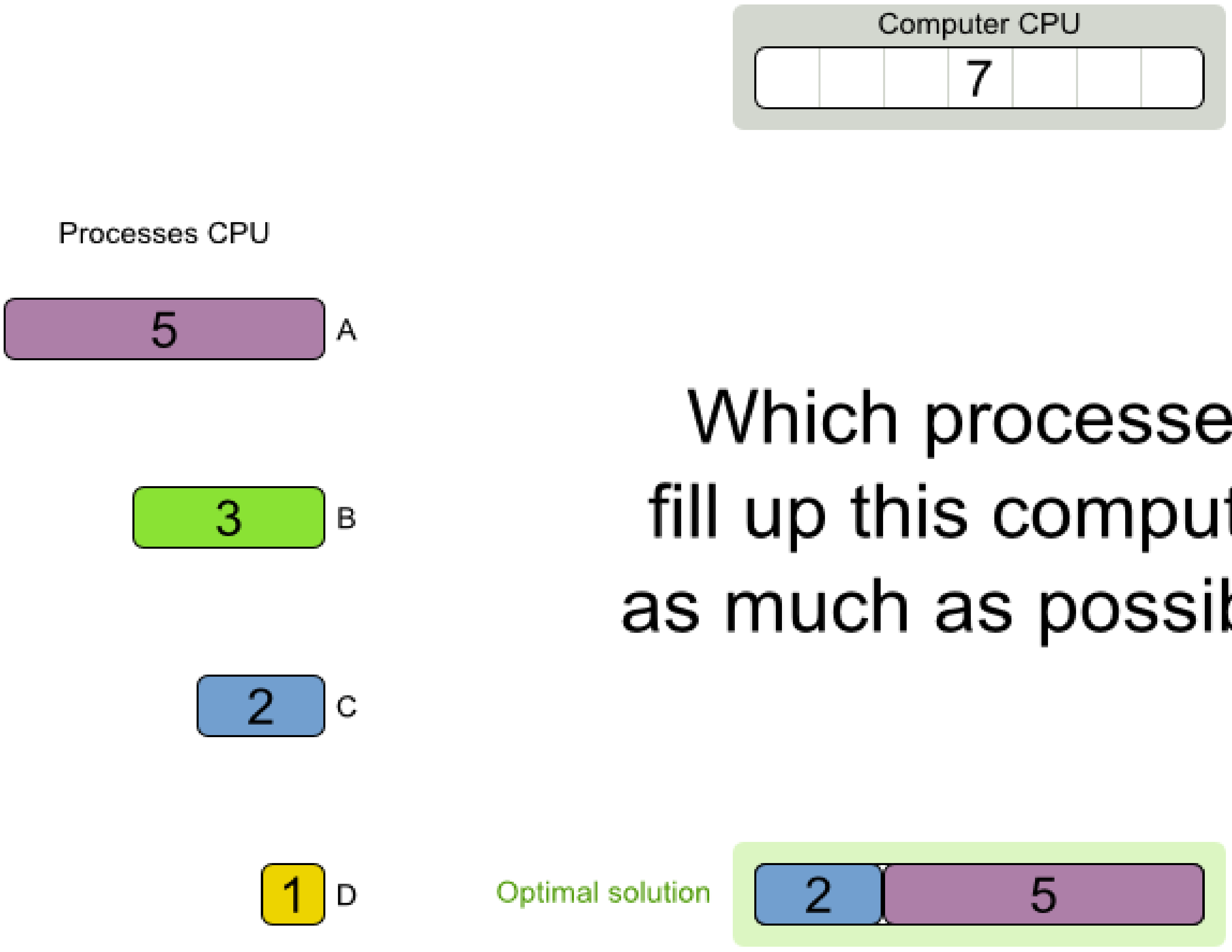
Real-time constraint solving (with OptaPlanner)

by Geoffrey De Smet

OptaPlanner lead

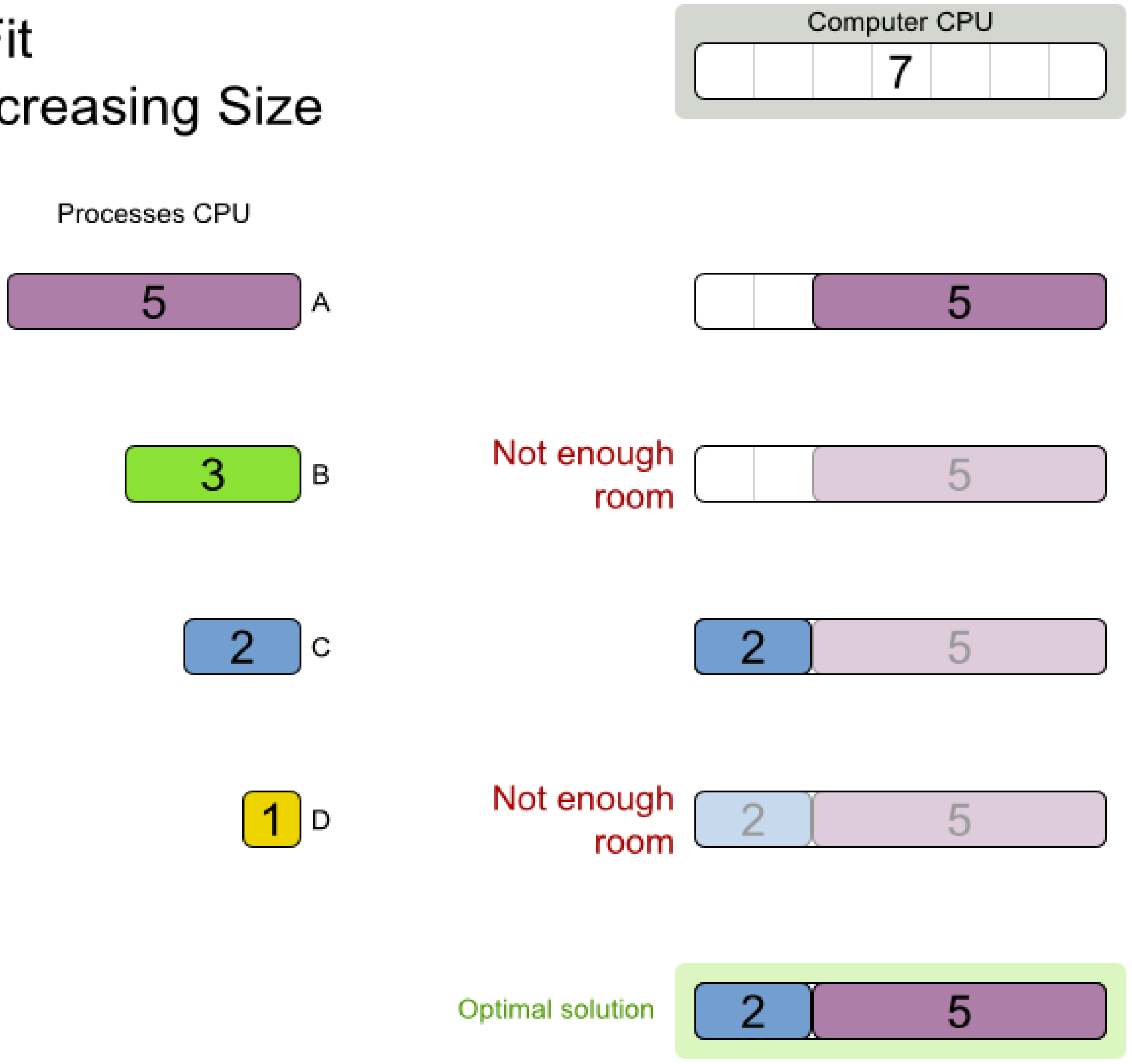
A different kind of decisions?



