NP-Complete

CSCI 570

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

DISCUSSION 13

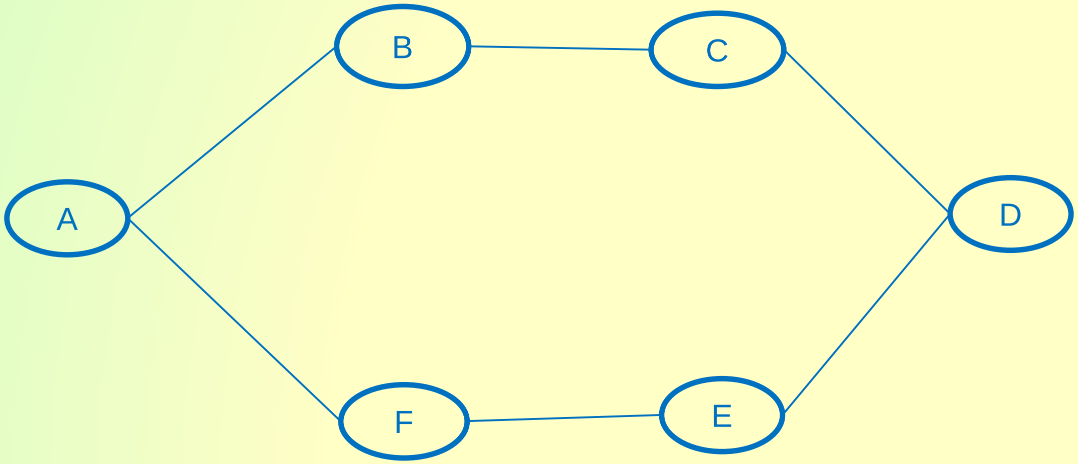
HTTP://WWW-SCF.USC.EDU/~CSCI570

Outline

- Problems with Solutions

Problem 1

- Some **NP**-complete problems are polynomial-time solvable on special types of graphs, such as bipartite graphs. Others are still **NP**-complete.
- Show that the problem of finding a Hamiltonian Cycle in a bipartite graph is still **NP**-complete.

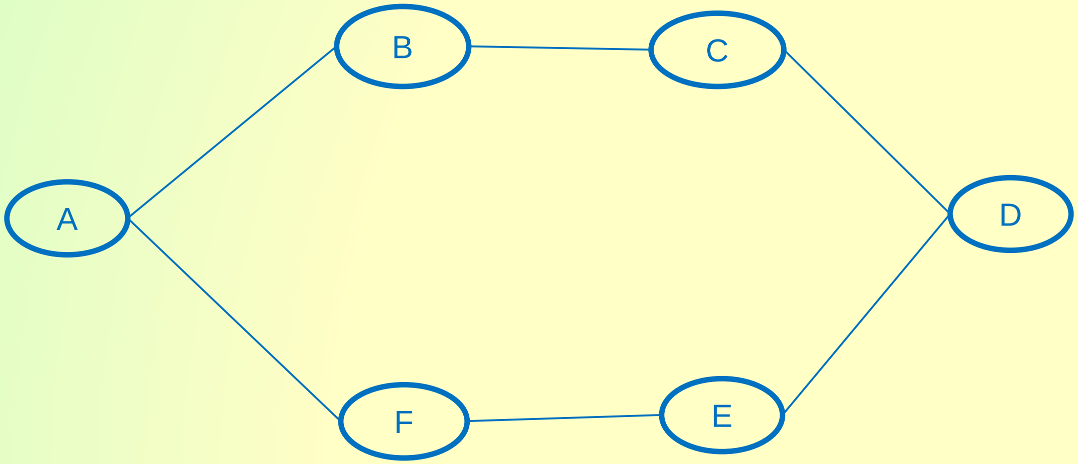


Solution 1

- To prove a problem is NP complete, you must prove two things:
  - The problem is in NP – a certificate can be verified in polynomial time
  - The problem is NP-Hard with respect to NPC problems – given a solution to this problem, convert a known NPC problem to this problem in polynomial time
- The prove the Bipartite Hamiltonian Cycle is in NP, we can use the same proof used for showing that Hamiltonian Cycle is in NP
  - In other words, if we prove that Hamiltonian Cycle is in NP, we have also proven that Bipartite Hamiltonian Cycle is in NP
- The certificate for a Hamiltonian Cycle problem is a set of nodes
- We need to check that the set contains n nodes – O(n)
- We need to check that there is an edge connecting each pair of adjacent nodes in the set – O(n)
- We need to check that there is an edge between the last node and the first node – O(1)

