Multidimensional Assignment Problems for Semiconductor Plants

Trivikram Dokka, Yves Crama, Frits Spieksma

ORSTAT, KULeuven

April 1, 2014
About merging vectors

Our problem - a prologue

- Let $u = (12 \ 91 \ 7)$, and $v = (47 \ 32 \ 12)$. 
About merging vectors

Our problem - a prologue

- Let $u = (12 \ 91 \ 7)$, and $v = (47 \ 32 \ 12)$.  
- How do we merge $u$ and $v$?
About merging vectors

Our problem - a prologue

- Let $u = (12 \ 91 \ 7)$, and $v = (47 \ 32 \ 12)$.
- How do we merge $u$ and $v$?
- Well, we say that $u \vee v = (\max(u_1, v_1), \max(u_2, v_2), \max(u_3, v_3)) = (47 \ 91 \ 12)$

Oh, and the cost of a vector is represented by a function $c(u) : \mathbb{Z}^p_+ \to \mathbb{R}_+$. 
Our Problem

Instance:

- $m$ sets: $V_1, V_2, \ldots, V_m$
- Each $V_i$ consists of $n$ vectors each of size $p$, $1 \leq i \leq m$
- Each entry of a vector is a non-negative integer

Objective:

- partition the given $m$ sets into $n m$-tuples, such that each $m$-tuple contains one vector from each set $V_i$
- minimize the total cost of this partition

We will abbreviate the name of this problem as MVA.
Let $m = 3$, and let the three sets be denoted by $V_1$, $V_2$, and $V_3$. The length of each vector, $p$, equals 3, and $n = 4$, and let us specify $c$ as the sum of the entries of a vector, ie, $c(u) = \sum_{i=1}^{p} u_i$.

\[
\begin{align*}
V_1 & \quad V_2 & \quad V_3 \\
(12 & 91 & 7) & \quad (47 & 31 & 12) & \quad (83 & 3 & 37) \\
(54 & 29 & 64) & \quad (5 & 44 & 73) & \quad (37 & 2 & 80) \\
(92 & 32 & 26) & \quad (40 & 15 & 71) & \quad (38 & 13 & 68) \\
(2 & 97 & 43) & \quad (32 & 32 & 32) & \quad (12 & 91 & 7)
\end{align*}
\]
Let $m = 3$, and let the three sets be denoted by $V_1$, $V_2$, and $V_3$. The length of each vector, $p$, equals 3, and $n = 4$, and let us specify $c$ as the sum of the entries of a vector, ie, $c(u) = \sum_{i=1}^{p} u_i$.

$$
\begin{array}{ccc}
\text{V}_1 & \text{V}_2 & \text{V}_3 \\
(12 \ 91 \ 7) & (47 \ 31 \ 12) & (83 \ 3 \ 37) \\
(54 \ 29 \ 64) & (5 \ 44 \ 73) & (37 \ 2 \ 80) \\
(92 \ 32 \ 26) & (40 \ 15 \ 71) & (38 \ 13 \ 68) \\
(2 \ 97 \ 43) & (32 \ 32 \ 32) & (12 \ 91 \ 7) \\
\end{array}
$$

A particular $m$-tuple could consist of the second vector of $V_1$ ((54 29 64)), the first vector of $V_2$ ((47 31 12)), and the fourth vector of $V_3$ ((12 91 7)), coming out at: (54 91 64).
1 Relevance

2 Our problem: MVA
   - On the cost function
   - Heuristics for MVA
   - An instance

3 Results
   - Analysis of Heuristics
     - Monotone and Submodular Case
   - Hardness
   - Polynomial Special case
   - Questions
A wafer

Emerging Technology
Through Silicon Vias (TSV) based
Three-Dimensional Stacked
Integrated Circuits (3D-SIC)

Benefits
• smaller footprint
• higher interconnect density
• higher performance
• lower power consumption
  compared to planar IC’s
Stacking wafers

From lot 1  
From lot 2  
From lot 3  
Stacking  
Stack
Yield optimization: bad dies and good dies

(0,..,0,1,1,0,...0,1,0,...,0,1,0,...,0,1,0,...,0,1,0,1)

Defect map
Yield optimization: superimposing dies

Defect map of resulting stack:
(0,...,0,1,1,0,...0,1,0,...,0,1,0,...,0,1,0,...,0,1,0,1)

Yield = no. of zeros in defect map vector
Yield optimization: an example

Total number of bad dies in stack 1 + stack 2 = 23

Trivikram Dokka, Yves Crama, Frits Spieksma
Yield optimization: an example

Total number of bad dies in stack 1 + stack 2 = 17

Stack 1

Stack 2
Previous work

Yield optimization is a special case of MVA

Observe that in the yield optimization application, all vectors are \( \{0, 1\} \)-vectors, and that the cost-function \( c \) is additive, ie, \( c(u) = \sum_{i=1}^{p} u_i \).

