Multidimensional Binary Vector Assignment problem: standard, structural and above guarantee parameterizations

Rémi Watrigant[1]

joint work with Marin Bougeret[2], Guillerme Duvillié[2], Rodolphe Giroudeau[2]

[1] Hong Kong Polytechnic University, Hong Kong
[2] LIRMM, Montpellier, France

FCT 2015, Gdansk, Poland. August 17-19 2015

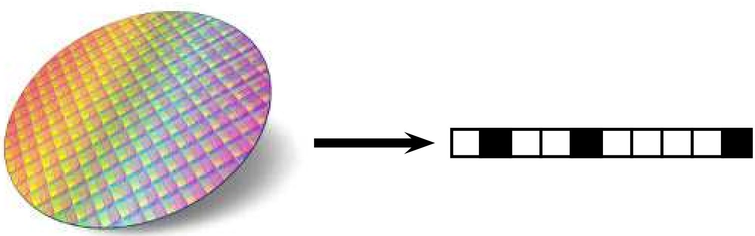
Contents

Applications

Yield maximization in wafer-to-wafer 3D chip integration.

Multidimensional Binary Vector Assignment

- a wafer = a binary vector of good/bad dies (1/0)

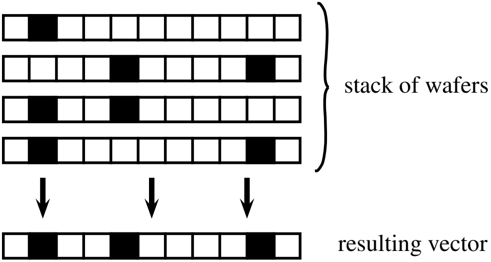


Applications

Yield maximization in wafer-to-wafer 3D chip integration.

Multidimensional Binary Vector Assignment

- a wafer = a binary vector of good/bad dies (1/0)
- a stack = superposition of several wafers

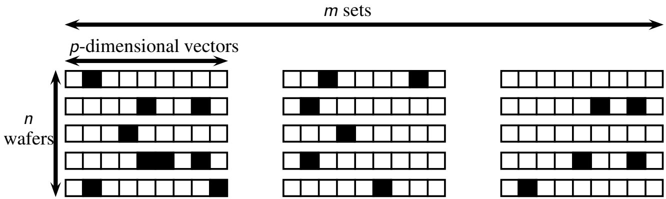


Applications

Yield maximization in wafer-to-wafer 3D chip integration.

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- a wafer = a binary vector of good/bad dies (1/0)
- a stack = superposition of several wafers
- Input: m sets of n wafers (p-dimensional binary vectors)

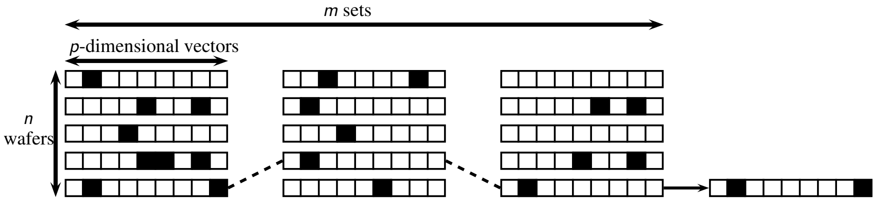


Applications

Yield maximization in wafer-to-wafer 3D chip integration.

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- a wafer = a binary vector of good/bad dies (1/0)
- a stack = superposition of several wafers
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks

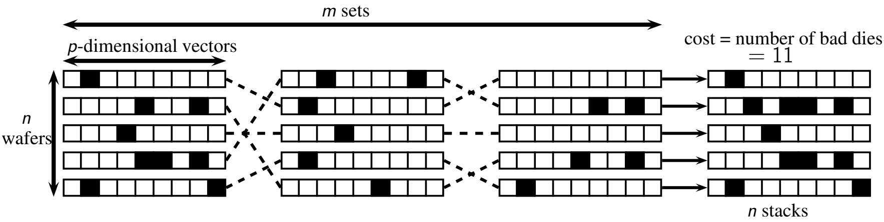


Applications

Yield maximization in wafer-to-wafer 3D chip integration.

Multidimensional Binary Vector Assignment

- a wafer = a binary vector of good/bad dies (1/0)
- a stack = superposition of several wafers
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total



Related works

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)

Related works

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):

Related works

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):
  - $f(m)$-approximation

Related works

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):
  - $f(m)$-approximation
  - $O(p^{1-\epsilon})$ and $O(m^{1-\epsilon})$ inapproximability unless $P = NP$

Related works

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):
  - $f(m)$-approximation
  - $O(p^{1-\epsilon})$ and $O(m^{1-\epsilon})$ inapproximability unless $P = NP$
  - $\frac{p}{c}$-approximation for any $c \in \mathbb{N}$

