Reservoir and Mixer Constrained Scheduling for Sample Preparation on Digital Microfluidic Biochips

Authored by:

Varsha Agarwal[1], Ananya Singla[1], Mahammad Samiuddin[2], Sudip Roy[1], Tsung-Yi Ho[3], Indranil Sengupta[2] and Bhargab B. Bhattacharya[4]

[1]CoDA Laboratory, Dept. of CSE, IIT Roorkee, India
[2]Dept. of CSE, IIT Kharagpur, India
[3]Dept. of CS, National Tsing Hua University, Hsinchu, Taiwan
[4]Advanced Computing and Microelectronics Unit, ISI Kolkata, India

Presented by: **Varsha Agarwal,** *Department of Computer Science and Engineering,*
Indian Institute of Technology Roorkee, India

In the 22nd Asia and South Pacific Design Automation Conference (ASP-DAC) 2017

January 16-19, 2017

Outline of the Talk

- Introduction
- Basic Preliminaries
- Prior Work
- Motivation
- Reservoir Constrained Optimal Scheduling
- Reservoir and Mixer Constrained Scheduling
    - Problem Formulation
    - Proposed Scheme
- Simulation Results
- Conclusions and Future Work

Introduction

- Microfluidic biochip, also termed as lab-on-a-chip (LoC), automates the repetitive work in laboratory by replacing complex equipment with compact integrated systems

Microfluidic Biochip

- Digital Microfluidic Biochip (DMFB)
- Continuous flow Microfluidic Biochip (CMFB)

Digital Microfluidic Biochip (DMFB)

- It is based on the principle of **electrowetting-on-dielectric (EWOD)** [Chakrabarty *et al.*, CRC Press, 2007]
- It manipulates droplets of small size

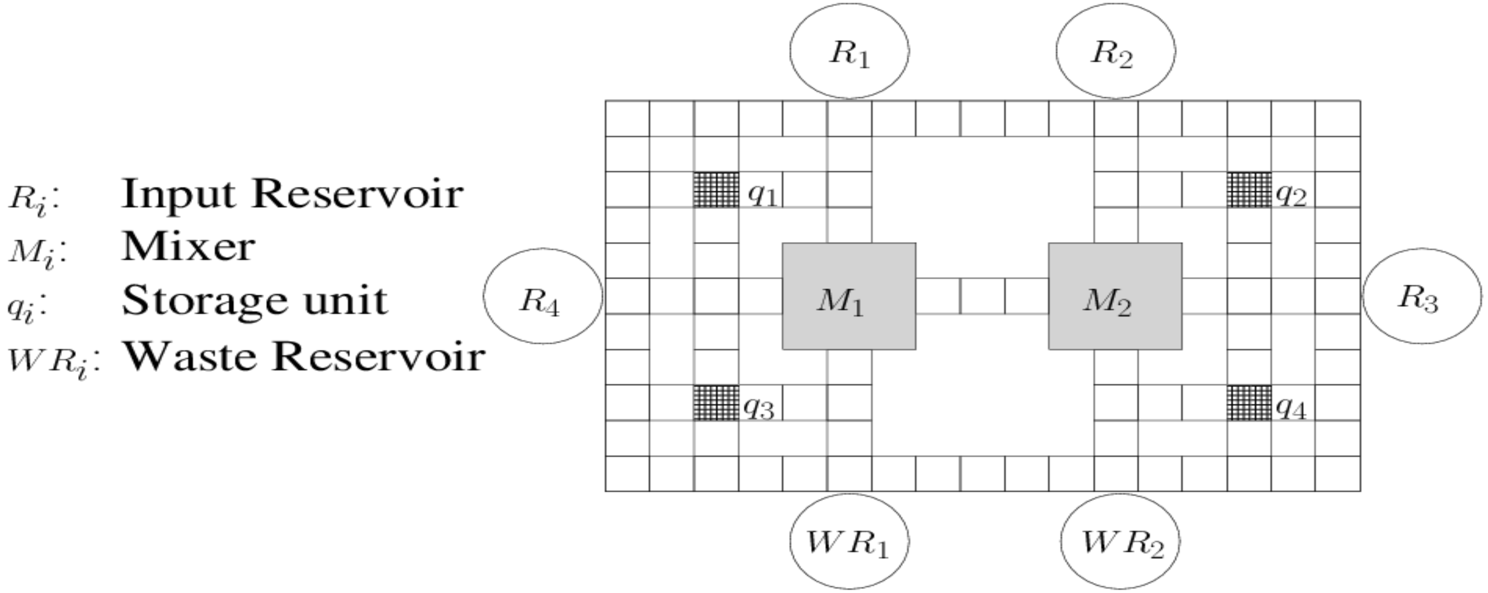


Figure 1: Schematic view of a DMF biochip with two mixers and four reservoirs

Basic Preliminaries

- **Sample Preparation**
  - Mixing of two or more fluids to get desired concentration
- **Mixing tree**
  - The step-wise processing for the generation of target concentration from the supply of input fluids is represented by a binary mixing tree

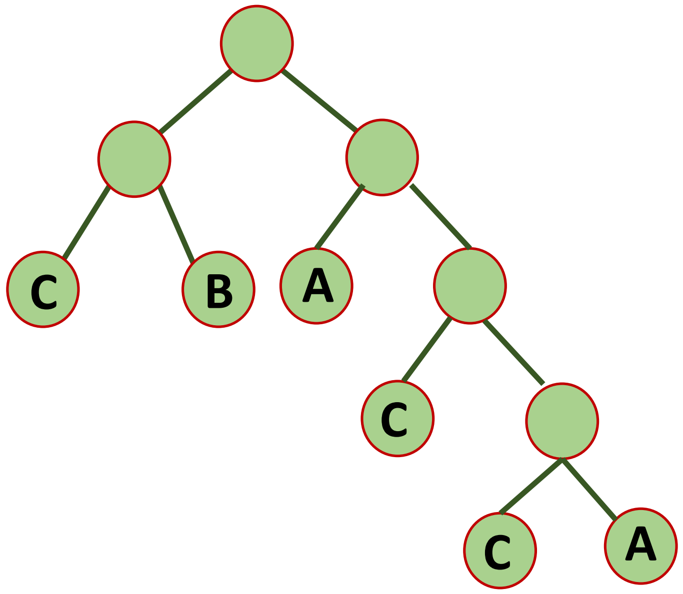


Figure 2: MinMix mixing tree for the mixture (A:B:C) = (5:4:7)

Basic Preliminaries (cont.)

- **Reservoir Switching**
  - At some instance of time, it is required to unload the previous reagent, wash the reservoir and load this reservoir with new reagent. We refer this process of unloading, washing and loading as ‘switching’
  - Reservoir switching takes more time than mixing [Rensch *et al.*, Lab Chip, 2014]

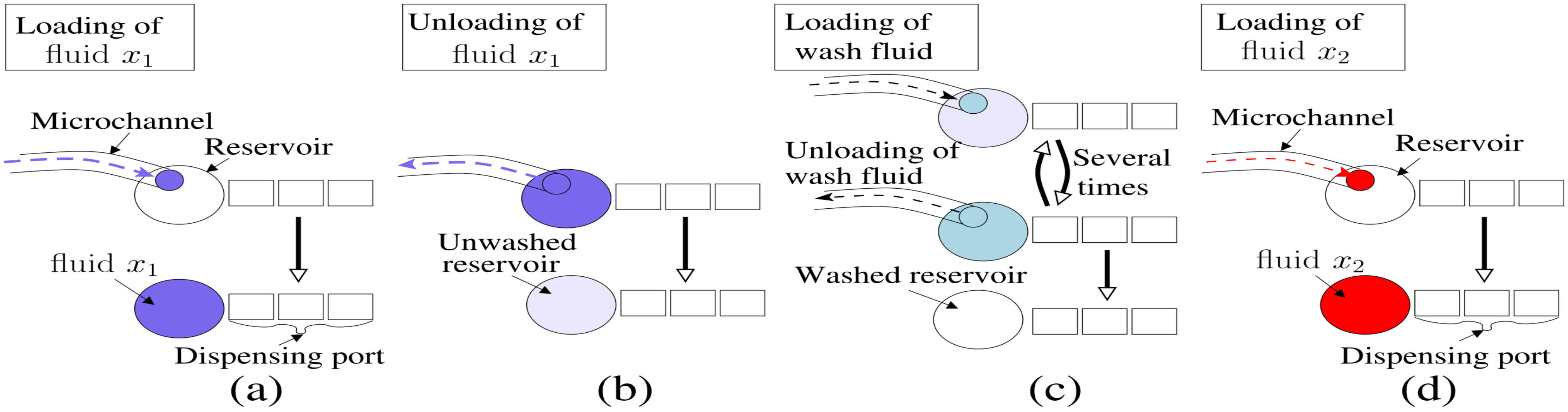


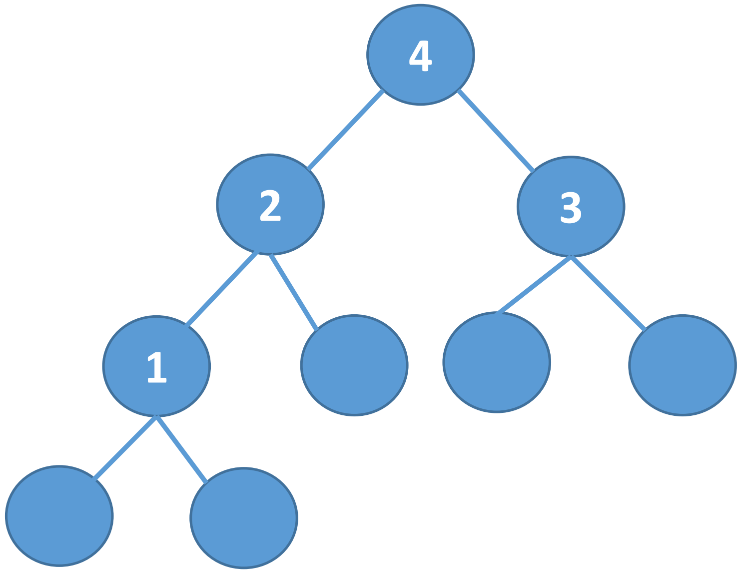
Figure 3: Schematic view of the mechanical process of (a) loading, (b) unloading, (c) washing, and (d) reloading of the reservoir at a dispenser

Prior Work: Mixing algorithms

- **MinMix** [Thies *et al.,* Natural Computing, 2008]
  - Scans N d-bits of the binary fractions corresponding to the target CFs N reagents from right to left
- **RMA** [Roy *et al.,* VLSID, 2011 & Roy *et al.,* TODAES, 2015]
  - Based on fractional decomposition of algebraic expression of a target ratio
  - Minimize reagent usage, droplet transportation time and storage requirement
- **CoDOS** [Liu *et al.,* ICCAD, 2013]
  - Generates a recipe matrix that contains the CF of each reagent fluid in binary representation
  - Find out the rectangle within the recipe matrix for possible dilution operations that can be shared
  - Droplet sharing in the mixing tree

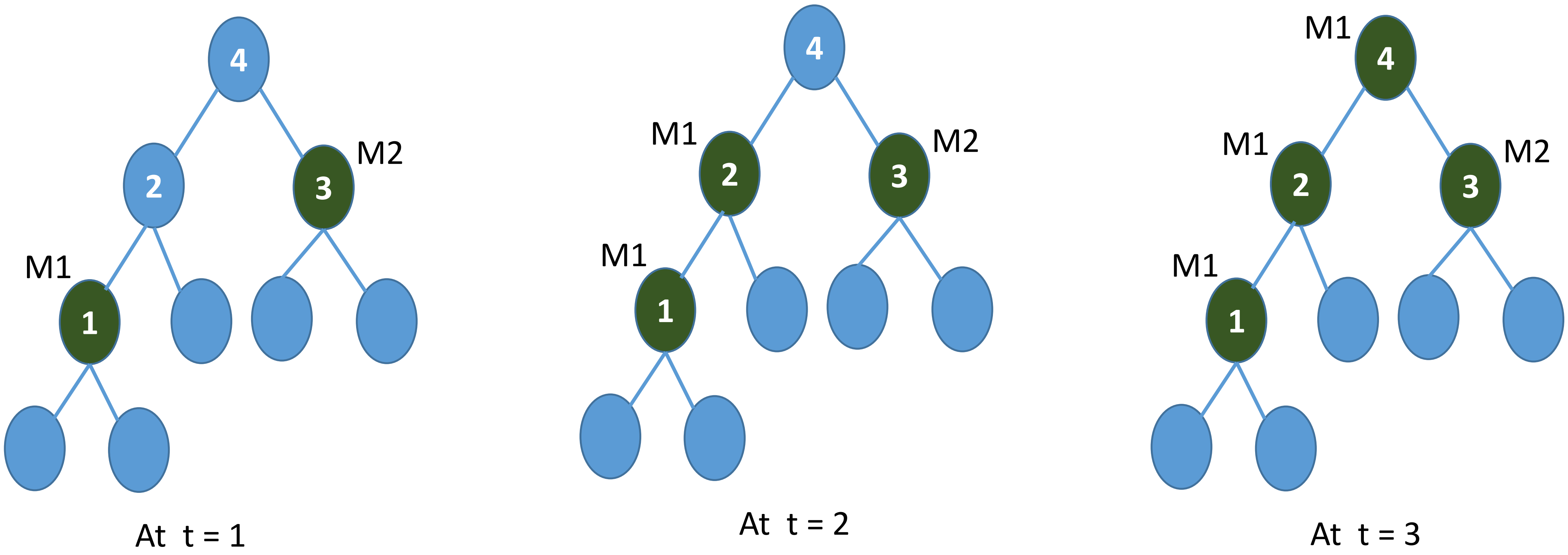
Prior Work (cont.): Scheduling algorithm

- **Optimal Scheduling** [Luo and Akella, TASE, 2011]
  - It schedules the given mixing tree in the minimum total mixing (completion) time
  - Optimal scheduling algorithm work as follows:
    - Given a following mixing tree to be scheduled with 2 mixers and no reservoir constrained



Prior Work (cont.)

- **Optimal Scheduling** [Luo and Akella, TASE, 2011]



Motivation: Why Reservoir Switching Required?

