Mauro Bampo

Supervisors: Prof. Michela Milano,
Mark Wallace (Monash)

Collaborative Scheduling Algorithms for Multi-Processor Architectures

PhD Outline

Combinatorial Optimisation

Distributed Optimisation Frameworks

Scheduling

Hybrid Algorithms for Scheduling

Constructive-Local Hybrids for Shop Scheduling

Collaborative Algorithms for Processor Scheduling

Agenda

- Overview
    - Scheduling problems & decomposition
    - Collaborative algorithms
- A Practical Problem
- Implementation
- Results & Future work

Agenda

- Overview
  - Scheduling problems & decomposition
  - Collaborative algorithms
- A Practical Problem
- Implementation
- Results & Future work

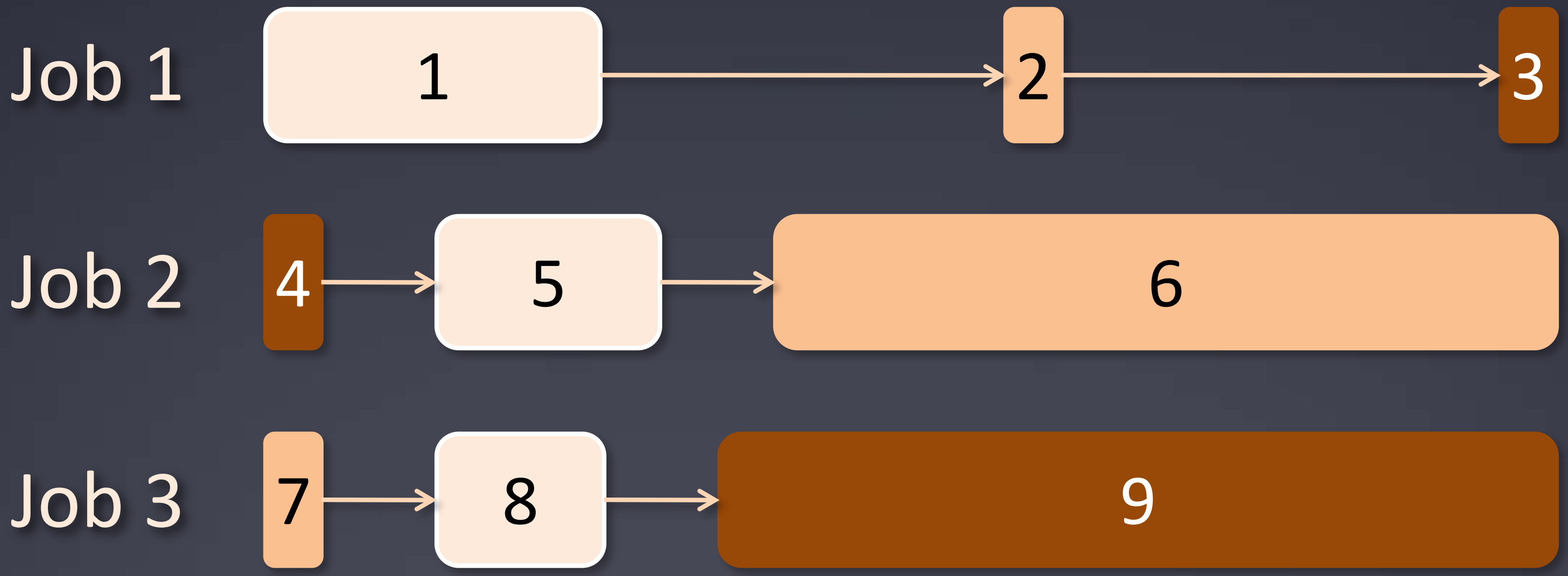
Scheduling Problems

Scheduling

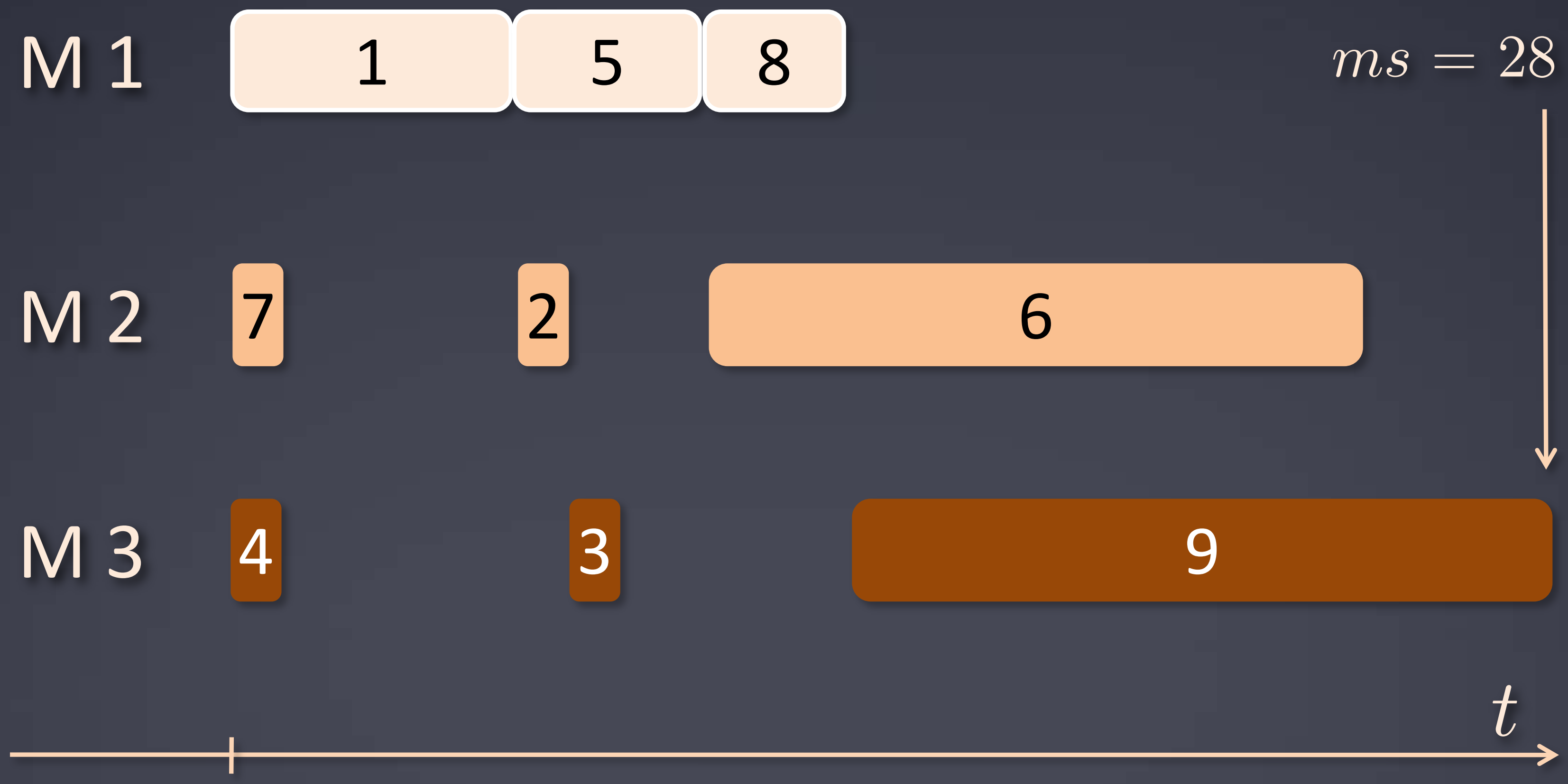
“ a **decision-making**
process that deals with the
allocation of **resources**
to **tasks** over given **time** periods
and its goal is to optimize
one or more **objectives**. ”

— M.L. Pinedo

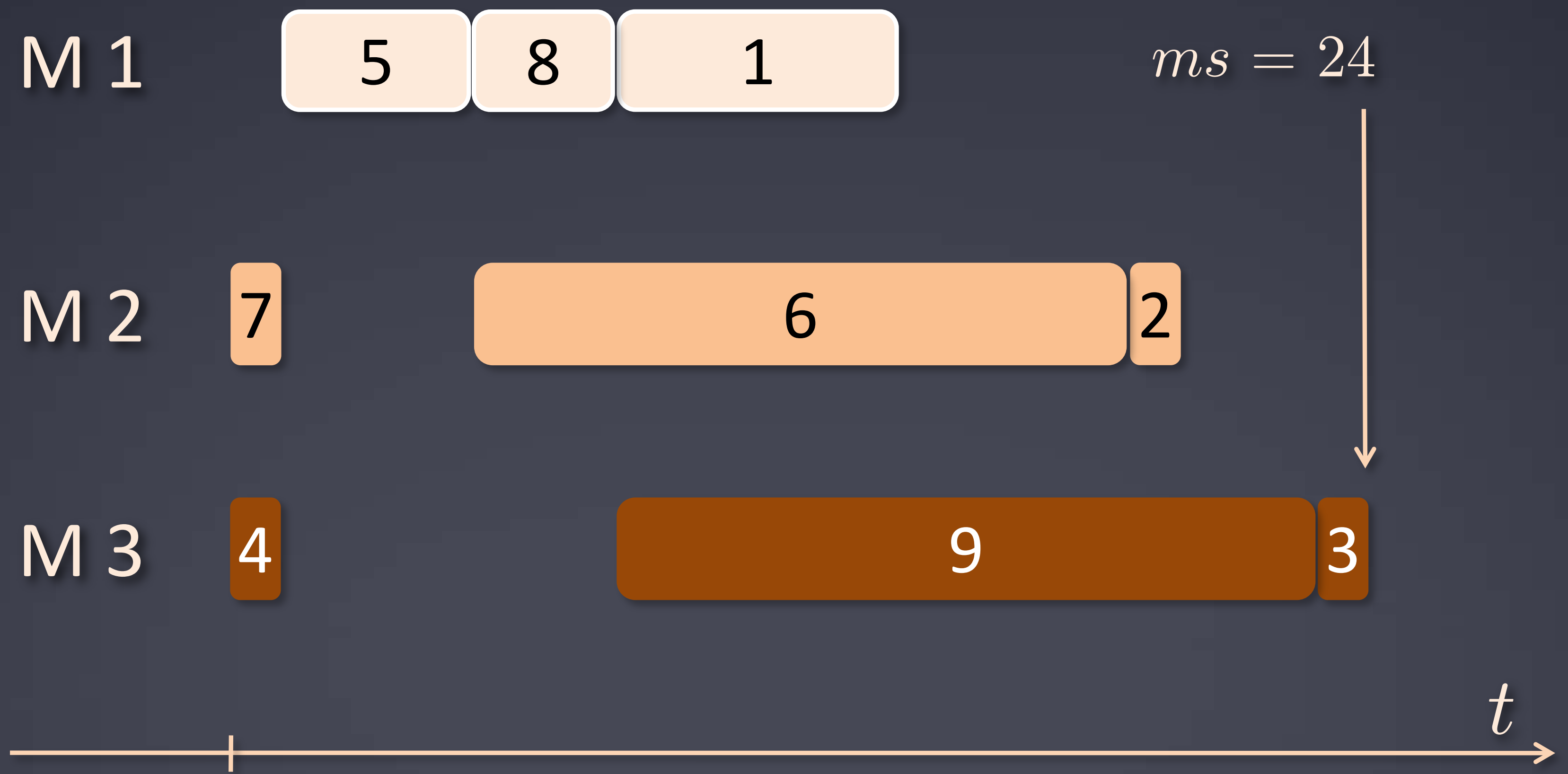
Example



Trivial Solution



Optimal Solution



Solving a complex problem

Operations Research:

Mixed Integer Programming

Branch + Bound/Cut/Price

Strength: Optimal Algorithms

AI:

Tree Search +

Constraint Programming

Strength: Feasibility

When the problem grows too large

Heuristic Algorithms:

Finding good solutions
in acceptable time

We adopt both

Problem Decomposition:

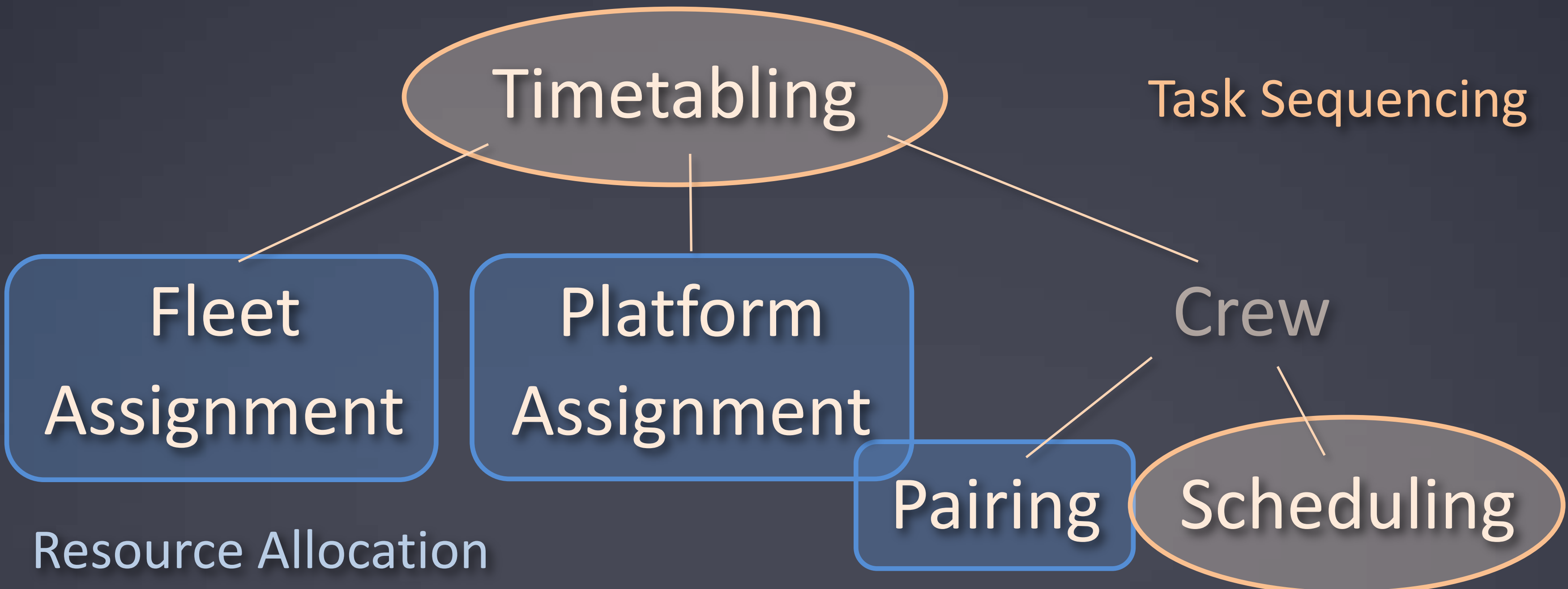
- Break the problem into simpler parts
- Solve the sub-problems separately
- Determine a solution to the original problem

Tricky ↓

Decomposition:

Scheduling = Allocation + Sequencing

Example: ***Rail Transport***



The classical way

Master Problem

Allocation

Solution

No-Goods

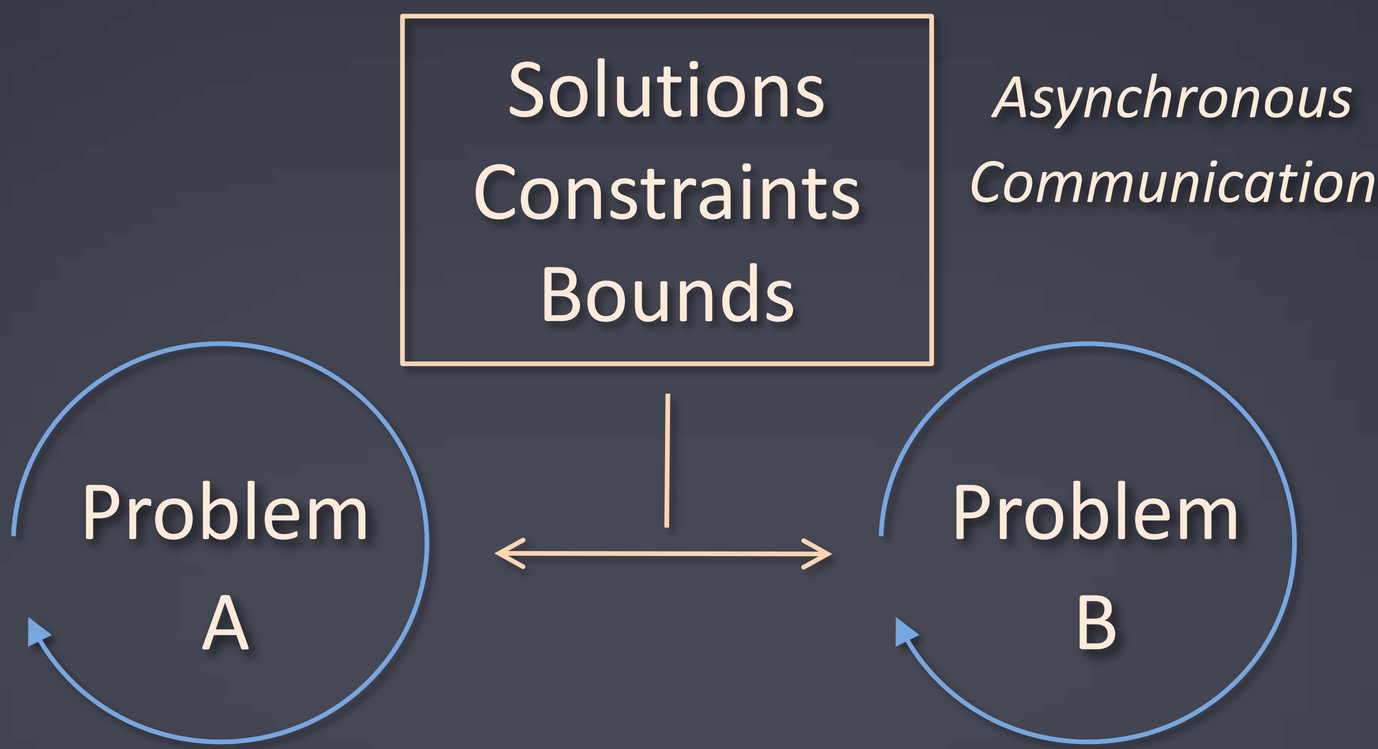
Bounds

Sub-Problem

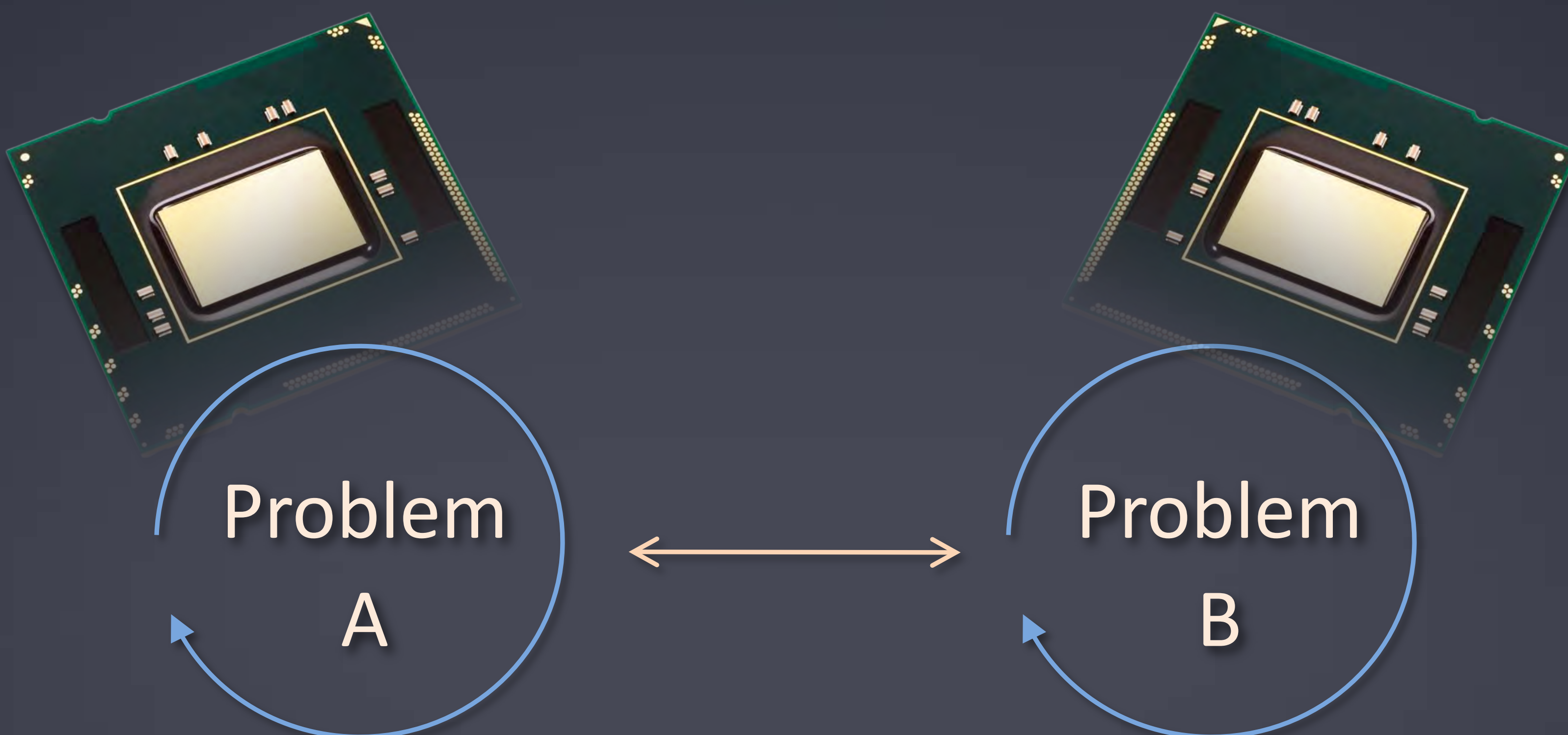
Sequencing

A new methodology

Collaborative Optimization



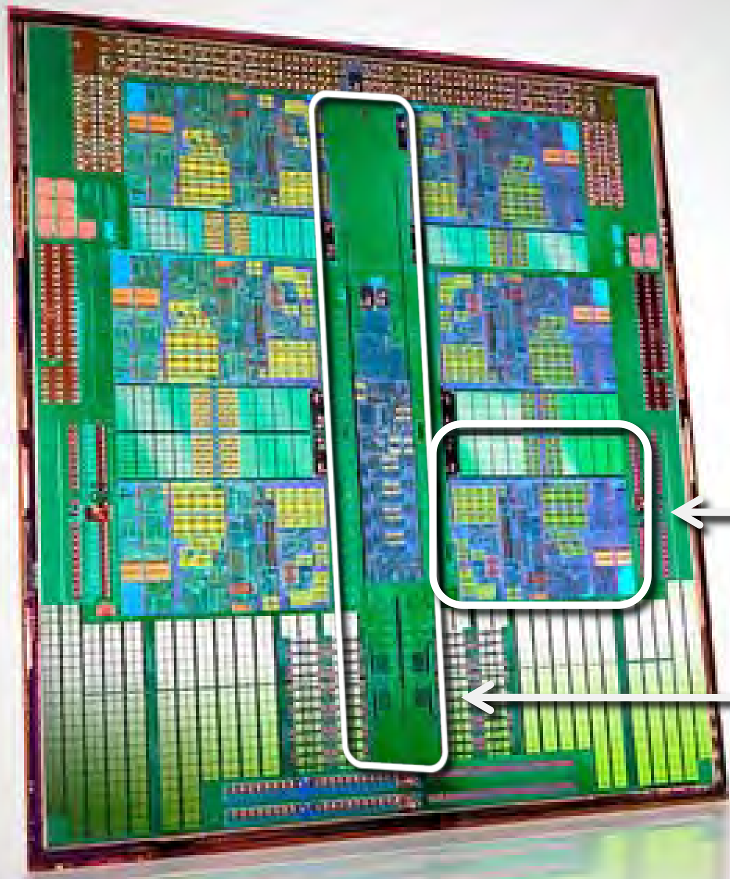
Exploiting Parallel Architectures



Agenda

- *Overview*
  - *Scheduling problems & decomposition*
