Randomized Separations in Black-Box TFNP

Fedor Kiselev
MIPT, Moscow, Russia

Abstract

We study the relationship between deterministic and randomized black-box reducibility between problems in TFNP. Our main contribution is a general technique that establishes equivalence between these reducibility types from specific TFNP problems to any TFNP problem. In particular, we show that this equivalence holds for reductions from complete problems in PPP, PPAD, PPA, and t-PPP. In turn, it strengthens all known black-box separations, originating from these classes, to randomized separations.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity classes

Keywords and phrases TFNP, Pigeonhole Principle

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.42

1 Introduction

TFNP is the class of search problems such that for any input there is at least one correct solution, and the correctness of solutions can be checked in polynomial time. It contains many important problems, such as integer factorization and finding a Nash equilibrium. Being a semantic class, TFNP is conjectured to have no complete problems [15]. Consequently, most of the research is focused on studying subclasses of TFNP, defined by complete problems in them. The most well-studied of these classes are PLS [10], PPP, PPA, PPAD, and PPADS [15].

It is straightforward to show that PPAD $\subset$ PPADS $\subset$ PPP and PPAD $\subset$ PPA [15], and it is conjectured that there are no other inclusions between them. However, proving such separations unconditionally would imply P $\neq$ NP. Instead, we can look at black-box analogs of these classes. Since all known inclusions between TFNP classes also hold for their black-box analogs, separations between black-box classes rule out known techniques for showing inclusion.

After years of progress, for every pair of the above mentioned classes, it has been shown that either the inclusion holds unconditionally or it does not hold in the black-box setting [1, 14, 2, 6]. Subsequently, this result was extended to all possible intersections of these classes [7, 13]. Naturally, a question arises whether these black-box separations could be further strengthened. One possible way to do this is to show that these separations persist even if we consider randomized reductions between complete problems in classes instead of deterministic ones. Such stronger separations were achieved for some pairs of classes: in [12] it was shown that in the black-box setting there are no randomized reductions from any of the above mentioned TFNP subclasses to PWPP (introduced in [9]), and in [8] it was shown that there is no randomized reduction from PPP to the newly introduced class n-PWPP. In addition, for most of the studied TFNP classes, it is easy to show that there is no randomized reduction from them to FP. A notable exception is the problem Lossy, introduced in [11], which admits a randomized reduction to FP but no deterministic one. It demonstrates that these two relations do not always coincide. Overall, randomized reducibility in the black-box TFNP remains largely unexplored.

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 42; pp. 42:1–42:17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


1.1 Our Results

We investigate the conditions under which it is possible in the black-box model to strengthen a deterministic separation to a randomized one. Our main result, stated in Theorem 13, shows that for certain well-structured problems, if there is no reduction to some $S \in \mathsf{TFNP}$, there is also no randomized reduction to S. Notably, S is an arbitrary problem in TFNP with no additional restrictions. The proof technique applies to highly symmetric problems. In addition, it requires that, under particular restrictions, the problem can be reduced to a smaller instance of the same problem. Many central TFNP problems possess these required structural properties.

We then demonstrate that all the requirements of Theorem 13 are met for certain complete problems in classes $\mathsf{PPP}, \mathsf{PPA}, \mathsf{PPAD}$ [1], and t-PPP [8]. In particular, it automatically strengthens all known separations originating from them to randomized separations.

1.2 Connection to Proof Complexity

Black-box TFNP has a very tight connection with proof complexity. For many TFNP subclasses it has been shown that they are characterized by some proof system [6]. That is, problem $S \in \mathsf{TFNP}$ is in the subclass if and only if the totality of S has an easy proof in the proof system corresponding to that subclass. Moreover, in [4] it was shown that this connection can be extended to randomized reductions, using randomized versions of proof systems. More specifically, problem $S \in \mathsf{TFNP}$ is randomized-reducible to a complete problem in some subclass if and only if the totality of S has an easy proof in the randomized version of the corresponding proof system. This connection can be used to translate our results to proof complexity. Let us call a proof system well-behaved if it has corresponding proof system (see more on that in [3]). Then for any problem S such that it meets the conditions of Theorem 13, if the totality of S is hard to prove in some well-behaved proof system, then it is also hard to prove in its randomized version.

Finally, it should be mentioned that the result, similar to ours, was achieved in [16], where it was essentially shown that PPP is not randomized-reducible to PLS.

1.3 Paper Organization

The remainder of the paper is structured as follows. In Section 2 we introduce most of the required definitions. In Section 3 we prove our main theorem, as well as a few auxiliary results. In Section 4 we define classes $\mathsf{PPP}, \mathsf{PPAD}, \mathsf{PPA}$ and t-PPP and demonstrate that the main theorem is applicable to them. In Section 5 we briefly discuss limitations of the main theorem and future directions.

2 Preliminaries

2.1 Black-Box TFNP

Black-box classes are usually distinguished from their white-box analogs by using superscript "dt", which stands for "decision tree". Since our work is dedicated to black-box classes only, we will omit this superscript.

▶ **Definition 1.** *A search problem is a sequence $R = \{R_N\}_{N \in \mathbb{N}}$, where $R_N \subset \Sigma_N^{l_N} \times O_N$ for some finite alphabet Σ_N, a finite set of solutions O_N, and input length l_N. Any $x \in \Sigma_N^{l_N}$ is called the input of R_N. A solution o is correct for an input x if $(x, o) \in R_N$. A search problem is considered total if, for any input, there is a correct solution. Search problem R belongs to* TFNP *if it is total and there is a family of decision trees $T = \{T_{N,o}\}_{N \in \mathbb{N}, o \in O_N}$ of depth* $\mathrm{poly}(\log N)$ *such that $(x, o) \in R_N \Leftrightarrow T_{N,o}(x) = 1$.*

Note that this definition is more general than the commonly used one, where $\Sigma_N = \{0,1\}$ and $l_N = N$. We believe that this generalization makes reasoning easier since it allows us to omit the binary encoding of input. All problems defined in Section 4 can be transformed into equivalent problems with $\Sigma_N = \{0,1\}$ and $l_N = N$.

We will sometimes abuse the notation by calling R_N a search problem. Also, if there are several search problems in the context, we will distinguish related objects by subscript letters.

▶ **Definition 2.** *Given search problems R_N and S_M, a pseudo-reduction from R_N to S_M is a pair $(f = \{f_i\}_{i\in[l_M]}, g = \{g_o\}_{o\in O_M})$ of families of decision trees over inputs of R_N. Each f_i defines a mapping $\Sigma_N^{l_N} \to \Sigma_M$; each g_o defines a mapping $\Sigma_N^{l_N} \to O_N$. We say that the pseudo-reduction is successful on input x of R_N if:*

$$\forall o \in O_M : [(x, g_o(x)) \in R_N \Leftarrow (f(x), o) \in S_M],$$

where $f(x) = f_1(x)f_2(x)\dots f_{l_M}(x)$. The depth d of the pseudo-reduction is the maximum depth of decision trees in f and g. The complexity of the pseudo-reduction is $\max(d, \log M)$. *A reduction from R to S is a sequence of pseudo-reductions from R_N to S_M for $M = M(N)$, such that N-th pseudo-reduction is successful on all inputs of R_N and has complexity* $\mathrm{poly}(\log N)$.