Related works

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):
  - $f(m)$-approximation
  - $O(p^{1-\epsilon})$ and $O(m^{1-\epsilon})$ inapproximability unless $P = NP$
  - $\frac{p}{c}$-approximation for any $c \in \mathbb{N}$
  - FPT parameterized by p

Related works

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Previous results:

- NP-hard even when $m = 3$ (reduction from 3D matching)
- Approximating the maximization version (at least k good dies):
  - $f(m)$-approximation
  - $O(p^{1-\epsilon})$ and $O(m^{1-\epsilon})$ inapproximability unless $P = NP$
  - $\frac{p}{c}$-approximation for any $c \in \mathbb{N}$
  - FPT parameterized by p
  - $W[1]$-hard for standard parameter (maximization version)

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I})$

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I}) = m$

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I}) = n$

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I}) = p$

Parameterized algorithms

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I}) = k$

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I})$=?

Fixed-Parameter Tractability

A problem is **FPT** if there is an algorithm solving any instance $\mathcal{I}$ in time $O(\ f(\kappa(\mathcal{I}))\ \ poly(|\mathcal{I}|)\ )$

Parameterized algorithms

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I})$=?

Fixed-Parameter Tractability

A problem is **FPT** if there is an algorithm solving any instance $\mathcal{I}$ in time $O(\ f(\kappa(\mathcal{I}))\ \ poly(|\mathcal{I}|)\ )$

Corresponding lower bounds:

Parameterized algorithms

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I})$=?

Fixed-Parameter Tractability

A problem is **FPT** if there is an algorithm solving any instance $\mathcal{I}$ in time $O(\ f(\kappa(\mathcal{I}))\ \ poly(|\mathcal{I}|)\ )$

Corresponding lower bounds:

- $W[1]$, $W[2]$-hardness: we suppose $FPT \neq W[1], W[2]$

Parameterized algorithms

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: pick one wafer from each set to form n stacks
- Goal: obtain at most k bad dies in total

For an instance $\mathcal{I}$ of the problem, choose a parameter $\kappa(\mathcal{I})$=?

Fixed-Parameter Tractability

A problem is **FPT** if there is an algorithm solving any instance $\mathcal{I}$ in time $O(\ f(\kappa(\mathcal{I}))\ \ poly(|\mathcal{I}|)\ )$

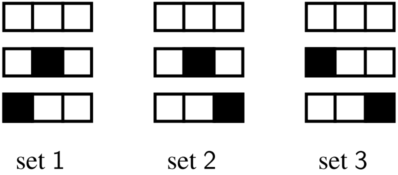
Corresponding lower bounds:

- $W[1]$, $W[2]$-hardness: we suppose $FPT \neq W[1], W[2]$
- Exponential Time Hypothesis: we suppose that 3-SAT cannot be solved in time $O^*(2^{o(n)})$ (n = number of variables)

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Warmup: is it alway safe to create "full-good stacks" ?

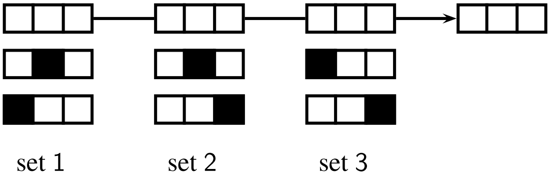


First observations

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Warmup: is it alway safe to create "full-good stacks" ?

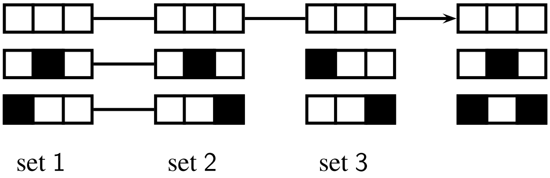


First observations

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

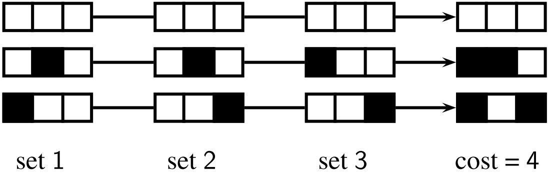
Warmup: is it alway safe to create "full-good stacks" ?



First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

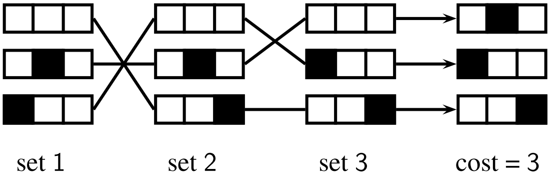
Warmup: is it alway safe to create "full-good stacks" ?



First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Warmup: is it alway safe to create "full-good stacks" ? No!



