

# 0~1规划模型在信用评级卡组合优化问题中的应用

刘显鹤, 鲁建辉, 白雪健

沈阳航空航天大学, 理学院, 辽宁 沈阳

收稿日期: 2023年7月16日; 录用日期: 2023年8月6日; 发布日期: 2023年8月16日

---

## 摘要

针对信用评级卡组合优化问题, 首先运用穷举法解出原问题, 之后参照二次无约束二值优化模型与0~1规划模型, 将信用评级卡组合问题转化成整数规划模型, 利用python程序进行模拟求解不同组合不同阈值下的最终收益, 并对结果进行讨论分析, 得出最优的信用评级卡组合。

## 关键词

二次无约束二值优化模型, 0~1规划, 整数规划

---

# The Application of 0~1 Programming Model of Credit Scorecard Combinatorial Optimization Problem

Xianhe Liu, Jianhui Lu, Xuejian Bai

College of Science, Shenyang Aerospace University, Shenyang Liaoning

Received: Jul. 16<sup>th</sup>, 2023; accepted: Aug. 6<sup>th</sup>, 2023; published: Aug. 16<sup>th</sup>, 2023

---

## Abstract

For the credit scorecard combinatorial optimization problem, first using the exhaustive method to solve the original problem, and then refer to the quadratic unconstrained binary optimization model and 0~1 programming model to convert the credit scorecard combination problem into an integer programming model, and using the python program carry out simulation to solve the final income of different combinations and different thresholds, and discussing and analyzing the results to obtain the optimal combination of credit scorecards.

## Keywords

Quadratic Unconstrained Binary Optimization Model, 0~1 Programming, Integer Programming

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

在银行信用卡或相关的贷款等业务中，需要先通过各种审核规则对客户的信用等级进行评定，通过评定后的客户才能获得信用或贷款资格。银行需要根据各种审核规则对客户的信用等级进行评定，这些规则就称为信用评分卡，每个信用评分卡又有多种阈值设置，这就使得不同的信用评分卡在不同的阈值下，对应不同的通过率和坏账率。因此，银行的目标是选择最合理的信用评分卡组合以及其阈值，使得银行最终收入最多。规则审核过程实际是经过一重或者多重组合规则后对客户进行打分，这些规则就被称为信用评分卡，每个信用评分卡又有多种阈值设置(但且只有一个阈值生效)，这就使得不同的信用评分卡在不同的阈值下，对应不同的通过率和坏账率，因为信用评分卡在现实中有广泛的作用，所以有越来越多的人来对此问题进行研究，如吴佳辉[1]利用特征处理的方法建立的评分模型，但是该模型只适用于特征比较少的情况；谢瑞[2]利用逻辑回归、随机森林等方法建立了评分模型，但该模型计算复杂度较高；高嘉，任亚明[3]利用模拟退火算法来建立优化模型，但该模型因遵守模拟退火算法的核心 Metropolis 准则导致在迭代末期，结果的出现的概率低，精确性不高；而国外的一些学者，如 M.H.M.A. Jahromi [4]等人利用 0~1 混合整数目标规划模型来解决机床选择和操作分配问题，但该模型方法非精确，不能得到全局最优解；Özgür Ülker [5]等人利用 0~1 规划模型来解决办公室空间分配问题，但该模型约束条件过多且计算复杂；Khaled Alhamad [6]等人利用非线性整数规划来建立模型解决预防性维护计划中的设备维护及水电分配问题，该模型以 0~1 规划为主，模型结果较优，符合实际情况，所以本文也利用了 0~1 规划模型进行建模。

我们由简单到复杂进行研究，我们先研究在我们所选的 100 张评分卡中选出一张卡的一个阈值，使我们最终受益最大的问题。对此问题，采用穷举法将附件中的 100 张评分卡的每一个阈值的数据进行代入计算，找到最终收益最大的一张卡的一个阈值。

进行讨论分析，之后，我们研究选定的 3 张评分卡的各自 10 个阈值进行组合的问题，最后我们考虑在 100 张信用评分卡中选出 3 张信用评分卡来进行组合，得到最大收益的问题。

