NP-Completeness : Proofs

• Proof Methods

A method to show a decision problem Π NP-complete is as follows.

(1) Show $\Pi \in$ NP.

(2) Choose an NP-complete problem Π'.

(3) Show $\Pi' \propto \Pi$.

A method to show an optimization problem Ψ NP-hard is as follows.

(1) Choose an NP-hard problem Ψ' (Ψ' may be NP-complete).

(2) Show $\Psi' \propto \Psi$.

An alternative method to show Ψ NP-hard is to show the decision version of Ψ NP-complete.

- **Two Simple Examples**

Ex. **Sum of Subsets**

Instance : **A finite set A of positive integers and a positive integer c.**

Question : **Is there a subset A' of A whose elements sum to c ?**

For example, if $A = \{7, 5, 19, 1, 12, 8, 14\}$ and $c = 21$, then the answer is *yes* ($A' = \{7, 14\}$).

NP-completeness of Sum of Subsets is shown below.

♣ **Sum of Subsets $\in$ NP.**

♣ **A chosen NP-complete problem : Exact Cover.**

Exact Cover

Instance : A finite set S and k subsets $S_1, S_2, \ldots, S_k$ of S.

Question : Is there a subset of $\{S_1, S_2, \ldots, S_k\}$ that forms a partition of S ?

For example, if $S = \{7, 5, 19, 1, 12, 8, 14\}$, $k = 4$, $S_1 = \{7, 19, 12, 14\}$, $S_2 = \{7, 5, 8\}$, $S_3 = \{5, 1, 8\}$, and $S_4 = \{19, 1, 8, 14\}$, then the answer is *yes* ($\{S_1, S_3\}$ forms a partition of S).

♣ Exact Cover $\propto$ Sum of Subsets.

Let $S = \{u_1, u_2, \ldots, u_m\}$ and $S_1, S_2, \ldots, S_k$ be an arbitrary instance of Exact Cover.

An instance of Sum of Subsets can be obtained in polynomial time as follows.

$$A = \{a_1, a_2, \ldots, a_k\} \text{ and } c = \sum_{i=0}^{m-1} (k+1)^i ,$$

where for $1 \le j \le k$,

$$a_j = \sum_{i=1}^{m} e_{j,i} (k+1)^{i-1} ,$$

with $e_{j,i} = 1$ if $u_i \in S_j$ and $e_{j,i} = 0$ if $u_i \notin S_j$.

$\Rightarrow$ **Sum of Subsets has the answer *yes* if and only if Exact Cover has the answer *yes*.**

For example, given the following instance of Exact Cover :

$S = \{7, 5, 19, 1, 12, 8, 14\}$, $k = 4$,

$S_1 = \{7, 19, 12, 14\}$, $S_2 = \{7, 5, 8\}$,

$S_3 = \{5, 1, 8\}$, and $S_4 = \{19, 1, 8, 4\}$,

a matrix e is defined as follows.

$$\begin{array}{c}
 \\ S_1 \\ S_2 \\ S_3 \\ S_4
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} 7 & 5 & 19 & 1 & 12 & 8 & 14 \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1
\end{bmatrix}
\end{array}$$

An instance of Sum of Subsets is constructed as follows.

$A = \{a_1, a_2, a_3, a_4\}$ and $c = 5^0 + 5^1 + 5^2 + \ldots + 5^6$,

where

$$a_1 = 5^0 + 5^2 + 5^4 + 5^6 \qquad (1^{\text{st}} \text{ row of } e);$$

$$a_2 = 5^0 + 5^1 + 5^5 \qquad (2^{\text{nd}} \text{ row of } e);$$

$$a_3 = 5^1 + 5^3 + 5^5 \qquad (3^{\text{rd}} \text{ row of } e);$$

$$a_4 = 5^2 + 5^3 + 5^5 + 5^6 \qquad (4^{\text{th}} \text{ row of } e).$$

The construction relates a_i with S_i and c with S.

It is not difficult to see

$$a_1 + a_3 = c \iff S_1 \cup S_3 = S \text{ and } S_1 \cap S_3 = \varnothing.$$

Ex. Partition

Instance : A multiset $B = \{b_1, b_2, \ldots, b_n\}$ of positive integers.

Question : Is there a subset $B' \subseteq B$ such that

$$\sum_{b_i \in B'} b_i = \sum_{b_j \in B - B'} b_j \ ?$$

For example, when $B = \{17, 53, 9, 35, 41, 32, 35\}$, then the answer is *yes* ($B' = \{17, 53, 41\}$).

NP-completeness of Partition is shown below.

- ♣ **Partition $\in$ NP.**

- ♣ **A chosen NP-complete problem :**
  Sum of Subsets.

♣ Sum of Subsets $\propto$ Partition

Let $A = \{a_1, a_2, \ldots, a_m\}$ and c be an arbitrary instance of Sum of Subsets.

An instance of Partition can be obtained in polynomial time as follows:

$$B = A \cup \{a_{m+1}, a_{m+2}\},$$

where $a_{m+1} = c + 1$ **and** $a_{m+2} = 1 - c + \sum_{a_i \in A} a_i$.

Since $a_{m+1} + a_{m+2} = \sum_{a_i \in A} a_i + 2$, **we have**

$\{a_{m+1}, a_{m+2}\} \not\subset B'$ **and** $\{a_{m+1}, a_{m+2}\} \not\subset B - B'$.

We show below that $\sum_{a_i \in A'} a_i = c$ **if and only if**

$$a_{m+2} + \sum_{a_i \in A'} a_i = a_{m+1} + \sum_{a_i \in A - A'} a_i$$

(i.e., $B' = A' + \{a_{m+2}\}$**).**

($\Rightarrow$) **Suppose** $\sum_{a_i \in A'} a_i = c$.

$$
\begin{aligned}
a_{m+2} + \sum_{a_i \in A'} a_i &= (1 - c + \sum_{a_i \in A} a_i) + \sum_{a_i \in A'} a_i \\
&= 1 + \sum_{a_i \in A} a_i .
\end{aligned}
$$

$$
\begin{aligned}
a_{m+1} + \sum_{a_i \in A - A'} a_i &= (c + 1) + \sum_{a_i \in A - A'} a_i \\
&= 1 + \sum_{a_i \in A} a_i .
\end{aligned}
$$

($\Leftarrow$) **Suppose** $a_{m+2} + \sum_{a_i \in A'} a_i = a_{m+1} + \sum_{a_i \in A - A'} a_i$,

i.e., $B' = A' + \{a_{m+2}\}$.

