Abstract

Informally, Average case Complexity theory considers the average complexity with respect to a probability distribution over the set of inputs, instead of the complexity on the worst case input, as a criterion for the hardness of a problem.

With regard to this theory, we explore the notion of a natural distribution. In particular, we consider different bounds on the min-entropy of such distributions over input strings of a fixed length, and generalize the incompleteness theorem (under deterministic reductions) for distributional problems using techniques based on the work by Gurevich[1991]. We also present some weak restrictions on randomized reductions under which problems with a uniform distribution are average case complete. These restrictions have generalizations to the case when the distribution is not uniform but still has an upper bound on its min entropy, over strings of a given length.

We also describe our attempts to study the effect of unnatural encodings on average case completeness.
1 Introduction

1.1 Background

Computational Complexity Theory is, roughly, the study of hardness of problems. It considers how algorithms for solving a problem scale as the complexity of an instance of the problem increases. For instance, if we consider the simple problem of multiplying two $n \times n$ matrices, the trivial algorithm takes of the order of $n^3$ (represented as $O(n^3)$) steps, even in the worst case. Thus we say that the problem is solvable in polynomial time. In general, it considers the worst case hardness of a problem. Thus the hardness of a problem would be determined by the worst case performance of the best possible algorithm for solving the problem.

Average-case Complexity Theory deals with the practically important problem of characterizing problems on the basis of their average case, as opposed to the worst case, hardness. In practice, if the “hard” instances of a problem seldom occur, then the worst case hardness of the problem would not be of much significance. For instance, as shown by Wilf [9], the graph coloring problem has an algorithm whose average running time is a constant, assuming a uniform distribution over the inputs, where as in the worst case it can take time exponential in the size of the input.

In general, the possible instances of a problem that can occur in practice would be distributed with respect to a probability distribution. With respect to this distribution, an algorithm may perform significantly better on average than on a worst case.

Thus we are naturally led to the following questions:

- What distributions do we consider?
- What is the criterion for hardness?

The theory of average case-complexity originated from the works of Levin[5]. He described the class of distributions which ought to be considered and developed a criterion of hardness in analogy with the notion of $\mathcal{NP}$ completeness. Thus, the answer to the above questions would require a notion of efficient reductions between problems, analogous to the notion of polynomial time reductions in the theory of $\mathcal{NP}$ completeness. Also, in order to justify the usefulness of the theory, one would also have to use distributions that are natural, that is, they are not very different from the ones that actually
occur in practice. We try to present a class of distributions which seem to be natural for most natural problems, in section 3.

Once these basic definitions and ideas are fixed, the next question is to see if $\mathcal{NP}$ complete problems themselves are average case hard under plausible distributions. Livne[6] has given sufficient conditions on an $\mathcal{NP}$ complete problem under which it would have an average case complete version. He also showed that these conditions hold, in particular, for the 21 $\mathcal{NP}$ complete problems listed by Karp[4]. However, the average case complete versions obtained under these conditions depend significantly on the encoding used for the problem. This is because the conditions themselves and the construction of the appropriate underlying probability distribution, are based on the encoding used.

The importance of encodings above led us to consider unconventional encodings like the Short Circuit Representations proposed by Wigderson et al[1]. They showed that certain graphs can be encoded as circuits in which the number of gates is polylogarithmic in the number of vertices of the graph. Under such encodings, many “trivial” problems like detecting TRIANGLES in graphs become $\mathcal{NP}$ complete. Another question that one may ask is whether Livne’s criteria apply to these problems. We address this question in section 6.

Another issue of importance in this connection is the power of randomised reductions. It has been proved (under a plausible assumption)[2] that under sufficiently random distributions, a problem cannot be average case complete under deterministic reductions. We tried to improve the bound on the randomness on this of the distributions allowed by the hypotheses of the above theorem, in section 4. We also attempt to give certain restrictions on the reductions between average case complete problems, in section 5.

The rest of this report is organised as follows: we first present some important definitions, and then present a description of our work on the above problems.

2 Definitions and Preliminaries

We begin by defining the notion of distributional problems and the notions of average-case preserving reductions [7].
A problem as described below is technically a decision problem, that is given an underlying alphabet \( \Sigma \), a set of strings \( L \subset \Sigma^* \) over this alphabet, and an input \( x \), the problem is to decide whether \( x \in L \).

