Analysis and Approximation of Optimal Co-Scheduling on Chip Multiprocessors

Yunlian Jiang

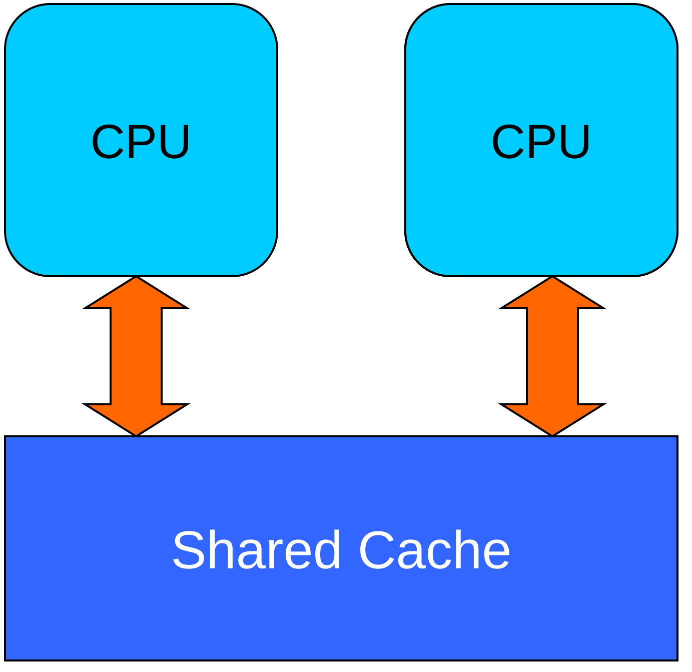
Xipeng Shen   The College of William & Mary, USA

Jie Chen   DoE Jefferson Lab , USA

Rahul Tripath   University of South Florida, USA

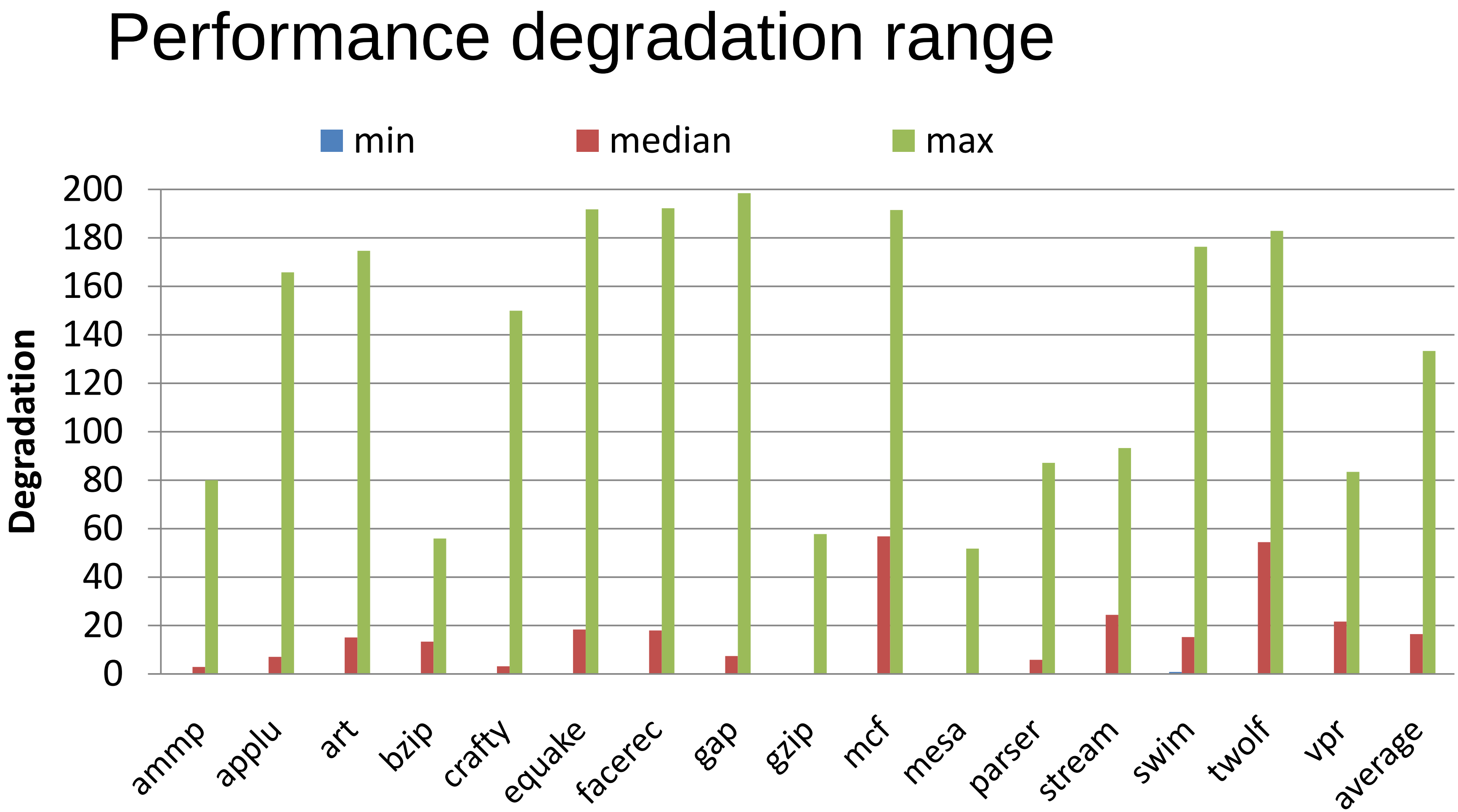
Cache Sharing on CMP

- Shorten inter-thread communication
- Flexible usage of cache



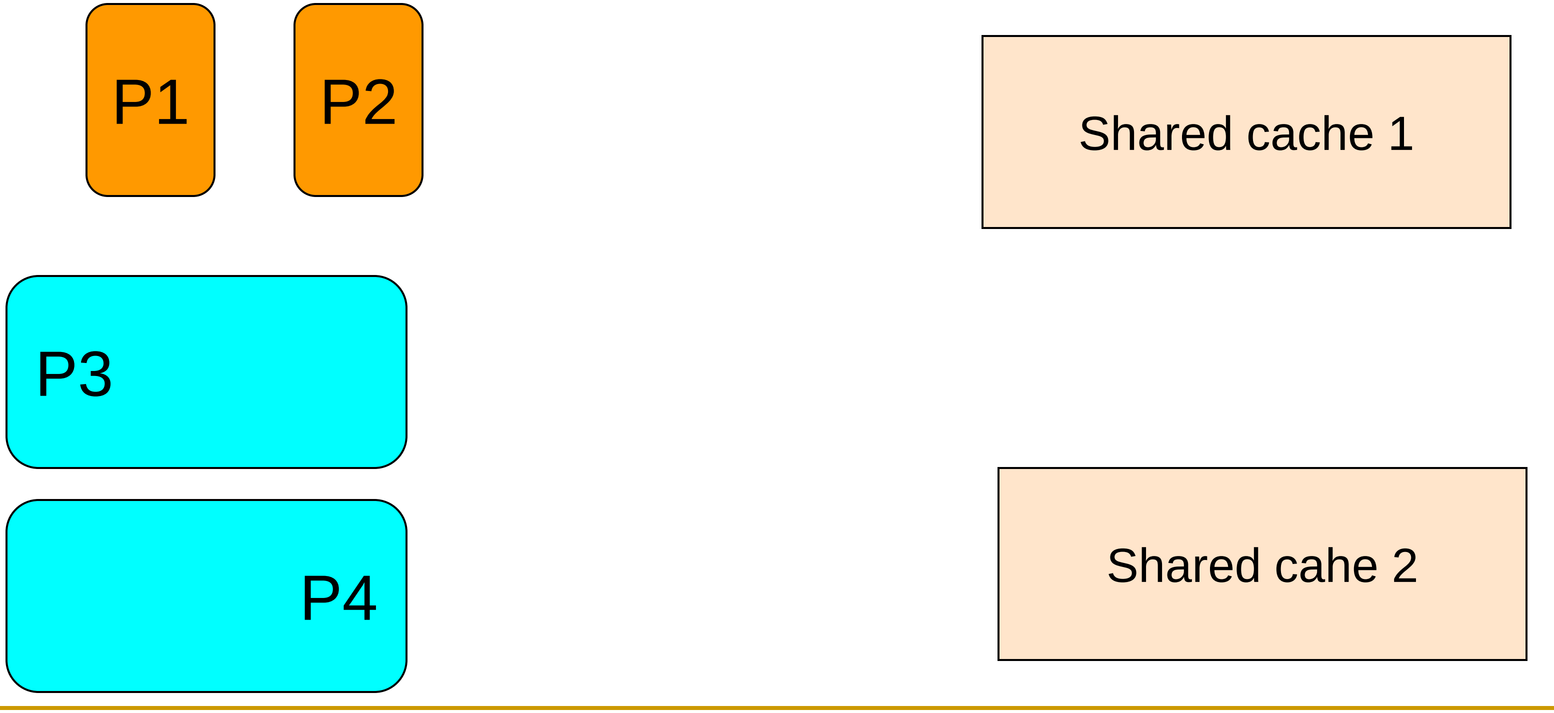
- degrade performance
- impair fairness
- hurt performance isolation

