

CS/ECE 374 ✧ Spring 2017

☞ Homework 10 ☞

Due Friday, April 28, 2017 at 10am

Groups of up to three people can submit joint solutions. Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.

The following unnumbered problems are not for submission or grading. No solutions for them will be provided but you can discuss them on Piazza.

- Consider an instance of the Satisfiability Problem, specified by clauses C_1, \dots, C_k over a set of Boolean variables x_1, \dots, x_n . We say that the instance is *monotone* if each term in each clause consists of a nonnegated variable; that is each term is equal to x_i , for some i , rather than \bar{x}_i . Monotone instances of Satisfiability are very easy to solve: They are always satisfiable, by setting each variable equal to 1.

For example, suppose we have the three clauses

$$(x_1 \vee x_2), (x_1 \vee x_3), (x_2 \vee x_3)$$

This is monotone, and indeed the assignment that sets all three variables to 1 satisfies all the clauses. But we can observe that this is not the only satisfying assignment; we could also have set x_1 and x_2 to 1 and x_3 to 0. Indeed, for any monotone instance, it is natural to ask how few variables we need to set to 1 in order to satisfy it.

Given a monotone instance of Satisfiability, together with a number k , the problem of *Monotone Satisfiability with Few True Variables* asks: Is there a satisfying assignment for the instance in which at most k variables are set to 1? Prove that this problem is NP-Complete. *Hint:* Reduce from Vertex Cover.

- Given an undirected graph $G = (V, E)$, a partition of V into V_1, V_2, \dots, V_k is said to be a clique cover of size k if each V_i is a clique in G . Prove that the problem of deciding whether G has a clique cover of size at most k is NP-Complete. *Hint:* Consider the complement of G .
1. Given an undirected graph $G = (V, E)$ a *matching* in G is a set of edges $M \subseteq E$ such that no two edges in M share a node. A matching M is *perfect* if $2|M| = |V|$, in other words if every node is incident to some edge of M . PerfectMatching is the following decision problem: does a given graph G have a perfect matching? Describe a polynomial-time reduction from PerfectMatching to SAT. Does this problem that PerfectMatching is a difficult problem?
 2. A *balloon* is a directed graph on an even number of nodes, say $2n$, in which n of the nodes form a directed cycle and the remaining n vertices are connected in a “tail” that consists of a directed path joined to one of the nodes in the cycle. See figure below for a balloon with 8 nodes.



Given a directed graph G and an integer k , the BALLOON problem asks whether or not there exists a subgraph which is a balloon that contains $2k$ nodes. Prove that BALLOON is NP-Complete.

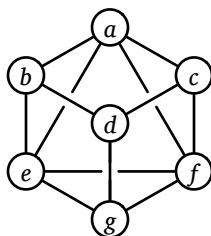
3. Consider the following problem. You are managing a communication network, modeled by a directed graph $G = (V, E)$. There are c users who are interested in making use of this network. User i (for each $i = 1, 2, \dots, c$) issues a request to reserve a specific path P_i in G on which to transmit data.

You are interested in accepting as many of these path requests as possible, subject to the following restriction: if you accept both P_i and P_j , then P_i and P_j can not share any nodes. Thus the Path Selection Problem asks: Given a directed graph $G = (V, E)$, a set of requests P_1, \dots, P_c -each of which must be a path in G - and a number k , is it possible to select at least k of the paths so that no two of the selected paths share any nodes?

Prove that the Path Selection is NP-Complete.

Solved Problem

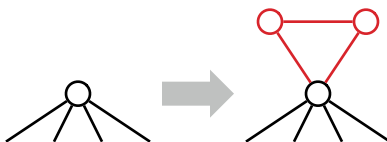
4. A double-Hamiltonian tour in an undirected graph G is a closed walk that visits every vertex in G exactly twice. Prove that it is NP-hard to decide whether a given graph G has a double-Hamiltonian tour.



This graph contains the double-Hamiltonian tour $a \rightarrow b \rightarrow d \rightarrow g \rightarrow e \rightarrow b \rightarrow d \rightarrow c \rightarrow f \rightarrow a \rightarrow c \rightarrow f \rightarrow g \rightarrow e \rightarrow a$.

