1. **充电回路构建**
2. 周期性充电

周期性充电：每次为分区内所有的电单车都按一定次序进行充电，假设分区中的电单车个数为N；

基本思想：（1）充电序列确定；（2）构建Hamilton圈；（3）充电可调度性条件；（4）充电回路子路径构建

（1）充电序列确定

提出充电回路构建问题：该问题的目标是构建最小数量的充电回路，NP-难问题，这是由于该问题包含了DVRP问题，而DVRP问题是NP-难问题。

将所有电单车按照一定次序排列成序列，然后再在其上构建Hamilton回路

注：**A charging schedule** determines when does the MC charge which sensor nodes for how long in what sequence.

**A charging scheme** gives several principles to rule the charging activities.

**The charging mission** in a WRSN is to schedule the charging activities of the MC to ensure that all sensor nodes will not exhaust their energy.

When applying **a charging scheme** on different **charging missions** or even on the same charging mission, one can obtain **different valid charging schedules.**

**A charging schedule is called valid** if it ensures that no sensor node will die.

**A charging mission** is schedulable (or can be scheduled) if there exists a valid charging schedule, and is unschedulable (or cannot be scheduled) if there is no valid charging schedule

（3）充电可调度性决策条件

MCV到达电单车i的时间如下：

由于在求解depot部署位置的时候，其待选部署位置是一系列电单车的位置，因此此时假设MCV从任一电单车处出发是合理的，忽略了depot距离第一个访问的电单车之间的距离，可MCV到达节点i与该轮充电开始时间之间的关系如下：

此时考虑的周期性充电是在一轮充电周期中为分区内所有的节点进行充电，假设白天是不进行电量补充的，每天晚上进行一轮充电，每次为节点充满电（每次根据节点白天的使用频率确定充电量（自适应充电量）再确定），在此充电过程中节点全部为静态点且不消耗能量。

节点能量约束条件为：

这样在新的一天开始时节点的能量都是相同的，即每一轮调度周期开始的时候的能量都是相同的。在一轮调度周期中节点消耗的能量与最终补充的能量是相同的，如下所示：

其中，充电周期为，其中是指MCV返回服务站后补充能量等待下次出发的时长。

MCV的能量限制为：，其中

由于MCV返回depot后要为自身进行能量补充，所以整个充电周期T的长度要大于为需要充电的N个节点充电所花时间，如下所示：

将该式子与相结合可以得到如下：

因此，上面的式子是充电周期T的一个下限，如果，则MCV无法返回depot

将与相结合在一起，可以得到如下MCV构建充电的能量约束：

上述式子需要同时满足，可以将上述式子转化为：

综上所述，得到周期性充电方式下路径可调度性的充分必要条件：(1)；(2)

|  |
| --- |
| Algorithm 确定充电回路R是否可调度 |
| 1：**输入**：N，，，，η，R={}，；  2：**输出**：充电回路R是否可调度  3：应用A\*寻路算法确定N个电单车之间的最佳路径；  4：计算路径长度D；  5：收集N个节点的最大，并计算；  6：**If**满足周期性充电方式路径可调度性条件**then**  7： **return**充电回路R可调度；  8:  **else**  9:  **return** 不可调度；  10: **End If** |

|  |
| --- |
| Algorithm N个电单车的充电子回路构建 |
| 1：**输入**：N个电单车，  2：**输出**：充电子回路及其个数值z；  3：z=1，i=1；  4：应用A\*寻路算法确定N个电单车之间的最佳路径，构造无向图G(V,E)；  5：针对G(V,E)应用蚁群算法得到N个电单车的最短哈密顿回路(起始结束于)；  6：从开始顺时针标志最短哈密顿回路中的电单车分别为，；  7： **While**  do  8： **for** k=i to N **do**  8:  将加入中；  9: 应用可调度性判定算法判定的可调度性；  10:  **If** 是不可调度的 **then**  11: 将从回路中取出；  12: i=k；  13: z=z+1；  14:  **break；**  15: **End If**  16: **End for**  17: **End While** |

注意：最初构建的最短哈密顿回路可以按照电单车的功耗由高到低（或由低到高）的顺序生成H，则最终得到的各个子回路的电单车功耗相近，因此不存在瓶颈电单车，从而MCV为每个回路充电的效率也更高。

**2)按需充电方式**

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。

**存在问题：**

* 第一种按需充电方式可能存在问题：等待任务缓冲池为满后才出发为需要充电的电单车充电在工作负载较小的情况下，可能导致部分需要充电的电单车充电延迟较大。
* 第二种按需充电方式可能存在的问题：MCV在接收到充电请求后随即出发为需要充电的电单车充电，会导致充电过程较为频繁，可能出现刚刚返回服务站就需要再次出发为电单车充电，而且可能请求的电单车数量仅有几个，会降低各个充电单车的充电延迟，但是会导致资源浪费，MCV的充电效率较低。

**解决方法：**针对以上两种情况，分别对MCV设定缓冲池阈值或为电单车设置能量阈值。

* 针对第一种情况，对MCV设定缓冲池阈值。该阈值在不同分区中是不同的，电单车使用量较频繁的区域的阈值较大，而使用量较小的区域阈值较小，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。

**解决方案**：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：

其中，S为分区数，指一段采样时间内，服务站接收到分区*i*的充电请求数，对于分区i，其区域内MCV的阈值缓冲池大小为：

+

* 针对第二种情况，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。

**解决方案**：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，，，

然后利用多目标优化的方法来优化的值[1]

[1]M.Ehrgott, Multicriteria Optimization, second ed., Springer, 2005.

