Explanation Framework for Optimization-Based Scheduling: Evaluating Contributions of Constraints and Parameters by Shapley Values

Yuta Tsuchiya, Masaki Hamamoto

Research and Development Group, Hitachi, Ltd. 1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
yuta.tsuchiya.gf@hitachi.com

Abstract

Although automated planning and scheduling systems based on optimization models are increasingly being adopted into socially responsible tasks, the derived plan is often counterintuitive under complicated considerations. Users will claim the right to know the reason for "Why did the optimal plan include something or not include something else (that I would have chosen)?" Explanations of constraints and parameters that cause the unexpected plan derivation can play an important role in building trust between users and the scheduling system. However, existing approaches require an assumption of a specific problem setting, and have not addressed quantitative analysis for multiple types of factors. In this paper, we propose a general explanation framework to quantitatively evaluate the effect of constraints and parameters on the plan derivation by applying the concept of Shapley values, which satisfy the desirable axioms for explanations. The coalitional game based on optimization models is formulated to calculate the contributions of these factors to the fulfillment of values or conditions in which users are interested. Through numerical experiments of the typical personnel assignment problem, we show that our framework can identify the major causes efficiently under various parameter settings and provide directly understandable explanations compared to the basic contrastive explanations.

Introduction

Automated planning and scheduling based on mathematical optimization provide a solution to optimize a set of objectives that satisfy several constraints. As the performance of optimization algorithms becomes more advanced, these systems are applied to real-world decision-making tasks with significant social responsibility. For example, the optimization of personnel assignments is vital to not only the improvement of employee working conditions but also the success of high-risk tasks represented by medical care (Legrain, Bouarab, and Lahrichi 2015). Scheduling failures on a Mars rover mission might result in massive time and financial losses (Agrawal, Yelamanchili, and Chien 2020).

However, the solutions derived by optimization techniques are often counterintuitive because of the complicated considerations. In the context of personnel assignment, employees will claim the right to know the reason, "Why am I assigned to job 1 while I am more suited for job 2?" Even if the objective function was optimized, the lack of employees' agreement would be sufficient to prevent a plan from being executed.

The person in charge of ordering the transfer based on the derived plan has to explain the reason for the assignment to job 1. In this paper, we focus on the user query of "Why did the optimal plan include something or not include something else (that I would have chosen)?" which is considered as one of the most fundamental questions in the field of scheduling (Fox, Long, and Magazzeni 2017). However, even theoretically well-understood algorithms represented by linear programming can still yield decisions that are hard to build trust; they only present the derived plan and do not explain why it was selected as the optimal solution.

To address the issue of trustworthiness, the eXplainable AI Planning (XAIP) community has proposed numerous explanation algorithms and successful interfaces (Chakraborti, Sreedharan, and Kambhampati 2020). Although it is too complex to clarify what an explanation should actually contain, revealing constraints and parameters that directly cause the unexpected plan derivation can play a key role in getting a deeper understanding of the problem setting. The resulting plan varies greatly depending on the presence or absence of constraints and the value of input parameters, e.g., coefficients and weights for decision variables (Gupta, Genc, and O'Sullivan 2022).

Several studies have discussed basic architectures to extract those factors of optimization models that affect derivation of the solution. (Pozanco et al. 2022; Burt, Klimova, and Primas 2018). However, existing approaches require an assumption of a specific problem setting, and methods to quantitatively evaluate multiple factor types have not been well explored.

On the other hand, XAI techniques for machine learning models have been developed rapidly after the launch of the DARPA project (Gunning and Aha 2019). In particular, the concept of the Shapley value, which additively distributes allocation credits to players in coalitional games, is commonly-used for quantifying the contribution of features to the prediction derived by the model (Lundberg and Lee 2017). For optimization problems, Shapley values are also utilized to calculate resource contributions for optimizing a network configuration (Iturralde et al. 2011) and as an alternative to Sobol's sensitivity analysis index (Song, Nelson, and Staum 2016). Although the Shapley value is a model-agnostic and general explanation tool, no method has been proposed that applies this value to explain the reason for a plan derivation.

In this paper, we propose an explanation framework to quantitatively evaluate the effect of constraints and parameters on the plan derivation by Shapley values. We define a coalitional game based on optimization models in accordance with the types of input factor candidates and user questions. Shapley values of these factors are calculated as the contributions to the fulfillment of values or conditions in which users are interested. This framework enables users to analyze the impactful constraints and parameters along with the magnitude of their contributions, and simplifies the understanding process of the reason why the counterintuitive plan was selected, which is often performed manually with considerable effort and time.

Since the computation of Shapley values does not depend on a specific class of tasks and algorithms, our method can be adopted a wide range of optimization models and user queries. To illustrate the explanation process, we applied our framework to the typical optimization problem: personnel assignments using linear programming. The result shows that this study can provide an explanation tool to extract the important factors under various parameter settings and facilitate the implementation of the plan in high-responsibility decisions.

In summary our contributions are as follows:

- Propose the general and quantitative explanation method for answering “Why did the optimal plan include something or not include something else (that I would have chosen)?” by calculating the contributions of constraints and parameters calculated as Shapley values.
- Provide the framework for formulating a coalitional game based on mathematical optimization models to calculate Shapley values of input factors.
- Implement a case study through numerical experiments of typical personnel assignment problems. We discuss how to interpret the obtained contributions as the impact on the plan derivation.

The rest of this paper is organized as follows. We first describe typical explanation methods in the field of XAIP and clarify our focus. We then formalize the general mathematical optimization problem and introduce the definition of the Shapley value that satisfies the desirable axioms for explanations. After that, the procedure for the calculation of Shapley values to the optimization task is described. Then, we empirically show that our proposed method can extract factors to the plan derivation efficiently for typical personnel assignment problems through numerical experiments. Finally, we draw our main conclusions and outline future work.

