

# 大作业-物流系统仿真案例报告

## 1.问题描述

考虑一个生产车间，该车间简化后的平面布局图如下（完整布局图见附件）。车间共分为生产区、预装配区、原材料库、AGV 充电站、产品上线区共 5 个区域，生产开始前，AGV 从车间的右上角发车，需前往原材料库将生产所需原材料运送至生产区进行生产，生产完成后的成品需经检查站检验后，再由 AGV 其运送至产品上线区进行上线。

车间中，有部分原材料无法直接参与成品加工，需进行预装配，此时需由 AGV 将原材料先运至预装配区进行加工，待加工成预装配件之后，再将该预装配件从预装配区运送至生产区的相应工位进行生产。车间还设有 AGV 自助充电桩，当小车电量低于设定的保留电量值时，小车可自行前往 AGV 充电桩进行充电。

本报告拟对该车间整体的作业流程进行仿真，以确定所需 AGV 小车的最佳数量，并进一步改进生产流程。

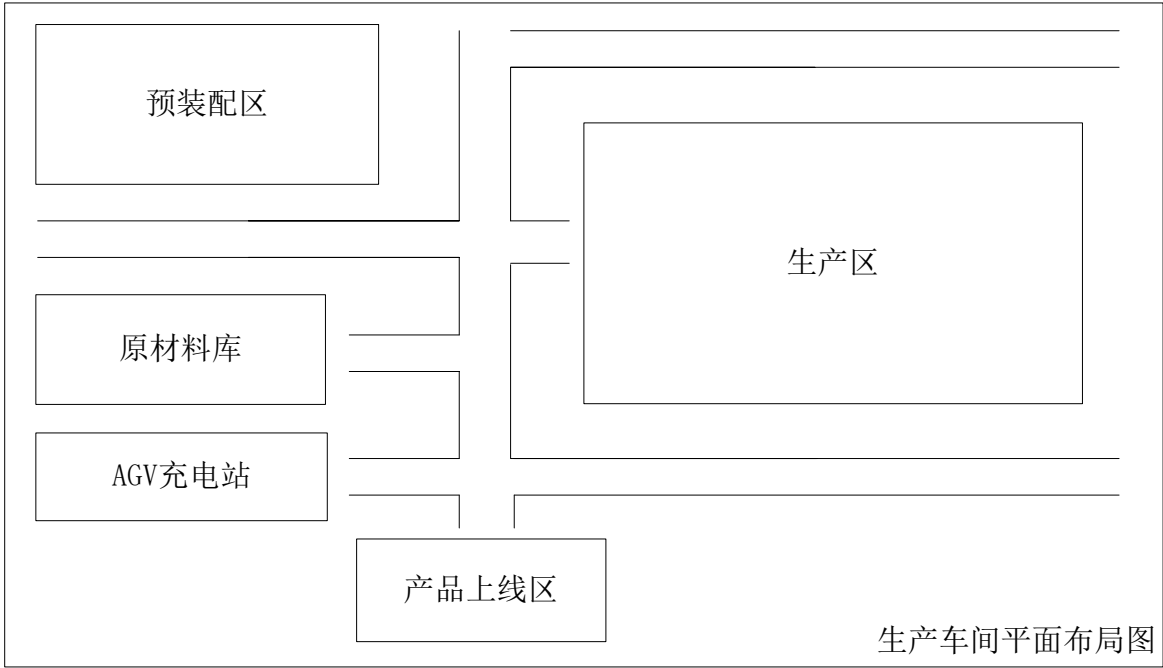


图 1 车间平面布局简图

## 2.实体对象与参数设计

该生产车间及其相关作业需要用到的实体设备均可在 plant simulation 的实体库中找到对应的仿真对象，主要设备的相关对应关系如下表：

表 1 实体设备及其对应的对象

实体设备	plant simulation 实体库对象
AGV	Transporter
AGV 行驶轨道	Track、TwoLaneTrack
生产线	Converter
立体货架	Store
充电桩、检查站	Station
装配工位	AssemblyStation
大、小料箱	Container

### 相关参数设计：

除下文提到的参数外，本报告涉及的其余参数均为系统默认值。

（1）AGV 电量。AGV 的初始电量设置为 2~50mA 之间服从均匀分布的随机值（z\_uniform(2,50)），保留电量设置为 10，即当小车电量低于 10mA 时，小车需前往充电桩进行充电。

