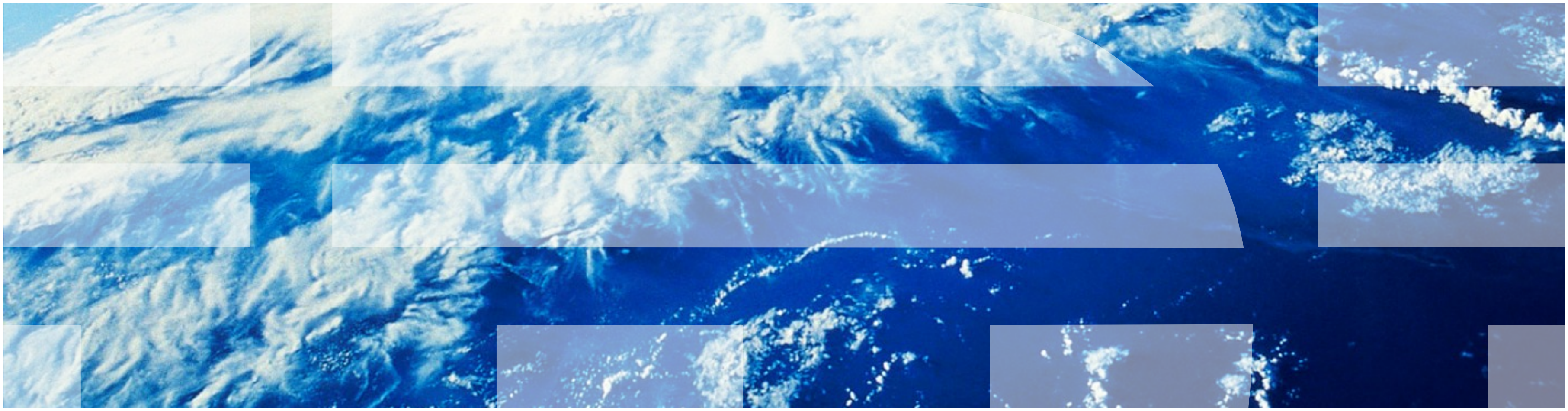
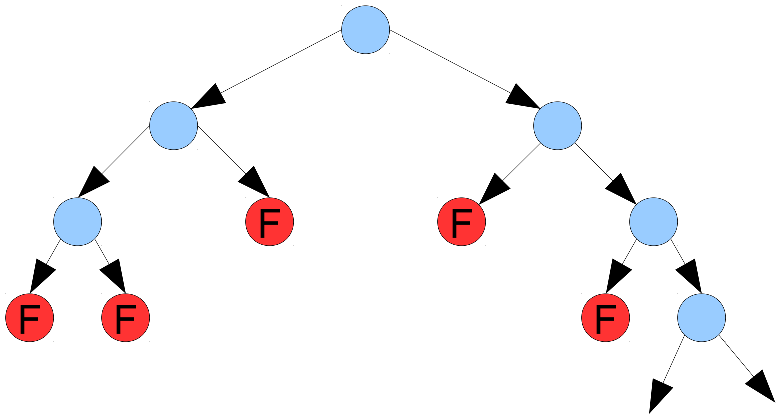
Scheduling Search (in IBM ILOG CP Optimizer)

