|  |  |
| --- | --- |
| 队伍编号 | MCB2201536 |
| 赛道 | A |

**题目（待定）**

**摘 要**

~~“58到家”上门家政服务平台能够向用户提供家政保洁、搬家、维修等生活服务，每天都有大量的订单通过平台得到分配。平台在分配订单时，须综合考虑服务质量和服务效率，尽可能在分配服务分高的阿姨的同时，缩减阿姨相邻单的通行时间，帮助阿姨提高接单量。本文建立订单分配模型，研究并优化当前的系统分配算法，提升算法的求解能力，为改善顾客体验、节省阿姨时间出一份绵薄之力。~~

~~针对问题一，主要需要确定“服务开始时间”对后续的订单分配以及目标值的影响，对算法进行优化。本文建立了订单分配模型，通过划分出不同区域的方式，定时将订单分区域处理，并运用求解器比对不同分配方案的目标值，选出使用当前区域划分形式时的最佳分配方案。接着，改变区域划分形式，运用求解器解出~~（待定）

**关键词：动态规划、混合整数规划、（待定）**

**目 录**（待更新）

[一、问题重述 1](#_Toc124708211)

[1.1问题背景 1](#_Toc124708212)

[1.2问题重述 1](#_Toc124708213)

[二、问题分析 2](#_Toc124708214)

[三、模型假设 3](#_Toc124708215)

[四、符号说明及数据集介绍 3](#_Toc124708216)

[五、问题一模型的建立与求解 5](#_Toc124708217)

[5.1模型准备与分析 5](#_Toc124708218)

[5.2模型建立 7](#_Toc124708219)

[5.2.1订单分配模型 7](#_Toc124708220)

[5.2.2求解器模型 9](#_Toc124708221)

[5.2.3混合整数规划模型 14](#_Toc124708222)

[5.3模型求解 16](#_Toc124708223)

[六、问题二模型的建立与求解 20](#_Toc124708224)

[6.1模型准备与分析 20](#_Toc124708225)

[6.2模型建立 20](#_Toc124708226)

[6.2.1求解器模型 21](#_Toc124708227)

[6.3模型求解 21](#_Toc124708228)

[七、模型的评价与推广 23](#_Toc124708229)

[7.1模型优点 23](#_Toc124708230)

[7.2模型缺点 23](#_Toc124708231)

[7.3模型推广 23](#_Toc124708232)

[八、参考文献 23](#_Toc124708233)

# 一、问题重述

1.1问题背景

“58到家”上门家政服务平台能够向用户提供家政保洁、搬家、维修等生活服务。用户在平台下单家政保洁服务后，平台会将订单分配给一位保洁阿姨，阿姨接单后会在用户指定的服务时间上门，提供保洁服务。

平台在分配订单时，会遵循两条原则。首先，应考虑服务质量。根据每位阿姨历史订单的评价，平台会计算出阿姨的服务分。服务分取值为[0,1]，值越大代表阿姨的服务水平越好、用户的评价越高。为了提升用户体验，平台会尽量分配服务分较高的阿姨。其次，应考虑服务效率。为了节省阿姨的时间、帮助阿姨提高接单量，平台在分配时需尽量缩减阿姨相邻单的通行时间。

每天都有大量的订单通过平台得到分配。为了改善算法的分配方案、提升算法的运行效率、改善用户的体验、节省保洁阿姨的时间，需要对当前平台的订单分配算法进行研究和优化。

1.2问题重述

基于上述背景，本文根据题目所给的约束条件及假设，建立数学模型、设计相关算法进行研究、分析和优化，解决问题如下：

**问题一：离线批量派单模式下，优化订单分配算法**

离线批量派单模式下，根据附件中给出的所有订单信息和阿姨信息，设计最优的订单分配算法，将所有的订单进行分配，并将求解结果导出。此外，再对附件中前50个订单和前20位阿姨运行优化算法，给出阿姨的执行任务列表，并画出阿姨的行动轨迹图。

**问题二：线上批量派单模式下，设计考虑压单的分配算法**

线上批量派单模式下，通常采用固定的频率派单，每30分钟将该时间段内产生的新订单统一分配。分配时允许部分订单暂时不派单，称为压单。压单订单须满足服务开始时间的最早时间比当前时间晚2小时以上。在考虑压单的情况下，设计最优的订单分配算法，对所有订单进行分配，并将最终决策结果和每次决策结果导出。

# 二、问题分析

问题一需要我们在离线批量派单模式下，设计一个订单分配算法，对附件中的所有订单和阿姨进行分配，是一个动态规划（Dynamic Programming）以及混合整数规划（Mixed Integer Programming）问题；问题二则需要我们在线上批量派单模式下，设计一个考虑压单的订单分配算法，以固定的频率分配新产生的订单。考虑到问题一与问题二的算法框架大致相同，我们仅需针对相应的模块和功能进行设计和修改，在此，我们将问题一与问题二合并为整体进行分析。

综合考虑后，本文决定采用Python中的cvzpy、pandas、numpy、matplotlib等库来处理数据、建立模型、绘制图像。本文首先对数据进行预处理，引入约束条件，将状态初始化。因阿姨及订单数量均超过2000，为了减少变量个数、降低算法对算力的需求、提高运算效率，本文尝试按照时间戳针对每一时刻划分网格，将订单和阿姨按照坐标划分至不同区域，对各区域的订单和阿姨进行独立分配求解。为了进一步提升算法效率、获得全局最优分配方案，本文尝试更换网格划分形式、设置订单优先级、采用“优质阿姨”、“未来订单”、“未来阿姨”策略，使用cvzpy库中适用的GLPK\_MI求解器对各网格进行独立递归求解，不断优化目标值，选出当前最优解。递归完成后，更新阿姨、订单的状态量，进入下一时刻，重复上述过程。待到当前时刻超出订单分配时间区间后，输出全局最优解，获得最优分配方案。下面给出了算法的总流程图。

