A. On Justification of Assumptions

Assumption 1 can be made with a loss of $(1+\epsilon)$-factor in the competitive ratio. If $\frac{p_{i',j}}{p_{i,j}} \geq \frac{m}{\epsilon}$ for some $j \in J, i, i' \in M_j$, we can change $p_{i',j}$ to ∞. If the job j is assigned to i' in the optimum solution, we assign it to the machine i^* with the minimum $p_{i^*,j}$ instead. Thus, the processing time of j is decreased by at least a factor of $\frac{m}{\epsilon}$. We apply the operation for all violations of the assumption. Then the makespan of a machine i will be increased by at most $\frac{(m-1)T}{m/\epsilon} \leq \epsilon T$. This holds since the total processing time of machines other than i in the optimum solution is at most $(m-1)T$. We also remark the procedure that guarantees the assumption can run online, as jobs are handled separately in the procedure.

Consider Assumption 2 for designing online rounding algorithms. We show the assumption can be made by losing a factor of 4 in the competitive ratio. Suppose when T is known the algorithm has competitive ratio α.

We start from $T = 0$. The algorithm is broken into phases. Within each phase, the T value does not change, and it is at least $\mathrm{mspn}(x)$ for any x we see in the phase. Within each phase, we run the α-competitive rounding algorithm with the T value. Upon the arrival of a client j, we check if $\mathrm{mspn}(x)$ exceeds T for the updated x. If yes we then change T to $2 \cdot \mathrm{mspn}(x)$ and start a new phase.

In each phase, the α-competitive rounding algorithm gives an assignment of makespan at most αT. The values of T at least double from phase to phase, and the value of T in the last phase is at most $2\mathrm{mspn}(x)$ for the final x. Therefore, the makespan of the assignment produced by the online rounding algorithm is at most $\alpha \cdot 2 \cdot \mathrm{mspn}(x) \cdot (1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots) \leq 4\alpha \cdot \mathrm{mspn}(x)$, resulting in an 4α-competitive online rounding algorithm in the case when T is not known.

There is a small caveat on the failure probability when the rounding algorithm is randomized. The proof works only if the number of phases is polynomial in m, since the failure probability is multiplied by the number of phases. This holds when $\frac{\max_{(i,j)\in E} p_{i,j}}{\min_{(i,j)\in E} p_{i,j}} \leq 2^{\mathrm{poly}(m)}$, which is a mild condition. Indeed, if $\frac{\max_{(i,j)\in E} p_{i,j}}{\min_{(i,j)\in E} p_{i,j}} \gg 2^{\mathrm{poly}(m)}$, then an adversary can release super-polynomial number of instances sequentially, so that the total makespan of all previous instances is neglectable compared to the current one. Then the failure probability has to scale by the number of instances.

B. Omitted Proofs

B.1. Proportional Allocation Scheme of Agrawal et al. and Proof of Lemma 2.3

In this section, we first describe the proportional allocation scheme of Agrawal et al. (2018) for the maximum throughput problem in the P|restricted setting. As usual we are given $M, J, |M| = m$ and $|J| = n$. Every job $j \in J$ has a size $p_j \in \mathbb{R}_{>0}$ and $p_{i,j} \in \{p_j, \infty\}$ for every i, j. E, M_j's and J_i's are defined as before. They considered a more general setting where every machine i is given a makespan budget $T_i > 0$. A valid fractional solution to the instance is a vector $x \in [0,1]^E$ such that $\sum_{i\in M_j} x_{i,j} = 1$ for every $j \in J$. The fractional throughput of x is defined as

$$\mathrm{Thr}(x) := \sum_{i\in M} \min\left\{T_i, \sum_{j\in J_i} p_j x_{i,j}\right\}.$$

So, the portion of the load on a machine i that exceeds T_i is discarded and not considered in the throughput. (It does not matter what fractional jobs we discard.) The optimum fractional makespan is then the maximum of $\mathrm{Thr}(x)$ over all valid fractional solutions x. We call the problem the *throughput maximization* problem in the P|restricted setting, to distinguish it from the load balancing problem we are considering.

Given a weight vector $w \in \mathbb{R}^M_{>0}$, recall that the fractional solution $x^{(w)} \in [0,1]^E$ assigns every job j to the machines M_j proportionally to their weights. The main result of Agrawal et al. (2018) is that some weight vector w gives a $(1-\delta)$-optimum solution $x^{(w)}$ for any $\delta \in (0,1)$:

Theorem B.1 (Agrawal et al. (2018))**.** *Given a throughput maximization problem in the* P|restricted *setting, and $\delta \in (0,1)$, there exits a vector $w \in \mathbb{R}^M_{>0}$ such that $\mathrm{Thr}(x^{(w)})$ is at least $1-\delta$ times the optimum fractional makespan.*

Theorem B.1 is Theorem 1 of Agrawal et al. without considering the precision requirement of w; we handle the precision issue inside the proof of Lemma 2.3, which is repeated below.

Lemma 2.3. *Given a load balancing instance in the* Q|restricted *setting, for any $\epsilon \in (0,1)$, there is a weight vector $w \in \mathrm{powers}^M_{1+\epsilon,K}$ for some $K = O\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ such that $x^{(w)}$ is a $(1+\epsilon)^3$-approximate solution to* (P-LP)*.*

Proof. Recall that in our load balancing instance, every job $j \in J$ has a size p_j and every machine $i \in M$ has a speed s_i, and $p_{i,j} = \frac{p_j}{s_i}$ for every $(i,j) \in E$. Let T be the optimum value of (P-LP) for the instance.

To construct a throughput maximization instance in the P|restricted setting, we set $T_i = Ts_i$, which is total size of jobs that can be processed on machine i in time T. The p_j values in the instance are the same as that in the load balancing instance. Since (P-LP) has a fractional solution of makespan at most T, the throughput maximization instance has a fractional solution with throughput $\sum_{j\in J} p_j$.

Let $s_{\max} = \max_{i\in M} s_i$ and $s_{\min} = \min_{i\in M} s_i$. We set $\delta = \frac{\epsilon \cdot s_{\min}}{m \cdot s_{\max}}$ and apply Theorem 2.4. Then, we have a vector $w \in \mathbb{R}^M_{>0}$ such that $\mathrm{Thr}(x^{(w)}) \geq (1-\delta)\sum_{j\in J} p_j$. So at most $\delta \sum_{j\in J} p_j$ total size of fractional jobs are discarded.