Solution: We prove the problem is NP-hard with a reduction from the standard Hamiltonian cycle problem. Let G be an arbitrary undirected graph. We construct a new graph H by attaching a small gadget to every vertex of G . Specifically, for each vertex v , we add two vertices v^\sharp and v^\flat , along with three edges vv^\flat , vv^\sharp , and $v^\flat v^\sharp$.

I claim that G has a Hamiltonian cycle if and only if H has a double-Hamiltonian tour.



A vertex in G , and the corresponding vertex gadget in H .

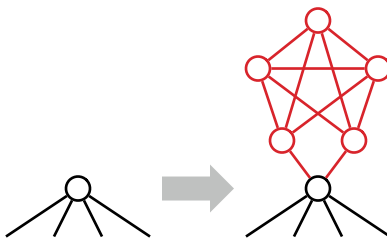
\implies Suppose G has a Hamiltonian cycle $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n \rightarrow v_1$. We can construct a double-Hamiltonian tour of H by replacing each vertex v_i with the following walk:

$$\dots \rightarrow v_i \rightarrow v_i^b \rightarrow v_i^\sharp \rightarrow v_i^b \rightarrow v_i^\sharp \rightarrow v_i \rightarrow \dots$$

\Leftarrow Conversely, suppose H has a double-Hamiltonian tour D . Consider any vertex v in the original graph G ; the tour D must visit v exactly twice. Those two visits split D into two closed walks, each of which visits v exactly once. Any walk from v^b or v^\sharp to any other vertex in H must pass through v . Thus, one of the two closed walks visits only the vertices v, v^b , and v^\sharp . Thus, if we simply remove the vertices in $H \setminus G$ from D , we obtain a closed walk in G that visits every vertex in G once.

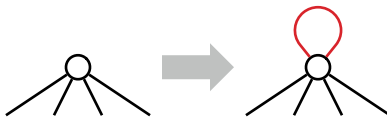
Given any graph G , we can clearly construct the corresponding graph H in polynomial time.

With more effort, we can construct a graph H that contains a double-Hamiltonian tour *that traverses each edge of H at most once* if and only if G contains a Hamiltonian cycle. For each vertex v in G we attach a more complex gadget containing five vertices and eleven edges, as shown on the next page. ■



A vertex in G , and the corresponding modified vertex gadget in H .

Common incorrect solution (self-loops): We attempt to prove the problem is NP-hard with a reduction from the Hamiltonian cycle problem. Let G be an arbitrary undirected graph. We construct a new graph H by attaching a self-loop every vertex of G . Given any graph G , we can clearly construct the corresponding graph H in polynomial time.

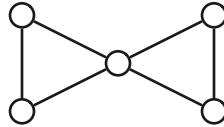


An incorrect vertex gadget.

Suppose G has a Hamiltonian cycle $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n \rightarrow v_1$. We can construct a double-Hamiltonian tour of H by alternating between edges of the Hamiltonian cycle and self-loops:

$$v_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_2 \rightarrow v_3 \rightarrow \dots \rightarrow v_n \rightarrow v_n \rightarrow v_1.$$

On the other hand, if H has a double-Hamiltonian tour, we *cannot* conclude that G has a Hamiltonian cycle, because we cannot guarantee that a double-Hamiltonian tour in H uses *any* self-loops. The graph G shown below is a counterexample; it has a double-Hamiltonian tour (even before adding self-loops) but no Hamiltonian cycle.



This graph has a double-Hamiltonian tour.



Rubric (for all polynomial-time reductions): 10 points =

- + 3 points for the reduction itself
 - For an NP-hardness proof, the reduction must be from a known NP-hard problem. You can use any of the NP-hard problems listed in the lecture notes (except the one you are trying to prove NP-hard, of course).
- + 3 points for the “if” proof of correctness
- + 3 points for the “only if” proof of correctness
- + 1 point for writing “polynomial time”
- An incorrect polynomial-time reduction that still satisfies half of the correctness proof is worth at most 4/10.
- A reduction in the wrong direction is worth 0/10.