Mass Customization and “Forecasting Options’ Penetration Rates Problem”

大规模定制及“预测每个选项的覆盖率问题”

Operations Research，2019

作者：

- Ali Fattahi，博士生， University of California，研究方向：大规模定制的零部件和产品组合，电力高峰负荷需求管理，高维统计。
- Sriram Dasu，助理教授， University of Southern California ，研究方向：医疗运营，全球健康，以客户心理为中心的服务运营，以及供应链管理。
- Reza Ahmadi，教授， University of California，研究方向：零件和产品设计，灰色市场的供应链，网络和小额信贷。

目录

1 问题来源

大型汽车制造商，通常允许客户自行配置汽车。一辆车包含多个模块：引擎、内部、悬挂装置…每个模块还有多种选项（options）。所以可组合的配置有非常多种，如Mercedes C-Class可以产生10^{21}种可行配置。

每种配置产生不同的BOM单。汽车制造商需要预测每种配置的销量，以制定生产计划、供应商合约、定价等。但配置太多，很难直接在配置层面预测销量。

所以大部分会选择在**模块选项（options）的层面**预测，称为**覆盖统计**（Penetration Statistics，PS），即预测每个模块选项的覆盖率，覆盖率**指使用这个选项的车占所有车的比例**。如，预测引擎其中一个选项（option）的覆盖率是0.2，就意味着认为卖出去的车中有20%会使用这种引擎。

如果覆盖率预测不准，可能导致库存过剩、短缺和客户满意度下降。

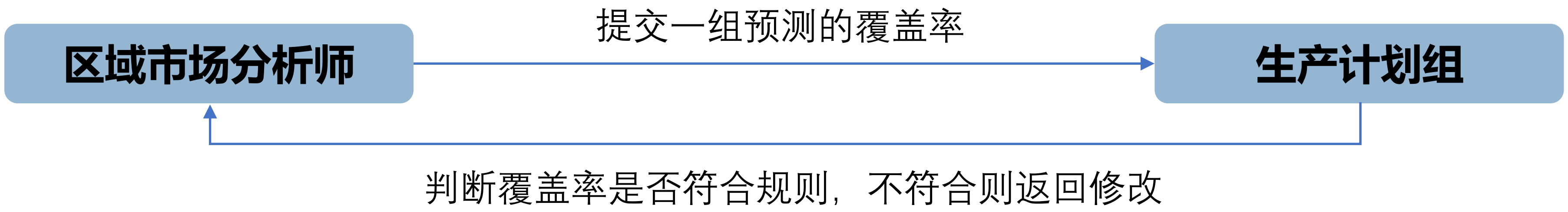
1 问题来源

选项之间的排列组合并不是全部都有效，还受到与设计、制造可行性等相关的**规则约束**。规则分为两类：

- 模块内规则（Family Cardinality Rules）：在这个模块的选项中，必须选择一种或者最多选择一种。
- 模块间规则（Option Implication Rules）：不同模块特定选项之间的约束关系。

做出的覆盖率预测，应当满足这些规则约束，否则就是不可行的预测。

而这是很多汽车制造商面临的难题：**销售部门给出的覆盖率预测是否符合这些规则，如果不符合，如何修正。**



1 问题来源

一个简化的例子

Figure 1. (Color online) A Simplified Version of the Features and Specifications of the 2016 Hyundai Tucson

2016 HYUNDAI TUCSON
FEATURES & SPECIFICATIONS

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

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

车型 2016 Hyundai Tucson

有三种配置：SE（高配）、ECO（标配）、SPORT/Limited（低配）

可配置**三个模块**：引擎、变速器、轮胎

每个模块有两种选项，所以共有**六种选项**：

引擎1、引擎2、变速器1、变速器2、轮胎1、轮胎2。

规则：

- 3个模块内规则：引擎{1,2}中必须选一个，变速器和轮胎同理
- 2个模块间规则：引擎1⇔变速器1，引擎1⇒轮胎1。

六种选项任意排列组合有2^6种配置，但符合上述规则的有效配置只有3种：

高配{引擎1，变速器1，轮胎1}；标配{引擎2，变速器2，轮胎1}；低配{引擎2，变速器2，轮胎2}

1 问题来源

一个简化的例子

Figure 1. (Color online) A Simplified Version of the Features and Specifications of the 2016 Hyundai Tucson

