可能为差评,将造成外卖配送员收入降低、奖金被扣等负面影响^[1-3].

外卖配送问题,可以简化为带有时间窗和容量限制的车辆路径问题^[4].针对车辆路径问题(VRP),常见的求解方法有遗传算法^[5]、蚁群算法^[6]、禁忌搜索^[7]等智能算法.遗传算法是通过模拟生物进化适者生存的过程,在解决组合优化问题时使用选择、交叉、变异操作找出全局最优解.遗传算法的操作较简单,在不同的邻域可以表现出很好的兼容性,但是在解决具有针对性,并且要求一定精度的问题时,算法所得到的解在局部表现得不令人满意^[8].禁忌搜索算法需要对每个可能的解进行检查,因此搜索效率较低.蚁群算法(Ant Colony Optimization, ACO)的基本思想是模拟蚂蚁寻找食物的过程,当得到一条有效路径时会释放一定的信息素标记该路径,对其他蚂蚁起到引导作用,从而加速整个搜索过程^[9].蚁群算法可以处理大规模的解空间,并且自适应地调整搜索范围,因此蚁群算法对多种组合优化和运输调度问题具有较好效果.近年来,国内外学者们针对基本蚁群算法易陷入局部最优解和收敛速度慢的缺点,进行了大量的改进研究.Zhang等^[10]将邻近启发式算法与蚁群算法结合,提高了算法的性能和求解的质量;王晓东等^[11]引入节约矩阵 U 引导蚂蚁搜索,通过不同搜索时段采用不同的信息素挥发因子,平衡优化蚁群算法中权重的分配,提高算法的性能;Donati等^[12]使用多蚁群系统(MACS-VRPTW)算法求解VRPTW问题,通过设计多个蚁群系统对多目标进行优化,提高求解速度和解的质量;Kalayci等^[13]基于蚁群系统(ACS)和可变邻域搜索(VNS)的混合算法,防止算法求解陷入局部最优解,提高算法的性能和求解的质量;李琳等^[14]将蚁群系统(ACS)与最

1 东北大学 机械工程与自动化学院,沈阳,110819

2 上海电机学院 机械学院,上海,201306

3 浙江理工大学 机械工程学院,杭州,310018

4 东北大学 工商管理学院,沈阳,110167

大最小蚂蚁系统(MMAS)结合,并且设计在算法不同阶段采用不同的信息素蒸发策略防止陷入局部最优,用2-opt和2-opt*方法对每次迭代所得最优解进行局部优化,提高算法性能.针对传统智能算法易出现局部最优解情况,Blum等^[15]提出进化算法(EA)与大规模邻域搜索算法(Large Neighborhood Search, LNS)结合可以提高进化算法生成解的质量,从而提高算法的性能;徐倩等^[16]提出一种自适应大规模邻域搜索算法求解外卖配送车辆调度问题,提高了外卖配送效率.

为了解决外卖配送路径规划问题,本文提出一种改进蚁群算法(Improved Ant Colony Optimization, IACO),综合考虑外卖配送员在进行外卖配送过程中的实际情况,对外卖配送路径进行规划.外卖配送过程中不仅有时间窗要求,对配送数量同样有要求,并且要求配送总路径尽可能短.因此,本文所研究的外卖配送过程可以简化为有时间窗约束和车载容量约束的车辆路径问题(CVRPTW).引入外卖配送过程中以费用最小为目标的罚函数,建立外卖配送的数学模型,结合蚁群算法和大规模邻域搜索算法(LNS)对所建立的数学模型进行分析求解,并通过仿真实验对模型和所提出的改进蚁群算法进行验证分析.

本文的贡献为:1)从外卖配送员的角度出发,在外卖配送过程中考虑到时间窗约束、容量约束和最短总路径,所设计的算法更加贴近实际;2)根据外卖配送员配送过程的实际情况设计合理的外卖配送数学模型,以配送过程中的费用最小为目标设计罚函数,充分考虑到对于配送员的最优方案的设计;3)将蚁群算法和大规模邻域搜索算法结合,提高单一算法的求解质量,并且合理运用在本文所设计的外卖配送方案模型中,通过仿真结果验证了所提出算法的有效性,从而提高外卖配送效率.

1 外卖配送问题描述

本文研究的问题是针对一个外卖配送中心向多位顾客配送外卖或在一个紧凑的商业区向多位顾客进行外卖配送的场景.因此,该外卖配送路径规划问题可以描述为在满足配送员车载容量约束和送餐时间窗约束的前提下,对配送员的外卖配送路径进行合理规划,以提高配送效率并满足约束条件.整体规划出的路径需要使车辆的使用数目、配送员行驶总距离和违反约束条件的惩罚函数的各项成本之和达

到最小.对本文所研究的问题可以简化为带有时间窗约束和车载容量约束的车辆路径问题(CVRPTW)^[17].因此,对研究的CVRPTW问题提出如下假设:

- 1) 外卖配送员配送起点和终点都为配送中心;
- 2) 所有外卖车辆的车载容量相同,配送容量不能超过车载容量;
- 3) 每位顾客的外卖只能由一位外卖配送员进行配送;
- 4) 每位顾客的外卖质量已知,并且提供送餐时间窗.

