ALGORITHM 2: RANDOMDICTATOR$_{k,q}$

Pick $i \in N$ uniformly at random
Let $X \leftarrow \mathsf{top}_q(i, N; d)$ // Ensure that $i \in X$
Pick $S \in \mathcal{S}_{k-q}(N \setminus X)$ uniformly at random
return $X \cup S$

A Proof of Theorem 3

Proof. Assume that $n > 2 \cdot \max\{\sqrt{kq/\epsilon}, k+1\}$, and let $m = \lfloor k/q \rfloor - 1$. Consider the real line, and suppose there are sets of q individuals at each position in $\{1, 2, \ldots, m\}$, denoted by $X_1, \ldots, X_m$, respectively, and the set of remaining $n - mq$ individuals, denoted by X_{m+1}, is at position $m+1$. The optimal panel P^* would have at least q people from each position, i.e., $|P^* \cap X_i| \geq q$ for all $i \in [m+1]$. The q-cost of each person for P^* is 0 as at least q people are selected from her own position. Hence, $\mathrm{SC}_q(P^*) = 0$.

Turning to the analysis of $\mathcal{A}_{k,q}$, we claim that

$$\mathbb{E}_{P \sim \mathcal{A}_{k,q}}[|X_{m+1} \cap P|] \geq q + (k \bmod q) + \epsilon. \tag{2}$$

To prove this, note that since each individual is included with a marginal probability of at least $\frac{q+(k \bmod q)}{n} + \epsilon$, we have

$$\begin{aligned}\mathbb{E}_{P \sim \mathcal{A}_{k,q}}[|X_{m+1} \cap P|] &\geq \left(\frac{q + (k \bmod q)}{n} + \epsilon\right)(n - mq) \\ &= q + (k \bmod q) - \frac{mq \cdot (q + (k \bmod q))}{n} + (n - mq) \cdot \epsilon\end{aligned}$$

We will show that the right hand side is at least $q+(k \bmod q)+\epsilon$. Because $mq < k$, $q+k \bmod q < 2q$, and $n - mq > n - k \geq n/2 + 1$, the right hand side is at least

$$q + (k \bmod q) - \frac{qk}{n/2} + n\epsilon/2 + \epsilon \geq q + (k \bmod q) + \epsilon,$$

where the inquality follows from our choice of $n > 2\sqrt{kq/\epsilon}$. This establishes Equation (2).

Now, as the panel size is k, it holds that $\sum_{i \in [m+1]} \mathbb{E}_{P \sim \mathcal{A}_{k,q}}[|X_i \cap P|] = k$. By Equation (2),

$$\sum_{i \in [m]} \mathbb{E}_{P \sim \mathcal{A}_{k,q}}[|X_i \cap P|] < k - (q + (k \bmod q)) - \epsilon = mq - \epsilon.$$

Therefore, there exists $i \in [m]$ such that $\mathbb{E}_{P \sim \mathcal{A}_{k,q}}[|X_i \cap P|] \leq q - \epsilon/q$. Using Markov's inequality,

$$\Pr_{P \sim \mathcal{A}_{k,q}}(|X_i \cap P| \geq q) \leq \tfrac{q - \epsilon/q}{q} \leq 1 - \epsilon.$$

Thus, with probability at least ϵ, less than q people are selected from position i, in which case the q-cost of each person in X_i will be at least 1. Hence, $\mathbb{E}_{P \sim \mathcal{A}_{k,q}}[\mathrm{SC}_q(\mathcal{A}_{k,q})] \geq q\epsilon$ while $\mathrm{SC}_q(P^*) = 0$. □

B Tradeoffs between Representation and Fairness

We start with the case of $q > k/2$ and show that a simple algorithm, which is a variant of the natural *random dictatorship* rule, provides constant representation by sacrificing some quantity of perfect fairness. Specifically, the algorithm RANDOMDICTATOR$_{k,q}$, presented as Algorithm 2, works as follows: Given an instance d, it chooses an individual i from the underlying population uniformly at random, and returns the panel $P = \mathsf{top}_q(i, N; d) \cup S$, where $\mathsf{top}_q(i, N; d)$ is the set of q people closest to i (we break ties in a way to ensure that this contains i herself), and S is a panel of size $k - q$ chosen uniformly at random from the remaining people.

