Challenges of designing explanation tools for optimization systems

FIIA 2021

Mathieu Lerouge
mathieulerouge.github.io

MICS, CentraleSupélec Université Paris-Saclay

7 October 2021

A talk about optimization and explanation?

Our observation

End-users of an optimization system (*e.g.* DecisionBrain's) may be **reluctant to accept** the decisions computed by the system sometimes.

Our approach

Providing explanations to end-users as a way to **reach some transparency** and **improve their confidence** in the system.

Plan

Plan

Context

3 components

(A) Real-world complex situation that can be modeled as an **optimization problem**;

(B) **Optimization system** for solving the problem;

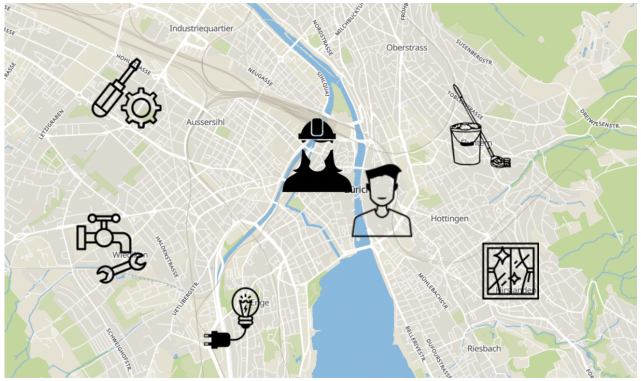
(C) **Non-expert end-users** using the optimization system (*via* an interface) who need explanations.

↪ What are (A), (B) and (C) in a given use case?

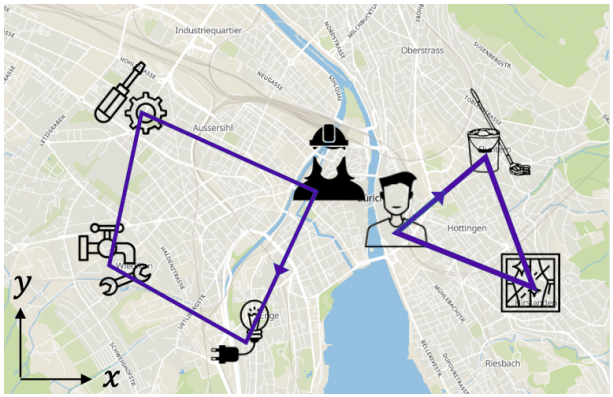
Our use case - (A) Optimization problem

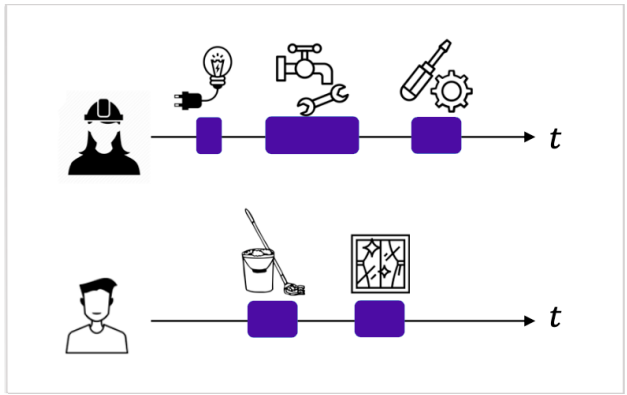
Workforce Scheduling and Routing Problem (WSRP)

Instance:



Solution:





Our use case - (A) Optimization problem

Inputs of the WSRP

- $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$

  set of n **mobile employees** $\mathcal{E}_i$ characterized by:
  - a **skill level**, $s_i \in \mathbb{N}$;
  - a **working time-window**, $[\![a_i, b_i]\!] \subset \mathbb{N}$;
  - a **location**, $l_i \in \mathbb{R}^2$.

- $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_m\}$

  set of m **tasks** $\mathcal{T}_j$ characterized by:
  - a **skill level**, $s_j \in \mathbb{N}$;
  - an **availability time-window**, $[\![a_j, b_j]\!] \subset \mathbb{N}$;
  - a **duration**, $d_j \in \mathbb{N}$;
  - a **location** $l_j \in \mathbb{R}^2$.

Our use case - (A) Optimization problem

Mathematical formulation of the WSRP

$$\max \left(\underbrace{\sum_{i \in \mathcal{E}} \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{T}} U_{ijk}\ d_j}_{\text{total working duration}} \quad , \quad \underbrace{- \sum_{i \in \mathcal{E}} \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{T}} U_{ijk}\ \Delta t^i_{jk}}_{\text{total traveling duration}} \right)$$

s.t.
- employees must work within their time windows;
- tasks must be performed within their time windows;
- employees must be skilled enough to perform the tasks;
- ...

$U_{ijk} \in \{0, 1\}$ whether or not $\mathcal{E}_i$ goes from $\mathcal{T}_j$ to $\mathcal{T}_k$;
$T_j \in \mathbb{N}$ performing time of $\mathcal{T}_j$.