Then, $(1 - c + \sum_{a_i \in A} a_i) + \sum_{a_i \in A'} a_i = (c + 1) +$

$\sum_{a_i \in A - A'} a_i$,

from which $\sum_{a_i \in A'} a_i = c$ **can be derived.**

Exercise 5. **Read Example 8-14 on page 367 of the textbook.**

(1) Give a reduction from Partition to the bin packing problem.

(2) Illustrate the reduction by an example.

(3) Verify the reduction.

Exercise 6. **Read Theorem 11.2 on page 518 of Ref. (2).**

(1) Give a reduction from Satisfiability to Clique.

(2) Illustrate the reduction by an example.

(3) Verify the reduction.

- **Three Proof Techniques**

  Restriction

  Local Replacement

  Component Design

- **Restriction**

If a problem Π contains an NP-hard problem Π' as a special case (i.e., Π' is a restricted subproblem of Π), then Π is NP-hard.

Ex. Exact Cover

Instance: A finite set S and k subsets $S_1, S_2, \ldots, S_k$ of S.

Question: Is there a subset of $\{S_1, S_2, \ldots, S_k\}$ that forms a partition of S?

Exact Cover by 3-Sets

Instance: A finite set S with $|S| = 3p$ and k 3-element subsets $S_1, S_2, \ldots, S_k$ of S.

Question: Is there a subset of $\{S_1, S_2, \ldots, S_k\}$ that forms a partition of S?

Exact Cover by 3-Sets is a special case of Exact Cover.

3-Dimensional Matching

Instance : **A set** $M \subseteq W \times X \times Y$**, where** W**,** X **and** Y **are three disjoint** q**-element subsets.**

Question : **Does** M **contain a** ***matching*****, i.e., a subset** $M' \subseteq M$ **such that** $|M'| = q$ **and no two elements of** M' **agree in any coordinate ?**

For example, if $W = \{0, 1\}$**,** $X = \{a, b\}$**,** $Y = \{+, -\}$**, and** $M = \{(0, a, +), (1, b, +), (1, b, -)\}$**, then the answer is** ***yes*** **(**$M' = \{(0, a, +), (1, b, -)\}$**).**

3-Dimensional Matching is a special case of Exact Cover by 3-Sets.

For example, the following instance of 3-Dimensional Matching :

$W = \{0, 1\}$, $X = \{a, b\}$, $Y = \{+, -\}$,

and $M = \{(0, a, +), (1, b, +), (1, b, -)\}$

can be transformed into an instance of Exact Cover by 3-Sets as follows :

$S_1 = \{0_W, a_X, +_Y\}$, $S_2 = \{1_W, b_X, +_Y\}$, $S_3 = \{1_W, b_X, -_Y\}$,

and $S = W \cup X \cup Y = \{0_W, 1_W, a_X, b_X, +_Y, -_Y\}$.

Therefore,

3-Dimensional Matching is NP-complete.

$\Rightarrow$ **Exact Cover by 3-Sets is NP-complete.**

$\Rightarrow$ **Exact Cover is NP-complete.**

Ex. Hamiltonian Cycle

Instance: An undirected graph $G = (V, E)$.

Question: Does G contain a *Hamiltonian Cycle*, i.e., an ordering $(v_1, v_2, \ldots, v_{|V|})$ of the vertices of G such that $(v_1, v_{|V|}) \in E$ and $(v_i, v_{i+1}) \in E$ for all $1 \le i < |V|$?

Directed Hamiltonian Cycle

Instance: A directed graph $G = (V, A)$, where A is a set of arcs (i.e., ordered pairs of vertices).

Question: Does G contain a directed *Hamiltonian cycle*, i.e., an ordering $(v_1, v_2, \ldots, v_{|V|})$ of the vertices of G such that $(v_1, v_{|V|}) \in A$ and $(v_i, v_{i+1}) \in A$ for all $1 \le i < |V|$?

Hamiltonian Cycle is a special case of Directed Hamiltonian Cycle (or Hamiltonian Cycle $\propto$ Directed Hamiltonian Cycle, where each $(u, v) \in E$ corresponds to two arcs $(u, v), (v, u) \in A$).

Therefore,

Hamiltonian Cycle is NP-complete.

$\Rightarrow$ **Directed Hamiltonian Cycle is NP-complete.**

Hamiltonian Path between Two Vertices

Instance : **An undirected graph $G = (V, E)$ and two distinct vertices $u, v \in V$.**

Question : **Does G contain a *Hamiltonian path* starting at u and ending at v, i.e., an ordering $(v_1, v_2, \ldots, v_{|V|})$ of the vertices of G such that $u = v_1$, $v = v_{|V|}$, and $(v_i, v_{i+1}) \in E$ for all $1 \le i < |V|$?**

Hamiltonian Cycle $\propto$ Hamiltonian Path between Two Vertices :

if the latter is polynomial time solvable, then the former is also polynomial time solvable (considering all edges $(u, v) \in V$ for the latter).

$\Rightarrow$ Hamiltonian Path between Two Vertices is NP-complete.

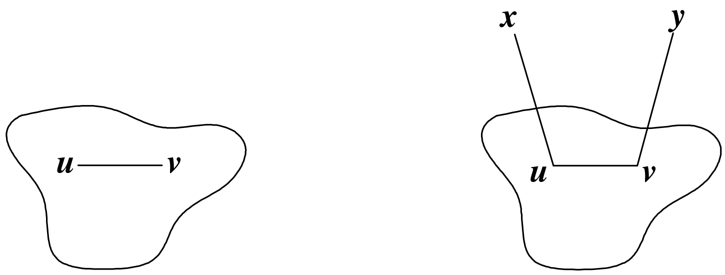
Hamiltonian Path

Instance : An undirected graph $G = (V, E)$.

Question : Does G contain a Hamiltonian path ?

Hamiltonian Cycle $\propto$ Hamiltonian Path :

For each $(u, v) \in E$, construct an instance of Hamiltonian Path by adding x, y to V and $(x, u), (y, v)$ to E (thus G' is induced).



G $\qquad\qquad\qquad\qquad$ G'

G has a Hamiltonian cycle if and only if G' has a Hamiltonian x-y path.

$\Rightarrow$ If Hamiltonian Path is polynomial time solvable, then Hamiltonian Cycle is also polynomial time solvable.

$\Rightarrow$ Hamiltonian Path is NP-complete.

Exercise 7. Show the following two problems NP-complete by restriction to Hamiltonian Path and Partition, respectively.

Bounded Degree Spanning Tree

Instance: An undirected graph $G=(V, E)$ and a positive integer $k \le |V|-1$.