Note that pseudo-reduction is not an established term. We introduce it to explicitly distinguish the cases where we have no guarantee that (f, g) has small complexity and is successful on all inputs.

▶ **Definition 3.** *Given search problems R_N and S_M, a randomized pseudo-reduction from R_N to S_M is a distribution D over some finite set of pseudo-reductions from R_N to S_M. The success rate on input x is:*

$$\mathbb{P}_{(f,g)\sim D}[\forall o \in O_M : (x, g_o(x)) \in R_N \Leftarrow (f(x), o) \in S_M].$$

The depth of randomized pseudo-reduction is equal to the maximum depth of pseudo-reductions in D. Complexity is defined in the same way. A randomized reduction from R to S is a sequence of randomized pseudo-reductions from R_N to S_M for $M = M(N)$, such that N-th randomized pseudo-reduction has a success rate of at least $\frac{1}{\mathrm{poly}(\log N)}$ and complexity $\mathrm{poly}(\log N)$.

This definition can be seen as one-round communication with an adversary, who, given an arbitrary x and sampled (f, g), tries to provide worst possible solution for $f(x)$. Note that the adversary knows which (f, g) was sampled, which makes it similar to public coin communication. As far as we know, this definition of randomized reduction between TFNP problems is most common, so we are going to use it. Nonetheless, it would be interesting to see if this result holds for private coin definition.

Also note that some definitions of randomized reductions require a higher success rate threshold. We have chosen one with a weaker requirements. Since we are going to show that there is no randomized reduction, our result will also work for more demanding definitions.

We are interested in proving that there is no randomized reduction. For this purpose, it is more convenient to use an equivalence, which follows from Yao's minimax principle [17].

▶ **Lemma 4.** *There is a randomized reduction from a search problem R to a search problem S if and only if there are $p(N) = \mathrm{poly}(\log N)$ and $M = M(N)$ such that for any N and distribution $\mathcal{X}$ over input of R_N there is a pseudo-reduction (f, g) from R_N to S_M with complexity at most $p(N)$ and success rate at least $\frac{1}{p(N)}$.*

Finally, sometimes we limit our search problem R to some family of inputs $X = \{X_N \subset \Sigma_N^{l_N}\}_{N\in\mathbb{N}}$. We denote it as (R, X). In that case, for any pseudo-reduction (f, g) from (R_N, X_N) to (S_M, Y_M) we require that $x \in X \Rightarrow f(x) \in Y$; also, for a reduction, we require success only on $x \in X_N$.

2.2 Partial Assignment

▶ **Definition 5.** *A partial assignment on inputs of the search problem R_N is any element $\rho \in (\Sigma_N \cup \{*\})^{l_N}$. The size of ρ is $|\rho| = |\{i \mid \rho_i \neq *\}|$. We say that ρ_1 extends ρ_2 (denoted $\rho_1 \sqsupset \rho_2$) if they agree in all non-$*$ positions of ρ_2. Partial assignments ρ_1 and ρ_2 are consistent (denoted $\rho_1 \uparrow\uparrow \rho_2$) if there is an input that extends both of them; consistent relative to $X_N \subset \Sigma_N^{l_N}$ ($\rho_1 \overset{X_N}{\uparrow\uparrow} \rho_2$) if such an input exists in X_N.*

Partial assignments represent information we know about an input of a search problem. They are particularly useful when we are working with decision trees. A partial assignment naturally corresponds to every path from the root to a leaf in a decision tree. This partial assignment consists of answers to all queries made on this path of the decision tree.

▶ **Definition 6.** *Given the search problem R_N and a partial assignment ρ, we say that ρ witnesses a solution $o \in O_N$ if $(x, o) \in R_N$ for all inputs x extending ρ. A partial assignment ρ is called witnessing if it witnesses some solution.*

2.3 Properties of Pseudo-Reductions

In this section, we introduce definitions for honest and perceptive pseudo-reductions. To the best of our knowledge, we are the first to introduce these properties. They are useful for us because they simultaneously are easy to enforce and guarantee a nice structure for a set of inputs on which a pseudo-reduction is not successful.

Without loss of generality, we assume that all studied search problems contain a dedicated always incorrect solution $\perp$. This allows a pseudo-reduction to explicitly show that it does not know a correct solution.

▶ **Definition 7.** *We call a pseudo-reduction (f, g) from R_N to S_M honest if, for all $o' \in O_M$ and $x \in \Sigma_N^{l_N}$, either $g_{o'}(x) = \perp$ or the partial assignment corresponding to the computation of $g_{o'}$ on x witnesses the solution $g_{o'}(x)$.*

Intuitively, this property means that whenever the pseudo-reduction is unable to find a correct solution, it says so instead of trying to guess it. Any pseudo-reduction can be transformed into an honest one with an overhead of $\text{poly}(\log N)$ in depth. To achieve this, for every $o' \in O_M$ and for every leaf in $g_{o'}$, we modify $g_{o'}$ so that it additionally checks if the solution in this leaf is correct, and if not, it returns $\perp$ instead. Since $R \in \mathsf{TFNP}$, this check can be made by a decision tree of $\text{poly}(\log N)$ depth. Note that if the pseudo-reduction was successful on an input x, it will remain so.

▶ **Definition 8.** *Let (f, g) be a pseudo-reduction from R_N to S_M, where $R, S \in \mathsf{TFNP}$. Let us fix some family of decision trees $\mathcal{T}_M = \{T_{M,o}\}_{o\in O_M}$ that checks the correctness of solutions for S_M. We call a pseudo-reduction (f, g) perceptive (with respect to $\mathcal{T}_M$) if:*

1. *For any leaf l in g_o, if the corresponding partial assignment is witnessing, then l returns any witnessed solution.*
2. *Let $\overline{T}_{M,o}$ be the decision tree over the inputs of R_N, constructed from $T_{M,o}$ by substituting every query to i-th position of an input of S_M with the tree f_i. We require that g_o contains $\overline{T}_{M,o}$ in the sense that any path in g_o as a partial assignment extends some path in $\overline{T}_{M,o}$.*

This is complementary to the honesty property, which intuitively means that if a pseudo-reduction has enough information to give a correct solution, it does so.

We will usually omit the choice of the family of decision trees that checks the correctness of a solution. All we want from it is to check accurately and to have polylogarithmic depth. With respect to such a family, any pseudo-reduction can be transformed into a perceptive one with at most a polylogarithmic blowup in size. To achieve this, we apply the following operations:

1. We extend every g_o by replacing all leaves with the decision tree $\overline{T}_{M,o}$. New leaves inherit the value of the replaced leaf.
2. In every leaf in the new decision trees, if the corresponding path is witnessing, we replace the leaf's value with any witnessed solution.

