

# 一类带指示变量的凸二次优化问题的研究

刘宗丹

北京邮电大学理学院, 北京

收稿日期: 2025年2月6日; 录用日期: 2025年3月27日; 发布日期: 2025年4月8日

---

## 摘要

本文研究一类带指示变量的凸二次优化问题。该问题在多个领域具有广泛应用, 如投资组合选择、信号处理和资源分配等。作为混合整数二次优化问题, 其NP-hard特性使得大规模实例求解面临较大挑战。本文分别提出了三种求解方法: 启发式搜索下降法、精确分支定界法以及两者结合的混合方法。首先, 启发式搜索下降法通过局部优化获得近似解, 显著提高求解速度; 其次, 精确分支定界法通过逐步分解问题空间, 保证了全局最优解的精确性。最后, 结合启发式搜索下降法和分支定界法的优点, 提出了一种新的混合方法, 利用启发式方法提供的初始解加速分支定界法的收敛过程, 减少计算资源消耗, 同时确保解的精确性。实验表明, 启发式搜索下降法在低维和高维问题上均具有显著的计算效率优势。与SCIP和Gurobi求解器相比, 混合方法展现出了更优的求解效率。

---

## 关键词

混合整数二次优化, 启发式搜索下降, 分支定界

---

# The Study of Convex Quadratic Optimization Problems with Indicator Variables

Zongdan Liu

School of Science, Beijing University of Posts and Telecommunications, Beijing

Received: Feb. 6<sup>th</sup>, 2025; accepted: Mar. 27<sup>th</sup>, 2025; published: Apr. 8<sup>th</sup>, 2025

---

## Abstract

This paper studies a class of convex quadratic optimization problems with indicator variables, which have broad applications in various fields such as portfolio selection, signal processing, and resource allocation. As a mixed-integer quadratic optimization problem, its NP-hard nature makes solving large-scale instances challenging. This paper proposes three solution methods: a heuristic search descent method, an exact branch-and-bound method, and a hybrid method combining the

two. First, the heuristic search descent method provides approximate solutions through local optimization, significantly improving solving speed. Second, the exact branch-and-bound method ensures global optimality by gradually decomposing the problem space. Finally, a new hybrid method is introduced, combining the advantages of both approaches by using the initial solution provided by the heuristic method to accelerate the convergence of the branch-and-bound method, reducing computational resource consumption while ensuring solution accuracy. Experimental results show that the heuristic search descent method exhibits significant computational efficiency advantages in both low-dimensional and high-dimensional problems. Compared with the Gurobi and SCIP solvers, the hybrid method demonstrates superior solving efficiency.

## Keywords

Mixed-Integer Quadratic Optimization, Heuristic Search Descent, Branch-and-Bound

Copyright © 2025 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

带指示变量的凸二次优化问题广泛应用于运筹学、工程、金融等多个领域，如投资组合选择、信号与图像处理、最佳子集选择、资源分配以及电力系统等[1]。由于其 NP-hard 特性，在计算资源有限的情况下，求解问题通常面临较大的挑战。常见的求解策略包括分支定界法和外逼近法。分支定界法作为一种传统方法，其基本思路是将原问题拆解为若干个子问题，通过构建分支树逐步逼近最优解。外逼近法通过逐步逼近目标函数的凸包完成求解，其关键在于如何选择和使用切割平面。外逼近法通过逐步添加切割平面，逼近目标函数的凸包，从而将目标问题转化为凸问题，并有效求解大规模问题。近年来，开发了新的外逼近算法，通过从不同的凸松弛中导出切割平面，进一步提升了大规模问题的求解效率。目前市面上有多种求解运筹学和优化问题的求解器，在解决混合整数二次规划问题时，常用的求解器包括 CPLEX、Gurobi 和 SCIP。CPLEX 因其高效的求解算法和强大的大规模问题处理能力，广泛应用于混合整数优化领域，但其高昂的许可证费用以及在处理非凸问题时的局限性仍然是其主要缺点。Gurobi 与 CPLEX 类似，具有卓越的性能和强大的并行计算能力，特别适用于复杂的 MIQP 问题。它同样支持多种编程语言，然而，Gurobi 的许可证费用较高，且在处理某些非凸问题时效果有限。SCIP 作为一款开源求解器，适用于预算有限的学术研究。然而，SCIP 在求解速度上较慢，尤其是在处理大规模问题时，无法与商业求解器相媲美。