2016 HYUNDAI TUCSON
FEATURES & SPECIFICATIONS

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

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

现在给出一种覆盖率预测：

引擎1=0.6；（卖出的车中有60%会使用引擎1）

引擎2=0.4；

变速器1=0.6；

变速器2=0.4；

轮胎1=0.3；

轮胎2=0.7。

符合模块内规则

不符合模块间规则

2个模块间规则：引擎1⇔变速器1，引擎1⇒轮胎1

用引擎1的，一定也用轮胎1；用轮胎1的，不一定用引擎1。所以轮胎1的覆盖率应当高于引擎1。

1 问题来源

一个简化的例子

Table 1. Unused Inventory and Lost Sales as a Result of Infeasible Forecasting (Total Sale = 1,000)

		ENG1	ENG2	TRN1	TRN2	WHL1	WHL2
预测的覆盖率	Forecast PS	0.6	0.4	0.6	0.4	0.3	0.7
按覆盖率备货	Inventory	600	400	600	400	300	700
最大使用数量	Maximum usable quantities	300	400	300	400	300	400
剩余存货	Unused quantities	300	0	300	0	0	300
失去的收入	Shortage (lost sale)	1, 000 – (300 + 400) = 300					

假设2016年预计该型号车的销量是1000台，按照覆盖率备货。

- 虽然存货有600个引擎1，但最多只能卖出300个（因为只有300个轮胎1和它配套），剩下300个积压。
- 按这个存货，只够生产300台高配和400台低配，距离预计的1000台还有300台的差距，是流失收入。

检验一组覆盖率预测是否可行，如果不可行要找到一组可行的替代，对于汽车制造商很重要。

实际中，LAM公司的每辆车，有约400种选项和4000条规则，依靠生产计划组人工判断预测覆盖率是否违反规则，如果违反，返回区域销售处修改。**耗时，不能在不可行时快速找到替代。**

2 研究内容

研究内容：建模，求解。

模型输入：规则，一组预测覆盖率 $\widehat{p}$

模型输出：这组覆盖率预测是否可行，不可行给出一组最近替代 p

目标函数：最小化 p与输入预测覆盖率 $\widehat{p}$的欧式距离

模型核心是**概率可满足性问题（probabilistic satisfiability problem，PSAT）**，这类问题有三个要素：

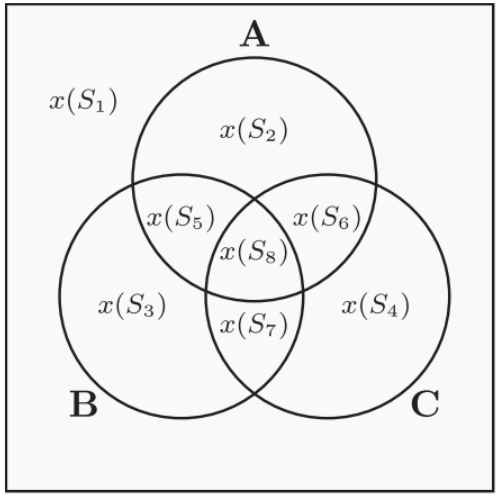
- m个逻辑变量
- 基本式，一个基本式是一个逻辑变量或者它的非
- n个子句，每个子句由逻辑“或” 连接基本式组成（规则）

目的：确定是否存在一组变量的逻辑取值使得所有基本句都为真。

P问题：多项式时间可解；**NP问题**：先给一个答案，多项式时间内可验证这个答案对不对；**NP难问题**：任意一个NP问题都可以多项式规约成该问题；**NP完全问题**：既是NP问题，又是NP难问题。

2 研究内容

Figure 2. Graphical Illustration: Three Options, Eight Subsets, and Probabilities of Intersections



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

Note. For example, $x_2 := x(S_2)$.