## 2. 符号说明

符号说明见表 1。

**Table 1.** Symbol description

**表 1.** 符号说明

序号	符号	符号说明
1	$t$	通过率
2	$h$	坏账率
3	$t = t_1 \cdot t_2 \cdots t_n$	总通过率

Continued

4	$h = \frac{1}{n}(h_1 + h_2 + \dots + h_n)$	总坏账率
5	$w$	资金
6	$s$	利息收入率
7	$z$	收入

### 3. 最优信用评分卡组合问题研究

#### 3.1. 一张评分卡的一个阈值的最大收益问题

表 2 为本文研究所运用的部分数据。

**Table 2.** Part of the scorecard data of 100 credit scorecards

**表 2.** 100 张信用评分卡中部分评分卡数据

阈值	卡 1 通过率	卡 1 坏账率	卡 2 通过率	卡 2 坏账率	卡 3 通过率	卡 3 坏账率	卡 4 通过率	卡 4 坏账率
1	0.76	0.013	0.72	0.032	0.8	0.012	0.79	0.004
2	0.77	0.015	0.73	0.038	0.82	0.013	0.8	0.007
3	0.78	0.017	0.76	0.05	0.83	0.024	0.81	0.011
4	0.8	0.024	0.77	0.053	0.86	0.04	0.84	0.017
5	0.82	0.026	0.79	0.065	0.89	0.042	0.85	0.028
6	0.84	0.028	0.82	0.074	0.92	0.057	0.87	0.04
7	0.87	0.03	0.86	0.076	0.93	0.068	0.88	0.046
8	0.93	0.036	0.87	0.079	0.94	0.069	0.9	0.051
9	0.94	0.043	0.91	0.08	0.97	0.07	0.92	0.052
10	0.96	0.052	0.92	0.084	0.98	0.073	0.93	0.059

在 100 个信用评分卡中找出 1 张及其对应阈值，使最终收入最多，所以我们只需要将每个阈值的通过率与坏账率带入到所建立的最终收益模型中，找到最优解，我们将其作为已知数据，作为 0~1 规划模型的验证，之后我们将 1 张信用评分卡的不同阈值最终收益模型转化为 0~1 规划模型，之后利用 python 程序进行求解：

首先，我们利用 matlab 进行穷举，先写出同一张信用评分卡的不同阈值的最终收益的模型：

$$z = w \cdot t \cdot (s - s \cdot h - h)$$

我们先假设利息收入率为 5%，要贷款 10,000 元，这时我们的模型中，不知道的未知量就只剩两个了，一个是通过率  $t$ ，另一个是坏账率  $h$ ，按照每一张评分卡的不同阈值对应的不同的通过率和坏账率，我们在 matlab 中利用 for 循环将每一个阈值中的通过率和坏账率输入进去，算出最终收益，并将其比较，得到最高收益为卡 49 的阈值 1。

但是这只是针对利息收入率为 5% 的收益模型的最优值，于是我们改变利息收入率，代回这个模型中，得出利息收入率在 1%~10% 之间的得到最优收益时所选择的信用评分卡的序号，如图 1 所示。

