## 任务调度

### 1 任务调度简介

RCS获取wms生成的未执行任务（排除没有fromStation、toStation的不合格任务），获取所有可用小车，并根据车型、任务优先级、任务到车辆的距离等因素计算出相对较优的分配方案，将任务分配到所有小车的过程。 具体可分为两个部分，一个部分是预处理部分，获取任务后先对一些绑定的任务、配置好的任务与车辆关系的任务做预处理操作，对车辆故障情况记录及更新等，任务预处理包括如下三个方面：

（1）筛选可以接任务的小车。只有同时满足以下条件的小车才会参与任务分配：小车在线，不是低电量，小车可调度，系统有小车的当前站点信息，没有故障和异常。

（2）车辆绑定工位。某些工位生产节拍快 ，可通过绑定的方法固定车辆的方法让小车优先接该工位的任务。

（3）多个任务依赖绑定。 有些任务与任务之间有依赖，执行完当前任务后需要做后续任务，可以将任务绑定在一起。

（4）分区域接任务。车辆绑定区域，车辆只能接指定区域的任务。

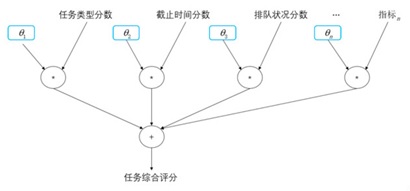
另一个部分则是算法部分，通过迭代进化的机制，综合考虑任务优先级、任务到小车距离、任务创建时间、工位的节拍及车辆故障等情况计算出全局较优的分配结果，提高车辆利用率，降低空载率 。

### 2 优先级评分机制

对于每一个任务，都可以根据其指标属性给出一个综合打分，来评价其优先程度；分数越低，优先级越高。

对于单个指标，限定每个任务经过归一化处理后对应分数在0-1之间，且可配置。比如配置入库任务分数0.5，出库任务分数0.8，则代表出库任务先于入库任务。

不同指标同样存在优先考虑顺序，一种简单的方法是为不同指标的评分乘以一个具有显著区分度的权重，得到每个任务加权后的综合分数，如下图所示。



### 3 调度算法设计

因为每个工厂的客户要求、生产节拍不同，任务量及车辆数目也有差异，因此针对不同的情况制定不同的调度策略，本方案三种不同情况采用不同的调度策略，用两种不同算法来求解。

一种是贪心算法，贪心算法不需要考虑全局，只需考虑局部最优。适用于两种情况的求解，第一种是任务量少（大部分时间任务数量少于车的数量）的情况， 这种情况根据任务来选择车辆，即遍历所有任务，分配给最近的小车即可，另一种情况是任务数量比较多的情况（任务数量大于车的数量），每辆小车就近接任务即可。

另一种是基于全局优化的遗传算法，适合大量任务的车辆比较多的情况，需要根据任务类型，搬运时间，拥堵情况等诸多因素迭代运算，得出较优结果。

任务数量比较多的情况：

该情况采用贪心算法进行求得局部最优解，是算法具体步骤如下：

步骤一：筛选所有小车，选择不是低电量，没有故障且不上收到状态的小车。

步骤二：遍历所有可用车辆，计算小车接每个任务的时间T=S/V,距离S计算采用曼哈顿距离 h(n)＝|x1-x2|+|y1-y2|，正在执行任务的小车也要加上当前正在执行的任务的时间。

步骤三：对所有任务的接取时间和优先级做归一化处理后取倒数，得到一个综合评价值。

步骤四： 比较所有任务对应的评价值，选择评价值最高的任务分配给当前小车。

任务数量比较少的情况：

该情况采用贪心算法进行求得局部最优解，是算法具体步骤如下：

步骤一：筛选所有小车，选择不是低电量，没有故障且不上收到状态的小车。

步骤二：任务根据创建时间排序，遍历所有任务。

步骤三：计算每辆小车接任务的时间T=S/V,距离S计算采用曼哈顿距离 h(n)＝|x1-x2|+|y1-y2|，正在执行任务的小车也要加上当前正在执行的任务的时间。

步骤四： 比较所有任务对应的接取时间，选择评价值最高的任务分配给当前小车。

任务和车辆都多的情况：

该种情况需构建优化模型，确定合适的目标函数，考虑相关约束，具体求解适合采用迭代优化算法（粒子群算法，模拟退火算法，遗传算法）进行求解， 本方案采用遗传算法对模型进行求解。