**Definition 2.1. (distributional problem)** A distributional problem is a problem \( A \) along with a probability distribution \( \mu \) over its instances. We represent this distributional problem as an ordered pair \( (A, \mu) \).

**Definition 2.2. (average-case preserving reductions)** A function \( f \) is an average-case preserving reduction from a distributional problem \( (A, \mu_A) \) to the distributional problem \( (B, \mu_B) \) if the following conditions are satisfied:

- \( f \) is a polynomial time computable reduction.
- \[
\mu_B(y) \geq \frac{1}{q(y)} \sum_{x \in f^{-1}(y)} \mu_A(x)
\]

for some polynomial \( q \).

These reductions are referred to as AP-reductions\(^8\).

The second condition above ensures that if a distributional problem \( (A, \mu) \) is efficiently solvable on average, and the distributional problem \( (B, \tau) \) reduces to it via an average-case preserving reduction, then \( (B, \tau) \) itself is efficiently solvable on average. The notion of average case efficiency is analogous to the idea of a polynomial time algorithm being considered efficient. Thus we define the class of which functions are polynomial on average with respect to a distribution \( \mu \), and define those algorithms to be efficient on average which run in time which is polynomial on average in the size of the input.

**Definition 2.3. (polynomial on average functions)** A function \( f : \Sigma^* \rightarrow \mathbb{R} \) is called polynomial on \( \mu \) average for a probability distribution \( \mu \) if the following infinite series converges for some \( \lambda > 0 \):

\[
\sum_{x \in \Sigma^*} \frac{\mu(x) f(x)^\lambda}{|x|}
\]

We now present the definitions of the distributional analogues of \( \mathcal{NP} \) and \( \mathcal{NP} \) complete problems. These definitions require some restrictions to be put on the probability distributions that can be used. For example, if a probability distribution gives a large weight to an hard instance of an \( NP \)
complete problem, then we do not make sufficient gains by considering the average case complexity of the problem under this distribution. Also such distributions are not likely to arise in practice, or to be natural to a problem. Thus the definitions below use the class of \( \mathcal{P} \)-computable distributions as defined by Levin. These are distributions over \( \Sigma^* \) whose cumulative probability for an string \( x \in \Sigma^* \) can be computed in time polynomial in \(|x|\) and the precision required in the computation.

**Definition 2.4.** (\( \text{dist} \mathcal{NP} \)) The class of distributional problems \((A, \mu)\) where \( A \in \mathcal{NP} \) and \( \mu \) is a \( \mathcal{P} \)-time computable distribution is called \( \text{dist} \mathcal{NP} \).

**Definition 2.5.** (\( \text{dist} \mathcal{NP} \) complete problem) A \( \text{dist} \mathcal{NP} \) complete problem is a distributional problem \((A, \mu) \in \text{dist} \mathcal{NP}\), such that for every problem \((B, \eta) \in \text{dist} \mathcal{NP}\), there is an \( \mathcal{AP} \) - reduction \( f \) from \((B, \eta)\) to \((A, \mu)\).

Our definition of average case preserving reductions required them to be \( \mathcal{P} \)-time computable, following [6]. However, this condition is sometimes relaxed and they are required to be just computable in time which is polynomial on \( \mu \) average (e.g. as in [7]), where \( \mu \) is the distribution underlying the original distributional problem. We use the relaxed version of the definition in section 4.

We present below the definitions of some terms from probability theory which would be needed in sections 3, 5 and 4. The following definition follows [8]:

**Definition 2.6.** (uniform distribution) A probability distribution \( \mu \) over the set of binary strings is said to be uniform if it satisfies

\[
\mu(x) = \rho(|x|)2^{-|x|},
\]

where \( \rho \) satisfies the following condition asymptotically for some polynomial \( p \),

\[
\rho(|x|) > \frac{1}{p(|x|)}.
\]

**Definition 2.7.** (min-entropy) The min-entropy of a probability distribution \( \mu \) over a set \(|S|\) is defined as

\[
\text{min-entropy}(\mu) = -\log \left( \max_{x \in S} (\mu(x)) \right)
\]

To define the min-entropy of a distribution over \( \Sigma^* \), we would define it separately over strings of different lengths, possibly as a function of this length.
3 Natural Distributions

As stated above, one of the important considerations about average case complexity theory is to use distributions that satisfy the following:

- The distributions should be natural to the problem at hand.
- The distribution should not assign very large weights to particular instance, as this might cause the average case complexity to become artificially close to the worst, or the best case complexity.