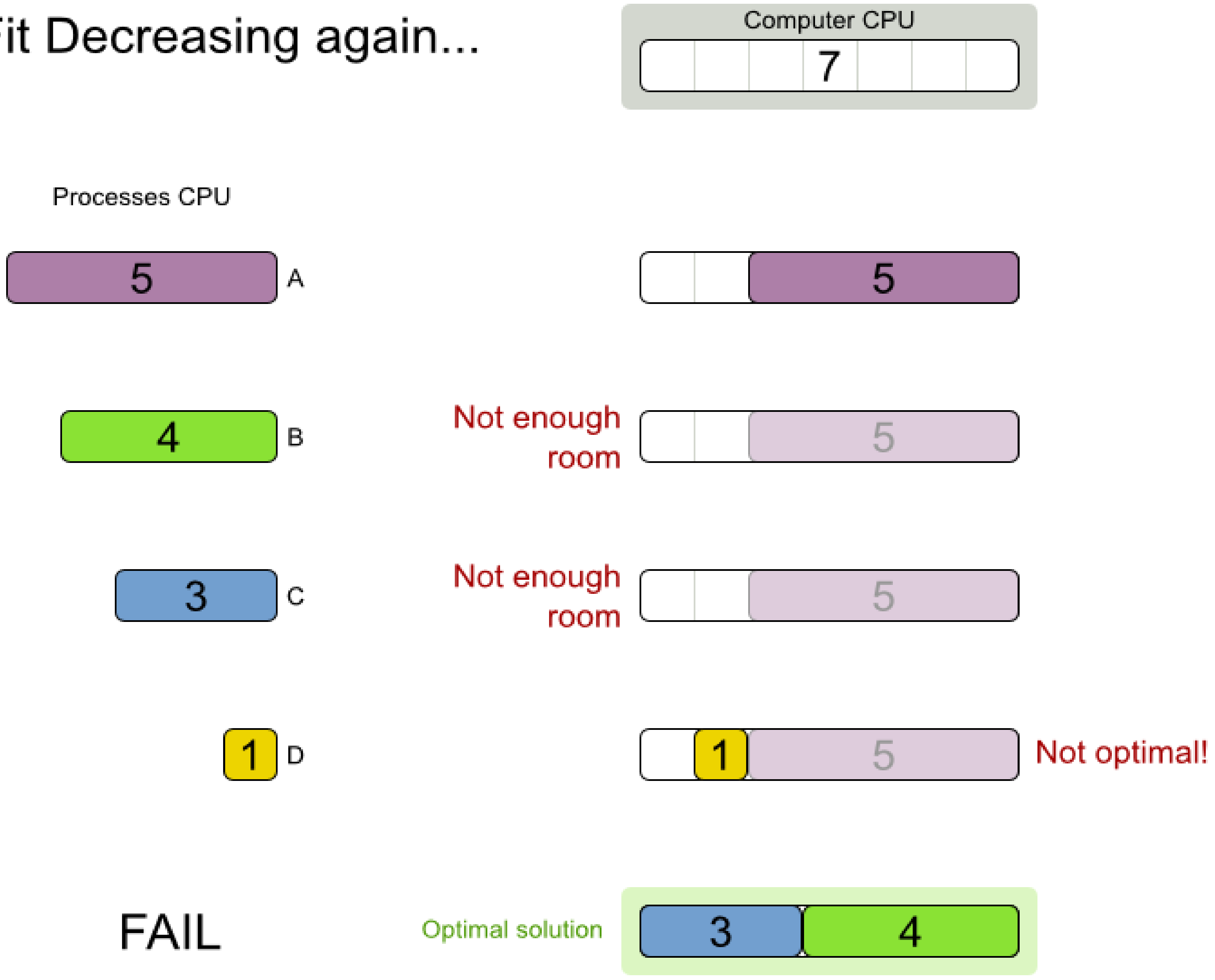
Which processes fill up this computer as much as possible?

How did we find this solution?

First Fit by Decreasing Size

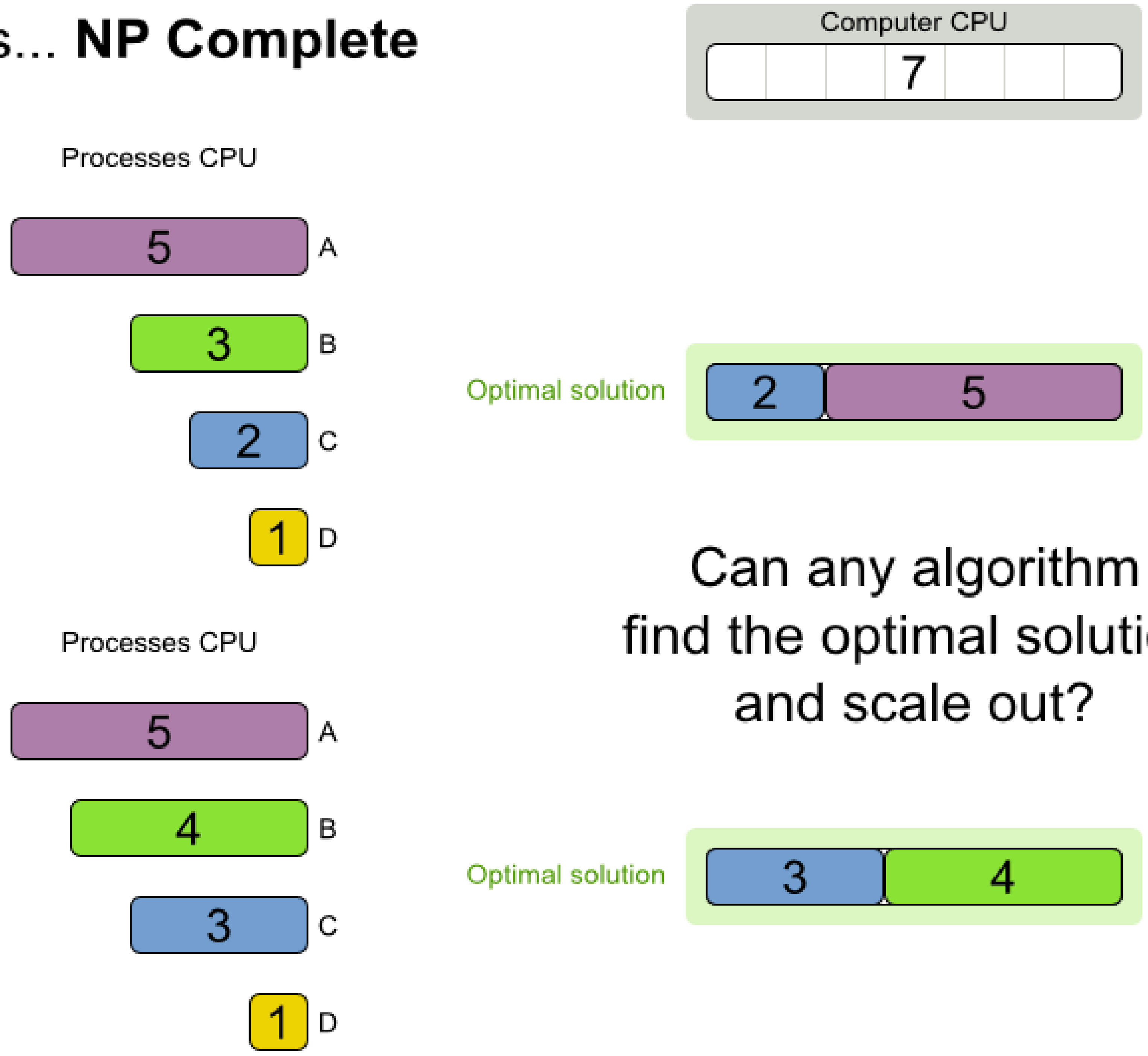


First Fit Decreasing again...