Note that if the pseudo-reduction was successful on an input x, it will remain so. Moreover, this operation preserves honesty.

Now we are ready to show why these properties are useful. For a pseudo-reduction of R_N to S_M, we call an input of R_N *bad* if the pseudo-reduction is not successful on it. The following lemma shows how such inputs are distributed for an honest and perceptive pseudo-reduction.

▶ **Lemma 9.** *Let (f, g) be an honest and perceptive pseudo-reduction of R_N to S_M of depth d. Then, for any bad input x, there is a non-witnessing partial assignment ρ such that:*

- $|\rho| \le d$,
- $\rho \sqsubset x$,
- *Any x' such that $\rho \sqsubset x'$ is also bad.*

Proof. Since x is bad, there is $o \in O_M$ such that $(f(x), o) \in S_M$, but $(x, g_o(x)) \notin R_N$. Let ρ be the partial assignment corresponding to the execution of g_o on x. It is clear to see that $|\rho| \le d$ and $\rho \sqsubset x$. Since the pseudo-reduction is perceptive and $(x, g_o(x)) \notin R_N$, ρ must be non-witnessing. Also, since our pseudo-reduction is honest, $g_o(x) = \perp$. Then, for any $x' \sqsupset \rho$, we have $(x', g_o(x')) = (x', \perp) \notin R_N$. Lastly, $(f(x'), o) \in S_M$ since $T_{M,o}(f(x')) = T_{M,o}(f(x)) = 1$ due to perceptiveness. ◀

▶ **Corollary 10.** *We can apply this lemma to every bad input and combine the resulting ρ into one set B. Then B is a set of non-witnessing partial assignments with sizes at most d such that x is bad if and only if it extends some $\rho \in B$.*

2.4 Search Problem Restriction

▶ **Definition 11.** *Given a search problem R_N, a set of inputs X_N, and a partial assignment ρ, a restriction of the pair (R_N, X_N) by ρ (denoted as $(R_N, X_N)\restriction\rho$) is a pair (R'_N, X'_N), where $\Sigma'_N = \Sigma_N$, $l'_N = l_N - |\rho|$, $O'_N = O_N$, $X'_N \subset \Sigma_N^{l'_N}$ is a set of strings such that their substitution in ρ in place of all $*$ gives an element of X_N, $R'_N = \{(x', o) \mid$ substituting x' into ρ gives x such that $(x, o) \in R_N\}$.*

If we have a pseudo-reduction (f, g) from (R_N, X_N) to a search problem S_M, then we can also apply the restriction to (f, g) (denoted as $(f, g)\restriction\rho$) to get a pseudo-reduction from $(R_N, X_N)\restriction\rho$ to S_M. It is achieved simply by resolving in (f, g) all queries that are already specified in ρ. Moreover, the following relation holds:

▶ **Lemma 12.** *Given a pseudo-reduction (f, g) from (R_N, X_N) to a search problem S_M and a partial assignment ρ, if $x' \in X_N\restriction\rho$ is a bad input for pseudo-reduction $(f, g)\restriction\rho$, then $x \in X_N$, obtained by substituting x' into ρ, is bad for pseudo-reduction (f, g).*

Proof. Since x' is bad, there is o such that $((f\restriction\rho)(x'), o) \in S_M$, $(x', (g_o\restriction\rho)(x')) \notin R_N\restriction\rho$. But $(f\restriction\rho)(x') = f(x)$, which means $(f(x), o) \in S_M$; and $(g_o\restriction\rho)(x') = g_o(x)$, which means $(x', g_o(x)) \notin R_N\restriction\rho \Leftrightarrow (x, g_o(x)) \notin R_N$. ◀

3 Main Theorem

▶ **Theorem 13** (Main theorem). *Let $R, S \in$ TFNP, $X = \{X_N \subset \Sigma_N^{l_N}\}_{N\in\mathbb{N}}$. Let the following conditions be met:*

1. *There is no reduction from (R, X) to S.*
2. *For any $p(N) = \mathrm{poly}(\log N)$ and for any sufficiently large N, there are sets $\mathcal{P}_1 = \mathcal{P}_1(N)$ and $\mathcal{P}_2 = \mathcal{P}_2(N)$ of partial assignments over inputs of R_N such that:*
   - **a.** *For any non-witnessing partial assignment ρ over inputs of R_N with $|\rho| \le p(N)$ and such that ρ is extendable to some $x \in X_N$, there is $A_\rho \subset \mathcal{P}_1$ such that $\forall x \in X_N$ $(\rho \sqsubset x \Leftrightarrow \exists \tau \in A_\rho : \tau \sqsubset x)$.*
   - **b.** *For any sequence $\{\kappa_N \in \mathcal{P}_2(N)\}_N$, (R, X) is reducible to $\{(R_N, X_N)\restriction\kappa_N\}_N$.*
   - **c.** *For κ taken uniformly from $\mathcal{P}_2$, and then x taken uniformly from $\{x \in X_N : x \sqsupset \kappa\}$, the resulting distribution of x is uniform on X_N.*
   - **d.** *There is $\delta > 0$ such that for any $T \subset \mathcal{P}_1$:*

$$|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \sqsupset \tau\}| \ge (1 - o(N^{-\delta}))|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \overset{X_N}{\upuparrows} \tau\}|.$$

Then there is no randomized reduction from (R, X) to S.

Before we proceed with the proof, let us informally explain what sets $\mathcal{P}_1$ and $\mathcal{P}_2$ represent and how this proof works.

The set $\mathcal{P}_1$ consists of "small" partial assignments. Usually we want to choose $\mathcal{P}_1$ so that all its elements are non-witnessing and have the same size and structure. The purpose of this set is to refine the set of partial assignments, resulting from the application of Corollary 10, into the set $T \subset \mathcal{P}_1$ with the same properties. This allows us to work with small partial assignments of the same size and structure, which is convenient.

The set $\mathcal{P}_2$ consists of "big" partial assignments. As with $\mathcal{P}_1$, we usually want to choose $\mathcal{P}_2$ so that all its elements are non-witnessing and have the same size and structure. Moreover, from $\kappa \in \mathcal{P}_2$ we expect that restricting (R_N, X_N) by κ turns it into essentially the same problem, but smaller.

For a quick example of such $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N we refer to Figure 1.

