|  |  |
| --- | --- |
| 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](#_Toc123755040)

[1.1问题背景 1](#_Toc123755041)

[1.2问题重述 1](#_Toc123755042)

[二、问题分析 2](#_Toc123755043)

[2.1问题一的分析 2](#_Toc123755044)

[2.2问题二的分析 2](#_Toc123755045)

[三、模型假设 2](#_Toc123755046)

[四、符号说明 3](#_Toc123755047)

[五、问题一模型的建立与求解 4](#_Toc123755048)

[5.1模型准备与分析 4](#_Toc123755049)

[5.2模型建立 10](#_Toc123755050)

[5.3模型求解 11](#_Toc123755051)

[六、问题二模型的建立与求解 13](#_Toc123755052)

[6.1模型准备与分析 13](#_Toc123755053)

[6.2模型建立 14](#_Toc123755054)

[6.3模型求解 15](#_Toc123755055)

[七、模型的评价与推广 16](#_Toc123755056)

[7.1模型优点 16](#_Toc123755057)

[7.2模型缺点 16](#_Toc123755058)

[7.3模型推广 16](#_Toc123755059)

[八、参考文献 16](#_Toc123755060)

# 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。为了提升算法的求解能力，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

2.1问题一的分析

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。综合考虑后，我们决定采用Python中的pandas、numpy、matplotlib等库来处理数据、建立模型、绘制图像。因阿姨及订单的数量均超过了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，我们尝试启用“网格划分”、“优质阿姨”策略，将订单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。为了求解全局最优分配方案，我们按照时间戳，采用不同的区域划分方式，针对每种方案使用cplex、cvxpy库中的求解器以逆向递归的方式不断优化目标值，选出当前的最佳分配方案，即局部最优解。比对多个局部最优解，最终选出全局最优解，得到最佳分配方案。此外，我们对算法进行改进，设计在当前时间戳将未来数小时的订单纳入计算的功能，进一步完成对目标值的优化，获得更优的订单分配方案。

流程图（待讨论）

2.2问题二的分析

问题二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。基于问题一中已建立的模型，我们对其进行修改和完善。为了满足压单的限制条件，我们引入新的约束，并对当前时间戳的订单赋予不同的优先级。对订单进行预处理后，放入求解器中进行分配。考虑不同的区域分配方案，比对各方案的目标值，逆向递归选出最终决策结果。

流程图（待讨论）

# 三、模型假设

1.假设所有订单都要分配一个且只有一个阿姨；

2.假设每个订单指定的服务开始时间在[最早时间，最晚时间]范围内，且是半点的整数倍；

3.假设一个阿姨同时只能服务一个订单；

4.假设阿姨需要在订单的服务开始时间之前到达客户位置；

5.假设阿姨每天开始任务时须从初始点位置出发；

6.假设任意两点的距离均为欧氏距离；

7.假设保洁阿姨的行驶速度为恒定的15千米/小时。

# 四、符号说明

**表1 符号和变量说明表**

|  |  |
| --- | --- |
| 符号 | 符号说明 |
|  | 当前时刻 |
|  | 阿姨编号，阿姨的唯一标识 |
|  | 订单编号，订单的唯一标识 |
|  | 时刻的距离判断矩阵 |
|  | 距离判断矩阵元，阿姨能够在服务时间区间内到达则置1，反之置0 |
|  | 时刻的订单分配矩阵 |
|  | 订单分配矩阵元，订单分配给阿姨则置1，反之置0 |
|  | 优质阿姨矩阵 |
|  | 优质阿姨矩阵元，阿姨被定义为优质阿姨置1，反之置0 |
|  | 地图区域横坐标，表示第行的区域 |
|  | 地图区域纵坐标，表示第列的区域 |
|  | 当前时刻区域中的阿姨数 |
|  | 当前时刻区域中的订单数 |
|  | 网格扩大化的迭代次数 |
|  | 网格参数，表示将地图划分为行列的区域 |
|  | 订单匹配的阿姨的服务分 |
|  | 订单匹配的阿姨的通行距离 |
|  | 订单匹配的阿姨的服务间隔时间 |
|  | 目标函数中阿姨服务分的权重 |
|  | 目标函数中阿姨通行距离的权重 |
|  | 目标函数中阿姨服务间隔时间的权重 |
|  | 表示“未来规划”考虑的小时数 |
|  | 求解器参数，表示当前设定的紧急订单数量 |
|  | 针对问题二设定的新变量，表示当前时刻 |
|  | 订单的下单时间 |
|  | 订单的最早服务时间 |
|  | 压单判断矩阵 |
|  | 压单判断矩阵元，订单当前时刻可压单置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，将所有订单中最小的服务开始时间设定为0时刻，并对订单下单时间和最晚服务时间进行相同的分析处理。分析得知，所有订单的工作时间、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，即订单和阿姨的状态都只会在半点的整数倍时进行更新。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即可。现对订单进行统计，可以得出各时刻订单的最早服务时间分布图。图1、图2分别给出了订单分布的折线图和频率直方图。

