RESOURCE RESERVATION PROBLEM (RRP)

Ex. 8.4 pg. 506

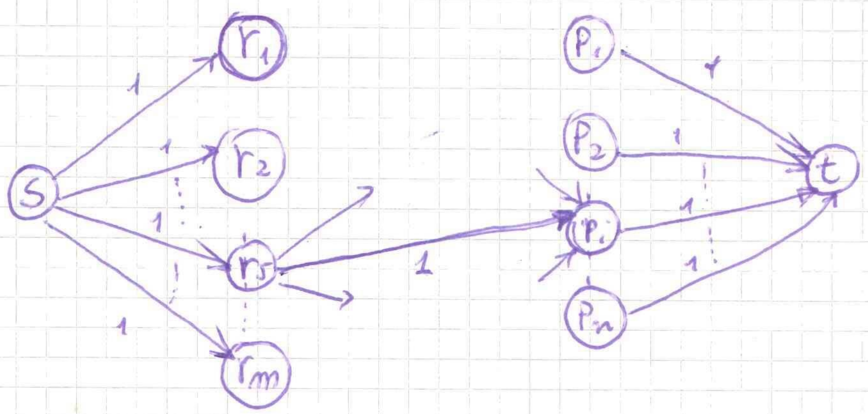
LAB/6

In a Real-Time systems there are:

- n asynchronous Processes $p_1, \dots, p_n$
- m Resources, $r_1, \dots, r_m$
- Each p_i requests a set R_i of resources
- Each r_i can be used by at-most 1 process at a time.
- If a p_i is allocated at least an $r_J \in R_i$ then p_i is ACTIVE, otherwise, it is BLOCKED. ← important assumption

Problem 1: It is possible to allocate r_J to p_i so that at-least K processes will be active?

Sol 1:



- There is a node for each r_i & p_J
- There is an edge (r_i, p_J) if $r_i \in R_J$
- Capacities are all set to 1

The problem has a solution iff $val(f_{max}) \geq K$

Problem 2 We change the definition of ACTIVE:
A process p_i is ACTIVE if it is allocated ALL resources in R_i.

SoL 2. We can model as an ~~[illegible]~~ independent set problem.
- The graph contains just processes nodes
- There is an edge (p_i, p_J) if $R_i \cap R_J \neq \emptyset$

Problem 3. Show that RRP is NP-complete.

SOL 3 →

SOL 3

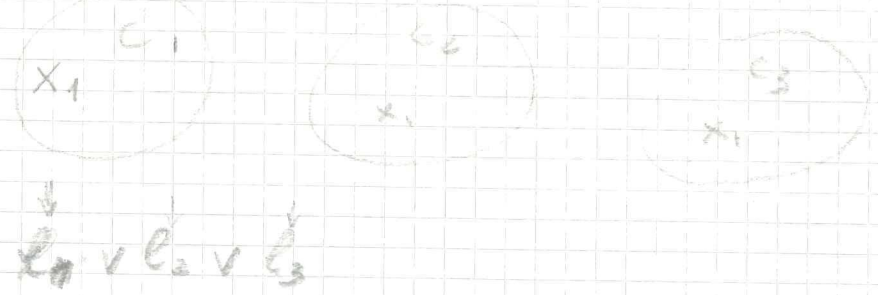
INDEPENDENT_SET $\leq_p$ RRP

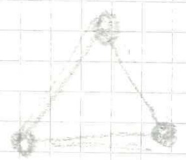
Given a graph $G = (V, E)$ and an integer K, then we generate an instance of RRP:

- For each $v \in V \rightarrow P_v$ be a process
- For each $e = (u,v) \in E \rightarrow$ resourse r_e
- ~~$R_{P_v} = \{ r_{uv} \mid (u,v) \in E \} \cup \{ r_{vv} \}$~~
- $PR_{P_v} = \{ r_e \mid$ edge e is incident to $v \}$
- K is the same as in Ind. Set.