（2）生产区加工作业时间。该车间所有类种类间均设置为 180s，预装配区的装配工位加工时间维持默认值。

（3）各装配工位加工所需的各类原材料的数目根据所给的 excel 表格设定，表格中未给出具体数值的零件均设置为 1。

（4）大、小料箱中装有的原材料类型和数目根据所给的 excel 表格设定，表格中缺失具体数目的零件一律设置为 1，预装配件单独装了一个料箱，数目设置为每种 280 件。

（5）车间现有布局一共有 3 个预装配工位、6 个充电桩，2 条相同的生产线，每条生产线上有 6 个不同的加工工位，用于生产不同类型的产品。实际生产时，可自行设定实际使用多少个充电桩、生产线等。

## 3.仿真模块设计

### 3.1 整体模块设计

根据所给的车间平面布局图，搭建出本次仿真模型的 2D 布局图如下：

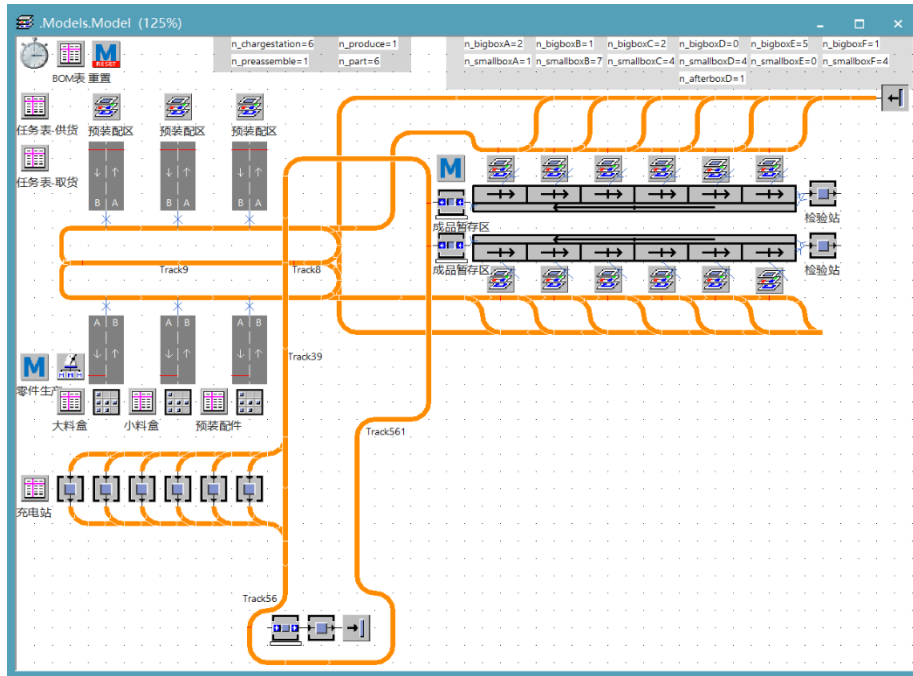


图 2 仿真模型布局图

在该模型中，小车从右上角的 source 中产生，同时设置其目的地为预装配区与原材料区中间的轨道 Track9，Track9 的合适位置设有已传感器，当小车触发传感器时，首先判断成品暂存区有无待上线的成品，若有，先将成品上线；接着判断预装配区有无待取的预装配件，若有，则将待取零件运至对应的装配工位，若无，预装配区向小车发出供货指令；最后，判断生产区装配工位有无可供装配的原材料，若无，生产区向小车发出供货指令。小车接收指令后，前往原材料库中取货，运往预装配区或生产区进行生产加工若小车暂无任务可供指派，则设置其目的地仍为 Track9，让其绕着轨道空跑。

```

Models.Model.Track9.OnSensor1
--判断成品区和预装配区是否有货物需要取货，若有货，先取货
if DataTable8.ydim > 0 and @.empty
  y := DataTable8.ydim
  for var i := 1 to y
    s := obj_to_str(DataTable8[5,i])//Models.Model1.preAssemble1
    --成品区
    if DataTable8[3,i] = false and (DataTable8[1,i] = "finalProduct1" or DataTable8[1,i] = "finalProduct2")
      @.destination := str_to_obj("Track561")
      DataTable8[3,i] := true
      DataTable8.cutrow(i)
      //@.stopped:=false
      return
    --预装配区
    elseif DataTable8[3,i] = false and DataTable8[1,i] = "Buffer5"
      s := "TwoLaneTrack"+strcopy(s,26,1)
      @.destination := str_to_obj(s)
      DataTable8[3,i] := true
      DataTable8.cutrow(i)
      //@.stopped:=false
      return
    end
  next
end

