Generating explanations of various types for end-users of optimization systems

Application to the workforce scheduling and routing problem

Mathieu Lerouge
mathieulerouge.github.io
MICS, CentraleSupélec Université Paris-Saclay

ROADEF
21 February 2023

Plan

Plan

Plan

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

Let see (1), (2) and (3) in our use case.

Motivations

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

Let see (1), (2) and (3) in our use case.

Motivations

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

Let see (1), (2) and (3) in our use case.

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

Let see (1), (2) and (3) in our use case.

Motivations

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

Let see (1), (2) and (3) in our use case.

Motivations

3 observations:

(1) Real-world situations modeled as **Combinatorial Optimization (CO) problems** (*e.g.* workforce management);

(2) CO problems solved using **optimization systems** that are developed by experts (*e.g.* DecisionBrain);

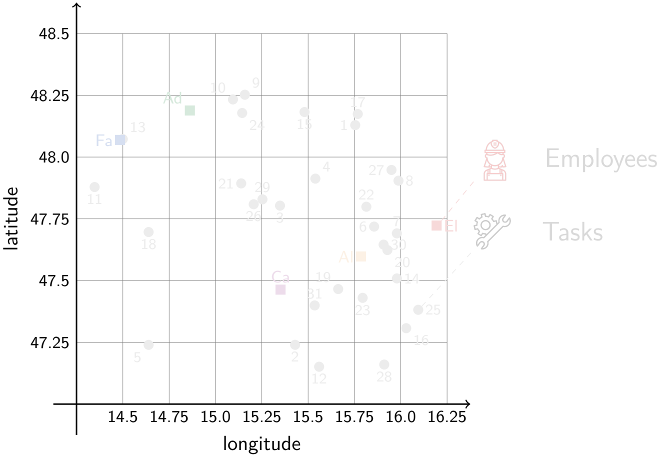
(3) Optimization systems are used as **black boxes** by **non-expert people** to make decisions.

↪ End-users may experience a **lack of trust and confidence**.

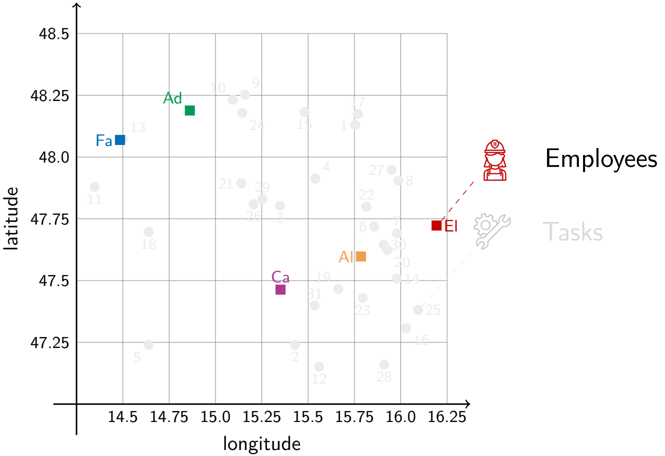
Let see (1), (2) and (3) in our use case.

Plan

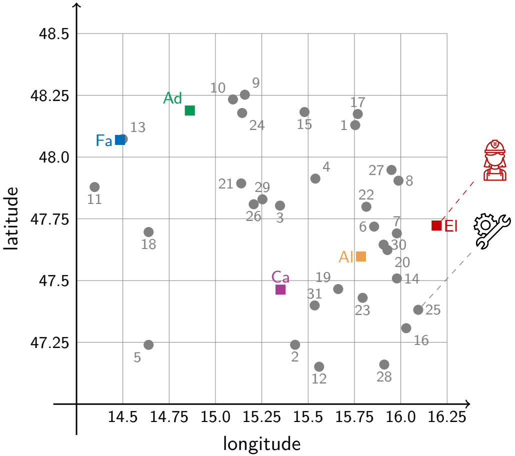
Workforce Scheduling and Routing Problem (WSRP):



Workforce Scheduling and Routing Problem (WSRP):



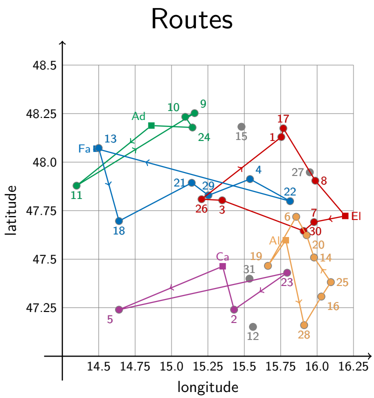
Workforce Scheduling and Routing Problem (WSRP):



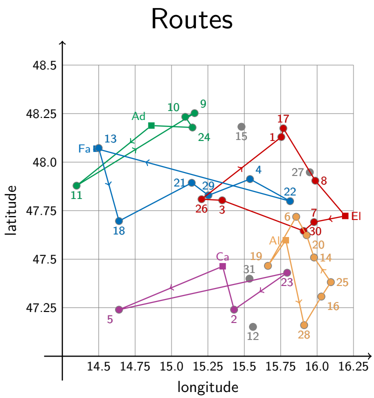
Employees

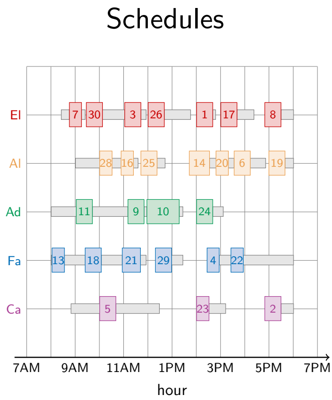
Tasks

Workforce Scheduling and Routing Problem (WSRP):