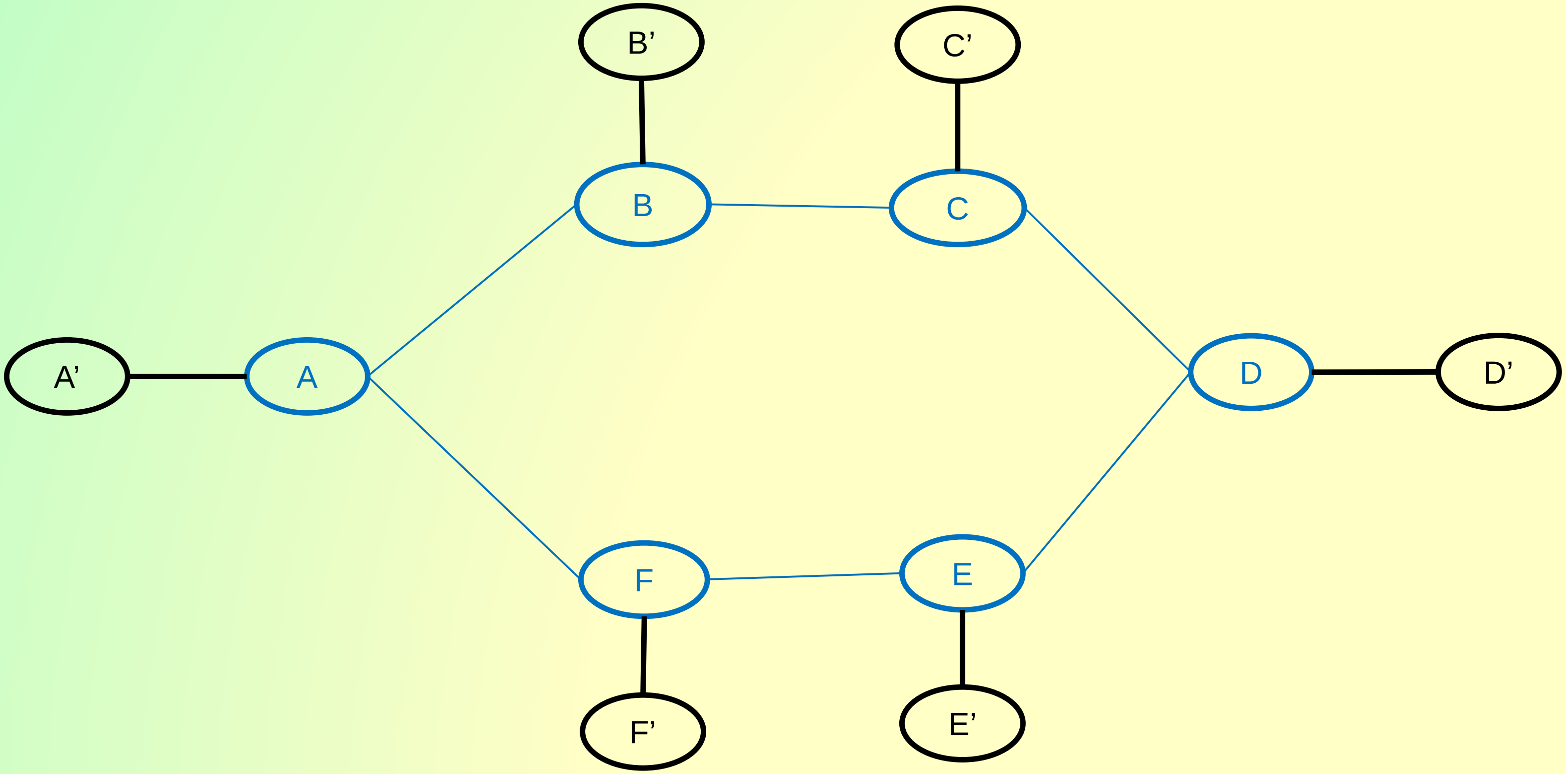
Solution 1

- To prove the problem is NP-hard with respect to another NPC algorithm, we will convert the Hamiltonian Cycle problem to the Bipartite Hamiltonian Cycle problem
  - In other words, assume we have a solution to the Bipartite Hamiltonian Cycle problem. Can we convert the Hamiltonian Cycle problem to the Bipartite Hamiltonian Cycle problem in polynomial time?
  - Given a graph G, we want to convert it to a bipartite graph G' that has a Hamiltonian Cycle only if G has a Hamiltonian Cycle