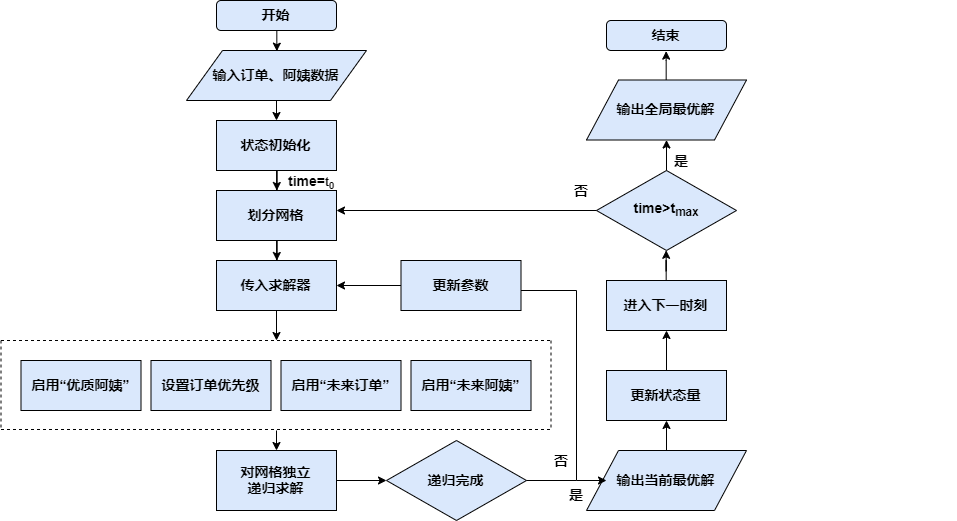


图1 优化算法总流程图

# 三、模型假设

1.假设在离线批量派单模式下，阿姨当天的第一单可以不考虑通行时间的约束，准时到达任意订单的位置；在线上批量派单模式下，阿姨当天第一单需考虑通行时间约束；

2.假设在线上批量派单模式下，平台将订单分配给阿姨的那一刻阿姨立刻前往所接订单处，中途没有时间损耗；

3.假设平台分配订单后会将阿姨的“服务开始时间”反馈给顾客，“服务开始时间”位于服务时间区间内且“服务开始时间”越早，顾客的满意度越高；

4.假设划分网格后，阿姨只能在当前网格内活动，不同网格内的订单与阿姨彼此间不存在干扰。

# 四、符号说明及数据集介绍

**表1 符号和变量说明表**

|  |  |
| --- | --- |
| 符号 | 符号说明 |
|  | 当前时刻 |
|  | 阿姨数据集 |
|  | 订单数据集 |
|  | 网格参数，表示将地图划分为行列的区域 |

**表2 阿姨数据集**

|  |  |  |
| --- | --- | --- |
| 参量 | 类型 | 参量说明 |
|  | List | 包含阿姨的所有参量 |
|  | Integer | 阿姨的唯一标识 |
|  | Float | 阿姨的服务分 |
|  | Integer | 阿姨当前位置的横坐标，默认值为初始点横坐标 |
|  | Integer | 阿姨当前位置的纵坐标，默认值为初始点纵坐标 |
|  | Float | 阿姨的状态下一次变更为“可派遣”的时间点 |
|  |  | 阿姨的服务状态，可派遣时置0，反之置1 |
|  | List | 包含阿姨当天接取的所有订单编号，默认值为空列表 |
|  | List | 包含阿姨当天所有的接单时间，默认值为空列表 |
|  |  | 阿姨当天未接单时置1，接取第一单后置0 |

**表3 订单数据集**

|  |  |  |
| --- | --- | --- |
| 参量 | 类型 | 参量说明 |
|  | List | 包含订单的所有参量 |
|  | Integer | 订单的唯一标识 |
|  | Integer | 订单的下单时间 |
|  | Integer | 订单服务地点的横坐标 |
|  | Integer | 订单服务地点的纵坐标 |
|  | Integer | 服务时间区间的最早时间 |
|  | Integer | 服务时间区间的最晚时间 |
|  | Integer | 服务时长 |
|  |  | 订单的开启状态，订单开启时置1，反之置0 |
|  | Integer | 订单分配的阿姨编号 |
|  | Float | 订单的服务开始时间 |
|  |  | 订单的保留状态，成功被分配后置0，反之置1 |
|  | Float | 当前时刻距离订单最晚服务时间的小时数 |
|  |  | 订单压单参数，当前时刻可压单置1，反之置0 |

# 五、问题一模型的建立与求解

5.1模型准备与分析

本文首先对订单和阿姨的数据进行预处理，将状态初始化。将所有订单中最小值设定为订单分配初始时刻，并将订单中除服务时长外所有时间参量减去，将单位转换为小时：









获得一条从0时刻开始的新时间轴。计算得知，所有订单中最晚服务开始时间为：



确定时间轴范围为：



对所有阿姨、订单的位置坐标进行处理，将单位转换为千米，并求出坐标范围。设、、、表示横、纵坐标最小最大值。

，

，

，

，

针对每份订单引入新参数——开启状态。订单的状态默认值为“关闭”。若当前时刻处于订单的服务时间区间内，则订单状态转为“开启”，并将订单加入到求解器中。

针对每位阿姨引入新参数——派遣状态。阿姨的状态默认值为“可派遣”。当阿姨被分配订单后，状态会从“可派遣”转变为“工作中”，同时，阿姨的坐标会更新为订单的坐标。

针对每位阿姨引入新参数——空闲时刻。空闲时刻表示阿姨下一次状态由“工作中”转变为“可派遣”的时间点，默认值为0。当时间戳到达空闲时刻的值以后，阿姨的状态会从“工作中”转变为“可派遣”，标志着阿姨已完成该订单，可以分配下一个订单。

针对每位阿姨引入新参数——是否首次接单。若阿姨是首次接单，则当前订单目标值中的“服务间隔时间”赋予定值0.5。最后，对所有参量进行初始化。

设置半点的整数倍时刻为求解时间点。分析得知，所有订单的开始时间、最早服务时间和最晚服务时间均为半点的整数倍，因此，阿姨开始服务的时间和结束服务的时间也均为半点的整数倍，订单和阿姨的状态都只会在半点的整数倍时刻进行更新。而在离线批量派单模式下，阿姨当天的第一份订单可以不考虑通行时间的约束，准时到达任意订单的位置。因此，在分配订单时，我们只需在半点的整数倍时刻更新订单和阿姨的状态，对未分配的订单和未接单的阿姨实施匹配。