Instances from practice may have \( m = 10, \ n = 75, \) and \( p = 1000 \).

We refer to this special case of MVA as the \textit{Wafer-to-Wafer Integration} problem (WWI).
### Cost Functions

**Monotonicity**

If \( u, v \in \mathbb{Z}_+^p \) and \( u \leq v \), then \( 0 \leq c(u) \leq c(v) \).

**Subadditivity**

If \( u, v \in \mathbb{Z}_+^p \), then \( c(u \lor v) \leq c(u) + c(v) \).

**Submodularity**

If \( u, v \in \mathbb{Z}_+^p \), then \( c(u \lor v) + c(u \land v) \leq c(u) + c(v) \).

**Modularity**

If \( u, v \in \mathbb{Z}_+^p \), then \( c(u \lor v) + c(u \land v) = c(u) + c(v) \).
Heuristics

- **Sequential Heuristics**
  - **Sequential Heuristic** ($H^{seq}$): Solve a bipartite assignment problem between $H_{i-1}$ and $V_i$. Let $H_i$ be the resulting assignment for $V_1 \times \ldots \times V_i$; $i = 2, \ldots, m$. Return $H_m$.
  - **Heavy Heuristic** ($H^{heavy}$): Rearrange the sets such that $c(V_1)$ is the heaviest. Apply $H^{seq}$.

- **Hub Heuristics**
  - **Single-hub Heuristic** ($H^{shub}$): Choose a hub $h \in \{1, \ldots, m\}$. Solve an assignment problem between $V_h$ and $V_i$ (call the resulting solutions $M_{hi}$). Construct a feasible solution by combining the solutions $M_{hi}$.
  - **Multi-hub Heuristic** ($H^{mhub}$): Apply $H^{shub}$ for each possible choice of hub and output the best solution among all.
Example

```
V_1   V_2   V_3
00    00    10
01    10    01
```

Trivikram Dokka, Yves Crama, Frits Spieksma
Example: the optimum
When $c$ is monotone and subadditive: every heuristic is an $m$-approximation algorithm.
Results Overview

- When $c$ is monotone and subadditive: every heuristic is an $m$-approximation algorithm.

- When $c$ is monotone and submodular, both the sequential heuristic, as well as the multi-hub heuristic have a worst-case ratio of $\frac{1}{2}m$. 
Results Overview

- When $c$ is monotone and subadditive: every heuristic is an $m$-approximation algorithm.

- When $c$ is monotone and submodular, both the sequential heuristic, as well as the multi-hub heuristic have a worst-case ratio of $\frac{1}{2}m$.

- When $c$ is additive, the Heaviest-first has a better performance:
  \[\rho_{\text{heavy}}(m) \leq \frac{1}{2}(m + 1) - \frac{1}{4}\ln(m - 1).\]
When $c$ is monotone and subadditive: every heuristic is an $m$-approximation algorithm.

When $c$ is monotone and submodular, both the sequential heuristic, as well as the multi-hub heuristic have a worst-case ratio of $\frac{1}{2}m$.

When $c$ is additive, the Heaviest-first has a better performance: 

$$\rho_{\text{heavy}}(m) \leq\frac{1}{2}(m + 1) - \frac{1}{4}\ln(m - 1).$$

WWI-3 is APX-hard.
Results

Results Overview

- When $c$ is monotone and subadditive: every heuristic is an $m$-approximation algorithm.

- When $c$ is monotone and submodular, both the sequential heuristic, as well as the multi-hub heuristic have a worst-case ratio of $\frac{1}{2}m$.

- When $c$ is additive, the Heaviest-first has a better performance: $\rho^\text{heavy}(m) \leq \frac{1}{2}(m + 1) - \frac{1}{4}\ln(m - 1)$.

- WWI-3 is APX-hard.

- WWI with fixed $p$ is solvable in polynomial time.
Overview of results

Monotone
- ratio: unbounded

Monotone and Submodular
- ratio: $O(m/2)$

Monotone and Modular (Additive)
- ratio: $O(m/2 - \ln(m)/4)$

Submodular
- ratio: unbounded
Monotone and Submodular Case
Analysis of $H^{seq}$

Notation:

- $c(H_r) =$ value of partial solution restricted to $V_1 \times \ldots V_r$,
- $c(A_{m-2,m}) =$ value of the partial solution corresponding to an optimal assignment between $H_{m-2}$ and $V_m$,
- $c(V_i) =$ total weight of the set $V_i$, $i = 1, \ldots, m$. 
Recall: $A_{m-2,m} = \text{solution of optimal assignment between } V_m \text{ and } H_{m-2}$

