队伍编号	MC2308535
题号	A

基于 QUBO 模型的信用评分卡组合优化

摘　要

银行进行信用卡及贷款业务时，需要选择最合理的信用评分卡组合以及其阈值，使最终收入最多，这是典型的二值优化问题。针对银行建立基于 QUBO 模型的专用量子算法，可极大提高运算效率，具有重大实际意义。

针对问题一，首先根据题给收益计算方法，计算出每张信用评分卡所能得到的的最大收益；然后建立 100 个二值变量的优化模型，包括目标函数和约束条件；将目标函数和约束条件进行处理得到标准 QUBO 模型；采用**量子退火算法**求解 QUBO 模型，得到最大收入为 **61172 元**，选择第 **49 张**信用卡的**第 1 个**阈值（**0.82，0.005**），计算时间为 **0.5369s.**同时采用 **DPSO 算法**对传统二值优化模型进行计算，计算时间为 **6.384s.**

针对问题二，首先根据题给收益计算方法，推导出三重信用卡组合策略的阈值选择和总收益的关系式；然后推导出总收益与单张信用卡通过率与收益表达式，消除坏账率及多元耦合项；最后根据目标函数和约束条件，将优化模型转化为标准 QUBO 模型；采用**量子退火算法**求解 QUBO 模型，得到最大收入为 **27915 元，**阈值选择分别为**第 8 个，第 1 个，第 2 个，**计算时间为 **4.3527s.** 同时采用 **DPSO 算法**对传统二值优化模型进行计算，计算时间为 **13.9578s.**

针对问题三，结合问题二模型，首先建立 130 个二值变量表示选择的信用评分卡及其阈值，推导总收益表达式和约束条件；然后用对数法将高次多元项分解为二次项和一次项，消除总收益表达式中多元变量耦合项，将优化模型转化为标准 QUBO 模型，采用**量子退火算法**进行计算，得到最大收入为 **43881 元**，选择信用卡 88 阈值 2，信用卡 33 阈值 6，信用卡 49 阈值 3，计算时间为 **2.9622s.**同时采用 **DPSO 算法**对传统二值优化模型进行计算，计算时间为 **59.6842s.**

关键词：QUBO 模型，量子退火算法，DPSO 算法，信用评分卡，多项式化简

目　录

1 问题背景与问题重述

1.1 问题背景

在银行信用卡或相关的贷款等业务中，对客户授信之前，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。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。选择不同的信用评分卡，不同的阈值组合，会给银行带来不同的收入与损失，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。

QUBO 模型是指二次无约束二值优化模型，可以运行在量子计算机硬件上，研究基于 QUBO 模型的量子专用算法十分有应用价值，建立 QUBO 模型选择不同的信用评分卡及其阈值，可以极大加快计算效率。

1.2 问题重述

根据上述背景以及所给附件数据，本题需要建立 QUBO 模型，在三个问题中所给定的三种情况下，分别找到所需的信用评分卡及其对应阈值，使得银行的收益达到最大。

问题一中，需要在 100 个信用评分卡中找出 1 张及其对应阈值，使最终收益最大化。问题二中采用三重信用卡组合策略，需要在三个给定的信用评分卡中分别选取一个阈值使收益最大化。问题三可视为前两个问题情况的综合，需要在 100 种信用评分卡中，选择三个信用评分卡，同时还要确定每个评分卡的阈值，使最终收益最大化。

2 模型假设，问题分析与符号说明

2.1 模型假设

- 贷款资金为 1000000 元，银行贷款利息收入率为 8%。
- 单张信用评分卡对应阈值收益为负时，其收益记为 1。
- 在 QUBO 模型中，惩罚函数系数设置是合理的。

2.2 问题分析

问题一，需要从 100 张信用评分卡中选出 1 张及其对应阈值，使最终收入最大。对于此问题可以根据题给收益计算方法，先计算出每张信用评分卡所能得到的的最大收益；然后建立 100 个二值变量的优化模型，包括目标函数和约束条件；将目标函数和约束条件进行处理得到标准 QUBO 模型；最后求解即得出最大收益的信用评分卡及其对应阈值。

问题二，需要分别从信用评分卡 1、2、3 中分别选择阈值，使最终收益最大。对于此问题，先根据题目定义的收益计算方法，推导出三重信用卡组合策略的阈值选择和总收益的关系式；然后建立 30 个二值变量的优化模型，包括目标函数和约束条件，将问题转换为 30 选 3 的标准 QUBO 模型；最终选出三种信用评分卡所对应阈值。

问题三是前两个问题的综合情形，需要从 100 个信用评分卡中选择 3 个，并选择每个评分卡对应的阈值，使最终收益最多。对于此问题，直接的方法是设 130 个二值变量，前 100 个表示选中的评分卡编号，后 30 个表示阈值编号。然后推导总收益表达式和约束条件，将优化模型转化为标准 QUBO 模型，尝试求解。此问题中优化变量数目剧增，因此还需设法进行化简，降低问题复杂度。

2.3 符号说明

符号	意义
M	贷款资金
r	贷款利息收入率
y	贷款最终收入
h_{ij}	第 i 个信用评分卡中 第 j 个阈值的坏账率
t_{ij}	第 i 个信用评分卡中 第 j 个阈值的通过率
$H_i=[h_{i1} \quad h_{i2} \quad \dots \quad h_{i10}]^T$ $i=1,2,\dots,100$	第 i 个信用评分卡中 10 个不同阈值的坏账率
$T_i=[t_{i1} \quad t_{i2} \quad \dots \quad t_{i10}]^T$ $i=1,2,\dots,100$	第 i 个信用评分卡中 10 个不同阈值的通过率
$X_i=[x_{i1} \quad x_{i2} \quad \dots \quad x_{i10}]^T$ $i=1,2,\dots,100$	$x_{ij}=\begin{cases}0,\text{评分卡}i\text{中阈值}j\text{未选中}\\1,\text{评分卡}i\text{中阈值}j\text{被选中}\end{cases}$
$Y_i=[y_{i1} \quad y_{i2} \quad \dots \quad y_{i10}]^T$ $i=1,2,\dots,100$	y_{ij}表示单一信用卡策略下，仅选中信用 评分卡 i 中阈值 j 所能带来的收益
$U=[u_1 \quad u_2 \quad \dots \quad u_{100}]^T$	$u_i=\begin{cases}0,\text{评分卡}i\text{未选中}\\1,\text{评分卡}i\text{被选中}\end{cases}$ 其中 $i=1,2,\dots,100$

3 问题一的模型建立与求解

问题一的建模与求解遵循计算收益，构建优化模型，转化为 QUBO 模型，MATLAB 求解几个主要步骤进行。基本思路如图 1 所示。

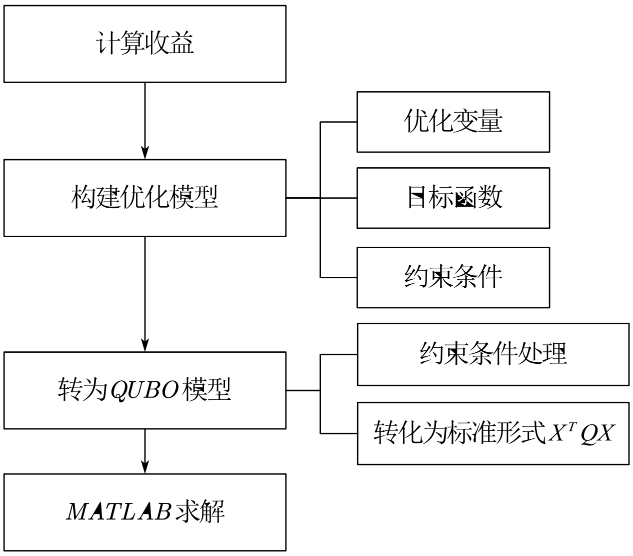

图 1 问题一思路简图

3.1 收益计算

问题 1 中，采用单信用评分卡策略，即在 100 个信用评分卡中选择 1 个，再从中选择 1 个阈值，使得贷款总收益达到最大。若选中信用评分卡 i 中的阈值 j，则通过率为 t_{ij}，坏账率为h_{ij}，银行获得收益为

$$\begin{aligned} y_{ij} &= Mrt_{ij}(1-h_{ij}) - Mt_{ij}h_{ij} \\ &= Mt_{ij}[r-(1+r)h_{ij}] \\ &= 1000000\ t_{ij}(0.08-1.08h_{ij}) \end{aligned} \tag{1}$$

