CPSC 320 2023S2: Assignment 5 Solutions

3 4-Dimensional Matching

The *3-Dimensional Matching* (3DM) problem is NP-Complete—see Section 8.6 of Kleinberg & Tardos. In this question you will show the analogous *4-Dimensional Matching* problem is also NP-Complete by completing the reduction below.

Formally an instance I of the 4-Dimensional Matching (4DM) problem consists of four disjoint sets of equal size, namely $A = \{a_1, a_2, \ldots, a_n\}, B = \{b_1, b_2, \ldots, b_n\}, C = \{c_1, c_2, \ldots, c_n\}$, and $D = \{d_1, d_2, \ldots, d_n\}$, and a collection $\mathcal{S}$ of 4-tuples of the form (a_i, b_j, c_k, d_l). The problem is to determine whether there are n 4-tuples of $\mathcal{S}$ such that no two tuples share a common element.

Complete the following argument that 4-Dimensional Matching is NP-complete.

1. (2 marks) First show that the 4-Dimensional Matching problem is in NP.

   SOLUTION: Let $I = (A, B, C, D, \mathcal{S})$ be an instance of 4-Dimensional Matching and let M be a potential solution for I (that is, M is a collection of 4-tuples of elements from A, B, C and D). Our certification algorithm works as follows. We first check that M is of size n, and output "No" otherwise. We also check that M is a subset of S, and output "No" otherwise. For each tuple in M, we flag each set element in the tuple. If any set element is flagged twice, we output "No". Otherwise we output "Yes" (reporting that the solution is good). As there are a total of $4n$ elements in the sets A, B, C and D, the runtime of this procedure is polynomial in n (in fact linear in n).

2. (2 marks) Give a reduction from the 3DM problem to the 4DM problem. Specifically, let X, Y, Z be the sets of size n and $\mathcal{T}$ the collection of 3-tuples of the 3-Dimensional Matching instance. Construct a corresponding 4-Dimensional Matching instance with sets A, B, C, D, and collection $\mathcal{S}$ of tuples.

   SOLUTION: Let $A = X$, $B = Y$, and $C = Z$ with a one-to-one correspondence between set elements (for example, $a_i = x_i$.) We populate the set D with n elements d_i for i from 1 to n. For each tuple $(x_i, y_j, z_k) \in \mathcal{T}$, we create the tuple$(a_i, b_j, c_k, d_l)$ in $\mathcal{S}$ for all l in the range 1 to n.

3. Finally, show correctness of your reduction.

   (a) (2 marks) Show that if $\mathcal{T}$ contains n disjoint 3-tuples, then $\mathcal{S}$ must contain n disjoint 4-tuples.

   (b) (2 marks) Show that if $\mathcal{S}$ contains n disjoint 4-tuples, then $\mathcal{T}$ must contain n disjoint 3-tuples.

   SOLUTION:

   (a) For each of tuple of $\mathcal{T}$, say (x_i, y_j, z_k), our reduction ensures that the corresponding 4-tuple (a_i, b_j, c_k, d_k) exists in $\mathcal{S}$. We claim that these n corresponding tuples in $\mathcal{S}$ are distinct. Clearly the elements a_i, b_j, and c_k of each 4-tuple are disjoint. Since in any two of the disjoint 4-tuples the elements c_k and $c_{k'}$ are distinct, the elements d_k and $d_{k'}$ in the two 4-tuples are also distinct. Thus the n 4-tuples are disjoint.

(b) Next, suppose that $\mathcal{S}$ contains n disjoint 4-tuples. For each such 4-tuple (a_i, b_j, c_k, d_l), let the corresponding 3-tuple be (x_i, y_j, z_k). These 3-tuples are also disjoint, and by our construction they are also in $\mathcal{T}$. Thus $\mathcal{T}$ contains n disjoint 3-tuples.