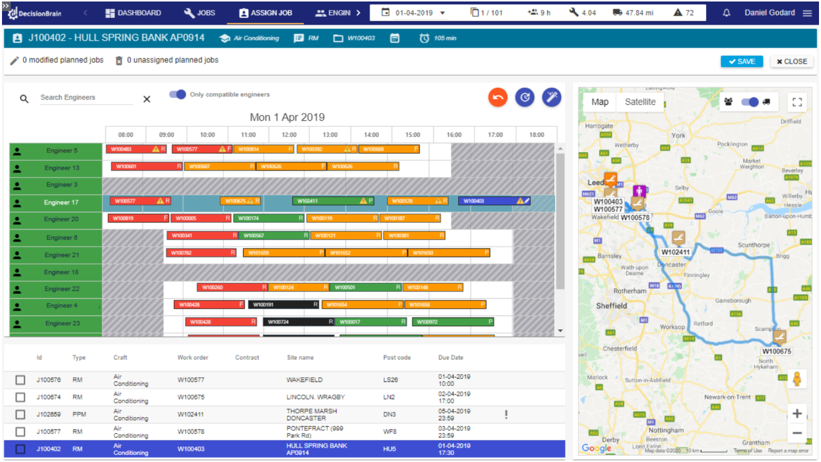
Workforce Scheduling and Routing Problem (WSRP):



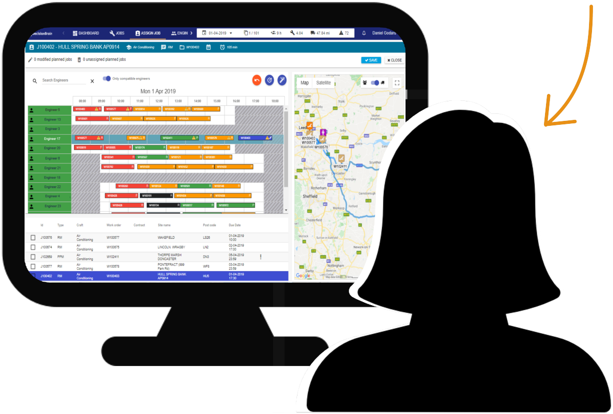


WSRP-solving system:

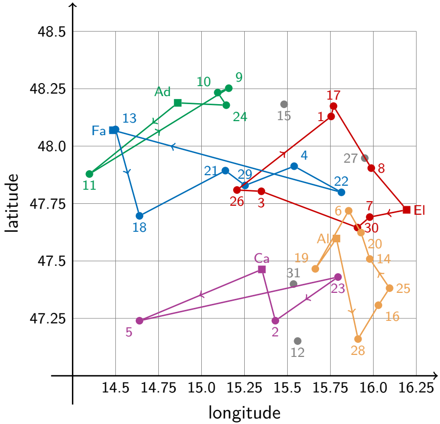
e.g DecisionBrain's Dynamic Scheduler



Planner:


DASHBOARD
JOBS
ASSIGN JOB
J100402 - HULL SPRING BANK AP0914
SAVE
CLOSE
Mon 1 Apr 2019
Map
Satellite

A problematic situation for a planner:



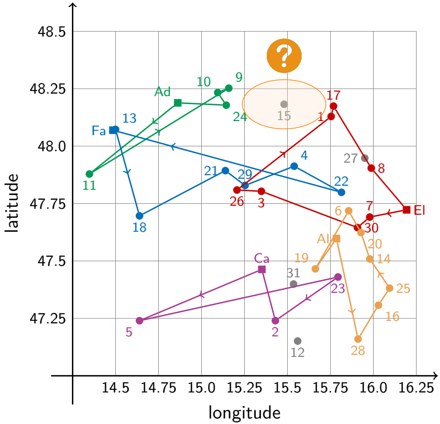
"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

A problematic situation for a planner:



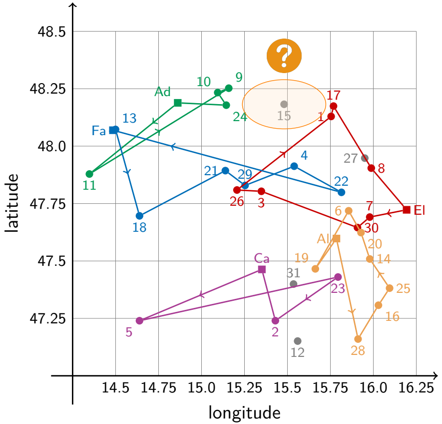
"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

A problematic situation for a planner:



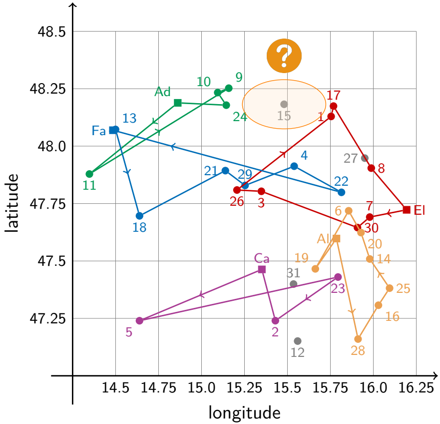
"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

A problematic situation for a planner:



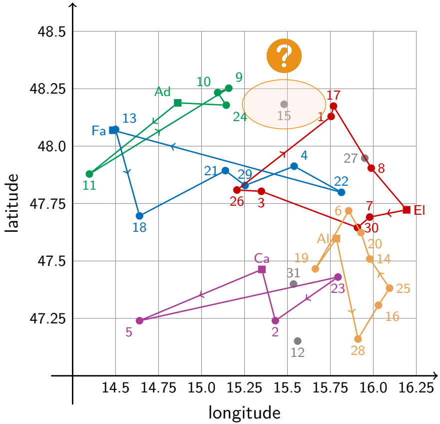
"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

A problematic situation for a planner:



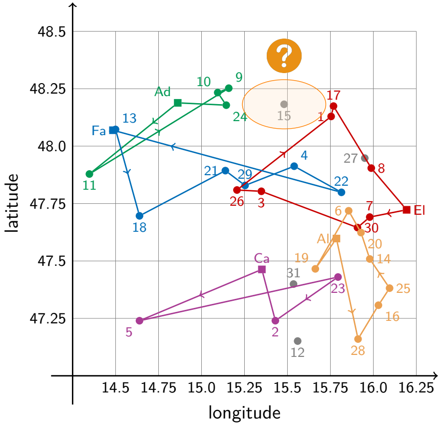
"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

A problematic situation for a planner:



"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

Plan

Our proposition

Tackling the lack of trust and confidence experienced by non-expert end-users solving **WSRP** instances, by generating **explanations** to help them better understand their solutions.

Our proposition

Tackling the lack of trust and confidence experienced by non-expert end-users solving **WSRP** instances, by generating **explanations** to help them better understand their solutions.

Our proposition

Tackling the lack of trust and confidence experienced by non-expert end-users solving **WSRP** instances, by generating **explanations** to help them better understand their solutions.

Plan

Plan

Literature on eXplainable Artificial Intelligence (XAI):

Works on explanations:

- **Many** in **Machine Learning** [Barredo Arrieta et al., 2020].
- **Some** in **other AI fields** including
  - Expert Systems, [Wick and Thompson, 1992],
  - Planning, *e.g.* [Chakraborti et al., 2020],
  - Constraint Programming, *e.g.* [Junker, 2004].
- **Few** ones in **Combinatorial Optimization (CO)**, *e.g.* [Korikov et al., 2021].

↪ Survey **concepts about explanations** in AI fields other than CO and transpose them to CO.

[Barredo Arrieta et al., 2020] XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI
[Wick and Thompson, 1992] Reconstructive Expert System explanation
[Chakraborti et al., 2020] The emerging landscape of explainable AI planning and decision making
[Junker, 2004] QuickXplain: preferred explanations and relaxations for over-constrained problems
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Literature on eXplainable Artificial Intelligence (XAI):

Works on explanations:

- **Many** in **Machine Learning** [Barredo Arrieta et al., 2020].
- **Some** in **other AI fields** including
  - Expert Systems, [Wick and Thompson, 1992],
  - Planning, *e.g.* [Chakraborti et al., 2020],
  - Constraint Programming, *e.g.* [Junker, 2004].
- **Few** ones in **Combinatorial Optimization (CO)**, *e.g.* [Korikov et al., 2021].

↪ Survey **concepts about explanations** in AI fields other than CO and transpose them to CO.

[Barredo Arrieta et al., 2020] XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI
[Wick and Thompson, 1992] Reconstructive Expert System explanation
[Chakraborti et al., 2020] The emerging landscape of explainable AI planning and decision making
[Junker, 2004] QuickXplain: preferred explanations and relaxations for over-constrained problems
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Literature on eXplainable Artificial Intelligence (XAI):

Works on explanations:

- **Many** in **Machine Learning** [Barredo Arrieta et al., 2020].
- **Some** in **other AI fields** including
  - Expert Systems, [Wick and Thompson, 1992],
  - Planning, *e.g.* [Chakraborti et al., 2020],
  - Constraint Programming, *e.g.* [Junker, 2004].
- **Few** ones in **Combinatorial Optimization (CO)**, *e.g.* [Korikov et al., 2021].

↪ Survey **concepts about explanations** in AI fields other than CO and transpose them to CO.

[Barredo Arrieta et al., 2020] XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI
[Wick and Thompson, 1992] Reconstructive Expert System explanation
[Chakraborti et al., 2020] The emerging landscape of explainable AI planning and decision making
[Junker, 2004] QuickXplain: preferred explanations and relaxations for over-constrained problems
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Literature on eXplainable Artificial Intelligence (XAI):

Works on explanations:

- **Many** in **Machine Learning** [Barredo Arrieta et al., 2020].
- **Some** in **other AI fields** including
  - Expert Systems, [Wick and Thompson, 1992],
  - Planning, *e.g.* [Chakraborti et al., 2020],
  - Constraint Programming, *e.g.* [Junker, 2004].
- **Few** ones in **Combinatorial Optimization (CO)**, *e.g.* [Korikov et al., 2021].

↪ Survey **concepts about explanations** in AI fields other than CO and transpose them to CO.

[Barredo Arrieta et al., 2020] XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI
[Wick and Thompson, 1992] Reconstructive Expert System explanation
[Chakraborti et al., 2020] The emerging landscape of explainable AI planning and decision making
[Junker, 2004] QuickXplain: preferred explanations and relaxations for over-constrained problems
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Explanations in Artificial Intelligence

Literature on eXplainable Artificial Intelligence (XAI):

Works on explanations:

- **Many** in **Machine Learning** [Barredo Arrieta et al., 2020].
- **Some** in **other AI fields** including
  - Expert Systems, [Wick and Thompson, 1992],
  - Planning, *e.g.* [Chakraborti et al., 2020],
  - Constraint Programming, *e.g.* [Junker, 2004].
- **Few** ones in **Combinatorial Optimization (CO)**, *e.g.* [Korikov et al., 2021].