数学模型描述如下:

设 K 为外卖配送中心配送员总数的集合; N 为需要配送服务的 n 个顾客点的集合, $N = \{1, 2, 3, \dots, n\}$; $V = \{0\} \cup N$.

$$\min \sum_{k \in K} \sum_{i \in V_k} \sum_{j \in V_k \setminus \{i\}} C_{ij} x_{ij k}, \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in V_k \setminus \{i\}} x_{ij k} = y_{i k}, \quad \sum_{i \in V_k \setminus \{j\}} x_{ij k} = y_{i k}, \\ \forall i \in V, \forall j \in V, \forall k \in K, \quad (2)$$

$$\sum_{j \in V_k} q_i y_{i k} \leq Q, \quad \forall k \in K, \quad (3)$$

$$\sum_{j \in N} x_{0j k} = \sum_{i \in N} x_{i0 k} = 1, \quad \forall k \in K, \quad (4)$$

$$\sum_{k \in K} y_{i k} = 1, \quad \forall i \in N, \quad (5)$$

$$\sum_{i, j \in V_k} x_{ij k} \leq |V_k| - 1, \quad \forall k \in K, \quad (6)$$

$$T_{E, i} \leq a_i \leq T_{L, i}, \quad \forall i \in N. \quad (7)$$

其中:0为配送中心; V_k 为车辆访问顾客的集合; i, j 表示顾客编号; $x_{ij k}$ 表示配送员 k 对顾客 i 的外卖配送结束后是否需要配送顾客 j ; $y_{i k}$ 表示配送员 k 是否完成了对顾客 i 的外卖配送; Q 表示配送员配送外卖容量的约束; q_i 表示顾客 i 的外卖需求量; C_{ij} 为配送员从 i 点驶向 j 点的运输成本; $T_{E, i}$ 为顾客 i 允许的最早外卖配送到达时间; $T_{L, i}$ 为顾客 i 允许的最迟外卖配送到达时间; a_i 为配送员到达顾客 i 的时间. $x_{i k}$, $y_{i k}$ 为决策变量,当配送员 k 对顾客 i 的外卖配送完成后对顾客 j 进行外卖配送, $x_{ij k} = 1$,否则 $x_{ij k} = 0$;当顾客 i 由配送员 k 进行外卖配送时, $y_{i k} = 1$,否则 $y_{i k} = 0$.

式(1)表示配送员外卖配送过程中费用最小的目标罚函数;式(2)表示配送员 k 在配送完顾客 i 后只配送顾客 j ,配送员 k 在配送顾客 j 之前只配送顾客 i ;式(3)表示每个配送员配送外卖的总容量不能超过车载总容量;式(4)表示每个配送员从配送中心出发,送完最后一个顾客后返回配送中心;式(5)表示

每位顾客只能有一位配送员进行外卖配送;式(6)表示避免配送过程中产生子回路;式(7)表示配送员给每位顾客配送外卖到达时间应该在规定的时间内。

2 改进蚁群算法设计

2.1 蚁群算法原理

蚁群算法是一种源于自然界的仿生进化算法,主要思想是蚂蚁在觅食的过程中会在经过的路径留下信息素从而引导整个蚁群的搜索过程,人类通过计算机模拟自然界中蚁群觅食的行为进行蚁群算法的开发.每只蚂蚁都会进行觅食行为,并且在觅食的路径上留下一一种特有物质——信息素,其他蚂蚁在经过该路径时,可以感应到该信息素,并以此引导其他蚂蚁觅食.一般情况下,蚂蚁离开巢穴,会选择一条通往目的地的路径,在路径的每一个交叉点上都会将前一只蚂蚁所释放的信息素作为标记来选择接下来前进的路径,以此进行多次迭代可以得到最优路径。

2.2 配送路径生成

配送员 k 从顾客 i 所在地方出发到顾客 j 所在地方那一刻时间为 a_j , 根据式(7) 要求 a_j 的范围, 如果 a_j 小于顾客 j 时间窗的左边界, 会导致配送员需等待顾客所设定的外卖最早到达时间, 影响外卖的配送效率, 配送成本增加; 如果 a_j 大于顾客 j 时间窗的右边界, 会导致配送员配送外卖超时, 极易引起顾客差评等不满评价, 导致配送员利益受损. 因此, 在配送员配送外卖的到达时间必须在规定的时间内, 所规划出的路径对于配送员来说比较合理, 需要在状态转移过程中加入等待时间的影响因素. 当 $T_{E,j} \leq a_j \leq T_{L,j}$ 时, 配送员 k 外卖配送到顾客 j 的等待时间 $t_{w,j} = 0$, 将 $t_{w,j}$ 取一个很小的正数; 当 $a_j \leq T_{E,j}$ 时, $t_{w,j} = T_{E,j} - a_j$; 当 $a_j \geq T_{L,j}$ 时, $t_{w,j} = a_j - T_{L,j}$. 同时, 配送员对顾客配送外卖的时间窗跨度也会影响对配送路径规划, 在配送时间窗开始时刻一样的顾客 j_1 和顾客 j_2 时, 如果顾客 j_1 的时间窗跨度小于顾客 j_2 , 则顾客 j_1 对于外卖的配送要求更加紧急, 对于配送员来说超时的可能性会越大, 因此会优先顾客 j_1 的外卖配送. 在状态转移过程中加入时间窗跨度影响因素 $f_{width,j} = T_{L,j} - T_{E,j}$.

状态转移概率 $P_{ij|k}(t)$ 表示在第 t 次迭代时, 蚂蚁 k 从节点 i 移动到节点 j 的概率, 移动规则如下所示:

$$j = \begin{cases} \max_{j \in N_{i|k}} \{ [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [1/f_{width,j}]^\gamma [1/t_{w,j}]^\delta \}, \\ r \leq r_0; \\ P_{ij|k}(t), \quad r > r_0. \end{cases} \quad (8)$$

其中: $P_{ij|k}(t)$ 为使用轮盘赌法选择点 j 的概率, 可表示为

$$\frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [1/f_{width,j}]^\gamma [1/t_{w,j}]^\delta}{\sum_{j \in N_{i|k}} ([\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [1/f_{width,j}]^\gamma [1/t_{w,j}]^\delta)};$$

$N_{i|k}$ 是未被蚁群访问过的节点集合, 可以通过概率值选择下一个节点; $\tau_{ij}(t)$ 表示在第 t 次迭代时, 节点 i 与节点 j 形成的边上的信息素的量; α 表示信息素对蚁群搜索的影响大小; $\eta_{ij}(t)$ 表示形成的边对蚂蚁的可见度大小, $\eta_{ij}(t) = 1/d_{ij}$, d_{ij} 为节点 i 到节点 j 的距离, 如果两个节点距离越远, 表示蚂蚁的可见度越小; β 是可见性对蚁群搜索的影响, β 越大, 蚁群更有可能采取可见度高的路径; γ 表示时间窗跨度因素导致蚂蚁对节点选择影响的大小; δ 表示等待时间因素导致蚂蚁对节点选择影响的大小; r 为 $[0, 1]$ 上服从均匀分布的随机变量; r_0 为控制转移规则参数, 并且 $0 \leq r_0 \leq 1$.