FAIL

This is... **NP Complete**



Can any algorithm find the optimal solution and scale out?

Find optimal solution and scale out for an NP-complete problem?

⇔ Is P = NP?

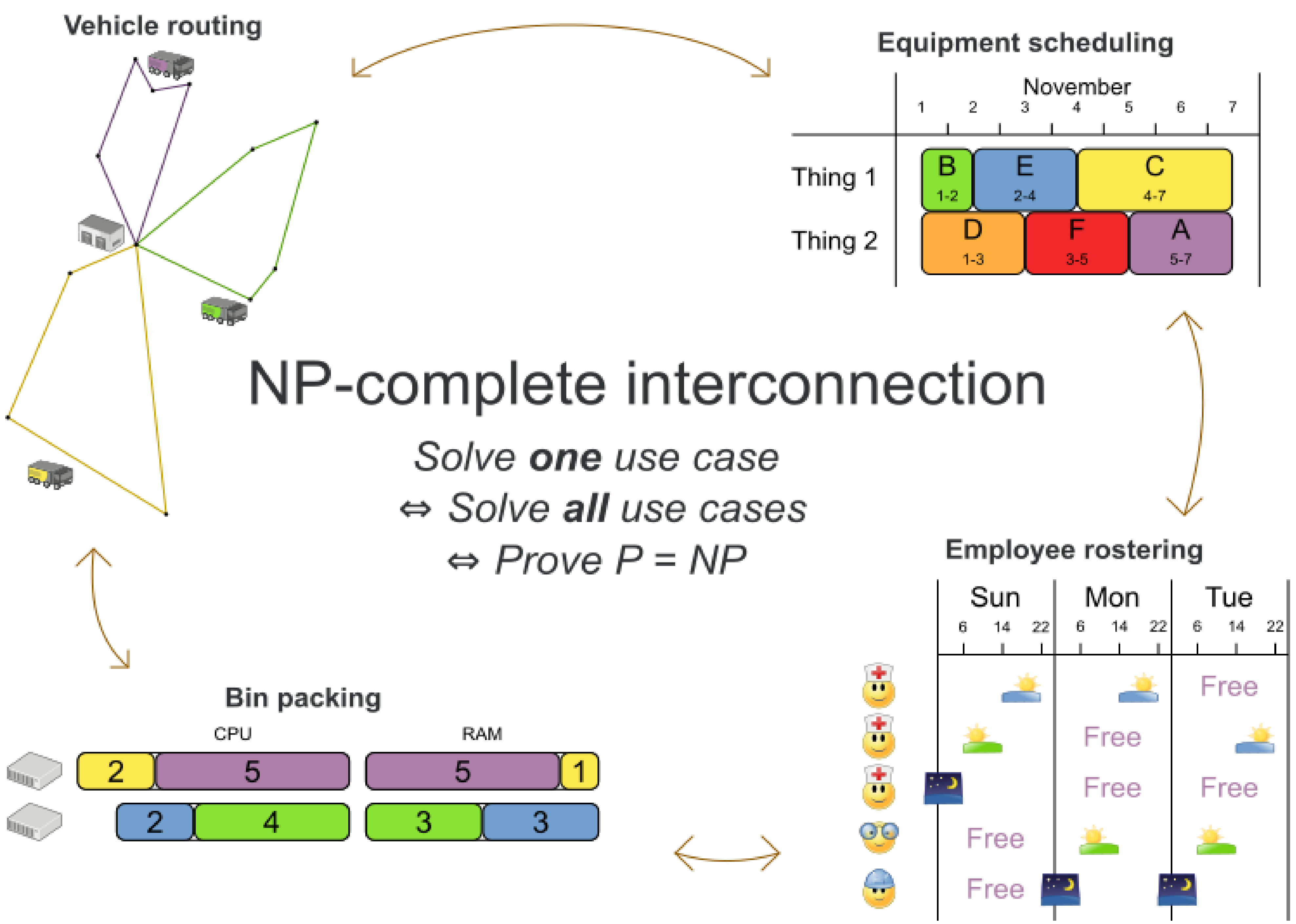
- Unresolved since 1971
- 1 000 000 $ reward since 2000
  - One of the 7 Millennium Problems (http://www.claymath.org/millennium-problems)
- Most believe P ≠ NP
  - ⇔ **Impossible to find optimal solution and scale out**
- 3000+ known NP-complete problems (wikipedia (http://en.wikipedia.org/wiki/List_of_NP-complete_problems))

NP-complete interconnection

*Solve **one** use case*

⇔ *Solve **all** use cases*

⇔ *Prove P = NP*



Use the right tool for the job.

- Insurance rate calculation: decision table
- License plate recognition: neural net
- Employee shift rostering: constraint solver

Don't use a hammer on a screw.

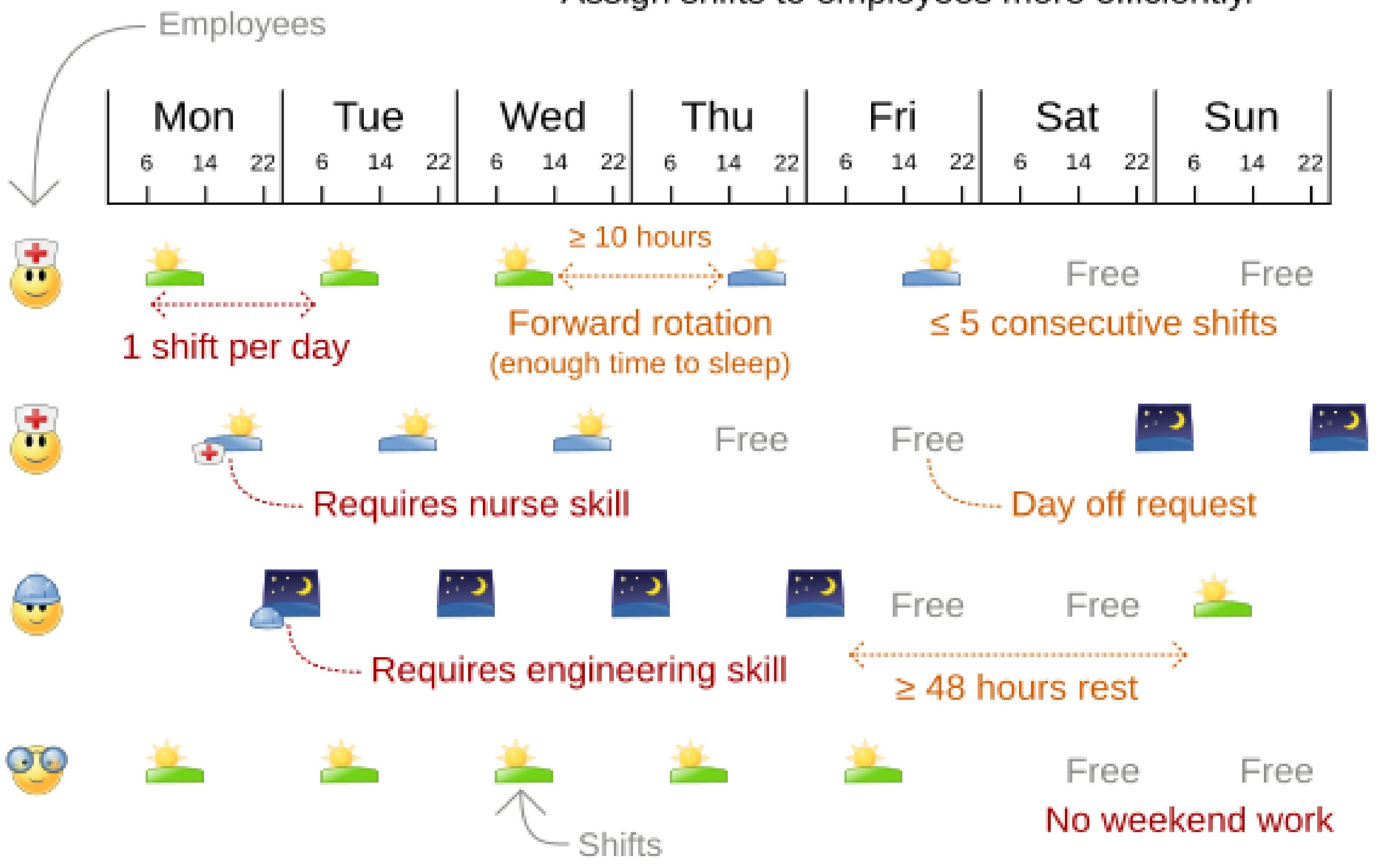
Constraint solver use cases...

