A Top-Down Synthesis Methodology for Flow-Based Microfluidic Biochips Considering Valve-Switching Minimization

2013 ACM International Symposium on Physical Design

Kai-Han Tseng, Sheng-Chi You, Jhe-Yu Liou and Tsung-Yi Ho

http://eda.csie.ncku.edu.tw
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Outline

Introduction

↓

Problem Formulation

↓

Algorithms

↓

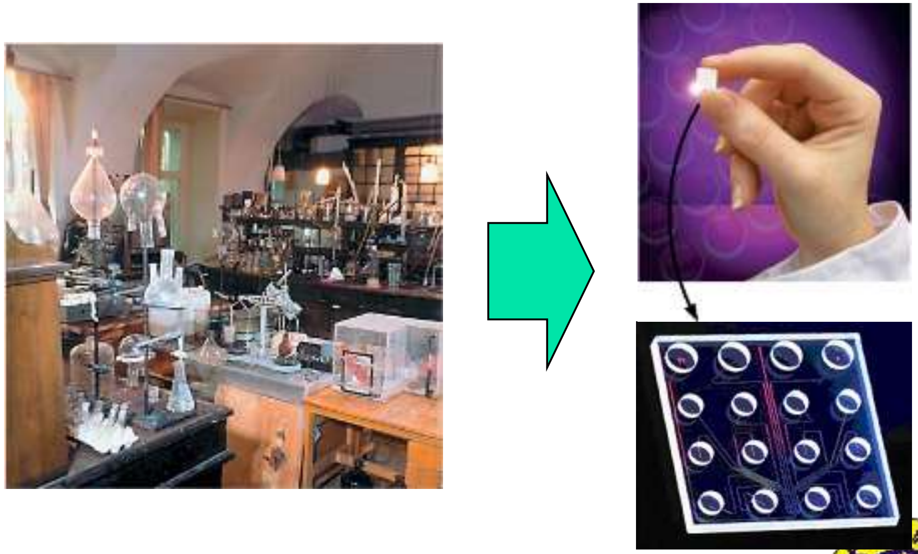
Experimental Results

↓

Conclusions

What is Biochips?

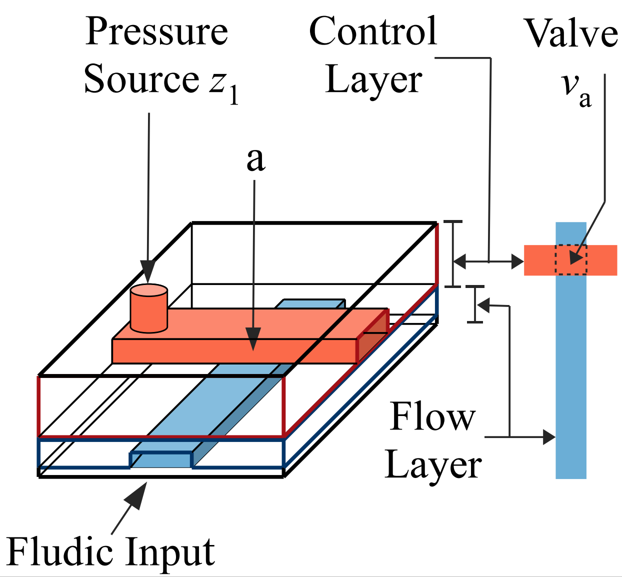
- General definition
  - A chip with a small solid platform made of glass, plastic, or membrane which deals with the behavior, precise control and manipulation of fluids that are constrained to a small scale
- Functionality
  - Analysis, reaction, or detection of biological samples (DNA or human blood)
- Application
  - Clinical diagnostics
  - Environmental monitoring
  - Automated drug discovery
  - Cell culturing
  - Protein crystallization

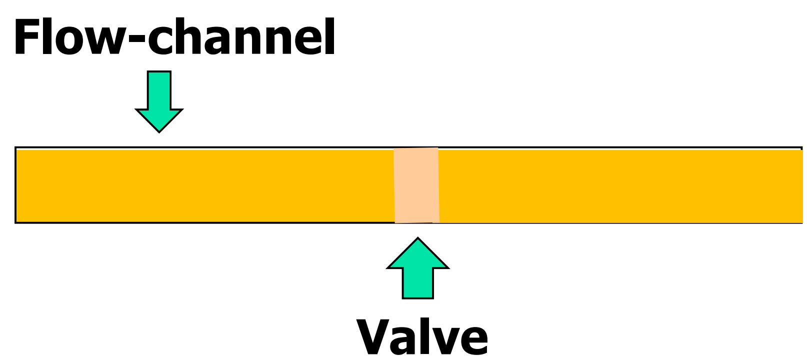


Classification of Biochips

- Biochips
  - Microarray
    - DNA chip
    - Protein chip
  - Microfluidic biochips
    - Continuous-flow
    - Droplet-based

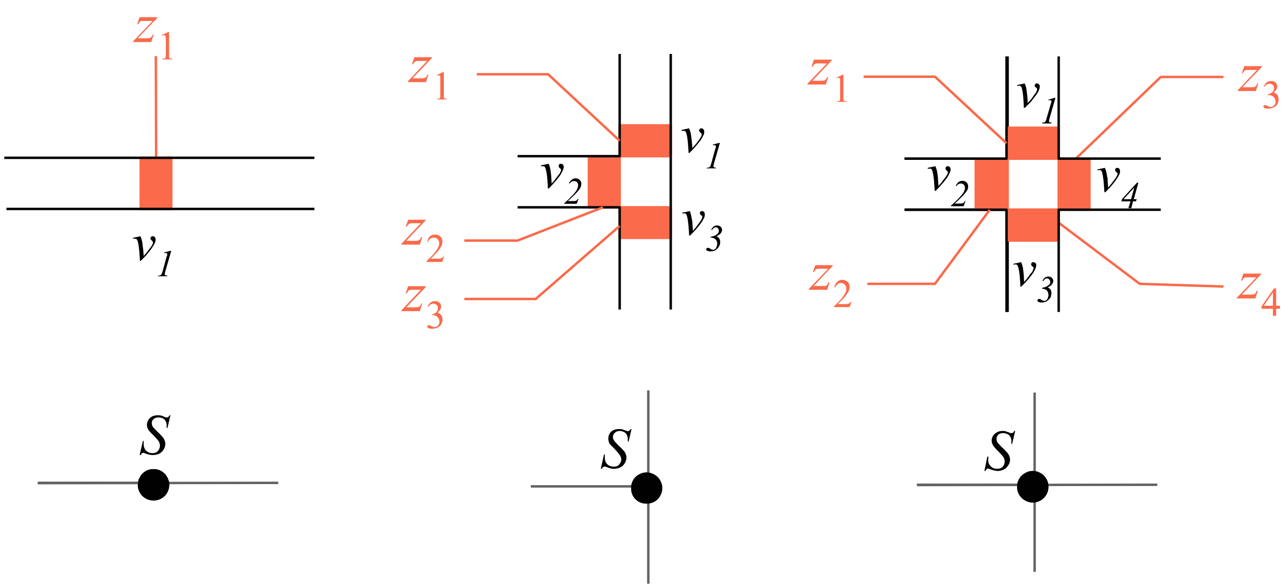
Flow-based microfluidic biochips





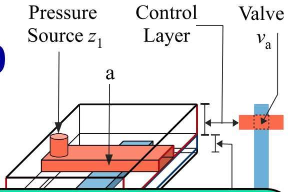
Valve: The Basic Unit of Flow-based Microfluidic Biochips

- Functional component
  - Valves can be combined to form more complex units, e.g., latches, switches, mixers, multiplexers, micropumps
- Reliability
  - The open and close activities (**valve-switching**) for a valve is limited



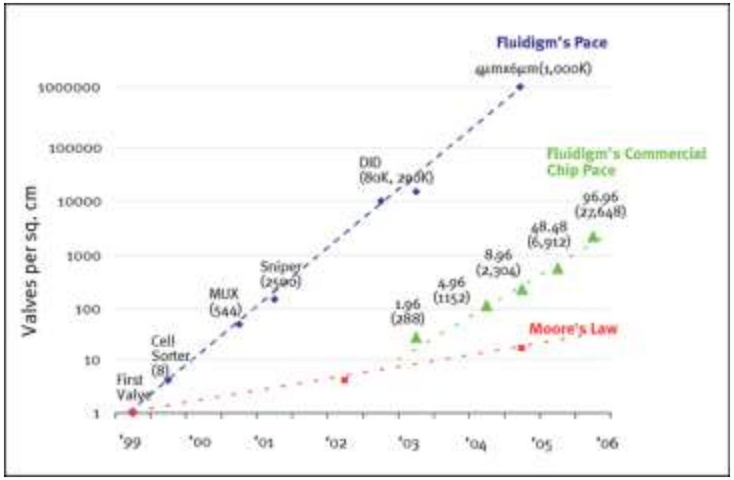
The Need of CAD Support

- Applications become more complicated
  - Large-scale bioassays
  - Multiple and concurrent assay operations on a biochip



A top-down synthesis methodology is needed

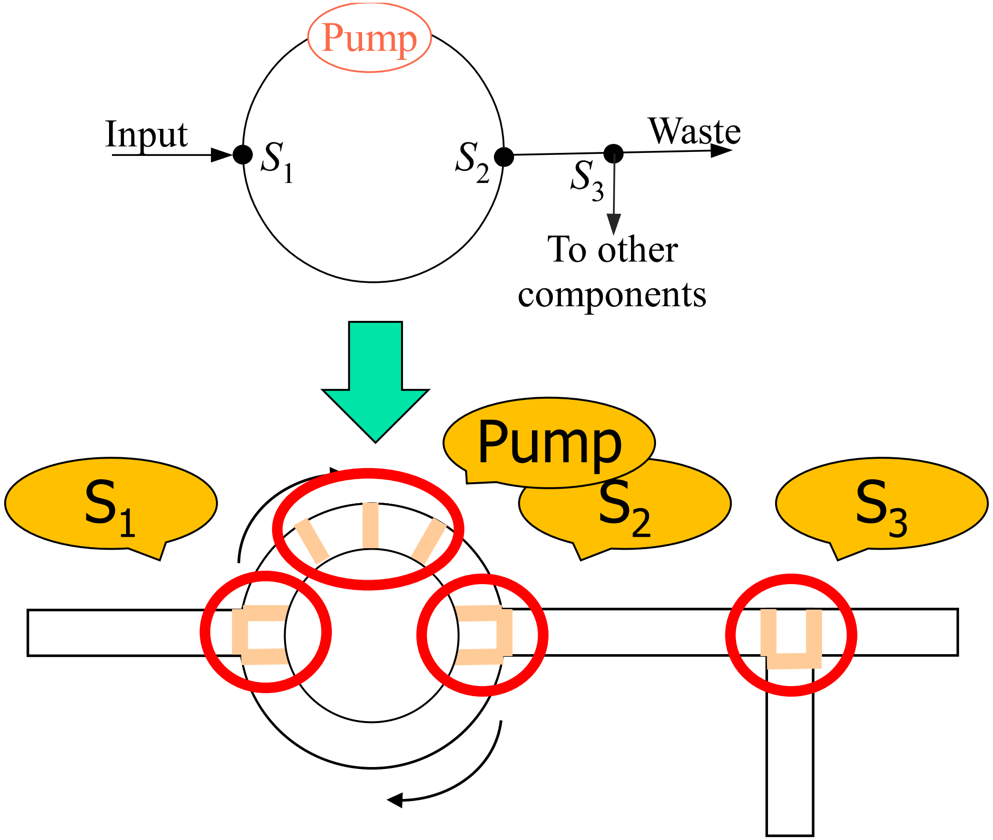
- Current methodologies
  - Manual
  - Full-custom



