

O2O 模式下外卖骑手的配送路径优化

靳志宏^{*1}, 鞠新诚¹, 郭加佳¹, 杨珍花²

(1. 大连海事大学 交通运输工程学院, 辽宁 大连 116026;

2. 山东科技大学 经济管理学院, 山东 青岛 266590)

摘要: 基于大连市某外卖配送站点的运营实例, 针对午餐高峰时段涌入的大量订单, 对骑手的配送路径进行优化。以最大化运输效率为目标, 综合考虑外卖配送的实际约束, 有针对性地构建骑手配送路径优化的混合整数规划模型, 开发改进型蚁群算法对实例进行求解。然后将所得结果与行业实际数据、小规模算例的 CPLEX 精确解以及传统的蚁群算法进行对比, 从多方面验证了算法的合理性与有效性。最后, 详细分析了算法参数对优化结果的影响机理, 可为外卖配送行业提供决策支持。

关键词: O2O; 外卖; 配送路径优化; 取送货; 蚁群算法

中图分类号: F719.3; F724.6

文献标志码: A

Optimization on distribution routes of the takeaway delivery staff under the O2O mode

JIN Zhi-hong^{*1}, JU Xin-cheng¹,

GUO Jia-jia¹, YANG Zhen-hua²

(1. Transportation Engineering College,
Dalian Maritime University, Dalian 116026, China;

2. College of Economics and Management,
Shandong University of Science and
Technology, Qingdao 266590, China)

Abstract: Based on the operation example of a takeaway distribution station in Dalian, the distribution route of riders was optimized for the influx of orders during the lunch rush hour. Aiming at maximizing transportation efficiency and by considering the actual constraints of takeout distribution, a mixed integer programming model for rider distribution route optimization was constructed, and an improved ant colony algorithm was developed to solve the case. Then the results were com-

pared with the actual industry data, the CPLEX exact solution of the small-scale example and the traditional ant colony algorithm, the rationality and effectiveness of the algorithm were verified from many aspects. Finally, the influence mechanism of the algorithm parameters on the optimization results was analyzed in detail, which can provide decision support for take-out distribution industry.

Key words: online to offline (O2O); takeaway; distribution routing optimization; pickup and delivery; ant colony algorithm

0 引言

近年来, 我国的 O2O (Online to Offline) 电子商务发展高度繁荣, 商户的生产力过剩, 消费者对订餐的便捷性要求更高, 外卖成为该双向需求下的衍生产物。O2O 模式下, 一项外卖订单自产生至完成的流程为: (1) 消费者在第三方平台下单; (2) 相应商家接受订单并开始备餐, 同时相关骑手在第三方平台接收到订单配送任务, 从站点出发, 前往商家取餐; (3) 骑手取到餐品后将其送至下单客户处, 完成订单。简言之, 外卖配送系统中存在三类主体, 即商家、客户和骑手, 商家与客户之间形成了一定量的 OD (origin-destination), 明确订单配送需求, 需要为骑手规划适当的行车路线, 以满足需求并达到距离之和最小、行驶时间最短、工作效率最高等目标。在这一过程中, 第三方平台负责实时监测订单出现及被执行的状态, 为

收稿日期: 2019-06-20; 修回日期: 2019-07-19.

基金项目: 国家自然科学基金面上项目(71572023; 71702019); 欧盟 H2020 项目(MSCA-RISE 777742-56); 大连市领军人才项目(2018-573); 中央高校基本科研业务费专项资金资助项目(3132019301; 3132019031)。

作者简介: 靳志宏^{*} (1963 -), 男, 博士, 教授, 博士生导师, E-mail: jinzhihong@dlmu.edu.cn; 鞠新诚 (1995 -), 男, 硕士生, E-mail: maxandmaria@qq.com.

订单指派骑手并提供服务评价等附加功能。

外卖配送的订单存在待取餐和待送餐的不同需求点,且成对出现;各需求点存在严格的时间窗要求;待取餐与待送餐的节点有严格的被访问顺序,即必须先前往商家点取餐,随后才能去对应的客户点送餐。这些特点与车辆路径问题^[1]的一个分支,即带时间窗的取送货车车辆路径问题(Pickup and Delivery Vehicle Routing Problem with Time Window, PDVRPTW)十分相似,可以借鉴相关研究成果。

目前 O2O 外卖平台所采取的生成配送路径的方法为骑手与调度员的双重主观协调,每个站点配有一名专门的调度,负责协调隶属于该站点的骑手与订单的匹配,订单爆发高峰期,凭借经验对骑手的订单指派情况进行调整。同时平台允许骑手结合自身实际情况进行订单互换、路线变更等。然而这两种方式都是基于历史经验和主观判断进行操作的,并无实际数值与科学依据。外卖的商品属性决定了订单配送的强时效性。同时订单极易在特定时间段爆单,需要对骑手的配送路径进行高效地调度优化,因此,行业亟待对该问题进行研究,并开发高效的算法进行求解。

针对 PDVRPTW 问题,吴腾宇等^[2]基于 O2O 模式下需求无法确定的现状,提出了带有取送货的在线旅行商问题,当出现新需求需要返回原点时,寻求最短的配送服务时间。张庆华等^[3]将电商平台配送以及退换货两个运输方向进行整合,模型中的节点同时存在取货需求与送货需求,并且对于不满足时间窗的部分采取惩罚成本;随后,运用改进的蚁群算法对问题进行求解,采用时差法来提高蚁群算法的初始搜索效率,通过轮盘赌进行状态转移来构建路径,同时对已有的解进行变邻域搜索来扩大解的空间,从而跳出局部最优解陷阱。Marilène 等^[4]对存在多堆栈以及时间窗约束的取送货问题进行了研究。针对问题中存在多个堆栈,且货物体积不同、占据的空间以及堆栈数不同,相应地对模型进行改进,并利用分支定价-割平面法对问题进行求解。

