Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Mass Customization and "Forecasting Options' Penetration Rates Problem"

Ali Fattahi
Anderson School of Management, University of California, Los Angeles, Los Angeles, California 90095, ali.fattahi.1@anderson.ucla.edu

Sriram Dasu
Marshall School of Business, University of Southern California, Los Angeles, Los Angeles, California 90089, dasu@marshall.usc.edu

Reza Ahmadi
Anderson School of Management, University of California, Los Angeles, Los Angeles, California 90095, rahmadi@anderson.ucla.edu

Auto manufacturers produce a very large number of feasible configurations that makes it impossible to forecast the demand of individual configurations. What the companies do forecast is the penetration rates of options that is also called the penetration statistic (PS). The penetration rate of an option is the percentage of cars that include that option. The current forecasting approach ignores rules for selecting options and, as a result, PSs are frequently infeasible, which results in excess inventories, shortages, and customer dissatisfaction. The problem of determining the feasibility of the forecast PS and finding the best feasible PS in the case of infeasibility is NP-complete. This paper proposes an approach that involves presenting rules in a new format and solving a mathematical program. The feasible region for a PS is a polyhedron inscribed inside a unit cube and a one-to-one correspondence exists between feasible configurations and the extreme points of the polyhedron. We present an approach that sequentially constructs the feasible region and stops when it finds the best feasible PS. We analyse the theoretical properties of our approach and provide insights on its convergence rate. We also show the effectiveness of our approach on a set of real instances.

Subject classifications: Optimization. Mass customization. Forecasting.

1. Introduction

This paper is based on the problems faced by large auto manufacturers (LAMs) that allow customers to configure their cars on the Internet. Automobiles consist of a number of modules such as engines, interiors, and suspensions and each module has a number of different variants or options. A customer configures his or her product by choosing options that are compatible. The number of valid configurations can be extremely large. For example, the number of valid configurations for the Mercedes C-Class is in the order of 10^{21} (Kulber et al. 2010). Manufacturers need demand forecasts at the level of configurations for production planning, supplier contracts, and pricing decisions. Configurations drive the bill of materials. Unfortunately, the number of potential configurations and the difficulty in enumerating feasible configurations make it impossible to forecast configuration demand.

Most firms forecast at the option level. It is common practice to present this forecast in terms of a penetration statistic (PS). A PS consists of an assigned penetration rate to each option over the planning horizon that is simply the fraction of cars that they believe will have that option. In other words, if the penetration rate of an engine is 0.2, it is forecast that 20% of cars sold will have that engine. PSs are used to plan inventories and contracts with suppliers. Errors in PSs can be extremely costly since inaccurate PS forecasts can result in excess inventories, shortages, and customer dissatisfaction.

Valid configurations conform to several restrictions between options—which are called rules—and represent marketing, manufacturing, process, and engineering constraints that allow products to be producible. The total demand of configurations and hence the forecast PS must satisfy these rules. A major drawback of forecasting PSs is the difficulty in ensuring their consistency with the rules. As a consequence, forecast PSs are frequently infeasible. LAMs face the major challenge of determining the feasibility of the forecast PS and modifying it in the case of infeasibility.

Rules can be divided into two general categories: *Family Cardinality Rules* (FCRs), which require the selection of exactly/at most one option from a family of options, and *Option Implication Rules*

(OIRs), which consist of logical relationships among options from different families. The following example illustrates these terms.

EXAMPLE 1. (Fig. 1) Features and specifications of the 2016 Hyundai Tucson[1] are grouped into ten families such as mechanical, exterior, and interior features. The four trim packages of this car are the SE, the Eco, the Sport, and the Limited. Fig. 1 compares the engine, transmission, and wheels of these trims. The SE's engine and transmission are `ENG1` and `TRN1`, respectively, while the Eco, Sport and Limited's engine and transmission are `ENG2` and `TRN2`, respectively. The wheels of the SE and Eco are `WHL1`, and the wheels of the Sport and Limited are `WHL2`. Note that, in the Sport and Limited, the engine, transmission, and wheels are identical.

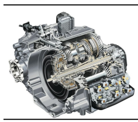
Assume for the sake of illustration that the 2016 Hyundai Tucson has only six options: `ENG1, ENG2, TRN1, TRN2, WHL1`, and `WHL2`. As shown in Fig. 1, there exist three FCRs, and two OIRs; exactly one option must be chosen from each family. According to the rule $\texttt{ENG1}\Longleftrightarrow\texttt{TRN1}$, `ENG1` is selected if and only if `TRN1` is selected. Also, $\texttt{ENG1}\Longrightarrow\texttt{WHL1}$ means that if `ENG1` is chosen, then `WHL1` must be chosen as well. Note that engine FCR is also equivalent to $\texttt{ENG1}\Longleftrightarrow\neg\ \texttt{ENG2}$. The number of OIRs may vary based on the format—one could write $\texttt{ENG1}\Longleftrightarrow\texttt{TRN1}$ or as two OIRs $\texttt{ENG1}\Longleftarrow\texttt{TRN1}$ and $\texttt{ENG1}\Longrightarrow\texttt{TRN1}$.

Configurations that satisfy the FCRs and OIRs are: `SE={ENG1, TRN1, WHL1}`, `ECO={ENG2, TRN2, WHL1}`, and `SPORT/LIMITED={ENG2, TRN2, WHL2}`. The number of candidates for configurations is 2^6, while only three feasible configurations exist.

Suppose the rates of `ENG1, ENG2, TRN1, TRN2, WHL1`, and `WHL2` are forecast as 0.6, 0.4, 0.6, 0.4, 0.3, and 0.7, respectively, meaning that 60% of 2016 Hyundai Tucson sales will have `ENG1`, 40% will have `ENG2`, and so on. These rates satisfy the FCRs—the rates of `ENG1` and `ENG2` add up to 1—but they are infeasible as they do not satisfy the OIRs. Due to the rule $\texttt{ENG1}\Longrightarrow\texttt{WHL1}$, any configuration that includes `ENG1` must also include `WHL1`; however, a configuration might have `WHL1`, but not `ENG1`. Thus, the rate of `WHL1` must be greater than or equal to the rate of `ENG1`; hence, the rate of `WHL1` must be at least 0.6, which also implies an upper bound of 0.4 on the rate of `WHL2`.

2016 HYUNDAI TUCSON

FEATURES & SPECIFICATIONS

		SE	ECO	SPORT/ LIMITED
	ENGINE	ENG1	ENG2	ENG2
	TRANSMISSION	TRN1	TRN2	TRN2
	WHEELS	WHL1	WHL1	WHL2

NOTATIONS:

ENG1: Inline 4-cylinder
ENG2: Inline 4-cylinder turbocharged
TRN1: 6-speed automatic transmission with SHIFTRONIC and Active ECO System
TRN2: 7-speed EcoShift Dual Clutch Transmission
WHL1: 17-inch alloy wheels with 225/60HR17 tires
WHL2: 19-inch Sport alloy wheels with 245/45HR19 tires

FAMILIES: (exactly one option from each family)

Engine Family: {ENG1,ENG2}
Transmission Family: {TRN1,TRN2}
Wheels Family: {WHL1,WHL2}

OPTION IMPLICATION RULES:

ENG1 $\Longleftrightarrow$ TRN1
ENG1 $\Longrightarrow$ WHL1

Figure 1 A simplified version of the features and specifications of the 2016 Hyundai Tucson.

Consider Table 1, and assume that Hyundai has forecast a total of 1000 Tucsons to be sold in 2016 and procures engines, transmissions, and wheels accordingly (row 3 of Table 1). Although Hyundai has 600 `ENG1` in inventory, they can use at most 300 `ENG1` since there are only 300 `WHL1` available. Unused quantities are found by subtracting row 4 from row 3, and shortages are calculated by noting that they can sell at most 300 SE and 400 SPORT/LIMITED. Therefore, Hyundai incurs significant inventory costs due to unused inventory, and lost profits as a result of lost sales.

Table 1 Unused inventory and lost sales as the result of infeasible forecasting (total sale = 1000)

	ENG1	ENG2	TRN1	TRN2	WHL1	WHL2
Forecast PS	0.6	0.4	0.6	0.4	0.3	0.7
Inventory	600	400	600	400	300	700
Maximum usable quantities	300	400	300	400	300	400
Unused quantities	300	0	300	0	0	300
Shortage (lost sale)	$1000-(300+400)=300$					

Checking if a PS is feasible and finding the best feasible PS in the case of infeasibility is very important not only in the auto industry, but also in any company producing configurable products. The LAM that we have been in contact with has around 400 options and 4000 rules for each car and the company's production planning group currently employs a manual process to detect some of the rule violations. Once a set of violations is detected they send an error report to the regional marketing analysts who then seek appropriate modifications to the PS. These cycles are repeated until all evident violations are eliminated. However, this approach does not ensure that the PS is feasible. As a result, the planning group next employs a column generation method to verify feasibility; often, the PS is indeed infeasible. Once infeasibility is detected the current method fails to generate a feasible PS that is "close" to the forecast. The current approach is time-consuming and fails to generate good alternatives when the forecast is infeasible.

In this paper, we propose an approach that involves translating the rules to a new format and formulating the problem as a quadratic program. The feasible region of our problem is a polyhedron inscribed inside a unit cube in $\mathbb{R}^n$, where n is the number of options. We show that a one-to-one correspondence exists between feasible configurations and the extreme points of this polyhedron. Because the problem has an exponentially large number of variables and constraints, we develop an approach that sequentially constructs the feasible region and stopping when it either finds a set of feasible configurations that are consistent with the given PS or when it determines the infeasibility of the given PS and finds the closest feasible PS and the corresponding configuration set. Although the problem is NP-complete, we show that under some mild assumptions, our approach converges at a rate proportional to $\frac{1}{\sqrt{k}}$, where k is the iteration counter. We establish the tightness of this bound through an example. We test our methodology on a set of real instances and obtain very good solutions in a short time.

The remainder of the paper is organized as follows. Section 2 presents related literature. Section 3 introduces the alternative format for presenting rules and our model. In section 4, we present our algorithm and discuss its theoretical properties. In section 5, we test the effectiveness of our algorithm and show its sensitivity to the number of options and rules. We then conclude and offer directions for future work.

2. Literature

In addition to the automotive industry, a number of others, including consumer electronics, computers, furniture, and aircraft (Feitzinger and Lee 1997, Fohn et al. 1995, Kristianto et al. 2015, Rodriguez and Aydin 2011) allow customers to configure products by selecting among options. While the use of modular design techniques and option-based product architecture (Ulrich 1994, Siddique et al. 1998, Sanchez and Mahoney 1996) increases the variety that can be made available, it also presents a number of challenges for customers (Franke and Piller 2004, Huffman and Kahn 1998, Chen and Wang 2010, Walker and Bright 2013) and producers (Woehler 2011, Ostrosi et al. 2012).

Researchers have studied ways to price different options (Rodriguez and Aydin 2011), and present assortments to consumers to learn about demand (Caro and Jeremie 2007, Balseiro et al. 2014, Kok et al. 2008). In the assortment planning literature, finished goods are presented to consumers and their choices are used to infer the relative attractiveness of different variants. Much of this literature is concerned with digital goods such as Internet advertisements where inventory of goods is not an issue. Assortment planning models have also been developed for retailers, but the focus is on the set of products to be presented to the customer. Our work is concerned with developing demand forecasts for configurable products. In this setting, parts needed for the configurable products have to be produced in advance or at least the production facilities and supply contracts have to be set up well ahead of demand realization. As previously stated, these decisions are based on PSs and not forecasts for individual configurations. We are concerned with determining whether the forecast PS is consistent with the rules that bind options. We develop a mathematical programming-based model, present an algorithm, and provide insights into its convergence rate.

Our approach requires us to represent rules in a manner that is amenable to mathematical programming. Work by Barker et al. (1989), Roller and Kreuz (2003) and Amilhastre et al. (2002) is representative of the studies on representing rules related to configurable products. This literature is concerned with facilitating activities such as product design, engineering of production lines, and managing knowledge while we are concerned with exploring whether a forecast PS is consistent with the given rules. The current format of the rules used by the LAM in fact has been set up to facilitate activities such as design and manufacturing. Consequently, we had to develop an alternative representation.

At the core of our problem is a variant of the well-studied probabilistic satisfiability problem (PSAT). PSAT is known to be NP-complete (Nilsson 1994, Finger and De Bona 2011, Georgakopoulos et al. 1988) and is typically solved using a column generation technique (Kavvadias and Papadimitriou 1990, Hansen et al. 1995). There are at least two limitations to column generation techniques. One limitation is that these methods typically converge slowly and more importantly if they fail to find a feasible solution, we are then unable to determine the nearest feasible solution. Our approach always finds the nearest feasible solution. If the PS is feasible then it is the closest and our algorithm verifies feasibility. We are also able to provide insights into the convergence rate of our algorithm. To the best of our knowledge, no convergence bounds are available for column generation techniques. The approach we propose is similar to the Frank-Wolfe method that was originally developed to maximize a smooth concave function over a polytope (Frank and Wolfe 1956) and then extended to more general problems (see for example Demyanov and Rubinov (1970)). Freund and Grigas (2014) discuss computational guarantees for various step-size rules within this algorithm.

3. Problem Formulation

Given a set of options with associated PS and a set of rules, we have to determine whether the PS satisfies option related rules and if not, find the best feasible PS. Let $N = \{i | i = 1, \cdots, n\}$ denote the set of options, and $p(E) \in [0,1]$ be the probability of event E (for the complete set of notations, see Tables A1 and A2). We will also use $p(E)$ to denote the percentage of configurations ordered by customers in which event E occurs—e.g., $p(i \wedge j)$ is the fraction of configurations in which customers select both options i and j. For ease of notation we let $p_i = p(i)$. The vector $\mathbf{p} = (p_1, \cdots, p_n) \in \mathbb{R}^n$ is the PS. Let $\mathbb{P}$ be the set of all PSs that satisfy rules between options; hence, if $\mathbf{p} \in \mathbb{P}$, then $\mathbf{p}$ is feasible.

We use the fact that if it is possible to produce $\mathbf{p}$, then it is also possible to produce $\lambda\mathbf{p}$ for $\lambda \geq 0$. Thus, the cone generated by $\mathbb{P}$—i.e., $cone(\mathbb{P}) = \{\lambda\mathbf{p} | \mathbf{p} \in \mathbb{P}, \lambda \geq 0\}$—contains the set of feasible PSs.

Let $\mathbf{q}_1, \cdots, \mathbf{q}_M \in \mathbb{R}^n$ denote the real sales data. We define function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ as $f(\mathbf{p}) = \sum_{m=1}^{M} (\mathbf{p} - \mathbf{q}_m)^T (\mathbf{p} - \mathbf{q}_m)$ which returns sum of squared errors for a given $\mathbf{p}$. The LAM's forecast $\hat{\mathbf{p}}$ is obtained by minimizing f over all $\mathbf{p} \in \mathbb{R}^n$ without requiring $\mathbf{p} \in cone(\mathbb{P})$. Let $\|.\|$ denote the Euclidean norm. The following lemma shows that finding the nearest point (Euclidean distance) to $\hat{\mathbf{p}}$ in the set $cone(\mathbb{P})$ solves the original problem $\min_{\mathbf{p} \in cone(\mathbb{P})} f(\mathbf{p})$.

LEMMA 1. *Problem* $\min_{\mathbf{p} \in cone(\mathbb{P})} \|\mathbf{p} - \hat{\mathbf{p}}\|$ *is equivalent to* $\min_{\mathbf{p} \in cone(\mathbb{P})} f(\mathbf{p})$.

Therefore, if $\hat{\mathbf{p}}$ is infeasible, we have to find a feasible PS such that it has the minimum Euclidean distance to $\hat{\mathbf{p}}$. The result validates the LAM's desire to find the closest feasible PS if the forecast $\hat{\mathbf{p}}$ is infeasible.

Our problem can be formulated as $\mathscr{P} : \min_{\mathbf{p} \in \mathbb{P}, \lambda \geq 0} \|\lambda\mathbf{p} - \hat{\mathbf{p}}\|$. Let λ^* and $\mathbf{p}^*$ denote the optimal solution of this problem; if $\|\lambda^*\mathbf{p}^* - \hat{\mathbf{p}}\| = 0$, then $\hat{\mathbf{p}}$ is feasible—i.e., $\hat{\mathbf{p}} \in cone(\mathbb{P})$—and if $\|\lambda^*\mathbf{p}^* - \hat{\mathbf{p}}\| > 0$, then $\hat{\mathbf{p}}$ is infeasible and the best feasible PS, $\lambda^*\mathbf{p}^*$, has been found.

The objective function $\|\lambda\mathbf{p} - \hat{\mathbf{p}}\|$ is not convex; however, it is convex if λ is fixed. Problem $\mathscr{P}$ is equivalent to $\min_{\lambda \geq 0} \min_{\mathbf{p} \in \mathbb{P}} \|\lambda\mathbf{p} - \hat{\mathbf{p}}\|$. In the remainder, we propose a method to formulate $\mathbb{P}$ as a linear system and we show that $\mathbb{P}$ is a polyhedron. For fixed λ, the inner problem becomes a convex quadratic programming problem and line search methods such as bisection search and golden section search can be applied.

3.1. Rules for selecting options

In the LAM's current format, each OIR consists of a left-hand-side (LHS) option, a set of positive right-hand-side (RHS) options, and a set of negative RHS options. The RHS of an OIR is true if and only if all positive RHS options are true and all negative RHS options are false. There are 3 types of OIRs that we call `A`, `B`, and `R`. For a LHS option to be true, at least one of its associated RHSs, regardless of the rule type, must be true; hence, if all RHSs associated with a LHS option are false, then the LHS option must be false as well. If all RHSs from type `R` OIRs are true, then the LHS option must be true. An FCR can be of type `E` or `L` in which exactly one option must be chosen from each type `E` family and at most one option can be selected from a type `L` family. We illustrate these rules using example 2.

EXAMPLE 2. A hypothetical example is provided in Fig. 2 and includes 5 OIRs and 2 FCRs to represent the relationships among 17 options termed `OP01, OP02, ..., OP17`. In the first OIR, for instance, `OP01` is the LHS option, the rule type is `A`, the positive RHS options are `OP02` and `OP03`, and the negative RHS option is `OP04`. The RHS of the first OIR is true if and only if `OP02` and `OP03` are true and `OP04` is false. The first 3 OIRs have the same LHS option; hence, `OP01` is true if at least one of its associated RHSs is true. This logic can be written as: $\texttt{OP01} \implies (\texttt{OP02} \wedge \texttt{OP03} \wedge \neg \texttt{OP4}) \vee (\texttt{OP02} \wedge \texttt{OP03} \wedge \neg \texttt{OP05} \wedge \neg \texttt{OP06}) \vee (\texttt{OP02} \wedge \texttt{OP03} \wedge \neg \texttt{OP06} \wedge \neg \texttt{OP07})$.

