

垃圾回收物流仿真系统设计

1. 垃圾回收物流系统介绍

城市某一区域，共有 7 个居民小区，每个小区有一个固定垃圾投放处，两个垃圾箱分别投放完全废弃垃圾和可回收利用垃圾；每天有专门公司（垃圾处理公司）派运输车收集垃圾，将垃圾从居民区运送之中转站，再由中转站运至目的地—垃圾处理中心。

仿真程序研究如何设计物流系统，能够使收集系统在满足时间约束、载重约束的条件下，使垃圾处理公司的物流总成本最低。系统配置的项目主要有车辆载重量、随车工作人员数、客户满意度。

2. 数据信息

仿真系统涉及的数据信息如下：

(1) 地理及需求信息

各个收集点所有的人口数、距离垃圾转运站、公司停车场的距离信息如表一所示。

各收集点之间以及停车场和转运站之间的距离信息如表二所示。

表一 收集点信息

收集点名称	收集点居民数 people(i)(千人)	距离停车场距离 distance(ip) (km)	距离垃圾转运站距离 distance(it)(km)
Dump001	1.5	6	16
Dump002	1.45	16	27
Dump003	1.65	26	24
Dump004	1.85	38	36
Dump005	1.4	28	26
Dump006	2.05	16	14
Dump007	1.6	10	8

表二 点间距离

路径起点	路径终点	距离 (km)	符号
Tran_station	Corp park	11	Distance(pt)
Dump001	Dump002	10	Distance(12)
Dump002	Dump003	10	Distance(23)
Dump003	Dump004	12	Distance(34)
Dump004	Dump005	10	Distance(45)
Dump005	Dump006	12	Distance(56)
Dump006	Dump007	6	Distance(67)

(2) 与收集车辆有关的信息

- ① 固定成本[price(i)]：收集车辆的购买费用；
- ② 可变成本[cost(i)]：车辆的行驶费用、维护费用和雇员工资；
- ③ 行驶速度：分为空车行驶速度[unloadspeed(i)]和载重行驶速度[loadspeed(i)]；
- ④ 收集时间[loadtime(i)]：在垃圾收集点收集垃圾所需的时间
- ⑤ 卸载时间[unloadtime(i)]：在转运站卸载垃圾所需的时间

具体信息如表三所示：

表三 收集车辆信息

车辆名称	载重 t	固定成本 万元	可变成本 元/km	载重速度 km/h	空载速度 km/h	人员 人	人员工资 元/天.人	收集时间 min	卸载时间 min
车 1	X_1	$2+ X_1$	X_1	60	72	P_1	60	$3+y*6/ P_1$	$3+ X_1*6/ P_1$

车 2	X_2	$2+1.2 * X_2$	X_2	60	72	P_2	60	$2+y*8/ P_2$	$2+ X_2*6/ P_2$
-----	-------	---------------	-------	----	----	-------	----	--------------	-----------------

说明：

- 车 1 用来收集不可回收垃圾；车 2 用来收集可回收利用垃圾；
- X_1, X_2 为现有两种车的吨位数， X_1 取值：3，4 或 5 吨； X_2 取值：1.5；2；2.5；
- y 为每次收集的垃圾量，单位--吨；
- P_1, P_2 为两种车可配备的随车工作人员数（可以是 1，2，3）。

(3) 与垃圾相关的信息

- ① 人均垃圾量：平均每人每天产生的垃圾数量
 - ② 垃圾比例：各种垃圾成分所占比例
- 假设人均垃圾量服从均值为 1.2kg/天的埃尔郎分布，垃圾成分分为两类：一类为可重复利用资源，一类为无需用销毁垃圾，两者比例为 1：2。

(4) 收集垃圾的成本函数

现在假设仿真一年 365 天的垃圾回收工作，决定两种型号的车配置何种吨位以及随车工作人员数，垃圾回收物流成本 (Total Logistic Cost, TLC) 最低。

$TLC = \text{车 1 成本} + \text{车 2 成本}$

车 I 所耗成本 = I 的固定成本 + I 的可变成本 + I 的服务时间成本 + I 的人员成本

即：

$TLC = CC_1 + CC_2$

$CC_{[i]} = X_{[i]} + V_{[i]} + S_{[i]} + P_{[i]} * 365 * 60$

$S_{[i]} = (\text{runtime}_{[i]} - 240) * m$

$S_{[i]}$ 为垃圾回收的服务时间成本， m 为收集时间对服务时间成本的惩罚因子。垃圾回收公司在十二点之前收集完毕，社会效应好，给予奖励；在十二点后完成，社会满意度低，影响公司信誉，成本增加。

$\text{runtime}_{[i]}$ 为垃圾车将垃圾全部回收完成的最终时间，也即垃圾车每天的运行时间；

3. 系统逻辑结构

此垃圾回收物流系统的逻辑结构分为四个字模块：垃圾产生模块、叫车模块、收集模块、收工模块、数据处理模块。

(1) 垃圾产生模块 每天收集工作开始时，系统根据小区的人数、人均垃圾量、两种垃圾量的比例，通过爱尔朗随机分布函数，随机产生各个小区的两种垃圾数量，分别存入各个小区的两个垃圾箱内。

(2) 叫车模块 在垃圾存入各个小区的两个垃圾箱过程中，根据每种垃圾的总量以及每种垃圾车的载重量，生成叫车的次数。

(3) 收集模块 叫车次数确定之后，车辆根据需求，沿着最短路径依次收集垃圾。在收集过程中，在一个垃圾点，如果垃圾车收集满了或当天的收集工作全部完成，车辆驶向垃圾中转站，进行卸载操作；如果垃圾车未满载，则驶向下一最近垃圾收集点继续收集。

(4) 收工模块 当天的收集工作全部完成之后，车辆驶向公司停车场，当天收集工作完毕。

(5) 数据处理模块 每天车辆收工时，

4. 仿真模型的建立

根据系统逻辑结构和数据信息，我们建立具有 7 个垃圾收集点、一个垃圾转运站的 WITNESS 垃圾回收仿真模型。具体步骤如下：

(1) 元素定义 Define

建立仿真模型时，首先定义仿真模型中所需的元素，并设计它们的可视效果。我们将设计的元素分为两类——实体元素和逻辑元素，名称如下。

表四 实体元素列表：

元素名称	类型	说明
dumpa(1)~ dumpa(7)	Part	七个小区不可回收的垃圾
dumpb(1)~ dumpb(7)	Part	七个小区可回收利用的垃圾
dumpcart(1)	Vehicle	专门回收 dumpa 的垃圾回收车
dumpcart(2)	Vehicle	专门回收 dumpb 的垃圾回收车

buffera(1) ~ buffera(7)	Buffer	各小区不可回收垃圾的垃圾箱
bufferb(1) ~ bufferb(7)	buffer	各小区可回收利用垃圾的垃圾箱
Corppark	Track	公司停车场，
Transfer_station	Buffer	垃圾转运站
road(i, j)	Track	由节点实体 i 向节点实体 j 方向的道路
road(j, i)	track	由节点实体 j 向节点实体 i 方向的道路
Geta(1)~geta(7)	Track	车辆将通过它们来进行垃圾 a 的回收
Getb(1)~getb(7)	Track	车辆将通过它们来进行垃圾 b 的回收