针对车辆路径问题的求解算法,Chávez^[5]在遗传算法生成初始解的过程中采用两阶段策略,第一阶段运用聚类算法,第二阶段插入 4 种杂交算子,使初始种群呈现多样化。数例实验证明所提

出的方法有利于产生高度多样化的种群,提高优化效率。Inmaculada 等^[6]针对固定车辆(Driver Consistency)服务同一客户的周期性车辆路径问题(Periodic Vehicle Routing Problem, PVRP)进行研究,在连续的时间段中可以一次或多次服务客户,且部分客户存在固定的服务顺序。数学模型以总运输成本最低为目标,同时根据数学关系提出三个强约束进行对比。运用分支定界-割平面的方法对问题进行求解,验证了模型以及强约束的有效性。Armando 等^[7]对有多个堆栈的旅行商问题进行了研究,在“后上先下”的先决条件下,运用分支定界和割平面法寻求满足时间窗等约束的最短路径的最优解。

针对外卖配送问题,王帅等^[8]对 O2O 外卖配送路径的优化问题进行了研究。由于外卖受人为因素、交通路况等干扰,无法精确提供完整的预订路线。因此,在传统的车辆路径问题的基础上,增加了问题的随机性并大大提高了难度。该研究通过定义随机旅行时间,即任意两点之间的旅行时间为一个范围内随机波动的时间来中和实际路况带来的影响,同时建立动态模型来应对订单的变化,并运用遗传算法对其求解。陈萍等^[9]提出一个基于时间参数的顾客满意度函数,优化目标是使各订单总的时间满意度最大,并采用遗传算法对外卖配送路径问题进行求解。

近年来,蚁群算法已经被国内外大量学者应用于求解 VRP 问题,并取得了良好的效果。陈希琼等^[10]对同时取送货的车辆路径问题进行了研究,构建了双目标模型,并在蚁群算法中加入禁忌搜索表以及贪婪转移准则,从而在迭代过程中进行局部的禁忌搜索以产生更优解。柴获等^[11]对带时间窗的车辆路径问题(Vehicle Routing Problem with Time Window, VRPTW)进行了研究,通过加权将双目标转化为单目标,随后构建双层蚁群算法。首先,根据路径变化产生动态的信息素,进而产生不同的搜索空间;随后,根据状态转移规则进行路线规划。裴振兵等^[12]针对车辆路径问题提出了一种改进型蚁群算法,将信息素和期望互锁来动态自适应调整,同时对距离因子重新定义来提高算法的效率。杨鹏等^[13]对需求可拆分的 VRPTW 问题进行了研究,对蚁群算法的信息素采用全局更新,从而提高搜索效率。Schyns^[14]采用蚁

群算法求解动态车辆路径问题.

综上所述,目前大多研究的是车辆路径问题与PDVRPTW,尽管有部分相似点,但是与O2O外卖配送路径优化的实际情况仍有部分差异.现有研究对O2O模式下外卖配送路径优化的研究较少,且并没有针对O2O外卖的实际特点设计算法,因此实际优化效果并不理想.

本文对O2O外卖骑手的配送路径优化问题进行研究,与以往的研究相比,引入了订单效率最大化的概念,即完成一项订单的平均耗时最少.数学模型以站点的订单完成效率最大为优化目标,并针对外卖配送路径优化这一现实约束较强的问题建立模型;根据外卖配送存在同一地点多个订单、可连续取送餐以及订单中两点相对应的这三个特点提出了几种新的适用性强的算子加入蚁群算法中;随后,根据大连市某站点的外卖平台提供的数据,以真实的运营数据构造测试实例,通过求解验证了模型与算法的有效性;采用CPLEX对小规模算例进行求解,对比本文所提出的算法,同时,运用传统蚁群算法对比,验证了算法的改良效果;最后,针对算法中各参数的影响对其进行分析,并结合实际O2O外卖的特点提出建议.

1 模型构建

1.1 假设条件

为更好地对问题进行研究,便于模型的建立和求解,结合实际情况对问题作出以下假设:

(1) 同一商家可接多个客户的不同订单,同一客户仅在一个商家处下订单.

(2) 调度期初始时平台骑手均从站点出发,完成所有订单配送任务后返回站点.

(3) 每位骑手只经过各客户点一次,且每位客户有且仅有一名骑手提供配送服务.

(4) 由于外卖订单量的集中爆发产生在午餐高峰期,即11点至13点间,其他时间段压力较小,运力充足,因此,本文研究的问题中调度期跨度为2h.

(5) 由于外卖体积较小,且骑手的装载量上限较高,因此,装载量限制可忽略.而在实际生产作业中,若存在量大的订单,站点则会指派专人提供配送,该种情形不在优化范围内.

(6) 骑手只在商家点取餐,只在客户点送餐.

(7) 骑手在商家点处与客户点处取/送餐品的耗时忽略不计.

(8) 站点、商家点以及客户点之间的距离已知,骑手在任意两点之间的行驶时间已知.

(9) 所有订单均有硬时间窗限制,硬时间窗的最早开始时刻代表相应商家所允许的最早取餐时间,骑手早于这一时刻到达需等待;硬时间窗的最晚结束时刻代表相应客户节点所能接受的最迟送餐时刻,骑手晚于这一时刻到达则会被拒收.