Our use case - (B) Optimization system

WSRP-solving systems

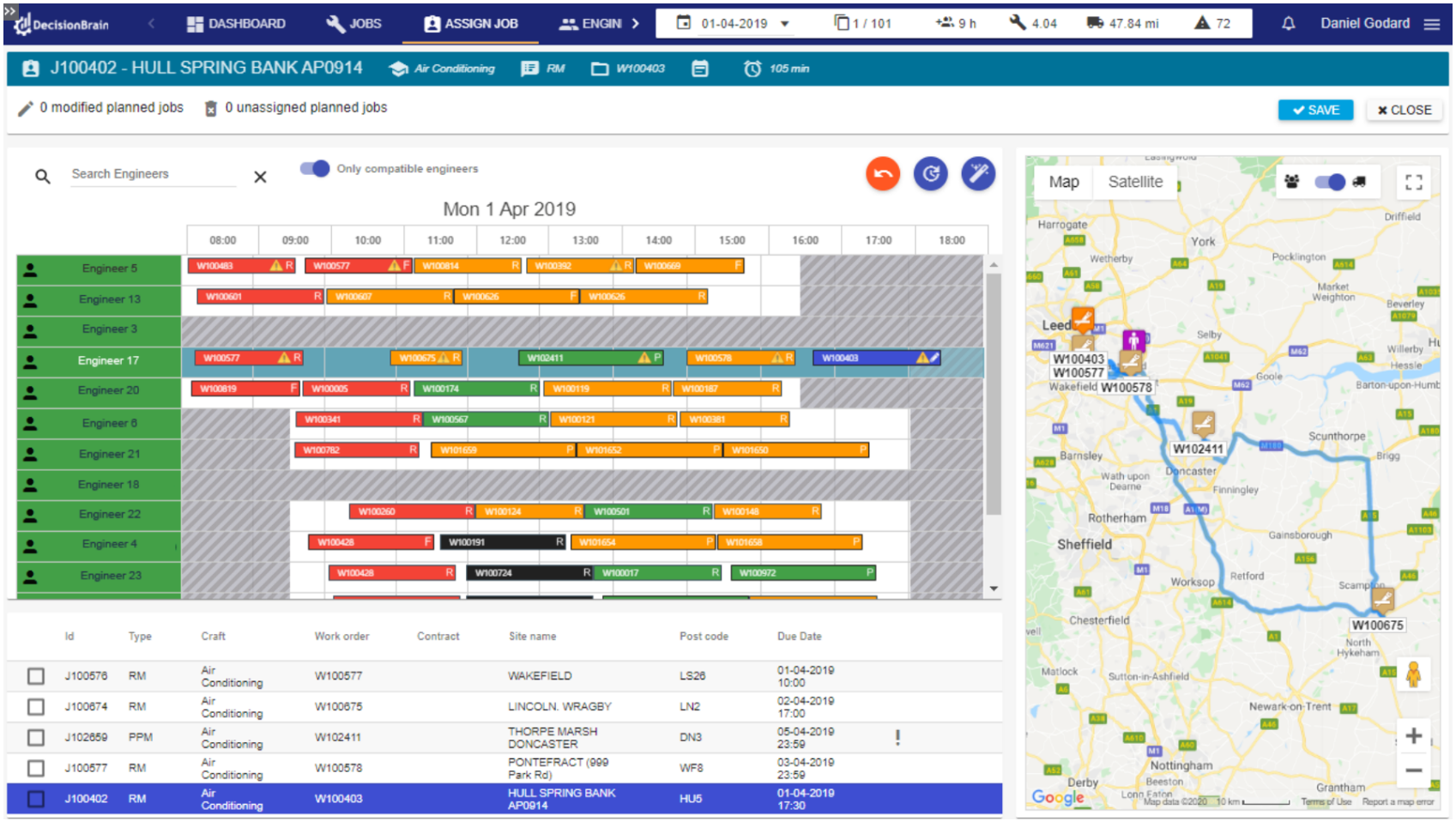
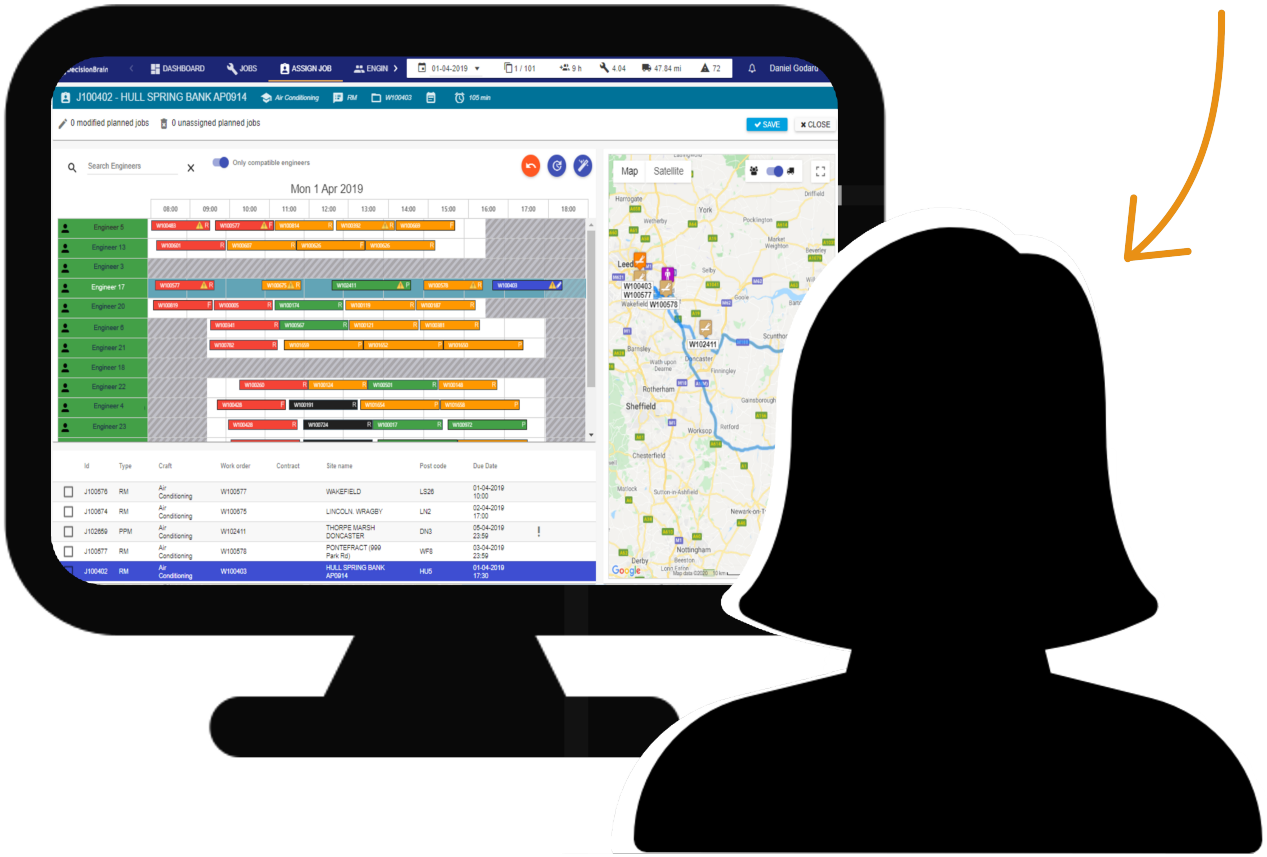


