1 Proof of Prop. 4.3

1.1 Statement of Prop. 4.3

DP-PartRobTF is $\mathbf{\Sigma_2^P}$-complete.

1.2 Proof that DP-PartRobTF is in $\mathbf{\Sigma_2^P}$

Consider the following algorithm:

1. Guess a set of agents $T \subseteq A$;
2. Check that T is efficient and $f(T) \leq c$;
3. Check using an NP-oracle that there does not exist a team $T' \subseteq T$ such that $|T'| \leq k$ and $cov(T \setminus T') < t$;

Obviously enough, this non-deterministic algorithm with a NP oracle runs in polynomial time and decides DP-PartRobTF, which shows that DP-PartRobTF is in $\mathbf{\Sigma_2^P}$.

1.3 Proof that the problem MAXMIN-SAT is $\mathbf{\Pi}_2^P$-hard

Let us first consider the following decision problem, MINMAX-SAT [1]:

Definition 1 (MINMAX-SAT)

- ***Input:*** *A tuple $\langle X, Y, \varphi, p\rangle$, where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are two disjoint sets of propositional atoms, φ is a 3-CNF propositional formula such that $Var(\varphi) = X \cup Y$, and p is a non-negative integer.*
- ***Question:*** *For every truth-assignment to X, is there a truth-assignment to Y making at least p clauses in φ true?*

MINMAX-SAT has been proven to be $\mathbf{\Pi}_2^P$-hard in [1], where $\mathbf{\Pi}_2^P = co\mathbf{\Sigma}_2^P$.

We now consider a variant of the MINMAX-SAT problem, which we call MAXMIN-SAT:

Definition 2 (MAXMIN-SAT)

- ***Input:*** *A tuple $\langle X, Y, \varphi, p\rangle$, where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are two disjoint sets of propositional atoms, φ is a 3-CNF propositional formula such that $Var(\varphi) = X \cup Y$, and p is a non-negative integer.*
- ***Question:*** *For every truth-assignment to X, is there a truth-assignment to Y making **at most** p clauses in φ true?*

In the first part of this proof, we intend to show that MAXMIN-SAT is $\mathbf{\Pi}_2^P$-hard, by providing a polynomial-time reduction to it from MINMAX-SAT.

The reduction is defined as follows. Let $\langle X, Y, \varphi, p\rangle$ be an instance of MINMAX-SAT, i.e., $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are two disjoint sets of propositional atoms, φ is a 3-CNF formula consisting of q clauses such that $Var(\varphi) = X \cup Y$, and p is a non-negative integer. The formula φ can be viewed as a set of clauses written as (l_i, l_j, l_k), where l_i, l_j, l_k are literals from $X \cup Y$. With each clause $c_r \in \varphi$ we associate two fresh propositional atoms z_1^r, z_2^r and define the set $Z = \{z_1^r, z_2^r \mid c_r \in \varphi\}$ (note that Z is disjoint from X and Y). Now, for each clause $c_r = (l_i, l_j, l_k)$ from φ we associate the set of three clauses $C_r = \{(\overline{l_i}, z_1^r, z_2^r), (\overline{l_j}, z_1^r, \overline{z_2^r}), (\overline{l_k}, \overline{z_1^r}, z_2^r)\}$. Lastly, let us define the 3-CNF formula α made of the set of clauses $\bigcup_{c_r \in \varphi} C_r$. Note that $Var(\alpha) = X \cup Y \cup Z$.

Let us show that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MINMAX-SAT if and only if $\langle X, Y \cup Z, \alpha, |\alpha| - p\rangle$ is a "yes" instance for MAXMIN-SAT, where $|\alpha|$ is the number of clauses in α.