This leads to the following question: which distributions do we consider natural?

To try to answer such a question, we try to see which distributions appear to be a natural for some common problems.

Let us consider the following problem, called CLIQUE, as an example.

Definition 3.1. CLIQUE. Given an undirected graph $G = (V, E)$, and a natural number $k$, is there a induced complete subgraph of $G$ of size $k$?

As the definition of the distribution depends upon the encoding, let us fix an encoding where the $G$ is described by a lower triangular adjacency matrix, represented in row-major form as a sequence of $|V| \times (|V| - 1)/2$ bits, and $k$ is represented as a sequence of $\lg |V|$ bits.

For this problem, probably the most natural distribution appears to be this:

Description 3.1. Having chosen $k \in \mathbb{N}$, we take any two vertices of the graph $G$ with $|V|$ vertices, and with probability $1/2$ put an edge between them. Also, we can choose $k$ with respect to any standard probability distribution on $\mathbb{N}$.

This probability distribution turns out to be a uniform distribution over the set of strings which are instances of CLIQUE.

In general, for most common problems, the most natural distributions appear to be close to the uniform distribution, or appear as a combination of uniform distributions over subsets of the complete set of the instances of the problem. The following well known result then indicates that one can thus consider characterizing probability distribution based on their min-entropy.
Fact 3.1. Any probability distribution of min-entropy $k$ over a set $S$ can be expressed as a convex combination of uniform probability distributions over subsets of $S$ of size $2^k$.

4 Incompleteness Results

In [2], Gurevich presented the following incompleteness result about dist $\mathcal{NP}$ problems with sufficiently high min-entropy:

Theorem 4.1. If $(D, \tau)$ is complete for dist $\mathcal{NP}$ under AP reductions, where $\tau$ satisfies

$$\tau(x) \leq 2^{-|x|^\epsilon} \text{ where } \epsilon > 0 \text{ is a constant.}$$

then $\text{NEXP} = \text{EXP}$

We interpreted the incompleteness theorem 4.1 in the following way: we consider the distribution on inputs of a given length to have a min-entropy depending only upon the length of the string. Then the incompleteness theorem says that if the min-entropy is bounded from below by a function of the form $p(x) = x^\epsilon, 0 < \epsilon \leq 1$, then completeness under such a distribution would imply a complexity collapse.

We ask if we can further lower the min-entropy and yet ask for such a result to hold. It turns out that the incompleteness theorem can be extended to a case where the min-entropy is described by even more slowly-growing functions of the length. The following theorem holds:

Theorem 4.2. If $(D, \tau)$ is complete for dist $\mathcal{NP}$ under AP reductions, where $\tau$ satisfies

$$\tau(x) \leq 2^{-|x|^{\frac{1}{k(|x|)}}} \text{ where } k(|x|) \in O(\log \log |x|),$$

then

$$\text{NTIME} \left(2^{2^{\Omega(\text{poly}(|x|))}}\right) = \text{DTIME} \left(2^{2^{\Omega(\text{poly}(|x|))}}\right).$$

Our proof of the above theorem is based on techniques used for the proof of the original incompleteness theorem in [2], however a small trick is needed in the latter steps to make the passage to “sub-polynomial” min-entropy.

The interpretation in terms of the min-entropy also shows that if we were to consider the distribution as a convex combination of distributions over subsets of the set of inputs, then even with the size of such subsets made
smaller to the extent of \(2^{-\frac{1}{n \log \log n}}\) (over strings of length \(n\)), the corresponding distributional problem turns out to be average case incomplete (under a plausible assumption).

4.1 Proof of Theorem 4.2

As stated earlier, this proof is quite similar (in terms of the techniques used) to the proof of the original incompleteness theorem given by Gurevich.