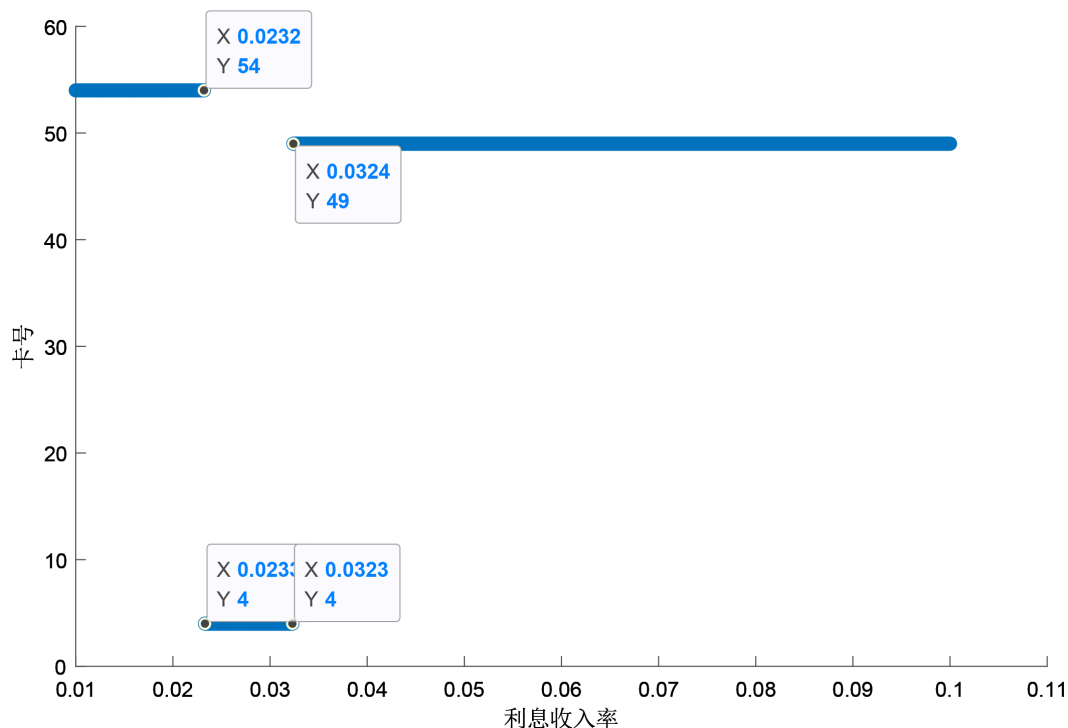


Figure 1. Optimal scorecard sequence numbers at different interest rates

图 1. 不同利率下的最优评分卡序号

由图 1 可知:

$$y = \begin{cases} 54 & 0 < x \leq 0.0232 \\ 4 & 0.0232 < x \leq 0.0323 \\ 49 & x > 0.0323 \end{cases}$$

其中  $x$  是银行贷款利息收入率,  $y$  表示信用评分卡的编号, 例如在银行贷款利息收入率小于 2.32% 的时候, 银行选第 54 张信用评分卡的时候收益最高, 当银行贷款利息收入率大于 2.32% 小于 3.23% 的时候, 银行选择第 4 张信用评分卡的时候收益最高, 当银行贷款利息收入率大于 3.23% 的时候, 银行选择第 49 张信用评分卡的时候收益最高。

之后我们将这个一张信用评分卡的最终收益的模型转换成 0~1 规划模型。

首先我们需要将所建立的模型符合自变量只能取得 0 或 1 的限制, 所以我们把是否选择这张信用评分卡的某个阈值作为自变量, 选择该阈值, 则变量值为 1, 否则值为 0, 这样我们就可以把模型变成如下形式:

$$\begin{aligned} \max f(x) &= \sum_{i=1}^{10} x_i z_i \\ \text{s.t. } x_i &\in \{0, 1\}, \sum_{i=1}^{10} x_i = 1 \end{aligned}$$

此时, 这种形式还没有符合 0~1 规划模型, 因为 0~1 规划模型中可以转换成  $x^T A x$  的形式, 找到矩阵  $A$  后可以代入计算, 所以, 我们为了得到  $Ax^2$ , 我们需要将这个式子变为二次多项式, 我们令  $x_i = x_i^2$ , 这样我们的模型就变成了二次模型, 就可以利用 0~1 规划模型的求解算法进行求解, 此时的  $A$  有一个特点因为本题中的每个阈值之间并无关系, 所以此时的矩阵  $A$  没有自变量的交叉项(例如  $Cx_1x_2$ ,  $C$  为常数),

只是一个对角矩阵，只有对角线上有元素，形式如下：

$$A = \begin{bmatrix} z_1 & & & \\ & z_2 & & \\ & & \ddots & \\ & & & z_{10} \end{bmatrix}$$

则利用 python 的 Pulp 库可以计算 0~1 规划模型的最优值，numpy 库可以使对  $n$  维数组的处理更加便利，这样可以求解出得到最优解时，所选择的信用评分卡以及其对应的阈值。