Source: Fluidigm

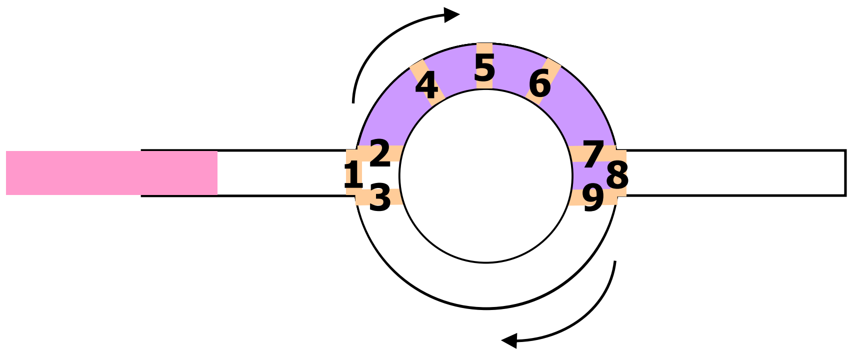
Component Model : Mixer

Microfluidic mixer



Motivation

Microfluidic mixer



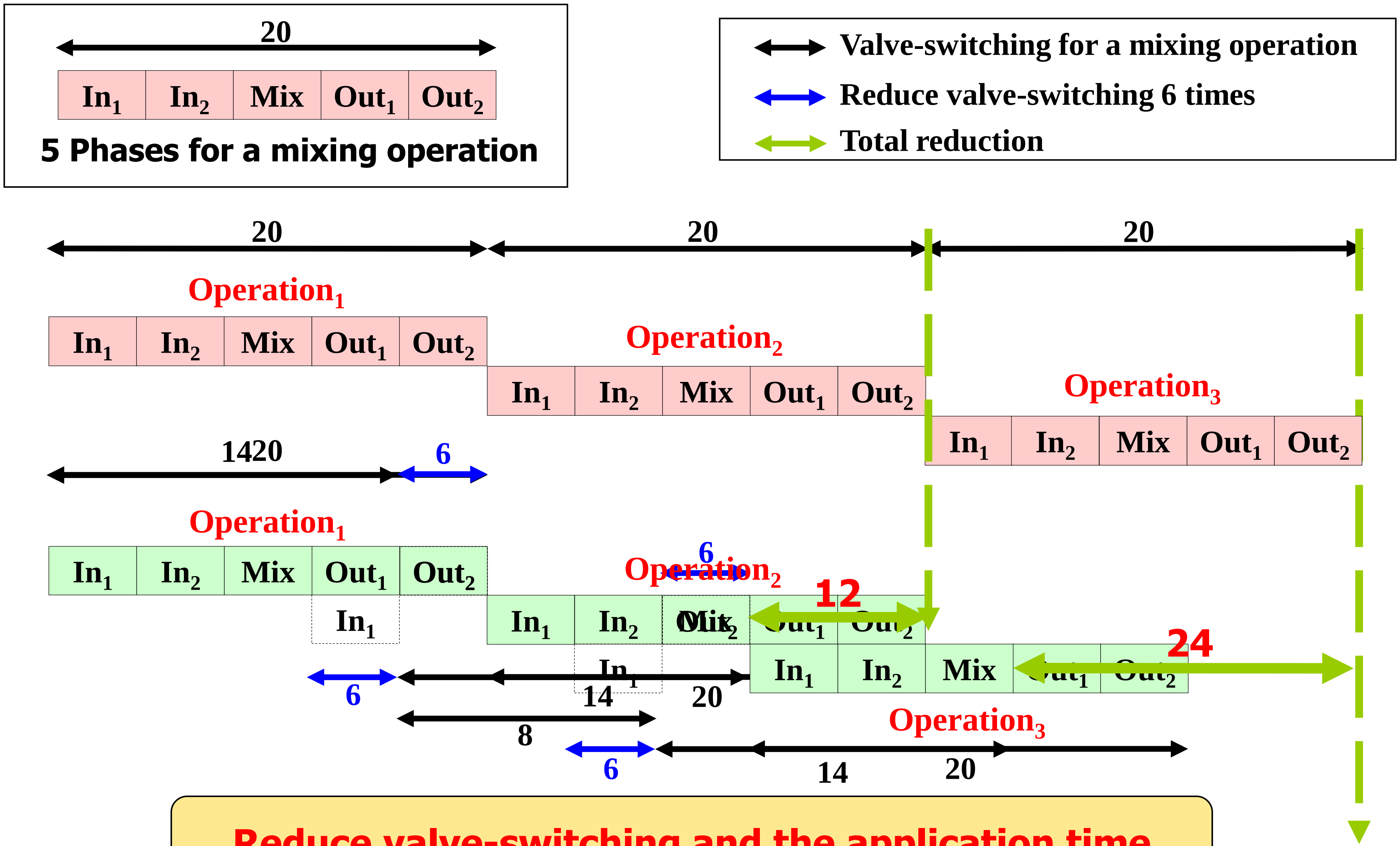
Open the valve

Close the valve

Valve-switching： 20 + Mixing

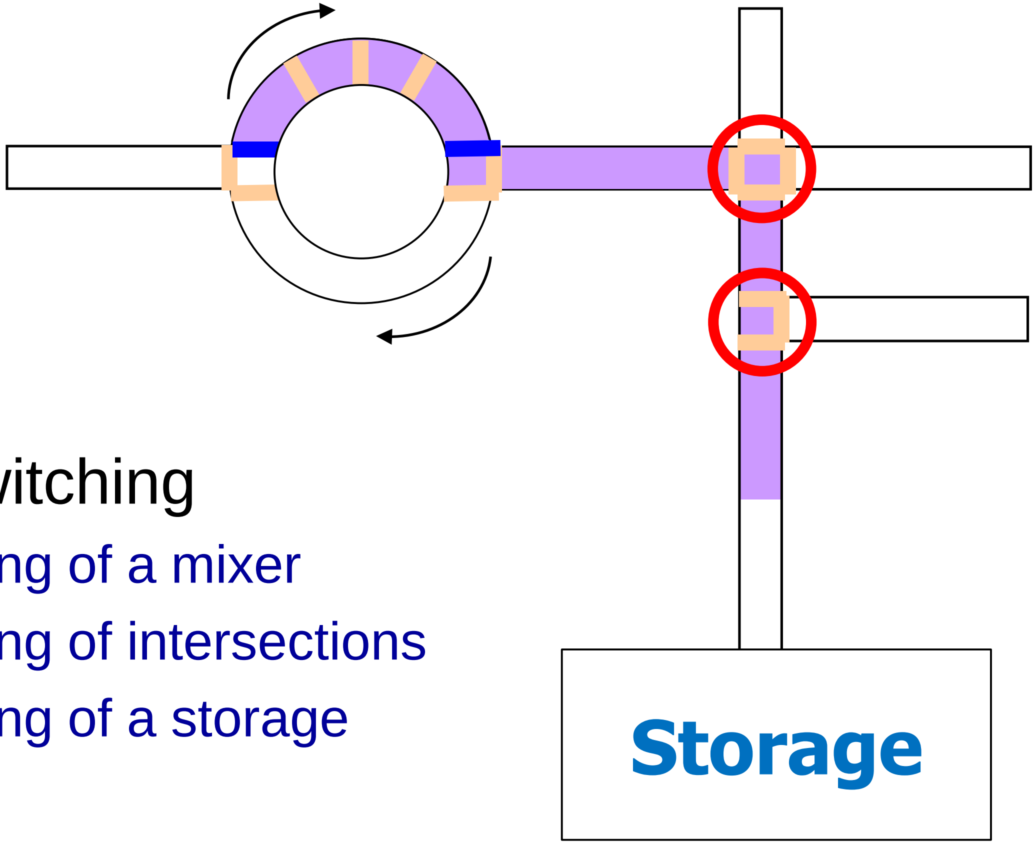
Phase	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
1. Ip1	0	0 1	0	0	0	0	0 1	0	0
2. Ip2	0	0	0 1	0	0	0	0	0	0 1
3. Mix	0 1	0	0	0 Mix	0 Mix	0 Mix	0	0 1	0
4. Op1	0	0 1	0	0	0	0	0 1	0	0
5. Op2	0	0	0 1	0	0	0	0	0	0 1

Group Continuous Operations



Reduce valve-switching and the application time

Transportation Issue



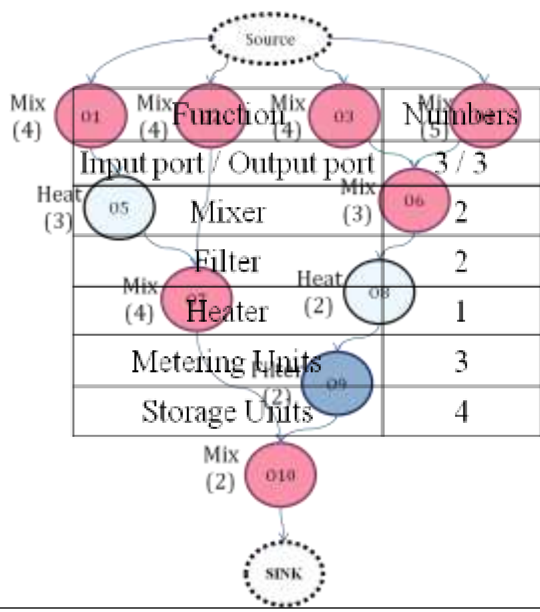
- Total valve-switching
  - Valve-switching of a mixer
  - Valve-switching of intersections
  - Valve-switching of a storage

Outline

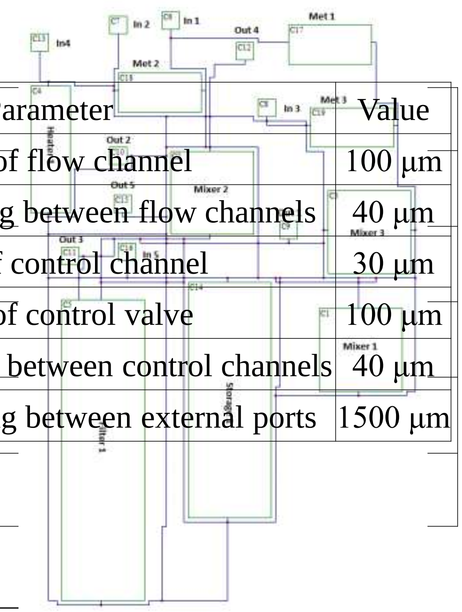
- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

Problem Formulation

- **Input:** A biochemical application modeled as a sequential graph and a component library
- **Objective:** Synthesize a flow-based microfluidic biochip such that the valve-switching amount and application completion time are minimized
- **Constraint:** Resource constraints and design rules



Function	Numbers
Input port / Output port	3 / 3
Mixer	2
Filter	2
Heater	1
Metering Units	3
Storage Units	4



Component
Mixer
Filter
Detector
Separator
Heater

Parameter	Value
Width of flow channel	100 μm
Minimum spacing between flow channels	40 μm
Width of control channel	30 μm
Width of control valve	100 μm
Minimum spacing between control channels	40 μm
Minimum spacing between external ports	1500 μm

Outline

Introduction

↓

Problem Formulation

↓

Algorithms

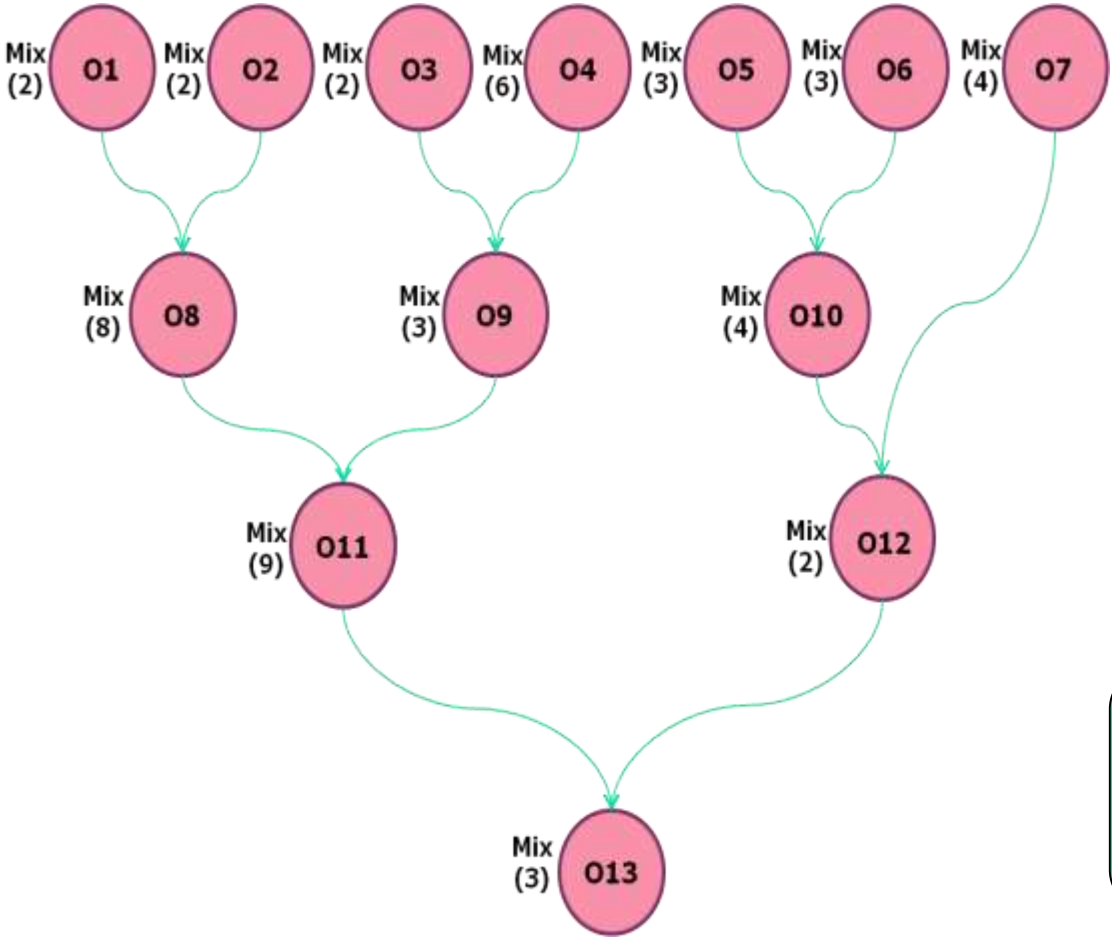
↓

Experimental Results

↓

Conclusions

Input



Application Graph

Component	Phases
Mixer	Ip_1 / Ip_2 / Mix / Op_1 / Op_2
Filter	Ip / Filter / Op_1 / Op_2
Detector	Ip / Detect / Op
Separator	Ip_1 / Ip_2 / Separat/ Op_1 / Op_2
Heater	Ip / Heat / Op