We first present the following lemma:

**Lemma 4.1.** Let \(p\) be a polynomial, such that \(t : \Sigma^* \mapsto \Sigma^*\) is a one-one map satisfying:

\[
|t(x)| \leq 2^{2p(|x|)}.
\]

Let \(\mu\) be the probability distribution defined over the set \(\Sigma^*\) as follows:

\[
\mu(y) = \begin{cases} 
0, & \text{if there is no } x \text{ such that } t(x) = y \\
\frac{g^{-|x|}}{|x||x|+1}, & \text{if } t(x) = y
\end{cases}
\]

Then, if a function \(g : \Sigma^* \mapsto \mathbb{R}\) is polynomial on \(\mu\) average, then,

\[
g(t(x)) = O\left(2^{2p(|x|)}\right).
\]

**Proof.** As \(g\) is polynomial on \(\mu\) average, the following series converges for some \(\lambda > 0\):

\[
\sum_{x \in \Sigma^*} \mu(t(x)) \frac{(g(t(x)))^\lambda}{|t(x)|}
\]

The convergence of this series implies in particular that

\[
g(t(x)) = O\left(|t(x)|2^{2p(|x|)}(|x|+1)^{\frac{1}{\lambda}}\right).
\]

Using the condition on \(t\), we get:

\[
g(t(x)) = O\left(2^{2p(|x|)+|x|(|x|+1)^{\frac{1}{\lambda}}}\right).
\]

which gives us the result in the lemma.

\(\square\)

Using the above lemma, we now proceed to prove Theorem 4.2.
Proof. Let \( D \) be a problem in the class \( NTIME\left(2^{2^{O(p(|x|))}}\right) \), for some polynomial \( p \). Suppose \( x \) is any instance of the decision problem corresponding to \( D \). We can pad all possible instances of this problem so as to get a problem in \( \mathcal{NP} \). The padding function \( t : \Sigma^* \mapsto \Sigma^* \) is defined as follows:

\[
t(x) = 1^{(2^{2^{O(p(|x|))}})-|x|-1}.x
\]

Here the operator \( . \) denotes string concatenation. Clearly, \( t \) can be inverted efficiently.\( ^{1} \) Now we can define the problem \( E \) as follows: given a string \( y \),

\[
y \in E \iff t^{-1}(y) \in D
\]

Now, \( t \) is \( \mathcal{P} \)-time invertible, and

\[
|t(x)| = 2^{2^{O(|x|)}}.
\]

Also \( D \in NTIME\left(2^{2^{O(p(|x|))}}\right) \). So we have the following Algorithm for deciding membership in \( E \), given an input \( y \):

1. Find \( x \) such that \( t(x) = y \)

2. Simulate the \( NTIME\left(2^{2^{O(p(|x|))}}\right) \) Turing machine for \( D \) on \( x \).

From the above it is clear that \( E \in \mathcal{NP} \), as the running time of the non-deterministic Turing machine deciding \( x \in D \) is clearly polynomially bounded in \( |y| \), by the definition of \( x \). We now need a \( \mathcal{P} \)-time computable distribution to get a dist \( \mathcal{NP} \) version of \( E \).

Define the probability distribution \( \mu \) as:

\[
\mu(y) = \begin{cases} 
0, & \text{if there is no } x \text{ such that } t(x) = y \\
2^{2^{O(|x|)}}^{-|x|(|x|+1)}, & \text{if } t(x) = y
\end{cases}
\]

This distribution is clearly \( \mathcal{P} \)-time computable. This strange definition is used because using the domination condition for an \( \mathcal{AP} \) preserving reduction, we would be able to use the form of this reduction to show the required relation between \( NTIME\left(2^{2^{O(p(|x|))}}\right) \) and \( DTIME\left(2^{2^{O(p(|x|))}}\right) \).

Now suppose that \((T, \tau)\) is a dist \( NP \) complete problem with an underlying distribution \( \tau \) as in the hypothesis of Theorem 4.2. Then there is an \( \mathcal{AP} \)

\(^{1}\)It can clearly be inverted in time linear in the input length, by simply scanning through \( t(x) \) once.
time average case preserving reduction \( f \) from \( E \) to \( T \).

