Lecture 23 Classical NP-Hard Problems

April 2020

1 Hamiltonian Path to TSP Cycle

1.1 Problem Analysis

Hamiltonian Path: Given a graph G (directed/undirected), a start vertex s and an end vertex t, find a path from s to t that visits every vertex in G exactly once.
Travelling Salesman Problem: Given a weighted graph G, and there is a salesman who wants to start at a vertex s, visit all vertices and come back to s.
Similarity: Visit all vertices
Differences:

1. Hamiltonian Path is on unweighted graphs while TSP is on weighted graphs
2. Path vs. Cycle

1.2 Proof

Given a Hamiltonian Path instance with n vertices.To make it a cycle, we can add a vertex x, and add edges (t,x) and (x,s). To make the path weighted, we can give a weight 1 to all edges. Set $L = n + 1$, we now have a TSP cycle instance.
Thus we can conclude that for any Hamiltonian path P in the original graph, the new set of edges $P \cap (t, x), (x, s)$ form a TSP cycle of length $L = n + 1$
For any cycle of length $n + 1$ that visits every vertex, it must visit every vertex exactly once. Cut the cycle when it visits vertex x, (either by (t, x) or (x, s)). We can thus get a path from s to t that visits every vertex in the original graph exact once.

2 3-SAT to quadratic programming

2.1 Problem Analysis

2.1.1 3-SAT Problem

example: $(X_1 \vee X_2 \vee \overline{X_3}) \wedge (X_2 \vee \overline{X_3} \vee \overline{X_4}) \wedge ...$
clause: or of 3 literals
literal: X or $\overline{X}$
formula: logical and of many clauses
goal: is there any assignment to the variable such that all clauses are satisfied.

2.1.2 Quadratic Programming

variables: $x_1, x_2, ..., x_n \in \mathcal{R}$
constraints:

$$\begin{aligned} x_1^2 &\leq 3 \\ x_1^2 + 2x_1x_2 - x_2^2 &\geq 5 \\ x_1^2 - 2x_2 &\geq x_3^2 - x_2x_3 \\ &..... \end{aligned}$$

goal: Is there an assignment to the variables such that all constraints are satisfied.

2.1.3 Comparison

1. boolean variables vs. real variables
2. 3-SAT clauses vs. quadratic constraints

Decision Problem: Given a set of quadratic constraints, does there exist a feasible solution for 3-SAT?

2.2 Reduction

In order to create an instance of Quadratic programming from an instance of 3-SAT, we need to add constraints to make the real variables binary(0/1) but adding constraint $y_i^2 = y_i$ for all variables y_i. The negation of the variables in 3-SAT clauses thus becomes $1 - y_i$ in quadratic constraints.
Thus, for every clause in 3-SAT, we thus create a quadratic constraint out of it. For instance, from the clause $x_a \vee x_b \vee \overline{x_c}$ we can create a constraint $y_a + y_b + (1 - y_c) \geq 1$.

3 Tripartite matching to subset vector

3.1 Problem Analysis

3.1.1 Tripartite matching

Given three sets U, V and W, each containing n vertices, and hyperedges (u, v, w), where $u \in U$, $v \in V$ and $w \in W$. A tripartite matching is a way of selecting n hyperedges, so that every vertex is adjacent to a hyperedge. If we want to use n hyperedges to cover all $3n$ vertices, then we must use exactly one vertex is each of the n edges. Is there is a tripartite matching in the given graph?

3.1.2 Subset Vector

Given n vectors $\{v_1, v_2, ..., v_n\}$ and a target vector u. The answer is YES iff a subset of these vectors sums up to u.

3.1.3 Comparison

1. select hyperedges vs. select vectors
2. exact n hyperedges vs. select any number of vectors.

Idea: Encode hyperedges as vectors. For each hyperedges, it includes 3 out of $3n$ vertices. Thus, we can convert each hyperedge to a one-hot encoding the vertices in the edge. The encoding will be vectors of $3n$ dimension. For instance, for edge (u_1, v_1, w_1), the encoding E will have 0s in all other indices while $E[0], E[n+1], E[2n+1]$ will be 1s.
Thus, selecting n hyperedges in order for every vertex in the graph is adjacent to exactly 1 hyper-edge is equivilent to having the sum of these hyper-edge encoded vectors equals to $\vec{1}$.

4 Subset vector to subset sum

4.1 Problem Analysis

4.1.1 Subset vector

Given n vectors $\{v_1, v_2, ..., v_n\}$ and a target vector u. The answer is YES iff a subset of these vectors sums up to u.

4.1.2 Subset sum

Given n integers $\{a_1, a_2, ..., a_n\}$ and a target m. The answer is YES iff a subset of these integers sums up to m.

4.1.3 Comparison

1. Any integer can be viewed as a vector if we take its base B representation. For instance, integer 9, if in base 2, can be seen as a vector $[1, 0, 0, 1]$.
2. Sum of numbers are thus, behave like sum of vectors as long as there is no carry operation.
3. When B is very large, when taking sums, there will be no carry operation.