经验证，模型得到的结果与穷举法得到的结果相同。

### 3.2. 固定三张评分卡的阈值组合的最大收益问题

在已经选定的 3 张信用评分卡中选择相应的阈值组合，使最终收入最多，让我们建立 0~1 规划模型并求解：

我们将 3 张评分卡的各自 10 个阈值进行组合，利用 matlab 穷举法程序，将数据代入我们所建立 3 张评分卡阈值组合的最终收益模型中进行计算，选出最终收益最大的阈值组合，并将此模型转换成 0~1 规划模型，再利用 python 程序进行求解。

首先我们利用 matlab 进行穷举，然后将三个信用评分卡阈值对应的通过率相乘、坏账率相加，放入 matlab 程序进行穷举：

假设贷款利息收入率为 5%，贷款金额为 10,000 元，然后对选定的三张信用评分卡的各个阈值进行排列组合，通过将不同评分卡阈值对应通过率相乘得到总通过率，将坏账率相加得到总坏账率，我们在 matlab 中利用 for 循环将每一组阈值获得的总通过率和总坏账率输入进去，计算比较出最高收益及此时的评分卡阈值，通过使用 matlab 来遍历这 1000 组数据，找出其中收益最大的阈值组合，并且求出其最大值。

之后我们将这个一组信用评分卡的最终收益的模型转换成 0~1 规划模型。

首先我们需要将所建立的模型符合自变量只能取得 0 或 1 的限制，所以我们把是否选择这张信用评分卡的某个阈值作为自变量，选择该阈值，则变量值为 1，否则值为 0，这样我们就可以把模型变成如下形式：

$$\begin{aligned} \max f(x) &= \sum_{i=0}^{999} x_i z_i \\ \text{s.t. } x_i &\in \{0,1\}, \sum_{i=0}^{999} x_i = 3 \end{aligned}$$

我们再将这个式子变为二次模型，就可以利用 0~1 规划模型的求解算法进行求解，而固定三张评分卡的三个阈值组合的最大收益问题中的每个阈值之间仍并无关系，所以此时的矩阵  $A$  也没有自变量的交叉项，只是一个对角矩阵，只有对角线上有元素：

$$A = \begin{bmatrix} z_0 & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_{999} \end{bmatrix}$$

则利用 python 的 Pulp 库可以解决 0~1 规划问题，求解出得到最优解时，所选择的信用评分卡以及其对应的阈值。我们通过将选定三张信用评分卡的不同阈值进行重新组合，然后我们使用了一种新型定位

方式来确定使用的信用评分卡对应的阈值，我们首先固定前两个信用评分卡的阈值，然后通过改变第三个信用评分卡的阈值来进行逐次遍历，即 1000 种组合，我们将其记为 000 到 999。

经验证，模型得到的结果与穷举法得到的结果相同。

### 3.3. 三张评分卡的阈值组合的最大收益问题

从 100 张信用评分卡中任选取三张信用评分卡，并选择阈值组合，使得最终收入最多，让我们建立 0~1 规划模型并求解。

我们在固定三张评分卡的三个阈值组合的最大收益问题中的收益模型是依旧可以适用于此问题的，我们只需要利用程序将 100 个信用评分卡取 3 张的不同的阈值组合，并使用收益模型比较即可，然后利用 python 程序进行求解，得出最优解。

首先，我们还是利用 matlab 穷举法，将所有可能出现的 3 张卡的组合中 3 个不同阈值的组合进行穷举计算，比较出最优的阈值组合，而此题照比固定三张评分卡的三个阈值组合的最大收益问题中复杂的地方在于穷举时需要取 3 张信用评分卡作为组合，之后每一个组合进行 10 乘 10 乘 10 次共 1000 次运算，所以穷举法计算时间较长，结果整理如表 3 所示。

**Table 3.** The result of matlab exhaustive method

**表 3.** matlab 穷举法的结果

最终收入	244.4941
卡号 1	8
卡号 1 阈值	2
卡号 2	49
卡号 2 阈值	2
卡号 3	73
卡号 3 阈值	2
运行时间	2554.04 秒