对订单数据进行统计，绘制订单分布三维插值图和热力图。

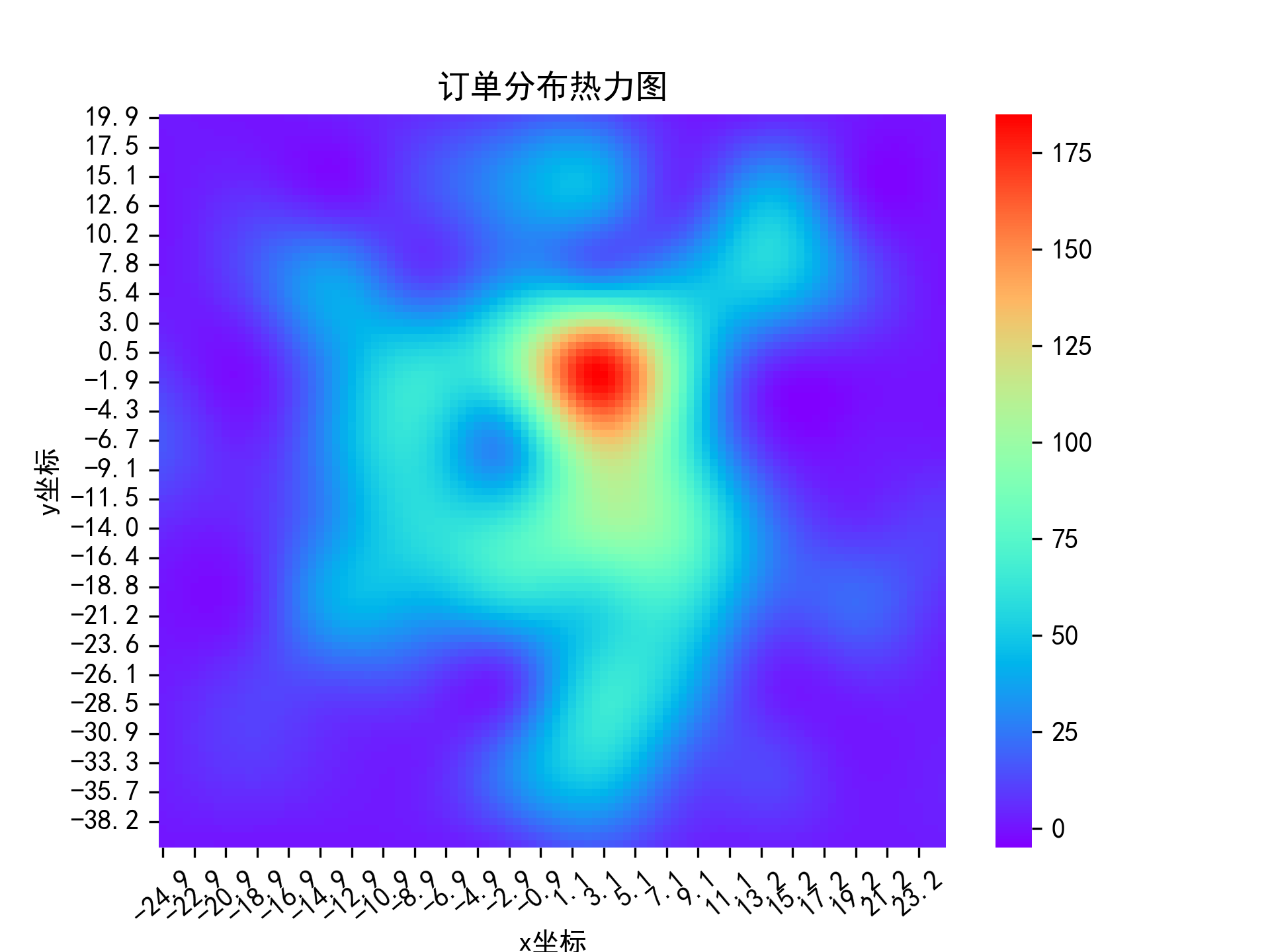
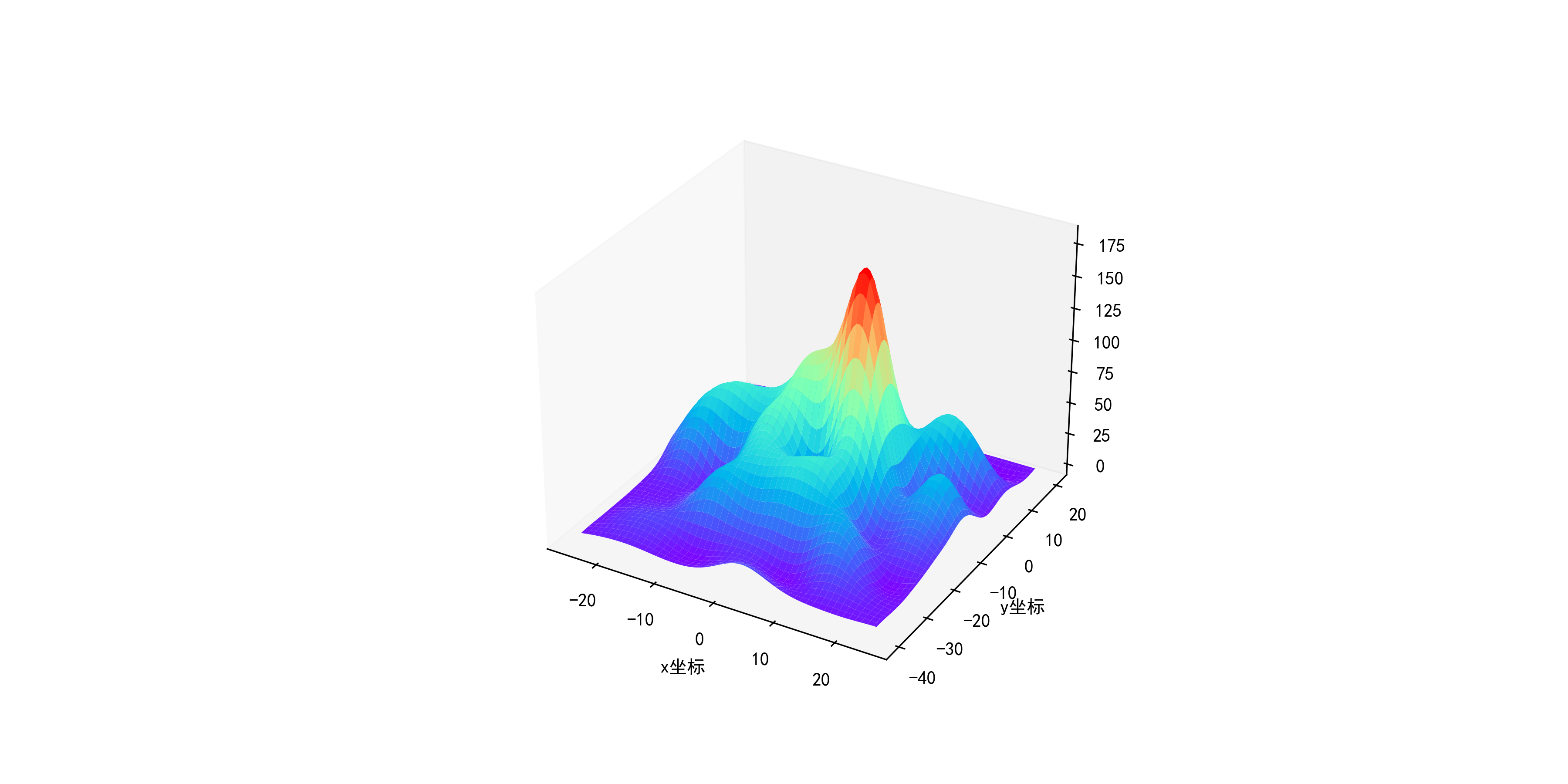


图2 订单分布三维插值图、热力图

图2展示订单的分布状况和密集程度。三维插值图中，坐标表示订单服务地点横坐标，单位千米；坐标表示订单服务地点纵坐标，单位千米；坐标表示当前地点区域订单数量，单位份。针对订单的密集程度，三维插值图和热力图以特殊高亮的形式显示订单密集的区域。

对热力图像进行处理，绘制订单分布等值线图。从图中我们可以更为具体地知晓订单的位置分布密集程度。图中仅有一个区域订单较为密集，数量达到175份，考虑到订单具有不同服务时间，且阿姨共有2795位，这个数量远大于某时刻待分配的订单数，可以认为一般情况下所有订单都能得到分配，即存在可行解。

