

Explaining Preference-Driven Schedules: The EXPRES Framework

Alberto Pozanco¹, Francesca Mosca¹, Parisa Zehtabi¹, Daniele Magazzeni¹, Sarit Kraus²

¹J.P. Morgan AI Research

²Department of Computer Science, Bar-Ilan University

{alberto.pozanco, francesca.mosca, parisa.zehtabi, daniele.magazzeni}@jpmorgan.com, sarit@cs.biu.ac.il

Abstract

Scheduling is the task of assigning a set of scarce resources distributed over time to a set of agents, who typically have preferences about the assignments they would like to get. Due to the constrained nature of these problems, satisfying all agents’ preferences is often infeasible, which might lead to some agents not being happy with the resulting schedule. Providing explanations has been shown to increase satisfaction and trust in solutions produced by AI tools. However, it is particularly challenging to explain solutions that are influenced by and impact on multiple agents. In this paper we introduce the EXPRES framework, which can explain why a given preference was unsatisfied in a given optimal schedule. The EXPRES framework consists of: (i) an *explanation generator* that, based on a Mixed-Integer Linear Programming model, finds the best set of reasons that can explain an unsatisfied preference; and (ii) an *explanation parser*, which translates the generated explanations into human interpretable ones. Through simulations, we show that the explanation generator can efficiently scale to large instances. Finally, through a set of user studies within J.P. Morgan, we show that employees preferred the explanations generated by EXPRES over human-generated ones when considering workforce scheduling scenarios.

Introduction

Scheduling is the task of assigning a set of scarce resources distributed over time to a set of agents. This is the case of many well-known problems, such as assigning jobs to machines (Watson et al. 2003), nurses to work shifts (Legrain, Bouarab, and Lahrichi 2015), or teachers to courses (Gunawan, Ng, and Poh 2007), among others. Another application that has become very relevant to organisations (including J.P. Morgan) due to COVID-19 restrictions is that of scheduling or assigning employees to a limited number of desks (less than in normal situations in order to guarantee social distancing) over a fixed time period. In this context, employees may have specific preferences regarding their schedules, such as the dates or the number of days a week that they want to be at the workplace, or the peers they would like to meet more often, etc. However, the limited availability of desks may preclude the fulfillment of all of the employees’ preferences, and this may hinder their satisfaction with

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

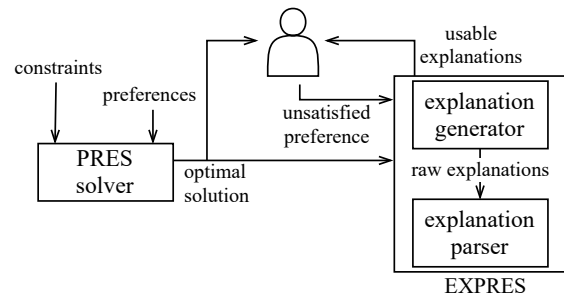


Figure 1: Overview of the EXPRES framework.

the schedule. Presenting a contrastive explanation of the reasons why the employees could not be scheduled in any other way may promote the acceptance of the schedule (Bradley and Sparks 2009). Furthermore, beside the intrinsic difficulty of explaining a schedule for an individual, more challenges arise when considering the automated explanation of decisions regarding multiple agents. Among the features that (Kraus et al. 2020) argue for, an explainable system should (i) allow its users to understand its decision-making, (ii) be able to generate different types of explanations, so to provide tailored outputs for its users, (iii) while preserving the agents’ privacy.

In this paper we present the EXPRES framework (see Fig. 1), a novel approach to providing EXplanations for PReference-driven Scheduling problems (PRES). Our formalisation is general, allowing it to be applied in different scenarios; however, for the sake of clarity, we discuss and empirically evaluate a complete example of EXPRES in the context of workforce scheduling at J.P. Morgan.

We first define PRES problems as scheduling tasks where an optimal solution is identified not only by a set of constraints, but also by a totally ordered set of preferences. We then formalise the problem of explaining PRES solutions (EXPRES) as an optimisation task where, given a schedule and an agent’s unsatisfied preference, we find the best set of reasons that can justify it. On one hand, EXPRES explanations simplify the task, frequently manually performed with notable effort and time-consumption, of providing justification for a specific unsatisfied preference in a given schedule. On the other hand, these explanations aim to support

the individual’s understanding of the other factors, beyond their own preferences, that influenced the schedule. We propose to model EXPRES as a Mixed-integer Linear Programming (MILP) problem. After that, we show how to group and translate the computer generated explanations to natural language in order for them to be easily interpreted by humans. After discussing how to convey the explanations to end users, we show through software simulations that EXPRES is able to (i) scale to large instances, in terms of number of employees and preferences; and (ii) provide different explanations, in terms of reasons to justify the unsatisfied preference. Later, we also present the results of a user study within J.P. Morgan that shows how EXPRES explanations are better appreciated than human-generated ones. Finally, we draw our main conclusions and outline future work.

PRES: Preference-driven Scheduling

In this section we formalise a scheduling problem where a finite set of *resources* R distributed across different *time slots* T needs to be assigned to a set of *agents* Ag . Each agent might have a set of constraints as well as preferences of different types. An external actor, namely the Principal, specifies a set of global constraints and a total order over the agents’ preference types¹. We refer to this set of scheduling problems as PRES, and formally define them as follows:

Definition 1. A **PRES problem** is a tuple $\text{PRES} = \langle R, Ag, T, \mathcal{C}, \mathcal{P}, \mathcal{O} \rangle$ where:

- R is a set of resource types
- Ag is a set of agents
- T is a set of time slots
- $\mathcal{C} = \bigcup_{ag \in Ag} C_{ag} \cup C_{Princ}$ is the set of all the constraints, where C_{ag} are the constraints of the agent ag , and C_{Princ} are the constraints imposed by the Principal
- $\mathcal{P} = \bigcup_{ag \in Ag} P_{ag}$ is the set of all agents’ preferences
- $\mathcal{O} = \{\tau_1 \prec \dots \prec \tau_{|\mathcal{O}|}\}$ is a totally ordered set of preference types τ_i defined by the Principal; $\tau_i \prec \tau_j$ means that τ_i precedes (is more important than) τ_j in the order.