例：3个选项，划分出8个互斥但覆盖全集的子集S1-S8，每个子集被选择的概率是x1-x8。此时可行域（所有可行覆盖率的凸集）P，就由至少8个变量和约束定义。

当有n个选项时，变量和约束数量超过2^n。

2 研究内容

求解PSAT问题的方法

1.列生成算法

- 先把原问题修改到一个规模更小（即变量数比原问题少的）的问题，新问题用单纯形法求最优解。
- 再通过一个子问题去确认在那些未被考虑的变量中是否有使得reduced cost小于零的，如果有，那么就把这个变量的相关系数列加入到上一步的系数矩阵中，重复第一步。
- 经过反复的迭代，直到子问题中的reduced cost rate大于等于零，那么原问题就求到了最优解。

例：可以求解200个变量、800个子句的PSAT问题。　缺点：迭代缓慢、理论上不知道收敛速度。

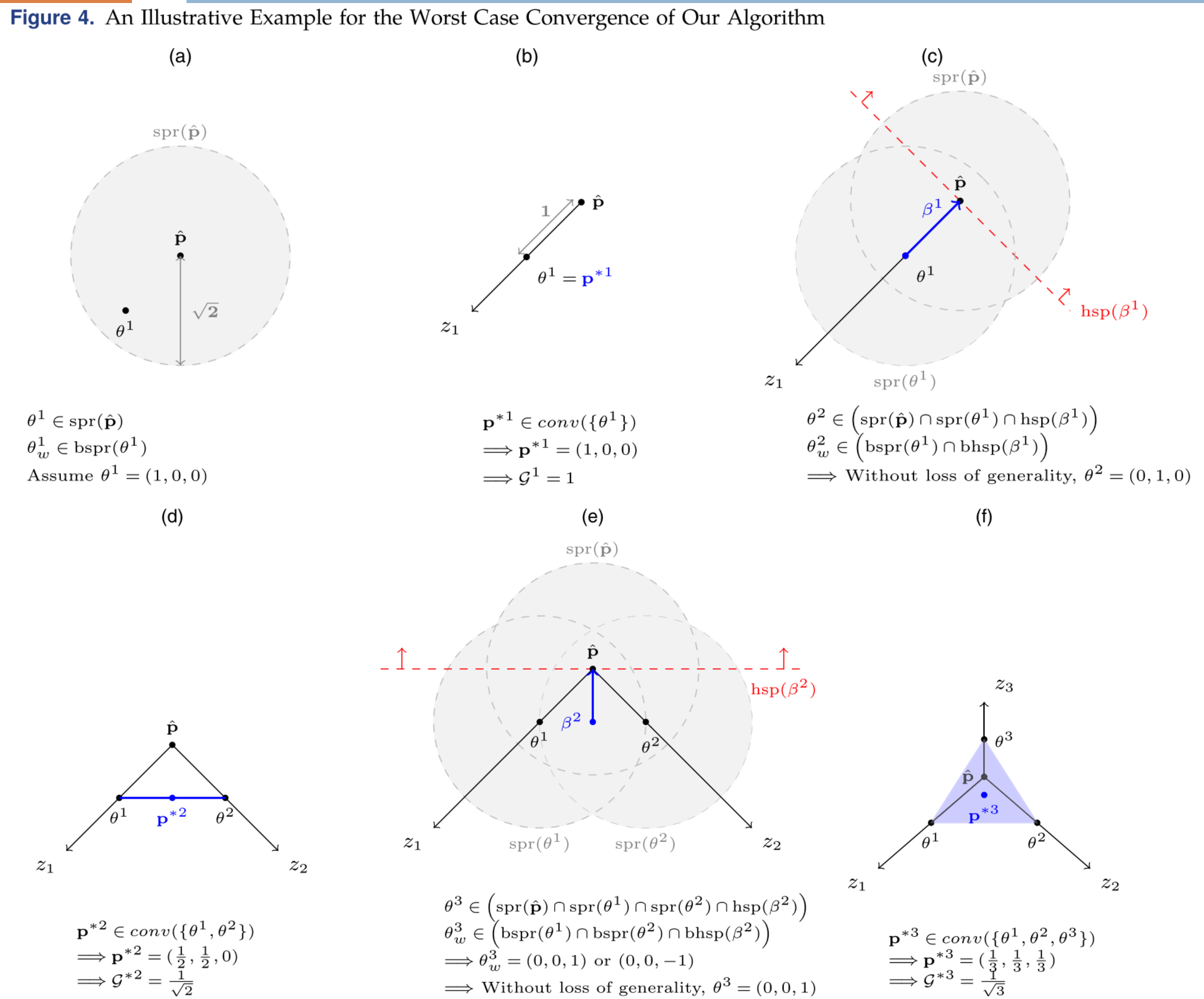
2.Frank-Wolfe(FW)算法

- 把求解非线性最优化问题转化为求解一系列线性规划问题，而且各线性规划具有相同的约束条件。
- 基本思想是将目标函数作线性近似，通过求解线性规划求得可行下降方向,并沿该方向在可行域内作一维搜索。

可以从理论上算出，迭代k次后，和最优解的距离下降到 $\mathscr{O}\left(1/\sqrt{k}\right)$

例：400个选项，4000条规则，在PC上可以在5000s内得到1%误差的可行解。

2 研究内容



演示一个Frank-Wolfe(FW)算法3维情况的收敛过程。

假设输入的$\hat{p}$是可行解。

目标：最小化与 $\hat{p}$ 的距离

每次迭代有两个关键步骤

- **方向步**，找到向量β（朝向 $\hat{p}$ ）
- **最大化步**，最大化βθ，找到最大的θ

2 研究内容

Algorithm 1

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 $\hat{\mathbf{p}}$.

1. $\theta^1 :=$ an arbitrary point in $\mathcal{FH}_{\hat{\mathbf{p}}}$; ▷ Assume $\mathcal{FH}_{\hat{\mathbf{p}}} \neq \{\}$.
2. $\delta := 0$; ▷ $\delta = 1$ means that it is known $\hat{\mathbf{p}} \notin cone(\mathbb{P})$, and $\delta = 0$ means otherwise!