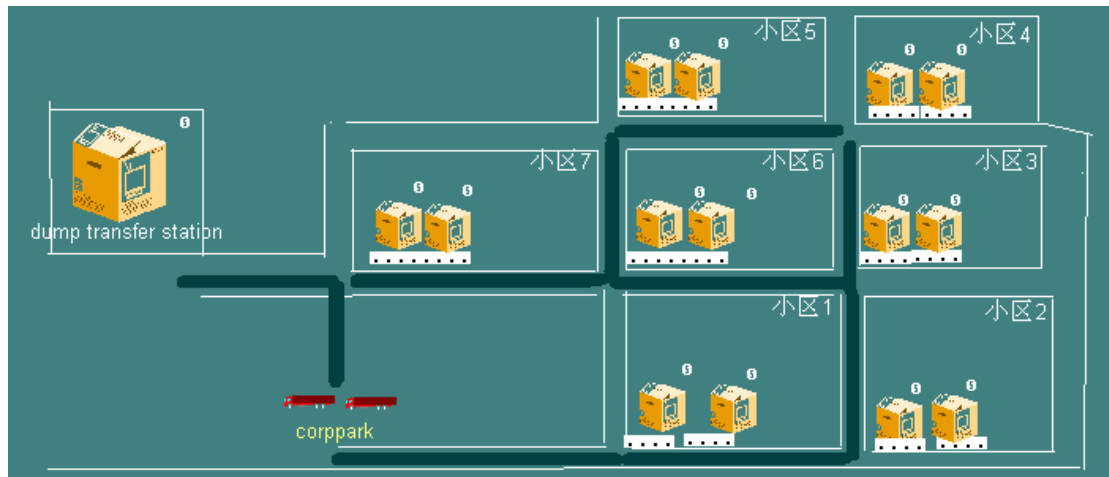
表五 逻辑元素—变量列表：

变量名称	类型	说明
people(7)	integer	各小区的居民数量
capacitycart(2)	integer	两种垃圾车每次可收集的垃圾数量，即载重量
ndemand(2)	integer	每天对两种垃圾车的需求次数
labor(2)	integer	两种垃圾车的随车工作人员数
Salary	Real	工作人员每天工资
price(2)	Real	两种垃圾车的购买价格
costrate(2)	Real	两种垃圾车的运行费率
unloadspeed(2)	Real	两种垃圾车的空载速度
loadspeed(2)	Real	两种垃圾车的实载速度
loadnum (2)	integer	两车辆到达垃圾收集点可收集的垃圾数量
nfreet(2)	integer	两车辆剩余载重能力
Nparts(7)	integer	每一垃圾收集点在特定仿真时点剩余垃圾量
loadtime(2)	Real	两车辆收集垃圾所需的时间
unloadtime(2)	Real	两车辆卸载垃圾所需的时间
runtime(2,3)	Real	存储车辆的时间参数，runtime(k,1)存储仿真钟的当前数值，runtime(k,2)存储车辆 k 在当天的运行时间，runtime(k,3)存储车辆 k 在仿真过程中总的运行时间
Timeweigh	Real	时间惩罚因子
timecost(2)	Real	时间惩罚成本
sumcost	Real	目标函数 objfun 中用于统计系统运行的总费用

定义一个实数型逻辑函数 objfun()，用于计算和统计系统运行一年 365 天所花费的总费用。

(2) (2) 元素显示 display

各个元素的显示设置如下图所示。



(3) (3) 元素详细设计

在该子模块中定义随着仿真钟的推进，装载垃圾的逻辑条件。该系统的工作班次制度采用每天工作8小时，每一仿真时间等价于现实时间一分钟。

为了实现系统的仿真运行，需要对系统中的元素进行详细设计。具体设计如下：

a . a . 系统初始化程序设计 (initialize actions)

personum (1) = 1.5

personum (2) = 1.45

personum (3) = 1.65

personum (4) = 1.85

personum (5) = 1.4

personum (6) = 2.05

personum (7) = 1.6

!给各小区的人数赋初值，单位：千人

FOR num = 1 TO 7

meandump (num) = 1.2 * personum (num) * 1000

NEXT

!分别生成每天七个小区产生垃圾的均值

set capacity of dumpcart1 to capacitycart(1)

set capacity of dumpcart2 to capacitycart(2)

!分别设定两个车辆的载重量，由变量数组 capacitycart(1)和 capacitycart(2)决定。

unloadtime = 0 !卸载时间

FOR num = 1 TO 7

moddemanda (num) = 0

moddemandb (num) = 0 !决定叫车次数的变量

NEXT

FOR num = 1 TO 4

roadchoicea (num) = 0

roadchoiceb (num) = 0 !最短路径控制变量

NEXT

FOR num = 1 TO 3

runtimea (num) = 0

runtimeb (num) = 0 !车辆运行时间控制变量

NEXT

intervaltime = 1440 !每天的分钟数

timeweigh = 5 !时间惩罚因子

laborneeda = 1

laborneedb = 1 !两辆车随车工作人员数

!对变量赋初始值

b . 各个小区垃圾的详细设计

垃圾元素详细设计表

Name	First arrival	Inter arrival	Lot size	To
Dumpa1	0.0001	1440	ERLANG (meandump (1) / 3,3,1)	PUSH to buffera1
Dumpa2	0.0002	1440	ERLANG (meandump (2) / 3,3,2)	PUSH to buffera2
Dumpa3	0.0003	1440	ERLANG (meandump (3) / 3,3,3)	PUSH to buffera3
Dumpa4	0.0004	1440	ERLANG (meandump (4) / 3,3,4)	PUSH to buffera4
Dumpa5	0.0005	1440	ERLANG (meandump (5) / 3,3,5)	PUSH to buffera5
Dumpa6	0.0006	1440	ERLANG (meandump (6) / 3,3,6)	PUSH to buffera6
Dumpa7	0.0007	1440	ERLANG (meandump (7) / 3,3,7)	PUSH to buffera7
Dumpb1	0.0001	1440	ERLANG (meandump (1) / 3,3,8)	PUSH to bufferb1
Dumpb2	0.0002	1440	ERLANG (meandump (2) / 3,3,9)	PUSH to bufferb2
Dumpb3	0.0003	1440	ERLANG (meandump (3) / 3,3,10)	PUSH to bufferb3
Dumpb4	0.0004	1440	ERLANG (meandump (4) / 3,3,11)	PUSH to bufferb4
Dumpb5	0.0005	1440	ERLANG (meandump (5) / 3,3,12)	PUSH to bufferb5
Dumpb6	0.0006	1440	ERLANG (meandump (6) / 3,3,13)	PUSH to bufferb6
Dumpb7	0.0007	1440	ERLANG (meandump (7) / 3,3,14)	PUSH to bufferb7
垃圾名称	*	一天的分钟数	Meandump 为通过各小区人数计算出来的变量数组	决定垃圾的存放