1.2 符号含义

0: 站点编号;

K : 站点的骑手集合;

k : 骑手编号;

$D = \{1, \dots, N\}$: 客户点集合;

$P = \{N+1, \dots, 2N\}$: 商家点集合,其中包含虚拟商家.由于每个客户节点只对应一项订单,商家可接受多个客户的订单,二者存在一对多的关系,为清晰地表示订单的OD并方便建模,当商家 i 接受了多个订单时,设置虚拟商家 i' ,令 $d_{ii'} = 0$,使得商家节点与客户节点一一对应, $(N+i')$ 为某项订单的OD;

$V = \{0\} \cup D \cup P$: 所有点的集合;

$A = \{(i, j) \mid i, j \in V\}$: 弧的集合;

t_{ij} : 骑手从 i 点行驶至 j 点所需时间;

R : 订单集合;

r : 订单编号;

O_r : 订单 r 对应的商家节点;

D_r : 订单 r 对应的客户节点;

$[t_{rl}, t_{ru}]$: 订单 r 的硬时间窗;

T_{\max} : 调度期的时长,即午餐高峰的2h(11:00–13:00);

t_0 : 骑手从站点出发的时刻(将11:00记为0);

ψ : 累积工作效率,即完成的订单总量与骑手实际配送用时的总和之比;

x_{ijk} : 0–1变量,若骑手 k 连续从节点 i 行驶至节点 j ,其值为1,否则为0;

y_{rk} : 0–1变量,若订单 r 指派给骑手 k ,其值为1,否则为0;

t_i^k : 骑手 k 到达节点 i 的时刻;

t_k^{end} : 骑手 k 返回站点的时刻.

1.3 模型建立

$$\max \quad \psi = \frac{\sum_{k \in K} \sum_{r \in R} y_{kr}}{\sum_{k \in K} (t_k^{\text{end}} - t_0)} \quad (1)$$

$$\sum_{k \in K} y_{kr} = 1, \quad \forall r \in R \quad (2)$$

$$\sum_{j \in D \cup P} \sum_{k \in K} x_{ijk} = 1, \quad i \in D \cup P, i \neq j \quad (3)$$

$$\sum_{i \in D \cup P} \sum_{k \in K} x_{ijk} = 1, \quad j \in D \cup P, i \neq j \quad (4)$$

$$t_{O_r}^k \geq t_{r_l}, \quad r \in R, k \in K \quad (5)$$

$$t_{D_r}^k \leq t_{r_u}, \quad r \in R, k \in K \quad (6)$$

$$\sum_{k \in K} (\sum_{i \in V} x_{ijk} - \sum_{h \in V} x_{jhk}) = 0, \quad j \in D \cup P, i \neq j \neq h \quad (7)$$

$$\sum_{i \in V} x_{iO_r k} + \sum_{j \in V} x_{D_r j k} = 2, \quad r \in R, k \in K \quad (8)$$

$$t_{O_r}^k \leq t_{D_r}^k, \quad r \in R, k \in K \quad (9)$$

$$x_{ijk} + x_{jik} \leq 1, \quad i, j \in D \cup P, k \in K \quad (10)$$

$$t_0^k = 0, \quad k \in K \quad (11)$$

$$t_j^k = t_i^k + t_{ij}, \quad x_{ijk} = 1, \quad k \in K, i, j \in V \quad (12)$$

$$t_k^{\text{end}} \leq T_{\max}, \quad k \in K \quad (13)$$

$$\sum_{j \in P} x_{0jk} = 1, \quad k \in K \quad (14)$$

$$\sum_{i \in D} x_{i0k} = 1, \quad k \in K \quad (15)$$

$$x_{ijk}, y_{rk} \in \{0, 1\}, i, j \in V, r \in R, k \in K \quad (16)$$

式(1)表示目标函数使订单累积工作效率最大,订单完成数量与各骑手所用总时间之和之比最大;式(2)表示每个订单均有骑手配送;式(3)和式(4)表示每一个节点会且只会被一个骑手服务一次;式(5)和式(6)为时间窗约束;式(7)为流量平衡约束;式(8)限制同一订单的OD点由同一骑手访问;式(9)表示针对某一订单,骑手必须先由商家点取餐,随后才能去顾客点送餐;式(10)限制同一骑手对弧的访问次数;式(11)~式(13)表示骑手工作的时间递推关系;式(14)、式(15)表示骑手必须从站点出发,且最终必须返回站点;式

(16)为决策变量取值范围约束。

2 蚁群算法

2.1 生成初始解

算法的编码方式如图1所示,表示2个骑手完成9个订单,9个订单分别是(10,1)、(11,2)、(12,3)、(13,4)、(14,5)、(15,6)、(16,7)、(17,8)、(18,9)。可以看出,所有骑手的配送路径连接在一起,骑手间的配送路径由0隔开。在蚁群算法中,将连接后的路径视作蚂蚁的一条搜索路径。

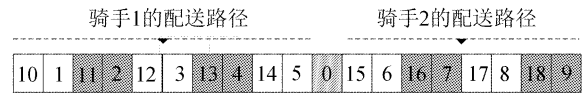


图1 编码方式示意图

Fig.1 Coding mode

运用蚁群算法得到初始解的伪代码及各参数含义如下。

NC: 目前迭代次数; MAXGEN: 最大迭代次数; m : 蚂蚁算法中蚂蚁数; $\text{Tabu}()$: 禁忌搜索表。

初始化: 读取骑手、商家、顾客数据信息,并初始化节点。

While (NC < MAXGEN)

Step0: for $i = 1: m$ (为每只蚂蚁分布初始节点)

Step1: 载入骑手节点

$\text{Tabu}(:, 1) = 1$.

Step2: 初始化商家

while(1)

寻找可使用商家

$\text{Tabu}(:, 2) =$ 符合要求的商家编号;

End.

Step3: 为每个骑手分派顾客

(1) 根据信息素的浓度,将尚未分配的最有可能的订单对应的点分派给该骑手;

(2) 更新骑手的累积工作时间 $T = T + \text{runST}$;

(3) If $T >$ 顾客最晚需求时间或 $T >$ 骑手最大工作时间,则重新分派骑手,否则,将该顾客加入到该节点对应的 $\text{Tabu}(i, \text{length}(\text{visited}) + 1) = \text{to_visit}(\text{Select}(1))$;

(4) 判断所有顾客是否访问完毕:

是: 该路线分配完毕,重新分派新骑手和商家,运行时间清零, $\text{NC} = \text{NC} + 1$;

Tabu(i ,length(visited) + 1) = 1;

Tabu(i ,length(visited) + 2) = 商家编号;

$T = 0$;

否: 回到 Step3.

