Aalto University
School of Science

CS-E4530 Computational Complexity Theory

Lecture 9: $\mathbf{NP}$-Complete Problems II: Paths, Colourings, Sets and Numbers

Aalto University
School of Science
Department of Computer Science

Spring 2020

Topics

- Graphs: paths and colourings
- Sets and numbers
- Pseudopolynomial algorithms

(C. Papadimitriou: *Computational Complexity*, Chapters 9.3–9.4)

9.4 Graphs: Paths and Colourings

Theorem (9.15)

HAMILTON PATH is $\mathbf{NP}$*-complete.*

Proof

- Reduction from 3SAT to HAMILTON PATH:
  Given a formula ϕ in CNF with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$ each with three literals, we construct a graph $R(\phi)$ that has a Hamilton path iff ϕ is satisfiable.
- *Choice gadgets* select a truth assignment for variables x_i.
- *Consistency gadgets* (XOR) enforce that all occurrences of x_i have the same truth value and all occurrences of $\neg x_i$ the opposite.
- *Constraint gadgets* guarantee that all clauses are satisfied.

Gadgets [Papadimitriou, 1994]

Figure 9-4. The choice gadget.

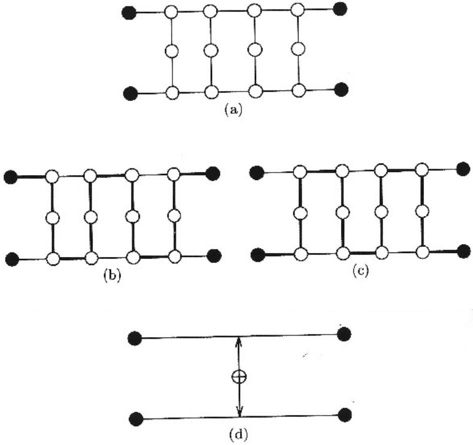


Figure 9-5. The consistency gadget.

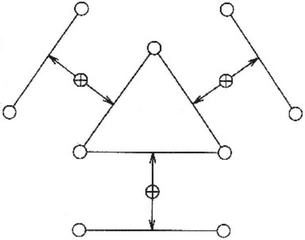


Figure 9-6. The constraint gadget.

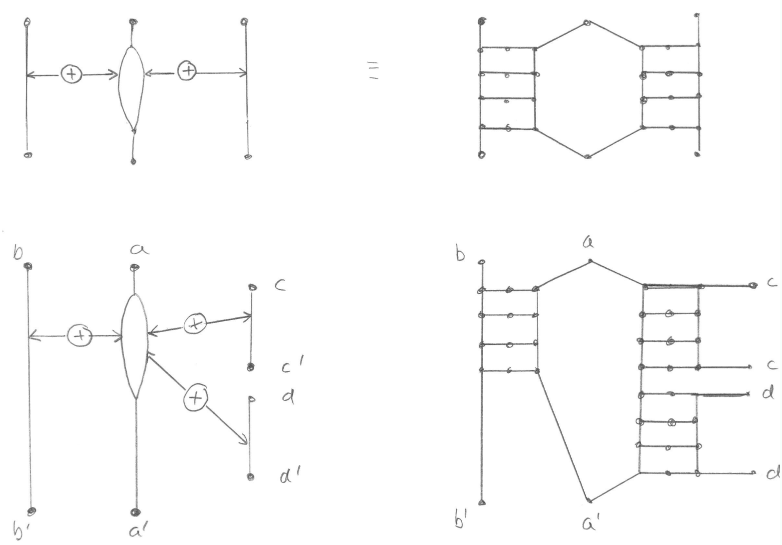
Proof (cont'd)

The graph $R(\phi)$ is constructed as follows:

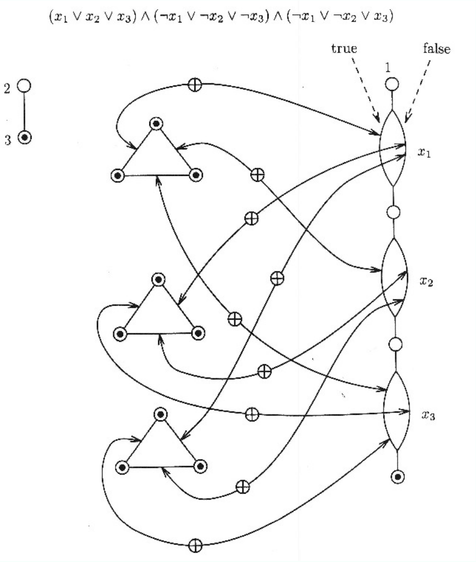
- The *choice gadgets* for each variable x_i are connected in series.
- A *constraint gadget* (triangle) is added for each clause, with an edge associated to each literal l in the clause.
  – If l is x_i, then XOR paired with **true** edge of choice gadget x_i.
  – If l is $\neg x_i$, then XOR paired with **false** edge of choice gadget x_i.
- All end vertices of the constraint triangles, the end vertices of the choice gadget series, and a new vertex 3 are connected together as a clique. Another new vertex 2 is connected to vertex 3.

