

Appendix A: Further Details on FCFS and Greedy Policies

A.1. Further Details on FCFS

The pseudocode for FCFS is presented in Alg. 3. It takes an arbitrary participation decision $\mathbf{y}$ as an input and outputs the allocations $\mathbf{x}^{\text{FCFS}}(\mathbf{y})$. For simplicity, we assume workers arrive uniformly at random (Line 6). The bandwidth preference (2) of worker $i \in [N]$ is accounted for in Lines 5 and 8. The randomness in $\mathbf{x}^{\text{FCFS}}$ is driven by Line 4 (J_t is random) and Line 6 (workers arrive randomly). To map to the §2 notation, note that $\mathbb{W}_{j_t}$ depends on history $\mathcal{H}_{j_t}$ and Lines 6 and 7 are equivalent to $[z_{ijt}^{\text{FCFS}}]_{i \in \mathbb{W}_{j_t}} \sim \text{Categorical}(1/|\mathbb{W}_{j_t}|, \dots, 1/|\mathbb{W}_{j_t}|)$.

Algorithm 3 FCFS($\mathbf{y}$)

```

1: for  $t = 1$  to  $T$  do
2:    $x_{it}^{\text{FCFS}} = 0$  for all  $i \in [N]$                                 % initialize allocations for this period
3:    $b_{it} = b_i$  for all  $i \in [N]$                                     % initialize bandwidth for this period
4:   for  $j = 1$  to  $J_t$  do
5:      $\mathbb{W}_{j_t} = \{i : y_i = 1, b_{it} \geq 1\}$                         % active workers with bandwidth
6:      $i^* \sim \text{Uniform}(\mathbb{W}_{j_t})$                                     % random worker arrives first
7:      $x_{i^*t}^{\text{FCFS}} \leftarrow x_{i^*t}^{\text{FCFS}} + 1$                         % allocate
8:      $b_{i^*t} \leftarrow b_{i^*t} - 1$                                 % remaining bandwidth
9:   end for
10: end for
11: return  $\mathbf{x}^{\text{FCFS}}$ 

```

Example 5. There are N workers with $a_i = a$ and $b_i = b$ for all i . The quality $\mathbf{q}$ is arbitrary and so is the demand J (subscript t omitted). Under FCFS, if n workers are active, then the total bandwidth equals nb and $\bar{x}_i^{\text{FCFS}} = \mathbb{E}[\min\{J, nb\}]/n$ for all i that are active. Hence, under every FCFS equilibrium, exactly n^* workers will be active where $n^* := \max\{n : \mathbb{E}[\min\{J, nb\}]/n \geq a\}$. Any n^* (out of N) workers form an equilibrium and hence, there are $\binom{N}{n^*}$ possible equilibria. However, the optimal market consists of the top few workers w.r.t. $\mathbf{q}$. Clearly, FCFS would be sub-optimal in almost every equilibrium.

A.2. Further Details on Greedy

The pseudocode for **Greedy** is presented in Alg. 4. It takes an arbitrary participation decision $\mathbf{y}$ as an input and outputs the allocations $\mathbf{x}^{\text{greedy}}(\mathbf{y})$. It is very similar to FCFS except Line 6. Unlike FCFS, it is possible to characterize the equilibrium under **Greedy**. We do so in Prop. 7.

Proposition 7. For an arbitrary $(\mathbf{a}, \mathbf{b}, \mathbf{q})$ with $q_1 > \dots > q_N$, the unique **Greedy** equilibrium $\mathbf{y}^{\text{greedy}}$ and the corresponding average allocations $\bar{\mathbf{x}}^{\text{greedy}}$ are as follows: for all $i \in [N]$,

$$\begin{aligned}
y_i^{\text{greedy}} &= \mathbb{I} \left\{ h_i(y_1^{\text{greedy}}, \dots, y_{i-1}^{\text{greedy}}, 1) \geq a_i \right\} \\
\bar{x}_i^{\text{greedy}} &= h_i(y_1^{\text{greedy}}, \dots, y_i^{\text{greedy}}),
\end{aligned}$$

where

$$h_i(y_1, \dots, y_i) := \mathbb{E} \left[\min \left\{ J, \sum_{k=1}^i b_k y_k \right\} \right] - \sum_{k=1}^{i-1} h_k(y_1, \dots, y_k) \quad \forall i \in [N]. \quad (19)$$

Algorithm 4 Greedy($\mathbf{y}$)

```

1: for  $t = 1$  to  $T$  do
2:    $x_{it}^{\text{greedy}} = 0$  for all  $i \in [N]$                                 % initialize allocations for this period
3:    $b_{it} = b_i$  for all  $i \in [N]$                                 % initialize bandwidth for this period
4:   for  $j = 1$  to  $J_t$  do
5:      $\mathbb{W}_{jt} = \{i : y_i = 1, b_{it} \geq 1\}$                         % active workers with bandwidth
6:      $i^* \sim \arg \max_{i \in \mathbb{W}_{jt}} q_i$                                 % pick a worker with highest quality
7:      $x_{i^*t}^{\text{greedy}} \leftarrow x_{i^*t}^{\text{greedy}} + 1$                     % allocate
8:      $b_{i^*t} \leftarrow b_{i^*t} - 1$                                 % remaining bandwidth
9:   end for
10: end for
11: return  $\mathbf{x}^{\text{greedy}}$ 

```

Proof. We have $q_1 > \dots > q_N$ (strict). Consider worker 1 and observe that she receives priority over every other worker and hence, her average allocation $\bar{x}_1^{\text{greedy}}$ as a function of y_1 equals

$$h_1^{\text{greedy}}(y_1) = \mathbb{E}[\min\{J, b_1 y_1\}]. \quad (20a)$$

This holds irrespective of which other workers are active. Thus, if $h_1^{\text{greedy}}(1) \geq a_1$, then $y_1^{\text{greedy}} = 1$:

$$y_1^{\text{greedy}} = \mathbb{I}\left\{h_1^{\text{greedy}}(1) \geq a_1\right\}. \quad (20b)$$

For worker 2, observe that her average allocation $\bar{x}_2^{\text{greedy}}$ as a function of (y_1, y_2) equals

$$\begin{aligned} h_2^{\text{greedy}}(y_1, y_2) &= \mathbb{E}[\min\{J, b_1 y_1 + b_2 y_2\} - \min\{J, b_1 y_1\}] \\ &= \mathbb{E}[\min\{J, b_1 y_1 + b_2 y_2\}] - h_1^{\text{greedy}}(y_1). \end{aligned}$$

This is irrespective of whether workers 3 to N are active and thus, $y_2^{\text{greedy}} = \mathbb{I}\left\{h_2^{\text{greedy}}(y_1^{\text{greedy}}, 1) \geq a_2\right\}$. In general, for worker i , we have the following recursive relationship with base case in (20):

$$\begin{aligned} h_i(y_1, \dots, y_i) &= \mathbb{E}\left[\min\left\{J, \sum_{k=1}^i b_k y_k\right\}\right] - \sum_{k=1}^{i-1} h_k(y_1, \dots, y_k) \\ y_i^{\text{greedy}} &= \mathbb{I}\left\{h_i(y_1^{\text{greedy}}, \dots, y_{i-1}^{\text{greedy}}, 1) \geq a_i\right\}. \end{aligned}$$

Worker i 's allocation is independent of whether workers $i+1$ to N are active. Hence, when $q_1 > \dots > q_N$, there exists exactly one **Greedy** equilibrium. $\square$

In Prop. 7, the sorting of q_i 's is w.l.o.g. but we assume $q_i \neq q_{i'}$ for $i \neq i'$ for ease of notation and uniqueness. We can relax this assumption and generalize Prop. 7 but then, the uniqueness might not be in the “workers space” but in the “quality space”. It will depend on the tie-breaking rule in Line 6 of Alg. 4. We leverage Prop. 7 in our §6 numerics to efficiently compute the **Greedy** equilibrium. In fact, one pass over the N workers (sorted w.r.t. $\mathbf{q}$) is sufficient to compute this equilibrium, as we summarize in Alg. 5. Furthermore, when doing the computation for worker i in Alg. 5, we can recycle $[\bar{x}_1^{\text{greedy}}, \dots, \bar{x}_{i-1}^{\text{greedy}}]$ from the previous $i-1$ iterations.

Algorithm 5 GreedyEquilibrium($\mathbf{a}, \mathbf{b}, \mathbf{q}$)

```

1: Sort workers such that  $q_1 > \dots > q_N$ 
2: for  $i = 1$  to  $N$  do
3:    $y_i^{\text{greedy}} = \mathbb{I} \left\{ h_i(y_1^{\text{greedy}}, \dots, y_{i-1}^{\text{greedy}}, 1) \geq a_i \right\}$  where  $h_i(\cdot)$  as in (19)
4:    $\bar{x}_i^{\text{greedy}} = h_i(y_1^{\text{greedy}}, \dots, y_i^{\text{greedy}})$ 
5: end for
6: return  $(\bar{\mathbf{x}}^{\text{greedy}}, \mathbf{y}^{\text{greedy}})$ 

```

Proposition 2. Under *Greedy*, the quality ratio can be arbitrarily low, i.e., for all $\epsilon > 0$, there exists a market such that under every *Greedy* equilibrium $\mathbf{y}^{\text{greedy}}$, $q^{\text{greedy}}/q^* < \epsilon$.

Proof. It suffices to construct an appropriate market. Suppose the demand for each period t is as follows for $m \in \mathbb{N}$:

$$J_t = \begin{cases} 1 & \text{w.p. } p \\ m & \text{w.p. } 1 - p. \end{cases}$$

There are $N = 2$ workers with $(a_1, b_1, q_1) = (1, 1, 1)$ and $(a_2, b_2, q_2) = ((1-p)m, m, 0.9)$. (We can ensure $a_2 \in \mathbb{N}$ by restricting (m, p) such that $(1-p)m \in \mathbb{N}$.) Worker 1 is (slightly) higher quality but worker 2 provides more bandwidth. Under *Greedy*, worker 1 is prioritized over worker 2 since $q_1 > q_2$. This results in worker 1 being active ($y_1^{\text{greedy}} = 1$) with $x_{1t}^{\text{greedy}} = 1$ w.p. 1 for all t (irrespective of y_2). Given this, if worker 2 is active, her allocation in each period t would be

$$x_{2t}^{\text{greedy}} = \begin{cases} 0 & \text{w.p. } p \\ m - 1 & \text{w.p. } 1 - p. \end{cases}$$

This implies a per-period average allocation of $(1-p)(m-1)$, which is less than a_2 . Hence, worker 2 does not participate, i.e., $y_2^{\text{greedy}} = 0$. Thus, only worker 1 is active (unique equilibrium) and $q^{\text{greedy}} = 1$. Consider an alternative online allocation policy under which worker 2 is prioritized. Then, only worker 2 will be active (unique equilibrium) with period t allocation as follows: $x_{2t} = J_t$. Hence, the expected allocation to worker 2 is $p + (1-p)m$, which is greater than a_2 . The objective equals $0.9(p + (1-p)m)$, which is at most q^* (by the optimality of q^*). Hence, the quality ratio under *Greedy* can be bounded as follows:

$$\frac{q^{\text{greedy}}}{q^*} = \frac{1}{q^*} \leq \frac{1}{0.9(p + (1-p)m)}.$$

Finally, given arbitrary $\epsilon > 0$, if $m > \frac{1-0.9p\epsilon}{0.9(1-p)\epsilon}$, then

$$\frac{1}{0.9(p + (1-p)m)} < \epsilon.$$

Hence, for all $\epsilon > 0$, there exists m such that the quality ratio is below ϵ . $\square$

Remark 2. Observe that in the example constructed above, only worker 2 being active is an equilibrium under FCFS as well. In fact, it will be unique if $p \geq 2/3$: irrespective of whether worker 1 is active or not, worker 2 will receive an average per-period allocation of at least $p(1/2) + (1-p)(m-1)$, which exceeds her min. requirement of $(1-p)m$ if $p \geq 2/3$. Finally, worker 2 being active will result in worker 1 being not active, making the equilibrium unique. Thus, FCFS can result in an arbitrarily better equilibrium than *Greedy*.

Example 6. There are $N = 2$ workers with $a_1 = a_2 = 4$, $b_1 = b_2 = 5$, and $(q_1, q_2) = (1, 1 - \epsilon)$, where $\epsilon > 0$ is small. Suppose the demand in each period is $J = 8$ (deterministic). Under **Greedy**, only worker 1 is active, receiving 5 jobs per period: $q^{\text{greedy}} = 5$. However, an optimal policy allocates 4 jobs to each worker: $q^* = 8 - 4\epsilon$. Similar to Example 2, the driving force here is **Greedy** blindly prioritizing quality, which leads to bandwidth churn as worker 2 leaves the market.

Appendix B: Proofs for §4

B.1. Proof of Lemma 1

Lemma 1. (10) provides an upper bound to (7), i.e., $q^* \leq q^{\text{det}}$.

Proof. Recall the optimization problem (7):

$$\max_{\mathbf{P}} \mathbb{E} \left[\sum_{i=1}^N q_i \bar{x}_i^{\mathbf{P}} \right] \quad (7a)$$

$$\text{s.t. } x_{it}^{\mathbf{P}} = \sum_{j=1}^{J_t} z_{ijt}^{\mathbf{P}} \quad \forall i \in [N] \quad \forall t \quad (7b)$$

$$\sum_{i=1}^N z_{ijt}^{\mathbf{P}} \leq 1 \quad \forall j \in [J_t] \quad \forall t \quad (7c)$$

$$\bar{x}_i^{\mathbf{P}} = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\mathbf{P}} \quad \forall i \in [N] \quad (7d)$$

$$\bar{x}_i^{\mathbf{P}} \geq a_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \quad (7e)$$

$$x_{it}^{\mathbf{P}} \leq b_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \quad \forall t \quad (7f)$$

$$y_i^{\mathbf{P}}, z_{ijt}^{\mathbf{P}} \in \{0, 1\} \quad \forall i \in [N] \quad \forall j \in [J_t] \quad \forall t. \quad (7g)$$

First, use (7b) to eliminate $\mathbf{z}^{\mathbf{P}}$ variables from (7) to obtain the following relaxation:

$$\max_{\mathbf{P}} \mathbb{E} \left[\sum_{i=1}^N q_i \bar{x}_i^{\mathbf{P}} \right] \quad (22a)$$

$$\text{s.t. } \sum_{i=1}^N x_{it}^{\mathbf{P}} \leq J_t \text{ w.p. } 1 \quad \forall t \quad (22b)$$

$$\bar{x}_i^{\mathbf{P}} = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\mathbf{P}} \quad \forall i \in [N] \quad (22c)$$

$$\bar{x}_i^{\mathbf{P}} \geq a_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \quad (22d)$$

$$x_{it}^{\mathbf{P}} \leq b_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \quad \forall t \quad (22e)$$

$$y_i^{\mathbf{P}} \in \{0, 1\}, x_{it}^{\mathbf{P}} \in \mathbb{Z} \quad \forall i \in [N] \quad \forall t. \quad (22f)$$

It is a relaxation because $\sum_i z_{ijt}^{\mathbf{P}} \leq 1 \quad \forall (j, t)$ implies $\sum_i x_{it}^{\mathbf{P}} \leq J_t \quad \forall t$. Second, observe that (10) simply relaxes all the w.p. 1 constraints of (22) to hold only in expectation:

$$\max_{\boldsymbol{\pi}, \mathbf{y}} \sum_{i=1}^N q_i \pi_i \quad (10a)$$

$$\text{s.t. } \sum_{i=1}^N \pi_i \leq \mu \quad (10b)$$

$$\pi_i \geq a_i y_i \quad \forall i \in [N] \quad (10c)$$

$$\pi_i \leq b_i y_i \quad \forall i \in [N] \quad (10d)$$

$$y_i \in \{0, 1\} \quad \forall i \in [N]. \quad (10e)$$

The proof is now complete. $\square$

B.2. Proof of Lemma 2

Lemma 2. (11) provides a tighter upper bound to (7) than (10), i.e., $q^* \leq q^{\text{relax}} \leq q^{\text{det}}$.