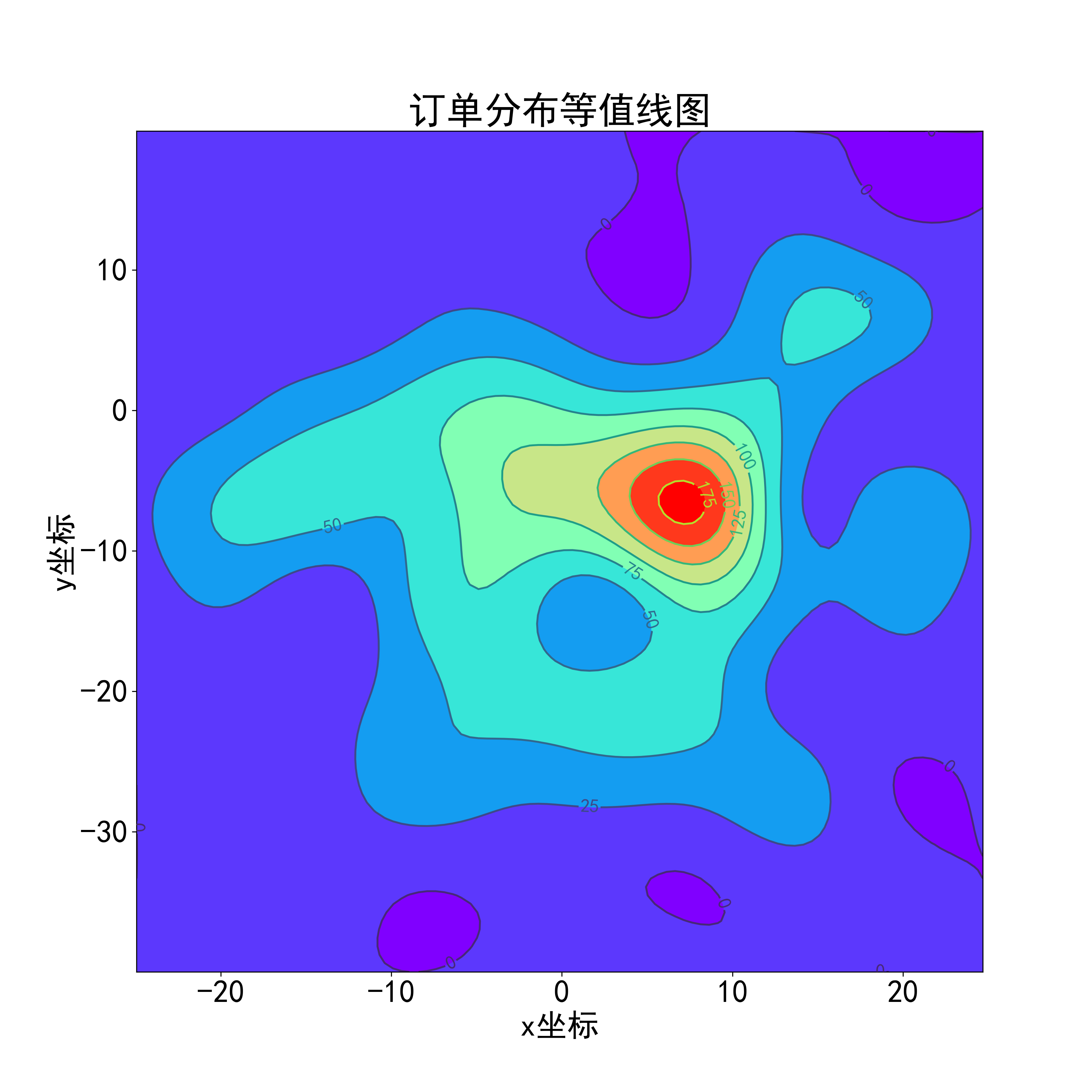


图3 订单分布等值线图

5.2模型建立

5.2.1订单分配模型

定义距离判断矩阵，判断阿姨能否及时到达订单位置。因阿姨的速度为15千米/小时，可能存在阿姨当前距离订单过远，无法在服务时间区间以内到达订单位置的情况。对此，我们定义距离判断矩阵，以此对订单的分配施加约束。



在离线批量派单模式下，订单分配时刻与阿姨行动时刻可以并不相同。即：本文设计的算法在时刻给阿姨分配了订单，但阿姨可在时刻前就开始前往订单处，等待服务开始。针对阿姨当天接取的第一单，阿姨可以很早从初始位置出发，前往订单所在坐标，因此，每位阿姨的首单不受距离判断矩阵的限制；而对于阿姨首单之后的订单，由于阿姨均在半点的整数倍时刻开始服务、结束服务，状态也仅在此时发生变动，对当前算法没有影响。因此，本文对模型作出假设：在离线批量派单模式下，阿姨当天的第一单可以不考虑通行时间的约束，准时到达任意订单的位置。

为了提升运行效率，采用“网格划分”方案。若每次求解时，都对当前时刻所有开启的订单和可派遣的阿姨运行算法，变量数会达到一个惊人的值，致使算法运行时间大大延长。为了提升运行效率，缩减运算时间，本文设计并采用“网格划分”策略。

**方案一：网格划分，独立求解**

本文将地图划分为行列共个区域，以表示第行第列的网格。、越大，则划分的网格越多，每个区域的面积越小。我们对模型作出假设，各区域中的阿姨只能在该区域内移动、接取订单。网格划分完成后，本文针对每一个网格进行独立的订单分配求解。如此，单次求解时阿姨和订单的数量将显著下降，算法运行速度得到大幅提升。下图给出了网格划分策略的伪代码。