*决定每天各个小区的垃圾依次产生，以便于生产叫车次数。

C. 运输车辆详细设计

Name	Unload speed	Load speed	to
Dumpcart1	1 . 2	1 . 0	PUSH to corppark(1)
Dumpcart2	1 . 2	1 . 0	PUSH to corppark(2)
车辆名称	空载速度 (km/min)	实载速度	生成时，放入公司停车场的车位

D. 当垃圾放入垃圾箱时的活动，也即每个垃圾箱的“actions on input”中的程序

元素 buffera1

```
IF MOD (NPARTS (buffera1),cartcapacity(1)) = 1
  CALL dumpcart1, get1, road7_t, 1
  VSEARCH road0_1, corppark, road1_2, road1_0, road2_3, road3_4, road4_5,
    road5_6, road6_5, road6_7, road7_t, roadt_7, road1_0, road3_6,
    road6_3, road7_0, road0_7, road2_1, geta1, geta2, geta3, geta4,
    geta5, geta6, geta7, getb1, getb2, getb3, getb4,
    getb5, getb6, getb7
```

ENDIF

```
moddemanda (1) = MOD (NPARTS (buffera1),cartcapacity(1))
```

程序解释：

第一行：判断当 buffera1 中的垃圾的数量同车辆 1 的载重量取余为 1 时，发生 if...endif 之间的活动；

第二行：叫车 dumpcart1，该车将在路径 get1 上装载垃圾，在路径 road7_t 上卸载垃圾，优先级为 1；

第三行至第五行：在所有路径上搜索车辆 dumpcart1；

第七行：汇总非整车垃圾的数量，带到 buffera2 中继续计算并叫车。

Buffera2~buffera7 的“actions on input”中的程序如下：

```
IF MOD (moddemanda (i-1) + NPARTS (buffera(i)),cartcapacity1) = 1
  CALL dumpcart1, geta(i), road7_t, 0
  VSEARCH road0_1, corppark, road1_2, road1_0, road2_3, road3_4, road4_5,
    road5_6, road6_5, road6_7, road7_t, roadt_7, road1_0, road3_6,
    road6_3, road7_0, road0_7, road2_1, geta1, geta2, geta3, geta4,
    geta5, geta6, geta7, getb1, getb2, getb3, getb4,
    getb5, getb6, getb7
```

ENDIF

```
moddemanda (i) = MOD (moddemanda (i-1) + NPARTS
(buffera(i)),cartcapacity1)
```

i 表示本垃圾箱的序号，为 2, 3, 4, 5, 6, 7；

第一行：将其上一个垃圾箱的非整车垃圾数量同本垃圾箱中的垃圾数相加，然后与车辆一的载重量取余，当结果为 1 时，叫车。

垃圾箱 bufferb1~bufferb7 的“actions on input”同垃圾箱 buffera1~buffera7 的“actions on input”处理逻辑完全一样，只需要将 buffera 改为 bufferb、dumpcart1 改为 dumpcart2、moddemanda 改为 moddemandb、geta 改为 getb 即可。

E. 两点间运输路径上的程序设计

下面以 road2_3 为例，加以说明其逻辑流程。其它路径类似，可以在路径的 general detail 中的 output to 中加以查看。

```
IF VEHICLE (road2_3,1) = dumpcart1
  IF NPARTS (buffera3) > 0
    PUSH to geta3(1)
  ELSE
    PUSH to road3_4
  ENDIF
ELSEIF VEHICLE (road2_3,1) = dumpcart2
  IF NPARTS (bufferb3) > 0
    PUSH to getb3
  ELSE
    PUSH to road3_4
  ENDIF
ELSE
  Wait
ENDIF
```

第一行到第六行：决定车辆 dumpcart1 到达路径 road2_3 末端时，它的运行路径。如果此时垃圾箱 buffera3 中有垃圾，则车辆 dumpcart1 驶向路径 geta3，进行垃圾收

集（第二行到第四行）；如果垃圾箱 buffera3 中没有垃圾，则车辆 dumpcart1 驶向路径 road3_4。

第七行到第十二行：决定车辆 dumpcart2 到达路径 road2_3 末端时，它的运行路径。如果此时垃圾箱 bufferb3 中有垃圾，则车辆 dumpcart2 驶向路径 getb3，进行垃圾收集（第八行到第十行）；如果垃圾箱 bufferb3 中没有垃圾，则车辆 dumpcart2 驶向路径 road3_4。

如果是其他情况，等待（第十三行到第十四行）。

F. 路径 get 系列的设置

在 Get 系列 (Geta1~geta7, Getb1~getb7) 路径中设置车辆的装载 (loading) 程序和条件，它们的设定以及程序的处理流程基本相同，下面举 geta2 加以说明。其中装载数量 (loadnum) 和装载时间(loadtime)在 general detail 页的 actions on front 中设定；装载程序和条件在 loading detail 页中进行设定。

General detail 页的 actions on front 程序如下：

```
IF NFREE (dumpcart1) >= NPARTS (buffera2)
```

```
  loadnum = NPARTS (buffera2)
```

```
ELSE
```

```
  loadnum = NFREE (dumpcart1)
```

```
ENDIF
```

```
loadtimea (1) = 3 + loadnum / 1000 * loadindexa / laborneeda
```

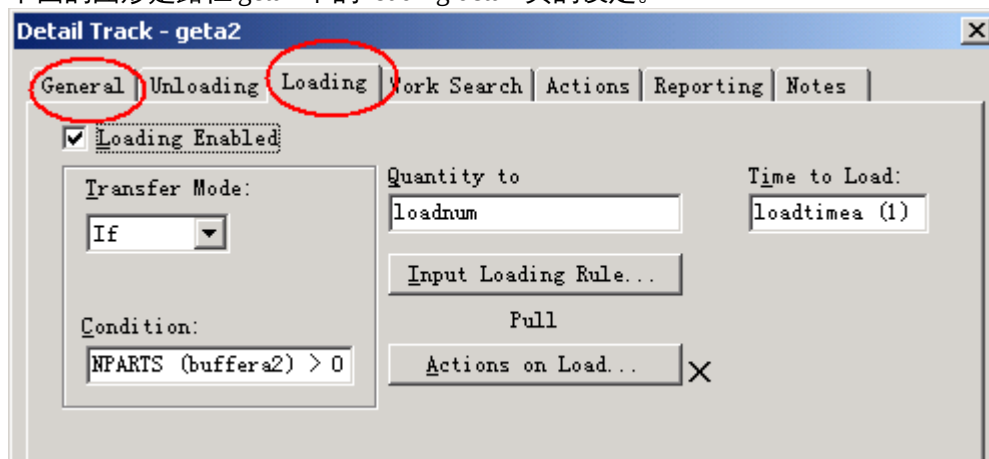
```
dayloadtimea = dayloadtimea + loadtimea (1)
```