本文提出了三种求解方法：启发式搜索下降法、精确分支定界法和两者结合的混合方法。首先，启发式搜索下降法通过局部优化策略，能够快速找到局部最优解，显著提高求解速度。其次，精确分支定界法通过逐步分解问题空间，能够保证全局最优解的精确性，特别适用于精度要求较高的场合。然而，单独使用分支定界法时，计算成本较高，求解时间较长。为了克服这些问题，本文还提出了启发式搜索下降法与分支定界法结合的混合方法。该混合方法利用启发式方法提供的初始解，能够有效加速分支定界法的收敛过程，减少计算资源的消耗，同时确保解的精确性。

## 2. 背景

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2, 3, \dots, n. \end{aligned} \quad (1)$$

考虑一类带指示变量  $z \in \{0,1\}^n$  的凸二次优化问题。其目标函数和约束条件如下：其中  $Q \in R^{n \times n}$  是对称正定矩阵， $a, c \in R^n$  是给定向量。当  $z_i = 0$  时，则  $x_i = 0$ 。当  $z_i = 1$  时，则  $x_i$  取值不受限制。这类问题常出现在信号重建问题，其中  $z \in \{0,1\}^n$  用于保证信号的稀疏性，目标函数则通过最小化重建信号与接收信号之间的误差来优化信号重建的精度。该问题在投资组合优化中也有广泛应用，旨在从多个投资选项(如股票、债券、房地产等)中进行选择，以实现收益最大化与风险最小化的目标。二次目标函数通常用来建模投资组合收益的风险(如方差或协方差)，而二元变量  $z_i = 1$  则表示将资产  $i$  包括在投资组合[2]中。

针对问题(1)，一些学者根据问题二次型中矩阵的结构提出了特定的求解方法。一个典型的特殊情况是问题规模为 1 或者矩阵  $Q$  为对角矩阵，即问题变为完全可分离形式。在这种情况下，问题(1)可以通过透视重构方法转化为凸优化问题。通过选择适当的对角矩阵  $D$  并结合透视重构[3]与分支定界法，可以有效简化优化问题并提高求解效率。当稀疏矩阵  $Q$  是三对角矩阵时，可以通过构造与  $Q$  相关的辅助图[4]，将原问题的求解转化为辅助图中的最短路径问题[5]。在这种情况下，可以获得问题的精确解。对于一般的稀疏矩阵  $Q$ ，可以基于图方法求解目标问题的松弛问题，进而获得问题的近似解。

## 符号说明

$z^0 \in \{0,1\}^n$  表示初始给定的向量。对于任意向量  $z \in \{0,1\}^n$ ， $z_1$  表示向量  $z$  的第一个分量， $z_k$  表示向量  $z$  的第  $k$  个分量。

$x^0 \in R^n$  表示初始给定的向量。对于任意向量  $x \in R^n$ ， $x_1$  表示向量  $x$  的第一个分量， $x_k$  表示向量  $x$  的第  $k$  个分量。

$Q_{i,j,i,j} \in R^{(j-i+1) \times (j-i+1)}$  表示矩阵  $Q$  的行标从第  $i$  行到第  $j$  行，列标从第  $i$  列到第  $j$  列组成的子矩阵。  
 $z_{i,j} \in \{0,1\}^{j-i+1}$  表示向量  $z$  的第  $i$  维到第  $j$  维组成的子向量。 $x_{i,j} \in R^{j-i+1}$  表示向量  $x$  的第  $i$  维到第  $j$  维组成的子向量。

在二维子问题中， $Q_{[i,j]} \in R^{2 \times 2}$  表示二次型中与  $x_i$ 、 $x_j$  有关的系数矩阵， $\tilde{c}_{[i,j]} \in R^2$  表示线性项中与  $x_i$ 、 $x_j$  有关的系数向量， $\tilde{a}_{[i,j]} \in R^2$  表示线性项中与  $z_i$ 、 $z_j$  有关的系数向量。

## 3. 基于启发式搜索下降法和分支定界法的优化求解框架

随着问题规模的不断扩大，优化问题的解空间呈指数级增长，传统的优化方法在求解这些大规模问题时往往需要消耗大量的计算时间和资源。针对这一挑战，本章提出启发式搜索下降法，通过灵活的搜索策略能够在较短的时间内找到局部最优解。而分支定界法则通过系统地划分解空间并对解进行约束，有效地验证启发式方法得到的解是否为全局最优解。最后提出两者结合的混合方法。本章首先介绍了启发式搜索下降法的基本原理与算法设计，重点说明了该方法如何在每一轮迭代中通过选择不同维度进行优化，逐步趋近全局最优解。接着，详细阐述了分支定界法在优化求解中的应用，探讨了如何通过分支、剪枝和定界规则有效缩小搜索空间，提高计算效率，并确保全局最优解的准确性。通过这种结合，能够在保证全局最优解的同时，大幅降低计算开销，加速收敛过程。

