A Parallel Complex Coloring Algorithm for Scheduling of Input-Queued Switches

Lingkang Wang, Tong Ye, Tony T. Lee, and Weisheng Hu

State Key Lab of Advanced Optical Communications and Networks

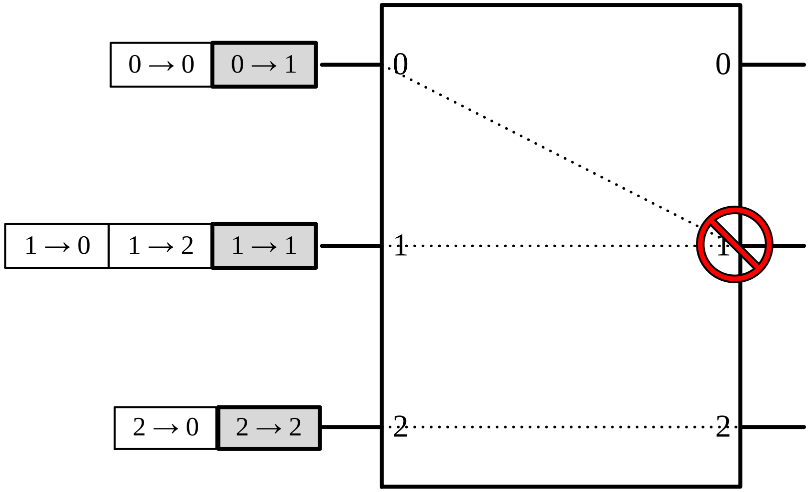
Shanghai Jiao Tong University

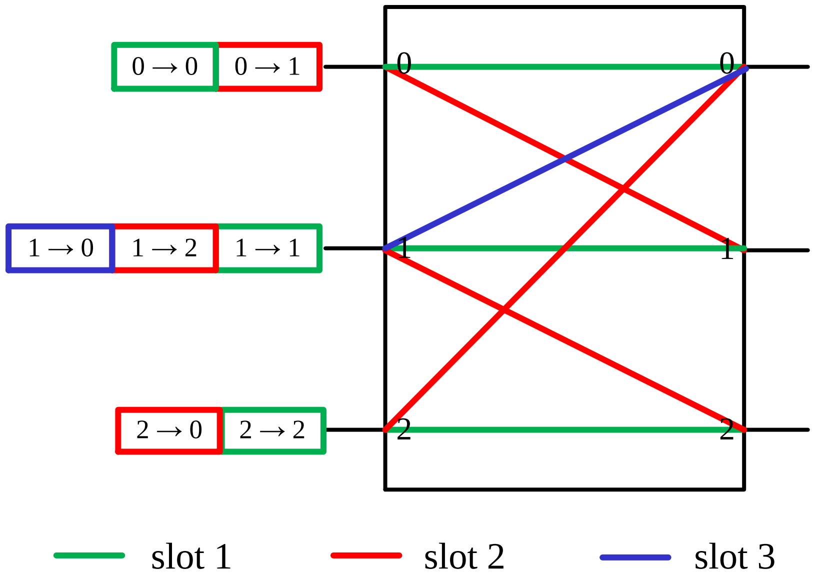
Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms

Scheduling Problem[1]

- Cell scheduling is indispensable to properly set up connection patterns to avoid output contentions.

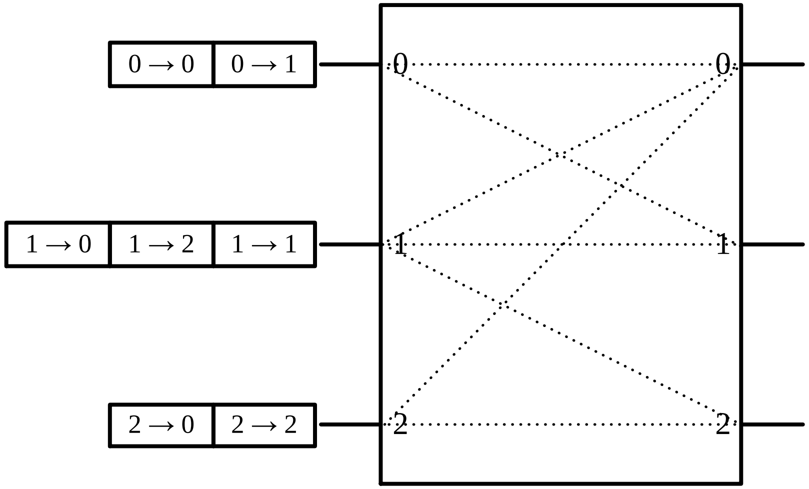


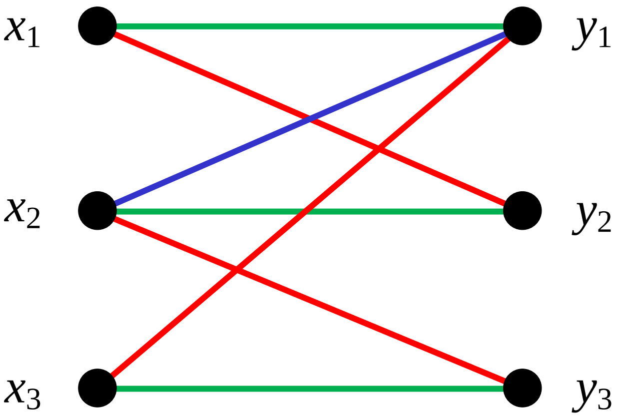


[1] H. J. Chao, Z. Jing, and S. Y. Liew, *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 46–54, Oct. 2003.

Bipartite Graph Model

- The cell-scheduling problem can be formulated as the bipartite-graph matching (or edge coloring) problem.





Vertex x_i (y_j): input (output) port i (j)
Edge e_{ij}: arrival packets from x_i to y_j
Color: assigned timeslot for transmission

Current Scheduling Algorithms

- Maximum Size Matching (iSLIP[1])
  - Pros: 100% throughput under any uniform traffic
  - Cons: $O(N \log N)$ on-line complexity
- Maximum Weighted Matching (iLQF, iOCF[2])
  - Pros: 100% throughput under any traffic
  - Cons: $O(N^2 \log N)$ on-line complexity
- Frame-based scheduling (Fair-Frame[3])
  - Pros: 100% throughput under any traffic
  - Cons: $O(Nf)$ on-line complexity (f: frame size)

[1] N. McKeown, *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.

[2] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

[3] M. J. Neely, E. Modiano, Y. S. Cheng, *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 657–668, 2007.

Our Contribution

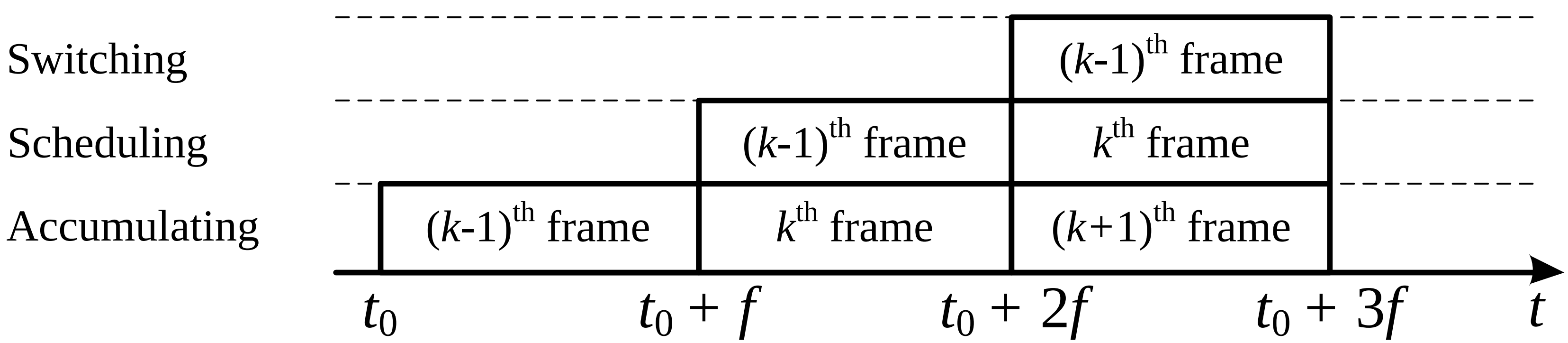
- A frame-based scheduling algorithm based on an algebraic edge coloring method
  - $O(\log N)$ time complexity per timeslot
  - Nearly 100% throughput
  - Microsecond level latency
  - Work well under any traffic patterns

Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
  - Frame-based Scheduling
  - Complex Coloring of Bipartite Graph
    - Properties of Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms

Frame-based Scheduling

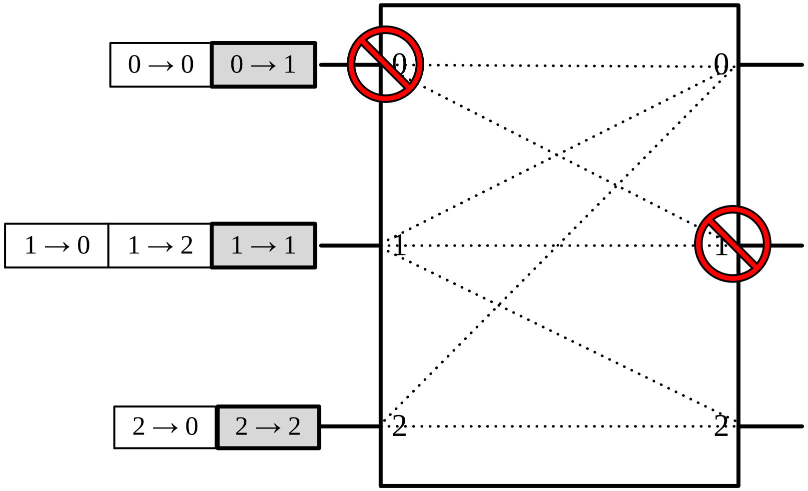
- Assumptions
  - Time is slotted and packet size is fixed.
  - A batch of f consecutive timeslots is scheduled together.
- Pipelining implementation[1]



[1] B. Wu, K. L. Yeung, M. Hamdi, X. Li, *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 632-645, Apr. 2009.

Frame-based Scheduling

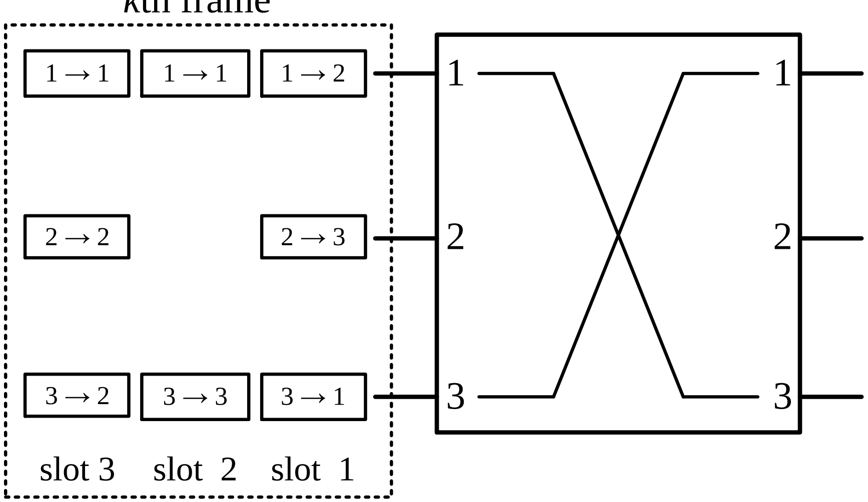
- Constraint
  - In each timeslot, at most one packet can be sent from each input and at most one packet can be received by each output.
  - It corresponds to the constraint of edge coloring problem that two edges incident to the same vertex must be colored with distinct colors.



