Counterfactual explanations for Workforce Scheduling and Routing Problems

Mathieu Lerouge[1,3,a], Céline Gicquel[2,b], Vincent Mousseau[1,c] and Wassila Ouerdane[1,d]

[1]*MICS CentraleSupelec Universite Paris-Saclay Gif-sur-Yvette France*

[2]*LISN Universite Paris-Saclay Gif-sur-Yvette France*

[3]*DecisionBrain Paris France*

{*mathieu.lerouge, vincent.mousseau, wassila.ouerdane*}*@centralesupelec.fr, gicquel@lri.fr*

Keywords: Workforce scheduling and routing problem, Integer Linear Programming, Counterfactual explanation

Abstract: End-users of system solving combinatorial optimization problems such as the Workforce Scheduling and Routing Problem usually do not have the necessary background for understanding the reasons which has lead the system to output specific solutions and may have questions about them, especially questions about non-expected facts observed in the solutions. Developing techniques to automatically generate explanations in response to end-users questions would then help enhancing their understanding of the results and preserve their trust in the system. In this paper, we propose a new mathematical programming-based approach to compute counterfactual explanations in response to end-users' questions. Such explanations emphasize the few changes that may be operated on the instance data for obtaining solutions corresponding to the end-users' expectations.

1 INTRODUCTION

The Workforce Scheduling and Routing Problem (WSRP) is a Combinatorial Optimization (CO) problem which involves assigning geographically dispersed tasks to members of a mobile workforce and creating a pair of route and schedule for each of these members. It occurs in various contexts like home health care (*e.g.* (Euchi et al., 2022)) or technical service (*e.g.* (Chen et al., 2016)). A literature review on this problem may be found in (Castillo-Salazar et al., 2016).

The WSRP is an NP-hard problem, and various approaches have been proposed to solve it. In the industry, practitioners often solve WSRP instances thanks to decision-aid tools based on mathematical optimization. Our industrial partner in this research project (DecisionBrain, 2022) develops such tools to help its customers solve WSRP instances.

However, most often, the end-users of this software are not experts in operations research and thus do not have the necessary background to understand the mathematical modeling or the algorithmic principles on which these decision-aid tools are built. Thus, when provided with solutions of WSRP instances, they may be surprised by some unexpected facts observed in the solutions and may wish to obtain additional explanations about them.

To illustrate the context, let us consider a small instance of the WSRP involving five mobile members, also called employees, and 31 tasks to be performed over a one-day horizon. Figure 1 displays the solution obtained with an optimization software. Each employee is associated with a color (*e.g.* the red one is related to the employee Ellen). The graph on the left represents the employees' routes: colored dots are tasks performed by employees and gray ones non-performed tasks, while squares correspond to employees' starting locations. The Gantt chart on the right depicts the employees' schedules: higher and colored rectangles represent tasks; smaller and gray ones represent employees' traveling times to go from one task to another; for both groups, the width of a rectangle matches the duration of what it represents.

Facing this solution, an end-user may be surprised by the fact that the task 15 is not performed by Ellen (El in Figure 1) while it is not far from her route. Then, the end-user may wish to get an explanation to clear up the situation.

[a] https://orcid.org/0000-0002-0949-1737

[b] https://orcid.org/0000-0002-2719-7443

[c] https://orcid.org/0000-0001-8574-3337

[d] https://orcid.org/0000-0002-4515-7461

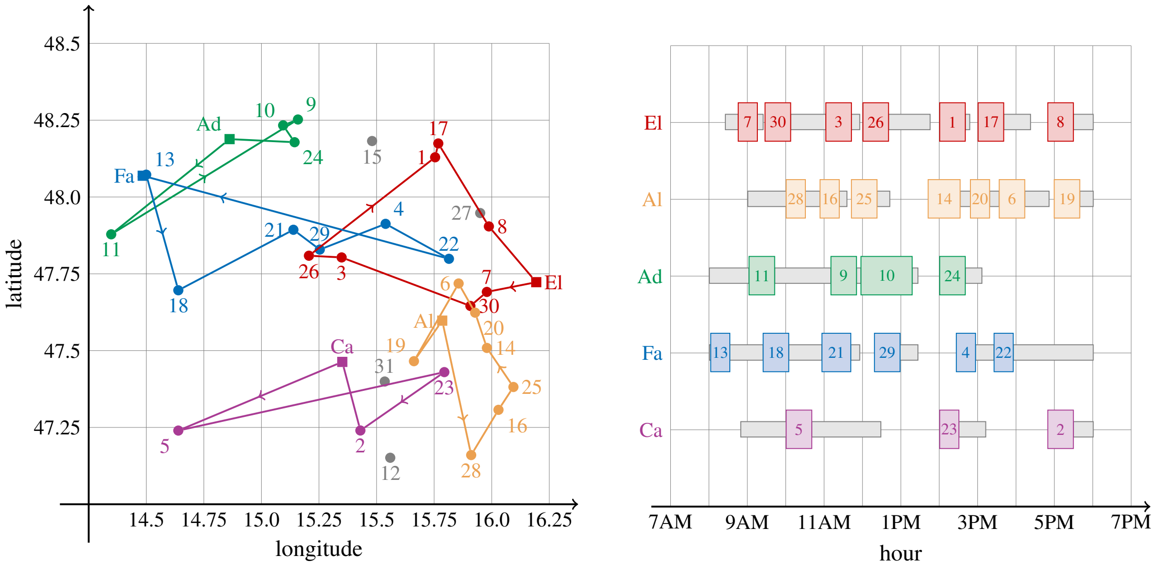

Figure 1: Representation of a WSRP solution (Lerouge et al., 2022).

For optimization experts, answering such queries for explanations requires a lot of time and effort. It namely involves getting familiar with the instance and the solution, investigating the data without being overwhelmed by the high combinatorial aspect of the problem and finding concise pieces of explanations to provide to the end-user. However, providing these explanations is necessary as it helps to preserve the end-users' trust in the system and their confidence at work by enhancing their understanding of the results. Thus, developing techniques for automatically generating explanations in response to end-users questions would improve both end-users' and optimization experts' situations.

Automatic generation of explanations in the context of decision support tools falls within the field of eXplainable Artificial Intelligence (XAI) (Gunning and Aha, 2019) and has been heavily studied by the machine learning community over the past decade (Barredo Arrieta et al., 2020). However, to our knowledge, few works addressed this topic for decision support tools solving CO problems: see (Ludwig et al., 2018; Čyras et al., 2019; Korikov et al., 2021). Moreover, these works rely on strong assumptions about the studied CO problem and the provided explanations that limit their applicability to other CO problems, such as the WSRP.