Let y_j be the fraction of the job j that is discarded. Then, we have $\sum_{j\in J} y_j p_j \le \delta \sum_{j\in J} p_j$.

We then go back the original load balancing instance in the Q|restricted setting. Without considering the discarded fractional jobs, all machines have makespan at most T. Even if all these fractional jobs are scheduled in the slowest machine before discarded, the total time for processing them will be at most $\frac{\delta\sum_{j\in J} p_j}{s_{\min}} = \frac{\epsilon\sum_{j\in J} p_j}{ms_{\max}} \le \epsilon T$, where the last inequality holds since $\sum_{j\in J} p_j \le mTs_{\max}$. Therefore, $x^{(w)}$ has makespan at most $(1+\epsilon)T$.

Finally, we make the aspect ratio of w small using the following procedure. We sort all the jobs according to their w values from the smallest to the biggest. Whenever we see two adjacent jobs j_1, j_2 in the ordering with $\frac{w_{j_2}}{w_{j_1}} > \frac{m^2}{\epsilon^2}$, we scale down the w values of all jobs after j_1 in the sequence by the same factor so that $\frac{w_{j_2}}{w_{j_1}}$ becomes $\frac{m^2}{\epsilon^2}$. So, after the operation, the aspect ratio of w becomes at most $\left(\frac{m^2}{\epsilon^2}\right)^{m-1}$.

Due to the procedure, some $x^{(w)}_{i,j}$ values increase. However, they will never be increased to more than $\frac{1}{1+m^2/\epsilon^2} < \frac{\epsilon^2}{m^2}$; this holds since if some $x^{(w)}_{i,j}$ is increased, then after the procedure, there must be some other job $i' \in M_j$ with $w_{i'} \ge \frac{m^2}{\epsilon^2} w_i$. The total time of running all jobs in their respective fastest permissible machines is at most mT. Running $\frac{\epsilon^2}{m^2}$ fraction of all jobs in J_i on a machine i takes time at most $\frac{\epsilon^2}{m^2}\cdot mT\cdot\frac{m}{\epsilon} = \epsilon T$, where the factor of $\frac{m}{\epsilon}$ comes from Assumption 1. Therefore, the procedure increases the makespan by at most ϵT. So, for the new w, $x^{(w)}$ has makespan at most $(1+2\epsilon)T$.

Then we round each w_i value down to its nearest integer power of $1+\epsilon$. This will increase the makespan of any machine by at most a multiplicative factor of $(1+\epsilon)$. Therefore, our final w has coordinates in powers_K with $K = \left\lfloor \log_{1+\epsilon}(m^2/\epsilon^2)^{m-1}\right\rfloor = O\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$. The makespan of $x^{(w)}$ is at most $(1+\epsilon)^3 T$. □

B.2. Omitted Proofs in Sections 3 and 4

Lemma 3.1. *For every $t \in [n]$, we have $\Phi_t \le \Phi_{t-1}$.*

Proof. Assume we are at the beginning of some time $t \in [n]$ in the algorithm. Now suppose at time t, instead of assigning job t deterministically as in the algorithm, we randomly assign t to a machine, such that the probability that t is assigned to i is $x_{i,t}$. We upper bound $\mathbb{E}[\Phi_t]$ by Φ_{t-1}:

$$
\begin{aligned}
&\mathbb{E}[\Phi_t] \\
&= \sum_{i\in M} \mathbb{E}\left[\exp\left(\frac{aL_{i,t}}{T} + (e^a-1)\Big(1-\frac{1}{T}\sum_{j=1}^{t} x_{i,j}p_{i,j}\Big)\right)\right] \\
&= \sum_{i\in M} \exp\left(\frac{aL_{i,t-1}}{T} + (e^a-1)\Big(1-\frac{1}{T}\sum_{j=1}^{t} x_{i,j}p_{i,j}\Big)\right) \\
&\quad\cdot\left(x_{i,t}e^{\frac{ap_{i,t}}{T}} + 1 - x_{i,t}\right) \\
&\le \sum_{i\in M} \exp\left(\frac{aL_{i,t-1}}{T} + (e^a-1)\Big(1-\frac{1}{T}\sum_{j=1}^{t} x_{i,j}p_{i,j}\Big)\right) \\
&\quad\cdot \exp\left((e^a-1)\frac{x_{i,t}p_{i,t}}{T}\right) \\
&= \sum_{i\in M} \exp\left(\frac{aL_{i,t-1}}{T} + (e^a-1)\Big(1-\frac{1}{T}\sum_{j=1}^{t-1} x_{i,j}p_{i,j}\Big)\right) \\
&= \Phi_{t-1}.
\end{aligned}
$$

The first equation used is just by the definition of Φ_t and linearity of expectation. For the second equation, notice that for every $i \in M$, we have $L_{i,t} = L_{i,t-1} + p_{i,t}$ with probability $x_{i,t}$ and $L_{i,t} = L_{i,t-1}$ with probability $1 - x_{i,t}$. To see the inequality, we define θ to be $\frac{p_{i,t}}{T}$ with probability $x_{i,j}$ and 0 with probability $1 - x_{i,j}$. Since $\exp(a\cdot\theta)$ is a convex function on θ, and θ is a random variable over $[0,1]$, we have

$$
\begin{aligned}
&\left(x_{i,j}e^{\frac{ap_{i,j}}{T}} + 1 - x_{i,j}\right) = \mathbb{E}[\exp(a\cdot\theta)] \\
&\le (1-\mathbb{E}[\theta])\cdot 1 + \mathbb{E}[\theta]\cdot e^a = 1 + (e^a-1)\,\mathbb{E}[\theta] \\
&= 1 + (e^a-1)\frac{x_{i,t}p_{i,t}}{T} \\
&\le \exp\left((e^a-1)\frac{x_{i,t}p_{i,t}}{T}\right).
\end{aligned}
$$

The last equality used the definition of Φ_{t-1}.