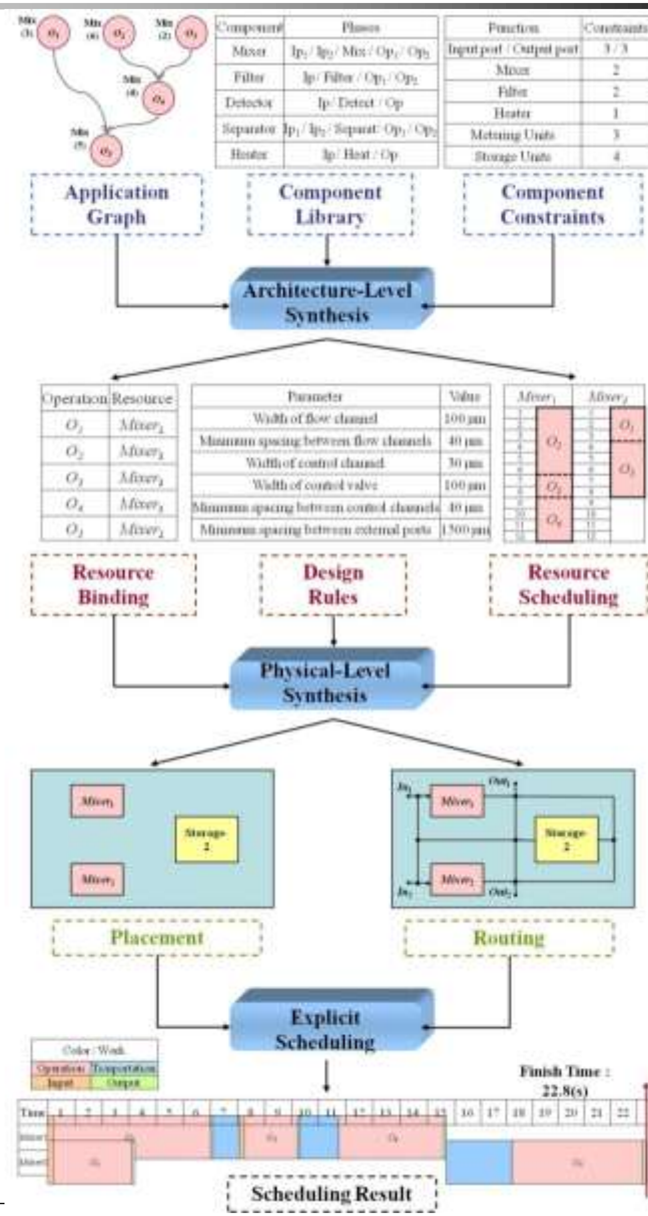
Component Library

Mixer : 3

Resource Constraint

Top-Down Synthesis Methodology

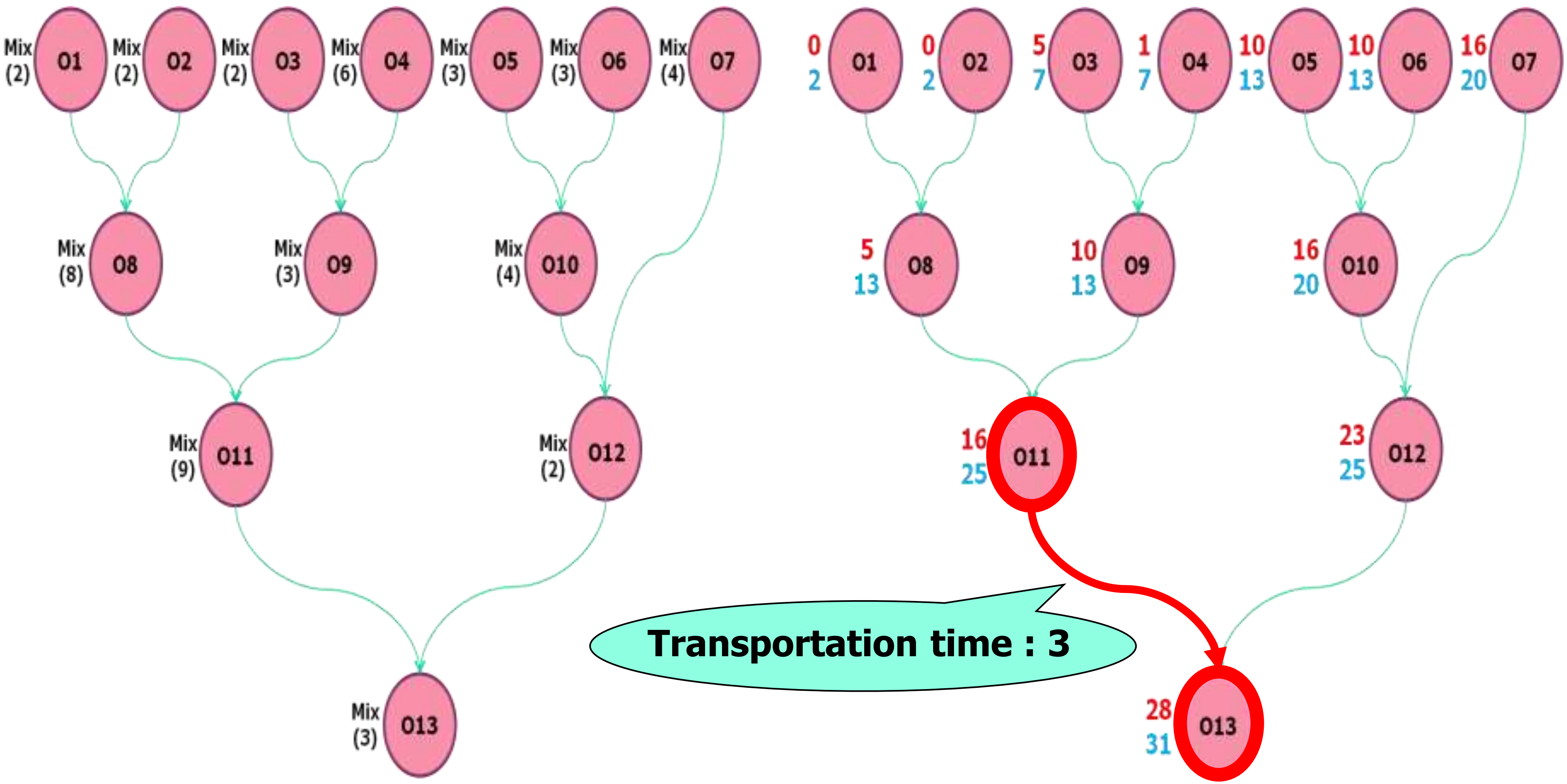
- **Architecture-level synthesis**
  - **Set-based minimum cost maximum flow (SMCMF)**
    - Resource binding
    - Resource scheduling
- **Physical-level synthesis**
  - **Incremental cluster expansion (ICE)**
    - Placement
    - Routing
- **Explicit scheduling**
  - **List-scheduling based approach**
    - Total valve-switching amount
    - Application completion time



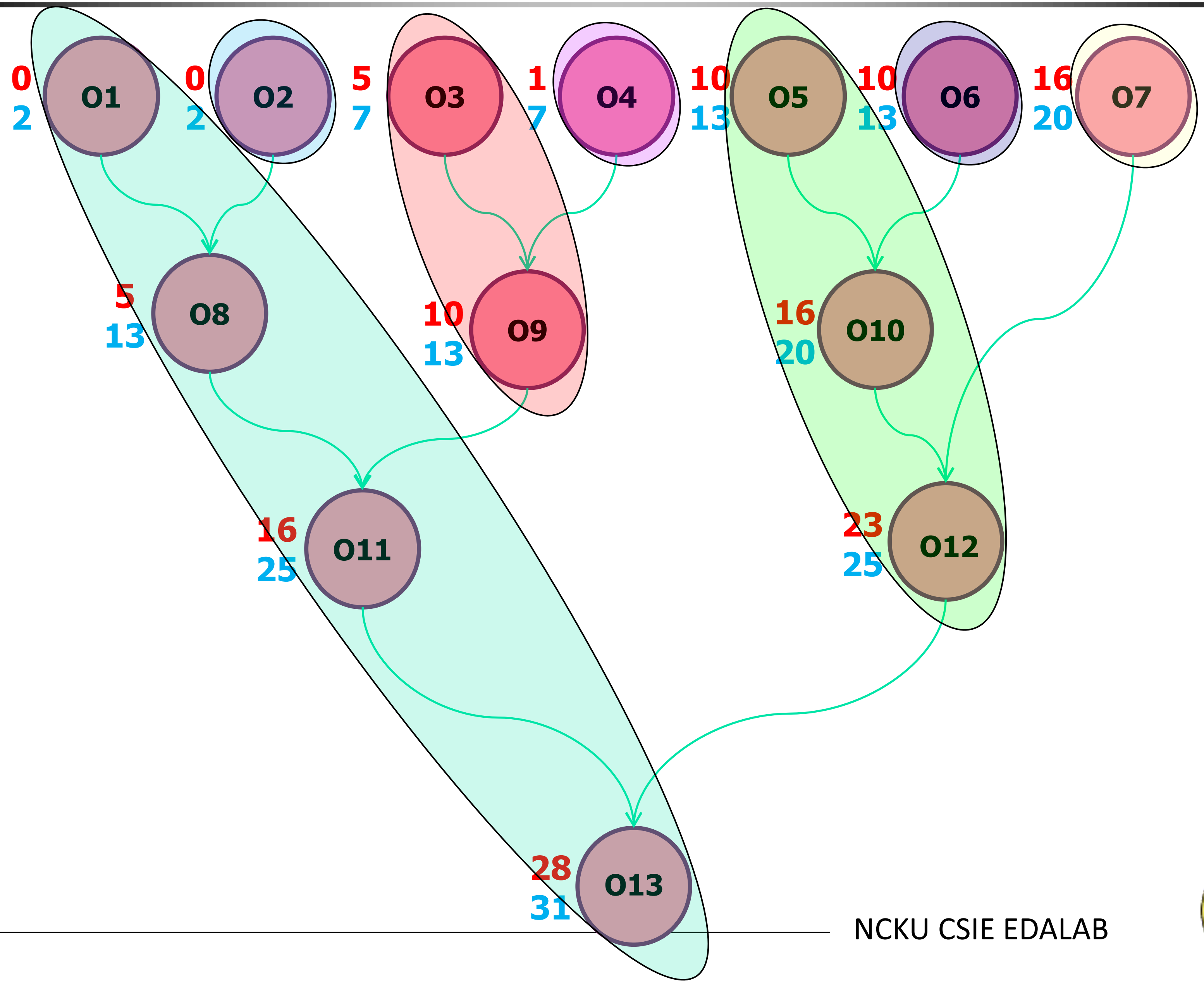
Set-based Minimum Cost Maximum Flow

- Set-based
  - **Group continuous operations in a set** to reduce the valve-switching activities
- Maximum Flow
  - Each flow path represents a component. Our goal here is to **maximize the component parallelization**.
- Minimum Cost
  - Find a way that not only satisfies the parallelization but also **minimize the application completion time**