↪ Survey **concepts about explanations** in AI fields other than CO and transpose them to CO.

[Barredo Arrieta et al., 2020] XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI
[Wick and Thompson, 1992] Reconstructive Expert System explanation
[Chakraborti et al., 2020] The emerging landscape of explainable AI planning and decision making
[Junker, 2004] QuickXplain: preferred explanations and relaxations for over-constrained problems
[Korikov et al., 2021] Counterfactual Explanations for Optimization-Based Decisions in the Context of the GDPR

Some recurrent concepts in XAI methods (1/2):

Explanations are often:

- **local**, *i.e.* focusing on outputs generated by the AI system [Wick and Thompson, 1992];
- expressed as texts using **templates**, *e.g.* [Krarup et al., 2021];
- **contrastive** *i.e.* answering questions having the following form [Lipton, 1990]:

  "Why not that $\underbrace{\text{other result}}_{\text{alternative}}$ instead of this $\underbrace{\text{current one}}_{\text{fact}}$?"

[Wick and Thompson, 1992] Reconstructive expert system explanation
[Krarup et al., 2021] Contrastive explanations of plans through model restrictions
[Lipton, 1990] Contrastive explanation

Some recurrent concepts in XAI methods (1/2):

Explanations are often:

- **local**, *i.e.* focusing on outputs generated by the AI system [Wick and Thompson, 1992];
- expressed as texts using **templates**, *e.g.* [Krarup et al., 2021];
- **contrastive** *i.e.* answering questions having the following form [Lipton, 1990]:

  "Why not that $\underbrace{\text{other result}}_{\text{alternative}}$ instead of this $\underbrace{\text{current one}}_{\text{fact}}$?"

[Wick and Thompson, 1992] Reconstructive expert system explanation
[Krarup et al., 2021] Contrastive explanations of plans through model restrictions
[Lipton, 1990] Contrastive explanation

Some recurrent concepts in XAI methods (1/2):

Explanations are often:

- **local**, *i.e.* focusing on outputs generated by the AI system [Wick and Thompson, 1992];
- expressed as texts using **templates**, *e.g.* [Krarup et al., 2021];
- **contrastive** *i.e.* answering questions having the following form [Lipton, 1990]:

  “Why not that $\underbrace{\text{other result}}_{\text{alternative}}$ instead of this $\underbrace{\text{current one}}_{\text{fact}}$?”

[Wick and Thompson, 1992] Reconstructive expert system explanation
[Krarup et al., 2021] Contrastive explanations of plans through model restrictions
[Lipton, 1990] Contrastive explanation

Some recurrent concepts in XAI methods (1/2):

Explanations are often:

- **local**, *i.e.* focusing on outputs generated by the AI system [Wick and Thompson, 1992];
- expressed as texts using **templates**, *e.g.* [Krarup et al., 2021];
- **contrastive** *i.e.* answering questions having the following form [Lipton, 1990]:

  "Why not that $\underbrace{\text{other result}}_{\text{alternative}}$ instead of this $\underbrace{\text{current one}}_{\text{fact}}$?"

[Wick and Thompson, 1992] Reconstructive expert system explanation
[Krarup et al., 2021] Contrastive explanations of plans through model restrictions
[Lipton, 1990] Contrastive explanation

Some recurrent concepts in XAI methods (2/2):

Explanations are often:

- **scenario**, *i.e.* describing how **changes in inputs data** or in model parameters [...] **affect** the outputs [Mohseni et al., 2021]
  - ↪ ≃ answering "What if ... ? Would it be ... ?" questions;
- **counterfactual**, *i.e.* presenting **alterations in inputs data** that would have resulted in a different outputs, such as an **end-user-specified outputs** [Wachter et al., 2018]
  - ↪ ≃ answering "How to ... ?" questions.

[Mohseni et al., 2021] A multidisciplinary survey and framework for design and evaluation of XAI systems
[Wachter et al., 2018] Counterfactual explanations w/o opening the black box: automated decisions and the GDPR

Some recurrent concepts in XAI methods (2/2):

Explanations are often:

- **scenario**, *i.e.* describing how **changes in inputs data** or in model parameters [...] **affect** the outputs [Mohseni et al., 2021]
  ↪ $\simeq$ answering "What if ... ? Would it be ... ?" questions;
- **counterfactual**, *i.e.* presenting **alterations in inputs data** that would have resulted in a different outputs, such as an **end-user-specified outputs** [Wachter et al., 2018]
  ↪ $\simeq$ answering "How to ... ?" questions.

[Mohseni et al., 2021] A multidisciplinary survey and framework for design and evaluation of XAI systems
[Wachter et al., 2018] Counterfactual explanations w/o opening the black box: automated decisions and the GDPR

Plan

Our proposition (with more details)

Tackling the lack of trust and confidence experienced by non-expert end-users solving WSRP instances, by generating **explanations** to help them better understand their solutions.

Generated **explanations** are:

- focusing **on a given WSRP solution**,
- expressed as **texts** using **templates**,
- **contrastive** ("Why not ...?"), **scenario** ("What if ...?") or **counterfactual** ("How to ...?").