Case 1: $c(V_{m-1}) \leq \frac{1}{2} c_m^{OPT}$

\[
c(H_m) \leq c(A_{m-2,m}) + c(V_{m-1})
\]

\[
c(A_{m-2,m}) \leq \frac{1}{2} (m - 1) c^{OPT}(W) \leq \frac{1}{2} (m - 1) c_m^{OPT}
\]

where $W = V_1 \times \ldots \times V_{m-2} \times V_m$

\[
c(H_m) \leq \left( \frac{m - 1}{2} + \frac{1}{2} \right) c_m^{OPT} = \frac{m}{2} c_m^{OPT}.
\]
Analysis of Heuristic $H^{seq}$

- $M_{m-1,m} = $ solution of optimal assignment between $V_{m-1}$ and $V_m$

**Case 2: $c(V_{m-1}) \geq \frac{1}{2} c^{OPT}_m$**

\[
\begin{align*}
c(H_m) & \leq c(H_{m-1}) + c(M_{m-1,m}) - c(V_{m-1}) \\
& \leq \frac{m - 1}{2} \cdot c^{OPT}_{m-1} + c^{OPT}_m - \frac{1}{2} \cdot c^{OPT}_m \\
& \leq \left( \frac{m - 1}{2} + \frac{1}{2} \right) c^{OPT}_m \\
& \leq \frac{m}{2} c^{OPT}_m
\end{align*}
\]
Theorem

When the cost-function $c$ is monotone and submodular, the sequential heuristic has a performance ratio of $\rho_{seq}(m) = \frac{1}{2}m$. This bound is tight even when the input of MVA-m is restricted to binary vectors.

Tight example

- $c(u) = f(\sum_{i=1}^{p} u_i)$, where $f : \mathbb{R} \rightarrow \mathbb{R}$ is defined by $f(x) = x$ when $x \leq 2$, and $f(x) = 2$ when $x \geq 2$.
- $f$ is monotone nondecreasing and concave, and $c$ is monotone and submodular.
- $p = n = m - 1$, $V_i = \{e_i, 0, \ldots, 0\}$ for $i = 1, \ldots, m$, where $e_i$ is the $i^{th}$ unit vector.
- $c(H_m) = m$ and $c_m^{OPT} = 2$. 
Theorem

WWI-3 is APX-hard even when all vectors in $V_1 \cup V_2 \cup V_3$ are $\{0, 1\}$ vectors with exactly two nonzero entries per vector.

Sketch

L-reduction from 3-bounded MAX-3DM to WWI-3.
**Theorem**

*Binary MVA can be solved in polynomial time for each fixed p.*

**Binary MVA - MIP**

A mixed integer formulation of MVA with variables:

For each $t = 1, \ldots, 2^p$,

$$x_t = \text{number of } m\text{-tuples of type } t \text{ in the assignment},$$

For each $i = 1, \ldots, m; j = 1, \ldots, n; t = 1, \ldots, 2^p$,

$$z_{jt}^i = 1 \text{ if } v_{ij} \text{ is assigned to an } m\text{-tuple of type } t.$$
Binary inputs and fixed $p$ case

**Binary MVA - MIP**

\[
\begin{align*}
\text{min} & \quad \sum_{t=1}^{2^p} c(b_t) x_t \\
\sum_{j: \ b_t \geq v_{ij}} z^i_{jt} &= x_t \quad \text{for each } t, i \\
\sum_{t: \ b_t \geq v_{ij}} z^i_{jt} &= 1 \quad \text{for each } j, i \\
x_t \text{ integer} & \quad \text{for each } t \\
z^i_{jt} \geq 0 & \quad \text{for each } j, t, i.
\end{align*}
\]

**Claim:** Binary MVA - MIP can be solved in polynomial time for every fixed $p$. 
Future work and extensions

Questions

1. What is the exact approximation ratio of the multi-hub heuristic in case of additive costs? We know that it lies between $m/4$ and $m/2$.

2. What is the exact approximation ratio of the heaviest-first sequential heuristic in case of additive costs? We know that it lies between $\Omega(\sqrt{m})$ and $O(m - \ln m)$.

3. Does there exist a polynomial-time algorithm with constant (i.e., independent of $m$) approximation ratio for MVA-$m$?

4. Can we design practical exact algorithm based on Binary MVA - MIP for reasonable $n,p$ and $m$?
MAPSP2015 takes place in La Roche, 2015, June 8 - June 12
MAPSP2015 takes place in La Roche, 2015, June 8 - June 12

see www.mapsp2015.com
MAPSP2015 takes place in La Roche, 2015, June 8 - June 12
THANKS!