A noticeable exception is the work of (Lerouge et al., 2022). The authors propose to tackle this issue in the context of the WSRP by focusing on explanations provided in response to "why-not" questions, also known as contrastive questions, *i.e.* questions having the following form "why not that fact instead of this one?" (Lipton, 1990). Essentially, the proposed method is based on identifying and narrating conflicts between the instance data and "that other fact" which the end-user was expecting to observe in the solution. If we go back to our example, such a contrastive question is *e.g.* "*Why is Ellen not performing task 15 in addition to her already-performed tasks?*". In response to it, the authors design an explanation that highlights conflicts, *e.g.* conflicts between the availability time-windows of the tasks, which tells why it is impossible to make Ellen performs the task 15 in her already-performed tasks without violating some of the availability constraints.

From now on, let us name *desideratum* "that other fact" which the end-user was expecting to observe in the solution. The end-user may also raise questions differing from "why-not" questions, especially questions like "how to make the *desideratum* possible?". For instance, in the context of our illustrative example, such a question may be "*How to make Ellen perform task 15 in addition to her already-performed tasks?*". These "how-to" questions ask for counterfactual explanations: given an input of a system and its corresponding output, a counterfactual explanation consists in presenting a change in the input that would have resulted in a different output such as one satisfying the end-user-specified *desideratum*. Counterfactual explanations are attractive as they are easy to comprehend and can be used to offer a path of recourse to end-users receiving unfavorable decisions. Moreover, it was argued by (Wachter et al., 2018) that counterfactual explanations are well aligned with the European Union's General Data Protection Regulation (GDPR) requirements (GDPR, 2016).

Recently, (Korikov et al., 2021) proposed a method to compute counterfactual explanations for CO problems by leveraging inverse optimization. However, their method is restricted to a specific class of CO problem which does not include the WSRP. Besides, it assumes that at most one data element (specifically a coefficient in the objective function which does not appear in any constraint) may be changed to obtain the counterfactual explanation. In our case, such an assumption does not hold, as all the coefficients involved in the objective function also are in the constraints. Therefore, their method cannot be straightforwardly applied to our WSRP case.

Thus, in this paper, we propose a new mathematical programming-based approach to compute counterfactual explanations for an end-user of a WSRP-solving system. Given a solution of a WSRP instance as well as end-user's questions focusing on this solution and satisfying certain assumptions, this method provides counterfactual explanations. Such explanations emphasize the few changes that may be operated on the instance data, which would be enough for obtaining outputs satisfying the end-user's *desiderata*.

The remainder of this article is organized as follows. In Section 2, we clarify our WSRP use case by detailing the content of an instance, introducing at the same time various notations, and presenting an Integer Linear Program modeling this problem. In Section 3, we describe our approach for generating counterfactual explanations, which are then provided to the end-user of a WSRP-solving system. Finally, conclusions and perspectives are given in Section 4.

2 WORKFORCE SCHEDULING AND ROUTING PROBLEM

As mentioned in Section 1, the Workforce Scheduling and Routing Problem (WSRP) consists of assigning tasks to mobile workforce members and creating a pair of routes and schedules for these members. In this section, we describe our WSRP use case, more precisely the content of any instance and its corresponding Integer Linear Programming (ILP) model.

In our use case, we consider a scheduling horizon of one day *i.e.* 1440 minutes. An instance involves a set of employees $\mathcal{E} = \{1, \ldots, n\}$ and a set of tasks $\mathcal{T} = \{1, \ldots, m\}$. Each employee $i \in \mathcal{E}$ has a name, a starting location, which we call home for the sake of simplicity, a skill level $ske_i \in \mathbb{N}$ and working time-window $[lbe_i,\ ube_i] \subseteq [0,\ 1440] \cap \mathbb{N}$. Each task $j \in \mathcal{T}$ has a specific location, a required skill level $skt_j \in \mathbb{N}$, a duration $dt_j \in \mathbb{N}$ and an availability time-window $[lbt_i,\ ubt_i] \subseteq [0,\ 1440] \cap \mathbb{N}$. We assume that hours in time-windows are expressed in minutes from 12:00AM (*e.g.* 8:00AM = 480).

For performing tasks, employees leave their homes when they start their working day and return there when they end it. In order to model these two home events, we introduce, for each employee $i \in \mathcal{E}$, a departure index d_i as well as a return index r_i - where all the indices d_i and r_i for $i \in \mathcal{E}$ are different from each other and different from the indices of $\mathcal{T} = \{1,\ \ldots,\ m\}$. We then extend the set of tasks $\mathcal{T}$ with the set of departure indices and the set of return indices to form a set of activities $\mathcal{A} = \mathcal{T} \cup \{d_1,\ \ldots,\ d_n\} \cup \{r_1,\ \ldots,\ r_n\}$. Besides, for each employee $i \in \mathcal{E}$, we note $\mathcal{A}_i = \mathcal{T} \cup \{d_i,\ r_i\}$ the set of activities obtained by extending $\mathcal{T}$ with the departure and return indices of i.

Finally, an instance also involves traveling times. The time needed to travel between two activities $(j,k) \in \mathcal{A}^2$ is assumed to be independent of the employee and is noted $tr_{jk} \in \mathbb{N}$.

The instance data on which Figure 1 is based are given in the Table 2 of the Appendix A.

In order to model the WSRP using Integer Linear Programming (ILP), we introduce two sets of decision variables.

The first one corresponds to spatial decisions. We introduce, for $i \in \mathcal{E}$ and $(j,k) \in \mathcal{A}_i^2$ with $j \neq k$, a binary variable U_{ijk} such that $U_{ijk} = 1$ if employee i performs activity j and then moves to activity k; $U_{ijk} = 0$ otherwise. By assigning tasks to employees and specifying the order in which the tasks are performed, these variables define the route of each employee.

The second set of variables corresponds to temporal decisions. We introduce, for each $j \in \mathcal{T}$, the integer variable T_j that sets the time at which j starts to be performed by an employee if it is.

Based on the data of the instance and the decision variables that we previously described, we model the WSRP using ILP as presented in Model 1.

The bi-objective function (1) is maximized according to a lexicographic order: the first objective equals the total working time, and the second to the opposite of the total travel time.

