Average Case Complexity, Revisited

Oded Goldreich

Abstract. More than two decades elapsed since Levin set forth a theory of average-case complexity. In this survey we present the basic aspects of this theory as well as some of the main results regarding it. The current presentation deviates from our old "Notes on Levin's Theory of Average-Case Complexity" (*ECCC*, TR97-058, 1997) in several aspects. In particular:

- We currently view average-case complexity as referring to the performance on "average" (or rather typical) instances, and not as the average performance on random instances. (Thus, it may be more justified to refer to this theory by the name typical-case complexity, but we retain the name average-case for historical reasons.)
- We include a treatment of search problems, and a presentation of the reduction of "NP with sampleable distributions" to "NP with P-computable distributions" (due to Impagliazzo and Levin, *31st FOCS*, 1990).
- We include Livne's result (*ECCC*, TR06-122, 2006) by which all natural NPC-problems have average-case complete versions. This result seems to shed doubt on the association of P-computable distributions with natural distributions.

Keywords: Average-Case Complexity.

This text has been revised based on [6, Sec. 10.2].

1 Introduction

In light of the apparent infeasibility of solving numerous useful computational problems, it is natural to ask whether these problems can be relaxed such that the relaxation is both useful and allows for feasible solving procedures. We stress two aspects about the foregoing question: on one hand, the relaxation should be sufficiently good for the intended applications; but, on the other hand, it should be significantly different from the original formulation of the problem so to escape the infeasibility of the latter. We note that whether a relaxation is adequate for an intended application depends on the application, and thus much of the material in this chapter is less robust (or generic) than the treatment of the non-relaxed computational problems.

One commonly considered type of relaxation refers to the computational problems themselves; that is, for each problem instance we *extend the set of admissible solutions*. In the context of search problems this means settling for

solutions that have a value that is "sufficiently close" to the value of the optimal solution (with respect to some value function). Needless to say, the specific meaning of 'sufficiently close' is part of the definition of the relaxed problem. In the context of decision problems this means that for some instances both answers are considered valid; specifically, we shall consider promise problems in which the no-instances are "far" from the yes-instances in some adequate sense (which is part of the definition of the relaxed problem).

In this survey, we consider a different type of relaxation. We do not relax the computational problems themselves, but rather the notion of solving them efficiently. Specifcally, this type of relaxation deviates from the requirement that the solver provides an adequate answer on each valid instance. Instead, the behavior of the solver is analyzed with respect to a predetermined input distribution (or a class of such distributions), and bad behavior may occur with negligible probability where the probability is taken over this input distribution. That is, we replace worst-case analysis by *average-case* (or rather *typical-case*) *analysis*. Needless to say, a major component in this approach is limiting the class of distributions in a way that, on one hand, allows for various types of natural distributions and, on the other hand, prevents the collapse of the corresponding notion of average-case hardness to the standard notion of worst-case hardness.

1.1 The basic mindframe of average-case complexity

The common approach of complexity theory is termed worst-case complexity, because it refers to the performance of potential algorithms on each legitimate instance (and hence to the performance on the worst possible instance). That is, computational problems were defined as referring to a set of instances and performance guarantees were required to hold for each instance in this set. In contrast, average-case complexity allows ignoring a negligible measure of the possible instances, where *the identity of the ignored instances is determined by the analysis of potential solvers and not by the problem's statement*.

A few comments are in place. Firstly, as just hinted, the standard statement of the worst-case complexity of a computational problem (especially one having a promise) may also ignores some instances (i.e., those considered inadmissible or violating the promise), but these instances are determined by the problem's statement. In contrast, the inputs ignored in average-case complexity are not inadmissible in any inherent sense (and are certainly not identified as such by the problem's statement). It is just that they are viewed as exceptional when claiming that a specific algorithm solve the problem; that is, these exceptional instances are determined by the analysis of that algorithm. Needless to say, these exceptional instances ought to be rare (i.e., occur with negligible probability).

The last sentence raises a couple of issues. Most importantly, a distribution on the set of admissible instances has to be specified. In fact, we shall consider a new type of computational problems, each consisting of a standard computational problem coupled with a probability distribution on instances. Consequently, the question of which distributions should be considered in a theory of average-case

complexity arises. This question and numerous other definitional issues will be addressed in Section 2.1.

Before proceeding, let us spell out the rather straightforward motivation to the study of the average-case complexity of computational problems: It is that, in real-life applications, one may be perfectly happy with an algorithm that solves the problem fast on almost all instances that arise in the relevant application. That is, one may be willing to tolerate error provided that it occurs with negligible probability, where the probability is taken over the distribution of instances encountered in the application. The study of average-case complexity is aimed at exploring the possible benefit of such a relaxation, distinguishing cases in which a benefit exists from cases in which it does not exist. A key aspect in such a study is a good modeling of the type of distributions (of instances) that are encountered in natural algorithmic applications.

Let us consider the foregoing motivation from a slightly different perspective: The conjecture that $\mathcal{P} \neq \mathcal{NP}$ (or rather $\mathcal{NP} \not\subseteq \mathcal{BPP}$) only asserts that intractability is a feature of some instances of some problems in $\mathcal{NP}$. These intractable instances may be very rare and pathological. The theory of average-case complexity addresses the question of whether intractability can also be a feature of "typical" instances (i.e., whether intractable instances may occur with noticeable probability with respect to some simple distributions). Needless to say, the meaningfulness of the latter question depends on restricting the class of distributions such that only simple (rather than pathological) distributions are allowed. We shall consider two such classes of distributions (see Section 2.1 and Section 3.2, respectively) and show that if intractability occurs with respect to the wider class then it occurs also with respect to the more restricted class (see Theorem 14).

An average-case version of the $\mathcal{P} \neq \mathcal{NP}$ question. Indeed, a fundamental question that arises is *whether every natural computational problem can be solved efficiently when restricting attention to typical instances?* The conjecture that underlies this section is that, for a well-motivated choice of definitions, the answer is negative; that is, our conjecture is that the "distributional version" of NP is not contained in the average-case (or typical-case) version of P. This means that some NP problems are not merely hard in the worst-case, but are rather "typically hard" (i.e., hard on typical instances drawn from some simple distribution). This suggests that *hard instances may occur in natural algorithmic applications* (and not only in cryptographic (or other "adversarial") applications that are design on purpose to produce hard instances).[1]

[1] We highlight two differences between the current context (of natural algorithmic applications) and the context of cryptography. Firstly, in the current context and when referring to problems that are typically hard, the simplicity of the underlying input distribution is of great concern: the simpler this distribution, the more appealing the hardness assertion becomes. This concern is irrelevant in the context of cryptography. On the other hand (see, e.g., [5]), cryptographic applications require the ability to efficiently generate hard instances *together with corresponding solutions*.

The foregoing conjecture motivates the development of an average-case analogue of NP-completeness, which will be presented in this survey. In particular, this (average-case) theory identifies distributional problems that are "typically hard" provided that distributional problems that are "typically hard" exist at all. If one believes the foregoing conjecture then, for such complete (distributional) problems, one should not seek algorithms that solve these problems efficiently on typical instances.

1.2 Organization

A significant part of our exposition is devoted to the definitional issues that arise when developing a general theory of average-case complexity. These issues are discussed in Section 2.1. In Section 2.2 we prove the existence of distributional problems that are "NP-complete" in the corresponding average-case complexity sense. Furthermore, we show how to obtain such a distributional version for any natural NP-complete decision problem. In Section 2.3 we extend the treatment to randomized algorithms. Additional ramifications are presented in Section 3.

2 The Basic Theory

In this section we provide a basic treatment of the theory of average-case complexity, while postponing important ramifications to Section 3. The basic treatment consists of the preferred definitional choices for the main concepts as well as the identification of complete problems for a natural class of average-case computational problems.

2.1 Definitional issues

The theory of average-case complexity is more subtle than may appear at first thought. In addition to the generic conceptual difficulty involved in defining relaxations, difficulties arise from the "interface" between standard probabilistic analysis and the conventions of complexity theory. This is most striking in the definition of the class of feasible average-case computations. Referring to the theory of worst-case complexity as a guideline, we shall address the following aspects of the analogous theory of average-case complexity.

1. *Setting the general framework.* We shall consider distributional problems, which are standard computational problems coupled with distributions on the relevant instances.
2. *Identifying the class of feasible* (distributional) *problems.* Seeking an average-case analogue of classes such as $\mathcal{P}$, we shall reject the first definition that comes to mind (i.e., the naive notion of "average polynomial-time"), briefly discuss several related alternatives, and adopt one of them for the main treatment.

3. *Identifying the class of interesting* (distributional) *problems*. Seeking an average-case analogue of the class $\mathcal{NP}$, we shall avoid both the extreme of allowing arbitrary distributions (which collapses average-case hardness to worst-case hardness) and the opposite extreme of confining the treatment to a single distribution such as the uniform distribution.
4. *Developing an adequate notion of reduction among* (distributional) *problems*. As in the theory of worst-case complexity, this notion should preserve feasible solveability (in the current distributional context).

We now turn to the actual treatment of each of the aforementioned aspects.

Step 1: Defining distributional problems. Focusing on decision problems, we define distributional problems as pairs consisting of a decision problem and a probability ensemble.[2] For simplicity, here a probability ensemble $\{X_n\}_{n\in\mathbb{N}}$ is a sequence of random variables such that X_n ranges over $\{0,1\}^n$. Thus, $(S, \{X_n\}_{n\in\mathbb{N}})$ is the distributional problem consisting of the problem of deciding membership in the set S with respect to the probability ensemble $\{X_n\}_{n\in\mathbb{N}}$. (The treatment of search problem is similar; see Section 3.1.) We denote the uniform probability ensemble by $U = \{U_n\}_{n\in\mathbb{N}}$; that is, U_n is uniform over $\{0,1\}^n$.

Step 2: Identifying the class of feasible problems. The first idea that comes to mind is defining the problem $(S, \{X_n\}_{n\in\mathbb{N}})$ as feasible (on the average) if there exists an algorithm A that solves S such that the *average running time* of A on X_n is bounded by a polynomial in n (i.e., there exists a polynomial p such that $\mathsf{E}[t_A(X_n)] \leq p(n)$, where $t_A(x)$ denotes the running-time of A on input x). The problem with this definition is that it is very sensitive to the model of computation and is not closed under algorithmic composition. Both deficiencies are a consequence of the fact that t_A may be polynomial on the average with respect to $\{X_n\}_{n\in\mathbb{N}}$ but t_A^2 may fail to be so (e.g., consider $t_A(x'x'') = 2^{|x'|}$ if $x' = x''$ and $t_A(x'x'') = |x'x''|^2$ otherwise, coupled with the uniform distribution over $\{0,1\}^n$). We conclude that the *average running-time* of algorithms is not a robust notion. We also doubt the soundness of the appeal of this notion, and view the *typical running time* of algorithms (as defined next) as a more natural notion. Thus, we shall consider an algorithm as feasible if its running-time is typically polynomial.[3]

[2] We mention that even this choice is not evident. Specifically, Levin [10] (see discussion in [4]) advocates the use of a single probability distribution defined over the set of all strings. His argument is that this makes the theory less representation-dependent. At the time we were convinced of his argument (see [4]), but currently we feel that the representation-dependent effects discussed in [4] are legitimate. Furthermore, the alternative formulation of [10, 4] comes across as unnatural and tends to confuse some readers.

[3] An alternative choice, taken by Levin [10] (see discussion in [4]), is considering as feasible (w.r.t $X = \{X_n\}_{n\in\mathbb{N}}$) any algorithm that runs in time that is polynomial in a function that is linear on the average (w.r.t X); that is, requiring that there

We say that A is typically polynomial-time on $X = \{X_n\}_{n\in\mathbb{N}}$ if there exists a polynomial p such that the probability that A runs more that $p(n)$ steps on X_n is *negligible* (i.e., for every polynomial q and all sufficiently large n it holds that $\mathsf{Pr}[t_A(X_n) > p(n)] < 1/q(n)$). The question is what is required in the "untypical" cases, and two possible definitions follow.