```

图 3 Track9 传感器控件方法（部分）

Track9 之前的 Track8 也设有一个传感器，当小车触发该传感器时，会判断小车的剩余电量是否低于保留电量，若低于，小车前往充电站充电。当经过一段时间，成品暂存区出现待上线的成品时，暂存区向小车发出指令，小车接收指令后，将成品运至上线区。

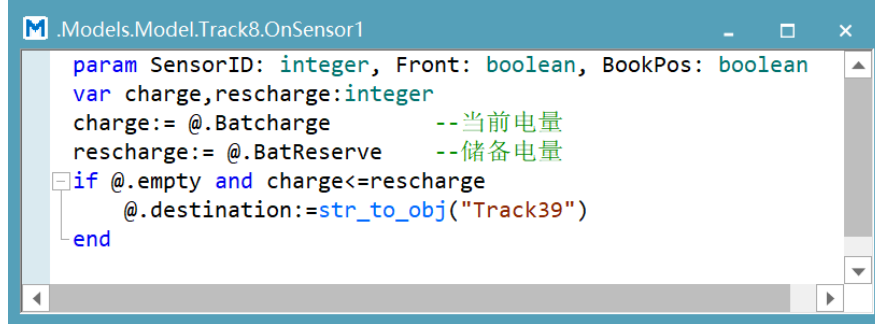


图 4 Track8 传感器控件方法

模型设计了取货和供货两个任务表，当任意区域的暂存区需要 AGV 前来取货或者供货时，该暂存区将会对应的任务表中写入取货或供货需求，当小车行驶至 Track9 并触发传感器时，传感器控件方法会挨个扫描任务表，并根据任务表给小车指派对应的任务，当某一任务指派给小车后，会删除任务表中对应的任务。

在模型生产区附近，每一个靠近装配工位的 Track 均设置有一个传感器，用于控制触发该传感器的 AGV 小车将料箱卸入各个工位。除 D 工位处的 Track 外，其余 Track 均调用 m\_in 方法，TrackD001 则调用其内部创建的 OnSensor1 方法。

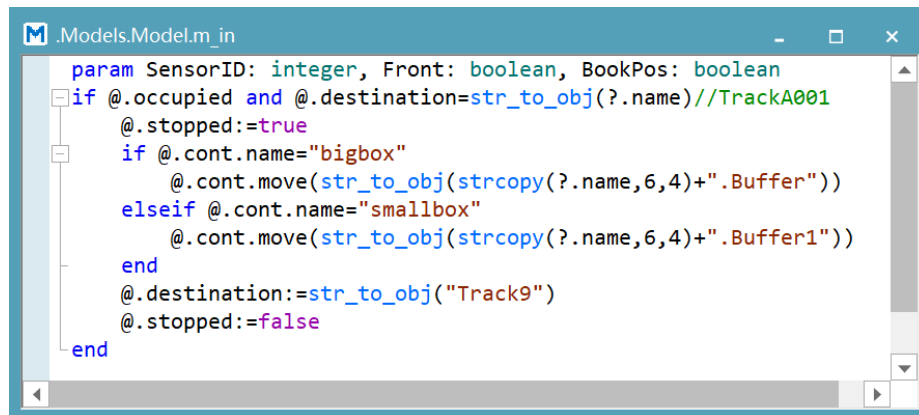


图 5 m\_in 方法

同理，在靠近成品上线区的 Track56，以及靠近成品暂存区的 Track561 处，也设有传感器，分别用于控制成品的上线和装车。

### 3.2 初始化和重置模块设计

模型中插入的 reset 方法可以对模型进行初始化，当点击重置按钮时，该方法执行一次。本模型中，reset 方法的目的是初始化任务表并往表中写入初始任务。

```

.Matlab.Model.reset

--清除所有MU
EventController.DeleteMUsOnReset := true

--清除所有表格内容
DataTable.delete
DataTable6.delete
DataTable8.delete

var n:integer

--初始化充电站记录表
for var i := 1 to n_chargestation
    DataTable[1,i]:=str_to_obj("chargeStation"+to_str(i))
    DataTable[2,i]:=false
    DataTable[3,i]:=EventController.simTime
    DataTable[4,i]:=EventController.simTime
    DataTable[5,i]:=DataTable[4,i]-DataTable[3,i]
    DataTable[6,i]:=0
next

--初始化预装备区任务表

```

图 6 reset 方法（部分）

模型将启用的充电桩数目、预装配工位数目、生产线数目、装配工位数目以及各装配工位配备的大、小料箱数目均定义为全局变量，通过修改这些全局变量，可以模拟该车间启用不同数量的物流资源时车间可能发生的各类情况，更方便地方便进行瓶颈分析和优化。