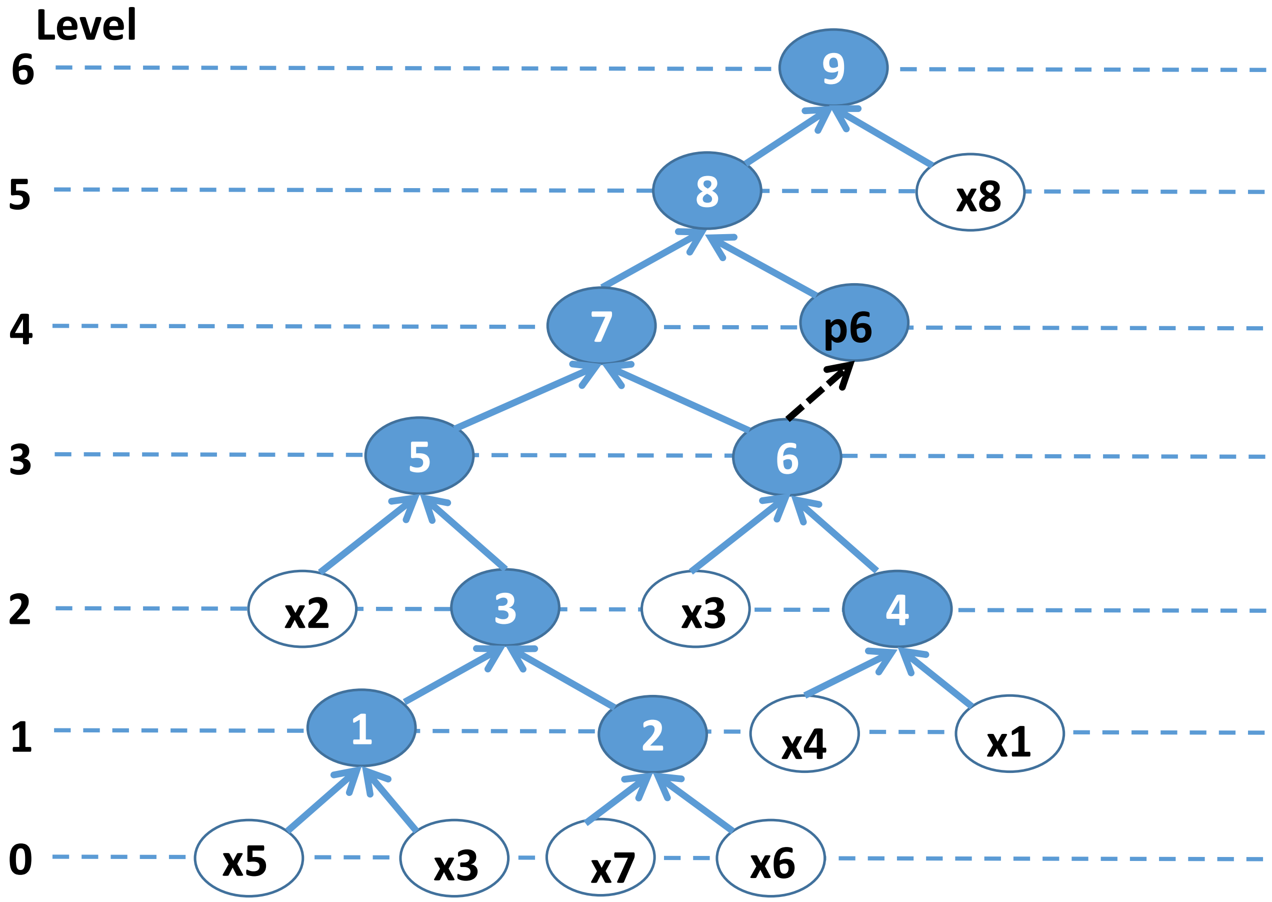
- DMFB is small in size, so there is limited amount of resources that can be placed on it
- For real-life applications, while performing automated sample preparation we need more reagents than the number of reservoirs available

Reservoir Constrained Optimal Scheduling (ROS)

- Modified optimal scheduling considering reservoir as a constrained is called Reservoir Constrained Optimal Scheduling (ROS)
- The reservoirs are loaded or switched with the required reagents in level wise order

ROS Example

- Given a following CoDOS tree with target ratio 500 : 300 : 1000 : 500 : 100 : 100 : 25 : 2475 (approximated as 6 : 4 : 13 : 6 : 1 : 1 : 1 : 32 in 64-scale). Consider the DMFB architecture given in Fig. 1 with 2 mixers and 4 reservoirs



Schedule for $N_M = 2$ and $N_R = 4$

ROS Example (cont.)

At t = 0

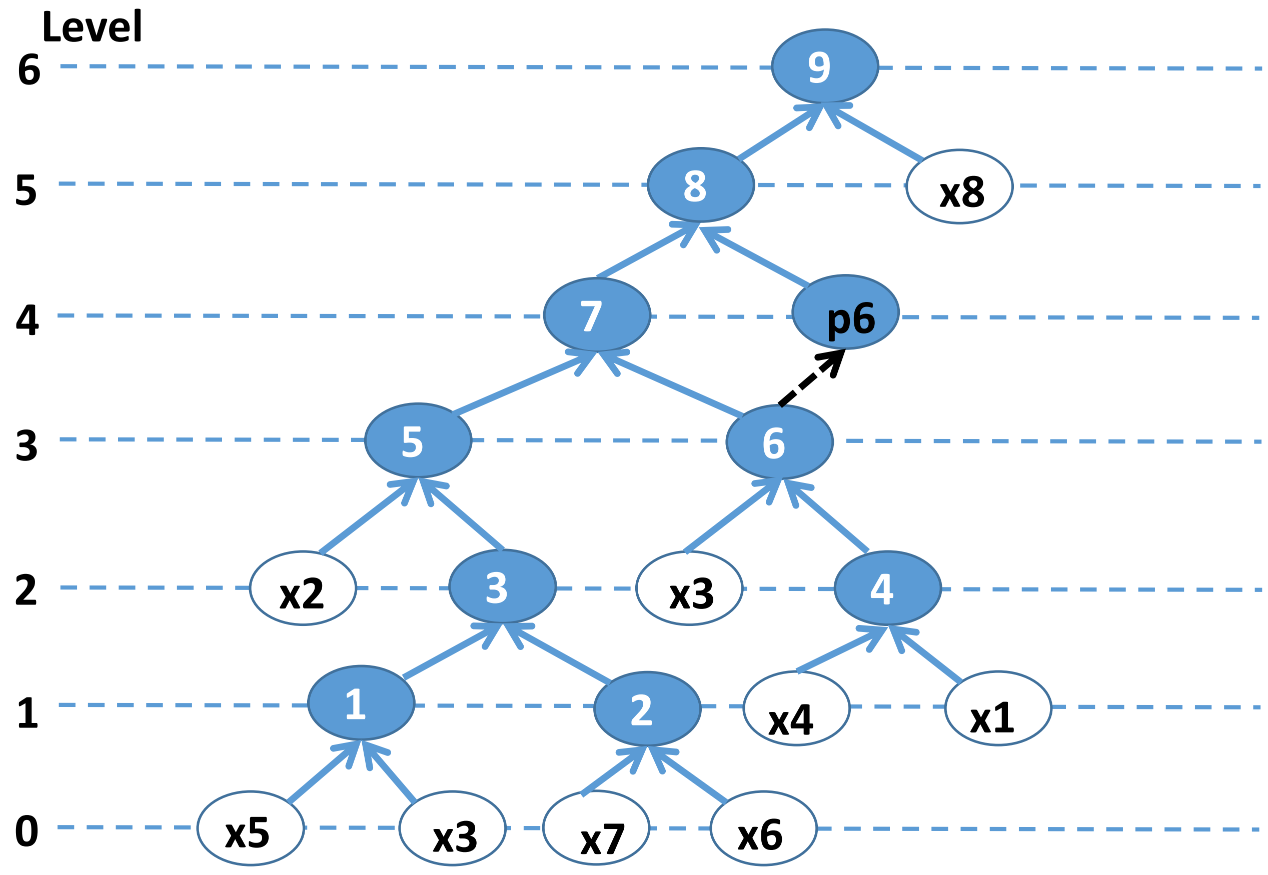
Probable list of nodes

1	2	4

Reservoirs

Available Mixers

M1	M2



ROS Example (cont.)

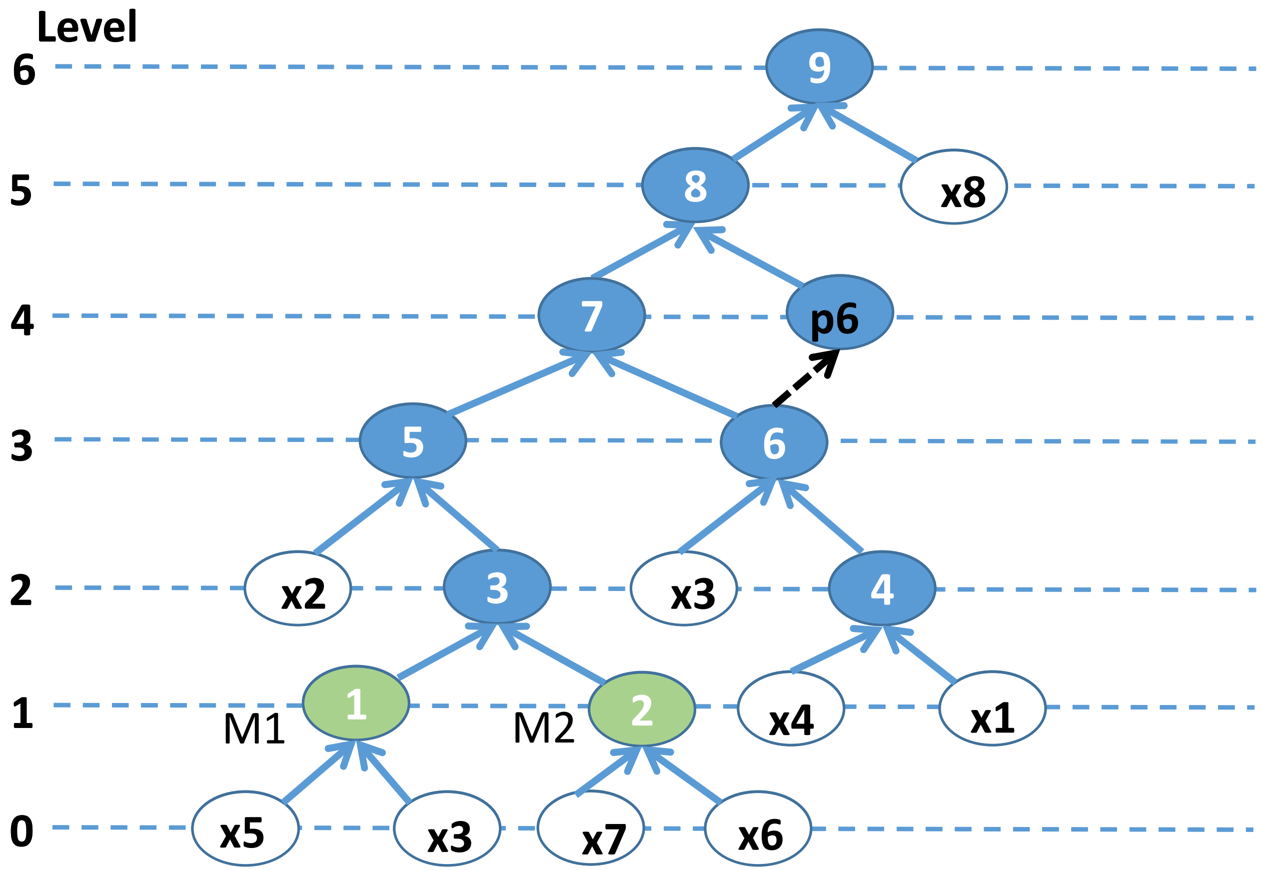
At t = 1

Probable list of nodes ~~1~~ ~~2~~ 4

Reservoirs

- x5
- x3
- x7
- x6

Available Mixers



ROS Example (cont.)

After t = 1

Probable list of nodes

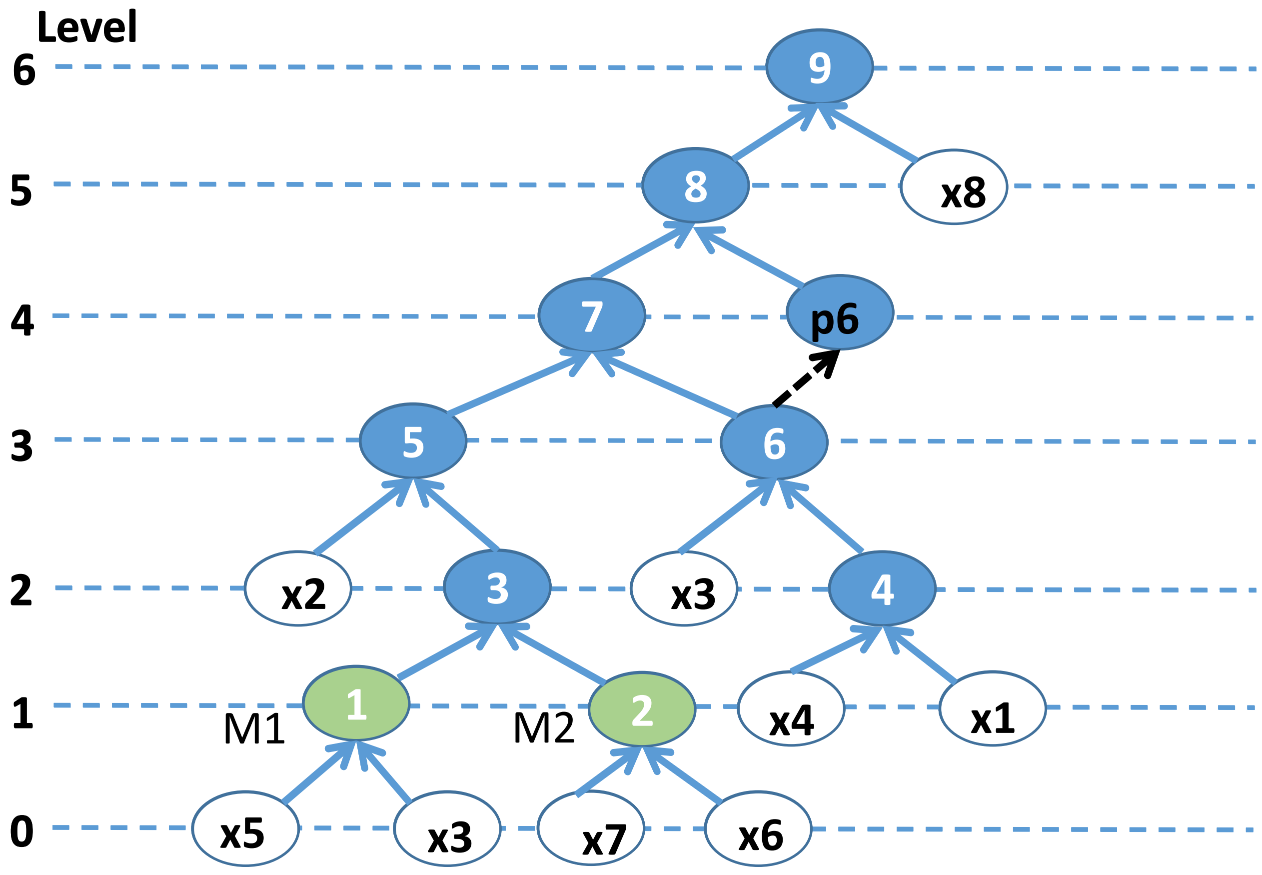
3	4

Reservoirs

x5
x3
x7
x6

Available Mixers

M1	M2



ROS Example (cont.)

