```
! The dock or gate assignment problem. (GateAssign)
A number of vehicles (trucks, airplanes...)
arrive over time at an exchange facility.
The facility has a number of docks or gates.
Each gate can handle at most one vehicle at a time.
We would like to assign vehicles to gates so that
the value of the assignments is maximized, and
at most one vehicle is assigned to specific gate
at a specific instance.
 Additional realistic features not included:
a) allowing some vehicles to be slightly delayed,
b) not allowing certain, e.g., large, vehicles at adjacent gates,
c) value of assignment to a gate depend upon what other vehicles
are at nearby gates, e.g., if people or cargo are transferred
between vehicles;
! Keywords: Aircraft, Assignment, Depot, Dock assignment, Gate assignment, LINGO, Truck;
SETS:
  VEH: ARVT, LEVT;
  GATE;
  VXG( VEH, GATE) : VAL, Z;
ENDSETS
DATA:
 ! Vehicles, their arrival times, and their
   leave or departure times;
  VEH ARVT LEVT =
  V01  615  650
  V02  617  651
  V03  630  700
  V04  644  720
  V05  651  731
  V06  702  740
  V07  703  739
  V08  716  750
  V09  717  748
  V10  720  752;

! The available gates;
 GATE =
   G01 G02 G03 G04;
! The value of assigning vehicle v to gate g;
 VAL =
     3    4    4    3
     4    6    7    3
     5    5    5    5
     4    6    5    3
     2    6    7    3
     6    4    2    3
     3    6    7    3
     4    6    6    3
     1    9    7    2
     4    6    7    3;
ENDDATA
! Variables:
     Z( v, g) = 1 if vehicle v is assigned to gate g;

! Maximize the value of the assignments;
 MAX = @SUM( VXG(v,g): VAL( v, g)* Z( v, g));

! Each vehicle can be assigned to at most one gate;
 @FOR( VEH( v):
   @SUM( VXG( v, g): Z( v, g)) <= 1;
     );

! If vehicle v is assigned to gate g, starting at time ARVT(v),
  we must enforce that no other vehicle v2 is using the
  gate at that time;
! With a little bit of care, one can reduce the number
 of the following constraints that need be generated;
```

```
@FOR( VXG( v, g):
  ! Sum over all vehicles v2 arrived already and
    are still around at ARVT( v);
    @SUM( VXG( v2, g) | ARVT( v2) #LE# ARVT(v) #AND# LEVT( v2) #GT# ARVT(v): Z( v2, g)) <=
1;
    );

! The Z( v, g) are 0/1 variables;
@FOR( VXG( v, g):
    @BIN( Z( v, g));
    );
```