  - *Collaborative algorithms*
- *A Practical Problem*
- *Implementation*
- *Results & Future work*

Multi**P**rocessor **S**ystems-**o**n-**C**hip



- Many low power cores
- General purpose
- Often heterogeneous

Processing Element

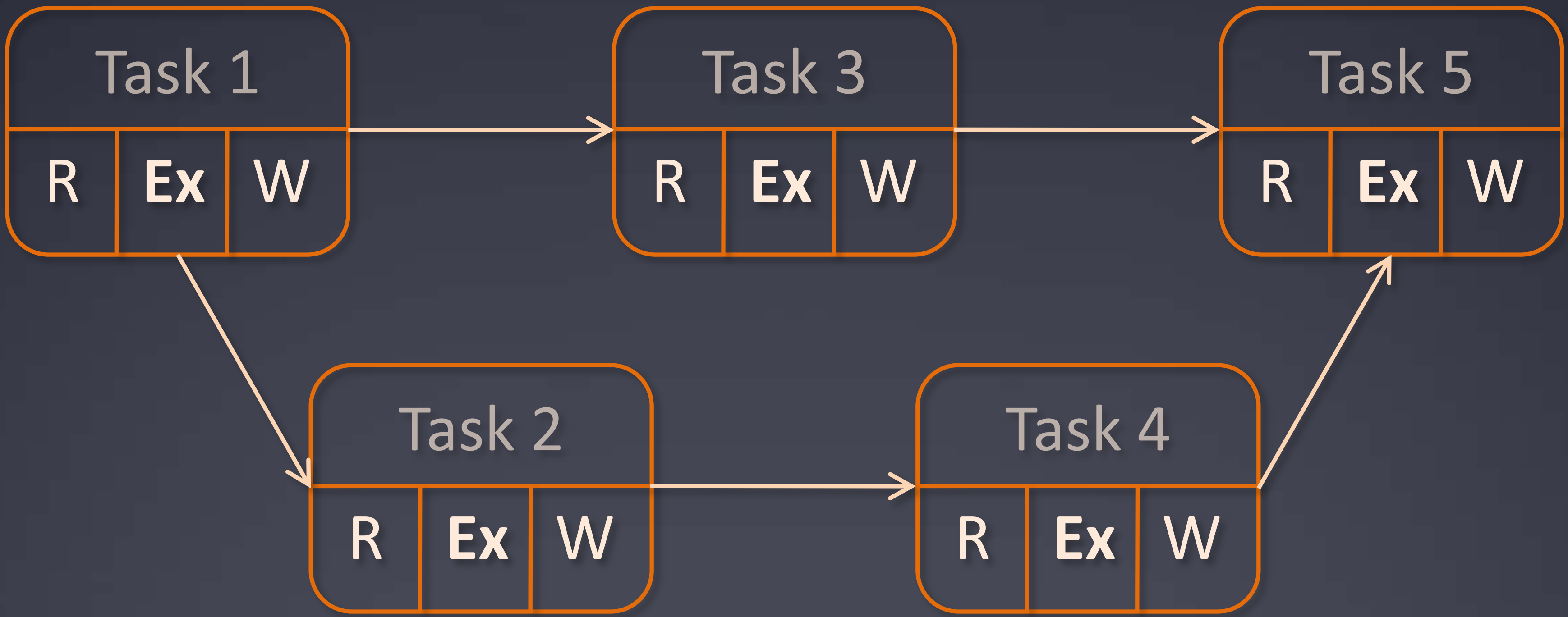
Interconnection Bus
& Memory Interface

MPSoC Applications

Embedded systems with parallel applications/multitasking and low power consumption