程序解释：

第一行至第五行：如果车辆 dumpcart1 的空余容量 NFREE(dumpcart1)不小于垃圾箱 buffera2 中的垃圾数量，则收集数量 loadnum 为垃圾箱 buffera2 中的所有垃圾；否则，收集数量 loadnum 为车辆的空余容量。

第六行：计算本次收集所需的时间 loadtimea(1)，它是本次垃圾收集量、随车工作人员数以及收集时间系数的函数。

下面的图形是路径 geta2 中的 loading detail 页的设定。



在选中 loading enabled（能够装载）前的复选框后，将会出现该界面上的其他内容。Transfer Mode（装载模式）有三种：if、call、always。我们选择条件模式 if，在条件 condition: 框中输入条件表达式 NPARTS(buffera2)>0，即当垃圾箱 buffera2 中的垃圾数量大于零时，能够装载；装载数量等于 Quantity to 框中的变量 loadnum 的值；装载的时间需要 Time to Load 框中的变量 loadtimea(1)；装载的源在 Input Loading Rule 规则中进行设定，为“PULL from buffera2”，从垃圾箱 buffera2 中收集。

G. road7_t 的设定

车辆每次到达路径 road7_t 的末端时，都要进行卸载处理，所以对路径 road7_t 的详细设计项目包括卸载所需的时间，卸载模式等。

General detail 页的 actions on front 中设定卸载所需的时间，程序如下：

```
IF VEHICLE (road7_t,1) = dumpcart1
```

```
  unloadtime = 3 + 6 * NPARTS (dumpcart1) / 1000 / laborneeda
```

```

dayunloadtimea = dayunloadtimea + unloadtime
ELSE
  unloadtime = 2 + 6 * NPARTS (dumpcart2) / 1000 / laborneedb
  dayunloadtimeb = dayunloadtimeb + unloadtime
ENDIF

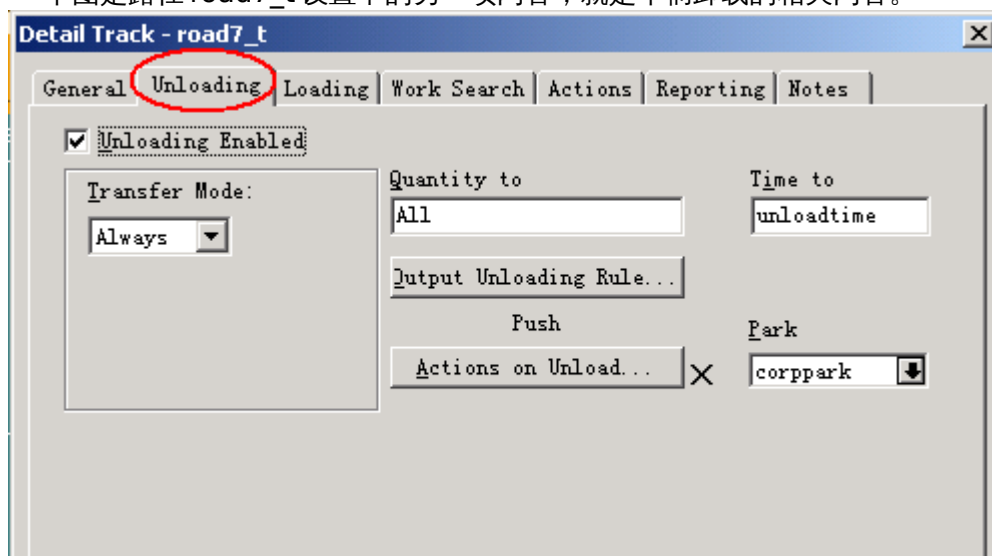
```

程序解释：

第一行至第三行，根据函数 VEHICLE()的结果，如果到达车辆是 dumpcart1，就通过车辆 dumpcart1 中所装载的垃圾数量 NPARTS (dumpcart1)、车辆 dumpcart1 的随车工作人员数 laborneeda 来确定卸载时间 unloadtime（第二行），然后统计车辆 dumpcart1 的总的卸载时间 dayunloadtimeb（第三行）。

第四行至第七行，否则，也就是到达的车辆是 dumpcart2，就通过车辆 dumpcart2 中所装载的垃圾数量 NPARTS (dumpcart2)、车辆 dumpcart2 的随车工作人员数 laborneedb 来确定卸载时间 unloadtime(第五行)，然后统计车辆 dumpcart2 的总卸载时间 dayunloadtimeb（第六行）。

下图是路径 road7_t 设置中的另一项内容，就是车辆卸载的相关内容。



在选中 Unloading Enabled（能够卸载）前的复选框后，该页面上的其他选项将能够被用户设置。卸载模式（Transfer Mode）也有三种模式 if、call、always，我们选择模式 always，只要车辆到达路径 road7_t 的末端，就进行卸载活动。卸载数量 Quantity to 设定为 All，就是卸载车辆上的所有垃圾。卸载时间 Time to 设定为变量 unloadtime，决定本次卸载所需要的时间。停车 Park 设定为公司停车场 corppark，决定在当天回收任务完成后，车辆泊放的位置。卸载到什么地方，将由“output unloading rule”中的程序决定，程序为：push to ship，将所有垃圾送出系统。

(4) (4) 数据处理子模块

每天车辆完成收集工作完成后，驶向垃圾处理公司停车场停泊时，都将进行一些数据统计和处理活动。这些程序写在路径 road7_0 的“Actions on Front”中，程序及其说明如下：

```

IF VEHICLE (road7_0,1) = dumpcart1
  IF NDemands (dumpcart1) = 0
    runtimea (1) = TIME
    runtimea (2) = runtimea (2) + 1
    runtimea (3) = runtimea (1) - 1000 * (runtimea (2) - 1)
    timecosta = timecosta + timeweigh * (runtimea (3) - 240)
    daydistance = DISTANCE (dumpcart1) / 10 - lastdaydistance
    lastdaydistance = DISTANCE (dumpcart1) / 10
    dayunloadtimea = 0
    dayloadtimeb = 0
  
```

```

ENDIF
ELSEIF VEHICLE (road7_0,1) = dumpcart2
  IF NDemands (dumpcart2) = 0
    runtimeb (1) = TIME
    runtimeb (2) = runtimeb (2) + 1
    runtimeb (3) = runtimeb (1) - 1000 * (runtimeb (2) - 1)
    timecostb = timecostb + timeweigh * (runtimeb (3) - 240)
    daydistancea = DISTANCE (dumpcart2) / 10 - lastdaydistancea
    lastdaydistancea = DISTANCE (dumpcart2) / 10
    dayunloadtimeb = 0
    dayloadtimeb = 0
  ENDIF
ENDIF
ENDIF
FOR num = 1 TO 4
  IF VEHICLE (road7_0,1) = dumpcart1
    roadchoicea (num) = 0
  ELSEIF VEHICLE (road7_0,1) = dumpcart2
    roadchoiceb (num) = 0
  ENDIF
NEXT