We need to show only that \(| f ( t ( x ) ) |\) is bounded by an exponential function of \(| x |\), since if this is the case, then given \( x \), we can compute \( f ( x ) \) in time which is at most exponential in \(| x |\), and then use the fact that \( E \in \mathcal{NP} \subseteq \mathcal{EXP} \), to decide the membership of \(| x |\) in \( D \) in time which is at most doubly-exponential in \(| x |\).

Now it follows from the completeness of \(( T, \tau )\) that

\[
\tau ( f ( t ( x ) ) ) \geq \frac{1}{h(|f(t(x))|)} \mu ( t ( x ) ) , \text{for a polynomial } h. \tag{1}
\]

\[
= \frac{1}{h(|f(t(x))|)} \frac{1}{|x|(|x| + 1)} 2^{-|x|} \tag{2}
\]

\[
\geq \frac{1}{h(|f(t(x))|)} 2^{-3|x|}. \tag{3}
\]

where we use definition 2.2 in the first step.

Now, since \( f \) is an \( \mathcal{AP} \) time reduction, it must be polynomial on \( \mu \) average. Also the padding function \( t \) satisfies the conditions of the lemma 4.1. Thus we have

\[
| f ( t ( x ) ) | = O \left( 2^{O(\text{poly}(|x|))} \right)
\]

Thus for every polynomial \( h \) there is a polynomial \( \alpha \) satisfying

\[
h (|f(t(x))|) \leq 2^{2^\alpha(|x|)} \tag{4}
\]

Now using equations 3 and 4 we get

\[
\tau ( f ( t ( x ) ) ) \geq 2^{-2^{\alpha(|x|)-3|x|}} \tag{5}
\]

\[
\geq 2^{-2^{\beta(|x|)}}, \text{ for a polynomial } \beta. \tag{6}
\]

Thus it follows that:

\[
| f ( t ( x ) ) | \leq \frac{1}{\tau ( f ( t ( x ) ) )} \leq - \log \tau ( f ( t ( x ) ) ) \leq 2^{\beta(|x|)}
\]

where \( k \) is as in the hypothesis of the theorem.

But as noted above, the lemma 4.1 implies that

\[
\log \log | f ( t ( x ) ) | = O ( \text{poly} (|x|))
\]

So,

\[
| f ( t ( x ) ) | \leq 2^{O(\text{poly}(|x|))}.
\]

\( \square \)
5 Conditions on the reductions

As discussed in the previous section, (under very plausible assumptions) there
cannot be a dist NP complete problem under a uniform distribution, under
deterministic reductions, although as described, for example, in \[3\], such
problems can be average-case complete under randomised reductions. We
attempt to show certain bounds on these reductions. In particular, we show
that such reductions between distN\(P\) complete problems can at most involve
an additive logarithmic blowup, that is if \(f\) is the reduction, then we would
have \(f(|x|) - |x| = O(\log |x|)\). The details follow.

In the following theorems, we assume that \((T,\tau)\) is a distN\(P\) complete
problem, with \(\tau\) being a uniform distribution. Clearly, the reductions under
which \((T,\tau)\) is distN\(P\) complete may possibly be randomized.

**Theorem 5.1.** Let \(\mu\) be a uniform distribution such that \((A,\mu)\) is a dist-N\(P\)
complete problem. Then every one-one AP reduction \(f\) from \(T\) to \(A\) satisfies
\(|f(x)| - |x| = O(\log |x|)\).

**Proof.** Consider the injective AP-reduction \(f\) from \(T\) to \(A\). So we must
have:
\[
\mu(f(x)) \geq \frac{\tau(x)}{q(|x|)},
\]
for some polynomial \(q\). Since \(\mu, \tau\) are both uniform distributions, we have,
\[
\mu(y) = \rho_1(|y|)2^{-|y|}, \tau(y) = \rho_2(|y|)2^{-|y|}
\]
where there is a polynomial \(p\) satisfying
\[
1 \geq \rho_i(|y|) > \frac{1}{p(|y|)}, \text{ for } i = 1, 2
\]
for all but finitely many \(y\). Now using the equations 7, 8 and 9, we get
\[
2^{|f(x)| - |x|} \leq \rho_1(|x|)q(|x|)p(|f(x)|)
\]
\[
\leq m(|x|), \text{ for all but finitely many values of } x.
\]
since as \(f\) is poly-time computable, \(p(f(x))\) is asymptotically bounded by a
polynomial in \(|x|\). This shows that \(|f(x)| - |x| \in O(\log |x|)\). \(\square\)