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

* 针对第一种情况：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；
* 针对第二种情况：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**

在按需充电模式下，无论第一种情况或者第二种情况下，在电单车收到CONFIRM信息之前仍会消耗能量，一轮充电的充电周期T如下：

在第一种情况下，缓冲池的大小就为，n=；

在第二种情况下，不考虑缓冲池大小，MCV接收到n个充电请求。

MCV到达电单车*i*的时间如下：

整个充电周期可以变化为：

为了保证电单车的正常工作，应当保证电单车在被MCV服务之前剩余能量不会变为0，需要满足以下公式，从发送充电请求开始到被服务之间能量不会变化到0，考虑最差的情况，即在这段时间内电单车*i*一直在被使用，只要满足在这种情况下剩余能量不会变为0即可：

充电过程中是不消耗能量的，充电过程中的能量变化如下所示：

为保证充电公平性，在每一轮充电过程中，服务的电单车仅被服务一次，因此需要保证在下一轮充电开始前，这个电单车的剩余能量不会达到0：

将代入上式：

将上述式子延伸至整个充电周期，可以得到下面的式子：

在整个充电周期中，包含MCV的移动时间以及充电时间，所以可以得到如下：

所以一定可以得到的是

为保证电单车的剩余能量不会等于0，因此在被服务前所消耗的能量一定不能超过，即：

将与相结合，可以得到

结合和可以得到

由周期性充电推知的T的下限为，所以需要满足

即，只要满足大于最大值即可，所以，这个条件为

此外，也需要满足MCV能够有足够的能量返回服务站，如同周期性充电，MCV需要满足如下条件：

综上所述，得到按需充电方式下路径可调度性的充分必要条件：(1)；(2)；(3)

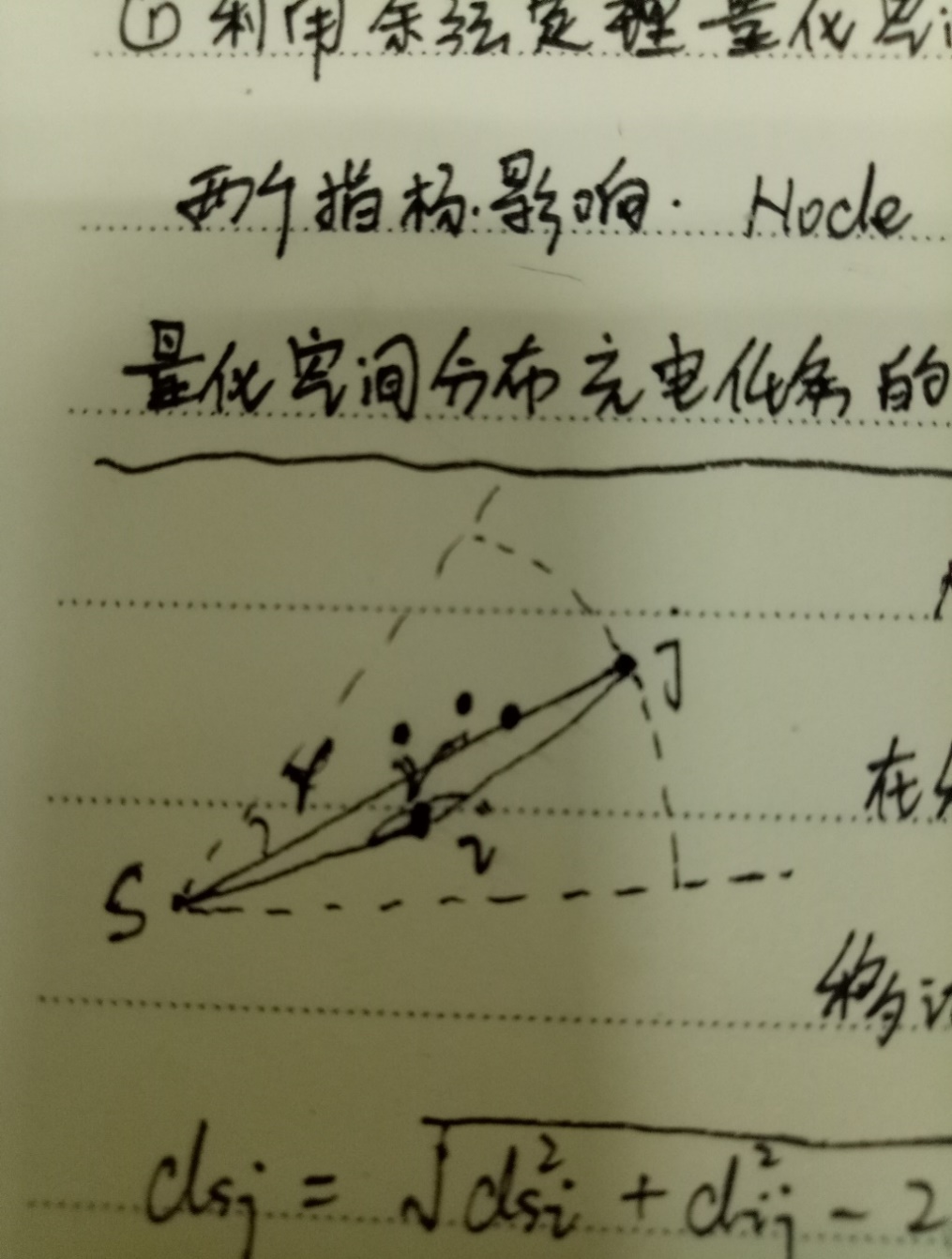
针对第一种情况下，与周期性充电的差别在于周期性充电每一轮充电过程中的电单车以及MCV行走路径是相同的，而第一种情况的按需充电，确定完接收到的发出充电请求的电单车位置后按照周期性充电方式构建这些电单车的充电路径，每次发出充电请求的电单车是不同的，因此，每次构建的回路以及MCV行走路径也是不同的。