- **Agenda scheduling**: doctor appointments, court hearings, maintenance jobs, TV advertisements, ...
- **Educational timetabling**: lectures, exams, conference presentations, ...
- **Task assignment**: affinity/skill matchmaking for tax audits, wage calc, ...
- **Employee shift rostering**: nurses, repairmen, help desk, firemen, ...
- **Vehicle routing**: route trucks, buses, trains, boats, airplanes, ...
- **Bin packing**: fill containers, trucks, ships, storage warehouses, cloud computers nodes, prisons, hospitals, ...
- **Job shop scheduling**: assembly lines for cars, furniture, books, ...
- **Cutting stock**: minimize waste while cutting paper, steel, carpet, ...
- **Sport scheduling**: football/baseball league, tennis court utilization, ...
- **Financial optimization**: investment portfolio balance, risk spreading, ...

Employee shift rostering

Employee rostering

Assign shifts to employees more efficiently.



Users

Hospitals

Courts of Justice

Call centers

Police and fire department

NurseRostering benchmark	Average	Min/Max	# datasets	Biggest dataset
Employee well-being	+53%	+19% +85%	26	752 assignments 50 employees
OptaPlanner versus traditional algorithm with domain knowledge				5 mins Tabu Search vs First Fit Decreasing

Don't believe us? Run our open benchmarks yourself: https://www.optaplanner.org/code/benchmarks.html

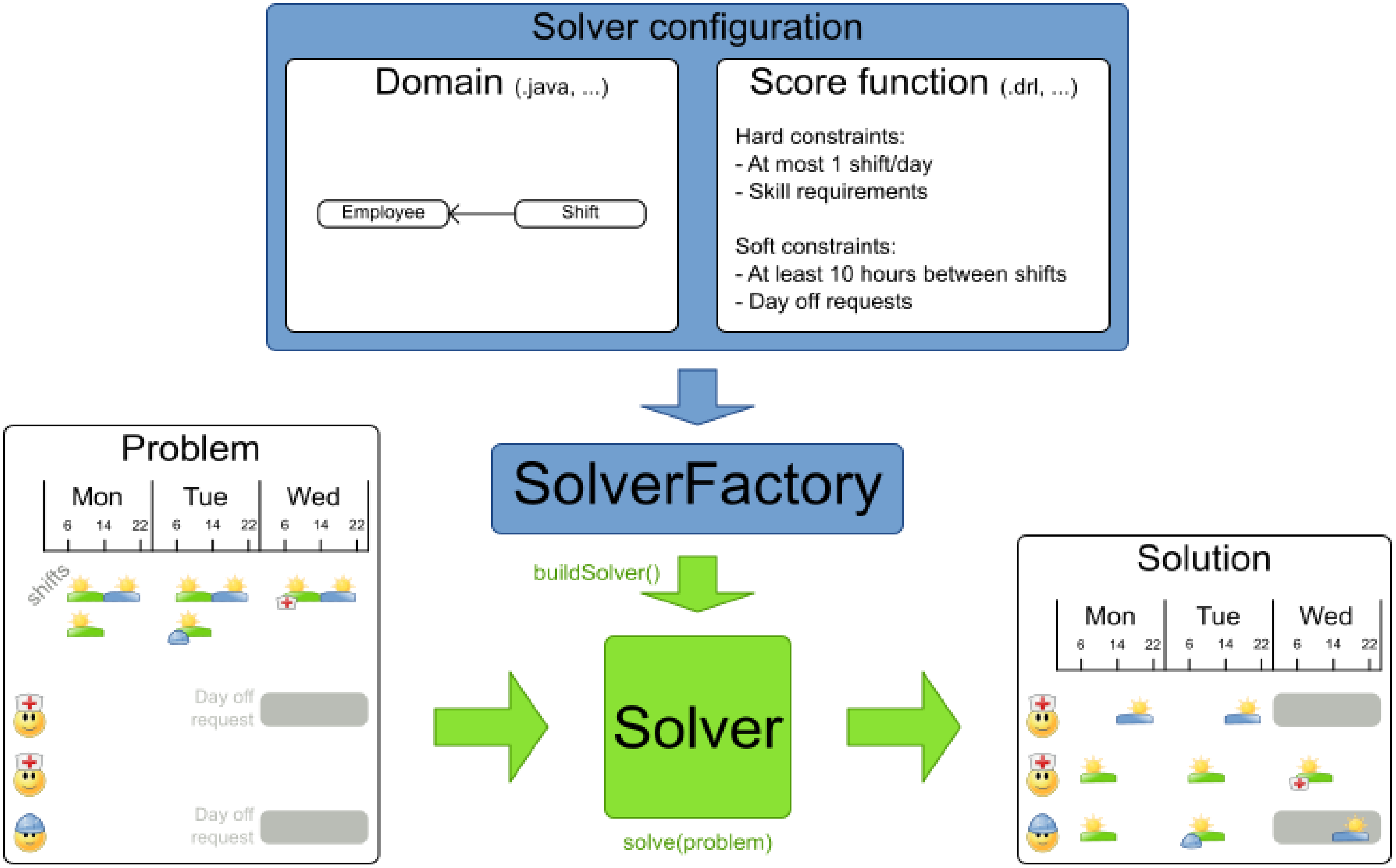
What is constraint solving?