(Only if part) Assume that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MINMAX-SAT. Let ω_X be any assignment of X. Since $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MAXMIN-SAT, this means that there exists an assignment ω_Y of Y such that the assignment $\omega_X \cup \omega_Y$ makes at least p clauses in φ true. Now, for each clause $c_r = (l_i, l_j, l_k)$ from φ that is made true by the assignment $\omega_X \cup \omega_Y$, let us define the assignment ω_Z^r of the two variables z_1^c, z_2^c as follows. Since at least one of the literals l_i, l_j, l_k is true in c_r, if l_i is true in c_r, one sets $z_1^c = z_2^c = 0$; otherwise if l_j is true in c_r, one sets $z_1^c = 0$ and $z_2^c = 1$; and otherwise, if l_k is true in c_r, one sets $z_1^c = 1$ and $z_2^c = 0$. Doing so, one can verify that the assignment $\omega_X \cup \omega_Y \cup \omega_Z^r$ makes at least one clause from C_r false. Thus for each clause $c_r \in \varphi$ that is made true by the assignment $\omega_X \cup \omega_Y$, one can find an assignment ω_Z of Z so that the assignment $\omega_X \cup \omega_Y \cup \omega_Z$ makes one clause from C_r false[1]. Yet we know that the assignment $\omega_X \cup \omega_Y$ makes at least p clauses in φ true. Thus the assignment $\omega_X \cup \omega_Y \cup \omega_Z$ makes at least p clauses from C_r false, or equivalently it makes at most $|\alpha| - p$ clauses from C_r true. This means that $\langle X, Y \cup Z, \alpha, |\alpha| - p\rangle$ is a "yes" instance for MAXMIN-SAT.

(If part) Assume now that $\langle X, Y, \varphi, p\rangle$ is a "no" instance for MINMAX-SAT. So let ω_X be an assignment of X, then we know that for any assignment ω_Y of Y, the assignment $\omega_X \cup \omega_Y$ makes at most $p - 1$ clauses true in φ. Now, let c_r be any clause from φ, and let ω_Y be any assignment of Y. One can easily see that for any assignment ω_Z^r of the two variables z_1^c, z_2^c, the assignment $\omega_X \cup \omega_Y \cup \omega_Z^r$ (i) makes at most one clause from C_r false if c_r is made true by $\omega_X \cup \omega_Y$, (ii) makes no clause from C_r false if c_r is made false by $\omega_X \cup \omega_Y$. But since we know that for any assignment ω_Y of Y, the assignment $\omega_X \cup \omega_Y$ makes at most $p - 1$ clauses true in φ, this means that for any assignment ω_Y of Y and for any assignment ω_Z of Z, the assignment $\omega_X \cup \omega_Y \cup \omega_Z$ makes at most $p - 1$ clauses from C_r false, or equivalently it makes at least $|\alpha| - p + 1$ clauses from C_r true. This means that $\langle X, Y \cup Z, \alpha, |\alpha| - p\rangle$ is a "no" instance for MAXMIN-SAT.

We have shown that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MINMAX-SAT if and only if $\langle X, Y \cup Z, \alpha, |\alpha| - p\rangle$ is a "yes" instance for MAXMIN-SAT. Since

[1]Note that all pairs of sets $\{z_1^r, z_2^r\}$ and $\{z_1^{r'}, z_2^{r'}\}$ are pairwise disjoint when $r \neq r'$, so that all assignments $\{\omega_Z^r \mid c_r \in \varphi\}$ can be defined independently of each other

MINMAX-SAT is $\mathbf{\Pi}_2^P$-hard, this proves that MAXMIN-SAT is $\mathbf{\Pi}_2^P$-hard.

1.4 Proof that DP-PR-TF is $\mathbf{\Sigma}_2^P$-hard

We intend to show that DP-PR-TF is $\mathbf{\Sigma}_2^P$-hard, by providing a polynomial-time reduction to its complementary problem from MAXMIN-SAT.