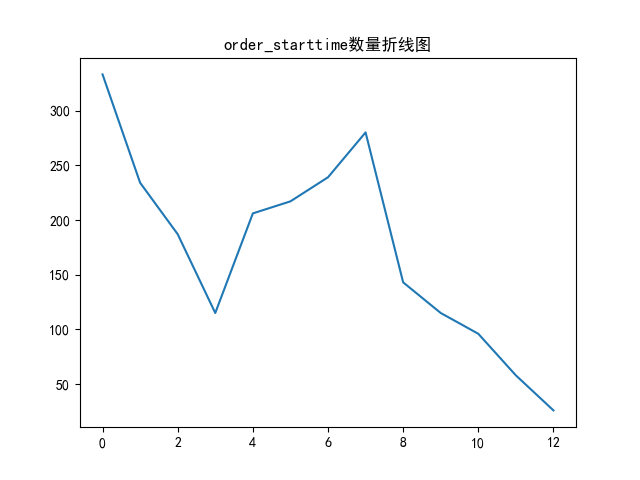


图1 订单最早服务时间分布折线图

7

图2 订单最早服务时间分布频率直方图

由图1、图2可知，订单在0时刻和7时刻附近是高峰期，订单数量较多，但是订单最多的时刻数量也不会超过350份。考虑到订单具有服务时间区间，我们粗略估计某时刻待分配的订单数也大概率不会超过700份。由于阿姨共有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。

