#### 1.货物分组策略:

(1)在没有送货的起始节点时，对每一个节点的货物进行组合。使货物组合的重量小于小车的剩余容量，但尽可能贴近剩余容量，即做一次0-1背包。

初始节点送出的货物是这些货物组合而非单件货物。

初始节点可以随意选择。

(2)从初始节点开始的每个节点，都需要考虑捎带另一个节点的货物组合的问题。即每次跳转到另一个节点时，都需要对其余节点都做一次0-1背包规划，找出可能的下一次送出的货物组合。

(2.1)如果配送增加路径/捎带货物重量≤当前配送路径/当前货物重量。

就意味着捎带该货物会提高当次的配送效率，因此一定应捎带配送该货物组合。

(2.2)如果配送增加路径/捎带货物重量≥当前配送路径/当前货物重量

就决定不捎带该货物组合，转而在以后送出。

(2.3)这样产生的问题是: 捎带货物虽然可能会降低效率，但也可能避免了多出一次的从头配送。如第一轮配送的效率是 500m/kg，第二轮配送的效率是10000m/kg，这样即使在第一轮配送中捎带送货使得配送效率下降到750m/kg，但第二次的效率更低，因此这两部分本应该一起配送的。解决方案如下:

(2.3.1)形成两次配送效率差距极大的原因往往是因为: 必须单独配送一个或一小堆远距离的小件。因此在每次配送开始时，可以优先配送增加路径/捎带货物重量最大者，即配送效率最低的节点。以避免因为贪心策略，导致最终形成一系列的货物孤岛。

(2.3.2)除此之外，还可以稍微鼓励捎带，将是否捎带的标准改为“配送增加路径/捎带货物重量≤当前配送路径/当前货物重量\*经验系数”，其中经验系数大于1，从而提高对效率降低的容忍。  
 简而言之，捎带时选择效率最高的，而起始时选择效率最低的。

#### 2.捎带剪枝

每次对所有节点都使用0-1背包的代价可能比较大。我们可以采取两阶段的剪枝策略。

(1)以最理想的状态估算每个节点是否可以带来配送效率的提升，即假设下个跳转节点的货物组合重量就是当前小车送货的容量(如果该节点剩下货物重量大于小车送货容量)或者剩下货物的总量之和(如果该节点剩下货物重量小于小车送货容量)。以此来估算是否可以提升效率，如果不可以就直接跳过该节点。

(2)如果有数个节点在最理想状态下都可以提升配送效率，就取效率最高的前五个节点。只针对这五个节点做0-1背包规划，选取实际效率最高的节点作为下一次捎带的跳转节点。

#### 3.转化时间因素:

当加入时间限制时，小车速度是3000m/小时，即50m/分钟。

(1) 装车时间2min可以等效视为原问题起点与一隐藏起点相连且相距50m，装货发生在隐藏起点而非原问题起点。(从原问题节点到隐藏节点再返回原问题节点即经过了100m,用时2min)

(2) 同理，卸货时间3min可以等效视为原问题节点与一隐藏节点相连且相距75m，卸货发生在隐藏节点而非原问题节点。

(3) 因此时间限制只是需要调整一下地图即可。具体来说，① 所有节点到起点的距离增加50+75=125 m ② 所有节点到节点之间的距离增加75+75=150m



(4)同样地，我们将配送的时间段也转化为距离而非时间段。如，早上9点到12点意味着送货车最多可以行驶3h即9000m，响应地，下午是9000m，晚上是6000m。

这样我们将时间因素除去，统一转化为无时间因素的一般问题。

#### 4.额外时间利用

(1)提前送货

这里需要注意一些细节，以早上时间段为例，送货小车可以在9点之前出发，在9点整到达第一个送货点并完成卸货，从而节省一次装货与一次卸货与一次单程到达时间，即5min+路程时间。提前送货可以保证总是可以发生的(因为早上和中午的时间间距有2小时，足够到达地图的任意位置)。

在转化后的新地图中，可以节省的距离是“从起点到首个节点距离+50m(不需要在返回时，额外走原问题起点到新问题起点的距离)+75m(不需要返回时，额外走原问题节点到新问题起节点的距离)。

提前送货总是可发生的。

(2)滞后返程

类似地，在早上最后一次配送过程中，小车在12点前完成最后一次卸货即可，返程可以发生在十二点之后。类似地，在新的转化的地图后，这意味着可以节省从节点到起点距离-50m-75m。

滞后返程并不总是可行的，需要额外判断。

#### 5.优先级策略

(1)以早阶段送货为例，实际上早阶段可送的货物只分为四类。

①只允许早阶段收货的货物

②允许早午收货的货物  
 ③允许早晚收货的货物  
 ④允许全时间段收货的货物  
 其中①的优先级一定是在早阶段送货时优先级最高的，④一定是最晚的。②和③的优先级还有待讨论。

