Vehicle Routing with Explainable AI

Lukáš Petrovický,
Solver Engineer

April 17, 2024

timefold

Agenda

- What’s Timefold?
- Intro to planning optimization
- Modelling with Timefold
- Explaining results to humans
- Working together with humans

timefold

What is Timefold

You may know this one...

What you may not know…

- “Red Hat build of OptaPlanner 8 reaches End of Life on 30 May 2024…”
- “Red Hat does not intend to offer the next version, Red Hat build of OptaPlanner 9, as a standalone product…”

Source: Announcement by Red Hat on March 19th 2024 (https://access.redhat.com/articles/7060671)

And therefore…

- Timefold is a continuation of OptaPlanner.
  - Open-core company founded in 2022.
  - Hired most of the original team.

- Timefold Solver CE is an open-source fork of OptaPlanner
  - Apache-licensed.
  - Monthly releases with major new features and bugfixes.
  - Compatible with OptaPlanner, easy migration.
  - Active on StackOverflow.

- Timefold Solver Enterprise Edition brings commercial features:
  - Multi-threaded solving.
  - Nearby selection.

timefold

Planning optimization

Planning problems are everywhere!

What makes optimization difficult?

- Scale
- Number of constraints
- Humans

Many automate, too few optimize

- Loading autonomous trucks by robots doesn’t mean that:
    - It’s done as quickly as possible.
    - It uses as few trucks as possible.
    - Those trucks will drive the most efficient route.

- Mathematical optimization to the rescue
    - AI is not just machine learning!
    - Various techniques.
    - Implementations typically called “solvers”.

How Timefold Solver works

1. Model a solution
   - ○ Describe your domain.

2. Describe constraints
   - ○ A constraint penalizes an unwanted state of your domain.
   - ○ Hard constraint: One vehicle at two places at the same time.
   - ○ Soft constraint: Delivery window exceeded by N minutes.

3. Solver searches for a solution with best score.
   - ○ Score: Sum of constraints broken by a solution.
   - ○ Typically results in near-optimal solutions.

timefold

Modeling with Timefold

Let’s stick with Vehicle Routing

- The scenario
  - A given number of vehicles sit in a depot.
    - Vehicles have a predetermined cargo capacity.
  - A given number of customers expect delivery.
    - Only available within a time window.
  - Also called CVRPTW (Capacitated VRP with Time Windows)

- Hard constraints
  - Don’t exceed vehicle capacity.
  - Fit within the time window.

- Soft constraints
  - Minimize distance traveled.

Visit represents a customer

```
public class Visit {

   private Location location;
   private int demand;
   private LocalDateTime minStartTime;
   private LocalDateTime maxEndTime;


  …


}
```

`Vehicle` visits customers in a sequence

```java
@PlanningEntity
public class Vehicle {

   private String id;
   private int capacity;
   private Location homeLocation;

   @PlanningListVariable
   private List<Visit> visits;

   …

}
```

timefold

`VehicleRoutePlan` represents the entire domain

```
@PlanningSolution
public class VehicleRoutePlan {

    @PlanningEntityCollectionProperty
    private List<Vehicle> vehicles;

    @PlanningEntityCollectionProperty
    @ValueRangeProvider
    private List<Visit> visits;

    @PlanningScore
    private HardSoftScore score;


   …
```