Consider the OIRs 4 and 5, which are type `R` with the same LHS option; hence, if both RHSs in OIRs 4 and 5 are true, then option `OP08` must be true. This logic is written as $\texttt{OP08} \impliedby (\texttt{OP09} \wedge \neg \texttt{OP10} \wedge \neg \texttt{OP11}) \wedge (\texttt{OP09} \wedge \neg \texttt{OP10} \wedge \neg \texttt{OP12})$, which can be simplified as $\texttt{OP08} \impliedby \texttt{OP09} \wedge \neg \texttt{OP10} \wedge \neg \texttt{OP11} \wedge \neg \texttt{OP12}$.

The second column of Fig. 2 shows two families, termed `FM01` and `FM02`, with types `E` and `L`, respectively. Based on these FCRs, exactly one of the options `OP13,OP14`, or `OP15`, and at most one of `OP16` or `OP17` must be chosen.

```
OIRs (Option Implication Rules):

OP01, A {OP02, OP03} {OP04}
OP01, B {OP02, OP03} {OP05, OP06}
OP01, R {OP02, OP03} {OP06, OP07}
OP08, R {OP09} {OP10, OP11}
OP08, R {OP09} {OP10, OP12}

FCRs (Family Cardinality Rules):

FM01, E {OP13, OP14, OP15}
FM02, L {OP16, OP17}
```

Figure 2 The LAM's default format for representing option implication and family cardinality rules.

Although the current LAM format is easy for the technicians to use, it is not suitable for modeling and theoretical analysis; hence, we present the following format.

Format (ASR): *Each option, say option* `OP01`*, has rules either in the form of* `OP01`$\Longrightarrow$`RHS`*, where RHS is the disjunction of some options, or in the form of* `OP01`$\Longleftarrow$`RHS`*, where RHS is the conjunction of some options.*

In example 2, two rules are written for option `OP01` as follows: `OP01` $\Longrightarrow$ `(OP02`$\wedge$`OP03`$\wedge\neg$`OP04)`$\vee$`(OP02`$\wedge$`OP03`$\wedge\neg$`OP05`$\wedge\neg$`OP06)`$\vee$`(OP02`$\wedge$`OP03`$\wedge\neg$`OP06`$\wedge\neg$`OP07)` and `OP01` $\Longleftarrow$ `OP02` $\wedge$ `OP03` $\wedge$ $\neg$`OP06` $\wedge$ $\neg$`OP07`. The second rule is in Format (ASR). So, consider the first rule and note that its RHS is written as a disjunction of conjunctive clauses (DC). Factoring out the greatest common factor—i.e., `OP02`$\wedge$`OP03`—and using the distributive property of union over intersection, we obtain `OP01` $\Longrightarrow$ `OP02` $\wedge$ `OP03` $\wedge$ `(`$\neg$`OP04` $\vee$ $\neg$`OP05` $\vee$ $\neg$`OP06)` $\wedge$ `(`$\neg$`OP04` $\vee$ $\neg$`OP05` $\vee$ $\neg$`OP07)` $\wedge$ `(`$\neg$`OP04` $\vee$ $\neg$`OP06)` $\wedge$ `(`$\neg$`OP04` $\vee$ $\neg$`OP06` $\vee$ $\neg$`OP07)`. The RHS is a conjunction of disjunctive clauses (CD); hence, this rule is equivalent to rules 1-6 in Fig. 3. This reformatting is provable for the general case.

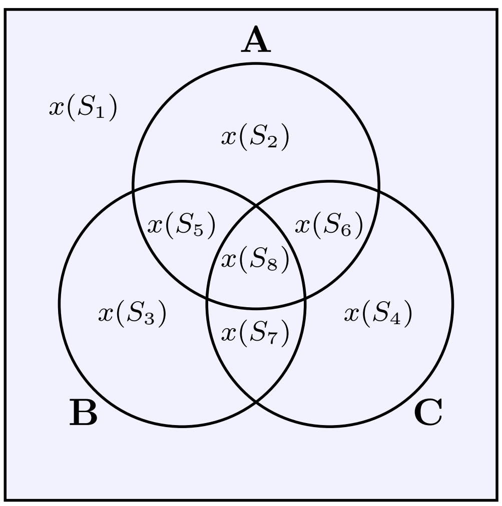
```
Format (ASR):
OP01 ⟹ OP02
OP01 ⟹ OP03
OP01 ⟹ ¬OP04 ∨ ¬OP05 ∨ ¬OP06
OP01 ⟹ ¬OP04 ∨ ¬OP05 ∨ ¬OP07
OP01 ⟹ ¬OP04 ∨ ¬OP06
OP01 ⟹ ¬OP04 ∨ ¬OP06 ∨ ¬OP07
OP01 ⟸ OP02 ∧ OP03 ∧ ¬OP06 ∧ ¬OP07
OP08 ⟹ OP09
OP08 ⟹ ¬OP10
OP08 ⟹ ¬OP11 ∨ ¬OP12
OP08 ⟸ OP09 ∧ ¬OP10 ∧ ¬OP11 ∧ ¬OP12
OP13 ⟹ ¬OP14
OP13 ⟹ ¬OP15
OP13 ⟸ ¬OP14 ∧ ¬OP15
OP14 ⟹ ¬OP15
OP16 ⟹ ¬OP17
```

Figure 3 The equivalent representation of Fig. 2 using Format (ASR).

Lemma 2. *Format (ASR) is sufficient to represent all rules for selecting options.*

3.2. Mathematical Model

We employ an approach similar to that of the probability consistency problem (Bertsimas and Tsitsiklis 1997) to formulate $\mathbb{P}$. Let S denote a subset of options, and $\mathbb{S} := \{S|S \subseteq N\}$; hence, $|\mathbb{S}| = 2^n$. We define variables $x(S) \in [0,1]$ for all $S \in \mathbb{S}$. Consider Fig. 4 for illustration. Options A, B, and C divide the universal set into 8 mutually exclusive and collectively exhaustive regions with probability values $x(.)$. The probability of some intersecting options is equivalent to the sum of all $x(S)$'s where S includes such options.



$$S_1 = \{\}$$
$$S_2 = \{A\}$$
$$S_3 = \{B\}$$
$$S_4 = \{C\}$$
$$S_5 = \{A, B\}$$
$$S_6 = \{A, C\}$$
$$S_7 = \{B, C\}$$
$$S_8 = \{A, B, C\}$$
$$p(A) = x_2 + x_5 + x_6 + x_8$$
$$p(B) = x_3 + x_5 + x_7 + x_8$$
$$p(C) = x_4 + x_6 + x_7 + x_8$$
$$p(A \wedge B) = x_5 + x_8$$
$$p(A \wedge C) = x_6 + x_8$$
$$p(B \wedge C) = x_7 + x_8$$
$$p(A \wedge B \wedge C) = x_8$$

Figure 4 Graphical illustration: three options; eight subsets; probabilities of intersections. Note: e.g., $x_2 := x(S_2)$.

The probabilities of intersections are modeled as follows:

$$p\Big(\bigwedge_{j \in S} j\Big) = \sum_{\hat{S}: S \subset \hat{S}} x(\hat{S}), \qquad \forall S \in \mathbb{S}, \tag{1}$$

$$\sum_{S \in \mathbb{S}} x(S) = 1, \tag{2}$$

$$x(S) \geq 0, \qquad \forall S \in \mathbb{S}. \tag{3}$$

Using Format (ASR), there are two forms of rules that we need to model. Consider a rule in the form of $i \Rightarrow j_1 \vee \cdots \vee j_l$. This rule is formulated as $p(j_1 \vee \cdots \vee j_l | i) = 1$—i.e., given option i is selected, at least one of options $j_1, \cdots, j_l$ will be selected with probability 1. Using *Bayes theorem* and the *inclusion-exclusion* principle, $p(i) = p\big((i \wedge j_1) \vee (i \wedge j_2) \cdots \vee (i \wedge j_l)\big)$. Expanding the right-hand-side, we obtain:

$$\begin{aligned} p(i) - \sum_{m=1}^{l} p(i \wedge j_m) + \sum_{1 \le j_{m_1} < j_{m_2} \le l} p(i \wedge j_{m_1} \wedge j_{m_2}) - \sum_{1 \le j_{m_1} < j_{m_2} < j_{m_3} \le l} p(i \wedge j_{m_1} \wedge j_{m_2} \wedge j_{m_3}) \\ + \cdots + (-1)^{(l-1)} \sum_{1 \le j_{m_1} < \cdots < j_{m_{l-1}} \le l} p(i \wedge j_{m_1} \wedge \cdots \wedge j_{m_{l-1}}) \\ +(-1)^l p(i \wedge j_1 \wedge \cdots \wedge j_l) = 0, \qquad \forall \text{ rules} : i \Rightarrow j_1 \vee \cdots \vee j_l. \end{aligned} \tag{4}$$

Now consider a rule in the form of $i \Leftarrow j_1 \wedge \cdots \wedge j_l$. This rule is equivalent to $p(i | j_1 \wedge \cdots \wedge j_l) = 1$. Again, using *Bayes theorem*, we obtain:

$$p(j_1 \wedge \cdots \wedge j_l) - p(i \wedge j_1 \wedge \cdots \wedge j_l) = 0, \qquad \forall \text{ rules} : i \Leftarrow j_1 \wedge \cdots \wedge j_l. \tag{5}$$

Once rules are written using Format (ASR), Eqs. (4) and (5) are used to formulate them in terms of probabilities of intersections that are modeled in Eqs. (1), (2), and (3). Thus, we formulate $\mathbb{P}$ as follows: $\mathbb{P} = \{\mathbf{p} = (p_1, \cdots, p_n) |\ \mathbf{p}$ satisfies Eqs. (1), (2), (3), (4), and (5)$\}$. In the remainder, we present some interesting properties of the set $\mathbb{P}$, which help us propose our algorithm in section 4. Lemma 3 presents equivalent forms of Eqs. (4) and (5), respectively, which will be used later in this section.

Lemma 3. *Eq. (4) is equivalent to* $\sum_{\substack{S: i \in S, \\ \{j_1, \cdots, j_l\} \cap S = \emptyset}} x(S) = 0$*, for all rules* $i \Rightarrow j_1 \vee \cdots \vee j_l$*. Eq. (5) is equivalent to* $\sum_{\substack{S: \{j_1, \cdots, j_l\} \subset S \\ , i \notin S}} x(S) = 0$*, for all rules* $i \Leftarrow j_1 \wedge \cdots \wedge j_l$*.*

Interestingly, the set $\mathbb{P}$ is a polyhedron with integer vertices. Let $V_{\mathbb{P}}$ be the set of vertices of $\mathbb{P}$. We define:

$$\begin{aligned} \mathbb{S}^0 := \Big\{ S \in \mathbb{S} : \ & \big(\exists \text{ rule } i \Leftarrow j_1 \wedge \cdots \wedge j_l \text{ s.t. } \{j_1, \cdots, j_l\} \subset S, i \notin S\big) \\ & \vee \ \big(\exists \text{ rule } i \Rightarrow j_1 \vee \cdots \vee j_l \text{ s.t. } i \in S, \{j_1, \cdots, j_l\} \cap S = \emptyset\big) \Big\}, \end{aligned}$$

and let $\mathbb{S}^1 := \mathbb{S} \backslash \mathbb{S}^0$.

THEOREM 1. *The set $\mathbb{P}$ is a polyhedron with integer vertices and a one-to-one correspondence exists between $\mathbb{S}^1$ and $V_{\mathbb{P}}$. Moreover, $\forall S \in \mathbb{S}^1, \exists \mathbf{p} \in V_{\mathbb{P}}$ such that $p_i = 1$, $\forall i \in S$, and 0 otherwise.*

Let $\mathbb{Y}$ denote the set of feasible configurations. We formulate $\mathbb{Y}$ using binary variables and show that a one-to-one correspondence exists between $\mathbb{Y}$, $\mathbb{S}^1$, and $V_{\mathbb{P}}$. Define binary variable y_i which is 1 if option i is chosen, and 0 otherwise. Hence, once rules are written using Format (ASR), they can be formulated as follows.

$$y_i \le y_{j_1} + y_{j_2} + \cdots + y_{j_l}, \qquad \forall \text{ rules} : i \Rightarrow j_1 \vee \cdots \vee j_l, \tag{6}$$

$$y_i + l - 1 \ge y_{j_1} + y_{j_2} + \cdots + y_{j_l}, \qquad \forall \text{ rules} : i \Leftarrow j_1 \wedge \cdots \wedge j_l, \tag{7}$$

$$y_i \in \{0,1\}, \qquad \forall i \in N. \tag{8}$$

Hence, $\mathbb{Y} = \{\mathbf{y} = (y_1, \cdots, y_n) | \mathbf{y} \text{ satisfies Eqs. (6), (7), and (8)}\}$.

THEOREM 2. *A one-to-one correspondence exists between $\mathbb{Y}$ and $V_{\mathbb{P}}$.*

Fig. 5 illustrates Theorems 1 and 2, using an example with three options: a, b, and c. Three scenarios are considered: no rules, three rules, and five rules. The set $\mathbb{P}$ and its vertices, sets $\mathbb{S}^0$ and $\mathbb{S}^1$, and the feasible configurations in each scenario are illustrated. One can see the existence of a one-to-one correspondence between $\mathbb{S}^1$, $\mathbb{Y}$, and $V_{\mathbb{P}}$.

Recall that our problem can be formulated as $\mathscr{P} : \min_{\mathbf{p} \in \mathbb{P}, \lambda \ge 0} \|\lambda \mathbf{p} - \hat{\mathbf{p}}\|$, where $\mathbb{P}$ is a polyhedron defined by Eqs. (1)-(5). Although the objective function is convex for a fixed λ, and the feasible region is a polyhedron, this problem is extremely difficult. In fact, it can be easily shown that $\mathscr{P}$ is NP-complete. Eqs (1) and (3), as well as the variable $x(.)$, are defined for all $S \in \mathbb{S}$ where $|\mathbb{S}| = 2^n$; hence, $\mathbb{P}$ is defined by more than 2^n variables and constraints. A possible approach to solving this problem without explicitly enumerating all variables and constraints is to use column generation (Kavvadias and Papadimitriou 1990, Hansen et al. 1995), which has been the most popular technique for solving PSAT problems. Column generation consists of a master problem

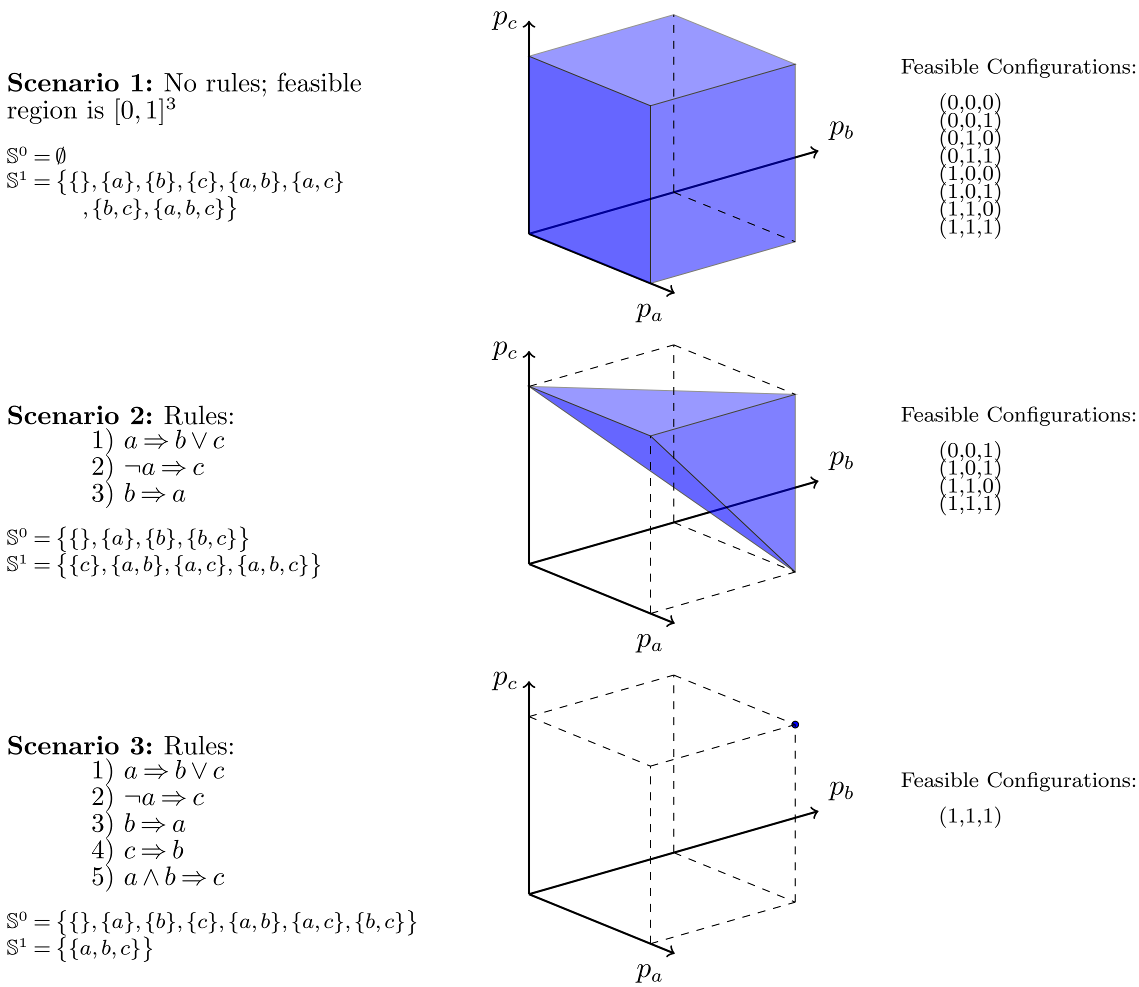


Figure 5 Graphical illustration: existence of a one-to-one correspondence between $\mathbb{S}^1$, $\mathbb{Y}$, and $V_{\mathbb{P}}$.

that is identical to the original problem but with only a small number of columns and a subproblem for determining the entering column. The subproblem can be expressed as optimizing a nonlinear 0-1 function and is solved using various techniques including algebraic methods, cutting-plane algorithms, enumerative algorithms and linearization methods (Hansen et al. 1993, 2000). An alternative is to use mixed integer packages such as CPLEX MIP (Hansen et al. 1999). Heuristics such as Tabu search can also be used (Glover 1997). Recently, Finger and De Bona (2011) used column generation to solve PSAT problems with 200 variables, 100 probability assignments, and 800 disjunctive clauses, and Finger et al. (2013) studied optimizing an objective function over all feasible solutions.

In general, column generation converges very slowly and existing literature does not include any theoretical bounds on the convergence rates. In addition, existing PSAT algorithms cannot find the closest feasible PS when the forecast PS is infeasible. In the remainder, we present an algorithm that does not suffer from these shortcomings. We prove under some mild assumptions that the worst-case convergence rate for our proposed algorithm is proportional to $\frac{1}{\sqrt{k}}$. Each iteration runs very fast and our proposed algorithm finds very good solutions after a small number of iterations. We solve real instances of problem $\mathscr{P}$ that we have received from the LAM that has 400 options and 4000 rules in a reasonable computing time. Our algorithm, implemented on a personal computer, spends less than 5000 seconds to obtain solutions with 1% error for real instances.