现对阿姨的初始坐标和订单的坐标进行统计分析，绘制阿姨分布图、订单分布图。图3给出了所有阿姨的初始坐标分布图，图4给出了所有订单的坐标分布图。

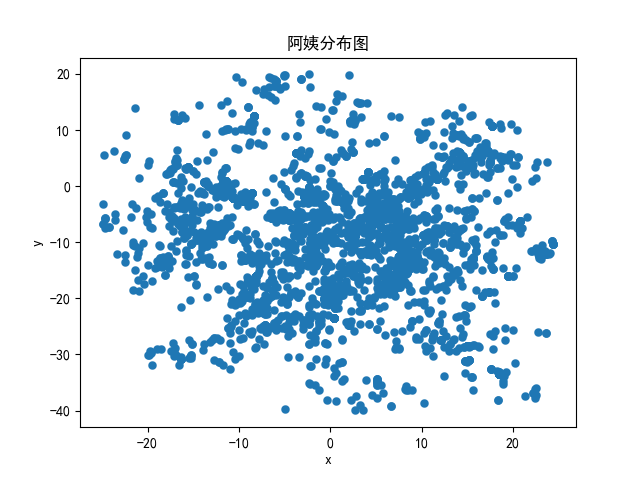


图3 阿姨初始坐标分布图

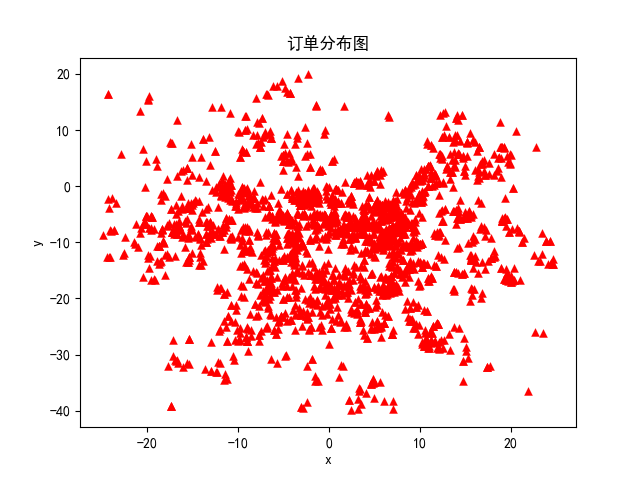
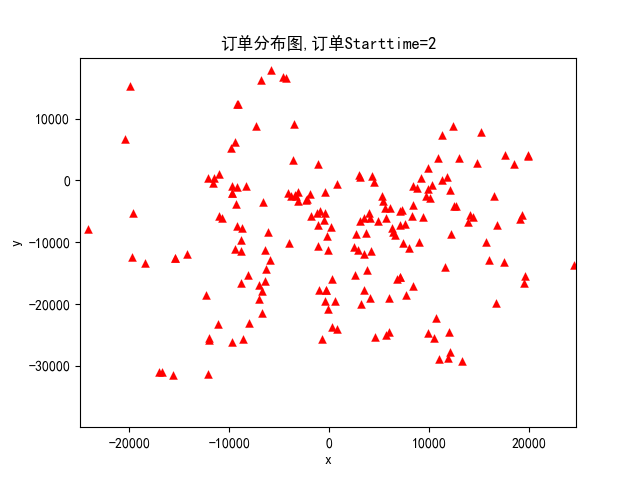
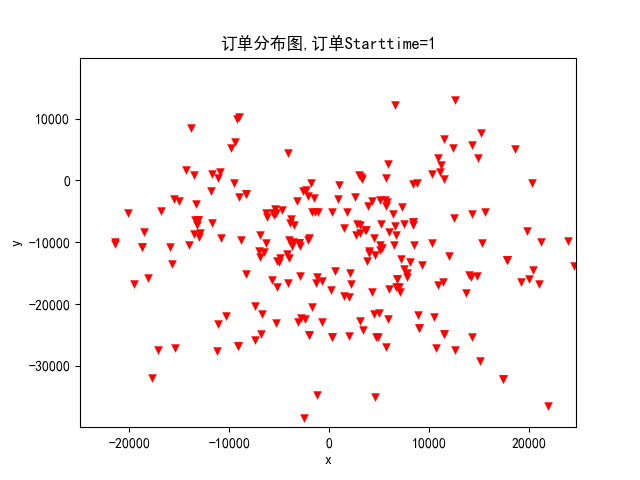
