Supplementary Material for the Paper "Assignments for Congestion-Averse Agents: Seeking Competitive and Envy-Free Solutions"

A Additional Material for Section 2

A.1 Proof of Lemma 1

Lemma 1 ($\star$). *(1) CP implies EF, but the converse does not hold.*
(2) CP implies NS, but the converse does not hold.
(3) NS implies TG and NW, but the converse does not hold.
(4) EF is incomparable to NS, to TG, and to NW, respectively; TG is incomparable to NW.

Proof. (1) Clearly, CP implies EF by definition. Now, to show that the converse does not hold, let us consider Example 1. Clearly assigning every agent to post a_1 is EF, but it is not CP since it is not TG.

(2) As already mentioned, the implication has been discussed by Bogomolnaia and Moulin [6] already. For the sake of completeness, we provide a proof by showing the contra-positive. Assume that assignment Π is not NS and let there be an agent $v \in V$ and a post $a \in A$ such that v prefers $(a, |\Pi(a)| + 1)$ to $(a^*, |\Pi(a^*)|)$ where a^* denotes the post that v is assigned to. If a is empty, then clearly, Π is wasteful and hence not CP. If a is non-empty, then since every agent is averse against congestions, we infer that v prefers $(a, |\Pi(a)|)$ to $(a^*, |\Pi(a^*)|)$, and hence not CP either.

Now, to show that the converse does not hold, let us consider Example 1 again. Π_1 is NS, but not CP.

(3) That NS implies NW follows directly from definition. That "NS implies TG" has also been shown by Bogomolnaia and Moulin [6]. We show the contra-positive for the sake of completeness. Let Π be an assignment that is not TG, and let $v \in V$ be an agent that is assigned to post a^* such that $(a^*, |\Pi(a^*)|)$ is *not* among his top-$|V|$ choices. Let $X = \{(a, d) \mid (a, d) \succ_v (a^*, |\Pi(a^*)|)\}$. Then, $|X| \geq |V|$. For each $a \in A$, let $\delta(a)$ denote the largest congestion such that $(a, \delta(a)) \in X$, i.e., $\delta(a) = \max_{(a,d)\in X}\{d\}$. Then,$\sum_{a\in A} \delta(a) = |X| \geq |V| = \sum_{a\in A} |\Pi(a)|$. By assumption, we have that $\delta(a^*) < |\Pi(a^*)|$. This implies that there must exist a post $a \in A \setminus \{a^*\}$ such that $\delta(a) > |\Pi(a)|$. By definition, we infer that $(a, |\Pi(a)| + 1) \in X$, and hence $(a, |\Pi(a)| + 1) \succ_v (a^*, |\Pi(a^*)|)$, witnessing that Π is not NS.

It is quite easy to come up with a TG and NW assignment which is not NS. For the sake of completeness, let us consider the following example.

$$\begin{aligned} v_1\colon\ & (a_1, 1) \succ (a_1, 2) \succ (a_2, 1) \succ \cdots, \\ v_2\colon\ & (a_1, 1) \succ (a_2, 1) \succ (a_1, 2) \succ \cdots, \\ v_3\colon\ & (a_3, 1) \succ (a_3, 2) \succ (a_3, 3) \succ \cdots. \end{aligned} \qquad \Pi_3\colon \begin{array}{c|c|c} a_1 & a_2 & a_3 \\ \hline v_2 & v_1 & v_3 \end{array} \qquad \Pi_4\colon \begin{array}{c|c|c} a_1 & a_2 & a_3 \\ \hline v_1, v_2 & & v_3 \end{array}$$

Π_3 is clearly TG and NW. It is not NS however, since v_1 prefers $(a_1, 2)$ to $(a_2, 1)$.

(4) Let us consider Example 1. Assigning every agent to the same post is clearly EF, but not TG and not NW. Hence, it is not NS. As already argued in Example 1, Π_1 is NS, TG, and NW, but not EF. Assigning v_1 and v_2 to a_2, and v_3 to a_1 is NW, but not TG: For v_1, tuple $(a_2, 2)$ is *not* his top 3 choices.

Now, let us consider the example from previous item. Π_4 is TG but not NW. □

By definition, we observe the following:

Observation 2. *For an arbitrary tie-breaking rule, $\sum_{a\in A} \lambda(v, a) = |V|$ holds for every $v \in V$.*

A.2 Proof of Lemma 2

Lemma 2 ($\star$). *Given a congestion vector $\vec{s}$ with $\sum_{a\in A} \vec{s}[a] = |V|$, in polynomial-time one can determine the smallest number t of unsatisfied agents among all assignments whose congestion vectors equal $\vec{s}$; the corresponding assignment can found in polynomial time.*

Proof. The idea is to iterate over all possible number $t \in \{0, 1, \ldots, |V|\}$ and check whether there exists an assignment with congestion vector $\vec{s}$ and exactly t unsatisfied agents. The later problem can be solved via determining whether a perfect $\vec{b}$-matching exists, which can be done in polynomial time [29, Chapter 12].

Let $(A, V, (\succeq)_{v\in V})$ be an instance of CONGESTED ASSIGNMENT. To check whether there exists an assignment with congestion vector $\vec{s}$ and exactly t unsatisfied agents, we construct a bipartite graph G on two disjoints X and Y with $X = A \cup \{a_0\}$ and $Y = V \cup \{w_1, \ldots, w_t\}$, where the w_z's are the dummy *agent-vertices* and a_0 is a dummy *post-vertex*.

We add an edge between every original post a_j and every dummy agent-vertex w_z, and an edge between the dummy post-vertex a_0 and every original agent v_i. We also add an edge between every original post and original agent, but will delete some according to the congestions as follows:

For each original agent v and each two original posts a and a', if v strictly prefers $(a, \vec{s}[a])$ to $(a', \vec{s}[a'])$, then we delete the edge $\{a', v\}$. This is because if v would be satisfied, he will never be assigned to a' since he will envy some agent that is assigned to a.

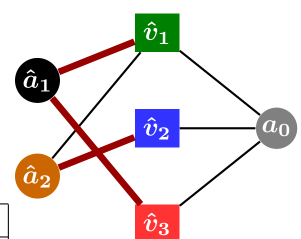
This completes the construction of the graph G. We check whether there exists a *perfect* $\vec{b}$-matching[3] for G where $\vec{b}[a_0] = t$, $\vec{b}[a_j] = \vec{s}[a_j]$ for all $a_j \in A$, and $\vec{b}[x] = 1$ for all $x \in X$, and answer yes if such matching exists, and no otherwise.

For the correctness, it is straightforward that if Π is an assignment with congestion vector $\vec{s}$ and exactly t unsatisfied agents V', then the following matching is a perfect $\vec{b}$ matching: $M[v_i] = a_j$ if $v_i \in V \setminus V'$, and $M[v_i] = a_0$ otherwise. Finally, for each original post a_j, if it is assigned $\hat{n}$ unsatisfied agents, then we pick $\hat{n}$ distinct dummy agent-vertices and match them to a_j.

If M is a perfect $\vec{b}$-matching for G, then a_0 is matched with exactly t original agents who will be the unsatisfied agents by assigning them to the post to replace the dummy agents one-by-one.

Since checking the existence of a perfect $\vec{b}$-matching can be done in polynomial time by reducing to finding a perfect matching, the whole approach can be done in polynomial time as well. This completes the proof.

For an illustration, consider the first instance in Example 1. Let the congestion vector be $\vec{s} = (2, 1)$ and the number of unsatisfied agents be $t = 0$. The following bipartite graph G has a perfect $\vec{b}$-matching, indicated by the red lines.



$\vec{b}:$	a_0	a_1	a_2	v_1	v_2	v_3
	0	2	1	1	1	1

Indeed, the corresponding $\vec{b}$-matching yields a CP assignment which is Π_2 and it has only satisfied agents. □

B Additional Material for Section 3

C Additional Material for Section 3.1

C.1 Example of Construction 1

Let us consider the four agents with the following preference lists.

$$
\begin{aligned}
v_1 :& \quad (a_1, 1) \succ (a_2, 1) \succ (a_3, 1) \sim (a_1, 2) \sim (a_2, 2) \succ \cdots, \\
v_2, v_3 :& \quad (a_2, 1) \succ (a_1, 1) \succ (a_3, 1) \sim (a_1, 2) \sim (a_2, 2) \succ \cdots, \\
v_4 :& \quad (a_1, 1) \sim (a_2, 1) \succ (a_1, 2) \sim (a_2, 2) \succ (a_3, 1) \succ \cdots,
\end{aligned}
$$

[3] A $\vec{b}$-matching M is *perfect* if $\sum_{e\in M:\, u\in e} 1 = \vec{b}[u]$ holds for all vertices u.

One can observe that if every post is filled, then a_1 or a_2 will have congestion one. If a_1 has congestion one, then v_1 has to be assigned to a_1 alone and v_4 to a_2 alone, leaving v_2 and v_3 to be envious. If a_2 has congestion one, then v_2 or v_3 will be envious. One can verify that assigning any two agents to a_1 and the remaining two to a_2 is competitive, leaving a_3 empty.

Now, let us "guess" that the number of empty post is $k = 1$. For $k = 1$, we augment the instance with one dummy agent u_1 and two auxiliary agents p_1 and p_2, and two dummy posts b_1 and b_2. Their preference lists are as follows:

$$\begin{aligned} u_1\colon\ & (a_1, 1) \sim (a_2, 1) \sim (a_3, 1) \succ (b_1, 1) \succ (b_1, 2) \succ (b_1, 3);\\ p_1\colon\ & (b_1, 1) \succ (b_2, 1) \succ (b_2, 2) \succ \cdots \succ (b_2, 5);\\ p_2\colon\ & (b_2, 1) \succ (b_1, 1) \succ (b_1, 2) \succ \cdots \succ (b_1, 5). \end{aligned}$$

One can verify that in the original instance, every CP assignment will leave a_3 empty, and in the augmented instance, every CP assignment will assign the dummy agent u_1 to a_3 alone. The correctness is given by Lemma 3.

