1 Level 1: Goal Regression

Given a set of entities $e \in E$, preferences $R = \{r_1, r_2, r_3, \ldots, r_n\}$, and a value function over preferences $h(e, r) \in [0, 1]$, it is possible to produce an ideal match over the set of entities $q(e_i, e_j)$. To do this, a few functions need to be defined:

- properties, $P(e, r) \in \{0, 1\}$, denotes if an entity e has property r,
- value function $h(e, r) \in [0, 1]$ is the level of disappointment e will experience when matched with property r,
- directional quality of match $q(e_i, e_j)$, which expresses the disappointment e_i experiences when matched with e_j: $q(e_i, e_j) = \sum_{\{r|P(e_j,r)=1\}} h(e_i, r)$,
- a match indicator function, which indicates that a match has been assigned: $m(e_i, e_j) = \begin{cases} 1, & \text{match assigned \& } b(e_i, e_j) = 0 \\ 0, & \text{not matched} \end{cases}$ $(m(e_i, e_j) = m(e_j, e_i))$,
- a value $b(e_i, e_j) \in \{0, 1\}$ expresses if an entity e_i or e_j is too busy to be matched.

Given these match characteristics and quality values, two matrices may be maintained:

$$Q = \begin{bmatrix} q(e_1, e_1) & & \cdots & & \\ & \ddots & & & \\ \vdots & & q(e_i, e_j) & & \\ & & & \ddots & \vdots \\ & & \cdots & & q(e_n, e_n) \end{bmatrix} \tag{1}$$

and

$$M = \begin{bmatrix} m(e_1, e_1) & & \cdots & & \\ & \ddots & & & \\ \vdots & & m(e_i, e_j) & & \\ & & & \ddots & \vdots \\ & & \cdots & & m(e_n, e_n) \end{bmatrix}. \tag{2}$$

Given a set E, then the cost of a match arrangement M is

$$C(Q, M) = QM^T \tag{3}$$

and the optimization goal is to assign matches to unmatched entities:

$$M_{t+1} \leftarrow \frac{\arg\max_M \sum_{m \in M} M\alpha(Q, M)}{QM^T \left(1 - \alpha(Q, M)\right)} \tag{4}$$

where α is a scaling factor, $\alpha(Q, M) \in [0, 1]$, which can emphasize quantity of matches or quality of matches. Specifically:

$\alpha(Q, M) = 0 \rightarrow$ only the cost of matches is significant. Trivially, $\sum_{m \in M} m| = \varnothing$ will always satisfy this condition.

$\alpha(Q, M) = 1 \rightarrow$ only the amount of assignments matter. Trivially, $\sum_{m \in M} m = \sum_{m \in M} 1$ satisfies this requirement.

Thus, the ideal matching characteristics depend on an optimal value α. For algorithmic stability and quantity, low values of α should be attempted, which will prioritize quality over quantity.