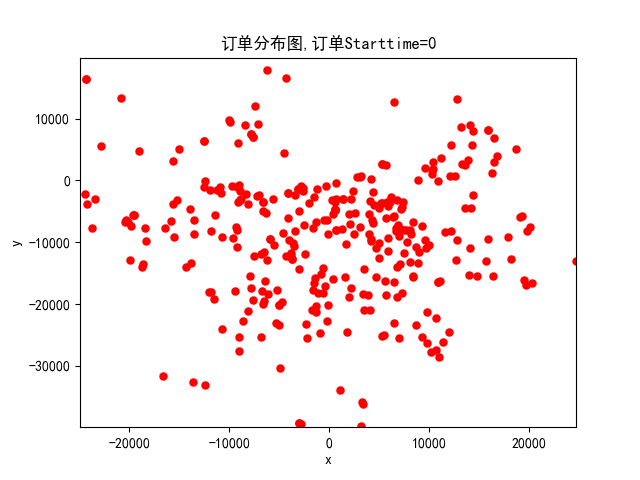
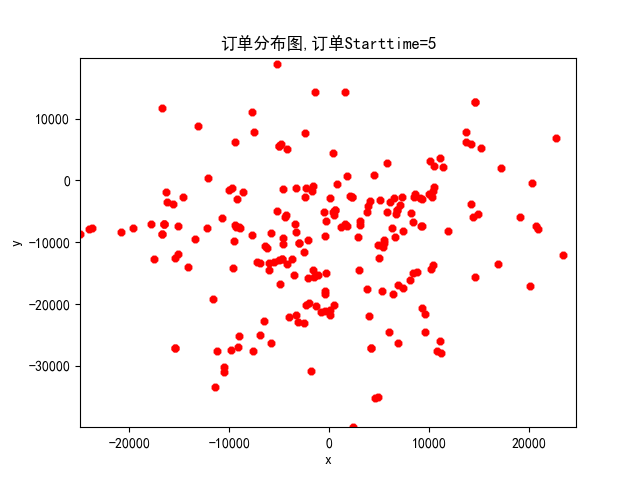
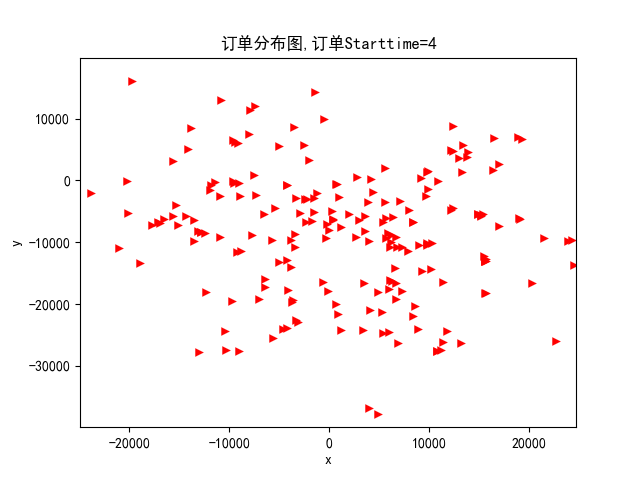
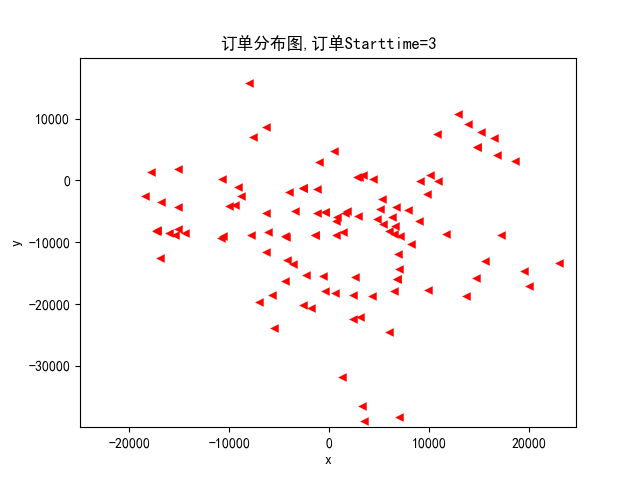
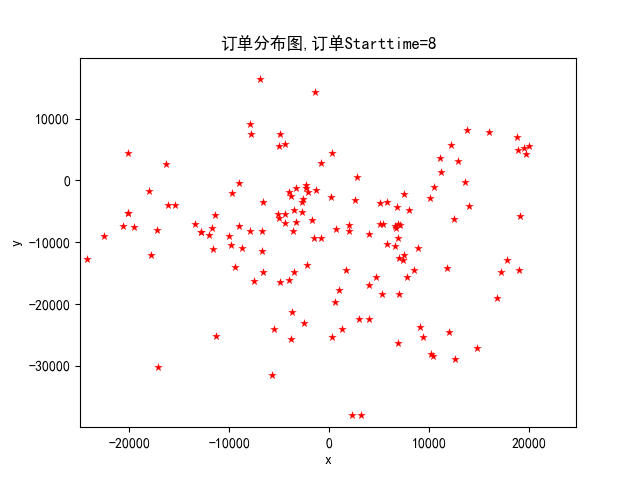
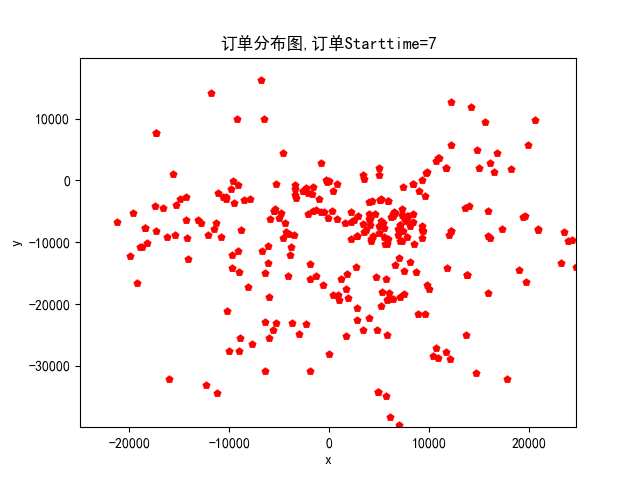
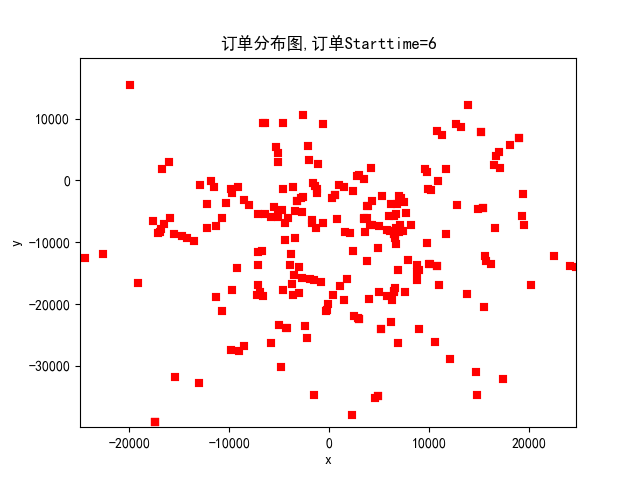