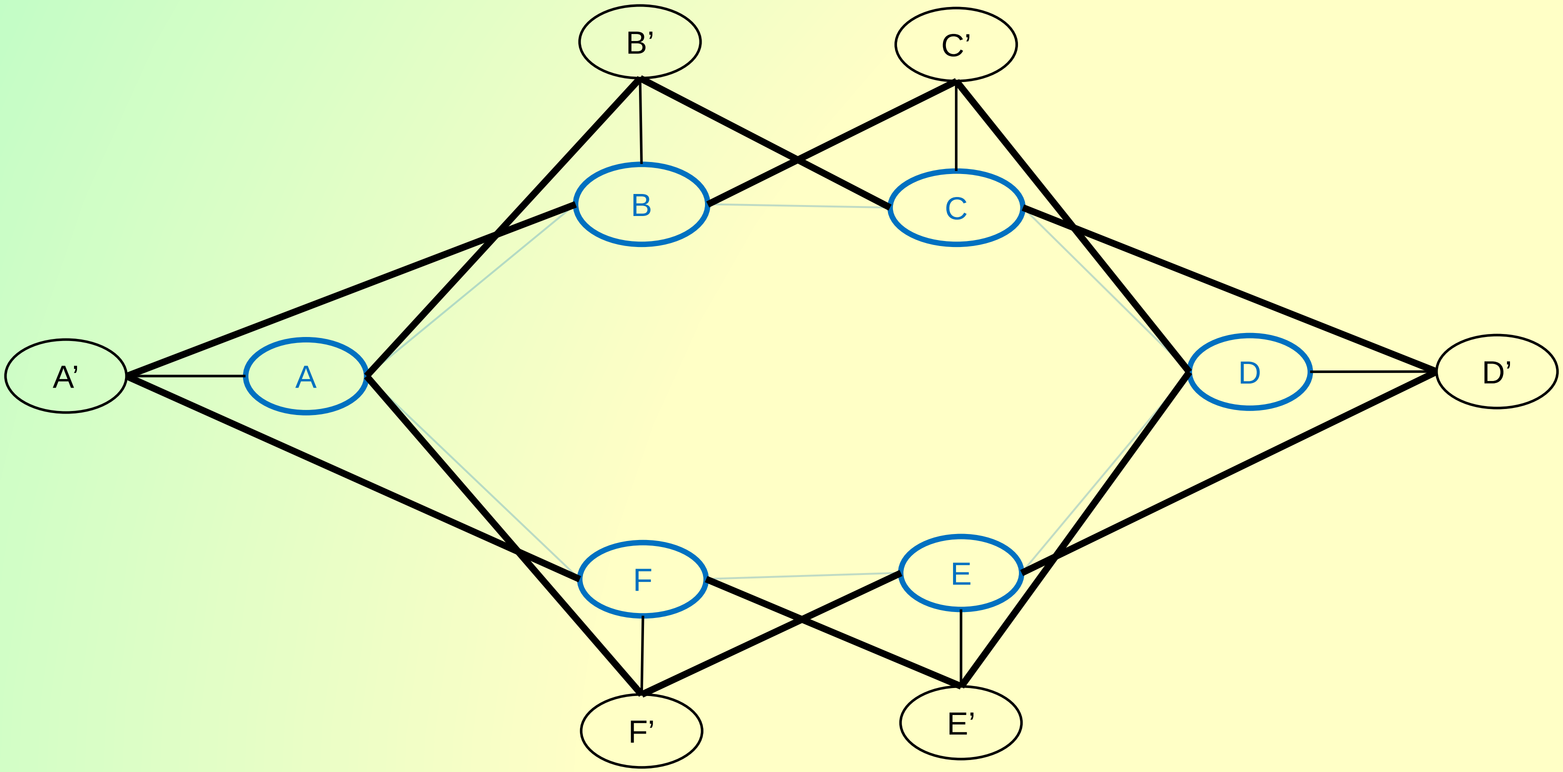
Solution 1

- For each vertex u in the graph, add another vertex u’
- For each vertex u in the graph, add an edge (u, u’)