C.2 Proof of Observation 1

Observation 1 ($\star$)**.** *Let $I_k = (A^*, V^*, (\succeq_v^*)_{v \in V^*})$ denote the instance created by Construction 1 with $A^* = A \cup \{b_1, b_2\}$ and $V^* = V \cup \{u_i \mid i \in [k]\} \cup \{p_1, p_2\}$. Every CP assignment of I_k (if it exists) satisfies the following: (1) p_1 is assigned to b_1 alone, and p_2 to b_2 alone. (2) Every dummy u_z with $1 \le z \le k$ is assigned to some $a_j \in A$ alone. (3) Every original $v_i \in V$ is assigned to some original post.*

Proof. Let Π be a CP assignment of I_k with $\Pi = (S_a)_{a \in A^*}$. To show the first part of statement (1), we observe that if p_1 is assigned to b_1, then $|S_{b_1}| = 1$, due to the maximum congestion of p_1 towards b_1. Thus, it suffices to show that p_1 is indeed assigned to b_1. Suppose, for the sake of contradiction, that p_1 was assigned to b_2 instead; note that he will not be assigned to any original post a_j due to his maximum congestion towards a_j. Then, by the maximum congestion of p_2 towards b_2, agent p_2 could not be assigned to b_2. No original agent could be assigned to b_2 either, due to his maximum congestion towards b_2. Hence, we would have $S_{b_2} = \{p_1\}$ and p_2 would envy p_1, a contradiction to the competitiveness.

The second part of statement (1) follows directly from the first part since b_2 is the only acceptable post left for p_2 and his maximum congestion towards b_2 is one.

By statement (1) and the maximum congestions, every dummy agent u_z can only be assigned to some original post alone, proving statement (2). The last statement follows directly from the first two statements. □

C.3 Continuation of the proof of Lemma 3

Lemma 3 ($\star$)**.** *An instance I admits a CP assignment if and only if there exists an integer k with $\max(0, m - n) \le k \le m - 1$ such that the instance I_k created by Construction 1 admits a CP assignment with only filled post.*

We continue to show why Π_k is CP. Since no post is empty, showing competitiveness reduces to showing that no agent is envious. This is clearly the case for all dummy agents including p_1 and p_2 since they are assigned to one of their most preferred posts alone. No original agent envies any other original agent or dummy agent since Π is CP. No original agent $v \in V$ envies p_1 or p_2 since p_1 and p_2 are assigned to b_1 and b_2, and b_1 and b_2 occurs at the end of $\succeq_v^*$. This shows that Π_k is CP for I_k, as desired.

For the "if" part, let k be an integer between $\max(0, m - n)$ and $m - 1$ such that the created instance I_k admits a CP assignment Π_k without empty post. We claim that the assignment Π derived from Π_k by omitting all dummy agents and the posts p_1 and p_2 is CP for I.

We first show that Π is a valid assignment for I. By Observation 1(2), every dummy agent is assigned to some original post alone, and hence for each $a_j \in A$ that is not assigned any dummy post (i.e., $\{u_1, \ldots, u_k\} \cap \Pi_k(a_j) = \emptyset$), we have $\Pi(a_j) = \Pi_k(a_j)$; and $\Pi(a_j) = \emptyset$, otherwise. This implies that $\Pi(a_j) \subseteq V$.

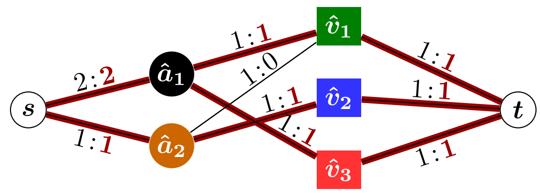


Figure 2: Flow network for iteration 2 where $T = 21$.

By Observation 1(3), every original agent is assigned to an original post, confirming that Π is indeed a valid assignment for I.

Next, suppose, for the sake of contradiction, that Π is not competitive and let v and a be an agent and a post, respectively, such that $(a, \max(|\Pi(a)|, 1)) \succ_v (a', |\Pi(a')|)$ where a' is the post that v is assigned to by Π. We infer that $\Pi(a)$ cannot be empty since otherwise by Construction 1 and by Observation 1(2) we would have that $\Pi_k(a) = \{u_z\}$ for some dummy agent u_z. This further implies that v envies u_z in I_k, a contradiction to the competitiveness of Π_k. Since $\Pi(a)$ is not empty and $v \in \Pi(a')$, again by Construction 1 and by Observation 1(2), we have that $\Pi(a) = \Pi_k(a)$ and $\Pi(a') = \Pi_k(a')$. Since $\succeq^*_v$ is an extension of $\succeq_v$ for each $v \in A$, we obtain that $(a, |\Pi_k(a)|) \succ^*_v (a', |\Pi_k(a')|)$ a contradiction to the competitiveness of Π_k.

C.4 Example of Algorithm 1

Consider the first instance in Example 1. Initially, $T[a_1] = T[a_2] = 1$, implying that the condition in Line 4 is satisfied, so we can start with the first iteration. In the first iteration, where $T = 11$, the network and its maximum flow constructed in Phase 1 (Lines 5–6) have been discussed in Example 2 already. Hence, we proceed with Line 7. Since it does not have value $|V| = 3$, we proceed with Phase 2, where we find an obstruction in Lines 8–14. We start with $V' = \{\hat{v}_3\}$ and $A' = \emptyset$ since $\hat{v}_3$ is the only vertex with $f(\hat{v}_3, t) = 0$. In the while-loop in Lines 10–14, we first find $\hat{a}_1$ and enlarge $A' = \{\hat{a}_1\}$. Then, we enlarge $V' = \{\hat{v}_3, \hat{v}_1\}$ in Line 13. Since no further post $\hat{a}$ exists that has an arc to any agent $\hat{v}$ in V'. We stop with $A' = \{\hat{a}_1\}$ and $V' = \{\hat{v}_3, \hat{v}_1\}$. One can check that if a_1 would stay with congestion one, then no CP assignment can exists since both v_1 and v_3 would envy the only agent that is assigned to a_1. We will later show that in order to have a CP assignment, it is necessary to increase the congestion of every post in A'. In Phase 3, we increment $T[a_1]$ to 2. This completes the first iteration.

At Line 4, we verify that $T[a_1] + T[a_2] = 3 \le |V|$, we continue with the second iteration. In the following, the tuples considered in Construction 2(v) are boldfaced.

$$\begin{aligned}
v_1\colon\ & (a_1, 1) \succ \boldsymbol{(a_1, 2)} \sim \boldsymbol{(a_2, 1)} \succ (a_2, 2) \succ \cdots,\\
v_2\colon\ & (a_1, 1) \sim \boldsymbol{(a_2, 1)} \succ (a_1, 2) \succ (a_2, 2) \succ \cdots,\\
v_3\colon\ & (a_1, 1) \succ \boldsymbol{(a_1, 2)} \succ (a_2, 1) \succ (a_2, 2) \succ \cdots.
\end{aligned}$$

One can verify that among all tuples $(a, T[a])$, $a \in A$, tuple $(a_1, 2)$ is the most preferred tuple of v_1 and v_3, while $(a_2, 1)$ is the most preferred tuple of v_1 and v_2. Hence, the network and maximum flow (highlighted with red lines) constructed in Phase 1 are as given in Figure 2; we note that this corresponds to the bipartite graph in Appendix A.2.

In Line 6, we verify that the maximum flow is a perfect flow (i.e., with value $|V|$). Hence, we derive and return an assignment according to Definition 2. This is exactly Π_2 from Example 1.

One can verify that the second instance of Example 1 where every agent has the same preference list as v_2 will lead to the sum of the congestion entries to exceed $|V| = 3$ in the second iteration, certifying that the instance does not have a CP assignment.

C.5 Proof of Lemma 4

Lemma 4 ($\star$)**.** *Each (A', V') computed in Phase 2 in lines 8–14 is an obstruction.*

Proof. Let (A', V') be the pair computed in lines 8–14 in some iteration z. Let (G, c) be the network with $G = (\hat{A} \cup \hat{V} \cup \{s, t\}, E)$ and f the maximum flow computed in this iteration. We aim to show that (A', V') satisfies the properties in Definition 3.

By line 7, f fails to have value $|V|$, i.e., $\sum_{\hat{v} \in \hat{V}} f(\hat{v}, t) < |V|$. Hence, there must be a vertex $\hat{v}^* \in \hat{V}$ with $f(\hat{v}^*, t) = 0$. Let $\hat{v}^*$ be such a vertex that is added to V' in line 9. Then, the first part of property (i) is clear since $\hat{v}^* \in V'$ (line 9) and we only add vertices from $\hat{V}$ to V'; see line 13.

Property (ii) is also clear due to line 11.

Let us consider property (iii). Clearly, for every vertex $\hat{a} \in A'$ with $(\hat{a}, \hat{v}^*) \in E(G)$ we must have that $f(s, \hat{a}) = \mathsf{c}(s, \hat{a})$ as otherwise we could increase the flow by one by setting $f(s, \hat{a}) = f(s, \hat{a}) + 1$ and $f(\hat{a}, \hat{v}^*) = f(\hat{v}^*, t) = 1$. By line 13, every out-neighbor $\hat{v}$ of $\hat{a}$ with $f(\hat{a}, \hat{v}) = 1$ is added to V'. Together with $\hat{v}^*$, we obtain that $\mathsf{c}(s, \hat{a}) < |\{\hat{v} \in V' \mid (\hat{a}, \hat{v}) \in E\}|$, as desired.

Consider an arbitrary vertex $\hat{a} \in A'$ with $(\hat{a}, \hat{v}^*) \notin E(G)$. Suppose, towards a contradiction, that $\hat{a}$ does not satisfy property (iii), meaning that $\mathsf{c}(s, \hat{a}) \geq |\{\hat{v'} \in V' \mid (\hat{a}, \hat{v'}) \in E(G)\}|$. We aim to show that there is an "augmenting" path from $\hat{a}$ to $\hat{v}^*$, with arcs having flow values alternating between zero and one, which is a witness for the flow to be not maximum.

Let us review the repeat-loop in lines 10–14. Observe that in each round of this loop, we aim at finding a vertex $\hat{a}$ not already in A' that has an out-arc to some agent-vertex from V'; V' is initialized with $V' = \{\hat{v}^*\}$. This implies that we can find a vertex in $\hat{v}_x \in V' \setminus \{\hat{v}^*\}$ due to which we add $\hat{a}$ in line 11. Further, for each vertex $\hat{v'}$ in $V' \setminus \{\hat{v}^*\}$, we can also find a vertex $\hat{a'}$ in a previous round such that $f(\hat{a'}, \hat{v'}) = 1$ in line 13. Let $\hat{a}_x$ be the vertex from A' due to which we add $\hat{v}_x$, i.e., $f(\hat{a}_x, \hat{v}_x) = 1$. Since each vertex in V' has only one out-arc with capacity one, due to the conservation constraint of the flow f, we infer that $f(\hat{a}, \hat{v}_x) = 0$; recall that $f(\hat{a}_x, \hat{v}_x) = 1$. Repeating the above reasoning, there must be a vertex $\hat{v}_{x-1}$ from $V' \setminus \{\hat{v}_x\}$ due to which we add $\hat{a}_x$. Then, either $\hat{v}_{x-1} = \hat{v}^*$ or $\hat{v}_{x-1} \neq \hat{v}^*$.