3. **For** $k = 1, 2, 3, \cdots$ **do**
4. $\mathbf{p}^{*k} := \sum_{i=1}^{k} \alpha_i^* \theta^i$, where α^* is obtained by solving Equations (9)–(11);
5. $\beta^k := \hat{\mathbf{p}} - \mathbf{p}^{*k}$; $\mathcal{G}^k := \|\beta^k\|$;
6. **If** $\mathcal{G}^k = 0$, **then**
7. Report "$\hat{\mathbf{p}} \in cone(\mathbb{P})$"; stop!
8. **End if**
9. Solve $\mathcal{M}(\beta^k)$; $\theta^{k+1} :=$ the optimal value of θ;
10. $\mathcal{U}^k := \min\left\{\mathcal{G}^k, \frac{\beta^{kT}(\theta^{k+1} - \mathbf{p}^{*k})}{\mathcal{G}^k}\right\}$; $\mathcal{L}^k := \mathcal{G}^k - \mathcal{U}^k$;
11. **If** $\mathcal{L}^k > 0$ **and** $\delta = 0$, **then**
12. Report "$\hat{\mathbf{p}} \notin cone(\mathbb{P})$"; ▷ continue to find the nearest $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$!
13. $\delta = 1$; ▷ to prevent reporting "$\hat{\mathbf{p}} \notin cone(\mathbb{P})$" in next iterations!
14. **End if**
15. **If** $\mathcal{U}^k = 0$, **then**
16. Report "$\mathbf{p}^{*k}$ is the nearest $\mathbf{p} \in \mathcal{FH}_{\hat{\mathbf{p}}}$ to $\hat{\mathbf{p}}$"; stop!
17. **End if**
18. **End for**