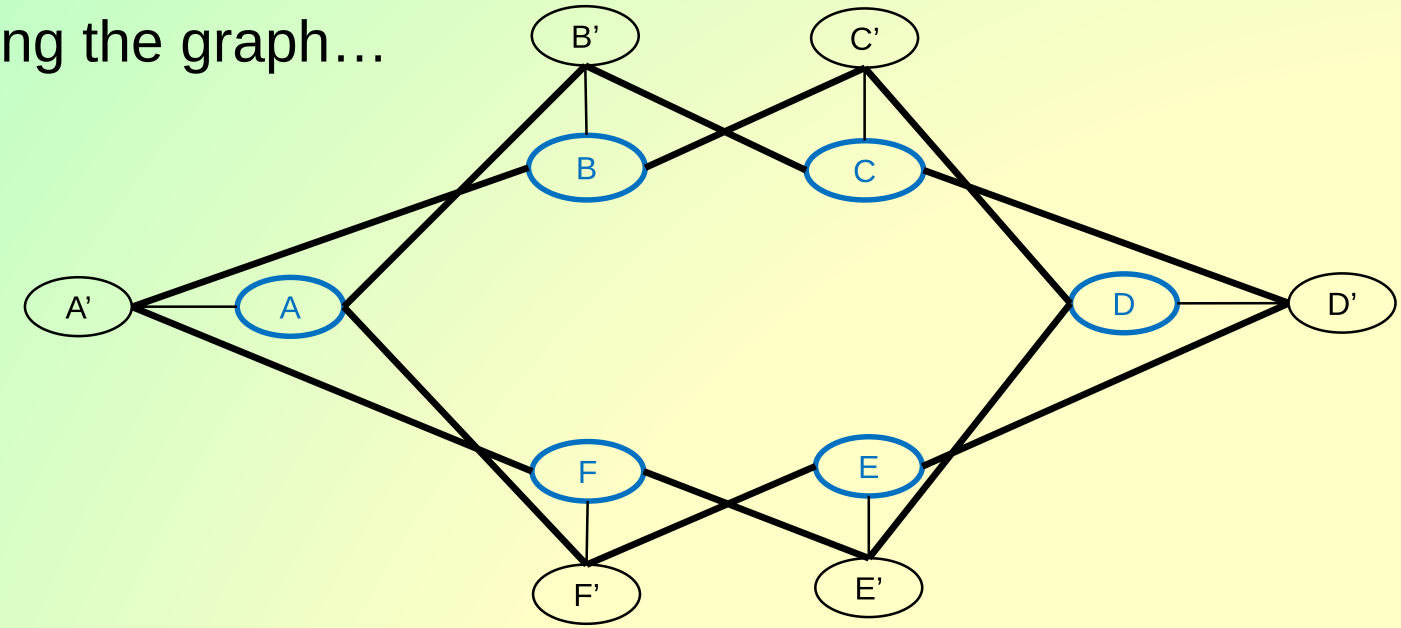
Solution 1

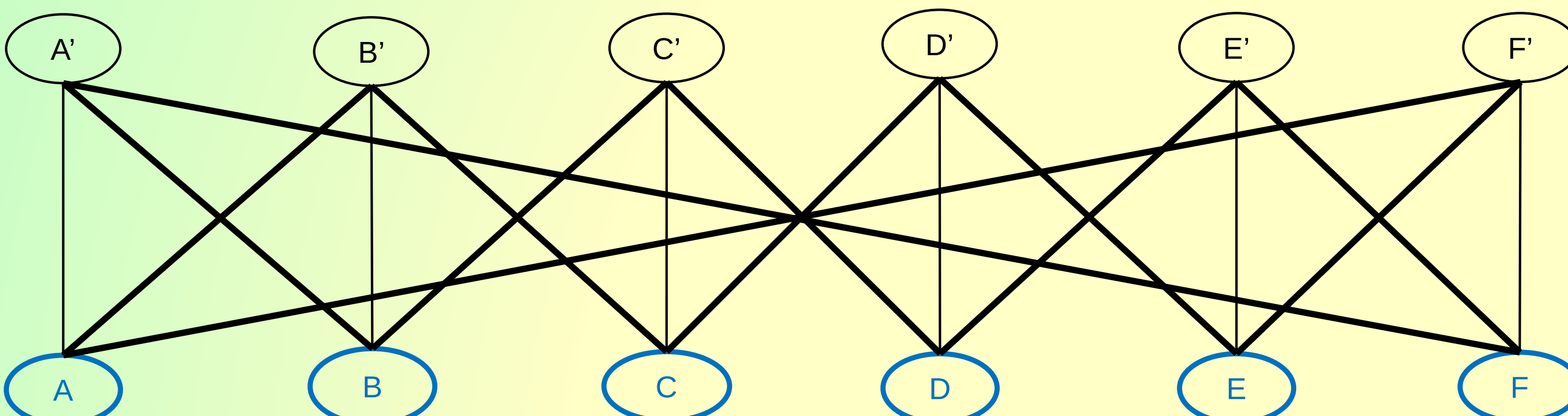
- Replace each edge (u, v) in the original graph with an edge (u’, v)



Solution 1

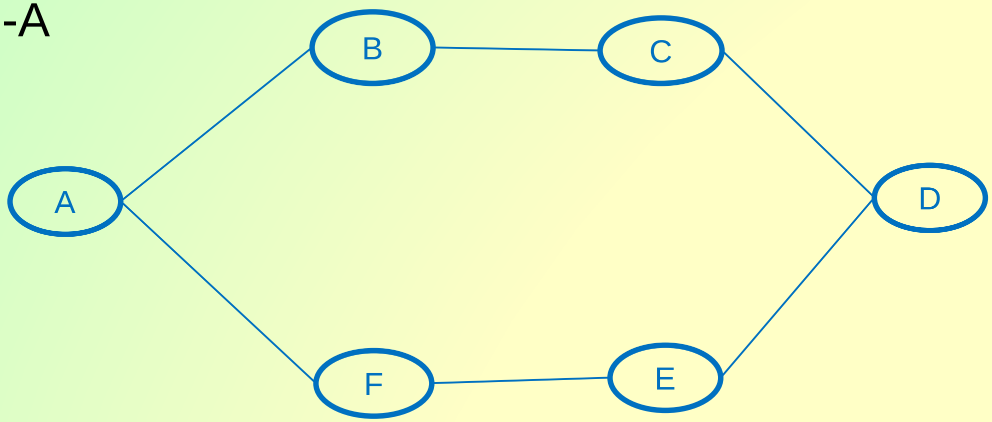
- Rearranging the graph…

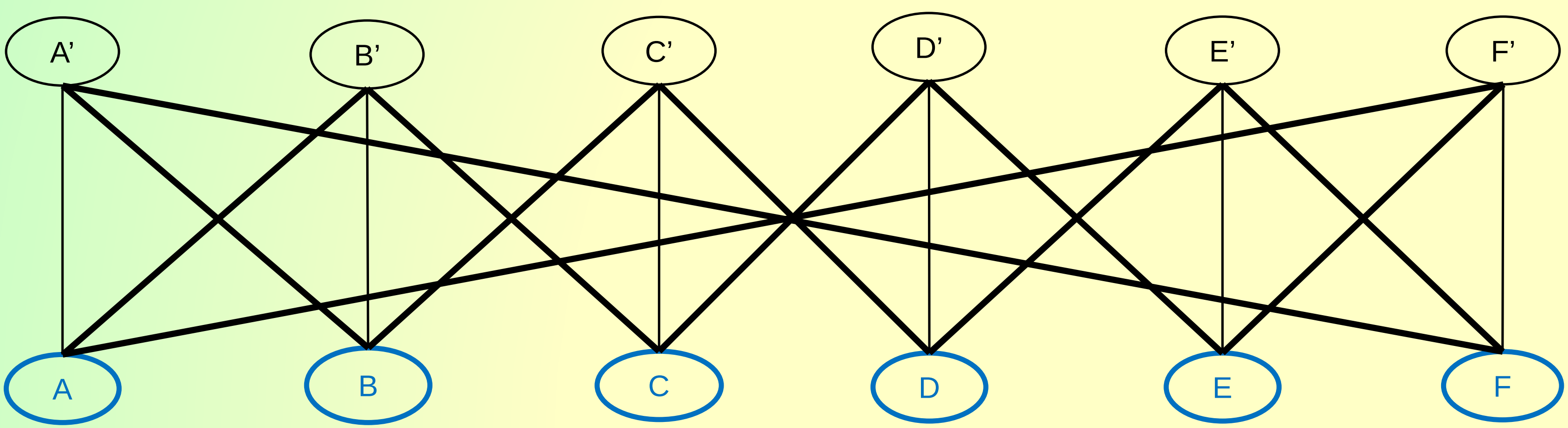




Solution 1

- In the original graph, if there is a Hamiltonian Cycle, there is also a Hamiltonian Cycle in the bipartite graph
  - For the Hamiltonian Cycle A-B-C-D-E-F-A in the original graph, the corresponding Hamiltonian Cycle in the bipartite graph is A-A'-B-B'-C-C'-D-D'-E-E'-F-F'-A