Flow constraints (2) to (4) ensure that the employees start from their homes, go from tasks to others without splitting into multiple directions, and end their working day at home. Skill constraints (5) ensure that an employee i can be assigned to a task j only if i has a skill level higher than the one required for performing j. Occurrence constraints (6) ensure that a task j must occur at most once within all employees' routes. Availability constraints (7) to (8)

$$
\begin{aligned}
& lex\,max \quad \Big(\sum_{i\in\mathcal{E}} \sum_{j\in\mathcal{T}} \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk}\, dt_j, \quad -\sum_{i\in\mathcal{E}} \sum_{j\in\mathcal{A}_i,\ j\neq r_i} \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk}\, tr_{jk} \Big) && (1)\\
& s.t. \\
& \sum_{k\in\mathcal{A}_i,\ k\neq d_i} U_{id_ik} = 1 && \forall\, i\in\mathcal{E} \quad (2)\\
& \sum_{j\in\mathcal{A}_i,\ j\neq r_i} U_{ijr_i} = 1 && \forall\, i\in\mathcal{E} \quad (3)\\
& \sum_{j\in\mathcal{A}_i,\ j\neq k,r_i} U_{ijk} = \sum_{j'\in\mathcal{A}_i,\ j'\neq d_i,k} U_{ikj'} && \forall\, i\in\mathcal{E},\ \forall\, k\in\mathcal{T} \quad (4)\\
& \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk} \leq \mathbb{1}_{\{skt_j \leq ske_i\}} && \forall\, i\in\mathcal{E},\ \forall\, j\in\mathcal{T} \quad (5)\\
& \sum_{i\in\mathcal{E}} \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk} \leq 1 && \forall\, j\in\mathcal{T} \quad (6)\\
& \sum_{i\in\mathcal{E}} \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk}\, lbt_j \leq T_j && \forall\, j\in\mathcal{T} \quad (7)\\
& T_j \leq \sum_{i\in\mathcal{E}} \sum_{k\in\mathcal{A}_i,\ k\neq d_i,j} U_{ijk}\, (ubt_j - dt_j) && \forall\, j\in\mathcal{T} \quad (8)\\
& \sum_{i\in\mathcal{E}} U_{id_ik}(lbe_i + tr_{d_ik}) \leq T_k && \forall\, k\in\mathcal{T} \quad (9)\\
& T_j + dt_j + \sum_{i\in\mathcal{E}} U_{ijk}\, tr_{jk} \leq T_k + \Big(1 - \sum_{i\in\mathcal{E}} U_{ijk}\Big) ubt_j && \forall\, (j,k)\in\mathcal{T}^2,\ j\neq k \quad (10)\\
& T_j + dt_j \leq \sum_{i\in\mathcal{E}} U_{ijr_i}(ube_i - tr_{jr_i}) + \Big(1 - \sum_{i\in\mathcal{E}} U_{ijr_i}\Big) ubt_j && \forall\, j\in\mathcal{T} \quad (11)\\
& U_{ijk} \in \{0,1\} \qquad \forall\, i\in\mathcal{E},\ \forall\, j\in\mathcal{A}_i\setminus\{r_i\},\ \forall\, k\in\mathcal{A}_i\setminus\{d_i, j\} \\
& T_j \in \mathbb{N} \qquad \forall\, j\in\mathcal{T}
\end{aligned}
$$

Model 1: Bi-objective Integer Linear Program modeling our Workforce Scheduling and Routing Problem use case.

ensure that if a task j is performed by an employee then j must be started and ended within its availability time-window $[lbt_j,\ ubt_j]$. Work and sequencing constraints (9) to (11) ensure that if an employee i performs a task j then j must be started and ended within the working time-window $[lbe_i,\ ube_i]$ of i.

The solution represented in Figure 1 is given in Table 3 of the Appendix B.

Now that our use case has been specified, we can proceed to the generation of counterfactual explanations, which is the focus of the next section.

3 GENERATING COUNTERFACTUAL EXPLANATIONS

As mentioned in Section 1, this paper presents an approach for generating counterfactual explanations for an end-user who works with an optimization system solving instances of Workforce Scheduling and Routing Problem (WSRP). In this section, we describe how, by starting from the end-user's questions and moving to Integer Linear Programming (ILP), we manage to produce such counterfactual explanations.

3.1 End-user's questions

In our approach, we consider that the end-user specifies a question to request an explanation. Moreover, the question - explanation pairs are such that: *i)* They relate only to a given solution $\mathcal{S}$ of a instance, noted I. *ii)* The question starts with the interrogative form "how-to" to explicitly ask for counterfactual explanations. *iii)* The question mentions a *desideratum i.e.* a fact which is not observed in $\mathcal{S}$ that the end-user would like to have in $\mathcal{S}$ (*e.g.* seeing a task that is not performed by an employee in $\mathcal{S}$ be now performed by this employee). In addition, we assume that we can always find ways to transform the solutions so as to satisfy the end-user's *desideratum* (*e.g.* inserting the non-performed task in the employee's route).

Table 1 lists potential questions which satisfy the assumptions mentioned above. More precisely, it lists template texts that can be used to express questions. Each template text is associated with an identifying label and contains one or several symbols $\langle . \rangle$ indicating fields to be specified with data from the instance I (*e.g.* an employee or a task).

For instance, the end-user's question, mentioned in Section 1, "*How to make Ellen perform the task 15 in addition to her already-performed tasks?*" can be

Labels	Template texts for end-user's questions
(Q1)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ just after ⟨activity k^*⟩?"
(Q2)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ between two consecutive activities of their route?"
(Q3)	"How to make ⟨employee i^*⟩ perform any non-performed task between two consecutive activities of their route?"
(Q4)	"How to make any employee perform ⟨task j^*⟩ between two consecutive activities of their route?"
(Q5)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ in addition to their already-performed tasks (even if it means changing their order)?"
(Q6)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ in place of ⟨task k^*⟩?"
(Q7)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ in place of any of their already-performed task?"
(Q8)	"How to make ⟨employee i^*⟩ perform any non-performed task in place of ⟨task k^*⟩?"
(Q9)	"How to make any employee perform ⟨task j^*⟩ in place of one of their already-performed tasks?"
(Q10)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ instead of any of their already-performed tasks (even if it means changing their order)?"
(Q11)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩, later in their route, just after ⟨task k^*⟩?"
(Q12)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩, earlier in their route, just before ⟨task k^*⟩?"
(Q13)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ later in their route?"
(Q14)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ earlier in their route?"
(Q15)	"How to make ⟨employee i^*⟩ perform ⟨task j^*⟩ at another position in their route?"
(Q16)	"How to make ⟨employee i^*⟩ perform their already-performed tasks in a different order?"

Table 1: Non-exhaustive list of template texts for end-user's questions.

obtained by filling (Q5) template text with fields values "Ellen" and "15". In order to invoke this question again, in future examples, we note it q_{ex}.

Before moving to the next section, let us highlight what can be seen as a limiting requirement of our approach: as a person who provides explanations, one needs to enumerate the end-user's questions be enumerated and define them as template texts. However, such a requirement helps providing a clear and rigorous framework to our approach, while reaching our main goal, which is generating counterfactual explanations in response to various end-user's questions. That is why we also consider this requirement to be reasonable.

In the following subsections, we describe how we process an end-user's question like q_{ex}.

3.2 From questions to ILP modeling

In order to answer an end-user's question with a counterfactual explanation, we exploit ILP. More precisely, given a question template text listed in Table 1, we build a new small-size ILP model upon Model 1. This model aims at finding how to change the instance data to obtain a solution satisfying the end-user's *desideratum* while minimizing the number and magnitude of these changes. To do so, this ILP involves decision variables and constraints which artificially alter the instance data.

