```
set PEOPLE;
set PROJECTS;

param supply {PEOPLE} >= 0;   # hours each person is available
param demand {PROJECTS} >= 0; # hours each project requires

   check: sum {i in PEOPLE} supply[i]
          = sum {j in PROJECTS} demand[j];

param cost {PEOPLE,PROJECTS} >= 0;   # cost per hour of work
param limit {PEOPLE,PROJECTS} >= 0;  # maximum contributions
                                     # to projects

var M;
var Assign {i in PEOPLE, j in PROJECTS} >= 0, <= limit[i,j];

minimize Max_Cost: M;

subject to M_def {i in PEOPLE}:
   M >= sum {j in PROJECTS} cost[i,j] * Assign[i,j];

subject to Supply {i in PEOPLE}:
   sum {j in PROJECTS} Assign[i,j] = supply[i];

subject to Demand {j in PROJECTS}:
   sum {i in PEOPLE} Assign[i,j] = demand[j];
```