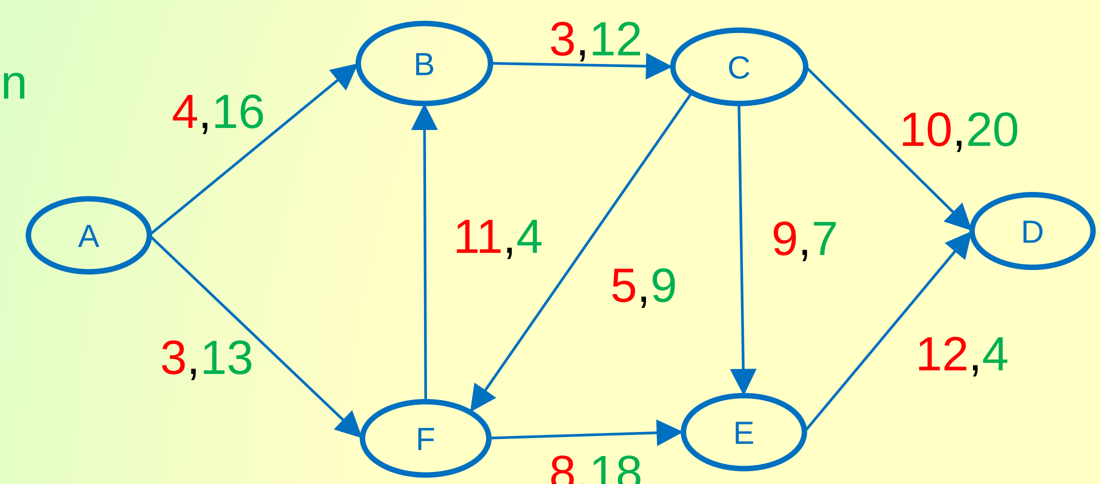




Problem 2

- In the *Min-Cost Fast Path* problem, we are given a directed graph *G=(V,E)* along with positive integer times t_e and positive costs c_e on each edge. The goal is to determine if there is a path *P* from *s* to *t* such that the total time on the path is at most *T* and the total cost is at most *C* (both *T* and *C* are parameters to the problem). Prove that this problem is **NP**-complete.

