Kickstarting Your OptaPlanner Project

Patterns and Common Practices

Duncan Doyle
Product Manager

What is a Planning Problem?

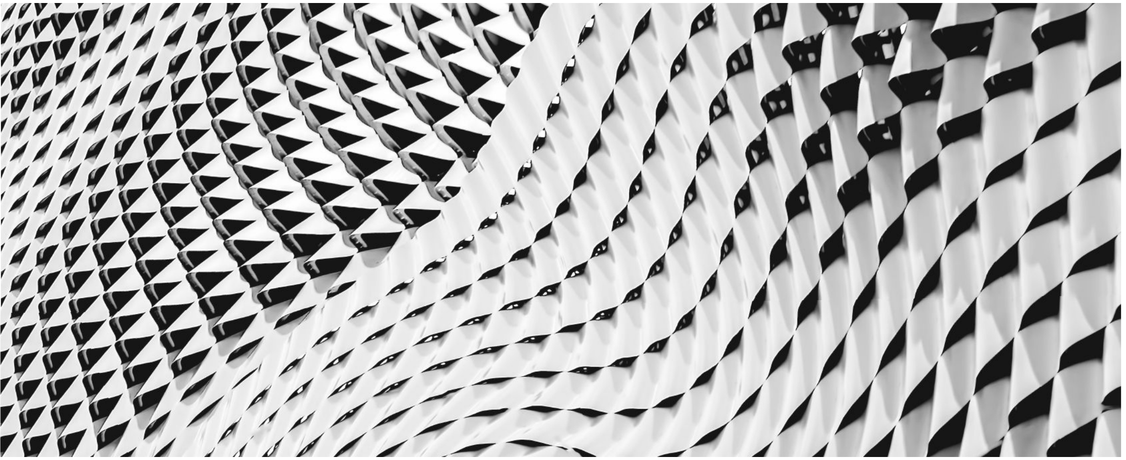
Optimize Goals

With limited Resources

Under Constraints

Source:
Insert source data here
Insert source data here
Insert source data here

Value Proposition



Vehicle Routing

Assign the delivery order more efficiently

Goals

Minimize fuel consumption
Minimize driving time
Minimize required vehicles

Resources

Vehicles (capacity, fuel)
Deliveries (location, packages)

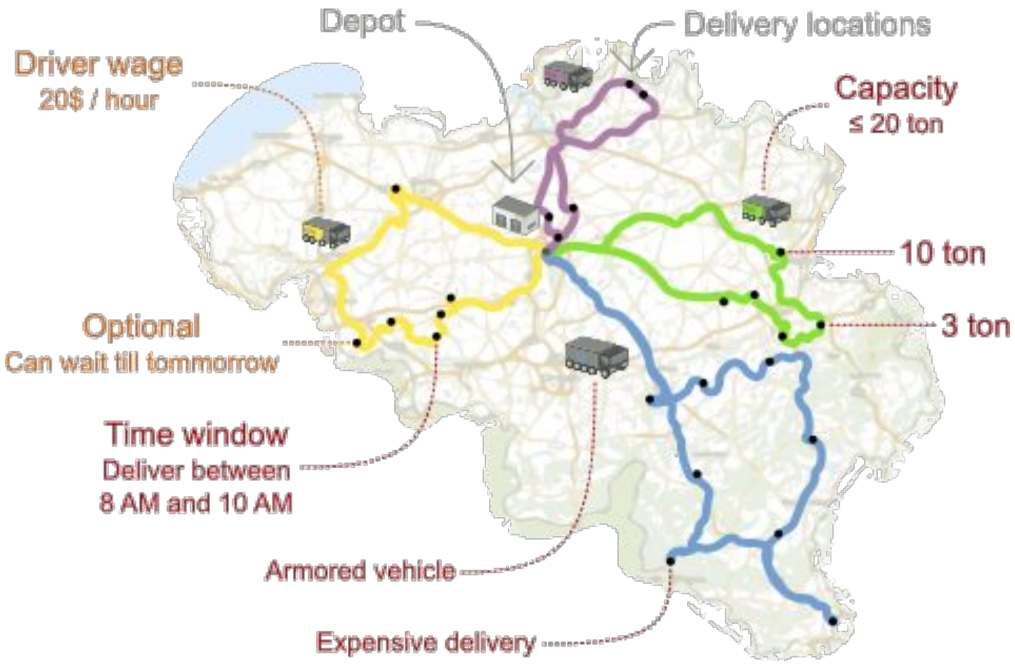
Constraints

Max 8hrs consecutive driving
Arrive before due time
Max vehicle capacity

Source:
Insert source data here
Insert source data here
Insert source data here

Vehicle Routing

Assign the delivery order more efficiently



Business Value

-15% Driving Time

(based on real benchmark versus traditional algorithms, Belgium datasets)

Users:

- Supermarket & Retail Stores
- Freight Transportation
- Buses, Taxis & Airlines
- Technicians on the road

Source:
Insert source data here
Insert source data here
Insert source data here

Employee Rostering

Assign shifts to employee more efficiently

Goals

Increase Employee well-being

Resources

Nurses
Security Guards
...

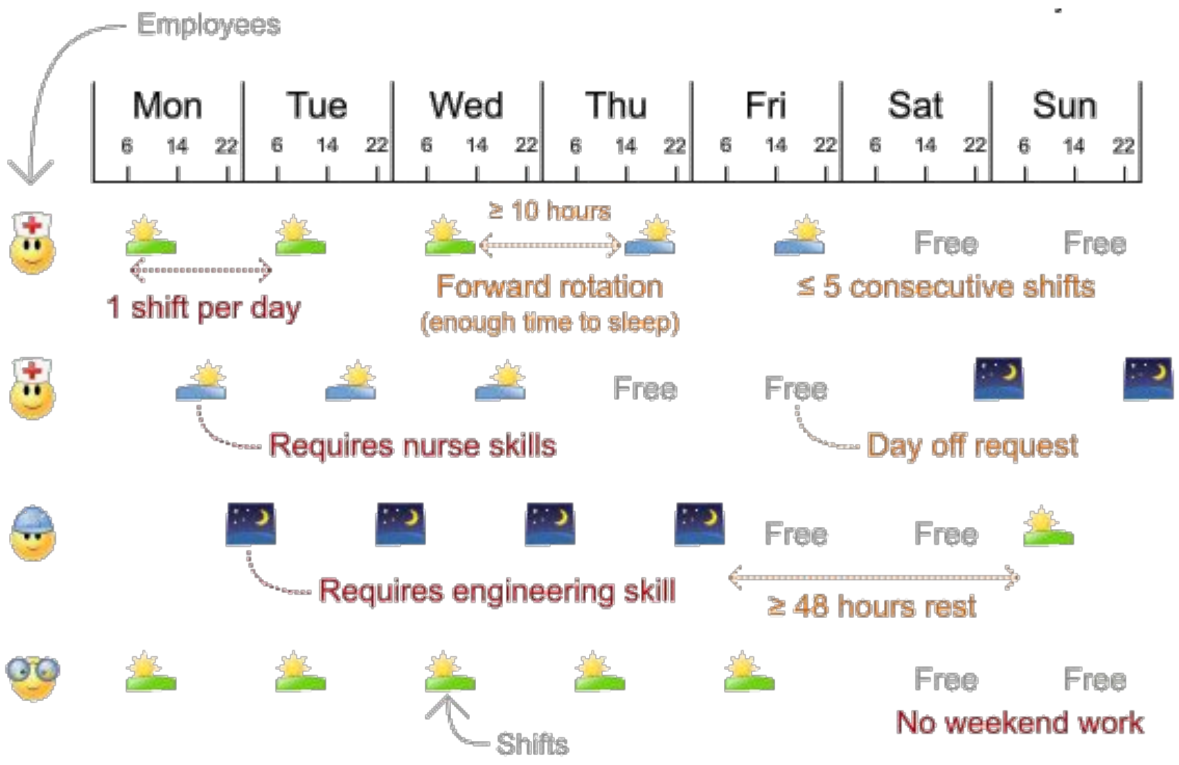
Constraints

Work 1 shift per day
Max consecutive working days
Requested days off

Source:
Insert source data here
Insert source data here
Insert source data here

Employee Rostering

Assign shifts to employee more efficiently



Business Value

+53% Employee well-being

(average on real benchmark versus traditional algorithms, Nurses case)

Users:

- Hospitals
- Call Centers
- Pole and Fire Departments
- Court of Justice

Source:
Insert source data here
Insert source data here
Insert source data here

Cloud Optimization

Assign processes to machine more efficiently

Goals

Minimize costs
Optimize resource utilisation

Resources

Computers
(CPU, Memory, Bandwidth)

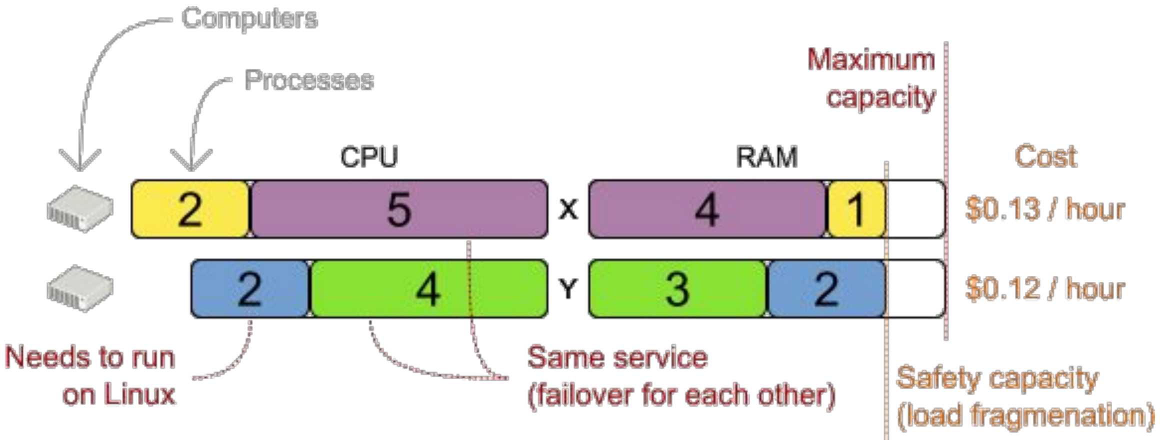
Constraints

Process requires {x} CPU
Process requires {y} Memory
Process requires {z} Bandwidth

Source:
Insert source data here
Insert source data here
Insert source data here

Cloud Optimization

Assign processes to machine more efficiently



Business Value

-18% Cloud Hosting Costs (avg)