In the former case, we infer that $(\hat{a}, \hat{v}_x, \hat{a}_x, \hat{v}^*)$ is an augmenting path, so flipping the flow values along the path would increase the value of the flow:

$$\begin{aligned} f(s, \hat{a}) &= f(s, \hat{a}) + 1, & f(\hat{a}, \hat{v}_x) &= 1, \\ f(\hat{a}_x, \hat{v}_x) &= 0, & f(\hat{a}_x, \hat{v}^*) &= f(\hat{v}^*, t) = 1, \end{aligned}$$

a contradiction.

In the latter case, since V' is finite and no vertex from $\hat{V}$ can obtain more than one positive flow, by repeating the above reasoning, we must end up with an arc to $\hat{v}^*$ with zero flow; recall that $\hat{v}^* \in V'$. Then, we again obtain an augmenting path $P = (\hat{a}, \hat{v}_x, \hat{a}_x, \ldots, \hat{a}_0, \hat{v}_0 = \hat{v}^*)$. Analogously, we can increase the total flow by flipping the flow values along this path, a contradiction.

It remains to show property (iv). This is clear since otherwise for each vertex $\hat{v} \in V'$ we could find a vertex $\hat{a} \in A'$ and set $f(\hat{a}, \hat{v}) = f(\hat{v}, t) = 1$. In particular, the starting vertex $\hat{v}^*$ would have positive flow going through it, a contradiction. ☐

C.6 Proof of Lemma 5

Lemma 5 ($\star$)**.** *Assume that I admits a CP assignment Π with no posts being empty. Then, for each iteration $z \geq 1$ and each obstruction (A', V') found in iteration z, for each $a \in A$, if $\hat{a} \in A'$, then $|\Pi(a)| \geq T_z[a] + 1$; otherwise $|\Pi(a)| \geq T_z[a]$.*

Proof. Let us consider the first iteration and let (A', V') be the found obstruction. Since Π does not have empty posts, the statement clearly holds for all posts $a \in A$ with $\hat{a} \notin A'$. Let $N = (G, \mathsf{c})$ denote the network and f the maximum flow of N computed in the first phase. Let $P = \{v \in V \mid \hat{v} \in V'\}$ and $Q = \{a \in A \mid \hat{a} \in A'\}$ be the set of agents and posts that correspond to the vertices in V' and A', respectively.

Suppose, for the sake of contradiction, that there exists a post $a \in Q$ with $|\Pi(a)| \leq T_1[a] = 1$. By Definition 3(iii), more than $\mathsf{c}(s, \hat{a}) = T_1[a] = 1$ vertex from V' is incident to $\hat{a}$ in G. By Construction 2(v), at least two agents from P consider $(a, 1)$ as one of the most preferred tuples.

Since $|\Pi(a)| \leq 1$, at least one agent from P is not assigned to a but considers $(a, 1)$ as one of the most preferred tuples. Let $v_0 \in P$ be such an agent. Then, he must be assigned to some other post a_0 such that $(a_0, |\Pi(a_0)|)$ is one of the most preferred tuples for v_0 as well. This implies that $|\Pi(a_0)| = 1$ since we are in the first iteration. By Construction 2(v), we infer that $(\hat{a}_0, \hat{v}_0) \in E(G)$, and by line 11, that $\hat{a}_0 \in A'$.

By Definition 3(iii), more than $\mathsf{c}(s, \hat{a}_0) = T_1[a_0] = 1$ vertex from V' is incident to $\hat{a}_0$ in G, and we can find another agent $v_1 \in P$ that is not assigned to a_0 but considers $(a_0, 1)$ as one of the most preferred tuples. Again, this agent v_1 will be assigned to some post a_1 with $(a_1, 1)$ being one of the most preferred tuples of v_1. By Construction 2(v), we infer that $(\hat{a}_1, \hat{v}_1) \in E(G)$, and by line 11 that $\hat{a}_1 \in A'$. Repeating the above reasoning, we will be able to find a distinct vertex $\hat{a}_i \in A'$ for each vertex $\hat{v}_i \in V'$ such that $\Pi(a_i) = \{v_i\}$. That is, $|A'| \geq |V'|$, a contradiction to Definition 3(iv) since $|A'| = \sum_{\hat{a} \in A'} \mathsf{c}(s, \hat{a}) < |V'|$ in this case.

Now, let us consider other iterations. For each z, let (A'_z, V'_z) be the obstruction found in iteration z. Note that the table entries never decrease. Hence, if the statement were incorrect, there must be an iteration $z \geq 2$ where the statement holds in all iterations $z' \leq z - 1$ but not in iteration z.

Suppose, for the sake of contradiction, that the statement is incorrect and let z be the index of the first such iteration where the statement is incorrect. That is, in all iterations $z' \leq z - 1$, we have that for all $a' \in A$,

$$\text{if } \hat{a}' \in A'_z,\ |\Pi(a')| \geq T_{z'}[a'] + 1; \text{otherwise } |\Pi(a')| \geq T_{z'}[a'] \tag{1}$$

but there exists a post a such that

$$\text{if } \hat{a} \in A'_z,\ |\Pi(a)| \leq T_z[a];\ \text{ if } \hat{a} \notin A'_z,\ |\Pi(a)| < T_z[a]. \tag{2}$$

First, observe that $\hat{a} \in A'_z$ since otherwise $\Pi(a) \geq T_{z-1}[a] = T_z[a]$ by line 15, a contradiction to the assumption.

Next, we claim that $T_z[a] = |\Pi(a)|$. If $\hat{a} \in A'_{z-1}$ (i.e., $\hat{a}$ was in the obstruction found in iteration $z - 1$), then by assumption (1)–(2) and by line 15, we infer $|\Pi(a)| \geq T_{z-1}[a] + 1 = T_z[a] \geq |\Pi(a)|$, as desired. If $\hat{a} \notin A'_{z-1}$, then again by assumption (1)–(2) and by line 15, we infer $|\Pi(a)| \geq T_{z-1}[a] = T_z[a] \geq |\Pi(a)|$, as desired as well.

Recall that we inferred that $\hat{a} \in A'_z$. Let (G, c) denote the network constructed in iteration z. By Lemma 4, (A'_z, V'_z) is an obstruction for (G, c). Let $\hat{v}^* \in V'$ be the starting vertex with $f(\hat{v}^*, t) = 0$. By Definition 3(iii), more than $\mathsf{c}(s, \hat{a}) = T_z[a]$ vertices from V'_z exist that have a as an in-neighbor. By Construction 2 (v), more than $T_z[a] = |\Pi(a)|$ agents from V'_z consider $(a, T_z[a])$ as one of the most-preferred tuples among all $(a', T_z[a'])$. Hence, at least one of such agents is not assigned to a by Π.

Let $\hat{v} \in V'_z$ be a vertex whose corresponding agent v considers $(a, |\Pi(a)|)$ as one of the most preferred tuples but is assigned to some other post $a' \neq a$. Then, $(\hat{a}, \hat{v}) \in E(G)$. To prevent v from being envious (recall that no post is empty), we must have that $(a', |\Pi(a')|) \succeq_v (a, |\Pi(a)|)$.

We claim that $\hat{a}' \in A'_z$ as well. Since $(a, |\Pi(a)|)$ is one of the most-preferred tuples of v among all $(a', T_z[a'])$, by previous paragraph and by congestion aversion, we infer that $T_z[a'] \geq |\Pi(a')|$. If $T_z[a'] > |\Pi(a')|$, then by line 15, there must exists an iteration $z' \in [z - 1]$ where $\hat{a}' \in A'_{z'}$ and $T_{z'}[a'] = |\Pi(a')|$, a contradiction to (1). Hence, $T_z[a'] = |\Pi(a')|$, implying that $(a', T_z[a'])$ is also one of the most preferred tuples of v among all $(a'', T_z[a''])$. By Construction 2(v), we have $(\hat{a}', \hat{v}') \in E(G)$, and by line 11, we have $\hat{a}' \in A'_z$, as desired.

By Definition 3(iii), we infer that more than $\mathsf{c}(a') = T_z[a'] = |\Pi(a')|$ vertices from V'_z have in-arcs from $\hat{a}'$. By Construction 2(v), more than $\mathsf{c}(a') = T_z[a'] = |\Pi(a')|$ agents consider $(a', |\Pi(a')|)$ as one of the most preferred tuples among all $(p, T[p]), p \in A$.

Analogously, we can again find another vertex $\hat{v'} \in V'_z$ such that v' considers $(a', |\Pi(a')|)$ as one of the most preferred tuples among all $(p, T[p]), p \in A$, but is assigned to some other post $a'' \neq a'$ with $\hat{a''} \in A'_z$ and $T_z[a''] = |\Pi(a'')|$. By repeating this argument, we infer that every vertex $\hat{\alpha} \in A'_z$ has $T_z[\alpha] = |\Pi(\alpha)|$. By Definition 3(iv), we have that $|V'_z| > \sum_{\hat{\alpha} \in A'_z} \mathsf{c}(s, \hat{\alpha}) = \sum_{\hat{\alpha} \in A'_z} T_z[\alpha] = \sum_{\hat{\alpha} \in A'_z} |\Pi(\alpha)|$. So there must be a vertex $\hat{\mu} \in V'_z$ such that μ is assigned to a post b with $\hat{b} \notin A'_z$. By line 11 and Construction 2(v), let $\hat{\alpha} \in A'_z$ with $(\hat{\alpha}, \hat{\mu}) \in E(G)$ such that $(\alpha, T_z[\alpha])$ is a most preferred tuple of μ among all tuples $(p, T_z[p]), p \in A$.

By our previous argument, we have that $T_z[\alpha] = |\Pi(\alpha)|$. By CP, we have that $(b, |\Pi(b)|) \succeq_v (\alpha, |\Pi(\alpha)|)$. Since $(\alpha, T_z[\alpha])$ is a most preferred tuple of μ among all tuples $(p, T_z[p])$, $p \in A$, we further infer that $T_z[b] \geq |\Pi(b)|$.