1. The simpler option is saying that $(S, \{X_n\}_{n\in\mathbb{N}})$ is (typically) feasible if there exists an algorithm A that solves S such that A is typically polynomial-time on $X = \{X_n\}_{n\in\mathbb{N}}$. This effectively requires A to *correctly solve S on each instance*, which is more than was required in the motivational discussion. (Indeed, if the underlying motivation is ignoring rare cases, then we should ignore them altogether rather than ignoring them in a partial manner (i.e., only ignore their affect on the running-time).)
2. The alternative, which fits the motivational discussion, is saying that (S, X) is (typically) feasible if there exists an algorithm A such that A typically solves S on X in polynomial-time; that is, there exists a polynomial p such that *the probability that on input X_n algorithm A either errs or runs more that $p(n)$ steps is negligible*. This formulation totally ignores the untypical instances. Indeed, in this case we may assume, without loss of generality, that A always runs in polynomial-time, but we shall not do so here (in order to facilitate viewing the first option as a special case of the current option).

We stress that both alternatives actually define *typical* feasibility and not *average-case* feasibility. To illustrate the difference between the two options, consider the distributional problem of deciding whether a uniformly selected (n-vertex) graph is 3-colorable. Intuitively, this problem is "typically trivial" (with respect to the uniform distribution),[4] because the algorithm may always say `no` and be wrong with exponentially vanishing probability. Indeed, this trivial algorithm is admissible by the second approach, but not by the first approach. In light of the foregoing discussions, we adopt the second approach.

Definition 1 (the class tpc$\mathcal{P}$): *We say that A* typically solves *$(S, \{X_n\}_{n\in\mathbb{N}})$* in polynomial-time *if there exists a polynomial p such that the probability that on input X_n algorithm A either errs or runs more that $p(n)$ steps is negligible.*[5] *We denote by* tpc$\mathcal{P}$ *the class of distributional problems that are typically solvable in polynomial-time.*

---

exists a polynomial p and a function $\ell : \{0,1\}^* \to \mathbb{N}$ such that $t(x) \leq p(\ell(x))$ for every x and $\mathsf{E}[\ell(X_n)] = O(n)$. This definition is robust (i.e., it does not suffer from the aforementioned deficiencies) and is arguably as "natural" as the naive definition (i.e., $\mathsf{E}[t_A(X_n)] \leq \mathrm{poly}(n)$).

[4] In contrast, testing whether a given graph is 3-colorable seems "typically hard" for other distributions (see, e.g., Theorem 7). Needless to say, in the latter distributions both yes-instances and no-instances appear with noticeable probability.

[5] Recall that a function $\mu : \mathbb{N} \to \mathbb{N}$ is negligible if for every positive polynomial q and all sufficiently large n it holds that $\mu(n) < 1/q(n)$. We say that A errs on x if $A(x)$ differs from the indicator value of the predicate $x \in S$.

Clearly, for every $S \in \mathcal{P}$ and every probability ensemble X, it holds that $(S, X) \in \text{tpc}\mathcal{P}$. However, tpc$\mathcal{P}$ contains also distributional problems (S, X) with $S \notin \mathcal{P}$ (albeit this assertion refers to unnatural distributional versions of problems not in $\mathcal{P}$). The big question, which underlies the theory of average-case complexity, is whether all *natural distributional versions* of $\mathcal{NP}$ are in tpc$\mathcal{P}$. Thus, we turn to identify such versions.

Step 3: Identifying the class of interesting problems. Seeking to identify reasonable distributional versions of $\mathcal{NP}$, we note that two extreme choices should be avoided. On the one hand, we must limit the class of admissible distributions so as to prevent the collapse of average-case hardness to worst-case hardness (by a selection of a pathological distribution that resides on the "worst case" instances). On the other hand, we should allow for various types of natural distributions rather than confining attention merely to the uniform distribution.[6] Recall that our aim is addressing all possible input distributions that may occur in applications, and thus there is no justification for confining attention to the uniform distribution. Still, arguably, the distributions occuring in applications are "relatively simple" and so we seek to identify a class of simple distributions. One such notion (of simple distributions) underlies the following definition, while a more liberal notion will be presented in Section 3.2.

Definition 2 (the class dist$\mathcal{NP}$): *We say that a probability ensemble* $X = \{X_n\}_{n\in\mathbb{N}}$ *is* simple *if there exists a polynomial time algorithm that, on any input* $x \in \{0,1\}^*$*, outputs* $\mathsf{Pr}[X_{|x|} \leq x]$*, where the inequality refers to the standard lexicographic order of strings. We denote by* dist$\mathcal{NP}$ *the class of distributional problems consisting of decision problems in* $\mathcal{NP}$ *coupled with simple probability ensembles.*

Note that the uniform probability ensemble is simple, but so are many other "simple" probability ensembles. Actually, it makes sense to relax the definition such that the algorithm is only required to output an approximation of $\mathsf{Pr}[X_{|x|} \leq x]$, say, to within a factor of $1 \pm 2^{-2|x|}$. We note that Definition 2 interprets simplicity in computational terms; specifically, as the feasibility of answering very basic questions regarding the probability distribution (i.e., determining the probability mass assigned to a single (n-bit long) string and even to an interval of such strings).

Doudts regarding Definition 2. We admit that the identification of simple distributions as the class of interesting distribution is significantly more questionable than any other identification advocated in this book. Nevertheless, we believe that we were fully justified in rejecting both the aforementioned extremes (i.e.,

[6] Confining attention to the uniform distribution seems misguided by the naive belief according to which this distribution is the *only* one relevant to applications. In contrast, we believe that, for most natural applications, the uniform distribution over instances is not relevant at all.

of either allowing all distributions or allowing only the uniform distribution). Yet, the reader may wonder whether or not we have struck the right balance between "generality" and "simplicity" (in the intuitive sense). One specific concern is that we might have restricted the class of distributions too much. We briefly address this concern next.

A more intuitive and very robust class of distributions, which seems to contain all distributions that may occur in applications, is the class of polynomial-time sampleable probability ensembles (treated in Section 3.2). Fortunately, the combination of the results presented in Section 2.2 and Section 3.2 seems to retrospectively endorse the choice underlying Definition 2. Specifically, we note that enlarging the class of distributions weakens the *conjecture* that the corresponding class of distributional NP problems contains infeasible problems. On the other hand, the *conclusion* that a specific distributional problem is not feasible becomes more appealing when the problem belongs to a smaller class that corresponds to a restricted definition of admissible distributions. Now, the combined results of Section 2.2 and Section 3.2 assert that a conjecture that refers to the larger class of polynomial-time sampleable ensembles implies a conclusion that refers to a (very) simple probability ensemble (which resides in the smaller class). Thus, the current setting in which both the conjecture and the conclusion refer to simple probability ensembles may be viewed as just an intermediate step.

Does dist$\mathcal{NP}$ *contain only feasible problems?* Indeed, the big question in the current context is whether dist$\mathcal{NP}$ is contained in tpc$\mathcal{P}$. A positive answer (especially if extended to sampleable ensembles) would deem the P-vs-NP Question to be of little practical significant. However, our daily experience as well as much research effort indicate that some NP problems are not merely hard in the worst-case, but rather "typically hard". This leads to the *conjecture that* dist$\mathcal{NP}$ *is not contained in* tpc$\mathcal{P}$.

Needless to say, the latter conjecture implies $\mathcal{P} \neq \mathcal{NP}$, and thus we should not expect to see a proof of it. In particular, we should not expect to see a proof that some specific problem in dist$\mathcal{NP}$ is not in tpc$\mathcal{P}$. What we may hope to see is "dist$\mathcal{NP}$-complete" problems; that is, problems in dist$\mathcal{NP}$ that are not in tpc$\mathcal{P}$ unless the entire class dist$\mathcal{NP}$ is contained in tpc$\mathcal{P}$. An adequate notion of a reduction is used towards formulating this possibility.

Step 4: Defining reductions among (distributional) problems. Intuitively, such reductions must preserve average-case feasibility. Thus, in addition to the standard conditions (i.e., that the reduction be efficiently computable and yield a correct result), we require that the reduction "respects" the probability distribution of the corresponding distributional problems. Specifically, the reduction should not map very likely instances of the first ("starting") problem to rare instances of the second ("target") problem. Otherwise, having a typically polynomial-time algorithm for the second distributional problem does not necessarily yield such an algorithm for the first distributional problem. Following is the adequate analogue of a Cook reduction (i.e., general polynomial-time re-

duction), and the analogue of a Karp-reduction (many-to-one reduction) can be easily derived as a special case.[7]

Definition 3 (reductions among distributional problems): *We say that the oracle machine M* reduces *the distributional problem (S, X) to the distributional problem (T, Y) if the following three conditions hold.*

1. Efficiency: *The machine M runs in polynomial-time.*[8]
2. Validity: *For every $x \in \{0,1\}^*$, it holds that $M^T(x) = 1$ if an only if $x \in S$, where $M^T(x)$ denotes the output of the oracle machine M on input x and access to an oracle for T.*
3. Domination:[9] *The probability that, on input X_n and oracle access to T, machine M makes the query y is upper-bounded by* $\mathrm{poly}(|y|) \cdot \mathsf{Pr}[Y_{|y|} = y]$. *That is, there exists a polynomial p such that, for every $y \in \{0,1\}^*$ and every $n \in \mathbb{N}$, it holds that*

$$\mathsf{Pr}[Q(X_n) \ni y] \leq p(|y|) \cdot \mathsf{Pr}[Y_{|y|} = y], \tag{1}$$

where $Q(x)$ denotes the set of queries made by M on input x and oracle access to T.
In addition, we require that the reduction does not make too short queries; that is, there exists a polynomial p' such that if $y \in Q(x)$ then $p'(|y|) \geq |x|$.

In this case we say that the distributional problem (S, X) is reducible *to the distributional problem (T, Y).*

The l.h.s. of Eq. (1) refers to the probability that, on input distributed as X_n, the reduction makes the query y. This probability is required not to exceed the probability that y occurs in the distribution $Y_{|y|}$ by more than a polynomial factor in $|y|$. In this case we say that the l.h.s. of Eq. (1) is dominated by $\mathsf{Pr}[Y_{|y|} = y]$.

Indeed, the domination condition is the only aspect of Definition 3 that extends beyond the worst-case treatment of reductions and refers to the distributional setting. The domination condition does not insist that the distribution induced by $Q(X)$ equals Y, but rather allows some slackness that, in turn, is bounded so to guarantee preservation of typical feasibility. [10]

[7] See Footnote 9. We mention that the special case of many-to-one reductions, which suffices for the dist$\mathcal{NP}$-completeness results (e.g., Theorem 5).

[8] In fact, one may relax the requirement and only require that M is typically polynomial-time with respect to X. The validity condition may also be relaxed similarly.

[9] Let us spell out the meaning of Eq. (1) in the special case of many-to-one reductions (i.e., $M^T(x) = 1$ if and only if $f(x) \in T$, where f is a polynomial-time computable function): in this case $\mathsf{Pr}[Q(X_n) \ni y]$ is replaced by $\mathsf{Pr}[f(X_n) = y]$. That is, Eq. (1) simplifies to $\mathsf{Pr}[f(X_n) = y] \leq p(|y|) \cdot \mathsf{Pr}[Y_{|y|} = y]$. Indeed, this condition holds vacuously for any y that is not in the image of f.