In general, the proof is structured as follows:

- We are given (f, g) with small complexity and we want to show that it performs poorly under the uniform distribution over X_N.
- Assuming (f, g) is honest and perceptive, apply Corollary 10 to obtain the set B of partial assignments. For B holds that $x \in X_N$ is bad iff $\exists \rho \in B : x \sqsupset \rho$.
- Use condition 2a to refine B into $T \subset \mathcal{P}_1$ with the same property as B.
- Show that (for infinitely many N) any $\kappa \in \mathcal{P}_2$ must be consistent (relative to X_N) with some $\tau \in T$. Indeed, otherwise, we can restrict our pseudo-reduction by κ to kill all $\tau \in T$, which means that the resulting pseudo-reduction is always successful. But that would contradict condition 1, since a restriction by κ turns our problem into essentially the same problem, but smaller.
- Use condition 2d to show that almost all $\kappa \in \mathcal{P}_2$ must extend some $\tau \in T$.
- Use condition 2c to show that the same holds for almost all $x \in X_N$, which due to the property of T means that almost all $x \in X_N$ are bad. This concludes the proof.

Proof. Let us fix any $p(N) = \text{poly}(\log N)$. For arbitrary N let (f, g) be any pseudo-reduction of R_N to S_M with complexity $p(N)$. Our goal is to show that this pseudo-reduction performs poorly on x, taken uniformly from X_N. We can assume (f, g) to be honest and perceptive: as shown in Subsection 2.3, these properties can be achieved with polynomial blowup in complexity. Moreover, if the pseudo-reduction was successful on an input, it will remain so. Then, from Corollary 10, there is a set of non-witnessing partial assignments B such that $x \in X_N$ is bad if and only if it extends some $\rho \in B$. We exclude from B all ρ that are not extendable to any $x \in X_N$. Since all $\rho \in B$ have a size at most $p(N)$, we can use condition 2a to construct $T = \bigcup_{\rho \in B} A_\rho \subset \mathcal{P}_1$. By design, for such T we also have that $x \in X_N$ is bad if and only if it extends some $\tau \in T$.

Now, we proceed to show that for infinitely many N, any $\kappa \in \mathcal{P}_2$ is consistent relative to X_N with at least one $\tau \in T$. Assume that there is κ such that it is not consistent relative to X_N with any $\tau \in T$. Then the pseudo-reduction $(f_N, g_N)\restriction\kappa$ from $(R_N, X_N)\restriction\kappa_N$ to S_M is successful on all inputs of $(R_N, X_N)\restriction\kappa_N$. Indeed, if x' is a bad input, from Lemma 12 we have that x, obtained by substituting x' into κ, is also bad. But it is impossible, since a bad input must extend some $\tau \in T$, which would contradict inconsistency of κ. Now, if such κ exists for all sufficiently large N, then $\{(R_N, X_N)\restriction\kappa_N\}_N$ is reducible to S, which together with conditions 2b and 1 leads to a contradiction.

Finally, from condition 2d we can conclude that for some $\delta > 0$ for infinitely many N at least $1 - o(N^{-\delta})$ of all $\kappa \in \mathcal{P}_2$ are not just consistent, but even extend some $\tau \in T$. Since from condition 2c random x can be generated by extending random $\kappa \in \mathcal{P}_2$, at least $1 - o(N^{-\delta})$ of all $x \in X_N$ also extend some $\tau \in T$, which means that they are bad. So, at most $o(N^{-\delta})$ of all inputs are good. But in Lemma 4 for randomized reduction to exist we require a success rate of at least $\frac{1}{\text{poly}(\log N)}$. ◀

Auxiliary Results

The following results are useful for showing that the premise of the main theorem holds. In particular, the following lemma shows that under certain conditions on $R \in \mathsf{TFNP}$ and its inputs X_1 and X_2, for any $S \in \mathsf{TFNP}$, (R, X_1) is reducible to S if and only if (R, X_2) is reducible to S. This is useful to get condition 1 of the theorem.

▶ **Lemma 14.** *Let $R = (R_N)$ be any search problem from TFNP, let X and X' be inputs of R such that $X_N \subset X'_N$ for all N. Assume that for any $p(N) = \text{poly}(\log N)$ for all sufficiently large N for any non-witnessing partial assignment ρ of size at most $p(N)$ it holds that if ρ can be extended to $x' \in X'_N$, it can also be extended to $x \in X_N$. Then, for any $S \in \mathsf{TFNP}$, (R, X) is reducible to S if and only if (R, X') is reducible to S.*

Proof. It suffices to show that if (R, X') is not reducible to S, then (R, X) is also not reducible to S. In addition, we can consider only honest and perceptive pseudo-reductions. Let $p(N) = \text{poly}(\log N)$, let (f, g) be an arbitrary honest and perceptive pseudo-reduction from (R_N, X') to S_M with the complexity at most $p(N)$. For infinitely many N, there is a bad input $x' \in X'_N$ for any such pseudo-reduction. Then we can take the partial assignment $\rho \sqsubset x'$ from Lemma 9. For sufficiently large N, ρ can be extended to $x \in X_N$, which means that x is also bad. That concludes the proof. ◀

If we take $\mathcal{P}_2$ and X_N as in Theorem 13, we can associate with them graph G with $V(G) = \mathcal{P}_2$, and $E(G)$ being pairs of elements from $\mathcal{P}_2$ that are consistent relative to X_N. If $\mathcal{P}_2$ and X_N are very symmetric, this graph has useful properties. These properties are demonstrated in the following lemma:

▶ **Lemma 15.** *Let G be a graph, $V_0 \sqcup \ldots \sqcup V_n \subset V(G)$. Let there be $d_{0\to k}, d_{k\to 0} > 0$ for all $k \in [n]$ such that:*

- $\forall v \in V_0 : |N(v) \cap V_k| = d_{0\to k}$;
- $\forall v \in V_k : |N(v) \cap V_0| = d_{k\to 0}$.

There $N(\cdot)$ is the neighborhood function. Also, let $N(V_0) \subset V_0 \sqcup \ldots \sqcup V_n$. Then for any nonempty $V' \subset V_0$, it holds:

$$\frac{|N(V') \setminus V_0|}{|V'|} \geq \frac{|N(V_0) \setminus V_0|}{|V_0|}$$

Proof.

$$|N(V') \setminus V_0| = \sum_{k\in[n]} |N(V') \cap V_k| \geq \sum_{k\in[n]} |V'| \frac{d_{0\to k}}{d_{k\to 0}} = |V'| \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}}$$

$$|N(V_0) \setminus V_0| = \sum_{k\in[n]} |N(V_0) \cap V_k| = \sum_{k\in[n]} |V_0| \frac{d_{0\to k}}{d_{k\to 0}} = |V_0| \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}}$$

$$\frac{|N(V') \setminus V_0|}{|V'|} \geq \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}} = \frac{|N(V_0) \setminus V_0|}{|V_0|}.$$

◀

For nonempty $T \subset \mathcal{P}_1$ let us look at the ratio between the number of $\kappa \in \mathcal{P}_2$ that are consistent with some $\tau \in T$ relative to X_N and the number of $\kappa \in \mathcal{P}_2$ that extend some $\tau \in T$. The condition 2d of Theorem 13 requires this ratio to be sufficiently close to 1. The following lemma uses the previous result to show that under certain conditions, if we already showed that this ratio is close to 1 for $|T| = 1$, the same holds for arbitrary T.