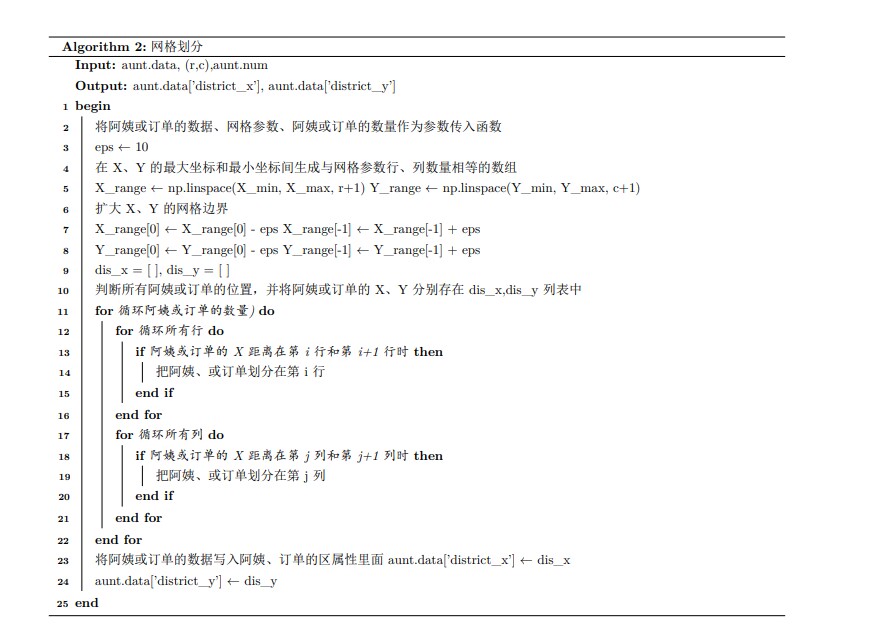


图4 网格划分伪代码

对阿姨初始坐标和订单位置坐标进行统计分析，得知订单和阿姨的初始位置重合度很高，利于我们划分网格，寻找可行解。本文尝试绘制、网格分布图如下。

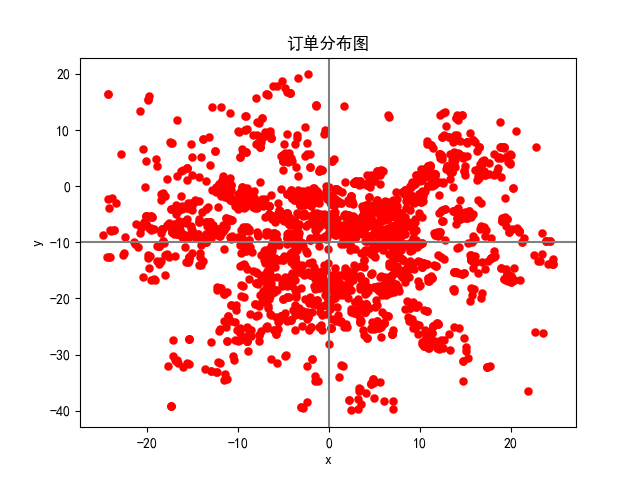
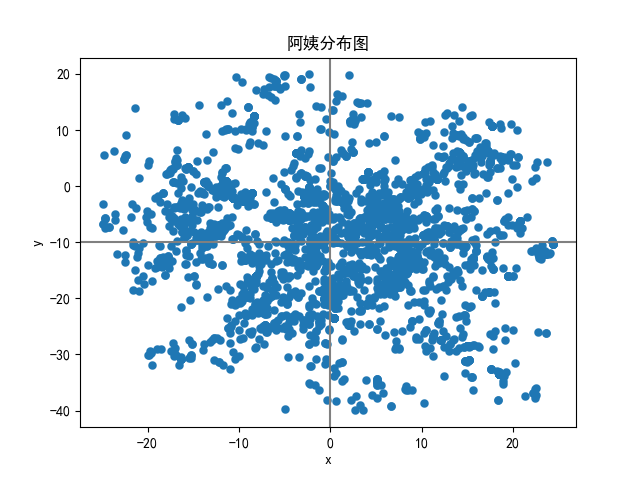


图5 2\*2网格划分形式下阿姨、订单分布图

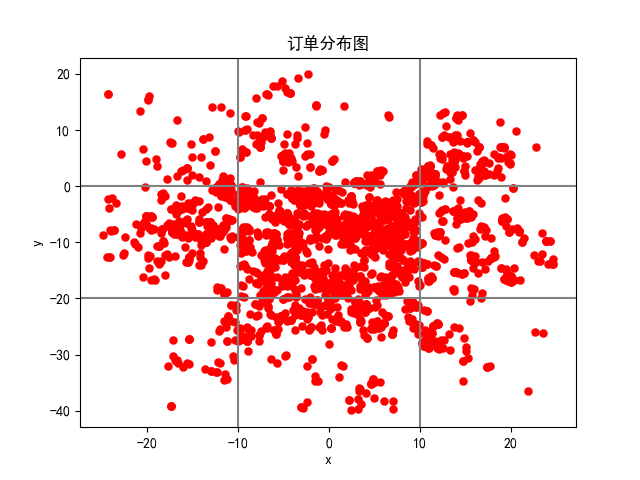
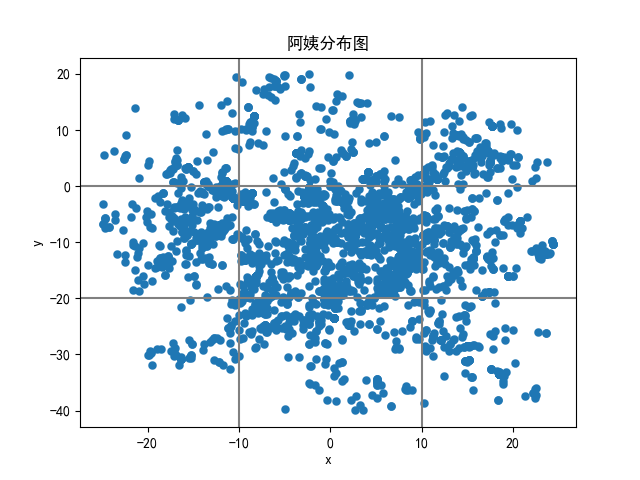


图6 3\*3网格划分形式下阿姨、订单分布图

5.2.2求解器模型

为了进一步降低运算量，本文设计并采用“优质阿姨”策略。

**方案二：设置“优质阿姨”**

若当前网格内阿姨数远大于订单数，按照服务分由高至低对阿姨排序，选取前数位阿姨定义为“优质阿姨”。设为优质阿姨数量，表示第行第列的网格，为时刻网格订单数，为时刻网格阿姨数。启用“优质阿姨”方案需当前网格满足：



若满足该条件，设置优质阿姨数1.5倍于订单数，即：



设置优质阿姨向量。



使用sort函数对当前区域阿姨按照降序排列，取前位阿姨定义为“优质阿姨”。

在此解释设置“优质阿姨”的原理。在对目标值的优化过程中，目标值受阿姨服务分、通行距离和服务间隔时间影响。采用网格划分策略后，阿姨仅能接取自身所处区域的订单，这一约束使得阿姨的通行范围缩小，将通行距离限制在网格对角线距离内，通行距离对目标值的影响减少；服务间隔时间则存在小范围波动，无法准确量化；在此情况下，影响目标值的主要可控因素即为阿姨的服务分。因此，若当前网格内阿姨数量远大于订单数量，易知订单必定能够全部得到分配。此时，仅需考虑让服务分较高的阿姨接取订单，即可在优化目标值的同时，显著减少阿姨相关的变量数。启用该方案求解时，仅将当前区域优质阿姨和订单取出，进行独立求解。

上述“网格划分”和“优质阿姨”方案会在一定程度上削减阿姨的移动范围，限制阿姨的行动，剪去部分分支。但综合考虑目标值优化和运算效率，“网格划分”和“优质阿姨”策略可以在不使目标值下降过多的情况下显著减少运算时间，依旧是十分优秀、行之有效的方案。

网格划分完成后，可能会出现某时刻某区域未找到可行解的情况，下面对该现象进行分析。

设为时刻网格订单数，为时刻网格阿姨数。若满足，且每份订单考虑距离判断矩阵约束后至少能够匹配一位不冲突的阿姨，显然订单可以全部分配出去，存在可行解。若，显然，这种情况下订单不可能全部分配出去。针对该状况，本文设计并采用“网格扩大化”和“设置订单优先级”方案。

**方案三：网格扩大化**

根据上文中的分布图易知，绝大部分区域中阿姨数都大于甚至远大于订单数。

若某时刻某区域在当前情况下不存在可行解，可将区域扩大，缩减网格数量。区域扩大后，重新对订单和阿姨进行划分，再次计算当前网格的阿姨数和订单数。若依旧存在阿姨数小于订单数的情况，则继续迭代、扩大区域，直至可求出可行解或网格划分缩减为为止。下图给出“网格扩大化”方案的伪代码。

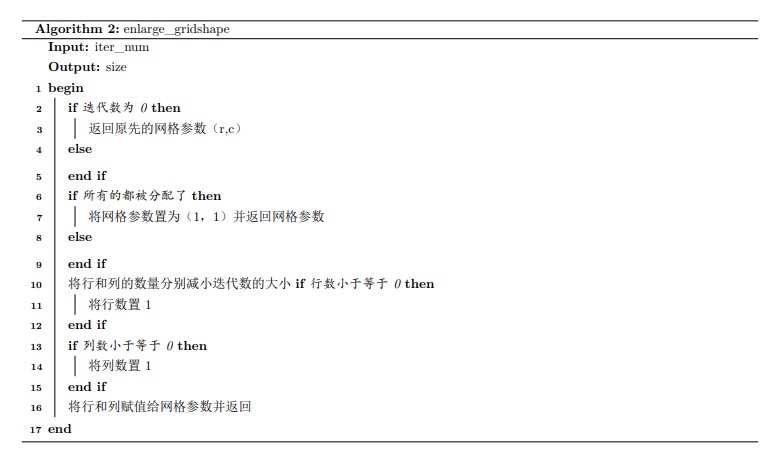


图7 网格扩大化伪代码

本文先后尝试三种网格扩大化迭代方式，最终选择效果最佳的迭代方式三。设为迭代次数，为当前网格参数，其中，为网格行数，为网格列数，为初始网格参数。三种迭代方式如下：

，，

三种迭代方式最终都会收束至网格。

使用迭代方式一时，随着迭代次数的增加，网格扩大化的速度也在剧烈增大，不易控制且后期容易剪去过多分支；相较于方式一，方式二网格扩大化的速度在后期有所收敛，但前期增速更快，依旧不易控制。