Related Works

Explanations are vital to building trust between AI for automated planning and humans. Fox, Long, and Magazzeni (2017) established the concept of XAIP, and showed important questions from users that XAIP should address. Since the required explanations depend on the question types (Soni, Sreedharan, and Kambhampati, 2021), various types of explanations have been discussed.

To answer the users query of “Why did the optimal plan include something or not include something else (that I would have chosen)?,” a basic approach would be contrastive explanation to demonstrate a flaw in a plan that adopts the alternative proposed by users compared with the optimal plan (Cashmore et al. 2019). However, since the explanation cannot explicitly provide impactful constraints and parameters, the process of understanding can take a considerable amount of time, especially for non-experts.

Counterfactual explanations derive the change to the problem setting that would have resulted in the alternative plan indicated by users. Several studies have discussed how to generate appropriate counterfactuals efficiently. Generalized inverse combinatorial optimization was proposed to minimize objective function deterioration of counterfactuals (Korikov Shleyfman, and Beck 2021). Gupta, Genc, and O'Sullivan (2022) presented an explanation that provides applicable changes of the existing constraints with the cost function of counterfactuals to recover feasibility in staffing problems.

In the field of robotics, Brandão, Coles, and Magazzeni (2021) formulated an efficient inverse problem that changes the route plan derived by a robot to the one by a human with small calculation cost. Gragera, García, and Fernández (2022) discussed an explanation scheme when a plan cannot be executed due to a lack of appropriate actions in the Mars rover 's load hauler. It compiles the unresolvable task into a new extended planning task and suggests a repair action to the operator.

An interface to show the explanation results is also an important research target of XAIP. For example, Crosscheck proposes an interface to find the reason for schedule failures in the Mars rover's motion using binary search trees and recursive programming (Agrawal, Yelamanchili, and Chien 2020). A negotiation tree is an interactive framework that repeatedly creates counterfactual plans until humans are satisfied (Zahedi, Sengupta, and Kambhampati 2020).

In (Pozanco et al. 2022), the EXPRES framework was proposed, which explains why a preference of seat allocation given by users was unsatisfied in an optimal schedule.

The authors defined two kinds of functions to identify involved assignments and satisfied preferences of other agents. However, the framework requires a completely ordered list of preferences that does not hold in all scheduling problem. Burt, Klimova, and Primas (2018) presented an algorithm that imitates a sensitivity analysis to evaluate the feasibility of solutions for various constraint sets to explain why optimal solutions cannot be obtained by mixed integer programming. Although it provides a basic architecture to extract the impactful constraint, there is no specific method for quantifying the effect to the plan derivation.

As indices of quantitative evaluation in mathematical optimization, a global sensitivity analysis has been applied to decompose the model output variance caused by the uncertainty in the contribution of inputs (Hall and Posner 2004). The experimental design is a methodology to observe the effect of input factors with a small number of experiments by an orthogonal table (Hedayat, Sloane, and Stufken 1999). However, assumptions of these approaches may not hold if there are strong interaction effects between input factors.

The Shapley value has been applied in the field of sensitivity analysis because of its ability to output contributions that incorporate interaction effects (Benoumechiara and Cosaque 2019). It is also used to calculate the contribution of features in a machine learning model as a standard index that satisfies the axioms for explanation (Lundberg and Lee 2017). In this paper, we discuss a domain-agnostic explanation method to quantitatively evaluate multiple constraints and parameters by applying the Shapley value.

Background

Explanation Task for Mathematical Optimization

We first formalize a scheduling problem that is the subject of explanation. Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of decision variables, $C = \{c_1, c_2, \dots, c_M\}$ be a set of constraints, $O = \{o_1, o_2, \dots, o_L\}$ be a set of objective functions, and $P = \{P^C, P^O\}$ be a set of parameters such as coefficients, constants, and weights. P^C represents the parameters for constraints C and P^O is that for objectives O. With the aforementioned set of variables, the typical optimization problem, e.g., linear programming, can be defined as follows:

$$\max\{\,\boldsymbol{P}_{\boldsymbol{x}}^{O}\boldsymbol{x} \mid \boldsymbol{P}_{\boldsymbol{x}}^{C}\boldsymbol{x} \leq \boldsymbol{P}_{const}^{C}, \boldsymbol{x} \geq 0\}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is a vector for decision variables, $\boldsymbol{P}_{\boldsymbol{x}}^{O} \in \mathbb{R}^{L\times N}$ and $\boldsymbol{P}_{\boldsymbol{x}}^{C} \in \mathbb{R}^{M\times N}$ represent a coefficient matrix, and $\boldsymbol{P}_{const}^{C} \in \mathbb{R}^{M}$ is a vector of constants for constraints C.

Optimization model f makes X optimal for O under the given C and P. Let $Solution = \langle X^*, O^*, C^*\rangle$ be a tuple of a plan derived by model f, where X^* is a set of optimized decision variables, O^* is a set of derived objective values, and $C^* = \{c_1^*, c_2^*, \dots, c_M^*\}$ is a set of binary variables that represents whether $c \in C$ is satisfied.

In some cases, the variables of $Solution$ are often counterintuitive under complicated problem setting. Providing explanation to answer user questions such as “Why is the value of x_1 zero?” and “Why is there no optimal plan that satisfies constraint c_1?” is vital to building trust between humans and the scheduling system, especially for socially high-responsible tasks in the real world.

In this paper, we aim to answer the question, “Why did the optimal plan include something or not include something else (that I would have chosen)?” There are numerous approaches to explain the output of models; in particular, it is important to understand which $c \in C$ and $p \in P$ caused the counterintuitive plan derivation by the model f. Related to these factors, users are interested in whether constraint c should be included in the problem setting or whether a specific value of p is suitable for deriving a desired plan.