Implementation

1. Define domain
2. Define constraints
3. Solve

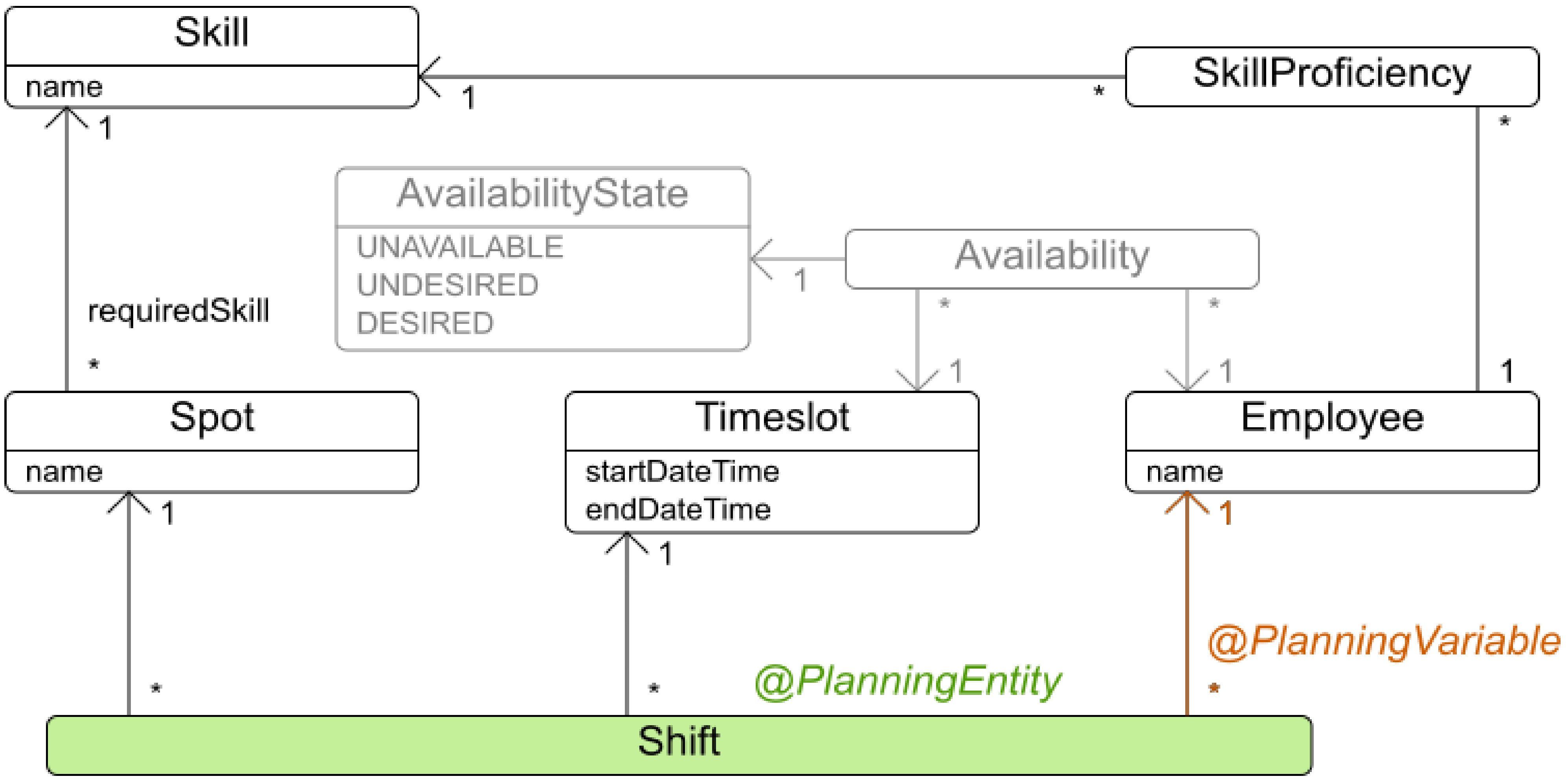
Input/Output overview employee rostering

Use 1 SolverFactory per application and 1 Solver per dataset.



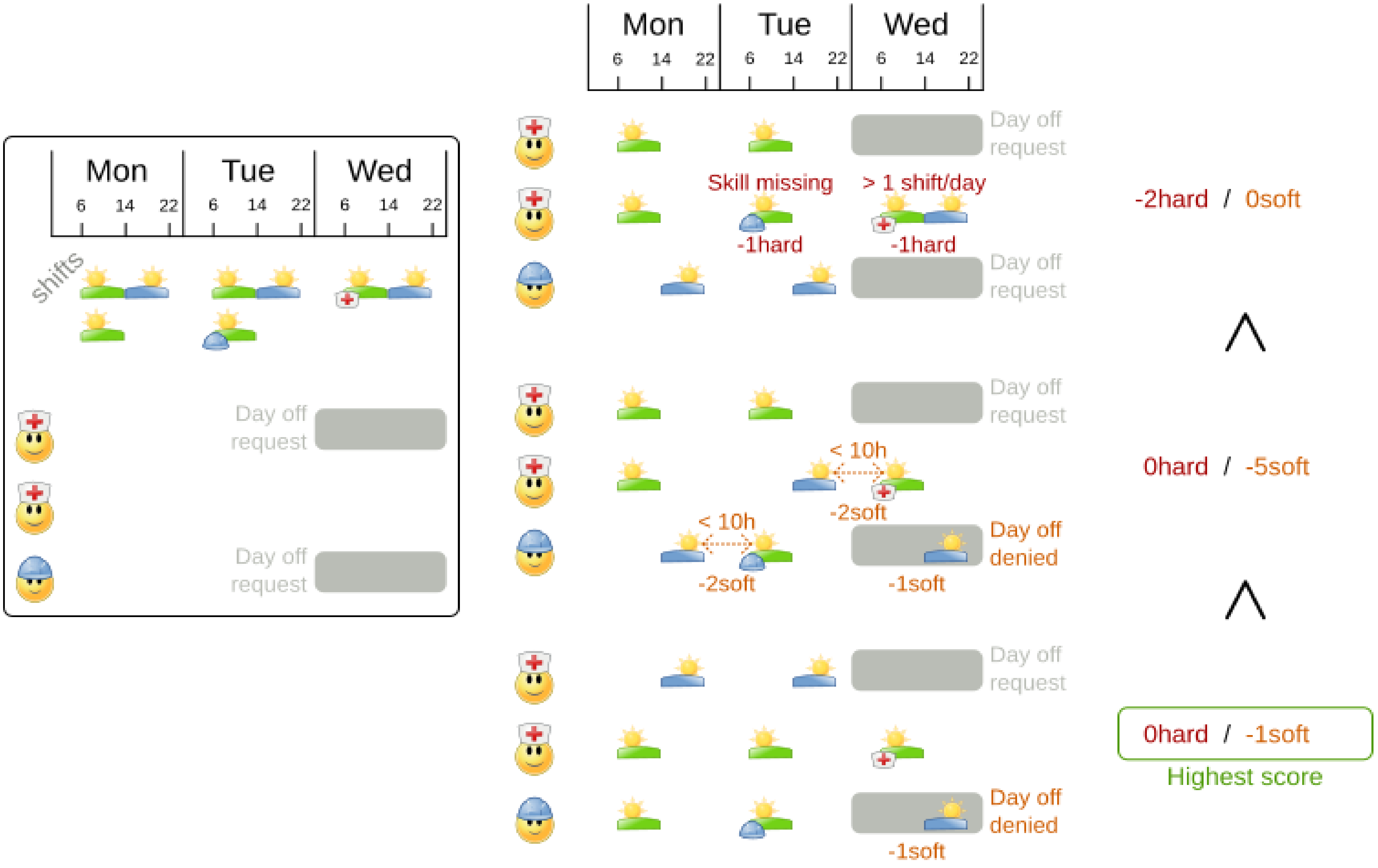
Define domain

Employee rostering class diagram



Score Comparison Employee Rostering

Hard constraints always outweight soft constraints.