(5) 判断所有蚂蚁是否分配到路径,是:跳出 step0; 否则继续.

Step4: 计算每条路径的效率,并选出最优效率输出、保存

2.2 节点转移原则

利用蚁群算法进行求解时,每只蚂蚁由一个点转移到另一个点的概率取决于两点之间连线上信息素的浓度.若存在蚂蚁 k 按照这一规则在点 i 与点 j 之间进行转移,且设 $P_{ij}^k(t)$ 为 t 时刻蚂蚁 k 从点 i 转移到点 j 的概率,则有:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & s \in allow_k \\ 0, & s \notin allow_k \end{cases} \quad (17)$$

式中: $\eta_{ij}(t) = 1/d_{ij}$ 为期望函数,用以表示蚂蚁在 t 时刻从点 i 转移到点 j 的期望程度; $allow_k$ 为蚂蚁 k 仍需访问的点的集合; α 为信息素比重因子,即 α 值越大,在相应路径中信息素浓度起到的作用越大; β 为期望函数比重因子, β 值越大,在相应路径中期望函数起到的作用越大,蚂蚁更倾向于转移到距离短的点.

2.3 邻域搜索算子

邻域搜索由 Hensen^[15] 等在 1997 年提出,指根据已有的解,对其进行变换,从而产生不同的新解,跳出局部最优解的陷阱,提升解的质量.

然而在本文的外卖骑手路径优化问题中,若使用传统的 Swap、2-opt、Insert 算子,随机调整一个点极大概率会出现骑手在执行同一订单时,先去顾客点后去商家点的现象.因此,本文根据外卖配送的特点,将传统的邻域搜索算子进行改进,把单个点替换为订单对应的成对的商家点与顾客点,避开不必要的邻域搜索空间,提升搜索与优化效率.本文设计的三种新型邻域搜索算子如下:

(1) Swap* 算子: 交换某两条路径的商家(或客户)点与对应的客户(或商家)点.

(2) Insert* 算子: 将某条路径上的商家(或客户)点与对应的客户(或商家)点插入该路径任意

两点中.

(3) 2-opt* 算子: 将某条路径上的任意两商家(或客户)点以及对应的客户(或商家)点进行交换.

邻域搜索的过程按如下步骤进行:

Step1: 初始化参数设置,令三种邻域搜索算子为 VNS_k $k = \{1, 2, 3\}$,邻域搜索的最大迭代次数为 VM,当前已迭代次数 $Ite = 1$;

Step2: 若 $Ite > VM$,输出当前解,结束算法,否则令 $k = 1$;

Step3: 执行邻域 VNS_k ;

Step4: 对产生的邻域搜索解空间进行局部搜索;

Step5: 判断最优解是否更新,若是,更新最优解,否则令 $k = k + 1$;

Step6: 若 $k < 3$,返回 Step3,否则令 $Ite = Ite + 1$,返回 Step2.

2.4 信息素的更新与释放

在算法迭代过程中,蚂蚁在释放信息素的同时,路径上已有的信息素在逐渐挥发消失,该处以参数 ρ ($0 < \rho < 1$) 表示信息素的挥发度.每当一次路径构造完成后,信息素需要进行一次更新,则有:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}, \quad 0 < \rho < 1 \quad (18)$$

式中: τ_{ij} 为路径 (i, j) 上的信息素浓度; $\Delta\tau_{ij}$ 为路径 (i, j) 上的信息素浓度之和,其计算公式为:

$$\Delta\tau_{ij} = \sum_{k \in K} \Delta\tau_{ij}^k \quad (19)$$

式中: $\Delta\tau_{ij}^k$ 为第 k 只蚂蚁在路径 (i, j) 上释放的信息素的浓度,其计算公式为:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & x_{ijk} = 1 \\ 0, & \text{其他} \end{cases} \quad (20)$$

式中: Q 取值为 1; L_k 表示第 k 只蚂蚁对应的路径总长度.

3 算例分析

本文算例依托大连市若干外卖配送站点的实际案例设置.在实际运营中,各站点的商家点与客户点的数量、分布、距离、聚集度均不同,且每日波动性较大.基于该种情形,本文对几个不同的站点进行跟踪性的数据搜集,获取 10 日的跟踪数据,

并利用 MATLAB 软件编译本文所设计的蚁群算法. 每个算例求解 10 次, 取其最好解作为该算例的解.

3.1 不同站点的优化结果对比

不同站点对应的算例差异性较大, 如订单量不同、节点分布密度不同等. 本节将算法中的各参数固定, 对不同站点对应的算例分别进行求解, 从而发现站点的业务数据不同对优化的影响.

3.1.1 分散型城区站点

A 站点位于城市居民生活区域, 区域内客户节点较为分散, 聚集度低, 订单量少, 且每日订单变化较为明显. 根据调研, 对算例中各参数控制如表 1 所示, 求解结果如表 2 所示.