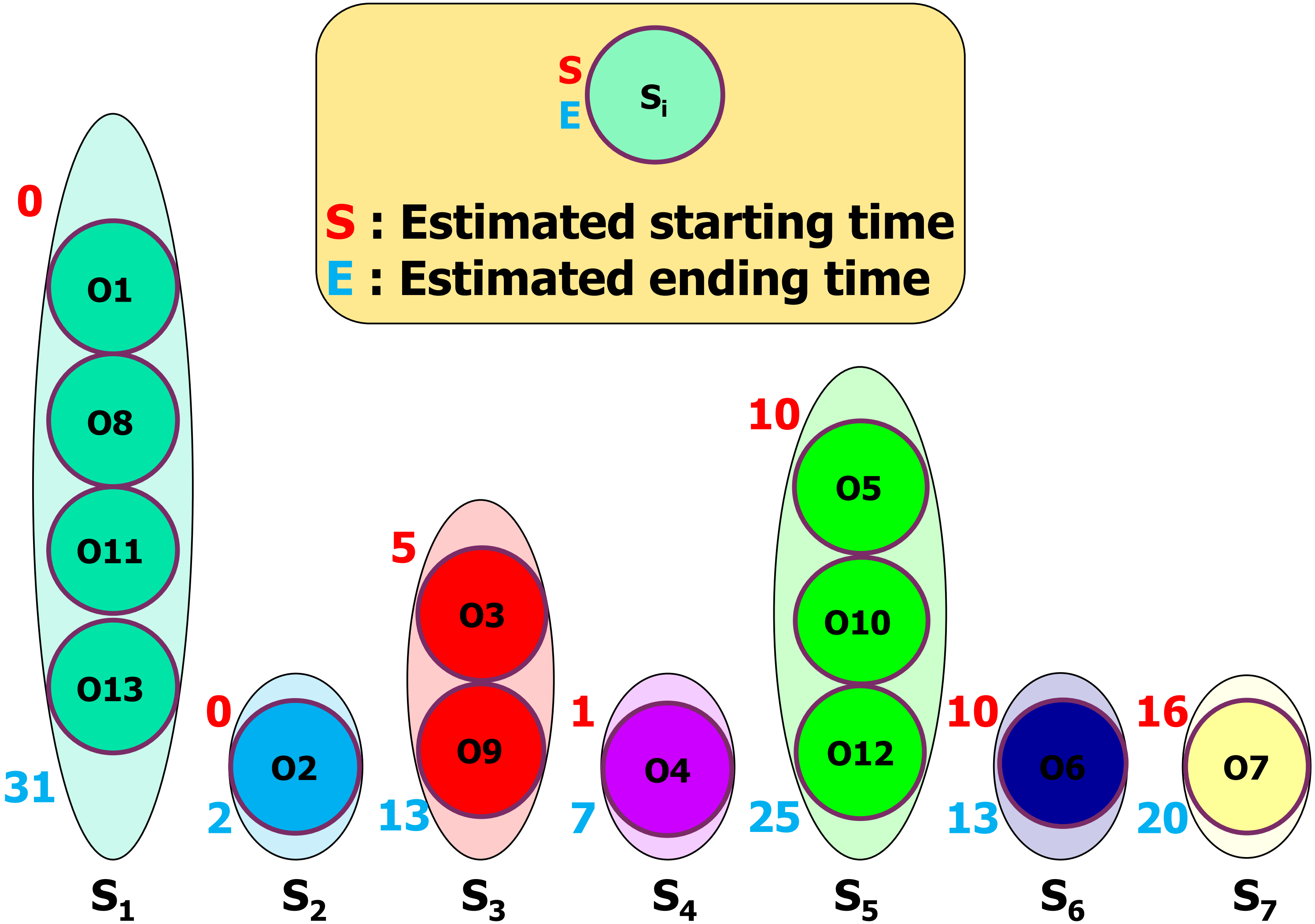
Time Evaluation



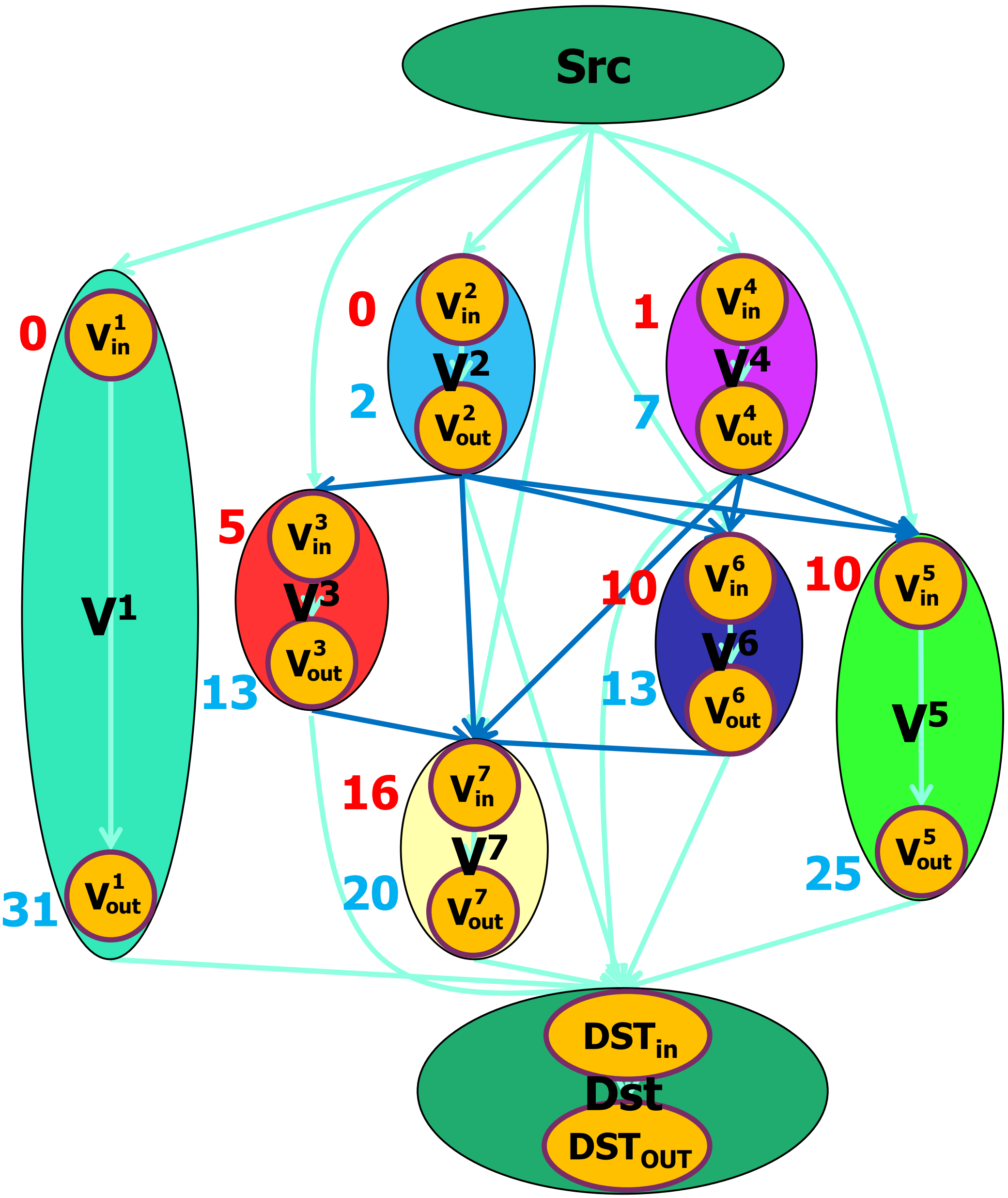
Find Continuous Sets



Continuous Sets

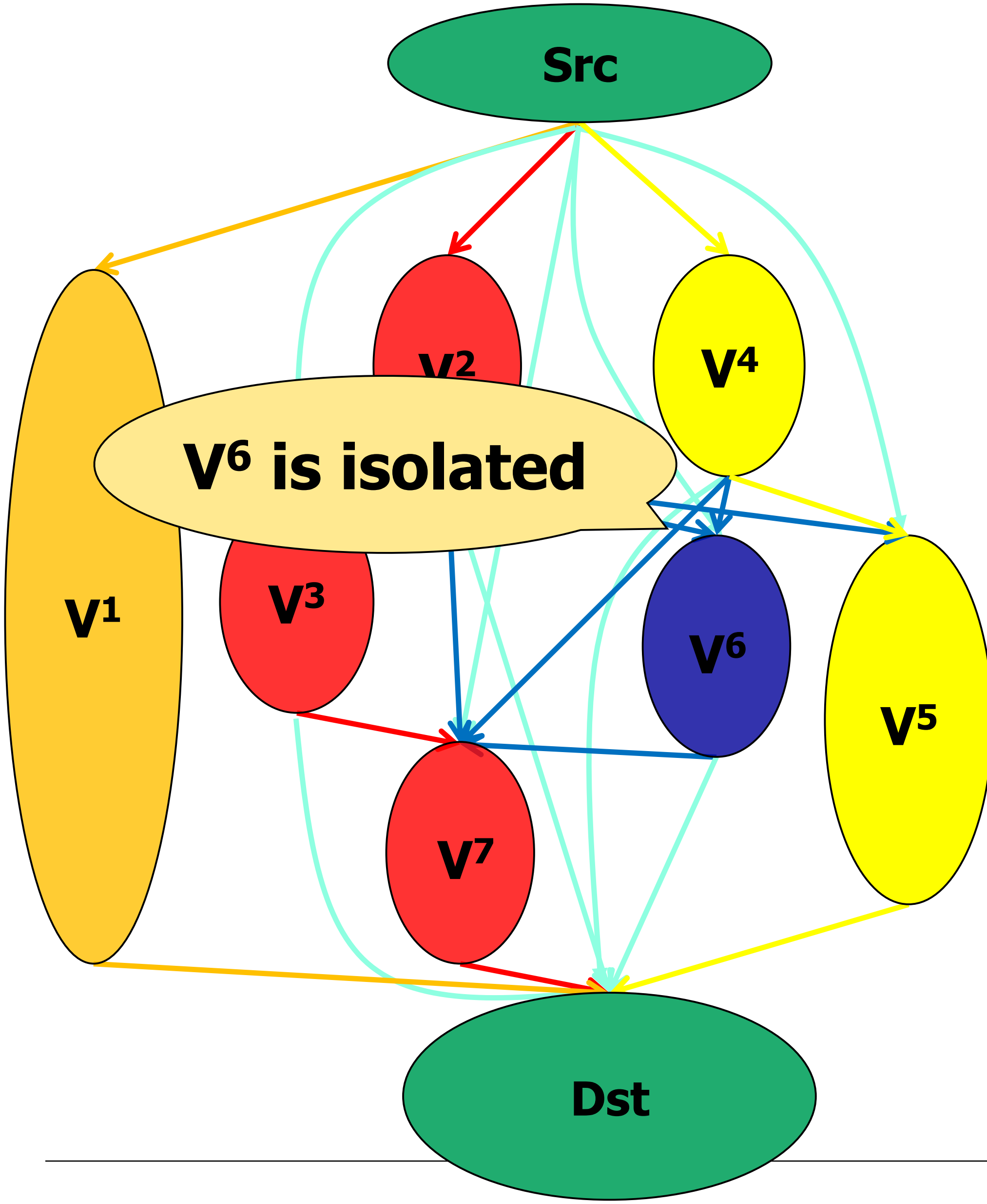


Build the Flow Network

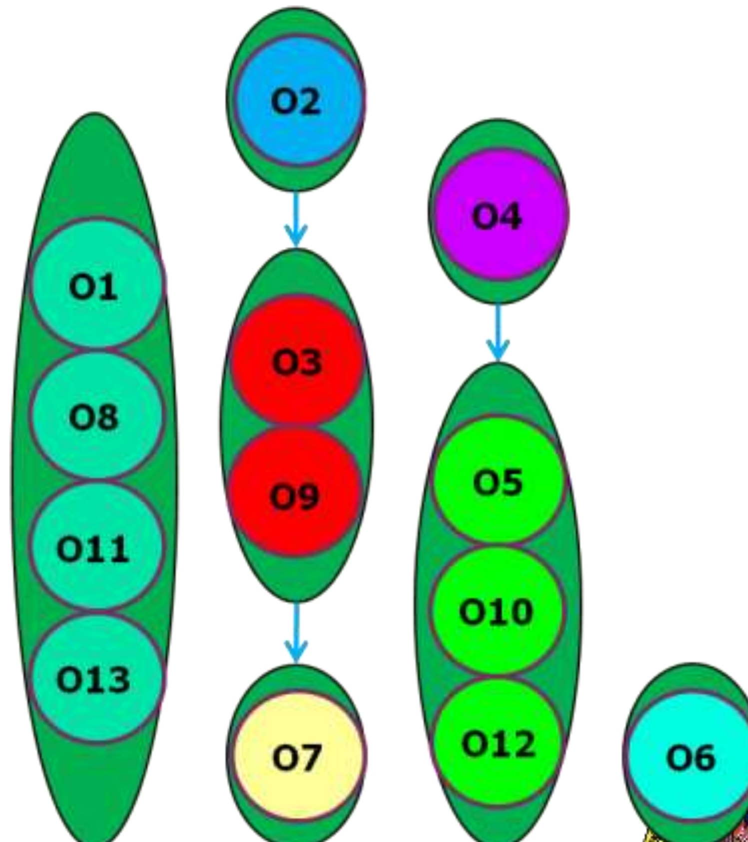


- Step1
  - Create Src and Dst vertex
  - Create a vertex Vi ϵ V for each Si
- Step2
  - Separate $V^i \rightarrow (V^i_{in}, V^i_{out})$
  - Create a directed edge $V^i_{in} \rightarrow V^i_{out}$, cost=0 and capacity=1
- Step3
  - Separate Dst $\rightarrow$ (Dst_{in}, Dst_{out})
  - Create a directed edge $Dst_{in} \rightarrow Dst_{out}$, cost=0 and capacity=3 (given component numbers)
- Step4
  - Create directed edges Src $\rightarrow V^i_{in}$ and $V^i_{out} \rightarrow Dst_{in}$, cost = 0 and capacity=1
  - For each V^i_{out} and V^j_{in}, if V^i_{out} + T (transportation time) $\leqq V^j_{in}$, create a directed edge $V^i_{out} \rightarrow V^j_{in}$, cost=f(x) and capacity=1. Here, x is defined as $|(V^i_{out} - V^i_{in})|$ and f is proportional to x