We have proved $\mathbb{E}[\Phi_t] \le \Phi_{t-1}$. In our actual deterministic algorithm, we assign t to the machine i that minimizes Φ_t. So $\Phi_t \le \Phi_{t-1}$. □

Lemma 4.1. *With probability at least $1 - \frac{1}{4m}$, $\forall i \in M$, the total load of small jobs assigned to i is at most $8T$.*

Proof. For every $i \in M, j \in J_i^{\text{small}}$, let $\tilde{x}_{i,j} \in \{0,1\}$ indicate whether j is assigned to i or not in the assignment we constructed. Focus on each $i \in M$ and we shall apply Chernoff bound (Theorem E.1) to the sum $\sum_{j\in J_i^{\text{small}}} \frac{\rho p_{i,j}}{T}\tilde{x}_{i,j}$. For any small job $j \in J_i^{\text{small}}$, $p_{i,j} \le \frac{T}{\rho}$ and thus we always have $\frac{\rho p_{i,j}}{T}\tilde{x}_{i,j} \in [0,1]$. The expectation of the sum is $\mu := 2\sum_{j\in J_i^{\text{small}}} \frac{\rho p_{i,j}}{T} x_{i,j} \le 2\rho$ by (P4). Applying Chernoff bound with $U = 2\rho$ and $\delta = 3$ gives us

$$
\Pr\left[\sum_{j\in J_i^{\text{small}}} \frac{\rho p_{i,j}}{T}\tilde{x}_{i,j} > 8\rho\right] < e^{-3^2\cdot 2\rho/5} \le \frac{1}{4m^2}.
$$

The event in the bracket is precisely $\sum_{j\in J_i^{\text{small}}} p_{i,j}\tilde{x}_{i,j} > 8T$. The lemma follows by applying the union bound over all machines i. □

Lemma 4.2. *With probability at least $1 - \frac{1}{4m}$, for every $i \in M$, we have $\sum_{j \in J_i^{\text{big}}} p_{i,j} x'_{i,j} \le 5T$.*

Proof. Focus on each $i \in M$. Let $J' = \{j \in J_i^{\text{big}} : x_{i,j} < 1/\rho\}$. Notice that for every $j \in J'$ we have $\mathbb{E}[x'_{i,j}] = x_{i,j}$ and $\frac{\rho p_{i,j}}{T} x'_{i,j} \in [0,1]$. Moreover, the random variables $\{x'_{i,j}\}_{j \in J'}$ are independent. So, we can apply Chernoff bound (Theorem E.1) to the sum $\sum_{j \in J'} \frac{\rho p_{i,j}}{T} x'_{i,j}$. Its expectation is $\mu := \sum_{j \in J'} \frac{\rho p_{i,j}}{T} x_{i,j} \le \rho$ by (P4). Applying the bound with $U = \rho$ and $\delta = 4$ gives us

$$\Pr\left[\sum_{j \in J'} \frac{\rho p_{i,j}}{T} x'_{i,j} > \sum_{j \in J'} \frac{\rho p_{i,j}}{T} x_{i,j} + 4\rho\right] < e^{-\frac{4^2 \rho}{6}} \le \frac{1}{4m^2}.$$

Focus on the inequality in the bracket above. Multiplying both sides by T/ρ and adding $\sum_{j \in J_i^{\text{big}} \setminus J'} x'_{i,j} p_{i,j} = \sum_{j \in J_i^{\text{big}} \setminus J'} x_{i,j} p_{i,j}$ to both sides, the inequality becomes

$$\sum_{j \in J_i^{\text{big}}} x'_{i,j} p_{i,j} > \sum_{j \in J_i^{\text{big}}} x_{i,j} p_{i,j} + 4T.$$

The inequality is implied by $\sum_{j \in J_i^{\text{big}}} x'_{i,j} p_{i,j} > 5T$. Thus, $\Pr\left[\sum_{j \in J_i^{\text{big}}} x'_{i,j} p_{i,j} > 5T\right] < \frac{1}{4m^2}$. The lemma holds by the union bound over all machines $i \in M$. □

Lemma 4.3. *For every $i \in M$, we have $\Pr[i \in M^{\text{marked}}] \le \frac{1}{700\rho^{12}}$.*

Proof. Let $\tilde{x}_{i,j}$ indicate whether we are trying to assign j to i or not in Algorithm 1. If i is marked, then $\sum_{j \in J_i^{\text{big}}} \frac{\tilde{x}_{i,j} p_{i,j}}{T} > \frac{15 \log\log m}{\log\log\log m}$. Note that $p_{i,j} \le T$, $0 \le x'_{i,j} \le 1$ and $\mathbb{E}\left[\sum_{j \in J_i^{\text{big}}} \frac{p_{i,j} \tilde{x}_{i,j}}{T}\right] \le 3 \sum_{j \in J_i^{\text{big}}} \frac{p_{i,j} x'_{i,j}}{T} \le 15$ by Lemma 4.2. Applying Chernoff bound (Theorem E.1) to the sum $\sum_{j \in J_i^{\text{big}}} \frac{p_{i,j} \tilde{x}_{i,j}}{T}$ with $\delta = \frac{\log\log m}{\log\log\log m} - 1$ and $U = 15$, we have that

$$\Pr[i \in M^{\text{marked}}] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^U \le \left(\frac{e}{1+\delta}\right)^{(1+\delta)U} = \left(\frac{e \log\log\log m}{\log\log m}\right)^{\frac{15\log\log m}{\log\log\log m}} < \frac{1}{700\rho^{12}}.$$

This finishes the proof. □

Claim 4.4. *In G', every job $j \in J^{\text{big}}$ has degree at most $\rho/2 + 1$, and every machine $i \in M$ has degree at most $5\rho^2$. $G'^2[M]$ has maximum degree at most $5\rho^3/2$, $G'^4[M]$ has maximum degree at most $25\rho^6/4$, and $G'^8[M]$ has maximum degree at most $625\rho^{12}/16$.*

Proof. The first half of the first sentence follows from that $x'_{i,j} \ge 1/\rho$ for every $(i,j) \in E'$ and (7). The second half of the sentence follows from Lemma 4.2, and the fact that every $(i,j) \in E'$ has $p_{i,j} \ge \frac{T}{\rho}$ and $x'_{i,j} \ge \frac{1}{\rho}$.

