CMSC 858M: Algorithmic Lower Bounds: Fun With Hardness Proofs
Fall 2021

Instructor: Mohammad T. Hajiaghayi
Scribe: Alexander Mendelsohn

1 Overview

This chapter discusses different classes of NP-hardness as well as polynomial algorithms. It will look at the 3-PARTITION problem to introduce the strongest form of NP-hardness, and use reductions from it to prove several problems are NP-hard.

2 Types of NP-Hardness

A number problem is a problem which takes at least one integer as an input (rationals can be encoded as integers, but irrational numbers won't be addressed here).

A problem is **weakly NP-hard** if it is NP-hard when the numbers are represented in binary.

A problem is **strongly NP-hard** if it is still NP-hard when the numbers are given in unary.

Let $a_1, ..., a_n$ be the input to a problem, and let a_{max} be the largest element of the input.

A problem is **pseudopolynomial** if it is polynomial on both n and a_{max}.

A problem is **weakly polynomial** if it is polynomial on n and $\log(a_{max})$. This is typically what is meant when an algorithm is considered polynomial.

A problem is **strongly polynomial** if it is polynomial on n by itself, regardless of the value of a_{max}.

If $\mathsf{P} \neq \mathsf{NP}$, a problem that is weakly NP-Hard has no weakly polynomial algorithm to solve it, even if there might be a pseudopolynomial algorithm. If a problem is strongly NP-hard, then there is no pseudopolynomial algorithm to solve it either.

In practice, a_{max} is often polynomial in n, in which case pseudopolynomial algorithms are polynomial in n and can often be efficient. The KNAPSACK problem, for example, has a pseudopolynomial dynamic programming algorithm and is often efficient in practice.

3 Partition Problems and Scheduling

PARTITION (often called 2PARTITION but not here) is the problem of whether a multiset (a set that can contain duplicates) of numbers can be split into two subsets of equal sum. **SUBSET SUM** is the generalization where you just need to find one subset whose sum is some specified target. Both problems are weakly NP-complete.

3PARTITION is a very useful strongly NP-complete problem. It is, given a multiset A of n integers where n is a multiple of 3, can A be partitioned into $\frac{n}{3}$ subsets of equal sum?

3PARTITION is NP-hard even when the elements of A are expressed in unary, which makes it strongly NP-hard.

If we let t be the target sum that all of the subsets of A need to sum to, 3PARTITION is still strongly NP-hard if we restrict every element of A to be strictly between $\frac{t}{4}$ and $\frac{t}{2}$. This requires that every subset is of size exactly 3.

NUMERICAL-3D-MATCHING is the version of 3D-MATCHING which has numbers in it. It is a specialization of 3PARTITION in which there are 3 multisets, A, B, C and the task is to break them up into n subsets which each contain exactly one element of each of A, B, and C and which all have the same sum. This problem is also NP-hard, even when the elements are expressed in unary, and is thus strongly NP-hard.

3DM (defined in chapter 3) is a generalization of N3DM, and X3C (also defined in chapter 3) is a generalization of 3DM. Both are strongly NP-complete.

The MULTIPROCESSOR SCHEDULING problem is a very useful problem in optimization. It goes, given p identical processors and n jobs with completion times $a_1, ..., a_n$ which all need to be completed on a single processor, is it possible to assign jobs to processors such that all finish before time t. This problem can be trivially reduced from 3PARTITION and is therefore strongly NP-complete.

3.1 Packing Problems

Packing puzzles are (sometimes) fun games to play in which the player needs to fit some shapes into a larger shapes.

One example of a packing problem is RECT-RECT PACKING in which the goal is to pack a collection of rectangles into a larger rectangle. Rotation and translation are both allowed, and they do not need to cover the entire larger rectangle. Unlike many problems in this book, it is not obvious that this problem is in NP because encoding rotations efficiently is difficult. Whether this problem is in NP is actually an open question. It is, however NP-hard even when restricted to not allow gaps or rotations. This is shown by a reduction from 3PARTITION.

SQ-RECT-PACKING is packing squares into rectangles, and is also strongly NP-complete by a reduction from 3PARTITION. So is SQ-SQ-PACKING (packing squares into squares).

4 Puzzles

This section will show that four puzzles are equivalent and NP-complete.

An EDGE MATCHING PUZZLE (EMP) is a set of unit squares with colored edges, and a target rectangle. The goal is to fit all the squares into the rectangle such that adjacent edges all have the same color. This is proven NP-complete by a reduction from 3PARTITION, by first using a gadget composed of tiles forced together in a specific way to create rectangles of a specific length that can represent the numbers in the 3PARTITION problem.

SIGNED-EDGE-MATCHING is like EDGE-MATCHING except instead of adjacent edges having the same color, there are some predetermined pairs of colors and each edge has to belong to one of the pairs. There is a reduction from EMP to SEMP which shows SEMP is NP-complete.

JIGSAW PUZZLES, when defined rigorously, have a reduction from SEMP which shows they are NP-complete.

POLYOMINO PACKING is fitting polyomino pieces (sets of unit squares glued together) into a rectangle with translation and orthogonal rotation allowed (Tetris pieces are polyominos with 4 unit squares). This is a generalization of RECT-RECT-packing, so it is obviously NP-hard. However, it is still NP-hard when the polyominos are restricted to be small (area $O(\log^2(n)$ where n is the number of pieces).

