**在按需充电方式下可以考虑两种情况：**

(1)MCV的任务缓冲池接收到的充电请求为满后随即出发为这一系列需要充电的节点进行服务；

(2)MCV在收到充电请求后随即出发为需要充电的电单车进行充电，在充电的过程中动态选择下一个充电节点。

**针对(1)中的情况：**

对MCV设定缓冲池阈值，MCV从接收到第一个充电请求开始计数，一直到事先设定好的阈值之后，MCV随即开始一轮充电。设MCV缓冲池的最大值与最小值。是指最多可以为几个电单车服务，由于MCV在移动过程中所耗能量要远大于用于充电的能量，因此相当于MCV最远可以移动的距离，。指最少可以为几个电单车进行充电，可以设置一个固定值，阈值计算过程如下：

(参数)

其中，S为不同服务站，指一段采样时间内，服务站接收到的充电请求总数，对于服务站*i*所涵盖的回路中，其区域内MCV的阈值缓冲池大小为：

+

**针对(2)中的情况：**

对电单车设置两个能量阈值，和，，当电单车的剩余能量低于时，电单车发送充电请求给MCV，当电单车剩余能量低于时，发送ALERT信息给MCV，表明自己的能量已经达到警示值，同时停止工作，变为静态点。MCV在接收到第一个ALERT信息之后开始一轮充电。由于要比重要，所以将设定为固定值，通过动态变化求解，通过数值仿真求解，利用单位时间内电单车的平均能量消耗average energy consumption rate(AECR)与平均电单车失效率，这里的失效含义定义为完全没有电量，average failure probability(AFP)两个量以求取。通过数值仿真求取变化过程中AECR与AFP变化情况，可以得到两个近似线性关系：