First observations

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

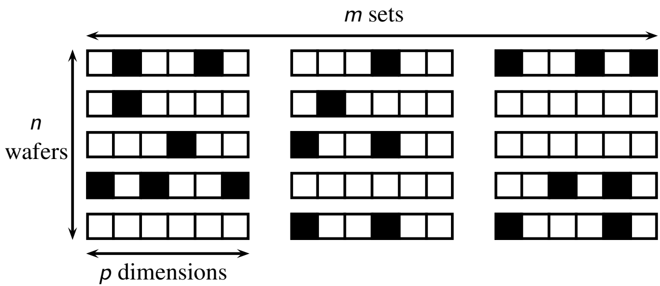
is k a good parameter ? Let us show that we can suppose $n \leq k$

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Suppose $n > k$

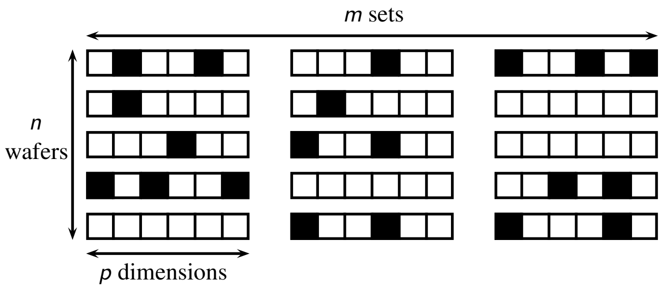


First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Suppose $n > k$



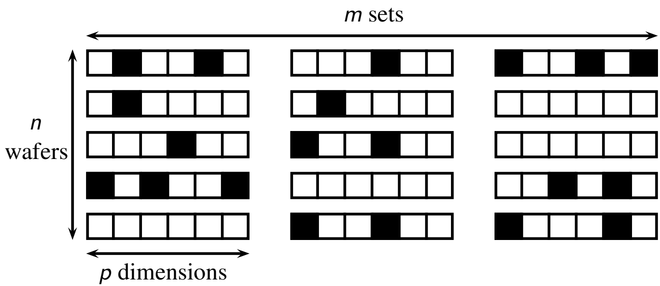
- each set must have a full-good wafer (otherwise cost $\geq n > k$)

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Suppose $n > k$



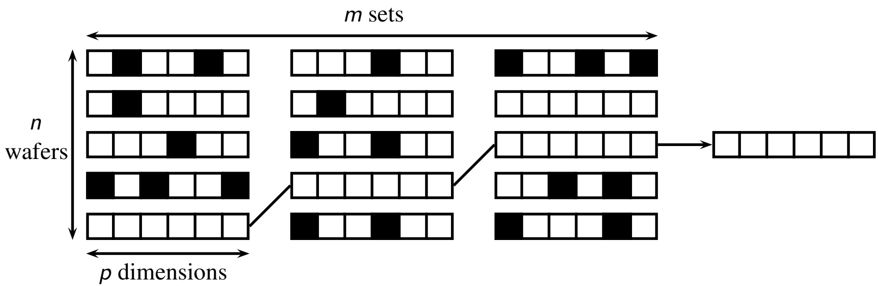
- each set must have a full-good wafer (otherwise cost $\geq n > k$)
- any solution must have a full-good stack (same argument)

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Suppose $n > k$



- each set must have a full-good wafer (otherwise cost $\geq n > k$)
- any solution must have a full-good stack (same argument)

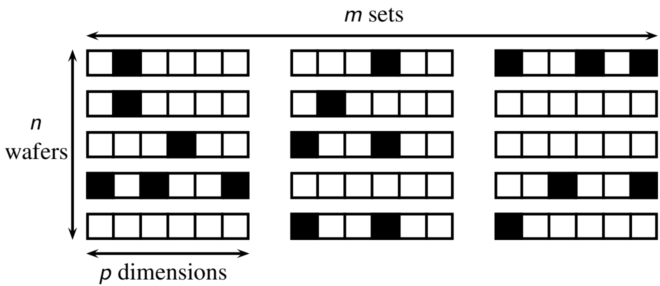
$\Rightarrow$ we can arbitrarily form a full-good stack

remove it, and continue until $n \leq k$

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$
Let us show that we can suppose $p \leq k$ as well

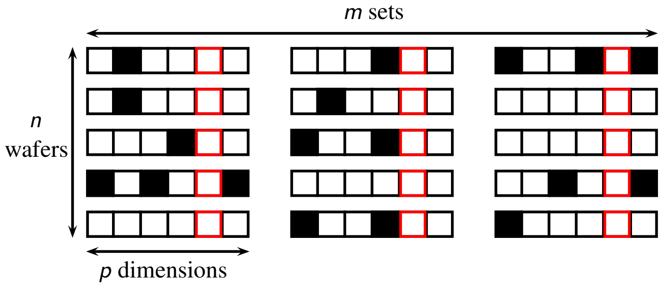


First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Let us show that we can suppose $p \leq k$ as well