Thus, we rephrase the question to “What impacts do c and p have on determining the $Solution$?” To the best of our knowledge, it is still challenging to quantitatively evaluate the effect of those factors by existing approaches in the field of XAIP, e.g., contrastive explanation and other basic architectures (Cashmore et al. 2019; Burt, Klimova, and Primas 2018). In the field of machine learning, one of the standard explanation methods is to calculate the contribution of the input features to the model’s prediction by the Shapley value (Chen et al. 2022). In this paper, we propose the introduction of the Shapley value to achieve a general explanation framework for c and p. We assume c and p as input factors of model f, and the values of the variables in $Solution$ as a model output, in order to define a coalitional game based on mathematical optimization models.

Shapley Value

The Shapley value is a concept designed to achieve the fairest allocation of gained profits between several players in coalitional games (Shapley 1953). Let $D = \{1, 2, \dots, d\}$ be a set of players, S be a subset of D, and $v(S)$ is a profit made by S. The relative importance of player i is calculated by averaging the difference of $v(S)$ over all possible S. Subsequently, the Shapley value ϕ_i, which assigns credit to each player i, is derived as follows:

$$\phi_i = \sum_{S\subseteq D\setminus\{i\}} \frac{|S|!\,(|D|-|S|-1)!}{|D|!}\big(v(S\cup\{i\}) - v(S)\big). \tag{2}$$

The Shapley value satisfies the properties of the following axioms:

- **Efficiency:** The sum of individual contributions should equal the total profit achieved through the cooperation of all the players.

- **Monotonicity:** If a player always contributes more to one game than another, they should receive a higher level of credit.
- **Symmetry:** If a player always contributes as much as another player, they should have an equal level of credit.
- **Dummy:** If a player does not contribute to gain profit, it should have zero credit.

These four axioms have been considered as desirable properties for fair reward distribution in the game theory (Shapley 1953). In the field of machine learning, these simple and intuitive axioms are widely recognized as a necessary quality for feature importance metrics (Fryer, Strumke, and Nguyen, 2021). Therefore, we adopt the Shapley value as the contribution of the input factors to the optimization model.

Explanation Approach by Shapley Values

Coalitional Game Based on Optimization Models

In this section, we introduce how to calculate the Shapley value of input factors. First, we must define a coalitional game that consists of players D and profit $v(S)$ on the basis of the optimization model f.

In accordance with the users' interest, we define player $i \in D$ of the constraint c and the parameter p as follows:

- The constraint c: let $\boldsymbol{B}^C = \{B^{c_1}, B^{c_2}, \dots, B^{c_M}\}$ be binary variables that represents the inclusion of c in the problem setting (1: including the constraint in the problem, and 0: excluding the constraint) as player i.
- The parameter p: the specific value of p itself is suitable for player i.

Also, instead of considering each c and p as an independent player, it is possible to group them together as one player. For example, a player vector of coefficient parameters for c_1 is as follow:

$$player\ i = \left(P_{x_1}^{c_1}, P_{x_2}^{c_1}, \dots, P_{x_N}^{c_1}\right). \tag{3}$$

The profit $v(S)$ should be defined as a numeric or boolean value by the users' question. The original question, "The optimal plan includes something or does not include something else," can be represented by the variables of $Solution = \langle X^*, O^*, C^* \rangle$. Thus, we propose the following formulations for each case where the question is about a continuous value or conditional expression:

- For questions regarding the continuous value, such as "Why is X^* or O^* the specific value as shown in the optimal plan?," the obtained value in $Solution$ for the interested variable itself is suitable for the profit $v(S)$.
- For questions regarding the conditional expression, such as "Why is x_1 (not) selected in $Solution$?," "Why is constraint c_1 (not) satisfied in $Solution$?," and "Why is there no $Solution$," we can define $v(S) = 1$ if the condition is satisfied in $Solution$ obtained by a player set of S, and $v(S) = 0$ if it is not.

Algorithm 1: Shapley values for an optimization model

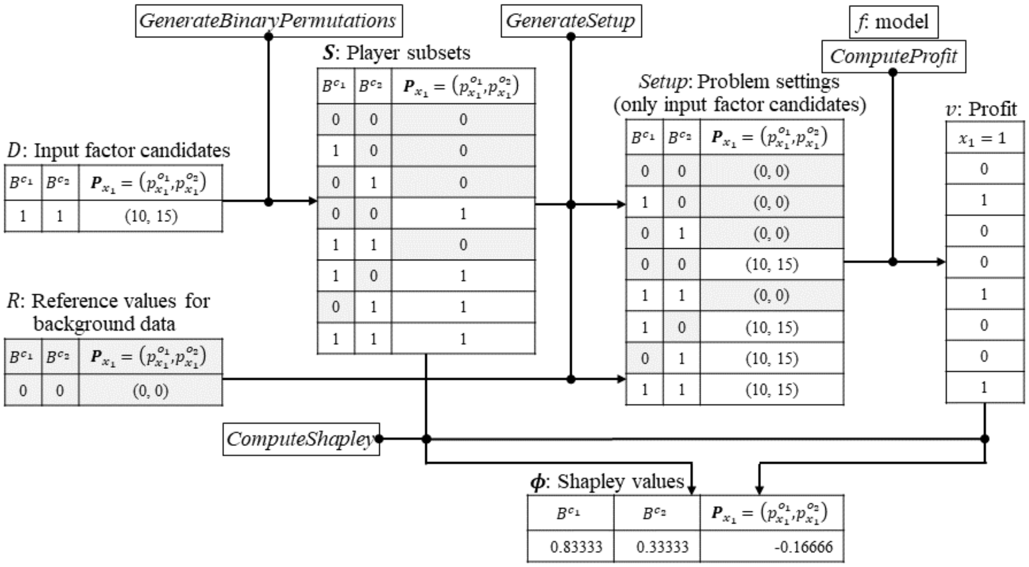
```
Input:
    $D$: List of input factor candidates $C$ and $P$
    $R$: List of background data for input factor candidates
    $Problem = \langle X, C, O, P \rangle$: Tuple of variables in the
    original problem setting.
    $f$: An optimization model
Output: List of Shapley values $\boldsymbol{\phi}$ for all $i \in D$
$\boldsymbol{\phi} \leftarrow EmptyList()$
$\boldsymbol{S} \leftarrow GenerateBinaryPermutations(D)$
$perturbations \leftarrow EmptyList()$
for each $S \in \boldsymbol{S}$ do
    $Setup \leftarrow GenerateSetup(Problem, S, D, R)$
    $Solution \leftarrow f(Setup)$
    $v \leftarrow ComputeProfit(Solution)$
    $perturbations.add((S, v))$
end for
for each $i \in D$ do
    $\phi_i \leftarrow ComputeShapley(perturbations, i)$
    $\boldsymbol{\phi}.add(\phi_i)$
end for
return $\boldsymbol{\phi}$
```