本文最终敲定具有鲁棒性，总能在不使目标值下降过多的情况下大大提升运行效率的迭代方式三。迭代方式三严格控制了网格扩大化的速度，使网格每次迭代仅减少一行、一列。这虽然增加了迭代次数与运算时间，但可以显著增加求得可行解的概率，保证算法的正常运行，是综合效果最好的迭代方式。

**方案四：设置订单优先级**

订单存在服务时间区间，并非必须在最早服务时间分配阿姨。因此，若当前时刻某区域中阿姨数小于订单数，除了即将超过服务时间区间的订单外，其它订单均可保留至下一次分配时刻。

本文为求解器引入一个新参数——订单优先级。设为当前时刻距离订单最晚服务时间的小时数。即：



本文使用sort函数按照值对订单升序排列并分类，定义“最优先订单”、“次优先订单”等多类订单，划分不同的优先级。最优先订单的优先级为1，次优先订单的优先级为2，以此类推。其中，值为0的订单会在状态开启的第一时间被纳入待分配订单，在阿姨资源允许的条件下被优先分配。当阿姨数小于订单数时，只要每份紧急订单至少能够匹配一位不冲突的阿姨，即可继续进行求解。若不满足每份紧急订单至少能够匹配一位不冲突的阿姨，则可以利用网格扩大化来重新划分区域，直至找出可行解。

**递归求解**

本文利用GLPK\_MI求解器针对每个时刻每个网格独立递归求解混合整数规划模型。

求解时，首先针对当前订单的紧急程度排序并划分优先级。设置求解器参数，表示此次求解时将优先级小于等于该参数的订单定义为“必须分配订单”，当前时刻必须分配出去。默认值为1。

本文先尝试递归求解当前时刻将所有订单全部分配出去的方案。将求解器初始化并放入订单和阿姨，计算时的分配方案，即最低限度的分配方案。

1. 若解得的分配方案不满足约束条件或订单无法全部分配出去，说明继续分配订单仍不能找到可行解，考虑将部分订单保留至下一时刻，转为求解将订单尽可能分配并使目标函数尽可能大的方案，返回解矩阵作为临时解；
2. 若求解将订单尽可能分配时，依旧未找到可行解，返回，将0矩阵作为临时解；
3. 若方案满足约束条件，则判断其为可行解，返回解矩阵作为临时解，并将的值增加1，即将下一优先级的订单加入“必须分配订单”列表中，再重新进行求解，直至所有订单均被定义为“必须分配订单”后，停止求解；
4. 若时，方案不满足约束条件或订单无法全部分配出去，则舍弃该方案，返回，将0矩阵作为临时解。

综上，本求解器每次求解必定能输出解矩阵。存在可行解时解矩阵为可行解矩阵，不存在可行解时解矩阵为0矩阵。

每次递归获得新的返回值后，与上一次获得的返回值进行比对，选出目标函数更大的解矩阵作为当前解。经过数次递归，获得的当前解即为该时刻该网格划分方案下的最优解。下图给出了递归求解伪代码。

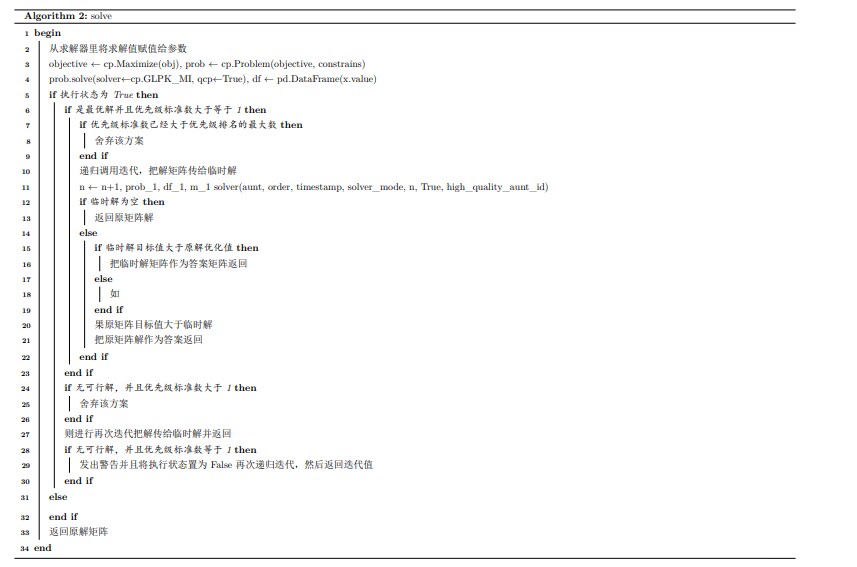


图8 递归求解伪代码

求解完成后，对各订单和阿姨的状态量进行更新，并进入下一时刻，重新划分网格求解，直至所有订单均被分配，输出各时刻的最优解，即为全局最优解。

此外，不同网格划分形式下求得的全局最优解不同，目标函数值也有差异。为了求解出最优方案，需针对各网格划分形式下的全局最优局进行比对，选取目标函数最大的方案作为最优分配方案。

5.2.3混合整数规划模型

**（1）约束条件**

设时刻当前求解区域有份订单，位阿姨。

设置时刻订单分配矩阵。



设置时刻距离判断矩阵。



设置时刻优质阿姨向量。



由于当前待分配的订单受到阿姨必须在服务时间区间内到达的限制，可以得到约束条件①：



由于每份订单不可被分配给一位以上的阿姨，且仅有紧急订单必须被分配，非紧急订单可以保留至下一分配时刻，可以得到约束条件②：



由于当前时刻每位阿姨最多接取一份订单，可以得到约束条件③：



由于算法拥有使用“优质阿姨”策略求解的功能，当不启用优质阿姨策略时，矩阵元默认值为1；启用优质阿姨时，重新计算矩阵元。可以得到约束条件④：



**（2）目标函数**

根据题意，我们求解的订单分配方案应满足：订单匹配的阿姨服务分平均值尽量大；每单的平均通行距离尽量小；阿姨平均服务间隔时间尽量小。

设当前时刻为，当前网格参数为，表示第行第列的网格，时刻区域有份订单。设为订单分配的阿姨服务分，为阿姨的通行距离，为阿姨的服务间隔时间。则目标函数为：









**（3）混合整数规划模型**





5.3模型求解

上述模型建立完成后，本算法针对模型进行求解。

输入订单、阿姨数据后，对二者的状态进行初始化，完成数据预处理。针对当前时刻选择网格参数，划分网格，并依次将各网格中开启订单和可派遣阿姨传入求解器中，设置“优质阿姨”和订单优先级相关参数，进行独立求解并更新参数，完成递归求解。输出当前最优解，并更新订单、阿姨的状态量，进入下一时刻，重新划分网格求解，直至所有订单分配完成，获得最优解。最后，比对各网格划分形式最优解，选取并输出全局最优解、目标值和算法运行时间。