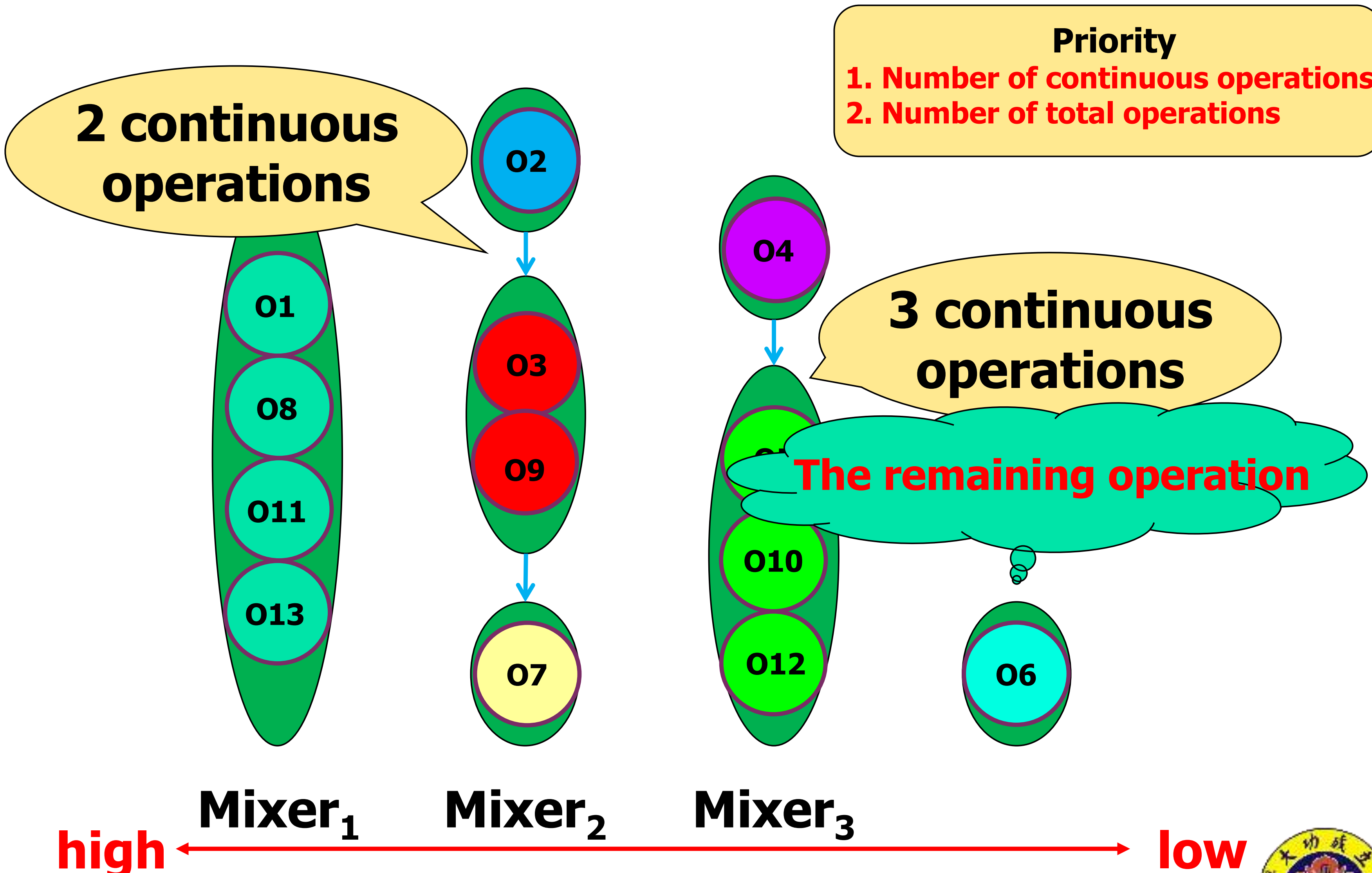
Apply Minimum Cost Maximum Flow



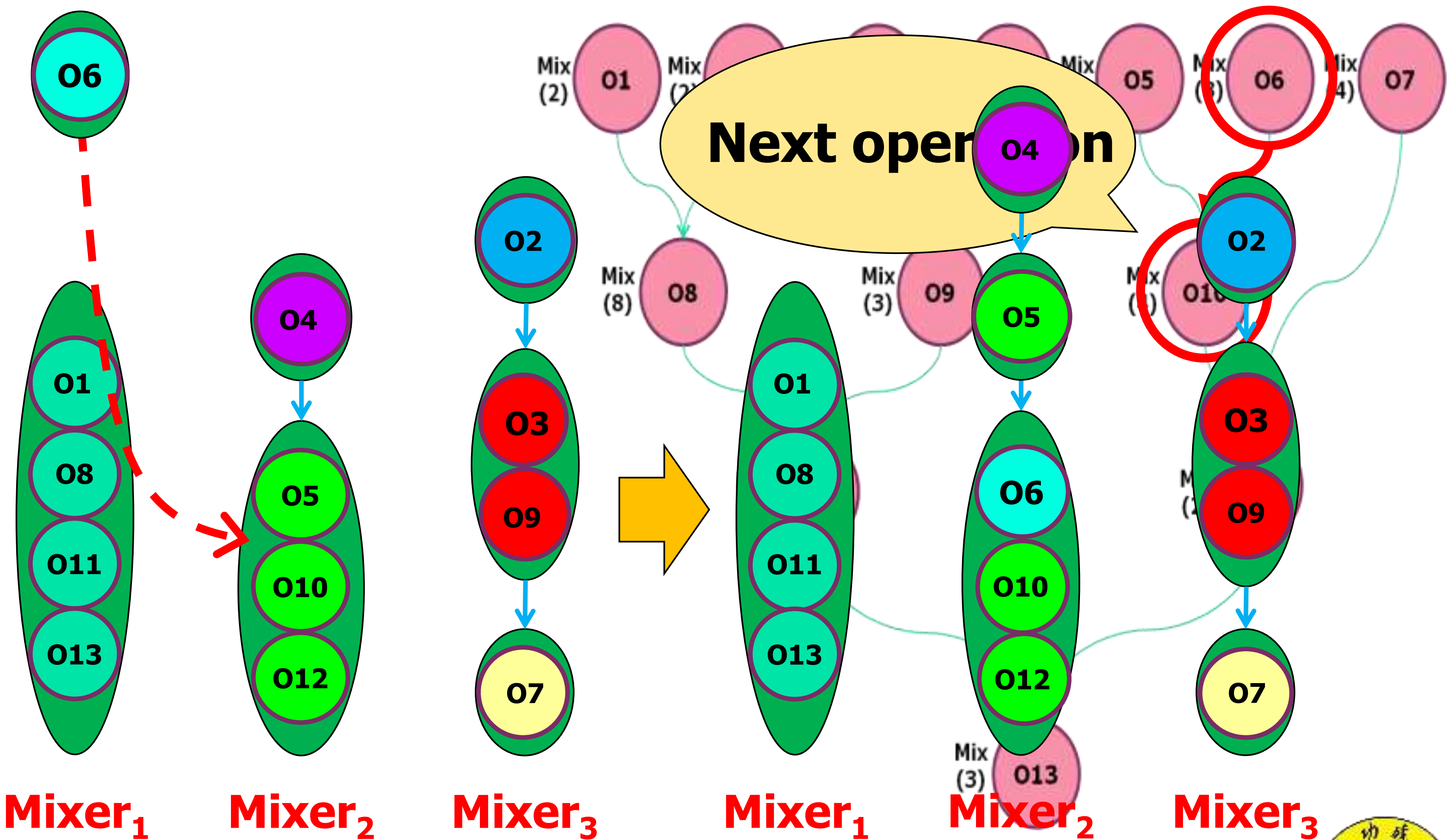
Find 4 concatenated-sets



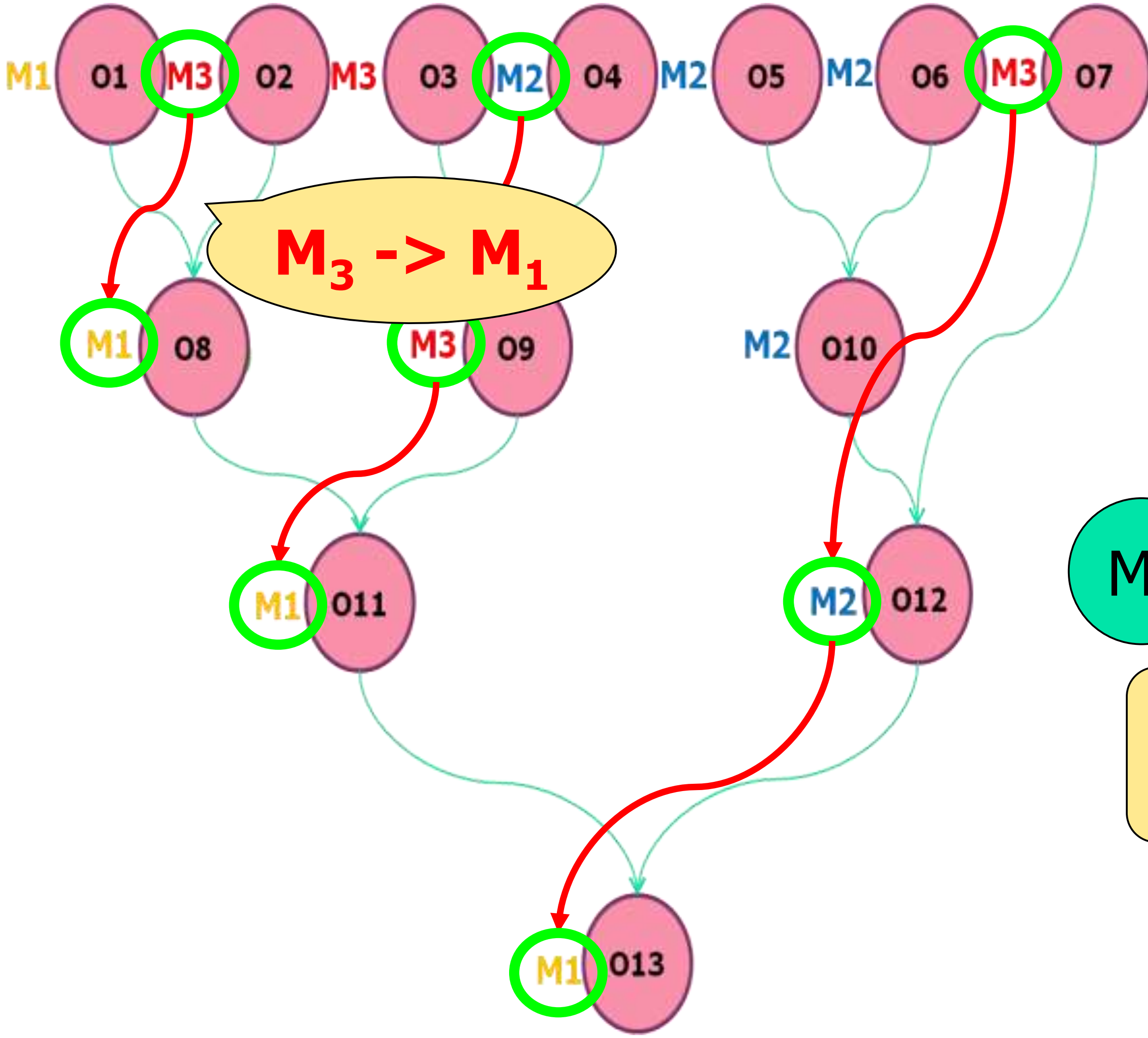
Sort by the Priority

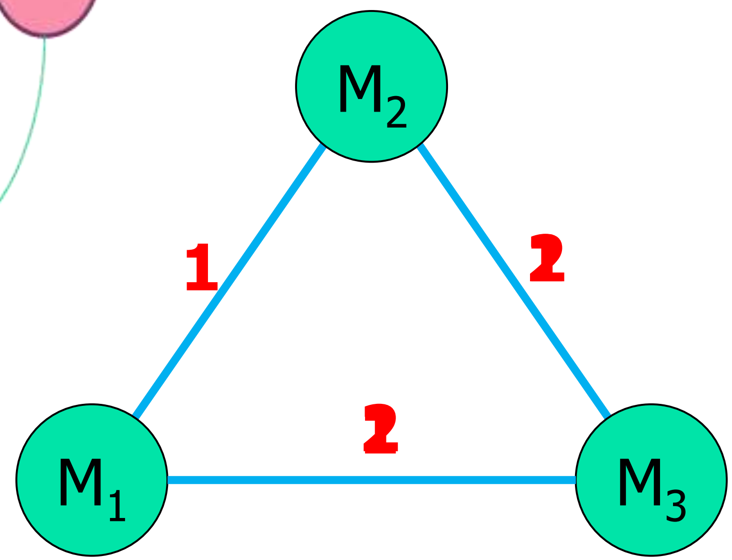


Insert the Remaining Operation



Relational Graph

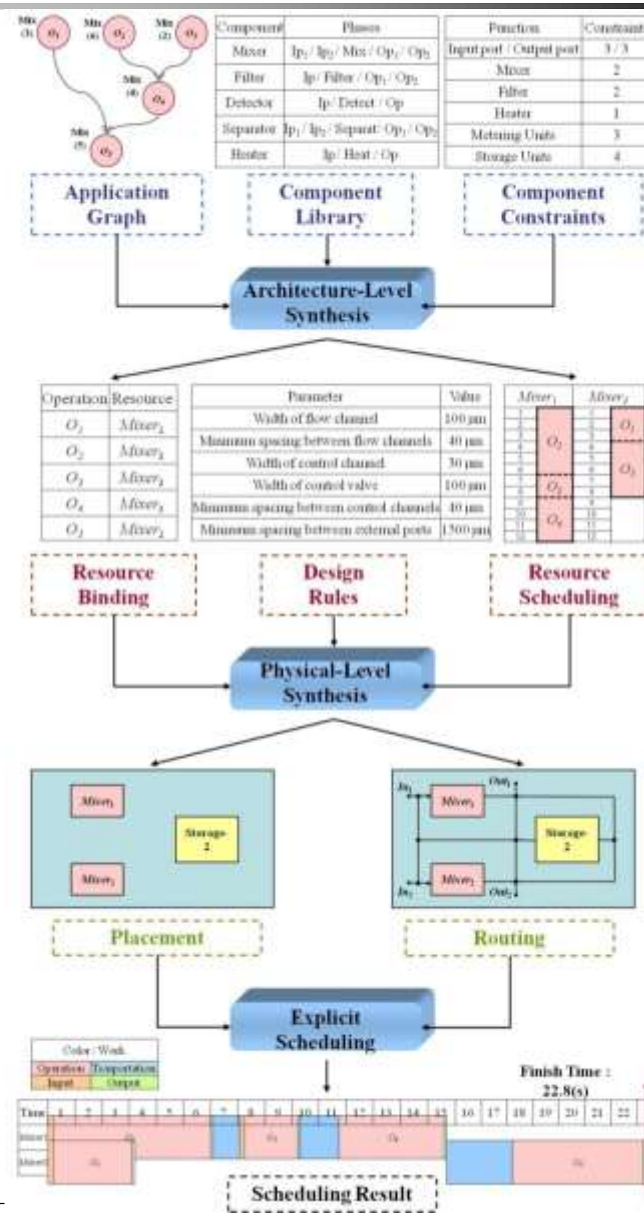




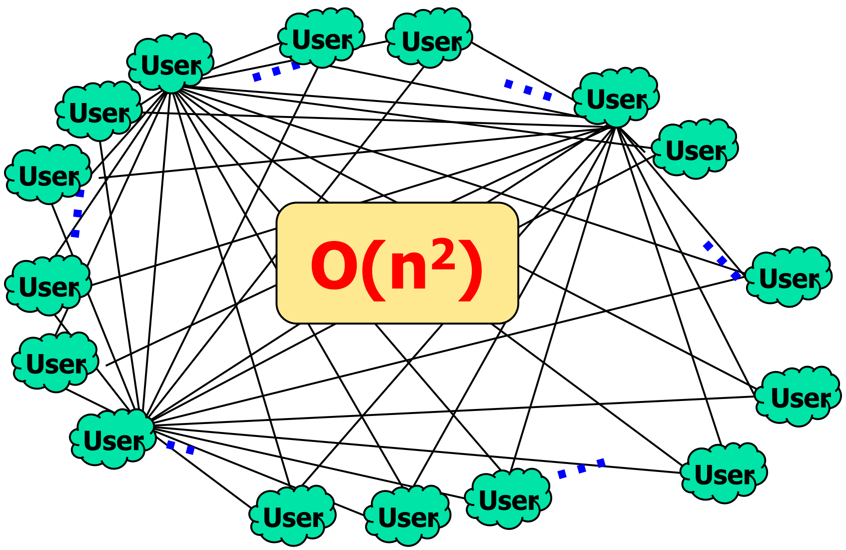
Relational graph

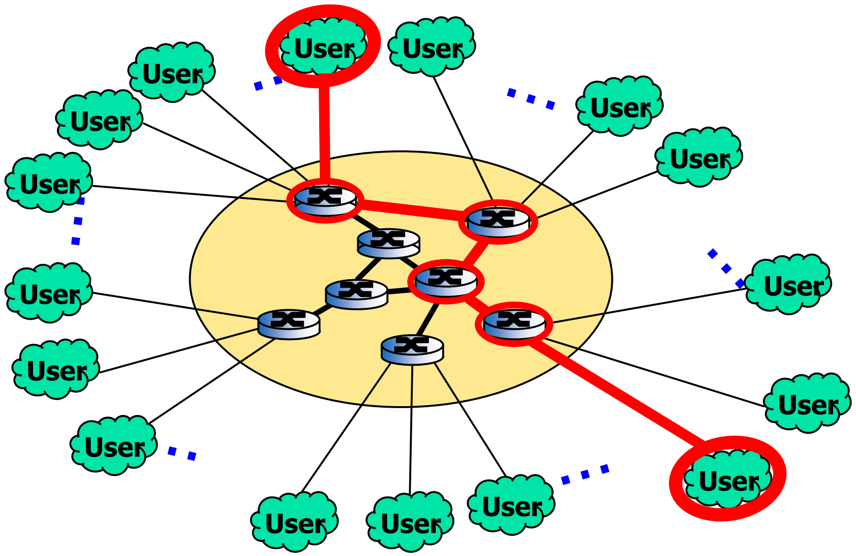
Top-Down Synthesis Methodology

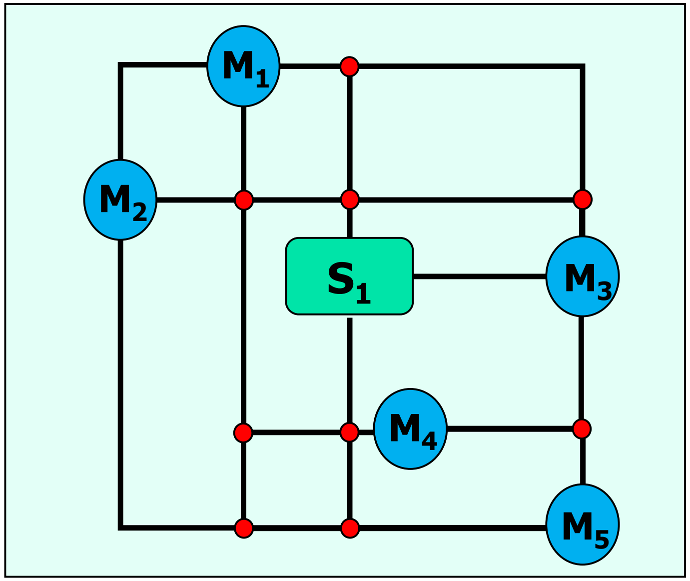
- Architecture-level synthesis
  - Set-based minimum cost maximum flow (SMCMF)
    - Resource binding
    - Resource scheduling
- Physical-level synthesis
  - Incremental cluster expansion (ICE)
    - Placement
    - Routing
- Explicit scheduling
  - List-scheduling based approach
    - Application completion time
    - Total valve-switching amount