At t = 2

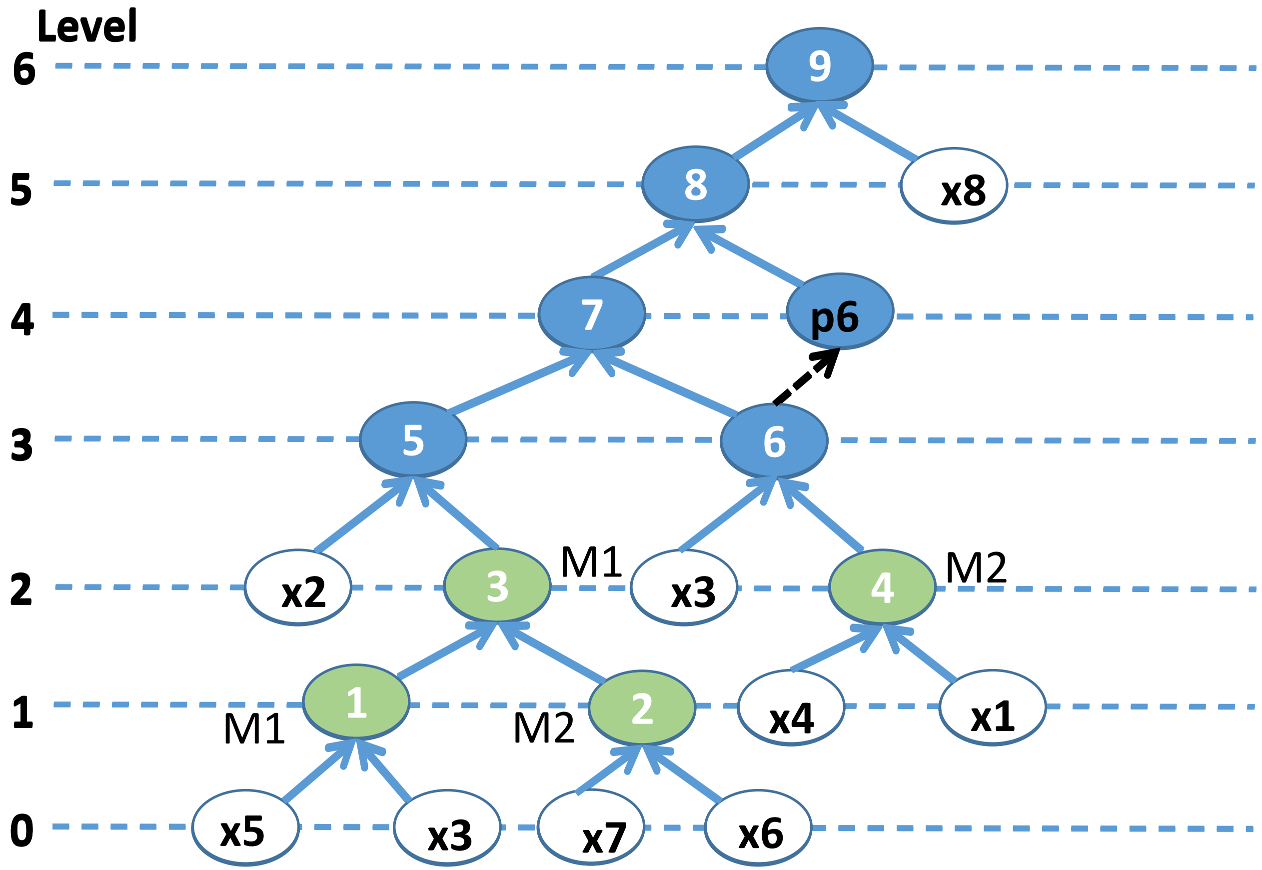
Probable list of nodes ~~3~~ ~~4~~

Number of switching required = 1

Reservoirs

x4
x1
x7
x6

Available Mixers



ROS Example (cont.)

After t = 2

Probable list of nodes

5	6

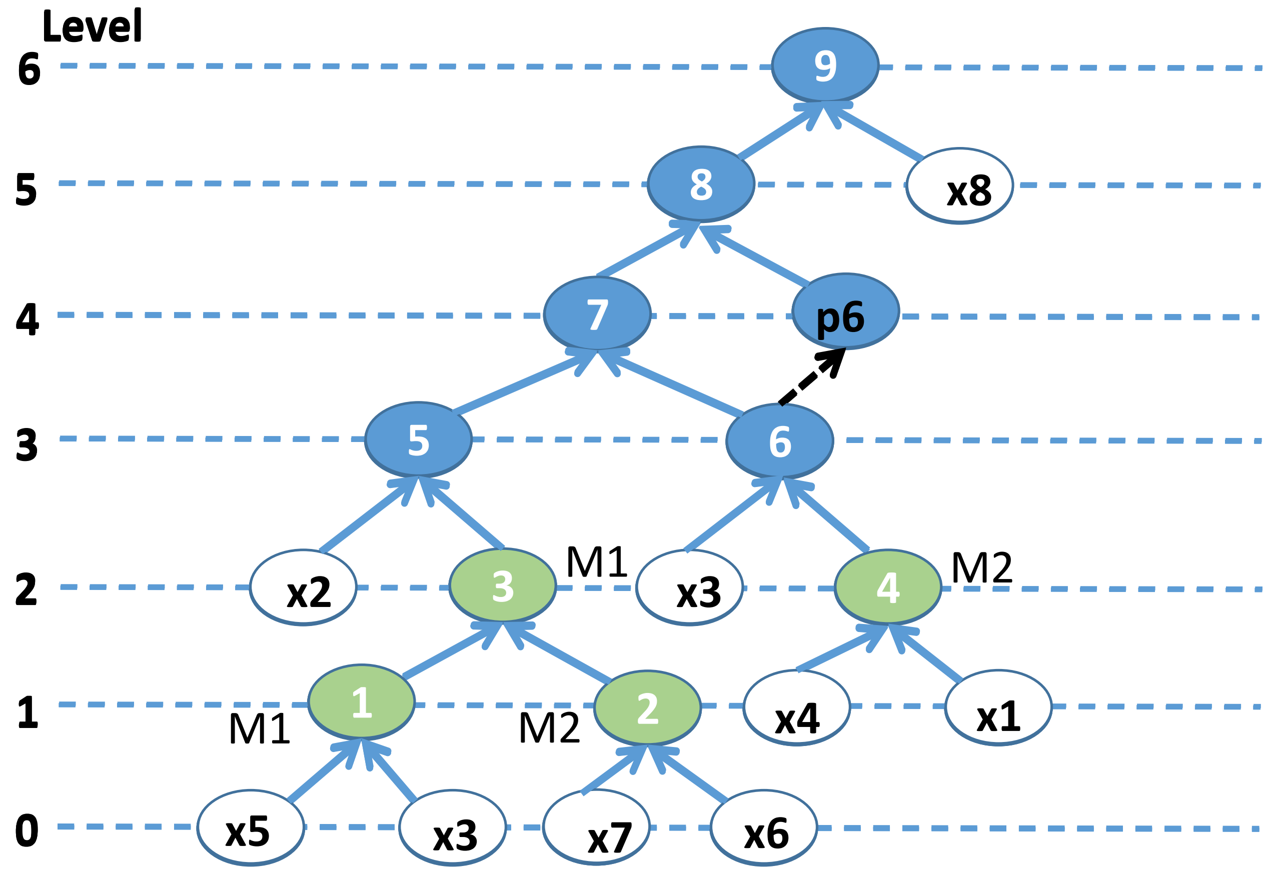
Number of switching required = 1

Reservoirs

x4
x1
x7
x6

Available Mixers

M1	M2



ROS Example (cont.)

At t = 3

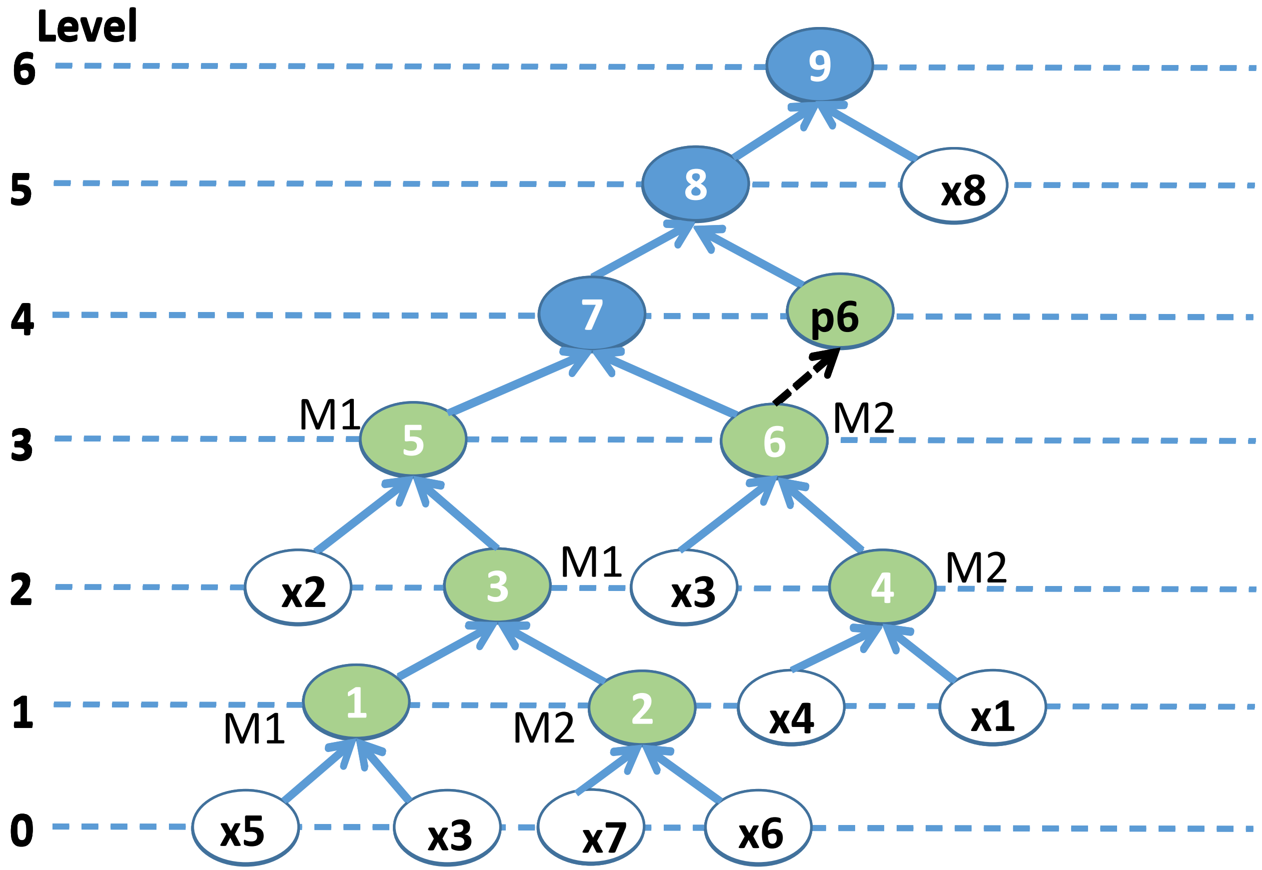
Probable list of nodes ~~5~~ ~~6~~ 7

Number of switching required = 2

Reservoirs

x2
x3
x7
x6

Available Mixers



ROS Example (cont.)

At t = 4

Probable list of nodes ~~7~~ 8

Number of switching required = 2

Reservoirs

x2

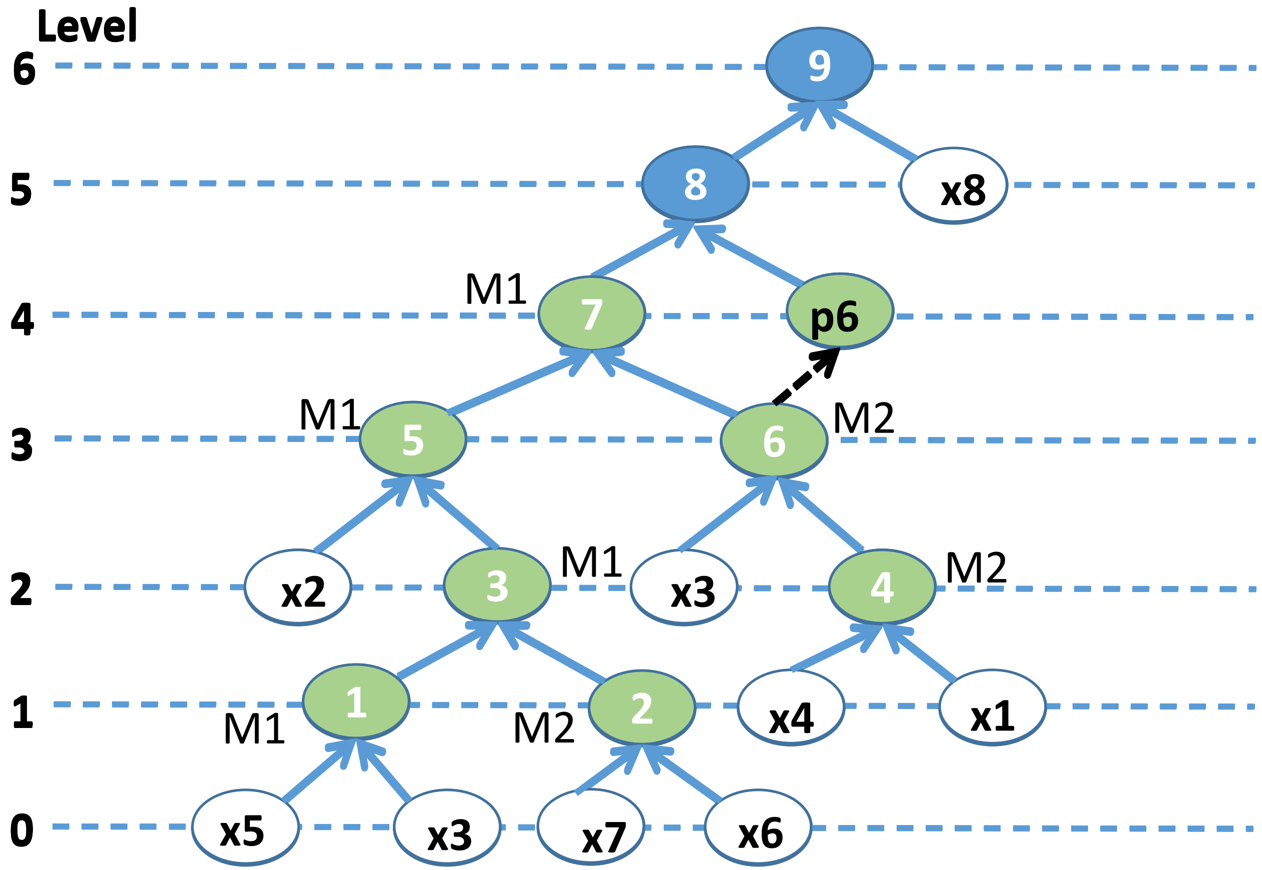
x3

x7

x6

Available Mixers

M2



ROS Example (cont.)