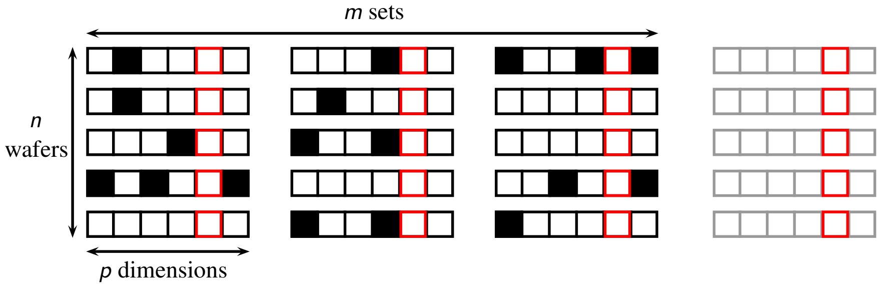
- if there is a component with good dies everywhere

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Let us show that we can suppose $p \leq k$ as well



- if there is a component with good dies everywhere

$\Rightarrow$ any solution will also have this good die

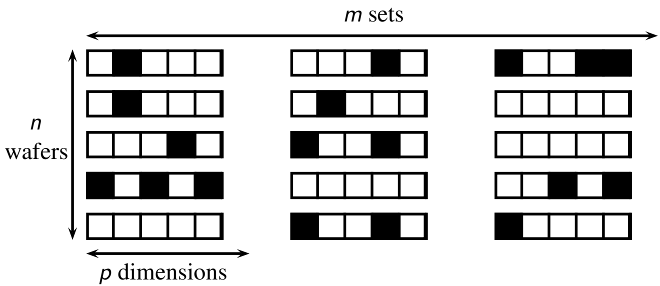
First observations

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Let us show that we can suppose $p \leq k$ as well



- if there is a component with good dies everywhere

$\Rightarrow$ any solution will also have this good die

- we can remove the component from the instance

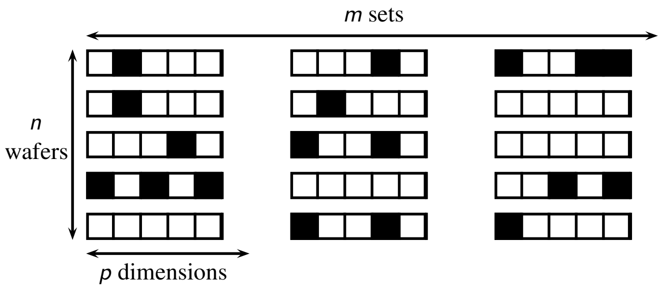
First observations

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Let us show that we can suppose $p \leq k$ as well



- if there is a component with good dies everywhere

$\Rightarrow$ any solution will also have this good die

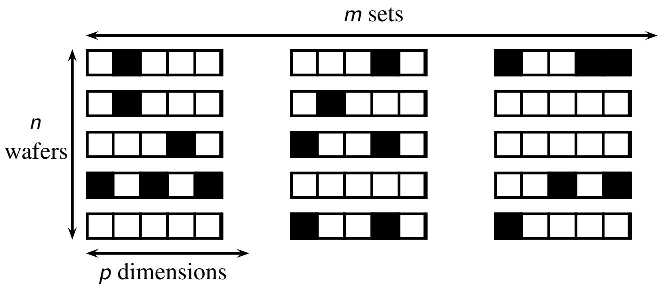
- we can remove the component from the instance

$\Rightarrow$ the cost of any solution will be at least p

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$
Let us show that we can suppose $p \leq k$ as well



Theorem

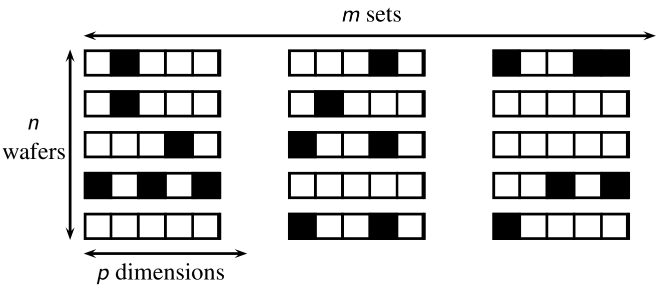
- From any instance, we can produce in polynomial time an equivalent one of size at most $O(k^2 m)$ $\Rightarrow$ kernel of size $O(k^2 m)$

First observations

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

is k a good parameter ? Let us show that we can suppose $n \leq k$

Let us show that we can suppose $p \leq k$ as well



Theorem

- From any instance, we can produce in polynomial time an equivalent one of size at most $O(k^2m)$ $\Rightarrow$ kernel of size $O(k^2m)$
- No kernel polynomial in p even for $m = 3$ (unless $NP \subseteq coNP/poly$)

Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

We just proved that we can suppose $n, p \leq k \Rightarrow k$ is a too large parameter
Idea: substracting a lower bound to the objective function

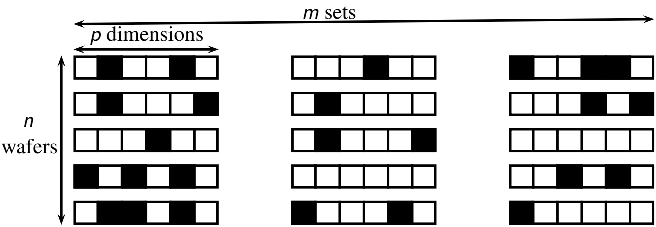
Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