蚂蚁在访问完该条路径后, 会在该路径上留下信息素, 对其他蚂蚁起引导作用. 每次迭代后, 蚂蚁会重复经过该路径, 会使得信息素更新, 信息素的表达式为

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij|k}(t), \quad (9)$$

$$\Delta\tau_{ij|k}(t) = \begin{cases} Q/L_k, & j \in V; \\ 0, & \text{其他}. \end{cases} \quad (10)$$

其中: ρ 为信息素挥发因子, 大小为 $0 < \rho < 1$; $\Delta\tau_{ij|k}(t)$ 表示为第 t 次迭代的第 k 只蚂蚁经过路径留下的信息素, 初始 $\Delta\tau_{ij|k}(0)$ 设为 1; L_k 表示第 k 只蚂蚁从配送中心出发到回到配送中心访问的总路径长度; Q 为设定的常数, 本文实验给定为 5.

令一组蚂蚁从配送中心出发, 每一只蚂蚁按照式(8) 规则选择下一访问节点 j , 当根据规则找不到合理的下一节点 j 时, 蚂蚁返回配送中心, 则该只蚂蚁对应的配送路径规划完成, 一组蚂蚁全部遍历完节点后, 即生成规划的外卖的配送路径。

2.3 大规模邻域搜索算法(LNS)

大规模邻域搜索算法从蚁群算法所得出的初始解进行优化, 通过算法中的“破坏”和“修复”两个阶段对初始解进行优化操作, 从而得到最优解. 假设通过蚁群算法得到的配送路径为 S_0 , 设定最优解 $S_{best} =$

S_0 , 即记录当前路径; 将路径 S_{best} 中的顾客随机抽出, 即进行破坏操作, 记录当前的路径为 S_d ; 对路径 S_d 进行顾客插入的操作, 即进行修复操作, 记录当前的路径为 S_r . 比较路径 S_r 罚函数的值与最优解路径 S_{best} 罚函数的值大小: 如果路径 S_r 罚函数的值小于最优解路径 S_{best} 罚函数的值, 则认为破坏、修复后的路径 S_r 优于路径 S_{best} , 于是有 $S_{best} = S_r$, 实现更新最优解, 否则重新进行破坏、修复两个操作. 当满足终止条件的罚函数值时, 即输出最优路径 S_{best} , 其流程如图 1 所示.

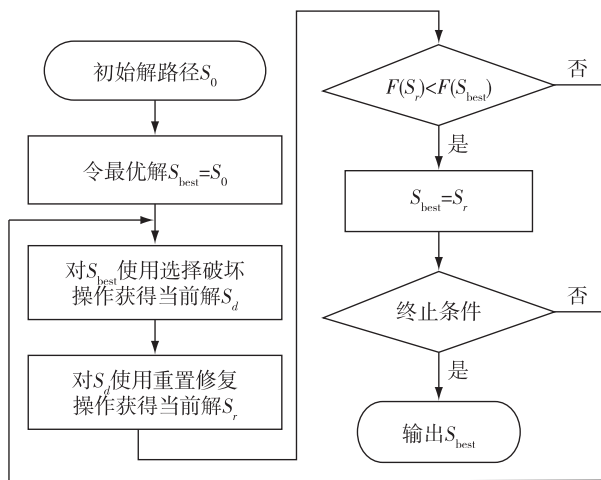


图 1 大规模邻域搜索算法改善初始解质量流程
Fig. 1 Processes to improve initial solution quality by LNS algorithm

2.4 改进蚁群算法流程

将蚁群算法和大规模邻域搜索算法结合, 使用大规模邻域搜索算法对蚁群算法所求得的初始解进行优化操作. 大规模邻域搜索算法优化初始解的实质是通过将蚁群所遍历路径进行破坏和重新修复操作, 防止蚁群陷入局部最优解, 从而改善初始解, 得到优质解. 解决带载重约束和时间窗约束的外卖配

送算法原理如图 2 所示.

2.5 改进蚁群算法步骤

改进蚁群算法的伪代码如表 1 所示.

表 1 改进蚁群算法的伪代码

Table 1 Pseudocode to improve ACO algorithm	
改进蚁群算法	
输入:	外卖配送信息、迭代终止次数 T
输出:	最终配送路线 S_{best} 、最优成本 $F(S_{best})$
1)	$t = 1$ (initialization)
2)	初始化蚁群参数, 设置蚂蚁在配送中心
3)	蚂蚁根据式 (11) 规则, 依次访问节点, 生成初始可行路径 S
4)	while $t \leq T$ do
5)	信息素 τ 更新, 计算当前成本 $F(S)$
6)	最优路径 $S_{best} = S$
7)	路径 S_{best} 经过 LNS 算法“破坏”、“修复”后得到路径 S_r
8)	比较 $F(S_{best})$ 与 $F(S_r)$ 大小, 令 $F(S) = \min(F(S_{best}), F(S_r))$
9)	$S_{best} = S$
10)	$t = t + 1$
11)	end
12)	return $S_{best}, F(S_{best})$

3 仿真分析

在 MATLAB 平台进行仿真实验, 对本文所提出的改进蚁群算法、未改进的蚁群算法和遗传算法, 针对不同数量的顾客进行外卖配送仿真分析. 根据外卖配送实际情况, 模拟外卖订单的低峰期和高峰期两个场景, 验证本文所提出的改进蚁群算法的有效性. 设定外卖订单的低峰期和高峰期顾客人数分别为 25 和 110, 配送员由配送中心出发, 在规定的窗范围内进行外卖配送, 最后返回配送中心. 设定迭代次数 $T = 50$, 配送员的最大载货容量设定为 15.