利用数值仿真中求解，，，

通过最小化AECR和AFP的值作为目标，引入权重因子*w*，采用最小二乘法优化方法中的平方和公式进行优化，如下：

最优是能够最小化上述目标函数的值，其值如下：

**此处需要说明电单车的工作充电机制：**

当电单车的能量达到一定阈值下限后，向服务站发送充电请求，服务站准备调度MCV出发为请求充电电单车进行充电

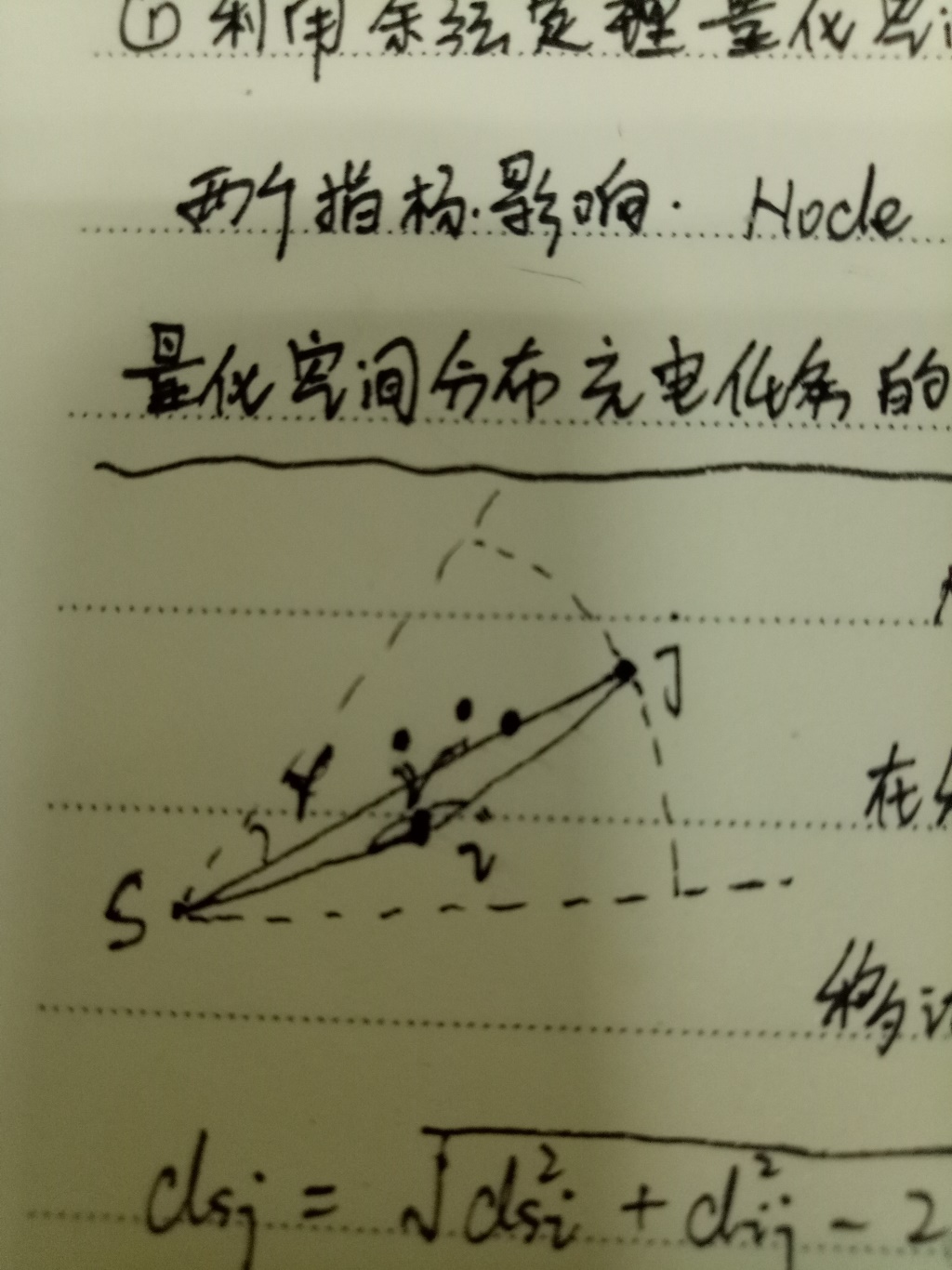
* 针对(1)情况：MCV在准备出发时为所有将要服务的电单车发送CONFIRM信息，接收到CONFIRM信息的所有电单车将停止使用，因此在接收到CONFIRM信息之前的电单车仍然可以被使用；
* 针对(2)情况：MCV每次在确定好将要为之服务的下一个电单车后会给它发送CONFIRM信息，接收到CONFIRM信息的电单车将停止使用。第一种情况是CONFIRM一对多发送，第二种情况是CONFIRM一对一发送。

**按需充电模式下的充电可调度性决策条件**

按需充电方式下路径可调度性的充分必要条件：(1)；(2)；(3)

**电单车选择性插入算法**

情况描述：当MCV选定电单车作为目标充电节点后，在移动至的路途上收到电单车发出的ALERT信息，这说明也急需充电。由于MCV在工作时的能量消耗主要集中于移动能耗以及充电能耗，而充电能耗要远小于移动能耗，所以在考虑电单车选择性插入时主要考虑移动距离。



说明：当MCV从S处出发要行驶向j处为之服务，中途遇到i发出请求，判断是否为i服务的标准是判断si与ij之间的夹角是否接近于180度，因为

因此如果原始是选择从电单车i-j进行充电，中途遇到k，则构建i-k以及k-j的真实路径，可以得到路径长度，原始路径长度i-j的也可以得到，构建三角形ikj，已知三边长度，可以得到任意两边的夹角，判断该夹角是否大于某个阈值角度，如果大于就可以插入，反之就不能插入。

**电单车选择充电算法**

在(1)情况下，每次是已经确定为哪几个节点进行充电，可以直接采用周期性充电方式来做。

在(2)情况下，可能某一时刻有多个电单车为MCV发送充电请求，MCV每次在确定好将要为之服务的下一个电单车后会给它发送CONFIRM信息，接收到CONFIRM信息的电单车将停止使用，而其它发送了充电请求的电单车在发送充电请求后仍可局部使用，当剩余能量在达到后停止使用，固定在当下位置。

**目标**：在第二种情况下时对于一系列发送充电请求的电单车确定为哪一个电单车进行服务。

**需要考虑的因素**：发送充电请求的电单车在过去一段时间内的使用频率k（这是由于在某一时刻接收到的充电请求可能某一电单车是经过了一个月才需要充电，某一电单车是3天就需要充电了，所以考虑充电频率是很必要的）；发送请求的电单车距离MCV的距离d；剩余能量；

**MC出发点depot部署与充电回路分配**

假设：已经构建好充电回路，每个depot有x个charging pile，每个charging pile可服务的回路条数是Y条，那么一个depot至多可服务条充电回路。

