Integer and Combinatorial Optimization:
Node Packing is NP-Complete

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

January 2019
**Node packing**

**Definition**

A node packing on a graph $G = (V, E)$ is a subset $U \subseteq V$ of the vertices so that no two vertices in $U$ are adjacent.

Example:

- $U = \{4, 5\}$ is a node packing of cardinality 2.

**Definition**

Let $G = (V, E)$ be a graph and let $z$ be a positive integer. The unweighted node packing with lower bound problem is to determine whether there is a node packing of cardinality at least $z$. 

**Mitchell**

Node Packing is NP-Complete
Node packing

Definition

A node packing on a graph \( G = (V, E) \) is a subset \( U \subseteq V \) of the vertices so that no two vertices in \( U \) are adjacent.

\[ U = \{4, 5\} \] is a node packing of cardinality 2

Definition

Let \( G = (V, E) \) be a graph and let \( z \) be a positive integer. The unweighted node packing with lower bound problem is to determine whether there is a node packing of cardinality at least \( z \).
Theorem

The unweighted node packing with lower bound problem is \( NP \)-complete.

First show the problem is in \( NP \):
Our certificate of feasibility consists of a list of the nodes in \( U \).

We can check quickly that none of these vertices are adjacent, and there are at least \( z \) of them.
Polynomial transformation

We show that 3-SAT can be polynomially transformed to our node packing problem.

We have an instance of 3-SAT with *n* boolean variables \( y_1, \ldots, y_n \) and *m* clauses \( C_1, \ldots, C_m \), with each clause containing exactly 3 literals.

We construct a graph with \( 3m \) vertices which has a node packing of size at least *m* if and only if the instance of 3-SAT is feasible.

A clique of size 3 is constructed corresponding to each clause.

One vertex is constructed for each literal in the clause, and three edges are constructed so the three vertices are adjacent to each other.
Mapping a clause to a clique

\[ C_i = y_j + y_k + \bar{y}_l \]

\[ C_i \rightarrow k_i \quad (n + l)_i \quad j_i \]
Ensuring consistency
Any packing of cardinality $m$ must contain exactly one vertex from each of these $m$ cliques.

This vertex is a “representative” for the clause.

We need to add additional edges to ensure consistency between clauses.

Thus, if the representative for clause $i$ is $j_i$ then that corresponds to $y_j = \text{TRUE}$, so there cannot be another chosen representative for another clause which corresponds to $y_j = \text{FALSE}$.

Consistency can be ensured by adding edges of the form $(j_i, (n + j_q))$ for any two clauses $C_i$ and $C_q$, where $y_j$ is a literal in clause $C_i$ and $\bar{y}_j$ is a literal in clause $C_q$.

This prevents us having simultaneously:
- $y_j = \text{TRUE}$ as the representative for clause $C_i$, and
- $y_j = \text{FALSE}$ as the representative for clause $C_q$. 
Ensuring consistency

Any packing of cardinality \( m \) must contain exactly one vertex from each of these \( m \) cliques.

This vertex is a “representative” for the clause.

We need to add additional edges to ensure consistency between clauses.

Thus, if the representative for clause \( i \) is \( j_i \) then that corresponds to \( y_j = \text{TRUE} \), so there cannot be another chosen representative for another clause which corresponds to \( y_j = \text{FALSE} \).

Consistency can be ensured by adding edges of the form \((j_i, (n + j)_q)\) for any two clauses \( C_i \) and \( C_q \), where \( y_j \) is a literal in clause \( C_i \) and \( \bar{y}_j \) is a literal in clause \( C_q \).

This prevents us having simultaneously:

- \( y_j = \text{TRUE} \) as the representative for clause \( C_i \), and
- \( y_j = \text{FALSE} \) as the representative for clause \( C_q \).
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]
Example

\[ C_1 = y_1 + y_2 + \bar{y}_3 \]
\[ C_2 = y_2 + y_3 + \bar{y}_4 \]
\[ C_3 = \bar{y}_1 + \bar{y}_2 + y_4 \]

There are many packings of cardinality 3, including \( U = \{1_1, 8_2, 6_3\} \). This corresponds to the valid truth assignment

\[ y_1 = \text{TRUE}, \quad y_4 = \text{FALSE}, \quad y_2 = \text{FALSE}, \quad y_3 \text{ unspecified.} \]
Vertex cover

Given a graph $G = (V, E)$, a vertex cover is a subset $U \subseteq V$ of the vertices so that every edge in $E$ has at least one endpoint in $U$.

The vertex cover with upper bound problem is specified by a graph $G = (V, E)$ and a scalar $K$, and it is desired to find a vertex cover with cardinality $|U| \leq K$.

Exercise

Show that vertex cover with upper bound is NP-complete, by transforming from node packing with lower bound.

Hint: If $W$ is a node packing in graph $G = (V, E)$ then its complement $U = V \setminus W$ is a vertex cover.
A vertex cover illustration

node packing
A vertex cover illustration

vertex cover