Procedure of Proposed Explanation Approach

In accordance with the definition of coalitional games for the explanation task of an optimization problem, we propose the contribution calculation process as shown in Algorithm 1. First, this algorithm takes D, R, $Problem$, and f as input. D consists of player i, which is a vector or variable of B^c or p. R is background data: to determine the effect of input factors based on the definition of Shapley values as (2), we observe the change in the scheduling solution due to the presence or absence of the factor. We simulate "absence" of player i by replacing the factor in the original problem setting with the reference values it takes in the background data. This value should be carefully determined depending on the problem setting and users' interest. We show the typical guidelines for each player types, B^c and p, as follows:

- B^c: The binary variable B^c is introduced to evaluate how much the constraint affects the solution relative to "no constraint". When the original problem setting defines $B^c = 1$, i.e., the constraint exists, its absence can be directly expressed as the opposite, $B^c = 0$, to indicate that the constraint does not exist.
- p: The reference value for representing the absence of parameter p depends heavily on users' interest. For example, the average, minimum, or maximum value of parameter p in a certain set of O, X, or C would be typical baseline. If there are a certain value defined by users, it should be used.

$Problem$ is a tuple of variables in the original problem setting to derive the optimal $Solution$ by model f. Note that there is no limitation of the optimization model f.



D: Input factor candidates

B^{c_1}	B^{c_2}	$\boldsymbol{P}_{x_1} = (p_{x_1}^{o_1}, p_{x_1}^{o_2})$
1	1	(10, 15)

R: Reference values for background data

B^{c_1}	B^{c_2}	$\boldsymbol{P}_{x_1} = (p_{x_1}^{o_1}, p_{x_1}^{o_2})$
0	0	(0, 0)

$\boldsymbol{S}$: Player subsets

B^{c_1}	B^{c_2}	$\boldsymbol{P}_{x_1} = (p_{x_1}^{o_1}, p_{x_1}^{o_2})$
0	0	0
1	0	0
0	1	0
0	0	1
1	1	0
1	0	1
0	1	1
1	1	1

Setup: Problem settings (only input factor candidates)

B^{c_1}	B^{c_2}	$\boldsymbol{P}_{x_1} = (p_{x_1}^{o_1}, p_{x_1}^{o_2})$
0	0	(0, 0)
1	0	(0, 0)
0	1	(0, 0)
0	0	(10, 15)
1	1	(0, 0)
1	0	(10, 15)
0	1	(10, 15)
1	1	(10, 15)

v: Profit

$x_1 = 1$
0
1
0
0
1
0
0
1

$\boldsymbol{\phi}$: Shapley values

B^{c_1}	B^{c_2}	$\boldsymbol{P}_{x_1} = (p_{x_1}^{o_1}, p_{x_1}^{o_2})$
0.83333	0.33333	-0.16666

Figure 1: Calculation flow example of Shapley values.

Then, the vector $\boldsymbol{S}$ of player subsets S is generated as binary permutations of size D (*GenerateBinaryPermutations*). For each subset S, $Problem$ is updated as $Setup$ by adopting the value of players in D (*GenerateSetup*). When the *i-th* element of S indicates the presence of player i ($s_i = 1$), the original value of player i in $Problem$ is utilized, and should $s_i = 0$, the background data R of player i is applied to simulate the absence. If player i represents the group of B^c or p, all values in the player are changed simultaneously.

Next, under the generated $Setup$, the optimal $Solution$ is derived by model f. The profit v for subset S is calculated in accordance with the user's question (*ComputeProfit*). If multiple $Solution$s under the same subset S is obtained, the average of profits is utilized as v. The pair of v and S is added to *perturbations*. After all profits for subset S are derived, we can calculate the Shapley value ϕ_i for every player i defined as (2) (*ComputeShapley*).

Figure 1 shows the example calculation flow of our framework. We assume that the user's question is "Why is the value of $x_1 = 1$?" Let us assume there are three kinds of input factor candidates: B^{c_1}, B^{c_2}, and $\boldsymbol{P}_{x_1}$. This means that the user is interested in the effects of whether the inclusion of c_1 or c_2 and the value of $\boldsymbol{P}_{x_1}$ on the result of x_1 in the optimal solution. The background data R is defined as the removal of the constraints ($B^{c_1} = 0$ and $B^{c_2} = 0$) and vector for $\boldsymbol{P}_{x_1}$ prepared by the user (0, 0). First, the permutations of 2^3 player subsets $\boldsymbol{S}$ is generated for input candidates. Next, for each subset, the new problem settings $Setup$ are generated by combining D and R, and $Solution$ is calculated via model f. After that, we obtain the profit $v = 1$ when $x_1 = 1$ in $Solution$, and vice versa. Finally, the Shapley value for each input factor can be obtained from *perturbations* (S, v).

The contributions mean the impact on the value of x_1. Note that the sum of contributions is equal to the difference between profit in the original $Solution$ and the background data. In Fig.1, B^{c_1} and B^{c_2} have positive values, and therefore the inclusion of these constraints keeps $x_1 = 1$. In contrast, $\boldsymbol{P}_{x_1}$has a negative value, which indicates that setting the parameters as (10, 15) prevents x_1 from being 1.