[10] We stress that the notion of domination is incomparable to the notion of statistical (resp., computational) indistinguishability. On one hand, domination is a local re-

Proposition 4 (typical feasibility is preserved by reduction): *Suppose that the distributional problem (S, X) is reducible to the distributional problem (T, Y), and that $(T, Y) \in \text{tpc}\mathcal{P}$. Then, $(S, X) \in \text{tpc}\mathcal{P}$.*

Proof Sketch: Let M, Q, p, and p' be as in Definition 3, and suppose that A is an algorithm that typically solves (T, Y) in polynomial-time. Let B denote the set of instances on which A errs (or runs more than polynomial time), and $B_m \stackrel{\text{def}}{=} B \cap \{0,1\}^m$. By the domination condition, for every n, it holds that

$$\begin{aligned}
\mathsf{Pr}[M^A(X_n) \text{ errs}] &\leq \mathsf{Pr}[Q(X_n) \cap B \neq \emptyset] \\
&\leq \sum_{m:p'(m)\geq n} \sum_{y \in B_m} \mathsf{Pr}[Q(X_n) \ni y] \\
&\leq \sum_{m:p'(m)\geq n} \sum_{y \in B_m} p(m) \cdot \mathsf{Pr}[Y_m = y] \\
&\leq \sum_{m:p'(m)\geq n} p(m) \cdot \mathsf{Pr}[Y_m \in B_m]
\end{aligned}$$

where the second (resp., third) inequality uses the additional (resp., main) guarantee in the domination condition. It follows that the probability that M^A errs on X_n is negligible (as a function of n). □

Perspective. We note that the reducibility arguments that are extensively used in cryptopgraphy (see, e.g., [5, Chap. 2]) are actually reductions in the spirit of Definition 3 (except that they refer to different types of computational tasks).

2.2 Complete problems

Recall that our conjecture is that dist$\mathcal{NP}$ is not contained in tpc$\mathcal{P}$, which in turn strengthens the conjecture $\mathcal{P} \neq \mathcal{NP}$ (making infeasibility a typical phenomenon rather than a worst-case one). Having no hope of proving that dist$\mathcal{NP}$ is not contained in tpc$\mathcal{P}$, we turn to the study of complete problems with respect to that conjecture. Specifically, we say that a distributional problem (S, X) is dist$\mathcal{NP}$-complete if $(S, X) \in \text{dist}\mathcal{NP}$ and every $(S', X') \in \text{dist}\mathcal{NP}$ is reducible to (S, X) (under Definition 3).

Distributional Bounded Halting. Recall that it is quite easy to prove the mere existence of NP-complete problems and that many natural problems are

quirement (i.e., it compares the two distribution on a point-by-point basis), whereas indistinguishability is a global requirement (which allows rare exceptions). On the other hand, domination does not require approximately equal values, but rather a ratio that is bounded in one direction. Indeed, domination is not symmetric. We comment that a more relaxed notion of domination that allows rare violations (as in Footnote 8) suffices for the preservation of typical feasibility.

NP-complete. In contrast, in the current context, establishing completeness results is quite hard. This should not be surprising in light of the restricted type of reductions allowed in the current context. The restriction (captured by the domination condition) requires that "typical" instances of one problem should not be mapped to "untypical" instances of the other problem. In contrast, it is fair to say that standard Karp-reductions (used in establishing NP-completeness results) map "typical" instances of one problem to somewhat "bizarre" instances of the second problem. Thus, the current section may be viewed as a study of reductions that do not commit this sin.[11]

Theorem 5 (dist$\mathcal{NP}$-completeness): dist$\mathcal{NP}$ *contains a distributional problem* (T, Y) *such that each distributional problem in* dist$\mathcal{NP}$ *is reducible* (per Definition 3) *to* (T, Y). *Furthermore, the reductions are via many-to-one mappings.*

Proof: We start by introducing such a (distributional) problem, which is a natural distributional version of the "universal decision problem", denoted $S_{\mathtt{u}}$, and often referred to as *Bounded Halting*. Specifically, we define $S_{\mathtt{u}}$ such that the instance $\langle M, x, 1^t\rangle$ is in $S_{\mathtt{u}}$ if there exists $y \in \cup_{i\leq t}\{0,1\}^i$ such that machine M accepts the input pair (x, y) within t steps. We couple $S_{\mathtt{u}}$ with the "quasi-uniform" probability ensemble U' that assigns to the instance $\langle M, x, 1^t\rangle$ a probability mass proportional to $2^{-(|M|+|x|)}$. Specifically, for every $\langle M, x, 1^t\rangle$ it holds that

$$\mathsf{Pr}[U'_n = \langle M, x, 1^t\rangle] = \frac{2^{-(|M|+|x|)}}{\binom{n}{2}} \tag{2}$$

where $n \stackrel{\text{def}}{=} |\langle M, x, 1^t\rangle| \stackrel{\text{def}}{=} |M| + |x| + t$. Note that, under a suitable natural encoding, the ensemble U' is indeed simple.[12]

The reader can easily verify that the generic reduction used when reducing any set in $\mathcal{NP}$ to $S_{\mathtt{u}}$ (see the proof of [6, Thm. 2.19]), fails to reduce dist$\mathcal{NP}$ to $(S_{\mathtt{u}}, U')$. Specifically, in some cases (see next paragraph), these reductions do not satisfy the domination condition. Indeed, the difficulty is that we have to reduce all dist$\mathcal{NP}$ problems (i.e., pairs consisting of decision problems and simple distributions) to one single distributional problem (i.e., $(S_{\mathtt{u}}, U')$). In contrast, considering the distributions induced by the aforementioned reductions, we end up with many distributional versions of $S_{\mathtt{u}}$, and furthermore the corresponding distributions are very different (and are not necessarily dominated by a single distribution).

---

[11] The latter assertion is somewhat controversial. While this assertion seems totally justified with respect to the proof of Theorem 5, opinions regarding the proof of Theorem 7 may differ.

[12] For example, we may encode $\langle M, x, 1^t\rangle$, where $M = \sigma_1 \cdots \sigma_k \in \{0,1\}^k$ and $x = \tau_1 \cdots \tau_\ell \in \{0,1\}^\ell$, by the string $\sigma_1\sigma_1 \cdots \sigma_k\sigma_k 01\tau_1\tau_1 \cdots \tau_\ell\tau_\ell 01^t$. Then $\binom{n}{2} \cdot \mathsf{Pr}[U'_n \leq \langle M, x, 1^t\rangle]$ equals $(i_{|M|,|x|,t} - 1) + 2^{-|M|} \cdot |\{M' \in \{0,1\}^{|M|} : M' < M\}| + 2^{-(|M|+|x|)} \cdot |\{x' \in \{0,1\}^{|x|} : x' \leq x\}|$, where $i_{k,\ell,t}$ is the ranking of $\{k, k+\ell\}$ among all 2-subsets of $[k + \ell + t]$.

Let us take a closer look at the aforementioned generic reduction (of S to $S_{\mathtt{u}}$), when applied to an arbitrary $(S,X) \in \mathrm{dist}\mathcal{NP}$. This reduction maps an instance x to a triple $(M_S, x, 1^{p_S(|x|)})$, where M_S is a machine verifying membership in S (while using adequate NP-witnesses) and p_S is an adequate polynomial. The problem is that x may have relatively large probability mass (i.e., it may be that $\mathsf{Pr}[X_{|x|}=x] \gg 2^{-|x|}$) while $(M_S, x, 1^{p_S(|x|)})$ has "uniform" probability mass (i.e., $\langle M_S, x, 1^{p_S(|x|)}\rangle$ has probability mass smaller than $2^{-|x|}$ in U'). This violates the domination condition, and thus an alternative reduction is required.

The key to the alternative reduction is an (efficiently computable) encoding of strings taken from an arbitrary *simple* distribution by strings that have a similar probability mass under the uniform distribution. This means that the encoding should shrink strings that have relatively large probability mass under the original distribution. Specifically, this encoding will map x (taken from the ensemble $\{X_n\}_{n\in\mathbb{N}}$) to a codeword x' of length that is upper-bounded by the logarithm of $1/\mathsf{Pr}[X_{|x|}=x]$, ensuring that $\mathsf{Pr}[X_{|x|}=x] = O(2^{-|x'|})$. Accordingly, the reduction will map x to a triple $(M_{S,X}, x', 1^{p'(|x|)})$, where $|x'| < O(1)+\log_2(1/\mathsf{Pr}[X_{|x|}=x])$ and $M_{S,X}$ is an algorithm that (given x' and x) first verifies that x' is a proper encoding of x and next applies the standard verification (i.e., M_S) of the problem S. Such a reduction will be shown to satisfy all three conditions (i.e., efficiency, validity, and domination). Thus, instead of forcing the structure of the original distribution X on the target distribution U', the reduction will incorporate the structure of X in the reduced instance. A key ingredient in making this possible is the fact that X is simple (as per Definition 2).

With the foregoing motivation in mind, we now turn to the actual proof; that is, proving that any $(S,X) \in \mathrm{dist}\mathcal{NP}$ is reducible to $(S_{\mathtt{u}}, U')$. The following technical lemma is the basis of the reduction. In this lemma as well as in the sequel, it will be convenient to consider the (accumulative) distribution function of the probability ensemble X. That is, we consider $\mu(x) \stackrel{\mathrm{def}}{=} \mathsf{Pr}[X_{|x|} \leq x]$, and note that $\mu : \{0,1\}^* \to [0,1]$ is polynomial-time computable (because X satisfies Definition 2).

Coding Lemma:[13] Let $\mu : \{0,1\}^* \to [0,1]$ be a polynomial-time computable function that is monotonically non-decreasing over $\{0,1\}^n$ for every n (i.e., $\mu(x') \leq \mu(x'')$ for any $x' < x'' \in \{0,1\}^{|x'|}$). For $x \in \{0,1\}^n \setminus \{0^n\}$, let $x-1$ denote the string preceding x in the lexicographic order of n-bit long strings. Then there exist an encoding function C_μ that satisfies the following three conditions.

1. Compression: For every x it holds that $|C_\mu(x)| \leq 1+\min\{|x|, \log_2(1/\mu'(x))\}$, where $\mu'(x) \stackrel{\mathrm{def}}{=} \mu(x) - \mu(x-1)$ if $x \notin \{0\}^*$ and $\mu'(0^n) \stackrel{\mathrm{def}}{=} \mu(0^n)$ otherwise.
2. Efficient Encoding: The function C_μ is computable in polynomial-time.

[13] The lemma actually refers to $\{0,1\}^n$, for any fixed value of n, but the efficiency condition is stated more easily when allowing n to vary (and using the standard asymptotic analysis of algorithms). Actually, the lemma is somewhat easier to state and establish for polynomial-time computable functions that are monotonically non-decreasing over $\{0,1\}^*$ (rather than over $\{0,1\}^n$); see [4, Sec. 3].

3. Unique Decoding: For every $n \in \mathbb{N}$, when restricted to $\{0,1\}^n$, the function C_μ is one-to-one (i.e., if $C_\mu(x) = C_\mu(x')$ and $|x| = |x'|$ then $x = x'$).

Proof: The function C_μ is defined as follows. If $\mu'(x) \leq 2^{-|x|}$ then $C_\mu(x) = 0x$ (i.e., in this case x serves as its own encoding). Otherwise (i.e., $\mu'(x) > 2^{-|x|}$) then $C_\mu(x) = 1z$, where z is chosen such that $|z| \leq \log_2(1/\mu'(x))$ and the mapping of n-bit strings to their encoding is one-to-one. Loosely speaking, z is selected to equal the shortest binary expansion of a number in the interval $(\mu(x) - \mu'(x), \mu(x)]$. Bearing in mind that this interval has length $\mu'(x)$ and that the different intervals are disjoint, we obtain the desired encoding. Details follows.

We focus on the case that $\mu'(x) > 2^{-|x|}$, and detail the way that z is selected (for the encoding $C_\mu(x) = 1z$). If $x > 0^{|x|}$ and $\mu(x) < 1$, then we let z be the longest common prefix of the binary expansions of $\mu(x-1)$ and $\mu(x)$; for example, if $\mu(1010) = 0.10010$ and $\mu(1011) = 0.10101111$ then $C_\mu(1011) = 1z$ with $z = 10$. Thus, in this case $0.z1$ is in the interval $(\mu(x-1), \mu(x)]$ (i.e., $\mu(x-1) < 0.z1 \leq \mu(x)$). For $x = 0^{|x|}$, we let z be the longest common prefix of the binary expansions of 0 and $\mu(x)$ and again $0.z1$ is in the relevant interval (i.e., $(0, \mu(x)]$). Finally, for x such that $\mu(x) = 1$ and $\mu(x-1) < 1$, we let z be the longest common prefix of the binary expansions of $\mu(x-1)$ and $1 - 2^{-|x|-1}$, and again $0.z1$ is in $(\mu(x-1), \mu(x)]$ (because $\mu'(x) > 2^{-|x|}$ and $\mu(x-1) < \mu(x) = 1$ imply that $\mu(x-1) < 1 - 2^{-|x|} < \mu(x)$). Note that if $\mu(x) = \mu(x-1) = 1$ then $\mu'(x) = 0 < 2^{-|x|}$.

