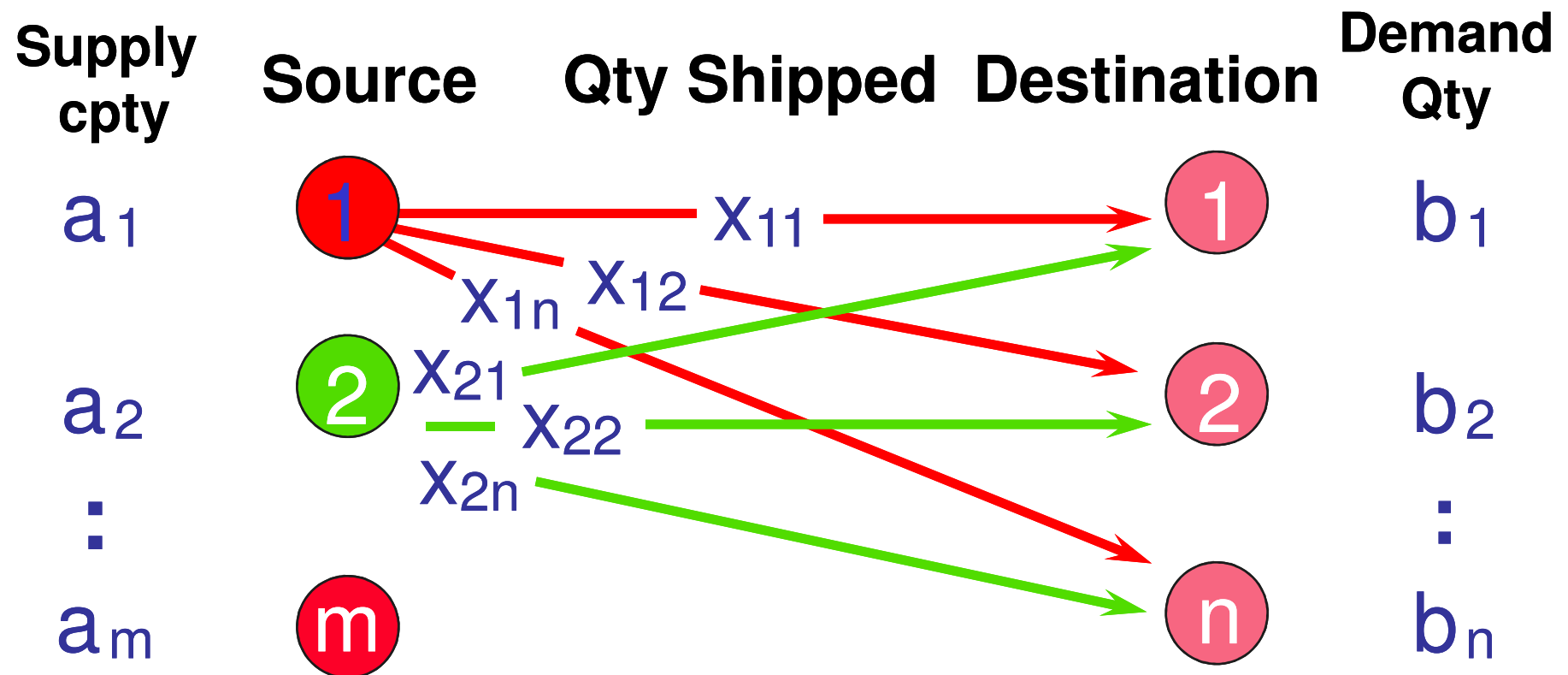


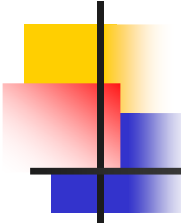
Examples for Discrete Constraint Programming

Belaid MOA
UVIC, SHRINC Project

A Tiny Transportation problem

- How much should be shipped from several sources to several destinations

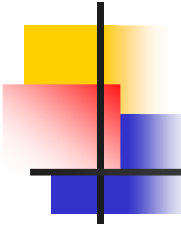




A Tiny Transportation problem

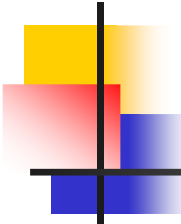
- 3 plants with known capacities, 4 clients with known demands and transport costs per unit between them.