Analysis of Obtained Contributions

The contributions of several types of input factors can be quantitatively evaluated. Contributions $\boldsymbol{\phi}$ are expressed in the form of an additive decomposition over profit v. For a profit of continuous value, the contribution corresponds to a direct increase or decrease in that value. For conditional expression, it can be interpreted as the probability of condition fulfillment. The extraction of high impact c and p on the derivation of optimal plan will be helpful for efficient understanding of the problem setting. For example, when c_1 has a large positive contribution, we can answer the users question as "Because there is constraint c_1." This achieves simple and uniform comparison of input factors under complicated considerations in optimization models.

Quantification also helps users finding a desired plan efficiently. If users want to change the specific value in $Solution$, removing a factor that has a significant positive contribution to the fulfillment of the variable will make it possible to obtain the desired solution.

Note that the computing cost of the Shapley value is $O(2^d)$: it requires a high calculation cost depending on the number of input candidates. Therefore, it is helpful to reduce the number of input candidates in advance on the basis of the user's interest or prior knowledge. Grouping the candidates into one player is also effective and can make interpretation of the contribution easy in some cases. Furthermore, we can apply existing efficient calculation approaches like Monte Carlo sampling (Song, Nelson, and Staum 2016). In addition, Cohort Shapley (Mase, Owen, and Seiler 2019) and Causal Shapley values (Heskes et al. 2020) may be useful to reflect dependencies among constraints or parameters.

This explanation is applicable to a variety of continuous value/conditional questions and is independent of the type of optimization model. From next section, we show a typical combinatorial optimization problem, the personnel assignment, as an example of formulation and explanation process.

Numerical Experiment of Allocation Problem

Experimental Setting

The assignment problem determines which elements of set A should be assigned to the elements of set B. In this paper, we focus on the typical problem of assigning personnel to appropriate jobs. We aim to verify that the proposed method can generate different explanations for two parameter settings. Let $W = \langle w_1, w_2, \dots, w_6 \rangle$ be a set of six employees and $J = \langle j_1, j_2, j_3 \rangle$ be a set of three jobs. We define the following optimization problem to maximize the skill match as mixed integer linear programming (MILP):

$$maximize\ \boldsymbol{O}\colon \sum_{l=1}^{3}\sum_{n=1}^{6}\sum_{m=1}^{3} p_{w_n}^{o_l} \cdot \left(p_{j_m}^{o_l}\right)^T \cdot x_{w_n}^{j_m} \tag{4}$$

$$s.t. \quad c_1\colon\ x_{w_n}^{j_1} + x_{w_n}^{j_2} + x_{w_n}^{j_3} = 1\ \ for\ (w_1, w_2, \dots, w_6) \tag{5}$$

$$c_2\colon\ x_{w_1}^{j_m} + x_{w_2}^{j_m} + \cdots + x_{w_6}^{j_m} \leq 2\ \ for\ (j_1, j_2, j_3) \tag{6}$$

$$c_3\colon \begin{cases} x_{w_1}^{j_1} + x_{w_2}^{j_1} = 1 \\ x_{w_3}^{j_2} + x_{w_4}^{j_2} = 1 \\ x_{w_5}^{j_3} + x_{w_6}^{j_3} = 1. \end{cases} \tag{7}$$

The following vector of binary decision variables $\boldsymbol{x}$ represents the assignment of each employee:

$$\boldsymbol{x}_{w_n} = \left(x_{w_n}^{j_1}, x_{w_n}^{j_2}, x_{w_n}^{j_3}\right)\ \ for\ (w_1, w_2, \dots, w_6). \tag{8}$$

Employee w is assigned to only one job j where $x_{w_n}^{j_m} = 1$ as the constraint c_1. the capacity of each job is only two employees as the constraint c_2. Each job also has a specified assignment constraint c_3. As the objective function $\boldsymbol{O}$, we assume there are three skills (o_1, o_2, o_3) and aim to maximize the sum of products of skill sets for each employee $\boldsymbol{P}_W = \left(p_W^{o_1}, p_W^{o_2}, p_W^{o_3}\right)$ and job $\boldsymbol{P}_J = \left(p_J^{o_1}, p_J^{o_2}, p_J^{o_3}\right)$. Table 1 shows the required skills of each job, and Table 2 shows the skill values in the two parameter settings.

In this experiment, the optimization problem was solved using the Python library, *pulp* (Mitchell, O'Sullivan, and Dunning 2011). Figures 2 and 3 show the optimal placement plan in each problem setting, respectively. The black squares indicate that corresponding employees assigned to their jobs ($x_w^j = 1$) and the values of objective function (4) for each employee and job. Note that each placement is common in the two settings.

	o_1	o_2	o_3
j_1	3	1	1
j_2	1	3	1
j_3	1	1	3

Table 1: Required skills of each job.

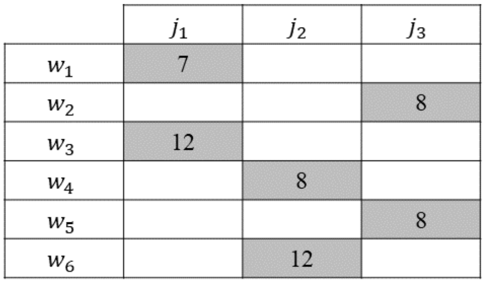
	o_1	o_2	o_3
w_1	1	3	1
w_2	1	1	2
w_3	3	1	2
w_4	1	2	1
w_5	1	1	2
w_6	2	3	1