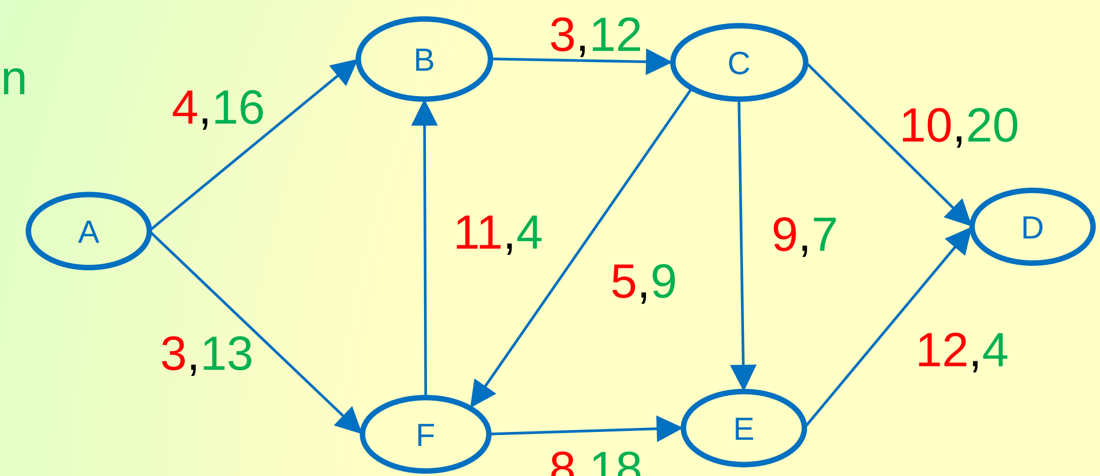
Cost is red
Time is green



Solution 2

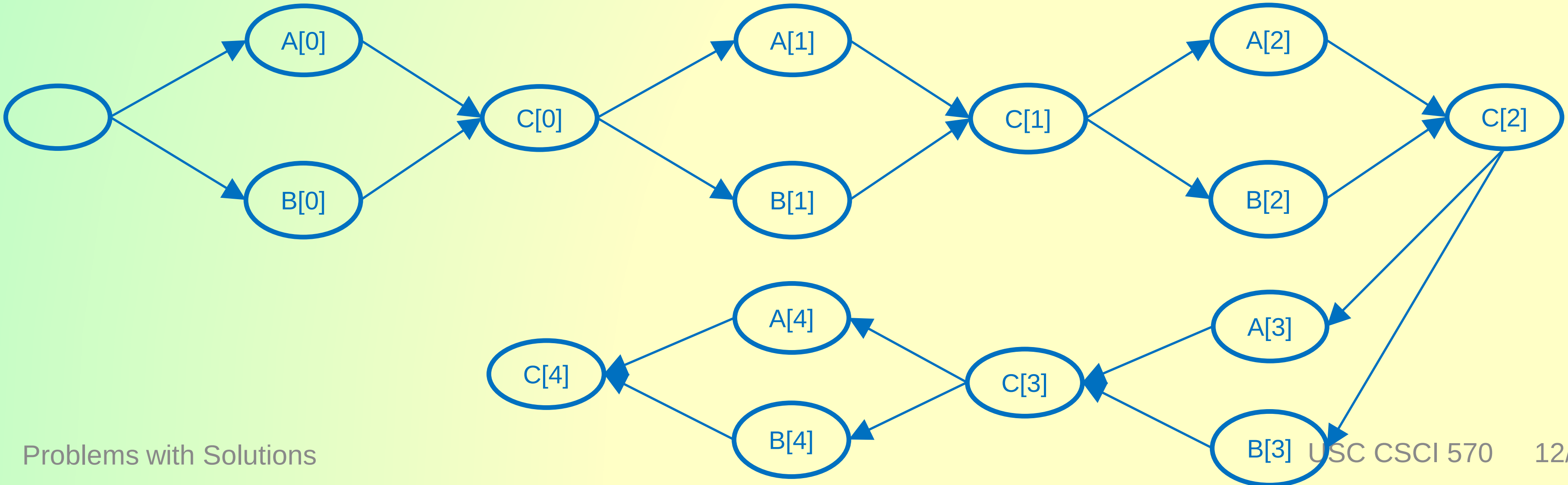
- To prove the Min-Cost Fast Path problem is in NP, we need to verify a certificate in polynomial time
- A certificate would be the path from s to t, a cost C, and a time T
- To verify this certificate, we need to make sure all of the edges in the path from s to t exist – O(n)
- We need to make sure the total cost of the path is at most C – O(n)
- We need to make sure the total time of the path is at most T – O(n)

Cost is red
Time is green



Solution 2

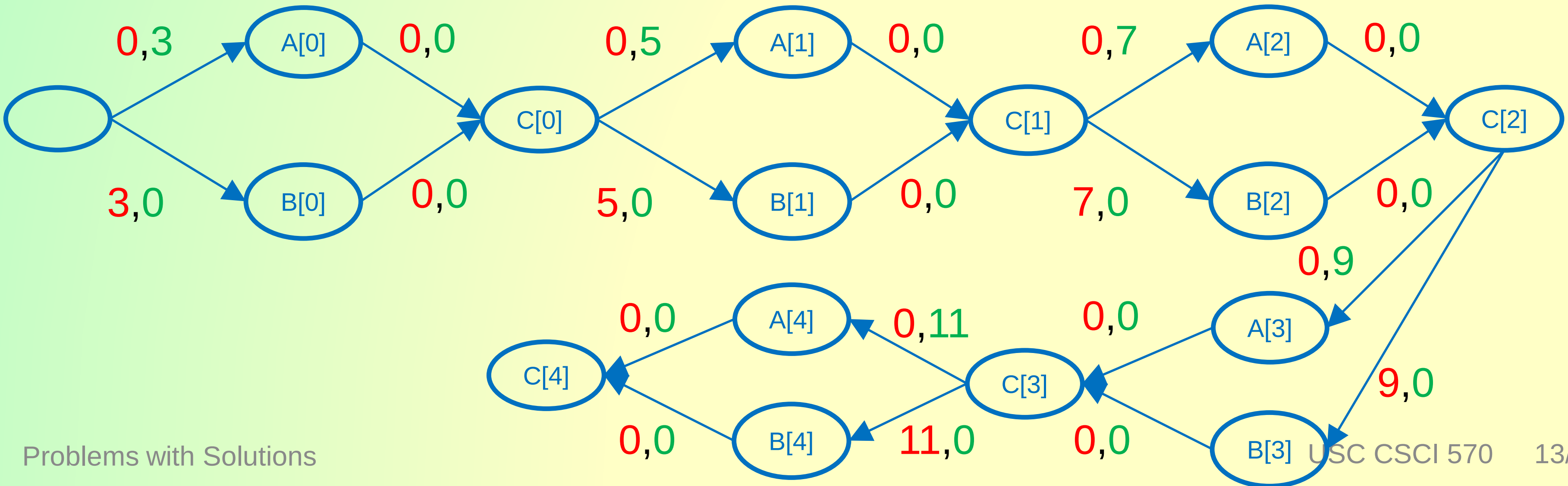
- To prove the Min-Cost Fast Path problem is in NP-Hard with respect to an NPC problem, we need to reduce an existing NPC problem to Min-Cost Fast Path in polynomial time
- Let's use subset sum
- As an example, take the set as {3, 5, 7, 9, 11} with s=21
- For each value in the set, create three nodes
  - The first node A[i] will represent when the value **is** in the solution
  - The second node B[i] will represent when the value is **not** in the solution
  - The third node C[i] will be connected from both the first node and second node
- Create a start node connecting to both A[0] and B[0]



Solution 2

- Add an edge from the start vertex for value i-1 to node A[i] with zero cost and S[i] time
- Add an edge from the start vertex for value i-1 to B[i] with S[i] cost and 0 time
- Both A[i] and B[i] connect to C[i] with zero cost and zero time
- Remember the set was {3, 5, 7, 9, 11} with s=21