表 2 全局变量及其含义说明

全局变量名	含义
n_chargestation	启用的充电桩数目
n_preassemble	启用的预装配工位数目
n_produce	启用的生产线数目
n_part	生产线上启用的装配工位数目
n_bigbox(A/B/C/D/E/F)	装配工位配备的大料箱数目
n_smallbox(A/B/C/D/E/F)	装配工位配备的小料箱数目
n_afterbox(A/B/C/D/E/F)	装配工位配备的预装配件料箱数目

综合上述整体模块及初始化和重置模块的设计，结合该车间生产作业的全流程分析，模型整体框架已初步搭建完成，可绘制出具体的仿真逻辑流程图如下：

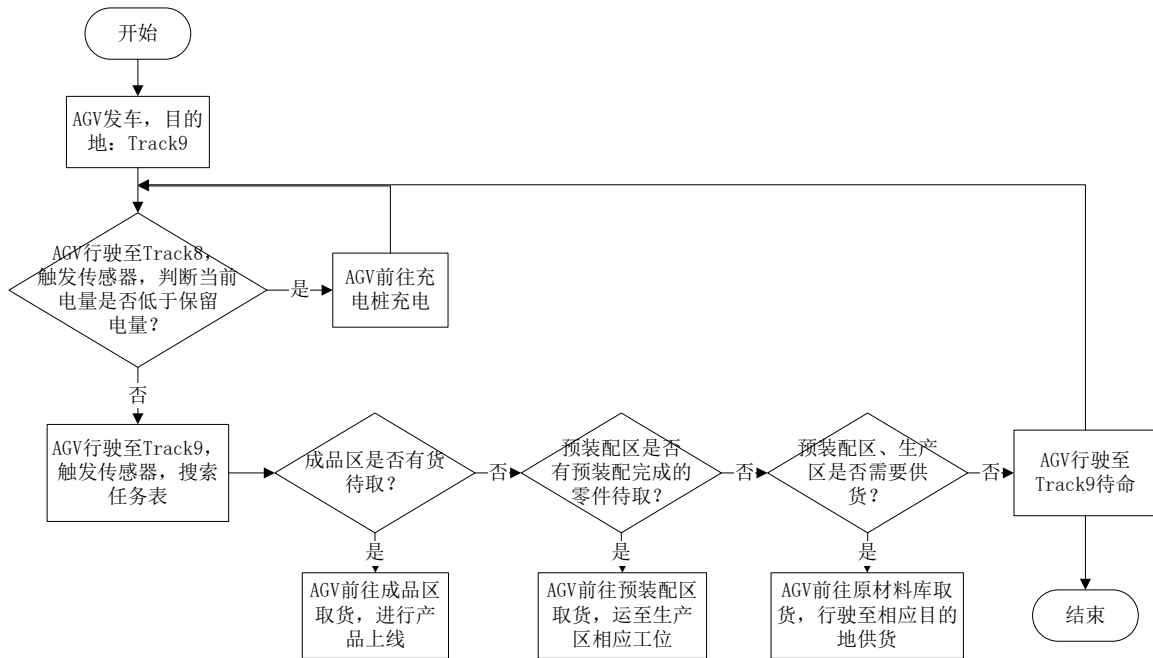


图 7 车间仿真模型逻辑流程图

### 3.3 原材料区模块设计

现实中的原材料库的所有原材料零件均以箱为单位装载，有大料箱和小料箱两种。AGV 取货时将整个箱子一并取走，送至目的地再将所需的零件取出进行加工。对于需要进行预装配才能参与正式生产加工的零件，需将其按比例统一装入一个料箱中，由 AGV 一并送至预装配区。

为更好的区分大小料箱，本报告将原材料库拆成两个 store，分别存储大料箱、小料箱，大小料箱中的零件种类和数目根据 excel 表格设定。另外再设置一个 store，用于存储存放需要进行预装配的零件的料箱，每一个料箱中存储每种预装配件各 280 个。