Let us emphasize that the ILP model to be built differs according to the template text of Table 1 used to ask the question. Due to space limitation, we choose to only describe the ILP model associated to a question expressed thanks to (Q5) template text, like q_{ex}. However, similar contents could be developed for questions based on other template texts from Table 1.

3.2.1 Preliminaries

Before presenting the ILP model corresponding to the question q, we need to introduce some notations, make additional assumptions about (Q5) template text and specify which parts of the instance I we plan to alter for the counterfactual explanation.

The text of q refers to a specific employee i^* as well as a specific task j^* (both mentioned by (Q5) text fields $\langle . \rangle$) and is relative to the solution $\mathcal{S}$. We introduce $\mathcal{T}^* = \{j_1,\ j_2,\ \ldots,\ j_p\} \cup \{j^*\} \subseteq \mathcal{T}$ the subset of tasks formed by the tasks performed by i^* in $\mathcal{S}$ plus the task j^*. For instance, in the case of q_{ex}, i^* is Ellen, j^* is the task 15 and $\mathcal{T}^* = \{1,3,7,8,15,17,26,30\}$.

For obvious reasons, the task j^* should not be performed by the employee i^* in the solution $\mathcal{S}$. For the sake of simplicity, we assumed more generally that j^* is not performed by any employee, otherwise we would have to deal with the changes to apply to this other employee's route and schedule which is not of major significance. Besides, we also assumed i^* is skilled enough for performing j^* otherwise we would have to alter either the skill level of i^* or j^* which is not really reasonable in practice.

About instance alterations, we need to choose which data of I can be changed and by how much. Such choices depend on the application context. In some cases, it might be possible to extend the time window of a task; sometimes, it might be inconceivable. Ultimately, the end-user should be the one who decides which data can be altered or not. To illustrate our approach, we propose in this paper to allow changes of the availability time-window $[lbt_j,\ ubt_j]$ and duration dt_j for all tasks j in $\mathcal{T}^*$. At the end of Subsection 3.2, we will discuss other possible alteration choices and their impact the ILP modeling.

3.2.2 The ILP Model

We can now proceed to the description of Model 2, *i.e.* the ILP model used for computing the counterfactual explanation that answers q. Model 2 is based on Model 1 but differs from it on various aspects, as detailed in the following paragraphs.

An overall difference between the two models is that Model 2 focuses on optimizing only the route and schedule of a single employee, namely i^*, whereas Model 1 deals with the routes and schedules of all the employees of $\mathcal{E}$. Such a restriction to i^* in Model 2 is possible as the question q focuses on an end-user's *desideratum* about $\mathcal{S}$ which only relates to i^* and the subset of tasks $\mathcal{T}^*$.

The main consequence of this restriction is that not all the decision variables of Model 1 are involved in Model 2: only the binary variables U_{ijk} with $i = i^*$ and the integer variables T_j with $j \in \mathcal{T}^*$ are indeed necessary for the Model 2. In other words, optimizing Model 2 can be seen as fixing in Model 1 all the binary variables U_{ijk} with $i \neq i^*$ as well as all the integer variables T_j with $j \in \mathcal{T} \setminus \mathcal{T}^*$ to the values they take in $\mathcal{S}$ and focusing the optimization over the set of the remaining variables - plus other decision variables which will be described later in this subsection. Due to this reduction of the set of decision variables, a Model 2 solution does not provide directly a Model 1 solution *i.e.* a solution of the instance I. However, in Subsection 3.3, we will see how build a Model 1 solution from $\mathcal{S}$ and a Model 2 solution.

Other consequences of this restriction to i^* are that: there are no sums over $\mathcal{E}$ or constraints repeated over $\mathcal{E}$ in Model 2; sums indexed over $\mathcal{T}$ in Model 1 are indexed over $\mathcal{T}^*$ in Model 2; constraints repeated over $\mathcal{T}$ in Model 1 are repeated over $\mathcal{T}^*$ in Model 2; *etc.* Besides, this restriction leads to a drastic decrease in the size of Model 2 compared with Model 1, which helps to solve Model 2 faster.

Decision Variables. New kinds of decision variables are involved in Model 2, in addition to the spatial ones, U_{i^*jk} with $(j,k) \in \mathcal{A}^2_{i^*}$ such that $j \neq k$, and the temporal ones, T_j for $j \in \mathcal{T}^*$, both introduced in Section 2.

- First, two new integer variables $T^{lb}_{j^*}$ and $T^{ub}_{j^*}$ are brought into play. Like T_{j^*}, they are related to the time at which the task j^* starts to be performed (by i^*): $T^{lb}_{j^*}$ (resp. $T^{ub}_{j^*}$) corresponds to the time at which i^* can start to perform j^* while having all the time constraints related to the activities performed by i^* before (resp. after) j^* satisfied and while respecting the lower (resp. upper) bound of the availability window of j^*. By the way we involve them in Model 2, these two variables allow us to guarantee the feasibility of this model: either $T^{lb}_{j^*} > T^{ub}_{j^*}$ and it means that inserting j^* in the route of i^* is impossible despite the potential alterations of I, either $T^{lb}_{j^*} = T^{ub}_{j^*}$ and it means that the insertion is possible, since all the time constraints of the activities performed by i^* before and after j^* are satisfied.
- Second, several integer variables, so-called *altering variables*, are involved in Model 2 to artificially alter some data of the instance I. For $j \in \mathcal{T}^*$, three integer variables ΔDT_j, ΔLBT_j and ΔUBT_j are introduced allowing respectively to reduce the duration dt_j of task j, to decrease the lower bound lbt_j of its time-window and to increase its upper bound ubt_j. In addition to these altering variables, we introduce ΔT_{max}, an integer variable for measuring the greatest value taken by any of the altering variables.
- Third, for $j \in \mathcal{T}^*$, we associate to the altering variables ΔDT_j, ΔLBT_j and ΔUBT_j, the binary variables XDT_j, $XLBT_j$ and $XUBT_j$, which whether equal to 1 or 0 indicates whether or not their corresponding altering variable take positive value.

Objective Function. In Model 2, a multi-objective function (1') is minimized according to a lexicographic order.