CP Optimizer Development Team

Petr Vilím

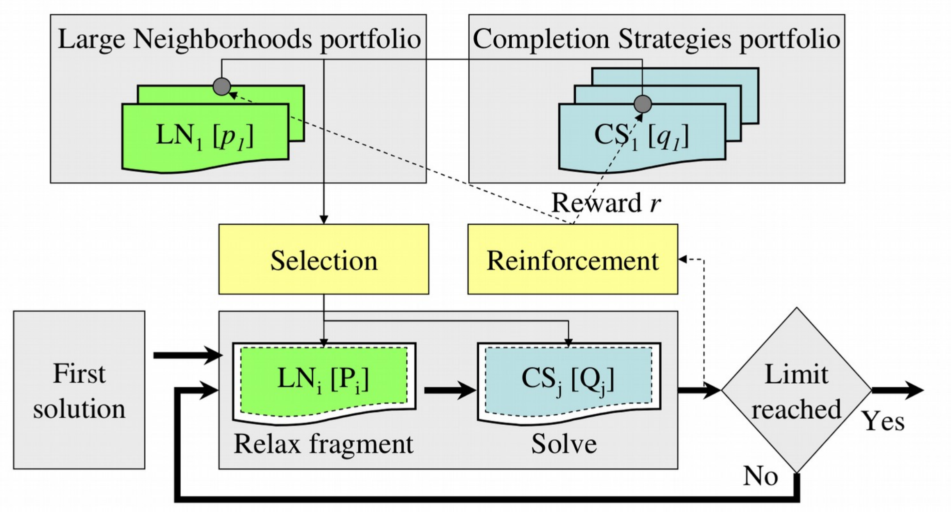


Search Types



- **Depth First**
  - Chronological SetTimes

- **Large Neighborhood Search** (default)
  - With Failure-directed search

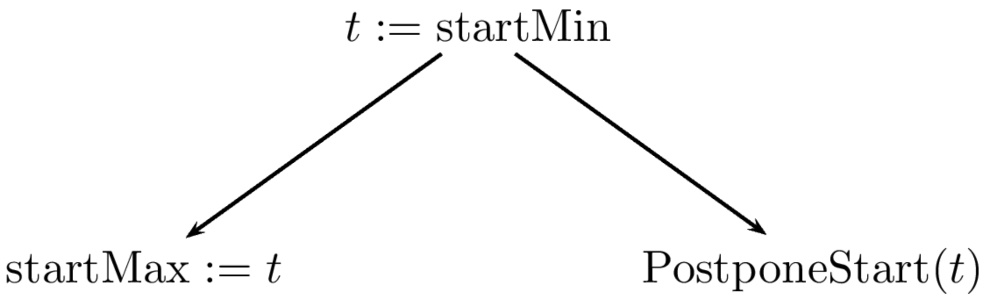


- **Multipoint Search**
  - Uses genetic algorithms to combine solutions

Chronological SetTimes

SetTimes search

- Build schedule chronologicaly, scheduling ASAP.
- Take unscheduled and not-postponed interval with minimum startMin. The value of the startMin is current “horizon”.
- In left branch, assign the start time to the startMin.
- In right branch mark the interval as “postponed”.
- Dominance rule: interval variables with endMin < horizon are set to absent.



