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

**题目（待定）**

**摘 要**

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。

**关键词：动态规划、混合整数规划、（待定）**

目录

[一、问题重述 1](#_Toc124785320)

[1.1问题背景 1](#_Toc124785321)

[1.2问题重述 1](#_Toc124785322)

[二、问题分析 2](#_Toc124785323)

[三、模型假设及符号说明 3](#_Toc124785324)

[3.1模型假设 3](#_Toc124785325)

[3.2符号说明 3](#_Toc124785326)

[四、问题一模型的建立与求解 4](#_Toc124785327)

[4.1模型准备与分析 4](#_Toc124785328)

[4.2模型建立 6](#_Toc124785329)

[4.2.1订单分配模型 6](#_Toc124785330)

[4.2.2“网格划分”策略 7](#_Toc124785331)

[4.2.3“网格扩大化”策略 8](#_Toc124785332)

[4.2.4求解器模型 10](#_Toc124785333)

[4.3模型求解 15](#_Toc124785334)

[五、问题二模型的建立与求解 17](#_Toc124785335)

[5.1模型准备与分析 17](#_Toc124785336)

[5.2模型建立 18](#_Toc124785337)

[5.2.1求解器模型 19](#_Toc124785338)

[5.3模型求解 19](#_Toc124785339)

[六、模型的评价与推广 20](#_Toc124785340)

[6.1模型优点 20](#_Toc124785341)

[6.2模型缺点 20](#_Toc124785342)

[6.3模型推广 20](#_Toc124785343)

[七、参考文献 20](#_Toc124785344)

# 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。服务分介于0.5和1之间，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，我们团队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

**针对问题一：**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）以及混合整数规划问题（以下简称MIP）的结合体。观察所给数据发现，阿姨及订单数量均超过2000，对于近650万个变量的规划问题，计算的时间成本极高。同时，在时间更迭中阿姨坐标变化引起的复杂约束也是难以负担的。

因此，为了克服上述困难，我们团队采用“**网格划分”策略**，彼此网格独立求解，将一个复杂问题拆解为几个独立的子问题。同时，采用**“优质阿姨”、**“**紧急订单优先”、“递归求解”策略**，在算力和时间允许的条件下，尽可能保留多种分配分支。当完成一个时刻的订单分配后，更新阿姨、订单的状态并进入下一时刻。重复上述过程，直至当前时刻超出订单分配时间区间后，输出订单分配方案。

**针对问题二：**在线上批量派单模式下，下一个时刻的订单是现有订单中未分配的订单以及下一时刻新下单的订单，因此需要我们考虑订单的下单时间。同时，由于题目要求考虑“压单”策略，而我们在第一问中割裂了相邻时间的连续性，因此我们在第一问的基础上，新增了**“未来阿姨”、“未来订单”策略。**在考虑压单的条件下，不仅考虑当前时间状态有空的阿姨，同时考虑未来一段时间内有空的阿姨。下面给出了算法的总流程图。

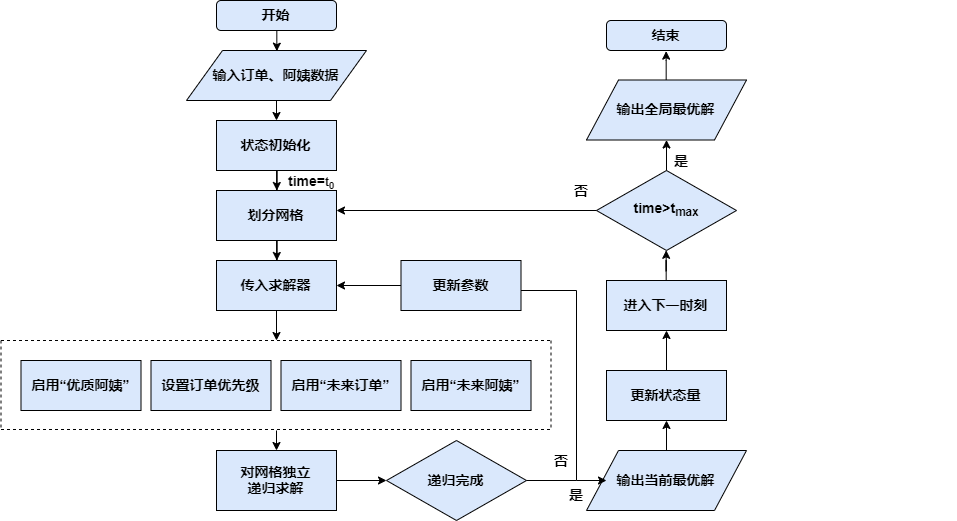


图1 优化算法总流程图

# 三、模型假设及符号说明

**3.1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。

**3.2符号说明**

**表1 符号和变量说明表**

|  |  |  |  |
| --- | --- | --- | --- |
| 当前时刻 | 阿姨数据集 | 订单数据集 | 网格参数 |
|  |  |  |  |

**表2 阿姨数据集**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参量 | 类型 | 参量说明 | 参量 | 类型 | 参量说明 |
|  | *Int* | 阿姨标签 |  | *Float* | 阿姨的服务分 |
|  | *Bool* | 阿姨是否是第一单 |  | *Float* | 阿姨可派遣时间 |
|  | *Int* | 阿姨的横坐标 |  | *Bool* | 阿姨的服务状态 |
|  | *Int* | 阿姨的纵坐标 |  | *List* | 阿姨接单的时间 |
|  | *List* | 阿姨接单的单号列表 |  |  |  |

**表3 订单数据集**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参量 | 类型 | 参量说明 | 参量 | 类型 | 参量说明 |
|  | *Int* | 订单标签 |  | *Float* | 服务结束时间 |
|  | *Int* | 订单的下单时间 |  | *Float* | 服务时长 |
|  | *Int* | 订单分配的阿姨编号 |  | *Float* | 订单的服务开始时间 |
|  | *Int* | 订单的横坐标 |  | *Bool* | 订单的开启状态 |
|  | *Int* | 订单的纵坐标 |  | *Bool* | 订单的保留状态 |
|  | *Float* | 服务开始时间 |  | *Bool* | 订单压单参数 |

# 四、问题一模型的建立与求解

**4.1模型准备与分析**

我们团队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：







将派单始点设置为0，最大派单时间设置为所有订单中服务最晚开始时间。得到时间轴范围为

对所有阿姨、订单的位置坐标进行处理，将单位转换为千米：





**订单的开启状态**，默认值为0，表示关闭状态。该状态会在当前时间满足：时转为1。

**阿姨的派遣状态**，默认值为0，表示可派遣。该状态会根据网格求解结果，将已经分配的阿姨的派遣状态置1。

**阿姨的可派遣时刻**，默认值为求解初时间点。当阿姨接单后，根据：计算出阿姨下一次接单最早时间。

**阿姨是否首次接单**，默认值为1，表示是阿姨的第一单。

根据题目所给数据和条件，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，订单和阿姨的状态只会在半点的整数倍时刻进行更新。因此我们团队将时间轴按每30分钟划分，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。