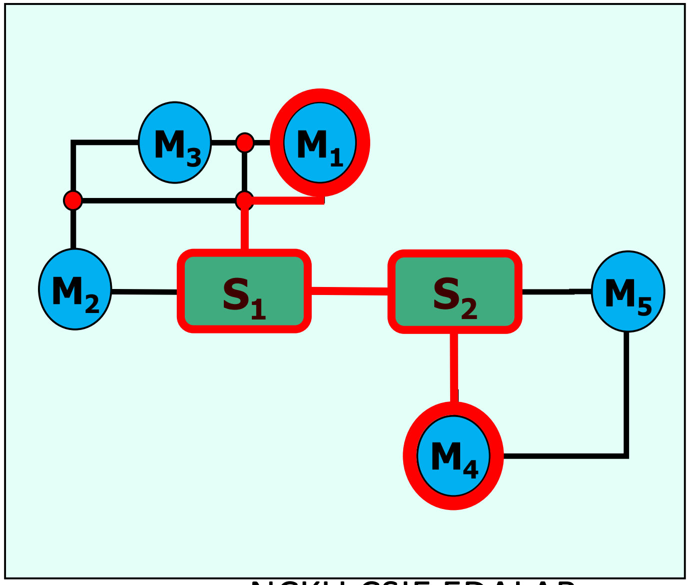


Motivation

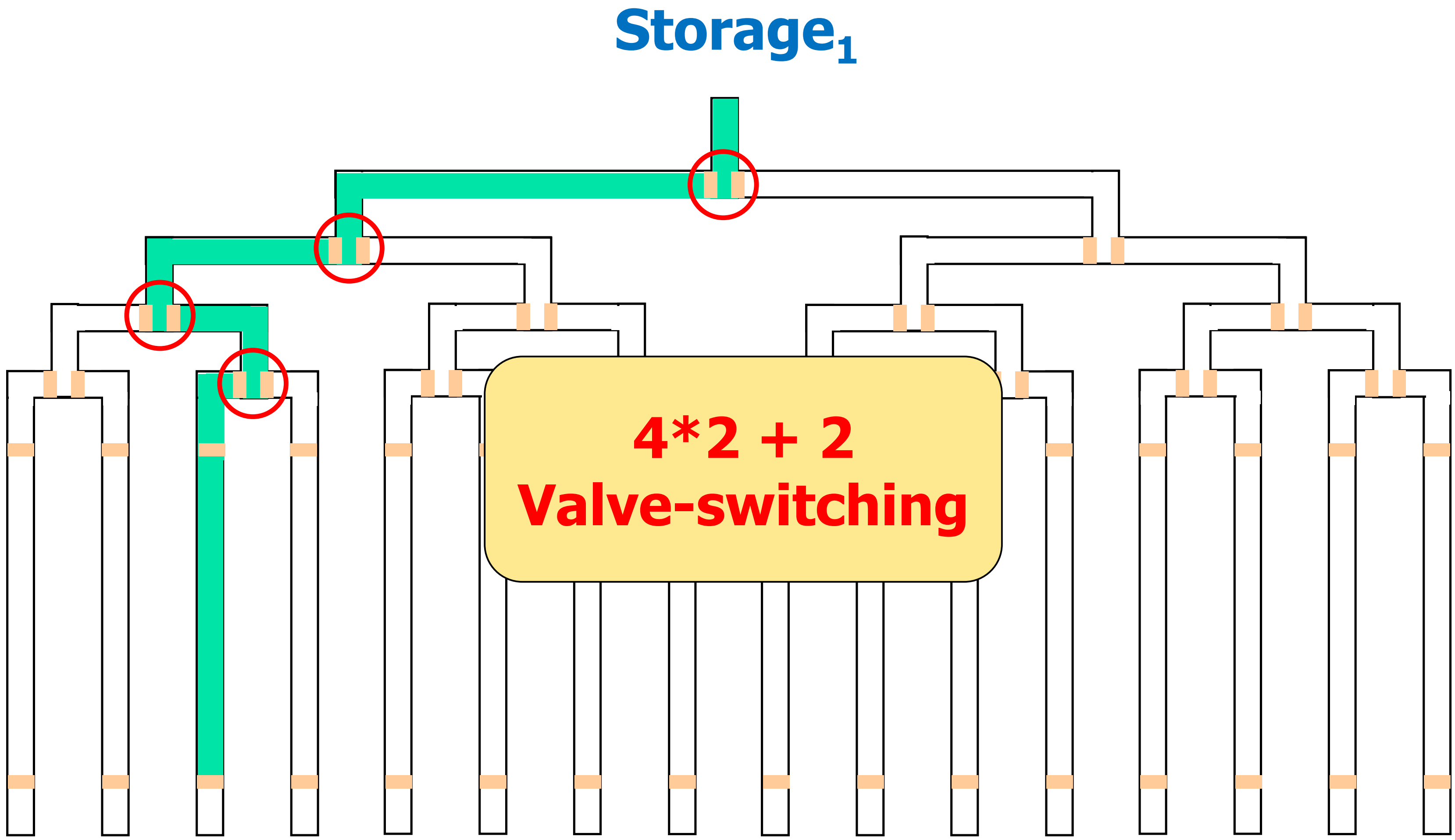




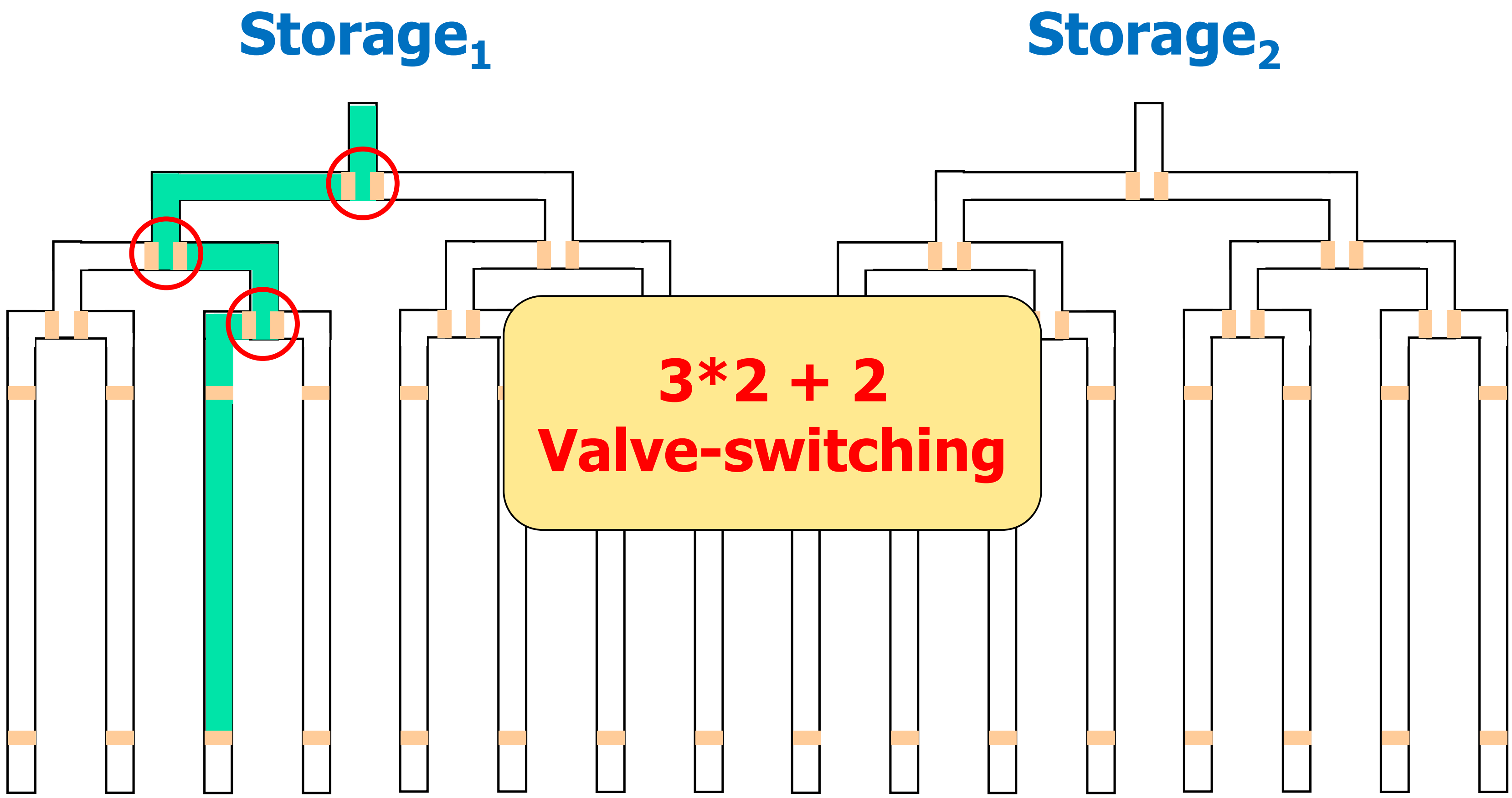




One Storage

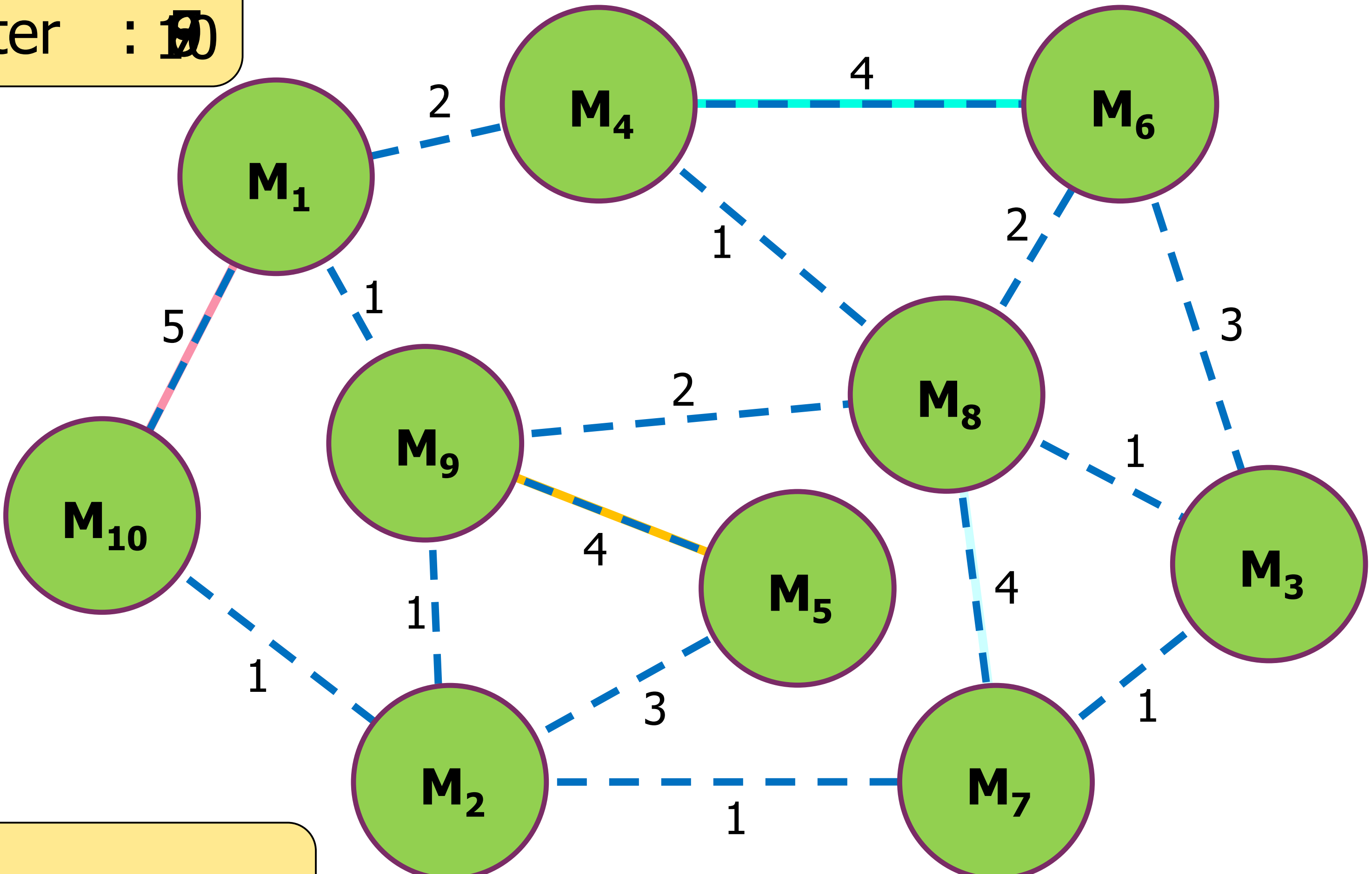


Two Storage



Find the Clusters

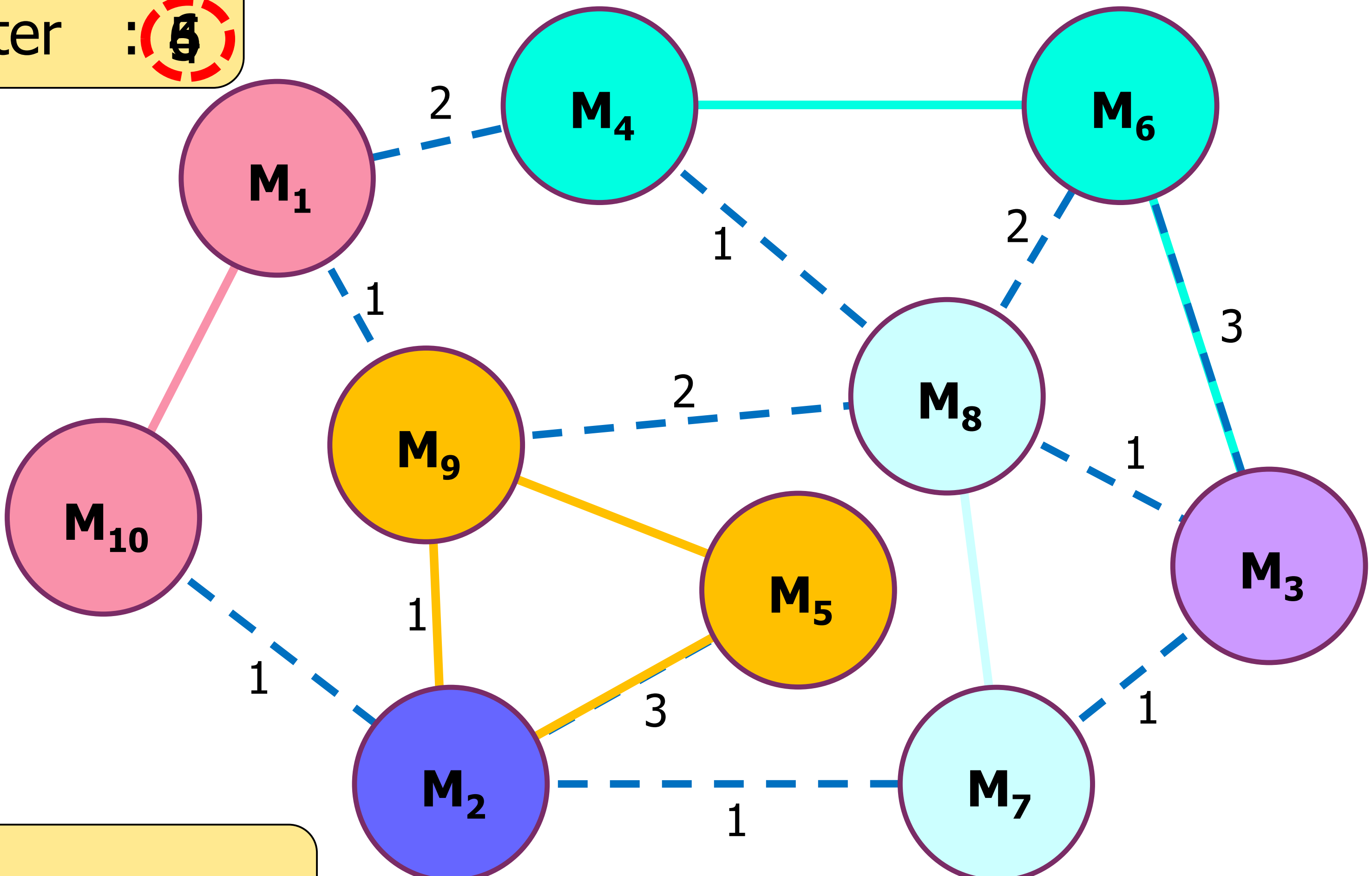
Iteration : 0
Cluster : 10



Storage : 4

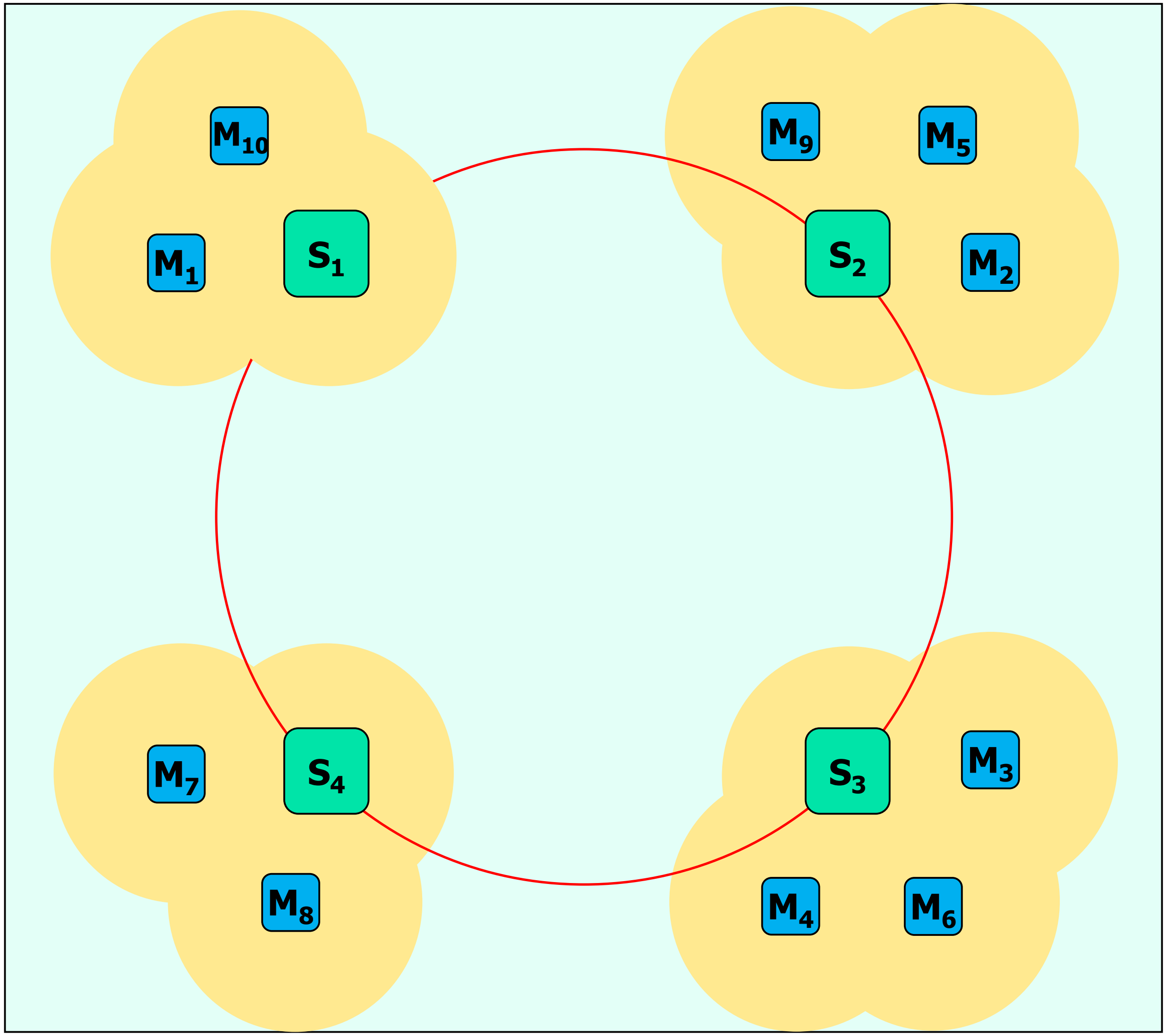
Find the Clusters

Iteration : 2
Cluster : 4

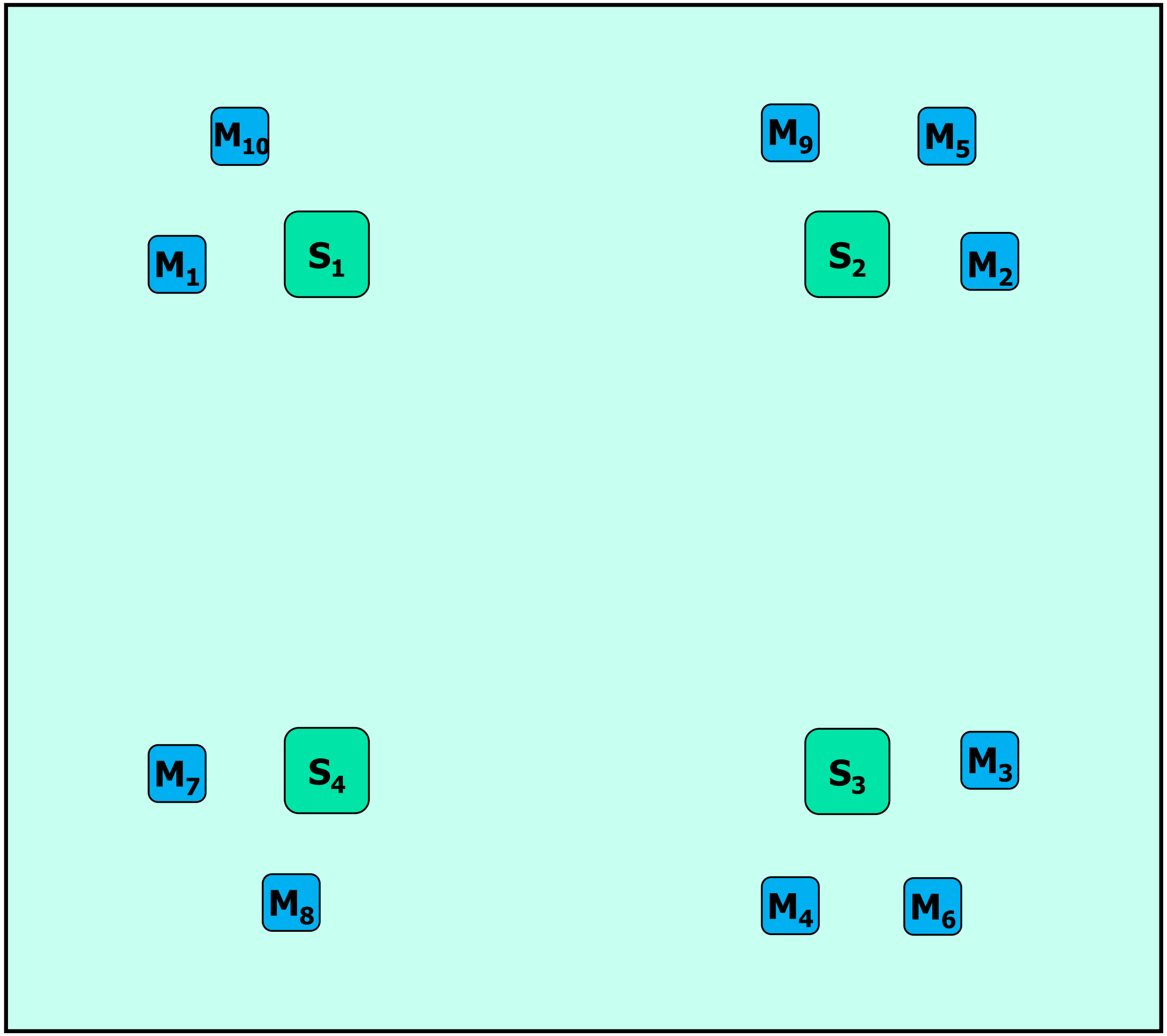


Storage : 4

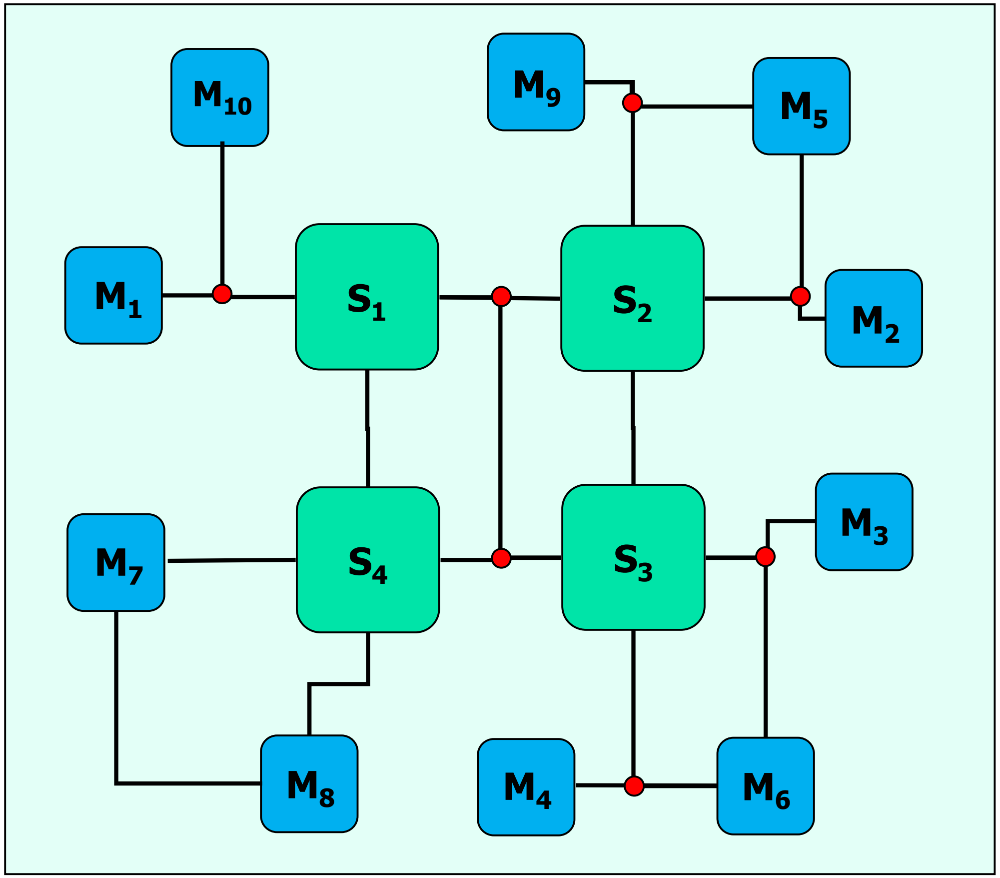
Incremental Cluster Expansion



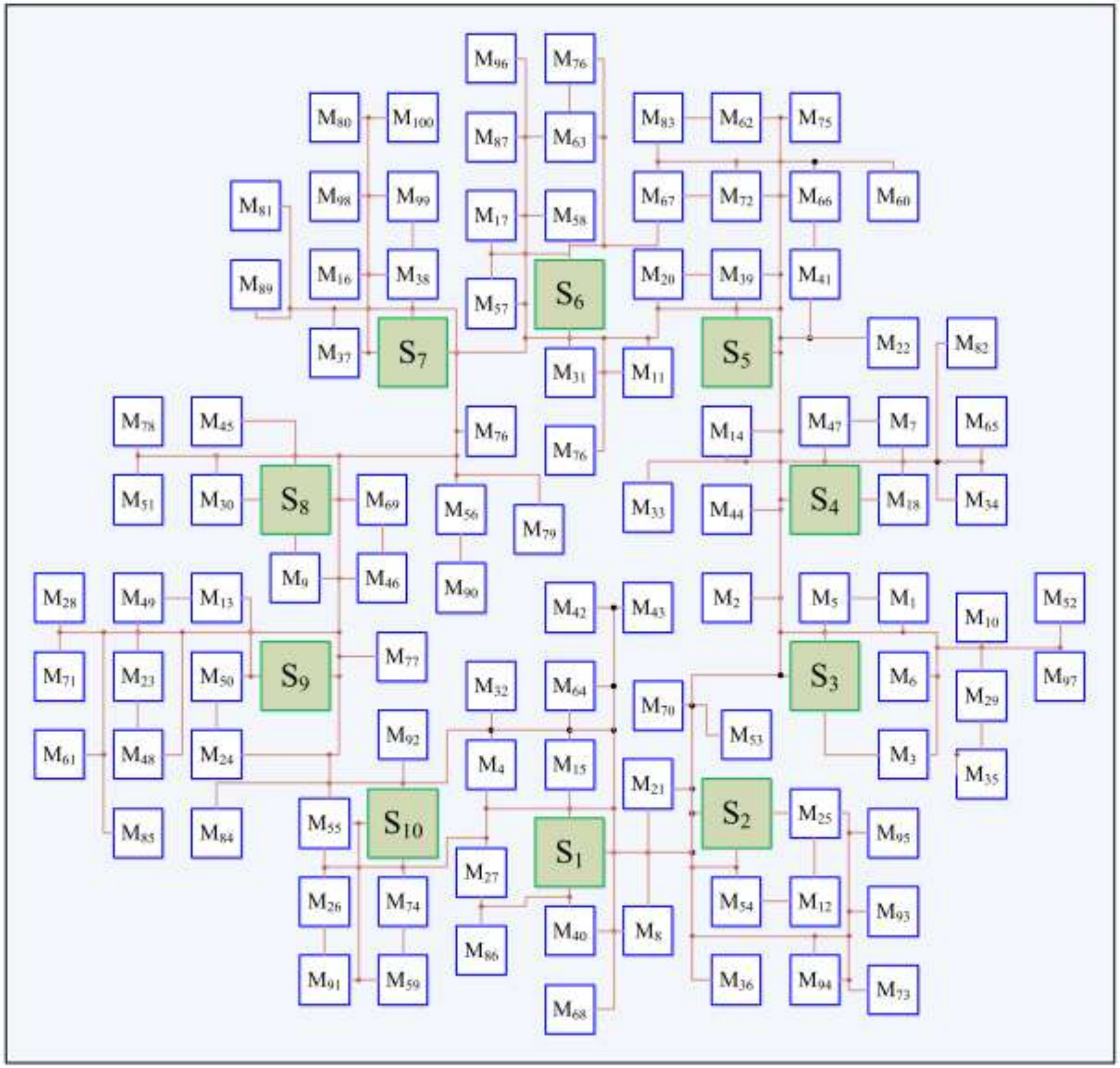
Compaction



Routing by dijkstra's algorithm

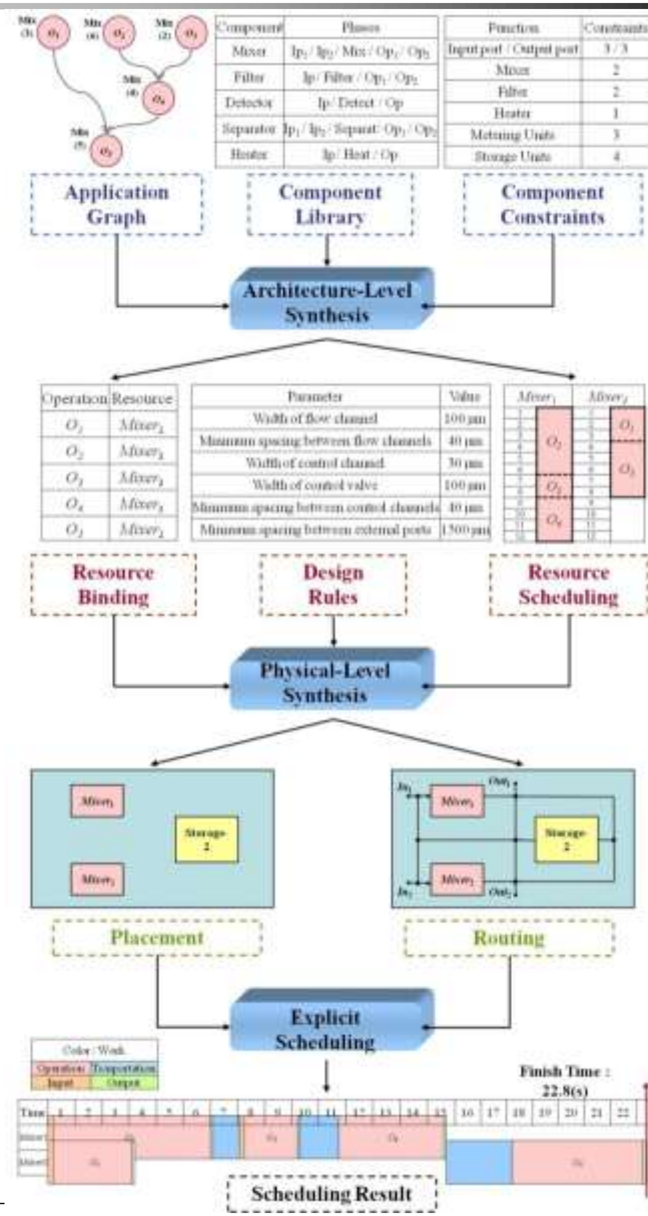


100 Mixers and 10 Storages

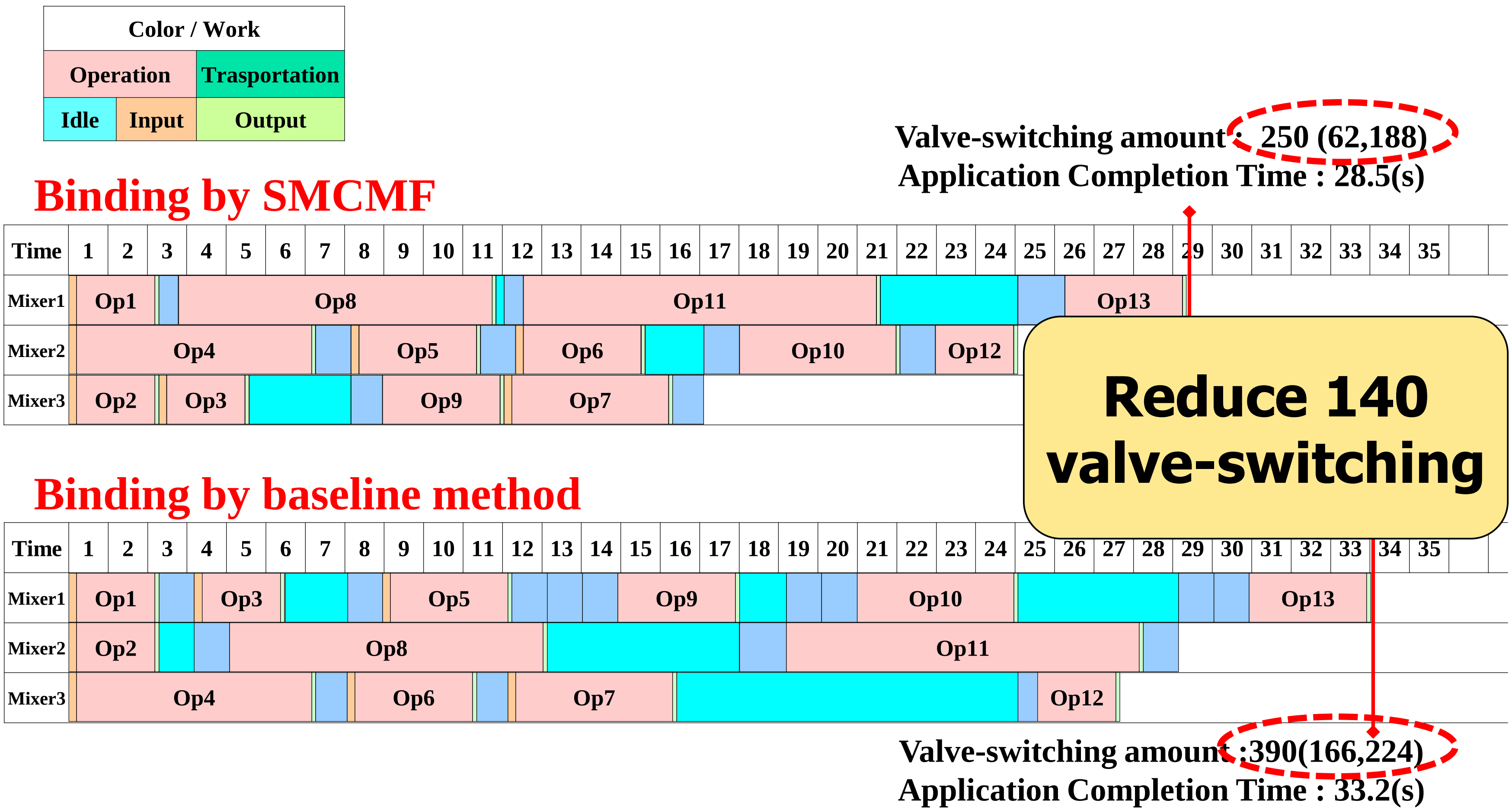


Top-Down Synthesis Methodology

- Architecture-level synthesis
  - Set-based minimum cost maximum flow (SMCMF)
    - Resource binding
    - Resource scheduling
- Physical-level synthesis
  - Incremental cluster expansion (ICE)
    - Placement
    - Routing
- **Explicit scheduling**