1. As having $T^{lb}_{j^*} = T^{ub}_{j^*}$ means that the insertion of j^* in the route of i^* is possible, the first objective aims at tightening the gap between the two variables $T^{lb}_{j^*}$ and $T^{ub}_{j^*}$ by minimizing the difference $T^{lb}_{j^*} - T^{ub}_{j^*}$. Note that the constraint (12.a) prevents the difference $T^{lb}_{j^*} - T^{ub}_{j^*}$ from being negative.
2. The second objective minimizes the total reduction of tasks duration induced by altering variables ΔDT_j for $j \in \mathcal{T}^*$. Its purpose is to enable decreasing the duration of the performed tasks only if extending the tasks' time-windows, *via* ΔLBT_j and ΔUBT_j, is not enough for successfully inserting j^*.

$$lex\ min\ \Big(\ T^{lb}_{j^*} - T^{ub}_{j^*},\quad \sum_{j\in\mathcal{T}^*}\Delta DT_j,\quad \Delta T_{max},\quad \sum_{j\in\mathcal{T}^*} XDT_j + XLBT_j + XUBT_j,\quad \sum_{j\in\mathcal{A}_{i^*},\ j\neq r_{i^*}}\ \sum_{k\in\mathcal{A}_{i^*},\ k\neq d_{i^*},j} U_{i^*jk}\ tr_{jk}\ \Big) \tag{1'}$$

$s.t.$

$$\sum_{k\in\mathcal{A}_{i^*},\ k\neq d_{i^*}} U_{i^*d_{i^*}k} = 1 \tag{2'}$$

$$\sum_{j\in\mathcal{A}_{i^*},\ j\neq r_{i^*}} U_{i^*jr_{i^*}} = 1 \tag{3'}$$

$$\sum_{j\in\mathcal{A}_{i^*},\ j\neq k,r_{i^*}} U_{i^*jk} = \sum_{j'\in\mathcal{A}_{i^*},\ j'\neq d_{i^*},k} U_{i^*kj'} \qquad \forall\ k\in\mathcal{T}^* \tag{4'}$$

$$\sum_{k\in\mathcal{A}_{i^*},\ k\neq d_{i^*},j} U_{i^*jk} = 1 \qquad \forall\ j\in\mathcal{T}^* \tag{6'}$$

$$lbt_j - \Delta LBT_j \leq T_j \qquad \forall\ j\in\mathcal{T}^*\setminus\{j^*\} \tag{7'.a}$$

$$lbt_j - \Delta LBT_{j^*} \leq T^{lb}_{j^*} \tag{7'.b}$$

$$T_j \leq ubt_j + \Delta UBT_j - dt_j + \Delta DT_j \qquad \forall\ j\in\mathcal{T}^*\setminus\{j^*\} \tag{8'.a}$$

$$T^{ub}_{j^*} \leq ubt_{j^*} + \Delta UBT_{j^*} - dt_{j^*} + \Delta DT_{j^*} \tag{8'.b}$$

$$lbe_{i^*} + tr_{d_{i^*}k} \leq T_k \qquad \forall\ k\in\mathcal{T}^*\setminus\{j^*\} \tag{9'.a}$$

$$lbe_{i^*} + tr_{d_{i^*}j^*} \leq T^{lb}_{j^*} \tag{9'.b}$$

$$T_j + dt_j - \Delta DT_j + U_{i^*jk}\ tr_{jk} \leq T_k + \Big(1 - U_{i^*jk}\Big)ubt_j \qquad \forall\ j\neq k\in\mathcal{T}^*\setminus\{j^*\} \tag{10'.a}$$

$$T^{ub}_{j^*} + dt_{j^*} - \Delta DT_{j^*} + U_{i^*j^*k}\ tr_{j^*k} \leq T_k + \Big(1 - U_{i^*j^*k}\Big)ubt_{j^*} \qquad \forall\ k\in\mathcal{T}^*\setminus\{j^*\} \tag{10'.b}$$

$$T_j + dt_j - \Delta DT_j + U_{i^*jj^*}\ tr_{jj^*} \leq T^{lb}_{j^*} + \Big(1 - U_{i^*jj^*}\Big)ubt_j \qquad \forall\ j\in\mathcal{T}^*\setminus\{j^*\} \tag{10'.c}$$

$$T_j + dt_j - \Delta DT_j \leq U_{i^*jr_{i^*}}(ube_{i^*} - tr_{jr_{i^*}}) + \Big(1 - U_{i^*jr_{i^*}}\Big)ubt_j \qquad \forall\ j\in\mathcal{T}^*\setminus\{j^*\} \tag{11'.a}$$

$$T^{ub}_{j^*} + dt_{j^*} - \Delta DT_{j^*} \leq U_{i^*j^*r_{i^*}}(ube_{i^*} - tr_{j^*r_{i^*}}) + \Big(1 - U_{i^*j^*r_{i^*}}\Big)ubt_{j^*} \tag{11'.b}$$

$$T^{lb}_{j^*} - T^{ub}_{j^*} \geq 0 \tag{12.a}$$

$$T^{ub}_{j^*} \leq T_{j^*} \leq T^{lb}_{j^*} \tag{12.b}$$

$$\Delta DT_j \leq XDT_j\ dt_j \qquad \forall\ j\in\mathcal{T}^* \tag{13.a}$$

$$\Delta LBT_j \leq XLBT_j\ max(lbt_j - lbe_{i^*},\ 0) \qquad \forall\ j\in\mathcal{T}^* \tag{13.b}$$

$$\Delta UBT_j \leq XUBT_j\ max(ube_{i^*} - ubt_j,\ 0) \qquad \forall\ j\in\mathcal{T}^* \tag{13.c}$$

$$\Delta DT_j,\ \Delta LBT_j,\ \Delta UBT_j \leq \Delta T_{max} \qquad \forall\ j\in\mathcal{T}^* \tag{14}$$

$$U_{i^*jk}\in\{0,1\} \qquad \forall\ j\in\mathcal{A}_{i^*}\setminus\{r_{i^*}\},\ \forall\ k\in\mathcal{A}_{i^*}\setminus\{d_{i^*},j\}$$

$$T_j\in\mathbb{N} \qquad \forall\ j\in\mathcal{T}^*$$

$$T^{lb}_{j^*},\ T^{ub}_{j^*}\in\mathbb{N}$$

$$XDT_j,\ XLBT_j,\ XUBT_j\in\{0,1\} \qquad \forall\ j\in\mathcal{T}^*$$

$$\Delta DT_j,\ \Delta LBT_j,\ \Delta UBT_j\in\mathbb{N} \qquad \forall\ j\in\mathcal{T}^*$$

$$\Delta T_{max}\in\mathbb{N}$$

Model 2: Multi-objective program used for computing the explanation that answers a question based on (Q5) template text.

Such a preference for altering first time-windows and then duration is applied as decreasing tasks' duration actually depreciates the solution since, by Model 1, the primary objective is to maximize the total tasks duration performed by the employees.

3. The purpose of the third objective which minimizes ΔT_{max} is to prevent alterations from being too large, as much as possible.
4. The one of the fourth is to prevent them being too numerous, as much as possible.
5. Lastly, the fifth objective is linked to the second of Model 1 as it minimizes the total traveling time.