其中 i =1,2,…,100，j =1,2,…,10。由于坏账率的不同，式中$(0.08-1.08h_{ij})$一项可能为负，导致亏损。

在单信用评分卡策略下，选择每种阈值能带来的收益均为已知的，所以选择评分卡 i 的最大收益为

$$y_i = \max\{y_{i1}, y_{i2}, \ldots, y_{i10}\},\quad i=1,2,\ldots,100 \tag{2}$$

记$Y=[y_1 \quad y_2 \quad \ldots \quad y_{100}]^T$。

3.2 模型建立

3.2.1 优化问题建模

设问题 1 的决策变量为$U=[u_1 \quad u_2 \quad \ldots \quad u_{100}]^T$，则最终收益为

$$y=\sum_{i=1}^{100} u_i y_i = U^T Y \tag{3}$$

由于二值优化问题中，每个待优化变量只能取 0 或 1，所以有$u_i = u_i^2$。应用到上式中得到

$$y=\sum_{i=1}^{100} u_i^2 y_i \tag{4}$$

这是一个齐二次式，将其用矩阵形式写为

$$y=[u_1 \quad u_2 \quad \ldots \quad u_{100}]\underbrace{\begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & y_{100} \end{bmatrix}}_{\triangleq \mathrm{P}}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{100} \end{bmatrix}=U^T P U \tag{5}$$

决策变量 U 中，每个u_i表示选取或不选取信用评分卡 i，选取时记为 1，不选时记为 0。而单信用评分卡策略下，仅有一个评分卡能被选取。即

$$\begin{cases} u_i \in \{0,1\},\ i=1,2,...,100 \\ \sum_{i=1}^{100} u_i = 1 \end{cases} \tag{6}$$

因此，问题 1 优化模型表示为

$$\begin{aligned} & \max\ y = U^T P U \\ & U=[u_1 \quad u_2 \quad \ldots \quad u_{100}]^T \\ & \text{s.t.}\ \sum_{i=1}^{100} u_i = 1 \\ & \qquad u_i \in \{0,1\} \end{aligned} \tag{7}$$

3.2.2 优化模型转化为 QUBO 模型

式(7)表示的优化模型是一个二次二值有约束优化模型，需将其转化为标准的 QUBO 模型。根据式(7)中的约束条件定义惩罚函数

$$H(U)=-\alpha\left(1-\sum_{i=1}^{100}u_i\right)^2 \tag{8}$$

其中 α 为正的常数。新的目标函数变为

$$J=y+H(U)=U^TPU-\alpha\left(1-\sum_{i=1}^{100}u_i\right)^2 \tag{9}$$

惩罚函数 $H(U)\leqslant 0$，当且仅当约束条件满足时 $H(U)=0$。选取足够大的 α，使得约束条件不满足时，目标函数总是不能被最大化。通常选取 α 使得惩罚函数取值至少能达到原目标函数的 75%~150%。为此，计算附件数据中各信用评分卡的最大收益，发现前三个信用评分卡的最大收益分别为 50130，37217 和 54087（元）。因此在惩罚项中，取 α=30000。新的目标函数为

$$\begin{aligned}
J&=\sum_{i=1}^{100}y_i{u_i}^2-\alpha\left(\sum_{j=1}^{100}\sum_{k=1}^{100}u_ju_k-\sum_{i=1}^{100}2u_i+1\right)\\
&=\sum_{i=1}^{100}y_i{u_i}^2-\alpha\sum_{j=1}^{100}\sum_{k=1}^{100}u_ju_k+2\alpha\sum_{i=1}^{100}u_i-\alpha\\
&=\sum_{i=1}^{100}y_i{u_i}^2-\alpha\left(\sum_{j=1}^{100}\sum_{\substack{k=1\\k\neq j}}^{100}u_ju_k+\sum_{i=1}^{100}u_i^2\right)+2\alpha\sum_{i=1}^{100}{u_i}^2-\alpha\\
&=\sum_{i=1}^{100}(y_i+\alpha){u_i}^2-\alpha\sum_{j=1}^{100}\sum_{\substack{k=1\\k\neq j}}^{100}u_ju_k-\alpha\\
&=U^TQU+c
\end{aligned} \tag{10}$$

其中矩阵 Q 为（已将 α=30000 代入）

$$Q=\begin{bmatrix}
y_1+30000 & -30000 & \cdots & \cdots & -30000\\
-30000 & y_2+30000 & \cdots & \cdots & -30000\\
\vdots & \vdots & \ddots & & \vdots\\
\vdots & \vdots & & \ddots & \vdots\\
-30000 & -30000 & \cdots & \cdots & y_{100}+30000
\end{bmatrix},\quad c=-30000 \tag{11}$$

由于常数 c 对求解结果没有影响，不对其进行考虑。为了求解方便，将问题表示为求最小值的形式。所以，问题 1 的 QUBO 模型为

$$\begin{aligned}
&\min\ J=-U^TQU\\
&U=[u_1\ \ u_2\ \ \ldots\ \ u_{100}]^T\\
&\text{s.t.}\ \ u_i\in\{0,1\}
\end{aligned} \tag{12}$$

3.3 模型求解

3.3.1 量子退火算法在 QUBO 问题中的应用

模拟退火算法来源于固体退火原理：将固体加温至充分高，再让其逐渐冷却。冷却过程中固体的内能逐渐降低，内部粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。模拟退火算法从某一较高初温出发，伴随温度参数的不断下降，结合一定的概率突跳特性在解空间中随机寻找目标函数的全局最优解，即在局部最优解能概率性地跳出并最终趋于全局最优。

假设材料在状态i之下的能量为$E(i)$，那么材料在温度 T 时状态i进入状态j就遵循如下规律：

（1）如果$E(j) \le E(i)$，则接受该状态下被转化。

（2）如果$E(j) > E(i)$，则状态转化以如下概率p被接受

$$p = e^{\frac{E(i)-E(j)}{KT}} \tag{13}$$

其中：K是物理学中的波尔斯曼常数；T是材料温度。

在某一个特定的温度下，进行充分的转化后，材料将达到热平衡。这是材料处于状态i的概率满足波尔斯曼分布

$$P(X=i) = \frac{e^{-\frac{E(i)}{KT}}}{\sum_{j \in S} e^{-\frac{E(i)}{KT}}} \tag{14}$$

其中：X表示材料当前状态的随机变量；S表示状态空间集合。在温度逐渐降低的过程中解的分布为

$$P_i^* = \begin{cases} \frac{1}{|S_{min}|} & x_i \in S_{min} \\ 0 & others \end{cases} \tag{15}$$

并且

$$\sum_{x_i \in S_{min}} P_i^* = 1 \tag{16}$$

这说明如果温度下降缓慢，而在每个温度下都有足够多次的转移，使之在每个温度下达到热平衡，则全局最优解以 1 的概率被找到。说明模拟退火算法可以找到全局最优解。

量子退火算法是模拟退火算法的一种延伸和改进。量子退火算法模型一般由两个部

分构成：第一部分为量子势能，其目的是将量子优化问题与量子系统形成映射，将优化的目标函数映射为施加在该量子系统的一个势场；第二部分为量子动能，通过引入动能项(幅度可控)作为控制量子波动的穿透场。在势能和动能两个场的作用下，量子系统的演化就可通过式(17)的薛定谔方程来描述：

$$ih\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{17}$$

实际中，直接求解薛定谔方程难度很大，其计算复杂度随着问题复杂度的增加呈指数增长，所以研究中通常采取随机过程进行模拟量子退火的过程(薛定谔方程直接求解的代价较大)，其中路径积分蒙特卡罗(Path Integral Monte Carlo, PIMC)是模拟量子退火过程的有效随机过程方法。

$$H(t) = H_{pot}(t) + H_{kin}(t) \tag{18}$$

式中，量子哈密顿函数 $H(t)$表示量子退火算法中的评价函数(Cost Function)，$H_{pot}(t)$表示势能，对应模拟退火算法的评价函数，$H_{kin}(t)$表示动能，在动能中加入量子波动(初始值较大，然后按照一定的进度表慢慢减小到零)。