### 3.1. 启发式搜索下降法

搜索方向法是一类重要的启发式优化算法，其核心思想是通过合理选择搜索方向，引导算法逐步逼近全局最优解。常用的搜索方向包括梯度方向、牛顿方向以及基于启发式策略确定的其他方向。与基于导数信息的优化方法不同，直接搜索方法不依赖于目标函数的导数信息，而是通过搜索方向逐步逼近最优解。典型的直接搜索方法包括随机搜索法、网格搜索法和模式搜索法等[6]。近年来，搜索方向校正(Search Direction Correction, SDC)作为一种改进的一阶优化方法受到广泛关注。SDC 通过线性组合历史搜

索方向、当前梯度以及归一化梯度方向来更新搜索方向[7]。例如，快速惯性搜索方向校正(Fast Inertial Search Direction Correction, FISC)算法在稀疏优化、逻辑回归和深度学习等领域展现了优异的性能。此外，内点算法通过定义新的搜索方向，在线性优化和半定优化问题中取得了显著成果[8]。在多目标优化领域，搜索方向法通过改进搜索方向的计算方法，进一步提升了算法性能，使得每一步迭代都能有效改善解的质量[9]。混合整数规划问题的解空间通常是离散的，这使得搜索方向法能够选择适当的维度或变量进行局部优化，逐步解决各个子问题。在每一轮迭代中，算法通过降维策略将原问题转化为较小的子问题，并基于子问题的解更新原问题的解。这种降维与局部优化相结合的策略不仅能够加速搜索过程，还能显著提高算法的收敛性，最终趋近于全局最优解。

基于搜索方向法的核心思想，本文提出了一种新的启发式搜索下降法。每次迭代时，首先给定初始点，并计算初始的最优解及其对应的函数值。然后，算法通过将与固定维度无关的维度代入问题(1)，将原问题转化为一个二维子问题并求解，得到新的解，将其扩展回原问题中。在每次迭代后，算法会检查当前解是否满足收敛条件。如果当前解不发生变化，即目标函数值不发生下降，算法记录下连续不变的次数，并跳转到下一步；如果未满足收敛条件，则在固定某些维度的前提下，重新求解问题(1)，得到新的最优解并更新相应的函数值，继续进行优化。每完成一次迭代，算法会更新最优解的不变次数，并检查是否超过  $\frac{n(n-1)}{2}$  次。如果超过该次数，算法将终止并返回当前的最优解；否则，算法将继续进行下一轮迭代。

### 3.1.1. 构建二维子问题

给定初始点  $z^0$ ，求解问题(1)得到当前最优解  $x^0$ 。每一轮迭代中，选择  $z^0$  第  $i$  维和第  $j$  维，作为自由变量，其余维度不变，将问题(1)降维为一个关于连续变量  $x_i$ 、 $x_j$  和二元变量  $z_i$ 、 $z_j$  的问题(2)。

$$\begin{aligned} \min_{x_i, x_j \in R, z_i, z_j \in \{0,1\}} \quad & \frac{1}{2} \begin{bmatrix} x_i x_j \end{bmatrix} Q_{[ij]} \begin{bmatrix} x_i x_j \end{bmatrix}^T + \tilde{c}_{[ij]}^T \begin{bmatrix} x_i x_j \end{bmatrix} + \tilde{a}_{[ij]}^T \begin{bmatrix} z_i z_j \end{bmatrix} \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, x_j (1 - z_j) = 0. \end{aligned} \quad (2)$$

### 3.1.2. 求解二维子问题

求解问题(2)，并将其最优解扩展为原问题的解。对于  $z_i$ 、 $z_j$  的四种可能组合  $(0,0)$ 、 $(0,1)$ 、 $(1,0)$ 、 $(1,1)$ ，分别求解对应的问题(2)。通过求解目标函数的一阶导数，可得显式解：

当  $z_i = 0$ ， $z_j = 0$  时， $x_i = 0$ ， $x_j = 0$ 。

当  $z_i = 0$ ， $z_j = 1$  时， $x_i = 0$ ， $x_j = -\frac{\tilde{c}_j}{Q_{jj}}$ 。

当  $z_i = 1$ ， $z_j = 0$  时， $x_i = -\frac{\tilde{c}_i}{Q_{ii}}$ ， $x_j = 0$ 。

当  $z_i = 0$ ， $z_j = 0$  时， $\begin{bmatrix} x_i x_j \end{bmatrix}^T = -Q_{[ij]}^{-1} \tilde{c}_{[ij]}^T$ 。