There is also reduction from POLYOMINO-PACKING to EMP. This closes the loop of equivalence for these four games, and all of the reductions in this loop have blowup at most logarithmic.

5 Three Dimensional Games

The EDGE FOLDING PROBLEM is the question of whether a 3D polyhedra can be cut along the edges so that it unfolds into a connected non-overlapping 2D shape. This is strongly NP-hard, by a reduction from SQ-SQ-Packing

SNAKECUBE is given a set of 'cubras' (sets of cubes connected by elastic, some of which are forced into a straight line, but some of which can move 90 degrees in any direction), can the 'cubras' be assembled into a cube. This problem is NP-hard by a reduction from 3PARTITION.

DISK PACKING is given a set of disks and a square, can you place all of the disks non-overlapping so that all of the centers are within the square. This is also NP-hard by a reduction from 3PARTITION.

Clickomania is a computer game which can be found easily on the internet. The goal is to clear blocks by clicking on (and destroying) contiguous groups of blocks, after which the blocks above will fall to fill the empty space. Clickomania is P with only one column of blocks, but it is NP-hard with at least 2 columns and 5 colors, or 5 columns and 3 colors. This is by reduction from 3PARTITION.

In TETRIS, it is NP-hard to decide whether there is a winning sequence of moves, or how to maximize the number of blocks (or lines of blocks) that can be placed before death. Even just getting a good approximation of the latter is NP-hard. Note that this is when all of the future pieces are known in advance.

With the following restrictions, however, it is an open problem whether TETRIS is NP-hard:

1. Tetris with an initially open board
2. Tetris with a constant number of rows or columns
3. Tetris with a restricted set of pieces
4. Tetris without last-minute slides (every piece needs to be dropped from the sky
5. Tetris where you don't know the future sequence of pieces
6. 2-Player Tetris

6 1-Planar Graphs

A 1-planar graph is a graph that can be drawn with each edge crossing at most one other edge. Deciding whether a graph is 1-planar is NP-complete by a reduction from 3PARTITION. The proof is based on a gadget for an uncrossable edge (an edge that can't be crossed without causing some edge to cross two edges).

7 Ivan's Hinge

Ivan's Hinges are hinged loops of colored triangles in a plane. It is NP-hard by reduction from 3PARTITION to decide whether they can fold into a particular shape.

8 Carpenter's Rule

A Carpenter's Rule is a foldable ruler which has sections that can fold back on each other. The task is to fit the ruler into a long straight box of a given size. This is *weakly* NP-hard by a reduction from PARTITION, and like PARTITION it has a pseudopolynomial algorithm.

9 Map Folding

The problem of whether a map and a given pattern of creases is foldable is weakly NP-hard for orthogonal paper and creases, or for rectangular paper and orthogonal or diagonal folds. It is an open problem whether this is strongly NP-hard or if there exists a pseudopolynomial algorithm.

10 Other Interesting Problems

SRP[2] (Subset Sum with Repetitions) is a generalization of the subset sum problem where you have positive integers $a_1, ..., a_n$ and $r_1, ..., r_n$ and a target t, and the task is to determine whether there are integers x_i, such that $0 \leq x_i \leq r_i$ and $\sum_{i=1}^{n} a_i x_i = t$. This problem contains the subset sum problem so is clearly NP-hard. The interesting thing is that while subset sum is polynomial if you restrict the elements to be superincreasing (every element is greater than or equal to the sum of all of the previous elements), SRP is still NP-hard even if the a's are superincreasing and the r's are all equal to 1 or 2.

The k-Subset-Sum problem is whether the multiset has a subset of size exactly k which sums to the target. This problem is also weakly NP-hard.

The Bin Packing problem is similar to the multiprocessor scheduling problem. In the multiprocessor scheduling problem you are given a number of processors p and need to know if it is possible to finish before the given time limit. In the bin packing problem, you are given only a time limit (volume of the bins) and need to figure out how many processors (number of bins) you need to finish in time. Obviously this is also NP-complete, because you can reduce the multiprocessor scheduling problem to it by just adjusting the time limit until you get the right number of bins.

The Multi-Subset-Sum problem is given a multiset of integers and a target, what is the maximum number of disjoint subsets that sum to the target. This is strongly NP-hard, by a reduction from 3PARTITION.

4Partition is the same as 3Partition but with $\frac{n}{4}$ sets instead of $\frac{n}{3}$. It is also NP-complete because 3PARTITION can be reduced to it.

TRI-RECT-PACKING is the problem of packing triangles into a rectangle.[3] This problem is strongly NP-hard even when restricted to right triangles by a reduction from 3PARTITION.

TRI-TRI-PACKING is the problem of packing triangles into a triangle.[3] This problem is strongly NP-hard even all of the triangles are right triangles, or when all of them are equilateral triangles. The right triangle version is proven by a reduction from 3PARTITION, while the equilateral triangle version uses a reduction from 4PARTITION instead.

References

[1] Erik Demaine, William Gasarch, and Mohammad Hajiaghayi *Fun With Hardness: Algorithmic Lower Bounds*

[2] J.L. Ramirez Alfonsin *On variations of the subset sum problem* May 1995

[3] Amy Chou *NP-Hard Triangle Packing Problems* Jan 2016