如图 1 所示，在经典的热退火情况下，系统要达到全局最小值，必须克服 O(N)的较大势垒ΔE，其中 N 为系统的大小(温度为 T 时，逃逸概率为 $\exp(-\Delta E/T)$，而在量子退火的情况下，系统可以隧穿势垒。如果势垒较窄，隧穿概率为$\exp\left(-\omega\sqrt{\Delta E}/\Gamma\right)$，其中 Γ 为隧穿波动场，ω 为势垒宽度。

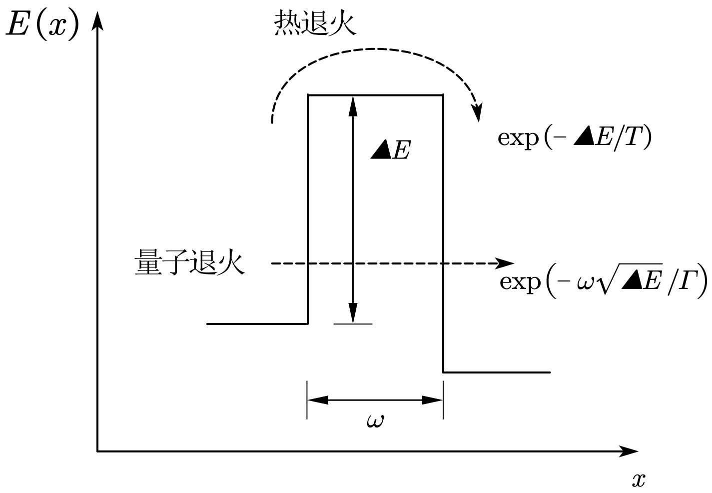

图 2 热退火与量子退火穿越势垒的比较

如图 3 所示，模拟退火算法只能通过翻越势垒的方式从局部最小值 P 到达全局最小值 P′，而量子退火算法凭借其量子隧穿效应，可以直接从局部最小点 P 到达 P′。量子退火算法凭借其量子隧穿效应来跳出局部最优，这也是与模拟退火相比一个最大的不同。所以量子退火算法在某些问题上具有比模拟退火算法更好的性能。

量子退火算法的测试模型为横向场随机伊辛模型.许多组合优化类问题先实现对伊辛模型的映射，然后再通过量子退火算法进行求解。横向场随机伊辛模型哈密顿函数为

$$H_p=\sum_{i=1}^{N}h_i\sigma_i^z+\sum_{i,j=1}^{N}J_{ij}\sigma_i^z\sigma_j^z \tag{19}$$

其中，h_i 为能量偏移度,σ_i^z 表示泡利自旋矩阵,J_{ij} 表示自旋量i 和 j 之间的耦合度。

根据伊辛模型的哈密顿函数，可以得到量子退火的哈密顿函数为

$$H(t)=H_p+\Gamma(t)\sum_{i=1}^{N}\Delta_i\sigma_i^x \tag{20}$$

式中，Γ 代表场强，其诱导单个自旋状态向上和向下的转变，它和模拟退火算法中的温度T 作用类似。任何表示为上述形式的优化问题都可以由量子退火算法处理。

与模拟退火基于热力学的原理不同，量子退火算法主要利用量子涨落的机制，即量子隧穿效应，来完成优化过程.量子退火算法的步骤表示如下：

步骤 1：根据待优化问题，构造量子系统的评价函数$H_q=H_{pot}+H_{kin}$，即量子哈密顿函数。其中，H_{pot} 为势能，即模拟退火算法中的评价函数，H_{kin} 为动能；

步骤 2：初始化各个参数，T_0 为量子退火的初始温度，Γ 为横向场强，变化的横向场强引起不同量子状态之间的量子跃迁，最大迭代次数为$MaxSteps$，初始化状态为x，对应的状态能量为$H_{pot}(x)$；

步骤 3：随机微扰产生新状态x'，对应的状态能量为$H_{pot}(x')$；

步骤 4：计算能量差$\Delta H_{pot}=H_{pot}(x')-H_{pot}(x)$以及$\Delta H_q=H_q(x')-H_q(x)$，如果$\Delta H_{pot}<0$ 或者 $\Delta H_q<0$，则系统接受新解 $x=x'$，反之，如果$\exp\left(\frac{\Delta H_q}{T}\right)<rangdom(0,1)$，则$x=x'$，否则，重复执行步骤 3；

步骤 5：进行退温操作，Γ 的变化和模拟退火中的温度T 作用类似，横向场强变化形式为$\Gamma=\Gamma-(\Gamma_0/MaxSteps)$；

步骤 6：判断是否满足终止条件$\Gamma=0$，如果满足，量子退火算法终止，否则，重复步骤 3。

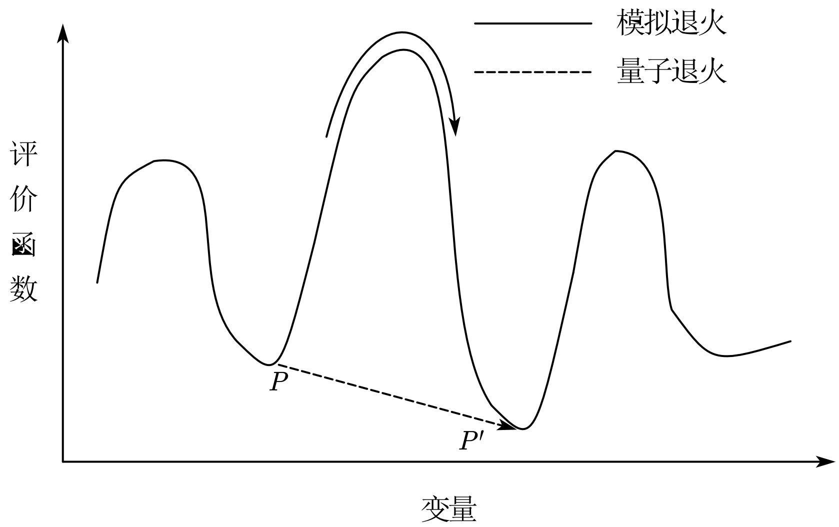

图 3 模拟退火算法与量子退火算法工作原理的比较

3.3.2 DPSO 算法在离散问题中的应用

粒子群优化算法(PSO：Particle swarm optimization) 是一种进化计算技术（evolutionary computation)。源于对鸟群捕食的行为研究。粒子群优化算法的基本思想：是通过群体中个体之间的协作和信息共享来寻找最优解，

在使用粒子群模型前，定义以下参数：

粒子 i 在 t 时刻的位置为：

$$x_i{}^t=(x_{i1}{}^t,x_{i2}{}^t,...,x_{id}{}^t),x_{id}{}^t\in[I_d,u_d] \tag{21}$$

粒子 i 在 t 时刻的速度为：

$$v_i{}^t=(v_{i1}{}^t,v_{i2}{}^t,...,v_{id}{}^t),v_{id}{}^t\in[v_{\min},v_{\max}] \tag{22}$$

式中，$v_{\max}$，$v_{\min}$ 分别表示粒子群的最大值和最小值；I_d，u_d 分别表示粒子搜索空间的上下限。

粒子每次迭代时记录两个位置，分别为个体最优值 Pbset 和群体最优值 Gbset，分别

表示为：

$$
\begin{aligned}
p_i{}^t &= (p_{i1}{}^t, p_{i2}{}^t, \ldots, p_{id}{}^t) \\
p_g{}^t &= (p_{g1}{}^t, p_{g2}{}^t, \ldots, p_{gd}{}^t)
\end{aligned} \tag{23}
$$

根据以上定义，粒子 i 在 t+1 时刻的速度、位置更新公式为：

$$
\begin{cases}
v_{id}{}^{t+1} = \omega v_{id}{}^t + c_1 r_1 (p_{id}{}^t - x_{id}{}^t) + c_2 r_2 (p_{gd}{}^t - x_{gd}{}^t) \\
x_{id}{}^{t+1} = x_{id}{}^t + v_{id}{}^{t+1}
\end{cases} \tag{24}
$$

在计算上仍保留经典粒子群算法速度-位置更新运算规则，将离散问题空间映射到连续粒子运动空间，并适当求改粒子群算法来求解，

$$
s(v_i{}^t) = \frac{1}{1 + e^{-v_i{}^t}} \tag{25}
$$

$$
x_{id}{}^t = \begin{cases} 1 & r < s(v_i{}^t) \\ 0 & otherwise \end{cases} \tag{26}
$$