Let $\langle X, Y, \varphi, p\rangle$ be an instance of MAXMIN-SAT, i.e., $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ are two disjoint sets of propositional atoms, φ is a 3-CNF formula consisting of q clauses such that $Var(\varphi) = X \cup Y$, and p is a non-negative integer. Note that without loss of generality, we have here $|X| = |Y| = n$. Assume also without loss of generality that $p < |\varphi|$ (the case where $p = |\varphi|$ makes the instance trivially a "yes" one).

Let us associate with it a set of agents A, a set of skills S, a deployment cost function $f : A \mapsto \mathbb{N}$, a skill weight function $w : 2^S \mapsto [0,1]$, and a skill-to-agent function $\beta : S \mapsto 2^A$. Note that these objects are not exactly the components of a weighted TF problem description, since one considers a skill-to-agent function $\beta : S \mapsto 2^A$ instead of an agent-to-skill function $\alpha : A \mapsto 2^S$. Intuitively, the function β associates with every skill from S the set of agents that possess the skill. This is made for simplicity in the reduction, however given A and S, an agent-to-skill function α can simply be derived from β as $\alpha(a) = \{s \in S \mid a \in \beta(s)\}$ for every agent $a \in A$. Then for instance, a skill $s \in S$ is covered by a team $T \subseteq A$ if and only if there is an agent $a \in T$ such that $a \in \beta(s)$; and a team is efficient if for all skills $s \in S$, $\beta(s) \cap T \neq \emptyset$.

Let us now define these objects in detail.

We define the set A of $4n + 1$ agents as $A = \{a_0\} \cup \{a_i, \overline{a_i}, b_i, \overline{b_i} \mid i \in \{1, \ldots, n\}\}$.

The cost function f is defined as $f(\{a_0\}) = 0$, and for every agent $a \in A \setminus \{a_0\}$, $f(a) = 1$.

The set S is formed of $4n + |\varphi|$ skills, where $|\varphi|$ is the number of clauses in φ, and is divided in two parts $S = S^* \cup S^\varphi$, with $|S^*| = 4n$ and $|S^\varphi| = \varphi$: the set S^φ depends on the clauses of φ, as opposite to the set S^* which only depends on A. So S^* is defined as $S^* = \{s_i^I, s_i^{II}, s_i^{III}, s_i^{IV} \mid i \in \{1, \ldots, n\}\}$, and S^φ is defined as $S^\varphi = \{s_1^\varphi, \ldots, s_q^\varphi\}$, where $q = |\varphi|$.

The skill weight function $w : 2^S \mapsto [0,1]$ is defined as follows. For every skill $s \in S$, one sets $w(s) = 1/|S|$. In addition, for every subset of skills $S' \subseteq S$, one defines $w(S') = 1$ if there exists $i \in \{1, \ldots, n\}$ such that $\{a_i, b_i\} \subseteq S'$ or $\{\overline{a_i}, \overline{b_i}\} \subseteq S'$, or if $S^\varphi \subseteq S'$; otherwise $w(S') = \sum_{s \in S'} w(s)$.

Lastly, the skill-to-agent function $\beta : S \mapsto 2^A$ is defined as follows. For each $i \in \{1, \ldots, n\}$:

- $\beta(s_i^I) = \{a_i, \overline{a_i}\}$
- $\beta(s_i^{II}) = \{b_i, \overline{b_i}\}$
- $\beta(s_i^{III}) = \{a_i, \overline{b_i}\}$
- $\beta(s_i^{IV}) = \{\overline{a_i}, b_i\}$.