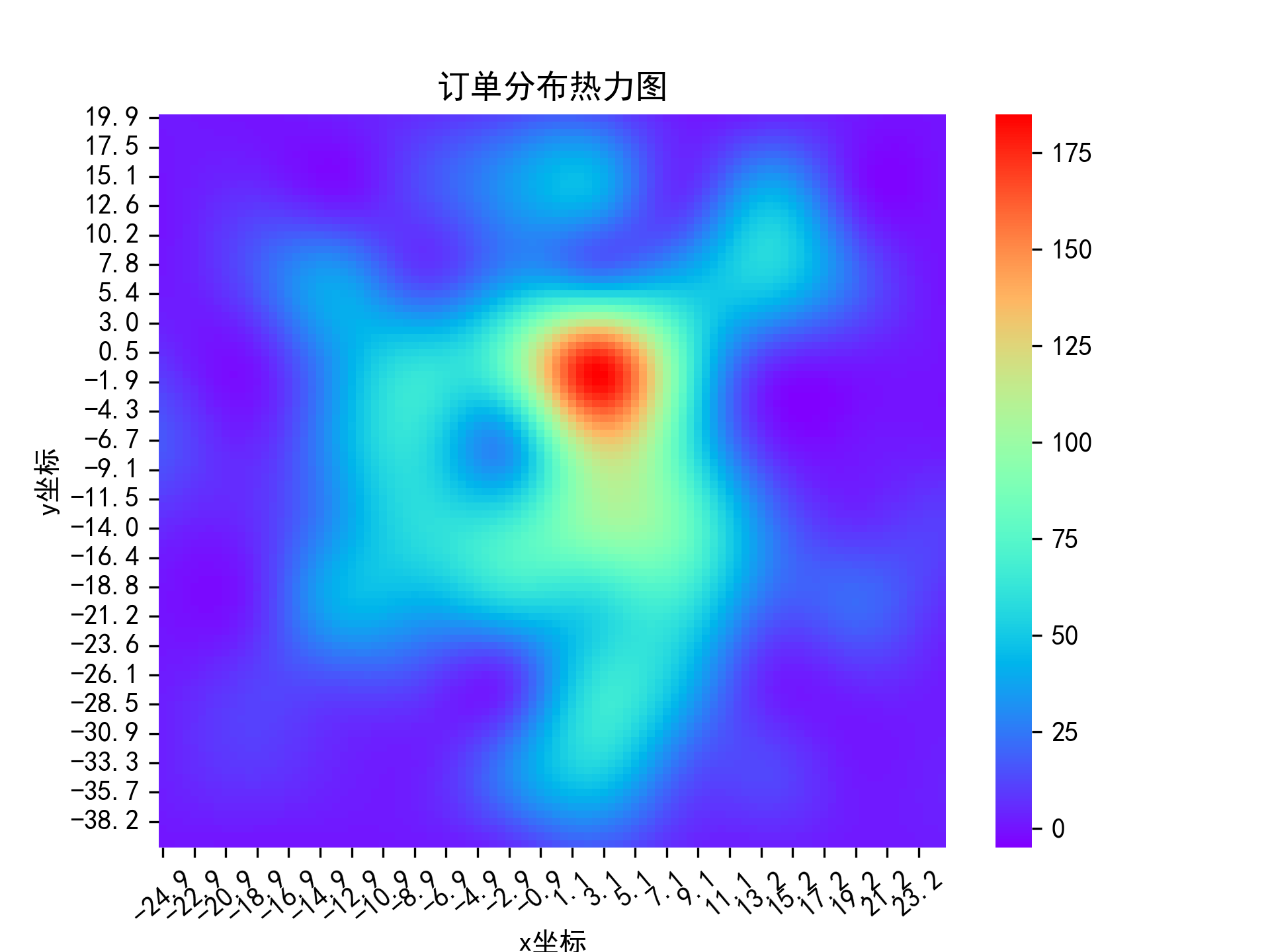
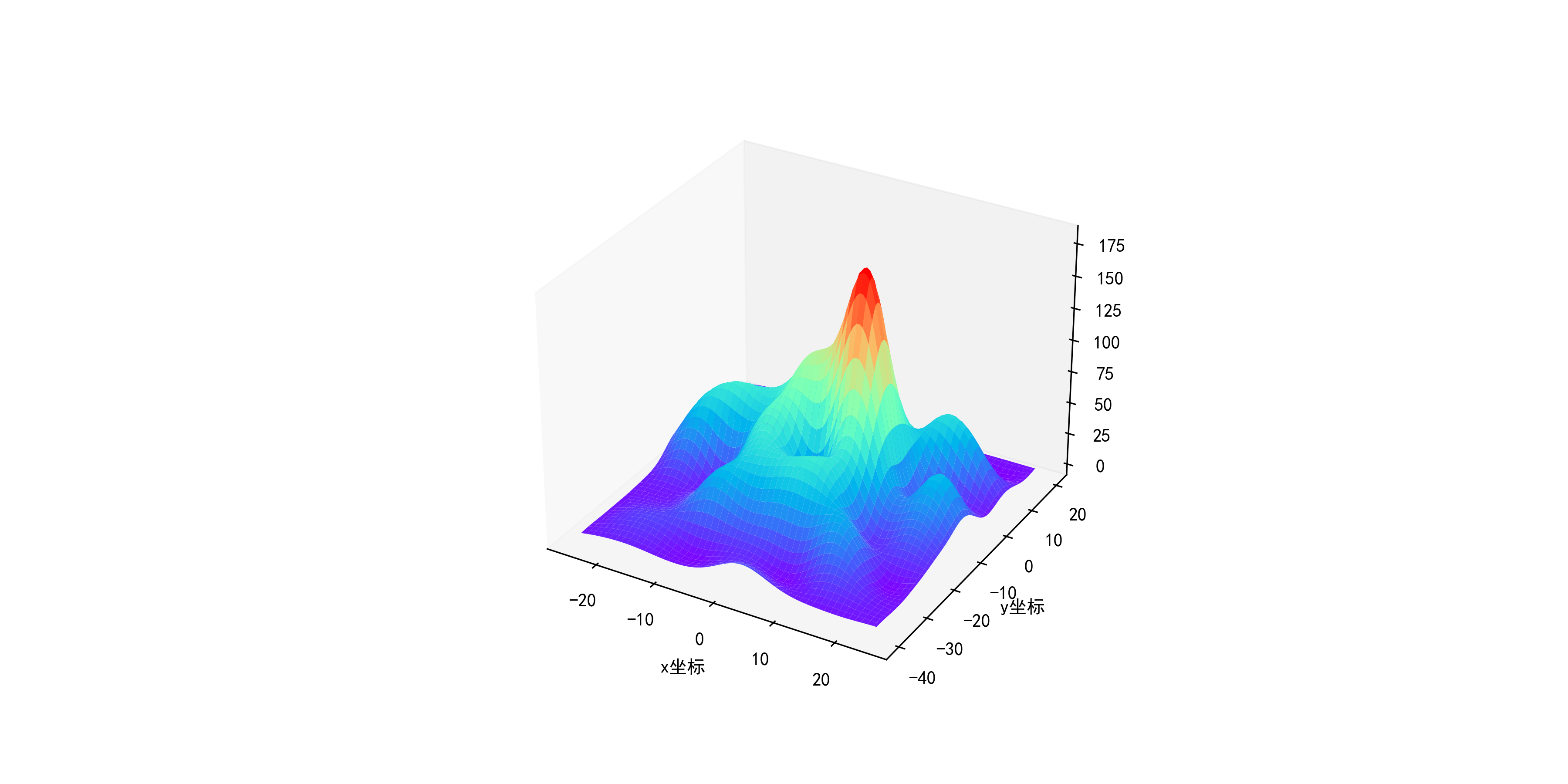


图2 订单分布三维插值图、热力图

图2展示订单的分布状况和密集程度。图2中，、单位均为千米；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。同时，粗略绘制了订单分布的等值线图。