Proof. $q^{\text{relax}} \leq q^{\text{det}}$ is discussed in the main body (§4.1) and we establish $q^* \leq q^{\text{relax}}$ here. Recall that (22) (from the proof of Lemma 1) is a relaxation of (7). Consider the following modification of (22) (see (23b) and (23e)):

$$\max_{\mathbf{P}} \mathbb{E} \left[\sum_{i=1}^N q_i \bar{x}_i^{\mathbf{P}} \right] \tag{23a}$$

$$\text{s.t. } \sum_{i=1}^N x_{it}^{\mathbf{P}} \leq \min \left\{ J_t, \sum_{i=1}^N b_i y_i^{\mathbf{P}} \right\} \text{ w.p. } 1 \quad \forall t \tag{23b}$$

$$\bar{x}_i^{\mathbf{P}} = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\mathbf{P}} \quad \forall i \in [N] \tag{23c}$$

$$\bar{x}_i^{\mathbf{P}} \geq a_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \tag{23d}$$

$$x_{it}^{\mathbf{P}} \leq \min \{ J_t, b_i \} y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \quad \forall t \tag{23e}$$

$$y_i^{\mathbf{P}} \in \{0, 1\}, x_{it}^{\mathbf{P}} \in \mathbb{Z} \quad \forall i \in [N] \quad \forall t. \tag{23f}$$

Given the presence of (22b) and (22e), this modification is redundant. Hence, (23) is equivalent to (22), meaning (23) is also a relaxation of (7). Relaxing (23) by replacing the w.p. 1 constraints to hold only in expectation gives us (11). This completes the proof. $\square$

B.3. Proof of Lemma 3

Lemma 3. (13) provides a tighter upper bound to (7) than (11), i.e., $q^* \leq q^{**} \leq q^{\text{relax}} \leq q^{\text{det}}$.

Proof. $q^{**} \leq q^{\text{relax}}$ is discussed in the main body (§4.2), $q^{\text{relax}} \leq q^{\text{det}}$ in Lemma 2, and we establish $q^* \leq q^{**}$ here. Recall that (23) (from the proof of Lemma 2) is a relaxation of (7). We now add redundant constraints to (23) (see (24c)), which results in the following relaxation of (7):

$$\max_{\mathbf{P}} \mathbb{E} \left[\sum_{i=1}^N q_i \bar{x}_i^{\mathbf{P}} \right] \tag{24a}$$

$$\text{s.t. } \sum_{i=1}^N x_{it}^{\mathbf{P}} \leq \min \left\{ J_t, \sum_{i=1}^N b_i y_i^{\mathbf{P}} \right\} \text{ w.p. } 1 \quad \forall t \tag{24b}$$

$$\sum_{i \in \mathbb{W}} x_{it}^{\mathbf{P}} \leq \min \left\{ J_t, \sum_{i \in \mathbb{W}} b_i y_i^{\mathbf{P}} \right\} \text{ w.p. } 1 \quad \forall t \quad \forall \mathbb{W} \subset [N] \tag{24c}$$

$$\bar{x}_i^{\mathbf{P}} = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\mathbf{P}} \quad \forall i \in [N] \tag{24d}$$

$$\bar{x}_i^{\mathbf{P}} \geq a_i y_i^{\mathbf{P}} \text{ w.p. } 1 \quad \forall i \in [N] \tag{24e}$$

$$y_i^{\mathbf{P}} \in \{0, 1\}, x_{it}^{\mathbf{P}} \in \mathbb{Z} \quad \forall i \in [N] \quad \forall t. \tag{24f}$$

(24c) enumerates all $2^N - 1$ proper subsets of $[N]$ and generalizes (23e). For example, when $\mathbb{W} = \{i\}$ (for arbitrary $i \in [N]$), then (24c) recovers (23e). Given the presence of (23e), the modification in (24) is redundant. Hence, (24) is equivalent to (23), meaning (24) is also a relaxation of (7). Relaxing (24) by replacing the w.p. 1 constraints to hold only in expectation gives us (13). This completes the proof. $\square$

B.4. Proof of Theorem 1

Theorem 1. (13) is tight, i.e., $q^{**} = q^*$.

Proof. We proceed in two steps. First, we reformulate (13) into an equivalent optimization where each worker has a bandwidth of 1. Second, we establish the reformulation is tight.

Step 1. Recall $b_i \in \{1, 2, \dots, b_{\max}\}$ is assumed to be an integer for all i , where $b_{\max}$ is the maximum bandwidth (assumed finite). To formulate an equivalent optimization but with a bandwidth equal to 1 for each worker, we split each worker i into b_i workers with the following parameters:

$$a_i^{(k)} := \frac{a_i}{b_i}, \quad b_i^{(k)} := \frac{b_i}{b_i} = 1, \quad q_i^{(k)} := q_i \quad \forall k \in [b_i]. \quad (25)$$

That is, the min. requirement and bandwidths are normalized by b_i but the quality remains the same. Now, consider the following optimization:

$$\max_{\pi, \mathbf{y}} \sum_{i=1}^N \sum_{k=1}^{b_i} q_i^{(k)} \pi_i^{(k)} \quad (26a)$$

$$\text{s.t.} \quad \sum_{i \in \mathbb{W}} \sum_{k \in \mathbb{B}_i} \pi_i^{(k)} \leq \mathbb{E} \left[\min \left\{ J, \sum_{i \in \mathbb{W}} \sum_{k \in \mathbb{B}_i} y_i^{(k)} \right\} \right] \quad \forall \mathbb{W} \subseteq [N] \quad \forall \mathbb{B}_i \subseteq [b_i] \quad (26b)$$

$$\pi_i^{(k)} \geq a_i^{(k)} y_i^{(k)} \quad \forall i \in [N] \quad \forall k \in [b_i] \quad (26c)$$

$$y_i^{(k)} \in \{0, 1\} \quad \forall i \in [N] \quad \forall k \in [b_i] \quad (26d)$$

$$y_i^{(k)} = y_i^{(k+1)} \quad \forall i \in [N] \quad \forall k \in [b_i - 1]. \quad (26e)$$

(26) is analogous to (13) but with two differences. First, instead of N workers, it has $\sum_{i=1}^N b_i$ workers with parameters as in (25). Second, we have added (26e) so that we do not use “half” of any worker i . It should be clear that (26) is tighter than (13) since it has all the constraints as (13) and more (i.e., a reduced feasibility set). In Lemma 5 below, we prove that (26) is in fact equivalent to (13). The key observation we leverage is that the RHS in (26b) is concave w.r.t. the bandwidth.

Step 2. Consider an arbitrary feasible solution $(\pi, \mathbf{y})$ to (26). It suffices to show that there exists an online allocation algorithm that achieves π when the allocations are averaged over the infinite horizon. This directly follows Lemma 2 of [Aouad and Ma \(2026\)](#), where they show the randomized “Lossless Rounding” algorithm achieves such a π in expectation. To achieve it almost surely over the infinite horizon, one can use their “Lossless Rounding” algorithm in each period (i.i.d. across periods) and invoke the law of large numbers. The proof is now complete. $\square$

Lemma 5. (26) is equivalent to (13).

Proof. As (26)’s feasibility set is a subset of (13)’s, it suffices to show that given a feasible solution $(\pi, \mathbf{y})$ to (13), we can construct a feasible solution to (26) with the same objective value. Given $(\pi, \mathbf{y})$, consider $\pi_i^{(k)} = \pi_i / b_i$ and $y_i^{(k)} = y_i$ for all (i, k) as a candidate solution to (26). The objective values equal each other as $\sum_{i=1}^N \sum_{k=1}^{b_i} q_i^{(k)} \pi_i^{(k)} = \sum_{i=1}^N q_i \sum_{k=1}^{b_i} \pi_i / b_i = \sum_{i=1}^N q_i \pi_i$. We next show that the candidate solution is feasible w.r.t. (26). The constraint of interest is (26b), as others are straightforward to handle. First, consider the simpler case of $\mathbb{W} = \{i\}$ being a singleton for $i \in [N]$. The case with $y_i = 0$ is trivial (as all RHS equal 0 then) and hence, suppose $y_i = 1$. We know (13b) holds, i.e., $\pi_i \leq \mathbb{E}[\min\{J, b_i\}]$. We would like to show

(26b) holds, i.e., $\sum_{k \in \mathbb{B}_i} \pi_i^{(k)} \leq \mathbb{E}[\min\{J, B_i\}] \forall \mathbb{B}_i \subseteq [b_i]$, where $B_i := |\mathbb{B}_i|$. With $\pi_i^{(k)} = \pi_i/b_i$, the LHS equals $\sum_{k \in \mathbb{B}_i} \pi_i^{(k)} = (B_i/b_i)\pi_i$. Hence, we wish to show $(b_i/b_i)\pi_i \leq \mathbb{E}[\min\{J, b_i\}]$ implies $(B_i/b_i)\pi_i \leq \mathbb{E}[\min\{J, B_i\}]$. This is true because $0 \leq B_i \leq b_i$, the LHS $(B_i/b_i)\pi_i$ is linear w.r.t. B_i , and the RHS $\mathbb{E}[\min\{J, B_i\}]$ is concave w.r.t. B_i (see Lemma 6 in §C.4).

Second, we generalize this argument to an arbitrary $\mathbb{W} \subseteq [N]$. Let $\mathbb{W} = \{i_1, \dots, i_W\}$ where $W := |\mathbb{W}|$. Suppose $y_i = 1$ for all $i \in \mathbb{W}$ (if $y_i = 0$ for some i , then that case is handled by $\mathbb{W} \setminus \{i\}$). We know (13b) holds, i.e., $\sum_{i \in \mathbb{W}} \pi_i \leq \mathbb{E}[\min\{J, \sum_{i \in \mathbb{W}} b_i\}]$ (in fact, it holds for all subsets of $\mathbb{W}$). We would like to show (26b) holds, i.e., $\sum_{i \in \mathbb{W}} \sum_{k \in \mathbb{B}_i} \pi_i^{(k)} \leq \mathbb{E}[\min\{J, \sum_{i \in \mathbb{W}} B_i\}] \forall \mathbb{B}_i \subseteq [b_i]$ where $B_i := |\mathbb{B}_i|$. With $\pi_i^{(k)} = \pi_i/b_i$, the LHS equals $\sum_{i \in \mathbb{W}} \sum_{k \in \mathbb{B}_i} \pi_i^{(k)} = \sum_{i \in \mathbb{W}} (B_i/b_i)\pi_i$. Hence, we wish to show: $\sum_{i \in \mathbb{W}} (B_i/b_i)\pi_i \leq \mathbb{E}[\min\{J, \sum_{i \in \mathbb{W}} B_i\}]$. This is true because $0 \leq B_i \leq b_i$ for all $i \in \mathbb{W}$ and $\mathbb{E}[\min\{J, b\}]$ is concave in b (see Lemma 6 in §C.4). In particular, define $h(b) := \mathbb{E}[\min\{J, b\}]$ and observe that the following stronger statement is true (with x_i mapping to B_i/b_i ¹⁷: $\sum_{i \in \mathbb{W}} x_i \pi_i \leq h(\sum_{i \in \mathbb{W}} x_i b_i)$ for all $\mathbf{x} \in [0, 1]^W$. To see this, observe $[0, 1]^W = \text{conv}(\mathbf{y}^S : \mathbf{y}^S \in \{0, 1\}^W, \mathbf{y}_i^S = \mathbb{I}\{i \in S\}, S \subseteq \mathbb{W})$, where $\text{conv}(\cdot)$ denotes the convex hull. We know that $\sum_{i \in S} \pi_i \leq h(\sum_{i \in S} b_i)$ for all $S \subseteq \mathbb{W}$ or equivalently $\sum_{i \in \mathbb{W}} y_i^S \pi_i - h(\sum_{i \in \mathbb{W}} y_i^S b_i) \leq 0$ for all $S \subseteq \mathbb{W}$. Since the LHS is convex, it follows that the inequality holds for all $\mathbf{x} \in \text{conv}(\mathbf{y}^S : \mathbf{y}^S \in \{0, 1\}^W, \mathbf{y}_i^S = \mathbb{I}\{i \in S\}, S \subseteq \mathbb{W}) = [0, 1]^W$. The proof is now complete. $\square$

B.5. Proof of Theorem 2

Theorem 2. *Tracker implements π^{**} , i.e., for all $i \in [N]$ and all $\epsilon > 0$,*

$$\liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\text{tracker}} \geq \pi_i^{**} - \epsilon.$$

Proof. Throughout the proof, we invoke Lemma 5 (§B.4) where we show that we can reformulate our problem into one in which each worker has a bandwidth of 1, i.e., $b_i = 1$ for all $i \in [N]$. Also, we focus on workers for which $y_i^{**} = 1$ and define $n := \sum_{i=1}^N y_i^{**}$ to be the corresponding number of workers.¹⁸ Given this setup, our discrete-time system can be described as follows. There are n workers who can each work on 1 job per period, and it takes 1 unit to process each job. In each period, J_t jobs arrive, where J_t is distributed i.i.d., and any job that is not processed in a period is lost. Hence, at most $\min\{J_t, n\}$ jobs are allocated in period t . Let $\sigma : J \mapsto \{0, 1\}^n$ denote any randomized allocation scheme that maps arriving jobs to workers such that

- (i) For all J , $\sum_{i=1}^n \sigma_i(J) = \min\{J, n\}$, where $\sigma_i(J) = \{0, 1\}$ is the allocation to worker i when J jobs arrive.
- (ii) $\mathbb{E}[\sigma(J)] = \pi^{**}$, where the expectation is taken over both J and the randomization in the allocation.

Since $\sum_{i \in \mathbb{W}} \pi_i^{**} \leq \mathbb{E}[\min\{J, |\mathbb{W}|\}]$ for all $\mathbb{W} \subseteq [n]$ (by definition of π^{**}), it follows from Lemma 2 of Aouad and Ma (2026) that the “Lossless Rounding” scheme they propose is an example of such an allocation scheme. The existence of this scheme will prove useful in our proof.

Our proof is based on Lyapunov stability analysis of Markov chains and we split it into 4 steps, with the outline as follows. In Step 1, we define the underlying Markov chain. In Step 2, we establish that the “drift”

¹⁷ The x and y notation used in the remainder of this proof is unrelated to the usage of (x, y) elsewhere in the paper.

¹⁸ For the remaining workers (i.e., $y_i^{**} = 0$), the Theorem 2 statement holds trivially as they never receive an allocation.

of the Markov chain is strictly negative whenever the “potential” is high (for an appropriately constructed Lyapunov function). We use this result and show in Step 3 that the expected shortfall under **Tracker** (w.r.t. π^{**}) is bounded, where the expectation is under the stationary distribution of the Markov chain. We conclude in Step 4.

Step 1. Let $x_i(t) := \sum_{t'=1}^t x_{it'}$ denote the total number of jobs allocated to worker i over periods 1 to t , where $x_{it'}$ is the number of jobs allocated to worker i during period t' (as before). Following **Tracker**, let $\delta_i(t) := t\pi_i^\epsilon - x_i(t)$ denote the degree of under-allocation/over-allocation to worker i at the end of period t , with $\pi^\epsilon := \pi^{**} - \epsilon$ for some $\epsilon > 0$.¹⁹ Observe that $\delta_i(t+1) = \delta_i(t) + \pi_i^\epsilon - x_{it}$ for all (i, t) , where π_i^ϵ is exogenous and $x_{it} \in \{0, 1\}$ (allocation to worker i during period t)²⁰ is a function of the allocation policy (**Tracker** in our case), random demand J_t , and period t state $\delta(t)$ (under **Tracker**). Thus, under **Tracker**, the stochastic process $[\delta(t)]_t$ forms a discrete-time Markov chain (with a countable state space).