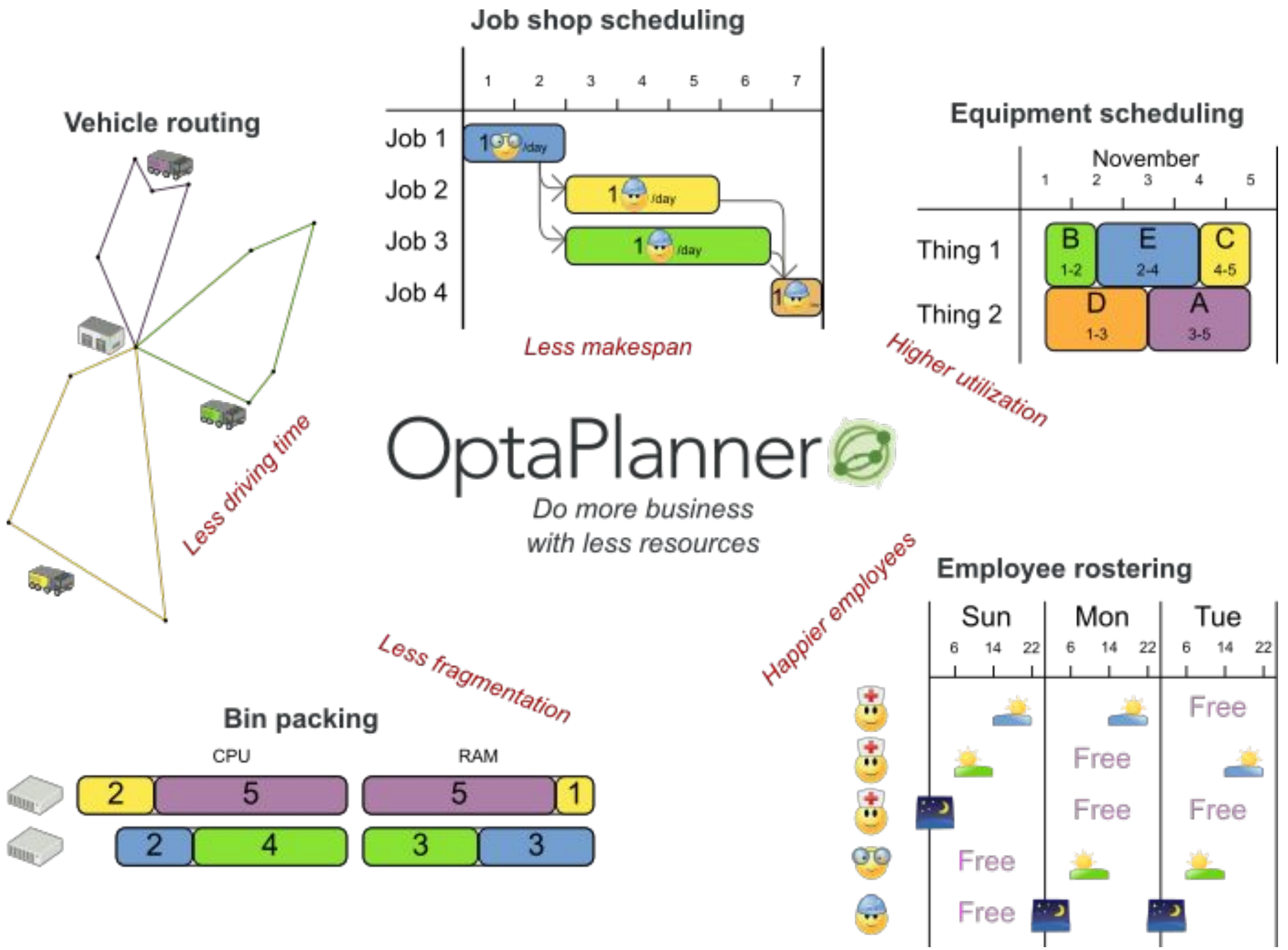
-63% Hardware Congestion (avg)

Users:

- Datacenters automation

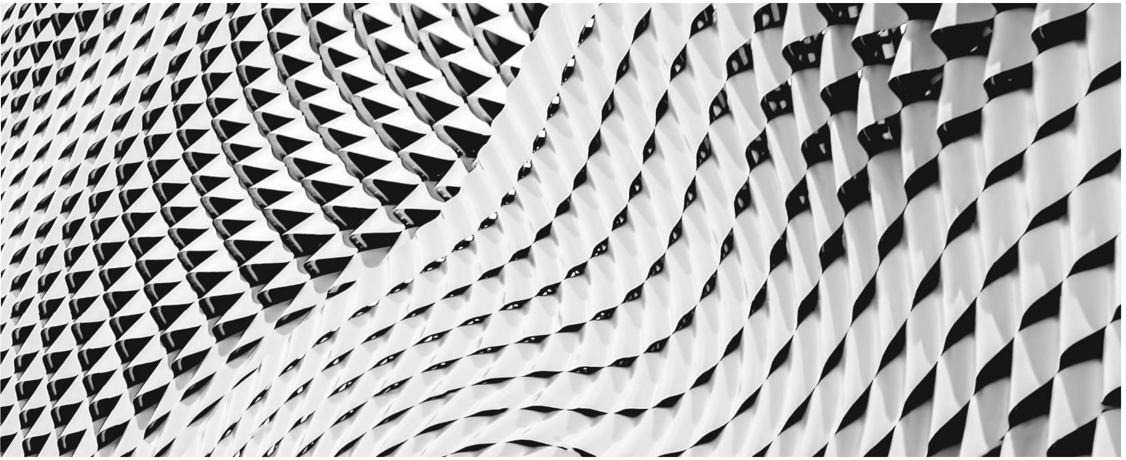
Source:
Insert source data here
Insert source data here
Insert source data here

Planning Problems are everywhere



Source:
Insert source data here
Insert source data here
Insert source data here

Kickstarting Your Project



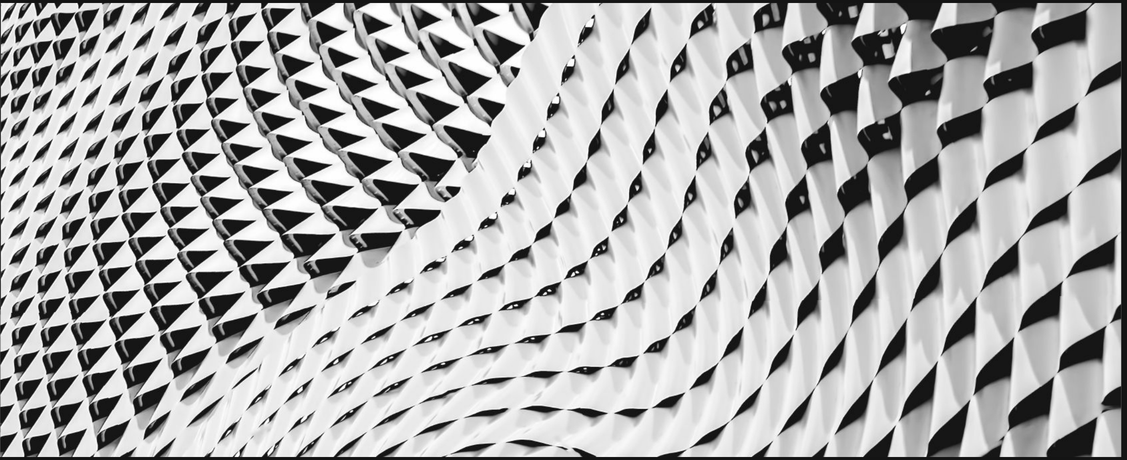
Patterns & Common Practices

Making your OptaPlanner Successful

- Domain Modelling: Understand your Problem: Solution Space
- Benchmarker
- Score Calculation Types
- Environment Modes
- Keep the User in Control
- Explain the Score
- Reproducibilty

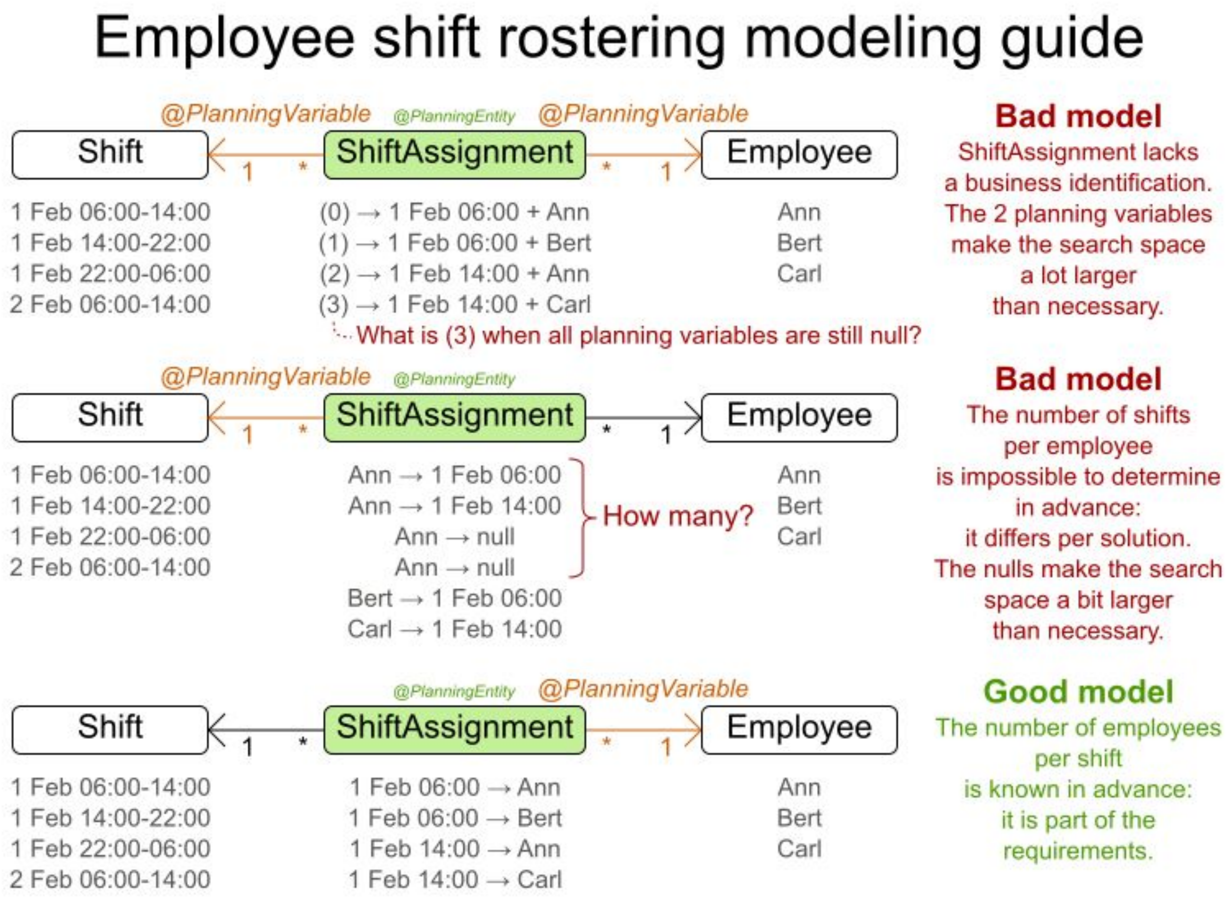
Source:
Insert source data here
Insert source data here
Insert source data here

Domain Modelling



Domain Modelling

Employee Shift Rostering example



What is (3) when all planning variables are still null?

Bad model

ShiftAssignment lacks a business identification. The 2 planning variables make the search space a lot larger than necessary.