500	①		①	200
300	②	Find qty shipped?	②	400
			③	300
400	③		④	100



Modeling TTP

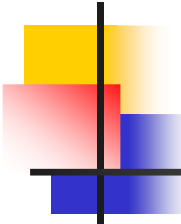
- TTP can be modeled by a CSP with optimization as follows:
 - Variables: $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, D_1, D_2, D_3$
 - Domain: $0.0..Inf$
 - constraints:
 - Demand constraints:
 $A_1 + A_2 + A_3 = 200; B_1 + B_2 + B_3 = 400; C_1 + C_2 + C_3 = 300; D_1 + D_2 + D_3 = 100;$
 - Capacity constraints:
 $A_1 + B_1 + C_1 + D_1 \leq 500; A_2 + B_2 + C_2 + D_2 \leq 300; A_3 + B_3 + C_3 + D_3 \leq 400;$
 - Minimization of the objective function:
 $10 \cdot A_1 + 7 \cdot A_2 + 11 \cdot A_3 + 8 \cdot B_1 + 5 \cdot B_2 + 10 \cdot B_3 +$
 $5 \cdot C_1 + 5 \cdot C_2 + 8 \cdot C_3 + 9 \cdot D_1 + 3 \cdot D_2 + 7 \cdot D_3$



Coding TTP

- OPL Studio - Cplex Solver

```
var float+ productA[Range];var float+ productB[Range];
var float+ productC[Range]; var float+ productD[Range];
minimize
    10*productA[1] + 7*productA[2] + 11*productA[3] +
      8*productB[1] + 5*productB[2] + 10*productB[3] +
      5*productC[1] + 5*productC[2] + 8*productC[3] +
      9*productD[1] + 3*productD[2] + 7*productD[3]
subject to {
    productA[1] + productA[2] + productA[3] = 200;
    productB[1] + productB[2] + productB[3] = 400;
    productC[1] + productC[2] + productC[3] = 300;
    productD[1] + productD[2] + productD[3] = 100;
    productA[1] + productB[1] + productC[1] + productD[1] <= 500
    productA[2] + productB[2] + productC[2] + productD[2] <= 300;
    productA[3] + productB[3] + productC[3] + productD[3] <= 400;
};
```



Coding TTP

- OPL Studio - Cplex Solver: Solution found

Optimal Solution with Objective Value: 6200.0000

productA[1] = 100.0000

productA[2] = 0.0000

productA[3] = 100.0000

productB[1] = 100.0000

productB[2] = 300.0000

productB[3] = 0.0000

productC[1] = 300.0000

productC[2] = 0.0000

productC[3] = 0.0000

productD[1] = 0.0000

productD[2] = 100.0000

productD[3] = 0.0000

Coding TTP

- ECLiPSe

`:- lib(eplex_cplex).`

`main1(Cost, Vars) :-`

`Vars = [A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3],`

`Vars :: 0.0..inf,`

`A1 + A2 + A3 $= 200, B1 + B2 + B3 $= 400,`

`C1 + C2 + C3 $= 300, D1 + D2 + D3 $= 100,`

`A1 + B1 + C1 + D1 $=< 500, A2 + B2 + C2 + D2 $=< 300,`

`A3 + B3 + C3 + D3 $=< 400,`

`optimize(min(10*A1 + 7*A2 + 11*A3 + 8*B1 + 5*B2 + 10*B3 +
5*C1 + 5*C2 + 8*C3 + 9*D1 + 3*D2 + 7*D3), Cost).`

It didn't run!



calling an undefined procedure `cplex_prob_init(...)`