Then every machine i has at most $5\rho^2(\rho/2+1-1) = 5\rho^3/2$ neighbors in $G'^2[M]$. Notice that $G'^4[M] = (G'^2[M'])^2$. Thus $G'^4[M]$ has degree at most $5\rho^3/2 + 5\rho^3/2 \times (5\rho^3/2 - 1) = 25\rho^6/4$. Similarly, $G'^8[M]$ has maximum degree at most $(25\rho^6/4)^2 = 625\rho^{12}/16$. □

Lemma 4.6. *The machines in any connected component of $G'[M \cup J^{\text{failed}}]$ are in a same connected component of H.*

Proof. Suppose i and i' are in a same connected component in $G'[M \cup J^{\text{failed}}]$. Then there is a path $(i_0 = i, j_1, i_1, j_2, i_2, \cdots, j_o, i_o = i')$ in $G'[M \cup J^{\text{failed}}]$. Notice that every job in J^{failed} is adjacent to a marked machine. So, there is a marked machine κ_a adjacent to j_a for every $a \in [o]$. For every $a \in [o-1]$, $(\kappa_a, \kappa_{a+1}) \in G'^4[M^{\text{marked}}]$ or it is a self-loop, since κ_a-j_a-i_a-j_{a+1}-κ_{a+1} is a path of length 4 in G'. So, κ_1 and κ_o are connected in $G'^4[M^{\text{marked}}]$. Also both (i, κ_1) and (i', κ_o) are in G'^2 (if they are not self-loops). By the definition of H, i and i' are in the same connected component of H. □

Lemma 4.7. *The marked machines in any connected component of H are in a same connected component of $G'^4[M^{\text{marked}}]$.*

Proof. Notice that H is obtained from $G'^4[M^{\text{marked}}]$ by adding unmarked machines and edges between marked and unmarked machines in G'^2. This operation does not merging any two connected components of $G'^4[M^{\text{marked}}]$: Suppose we have three machines $i \in M \setminus M^{\text{marked}}$, $i', i'' \in M^{\text{marked}}$ such that $(i,i'), (i,i'') \in G'^2[M]$, then $(i', i'') \in G'^4[M^{\text{marked}}]$. That is, i' and i'' were already in the same connected component in $G'^4[M^{\text{marked}}]$. □

Lemma 4.8. *If some connected component of $G'[M \cup J^{\text{failed}}]$ contains $\rho(5\rho^3/2+1)^2$ machines, then some connected component of $G'^4[M^{\text{marked}}]$ has size at least $\rho(5\rho^3/2+1)$.*

Proof. If the condition holds, then by Lemma 4.6, some connected component of H will have $\rho(5\rho^3/2+1)^2$ machines. In the connected component, there are no edges between unmarked machines. So, the number of marked machines in the connected component is at least $\frac{\rho(5\rho^3/2+1)^2}{5\rho^3/2+1} = \rho(5\rho^3/2+1)$ by Claim 4.4 about the degree of $G'^2[M]$. Then the lemma follows from Lemma 4.7. □

Lemma 4.9. *Suppose we have a set M^* of at least $\rho(5\rho^3/2+1)$ machines such that $G'^4[M^*]$ is connected. Then there is an interesting set $M' \subseteq M^*$ of size at least ρ.*

Proof. Indeed, let M' be any *maximal* independent set of $G'^2[M^*]$. First, the size of M' is at least $\frac{|M^*|}{5\rho^3/2+1} \geq \rho$ since every machine $i \in M^*$ has at most $5\rho^3/2$ neighbors in $G'^2[M^*]$ by Claim 4.4. It remains to show that $G'^8[M']$ is connected. Assume towards the contradiction that this is not the case. Then M' can be partitioned into two non-empty sets M'^1 and M'^2 such that there are no edges between M'^1 and M'^2 in G'^8.

We focus on the graph $G'^4[M^*]$, which, by the condition of the lemma, is connected. For every edge (i, i') in $G'^4[M^*]$, we define its length to be the minimum number of edges in a path connecting i and i' in G'. Notice that the length of (i, i') is either 2 or 4. Then we focus on the shortest path between M'^1 and M'^2 in $G'^4[M^*]$. The length of the shortest path is at least 10. Assume the path connects $i_1 \in M'_1$ to $i_2 \in M'_2$. If the first edge on the path has length 4, then the second machine on the path could have been added to M'. If the last edge on the path has length 4, then the second-to-last machine on the path could have been added to M'. By the maximality of M', they can not happen. So, both the first and last edges of the path have length 2. Then the path contains at least 4 edges. Therefore, the middle machine on the path could be added to M', leading to a contradiction. Thus, $G'^8[M']$ is connected. □

Lemma 4.10. *With probability at least $1 - \frac{1}{4m}$, every interesting set M' of size ρ contains an unmarked machine.*

Proof. Focus on the graph $G'^8[M]$, and let d denote the maximum degree of the graph and thus $d \leq 625\rho^{12}/16$ by Claim 4.4.

We show that there are at most $\binom{2(\rho-1)}{\rho-1} m(d-1)^{\rho-1} \leq 2^{2\rho} \cdot md^\rho = m \cdot (4d)^\rho$ different subsets $M' \subseteq M$ with $|M'| = \rho$ and $G'^8[M']$ being connected. To see this, we can use a spanning tree of $G'^8[M']$ to represent such a M'. To describe the spanning tree, we construct a traveling-salesman tour that starts from an arbitrary vertex in M', and contains each edge in the spanning tree exactly twice. Each edge in the tour is an either a backward edge or a forward edge, and there are at most $\binom{2(\rho-1)}{\rho-1}$ possibilities for splitting the $2(\rho-1)$ edges into $\rho-1$ forward edges and $\rho-1$ backward edges. Thus to describe the tour, we specify the starting vertex, the split, and the actual forward edges. There are m possibilities for the starting vertex, and at most $d-1$ possibilities for each of the forward edge. The bound then follows. It implies that the number of interesting subsets M' of size ρ is at most $m \cdot (4d)^\rho$.

For every interesting subset $M' \subseteq M$ of size ρ, the probability that all vertices in M' are marked is at most $\left(\frac{1}{700\rho^{12}}\right)^\rho$ due to Lemma 4.3 and that machines in $M' \subseteq M$ do not share neighbors in G'. Using union bound we obtain the following. With probability at least $1 - m \cdot (4d)^\rho \cdot \left(\frac{1}{700\rho^{12}}\right)^\rho \geq 1 - m \cdot \left(\frac{4 \cdot 625\rho^{12}}{16 \cdot 700\rho^{12}}\right)^\rho = 1 - m \cdot \left(\frac{25}{112}\right)^\rho \geq 1 - \frac{1}{4m}$, every interesting M' of size ρ contain at least one unmarked machine. □

C. Online Algorithm for Unrelated Machine Load Balancing with Prediction

In this section, we first prove Corollary 2.5 about the prediction for the unrelated machine load balancing problem. Then we show how to handle the errors in the prediction. The corollary is repeated below:

Corollary 2.5. *Given an unrelated machine load balancing instance, there are $\beta, w \in \text{powers}^M_{1+\epsilon,K}$ for some $K = O\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ such that $x^{(\beta,w)}$ is $(1+\epsilon)^4$-approximate to* (P-LP).