PSO 算法流程图如图 4 所示。

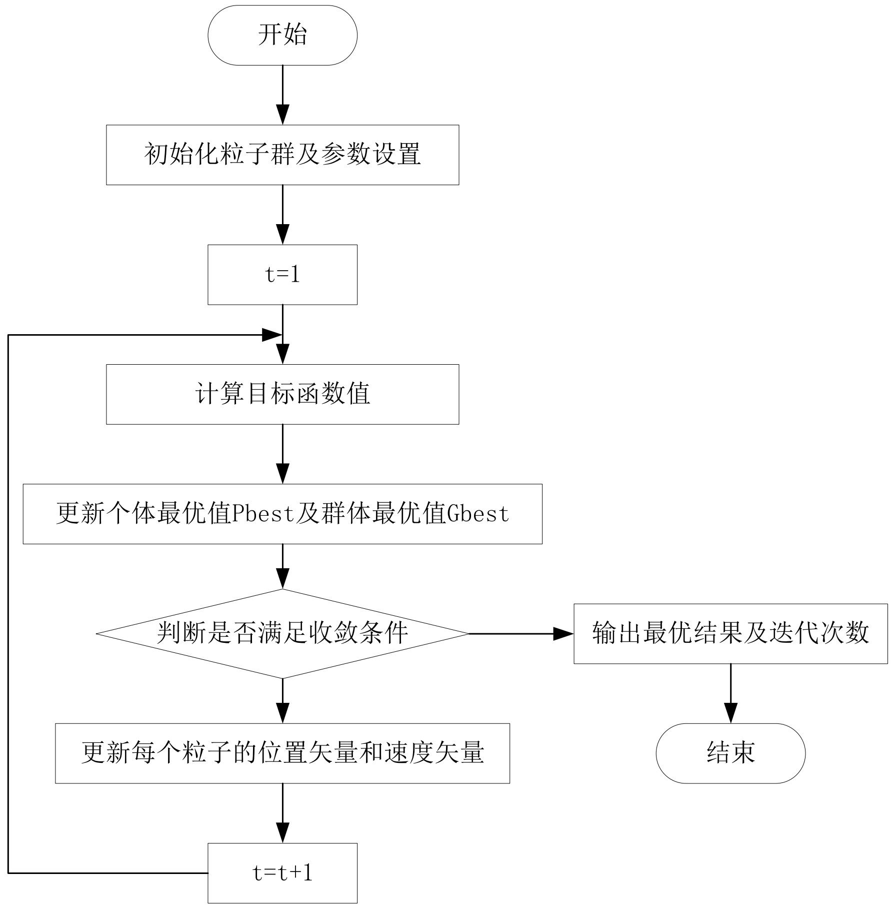

图 4 PSO 算法流程图

主要包括以下几个步骤：

Step1：初始化粒子群。

Step2：计算每个粒子的适应度。

Step3：对每个粒子进行当前适应度和个体最优值 Pbset 的适应度的值进行比较，若当前适应度值较好，则将其设为个体最优值。

Step4：对每个粒子进行当前适应度和群体最优值 Gbset 的适应度的值进行比较，若当前适应度值较好，则将其设为全局最优值。

Strp5：对每个粒子的位置和速度更新。

Step6：如果达到最大迭代次数或者最佳适应度值的增量小于某个给定的阈值时，则输出最优解，否则返回 Step2。

3.3.3 求解过程及求解结果

在前文建立的 QUBO 模型的基础上，利用 MATLAB　OPTI 优化工具箱中的 SCIP 求解器对该模型进行求解。步骤如下：

(1) 代入数据计算得出 Q 矩阵；

```
E0=load('shouru.mat');
Q0=E0.E1;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;
```