下表给出了不同参数下算法最优解的目标值与运行时间。

**表4 问题一第一问求解结果表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 求解方案 | 启用优质阿姨 | 初始网格参数 | 运行时间 | 目标函数值 |
| 1 | 否 | （1，1） | 1时47分58秒 | **0.587064** |
| 2 | 否 | （2，2） | 1时02分18秒 | 0.584485 |
| 3 | 否 | （3，3） | 0时34分31秒 | 0.583332 |
| 4 | 否 | （4，4） | 0时15分17秒 | 0.580127 |
| 5 | 否 | （5，5） | 0时07分41秒 | 0.576018 |
| 6 | 是 | （1，1） | 1时17分12秒 | 0.530761 |
| 7 | 是 | （2，2） | 0时43分02秒 | 0.526610 |
| 8 | 是 | （3，3） | 0时22分42秒 | 0.515351 |
| 9 | 是 | （4，4） | 0时10分15秒 | 0.523255 |
| 10 | 是 | （5，5） | 0时06分09秒 | 0.524255 |

算法求得全局最优分配方案的目标值为0.587064。

由上表分析可知，目标函数并非随初始网格参数扩大的单调递减函数。本问题为动态规划问题，当前时刻各方案求出的局部最优解均会对下一时刻的分配产生不确定的影响，令目标函数在每次求解时处于微小范围的波动中，并非单调函数。

分析可知，采用网格划分策略、启用优质阿姨均能显著减少算法运行时间，网格数越多，运行时间越短。划分网格、使用优质阿姨会在一定程度上削减阿姨的移动范围，限制阿姨的行动，剪去部分分支，在大幅减少计算量的同时，会造成最优解目标函数值略微缩小。

相较于启用优质阿姨，划分网格策略对目标函数值的影响更小。未采用优质阿姨时，初始网格参数为的方案相比的方案目标值仅下降了，而运行时间却缩减了；初始网格参数为的方案，启用优质阿姨后运行时间减少，目标值下降了。总体而言，网格划分策略对目标函数值的影响较小，且能有效提升运行效率；优质阿姨方案对目标函数值有一定的影响，缩减运行时间的效果有限。综合来看，网格划分更优，但两个方案并不冲突，实际运用时可配合参考使用。

虽然最后本文求解全局最优解时网格划分形式为，即实际上并未采用网格划分方案，也未启用优质阿姨，但该策略能够有效降低变量数、显著节省算法运行时间、大幅减少计算量。在现实生活中，分配订单时既要考虑对目标值的优化，也要考虑分配效率，故可选择性使用网格划分功能和优质阿姨功能。

本文基于已设计的算法，对附件中前50份订单和前20位阿姨进行匹配，解得最优分配方案目标值为0.434847，并给出阿姨的执行任务列表如下：

**表5 问题一第二问求解结果表**

|  |  |  |
| --- | --- | --- |
| 订单编号 | 服务开始时间 | 阿姨编号 |
| 0 | 1662787800 | 6 |
| 1 | 1662769800 | 17 |
| 2 | 1662773400 | 7 |
| 3 | 1662791400 | 2 |
| 4 | 1662802200 | 2 |
| 5 | 1662777000 | 1 |
| 6 | 1662773400 | 16 |
| 7 | 1662769800 | 19 |
| 8 | 1662773400 | 8 |
| 9 | 1662777000 | 10 |
| 10 | 1662795000 | 5 |
| 11 | 1662769800 | 14 |
| 12 | 1662773400 | 12 |
| 13 | 1662773400 | 4 |
| 14 | 1662784200 | 4 |
| 15 | 1662769800 | 10 |
| 16 | 1662798600 | 10 |
| 17 | 1662769800 | 5 |
| 18 | 1662791400 | 5 |
| 19 | 1662802200 | 7 |
| 20 | 1662777000 | 7 |
| 21 | 1662802200 | 5 |
| 22 | 1662787800 | 5 |
| 23 | 1662795000 | 12 |
| 24 | 1662769800 | 6 |
| 25 | 1662787800 | 0 |
| 26 | 1662787800 | 10 |
| 27 | 1662795000 | 10 |
| 28 | 1662777000 | 12 |
| 29 | 1662769800 | 18 |
| 30 | 1662769800 | 0 |
| 31 | 1662795000 | 4 |
| 32 | 1662773400 | 2 |
| 33 | 1662784200 | 12 |
| 34 | 1662787800 | 14 |
| 35 | 1662769800 | 13 |
| 36 | 1662787800 | 9 |
| 37 | 1662802200 | 12 |
| 38 | 1662787800 | 2 |
| 39 | 1662769800 | 1 |
| 40 | 1662805800 | 10 |
| 41 | 1662802200 | 15 |
| 42 | 1662773400 | 3 |
| 43 | 1662795000 | 7 |
| 44 | 1662787800 | 7 |
| 45 | 1662777000 | 2 |
| 46 | 1662777000 | 4 |
| 47 | 1662773400 | 9 |
| 48 | 1662775200 | 6 |
| 49 | 1662796800 | 6 |