At t = 5

Probable list of nodes ~~8~~ 9

Number of switching required = 2

Reservoirs

x2

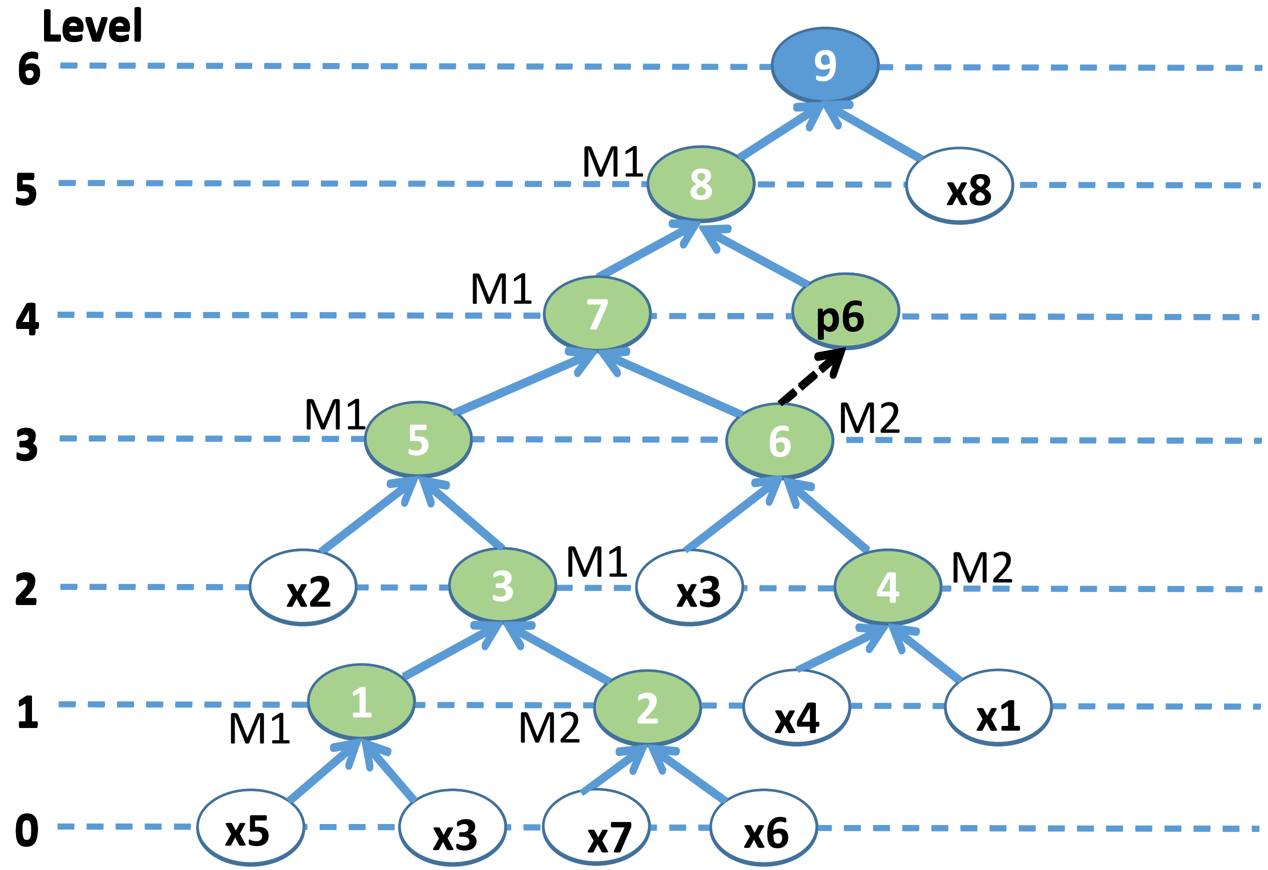
x3

x7

x6

Available Mixers

M2



ROS Example (cont.)

At t = 6

Probable list of nodes ~~9~~

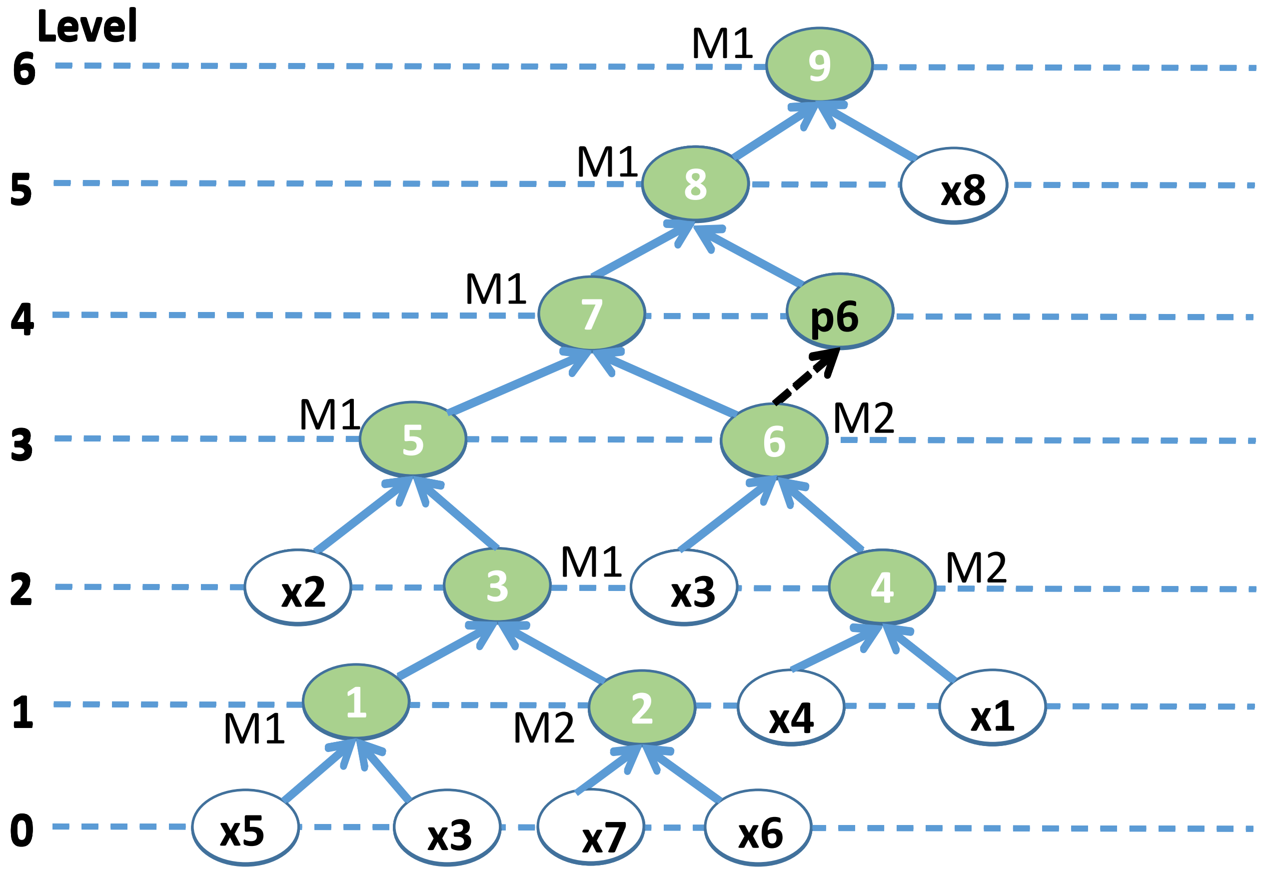
Number of switching required = 3

Reservoirs

x8
x3
x7
x6

Available Mixers

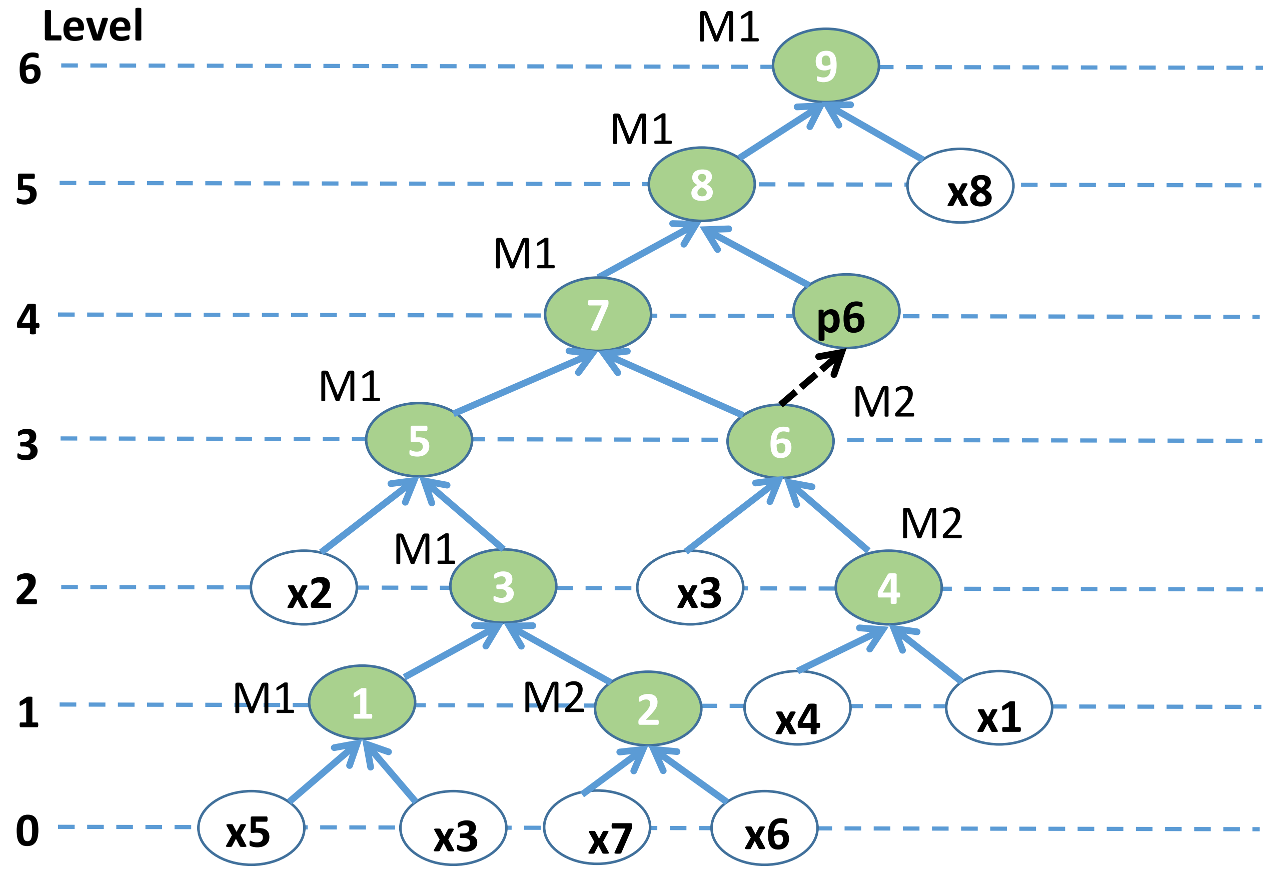
M2



ROS Example (cont.)

Total Mixing time = 6

Number of switching required = 3



Reservoir and Mixer Constrained Scheduling (RMS)

- In ROS, the priority is given to reducing the **mixing time** rather than **reservoir switching**
- But reservoir switching takes more time than mixing [Rensch *et al.*, Lab Chip, 2014]
- In RMS, reducing reservoir switching is main objective

Problem Formulation- RMS

- **Inputs**: Mixing tree (T), number of mixers available (N_M) and number of reservoirs available (N_R)
- **Output**: Scheduled mixing tree and total mixing time (T_m)
- **Constraints**:
  - Number of reservoirs used at any time t, N_R
  - Number of mixers used at any time t, N_M
- **Objectives:**
  - Minimize the number of switching (S)
  - Minimize the total mixing time (T_m)

Proposed Scheme - RMS

1. Approximate given target ratio at accuracy level d

2. Create a dynamic list ($D = \{x_i(c_i)\}$), where x_i represents the reagent i and $c_i/2^d$ is the target concentration of x_i

3. The dynamic list is sorted in decreasing order of (c_i)'s and in which the reagent's priorities are set in increasing order

4. A probable list is created which contains the intermediate nodes that can be scheduled at the current time

RMS (cont.)