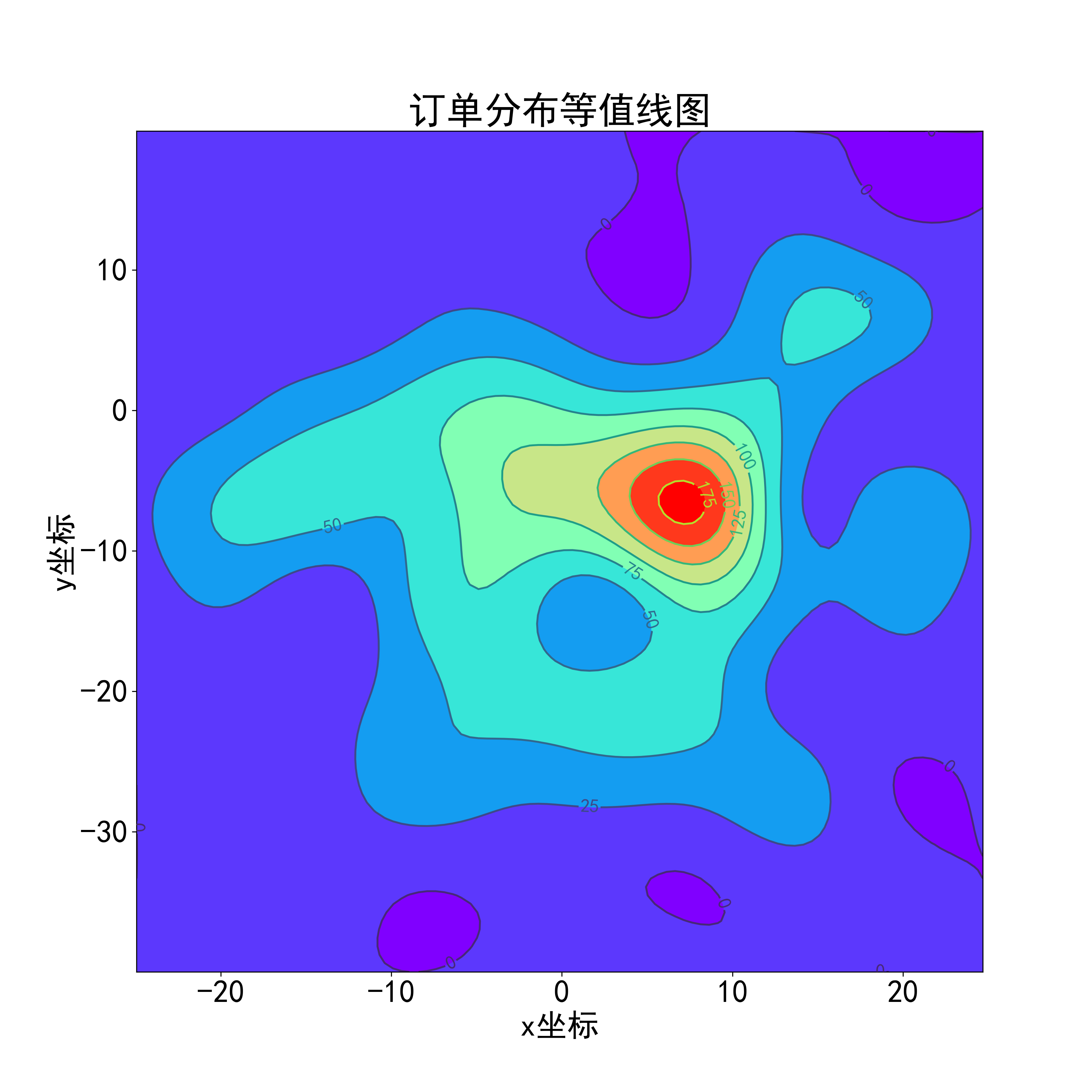


图3 订单分布等值线图

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，数量大于175份；而除了整个区域的中心，其他区域的订单数相当稀少，普遍在50-75单左右。而且随着地点愈发远离中心，订单更加稀少，并且出现了相当一部分区域的订单数从始至终都为0的情况。

**4.2模型建立**

4.2.1订单分配模型

**定义距离判断矩阵**，判断阿姨在某一时刻是否能及时到达订单位置。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，阿姨可在订单派送初始时刻前就开始前往订单处，等待服务开始。因此，我们团队对模型作出假设：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，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，虽然在理论上获得更高的目标值，但是付出的时间成本很可能远超这一丁点的目标提升。所以，为了综合提升结果的时效性，我们团队设计并采用“网格划分”策略。

4.2.2“网格划分”策略

我们团队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，我们团队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。

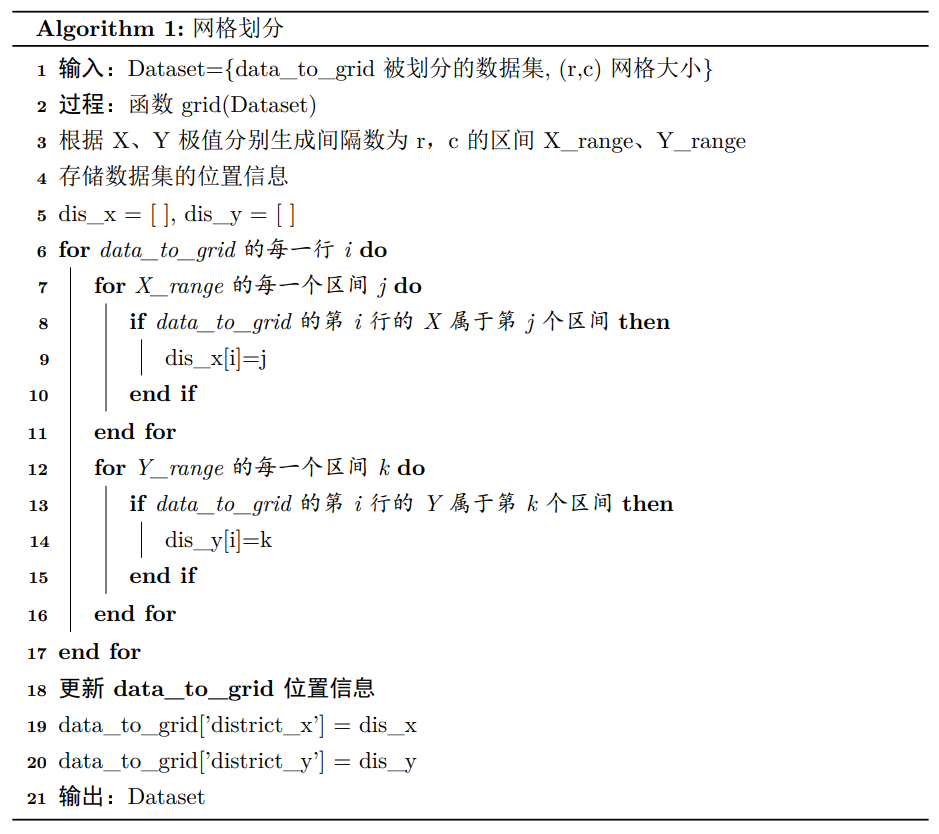


图4 网格划分伪代码

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。我们团队尝试绘制网格分布图如下。

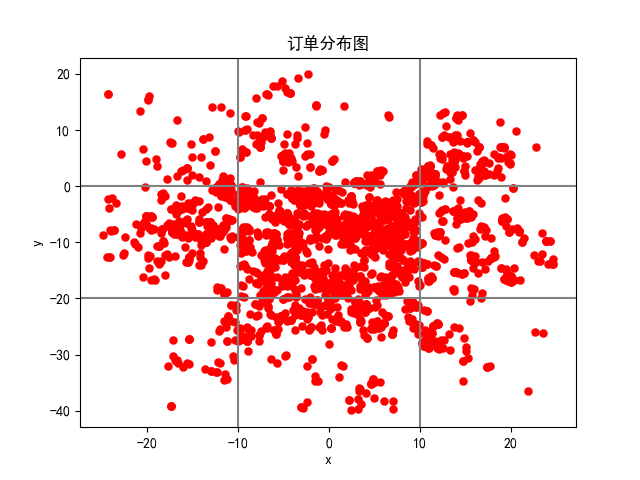
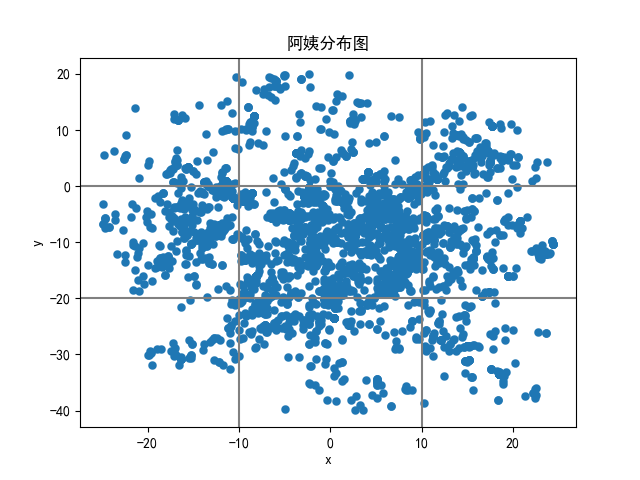