针对已求解出的分配方案，利用相关数据绘制阿姨的行动轨迹图如下：

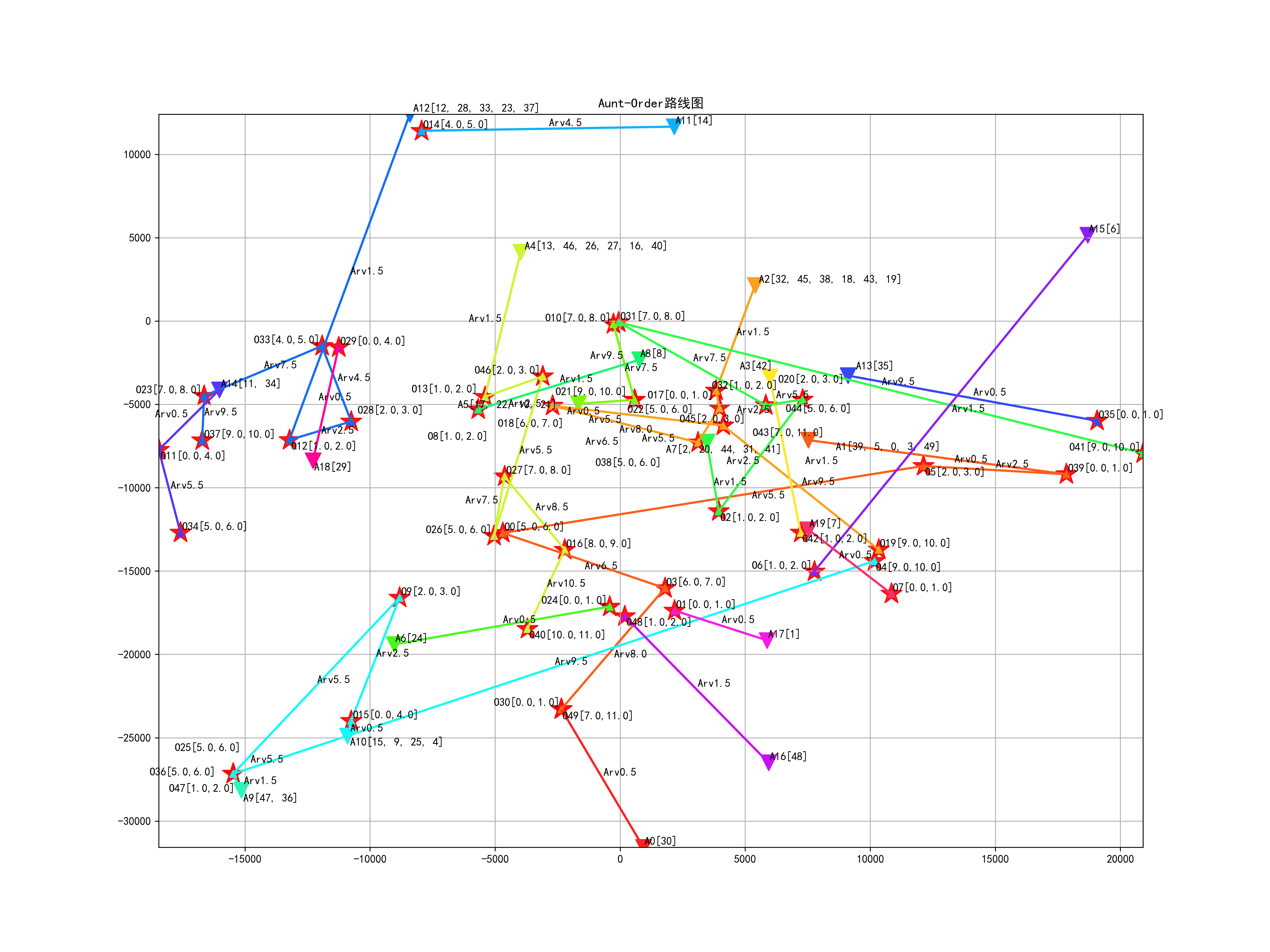


图9 阿姨行动轨迹图

图中，五角星表示订单所在处；下三角表示阿姨初始位置所在处；上三角表示阿姨到达此处开始服务，五角星与上三角重合表明该处订单成功匹配阿姨。各阿姨用不同颜色表示，使用有色线段顺序连接阿姨各坐标点，得到阿姨的行动轨迹如上图所示。

# 六、问题二模型的建立与求解

6.1模型准备与分析

不同于问题一，在线上批量派单模式下，我们无法知晓未下单的订单的信息，也无法对其作准备。因此，阿姨在接取订单时，不可提前行动，必须在订单分配以后方可从当前位置出发。

考虑到服务的开始时间和结束时间仍为半点的整数倍，本文依旧选取半点的整数倍时刻对新产生的订单统一分配。

对于阿姨而言，当天接取的第一份订单只能在分配后开始前往订单处。因此，针对问题二，所有的订单都需要考虑订单距离和通行时间，故所有订单分配时都要受到距离判断矩阵的约束。

针对压单体系和按时间戳派单模式，本文在问题一的模型中已有涉猎。为了满足线上派单模式的需求，对问题一中的模型进行适当修改。

设定新的时间轴范围。在原模型中，当时间戳抵达订单的最早服务时间时，订单才会开启，被放入求解器中求解；但引入压单体系后，订单分配时，当前时间必须至少早于最早服务时间2小时。因此，我们对原时间戳范围进行了调整，形成新的时间戳范围：



当订单满足：



即当前时刻晚于订单下单时间时，我们将订单放入求解器中，直至订单被成功分配。

至此，针对问题二的模型准备已经完成，下面着手建立模型。

6.2模型建立

相较于问题一建立的原模型，新模型仅仅对求解器模块进行了微小改动，订单分配模块和混合整数规划模块并无变化。

6.2.1求解器模型

由于问题二新增了限制条件：



本文针对订单引入新的压单判断参数。



对于不可压单的订单，将其优先级设定为1，加入“必须分配订单”列表中，在此次分配中必须被分配出去。

为了解除当前时刻的信息限制，并进一步优化目标值，本文设计并采用“未来订单”和“未来阿姨”策略。

**方案五：考虑“未来订单”、“未来阿姨”**

本文按照时间戳分配订单时，仅将当前开启的订单和可派遣的阿姨取出，放入求解器中求解。在问题二中，由于存在压单约束条件，我们必须在订单开启前将其分配。此外，本问题为动态规划问题，算法获得的当前解仅为局部最优解，且会影响后续时刻的求解。对此，本文考虑设计“未来订单”、“未来阿姨”功能，更加整体地考虑对订单进行分配。

本文定义新参数、。若订单、阿姨满足：





则将订单、阿姨取出，放入求解器中求解。通过该策略可获得整体更优的分配方案，对目标值进行再优化。

6.3模型求解

利用针对新模型调整的算法进行递归求解，得到满足约束条件和压单限制的最优分配方案。同样改变网格划分形式，比对各网格划分形式最优解，选取并输出全局最优解。

下表给出了不同参数下算法最优解的目标值与运行时间。

**表6 问题二求解结果表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 求解方案 | 初始网格参数 | 未来订单参数 | 未来阿姨参数 | 运行时间 | 目标函数值 |
| 1 | （1，1） | 2 | 0 | 2时36分06秒 | 0.481430 |
| 2 | （1，1） | 2 | 2 | 3时22分57秒 | 0.481430 |
| 3 | （2，2） | 2 | 0 | 0时30分26秒 | 0.471946 |
| 4 | （2，2） | 2 | 2 | 0时37分42秒 | 0.471946 |
| 5 | （3，3） | 2 | 0 | 0时18分24秒 | 0.471751 |
| 6 | （3，3） | 2 | 2 | 0时23分38秒 | 0.471751 |
| 7 | （4，4） | 2 | 0 | 0时10分29秒 | 0.453279 |
| 8 | （4，4） | 2 | 2 | 0时11分59秒 | 0.453279 |
| 9 | （5，5） | 2 | 0 | 0时07分09秒 | 0.466268 |
| 10 | （5，5） | 2 | 2 | 0时07分26秒 | 0.466268 |
| 11 | （1，1） | 3 | 0 | 3时52分03秒 | 0.484081 |
| 12 | （1，1） | 4 | 3 | 6时53分52秒 | **0.484569** |
| 13 | （2，2） | 3 | 0 | 1时20分00秒 | 0.473002 |
| 14 | （2，2） | 4 | 3 | 2时04分50秒 | 0.472807 |
| 15 | （3，3） | 3 | 0 | 0时34分40秒 | 0.472147 |
| 16 | （3，3） | 4 | 3 | 0时49分57秒 | 0.472331 |
| 17 | （4，4） | 3 | 0 | 0时23分41秒 | 0.470010 |
| 18 | （4，4） | 4 | 3 | 0时33分11秒 | 0.469277 |
| 19 | （5，5） | 3 | 0 | 0时09分46秒 | 0.469026 |
| 20 | （5，5） | 4 | 3 | 0时15分32秒 | 0.469659 |

算法求得全局最优分配方案的目标值为0.484569。

同样，目标函数不为随初始网格参数扩大的单调递减函数；划分网格可大幅缩减运行时间，提升运算效率。网格数越多，运行时间越短。

分析可知，采用未来订单、未来阿姨能一定程度优化最优解目标值，考虑范围越大，目标值越大。该方案也会增加运行时间，考虑范围越大，运行时间越长。在实际运用中，应合理使用未来订单、未来阿姨功能。

# 七、模型的评价与推广

7.1模型优点

（待讨论）

7.2模型缺点

1.采用“网格划分”、“优质阿姨”方案时，会剪去部分分支，其中可能包含全局最优解。

（待讨论）

7.3模型推广

（待讨论）

# 八、参考文献

（待定）