- For each node and current reservoir following cases may occur:

Case -1

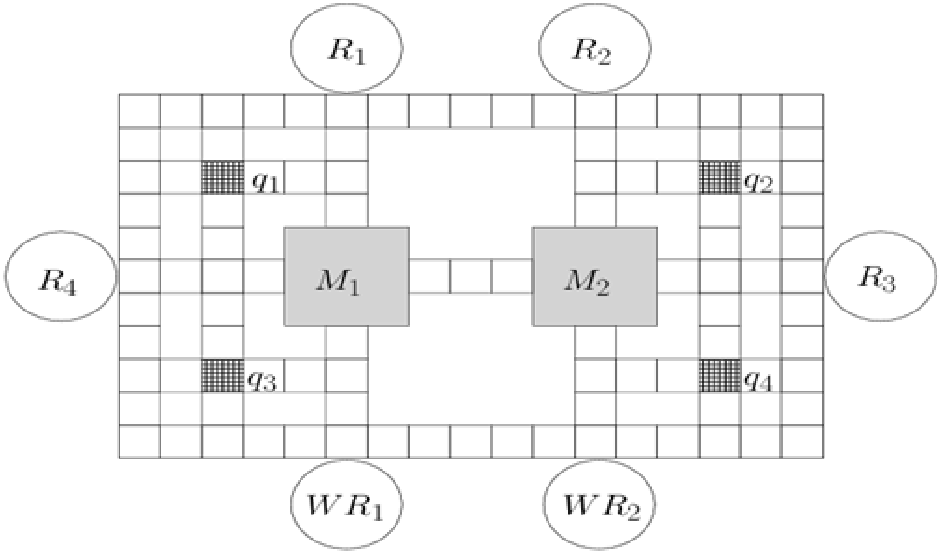
- **Both Reservoirs are empty.**
  - If reagents of current node is not loaded, then load them

Case -2

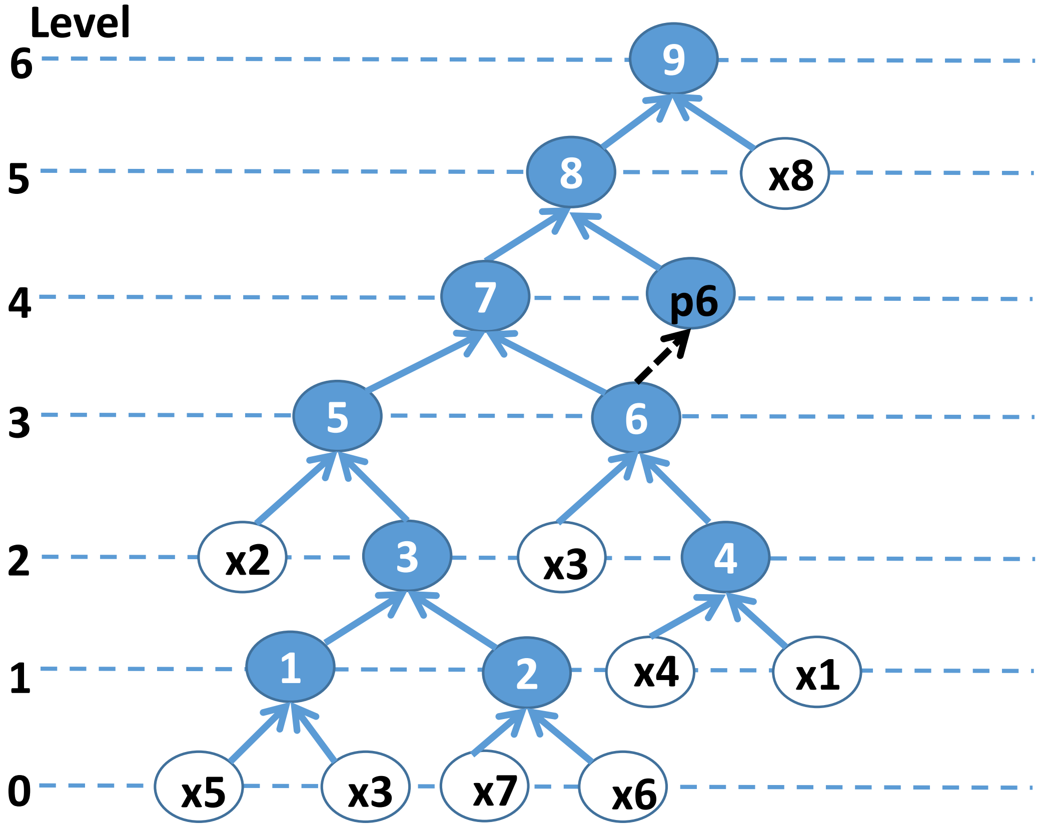
- **Reservoirs are loaded.**
  - Switch the reagents, if it has high priority

RMS - example

- Given a following CoDOS tree with target ratio 500 : 300 : 1000 : 500 : 100 : 100 : 25 : 2475 (approximated as 6 : 4 : 13 : 6 : 1 : 1 : 1 : 32 in 64-scale)
- Consider the DMFB architecture with 2 mixers and 4 reservoirs given below