C.1. Proof of Corollary 2.5

For convenience we call the given unrelated machine load balance instance $\mathcal{I}$. Let $K = O\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ that satisfies the requirements in both Lemma 2.3 and Theorem 2.4.

Let $\beta \in \text{powers}^M_{1+\epsilon,K}$, $\alpha_j = \min_{i \in M_j} p_{i,j}\beta_i, \forall j$ and x be the objects satisfying the property of Theorem 2.4. We define a load balancing instance $\mathcal{I}'$ in the $\mathsf{Q}|\text{restricted}$ setting as follows. We set $p'_j = \alpha_j$ for every $j \in J$, and $s'_i = \beta_i$ for every $i \in M$. We set $p'_{i,j} = \frac{p'_j}{s'_i} = \frac{\alpha_j}{\beta_i}$ if $p_{i,j}\beta_i \leq (1+\epsilon)\alpha_j$, and $p'_{i,j} = \infty$ otherwise. Then the instance $\mathcal{I}'$ is defined by $(p'_{i,j})_{i \in M, j \in J}$. Let $E' = \{(i,j) \in E : p_{i,j} \neq \infty\}$ and $M'_j = \{i : (i,j) \in E'\}, \forall j \in J$. Then $x|_{E'}$ is a valid solution to (P-LP) for $\mathcal{I}'$. As $p'_{i,j} \neq \infty$ implies $p'_{i,j} = \frac{\alpha_j}{\beta_i} \in \left[\frac{p_{i,j}}{1+\epsilon}, p_{i,j}\right]$, the value of x to (P-LP) w.r.t $\mathcal{I}'$ is at most T^*.

So we can apply Lemma 2.3 to the instance $\mathcal{I}'$. There is a vector $w \in \text{powers}^M_{1+\epsilon,K}$ such that $x^{(w)}$ has value at most $(1+\epsilon)^3 T^*$ to (P-LP) w.r.t $\mathcal{I}'$. Notice that $x^{(w)}$ is defined w.r.t $\mathcal{I}'$. That is, for every $(i,j) \in E'$, we have $x^{(w)}_{i,j} = \frac{w_i}{\sum_{i' \in M'_j} w_{i'}}$. As $E' \subseteq E$ and for every $(i,j) \in E'$, we have $p_{i,j} \leq (1+\epsilon)p_{i,j}$, $x^{(w)}$ has value at most $(1+\epsilon)(1+\epsilon)^3 T^* = (1+\epsilon)^4 T^*$ to (P-LP) w.r.t the original instance $\mathcal{I}$, when extended to the domain E by adding 0's. This is precisely the $x^{(\beta,w)}$ in Corollary 2.5.

C.2. Handle Errors in the Prediction

When we are given the (β, w) in Corollary 2.5, then our algorithm can construct the fractional solution $x^{(\beta,w)}$ online, which can be passed to the $O\left(\frac{\log\log m}{\log\log\log m}\right)$-competitive randomized rounding algorithm.

If the prediction has an error, then intuitively we can make the competitive ratio deteriorate smoothly as the error grows. Since our prediction contains two vectors β and w, it is natural to measure the error of each vector separately. Recall that $K = O\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ is the number satisfying the requirements of both Lemma 2.3 and Theorem 2.4. Let $\beta^*, w^* \in \text{powers}^M_{1+\epsilon,K}$ be the perfect β, w satisfying the statement in Lemma 2.3.

There are two issues to address. First, how do we measure the error of a dual (weight) vector? For convenience we focus on the weight vector part. We could simply define the error of a prediction w as the multiplicative difference between w and w^*, which is $\max_{i\in M}\frac{w_i}{w_i^*}\max_{i\in M}\frac{w_i^*}{w_i}$. This is indeed the metric used by Lattanzi et al. (2020). However the metric has a drawback: If there are two very different vectors which both satisfy the statement of Theorem 2.4, then one of them will have large error, depending on which vector we choose as w^*. The issue becomes more severe in our case as the coordinates in w are in $\text{powers}_{1+\epsilon,K}$, which has an exponential multiplicative gap between the maximum and minimum number.

We believe a more natural metric to use is the quality of the vector w, since this directly determines how good w is. Moreover, the definition does not depend on the choice of the truth vector w^*. Moreover it is consistent with the goals of many machine learning tasks. For example, in PAC learning, we measure the quality of a hypothesis by the fraction of errors it produces, rather than the difference between its parameters and the true ones.

With this guildline, we define ρ-good dual vectors and η-good weight vectors as follows:

Definition C.1. *Assume we are given an unrelated machine load balancing instance. We say a vector $\beta \in \text{powers}(1+\epsilon, K)^M$ is a ρ-good dual vector for some $\rho \geq 1$, if there exists an optimum solution $x \in [0,1]^E$ to* (P-LP) *such that $x_{i,j} > 0$ implies $p_{i,j}\beta_i \leq \rho \min_{i'\in M_j} p_{i',j}\beta_j$.*

Thus, Theorem 2.4 says there is a $(1+\epsilon)$-good dual vector β.

Similarly, we define what is a η-good weight vector:

Definition C.2. *Given a load-balancing instance in the* $\mathsf{Q}|\text{restricted}$ *setting, we say a vector $w \in \text{powers}^M_{1+\epsilon,K}$ is an η-good weight vector for some $\eta \geq 1$, if $x^{(w)}$ is an η-approximate solution to* (P-LP).

So, Lemma 2.3 guarantees the existence of a $(1+\epsilon)^3$-good weight vector w.

We remark that an η-good weight vector may not be η-multiplicative factor distance away from any 1-good weight vector. For example consider the identical machine case where all jobs can be assigned to all machines. Then the uniform vector $w^* = (1, 1\cdots, 1)$ is 1-good. The vector w with $\left(1-\frac{1}{\eta}\right)m$ coordinates being 1, and the other $\frac{m}{\eta}$ coordinates being $(1+\epsilon)^K$ is η-good. But the vector w is exponentially far away from w^* in terms of the multiplicative distance. As a result, the $O(\log\eta)$ dependence in the result in Lattanzi et al. does not hold for our new metric. Instead, we only obtain a dependence of $O(\eta)$.