3.1 订单数量较少场景仿真分析

表 2 为 25 位顾客和配送中心的信息数据, 使用 MATLAB 软件进行模拟配送员进行外卖配送的场

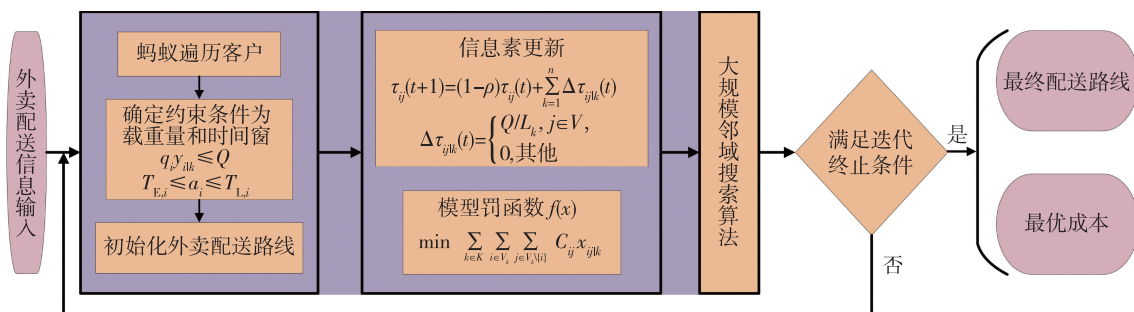


图 2 改进蚁群算法流程
Fig. 2 Processes to improve ACO algorithm

景,分别使用未改进的蚁群算法、遗传算法和本文提出的改进蚁群算法进行配送员的配送路径规划,配送路线和罚函数最优值变化曲线分别如图3、图4和图5所示,详细配送路径分别如表3、表4和表5所示。

表2 25位顾客信息

Table 2 Information of 25 customers

编号	x 坐标	y 坐标	需求量	时间窗开始	时间窗结束
0	40	50	0	0	1 236
1	45	68	2	912	967
2	45	70	2	825	870
3	42	66	2	65	146
4	42	68	2	727	782
5	42	65	2	15	67
⋮	⋮	⋮	⋮	⋮	⋮
22	28	52	2	812	883
23	28	55	2	732	777
24	25	50	2	65	144
25	25	52	2	169	224

表3 蚁群算法25位顾客配送路径规划

Table 3 Delivery routes for 25 customers planned by ACO algorithm

序号	配送路线
①	0→20→5→13→3→17→18→25→0
②	0→24→10→15→16→11→9→6→0
③	0→8→19→14→12→4→23→22→0
④	0→7→2→1→21→0

表4 遗传算法25位顾客配送路径规划

Table 4 Delivery routes for 25 customers planned by genetic algorithm

序号	配送路线
①	0→20→24→25→23→22→21→0
②	0→13→17→18→19→15→16→4→0
③	0→5→3→7→8→9→12→14→0
④	0→1→2→6→11→10→0

由图3—5可知:使用蚁群算法在迭代50次后,所得到外卖配送路径的罚函数最优值为8 180;使用遗传算法在迭代50次后,所得到外卖配送路径的罚函数最优值为6 886;本文设计的改进蚁群算法在迭代50次后,所得到外卖配送路径的罚函数最优值为6 615。因此,可以求得改进蚁群算法在订单数量较少的场景下,与蚁群算法相比提升性能为19.13%,与遗传算法相比提升性能为3.94%。

表5 改进蚁群算法25位顾客配送路径规划

Table 5 Delivery routes for 25 customers planned by IACO algorithm

序号	配送路线
①	0→13→17→18→19→15→23→0
②	0→5→8→10→11→16→14→12→0
③	0→3→7→9→6→4→2→1→0
④	0→20→24→25→22→21→0