In fact, as is already known, the above condition on \(f\) is sufficient in
the sense that if a given reduction to a N\(P\) complete problem satisfies this
“length-preserving” criterion, then it can be used to show that the problem is
dist N\(P\) complete under a uniform distribution. The argument is essentially
the previous argument in reverse.
Theorem 5.2. Let $f$ be a one-one reduction from the $NP$ complete problems $T$ to $A$. If $f$ satisfies
\[ |f(x)| - |x| \in O(\log x), \]
then $A$ is dist-$NP$ complete under any uniform distribution $\mu$.

Proof. Let us consider the distributional problem $(A, \mu)$. Now, we consider the ratio
\[ \frac{\tau(x)}{\mu(f(x))}. \]  
Using the notation from the proof of the previous theorem, we have
\[ \frac{\tau(x)}{\mu(f(x))} = \frac{2^{f(x)|x|} \rho_2(|x|)}{\rho_1(f(|x|))} \leq 2^{f(x)|x|} m(x) \text{ for some polynomial } m \]
\[ \leq l(|x|) \text{ for a polynomial } l \text{ by the condition on } f. \]
But this is exactly the condition for $f$ to be an AP-reduction from $(T, \tau)$ to $(A, \mu)$. As $(T, \tau)$ is a dist-$NP$ complete problem, it follows that $(A, \mu)$ is also a dist-$NP$ complete problem under a uniform distribution.

However, we should note that the last result has the inherent limitation that it holds true only for one-one reductions, although Theorem 5.1 is easily seen to hold also for the case when $f$ is not one-one.

We should also note that in theorem 5.1, instead of assuming a uniform distribution, if put a weaker restriction on the min-entropy of $\mu$, we can still get weaker bounds on the blow-up involved in the reduction. Specifically, if we have
\[ \text{min-entropy } (\mu) \geq k(|x|) \]
then we get the condition
\[ k(|f(x)|) - |x| = O(\log |x|) \]

6 Succinct Representation of Graphs

We now turn to a different direction from that of the last few sections, and use Livne’s sufficient criteria for dist-$NP$ completeness on class of graph problems with succinctly encoded graphs. These encodings, are in the form

\[ \text{As described earlier, the min-entropy would have to be defined separately over strings of different lengths.} \]
of Boolean circuits, were proposed by Galperin and Wigderson[1], who also showed that under such encodings, many trivial problems become \( \mathcal{NP} \) complete. We show that Livne’s criteria of paddability apply to the encoding in such a manner that the underlying circuit essentially remains the same, and hence all such problems would have dist \( \mathcal{NP} \) complete versions too.

We describe the encoding below, followed by an example of how Livne’s criteria apply to the problem. The Circuit Representation for a graph \( G(V,E) \) is defined in terms of the Boolean circuit \( C_G \), which is defined as follows:

**Definition 6.1.** If \( G(V,E) \) is a graph such that \( V = \{v_i|0 \leq i < m\}, |V| = m \) then a circuit representation of \( G \) is defined as the boolean circuit with the inputs \( \bar{i} \) and \( \bar{j} \) such that

\[
C(\bar{i}, \bar{j}) = \begin{cases} 
0, & \text{if } (v_i, v_j) \notin E, \\
1, & \text{if } (v_i, v_j) \in E, \\
\text{Don't care,} & \text{if } v_i \notin V \text{ or } v_j \notin V.
\end{cases}
\]

The above description is said to be a Succinct Circuit Representation (SCR) if the number of gates in \( C \) is \( O(n^{O(1)}) \) where \( n \) is the smallest possible number of bits required to encode \( m - 1 \).