$t := \text{startMin}$

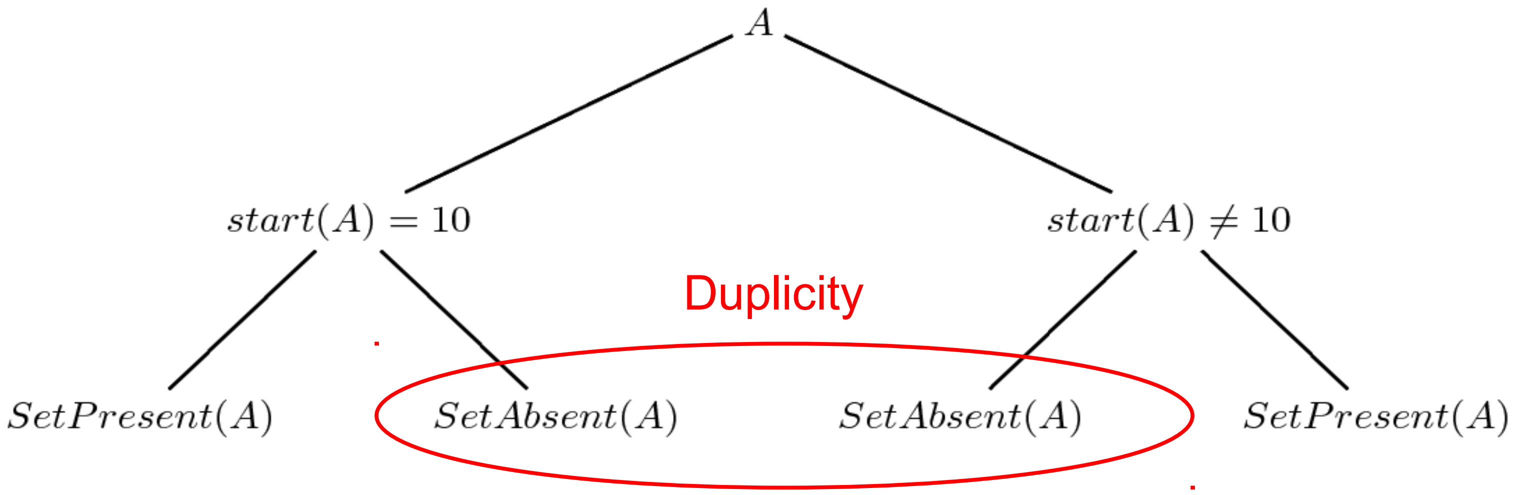
$\text{startMax} := t$

$\text{PostponeStart}(t)$

- Because of the dominance rule, SetTimes search is often incomplete (e.g. when there are precedences with negative delays).

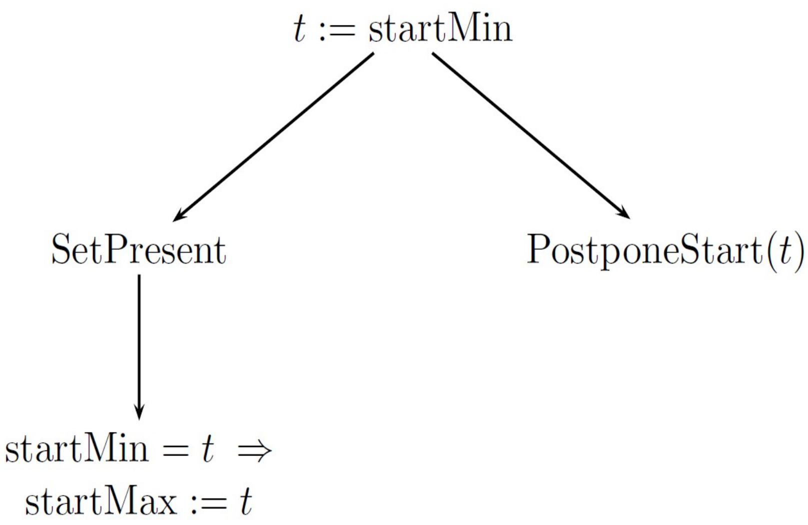
SetTimes search with Optional Interval Variables

- For a given optional interval variable first decide its presence and only then (if present) its start time and end time.
    - if we decide first the time then we can duplicate part of the search tree: the part where the interval is absent:



SetTimes search with Optional Interval Variables

- We want to schedule optional interval on its current startMin time *t:*



$t := \mathrm{startMin}$

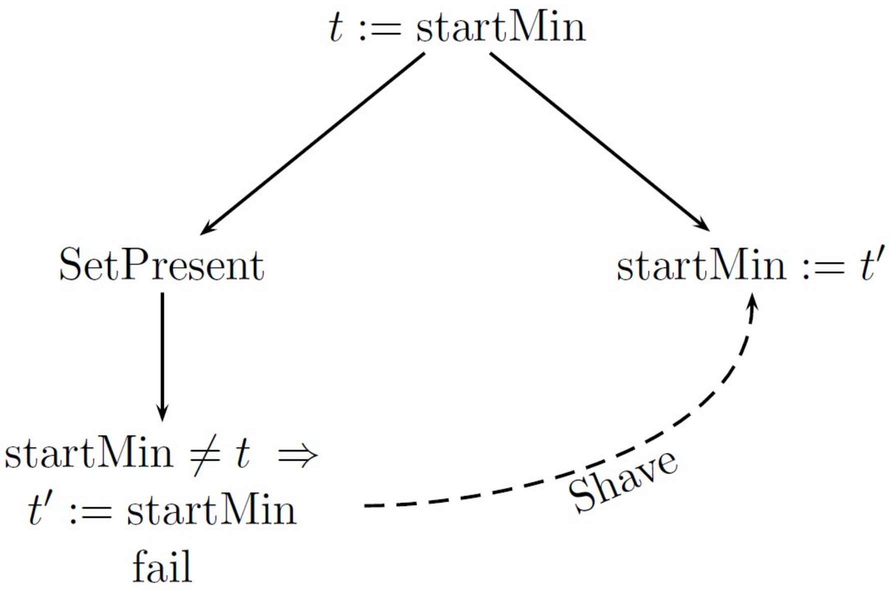
SetPresent

PostponeStart(t)