4. Solution Methodology

Our methodology iteratively performs two main tasks—obtaining the best direction for improvement and finding a feasible configuration that is the farthest in that direction. Due to these tasks, we refer to our method as direction-maximization (DM). We use the geometric structure of DM's mechanism to show that its worst-case convergence rate, under some assumptions, is proportional to $\frac{1}{\sqrt{k}}$. We also present a lower bound on the remaining distance to $\hat{\mathbf{p}}$, which is shown in section 5 to be extremely effective in determining when to terminate.

Our approach is similar to the Frank-Wolfe (FW) (Frank and Wolfe 1956, Demyanov and Rubinov 1970). At each iteration, the basic FW method evaluates the gradient of the objective function at the current feasible point and maximizes the linear approximation of the objective function to find the next feasible point. Depending on the step-size, the method moves towards the obtained feasible point. The maximization subproblem at each iteration needs to be simpler than the original problem. The FW produces a sequence of feasible solutions, for which, if the step-size is chosen appropriately, an $\mathcal{O}(\frac{1}{k})$ rate of convergence can be obtained Freund and Grigas (2014).

Let $\mathcal{H}_{\hat{\mathbf{p}}} = \{\mathbf{p} \in \mathbb{R}^n | \hat{\mathbf{p}}^T\mathbf{p} = \hat{\mathbf{p}}^T\hat{\mathbf{p}}, \mathbf{p} \geq 0\}$—i.e., the nonnegative part of the hyperplane generated by the normal vector $\hat{\mathbf{p}}$ and the point $\hat{\mathbf{p}}$. We denote by $\mathcal{FH}_{\hat{\mathbf{p}}}$ the feasible part of $\mathcal{H}_{\hat{\mathbf{p}}}$, which is obtained by projecting $\mathbb{P}$ onto $\mathcal{H}_{\hat{\mathbf{p}}}$—i.e., $\mathcal{FH}_{\hat{\mathbf{p}}} = \{\mathbf{p} \in \mathcal{H}_{\hat{\mathbf{p}}} | \exists \bar{\mathbf{p}} \in \mathbb{P}, \lambda \geq 0 : \lambda\bar{\mathbf{p}} = \mathbf{p}\}$. We aim to check if $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$, and, otherwise, find the best $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$. Vectors $\mathbf{y}$ and θ denote feasible configurations and their images on $\mathcal{H}_{\hat{\mathbf{p}}}$, respectively. Fig. 6 provides an illustration using two scenarios. In scenario 1, the point $\theta =$"001" for instance, is the projection of $\mathbf{y} = (0,0,1)$. Although factually $\theta = (0,0,\frac{13}{12})$, for simplicity and demonstrating that this is the projection of $\mathbf{y} = (0,0,1)$, we use the notation $\theta =$"001". An important difference between these scenarios is the boundedness of $\mathcal{H}_{\hat{\mathbf{p}}}$, which is a result of having/not having zero entries in $\hat{\mathbf{p}}$; in scenario 1, $\mathcal{H}_{\hat{\mathbf{p}}}$ is unbounded with an extreme direction $\mathbf{e}^1$, while $\mathcal{H}_{\hat{\mathbf{p}}}$ in scenario 2 is bounded. In the following lemmas, we present some of the interesting properties of the sets $\mathcal{H}_{\hat{\mathbf{p}}}$ and $\mathcal{FH}_{\hat{\mathbf{p}}}$ as well as the correspondence between the zero entries of $\hat{\mathbf{p}}$ and the extreme directions of the set $\mathcal{H}_{\hat{\mathbf{p}}}$.

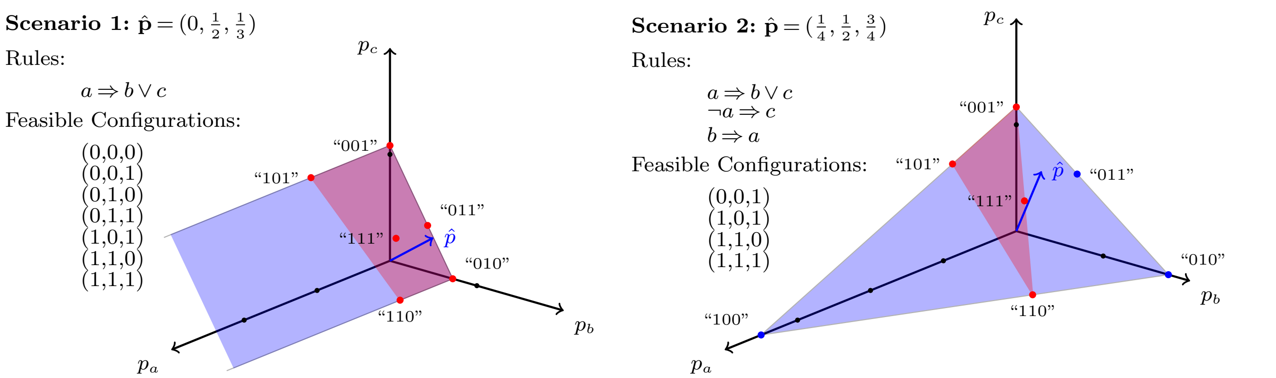


Figure 6 Graphical illustration of $\mathcal{H}_{\hat{\mathbf{p}}}$, $\mathcal{FH}_{\hat{\mathbf{p}}}$.

LEMMA 4. *Given $\hat{\mathbf{p}} \in \mathbb{R}^n$, if $\hat{p}_i > 0, \forall i \in N$, then $\mathcal{H}_{\hat{\mathbf{p}}}$ is a (n-1)-simplex with vertices $\frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i, \forall i \in N$.*

LEMMA 5. *Given $\hat{\mathbf{p}} \in \mathbb{R}^n$, if $\hat{\mathbf{p}} \geq 0$, $\hat{\mathbf{p}} \neq 0$, $I_\emptyset = \{i \in N | \hat{p}_i = 0\}$, and $I_1 = \{i \in N | \hat{p}_i > 0\}$. The vertices of $\mathcal{H}_{\hat{\mathbf{p}}}$ are then $\frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i, \forall i \in I_1$, and the extreme directions of $\mathcal{H}_{\hat{\mathbf{p}}}$ are $\mathbf{e}^i, \forall i \in I_\emptyset$.*

As a corollary of Lemma 5, if $\mathbf{p} \in \mathcal{H}_{\hat{\mathbf{p}}}$, then $\exists \gamma_i \geq 0, \forall i \in I_1$, satisfying $\sum_{i \in I_1} \gamma_i = 1$, and $\exists \xi_i \geq 0, \forall i \in I_\emptyset$ such that $\mathbf{p} = \sum_{i \in I_1} \gamma_i \frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i + \sum_{i \in I_\emptyset} \xi_i \mathbf{e}^i$.

REMARK 1. $\hat{\mathbf{p}} \in cone(\mathbb{P})$ if and only if $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$, $\forall \hat{\mathbf{p}} \in \mathbb{R}^n$.

PROPOSITION 1. *Let $\hat{\mathbf{p}} \in \mathbb{R}^n$, $\hat{\mathbf{p}} \geq 0$, $\hat{\mathbf{p}} \neq 0$, and $\tilde{\mathbf{p}}$ be the vector generated by eliminating zero elements of $\hat{\mathbf{p}}$. Then, $\hat{\mathbf{p}} \in cone(\mathbb{P})$ if and only if $\tilde{\mathbf{p}} \in \mathcal{FH}_{\tilde{\mathbf{p}}}$.*

Proposition 1 is very helpful in simplifying the problem if the objective is merely to determine feasibility when $\hat{\mathbf{p}}$ has 0-1 entries. In that case, it permits us to fix the options with a penetration rate of 0 or 1. Such simplifications are not possible if the objective is to determine the closest feasible solution.

4.1. Foundations of the Direction-Maximization Algorithm

At iteration k, $\forall k = 1, 2, \cdots$, *direction step* finds vector β^k, and *maximization step* maximizes $\beta^{kT}\theta$ over all projected feasible configurations $\theta \in \mathcal{FH}_{\hat{\mathbf{p}}}$ and obtains an optimal solution θ^{k+1}. We assume $\theta^1 \in \mathcal{FH}_{\hat{\mathbf{p}}}$ is given before the DM algorithm starts. In fact, we obtain θ^1 by solving a satisfiability problem and that θ^1 always exists since we assume in this paper that rules are satisfiable. At iteration $k = 1$, the algorithm finds β^1 and maximizes $\beta^{1T}\theta$ to obtain θ^2.

In the *direction step*, equation $\beta^k = \hat{\mathbf{p}} - \mathbf{p}^{*k}$ is used to find an improving direction, where $\mathbf{p}^{*k}$ is the nearest penetration statistic to $\hat{\mathbf{p}}$ and $\mathbf{p}^{*k}$ is in the convex hull of $\{\theta^1, \cdots, \theta^k\}$. Hence, $\mathbf{p}^{*k} = \sum_{i=1}^{k} \alpha_i^* \theta^i$, where $\alpha^* = (\alpha_1^*, \cdots, \alpha_k^*)$ is found by solving the following quadratic program.

$$\alpha^* = \arg\min_{\alpha} \quad \|\hat{\mathbf{p}} - \sum_{i=1}^{k} \alpha_i \theta^i\| \tag{9}$$

$$\text{subject to:} \quad \sum_{i=1}^{k} \alpha_i = 1 \tag{10}$$

$$\alpha_i \geq 0, \qquad \forall i = 1, \cdots, k. \tag{11}$$

This quadratic problem can be converted to a system of linear equations using the KKT conditions. Based on our experiment in section 5, solving this problem takes a negligible amount of time.

The distance between $\hat{\mathbf{p}}$ and $\mathbf{p}^{*k}$, which we refer to as *feasibility gap*, is denoted by $\mathcal{G}^k$. Then, $\mathcal{G}^k = \|\beta^k\|$, and the feasibility of $\hat{\mathbf{p}}$ is ascertained if $\mathcal{G}^k = 0$. In the remainder, we formally state this result and comment on the monotonicity of $\mathcal{G}^k$.

LEMMA 6. *At iteration k of the DM algorithm, if $\mathcal{G}^k = 0$, then $\hat{\mathbf{p}} \in cone(\mathbb{P})$.*

Lemma 7. *During the execution of the DM algorithm, $\mathcal{G}^k$ is non-increasing in k.*

In the *maximization step* at iteration k, we maximize $\beta^{kT}\theta$ to obtain θ^{k+1}:

$$\mathcal{M}(\beta^k) = \max_{\lambda,\theta,\mathbf{y}} \quad \beta^{kT}\theta \tag{12}$$

$$\text{subject to:} \quad \hat{\mathbf{p}}^T\theta = \hat{\mathbf{p}}^T\hat{\mathbf{p}}, \tag{13}$$

$$\lambda - M(1-y_i) \le \theta_i \le \lambda, \qquad \forall i \in N, \tag{14}$$

$$0 \le \theta_i \le My_i, \qquad \forall i \in N, \tag{15}$$

$$\lambda \ge 0, \tag{16}$$

$$\lambda \in \mathbb{R}, \theta \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Y}. \tag{17}$$

The above model is denoted by $\mathcal{M}(\beta^k)$ since β^k is the only parameter that varies from one iteration to the next. Recall that $\mathbf{y}$ is a vector of zeros and ones that denotes a feasible configuration, and θ is the image of $\mathbf{y}$ on $\mathcal{H}_{\hat{\mathbf{p}}}$ through the projection coefficient $\lambda \ge 0$; hence, $\theta = \lambda\mathbf{y}$. Eq. (13) models $\theta \in \mathcal{H}_{\hat{\mathbf{p}}}$. Consider $\theta = \lambda\mathbf{y}$: if $y_i = 1$, then $\theta_i = \lambda$, and otherwise, if $y_i = 0$, then $\theta_i = 0$, $\forall i \in N$. Based on this logic, $\theta = \lambda\mathbf{y}$ is represented by Eqs. (14)-(16), where M is a sufficiently large number. Finally, $\mathbf{y}$ must satisfy the rules for selecting options.

Proposition 2. *$\mathcal{M}(\beta^k)$ is NP-complete.*

Clearly, $\mathcal{M}(\beta^k)$ is computationally complex; however, in the following, we show that this problem can be solved easily if the maximum weighted satisfiability (MWSAT) can be solved in polynomial time. MWSAT is defined as follows: given a CD formula and a weight associated to each literal, find a truth assignment that satisfies the CD formula and maximizes the summation of weights for true literals (Wahlstrom 2008, Dahllof et al. 2005). We show that solving $\mathcal{M}(\beta^k)$ is equivalent to finding the maximum value of $\omega \in \mathbb{R}$ for which the optimal value of MWSAT with weight vector $\beta^k + \omega\hat{\mathbf{p}}$ is zero. This condition is denoted by MWSAT$(\beta^k + \omega\hat{\mathbf{p}}) = 0$.

Lemma 8. *(a) MWSAT$(\beta^k + \omega\hat{\mathbf{p}})$ is continuous and convex in ω, (b) MWSAT$(\beta^k + \omega\hat{\mathbf{p}}) \longrightarrow +\infty$ as $\omega \longrightarrow +\infty$, and (c) MWSAT$(\beta^k + \omega\hat{\mathbf{p}}) \le 0$ as $\omega \longrightarrow -\infty$.*

Proposition 3. *$\mathcal{M}(\beta^k)$ is equivalent to* $\max\left\{\omega \in \mathbb{R} | MWSAT(\beta^k + \omega\hat{\mathbf{p}}) = 0\right\}$.

Solving $\mathcal{M}(\beta^k)$ is equivalent to finding the greatest root of MWSAT($\beta^k + \omega\hat{\mathbf{p}}$) that can be obtained by applying the well-known Newton's method. Hence, if MWSAT is easy, then $\mathcal{M}(\beta^k)$ is also easy. Wahlstrom (2008) studied counting MWSAT and its special cases and presented effective algorithms with an upper bound on their running time. His algorithm for formulae with maximum of two variables per clause runs with an upper bound on its running time of $\mathcal{O}(1.2377^n)$. Wahlstrom (2008) also proved that his algorithm runs in polynomial time if, in addition, the degree of the formula is less than 3.

Based on our experiment in solving $\mathcal{M}(\beta^k)$, we find that the average running time is less than 100 seconds (on a personal computer) for all of the instances that we solve in this paper including the real problems we received from the LAM. In the remainder, we will show that rate of convergence of our approach is $\mathcal{O}(\frac{1}{\sqrt{k}})$ under some mild conditions, and prove its tightness for iterations $1 \leq k \leq n$.

4.2. Worst-case convergence of the DM algorithm

As the DM algorithm iterates, $\mathbf{p}^{*k}$ moves closer to $\hat{\mathbf{p}}$. In this section, we show that the amount of movement cannot be arbitrarily small by presenting an upper bound on the remaining Euclidean distance between $\mathbf{p}^{*k}$ and $\hat{\mathbf{p}}$ after k iterations. To gain insights, we begin with a three-dimensional (3-D) example and then extend the result to the general case.

Example 3 (Fig. 7): Assumptions: (i) $\hat{\mathbf{p}}$ is feasible, (ii) $\mathcal{FH}_{\hat{\mathbf{p}}}$ is a 3-D polyhedron with diameter $\sqrt{2}$, and (iii) θ^1 is located at 1 unit Euclidean distance from $\hat{\mathbf{p}}$. Consider a 3-D Cartesian coordinate system, with origin at $\hat{\mathbf{p}}$ and axis lines z_1, z_2, and z_3.

Let spr(†) denote the sphere with radius $\sqrt{2}$ which is centered at †. Because of assumption (i), θ^1 must belong to spr($\hat{\mathbf{p}}$) (Fig. 7(a)). Let θ^k_w denote the worst-case of θ^k, $\forall k \geq 1$, such that it maximizes $\mathcal{G}^k$. Then, θ^1_w must belong to the boundary of spr($\hat{\mathbf{p}}$) that we denote by bspr($\hat{\mathbf{p}}$). According to assumption (iii), we let $\theta^1 = (1, 0, 0)$.

As shown in Fig. 7(b), we obtain $\mathbf{p}^{*1} = \theta^1$ and $\mathcal{G}^1 = 1$.

Direction β^1 is shown in Fig. 7(c). Because of assumptions (i) and (ii), θ^2 must be within $\sqrt{2}$ Euclidean distance from $\hat{\mathbf{p}}$ and θ^1—i.e., $\theta^2 \in \text{spr}(\hat{\mathbf{p}}) \cap \text{spr}(\theta^1)$. In addition, because of assumption (i), we must have $\beta^{1T}\theta^2 \geq \beta^{1T}\hat{\mathbf{p}}$. Note that otherwise $\hat{\mathbf{p}}$ is separable from all feasible θ's and this contradicts assumption (i). We define half-space $\text{hsp}(\ddagger){:=}\{\mathbf{z} \in \mathbb{R}^3 : \ddagger^T\mathbf{z} \geq \ddagger^T\hat{\mathbf{p}}\}$ and denote its boundary by $\text{bhsp}(\ddagger)$. Then, $\theta^2 \in \text{spr}(\hat{\mathbf{p}}) \cap \text{spr}(\theta^1) \cap \text{hsp}(\beta^1)$. It is easy to see that $\theta_w^2 \in \text{bspr}(\theta^1) \cap \text{bhsp}(\beta^1)$. Therefore, take, w.l.o.g., $\theta^2 = (0,1,0)$.

As shown in Fig. 7(d), $\mathbf{p}^{*2}$ is the closest point in $conv\{\theta^1, \theta^2\}$ to $\hat{\mathbf{p}}$. We obtain $\mathbf{p}^{*2} = (\frac{1}{2}, \frac{1}{2}, 0)$ and $\mathcal{G}^2 = \frac{1}{\sqrt{2}}$.

Figs. 7(d) and 7(f) continue the analysis in a similar manner.

Recall that $\mathcal{G}^1 = 1$, $\mathcal{G}^2 = \frac{1}{\sqrt{2}}$, and $\mathcal{G}^3 = \frac{1}{\sqrt{3}}$. This is the tightest bound on $\mathcal{G}^k$ for $1 \leq k \leq 3$, which is confirmed by the the following example. Consider 4 options and assume that $\hat{\mathbf{p}} = (0,0,0,1)$ and there are two rules as follows: option 4 must be always chosen and at most one option from the set $\{1,2,3\}$ can be selected. There then exists 4 feasible configurations (1,0,0,1), (0,1,0,1), (0,0,1,1), and (0,0,0,1). Note that $\mathcal{H}_{\hat{\mathbf{p}}} = \{\mathbf{p} \in \mathbb{R}^4 | \mathbf{p}_4 = 1, \mathbf{p} \geq 0\}$ and all feasible configurations are in $\mathcal{H}_{\hat{\mathbf{p}}}$; hence, $\mathcal{FH}_{\hat{\mathbf{p}}}$ is a 3-D polyhedron and it is the set of all convex combinations of (1,0,0,1), (0,1,0,1), (0,0,1,1), and (0,0,0,1). The diameter of $\mathcal{FH}_{\hat{\mathbf{p}}}$ is $\sqrt{2}$. Letting $\theta^1 = (1,0,0,1)$, one could verify that $\mathcal{G}^k = \frac{1}{\sqrt{k}}$ for $1 \leq k \leq 3$.

