All models are wrong, some are useful.

You are trying to transport goods from your 5 regional distribution centers to your 8 warehouses. The table below represents the transportation costs along with the capacities and demands. Furthermore, there is a 20000 cost to open any facility.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **distribution center\warehouse** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Capacity** |
| **1** | 3 | 4 | 6 | 8 | 9 | 1 | 3 | 5 | **1500** |
| **2** | 3 | 2 | 5 | 4 | 5 | 3 | 2 | 1 | **2000** |
| **3** | 9 | 8 | 6 | 4 | 5 | 6 | 7 | 8 | **1500** |
| **4** | 2 | 4 | 6 | 8 | 5 | 7 | 3 | 6 | **900** |
| **5** | 4 | 5 | 6 | 8 | 3 | 2 | 7 | 8 | **800** |
| **Demand** | **500** | **700** | **900** | **700** | **300** | **900** | **1200** | **300** |  |

**OPL 12.6.2, should work on several older versions too**

START COPYING WITH range

/\* You must set up a run configuration to run this. They are in the OPL project.

Don’t set up just the mode. Right click on run configuration and hit run this. \*/

range distribution = 1..5;

range warehouse = 1..8;

float cost[distribution, warehouse] = [[3, 4, 6, 8, 9, 1, 3, 5],

[3, 2, 5, 4, 5, 3, 2, 1],

[9, 8, 6, 4, 5, 6, 7, 8],

[2, 4, 6, 8, 5, 7, 3, 6],

[4, 5, 6, 8, 3, 2, 7, 8]];

float demand[warehouse] = [500, 700, 900, 700, 300, 900, 1200, 300];

float capacity[distribution] = [1500, 2000, 1500, 900, 800];

//Examples of setting up both a 1 d and 2 d array.

// three d follows the same format.

dvar float+ sent[distribution, warehouse] ;

dvar int+ open [distribution] in (0..1);

minimize sum(i in distribution) (sum (j in warehouse) cost[i,j]\*sent[i,j])+sum(i in distribution)20000\*open[i];

subject to {

forall (i in distribution) sum(j in warehouse) sent[i,j]<=capacity[i]\*open[i];

forall (j in warehouse:j<=8 && j>=1) sum(i in distribution) sent[i,j]>=demand[j];

};

// forall (i in distribution:i>2) sum(j in warehouse:j!=3)sent [i,j]>=0; // IF YOU WANT to sum or forall over a restricted set.

main {

if (!thisOplModel.generate()){ // generates this model as a solution.

if (cplex.solve()) {// If there exists a solution enter

writeln( "Objective Value ", cplex.getObjValue() ); //writes objective value

for ( var i in thisOplModel.distribution) { //loop over distribution

if (thisOplModel.open[i]>.99){ // If sent[i][j] NOTE: not [i,j] is one.

// NOTE: Due to round off error, never use ==1

writeln("Distribtion Center ", i, " is open ");

}

} ///Due to round off error, never use ==1

for ( var i in thisOplModel.distribution) { //loop of distribution

for ( var j in thisOplModel.warehouse) { //loop of warehouse

if (thisOplModel.sent[i][j]>.99){ // If sent[i][j] = 1

writeln("From Dist ",i," to ",j, " send ", thisOplModel.sent[i][j]," ");

}

}

}

} else {

writeln("Either Unbounded or Infeasible");

}

}else{

writeln("OPL could not read problem")

}

}

END COPYING AT THE }

This the print out from the scripting log.

Objective Value 95900

Distribtion Center 1 is open

Distribtion Center 2 is open

Distribtion Center 3 is open

Distribtion Center 4 is open

From Dist 1 to 6 send 900

From Dist 1 to 7 send 300

From Dist 2 to 2 send 700

From Dist 2 to 3 send 100

From Dist 2 to 7 send 900

From Dist 2 to 8 send 300

From Dist 3 to 3 send 800

From Dist 3 to 4 send 700

From Dist 4 to 1 send 500

From Dist 4 to 5 send 300