Step 2. For all (i, t) , define $z_i(t) := \max\{\delta_i(t), 0\} = (t\pi_i^\epsilon - x_i(t))^+$ to be the amount of shortfall. **Tracker** is an example of a largest-shortfall first policy, i.e., in each period t , it allocates the $\min\{J_t, n\}$ jobs to workers in decreasing order of shortfall at the end of period $t-1$. Simply put, it allocates the first job to the worker with the highest shortfall, the next job to the one that has the next highest shortfall, and so on.²¹ Let $\Delta \in \{0, 1\}^n$ denote the allocations by **Tracker** in period $t+1$. Then,

$$0 \leq z_i(t+1) = (t\pi_i^\epsilon - x_i(t) + \pi_i^\epsilon - \Delta_i)^+ = (z_i(t) + \pi_i^\epsilon - \Delta_i)^+ \leq |z_i(t) + \pi_i^\epsilon - \Delta_i|,$$

where the last inequality follows from the fact that $|y| = y^+ + y^-$ and $y^- \geq 0$. Therefore, $z_i^2(t+1) \leq |z_i(t) + \pi_i^\epsilon - \Delta_i|^2 = (z_i(t) + \pi_i^\epsilon - \Delta_i)^2$. Hence, with $V(\mathbf{z}) := \sum_{i=1}^n z_i^2$ as the Lyapunov function, we get

$$\begin{aligned} V(\mathbf{z}(t+1)) - V(\mathbf{z}(t)) &= \sum_{i=1}^n (z_i^2(t+1) - z_i^2(t)) \\ &\leq \sum_{i=1}^n ((z_i(t) + \pi_i^\epsilon - \Delta_i)^2 - z_i^2(t)) \\ &= \sum_{i=1}^n (2z_i(t) + \pi_i^\epsilon - \Delta_i)(\pi_i^\epsilon - \Delta_i) \\ &= 2 \sum_{i=1}^n z_i(t)(\pi_i^\epsilon - \Delta_i) + \sum_{i=1}^n (\pi_i^\epsilon - \Delta_i)^2 \\ &\leq 2 \sum_{i=1}^n z_i(t)(\pi_i^\epsilon - \Delta_i) + \underbrace{\sum_{i=1}^n \max\{(\pi_i^\epsilon)^2, (\pi_i^\epsilon - 1)^2\}}_{=: C}, \end{aligned}$$

where the last inequality follows from taking the maximum over $\Delta_i \in \{0, 1\}$ and ignoring the fact that $\sum_{i=1}^n \Delta_i = \min\{J_{t+1}, n\}$. Now, consider the term, $\sum_{i=1}^n z_i(t)(\pi_i^\epsilon - \Delta_i)$. The allocation Δ corresponding to **Tracker** minimizes this term over all possible allocations that respect the fact that each worker can receive an allocation of 0 or 1 in each period, and the total number of allocations to all the workers is $\min\{J_{t+1}, n\}$. Since the randomized policy σ (discussed before Step 1) satisfies these constraints, it follows that

$$\sum_{i=1}^n z_i(t)(\pi_i^\epsilon - \Delta_i) \leq \sum_{i=1}^n z_i(t)(\pi_i^\epsilon - \mathbb{E}_\omega[\sigma_i(J_{t+1})]),$$

¹⁹ For technical reasons that will become clear below, we adjust π^{**} by ϵ , where $\epsilon > 0$ is arbitrarily small but strictly positive. We can do so in the definition of **Tracker** (Alg. 1) as well but it is not needed as the term corresponding to ϵ would be an additive constant in the Line 7 maximization (and hence, not affect the dynamics).

²⁰ Allocation is binary, i.e., in $\{0, 1\}$, because bandwidth is 1 (cf. Lemma 5).

²¹ For the purposes of this analysis, the behavior of the policy when all workers are over-allocated (i.e., deltas are negative) will not matter, as long as the policy allocates the job to some worker with remaining bandwidth.

where $\mathbb{E}_\omega$ denotes the expectation over the internal randomization of σ . Now, taking expectation over J_{t+1} , and using the fact that $\mathbb{E}[\sigma_i(J_{t+1})] = \pi_i^{**}$, we get

$$\mathbb{E} \left[\sum_{i=1}^n z_i(t) (\pi_i^\epsilon - \Delta_i) \right] \leq \mathbb{E} \left[\sum_{i=1}^n z_i(t) (\pi_i^\epsilon - \mathbb{E}_\omega[\sigma_i(J_{t+1})]) \right] = -\epsilon \sum_{i=1}^n z_i(t).$$

From this inequality, it follows that

$$\underbrace{\mathbb{E}[V(\mathbf{z}(t+1))] - V(\mathbf{z}(t))}_{\text{"drift"}} \leq -2\epsilon \sum_{i=1}^n z_i(t) + C < -C, \quad (27)$$

whenever

$$\sum_{i=1}^n z_i(t) = \|\mathbf{z}(t)\|_1 > \frac{C}{\epsilon}.$$

Since, $\|\mathbf{z}\|_2 \leq \|\mathbf{z}\|_1$, it follows that (27) holds if $\|\mathbf{z}(t)\|_2 > \frac{C}{\epsilon}$ or equivalently, $V(\mathbf{z}(t)) > \frac{C^2}{\epsilon^2}$. Hence, the “drift” is strictly negative whenever the “potential” $V(\mathbf{z}(t))$ is high.

Step 3. We now establish that the expected shortfall under **Tracker** is bounded. To do so, we invoke the “Combined Foster-Lyapunov stability criterion and moment bound” (Corollary 2.1.5 in Hajek (2006)) and state it below as Prop. 8.

Proposition 8. *Suppose V , f , and g are nonnegative functions such that*

$$\mathbb{E}[V(\mathbf{z}(t+1))] - V(\mathbf{z}(t)) \leq -f(\mathbf{z}(t)) + g(\mathbf{z}(t)).$$

In addition, suppose for some $\epsilon' > 0$, the set $C = \{\mathbf{z} : f(\mathbf{z}) < g(\mathbf{z}) + \epsilon'\}$ is bounded. Then, the underlying Markov chain $[\boldsymbol{\delta}(t)]_t$ is positive recurrent and $\bar{\mathbb{E}}[f(\mathbf{z})] \leq \bar{\mathbb{E}}[g(\mathbf{z})]$, where $\bar{\mathbb{E}}[\cdot]$ denotes the expectation under the stationary distribution (which exists uniquely since the Markov chain is positive recurrent).

To map Prop. 8 to our setting, take $f(\mathbf{z}) = 2\epsilon \sum_{i=1}^n z_i$ and $g(\mathbf{z}) = C$. Then, given the developments in Step 2 above, the hypothesis holds true and thus, it follows that $\sum_{i=1}^n \bar{\mathbb{E}}[z_i] \leq \frac{C}{2\epsilon}$, i.e., the expected shortfall under **Tracker** is bounded above by a constant.

Step 4. Since the underlying Markov chain is positive recurrent, it follows that the time-average equals the expectation under the stationary distribution, i.e.,

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{s=1}^t z_i(s) = \bar{\mathbb{E}}[z_i] \text{ a.s.}$$

From Step 3, we know that $\bar{\mathbb{E}}[z_i]$ is bounded above by a constant and hence, is finite. Fix a set $A \subseteq \Omega$ such that $\mathbb{P}(A) = 1$, and for all $\omega \in A$, $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{s=1}^t z_i(s, \omega) = \bar{\mathbb{E}}[z_i]$. Then, for all $\omega \in A$ and t , we have that

$$\frac{z_i(t, \omega)}{t} = \frac{1}{t} \left(\sum_{s=1}^t z_i(s, \omega) - \sum_{s=1}^{t-1} z_i(s, \omega) \right).$$

Since $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{s=1}^t z_i(s, \omega) = \lim_{t \rightarrow \infty} \frac{1}{t-1} \sum_{s=1}^t z_i(s, \omega)$ and $\lim_{t \rightarrow \infty} \frac{t-1}{t} = 1$, it follows that

$$\lim_{t \rightarrow \infty} \frac{z_i(t, \omega)}{t} = \lim_{t \rightarrow \infty} \frac{1}{t} \left(\sum_{s=1}^t z_i(s, \omega) - \sum_{s=1}^{t-1} z_i(s, \omega) \right) = 0.$$

Thus, we have that for all $\omega \in A$,

$$z_i(t, \omega) = (t\pi_i^\epsilon - x_i(t, \omega))^+ \geq t\pi_i^\epsilon - x_i(t, \omega).$$

This implies

$$\frac{x_i(t, \omega)}{t} \geq \pi_i^\epsilon - \frac{z_i(t, \omega)}{t}$$

and therefore,

$$\liminf_{t \rightarrow \infty} \frac{x_i(t, \omega)}{t} \geq \pi_i^\epsilon - \limsup_{t \rightarrow \infty} \frac{z_i(t, \omega)}{t} = \pi_i^\epsilon.$$

The proof is now complete. $\square$

Appendix C: Proofs and Further Details for §5

C.1. Multiple Binary Violations

We show here that more than one $\mathbf{y}$ variable can be fractional in the optimal solution of (13)-continuous. Consider $N = 3$ with $\mathbf{q} = (1, 0.5, 0.1)$, $\mathbf{a} = (1, 2, 1)$, and $\mathbf{b} = (1, 2, 2)$. The demand $J = 2$ or 3 w.p. $1/2$ each. The unique optimal solution to (13)-continuous is $\hat{\mathbf{y}} = (1, 0.5, 0.5)$ and $\hat{\boldsymbol{\pi}} = (1, 1, 0.5)$. Our initial conjecture was that since $q_2 > q_3$, we could perhaps transfer jobs from worker 3 to worker 2 (allowing us to make $y_2 = 1$) since doing so does not adversely impact the objective. However, such a “transfer” is not feasible. In particular, when $y_2 = 1$, we need $\pi_2 \geq a_2 = 2$. But that is not possible since $\pi_1 + \pi_2 \leq \mathbb{E}[\min\{J, b_1 + b_2\}] = 5/2$ and $\pi_1 = 1$ together imply $\pi_2 \leq 3/2$.

C.2. Proof of Proposition 5

Proposition 5. *In the κ -scaled market, there exists a constant β (independent of κ) such that*

$$\frac{\tilde{q}^{(\kappa)}}{\hat{q}^{(\kappa)}} \geq 1 - \frac{\beta}{\kappa}.$$

Proof. Observe that for an arbitrary $\kappa \in \mathbb{N}$, (13) can be generalized as follows:

$$\max_{\boldsymbol{\pi}, \mathbf{y}} \sum_{i=1}^N q_i \pi_i \tag{28a}$$

$$\text{s.t. } \sum_{i \in \mathbb{W}} \pi_i \leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i y_i \right\} \right] \quad \forall \mathbb{W} \subseteq [N] \tag{28b}$$

$$\pi_i \geq a_i y_i \quad \forall i \in [N] \tag{28c}$$

$$y_i \in \{0, 1, \dots, \kappa\} \quad \forall i \in [N]. \tag{28d}$$

Here, y_i denotes the *number* of “type” i workers that are active and π_i denotes the *total* allocation to “type” i workers. For completeness, we note that we should also have constraints of the following form:

$$\sum_{i \in \mathbb{W}} \sum_{k \in \mathbb{K}} \pi_i^{(k)} \leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i \sum_{k \in \mathbb{K}} y_i^{(k)} \right\} \right] \quad \forall \mathbb{W} \subseteq [N] \quad \forall \mathbb{K} \subseteq [\kappa].$$

Here, $\pi_i^{(k)}$ is the allocation to the k -th copy (recall there are κ copies) of type i worker and $y_i^{(k)} \in \{0, 1\}$ denotes whether the k -th copy of type i worker is active. An argument analogous to the one in the proof of Lemma 5 implies that it suffices to enforce these constraints only for $\mathbb{K} = [\kappa]$, i.e., (28b) is sufficient. Hence, we work with (28) for the rest of this proof.

We are interested in the continuous relaxation of (28) (“(28)-continuous” from now on) where we relax the constraint “ $y_i \in \{0, 1, \dots, \kappa\}$ ” to “ $y_i \in [0, \kappa]$ ” for all i , and then use our rounding heuristic to convert the fractional solution $(\hat{q}^{(\kappa)}, \hat{\boldsymbol{\pi}}^{(\kappa)}, \hat{\mathbf{y}}^{(\kappa)})$ to a discrete solution $(\tilde{q}^{(\kappa)}, \tilde{\boldsymbol{\pi}}^{(\kappa)}, \tilde{\mathbf{y}}^{(\kappa)})$. We proceed in three steps. In Step 1, we show that scaling an optimal solution to (13)-continuous by a factor of κ yields an optimal solution to (28)-continuous. In Step 2, we establish that our rounding heuristic results in a feasible solution to (28). In Step 3, we characterize the performance of our rounding heuristic as a function of κ .

Step 1. We claim that scaling an optimal solution to (13)-continuous by κ yields an optimal solution to (28)-continuous. Let $(\hat{\boldsymbol{\pi}}, \hat{\mathbf{y}})$ be any optimal solution to (13)-continuous. We want to show $(\kappa \hat{\boldsymbol{\pi}}, \kappa \hat{\mathbf{y}})$ is an

optimal solution to (28)-continuous. First, we establish the feasibility of $(\kappa\dot{\boldsymbol{\pi}}, \kappa\dot{\boldsymbol{y}})$ w.r.t. (28)-continuous. There are three sets of constraints in (28)-continuous:

$$\begin{aligned} \sum_{i \in \mathbb{W}} \pi_i &\leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i y_i \right\} \right] & \forall \mathbb{W} \subseteq [N] \\ \pi_i &\geq a_i y_i & \forall i \in [N] \\ y_i &\in [0, \kappa] & \forall i \in [N]. \end{aligned}$$

Since $(\dot{\boldsymbol{\pi}}, \dot{\boldsymbol{y}})$ is feasible w.r.t. (13)-continuous, or equivalently, these constraints for $\kappa = 1$, it is straightforward to verify $(\kappa\dot{\boldsymbol{\pi}}, \kappa\dot{\boldsymbol{y}})$ satisfies these constraints for an arbitrary κ .

Next, suppose $(\kappa\dot{\boldsymbol{\pi}}, \kappa\dot{\boldsymbol{y}})$ is not optimal for (28)-continuous, and there exists another solution $(\boldsymbol{\pi}^{(\kappa)}, \boldsymbol{y}^{(\kappa)})$ to (28)-continuous with a strictly higher objective. But then, $\frac{1}{\kappa}(\boldsymbol{\pi}^{(\kappa)}, \boldsymbol{y}^{(\kappa)})$ is feasible w.r.t. (13)-continuous and strictly better than $(\dot{\boldsymbol{\pi}}, \dot{\boldsymbol{y}})$. This contradicts the optimality of $(\dot{\boldsymbol{\pi}}, \dot{\boldsymbol{y}})$ w.r.t. (13)-continuous and completes Step 1.

Step 2. To establish that our rounding heuristic results in a feasible solution to (28), we first briefly discuss the heuristic. Recall from §5 that to convert the solution produced by (13)-continuous to binary, we round down the $\boldsymbol{y}$ variables and re-optimize over $\boldsymbol{\pi}$. For the κ -scaled market, we again round down the $\boldsymbol{y}$ variables (i.e., $\tilde{y}_i^{(\kappa)} = \lfloor \dot{y}_i^{(\kappa)} \rfloor$ for all i). However, for the purposes of this proof, instead of re-optimizing $\boldsymbol{\pi}$, we set $\tilde{\pi}_i^{(\kappa)} = \frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \dot{\pi}_i^{(\kappa)}$ for all i . We establish next that $(\tilde{\boldsymbol{\pi}}^{(\kappa)}, \tilde{\boldsymbol{y}}^{(\kappa)})$ is feasible w.r.t. (28) and thus, the objective value of $(\tilde{\boldsymbol{\pi}}^{(\kappa)}, \tilde{\boldsymbol{y}}^{(\kappa)})$ serves as a lower bound to the objective value of the re-optimized $\boldsymbol{\pi}$. (In short, we safely ignore the re-optimization step in this analysis since that only improves the objective.)