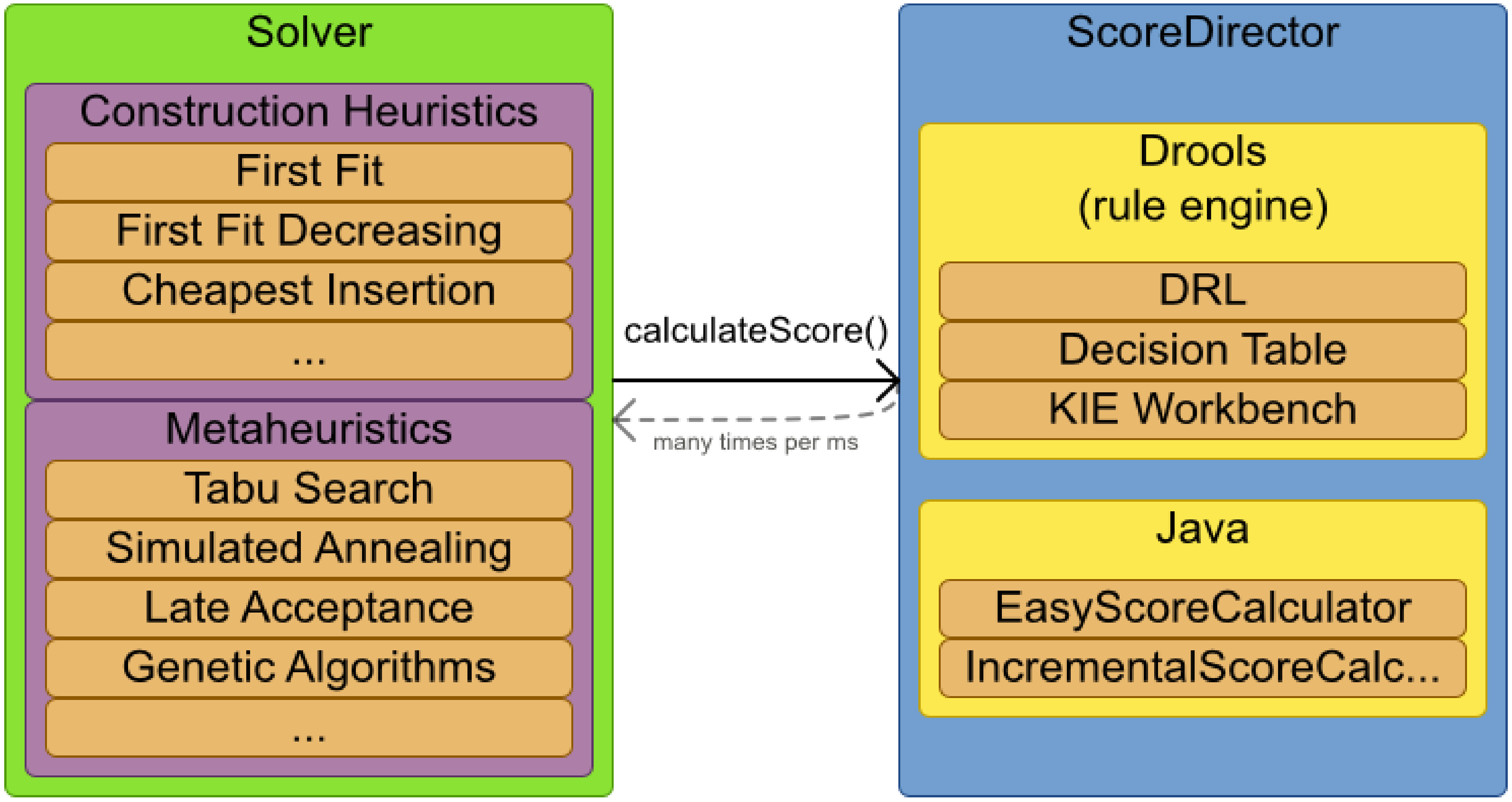


Score calculation

- Easy Java (slow)
- Incremental Java (painful)
- Drools DRL (also incremental)

Architecture overview

The Solver wades through the search space of solutions efficiently.
The ScoreDirector calculates the score of every solution under evaluation.



Find a better solution

Calculate the score
of a solution

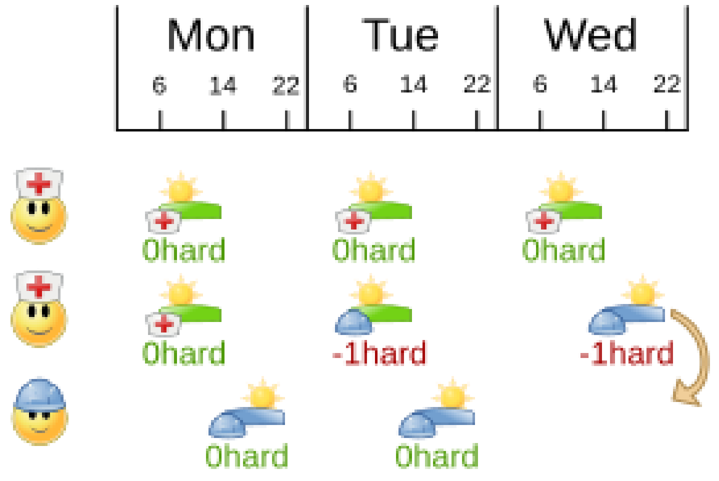
Required skill constraint (easy Java)

```
public class MyScoreCalculator
    implements EasyScoreCalculator<Roster> {

  public Score calculateScore(Roster roster) {
    int hardScore = 0;
    for (Shift shift : roster.getShiftList()) {
      Skill requiredSkill = shift.getSpot().getRequiredSkill();
      if (shift.getEmployee() != null
          // Employee lacks required skill
          && !shift.getEmployee().hasSkill(requiredSkill)) {
        // Lower hard score
        hardScore--;
      }
    }
    ...
    return HardSoftScore.valueOf(hardScore, softScore);
  }
```

Incremental score calculation

Calcuting delta's is much faster than calculating the entire's solution's score.



Check every shift:

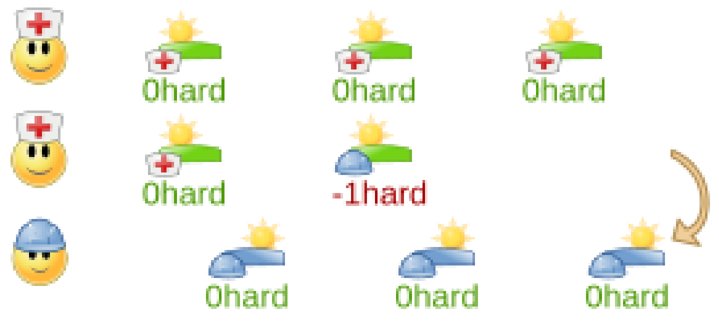
0 + 0 + 0 + 0 - 1 - 1 + 0 + 0

Required skill score: -2hard

BigO for n shifts

Constraint	From scratch	Incremental
Required skill	O(n)	O(1)
At most 1 shift/day	O(n²)	O(n)
...	...	...

Calculation from scratch (easy java)

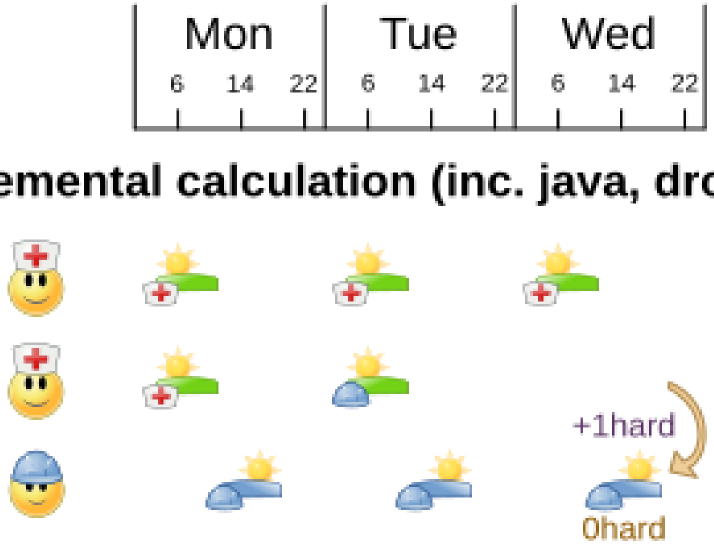


Check every shift again:

0 + 0 + 0 + 0 - 1 + 0 + 0 + 0

Required skill score: -1hard

Incremental calculation (inc. java, drools)



Check one shift (old & new)