  - **List-scheduling based approach**
    - Application completion time
    - Total valve-switching amount



Explicit Scheduling



Outline

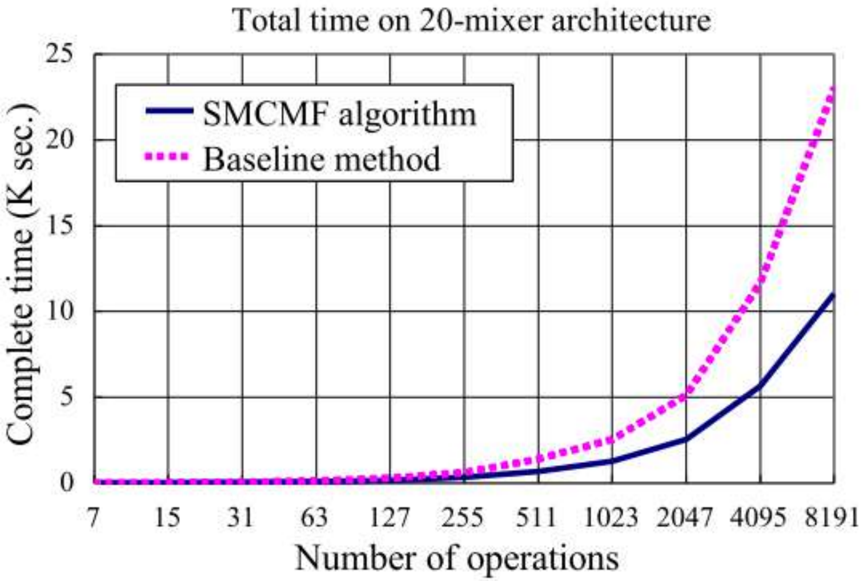
- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

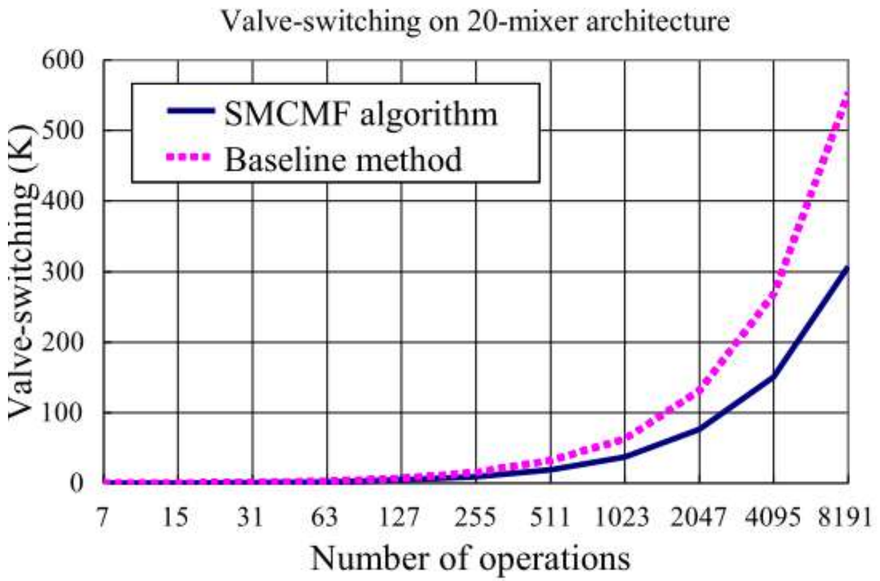
Experimental Settings

- Implemented our algorithm in C++ language on a PC with Core2 Quad processors at 2.66GHz and 3.25GB of RAM
- Compared set-based minimum cost maximum flow binding algorithm with list scheduling based binding algorithm (baseline method) on top-down synthesis methodology
- Tested on several synthetic benchmarks by
  - Adjusted operation numbers from 7 to 8191 and fixed mixers as 20 and storage as 1
  - Adjusted mixers from 10 ~ 100 and fixed operation numbers as 1023 and storage as 1
  - Adjusted storage from 1 ~ 10 and fixed operation numbers as 1023 and mixer as 100

Experimental Results (1/3)

Operation : **7~8191**
Mixer : 20
Storage : 1



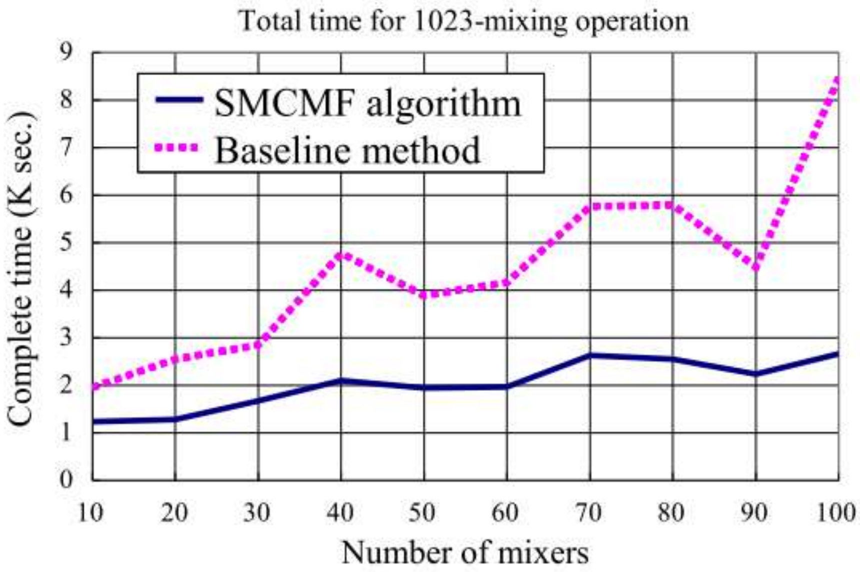


Application Completion Time