We claim that $(\tilde{\boldsymbol{\pi}}^{(\kappa)}, \tilde{\boldsymbol{y}}^{(\kappa)})$ is feasible w.r.t. (28). There are three sets of constraints in (28):

$$\begin{aligned} \sum_{i \in \mathbb{W}} \pi_i &\leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i y_i \right\} \right] & \forall \mathbb{W} \subseteq [N] \\ \pi_i &\geq a_i y_i & \forall i \in [N] \\ y_i &\in \{0, 1, \dots, \kappa\} & \forall i \in [N]. \end{aligned}$$

First, consider the last set of constraints: $y_i \in \{0, 1, \dots, \kappa\}$ for all i . Observe that $(\dot{\boldsymbol{\pi}}^{(\kappa)}, \dot{\boldsymbol{y}}^{(\kappa)})$ is feasible w.r.t. (28)-continuous and hence, $\dot{y}_i^{(\kappa)} \in [0, \kappa]$ for all i . It follows that $\tilde{y}_i^{(\kappa)} = \lfloor \dot{y}_i^{(\kappa)} \rfloor \in \{0, 1, \dots, \kappa\}$ for all i . Now, consider the second set of constraints: $\pi_i \geq a_i y_i$ for all i . It suffices to show that $\frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \dot{\pi}_i^{(\kappa)} \geq a_i \lfloor \dot{y}_i^{(\kappa)} \rfloor$ for all i . Via basic rearranging of terms, this is equivalent to $\dot{\pi}_i^{(\kappa)} \geq a_i \dot{y}_i^{(\kappa)}$ for all i , which is true because $(\dot{\boldsymbol{\pi}}^{(\kappa)}, \dot{\boldsymbol{y}}^{(\kappa)})$ is feasible w.r.t. (28)-continuous. Finally, consider the first set of constraints: $\sum_{i \in \mathbb{W}} \pi_i \leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i y_i \right\} \right]$ for all $\mathbb{W} \subseteq [N]$. It suffices to show that

$$\sum_{i \in \mathbb{W}} \frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \dot{\pi}_i^{(\kappa)} \leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i \lfloor \dot{y}_i^{(\kappa)} \rfloor \right\} \right] \quad \forall \mathbb{W} \subseteq [N]. \quad (29a)$$

Since $(\dot{\boldsymbol{\pi}}^{(\kappa)}, \dot{\boldsymbol{y}}^{(\kappa)})$ is feasible w.r.t. (28)-continuous, we know that

$$\sum_{i \in \mathbb{W}} \dot{\pi}_i^{(\kappa)} \leq \mathbb{E} \left[\min \left\{ \kappa J, \sum_{i \in \mathbb{W}} b_i \dot{y}_i^{(\kappa)} \right\} \right] \quad \forall \mathbb{W} \subseteq [N]. \quad (29b)$$

We have already established (29b) implies (29a) in the proof of Lemma 5 (§B.4). This completes Step 2.

Step 3. Having established feasibility, we now characterize the performance of $(\tilde{\boldsymbol{\pi}}^{(\kappa)}, \tilde{\boldsymbol{y}}^{(\kappa)})$. Observe that $(\dot{\boldsymbol{\pi}}^{(\kappa)}, \dot{\boldsymbol{y}}^{(\kappa)})$ provides an upper bound to the optimal objective of (28) (as it corresponds to (28)-continuous).

Hence, it suffices to understand how much we lose when going from $\dot{\pi}^{(\kappa)}$ to $\tilde{\pi}^{(\kappa)} = \frac{\lfloor \dot{y}^{(\kappa)} \rfloor}{\dot{y}^{(\kappa)}} \dot{\pi}^{(\kappa)}$. Observe that the difference $\dot{\pi}_i^{(\kappa)} - \frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \dot{\pi}_i^{(\kappa)}$ can be upper bounded as follows for all i :

$$\begin{aligned}
 \dot{\pi}_i^{(\kappa)} - \frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \dot{\pi}_i^{(\kappa)} &= \dot{\pi}_i^{(\kappa)} \left(1 - \frac{\lfloor \dot{y}_i^{(\kappa)} \rfloor}{\dot{y}_i^{(\kappa)}} \right) \\
 &= \kappa \dot{\pi}_i \left(1 - \frac{\lfloor \kappa \dot{y}_i \rfloor}{\kappa \dot{y}_i} \right) && \text{[due to Step 1]} \\
 &= \kappa \dot{\pi}_i \times \frac{\kappa \dot{y}_i - \lfloor \kappa \dot{y}_i \rfloor}{\kappa \dot{y}_i} \\
 &\leq \kappa \dot{\pi}_i \times \frac{1}{\kappa \dot{y}_i} && [\kappa \dot{y}_i - \lfloor \kappa \dot{y}_i \rfloor \leq 1] \\
 &= \frac{\dot{\pi}_i}{\dot{y}_i}.
 \end{aligned}$$

Hence, for each i , our rounding heuristic loses at most $\dot{\pi}_i/\dot{y}_i$ allocations. Thus, when aggregated over i , it loses at most $\sum_{i \in [N]: \dot{y}_i > 0} q_i \dot{\pi}_i/\dot{y}_i$ in objective value, i.e., $\tilde{q}^{(\kappa)} \geq \dot{q}^{(\kappa)} - \sum_{i \in [N]: \dot{y}_i > 0} q_i \dot{\pi}_i/\dot{y}_i$. Furthermore, due to Step 1, the optimal objective value of (28)-continuous obeys $\dot{q}^{(\kappa)} = \kappa \sum_{i \in [N]: \dot{y}_i > 0} q_i \dot{\pi}_i$. Thus, $\tilde{q}^{(\kappa)}/\dot{q}^{(\kappa)} \geq 1 - \beta/\kappa$, where $\beta := \left(\sum_{i \in [N]: \dot{y}_i > 0} q_i \dot{\pi}_i/\dot{y}_i \right) / \left(\sum_{i \in [N]: \dot{y}_i > 0} q_i \dot{\pi}_i \right)$. Finally, since $(\dot{\pi}, \dot{y})$ is an optimal solution to (13), it is independent of κ and hence, so is β . The proof is now complete. $\square$

C.3. Connection to the Knapsack Problem

In this section, we analyze our problem (recall (7) from §2 and the corresponding tight relaxation (13) from §4.2) under the assumption that $a_i = b_i$ for all i . Consider an arbitrary worker i . Since $a_i = b_i$, if worker i is active, then $\pi_i = b_i$. Thus, the only relevant decision here is y_i , i.e., whether worker i is active or not. In each period, the total allocation across workers equals $\sum_i b_i y_i$, which must not exceed any demand realization: $\sum_i b_i y_i \leq J$ w.p. 1. Thus, it follows that $\sum_i b_i y_i \leq J_{\min}$, where $J_{\min}$ denotes the minimum possible value of demand, and the optimization problem reduces to the knapsack problem:

$$\max_{\mathbf{y} \in \{0,1\}^N} \sum_{i=1}^N q_i b_i y_i \quad (30a)$$

$$\text{s.t. } \sum_{i=1}^N b_i y_i \leq J_{\min}. \quad (30b)$$

Each worker maps to an “item”. The value of each item i is $q_i b_i$ and its weight is b_i . The knapsack’s weight capacity is $J_{\min}$.

It is well-known that the optimal solution to the continuous relaxation, where binary constraints “ $y_i \in \{0,1\}$ ” are replaced by “ $y_i \in [0,1]$ ” for all i (“(30)-continuous” from now on), is to prioritize using the value-to-weight ratio, which equals $q_i b_i/b_i = q_i$ for all i in our setting. Thus, the optimal solution to (30)-continuous is the same as **Greedy** where we sort the items in decreasing order of quality (i.e., $q_1 \geq \dots \geq q_N$) and keep adding them to the knapsack until we hit capacity. Consequently, at most one $\mathbf{y}$ variable will be non-binary. Our rounding heuristic will round down the non-binary $\mathbf{y}$ variable and we will lose at most $q_{\max} b_{\max}$ in objective value (compared to the optimal objective value of (30)-continuous), where $q_{\max} := \max_i q_i$ and $b_{\max} := \max_i b_i$. We summarize this discussion in the following proposition, with $\dot{q}$ denoting the optimal objective value of (30)-continuous and $\tilde{q}$ denoting the value obtained after our rounding heuristic (as in §5).

Proposition 9. *If $\mathbf{a} = \mathbf{b}$, then $\tilde{q} \geq \dot{q} - q_{\max} b_{\max}$.*

C.4. Efficient Computation and Concavity of $\mathbb{E}[\min\{J, B\}]$

In general, we can compute $\mathbb{E}[\min\{J, B\}]$ via a one-dimensional integral over the random demand J . As a special case, if J follows a normal distribution with mean μ and standard deviation σ , then

$$\mathbb{E}[\min\{J, B\}] = B + (\mu - B)\Phi(z) - \sigma\phi(z),$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of a standard normal distribution, and $z = \frac{B-\mu}{\sigma}$. Furthermore, as we establish next, $\mathbb{E}[\min\{J, B\}]$ is concave in B .

Lemma 6. $\mathbb{E}[\min\{J, B\}]$ is concave in B .

Proof. For a fixed J , $\min\{J, B\}$ is concave in B (trivial). Any non-negative weighted sum/integral preserves concavity (see p. 79 of [Boyd and Vandenberghe \(2004\)](#)). Hence, $\mathbb{E}[\min\{J, B\}]$ is concave in B . $\square$

C.5. Proof of Lemma 4

Lemma 4. $g(\mathbb{W} \mid \boldsymbol{\pi}, \mathbf{y})$ is a submodular set function with respect to $\mathbb{W}$.

Proof. Recall that

$$g(\mathbb{W} \mid \boldsymbol{\pi}, \mathbf{y}) = \mathbb{E} \left[\underbrace{\min \left\{ J, \sum_{i \in \mathbb{W}} b_i y_i \right\}}_{=:\hat{g}(\mathbb{W} \mid \mathbf{y})} \right] - \sum_{i \in \mathbb{W}} \pi_i.$$

Consider $\mathbb{W}_1, \mathbb{W}_2 \subseteq [N]$ with $\mathbb{W}_1 \subseteq \mathbb{W}_2$ and an arbitrary $j \in [N] \setminus \mathbb{W}_2$. It suffices to show $g(\mathbb{W}_1 \cup \{j\} \mid \boldsymbol{\pi}, \mathbf{y}) - g(\mathbb{W}_1 \mid \boldsymbol{\pi}, \mathbf{y}) \geq g(\mathbb{W}_2 \cup \{j\} \mid \boldsymbol{\pi}, \mathbf{y}) - g(\mathbb{W}_2 \mid \boldsymbol{\pi}, \mathbf{y})$. Observe that the LHS equals $\hat{g}(\mathbb{W}_1 \cup \{j\} \mid \mathbf{y}) - \hat{g}(\mathbb{W}_1 \mid \mathbf{y}) - \pi_j$ and the RHS equals $\hat{g}(\mathbb{W}_2 \cup \{j\} \mid \mathbf{y}) - \hat{g}(\mathbb{W}_2 \mid \mathbf{y}) - \pi_j$. Hence, it suffices to show $\hat{g}(\mathbb{W}_1 \cup \{j\} \mid \mathbf{y}) - \hat{g}(\mathbb{W}_1 \mid \mathbf{y}) \geq \hat{g}(\mathbb{W}_2 \cup \{j\} \mid \mathbf{y}) - \hat{g}(\mathbb{W}_2 \mid \mathbf{y})$, i.e., the function $\hat{g}(\mathbb{W} \mid \mathbf{y})$ is submodular w.r.t. $\mathbb{W}$ (for a given $\mathbf{y} \geq 0$). Since $\hat{g}(\mathbb{W} \mid \mathbf{y}) = \mathbb{E}[\min\{J, \sum_{i \in \mathbb{W}} b_i y_i\}]$, its submodularity will be implied by the concavity of $\mathbb{E}[\min\{J, B\}]$ w.r.t. B , which is true (see Lemma 6 in §C.4). The proof is now complete. $\square$

Appendix D: Further Details on the Numerical Study

D.1. Computing Infrastructure

We code in MATLAB ([The MathWorks Inc. 2025](#)) and parallelize our computations over the 100 seeds on a Linux cluster equipped with a mix of CPU cores, most prominent being AMD EPYC (clock speeds: 2.4-2.95 GHz). Unless stated otherwise, each seed is run on a single core with at most 16 GB memory.

For our NLP approach (§5.1), we use the IPOPT solver ([Wächter and Biegler 2006](#)) along with CasADi²² ([Andersson et al. 2019](#)) to solve the convex program in Line 3 of Alg. 2 and the SFO toolbox ([Krause 2010](#)) to solve the submodular function minimization in Line 4 of Alg. 2. For consistency, we use the same tools (IPOPT, CasADi, SFO) to solve the rounding LP (§5). Finally, for our SAA approach (§5.2), we use Gurobi ([Gurobi Optimization, LLC 2024](#)) to solve the linear program (17) and we use the same software (Gurobi) to solve the corresponding rounding LP (§5).

²² CasADi builds the convex program symbolically and automatically computes exact gradients and Hessians, which IPOPT requires for its Newton-type interior-point iterations.

D.2. Details on Simulating Various Policies

Simulating **Tracker** is straightforward (Alg. 1), with the input $(\pi, \mathbf{y})$ computed via the NLP (§5.1) or the SAA (§5.2) approach. Unless stated otherwise, both approaches employ the rounding heuristic described in §5. As we demonstrate in §6.2, this heuristic produces solutions that are near-optimal. We compare the runtimes of the two approaches in §6.3.

For **Greedy**, we compute the equilibrium using Alg. 5 in §A.2 (it took less than a second). For **FCFS**, we compute an equilibrium via brute force (Alg. 6). It initially sets some workers to be active (Line 1)²³ and simulates **FCFS** allocations (Line 2). Lines 6 to 20 loop over each worker, computes her first-best response, and updates the allocations accordingly. If a worker is active but is not meeting her min. requirement²⁴, she becomes inactive (Lines 7 to 10). Conversely, if a worker is inactive but will meet her min. requirement by being active, she becomes active (Lines 11 to 19). The procedure converges when no worker changes her decision. Due to the high computational burden, we use a smaller simulation horizon ($T = 100$ rather than $T = 5,000$) when computing allocations in Lines 2, 9, and 13.²⁵ In the §6.2 numerics, we found the procedure to converge in 4 to 6 iterations of the Line 4 while loop, with each iteration requiring approx. 1 to 5 minutes. Finally, note that there can possibly be multiple **FCFS** equilibria, but we only compute one per seed. However, we use 100 seeds and pick $\mathbf{y}^{\text{initial}}$ randomly.

D.3. Sensitivity Analysis

We perform sensitivity analysis around the mean and standard deviation of demand (Fig. 7 and 8), supply-side quality, bandwidth, and min. requirements (Fig. 9 to 11), and the number of workers (Fig. 12). Remarkably, we find the directional findings reported in the main body (**FCFS** < **Greedy** < **Tracker**) to be robust all across. The average gains provided by **Tracker** vary between approx. 10-45% and 2-15% with respect to **FCFS** and **Greedy**, respectively (average over 100 seeds).

²³ We picked around half of the workers (at random) to be active but one could provide $\mathbf{y}^{\text{initial}} = \mathbf{0}$ too.

²⁴ To accommodate simulation noise, we allow an ϵ -tolerance when checking min. requirements (Lines 7 and 14 in Alg. 6), setting $\epsilon = 0.02$. This value is motivated by the NLP results (§6.2), where **Tracker** occasionally under-allocated some workers by roughly 2% despite the NLP solution being feasible, indicating that discrepancies of this magnitude can arise from Monte Carlo noise.

²⁵ To mitigate the resulting simulation noise, we apply a smoothing step in which allocations are averaged across workers with identical bandwidths; under **FCFS**, any two active workers with the same bandwidth receive the same long-run average allocation. In addition, after convergence, we run one final **FCFS** simulation using the equilibrium active set $\mathbf{y}$ but with $T = 5,000$.