We now verify that the foregoing C_μ satisfies the conditions of the lemma. We start with the compression condition. Clearly, if $\mu'(x) \leq 2^{-|x|}$ then $|C_\mu(x)| = 1 + |x| \leq 1 + \log_2(1/\mu'(x))$. On the other hand, suppose that $\mu'(x) > 2^{-|x|}$ and let us focus on the sub-case that $x > 0^{|x|}$ and $\mu(x) < 1$. Let $z = z_1 \cdots z_\ell$ be the longest common prefix of the binary expansions of $\mu(x-1)$ and $\mu(x)$. Then, $\mu(x-1) = 0.z0u$ and $\mu(x) = 0.z1v$, where $u, v \in \{0,1\}^*$. We infer that

$$\mu'(x) \;=\; \mu(x) - \mu(x-1) \;\leq\; \left(\sum_{i=1}^{\ell} 2^{-i} z_i + \sum_{i=\ell+1}^{\mathrm{poly}(|x|)} 2^{-i} \right) - \sum_{i=1}^{\ell} 2^{-i} z_i \;<\; 2^{-|z|},$$

and $|z| < \log_2(1/\mu'(x)) \leq |x|$ follows. Thus, $|C_\mu(x)| \leq 1 + \min(|x|, \log_2(1/\mu'(x)))$ holds in both cases. Clearly, C_μ can be computed in polynomial-time by computing $\mu(x-1)$ and $\mu(x)$. Finally, note that C_μ satisfies the unique decoding condition, by separately considering the two aforementioned cases (i.e., $C_\mu(x) = 0x$ and $C_\mu(x) = 1z$). Specifically, in the second case (i.e., $C_\mu(x) = 1z$), use the fact that $\mu(x-1) < 0.z1 \leq \mu(x)$. □

In order to obtain an encoding that is one-to-one when applied to strings of different lengths, we augment C_μ in the obvious manner; that is, we consider $C'_\mu(x) \stackrel{\mathrm{def}}{=} (|x|, C_\mu(x))$, which may be implemented as $C'_\mu(x) = \sigma_1\sigma_1 \cdots \sigma_\ell\sigma_\ell 01 C_\mu(x)$ where $\sigma_1 \cdots \sigma_\ell$ is the binary expansion of $|x|$. Note that $|C'_\mu(x)| = O(\log |x|) + |C_\mu(x)|$ and that C'_μ is one-to-one (over $\{0,1\}^*$).

The machine associated with (S, X). Let μ be the accumulative probability function associated with the probability ensemble X, and M_S be the polynomial-time machine that verifies membership in S while using adequate NP-witnesses (i.e., $x \in S$ if and only if there exists $y \in \{0,1\}^{\mathrm{poly}(|x|)}$ such that $M(x,y) = 1$). Using the encoding function C'_μ, we introduce an algorithm $M_{S,\mu}$ with the intension of reducing the distributional problem (S, X) to $(S_{\mathtt{u}}, U')$ such that all instances (of S) are mapped to triples in which the first element equals $M_{S,\mu}$. Machine $M_{S,\mu}$ is given an alleged encoding (under C'_μ) of an instance to S along with an alleged proof that the corresponding instance is in S, and verifies these claims in the obvious manner. That is, on input x' and $\langle x, y\rangle$, machine $M_{S,\mu}$ first verifies that $x' = C'_\mu(x)$, and next verifiers that $x \in S$ by running $M_S(x,y)$. Thus, $M_{S,\mu}$ verifies membership in the set $S' = \{C'_\mu(x) : x \in S\}$, while using proofs of the form $\langle x, y\rangle$ such that $M_S(x,y) = 1$ (for the instance $C'_\mu(x)$).[14]

The reduction. We maps an instance x (of S) to the triple $(M_{S,\mu}, C'_\mu(x), 1^{p(|x|)})$, where $p(n) \stackrel{\mathrm{def}}{=} p_S(n) + p_C(n)$ such that p_S is a polynomial representing the running-time of M_S and p_C is a polynomial representing the running-time of the encoding algorithm.

Analyzing the reduction. Our goal is proving that *the foregoing mapping constitutes a reduction of* (S, X) *to* $(S_{\mathtt{u}}, U')$. We verify the corresponding three requirements (of Definition 3).

1. Using the fact that C'_μ is polynomial-time computable (and noting that p is a polynomial), it follows that the foregoing mapping can be computed in polynomial-time.
2. Recall that, on input $(x', \langle x, y\rangle)$, machine $M_{S,\mu}$ accepts if and only if $x' = C'_\mu(x)$ and M_S accepts (x,y) within $p_S(|x|)$ steps. Using the fact that $C'_\mu(x)$ uniquely determines x, it follows that $x \in S$ if and only if $C'_\mu(x) \in S'$, which in turn holds if and only if there exists a string y such that $M_{S,\mu}$ accepts $(C'_\mu(x), \langle x, y\rangle)$ in at most $p(|x|)$ steps. Thus, $x \in S$ if and only if $(M_{S,\mu}, C'_\mu(x), 1^{p(|x|)}) \in S_{\mathtt{u}}$, and the validity condition follows.
3. In order to verify the domination condition, we first note that the foregoing mapping is one-to-one (because the transformation $x \to C'_\mu(x)$ is one-to-one). Next, we note that it suffices to consider instances of $S_{\mathtt{u}}$ that have a preimage under the foregoing mapping (since instances with no preimage trivially satisfy the domination condition). Each of these instances (i.e., each image of this mapping) is a triple with the first element equal to $M_{S,\mu}$ and the second element being an encoding under C'_μ. By the definition of U', for every such image $\langle M_{S,\mu}, C'_\mu(x), 1^{p(|x|)}\rangle \in \{0,1\}^n$, it holds that

$$\begin{aligned}\mathsf{Pr}[U'_n = \langle M_{S,\mu}, C'_\mu(x), 1^{p(|x|)}\rangle] &= \binom{n}{2}^{-1} \cdot 2^{-(|M_{S,\mu}|+|C'_\mu(x)|)} \\ &> c \cdot n^{-2} \cdot 2^{-(|C_\mu(x)|+O(\log|x|))},\end{aligned}$$

[14] Note that $|y| = \mathrm{poly}(|x|)$, but $|x| = \mathrm{poly}(|C'_\mu(x)|)$ does not necessarily hold (and so S' is not necessarily in $\mathcal{NP}$). As we shall see, the latter point is immaterial.

where $c = 2^{-|M_{S,\mu}|-1}$ is a constant depending only on S and μ (i.e., on the distributional problem (S, X)). Thus, for some positive polynomial q, we have

$$\mathsf{Pr}[U'_n = \langle M_{S,\mu}, C'_\mu(x), 1^{p(|x|)}\rangle] > 2^{-|C_\mu(x)|}/q(n). \quad (3)$$

By virtue of the compression condition (of the Coding Lemma), we have $2^{-|C_\mu(x)|} \geq 2^{-1-\min(|x|,\log_2(1/\mu'(x)))}$. It follows that

$$2^{-|C_\mu(x)|} \geq \mathsf{Pr}[X_{|x|} = x]/2. \quad (4)$$

Recalling that x is the only preimage that is mapped to $\langle M_{S,\mu}, C'_\mu(x), 1^{p(|x|)}\rangle$ and combining Eq. (3) & (4), we establish the domination condition.

The theorem follows. ■

Reflections: The proof of Theorem 5 highlights the fact that the reduction used in establishing the NP-completeness of $S_{\mathtt{u}}$ does not introduce much structure in the reduced instances (i.e., does not reduce the original problem to a "highly structured special case" of the target problem). Put in other words, unlike more advanced worst-case reductions, this reduction does not map "random" (i.e., uniformly distributed) instances to highly structured instances (which occur with negligible probability under the uniform distribution). Thus, the reduction used in establishing the NP-completeness of $S_{\mathtt{u}}$ suffices for reducing any distributional problem in dist$\mathcal{NP}$ to a distributional problem consisting of $S_{\mathtt{u}}$ coupled with *some* simple probability ensemble. [15]

However, Theorem 5 states more than the latter assertion. That is, it states that any distributional problem in dist$\mathcal{NP}$ is reducible to the *same* distributional version of $S_{\mathtt{u}}$. Indeed, the effort involved in proving Theorem 5 was due to the need for mapping instances taken from any simple probability ensemble (which may not be the uniform ensemble) to instances distributed in a manner that is dominated by a single probability ensemble (i.e., the quasi-uniform ensemble U').

Other dist$\mathcal{NP}$-complete problems. Once we have established the existence of one dist$\mathcal{NP}$-complete problem, we may establish the dist$\mathcal{NP}$-completeness of other problems (in dist$\mathcal{NP}$) by reducing some dist$\mathcal{NP}$-complete problem to them (and relying on the transitivity of reductions).[16] Thus, the difficulties encountered in the proof of Theorem 5 are no longer relevant. Unfortunately, a seemingly more severe difficulty arises: almost all known reductions in the theory of NP-completeness work by introducing much structure in the reduced instances (i.e., they actually reduce to highly structured special cases). Furthermore, this structure is too complex in the sense that the distribution of reduced instances

[15] Note that this cannot be said of most known Karp-reductions, which do map random instances to highly structured ones.

[16] When establishing the transitivity of reductions, it is again essential to use the additional guarantee in the domination condition. Compare Proposition 4.

does not seem simple (in the sense of Definition 2). Actually, as demonstrated next, the problem is not the existence of a structure in the reduced instances but rather the complexity of this structure. In particular, if the aforementioned reduction is "monotone" and "length regular" then the distribution of the reduced instances is simple enough (i.e., is simple in the sense of Definition 2):

Proposition 6 (sufficient condition for dist$\mathcal{NP}$-completeness): *Suppose that f is a Karp-reduction of the set S to the set T such that, for every $x', x'' \in \{0,1\}^*$, the following two conditions hold:*

1. (f is monotone): *If $x' < x''$ then $f(x') < f(x'')$, where the inequalities refer to the standard lexicographic order of strings.*[17]
2. (f is length-regular): $|x'| = |x''|$ *if and only if* $|f(x')| = |f(x'')|$.

Then, if there exists an ensemble X such that (S, X) is dist$\mathcal{NP}$*-complete, then there exists an ensemble Y such that (T, Y) is* dist$\mathcal{NP}$*-complete.*

Proof Sketch: Note that the monotonicity of f implies that f is one-to-one and that for every x it holds that $f(x) \geq x$. Furthermore, as shown next, f is polynomial-time invertible. Intuitively, the fact that f is both monotone and polynomial-time computable implies that a preimage can be found by a binary search. Specifically, given $y = f(x)$, we search for x by iteratively halving the interval of potential solutions, which is initialized to $[0, y]$ (since $x \leq f(x)$). Note that if this search is invoked on a string y that is not in the image of f, then it terminates while detecting this fact.

Relying on the fact that f is one-to-one (and length-regular), we define the probability ensemble $Y = \{Y_n\}_{n\in\mathbb{N}}$ such that for every x it holds that $\mathsf{Pr}[Y_{|f(x)|} = f(x)] = \mathsf{Pr}[X_{|x|} = x]$. Specifically, letting $\ell(m) = |f(1^m)|$ and noting that ℓ is one-to-one and monotonically non-decreasing, we define

$$\mathsf{Pr}[Y_{|y|}=y] = \begin{cases} \mathsf{Pr}[X_{|x|}=x] & \text{if } x = f^{-1}(y) \\ 0 & \text{if } \exists m \text{ s.t. } y \in \{0,1\}^{\ell(m)} \setminus \{f(x) : x \in \{0,1\}^m\} \\ 2^{-|y|} & \text{otherwise (i.e., if } |y| \notin \{\ell(m) : m \in \mathbb{N}\})^{18}. \end{cases}$$

Clearly, (S, X) is reducible to (T, Y) (via the Karp-reduction f, which, due to our construction of Y, also satisfies the domination condition). Thus, using the hypothesis that dist$\mathcal{NP}$ is reducible to (S, X) and the transitivity of reductions, it follows that every problem in dist$\mathcal{NP}$ is reducible to (T, Y). The key observation, to be established next, is that Y is a simple probability ensemble, and it follows that (T, Y) is in dist$\mathcal{NP}$.