And for each skill $s_r^\varphi \in S^\varphi$, one identifies $\beta(s_r^\varphi)$ depending on the clause $c_r = (l_i, l_j, l_k)$ from φ. Beforehand, let us first consider the mapping γ associating any literal over $X \cup Y$ with a pair of elements of A, defined for every (possibly negated) literal l_i as

$$\begin{array}{rl} \gamma(l_i) = & \{a_i, b_i\} \text{ if } l_i \text{ is a positive literal over } X, \\ & \{\overline{a_i}, \overline{b_i}\} \text{ if } l_i \text{ is a negative literal over } X, \\ & \{a_i, \overline{a_i}\} \text{ if } l_i \text{ is a positive literal over } Y, \\ & \{b_i, \overline{b_i}\} \text{ if } l_i \text{ is a negative literal over } Y. \end{array}$$

Now, for each clause $c_r = (l_i, l_j, l_k)$ from φ, we define $\beta(s_r^\varphi)$ as $\beta(s_r^\varphi) = \{a_0\} \cup \gamma(l_i) \cup \gamma(l_j) \cup \gamma(l_k)$.

Example 1 *For the sake of illustration, let us give an example of how the skill-to-agent function $\beta : S \mapsto 2^A$ is constructed from an instance $\langle X, Y, \varphi, p\rangle$ of* MAXMIN-SAT*, for skills from S^φ. Let $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4\}$, and φ is formed of the set of four clauses $\{(x_1, x_2, \overline{x_3})$, $(\overline{x_1}, x_4, \overline{y_1})$, $(x_2, y_2, \overline{y_3})$, $(y_1, \overline{y_2}, y_3)\}$. Since φ has four clauses, S^φ is formed of four skills $s_1^\varphi, s_2^\varphi, s_3^\varphi, s_4^\varphi$ (one skill for each clause from φ), and for each one of these skills s_i^φ, $\beta(s_i^\varphi)$ is defined as follows:*

$$\begin{array}{ll} \beta(s_1^\varphi) = \{a_0, a_1, b_1, a_2, b_2, \overline{a_3}, \overline{b_3}\} & \textit{(clause } (x_1, x_2, \overline{x_3})\textit{)} \\ \beta(s_2^\varphi) = \{a_0, \overline{a_1}, \overline{b_1}, a_4, b_4, b_1, \overline{b_1}\} & \textit{(clause } (\overline{x_1}, x_4, \overline{y_1})\textit{)} \\ \beta(s_3^\varphi) = \{a_0, a_2, b_2, \overline{a_2}, b_3, \overline{b_3}\} & \textit{(clause } (x_2, y_2, \overline{y_3})\textit{)} \\ \beta(s_3^\varphi) = \{a_0, a_1, \overline{a_1}, b_2, \overline{b_2}, a_3, \overline{a_3}\} & \textit{(clause } (y_1, \overline{y_2}, y_3)\textit{).} \end{array}$$

Let us associate now the skill-to-agent function β with the agent-to-skill function $\alpha : A \mapsto 2^S$ as $\alpha(a) = \{s \in S \mid a \in \gamma(s)\}$ for every agent $a \in A$. So we have associated with any instance $\langle X, Y, \varphi, p\rangle$ of MAXMIN-SAT a weighted TF problem description $\langle A, S, f, w, \alpha\rangle$ (with in addition β serving as an intermediate function to characterize α).

Let us now show that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance of MAXMIN-SAT if and only if there does not exist a $\langle k, t\rangle$-partially robust team $T \subseteq A$ such that T is efficient and $f(T) \leq c$, with $k = n + 1$, $t = (2n + p + 1)/|S|$, and $c = 2n$.

(Only if part) Assume that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MAXMIN-SAT. So for any assignment ω_X of X, there exists an assignment ω_Y of Y such that the assignment $\omega_X \cup \omega_Y$ makes at most p clauses in φ true. Now, let $T \subseteq A$ be any team such that T is efficient and $f(T) \leq 2n$. We need to show that T is not $\langle k, t\rangle$-partially robust, with $k = n + 1$ and $t = (2n + p + 1)/|S|$.