3.2 订单数量较多场景下的仿真分析

表6为110位顾客和配送中心的信息数据,使用MATLAB模拟配送员进行外卖配送的场景。分别

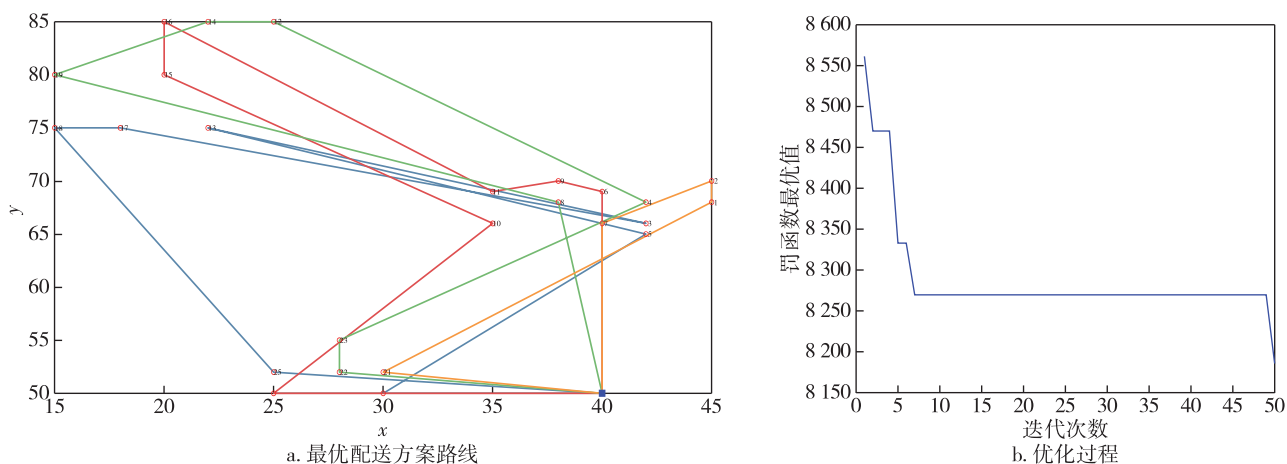


图3 蚁群算法规划25位顾客配送路径

Fig. 3 Using ACO algorithm to plan delivery routes for 25 customers

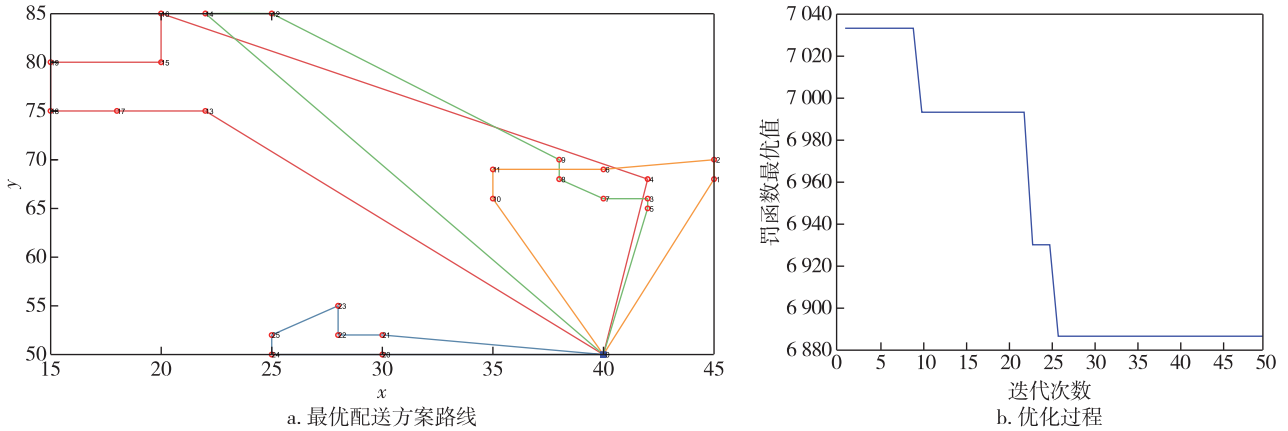


图4 遗传算法规划 25 位顾客配送路径

Fig. 4 Using genetic algorithm to plan delivery routes for 25 customers

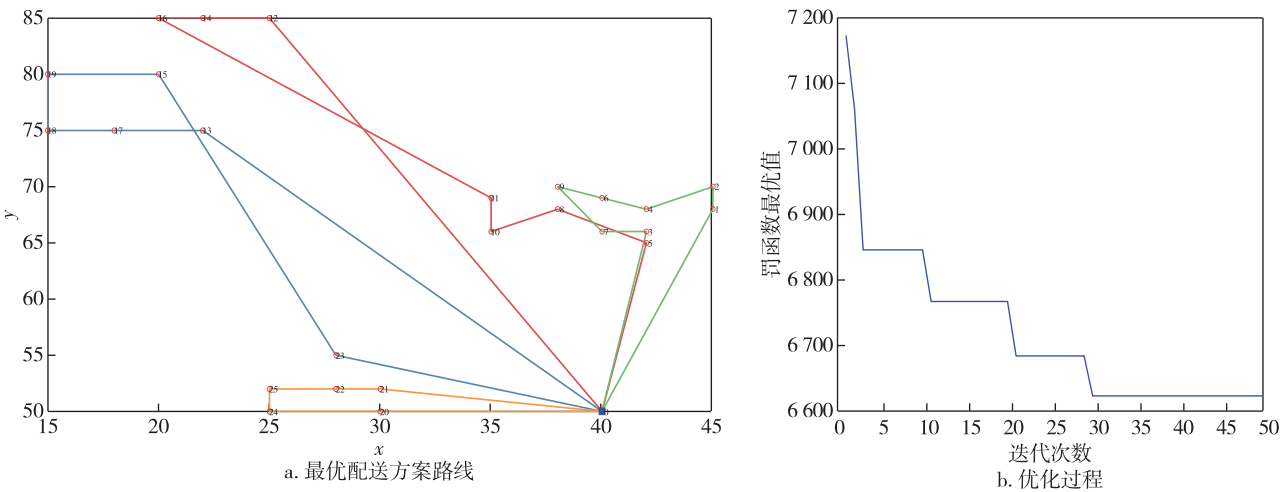


图5 改进蚁群算法规划 25 位顾客配送路径

Fig. 5 Using IACO algorithm to plan delivery routes for 25 customers

使用未改进的蚁群算法、遗传算法和本文提出的改进蚁群算法进行配送路径规划.配送路线图和罚函数最优值变化曲线如图 6、图 7 和图 8 所示,详细配送路径如表 7、表 8 和表 9 所示.

由图 6—8 可知:使用蚁群算法在迭代 50 次后,所得到外卖配送路径的罚函数最优值为 3.945×10^4 ;使用遗传算法在迭代 50 次后,所得到外卖配送路径的罚函数最优值为 4.413×10^4 ;改进蚁群算法在迭代 50 次后,所得到外卖配送路径的罚函数最优值为 2.215×10^4 .本文提出的改进蚁群算法在订单数量较多的场景下,与蚁群算法相比,性能提升为 43.85%,与遗传算法相比性能提升为 49.81%.

表 6 110 位顾客信息

Table 6 Information of 110 customers

序号	x 坐标	y 坐标	需求量	时间窗开始	时间窗结束
0	40	50	0	0	1 340
1	45	68	1	912	967
2	45	70	1	825	870
3	42	66	1	65	146
4	42	68	1	727	782
⋮	⋮	⋮	⋮	⋮	⋮
104	40	90	1	975	1 010
105	10	80	1	1 200	1 262
106	15	60	1	820	880
107	20	40	1	900	940
108	10	20	1	1 100	1 163
109	40	30	1	920	980
110	70	20	1	800	867