Basic idea: Each side of a constraint gadget is traversed by the Hamilton path iff the corresponding literal is **false**. Hence, a Hamilton path exists iff at least one literal in each clause is **true**, because otherwise the path should traverse all sides in that triangle, which by design is not possible.

Proof (cont'd)



Proof (cont'd)



[Papadimitriou, 1994]

Proof (cont'd)

Correctness of the reduction

- If ϕ is satisfiable, then there is a Hamilton path:
  From a satisfying truth assignment, we construct a Hamilton path by starting at 1 and traversing choice gadgets according to the truth assignment; the remainder of the graph is then connected into the core clique in a way for which a trivial path can be found that eventually leads to 3 and finally to 2.
- If there is a Hamilton path, then ϕ is satisfiable:
  The path starts at 1, makes a truth assignment, traverses the triangles in some order and ends up in 2. The truth assignment satisfies ϕ as there cannot be a triangle where all sides are traversed by the path, i.e., where all literals would be **false**. □

Travelling salesperson (TSP) revisited

Corollary (9.16)

TSP(D) is $\mathbf{NP}$*-complete.*

Proof

A reduction from HAMILTON PATH to TSP(D). Given a graph G with n vertices, construct a distance matrix d_{ij} and a budget B so that there is a tour of length at most B iff G has a Hamilton path.

- There are n cities, with distance $d_{ij} = 1$ if $\{i, j\} \in G$ and $d_{ij} = 2$ otherwise. The budget is set to $B = n + 1$.
- If there is a TSP tour of length $n + 1$ or less, then it contains at most one pair $(\pi(i), \pi(i+1))$ with cost 2, i.e., a pair for which $\{\pi(i), \pi(i+1)\}$ is not an edge. Removing this pair (if any) gives a Hamilton path.
- If G has a Hamilton path, then its TSP cost is $n - 1$, and it can be made into a tour with an additional cost of at most 2. $\square$

Graph colouring

Definition (9.8)

k-COLOURING:

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

QUESTION: Is there an assignment of one of k colours to each of the vertices in V such that no two vertices i,j connected by an edge $\{i,j\} \in E$ have the same colour?

☞ Colouring with $k = 2$ colours is easy (in $\mathbf{P}$), but probably difficult with $k \geq 3$.

Theorem (9.17)

3-COLOURING is **NP***-complete.*

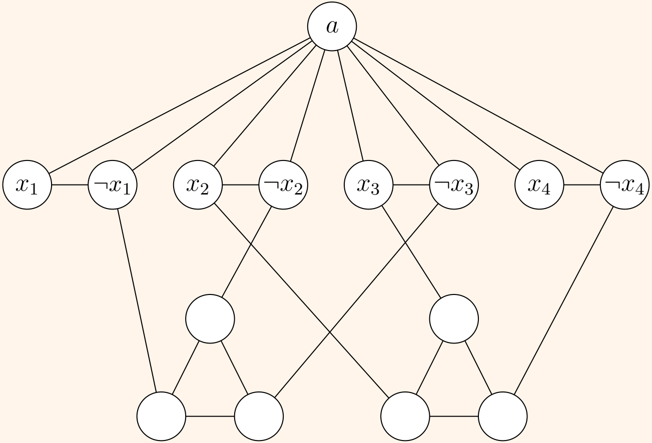
Proof

A reduction from NAESAT to 3-COLOURING.

- For a conjunction of clauses $\phi = C_1 \wedge \ldots \wedge C_m$ with variables $x_1, \ldots, x_n$, construct a graph $G(\phi)$ that can be coloured with $\{0, 1, 2\}$ iff there is a truth assignment satisfying all clauses in ϕ in the NAESAT sense.
- *Choice gadgets:* For each variable x_i, introduce a triangle $[a, x_i, \neg x_i]$, i.e. all triangles share a vertex a.
- *Constraints:* For each clause C_i, introduce a triangle $[C_{i1}, C_{i2}, C_{i3}]$ where each C_{ij} is further connected to the vertex representing the jth literal in C_i.