▶ **Lemma 16.** *Let $X_N \subset \Sigma_N^{l_N}$, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be sets of partial assignments over X_N. Let it be true for some $\gamma > 1$ that for every $\tau \in \mathcal{P}_1$, it holds:*

1. *For every $x \in X_N$ such that $\tau \sqsubset x$, there is $\kappa \in \mathcal{P}_2$ such that $\tau \sqsubset \kappa \sqsubset x$.*
2. $|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \gamma |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$
3. *For graph G with $V(G) = \mathcal{P}_2$, $E(G) = \{(\kappa_1, \kappa_2) \mid \kappa_1 \overset{X_N}{\uparrow\uparrow} \kappa_2\}$, $V_0 = \{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}$, $V_k = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $|\tau| - k$ common non-$*$ positions$\}$, all conditions of Lemma 15 are met.*

Then, for any $T \subset \mathcal{P}_1$, it holds that:

$$|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| \leq \gamma |\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \sqsupset \tau\}|.$$

Proof. The first condition gives us the following equivalence:

▷ Claim 17. For any $\tau \in \mathcal{P}_1$ and $\kappa \in \mathcal{P}_2$: $\kappa \overset{X_N}{\uparrow\uparrow} \tau \Leftrightarrow \exists \kappa' \in \mathcal{P}_2 : \kappa' \sqsupset \tau \wedge \kappa \overset{X_N}{\uparrow\uparrow} \kappa'$.

Now, let us prove the claim of the lemma by induction on the size of T. For $|T| = 1$, we already know that the equality holds. Let $T = T_0 \cup \{\tau\}$ and for T_0 the required inequality holds. Define $F(\tau) = \{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}$, $F_w(\tau) = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}$, $N(\kappa') = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \kappa'\}$. For sets of τ or κ definitions are similar. Note that $F_w = N \circ F$ due to Claim 17. In these terms, we need to prove that:

$$|F_w(T)| \leq \gamma |F(T)|.$$

First, assume the case $F(T_0) \cap F(\tau) = \varnothing$. Then:

$$|F_w(T)| \leq |F_w(T_0)| + |F_w(\tau)| \leq \gamma(|F(T_0)| + |F(\tau)|) = \gamma |F(T)|.$$

Now assume that $F(T_0) \cap F(\tau) = V' \neq \varnothing$, $F(\tau) \setminus V' = V''$. Then we have:

$$
\begin{aligned}
&|F_w(T)| = \\
&|F_w(T_0)| + |F_w(\tau) \setminus F_w(T_0)| \leq && \text{(splitting } F_w(T) \text{ into 2 parts)} \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus F_w(T_0)| = && \text{(applying induction)} \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus N(F(T_0))| \leq && \text{(using that } F_w = N \circ F) \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus N(V')| = && \text{(using that } V' \subset F(T_0)) \\
&\gamma|F(T_0)| + |F_w(\tau)| - |N(V')| = && (F_w(\tau) = N(F(\tau)) \supset N(V')) \\
&\gamma|F(T_0)| + \gamma|F(\tau)| - |N(V')| = && \text{(applying the second condition)} \\
&\gamma|F(T_0)| + \gamma|V''| + \gamma|V'| - |N(V')| = && (F(\tau) = V' \sqcup V'') \\
&\gamma|F(T)| + \gamma|V'| - |N(V')|. && (F(T) = F(T_0) \sqcup V'')
\end{aligned}
$$

Finally, $\gamma|F(T)| + \gamma|V'| - |N(V')| \leq \gamma|F(T)|$, since:

$$
\begin{aligned}
&|N(V')| \geq \\
&|V'| + |N(V') \setminus F(\tau)| \geq && (V' \subset N(V') \text{ and } V' \subset F(\tau)) \\
&|V'| + \frac{|V'|}{|F(\tau)|}|N(F(\tau)) \setminus F(\tau)| = && \text{(using Lemma 15 with } V_0 = F(\tau)) \\
&\frac{|N(F(\tau))|}{|F(\tau)|}|V'| = && (F(\tau) \subset N(F(\tau))) \\
&\gamma|V'|. && \text{(applying the second condition)}
\end{aligned}
$$

◀

4 Applications of the Main Theorem

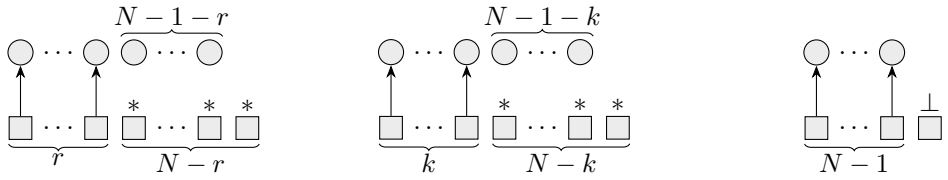
4.1 PPP

▶ **Definition 18.** PPP *is a class of* TFNP *problems that are reducible to the search problem* PIGEON.
$\textsc{Pigeon} = \{\textsc{Pigeon}_N \subset \Sigma_N^{l_N} \times O_N\}$, *where:*
$\Sigma_N = [N-1] \cup \{\perp\}$, $l_N = N$, $O_N = [N] \cup \{(i,j) \in [N]^2 \mid i \neq j\}$.
Positions $i \in [N]$ *are called pigeons, and values* $j \in [N-1]$ *are called holes,* $x_i = j$ *represents pigeon* i *being mapped to hole* j.
Correct solutions for input x*: any* i *such that* $x_i = \perp$ *(homeless pigeon) or* (i,j) *such that* $x_i = x_j$ *(collision).*

Our goal is to show that there is no randomized reduction from PIGEON to $S \in$ TFNP, given that there is no deterministic reduction from PIGEON to $S \in$ TFNP. To apply Theorem 13, we first need to come up with a set of inputs X_N for each N. We take X_N to be the set of all bijections from $[N]$ to $[N-1] \cup \{\perp\}$. Now we can apply Lemma 14 (with X' being all inputs) to show that (PIGEON, X) is also not reducible to S. The conditions of the lemma hold, since any non-witnessing partial assignment contains neither a homeless pigeon nor a collision, which means it can be extended to $x \in X_N$.

Now, given $p(N) = \text{poly}(\log N)$, we need to come up with $\mathcal{P}_1$ and $\mathcal{P}_2$ for every sufficiently large N. As $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of all non-witnessing partial assignments of size r and k, respectively (see Figure 1). There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Let us go through the sub-items of Theorem 13:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \le p(N)$ we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N} = O(N^{\frac{1}{\varepsilon}})$ such that $\lfloor \hat{N}^\varepsilon \rfloor = N$. Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, the search problem $(\textsc{Pigeon}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ is equivalent to $(\textsc{Pigeon}_N, X_N)$ up to an alphabet renaming, which means there is a trivial always successful pseudo-reduction from $(\textsc{Pigeon}_N, X_N)$ to $(\textsc{Pigeon}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$, with complexity $\log \hat{N} = O(\log N)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: For this one, we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on the choice of $\tau \in \mathcal{P}_1$. Other conditions of the lemma also hold. Finally, simple combinatorial estimates (see Appendix A) show that $\gamma = (1 + O(N^{-\frac{1}{4}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{5}}))$, which concludes the proof.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 1 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for $\textsc{Pigeon}_N$ up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons are mapped to holes.