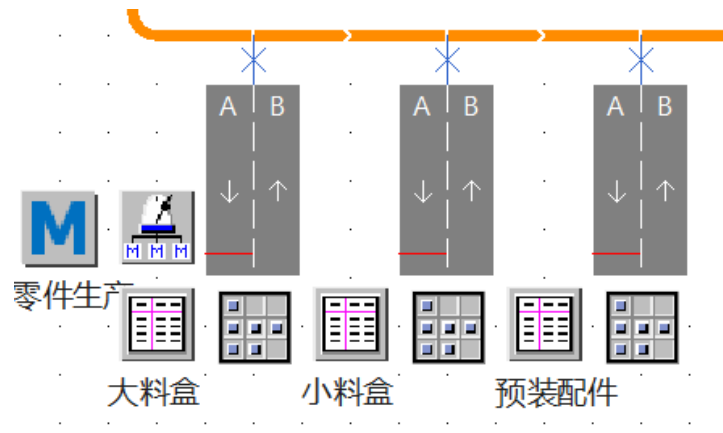


图 8 原材料库模型布局

仿真开始时，执行设定的 creatParts 方法，利用 create 方法将每个 store 装满料箱，然后在每个料箱上根据设置好的 DataTable 装载零件。由于 creatParts 方法只在仿真开始时执行一次，在模型中加入了 Generator，每隔 1 小时调用一次 creatParts 方法，以持续地生成原材料零件，保证原材料库能够及时供应。



图 9 createParts 方法（部分）

### 3.3 预装配区模块设计

同原材料库一样，该车间的预装配区通过 TwoLaneTrack 与 Track 相连，该预装配区共有 3 个完全相同的预装配工位，可通过修改全局变量 n\_preassemble 的值设定启用预装配工位的数目。预装配区的模型布局图如下所示。

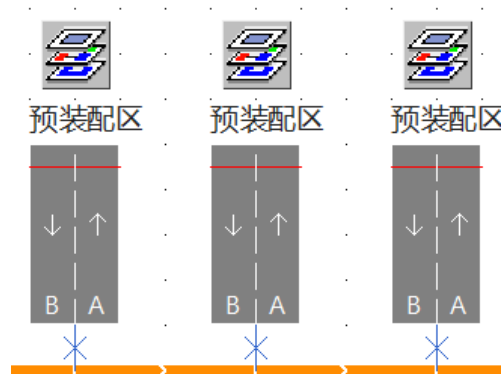


图 10 预装配区模型布局图

预装配工位被设置成一个 frame 框架，3 个预装配工位完全相同，其布局图如上图所示。预装配工位前的 TwoLaneTrack 的 A、B 两个方向均设有传感器控件，当小车行驶至相应位置时，触发传感器的控制方法，完成供货（A 侧传感器）和取货（B 侧传感器）操作。

装有预装配件的料箱从原材料库被运送至预装配区后，先由一个 `dismantleStation` 工位对送来的料箱进行拆箱，将拆下来的原材料零件送至 `AssemblyStation` 工位进行预装配。装配完成后，对预装配件进行装箱，装箱完成后，在“任务表-取货”中写入一条取货指令，让 AGV 前来将预装配件送入生产区对应的装配工位。

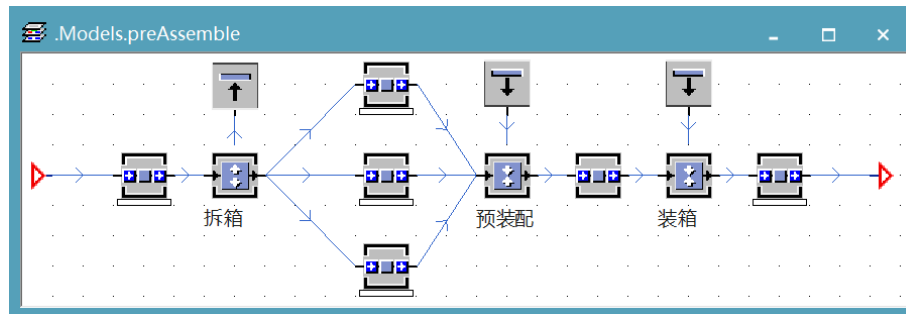


图 11 预装配工位模型布局图

### 3.4 生产区模块设计

该车间共有 2 条完全相同的生产线，每条生产线上有 6 个不同的工位，负责装配和加工不同种类的产品。每个工位配备的大、小料箱数目根据所给的 excel 表格设定，并以全局变量的形式在模型中进行定义，如 `n_bigboxA`、`n_smallboxA` 分别表示 A 工位配备的大、小料箱数目，其余工位的料箱数依据该命名形式类推。