Example

Reducing NAESAT to 3-COLOURING:



$\neg x_1 \vee \neg x_2 \vee \neg x_3$ $\quad$ $x_2 \vee x_3 \vee \neg x_4$

Correctness of the reduction

($\Rightarrow$) Assume that ϕ is satisfied by T in the sense of NAESAT. Then we can extract a colouring for G from T as follows:

1. Colour vertex a with colour 2.
2. If $T(x_i) = \mathbf{true}$, then colour x_i with 1 and $\neg x_i$ with 0, else vice versa.
3. From each $[C_{i1}, C_{i2}, C_{i3}]$, colour two literals having opposite truth values with 0 (**true**) and 1 (**false**). Colour the third with 2.

($\Leftarrow$) Assume that G can be coloured with $\{0, 1, 2\}$ and a has colour 2. This induces a truth assignment T via the colours of the vertices x_i: if the colour of x_i is 1, then set $T(x_i) = \mathbf{true}$ else set $T(x_i) = \mathbf{false}$.

Suppose that T assigns all literals of some clause C_i to **true**/**false**. Then the colour $1/0$ cannot be used in colouring $[C_{i1}, C_{i2}, C_{i3}]$, contradicting the assumption that G can be coloured with $\{0, 1, 2\}$.

Thus, ϕ is satisfied in the sense of NAESAT. $\square$

9.5 Sets and Numbers

Definition (9.9)

TRIPARTITE MATCHING:
INSTANCE: Three sets B (boys), G (girls), and H (houses) each containing n elements, and a ternary relation $T \subseteq B \times G \times H$.
QUESTION: Is there a set of n triples in T, no two of which have a component in common?

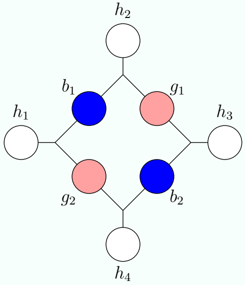
Theorem (9.18)

TRIPARTITE MATCHING is $\mathbf{NP}$*-complete.*

Proof

By reduction from 3SAT. Each variable x has a combined choice and consistency gadget, and each clause c a dedicated pair of boy b_c and girl g_c, together with three triples (b_c, g_c, h_l) where h_l ranges over the three houses corresponding to the occurrences of literals in the clause (appearing in the combined gadgets).

The combined gadget for choice and consistency

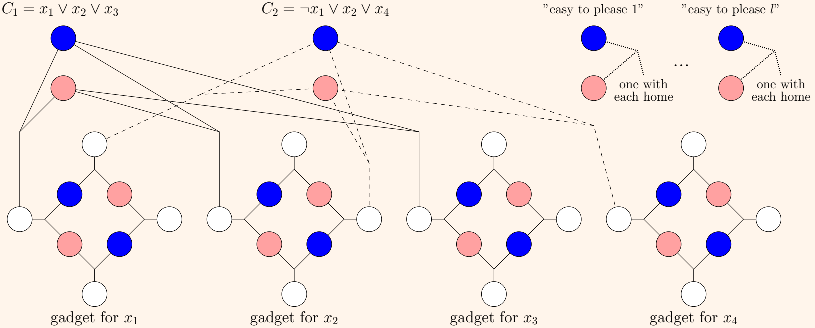


The gadget for a variable x involves k boys, k girls and $2k$ houses forming a "k-circle", where k is either the number of occurrences of x or its negation whichever is larger. (Recall from Proposition 9.3 that we can in fact assume $k = 2$.) The case $k = 2$ is given alongside.

- Occurrences of x in the clauses are connected to the odd houses h_{2i-1} for x and to the even houses h_{2i} for $\neg x$.
- Exactly two kinds of matchings in the variable gadget for x are possible:
  – "$T(x) = \mathbf{true}$": each b_i with g_i and h_{2i}.
  – "$T(x) = \mathbf{false}$": each b_i with g_{i-1} (g_k if $i = 1$) and h_{2i-1}.