Programming Model & Motivation

A Parallel Application



→ Task Graph

Programming Model

Programmer

→ *Application (Task Graph)*

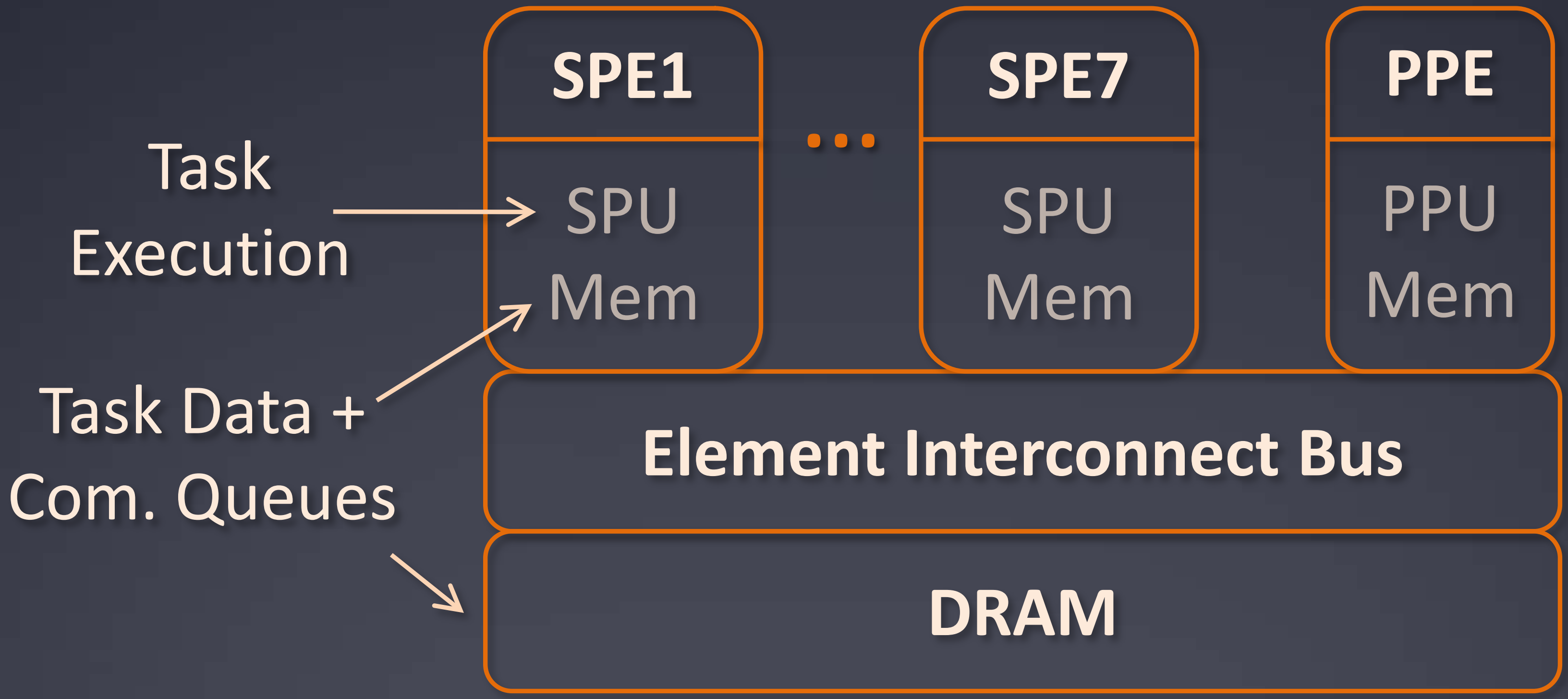
→ Compiler

Optimise this step! → Compiler

→ *Executable (Tasks + Control)*

The Target Platform

The IBM CELL Architecture



Read/Write times (resource usage)
depend on memory choices: hard!

Previous work

Integer Programming

Best on communication- intensive applications

Constraint Programming

Best on communication- intensive applications

Complete approaches: limited scalability

Objectives

Testing the collaborative algorithm idea

Higher scalability at price of completeness

Heuristic choices

Agenda

- Overview
    - Scheduling problems & decomposition
    - Collaborative algorithms
- A Practical Problem
- Implementation
- Results & Future work