We just proved that we can suppose $n, p \leq k \Rightarrow k$ is a too large parameter

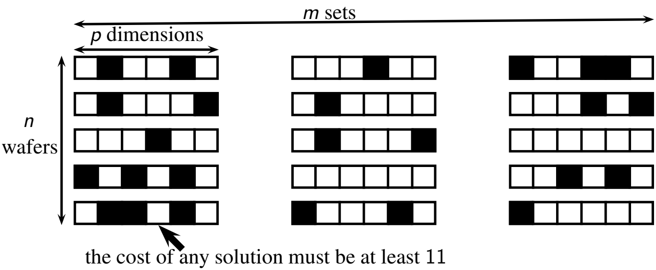
Idea: substracting a lower bound to the objective function



Above guarantee parameterization

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

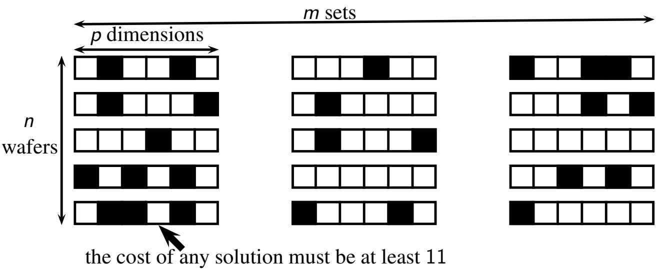
We just proved that we can suppose $n, p \leq k \Rightarrow k$ is a too large parameter
Idea: substracting a lower bound to the objective function



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

We just proved that we can suppose $n, p \leq k \Rightarrow k$ is a too large parameter
Idea: substracting a lower bound to the objective function
Let $\mathcal{B}$ be the maximum number of bad dies in a set
$\Rightarrow$ parameter $\zeta_{\mathcal{B}} = k - \mathcal{B}$

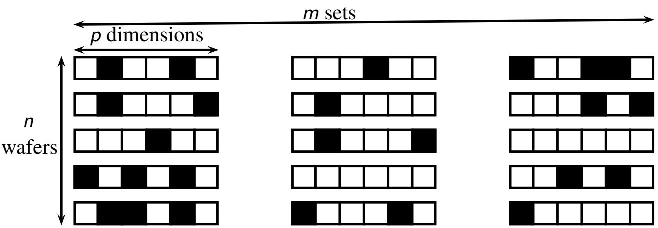


Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

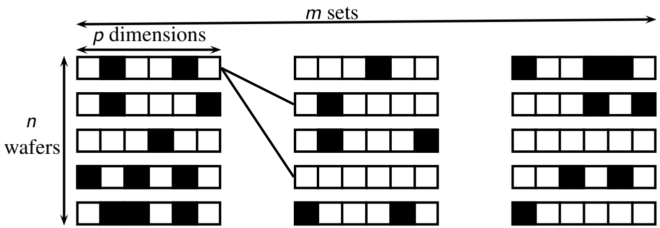
- Find in polynomial time if there is a costless assignment between two sets



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

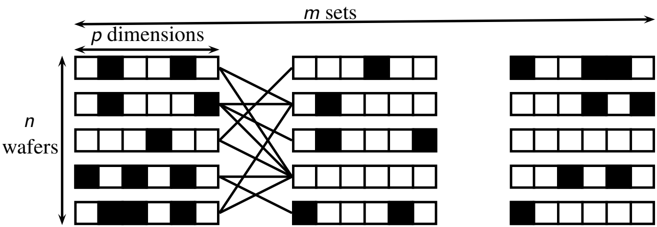
- Find in polynomial time if there is a costless assignment between two sets



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

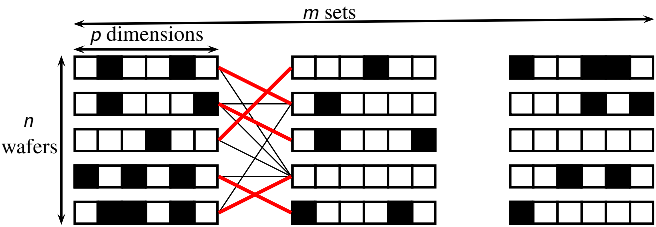
- Find in polynomial time if there is a costless assignment between two sets



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

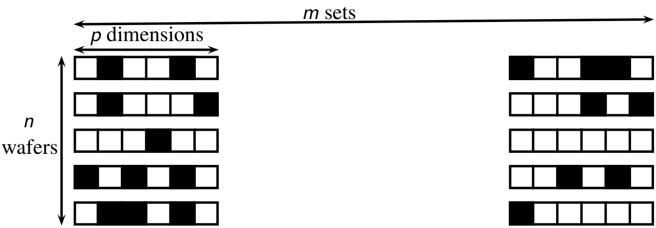
- Find in polynomial time if there is a costless assignment between two sets