Question: Does G contains a spanning tree in which each node has degree at most k?

0/1 Knapsack

Instance: A finite set U, a "size" $s(u) \in Z^+$ and a "value" $v(u) \in Z^+$ for each $u \in U$, a size constraint $b \in Z^+$, and a value goal $k \in Z^+$.

Question: Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \le b$ and $\sum_{u \in U'} v(u) \ge k$?

• Local Replacement

In order to show $\Pi' \propto \Pi$, local replacement specifies the "basic units" for Π' and replaces them with others, while constructing a corresponding instance of Π.

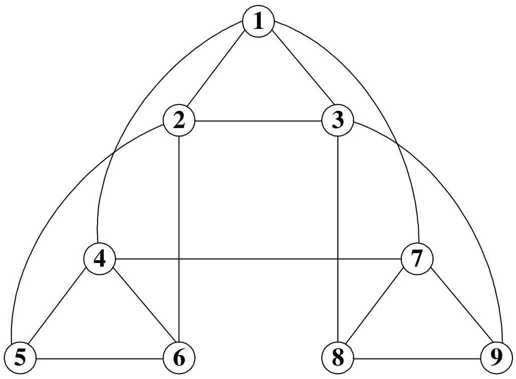
Usually, local replacement has one kind of basic units that each are replaced with the same structure.

Ex. Partition into Triangles

Instance : An undirected graph $G = (V, E)$ with $|V| = 3p$ for some integer $p > 0$.

Question : Is there a partition of V into 3-vertex subsets $V_1, V_2, \ldots, V_p$, such that each subgraph induced by some V_i $(1 \le i \le p)$ forms a triangle ?

For example, the answer for the following instance is *yes*, because V can be partitioned into {1, 2, 3}, {4, 5, 6}, {7, 8, 9} or {1, 4, 7}, {2, 5, 6}, {3, 8, 9}.

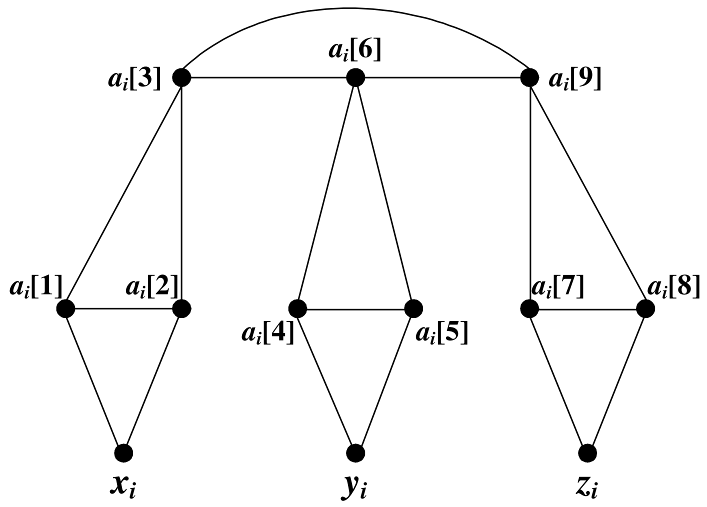


Exact Cover by 3-Sets $\propto$ Partition into Triangles is shown below.

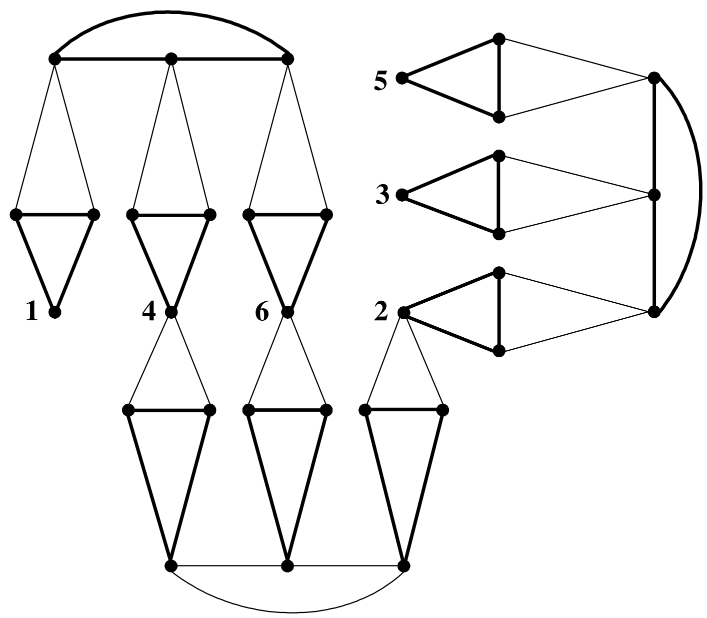
Let a set S, where $|S| = 3p$, and a collection C of 3-element subsets of S denote an arbitrary instance of Exact Cover by 3-Sets.

Construct an instance of Partition into Triangles as follows.

Consider each subset $\{x_i, y_i, z_i\} \in C$ a basic unit, and replace it with the following structure.



For example, if $S = \{1, 2, 3, 4, 5, 6\}$ and $C = \{\{1, 4, 6\}, \{2, 4, 6\}, \{2, 3, 5\}\}$, then an instance of Partition into Triangles is obtained as follows.



It is not difficult to check that if Exact Cover by 3-Sets has an answer *yes* (e.g., {{1, 4, 6}, {2, 3, 5}} is a partition of *S*), then Partition into Triangles has an answer *yes* (the triangles are shown with bold edges).

Also, if Partition into Triangles has an answer *yes*, then Exact Cover by 3-Sets has an answer *yes*.

Exercise 8. Read Example 8-9 on page 353 of the textbook.

(1) Give a reduction from Satisfiability to 3-Satisfiability.

(2) Illustrate the reduction by an example.

(3) Verify the reduction.

Sometimes, additional structures are required, while using the technique of local replacement.

Ex. Sequencing within Intervals

Instance : A finite set T of "tasks" and for each $t \in T$, a "release time" $r(t) \in Z^+ \cup \{0\}$, a "deadline" $d(t) \in Z^+$, and a "length" $l(t) \in Z^+$.

Question : Does there exist a *feasible schedule* for T, i.e., a function $f : T \rightarrow Z^+$ such that for each $t \in T$, $f(t) \geq r(t)$, $f(t) + l(t) \leq d(t)$, and $f(t') + l(t') \leq f(t)$ or $f(t) + l(t) \leq f(t')$ for each $t' \in T - \{t\}$?

(It means that the task t, which is "executed" from time $f(t)$ to $f(t) + l(t)$, cannot start execution until time $r(t)$, must be completed by time $d(t)$, and its execution cannot overlap the execution of any other task t'.)