-2 + 1 - 0

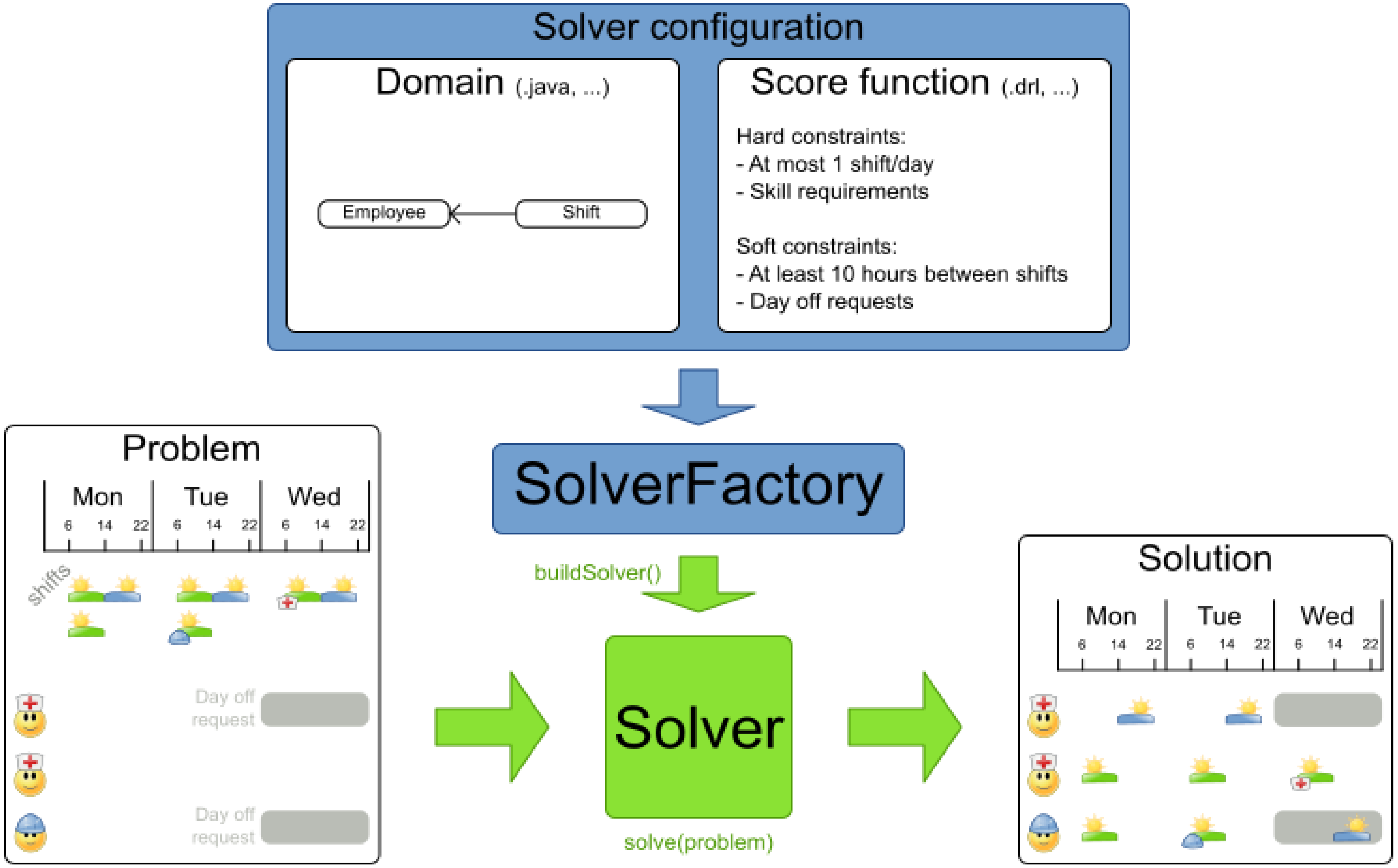
Required skill score: -1hard

Required skill constraint (Drools DRL)

```
rule "Required skill"
  when
    Shift(
        getEmployee() != null,
        // Employee lacks required skill
        !getEmployee().hasSkill(getSpot().getRequiredSkill()))
  then
    // Lower hard score
    scoreHolder.addHardConstraintMatch(kcontext, -1);
end
```

Input/Output overview employee rostering

Use 1 SolverFactory per application and 1 Solver per dataset.

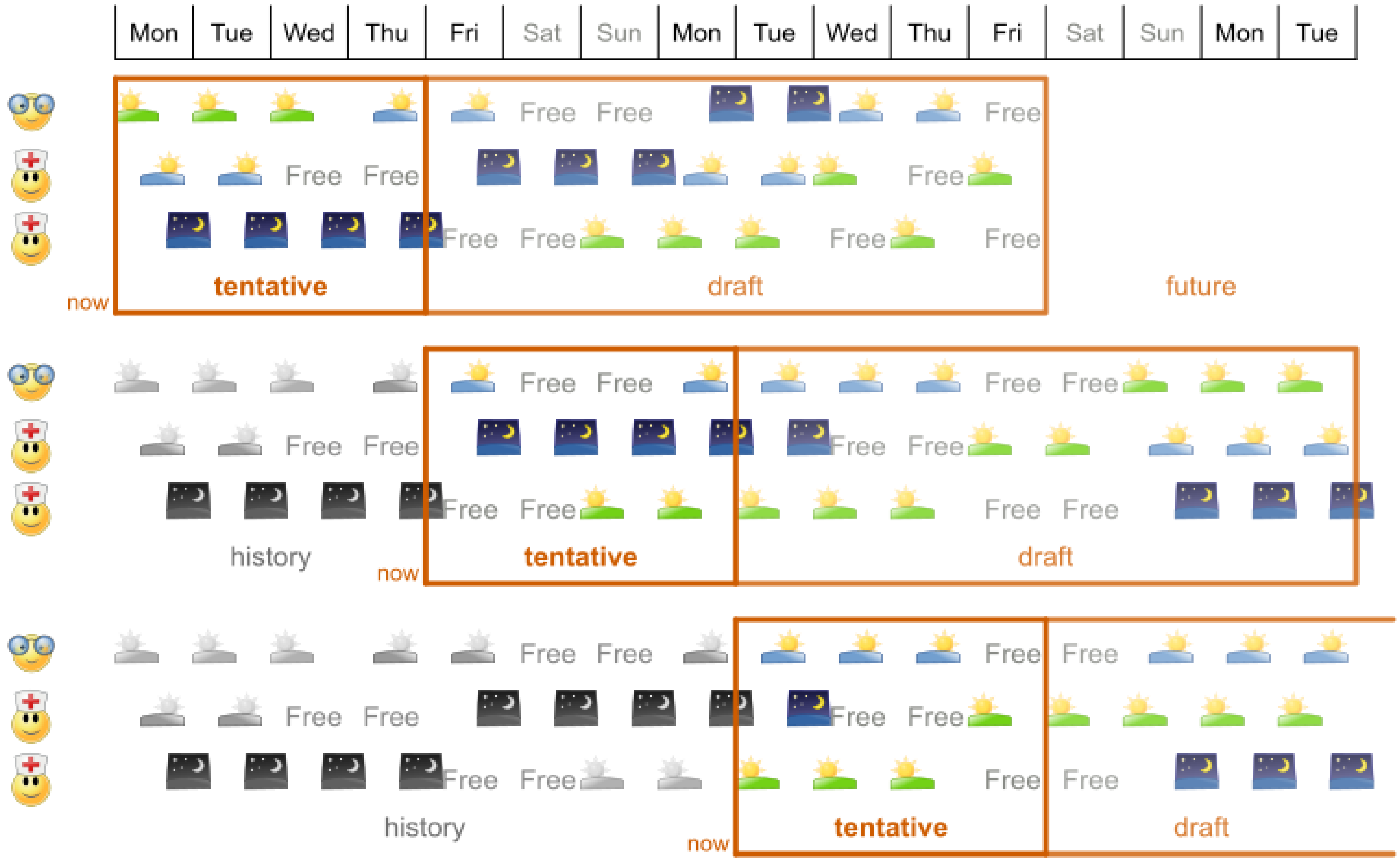


When do we solve?