表 1 分散型城区站点算例的参数设置

Tab. 1 Parameter setting of decentralized station in urban areas

参数	数值	参数	数值
T_{max}	2	Q	1
α	1	VM	300
β	5	$ K $	29
ρ	0.1	m	80

表 2 分散型城区站点算例的求解结果

Tab. 2 Calculation results of decentralized station in urban areas

日期	订单量	现有效率	优化后效率	优化程度/%
1	276	5.08	5.62	10.63
2	303	5.12	5.67	10.81
3	287	5.12	5.63	10.07
4	284	5.12	5.64	10.04
5	271	5.04	5.59	10.85
6	292	5.1	5.61	9.9
7	299	5.11	5.67	11
8	301	5.1	5.57	9.13
9	284	5.11	5.64	10.38
10	288	5.11	5.68	11.26

3.1.2 聚集型商务区站点

B 站点位于聚集型商务区, 与 A 站点相比, 具有客户点呈现区域化的集中、订单量较多, 且订单变化相对较小的特点. 在参数控制上, 与表 1 相比, 骑手数量在该处设置为 37, 其他参数与表 1 一致, 求解结果如表 3 所示.

3.1.3 结果对比分析

A 站点的面积约 30 km², 日高峰期订单量约

表 3 密集型商务区站点算例的求解结果

Tab. 3 Calculation results of intensive business district station

日期	订单量	现有效率	优化后效率	优化程度/%
1	617	9.11	9.57	5.17
2	633	9.14	9.56	4.66
3	624	9.04	9.49	5
4	627	9.02	9.5	5.3
5	639	9.07	9.54	5.16
6	628	9.02	9.5	5.3
7	620	9.1	9.47	4.04
8	623	9.1	9.47	4.13
9	628	9.07	9.53	5.05
10	619	9.09	9.57	5.26

为 290 单. 由于配送区域大多为城市居民区, 订单数量较少, 顾客点位置的重合度较低, 目前, 该区域骑手的路径规划方式大多为结束一单后再指派新一单, 且路径大多在骑手主观意愿下构建而成, 因此配送效率较低. B 站点的面积约 24 km², 日高峰订单量至少为 600 单. 由于商务区午餐高峰订餐较为集中, 因此经常会有临近时间窗的顾客点重合的情况. 在该种情况下, 目前的路径规划与骑手主观意愿均倾向于多次取餐、一次送达的方式, 配送效率略高.

通过表 2 与表 3 的结果对比不难发现, 在分散型的场站中, 改进型的蚁群算法对现有方案的提升较大, 平均优化程度约为 10.41%; 聚集型场站订单较为集中, 经常存在同一商家点或客户点对应多个订单的情形, 平均优化程度较低, 约为 4.91%. 由此可见, 与现有的实际路径选择方案相比, 本文的蚁群算法对站点区域内的运输效率有一定的提升. 同时, 不同的站点类型影响着优化程度, 密集型站点订单数量多, 优化难度大, 因此优化程度较为有限, 而反观离散型场站, 由于订单数量少, 现有配送效率低, 因此优化程度较高.

3.2 算子有效性分析

为验证针对外卖问题提出的改进型邻域搜索算子的有效性, 本文从两个方面入手对其进行对比. 首先, 基于小规模实例, 运用 CPLEX 和本文的算法同时求解, 并对比两者之间值的差异和求解时间; 随后, 基于较大规模的实例, 运用没有改进算子的传统邻域搜索蚁群算法与本文算法对实例进行求解, 分析两者求解精度与速度; 最后, 根据

两次对比的实际情况对其进行分析,结合 O2O 外卖的实际情况提出意见。

3.2.1 小规模算例分析

由于 CPLEX 的求解能力有限,因此,本文选择在分散型站点实例中,根据时间窗,抽取 80 单,使订单在 2 h 高峰时段分布情况与原实例大致相同。骑手总人数 $|K|$ 调整为 8 人。同时由于 O2O 外卖受到实际生产作业中的限制,因此,本文限制 CPLEX 求解时间为 1.5 h。对每日小规模算例重复 10 次,取其有解的平均耗时为 CPLEX 求解所需时间;同时若无解的次数超过 4 次,则该日算例即判定为 CPLEX 失败无解。CPLEX 和算法的求解情况如表 4 所示。

表 4 小规模算例 CPLEX 与算法求解对比

Tab.4 Comparison of small-scale example CPLEX and algorithm solution

日期	CPLEX 效率	CPLEX 时间	算法效率	精度差/%
1	6.03	1.36	5.61	7.49
2	N/A	N/A	5.7	N/A
3	5.88	1.44	5.64	4.26
4	5.97	1.47	5.59	6.80
5	6.12	1.41	5.5	11.27
6	6	1.38	5.63	6.57
7	5.91	1.39	5.62	5.16
8	N/A	N/A	5.59	N/A
9	5.93	1.45	5.64	5.14
10	6.05	1.42	5.66	6.89

在算例求解过程中,CPLEX 时常会有无解的情况出现,甚至有 2 日的小规模算例无解超过 4 次而使该日算例被判为失败。同时,CPLEX 的平均求解时间大多趋近上限,而本文的算法求解时间大多在 10~20 min,其速度远高于 CPLEX 的求解速度。尽管在最终的优化效果上,CPLEX 的解精度平均高于算法解 6.7%,然而在小规模算例中,考虑到用时以及求解的成功率上,其误差是可以接受的。

3.2.2 大规模实例分析

为验证算子的有效性,本文基于离散型场站实例,同时采用有新算子的改进型蚁群算法与传统邻域搜索蚁群算法对问题进行求解。每个算例求解 10 次,取其最优解作为该算例的解。两种算法求解情况如表 5 所示。

表 5 分散型站点传统算法与改进算法求解对比

Tab.5 Comparison of traditional algorithm and improved algorithm for distributed site

日期	订单量	传统效率	改进效率	精度差/%
1	276	5.22	5.62	7.66
2	303	5.3	5.67	6.98
3	287	5.23	5.63	7.65
4	284	5.19	5.64	8.67
5	271	5.14	5.59	8.75
6	292	5.26	5.61	6.65
7	299	5.27	5.67	7.59
8	301	5.29	5.57	5.29
9	284	5.18	5.64	8.88
10	288	5.16	5.68	10.08