4.2 PPAD and PPADS

▶ **Definition 19.** PPAD *and* PPADS *are classes of* TFNP *problems that are reducible to search problems* InjPigeon *(Injective Pigeon) and* BijPigeon *(Bijective Pigeon), respectively.*
$\textsc{InjPigeon} = \{\textsc{InjPigeon}_N \subset \Sigma_N^{l_N} \times O_N\}$*, where:*
$\Sigma_N = [N] \cup \{\bot\}$, $l_N = 2N - 1$, $O_N = [N]$.
Any $x \in \Sigma_N^{l_N}$ *can be interpreted as a pair* $(s, p) \in \Sigma_N^N \times \Sigma_N^{N-1}$. *We count* $i \in [N]$ *as the correct solution for* $x = (s, p)$ *if either* $s_i \in \{\bot, N\}$ *or* $p_{s_i} \ne i$.
BijPigeon is defined in the same way, except that we also count $i \in [N]$ *as the correct solution for* $x = (s, p)$ *if* $i < N$ *and either* $p_i = \bot$ *or* $s_{p_i} \ne i$.

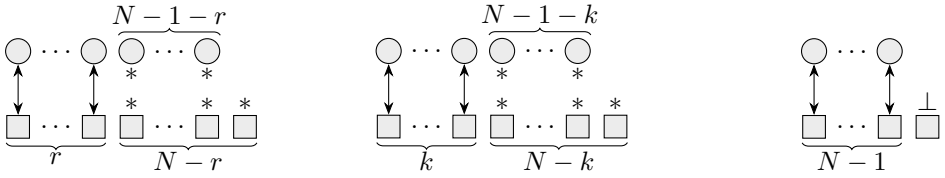
The search problems InjPigeon and BijPigeon are introduced and shown to be complete in the corresponding classes in [5]. For original definitions of PPAD and PPADS, see [15]. These problems are similar to Pigeon: there are N pigeons and $N - 1$ holes. The difference is that there is mapping not only from pigeons to holes but also from holes to pigeons. We say that pigeon i is homeless, if $s_i \in \{N, \bot\}$; hole j is empty if $p_j = \bot$, and pigeon-hole pair is a pair (i, j) such that $s_i = j$, $p_j = i$.

As with Pigeon, our goal is to show the equivalence of deterministic and randomized reducibility from these problems to any $S \in$ TFNP. First, we restrict our attention to consistent inputs: $X'_N = \{x = (s, p) \mid (\forall i \in [N]\ \forall j \in [N-1] : s_i = j \Leftrightarrow p_j = i) \wedge (\forall i \in [N] : s_i \ne N)\}$. It is easy to see that both of these problems are reducible to themselves, restricted to such inputs. From now on, we proceed with each problem separately.

4.2.1 PPAD

First, we can further refine X'_N to X_N so that there are no empty holes: $X_N = \{(s,p) \in X'_N \mid \forall j : p_j \neq \bot\}$. As with PIGEON, Lemma 14 can be applied here: any non-witnessing ρ, extendable to $x' \in X'$, consists only of pigeon-hole pairs, which means it is extendable to $x \in X_N$.

Similarly to PIGEON, as $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of all non-witnessing partial assignments of size $2r$ and $2k$, respectively, with the additional requirement on partial assignments: $s_i = j \Leftrightarrow p_j = i$. There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Just like for PIGEON, the elements of these sets represent partially defined injections from $[N]$ to $[N-1]$ with domain sizes r and k, respectively. As it can be seen from Figure 1 and Figure 2, $\mathcal{P}_1, \mathcal{P}_2$ and X_N for PIGEON_N and for BIJPIGEON_N look very similar. From now on, the proof almost exactly repeats the proof for PIGEON, including combinatorial calculations.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 2 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for BIJPIGEON_N up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons and holes are mapped to each other.

4.2.2 Obstacles with PPADS

It may seem that INJPIGEON can be handled in the same way as BIJPIGEON. However, unlike BIJPIGEON, for INJPIGEON, partial assignments with empty holes do not witness an answer. From this it follows that if we want to apply Theorem 13, we require that elements of sets $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N contain empty holes. It seems that this difference is significant, since we were unable to come up with such $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N that all conditions of Theorem 13 hold.

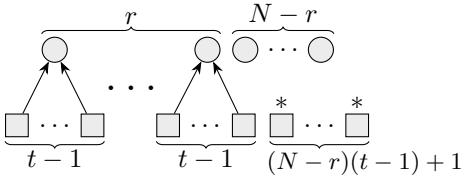
4.3 t-PPP

▶ **Definition 20.** *For nondecreasing $t = t(N)$, $2 \leq t \leq \text{poly}(\log N)$, t-*PPP *is a class of* TFNP *problems that are reducible to the search problem t-$\text{PIGEON} = \{t(N)\text{-PIGEON}_N\}_N$. t-PIGEON_N is defined as follows:*
$\Sigma_N = [N] \cup \{\bot\}$, $l_N = (t-1)N+1$, $O_N = [l_N] \cup \{(i_1, \ldots, i_t) \in [l_N]^t \mid i_s$ are pairwise distinct$\}$. Correct solution for input x: any i such that $x_i = \bot$ or $(i_1, \ldots, i_t)$ such that $x_{i_1} = \ldots = x_{i_t}$.

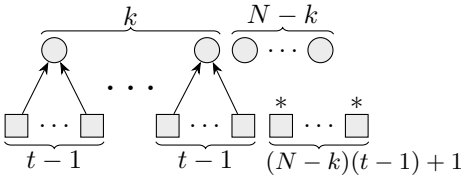
These problems were introduced in [8]. There, it was shown that for $n = \log N$, $t_1(n), t_2(n) = \text{poly}(n)$, if $t_1(n) \leq t_2(n^c)$ and $t_2(n) \leq t_1(n^c)$ for some c and all sufficiently large N, then t_1-PIGEON and t_2-PIGEON are reducible to each other. Using that, we may consider only $t(N) \leq \sqrt{\log N}$. Indeed, since $t(n) \leq n^c$ for some c and sufficiently large N, $\hat{t}(n) = t(\lfloor n^{\frac{1}{2c}} \rfloor) \leq \sqrt{\log N}$, and $\hat{t}(n) \leq t(n) \leq \hat{t}(n^{2c+1})$.