Degradation is affected by co-runner



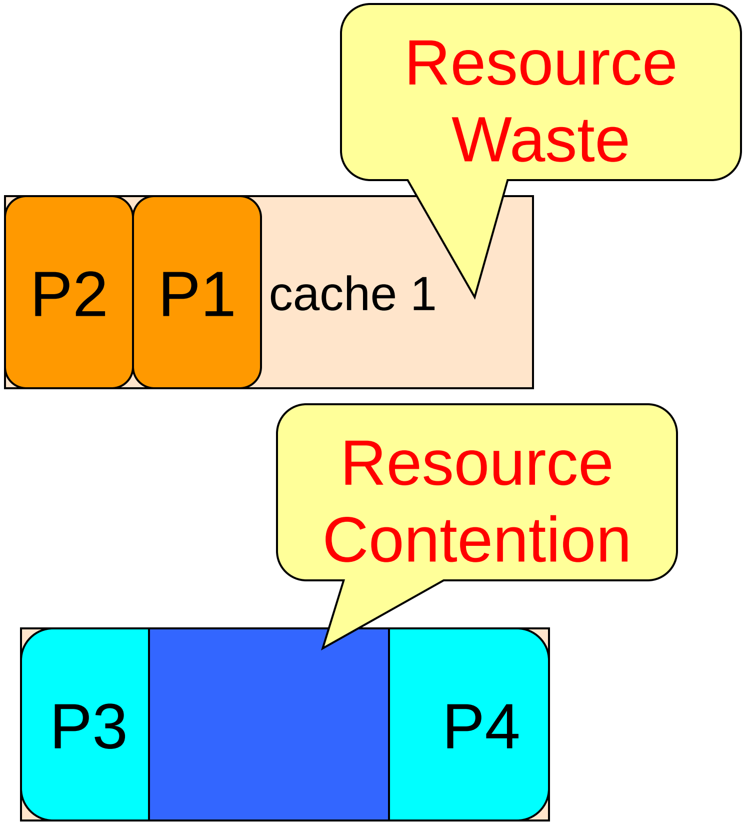
Job Co-Scheduling

- To assign jobs to chips in a manner to minimize contention



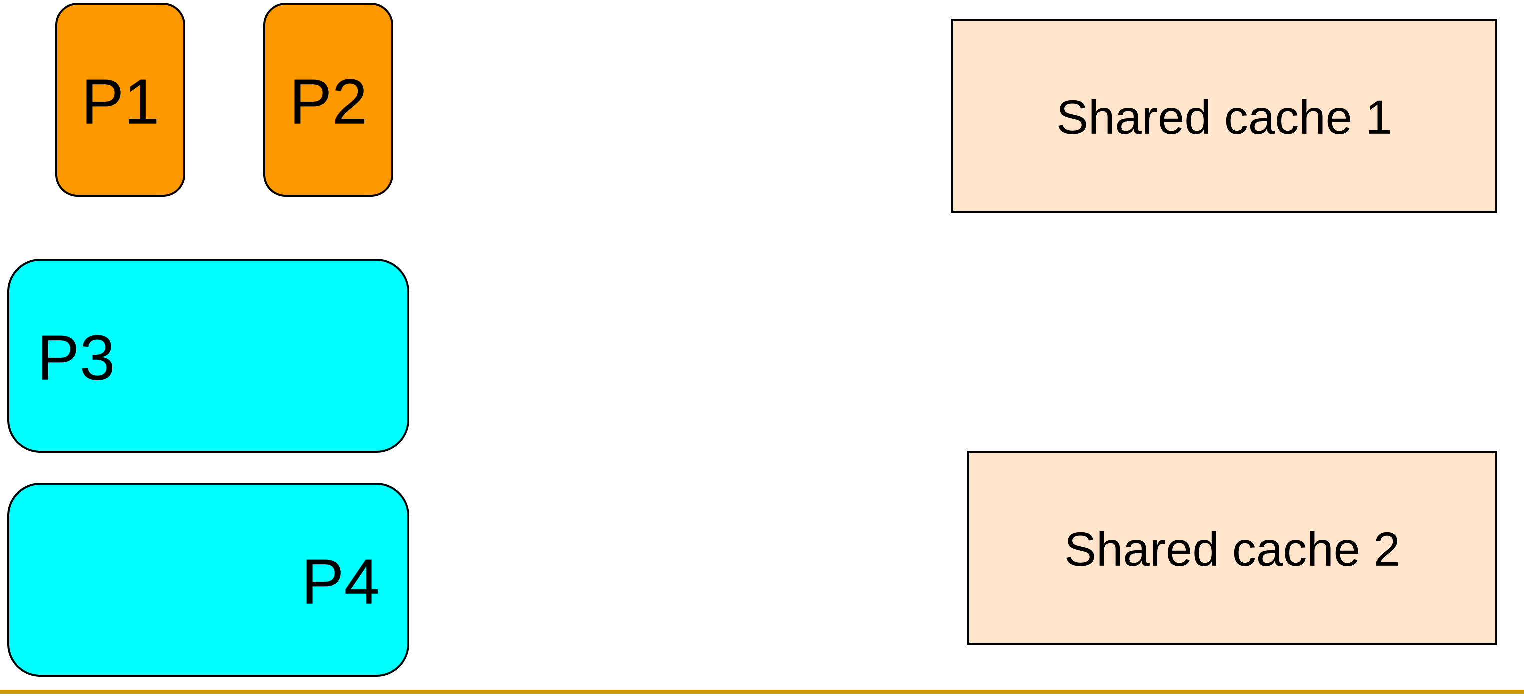
Job Co-Scheduling

- To assign jobs to chips in a manner to minimize contention



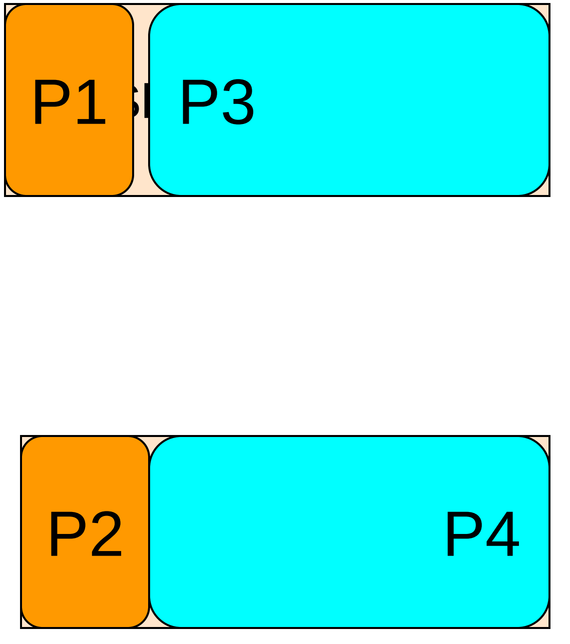
Job Co-Scheduling

- To assign jobs to chips in a manner to minimize contention



Job Co-Scheduling

- To assign jobs to chips in a manner to minimize contention



The Goal of this Work

- Related work
  - Snavely etc. [00' ASPLOS]
- Goal of this work
  - Find the optimal schedule on CMP system
- Benefits
  - Evaluate current schedule quality
  - Applied in real system

Contributions

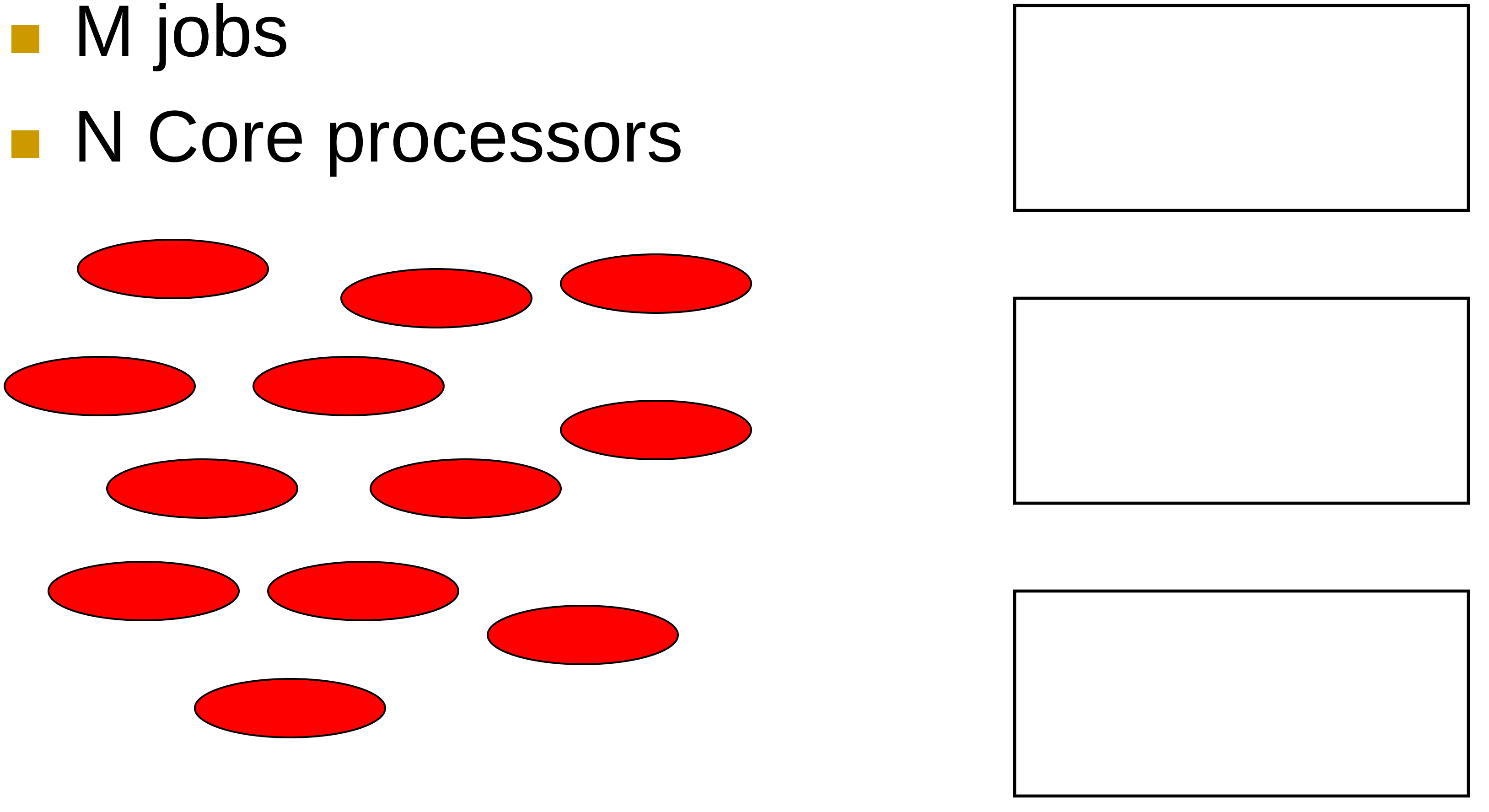
- Polynomial optimal solution on Dual-core systems
- NP-Completeness proof on K-core (K>2) systems
- Polynomial approximation algorithms on K-core (K>2) systems

Contributions

- Polynomial optimal solution on Dual-core systems
- NP-Completeness proof on K-core (K>2) systems
- Polynomial approximation algorithms on K-core (K>2) systems