timefold

Constraints complete the picture

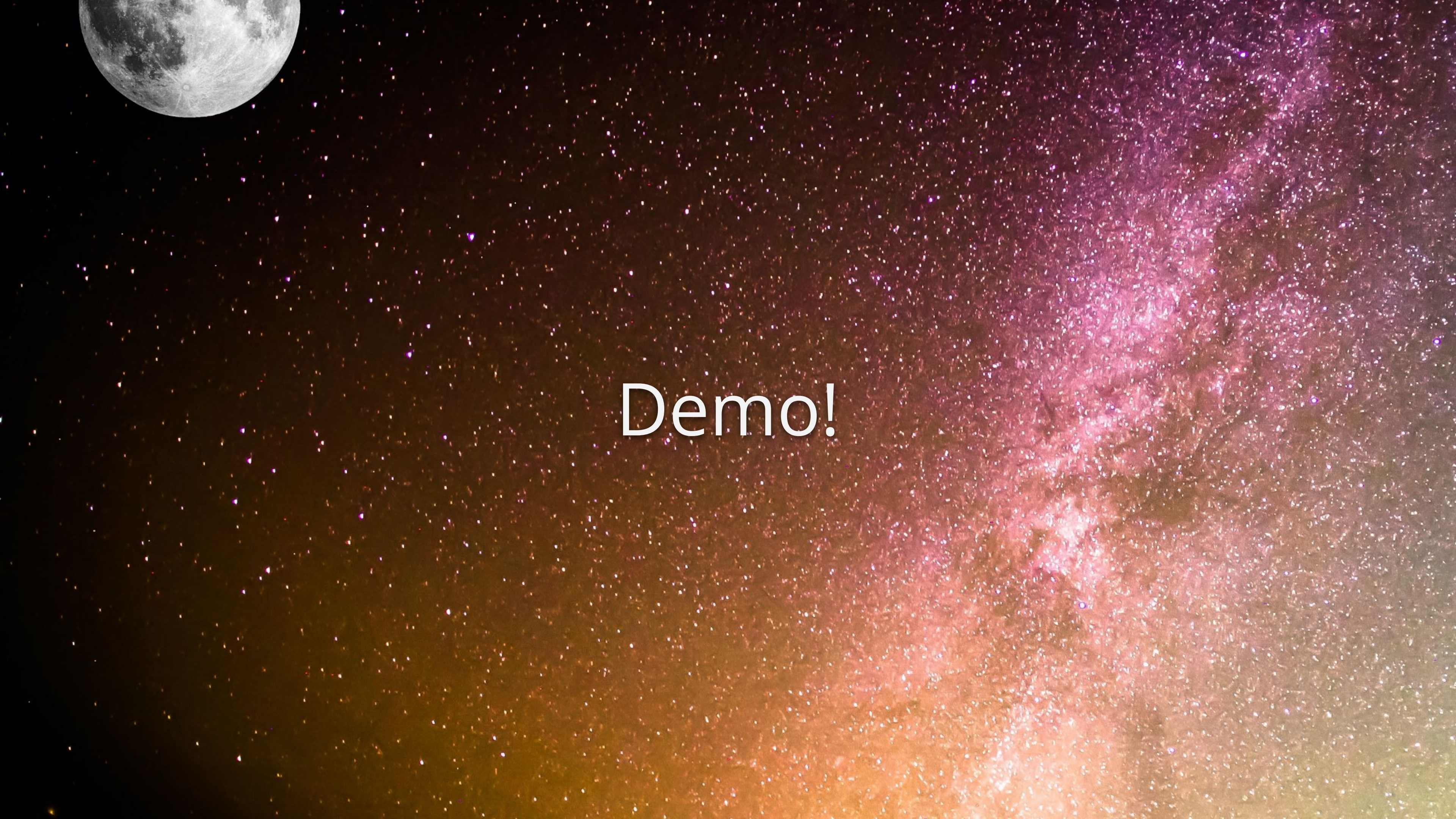
```
public class VehicleRoutingConstraintProvider
   implements ConstraintProvider {

   @Override
   public Constraint[] defineConstraints(ConstraintFactory factory) {
       return new Constraint[] {
           vehicleCapacity(factory),
           …
       };
   }

   Constraint vehicleCapacity(ConstraintFactory factory) {
       return factory.forEach(Vehicle.class)
           .filter(vehicle -> vehicle.getTotalDemand() > vehicle.getCapacity())
           .penalize(HardSoftScore.ONE_HARD,
                vehicle -> vehicle.getTotalDemand() - vehicle.getCapacity())
           .asConstraint(“Vehicle capacity”);
   }

   …

}
```

Running the solver

```
// Fill your domain with data.
var solution = new VehicleRoutePlan(...);

// Prepare solver configuration.
var solverConfig = new SolverConfig()
    .withEntityClasses(...)
    .withSolutionClass(...)
    .withConstraintProviderClass(...)
    …;

// Build the solver.
var solver = SolverFactory.create(solverConfig)
    .buildSolver();

// Run the solver and get the best found solution.
var bestSolution = solver.solve(solution);
```

Devoxx Belgium 2023 Deep Dive

Three hours’ worth of details!



Demo!