```

程序解释：

第一行至第十一行，如果即将停泊的车辆是 dumpcart1，同时系统此时对车辆 dumpcart1 的需求是零（第三行）；先统计车辆 dumpcart1 在当天收集工作的完成时间，并判断是否超过了 12 点，然后计算当天的时间惩罚成本 timecost（第三至第六行）；统计车辆的运行距离（第七至第八行）；对两个变量置零（第九至第十行）。

第十二行至第二十二行，统计如果即将停泊的车辆是 dumpcart2 的一些数据，解释同 dumpcart1 一样。

第二十四行至第三十行，对路径选择变量数组置零。

目标函数 objfun() 中的程序如下：

```

sumcost = 0 !统计总费用的变量置零
sumcost = initialcosta + initialcostb + sumcost
!将车辆的购置成本加到总费用变量中
sumcost = (laborneedb + laborneeda) * 60 * (runtimea (2) + 1) + sumcost
!将工作人员工资添加到总费用变量
sumcost = DISTANCE (dumpcart001) / 10 * permilecosta + DISTANCE
(dumpcart002) / 10 * permilecostb + sumcost
!将车辆的运行费用添加到总费用变量
sumcost = timecosta + timecostb + sumcost
!将时间惩罚成本添加到总费用变量
RETURN sumcost !返回变量 sumcost 给目标函数

```

5. 仿真运行与结果分析

我们现在采用如下两种配置，来运行该仿真模型，仿真时间为一年（即 365*1440 分钟）。

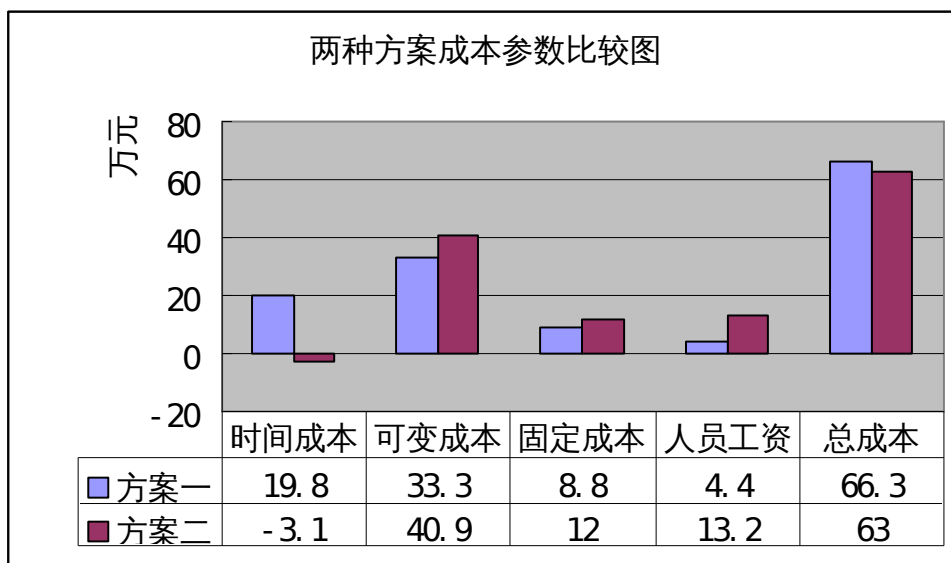
项目 方案	车辆名称	载重量 (吨)	随车人 员 (人)	时间惩罚因子
方案一	Dumpcart(1)	3	1	2
	Dumpcart(2)	1.5	1	2
方案二	Dumpcart(1)	5	3	2
	Dumpcart(2)	2.5	3	2

其它的项目在两个方案中的值完全相同，参见前面的表格。

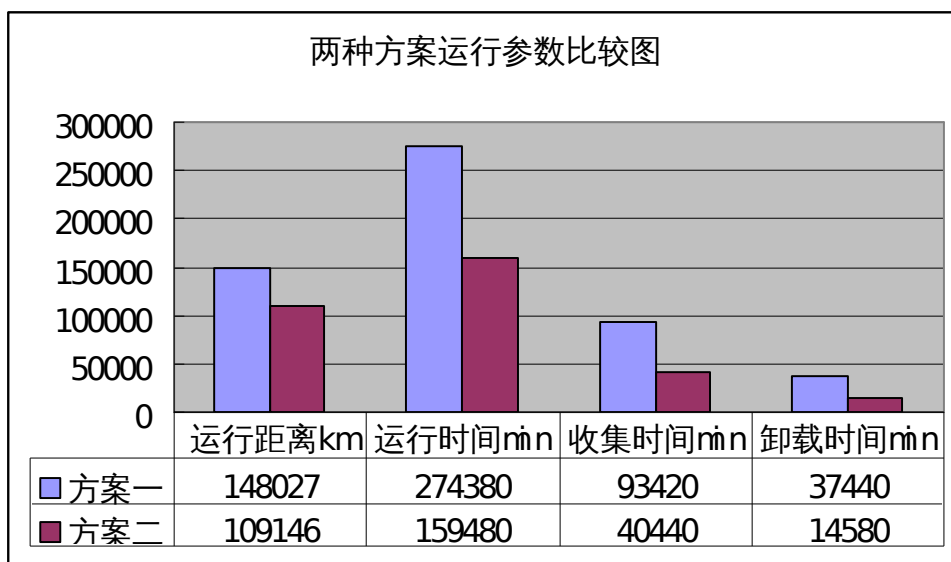
通过仿真运行后得到如下数据：

方案		运行距离 km	运行时间 h	时间成本 万元	收集时间 h	卸载时间 h	可变成本 万元	固定成本 万元	人员工资 万元	总成本 万元	总成本万元
—	Dumpcart(1)	73691	2310	10.2	744	407	22.1	5	2.2	39.5	66.3
—	Dumpcart(2)	74336	2263	9.6	813	217	11.2	3.8	2.2	26.8	
—	Dumpcart(1)	54657	1366	-1.1	342	157	27.3	7	6.6	39.8	63
—	Dumpcart(2)	54489	1292	-2	332	86	13.6	5	6.6	23.2	

为了更清晰的比较两种方案的成本参数，将上表的数据绘制成直方图如下：



从成本数据比较图上可以看出，当采用方案二时，车辆吨位数和随车人员的增加，使得固定成本、人员工资都比方案一要大；同时，回收速度要比方案一快的多，使得可变成本和居民满意度较高；时间惩罚成本低于零，即取得了很好的市场评价和公司品牌效应。在这两种方案下，每年方案二的物流成本比方案一要少 3.3 万元。



从运行参数比较图上可以看出，两种方案的物流成本差异收集时间、卸载时间、运行时间、运行距离的差异引起的。由于方案一的车辆吨位和随车人员都比方案二要少，所以同是收集等量垃圾，它的收集时间和卸载时间都比方案二长，它的收集次数要比方案二多即运行距离比方案二长。