$\mathrm{startMin} = t \ \Rightarrow$

$\mathrm{startMax} := t$

- SetPresent may change current startMin of the interval variable.
- We repair it and branch on possibly different interval with better startMin.



$t := \mathrm{startMin}$

SetPresent

$\mathrm{startMin} := t'$

$\mathrm{startMin} \neq t \ \Rightarrow$

$t' := \mathrm{startMin}$

fail

Shave

Large Neighborhood search

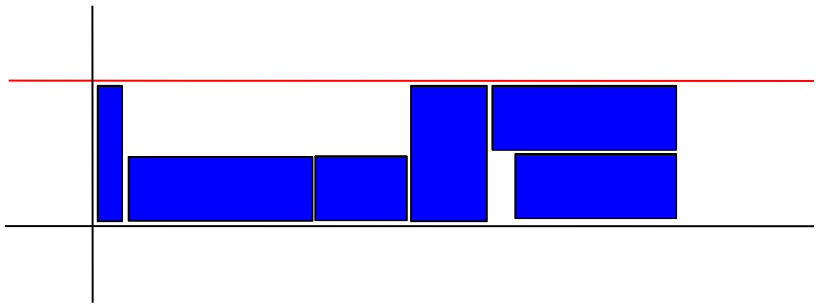
[1] Laborie, Godard: Self-Adapting Large Neighborhood Search:
Application to Single-mode Scheduling Problems.
MISTA-07

Interval (scheduling) search

- Scheduling search is based on Large Neighbourhood Search (LNS) also known as shuffling
  - First solution built by a portfolio of probing methods (mostly SetTimes)
- LNS:
  - Choose a fragment of the current solution to relax
  - Free all variables (backtrack to root)
  - Keep structure of the solution for variables outside of the fragment
  - Limit the objective variable to a better value
  - Run (limited) SetTimes search for similar but better solution

Improvement example

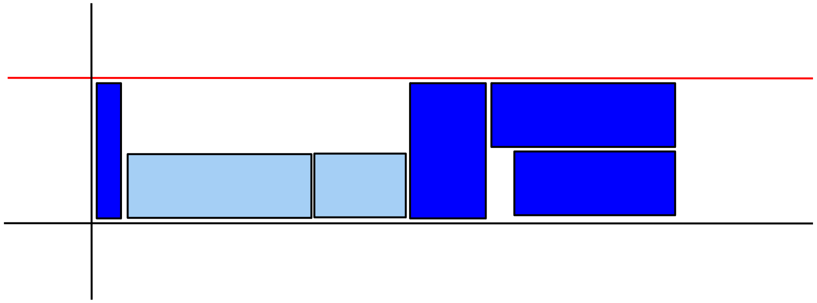
1. We have a solution.