Theorem 6. *For any $q > k/2$, it holds that*

$$\mathsf{repr}_q(\text{RANDOMDICTATOR}_{k,q}) \geq \frac{1}{3} \quad \textit{and} \quad \mathsf{fairness}(\text{RANDOMDICTATOR}_{k,q}) \geq \frac{k - q + 1}{k}.$$

Proof. We start by proving the fairness guarantee of the algorithm. Note that each individual i is included in the panel P returned by $\text{RANDOMDICTATOR}_{k,q}$ either if i is selected at the first step, which happens with probability $1/n$, or if i is not selected in the first step, it is selected in the second step with probability at least $(k-q)/(n-q)$. Hence, the probability of i being selected is at least $(1/n) + (1-1/n) \cdot (k-q)/(n-q) \geq (k-q+1)/n$, yielding $\mathsf{fairness}(\text{RANDOMDICTATOR}_{k,q}) \geq (k-q+1)/k$.

For $q > k/2$, Caragiannis et al. [13] (Corollary 2) show that random dictatorship, i.e. returning a panel minimizing q-cost to a randomly selected individual i, achieves a representation of at least $1/3$. The panel returned by $\text{RANDOMDICTATOR}_{k,q}$ consists of q closest neighbors of i which obtains the minimum q-cost with respect to i, and fills the other $k-q$ members of this panel randomly which does not affect the q-cost of the returned panel to i. Hence, $\text{RANDOMDICTATOR}_{k,q}$ can be seen as a variant of the *random dictatorship* rule which randomly breaks ties between top panel choices of a randomly selected individual. □

Now, we turn our attention to the case that $q \leq k/2$. In Section 4, we introduced RANDOMREPLACE_q with fairness q/k and representation $\alpha/(q+1)$ if given an α-representative panel. In fact, if we replace q with any $r \in [q]$, we can show that the algorithm provides representation of at least $\alpha/(r+1)$ with fairness r/k. Essentially, RANDOMREPLACE_r chooses a subset S of the underlying population with size r instead of q uniformly at random in Line 1 of Algorithm 1.

Proposition 1. *For any $q \in [k]$, $r \in [q]$, and panel P with $\mathsf{repr}_q(P) = \alpha$ it holds that*

$$\mathsf{repr}_q(\text{RANDOMREPLACE}_r(P)) \geq \frac{\alpha}{r+1} \quad \textit{and} \quad \mathsf{fairness}(\text{RANDOMREPLACE}_r) \geq \frac{r}{k}.$$

We omit the proof of this proposition as it is essentially identical to the proof of Theorem 4 with q replaced by r in the appropriate places.

C Average Cost Function

Let $c_{\text{avg}}(i, P) = \frac{1}{k} \sum_{j \in P} d(i,j)$ denote the average cost of panel P of size k to an individual i. Similarly, define $\text{SC}_{\text{avg}}(P) = \sum_{i \in N} c_{\text{avg}}(i, P)$, and let $\mathsf{repr}_{\text{avg}}(\mathcal{A}_k)$ denote the representation of a selection algorithm $\mathcal{A}_k$ with respect to the average cost function. It turns out that uniform selection (or any algorithm with perfect fairness) performs very well with the $\mathsf{repr}_{\text{avg}}$ objective and achieves a representation of $1/2$.

Proposition 2. *For all $k \geq 1$, uniform selection satisfies $\mathsf{repr}_{\text{avg}}(\mathcal{U}_k) > 1/2$.*

Proof. Sort the population as $N = (i_1, i_2, \ldots, i_n)$ in a non-decreasing order of $\text{SC}(i_\ell) = \sum_{i \in N} d(i, i_\ell)$, so that $\text{SC}(i_1) \leq \text{SC}(i_2) \leq \ldots \leq \text{SC}(i_n)$. Note that for any panel P, $\text{SC}_{\text{avg}}(P) = \frac{1}{k} \sum_{i \in P} \text{SC}(i)$, so the optimal panel is $P^* = \{i_1, \ldots, i_k\}$. Then,

$$\text{SC}_{\text{avg}}(P^*) = \frac{1}{k} \sum_{i_\ell \in P^*} \text{SC}(i_\ell) \geq \min_{i_\ell \in P^*} \text{SC}(i_\ell) = \text{SC}(i_1).$$