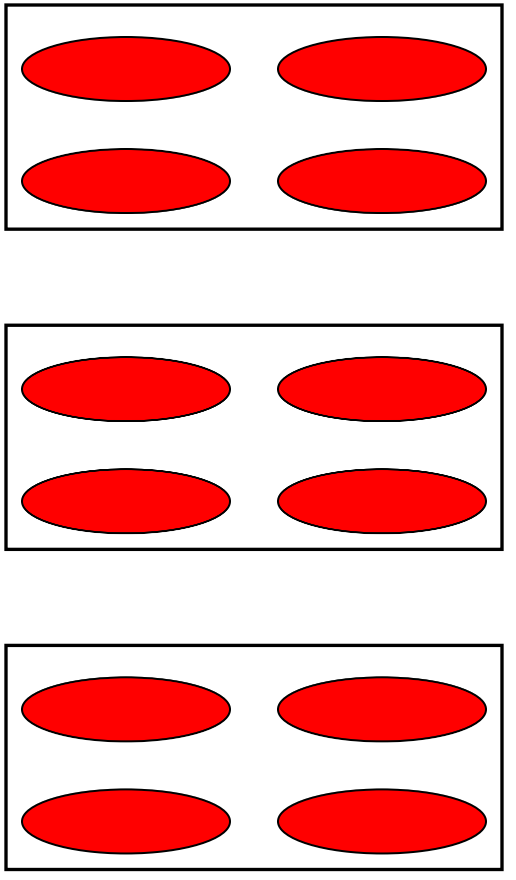
Problem Formulation

- M jobs
- N Core processors



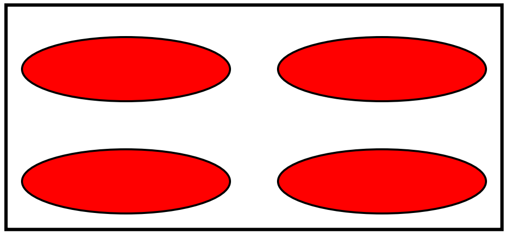
Problem Formulation

- M jobs
- N Core processors



Problem Formulation

- Assignment



$$Deg_i = \frac{cCPI_i - sCPI_i}{sCPI_i}$$

- Goal Minimize $\sum Deg_i$

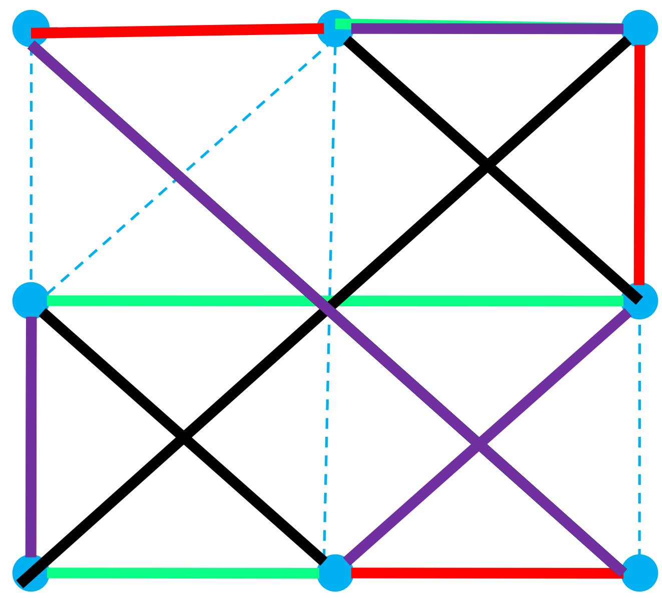
Dual-Core System

- Polynomial Solution
  - Minimum-weight perfect matching

    [Edmonds: 1965]
- Matching
  - A matching M in graph G is a set of edges with no common vertex.
- perfect matching is a matching which matches all vertices of the graph

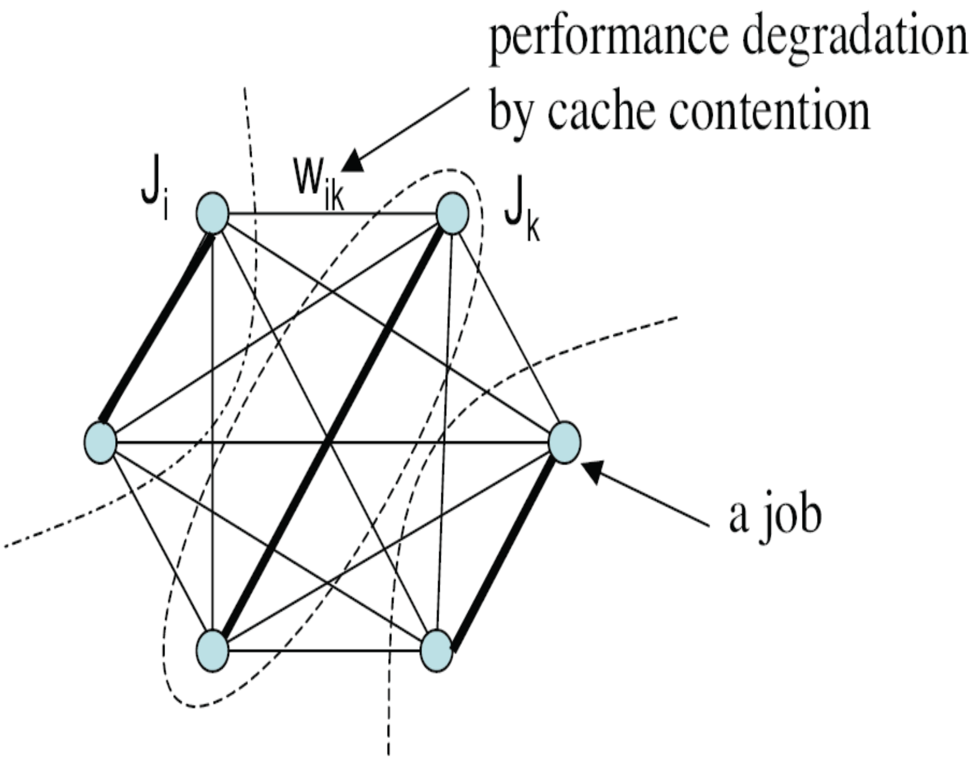
Dual-Core System

- Minimum-weight perfect matching
  - In edge weighted graph
  - Sum of weight of edges in the match is minimum



Dual-Core System

- Job
  - Nodes
- Corun-Degradation
  - Edge Weight
- Optimal Schedule
  - Minimum weight perfect matching



Contributions

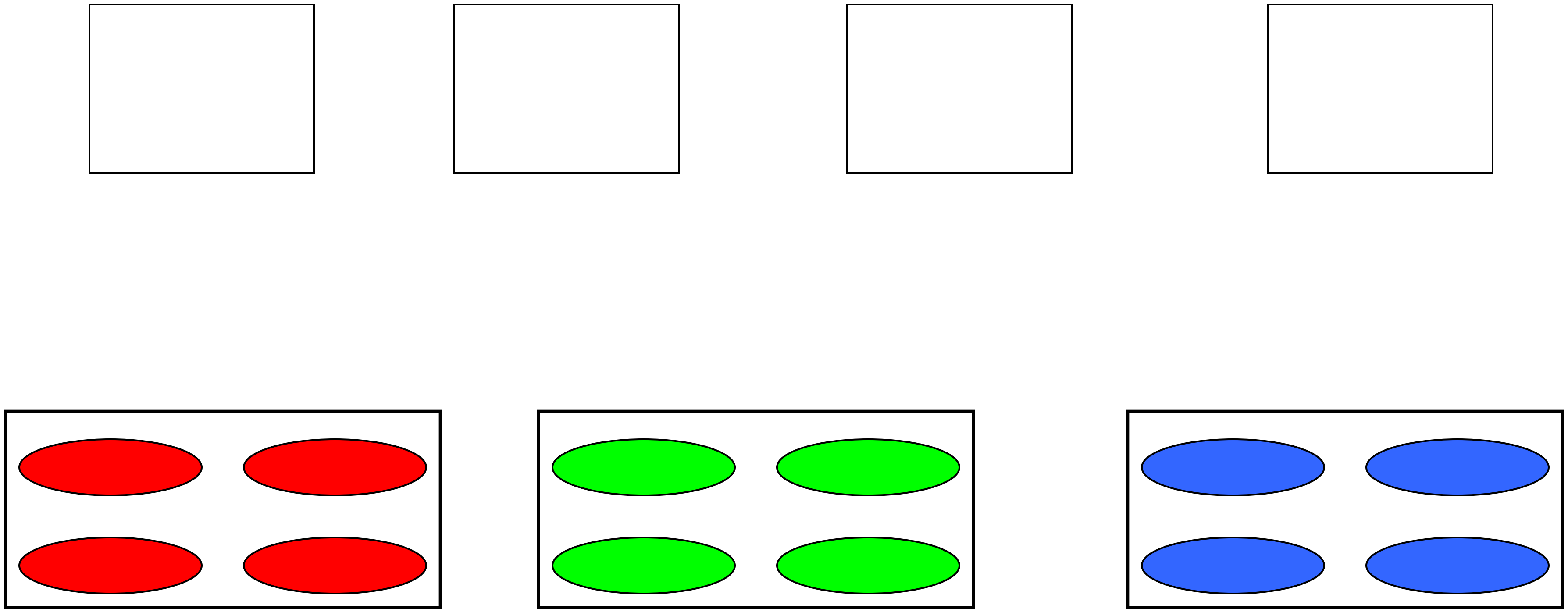
- Polynomial optimal solution on Dual-core systems
- NP-Completeness proof on K-core (K>2) systems
- Polynomial approximation algorithms on K-core (K>2) systems

NP-Completeness proof

- NP proof
  - Given a schedule, can compute
- Reduction
  - NP-Complete problem → Job Co-scheduling
    - Multidimensional Assignment Problem (MAP)