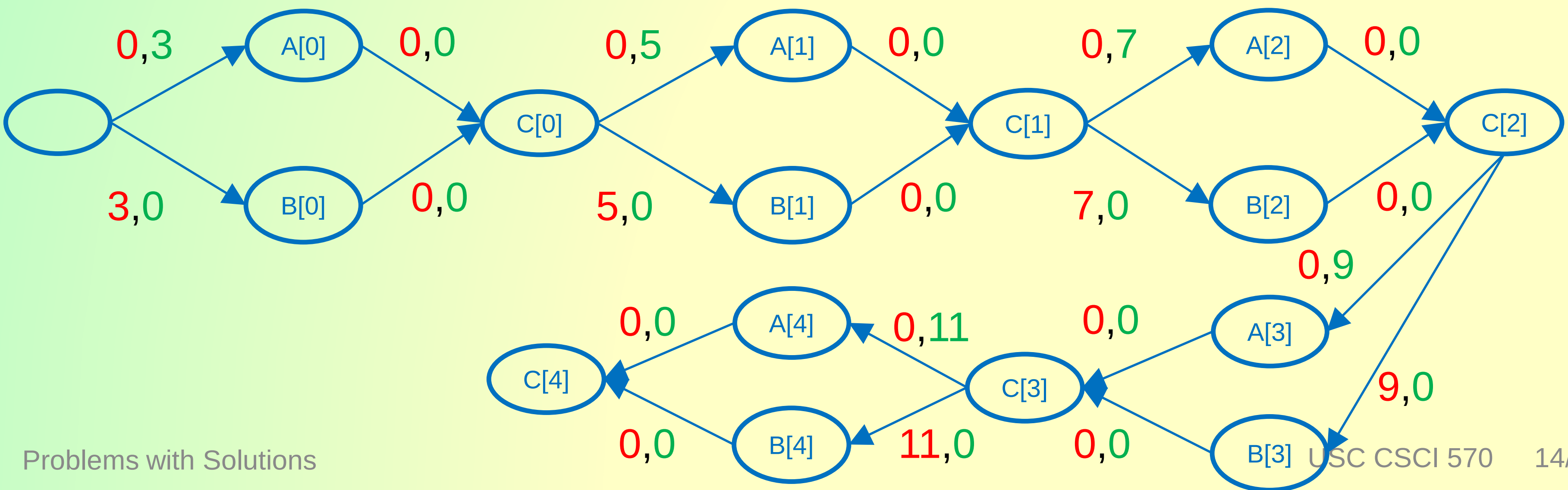
Cost is red
Time is green



Solution 2

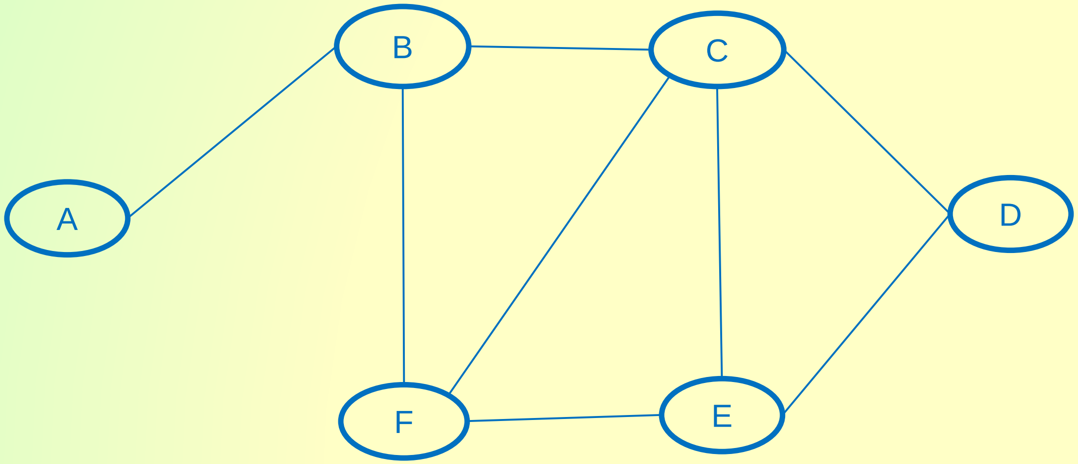
- Remember the set was {3, 5, 7, 9, 11} with s=21
- Now we ask if there's a path from start to the end with total time T equal to our target number s from subset sum, and with total cost C equal to the (sum of elements in subset sum minus target value s)
- Note that a path exists if and only if the subset sum does
  - We add time values when we visit A[i], meaning that we **are** including that value in our subset sum
  - We add cost values when we visit B[i], meaning that we are **not** including that value in our subset sum