根据两种算法的求解对比结果,有新算子的改进型蚁群算法较传统邻域搜索蚁群算法在最优解的效率上平均高出 7.82%。同时,在实际求解过程中,有算子的蚁群算法的求解速度、收敛速度等算法表现方面上均远优于传统邻域搜索蚁群算法。

3.2.3 算子有效性分析

通过小规模算例与 CPLEX 进行对比、分散性站点与传统邻域搜索蚁群算法进行对比不难发现,由于算子针对 O2O 外卖的特殊性,大幅度减少了不合理的错误解的空间,提升了求解速度,因此在速度上要远优于其他两种方法。由于本文研究的外卖配送的高峰时间段为 2 h,且规模较大,同时与 CPLEX 的解误差较小,因此,CPLEX 求解在实际生产作业中并不适用。同时,由于传统邻域搜索蚁群算法受商家点与客户点的先后关系影响,常常出现无法规避先去客户点后去商家点的解、求解速度与收敛速度相对较慢等问题,因此,较传统的邻域搜索蚁群算法,改进型的蚁群算法在 O2O 外卖实际生产作业中具有更高效、更出色的表现。

3.3 算法参数的敏感性分析

为考察蚁群算法中各启发式参数对优化效果的影响程度,本节分别对蚂蚁数量 m 、信息素比重因子 α 、期望比重因子 β 和信息素挥发因子 ρ 的敏感性进行分析。相对而言,由于所设计的蚁群算法求解离散型站点算例的优化效果更佳,因此,本文以 A 站点第 9 日的数据作为算例,对 4 项参数的影响机理进行研究。

3.3.1 蚂蚁数量 m 对算法的影响

固定其他参数,对比算法在不同蚂蚁数量 m 下的求解结果.算法在每个 m 值下运行 20 次,对比平均的优化后效率.参数设置方面,蚂蚁数量 m 从 80 开始,以 5 为阶跃单位逐步增加至 125;其他参数设定与表 1 一致,求解结果如图 2 所示,当蚂蚁数 $m = 90$ 时,平均优化效率最高为 5.68.

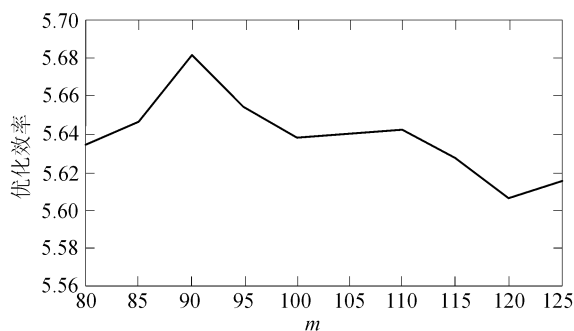


图 2 蚂蚁数量不同时算例的求解结果

Fig. 2 Calculation results with different numbers of ants

在蚁群算法中,每只蚂蚁没有先后顺序,皆为并行爬行,利用大量蚂蚁们随机生成的多个可行解来达到群体进化的目的.因此,蚂蚁的数量越多,其内部合作能力越高,从而提高整个算法的优化效果与稳定程度,所能解决的问题规模便越大;但若蚂蚁的数量过多,并行的随机性会使每条通过路径上的信息素的分布趋于分散化,使整个算法的收敛速度降低.因此,对于离散型的场站而言,由于订单量较少,可以选用较小的蚂蚁数量,在不影响稳定性与精度的前提下提高收敛速度;对于聚集型的场站而言,由于订单量较多,可以选择较大的蚂蚁数量,尽管收敛速度会降低,但在面对大规模问题时,可以扩大优化规模,增加系统稳定性.

3.3.2 信息素比重因子 α 对算法的影响

固定其他参数,对比算法在不同信息素比重因子 α 下的求解结果.算法在每个 α 值下运行 20 次,对比平均的优化后效率.参数设定方面, $m = 90$; α 值从 1 开始,以 1 为阶跃单位增加至 10;其他参数设置与表 1 一致,求解结果如图 3 所示,当 $\alpha = 5$ 时,平均优化效率最高为 5.69.

相对而言,信息素比重因子 α 对蚁群算法的影响较大.在蚁群算法的迭代过程中,蚂蚁的路径

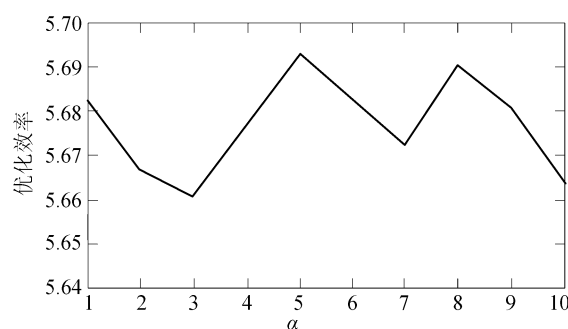


图 3 信息素比重因子不同时算例的求解结果

Fig. 3 Calculation results with different pheromone specific gravity factors

选择同时受到信息素与期望的双重影响.信息素比重因子 α 反映的是在这种混合的影响中,信息素的权重,即蚂蚁根据信息素的浓度,选择重复路径的可能性.信息素比重因子 α 值过低,会导致收敛速度较慢,且容易出现局部最优解;信息素比重因子 α 值过高,则会产生较强的正反馈作用,从而使整个蚁群算法收敛速度极度加剧,破坏算法稳定性,并过早收敛.因此,离散型场站和聚集型场站均需根据实际情况调整信息素比重因子 α 的值.

3.3.3 期望比重因子 β 对算法的影响

固定其他参数,对比在不同期望比重因子 β 下的求解结果.算法在每个 β 值下运行 20 次,对比平均的优化后效率.参数设定方面, β 值从 1 开始,以 1 为阶跃单位增加至 10; $\alpha = 5$; $m = 90$;其余参数设置与表 1 一致,求解结果如图 4 所示,当 $\beta = 4$ 时,平均优化效率最高为 5.70.