模型输入：规则，一组预测覆盖率p(hat)

模型输出： p(hat)是否可行，如不可行，在可行域中找出离p(hat)最近的替代p

1：在可行域中任意找一个点，作为起始

2：一个指示变量，为1表示p(hat)不可行

3：开始第k次迭代

4-5：从起始点出发，计算**方向步**，找到下一个更近的点p(k)，计算这个点和p(hat)的欧式距离

6-8：如果距离为0，那么p(hat)可行，程序结束。

9：计算**最大化步**，这个问题NP难，用启发式算法，并且可以设置允许相对最优误差

10：计算第k次迭代的距离的下界，和距离可能改进量的上界

11-14：如果下界大于0， 说明p(hat)不是可行解

15-18：如果上界为0，说明当前已经迭代到了离p(hat)最近的可行解p(k)，程序结束

3 问题解决效果

LAM公司实例

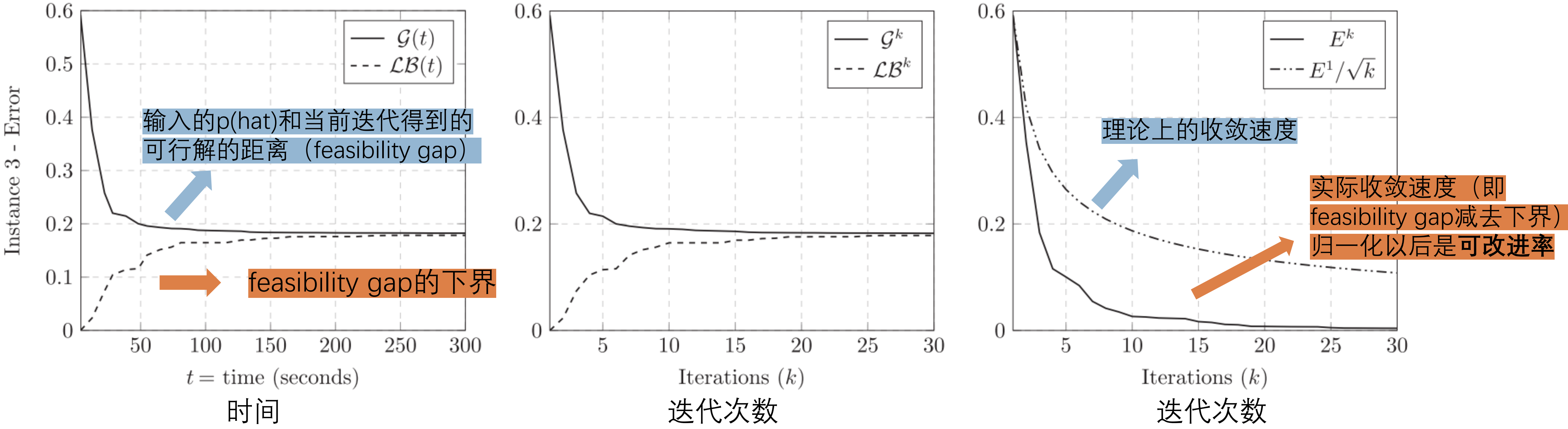
LAM：每隔几个月预测一次覆盖率，提前三年。

即2019年发售的车在2016年就要预测。

三个例子（车型、发行地区不同，所以选项规则都不同）

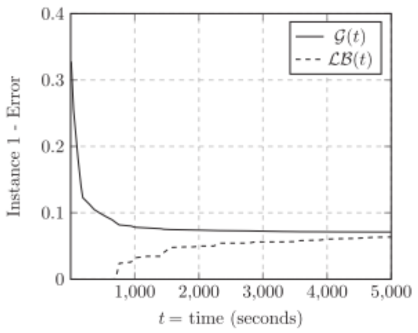
Table 3. The Specifications of Our Industrial Instances from the LAM