附：以上程序当中所涉及的函数介绍如下

MODEL INFORMATION:

Model Name: dumpreverse
Model Title: the simulation for reverse logistics
Model Author: jiannywang, NUAA
Creation Date: Sun Mar 10 09:26:20 2004
Initial Actions: XLReadArray ("\\reverse.xls","input","\$B\$3:\$B\$9",personum)
FOR num = 1 TO 7
meandump (num) = 1.2 * personum (num) * 1000
NEXT

ELEMENT NAME: dump1

Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump001(1)
Actions
Create: IF IUNIFORM (1,3,101) = 1
dumpreusable = "yes"
ELSE
dumpreusable = "no"
ENDIF
moddemand (1) = MOD (moddemand (1),cartcapacity) + 1
IF moddemand (1) = cartcapacity
moddemand (1) = 0
ENDIF

ELEMENT NAME: dump1a

Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump001a(1)
Actions
Create: IF IUNIFORM (1,3,101) = 1
dumpreusable = "yes"
ELSE
dumpreusable = "no"
ENDIF
moddemand (1) = MOD (moddemand (1),cartcapacity) + 1
IF moddemand (1) = cartcapacity
moddemand (1) = 0
ENDIF

ELEMENT NAME: dump2

Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump002(1)
Actions
Create: moddemand (2) = MOD (moddemand (2),cartcapacity) + 1
IF moddemand (2) = cartcapacity
moddemand (2) = 0
ENDIF

ELEMENT NAME: dump2a
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump002a(1)
Actions
Create: moddemand (2) = MOD (moddemand (2),cartcapacity) + 1
IF moddemand (2) = cartcapacity
moddemand (2) = 0
ENDIF

ELEMENT NAME: dump3
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0001
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump003(1)
Actions
Create: moddemand (3) = MOD (moddemand (3),cartcapacity) + 1
IF moddemand (3) = cartcapacity
moddemand (3) = 0
ENDIF

ELEMENT NAME: dump3a
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0001
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump003a(1)
Actions
Create: moddemand (3) = MOD (moddemand (3),cartcapacity) + 1
IF moddemand (3) = cartcapacity
moddemand (3) = 0
ENDIF

ELEMENT NAME: dump4
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0002
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump004(1)
Actions
Create: moddemand (4) = MOD (moddemand (4),cartcapacity) + 1
IF moddemand (4) = cartcapacity
moddemand (4) = 0
ENDIF

ELEMENT NAME: dump4a
Element Type: Part

Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0002
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump004a(1)
Actions
Create: moddemand (4) = MOD (moddemand (4),cartcapacity) + 1
IF moddemand (4) = cartcapacity
moddemand (4) = 0
ENDIF

ELEMENT NAME: dump5
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0003
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump005(1)
Actions
Create: moddemand (5) = MOD (moddemand (5),cartcapacity) + 1
IF moddemand (5) = cartcapacity
moddemand (5) = 0
ENDIF

ELEMENT NAME: dump5a
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0003
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump005a(1)
Actions
Create: moddemand (5) = MOD (moddemand (5),cartcapacity) + 1
IF moddemand (5) = cartcapacity
moddemand (5) = 0
ENDIF

ELEMENT NAME: dump6
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0004
Maximum Arrivals: Unlimited
Input / Output Rules
Output: PUSH to dump006(1)
Actions
Create: moddemand (6) = MOD (moddemand (6),cartcapacity) + 1
IF moddemand (6) = cartcapacity
moddemand (6) = 0
ENDIF

ELEMENT NAME: dump6a
Element Type: Part
Type: Variable attributes
Group number: 1
Inter Arrival Time: 480.0
First Arrival at: 0.0004

Maximum Arrivals: Unlimited
 Input / Output Rules
 Output: PUSH to dump006a(1)
 Actions
 Create: moddemand (6) = MOD (moddemand (6),cartcapacity) + 1
 IF moddemand (6) = cartcapacity
 moddemand (6) = 0
 ENDIF

ELEMENT NAME: dump7
 Element Type: Part
 Type: Variable attributes
 Group number: 1
 Inter Arrival Time: 480.0
 First Arrival at: 0.0005
 Maximum Arrivals: Unlimited
 Input / Output Rules
 Output: PUSH to dump007(1)
 Actions
 Create: moddemand (7) = MOD (moddemand (7),cartcapacity) + 1
 IF moddemand (7) = cartcapacity
 moddemand (7) = 0
 ENDIF

ELEMENT NAME: dump7a
 Element Type: Part
 Type: Variable attributes
 Group number: 1
 Inter Arrival Time: 480.0
 First Arrival at: 0.0005
 Maximum Arrivals: Unlimited
 Input / Output Rules
 Output: PUSH to dump007a(1)
 Actions
 Create: moddemand (7) = MOD (moddemand (7),cartcapacity) + 1
 IF moddemand (7) = cartcapacity
 moddemand (7) = 0
 ENDIF

ELEMENT NAME: dump001
 Element Type: Buffer
 Quantity: 1
 Capacity: 65000
 Input Option: Rear
 Output Option: First
 Search From: Front
 Actions
 Input: IF MOD (NPARTS (dump001),cartcapacity) = 1
 CALL dumpcart001,get1,road7_t,1
 VSEARCH
 road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
 _7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
 ENDIF
 moddemand (1) = MOD (NPARTS (dump001),cartcapacity)