In example 3, if one moves θ^1 further away from $\hat{\mathbf{p}}$, the convergence of the DM algorithm will improve in the next iteration. For example, if $\theta^1 \in \text{bspr}(\hat{\mathbf{p}})$, the algorithm converges in one iteration.

This worst-case convergence rate can be rigorously proven for a more general case in which $\hat{\mathbf{p}} \in cone(\mathbb{P})$ and $\mathcal{G}^1 = \frac{\mathcal{D}}{\sqrt{2}}$, where $\mathcal{D}$ denotes the diameter of the set $\mathcal{FH}_{\hat{\mathbf{p}}}$.

We define $\Theta^k = \{\theta \in \mathbb{R}^n : \|\theta - \theta^i\|^2 \leq \mathcal{D}^2, \forall i = 1, \cdots, k, \beta^{kT}(\theta - \hat{\mathbf{p}}) \geq 0\}$ and $\mathcal{A}^k = \{\alpha \in \mathbb{R}^{k+1} | \alpha \geq 0, \mathbf{1}_{k+1}^T\alpha = 1\}$. The following remark presents a way to obtain θ_w^{k+1}.

Remark 2. If $\hat{\mathbf{p}} \in cone(\mathbb{P})$, then at iteration k of the DM algorithm, we have:

$$\theta_w^{k+1} = \arg\max_{\theta \in \Theta^k} \left\{ \min_{\alpha \in \mathcal{A}^k} \left\| \sum_{i=1}^{k} \alpha_i \theta^i + \alpha_{k+1}\theta - \hat{\mathbf{p}} \right\| \right\}, \qquad \forall k \geq 1.$$

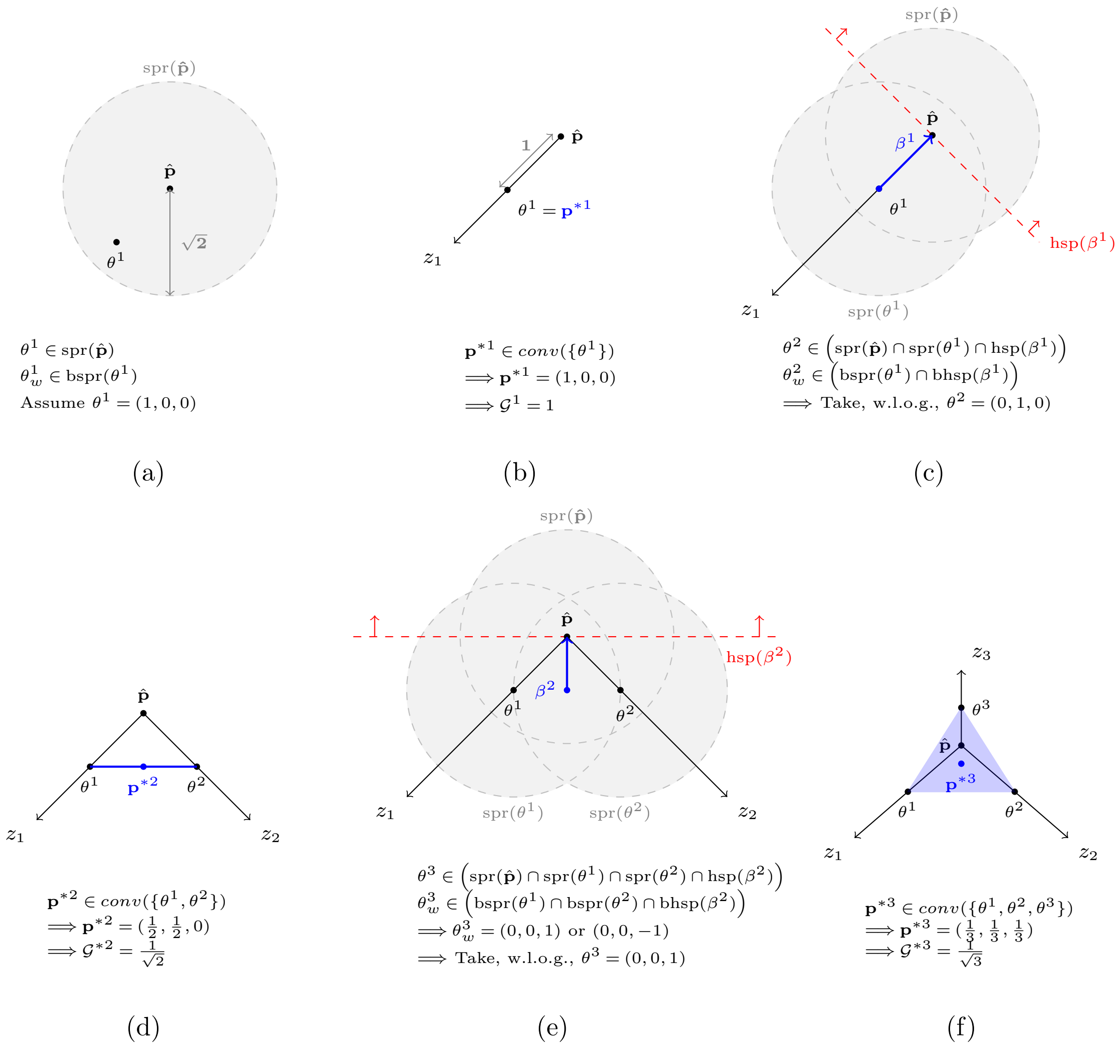


Figure 7 Illustrative example for the worst-case convergence of the DM algorithm. Note: spr(†) = sphere with radius $\sqrt{2}$ and center at †, hsp(‡)=$\{\mathbf{z} \in \mathbb{R}^3 : ‡^T\mathbf{z} \geq ‡^T\hat{\mathbf{p}}\}$, and b = boundary of.

For the general case which does not require feasibility of $\hat{\mathbf{p}}$, one needs to modify Θ^k as $\{\theta \in \mathbb{R}^n : \|\theta - \theta^i\|^2 \leq \mathcal{D}^2, \forall i = 1, \cdots, k\}$.

Theorem 3. *In the DM algorithm, if $\mathcal{G}^1 = \frac{\mathcal{D}}{\sqrt{2}}$ and $\hat{\mathbf{p}} \in cone(\mathbb{P})$, then $\mathcal{G}^k \leq \frac{\mathcal{G}^1}{\sqrt{k}}$, $\forall k = 1, 2, \cdots$. This is the tightest bound for $1 \leq k < n$.*

Theorem 3 assumes $\mathcal{G}^1 = \frac{\mathcal{D}}{\sqrt{2}}$ and $\hat{\mathbf{p}} \in cone(\mathbb{P})$, which are required to obtain clear results. As we previously discussed, if $\mathcal{G}^1 > \frac{\mathcal{D}}{\sqrt{2}}$, we can still have $\mathcal{G}^k \leq \frac{\mathcal{G}^1}{\sqrt{k}}$, $\forall k = 1, 2, \cdots$. In the remainder, we present a lower bound on $\mathcal{G}^k$ which we denote by $\mathcal{LB}^k, \forall k \geq 1$. If $\hat{\mathbf{p}}$ is infeasible, according to

Theorem 3, it is desirable to observe that $(\mathcal{G}^k - \mathcal{LB}^k)$ converges at a rate proportional to $\frac{1}{\sqrt{k}}$. In section 5, we show on a set of real instances that if $\hat{\mathbf{p}}$ is infeasible, the convergence rate of Theorem 3 is observed for $(\mathcal{G}^k - \mathcal{LB}^k)$.

4.3. Lower bound on $\mathcal{G}^k$

Let $\mathcal{L}^k$ be a lower bound on the feasibility gap at iteration k and $\mathcal{U}^k$ be an upper bound on the possible improvement in the feasibility gap at iteration k. An advantage of having a lower bound is that by comparing it with $\mathcal{G}^k$, we can decide to terminate the algorithm if the difference is satisfactory.

Definition 1. Given $\mathcal{G}^k \neq 0, \mathcal{U}^k := \min\left\{\mathcal{G}^k, \frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}\right\}$, *and* $\mathcal{L}^k := \mathcal{G}^k - \mathcal{U}^k$, $k \geq 1$.

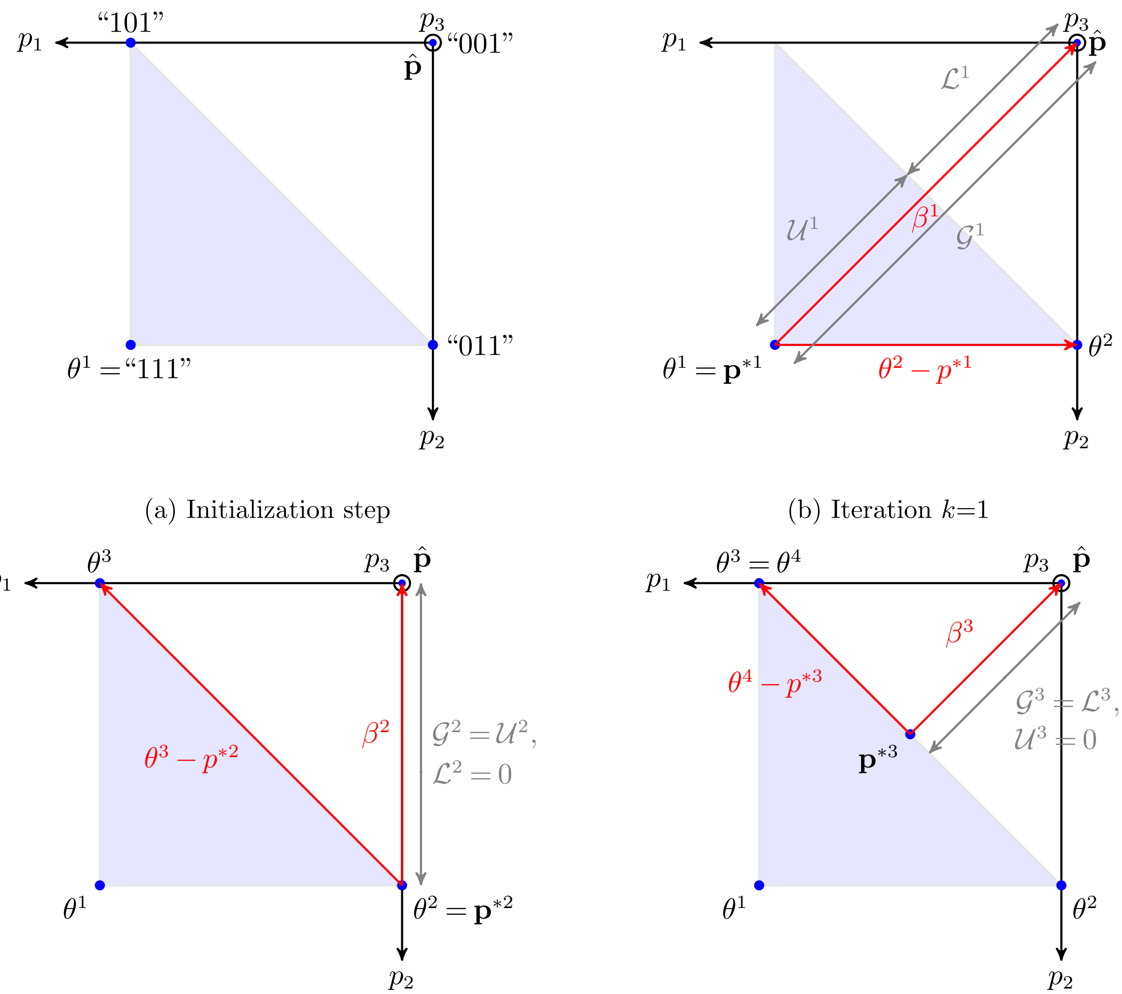
Since $\mathcal{G}^k \neq 0$, then $\beta^k \neq 0$. Let θ^{k+1} be the optimal solution of $\mathcal{M}(\beta^k)$ at iteration k. The orthogonal projection of $(\theta^{k+1} - \mathbf{p}^{*k})$ on the vector β^k is the vector $\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^{k2}}\beta^k$ and its length is given by $\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}$. The maximum improvement on $\mathcal{G}^k$ is $\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}$ and this upper bound is useful if it is less than $\mathcal{G}^k$. Thus, $\mathcal{U}^k = \min\left\{\mathcal{G}^k, \frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}\right\}$. Obviously, once we have $\mathcal{U}^k$, the lower bound on the feasibility gap is the difference between $\mathcal{U}^k$ and $\mathcal{G}^k$; hence, $\mathcal{L}^k = \mathcal{G}^k - \mathcal{U}^k$. Example 4 provides a graphical explanation on how $\mathcal{G}^k$, $\mathcal{U}^k$, and $\mathcal{L}^k$ are obtained at each iteration.

Example 4. (Fig. 8) Consider 3 options `OP01, OP02`, and `OP03`, where there exist only 3 feasible configurations {`OP01,OP03`}, {`OP02,OP03`}, and {`OP01,OP02,OP03`}, and the forecast rates of options are given by $\hat{\mathbf{p}} = (p_1, p_2, p_3) = (0,0,1)$. The set $\mathcal{FH}_{\hat{\mathbf{p}}}$ is shown in Fig. 8(a). Let the algorithm initialize at $\theta^1 = (1,1,1)$.

Iteration 1: $\mathbf{p}^{*1}$ is the closest point to $\hat{\mathbf{p}}$ in the convex hull of $\{\theta^1\}$; hence, $\mathbf{p}^{*\mathbf{1}} = \theta^1$. The vector $\beta^1 = \hat{\mathbf{p}} - \mathbf{p}^{*1}$ is shown and solving $\mathcal{M}(\beta^1)$ gives $\theta^2 = (0,1,1)$. The length of β^1 is $\mathcal{G}^1$ and the length of the projection of the vector $(\theta^2 - \mathbf{p}^{*1})$ on the vector β^1 is $\mathcal{U}^1$; moreover, $\mathcal{L}^1 = \mathcal{G}^1 - \mathcal{U}^1$.

Iteration 2: $\mathbf{p}^{*2}$ is the closest point to $\hat{\mathbf{p}}$ in the convex hull of $\{\theta^1, \theta^2\}$; hence, $\mathbf{p}^{*2} = \theta^2$. Finding β^2 and solving $\mathcal{M}(\beta^2)$, we obtain $\theta^3 = (1,0,1)$. The vector $(\theta^3 - \mathbf{p}^{*2})$ is shown. It is seen that $\mathcal{G}^2 = \mathcal{U}^2$, and $\mathcal{L}^2 = 0$.

Iteration 3: $\mathbf{p}^{*3} = (0.5, 0.5, 1)$, and β^3 are shown. Solving $\mathcal{M}(\beta^3)$, we obtain $\theta^4 = \theta^3$. □



(a) Initialization step (b) Iteration k=1 (c) Iteration k=2 (d) Iteration k=3

Figure 8 An example to illustrate $\mathcal{L}^k$ and $\mathcal{U}^k$. Note that $\mathcal{L}^k$ and $\mathcal{U}^k$ are not monotone in k.

In example 4, we note that $\mathbf{p}^{*3}$ is the closest point in $\mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$; moreover, one could see that continuing the execution of the algorithm, iteration 3 will be repeated in all iterations $k > 3$; hence, a criteria is required to recognize this situation and terminate the algorithm. In the following proposition, we show that the infeasibility of $\hat{\mathbf{p}}$ is known once $\mathcal{L}^k > 0$ and that the best penetration statistic is obtained once $\mathcal{U}^k = 0$, for some k.

Proposition 4. *Given $\mathcal{G}^k \neq 0$ at iteration k of the DM algorithm: (i) if $\mathcal{L}^k > 0$, then $\hat{\mathbf{p}} \notin cone(\mathbb{P})$, and (ii) if $\mathcal{U}^k = 0$, then $\mathbf{p}^{*k}$ is the nearest $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$.*

Having shown how $\mathcal{U}^k$ and $\mathcal{L}^k$ are used in terminating the DM algorithm, in the following, we investigate the monotonicity of $\mathcal{U}^k$ and $\mathcal{L}^k$.

LEMMA 9. *$\mathcal{L}^k$ and $\mathcal{U}^k$ are not necessarily monotone in k.*

Since $\mathcal{L}^k$ is not monotone in k, one could use the maximum of all $\mathcal{L}^k$'s obtained. Let $\mathcal{LB}^k := \max_{i=1,\cdots,k}\{\mathcal{L}^i\}$. Obviously, $\mathcal{LB}^k$ is monotone non-decreasing in k, and $\mathcal{G}^k \geq \mathcal{LB}^k$, $\forall k \geq 1$.

The DM is given in Algorithm 1. The accuracy of this algorithm is formally stated in the following theorem. The DM algorithm does not iterate forever but stops after a finite number of iterations.

Algorithm 1 Direction-Maximization Algorithm

Input: Rules, $\hat{\mathbf{p}}$.

Output: Is $\hat{\mathbf{p}} \in cone(\mathbb{P})$? If no, find the nearest $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$.

$\theta^1 :=$ an arbitrary point in $\mathcal{FH}_{\hat{\mathbf{p}}}$; ▷ `Assume rules are satisfiable!`
$\delta := 0$; ▷ $\delta = 1$ `means it is known` $\hat{\mathbf{p}} \notin cone(\mathbb{P})$, `and` $\delta = 0$ `means otherwise!`
for $k = 1, 2, 3, \cdots$ **do**
    $\mathbf{p}^{*k} := \sum_{i=1}^{k} \alpha_i^* \theta^i$, where α^* is obtained by solving Eqs. (9)-(11);
    $\beta^k := \hat{\mathbf{p}} - \mathbf{p}^{*k}$; $\mathcal{G}^k := \|\beta^k\|$;
    if $\mathcal{G}^k = 0$ **then**
        Report "$\hat{\mathbf{p}} \in cone(\mathbb{P})$"; Stop!
    end if
    Solve $\mathcal{M}(\beta^k)$; $\theta^{k+1} :=$ the optimal value of θ;
    $\mathcal{U}^k := \min\left\{\mathcal{G}^k, \frac{\beta^{kT}(\theta^{k+1} - \mathbf{p}^{*k})}{\mathcal{G}^k}\right\}$; $\mathcal{L}^k := \mathcal{G}^k - \mathcal{U}^k$;
    if $\mathcal{L}^k > 0$ & $\delta = 0$ **then**
        Report "$\hat{\mathbf{p}} \notin cone(\mathbb{P})$"; ▷ `Continue to find the nearest` $\mathbf{p} \in cone(\mathbb{P})$ `to` $\hat{\mathbf{p}}$`!`
        $\delta = 1$; ▷ `To prevent reporting ‘‘`$\hat{\mathbf{p}} \notin cone(\mathbb{P})$`" in next iterations!`
    end if
    if $\mathcal{U}^k = 0$ **then**
        Report "$\mathbf{p}^{*k}$ is the nearest $\mathbf{p} \in cone(\mathbb{P})$ to $\hat{\mathbf{p}}$"; Stop!
    end if
end for

THEOREM 4. *The DM algorithm finds the nearest $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$ and stops after finite iterations.*

5. Computational Experiments

We now report the results of our computational experiment for evaluating the effectiveness of our methodology and the sensitivity of our findings to the number of options and rules.