Bad model

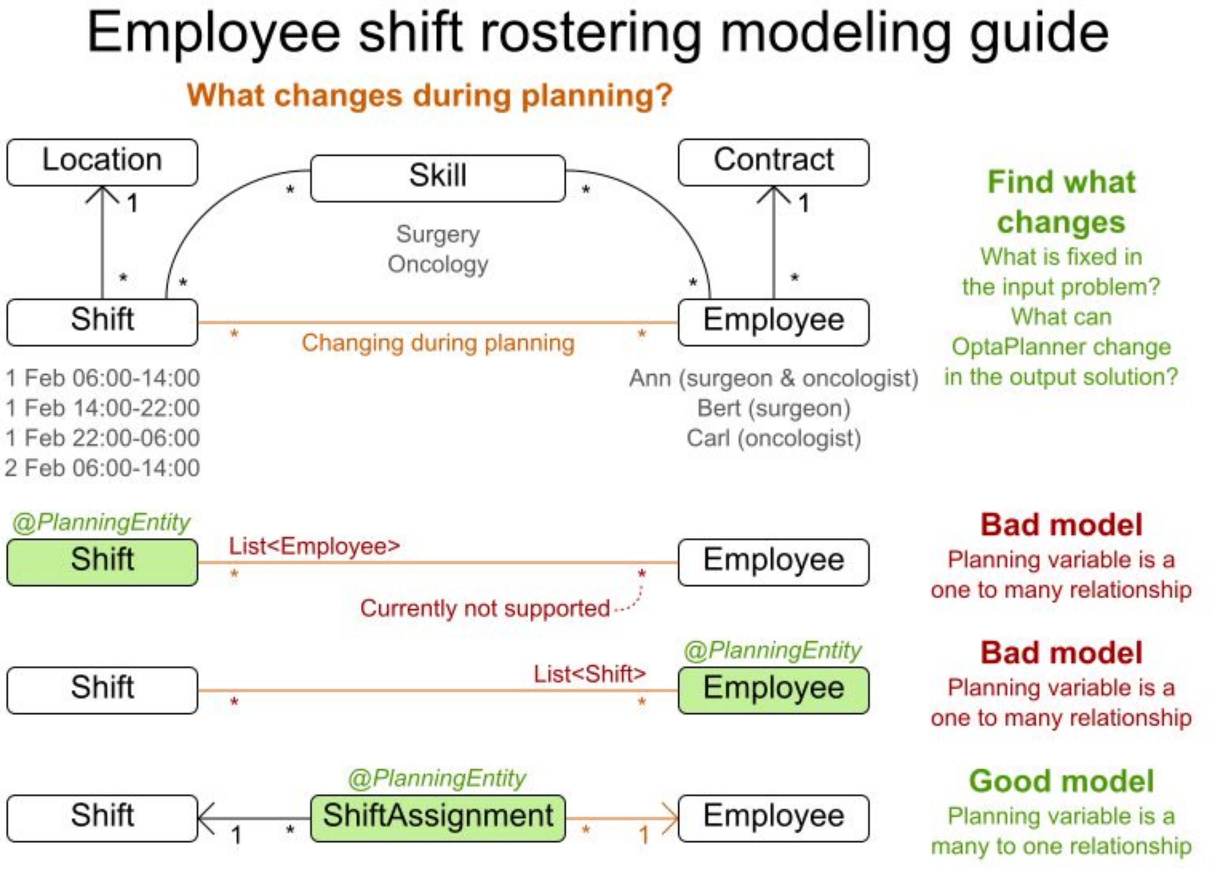
The number of shifts per employee is impossible to determine in advance: it differs per solution. The nulls make the search space a bit larger than necessary.

Good model

The number of employees per shift is known in advance: it is part of the requirements.

Source:
Insert source data here
Insert source data here
Insert source data here

Domain Modelling
Employee Shift Rostering example



Employee shift rostering modeling guide

What changes during planning?

Find what changes
What is fixed in the input problem? What can OptaPlanner change in the output solution?

Bad model
Planning variable is a one to many relationship

Bad model
Planning variable is a one to many relationship

Good model
Planning variable is a many to one relationship

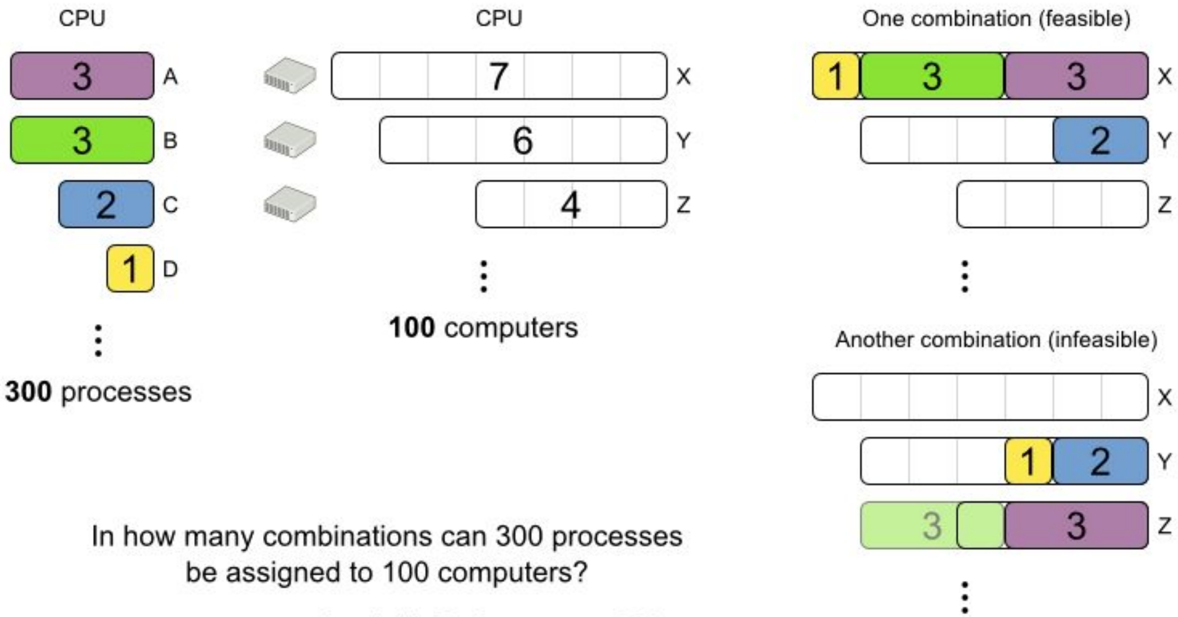
Source:
Insert source data here
Insert source data here
Insert source data here

Domain Modelling

Solution Space

What is the size of the search space?

How big is the haystack?



In how many combinations can 300 processes be assigned to 100 computers?

$$|\text{valueSet}|^{|\text{variableSet}|} = \mathbf{100^{300}}$$
$$= 10^{600} = 1000000000000000000000\ldots$$

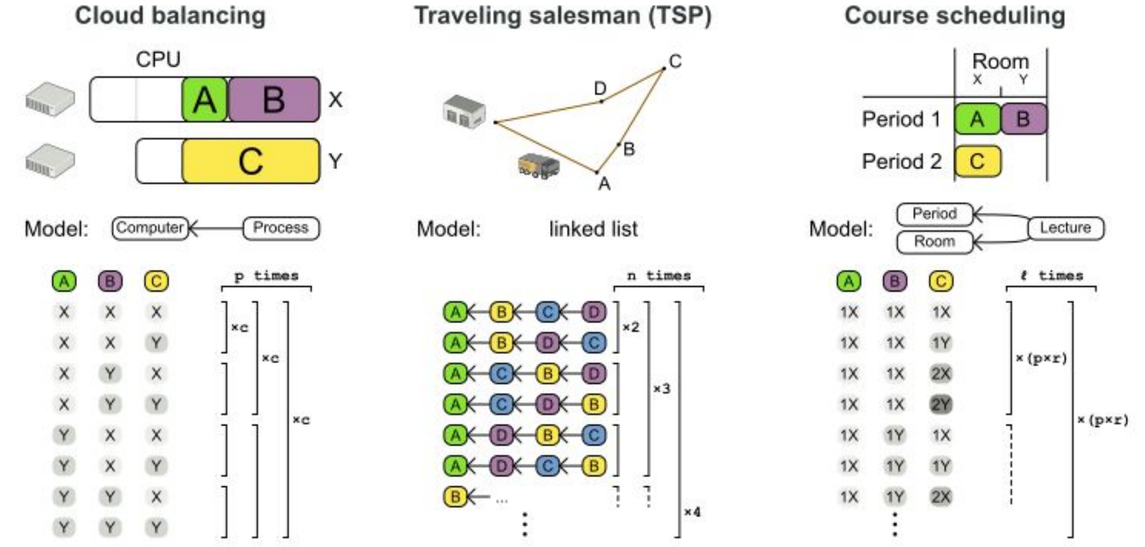
Source:
Insert source data here
Insert source data here
Insert source data here

Domain Modelling

Solution Space

Calculate the size of the search space

Given a Solution model, how many different combinations can it represent?



Search space: c^P

# computers	# processes	search space
2	3	8
100	300	10^{600}
200	600	10^{1380}
400	1200	10^{6967}

Search space: $n!$

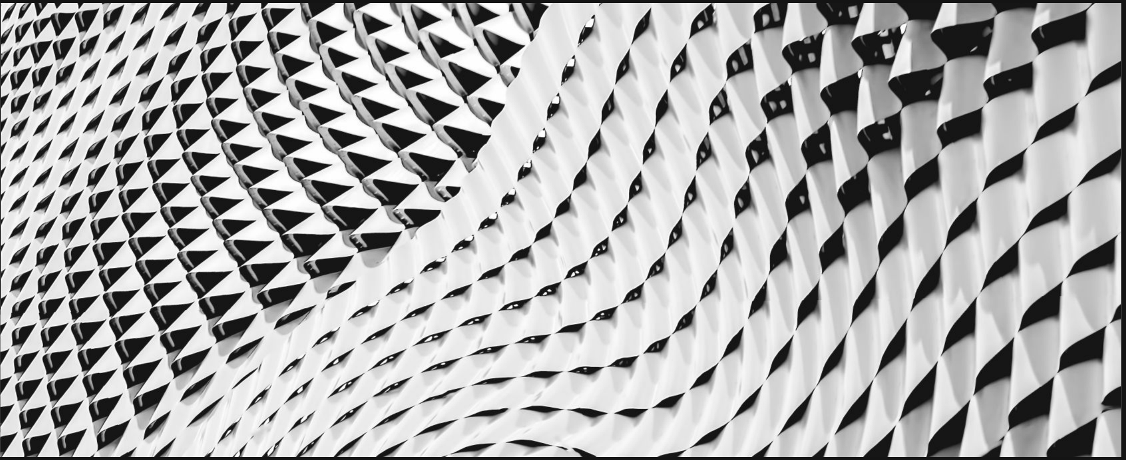
# customers	search space
4	24
100	10^{157}
1000	10^{2567}
10000	10^{35659}