- Publish schedule weeks in advance
  - Affects family/social lives
- Ad hoc changes
  - Sick employees
  - Shift changes

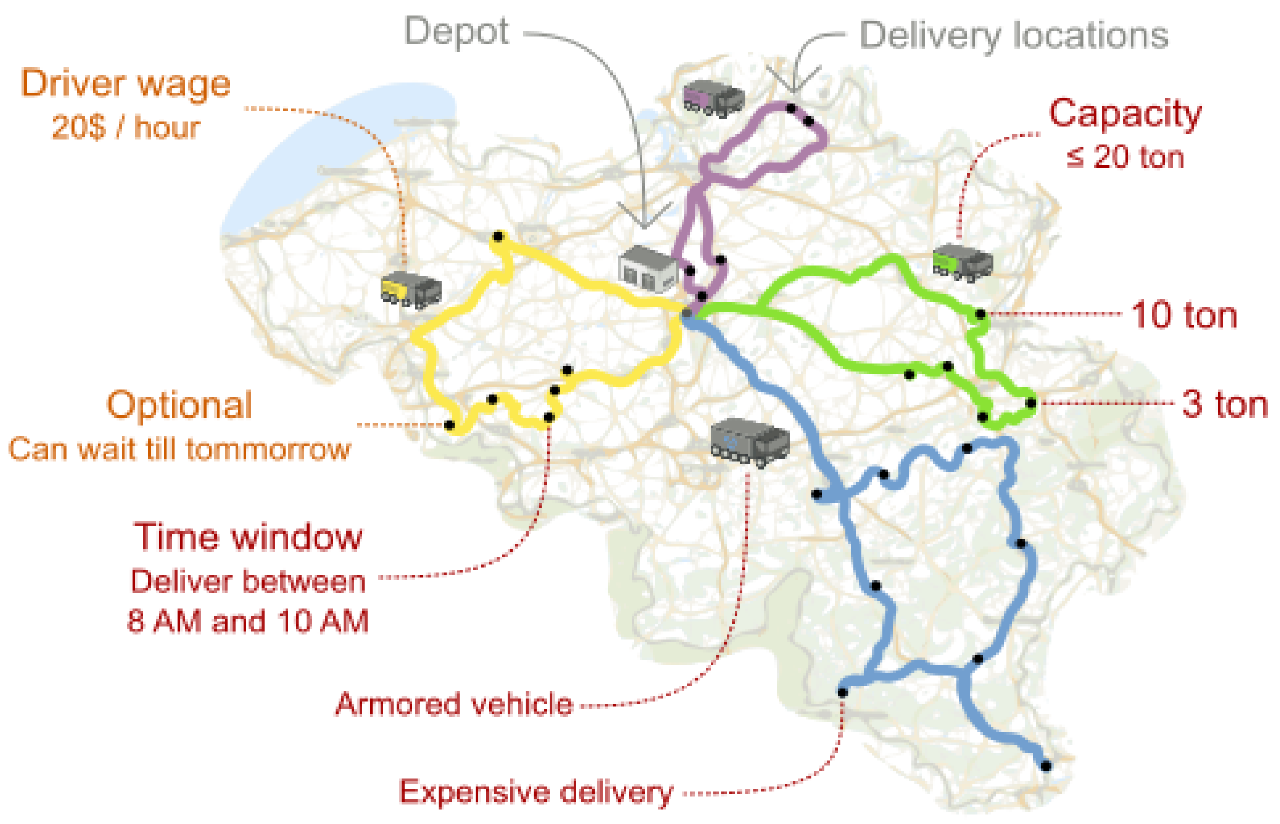
Continuous planning

Replan at the start of every period. Plan 3 periods, but only share the first period.



Vehicle routing

Assign the delivery order of vehicles more efficiently.



Users

Supermarkets & retail stores

Freight transportation

Buses, taxi's & airlines

Technicians on the road

VehicleRouting benchmark (Belgium datasets)	Average	Min/Max	# datasets	Biggest dataset
Driving time	-15%	-9% -18%	5	2750 deliveries 55 vehicles

OptaPlanner versus traditional algorithm with domain knowledge

5 mins Late Acceptance Nearby vs First Fit Decreasing

Don't believe us? Run our open benchmarks yourself: http://www.optaplanner.org/code/benchmarks.html

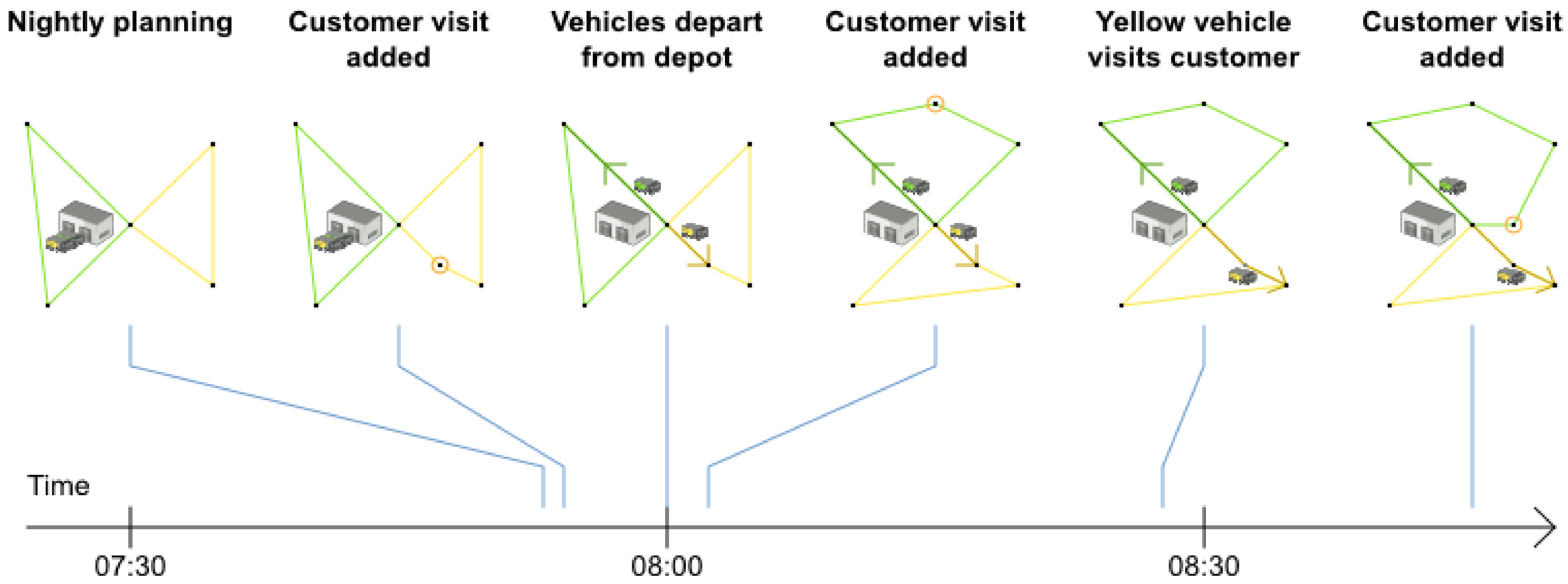
Vehicle Routing Problem

Real-time planning

Warm starts to solve in milliseconds

Real-time planning

When the problem changes in real-time, the plan is adjusted in real-time.



Vehicle Routing Problem

Q & A

OptaPlanner	www.optaplanner.org (https://www.optaplanner.org)
Feedback	@GeoffreyDeSmet (https://twitter.com/GeoffreyDeSmet)