Improvement example

1. We have a solution.

2. We take part of it and relax it.

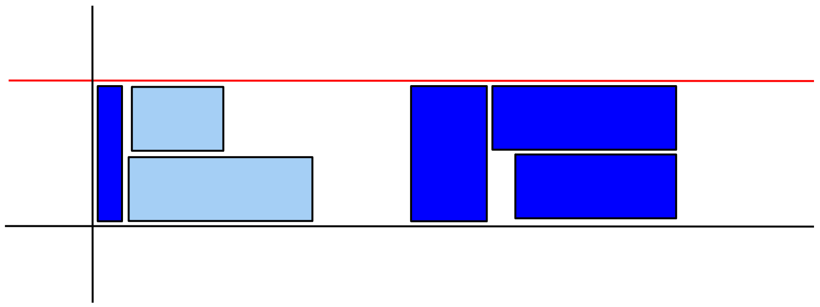


Improvement example

1. We have a solution.

2. We take part of it and relax it.

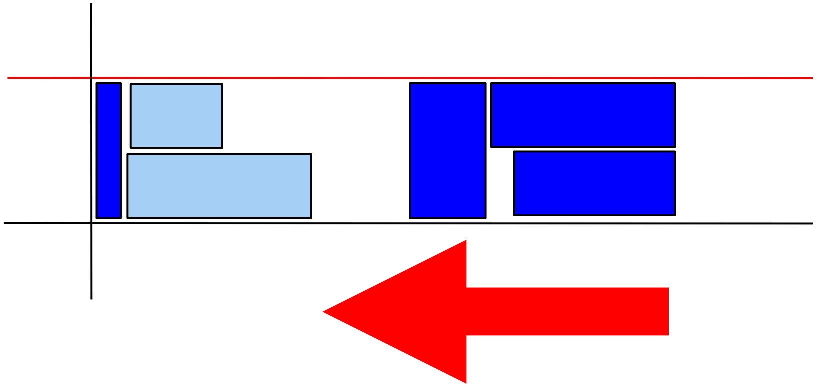
3. We find a new solution.



Improvement example

1. We have a solution.
2. We take part of it and relax it.
3. We find a new solution.

But to find a better solution we must relax also times in the fixed part!

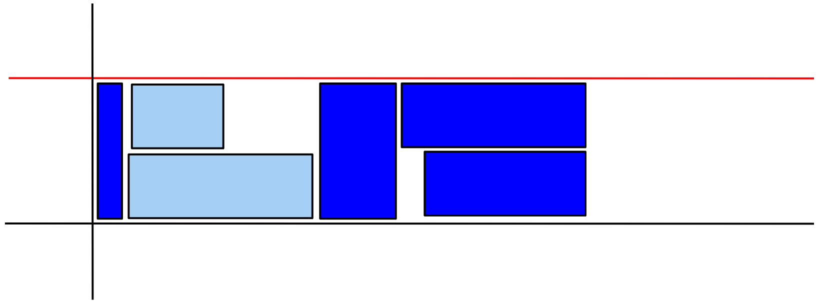


Improvement example

1. We have a solution.
2. We take part of it and relax it.
3. We find a new solution.

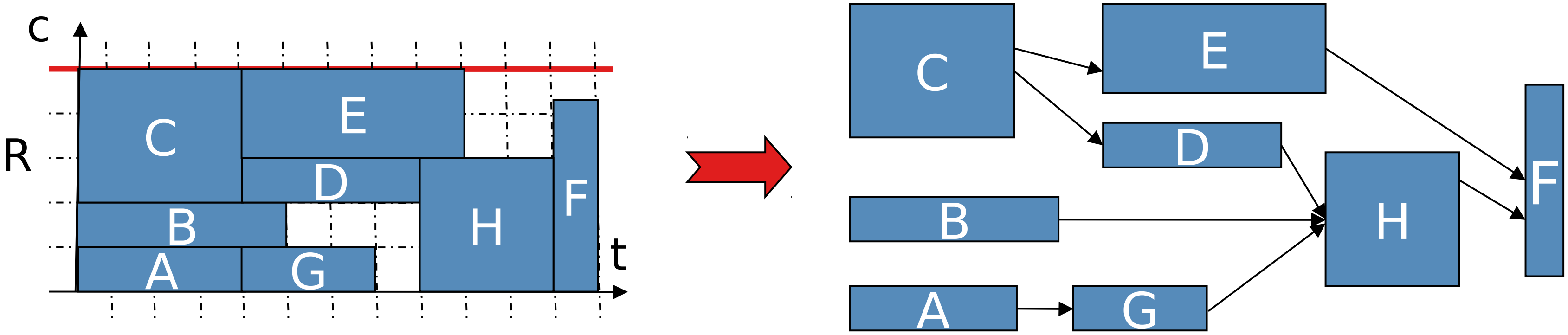
But to find a better solution we must relax also times in the fixed part!