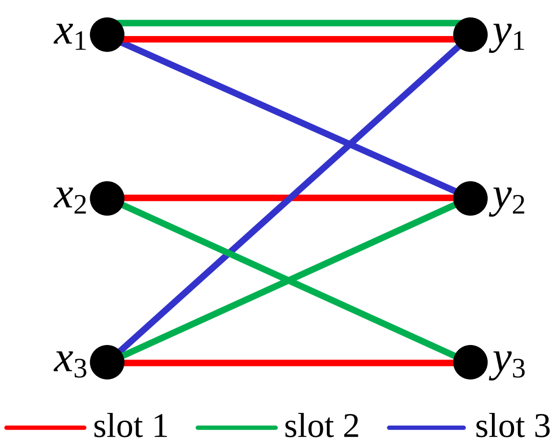
[1] H. J. Chao, Z. Jing, and S. Y. Liew, *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 46–54, Oct. 2003.

Bipartite Graph Model

- An $N \times N$ input-queued switch
  - Input/output ports ⇔ vertex set X/Y
  - Packets ⇔ edge set
  - Timeslots ⇔ color set



(a) A 3×3 frame-based packet switch

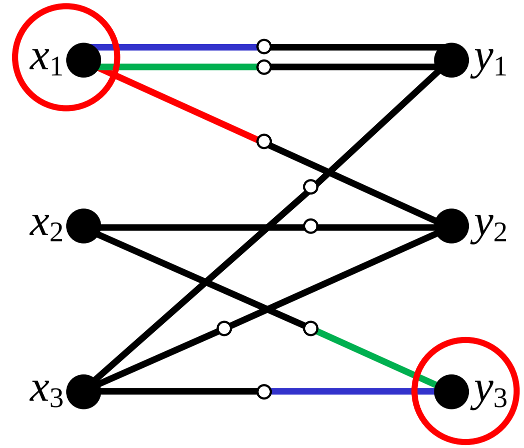


(b) The corresponding bipartite graph model

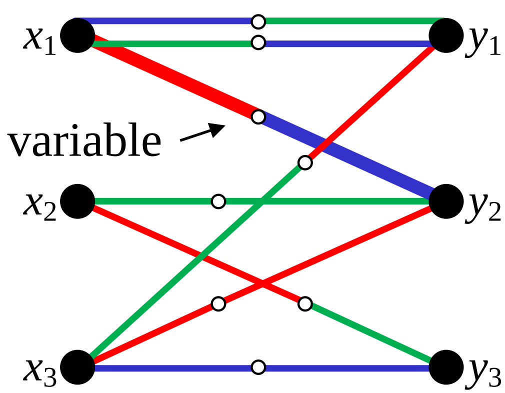
Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
  - Frame-based Scheduling
  - Complex Coloring of Bipartite Graph
    - Properties of Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms

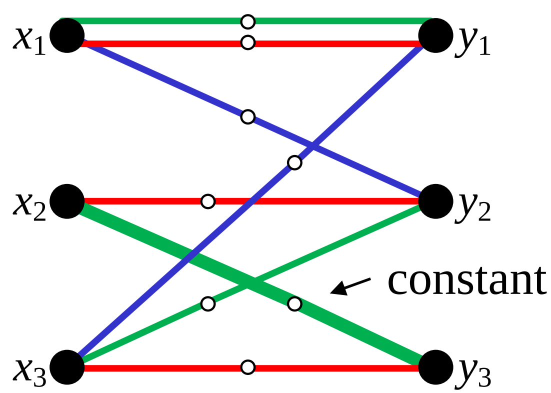
Edge Coloring Constraints



- Vertex constraint
  - Colors assigned to links incident to the same vertex are all distinct.
- Edge constraint
  - Variable-colored edge
  - Constant-colored edge



Consistent coloring of G

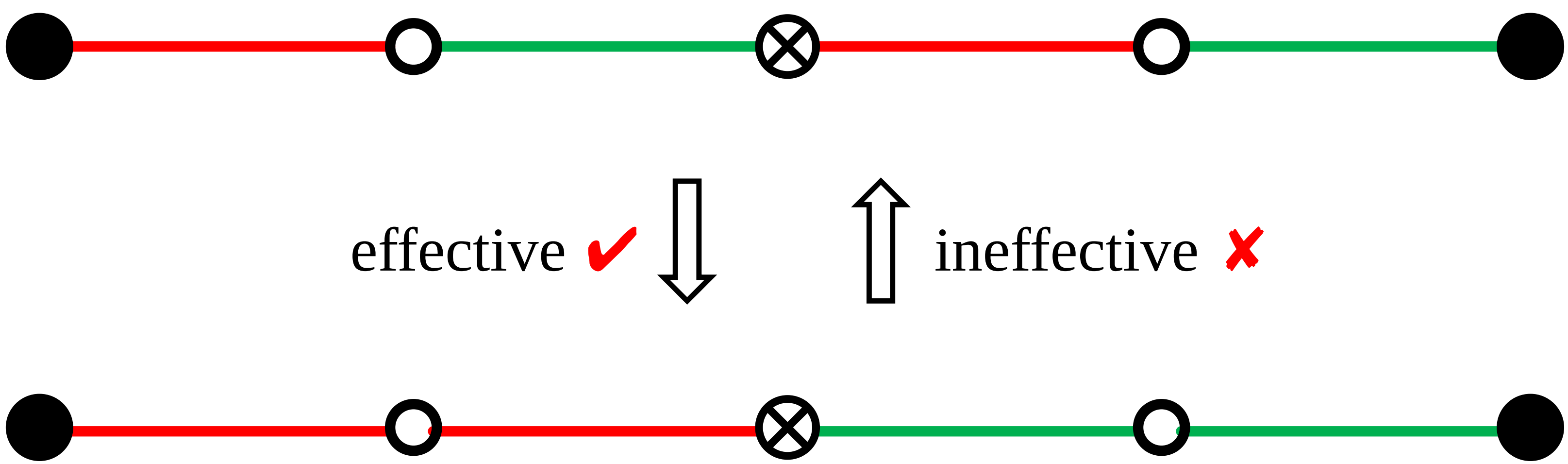


Proper coloring of G

[1] T. Lee, Y. Wan, and H. Guan, Int. J. Comput. Math. 90 (2013), 228–245.

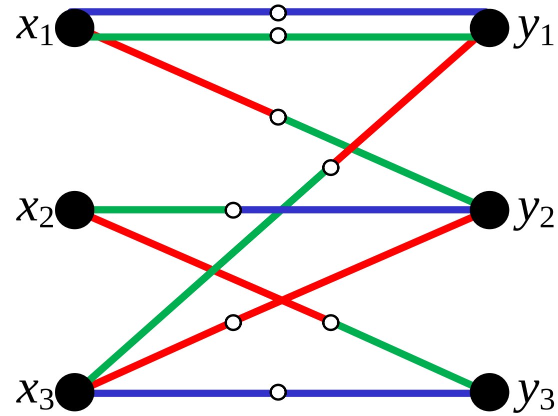
Color-Exchange Operation

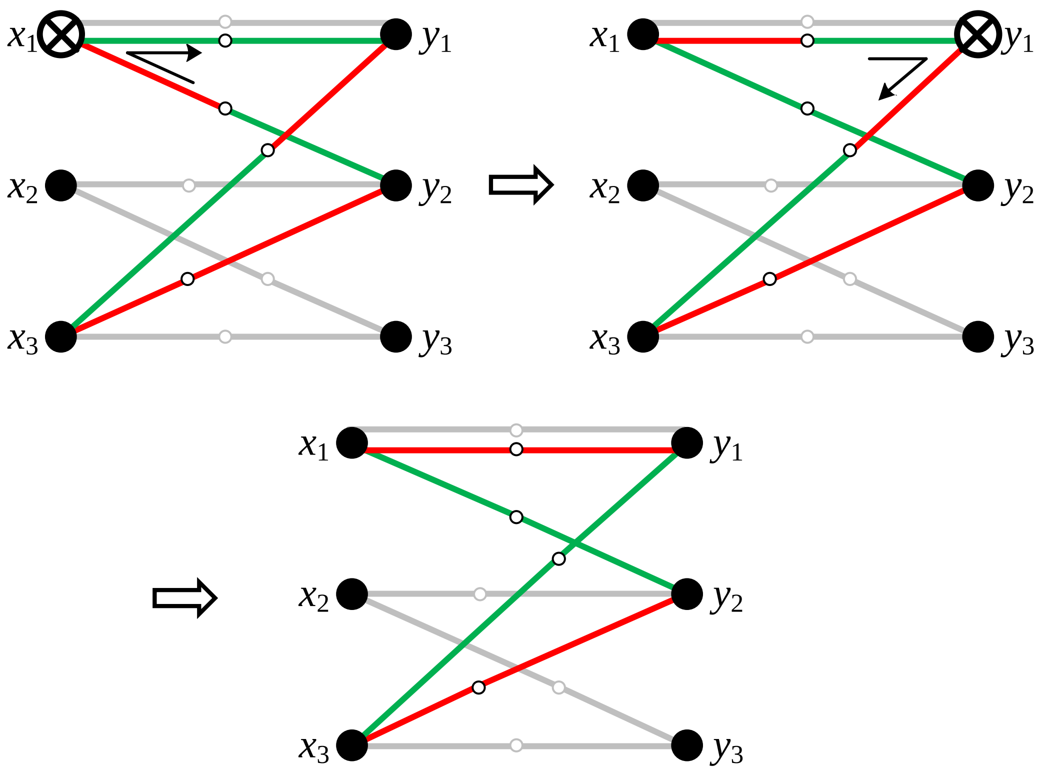
- Color-exchange operation preserves the consistency of vertex constraint.
- A color-exchange operation is effective if it does not increase the number of variables.



(a, b) Subgraph

- A (a, b) variable is only allowed to move within a two-colored (a, b) subgraph to meet another variable.