Schedule for N_M = 2 and N_R = 4



Initial Dynamic sorted list

x8	x3	x1	x4	x2	x5	x6	x7
32	13	6	6	4	1	1	1

Initial Probable list of nodes

1	2	4

RMS - example

At t = 0

Dynamic sorted list

x8	x3	x1	x4	x2	x5	x6	x7
32	13	6	6	4	1	1	1

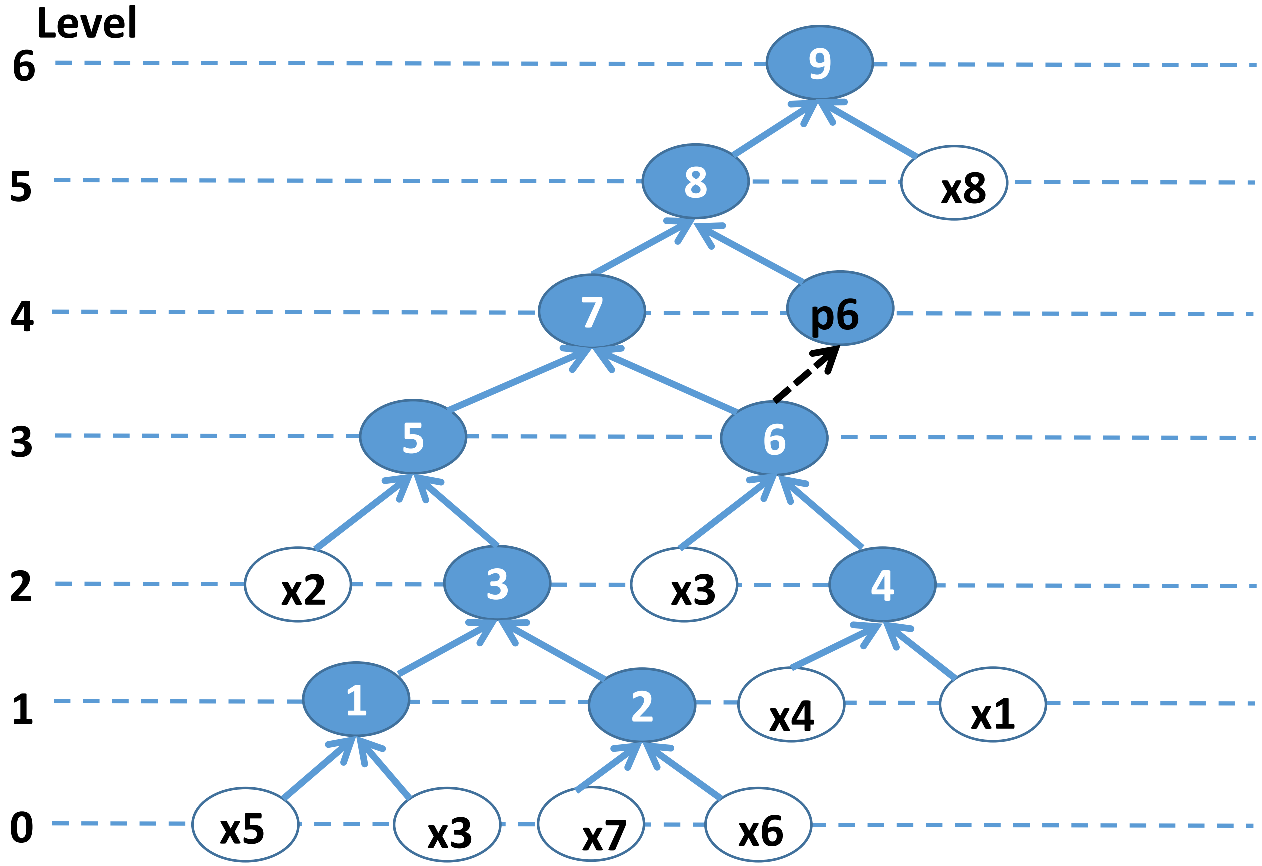
Probable list of nodes

1	2	4

Reservoirs

Available Mixers

M1	M2



RMS - example

At t = 1

Dynamic sorted list

x8	x3	x1	x4	x2	x5	x6	x7
32	13	6	6	4	~~1~~ 0	~~1~~ 0	1

Probable list of nodes

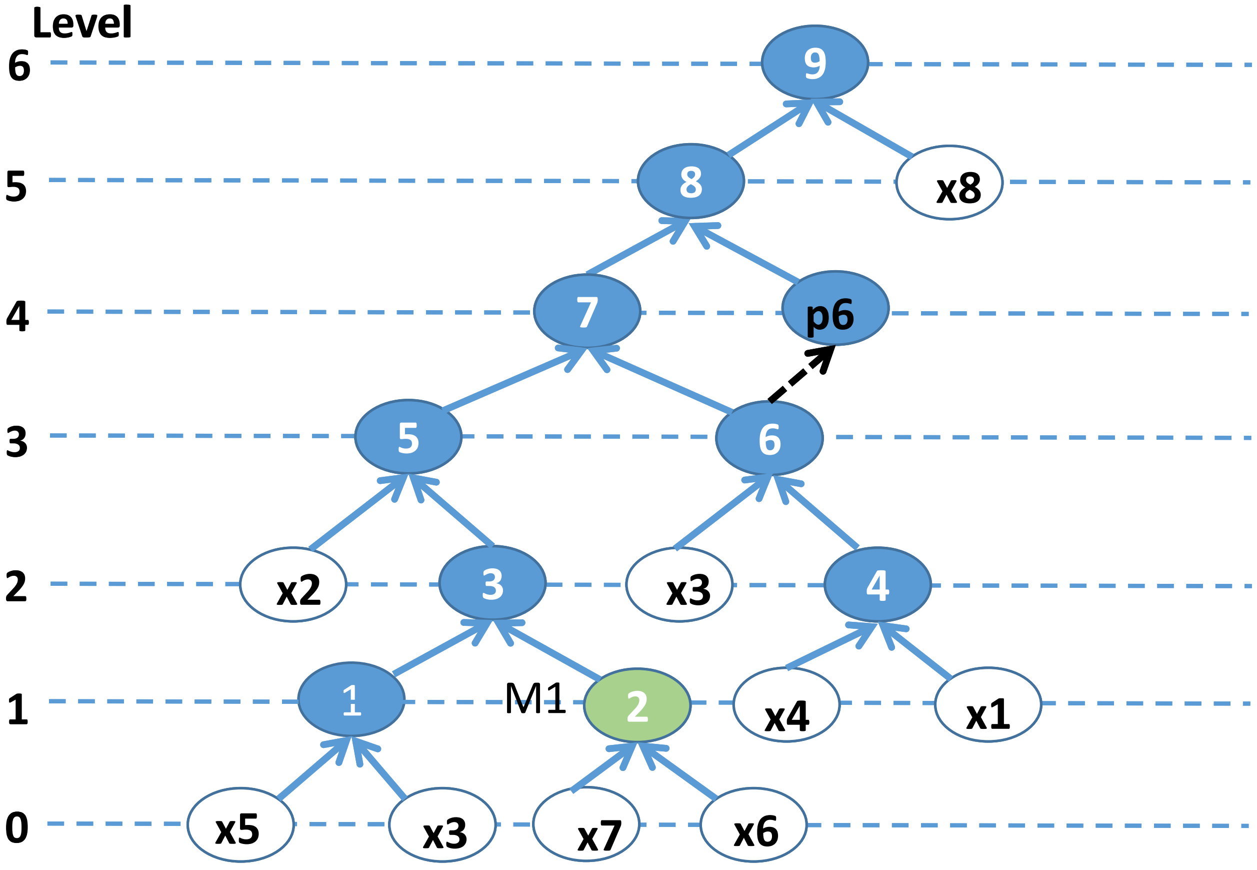
1	~~2~~	4

Reservoirs

x6
x7

Available Mixers

M2



RMS - example

At t = 1

Dynamic sorted list

x8	x3	x1	x4	x2	x5	x6	x7
32	13	~~6~~ 0	~~6~~ 0	4	1	~~1~~ 0	~~1~~ 0

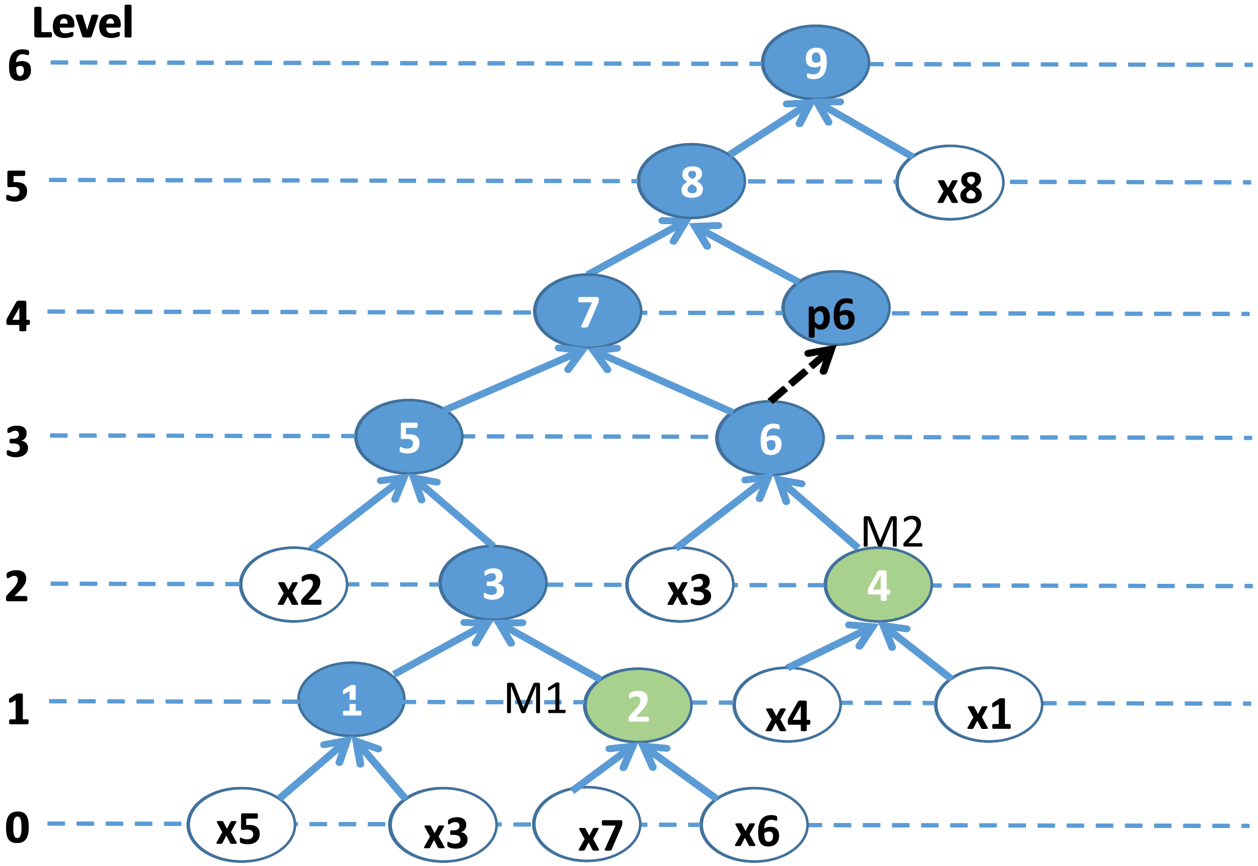
Probable list of nodes

1	~~2~~	~~4~~

Reservoirs

x6
x7
x1
x4

Available Mixers



RMS - example

After t = 1

Dynamic sorted list

x8	x3	x2	x5	x1	x4	x6	x7
32	13	4	1	0	0	0	0

Probable list of nodes

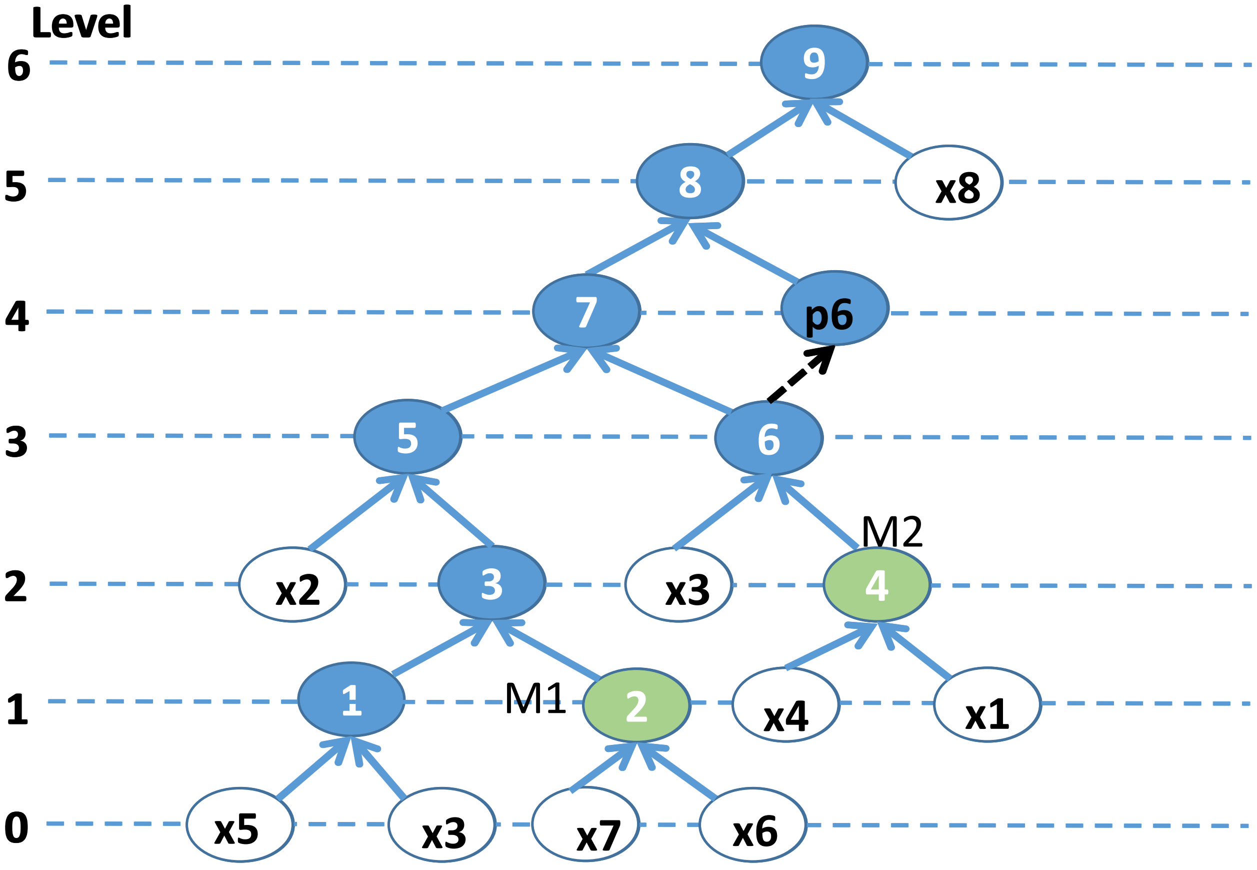
1	6

Reservoirs

x6
x7
x1
x4

Available Mixers

M1	M2



RMS - example

At t = 2

Dynamic sorted list

x8	x3	x2	x5	x1	x4	x6	x7
32	~~13~~ 0	4	~~1~~ 0	0	0	0	0

Number of switching required = 1

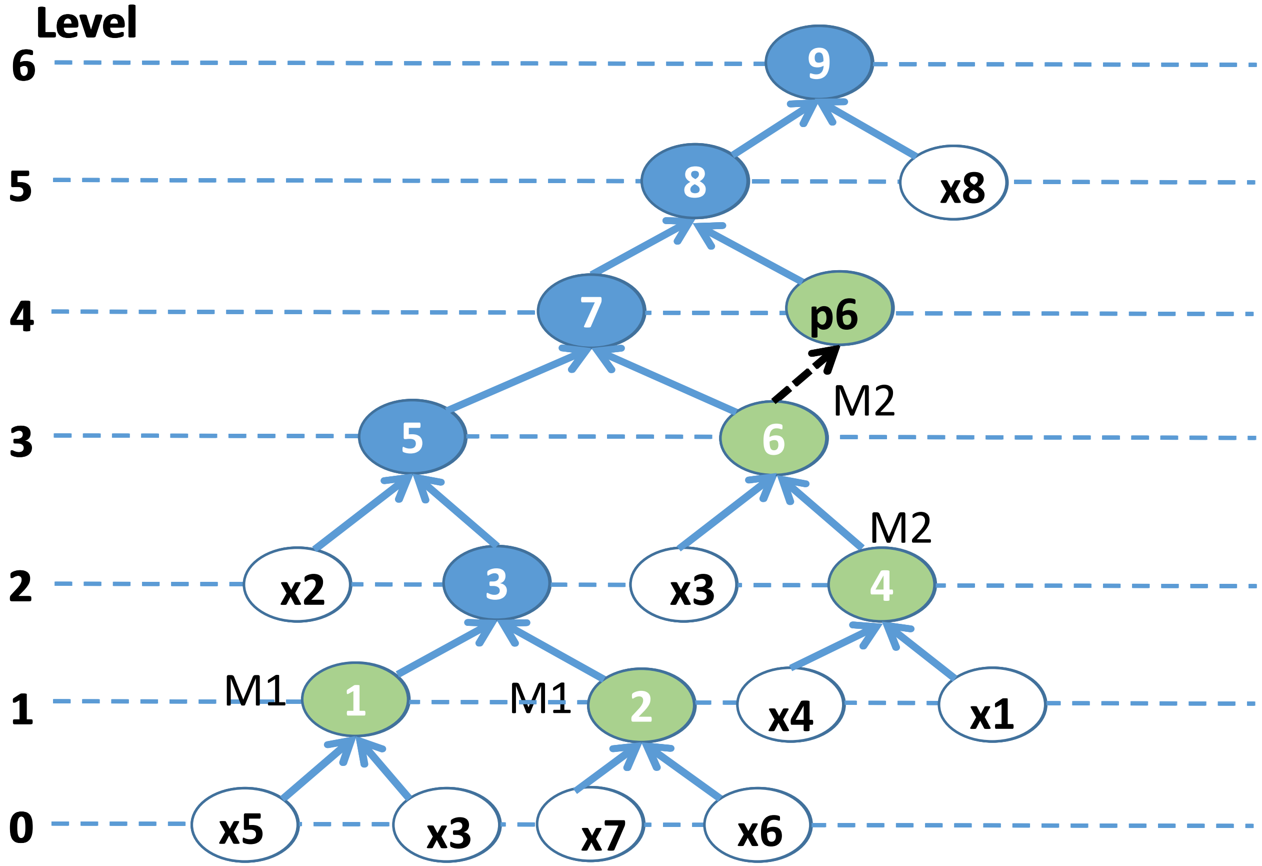
Probable list of nodes

~~1~~	~~6~~

Reservoirs

x5
x3
x2
x8

Available Mixers



RMS - example

After t = 2

Dynamic sorted list

x8	x2	x1	x3	x4	x5	x6	x7
32	4	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

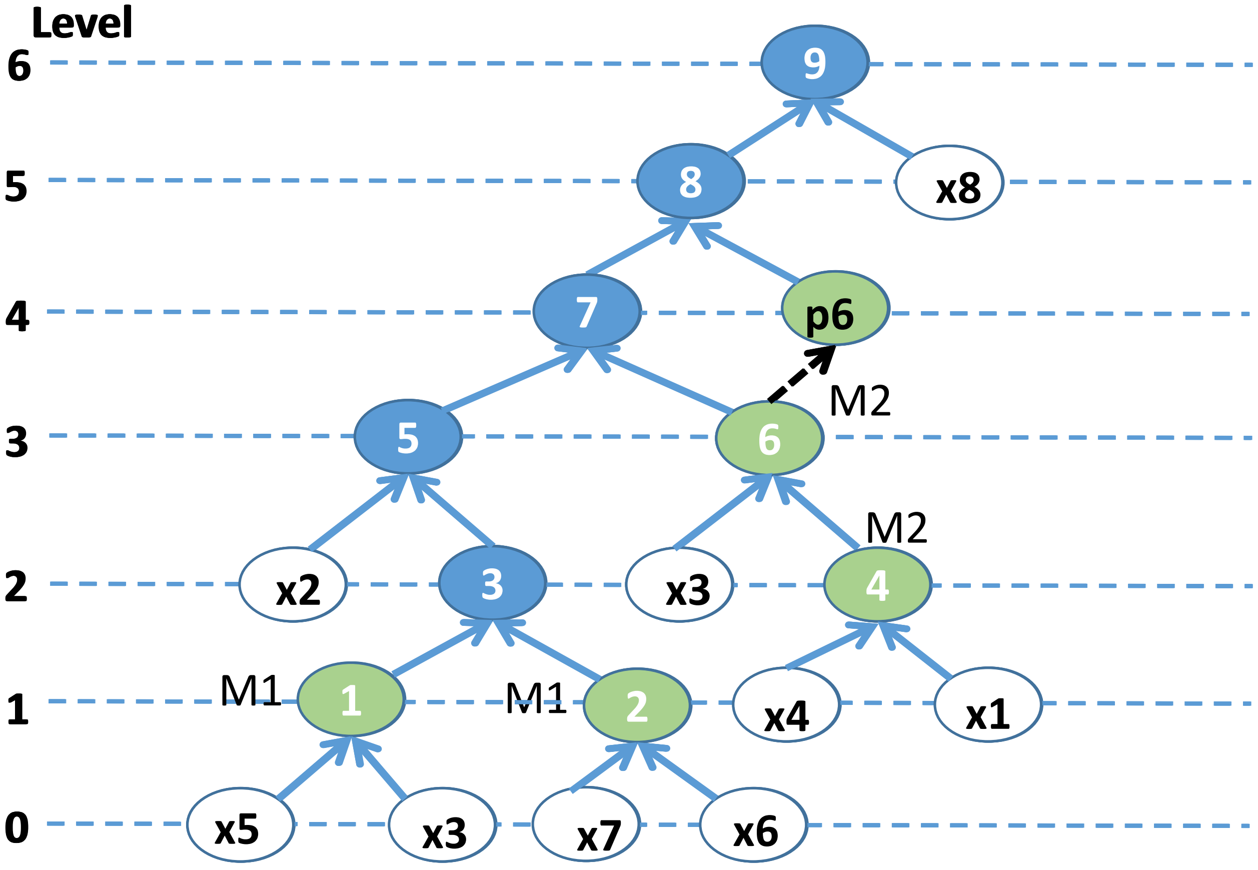
3

Reservoirs

x5
x3
x2
x8

Available Mixers

M1	M2



RMS - example

At t = 3

Dynamic sorted list

x8	x2	x1	x3	x4	x5	x6	x7
32	4	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

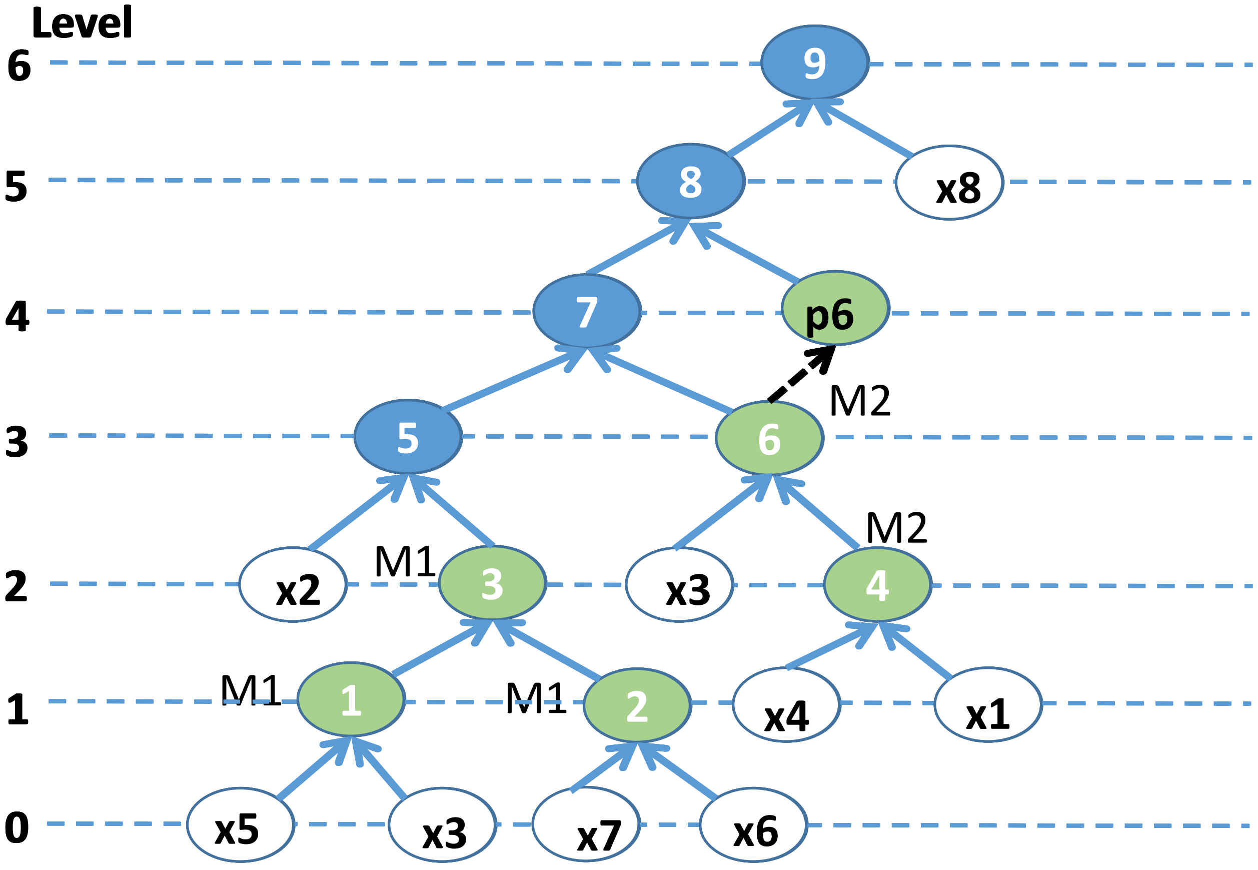
~~3~~ 5

Reservoirs

x5
x3
x2
x8

Available Mixers

M2



RMS - example

At t = 4

Dynamic sorted list

x8	x2	x1	x3	x4	x5	x6	x7
32	~~4~~ 0	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