Partition $\propto$ Sequencing within Intervals is shown below.

An arbitrary instance of Partition:

a multiset $B = \{b_1, b_2, \ldots, b_n\}$ of positive integers.

Consider each b_i $(1 \le i \le n)$ a basic unit, and let $m = \sum_{1 \le i \le n} b_i$.

Construct an instance of Sequencing within Intervals as follow:

each b_i corresponds to a task t_i with $r(t_i) = 0$, $d(t_i) = m + 1$, and $l(t_i) = b_i$.

An additional structure:

a task $\tilde{t}$ with $r(\tilde{t}) = \lceil m/2 \rceil$, $d(\tilde{t}) = \lceil (m+1)/2 \rceil$, and $l(\tilde{t}) = 1$.

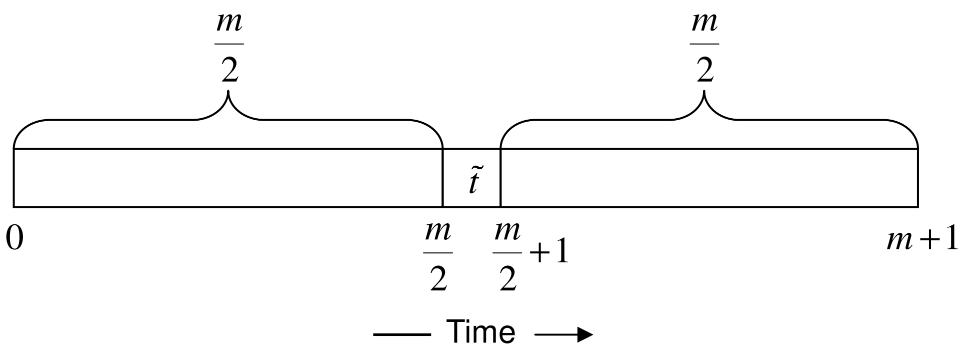
$\Rightarrow$ m should be even

(for otherwise, $r(\tilde{t}) = d(\tilde{t})$, i.e.,

it is impossible to schedule $\tilde{t}$)

$\Rightarrow \quad \boldsymbol{r(\tilde{t}) = m/2, \;\; d(\tilde{t}) = (m/2) + 1}$

$\Rightarrow \quad \boldsymbol{f(\tilde{t})}$ **must be** $\boldsymbol{m/2}$**.**



$\Rightarrow$ **Partition has the answer *yes* if and only if Sequencing within Intervals has the answer *yes*.**

- **Component Design**

While showing $\Pi' \propto \Pi$, component design is similar to local replacement in replacing the structures (i.e., basic units) of Π' with other structures, in order to obtain an instance of Π.

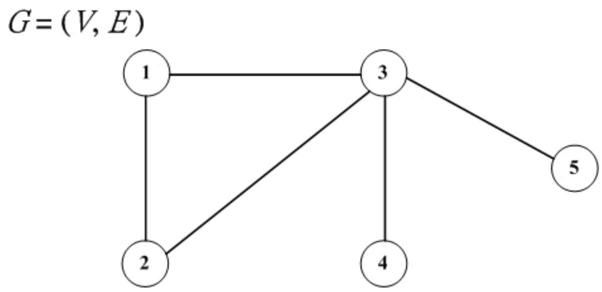
Usually, component design adopts multiple kinds of basic units, and different basic units are replaced with different structures.

Ex. Vertex Cover

Instance : An undirected graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question : Does G contain a *vertex cover* of size at most k, i.e., a subset $V' \subseteq V$ such that $|V'| \leq k$ and for each $(u, v) \in E$, at least one of u and v belongs to V' ?

For example, {1, 3}, {1, 2, 3} and {1, 2, 4, 5} are three vertex covers of the following graph. If $k \geq 2$, the answer is *yes*. If $k = 1$, the answer is *no*.



We show below 3-Satisfiability $\propto$ Vertex Cover.

3-Satisfiability

Instance: **A set U of variables and a collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses over U, where each clause of C contains three literals.**

Question: **Is there a satisfying truth assignment for C?**

For example, when $U = \{x_1, x_2, x_3\}$ and $C =$ $\{x_1 \vee x_2 \vee x_3,\ \overline{x}_1 \vee x_2 \vee \overline{x}_3,\ x_1 \vee \overline{x}_2 \vee x_3\}$, the answer is *yes*, because the assignment of U: $x_1 \leftarrow F$, $x_2 \leftarrow F$, and $x_3 \leftarrow T$, can satisfy C (i.e., $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3) = T$).

Let $U = \{u_1, u_2, \ldots, u_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ be an arbitrary instance of 3-Satisfiability.

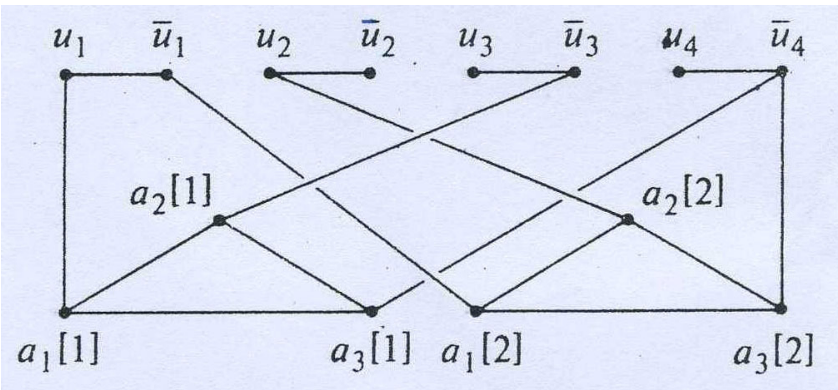
- **For each $u_i \in U$, construct a component $T_i = (V_i, E_i)$, where $V_i = \{u_i, \overline{u}_i\}$ and $E_i = \{(u_i, \overline{u}_i)\}$.**
- **For each $c_j \in C$, construct a component $S_j = (V'_j, E'_j)$, where $V'_j = \{a_1[j], a_2[j], a_3[j]\}$ and $E'_j = \{(a_1[j], a_2[j]), (a_1[j], a_3[j]), (a_2[j], a_3[j])\}$.**
- **For each $c_j \in C$, construct an edge set $E''_j = \{(a_1[j], x_j), (a_2[j], y_j), (a_3[j], z_j)\}$, where x_j, y_j and z_j are the three literals in c_j.**

An instance of Vertex Cover can be constructed as $G=(V, E)$ and $k=n+2m$, where

$$V = (\bigcup_{i=1}^{n} V_i) \cup (\bigcup_{j=1}^{m} V'_j) \text{ and}$$

$$E = (\bigcup_{i=1}^{n} E_i) \cup (\bigcup_{j=1}^{m} E'_j) \cup (\bigcup_{j=1}^{m} E''_j).$$

For example, if $U=\{u_1, u_2, u_3, u_4\}$ and $C= \{u_1 \vee \overline{u}_3 \vee \overline{u}_4,\ \overline{u}_1 \vee u_2 \vee \overline{u}_4\}$, then the following instance of Vertex Cover is constructed, where $k=8$.