(a) Problem 1

	o_1	o_2	o_3
w_1	2	4	1
w_2	2	1	3
w_3	1	3	2
w_4	0	3	1
w_5	2	1	2
w_6	0	4	1

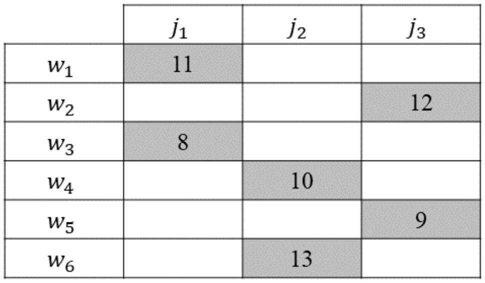
(b) Problem 2

Table 2: Skill sets of each employee.



	j_1	j_2	j_3
w_1	7		
w_2			8
w_3	12		
w_4		8	
w_5			8
w_6		12	

Figure 2: Derived optimal plan in Problem 1.



	j_1	j_2	j_3
w_1	11		
w_2			12
w_3	8		
w_4		10	
w_5			9
w_6		13	

Figure 3: Derived optimal plan in Problem 2.

Here, we focus on employee w_1 as an explanation target. Although this employee has a high o_2 skill values and is suitable for job j_2, the optimal solutions under both settings placed w_1 at j_1. Thus, we try to answer the question of the conditional expression, “Why was w_1 placed in j_1?” As input factor candidates D, we selected a total of 12 factors: three binary variables B_j^{c2} for (j_1, j_2, j_3) on the maximum number of employees, three specified assignment constraints B_j^{c3} for (j_1, j_2, j_3), and six grouped skill sets for each employee $\boldsymbol{P}_w = (p_w^{o_1}, p_w^{o_2}, p_w^{o_3})$ for $(w_1, w_2, \dots, w_6)$. For the background data, each constraint condition is set to OFF $(B_j^c = 0)$, and the average values of each employee skill sets are adopted for $\boldsymbol{P}_w$ as shown in Table 3. Profit $v(S)$ is represented by the value of $x_{w_1}^{j_1}$ in derived $Solution$.

In addition, a basic contrastive explanation is presented as a conventional method to evaluate the efficiency of extracting major causes. A suboptimal plan with w_1 in j_1 is generated by adding the constraint $x_{w_1}^{j_1} = 1$ into *pulp* model.

Experimental Result

First, we discuss the results of the obtained contributions. Table 4 shows the contributions of input factors calculated as the Shapley value for each problem setting. In Problem 1, the top three positive contributions are $B_{j_1}^{c3}$, P_{w_2}, and $B_{j_2}^{c2}$: the specified assignment constraint c_3 for j_1, the skills of w_2, and the capacity constraint c_2 of j_2. This means that these factors strongly force w_1 to be placed in j_1. In contrast, the contribution of w_1 skills is negative: this value indicates that the skill is not suitable for j_1, and w_1 would have preferred to move to a different job from the viewpoint of the skill set. However, the sum of positive contributions exceeded w_1 skills, and therefore, w_1 had to remain in j_1.

Furthermore, we can interpret the contributions with the meaning of these factors in the problem setting as the answer to the question “Why was w_1 placed in j_1?”.

- The skills of w_1 were suited for job j_2; however, there was no slot available for w_1 because of the capacity constraint c_2 in j_2.
- The constraint c_3 for j_1 required that w_1 or w_2 must be in j_1. Compared with the contributions of w_2 skills, it is better to assign w_1 rather than w_2 to j_1.

Although there are multiple types of input factors, we can extract the major factors by the contributions. The further analysis is possible by asking additional questions and calculating the contribution, e.g., “Why was w_4 and w_6 assigned to j_2?” Users can reveal the contributing factors on the basis of the quantitative indicators, not human intuition.

Table 4 (b) shows the contributions in Problem 2. There are three factors with high contributions, $B_{j_1}^{c3}$, $B_{j_2}^{c2}$, and P_{w_6}: the specified assignment constraint c_3 for j_1, the capacity constraint c_2 of j_2, and the skills of w_6. In contrast to Problem 1, the contribution of w_2 skills is relatively low. The contribution of w_1 skills is also negative; however, the value is too low. Then, we can interpret the contributions as follows:

- There was no slot available for w_1 in j_2 because of the capacity constraint c_2 in j_2.
- Since the skill of w_6 has a high contribution and w_6 is suitable for job j_2, there were conflicts between w_1 and w_6, and the plan that assigned w_6 in j_2 rather than w_1 was appropriate from the viewpoint of their skill sets.
- The constraint c_3 for j_1 required that w_1 should be in j_1. Compared to the contributions of w_2 skills, it is better to assign w_1, however, this effect is lower than Problem 1.

In accordance with the Shapley value, we can extract the important factors under various parameters settings.

	o_1	o_2	o_3
w	1.5	1.8333	1.5

(a) Problem 1

	o_1	o_2	o_3
w	1.1667	2.6667	1.6667

(b) Problem 2

Table 3: Background data of employee’s skill sets.

	Shapley value
$B_{j_1}^{c_2}$	-0.046781
$B_{j_2}^{c_2}$	**0.278331**
$B_{j_3}^{c_2}$	-0.019892
$B_{j_1}^{c_3}$	**0.438896**
$B_{j_2}^{c_3}$	0.096419
$B_{j_3}^{c_3}$	0.042261
$\boldsymbol{P}_{w_1}$	**-0.272843**
$\boldsymbol{P}_{w_2}$	**0.320272**
$\boldsymbol{P}_{w_3}$	0.026447
$\boldsymbol{P}_{w_4}$	0.040454
$\boldsymbol{P}_{w_5}$	-0.00809
$\boldsymbol{P}_{w_6}$	0.104524

(a) Problem 1

	Shapley value
$B_{j_1}^{c_2}$	-0.028099
$B_{j_2}^{c_2}$	**0.318639**
$B_{j_3}^{c_2}$	-0.002180
$B_{j_1}^{c_3}$	**0.289337**
$B_{j_2}^{c_3}$	0.039251
$B_{j_3}^{c_3}$	-0.014112
$\boldsymbol{P}_{w_1}$	**-0.079754**
$\boldsymbol{P}_{w_2}$	0.121222
$\boldsymbol{P}_{w_3}$	-0.005629
$\boldsymbol{P}_{w_4}$	0.051132
$\boldsymbol{P}_{w_5}$	-0.003588
$\boldsymbol{P}_{w_6}$	**0.306605**

(b) Problem 2

Table 4: Shapley values for each input factor.