Since $b \notin A'_z$, meaning by line 11 that $(\hat{b}, \hat{\mu}) \notin E(G)$, by Construction 2(v), we further infer that $|\Pi(b)| < T_z[b]$. By line 15, there must exist an iteration $z' \in [z-1]$ with $T_{z'}[b] = \Pi(b)$ and $T_{z'}[b]$ was incremented. This is a contradiction to (1) however. ☐

C.7 Proof of Lemma 6

Lemma 6 ($\star$). *If Π is an assignment returned in line 7, then Π is CP and has no empty post.*

Proof. Let z be the integration and f be the perfect flow based on which Π is computed in line 7. By the definition of perfectness (see Definition 2), the value of f equals the number $|V|$ of agents. This means that $\sum_{a \in A} |\Pi(a)| = |V|$. By the capacity constraints, we obtain that $|V| = \sum_{a \in A} |\Pi(a)| \leq \sum_{a \in A} T_z[a] \leq |V|$, the last inequality holds due to the while-loop-condition in line 4. Hence, for each post $a \in A$ we must have that $|\Pi(a)| = T_z[a]$ since $|\Pi(a)| \leq T_z[a]$ holds by the capacity constraints in Construction 2(iii).

This implies that $\Pi(a) \neq \emptyset$ since $T_z[a] \geq 1$. Hence, in order to show that Π is CP, it suffices to show that for each agent v that is assigned to a post a and for each post a' with $a' \neq a$ we have that $(a, |\Pi(a)|) \succeq_v (a', |\Pi(a')|)$. Suppose, for the sake of contradiction, that $(a', |\Pi(a')|) \succ_v (a, |\Pi(a)|)$, meaning by the first paragraph that $(a', T_z[a']) \succ_v (a, T_z[a])$, a contradiction to Construction 2(v). ☐

C.8 Continuation of the proof of Theorem 1

Theorem 1 ($\star$). *Algorithm 1 correctly decides whether an instance has a CP assignment in $O(m^2 \cdot (n+m)^2)$ time, where m and n denote the number of posts and agents, respectively.*

It remains to analyze the running time. The main body of the algorithm is a for-loop (line 1) and has at most m iterations. In each iteration k, the algorithm constructs a new instance I according to Construction 1. Note that I has $O(n+m)$ agents and $O(m)$ posts, and it can be constructed in $O((n+m)^2)$ time since each agent has $O(n+m)$ tuples in his preference list. Then, we continue with the big **while**-loop in lines 4–16. If we can show the **while**-loop run in $O(m \cdot (n+m)^2)$ time, we obtain our desired running time of $O(m^2 \cdot (n+m)^2)$.

So, it remains to analyze the **while**-loop. In line 3, initializing the table T needs $O(m)$ time. The while-loop (lines 4–15) runs at most n times since no table entries are ever decreased and in each iteration at least one table entry is increased by one. For each iteration, we first construct a network $N = (G, \mathsf{c})$ based on (I, T); see Construction 2. The directed graph G has $O(n+m)$ vertices and $O(m \cdot n)$ arcs, and each capacity value is in $O(n)$. Hence, constructing the network needs $O(m \cdot n)$ time.

Afterwards, there are three phases. The first phase (lines 5–7) finds a maximum flow for N and checks whether its value is $|V|$. Computing a maximum flow can be done in $O(m \cdot n)$ time and comparing two values needs constant time. Hence, the first phase needs $O(m \cdot n)$ time.

The second phase (lines 8–14) finds an obstruction (A', V') by first finding a vertex $\hat{v}^*$ with $f(\hat{v}^*, t) = 0$. This can be done in $O(1)$ time if we store such information when we compare the value of the flow with $|V|$ in the first phase. Hence, the initialization of V' and A' needs $O(1)$ time. Then, the algorithm goes to the repeat-loop in lines 10–14. To analyze the running time of this loop, we observe that there are $O(m \cdot n)$ arcs between A' and V' and each arc only needs to be checked at most once during the whole loop (line 11). Adding new vertices to V' can be done in $O(m \cdot n)$ time as well since for each newly added alternative $\hat{a}$ there are at most n vertices $\hat{v}$ from $\hat{V}$ with positive flow from $\hat{a}$ to $\hat{v}$. Hence, the repeat-loop needs $O(m \cdot n)$ time.

It is straightforward that the last phase (lines 15–15) runs in $O(m)$ time. Summarizing, we obtain that the desired $O(m \cdot n^2)$ time for the **while**-loop.

D Additional Material for Section 4

D.1 Correctness of the Construction in the Proof of Theorem 2

Theorem 2 ($\star$)**.** EF+TG *is NP-complete; hardness holds even if there are no ties.*

Proof of the correctness of the construction. One can verify that the constructed preferences do not contain ties. Due to TG, the maximum congestions of the agents are depicted in the following table, where v_i is an element-agent with u_i appearing in C_1, C_2, C_m:

	a_1	a_2	a_3	$\dots$	a_m	b_1	b_2
v_i	3	3	0	$\dots$	3	0	$3n+4m-9$
p_z	2	2	2	$\dots$	2	$3n+2m$	0
q_z	0	0	0	$\dots$	0	$3n+2m$	$2m$

Correctness. It remains to show the correctness, i.e., I has an exact cover if and only if I' admits an EF and TG assignment.

For the "only if" part, let $J \subseteq [m]$ denote an exact cover for I. Then, we claim that the following assignment Π is EF+TG:

– For each $j \in J$, let $\Pi(a_j) = \{v_i \mid u_i \in C_j\}$.
– For each $j \in [m] \setminus J$, let $\Pi(a_j) = \emptyset$.
– Let $\Pi(b_1) = \{p_j \mid j \in [2m]\}$ and $\Pi(b_2) = \{q_j \mid j \in [2m]\}$.

Since each set-post contains either zero or three agents, no dummy agent envies any element-agent. The dummy agents also do not envy each other due to their preferences. Similarly, no two element-agents envy each other and no element-agent envies any dummy agent since he does not like b_1 or b_2 more.

For "if" part, let Π be an EF and TG assignment for the constructed instance. We aim at showing that the set-posts that are assigned element-agents constitute an exact cover. To this end, let $J = \{j \mid \exists v_i \text{ with } v_i \in \Pi(a_j)\}$. We first show two claims.

Claim D.1.1. *For each set-post a_j it holds that $|\Pi(a_j)| \in \{0, 3\}$.*

Proof. Since there are $2m$ dummy agents $\{p_1, p_2, \dots, p_{2m}\}$, but there are only m set-posts, at least one dummy agent, say p_z, is not assigned to a set-post alone. Hence, for every set-post a_j, it holds that $|\Pi(a_j)| \neq 1$ since otherwise p_z would envy the agent that is assigned to a_j. Since the maximum congestion for every set-post is 3, we further infer that $|\Pi(a_j)| \in \{0, 2, 3\}$ holds for every set-post a_j. In particular, this implies that no dummy agent p_z with $1 \le z \le 2m$ is assigned to a set-post alone.

Towards a contradiction, suppose that there exists a set-post a_j with $|\Pi(a_j)| \notin \{0, 3\}$. This implies that $|\Pi(a_j)| = 2$. Then, every dummy agent p_z with $1 \le z \le 2m$ is to be assigned a set-post since otherwise he would envy the two agents that are assigned to a_j. Since there are exactly $2m$ dummy agents$p_1, \dots, p_{2m}$, this means that every set-post a_x, $x \in [m]$, must have $|\Pi(x)| = 2$. Then, no other agent can be assigned to the set-post. However, all element-agents will envy all p_i's, a contradiction. This concludes the proof.

(end of the proof of Claim D.1.1 $\diamond$)

By Claim D.1.1, we know that each set-post is assigned either zero or three agents. Next, we show that every element-agent v_i is assigned to an *acceptable* set-post.

Claim D.1.2. *For each element-agent v_i it holds that $\Pi(v_i) \in \{a_j \mid j \in [m]$ and $u_i \in C_j\}$.*

Proof. Suppose this is not true, and by TG let v_i denote an element-agent that is assigned to b_2; note that v_i does not find b_1 acceptable. Since there are $2m$ dummy agents q_z each with congestion $2m$ for b_2, at least one of them is *not* assigned to b_2. This agent envies v_i, a contradiction.

(end of the proof of Claim D.1.2 $\diamond$)

Claim D.1.2 implies that J is a set cover, while Claim D.1.1 implies that $|J| \le n$. Altogether we conclude that J is an exact cover. $\square$

D.2 Proof of Theorem 3

Theorem 3 ($\star$)**.** EF+TG *is FPT with respect to the number* n *of agents and the number* m *of posts, respectively.*

Proof. We first consider the parameter n. Let $I = (A, V, (\succeq_v)_{v \in V})$ be an instance of CONGESTED ASSIGNMENT. Due to TG, each agent is assigned to one of his first n tuples. Hence, for each agent $v_i \in V$, we guess (by brute-force searching) which of his first n tuples that v_i is "assigned" to, i.e., (a, d). After assigning all the agents, we check in linear time whether this results in a valid assignment, i.e., if we v_i "assign" (a, d), then there must be exactly d agents that are guessed to be "assigned" to (a, d). We abandon the current guess if it does not give a valid assignment; otherwise we proceed to check EF in $O(n^2)$ time.

Since there are n agents, each with n choices, the whole procedure can be done in $O(n^n \cdot (n^2 + m))$ time.

Now, we consider the parameter m. Let $I = (A, V, (\succeq_v)_{v \in V})$ be an instance of CONGESTED ASSIGNMENT. We guess (by brute-force searching) the set of empty posts $A' \subseteq A$ in the sought solution. Then, we modify I by removing the posts in A' from the preference lists of all agents $v \in V$; denote the new preference list as $\succeq'_v$. Let $I' = (V, A \setminus A', (\succeq'_v)_{v \in V})$ denote the modified instance.

Since for assignments with only filled posts, CP and EF are equivalent, we obtain that I admits an EF and TG assignment where all posts in A' are empty and the rest is non-empty if and only if I' admits CP assignment where all posts are non-empty. The latter problem can be checked in polynomial via lines 3–16 in Algorithm 1. Since there are 2^m subsets of empty posts to check, the overall running time is $2^m \cdot (n + m)^{O(1)}$, which is an FPT time with respect to m. $\square$

> CLIQUE
> **Input:** An undirected graph $G{=}(U, E)$, an integer $h \geq 0$.
> **Question:** Does G admit a *clique* of size h, i.e., a size-h subset $U' \subseteq U$ which induces a complete subgraph?