Given a preference $p \in \mathcal{P}$, we refer to the agent having the preference, to its type and to the order of its type as $ag(p)$, $\tau(p)$ and $\mathcal{O}(p)$ respectively. The solution to a PRES problem is a **schedule** S , which consists of a set of **assignments** $as = \langle ag, r, t \rangle \in \{Ag \times R \times T\}$ that assign agents to resources and time slots. Again, $ag(as)$ and $r(as)$ refer to the agent and the resource of the assignment respectively. We refer as $\mathcal{S} \subseteq \{Ag \times R \times T\}$ to the set of all **feasible schedules** subject to the constraints in \mathcal{C} . The optimal solution to a PRES problem is defined according to the totally ordered set of preference types \mathcal{O} .

Definition 2. Given a set of solutions $\tilde{\mathcal{S}}$ and a preference type τ , a **preference filter function** $f_\tau : \tilde{\mathcal{S}} \rightarrow \tilde{\mathcal{S}}_\tau$ returns the set of solutions that maximises the number of preferences $p \in \mathcal{P}$ of type τ that are satisfied.

¹This problem setting was considered in a real-world scenario, namely the return to the office at J.P. Morgan, as we discuss later.

We formally define an optimal schedule given a totally ordered set of preference types and the set of feasible schedules as follows:

Definition 3. A **set of optimal schedules** S^* is the output of a composition of preference filter functions, where the order of composition is given by the total order \mathcal{O} :

$$S^* = f_{\tau_{|\mathcal{O}|}} \circ \dots \circ f_{\tau_1}(S)$$

where τ_1 is the first (most important) preference type.

In the rest of the paper we assume that the computed solutions are optimal and that the agents’ preferences do not contradict their own constraints or the constraints defined by the Principal (e.g., an agent cannot ask for a number of resources higher than their own maximum number or higher than the number of available resources).

EXPRES: Explaining PRES Solutions

Given the constrained nature of the PRES problems, it may happen that in a schedule S some agents’ preferences are not accommodated. We refer to these as **unsatisfied preferences** $\text{UNSAT} \subseteq \mathcal{P}$, using $\text{SAT} \subseteq \mathcal{P}$ to denote the set of **satisfied preferences**. Note that $\text{UNSAT} \cap \text{SAT} = \emptyset$. In such scenarios we aim to provide agents with informative explanations that clarify why some of their preferences were unsatisfied. Particularly, we explain unsatisfied preferences by reporting the *reasons* why the preferred assignments had to be assigned in a different way. We refer to this task as *explaining* PRES solutions (EXPRES); before formalising it, we define some concepts.

In order to provide an explanation for why preference $u \in \text{UNSAT}$ is unsatisfied, we first need to identify the set of assignments that were responsible for or involved in u being unsatisfied.

Definition 4. $\text{INVOLVED}(S, u)$ is a domain-specific function that computes the **set of assignments** $S' \subseteq S$ **involved** in $u \in \text{UNSAT}$ being **unsatisfied**.

For example, assuming an unsatisfied preference u , an assignment as is involved in u if $t(as) = t(u)$, i.e., if there is another agent assigned to the time slot to which agent $ag(u)$ preferred to be assigned. The INVOLVED function relates assignments to unsatisfied preferences, returning the set of assignments to be justified or explained.

We also need a way of relating satisfied preferences to assignments, i.e., if a given assignment is affected or not by a preference.

Definition 5. $\text{AFFECTED}(p, as)$ is a domain-specific function that returns a binary output indicating whether **assignment** $as \in S$ is **affected by satisfied preference** $p \in \text{SAT}$.

For example, assuming a satisfied preference p , an assignment as is affected by p if $ag(p) = ag(as)$, i.e., the agent ag had a preference regarding that assignment.

Explanations in our setting are formed by reasons that justify why the preferred assignments had to be assigned in a different way. We formally define a reason as follows:

Definition 6. Given a PRES problem and a schedule S that solves it, a **reason** is a tuple $\mathcal{R} = \langle p, as, u \rangle$, where $p \in$

$SAT \subseteq \mathcal{P}$ is a satisfied preference, $as \in S$ is an assignment, and $u \in UNSAT \subseteq \mathcal{P}$ is an unsatisfied preference.

Reasons can be read as “preference u about assignment as was unsatisfied due to preference p being satisfied”.

Definition 7. A reason $\mathcal{R} = \langle p, as, u \rangle$ is **well defined** iff (1) $AFFECTED(p, as) = 1$; (2) $as \in INVOLVED(S, u)$; and (3) $RANK(p, \leq, u)$.

Here we assume a RANK function that returns 1 (True) if $\mathcal{O}(p) \leq \mathcal{O}(u)$. A well defined reason employs a more important preference over an assignment to explain why a lower (or equally) important preference over that same assignment was unsatisfied. If both preferences have the same rank ($\mathcal{O}(p) = \mathcal{O}(u)$), the reason suggests that there exists another optimal schedule in which u was satisfied, because they are equally important. With these definitions at hand we are ready to formalise an EXPRES problem as follows:

Definition 8. An **EXPRES problem** is a tuple $EXPRES = \langle PRES, S, u \rangle$ where:

- $PRES = \langle R, Ag, T, C, \mathcal{P}, \mathcal{O} \rangle$ is a PRES problem
- S is a set of assignments that optimally solve PRES
- $u \in UNSAT$ is an unsatisfied preference

The **solution to an EXPRES problem** is an explanation $\mathcal{E}_{S,u}$, which is a set of reasons that describe why a preference u was unsatisfied in a schedule S that optimally solves a PRES problem.