1、术语表

基因型(genotype)：染色体的内部表现；

表现型(phenotype)：染色体决定的性状的外部表现，或者说，根据基因型形成的个体的外部表现；

进化(evolution)：种群逐渐适应生存环境，品质不断得到改良。生物的进化是以种群的形式进行的。

适应度(fitness)：度量某个物种对于生存环境的适应程度。

选择(selection)：以一定的概率从种群中选择若干个个体。一般，选择过程是一种基于适应度的优胜劣汰的过程。

交叉(crossover)：两个染色体的某一相同位置处DNA被切断，前后两串分别交叉组合形成两个新的染色体。也称基因重组或杂交；

变异(mutation)：复制时可能（很小的概率）产生某些复制差错，变异产生新的染色体，表现出新的性状。

编码(coding)：DNA中遗传信息在一个长链上按一定的模式排列。遗传编码可看作从表现型到基因型的映射。

解码(decoding)：基因型到表现型的映射。

个体（染色体）（individual）：指染色体带有特征的实体；

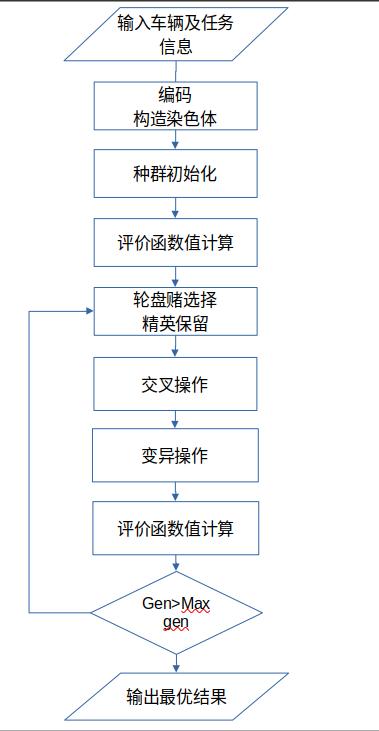
种群（population）：个体的集合，该集合内个体数称为种群

归一化：利用个体的目标值与最大目标值的比例，将目标值投射到（0，1）范围内，从而消除量纲，将多目标问题转化为单目标问题。

任务取货时间因子：用于评价小车取到货物的时间。

优先级因子：用于评价优先级要素，目标是让优先级高的任务先执行。

3、算法流程图



4、遗传算法相应操作的作用及实现方法

(1) 相应操作的作用

编码：将表现型（任务与车辆）映射成基因型（实数），便于交叉对其及进行交叉变异操作。采用实数编码 编码序列举例：12345｜3124

其中前半部分为任务序列，后半部分为车辆序列

轮盘赌选择：优胜劣汰。将评价值高的个体保留，淘汰评价值低的个体。

种群初始化：随机得到个体的集合，是后续迭代进化的基础。

评价值函数：用于评价个体的优劣，通常适应度值越大，个体越好。

交叉操作：产生不同个体的关键操作。

变异：增加后代的多样性，跳出局部最优解，防止过早收敛。

逆转操作：在现有种群个体的基础上寻找更优的个体。

精英保留策略：将父代的优良个体保留到下一代，保证种群平均适应度值不断增大,精英个体能够代代相传。

(2) 实现方法

编码设计：采用实数编码，染色体（编码串）由两部分组成，一部分由任务序列构成，另一部分由每一辆车对应的任务数构成。染色体长度任务数量与车辆数之和。

比如，染色体 “1 2 3 4 ｜2 1 4 3 ” 表示将第2 个任务分配给第 1 辆车，第 1个任务分配给第 2 辆车，第 4 个任务分配给第 3 辆车，第 3 个任务分配给第 4辆车。

种群初始化：采用随机生成的基因序列构成个体，种群由随机个体组成。

评价值计算：取目标值的倒数（目标函数值越小评价值越高），目标函数由两部分构成一部分是任务完成时间，另一部分是优先级因子， 由优先级评价机制计算得到，包括优先级、任务创建时间、拥堵时间、超时惩罚时间。

轮盘赌选择：将适应度值累加并除以总的适应度值，累计的适应度值与总适应值的比值与（0，1）之间的随机数进行比较，如果累计适应度值大于随机数则选择当前个体，否则继续累加适应度值，直到累计适应度值大于随机数，选择该个体。

任务序列、车辆任务数交叉操作：采用部分匹配交叉，用于产生新的个体。父代1与2交叉，保留父代1和2染色体中的自身的部分基因，其它基因来自对方的。

变异操作：不定向变异，增加后代个体多样性。随机交换染色体中的基因。

逆转操作：随机选择两个基因位，将两个基因位之间的基因逆序。逆序完后计算适应度值，适应度值比原来大的个体替换原来个体，否则不进行替换。

精英个体保留：获取父代个体适应度值最大的个体，不参与子代的交叉，直接替换掉子代的最差个体。