A contribution analysis is also useful to change the problem setting to move w_1 from j_1. Figure 4 shows the optimization result under Problem 1 without constraint c_3 for j_1. We could confirm that w_1 is assigned to j_2 by removing the constraint that has the strongest contribution to keep w_1 in j_1. If all contributions are low and the desired result cannot be obtained by moving any factors, it suggests that none of the candidate parameters influenced the assignment in which users are interested.

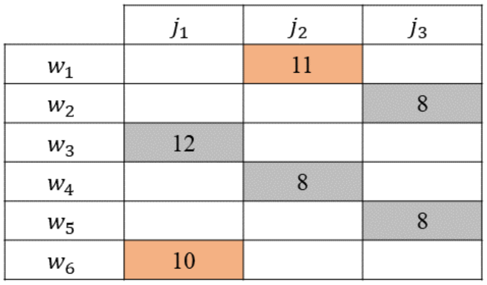
In our experiments, it took about 3 minutes on average to generate the explanation on Intel Core i7-1065G7 CPU @1.30GHz with 16GB RAM. Despite the numerous combinations of factors (2^{12}), the calculation time of the optimization task was minimal, and therefore, the exact Shapley values could be computed quickly.

To compare the derived explanations, Figure 5 shows a contrastive suboptimal solution with the constraint that w_1 must be assigned to j_1 in Problem 1. Blue squares indicate that the values of objective function (4) are lower than those of the optimal solution, and orange square shows higher value. From Fig. 5, although the match of the w_1 assignment is better, the values of w_2, w_3, and w_6 became worse. Then, it is not possible to tell which of the values of w_2, w_3, and w_6 had a strong effect on the w_1 assignment only by the comparison of derived plans. Fig. 6 shows the optimal solution with the background data of w_6 skills, however, w_1 could not be transferred to j_1. Compared with Table 4 (a), the contribution of w_6 skills is a positive low value, and therefore only removing the effect of w_6 skills is insufficient for w_1. In addition, the explanation of constraints is not shown in the suboptimal solution. Thus, it is difficult to evaluate the impact of multiple factor types on w_1. In the proposed method, the Shapley value distributes the contributions appropriately, and we can discuss the impact of constraints and parameters under a unified framework.

Conclusion and Future Work

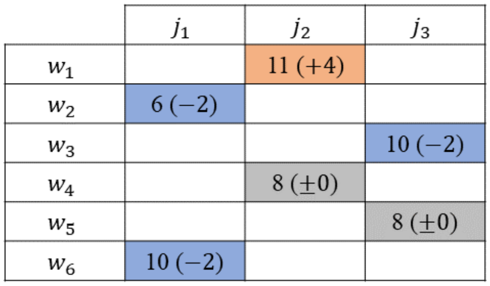
In this paper, to derive reasons for counterintuitive cases in optimization problems, we proposed a general explanation framework that quantitatively evaluates the contribution of constraints and parameters to the plan derivation. The wide range of question patterns and optimization models can be applied to our framework. We formulated a coalitional game based on the optimization model to calculate the Shapley value. To show an example of explanation process, we experimented with typical personnel assignment problem. The results show that our framework could extract highly contributing factors under various parameter settings, and provide directly understandable explanations compared with the traditional contrastive explanation approach.

In the future, we should collect objective data such as user opinions on the usefulness of these explanations. There is a gap between the obtained contributions and easy interpretation for users. We should develop an explanation method combined with domain knowledge such as a causal model. From the viewpoint of feasibility, the bottleneck is the computational cost of the Shapley value. In the experiment, the number of permutations of factor candidates was 2^{12}. However, in large-scale problems, an approximation of the calculation process is required. It is important to assess whether a reasonable contribution can be obtained under approximated conditions. Furthermore, by applying this framework to various real-world problems, we contribute to expanding the application range of planning optimization and building trust between humans and systems.



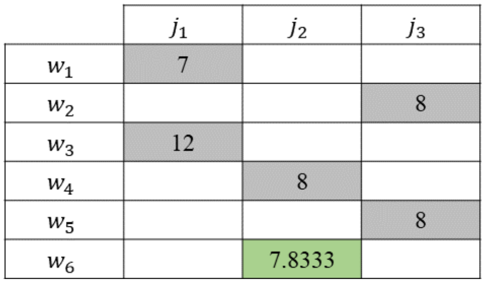
	j_1	j_2	j_3
w_1		11	
w_2			8
w_3	12		
w_4		8	
w_5			8
w_6	10		

Figure 4: Derived plan constraint c_3 for j_1.



	j_1	j_2	j_3
w_1		11 (+4)	
w_2	6 (−2)		
w_3			10 (−2)
w_4		8 (±0)	
w_5			8 (±0)
w_6	10 (−2)		

Figure 5: Suboptimal plan in Problem 1.



	j_1	j_2	j_3
w_1	7		
w_2			8
w_3	12		
w_4		8	
w_5			8
w_6		7.8333	

Figure 6: Derived plan with background data of w_6.

References

Agrawal, J.; Yelamanchili, A.; and Chien S. 2020. Using Explainable Scheduling for the Mars 2020 Rover Mission. Paper presented at the International Workshop of Explainable AI Planning. Virtual, October 19–30.

Benoumechiara, N., and Cosaque, E. D. K. 2018. Shapley effects for sensitivity analysis with dependent inputs: bootstrap and kriging-based algorithms. *ESAIM: Proceedings and Surveys* 65: 266–293. doi.org/10.1051/proc/201965266.