First, let us remark that if $a_0 \notin T$, since T is efficient and $f(T) \leq 2n$, the team $T \cup \{a_0\}$ is also efficient and $f(T \cup \{a_0\}) \leq 2n$; in addition, T is $\langle k, t\rangle$-partially robust only if $T \cup \{a_0\}$ is $\langle k, t\rangle$-partially robust. This means that we can assume that $a_0 \in T$ without any harm. Second, each agent from A except a_0 has a unit cost, i.e., for each $a \in A \setminus \{a_0\}$, $f(a) = 1$. So if $|T \setminus \{a_0\}| = m < 2n$, then any addition of $2n - m$ agents $T' \subseteq A \setminus T$ to T can be done without any harm. That is to say, we still have that $T \cup T'$ is efficient, $f(T \cup T') \leq 2n$, and

T is $\langle k,t\rangle$-partially robust only if $T \cup T'$ is $\langle k,t\rangle$-partially robust. So overall, let us assume that $a_0 \in T$ and $|T \setminus \{a_0\}| = 2n$, and it is enough to prove that T is not $\langle k,t\rangle$-partially robust. Lastly, since T is efficient, it necessarily covers all skills from $S = S^* \cup S^\varphi$. On the one hand, all skills from S^φ are trivially covered by T since $a_0 \in T$ and for each $s_i^\varphi \in S^\varphi$, $a_0 \in \beta(s_i^\varphi)$. On the other hand, all skills from $S^* = \{s_i^I, s_i^{II}, s_i^{III}, s_i^{IV} \mid i \in \{1,\dots,n\}\}$ are covered as well by T. So for each $i \in \{1,\dots,n\}$, $\beta(s_i^I) \cap T \neq \emptyset$, $\beta(s_i^{II}) \cap T \neq \emptyset$, $\beta(s_i^{III}) \cap T \neq \emptyset$, and $\beta(s_i^{IV}) \cap T \neq \emptyset$. By construction of those $\beta(s_i^I)$, $\beta(s_i^{II})$, $\beta(s_i^{III})$, $\beta(s_i^{IV})$, for $i \in \{1,\dots,n\}$, and since $|T \setminus \{a_0\}| = 2n$, it means that for each $i \in \{1,\dots,n\}$, one has either (i) $\{a_i, b_i\} \subseteq T$ and $\{\overline{a_i}, \overline{b_i}\} \cap T = \emptyset$, either (ii) $\{\overline{a_i}, \overline{b_i}\} \subseteq T$ and $\{a_i, b_i\} \cap T = \emptyset$.

Let us now show that T is not $\langle k,t\rangle$-partially robust, i.e., one can find a set $T' \subseteq T$, $|T'| \leq k$, and such that $cov(T \setminus T') < t$. Let us now define the assignment ω_X of X from T as follows: for each $i \in \{1,\dots,n\}$, $\omega_X(x_i) = 1$ if and only if $\{a_i, b_i\} \subseteq T$. In particular, from this definition of ω_X and because of the structure of T we know that (i) $\omega_X(x_i) = 1$ if and only if ($\{a_i, b_i\} \subseteq T$ and $\{\overline{a_i}, \overline{b_i}\} \cap T = \emptyset$), and (ii) $\omega_X(x_i) = 0$ if and only if ($\{\overline{a_i}, \overline{b_i}\} \subseteq T$ and $\{a_i, b_i\} \cap T = \emptyset$). Yet we know that $\langle X, Y, \varphi, p\rangle$ is a "yes" instance for MAXMIN-SAT. This means that there is an assignment ω_Y of Y such that the assignment $\omega_X \cup \omega_Y$ makes at most p clauses in φ true. We associate with such an assignment ω_Y a set of agents T' to remove from T as follows. First, let $a_0 \in T'$ (i.e., one removes a_0 from T). Second, for each $i \in \{1,\dots,n\}$, if $\omega_X(y_i) = 1$ then one removes either a_i or $\overline{a_i}$ from T depending on whether a_i or $\overline{a_i}$ is in T; and if $\omega_X(y_i) = 0$ then one removes either b_i or $\overline{b_i}$ from T depending on whether b_i or $\overline{b_i}$ is in T. At this stage, we can remark that (i) T' contains a_0 and exactly one element of $\{a_i, b_i, \overline{a_i}, \overline{b_i}\}$ for each $i \in \{1,\dots,n\}$; and (ii) $T \setminus T'$ contains exactly one element of $\{a_i, b_i, \overline{a_i}, \overline{b_i}\}$ for each $i \in \{1,\dots,n\}$. Accordingly, $|T'| = n+1$, so $|T'| \leq k$. It remains to show that $cov(T \setminus T') < t$.