图5 3\*3网格划分形式下阿姨、订单分布图

4.2.3“网格扩大化”策略

由于每个网格内的阿姨并不是总能满足该网格内需要分配的订单。特别是在网格数较多的时候，容易出现需要分配的订单数大于现有阿姨数的情形。因此，我们需要将周围网格空闲的阿姨调度至附近紧缺阿姨的网格中。

要实现这一目标，最为简单可行的方案是以某种方式得到一个更小的网格参数，再对数据集进行划分，并在网格中求解MIP问题。这种方案相对于只对数据集进行一次网格划分，在进行完一次求解之后，对网格内存在未分配订单的网格，抽调该网格周围最多8个网格中空闲阿姨的方案而言，更为优秀。理由如下：

1. 由于是网格独立求解，在抽调周围空闲阿姨的过程中会存在一个先后顺序，这个顺序会影响目标函数值。
2. 由于抽调的最多8个网格中空闲阿姨的数量会很大，这会极大的影响计算时间。

而网格扩大化方案能够避免（1）。对（2）而言，可以选择变化较为平缓的函数，来冲抵这一不良效应，因此我们选择网格扩大化方案。下图展示了我们网格扩大化方案的伪代码。

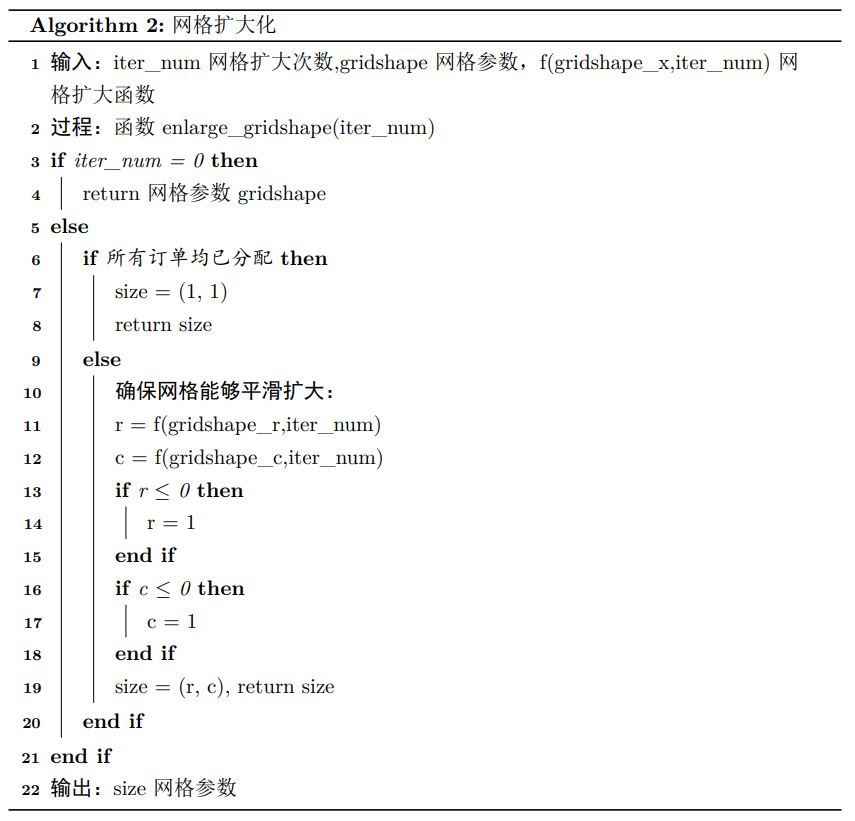


图6 网格扩大化伪代码

我们团队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。

根据上述分析及实验，函数能够满足我们的要求。所以我们团队决定使用具有鲁棒性，总能在不使目标值下降过多的情况下显著提升运行效率的迭代方式三。

当然函数在这个问题中是优秀的，并不一定代表它在新的数据集中同样具备优良的性质，一切结果需经过实证。

4.2.4求解器模型

**一、“优质阿姨”策略**

由于网格的大小限制和目标函数中参数的大小，我们可以认为在一个较小的网格中，阿姨整体的平均通行距离和服务间隔时间处于一个较小的值，对目标函数起次要影响作用；而认为阿姨的服务分成为了影响网格内目标函数值的主要因素。因此，我们设立了“优质阿姨”方案。

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”。

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。但综合考虑订单分配的时效性，“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。

**二、“紧急订单优先”策略**

分析可知，在非初始时刻时，每次纳入求解器中需要分配的订单中，存在两种状态：之前时刻未分配的订单、当前时刻开启分配状态的订单。而对于那些之前未分配同时又临近最晚服务时间的订单，将他们与新开启分配状态的订单做等同处理，显然是不合理的。因此，我们团队采用“紧急订单优先”策略。