ELEMENT NAME: dump001a
 Element Type: Buffer
 Quantity: 1
 Capacity: 65000
 Input Option: Rear
 Output Option: First
 Search From: Front
 Actions
 Input: IF MOD (NPARTS (dump001a),cartcapacity) = 1

```
CALL dumpcart001,get1,road7_t,1
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (1) = MOD (NPARTS (dump001a),cartcapacity)
```

```
ELEMENT NAME: dump002
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (1) + NPARTS (dump002),cartcapacity) = 1
CALL dumpcart001,get2,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (2) = MOD (moddemand (1) + NPARTS (dump002),cartcapacity)
```

```
ELEMENT NAME: dump002a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (1) + NPARTS (dump002a),cartcapacity) = 1
CALL dumpcart001,get2,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (2) = MOD (moddemand (1) + NPARTS (dump002a),cartcapacity)
```

```
ELEMENT NAME: dump003
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (2) + NPARTS (dump003),cartcapacity) = 1
CALL dumpcart001,get3,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (3) = MOD (moddemand (2) + NPARTS (dump003),cartcapacity)
```

```
ELEMENT NAME: dump003a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (2) + NPARTS (dump003a),cartcapacity) = 1
```

```
CALL dumpcart001,get3,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (3) = MOD (moddemand (2) + NPARTS (dump003a),cartcapacity)
```

```
ELEMENT NAME: dump004
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (3) + NPARTS (dump004),cartcapacity) = 1
CALL dumpcart001,get4,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (4) = MOD (moddemand (3) + NPARTS (dump004),cartcapacity)
```

```
ELEMENT NAME: dump004a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (3) + NPARTS (dump004a),cartcapacity) = 1
CALL dumpcart001,get4,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (4) = MOD (moddemand (3) + NPARTS (dump004a),cartcapacity)
```

```
ELEMENT NAME: dump005
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (4) + NPARTS (dump005),cartcapacity) = 1
CALL dumpcart001,get5,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (5) = MOD (moddemand (4) + NPARTS (dump005),cartcapacity)
```

```
ELEMENT NAME: dump005a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (4) + NPARTS (dump005a),cartcapacity) = 1
```

```
CALL dumpcart001,get5,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (5) = MOD (moddemand (4) + NPARTS (dump005a),cartcapacity)
```

```
ELEMENT NAME: dump006
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (5) + NPARTS (dump006),cartcapacity) = 1
CALL dumpcart001,get6,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (6) = MOD (moddemand (5) + NPARTS (dump006),cartcapacity)
```

```
ELEMENT NAME: dump006a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (5) + NPARTS (dump006a),cartcapacity) = 1
CALL dumpcart001,get6,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (6) = MOD (moddemand (5) + NPARTS (dump006a),cartcapacity)
```

```
ELEMENT NAME: dump007
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (6) + NPARTS (dump007),cartcapacity) = 1
CALL dumpcart001,get7,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (7) = MOD (moddemand (6) + NPARTS (dump007),cartcapacity)
```

```
ELEMENT NAME: dump007a
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front
Actions
Input: IF MOD (moddemand (6) + NPARTS (dump007a),cartcapacity) = 1
```

```

CALL dumpcart001,get7,road7_t,0
VSEARCH
road0_1,corppark,road1_2,road1_0,road2_3,road3_4,road4_5,road5_6,road6_5,road6_7,road7_t,roadt
_7,road1_0,road3_6,road6_3,road7_0,road0_7,road2_1,get1,get2,get3,get4,get5,get6,get7
ENDIF
moddemand (7) = MOD (moddemand (6) + NPARTS (dump007a),cartcapacity)

```

ELEMENT NAME: dump008
Element Type: Buffer
Quantity: 1
Capacity: 65000
Input Option: Rear
Output Option: First
Search From: Front

ELEMENT NAME: processmach
Element Type: Machine
Quantity: 1
Priority: Lowest
Type: Assembly
Assembly Qty: 5000
Cycle Time: 1.0
Input / Output Rules
Input: IF MOD (TIME,480) > 400 AND MOD (TIME,480) < 460
PULL from dump008
ELSE
Wait
ENDIF
Output: PUSH to SHIP

ELEMENT NAME: corppark
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search: None
Input / Output Rules
Output: PUSH to road0_1(1)

ELEMENT NAME: get1
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search: None
Loading: If NPARTS (dump001) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: IF NFREE (dumpcart001) = 0
PUSH to road1_0
ELSE
PUSH to road1_2
ENDIF
Loading: PULL from dump001
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump001)
loadnum = NPARTS (dump001)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get1a

Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump001) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: IF NFREE (dumpcart001) = 0
PUSH to road1_0
ELSE
PUSH to road1_2
ENDIF
Loading: PULL from dump001
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump001)
loadnum = NPARTS (dump001)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get2
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump002) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: IF NFREE (dumpcart001) = 0
PUSH to road2_1
ELSE
PUSH to road2_3
ENDIF
Loading: PULL from dump002
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump002)
loadnum = NPARTS (dump002)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get2a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump002) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: IF NFREE (dumpcart001) = 0
PUSH to road2_1
ELSE
PUSH to road2_3
ENDIF
Loading: PULL from dump002
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump002)
loadnum = NPARTS (dump002)
ELSE
loadnum = NFREE (dumpcart001)

ENDIF

ELEMENT NAME: get3
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search: None
Loading: If NPARTS (dump003) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: IF NFREE (dumpcart001) = 0
 PUSH to road3_6
 ELSE
 PUSH to road3_4
 ENDIF
 Loading: PULL from dump003
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump003)
 loadnum = NPARTS (dump003)
 ELSE
 loadnum = NFREE (dumpcart001)
 ENDIF

ELEMENT NAME: get3a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search: None
Loading: If NPARTS (dump003) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: IF NFREE (dumpcart001) = 0
 PUSH to road3_6
 ELSE
 PUSH to road3_4
 ENDIF
 Loading: PULL from dump003
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump003)
 loadnum = NPARTS (dump003)
 ELSE
 loadnum = NFREE (dumpcart001)
 ENDIF

ELEMENT NAME: get4
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search: None
Loading: If NPARTS (dump004) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: PUSH to road4_5(1)
 Loading: PULL from dump004
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump004)
 loadnum = NPARTS (dump004)
 ELSE
 loadnum = NFREE (dumpcart001)

ENDIF

ELEMENT NAME: get4a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump004) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: PUSH to road4_5(1)
Loading: PULL from dump004
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump004)
loadnum = NPARTS (dump004)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get5
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump005) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: PUSH to road5_6(1)
Loading: PULL from dump005
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump005)
loadnum = NPARTS (dump005)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get5a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump006) > 0
Load Quantity: loadnum
Load Time: 2.0
Input / Output Rules
Output: PUSH to road6_7
Loading: PULL from dump006
Actions
Front: IF NFREE (dumpcart001) >= NPARTS (dump006)
loadnum = NPARTS (dump006)
ELSE
loadnum = NFREE (dumpcart001)
ENDIF

ELEMENT NAME: get6
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0

Work Search None
Loading: If NPARTS (dump006) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: PUSH to road6_7
 Loading: PULL from dump006
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump006)
 loadnum = NPARTS (dump006)
 ELSE
 loadnum = NFREE (dumpcart001)
 ENDIF

ELEMENT NAME: get6a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump005) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: PUSH to road5_6(1)
 Loading: PULL from dump005
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump005)
 loadnum = NPARTS (dump005)
 ELSE
 loadnum = NFREE (dumpcart001)
 ENDIF

ELEMENT NAME: get7
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump007) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: PUSH to road7_t
 Loading: PULL from dump007
Actions
 Front: IF NFREE (dumpcart001) >= NPARTS (dump007)
 loadnum = NPARTS (dump007)
 ELSE
 loadnum = NFREE (dumpcart001)
 ENDIF