- **Each edge in E''_j represents a satisfying truth assignment for c_j.**

  For example, $(u_1, a_1[1]) \in E''_1$ implies that $u_1 \leftarrow T$ can satisfy c_1.

- **Any vertex cover $V' \subseteq V$ of G contains at least one from $\{u_i, \overline{u}_i\}$ and at least two from $\{a_1[j], a_2[j], a_3[j]\}$.**

  $\Rightarrow \quad |V'| \geq n + 2m = k$

As explained below, C is satisfiable if and only if G has a vertex cover $V' \subseteq V$ with $|V'| \leq k$.

♣ **C is satisfiable $\Rightarrow$ $V' \subseteq V$ with $|V'| \leq k$ exists**

Consider the example above, where $u_1 \leftarrow T$, $u_2 \leftarrow T$, $\overline{u}_3 \leftarrow T$, and $\overline{u}_4 \leftarrow T$ can satisfy C.

$\Rightarrow$ **include $u_1, u_2, \overline{u}_3, \overline{u}_4$ in V'**

In order to make V' a vertex cover, V' must be augmented with two vertices from each set $\{a_1[j], a_2[j], a_3[j]\}$, while covering all edges in E''_j.

$\Rightarrow$ **augment V' with any two from $\{a_1[1], a_2[1], a_3[1]\}$ and $a_1[2], a_2[2]$ (or $a_1[2], a_3[2]$) from $\{a_1[2], a_2[2], a_3[2]\}$**

($a_1[2]$ must be included in V', in order to cover the edge $(\overline{u}_1, a_1[2])$)

♣ $V' \subseteq V$ with $|V'| \leq k$ exists $\Rightarrow$ C is satisfiable

V' contains exactly $k = n + 2m$ vertices: one for each $\{u_i, \overline{u}_i\}$ and two for each $\{a_1[j], a_2[j], a_3[j]\}$.

Consider the example above, where $k = 8$ and $V' = \{\overline{u}_1, u_2, \overline{u}_3, u_4, a_1[1], a_3[1], a_1[2], a_3[2]\}$ is a vertex cover.

$\Rightarrow$ $\overline{u}_1 \leftarrow T$, $u_2 \leftarrow T$, $\overline{u}_3 \leftarrow T$ and $u_4 \leftarrow T$ can satisfy C ($\overline{u}_1, u_2, \overline{u}_3, u_4 \in V'$)

Since two (e.g., $a_1[1]$ and $a_3[1]$) from $\{a_1[j], a_2[j], a_3[j]\}$ are included in V', the other (e.g., $a_2[1]$) must be connected to u_i or $\overline{u}_i$ (e.g., $\overline{u}_3$) that is included in V'.

$\Rightarrow$ each c_j is satisfiable.

Ex. Minimum Tardiness Sequencing

Instance: A finite set T of "tasks", where each $t \in T$ has "length" 1 and "deadline" $d(t) \in Z^{+}$, a partial order $\prec$ on T, and a non-negative integer $r \leq |T|$.

Question: Is there a "schedule" $f: T \rightarrow \{0, 1, \ldots, |T| - 1\}$ such that $f(t) \neq f(t')$ if $t \neq t'$, $f(t) < f(t')$ if $t \prec t'$, and $|\{t \in T : f(t) + 1 > d(t)\}| \leq r$?

A task $t \in T$ is *tardy*, if $f(t) + 1 > d(t)$.

The schedule f is required not to cause more than r tasks tardy.

We show below Clique $\propto$ Minimum Tardiness Sequencing.

Clique

Instance : An undirected graph $G=(V,E)$ and a positive integer $k \le |V|$.

Question : Does there exist a subset $V' \subseteq V$ such that $|V'| \ge k$ and every two vertices of V' are adjacent in G?

Let $G=(V,E)$ and $k \le |V|$ be an arbitrary instance of Clique.

An instance of Minimum Tardiness Sequencing can be constructed as follows.

$T = V \cup E$;

$r = |E| - k(k-1)/2$;

$v \prec e \iff v \in V$, $e \in E$, and v is an endpoint of e;

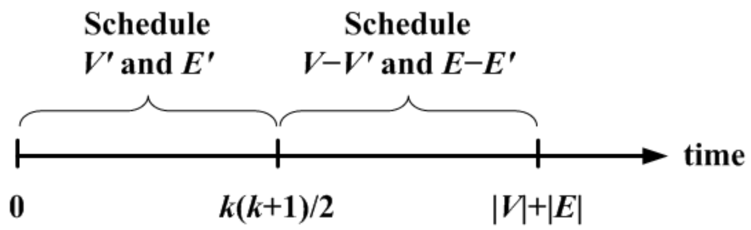
$d(v) = |V| + |E|$ for $v \in V$, and

$d(e) = k(k+1)/2$ for $e \in E$.

♣ a clique of size $\geq k$ for G $\Rightarrow$ a feasible schedule for T

Suppose that $G' = (V', E')$ is a k-vertex complete subgraph of G ($|V'| = k$ and $|E'| = k(k-1)/2$).

A feasible schedule is shown below.



Tasks in V and in E' are not tardy.

$\Rightarrow$ There are at most $|E - E'| = |E| - k(k-1)/2$ tardy tasks.

♣ a feasible schedule for T $\Rightarrow$ a clique of size $\geq k$ for G

Suppose that f is a feasible schedule, and there are x tasks from V and y tasks from E scheduled in $\{0, 1, \ldots, (k(k+1)/2)-1\}$ under f.

Then,

$$x + y = k(k+1)/2. \qquad (1)$$

Since only tasks in E may be tardy, we have

$$|E| - y \quad \leq \quad |E| - k(k-1)/2 \quad (= r).$$

$$\Rightarrow \quad y \geq k(k-1)/2 \qquad (2)$$

With (1) and (2), we have

$$x \leq (k(k+1)/2) - (k(k-1)/2) = k. \qquad (3)$$

The only situation that both (2) and (3) hold with the restriction of $\prec$ is when

$$x = k, \; y = k(k-1)/2, \text{ and}$$

the k vertices together with the $k(k-1)/2$ edges form a complete subgraph of G.