Depot就是S位置，其初始位置集合

1）对于每个，构造集合表示可以服务的充电回路个数；

2）对于和，如果，那么可以删除，用来代替。

优化问题需满足条件：

1）每个充电回路至少可由一个MC depot进行负责；

2）每个充电回路只能分配给一个depot。

**第一步：贪心列选择**

目的：选择最小数量的列以保证每一行的总和大于等于1，在这种情况下表明充电回路至少可以由一个depot服务。

步骤：将每一列所有元素相加，称之为该depot所对应的权重，该权重值越高表明这个depot的服务能力越强；2）选择具有最大权重值的一列，移除该列以及该列所覆盖的行，然后更新剩余的行和列；3）再次计算权重值，并重复上述步骤。

**第二步：贪心充电回路分配**

目的：上一步得到的是每个回路可用的depot待选集合，第二步主要用于将每个充电回路分配给一个最佳的depot服务。

步骤：属于depot 的充电回路定义为load()，这称之为depot 的负载，当一条充电回路可以由多个depot所服务时，需要一个优先方法来选择最佳的depot分配给。

假定充电回路可以由depot 进行覆盖，而当前的负载为load()，定义优先级函数如下：

定义|D()|指可以为充电回路服务的depot个数，选择具有最小|D()|值的先进行最优depot选择，这是由于有较小的机率被depots覆盖；然后选择该所对应的depot中具有最大权值的depot分配给充电回路。若两个depot具有相同的权重值，那么将分配给距离充电回路较近的depot。

|  |
| --- |
| Algorithm 3 Depot部署分配问题 |
| 1：**输入**：充电回路集合P，待选部署位置集合D，P()，D()  2：**输出**：depot最终部署位置D3以及所覆盖充电回路  3：第一步：贪心列选择  4：；；；  5： **While** ()  6: ;  6: 将从D1中删除；  7: **for**  8： 将从P1中删除；  9： 依据剩余P1更新D1中的每个值；(相当于更新每一列所对应总和)  10： **End While** |
| 11: 第二步：充电路径分配  12： ；  13： **While**()  14: ；  15： ；  16： 将充电回路分配给depot ；  17: 将已分配depot的充电回路从P2中删除；  18： 更新所有的负载，即load()；  19： ；  20： **End While** |

**充电回路分配**

定义：将充电回路集合，其中z为集合P中包含的充电回路个数，如果可以将这一充电回路集合中的回路分配给一个MCV进行充电，则称该MCV的元充电周期G为所有的充电周期的最大公约数，即，该MCV的负载率ζ为充电回路集合中所有回路的工作时间之和与MCV元充电周期之比：。

充电回路分配算法：

目标：将P条构建的充电回路分配给m个MCV以实现最小化MCV个数的目标。

算法核心思想：将P条充电回路按充电周期的上限进行排序，然后对每条充电回路判断其是否能够划分给当前的MCV，能够划分给当前的MCV需要满足：1）该回路未分配，分配应满足一覆盖原则；2）元充电周期的k倍，即应当包含在充电周期的最大与最小值之间；3）应该保证将充电回路分配给MCV后，MCV的负载率不应超过1。

|  |
| --- |
| Algorithm 4 充电回路分配算法 |
| 1：**输入**：集合P内有若干条充电回路，相关参数  2：**输出**：所需最少MCV个数m，每个MCV所负责充电回路  3：将所有充电回路按照周期的最大值从小到大排序并对其标序号；  4：m=0；//初始化MCV个数；  5：**for** i=1 to |P| **do**  6: label()=0；//label来表示回路是否已经分配MCV了，若分配了则label()=1，反之为0；  7：**end for**  8：**for** i=1 to |P| **do**  9： **if** label()=0 **then**  10： m=m+1；//增加m的数量  11： ；//将分配给；  12： 将的元充电周期设置为第一个分配的回路的充电周期最大值；  13： ；//计算当前分配回路后MCV的负载率；  14： label()=1；//更改的标签属性；  15： **for** k=i+1 to |P| **do**  16： **if** label()=0 **then**  17： **if** 存在整数*l*能够保证在的充电周期的上下限内 **then**  18： ;  19：  **if** **then**  20： ；  21： 将分配给；  label()=1；  22： **end if**  23： **end if**  24： **end if**  25：  **end if**  26：**end for** |