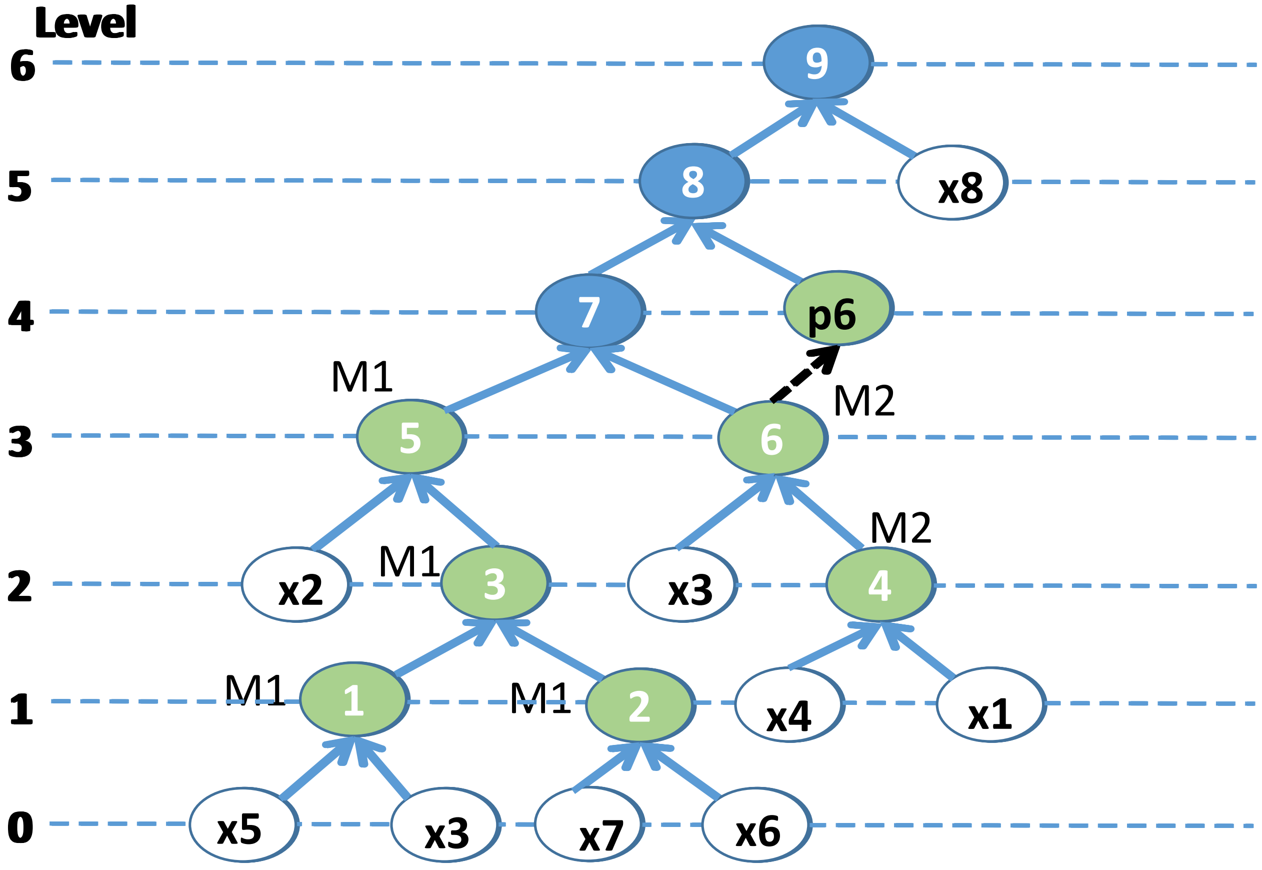
~~5~~ 7

Reservoirs

x5
x3
x2
x8

Available Mixers

M2



RMS - example

At t = 5

Dynamic sorted list

x8	x1	x2	x3	x4	x5	x6	x7
32	0	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

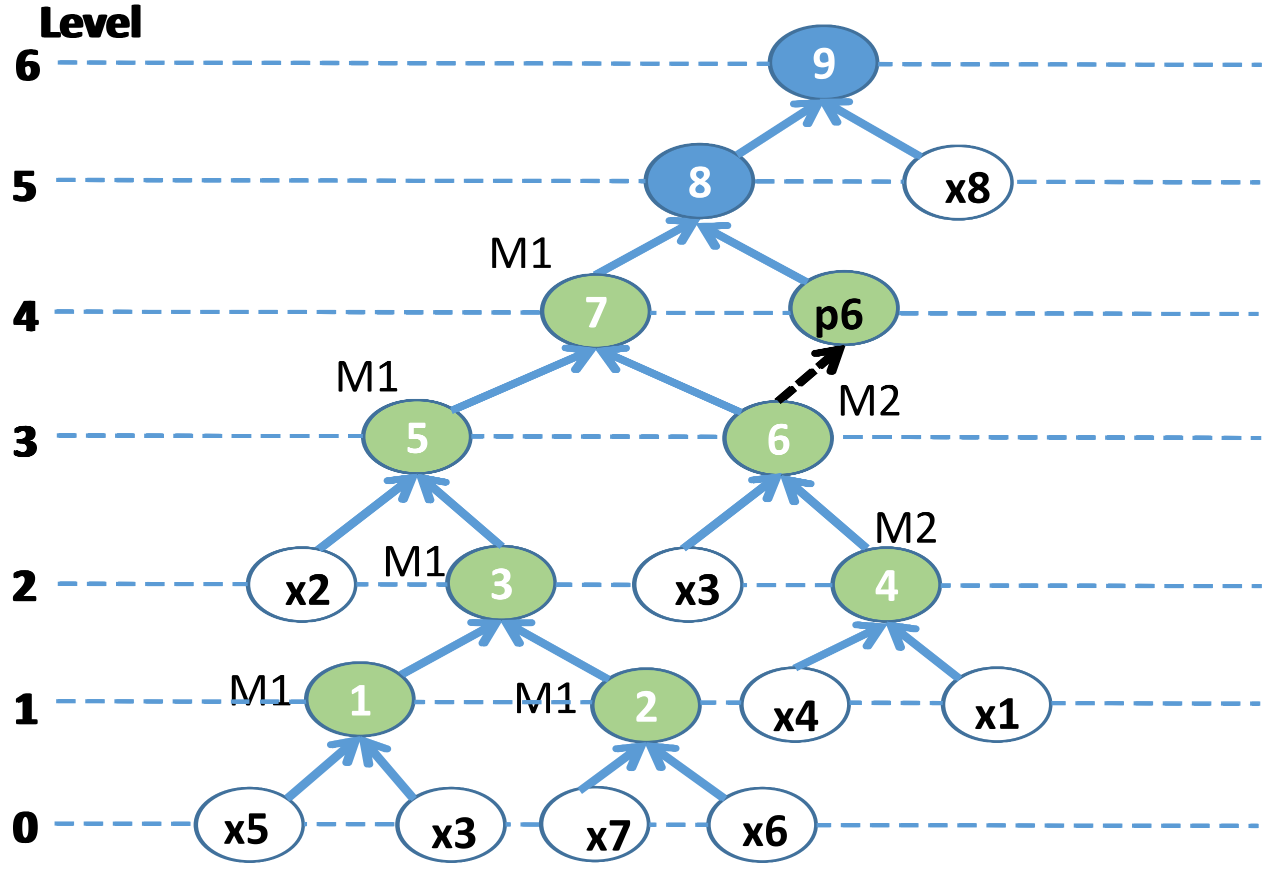
~~7~~	8

Reservoirs

x5
x3
x2
x8

Available Mixers

M2



RMS - example

At t = 6

Dynamic sorted list

x8	x1	x2	x3	x4	x5	x6	x7
32	0	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