之后，我们本应该应用 python 程序在 100 张卡里取 3 张，并在 3 张卡中运用第二题的模型，相当于是利用三次循环，取出其中的通过率和坏账率，之后利用：

$$t = t_1 \cdot t_2 \cdots t_n$$

$$h = \frac{1}{n}(h_1 + h_2 + \cdots + h_n)$$

将此时的总通过率和总坏账率算出来，并放入第二题所建立的三张评分卡的不同阈值收益模型中计算收益。

但是，根据问题一的收益模型：

$$z = w \cdot t \cdot (s - s \cdot h - h)$$

我们固定了贷款金额  $w = 1,000,000$ ，利息收益率  $s = 5\%$ ，由于要使  $z$  最大，将  $w$  与  $s$  代入，可得  $0.05t(1 - 21h)$  最大时， $z$  最大，那么我们推广到本题中的三个信用评分卡组合问题，这时

$$t = t_1 \cdot t_2 \cdot t_3$$

$$h = \frac{1}{3}(h_1 + h_2 + h_3)$$

我们在取一个表中最优值的时候,为了近似替代,将 $t_1, t_2, t_3$ 和 $h_1, h_2, h_3$ 均代替为此时阈值的 $t$ 和 $h$ ,只要求出每一张表中的能使 $0.05t(1-21h)$ 最大时的阈值即可,这样我们的每个表中只剩一个阈值,之后再根据2.2的模型将问题转化成0~1规划,之后利用类似于固定三张评分卡的三个阈值组合的最大收益问题中的python程序进行求解进行找到三个表中的三个阈值的组合即为最优解,结果整理如表4所示。

**Table 4.** Applying python to solve the maximum benefit problem of three scorecards  
**表 4.** 应用 python 求解三张评分卡组合的最大收益问题

最终收入	244.4941
卡号 1	8
卡号 1 阈值	2
卡号 2	49
卡号 2 阈值	2
卡号 3	73
卡号 3 阈值	2
运行时间	0.2220 秒

经验证,穷举法与模型求得结果均为卡 8 阈值 2、卡 49 阈值 2、卡 73 阈值 2,结果相同。

**Table 5.** The scorecard data of 3 credit scorecards  
**表 5.** 3 张信用评分卡数据

阈值	卡 8 通过率	卡 8 坏账率	卡 49 通过率	卡 49 坏账率	卡 73 通过率	卡 73 坏账率
1	0.79	0.01	0.82	0.005	0.82	0.008
2	0.88	0.013	0.83	0.006	0.84	0.01
3	0.89	0.016	0.85	0.01	0.85	0.017
4	0.9	0.017	0.86	0.019	0.89	0.027
5	0.93	0.032	0.89	0.021	0.9	0.044
6	0.94	0.036	0.9	0.041	0.91	0.045
7	0.96	0.039	0.92	0.042	0.92	0.047
8	0.97	0.047	0.93	0.044	0.94	0.051
9	0.98	0.069	0.94	0.049	0.97	0.054
10	0.99	0.07	0.95	0.052	0.99	0.068

我们列出卡 8、49、73 在不同阈值下的通过率和坏账率数据,如上表 5 所示,与其他卡对比发现,卡 8、49、73 的通过率高于大部分其他卡,坏账率低于大部分其他卡,如图 2、图 3 所示。我们再应用 3.2 的模型进行验证,得到卡 8 阈值 2、卡 49 阈值 2、卡 73 阈值 2 为最优组合,根据实验,我们建立的模型相对于穷举法极大的提高了运行速度,且实验结果合理,符合实际情况且与实际结果相同。综上,最优信用评分卡组合为卡 8 阈值 2、卡 49 阈值 2、卡 73 阈值 2。