Loosely speaking, the simplicity of Y follows by combining the simplicity of X and the properties of f (i.e., the fact that f is monotone, length-regular, and

[17] In particular, if $|z'| < |z''|$ then $z' < z''$. Recall that for $|z'| = |z''|$ it holds that $z' < z''$ if and only if there exists $w, u', u'' \in \{0,1\}^*$ such that $z' = w0u'$ and $z'' = w1u''$.

[18] Having Y_n be uniform in this case is a rather arbitrary choice, which is merely aimed at guaranteeing a "simple" distribution on n-bit strings (also in this case).

polynomial-time invertible). The monotonicity and length-regularity of f implies that $\mathsf{Pr}[Y_{|f(x)|} \leq f(x)] = \mathsf{Pr}[X_{|x|} \leq x]$. More generally, for any $y \in \{0,1\}^{\ell(m)}$, it holds that $\mathsf{Pr}[Y_{\ell(m)} \leq y] = \mathsf{Pr}[X_m \leq x]$, where x is the lexicographicly largest string such that $f(x) \leq y$ (and, indeed, if $|x| < m$ then $\mathsf{Pr}[Y_{\ell(m)} \leq y] = \mathsf{Pr}[X_m \leq x] = 0$).[19] Note that this x can be found in polynomial-time by the inverting algorithm sketched in the first paragraph of the proof. Thus, we may compute $\mathsf{Pr}[Y_{|y|} \leq y]$ by finding the adequate x and computing $\mathsf{Pr}[X_{|x|} \leq x]$. Using the hypothesis that X is simple, it follows that Y is simple (and the proposition follows). □

On the existence of adequate Karp-reductions. Proposition 6 implies that a sufficient condition for the dist$\mathcal{NP}$-completeness of a distributional version of a (NP-complete) set T is the existence of an adequate Karp-reduction from the set $S_{\mathtt{u}}$ to the set T; that is, this Karp-reduction should be monotone and length-regular. While the length-regularity condition seems easy to impose (by using adequate padding), the monotonicity condition seems more problematic. Fortunately, it turns out that the monotonicity condition can also be imposed by using adequate padding (or rather an adequate "marking" – see [6, Exer. 2.30] and [6, Exer. 10.21]. We highlight the fact that the existence of an adequate padding (or "marking") is a property of the set T itself, and mention that all popular NP-complete sets satisfy it. Observing that any Karp-reduction to a "monotonically markable" set T can be transformed into a Karp-reduction (to T) that is monotone and length-regular, we conclude that *any natural NP-complete decision problem can be coupled with a simple probability ensemble such that the resulting distributional problem is* dist$\mathcal{NP}$*-complete.* As a concrete illustration of this thesis, we state the corresponding (formal) result for the twenty-one NP-complete problems treated in Karp's paper on NP-completeness [9].

Theorem 7 (a modest version of a general thesis): *For each of the twenty-one NP-complete problems treated in [9] there exists a simple probability ensemble such that the combined distributional problem is* dist$\mathcal{NP}$*-complete.*

The said list of problems includes `SAT`, `Clique`, and `3-Colorability`.

2.3 Probabilistic versions

The definitions in Section 2.1 can be extended so to account for randomized computations. For example, extending Definition 1, we have:

Definition 8 (the class tpc$\mathcal{BPP}$): *For a probabilistic algorithm A, a Boolean function f, and a time-bound function $t : \mathbb{N} \to \mathbb{N}$, we say that the string x is t-*bad for *A* with respect to *f if with probability exceeding $1/3$, on input x, either $A(x) \neq f(x)$ or A runs more that $t(|x|)$ steps. We say that A* typically solves

[19] We also note that the case in which $|y|$ is not in the image of ℓ can be easily detected and taken care off accordingly.

$(S, \{X_n\}_{n\in\mathbb{N}})$ in probabilistic polynomial-time *if there exists a polynomial p such that the probability that X_n is p-bad for A with respect to the characteristic function of S is negligible. We denote by* tpc$\mathcal{BPP}$ *the class of distributional problems that are typically solvable in probabilistic polynomial-time.*

The definition of reductions can be similarly extended. This means that in Definition 3, both $M^T(x)$ and $Q(x)$ (mentioned in Items 2 and 3, respectively) are random variables rather than fixed objects. Furthermore, validity is required to hold (for every input) only with probability 2/3, where the probability space refers only to the internal coin tosses of the reduction. Randomized reductions are closed under composition and preserve typical feasibility.

Randomized reductions allow the presentation of a dist$\mathcal{NP}$-complete problem that refers to the (perfectly) uniform ensemble. Recall that Theorem 5 establishes the dist$\mathcal{NP}$-completeness of $(S_{\mathtt{u}}, U')$, where U' is a quasi-uniform ensemble (i.e., $\mathsf{Pr}[U'_n = \langle M, x, 1^t\rangle] = 2^{-(|M|+|x|)}/\binom{n}{2}$, where $n = |\langle M, x, 1^t\rangle|$). We first note that $(S_{\mathtt{u}}, U')$ can be randomly reduced to $(S'_{\mathtt{u}}, U'')$, where $S'_{\mathtt{u}} = \{\langle M, x, z\rangle : \langle M, x, 1^{|z|}\rangle \in S_{\mathtt{u}}\}$ and $\mathsf{Pr}[U''_n = \langle M, x, z\rangle] = 2^{-(|M|+|x|+|z|)}/\binom{n}{2}$ for every $\langle M, x, z\rangle \in \{0,1\}^n$. The randomized reduction consists of mapping $\langle M, x, 1^t\rangle$ to $\langle M, x, z\rangle$, where z is uniformly selected in $\{0,1\}^t$. Recalling that $U = \{U_n\}_{n\in\mathbb{N}}$ denotes the uniform probability ensemble (i.e., U_n is uniformly distributed on strings of length n) and using a suitable encoding we get.

Proposition 9 (dist$\mathcal{NP}$-completeness w.r.t the uniform distribition): *There exists $S \in \mathcal{NP}$ such that every $(S', X') \in$ dist$\mathcal{NP}$ is randomly reducible to (S, U).*

Proof Sketch: By the forgoing discussion, every $(S', X') \in$ dist$\mathcal{NP}$ is randomly reducible to $(S'_{\mathtt{u}}, U'')$, where the reduction goes through $(S_{\mathtt{u}}, U')$. Thus, we focus on reducing $(S'_{\mathtt{u}}, U'')$ to $(S''_{\mathtt{u}}, U)$, where $S''_{\mathtt{u}} \in \mathcal{NP}$ is defined as follows. The string $\mathrm{bin}_\ell(|u|)\cdot\mathrm{bin}_\ell(|v|)\cdot u\cdot v\cdot w$ is in $S''_{\mathtt{u}}$ if and only if $\langle u, v, w\rangle \in S'_{\mathtt{u}}$ and $\ell = \lceil\log_2|uvw|\rceil + 1$, where $\mathrm{bin}_\ell(i)$ denotes the ℓ-bit long binary encoding of the integer $i \in [2^{\ell-1}]$ (i.e., the encoding is padded with zeros to a total length of ℓ). The reduction maps $\langle M, x, z\rangle$ to the string $\mathrm{bin}_\ell(|x|)\cdot\mathrm{bin}_\ell(|M|)\cdot M\cdot x\cdot z$, where $\ell = \lceil\log_2(|M| + |x| + |z|)\rceil + 1$. Noting that this reduction satisfies all conditions of Definition 3, the proposition follows. □

3 Ramifications

In our opinion, the most problematic aspect of the theory described in Section 2 is the choice to focus on simple probability ensembles, which in turn restricts "distributional versions of NP" to the class dist$\mathcal{NP}$ (Definition 2). As indicated Section 2.1, this restriction raises two opposite concerns (i.e., that dist$\mathcal{NP}$ is either too wide or too narrow).[20] Here we address the concern that the class of simple probability ensembles is too restricted, and consequently that the conjecture

[20] On one hand, if the definition of dist$\mathcal{NP}$ were too liberal, then membership in dist$\mathcal{NP}$ would mean less than one may desire. On the other hand, if dist$\mathcal{NP}$ were too restricted, then the conjecture that dist$\mathcal{NP}$ contains hard problems would have been very questionable.

dist$\mathcal{NP}$ $\not\subseteq$ tpc$\mathcal{BPP}$ is too strong (which would mean that dist$\mathcal{NP}$-completeness is a weak evidence for typical-case hardness). An appealing extension of the class of simple probability ensembles is presented in Section 3.2, yielding an corresponding extension of dist$\mathcal{NP}$, and it is shown that *if this extension of* dist$\mathcal{NP}$ *is not contained in* tpc$\mathcal{BPP}$*, then* dist$\mathcal{NP}$ *itself is not contained in* tpc$\mathcal{BPP}$. Consequently, dist$\mathcal{NP}$-complete problems enjoy the benefit of both being in the more restricted class (i.e., dist$\mathcal{NP}$) and being hard as long as some problems in the extended class is hard.

A different extension appears in Section 3.1, where we extend the treatment from decision problems to search problems. This extension is motivated by the realization that search problem are actually of greater importance to real-life applications (see, e.g., discussions in [6, Sec. 2.1.1]) and hence a theory motivated by real-life applications must address such problems, as we do next.

Prerequisites: For the technical development of Section 3.1, we assume familiarity with the notion of unique solution and results regarding it (see, e.g., [6, Sec. 6.2.3]). For the technical development of Section 3.2, we assume familiarity with hashing functions (see, e.g., [6, Apdx. D.2]). In addition, the technical development of Section 3.2 relies on Section 3.1.

3.1 Search versus Decision

Indeed, as in the case of worst-case complexity, search problems are at least as important as decision problems. Thus, an average-case treatment of search problems is indeed called for. We first present distributional versions of the search problem classes $\mathcal{PF}$ and $\mathcal{PC}$ (which correspond to $\mathcal{P}$ and $\mathcal{NP}$, resp.),[21] following the underlying principles of the definitions of tpc$\mathcal{P}$ and dist$\mathcal{NP}$.

Definition 10 (the classes tpc$\mathcal{PF}$ and dist$\mathcal{PC}$): *We consider only polynomially bounded search problems; that is, binary relations $R \subseteq \{0,1\}^* \times \{0,1\}^*$ such that for some polynomial q it holds that $(x,y) \in R$ implies $|y| \leq q(|x|)$. We use the notation $R(x) \stackrel{\text{def}}{=} \{y : (x,y) \in R\}$ and $S_R \stackrel{\text{def}}{=} \{x : R(x) \neq \emptyset\}$.*

- *A* distributional search problem *consists of a polynomially bounded search problem coupled with a probability ensemble.*
- *The class* tpc$\mathcal{PF}$ *consists of all distributional search problems that are typically solvable in polynomial-time. That is, $(R, \{X_n\}_{n\in\mathbb{N}}) \in$ tpc$\mathcal{PF}$ if there exists an algorithm A and a polynomial p such that the probability that on*

[21] Specifically $\mathcal{PF}$ (standing for Polynomial-time Find) is the class of efficiently solvable search problems; that is, $R \in \mathcal{PF}$ if there exists a polynomial-time algorithm that on input x replies with $y \in R(x) \stackrel{\text{def}}{=} \{z : (x,z) \in R\}$ (and with $\perp$ if $R(x) = \emptyset$). The class $\mathcal{PC}$ (standing for Polynomial-time Check) is the class of search problems having efficiently checkable solutions; that is, the search problem of a polynomially bounded relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is in $\mathcal{PC}$ if there exists a polynomial time algorithm A such that, for every x and y, it holds that $A(x,y) = 1$ if and only if $(x,y) \in R$. For more deatils, see [6, Sec. 2.1.1].

input X_n algorithm A either errs or runs more that $p(n)$ steps is negligible, where A errs on $x \in S_R$ if $A(x) \notin R(x)$ and errs on $x \notin S_R$ if $A(x) \neq \perp$.
- *A distributional search problem (R, X) is in* dist$\mathcal{PC}$ *if $R \in \mathcal{PC}$ and X is simple* (as in Definition 2).