Figure: Graphic User Interface of DecisionBrain's Dynamic Scheduler

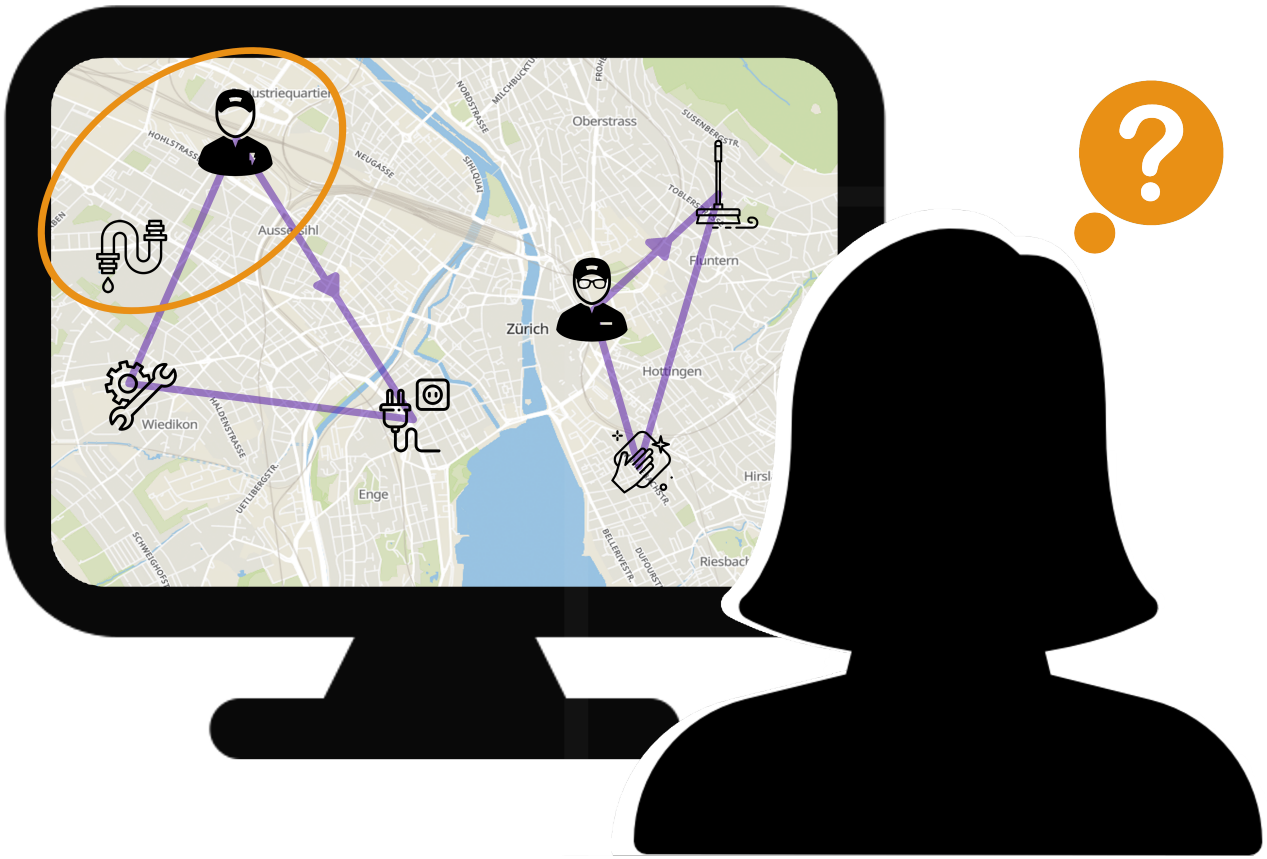
Our use case - (C) Non-expert end-users

Planners using WSRP-solving systems



Plan

Problematic situation with the optimization system



User: "Why is Adam not performing the plumbing task in addition to the two other tasks of his planning?"

Stakes

End-users' issues

- **Confusion**: in a given solution,
  - presence of unexpected decisions;
  - difference between its quality and the expected one.
- **Frustration**: optimization systems experienced as non-transparent systems / black boxes.

↪ **Users loosing confidence** in the optimization system.

Goals

Our proposals

Designing explanation tools, which are **independent** from the solving algorithm, and that enable users:

- to ask questions about a solution (by selecting template questions in a given list) and get explanations back;
- to explore the space of feasible solutions and the space of neighboring instances;
- to identify critical data in the inputs.

↪ **Tackling users' black-box feeling** about the system, **increasing users' trust** in the system.

Plan

Our observation

Very few works dealing with explanation in optimization

Article	Explanation approach				
	Based on	Applied to	Applicable to	Dependance	Questions
[Ludwig et al., 2018]	Solving algorithm memoriz.	Makespan Scheduling Problem	Specific MSP solved via specific algorithm	Depending on solving algorithm	1 type
[Čyras et al., 2019]	Abstract Argument.	Makespan Scheduling Problem	Specific problems with binary decision variables	Not depending on solving algorithm	3 types
[Korikov et al., 2021]	Inverse Optim.	Knapsack, Portfolio	Specific linear problems whose weights in OF are not in constraints	Depending on on solving algorithm	1 type

[Ludwig et al., 2018] Explaining Complex Scheduling Decisions
[Čyras et al., 2019] Argumentation for Explainable Scheduling
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Plan

One key type of questions

Contrastive questions

"Why $\underbrace{\text{this current decision}}_{\text{fact}}$ rather than $\underbrace{\text{that other one}}_{\text{foil (often implicit)}}$?"

e.g. "Why is Adam not performing the plumbing task in addition to his two other tasks?"