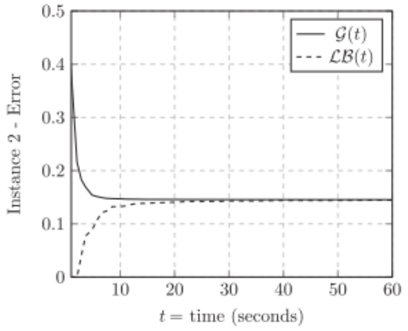
	Options	OIRs	FCRs
Instance 1	415	3,703	85
Instance 2	200	428	72
Instance 3	395	2,111	97

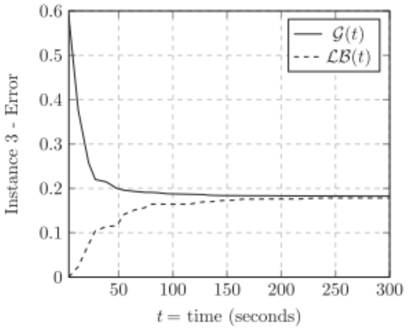


3 问题解决效果

LAM公司实例







1.**分别经过5000s,60s和300s的运行后，得到可改进率在1%以内的可行解。**

2.开始时可改进率很大，如实例1和实例2达到30%，实例3达到60%。分别经过40次、10次、10次迭代后，下降到1%。

3.三个实例中输入的一组覆盖率预测值都不可行，在分别迭代12、3、2次后就可以判断出来。

注：可改进率指当前迭代得到的可行解和输入的p(hat)之间的欧式距离/该距离的下界，归一化以后的值。

整体而言，三个实例中，算法表现基本一致，可改进率下降很快，**可以在可接受时间内找到满意解。**

3 问题解决效果

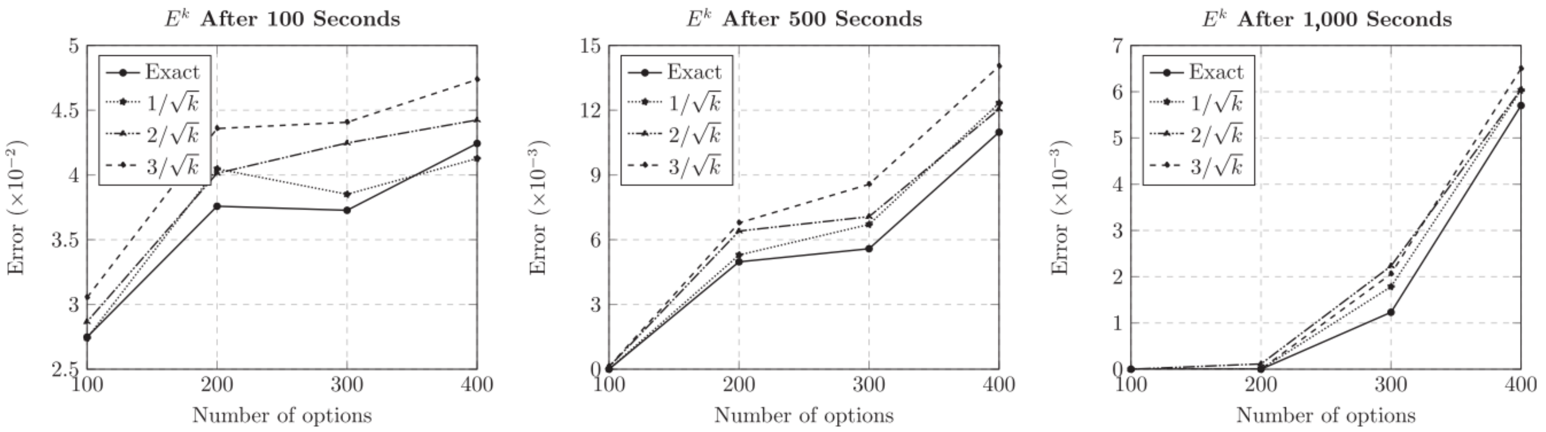
两组敏感性分析：

（一）规则都为4056条，选项数从100-400，启发式算法中允许的误差水平有4种。