### 3.1.3. 新解的扩展方式

通过计算四种组合对应的目标函数值，选择最优解  $(x_i^*, x_j^*, z_i^*, z_j^*)$ 。若存在多个最优解，且最优解包括  $(x_i^0, x_j^0, z_i^0, z_j^0)$ ，优先选择  $(x_i^*, x_j^*, z_i^*, z_j^*) = (x_i^0, x_j^0, z_i^0, z_j^0)$ 。可得如下解：

$$x_{new} = \begin{bmatrix} x_1^0, \dots, x_{i-1}^0, x_i^*, x_{i+1}^0, \dots, x_{j-1}^0, x_j^*, x_{j+1}^0, \dots, x_n^0 \end{bmatrix}$$

$$z_{new} = \begin{bmatrix} z_1^0, \dots, z_{i-1}^0, z_i^*, z_{i+1}^0, \dots, z_{j-1}^0, z_j^*, z_{j+1}^0, \dots, z_n^0 \end{bmatrix}$$

记录新解未发生变化的次数  $k$  :

如果  $z_{new} = z^0$ ，则解未发生改变，令  $k = k + 1$ 。

如果  $z_{new} \neq z^0$ ，说明解发生改变：令  $k = 0$ ，令  $z = z_{new}$ 。重新求解问题(1)，得到新的最优解  $\bar{x}$  和对应的函数值  $\bar{f}$ ，并令  $z^0 = z_{new}$ ， $x^0 = \bar{x}$ 。

### 3.1.4. 维度索引更新原则

第一轮迭代开始时：初始化  $i = 1$ ， $j = 2$ 。选择第一维变量  $z_1^0$  和第二维变量  $z_2^0$  作为自由变量。在每一轮迭代中，算法根据以下规则更新维度索引  $i$ ， $j$ ：

1. 如果  $j < n$ ，则  $j = j + 1$ ，计算新的二维子问题。
2. 如果  $j \geq n$  且  $i < n - 1$ ，则令  $i = i + 1$ ， $j = i + 1$ ，计算新的二维子问题。
3. 如果  $j \geq n$  且  $i \geq n - 1$ ，则令  $i = 1$ ， $j = i + 1$ ，计算新的二维子问题。

### 3.1.5. 算法终止条件

如果  $k \geq \frac{n(n-1)}{2}$ ，即新解未发生变化的次数超过所有二维子问题的组合数  $\frac{n(n-1)}{2}$ ，即目标函数值连续  $\frac{n(n-1)}{2}$  次不发生下降，此时找到最优解，算法终止。

---

### 算法 1 启发式搜索下降法

---

初始化：

1. 给定一个初始  $z^0$ ，求解问题(1)得到解  $x^0$ ，函数值为  $f_0$ 。
2. 参数  $k = 0$ ， $i = 1$ ， $j = 2$ 。

迭代过程：

1. 固定  $z_i$ 、 $z_j$  的四种可能组合  $(0,0)$ 、 $(0,1)$ 、 $(1,0)$ 、 $(1,1)$ ，分别求解对应的问题(2)。得到最优解  $(x_i^*, x_j^*, z_i^*, z_j^*)$ 。经扩展得到新解  $x_{new}$  与  $z_{new}$ 。转步 2。
  2. 如果  $z_{new} = z^0$ ，则转步 3；否则，令  $k = 0$ ， $z = z_{new}$ ，求解问题(1)，得到新的最优解  $\bar{x}$  和对应的函数值  $\bar{f}$ 。并令  $x^0 = \bar{x}$ ， $z^0 = z_{new}$ ， $f_0 = \bar{f}$ ，转步 4。
  3. 更新  $k = k + 1$ 。如果  $k \geq \frac{n(n-1)}{2}$ ，则算法终止，返回  $z^0$ 、 $x^0$  和  $f_0$ ；否则，转步 4。
  4. 更新  $i$  和  $j$ 。如果  $j < n$ ，则令  $j = j + 1$ ，转步 1；如果  $j \geq n$  且  $i < n - 1$ ，则令  $i = i + 1$ ， $j = i + 1$ ，转步 1；否则，令  $i = 1$ ， $j = i + 1$  转步 1。
- 

问题(1)有  $n$  个约束，这意味着在算法的迭代过程中，算法的搜索空间是有限的，从而保证其收敛性。每次迭代算法选择两个自由变量  $x_i$  和  $x_j$  并进行局部优化。每次局部优化的目标函数值能够逐步减小(或者至少不增加)，则该算法有可能逐步趋近于全局最优解。如果目标函数值在连续  $\frac{n(n-1)}{2}$  次迭代中都没有下降，表示已经找到局部最优解，并且无法进一步改善目标函数值，则算法终止。

启发式搜索下降法通过启发式策略逐步接近全局最优解并停在局部最优解，但由于其依赖启发式搜索，所得到的解通常难以直接验证是否为全局最优解。因此，本文提出分支定界法来对启发式搜索下降法所得解进行验证。