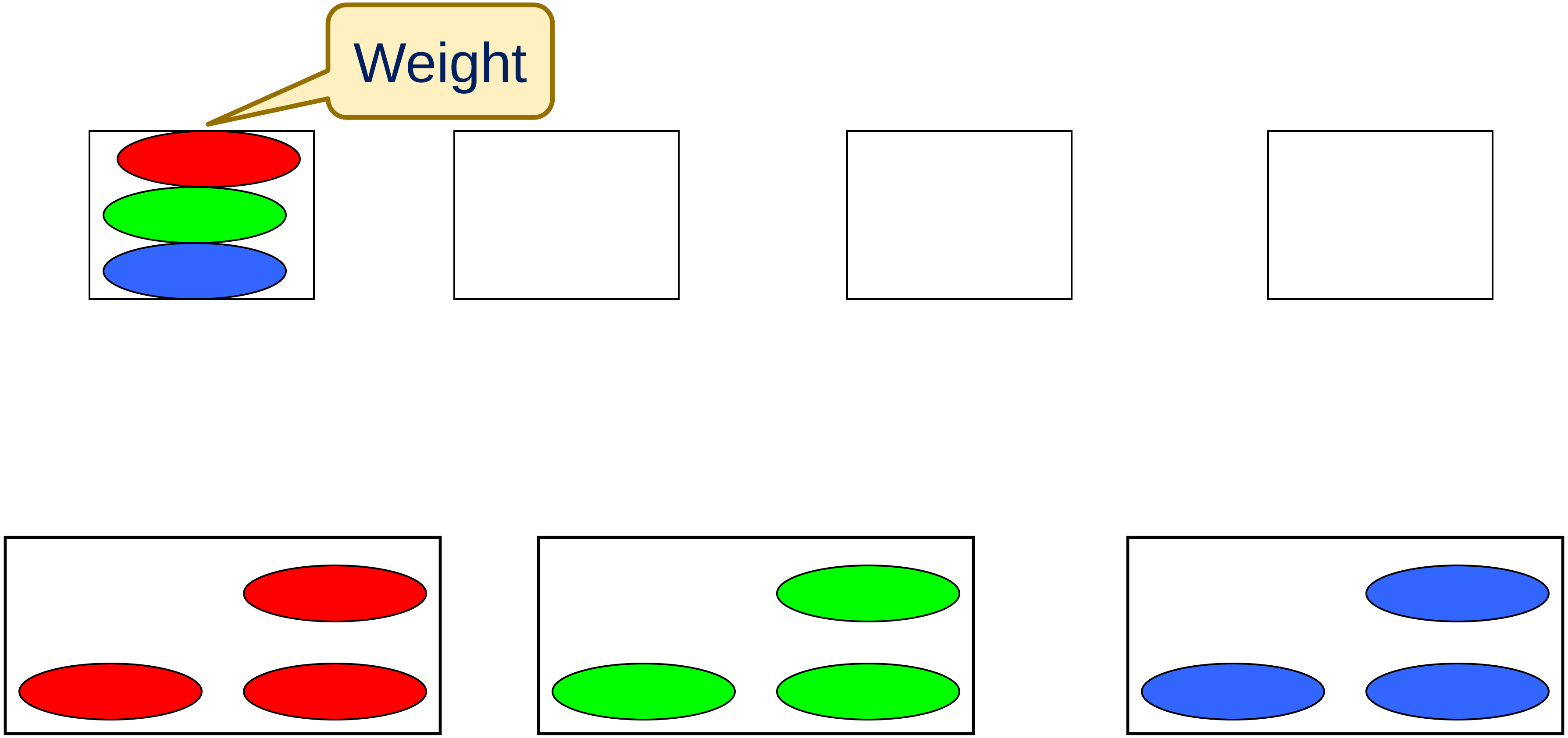
NP-Completeness Proof

- MAP



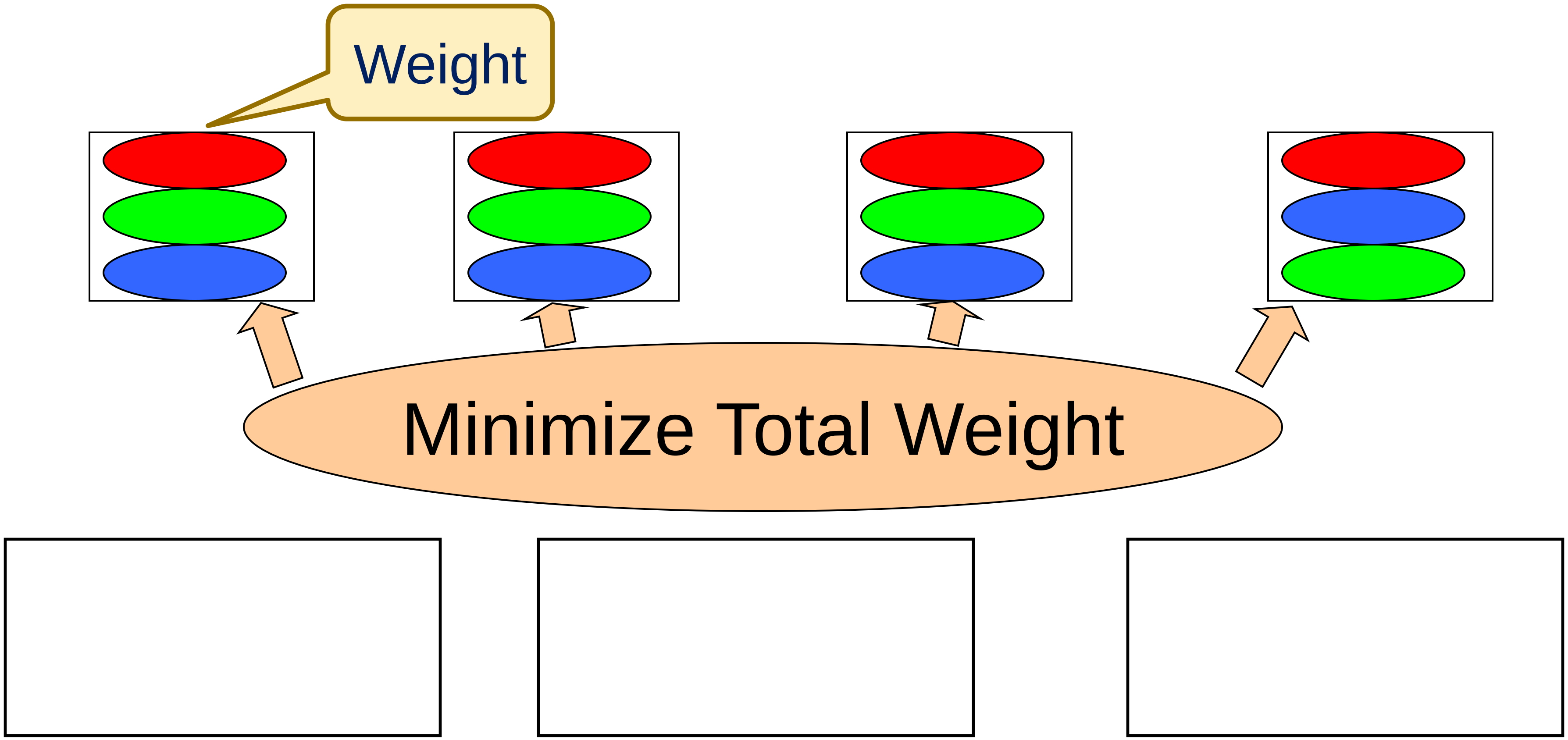
NP-Completeness Proof

- MAP



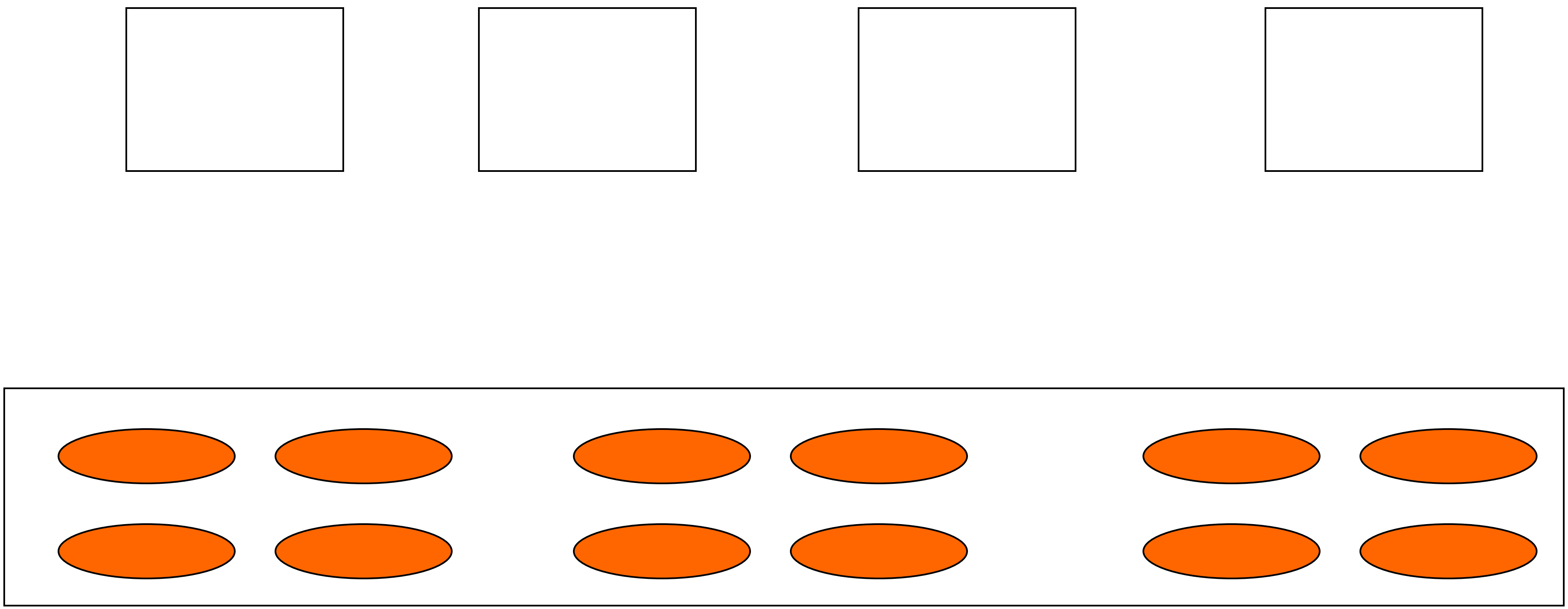
NP-Completeness Proof

- MAP



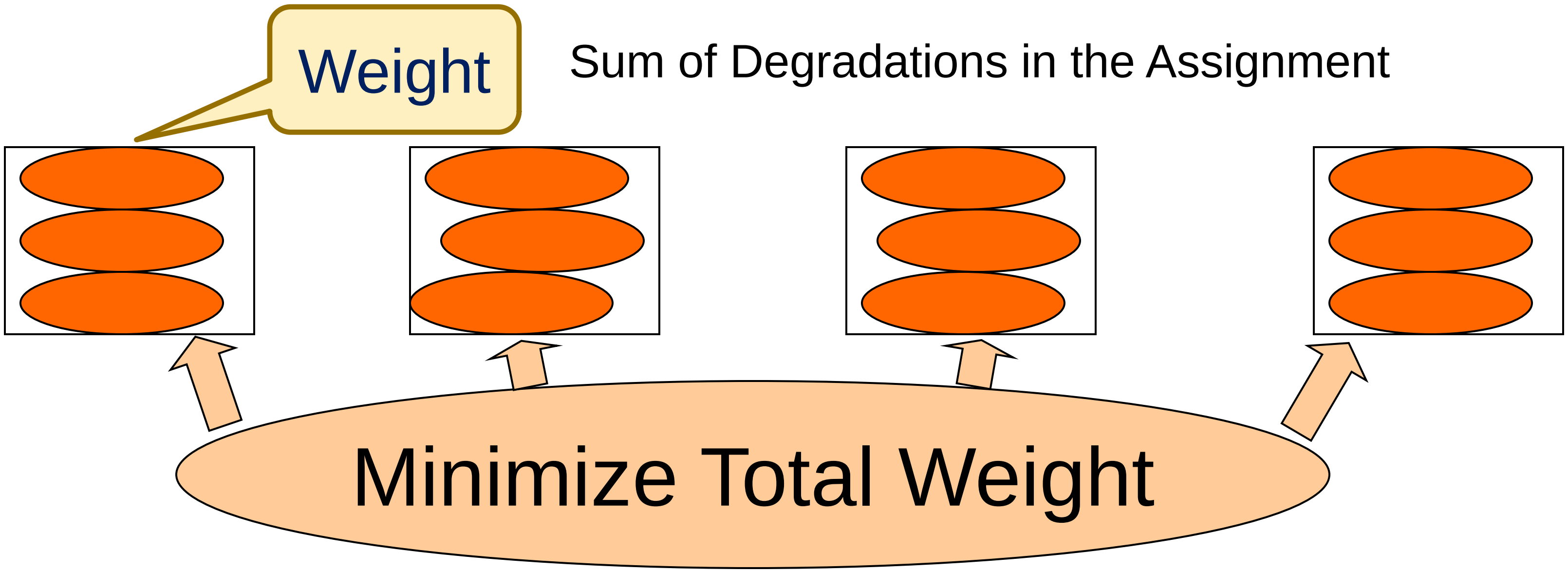
NP-Completeness Proof

- Job Co-Scheduling on CMP



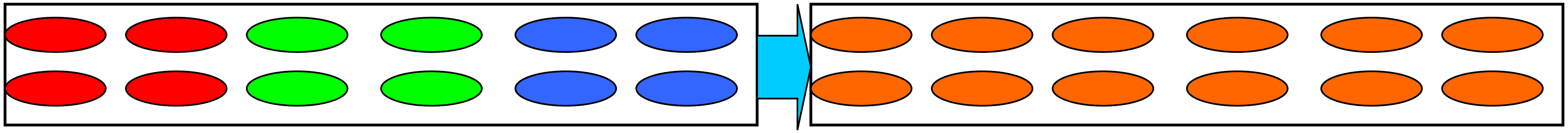
NP-Completeness Proof

- Job Co-Scheduling on CMP



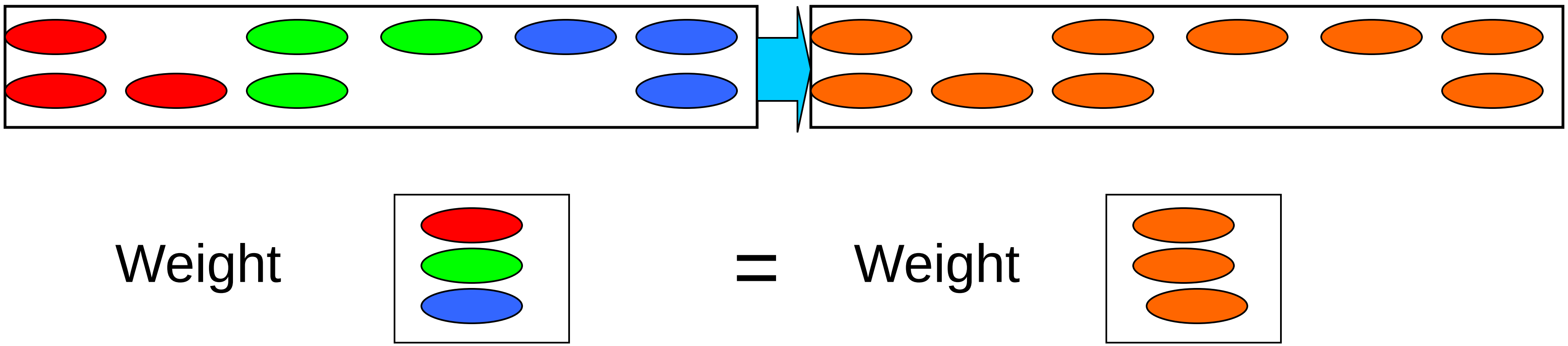
