COMMENTS FROM Erika Melder.

1. Page 223
   Would use explicit multiplication of some sort rather than relying on implicit multiplication by adjacency in $cOPT(x)$ and $cAlgorithm(x)$. You use $\times$ on page 224, so it's a consistency thing.
   In part 2 of the caveats: "or example" SHOULD BE "For example"

2. Pg 224: In Defn 9.7 bullet 5: does $c \geq 1$? If so, that's a typo; if not, then it should be highlighted since it breaks the pattern.

3. Pg 226:
   First paragraph: triangle inequality is backwards, should be $w(a,b) + w(b,c) \geq w(a,c)$
   Proof is not really "by contradiction" unless you're assuming that $P \neq NP$, it seems direct to me

4. Pg 232:
   Bullet 2 should say "such a c, d exist" rather than "exists"

5. Pg 233:
   Bullet 4 says "approximated within a $(1 - o(1)) \ln n". Either "a" should be removed or it should be "a factor of"
   Bullet 5 is missing a period after "together"

6. Pg 235:
   Bullet 3 should say "such a c, q exist" rather than "exists"

7. Pg 238:
   Bullet 3.(1).(c) has an extra overbar and set of parentheses around $G$
   Bullet 3.(2).(b) has the words "Independent Set" rather than "$OPT_{IndependentSet}"

8. Pg 239:
   Theorem 9.39 bullet 2: The bullet starts by citing Dinur and Safra but then concludes with a redundant second citation of Dinur and Safra
   Immediately after Theorem 9.39: "There are no NP-hardness result known" SHOULD BE should be "results"

9. Pg 240:
   There is a period after "epsilon-Gap-Max Rep" in the definition box that is not after any other problem names in their respective definition boxes.
   Is this because "Rep" is short for "representative"? That abbreviation period is not used elsewhere when the problem is named.
10. Pg 241:

In the definition of epsilon-Gap-Min Rep the same period is after "Rep"
"There is label covering of size 2S" SHOULD BE "There is a label covering
of size 2S"

COMMENTS FROM Gaston Bitro

1. Page 22 Example 0.9 14 and 15 0-1 / Integer programming not clear if d
   is a vector or an integer, it does not have the arrow in xc ≥ d.

2. I just noticed it also says "components are not to big" in page 22 example
   0.9 15

COMMENTS FROM Amin Shiraz Gilani

1. Page 24
   After Definition 0.16: should be x ∉ A (rather than x ∈ A)?

2. Page 27
   Should “We write Discrete Log for both the function and decision version.”
   be removed?

3. Page 30
   Exercise 0.27. 2.: Should not it be “replace 3/4 with 1/2 + α(|x|)” instead
   of “replace 3/4 with α(|x|)”?

4. Page 31
   0.12 Counting Problems: Should it not be
   “If #SAT(x) = L then #CLIQUE(f(x)) = L” rather than
   “If #SAT(x) = L then #f(CLIQUE) = L”
   And
   “If #SAT(x) = L then #CLIQUE(f(x)) = g(L)” rather than “If #SAT(x) =
   L then #f(CLIQUE) = g(L)”?

5. Page 33
   First paragraph: Instead of “if and only if Alice can win”, should not it
   be “if and only if you can win”?

6. Page 35
   Definition 0.40 in 0.17 Arithmetic Hierarchy: In Items 1 and 2, should it
   not be M(y) instead of M(x)?
The EOL problem, as stated, can be easily confused with the EOL augmented problem, which is actually PSPACE hard. Maybe it’s a good idea to clarify that there can be more than one unbalanced node other than 0^n and the goal is to find any one of those?

COMMENTS FROM Divesh (from the blog)

1. ABGS21 does not show $2^{(1-\epsilon p)n}$ hardness for $ShVecProb_p$, but shows $2^{C_p n}$ hardness where the constant $C_p$ grows with $p$, and $C_p$ approaches 1 when $p$ tends to infinity.

2. The following statement is ambiguous. "This problem is the basis for many post-quantum crypto systems. Thats just a fancy way of saying crypto systems that do not depend on number-theoretic hardness assumptions.” How would you define what a number-theoretic hardness assumption is? Why would, say, learning with errors not qualify?

COMMENTS FROM Eric Allender

Of course, the first thing that I checked was to see which papers of mine were cited. This led me to make the following suggestions:

1. On page 27, you discuss MCSP. Actually, both Shuichi Hirahara and I would now conjecture that MCSP might be NP-complete, under randomized or non-uniform reductions. And the citation that you give for [All20] points to an older version of the survey. I would suggest rewriting this paragraph along the following lines.

Kabanets & Cai [3] showed that if MCSP $\in$ P then there is no secure crypto system. We take that as evidence that MCSP $\notin$ P. Can we show that with a reduction? Probably not: Murry & R. Williams [6] showed that if MCSP is NP-complete under Karp reductions then some longstanding open problems would be solved. This does not indicate that there is no such reduction; however, it indicates that finding one will be hard.

For a survey of a large body of work on this topic, see the papers of Allender [1] and Hirahara-2022 [2].