Explanations have different properties depending on the reasons they contain. We say that an explanation is complete if it provides a reason for each of the assignments involved in an unsatisfied preference.

Definition 9. An **explanation** $\mathcal{E}_{S,u}$ that solves an EXPRES problem is **complete** iff $\forall as \in INVOLVED(S, u), \exists \mathcal{R} \in \mathcal{E}_{S,u} \mid as = as(\mathcal{R})$.

We also say that an explanation is sound if all of the reasons in the explanation are well-defined.

Definition 10. An **explanation** $\mathcal{E}_{S,u}$ that solves an EXPRES problem is **sound** iff \mathcal{R} is well-defined $\forall \mathcal{R} \in \mathcal{E}_{S,u}$.

Finally, we say that an explanation is optimal if it minimizes the order of the satisfied preferences used by its reasons.

Definition 11. An **explanation** $\mathcal{E}_{S,u}$ that solves an EXPRES problem is **optimal** iff $\min_{\mathcal{O}(p)} \sum_{\mathcal{R} \in \mathcal{E}_{S,u}} \mathcal{O}(p(\mathcal{R}))$.

For an example of well defined reasons, and complete, sound and optimal explanations, see the later section “Return to the Office at J.P. Morgan”.

Solving EXPRES Tasks using MILP

We propose to use MILP to solve EXPRES problems, i.e., to compute the set of reasons that explain why a given preference was unsatisfied. The use of MILP to solve EXPRES problems follows naturally, since we want to compute a set of reasons that optimise a given metric (the order of the preferences used in the reasons), subject to some constraints

(well-defined reasons). Given an EXPRES problem, we formulate it as a MILP as follows.²

$$\text{minimize} \quad \sum_{p \in SAT, as \in INVOLVED(S, u)} x_{p,as,u} * \mathcal{O}(p) \quad (1)$$

subject to the following constraints:

$$\sum_{p \in SAT} x_{p,as,u} = 1, as \in INVOLVED(S, u) \quad (2)$$

$$x_{p,as,u} \leq AFFECTED(p, as), p \in SAT, as \in INVOLVED(S, u) \quad (3)$$

$$x_{p,as,u} \leq RANK(p, u), p \in SAT, as \in INVOLVED(S, u) \quad (4)$$

There is only one type of decision variable, $x_{p,as,u}$, which represents all the possible reasons $\langle p, as, u \rangle$ in the EXPRES problem: $p \in SAT, as \in INVOLVED(S, u)$, and $u \in UNSAT$ which is the unsatisfied preference we want to explain. The variable $x_{p,as,u}$ gets a value of 1 if reason $\langle p, as, u \rangle$ is used in the explanation $\mathcal{E}_{S,u}$ that solves the EXPRES problem, and a value of 0 otherwise. Therefore, the computational complexity of our approach depends on the number of satisfied preferences and the number of assignments to be explained returned by the INVOLVED function.

Expr. (1) models the objective function of our MILP: to minimise the rank of the satisfied preferences used in the explanation’s reasons. This means that the MILP tries to use more important satisfied preferences to explain unsatisfied preferences. This objective function ensures that if the MILP finds an optimal solution, the explanation $\mathcal{E}_{S,u}$ extracted from such a solution is optimal (Def. 11).

Constr. (2) ensures that we provide exactly one reason for each of the assignments returned by $INVOLVED(S, u)$, i.e., for each assignment involved in u being unsatisfied in solution S . Therefore, if the MILP finds an optimal solution, the explanation is complete (Def. 9).

Constr. (3) ensures that we only select reasons where the given assignment is affected by the preference, $AFFECTED(p, as) = 1$. Finally, Constr. (4) ensures that we only select reasons where more (or equally) important preferences are used to justify less (or equally) important unsatisfied preferences. These two constraints ensure that, if the MILP finds an optimal solution, the explanation is sound (Def. 10).

In some cases, we might be interested in computing more than one explanation for a given EXPRES task. This could be the case when different users have a subjective order over the preference types that is different from the Principal’s order (\mathcal{O}), making them prefer some reasons/explanations over others. To compute multiple explanations, we can iteratively run the MILP, forcing the previously found solutions to not be accepted. In this way, we get explanations with equal or higher cost (lower quality) at each iteration.

²Without loss of generality, we model EXPRES to generate explanations for a single unsatisfied preference, but the problem can be generalised to generate explanations for all the unsatisfied preferences in a PRES problem.

Note that this MILP formulation is general and can be used to generate explanations for any EXPRES task regardless of the preferences in the associated PRES problem. Depending on the preferences in the PRES problem and its interactions, one just needs to appropriately define the INVOLVED and AFFECTED functions that describe when assignments are involved in or are affected by unsatisfied and satisfied preferences, respectively. Then the MILP automatically generates the set of reasons that better explain an unsatisfied preference.

Parsing and Clustering EXPRES Solutions

Explanations are the output of a cognitive process, meant to identify the necessary information to justify an event, and a social process, meant to convey that information to an explainee (Miller 2019). So, once MILP provides us with a solution to the EXPRES problem, i.e., it identifies the reasons why u was unsatisfied in the schedule S , it is crucial to make that solution understandable for any user, not only for expert ones. In this section we present a possible approach, which we validated with users, as described later, to parse the MILP explanations in natural language. Conveying MILP solutions can be challenging, especially (i) when the INVOLVED function returns many assignments that need to be explained; and (ii) because the variables in the MILP solution contain information about the agents and their preferences: to include them in an explanation may be beneficial in some cases, but harmful in others where the privacy of the agents needs to be preserved (Kraus et al. 2020).