The second issue comes from the two-step nature of our prediction. The instance in the $\mathsf{Q}|\text{restricted}$ setting we obtained from the reduction depends on T and the β vector in the prediction. So, the definition of the goodness of the weight vector w should be w.r.t this instance, instead of the instance when we have $\beta = \beta^*$.

With the two issues addressed, we can now argue about the dependence of the competitive ratio on the error parameters. Let $\mathcal{I}$ be the given load balancing instance in the unrelated machine setting and the prediction we have is (β, w). Then, we assume the dual vector $\beta \in \text{powers}^M_{1+\epsilon,K}$ is ρ-good w.r.t $\mathcal{I}$ for some $\rho \geq 1$. We define the instance $\mathcal{I}'$ in $\mathsf{Q}|\text{restricted}$ setting as before, but using ρ to replace $1+\epsilon$. Let $\alpha_j = \min_{i\in M_j} p_{i,j}\beta_i$ for every $j \in J$. Let $M'_j = \{i \in M_j : p_{i,j}\beta_i \leq \rho\alpha_j\}$. We set $p'_j = \alpha_j$ for every $j \in J$, and $s'_i = \beta_i$ for every $i \in M$. We set $p'_{i,j} = \frac{p'_j}{s'_i} = \frac{\alpha_j}{\beta_i}$ if $i \in M'_j$, and $p'_{i,j} = \infty$ otherwise. The instance $\mathcal{I}'$ defined by $(p'_{i,j})_{i\in M, j\in J}$ is clearly an instance in the $\mathsf{Q}|\text{restricted}$ setting.

The optimum value of (P-LP) w.r.t $\mathcal{I}$ is at most T^*. By the ρ-goodness of β, there is a solution x to (P-LP) of value at most T^* w.r.t $\mathcal{I}$ so that $x_{i,j} > 0$ for some $i \in M_j$ implies $\alpha_j \leq p_{i,j}\beta_i \leq \rho\alpha_j$, which is $p'_{i,j} = \frac{\alpha_j}{\beta_i} \in \left[\frac{p_{i,j}}{\rho}, p_{i,j}\right]$. Therefore,

(i) The value of x to (P-LP) w.r.t $\mathcal{I}'$ is at most T^* (when restricted to the allowed pairs (i,j) in $\mathcal{I}'$).
(ii) Any solution to (P-LP) w.r.t $\mathcal{I}'$ of value at most T' is has value at most $\rho T'$ w.r.t $\mathcal{I}$ (after we extend the domain to E).

Now we assume the weight vector $w \in \text{powers}^M_{1+\epsilon,K}$ given in the prediction is η-good w.r.t $\mathcal{I}'$, for some $\eta \geq 1$. Given this w, the fractional solution $x^{(w)}$ (defined w.r.t $\mathcal{I}'$) has value at most ηT^*. Thus, $x^{(w)}$ has value at most $\rho\eta T^*$ to (P-LP) w.r.t $\mathcal{I}$. Also, by Assumption 3, all the pairs $(i,j) \in E$ has $p_{i,j} \leq T^*$. So we have $\text{mspn}_{\mathcal{I}}(x^{(w)}) \leq \rho\eta T^*$, where the subscript $\mathcal{I}$ indicates the instance we are considering is the original instance $\mathcal{I}$. Then our online algorithm in Section 4 can construct an assignment with makespan at most $O\left(\frac{\rho\eta\log\log m}{\log\log\log m}\right)\cdot T^*$ with high probability.

We remark that the algorithms can guarantee the worst case competitive ratio of $O(\log m)$: Once the makespan of our schedule is about to exceed $(\log m)T^*$, we simply switch

to the $O(\log m)$-competitive online algorithm that does not use the prediction.

We need to know the values of ρ to define the instance $\mathcal{I}'$ (we do not need to know the value of η). The assumption can be removed if we can query an oracle about a weight vector in an adaptive way. We only give a high-level sketch on how we can do this. Initially, we ask the oracle to give a dual-vector β, whose goodness w.r.t $\mathcal{I}$ is not known. We break the algorithm into phases, where each phase corresponds to a guessed goodness parameter ρ of the vector β, where initially we have $\rho = 1+\epsilon$. At the beginning of a phase, we define $\mathcal{I}'$ as above by assuming β is ρ-good w.r.t $\mathcal{I}$. Then we ask the oracle to give a weight vector w for this instance $\mathcal{I}'$. Within each phase, we run the online algorithm as described above. Once we find that the current β is not ρ-good w.r.t instance $\mathcal{I}$ (this can be checked efficiently), we double ρ and start a new phase. Suppose the β at the beginning is ρ-good, and all weight vectors w returned by the oracle are always η-good, then it is not hard to show that the algorithm is $O(\frac{\rho\eta \log\log m}{\log\log\log m})$-competitive.

D. Learnability of Prediction

In this section, we first describe the model introduced by Lavastida et al. (2020) on the learnability of a prediction. Then we show that under the model our prediction (β, w) can be learned.

For the sake of convenience, we define $\mathbf{p}_j := (p_{i,j})_{i\in M}$ and $\mathbf{P} \in (0,\infty]^{M\times J}$ to denote the matrix $(p_{i,j})_{i\in M, j\in J}$. Then the whole instance is completely defined by $\mathbf{P}$. There is a distribution $\mathcal{D}_j$ of $\mathbf{p}_j$'s, for every $j \in J$. Let $\mathcal{D} = \prod_{j\in J} \mathcal{D}_j$ be the product distribution of all $\mathcal{D}_j$'s. We assume the instance $\mathbf{P}$ we need to solve is selected randomly from $\mathcal{D}$; that is, for each $j \in J$, $\mathbf{p}_j$ is chosen randomly and independently from $\mathcal{D}_j$. For notational convenience, we assume the distribution $\mathcal{D}$ is discrete.