We implement our algorithm in IBM ILOG CPLEX Optimization Studio 12.6.1 and use a PC with Processor Intel(R) Core(TM) i5-2520M CPU 2.50GHz, 4.00 GB of RAM, and 64-bit Operating System. We define error as $E^k = \mathcal{G}^k - \mathcal{LB}^k$. Because the largest possible Euclidean distance in an n-dimensional unit hypercube is $\sqrt{n}$, we normalize $\mathcal{G}^k$, $\mathcal{LB}^k$, and E^k by dividing by $\sqrt{n}$, which also eliminates scaling issues.

The LAM offers three general categories of cars: economy sedans, luxury sedans, and SUV/trucks. These products are offered in four regions of the world—North America (NA), Latin America (LA), Europe (EU), and Asia (AS). For each car, the LAM has a master problem that includes all options and rules. Depending on the region where the car is sold, some of the options are fixed to 0 or 1, which results in elimination of some of the rules. For example, typically a truck in Mexico has fewer options than it has in the U.S. The firm offers fewer luxury options, engines, and colors in Mexico. Red trucks are very rarely purchased in Mexico. As another example, consider Vietnam where luxury cars are rarely purchased and cars have few options. On the other hand, all options are available for the same car in Europe. When all options are available, penetration rates can be very small. Table 2 presents the number of options for different car categories and across different regions.

Table 2 The approximate number of options for different products and regions.

	NA	LA	EU	AS
Economy sedans	200	100	200	100
Luxury sedans	300	250	300	250
Trucks/SUVs	400	300	400	300

The computational complexity of our problem is primarily a function of the number of options. Generally, high variety leads to a large number of options with very small penetration rates, which makes the problem very difficult and increases the possibility of infeasibility of the forecast PS.

First, we focus on the impact of number of options on the performance of our algorithm. We perform a sensitivity analysis on the number of options that are offered by the LAM, which ranges from 100 to 400. To show the performance of our algorithm in an industrial setting, we create instances that resemble the real problems encountered for the LAM. We use a master problem provided by the LAM that includes 410 options with assigned penetration rates. We convert the rules from this master problem to CD form, which results in 4056 disjunctive clauses, each of which is considered as a rule. We randomly generate instances with 100, 200, 300, and 400 options. For each size, we create 10 instances. We ignore the penetration rates of the options that are not included. The DM algorithm is applied to each instance for 1000 seconds and the average E^k is shown in Fig. 9 after 100, 500, and 1000 seconds. The length of error bars are based on the standard deviation (std) over 10 instances. Note that the range of the vertical axis varies to show the change in E^k for the different number of options. The average running time per iteration of the DM algorithm is shown in Fig. 10 with error bars representing the standard deviation (STD). Fig. 11 demonstrates the average and the STD of the number of iterations that the DM algorithm spends to determine the feasibility/infeasiblility of $\hat{\mathbf{p}}$.

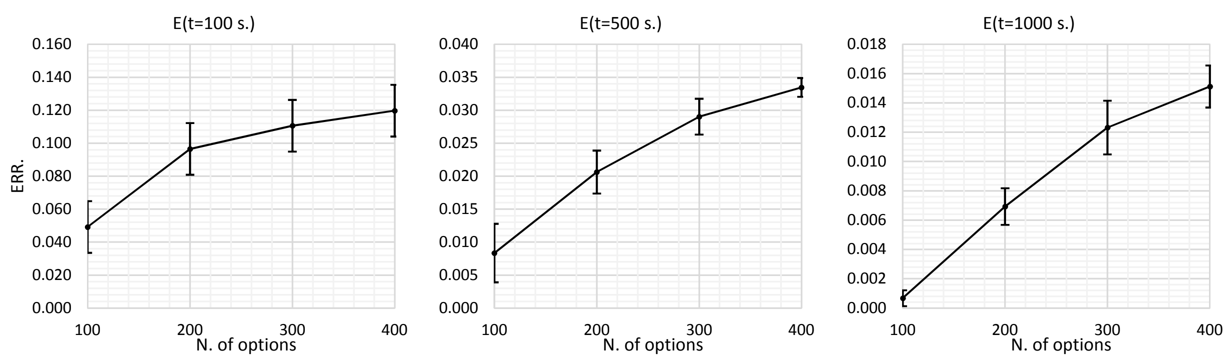


Figure 9 Normalized E^k for different number of options after 100, 500, and 1000 seconds.

Using Figs. 9-11, we make the following observations:

1. The average error after 1000 seconds is close to 0 for instances with 100 options and small for other sizes. Our algorithm appears to be very effective in solving large instances that arise in a real industrial setting and it finds very good solutions in less than 1000 seconds.

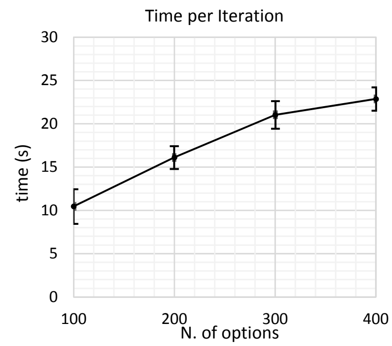


Figure 10 Average running time (seconds) per iteration for different number of options.

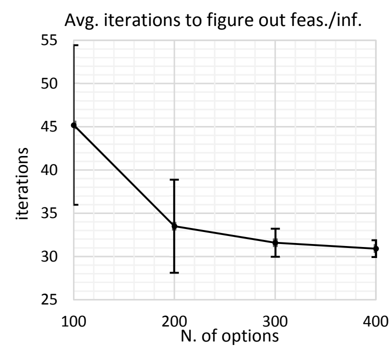


Figure 11 Average iterations to figure out the feasibility/infeasibility of $\hat{\mathbf{p}}$.

2. Average time per iteration increases as the number of options increases. This is mainly due to $\mathcal{M}(\beta^k)$, which becomes more difficult as the number of options increases.

3. An increased number of options negatively affects the performance of the DM algorithm. The normalized E^k becomes larger as the number of options increases. The main reason is due to the increased solution time of $\mathcal{M}(\beta^k)$.

4. The infeasibility of $\hat{\mathbf{p}}$ is determined quite quickly and the associated STD decreases when the number of options increases.

5. The STD of the normalized E^k significantly decreases over time—i.e., from 100 to 500 seconds and from 500 to 1000 seconds.

6. The STD of the time per iteration is almost unaffected by the number of options.

In the remainder, we examine the effects of changing the number of rules on the performance of our algorithm. The same master problem is used to generate instances with 1000, 2000, 3000, and 4000 rules. We create 10 instances for each size by randomly selecting the associated rules and relaxing the remaining rules. All instances have 410 options. We solve each instance for 1000 seconds and the result is shown in Figs. 12-14. In Fig. 14, all of the instances with 1000 and 2000 rules are still feasible after 1000 seconds; hence, we use the total number of iterations in 1000 seconds. All of the instances with 3000 and 4000 rules are infeasible; hence, Fig. 14 shows the average iteration number that the infeasibility of these instances are determined. We observe the following.

1. The DM algorithm finds very good solutions with low errors in all instances in less than 1000 seconds.

2. Average time per iteration and its STD decreases as the number of rules increases. In fact, rules strengthen $\mathcal{M}(\beta^k)$ by tightening the feasible region and help the CPLEX solver find the optimal solution very quickly. Eliminating some of the rules weakens $\mathcal{M}(\beta^k)$ and results in increased running time per iteration, which also increases the variability in the solution time of $\mathcal{M}(\beta^k)$.

3. An increased number of rules enhances the performance of the DM algorithm. A stronger $\mathcal{M}(\beta^k)$ is the only reason for this phenomenon.

4. The STD of the normalized E^k significantly decreases over time.

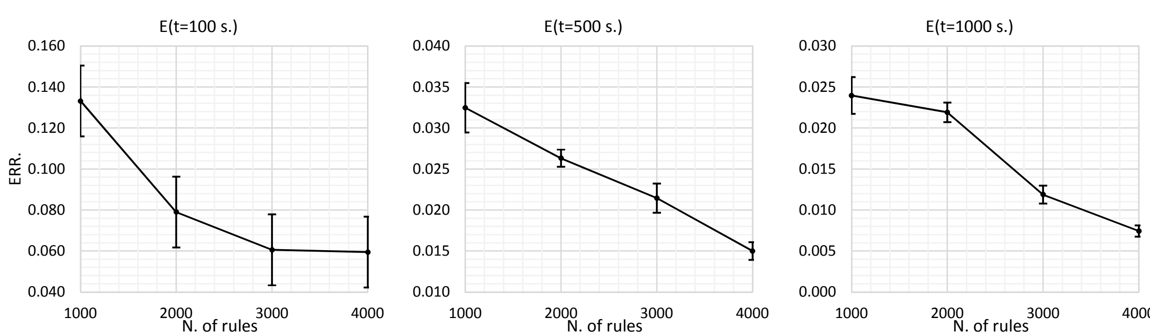


Figure 12 Normalized E^k for different numbers of rules after 100, 500, and 1000 seconds.

Therefore, our algorithm is very effective in solving real problems and an increased number of options and rules affect its performance negatively and positively, respectively.

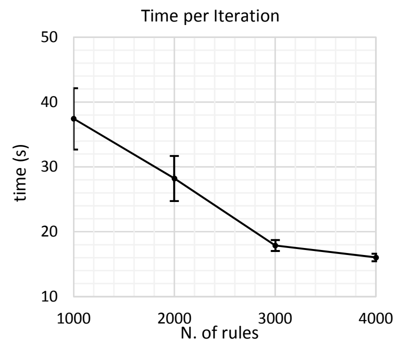


Figure 13 Average running time (seconds) per iteration for different numbers of rules.

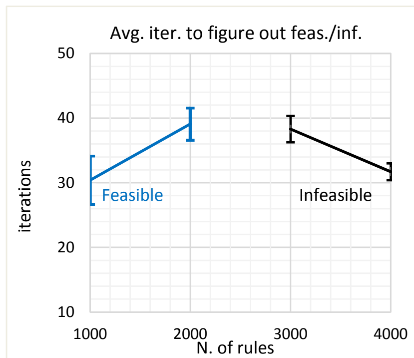


Figure 14 Average iterations to figure out the feasibility/infeasibility of $\hat{\mathbf{p}}$.

6. Experiment with Real Data

In this section, we present the results of solving three real instances that we received from the LAM. The LAM has been one of the major companies in the global auto industry for more than 100 years and delivers approximately 10 million vehicles per year to more than 100 countries. The LAM performs penetration exercises every few months and they forecast PSs approximately three years ahead, that is, PSs for 2019 model cars are forecast in 2016. Each car model is manufactured on one of 10-12 platforms. Each assignment to platform is based on the chassis size—e.g., subcompact sedans are manufactured on the same platform even though they are from different brands. On a given platform, the rules are always the same and they are usually in a simple and compact form.

The PSs are forecast by sales and marketing and sent to the engineering and finance departments for approval where some modifications are made to simplify the manufacturing process, reduce the cost of unnecessary variety, and enhance the economies of scale. The LAM has preprocessed the original set of options and rules to reduce them based on whether they are equal, opposite, or symmetric options. Equal options always appear together while for two opposite options, exactly one is always true. Options within a family that are symmetric have the same rules and are referenced by other rules in the same exact manner. For example, consider a vehicle's colors white and green; if these colors appear in exactly the same rules, one could combine white and green into one color and decompose it later to regenerate the original colors.

We received three instances from the LAM with specifications listed in Table 3.

Table 3 The specifications of our instances from the LAM.

	Options	OIRs	FCRs
Instance 1	415	3703	85
Instance 2	200	428	72
Instance 3	395	2111	97

The DM algorithm is applied to the three real instances, and normalized $\mathcal{G}^k$, $\mathcal{LB}^k$, and E^k over the running time of the algorithm and iterations are plotted in Fig. 15. Various ranges of axes for different instances is used to clearly show the behavior of the algorithm during the initial iterations. For example, while applying the DM algorithm to instance 1, the big changes of $\mathcal{G}^k$ and $\mathcal{LB}^k$ occur from 0 to 4000 seconds; hence, we only show from 0 to 4000 seconds.

In all instances, the starting error is large and ranges from 30% in instance 1 to 60% in instance 3. The error reduces significantly in a few iterations. In instances 2 and 3, the error becomes approximately 1% after 10 iterations. The 1% error is observed in instance 1 after about 40 iterations. The time per iteration varies across instances; it is big for instance 1 and small for instances 2 and 3. The number of options and rules in instance 2 are smaller than in other instances, and our algorithm spends about 2 seconds per iteration. Instance 1 and 3 require approximately 100 and 10 seconds per iteration, respectively.

To show the implication of Theorem 3, we plot the function $\frac{E^1}{\sqrt{k}}$ and compare it with E^k. In Theorem 3, we show if $\hat{\mathbf{p}}$ is feasible then the error decreases with the worst-case equal to $\frac{\mathcal{G}^1}{\sqrt{k}}$. Although we did not prove it, the extension of Theorem 3 to an infeasible case would use E^k

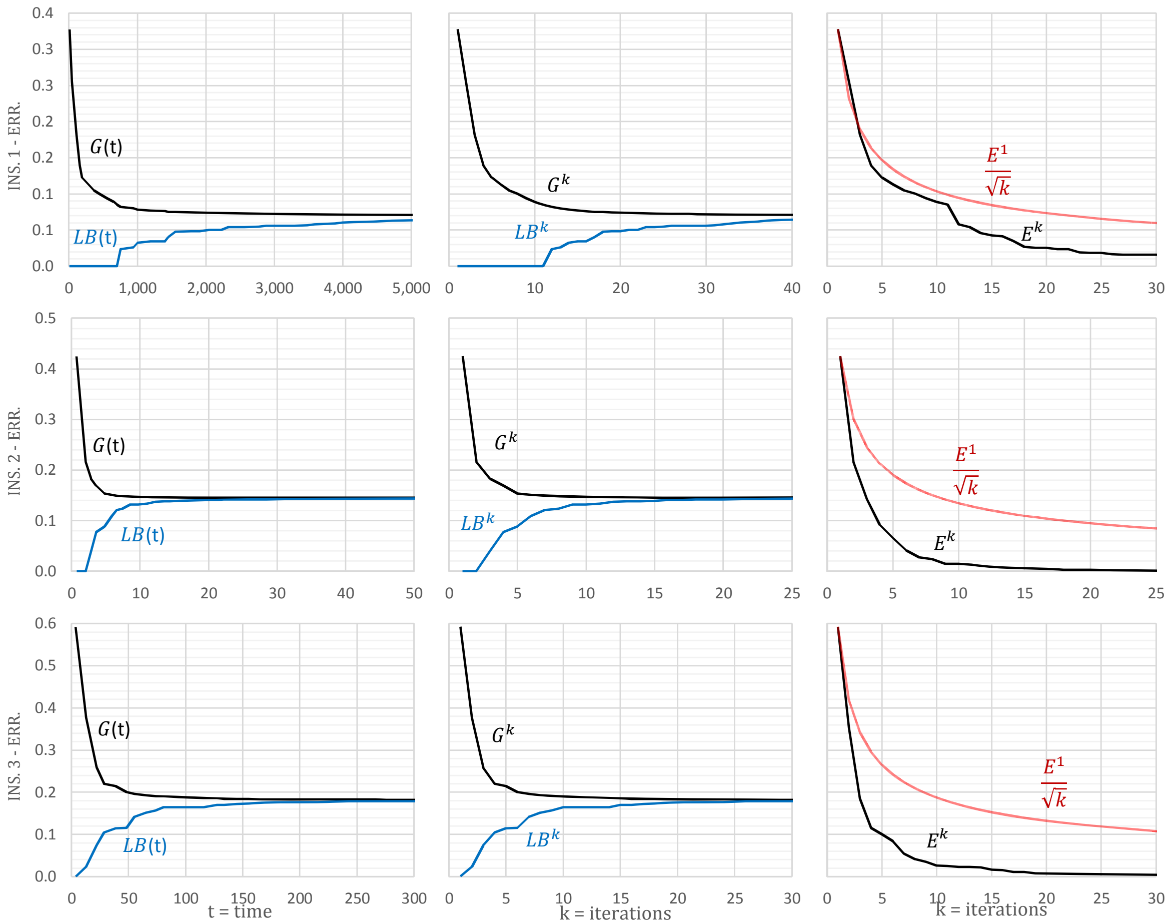


Figure 15 The performance of the DM algorithm on real instances.

instead of $\mathcal{G}^k$. If the error in iteration 1 is E^1, then the worst-case convergence of the DM algorithm suggests that $E^k \leq \frac{E^1}{\sqrt{k}}, \forall k \geq 1$. The comparison is shown in Fig. 15. It is seen that the convergence of the DM algorithm is significantly faster than the suggested worst case by Theorem 3.

All three instances are infeasible, which is determined as soon as $\mathcal{LB}^k$ becomes nonzero. This happens after 12, 3, and 2 iterations in instances 1, 2, and 3, respectively.Our methodology is then significantly fast in figuring out the infeasibility of $\hat{\mathbf{p}}$ in real instances.

Overall, the behavior of $\mathcal{G}^k$, $\mathcal{LB}^k$, and E^k is almost the same in all instances and it can be seen that, specially in the initial iterations, the error decreases rapidly. Hence, our algorithm is very effective in finding a very good solution in a short period of time.

7. Conclusions and Directions for Future Work

We study the problem of determining the feasibility of the forecast PS and finding the best feasible PS in the case of infeasibility. We present a new format for rules and formulate the problem as a quadratic program. We show that the feasible region is a polyhedron and a one-to-one correspondence exists between feasible configurations and the extreme points of this polyhedron. As the mathematical formulation is not suitable for solving large instances, we present an approach that sequentially constructs the feasible region and stops when it either verifies the feasibility of the forecast PS or it finds the best feasible PS. Although the problem is NP-complete, we show that under some mild assumptions, our approach is $\mathcal{O}(\frac{1}{\sqrt{k}})$. We test our approach on a set of real instances and observe the following results: (i) increasing the number of options and/or decreasing the number of rules add to the difficulty of the problem, (ii) our approach finds very good solutions for real instances in a reasonable time, and (iii) the convergence rate of our approach is significantly faster than the worst-case that we prove in this paper.

There are two important directions in which this work needs to be extended. We discuss them below.

The best feasible range/set: The DM algorithm finds the closest feasible PS in the case of the infeasibility of the given PS. The output of our algorithm is a point. However, in some applications, a feasible set might be more useful than a point. In k iterations, the DM algorithm finds $\{\theta^1, \cdots, \theta^{k+1}\}$ from which the closest feasible PS is obtained. We note that all points in the convex hull of $\{\theta^1, \cdots, \theta^{k+1}\}$ are feasible; hence, the starting point to find the best feasible set is to use the convex hull of $\{\theta^1, \cdots, \theta^{k+1}\}$. The first important issue is to logically define "best" based on some real applications—e.g., in the DM algorithm "best" means the minimum Euclidean distance. The second issue is to develop an effective method to improve the convex hull based on the definition of "best."