(2) 设定参数，调用求解器

```
m = 100;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

(3) 得到结果

表 1 问题一求解结果

收益（元）	信用卡选择	阈值选择（通过率，坏账率）
61172	第 49 张	第 1 个（0.82，0.005）

为验证 QUBO 模型结果的准确性以及算法的优越性，采用 DPSO 算法对同一问题进行建模求解，得到 DPSO 算法的收益进化曲线如图 5 所示。

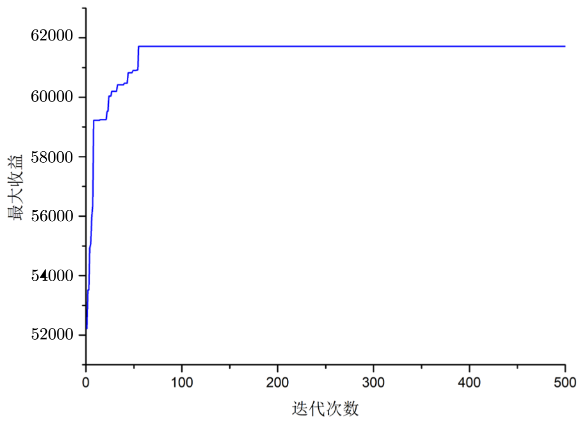

图 5 DPSO 算法最大收益进化曲线

将 DPSO 算法与 QUBO 模型对应的量子退火算法求解结果进行对比分析，如表 2，两种模型计算计算结果相同，但是 DPSO 模型计算时间为 QUBO 模型的 12 倍左右。

表 2 DPSO 模型与 QUBO 模型对比

模型	最终收入（元）	信用卡选择	计算时间
DPSO	61172	第 49 张	6.384s
QUBO	61172	第 49 张	0.5369s

4 问题二的模型建立与求解

问题二的建模与求解遵循计算收益，化简收益表达式，构建优化模型，转化为 QUBO 模型，MATLAB 求解几个主要步骤进行。相比问题一，问题二的总收益表达式复杂度增加，因此需要单独对收益表达式作化简。基本思路如图 6 所示。

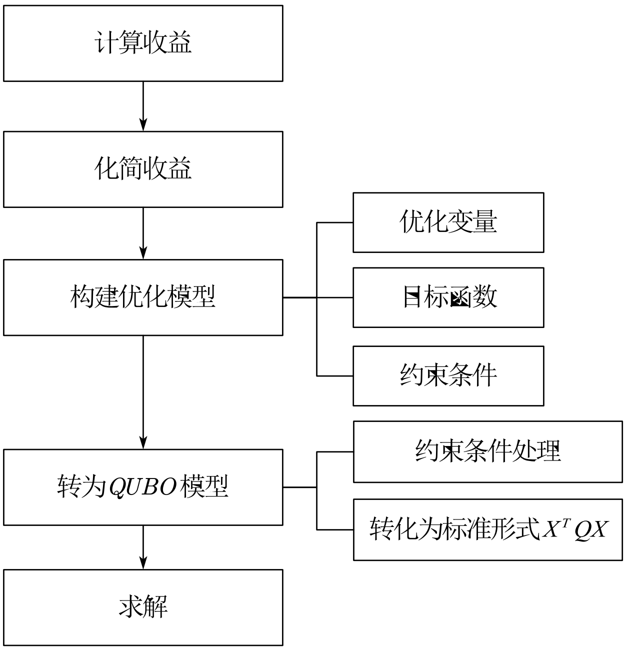

图 6 问题二思路简图

4.1 收益计算

在问题 1 中，已经得到了单信用评分卡策略下的贷款收益表达式(1)。问题 2 需采用三重信用卡组合策略，即选择评分卡 1，2，3，然后在评分卡 1，2，3 中分别选择阈值 $i,\ j,\ k$ (=1,2,…,10)来计算通过率、坏账率及最终收益。总通过率为所有信用评分卡通过率相乘，即

$$t_z = t_{1i} t_{2j} t_{3k} \tag{27}$$

总坏账率为三种信用评分卡对应坏账率的平均值，即

$$h_z = \frac{h_{1i} + h_{2j} + h_{3k}}{3} \tag{28}$$

将式(27)(28)代入式(1)，得到

$$
\begin{aligned}
y &= Mt_z[r-(1+r)h_z] \\
&= Mt_{1i}t_{2j}t_{3k}\left[r-(1+r)\frac{h_{1i}+h_{2j}+h_{3k}}{3}\right]
\end{aligned} \tag{29}
$$

为了使式(29)便于计算，考虑如下方法：在单信用卡策略下，评分卡 1 中选择阈值 i，评分卡 2 中选择阈值 j 的收益分别为

$$
y_{1i} = Mt_{1i}[r-(1+r)h_{1i}] \tag{30}
$$

$$
y_{2j} = Mt_{2j}[r-(1+r)h_{2j}] \tag{31}
$$

在双信用卡组合策略下，同时从评分卡 1 中选择阈值 i，评分卡 2 中选择阈值 j 的收益为

$$
y_{(1i,2j)} = Mt_{1i}t_{2j}\left[r-(1+r)\frac{h_{1i}+h_{2j}}{2}\right] \tag{32}
$$

对式(32)作等价变换：

$$
\begin{aligned}
y_{(1i,2j)} &= Mt_{2j}\left[rt_{1i}-(1+r)t_{1i}\frac{h_{1i}+h_{2j}}{2}\right] \\
&= Mt_{2j}\left[\frac{r}{2}t_{1i}+\left(\frac{r}{2}t_{1i}-(1+r)t_{1i}\frac{h_{1i}}{2}\right)-(1+r)t_{1i}\frac{h_{2j}}{2}\right] \\
&= Mt_{2j}\left[\frac{r}{2}t_{1i}+\frac{1}{2}(rt_{1i}-(1+r)t_{1i}h_{1i})-(1+r)t_{1i}\frac{h_{2j}}{2}\right] \\
&= \frac{1}{2}Mt_{2j}[rt_{1i}-(1+r)t_{1i}h_{1i}]+\frac{1}{2}Mt_{2j}[rt_{1i}-(1+r)t_{1i}h_{2j}] \\
&= \frac{1}{2}t_{2j}\underbrace{Mt_{1i}[r-(1+r)h_{1i}]}_{y_{1i}}+\frac{1}{2}t_{1i}\underbrace{Mt_{2j}[r-(1+r)h_{2j}]}_{y_{2j}} \\
&= \frac{1}{2}(y_{1i}t_{2j}+t_{1i}y_{2j})
\end{aligned} \tag{33}
$$

经过式(33)的等价变换后，双信用卡组合策略的收益可直接由两个单信用卡评分策略的收益和通过率计算得到，表达式变得简洁。对于三重信用卡组合策略的收益，即式(29)也可作类似的等价变换：

$$
\begin{aligned}
y_{(1i,2j,3k)} &= Mt_{1i}t_{2j}t_{3k}\left[r-(1+r)\frac{h_{1i}+h_{2j}+h_{3k}}{3}\right] \\
&= Mt_{3k}\left[rt_{1i}t_{2j}-(1+r)t_{1i}t_{2j}\frac{h_{1i}+h_{2j}+h_{3k}}{3}\right] \\
&= Mt_{3k}\left[\left(\frac{2rt_{1i}t_{2j}}{3}+\frac{rt_{1i}t_{2j}}{3}\right)-(1+r)t_{1i}t_{2j}\frac{h_{1i}+h_{2j}}{3}-(1+r)t_{1i}t_{2j}\frac{h_{3k}}{3}\right] \\
&= Mt_{3k}\left[\frac{2rt_{1i}t_{2j}}{3}-\frac{2}{3}(1+r)t_{1i}t_{2j}\frac{h_{1i}+h_{2j}}{2}+\frac{rt_{1i}t_{2j}}{3}-\frac{1}{3}(1+r)t_{1i}t_{2j}h_{3k}\right] \\
&= Mt_{3k}\left[\frac{2}{3}\left(rt_{1i}t_{2j}-(1+r)t_{1i}t_{2j}\frac{h_{1i}+h_{2j}}{2}\right)+\frac{1}{3}t_{1i}t_{2j}(r-(1+r)h_{3k})\right] \\
&= \frac{2}{3}t_{3k}\underbrace{Mt_{1i}t_{2j}\left(r-(1+r)\frac{h_{1i}+h_{2j}}{2}\right)}_{y_{(1i,2j)}}+\frac{1}{3}t_{1i}t_{2j}\underbrace{Mt_{3k}(r-(1+r)h_{3k})}_{y_{3k}} \\
&= \frac{2}{3}t_{3k}y_{(1i,2j)}+\frac{1}{3}t_{1i}t_{2j}y_{3k} \\
&= \frac{1}{3}(t_{1i}t_{2j}y_{3k}+t_{1i}y_{2j}t_{3k}+y_{1i}t_{2j}t_{3k})
\end{aligned} \tag{34}
$$

于是，三重信用卡组合策略的收益可直接由三个单信用卡评分策略的收益和通过率计算得到。

4.2 模型建立

4.2.1 优化问题建模

问题 2 的决策变量为

$$
\begin{aligned}
X &= [\ X_1^T \ \ |\ \ X_2^T \ \ |\ \ X_3^T\ ]^T \\
&= [\ x_{(1,1)} \ \dots \ x_{(1,10)} \ \ |\ \ x_{(2,1)} \ \dots \ x_{(2,10)} \ \ |\ \ x_{(3,1)} \ \dots \ x_{(3,10)}\ ]^T
\end{aligned} \tag{35}
$$

该决策变量 X 需满足的条件是

$$
\begin{cases}
x_{ij}\in\{0,1\} \\
\sum_{j=1}^{10} x_{ij}=1
\end{cases} \tag{36}
$$

其中 i=1,2,3，j=1,2,…,10。用决策变量将通过率和坏账率表示为

$$
\begin{aligned}
t_i &= X_i^T T_i \\
h_i &= X_i^T H_i
\end{aligned} \tag{37}
$$

总收益率和总坏账率为

$$
\begin{aligned}
&t_z = t_1 t_2 t_3 = X_1^T T_1 X_2^T T_2 X_3^T T_3 \\
&h_z = \frac{h_1 + h_2 + h_3}{3} = \frac{1}{3}\left(X_i^T H_i + X_2^T H_2 + X_3^T H_3\right)
\end{aligned}
\tag{38}
$$

进而可以根据式(29)计算总收益。但是将式(38)中各向量展开计算时产生的子项较多，可以直接利用式(29)和式(34)的等价性，得到总收益为

$$
\begin{aligned}
y &= \sum_{i=1}^{10}\sum_{j=1}^{10}\sum_{k=1}^{10} x_{1i} x_{2j} x_{3k} \frac{t_{1i} t_{2j} y_{3k} + t_{1i} y_{2j} t_{3k} + y_{1i} t_{2j} t_{3k}}{3} \\
&= \sum_{i,j,k} x_{1i} x_{2j} x_{3k} y_{(1i,\,2j,\,3k)}
\end{aligned}
\tag{39}
$$

对式(39)作变量代换

$$
z_s \triangleq x_{1i} x_{2j} x_{3k} \tag{40}
$$

其中 z 的下标 s 满足

$$
s = (i-1)\times 100 + (j-1)\times 10 + (k-1) \tag{41}
$$

则 s=0,1,2,…,999。当且仅当 $x_{1i} = x_{2j} = x_{3k} = 1$ 时 $z_s = 1$，其余情况 $z_s = 0$，所以 z_s 也是二值变量，有 $z_s = z_s^2$。要满足(36)，则有

$$
\sum_{s=0}^{999} z_s = 1 \tag{42}
$$

同时，令 $y_s = y_{(1i,\,2j,\,3k)}$，则式(39)变为

$$
\begin{aligned}
y &= \sum_{s=0}^{999} z_s y_s = \sum_{s=0}^{999} {z_s}^2 y_s \\
&= \begin{bmatrix} z_0 & z_1 & \dots & z_{999} \end{bmatrix} \underbrace{\begin{bmatrix} y_0 & & & \\ & y_1 & & \\ & & \ddots & \\ & & & y_{999} \end{bmatrix}}_{\triangleq \mathrm{P}} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{999} \end{bmatrix} \\
&\triangleq Z^T P Z
\end{aligned}
\tag{43}
$$

问题 2 的二次二值优化模型表示为

$$
\begin{aligned}
&\max\ y = Z^T P Z \\
&Z = \begin{bmatrix} z_0 & z_1 & \dots & z_{999} \end{bmatrix} \\
&s.t. \quad \sum_{s=0}^{999} z_s = 1,\ z_s \in \{0, 1\}
\end{aligned}
\tag{44}
$$

4.2.2 优化模型转化为 QUBO 模型

根据优化问题(44)中的约束条件，设惩罚函数

$$H(Z) = -\alpha\left(1 - \sum_{s=0}^{999} z_s\right)^2 \tag{45}$$

其中 α 是正的常数。新的目标函数为

$$\begin{aligned} J &= y + H(Z) \\ &= \sum_{s=0}^{999} {z_s}^2 y_s - \alpha\left(1 - \sum_{s=0}^{999} z_s\right)^2 \\ &= \sum_{s=0}^{999} {z_s}^2 y_s - \alpha\left(\sum_{p=0}^{999}\sum_{q=0}^{999} z_p z_q - 2\sum_{p=0}^{999} z_p + 1\right) \\ &= \sum_{s=0}^{999} {z_s}^2 y_s - \alpha\left(\sum_{p=0}^{999}\sum_{\substack{q=0 \\ q\neq p}}^{999} z_p z_q + \sum_{p=0}^{999} {z_p}^2\right) + 2\alpha\sum_{p=0}^{999} {z_p}^2 - \alpha \\ &= \sum_{s=0}^{999} (y_s + \alpha) u_i^2 - \alpha\sum_{p=0}^{999}\sum_{\substack{q=0 \\ q\neq p}}^{999} z_p z_q - \alpha \\ &= Z^T Q Z + c \end{aligned} \tag{46}$$

其中

$$Q = \begin{bmatrix} y_0 + \alpha & -\alpha & \cdots & -\alpha \\ -\alpha & y_1 + \alpha & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ -\alpha & \cdots & \cdots & y_{999} + \alpha \end{bmatrix},\quad c = -\alpha \tag{47}$$

取 α=30000，可满足惩罚函数 *H(Z)*至少能达到原目标函数的 75%~150%这一准则。同样不考虑常数 c 并将问题表示为最小化目标函数的形式，得到问题 2 的 QUBO 模型为

$$\begin{aligned} &\min\ -Z^T Q Z \\ &Z = [z_0 \quad z_1 \quad \ldots \quad z_{999}]^T \\ &\text{s.t.}\quad z_i \in \{0, 1\} \end{aligned} \tag{48}$$

4.3 模型求解

在前文建立的 QUBO 模型的基础上，利用 MATLAB 中的 OPTI 优化工具箱中的 SCIP 求解器对该模型进行求解。

步骤如下：

(1)代入数据计算得出 Q 矩阵；