4 Chemical Graphs

Chemists use graphs to model the structure of complex molecules. Nodes represent atoms, and undirected edges connect atoms that share a bond. Your chemist friend has access to a large database of such graphs, and is curious to discover structural patterns of significance that might be common across molecules.

After some discussion, your friend formulates the following problem as being of interest: Given two undirected graphs, what is the largest subgraph that they have in common?

More formally, let $G = (V, E)$ and $G' = (V', E')$ be undirected, unweighted graphs. Let $V = \{1, 2, \ldots, n\}$. We say that G is a *subgraph* of G' if there is a list $v'_1, v'_2, \ldots, v'_n$ of n distinct nodes of V' such that for all i, j with $1 \leq i, j, \leq n$, (i, j) is in E if and only if (v'_i, v'_j) is in E'.

The Largest Common Chemical Structure optimization (maximization) problem is as follows. A problem instance is a pair $G' = (V', E')$ and $G'' = (V'', E'')$ of undirected, unweighted graphs. The problem is to find a graph G with as many nodes as possible, such that G is a subgraph of both G' and G''.

Having learned about NP-completeness, you realize that there's little hope for an efficient algorithm.

1. (2 marks) Write down a decision version of the optimization problem, which we'll call LCCS (for Largest Common Chemical Structure).

   SOLUTION: An instance of LCCS is a pair of undirected, unweighted graphs $G' = (V', E')$ and $G'' = (V'', E'')$, plus a positive integer k. The problem is to determine whether G' and G'' have a common subgraph with at least k nodes.

2. (2 marks) Show that your LCCS decision problem is in NP.

   SOLUTION: A certification algorithm takes as the problem instance (G', G'', k), as well as two lists of nodes $v'_1, v'_2, \ldots, v'_k$ and $v''_1, v''_2, \ldots, v''_k$, where $k \leq \min\{|V'|, |V''|\}$. For each pair $(i, j), 1 \leq i, j \leq k$, the algorithm checks that $v'_i \neq v'_j$, that $v''_i \neq v''_j$, and that $(v'_i, v'_j) \in E'$ if and only if $(v''_i, v''_j) \in E''$. If all checks pass, the algorithm outputs "yes" and otherwise outputs "no".

   This algorithm can be implemented to run in time $O(n^2)$, where $n = \max\{|V'|, |V''|\}$, if we use an adjacency matrix to represent E' and E'', and so runs in polynomial time.

3. (8 marks) Show that your LLCS decision problem is NP-complete. Make sure that you include all of the needed steps.

   SOLUTION: LLSC is in NP by the previous part, so here we need to show that there is a polynomial-time reduction from a known NP-complete problem to LLCS. We describe a reduction from the Clique problem. An instance of Clique is an undirected graph G and a positive integer k, where k is at most the number of nodes of the graph. The problem is to determine whether the input graph has a clique of size at least k. A clique is a set of nodes of the graph, such that every pair is connected by an edge.

   Our reduction maps an instance (G, k) of Clique to instance (G, C, k) of LLCS, where C is a *complete* graph with k nodes. By complete, we mean that every pair of nodes of C is connected by an edge.

This reduction can be computed in time $O(n + m + k^2)$, where n is the number of nodes of G and m is the number of edges of G, because we can simply copy the description of G, and add a description of C, which has $O(k^2)$ edges. Since $k \leq n$, this is a polynomial-time reduction.

To show that the reduction is correct, we first show that if (G, k) is a Yes-instance of Clique then (G, C, k) is a Yes-instance of LLCS. This is simply because G must contain a subgraph that is a clique of size k, and so this subgraph has exactly the same structure as the graph C. That is, if the nodes of the clique subgraph of G are $v_1, \ldots, v_k$, then every pair (v_i, v_j) is connected by an edge. Also by construction, every pair of the k nodes of C is connected by an edge. So a clique of size k is a subgraph of both G and C.