Relevance of working with contrastive questions

- they correspond to most of the "Why" questions people ask [Miller, 2019];
- they tend to specify the question, to narrow the set of solutions to exam.

[Miller, 2019] Explanation in artificial intelligence: Insights from the social sciences

Plan

One key type of explanations

Counterfactual explanations

"To get that decision, the inputs should have been this way?"

counterfact in decisions

counterfact in inputs

e.g. "So that Adam performs the plumbing task in addition to his two other tasks, the electric task should be available 15 minutes earlier".

Benefits of employing counterfactual explanations

- they get around the challenge of explaining the functionality or the rationale of complex algorithmic decision-making systems [Wachter et al., 2018];
- they reveal critical data in the inputs.

[Wachter et al., 2018] Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR

Plan

Template contrastive questions in the WSRP context

Questions about assigning a new task to an employee

"**Why** is the employee $<\mathcal{E}_i>$ **not** performing the task $<\mathcal{T}_k>$...

- ... instead of the task $<\mathcal{T}_j>$?"
- ... instead of one of his/her planning's task?"
- ... just after the task $<\mathcal{T}_j>$?"
- ... in addition to his/her planning's tasks?"

Template contrastive questions in the WSRP context

Questions about changing the order of a task in a planning

"**Why** is the employee $<\mathcal{E}_i>$ **not** performing the task $<\mathcal{T}_k>$...

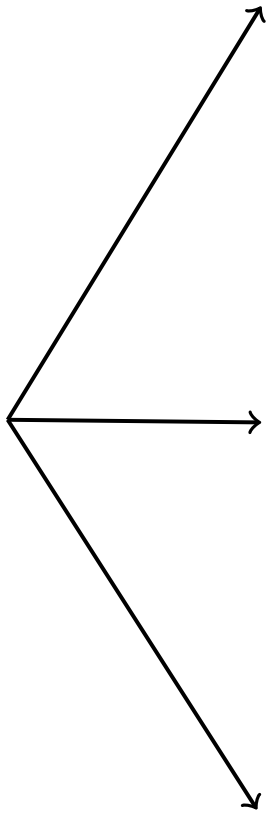
- ... just after the task $<\mathcal{T}_j>$?"
- ... in the portion after the task $<\mathcal{T}_j>$?"
- ... at another time of his/her planning?"

Plan

3 types

One template

"**Why** is Adam **not** performing the plumbing task in addition to his tasks?"



Type (QE_1)

"**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"

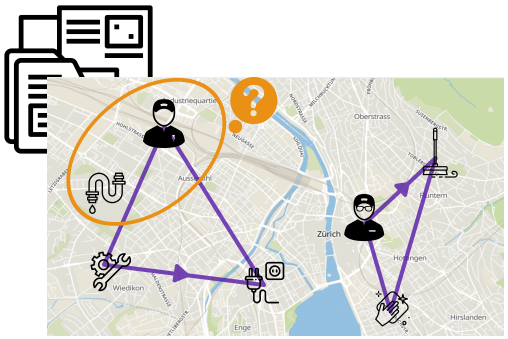
Type (QE_2)

"**Why** is Adam **not** performing the plumbing task in addition to his tasks?"

Type (QE_3)

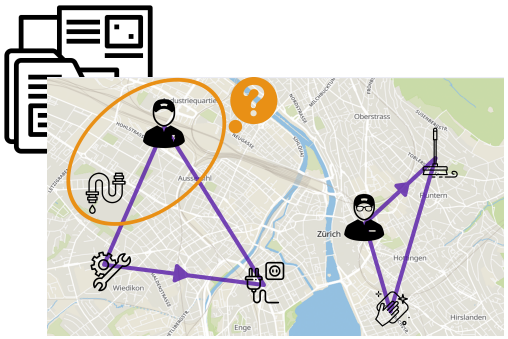
"**How to make it possible** for Adam to perform the plumbing task in addition to his tasks?"

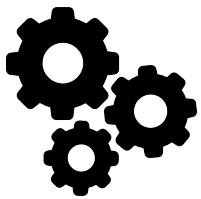
Type (QE_1)



U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"

Type (QE_1)

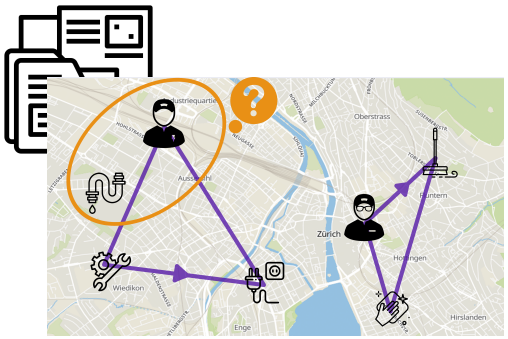




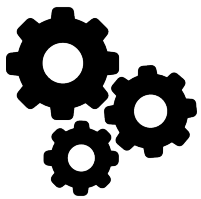
Local Search (LS)
Efficient algorithm applying LS

U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"

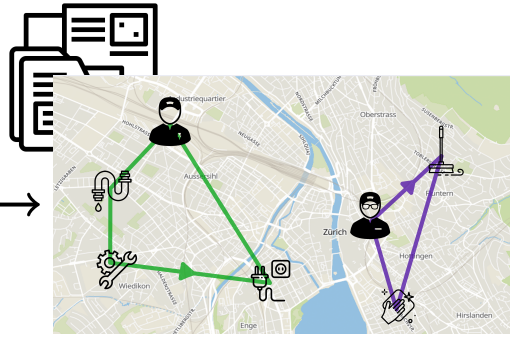
Type (QE_1)



U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"

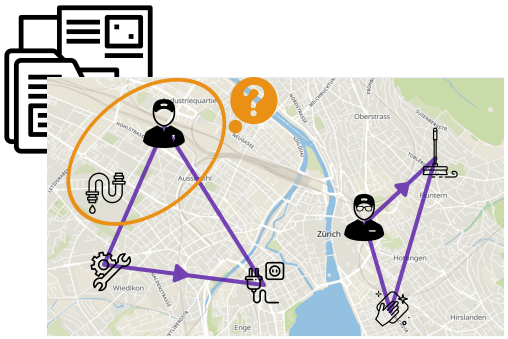


Local Search (LS)
Efficient algorithm applying LS

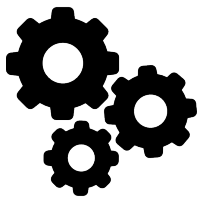


X: "Actually, Adam **could do so**. This new solution would be better."