In fixed part, we want to relax times, but not “structure“ of the solution.



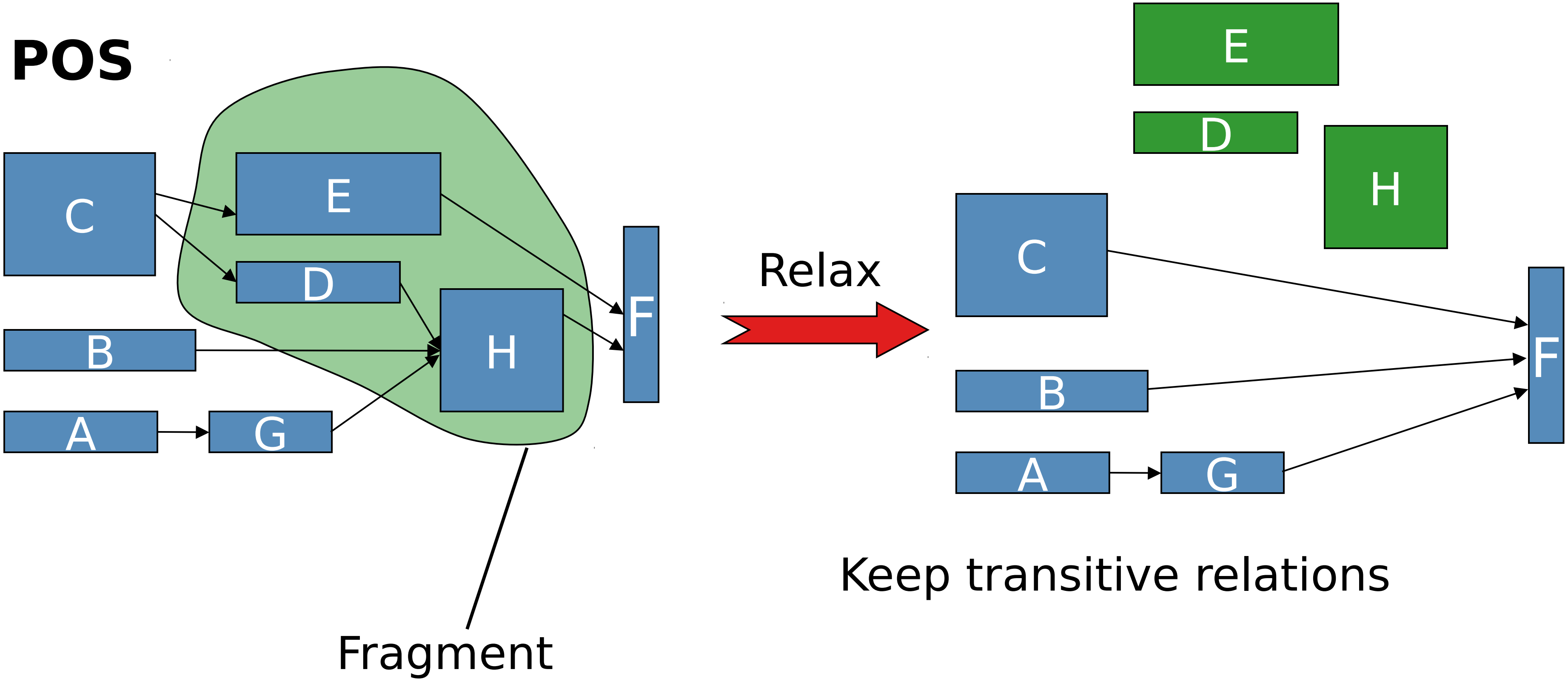
Partial Order Schedule

Partial Order Schedule (POS) is a relaxation which remembers the “structure“ of the solution without relaying on exact times.



It adds precedences such that any schedule satisfying these precedences also satisfies resource constraints.