As a hard set of inputs for t-PIGEON_N we take $X_{N,t}$, which is a set of such inputs x that x equals $\bot$ in exactly one position, and for any $a \in [N]$, there are exactly $t-1$ positions of x with that value. For such inputs, the only correct solution is the position of $\bot$. Any

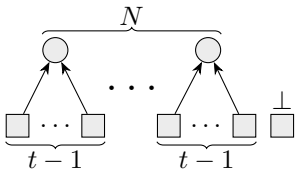
non-witnessing partial assignment can be extended to such x, so we can apply Lemma 14 (with X' being all inputs). Now we are ready to apply the main theorem with $X_N = X_{N,t(N)}$. Given $p(N) = \mathrm{poly}(\log N)$, as $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of partial assignments of size $r(t-1)$ and $k(t-1)$, respectively, with the condition that these partial assignments do not contain $\perp$, and every other value occurs either 0 or $t-1$ times (see Figure 3). Here $r = p(N)$, $k = N - \lfloor N^{\varepsilon} \rfloor$ for $\varepsilon = \varepsilon(N) = \frac{1}{4t}$.



(a) The structure of $\tau \in \mathcal{P}_1$.



(b) The structure of $\kappa \in \mathcal{P}_2$.



(c) The structure of $x \in X_N$.

Figure 3 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for t-PIGEON$_N$ up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons are mapped to holes.

Let us go through the sub-items of Theorem 13:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \le p(N)$ we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N}$ such that $\lfloor \hat{N}^{\varepsilon(\hat{N})} \rfloor = N$. Moreover:

$$N \ge \hat{N}^{\frac{1}{8t(\hat{N})}} \ge \hat{N}^{\frac{1}{8\sqrt{\log \hat{N}}}} = 2^{\frac{\sqrt{\log \hat{N}}}{8}} \Rightarrow \hat{N} \le 2^{64(\log N)^2} = 2^{O((\log N)^2)}$$

  Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, the search problem $(t(\hat{N})\text{-PIGEON}_{\hat{N}}, X_{\hat{N},t(\hat{N})})\restriction\kappa$ is equivalent to $(t(\hat{N})\text{-PIGEON}_N, X_{N,t(\hat{N})})$ up to an alphabet renaming. Now, there is an always successful pseudo-reduction from $(t(N)\text{-PIGEON}_N, X_{N,t(N)})$ to $(t(\hat{N})\text{-PIGEON}_N, X_{N,t(\hat{N})})$, achieved by introducing for each of N holes $(t(\hat{N}) - t(N))$ new pigeons, pointing to it. Then, there is an always successful pseudo-reduction from $(t(N)\text{-PIGEON}_N, X_{N,t(N)})$ to $(t(\hat{N})\text{-PIGEON}_{\hat{N}}, X_{\hat{N},t(\hat{N})})\restriction\kappa$ with complexity $O(\log \hat{N}) = O((\log N)^2)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: For this one we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\restriction\restriction} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on choice of $\tau \in \mathcal{P}_1$. Other conditions of lemma also follow from symmetry. Finally, simple combinatorial estimates (see Appendix B) show that $\gamma = (1 + O(N^{-\frac{1}{2}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{3}}))$, which concludes the proof.

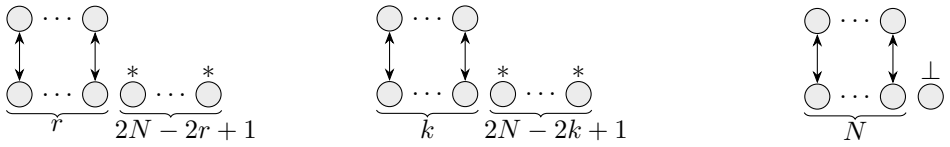
4.4 PPA

▶ **Definition 21.** PPA *is a class of* TFNP *problems that are reducible to the search problem* $\textsc{Lonely} = \{\textsc{Lonely}_N\}_N$. $\textsc{Lonely}_N$ *is defined as follows:*
$\Sigma_N = [2N+1] \cup \{\bot\}$, $l_N = 2N+1$, $O_N = [2N+1]$. *Correct solution for input* x*: any* i *such that* $x_i \in \{i, \bot\}$ *or* $x_{x_i} \neq i$.

First, we restrict our attention to consistent inputs: $X'_N = \{x \mid (\forall i, j \in [2N+1] : x_i = j \Leftrightarrow x_j = i) \wedge (\forall i \in [2N+1] : x_i \neq i)\}$. It is easy to see that PPA is reducible to itself, restricted to such inputs. Now all inputs represent graph, consisting of pairs and lonely nodes. We can further refine X' to $X \subset X'$ such that there is exactly one lonely node. Lemma 14 shows that X is sufficient for checking reducibility from PPA.

Now we move to the second part of Theorem 13. We need to specify $\mathcal{P}_1$ and $\mathcal{P}_2$, given $p(N) = \text{poly}(\log N)$. We take sets of partial assignments, extendable to inputs in X_N, that reveal exactly r and k pairs, respectively (see Figure 4). There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Now we are ready to apply Theorem 13. Let us go through the sub-items:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \leq p(N)$ and which is extendable to $x \in X_N$, we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N} = O(N^{\frac{1}{\varepsilon}})$ such that $\lfloor \hat{N}^\varepsilon \rfloor = N$. Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, search problem $(\textsc{Lonely}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ is equivalent to $(\textsc{Lonely}_N, X_N)$ up to renaming the alphabet, which means there is a trivial always successful pseudo-reduction from $(\textsc{Lonely}_N, X_N)$ to $(\textsc{Lonely}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ with complexity $\log \hat{N} = O(\log N)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: Here we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on choice of $\tau \in \mathcal{P}_1$. Other conditions of lemma also follow from symmetry. Finally, simple combinatorial estimates (see Appendix C) show that $\gamma = (1 + O(N^{-\frac{1}{4}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{5}}))$, which concludes the proof.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 4 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for $\textsc{Lonely}_N$ up to a permutation of nodes. Here circles represent nodes, and arrows show how they are matched.

5 Future Directions

In addition to all these classes, we also attempted to apply this method to complete problems in classes PLS, PPADS, and PWPP. In most cases, our method fails in combinatorial estimates (condition 2d of Theorem 13). We hope that this method can be generalized to also work for these classes. But it is quite possible that for some of these classes, similarly to Lossy, randomized reducibility from them does not imply deterministic reducibility.

References

1 Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of np search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998. doi:10.1006/jcss.1998.1575.

2 J. Buresh-Oppenheim and T. Morioka. Relativized np search problems and propositional proof systems. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 54–67, 2004. doi:10.1109/CCC.2004.1313795.

3 Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP Characterizations of Proof Systems and Monotone Circuits. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:40, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.30.

4 Noah Fleming, Stefan Grosser, Siddhartha Jain, Jiawei Li, Hanlin Ren, Morgan Shirley, and Weiqiang Yuan. Total Search Problems in ZPP. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:26, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2026.60.

5 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Further collapses in tfnp, 2022. arXiv:2202.07761.