By definition of T and T', we have that $T \setminus T'$ covers exactly $2n$ skills from the set S^*. And it can be verified by construction of the $\beta(s_r^\varphi)$, $c_r \in \varphi$, that for each clause c_r from φ, c_r is made true by the assignment $\omega_X \cup \omega_Y$ if and only if $\beta(s_r^\varphi) \cap (T \setminus T') \neq \emptyset$, if and only if the skill s_r^φ is covered by $T \setminus T'$. Thus the number of skills from S^φ that are covered by $T \setminus T'$ is equal to the number of clauses in φ that are made true by $\omega_X \cup \omega_Y$. Yet from the initial assumption, $\omega_X \cup \omega_Y$ makes at most p clauses in φ true. This means that at most p skills from S^φ are covered by $T \setminus T'$. To summarize, since on the one hand $T \setminus T'$ covers exactly $2n$ skills from the set S^*, and on the other hand $T \setminus T'$ covers at most p skills from S^φ, we get that $T \setminus T'$ covers at most $2n + p$ skills from S, i.e., $|\alpha(T \setminus T')| \leq 2n + p$.

Let us compute $w(T \setminus T')$. We already know that $T \setminus T'$ contains exactly one element of $\{a_i, b_i, \overline{a_i}, \overline{b_i}\}$ for each $i \in \{1,\dots,n\}$. So by definition of the skill weight function $w : 2^S \mapsto [0,1]$, we have that $w(\alpha(T \setminus T')) = \sum_{s_j \in T \setminus T'} w(s_j)$: indeed, we do not fall in the case where $w(\alpha(T \setminus T')) = 1$ since for each $i \in \{1,\dots,n\}$, $\{a_i, b_i\} \nsubseteq \alpha(T \setminus T')$ and $\{\overline{a_i}, \overline{b_i}\} \nsubseteq \alpha(T \setminus T')$, and $S^\varphi \nsubseteq \alpha(T \setminus T')$ (recall that p is initially assumed to be strictly lower than $|\varphi| = |S^\varphi|$).

So we got that $|\alpha(T \setminus T')| \leq 2n + p$ and $w(\alpha(T \setminus T')) = \sum_{s_j \in T \setminus T'} w(s_j)$. Thus $\sum_{s_j \in T \setminus T'} w(s_j) = (2n + p)/|S|$. Hence, $cov(T \setminus T') = w(\alpha(T \setminus T')) = (2n + p)/|S|$. Yet $t = (2n + p + 1)/|S|$, thus $cov(T \setminus T') < t$.

We have proved that for any team T such that T is efficient and $f(T) \leq c$, one can find a set $T' \subseteq T$, $|T'| \leq k$, such that $cov(T \setminus T) < t$, with $c = 2n$, $k = n + 1$, and $t = (2n + p + 1)/|S|$. This means that there does not exist a $\langle k, t\rangle$-partially robust team $T \subseteq A$ such that T is efficient and $f(T) \leq c$, with $k = n + 1$, $t = (2n + p + 1)/|S|$, and $c = 2n$. This concludes the (Only if) part of the proof.

(If part) Assume that there does not exist a $\langle k, t\rangle$-partially robust team $T \subseteq A$ such that T is efficient and $f(T) \leq c$, with $k = n+1$, $t = (2n+p+1)/|S|$, and $c = 2n$. Let ω_X be any assignment of X. We need to show that there is an assignment ω_Y of Y such that $\omega_X \cup \omega_Y$ makes at most p clauses in φ true.

