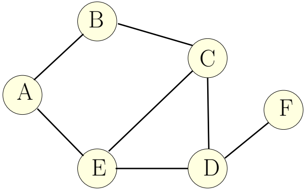
10.3.1
More examples of reductions

Maximum Independent Set in a Graph

Definition

Given undirected graph $G = (V, E)$ a subset of nodes $S \subseteq V$ is an independent set (also called a stable set) if for there are no edges between nodes in S. That is, if $u, v \in S$ then $(u, v) \notin E$.

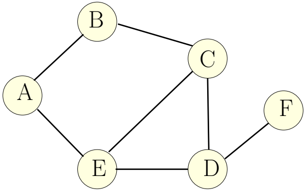


Some independent sets in graph above:

Maximum Independent Set Problem

Input Graph $G = (V, E)$

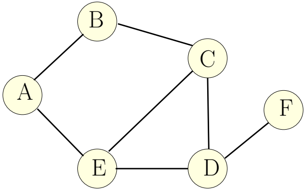
Goal Find maximum sized independent set in G



Maximum Weight Independent Set Problem

Input Graph $G = (V, E)$, weights $w(v) \geq 0$ for $v \in V$

Goal Find maximum weight independent set in G

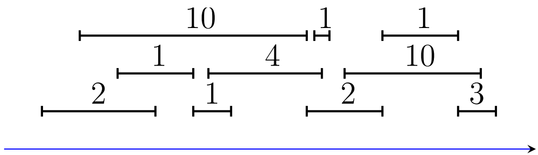


Weighted Interval Scheduling

Input A set of jobs with start times, finish times and weights (or profits).

Goal Schedule jobs so that total weight of jobs is maximized.

1. Two jobs with overlapping intervals cannot both be scheduled!

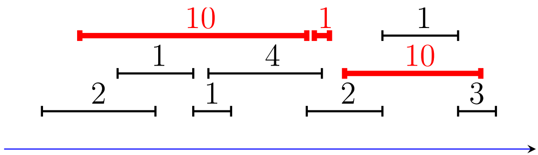


Weighted Interval Scheduling

Input A set of jobs with start times, finish times and weights (or profits).

Goal Schedule jobs so that total weight of jobs is maximized.

1. Two jobs with overlapping intervals cannot both be scheduled!



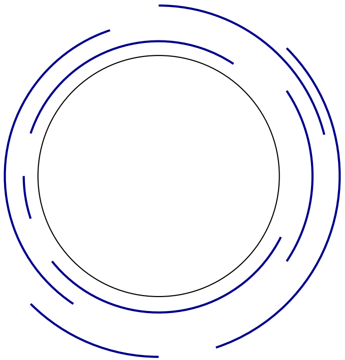
Reduction from Interval Scheduling to MIS

Question: Can you reduce Weighted Interval Scheduling to Max Weight Independent Set Problem?

Weighted Circular Arc Scheduling

Input A set of arcs on a circle, each arc has a weight (or profit).

Goal Find a maximum weight subset of arcs that do not overlap.



Reductions

Question: Can you reduce Weighted Interval Scheduling to Weighted Circular Arc Scheduling?

Question: Can you reduce Weighted Circular Arc Scheduling to Weighted Interval Scheduling? Yes!

```
MaxWeightIndependentArcs(arcs 𝒞)
    cur-max = 0
    for each arc C ∈ 𝒞 do
        Remove C and all arcs overlapping with C
        w_C = wt of opt. solution in resulting Interval problem
        w_C = w_C + wt(C)
        cur-max = max{cur-max, w_C}
    end for
    return cur-max
```

n calls to the sub-routine for interval scheduling

Reductions

Question: Can you reduce Weighted Interval Scheduling to Weighted Circular Arc Scheduling?

Question: Can you reduce Weighted Circular Arc Scheduling to Weighted Interval Scheduling? Yes!

```
MaxWeightIndependentArcs(arcs 𝒞)
    cur-max = 0
    for each arc C ∈ 𝒞 do
        Remove C and all arcs overlapping with C
        w_C = wt of opt. solution in resulting Interval problem
        w_C = w_C + wt(C)
        cur-max = max{cur-max, w_C}
    end for
    return cur-max
```

n calls to the sub-routine for interval scheduling

Reductions

Question: Can you reduce Weighted Interval Scheduling to Weighted Circular Arc Scheduling?

Question: Can you reduce Weighted Circular Arc Scheduling to Weighted Interval Scheduling? Yes!

```
MaxWeightIndependentArcs(arcs 𝒞)
    cur-max = 0
    for each arc C ∈ 𝒞 do
        Remove C and all arcs overlapping with C
        w_C = wt of opt. solution in resulting Interval problem
        w_C = w_C + wt(C)
        cur-max = max{cur-max, w_C}
    end for
    return cur-max
```

n calls to the sub-routine for interval scheduling

Reductions

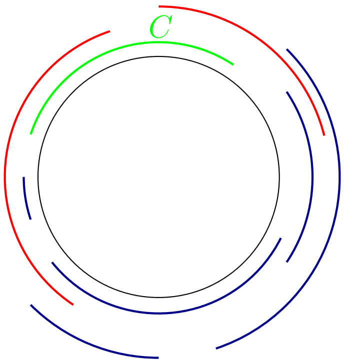
Question: Can you reduce Weighted Interval Scheduling to Weighted Circular Arc Scheduling?

Question: Can you reduce Weighted Circular Arc Scheduling to Weighted Interval Scheduling? Yes!

```
MaxWeightIndependentArcs(arcs 𝒞)
    cur-max = 0
    for each arc C ∈ 𝒞 do
        Remove C and all arcs overlapping with C
        w_C = wt of opt. solution in resulting Interval problem
        w_C = w_C + wt(C)
        cur-max = max{cur-max, w_C}
    end for
    return cur-max
```

n calls to the sub-routine for interval scheduling

Illustration



THE END

...

(for now)