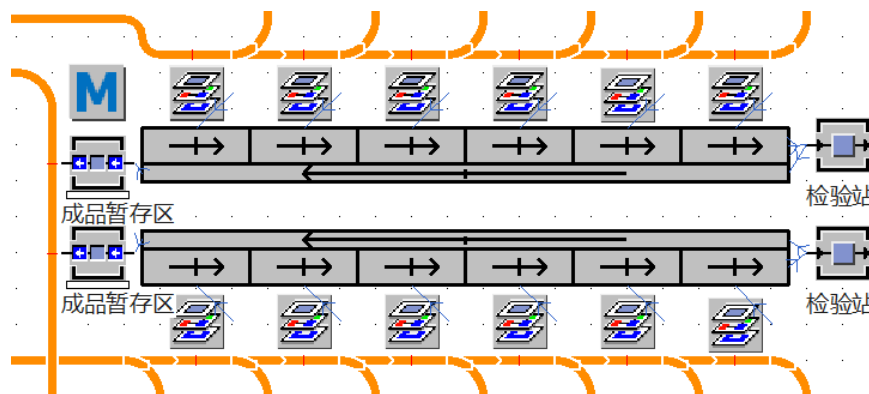


图 12 生产区模型平面布局图

对大部分工位，仅需配备一定数目的大、小料箱即可，而根据 excel 表，D 工位装配加工所需的零件由一部分来自于预装配区，即需要预装配区先对原材料进行一个预装配，才能送至 D 工位备用。因此，D 工位额外设置了一个全局变量 `n_afterboxD`，表示 D 工位需要配置的装有预装配件的料箱数目。

对于 A、E、F 工位，除大、小料箱中的原材料零件外，还需要部分工位常备件参与装配加工，工位常备件默认数量无限，现取现用。故在模型中，对 E、F 工位额外增加了一个 `source`，`source` 生成的 MU 直接送至 `AssemblyStation`，代表工位常备件的使用与消耗。



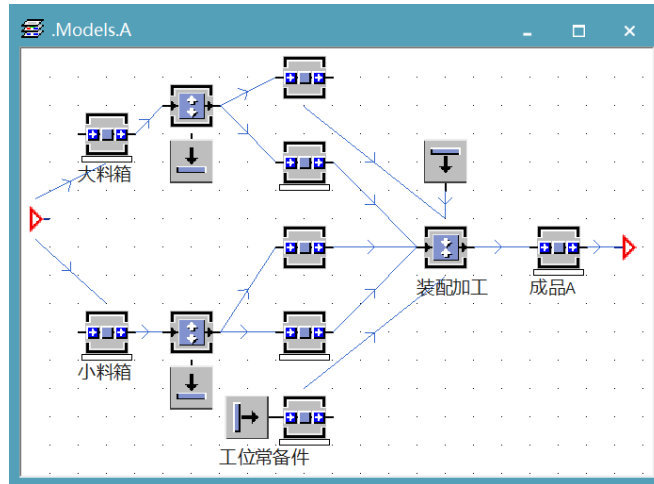


图 13 A 工位框架平面布局图

以 A 工位为例，本模型中各个工位的 frame 所包含的对象元素及其布局大致如上图所示。A 工位中的两个 DismantleStation 模拟从料箱中取出所需零件的过程。如 A 工位装配需要大料箱中的 600 个脚轮和 50 个内饰皮革，DismantleStation 则分别将 600 和 50 个 part 拆至后续的两个暂存区参与后续 AssemblyStation 的装配和加工，而大料箱中的其余零件连同箱子均被 Drain 接收，表示这些零件未在 A 工位使用。

A 工位中 Buffer 和 Buffer1 分别代表大、小料箱的暂存区，当其上边的料箱向后续节点移动时，触发暂存区定义的 OnExit 方法，开始扫描暂存区剩余的料箱数，若料箱少于规定数目，则向“任务表-供货”中写入一条指令，呼叫 AGV 小车前来供货。

### 3.5AGV 充电站模块设计

根据该车间的平面布局图，AGV 充电站设有 6 个并联的充电桩，可同时容纳最多 6 辆 AGV 小车充电，可通过修改全局变量 n\_chargestation 的值设定投入使用的 chargestation 数目。DataTable 记录了充电桩的占用情况，以及给小车充电的开始、结束以及持续时间。

*.Models.Model.chargeStation1						
object 1	boolean 2	time 3	time 4	time 5	integer 6	
string 充电站	是否被占用	充电开始时间	充电结束时间	充电...	充电站使用次数	
1 *.Models.Model.chargeStation1	false	0.0000	0.0000	0.0000	0	
2 *.Models.Model.chargeStation2	false	0.0000	0.0000	0.0000	0	
3 *.Models.Model.chargeStation3	false	0.0000	0.0000	0.0000	0	
4 *.Models.Model.chargeStation4	false	0.0000	0.0000	0.0000	0	
5 *.Models.Model.chargeStation5	false	0.0000	0.0000	0.0000	0	
6 *.Models.Model.chargeStation6	false	0.0000	0.0000	0.0000	0	