Exercise 9. **Read Example 8-10 on page 359 of the textbook.**

(1) Give a reduction from a satisfiability problem where each clause has at most three literals to the chromatic number problem.

(2) Illustrate the reduction by an example.

(3) Verify the reduction.

Exercise 10. **Read Theorem 3.5 on page 60 of Ref. (1).**

(1) Give a reduction from 3-Dimensional Matching to Partition.

(2) Illustrate the reduction by an example.

(3) Verify the reduction.

- **A Proof Technique for NP-Completeness of Subproblems**

Suppose that Π is an NP-complete problem and Π' is a restricted subproblem of Π.

A proof technique, which is based on local replacement, for the NP-completeness of Π' is introduced.

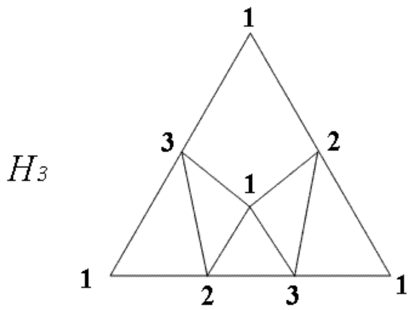
Ex. Graph 3-Colorability

Instance : An undirected graph $G=(V, E)$.

Question : Is G 3-colorable, i.e., does there exist a function $f: V \rightarrow \{1, 2, 3\}$ such that $f(u) \neq f(v)$ for all edges $(u, v) \in E$?

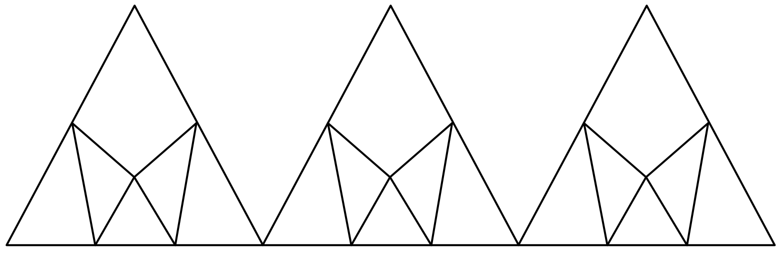
Graph 3-Colorability with Degrees at Most Four is a restricted subproblem of Graph 3-Colorability where each vertex degree of G is at most four.

For example, the following graph, denoted by H_3, is 3-colorable, and in each 3-coloring, the three endpoints of the largest triangle are assigned with the same color.



Let H_k be the concatenation of $k-2$ H_3's, where $k \geq 3$.

For example, H_5 is depicted as follows.



H_k has k "outlets" (i.e., the vertices of degree 2).

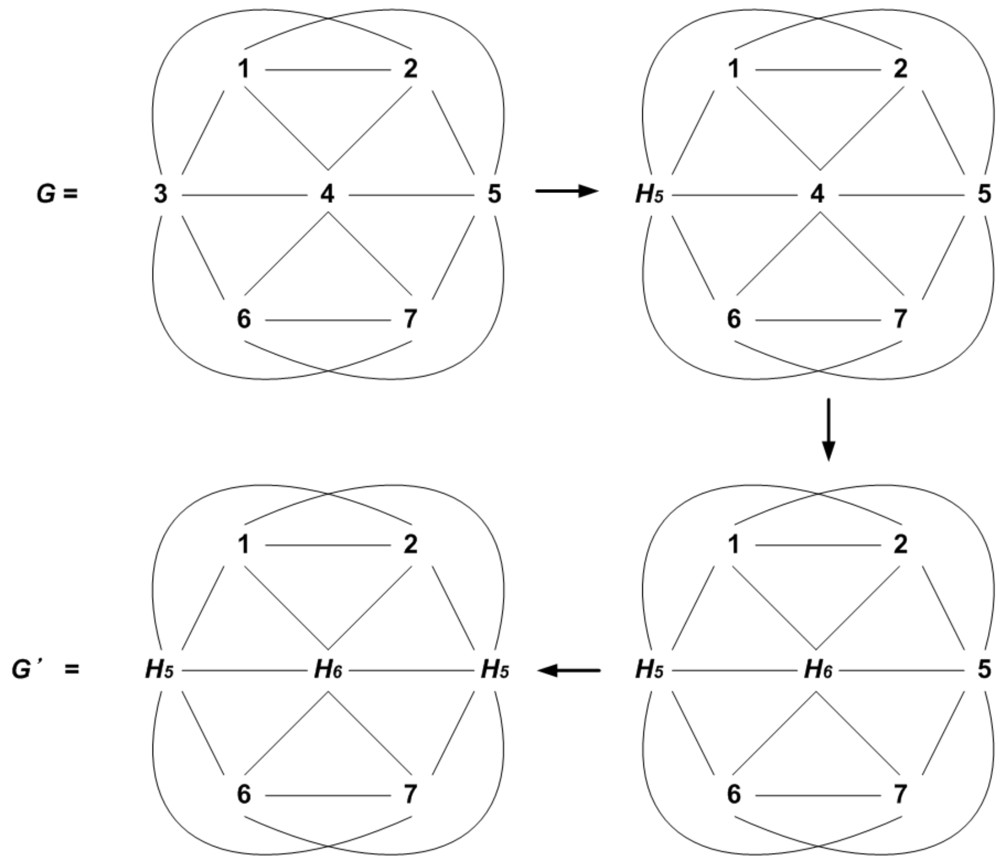
H_k is 3-colorable and in each 3-coloring, the k "outlets" are assigned with the same color.

Next we show Graph 3-Colorability $\propto$ Graph 3-Colorability with Degrees at Most Four.

Suppose that $G = (V, E)$ is an arbitrary instance of Graph 3-Colorability.

An instance $G' = (V', E')$ of Graph 3-Colorability with Degrees at Most Four can be obtained by sequentially replacing each vertex of G whose degree is $k > 4$ with H_k.

For example,

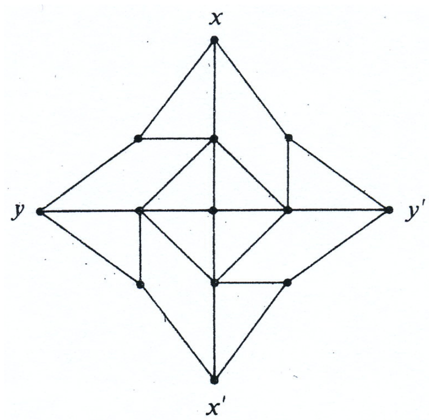


It is easy to see that *G* is 3-colorable if and only if *G'* is 3-colorable.