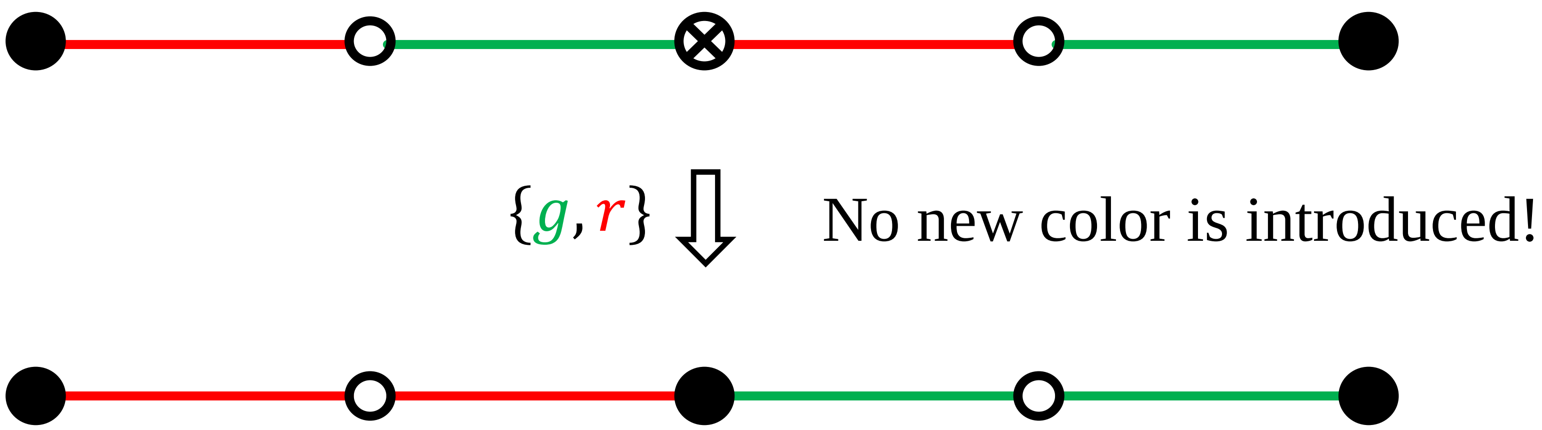
Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
  - Frame-based Scheduling
  - Complex Coloring of Bipartite Graph
    - Properties of Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms
- Conclusion

Optimality

- An optimal proper coloring of a bipartite graph only uses Δ colors. (Δ: the maximum degree)

- A consistent coloring can be easily achieved by Δ colors.

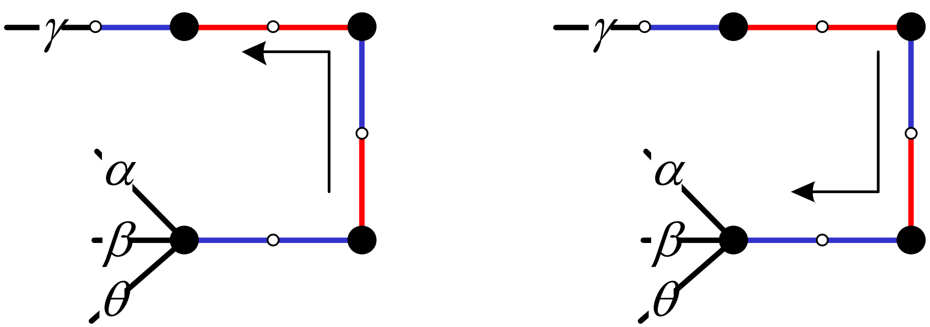


[1] T. Lee, Y. Wan, and H. Guan, Int. J. Comput. Math. 90 (2013), 228–245.

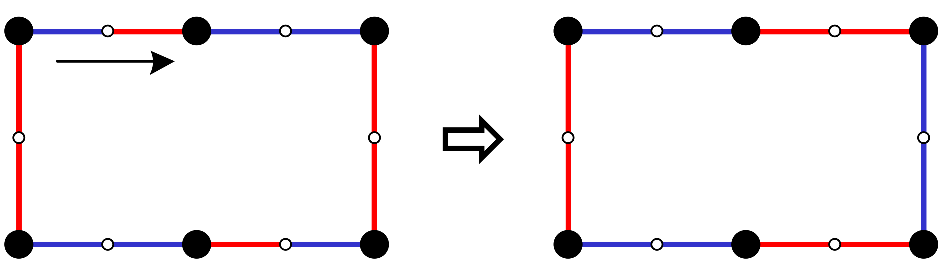
Optimality

- An optimal proper coloring of a bipartite graph only uses Δ colors. (Δ: the maximum degree)

- All variables of a bipartite graph can be eliminated by Kempe walks.



Variables on open path can always be eliminated! [1]

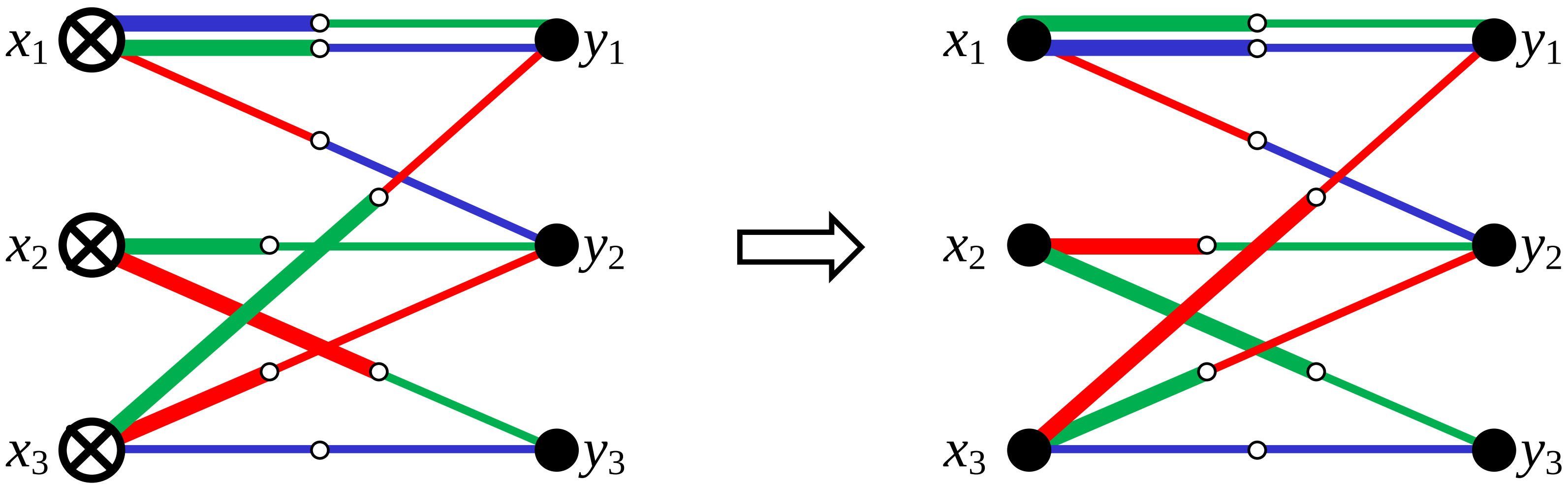


Only even cycles exist which contain even # of variables! [1]

[1] T. Lee, Y. Wan, and H. Guan, Int. J. Comput. Math. 90 (2013), 228–245.

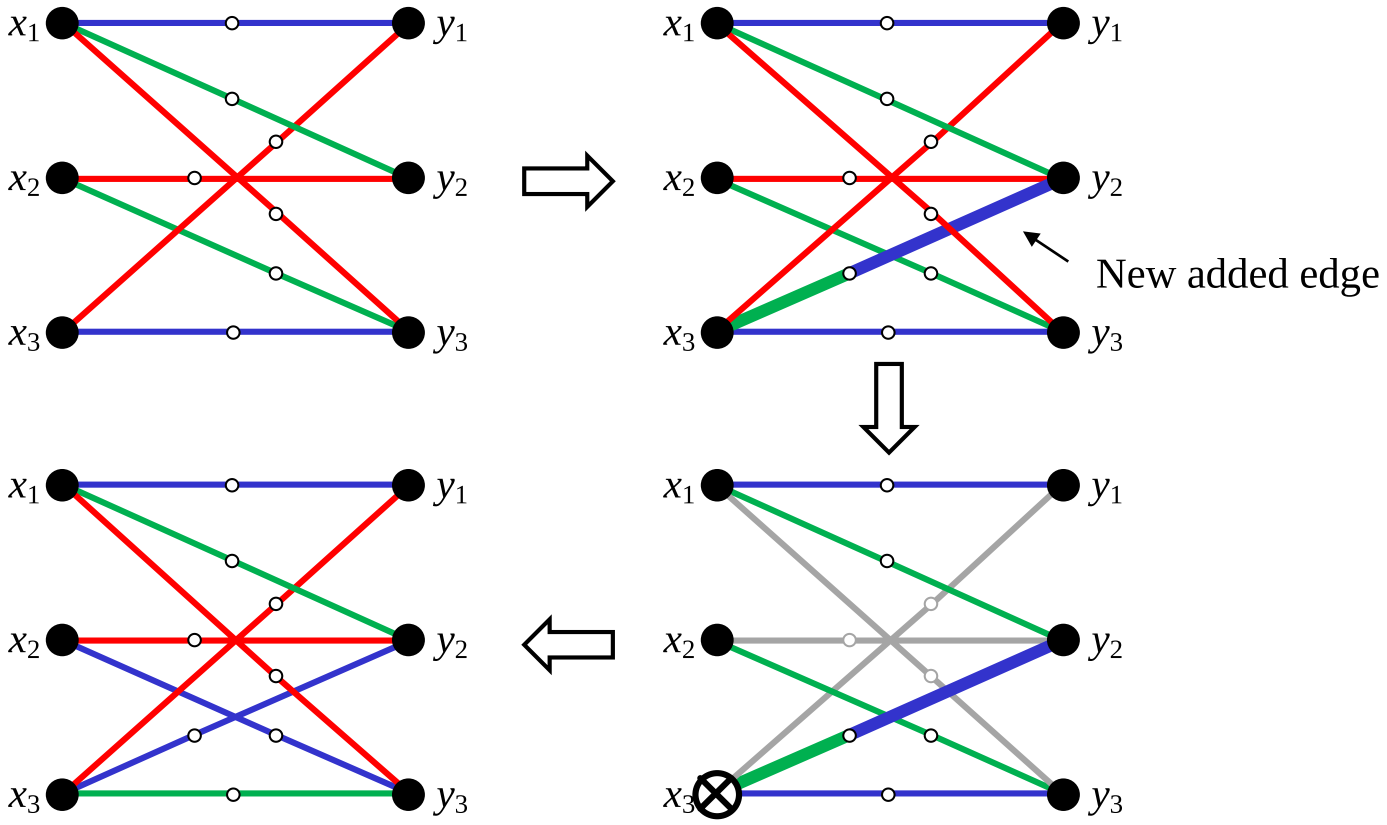
Parallelizability

- Variables can be eliminated by color-exchange simultaneously.



Rearrangeability

- When new edges are added, only partial changes of the existing coloring are needed.