```
E0=load('shouru2.mat');
Q0=E0.E2;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;
```

(2)设定参数，调用求解器

```
m = 1000;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

(3)得到结果

求解器输出结果中，$z_{701}=1$，其他$z=0$。根据式(41)得到，选择信用卡 1 的阈值 8，信用卡 2 的阈值 1，信用卡 3 的阈值 2 作为信用评价标准，可以使收益最大化。

表 3 问题 2 求解结果

最终收入（元）	信用卡选择	阈值选择（通过率，坏账率）
	第 1 张	第 8 个（0.94，0.043）
27915	第 2 张	第 1 个（0.72，0.032）
	第 3 张	第 2 个（0.82，0.013）

同时建立标准 01 模型，采用 DPSO 算法对同一问题进行求解，得到最大收入进化曲线如图 7 所示

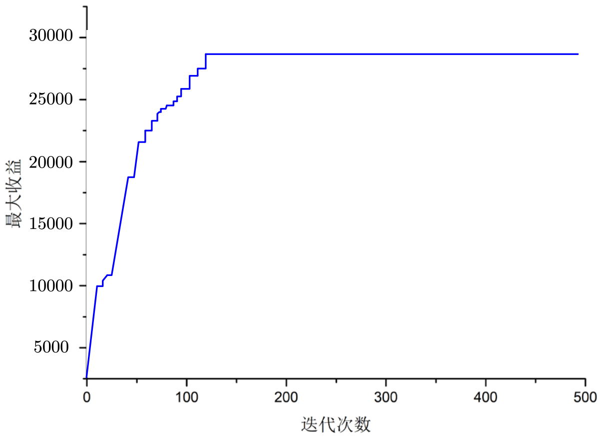

图 7 DPSO 算法最大收入进化曲线

将 DPSO 算法与 QUBO 模型对应的量子退火算法求解结果进行对比分析，如表 4，两种模型计算结果相同，但是 DPSO 算法计算时间远大于为 QUBO 模型下的量子退火求解算法。

表 4 DPSO 模型与 QUBO 模型对比

模型	最终收入（元）	阈值选择（通过率，坏账率）	计算时间
DPSO	27915	第 8 个（0.94，0.043） 第 1 个（0.72，0.032） 第 2 个（0.82，0.013）	13.9578s
QUBO	27915	第 8 个（0.94，0.043） 第 1 个（0.72，0.032） 第 2 个（0.82，0.013）	4.3527s

5 问题三的模型建立与求解

问题三的建模与求仍然解遵循计算收益，化简收益表达式，构建优化模型，转化为QUBO 模型，MATLAB 求解几个主要步骤进行。相比问题一和二，问题三的优化变量个数更多，在化简过程中首先处理带来负收益的阈值，并且对三次交叉项使用对数法分解为一次项，便于 QUBO 模型的建立。基本思路如图 8 所示。

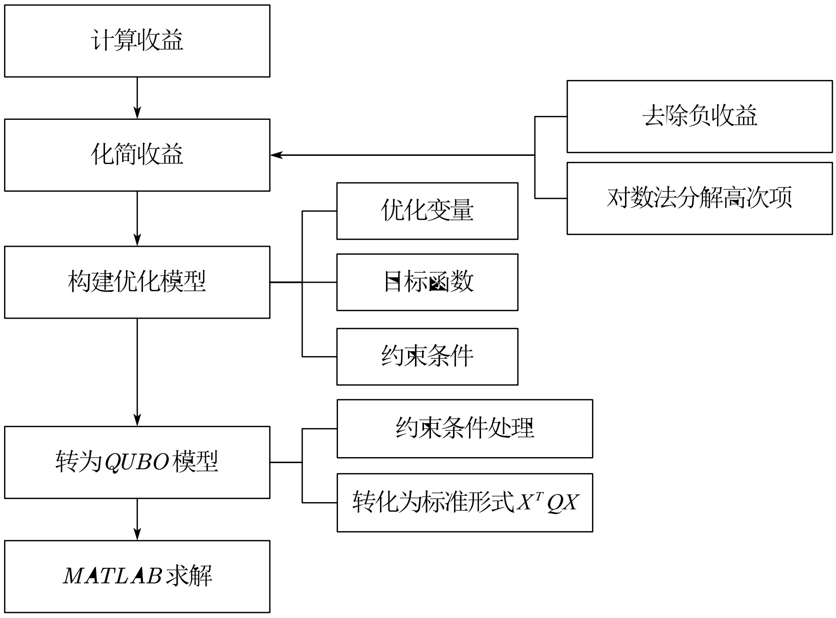

图 8 问题三思路简图

5.1 模型建立

5.1.1 目标函数

在问题二中，已经得到了三重信用卡评分策略的总收益计算式(34)，其中信用评分卡编号给定为 1，2，3。在问题三中没有这一限制，设选中的信用评分卡编号分别为$m,p,q=1,2,...,100$ 且 $m\neq p\neq q\neq m$；在第 m,p,q 个信用评分卡中选取的阈值为i,j,k=1,2,…,10。则问题三中的总收益可以表示为

$$y=\frac{1}{3}(t_{mi}t_{pj}y_{qk}+t_{mi}y_{pj}t_{qk}+y_{mi}t_{pj}t_{qk}) \tag{49}$$

由式(49)可知，计算贷款总收益 y 需要确定 m,p,q,i,j,k。设问题三的决策变量为

$$\boldsymbol{X}=[X_{mi}^T\quad X_{pj}^T\quad X_{qk}^T\quad U_m^T\quad U_p^T\quad U_q^T]^T \tag{50}$$

其中X_{mi}, X_{pj}, X_{qk}为 10 元向量，U_m, U_p, U_q为 100 元向量，其中元素均为 0 或 1，满足如下条件：

1. $X_{mi}=[x_{m1}\ \ x_{m2}\ \ \ldots\ \ x_{m10}]^T$，其中有且仅有第 i 个元素$x_{mi}=1$，其他元素为 *0*；同理，X_{pj}中仅有第 j 个元素$x_j=1$，X_{qk}中仅有第 k 个元素$x_k=1$。

2. $U_m=[u_{m1}\ \ u_{m2}\ \ \ldots\ \ u_{m100}]^T$，其中有且仅有第 m 个元素$u_{mm}=1$，其他元素为 *0*；同理，U_p 中仅有第 p 个元素$u_{pp}=1$，U_q 中仅有第 q 个元素$u_{qq}=1$。