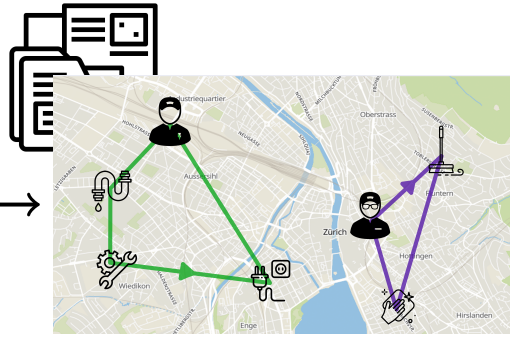
Type (QE_1)



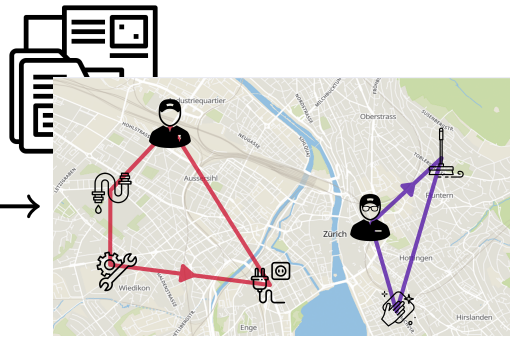
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"



Local Search (LS)
Efficient algorithm applying LS

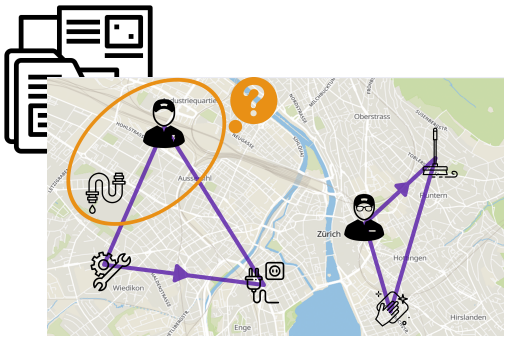


X: "Actually, Adam **could do so**.
This new solution would be better."

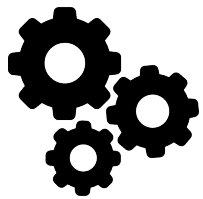


X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 30min."

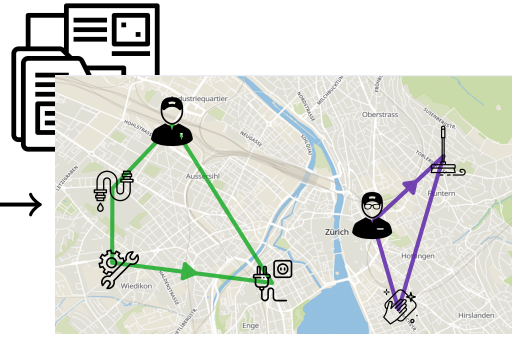
Type (QE_1)



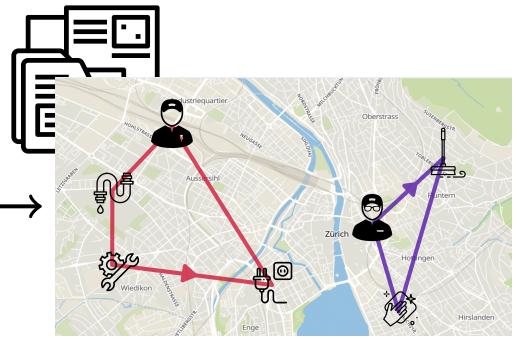
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"



Local Search (LS)
Efficient algorithm applying LS



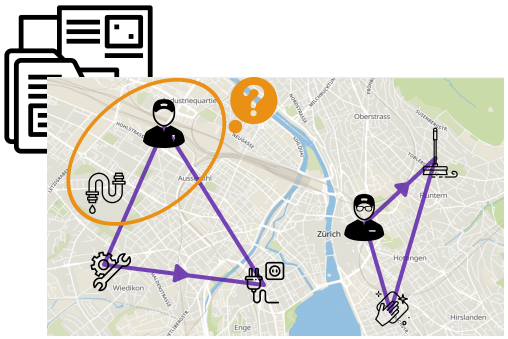
X: "Actually, Adam **could do so**.
This new solution would be better."



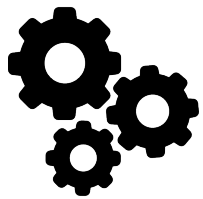
X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 30min."

↪ **Solutions space exploration**

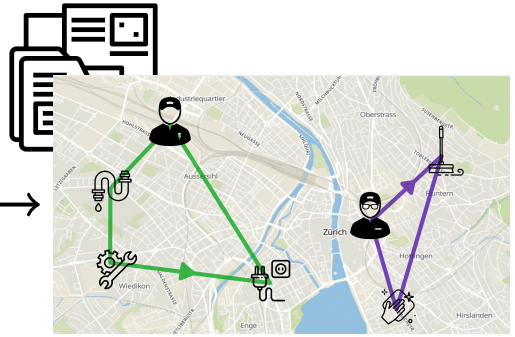
Type (QE_1)



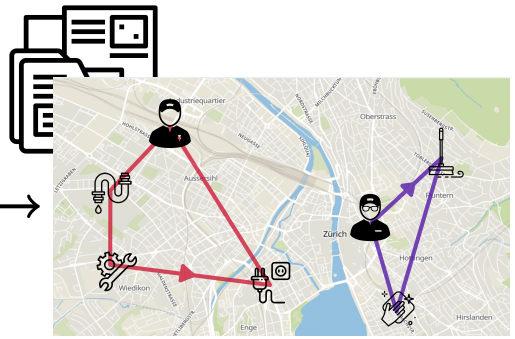
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks - **while keeping the order of his planning**?"



