Integer lineal model

Sets and parameters:

D	The number of days per month.
H	The number of hours per day.
T	Set of opening hours, index t.
E	Set of possible employees, index e.
ct_t	Number of customers per hour t.
p_t	Average time each customer spends at checkout station.
nCS_t	Number of checkout stations in time t.
I	Matrix for each pair of employees e_1 and e_2. $I[e_1][e_2]$ is 1, if these employees are incompatible and 0 otherwise.

Decision variables:

$hire_e$	binary, equal to 1, if employee $e$is hired or 0 otherwise.
x_{et}	binary, equal to 1, if employee e is assigned to task at time t or 0 otherwise.

The LP model for the problem is as follow:

$$minimize \sum_{e \in E} hire_e \tag{1}$$

Subject to:

$$hire_e \geq x_{et} \quad \forall e \in E, t \in T \tag{2}$$

$$hire_e \leq \sum_{t \in T} x_{et} \quad \forall e \in E \tag{3}$$

$$\sum_{e \in E} x_{et} == nCS_t \quad \forall t \in T \tag{4}$$

$$x_{e(h+d \cdot H)} + x_{e(h+d \cdot H+1)} + x_{e(h+d \cdot H+1)} \leq 2 \quad \forall e \in E, d \in \overline{0, D-1}, h \in \overline{1, H-2}$$

$$x_{e_1 t} + x_{e_2 t} + I[e_1][e_2] \leq 2 \quad \forall e \in E, t \in T \tag{6}$$

(1) - objective function: minimize the number of employees to hire;
(2) - the employee can be assigned to a task only if he (she) is hired;
(3) - if an employee is hired he (she) must be assigned to a task at least once;
(4) - the number of employees must be equal to the number of checkout station in each time t;
(5) - no employee can be assigned to a task for more than 2 consecutive hours;
(6) - if $I[e_1][e_2]$ is 1 employees e_1 and e_2 can not work simultaneously.

Experimental results

An integer linear programming model for the given problem is solved with IBM iLOG CPLEX Optimization Studio. As output the checkout station planning per employee has been got. The data used for the problem looks as follow:

Listing 1: **Example of Data File**

```
D = 5;
H = 8;
nEmloyees = 10;
nOpenHours = 40;

ct = [4 5 6 7 9 13 13 14 18 19 3 5 5 7 9 11
      12 14 16 19 5 6 7 8 11 11 15 16 20
      20 3 5 6 7 8 10 11 12 17 19];

pt = 10;

I = [[0 1 0 1 1 0 0 1 1 1],
     [1 0 1 1 0 1 0 1 0 0],
     [0 1 0 1 1 0 0 0 0 0],
     [1 1 1 0 1 1 0 1 1 1],
     [1 0 1 1 0 0 0 1 0 0],
     [0 1 0 1 0 0 0 1 1 1],
     [0 0 0 0 0 0 0 0 1 0],
     [1 1 0 1 1 1 0 0 0 1],
     [1 0 0 1 0 1 1 0 0 1],
     [1 0 0 1 0 1 0 1 1 0],];
```

The data was generated so as to be as much as possible similar to the real data. There was made some experiments fixing some parameters and changing others.