Likewise, the class tpc$\mathcal{BPPF}$ consists of all distributional search problems that are typically solvable in *probabilistic* polynomial-time (cf., Definition 8). The definitions of *reductions among distributional problems*, presented in the context of decision problem, extend to search problems.

Fortunately, as in the context of worst-case complexity, the study of distributional search problems "reduces" to the study of distributional decision problems.

Theorem 11 (reducing search to decision): dist$\mathcal{PC} \subseteq$ tpc$\mathcal{BPPF}$ *if and only if* dist$\mathcal{NP} \subseteq$ tpc$\mathcal{BPP}$. *Furthermore, every problem in* dist$\mathcal{NP}$ *is reducible to some problem in* dist$\mathcal{PC}$*, and every problem in* dist$\mathcal{PC}$ *is* randomly *reducible to some problem in* dist$\mathcal{NP}$.

Proof Sketch: The furthermore part is analogous to the actual contents of the proof that the devision and search versions of the P-vs-NP question are equivalent (see, e.g., [6, Thm. 2.6] and [6, Thm. 2.16]). Indeed the standard reduction of $\mathcal{NP}$ to $\mathcal{PC}$ extends to the current context. Specifically, for any $S \in \mathcal{NP}$, we consider a relation $R \in \mathcal{PC}$ such that $S = \{x : R(x) \neq \emptyset\}$, and note that, for any probability ensemble X, the identity transformation reduces (S, X) to (R, X).

A difficulty arises in the opposite direction. Recall that in the standard reduction of $\mathcal{PC}$ to $\mathcal{NP}$, one reduces the search problem of $R \in \mathcal{PC}$ to deciding membership in $S'_R \stackrel{\text{def}}{=} \{\langle x, y' \rangle : \exists y'' \text{ s.t. } (x, y'y'') \in R\} \in \mathcal{NP}$. The difficulty encountered here is that, on input x, this reduction makes queries of the form $\langle x, y' \rangle$, where y' is a prefix of some string in $R(x)$. These queries may induce a distribution that is not dominated by any simple distribution. Thus, we seek an alternative reduction.

As a warm-up, let us assume for a moment that R has unique solutions; that is, for every x it holds that $|R(x)| \leq 1$. In this case we may easily reduce the search problem of $R \in \mathcal{PC}$ to deciding membership in $S''_R \in \mathcal{NP}$, *where $\langle x, i, \sigma \rangle \in S''_R$ if and only if $R(x)$ contains a string in which the i^{th} bit equals σ*. Specifically, on input x, the reduction issues the queries $\langle x, i, \sigma \rangle$, where $i \in [\ell]$ (with $\ell = \text{poly}(|x|)$) and $\sigma \in \{0, 1\}$, which allows for determining the single string in the set $R(x) \subseteq \{0, 1\}^\ell$ (whenever $|R(x)| = 1$). The point is that this reduction can be used to reduce any $(R, X) \in$ dist$\mathcal{PC}$ (having unique solutions) to $(S''_R, X'') \in$ dist$\mathcal{NP}$, *where X'' equally distributes the probability mass of x* (under X) *to all the tuples $\langle x, i, \sigma \rangle$; that is, for every $i \in [\ell]$ and $\sigma \in \{0, 1\}$, it holds that* $\mathsf{Pr}[X''_{|\langle x,i,\sigma \rangle|} = \langle x, i, \sigma \rangle]$ *equals* $\mathsf{Pr}[X_{|x|} = x]/2\ell$.

Unfortunately, in the general case, R may not have unique solutions. Nevertheless, applying the main idea that underlies the reduction of NP to "unique-NP" (cf. [6, Thm. 6.29]), this difficulty can be overcome. We first note that the

foregoing mapping of instances of the distributional problem $(R, X) \in \text{dist}\mathcal{PC}$ to instances of $(S''_R, X'') \in \text{dist}\mathcal{NP}$ satisfies the efficiency and domination conditions even in the case that R does not have unique solutions. What may possibly fail (in the general case) is the validity condition (i.e., if $|R(x)| > 1$ then we may fail to recover any element of $R(x)$).

Recall that the core of the reduction of NP to unique-NP is a randomized mapping of instances x (of any $R \in \mathcal{PC}$) to triples of the form (x, m, h) such that m is uniformly distributed in $[\ell]$ and h is uniformly distributed in a family of hashing function H_ℓ^m, where $\ell = \text{poly}(|x|)$ and H_ℓ^m is a family of pairwise indepence hashing functions. Furthermore, if $R(x) \neq \emptyset$ then, with probability $\Omega(1/\ell)$ over the choices of $m \in [\ell]$ and $h \in H_\ell^m$, there exists a unique $y \in R(x)$ such that $h(y) = 0^m$. Defining $R'(x, m, h) \stackrel{\text{def}}{=} \{y \in R(x) : h(y) = 0^m\}$, this yields a randomized reduction of the search problem of R to the search problem of R' such that with noticeable probability[22] the reduction maps instances that have solutions to instances having a unique solution. Furthermore, this reduction can be used to reduce any $(R, X) \in \text{dist}\mathcal{PC}$ to $(R', X') \in \text{dist}\mathcal{PC}$, *where X' distributes the probability mass of x* (under X) *to all the triples (x, m, h) such that for every $m \in [\ell]$ and $h \in H_\ell^m$ it holds that* $\mathsf{Pr}[X'_{|(x,m,h)|} = (x, m, h)]$ *equals* $\mathsf{Pr}[X_{|x|} = x]/(\ell \cdot |H_\ell^m|)$. (Note that with a suitable encoding, X' is indeed simple.)

The theorem follows by combining the two aforementioned reductions. That is, we first apply the randomized reduction of (R, X) to (R', X'), and next reduce the resulting instance to an instance of the corresponding decision problem $(S''_{R'}, X'')$, where X'' is obtained by modifying X' (rather than X). The combined randomized mapping satisfies the efficiency and domination conditions, and is valid with noticeable probability. The error probability can be made negligible by straightforward amplification. □

3.2 Simple versus sampleable distributions

Recall that the definition of simple probability ensembles (underlying Definition 2) requires that the accumulating distribution function is polynomial-time computable. Recall that $\mu : \{0,1\}^* \to [0,1]$ is called the accumulating distribution function of $X = \{X_n\}_{n \in \mathbb{N}}$ if for every $n \in \mathbb{N}$ and $x \in \{0,1\}^n$ it holds that $\mu(x) \stackrel{\text{def}}{=} \mathsf{Pr}[X_n \leq x]$, where the inequality refers to the standard lexicographic order of n-bit strings.

As argued in Section 2.1, the requirement that the accumulating distribution function is polynomial-time computable imposes severe restrictions on the set of admissible ensembles. Furthermore, it seems that these simple ensembles are indeed "simple" in some intuitive sense, and that they represent a reasonable (alas disputable) model of distributions that may occur in practice. Still, in light

[22] Recall that the probability of an event is said to be noticeable (in a relevant parameter) if it is greater than the reciprocal of some positive polynomial. In the context of randomized reductions, the relevant parameter is the length of the input to the reduction.

of the fear that this model is too restrictive (and consequently that dist$\mathcal{NP}$-hardness is weak evidence for typical-case hardness), we seek a maximalistic model of distributions that may occur in practice. Such a model is provided by the notion of polynomial-time sampleable ensembles (underlying Definition 12). Our maximality thesis is based on the belief that the real world should be modeled as a *feasible* randomized process (rather than as an arbitrary process). This belief implies that all objects encountered in the world may be viewed as samples generated by a feasible randomized process.

Definition 12 (sampleable ensembles and the class samp$\mathcal{NP}$): *We say that a probability ensemble $X = \{X_n\}_{n\in\mathbb{N}}$ is* (polynomial-time) sampleable *if there exists a probabilistic polynomial-time algorithm A such that for every $x \in \{0,1\}^*$ it holds that $\mathsf{Pr}[A(1^{|x|}) = x] = \mathsf{Pr}[X_{|x|} = x]$. We denote by* samp$\mathcal{NP}$ *the class of distributional problems consisting of decision problems in $\mathcal{NP}$ coupled with sampleable probability ensembles.*

We first note that all simple probability ensembles are indeed sampleable, and thus dist$\mathcal{NP} \subseteq$ samp$\mathcal{NP}$. On the other hand, there exist sampleable probability ensembles that do not seem simple (and so it seems that dist$\mathcal{NP} \subset$ samp$\mathcal{NP}$).

Extending the scope of distributional problems (from dist$\mathcal{NP}$ to samp$\mathcal{NP}$) facilitates the presentation of complete distributional problems. We first note that it is easy to prove that every natural NP-complete problem has a distributional version in samp$\mathcal{NP}$ that is dist$\mathcal{NP}$-hard. Furthermore, it is possible to prove that all natural NP-complete problem have distributional versions that are samp$\mathcal{NP}$-complete. (In both cases, "natural" means that the corresponding Karp-reductions do not shrink the input, which is a weaker condition than the one in Proposition 6.)

Theorem 13 (samp$\mathcal{NP}$-completeness): *Suppose that $S \in \mathcal{NP}$ and that every set in $\mathcal{NP}$ is reducible to S by a Karp-reduction that does not shrink the input. Then, there exists a polynomial-time sampleable ensemble X such that any problem in* samp$\mathcal{NP}$ *is reducible to (S, X)*

The proof of Theorem 13 is based on the observation that *there exists a polynomial-time sampleable ensemble that dominates all polynomial-time sampleable ensembles*. The existence of this ensemble is based on the notion of a universal (sampling) machine.

Theorem 13 establishes a rich theory of samp$\mathcal{NP}$-completeness, but does not relate this theory to the previously presented theory of dist$\mathcal{NP}$-completeness (see Figure 1). This is essentially done in the next theorem, which asserts that the existence of typically hard problems in samp$\mathcal{NP}$ implies their existence in dist$\mathcal{NP}$.

Theorem 14 (samp$\mathcal{NP}$-completeness versus dist$\mathcal{NP}$-completeness): *If* samp$\mathcal{NP}$ *is not contained in* tpc$\mathcal{BPP}$ *then* dist$\mathcal{NP}$ *is not contained in* tpc$\mathcal{BPP}$.

Thus, the two "typical-case complexity" versions of the P-vs-NP Question are equivalent. That is, if some "sampleable distribution" versions of NP are not

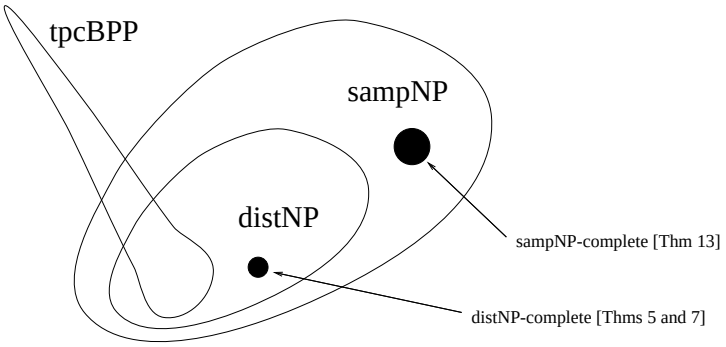

Fig. 1. Two types of average-case completeness

typically feasible then some "simple distribution" versions of NP are not typically feasible. In particular, if samp$\mathcal{NP}$-complete problems are not in tpc$\mathcal{BPP}$ then dist$\mathcal{NP}$-complete problems are not in tpc$\mathcal{BPP}$.