Local Search (LS)
Efficient algorithm applying LS



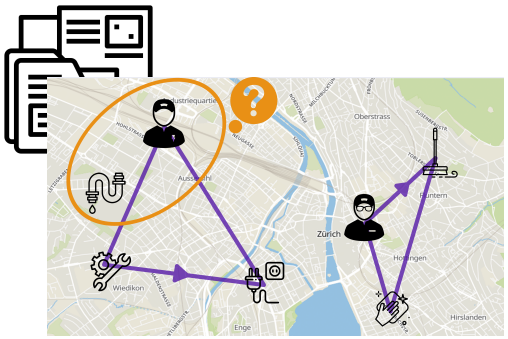
X: "Actually, Adam **could do so**.
This new solution would be better."



X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 30min."

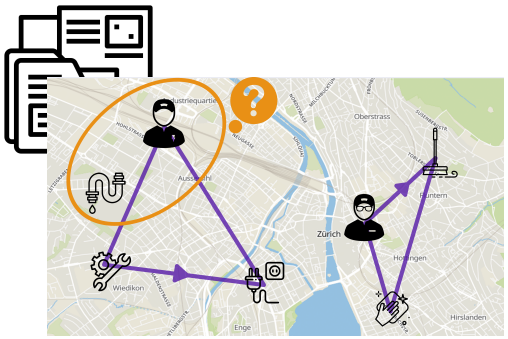
↪ **Solutions space exploration**

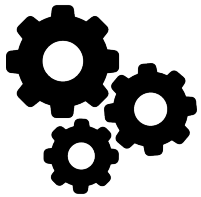
Type (QE_2)



U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"

Type (QE_2)

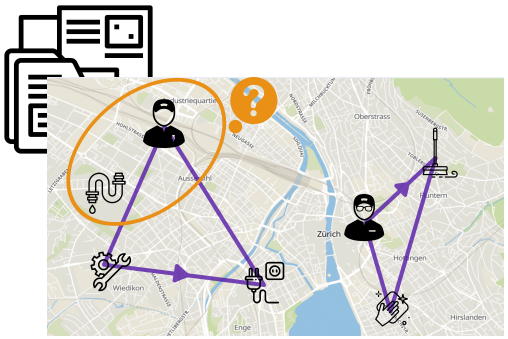




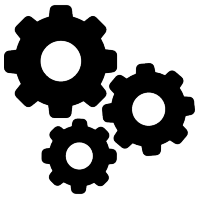
Optimization sub-problem
Integer Linear Programming algorithm

U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"

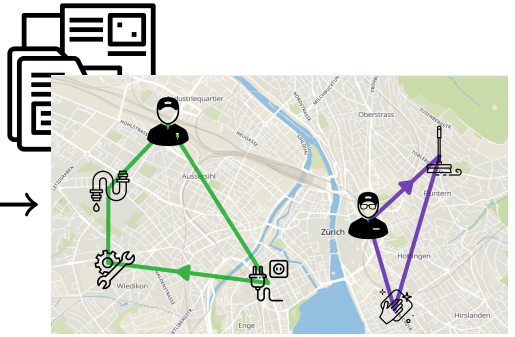
Type (QE_2)



U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"

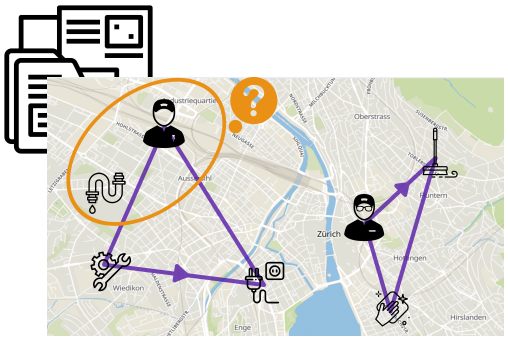


Optimization sub-problem
Integer Linear Programming algorithm

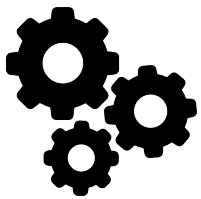


X: "Actually, Adam **could do so**. This new solution would be better."

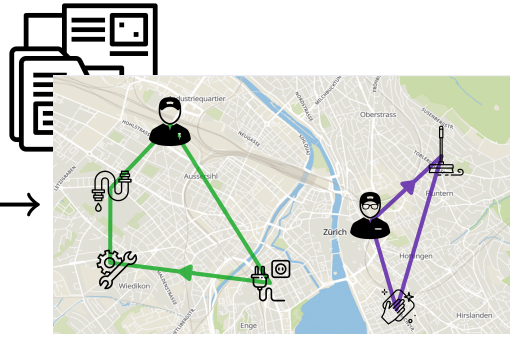
Type (QE_2)



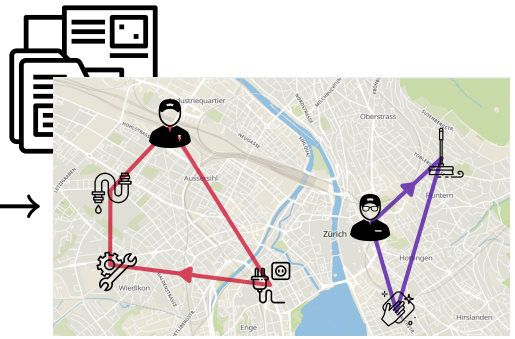
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"



Optimization sub-problem
Integer Linear Programming algorithm

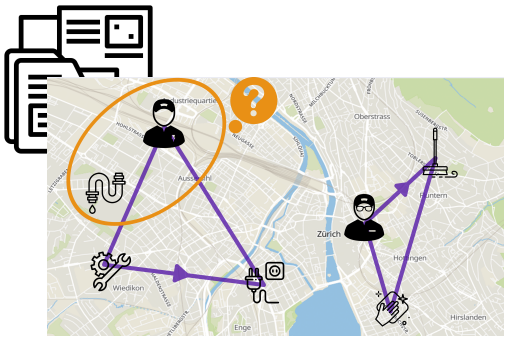


X: "Actually, Adam **could do so**.
This new solution would be better."