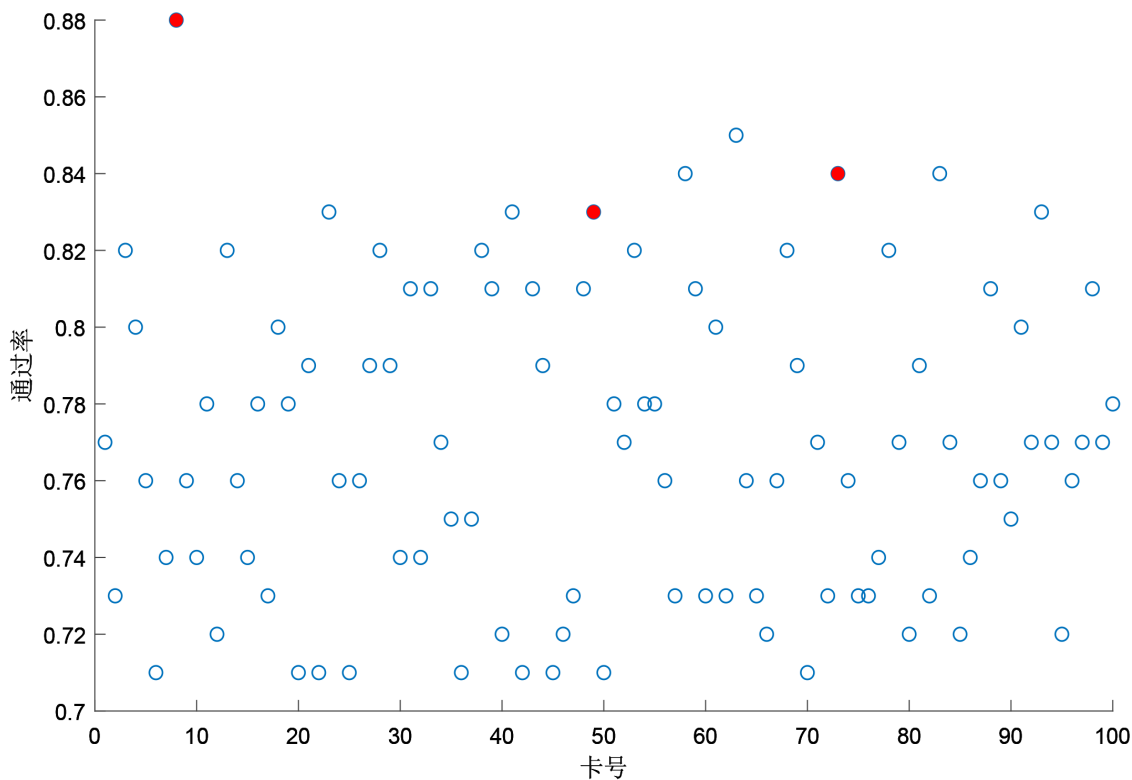


Figure 2. Comparison chart pass rates of all Scorecards when the threshold is 2  
图 2. 阈值为 2 时所有评分卡通过率对比图