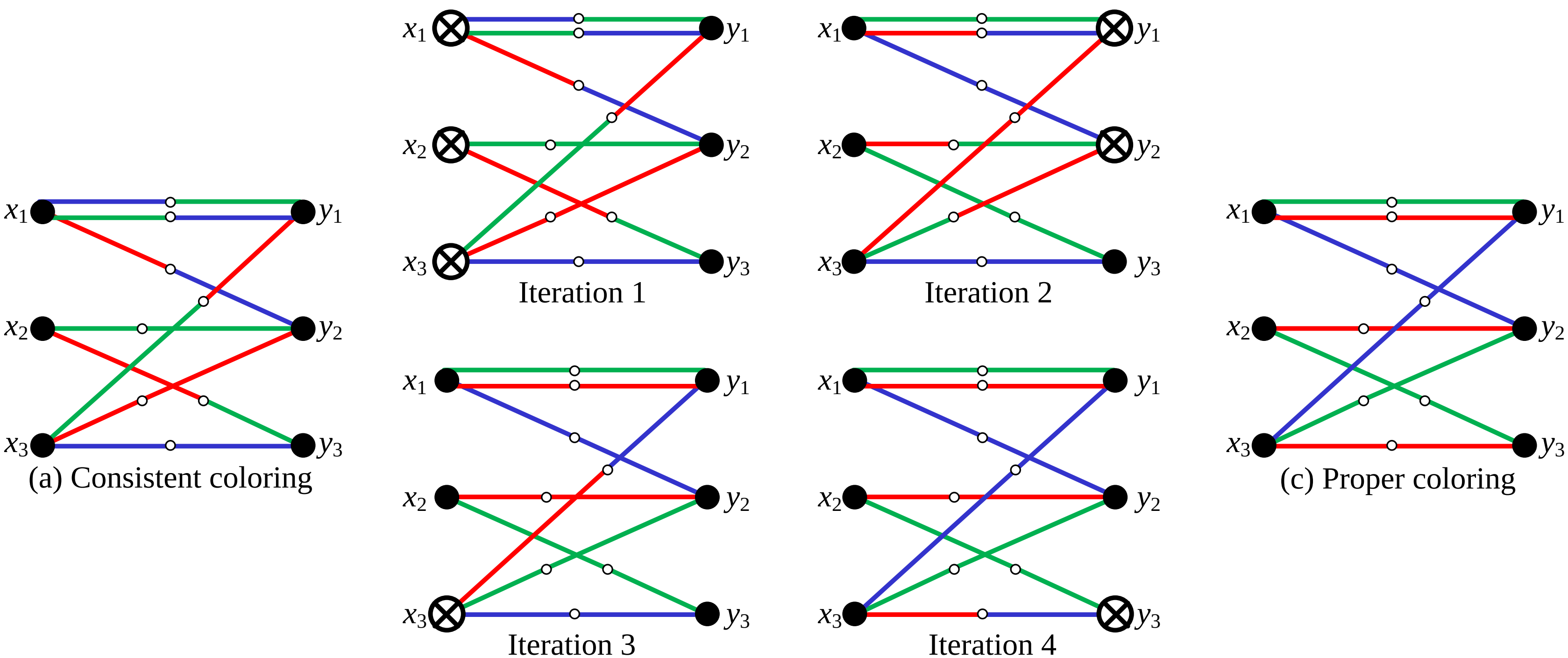


Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
  - Deadlock Variables
  - Stopping Rule
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms

Parallel Complex Coloring

- ***Principle of parallelization***
  - For $G = (X \cup Y, E)$, simultaneous color exchanges can be performed on vertices in X and Y alternatively.



(b) Parallel processing

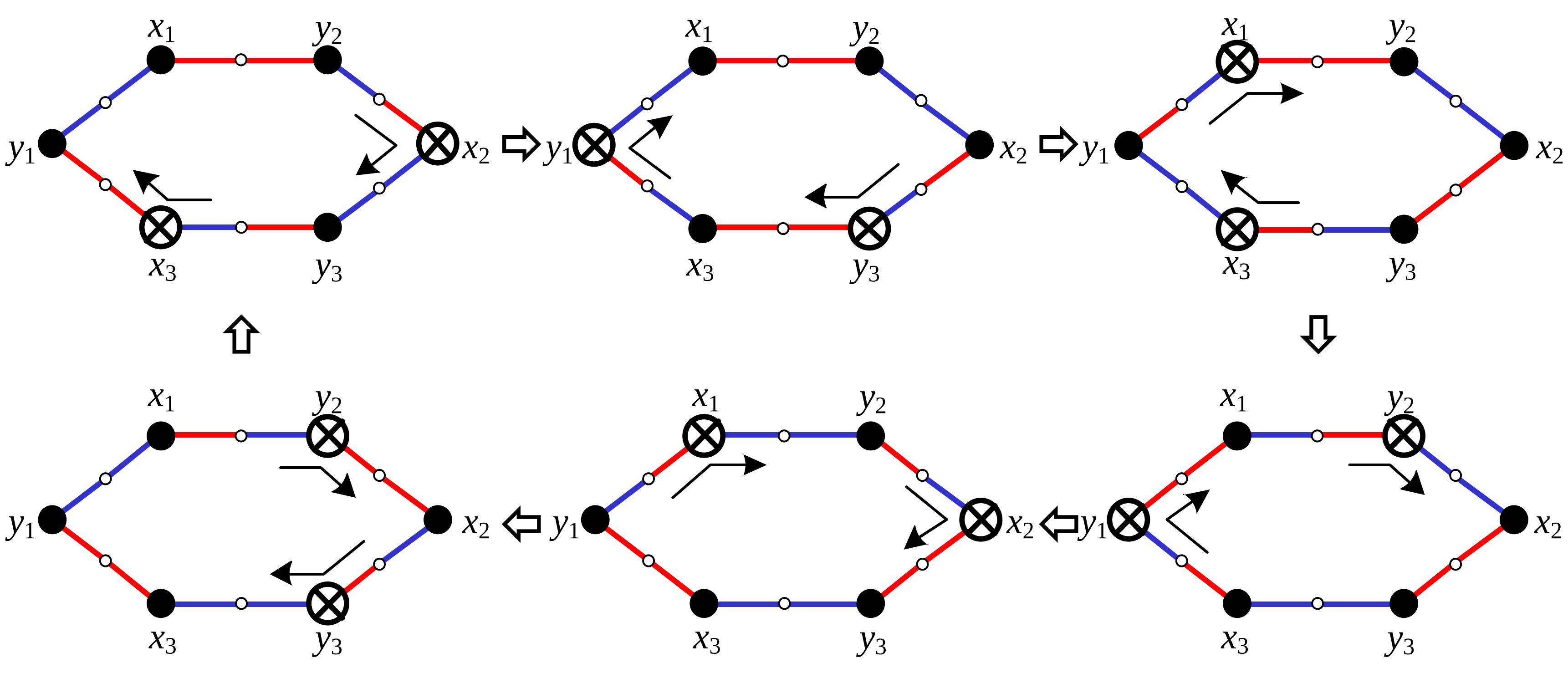
High efficiency of variable eliminations!

Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
  - Deadlock Variables
  - Stopping Rule
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms
- Conclusion

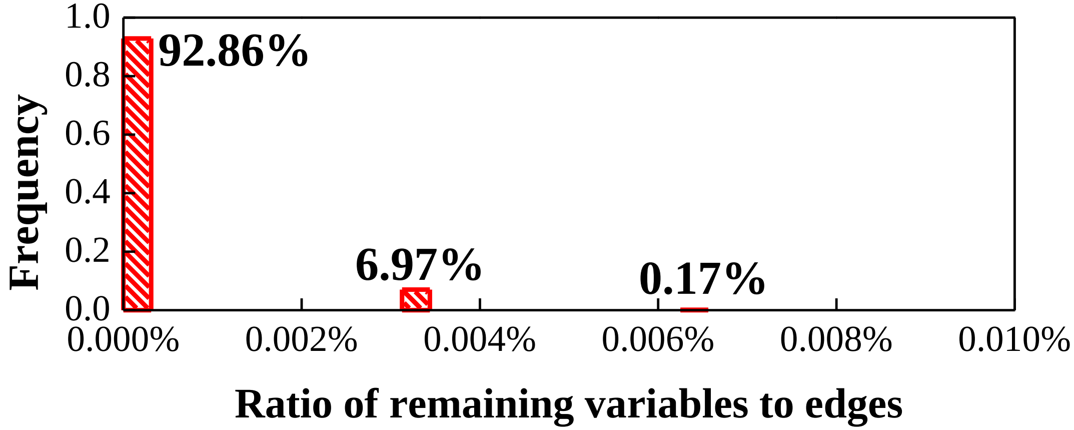
Infinite Loop

- When variables step forward in the same direction, they may be trapped in an infinite loop.

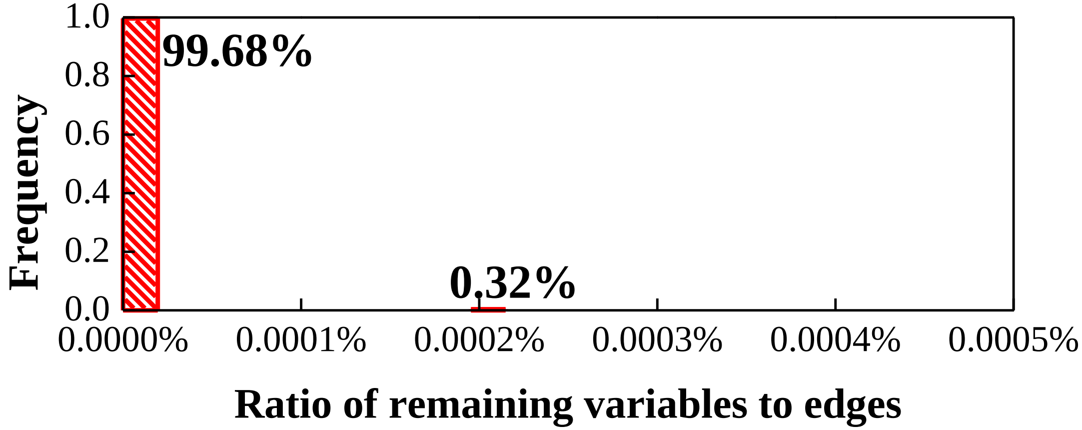


Deadlock Percentages V.S. Simulation Time

- **Variables in deadlock are rare.**



(a) $|V|$=128, Δ=1000



(b) $|V|$=1024, Δ=2000

Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
  - Deadlock Variables
  - Stopping Rule
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms
- Conclusion

Notation

- Variable density

$$R(t) = \frac{\text{\# of variables}}{\text{\# of edges}} \text{(after } t \text{ iterations)}$$

- Variable elimination rate

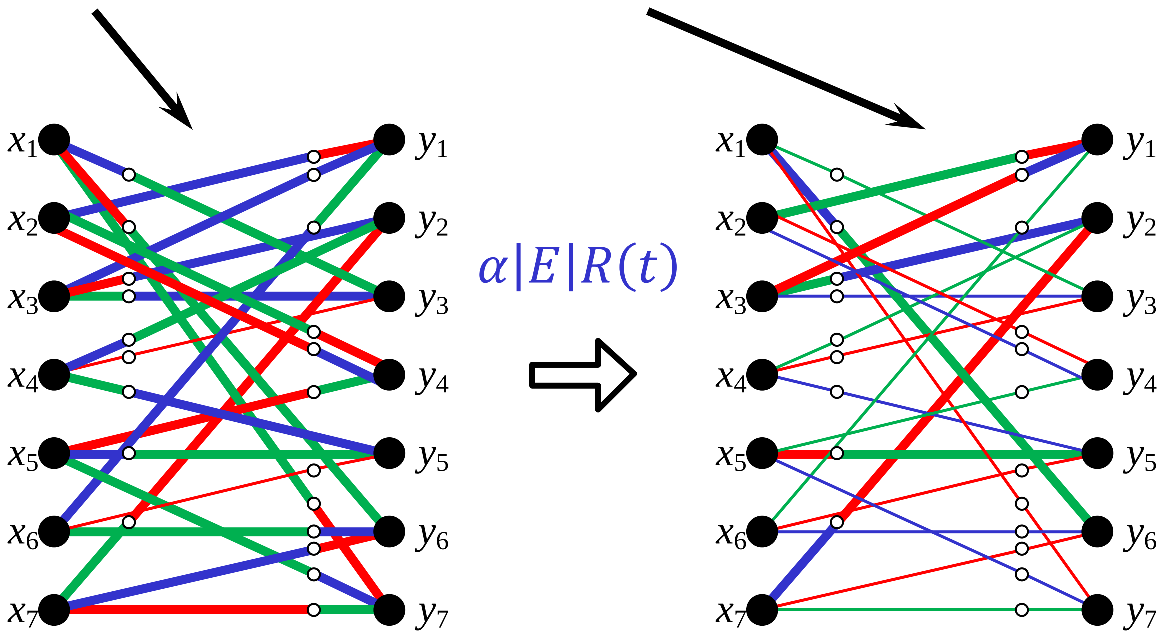
$$\alpha(t) = \frac{\text{\# of eliminated variables}}{\text{\# of variables}} \text{(of } t^{th} \text{ iteration)}$$

- Hitting time $h(t)$
  - Expected number of iterations needed for a variable to hit another variable of t^{th} iteration.
  - $h(t) \propto 1/\alpha(t)$.