6 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and tfnp. *Journal of the ACM*, 71(4):1–45, August 2024. doi:10.1145/3663758.

7 Pavel Hubáček, Erfan Khaniki, and Neil Thapen. TFNP Intersections Through the Lens of Feasible Disjunction. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:24, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.63.

8 Siddhartha Jain, Jiawei Li, Robert Robere, and Zhiyang Xun. On pigeonhole principles and ramsey in tfnp. *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–428, 2024. URL: https://api.semanticscholar.org/CorpusID:267094765.

9 Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016. doi:10.1016/j.jcss.2015.08.001.

10 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi:10.1016/0022-0000(88)90046-3.

11 Oliver Korten. Derandomization from Time-Space Tradeoffs. In Shachar Lovett, editor, *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:26, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2022.37.

12 Jiawei Li. Total NP Search Problems with Abundant Solutions. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.75.

13 Yuhao Li, William Pires, and Robert Robere. Intersection Classes in TFNP and Proof Complexity. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:22, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.74.

14 Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. Master's thesis, University of Toronto, 2001. URL: https://www.collectionscanada.ca/obj/s4/f2/dsk3/ftp04/MQ58775.pdf.

15 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

16 Pavel Pudlák and Neil Thapen. Random Resolution Refutations. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:10, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2017.1.

17 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227, 1977. URL: https://api.semanticscholar.org/CorpusID:143169.

A Estimate on the number of consistent κ for Pigeon

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 1. There $r = \text{poly}(\log N)$, $k = N - s$, $s = \lfloor N^{\varepsilon} \rfloor$, $\varepsilon = \frac{1}{4}$. For an arbitrary $\tau \in \mathcal{P}_1$, let $a_l = |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $r - l$ common non-$*$ positions$\}$. Let us write an exact formula for this value. To choose such κ, we need to:

1. Choose $r - l$ pigeons that are common with τ.
2. Choose the remaining $k - (r - l)$ pigeons among $N - r$ not used in τ.
3. Choose the remaining $k - (r - l)$ holes among $N - 1 - r$ not used in τ.
4. Choose bijection from $k - (r - l)$ pigeons to $k - (r - l)$ holes.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{N-r}{k-(r-l)}\binom{N-1-r}{k-(r-l)}(k-(r-l))! = \binom{r}{l}\binom{N-r}{s-l}\binom{N-1-r}{s-l-1}(N-s-(r-l))!$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r) \cdot O\left(\frac{s}{N}\right) \cdot O\left(\frac{s}{N}\right) \cdot O(N) = o(N^{-\frac{1}{4}}).$$

Note that this estimate is uniform over $l \in \{0, \ldots, r-1\}$. Finally:

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0 \left(1 + \sum_{l=1}^{r} \frac{a_l}{a_0}\right) = |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}| \left(1 + o\left(\sum_{l=1}^{r} N^{-\frac{l}{4}}\right)\right) = |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1 + o(N^{-\frac{1}{4}})).$$

B Estimate on the number of consistent κ for t-Pigeon

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 3. There $r = \text{poly}(\log N)$, $k = N - s$, $s \coloneqq \lfloor N^{\varepsilon} \rfloor$, $\varepsilon = \frac{1}{4t}$, $t \leq \sqrt{\log N}$. For arbitrary $\tau \in \mathcal{P}_1$, let $a_l = |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have exactly $r - l$ common non-$*$ symbols$\}|$. To choose such κ, we need to:

1. Choose $r-l$ "$t-1$ pigeons pointing to one hole" groups that are common with τ.
2. Choose the remaining $k-(r-l)$ holes among $N-r$ not used in τ.
3. Choose the remaining $(t-1)(k-(r-l))$ pigeons among $(t-1)(N-r)+1$ not used in τ.
4. Choose how to map $(t-1)(k-(r-l))$ pigeons to $k-(r-l)$ holes so that at each hole point exactly $t-1$ pigeons.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{N-r}{k-(r-l)}\binom{(t-1)(N-r)+1}{(t-1)(k-(r-l))}\frac{((t-1)(k-(r-l)))!}{((t-1)!)^{k-(r-l)}} = \binom{r}{l}\binom{N-r}{s-l}\binom{(t-1)(N-r)+1}{(t-1)(s-l)+1}\frac{((t-1)(N-s-(r-l)))!}{((t-1)!)^{N-s-(r-l)}}.$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r)\cdot O\left(\frac{s}{N}\right)\cdot O\left(\frac{((t-1)s)^{t-1}}{((t-1)N)^{t-1}}\right)\cdot O\left(\frac{((t-1)N)^{t-1}}{(t-1)!}\right) = O\left(\frac{rs^t(t-1)^{t-1}}{N(t-1)!}\right) = O\left(\frac{rs^t}{N}\frac{(t-1)^{t-1}}{\left(\frac{t-1}{e}\right)^{t-1}}\right) = O\left(\frac{rs^te^{t-1}}{N}\right) = o(N^{-\frac{1}{2}}).$$

Note that this estimate is uniform over $l \in \{0,\dots,r-1\}$. Finally:

$$|\{\kappa\in\mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0\left(1+\sum_{l=1}^{r}\frac{a_l}{a_0}\right) = |\{\kappa\in\mathcal{P}_2 \mid \kappa \sqsupset \tau\}|\left(1+o\left(\sum_{l=1}^{r}N^{-\frac{l}{2}}\right)\right) = |\{\kappa\in\mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1+o(N^{-\frac{1}{2}})).$$

C Estimate on the number of consistent κ for Lonely

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 4. There $r = \mathrm{poly}(\log N)$, $k = N-s$, $s \coloneqq \lfloor N^{\varepsilon}\rfloor$, $\varepsilon = \frac{1}{4}$. For arbitrary $\tau\in\mathcal{P}_1$, let $a_l = |\{\kappa\in\mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $r-l$ common pairs$\}|$. To choose such κ, we need to:

1. Choose $r-l$ pairs that are common with τ.
2. Choose the remaining $2(k-(r-l))$ nodes among $2N-1-2r$ not used in τ.
3. Choose how to divide those $2(k-(r-l))$ nodes into pairs.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{2N-1-2r}{2(k-(r-l))}(2(k-(r-l))-1)!! = \binom{r}{l}\binom{2N-1-2r}{2s-2l-1}(2(N-s-r+l)-1)!!$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r)\cdot O\left(\frac{s^2}{N^2}\right)\cdot O(N) = O\left(\frac{rs^2}{N}\right) = o(N^{-\frac{1}{4}}).$$

Note that this estimate is uniform over $l \in \{0, \dots, r-1\}$. Finally:

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0 \left(1 + \sum_{l=1}^{r} \frac{a_l}{a_0}\right) =$$

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}| \left(1 + o\left(\sum_{l=1}^{r} N^{-\frac{l}{4}}\right)\right) =$$

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1 + o(N^{-\frac{1}{4}})).$$