Our proposition (with more details)

Tackling the lack of trust and confidence experienced by non-expert end-users solving WSRP instances, by generating **explanations** to help them better understand their solutions.

Generated **explanations** are:

- focusing **on a given WSRP solution**,
- expressed as **texts** using **templates**,
- **contrastive** ("Why not ...?"), **scenario** ("What if ...?") or **counterfactual** ("How to ...?").

Our proposition (with more details)

Tackling the lack of trust and confidence experienced by non-expert end-users solving WSRP instances, by generating **explanations** to help them better understand their solutions.

Generated **explanations** are:

- focusing **on a given WSRP solution**,
- expressed as **texts** using **templates**,
- **contrastive** ("Why not ...?"), **scenario** ("What if ...?") or **counterfactual** ("How to ...?").

Our proposition (with more details)

Tackling the lack of trust and confidence experienced by non-expert end-users solving WSRP instances, by generating **explanations** to help them better understand their solutions.

Generated **explanations** are:

- focusing **on a given WSRP solution**,
- expressed as **texts** using **templates**,
- **contrastive** ("Why not ...?"), **scenario** ("What if ...?") or **counterfactual** ("How to ...?").

Our proposition (with more details)

Tackling the lack of trust and confidence experienced by non-expert end-users solving WSRP instances, by generating **explanations** to help them better understand their solutions.

Generated **explanations** are:

- focusing **on a given WSRP solution**,
- expressed as **texts** using **templates**,
- **contrastive** (“Why not ...?”), **scenario** (“What if ...?”) or **counterfactual** (“How to ...?”).

Plan

Plan

Observations about a solution

List of possible observations:

We identify **16 possible observations** based on **templates**, about **various desired changes** in the solution:

- **adding** a task in an employee route;
  e.g. "⟨employee i⟩ is not performing ⟨task j⟩ ...
  - ... just after ⟨task k⟩?"
  - ... in addition to the tasks of their route?"
- **swapping** two tasks outside - inside a route;
- **changing of the order** of tasks in a route.

Observations about a solution

List of possible observations:

We identify **16 possible observations** based on **templates**, about **various desired changes** in the solution:

- **adding** a task in an employee route;
  e.g. "⟨employee i⟩ is not performing ⟨task j⟩ ...
  - ... just after ⟨task k⟩?"
  - ... in addition to the tasks of their route?"
- **swapping** two tasks outside - inside a route;
- **changing of the order** of tasks in a route.

Observations about a solution

List of possible observations:

We identify **16 possible observations** based on **templates**, about **various desired changes** in the solution:

- **adding** a task in an employee route;
  e.g. "$\langle$employee $i\rangle$ is not performing $\langle$task $j\rangle$ ...
  - ... just after $\langle$task $k\rangle$?"
  - ... in addition to the tasks of their route?"
- **swapping** two tasks outside - inside a route;
- **changing of the order** of tasks in a route.

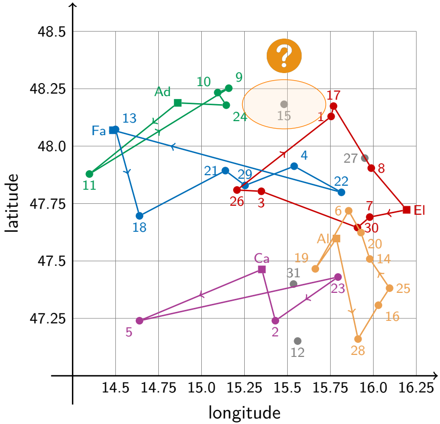
Observations about a solution

List of possible observations:

We identify **16 possible observations** based on **templates**, about **various desired changes** in the solution:

- **adding** a task in an employee route;
  e.g. "⟨employee i⟩ is not performing ⟨task j⟩ ...
  - ... just after ⟨task k⟩?"
  - ... in addition to the tasks of their route?"
- **swapping** two tasks outside - inside a route;
- **changing of the order** of tasks in a route.

A problematic situation for a planner:



"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

Plan

From observations to questions

From 1 observation, 3 questions for the end-user:

From an observation, *e.g.* "$\langle$employee $i\rangle$ is not performing $\langle$task $j\rangle$ in addition to the tasks of their route?", we can build:

- a **contrastive** question,
  "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **scenario** question,
  "What if $\langle$changes in the instance parameters$\rangle$?
  Would $\langle$employee $i\rangle$ be performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **counterfactual** question,
  "How to make $\langle$employee $i\rangle$ perform $\langle$task $j\rangle$
  in addition to the tasks of their route?"

From observations to questions

From 1 observation, 3 questions for the end-user:

From an observation, *e.g.* "$\langle$employee $i\rangle$ is not performing $\langle$task $j\rangle$ in addition to the tasks of their route?", we can build:

- a **contrastive** question,
  "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **scenario** question,
  "What if $\langle$changes in the instance parameters$\rangle$?
  Would $\langle$employee $i\rangle$ be performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **counterfactual** question,
  "How to make $\langle$employee $i\rangle$ perform $\langle$task $j\rangle$
  in addition to the tasks of their route?"

From observations to questions

From 1 observation, 3 questions for the end-user:

From an observation, *e.g.* "$\langle$employee $i\rangle$ is not performing $\langle$task $j\rangle$ in addition to the tasks of their route?", we can build:

- a **contrastive** question,
  "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **scenario** question,
  "What if $\langle$changes in the instance parameters$\rangle$?
  Would $\langle$employee $i\rangle$ be performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **counterfactual** question,
  "How to make $\langle$employee $i\rangle$ perform $\langle$task $j\rangle$
  in addition to the tasks of their route?"