Algorithm 6 FCFS-Equilibrium($\mathbf{y}^{\text{initial}}, \epsilon$)

```

1:  $\mathbf{y} = \mathbf{y}^{\text{initial}}$  % initialize which workers are active
2:  $\mathbf{x} = \text{FCFS}(\mathbf{y})$  % compute FCFS allocations as a function of  $\mathbf{y}$  (Alg. 3 in §A.1)
3: converged = FALSE % indicator for convergence
4: while not converged do
5:   converged = TRUE % set indicator to 1 to begin with
6:   for  $i = 1$  to  $N$  do
7:     if  $y_i > 0$  and  $\frac{\frac{1}{T} \sum_{t=1}^T x_{it}}{a_i} < 1 - \epsilon$  then
8:        $y_i = 0$  % set worker to be inactive since she is not meeting her min. requirement
9:        $\mathbf{x} = \text{FCFS}(\mathbf{y})$  % compute FCFS allocations using the updated  $\mathbf{y}$ 
10:      converged = FALSE % worker changed her decision and hence, we have not converged
11:    else if  $y_i = 0$  then
12:       $\mathbf{y}' = (1, \mathbf{y}_{-i})$  % suppose worker  $i$  is active and other workers do not change their decision
13:       $\mathbf{x}' = \text{FCFS}(\mathbf{y}')$  % compute FCFS allocations under  $\mathbf{y}'$ 
14:      if  $\frac{\frac{1}{T} \sum_{t=1}^T x'_{it}}{a_i} \geq 1 - \epsilon$  then
15:         $y_i = 1$  % set worker to be active since she will meet her min. requirement if active
16:         $\mathbf{x} = \mathbf{x}'$  % update allocations accordingly
17:      converged = FALSE % we have not converged
18:    end if
19:  end if
20: end for
21: end while
22: return  $(\mathbf{x}, \mathbf{y})$ 

```

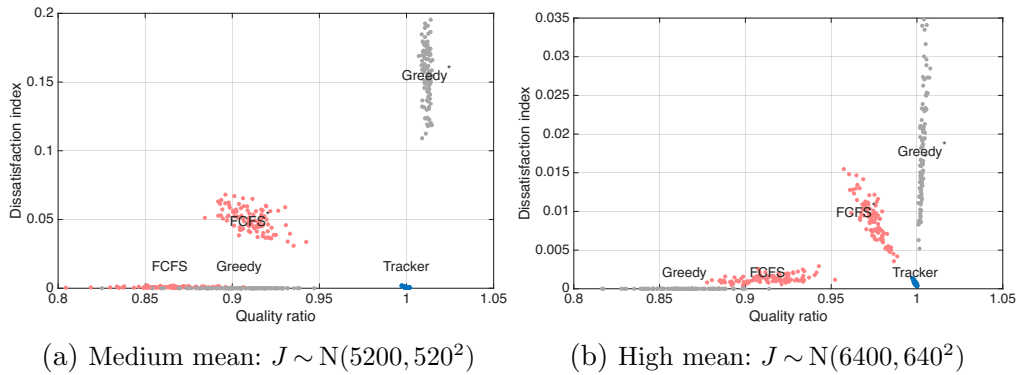


Figure 7 Sensitivity analysis w.r.t. demand mean μ . We hold the coefficient of variation σ/μ at 0.1, as in the base case. Under high mean (subplot (b)), the mean demand equals the expected total bandwidth.

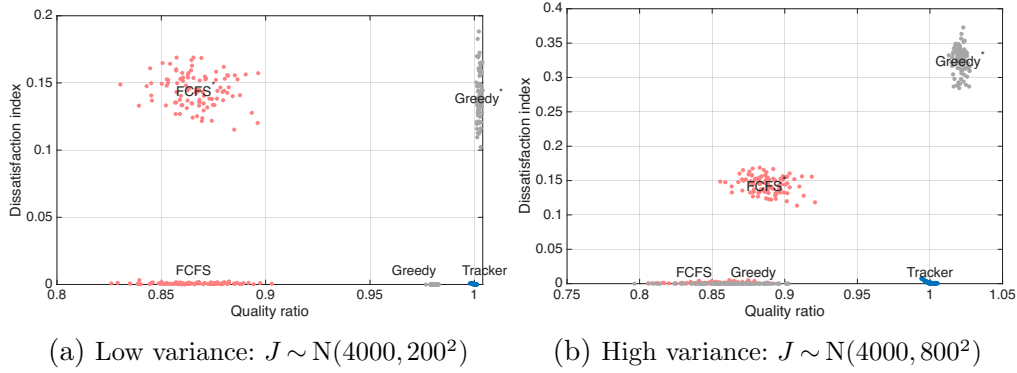


Figure 8 Sensitivity analysis w.r.t. demand standard deviation σ . Recall the base case is $J \sim N(4000, 400^2)$.

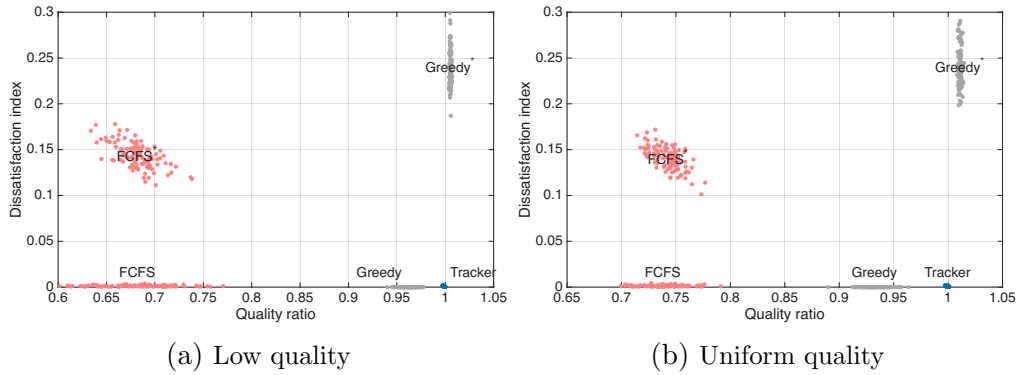


Figure 9 Sensitivity analysis w.r.t. quality distribution. Low quality flips the q distribution s.t. $q_i \sim U(0, 0.1)$, $U(0.1, 0.2)$, $U(0.2, 1)$ w.p. 0.5, 0.3, 0.2, respectively, for all i . Uniform quality refers to a $U(0, 1)$ quality distribution.

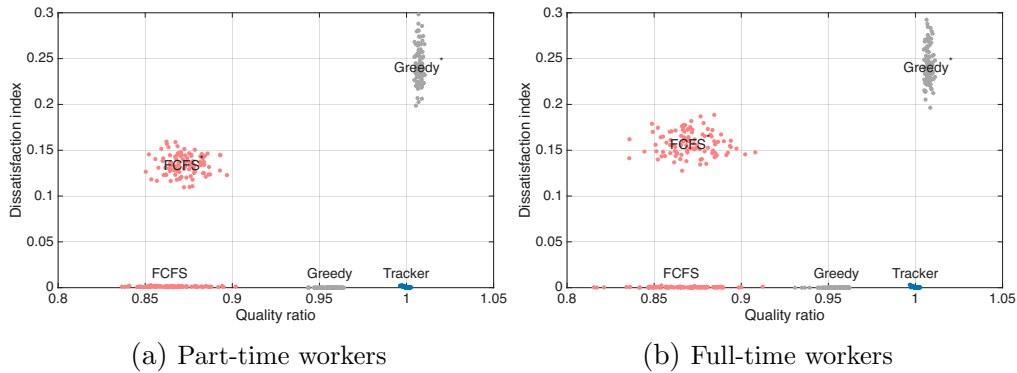


Figure 10 Sensitivity analysis w.r.t. bandwidth distribution. Part-time refers to a bandwidth of $\{4, 8, 12, 16, 20, 24, 28\}$ w.p. $\{1/4, 1/4, 1/4, 1/16, 1/16, 1/16, 1/16\}$ (more part-time workers). Full-time refers to a bandwidth of $\{4, 8, 12, 16, 20, 24, 28\}$ w.p. $\{1/16, 1/16, 1/16, 1/16, 1/4, 1/4, 1/4\}$ (more full-time workers). When we change the bandwidth distribution, we ensure the expected total bandwidth matches that of the base case (6400). This is done by scaling N (557 and 312 for part-time and full-time workers, respectively).

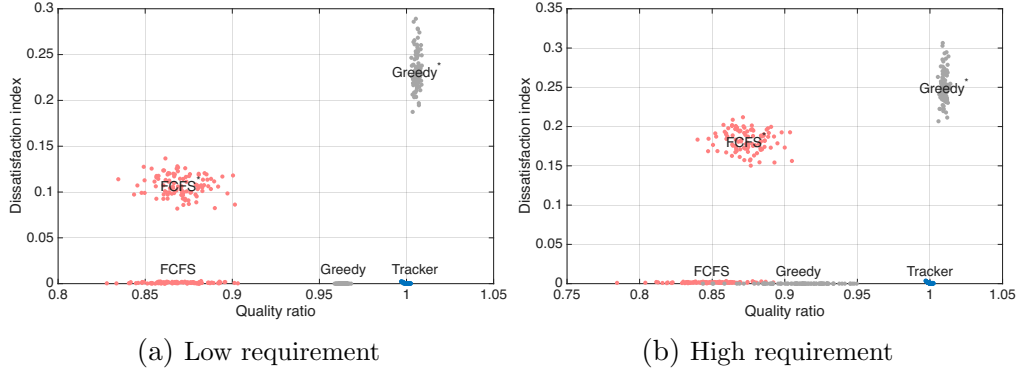


Figure 11 Sensitivity analysis w.r.t. min. requirement distribution. In the base case, requirement is $\{0.5b_i, 0.75b_i, b_i\}$ w.p. $\{1/3, 1/3, 1/3\}$ for all i . In the low case, the probabilities change to $\{0.5, 0.3, 0.2\}$ so that we have more workers with low requirement. In the high case, the probabilities change to $\{0.2, 0.3, 0.5\}$.

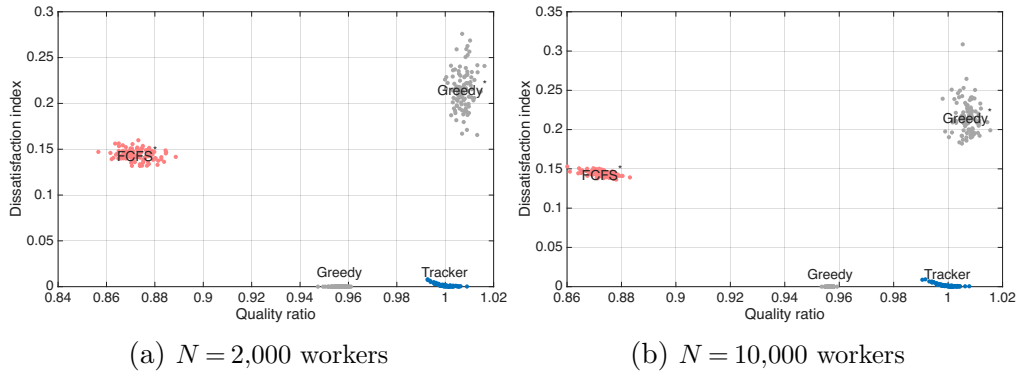


Figure 12 Sensitivity analysis w.r.t. number of workers N . In the base case, $N = 400$ with demand parameters $(\mu, \sigma) = (4000, 400)$. As we increase N , we also increase (μ, σ) appropriately. In particular, for $N = 2,000$, we use $(\mu, \sigma) = (20000, 2000)$ and for $N = 10,000$, we use $(\mu, \sigma) = (100000, 10000)$. For such large N , we no longer compute the FCFS equilibrium FCFS (but only FCFS*) as it is computationally too expensive. As seen in the earlier results, FCFS* often does at least as well as FCFS in terms of quality. For $N = 2,000$ and $10,000$, solving the NLP took around 2 minutes and an hour on average (over 100 seeds), respectively. Given higher N , we reduced T from 5,000 to 1,000 to allow us to simulate each policy in manageable compute time. This reduction in T explains the higher variance across the 100 seeds in these results (compared to the results reported above), especially for Tracker.

D.4. Details on the Computational Performance of SAA

As mentioned in §D.1, we use Gurobi to solve the SAA linear program (17). Three practical questions arise in using the SAA approach. First, what value of S is large enough? Second, is that value of S robust across parameter regimes? Third, how computationally tractable is the resulting linear program? We answer these questions next: $S = 100$ performs well (§D.4.1), is generally robust except under extremely high demand variance (§D.4.2), and is computationally very manageable (already discussed in §6.3).

D.4.1. What Value of S is Large Enough? To determine a suitable value of S , we fix all other model primitives at their base case values (from §6) and experiment with $S \in \{10, 20, \dots, 100, 200, \dots, 1000\}$. Larger values of S improve the accuracy of the SAA approximation but also increase the size (and cost) of the linear program. Our aim is therefore to find the smallest S that yields sufficiently accurate performance.

A key concern with finite S is feasibility: tracking the SAA solution may produce active workers whose avg. allocations fall short of their requirements, yielding a positive dissatisfaction index (as with FCFS* and Greedy* in Fig. 3). Fig. 13a reports the dissatisfaction index for SAA* as a function of S .²⁶ The black curve shows the *mean avg.* dissatisfaction index, computed by (a) averaging dissatisfaction across workers within a seed, and then (b) averaging across the 100 seeds. The red curve shows the corresponding *mean max.* dissatisfaction index, where we first take the max. across workers for each seed before averaging across seeds.

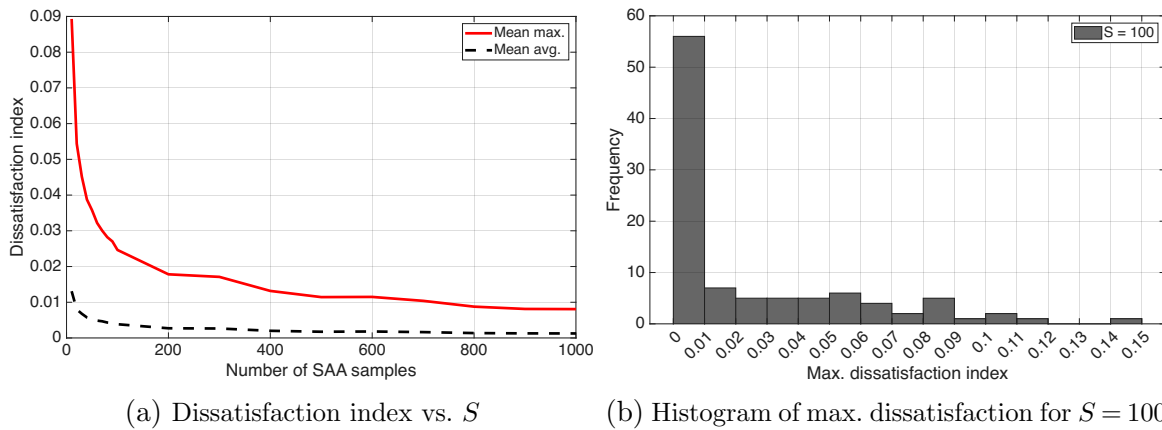


Figure 13 Dissatisfaction index under SAA*.

Avg. dissatisfaction remains relatively small (compare to approx. 0.15 for FCFS* and 0.25 for Greedy* in Fig. 3), and there is a sharp initial decline with an “elbow” around $S = 100$. Nevertheless, the max. dissatisfaction can be non-trivial for some workers. Fig. 13b shows the distribution of max. dissatisfaction at $S = 100$. Because dissatisfied workers may churn, such deviations can affect the quality ratio. To study this, we allow the dissatisfied workers to churn (SAA* vs. SAA). We classify a worker as dissatisfied if her ratio of avg. per-period allocation to min. requirement is below $1 - \epsilon$ and let such workers churn.²⁷ We re-simulate Tracker to track the original targets for the remaining workers. This procedure is conservative, as any re-optimization of targets would only improve performance.

The resulting dissatisfaction under SAA is shown in Fig. 14. All dissatisfaction values are essentially zero (note the scale is an order of magnitude smaller than in Fig. 13), supporting the interpretation of these outcomes as an equilibrium. Fig. 15a reports the corresponding avg. quality ratio across seeds. For $S \in \{10, 20, \dots, 50\}$, churn lowers the avg. quality ratio below 0.98. A clear elbow again occurs around $S = 100$, where the avg. quality ratio is 0.994. Fig. 15b shows the distribution underlying this avg. value of 0.994: 85 out of 100 seeds exceed 0.99. Thus, $S = 100$ appears to be large enough.

²⁶ As with FCFS* and Greedy*, we use SAA* to denote that the dissatisfied workers remain active.

²⁷ We use $\epsilon = 0.02$ (same rationale as in Footnote 24).