ELEMENT NAME: get7a
Element Type: Track
Quantity: 1
Capacity: 1
Zone: 0
Work Search None
Loading: If NPARTS (dump007) > 0
 Load Quantity: loadnum
 Load Time: 2.0
Input / Output Rules
 Output: PUSH to road7_t
 Loading: PULL from dump007
Actions

```
Front:      IF NFREE (dumpcart001) >= NPARTS (dump007)
            loadnum = NPARTS (dump007)
            ELSE
            loadnum = NFREE (dumpcart001)
            ENDIF
```

```
ELEMENT NAME: road0_1
Element Type:   Track
Quantity:       1
Capacity:       2
Zone:           0
Work Search     None
Input / Output Rules
Output:         IF NPARTS (dump001) = 0
                PUSH to road1_2
            ELSE
                PUSH to get1
            ENDIF
```

```
ELEMENT NAME: road0_7
Element Type:   Track
Quantity:       1
Capacity:       1
Zone:           0
Work Search     None
Input / Output Rules
Output:         PUSH to road7_t
```

```
ELEMENT NAME: road1_0
Element Type:   Track
Quantity:       1
Capacity:       2
Zone:           0
Work Search     None
Input / Output Rules
Output:         PUSH to road0_7
```

```
ELEMENT NAME: road1_2
Element Type:   Track
Quantity:       1
Capacity:       2
Zone:           0
Work Search     None
Input / Output Rules
Output:         IF NPARTS (dump002) = 0
                PUSH to road2_3
            ELSE
                PUSH to get2
            ENDIF
```

```
ELEMENT NAME: road2_1
Element Type:   Track
Quantity:       1
Capacity:       1
Zone:           0
Work Search     None
Input / Output Rules
Output:         PUSH to road1_0
```

```
ELEMENT NAME: road2_3
Element Type:   Track
Quantity:       1
```

Capacity: 2
Zone: 0
Work Search None
Input / Output Rules
Output: IF NPARTS (dump003) > 0
 PUSH to get3(1)
 ELSE
 PUSH to road3_4
 ENDIF

ELEMENT NAME: road3_4
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search None
Input / Output Rules
Output: PUSH to get4

ELEMENT NAME: road3_6
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search None
Input / Output Rules
Output: PUSH to road6_7(1)
Actions
On: roadchoice (1) = 1
 roadchoice (2) = 0
 roadchoice (3) = 0

ELEMENT NAME: road4_5
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search None
Input / Output Rules
Output: IF NPARTS (dump005) > 0 AND NFREE (dumpcart) > 0
 PUSH to get5(1)
 ELSE
 PUSH to road5_6
 ENDIF
Actions
On: IF NPARTS (dump004) = 0
 roadchoice (2) = 1
 roadchoice (1) = 0
 ENDIF

ELEMENT NAME: road5_6
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search None
Input / Output Rules
Output: IF NFREE (dumpcart001) > 0 AND NPARTS (dump006) > 0
 PUSH to get6(1)
 ELSE
 PUSH to road6_7
 ENDIF
Actions
On: IF NPARTS (dump005) = 0

```

        roadchoice (2) = 0
        roadchoice (3) = 1
    ENDIF

```

```

ELEMENT NAME: road6_3
Element Type:   Track
Quantity:       1
Capacity:       1
Zone:           0
Work Search     None
Input / Output Rules
  Output:       IF NPARTS (dump003) = 0
                PUSH to road3_4
              ELSE
                PUSH to get3(1)
              ENDIF

```

```

ELEMENT NAME: road6_5
Element Type:   Track
Quantity:       1
Capacity:       1
Zone:           0
Work Search     None
Input / Output Rules
  Output:       PUSH to get5(1)

```

```

ELEMENT NAME: road6_7
Element Type:   Track
Quantity:       1
Capacity:       2
Zone:           0
Work Search     None
Loading:        If NPARTS (dump007) > 0
  Load Quantity: loadnum
  Load Time:    12.0
Input / Output Rules
  Output:       IF NPARTS (dump006) = 0 AND NFREE (dumpcart001) > 0
                PUSH to get7
              ELSE
                PUSH to road7_t(1)
              ENDIF
  Loading:     PULL from dump007
Actions
  On:          IF NPARTS (dump006) = 0
                roadchoice (3) = 0
                roadchoice (4) = 1
              ENDIF

```

```

ELEMENT NAME: road7_0
Element Type:   Track
Quantity:       1
Capacity:       1
Zone:           0
Work Search     None
Input / Output Rules
  Output:       IF NDemands (dumpcart001) = 0
                PUSH to corppark
              ELSE
                PUSH to road0_1
              ENDIF

```

```

ELEMENT NAME: road7_6
Element Type:   Track

```

Quantity: 1
Capacity: 1
Zone: 0
Work Search: None
Input / Output Rules
Output: IF roadchoice (1) = 1
 PUSH to road6_3
 ELSEIF roadchoice (2) = 1
 PUSH to road6_5
 ELSE
 PUSH to get6
 ENDIF

ELEMENT NAME: road7_t
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search: None
Unloading: Always
Unload Time: 0.0
Park Position: corppark
Input / Output Rules
Output: PUSH to roadt_7(1)
Unloading: PUSH to dump008
Actions
On: IF NPARTS (dump007) = 0
 roadchoice (4) = 0
 ENDIF

ELEMENT NAME: roadt_7
Element Type: Track
Quantity: 1
Capacity: 2
Zone: 0
Work Search: None
Input / Output Rules
Output: IF roadchoice (1) = 1 OR roadchoice (2) = 1 OR roadchoice (3) = 1
 PUSH to road7_6
 ELSEIF roadchoice (4) = 1
 PUSH to get7
 ELSE
 PUSH to road7_0
 ENDIF

ELEMENT NAME: dumpcart001
Element Type: Vehicle
Quantity: 1
Capacity: 1000
Speed (unloaded): 30.0
Speed (loaded): 25.0
Input / Output Rules
Output: PUSH to corppark(1)
Actions
Enter: cartcapacity = NFREE (dumpcart001)

ELEMENT NAME: dumpcart002
Element Type: Vehicle
Quantity: 1
Capacity: 1000
Speed (unloaded): 30.0
Speed (loaded): 25.0
Input / Output Rules
Output: PUSH to corppark(1)

Actions

Enter: cartcapacity = NFREE (dumpcart002)

ELEMENT NAME: dumpreusable

Element Type: Attribute

Quantity: 1

ELEMENT NAME: cartcapacity

Element Type: Variable

Quantity: 1

ELEMENT NAME: cartname

Element Type: Variable

Quantity: 1

ELEMENT NAME: collectfashion

Element Type: Variable

Quantity: 1

ELEMENT NAME: kinddump

Element Type: Variable

Quantity: 1

ELEMENT NAME: loadnum

Element Type: Variable

Quantity: 1

ELEMENT NAME: meandump

Element Type: Variable

Quantity: 7

ELEMENT NAME: moddemand

Element Type: Variable

Quantity: 7

ELEMENT NAME: num

Element Type: Variable

Quantity: 1

ELEMENT NAME: personum

Element Type: Variable

Quantity: 7

ELEMENT NAME: roadchoice

Element Type: Variable

Quantity: 4

ELEMENT NAME: wholecost

Element Type: Variable

Quantity: 1

ELEMENT NAME: objfun

Element Type: Function

Type: Real

Parameters: 0

Actions

```
Execution:  wholecost = 0  
            wholecost = DISTANCE (dumpcart001) / 10  
            RETURN wholecost
```
