

1.4 Beispiel: Paarungen

Situation:

- $n = 2k$ Leute sollen paarweise auf k Büros verteilt werden.
- Für jedes Paar haben wir einen Parameter $w \in \mathbb{R}$, der einschätzt, wie gut die beiden zusammen arbeiten können (je kleiner desto besser).
- Aufgabe: Finde eine Paarung, für welche die Summe der w -Werte minimal ist.

AMPL-Modell (paar.mod)

```
set PEOPLE;
set PAIRS within PEOPLE cross PEOPLE;
# data file: define for i>j only

param weight {PAIRS};

var x {PAIRS} binary;

minimize Total_Weight:
sum {(i,j) in PAIRS} weight[i,j] * x[i,j];

s.t. DegreeConstraint {i in PEOPLE}:
sum {(i,j) in PAIRS} x[i,j]
  + sum {(j,i) in PAIRS} x[j,i] = 1;
```

AMPL-Daten (paar.dat)

```
set PEOPLE := 1 2 3 4 5 6 7 8 9 10 ;
```

```
param: PAIRS: weight:
```

	1	2	3	4	5	6	7	8	9	10	:=
1
2	-5
3	-6	3
4	-3	-3	-4
5	-2	-9	6	-5
6	3	-9	1	-8	-4
7	7	0	10	-3	8	-2
8	9	-9	-9	-4	-3	-2	4
9	-10	0	-9	2	10	-10	-4	-6	.	.	.
10	-1	0	7	3	-5	9	-4	-5	0	.	;

Lösen mit AMPL

```
ampl: model paar.mod;
ampl: data paar.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal integer solution; objective -4
11 MIP simplex iterations
0 branch-and-bound nodes
No basis.
ampl: display x;
x [*,*]
:      1      2      3      4      5      6      7      8      9      :=
2      0      .      .      .      .      .      .      .
3      0      0      .      .      .      .      .      .
4      0      0      0      .      .      .      .      .
5      0      1      0      0      .      .      .      .
6      0      0      0      1      0      .      .      .
7      0      0      0      0      0      0      .      .
8      0      0      1      0      0      0      0      .
9      1      0      0      0      0      0      0      0
10     0      0      0      0      0      0      1      0      0
;
```

1.5 Beispiel: Reihenfolgeplanung

Situation:

- n Jobs müssen nacheinander auf einer Maschine ausgeführt werden.
- Es sind Umrüstzeiten $t_{i,j}$ bekannt, die benötigt werden, wenn Job j unmittelbar nach Job i ausgeführt wird.
- Zur Vereinfachung: Job 1 muss als erster ausgeführt werden, Job n als letzter.
- Aufgabe: Finde eine Reihenfolge der Jobs, welche die Summe der Umrüstzeiten minimiert.