NP-Completeness Proof

- MAP → Job Co-Scheduling



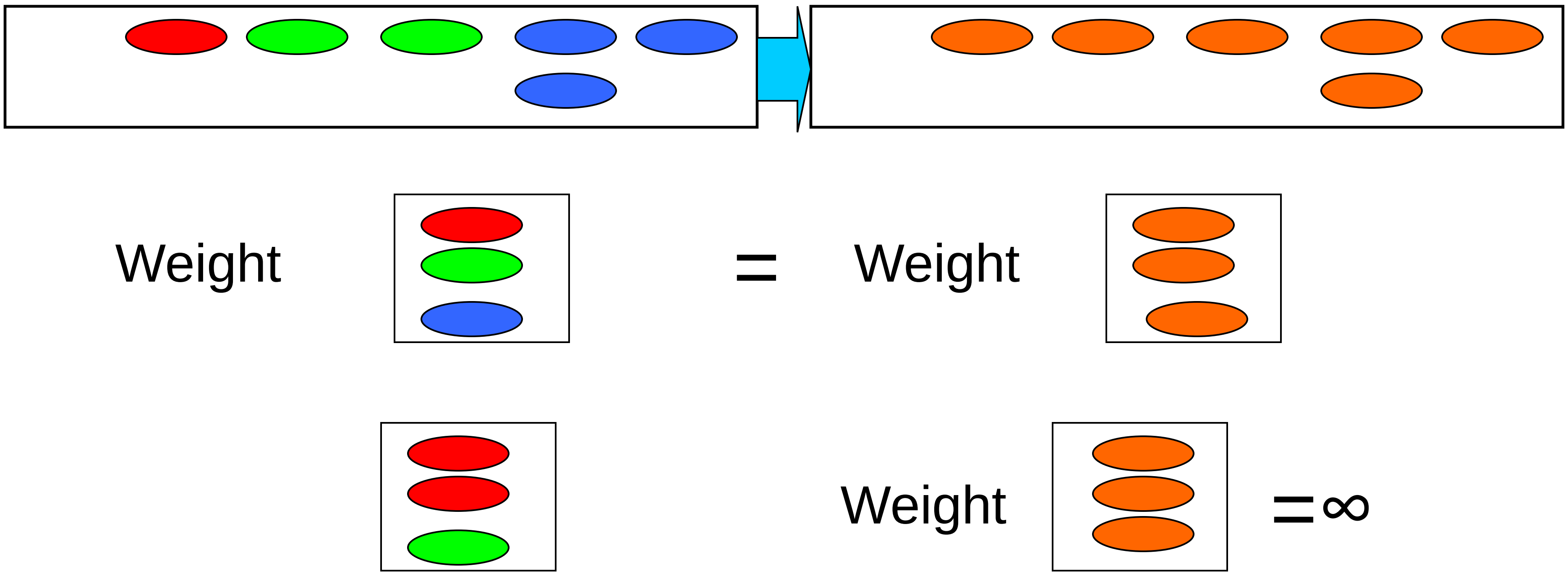
NP-Completeness Proof

- MAP → Job Co-Scheduling



NP-Completeness Proof

- MAP → Job Co-Scheduling



Contributions

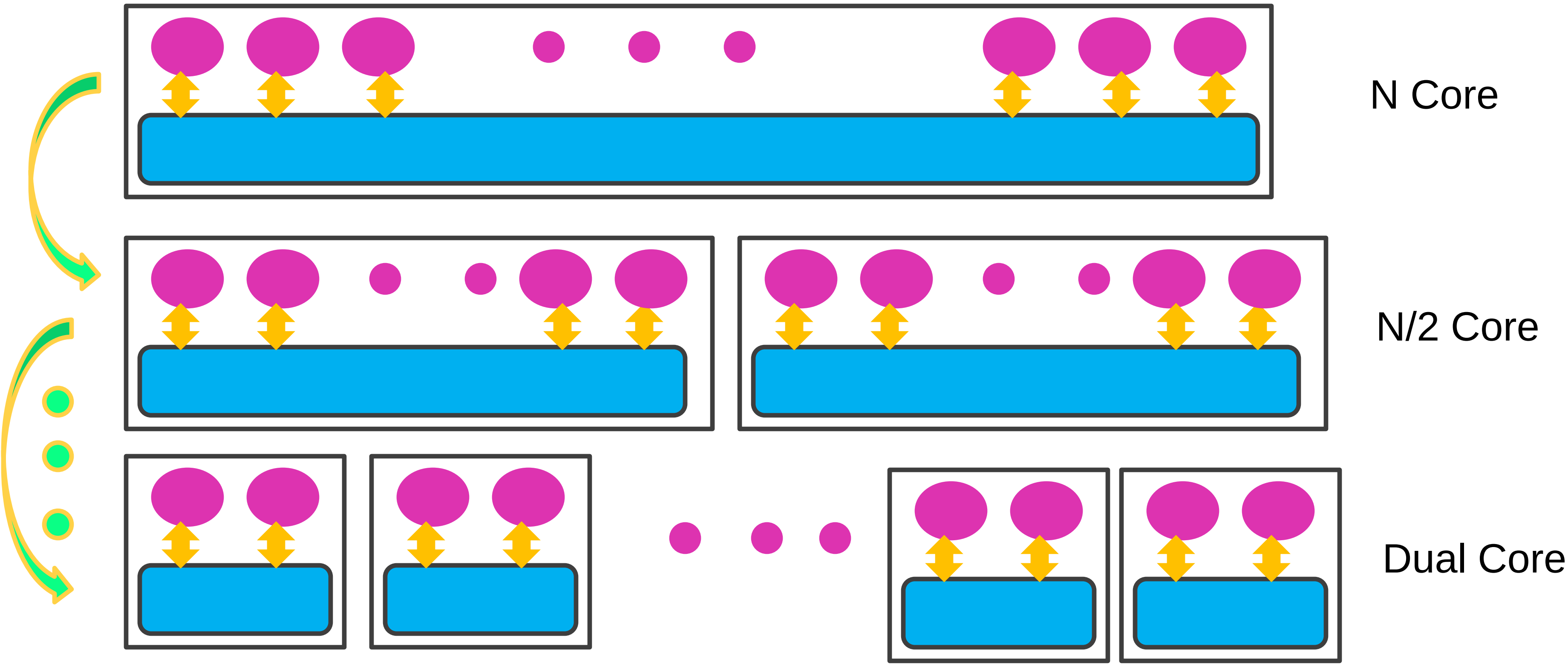
- Polynomial optimal solution on Dual-core systems
- NP-Completeness proof on K-core (K>2) systems
- Polynomial approximation algorithms on K-core (K>2) systems