## 3.2. 分支定界法

分支定界法是一种广泛应用于组合优化问题中的精确算法，尤其适用于整数规划、0~1 背包问题、旅

行商问题[10]等, 其通过系统地划分解空间并进行界限约束, 逐步排除不可能的解, 从而确保最终找到全局最优解或证明解的不可改进性。通过合理设计分支策略和定界方法, 可以有效减少搜索空间, 从而提高求解效率。例如, 在某些凸非线性整数规划问题中[11], 分支定界法能够通过剪枝策略快速排除大量不可能的解, 从而显著减少计算时间。对于混合整数非线性规划问题[12], 分支定界法结合参数化方法可以有效处理参数在一般位置的优化问题。通过在分支定界树的每个节点进行优化, 算法能够逐步趋近全局最优解。这种方法在处理复杂的混合整数非线性问题时表现出良好的适应性和效率。在使用分支定界法求解优化问题时, 明确分支规则、剪枝规则和定界规则至关重要。这些规则对于有效缩小搜索空间、提高算法效率以及确保全局最优解的获得发挥关键作用。

### 3.2.1. 分支规则

分支规则的选择直接影响分支定界算法的效率及结果的准确性, 我们先不讲解剪枝规则, 不进行剪枝, 仅介绍问题(1)的分支架构。将问题(1)的分解过程用一棵二叉树来表示, 每个节点对应一个子问题, 下面将逐层对问题进行分解。

具体的分支策略如下:

1、第一层求解问题(3), 即在问题(1)的基础上只保留第一个约束。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1. \end{aligned} \quad (3)$$

选择第一个分量  $z_1 \in \{0,1\}$ , 依次取集合中的值将  $z_1$  固定, 得到下列两个子问题:

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1. \\ & z_1 = 0. \end{aligned} \quad (4)$$

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1. \\ & z_1 = 1. \end{aligned} \quad (5)$$

当  $z_1 = 0$ , 此时  $x_1 = 0$ 。问题降维为  $n-1$  维无约束优化问题。通过一阶导数条件求得最优解  $x_{2n}^* = -Q_{2n,2n}^{-1} c_{2n}$ 。当  $z_1 = 1$ , 此时  $x_1$  取值不受限制。问题为  $n$  维无约束优化问题。通过一阶导数条件求得最优解  $x^* = -Q^{-1} c$ 。

2、第二层求解问题(6), 即在问题(1)的基础上只保留前两个约束。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2. \end{aligned} \quad (6)$$

在第一层的基础上, 进一步固定  $z_2$  的值, 即  $z_{12} \in \{[00]^T, [01]^T, [10]^T, [11]^T\}$ , 生成四个子问题。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2. \\ & z_1 = 0, z_2 = 0. \end{aligned} \quad (7)$$

当  $z_{12} = [00]^T$ ，此时  $x_{12} = [00]^T$ 。问题降维为  $n-2$  维无约束优化问题。通过一阶导数条件求得最优解  $x_{3n}^* = -Q_{3n,3n}^{-1} c_{3n}$ 。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2. \\ & z_1 = 0, z_2 = 1. \end{aligned} \quad (8)$$

当  $z_{12} = [01]^T$ ，此时  $x_1 = 0$ ， $x_2$  不受限制。问题降维为  $n-1$  维无约束优化问题。通过一阶导数条件求得最优解  $x_{2n}^* = -Q_{2n,2n}^{-1} c_{2n}$ 。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2. \\ & z_1 = 1, z_2 = 0. \end{aligned} \quad (9)$$

当  $z_{12} = [10]^T$ ，此时  $x_1$  不受限制， $x_2 = 0$ 。问题降维为  $n-1$  维无约束优化问题。通过一阶导数条件求得最优解。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2. \\ & z_1 = 1, z_2 = 1. \end{aligned} \quad (10)$$

当  $z_{12} = [11]^T$ ，此时  $x_1$  和  $x_2$  不受限制。问题为  $n$  维无约束优化问题。通过一阶导数条件求得最优解  $x_{1n}^* = -Q^{-1} c$ 。求解问题(7)、(8)、(9)和(10)等价于求解问题(6)。

3、一般地，第  $k$  层在问题(1)的基础上只保留前  $k$  个约束，枚举  $z_{1:k}$  的所有可能从而生成  $2^k$  个子问题。

$$\begin{aligned} \min_{x \in R^n, z \in \{0,1\}^n} \quad & f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z \\ \text{s.t.} \quad & x_i (1 - z_i) = 0, i = 1, 2, \dots, k. \end{aligned} \quad (11)$$

问题的分解过程用一棵二叉树来表示，每个节点对应一个子问题，分支表示固定某个  $z_i$  的值(0 或 1)，从而将问题分解为更小的子问题。每一层的节点数目将是上一层的两倍。通过这种方式，算法逐层对问题空间进行分解，从而逐渐逼近最优解。