Brandão, M.; Coles, A.; and Magazzeni, D. 2021. Explaining Path Plan Optimality: Fast Explanation Methods for Navigation Meshes Using Full and Incremental Inverse Optimization. In Proceedings of the International Conference on Automated Planning and Scheduling, 31(1), 56–64. California: Association for the Advancement of Artificial Intelligence. doi.org/10.1609/icaps.v31i1.15947.

Burt, C.; Klimova K.; and Primas, B. 2018. Generating Explanations for Mathematical Optimisation: Solution Framework and Case Study. Paper presented at the International Workshop of Explainable AI Planning. Delft, The Netherlands June 24–29.

Cashmore, M.; Collins, A.; Krarup, B.; Krivic, S.; Magazzeni, D.; and Smith, D. 2019. Towards Explainable AI Planning as a Service. Paper presented at the International Workshop of Explainable AI Planning. Berkeley, California USA July 11–15.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2020. The Emerging Landscape of Explainable Automated Planning & Decision Making. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. Survey track, 4803–4811. Yokohama:International Joint Conferences on Artificial Intelligence. doi.org/10.24963/ijcai.2020/669.

Chen, H.; Covert, Ian C.; Lundberg Scott M.; and Su-In Lee. 2022. Algorithms to estimate Shapley value feature attributions. arXiv preprint. arXiv:2207.07605v1 [cs.LG]. Ithaca, NY: Cornell University Library.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable Planning. arXiv preprint. arXiv:1709.10256v1 [cs.AI]. Ithaca, NY: Cornell University Library.

Fryer, D.; Strumke, I.; and Nguyen, H. 2021. Shapley values for feature selection: The good, the bad, and the axioms. arXiv preprint. arXiv:2102.10936[cs.LG]. Ithaca, NY: Cornell University Library.

Gragera, A.; García, O. A.; and Fernández, F. 2022. Repair Suggestions for Planning Domains with Missing Actions Effects. Paper presented at the International Workshop of Explainable AI Planning. Virtual, June 13–24.

Gunning, D., and Aha, D. 2019. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, 40(2): 44–58. https://doi.org/10.1609/aimag.v40i2.2850.

Gupta, S. D.; Genc, B.; and O'Sullivan, B. 2022. Finding Counterfactual Explanations through Constraint Relaxations. arXiv preprint. arXiv:2204.03429 [cs.AI]. Ithaca, NY: Cornell University Library.

Hall, N. G., and Posner, M. E. 2004. Sensitivity Analysis for Scheduling Problems. *Journal of Scheduling* 7: 49–83. doi.org/10.1023/B:JOSH.0000013055.31639.f6.

Hedayat, A. S.; Sloane, N. J. A.; and Stufken, J. 1999. *Orthogonal Arrays: Theory and Applications*. Springer New York.

Heskes, T.; Bucur, G. I.; Sijben, E.; and Claassen, T. 2020. Causal shapley values: exploiting causal knowledge to explain individual predictions of complex models. In Proceedings of the 34th International Conference on Neural Information Processing Systems: 4778–4789. New York: Curran Associates Inc.

Iturralde, M.; Yahiya, Ali, T.; Wei A.; and Beylot, A.-L. 2011. Resource allocation using Shapley value in LTE networks. In Proceedings of the 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, 31–35. Tronto: Institute of Electrical and Electronics Engineering. doi: 10.1109/PIMRC.2011.6139974.

Korikov, A.; Shleyfman, A.; and Beck J. 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, Main Track, 4097–4103. Montreal: International Joint Conferences on Artificial Intelligence. doi.org/10.24963/ijcai.2021/564.

Legrain, A.; Bouarab, H.; and Lahrichi, N. 2015. The Nurse Scheduling Problem in Real-life. *Journal of medical systems*, 39(1): 1–11. doi.org/10.1007/s10916-014-0160-8.

Lundberg, Scott M., and Lee, Su-In. 2017. A Unified Approach to Interpreting Model Predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 4768–4777. New York: Curran Associates Inc. doi.org/10.5555/3295222.3295230.

Mase, M.; Owen, A. B.; and Seiler, B. 2019. Explaining black box decisions by Shapley cohort refinement. arXiv preprint. arXiv: 1911.00467 [cs.LG]. Ithaca, NY: Cornell University Library.

Mitchell, S., O'Sullivan, M.J., & Dunning, I. 2011. PuLP: A Linear Programming Toolkit for Python. Department of Engineering Science. The University of Auckland. 65.

Pozanco, A.; Mosca, F.;Zehtabi, P.;Magazzeni, D.; and Kraus, S. 2022. Explaining Preference-Driven Schedules: The EXPRES Framework. In Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling. Human-Aware Planning and Scheduling Track, 710–718. California: Association for the Advancement of Artificial Intelligence. doi.org/10.1609/icaps.v32i1.19861.

Shapley, L. 1953. A Value for n-Person Games. *Contributions to the Theory of Games II*: 307–317. doi.org/10.1515/9781400881970-018.

Song, E.; Nelson, Barry L.; and Staum J. 2016. Shapley Effects for Global Sensitivity Analysis: Theory and Computation. SIAM/ASA Journal on Uncertainty Quantification 4(1): 1060–1083. doi.org/10.1137/15M1048070.

Soni, U.; Sreedharan, S.; and Kambhampati, S. 2021. Not all users are the same: Providing personalized explanations for sequential decision making problems. arXiv preprint. arXiv:2106.12207 [cs.AI]. Ithaca, NY: Cornell University Library.

Zahedi, Z.; Sengupta, S.; and Kambhampati S. 2020. 'Why didn't you allocate this task to them?' Negotiation-Aware Task Allocation and Contrastive Explanation Generation. arXiv preprint. arXiv:2002.01640v3 [cs.AI]. Ithaca, NY: Cornell University Library.