~~8~~ 9

Reservoirs

x5

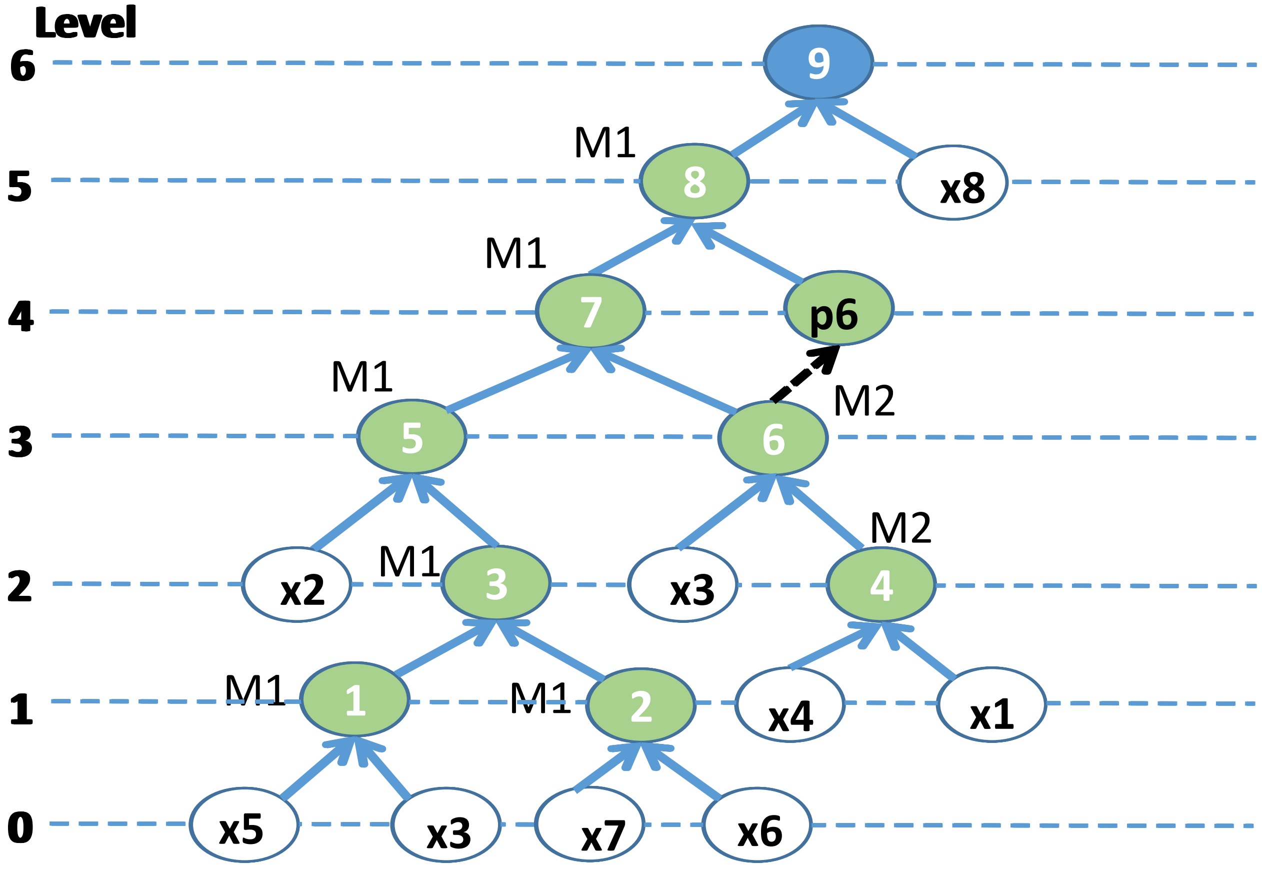
x3

x2

x8

Available Mixers

M2



RMS - example

At t = 7

Dynamic sorted list

x8	x1	x2	x3	x4	x5	x6	x7
~~32~~ 0	0	0	0	0	0	0	0

Number of switching required = 1

Probable list of nodes

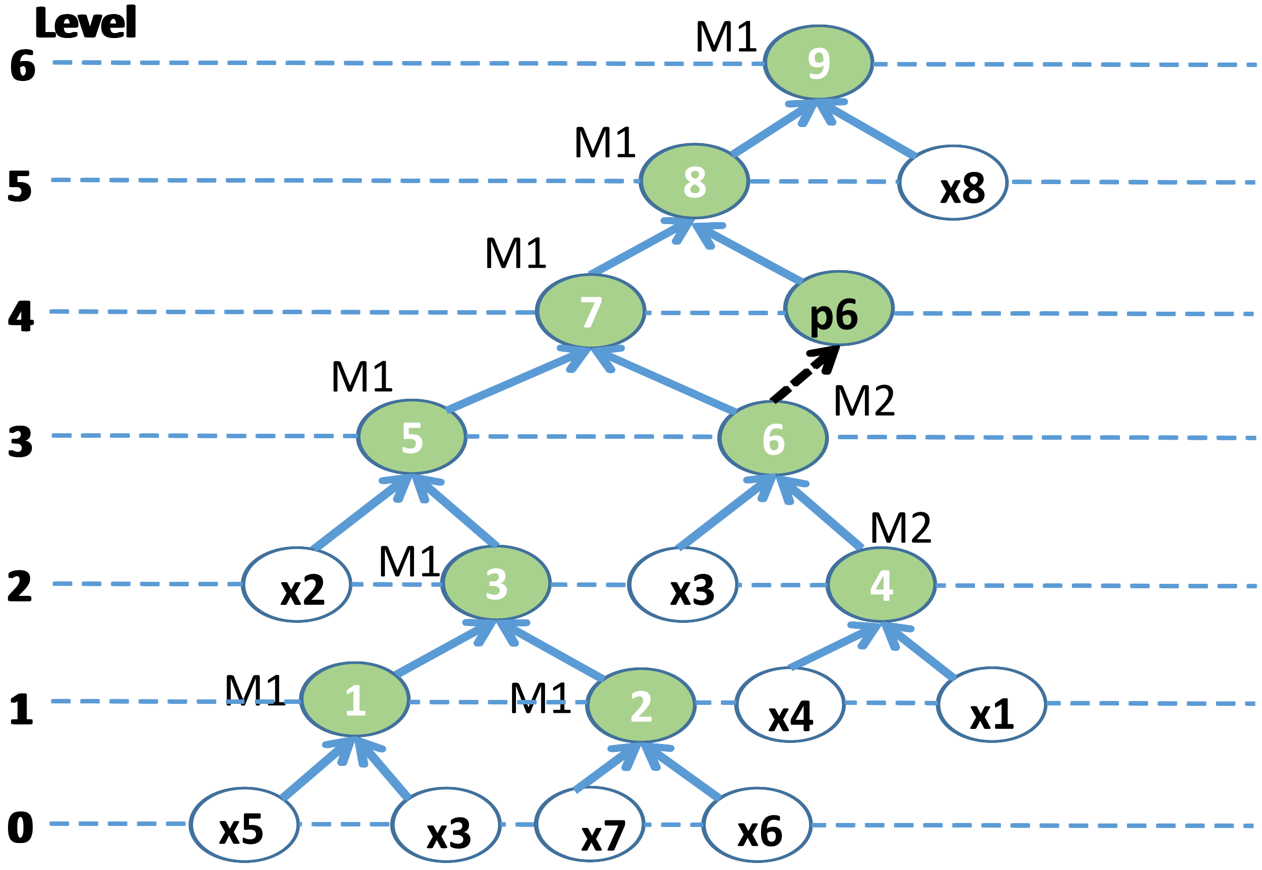
~~9~~

Reservoirs

x5
x3
x2
x8

Available Mixers

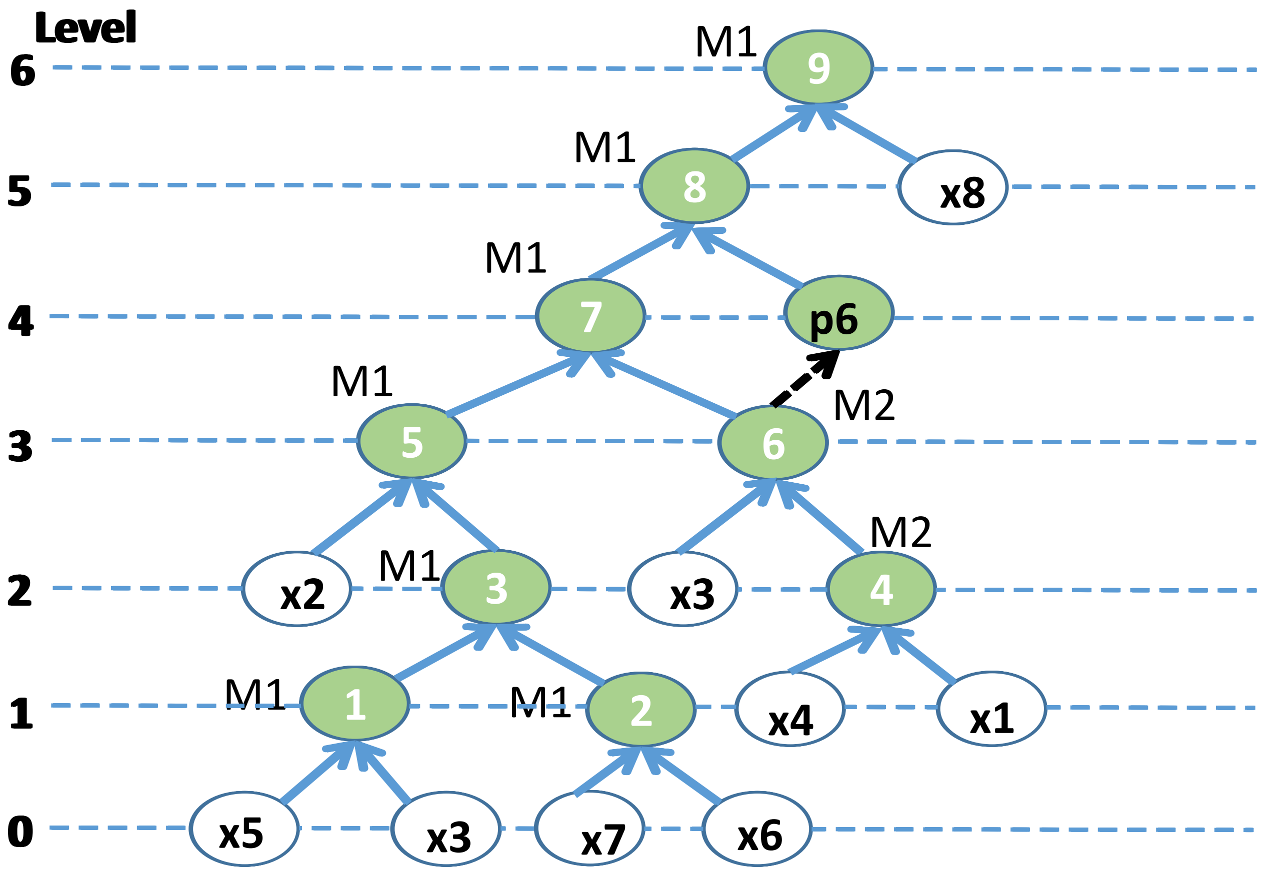
M2



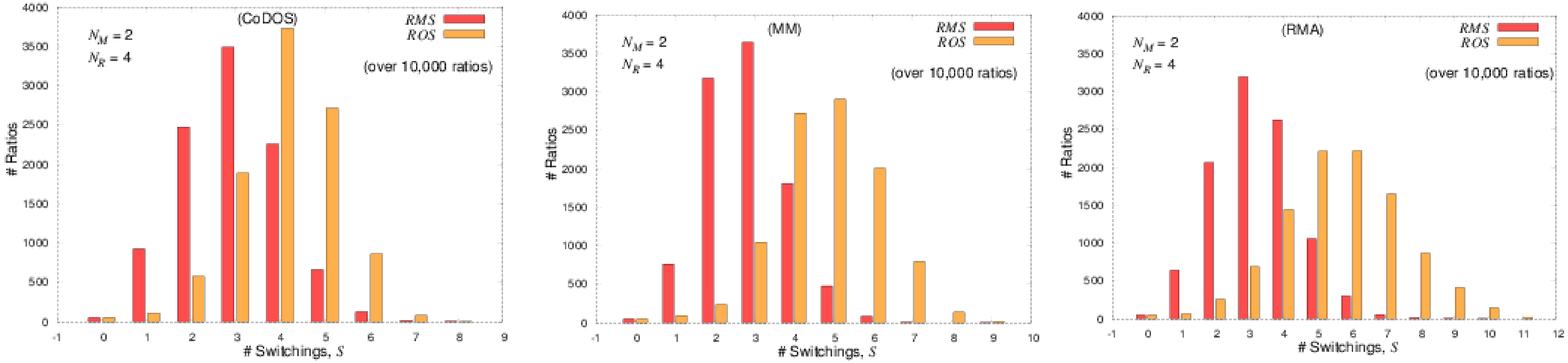
RMS - example

Total Mixing time required = 7

Number of switching required = 1



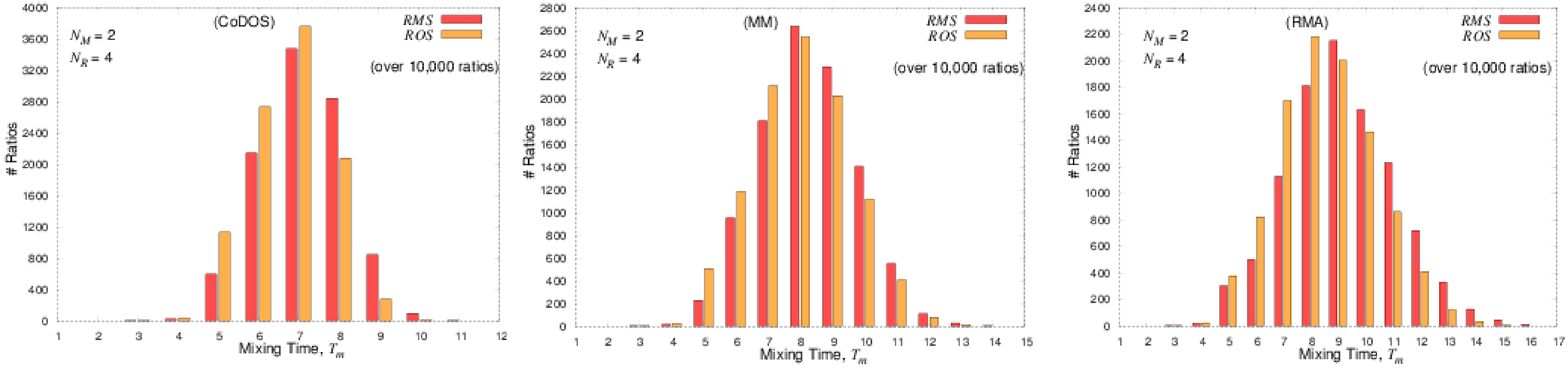
Simulation Results



Comparison of number of switching by RMS and ROS for CoDOS, MinMix, RMA mixing Trees

Test data set (10,000 ratios): http://faculty.iitr.ac.in/~sudiproy.fcs/codalab/research/microfluidics.html

Simulation Results (cont.)



Comparison of mixing time by RMS and ROS for CoDOS, MinMix, RMA mixing Trees

Test data set (10,000 ratios): http://faculty.iitr.ac.in/~sudiproy.fcs/codalab/research/microfluidics.html

Simulation Results (cont.)

Real life Bioprotocols Ratio used (Approximated ratio)	Mixing Time, T_m						Number of Switching, S						Number of Storage, q					
	MM [4]		*RMA* [8]		*CoDOS* [5]		*MM* [4]		*RMA* [8]		*CoDOS* [5]		*MM* [4]		*RMA* [8]		*CoDOS* [5]	
	RMS	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*	*RMS*	*ROS*
Plasmid DNA [15] 20:10:2:2:2:1:53 (28:14:3:3:3:1:76)	9	9	9	9	7	7	1	5	1	2	1	4	1	1	1	1	1	1
Splinkerette PCR [16] 40:10:1:1:48 (51:13:1:1:62)	7	7	7	7	7	7	1	3	1	3	1	3	0	0	0	0	1	1
Touchdown PCR [17] 500:300:1000:500:100:100:25:2475 (13:8:26:13:3:3:1:61)	11	10	14	12	8	8	5	5	3	6	4	2	3	1	2	3	3	2
Silver- Restriction Digest [17] 70:10:10:2:2:6 (90:13:13:3:3:6)	9	9	12	10	7	7	2	5	1	3	1	1	3	1	2	2	1	1
Molecular barcodes - PCR [17] 100:20:20:20:20:50:50:100:4:616 (13:3:3:3:3:6:6:13:1:77)	12	12	15	14	9	9	5	6	4	10	2	5	4	3	2	4	3	2

Comparative results of RMS over ROS for some example ratios used in real-life bioprotocols

Conclusions and Future Work

- Reservoir switching is required when number of reagents are more than reservoirs
- Reservoir switching requires more time than other fluidic operations
- RMS can reduce the number of switching as compared to ROS schemes with slight increase in mixing time
- As a future work, reducing the storage requirement can also be added to the optimization criteria

References

[1] K. Chakrabarty and F. Su, Digital Microfluidic Biochips: Synthesis, Testing and Reconfiguration Techniques. CRC Press, 2007.

[2] Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty, “A Reagent-Saving Mixing Algorithm for Preparing Multiple-Target Biochemical Samples Using Digital Microfluidics,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 31, no. 11, pp. 1656–1669, 2012.

[3] S. Roy, B. B. Bhattacharya, and K. Chakrabarty, “Optimization of Dilution and Mixing of Biochemical Samples using Digital Microfluidic Biochips,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 29, no. 11, pp. 1696–1708, 2010.

[4] W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe, “Abstraction Layers for Scalable Microfluidic Biocomputing,” Natural Computing, vol. 7, no. 2, pp. 255– 275, 2008.

[5] C.-H. Liu, H.-H. Chang, T.-C. Liang, and J.-D. Huang, “Sample Preparation for Many-Reactant Bioassay on DMFBs using Common Dilution Operation Sharing,” in Proc. of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2013, pp. 615–621.

References

[6] S. Kumar, S. Roy, P. P. Chakrabarti, B. B. Bhattacharya, and K. Chakrabarty, “Efficient Mixture Preparation on Digital Microfluidic Biochips,” in Proc. of the IEEE International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), 2013, pp. 205–210.

[7] S. Roy, B. B. Bhattacharya, S. Ghoshal, and K. Chakrabarty, “Theory and Analysis of Generalized Mixing and Dilution of Biochemical Fluids Using Digital Microfluidic Biochips,” ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 11, no. 1, pp. 2.1–2.33, 2014.

[8] S. Roy, P. P. Chakrabarti, S. Kuamr, K. Chakrabarty, and B. B. Bhattacharya, “Layout-Aware Mixture Preparation of Biochemical Fluids on Application-Specific Digital Microfluidic Biochips,” ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 20, no. 3, pp. 45.1–45.34, 2015.

[9] F. Su and K. Chakrabarty, “Architectural-Level Synthesis of Digital Microfluidicsbased Biochips,” in Proc. of the IEEE/ACM International Conference on Computer- Aided Design (ICCAD), 2004, pp. 223–228.

[10] L. Luo and S. Akella, “Optimal Scheduling of Biochemical Analyses on Digital Microfluidic Systems,” IEEE Transactions on Automation Science and Engineering (TASE), vol. 8, no. 1, pp. 216–227, 2011.

[11] D. Grissom and P. Brisk, “Path scheduling on Digital Microfluidic Biochips,” in Proc. of the IEEE/ACM Design Automation Conference (DAC), 2012, pp. 26–35.

References

[12] S. Srigunapalan, I. A. Eydelnant, C. A. Simmons, and A. R. Wheeler, “A Digital Microfluidic Platform for Primary Cell Culture and Analysis,” Lab Chip, vol. 12, pp. 369–375, 2012.

[13] P. Y. Paik, V. K. Pamula, M. G. Pollack, and R. B. Fair, “Electrowetting-Based Droplet Mixers for Microfluidic Systems,” Lab Chip, vol. 3, no. 1, pp. 28–33, 2003.

[14] C. Rensch, S. Lindner, R. Salvamoser, S. Leidner, C. Bold, V. Samper, D. Taylor, M. Baller, S. Riese, P. Bartenstein, C. Wangler, and B. Wangler, “A Solvent Resistant Lab-on-Chip Platform for Radiochemistry Applications,” Lab Chip, vol. 14, pp. 2556–2564, 2014.

[15] Preparation of Plasmid DNA by Alkaline Lysis with SDS: Minipreparation, Cold Spring Harb Protocols, http://cshprotocols.cshlp.org/content/2006/1/pdb.prot4084.citation, 2006.

[16] A. G. Uren, H. Mikkers, J. Kool, L. van der Weyden, A. H. Lund, C. H. Wilson, R. Rance, J. Jonkers, M. van Lohuizen, A. Berns, and D. J. Adams, “A High- Throughput Splinkerette-PCR Method for the Isolation and Sequencing of Retroviral Insertion Sites,” Nature Protocols, vol. 4, no. 5, pp. 789–798, 2009.

[17] OpenWetWare, 2009, http://openwetware.org/wiki/Main Page.

THANK YOU

Any Questions?