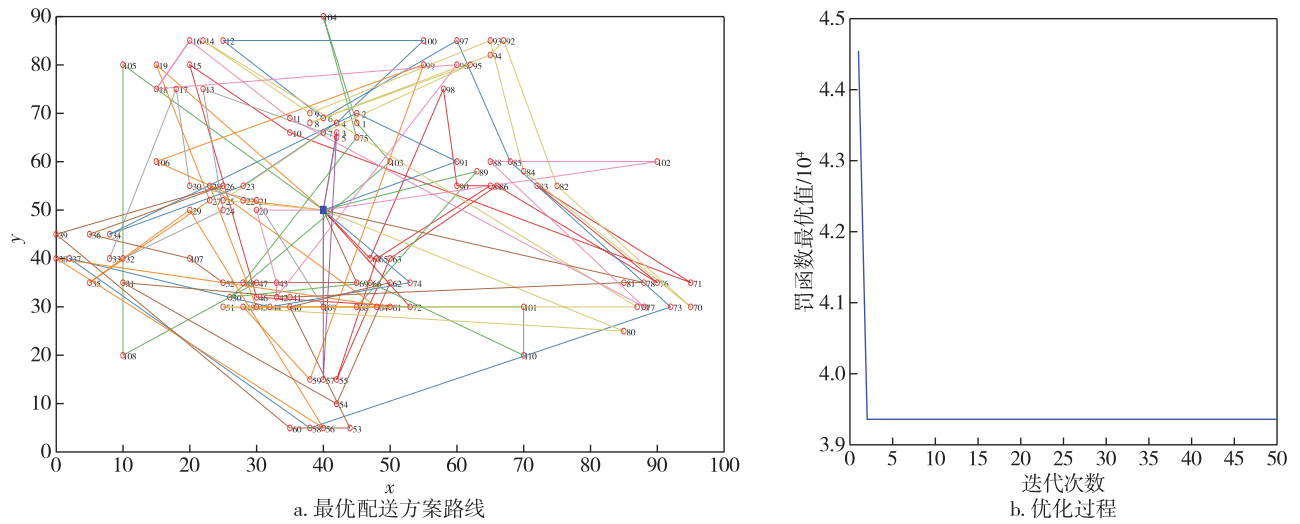


图 6 蚁群算法规划 110 位顾客配送路径

Fig. 6 Using ACO algorithm to plan delivery routes for 110 customers

表 7 蚁群算法 110 位顾客配送路径规划

Table 7 Delivery routes for 110 customers planned by ACO algorithm

序号	配送路线
①	0→67→98→90→87→65→55→63→86→76→83→71→10→15→46→72→0
②	0→74→44→37→58→73→85→97→6→12→100→4→23→34→2→91→0
③	0→89→68→101→110→66→50→103→1→104→75→49→47→108→105→0
④	0→19→27→35→29→56→38→61→45→28→64→48→59→99→106→22→0
⑤	0→25→7→95→8→94→92→82→70→84→93→9→14→79→51→80→0
⑥	0→78→81→41→31→54→62→40→53→60→39→26→36→107→52→69→0
⑦	0→20→43→42→96→18→16→11→77→88→102→0
⑧	0→5→13→24→32→33→17→30→21→109→0
⑨	0→57→3→0

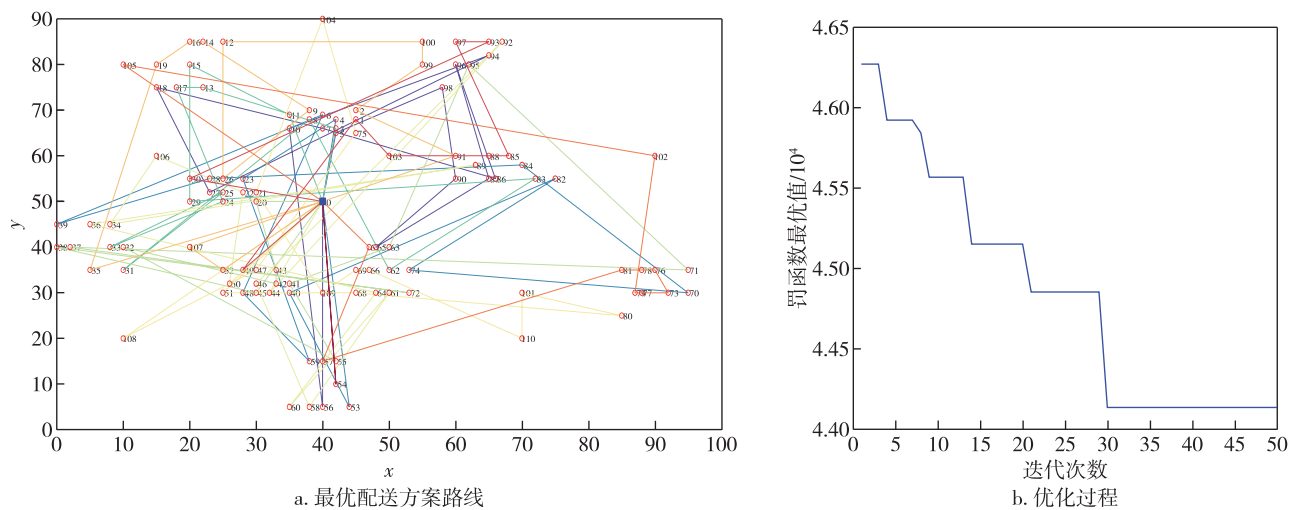


图 7 遗传算法规划 110 位顾客配送路径

Fig. 7 Using genetic algorithm to plan delivery routes for 110 customers

表 8 遗传算法 110 位顾客配送路径规划

Table 8 Delivery routes for 110 customers planned by genetic algorithm

序号	配送路线
①	0→5→98→90→65→87→96→86→7→18→27→94→10→56→0
②	0→53→40→82→74→70→84→28→39→6→48→59→23→4→0
③	0→3→13→17→24→33→25→31→8→62→83→29→15→11→0
④	0→20→43→42→32→55→41→63→95→71→37→72→38→45→0
⑤	0→44→92→46→58→61→60→64→36→89→34→106→22→50→0
⑥	0→51→68→80→101→110→66→69→109→21→104→75→47→108→0
⑦	0→35→19→16→14→9→26→12→100→99→2→91→107→52→0
⑧	0→67→57→81→78→76→73→77→79→102→105→0
⑨	0→30→93→97→85→88→103→1→49→0
⑩	0→54→0