Cost is red
Time is green



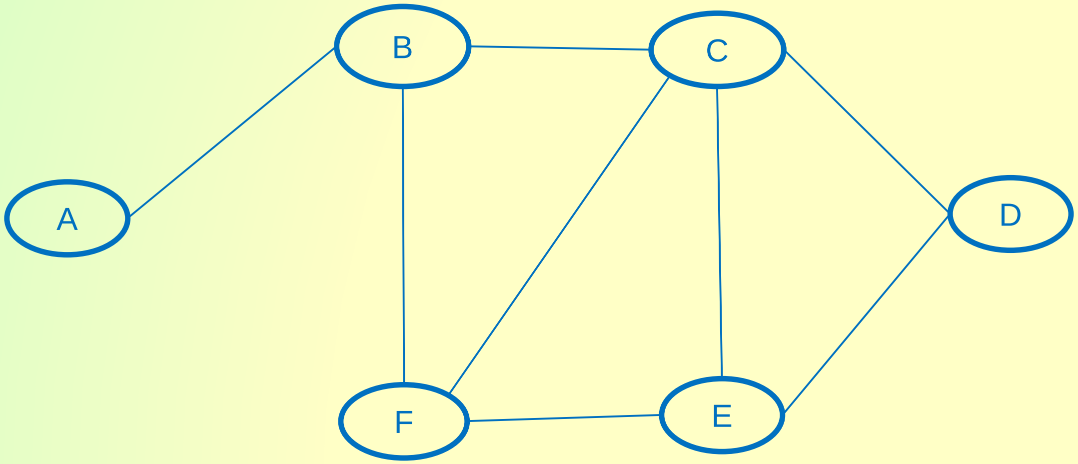
Problem 3

- We saw in lecture that finding a Hamiltonian Cycle in a graph is **NP**-complete. Show that finding a Hamiltonian Path -- a path that visits each vertex exactly once and isn't required to return to its starting point -- is also **NP**-complete.



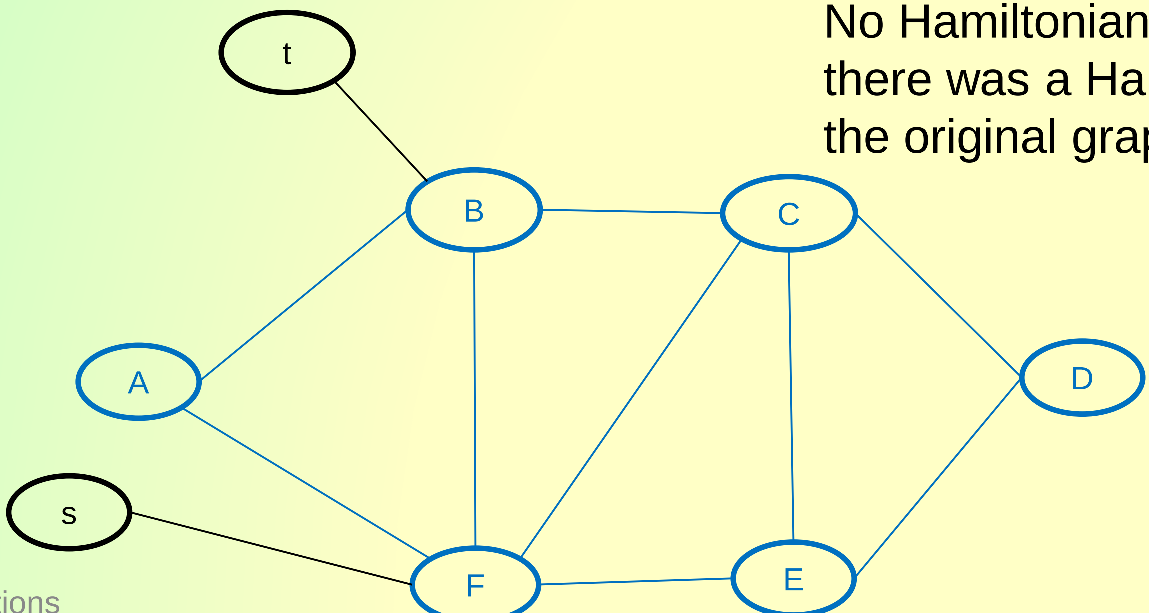
Solution 3

- To prove that Hamiltonian Path is in NP, we need to show that we can verify a certificate in polynomial time
- The certificate for a Hamiltonian Path problem is a set of nodes
- We need to check that the set contains n nodes – O(n)
- We need to check that there is an edge connecting each pair of adjacent nodes in the set – O(n)



Solution 3

- To prove that Hamiltonian Path is NP Hard with respect to an NPC problem, we need to show that we can reduce an existing NPC problem to Hamiltonian Path in polynomial time
- We will reduce Hamiltonian Cycle to Hamiltonian Path
- For an edge (u, v) in the graph
  - Create a new node s and add the edge (s, u)
  - Create a new node t and add the edge (t, v)
- If there is a Hamiltonian Path in the new graph, there was a Hamiltonian Cycle in the original graph
- If there is not a Hamiltonian Path in the new graph, we need to check another edge
  - If the edge chosen is not part of a Hamiltonian Cycle in the original graph, it will not produce a Hamiltonian Path in the new graph



No Hamiltonian Path even though there was a Hamiltonian Cycle in the original graph

Solution 3

- To prove that Hamiltonian Path is NP Hard with respect to an NPC problem, we need to show that we can reduce an existing NPC problem to Hamiltonian Path in polynomial time
- We will reduce Hamiltonian Cycle to Hamiltonian Path
- For an edge (u, v) in the graph
  - Create a new node s and add the edge (s, u)
  - Create a new node t and add the edge (t, v)
- If there is a Hamiltonian Path in the new graph, there was a Hamiltonian Cycle in the original graph
- If there is not a Hamiltonian Path in the new graph, we need to check another edge
  - If the edge chosen is not part of a Hamiltonian Cycle in the original graph, it will not produce a Hamiltonian Path in the new graph

There is a Hamiltonian Path in this graph, meaning that there was a Hamiltonian Cycle in the original graph