图4 订单坐标分布图

由图3、图4可以看出，订单和阿姨的初始位置重合度很高，有利于我们分配订单，寻找可行解。接下来针对每个时刻的订单位置进行统计分析，绘制图像。根据统计数据，12时刻至13时刻（不包括12时刻）仅有8份订单。因订单数量过小，数据可视化效果不佳，故该时间段内不予绘制订单分布图。图5给出了0时刻至12时刻共13张订单分布图。







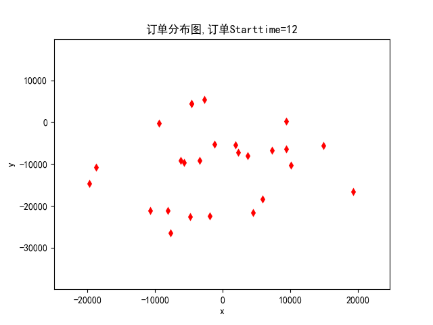
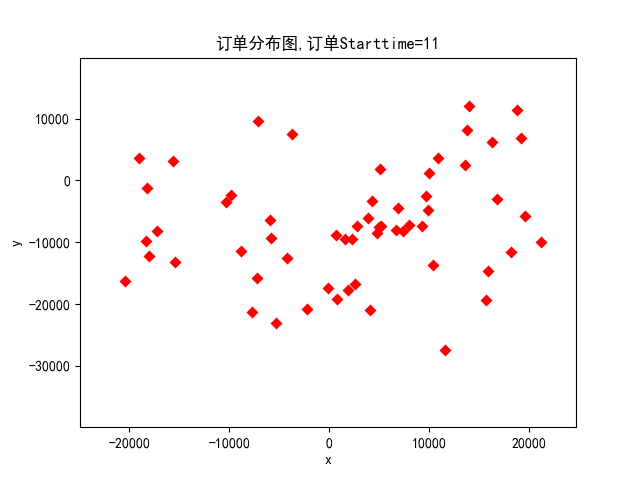
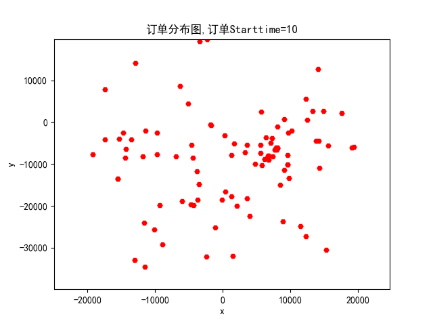
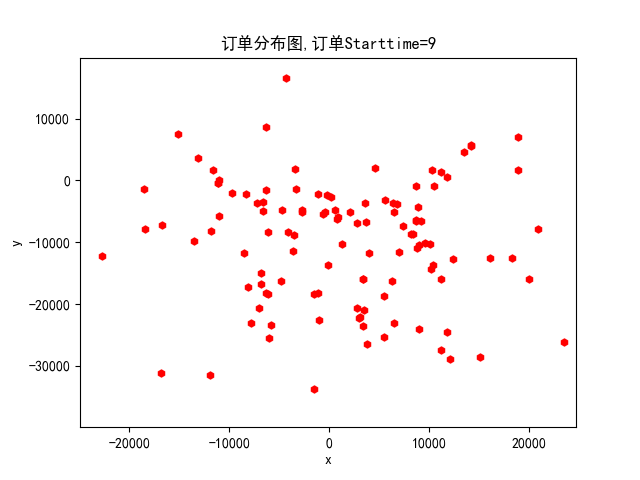