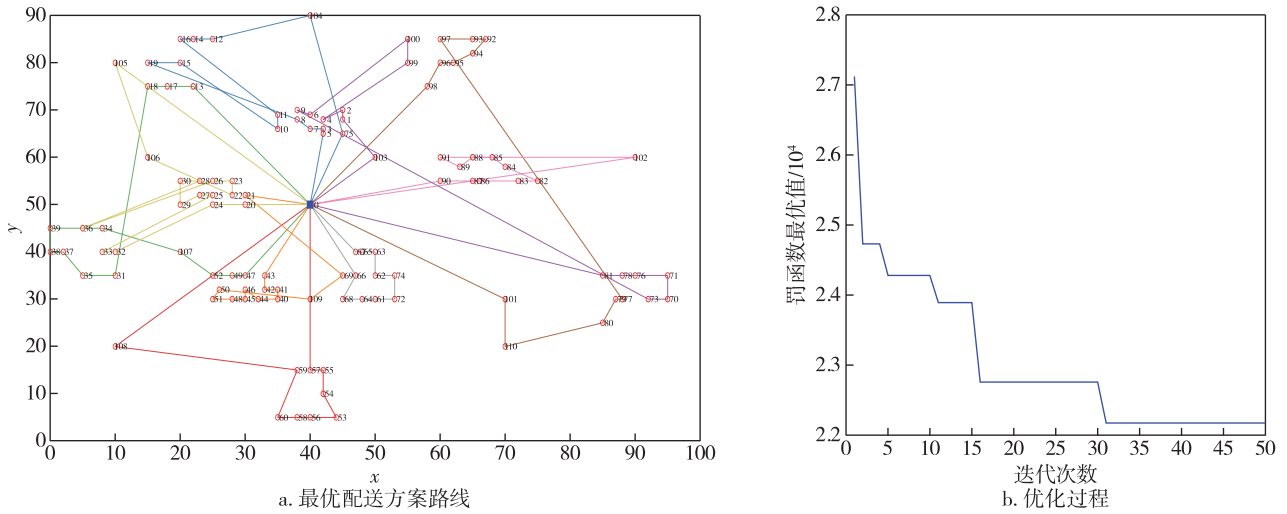


图 8 改进蚁群算法规划 110 位顾客配送路径

Fig. 8 Using IACO algorithm to plan delivery routes for 110 customers

表 9 改进蚁群算法 110 位顾客配送路径规划

Table 9 Delivery routes for 110 customers planned by IACO algorithm

序号	配送路线
①	0→57→55→54→53→56→58→60→59→108→0
②	0→5→3→7→8→19→15→10→11→16→14→12→104→75→0
③	0→13→17→18→31→35→37→38→39→34→107→52→49→47→0
④	0→43→42→41→40→44→46→45→48→51→50→109→69→21→0
⑤	0→20→24→32→33→25→27→29→30→28→36→26→23→22→106→105→0
⑥	0→98→96→95→94→92→93→97→77→79→80→110→101→0
⑦	0→90→87→86→83→82→84→85→88→89→91→102→0
⑧	0→67→65→63→62→74→72→61→64→68→66→0
⑨	0→81→78→76→71→70→73→9→6→100→99→4→2→1→103→0

3.3 仿真结果分析

由仿真结果可知,当订单数量较少时,蚁群算法、遗传算法和本文所提出的改进蚁群算法都可以根据顾客信息进行合理的外卖配送路径规划,并且

最后所得到的路径均未出现违反时间窗等情况.整个外卖配送过程中的一个综合判定,表现为所设定罚函数的最优值大小:最优值越小,说明配送过程中违反约束的情况越少,并且综合配送成本越低,反之

则违反约束情况多,并且配送成本高.蚁群算法、遗传算法和本文所提出的改进蚁群算法,进行外卖配送路径规划的罚函数最优值比较,结果如表 10 所示.根据表 10 中罚函数的最优值结果分析可得:顾客数量较少时,改进后蚁群算法的性能相比蚁群算法和遗传算法分别提升 19.13% 和 3.94%;顾客数量较多时,改进后蚁群算法的性能相比蚁群算法和遗传算法分别提升 43.85% 和 49.81%.结果表明,本文提出的改进蚁群算法所求解的外卖配送路径的性能有较大提升,说明所提出的算法是有效的,可以合理运用在订单数量较多的场景.

表 10 算法性能比较

Table 10 Algorithm performance comparison

顾客数量	罚函数最优值		
	蚁群算法	遗传算法	改进蚁群算法
25	8 180	6 886	6 615
110	3.945×10^4	4.134×10^4	2.215×10^4

4 结论

本文研究了有时间窗约束和车载容量约束的 VRP 问题,提出了一种改进蚁群算法,既有蚁群算法快速寻找解的优点,又有大规模邻域算法优化解质量的优点.结合 MATLAB 仿真平台进行验证,选取顾客订单数量较少和较多的两个场景,选用蚁群算法、遗传算法和本文所提出的改进蚁群算法进行对照仿真,根据所得到的最优配送方案路线和罚函数的最优值可以得出,本文提出的改进蚁群算法,可以较好地规划外卖配送路径,满足约束要求,合理提升外卖配送效率.

参考文献

References

[1] 初良勇, 闫森, 胡美丽, 等. 生鲜产品物流配送路径优化研究综述及展望[J]. 物流研究, 2021, 3(1): 5-13
CHU Liangyong, YAN Miao, HU Meili, et al. Review and prospect of fresh product logistics distribution route optimization[J]. Logistics Research, 2021, 3(1): 5-13

[2] 赵金龙, 蒋忠中, 万明重, 等. 考虑配送截止时间的“货到人”订单拣选优化问题研究[J/OL]. 中国管理科学: 1-12 [2022-12-08]. <https://doi.org/10.16381/j.cnki.issn1003-207x.2022.1049>
ZHAO Jinlong, JIANG Zhongzhong, WAN Mingzhong, et al. Optimization of parts-to-picker order picking with due dates[J/OL]. China Management Science: 1-12 [2022-12-08]. <https://doi.org/10.16381/j.cnki.issn1003-207x.2022.1049>