我们团队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：



我们团队使用****函数按照值的大小排名。中值并列最小者为1，其余依次类推，并将其定义为订单优先级。

**三、“递归求解”策略 & MIP模型**

**MIP模型**

**（1）约束条件**

设为时刻网格中纳入求解器的订单数，为时刻网格中空闲的阿姨数。

设置时刻订单分配矩阵。



设置时刻距离判断矩阵。



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：



设置时刻优质阿姨向量。



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：



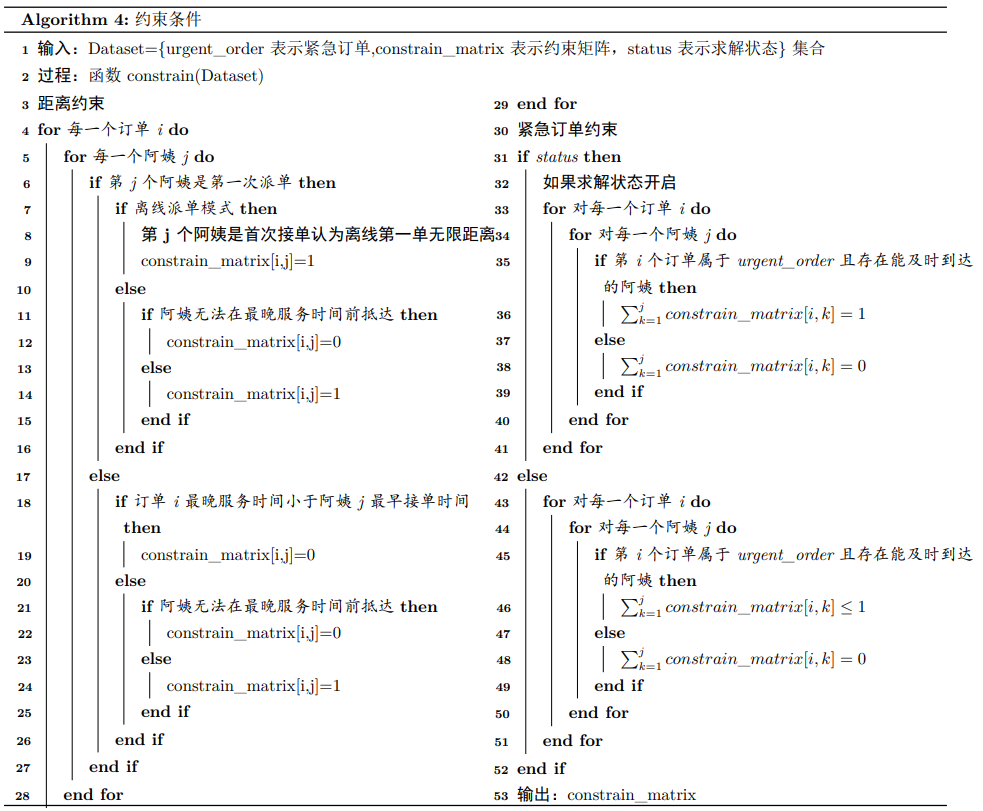
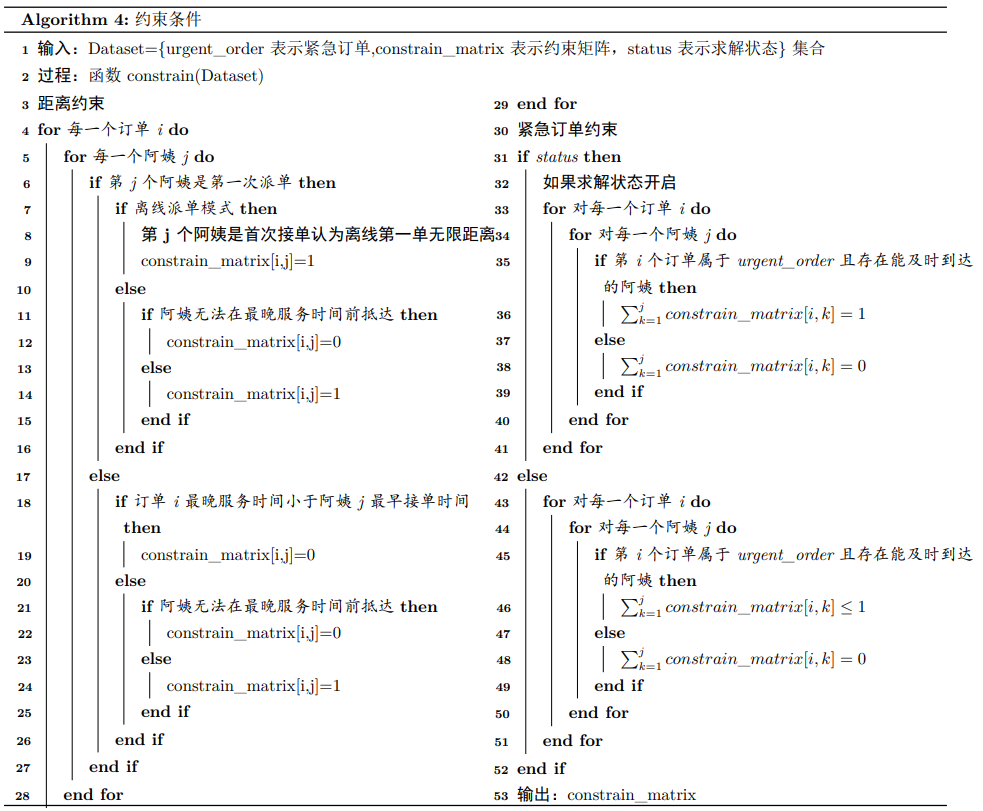
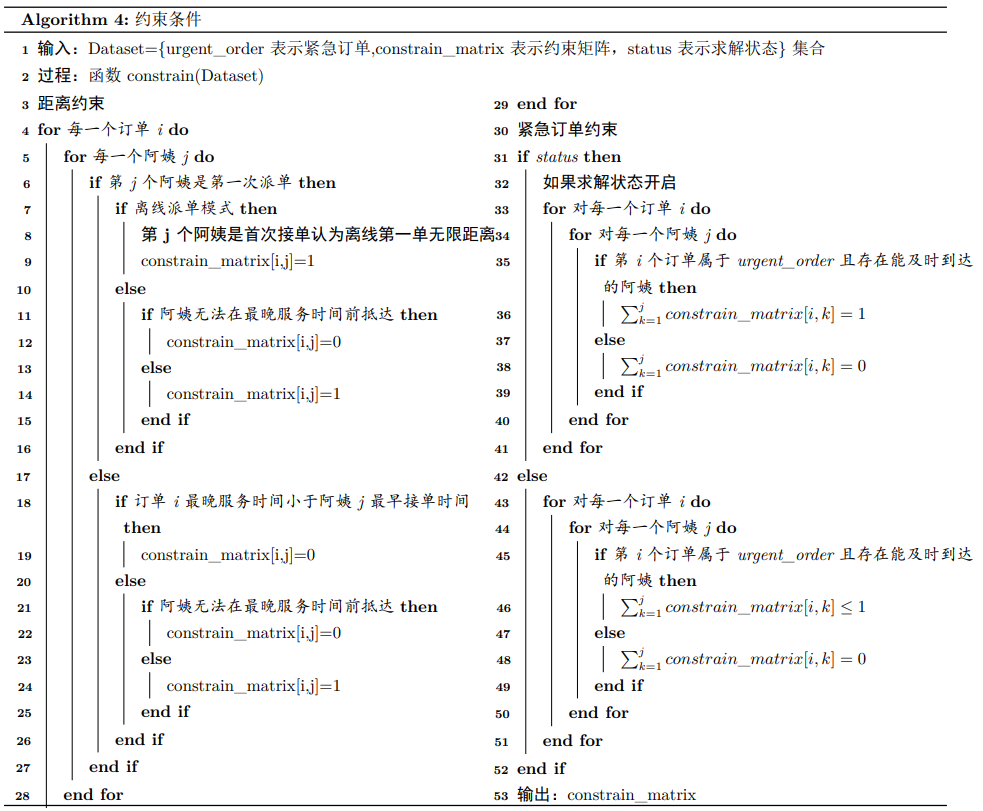
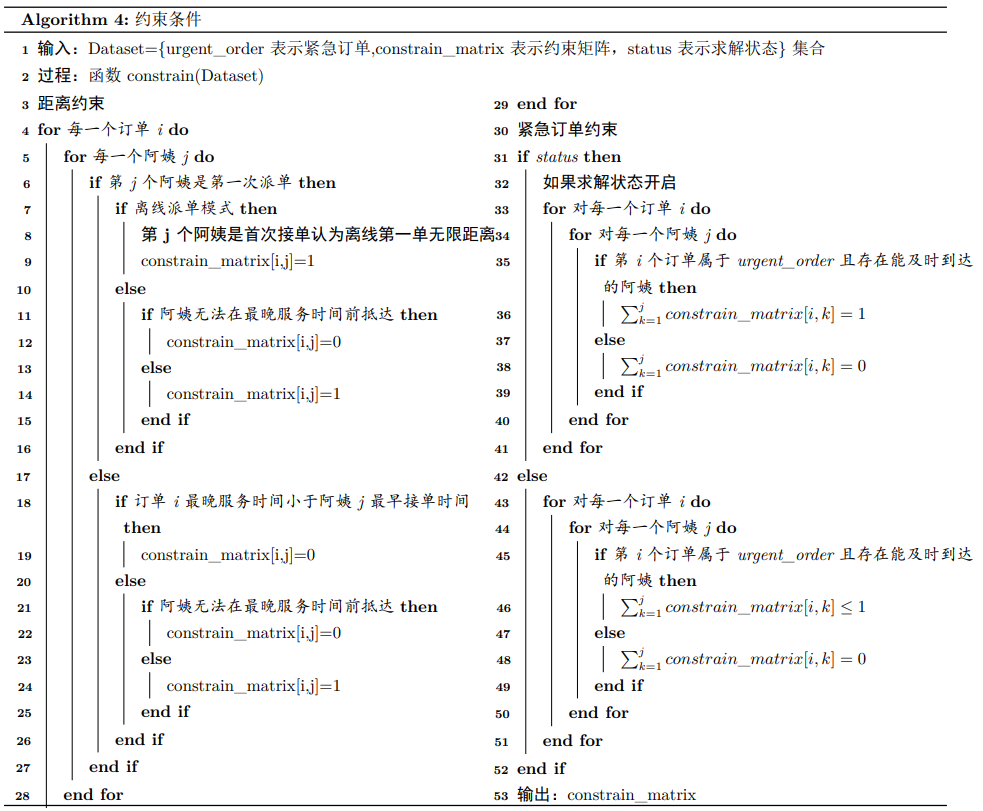
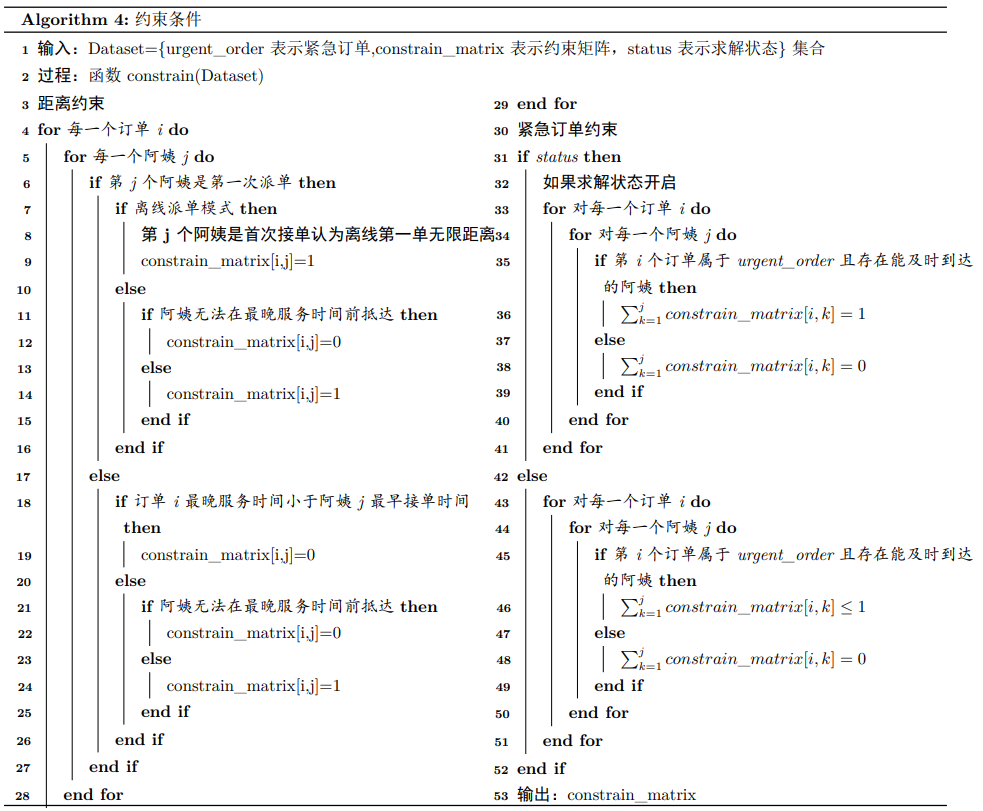
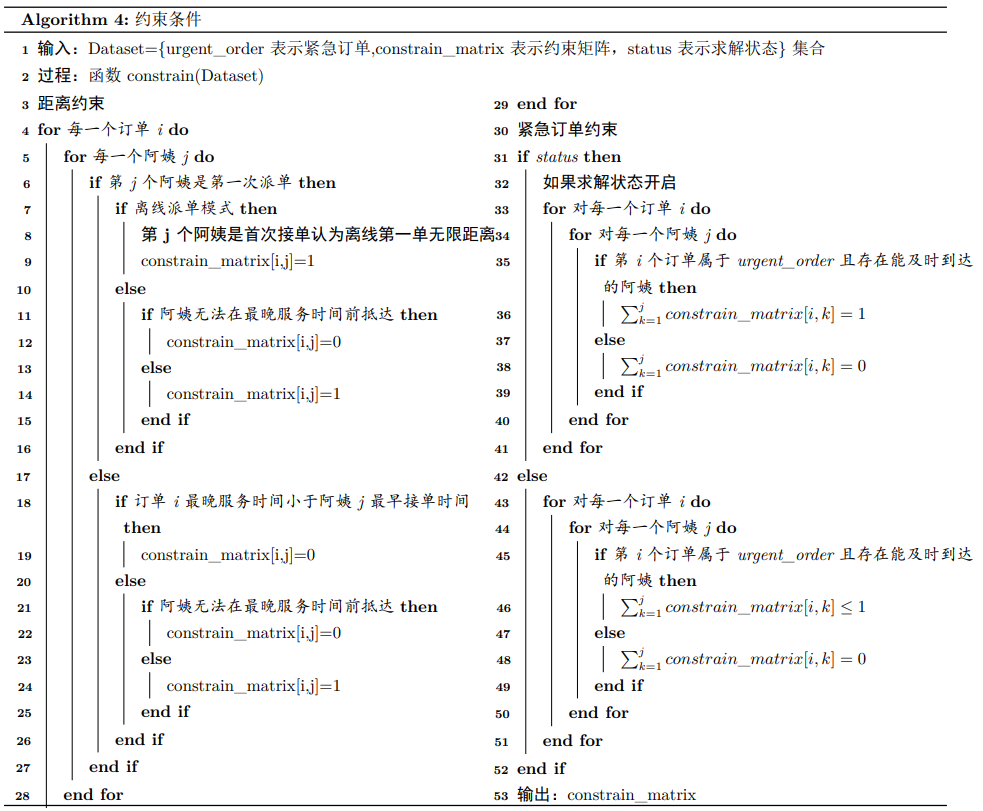