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets

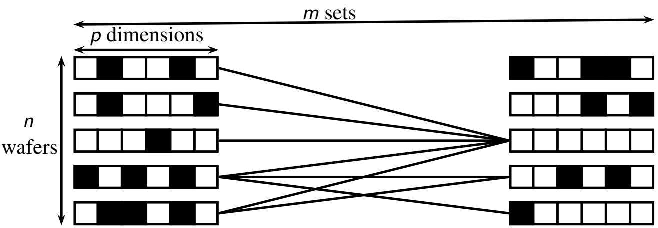


Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

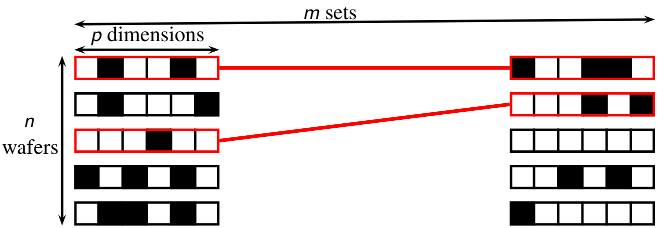
- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies

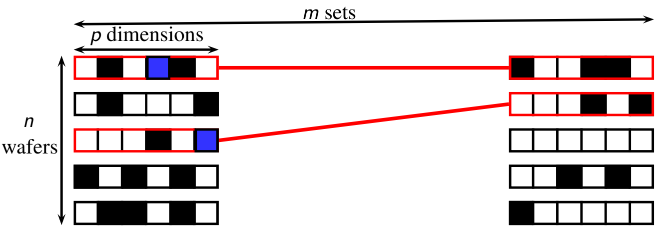


Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

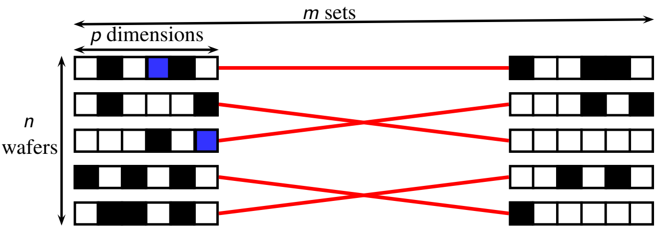
- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

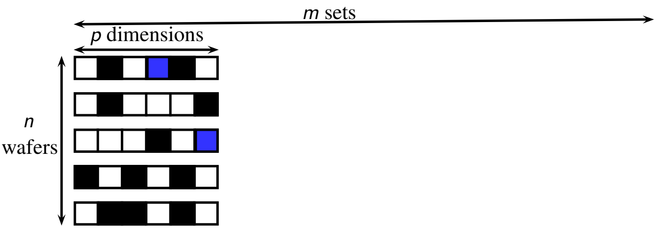
- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others



Above guarantee parameterization

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others

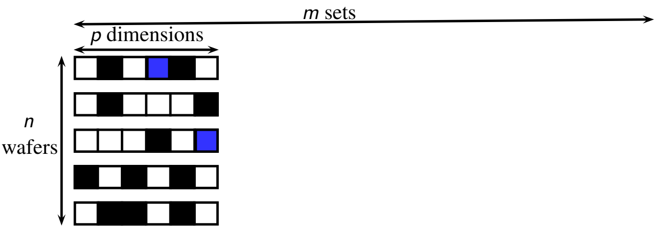


Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
    - guess all couples of wafers which will induce new bad dies
    - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times



Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Simple, but somehow tight, because:

Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Simple, but somehow tight, because:

- $W[2]$-hard parameterized by $\zeta_{\mathcal{B}}$ only

Above guarantee parameterization

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Simple, but somehow tight, because:
- $W[2]$-hard parameterized by $\zeta_{\mathcal{B}}$ only
- no $2^{o(\zeta_{\mathcal{B}}) \log(n)}$ nor $2^{\zeta_{\mathcal{B}} o(\log(n))}$ under *ETH*

Above guarantee parameterization

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Simple, but somehow tight, because:
- $W[2]$-hard parameterized by $\zeta_{\mathcal{B}}$ only
- no $2^{o(\zeta_{\mathcal{B}}) \log(n)}$ nor $2^{\zeta_{\mathcal{B}} o(\log(n))}$ under ETH
- no $2^{o(k)}$ (and thus no $2^{o(\zeta_{\mathcal{B}})}$) for fixed n under ETH.

Above guarantee parameterization

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

- Find in polynomial time if there is a costless assignment between two sets
- If no such assignment can be found, branch:
  - guess all couples of wafers which will induce new bad dies
  - find a costless assignment for the others
- branching of width n^2, applied at most $\zeta_{\mathcal{B}}$ times

Theorem

There is an exact algorithm solving the problem in $O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ time.