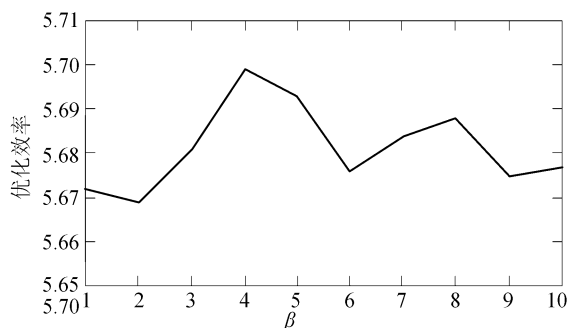


图 4 期望比重因子不同时算例的求解结果

Fig. 4 Calculation results with different expected specific gravity factors

期望比重因子 β 与信息素比重因子 α 类似,同样直接影响蚁群算法中蚂蚁的路径选择,反映

了在路径选择与搜索中,启发式搜索规则与确定性因素在路径选择规划中的影响比重.期望比重因子 β 过小会导致启发式规则的影响力较小,在大量重复搜索后会呈现均匀化、随机化的现象,导致无法寻找出最优解;期望比重因子 β 过大则会导致算法收敛极度加剧,过早收敛.因此,离散型场站和聚集型场站同样需要根据实际情况对期望比重因子 β 的值进行调整.

3.3.4 信息素挥发因子 ρ 对算法的影响

固定其他参数,对比算法在不同信息素挥发因子 ρ 下的求解结果.算法在每个 ρ 值下运行20次,对比平均的优化后效率.参数设定方面, ρ 的值从0.1开始,以0.1为阶跃单位增加至0.5; $\alpha=5$; $\beta=4$; $m=90$;其余参数设置与表1一致,求解结果如图5所示,当 $\rho=0.1$ 时平均优化效率最高为5.69.

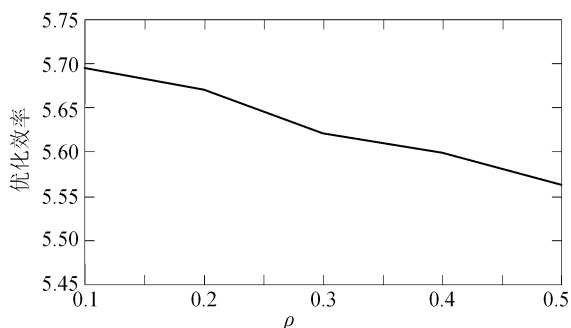


图5 信息素挥发因子不同时算例的求解结果

Fig.5 Calculation results with different pheromone volatile factors

在蚁群算法中,信息素代表着蚂蚁对已经搜索过的路径的记忆力,信息素挥发因子 ρ 代表搜索过的路径上信息素的挥发程度,而 $1-\rho$ 则代表搜索过的路径上信息素的残留程度.由于蚂蚁对重复路径的选择是基于信息素浓度的,因此,信息素挥发因子 ρ 直接影响蚁群算法的全局搜索能力和收敛速度.信息素挥发因子 ρ 过大时会使重复次数少的路径上信息素残留过少,被重复选择的可能性提高,从而使蚁群算法的随机性和全局搜索能力下降;同时,如果信息素挥发因子 ρ 过小,收敛速度则大大降低.对于聚集型场站而言,由于其问题规模较大,信息素挥发因子 ρ 的值应相对较小,从而提高全局搜索能力和算法随机性;对于离散型场站而言,由于问题规模较小,各节点呈离

散状分布,因此,应提高信息素挥发因子 ρ ,从而使算法的优化效率提升.

3.3.5 信息素释放总量 Q 对算法的影响

固定其他参数,对比算法在不同信息素释放总量 Q 下的求解结果.算法在每个 Q 值下运行20次,对比平均的优化后效率.参数设定方面, $Q=1,10,50,100,250,500,1000,2500,5000,10000$; $\alpha=5$; $\beta=4$; $\rho=1$; $m=90$;其余参数设置与表1一致,求解结果如图6所示,当 $Q=100$ 时,优化效率最大为5.72.

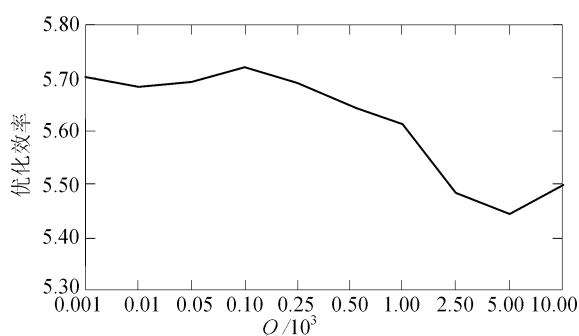


图6 信息素释放总量不同时算例的求解结果

Fig.6 Calculation results with different total release of pheromones

在蚁群算法中,信息素释放总量 Q 为循环一周时蚂蚁群体累计释放的信息素总量.作为算法中的反馈量,信息素释放总量 Q 越大,反馈作用越强,收敛速度越高,但同时也会导致搜索能力与可行解空间的下降,仅能得到局部最优解;反之,过小的信息素释放总量 Q 尽管会产生较为稳定的系统与较强的全局搜索能力,但可能使收敛速度过慢.因此,应根据实际情况调整信息素释放总量 Q 的值.