D.3 Proof of Theorem 4

Theorem 4 ($\star$)**.** MAXCP+TG *is W[1]-hard with respect to the number* t *of unsatisfied agents and can be solved in* $\mathcal{O}(n^t m^t)$ *time, where* n *and* m *denote the number of agents and the number of posts, respectively; hardness holds even if there are no ties.*

Proof. We first show the W[1]-hardness by providing a parameterized reduction from the CLIQUE problem. Let $I = (G = (U, E), h)$ denote an instance of CLIQUE with $U = \{u_1, \ldots, u_{|U|}\}$ and $E = \{e_1, \ldots, e_{|E|}\}$. We create a MAXCP+TG instance $I' = (A, V, (\succeq_v)_{v \in V}, k)$ as follows. Let $k = h + h(h-1)$. We will see that the agents corresponding to the vertices and edges of a size-h clique are the only unsatisfied agents.

- For each vertex $u_i \in U$, create a *vertex-post* a_i, a *vertex-agent* w_i, and $h - 1$ copies of w_i, denoted as $\tilde{w}_i^z$ with $z \in [h-1]$.
- For each edge $e_\ell \in E$ with $e_\ell = \{u_i, u_j\}$, create an *edge-post* b_ℓ and three *edge-agents* e_ℓ^*, e_ℓ^i, and e_ℓ^j.
- Create L dummy agents $x_1, \cdots, x_L$ with $L = |U|(h-2) + (|U| - h - 1) + (h(h-1) - 1) + (|E| - \binom{h}{2} - 1) + (k+1)$; note that L is a very large number.
- Create five auxiliary posts $a_0, \tilde{a_0}, b_0, y, c_0$. Post y shall accommodate all dummy agents, while c_0 is a "blocker" making sure that agents are assigned to desired posts.

Let $V = W \cup \bigcup_{u_i \in U} \tilde{W}_i \cup \{e_\ell^*, e_\ell^i, e_\ell^j \mid e_\ell \in E, e_\ell = \{u_i, u_j\}\} \cup \{x_i \mid i \in [L]\}$, and $\mathcal{A} = \{a_i \mid u_i \in U\} \cup \{b_\ell \mid e_\ell \in E\} \cup \{a_0, \tilde{a_0}, b_0, y, c_0\}$, where $W = \{w_i \mid u_i \in U\}$ and $\tilde{W}_i = \{\tilde{w}_i^z \mid z \in [h]\}$. Let $n = |V|$.

Preferences. We describe the preferences of the agents, restricted to the first n tuples. Here, $\langle \alpha, s, t \rangle = (\alpha, s) \succ (\alpha, s+1) \succ \cdots \succ (\alpha, t)$ depict the preference list on tuples for post α and congestions ranging between s and t.

– The dummy agent x_i with $i \in [L]$ has the following preference list:

$$\begin{aligned} x_i\colon & \langle a_1, 1, h-2\rangle \succ \langle a_2, 1, h-2\rangle \succ \cdots \succ \langle a_{|U|}, 1, h-2\rangle \succ \\ & \langle a_0, 1, |U|-h-1\rangle \succ \langle \tilde{a_0}, 1, h(h-1)-1\rangle \succ \\ & \langle b_0, 1, |E| - \binom{h}{2} - 1\rangle \succ \langle y, 1, L\rangle \succ \\ & \langle c_0, 1, n+(k+1)-2L\rangle. \end{aligned}$$

The dummy agents shall ensure some minimum number of agents assigned to each post (except b_ℓ and y): At least $h-1$, $|U|-h$, $h(h-1)$, and $|E|-\binom{h}{2}$ agents are to be assigned to $a_i (i \in [|U|])$, a_0, $\tilde{a_0}$, and b_0, respectively. The reason is that since at least $k+1$ dummy agents are to be assigned to y, they would envy the agents assigned to a post if its congestion is less than or equal to the maximum congestion of x_i to that post, which is not possible for a yes instance. Indeed, the dummy agents can only be assigned to y.

– The vertex-agent w_i with $i \in [|U|]$ has the following preference list:

$$\begin{aligned} w_i\colon & \langle a_0, 1, |U|-h-1\rangle \succ \langle a_i, 1, h-2\rangle \succ \\ & (a_0, |U|-h) \succ (a_i, h-1) \succ (a_i, h) \succ \langle c_0, 1, n-|U|\rangle. \end{aligned}$$

We will see that exactly $|U|-h$ vertex-agents w_i are assigned to a_0. Consequently, the remaining h vertex-agents must be assigned to a_j with $j \in [|U|]$, which shall correspond to the clique-vertices if G admit a size-h clique.

– Each copy $\tilde{w}_i^z$ with $i \in [|U|]$ and $z \in [h-1]$ of a vertex-agent has the same preference list:

$$\begin{aligned} \tilde{w}_i^z\colon & \langle \tilde{a_0}, 1, h(h-1)-1\rangle \succ \langle a_i, 1, h-1\rangle \succ \\ & (\tilde{a_0}, h(h-1)) \succ \langle c_0, 1, n-(h+1)(h-1)\rangle. \end{aligned}$$

The copy-agents shall ensure that all $\tilde{w}_i^z$, $z \in [h-1]$, are jointly assigned to either $\tilde{a_0}$ or a_i. If they are assigned to a_i, then no other agent (including w_i) can be assigned to a_i. This corresponds to the case that the vertex u_i is not in the clique.

– The edge-agents e_ℓ^*, e_ℓ^i and e_ℓ^j with $e_\ell = \{u_i, u_j\}$ have the preference lists:

$$\begin{aligned} e_\ell^*\colon & \langle b_0, 1, |E| - \binom{h}{2} - 1\rangle \succ (b_\ell, 1) \succ (b_0, |E| - \binom{h}{2}) \succ \\ & (b_\ell, 2) \succ \langle c_0, 1, n - (|E| - \binom{h}{2}) - 2\rangle. \\ e_\ell^i\colon & (b_\ell, 1) \succ (b_\ell, 2) \succ \langle a_i, 1, h\rangle \succ \langle c_0, 1, n-h-2\rangle. \\ e_\ell^j\colon & (b_\ell, 1) \succ (b_\ell, 2) \succ \langle a_j, 1, h\rangle \succ \langle c_0, 1, n-h-2\rangle. \end{aligned}$$

Note that e_ℓ^* can only be assigned to b_ℓ or b_0, and $e_\ell^i(resp.\ e_\ell^j)$ only to $a_i(resp.\ a_j)$ or b_ℓ. If e_ℓ^* is assigned to b_ℓ and does not envy other agents, then no other agent can be assigned to b_ℓ, as otherwise at least $|E| - \binom{h}{2} + 1$ agents that must be assigned to b_0, which is impossible due to top-guarantees. Therefore, if e_ℓ^* is assigned to b_ℓ, then e_ℓ^i and e_ℓ^j have to be assigned to a_i and a_j, respectively. We will see that e_ℓ^* cannot be unsatisfied, and having e_ℓ^i and e_ℓ^j assigned to a_i and a_j correspond to having the edge e_ℓ in the clique if G admits a clique of size h.

The maximum congestions of the agents are depicted in Table 1. Here, we assume that $e_\ell = \{u_1, u_{|U|}\}$.

Correctness. Clearly, the construction can be done in polynomial time. It remains to show the correctness, i.e., I has a clique of size h if and only if I' admits a TG assignment with $t = h + h(h-1)$ agents being unsatisfied.

The "only if" part. Let $\mathcal{C} \subseteq U$ denote an h-clique for I. Let $E^{\mathcal{C}} \subseteq E$ denote the edge set associated with $\mathcal{C}$, i.e., $E^{\mathcal{C}} = \{e_\ell = \{u_i, u_j\} \mid u_i, u_j \in \mathcal{C}\}$. Then, we claim that the following assignment Π is a TG with t unsatisfied agents.

– For each $u_i \in \mathcal{C}$, assign w_i to a_i, and assign $\tilde{w}_i^z$ with $z \in [h-1]$ to $\tilde{a_0}$.

	a_1	$\ldots$	$a_{\lvert U\rvert}$	a_0	$\tilde{a_0}$	b_ℓ	b_0	y	c_0
w_1	h	0	0	$\lvert U\rvert - h$	0	0	0	0	$n - \lvert U\rvert$
$\vdots$	0	h	0	$\lvert U\rvert - h$	0	0	0	0	$n - \lvert U\rvert$
$w_{\lvert U\rvert}$	0	0	h	$\lvert U\rvert - h$	0	0	0	0	$n - \lvert U\rvert$
$\tilde{w}_1^z$	$h-1$	0	0	0	$h(h-1)$	0	0	0	$n - h^2 + 1$
$\vdots$	0	$h-1$	0	0	$h(h-1)$	0	0	0	$n - h^2 + 1$
$\tilde{w}_{\lvert U\rvert}^z$	0	0	$h-1$	0	$h(h-1)$	0	0	0	$n - h^2 + 1$
e_ℓ^*	0	0	0	0	0	2	$\lvert E\rvert - \binom{h}{2}$	0	$n - (\lvert E\rvert - \binom{h}{2}) - 2$
e_ℓ^i	h	0	0	0	0	2	0	0	$n - h - 2$
e_ℓ^j	0	0	h	0	0	2	0	0	$n - h - 2$
x_α	$h-2$	$h-2$	$h-2$	$\lvert U\rvert - h - 1$	$h(h-1) - 1$	0	$\lvert E\rvert - \binom{h}{2} - 1$	L	$n + (k+1) - 2L$

Table 1: Maximum congestions of the agents constructed for Theorem 4

- For each $u_i \notin \mathcal{C}$, assign $\tilde{w}_i^z$ with $z \in [h-1]$ to a_i, and assign w_i to a_0.
- For each $e_\ell = \{u_i, u_j\} \in E^{\mathcal{C}}$, assign e_ℓ^i to a_i, e_ℓ^j to a_j, and e_ℓ^* to b_ℓ.
- For each $e_\ell = \{u_i, u_j\} \notin E^{\mathcal{C}}$, assign e_ℓ^i and e_ℓ^j to b_ℓ, and e_ℓ^* to b_0.
- Assign x_α to y with $\alpha \in [L]$.

Clearly, Π is TG with the following congestion vector.

