

1 Parameters

1.1 Data

- P : set of all persons (rowers),
- B : set of all boats,
- T : set of all training times of the week (day and time),
- B_p : set of boats that person p can row (based on her skill and weight class),
- T_p : set of times the person p is likely to row (union of her first and second choices),
- nb_asked_p : number of training person p wants to have in the week (number of first choices),
- E : set of sets of exclusive training times. Used for non-compatible training sessions, for examples training times that are directly consecutive,
- u_{pt} : utility for person p of rowing at time t . Values used are 1 if first choice, 0.1 if second choice, not defined if t not in choices of p .

1.2 Variables

- $s \in \mathbb{R}$: minimal utility across all people,
- $x_{pbt} \in \{0, 1\}$: binary variable, 1 if person p is scheduled to row boat b at time t , 0 otherwise.

2 Model

$$\max_{x,s} \quad \lambda s + \sum_{p \in P} \sum_{b \in B_p} \sum_{t \in T_p} u_{pt} x_{pbt} \quad (1)$$

$$\text{subject to:} \quad s \leq \sum_{b \in B_p} \sum_{t \in T_p} u_{pt} x_{pbt} \quad \forall p \in P \quad (2)$$

$$\sum_{b \in B_p} \sum_{t \in T_p} x_{pbt} \leq nb_asked_p \quad \forall p \in P \quad (3)$$

$$\sum_{b \in B_p} x_{pbt} \leq 1 \quad \forall p \in P, t \in T_p \quad (4)$$

$$\sum_{b \in B_p} \sum_{t \in E} x_{pbt} \leq 1 \quad \forall p \in P \quad (5)$$

$$\sum_{p \in P(b,t)} x_{pbt} \leq 1 \quad \forall b \in B, \forall t \in T \quad (6)$$

3 Explanations

This model is run weekly. A few days in advance, rowers submit their schedule preferences via an online form. These preferences consist of "first choices", that correspond to when a rowers would ideally want to row, and "second choices", which are backup times at which they would be available, in case it's not possible for them to row one of their "first choice".

More details about the objective function and the constraints:

- (1): The objective function is made of two components. The first one λs maximizes the lowest utility value that a person gets. The second one maximizes the sum of utilities across all people. The coefficient λ should be large enough to ensure that the maximum value of s is reached before optimizing the sum of utilities. This is to ensure fairness, i.e. make sure it's not possible that a small set of people receive a schedule with very low utility value, just to maximize the overall utility. The minimal value of this coefficient can be computed, but I've been using 500 which I am sure is enough for the size of my problem, and does not hit the CPU time in a significant way,
- (2): ensure that s is the minimum utility across people,
- (3): cap the max number of training at the number of trainings asked for each person. We don't want the utility to be increased by assigning more trainings to someone than they asked,
- (4): one person can only row one boat per training (otherwise, would increase utility artificially),
- (5): ensure that we don't assign someone to training that are mutually exclusive (according to E),
- (6): each boat can only accommodate one person per training.