Vehicle Routing Quickstart

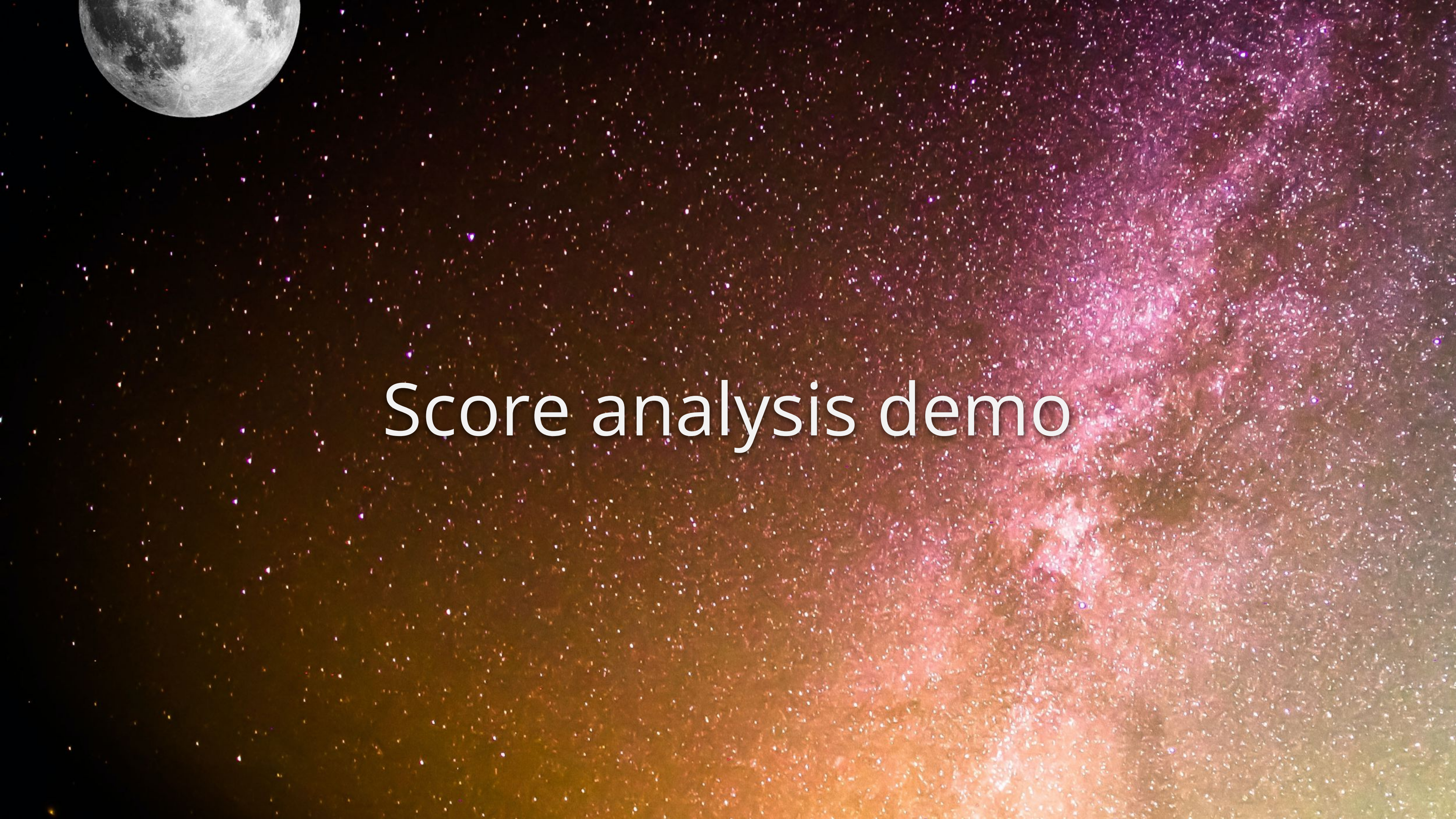
- Timefold Solver has many quickstarts
  - Small example applications.
  - Designed to get you started quickly.

Explaining results to humans

0hard/-91649soft … WHAT?

- So you got a result from the solver…
  - Does the score *mean* anything to you?
- How good is the solution?
  - Can I use it?
- What can I expect when I deploy the solution in the real world?
  - Will my customers be happy?
  - Will my drivers be happy?
- Score analysis to the rescue!
  - Unique feature of the Timefold Solver.

timefold



Score analysis demo

AI is here to help

- Scenario
  - A customer calls our hotline, requesting a delivery.
  - When can we best fit it in?

- Problem
  - It takes time to find a solution.
  - Customer doesn't have time to wait on the phone.

- Recommended Fit API solves the problem
  - Greedy algorithm to provide super-fast initial response.
  - Gives our operator several options to choose from.
  - Full solver run can be performed later.