Decomposition

Decomposition

Allocation

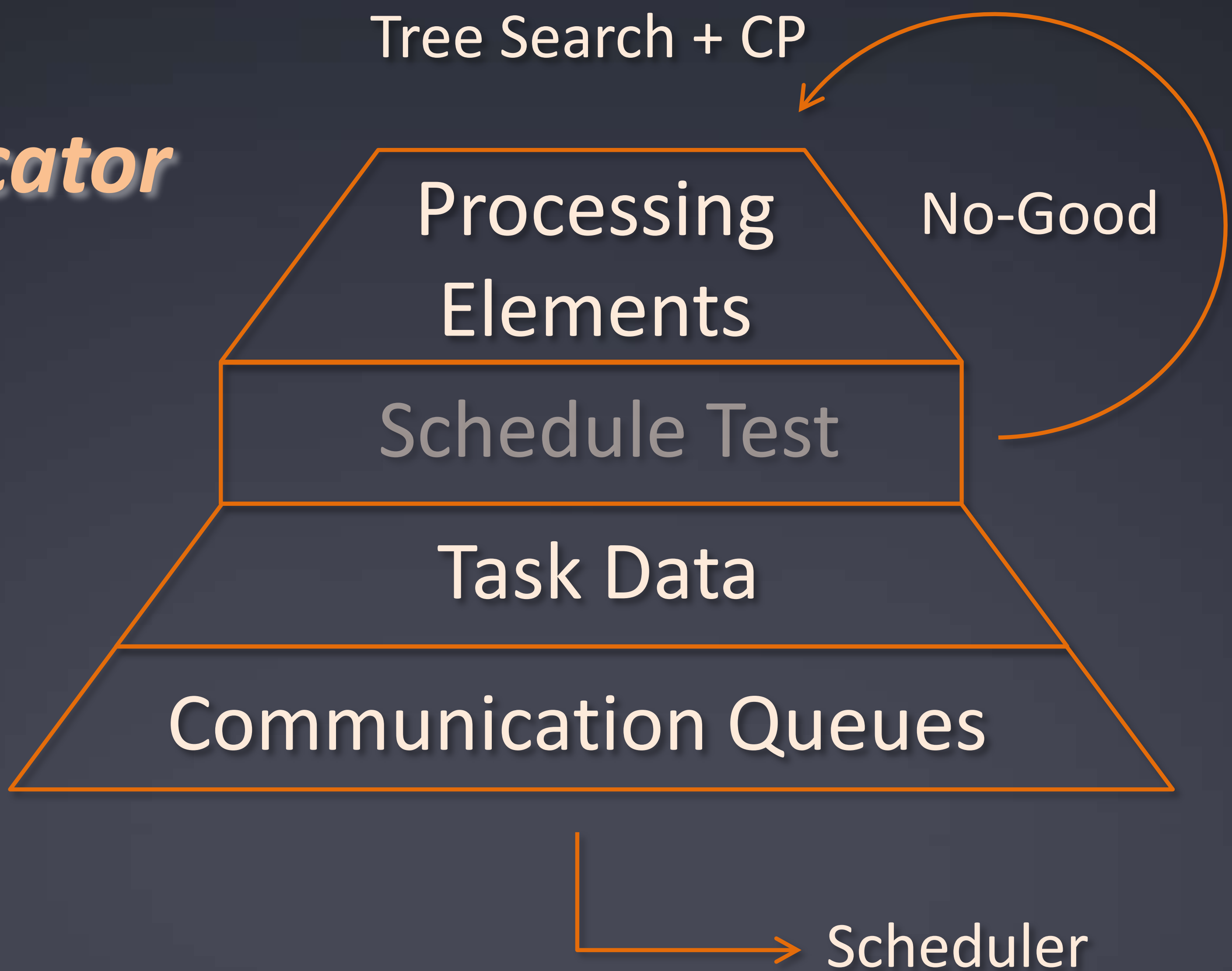
Processing Elements

Task Data

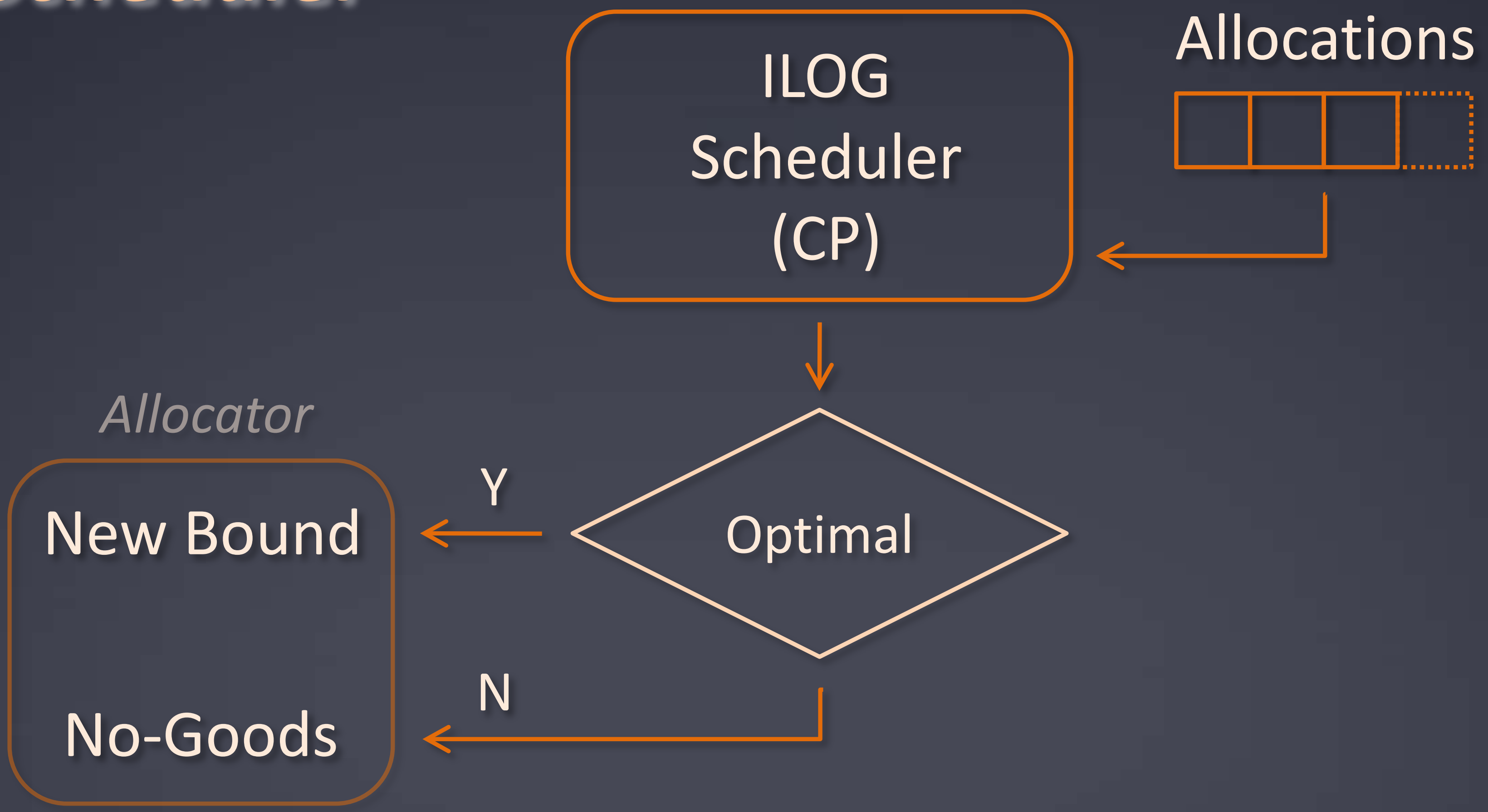
Communication Queues

Task Sequencing

Allocator


Tree Search + CP
No-Good
Processing Elements
Schedule Test
Task Data
Communication Queues
Scheduler

Scheduler


ILOG
Scheduler
(CP)
Allocations
Allocator
New Bound
No-Goods
Y
N
Optimal

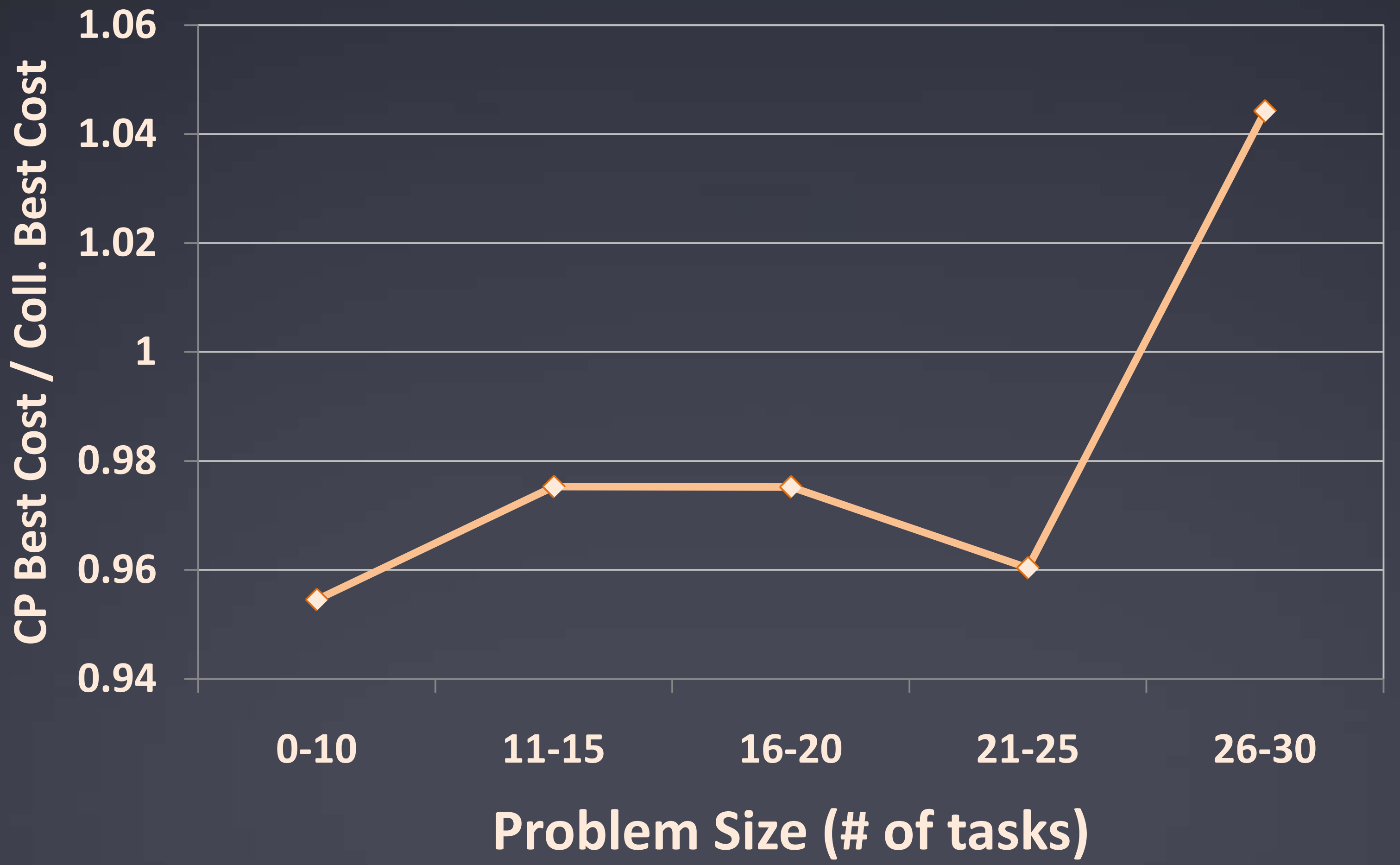
Agenda

- Overview
  - Scheduling problems & decomposition
  - Collaborative algorithms
- A Practical Problem
- Implementation
- Results & Future work

Expected results:

Lower performance than complete approaches on small problems, best with large instances thanks to the heuristic component.

Collaborative vs CP



~30% Speedup on 2 cores

Future work

More testing…

Allocation dominates…

More on allocation:
Local search

More on allocation:

Different heuristics and better reasoning

Collaborative solutions for different domains

→ Transport

?