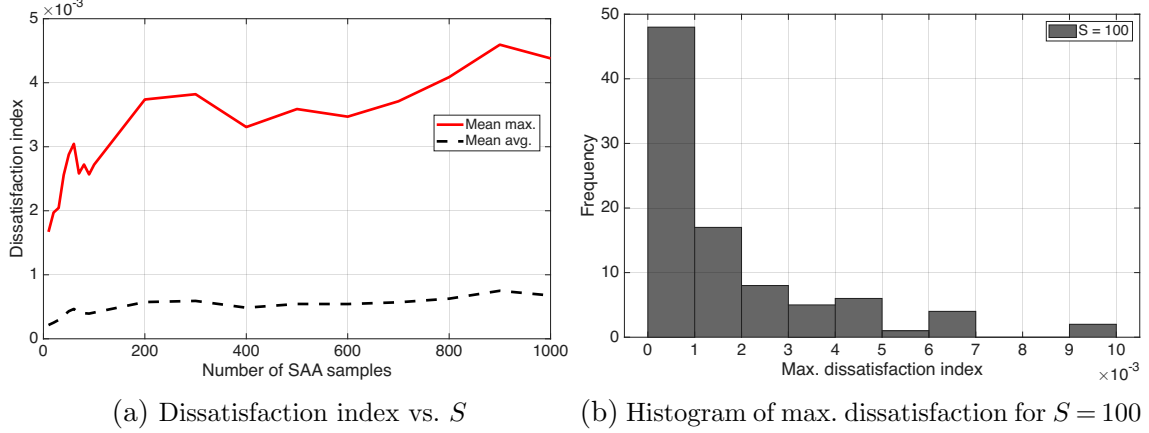


Figure 14 Dissatisfaction index under SAA.

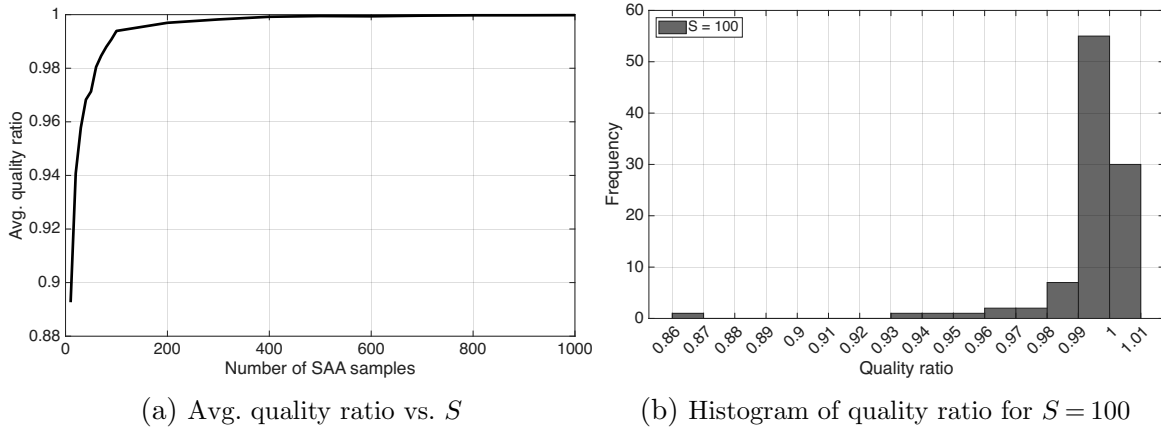


Figure 15 Quality ratio under SAA.

D.4.2. Is $S = 100$ a Robust Choice? To assess robustness, we compute avg. quality ratios across a range of parameter regimes under SAA (with dissatisfied workers churning as above). We evaluate performance over the 12 regimes considered in §D.3. The avg. quality ratio exceeds 0.985 in all but one regime; the lone exception (0.947) corresponds to the high-variance demand scenario of Fig. 8b. Hence, $S = 100$ appears to be a robust choice across a broad range of settings, except in cases with high demand variability.

Appendix E: Probabilistic Minimum Allocation Guarantees

As mentioned in §1, “unpredictable” and “low” income are key driving forces behind workers churn. Though our framework so far ensures the minimum allocations are guaranteed over the long-run average (and hence, addresses “low” income), these minimum allocations might not be realized in each period. In fact, in the §6 results (corresponding to seed 1 for example), $\approx 63\%$ of the periods under **Tracker** do not achieve minimum allocations. As alluded to in §2 (around (1)), we now extend our framework to guarantee minimum allocations in each period (with high probability). We begin by formulating the corresponding optimization problem in §E.1, discuss our solution methodology in §E.2, and present numerical results in §E.3. Towards the end of this section, in Remark 4, we also briefly discuss how one can extend our framework to encode minimum allocation guarantees at a group level.

E.1. The Optimization Problem

The new optimization problem (31) is the same as (7) except for one change: the minimum allocation constraint changes from “on average” (i.e., (7e)) to “w.p. at least γ ” (i.e., (31e)). The parameter γ is an input to the model that the user controls. Of course, a higher γ would provide more certainty to the workers (in terms of achieving minimum allocations in each period), but would hurt the average quality of allocations (i.e., the objective). Note that (31e) does not necessarily imply (7e) and we do not enforce (7e) in (31).

$$\max_{\mathbf{p}} \mathbb{E} \left[\sum_{i=1}^N q_i \bar{x}_i^{\mathbf{p}} \right] \quad (31a)$$

$$\text{s.t. } x_{it}^{\mathbf{p}} = \sum_{j=1}^{J_t} z_{ijt}^{\mathbf{p}} \quad \forall i \in [N] \quad \forall t \quad (31b)$$

$$\sum_{i=1}^N z_{ijt}^{\mathbf{p}} \leq 1 \quad \forall j \in [J_t] \quad \forall t \quad (31c)$$

$$\bar{x}_i^{\mathbf{p}} = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{it}^{\mathbf{p}} \quad \forall i \in [N] \quad (31d)$$

$$\mathbb{P} \{ x_{it}^{\mathbf{p}} \geq a_i y_i^{\mathbf{p}} \quad \forall i \} \geq \gamma \quad \forall t \quad (31e)$$

$$x_{it}^{\mathbf{p}} \leq b_i y_i^{\mathbf{p}} \text{ w.p. } 1 \quad \forall i \in [N] \quad \forall t \quad (31f)$$

$$y_i^{\mathbf{p}}, z_{ijt}^{\mathbf{p}} \in \{0, 1\} \quad \forall i \in [N] \quad \forall j \in [J_t] \quad \forall t. \quad (31g)$$

Let $q^*(\gamma)$ be the optimal objective value of (31).

Remark 3. As noted in Remark 1, the baseline model and all associated theoretical guarantees continue to hold under any demand process $[J_t]_t$ that converges to a stationary law, i.e., $J_t \sim F_t$ with $F_t \rightarrow F$ as $t \rightarrow \infty$. However, this level of generality does not extend to the w.h.p. formulation in (31). For the per-period guarantee in (31e), we require the demand process to be *stationary* in the sense that its marginal distribution is exactly F for all periods. Under this stationary assumption (which is weaker than i.i.d.), all of our developments for the w.h.p. extension continue to go through.

E.2. Our Solution Methodology

We start by focusing on the newly added constraint (31e). Observe that a necessary condition for satisfying (31e) is that, with probability at least γ , the demand $J \geq \sum_i a_i y_i$, or equivalently,

$$\sum_{i=1}^N a_i y_i \leq F^{-1}(1 - \gamma), \quad (32)$$

where (recall from §2) $F(\cdot)$ denotes the cdf of the random demand J (and $F^{-1}(\cdot)$ is the inverse cdf). In fact, we claim that (32) is sufficient as well if **Tracker** is modified to first “track” the minimum allocations in each period, i.e., ensure the first $\sum_i a_i y_i$ jobs in each period are allocated such that each active worker i gets exactly a_i jobs (formalized in Alg. 7 below). Of course, if $J_t < \sum_i a_i y_i$, then we won’t be able to achieve this in period t but this would only happen w.p. at most $1 - \gamma$ (cf. (32)). If the optimal solution satisfies (32) as a strict inequality, γ can be increased without changing the optimal solution to (31) and a higher γ is desirable (all else being equal). Given this, moving forward, we will focus on such “efficient” γ values for which, at optimality, (32) holds with equality. Let Γ be the set of efficient γ .

Following the discussion above, the modified **Tracker** (which we label **Tracker2** in Alg. 7) uses the first $J' := \min\{J, \sum_i a_i y_i\} = \min\{J, F^{-1}(1 - \gamma)\}$ jobs in each period to satisfy (31e). Next, we shift our focus to

the remaining $J'' := (J - \sum_i a_i y_i)^+ = (J - F^{-1}(1 - \gamma))^+$ jobs, where observe that $J = J' + J''$ by construction. We claim that our technical machinery from previous sections can be recycled to optimally allocate the J'' jobs. In order to do so, we split the average total allocation π_i to worker i into the allocation π'_i from the J' jobs and the allocation π''_i from the J'' jobs, and we define the residual bandwidth $b''_i := (b_i - a_i)$ for all i , as the initial a_i bandwidth is used up by J' . With this notation, from the discussion leading (13), it follows that we need the following constraints in order to get a tight deterministic relaxation:

$$\sum_{i \in \mathbb{W}} \pi''_i \leq \mathbb{E} \left[\min \left\{ J'', \sum_{i \in \mathbb{W}} b''_i y_i \right\} \right] \quad \forall \mathbb{W} \subseteq [N]. \quad (33)$$

As in (13), these constraints ((32) and (33) along with $y_i \in \{0, 1\}$ or $y_i \in [0, 1]$ for all i) lead to a good deterministic relaxation of (31). However, analyzing the objective $\sum_i q_i(\pi'_i + \pi''_i)$ (average allocation quality) is challenging. This is because we do not have an expression for π'_i . We know that $0 \leq \pi'_i \leq a_i y_i$ and $\pi'_i \geq a_i y_i \mathbb{P}\{x_{it} \geq a_i y_i\} \geq a_i y_i \gamma$. Hence, $a_i y_i \gamma \leq \pi'_i \leq a_i y_i$, meaning $\pi'_i \rightarrow a_i y_i$ as $\gamma \rightarrow 1$, and we approximate the objective by $\sum_i q_i(a_i y_i + \pi''_i)$. (Note that the proxy is an upper bound to the true objective. In Theorem 3 below, we use this fact to construct an error bound as a result of this approximation.) We are now in a position to present our deterministic static relaxation for (31), which combines this objective with the constraints discussed above ((32) and (33)):

$$\max_{\pi'', \mathbf{y}} \sum_{i=1}^N q_i(a_i y_i + \pi''_i) \quad (34a)$$

$$\text{s.t.} \quad \sum_{i \in \mathbb{W}} \pi''_i \leq \mathbb{E} \left[\min \left\{ J'', \sum_{i \in \mathbb{W}} b''_i y_i \right\} \right] \quad \forall \mathbb{W} \subseteq [N] \quad (34b)$$

$$\sum_{i=1}^N a_i y_i \leq F^{-1}(1 - \gamma) \quad (34c)$$

$$y_i \in \{0, 1\} \quad \forall i \in [N]. \quad (34d)$$

(34) is analogous to (13), and we use $q^{**}(\gamma)$ and $(\mathbf{y}^{**}(\gamma), \boldsymbol{\pi}''_{**}(\gamma))$ to denote its optimal objective and solution. Note that we can recycle the computational techniques from §5 to solve (34).

We “track” this solution using the slightly modified **Tracker** mentioned above, which we denote by **Tracker2** (pseudocode in Alg. 7). In each period, we first ensure the initial $\sum_i a_i y_i^{**}(\gamma)$ jobs are allocated such that each active worker i gets exactly a_i jobs (Lines 6 to 10). Then, if $J > \sum_i a_i y_i^{**}(\gamma)$, we track $\boldsymbol{\pi}''_{**}(\gamma)$ using the corresponding long-run average (Lines 11 to 16), similar to how we track $\boldsymbol{\pi}^{**}$ in Alg. 1.

Having presented our methodology, we now characterize its theoretical performance. Given the above discussions and Theorem 2, it directly follows that under **Tracker2**, (i) *every* active worker receives her minimum requirement w.p. at least γ in *each* period and (ii) $\boldsymbol{\pi}''$ is “tracked”, i.e., for all $\gamma \in \Gamma$ and $\epsilon > 0$,

$$\liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x''_{it}(\gamma) \geq \pi''_{i,**}(\gamma) - \epsilon.$$

The only approximation **Tracker2** makes is in the objective function in that it uses $\sum_i q_i(a_i y_i + \pi''_i)$ as a proxy for $\sum_i q_i(\pi'_i + \pi''_i)$, as discussed above. We now provide an explicit error bound due to this approximation. Denoting by $q^{\text{tracker2}}(\gamma)$ the objective value achieved by **Tracker2**, the following holds for all $\gamma \in \Gamma$ and $\epsilon > 0$:

$$\begin{aligned} q^{\text{tracker2}}(\gamma) &\geq \sum_i q_i(\gamma a_i y_i^{**}(\gamma) + (\pi''_{i,**}(\gamma) - \epsilon)) && \text{[discussion above]} \\ &= q^{**}(\gamma) - \sum_i q_i((1 - \gamma)a_i y_i^{**}(\gamma) + \epsilon) && \text{[rearranging terms]} \end{aligned}$$

Algorithm 7 Tracker2($\pi_{**}'(\gamma), \mathbf{y}^{**}(\gamma)$)

```

1:  $x_i'' = 0$  for all  $i \in [N]$                                 % initialize long-run allocations corresponding to  $J''$ 
2: for  $t = 1$  to  $T$  do
3:    $x'_{it} = 0$  for all  $i \in [N]$                                 % initialize period  $t$  allocations corresponding to  $J'$ 
4:    $b_{it} = b_i$  for all  $i \in [N]$                                 % initialize bandwidth for this period
5:   for  $j = 1$  to  $J_t$  do
6:     if  $x'_{it} < a_i$  for some  $i$  s.t.  $y_i^{**}(\gamma) > 0$  then
7:        $\mathbb{W}_{jt} = \{i : y_i^{**}(\gamma) > 0, x'_{it} < a_i\}$           % active workers with min. requirement unmet
8:        $i^* \sim \mathbb{W}_{jt}$                                           % any worker from  $\mathbb{W}_{jt}$ 
9:        $x'_{i^*t} \leftarrow x'_{i^*t} + 1$                                 % allocate
10:       $b_{i^*t} \leftarrow b_{i^*t} - 1$                             % remaining bandwidth
11:     else
12:        $\mathbb{W}_{jt} = \{i : y_i^{**}(\gamma) > 0, b_{it} \geq 1\}$           % active workers with bandwidth
13:        $\delta_{ijt} = t\pi_{i,**}'(\gamma) - x''_i$  for all  $i \in \mathbb{W}_{jt}$     % level of under-allocation/over-allocation
14:        $i^* \sim \arg \max_{i \in \mathbb{W}_{jt}} \delta_{ijt}$                     % worker who is most under-allocated / least over-allocated
15:        $x''_{i^*} \leftarrow x''_{i^*} + 1$                                 % allocate
16:        $b_{i^*t} \leftarrow b_{i^*t} - 1$                             % remaining bandwidth
17:     end if
18:   end for
19: end for
20: return  $(\mathbf{x}', \mathbf{x}'')$ 

```

$$\begin{aligned}
&\geq q^*(\gamma) - \sum_i q_i ((1-\gamma)a_i y_i^{**}(\gamma) + \epsilon) && [q^{**}(\gamma) \geq q^*(\gamma)] \\
&\geq q^*(\gamma) - \sum_i ((1-\gamma)a_i y_i^{**}(\gamma) + \epsilon) && [\max_i q_i \leq 1] \\
&\geq q^*(\gamma) - ((1-\gamma)F^{-1}(1-\gamma) + N\epsilon) && [\sum_i a_i y_i^{**}(\gamma) \leq F^{-1}(1-\gamma)] \\
&= q^*(\gamma) - (1-\gamma)F^{-1}(1-\gamma) - \epsilon'. && [\epsilon' := N\epsilon]
\end{aligned}$$

where $q^{**}(\gamma) \geq q^*(\gamma)$ holds because the approximation we make in our objective is an upper bound to the true objective (discussed above). We summarize this in Theorem 3.