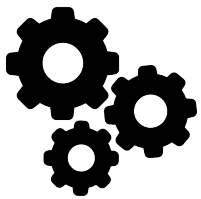


X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 15min."

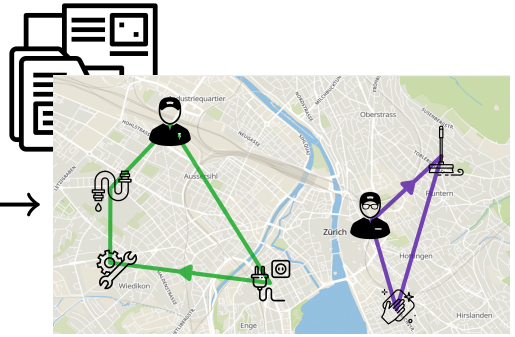
Type (QE_2)



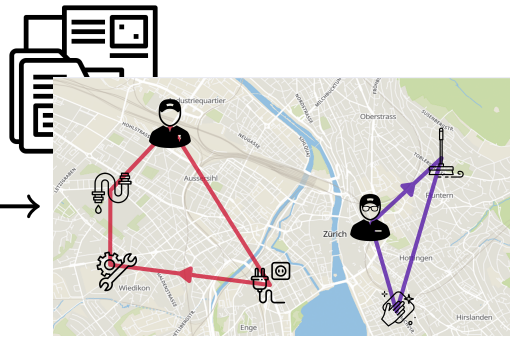
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"



Optimization sub-problem
Integer Linear Programming algorithm



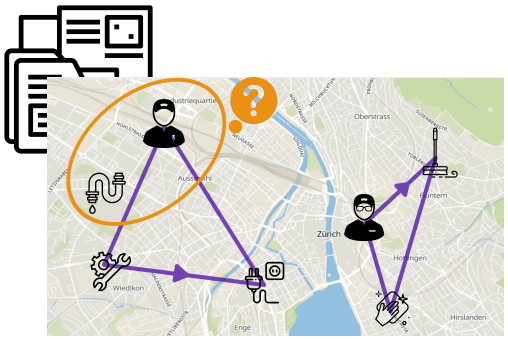
X: "Actually, Adam **could do so**.
This new solution would be better."



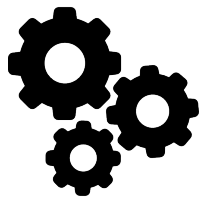
X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 15min."

↪ **Solutions space exploration**

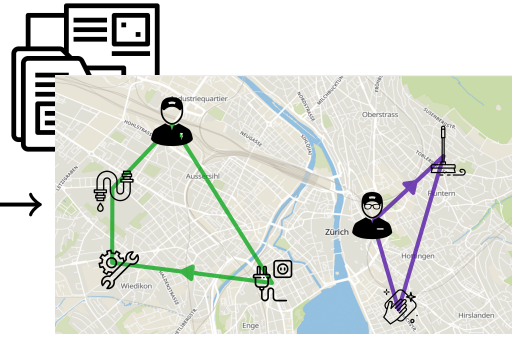
Type (QE_2)



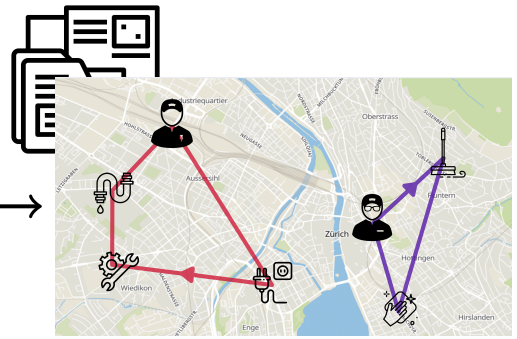
U: "**Why** is Adam **not** performing the plumbing task in addition to his tasks?"



Optimization sub-problem
Integer Linear Programming algorithm



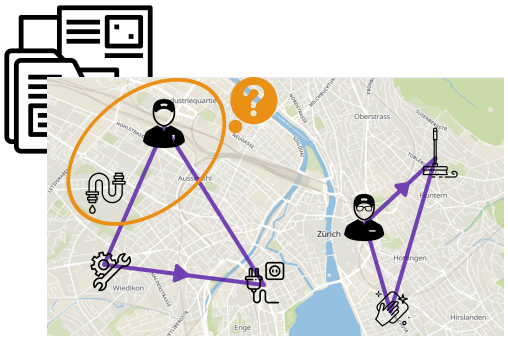
X: "Actually, Adam **could do so**.
This new solution would be better."



X: "Adam **could not do so**.
If he would perform the task, at best, he would be at home late by 15min."

↪ **Solutions space exploration**

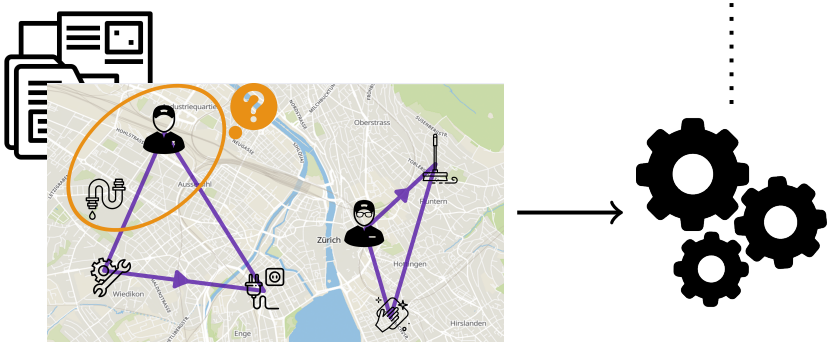
Type (QE_3)



U: "**How to make it possible** for Adam to perform the plumbing task in addition to his tasks?"