Using different weights for options: In the DM algorithm, all options have equal weights, that is, a 10% error in the penetration rate of an engine is as equally important as a 10% error in the rear-view mirror. Although is the result of using the minimum Euclidean distance in finding the best PS, this assumption may not hold in some applications. In this case, a weight vector, $\mathbf{w} = (w_1, \cdots, w_n)$, can be used to measure the relative importance of options. The objective function in Eq. (9) is modified to $\left(\hat{\mathbf{p}} - \sum_{i=1}^{k} \alpha_i \theta^i\right)^T Diag(\mathbf{w}) \left(\hat{\mathbf{p}} - \sum_{i=1}^{k} \alpha_i \theta^i\right)$. The extension of our results to this case is straightforward except for the worst-case analysis of the DM algorithm that requires additional investigation.

Appendix A: Proofs

Proof of Lemma 1

The function f is convex and differentiable, with minimum occurring at $\hat{\mathbf{p}}$. Using $\frac{\partial f}{\partial \mathbf{p}} = 0$, we obtain $\hat{\mathbf{p}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{q}_m$. We substitute $\hat{\mathbf{p}}$ by $\hat{\mathbf{p}} + \mathbf{\Delta}$, and it follows that:

$$
\begin{aligned}
f(\hat{\mathbf{p}} + \mathbf{\Delta}) &= \sum_{m=1}^{M} (\hat{\mathbf{p}} - \mathbf{q}_m + \mathbf{\Delta})^T (\hat{\mathbf{p}} - \mathbf{q}_m + \mathbf{\Delta}) \\
&= \sum_{m=1}^{M} \left((\hat{\mathbf{p}} - \mathbf{q}_m)^T (\hat{\mathbf{p}} - \mathbf{q}_m) + 2\mathbf{\Delta}^T (\hat{\mathbf{p}} - \mathbf{q}_m) + \|\mathbf{\Delta}\|^2 \right) \\
&= f(\hat{\mathbf{p}}) + 2M\mathbf{\Delta}^T (\hat{\mathbf{p}} - \frac{1}{M} \sum_{i=1}^{l} \mathbf{q}_i) + M\|\mathbf{\Delta}\|^2 \\
&= f(\hat{\mathbf{p}}) + M\|\mathbf{\Delta}\|^2
\end{aligned}
$$

Then, minimizing $f(\hat{\mathbf{p}} + \mathbf{\Delta})$ is equivalent to minimizing $\|\mathbf{\Delta}\|^2$.

Proof of Lemma 2

By convention, an empty clause is both conjunctive and disjunctive; hence, an empty clause is both in CD and DC form. W.l.o.g, all OIRs and FCRs can be written such that for each option, say option `OP01`, we have $\texttt{OP01} \Longrightarrow \texttt{RHSr}_k\texttt{(OP01)}$, $k = 1, \cdots, K_r$, for some $K_r \geq 0$, and $\texttt{OP01} \Longleftarrow \texttt{RHSl}_k\texttt{(OP01)}$, $k = 1, \cdots, K_l$, for some $K_l \geq 0$, where $\texttt{RHSr}_k\texttt{(OP01)}$ is the kth RHS associated with option `OP01` and "$\Longrightarrow$", and $\texttt{RHSl}_k\texttt{(OP01)}$ is the kth RHS associated with option `OP01` and "$\Longleftarrow$". If $K_r = 0$ (or $K_l = 0$), then no such rule exists. Each RHS can be written in either CD or DC form; hence, let $\texttt{RHSr}_k\texttt{(OP01)}$, $k = 1, \cdots, K_r$, be written as CD, and $\texttt{RHSl}_k\texttt{(OP01)}$, $k = 1, \cdots, K_l$, be written as DC. Thus, the above rules can be combined as follows: $\texttt{OP01} \Longrightarrow \texttt{RHSr}_1\texttt{(OP01)} \wedge \cdots \wedge \texttt{RHSr}_{K_r}\texttt{(OP01)}$, and $\texttt{OP01} \Longleftarrow \texttt{RHSl}_1\texttt{(OP01)} \vee \cdots \vee \texttt{RHSl}_{K_l}\texttt{(OP01)}$. Assume that the rules are written in this format.

Let option `OP01` be arbitrary; hence, $\texttt{OP01} \Longrightarrow \texttt{DIS}_1\texttt{(OP01)} \wedge \cdots \wedge \texttt{DIS}_{K_d}\texttt{(OP01)}$, for some $K_d \geq 0$, and $\texttt{OP01} \Longleftarrow \texttt{CON}_1\texttt{(OP01)} \vee \cdots \vee \texttt{CON}_{K_c}\texttt{(OP01)}$, for some $K_c \geq 0$, where $\texttt{DIS}_k\texttt{(OP01)}$ is the kth disjunctive clause in the RHS of "$\texttt{OP01} \Longrightarrow \ldots$", and $\texttt{CON}_k\texttt{(OP01)}$ is the kth conjunctive clause in the RHS of "`OP01`$\Longleftarrow$...". If $K_d = 0$ (or $K_c = 0$), then the RHS is empty. The above rules can be decomposed as follows: `OP01`$\Longrightarrow$`DIS`$_k$`(OP01)`, $k = 1, \cdots, K_d$, and `OP01`$\Longleftarrow$`CON`$_k$`(OP01)`, $k = 1, \cdots, K_c$. Therefore, Format (ASR) is obtained.

Proof of Lemma 3

The first part follows from combining Eqs. (1) and (4). The second part follows from combining Eqs. (1) and (5).

Proof of Theorem 1

Using Lemma 3, Eq. (4) is equivalent to $\sum_{\substack{S: i\in S,\\ \{j_1,\cdots,j_l\}\cap S=\emptyset}} x(S) = 0$, for all rules in the form of $i \Rightarrow j_1 \vee \cdots \vee j_l$, which is equivalent to:

$$x(S) = 0, \qquad \forall \text{ rules} : i \Rightarrow j_1 \vee \cdots \vee j_l, \text{ and } \forall S \text{ s.t. } i \in S, \{j_1, \cdots, j_l\} \cap S = \emptyset. \tag{18}$$

Similarly, using Lemma 3, Eq. (5) is equivalent to:

$$x(S) = 0, \qquad \forall \text{ rules} : i \Leftarrow j_1 \wedge \cdots \wedge j_l, \text{ and } \forall S \text{ s.t. } \{j_1, \cdots, j_l\} \subset S, i \notin S. \tag{19}$$

Using Eqs. (18) and (19), and the definition of $\mathbb{S}^0$, we have $x(S) = 0, \forall S \in \mathbb{S}^0$. Therefore, $\mathbb{P} = \{\mathbf{p} = (p_1, \cdots, p_n) | \ \mathbf{p}$ satisfies Eqs. (1), (2), (3), and $x(S) = 0, \forall S \in \mathbb{S}^0\}$. Eq. (1) does not impose any new constraint and is only used to calculate values of p_i's. Hence, consider solving Eqs. (2), (3), and $x(S) = 0, \forall S \in \mathbb{S}^0$. This system of equations include some nonnegative continuous variables of which some are fixed to zero and the rest must add up to 1. Thus, at any vertex of this system of equations, exactly one $x(.)$ where the corresponding S belongs to $\mathbb{S} \backslash \mathbb{S}^0$ is 1 and the rest are zero. Hence, the number of vertices are equivalent to $|\mathbb{S} \backslash \mathbb{S}^0|$. Moreover, using Eq. (1), we have $p_i = 0$ or 1 for all $i \in N$; therefore, the vertices of $\mathbb{P}$ are integer points.

Let us consider a particular vertex where $x(\hat{S}) = 1$ for exactly one $\hat{S}$ such that $\hat{S} \in \mathbb{S} \backslash \mathbb{S}^0$ and $x(S) = 0, \ \forall \ S \neq \hat{S}, S \in \mathbb{S}$. Using Eq. (1), we have:

$$p_i = x(\hat{S}) \mathbf{1}(i \in \hat{S}) + \sum_{S: i \in S, S \neq \hat{S}} x(S), \qquad \forall i \in N \tag{20}$$

where $\mathbf{1}(.)$ is an indicator function which returns 1 if its argument is true, and 0 otherwise. But the second term in the right-hand-side of Eq. (20) is zero and the first term is $\mathbf{1}(j \in \hat{S})$; hence, $p_i = \mathbf{1}(i \in \hat{S}), \forall i \in N$. Thus, the proof is complete.

Proof of Theorem 2

Since a one-to-one correspondence exists between $\mathbb{S}^1$ and $V_{\mathbb{P}}$, we need to prove the existence of a one-to-one correspondence between $\mathbb{S}^1$ and $\mathbb{Y}$. But this follows from noting that $\mathbb{S}^0$ is in fact the set of infeasible configurations.

Proof of Lemma 4

$\mathcal{H}_{\hat{\mathbf{p}}}$ is defined by 1 equality and n inequalities; hence, the vertices of $\mathcal{H}_{\hat{\mathbf{p}}}$ are obtained by setting $n-1$ of inequalities to equalities and solving them together with the equation $\hat{\mathbf{p}}^T\mathbf{p} = \hat{\mathbf{p}}^T\hat{\mathbf{p}}$. Since $\hat{p}_i > 0, \forall i \in N$, this results in vertices $\frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i, \forall i \in N$. Moreover, since there are n affinely independent vertices, then $\mathcal{H}_{\hat{\mathbf{p}}}$ is a (n-1)-simplex.

Proof of Lemma 5

Similar to the proof of Lemma 4, we obtain vertices $\frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i, \forall i \in I_1$ which are affinely independent. Extreme directions of $\mathcal{H}_{\hat{\mathbf{p}}}$ are the extreme points of the set $\{\mathbf{d} \in \mathbb{R}^n | \hat{\mathbf{p}}^T\mathbf{d} = 0, \mathbf{d} \geq 0, \mathbf{1}^T\mathbf{d} = 1\}$, where $\mathbf{1}$ is the vector of appropriate size with all entries equal to 1. This results in the extreme directions $\mathbf{e}^i, \forall i \in I_\emptyset$.

Proof of Remark 1

Using the definition of $\mathcal{FH}_{\hat{\mathbf{p}}}$, we have $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ if and only if the following two conditions hold: (i) $\hat{\mathbf{p}} \in \mathcal{H}_{\hat{\mathbf{p}}}$, and (ii) $\exists \bar{\mathbf{p}} \in \mathbb{P}, \lambda \geq 0$ s.t. $\lambda\bar{\mathbf{p}} = \hat{\mathbf{p}}$. Condition (i) is always satisfied and condition (ii) holds if and only if $\hat{\mathbf{p}} \in cone(\mathbb{P})$. Hence, the proof is complete.

Proof of Proposition 1

Using remark 1, it suffices to show $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ if and only if $\tilde{\mathbf{p}} \in \mathcal{FH}_{\tilde{\mathbf{p}}}$. As a corollary of Lemma 5 (also using the definitions of $\mathcal{H}_{\hat{\mathbf{p}}}$ and $\mathcal{FH}_{\hat{\mathbf{p}}}$), we have $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ if and only if

$$\hat{\mathbf{p}} = \lambda'\mathbf{p}' = \sum_{i\in I_1} \gamma_i' \frac{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}{\hat{p}_i}\mathbf{e}^i + \sum_{i\in I_\emptyset} \xi_i'\mathbf{e}^i, \tag{21}$$

for some $\lambda' \geq 0$, $\mathbf{p}' \in \mathbb{P}$, $\gamma_i' \geq 0, \forall i \in I_1$, satisfying $\sum_{i\in I_1}\gamma_i' = 1$, and $\xi_i' \geq 0, \forall i \in I_\emptyset$. Similarly, we have $\tilde{\mathbf{p}} \in \mathcal{FH}_{\tilde{\mathbf{p}}}$ if and only if

$$\tilde{\mathbf{p}} = \lambda''\mathbf{p}'' = \sum_{i\in I_1} \gamma_i'' \frac{\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}}{\tilde{p}_i}\mathbf{e}^i, \tag{22}$$

for some $\lambda'' \geq 0$, $\mathbf{p}'' \in \mathbb{P}$, and $\gamma_i'' \geq 0, \forall i \in I_1$, satisfying $\sum_{i\in I_1}\gamma_i'' = 1$.

Since $\hat{p}_i = 0, \forall i \in I_\emptyset$, then $\xi_i' = 0, \forall i \in I_\emptyset$. Moreover, $\hat{p}_i = \tilde{p}_i, \forall i \in I_1$, and $\hat{\mathbf{p}}^T\hat{\mathbf{p}} = \tilde{\mathbf{p}}^T\tilde{\mathbf{p}}$; hence, Eq. (21) is equivalent to $\hat{\mathbf{p}} = \lambda'\mathbf{p}' = \sum_{i\in I_1}\gamma_i'\frac{\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}}{\tilde{p}_i}\mathbf{e}^i$. Thus, the result follows once we set $\lambda' = \lambda''$, $\mathbf{p}' = \mathbf{p}''$, and $\gamma_i' = \gamma_i'', \forall i \in I_1$.

Proof of Lemma 6

$\mathcal{G}^k = 0$ if and only if $\beta^k = 0$; hence, $\hat{\mathbf{p}} - \mathbf{p}^{*k} = 0$ and it follows that $\hat{\mathbf{p}} = \sum_{i=1}^k \alpha_i^*\theta^i$ for some $\alpha_i^* \geq 0, \forall i = 1, \cdots, k$, satisfying $\sum_{i=1}^k \alpha_i^* = 1$. Moreover, $\theta^i \in \mathcal{FH}_{\hat{\mathbf{p}}}, \forall i = 1, \cdots, k$, and $\mathcal{FH}_{\hat{\mathbf{p}}}$ is a convex set. Thus, $\hat{\mathbf{p}} \in \mathcal{FH}_{\hat{\mathbf{p}}}$, which is also equivalent to $\hat{\mathbf{p}} \in cone(\mathbb{P})$, once remark 1 is applied.

Proof of Lemma 7

Note that $\mathcal{G}^k = \|\beta^k\| = \|\hat{\mathbf{p}} - \mathbf{p}^{*k}\|$; hence, at iteration k, $\mathcal{G}^k$ is the optimal value of the problem given in Eqs. (9)-(11). At iteration $k+1$, fixing $\alpha_{k+1} = 0$, reduces the feasible region of the problem given in Eqs. (9)-(11), and the problem becomes identical to that at iteration k; hence, fixing $\alpha_{k+1} = 0$, we find $\mathcal{G}^k$, which is not smaller than $\mathcal{G}^{k+1}$. Therefore, $\mathcal{G}^k \geq \mathcal{G}^{k+1}$, $\forall k = 1, 2, \cdots$.

Proof of Proposition 2

The result follows from using the proof by restriction and a reduction from satisfiability as it is imbedded in $\mathcal{M}(\beta^k)$.

Proof of Lemma 8

(a) Note that $\text{MWSAT}(\beta^k + \omega\hat{\mathbf{p}}) = \max_{\mathbf{y}\in\mathbb{Y}}(\beta^k + \omega\hat{\mathbf{p}})^T\mathbf{y} = \max_{\mathbf{y}_{\text{rel}}\in conv(\mathbb{Y})}(\beta^k + \omega\hat{\mathbf{p}})^T\mathbf{y}_{\text{rel}}$ where $\mathbf{y}_{\text{rel}}$ is a n-vector of continuous variables and $conv(\mathbb{Y})$ is the convex hull of the feasible configurations. The objective function, $(\beta^k + \omega\hat{\mathbf{p}})^T\mathbf{y}_{\text{rel}}$, is convex in $\mathbf{y}_{\text{rel}}$ for each ω, and $conv(\mathbb{Y})$ is convex; hence, the result follows.

(b) The proof follows from our assumptions: $\hat{\mathbf{p}} \geq 0$, $\hat{\mathbf{p}} \neq 0$, and there exists a nonzero feasible configuration.

(c) If $0 \in \mathbb{Y}$, then $\text{MWSAT}(\beta^k + (-\infty)\hat{\mathbf{p}}) = 0$; otherwise, $\text{MWSAT}(\beta^k + \omega\hat{\mathbf{p}}) \longrightarrow -\infty$ as $\omega \longrightarrow -\infty$.

Proof of Proposition 3

The proof consists of the following steps. First, we show that strong duality holds for $\mathcal{M}(\beta^k)$. Second, we derive the dual problem to show that $\mathcal{M}(\beta^k)$ is equivalent to $\max\left\{\omega \in \mathbb{R} | \text{MWSAT}(\beta^k + \omega\hat{\mathbf{p}}) \leq 0\right\}$. Finally, we use Lemma 8 to show that the condition holds as equality.

Step 1: Noting that Eqs. (14)-(16) are equivalent to $\theta = \lambda\mathbf{y}$. We substitute $\lambda\mathbf{y}$ for θ in $\mathcal{M}(\beta^k)$ and eliminate Eqs. (14)-(16). Hence,

$$\mathcal{M}(\beta^k) \Longleftrightarrow \max_{\lambda\geq 0, \mathbf{y}\in\mathbb{Y}} \lambda\beta^{kT}\mathbf{y} \text{ s.t. } \lambda\hat{\mathbf{p}}^T\mathbf{y} = \hat{\mathbf{p}}^T\hat{\mathbf{p}} \tag{23}$$

$$\Longleftrightarrow \max_{\mathbf{y}_\lambda\in cone(\mathbb{Y})} \beta^{kT}\mathbf{y}_\lambda \text{ s.t. } \hat{\mathbf{p}}^T\mathbf{y}_\lambda = \hat{\mathbf{p}}^T\hat{\mathbf{p}} \tag{24}$$

$$\Longleftrightarrow \max_{\mathbf{y}_{\lambda,\text{rel}}\in conv(cone(\mathbb{Y}))} \beta^{kT}\mathbf{y}_{\lambda,\text{rel}} \text{ s.t. } \hat{\mathbf{p}}^T\mathbf{y}_{\lambda,\text{rel}} = \hat{\mathbf{p}}^T\hat{\mathbf{p}}, \tag{25}$$

where, $\mathbf{y}_\lambda, \mathbf{y}_{\lambda,\text{rel}} \in \mathbb{R}^n$ are vectors of continuous variables, and $cone(.)$ and $conv(.)$ are respectively the cone and convex hull generated by a given set. Eq. (24) is obtained by defining variable $\mathbf{y}_\lambda \in \mathbb{R}^n$ and substituting it for $\lambda\mathbf{y}$; hence, $\mathbf{y}_\lambda = \lambda\mathbf{y} \in cone(\mathbb{Y})$. The feasible region of Eq. (24) consists of discrete points on hyperplane $\hat{\mathbf{p}}^T\mathbf{y}_\lambda = \hat{\mathbf{p}}^T\hat{\mathbf{p}}$ and objective function is linear in $\mathbf{y}_\lambda$; hence, Eq. (25) is obtained once we relax $\mathbf{y}_\lambda$ and use the convex hull of the feasible region. Since Eq. (25) is a convex optimization problem, then strong duality holds for $\mathcal{M}(\beta^k)$.