Theorem 3. *Tracker2 ensures every active worker receives her minimum requirement w.p. at least γ in each period while achieving a near-optimal allocation quality, i.e., for all $\gamma \in \Gamma$ and $\epsilon > 0$:*

$$\underbrace{q^{\text{tracker2}}(\gamma)}_{\text{Tracker2}} \geq \underbrace{q^*(\gamma)}_{\text{opt.}} - \underbrace{(1-\gamma)F^{-1}(1-\gamma) - \epsilon}_{\text{approximation error}}.$$

Clearly, our performance guarantee is stronger for a higher γ , with **Tracker2** approaching optimality as γ approaches 1 (which naturally is the regime of interest as we prefer giving more stable allocations to the workers). We found the empirical performance of **Tracker2** to be better than what is suggested by this bound. For example, with $\gamma = 0.9$, the above bound suggested a gap of $\approx 10\%$, but we found **Tracker2** to be within 1% of optimality, where we evaluated this 1% via the techniques we used to evaluate the “quality ratio” in §6.2 (recall $\dot{q}$).

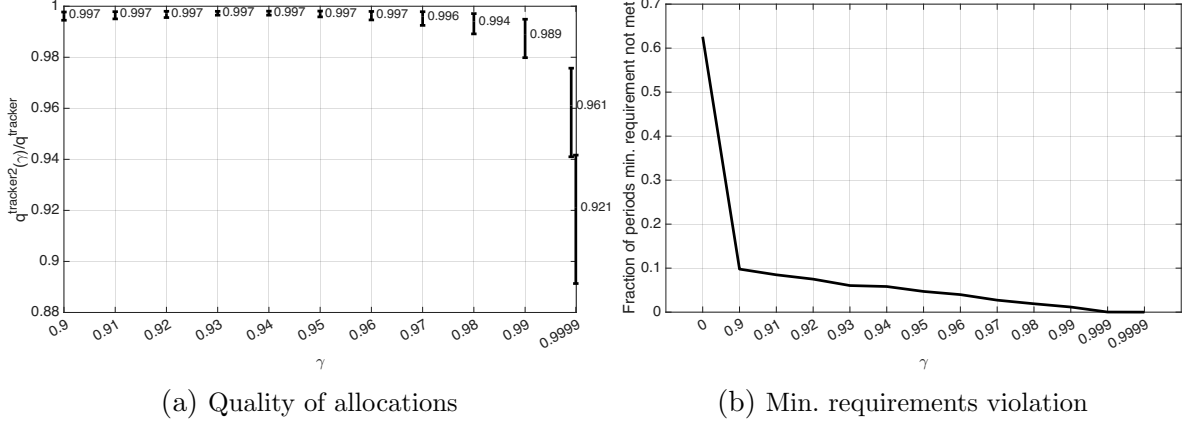


Figure 16 Results for Tracker2 for $\gamma \in \{0.90, 0.91, \dots, 0.99, 0.999, 0.9999\}$. In subplot (a), we summarize the results for all 100 seeds with the mean value in text and the bars corresponding to 95% empirical intervals (over 100 seeds). In subplot (b), we show the results for seed = 1 and note that other seeds have similar results.

E.3. Numerical Results and Discussion

We now numerically evaluate the performance of **Tracker2** on the setup presented in §6.1. Our primary goal here is to understand how the allocation quality decreases as we provide more stable allocations to workers (by increasing γ). To do so, for each of the 100 seeds used in §6, we solve (34) and simulate **Tracker2** for $\gamma \in \{0.90, 0.91, \dots, 0.99, 0.999, 0.9999\}$ ²⁸. To understand how much we lose in allocation quality, we divide the allocation quality achieved by **Tracker2** (i.e., $q^{\text{tracker2}}(\gamma)$) by the allocation quality **Tracker** achieved in the §6 results (i.e., q^{tracker}). Note that the denominator corresponds to not enforcing the “w.p. at least γ ” constraints but the “on average” constraints and is independent of γ . We show the corresponding results in Fig. 16a, where we plot this ratio as a function of γ . It is natural to expect this ratio to decrease in γ , but it is remarkable that it is very close to 1 for γ as high as 0.99. That is, the platform loses very little ($\approx 1\%$ or less) in allocation quality by providing a 99% guarantee to workers.

In Fig. 16b, we plot the fraction of periods for which the minimum requirements constraint is violated as a function of γ , with $\gamma = 0$ corresponding to the §6 results (i.e., **Tracker**). Of course, for $\gamma > 0$, we expect to see the y-value to be less than equal to $1 - \gamma$, and the plot serves as a sanity check. However, it is worth noting that **Tracker** ($\gamma = 0$) does not obey these requirements in $\approx 63\%$ of the periods (as alluded to at the beginning of this section). Putting Figures 16a and 16b together, we conclude that it is possible to provide much more stable allocations to workers while taking an extremely small hit on the allocation quality. Furthermore, we note that by implementing **Tracker2** and supplementing it with a small payment in periods where the min. requirement is not met, the platform can provide a w.p. 1 guarantee at a modest cost.

Given the minimum allocation constraints are at a worker level, we zoom into specific workers to understand how their allocation distributions change as a function of γ . We do so in Fig. 17, where we plot the cdf of

²⁸ It is possible that some of these γ values do not lie in Γ (the set of “efficient” γ values). Irrespective, our framework is well-defined and we can simulate **Tracker2**. Hence, the results here showcase the robustness of our theory to γ values outside Γ .

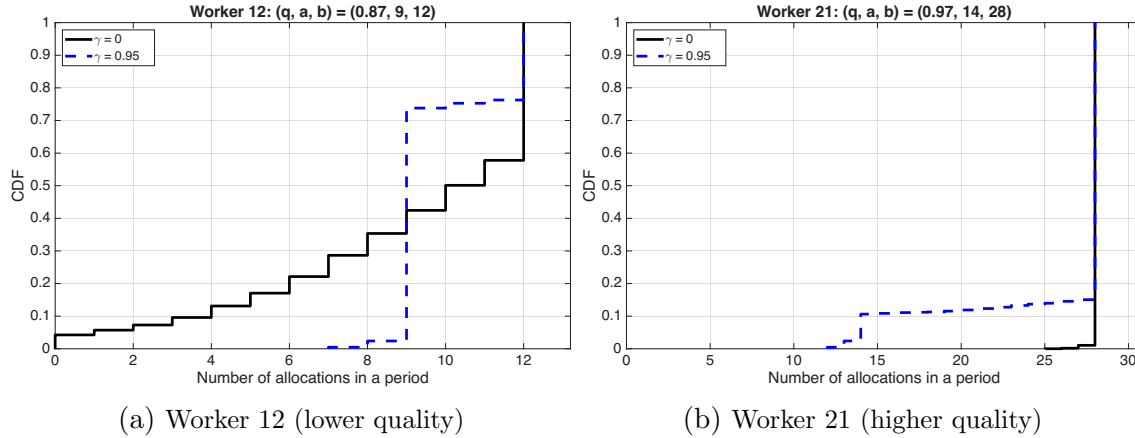


Figure 17 Allocation distributions (seed = 1) for two workers with $\gamma = 0$ (Tracker) and $\gamma = 0.95$ (Tracker2).

the allocations received by workers 12 (Fig. 17a) and 21 (Fig. 17b) for $\gamma = 0$ (Tracker from §6) and 0.95 (Tracker2), with the cdf being over periods, i.e., $[x_{it}]_t$. As shown at the top of the subplots, worker 12 is lower quality than worker 21 (0.87 vs. 0.97). For $\gamma = 0$, worker 12 has a large spread in her allocation distribution, with the allocation being below her min. requirement of 9 in more than 30% of the periods. This 30% naturally drops to below 5% for $\gamma = 0.95$, resulting in worker 12 receiving much more stable allocations. However, there is no “free lunch” here and the increased stability for worker 12 comes with an increased variance for worker 21. For $\gamma = 0$, worker 21 receives an allocation equal to her bandwidth of 28 in almost every period, and this drops to around 85% of the periods for $\gamma = 0.95$. Irrespective, worker 21 still receives her min. requirement of 14 in more than 95% of the periods. Note that the average allocations to workers 12 and 21 change from (9.1, 28.0) to (9.7, 26.2) when we change $\gamma = 0$ to 0.95. That is, the lower (higher) quality worker gets more (less) allocations when we increase γ . However, as we have shown in Fig. 16, the overall impact on quality is very low.

Remark 4 (Group-Level Minimum Allocation Guarantees). We now show how the minimum allocation guarantees can be enforced at a group level. Consider an arbitrary group $\mathbb{W} \subseteq [N]$ of workers for which we wish to ensure they receive at least $a_{\mathbb{W}}$ allocations per period (on average). There can be multiple reasons to encode such a constraint, with “group fairness” being one (see Ma et al. (2022) and Ma et al. (2023) for example). Using the notation from the corresponding individual-level constraint (1), we wish to enforce

$$\sum_{i \in \mathbb{W}} \bar{x}_i \geq a_{\mathbb{W}} \text{ w.p. } 1. \quad (35)$$

If (35) does not hold, then no worker in $\mathbb{W}$ is active. Our framework generalizes gracefully to this extension. In particular, all we need to do is add the following set of (linear) constraints in addition to introducing a binary variable $y_{\mathbb{W}} \in \{0, 1\}$ in the deterministic relaxation (13):

$$\sum_{i \in \mathbb{W}} \pi_i \geq a_{\mathbb{W}} y_{\mathbb{W}} \quad (36a)$$

$$y_i \leq y_{\mathbb{W}} \quad \forall i \in \mathbb{W} \quad (36b)$$

$$y_{\mathbb{W}} \in \{0, 1\}. \quad (36c)$$

The binary variable y_W captures whether the group-level minimum requirement is met (cf. (36a)). If it is met, then the workers in the group are allowed to be active (cf. (36b)). Note that we allow the workers to remain inactive (i.e., $y_i = 0$ for some $i \in W$) even if the group-level requirement is met (i.e., $y_W = 1$), because workers might be inactive for other reasons (e.g., their individual-level requirements being unmet). One can easily adapt our NLP and SAA methods from §5.1 and §5.2 to handle (36). Furthermore, there can be multiple such groups (e.g., $W_1, W_2, \dots, W_G$) and our technique from above can be applied to all of them simultaneously if we introduce the corresponding set of constraints and binary variables.

Appendix F: Periodic Churn

Fig. 18a illustrates the convergence behavior of **Tracker** for the base case parameters used in §6. As expected, the avg. allocations converge to their target levels over time (recall that we simulate for $T = 5,000$ periods). Some “chatter” is visible in the early periods, which is natural: allocations may temporarily fluctuate before settling into their steady-state values. This initial noise can cause certain workers to fall slightly below their min. requirements early on. Fig. 18b highlights this phenomenon by plotting avg. allocations relative to the requirement levels.

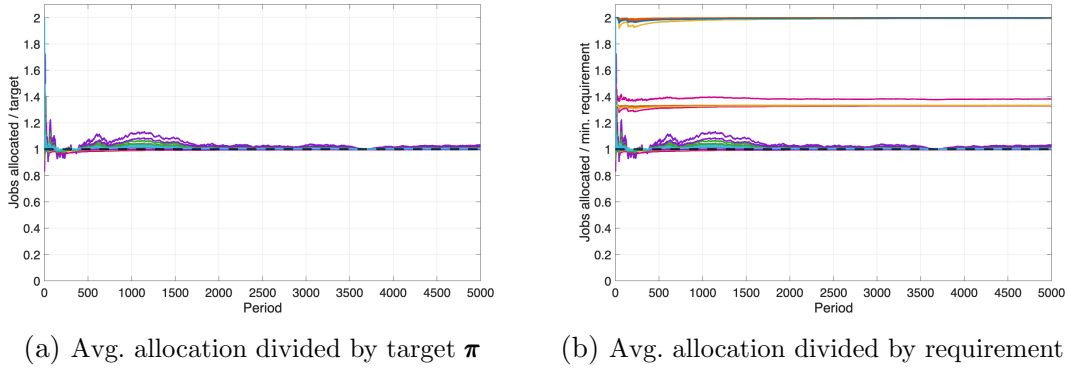


Figure 18 Visualizing the convergence behavior of **Tracker**. Each curve corresponds to an active worker i under **Tracker**, showing how the worker’s avg. per-period allocation evolves over time relative to the target level π_i (subplot (a)) and min. requirement a_i (subplot (b)). Naturally, in subplot (a), convergence corresponds to a y -value of 1. The converged y -values in subplot (b) are consistent with the distribution shown in Fig. 6e. Results are shown for seed = 1; other seeds yield qualitatively similar convergence patterns.

Up to this point, we have assumed that workers decide whether to stay or leave the platform based on their *long-run* avg. allocations. This implicitly assumes knowledge of steady-state performance and equilibrium outcomes. In practice, however, workers may not have such foresight and may, instead, periodically re-evaluate their participation using only the realized allocations. We refer to this dynamic as *periodic churn*. After every *evaluation interval* τ , each worker computes the ratio of her avg. allocation up to that time to the requirement, and remains active only if this ratio exceeds a *churn threshold* $(1 - \epsilon)$, where $\epsilon > 0$ is a leeway parameter. Otherwise, the worker exits the platform permanently. This periodic reassessment captures a realistic form of attrition.

Under **FCFS** and **Greedy**, we assume that all workers begin as active, and subsequently churn according to the periodic assessment rule. Under **Tracker**, only the workers admitted by the policy are initially active,²⁹ and if some of these active workers churn, the remaining workers continue to be tracked against their original targets. We do not re-optimize targets in response to churn, which makes our results for **Tracker** somewhat conservative.

We next quantify how periodic churn alters the quality ratio of each policy. Using the base case parameters from §6, we experiment with evaluation intervals $\tau \in \{3, 6, 12\}$ and churn thresholds $(1 - \epsilon) \in \{0.9, 0.95, 0.99\}$ across 100 seeds. (Interpreting each period as one month, $\tau = 3$ corresponds to workers evaluating their participation every three months.) Fig. 19 summarizes the resulting avg. quality ratios. Absent periodic churn, the avg. quality ratios were 0.87 for **FCFS**, 0.96 for **Greedy**, and 1.00 for **Tracker** (§6.2). As expected, performance deteriorates once churn is introduced.



Figure 19 Avg. quality ratio under periodic churn for the three policies. Each heatmap reports the avg. quality ratio across 100 seeds as a function of the evaluation interval τ (rows) and the churn threshold $(1 - \epsilon)$ (columns).

The definition of quality ratio remains the same as before, i.e., we use $\dot{q}$ in the denominator (see §6.2).

For **FCFS**, the avg. quality ratio drops sharply, from 0.87 to below 0.60 when $\epsilon = 0.1$, and to below 0.50 when $\epsilon = 0.01$. The reason is structural: under **FCFS**, job allocation is random and unrelated to worker quality, so temporary under-allocation drives workers of all qualities to exit uniformly, reducing both volume and quality of allocations. In contrast, **Greedy** is substantially more robust: the avg. quality ratio declines only slightly (to around 0.90), since high-quality workers almost never churn and thus the platform retains its best supply. **Tracker** exhibits intermediate robustness. For $\epsilon = 0.1$, its avg. quality ratio remains near-optimal (above 0.98), but it is more sensitive than **Greedy** to frequent evaluations or tight thresholds. For example, the avg. quality ratio falls to 0.72 for $(\tau, \epsilon) = (3, 0.01)$, since churn primarily affects workers who momentarily fall short of their targets, including some high-quality workers.

These results naturally raise the question of whether **Tracker** can be made more robust to periodic churn. In §E, we already took a step in this direction by proposing a variant, **Tracker2**, that provides per-period min. allocation guarantees with high probability. Specifically, recall that **Tracker2** ensures $\mathbb{P}\{x_{it} \geq a_i y_i \ \forall i\} \geq \gamma$ for all t (see (31e)). Such a guarantee intuitively mitigates the risk of temporary under-allocation caused by early-period noise. Fig. 20 shows **Tracker2** substantially reduces the under-allocation due to “initial chatter”

²⁹ Inactive workers remain inactive throughout; they are not permitted to join later.

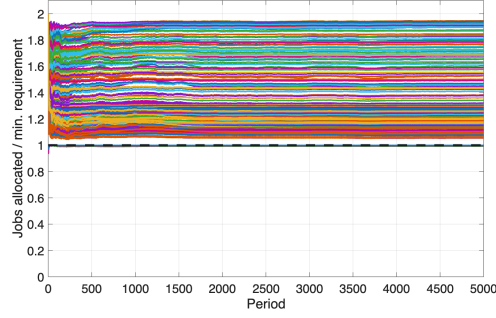


Figure 20 Avg. allocation divided by requirement a under Tracker2 for $\gamma = 0.95$ and seed = 1. Each curve corresponds to an active worker under Tracker2.

observed under Tracker. As a consequence, its avg. quality ratio under periodic churn remains near-optimal (Fig. 21).