According to the application context, we recommend the definition of natural language templates describing constraints, preferences and reasons. For instance, a reason $\mathcal{R} = \langle p, as, u \rangle$ can be parsed with the following template: “ $ag(p)$ was assigned $r(as)$ on $t(as)$ instead of $ag(u)$ because to satisfy $\tau(p)$ is more important than to satisfy $\tau(u)$ ”. As we mentioned, the solution returned by the MILP contains very granular information (for each assignment), even after parsing it in natural language. This can be useful for Principals, as it allows them to understand all the details, but might yield explanations that are tedious and difficult to interpret for general users. For this reason, we suggest to aggregate the reasons according to time slot and preference type before parsing: e.g., if there are n reasons $\langle p, as, u \rangle$ justifying the assignments of n agents having satisfied preferences of the same type on the time slot t , we could have “ $ag(p_1), \dots, ag(p_n)$ were assigned on $t(as)$ instead of $ag(u)$ because to satisfy $\tau(p)$ is more important than to satisfy $\tau(u)$ ”.

Finally, given an explanation $\mathcal{E}_{S,u}$ of why the preference u was not satisfied by the solution S to the PRES problem, we parse it as follows: “ u could not be satisfied because [list of constraints \mathcal{C}] [list of reasons in $\mathcal{E}_{S,u}$]”. Lastly and if the scenario requires it, we recommend to remove any identifying reference to the agents whose satisfied preferences are mentioned in the explanation, in order to better preserve their privacy. For a complete example of parsed explanations, see the next section, where we discuss an EXPRES problem and solution in the context of workforce scheduling, an application of interest at J.P. Morgan.

Employee	Min-Max Days	Preferred	Group	Meetings	OOO
Edith	1 - 2	Th	1	17/11	
George	4 - 5		1		15/11
Han	3 - 4	W,Th,F	1	17/11	
Bob	5 - 5		2	18/11	
Charlie	4 - 4	F	2		17/11
Daphne	2 - 3	M,Tu,Th	2	15/11	
Alice	2 - 4	M,Tu	3	18/11	
Fei	3 - 4	W,Th,F	3	15/11,17/11	

Table 1: Scheduling preferences of the employees in the working example.

	Mon 15 Nov	Tue 16 Nov	Wed 17 Nov	Thu 18 Nov	Fri 20 Nov
Edith					
George					
Han					
Bob					
Charlie					
Daphne					
Alice					
Fei					

Figure 2: Solution to the PRES problem in the working example.

Return to the Office at J.P. Morgan

In this section we exemplify the EXPRES problem formulation and solution through MILP that we have thus far introduced in a general manner. In order to do this, we refer to a real-world scenario, namely *return to the office* at J.P. Morgan. As outlined in the introduction, in this PRES problem a set of employees (Ag) needs to be assigned to a limited number of desks R (that is the only resource type) over a fixed time period (T). In the example we discuss, summarised by Tab. 1 and Fig. 2, we consider 8 employees and a time period of one working week (the 15th to the 20th of November, 2021). Coloured cells mark the assignments of employees to the office, with different colours representing different working groups (a set of agents that need to be co-assigned). The company, which acts as Principal, defines the following set of constraints \mathcal{C} (in the depicted example, $n_{desks} = 5$). For simplicity, here we assume assignments to be binary variables, i.e., $y_{ag,r,t} = 1$ if $\langle ag, r, t \rangle \in S$, and 0 otherwise.

$$\sum_{ag \in Ag} y_{ag,r,t} = n_{desks}, r \in R, t \in T \quad (5)$$

$$\sum_{t \in T} y_{ag,r,t} \leq \maxDays(ag), r \in R, ag \in Ag \quad (6)$$

$$y_{ag,r,t} = 0 \quad \text{if } \text{dayOut}(ag, t), ag \in Ag, r \in R, t \in T \quad (7)$$

Constr. (5) ensures that the sum of the employees assigned to a desk on the same day is equal to the number of available desks defined by the Principal. Constr. (6) ensures that employees are not assigned to the office more days than the maximum they requested. Constr. (7) ensures that employees are not assigned on the days they are out of office, for example due to vacations or personal matters.

Regarding the employees' preferences, we consider the following totally ordered set of preferences types $\mathcal{O} = \langle \tau_{min} \prec \tau_{meet} \prec \tau_{group} \prec \tau_{pref} \rangle$, as imposed by the Principal in J.P. Morgan, where:

- τ_{min} represents a preference type where an employee asks to have a desk at least $n \in \mathbb{N}$ days over the time period T (instantiated as $p = \langle min, ag, n \rangle$).
- τ_{meet} represents a preference type where an employee asks to have a desk on a given day because he/she has an important meeting (instantiated as $p = \langle meet, ag, t \rangle$).
- τ_{group} represents a preference type where an employee asks to have a desk together with another employee on a given day because they need to collaborate (instantiated as $p = \langle group, ag_1, ag_2, t \rangle$).
- τ_{pref} represents a preference type where an employee asks to have a desk on a given weekday, e.g., for personal convenience (instantiated as $p = \langle pref, ag, t \rangle$).

We can explain any unsatisfied preference, but in our example we focus on explaining $\hat{u} = \langle pref, Edith, Thursday \rangle$, i.e., why Edith's preferred day on Thursday could not be satisfied.