Planar Graph 3-Colorability is a restricted subproblem of Graph 3-Colorability where *G* is planar.

We show Graph 3-Colorability $\propto$ Planar Graph 3-Colorability below.

Let *H* denote the following graph.

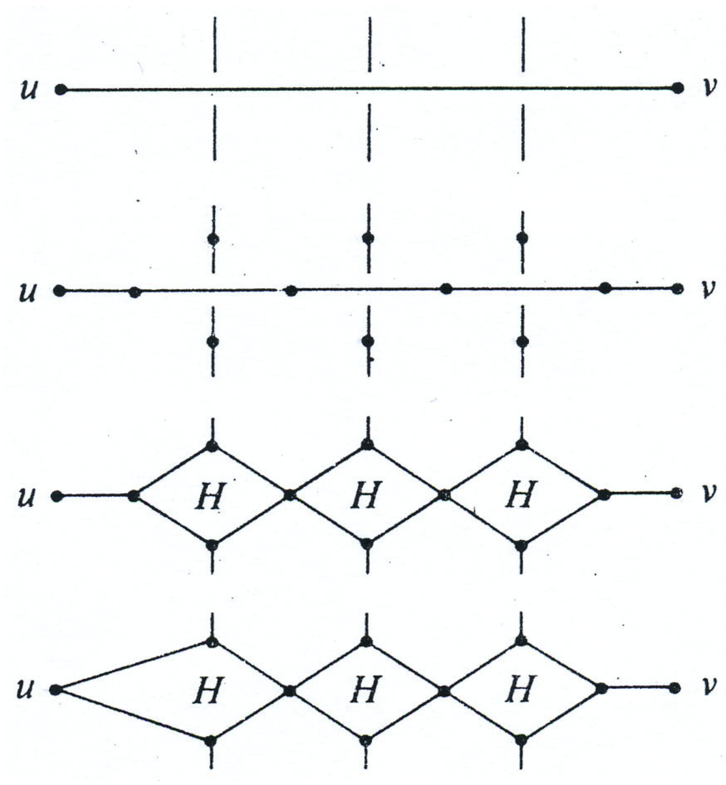


Notice that *H* is 3-colorable, and any 3-coloring *f* of *H* has $f(x)=f(x')$ and $f(y)=f(y')$.

Besides, there exist 3-colorings f_1 and f_2 of *H* with $f_1(x)=f_1(x')=f_1(y)=f_1(y')$ and $f_2(x)=f_2(x')\neq f_2(y)=f_2(y')$.

Suppose that $G = (V, E)$ is an arbitrary instance of Graph 3-Colorability.

An instance $G' = (V', E')$ of Planar Graph 3-Colorability can be obtained by performing the following replacement on the edge crossings of each edge $(u, v) \in E$.



It is not difficult to check that G is 3-colorable if and only if G' is 3-colorable.

- **More Examples**

Ex. VLSI Discrete Layout

Instance : A set $R = \{r_1, r_2, \ldots, r_n\}$ of rectangles, where each r_i is of size $h_i \times w_i$, and an integer $A > 0$.

Question : Is there a placement of R on the plane satisfying the following conditions:

(1) each vertex of r_i has an integral (x, y)-coordinate;

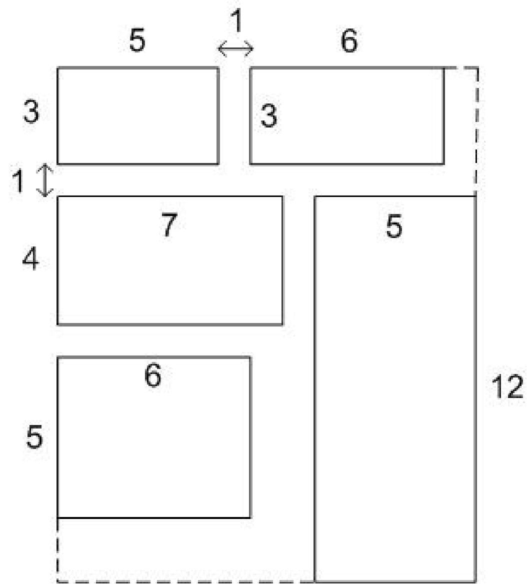
(2) each line of r_i is parallel to the x-axis or y-axis;

(3) no two rectangles overlap;

(4) every two neighboring rectangles are one distant from each other;

(5) R can be covered by a rectangle of area at most A ?

For example, if $n = 5$, r_1: 3×5, r_2: 5×12, r_3: 5×6, r_4: 3×6, r_5: 4×7, and $A = 210$, then the answer is affirmative, because the five rectangles can be covered by a rectangle of size 16×13.



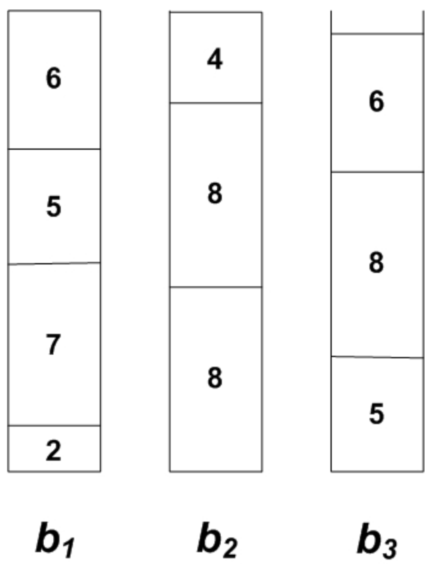
We show below Bin Packing $\propto$ VLSI Discrete Layout.

Bin Packing

Instance : A set $U = \{u_1, u_2, \ldots, u_n\}$ of items, where each u_i has size $s_i > 0$, and a set $B = \{b_1, b_2, \ldots, b_m\}$ of bins, where each b_j has capacity $c > 0$.

Question : Is there a distribution of U over B such that the items within the same bin has total size at most c ?

For example, if $n = 10$, $(c_1, c_2, \ldots, c_{10}) = (2, 7, 5, 8, 6, 8, 5, 4, 8, 6)$, $m = 3$, and $c = 20$, then the answer is affirmative.



Let $U = \{u_1, u_2, \ldots, u_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$ be an arbitrary instance of Bin Packing.

An instance of VLSI Discrete Layout can be constructed as follows.