Constraints. Most constraints of the Model 1 still apply in the Model 2 but must be adapted. Flow constraints (2) to (4) correspond to constraints (2') to (4'); however (2') to (4') zoom on the employee i^* and the subset $\mathcal{T}^*$ while (2) to (4) involve the whole sets $\mathcal{E}$ and $\mathcal{T}$. Occurrence constraint (5) coincide with constraint (5') but in the latter, as we look for adding j^* among the tasks performed by i^*, all the tasks of $\mathcal{T}^*$ must be performed by i^* hence the use of $=$ in (5') instead of $\leq$ in (5). No equivalent of the skill constraint (6) appears in the Model 2 as i^* is supposed to have a higher skill level than the ones of all the tasks of $\mathcal{T}^*$ (see our assumptions in Subsection 3.2.1). Availability, working hours and sequencing constraints spanning labels from (7) to (11) in Model 1 are also involved, with some changes, in Model 2 as labels spanning from (7'.a) to (11'.b). Each original constraint of Model 1 (*e.g.* constraint (7)) is split into two or three constraints in Model 2 (*e.g.* constraints (7'.a) and (7'.b)) in order to separate the case of j^*, which involves $T_{j^*}^{lb}$ and $T_{j^*}^{ub}$, from the one of any other task j in $\mathcal{T}^*$, which involves T_j.

In addition, new constraints are introduced in Model 2. As mentioned earlier, constraint (12.a) prevents the difference $T_{j^*}^{lb} - T_{j^*}^{ub}$ from being negative. Constraint (12.b) is simply used for controlling T_{j^*} value which is no longer involved in any constraint. Constraints (13.a) to (13.c) ensure that the binary variables XDT_j, $XLBT_j$ and $XUBT_j$ for $j \in \mathcal{T}^*$ play their expected role, namely indicating whether or not their corresponding altering variable take a positive value. Besides, these constraints limit the value of the altering variables ΔDT_j, ΔLBT_j and ΔUBT_j for $j \in \mathcal{T}^*$ to some "reasonable bounds": the duration dt_j of a task j can not be reduced by more than its own value; there is no point in having the lower and upper bounds of the availability time-window of task respectively smaller and higher than the lower and upper bounds of the working time-window of employee i^*. Finally, constraint (14) ensure that ΔT_{max} measures as expected the greatest value taken by any of the altering variables.

Discussion. Before ending this Subsection 3.2, let us comeback to the discussion, started up in Subsection 3.2.1, about the way I can be altered, in particular which data can be changed and up to what bounds.

It is clear that there are many ways to alter the data of I. In this paper, we chose to locate the potential alterations on task availability time-window and duration data, to allow such alterations on all the tasks of $\mathcal{T}^*$ and to limit such alterations to some "reasonable bounds". However, one could also choose, for instance, to locate alterations on employee working time-window data in addition to the ones on the task data, to allow task alterations only for some selected tasks of $\mathcal{T}^*$, and to bound altering variables with arbitrary smaller bounds than our "reasonable bounds".

Still, such other choices could actually be taken into account in Model 2 considering some adaptions. One would have *i)* to introduce a pair of integer variables ΔLBE_{i^*} and ΔUBE_{i^*}, as well as their corresponding binary variables $XLBE_{i^*}$ and $XUBE_{i^*}$, for altering the time-window of i^*; *ii)* to only consider ΔLBT_j, ΔUBT_j and ΔDT_j for the selected tasks of $\mathcal{T}^*$; *iii)* to adapt the fourth objective in the multi-objective function; *iv)* to adapt constraints (7'.a) to (11'.b) to involve ΔLBE_{i^*} and ΔUBE_{i^*} as well as ΔLBT_j, ΔUBT_j and ΔDT_j only for the selected tasks of $\mathcal{T}^*$; *v)* replace the "reasonable bounds" in constraints (13.a) to (13.c) with arbitrarily chosen bounds; *etc.*

Thus, we can let the end-user - who is actually the person who knows the best the application context of the WSRP solved - choose the locations and the bounds of the potential alterations to apply on the data of I and still be able to define a Model 2 adapted to such choices.

To conclude, this Subsection 3.2 was devoted to describe how we move from the end-user's question q to an ILP model, namely Model 2. In the following subsection, we present how we exploit an optimal solution of Model 2 to produce a counterfactual explanation.

3.3 From ILP modeling to explanations

As a continuation of Subsection 3.2, in this subsection, we keep focusing on a question q that relates to a solution $\mathcal{S}$ of a WSRP instance I that is expressed thanks to (Q5) template text. In addition, we assume that we obtained an optimal solution of Model 2 that is associated with q. Such a solution can be obtained using any ILP solver on the Model 2. Note that solving Model 2 requires a reasonable computational effort thanks to the drastic reduction in the model size that we obtained by restricting the model to the route and schedule of i^* (*c.f.* Subsection 3.2.2).

As mentioned in Subsection 3.2.2, due to the reduction of the set of decision variables in Model 2 to the U_{ijk} such that $i = i^*$ and the T_j such that $j \in \mathcal{T}^*$, a Model 2 solution is not directly a solution of Model 1 *i.e.* a solution of I. However, given $\mathcal{S}$ and a Model 2 solution, we can build a new Model 1 solution as follows. Let us note $\mathcal{S}_2$ a solution of Model 2. By replacing in $\mathcal{S}$ the values of the decision variables which are common to Model 1 and Model 2 with the values that these variables take in $\mathcal{S}_2$, we can build a new solution $\mathcal{S}'$ of the Model 1.

In particular, when $\mathcal{S}_2$ is an optimal solution of Model 2, we call the solution $\mathcal{S}'$ a *neighboring solution* of $\mathcal{S}$ *induced by* Model 2. Note that in most cases $\mathcal{S}'$ is likely to be infeasible relatively to I as $\mathcal{S}_2$ is likely to have positive altering variables corresponding to necessary alterations of I.

Similarly, we can define a notion of *neighboring instance* of I *induced by* Model 2. Indeed, Model 2 contains not only decision variables which are shared with Model 1 but also its own decision variables like the altering ones. By changing the data of I according to the values taken by these altering variables $\mathcal{S}_2$, we can build a new WSRP instance I'. Again, when $\mathcal{S}_2$ is an optimal solution of Model 2, we call such instance I' a neighboring instance of I induced by Model 2. Note that in most cases $I' \neq I$.

From now on, in addition to q, $\mathcal{S}$ and I already introduced, let us consider $\mathcal{S}_2$ an optimal solution of Model 2 that is related to q, $\mathcal{S}'$ a neighboring solution of $\mathcal{S}$ induced by Model 2 and I' a neighboring instance of I induced by Model 2.

We identify three possible cases based on the values taken by decision variables in the optimal solution $\mathcal{S}_2$ of Model 2.

Case 1: $T_{j^*}^{lb} - T_{j^*}^{ub} = 0$ and $\Delta T_{max} > 0$.
Case 2: $T_{j^*}^{lb} - T_{j^*}^{ub} = 0$ and $\Delta T_{max} = 0$.
Case 3: $T_{j^*}^{lb} - T_{j^*}^{ub} > 0$.