Search space: $(p\times r)^{\ell}$

# periods	# rooms	# lectures	space
2	2	3	64
36	6	100	10^{233}
36	18	400	10^{1124}
36	36	800	10^{2490}

Source:
Insert source data here
Insert source data here
Insert source data here

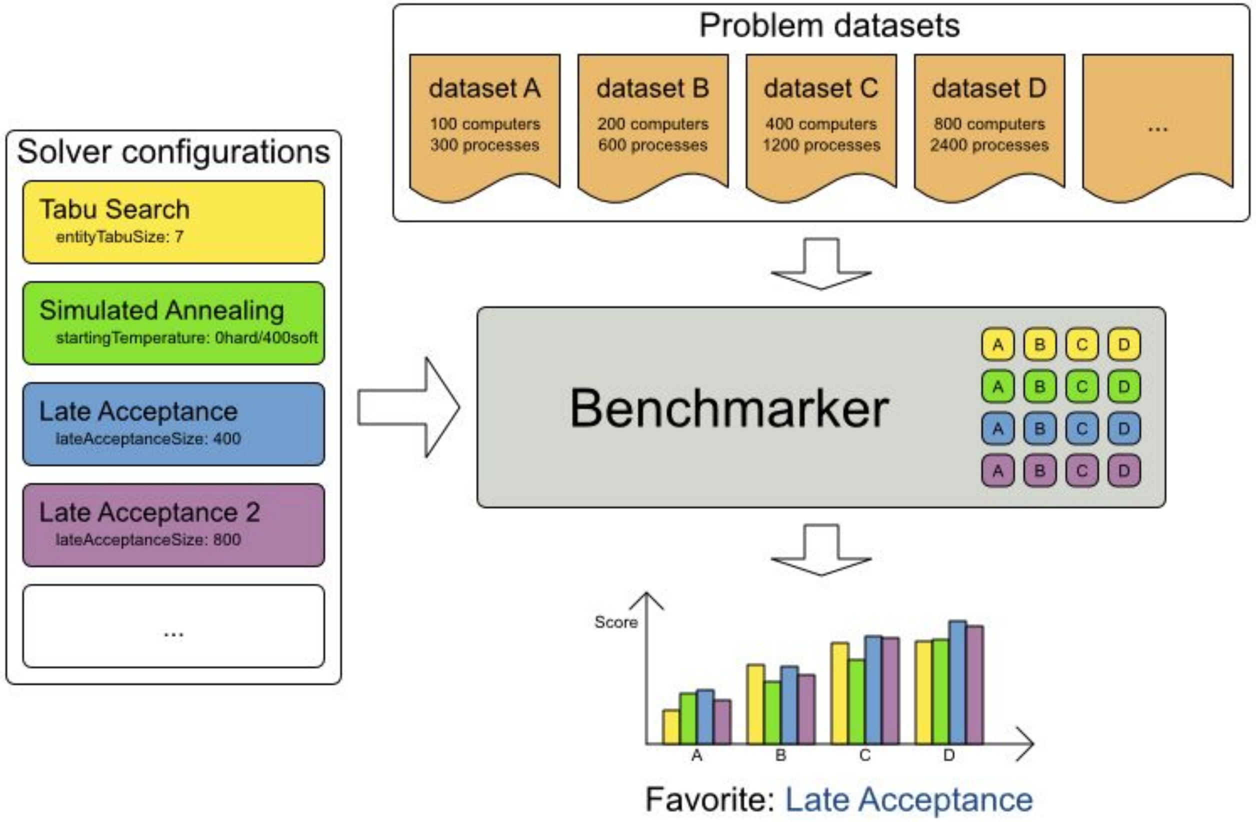
Benchmark



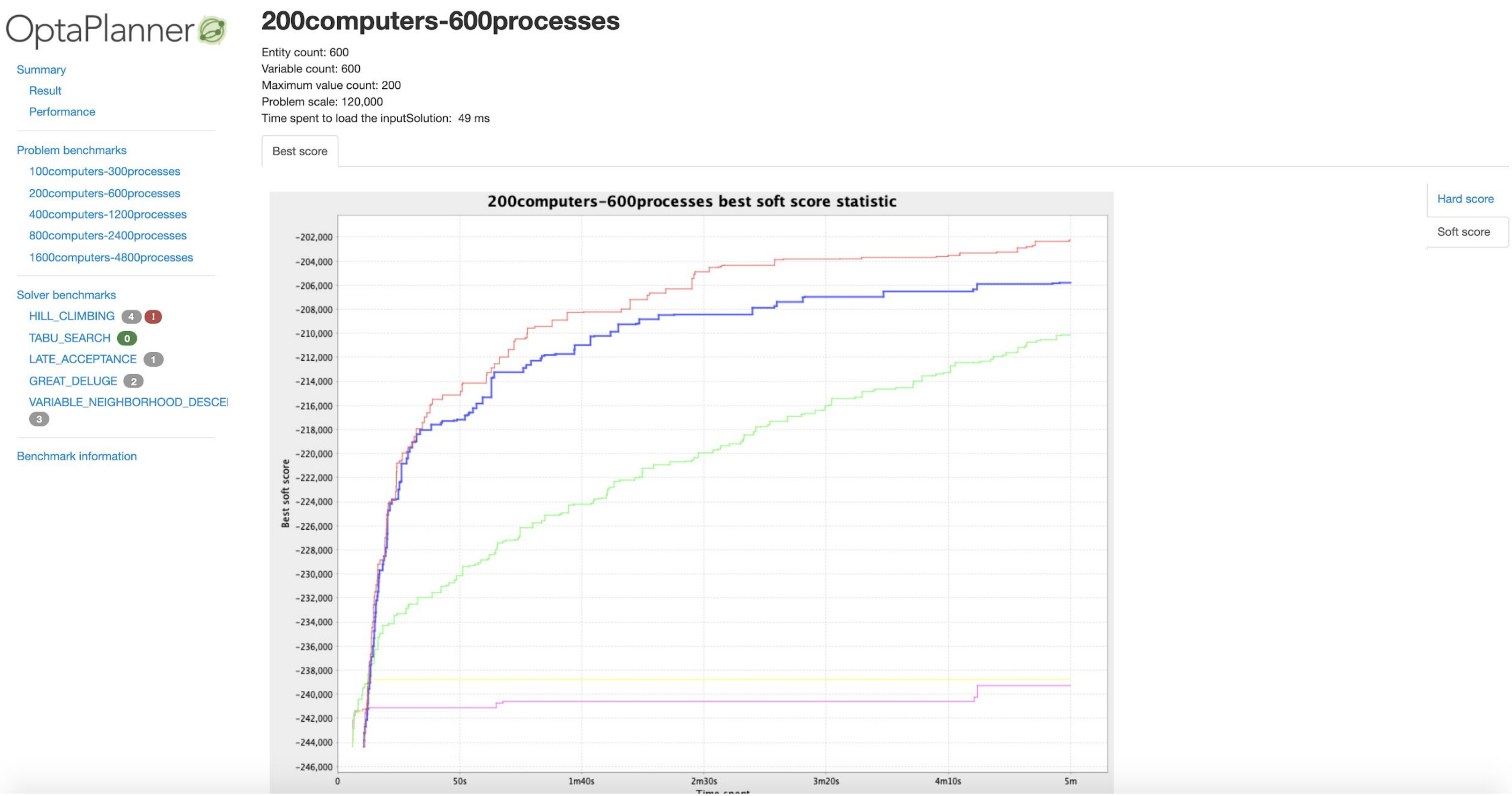
Benchmarker

Benchmark overview

What optimization algorithm should we configure in production? The Benchmarker will tell us.



Benchmarker



OptaPlanner

Summary
- Result
- Performance

Problem benchmarks
- 100computers-300processes
- 200computers-600processes
- 400computers-1200processes
- 800computers-2400processes
- 1600computers-4800processes

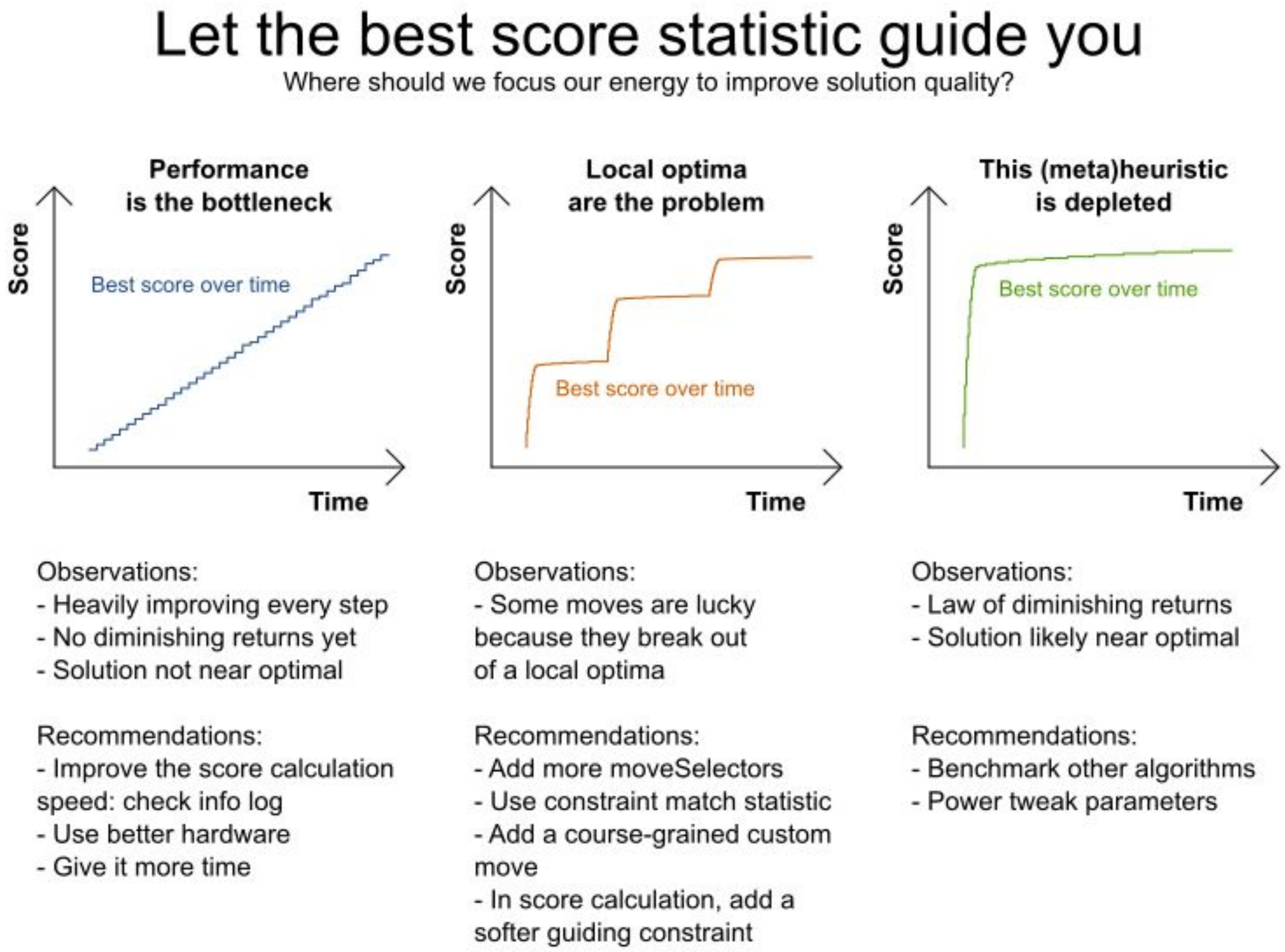
Solver benchmarks
- HILL_CLIMBING 4 !
- TABU_SEARCH 0
- LATE_ACCEPTANCE 1
- GREAT_DELUGE 2
- VARIABLE_NEIGHBORHOOD_DESCE 3

Benchmark information

200computers-600processes

Entity count: 600
Variable count: 600
Maximum value count: 200
Problem scale: 120,000
Time spent to load the inputSolution: 49 ms

Benchmarker



Let the best score statistic guide you

Where should we focus our energy to improve solution quality?