针对第二种情况的按需充电方式，还需要(1)确定每次充电的节点是哪一个；(2)电单车选择性插入算法（同样也适用于第一种情况）。

**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以及电单车的重要性（如果接收到充电请求，但是这个电单车距离上次充电时间很长，说明这个电单车并没有那么重要，因此也将不必耗能去为之充电）（暂时先不考虑）

解决方法：利用余弦定理，如下图所示证明过程：



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**

在第一种情况下，每次是已经确定为哪几个节点进行充电，所以直接构建这几个节点的最短哈密顿回路采用周期性充电方式解决即可。

在第二种情况下，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；

**算法核心思想**：对每个发送充电请求的电单车构建特征三元组（k，d，），然后利用skyline query选择最佳的电单车进行服务，每个需充电的电单车的特征三元组在其固定为静态点前为不断更新的过程，每次的更新发生在MCV为一个电单车服务完在选择下一个服务电单车时。

1. **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。

假设剩余节点或其附近区域都适合部署MC depot,所有剩余节点的位置都归属于初始位置集合

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。

优化问题定义：

已知一系列充电回路P以及一系列depot待选部署集合D，找到最少数量所需的depots部署区域以保证每一个充电回路只能由一个MC depot服务。

Subject to

表示如果充电回路分配给了depot ，那么，否则为0

这是一个**二元线性规划问题**，是一个NP-难问题，基于贪心策略的启发式算法两步进行解决

**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** |

服务站部署可以对比胡诚的博士毕业论文第55页。

**充电回路分配**

根据胡诚论文可知元充电周期与MCV负载率的定义如下：

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：。

定义MCV的元充电周期：在MCV的一个元充电周期中包含若干轮充电调度，每轮充电调度为一个充电回路中的所有传感器节点充电一次。

暂时考虑两条充电回路是否可以由一个MCV服务的情况：

1. 对于充电回路，MCV为其服务时的工作时长为：
2. 确定MCV的元充电周期
3. 考虑与发生的相对关系，当在内变化时，MCV负载率的变化有如下情况：

* 当时，，因此，可以求得MCV的负载率为:
* 当时，，因此，可以求得MCV的负载率为：
* 当时，G暂时无法确定，就先用G代指，因此，可以求得MCV的负载率为：

因为G为和的最大公约数，那么当时，G一定满足小于，所以可以得到三种情况下的MCV负载率的大小关系如下：

所以当时，MCV的负载率取到最小值；当时，MCV的负载率取到最大值。以上所考虑是在情况下，推广至多倍情况，即在情况下，，以上结论同样成立，即当时，MCV的负载率取到最小值，该值为；当时，MCV的负载率取到最大值，最大值为：。

很显然，对于z条充电回路，当时，这z条充电回路可由同一个MCV进行服务，这也就意味着当MCV的负载率时可为这z条回路服务。因此，我们需要使MCV的负载率小一些，由上述描述可知，当增大时，负载率会减小，因此我们将周期值取到上限值时可以得到最小的负载率，进而最大化各充电回路分配给同一个MCV的可能性。

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** |

**说明**：对于周期性充电方式，只需进行一次计算就可以得到最少所需的MCV数量，对于按需充电方式，按需充电方式每次都是一样数目的电单车个数，只是位置不同，这样可能导致每次生成的充电回路的个数是不同的，最终所需的MCV个数也是不同的，暂时考虑的是多实验几次取平均值。