Elimination Process

- Suppose $\alpha(t) = \alpha$ and $h(t) = a/\alpha$.

$$\underbrace{|E|R(t)}_{\substack{\#\ of\ variables \\ after\ t\ iterations}} - \underbrace{|E|R(t+1)}_{\substack{\#\ of\ variables \\ after\ (t+1)\ iterations}} = \underbrace{\alpha|E|R(t)}_{\substack{\#\ of\ eliminated\ variables \\ of\ (t+1)^{th}\ iteration}}$$



$$\Longrightarrow R(t) = (1-\alpha)^t R(0).$$

Elimination Process

For $0 < \epsilon \ll 1$, the required number of iterations T is given by

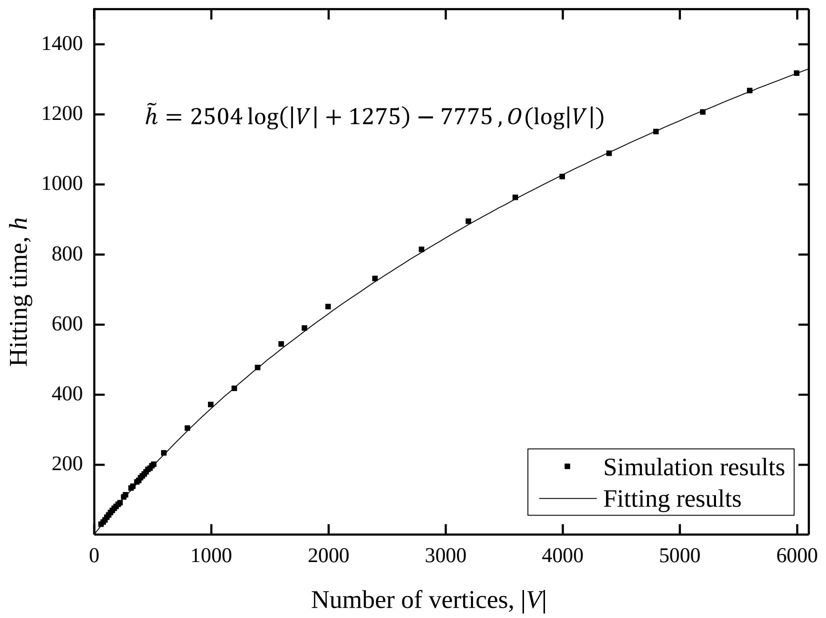
$$(1-\alpha)^T R(0) = \epsilon.$$

For $\alpha \ll 1$,

$$T = \frac{h}{a} \ln \frac{R(0)}{\epsilon}.$$

where h is $O(\log|V|)$.[1]

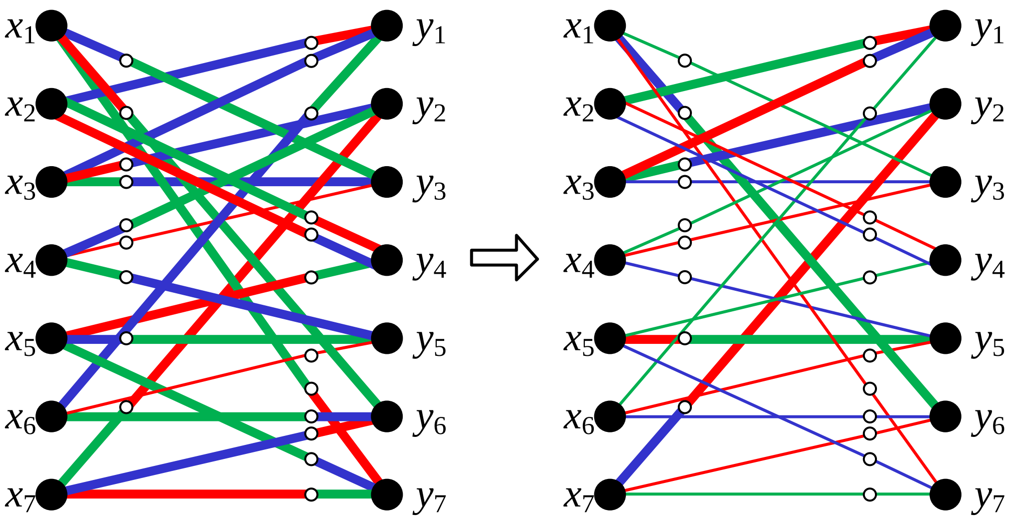
Therefore, T is $O(\log|V|)$.

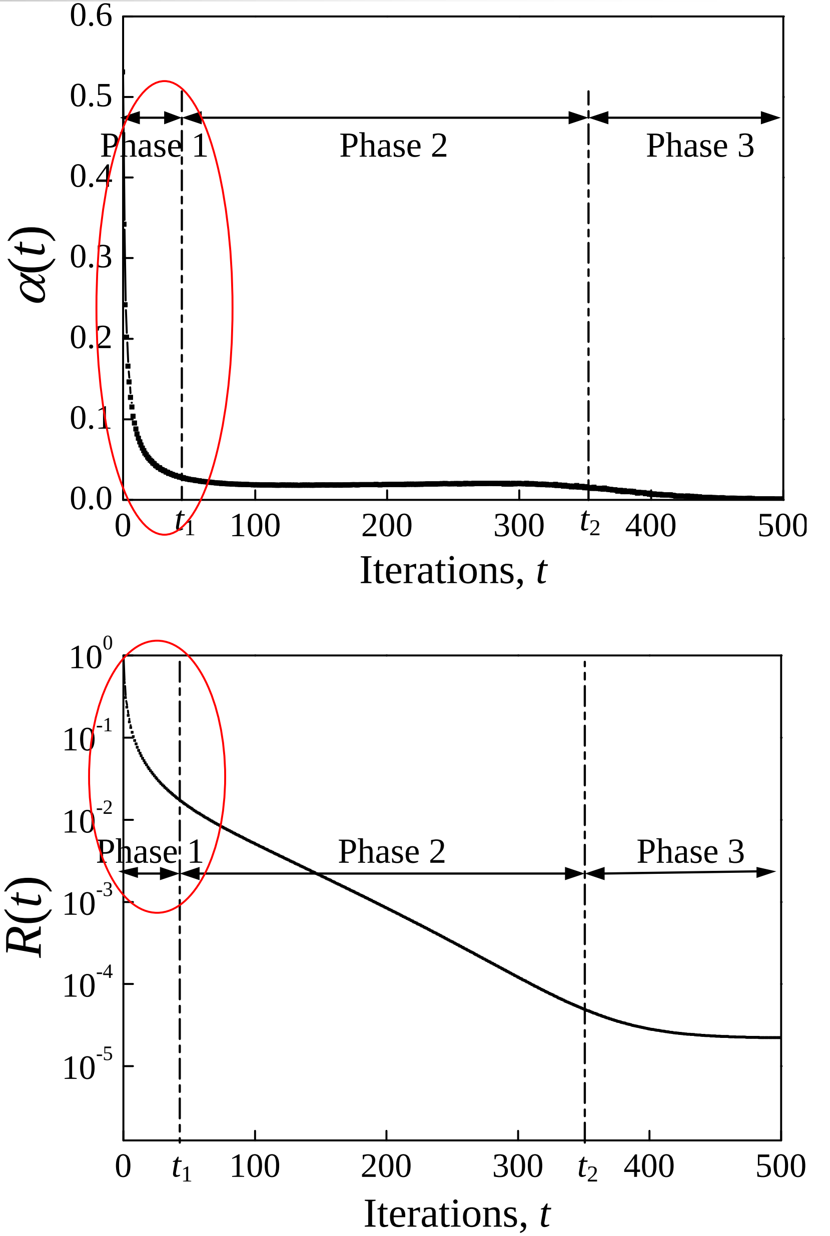


[1] A. Fronczak, P. Fronczak, and J. Hołyst, *Phys. Rev. E*, 70(5), 2004.

Phase 1: Initial

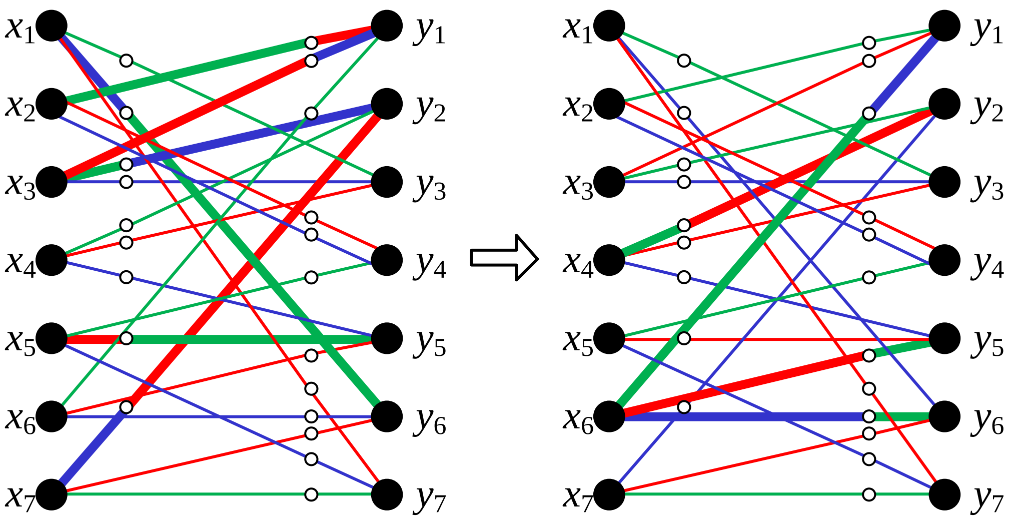
- Variables are more likely to be eliminated when they are close together.