We instantiate the INVOLVED and AFFECTED functions as follows. INVOLVED(S, u) returns the set of assignments $S' \subseteq S$ involved in an unsatisfied preference u depending on its type:

- If $\tau(u) = min$, INVOLVED returns all of the assignments in $as \in S$ where $ag(u) \neq ag(as)$, in order to provide a reason why the rest of the desks were better assigned in that way and could not be assigned to $ag(u)$.
- If u is of any other type, INVOLVED returns all of the assignments in S where $t(as) = t(u)$, in order to provide a reason why each of the desks on day t were better assigned to other agents and could not be assigned to $ag(u)$.

For example, when considering \hat{u} , INVOLVED(S, \hat{u}) returns the five assignments of agents to desks on Thursday: $\{\langle Bob, Thursday \rangle, \langle Charlie, Thursday \rangle, \dots\}$.

Likewise, AFFECTED(p, as) checks if a satisfied preference p is related to an assignment as depending on the preference type:

- If $\tau(p) = min$, AFFECTED returns 1 if $ag(p) = ag(as)$ and $|T| - \sum_{t \in T} \text{dayOut}(ag(p), t) = n(p)$. That is, a satisfied preference of this type affects an assignment if the number of days that the employee is available over the time period is equal to his/her minimum, i.e., his/her days at the office cannot be reduced; and if the employee in the preference is the same as the employee in the assignment.
- If $\tau(p) = meet$ or $\tau(p) = pref$, AFFECTED returns 1 if $ag(p) = ag(as)$ and $t(p) = t(as)$. That is, a satisfied preference of these types affects an assignment if the employee was assigned a desk on the day he/she requested.
- If $\tau(p) = group$, AFFECTED returns 1 if $ag_1(p) = ag(as)$, $t(p) = t(as)$, and $\exists as_2 \in S \mid ag_2(p) = ag(as_2)$, $t(p) = t(as_2)$. That is, a satisfied preference of this type affects an assignment if both employees are assigned a desk on the same day.

In the running example, considering a satisfied preference $p' = \langle group, Edith, George, Wednesday \rangle$, we have AFFECTED($p', \langle Edith, desk, Wednesday \rangle$) = 1 because the preference is related to the assignment; and AFFECTED($p', \langle Han, desk, Tuesday \rangle$) = 0 because the preference is not related to that assignment.

When explaining why \hat{u} was unsatisfied in the schedule S^* , to say that Edith could not be assigned on Thursday because Daphne was assigned on Monday is an ill-defined reason, but to refer to Bob being assigned on Thursday due to a meeting is a *well-defined* one (cf. Def. 7). An example of a *complete, sound* and *optimal* explanation (cf. Defs. 9,10,11) is **E**: "The preference could not be satisfied because the 5 available desks were assigned to other people with more important preferences: George, Bob and Charlie due to a minimum number of days per week; Alice due to meetings; and Fei due to 1 working group." EXPRES can generate other explanations, not optimal but still sound and complete, that may be of interest according to the circumstances. For example, Alice's assignment could be justified due to her working group, without mentioning her meeting as in **E**, in case that meeting was confidential, or if Edith considers more convincing explanations regarding working groups (if her subjective order over the preference types is different from the Principal's one).

Evaluation through Simulation

We evaluate our approach by providing explanations in simulated scenarios of our return to the office domain.

Experimental Setting

We generate problems in three configurations of increasing size: 10, 30 and 50 employees over a fixed period of a five-day week. For each configuration, we generate 100 PRES problems with random preferences for each agent. Each agent randomly has 1 or 2 meetings, 1 or 2 preferred days and 1 to 4 working group preferences. The number of days on which an agent has a preference defines his/her preference for minimum number of days; as a maximum number of days, they have a random number between their minimum and 5. For example, if an agent has preferences over 2 different days, its minimum number of days is 2 and its maximum is a random number between 2 and 5. Agents also have a 20% probability of having dates out (1 or 2, randomly picked). We set the number of available desks each day to be 50% of the total number of employees, in line with the policy followed in the actual return to the office at J.P. Morgan.

We optimally solve all of these PRES problems and automatically compute all of the satisfied and unsatisfied preferences. We have an average of 31.4 satisfied and 28.7 unsatisfied preferences in the problems with 10 employees; 89.4 and 141.7 in the problems with 30 employees; and 147.5 and 280.1 in the problems with 50 employees. For each problem, we randomly pick one unsatisfied preference of each type (if one exists) to build the EXPRES task to be solved by the MILP. This gives us a total of 1200 (3 agent configurations, 100 problems on each configuration, and 4 unsatisfied preferences on each problem) EXPRES tasks to solve. We run

the MILP on each task with a timeout of 30 seconds, or after 1000 explanations are produced. We use CPLEX³ to solve the MILP. Experiments were run in Intel(R) Xeon(R) CPU E3-1585L v5 @ 3.00GHz machines with 64GB of RAM.

Scalability Evaluation

First, we evaluate the scalability of our approach, by measuring the time needed to compute the first (optimal) explanation and the number of explanations provided for each scenario (see left plot of Figure 3). EXPRES tasks can be solved in a reasonable time for all the configurations: even with 50 agents, the solver returns the optimal explanation on average in less than 0.5 seconds. Regarding the number of generated explanations, we expect it to increase as we increase the number of agents, because with more agents there are more satisfied preferences and more ways of combining them to justify unsatisfied preferences. However, when increasing the number of agents, the complexity of the problems also increases, allowing less problems to be solved within the time bound. With 10 agents, we are able to generate all of the sound explanations in less than 1 second. For these problems, we can compute an average of 19.7 different explanations, with some EXPRES tasks where we can compute more than 250 explanations. With 30 and 50 agents we cannot generate all of the sound explanations within the given time bound. Without timing out, the solver generates an average of 157.2 and 402.3 different explanations with 30 and 50 agents, respectively. We conclude that EXPRES is scalable because, even though not *all* the sound explanations are found, the user will ultimately be interested in only one: how to identify and learn the preferred explanation for a user is part of our future work.