图5 各时刻订单位置分布图

针对数据进行部分处理后，开始着手建立模型。模型优化的目标值受三个因素影响：阿姨的服务分、每个订单的通行距离、订单的服务间隔时间。

我们针对每个订单引入一个新的参数——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。针对每个阿姨同样引入三个新的参数——状态、空闲时刻和是否首次接单。阿姨的状态默认值为“可派遣”；而空闲时刻表示阿姨下一次状态由“工作中”转变为“可派遣”的时刻，默认值为0；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。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。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。上文提到，我们仅需在半点的整数倍时刻分配订单即可，因此，从0时刻开始，每隔0.5小时，我们将此时状态处于开启中的订单和处于“可派遣”的阿姨取出，放入求解器中求解。

因为阿姨的速度为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，我们定义了距离判断矩阵，以此对订单的分配施加约束。此外，因为问题一是在离线批量派单模式下分配订单，对于每位阿姨当天的订单而言，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，但阿姨可以在时刻之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，故我们无需对算法作出更改。

（图？）

但是，若考虑所有订单和阿姨，变量数会达到一个惊人的值，而我们使用的cplex库的求解器拥有1000个变量的限制，为了求解能够进行下去，我们同时采用了两种方案来处理该情况。

**方案一：网格划分，独立求解**

首先，我们将地图划分为行列共个区域，以来表示行列的区域。、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，接取该区域的订单。网格划分完成后，我们针对每一个网格进行独立的订单分配求解，这样距离判断矩阵中阿姨和订单的数量显著下降，cplex的求解器得以正常运行。

（伪代码？）

**方案二：采用“优质阿姨”策略**

另外，为了进一步缩减变量个数，我们可以采用“优质阿姨”策略。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，阿姨只能接取自身所处区域的订单，故可认为该区域内所有阿姨前往同一订单的通行距离相差不大；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数倍于订单数量，易知订单必定能够全部得到分配。此时，我们仅考虑让服务分较高的阿姨接取订单，可在优化目标值的同时，大幅减少阿姨相关的变量数。

当然，这两个方案会在一定程度上削减阿姨的接单范围，限制阿姨的行动，剪去了部分分支，其中也可能包含全局最优解，但此时，我们只能采用该方案完成求解，后续会针对该现象进行改善。

在网格划分过程中，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。若当前区域中阿姨数大于等于订单数，且每个订单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；若阿姨数小于订单数，显然，这种情况下订单不可能全部分配出去。此时，有两种解决方案。

**第一种解决方案：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，若某时刻某区域发生阿姨数小于订单数的情况，可将区域扩大，缩减网格数量。区域扩大以后，会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，直至满足阿姨数大于等于订单数为止。