图 14 充电站记录表

小车经由 Track8 进入 Track39 时，会触发 Track39 的 OnEntrance 控件，判断小车是否需要充电，若需要充电，则为小车寻找空闲的充电桩，小车行驶至对应的充电桩进行充电。小车进入和离开充电桩时，每一个充电桩均会触发一次 OnEntrance 和 OnExit 控件，分别记录小车进入和离开的时间，以计算小车的充电时间。

## 4. 仿真结果与分析

启用 1 个预装配区、1 条生产线、6 个工位，分别设置 AGV 小车数目为 4、6、8 辆，根据前文设定的参数进行仿真，每次仿真时长设置为 10 小时，统计并分析仿真结果如下：

(1) 当 AGV 数量为 6 辆时，E、F 工位迟迟未开始生产，只有当其余工位结束一轮生产，重新要求供货时，E、F 工位才得到原材料并开始运作。分析 AGV 的调度逻辑，由于原材料的供应和成品的取货共用的是同一套调度系统，即并未设置某辆小车只能提供供货或者取货的专线服务，且取货较之供货有更高的优先权，当取货列表中有取货指令时，小车必须暂停对未完成供货的工位供货，先执行取货指令，导致 E、F 工位缺少原材料而无法启动生产。增加 AGV 小车数量或适当减少 A~D 工位所要求配备的大、小料箱数可一定程度上解决该问题。

(2) 当 AGV 数量为 4、6、8 时，AGV 的工作率均达到 99%，说明 AGV 几乎一直处于满负荷工作的状态。分析原因，每个工位都配备至少 1 个以上的大、小料箱，每个料箱里边有成百上千可供生产加工的零件，当工位启动生产时，成品几乎是不间断地被生产出来，其生产的速度远大于 AGV 小车将成品运输至上线区的速度，造成对工位供货暂停。当一轮生产暂停时，AGV 又得立即对生产区工位进行供货，几乎没有空闲时间。

(3) 部分工位的大、小料箱数目配备不合理。以 A 工位为例，A 工位配备了 2 个大料箱和 1 个小料箱，而在 BOM 表中，A 工位装配所需的原材料零件“内饰皮革喷胶”仅在小料箱中存有 8 个，当这 8 个材料使用完成后，A 工位则需要再次发出小料箱的补货需求并等待补货，小料箱的其余原材料零件则没有被使用。换句话说，A 工位小料箱中的零件只取用了其中的 8 个。解决这一问题需要重新调整料箱中装在的零件的种类和数目，或调整 A 工位配备的小料箱数目。

## 5. 存在问题与不足

本模型在最终运行时取得了较为合理的结果，但仍存在逻辑上的一些问题和不足，具体阐述如下：

(1) 未设置当待充电的 AGV 数目大于充电桩数目时，AGV 排队等待充电的区域。针对此问题，可在进入每个充电桩前与主干道的分支处添加传感器，设置 AGV 的充电等待区，以避免对主干道造成堵塞。

(2) 仿真后期取货指令超出供货指令时,可能出现所有 AGV 均忙着取成品上线而不向装配工位供货,导致某个工位由于缺少原材料而暂停生产。针对此问题可以考虑三个解决方式。一是可以考虑专线运输,一部分 AGV 只负责供货,另一部分只负责取货,分别对应供货和取货两个任务表;二是将成品在上线前进行装箱,以“料箱”作为最小单元,以减少 AGV 来回取货的次数;三是根据每个工位所需的零件,合理配备各个工位的所需料箱数,尽量使各个工位生产一个成品所需的时间大致相等,并在生产作业开始前的一段时间内设置只允许 AGV 进行供货而不进行取货,以保证生产线上各个工位的原材料供应充足。