Explanations per Unsatisfied Preference Type

Next, we analyse how the number of explanations is influenced by the type of unsatisfied preference that needs to be explained (see the central plot of Figure 3). We consider only the problems with 10 agents given that it is the only configuration for which we can compute all of the explanations in all of the problems. As expected, we can compute more explanations if we explain less important, unsatisfied preferences such as those of type τ_{pref} , where we can produce an average of 44.5 explanations for each unsatisfied preference. This happens because there is a larger number of more important preferences that can be used to explain these unsatisfied preferences. Coherently, there are fewer ways of explaining more important preferences such as preferences of type τ_{meet} being unsatisfied, since they can only be explained by using preferences of type τ_{meet} or τ_{min} , that are equally or more important. In fact, there are approximately 35% of problems involving an unsatisfied preference in τ_{min} where we cannot produce any explanation. This is because these problems were extremely overconstrained and the solution could not satisfy many requests for a minimum number of days. Since our reasons require satisfied preferences, there were some cases where we could not produce any explanation.

³<https://www.ibm.com/analytics/cplex-optimizer>

Understanding Explanation Structure

Finally, given an EXPRES problem and the set of generated explanations, we investigate how different the explanations within that set are. We focus on the 100 problems with 10 agents where the selected unsatisfied preference is of type τ_{pref} , since these are the cases where we can generate more explanations. Then, given two explanations \mathcal{E}_1 and \mathcal{E}_2 , we measure their distance as $|\mathcal{E}_1 \setminus \mathcal{E}_2|$, i.e., the number of different reasons they contain. Given the 5 available desks, the distance is bounded between 0, if both explanations are the same, and 5, if the two explanations have no common reason. We compute this pairwise distance for all of the pairs of explanations produced in an EXPRES problem (see the right-hand plot of Figure 3). As we can see, the standard deviation of the explanations' pairwise distance in each problem is close to 1, meaning that explanations tend to vary for one out of five reasons. The average distance between explanations in a problem is 2.3, and the average maximum distance is 3.8, with more than 75% of the problems having a maximum distance between explanations higher than 2.7. These results suggest that many of the explanations we provide for a problem only differ in 1–2 reasons (20–40% of the explanation). However, most of the time our set of explanations contains at least two explanations that are really different, differing in 3–4 reasons (60–80% of the explanation).

Evaluation through User Studies

We have designed and implemented two user studies in order to understand (i) how humans solve EXPRES tasks, (ii) how automated generated explanations compare to human generated ones, and (iii) what type of explanations are preferred by users of the tool. In particular, we wanted to validate the following hypotheses:

- **Hp1:** The EXPRES framework produces explanations faster than humans.
- **Hp2:** Humans find automatically generated explanations at least as satisfying as the human generated ones.

The first user study (US1), where we collected human-generated explanations, aimed to discuss Hp1; the second user study (US2), where we compared human-generated and EXPRES-generated explanations, aimed to discuss Hp2. We defined two PRES scenarios with different complexities by considering 8 employees in the first scenario and 20 employees in the second scenario, to be scheduled over a week. In US1 we showed all the preferences and the entire team's weekly schedule; in US2 we showed only one individual set of preferences and one individual weekly schedule.

User study 1

US1 consists of individual interviews with people (N=10) who have previously actively interacted with the workforce scheduling tool deployed at J.P. Morgan, i.e., team managers or assistants who generated schedules through the tool. The interviews took place virtually. After enquiring about the participant's familiarity with the tool (on a 5-point Likert scale, avg=4.9), the two scenarios were fully disclosed. The participants were asked to justify to one selected fictitious

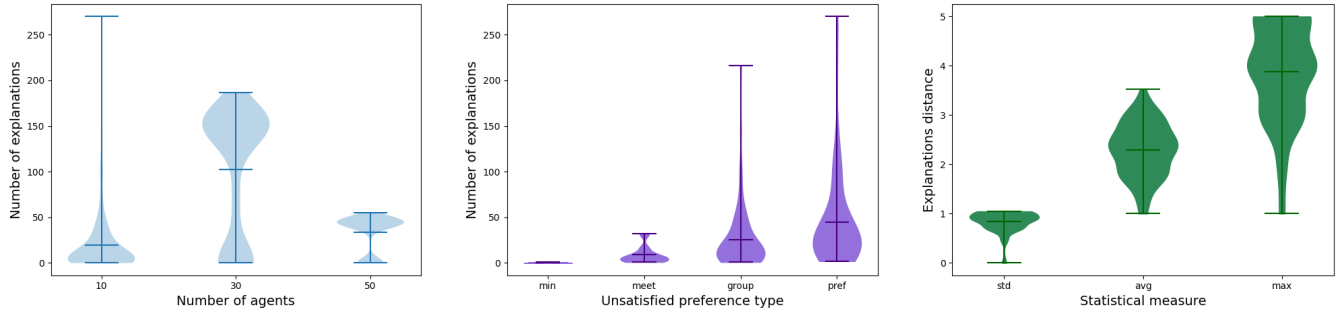


Figure 3: The figure on the left shows the number of explanations generated within 30 seconds as we increase the number of agents. The figure in the middle shows the number of explanations we can compute depending on the unsatisfied preference type in problems with 10 agents. The figure on the right shows different statistical measures over the distance between the generated explanations. For each violin plot, the central horizontal line depicts the mean of the distribution, while the other two lines depict the minimum and maximum values. A wider shadow indicates that more points have that value.