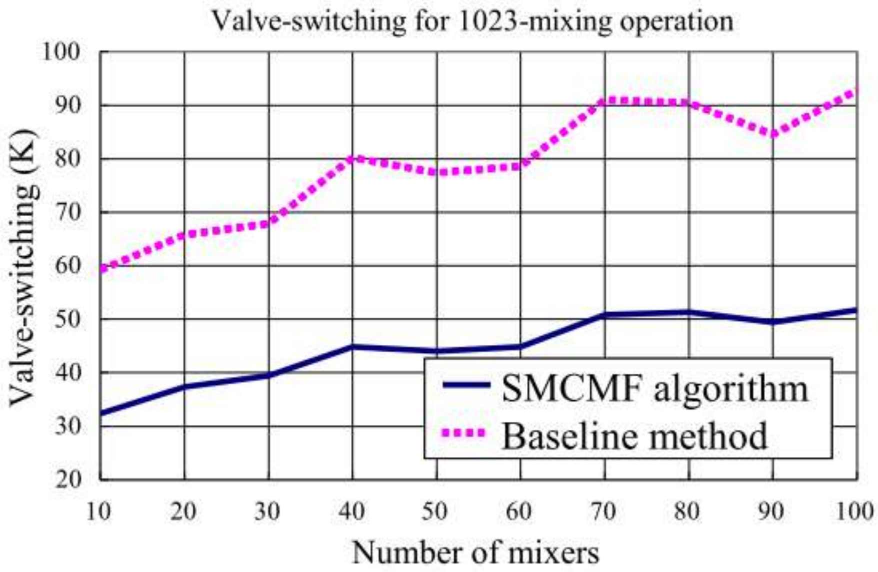
Valve-switching Amount

Experimental Results (2/3)

Operation : 1023
Mixer : **10 ~ 100**
Storage : 1



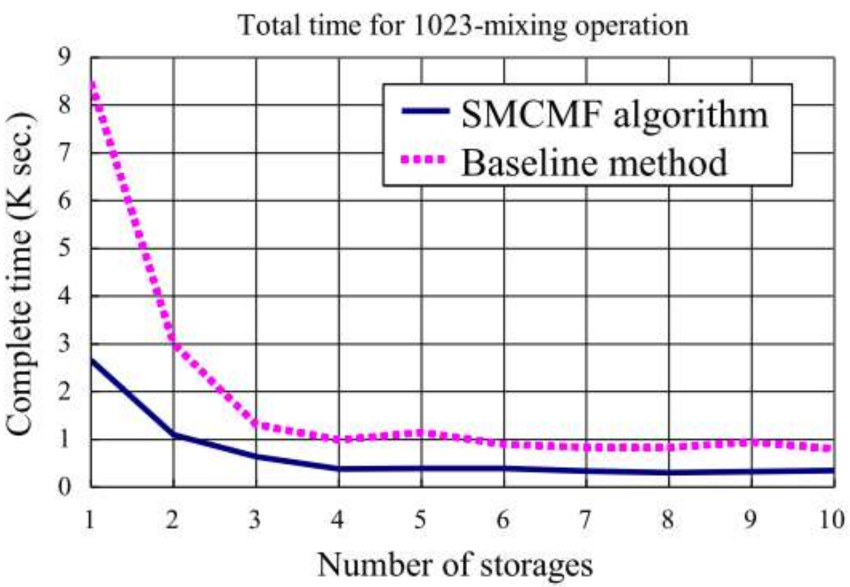
Application Completion Time



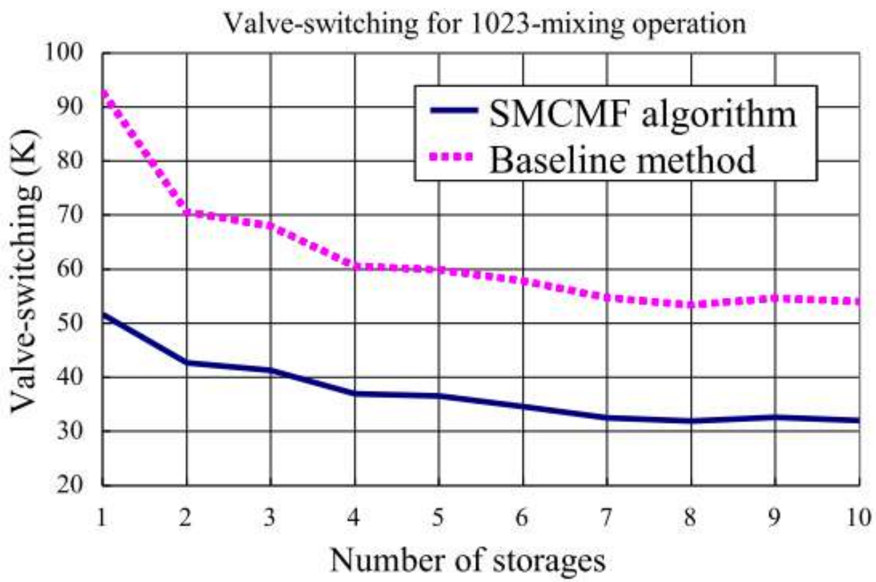
Valve-switching Amount

Experimental Results (3/3)

Operation : 1023
Mixer : 100
Storage : **1 ~ 10**



Application Completion Time



Valve-switching Amount

Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

Conclusions

- The valve-switching activities for the components such as mixer and storage were modeled
- A first top-down synthesis methodology for flow-based microfluidic biochip was presented
- Two algorithms for architecture-level synthesis and physical-level synthesis were proposed
  - Set-based minimum cost maximum flow (SMCMF)
  - Incremental cluster expansion (ICE)
- The experimental results showed that our top-down synthesis methodology not only minimized the valve-switching amount but also reduced the application completion time

Thank You for Your Attention!