Approximation algorithms

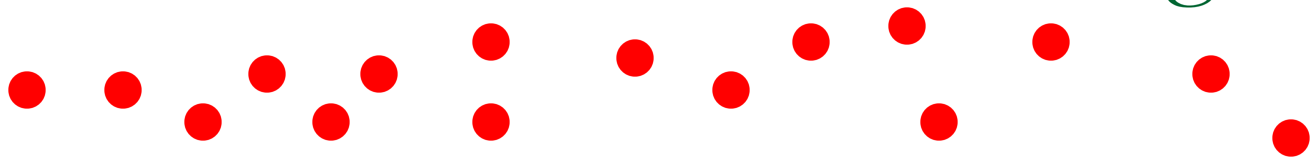
- Hierarchical Perfect Matching
- Greedy

Hierarchical Perfect Matching

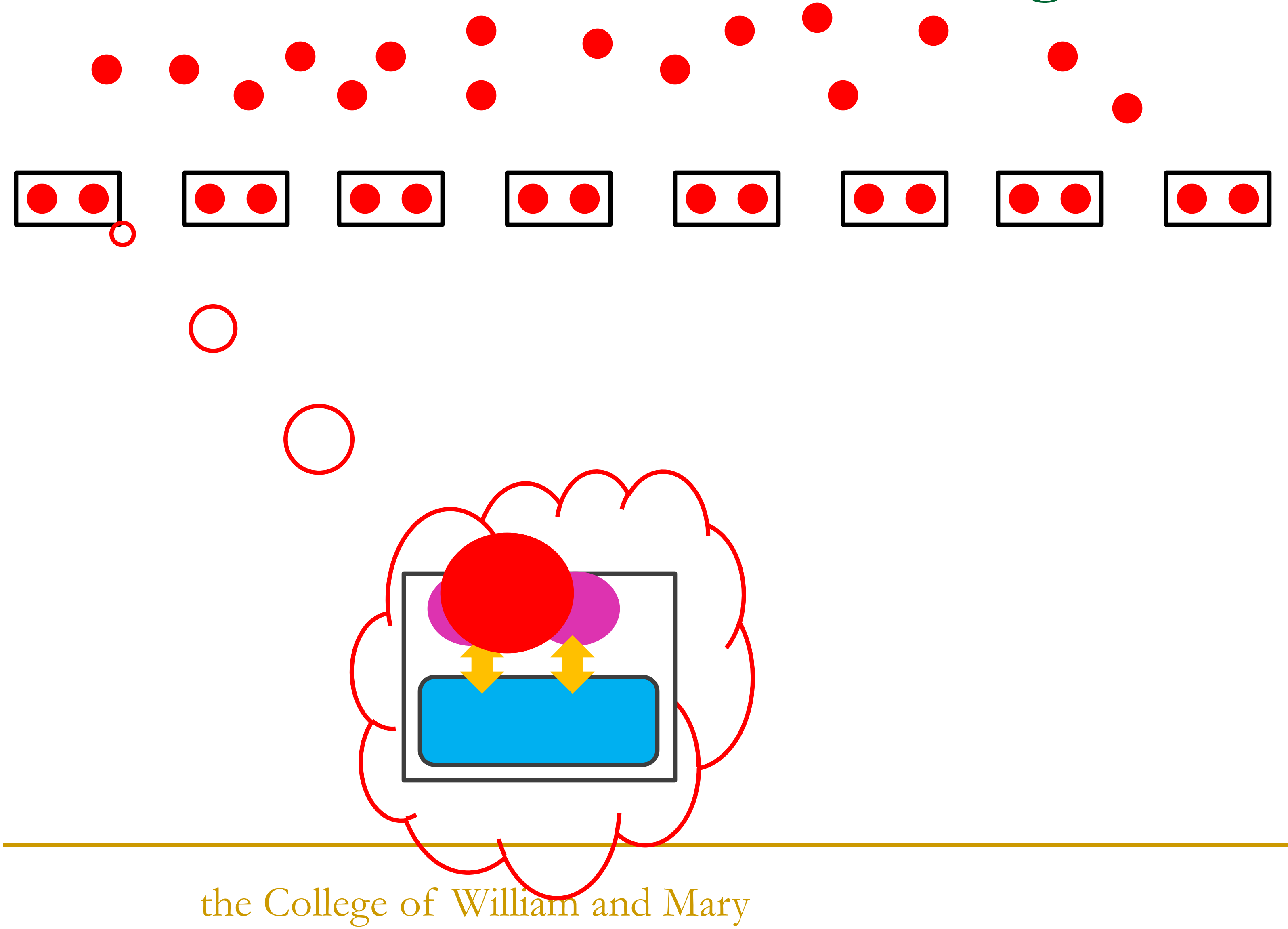
- Dual-core system optimal solution



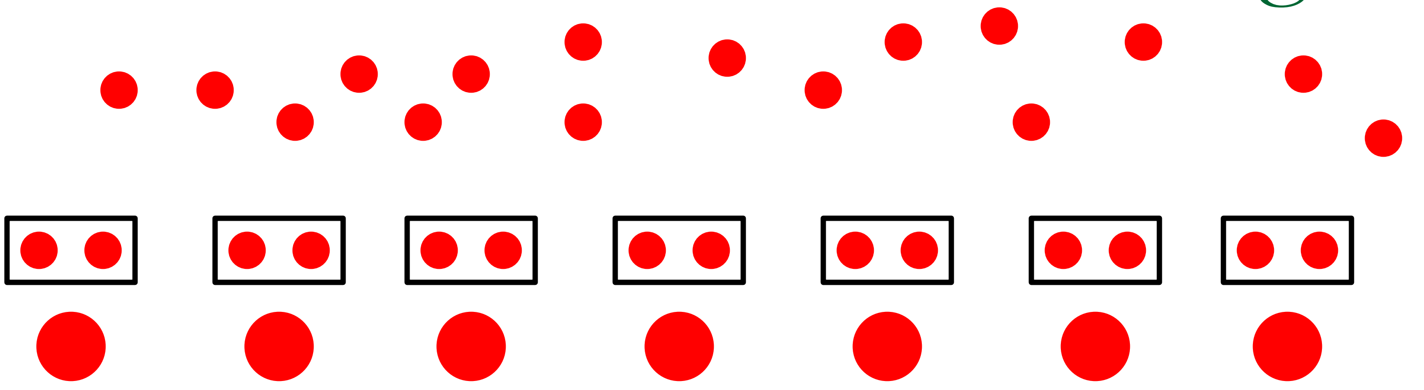
Hierarchical Perfect Matching



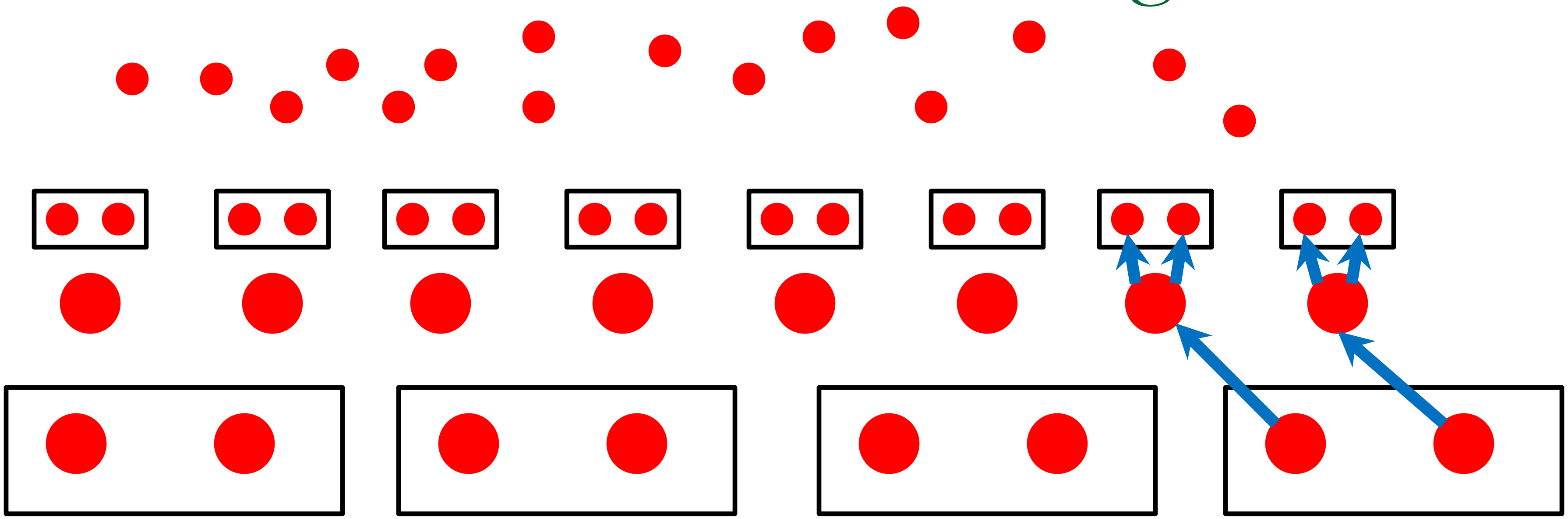
Hierarchical Perfect Matching



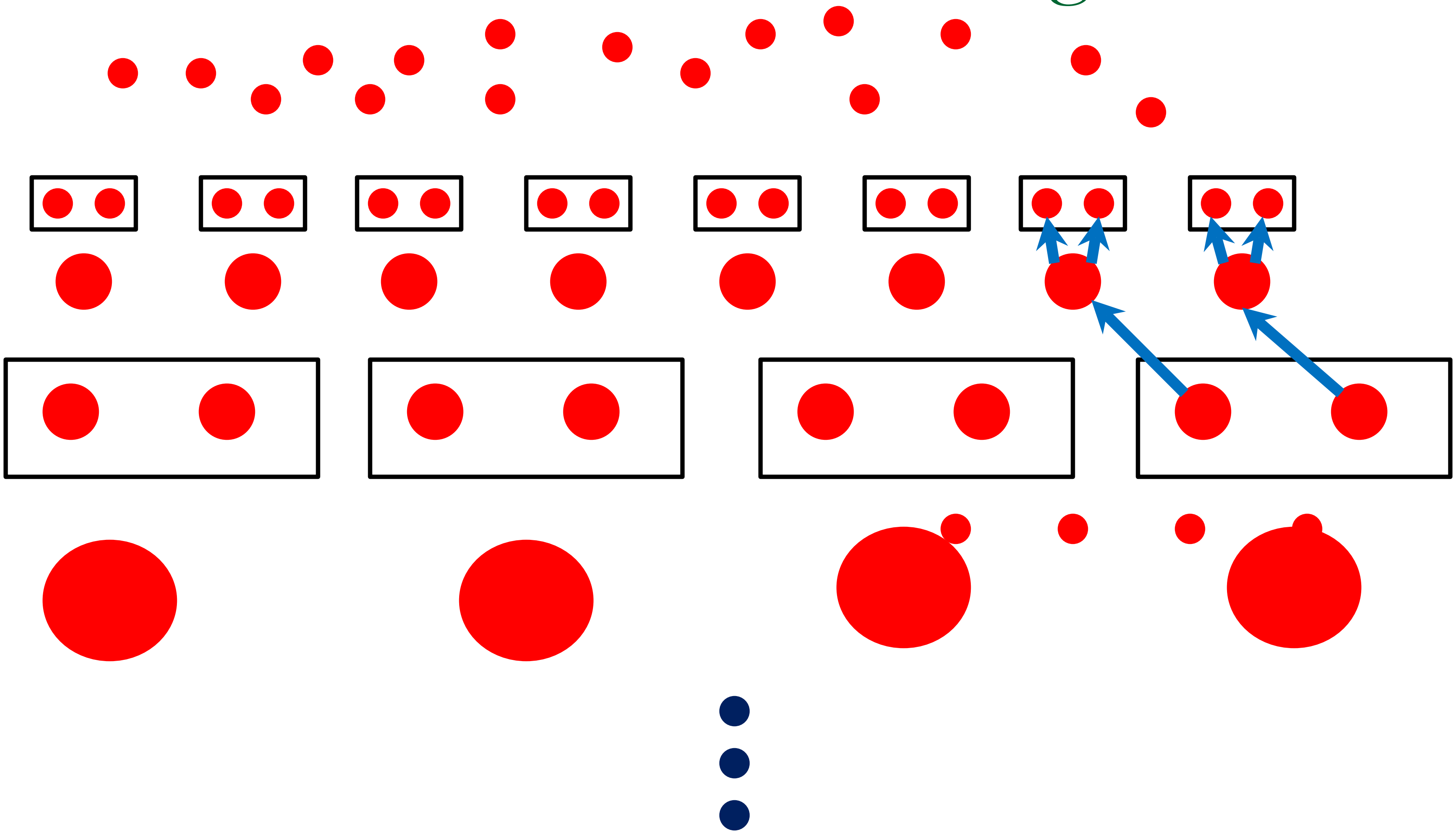
Hierarchical Perfect Matching



Hierarchical Perfect Matching



Hierarchical Perfect Matching



Greedy Algorithm

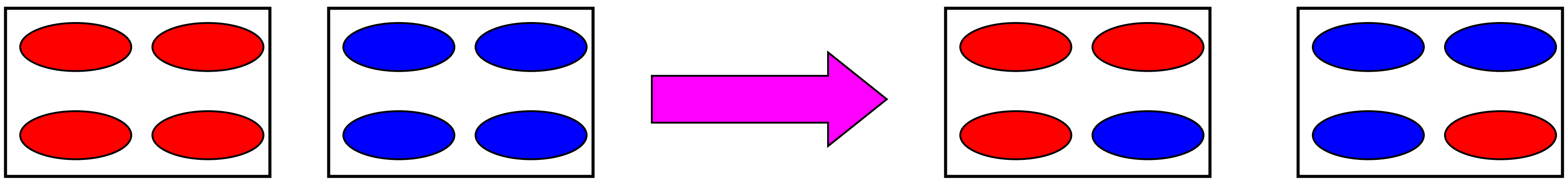