From observations to questions

From 1 observation, 3 questions for the end-user:

From an observation, *e.g.* "$\langle$employee $i\rangle$ is not performing $\langle$task $j\rangle$ in addition to the tasks of their route?", we can build:

- a **contrastive** question,
  "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **scenario** question,
  "What if $\langle$changes in the instance parameters$\rangle$?
  Would $\langle$employee $i\rangle$ be performing $\langle$task $j\rangle$
  in addition to the tasks of their route?"
- a **counterfactual** question,
  "How to make $\langle$employee $i\rangle$ perform $\langle$task $j\rangle$
  in addition to the tasks of their route?"

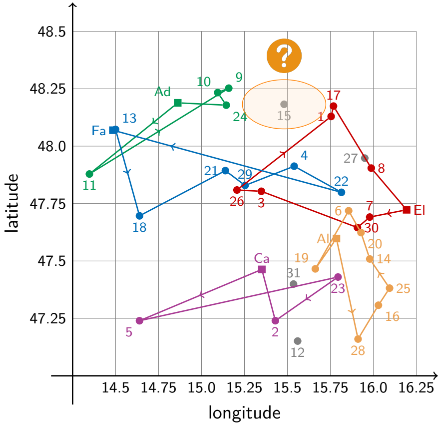
From observations to questions

From 1 observation, 3 questions for the end-user:

From an observation, *e.g.* "⟨employee i⟩ is not performing ⟨task j⟩ in addition to the tasks of their route?", we can build:

- a **contrastive** question,
  "Why is ⟨employee i⟩ not performing ⟨task j⟩
  in addition to the tasks of their route?"
- a **scenario** question,
  "What if ⟨changes in the instance parameters⟩?
  Would ⟨employee i⟩ be performing ⟨task j⟩
  in addition to the tasks of their route?"
- a **counterfactual** question,
  "How to make ⟨employee i⟩ perform ⟨task j⟩
  in addition to the tasks of their route?"

A problematic situation for a planner:



"Ellen is not performing task 15 in addition to the tasks of her route..."

"Why is Ellen not performing task 15 in addition to the tasks of her route?"

"How to make Ellen perform task 15 in addition to the tasks of her route?"

↪ **If no explanations**, then lack of trust and confidence...

Plan

From questions to computation

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is ⟨employee i⟩ not performing ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

  - either **polynomial algorithms**,
  - or **mathematical programming**.

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is ⟨employee i⟩ not performing ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one;
  if not, extract information for why.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):
  - either **polynomial algorithms**,
  - or **mathematical programming**.

From questions to computation

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is ⟨employee i⟩ not performing ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

- either **polynomial algorithms**,
- or **mathematical programming**.

From questions to computation

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is ⟨employee i⟩ not performing ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

- either **polynomial algorithms**,
- or **mathematical programming**.

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

- either **polynomial algorithms**,
- or **mathematical programming**.

From questions to computation

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is ⟨employee i⟩ not performing ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

- either **polynomial algorithms**,
- or **mathematical programming**.

From questions to computation

Computation related to a contrastive question:

Consider a contrastive question, *e.g.* "Why is $\langle$employee $i\rangle$ not performing $\langle$task $j\rangle$ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one**, *e.g.* the solutions obtained by inserting j in the route of i and choosing a permutation of the tasks in this route.
- To answer the question, we must test if these solutions are **feasible and better** than the current one; **if not, extract information for why**.

↪ We build **algorithms for checking** solutions feasibility and improvement which are (depending on the question):

- either **polynomial algorithms**,
- or **mathematical programming**.

From questions to computation

Computation related to a scenario question:

Consider a scenario question, *e.g.* "What if ⟨changes in the instance parameters⟩? Would ⟨employee i⟩ be performing ⟨task j⟩ in addition to the tasks of their route?"

- The reasoning is the same as in the contrastive case: the user implicitly defines **solutions neighboring the current one**, but for an **instance** that is slightly **different**.
- To answer the question, we must test if these solutions are **feasible and better** than the current one, relatively to the **new instance**; **if not, extract why**.

↪ We use the **same algorithms** as for the contrastive case.

From questions to computation

Computation related to a scenario question:

Consider a scenario question, *e.g.* "What if ⟨changes in the instance parameters⟩? Would ⟨employee i⟩ be performing ⟨task j⟩ in addition to the tasks of their route?"

- The reasoning is the same as in the contrastive case: the user implicitly defines **solutions neighboring the current one**, but for an **instance** that is slightly **different**.
- To answer the question, we must test if these solutions are **feasible and better** than the current one, relatively to the **new instance**; **if not, extract why**.

↪ We use the **same algorithms** as for the contrastive case.

Computation related to a scenario question:

Consider a scenario question, *e.g.* “What if ⟨changes in the instance parameters⟩? Would ⟨employee i⟩ be performing ⟨task j⟩ in addition to the tasks of their route?”

- The reasoning is the same as in the contrastive case: the user implicitly defines **solutions neighboring the current one**, but for an **instance** that is slightly **different**.
- To answer the question, we must test if these solutions are **feasible and better** than the current one, relatively to the **new instance**; **if not, extract why**.

↪ We use the **same algorithms** as for the contrastive case.

Computation related to a scenario question:

Consider a scenario question, *e.g.* “What if ⟨changes in the instance parameters⟩? Would ⟨employee i⟩ be performing ⟨task j⟩ in addition to the tasks of their route?”