Relaxation of a fragment

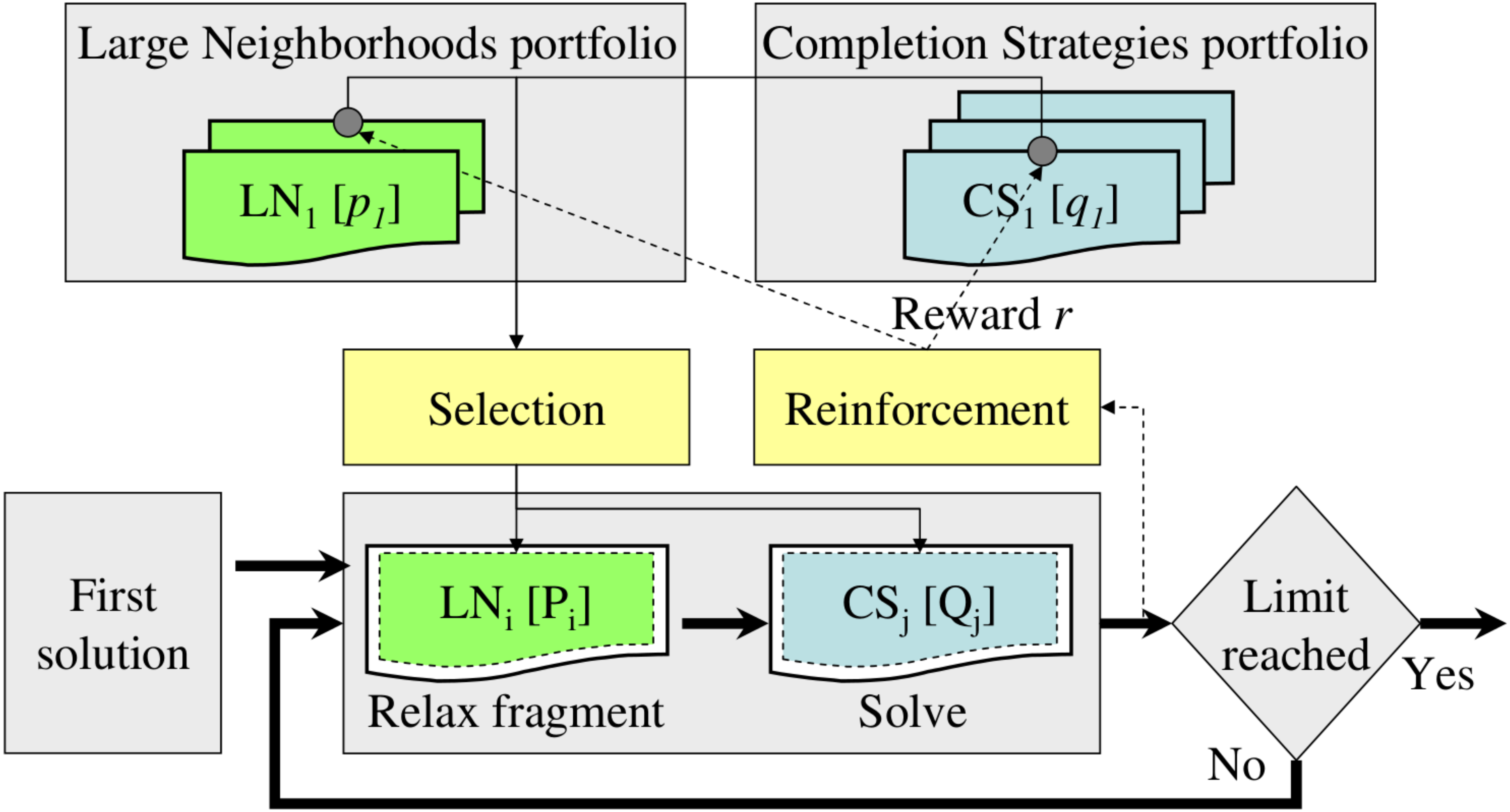


POS

Keep transitive relations

Fragment

Large Neighborhood Search


Large Neighborhoods portfolio
Completion Strategies portfolio
LN$_1$ [p_1]
CS$_1$ [q_1]
Reward r
Selection
Reinforcement
First solution
LN$_i$ [P$_i$]
CS$_j$ [Q$_j$]
Relax fragment
Solve
Limit reached
Yes
No