[3] 周成昊, 吕博轩, 周翰宇, 等. 以商圈为中心的 O₂O 动态外卖配送路径优化模型与算法[J]. 运筹学学报, 2022, 26(3): 17-30
ZHOU Chenghao, LÜ Boxuan, ZHOU Hanyu, et al. Optimization model and algorithm for online to offline dynamic take-out delivery routing problem centered on business districts[J]. Operations Research Transactions, 2022, 26(3): 17-30

[4] 刘云忠, 宣慧玉. 车辆路径问题的模型及算法研究综述[J]. 管理工程学报, 2005, 19(1): 124-130
LIU Yunzhong, XUAN Huiyu. Summarizing research on models and algorithms for vehicle routing problem[J]. Journal of Industrial Engineering and Engineering Management, 2005, 19(1): 124-130

[5] 毕华玲, 赵亚风, 卢福强, 等. 考虑客户风险偏好的第四方物流路径问题研究[J/OL]. 南京信息工程大学学报(自然科学版): 1-12 [2022-06-16]. <http://kns.cnki.net/kcms/detail/32.1801.N.20220615.1221.002.html>
BI Hualing, ZHAO Yafeng, LU Fuqiang, et al. Research on 4PL routing problem considering customer risk preference[J/OL]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition): 1-12 [2022-06-16]. <http://kns.cnki.net/kcms/detail/32.1801.N.20220615.1221.002.html>

[6] 苏涛, 韩庆田, 李文强, 等. VRP 问题蚁群算法研究[J]. 计算机与现代化, 2012(11): 18-21, 25
SU Tao, HAN Qingtian, LI Wenqiang, et al. Research on VRP problem based on ant colony algorithm[J]. Computer and Modernization, 2012(11): 18-21, 25

[7] 贺琪, 官礼和, 崔焕焕. 硬时间窗 VRP 的混合变邻域禁忌搜索算法[J]. 计算机工程与应用, 2023, 59(13): 82-91
HE Qi, GUAN Lihe, CUI Huanhuan. Hybrid variable neighborhood tabu search algorithm for vehicle routing problem with hard time window[J]. Computer Engineering and Application, 2023, 59(13): 82-91

[8] 王东. 改进多种群遗传算法的研究及其在车辆路径优化的应用[D]. 南宁: 广西大学, 2016
WANG Dong. Research on improved multi-population genetic algorithm and its application in vehicle routing optimization[D]. Nanning: Guangxi University, 2016

[9] 董平先, 郭放, 陈晨, 等. 基于优化蚁群算法的电缆敷设路径规划[J]. 南京信息工程大学学报(自然科学版), 2023, 15(2): 210-217
DONG Pingxian, GUO Fang, CHEN Chen, et al. Cable laying path planning based on optimized ant colony algorithm[J]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition), 2023, 15(2): 210-217

[10] Zhang X, Wang J Q. Hybrid ant algorithm and applications for vehicle routing problem[J]. Physics Procedia, 2012, 25: 1892-1899

[11] 王晓东, 张永强, 薛红. 基于改进蚁群算法对 VRP 线路优化[J]. 吉林大学学报(信息科学版), 2017, 35(2): 198-203
WANG Xiaodong, ZHANG Yongqiang, XUE Hong. Improved ant colony algorithm for VRP[J]. Journal of Jilin University (Information Science Edition), 2017, 35(2): 198-203

- 198-203
- [12] Donati A V, Montemanni R, Casagrande N, et al. Time dependent vehicle routing problem with a multi ant colony system[J]. *European Journal of Operational Research*, 2008, 185(3):1174-1191
- [13] Kalayci C B, Can K Y. An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery [J]. *Expert Systems with Applications*, 2016, 66:163-175
- [14] 李琳, 刘士新, 唐加福. 改进的蚁群算法求解带时间窗的车辆路径问题[J]. *控制与决策*, 2010, 25(9):1379-1383
LI Lin, LIU Shixin, TANG Jiafu. Improved ant colony algorithm for solving vehicle routing problem with time windows[J]. *Control and Decision*, 2010, 25(9):1379-1383
- [15] Blum C, Ereemeev A, Zakharova Y. Hybridizations of evolutionary algorithms with large neighborhood search[J]. *Computer Science Review*, 2022, 46:100512
- [16] 徐倩, 熊俊, 杨珍花, 等. 基于自适应大邻域搜索算法的外卖配送车辆路径优化[J]. *工业工程与管理*, 2021, 26(3):115-122
XU Qian, XIONG Jun, YANG Zhenhua, et al. Based on an adaptive large neighborhood search algorithm for take-out delivery vehicle routing optimization[J]. *Journal of Industrial Engineering and Management*, 2021, 26(3):115-122
- [17] Natalia C, Triyanti V, Setiawan G, et al. Completion of capacitated vehicle routing problem (CVRP) and capacitated vehicle routing problem with time windows (CVRPTW) using bee algorithm approach to optimize waste picking transportation problem [J]. *Journal of Modern Manufacturing Systems and Technology*, 2021, 5(2):69-77

Takeout delivery path planning based on improved ant colony optimization algorithm

TANG Chuanyin¹ ZHANG Mingli¹ LI Jinghong^{2,3} YUAN Ying⁴ Wei Meirong¹

¹ School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China

² School of Mechanical Engineering, Shanghai Dianji University, Shanghai 201306, China

³ School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China

⁴ School of Business Administration, Northeastern University, Shenyang 110167, China

Abstract It is impossible for takeaway delivery staff to plan the takeout delivery route balancing rationality and efficiency. To address this problem, an Improved Ant Colony Optimization (IACO) algorithm is proposed. The initial routes are obtained using the Ant Colony Optimization (ACO) algorithm and then optimized using Large Neighborhood Search (LNS) algorithm. The solution quality is improved by combining the ACO algorithm with the LNS algorithm. The proposed algorithm is verified by simulating the delivery routes for different number of takeout orders. Comparative analysis shows that the proposed IACO algorithm can increase the takeout delivery efficiency, according to the optimal distribution plan and the ideal value of the objective penalty function. The proposed strategy can enhance the intelligence and promote the long-term growth of the delivery system of Internet-connected takeout platforms.

Key words improved ant colony optimization (IACO); large neighborhood search (LNS); takeout delivery; distribution schemes