- Basic idea
  - Schedule the least “polite” job first
- “politeness” of a Job
  - Sum of degradations of all the assignments contain this job.
  - Impact of a job on others

Greedy Algorithm

I. Sort unassigned jobs based on politeness
II. Pick the least politeness job J to schedule
III. Add assignment contains J with least degradation to schedule
IV. Update unassigned job list

Local Optimization

- Main Scheme



for i ← 1 to K-1

  for j ← i+1 to K

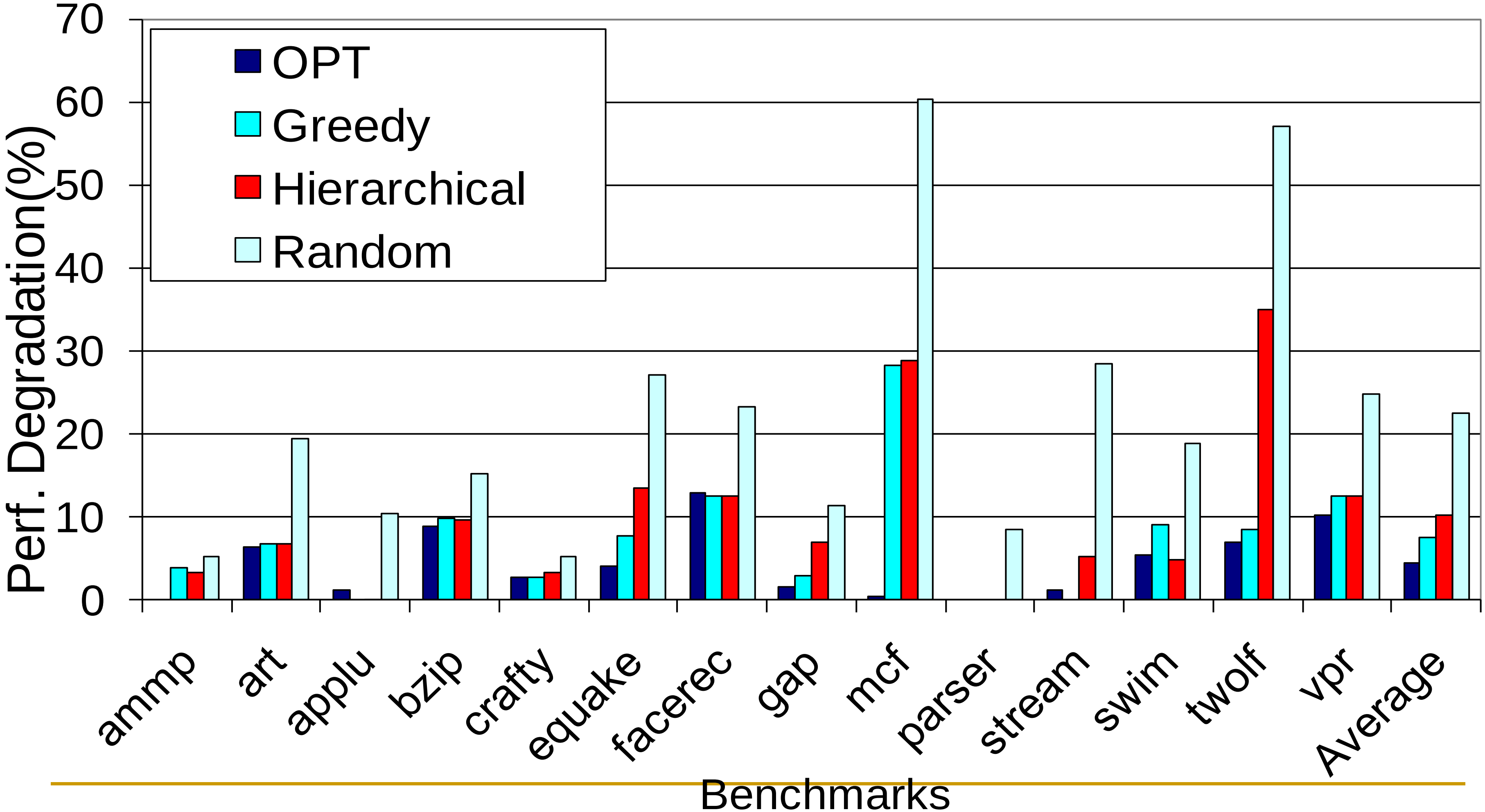
    Local-Optimization(i, j)