（伪代码？）

针对该方案，我们先后尝试了多种网格扩大化迭代的方式。设为迭代次数，为当前网格参数，其中，为网格的行数，为网格的列数。我们首先尝试迭代方式一：



使用该迭代方式时，随着迭代次数的增加，网格扩大化的速度也在剧烈增大，不易控制且容易超出求解器的变量数限制。接着，我们尝试迭代方式二：



使用该迭代方式时，相较于方式一，网格扩大化的速度在后期不会剧烈增长，但是前期增速更快，依旧不易控制，容易超出求解器的变量数限制。因此，我们接着尝试迭代方式三：



这次，我们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。

**第二种解决方案：设置订单紧急程度**

由于订单存在服务时间区间，并非必须在最早服务时间分配给阿姨，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以外，其它订单均可保留至下一次分配时刻。对此，我们为订单引入一个新的参数——紧急程度。对于即将超出服务时间区间的订单，我们将其定义为“紧急订单”，此类订单必须在当前时刻被成功分配。当阿姨数小于订单数时，只要每个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。若不满足每个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。

5.2模型建立

**（1）约束条件**

设时刻当前求解区域有份订单，位阿姨。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，且当前时刻订单不一定必须被分配，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。目标函数为：









**（3）混合整数规划模型**





5.3模型求解

本算法首先采用上文中提及的“网格划分”和“优质阿姨”策略，严格限制求解时的变量数，使其满足变量数不超过1000的条件，并利用cplex库中的求解器对已建立的模型进行求解。（求解结果如下：？）

（结果图？）

后来，我们查阅相关资料、专业文献，寻找到一个没有变量数限制的求解器：可在cvxpy库中调用的GLPK\_MI求解器。该求解器同样能够针对混合整数规划问题进行求解，且自身没有变量数的限制，更契合我们建立的模型。下图给出了各求解器对规划类问题的求解能力。

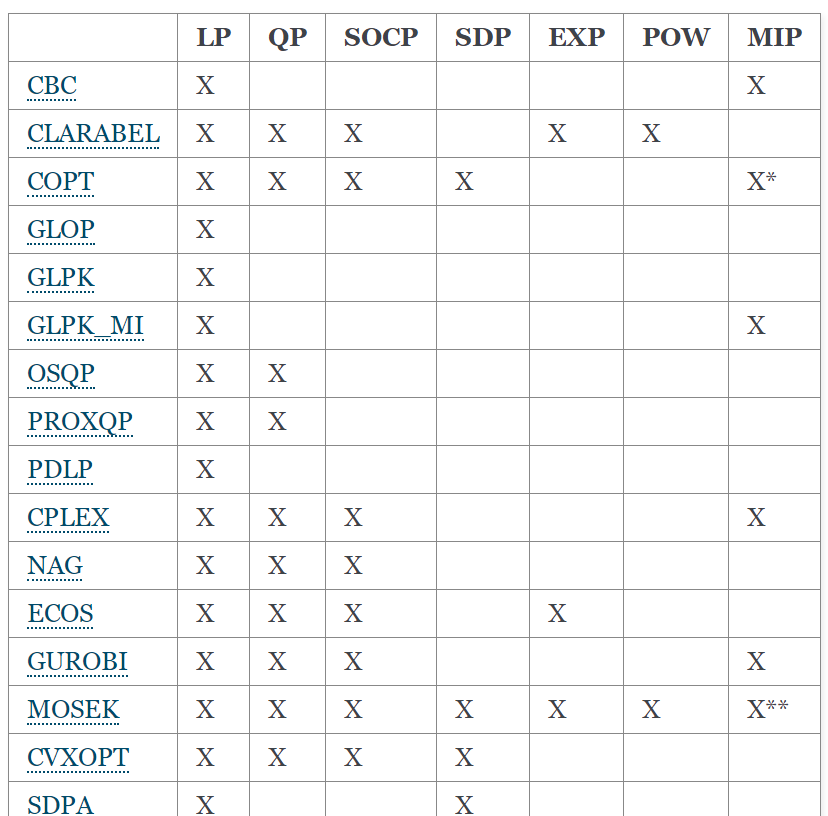


图 各求解器对规划类问题的求解能力

从图中可以看出，GLPK\_MI求解器和之前使用的CPLEX求解器都能解决混合整数规划问题（MIP）。

解除变量数的限制后，为了进一步优化目标值，寻找更优的分配方案，我们设计了一项新的功能——未来规划。因为处于离线批量派单模式，我们可以在当前时刻考虑未来数小时的订单，调整阿姨和订单的匹配方案，以获得更大的目标函数值。为求解器添加新的参量，让求解器考虑当前时刻未来小时内的所有订单。