Performance is the bottleneck

Observations:
- Heavily improving every step
- No diminishing returns yet
- Solution not near optimal

Recommendations:
- Improve the score calculation speed: check info log
- Use better hardware
- Give it more time

Local optima are the problem

Observations:
- Some moves are lucky because they break out of a local optima

Recommendations:
- Add more moveSelectors
- Use constraint match statistic
- Add a course-grained custom move
- In score calculation, add a softer guiding constraint

This (meta)heuristic is depleted

Observations:
- Law of diminishing returns
- Solution likely near optimal

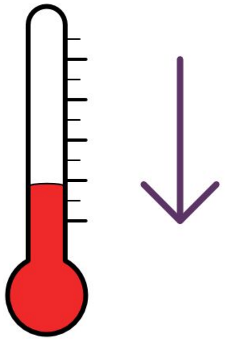
Recommendations:
- Benchmark other algorithms
- Power tweak parameters

Source:
Insert source data here
Insert source data here
Insert source data here

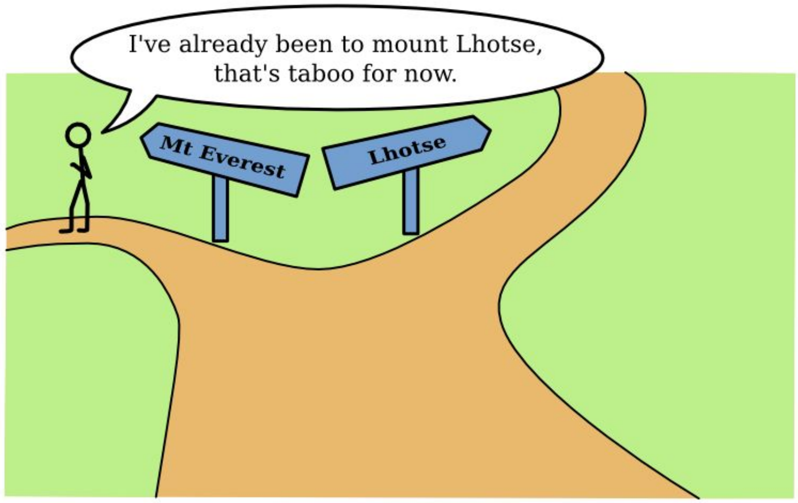
Benchmarker

Which Algorithm Should I Choose?

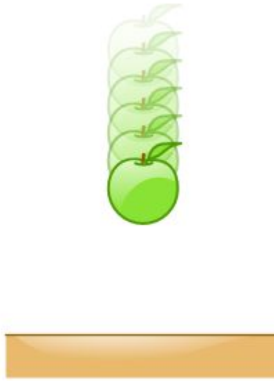
Simulated Annealing



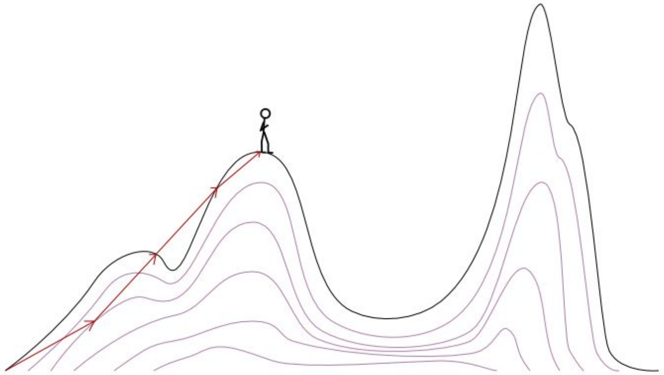
Tabu Search



Late acceptance

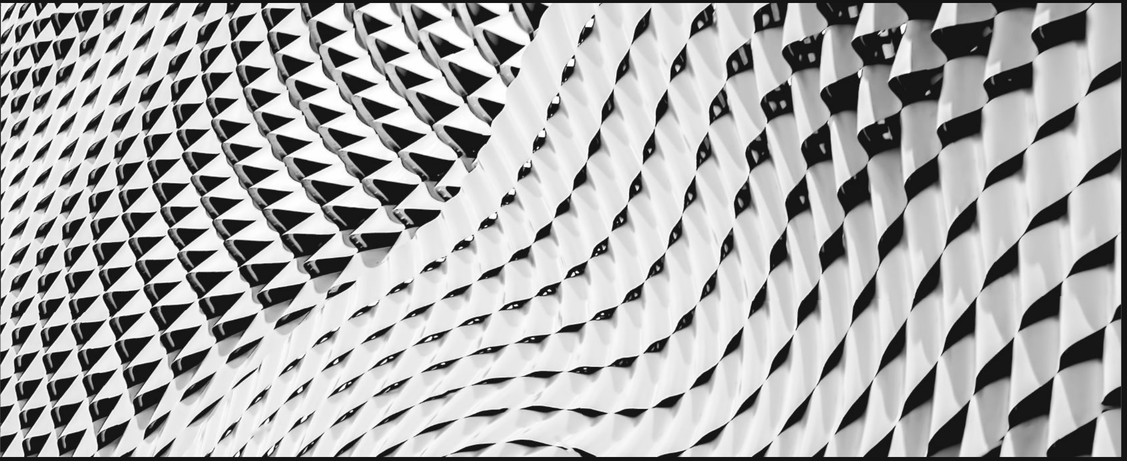


Hill climbing



Source:
Insert source data here
Insert source data here
Insert source data here

Score Calculation Types



Red Hat

Score Calculation Types

Incremental Java
Constraint Streams
imgflip.com
JAKE-CLARK.TUMBLR

Score Calculation Types

- Simple Java
  - Easy to write
  - Slow
  - DO NOT USE FOR PRODUCTION!!!
- Incremental
  - Hard to write
  - Hard to maintain
  - Error Prone
  - Potentially fast

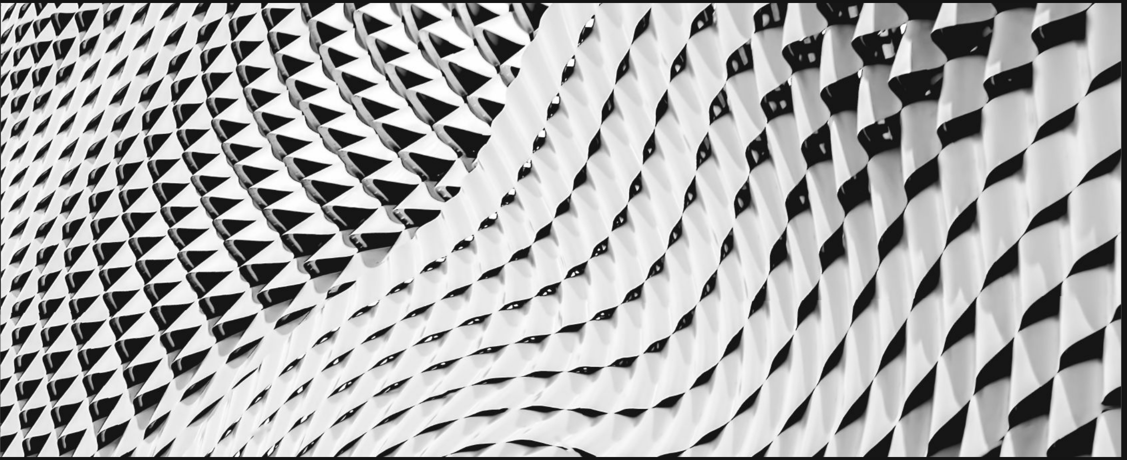
Source:
Insert source data here
Insert source data here
Insert source data here

Score Calculation Types

- DRL
  - Learning Curve
  - Relatively Easy to Write (once familiar with Drools)
  - Fast: implicit incremental calculation
- Constraints Streams
  - Java Streams-like API
  - Uses Drools under the covers
  - Relatively Easy to Write (if familiar with Java Streams API).
  - Fast: implicit incremental calculation

Source:
Insert source data here
Insert source data here
Insert source data here

Environment Modes



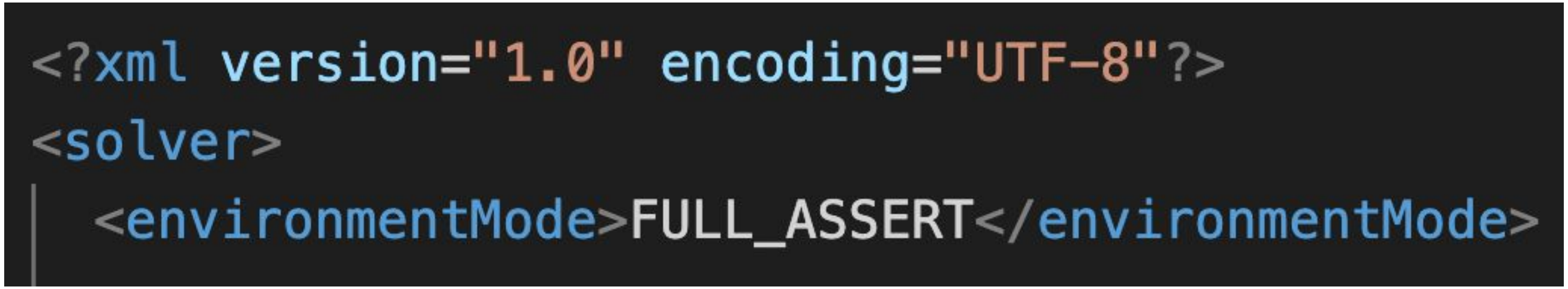
Environment Modes

Are There Bugs In My Code

- FULL_ASSERT
- NON_INTRUSIVE_FULL_ASSERT
- FAST_ASSERT
- REPRODUCIBLE
- NON_REPRODUCIBLE

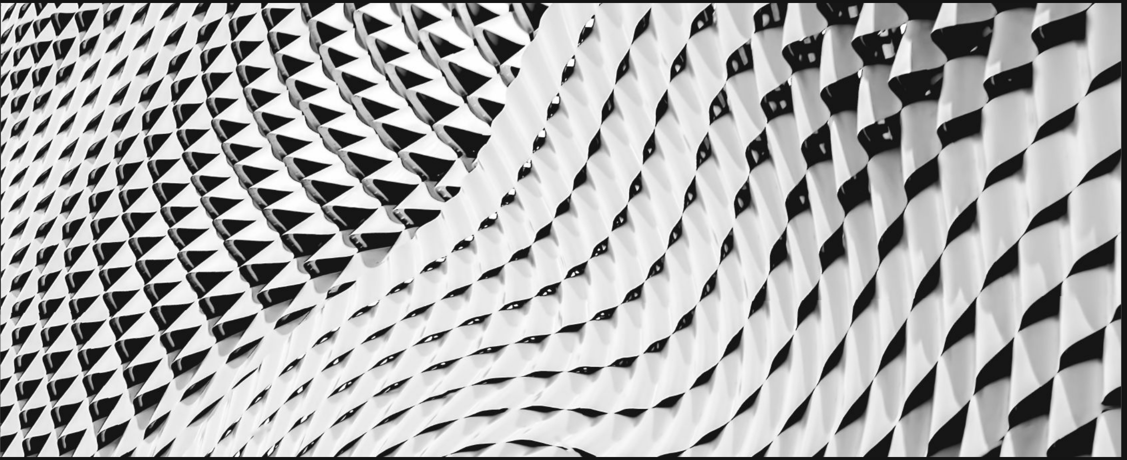
Environment Modes

Are There Bugs In My Code



```
<?xml version="1.0" encoding="UTF-8"?>
<solver>
  <environmentMode>FULL_ASSERT</environmentMode>
```