Simple, but somehow tight, because:

- $W[2]$-hard parameterized by $\zeta_{\mathcal{B}}$ only
- no $2^{o(\zeta_{\mathcal{B}}) \log(n)}$ nor $2^{\zeta_{\mathcal{B}} o(\log(n))}$ under *ETH*
- no $2^{o(k)}$ (and thus no $2^{o(\zeta_{\mathcal{B}})}$) for fixed n under *ETH*.

Lower bounds

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$

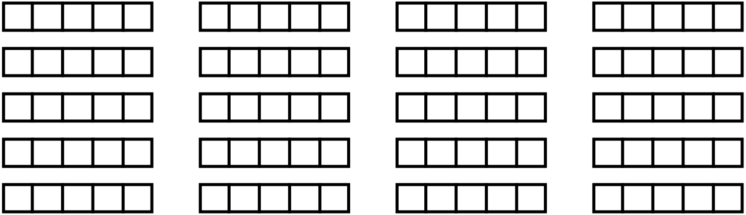
Lower bounds

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$



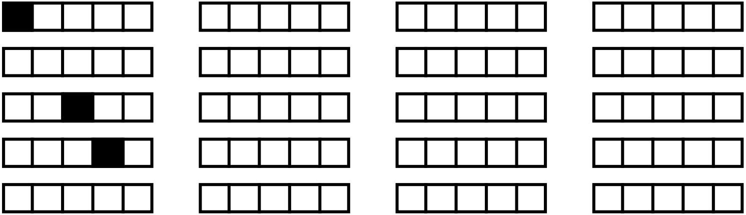
Lower bounds

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

$$\text{ground set} = \{1,2,3,4,5\}$$
$$\text{instance} = \{1,3,4\},\{2,3,4\},\{2,3,5\},\{1,4,5\}$$



Lower bounds

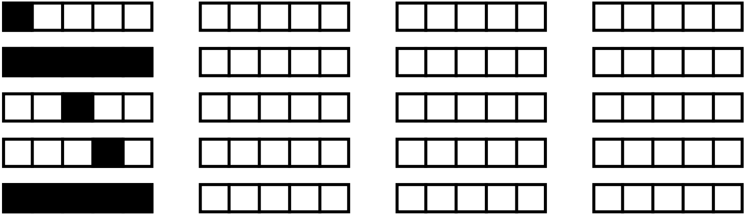
MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

ground set = $\{1, 2, 3, 4, 5\}$

instance = $\{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$



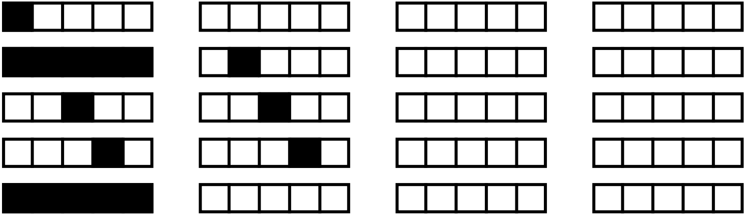
Lower bounds

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$



Lower bounds

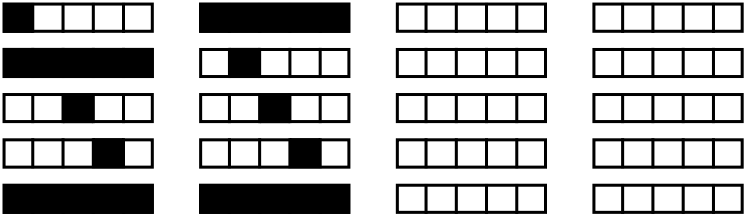
Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

ground set $= \{1, 2, 3, 4, 5\}$

instance $= \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$



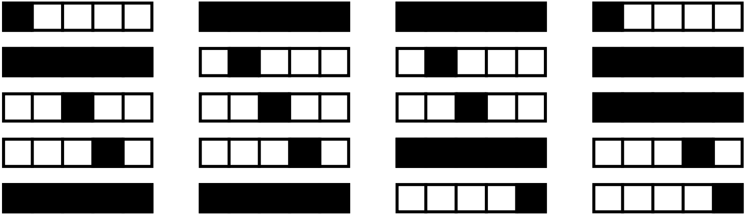
Lower bounds

Multidimensional Binary Vector Assignment

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$

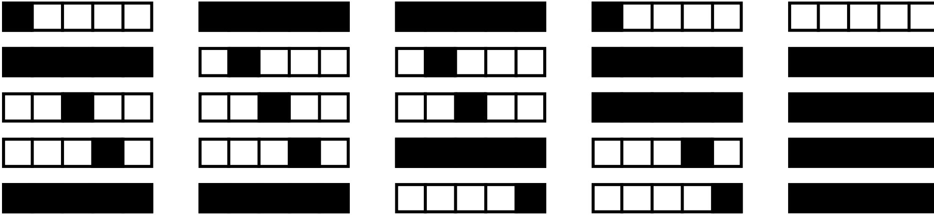


Lower bounds

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

$$\text{ground set} = \{1,2,3,4,5\}$$
$$\text{instance} = \{1,3,4\},\{2,3,4\},\{2,3,5\},\{1,4,5\}$$