我们使用GLPK\_MI求解器以逆向递归的方式完成对订单的分配。求解时，针对当前订单的紧急程度进行排序，设置紧急订单数量，默认值为1。首先，尝试求解将所有的订单全部分配出去的方案，其中，紧急订单必须在本次求解中分配出去，其它普通订单可以保留。

若时，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，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；时，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，重新进行求解，直至的值超过订单数，即所有订单均被定义为紧急订单后停止求解；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。每次获得新的返回值后，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。经过数次逆向递归，获得的当前解即为当前网格划分方案下的最优解。上文中提到，将地图网格化、分区域求解会剪去部分分支，影响算法对全局最优解的搜寻，故我们尝试缩减网格数量，将区域扩大，以获得更为优质的分配方案。经分析易知，当网格形状为，即实际上不划分网格时，算法不会剪去任何分支，求解器能够解得全局最优解。（计算结果如下：？）

（结果图）

由上图可以看出，虽然最后我们求解全局最优解时并未采用网格划分策略，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，这些在实际运用过程中是非常重要的参考。不采用网格划分策略能够求出全局最优解，但是时间成本极高，效率低下；采用网格划分策略仅能求出局部最优解，但运行速度快。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。

现基于已设计的算法，对附件中前50份订单和前20位阿姨进行匹配，获得最优的分配方案，并给出阿姨的执行任务列表如下：

（分配列表，一页20行/25行）

（结果分析？）

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：

（轨迹图）

# 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，即阿姨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，我们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，我们在问题一的模型中已有雏形。为了满足线上派单模式的需求，我们对问题一中的模型进行适当修改。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。所有订单中，最晚的最早服务时间为13时刻，这意味着在11时刻，所有的订单都会处于已被分配的状态，无需考虑之后的时刻；而订单中最早的最早服务时间为0时刻，我们至少需要在-2时刻就将这些订单分配出去，此时接单的阿姨仍有至少2个小时的时间移动至订单位置。经分析易知，若时间过早，就算将订单分配出去，阿姨到达后也只能在原地等待，无法进行服务，因此，时间戳下限应设定在-2时刻左右。考虑到可能存在误差，我们决定从-3时刻开始分配订单，-3时刻之前就已下单的订单，无条件进行压单。分析可知，这一行为并不会对目标函数造成影响。综上，我们设置新时间戳。当，即当前时刻晚于下单时间时，我们将订单状态开启，放入求解器中，直至订单被成功分配。

为了满足新的约束条件，我们引入新的压单判断矩阵，订单当前时刻可压单置1，反之置0。对于不可压单的订单，将其加入“紧急订单”，在此次分配中必须被分配出去。

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

相较于问题一的原模型，新模型仅仅对部分模块和功能进行了改动，整体的约束条件、目标函数、数学模型只发生了微小变化。

**（1）约束条件**

设时刻当前求解区域有份订单，位阿姨。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，且当前时刻订单不一定必须被分配，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。目标函数为：









**（3）混合整数规划模型**





6.3模型求解

通过GLPK\_MI求解器对新模型进行逆向递归求解，得出满足约束条件和压单限制的最优分配方案。同样改变网格划分方式，不断缩减网格数量，将区域扩大，最终计算出网格形状为下的全局最优解。（计算结果如下：？）

（结果图）

# 七、模型的评价与推广

7.1模型优点

（待讨论）

7.2模型缺点

（待讨论）

7.3模型推广

（待讨论）

# 八、参考文献

（待定）