Keep the User in Control



Keep the User in Control

Source:
Insert source data here
Insert source data here
Insert source data here

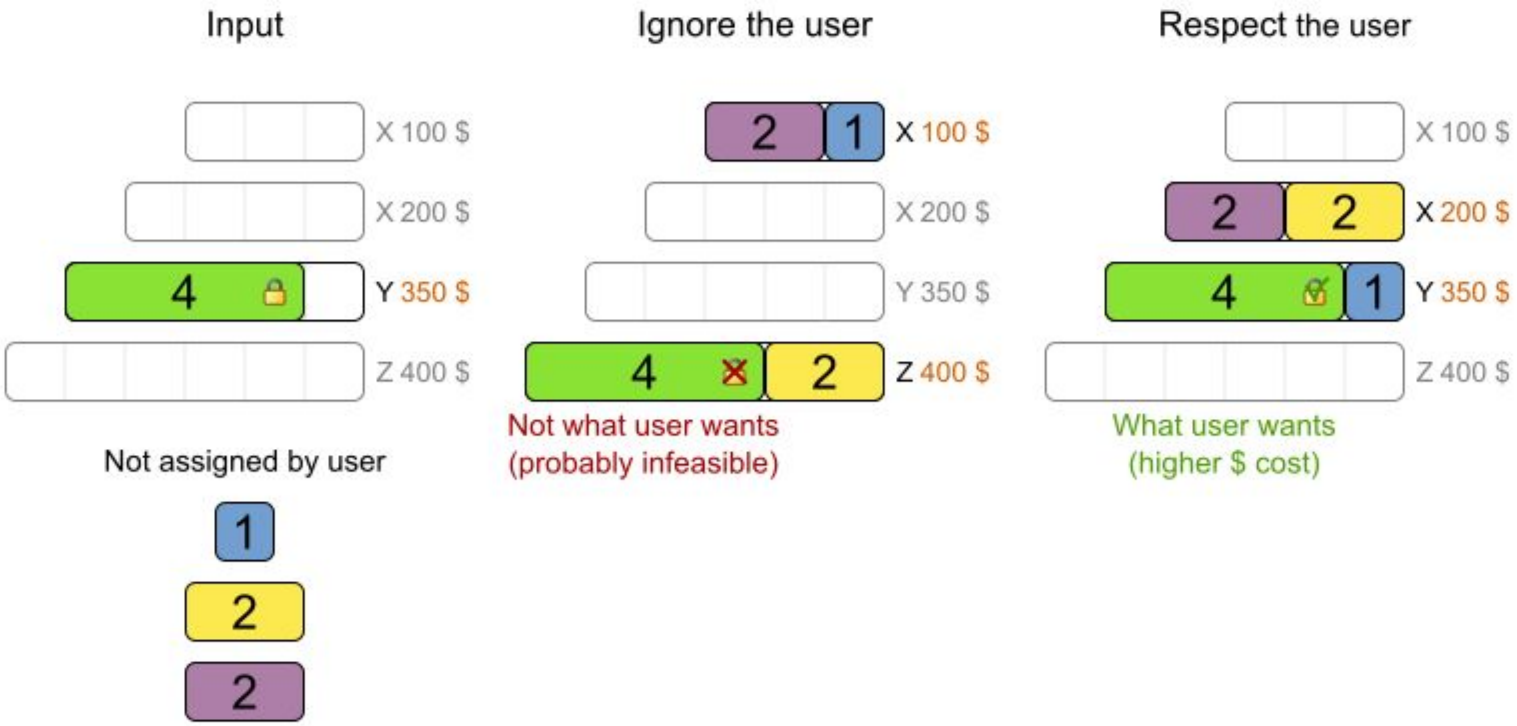
Keep the User in Control

Human Planners are your Product Owner

- Domain Knowledge
  - Domain Model
  - Constraints
- Setting Planning Priorities
  - Time?
  - Money?
  - Customer Satisfaction?
  - Ecological Footprint?
- Visualization and Publishing: Evolution vs Revolution
  - Excel is often a good choice to start with