Type (QE_3)

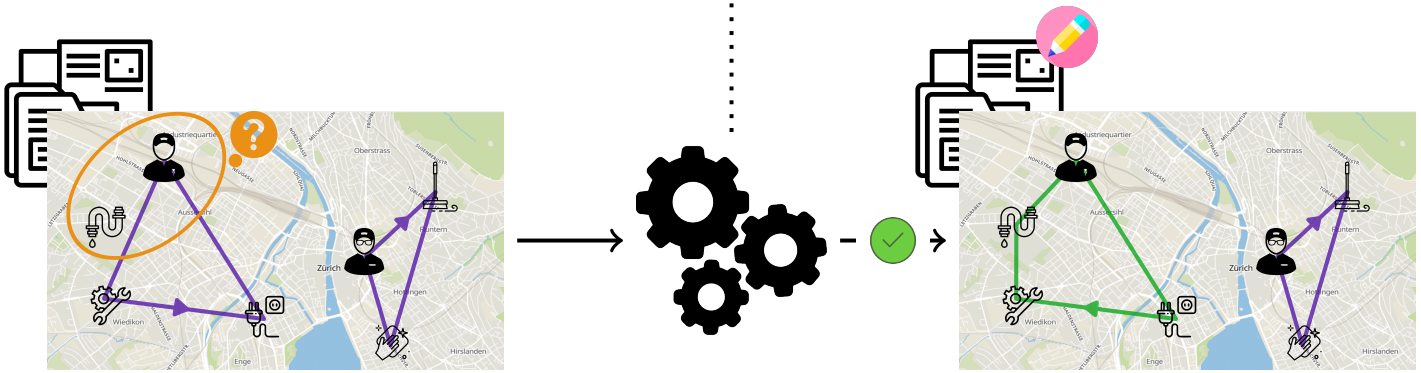
Optimization sub-problem with slacks
Integer Linear Programming algorithm



U: "**How to make it possible** for Adam to perform the plumbing task in addition to his tasks?"

Type (QE_3)

Optimization sub-problem with slacks
Integer Linear Programming algorithm

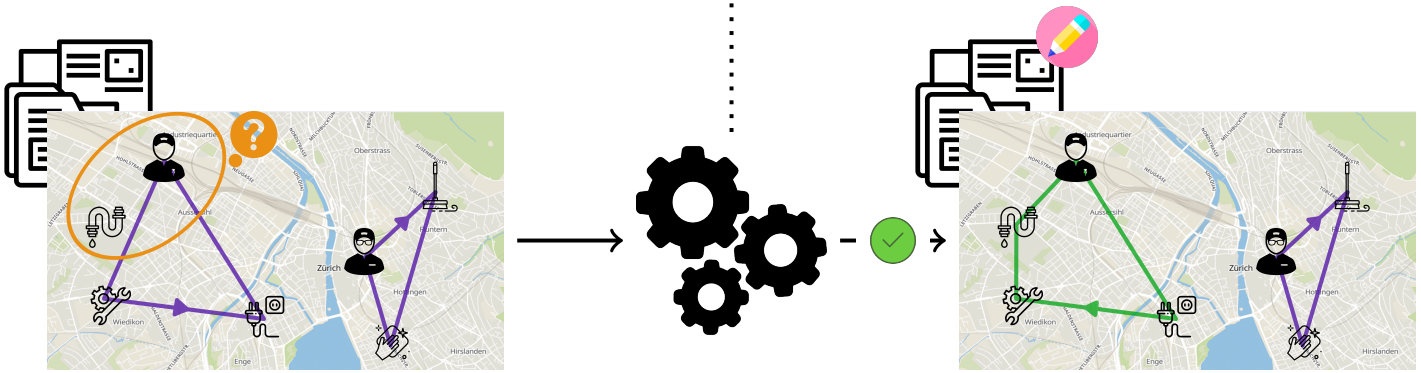


U: "**How to make it possible** for Adam to perform the plumbing task in addition to his tasks?"

X: "**So that** Adam performs it, the electric task should be available 15 minutes earlier."

Type (QE_3)

Optimization sub-problem with slacks
Integer Linear Programming algorithm



U: "**How to make it possible** for Adam to perform the plumbing task in addition to his tasks?"

X: "**So that** Adam performs it, the electric task should be available 15 minutes earlier."

↪ **Instances space exploration**

Plan

A first GUI prototype of WSRP explanation tool

Conclusion

Challenges

- How to deal with **less local / more global** users' questions?
  e.g. "Why Adam is working much less than Ellen?"
- How much **generic** our approach is? How to transpose it to other optimization problems?
- How to **structure the exploration** of solutions and instances?
- How to make the interaction with the user closer to a **dialog**?

References I

[Čyras et al., 2019] Čyras, K., Letsios, D., Misener, R., and Toni, F. (2019).
Argumentation for explainable scheduling.
Proceedings of the AAAI Conference on Artificial Intelligence.

[Korikov et al., 2021] **Korikov, A., Shleyfman, A., and Beck, J. C. (2021).**
Counterfactual explanations for optimization-based decisions in the context of the gdpr.
In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4097–4103. International Joint Conferences on Artificial Intelligence Organization.

References II

[Ludwig et al., 2018] **Ludwig, J., Kalton, A., and Stottler, R. (2018).**
Explaining complex scheduling decisions.
In *IUI Workshops*.

[Miller, 2019] **Miller, T. (2019).**
Explanation in artificial intelligence: Insights from the social sciences.
Artificial Intelligence, 267:1–38.

[Wachter et al., 2018] **Wachter, S., Mittelstadt, B., and Russell, C. (2018).**
Counterfactual explanations without opening the black box: Automated decisions and the gdpr.
Harvard Journal of Law & Technology, 31.