- The reasoning is the same as in the contrastive case: the user implicitly defines **solutions neighboring the current one**, but for an **instance** that is slightly **different**.
- To answer the question, we must test if these solutions are **feasible and better** than the current one, relatively to the **new instance**; **if not, extract why**.

↪ We use the **same algorithms** as for the contrastive case.

Computation related to a counterfactual question:

Consider a counterfactual qu., *e.g.* "How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one** and allow **alterations of the instance parameters**.
- To answer their question, we must test if these solutions are **feasible and better** than the current one, relatively to a **set of new instances**; **if not, extract why**.

↪ We use **mathematical programming** for exploring the neighboring solutions while altering the instance parameters.

From questions to computation

Computation related to a counterfactual question:

Consider a counterfactual qu., *e.g.* "How to make $\langle$employee $i\rangle$ perform $\langle$task $j\rangle$ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one** and allow **alterations of the instance parameters**.
- To answer their question, we must test if these solutions are **feasible and better** than the current one, relatively to a **set of new instances**; **if not, extract why**.

↪ We use **mathematical programming** for exploring the neighboring solutions while altering the instance parameters.

Computation related to a counterfactual question:

Consider a counterfactual qu., *e.g.* "How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one** and allow **alterations of the instance parameters**.
- To answer their question, we must test if these solutions are **feasible and better** than the current one, relatively to a **set of new instances**; **if not, extract why**.

↪ We use **mathematical programming** for exploring the neighboring solutions while altering the instance parameters.

Computation related to a counterfactual question:

Consider a counterfactual qu., *e.g.* "How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"

- Through their question, the end-user implicitly defines interesting **solutions neighboring the current one** and allow **alterations of the instance parameters**.
- To answer their question, we must test if these solutions are **feasible and better** than the current one, relatively to a **set of new instances**; **if not, extract why**.

↪ We use **mathematical programming** for exploring the neighboring solutions while altering the instance parameters.

Plan

From computation to explanations

Providing an explanation as a text:

We **ran the algorithm** associated with a question of any type.

We **fill explanation template texts** with values from the result.

Example of explanation text:

"How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"

"By ⟨changing the instance parameters as follow based on the algorithm result⟩,
⟨the desired observation⟩ would be possible;
in this case, the solution would be ⟨the one deduced from the algorithm result⟩."

From computation to explanations

Providing an explanation as a text:

We **ran the algorithm** associated with a question of any type.
We **fill explanation template texts** with values from the result.

Example of explanation text:

"How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"

"By ⟨changing the instance parameters as follow based on the algorithm result⟩,
⟨the desired observation⟩ would be possible;
in this case, the solution would be ⟨the one deduced from the algorithm result⟩."

From computation to explanations

Providing an explanation as a text:

We **ran the algorithm** associated with a question of any type.
We **fill explanation template texts** with values from the result.

Example of explanation text:

"How to make ⟨employee i⟩ perform ⟨task j⟩ in addition to the tasks of their route?"
"By ⟨changing the instance parameters as follow based on the algorithm result⟩,
⟨the desired observation⟩ would be possible;
in this case, the solution would be ⟨the one deduced from the algorithm result⟩."

Plan

Planner looking for explanations (1/3):

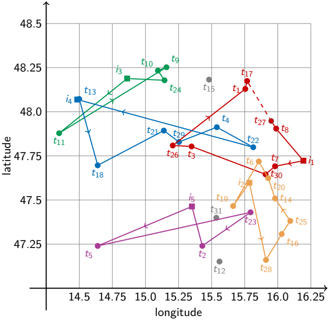
- Why is Ellen not performing t_{27} just after t_{17}?
- If Ellen performs t_{27} just after t_{17}, then she would end t_{27} at the earliest at 4:37PM while t_{27} is not available after 3:00PM.
  Therefore Ellen is not performing t_{27} just after t_{17}.

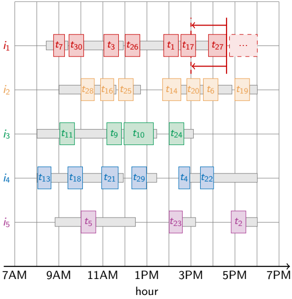
Planner looking for explanations (1/3):

- Why is Ellen not performing t_{27} just after t_{17}?
- If Ellen performs t_{27} just after t_{17}, then she would end t_{27} at the earliest at 4:37PM while t_{27} is not available after 3:00PM.
  Therefore Ellen is not performing t_{27} just after t_{17}.

Planner looking for explanations (1/3):

- Why is Ellen not performing t_{27} just after t_{17}?
- If Ellen performs t_{27} just after t_{17}, then she would end t_{27} at the earliest at 4:37PM while t_{27} is not available after 3:00PM.
  Therefore Ellen is not performing t_{27} just after t_{17}.





Planner looking for explanations (2/3):

- What if t_{27} was available up to 4:37PM (instead of 3:00PM)?
- If t_{27} was available until 4:37PM, then Ellen would be able to perform t_{27} during its availability time-window.
  However, in the following steps of her route, Ellen would start t_8 at the earliest at 4:44PM while t_8 is not available after 4:40PM.
  Therefore, it would still not make Ellen perform t_{27} just after t_{17}.