## 6.附录(核心代码)

### 6.1 OnReset 方法

```
--清除所有 MU
EventController.DeleteMUsOnReset := true
--清除所有表格内容
DataTable.delete
DataTable6.delete
DataTable8.delete
var n:integer
--初始化充电站记录表
for var i := 1 to n_chargestation
    DataTable[1,i]:=str_to_obj("chargeStation"+to_str(i))      --充电站名称
    DataTable[2,i]:=false                                       --是否被占用
    DataTable[3,i]:=EventController.simTime                     --充电开始时间
    DataTable[4,i]:=EventController.simTime                     --充电结束时间
    DataTable[5,i]:=DataTable[4,i]-DataTable[3,i]               --充电时间
    DataTable[6,i]:=0                                           --占用次数
next
--初始化预装备区任务表
for var i := 1 to n_preassemble
    DataTable6[1,i]:="Buffer0"
    DataTable6[2,i]:=false
    DataTable6[3,i]:=EventController.simtime
    DataTable6[4,i]:=str_to_obj("preassemble"+to_str(i))
next
n:=n_preassemble      --当前表中有的条数
--初始化生产区任务表
for var i := 1 to n_produce
    for var j := 1 to n_part
        switch j
        case 1
            for var k := n+1 to n+n_bigboxA
                DataTable6[1,k]:="Buffer"
                DataTable6[2,k]:=false
```

```

        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("A00"+to_str(i))
    next
    n:=n+n_bigboxA
    for var k := n+1 to n+n_smallboxA
        DataTable6[1,k]:="Buffer1"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("A00"+to_str(i))
    next
    n:=n+n_smallboxA
case 2
    for var k := n+1 to n+n_bigboxB
        DataTable6[1,k]:="Buffer"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("B00"+to_str(i))
    next
    n:=n+n_bigboxB
    for var k := n+1 to n+n_smallboxB
        DataTable6[1,k]:="Buffer1"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("B00"+to_str(i))
    next
    n:=n+n_smallboxB
case 3
    for var k := n+1 to n+n_bigboxC
        DataTable6[1,k]:="Buffer"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("C00"+to_str(i))
    next
    n:=n+n_bigboxC
    for var k := n+1 to n+n_smallboxC
        DataTable6[1,k]:="Buffer1"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("C00"+to_str(i))
    next
    n:=n+n_smallboxC
case 4
    for var k := n+1 to n+n_smallboxD
        DataTable6[1,k]:="Buffer1"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime

```

```

        DataTable6[4,k]:=str_to_obj("D00"+to_str(i))
    next
    n:=n+n_smallboxD
case 5
    for var k := n+1 to n+n_bigboxE
        DataTable6[1,k]:="Buffer"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("E00"+to_str(i))
    next
    n:=n+n_bigboxE
case 6
    for var k := n+1 to n+n_bigboxF
        DataTable6[1,k]:="Buffer"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("F00"+to_str(i))
    next
    n:=n+n_bigboxF
    for var k := n+1 to n+n_smallboxF
        DataTable6[1,k]:="Buffer1"
        DataTable6[2,k]:=false
        DataTable6[3,k]:=EventController.simTime
        DataTable6[4,k]:=str_to_obj("F00"+to_str(i))
    next
    n:=n+n_smallboxF
end
next
next

```

## 6.2 Track9 轨道传感器控件

```

var y:integer
var s:string
var charge,rescharge:integer
--判断是否有电，没电去充电
charge:= @.Batcharge      --当前电量
rescharge:= @.BatReserve   --储备电量
if @.empty and charge<=rescharge
    @.destination:=str_to_obj("Track39")
end
--判断成品区和预装配区是否有货物需要取货，若有货，先取货
if Datatable8.ydim >0 and @.empty
    y := DataTable8.ydim
    for var i :=1 to y
        s := obj_to_str(DataTable8[5,i])
    end
end

```

```

        --成品区
        if DataTable8[3,i] = false and (DataTable8[1,i] = "finalProduct1" or
DataTable8[1,i] = "finalProduct2")
            @.destination := str_to_obj("Track561")
            DataTable8[3,i] := true
            DataTable8.cutrow(i)
            //@.stopped:=false
            return
        --预装配区
        elseif DataTable8[3,i] = false and DataTable8[1,i] = "Buffer5"
            s := "TwoLaneTrack"+strcopy(s,26,1)
            @.destination := str_to_obj(s)
            DataTable8[3,i] := true
            DataTable8.cutrow(i)
            return
        end
    next
end
--判断预装配区、生产区是否需要供货
if Datatable6.ydim >0 and @.empty
    y := DataTable6.ydim
    for var i :=1 to y
        --预装配区
        if DataTable6[2,i] = false and DataTable6[1,i] = "Buffer0"
            @.destination := str_to_obj("TwoLaneTrack6")
            DataTable6[2,i] := true
            return
        --大料箱
        elseif DataTable6[2,i] = false and DataTable6[1,i] = "Buffer"
            @.destination := str_to_obj("TwoLaneTrack4")
            DataTable6[2,i] := true
            return
        --小料箱
        elseif DataTable6[2,i] = false and DataTable6[1,i] = "Buffer1"
            @.destination := str_to_obj("TwoLaneTrack5")
            DataTable6[2,i] := true
            return
        end
    next
end
--若暂无任务供指派
if @.empty
    @.destination:=str_to_obj("Track9")
    return
end

```