($\Rightarrow$ Completeness) Suppose G has an I.S. of size at-least K, say S. Then, since any two nodes $u, v \in S$ are not connected, then the corresponding processes P_u & P_v do not share any resourse. Thus, there are at least K Processes that can be allocated all the requested resouses.

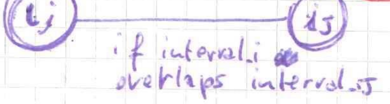
($\Leftarrow$ Soundness) Suppose RRP has a solution of size K, say S. Then, each pair of processes P_v & $P_t \in S$ do not share any resource, then the corresponding nodes in G are not connected.





EX 1. pg. 505

LAB 6



[Interval Overlapping. Given n intervals on a line, and a number K, does the collection contains a subset of non-overlapping intervals of size at least K?]

Q: Show that INT-OVERLAPPING $\leq_p$ IND-SET

We construct a graph G, such that,

- There is a node for each interval
- There is an edge (u, v) if interval-u overlaps interval-v

Th. G has an independent-set of size at least K iff there is a collection of at least K-intervals non-overlapping, i.e.,

Interval-Overlapping $\leq_p$ Independent-Set

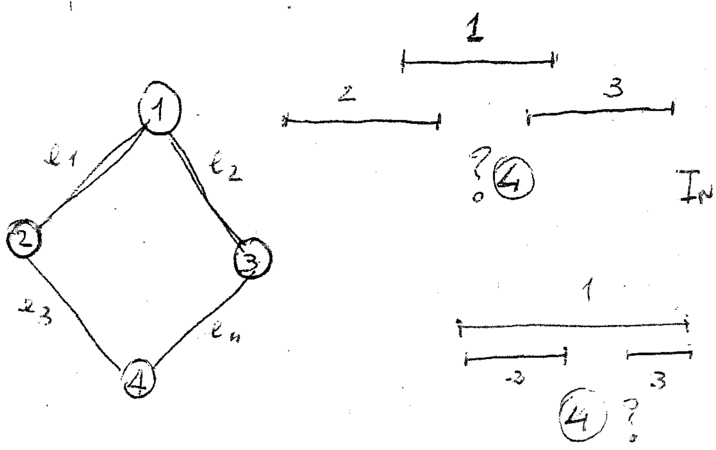
~~Show also the viceversa~~ Viceversa: IS the case that: Independent-Set $\leq_p$ Interval-Overlapping?

Answer: No, since Interval Overlapping $\in$ P-time. (See book, Section 6.1 for a P-Time algorithm)

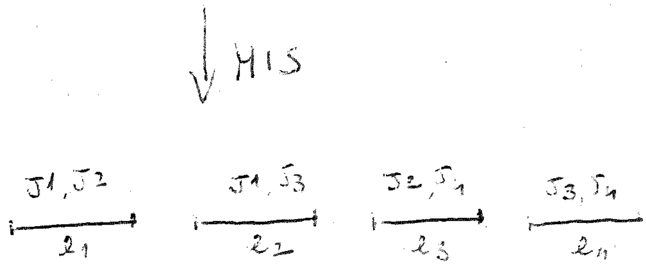
PROOF

(⇒ SOUNDNESS) Let S be an independent set and $x, y \in S$. Then, there is no edge connecting x & y. Thus, x & y are not overlapping by the way G is constructed.

(⇐ COMPLETENESS) Let I be the set of non overlapping intervals and $i, J \in I$. Then, there are nodes $i, J \in G$ such that i & J are not connected. Thus $S_I = \{ i \in V \mid i \in I \}$ is an IND-set of size K.



INTERVAL OVERLAPPING



$J_1 = \{e_1, e_2\}$ $\quad$ $J_3 = \{e_2, e_4\}$

$J_2 = \{e_1, e_3\}$ $\quad$ $J_4 = \{e_3, e_4\}$

$\{J_1, J_4\}$

$\{J_2, J_3\}$

Ex 8.14 pg. 512 <u>MULTIPLE INTERVAL SCHEDULING</u> (MIS)

- There is a single processor that run Jobs
- There are n Jobs, $J_1, \dots, J_n$
- Each Job can be modeled as the <u>set of TIME</u> Intervals that the Job requires to be executed

  $J_i = \{INT^i_1, \dots INT_{im}\}$

Q: Given a number K, are there at least K Jobs so that no two of them have any overlapping interval?