(2)我们给每个节点也安排优先级，节点的优先级取决于其剩余货物中最高的优先级。这是因为即便一个节点只有一个只允许早阶段收货的货物，都意味着这个节点必须在早阶段经过一次。

(3)当我们选择下一个捎带的节点时，只从优先级最高的节点中选择。且在我们使用0-1背包寻找捎带的货品组合时，对该节点下的货物，按照优先级依次使用0-1背包。以尽可能快地降低该节点本身的优先级。

(4)对于②和③的优先级判断，可以对每个节点统计

可于早/中/晚配送的货物重量和。如一个2kg的早/中配送货物，和一个1kg的全天配送货物。其早/中/晚配送货物重量和分别为3/3/1。当早上配送时，我们便知道早午货物的配送优先级更高(如果不送，中午就要多出来3kg配送)，而早晚货物的配送优先级更低(如果不送，晚上只多出来1kg要配送)。  
 鉴于晚上的时间更短，为2h而非3h，因此晚上的货物的重量权重定为1.5，即实际以3/3/1.5做判断。

#### 6.确定最少用的货车数量

当我们安排路径规划时，以轮转的方式处理，即以一个小车分别经历早/中/晚三次配送。然后进入第二天再经历早/中/晚三次配送，多天运完货品，运完的天数就是最少用的货车数量。

#### 7.解释补充

(1)我们对节点分配优先级，是因为有些节点在特定时间段一定要去，不能不去，在下面部分我们都以早上配货时间段为前提   
 比方说，即使地图上只剩一个只能在早上配货的节点。无论这个节点只能在早上配货的货物重量有多低，送货小车都至少需要过去一次。这是因为题目要求全部货物都必须送达。 通过为节点分配优先级，并按优先级配货，这确保了我们可以以较快的速度降低节点优先级，也就是说我们可以尽快消灭掉那些只在特定时间段才能配货的货物。当我们需要安排更多的小车时，这些小车可以配货的货物对时间段要求就会更低，从而使后续的小车也能充分利用早上，下午，晚上的三个时间段。避免出现由于下午配送货物已经送完，而早上和晚上货物还剩很多，送货小车在下午完全不送货或只送一小部分，浪费掉送货能力的情况。

(2)当我们确定好最高优先级的那一档节点时，我们对每个节点剪枝后，使用0-1背包寻找最大货物组合时，同样有些偏好。具体来说:   
 (2.1) 我们先尽可能地配送高优先级的货物，这是因为我们需要尽可能快地降低节点的优先级，使后续小车配送更灵活。   
 (2.2) 如果我们配送完最高优先级的货物后，该节点还有其他可以装配的货物，就一同组合进该节点的货物组合中，因为这样做百利无一害。所以我们在对第一优先级的货物通过0-1背包塞到小车载重之后，会再对第二优先级的货物以还剩下的容量再次使用0-1背包尽可能地多塞货物。然后是第三和第四优先级依次。   
 (2.3)当我们以多层贪心的方法得到该节点的货物组合时，我们会回到以前的捎带策略中。即配送效率增加最高的节点的货物组合。这一步并没有考虑优先级问题了，是因为我认为考虑进来后可能难以确定一个经验系数。 我给你解释一下这部分问题:  
 假如某个节点A的捎带货物增加路径是30m，捎带货物总重量是32kg(有31kg是第一优先级货物，1kg是第二优先级货物)   
 假如另一个节点B的捎带货物增加路径是32m，捎带货物总重量同样是32kg(但是1kg是第一优先级货物，31kg是第二优先级货物)

这样我们的策略就是配送节点A而非B了。

我认为这确实会有一定影响，但是并不是我们需要过多考虑的问题，因为我们在之前对节点分配了优先级，这意味着节点A和节点B反正都需要访问一次。如果我们一定要访问一次节点A，那A即使只有1kg高优先级货物，剩下空间难道能不装其他货物么？毕竟卸货3min，再加上路程的时间代价还是比较大的。

(3)这里我觉得并不是什么关键的问题，我这里的意思其实就是一遍一遍地送，而不是说一次就考虑3个小车能不能送完，4个呢，5个呢 因为我们每一次配送后，地图中节点的优先级整体就会下降一些，最后一辆小车配送时，地图中可能绝大多数节点的配送级都是最低的那一级。即早上，下午，晚上都可以配送。 一遍一遍地送确保了我们一遍一遍地对地图进行刷新，使送货限制越来越无关紧要。后面的小车就能保证送货能力，最快地送完货。