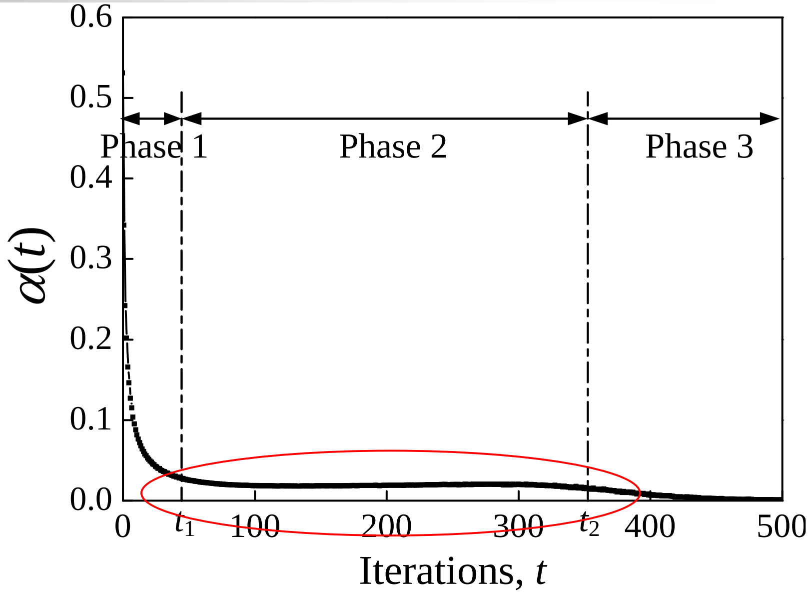


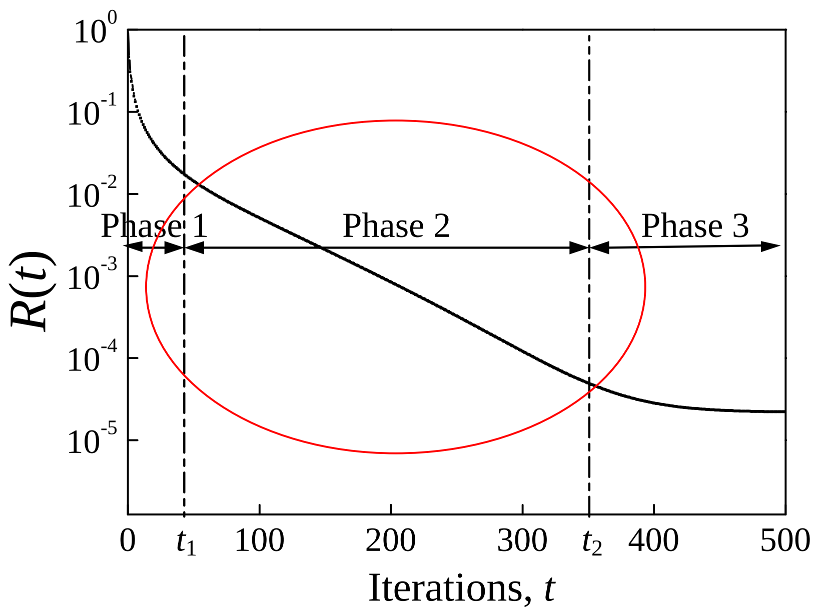


Phase 2: Steady

- As the variable density decreases, the hitting time increases and thus the elimination rate slows down.

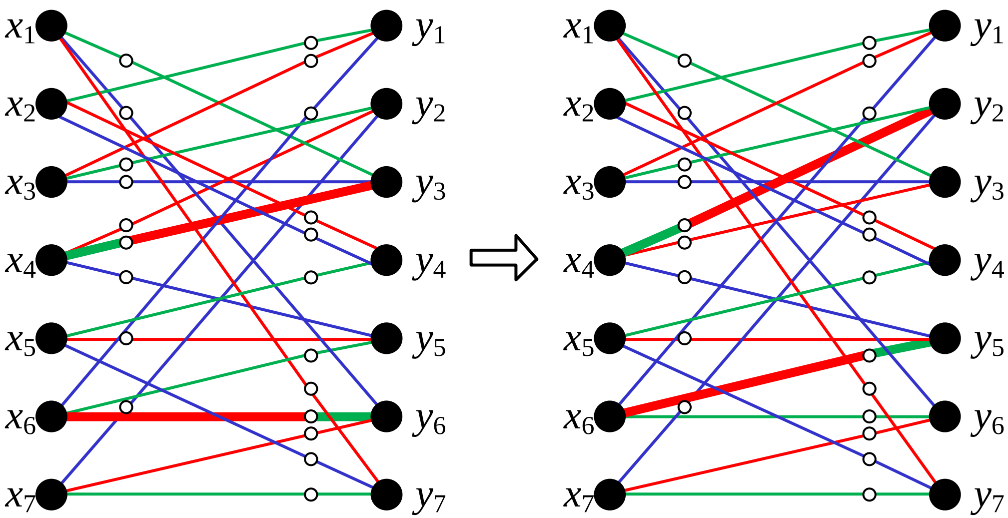


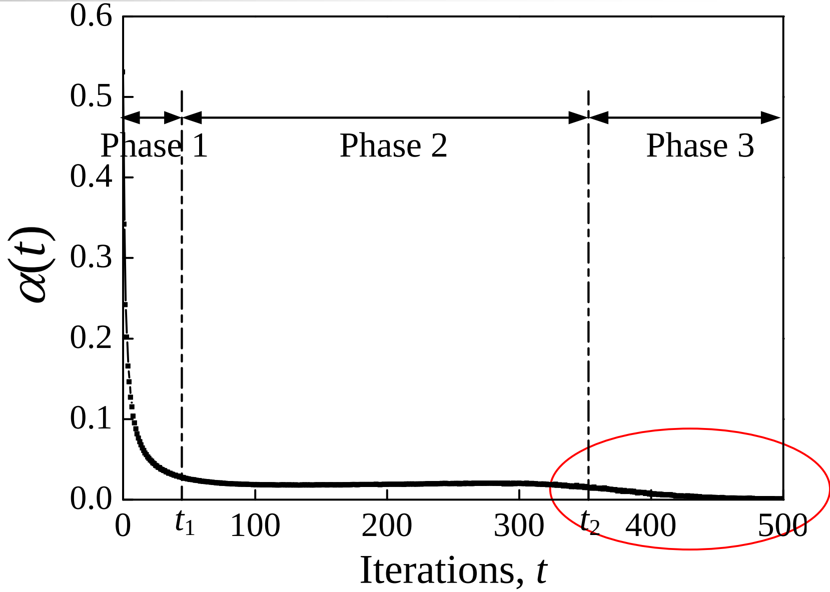


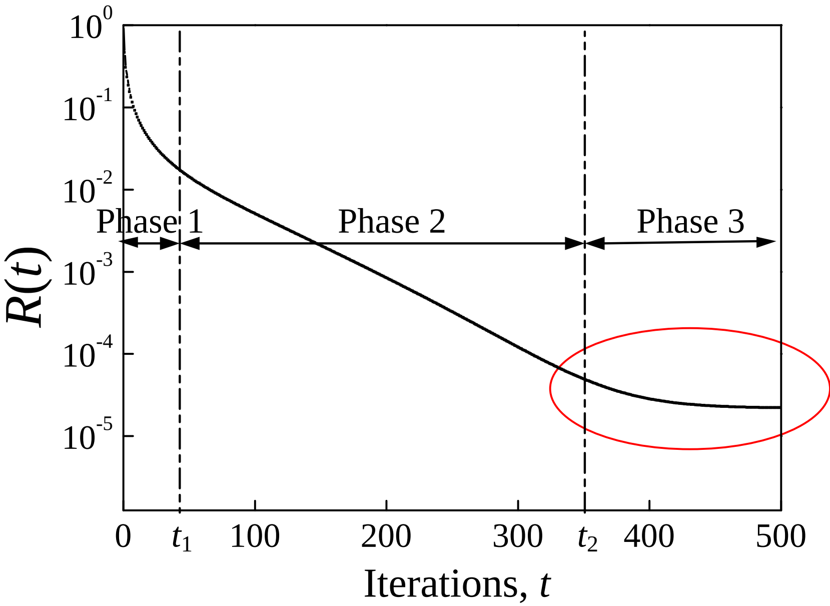


Phase 3: Deadlock

- Remaining variables are likely being blocked in deadlock loops.



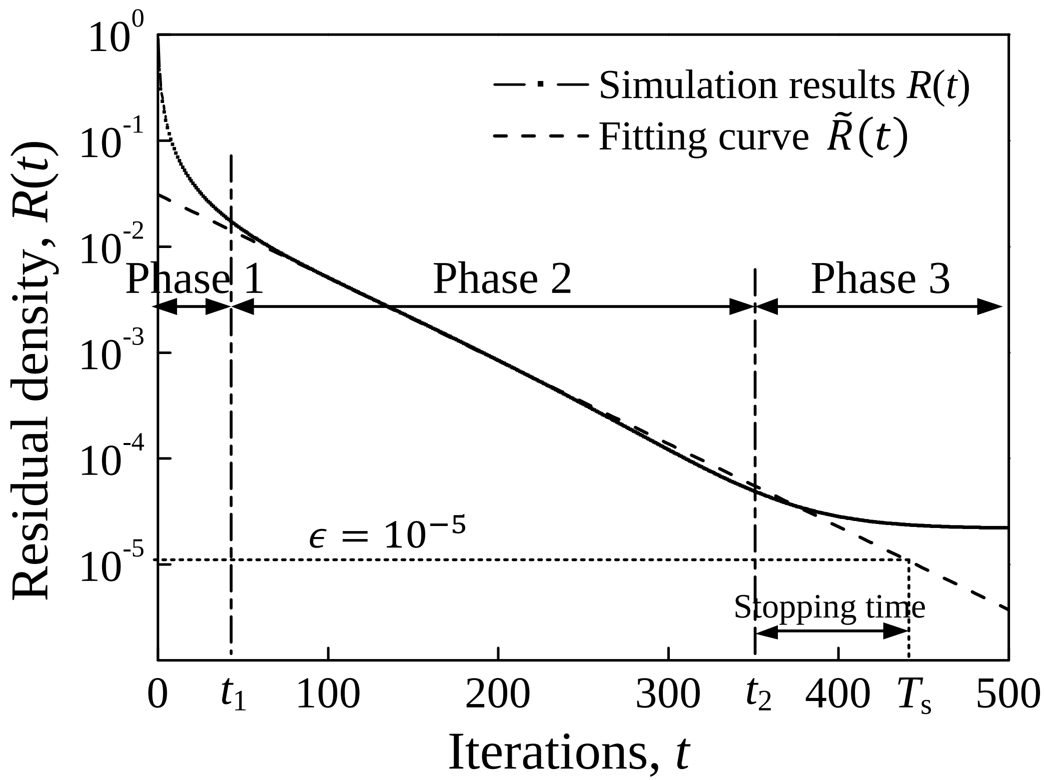




Selection of Stopping Time

- For a given variable density ϵ, the stopping time is

$$T_s \approx \frac{h}{a} \ln \frac{R(t_1)}{\epsilon} + t_1.$$



Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithm
- Performance of Scheduling Algorithms