（二）选项数都为410，规则数从1000-4000，启发式算法中允许的误差水平有4种。

$$\frac{\varphi}{\sqrt{k}}\quad(\varphi=0,1,2,3)$$

Figure 5. Normalized E^k for Different Numbers of Options After 100, 500, and 1,000 Seconds

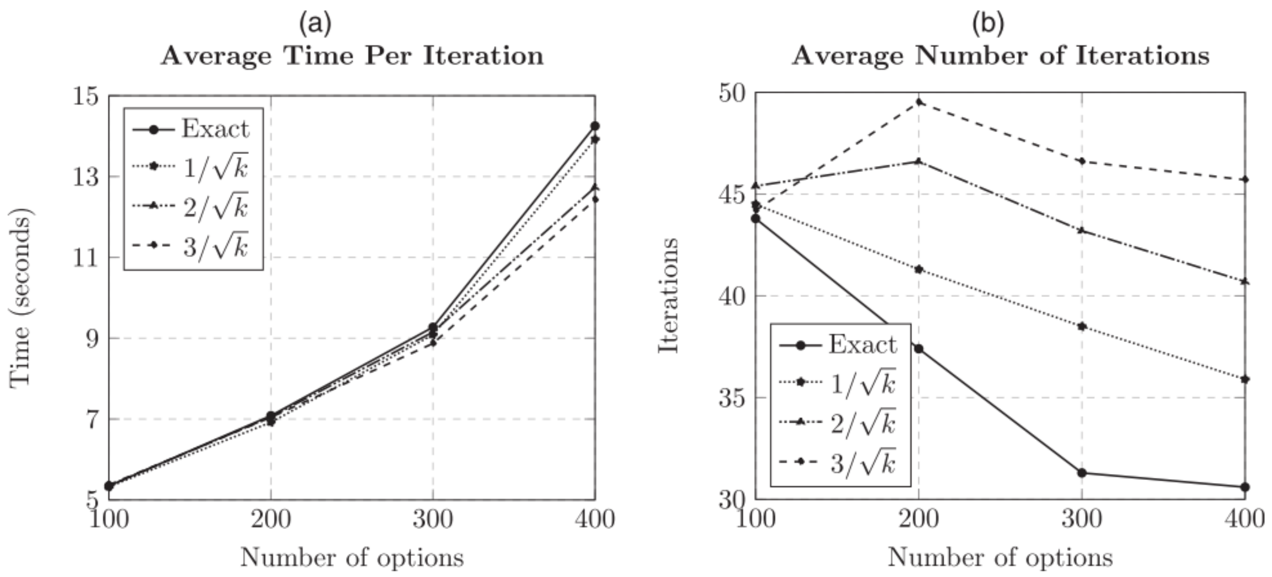


1.运行1000s以后，100选项数的可改进率已经下降到接近0，其他情况也已经很小。

2.启发式算法中不允许误差时，表现更好。

3 问题解决效果

Figure 6. (a) Average Running Time per Iteration and (b) Average Number of Iterations



敏感性分析结论

3.随着选项数的增加，平均每次迭代所需的时间在上升。主要是因为启发式算法的求解时间上升。

4.随着选项数的增加，算法表现逐渐下降。

4 启示

1.这篇文章中，LAM公司是业务部门先预估出一组覆盖率（没有指出使用的具体方法），再判断这组覆盖率是否符合现有规则。是否可以把“判断是否符合规则”这部分工作与业务部门的预测结合起来。

2.适当放弃对“最优解”的追求，牺牲精度换取速度。