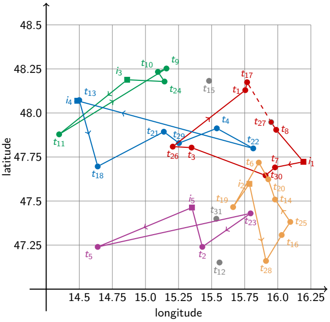
Example or demo

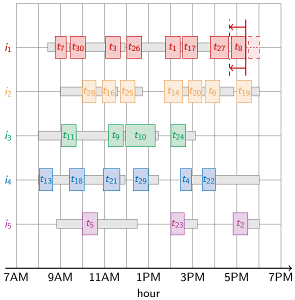
Planner looking for explanations (2/3):

- What if t_{27} was available up to 4:37PM (instead of 3:00PM)?
- If t_{27} was available until 4:37PM, then Ellen would be able to perform t_{27} during its availability time-window.
  However, in the following steps of her route, Ellen would start t_8 at the earliest at 4:44PM while t_8 is not available after 4:40PM.
  Therefore, it would still not make Ellen perform t_{27} just after t_{17}.

Planner looking for explanations (2/3):

- What if t_{27} was available up to 4:37PM (instead of 3:00PM)?
- If t_{27} was available until 4:37PM, then Ellen would be able to perform t_{27} during its availability time-window.
  However, in the following steps of her route, Ellen would start t_8 at the earliest at 4:44PM while t_8 is not available after 4:40PM.
  Therefore, it would still not make Ellen perform t_{27} just after t_{17}.





Example or demo

Planner looking for explanations (3/3):

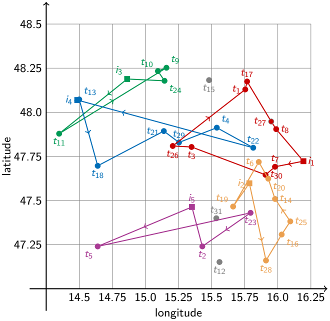
- How to make Ellen perform t_{27} just after t_{17}?
- If t_{27} was available until 4:37PM (instead of 3:00PM) and t_8 until 5:24PM (instead of 4:40PM), then it would make Ellen perform t_{27} just after t_{17}.

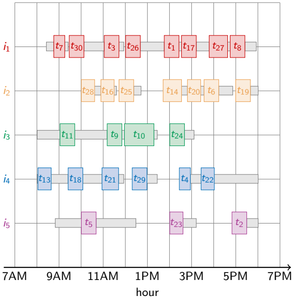
Planner looking for explanations (3/3):

- How to make Ellen perform t_{27} just after t_{17}?
- If t_{27} was available until 4:37PM (instead of 3:00PM) and t_8 until 5:24PM (instead of 4:40PM), then it would make Ellen perform t_{27} just after t_{17}.

Planner looking for explanations (3/3):

- How to make Ellen perform t_{27} just after t_{17}?
- If t_{27} was available until 4:37PM (instead of 3:00PM) and t_8 until 5:24PM (instead of 4:40PM), then it would make Ellen perform t_{27} just after t_{17}.





Plan

Conclusion

Achieved work:

Approach for **generating explanations** that:

- is thought for an end-user of a system solving a **WSRP**;
- starts from **observations** about a solution;
- handles **contrastive**, **scenario** or **counterfactual questions**;
- is (often) based on **mathematical programming**;
- outputs **texts** thanks to templates;

to prevent the end-user from loosing **trust and confidence**.

Work in progress:

- Evaluate how explanations influence **end-users' trust**.
- Perform an exhaustive study for assessing **computational efficiency** of explanations generation.

Perspectives:

How much **generic** is our approach?

Can we transpose it to **other optimization problems**?

Conclusion

Work in progress:

- Evaluate how explanations influence **end-users' trust**.
- Perform an exhaustive study for assessing **computational efficiency** of explanations generation.

Perspectives:

How much **generic** is our approach?
Can we transpose it to **other optimization problems**?

Conclusion

Thank you for your attention!

References I

[Barredo Arrieta et al., 2020] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020).
Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.
***Information fusion*, 58:82–115.**

[Chakraborti et al., 2020] Chakraborti, T., Sreedharan, S., and Kambhampati, S. (2020).
The emerging landscape of Explainable AI Planning and Decision Making.

In *Proceedings of the twenty-ninth International Joint Conference on Artificial Intelligence*, pages 4803–4811. IJCAI Organization.

[Junker, 2004] Junker, U. (2004).
Quickxplain: Preferred explanations and relaxations for over-constrained problems.
In *Proceedings of the ninetieth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, page 167–172. AAAI Press.

[Korikov et al., 2021] Korikov, A., Shleyfman, A., and Beck, C. (2021).
Counterfactual explanations for optimization-based decisions in the context of the GDPR.
In *Proceedings of the thirtieth International Joint Conference on Artificial Intelligence*, pages 4097 – 4103. IJCAI Organization.

[Krarup et al., 2021] Krarup, B., Krivic, S., Magazzeni, D., Long, D., Cashmore, M., and Smith, D. E. (2021).
Contrastive explanations of plans through model restrictions.

[Lipton, 1990] Lipton, P. (1990).
Contrastive explanation.
Royal Institute of Philosophy Supplement, 27:247–266.

[Mohseni et al., 2021] Mohseni, S., Zarei, N., and Ragan, E. D. (2021).
A multidisciplinary survey and framework for design and evaluation of Explainable AI systems.
ACM Transactions on Interactive Intellent Systems, 11(3–4).

References V

[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.**

[Wick and Thompson, 1992] **Wick, M. and Thompson, W. (1992).**
Reconstructive Expert System explanation.
***Artificial Intelligence*, 54:33–70.**