2. I notice that you’re giving the URLs for some papers via my home page, rather than the ”official” URL. Maybe this is so readers can avoid pay walls. It’s not clear that Rutgers will continue to keep my home page links up and running after I die, so those links may get stale. But hopefully not for a while.
I just looked at your latest draft. You’re defining the problem (on p.47) to have clauses of length at most 3 with exactly \(b\) occurrences per variable… you even mention that a clause can have the same variable twice.

As the reduction below indicates, we can actually set \(b = 3\) in that case, not just \(b = 4\), and still get an NP complete problem.

For \(b = 2\), we can apply the monotone literal rule and resolution to solve the problem in \(P\).

HERE IS OLD email of ryan that has that reduction.

I’m struggling to understand why the following thing doesn’t show that 3 variable occurrence version isn’t also hard, it’s a simple modification of the typical reduction from 3sat to 3sat with at most 3 occurrences per variable.

Let \(F\) be a 3CNF (at most 3 literals per clause).

Let \(M\) be an upper bound on the number of occurrences of any variable in \(F\).

For each variable \(x\) we make \(2M\) new variables, \(x_1, \ldots, x_{2M}\), and we relate them with the following clauses:

- For all odd \(i\), add \((-x_i \lor -x_{i+1}\),
- For all even \(i < 2M\), add \((x_i \lor x_{i+1}\).
- For \(i = 2M\), add \((x_{2M} \lor x_1\).

This has the effect of creating a ”cycle” of equivalent literals, where the odd variables are all equivalent to the *negations* of the even variables.

Now we can, for every variable occurrence of \(x\) in every clause, plug in one fresh copy of an odd or even variable \(x_i\) (depending on whether \(x\) appears positive or negative) to make every clause monotone. There are three occurrences total for each variable.

I’m not sure why this doesn’t work … even if you required clauses of length exactly 3, one can simply make another copy of one of the variables (thus bringing the total number of occurrences per variable to 4).

Is the issue that you wanted (for some reason) three *distinct* variables in each clause?

BASED ON Ryan’s email Erik D asked

Why do we cover monoton 3SAT-4 instead of 3SAT-3

Do you in fact show Monoton 3SAT-E3 or ESAT-4 or ESAT-3 hard? That would be different. (But its doot to add the appropriate E’s for Exact in this case– its very confusing when authors omit them.)
COMMENTS FROM Harry Lewis.

1. You are kind to cite my trivial result about Horn sets. More interesting are the attached two papers, which I would not have mentioned except that you are SO exhaustive and these are more worth noting than the Horn set paper. One is in the spirit of Schaefer’s dichotomy theorem, except that instead of assuming CNF, it works off what boolean operators are allowed, relying on an exhaustive but little known analysis by Post. The other pretty much exhausts what there is to say about the complexity of easily characterized decidable subclasses of the predicate calculus. The papers are [5] and [4].

2. Try to cut down on Google SO AND SO or Look at Wikipedia.

COMMENTS FROM BILL GASARCH

1. Check that the VC stuff was fixed: Here is what is true
   ALM: $\exists c$, VC cannot be done in cOPT. No explicit c.
   I proved 1.107 using theorems already known. That is the 1.107 result in
   the book.
   Hastad got 1.166, as stated in the book
   Dinur bot 1.3666.. as stead in the book
   Rewrite to say they prove there was a constant.

2. Put in stuff about Pocklington, Cobram, Edmonds

3. We now have an open problems env- use it.

4. We have a projects env- use it. (check that we have it)

5. Repetition between PCP chapter and UGC Chapter?

6. Put in new pictures that Auguste got.

7. Mention P-complete and a few problems which are P-complete. Mention that the exact relation of PRAM, MPC, and AMPC is not known (indeed these are big open problems in the field) and that so far NONE of P-complete problems can be solved in polylog rounds in MPC or AMPC

8. In Chapter 0 add that one of the parameters for MPC is number of machines.

9. PPAD chapter- look into making some of the figures in LaTeX, either me or Auguste. Also try to find where some of them came from.

10. Those figures that have some words in them that Auguste redid- check those out.
COMMENTS FROM MARCUS SCHAEFER

1. Why is this (Section 17.9) in the below NP section? Should be in above NP?

2. I’m missing a discussion why ER matters, beyond just NP-hardness. Many papers think the computational complexity of a problem is well-enough determined if you know it’s NP-hard. I’ve tried arguing against that. First time in Complexity of Some Geometric and Topological Problems in Graph Drawing 2009, but several times since.

3. One aspect that differentiates ER from NP is the lower bounds on the geometric complexity of realizations you typically get (double exponential precision, e.g. for segment intersection graphs or rectilinear crossing number). E.g. segment intersection graphs may need a double-exponential size grid (for endpoints) to realize. And that remains true even if each segment intersects only a fixed number of additional segments (as I showed in On the Complexity of Some Geometric Problems With Fixed Parameters, Journal of Graph Algorithms and Applications, 25(1), 2021).


References