3.4 算例分析总结

由上文可见,上述参数累计对该算例优化度可提升至11.964%.蚂蚁数量 m 、信息素比重因子 α 、期望比重因子 β 、信息素挥发因子 ρ 和信息素释放总量 Q 这5个参数共同影响整个蚁群算法的全局搜索、收敛速度、求解规模等能力,且每种参数均有两极化的现象,不可一味地过大或过小.因此,根据实际情况,对于离散型的场站,应使蚂蚁数 m 较小,信息素挥发因子 ρ 较大,选用适宜的信息素比重因子 α 、期望比重因子 β 和信息素释

放总量 Q 来增加收敛速度. 对于聚集型场站, 可以通过增大蚂蚁数 m , 降低信息素挥发因子 ρ 来解决面临大规模问题时算法的求解能力不足、系统的稳定性与随机性较差的问题.

因此, 合理设置蚁群算法各参数可以提升路径优化程度. 同时, 离散型场站订单较为分散, 采用改进型蚁群算法可使路径优化度高达 10% 以上, 在聚集型场站也有 5% 的优化程度, 从而证明了针对 O2O 外卖问题设计的改进型蚁群算法的优势与可行性.

4 结 语

本文剖析了 O2O 外卖问题的特点, 相应地建立数学模型, 并开发改进型蚁群算法进行求解. 数值实验证明, 所得的骑手调度方案与当前实际方案以及传统求解方法相比具有明显改进, 证明了模型和算法的有效性, 并进一步分析了算法参数对算法性能的影响机理, 可为求解不同类型和规模的问题提供算法调整的决策支持.

参考文献 (References):

- [1] DANTZIG G B, RAMSER J H. The truck dispatching problem[J]. *Management Science*, 1959, 6(1): 80-91.
- [2] 吴腾宇, 陈嘉俊, 蹇洁. O2O 模式下的配送车辆实时取送货路径选择问题[J]. *系统工程理论与实践*, 2018, 38(11): 2885-2891.
WU T Y, CHEN J J, JIAN J. The online pick-up and delivery vehicle routing problem under O2O delivery[J]. *System Engineer—Theory & Practice*, 2018, 38(11): 2885-2891. (in Chinese)
- [3] 张庆华, 吕小丹. 电商退换货车辆路径问题及蚁群算法研究[J]. *计算机工程与应用*, 2018, 54(22): 239-245.
ZHANG Q H, LV X D. Research on vehicle routing problem with return and replacement in e-commerce environment and its solution to ant colony algorithm[J]. *Computer Engineering and Applications*, 2018, 54(22): 239-245. (in Chinese)
- [4] MARILÈNE C, GUY D, STEFAN I, et al. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks[J]. *European Journal of Operational Research*, 2015, 250(1): 782-793.
- [5] CHÁVEZ M A C, MARTINEZ-OROPEZA A. Feasible initial population with genetic diversity for a population-based algorithm applied to the vehicle routing problem with time windows[J]. *Mathematical Problems in Engineering*, 2016(5): 1-11.
- [6] INMACULADA R M, JUAN-JOSÉ S G, HANDE Y. The periodic vehicle routing problem with driver consistency[J]. *European Journal of Operational Research*, 2018(273): 575-584.
- [7] ARMANDO H P, SEBASTIÁN U. Formulations and algorithms for the pickup and delivery traveling salesman problem with multiple stacks[J]. *Computers and Operations Research*, 2018, 93(1): 1-14.
- [8] 王帅, 赵来军, 胡青蜜. 随机旅行时间的外卖 O2O 配送车辆路径问题[J]. *物流科技*, 2017, 40(1): 93-101.
WANG S, ZHAO L J, HU Q M. Vehicle routing problem with O2O takeout delivery based on stochastic travel times[J]. *Logistics Sci-Tech*, 2017, 40(1): 93-101. (in Chinese)
- [9] 陈萍, 李航. 基于时间满意度的 O2O 外卖配送路径优化问题研究[J]. *中国管理科学*, 2016, S1(24): 171-176.
CHEN P, LI H. Optimization model and algorithm based on time satisfaction for O2O food delivery. [J]. *Chinese Journal of Management Science*, 2016, S1(24): 171-176. (in Chinese)
- [10] 陈希琼, 胡大伟, 杨倩倩, 等. 多目标同时取送货车车辆路径问题的改进蚁群算法[J]. *控制理论与应用*, 2018, 35(9): 1347-1356.
CHEN X Q, HU D W, YANG Q Q, et al. An improved ant colony algorithm for multi-objective vehicle routing problem with simultaneous pickup and delivery[J]. *Control Theory & Applications*, 2018, 35(9): 1347-1356. (in Chinese)
- [11] 柴获, 何瑞春, 苏江省, 等. 求解双目标带时间窗车辆路径问题的蚁群算法[J]. *交通运输系统工程与信息*, 2018, 18(4): 156-162.
CHAI H, HE R X, SU J S, et al. An ant colony optimization for the Bi-objective vehicle routing problem with time windows on multigraph[J]. *Journal of Transportation Systems Engineering and Information Technology*, 2018, 18(4): 156-162. (in Chinese)
- [12] 裴振兵, 陈雪波. 改进蚁群算法及在车辆运输调度中的应用[J]. *信息与控制*, 2015, 44(6): 753-758.
PEI Z B, CHEN X B. Improved ant colony algorithm and its application to vehicle routing and scheduling[J]. *Information and Control*, 2015, 44(6): 753-758. (in Chinese)
- [13] 杨鹏, 邹浩, 徐贤浩. 带时间窗集送货需求可分车辆路径问题的改进蚁群算法[J]. *系统工程*, 2015, 33(9): 58-62.
YANG P, ZOU H, XU X H. Improved ant colony algorithm for vehicle routing problem with time windows and split pickups and deliveries[J]. *Systems Engineering*, 2015, 33(9): 58-62. (in Chinese)
- [14] SCHYNS M. An ant colony system for responsive dynamic vehicle routing[J]. *European Journal of Operational Research*, 2015, 245(3): 704-718.
- [15] CAPOROSSI G, HANSEN P, MLADENOV N. Variable Neighborhood Search[M]. [S. l.]: Elsevier Science Ltd, 1997.