求解问题(4)等价于求解问题(7)和(8)。如果通过某些剪枝规则剪掉问题(4)，那么问题(7)和(8)乃至其对应的下层孩子结点的问题都被剪掉。具体剪枝规则由下一节阐述。通过递归构建二叉树，可以枚举所有可能的  $z$  组合。通过剪枝规则，可以避免计算那些不可能改善当前最优解的子问题。为了有效地表示分支定界法中的子问题结构，采用层次化的方式来组织每一层的子问题集合。

设第  $i$  层的子问题集合  $S_i$ ，其中每个子问题  $S_{ik}$  表示第  $i$  层的第  $k$  个子问题。形式上，第  $i$  层的子问题集合可以表示为： $S_i = \{S_{i1}, S_{i2}, S_{i3}, \dots, S_{im_i}\}$ 。

其中， $m_i$  表示第  $i$  层的子问题数量， $S_{ik}$  为第  $i$  层第  $k$  个子问题。每个子问题  $S_{ik}$  对应一个目标函数值  $f(S_{ik})$ ，根据以下规则更新上下界。

### 3.2.2. 剪枝规则

剪枝规则在分支定界过程中发挥着至关重要的作用。其核心思想是通过分析当前子问题的上下界，

剔除那些不可能包含最优解的子问题，从而减少搜索空间，节省计算资源并提高效率。若某子问题  $S_{ik}$  的目标函数值  $f(S_{ik})$  大于当前的上界 UB，则说明该子问题不可能给出比当前最优解更好的解，因此可以将其剪枝，即不再继续处理该子问题及其后续分支。剪枝规则是分支定界法的关键组成部分，通过合理的剪枝策略有效地缩小搜索空间，避免无意义的计算，从而保证最终能够找到全局最优解。

### 3.2.3. 定界规则

上界更新规则：如果某个子问题的  $S_{ik}$  的最优解满足问题(1)的所有约束，且  $f(S_{ik})$  小于当前上界 UB，则更新当前上界  $UB = f(S_{ik})$ 。

下界更新规则：令每一层的下界集  $down\_set = \{f(S_{i1}), f(S_{i2}), f(S_{i3}), \dots, f(S_{im_i})\}$ ，则当前下界 LB 更新为：

$$LB = \min\{f(S_{i1}), f(S_{i2}), f(S_{i3}), \dots, f(S_{im_i})\}。$$

分支定界法的终止条件：

1. 所有子问题均已处理完毕；
2. 所有剩余的子问题都无法改进当前的上下界。

在满足上述任一条件时，当前的下界 LB 或者当前找到的最优解即为最终解，即  $UB - LB < \varepsilon (\varepsilon > 0)$  时算法终止。

### 3.2.4. 分支定界算法框架

---

#### 算法 2 分支定界法

---

初始化：

1. 给定初始可行解  $x^0$ ，初始上界  $UB = f_0$ ，误差  $\varepsilon = 1 \times 10^{-6}$ 。
2. 参数  $k = 1$  (表示当前处理的约束层数)， $down\_set = \emptyset$ ， $z\_set = \emptyset$  (存储分支的  $z$  值)， $z\_set1 = \emptyset$  临时存储当前层生成的新分支)。
3.  $\bar{x}$  表示当前最优解， $\bar{x} = x^0$ 。 $\bar{f}$  表示当前最优函数值， $\bar{f} = UB$ 。初始下界：

$$LB = \min_{x \in R^n, z \in \{0,1\}^n} f(x, z) = \frac{1}{2} x^T Q x + c^T x + a^T z$$

迭代过程：

1. 计算仅含前  $k$  个约束的问题(1)，枚举  $z_{ik}$  所有情况。逐个求解固定  $z_{ik}$  对应的无约束问题，得到最优解  $x_{new}$  和最优函数值  $f_{new}$ 。将  $f_{new}$  添加到集合  $down\_set$  中。  
如果  $x_{new}$  满足约束且  $f_{new} < \bar{f}$ ，更新当前最优解  $\bar{x} = x_{new}$ ，当前最优值  $\bar{f} = f_{new}$ ， $UB = \bar{f}$ 。  
如果  $UB - LB < \varepsilon$ ，算法终止，输出  $\bar{x}$  和  $\bar{f}$ 。如果  $x_{new}$  不满足约束，则将  $z_{ik}$  末尾分别补 0、1 生成新的  $z_{ik+1}$  值并添加到集合  $z\_set1$  中。转步 2。
  2. 令  $LB = \min down\_set$ ， $z\_set = z\_set1$ ， $z\_set1 = \emptyset$ 。令  $k = k + 1$ 。如果  $k > n$  或  $UB - LB < \varepsilon$ ，则算法终止，输出  $\bar{x}$  和  $\bar{f}$ 。否则，转步 1。
- 