Let us associate with ω_X the team $T \subseteq A$ as follows:

$$\begin{aligned} T = \;& \{a_0\} \\ & \cup\{a_i, b_i \mid \omega_X(x_i) = 1, x_i \in X\} \\ & \cup\{\overline{a_i}, \overline{b_i} \mid \omega_X(x_i) = 0, x_i \in X\}. \end{aligned}$$

One can check that T is efficient: all skills from S^φ are covered by a_0, all for each $i \in \{1, \ldots, n\}$:

- the skill s_i^I is covered by either a_i or $\overline{a_i}$;
- the skill s_i^{II} is covered by either b_i or $\overline{b_i}$;
- the skill s_i^{III} is covered by either a_i or $\overline{b_i}$;
- the skill s_i^{IV} is covered by either $\overline{a_i}$ or b_i.

Yet from our initial assumption, we know that T is not $\langle k, t\rangle$-partially robust. This means that there exists a set $T' \subseteq T$, $|T'| \leq k$, such that $cov(T \setminus T') < t$. Yet we know that $t < 1$, since $|S| = 4n+|\varphi|$, $t = (2n+p+1)/|S|$, and we initially assumed that $p < |\varphi|$. So we know that $w(T \setminus T') < 1$, thus by definition of the skill weight function w, this means that:

(i) for each $i \in \{1, \ldots, n\}$, $\{a_i, b_i\} \nsubseteq \alpha(T \setminus T')$ and $\{\overline{a_i}, \overline{b_i}\} \nsubseteq \alpha(T \setminus T')$; and

(ii) $S^\varphi \nsubseteq \alpha(T \setminus T')$.

From (ii) above, since a_0 covers all skills from S^φ and $a_0 \in T$, this means that a_0 must necessary be removed from T and thus T' necessary contains a_0. Yet $|T'| \leq k = n + 1$. So from (i) above and by contruction of T, this means that for each $i \in \{1, \ldots, n\}$, one needs to remove from T exactly one element among $\{a_i, b_i\}$ (in the case where $\{a_i, b_i\} \subseteq T$), or exactly one element among $\{\overline{a_i}, \overline{b_i}\}$ (in the case where $\{\overline{a_i}, \overline{b_i}\} \subseteq T$). So to summarize the structure of T':

- T' contains a_0;

- for each $i \in \{1, \ldots, n\}$, T' contains either exactly one element from $\{a_i, \overline{a_i}\}$, or exactly one element from $\{b_i, \overline{b_i}\}$.

And as a consequence, to summarize the structure of $T \setminus T'$:

- $T \setminus T'$ does not contain a_0;
- for each $i \in \{1, \ldots, n\}$, $T \setminus T'$ contains either exactly one element from $\{a_i, b_i\}$, or exactly one element from $\{\overline{a_i}, \overline{b_i}\}$.

Now, we associate with T' the assignment ω_Y of Y defined for each $i \in \{1, \ldots, n\}$ as $\omega_Y(y_i) = 0$ in the case where $\{a_i, \overline{a_i}\} \cap T' \neq \emptyset$, and thus $\omega_Y(y_i) = 1$ in the other case where $\{b_i, \overline{b_i}\} \cap T' \neq \emptyset$.