Case 1 corresponds to the case where by resorting to some alterations of the data of I, it is possible to obtain the end-user's *desideratum* (*i.e.* it is possible to make employee i^* perform the task j^* in addition to their already-perform tasks, knowing that q is a question based on (Q5) template text). In other words, the neighboring solution $\mathcal{S}'$ is feasible relatively to the neighboring instance I' and $I' \neq I$.

The counterfactual explanation for Case 1 can be expressed as follows: "*By* $\langle$*applying the instance alterations associated with all the non-zero altering variables causing* $I \neq I'$$\rangle$, $\langle$*the desideratum*$\rangle$ *would be possible; in this case, the solution would be* $\langle$*description of* $\mathcal{S}'$$\rangle$"

Case 1 is actually the case encountered for answering q_{ex}, our illustrative example question introduced in Section 1. Indeed, the counterfactual explanation is "*By changing the opening time of task 17, from its value 12:30PM in the current input data, to 12:29PM instead, making Ellen perform the task 15 in addition to her already-performed tasks would be possible; in this case, the solution would be the one obtained by changing Ellen's sequence of performed tasks to [30, 7, 8, 1, 17, 15, 26, 3].*"

Case 2 corresponds to the case where it is possible to obtain the end-user's *desideratum* without even resorting to any alterations of the data of I. In other words, the neighboring solution $\mathcal{S}'$ is feasible relatively to the original instance I (and $I' = I$).

In practice, knowing that q is a question based on (Q5) template text, such case is actually expected not to occurred so often. Indeed, as q asks for adding a non-performed task j^* to the already-performed tasks of an employee i^*, if the transformed solution is feasible then it would necessarily be better than $\mathcal{S}$. However, if $\mathcal{S}$ has been obtained thanks to an optimization system applied on I, $\mathcal{S}$ should be a good solution, possibly even an optimal one, which can not be improved by local transformations. Therefore, transforming $\mathcal{S}$ to satisfy the end-user's *desideratum* should certainly result in an infeasible solution.

The counterfactual explanation for Case 2 can be expressed as follows: "*Without altering any data of the current instance,* $\langle$*the desideratum*$\rangle$ *is possible; the solution is* $\langle$*description of* $\mathcal{S}'$$\rangle$."

Case 3 corresponds to the case where it is not possible to obtain the end-user's *desideratum* despite the possibility of altering the date of I.

Facing this situation, we could have provided a simple explanation text telling that it is not possible to obtain the *desideratum* despite all the possible alterations of the data of I. However, thanks to the decision variables $T_{j^*}^{lb}$ and $T_{j^*}^{ub}$, which guarantee the feasibility of Model 2 (*c.f.* Subsection 3.2), even if the insertion of j^* in the route of i^* is not possible despite the alterations of I, we can still build a neighboring solution $\mathcal{S}'$ and a neighboring instance I'. In such case, $\mathcal{S}'$ is admittedly infeasible relatively to I' but $\mathcal{S}'$ can be interpreted as the solution satisfying the *desideratum* that is the closest to be feasible relatively to I. We can then use this information to enrich the explanation text provided to the end-user.

The counterfactual explanation for Case 3 can be expressed as follows: "*Despite all the possible alterations of the current instance data,* $\langle$*the desideratum*$\rangle$ *is not possible. The best that could be done consists in* $\langle$*applying the instance alterations associated with all the non-zero altering variables causing* $I \neq$

$I'\rangle$; *in this case, the best solution satisfying* $\langle$*the desideratum*$\rangle$ *is the solution* $\langle$*description of* $\mathcal{S}'\rangle$ *which is not valid though*."

To conclude, in this Subsection 3.3, we presented how the result of Model 2 can be used to build a counterfactual explanation answering to the end-user's question q by identifying three cases and adapting the explanation text in consequences.

4 CONCLUSION

In this paper, we proposed a new approach, based on mathematical programming, to compute counterfactual explanations for an end-user of an optimization system solving the Workforce Scheduling and Routing Problem (WSRP), that we modeled as a bi-objective Integer Linear Programming (ILP) model.

We considered that, facing a solution $\mathcal{S}$ of a WSRP instance I that was provided by an optimization system, the end-user requests explanations by expressing questions which satisfy certain assumptions. Among others, these questions must be about $\mathcal{S}$ and focus on an end-user's *recourse* about it. We provided a non-exhaustive list of template texts for formulating such questions.

Given an end-user's question q, we associated q with a multi-objective ILP model. This model involves altering variables which enable precise changes in the instance I data. The purpose of this ILP model is to look for the changes to operate on I for obtaining a new solution $\mathcal{S}'$ satisfying the end-user's *desideratum*.

In terms of future works, there are various research topics that we would like to explore. Among others, one of them is to evaluate our approach in terms of benefits brought by our explanations to the end-user. A second is to transpose our approach to other Combinatorial Optimization problems than the WSRP to investigate its genericity.

ACKNOWLEDGEMENTS

This work has been supported by the project PSPC AIDA: 2019- PSPC-09 funded by BPI-France. We thank Daniel Godard, Filippo Focacci and Desirée Rigonat from DecisionBrain for their time and their helpful feedback throughout this work.

REFERENCES

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.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2016). Workforce Scheduling and Routing Problems: literature survey and computational study. *Annals of Operations Research*, 239:39 – 67.

Chen, X., Thomas, B. W., and Hewitt, M. (2016). The Technician Routing Problem with experience-based service times. *Omega*, 61:49–61.

Čyras, K., Letsios, D., Misener, R., and Toni, F. (2019). Argumentation for Explainable Scheduling. In *Proceedings of the thirty-third Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 2752 – 2759. AAAI Press.

DecisionBrain (2022). Decisionbrain s.a.s. https://decisionbrain.com/. Accessed 2022-10-10.

Euchi, J., Masmoudi, M., and Siarry, P. (2022). Home health care routing and scheduling problems: a literature review. *4OR*, pages 1–39.

GDPR (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

Gunning, D. and Aha, D. (2019). DARPA's Explainable Artificial Intelligence (XAI) program. *AI Magazine*, 40:44–58.

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.

Lerouge, M., Gicquel, C., Mousseau, V., and Ouerdane, W. (2022). Explaining solutions stemming from optimization systems solving the Workforce Scheduling and Routing Problem to their end-users. Working paper https://hal.archives-ouvertes.fr/hal-03795653.

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

Ludwig, J., Kalton, A., and Stottler, R. (2018). Explaining complex scheduling decisions. In *Intelligent User Interfaces Workshops*.

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.