分支定界算法通过系统地枚举所有可能的解空间，并结合剪枝规则，能够保证找到全局最优解。由于算法在每一步都会更新上界 UB 和下界 LB，并且最终满足  $UB - LB < \varepsilon$ ，因此算法的收敛性得到保证。由于问题(1)的约束数量是  $n$  个，且每次迭代增加一个约束，算法最多进行  $n$  次迭代。在每次迭代中，枚举的  $z_{ik}$  组合数量是  $2^k$  个，算法复杂度呈指数级。但总的分支个数有限，算法会在有限步终止。

### 3.3. 基于启发式搜索下降法的分支定界法

结合启发式搜索下降法与分支定界法，可以显著提升大规模优化问题的求解效率。启发式方法提供

的初始解不仅能够有效确定合理的上界，还能在一定程度上帮助分支定界法缩小搜索空间，加速算法的收敛速度，并提高求解精度。此种结合方式使得分支定界法在保证全局最优解的同时，能够以更低的计算成本实现问题的快速而精准求解。

#### 4. 数值实验和结论

本研究通过数值实验证了启发式搜索下降法在不同维度问题上的性能表现，并与其他方法(包括混合方法、分支定界法、SCIP 和 Gurobi)进行了对比分析。

实验中的每个问题实例均由随机生成对称正定系数矩阵  $Q$ ，系数向量  $c$ 、 $a$  所得，且每个维度随机生成 100 个算例。本研究首先针对 10 维至 150 维的多个问题实例进行了数值实验，旨在验证启发式搜索下降法快速求解不同维度问题的优异性能。然后，对比分析不同方法在 3 维至 15 维度问题上的平均求解时间和总求解正确率。

**Table 1.** Search-based heuristic descent method average solving time (Seconds)

**表 1.** 启发式搜索下降法平均求解时间(秒)

维度	启发式搜索下降法
10	0.005
30	0.047
50	0.151
70	0.319
90	0.559
110	1.276
130	1.924
150	2.776

**Table 2.** Average solving time for different methods (Seconds)

**表 2.** 不同方法平均求解时间(秒)

维度	启发式搜索下降法	混合方法	分支定界法	SCIP	Gurobi
3	0.0003	0.0003	0.0002	31.6260	0.0238
5	0.0008	0.0010	0.0005	46.6370	0.0411
7	0.0030	0.0050	0.0024	48.0953	0.0435
9	0.0059	0.0130	0.0175	52.1391	0.0512
11	0.0045	0.0288	0.1098	56.5138	0.0548
13	0.0069	0.1365	0.5998	56.5818	0.0591
15	0.0175	0.4905	3.1246	59.5619	0.0604

**表 1** 实验结果表明：启发式搜索下降法在低维和高维问题上均表现出优异的计算效率。在 10 至 150 维的问题上，其求解时间随维度增加的速度较慢，展现了良好的扩展性。**表 2** 实验结果表明：特别是在低维问题(3 至 15 维)上，启发式搜索下降法的求解时间显著优于其他方法。这表明该方法适合需要快速求解的场景。

结合**表 2**、**表 3** 实验结果可以看出：混合方法和分支定界法在低维问题上表现出极高的求解正确率

(100%), 但其求解时间随维度增加而显著上升, 尤其是在高维问题上计算效率较低。因此, 这两种方法更适合对求解准确性要求极高且问题规模较小的场景。SCIP 在所有测试维度上均能保证 100% 的求解正确率, 但其求解时间显著高于其他方法, 尤其是在低维问题上表现尤为明显。因此, SCIP 更适合在计算资源充足且对准确性要求极高的场景中使用。Gurobi 在求解时间和准确性之间取得了较好的平衡, 但在低维问题上的求解时间仍高于启发式搜索下降法, 且其求解正确率略低于混合方法和分支定界法。因此, Gurobi 适合在需要兼顾计算效率和准确性的场景中使用。

**Table 3.** Accuracy of solutions for different methods

**表 3. 不同方法求解正确率**

维度	启发式搜索下降法	混合方法	分支定界法	SCIP	Gurobi
3	100%	100%	100%	100%	89%
5	94%	100%	100%	100%	91%
7	95%	100%	100%	100%	94%
9	94%	100%	100%	100%	92%
11	96%	100%	100%	100%	92%
13	93%	100%	100%	100%	95%
15	94%	100%	100%	100%	93%