K: number of assignments

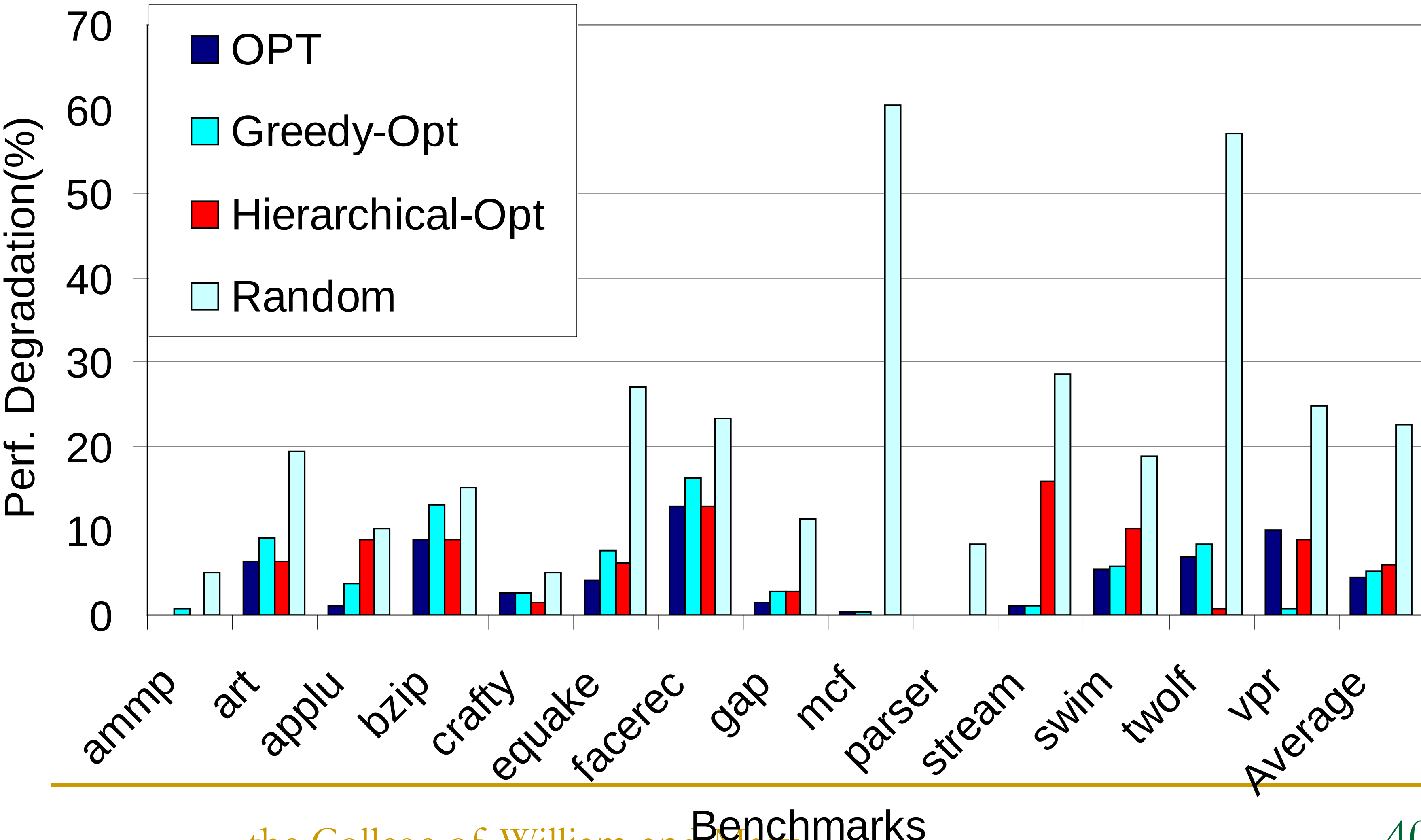
Performance Evaluation

- Machine
  - AMD Opteron 4 core processors
- Benchmarks
  - 15 SPEC CPU2000, 1 Stream
- Metrics
  - Performance Degradation
  - Scheduling time
  - Fairness

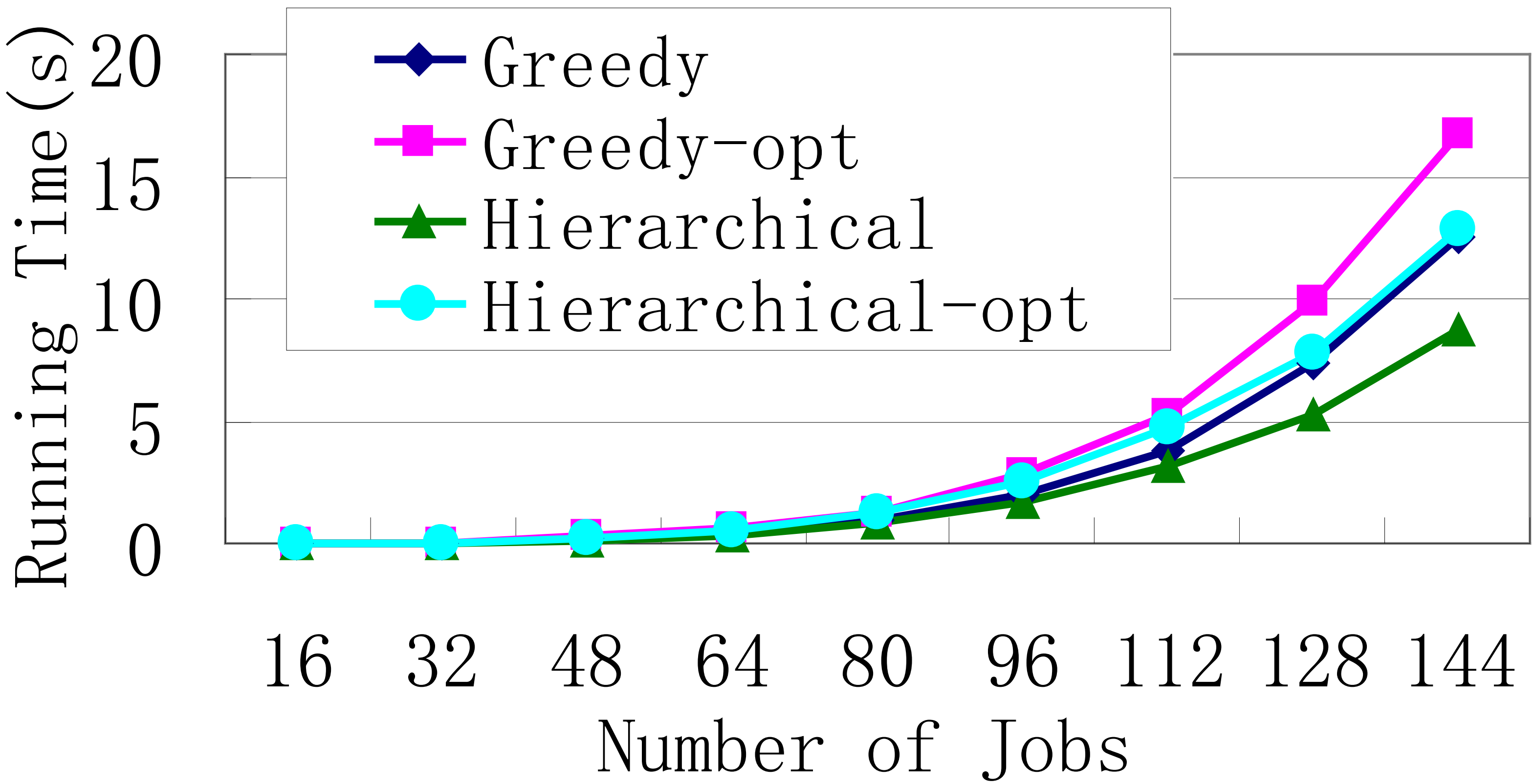
Performance Degradation



Performance Degradation

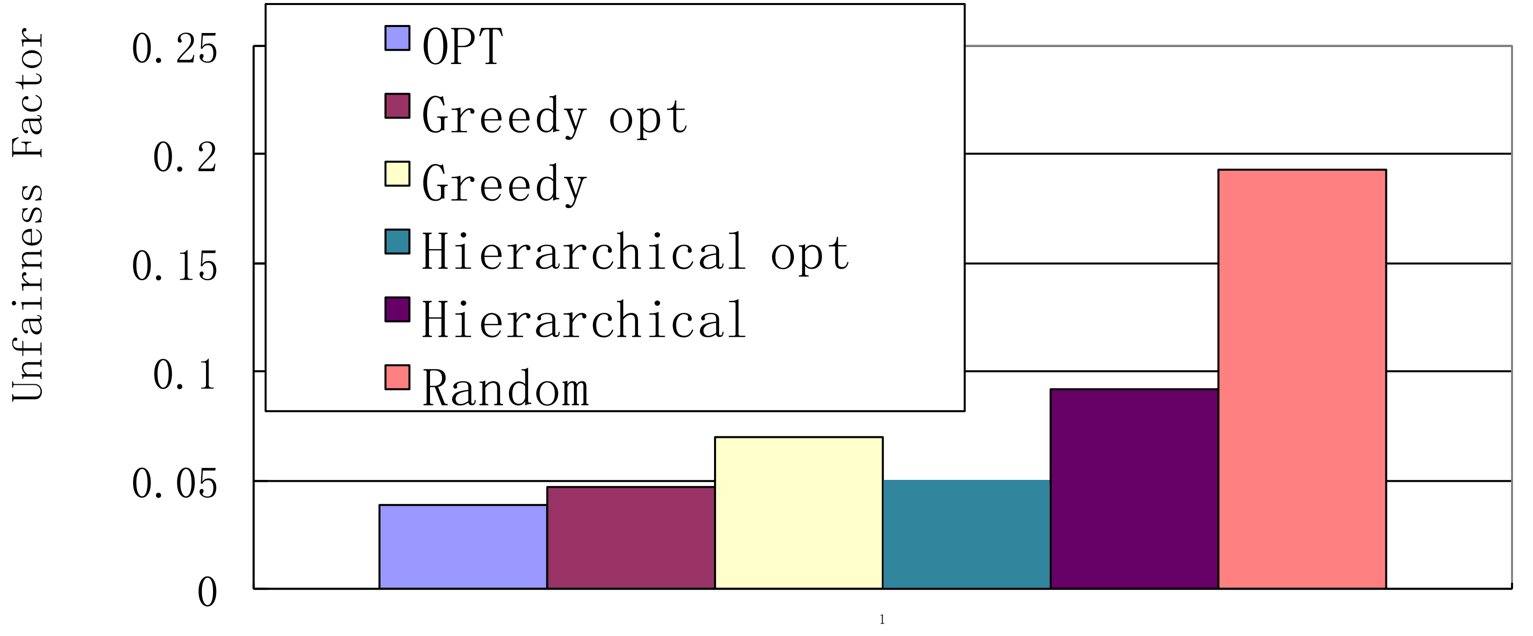


Scheduling Time



Fairness

- Unfairness Factor
  - Coefficient of Variation of normalized degradation



Conclusion

- Job co-scheduling on CMP is crucial
  - Different schedule performance varies
- Dual-core system
  - Polynomial solvable
- K-core (K>2) system
  - NP-Complete problem
  - Heuristics
    - Hierarchical
    - Greedy

Acknowledgement

- Weizhen Mao William and Mary
- Cliff Stein Columbia University
- William Cook Georgia Tech
- National Science Foundation
- IBM CAS Fellowship

Thanks!

Questions?