Keep the User in Control

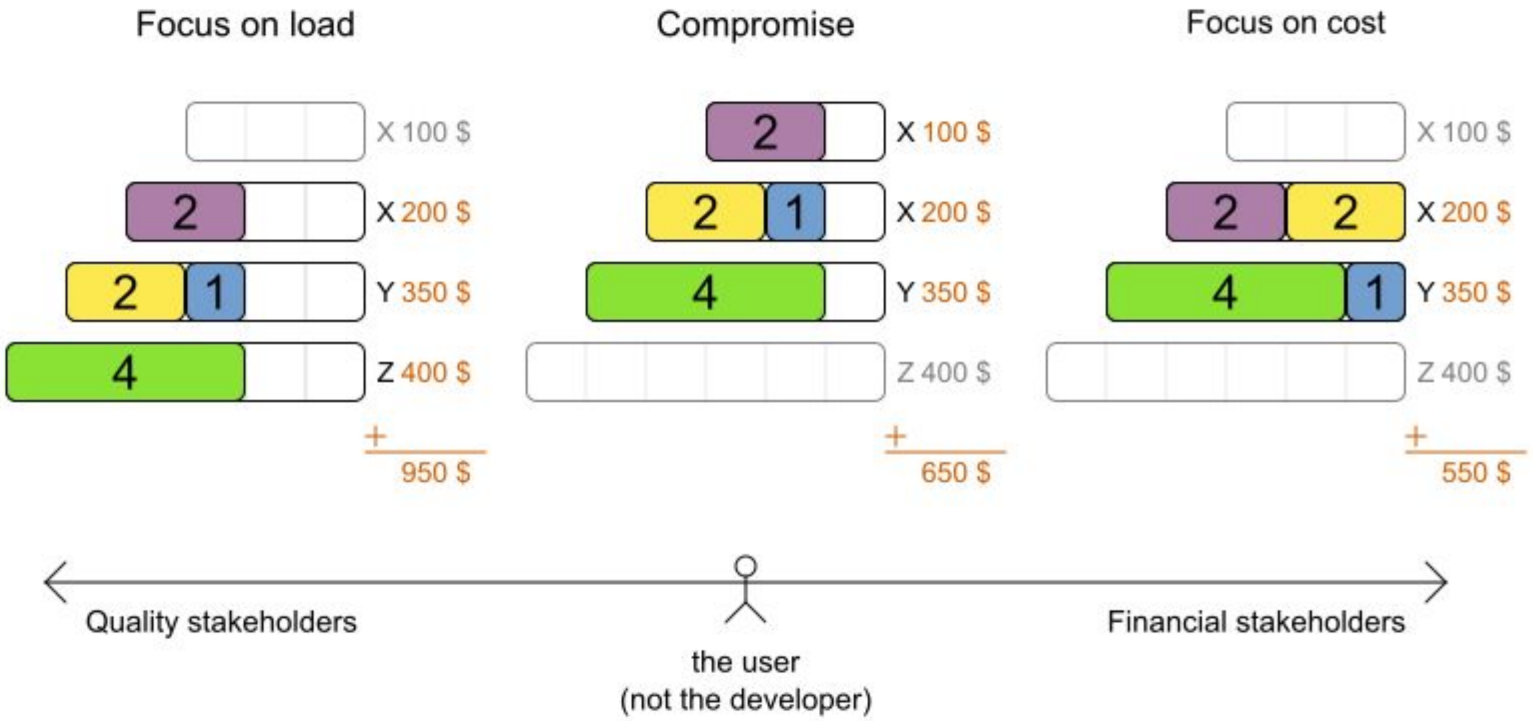
Pinning Planning Entities



Source:
Insert source data here
Insert source data here
Insert source data here

Keep the User in Control

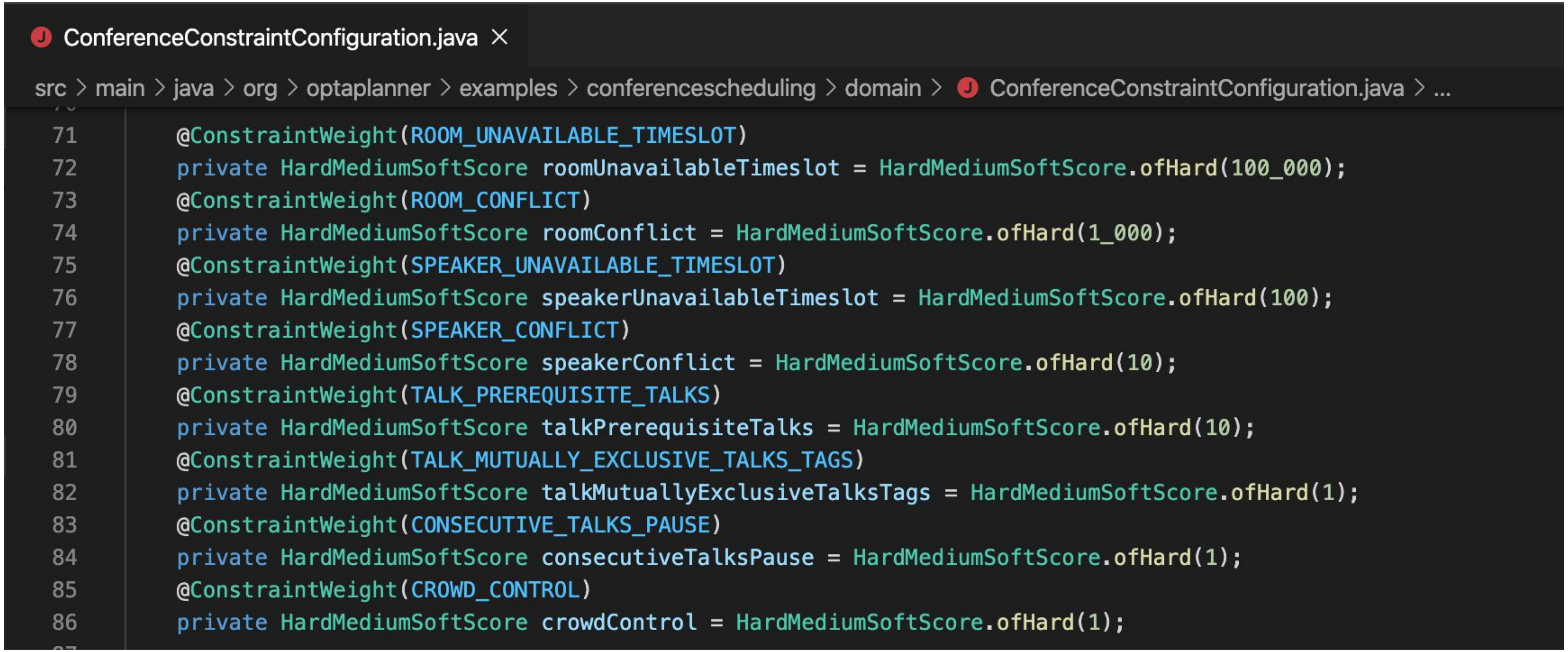
Constraint Weights



Source:
Insert source data here
Insert source data here
Insert source data here

Keep the User in Control

Constraint Weight Configuration

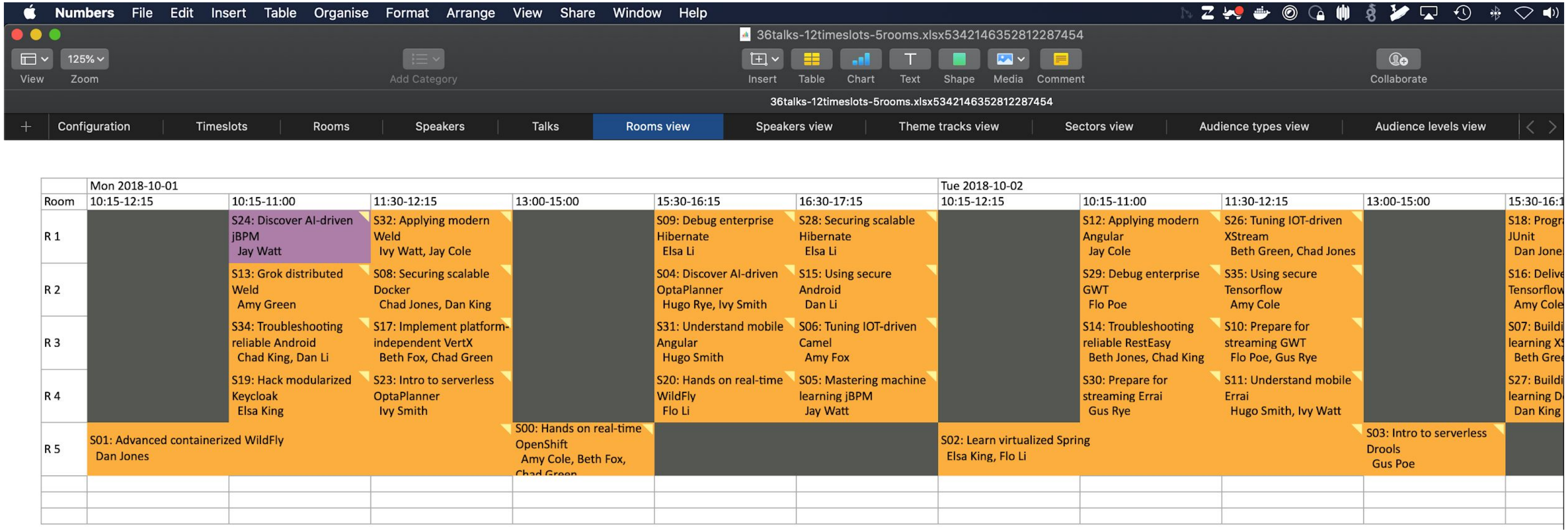

ConferenceConstraintConfiguration.java
src > main > java > org > optaplanner > examples > conferencescheduling > domain > ConferenceConstraintConfiguration.java > ...

```java
    @ConstraintWeight(ROOM_UNAVAILABLE_TIMESLOT)
    private HardMediumSoftScore roomUnavailableTimeslot = HardMediumSoftScore.ofHard(100_000);
    @ConstraintWeight(ROOM_CONFLICT)
    private HardMediumSoftScore roomConflict = HardMediumSoftScore.ofHard(1_000);
    @ConstraintWeight(SPEAKER_UNAVAILABLE_TIMESLOT)
    private HardMediumSoftScore speakerUnavailableTimeslot = HardMediumSoftScore.ofHard(100);
    @ConstraintWeight(SPEAKER_CONFLICT)
    private HardMediumSoftScore speakerConflict = HardMediumSoftScore.ofHard(10);
    @ConstraintWeight(TALK_PREREQUISITE_TALKS)
    private HardMediumSoftScore talkPrerequisiteTalks = HardMediumSoftScore.ofHard(10);
    @ConstraintWeight(TALK_MUTUALLY_EXCLUSIVE_TALKS_TAGS)
    private HardMediumSoftScore talkMutuallyExclusiveTalksTags = HardMediumSoftScore.ofHard(1);
    @ConstraintWeight(CONSECUTIVE_TALKS_PAUSE)
    private HardMediumSoftScore consecutiveTalksPause = HardMediumSoftScore.ofHard(1);
    @ConstraintWeight(CROWD_CONTROL)
    private HardMediumSoftScore crowdControl = HardMediumSoftScore.ofHard(1);
```

Source:
Insert source data here
Insert source data here
Insert source data here

Keep the User in Control

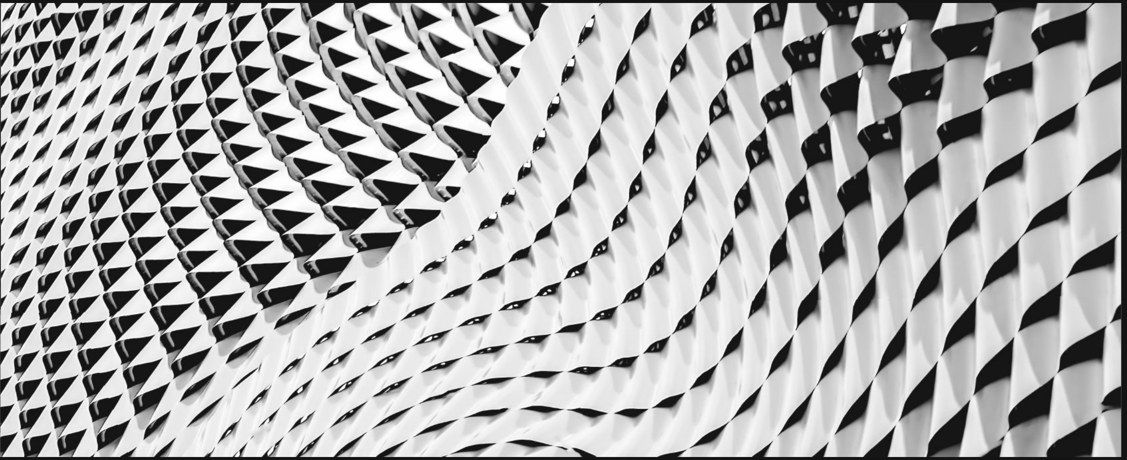
Visualization



	Mon 2018-10-01						Tue 2018-10-02				
Room	10:15-12:15	10:15-11:00	11:30-12:15	13:00-15:00	15:30-16:15	16:30-17:15	10:15-12:15	10:15-11:00	11:30-12:15	13:00-15:00	15:30-16:1
R 1		S24: Discover AI-driven jBPM Jay Watt	S32: Applying modern Weld Ivy Watt, Jay Cole		S09: Debug enterprise Hibernate Elsa Li	S28: Securing scalable Hibernate Elsa Li		S12: Applying modern Angular Jay Cole	S26: Tuning IOT-driven XStream Beth Green, Chad Jones		S18: Progr JUnit Dan Jone
R 2		S13: Grok distributed Weld Amy Green	S08: Securing scalable Docker Chad Jones, Dan King		S04: Discover AI-driven OptaPlanner Hugo Rye, Ivy Smith	S15: Using secure Android Dan Li		S29: Debug enterprise GWT Flo Poe	S35: Using secure Tensorflow Amy Cole		S16: Delive Tensorflow Amy Cole
R 3		S34: Troubleshooting reliable Android Chad King, Dan Li	S17: Implement platform-independent VertX Beth Fox, Chad Green		S31: Understand mobile Angular Hugo Smith	S06: Tuning IOT-driven Camel Amy Fox		S14: Troubleshooting reliable RestEasy Beth Jones, Chad King	S10: Prepare for streaming GWT Flo Poe, Gus Rye		S07: Buildi learning XS Beth Gree
R 4		S19: Hack modularized Keycloak Elsa King	S23: Intro to serverless OptaPlanner Ivy Smith		S20: Hands on real-time WildFly Flo Li	S05: Mastering machine learning jBPM Jay Watt		S30: Prepare for streaming Errai Gus Rye	S11: Understand mobile Errai Hugo Smith, Ivy Watt		S27: Buildi learning D Dan King
R 5	S01: Advanced containerized WildFly Dan Jones			S00: Hands on real-time OpenShift Amy Cole, Beth Fox, Chad Green			S02: Learn virtualized Spring Elsa King, Flo Li			S03: Intro to serverless Drools Gus Poe	

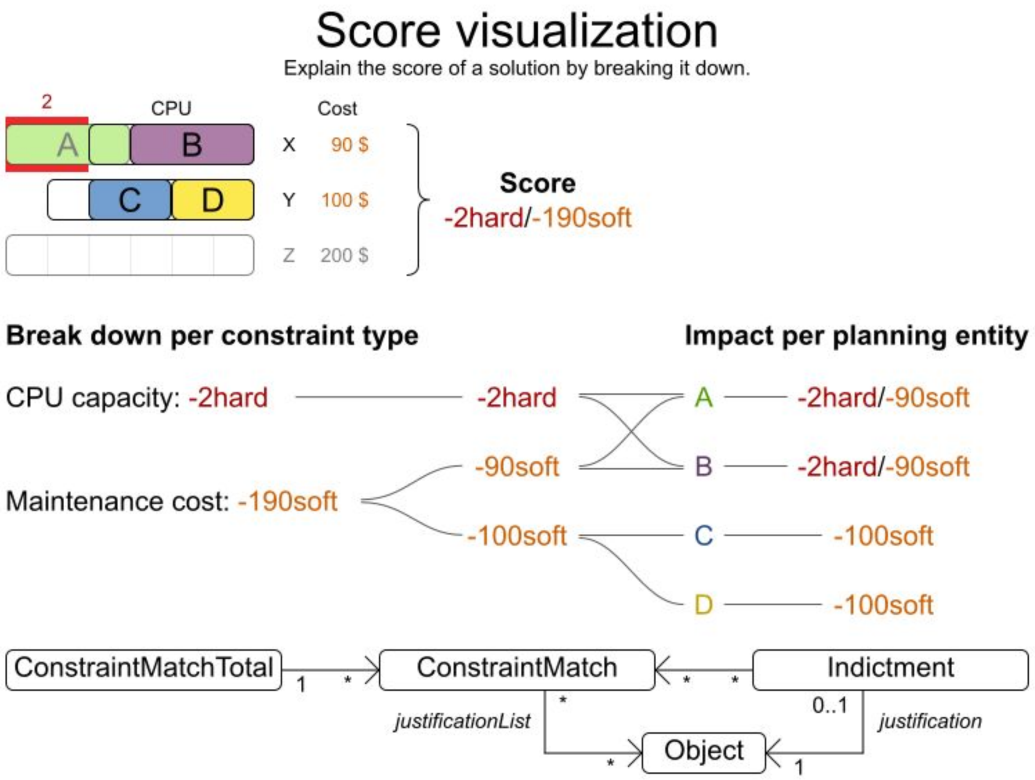
Source:
Insert source data here
Insert source data here
Insert source data here