Step 2: Using Eq. (23), the dual of $\mathcal{M}(\beta^k)$ is written as follows:

$$\min_{\omega} \max_{\lambda\geq 0,\mathbf{y}\in\mathbb{Y}} \lambda\beta^{kT}\mathbf{y}+\omega\lambda\hat{\mathbf{p}}^T\mathbf{y}-\omega\hat{\mathbf{p}}^T\hat{\mathbf{p}} \tag{26}$$

$$\Longleftrightarrow \min_{\omega}\left\{-\omega\hat{\mathbf{p}}^T\hat{\mathbf{p}}+\max_{\lambda\geq 0}\left\{\max_{\mathbf{y}\in\mathbb{Y}}\lambda(\beta^k+\omega\hat{\mathbf{p}})^T\mathbf{y}\right\}\right\} \tag{27}$$

$$\Longleftrightarrow \min_{\omega}\left\{-\omega\hat{\mathbf{p}}^T\hat{\mathbf{p}}+\max_{\lambda\geq 0}\lambda\left\{\max_{\mathbf{y}\in\mathbb{Y}}(\beta^k+\omega\hat{\mathbf{p}})^T\mathbf{y}\right\}\right\} \tag{28}$$

$$\Longleftrightarrow \min_{\omega}\left\{-\omega\hat{\mathbf{p}}^T\hat{\mathbf{p}}+\max_{\lambda\geq 0}\lambda\left(\text{MWSAT}(\beta^k+\omega\hat{\mathbf{p}})\right)\right\} \tag{29}$$

$$\Longleftrightarrow \min_{\omega}-\omega\hat{\mathbf{p}}^T\hat{\mathbf{p}} \text{ s.t. MWSAT}(\beta^k+\omega\hat{\mathbf{p}})\leq 0 \tag{30}$$

$$\Longleftrightarrow \max_{\omega}\omega \text{ s.t. MWSAT}(\beta^k+\omega\hat{\mathbf{p}})\leq 0. \tag{31}$$

Eq. (27) is obtained by rearranging terms. Since $\lambda\geq 0$, we take it outside of the maximization to obtain Eq. (28). The maximization problem $\max_{\mathbf{y}\in\mathbb{Y}}(\beta^k+\omega\hat{\mathbf{p}})^T\mathbf{y}$ is in fact MWSAT with weight vector $\beta^k+\omega\hat{\mathbf{p}}$. In Eq. (29), for a fixed ω, if the optimal value of MWSAT is strictly positive, then the optimal value of the maximization problem becomes $+\infty$; hence, we are interested in the values of ω for which the optimal value of MWSAT is non-positive (Eq. 30). Finally, note that $\hat{\mathbf{p}}^T\hat{\mathbf{p}}>0$ which results in Eq. (31). Since strong duality holds, $\mathcal{M}(\beta^k)$ is equivalent to Eq. (31).

Step 3: Let $\omega^*_{\max}$ denote the greatest solution of $\text{MWSAT}(\beta^k+\omega\hat{\mathbf{p}})=0$. Using lemma 8, $\omega^*_{\max}<+\infty$, and we have $\text{MWSAT}(\beta^k+\omega\hat{\mathbf{p}})>0$ for all $\omega>\omega^*_{\max}$. Therefore, $\mathcal{M}(\beta^k)\Longleftrightarrow \max\left\{\omega\in\mathbb{R}|\text{MWSAT}(\beta^k+\omega\hat{\mathbf{p}})=0\right\}$.

Proof of Remark 2

Once θ is fixed, the minimization problem is to find the closest point in the convex hull of $\{\theta^1,\cdots,\theta^k,\theta\}$ to $\hat{\mathbf{p}}$. Since we are interested in the worst-case, we maximize over all θ's requiring that the following two constraints are satisfied. First, the Euclidean distance between any pair of feasible points must be less than or equal to the diameter of the set $\mathcal{FH}_{\hat{\mathbf{p}}}$. Second, $\beta^{kT}(\theta-\hat{\mathbf{p}})\geq 0$, which is required to have $\hat{\mathbf{p}}\in cone(\mathbb{P})$ since otherwise a hyperplane exists that separates $\hat{\mathbf{p}}$ and $\mathcal{FH}_{\hat{\mathbf{p}}}$, and hence $\hat{\mathbf{p}}$ is infeasible.

Proof of Theorem 3

This theorem is proven in two steps. First, we show if $\hat{\mathbf{p}}\in cone(\mathbb{P})$, $\mathcal{G}^1=\frac{\mathcal{D}}{\sqrt{2}}$, and $\theta^k=\theta^k_w$, $\forall k\geq 2$, then: (1) $\mathcal{G}^k=\frac{\mathcal{G}^1}{\sqrt{k}}$, $\forall 1\leq k\leq n-1$, and (2) $\mathcal{G}^k\leq\frac{\mathcal{G}^1}{\sqrt{k}}$, $\forall k\geq n$. Second, we provide an example for which we have: $\mathcal{G}^k=\frac{\mathcal{G}^1}{\sqrt{k}}$, $\forall 1\leq k\leq n-1$.

To prove the first step, we use an inductive argument to show if $\hat{\mathbf{p}}\in cone(\mathbb{P})$, $\mathcal{G}^1=\frac{\mathcal{D}}{\sqrt{2}}$, and $\theta^k=\theta^k_w$, $\forall k\geq 2$, then, w.l.o.g., $\theta^k=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^k$, $\forall k=1,\cdots,n-1$, where $\mathbf{e}^k$ is the vector with all 0's except for a 1 in the kth coordinate—e.g., $\mathbf{e}^1=(1,0,0,\cdots,0)$.

Let $k=1$ and θ^1 be given as we previously discussed. Then, $\mathbf{p}^{*1}=\theta^1$ (since $\mathbf{p}^{*k}$ is the closest point in the convex combination of the set $\{\theta^1,\cdots,\theta^k\}$ to $\hat{\mathbf{p}}$). Thus, $\frac{\mathcal{D}}{\sqrt{2}}=\mathcal{G}^1=\|\beta^1\|=\|\hat{\mathbf{p}}-\mathbf{p}^{*1}\|=\|\hat{\mathbf{p}}-\theta^1\|$; hence, the Euclidean distance between $\hat{\mathbf{p}}$ and θ^1 is $\mathcal{G}^1$. Therefore, let $\theta^1=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^1$, which is w.l.o.g. since it can always be achieved by a rotation about $\hat{\mathbf{p}}$.

For $1\le k\le n-2$, assume that $\theta^i=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^i$, $\forall i=1,\cdots,k$. In the following, we show $\theta^{k+1}=\theta_w^{k+1}=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^{k+1}$. Substituting $\theta^i=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^i$, $\forall i=1,\cdots,k$, in Remark 2, we have:

$$\begin{aligned}
\theta^{k+1}=\theta_w^{k+1} &= \arg\max_{\theta\in\Theta^k}\left\{\min_{\alpha\in\mathcal{A}^k}\left\|\sum_{i=1}^k\alpha_i(\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^i)+\alpha_{k+1}\theta-\hat{\mathbf{p}}\right\|\right\}\\
&= \arg\max_{\theta\in\Theta^k}\left\{\min_{\alpha\in\mathcal{A}^k}\left\|\mathcal{G}^1\sum_{i=1}^k\alpha_i\mathbf{e}^i+\alpha_{k+1}(\theta-\hat{\mathbf{p}})\right\|\right\}\\
&= \arg\max_{\theta\in\Theta^k}\left\{\min_{\alpha\in\mathcal{A}^k}\frac{1}{2}\left(\mathcal{G}^1\sum_{i=1}^k\alpha_i\mathbf{e}^i+\alpha_{k+1}(\theta-\hat{\mathbf{p}})\right)^T\left(\mathcal{G}^1\sum_{i=1}^k\alpha_i\mathbf{e}^i+\alpha_{k+1}(\theta-\hat{\mathbf{p}})\right)\right\}\\
&= \arg\max_{\theta\in\Theta^k}\left\{\min_{\alpha\in\mathcal{A}^k}\frac{1}{2}\alpha^TQ\alpha\right\},
\end{aligned}\tag{32}$$

where $\Theta^k=\{\theta\in\mathbb{R}^n:\|\theta-\hat{\mathbf{p}}-\mathcal{G}^1\mathbf{e}^i\|^2\le\mathcal{D}^2,\forall i=1,\cdots,k,\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}\le 0\}$, $\mathcal{A}^k=\{\alpha\in\mathbb{R}^{k+1}:\alpha\ge 0,\mathbf{1}^{(k+1)T}\alpha=1\}$, and

$$Q=\begin{pmatrix}\mathcal{G}^{1^2}\mathbf{I}_k & \mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)}\\ \mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)T} & (\theta-\hat{\mathbf{p}})^T(\theta-\hat{\mathbf{p}})\end{pmatrix}$$

where $\mathbf{I}_k$ is a $k\times k$ identity matrix, $(\theta-\hat{\mathbf{p}})^{(k)}$ is a k-vector created by the first k entries of the vector $(\theta-\hat{\mathbf{p}})$, and Q is a $(k+1)\times(k+1)$ matrix.

Let $\mathscr{S}$ denote the Schur complement of the block $\mathcal{G}^{1^2}\mathbf{I}_k$ of the matrix Q; hence, $\mathscr{S}=(\theta-\hat{\mathbf{p}})^T(\theta-\hat{\mathbf{p}})-\mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)T}(\mathcal{G}^{1^2}\mathbf{I}_k)^{-1}\mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)}=\sum_{i=k+1}^n(\theta_i-\hat{p}_i)^2\ge 0$. The matrix Q is positive semi-definite, which can be shown by performing elementary row operations on Q and converting it to an upper diagonal matrix where the diagonal entries, and hence the eigenvalues, are $\mathcal{G}^{1^2},\cdots,\mathcal{G}^{1^2},\mathscr{S}$. For this, one needs to multiply rows $1,\cdots,k$ by $\frac{(\theta_i-\hat{p}_i)}{\mathcal{G}^1}$ and subtract their sum from row $k+1$. We have $\mathcal{G}^{1^2}>0$; then, if $\mathscr{S}>0$, Q is positive definite, and hence invertible. For now, we assume $\mathscr{S}>0$, and hence Q is positive definite and invertible, which is confirmed by the solution. We will analyse $\mathscr{S}=0$ later in this proof.

For a fixed θ, the problem $\min_{\alpha\in\mathcal{A}^k}\frac{1}{2}\alpha^TQ\alpha$ is convex since Q is positive definite and $\mathcal{A}^k$ is a convex set. The dual of this problem is $\max_{\vartheta\ge 0,\nu}g(\vartheta,\nu)$, where $g(\vartheta,\nu)=\inf_\alpha\mathscr{L}(\alpha,\vartheta,\nu)$, $\mathscr{L}(\alpha,\vartheta,\nu)=\frac{1}{2}\alpha^TQ\alpha-(\vartheta-\nu\mathbf{1}^{(k+1)})^T\alpha-\nu$, $\vartheta\in\mathbb{R}^{k+1}$, and $\nu\in\mathbb{R}$. Since Q is positive definite, then $\mathscr{L}(\alpha,\vartheta,\nu)$ is convex in α. Since $\mathscr{L}(\alpha,\vartheta,\nu)$ is differentiable, the minimizer of $\mathscr{L}(\alpha,\vartheta,\nu)$ is found by $\nabla_\alpha\mathscr{L}(\alpha,\vartheta,\nu)=0$, which results in $\alpha^*=Q^{-1}(\vartheta-\nu\mathbf{1}^{(k+1)})$, where:

$$Q^{-1}=\begin{pmatrix}\mathcal{G}^{1^{-2}}\mathbf{I}_k & 0\\ 0 & 0\end{pmatrix}+\frac{1}{\mathcal{G}^{1^2}\mathscr{S}}\begin{pmatrix}(\theta-\hat{\mathbf{p}})^{(k)}(\theta-\hat{\mathbf{p}})^{(k)T} & -\mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)}\\ -\mathcal{G}^1(\theta-\hat{\mathbf{p}})^{(k)T} & \mathcal{G}^{1^2}\end{pmatrix}$$

Hence, using α^*, we obtain $g(\vartheta,\nu)=\inf_\alpha \mathscr{L}(\alpha,\vartheta,\nu)=-\frac{1}{2}(\vartheta-\nu\mathbf{1}^{(k+1)})^TQ^{-1}(\vartheta-\nu\mathbf{1}^{(k+1)})-\nu$. The function $g(\vartheta,\nu)$ is concave in ϑ; hence, fixing ν, the maximum of $g(\vartheta,\nu)$ occurs in $\nabla_\vartheta g(\vartheta,\nu)=Q^{-1}(\vartheta-\nu\mathbf{1}^{(k+1)})=0$ or at the boundary $\vartheta=0$. From $\nabla_\vartheta g(\vartheta,\nu)=0$, it follows that $\vartheta=\nu\mathbf{1}^{(k+1)}$, and $\max_{\vartheta\geq 0,\nu} g(\vartheta,\nu)=\max_\nu -\nu=+\infty$, with $\nu=-\infty$; hence, $\vartheta<0$ which is a contradiction. Therefore, optimal ϑ is 0; thus, $\max_{\vartheta\geq 0,\nu} g(\vartheta,\nu)=\max_\nu -\frac{1}{2}\nu^2\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)}-\nu$. Since $\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)}>0$, then this problem is maximized in $\nu=-1/(\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)})$. Substituting optimal ν, we obtain $\max_{\vartheta\geq 0,\nu} g(\vartheta,\nu)=1/(\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)})$. Substituting this result in Eq. (32), we have $\theta^{k+1}=\underset{\theta\in\Theta^k}{\arg\max}\{1/(\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)})\}=\underset{\theta\in\Theta^k}{\arg\min}\{\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)}\}$. Moreover, using the definition of Q^{-1}, we have $\mathbf{1}^{(k+1)T}Q^{-1}\mathbf{1}^{(k+1)}=\frac{1}{\mathcal{G}^{1^2}}\left(k+\frac{1}{\mathscr{S}}\left(\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-\mathcal{G}^1\right)^2\right)$; hence, since $\mathcal{G}^{1^2}>0$ and k is constant, we have:

$$\theta^{k+1}=\theta_w^{k+1}=\underset{\theta\in\Theta^k}{\arg\min}\frac{1}{\mathscr{S}}\left(\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-\mathcal{G}^1\right)^2 \tag{33}$$

Consider the constraint $\|\theta-\hat{\mathbf{p}}-\mathcal{G}^1\mathbf{e}^i\|^2\leq\mathcal{D}^2,\forall i=1,\cdots,k$. This inequality can be written as $(\theta-\hat{\mathbf{p}})^T(\theta-\hat{\mathbf{p}})-2\mathcal{G}^1\mathbf{e}^{iT}(\theta-\hat{\mathbf{p}})+\mathcal{G}^{1^2}\leq\mathcal{D}^2,\forall i=1,\cdots,k$. Noting that $(\theta-\hat{\mathbf{p}})^T(\theta-\hat{\mathbf{p}})=\mathscr{S}+(\theta-\hat{\mathbf{p}})^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}$, and $\mathcal{D}^2=2\mathcal{G}^{1^2}$, we have $\mathscr{S}+(\theta-\hat{\mathbf{p}})^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-2\mathcal{G}^1\mathbf{e}^{iT}(\theta-\hat{\mathbf{p}})\leq\mathcal{G}^{1^2},\forall i=1,\cdots,k$. Thus, using Eq. (33), we have:

$$\theta^{k+1}=\theta_w^{k+1}=\underset{\theta\in\mathbb{R}^n}{\arg\min}\ \frac{1}{\mathscr{S}}\left(\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-\mathcal{G}^1\right)^2 \tag{34}$$

$$\text{s.t.}\ \ \mathscr{S}+(\theta-\hat{\mathbf{p}})^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-2\mathcal{G}^1\mathbf{e}^{iT}(\theta-\hat{\mathbf{p}})\leq\mathcal{G}^{1^2},\forall i=1,\cdots,k \tag{35}$$

$$\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}\leq 0. \tag{36}$$

Summing Eq. (35) over $i=1,\cdots,k$, and dividing by k, we have $\mathscr{S}+(\theta-\hat{\mathbf{p}})^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}-\frac{2}{k}\mathcal{G}^1\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}\leq\mathcal{G}^{1^2}$. Noting that $\mathscr{S}>0$, $(\theta-\hat{\mathbf{p}})^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}\geq 0$, and $\mathbf{1}^{(k)T}(\theta-\hat{\mathbf{p}})^{(k)}\leq 0$ (see Eq. 36), this problem is optimized if $(\theta-\hat{\mathbf{p}})^{(k)}=0$ and $\mathscr{S}=\mathcal{G}^{1^2}$. Hence, the optimal θ must satisfy the following conditions: (i) the vector $(\theta-\hat{\mathbf{p}})$ must be orthogonal to $(\theta^1-\hat{\mathbf{p}}),\cdots,(\theta^k-\hat{\mathbf{p}})$, and (ii) the Euclidean distance between the optimal θ and $\hat{\mathbf{p}}$ is equal to $\mathcal{G}^1$. Thus, although the optimal solution might not be unique, we let, w.l.o.g., $\theta^{k+1}=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^{k+1}$, which satisfies the above two conditions for optimality.

Using $\theta^{k+1}=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^{k+1}$, we have $Q=\mathcal{G}^{1^2}\mathbf{I}_{k+1}$, $Q^{-1}=\mathcal{G}^{1^{-2}}\mathbf{I}_{k+1}$, and $\nu=-\frac{\mathcal{G}^{1^2}}{k+1}$; hence, $\alpha^*=Q^{-1}(\vartheta-\nu\mathbf{1}^{(k+1)})=\frac{1}{k+1}\mathbf{1}^{(k+1)}$.

Thus, we showed that $\theta^k=\hat{\mathbf{p}}+\mathcal{G}^1\mathbf{e}^k$ for all $1\leq k\leq n-1$, and α^* at iteration k is $\frac{1}{k}\mathbf{1}^{(k)}$; hence, $\mathbf{p}^{k*}=\frac{1}{k}\sum_{i=1}^k\theta^i=\hat{\mathbf{p}}+\frac{\mathcal{G}^1}{k}\mathbf{1}^{(k)}$. Then, we have $\beta^k=\hat{\mathbf{p}}-\mathbf{p}^{*k}=-\frac{\mathcal{G}^1}{k}\mathbf{1}^{(k)}$, and it follows that $\mathcal{G}^k=\|\beta^k\|=\frac{\mathcal{G}^1}{\sqrt{k}}$.

Consider the case $\mathscr{S}=0$, which happens for $k \geq n$ as well. As we previously discussed, the problem given in Eq. (32) is optimized if $(\theta-\hat{\mathbf{p}})$ is orthogonal to $(\theta^1-\hat{\mathbf{p}}),\cdots,(\theta^k-\hat{\mathbf{p}})$; however, if $\mathscr{S}=0$, then $(\theta-\hat{\mathbf{p}})$ is in the linear combination of $(\theta^1-\hat{\mathbf{p}}),\cdots,(\theta^k-\hat{\mathbf{p}})$. Thus, $\mathscr{S}=0$ leads to suboptimal solutions. Hence, our earlier assumption $\mathscr{S}>0$ is valid. Moreover, $\mathcal{G}^k \leq \frac{\mathcal{G}^1}{\sqrt{k}}$ for $k \geq n$.