Recommendation demo

Summary

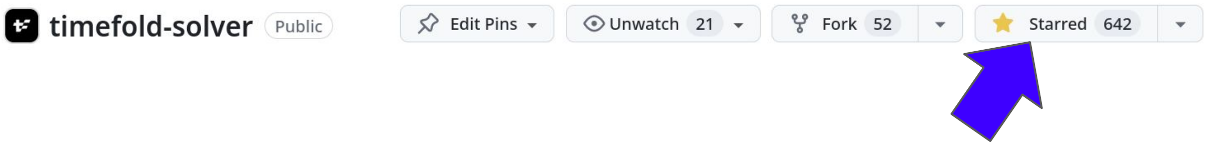
- AI need not be a black box.
- AI can be integrated into existing workflows.
- All of these features available in Community Edition!

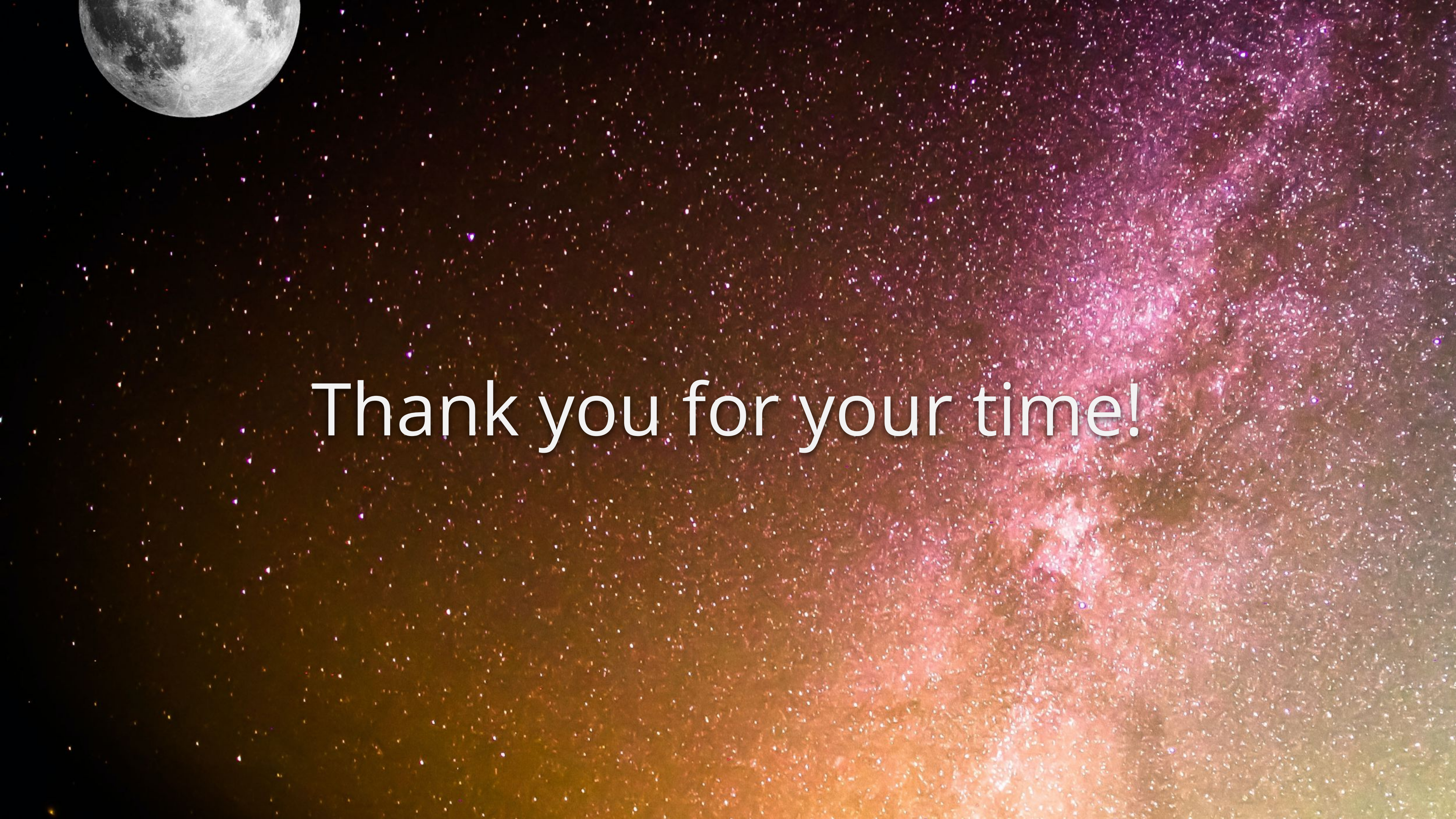
Where to go next

- Timefold:
  - https://timefold.ai

- Timefold Solver Documentation:
  - https://docs.timefold.ai/timefold-solver

- Star us now!





Thank you for your time!