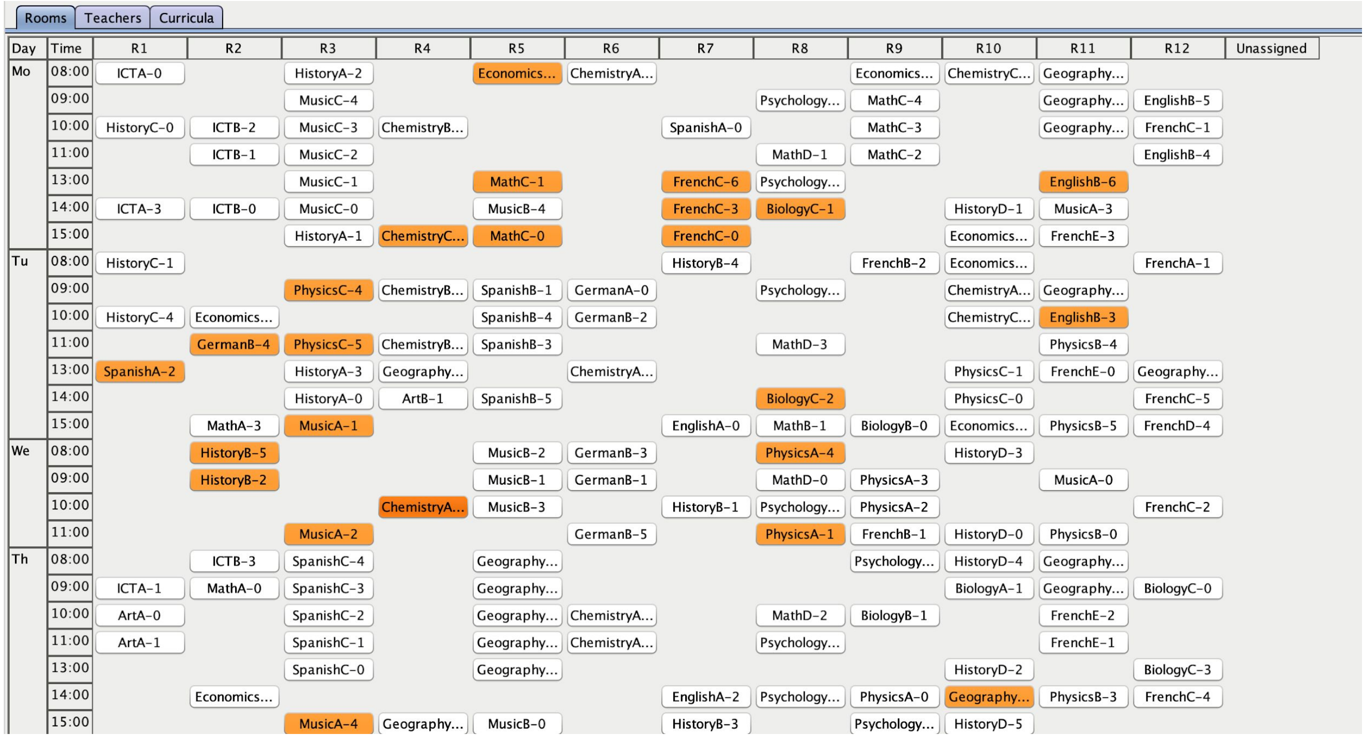
Explain the Score



Explain the Solution

Explaining the Score

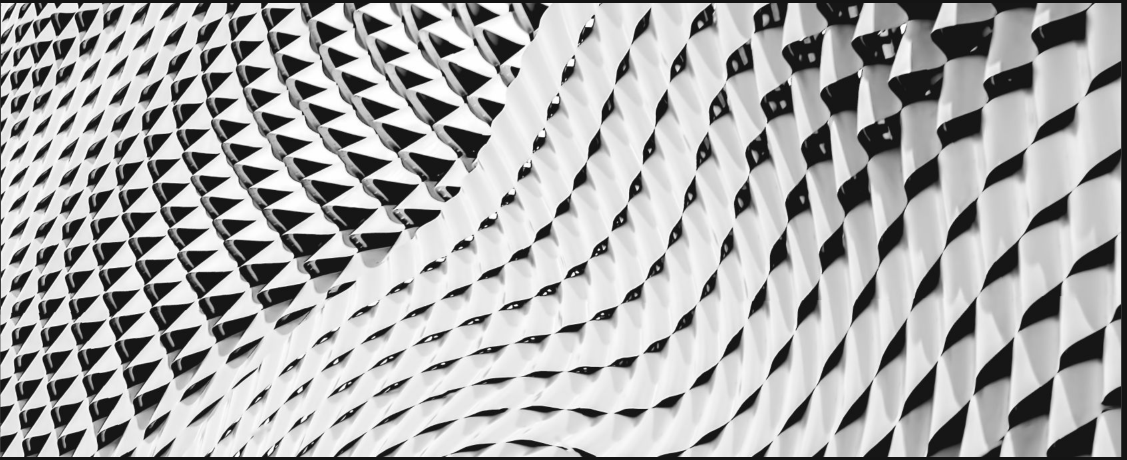




Day	Time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	Unassigned
Mo	08:00	ICTA–0		HistoryA–2		Economics...	ChemistryA...			Economics...	ChemistryC...	Geography...		
	09:00			MusicC–4					Psychology...	MathC–4		Geography...	EnglishB–5	
	10:00	HistoryC–0	ICTB–2	MusicC–3	ChemistryB...			SpanishA–0		MathC–3		Geography...	FrenchC–1	
	11:00		ICTB–1	MusicC–2					MathD–1	MathC–2			EnglishB–4	
	13:00			MusicC–1		MathC–1		FrenchC–6	Psychology...			EnglishB–6		
	14:00	ICTA–3	ICTB–0	MusicC–0		MusicB–4		FrenchC–3	BiologyC–1		HistoryD–1	MusicA–3		
	15:00			HistoryA–1	ChemistryC...	MathC–0		FrenchC–0			Economics...	FrenchE–3		
Tu	08:00	HistoryC–1						HistoryB–4		FrenchB–2	Economics...		FrenchA–1	
	09:00			PhysicsC–4	ChemistryB...	SpanishB–1	GermanA–0		Psychology...		ChemistryA...	Geography...		
	10:00	HistoryC–4	Economics...			SpanishB–4	GermanB–2				ChemistryC...	EnglishB–3		
	11:00		GermanB–4	PhysicsC–5	ChemistryB...	SpanishB–3			MathD–3			PhysicsB–4		
	13:00	SpanishA–2		HistoryA–3	Geography...		ChemistryA...				PhysicsC–1	FrenchE–0	Geography...	
	14:00			HistoryA–0	ArtB–1	SpanishB–5			BiologyC–2		PhysicsC–0		FrenchC–5	
	15:00		MathA–3	MusicA–1				EnglishA–0	MathB–1	BiologyB–0	Economics...	PhysicsB–5	FrenchD–4	
We	08:00		HistoryB–5			MusicB–2	GermanB–3		PhysicsA–4		HistoryD–3			
	09:00		HistoryB–2			MusicB–1	GermanB–1		MathD–0	PhysicsA–3		MusicA–0		
	10:00				ChemistryA...	MusicB–3		HistoryB–1	Psychology...	PhysicsA–2			FrenchC–2	
	11:00			MusicA–2			GermanB–5		PhysicsA–1	FrenchB–1	HistoryD–0	PhysicsB–0		
Th	08:00		ICTB–3	SpanishC–4		Geography...				Psychology...	HistoryD–4	Geography...		
	09:00	ICTA–1	MathA–0	SpanishC–3		Geography...					BiologyA–1	Geography...	BiologyC–0	
	10:00	ArtA–0		SpanishC–2		Geography...	ChemistryA...		MathD–2	BiologyB–1		FrenchE–2		
	11:00	ArtA–1		SpanishC–1		Geography...	ChemistryA...		Psychology...			FrenchE–1		
	13:00			SpanishC–0		Geography...					HistoryD–2		BiologyC–3	
	14:00		Economics...					EnglishA–2	Psychology...	PhysicsA–0	Geography...	PhysicsB–3	FrenchC–4	
	15:00			MusicA–4	Geography...	MusicB–0		HistoryB–3		Psychology...	HistoryD–5			

Source:
Insert source data here
Insert source data here
Insert source data here

Reproducibility



Reproducibility

Pitfalls

- Random calls that don't use seeded Random
- `HashMap` -> Use `LinkedHashMap`
- `HashSet` -> Use `LinkedHashSet`
- Time gradient algorithms (Simulated Annealing)
- Work stealing (`EnvironmentMode`)

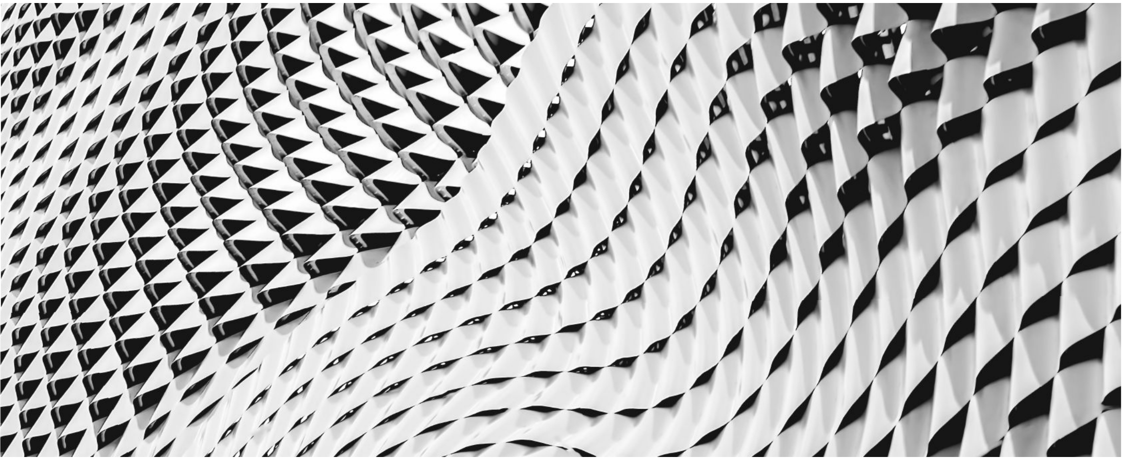
Source:
Insert source data here
Insert source data here
Insert source data here

Reproducibility
Is Gold!

- Debugging
- Support
- Demos
- Your Sanity

Source:
Insert source data here
Insert source data here
Insert source data here

Resources



References

- OptaPlanner Website: https://www.optaplanner.org
  - Learning: https://www.optaplanner.org/learn/slides.html
  - Examples
- Domain Modelling Guide:
  - https://www.optaplanner.org/blog/2016/10/26/DomainModelingGuide.html
- 7 Ways to Fail your Optimization Project:
  - https://www.optaplanner.org/blog/2016/04/18/7WaysToFailYourOptimizationProject.html

Source:
Insert source data here
Insert source data here
Insert source data here

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat	f facebook.com/redhatinc
▶ youtube.com/user/RedHatVideos	twitter.com/RedHat

Red Hat

Some more examples

- Task Assignment: assign task to the proper employee
- Hospital bed planning
- Course scheduling/timetabling
- Project job scheduling
- Meeting scheduling
- …. and my more!!

Source:
Insert source data here
Insert source data here
Insert source data here

Two Types of Constraints

Hard Constraints must be satisfied by any solution (for it to be a feasible solution)

- Crew must not exceed 8 hours in 24
- Truck must not be overloaded
- Every shift must have a full complement of nurses

Soft Constraints should be satisfied as much as possible (better solutions satisfy more soft constraints)

- Crews should return home every 5 days
- A nurse’s time preference should be honored

Source:
Insert source data here
Insert source data here
Insert source data here