Show that the problem is NP-complete.

Hint: Show that INDEPENDENT-SET $\leq_p$ MIS

<u>SOL</u>: Given a graph $G = (V, E)$ and a number K we construct an instance of MIS int the following way:

- For each node we create a Job: $v \rightarrow J_v$
- For each edge we associate a time interval, so that, if there are m edges we generate m disjoint time intervals: $I_{e_1}, I_{e_2}, \dots, I_{em}$.
- For each Job J_v, we set, $J_v = \{I_{e_K} \mid e_K$ is incident to node $v\}$

(⇒ Completeness) Suppose there is an ind. set of size K, say S, then, Jobs J_v, J_u for $u, v \in S$, cannot share any overlapping interval since u & v cannot be connected by an edge. Thus, the MIS problem has at least K Jobs non overlapping.

(⇐ Soundness) Suppose MIS has at least K non-overlapping Jobs, say NJ, then for each pair $J_u, J_v \in NJ$ the corresponding nodes u, v in G cannot be connected. Then G has an independent set of size at-least K.

Ex 8.5

<u>HITTING SET Problem</u>

Consider a set $A = \{a_1, \ldots, a_n\}$ ~~and~~, a collection $B_1, \ldots, B_m$ s.t. $B_i \subseteq A$, and an integer number K.

Q: Is there an HITTING SET $H \subseteq A$ and $H \cap B_i \neq \emptyset$, for all $i = 1, \ldots, m$, such that $|H| \leq K$?

<u>Problem</u>. Show that HITTING SET is NP-complete.
Use the reduction VERTEX_COVER $\leq_p$ HITTING_SET

<u>Solution</u>. It's easy to see that the problem is in NP. (left as an exercise).
We map an instance $G = (V, E)$, K of Vertex Cover into an instance of HITTING SET as follows:

- $A \equiv V = \{v_1, \ldots, v_n\}$
- For each $v_i \in V$ we construct a set B_i as follows:

$$B_i = \{ v_i, v_j \mid (v_i, v_j) \in E \}$$

EX 8.10 (pg. 508) STRATEGIC ADVERTISING (SAP)

Consider a company and its web site modeled as a DIRECTED Graph $G = (V, E)$. Let $P = \{P_1, \dots, P_q\}$ the set of most visited "tours", each modeled as a directed path in G.

Q: Given G, P and a number K, it is possible to place advertisements on at most K nodes of G so that each path $P_i \in P$ includes at least one node containing an advertisement?

Show that SAP is an NP-complete problem. Use the NP-complete VERTEX-COVER.

Solution

① There is a polynomial certifier. Indeed, we can guess a set of nodes $N = \{x_1, \dots, x_k\} \subseteq V$ and for each $x_i \in N$ check in $O(n)$ (where $|V| = n$) if $x_i \in P_J$. Thus in $O(k \cdot q \cdot n)$ we can verify whether N is a solution.

② VERTEX-COVER $\leq_p$ SAP

We show how to transform an instance of VERTEX_COVER into an instance of SAP. Given a graph $G = (V, E)$ and K $\Rightarrow$ we construct a Directed Graph $DG = (V', E')$ and a set of paths P in $DG = \{P_1, \dots, P_q\}$ in the following way:

1. We fix an order in V, and $V' = \{x_1, \dots, x_n\} = V$
2. $E' = \{(x_i, x_J) \in E \mid i < J\}$
3. We set $q = m$ (where $|E'| = m$) and $P = E'$.

Ex8.19

DANGEROUS CONTAINERS (DG)

Consider "n" Containers, each containing a different Dangerous Material. A Logistic Company will ship the containers using "m" different trucks, $T=\{t_1,\dots,t_m\}$, each holding up to "K" containers.