Observation 3. *Π is TG and satisfies the following.*

(i) $|\Pi(a_0)| = |U| - h$, $|\Pi(\tilde{a}_0)| = h(h-1)$, $|\Pi(b_0)| = |E| - \binom{h}{2}$, *and* $|\Pi(y)| = L$.
(ii) For each $u_i \in U$, *if* $u_i \in \mathcal{C}$, *then* $|\Pi(a_i)| = h$; *otherwise* $|\Pi(a_i)| = h - 1$.
(iii) For each $e_\ell \in E$, *if* $e_\ell \in E^{\mathcal{C}}$, *then* $|\Pi(b_\ell)| = 2$; *otherwise* $|\Pi(b_\ell)| = 1$.

Let $V' = \{w_i \mid u_i \in \mathcal{C}\} \cup \{e_\ell^i, e_\ell^j \mid e_\ell \in E^{\mathcal{C}} \text{ with } e_\ell = \{u_i, u_j\}\}$. Note that $|V'| = k$. We aim to show that all agents except those from V' are satisfied. By the above observation, it is straightforward that every dummy agent x_α is satisfied, every agent that *does not* correspond to the clique vertices is satisfied, and the copies $\tilde{w}_i^z$ of all vertex-agents are also satisfied. It remains to consider the edge-agents that are not in V'. Let $e_\ell \in E$ with $e_\ell = \{u_i, u_j\}$. Clearly, if $e_\ell \notin E^{\mathcal{C}}$, then the two edge-agents e_ℓ^i and e_ℓ^j are satisfied since they are assigned to their most preferred post. Agent e_ℓ^* is also satisfied since he is assigned to b_0 with congestion $|E| - \binom{h}{2}$ which is better than $(b_\ell, 2)$. If $e_\ell \in E^{\mathcal{C}}$, then agent e_ℓ^* is also satisfied since he is assigned to b_ℓ alone which is better than $(b_0, |E| - \binom{h}{2})$. Hence, only the agents in V' are unsatisfied. Since $|V'| = k$, this concludes the proof for the "only if" direction.

The "if" part. Let Π be a TG assignment with at most t unsatisfied agents. We aim to show that the following vertex subset $\mathcal{C}$ is a size-h clique: $\mathcal{C} = \{u_i \mid |\Pi(a_i)| \geq h\}$. Before we show this, we observe the following regarding the congestions and assignments of the posts.

Claim D.3.1. *(1)* $|\Pi(a_0)| = |U| - h$, $|\Pi(\tilde{a}_0)| = h(h-1)$, *and* $|\Pi(b_0)| = |E| - \binom{h}{2}$.
(2) For each $u_i \in U$, *it holds that* $|\Pi(a_i)| \in \{h-1, h\}$.
(3) For each $e_\ell \in E$, *it holds that* $|\Pi(b_\ell)| \leq 2$.
(4) $\Pi(a_0) \subseteq \{w_i \mid u_i \in U\}$, $\Pi(\tilde{a}_0) \subseteq \{\tilde{w}_i^z \mid i \in [|U|], z \in [h-1]\}$, *and* $\Pi(b_0) \subseteq \{e_\ell^* \mid e_\ell \in E\}$.
(5) All edge-agents $E^* = \{e_\ell^* \mid e_\ell \in E\}$ *are satisfied.*

Proof. We show the first two statements together by considering the dummy agents.

Since Π is TG and the maximum congestion of dummy x_α for $a_0, \tilde{a}_0, b_0$, and a_i with $i \in [|U|]$ are $|U| - h - 1$, $h(h-1) - 1$, $|E| - \binom{h}{2} - 1$, and $h - 2$, respectively, we infer by simple calculation that there are more than k dummy agents who are assigned to y or c_0. Since Π does not have more than t unsatisfied agents, this further implies that there is at least one satisfied dummy agent x_α who is assigned to y or c_0. By his preferences, every tuple that he prefers to $(y, k+1)$ must have congestion that exceeds his maximum durable congestion. This implies that $|\Pi(a_0)| \geq |U| - h$, $|\Pi(\tilde{a}_0)| \geq h(h-1)$, and $|\Pi(b_0)| \geq |E| - \binom{h}{2}$, for $|\Pi(a_i)| \geq h - 1$. Since no agent allows more than the aforementioned congestions (except for a_i), we further infer that $|\Pi(a_0)| = |U|-h$, $|\Pi(\tilde{a}_0)| = h(h-1)$, and $|\Pi(b_0)| = |E|-\binom{h}{2}$. For a_i, since the maximum congestion of any agent for a_i is h, we infer that $h-1 \le |\Pi(a_i)| \le h$.

Statement (3) is straightforward by observing the maximum congestion of any agent towards b_ℓ is two.

The first part of statement (4) follows from the fact that the only agents that have $(a_0, |U|-h)$ in their top n choices are the vertex-agents. Similarly, we can show that the other parts of the statement are also correct.

To show statement (5), let us analyze which agents are unsatisfied. To this end, define $W' = \{w_i \in W \mid w_i \notin \Pi(a_0)\}$. By statements (1) and (4), we infer that $|W'| = h$.

Further, every agent w_i in W' is unsatisfied since by statement (2) that $|\Pi(a_i)| \ge h-1$, any agent not assigned to a_0 will envy those that are assigned to a_0. This implies that at most $\binom{h}{2}$ agents other than W' can be unsatisfied.

By statement (2), partition $\{a_i \mid u_i \in U\}$ into A_1 and A_2 with $A_1 = \{a_i \mid u_i \in U \wedge |\Pi(a_i)| = h-1\}$ and $A_2 = \{a_i \mid u_i \in U \wedge |\Pi(a_i)| = h\}$. Note that by the top-guarantees, every post a_i from A_2 can only be assigned vertex-agents w_i or edge-agent e^i_ℓ for some edge $e_\ell \in E$ with $u_i \in e_\ell$. However, this implies that every agent assigned to post $a_i \in A_2$ is unsatisfied since w_i prefers $(a_0, |U|-h)$ to $(a_i, h-1)$ and every edge-agent e^i_ℓ (with $u_i \in e_\ell$) prefers $(b_\ell, 2)$ to (a_i, h); recall by statements (1) and (3) that $|\Pi(a_0)| = |U|-h$ and $|\Pi(b_\ell)| \le 2$. This further implies that $|A_2| \le h$ since $k = h + h(h-1) = h^2$.

By statement (1), we have that $|\Pi(\tilde{a}_0)| = h(h-1)$. Since every vertex-agent has $h-1$ copies, there are at least h vertices each of which has a copy-agent assigned to $\tilde{a}_0$. Since every copy-agent corresponding to vertex u_i prefers $(a_i, h-1)$ $(\tilde{a}_0, h(h-1))$, it follows that at least $h - |A_2|$ copy-agents will be unsatisfied, namely those whose corresponding vertex-post has congestion $h-1$.

Since at most h vertex-agents and at most $(|U|-h)(h-1)$ copy-agents can be assigned to any vertex-post a_i, the number of edge-agents e^i_ℓ that have to be assigned to some post a_i is at least

$$\begin{aligned} &|A_1|(h-1) + |A_2|h - h - (|U|-h)(h-1) \\ &\qquad = h(h-1) - (h - |A_2|). \end{aligned}$$

Observe that each edge-agent e^i_ℓ that is assigned to some vertex-post a_i is unsatisfied. This implies that at least $h(h-1) - (h-|A_2|)$ edge-agents are unsatisfied. Together with the $h - |A_2|$ unsatisfied copy-agents, no more other agent can be unsatisfied. In other words, every edge-agent e^*_ℓ must be satisfied, as desired.

Now, we turn to the XP result. The ideas is to guess the unsatisfied agents and the posts that they are assigned to, and replace them with dummies and run Algorithm 1 for the reduced instance. More precisely, we guess who are the unsatisfied agents in $O(n^t)$ time; denoting the set of unsatisfied agents as $V^* = \{v^*_1, \ldots, v^*_t\}$. For each V^*, we further guess which posts they are assigned to in $O(m^t)$ time; let a^*_z denote the guessed post that v^*_z will assigned to, $z \in [t]$.

Then we create t dummy agents $P = \{p_z \mid z \in [t]\}$ and set their preference list as $p_z\colon (a^*_z, 1) \succ \cdots \succ (a^*_z, n) \succ \cdots$. We replace the agents V^* with the dummies and use Algorithm 1 to solve the resulting instance. If Algorithm 1 returns no on the current guess, we proceed with the next guess; otherwise, let Π be CP assignment returned by Algorithm 1. It is straightforward that replacing each dummy P_z with v^*_z in the assignment yields a TG assignment with at most t unsatisfied agents.

The overall running time is $O(n^t m^t)$. (end of the proof of Claim D.3.1 $\diamond$)

Now, we are ready to show that $\mathcal{C}$ is a clique of size h.

We first show that $\mathcal{C}$ has size h. Define $W' = \{w_i \mid \Pi(w_i) \ne a_0\}$. By the preferences of the vertex-agents and by Claim D.3.1(1), $|W'| = h$ and every vertex-agent in W' is unsatisfied. By Claim D.3.1(1) and (5), every edge-agent e^*_ℓ that is *not* assigned to b_0 must be assigned to the corresponding edge-post b_ℓ with congestion one. This implies that the remaining two edge-agents e^i_ℓ and e^j_ℓ with $e_\ell = \{u_i, u_j\}$ are not assigned to b_ℓ and hence unsatisfied; they both envy e^*_ℓ. Define $E' = \{e^i_\ell, e^j_\ell \mid \Pi(e^*_\ell) = b_\ell\}$. Then, $|E'| = h(h-1)$ and it yields $h(h-1)$ unsatisfied edge-agents by Claim D.3.1(1). Together with the h unsatisfied vertex-agents in W', we infer that every copy-agent $\tilde{w}^z_i$ is satisfied. In particular, it means that for each copy-agent $\tilde{w}^z_i$ that is assigned to $\tilde{a}_0$ it must hold that $|\Pi(a_i)| = h$. Recall that there are at least h vertices u_i each of which has a copy-agent assigned to $\tilde{a}_0$. This further implies that there are at least h vertex-posts that each have congestion h, that is $|\mathcal{C}| = h$.

It remains to show that $\mathcal{C}$ is a clique.