图7 约束条件伪代码

基于“紧急订单优先”策略以及可能在求解过程中，无法找到可行解的情况。我们为每一个时刻每个网格的解矩阵定义了一个特殊约束，记为，上图给出了特殊约束构造的伪代码。同时为MIP问题增添约束：



**（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：



综上所述，混合整数规划模型为：





**“递归求解”策略**

下图为“递归求解”策略伪代码，求解时的重要参数均在伪代码中有相应解释。

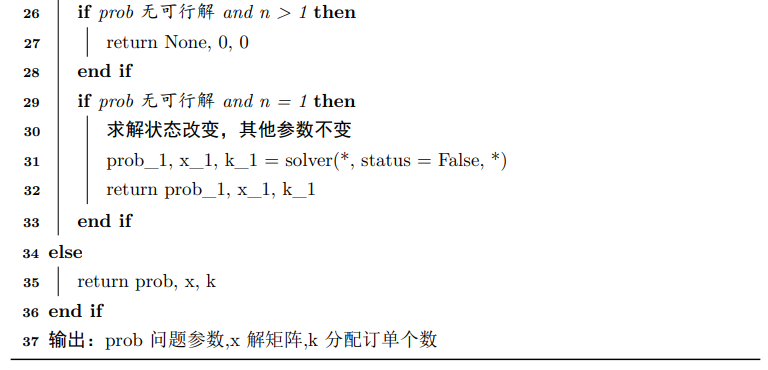
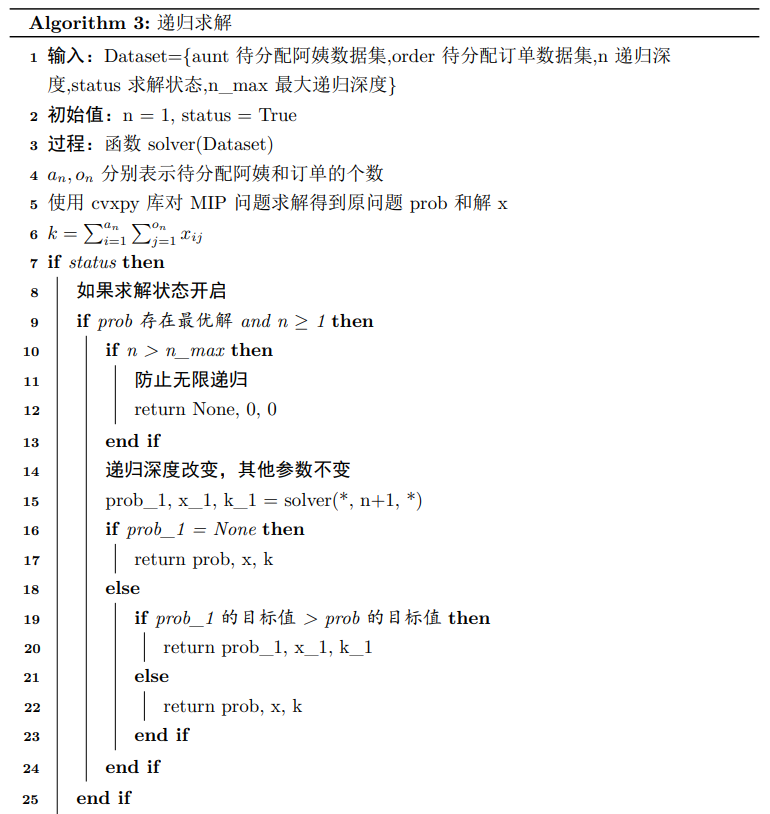


图8 递归求解伪代码

1.1：求解时率先将订单优先级最高的订单纳入求解器中，使得这些订单在此次分配中必须被分配。若存在最优解，则更新递归深度参数，并且继续调用该求解器，所有递归求解中目标函数最大者，作为该网格内订单分配的解矩阵。其中，递归深度参数控制的是紧急订单的范围，第一次只取优先级为1的订单，第二次取优先级为1和2的订单，以此类推，直至到达最大值。最后返回求解结果。

1.2：若在没有到达最大值时，求解器就已经出现了无可行解的情况。我们就将之前历史出现的最优求解结果返回。

2：若第一次求解时，求解情况是无可行解。说明紧急订单中有一部分无法分配，我们只能退而求其次，让更多的紧急订单配分配。于是，我们更新求解器的求解状态 为，并再次递归调用求解器，将此时的求解结果返回。

综上，求解器每次求解必定能输出至多退化为零矩阵的解。

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，重复上述过程。求解时，还有可能出现网格参数扩大至时，仍然存在订单未被未被分配的情况。这时，这些订单在下一时刻会被继续纳入求解器求解。

此外，我们的模型依赖于许多参数，如网格划分参数、是否采用“优质阿姨”策略、“网格扩大化”函数等。因此我们还需要进行多次实验，比较它们之间目标函数和运行时间。

**4.3求解结果**

**4.3.1问题一第一问的结果**

由于前两个“网格扩大化”函数会导致大量订单无法分配，因此以下结果均基于第三种“网格扩大化”函数得出。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） |  | **0.587064** |
| 2 | 是 | （1，1） |  | 0.530761 |
| 3 | **否** | （2，2） |  | **0.584485** |
| 4 | 是 | （2，2） |  | 0.526610 |
| 5 | **否** | （3，3） |  | **0.583332** |
| 6 | 是 | （3，3） |  | 0.515351 |
| 7 | **否** | （4，4） |  | **0.580127** |
| 8 | 是 | （4，4） |  | 0.523255 |
| 9 | **否** | （5，5） |  | **0.576018** |
| 10 | 是 | （5，5） |  | 0.524255 |

在不设置“优质阿姨”和初始网格参数为的条件下，算法求得全局最优分配方案的目标值为0.587064。

从结果表中分析可知：

1. 随着初始网格数量的下降，目标函数会缓慢上升，在现有求解条件下，呈现单调上升趋势。
2. 启用“优质阿姨”方案会显著拉低目标函数值。同时带来的计算速度提升，在初始网格数量较大时不明显；但在较小时，使运行时间下降了30.6%。
3. 启用“优质阿姨”后，目标函数值并不随着初始网格数量下降而上升，同时目标函数损失远大于不启用时，比不启用时平均下降了10.0%。

综上所述，在处理不同的问题时，需要多次实验。在考虑当前资源限制的条件下，选择是否使用“优质阿姨”和网格参数进行求解。

**4.3.2问题一第二问的结果**

我们团队基于已设计的算法，对附件中前50份订单和前20位阿姨进行匹配。经过和第一问类似的调参过程，最终选择了目标函数最大时的参数组合：初始网格参数为、不使用“优质阿姨”。下表根据最大参数组合得出的结果绘制。

**表5 问题一第二问阿姨任务列表**

|  |  |  |  |
| --- | --- | --- | --- |
| Aunt\_id | 任务列表 | Aunt\_id | 任务列表 |
| 0 | 30、25 | **10** | 15、9、26、27、16、40 |
| 1 | 39、5 | **11** | - |
| 2 | 32、45、38、3、4 | **12** | 12、28、33、23、37 |
| 3 | 42 | **13** | 35 |
| 4 | 13、46、14、31 | **14** | 11、34 |
| 5 | 17、22、18、10、21 | **15** | 41 |
| 6 | 24、48、0、49 | **16** | 6 |
| 7 | 2、20、44、43、19 | **17** | 1 |
| 8 | 8 | **18** | 29 |
| 9 | 47、36 | **19** | 7 |