## 5. 结论

本文提出了三种求解方法: 启发式搜索下降法、精确分支定界法和两者结合的混合方法, 并通过数值实验对这些方法进行了性能验证和比较分析。实验结果表明, 启发式搜索下降法在低维和高维问题上均具有显著的计算效率优势, 尤其在 10 至 150 维的测试中, 求解时间增长平缓, 扩展性良好。精确分支定界法和混合方法在低维问题上均表现出 100% 的求解正确率, 但其求解时间随维度增加而显著上升, 高维问题中计算效率相对较低。适合对求解准确性要求极高且问题规模较小的场景。混合方法通过启发式搜索下降法提供的初始解, 加速了分支定界法的收敛过程, 在全局最优解的同时, 提高了求解效率, 尤其在中小规模问题中表现突出。SCIP 算法在所有测试维度上均能够保证 100% 的求解正确率, 但由于其计算时间较长, 尤其是在低维问题中, 表现不如启发式搜索下降法。因此, SCIP 更适用于计算资源充足、对准确性要求极高的场景。Gurobi 则在求解时间与准确性之间达到了较好的平衡, 尽管在低维问题上其求解时间仍高于启发式搜索下降法, 但其在需要兼顾计算效率和准确性的应用场景中具有一定优势。综上所述, 本研究验证了启发式搜索下降法的优越性, 并展示了混合方法在提高求解效率的同时, 仍能确保获得全局最优解。未来的研究可进一步优化这些方法, 提升其在大规模和复杂问题中的应用性能。

## 参考文献

- [1] Tillmann, A.M., Bienstock, D., Lodi, A. and Schwartz, A. (2024) Cardinality Minimization, Constraints, and Regularization: A Survey. *SIAM Review*, **66**, 403-477. <https://doi.org/10.1137/21m142770x>
- [2] Bienstock, D. (1996) Computational Study of a Family of Mixed-Integer Quadratic Programming Problems. *Mathematical Programming*, **74**, 121-140. <https://doi.org/10.1007/bf02592208>
- [3] Ceria, S. and Soares, J. (1999) Convex Programming for Disjunctive Convex Optimization. *Mathematical Programming*, **86**, 595-614. <https://doi.org/10.1007/s101070050106>
- [4] Manzour, H., Küçükyavuz, S., Wu, H. and Shojaie, A. (2021) Integer Programming for Learning Directed Acyclic Graphs from Continuous Data. *Informs Journal on Optimization*, **3**, 46-73. <https://doi.org/10.1287/ijoo.2019.0040>

- 
- [5] Liu, P., Fattah, S., Gómez, A. and Küçükayuz, S. (2022) A Graph-Based Decomposition Method for Convex Quadratic Optimization with Indicators. *Mathematical Programming*, **200**, 669-701. <https://doi.org/10.1007/s10107-022-01845-0>
  - [6] Powell, M.J.D. (1998) Direct Search Algorithms for Optimization Calculations. *Acta Numerica*, **7**, 287-336. <https://doi.org/10.1017/s0962492900002841>
  - [7] Wang, Y., Jia, Z. and Wen, Z. (2021) Search Direction Correction with Normalized Gradient Makes First-Order Methods Faster. *SIAM Journal on Scientific Computing*, **43**, A3184-A3211. <https://doi.org/10.1137/20m1335480>
  - [8] Zaoui, B., Benterki, D. and Khelladi, S. (2024) Complexity Analysis and Numerical Implementation of a New Interior-Point Algorithm for Semidefinite Optimization. *Operations Research Letters*, **57**, Article 107192. <https://doi.org/10.1016/j.orl.2024.107192>
  - [9] Mohammadi, A. and Sheikholeslam, F. (2023) Intelligent Optimization: Literature Review and State-of-the-Art Algorithms (1965-2022). *Engineering Applications of Artificial Intelligence*, **126**, Article 106959. <https://doi.org/10.1016/j.engappai.2023.106959>
  - [10] Liang, H., Wang, S., Li, H., Zhou, L., Zhang, X. and Wang, S. (2024) BiGNN: Bipartite Graph Neural Network with Attention Mechanism for Solving Multiple Traveling Salesman Problems in Urban Logistics. *International Journal of Applied Earth Observation and Geoinformation*, **129**, Article 103863. <https://doi.org/10.1016/j.jag.2024.103863>
  - [11] Salahi, M., Ansary Karbasy, S., Almaadeed, T.A. and Hamdi, A. (2024) Improved Branch and Bound Algorithm and an Interpolation-Based Search Algorithm for Quadratic Minimization with One Negative Eigenvalue. *Optimization Letters*, **19**, 527-549. <https://doi.org/10.1007/s11590-024-02149-2>
  - [12] Gupta, O.K. and Ravindran, A. (1985) Branch and Bound Experiments in Convex Nonlinear Integer Programming. *Management Science*, **31**, 1533-1546. <https://doi.org/10.1287/mnsc.31.12.1533>