By Claim D.3.1(5), for each edge e_ℓ with $\Pi(e^*_\ell) = b_0$ it must hold that $\Pi(b_\ell) = \{e^i_\ell, e^j_\ell\}$ where $e_\ell = \{u_i, u_j\}$ as otherwise e^*_ℓ would envy the only agent that is assigned to b_ℓ (recall that $|\Pi(b_\ell)| \in \{1, 2\}$). This implies that there are at most $h(h-1)$ edge-agents that can be assigned to any vertex-post a_i. They all come from E'. By previous reasoning, there are exactly h vertex-posts that each have congestion h. This implies by the maximum congestions that each such vertex-post a_i of congestion h must accommodate vertex-agent u_i and $h-1$ "incident" edge-agents e^i_ℓ with $e^i_\ell \in E'$. These $h(h-1)$ edge-agents are exactly those from E' and correspond to at least $\binom{h}{2}$ incident edges of the h vertices from $\mathcal{C}$. They form a clique only if these incident edges are of size exactly $\binom{h}{2}$. This is the case since E' corresponds to exactly $\binom{h}{2}$ edges, they form a clique of size h, as desired. □

D.4 Proof of Theorem 5

Theorem 5 ($\star$)**.** MaxCP+TG *is FPT with respect to n, and XP with respect to m, where n and m denote the number of agents and the number of posts, respectively.*

Proof sketch. **Parameter n:** We first guess a subset V' of unsatisfied agents. Afterwards, similarly to Theorem 3, we guess for each satisfied agent $V \setminus V'$ one of his first n tuples and check whether the $|V \setminus V'|$ guesses yield a valid assignment $\Pi_{V'}$ and store the number of unsatisfied agents. Finally, we select one valid $\Pi_{V'}$ with fewest unsatisfied agents. The whole approach can be done in FPT time wrt. n.

Parameter m: We guess the congestion vector $\vec{s}$ with $\vec{s}[j] \in \{0, \ldots, n\}$ and $\Sigma\vec{s} = n$ and use the algorithm behind Lemma 2 to determine the minimum number of unsatisfied agents. The overall running time is $n^m \cdot (m+n)^{O(1)}$, which is XP wrt. m. □

D.5 Proof of Theorem 6

Theorem 6 ($\star$)**.** *Deciding whether an instance of* Congested Assignment *has an assignment with at most k unsatisfied agents is W[1]-hard.*

Proof. We reduce from the W[1]-complete problem Clique.

Let $I = (G = (U, E), h)$ denote an instance of Clique with $U = \{u_1, \ldots, u_{\hat{n}}\}$ and $E = \{e_1, \ldots, e_{\hat{m}}\}$ being the vertex set and edge set, respectively. Without loss of generality, we assume that $\hat{n} > 3h + \binom{h}{2}$ and $\hat{m} > 2h + 2\binom{h}{2}$ as the problem remains W[1]-hard in this case.

The idea is to construct an instance $I' = (\mathcal{A}, V, (\succeq_v)_{v \in V})$ of Congested Assignment such that the unsatisfied agents correspond to the vertices and edges of a size-h clique. We set the number of unsatisfied agents to $k = 2h + \binom{h}{2}$, and let L and R be two very large numbers such that $L > 2k$ and $R > (L+2) \cdot (\hat{n} + \hat{m}) + h$. For the sake of brevity, let $N = (L+2) \cdot (\hat{n} + \hat{m}) + 2R$, and we will create exactly N agents.

Posts and agents.

- For each vertex $u_i \in U$, create one *vertex-post* a_i and $L+2$ *vertex-agents* $w_i, p_i, p^z_i, z \in [L]$.
- For each edge $e_\ell \in E$, create one *edge-post* b_ℓ and $L+2$ *edge-agents* $e_\ell, f^z_\ell, z \in [L+1]$.
- Create $2R$ dummy agents $x_z, y_z, z \in [R]$.
- Create 3 auxiliary posts a_0, b_0, and c_0.

Let $A = \{a_i \mid i \in [\hat{n}]\}$, $B = \{b_\ell \mid \ell \in [\hat{m}]\}$, $W = \{w_i \mid i \in [\hat{n}]\}$, $P = \{p_i \mid i \in [\hat{n}]\}$, $P_i = \{p^z_i \mid z \in [L], i \in [\hat{n}]\}$, $F_\ell = \{f^z_\ell \mid z \in [L+1]\}$, $X = \{x_z \mid z \in [R]\}$, and $Y = \{y_z \mid z \in [R]\}$. Then, we set $\mathcal{A} = A \cup B \cup \{a_0, b_0, c_0\}$, and $V = W \cup P \cup \bigcup_{i \in [\hat{n}]} P_i \cup E \cup \bigcup_{\ell \in [\hat{m}]} F_\ell \cup X \cup Y$. In total, we have created $\hat{n} + \hat{m} + 3$ posts and N agents.

Preferences. For two numbers $s, t \in [N]$ and post $\alpha \in \mathcal{A}$, let $\langle \alpha, s, t \rangle = (\alpha, s) \succ (\alpha, s+1) \succ \dots \succ (\alpha, t)$ depict the preference list on tuples for post α and congestions ranging between s and t. The notation "$* * *$" refers to an arbitrary but congestion-averse preferences of the tuples that are not explicitly mentioned.

(i) For each vertex $u_i \in U$, the vertex-agents $w_i \in W$, $p_i \in P$, and $p_i^z \in P_i$, $z \in [L]$, have the following preference lists:

$$\begin{aligned}
w_i \colon & \langle a_0, 1, \hat{n} - h \rangle \succ \langle a_i, 1, L+2 \rangle \succ \langle c_0, 1, N \rangle \succ * * * \\
& \succ \langle b_0, 1, N \rangle \succ \langle a_0, \hat{n} - h + 1, N \rangle, \\
p_i \colon & \langle a_i, 1, L+1 \rangle \succ \langle a_0, 1, \hat{n} - h \rangle \succ (a_i, L+2) \succ \\
& \langle c_0, 1, N \rangle \succ * * * \succ \langle b_0, 1, N \rangle \succ \langle a_0, \hat{n} - h + 1, N \rangle, \\
p_i^z \colon & \langle a_i, 1, L+2 \rangle \succ \langle c_0, 1, N \rangle \succ * * * \succ \langle b_0, 1, N \rangle \succ \\
& \langle a_0, 1, N \rangle.
\end{aligned}$$

(ii) For each edge $e_\ell \in E$, the edge-agents e_ℓ, f_ℓ^z, $z \in [L+1]$, have the following preference lists, where we assume $e_\ell = \{u_i, u_j\}$:

$$\begin{aligned}
e_\ell \colon & \langle b_0, 1, \hat{m} - \binom{h}{2} \rangle \succ \langle b_\ell, 1, L+2 \rangle \succ \langle c_0, 1, N \rangle \succ \\
& * ** \succ \langle b_0, \hat{m} - \binom{h}{2} + 1, N \rangle \succ \langle a_0, 1, N \rangle. \\
f_\ell^z \colon & \langle b_\ell, 1, L+1 \rangle \succ \langle a_i, 1, L+1 \rangle \succ \langle a_j, 1, L+1 \rangle \succ \\
& (b_\ell, L+2) \succ \langle c_0, 1, N \rangle \succ * * * \succ \langle b_0, 1, N \rangle \succ \\
& \langle a_0, 1, N \rangle.
\end{aligned}$$

(iii) The preference lists of the dummy agent $x_z \in X$ and $y_z \in Y$ are as follows:

$$\begin{aligned}
x_z \colon & \langle a_0, 1, \hat{n} - h - 1 \rangle \succ \langle a_1, 1, L \rangle \succ \dots \succ \langle a_{\hat{n}}, 1, L \rangle \succ \\
& \langle c_0, 1, 2R \rangle \succ \langle a_1, L+1, N \rangle \succ \dots \succ \langle a_{\hat{n}}, L+1, N \rangle \succ \\
& \langle c_0, 2R+1, N \rangle \succ * * * \succ \langle b_0, 1, N \rangle \succ \langle a_0, \hat{n} - h, N \rangle \succ \\
& \langle a_0, 1, N \rangle. \\
y_z \colon & \langle b_0, 1, \hat{m} - \binom{h}{2} - 1 \rangle \succ \langle b_1, 1, L \rangle \succ \dots \succ \langle b_{\hat{m}}, 1, L \rangle \succ \\
& \langle c_0, 1, N \rangle \succ * * * \succ \langle b_0, \hat{m} - \binom{h}{2}, N \rangle \succ \langle a_0, 1, N \rangle.
\end{aligned}$$

This completes the construction of the instance I', which can clearly be done in polynomial time. Note that it is also a parameterized reduction since the parameter $k = 2h + \binom{h}{2}$ is a polynomial function in h. It remains to show the correctness, i.e., I has a size-h clique if and only if I' has an assignment with at most k unsatisfied agents.

For the **"only if"** part, let U' be a clique of size h. We construct the following assignment Π and show that it has at most k unsatisfied agents.

(1) For each vertex $u_i \in U$, assign all agents from $P_i \cup \{p_i\}$ to a_i. Additionally assign w_i to a_i if $u_i \in U'$; otherwise assign w_i to a_0.
(2) For each edge $e_\ell \in E$, assign all agents from F_ℓ to b_ℓ. Additionally assign e_ℓ to b_ℓ if $e_\ell \subseteq U'$, i.e., both its endpoints are in U'; otherwise assign e_ℓ to b_0.
(3) Assign all agents from $X \cup Y$ to c_0.

To see who is unsatisfied, let $W' = \{w_i \in W \mid w_i \in \Pi(a_i)\}$, $P' = \{p_i \in P \mid p_i \in \Pi(a_i)\}$, and $E' = \{e_\ell \in E \mid e_\ell \in \Pi(b_\ell)\}$. We claim that all agents but those from $W' \cup P' \cup E'$ are satisfied.

In the following, we say that a post α is the *most preferred post* for agent q if every tuple that is contains a post other than α is less preferred than $(\alpha, |\Pi(\alpha)|)$. Further, a tuple (α, d) is a *most preferred feasbile* tuple for agent q if every tuple (α', d') that is preferred to (α, d) has congestion $|\Pi(\alpha')| > d'$.

Clearly, every agent in $X \cup Y$ is satisfied since $(c_0, 2R)$ is his most preferred *feasible* tuple. Every agent in $(\bigcup_{i \in [\hat{n}]} P_i) \cup (W \setminus W') \cup (P \setminus P') \cup (E \setminus E')$ is satisfied since he is assigned to his most preferred post. Every agent $f_\ell^z \in F_\ell$ is also satisfied since either he is assigned to his most preferred post (if e_ℓ is not a "clique" edge) or $|\Pi(a_i)| = |\Pi(a_j)| = L + 2$ so $(b_\ell, L + 2)$ remains his most preferred feasible tuple. This concludes the proof for the "only if" direction.

For the **"if"** direction, let Π denote an assignment with at most k unsatisfied agents. Before we construct a clique, let us analyze the preferences and Π would look like.