Example

Reducing 3SAT to TRIPARTITE MATCHING:



Proof (cont'd)

Correctness of the reduction

- Note that a "$T(x) = \mathbf{true}$" matching in the variable gadget for x leaves the odd houses unoccupied, and a "$T(x) = \mathbf{false}$" matching respectively the even houses.
- For a clause c, the dedicated b_c and g_c can be matched to a house h in a variable gadget for x that is left unoccupied when x is assigned a truth value satisfying c.
- One more detail needs to be settled: there are now more houses H than boys B and girls G (but $|B| = |G|$).
- Solution: add $l = |H| - |B|$ new boys and l new girls. The ith new girl participates in $|H|$ triples containing the ith new boy and each house.
- Now a tripartite matching exists iff the set of clauses is satisfiable.

Other problems involving sets, viewed as generalisations of TRIPARTITE MATCHING

Definition (9.10)

EXACT COVER BY 3-SETS:
INSTANCE: A family $F = \{S_1, \ldots, S_n\}$ of subsets of a finite set U such that $|U| = 3m$ for some integer m and $|S_i| = 3$ for all i.
QUESTION: Is there a subfamily of m sets in F that are disjoint and have U as their union?

Corollary (9.19)

EXACT COVER BY 3-SETS is $\mathbf{NP}$*-complete.*

Proof (sketch)

TRIPARTITE MATCHING can be reduced to EXACT COVER BY 3-SETS by noticing that it is a special case where U is partitioned in three sets B, G, H with $|B| = |G| = |H|$ and $F = \{\{b, g, h\} \mid (b, g, h) \in T\}$.

Example

TRIPARTITE MATCHING:
$B = \{b_1, \ldots, b_n\}, G = \{g_1, \ldots, g_n\}$,
$H = \{h_1, \ldots, h_n\}$,
$T = \{(b_1, g_2, h_1), (b_1, g_2, h_2), \ldots\}$

EXACT COVER BY 3-SETS:
$U = \{b_1, \ldots, b_n, g_1, \ldots, g_n, h_1, \ldots, h_n\}$
$F = \{\{b_1, g_2, h_1\}, \{b_1, g_2, h_2\}, \ldots\}$

Definition (9.11)

SET COVER:
INSTANCE: A family $F = \{S_1, \ldots, S_n\}$ of subsets of a finite set U and an integer B.
QUESTION: Is there a subfamily of $\leq B$ sets in F whose union is U?

Corollary (9.20)

SET COVER is $\mathbf{NP}$*-complete.*

Proof (sketch)

EXACT COVER BY 3-SETS can be reduced to SET COVER as a special case where the universe has $3m$ elements, all sets in F have 3 elements and the budget $B = m$.

Example

EXACT COVER BY 3-SETS:
$U = \{e_1, \ldots, e_{93}\}$
$F = \{\{e_1, e_7, e_{11}\}, \{e_2, e_7, e_{11}\}, \ldots\}$

SET COVER:
$U = \{e_1, \ldots, e_{93}\}$
$F = \{\{e_1, e_7, e_{11}\}, \{e_2, e_7, e_{11}\}, \ldots\}$
$B = 31$

Definition (9.12)

SET PACKING:

INSTANCE: A family $F = \{S_1, \ldots, S_n\}$ of subsets of a finite set U and an integer K.

QUESTION: Is there a subfamily of $\geq K$ pairwise disjoint sets in F?

Corollary (9.21)

SET PACKING is $\mathbf{NP}$*-complete.*

Proof (sketch)

EXACT COVER BY 3-SETS can be reduced to SET PACKING as a special case where the universe has $3m$ elements, all sets in F have 3 elements and bound $K = m$.

Example

EXACT COVER BY 3-SETS:

$U = \{e_1, ..., e_{93}\}$

$F = \{\{e_1, e_7, e_{11}\}, \{e_2, e_7, e_{11}\}, ...\}$

SET PACKING:

$U = \{e_1, ..., e_{93}\}$

$F = \{\{e_1, e_7, e_{11}\}, \{e_2, e_7, e_{11}\}, ...\}$

$K = 31$

Some number problems

Definition (9.13)

INTEGER PROGRAMMING:

INSTANCE: A system of linear inequalities with integer coefficients.

QUESTION: Is there an integer solution to the system?