Problem 1 (DC-1)

- For each container, c_i we Know the set of trucks that can safely transport c_i, i.e,

  $TL_i = \{t_{i_1}, \dots, t_{i_q}\} \subseteq T$

Q: Is there a way to load all "m" Containers into the "m" Trucks so tath each truck is not overloaded, and each container is loaded in a "permitted" truck? Solve in P-Time!

Problem 2 (DC-2)

- Any container can be placed in any truck BUT there are PAIRS of Containers that can not be placed in the same truck: Not-Together = $\{(c_i, c_j), \dots\}$

Q: Is there a way to load all "n" Containers into the "m" trucks so that no Truck is overloaded, and no containers are in the same Truck if they are not supposed to be?

Show that Problem 2 is NP-complete.
Use the reduction 3-COLORABILITY $\leq_P$ DC-2.

Ex 8.19 SOLUTION To Problem 2

- The problem has an efficient certifier. Fixed a Truck allocation, we can check in P-Time whether the allocation respects the No-Overload condition, and that the containers allocated to each truck, t_i, can be placed together.

- 3-Colorability $\leq_P$ DC-2
  - $G = (V, E)$
  - 3 colors

  - "m" Trucks
  - "n" Containers
  - K = max num of containers per truck
  - NOT-TOGETHER = $\{ (c_i, c_j), \ldots \}$

We transform an instance of 3-COLORABILITY into an INSTANCE of DC-2 as follows:

- $m = 3$ (number of colors)
- $n = |V|$, i.e., containers are nodes in G.
- $K = n$, i.e, each truck can load all containers.
- NOT-TOGETHER = $\{ (v_i, v_j) \mid (v_i, v_j) \in E \}$.

Ex. 8.20 LOW DIAMETER CLUSTERING (LDC)

The main purpose is to group/cluster a set of objects into CLUSTERS of "SIMILAR" objects.

Given the following input:

- A set of n objects: $P = \{p_1, p_2, \dots, p_n\}$, $n \geq 1$
- On each pair of object a <u>DISTANCE</u> is associated:
  - $d(p_i, p_j) > 0$ if $i \neq j$, otherwise $d(p_i, p_i) = 0$
  - $d(p_i, p_j) = d(p_j, p_i)$ (symmetric)
- An integer bound, B.
- A number K.

Q: Can the set of objects P be partitioned into K sets, so that no two points are at distance greater than B?
i.e., Let $S_1, \dots, S_K$ a partition of P, then

$$\forall p_i, p_j \in S_q \,.\, d(p_i, p_j) \leq B \quad, \text{ for any } q = 1, \dots, K.$$

Show that LDC is NP-complete. Use the NP-complete problem K-COLORABILITY.

<u>SOLUTION</u>

(1) There is a polynomial certifier. Indeed, we can guess a partition an check in $O(K \cdot n^2)$ if it is a solution.

(2) K-COLORABILITY $\leq_P$ LDC

We show how to transform an instance of K-COLORABILITY into an instance of LDC. Given a graph $G = (V, E)$ and K colors ⇒ we construct the following LCD instance:

1. For each node in G we generate an object: $P = \{v_1, \dots, v_n\} = V$
2. We set $B = 2$, and the distance as:

$$d(v_i, v_j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } (v_i, v_j) \notin E \\ 2 & \text{if } (v_i, v_j) \in E \end{cases}$$

Ex 11.1.2 TRUCK LOADING PROBLEM (TLP)

Consider a shipping company with the following problem:

- There are "n" containers each of weight w_i;
- There are trucks, each holding at most W units of weight.
- Many containers can be transported by each truck subject to the weight restriction W.

GOAL: Minimize the number of trucks to transport all containers.

Problem: Consider a polynomial Greedy Algorithm that load in sequense the container till the weigh limit W is respected.
Give an example of a set of weights and a value W where this algorithm does not use the minimal number of trucks.

Solution

$w_1 = 1$, $w_2 = 1$, $w_3 = 3$, $w_4 = 1$, $W = 3$ $\Rightarrow$

$T_1 : \{w_1, w_2\}$

$T_2 : \{w_3\}$

$T_3 : \{w_4\}$

3 - trucks!