For each u_i $(1 \le i \le n)$, construct r_i of size $1 \times ((2m+1)s_i - 1)$.

Construct r_{n+1} of size $h \times w$, where $h = 2mw + 1$ and $w = (2m+1)c - 1$.

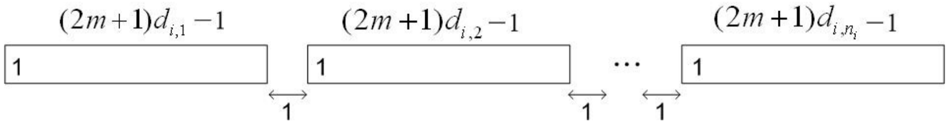
Set $A = (h + 2m)w$.

- ♣ **Bin Packing "yes" $\Rightarrow$ VLSI Discrete Layout "yes"**

Suppose that there are n_i items stored in b_i whose sizes are $d_{i,1}$, $d_{i,2}$, …, d_{i,n_i},

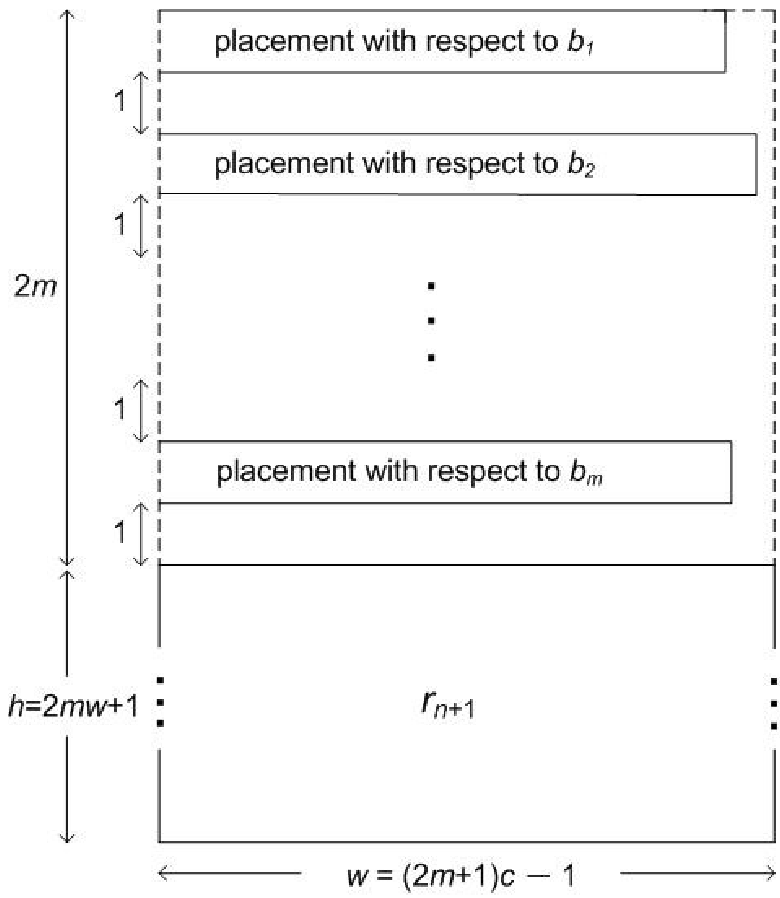
where $1 \le i \le m$ and $\sum_{r=1}^{n_i} d_{i,r} \le c$.

The placement of the corresponding rectangles with respect to b_i is as follows.



(placement with respect to b_i)

The placement of all $n+1$ rectangles is as follows.



The width of the placement with respect to b_i is computed as follows.

$$
\begin{aligned}
& (n_i - 1) + \sum_{r=1}^{n_i} ((2m+1)d_{i,r} - 1) \\
= \quad & (n_i - 1) - n_i + (2m+1)\sum_{r=1}^{n_i} d_{i,r} \\
\le \quad & (2m+1)c - 1 \\
= \quad & w.
\end{aligned}
$$

The area of the rectangle covering all $n+1$ rectangles is at most $(2m+h)w = A$.

♣ **VLSI Discrete Layout "yes" $\Rightarrow$ Bin Packing "yes"**

Suppose that $r_1, r_2, \ldots, r_{n+1}$ can be covered by a rectangle $\tilde{r}$ of area at most A.

There are the following three facts.

Fact 1. The width of $\tilde{r}$ is w, which is the width of r_{n+1}.

Proof. Suppose to the contrary that the width of $\tilde{r}$ is at least $w+1$.

Since the height of r_{n+1} is h, the area of $\tilde{r}$ is at least

$$
\begin{aligned}
& h(w+1) \\
= \quad & hw+h \\
= \quad & hw+(2mw+1) \\
= \quad & (h+2m)w+1 \\
= \quad & A+1, \quad \text{a contradiction !}
\end{aligned}
$$

Fact 2. Each r_i $(1 \le i \le n)$ is placed with height 1, not of height $(2m+1)s_i-1$.

Proof. If some r_i is placed with height $(2m+1)s_i-1$, then the height of $\tilde{r}$ is at least $h+((2m+1)s_i-1)+1 =$ $h+(2m+1)s_i$.

So, the area of $\tilde{r}$ is at least

$$
\begin{aligned}
& (h+(2m+1)s_i)w \\
= \;& hw + 2mws_i + ws_i \\
= \;& (2ms_i + h)w + ws_i \\
> \;& A + ws_i \qquad \text{a contradiction !}
\end{aligned}
$$

Fact 3. The total number of rows occupied by $r_1, r_2, \ldots, r_n$ is at most m.

Proof. If it is not true, then the area of $\tilde{r}$ is larger than $(2m+h)w = A$, a contradiction.

According to the three facts, the placement of $r_1, r_2, \ldots, r_{n+1}$ is like the one shown on page 49.

Then, put the items corresponding to the rectangles of row i into b_i.

Suppose that there are n_i items stored in b_i whose sizes are $d_{i,1}$, $d_{i,2}$, …, d_{i,n_i}.

Since the width of row i $(1 \le i \le m)$ is at most w, we have

$$
\begin{aligned}
& w \;\; (= (2m+1)c-1) \\
\ge \quad & (n_i - 1) + \sum_{r=1}^{n_i} ((2m+1)d_{i,r} - 1) \\
= \quad & (n_i - 1) + (2m+1)\sum_{r=1}^{n_i} d_{i,r} - n_i \\
= \quad & (2m+1)\sum_{r=1}^{n_i} d_{i,r} - 1.
\end{aligned}
$$

$$\Rightarrow \quad \sum_{r=1}^{n_i} d_{i,r} \le c$$