Let $T(\mathbf{P})$ be the optimum fractional makespan for the instance $\mathbf{P}$. That is $T(\mathbf{P})$ is the smallest $\mathrm{mspn}_{\mathbf{P}}(x)$ over all fractional assignment x, where $\mathrm{mspn}_{\mathbf{P}}(x)$ is $\mathrm{mspn}(x)$ when the underlying instance is $\mathbf{P}$. Let $T := \mathbb{E}_{\mathbf{P}\sim\mathcal{D}} T(\mathbf{P})$ be the average of $T(\mathbf{P})$ over all instances from $\mathcal{D}$. As in Lavastida et al., we make the following mild assumption:

Assumption 4. *For every $i \in M, j \in J$, for every $\mathbf{p}_j$ in the support of $\mathcal{D}_j$, we have $p_{i,j} \le \frac{T}{\gamma}$ or $p_{i,j} = \infty$, for some big enough $\gamma = \Theta(\frac{\log m}{\epsilon^2})$.*

Since the fractional assignment $x^{(\beta,w)}$ depends on the instance $\mathbf{P}$, we shall use $x^{(\mathbf{P},\beta,w)}$ to denote the $x^{(\beta,w)}$ when the instance is $\mathbf{P}$.

The main theorem regarding the learnability of the pair (β, w) is the following:

Theorem D.1. *There is a learning algorithm that samples $O\left(\frac{m}{\log m}\log\frac{m}{\epsilon}\right)$ independent instances from $\mathcal{D}$, and outputs two vectors $\beta, w \in \mathrm{powers}(1+\epsilon, K)$ for some $K = \Theta\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ such that the following event happens with probability at least $1 - \frac{1}{K^m}$. $x^{(\mathbf{P},\beta,w)}$ has makespan at most $(1+O(\epsilon))T$ with high probability over instances $\mathbf{P} \sim \mathcal{D}$.*

Throughout the section, we let $K = \Theta\left(\frac{m}{\epsilon}\log\frac{m}{\epsilon}\right)$ be large enough. The analysis contains two parts. First, we show that there is a good pair (β^*, w^*), by considering the "average instance" of the distribution. Second, there is a learning algorithm that outputs an approximately optimum (β, w) with $\mathrm{poly}(m, \frac{1}{\epsilon})$ number of samples. The analysis is similar to that of PAC learning, where we use concentration and union bounds to show that w.h.p, for every potential pair (β, w), the quality of a pair (β, w) over a random instance, is approximately preserved by its quality over the sampled instances.

As we argued, the value $x_{i,j}^{(\mathbf{P},\beta,w)}$ only depends on $\mathbf{p}_j, \beta$ and w. That is, it is independent of $\mathbf{p}_{j'}$'s for any other job j'. For an instance $\mathbf{P} = (\mathbf{p}_j)_{j\in J}$, $\beta, w \in \mathrm{powers}_{1+\epsilon,K}^{M}$, we define $F_{\mathbf{P}}(\beta, w, i) := \sum_{j\in J} x_{i,j}^{(\mathbf{P},\beta,w)} p_{i,j}$ to be the fractional load of machine i in the solution $x^{(\mathbf{P},\beta,w)}$, where we assume $0\times\infty = 0$. Let $F_{\mathbf{P}}(\beta, w) := \max_{i\in M} F_{\mathbf{P}}(\beta, w, i)$ be the value of the solution to (P-LP).

In order to show the existence of a good pair (β^*, w^*), we shall consider a combination of all instances in the distribution $\mathcal{D}$. We make the following definition.

Definition D.2. *For L processing time matrices $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \cdots, \mathbf{P}^{(L)}$, we define $\mathbf{P}^{(1)} \oplus \mathbf{P}^{(2)} \oplus \cdots \oplus \mathbf{P}^{(L)}$ to be the following instance defined by the nL jobs. For every $\ell \in [L]$ and $j \in J$, we have a the job $j^{(\ell)}$ with processing times defined by $\mathbf{p}_j^{(\ell)}$, the column of the matrix of $\mathbf{P}^{(\ell)}$ correspondent to j.*

So, the instance $\mathbf{P}^{(1)} \oplus \mathbf{P}^{(2)} \oplus \cdots \oplus \mathbf{P}^{(L)}$ is defined by the $(m \times Ln)$-size matrix obtained by concatenating the L matrices of size $m \times n$.

The following observation is immediate, since we can take the concatenation of L optimum fractional solutions for the L instances:

Observation D.3. $T(\mathbf{P}^{(1)} \oplus \mathbf{P}^{(2)} \oplus \cdots \oplus \mathbf{P}^{(L)}) \le T(\mathbf{P}^{(1)}) + T(\mathbf{P}^{(2)}) + \cdots + T(\mathbf{P}^{(L)})$.

Now with the observation, we can prove the following lemma, by considering the combination of all instances in $\mathcal{D}$, scaled by their respective probabilities.

Lemma D.4. *There exist $\beta^*, w^* \in \mathrm{powers}(1+\epsilon, K)^M$,*

such that for every $i \in M$, we have

$$\mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta^*, w^*, i) \le (1+\epsilon)^4 T.$$

Proof. Consider the instance $\mathbb{P} := \bigoplus_{\mathbf{P}\in\mathcal{D}} \Pr_{\mathcal{D}}[\mathbf{P}] \cdot \mathbf{P}$, where $\Pr_{\mathcal{D}}[\mathbf{P}]$ is the probability mass of $\mathbf{P}$ in $\mathcal{D}$, and $\Pr_{\mathcal{D}}[\mathbf{P}] \cdot \mathbf{P}$ is the matrix $\mathbf{P}$ multiplied by $\Pr_{\mathcal{D}}[\mathbf{P}]$. So, the instance is obtained by concatenating all instances in the distribution $\mathcal{D}$, scaled by their respective probability masses.