定义 10×100 矩阵 T 和 Y 如下，其中元素为通过率和单信用卡评分策略下的收益：

$$T\triangleq\begin{bmatrix} t_{1,1} & \cdots & \cdots & \cdots & t_{1,100} \\ & \ddots & & \ddots & \\ \vdots & & t_{ij} & & \vdots \\ & \ddots & & \ddots & \\ t_{10,1} & \cdots & \cdots & \cdots & t_{10,100} \end{bmatrix}_{10\times 100},\ Y\triangleq\begin{bmatrix} y_{1,1} & \cdots & \cdots & \cdots & y_{1,100} \\ & \ddots & & \ddots & \\ \vdots & & y_{ij} & & \vdots \\ & \ddots & & \ddots & \\ y_{10,1} & \cdots & \cdots & \cdots & y_{10,100} \end{bmatrix}_{10\times 100} \tag{51}$$

于是，式(49)中各参数表示为

$$\begin{aligned} t_{mi}=X_{mi}^TTU_m,\ t_{pj}=X_{pj}^TTU_p,\ t_{qk}=X_{qk}^TTU_q \\ y_{mi}=X_{mi}^TYU_m,\ y_{pj}=X_{pj}^TYU_p,\ y_{qk}=X_{qk}^TYU_q \end{aligned} \tag{52}$$

总收益为

$$\begin{aligned} y=\frac{1}{3}(X_{mi}^TTU_mX_{pj}^TTU_pX_{qk}^TYU_q+X_{mi}^TTU_mX_{pj}^TYU_pX_{qk}^TTU_q \\ +X_{mi}^TYU_mX_{pj}^TTU_pX_{qk}^TTU_q) \end{aligned} \tag{53}$$

式(53)的计算十分复杂，不利于模型的建立和求解。为了将模型简洁地表示为 QUBO 形式并求解，引入如下两个设定。

设定 *1*：式 *(49)* 中，任何一个单信用卡策略下的收益为负，都对整体收益有不利影响。所以只要某一阈值下出现负收益$y\leqslant 0$，则直接令$y=1$。

设定 *2*：设$0<x\leqslant a$, $0<y\leqslant b$, $0<z\leqslant c$，三元函数$f_1(x,y,z)=x+y+z$，$f_2(x,y,z)=\ln x+\ln y+\ln z$。求$f_1$, f_2对x,y,z的偏导数：

$$\frac{\partial f_1}{\partial x}=\frac{\partial f_1}{\partial y}=\frac{\partial f_1}{\partial z}=1>0$$

$$\frac{\partial f_2}{\partial x}=\frac{1}{x}>0,\ \frac{\partial f_2}{\partial y}=\frac{1}{y}>0,\ \frac{\partial f_2}{\partial z}=\frac{1}{z}>0$$

由于各偏导数均大于 0，f_1，f_2 对 x,y,z 都是增函数。因此在 x,y,z 各自的取值范围内最大化 $f_1(x,y,z)$ 的问题，也就可以等价为在 x,y,z 各自的取值范围内最大化 $f_2(x,y,z)$。若求得一组 x,y,z 使得 f_2 最大化，则也能同时使得 f_1 最大化。所以，最大化总收益式 *(49)*，等价于最大化式 *(54)*：

$$\bar{y}=\ln(t_{mi}t_{pj}y_{qk})+\ln(t_{mi}y_{pj}t_{qk})+\ln(y_{mi}t_{pj}t_{qk}) \tag{54}$$

由于设定 1 中已经规定收益小于等于 0 时会直接置于 1，所以式(54)中各个对数有意义。再定义 10×100 矩阵 $\bar{T}$ 和 $\bar{Y}$ 如下，也就是将式(51)定义的 T 和 Y 各个元素均取自然对数：

$$\bar{T}\triangleq\begin{bmatrix}\ln t_{1,1} & \cdots & \cdots & \cdots & \ln t_{1,100}\\ & \ddots & & \ddots & \\ \vdots & & \ln t_{ij} & & \vdots\\ & \ddots & & \ddots & \\ \ln t_{10,1} & \cdots & \cdots & \cdots & \ln t_{10,100}\end{bmatrix}_{10\times100}\quad \bar{Y}\triangleq\begin{bmatrix}\ln y_{1,1} & \cdots & \cdots & \cdots & \ln y_{1,100}\\ & \ddots & & \ddots & \\ \vdots & & \ln y_{ij} & & \vdots\\ & \ddots & & \ddots & \\ \ln y_{10,1} & \cdots & \cdots & \cdots & \ln y_{10,100}\end{bmatrix}_{10\times100} \tag{55}$$

将式(54)展开为

$$\bar{y}=2\ln t_{mi}+2\ln t_{pj}+2\ln t_{qk}+\ln y_{mi}+\ln y_{pj}+\ln y_{qk} \tag{56}$$

与式(52)类似，式(56)中

$$\begin{gathered}\ln t_{mi}=X_{mi}^T\bar{T}U_m,\ \ln t_{pj}=X_{pj}^T\bar{T}U_p,\ \ln t_{qk}=X_{qk}^T\bar{T}U_q\\ \ln y_{mi}=X_{mi}^T\bar{Y}U_m,\ \ln y_{pj}=X_{pj}^T\bar{Y}U_p,\ \ln y_{qk}=X_{qk}^T\bar{Y}U_q\end{gathered} \tag{57}$$

式(56)变为

$$\begin{aligned}\bar{y}&=X_{mi}^T\left(2\bar{T}+\bar{Y}\right)U_m+X_{pj}^T\left(2\bar{T}+\bar{Y}\right)U_p+X_{qk}^T\left(2\bar{T}+\bar{Y}\right)U_q\\ &=\boldsymbol{X}^T\boldsymbol{R}\boldsymbol{X}\end{aligned} \tag{58}$$

其中 $\boldsymbol{X}$ 由式(50)定义，矩阵 $\boldsymbol{R}$ 为

$$
\underbrace{\begin{bmatrix}
\boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & 2\bar{T}+\bar{Y} & \boldsymbol{0}_{10\times100} & \boldsymbol{0}_{10\times100} \\
\boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times100} & 2\bar{T}+\bar{Y} & \boldsymbol{0}_{10\times100} \\
\boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times10} & \boldsymbol{0}_{10\times100} & \boldsymbol{0}_{10\times100} & 2\bar{T}+\bar{Y} \\
\boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100} \\
\boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100} \\
\boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times10} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100} & \boldsymbol{0}_{100\times100}
\end{bmatrix}}_{330\times330} \tag{59}
$$

令$\boldsymbol{P}=(\boldsymbol{R}+\boldsymbol{R}^T)/2$，则$\boldsymbol{P}$为对称矩阵，有

$$
\bar{y}=\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X} \tag{60}
$$

5.1.2 约束条件

决策变量$\boldsymbol{X}$要满足的条件是

$$
\begin{gathered}
\sum_{i=1}^{10} x_{mi}=1,\ \sum_{j=1}^{10} x_{pi}=1,\ \sum_{k=1}^{10} x_{qk}=1 \\
\sum_{i=1}^{100} u_{mi}=1,\ \sum_{j=1}^{100} u_{pi}=1,\ \sum_{k=1}^{100} u_{qk}=1
\end{gathered} \tag{61}
$$

可以写成线性等式约束$\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$，其中$\boldsymbol{A}$为$6\times330$分块对角矩阵，$\boldsymbol{b}$为6元列向量：

$$
\boldsymbol{A}=\begin{bmatrix} \boldsymbol{1}_{1\times 10} & & & & & \\ & \boldsymbol{1}_{1\times 10} & & & & \\ & & \boldsymbol{1}_{1\times 10} & & & \\ & & & \boldsymbol{1}_{1\times 100} & & \\ & & & & \boldsymbol{1}_{1\times 100} & \\ & & & & & \boldsymbol{1}_{1\times 100} \end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}1\\1\\1\\1\\1\\1\end{bmatrix} \tag{62}
$$

问题三的优化模型为

$$
\begin{aligned}
&\max\ \bar{y}=\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X}\\
&s.t.\ x\in\{0,1\}\ \text{for}\ x\ \text{in}\ \boldsymbol{X}\\
&\qquad \boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}
\end{aligned} \tag{63}
$$