Claim D.5.1. Π *satisfies the following.*

(1) $|\Pi(a_0)| = \hat{n} - h$ *and* $|\Pi(b_0)| = \hat{m} - \binom{h}{2}$.
(2) Every agent from W *that is* not *assigned to* a_0 *is unsatisfied and every agent from* E *that is* not *assigned to* b_0 *is unsatisfied.*
(3) For each $a_i \in A$ *we have that* $|\Pi(a_i)| \geq L+1$ *and for each* $b_\ell \in B$ *we have that* $|\Pi(b_\ell)| \geq L+1$.
(4) It holds that $|\Pi(c_0)| \leq 2R$.
(5) For each $a_i \in A$ *we have that* $|\Pi(a_i)| \leq L+2$ *and for each* $b_\ell \in B$ *we have that* $|\Pi(b_\ell)| \leq L+2$.

Proof. Statement (1)*:* The lower bounds are straightforward since all agents from $W \cup X$ prefer $(a_0, \hat{n} - h - 1)$ to any other tuple that does not contain a_0: If $|\Pi(a_0)| < \hat{n} - h$ would hold, then more than $|W \cup X| - (\hat{n} - h) > R$ agents will be unsatisfied, which is not possible since $R > (L + 2) \cdot (\hat{n} + \hat{m}) > k$. Similar reasoning shows that $|\Pi(b_0)| \geq \hat{m} - \binom{h}{2}$ by considering the preferences of $E \cup Y$. Now, we show the upper bounds. Suppose, for the sake of contradiction, that $|\Pi(a_0)| > \hat{n} - h$. Then, since no agent considers $(a_0, \hat{n} - h + 1)$ more valuable than any other tuple that does not contain a_0, all agents assigned to a_0 are unsatisfied. Since we can assume that $\hat{n} > 3h + \binom{h}{2}$, it follows that more than k agents will be unsatisfied, a contradiction. Similarly, since we have just shown that $|\Pi(a_0)| \leq \hat{n} - h$, from the remaining possible tuples, no agent considers $(b_0, \hat{m} - \binom{h}{2})$ more valuable than any tuple that does not contain b_0 (except $(a_0, \hat{n} - h + z)$, $z \geq 1$, which is excluded). Consequently, by the fact that $\hat{m} > 2h + 2\binom{h}{2}$, we infer that $|\Pi(b_0)| \leq \hat{m} - \binom{h}{2}$ as otherwise all agents assigned to b_0 are unsatisfied the number of which exceeds k.

Statement (2)*:* This statement follows directly from the previous statement and from the preferences of the agents in $W \cup E$.

Statement (3): We show the lower bound by iterating through all $i \in [\hat{n}]$. By Statement (1), every agent in $X \cup P_1 \cup \{p_1\}$ prefers (a_1, L) to every other tuple that does not contain a_1 (excluding a_0). Hence, $|\Pi(a_1)| \geq L+1$ as otherwise more than $R - L > k$ agents from X are not assigned to a_1 and will be unsatisfied. By applying the above reasoning for the next $i \geq 2$, we infer that $|\Pi(a_i)| \geq L+1$ holds for all $i \in [\hat{n}]$. Similarly, we infer that $|\Pi(b_\ell)| \geq L + 1$ holds for every $\ell \in [\hat{m}]$.

Statement (4)*:* Suppose this is not true, i.e., $|\Pi(c_0)| \geq 2R + 1$. Then, by Statements (1)–(2) and by the bound $k = 2h + \binom{h}{2}$, at most h agents from $V \setminus (W \cup E)$ can be unsatisfied. By construction, every agent in X that is assigned to c_0 will be unsatisfied since he prefers $(a_1, L + 1)$ to $(c_0, 2R + 1)$. Hence, at most h agents from X can be assigned to c_0. This means that at least $2R + 1 - h$ agents from $W \cup P \cup \bigcup_{i \in [\hat{n}]} P_i \cup E \cup \bigcup_{\ell \in [\hat{m}]} F_\ell \cup Y$ need to be assigned to c_0. This is not possible however since $R > (L + 2) \cdot (\hat{n} + \hat{m}) + h$ and $|Y| = R$.

Statement (5)*:* Let $i \in [\hat{n}]$. The statement follows directly from the fact that every agent prefers $(c_0, 2R)$ to $(a_i, L + 3)$ and $|\Pi(c_0)| \leq 2R$ (see Claim D.5.1(1)): $|\Pi(a_i)| > L + 2$ would hold, then all agents assigned to a_i are unsatisfied, the number of which exceed k since $L > 2k$. (end of the proof of Claim D.5.1 $\diamond$)

The next statement is about the structure of the agents assigned to $A \cup B$.

Claim D.5.2. *Let* $A' = \{a_i \in A \colon |\Pi(a_i)| = L + 2\}$ *and* $B' = \{b_\ell \in B \colon |\Pi(b_\ell)| = L + 2\}$. *Then,* Π *satisfies the following.*

(1) $|A'| + |B'| \geq h + \binom{h}{2}$.

(2) For each post $a_i \in A'$, at least two agents in $\Pi(a_i)$ are unsatisfied; for each post $b_\ell \in B'$, at least one agent in $\Pi(b_\ell)$ is unsatisfied.

(3) $|A'| \le h$ and $|B'| \ge \binom{h}{2}$.

(4) For each post $b_\ell \in B'$ it holds that $|\Pi(a_i)| = |\Pi(a_j)| = L+2$ where $e_\ell = \{u_i, u_j\}$.

Proof. Statement (1)*:* This can be shown by simple calculation. By Claim D.5.1(1) and (4), at least $(L+2)\cdot(\hat{n}+\hat{m}) - (\hat{n}-h) - (\hat{m}-\binom{h}{2}) = (L+1)\cdot(\hat{n}+\hat{m}) + h + \binom{h}{2}$ agents are assigned to the posts of $A \cup B$. By Claim D.5.1(3) and (5), each post in $A \cup B$ is assigned either $L+1$ or $L+2$ agents. That is, at least $h + \binom{h}{2}$ of the post in $A \cup B$ are each assigned $L+2$ agents, confirming that $|A'| + |B'| \ge h + \binom{h}{2}$.

Statement (2)*:* For each post $a_i \in A'$, since $|\Pi(a_i)| = L+2$ and $|P_i| = L$, at least two agents in $\Pi(a_i)$ are not from P_i, i.e., $|\Pi(a_i) \setminus P_i| \ge 2$ We claim that the agents in $\Pi(a_i) \setminus P_i$ are unsatisfied by considering the preferences of all agents except P_i: Every agent from $W \cup P$ prefers $(a_0, \hat{n}-h)$ to $(a_i, L+2)$. Every agent from E prefers $(b_0, \hat{m} - \binom{h}{2})$ to $(a_i, L+2)$. Every agent from $(\bigcup_{i' \in [\hat{n}]\setminus\{i\}} P_i) \cup (\bigcup_{\ell \in [\hat{m}]} F_\ell) \cup X \cup Y$ prefers $(c_0, 2R)$ to $(a_i, L+2)$. Since $|\Pi(a_0)| = \hat{n} - h$, $|\Pi(b_0)| = \hat{m} - \binom{h}{2}$, and $|\Pi(c_0)| \le 2R$ (see Claim D.5.1s(1) and (4)), we infer that every agent in $\Pi(a_i) \setminus P_i$ is unsatisfied.

Similarly, for each post $b_\ell \in B'$, since $|\Pi(b_\ell)| = L+2$ and $|F_\ell| = L+1$, at least one agent in $\Pi(b_\ell)$ is not from F_ℓ. We claim that the agents in $\Pi(b_\ell) \setminus F_\ell$ are unsatisfied by considering the preferences of all agents except P_i: Every agent from $W \cup P \cup (\bigcup_{i \in [\hat{n}]} P_i) \cup (\bigcup_{\ell' \in [\hat{m}]\setminus\{\ell\}} F_\ell) \cup X \cup Y$ prefers $(c_0, 2R)$ to $(b_\ell, L+2)$. Every agent from E prefers $(b_0, \hat{m} - \binom{h}{2})$ to $(b_\ell, L+2)$. Again, since $|\Pi(b_0)| = \hat{m} - \binom{h}{2}$ and $|\Pi(c_0)| \le 2R$ (see Claim D.5.1(1) and (4)), we infer that every agent in $\Pi(b_\ell) \setminus F_\ell$ is unsatisfied.

Statement (3)*:* Statement (2) implies that at least $2|A'| + |B'|$ agents are unsatisfied. By the upper bound that $k \le 2h + \binom{h}{2}$ and by Statement (1), we infer that $|A'| \le h$, and hence $|B'| \ge \binom{h}{2}$.

Statement (4)*:* Suppose, towards a contradiction, that $|\Pi(a_i)| \ne L+2$. Then, by Claim D.5.1(3) and (5), it follows that $|\Pi(a_i)| = L+1$. We claim that every agent assigned to b_ℓ is unsatisfied. Let us consider an arbitrary agent $q \in \Pi(b_\ell)$. Clearly, if $q \in W \cup P \cup \bigcup_{\ell \in [\hat{m}]} P_\ell \cup X \cup Y \cup E \cup F \setminus (\{e_\ell\} \cup F_\ell)$, then he is unsatisfied since he prefers $(c_0, 2R)$ to $(b_\ell, L+2)$. If $q = e_\ell$, then he is unsatisfied since he prefers $(b_0, \hat{m} - \binom{h}{2})$ to $(b_\ell, L+2)$, while if $q \in F_\ell$, then he is unsatisfied since he prefers $(a_i, L+1)$ to $(b_\ell, L+2)$ as well. This concludes the proof that every agent in $\Pi(b_\ell)$ is unsatisfied, implying that more than $L > 2k$ agents is unsatisfied, a contradiction.

Using an analogous reasoning, we can show that $|\Pi(a_j)| = L + 2$. (end of the proof of Claim D.5.2 $\diamond$)

Now, we are ready to show the existence of a size-h clique. By Claim D.5.2(3), B' corresponds to at least $\binom{h}{2}$ edges. Hence, there are at least h vertices incident to any edge corresponding to B'. For each vertex a_i that is "incident" to any edge-post in B', we know by Claim D.5.2(4) that its corresponding vertex-post a_i must be assigned $L+2$ posts. By Claim D.5.2(3), there are at most h such vertex-posts. Hence, there are exactly h vertex-posts that are each assigned $L+2$ agents, and this is possible if and only if they form a size-h clique. ☐