Applying Observation D.3, we have

$$T(\mathbb{P}) \le \sum_{\mathbf{P}} \Pr_{\mathcal{D}}[\mathbf{P}] \cdot T(\mathbf{P}) = \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}}[T(\mathbf{P})] = T.$$

We can then apply Corollary 2.5 to the combined instance to show that there exists some β^*, w^* such that for every $i \in M$, we have

$$\sum_j x_{i,j}^{(\mathbb{P},\beta^*,w^*)} p_{i,j} \le (1+\epsilon)^4 T(\mathbb{P}) = (1+\epsilon)^4 T,$$

where j is over all jobs in $\mathbb{P}$. Notice that $x_{i,j}^{(\mathbb{P},\beta^*,w^*)}$ for a job j only depends on the processing time vector for the job j, which is included in the instance $\mathbf{P} \in \mathcal{D}$ that j belongs to. Therefore, the left side of the above inequality is exactly

$$\begin{aligned}\sum_{\mathbf{P}} \sum_{j\in J} x_{i,j}^{(\mathbf{P},\beta^*,w^*)} \Pr_{\mathcal{D}}[\mathbf{P}] p_{i,j} &= \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} \sum_{j\in J} x_{i,j}^{(\mathbf{P},\beta^*,w^*)} p_{i,j} \\ &= \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta^*, w^*, i).\end{aligned}$$

This finishes the proof of the lemma. □

Since we need to apply Chernoff bound multiple times, it is convenient to introduce the following notation:

Definition D.5. *For any real numbers $A, B, \epsilon, C \ge 0$, we use $A \approx_{\epsilon,C} B$ to denote $|A - B| \le \epsilon \cdot \max\{B, C\}$.*

Lemma D.6. *For any $\beta, w \in \text{powers}^M_{1+\epsilon,K}$, with high probability over $\mathbf{P} \sim \mathcal{D}$, we have*

$$\forall i \in M : F_{\mathbf{P}}(\beta, w, i) \approx_{\epsilon,T} \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i).$$

Proof. $F_{\mathbf{P}}(\beta, w, i)$ is the sum of n independent random numbers taking values in $[0, \frac{T}{\gamma}]$, one for each $j \in J$. Notice that $\gamma = \Theta(\frac{\log m}{\epsilon^2})$ is sufficiently large. We apply Chernoff bound (Theorem E.1) over the summation correspondent to $\frac{\gamma}{T} F_{\mathbf{P}}(\beta, w, i)$. Let $\mu := \mathbb{E}_{\mathbf{P}\sim\mathcal{D}}\left[\frac{\gamma}{T} F_{\mathbf{P}}(\beta, w, i)\right]$, $U = \max\{\mu, \gamma\}$ and $\delta = \epsilon$. Then applying the bound gives us that with probability at most $2e^{-\frac{\delta^2 U}{3}} \le 2e^{-\frac{\epsilon^2\gamma}{3}} \le 2e^{-\Theta(\log m)}$, we have $\frac{\gamma}{T} F_{\mathbf{P}}(\beta, w, i) - \mu \in [-\delta U, \delta U]$. This is equivalent to

$$\frac{\gamma}{T} F_{\mathbf{P}}(\beta, w, i) \approx_{\epsilon,\gamma} \mu.$$

Scaling by $\frac{T}{\gamma}$, the formula becomes

$$F_{\mathbf{P}}(\beta, w, i) \approx_{\epsilon,T} \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i).$$

The lemma holds from that γ is big enough and the union bound over all $i \in M$. □

Now we can describe the learning algorithm. We sample $H = O\left(\frac{m}{\log m} \log \frac{m}{\epsilon}\right)$ instances $\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_H$ independently and randomly from $\mathcal{D}$, where H is large enough. We output the (β, w) with the smallest $\max_{i\in M} \frac{1}{H} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i)$.

Lemma D.7. *With probability at least $1 - \frac{1}{K^m}$, the following event happens. For every pair $\beta, w \in \text{powers}(1+\epsilon, K)^M$ and $i \in M$, we have*

$$\frac{1}{H} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i) \approx_{\epsilon,T} \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i).$$

Proof. The term $\frac{\gamma}{T} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i)$ is the sum of nH independent random variables in the range $[0, 1]$. Its expectation is $\mu := \frac{H\gamma}{T} \mathbb{E}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i)$. We then apply Chernoff bound the sum with $U = \max\{\mu, H\gamma\}$ and $\delta = \epsilon$. With probability at moat $2e^{-\frac{\epsilon^2 U}{3}} \le 2e^{-\frac{\epsilon^2 H\gamma}{3}}$, we have

$$\frac{\gamma}{T} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i) \approx_{\epsilon, H\gamma} \mu.$$

Scaling by a factor of $\frac{T}{H\gamma}$, the above formula becomes

$$\frac{1}{H} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i) \approx_{\epsilon,T} \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i).$$

To make the probability to be at most $\frac{1}{m(K+1)^{3m}}$, it suffices to set $H = \frac{O(m \log K)}{\gamma\epsilon^2} = O\left(\frac{m}{\log m} \log \frac{m}{\epsilon}\right)$. Applying union bound over all $\beta, w \in \text{powers}^M_{1+\epsilon,K}$ and $i \in M$ finishes the proof. □

Now assume the event in Lemma D.7 happens. Then by Lemma D.4, we have $\max_{i\in M} \frac{1}{H} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta^*, w^*, i) \le (1+\epsilon)^5 T$. Then the algorithm will output a pair (β, w) satisfying $\max_{i\in M} \frac{1}{H} \sum_{h=1}^H F_{\mathbf{P}_h}(\beta, w, i) \le (1+\epsilon)^5 T$. Therefore, we have for every $i \in M$, $\mathbb{E}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i) \le \frac{(1+\epsilon)^5}{1-\epsilon} T = (1 + O(\epsilon))T$.

Then we apply Lemma D.6 to this (β, w). We have that with high probability over $\mathbf{P} \sim \mathcal{D}$, for every $i \in M$, the following holds:

$$\begin{aligned}F_{\mathbf{P}}(\beta, w, i) &\le \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i) \\ &\quad + \epsilon \max\{T, \mathop{\mathbb{E}}_{\mathbf{P}\sim\mathcal{D}} F_{\mathbf{P}}(\beta, w, i)\} \\ &\le (1 + O(\epsilon))T.\end{aligned}$$

That is precisely $F_{\mathbf{P}}(\beta, w) \leq (1 + O(\epsilon))T$. This finishes the proof of Theorem D.1.

E. Concentration Bounds

Theorem E.1 (Variant of Chernoff Bound)**.** *Let $X_1, X_2, \cdots, X_n$ be independent random variables taking values in $[0, 1]$. Let $X = \sum_{i=1}^{n} X_i, \mu = \mathbb{E}[X]$ and $U \geq \mu$. For every $\delta > 0$, we have*

$$\begin{aligned}\Pr\left[X > (1+\delta)U\right] &\leq \Pr\left[X > \mu + \delta U\right] \\ &< \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{U} \leq e^{-\frac{\delta^2 U}{2+\delta}},\end{aligned}$$

and

$$\Pr\left[X < \mu - \delta U\right] < e^{-\frac{\delta^2 U}{2}}.$$