A7.2 - Industrial Control Panel

Jaan Tollander de Balsch - 452056

November 23, 2018

The linear program for the optimization task is formulated as

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} f(d_j, w_j) \cdot p_i \cdot x_{ij}$$

subject to

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad \forall j = 1..N$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i = 1..N$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1..N$$

where

- x_{ij} denotes if an item i is assigned to position j.
- p_i is the frequency distribution of the menu items. There are two conditions that must hold for a the distribution: $\sum_{i=1}^{N} p_i = 1$ and $p_i \geq 0$.
- $d_j \geq 0$ is the distance of position j from the start of the menu.

The function f is the Fitts' law[1] and defined as

$$f(D, W) = a + b \log_2 \left(D/W + 1 \right), \tag{fitts-law}$$

where D is the distance to the target, W is the width of the target and a and b are constants.

Because we are only interested in the optimal configuration of buttons not the value of the minimum solution, the parameters can be set $a = 0$ and $b = 1$. The targets are also assumed to have a constant width and they are set $W = 1$.

---

[1] https://en.wikipedia.org/wiki/Fitts%27s_law

Two widgets i and i_2 can be collocated to neighboring positions by constraining the assignments of positions

$$\sum_{j=1}^{N} c(j) = 1$$

where (x_{ij} is denoted as $x[i,j]$)

$$c(j) = \begin{cases} x[i,j] \cdot x[i_2, j_n] + x[i_2, j] \cdot x[i, j_n] & j_n \neq Nan \\ 0 & j_n = Nan \end{cases}$$

and $j_n = \text{neighbor}(j)$ is the neighboring position of j.

The neighbor function can be defined for both, elements on the same row and on the same column. For example, for 4×4 matrix

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix}$$

the neighboring positions for being in the same row are

$$0 \to 1, 1 \to 2, 2 \to 3, 3 \to Nan, 4 \to 5, ...$$

and for being in the same columns are

$$0 \to 4, 4 \to 8, 8 \to 12, 12 \to Nan, 1 \to 5, ...$$

.

Combined, within the code both of the constraints can be achieved by:

```
model.addConstr(
    quicksum([x[(e1, i*columns+j)] * x[(e2, i*columns+(j+1))] +
              x[(e2, i*columns+j)] * x[(e1, i*columns+(j+1))] +
              x[(e1, i * columns + j)] * x[(e2, (i + 1) * columns + j)] +
              x[(e2, i * columns + j)] * x[(e1, (i + 1) * columns + j)]
              for i in range(rows-1)
              for j in range(columns-1)]) == 1,
    f"Elements {e1}{e2} are colocated")
```