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：

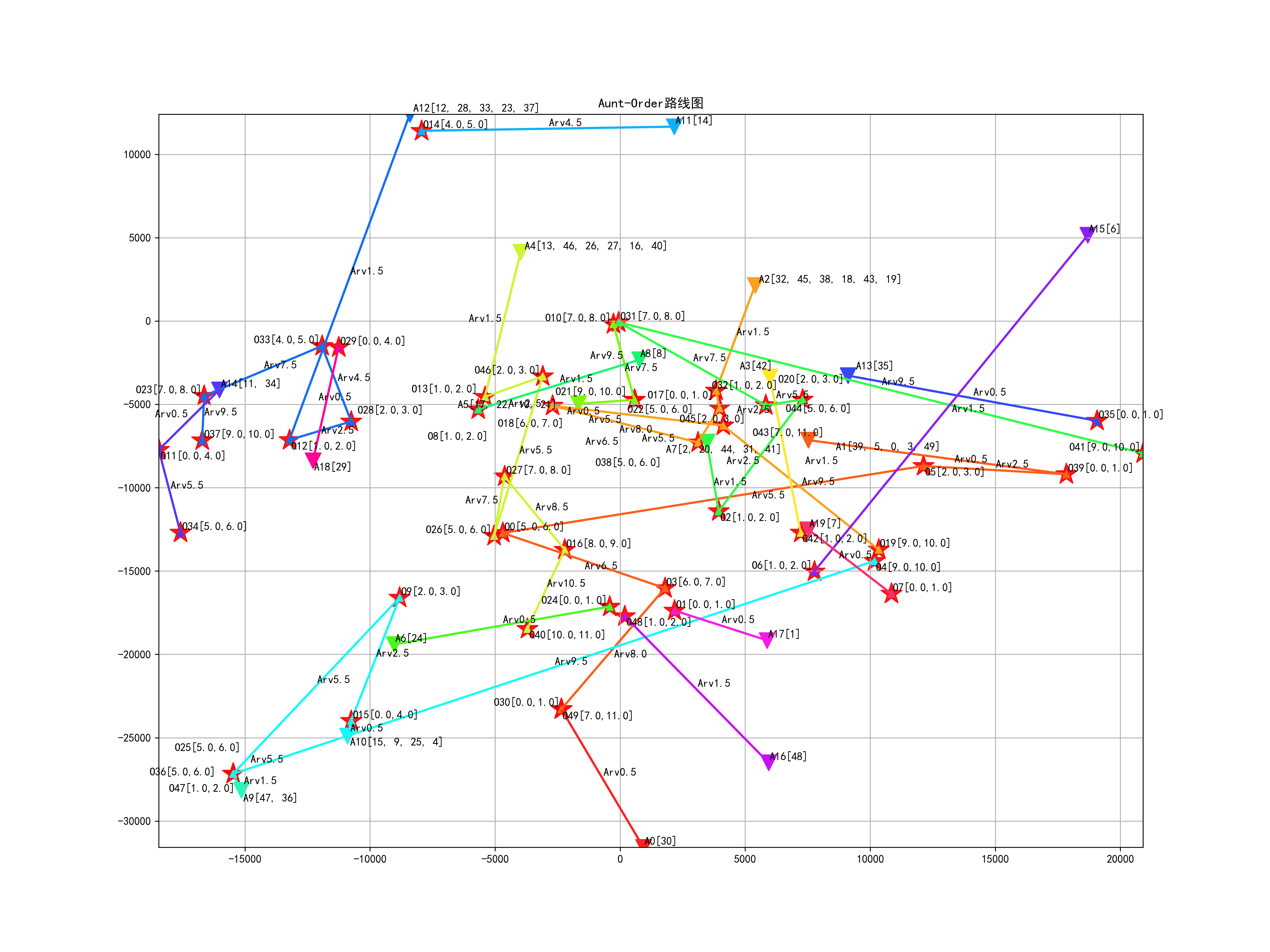


图9 阿姨行动轨迹图

图中，五角星表示订单所在处，同时标有订单ID和服务时间区间，如O48[1.0,2.0]；下三角表示阿姨初始位置所在处，标有阿姨ID和接取的全部订单，如A16[48]；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。

各阿姨和所接取的订单用同种颜色的线段按订单服务顺序连接阿姨和订单；阿姨开始服务的时间点在阿姨和订单之间的连线上标出，如连接A16和O48间的Arv1.5。得到阿姨的行动轨迹如上图所示。

# 五、问题二模型的建立与求解

**5.1模型准备与分析**

在线上批量派单模式下，以下几点与离线派单不同，需要进行模型推广：

1. 我们无法知晓未下单的订单的信息；
2. 我们需要考虑“压单”，即考虑已经下单、并需要在未来提供服务的订单；
3. 由于考虑未来订单的分配，可能出现当前时间空闲阿姨不足的情况；
4. 考虑未来订单的分配，需要重新设计订单分配的时间轴范围；
5. 考虑未来订单的分配意味着更新阿姨订单状态时的方式也要做相应的调整；
6. 阿姨的首单通行距离不再是无限，需要根据派单始点重新考虑。

对于（1），我们需要将订单的创建时间纳入考虑订单选择的条件，推广须分配订单的概念。

对于（2）、（3），我们采用**“未来订单”策略**和**“未来阿姨”策略。**也就是说，我们不仅考虑当前时间需分配的订单和空闲的阿姨，还考虑未来一段时间的需分配的订单和空闲的阿姨。

对于（4），我们需将订单分配的始点扩大，也就是在的某个时刻就开始分配订单**。**

对于（5），我们推广了阿姨和订单状态的更新公式，使离线模式成为了线上模式的一个特例。

对于（6），我们重新定义了中阿姨首单的通行距离。

**5.2模型建立**

**5.2.1参数推广公式**

此时**订单的开启状态**，在当前时间满足：时才转为1。

在线上派单模式下，定义新参数，在离线模式下默认为0。此时，线上派单时间轴的最大值和离线时相同，均为服务最晚开始时间；但时间轴始点设置为，由此可以计算出新的时间轴。

**5.2.2“未来订单”、“未来阿姨”策略**

在离线派单模式下，没有考虑未来时间的订单，因此无法在满足至多压单2小时的条件下，完成订单分配任务。因此，有必要采用 “未来订单”、“未来阿姨”策略。采用这两个策略，冲抵了一部分由于前后时间独立求解的不利因素，在一定程度上，起到了继续优化的效果。

我们团队定义新参数、。若订单、阿姨满足：





则将订单、阿姨在时刻的和调整为可以加入分配的状态，并将他们一并纳入求解器中。

**5.2.3求解器模型**

由于问题二新增了限制条件：



考虑到“未来订单”策略，在单独一次求解中，订单的时间类型有多种：有时间被“压单”、当前时间段开启的以及考虑未来一段时间将要开启的订单。为了确保在2小时压单极限前，将这一批存单及时分配。我们有必要引入一个控制最大压单限额的参数，和“紧急订单优先”策略同时存在，控制单次求解订单纳入求解器的顺序。



对于不可压单的订单，将在“紧急订单优先”策略中的订单优先级直接指定为1，加入“必须分配”订单中，在单独一次分配中会被优先考虑分配。

**5.3求解结果**

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 |  | 0.481430 |
| 2 | （1，1） | 2 | 2 |  | 0.481430 |
| 3 | （1，1） | 3 | 0 |  | 0.484081 |
| 4 | （1，1） | 4 | 3 |  | **0.484569** |
| 5 | （2，2） | 2 | 0 |  | 0.471946 |
| 6 | （2，2） | 2 | 2 |  | 0.471946 |
| 7 | （2，2） | 3 | 0 |  | 0.473002 |
| 8 | （2，2） | 4 | 3 |  | 0.472807 |
| 9 | （3，3） | 2 | 0 |  | 0.471751 |
| 10 | （3，3） | 2 | 2 |  | 0.471751 |
| 11 | （3，3） | 3 | 0 |  | 0.472147 |
| 12 | （3，3） | 4 | 3 |  | 0.472331 |
| 13 | （4，4） | 2 | 0 |  | 0.453279 |
| 14 | （4，4） | 2 | 2 |  | 0.453279 |
| 15 | （4，4） | 3 | 0 |  | 0.470010 |
| 16 | （4，4） | 4 | 3 |  | 0.469277 |
| 17 | （5，5） | 2 | 0 |  | 0.466268 |
| 18 | （5，5） | 2 | 2 |  | 0.466268 |
| 19 | （5，5） | 3 | 0 |  | 0.469026 |
| 20 | （5，5） | 4 | 3 |  | 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。~~

# 六、模型的评价与推广

**6.1模型优点**

1.采用多种优化策略，如“优质阿姨”、“未来阿姨”、“网格扩大化”等。能够显著的提升目标函数值，降低运行时间成本。

2.提供多种不同的参数组合，使用者可以综合考虑自身资源限制，选择最优的模型参数求解。

3.单独网格求解过程中，考虑了相邻时间，相邻网格之间的联系。一定程度上削弱了网格和时间独立假设对目标函数的影响。

4.独立网格求解采用“递归取优”的方式确定单独网格的解，使得目标函数进一步优化。

5.**（补充）**

**6.2模型缺点**

1.在模型简化的过程中，可能存在过度剪枝，存在目标函数并没有达到全局最优解的可能。

2.网格扩大化的方式多种多样，并不一定能够保证我们团队的扩大化方式是最优的方式。

3. **（补充）**

**6.3模型推广**

1.团队基于离线派单模型推广至线上派单模型，使得离线派单成为线上派单的一个特例情况。同时，对于非MIP问题，只需要重新定义目标和约束，便能很快适应一个新的问题，有很强的普适性。

2.模型内置多种优化策略，均为独立封装的代码包，便于使用和后续的升级

3.模型还接受更多新的优化策略，能够便捷地和其他优化策略结合使用。

4. **（补充）**

# 七、参考文献

（待定）