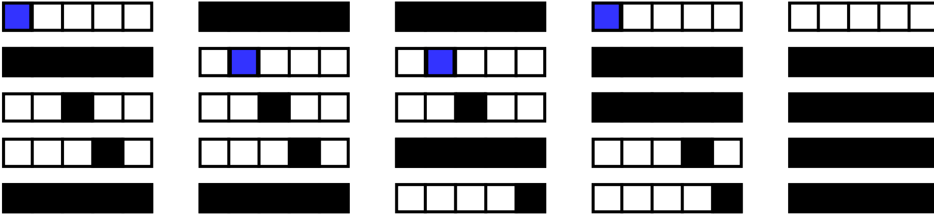
Lower bounds

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$

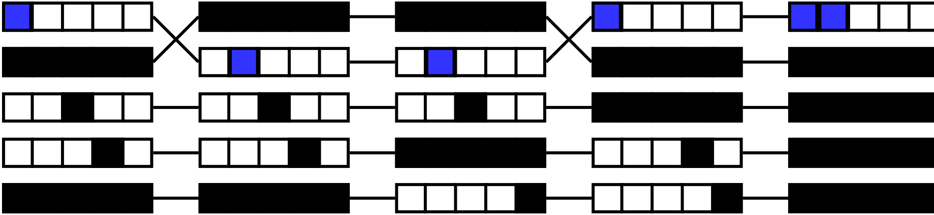


Lower bounds

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

$$\text{ground set} = \{1, 2, 3, 4, 5\}$$
$$\text{instance} = \{1, 3, 4\}, \{2, 3, 4\}, \{2, 3, 5\}, \{1, 4, 5\}$$



Lower bounds

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT

- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

- $\zeta_{\mathcal{B}}$ = size of the hitting set $\Rightarrow$ $W[2]$-hardness parameterized by $\zeta_{\mathcal{B}}$

Lower bounds

Multidimensional Binary Vector Assignment
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from Hitting Set:

- $\zeta_{\mathcal{B}}$ = size of the hitting set $\Rightarrow$ $W[2]$-hardness parameterized by $\zeta_{\mathcal{B}}$

Theorem [Lokshtanov, Marx, Saurabh, '11]

Assuming ETH, no $O^*(2^{o(k\log(k))})$ algorithm for Hitting Set where the goal is to find a hitting set of size k in an instance where the ground set is of size k^2

Lower bounds

MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT
- Input: m sets of n wafers (p-dimensional binary vectors)
- Output: n stacks (of m wafers each)
- Goal: obtain at most k bad dies in total

Reduction from HITTING SET:

- $\zeta_{\mathcal{B}}$ = size of the hitting set $\Rightarrow W[2]$-hardness parameterized by $\zeta_{\mathcal{B}}$

Theorem [Lokshtanov, Marx, Saurabh, '11]

Assuming *ETH*, no $O^*(2^{o(k \log(k))})$ algorithm for HITTING SET where the goal is to find a hitting set of size k in an instance where the ground set is of size k^2

$\Rightarrow$ no $O^*(2^{o(\zeta_{\mathcal{B}}) \log(n)})$ nor $O^*(2^{\zeta_{\mathcal{B}} o(\log(n))})$ for our problem, under *ETH*.

Summary of the results

Positive results	Negative results
$O(k^2 m)$ kernel	no $p^{O(1)}$ kernel unless $NP \subseteq coNP/poly$
$O^*(4^{\zeta_{\mathcal{B}} \log(n)})$ algorithm	$W[2]$-hard for $\zeta_{\mathcal{B}}$ only no $2^{o(\zeta_{\mathcal{B}}) \log(n)}$ nor $2^{\zeta_{\mathcal{B}} o(\log(n))}$ under ETH no $2^{o(k)}$ for fixed n under ETH
$O^*(d^{\zeta_p})$ algorithm for $n = 2$	NP-hard for $\zeta_p = 0$ and fixed $n \geq 3$

Summary of the results

Positive results	Negative results
$O(k^2 m)$ kernel	no $p^{O(1)}$ kernel unless $NP \subseteq coNP/poly$
$O^*(4^{\zeta_\mathcal{B} \log(n)})$ algorithm	$W[2]$-hard for $\zeta_\mathcal{B}$ only no $2^{o(\zeta_\mathcal{B}) \log(n)}$ nor $2^{\zeta_\mathcal{B} o(\log(n))}$ under ETH no $2^{o(k)}$ for fixed n under ETH
$O^*(d^{\zeta_p})$ algorithm for $n = 2$	NP-hard for $\zeta_p = 0$ and fixed $n \geq 3$

Open questions:

- algorithm in $O^*(2^k)$? (n part of the input)
- polynomial kernel parameterized by k only ?

Thank you for your attention !