To complete our proof that the reduction is correct, we show that if (G, C, k) is a Yes-instance of LLCS then (G, k) must be a Yes-instance of Clique. Since (G, C, k) is a Yes-instance of LLCS, G must have a subgraph of size k that is a clique, since this is the only subgraph of size k that can be common to both G and C. Therefore (G, k) is a Yes-instance of Clique.

5 Accordion Folds

As before, let $\mathcal{P}$ denote a thin strip of paper of length ℓ with n vertical folds, located at distances $f_1, f_2, \ldots, f_n$ from the left end of P. If you alternately fold $\mathcal{P}$ to the left and right along some or all of the folds, you get an *accordian fold.*

An instance of the ACCORDIAN FOLDS (AF) problem is the length ℓ of the paper strip, plus the positions $f_1, f_2, \ldots, f_n$ of the folds. The problem is to determine whether there is an accordian fold along a subset of the folds, such that the two ends of the strip of paper are aligned.

1. (6 marks) Show that AF is NP-Complete. You can use the fact that the following problem is NP-complete: An instance of the Split Subset Sum (SSS) Problem is a set $S = \{s_1, s_2, \ldots, s_n\}$ of positive integers. The problem is to determine whether there is a subset of S whose sum is exactly $(\sum_{i=1}^{n} s_i)/2$.

   SOLUTION: In the tutorial we showed that AF is in NP. Here we show a polynomial-time reduction from SSS AF. Together, these two properties show that AF is NP-complete.

   Let $S = \{s_1, \ldots, s_n\}$ be an instance of SSS. For $1 \leq i \leq n-1$, let $f_i = \sum_{j=1}^{i} s_j$ be the sum of the integers up to and including s_i, and let $\ell = f_n = \sum_{j=1}^{n} s_j$. Also let $f_0 = 0$. From instance I of SSS we construct instance I' of AF as follows. The strip of paper $\mathcal{P}$ has length ℓ, and has folds at $f_1, f_2, \ldots, f_{n-1}$. The time to compute the f_i and ℓ is $O(n)$, so the reduction can be computed in linear time.

   We claim that I is a Yes-instance of SSS if and only if I' is a Yes-instance of AF.

   - Suppose that I is a Yes-instance of SSS, and let the two sets S^+ and S^- be a partition of S such that $\sum_{j \in S^+} s_j = \sum_{j \in S^-} s_j$. Without loss of generality, assume that $s_1 \in S^+$. Using the notation from the tutorial, we let $d_0 = +1$ and specify a vector $(d_1, d_2, \ldots, d_{n-1})$ as follows: For each i from 1 to $n-1$, if s_i and s_{i-1} are in the same set, let $d_i = d_{i-1}$. Otherwise let $d_i = -1$ if $d_{i-1} = +1$ and vice versa. The corresponding accordian fold for instance I' is the set of f_i such that $d_i = -1$. This accordian fold ensures that the ith section of the strip (between folds f_i and f_{i-1}, which has length s_i) contributes positively to the net total distance if $s_i \in S^+$, and contributes negatively to the net total sum if $s_i \in S^-$. As a result, the net sum is 0, so I' is a Yes-instance of AF.
   - Suppose that I' is a Yes-instance of AF. Let $(d_1, d_2, \ldots, d_{n-1})$ be a vector of $+1$'s and -1's, with $d_0 = +1$, that specifies an accordian fold with net total distance equal to 0, i.e.,

$$\sum_{i=0}^{n-1} d_i(f_{i+1} - f_i) = \sum_{i=0}^{n-1} d_i s_{i+1} = 0.$$

Create a partition of S as follows: For $1 \leq i \leq n$ put s_i in S^+ if $d_{i-1} = +1$ and put s_i in S^- if $d_{i-1} = -1$. This ensures that

$$\sum_{i \in S^+} s_i = \sum_{i \in S^-} s_i,$$

and so S is a Yes-instance of SSS.