	Scenario 1			Scenario 2		
	avg	std dev	EXPRES	avg	std dev	EXPRES
time (s)	195.9	80.8	$\ll 1$	186.3	96.2	0.8
difficulty	2.4	1.4	-	2.7	1.1	-

Table 2: Results of user study 1. Time is expressed in seconds, difficulty is expressed on a 5-point Likert scale (1=very easy, 5=very difficult).

employee why their preference could not be satisfied in the team’s schedule (e.g., “Why wasn’t Edith assigned to the office on Thursday 18th November, as she requested?”). In the meeting chat, the participants wrote an explanation for the unsatisfied preference and then evaluated the difficulty of providing that explanation. We tracked the time that each participant required to provide each an explanation.

Tab. 2 shows a quantitative overview of the results. All of the participants spent considerably more time than EXPRES to provide an explanation (Hp1 is confirmed): despite scenario 2 being more complex in terms of quantity and density of information and being reported more challenging to explain (avg difficulty=2.7 vs 2.4 in scenario 1), 6 participants were quicker to provide the second explanation than the first one. We suppose this could be due to the participants being more familiar with the task by the time they faced the second scenario. Regarding the quality of explanations, in each scenario we gathered very different justifications (see Tab. 3 for a sample, and an extended version of this work⁴ for a complete list of the collected explanations), sometimes more explicit and detailed, sometimes more implicit and general. However, no explanation can be considered complete, sound or optimal, according to Defs. 9, 10 and 11.

User study 2

US2 consists of an online questionnaire with people (N=28) who have passively interacted with the tool, i.e., employees whose schedule was generated by the tool, within J.P.

⁴<https://arxiv.org/abs/2203.08895>

Scenario 1

H4: Too many of the team members have requested to be in the office on a Thursday.

H5: Edith wasn’t assigned into the office because she hadn’t requested the dates in on the 18th or because since her minimum was 1 and max was 2 and it maxed out the days she can come in

Scenario 2

H4: Due to being a large group and many of the employees requesting the same day, not everyone could get their desired day in the office.

H5: 3 people had dates in, another 2 had 5 days min, the rest are perhaps a combination of all the other factors, meaning they had to be in instead of Ivan

H6: He was limited by the seat availability that day, members of two other groups has selected the 1th specifically to be in so they took preference

Table 3: Details of some of the human-generated explanations selected from US1 and included in US2.

Morgan. Each participant was shown, sequentially, the preferences and the schedule of one fictitious employee (e.g., Edith’s) from the two scenarios previously defined. For each scenario, 6 explanations were listed in a random order: 3 were generated by EXPRES (E1-E3, see Tab. 4; explanations are anonymised and parsed as shown at the end of “Return to the Office at J.P. Morgan”) and 3 were selected from the human generated ones in US1 (H4-H6, see Tab. 3). When selecting the explanations to include in this study, we aimed to represent the diversity, in terms of structure and reasons mentioned, of the pool of explanations that were available, in order to explore the participants’ appreciation of different combinations of reasons. In both scenarios, E1 is the optimal explanation (cf. Def. 11). Each participant was asked to first select and then rank the three explanations that were the most satisfying. We report the results in Tab. 5.

Participants showed a strong preference in both scenarios for the EXPRES explanations, which were selected significantly (t-test with pvalue=0.01) more often than the human-generated ones (Hp2 is confirmed). Looking at the results in greater detail, we see that E1 has been the most selected

Reason type	Scenario 1 ($n=5$ desks)			Scenario 2 ($n=12$ desks)		
	E1	E2	E3	E1	E2	E3
R_{min}	3	1	1	3	0	3
R_{meet}	1	0	2	2	0	0
R_{group}	1	4	1	7	12	8
R_{pref}	0	0	1	0	0	1

Table 4: Details of the reasons part of the explanations generated by EXPRES and included in US2. *Reason type* refers to the preference-type mentioned as a justification for the unsatisfied preference.

		E1	E2	E3	tot.E	H4	H5	H6	tot.H
s1	# selections	22	19	14	55	17	5	7	29
	rank score	47	36	24	107	37	10	14	61
s2	# selections	23	17	11	51	14	8	11	33
	rank score	55	28	17	100	34	12	22	68

Table 5: Results of US2. The rank score is $3x_1 + 2x_2 + x_3$ (x_i is the n. of times an explanation has been ranked i).

explanation in both scenarios (by 78.6% and 82.1% of the participants). We interpret this as a predilection for explanations that are sound, complete and as consistent as possible with the Principal’s total order of preferences. Regarding the human-generated explanations, most people appreciated H4, which in both scenarios was the most general and vague explanation. This suggests that brief and simple explanations can satisfy the general user who does not look for detailed justifications.

Related Work

Explanations are essential for humans to understand the outputs and decisions made by AI systems (Core et al. 2006; Miller 2019). There exist many works that provide explanations for different AI use cases ranging from automated planning (Fox, Long, and Magazzeni 2017; Chakraborti et al. 2019) to machine learning (Carvalho, Pereira, and Cardoso 2019) or deep learning (Samek, Wiegand, and Müller 2017). Explanations are also crucial in multi-agent environments where some extra challenges arise, such as privacy preservation or fairness (Kraus et al. 2020). Scheduling of multiple agents is one of these problems, and explaining the resulting schedules is not a trivial task.

In (Agrawal, Yelamanchili, and Chien 2020), the authors propose CROSSCHECK, a tool that (i) diagnoses scheduling failures in the context of a Mars Rover mission, and (ii) guides users about which constraints need to be altered in order for the activity to be successfully scheduled. Their tool focuses on visualisation and improving user experience with the scheduler, but could hardly be adapted to provide explanations for multiple users having competing preferences.