Note that $\{i_1\} = \min_{P' \in S_{k=1}(N)} \text{SC}_{q=1}(P')$ is the optimal panel for the case where $q = k = 1$. As $q > k/2$ in this scenario, by Lemma 1, we have

$$\text{SC}_{\text{avg}}(P^*) \geq \text{SC}(i_1) = \text{SC}_1(\{i_1\}) \geq \frac{1}{2(n-1)} \sum_{i \in N} \sum_{j \in N \setminus \{i\}} d(i,j).$$

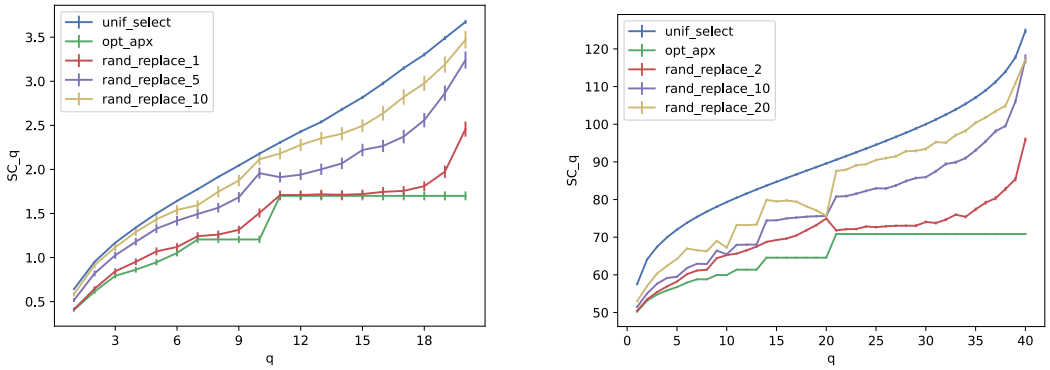
The average social cost of uniform selection is

$$\begin{aligned} \mathbb{E}[\text{SC}_{\text{avg}}(\mathcal{U}_k(N))] &= \frac{1}{k} \sum_{i \in N} \sum_{j \in N} d(i,j) \cdot \Pr_{P \sim \mathcal{U}_k}[j \in P] \\ &= \frac{1}{k} \sum_{i \in N} \sum_{j \in N \setminus \{i\}} d(i,j) \cdot \frac{k}{n}, \end{aligned}$$

where in the last transition we used the fact that $d(i,i) = 0$ and the marginal inclusion probabilities are equal to k/n. Putting all together, we have that $\mathsf{repr}_{\text{avg}}(\mathcal{U}_k) \geq \frac{n}{2(n-1)} > \frac{1}{2}$. □

D Experimental Results

D.1 Experiment Plots



(a) `Adult` with $k = 20$.

(b) `ESS` UK with $k = 40$.

Figure 4: Comparison of different algorithm for fixed k, where RANDOMREPLACE$_r$ is applied to the panel selected by OPTPROXY. As r ranges from 0 to k, the q-social cost of RANDOMREPLACE$_r$ interpolates between that of OPTPROXY and UNIFORMSELECTION.

D.2 Computation Time of OPTPROXY

The most computationally expensive task in our experiment is computing OPTPROXY, which is used as a subroutine in the RANDOMREPLACE method. We report the running time of our implementation, as described in Section 5, for low to very high values of k. We used the ESS UK dataset with $n = 2204$ and a standard laptop (quad-core, 1.7GHz CPU, 16GB RAM). The computation took less than 15 seconds for $k = 100$, which includes computing the pair-wise distance matrix (5 seconds) and running OPTPROXY (7 seconds). OPTPROXY itself scales well for practical values of k as shown in Table 1.

Table 1: Runtime of OPTPROXY on `ESS` UK with different values of k. The reported time is the maximum over all $q \in [k]$.

Panel size k	Runtime
20	$<$ 4s
100	$<$ 7s
200	$<$ 10s
500	$<$ 20s