Parallel Scheduling Algorithm

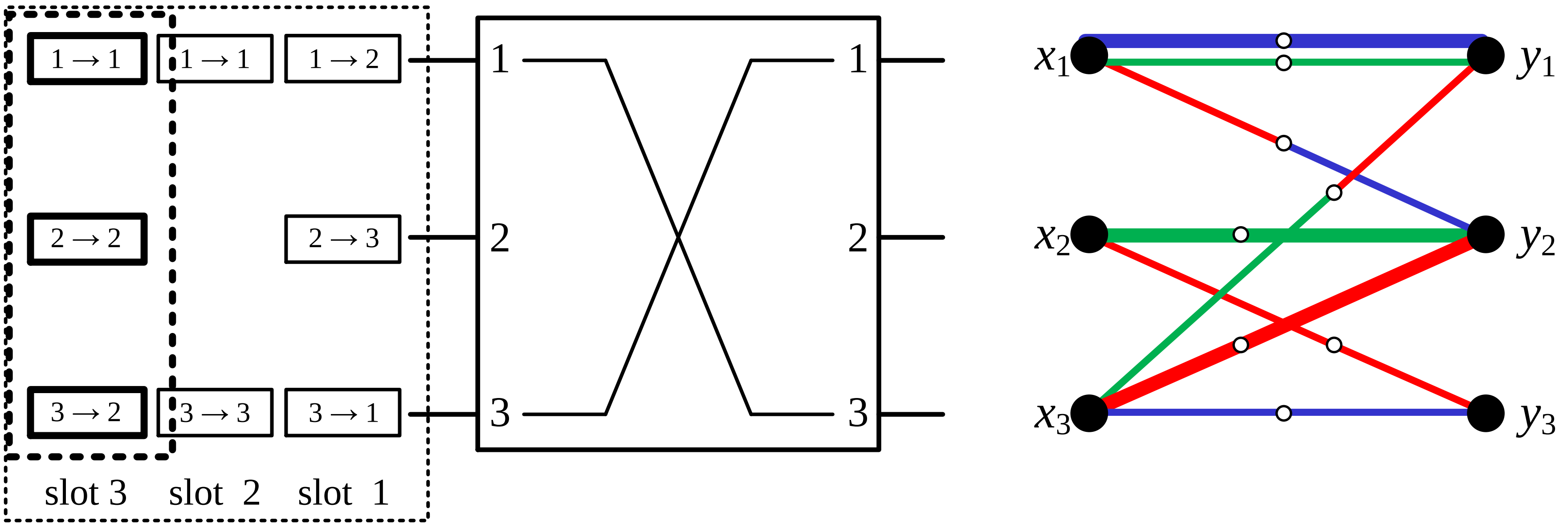
- Graph initialization
  - Arbitrary color assignment

- Perform color exchanges on vertices in X in parallel
- Perform color exchanges on vertices in Y in parallel
  - Repeat until no variable exists or stopping time expires.

- Coloring to Timeslot Assignment

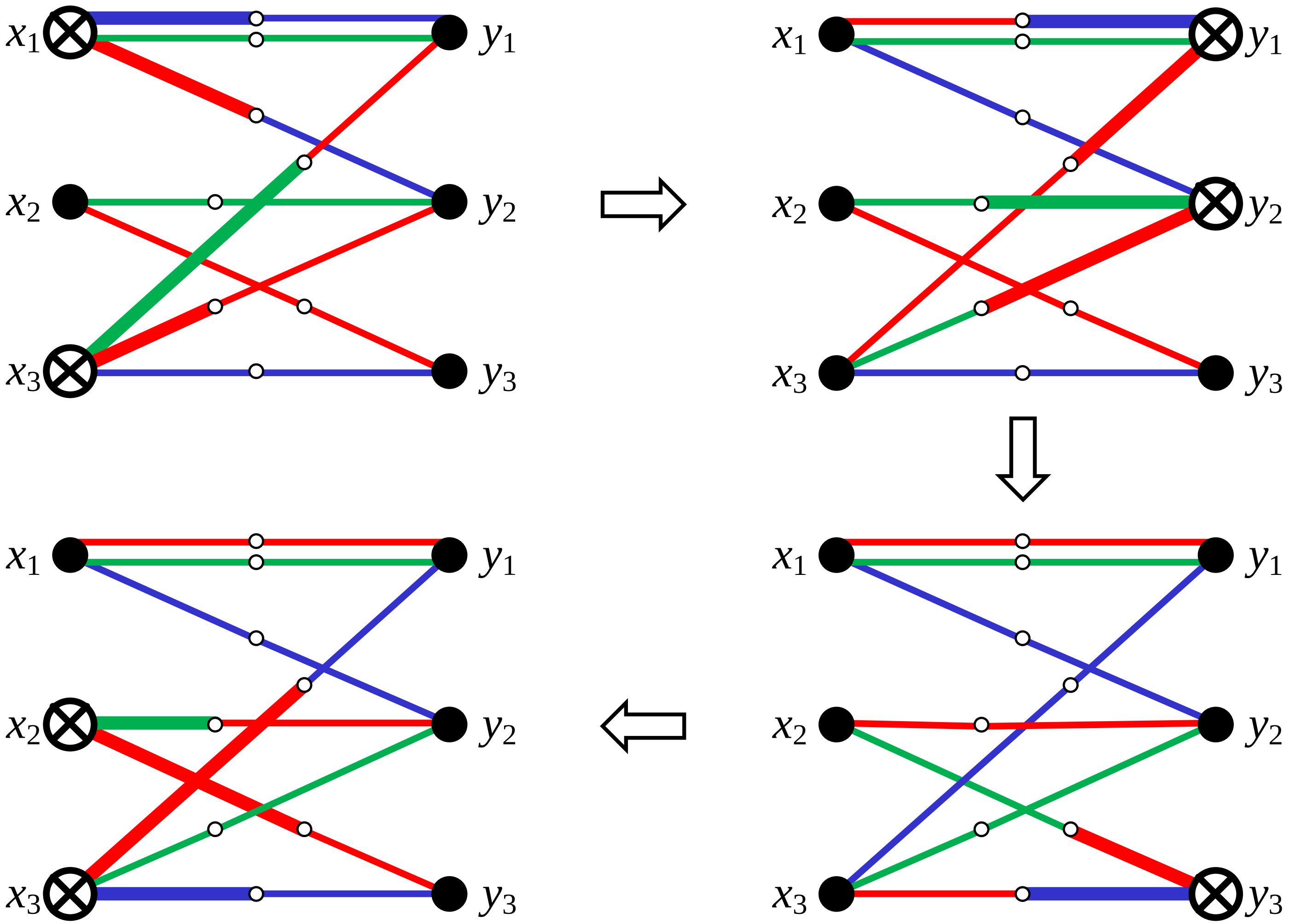
Initialization

- Random color assignment.



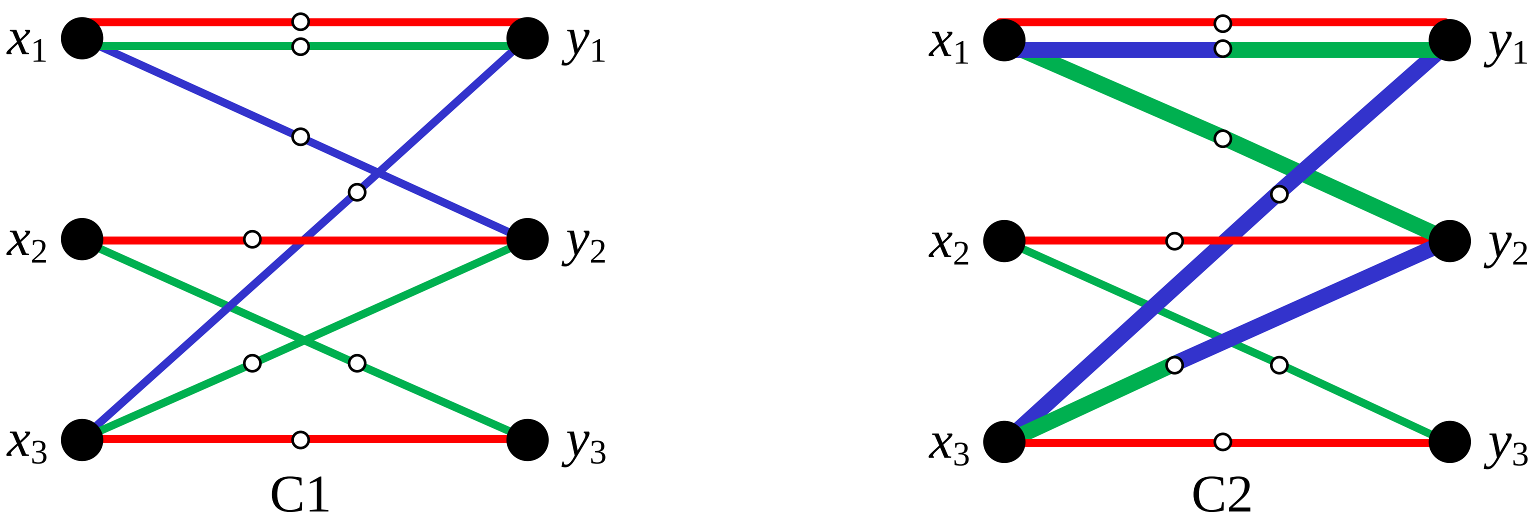
Parallel Complex Coloring

- Perform color exchanges on vertices in X and Y alternatively.



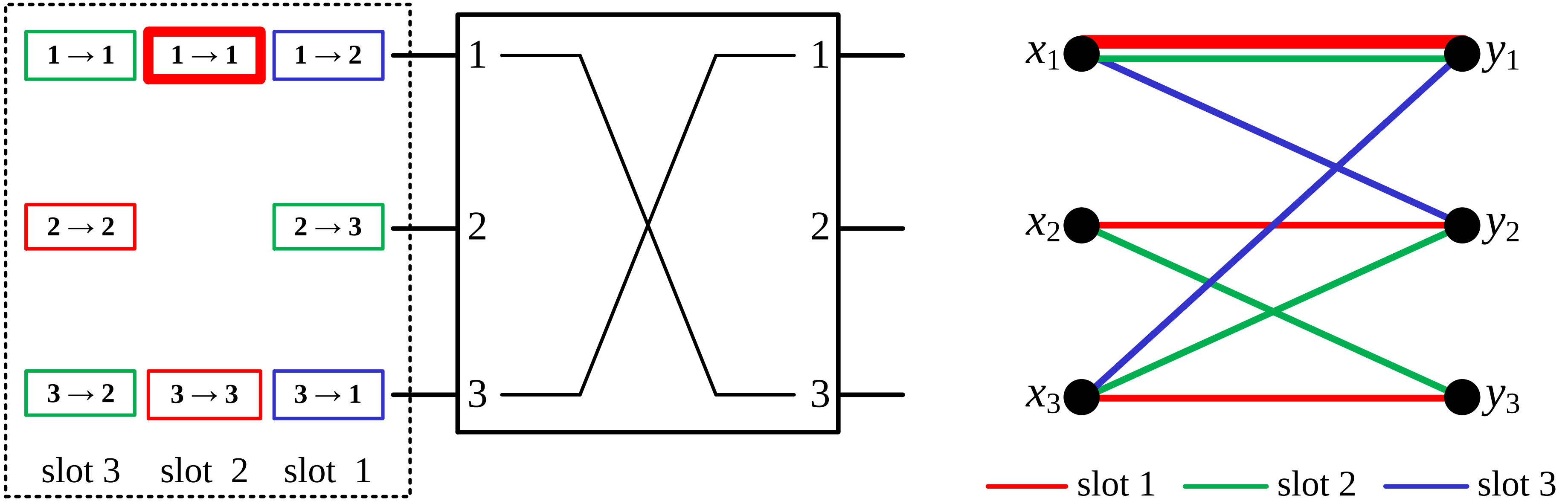
Stopping Condition

- C1: All variables have been eliminated.
- C2: The number of iterations reaches the stopping time T_s.
  - The remaining variables are ignored in the current frame and kept down in the next frame.