In (Zahedi, Sengupta, and Kambhampati 2020), the authors propose AITA, a centralised Artificial Intelligence Task Allocation that simulates a negotiation between the agents. If unhappy with the recommended allocation, an agent can question it using counterfactuals; these will be refuted

by AITA, which explains, using negotiation trees, how the agent’s proposal would entail a worse-off allocation than the recommended one. Despite not formally allowing counterfactual queries, in this paper we still enable users to get explanations (i) that are specifically targeted for preferences that are unsatisfied in the recommended schedule, and (ii) that contain more interpretable information (AITA refers to the overall allocation cost, while we include other agents’ satisfied preferences). Finally, the authors discuss the length of the explanations when 2-4 agents are involved in the scheduling process but do not share any results on the scalability of their approach (more agents and more tasks).

In (Cyras et al. 2019), the authors explain schedules using argumentation frameworks. They can explain (i) why a solution is not feasible or suboptimal (we assume feasible and optimal solutions are given); or (ii) why a preference was not satisfied in the solution, as in our EXPRES problem formulation. In order to provide the explanations, they need to manually generate the attack graphs, i.e., the relationship between the preferences and the assignments. This is similar to the effort needed to define the rules inside our INVOLVED and AFFECTED functions. A key difference between both works is that they are restricted to makespan scheduling problems with a very limited number of preferences. EXPRES can be used to generate explanations in any scheduling problem where there is a totally ordered set of preferences. On the evaluation side, they do not report any experiment. An interactive tool was presented (Čyras, Lee, and Letsios 2021), but its validation was left for future work.

Conclusion and Future Work

In this paper, we introduced the EXPRES framework, an approach to explain why a given preference was unsatisfied in a schedule. We framed this problem as an optimisation task that aims to find the best set of reasons that can explain an unsatisfied preference, and we solved it using MILP techniques. Then we showed how to group and translate the raw explanations in order to be easily interpreted by humans as well as to preserve agents’ privacy. Experimental results through simulations showed that EXPRES can efficiently scale. Finally, a set of user studies within J.P. Morgan showed how employees interacting with a workforce scheduling tool preferred our automatically-generated explanations over human-generated ones.

Currently, we assume a totally ordered set of preferences that is always respected in any optimal solution. This simplifies the definition of the INVOLVED and AFFECTED functions, but that assumption does not hold in all scheduling problems. We will explore how to treat problems where only partial orders over the preferences exist, and how to define reasons that justify unsatisfied preferences through a chain of satisfied ones, without limiting to only one. Also, we would like to investigate (i) whether providing explanations improves the users’ satisfaction with their schedules (Bradley and Sparks 2009), and (ii) how to learn the subjective preference order of a user and the type of explanations they prefer, in order to generate even more tailored explanations (Soni, Sreedharan, and Kambhampati 2021).

Acknowledgements

This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates (“JP Morgan”), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

- Agrawal, J.; Yelamanchili, A.; and Chien, S. A. 2020. Using Explainable Scheduling for the Mars 2020 Rover Mission. *CoRR*, abs/2011.08733.
- Bradley, G. L.; and Sparks, B. A. 2009. Dealing with Service Failures: The Use of Explanations. *Journal of Travel & Tourism Marketing*, 26(2): 129–143.
- Carvalho, D. V.; Pereira, E. M.; and Cardoso, J. S. 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*, 8(8): 832.
- Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D. E.; and Kambhampati, S. 2019. Explicability? Legibility? Predictability? Transparency? Privacy? Security? The Emerging Landscape of Interpretable Agent Behavior. In *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling (ICAPS’19)*, Berkeley, CA, USA, July 11-15, 2019, 86–96. AAAI Press.
- Core, M. G.; Lane, H. C.; van Lent, M.; Gomboc, D.; Solomon, S.; and Rosenberg, M. 2006. Building Explainable Artificial Intelligence Systems. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence (AAAI’06) July 16-20, 2006, Boston, Massachusetts, USA*, 1766–1773. AAAI Press.
- Čyras, K.; Lee, M.; and Letsios, D. 2021. Schedule Explainer: An Argumentation-Supported Tool for Interactive Explanations in Makespan Scheduling. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, 243–259. Springer.
- Cyras, K.; Letsios, D.; Misener, R.; and Toni, F. 2019. Argumentation for Explainable Scheduling. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI’19, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2752–2759. AAAI Press.
- Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable Planning. *CoRR*, abs/1709.10256.
- Gunawan, A.; Ng, K. M.; and Poh, K. L. 2007. Solving the Teacher Assignment-course Scheduling Problem by a Hybrid Algorithm. *International Journal of Computer, Information, and Systems Science, and Engineering*, 1(2): 136.
- Kraus, S.; Azaria, A.; Fiosina, J.; Greve, M.; Hazon, N.; Kolbe, L.; Lembcke, T.-B.; Muller, J. P.; Schleibaum, S.; and Vollrath, M. 2020. AI for Explaining Decisions in Multi-agent Environments. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI’20)*, volume 34, 13534–13538.
- Legrain, A.; Bouarab, H.; and Lahrichi, N. 2015. The Nurse Scheduling Problem in Real-life. *Journal of medical systems*, 39(1): 1–11.
- Miller, T. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artif. Intell.*, 267: 1–38.
- Samek, W.; Wiegand, T.; and Müller, K.-R. 2017. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *arXiv preprint arXiv:1708.08296*.
- Soni, U.; Sreedharan, S.; and Kambhampati, S. 2021. Not all Users are the Same: Providing Personalized Explanations for Sequential Decision Making Problems. *CoRR*, abs/2106.12207.
- Watson, J.; Beck, J. C.; Howe, A. E.; and Whitley, L. D. 2003. Problem Difficulty for Tabu Search in Job-shop Scheduling. *Artif. Intell.*, 143(2): 189–217.
- Zahedi, Z.; Sengupta, S.; and Kambhampati, S. 2020. ‘Why not give this work to them?’ Explaining AI-Moderated Task-Allocation Outcomes using Negotiation Trees. *CoRR*, abs/2002.01640.