Theorem (9.22)

INTEGER PROGRAMMING is $\mathbf{NP}$*-complete.*

Proof

SET COVER is reducible to INTEGER PROGRAMMING:

Given a family $F = \{S_1, \ldots, S_n\}$ of subsets of a finite set $U = \{u_1, \ldots, u_m\}$ and an integer B, construct a system:

$$
\begin{array}{ll}
0 \leq x_1 \leq 1, \ldots, 0 \leq x_n \leq 1 \qquad & a_{11}x_1 + \cdots + a_{1n}x_n \geq 1 \\
\Sigma_{i=1}^{n} x_i \leq B & \vdots \\
 & a_{m1}x_1 + \cdots + a_{mn}x_n \geq 1
\end{array}
$$

where $a_{ij} = 1$ if element u_i of U belongs to set S_j, otherwise $a_{ij} = 0$. (The idea: in a solution $x_j = 1$ if S_j is in the cover, otherwise $x_j = 0$.)

Example

Reducing SET COVER to INTEGER PROGRAMMING

SET COVER:

$U = \{e_1, \ldots, e_{93}\}$

$F = \{S_1 = \{e_1, e_7, e_{11}\}, S_2 = \{e_2, e_7, e_{11}\}, \ldots, S_{50} = \{e_2, e_{12}\}\}$

$B = 35$

reduces to

INTEGER PROGRAMMING:

$$\begin{array}{lr}
x_1 + 0 + \ldots + 0 \geq 1 & \text{(for } e_1\text{)} \\
0 + x_2 + \ldots + x_{50} \geq 1 & \text{(for } e_2\text{)} \\
\vdots & \\
x_1 + x_2 + \ldots + 0 \geq 1 & \text{(for } e_7\text{)} \\
\vdots &
\end{array}$$

$0 \leq x_1 \leq 1, \ldots, 0 \leq x_{50} \leq 1$

$\sum_{i=1}^{50} x_i \leq 35$

Note: LINEAR PROGRAMMING (i.e. INTEGER PROGRAMMING where non-integer solutions are allowed) is in $\mathbf{P}$. [L. Khachiyan 1979]

Definition (9.14)

KNAPSACK:
INSTANCE: A set of n items with each item i having a value v_i and a weight w_i (both positive integers), and integers K and W.
QUESTION: Is there a subset S of the items such that $\Sigma_{i \in S} v_i \geq K$ and $\Sigma_{i \in S} w_i \leq W$.

Theorem (9.23)

KNAPSACK is $\mathbf{NP}$-complete.

Proof

We show $\mathbf{NP}$-completeness even in the simple special case of $v_i = w_i$ for all i and $K = W$:
INSTANCE: A set of integers $w_1, \ldots, w_n$ and an integer W. QUESTION: Is there a subset S of the integers with $\Sigma_{i \in S} w_i = W$?

Proof (cont'd)

Reduction from EXACT COVER BY 3-SETS

The reduction is based on the set $U = \{1, 2, \ldots, 3m\}$ and the sets $S_1, \ldots, S_n$ given as bit vectors $\{0,1\}^{3m}$ and $W = 2^{3m} - 1$. Then the task is to find a subset of bit vectors that sum to W.

$$
\begin{array}{ccccccc}
\rightarrow & 0 & 1 & \ldots & 0 & 0 \\
 & 1 & 0 & \ldots & 0 & 0 \\
 & \vdots & & & & \\
\rightarrow & 0 & 0 & \ldots & 1 & 1 \\
\hline
 & 1 & 1 & \ldots & 1 & 1
\end{array}
$$

- This does not quite work because of the carry bit, but the problem can be circumvented by using $n+1$ as the base rather than 2.
- Now each S_i corresponds to $w_i = \Sigma_{j \in S_i} (n+1)^{3m-j}$.
- Then a set of these integers w_i adds up to $W = \sum_{j=0}^{3m-1} (n+1)^j$ iff there is an exact cover among $\{S_1, S_2, \ldots, S_n\}$. $\square$

Example

Reducing EXACT COVER BY 3-SETS to KNAPSACK

EXACT COVER BY 3-SETS:

$U = \{e_1, \ldots, e_6\}$

$F = \{S_1 = \{e_1, e_4, e_6\}, S_2 = \{e_1, e_3, e_6\}, S_3 = \{e_2, e_3, e_5\}\}$

reduces to

KNAPSACK:

Integers

$$w_1 = 1 \cdot 4^{6-1} + 0 \cdot 4^{6-2} + 0 \cdot 4^{6-3} + 1 \cdot 4^{6-4} + 0 \cdot 4^{6-5} + 1 \cdot 4^{6-6} = 1041$$
$$w_2 = 1 \cdot 4^{6-1} + 0 \cdot 4^{6-2} + 0 \cdot 4^{6-3} + 1 \cdot 4^{6-4} + 0 \cdot 4^{6-5} + 1 \cdot 4^{6-6} = 1089$$
$$w_3 = 0 \cdot 4^{6-1} + 1 \cdot 4^{6-2} + 1 \cdot 4^{6-3} + 0 \cdot 4^{6-4} + 1 \cdot 4^{6-5} + 0 \cdot 4^{6-6} = 324$$

$W = 4^5 + 4^4 + 4^3 + 4^2 + 4^1 + 4^0 = 1365$

9.6 Pseudopolynomial Algorithms

Proposition (9.24)

Any instance of KNAPSACK can be solved in time $O(nW)$, where n is the number of items and W is the weight bound.

Proof

- Define $V(w,i)$: the largest value attainable be selecting some among the first i items so that their total weight is exactly w.
- Each $V(w,i)$ with $w = 1,\ldots,W$ and $i = 1,\ldots,n$ can be computed by

  $$V(w,i+1) = \max\{V(w,i), v_{i+1} + V(w - w_{i+1}, i)\}$$

  where $V(w,i) = -\infty$ if $w \leq 0$, $V(0,i) = 0$ for all i, and $V(w,0) = -\infty$ if $w \geq 1$.
- For each entry this can be done in constant number of steps and there are nW entries. Hence, the algorithm runs in $O(nW)$ time.
- An instance is answered "yes" iff there is an entry $V(w,i) \geq K$.

Example

Items $\{(v_1 = 3, w_1 = 7), (v_2 = 4, w_2 = 5), (v_3 = 4, w_3 = 4), (v_4 = 7, w_4 = 3), (v_5 = 2, w_5 = 3)\}$
weight limit $W = 10$, capacity limit $K = 12$

$V(w,i)$

v_i	w_i	i \ w	0	1	2	3	4	5	6	7	8	9	10
2	3	5	0	$-\infty$	$-\infty$	7	4	4	9	11	11	8	13
7	3	4	0	$-\infty$	$-\infty$	7	4	4	$-\infty$	11	11	8	10
4	4	3	0	$-\infty$	$-\infty$	$-\infty$	4	4	$-\infty$	3	$-\infty$	8	$-\infty$
4	5	2	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	4	$-\infty$	3	$-\infty$	$-\infty$	$-\infty$
3	7	1	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	3	$-\infty$	$-\infty$	$-\infty$
		0	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Strong NP-completeness

- The preceding algorithm is not polynomial in the length of the input (which is $O(n \log W)$) but exponential ($W = 2^{\log W}$).
- An algorithm whose time bound is polynomial in the integer *values* in the input (not their logarithmic *lengths*) is called *pseudopolynomial*.
- A problem is called **strongly NP-complete** if the problem remains $\mathbf{NP}$-complete even if any instance of length n is restricted to contain integers of value at most $p(n)$, for a polynomial p.
  ☞ Strongly $\mathbf{NP}$-complete problems cannot have pseudopolynomial algorithms (unless $\mathbf{P} = \mathbf{NP}$).
- SAT, MAX CUT, TSP(D), HAMILTON PATH, $\ldots$ are strongly $\mathbf{NP}$-complete but KNAPSACK is not.

A strongly NP-complete number problem: BIN PACKING

Definition (9.15)

BIN PACKING
INSTANCE: Positive integers $a_1, \ldots, a_n$ (items) and integers C (capacity) and B (number of bins).
QUESTION: Is there a partition of the items into B subsets such that for each subset S, $\Sigma_{a_i \in S} a_i \leq C$?

- BIN PACKING is strongly $\mathbf{NP}$-complete: Even if the integers are restricted to have polynomial values (w.r.t. the length of input), the problem remains $\mathbf{NP}$-complete.
- For the proof, see Papadimitriou, pp. 204–205. (Reduction from TRIPARTITE MATCHING.)

Learning Objectives

- The concept of $\mathbf{NP}$-completeness and its characterisation in terms of succinct certificates.
- You know basic techniques to prove problems $\mathbf{NP}$-complete and are able to construct such proofs on your own.
- A basic repertoire of $\mathbf{NP}$-complete problems (related with satisfiability, graphs, sets, and numbers) to be used in further $\mathbf{NP}$-completeness proofs.
- The definition of strong $\mathbf{NP}$-completeness and awareness of number problems which are (or are not) strongly $\mathbf{NP}$-complete.