Coloring to Timeslot Assignment

- Edges of the same color constitute a matching that represents the scheduled permutation of a corresponding timeslot.



Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithms
- Performance of Scheduling Algorithms
  - Complexity
  - Delay and Throughput

Complexity

- The complexity of the frame-based scheduling algorithm is mainly determined by the processing time of *Parallel Complex Coloring*.
- The running time: $O(\Delta \log N) = O(\log^2 N)$
- The amortized complexity per timeslot: $O(\log N)$.

  (frame size f: $O(\log N)$)

Complexity Comparison

- Comparison of scheduling algorithms for input-queued switches.

Research Work	Complexity per timeslot	Parallel	Scheduling Granularity	Methodology
iSLIP [1]	$O(N\log N)$	Yes	Slot by slot	Maximal size matching
iLQF [2]	$O(N^2\log N)$	Yes	Slot by slot	Maximum weighted matching
LAURA [3]	$O(N\log^2 N)$	Yes	Slot by slot	Maximum weighted matching
Switch-Sched [4]	$O(Nf)$	No	Frame by frame (frame size f: $O(N^2)$)	Greedy edge coloring
Fair-Frame [5]	$O(N^{1.5}\log N)$	No	Frame by frame (frame size f: $O(\log N)$)	Maximum size matching
Our work	$O(\log N)$	Yes	Frame by frame (frame size f: $O(\log N)$)	Complex coloring

[1] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.
[2] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, 1999.
[3] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized scheduling algorithms for input-queued switches," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 4, pp. 642–655, 2003.
[4] G. Aggarwal, R. Motwani, D. Shah, and A. Zhu, "Switch scheduling via randomized edge coloring," in *Proc. IEEE FOCS*, 2003, pp. 502-512.
[5] M. J. Neely, E. Modiano, Y. S. Cheng, "Logarithmic delay for $N \times N$ packet switches under the crossbar constraint," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 657–668, 2007.

Performance Evaluation

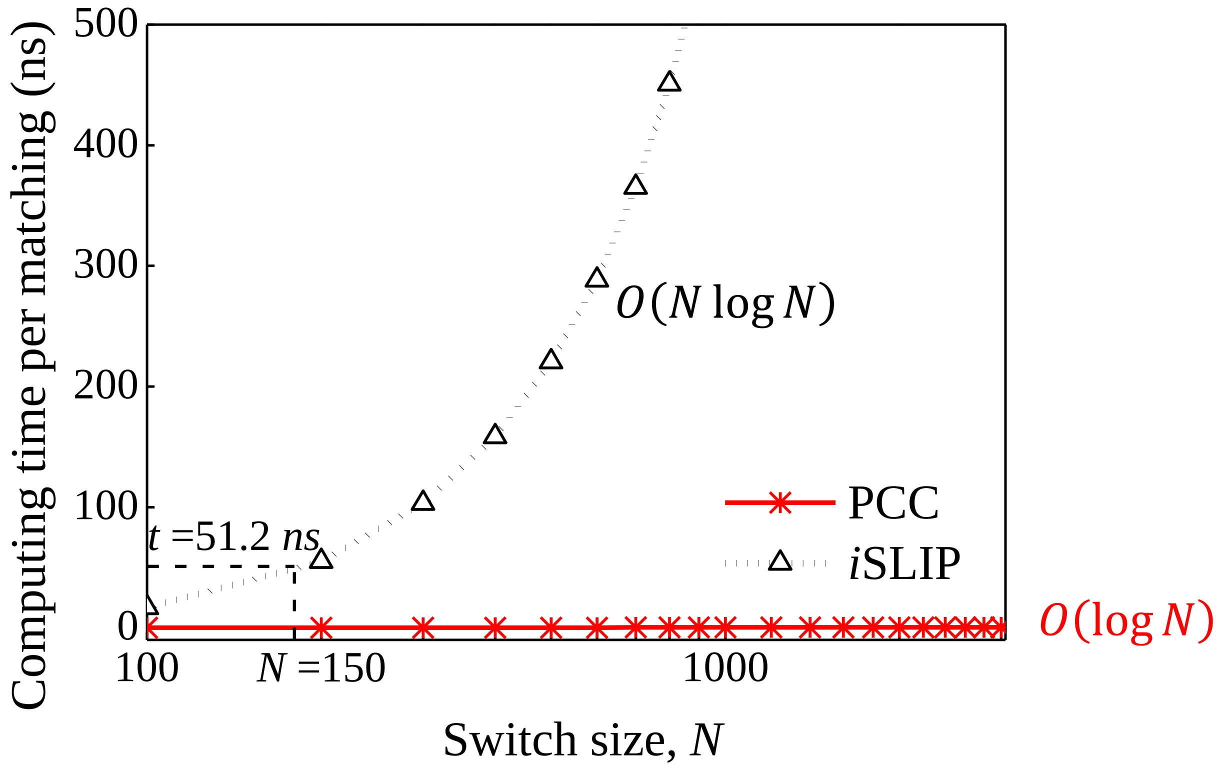
- Assumptions
  - Line rate: 10Gbps
  - Fixed cell size: 64Bytes (equivalent timeslot: 51.2ns)
  - Calculation capability: 20 GFLOPS (Cisco Nexus 5548P)
    (calculation requirement: $\leq$1024 operations per timeslot)

- Performance comparison: *i*SLIP[1]
  - A heuristic arithmetic scheduling algorithm which has been used in commercial switches, such as Cisco Nexus 5548P.
  - $O(N \log N)$ time complexity.

[1] N. McKeown, IEEE/ACM Trans. Netw., vol. 7, no. 2, pp. 188–201, Apr. 1999.

Complexity vs. N

- The amortized complexity per timeslot per matching: $O(\log N)$



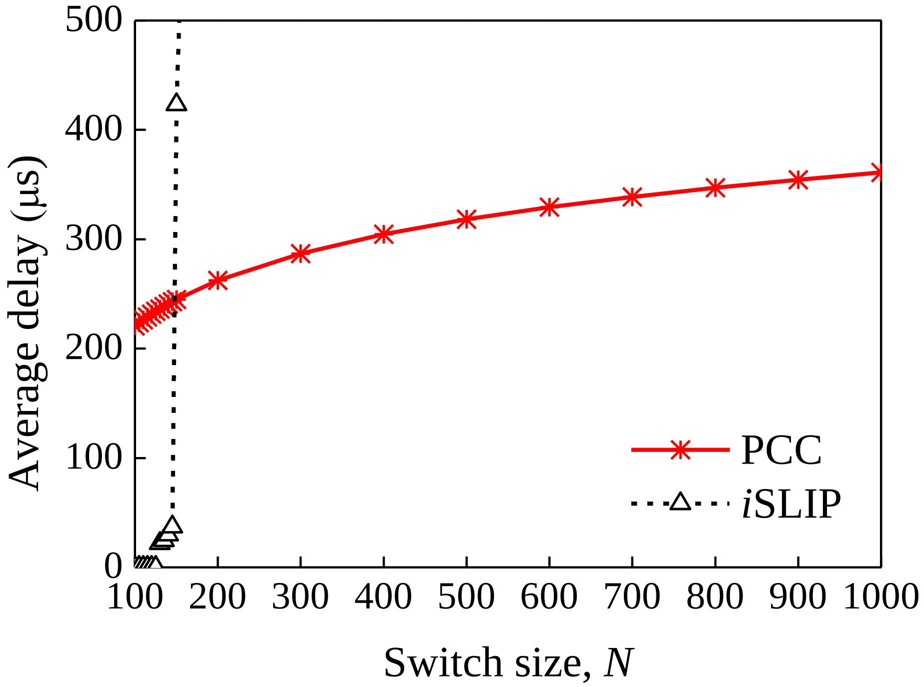
Simulation result when the switch is fully loaded

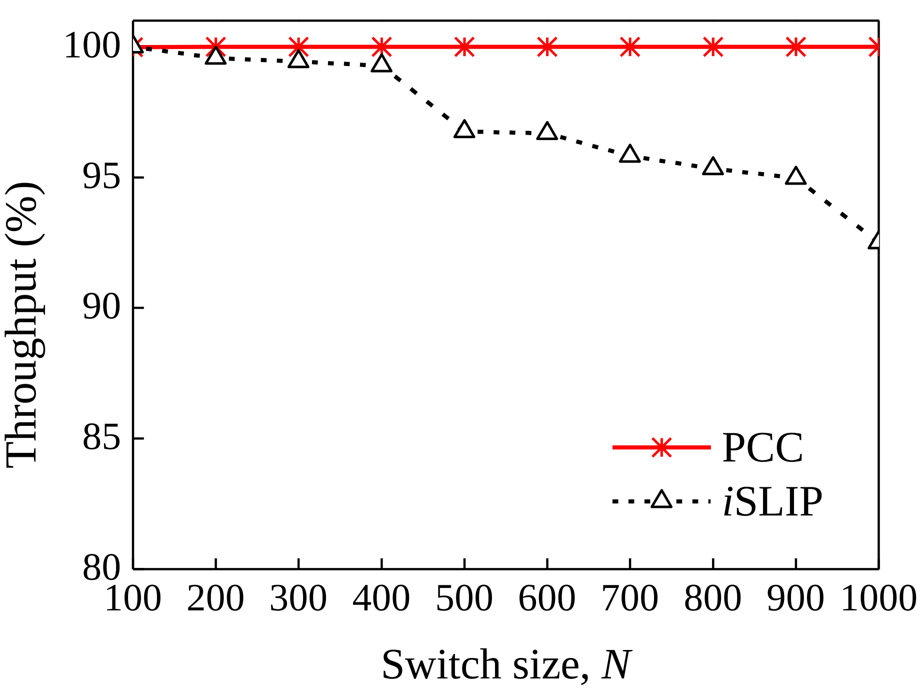
Outline

- Introduction and Overview
- Preliminaries of Scheduling and Complex Coloring
- Parallel Complex Coloring
- Parallel Scheduling Algorithms
- Performance of Scheduling Algorithms
  - Complexity
  - Delay and Throughput

Delay and Throughput vs. *N*

- End-to-end delay
  - If a algorithm fails to compute a matching within a timeslot, it may take two or more timeslots which is considered in delay calculations.
- Input rate $\lambda = 0.7$





Delay and Throughput when $N = 300$

- The performance under non-uniform traffic is as good as that under uniform traffic