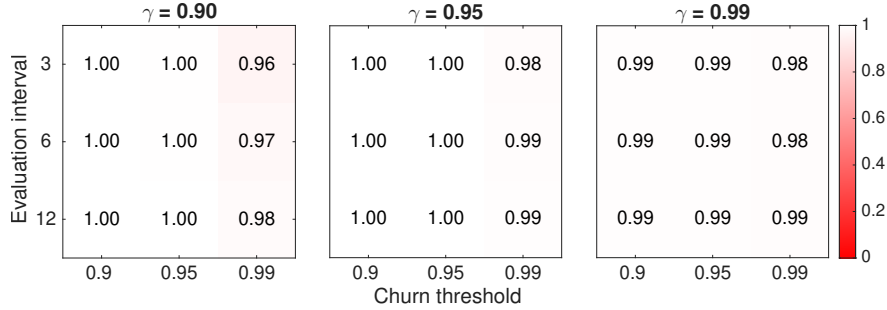


Figure 21 Avg. quality ratio under periodic churn for Tracker2 with $\gamma \in \{0.9, 0.95, 0.99\}$. For an apples-to-apples comparison with Fig. 19, the quality ratio is computed using the same normalization as before; specifically, the denominator equals the same $\hat{q}$ value used in Fig. 19.

It is possible, via a Bonferroni (union bound) argument, to bound the probability that any worker's running avg. allocation falls below $(1 - \epsilon)a_i$ as function of (γ, ϵ, τ) . We omit these details for brevity, but note that the analysis is relatively straightforward: the events $\{x_{it} \geq a_i y_i \forall i\}$ are independent across t under Tracker2, each occurring with probability at least γ , allowing one to treat $\{x_{it} \geq a_i y_i \forall i\}_{t=1}^T$ as independent Bernoulli trials.

Appendix G: Strategic Reporting

The parameters a_i and b_i are private information of each worker i . Throughout our analysis of Tracker so far, we assume that the platform knows these values, but in practice workers may strategically misreport them to improve their (expected) utility. Let $\tilde{a}$ and $\tilde{b}$ denote the reported values. *Incentive compatibility* (IC) would require that truthful reporting, i.e., $\tilde{a} = a$ and $\tilde{b} = b$, is optimal. This raises two natural questions: (1) Does IC hold under Tracker for a reasonable utility model?³⁰ (2) If not, how large is the performance loss when workers report strategically rather than truthfully?

³⁰ Since FCFS does not require workers to report either a or b , IC considerations do not arise for this policy. Depending on its implementation, Greedy may require workers to report their bandwidth b ; it is straightforward to verify that truthful reporting of b is a dominant strategy under Greedy.

In §G.1, we begin by addressing the first question when the workers have a linear utility function. This diagnostic step helps determine which dimensions of private information are strategically important. For this model, we show that truthful reporting of bandwidth is always optimal, whereas workers may find it beneficial to over-report their minimum requirement. Having identified over-reporting of $\mathbf{a}$ as the relevant strategic behavior, we examine its implications for **Tracker**. We first analytically characterize the *worst-case* impact of such misreporting (§G.2) and then numerically evaluate its “*average-case*” effects using a sequential best-response procedure (§G.3).

G.1. Incentive Compatibility under Tracker

In this section, we examine whether the workers have the incentive to truthfully report their private information under **Tracker** in a long-run equilibrium sense. Our goal is not to develop a full mechanism design model, but rather to identify which components of workers’ private parameters (minimum requirements $\mathbf{a}$ and bandwidths $\mathbf{b}$) are potentially susceptible to strategic misreporting. To this end, we begin with a simple and economically natural linear utility model that allows us to characterize workers’ incentives for misreporting in a transparent manner. This analysis reveals that the bandwidth $\mathbf{b}$ is always reported truthfully, whereas minimum requirement $\mathbf{a}$ may be over-reported. This insight guides the remainder of our strategic analysis in §G.2 and §G.3, which therefore focuses on the consequences of strategic inflation of $\mathbf{a}$. We also discuss an alternative (non-linear) utility model that induces incentive compatibility.

A reasonable model for utility is as follows. Each worker i has an outside option worth a_i jobs per period, which explains why her min. requirement is a_i . Her utility increases linearly with the average³¹ number of jobs she is allocated per period. Each worker i also faces a hard per-period cap b_i : she cannot physically complete more than b_i jobs. In addition, the platform employs a simple discipline mechanism: if a worker “shirks” (i.e., fails to complete allocated jobs despite the allocation respecting her reported capacity $\tilde{b}_i$), she is deactivated forever. This discourages workers from overstating their bandwidth.

We first consider IC w.r.t. b_i . All else equal, reporting a higher $\tilde{b}_i$ makes a worker appear more attractive to **Tracker**, as she can absorb more load (see (13)). Thus, she has no incentive to under-report her bandwidth ($\tilde{b}_i < b_i$). On the other hand, overstating bandwidth ($\tilde{b}_i > b_i$) risks deactivation due to shirking: for a worker to benefit from $\tilde{b}_i > b_i$, **Tracker** would have to allocate more than b_i jobs in some periods; otherwise the same allocation could be achieved with truthful reporting (see (13)). Consequently, over-reporting can only hurt, and truthful reporting of b_i is optimal (in the long term).

Next, we consider IC w.r.t. a_i . A worker never finds it optimal to under-report ($\tilde{a}_i < a_i$). While doing so may make her more attractive to **Tracker** and potentially activate her when she would otherwise be inactive (see (13)), this would result in her receiving fewer than a_i jobs on average, which is worse than her outside option. In contrast, over-reporting ($\tilde{a}_i > a_i$) may improve her allocation: a higher reported min. requirement can induce **Tracker** to allocate a larger average number of jobs. It is straightforward to construct a counterexample illustrating such profitable over-reporting.

³¹ We define utility in terms of the *average* per-period allocation; equivalently, one could define utility on realized per-period allocations and then compute expected utility by averaging over periods. Because utility is linear in the number of jobs, these two definitions coincide.

The incentive to over-report a_i is a function of the linear utility model. We now discuss a stylized utility model that is IC w.r.t. a_i . Retain the same outside option and hard cap as above, but assume a worker’s utility is *constant* whenever her average³² per-period allocation lies between a_i and b_i , with this constant strictly exceeding the outside option. This corresponds to a “binary” notion of utility capturing whether the worker is “happy”. The platform continues to use deactivation for shirking. Now the worker is indifferent as long as she receives at least a_i jobs and hence, there is no incentive to over-report her min. requirement. Moreover, all previous arguments ensuring $\tilde{b}_i = b_i$ and $\tilde{a}_i \geq a_i$ remain valid.

G.2. Analytical Characterization of the Worst-Case Performance Loss

Surprisingly, we can bound the worst-case performance loss due to strategic over-reporting. Observe that from the platform’s perspective, the worst case corresponds to workers reporting $\tilde{\mathbf{a}} = \mathbf{b}$, i.e., each worker declares her min. requirement to be equal to her bandwidth. This results in the smallest feasibility set under **Tracker** (see (13)), and is, therefore, the most adverse scenario. With $\tilde{\mathbf{a}} = \tilde{\mathbf{b}} = \mathbf{b}$, (13) reduces to a knapsack problem (see §C.3 for details). Therefore, Prop. 9 implies that, even in the worst case ($\tilde{\mathbf{a}} = \mathbf{b}$), **Tracker** performs at least as well as **Greedy** up to an additive $q_{\max} b_{\max}$ term. One would want to conclude that $\mathbf{Tracker} \geq \mathbf{Greedy} \geq \mathbf{FCFS}$, but as discussed at the end of §3, the second inequality need not always hold. Nonetheless, this ordering does hold in our “average-case” numerical experiments, which we discuss next.

G.3. Numerical Experiments Quantifying Average-Case Performance Loss

In order to understand the “average-case” performance loss due to strategic reporting, we numerically compute an equilibrium via a sequential best-response procedure. We initialize with truthful reports $\tilde{\mathbf{a}} = \mathbf{a}$, and then loop over workers one at a time, allowing worker i to select her optimal report $\tilde{a}_i$ while holding $\tilde{\mathbf{a}}_{-i}$ fixed. To determine her best response, worker i evaluates every report $\tilde{a}_i \in \{a_i, a_i + 1, \dots, b_i\}$; each such evaluation requires solving the underlying optimization problem, which we implement using the SAA formulation (17) with $S = 100$. After each worker updates her report, we recompute the resulting allocations and continue this process until no worker can improve her allocation by unilaterally deviating. The resulting outcome is a stable equilibrium under strategic reporting of $\tilde{\mathbf{a}}$.

This procedure is computationally intensive because it requires repeatedly solving SAA instances (17). To keep compute times manageable, we scale down the problem size by a factor of ten: specifically, compared to §6, we reduce N from 400 to 40 and proportionally scale down demand $(\mu, \sigma) = (4000, 400)$ to $(400, 40)$. Since $N = 40$ is not a “large” market, we enforce the binary constraints in (17). We cap the number of iterations at 1,000, where each iteration consists of a pass over all workers, and randomly permute the worker order before each iteration to discourage cycling. We also impose a 10-hour runtime limit and incorporate several structural insights to accelerate the search. For example, if worker i is currently active, it suffices to evaluate reports in $\{\pi_i^{\text{current}} + 1, \dots, b_i\}$, and if she is currently inactive, we only need $\{a_i, \dots, \tilde{a}_i^{\text{current}} - 1\}$. In both

³² As before, one could instead define utility in terms of realized allocations; however, because utility is now non-linear, these two definitions are no longer equivalent. In such a setting, the w.h.p. variant **Tracker2** (§E) is more appropriate, as a worker’s expected utility is proportional to the fraction of periods in which she receives at least a_i jobs, which aligns with the parameter γ .

cases, the search can be truncated early once activation/inactivation switches occur. As before, we conduct this experiment over 100 independent seeds.

Of the 100 seeds, 34 converged within the limits above. Among these converged seeds, the average number of iterations was 106 (range $[4, 885]$). Fig. 22a summarizes the resulting quality ratios.³³ As expected, FCFS and Greedy are unaffected by strategic reporting and therefore exhibit quality ratios similar to those in §6.2. Strategic reporting affects Tracker, whose quality ratio is no longer 1, but remains high: its average is 0.974, outperforming both FCFS and Greedy. Thus, the average performance loss induced by strategic reporting is modest (approx. $1 - 0.974 = 2.6\%$).

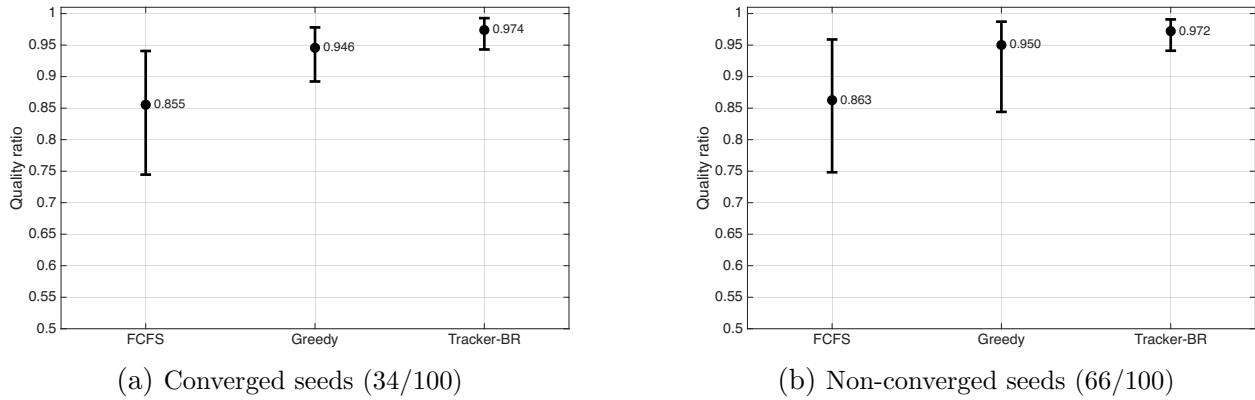


Figure 22 Quality ratios under sequential best-response over $\tilde{\mathbf{a}}$. Bars indicate 95% empirical intervals, with mean values shown in text. Since FCFS and Greedy are unaffected by strategic reporting, we continue to denote them by FCFS and Greedy. In contrast, strategic reporting alters the performance of Tracker, so we refer to its strategic variant as Tracker-BR, where “BR” denotes best-response.

The remaining 66 seeds did not converge: 61 reached the 1,000-iteration cap, while 5 hit the 10-hour limit (each with more than 400 iterations). Fig. 23 illustrates a representative non-converged seed. The time series of reported $\tilde{\mathbf{a}}$ (Fig. 23a) shows persistent, small adjustments across workers. These marginally beneficial local deviations continually reshuffle allocations, creating long transients without clear cycles. The corresponding quality ratio (Fig. 23b) fluctuates within a narrow band. To smooth these fluctuations, we average the quality ratios over the last 100 iterations (for each seed); the resulting values are summarized in Fig. 22b. Comparing Fig. 22a and Fig. 22b, we find that converged and non-converged seeds achieve similar performance.

To further probe convergence behavior, we repeated the experiment on an even smaller market with $N = 10$ and $(\mu, \sigma) = (100, 10)$. Out of 100 seeds, 64 converged (average of 27 iterations; range $[1, 182]$), while the remaining 36 reached the 1,000-iteration cap. Nevertheless, the quality ratios across converged and non-converged seeds were similar. Averaging over all 100 seeds, we obtained quality ratios 0.84 (FCFS) $\rightarrow$ 0.90 (Greedy) $\rightarrow$ 0.96 (Tracker-BR).

These observations raise several natural questions: What drives equilibrium existence? What do equilibria look like when they exist? What is an optimal platform policy under strategic considerations? Though we

³³ For the denominator, we use the optimal objective value of the SAA formulation (17) (with binary constraints) under truthful reporting $\tilde{\mathbf{a}} = \mathbf{a}$.

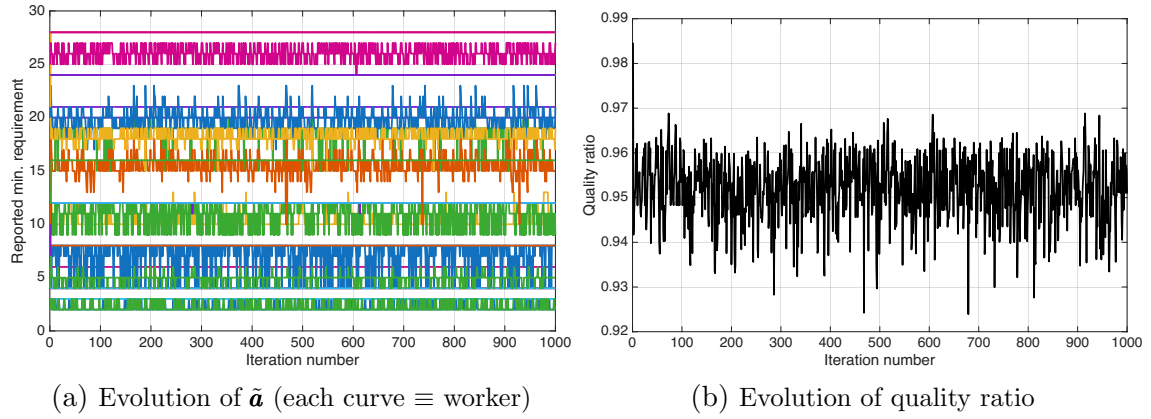


Figure 23 A representative non-converged seed: prolonged transients without fixed-point stabilization.

suspect that such strategic misreporting might not be too prevalent in practice³⁴, addressing these questions in full generality is beyond the scope of this paper, and our results here offer some preliminary insights.

³⁴ This is because the mapping from the problem primitives to the targets **Tracker** tracks is quite complex, making it hard for workers to evaluate their expected utility ex-ante.