For graphs which have a Succinct circuit representation, certain problems which are easy with the adjacency list representation become hard. As noted by Galperin and Wigderson[1], even problems like detecting a triangle in a graph become \( \mathcal{NP} \) complete. It is interesting to see if such problems satisfy Livne’s[6] conditions for having dist \( \mathcal{NP} \) complete versions.

As an example, we consider the problem of detecting a TRIANGLE in a graph with Succinct Circuit Representation. As shown in [1], this problem is \( \mathcal{NP} \) complete. We show below that it satisfies both of Livne’s conditions. We first describe the encoding for the circuit and Livne’s criteria formally.

### 6.1 Encoding for the SCR

We represent the circuit as a directed graph in its adjacency matrix representation, so that the nodes represent the gates and the edges represent the connections between them. The edges are directed from the output to the input. The output also is represented as a node with no incoming edges and the \( 2n \) inputs are represented as nodes with no out-going edges (Here \( n \) is as defined in Definition [6.1]). The resulting graph is acyclic, as the circuit is purely combinatorial.
Thus a circuit with $k$ gates can be represented by a binary string of size $O(poly(k))$. The evaluation of such a circuit can be done through a depth first search starting at the output node, after having assigned values to the input nodes. This would work because the graph we are considering is acyclic. Thus, the graph would admit a topological sort, and we evaluate the nodes in the reverse order of the topological sort.

**Description 6.1. Encoding for the SCR.** We can thus represent an instance of the problem, where the circuit has $k$ nodes, by a binary string as follows:

1. The first $\lceil \lg k \rceil$ bits represent the index of the output node.
2. The next $k$ bits represent which of the nodes are input nodes. The representation consists of a bit vector of $k$ bits, with the bits corresponding to the input nodes set as 1 and the others as 0.
3. The final $k^2$ bits represent the adjacency matrix of the directed graph.

Hence, the total length of the string is $k^2 + k + \lceil \lg k \rceil$. Here we note that we do not need to encode $k$ as a part of the input instance, as this can itself be determined in time polynomial in the input size by carrying out a binary or even linear search on integers in the interval $[1, k^2 + k + \lceil \lg k \rceil]$.

### 6.2 Livne’s sufficient criteria for dist $NP$ completeness

Here we describe the sufficient criteria given by Livne in [6]. We present Livne’s definitions of regular paddability and monotonous paddability.

**Definition 6.2. (Regular paddability)** The decision problem for $L \subset \Sigma^*$ is said to be regular-paddable if there is a monotonically increasing $q : \mathbb{N} \mapsto \mathbb{N}$ and a polynomial time computable function:

$$S : 1^* \times \Sigma^* \mapsto \Sigma^*$$

such that $S(1^n, x) \in L$ if and only if $x \in L$, for every $n$ and for every $x$,

$$|S(1^m, x)| = q(m), \text{for } m \geq |x|.$$ 

**Definition 6.3. (Monotonous paddability)** The decision problem for $L \subset \Sigma^*$ is said to be monotonous -paddable if there exists a polynomial time computable function

$$M : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$$

such that
• For all, \((p, x) \in \Sigma^* \times \Sigma^*\), \(M(p, x) \in L\) if and only if \(x \in L\).

• \(p < q\), \(|p| = |q|\) and \(|x| = |y|\) \(\Rightarrow M(p, x) < M(q, y)\).

• If \(|x| = |y|\) and \(|p| = |q|\) then \(|M(p, x)| = M(q, y)\) and if \(|x| < |y|\) and \(|p| \leq |q|\) then \(M(p, x) < |M(q, y)|\).

Livne’s criteria is the following:

**Theorem 6.1. (Livne)** If \(N\) is a regular paddable and monotonous paddable \(\mathcal{NP}\) complete problem, then there exists a polynomial time computable distribution \(\mu\) such that \((N, \mu)\) is \(\text{dist} \mathcal{NP}\) complete.

The limitation of these criteria is that the form of the underlying probability distribution is not specified, and in general, the probability distribution obtained would not be a distribution natural to the problem.

### 6.3 Dist \(\mathcal{NP}\) completeness under Succinct Circuit Encodings

We now have the following two theorems.