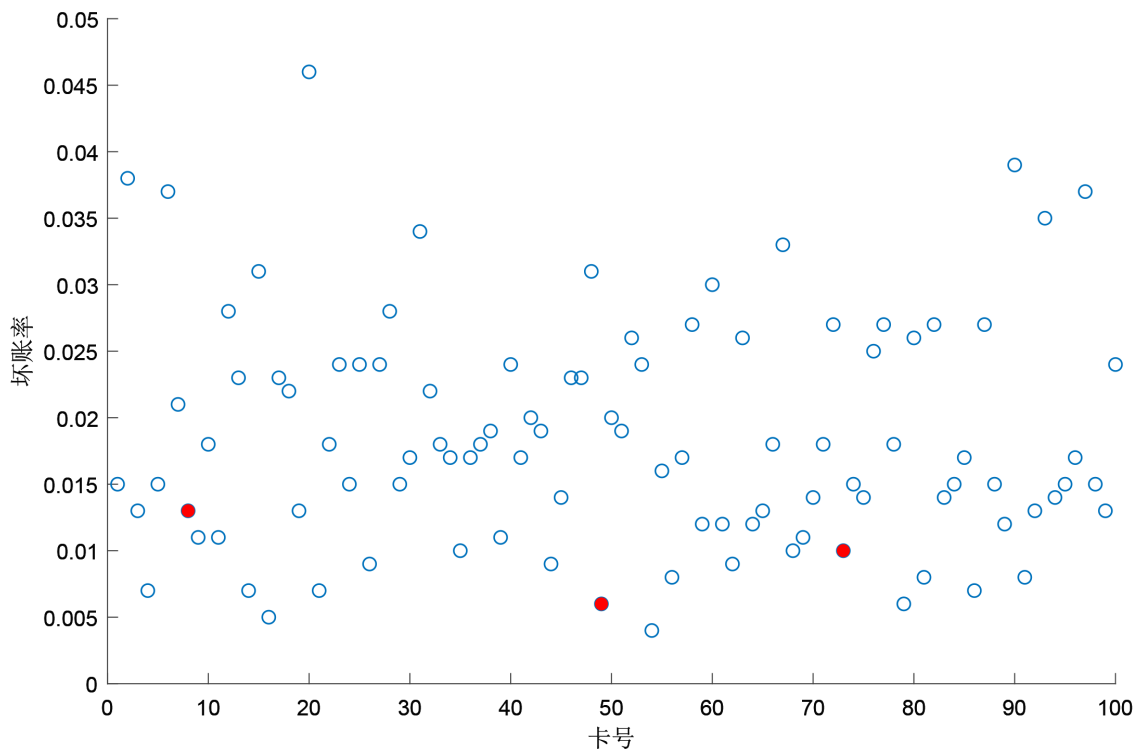


Figure 3. Comparison chart of bad debt rates for all Scorecards when the threshold is 2  
图 3. 阈值为 2 时所有评分卡坏账率对比图



## 4. 模型评估

### 4.1. 模型的优缺点

模型的优点:

- 1、问题的难度由浅入深，建模思路也循序渐进，便于理解。
- 2、深入考虑了在利息改变情况下，最高收益的阈值组合会发生变化的情况，对于其他关于利息问题的研究具有一定的参考意义。
- 3、每一步得到的结果都与穷举法的精确结果作对照，经检验得到的结果相同，模型的用时更短。
- 4、模型的求解采用 matlab 软件与 Python 软件，得到的结果可信度较高。

模型的缺点:

- 1、现实生活中的银行利息大约在 0.3%到 5%之间，本题题目所设置的利息为 5%，所以本文中的最优信用评分卡也会根据利息调整而不同。
- 2、现实生活中影响一个人的信用评分因素会更多，照比现实还有一定差距。

### 4.2. 模型的推广与改进

在问题分析中除了考虑现有的数据，还应该考虑其他与信用相关的数据，这样反映出的结果更合理，也更适合应用于实际。在模型建立时还可以使用最近提出的 QUBO 模型等新模型进行优化，从而降低程序运行的时长。

## 参考文献

- [1] 吴佳辉. 基于电信运营商大数据的用户信用评分卡的设计与实现[D]: [硕士学位论文]. 北京: 北京工业大学, 2019. <https://doi.org/10.26935/d.cnki.gbjgu.2019.000258>
- [2] 谢瑞. 基于逻辑回归的个人信用评分卡设计与分析[D]: [硕士学位论文]. 北京: 北京交通大学, 2022. <https://doi.org/10.26944/d.cnki.gbfju.2022.001451>
- [3] 高嘉, 任亚明. 基于模拟退火算法组合优化问题的求解[J]. 企业科技与发展, 2021, 475(5): 66-68.
- [4] Jahromi, M. and Tavakkoli-Moghaddam, R. (2012) A Novel 0-1 Linear Integer Programming Model for Dynamic Machine-Tool Selection and Operation Allocation in a Flexible Manufacturing System. *Journal of Manufacturing Systems*, **31**, 224-231. <https://doi.org/10.1016/j.jmsy.2011.07.008>
- [5] Ulker, O. and Landa-Silva, D. (2010) A 0/1 Integer Programming Model for the Office Space Allocation Problem. *Electronic Notes in Discrete Mathematics*, **36**, 575-582. <https://doi.org/10.1016/j.endm.2010.05.073>
- [6] Alhamad, K., Alkhezi, Y. and Alhajri, M.F. (2023) Nonlinear Integer Programming for Solving Preventive Maintenance Scheduling Problem for Cogeneration Plants with Production. *Sustainability*, **15**, 239. <https://doi.org/10.3390/su15010239>