The foregoing assertions would all follow if samp$\mathcal{NP}$ were (randomly) reducible to dist$\mathcal{NP}$ (i.e., if every problem in samp$\mathcal{NP}$ were reducible (under a randomized version of Definition 3) to some problem in dist$\mathcal{NP}$); but, unfortunately, we do not know whether such reductions exist. Yet, underlying the proof of Theorem 14 is a more liberal notion of a reduction among distributional problems.

Proof Sketch: We shall prove that if dist$\mathcal{NP}$ is contained in tpc$\mathcal{BPP}$ then the same holds for samp$\mathcal{NP}$ (i.e., samp$\mathcal{NP}$ is contained in tpc$\mathcal{BPP}$). Relying on Theorem 11 and an analogous result for the sampleable classes, it suffices to show that if dist$\mathcal{PC}$ is contained in tpc$\mathcal{BPPF}$, then the sampleable version of dist$\mathcal{PC}$, denoted samp$\mathcal{PC}$, is contained in tpc$\mathcal{BPPF}$. This will be shown by showing that, under a relaxed notion of a randomized reduction, every problem in samp$\mathcal{PC}$ is reduced to some problem in dist$\mathcal{PC}$. Loosely speaking, this relaxed notion (of a randomized reduction) only requires that the validity and domination conditions (of Definition 3 (when adapted to randomized reductions)) hold with respect to a noticeable fraction of the probability space of the reduction.[23] We start by formulating this notion, when referring to distributional *search* problems.

Definition: A relaxed reduction of the distributional problem (R, X) to the distributional problem (T, Y) is a probabilistic polynomial-time oracle machine M that satisfies the following conditions with respect to a family of sets $\{\Omega_x \subseteq \{0,1\}^{m(|x|)} : x \in \{0,1\}^*\}$, where $m(|x|) = \mathrm{poly}(|x|)$ denotes an upper-bound on the number of the internal coin tosses of M on input x:

[23] We warn that the existence of such a relaxed reduction between two specific distributional problems does not necessarily imply the existence of a corresponding (standard average-case) reduction. Specifically, although standard validity can be guaranteed (for problems in $\mathcal{PC}$) by repeated invocations of the reduction, such a process will *not* redeem the violation of the standard domination condition.

Density (of Ω_x): There exists a noticeable function $\rho : \mathbb{N} \to [0,1]$ (i.e., $\rho(n) > 1/\mathrm{poly}(n)$) such that, for every $x \in \{0,1\}^*$, it holds that $|\Omega_x| \geq \rho(|x|) \cdot 2^{m(|x|)}$.

Validity (with respect to Ω_x): For every $r \in \Omega_x$ the reduction yields a correct answer; that is, $M^T(x,r) \in R(x)$ if $R(x) \neq \emptyset$ and $M^T(x,r) = \perp$ otherwise, where $M^T(x,r)$ denotes the execution of M on input x, internal coins r, and oracle access to T.

Domination (with respect to Ω_x): There exists a positive polynomial p such that, for every $y \in \{0,1\}^*$ and every $n \in \mathbb{N}$, it holds that

$$\mathsf{Pr}[Q'(X_n) \ni y] \leq p(|y|) \cdot \mathsf{Pr}[Y_{|y|} = y], \tag{5}$$

where $Q'(x)$ is a random variable, defined over the set Ω_x, representing the set of queries made by M on input x, coins in Ω_x, and oracle access to T. That is, $Q'(x)$ is defined by uniformly selecting $r \in \Omega_x$ and considering the set of queries made by M on input x, internal coins r, and oracle access to T. (In addition, as in Definition 3, we also require that the reduction does not make too short queries.)

The reader may verify that this relaxed notion of a reduction preserves typical feasibility; that is, for $R \in \mathcal{PC}$, if there exists a relaxed reduction of (R,X) to (T,Y) and (T,Y) is in tpc$\mathcal{BPP}\mathcal{F}$ then (R,X) is in tpc$\mathcal{BPP}\mathcal{F}$. The key observation is that the analysis may discard the case that, on input x, the reduction selects coins not in Ω_x. Indeed, the queries made in that case may be untypical and the answers received may be wrong, but this is immaterial. What matter is that, on input x, with noticeable probability the reduction selects coins in Ω_x, and produces "typical with respect to Y" queries (by virtue of the relaxed domination condition). Such typical queries are answered correctly by the algorithm that typically solves (T,Y), and if x has a solution then these answers yield a correct solution to x (by virtue of the relaxed validity condition). Thus, if x has a solution then with noticeable probability the reduction outputs a correct solution. On the other hand, the reduction never outputs a wrong solution (even when using coins not in Ω_x), because incorrect solutions are detected by relying on $R \in \mathcal{PC}$.

Our goal is presenting, for every $(R,X) \in \mathrm{samp}\mathcal{PC}$, a relaxed reduction of (R,X) to a related problem $(R',X') \in \mathrm{dist}\mathcal{PC}$. (We use the standard notation $X = \{X_n\}_{n\in\mathbb{N}}$ and $X' = \{X'_n\}_{n\in\mathbb{N}}$.)

An oversimplified case: For starters, *suppose that X_n is uniformly distributed on some set $S_n \subseteq \{0,1\}^n$ and that there is a polynomial-time computable and invertible mapping μ of S_n to $\{0,1\}^{\ell(n)}$, where $\ell(n) = \log_2 |S_n|$*. Then, mapping x to $1^{|x|-\ell(|x|)}0\mu(x)$, we obtain a reduction of (R,X) to (R',X'), where X'_{n+1} is uniform over $\{1^{n-\ell(n)}0v : v \in \{0,1\}^{\ell(n)}\}$ and $R'(1^{n-\ell(n)}0v) = R(\mu^{-1}(v))$ (or, equivalently, $R(x) = R'(1^{|x|-\ell(|x|)}0\mu(x))$). Note that X' is a simple ensemble and $R' \in \mathcal{PC}$; hence, $(R',X') \in \mathrm{dist}\mathcal{PC}$. Also note that the foregoing mapping is indeed a valid reduction (i.e., it satisfies the efficiency, validity, and domination conditions). Thus, (R,X) is reduced to a problem in dist$\mathcal{PC}$ (and indeed the relaxation was not used here).

A simple but more instructive case: Next, we drop the assumption that there is a polynomial-time computable and invertible mapping μ of S_n to $\{0,1\}^{\ell(n)}$, but maintain the assumption that X_n is uniform on some set $S_n \subseteq \{0,1\}^n$ and assume that $|S_n| = 2^{\ell(n)}$ is easily computable (from n). In this case, we may map $x \in \{0,1\}^n$ to its image under a suitable randomly chosen hashing function h, which in particular maps n-bit strings to $\ell(n)$-bit strings. That is, we randomly map x to $(h, 1^{n-\ell(n)}0h(x))$, where h is uniformly selected in a set $H_n^{\ell(n)}$ of suitable hash functions (i.e., pairwise independent ones). This calls for redefining R' such that $R'(h, 1^{n-\ell(n)}0v)$ corresponds to the preimages of v under h that are in S_n. Assuming that h is a 1-1 mapping of S_n to $\{0,1\}^{\ell(n)}$, we may define $R'(h, 1^{n-\ell(n)}0v) = R(x)$ such that x is the unique string satisfying $x \in S_n$ and $h(x) = v$, where the condition $x \in S_n$ may be *verified by providing the internal coins of the sampling procedure that generate* x. Denoting the sampling procedure of X by S, and letting $S(1^n, r)$ denote the output of S on input 1^n and internal coins r, we actually redefine R' as

$$R'(h, 1^{n-\ell(n)}0v) = \{\langle r, y\rangle : h(S(1^n, r)) = v \wedge y \in R(S(1^n, r))\}. \tag{6}$$

We note that $\langle r, y\rangle \in R'(h, 1^{|x|-\ell(|x|)}0h(x))$ yields a desired solution $y \in R(x)$ if $S(1^{|x|}, r) = x$, but otherwise "all bets are off" (since y will be a solution for $S(1^{|x|}, r) \neq x$). Now, although typically h will not be a 1-1 mapping of S_n to $\{0,1\}^{\ell(n)}$, it is the case that *for each $x \in S_n$, with constant probability over the choice of h, it holds that $h(x)$ has a unique preimage in S_n under h*. In this case $\langle r, y\rangle \in R'(h, 1^{|x|-\ell(|x|)}0h(x))$ implies $S(1^{|x|}, r) = x$ (which, in turn, implies $y \in R(x)$). We claim that *the randomized mapping of x to $(h, 1^{n-\ell(n)}0h(x))$, where h is uniformly selected in $H_{|x|}^{\ell(|x|)}$, yields a relaxed reduction of (R, X) to (R', X'), where $X'_{n'}$ is uniform over $H_n^{\ell(n)} \times \{1^{n-\ell(n)}0v : v \in \{0,1\}^{\ell(n)}\}$*. Needless to say, the claim refers to the reduction that (on input x, makes the query $(h, 1^{n-\ell(n)}0h(x))$, and) returns y if the oracle answer equals $\langle r, y\rangle$ and $y \in R(x)$.

The claim is proved by considering the set Ω_x of choices of $h \in H_{|x|}^{\ell(|x|)}$ for which $x \in S_n$ is the only preimage of $h(x)$ under h that resides in S_n (i.e., $|\{x' \in S_n : h(x') = h(x)\}| = 1$). In this case (i.e., $h \in \Omega_x$) it holds that $\langle r, y\rangle \in R'(h, 1^{|x|-\ell(|x|)}0h(x))$ implies that $S(1^{|x|}, r) = x$ and $y \in R(x)$, and the (relaxed) validity condition follows. The (relaxed) domination condition follows by noting that $\mathsf{Pr}[X_n = x] \approx 2^{-\ell(|x|)}$, that x is mapped to $(h, 1^{|x|-\ell(|x|)}0h(x))$ with probability $1/|H_{|x|}^{\ell(|x|)}|$, and that x is the only preimage of $(h, 1^{|x|-\ell(|x|)}0h(x))$ under the mapping (among $x' \in S_n$ such that $\Omega_{x'} \ni h$).

Before going any further, let us highlight the importance of hashing X_n to $\ell(n)$-bit strings. On one hand, this mapping is "sufficiently" one-to-one, and thus (with constant probability) the solution provided for the hashed instance (i.e., $h(x)$) yield a solution for the original instance (i.e., x). This guarantees the validity of the reduction. On the other hand, for a typical h, the mapping of X_n to $h(X_n)$ covers the relevant range almost uniformly. This guarantees that the reduction satisfies the domination condition. Note that these two phenomena

impose conflicting requirements that are both met at the correct value of ℓ; that is, the one-to-one condition requires $\ell(n) \geq \log_2 |S_n|$, whereas an almost uniform cover requires $\ell(n) \leq \log_2 |S_n|$. Also note that $\ell(n) = \log_2(1/\mathsf{Pr}[X_n = x])$ for every x in the support of X_n; the latter quantity will be in our focus in the general case.

The general case: Finally, we get rid of the assumption that X_n is *uniformly distributed* over some subset of $\{0,1\}^n$. All that we know is that there exists a probabilistic polynomial-time ("sampling") algorithm S such that $S(1^n)$ is distributed identically to X_n. In this (general) case, we map instances of (R, X) according to their probability mass such that x is mapped to an instance (of R') that consists of $(h, h(x))$ and additional information, where h is a random hash function mapping n-bit long strings to ℓ_x-bit long strings such that

$$\ell_x \stackrel{\text{def}}{=} \lceil \log_2(1/\mathsf{Pr}[X_{|x|}=x]) \rceil. \tag{7}$$

Since (in the general case) there may be more than 2^{ℓ_x} strings in the support of X_n, we need to augment the reduced instance in order to ensure that it is uniquely associated with x. The basic idea is augmenting the mapping of x to $(h, h(x))$ with additional information that restricts X_n to strings that occur with probability at least $2^{-\ell_x}$. Indeed, when X_n is restricted in this way, the value of $h(X_n)$ uniquely determines X_n.