At this point, from the sole structure of $T \setminus T'$ we know that for each $i \in \{1, \ldots, n\}$, exactly one skill among $\{s_i^I, s_i^{II}\}$ and exactly one skill among $\{s_i^{III}, s_i^{IV}\}$ is covered by $T \setminus T'$. Thus exactly $2n$ skills from S^* are covered by $T \setminus T'$. And by definition of the skill weight function w, $w(\alpha(T \setminus T')) = \sum_{s \in \alpha(T \setminus T')} w(s) = |\alpha(T \setminus T')|/|S|$. Yet $w(\alpha(T \setminus T')) = cov(T \setminus T') < t = (2n + p + 1)/|S|$. Since $|\alpha(T \setminus T')| \cap S^* = 2n$, thus means that at most p skills from S^φ are covered by $T \setminus T'$, i.e., $|\alpha(T \setminus T')| < p$. Yet it can be verified by construction of $T \setminus T'$ and by definition of $\beta(s_r^\varphi)$ for each clause c_r from φ that $T \setminus T'$ covers a skill s_r^φ if and only if the assignment $\omega_X \cup \omega_Y$ makes the clause c_r true. This precisely means that the assignment $\omega_X \cup \omega_Y$ makes at most p clauses from φ true.

We have proved that for any assignment ω_X of X, there is an assignment ω_Y of Y that makes at most p clauses from φ true. This means that $\langle X, Y, \varphi, p \rangle$ is a "yes" instance for MAXMIN-SAT and concludes the (If) part of the proof.

We have proved that $\langle X, Y, \varphi, p \rangle$ is a "yes" instance of MAXMIN-SAT if and only if there does not exist a $\langle k, t \rangle$-partially robust team $T \subseteq A$ such that T is efficient and $f(T) \leq c$, with $k = n + 1$, $t = (2n + p + 1)/|S|$, and $c = 2n$. This provides a reduction from MAXMIN-SAT to the complementary problem of DP-PR-TF. Yet MAXMIN-SAT is $\mathbf{\Pi}_2^P$-hard. Therefore, DP-PR-TF is $\mathbf{\Sigma}_2^P$-hard.

This concludes the proof of Proposition 4.4.

2 Proof of Prop. 5.1

2.1 Statement of Prop. 5.1

Given a weighted TF problem description $\langle A, S, f, w, \alpha \rangle$, $k \in \mathbb{N}$ and a rational number t, a team $T \subseteq A$ is $\langle k, t \rangle$-partially robust if and only it is efficient and for each $S' \subseteq S$ such that $w(S \setminus S') < t$, we have that $|\{a_i \in T \mid \alpha(a_i) \cap S' \neq \emptyset\}| \geq k + 1$.

2.2 Proof of Prop. 5.1

(Only if) We show the contrapositive of the statement. If $T \subseteq A$ is not efficient, it is trivially not $\langle k, t \rangle$-partially robust. Now, let $S' \subseteq S$, $w(S \setminus S') < t$, let

$T' = \{a_i \in T \mid \alpha(a_i) \cap S' \neq \emptyset\}$ and assume that $|T'| \leq k$. By definition of T', for each agent $a_i \in T \setminus T'$, $\alpha(a_i) \cap S' = \emptyset$. Thus $\alpha(T \setminus T') \subseteq S \setminus S'$. Since $w(S \setminus S') < t$ and w is monotone, $w(\alpha(T \setminus T')) < t$. Hence, $cov(T \setminus T') < t$, and so $pc(T, k) < t$, which precisely means that T is not $\langle k, t\rangle$-partially robust.

(If) We show the contrapositive of the statement. Let $T \subseteq A$, and assume that T is not $\langle k, t\rangle$-partially robust and efficient. By definition, $pc(T, k) < t$, i.e., there exists $T' \subseteq T$, $|T'| \leq k$, $cov(T \setminus T') < t$, so $w(\alpha(T \setminus T')) < t$. Let $S' = S \setminus \alpha(T \setminus T')$. Accordingly, $w(S \setminus S') = w(S \setminus (S \setminus \alpha(T \setminus T'))) = w(\alpha(T \setminus T')) < t$. And by definition of S', for each $a_i \in T \setminus T'$, $\alpha(a_i) \cap S' = \emptyset$. Hence, since $|T'| \leq k$, we get that $|\{a_i \in T \mid \alpha(a_i) \cap S' \neq \emptyset\}| \leq k$.

References

[1] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory*, pages 125–129, 1972.