As the second step of the proof, we provide an example to show $\frac{\mathcal{G}^1}{\sqrt{k}}$ is the tightest upper bound on $\mathcal{G}^k$ for $1 \leq k \leq n-1$.

Example 5: Consider options $i=1,\cdots,n$ and assume that $\hat{\mathbf{p}}=\mathbf{e}^n$ and two FCRs are given as follows: (i) an `E` type family that consists of only option n, and (ii) an `L` type family that consists of options $1,\cdots,n-1$—i.e., option n must always be chosen and at most one of options $1,\cdots,n-1$ can be selected. Then, there exists n feasible configurations as follows: $\mathbf{e}^1+\mathbf{e}^n,\mathbf{e}^2+\mathbf{e}^n,\cdots,\mathbf{e}^{n-1}+\mathbf{e}^n,\mathbf{e}^n$. Note that $\mathcal{H}_{\hat{\mathbf{p}}}=\{\mathbf{p}\in\mathbb{R}^n|p_n=1,\mathbf{p}\geq 0\}$, and all feasible configurations are in $\mathcal{H}_{\hat{\mathbf{p}}}$; hence, $\mathcal{FH}_{\hat{\mathbf{p}}}$ is the convex combination of the n feasible configurations, which are the vertices of $\mathcal{FH}_{\hat{\mathbf{p}}}$. The diameter of $\mathcal{FH}_{\hat{p}}$ is $\mathcal{D}=\sqrt{2}$.

Let, w.l.o.g., $\theta^1=\mathbf{e}^1+\mathbf{e}^n$:

Iteration k, for $1\leq k\leq n-1$: $\mathbf{p}^{*k}=\frac{1}{k}(\mathbf{e}^1+\cdots+\mathbf{e}^k)+\mathbf{e}^n$, $\beta^k=\hat{\mathbf{p}}-\mathbf{p}^{*k}=-\frac{1}{k}(\mathbf{e}^1+\cdots+\mathbf{e}^k)$, and $\mathcal{G}^k=\|\beta^k\|=\frac{1}{\sqrt{k}}$. For $1\leq k\leq n-2$, one could verify that all of the points in the set $\{\mathbf{e}^{k+1}+\mathbf{e}^n,\mathbf{e}^{k+2}+\mathbf{e}^n,\cdots,\mathbf{e}^{n-1}+\mathbf{e}^n,\mathbf{e}^n\}$ are optimal for the problem $\mathcal{M}(\beta^k)$; then, let, w.l.o.g., $\theta^{k+1}=\mathbf{e}^{k+1}+\mathbf{e}^n$. For $k=n-1$, $\mathbf{e}^n$ is the only optimal solution for $\mathcal{M}(\beta^{n-1})$; hence, let $\theta^n=\mathbf{e}^n$.

Iteration n: $\mathbf{p}^{*n}=\mathbf{e}^n$, $\beta^n=\hat{\mathbf{p}}-\mathbf{p}^{*n}=0$, and $\mathcal{G}^n=\|\beta^n\|=0$. Then, the DM algorithm stops and reports that $\hat{\mathbf{p}}\in cone(\mathbb{P})$.

Thus, since in example 5 we have $\mathcal{G}^k=\frac{\mathcal{G}^1}{\sqrt{k}}$, for all $1\leq k\leq n-1$, we have proved that $\frac{\mathcal{G}^1}{\sqrt{k}}$ is the tightest upper bound on $\mathcal{G}^k$ for $1\leq k\leq n-1$.

Proof of Proposition 4

(i) Let $\mathcal{G}^k\neq 0$ and $\mathcal{L}^k>0$. We have $\mathcal{L}^k=\mathcal{G}^k-\mathcal{U}^k=\mathcal{G}^k-\min\left\{\mathcal{G}^k,\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}\right\}=\mathcal{G}^k+\max\left\{-\mathcal{G}^k,-\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}\right\}=\max\left\{0,\mathcal{G}^k-\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}\right\}$; hence, since $\mathcal{L}^k>0$, then $\mathcal{G}^k-\frac{\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})}{\mathcal{G}^k}>0$, and it follows that $\beta^{kT}(\theta^{k+1}-\mathbf{p}^{*k})<\mathcal{G}^{k2}=\beta^{kT}\beta^k=\beta^{kT}(\hat{\mathbf{p}}-\mathbf{p}^{*k})$; thus, $\beta^{kT}\hat{\mathbf{p}}>\beta^{kT}\theta^{k+1}$. On the other hand, since θ^{k+1} is the maximizer of $\mathcal{M}(\beta^k)$, then we have $\beta^{kT}\mathbf{p}\leq\beta^{kT}\theta^{k+1}$, $\forall\mathbf{p}\in\mathcal{FH}_{\hat{\mathbf{p}}}$. Therefore, we showed that $\beta^{kT}\mathbf{p}=\beta^{kT}\theta^{k+1}$ is a hyperplane that separates $\hat{\mathbf{p}}$ and $\mathcal{FH}_{\hat{\mathbf{p}}}$; hence, $\hat{\mathbf{p}}\notin\mathcal{FH}_{\hat{\mathbf{p}}}$, and it follows that $\hat{\mathbf{p}}\notin cone(\mathbb{P})$.

(ii) Let $\mathcal{G}^k\neq 0$ and $\mathcal{U}^k=0$; by the definition of $\mathcal{U}^k$, we have $\beta^{kT}\theta^{k+1}=\beta^{kT}\mathbf{p}^{*k}$. Since θ^{k+1} is the maximizer of $\mathcal{M}(\beta^k)$, then we have $\beta^{kT}\mathbf{p}\leq\beta^{kT}\theta^{k+1}$, $\forall\mathbf{p}\in\mathcal{FH}_{\hat{\mathbf{p}}}$. On the other hand, we have $0<\mathcal{G}^{k2}=\beta^{kT}\beta^k=\beta^{kT}(\hat{\mathbf{p}}-\mathbf{p}^{*k})$, then $\beta^{kT}\hat{\mathbf{p}}>\beta^{kT}\mathbf{p}^{*k}=\beta^{kT}\theta^{k+1}$. Hence, we showed that $\beta^{kT}\mathbf{p}=\beta^{kT}\theta^{k+1}$ is a hyperplane that separates $\hat{\mathbf{p}}$ and $\mathcal{FH}_{\hat{\mathbf{p}}}$. Moreover, since $\mathbf{p}^{*k}$ belongs to $\mathcal{FH}_{\hat{\mathbf{p}}}$ and the separating hyperplane, and the vector $(\hat{\mathbf{p}}-\mathbf{p}^{*k})$ is orthogonal to the separating hyperplane, then $\mathbf{p}^{*k}$ is the nearest point in $\mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$.

Proof of Lemma 9

Fig. 8 provides a counter example for the monotonicity of $\mathcal{L}^k$ and $\mathcal{U}^k$. Note that $\mathcal{L}^1 = \frac{1}{\sqrt{2}}$, $\mathcal{L}^2 = 0$, $\mathcal{L}^3 = \frac{1}{\sqrt{2}}$. Hence, $\mathcal{L}^k$ is not necessarily monotone in k. The same is true for $\mathcal{U}^k$.

Proof of Theorem 4

The first part of the theorem follows from Lemma 6 and Proposition 4. To prove the DM algorithm stops after finite iterations, we first claim that: if solving $\mathcal{M}(\beta^k)$ returns θ^{k+1} such that $\theta^{k+1} \in \{\theta^1, \cdots, \theta^k\}$, then the algorithm stops. To prove this claim, note that at iteration k of the DM algorithm, the hyperplane $\beta^{kT}\mathbf{p} = \beta^{kT}\mathbf{p}^{*k}$ separates $\hat{\mathbf{p}}$ and $\{\theta^1, \cdots, \theta^k\}$; moreover, $\mathbf{p}^{*k}$ and at least one of $\theta^1, \cdots, \theta^k$, say θ^j, belong to this separating hyperplane. Suppose $\theta^{k+1} \in \{\theta^1, \cdots, \theta^k\}$, then it must also belong to the separating hyperplane; hence, w.l.o.g. assume $\theta^{k+1} = \theta^j$. Then, $\beta^{kT}(\theta^{k+1} - \mathbf{p}^{*k}) = \beta^{kT}(\theta^j - \mathbf{p}^{*k}) = 0$; hence, $\mathcal{U}^k = 0$, and the algorithm stops.

Thus, we showed that the DM algorithm either finds a new θ at each iteration or stops. Noting that the number of feasible configurations is finite, the algorithm stops after finite iterations.

Endnotes

1. See hyundaiusa/tucson website for the complete detail.

References

Amilhastre J, Fargier H, Marquis P (2002) Consistency restoration and explanations in dynamic CSPsapplication to configuration. *Artificial Intelligence* 135(1):199–234.

Balseiro SR, Feldman J, Mirrokni V, Muthukrishnan S (2014) Yield optimization of display advertising with ad exchange. *Management Science* 60(12):2886-2907.

Barker VE, O'Connor DE, Bachant J, Soloway E (1989) Expert systems for configuration at Digital: XCON and beyond. *Communications of the ACM* 32(3):298–318.

Bertsimas D, Tsitsiklis JN (1997) Introduction to linear optimization (Vol. 6). Belmont, MA: Athena Scientific.

Caro F, Jeremie G (2007) Dynamic assortment with demand learning for seasonal consumer goods. *Management Science* 53(2):276-292.

Chen Z, Wang L (2010) Personalized product configuration rules with dual formulations: A method to proactively leverage mass confusion. *Expert Systems with Applications* 37(1):383–392.

Dahllof V, Jonsson P, Wahlstrom M (2005) Counting models for 2SAT and 3SAT formulae. *Theoretical Computer Science* 332(1):265–291.

Demyanov V, Rubinov A (1970) Approximate Methods in Optimization Problems. *American Elsevier Publishing.* New York.

Feitzinger E, Lee HL (1997) Mass customization at Hewlett-Packard: the power of postponement. Harvard business review, 75, 116-123.

Finger M, De Bona G (2011). Probabilistic satisfiability: Logic-based algorithms and phase transition. *In IJCAI* 528-533.

Finger M, Le Bras R, Gomes CP, Selman B (2013) Solutions for hard and soft constraints using optimized probabilistic satisfiability. *In Theory and Applications of Satisfiability TestingSAT* 233-249. Springer Berlin Heidelberg.

Franke N, Piller F (2004) Toolkits for user innovation and design: An exploration of user interaction and value creation. *Journal of Product Innovation Management* 21(6):401415.

Frank M, Wolfe P (1956) An algorithm for quadratic programming. *Naval Res. Logist. Q.* 3:95110.

Freund RM, Grigas P (2014) New analysis and results for the FrankWolfe method. *Mathematical Programming* 1-32.

Fohn M, Liau JS, Greef AR, Young RE, O'Grady PJ (1995) Configuration computer systems through constraint-based modeling and interactive constraint satisfaction. *Computers in Industry* 27(1):321.

Georgakopoulos G, Kavvadias D, Papadimitriou CH (1988) Probabilistic satisfiability. *J. of Complexity* 4(1):111.

Glover F (1997) Laguna M: Tabu Search.

Hansen P, Jaumard B, Mathon V (1993) State-of-the-Art SurveyConstrained Nonlinear 01 Programming. *ORSA Journal on Computing* 5(2):97-119.

Hansen P, Jaumard B, Nguetse GBD, De Aragao MP (1995) Models and algorithms for probabilistic and Bayesian logic (pp. 1862-1868). Institute of Informatics University of Fribourg.

Hansen P, Jaumard B, Parreira AD (1999) On the Relations between Probabilistic Logic and p-CMS. *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* 56-63. Morgan Kaufmann Publishers Inc.

Hansen P, Jaumard B, De Aragao MP, Chauny F, Perron S (2000) Probabilistic satisfiability with imprecise probabilities. *International Journal of Approximate Reasoning* 24(2):171-189.

Huffman C, Kahn B (1998) Variety for sale: Mass customization or mass confusion. *Journal of Retailing* 74:491513.

Kavvadias D, Papadimitriou CH (1990) A linear programming approach to reasoning about probabilities. *Annals of Mathematics and Artificial Intelligence*, 1(1-4), 189-205.

Kok AG, Fisher ML, Vaidyanathan R (2008) Assortment planning: Review of literature and industry practice. *In Retail supply chain management* 99-153. Springer US.

Kristianto Y, Helo P, Jiao RJ (2015) A system level product configurator for engineer-to-order supply chains. *Computers in Industry* 72:82-91.

Kubler A, Zengler C, Kuchlin W (2010) Model counting in product configuration. *arXiv preprint arXiv* 1007.1024.

Nilsson NJ (1994) Probabilistic logic revisited. *Artificial intelligence* 59(1-2):39-42.

Ostrosi E, Fougres AJ, Ferney M (2012) Fuzzy agents for product configuration in collaborative and distributed design process. *Applied Soft Computing* 12(8):2091–2105.

Rodriguez B, Aydin G (2011) Assortment selection and pricing for configurable products under demand uncertainty. *European Journal of Operational Research* 210(3):635–646.

Roller D, Kreuz I (2003) Selecting and parameterising components using knowledge based configuration and a heuristic that learns and forgets. *Computer-Aided Design* 35(12):1085-1098.

Sanchez R, Mahoney JT (1996) Modularity, flexibility, and knowledge management in product and organization design. *Strategic management journal* 17(S2):63-76.

Siddique Z, Rosen DW, Wang N (1998, September) On the applicability of product variety design concepts to automotive platform commonality. In ASME Design Engineering Technical Conferences-Design Theory and Methodology.

Ulrich K (1994) Fundamentals of product modularity. *Springer Netherlands* 219-231.

Wahlstrom M (2008) A tighter bound for counting max-weight solutions to 2SAT instances. *In Parameterized and Exact Computation. Springer Berlin Heidelberg* 202-213.

Walker AJ, Bright G (2013) Stabilisation and control of configurable product manufacturing through Biased Decision Feedback decoupling. *Journal of Manufacturing Systems* 32(1):271-280.

Woehler C (2011) U.S. Patent No. 8,050,957. Washington, DC: U.S. Patent and Trademark Office.

Table A1: Abbreviations

ASR	The new format that we use in this paper to represent rules
CD	Conjunction of disjunctive clauses
DC	Disjunction of conjunctive clauses
FCR	Family cardinality rule
LAM	Large auto manufacturer
LHS	Left-hand-side
MWSAT	Maximum weighted satisfiability
OIR	Option implication rule
PSAT	Probabilistic satisfiability problem
PS	Penetration statistic
RHS	Right-hand-side
STD	Standard deviation
w.l.o.g.	Without loss of generality

Table A2: Notations

n	Number of options
N	Set of options $N = \{i \| i = 1, \cdots, n\}$
$p(.)$	Penetration rate or probability value
p_i	Penetration rate of option i, $\forall i \in N$
$\mathbf{p}$	A PS, $\mathbf{p} = (p_1, \cdots, p_n) \in \mathbb{R}^n$
$\mathbb{P}$	Set of all PSs that satisfy rules between options
λ	Projection scalar, $\lambda \geq 0$
$cone(\mathbb{P})$	The cone generated by $\mathbb{P}$, $cone(\mathbb{P}) = \{\lambda\mathbf{p} \| \mathbf{p} \in \mathbb{P}, \lambda \geq 0\}$
$\mathbf{q}_m$	The real sales data, $\forall m = 1, \cdots, M$
$f(.)$	A function that returns the sum of squared errors for a given PS
$\\|.\\|$	Euclidean norm
$\hat{\mathbf{p}}$	The forecasted PS
$\mathscr{P}$	The problem of finding the closest feasible PS to $\hat{\mathbf{p}}$, $\mathscr{P} = \min_{\mathbf{p}\in\mathbb{P},\lambda\geq 0} Dist(\lambda\mathbf{p}, \hat{\mathbf{p}})$
$Dist(.,.)$	A distance function, e.g. Euclidean norm
$\lambda^*, \mathbf{p}^*$	The optimal solution of $\mathscr{P}$
S	A subset of options
$\mathbb{S}$	Set of all subsets of options, $\mathbb{S} := \{S \| S \subseteq N\}$
$x(S)$	Variable for probability values, $x(S) \in [0, 1], \forall S \in \mathbb{S}$. The probability of intersection of some options is equivalent to sum of all $x(S)$'s where S includes those options
$V_{\mathbb{P}}$	Set of vertices of $\mathbb{P}$
$\mathbb{S}^0$	Defined as $\mathbb{S}^0 := \Big\{S \in \mathbb{S} : \Big(\exists \text{ rule } i \Leftarrow j_1 \wedge \cdots \wedge j_l \text{ s.t. } \{j_1, \cdots, j_l\} \subset S, i \notin S\Big) \vee \Big(\exists \text{ rule } i \Rightarrow j_1 \vee \cdots \vee j_l \text{ s.t. } i \in S, \{j_1, \cdots, j_l\} \cap S = \emptyset\Big)\Big\}$
$\mathbb{S}^1$	Defined as $\mathbb{S}^1 := \mathbb{S} \backslash \mathbb{S}^0$
$\mathbb{Y}$	Set of feasible configurations
y_i	Binary variable which is 1 if option i is chosen, and 0 otherwise, $\forall i \in N$
$\mathbf{y}$	A vector of n binary variables, $\mathbf{y} = (y_1, \cdots, y_n)$
α	A vector of coefficients to create convex combination $\sum_{i=1}^{k} \alpha_i^* \theta^i$
α^*	The optimal value of α, or vector of coefficients to create $\mathbf{p}^{*k}$, i.e. $\mathbf{p}^{*k} = \sum_{i=1}^{k} \alpha_i^* \theta^i$
ω	The dual variable associated with $\hat{\mathbf{p}}^T\theta = \hat{\mathbf{p}}^T\hat{\mathbf{p}}$ in $\mathcal{M}(\beta^k)$,$\omega \in \mathbb{R}$
$\mathbf{y}_{\text{rel}}$	The n-vector of continuous variables, or the relaxation of $\mathbf{y}$
$\mathbf{y}_\lambda$	The n-vector of continuous variables such that $\mathbf{y}_\lambda = \lambda\mathbf{y}$
$\mathbf{y}_{\lambda,\text{rel}}$	The n-vector of continuous variables in the cone generated by $conv(\text{SAT})$
$conv(.)$	The convex hull of a given set
$\mathcal{L}^k$	The lower bound on the feasibility gap at iteration k
$\mathcal{U}^k$	The upper bound on the possible improvement in the feasibility gap at iteration k
$\mathcal{LB}^k$	The best lower bound on the feasibility gap found till iteration k, i.e. $\mathcal{LB}^k = \max_{i=1,\cdots,k}\{\mathcal{L}^k\}$
δ	1 if the infeasibility of the $\hat{\mathbf{p}}$ has been reported in previous iterations, and 0 otherwise
$\mathbf{w}$	A weight vector to show the relative importance of options, $\mathbf{w} = (w_1, \cdots, w_n)$
k	Iteration counter, $k = 1, 2, \cdots$
$\mathbf{e}^i$	The vector of all zeros except for i's entry which is 1