Let $q(n)$ denote the randomness complexity of S and $S(1^n, r)$ denote the output of S on input 1^n and internal coin tosses $r \in \{0,1\}^{q(n)}$. Then, we randomly map x to $(h, h(x), h', v')$, where $h : \{0,1\}^{|x|} \to \{0,1\}^{\ell_x}$ and $h' : \{0,1\}^{q(|x|)} \to \{0,1\}^{q(|x|)-\ell_x}$ are random hash functions and $v' \in \{0,1\}^{q(|x|)-\ell_x}$ is uniformly distributed. The instance (h, v, h', v') of the redefined search problem R' has solutions that consists of pairs $\langle r, y\rangle$ such that $h(S(1^n, r)) = v \wedge h'(r) = v'$ and $y \in R(S(1^n, r))$. As we shall see, this augmentation guarantees that, with constant probability (over the choice of h, h', v'), the solutions to the reduced instance $(h, h(x), h', v')$ correspond to the solutions to the original instance x.

The foregoing description assumes that, on input x, we can efficiently determine ℓ_x, which is an assumption that cannot be justified. Instead, we select ℓ uniformly in $\{0, 1, ..., q(|x|)\}$, and so with noticeable probability we do select the correct value (i.e., $\mathsf{Pr}[\ell = \ell_x] = 1/(q(|x|)+1) = 1/\mathrm{poly}(|x|)$). For clarity, we make n and ℓ explicit in the reduced instance. Thus, we randomly map $x \in \{0,1\}^n$ to $(1^n, 1^\ell, h, h(x), h', v') \in \{0,1\}^{n'}$, where $\ell \in \{0, 1, ..., q(n)\}$, $h \in H^\ell_n$, $h' \in H^{q(n)-\ell}_{q(n)}$, and $v' \in \{0,1\}^{q(n)-\ell}$ are uniformly distributed in the corresponding sets.[24] This mapping will be used to reduce (R, X) to (R', X'), where R' and $X' = \{X'_{n'}\}_{n' \in \mathbb{N}}$ are redefined (yet again). Specifically, we let

$$R'(1^n, 1^\ell, h, v, h', v') = \{\langle r, y\rangle : h(S(1^n, r)) = v \wedge h'(r) = v' \wedge y \in R(S(1^n, r))\} \tag{8}$$

[24] As in other places, a suitable encoding will be used such that the reduction maps strings of the same length to strings of the same length (i.e., n-bit string are mapped to n'-bit strings, for $n' = \mathrm{poly}(n)$). For example, we may encode $\langle 1^n, 1^\ell, h, h(x), h', v'\rangle$ as $1^n 0 1^\ell 0 1^{q(n)-\ell} 0 \langle h\rangle\langle h(x)\rangle\langle h'\rangle\langle v'\rangle$, where each $\langle w\rangle$ denotes an encoding of w by a string of length $(n' - (n + q(n) + 3))/4$.

and $X'_{n'}$ assigns equal probability to each $X_{n',\ell}$ (for $\ell \in \{0,1,...,n\}$), where each $X_{n',\ell}$ is isomorphic to the uniform distribution over $H^{\ell}_{n} \times \{0,1\}^{\ell} \times H^{q(n)-\ell}_{q(n)} \times \{0,1\}^{q(n)-\ell}$. Note that indeed $(R', X') \in \mathrm{dist}\mathcal{PC}$.

The foregoing randomized mapping is analyzed by considering the correct choice for ℓ; that is, on input x, we focus on the choice $\ell = \ell_x$. Under this conditioning (as we shall show), *with constant probability over the choice of h, h' and v', the instance x is the only value in the support of X_n that is mapped to $(1^n, 1^{\ell_x}, h, h(x), h', v')$ and satisfies $\{r : h(S(1^n, r)) = h(x) \wedge h'(r) = v'\} \neq \emptyset$.* It follows that (for such h, h' and v') any solution $\langle r, y\rangle \in R'(1^n, 1^{\ell_x}, h, h(x), h', v')$ satisfies $S(1^n, r) = x$ and thus $y \in R(x)$, which means that the (relaxed) validity condition is satisfied. The (relaxed) domination condition is satisfied too, because (conditioned on $\ell = \ell_x$ and for such h, h', v') the probability that X_n is mapped to $(1^n, 1^{\ell_x}, h, h(x), h', v')$ approximately equals $\mathsf{Pr}[X'_{n',\ell_x} = (1^n, 1^{\ell_x}, h, h(x), h', v')]$.

We now turn to analyze the probability, over the choice of h, h' and v', that the instance x is the only value in the support of X_n that is mapped to $(1^n, 1^{\ell_x}, h, h(x), h', v')$ and satisfies $\{r : h(S(1^n, r)) = h(x) \wedge h'(r) = v'\} \neq \emptyset$. Firstly, we note that $|\{r : S(1^n, r) = x\}| \geq 2^{q(n)-\ell_x}$, and thus, with constant probability over the choice of $h' \in H^{q(n)-\ell_x}_{q(n)}$ and $v' \in \{0,1\}^{q(n)-\ell_x}$, there exists r that satisfies $S(1^n, r) = x$ and $h'(r) = v'$. Furthermore, with constant probability over the choice of $h' \in H^{q(n)-\ell_x}_{q(n)}$ and $v' \in \{0,1\}^{q(n)-\ell_x}$, it also holds that there are at most $O(2^{\ell_x})$ strings r such that $h'(r) = v'$. Fixing such h' and v', we let $S_{h',v'} = \{S(1^n, r) : h'(r) = v'\}$ and we note that, with constant probability over the choice of $h \in H^{\ell_x}_{n}$, it holds that x is the only string in $S_{h',v'}$ that is mapped to $h(x)$ under h. Thus, with constant probability over the choice of h, h' and v', the instance x is the only value in the support of X_n that is mapped to $(1^n, 1^{\ell_x}, h, h(x), h', v')$ and satisfies $\{r : h(S(1^n, r)) = h(x) \wedge h'(r) = v'\} \neq \emptyset$. The theorem follows. □

Reflection: Theorem 14 implies that if samp$\mathcal{NP}$ is not contained in tpc$\mathcal{BPP}$ then every dist$\mathcal{NP}$-complete problem is not in tpc$\mathcal{BPP}$. This means that the hardness of some distributional problems that refer to sampleable distributions implies the hardness of some distributional problems that refer to simple distributions. Furthermore, by Proposition 9, this implies the hardness of distributional problems that refer to the uniform distribution. Thus, hardness with respect to some distribution in an utmost wide class (which arguably captures all distributions that may occur in practice) implies hardness with respect to a single simple distribution (which arguably is the simplest one).

Relation to one-way functions. We note that the existence of one-way functions (see, e.g., [5, Chap. 2]) implies the existence of problems in samp$\mathcal{PC}$ that are not in tpc$\mathcal{BPPF}$ (which in turn implies the existence of such problems in dist$\mathcal{PC}$). Specifically, for a length-preserving one-way function f, consider the distribu-

tional search problem $(R_f, \{f(U_n)\}_{n\in\mathbb{N}})$, where $R_f = \{(f(r), r) : r \in \{0,1\}^*\}$.[25] On the other hand, it is not known whether the existence of a problem in samp$\mathcal{PC}\setminus$tpc$\mathcal{BPPF}$ implies the existence of one-way functions. In particular, the existence of a problem (R, X) in samp$\mathcal{PC} \setminus$ tpc$\mathcal{BPPF}$ represents the feasibility of generating hard instances for the search problem R, whereas the existence of one-way function represents the feasibility of generating instance-solution pairs such that the instances are hard to solve. Indeed, the gap refers to whether or not *hard instances can be efficiently generated together with corresponding solutions*. Our world view is thus depicted in Figure 2, where lower levels indicate seemingly weaker assumptions.

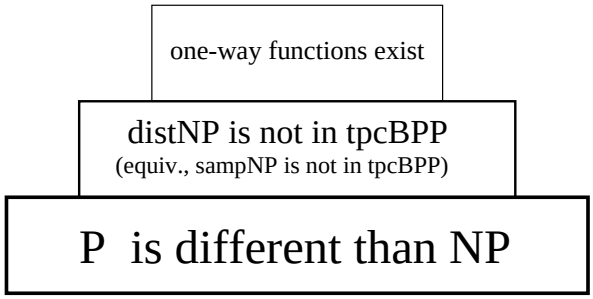

Fig. 2. Worst-case vs average-case assumptions

Bibliographic Notes

The theory of average-case complexity was initiated by Levin [10], who in particular proved Theorem 5. In light of the laconic nature of the original text [10], we refer the interested reader to a survey [4], which provides a more detailed exposition of the definitions suggested by Levin as well as a discussion of the considerations underlying these suggestions.

As noted in Section 2.1, the current text uses a variant of the original definitions. In particular, our definition of "typical-case feasibility" differs from the original definition of "average-case feasibility" in totally discarding exceptional instances and in even allowing the algorithm to fail on them (and not merely run for an excessive amount of time). The alternative definition was suggested by several researchers, and appears as a special case of the general treatment provided in [2].

Turning to Section 2.2, we note that while the existence of dist$\mathcal{NP}$-complete problems (cf. Theorem 5) was established in Levin's original paper [10], the exis-

[25] Note that the distribution $f(U_n)$ is uniform in the special case that f is a permutation over $\{0,1\}^n$.

tence of dist$\mathcal{NP}$-complete versions of all natural NP-complete decision problems (cf. Theorem 7) was established more than two decades later in [11].

Section 3 is based on [1, 8]. Specifically, Theorem 11 (or rather the reduction of search to decision) is due to [1] and so is the introduction of the class samp$\mathcal{NP}$. A version of Theorem 14 was proven in [8], and our proof follows their ideas, which in turn are closely related to the ideas underlying the construction of pseudorandom generators based on any one-way function [7].

Recall that we know of the existence of problems in dist$\mathcal{NP}$ that are hard provided samp$\mathcal{NP}$ contains hard problems. However, these distributional problems do not seem very natural (i.e., they either refer to somewhat generic decision problems such as $S_{\mathtt{u}}$ or to somewhat contrived probability ensembles (cf. Theorem 7)). The presentation of dist$\mathcal{NP}$-complete problems that combine a more natural decision problem (like `SAT` or `Clique`) with a more natural probability ensemble is an open problem.

A natural question at this point is what have we gained by relaxing the requirements. In the context of average-case complexity, the negative side seems more prevailing (at least in the sense of being more systematic). In particular, assuming the existence of one-way functions, every natural NP-complete problem has a distributional version that is (typical-case) hard, where this version refers to a sampleable ensemble (and, in fact, even to a simple ensemble). Furthermore, in this case, some problems in NP have hard distributional versions that refer to the uniform distribution.

References

1. S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the Theory of Average Case Complexity. *Journal of Computer and System Science*, Vol. 44 (2), pages 193–219, 1992.
2. A. Bogdanov and L. Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, Vol. 2 (1), 2006.
3. S.A. Cook. The Complexity of Theorem Proving Procedures. In *3rd ACM Symposium on the Theory of Computing*, pages 151–158, 1971.
4. O. Goldreich. Notes on Levin's Theory of Average-Case Complexity. This volume. See also *ECCC*, TR97-058, Dec. 1997.
5. O. Goldreich. *Foundation of Cryptography: Basic Tools*. Cambridge University Press, 2001.
6. O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.
7. J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby. A Pseudorandom Generator from any One-way Function. *SICOMP*, Volume 28, Number 4, pages 1364–1396, 1999. Combines papers of Impagliazzo et al. (*21st STOC*, 1989) and Håstad (*22nd STOC*, 1990).
8. R. Impagliazzo and L.A. Levin. No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random. In *31st IEEE Symposium on Foundations of Computer Science*, pages 812–821, 1990.
9. R.M. Karp. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, R.E. Miller and J.W. Thatcher (eds.), Plenum Press, pages 85–103, 1972.

10. L.A. Levin. Average Case Complete Problems. *SIAM Journal on Computing*, Vol. 15, pages 285–286, 1986.
11. N. Livne. All Natural NPC Problems Have Average-Case Complete Versions. *ECCC*, TR06-122, 2006.