APPENDIX

Appendix A. Instance

Employee i in $\mathcal{E}$	**Name**	**Skill level** ske_i in $\mathbb{N}^\star$	**Location** as (lat., long.) in deg.	**Time-window** $[lbe_i,\ ube_i]$ as time range	as integer range
1	Ellen	2	(47.773, 16.193)	[8:00AM, 6:00PM]	[480, 1080]
2	Alex	3	(47.598, 15.785)	[9:00AM, 6:00PM]	[540, 1080]
3	Adam	2	(48.188, 14.862)	[8:00AM, 6:00PM]	[480, 1080]
4	Fabian	1	(48.069, 14.485)	[8:00AM, 6:00PM]	[480, 1080]
5	Carlotta	1	(47.463, 15.351)	[8:00AM, 6:00PM]	[480, 1080]

Task j in $\mathcal{T}$	**Skill level** skt_j in $\mathbb{N}^\star$	**Location** as (lat., long.) in deg.	**Duration** dt_j in min.	**Time-window** $[lbt_j,\ ubt_j]$ as time range	as integer range
1	1	(48.129, 15.754)	40	[8:00AM, 6:00PM]	[480, 1080]
2	1	(47.240, 15.430)	40	[2:00PM, 6:00PM]	[840, 1080]
3	1	(47.803, 15.348)	40	[11:00AM, 6:00PM]	[660, 1080]
4	1	(47.913, 15.539)	30	[2:00PM, 6:00PM]	[840, 1080]
5	1	(47.240, 14.639)	40	[8:00AM, 12:00PM]	[480, 720]
6	3	(47.719, 15.856)	40	[8:00AM, 6:00PM]	[480, 1080]
7	2	(47.691, 15.979)	30	[8:45AM, 12:00PM]	[525, 720]
8	1	(47.904, 15.988)	40	[8:00AM, 6:00PM]	[480, 1080]
9	1	(48.252, 15.159)	40	[9:00AM, 12:00PM]	[540, 720]
10	2	(48.233, 15.094)	80	[9:00AM, 6:00PM]	[540, 1080]
11	2	(47.879, 14.348)	40	[8:00AM, 6:00PM]	[480, 1080]
12	2	(47.151, 15.559)	40	[8:00AM, 6:00PM]	[480, 1080]
13	1	(48.073, 14.500)	30	[8:00AM, 6:00PM]	[480, 1080]
14	1	(47.509, 15.979)	50	[8:00AM, 6:00PM]	[480, 1080]
15	2	(48.182, 15.480)	25	[8:00AM, 2:00PM]	[480, 840]
16	1	(47.307, 16.030)	30	[8:00AM, 6:00PM]	[480, 1080]
17	1	(48.174, 15.768)	40	[12:30PM, 4:00PM]	[750, 960]
18	1	(47.696, 14.639)	40	[8:00AM, 6:00PM]	[480, 1080]
19	2	(47.466, 15.662)	40	[8:00AM, 6:00PM]	[480, 1080]
20	2	(47.624, 15.928)	30	[8:00AM, 6:00PM]	[480, 1080]
21	1	(47.893, 15.139)	45	[8:00AM, 6:00PM]	[480, 1080]
22	1	(47.799, 15.814)	30	[3:00PM, 6:00PM]	[900, 1080]
23	1	(47.430, 15.794)	30	[10:00AM, 3:00PM]	[600, 900]
24	2	(48.178, 15.144)	40	[2:00PM, 3:00PM]	[840, 900]
25	1	(47.382, 16.094)	40	[8:00AM, 1:00PM]	[480, 780]
26	2	(47.809, 15.206)	40	[12:00PM, 6:00PM]	[720, 1080]
27	2	(47.948, 15.950)	50	[8:00AM, 3:00PM]	[480, 900]
28	3	(47.160, 15.991)	30	[8:00AM, 6:00PM]	[480, 1080]
29	1	(47.829, 15.253)	40	[8:00AM, 1:00PM]	[480, 780]
30	1	(47.646, 15.907)	40	[8:00AM, 12:00PM]	[480, 720]
31	2	(47.399, 15.535)	40	[8:00AM, 4:00PM]	[480, 960]

Table 2: An example of WSRP instance. The first table describes the data about the set of employees $\mathcal{E}$. Each employee $i \in \mathcal{E}$ is characterized by a skill level ske_i, a departure and return location, and a time-window $[lbe_i,\ ube_i]$. The second table describes the data about the tasks. Each task j is described by a skill level skt_j, a location, a duration dt_j and a time-window $[lbt_j,\ ubt_j]$. It is assumed that all the employees have the same traveling speed of 50km/h. Considering that the earth radius is 6731km, the traveling time, in minutes, between two locations $(lat_1,\ long_1)$ and $(lat_2,\ long_2)$ can be computed as follows: $6731 \times \arccos(\sin(lat_1) \times \sin(lat_2) + \cos(lat_1) \times \cos(lat_2) \times \cos(long_2 - long_1))/50 \times 60$. (Lerouge et al., 2022)

Appendix B. Solution

Employee i	**Binary decision variables** $U_{i,j,k}$
1	$U_{1,d_1,7} = U_{1,7,30} = U_{1,30,3} = U_{1,3,26} = U_{1,26,1} = U_{1,1,17} = U_{1,17,8} = U_{1,8,r_1} = 1$ $U_{1,j,k} = 0$ for all other couples of activities (j,k)
2	$U_{2,d_2,28} = U_{2,28,16} = U_{2,16,25} = U_{2,25,14} = U_{2,14,20} = U_{2,20,6} = U_{2,6,19} = U_{2,19,r_2} = 1$ $U_{2,j,k} = 0$ for all other couples of activities (j,k)
3	$U_{3,d_3,11} = U_{3,11,9} = U_{3,9,10} = U_{3,10,24} = U_{3,24,r_3} = 1$ $U_{3,j,k} = 0$ for all other couples of activities (j,k)
4	$U_{4,d_4,13} = U_{4,13,18} = U_{4,18,21} = U_{4,21,29} = U_{4,29,14} = U_{4,14,22} = U_{4,22,r_4} = 1$ $U_{4,j,k} = 0$ for all other couples of activities (j,k)
5	$U_{5,d_5,5} = U_{5,5,23} = U_{5,23,2} = U_{5,2,r_5} = 1$ $U_{5,j,k} = 0$ for all other couples of activities (j,k)

Task j	**Integer decision variables** T_j	**Task** j	**Integer decision variables** T_j
1	840	17	900
2	1009	18	564
3	662	19	1019
4	866	20	888
5	600	21	656
6	933	22	925
7	525	23	840
8	1009	24	840
9	670	25	702
10	717	26	720
11	542	27	0
12	0	28	600
13	482	29	738
14	822	30	567
15	0	31	0
16	653		

Table 3: An example of solution of the WSRP instance described in Table 2. This solution also represented in Figure 1. The first table gives the values of the binary decision variables U_{ijk} and the second table gives the values of the integer decision variables T_j. (Lerouge et al., 2022)