**Theorem 6.2.** The problem TRIANGLE as described above is regular paddable.

**Proof.** We have to show the existence of a polynomial time computable function:

\[
S : 1^* \times \Sigma^* \mapsto \Sigma^*
\]

such that \(S(1^n, x)\) is the SCR of a graph which has a triangle if and only if \(x\) is such a circuit, and that we can find a monotonic \(q : \mathbb{N} \mapsto \mathbb{N}\), such that:

\[
|S(1^m, x)| = q(m), \text{ for } m \geq |x|.
\]

To achieve this we take the following well known map: we just add unconnected nodes to the circuit, so that the new number of nodes is \(m\). The resulting graph would then be represented by a string of length \(m^2 + m + \lceil \log m \rceil\). Also, as the nodes are unconnected, they do not affect the output of the circuit. Thus all the above conditions are satisfied.

We now need to prove monotonous paddability

**Theorem 6.3.** TRIANGLE has a monotonous padding function.
Proof. We need to show the existence of a polynomial time computable function:

\[ M : \Sigma^* \times \Sigma^* \mapsto \Sigma^* \]

such that

- For all, \((p, x) \in \Sigma^* \times \Sigma^*\), \(M(p, x)\) is an SCR \(\in\) TRIANGLE if and only if \(x\) is an SCR \(\in\) TRIANGLE.
- \(p < q, |p| = |q|\) and \(|x| = |y|\) \(\Rightarrow\) \(M(p, x) < M(q, y)\).
- If \(|x| = |y|\) and \(|p| = |q|\) then \(|M(p, x)| = M(q, y)|\) and if \(|x| < |y|\) and \(|p| \leq |q|\) then \(M(p, x) < |M(q, y)|\).

We use the following scheme:

Given a representation \(x\) (in the above encoding) of a circuit \(C\) with \(n\) nodes, and a string \(p \in \Sigma^*\), we form another circuit \(C'\) with \(n + |p|\) nodes, which has the following properties:

1. The first \(n\) nodes in \(C'\) are exactly the same as \(C\).
2. The next \(|p|\) nodes are not connected to any of the previous \(n\) nodes.
   Their connections among themselves in the graph corresponding to \(C'\) are defined in terms of \(p\) as follows: there is an edge from the node \(n + 1\) to the node \(n + j\), if and only if \(p_{j+1} = 1\), where \(p_{j+1}\) is the \(j\)th bit of \(p\).
3. No other new edges are introduced apart from those in step 2 above.

The encoding for the resulting circuit \(C'\) therefore takes \((n + |p|)^2 + (n + |p|) + \log([n + |p|])\), and clearly satisfies all the above conditions. The only problem that may arise is that the graph for the resulting circuit may become cyclic if \(p_1 = 1\). However, since the only cycles that we can have in this graph are such self loops, the algorithm for evaluating the circuit’s output may eliminate all self-loops to start with.

The above proofs for the regular and monotonous paddability of TRIANGLE essentially do not change the outputs of the circuit for a given input, and hence would apply to all problems which can be shown \(\mathcal{NP}\) complete under such an encoding. Thus, we have effectively shown that any graph problem which is \(\mathcal{NP}\) complete under a succinct circuit encoding satisfies Livne’s criteria.
7 Conclusion and Future Work

The dist $\mathcal{NP}$ completeness of graph problems under Succinct Circuit Encodings presents further justification for the belief that all $\mathcal{NP}$ complete problems should have dist $\mathcal{NP}$ versions. Along with the incompleteness theorem, it also suggests that one should study those reductions between problems where the range of the reductions can be compressed using a succinct encoding. The intuition that this might lead to another version of Incompleteness results from the fact that even the Incompleteness theorems given above also apply a similar trick in the opposite direction, as both of them proceed by padding an instance of a problem in a given class and then taking advantage of the definition of dist $\mathcal{NP}$ completeness to show that completeness under the distributions in question would lead to a complexity collapse.

Acknowledgements. I would like to take this opportunity to express my gratitude to my mentors, Dr. Leonard Schulman and Dr. Christopher Umans for their guidance all through the project, and for the enriching discussions that I had with them on this and other topics in Computer Science and Quantum Computation. I also thank the CalTech Students Faculty Programs Office for their support throughout the summer.

References