5.1.3 QUBO 模型

对优化模型(63)中的约束$\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$，设惩罚函数

$$
\begin{aligned}
H&=-\alpha(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{X})^T(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{X})\\
&=-\alpha(\boldsymbol{b}^T\boldsymbol{b}-\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b}+\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X})
\end{aligned} \tag{64}
$$

总目标函数为

$$
\begin{aligned}
J&=\bar{y}+H\\
&=\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X}-\alpha(\boldsymbol{b}^T\boldsymbol{b}-\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b}+\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X})\\
&=\boldsymbol{X}^T(\boldsymbol{P}-\alpha\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{X}-\alpha(\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}+\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b})-\alpha\boldsymbol{b}^T\boldsymbol{b}
\end{aligned} \tag{65}
$$

式(65)中$\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}$和$\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b}$两项为标量，故$\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b}=(\boldsymbol{X}^T\boldsymbol{A}^T\boldsymbol{b})^T=\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}$。由式(62)可知，$\boldsymbol{b}^T\boldsymbol{A}$为全 1 行向量。又因为$\boldsymbol{X}$中元素均为 0/1 二值变量，所以

$$
\begin{aligned}
\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{X}&=\sum_{i=1}^{10}x_{mi}+\sum_{j=1}^{10}x_{pi}+\sum_{k=1}^{10}x_{qk}+\sum_{i=1}^{100}u_{mi}+\sum_{j=1}^{100}u_{pi}+\sum_{k=1}^{100}u_{qk}\\
&=\sum_{i=1}^{10}x_{mi}^2+\sum_{j=1}^{10}x_{pi}^2+\sum_{k=1}^{10}x_{qk}^2+\sum_{i=1}^{100}u_{mi}^2+\sum_{j=1}^{100}u_{pi}^2+\sum_{k=1}^{100}u_{qk}^2\\
&=\boldsymbol{X}^T\boldsymbol{I}_{330}\boldsymbol{X}
\end{aligned} \tag{66}
$$

其中$\boldsymbol{I}_{330}$是 330 阶单位矩阵。式(65)变为

$$
\begin{aligned}
J &= \boldsymbol{X}^T(\boldsymbol{P}-\alpha\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{X}-2\alpha\boldsymbol{X}^T\boldsymbol{I}_{330}\boldsymbol{X}-\alpha\boldsymbol{b}^T\boldsymbol{b} \\
&= \boldsymbol{X}^T(\boldsymbol{P}-\alpha\boldsymbol{A}^T\boldsymbol{A}-2\alpha\boldsymbol{I}_{330})\boldsymbol{X}-\alpha\boldsymbol{b}^T\boldsymbol{b} \\
&\triangleq \boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X}+c
\end{aligned}
\tag{67}
$$

其中$\boldsymbol{Q}=\boldsymbol{P}-\alpha\boldsymbol{A}^T\boldsymbol{A}-2\alpha\boldsymbol{I}_{330}$为 QUBO 矩阵，其中$\boldsymbol{P}$和$\boldsymbol{A}$由式(60)(62)给出。与问题 1，2 相同，取$\alpha=30000$。不考虑常数 c 并将问题表示为最小化目标函数的形式，问题三的 QUBO 模型为

$$
\begin{aligned}
&\min\ -\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X} \\
&s.t.\ \ x\in\{0,1\}\ \text{for}\ x\ \text{in}\ \boldsymbol{X}
\end{aligned}
\tag{68}
$$

5.2 模型求解

在前文建立的 QUBO 模型的基础上，利用 MATLAB 中的 OPTI 优化工具箱中的 SCIP 求解器对该模型进行求解。

步骤如下：

(1)代入数据计算得出 Q 矩阵；

```
E0=load('shouru3.mat');
Q0=E0.E3;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;
```

(2)设定参数，调用求解器

```
m = 1000;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

(3) 得到结果

表 5 问题 3 求解结果

最终收入（元）	信用卡选择	阈值选择（通过率，坏账率）
43881	第 8 张	第 2 个（0.88，0.013）
	第 33 张	第 6 个（0.93，0.024）
	第 49 张	第 3 个（0.85，0.001）

采用 DPSO 算法进行建模求解，得到最大收益进化曲线如图 9 所示。

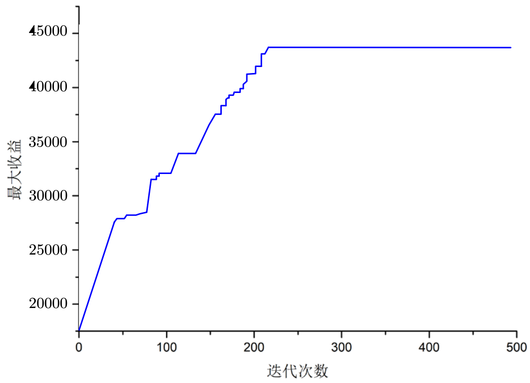

图 9 DPSO 算法最大收益进化曲线

将 DPSO 算法与 QUBO 模型对应的量子退火算法求解结果进行对比分析，如表 6 所示，两种模型计算结果相同，但是 DPSO 算法计算时间远大于为 QUBO 模型下的量子退火求解算法。

表 6 DPSO 模型与 QUBO 模型对比

模型	收益（元）	信用卡选择	阈值（通过率，坏账率）	计算时间
DPSO	43881	第 8 张	第 8 个（0.94，0.043）	59.6842s
		第 33 张	第 1 个（0.72，0.032）	
		第 49 张	第 2 个（0.82，0.013）	
QUBO	43881	第 8 张	第 8 个（0.94，0.043）	2.9622s
		第 33 张	第 1 个（0.72，0.032）	
		第 49 张	第 2 个（0.82，0.013）	

6 模型评价

6.1 优点

(1)本文建模方法在三个问题中，都得到了解析表示的 QUBO 模型。

(2)本文将三重信用卡组合策略的总收益简化表示为一个统一而简洁的形式，由三个单信用卡策略的收益和通过率计算得到，为 QUBO 模型的建立提供了极大的方便。

(3)本文建模方法可以推广到任意数量信用评分卡和阈值的情形。

(4)与离散粒子群等算法求解标准二值优化问题相比，本文方法具有有更高的计算效率。

6.2 缺点

(1)在问题二的模型建立中，为了得到标准 QUBO 模型引入了较多的变量，还存在改进空间。

(2)虽然用题给数据能得到正确结果，但是在对问题简化处理时没有考虑到一些极端和特殊的情况。若要对该模型作推广应用，仍需提高其严谨性。

参考文献

[1] 王宝楠,水恒华,王苏敏等.量子退火理论及其应用综述[J].中国科学:物理学 力学 天文学,2021,51(08):5-17.

[2] Glover F, Kochenberger G, Du Y. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models[J]. 4OR, 2019, 17: 335-371.

[3] Verma A, Lewis M. Penalty and partitioning techniques to improve performance of QUBO solvers[J]. Discrete Optimization, 2022, 44: 100594.

[4] Ding Yongcheng,Chen Xi,Lamata Lucas,Solano Enrique,Sanz Mikel. Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design[J]. SN Computer Science,2021,2(2).

[5] Pelofske Elijah,Hahn Georg,O'Malley Daniel,Djidjev Hristo N & Alexandrov Boian S.(2022).Quantum annealing algorithms for Boolean tensor networks.. Scientific reports(1). doi:10.1038/S41598-022-12611-9.

[6] Yumin Dong & Zhijie Huang.(2020).An Improved Noise Quantum Annealing Method for TSP. International Journal of Theoretical Physics(prepublish). doi:10.1007/s10773-020-04628-5.

附录 1：问题 1 求解代码

```
E0=load('shouru.mat');
Q0=E0.E1;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;

m = 100;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

附录 2：问题 2 求解代码

```
E0=load('shouru2.mat');
Q0=E0.E2;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;

m = 1000;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

附录 3：问题 3 求解代码

```
E0=load('shouru3.mat');
```

```
Q0=E0.E3;
P = 30000;
for i =1:100
    Q1(i,i) = Q0(i,1);
    A(i,i) = 2*P;
end
Q=Q1-P+A;

m = 330;
H = -Q;
f = zeros(m,1);
xtype = char(join( repmat("B",1,m),""));
Opt = opti('qp',H,f,'xtype',xtype)
% Solve the MIQP problem
[x,fval,exitflag,info] = solve(Opt)
```

附录 4：使用的软件工具

编程与计算软件：MATLAB R2020A

论文写作软件：WORD 2021

公式编辑软件：AxMath