

# 量子桥分析I：关于制定和使用QUBO模型的教程

Fred Glover<sup>1</sup>, Gary Kochenberger<sup>2</sup>, Yu Du<sup>2</sup>

## 摘要

量子桥分析一般涉及经典-量子混合计算的方法和系统，更特别的是致力于开发连接经典和量子计算的工具，以获得它们在当前联盟的好处，并在未来使量子计算的实际应用得到加强。

这是两部分教程中的第一部分，调查了量子桥分析及其应用的关键要素，重点是用数字图解来补充模型。在第一部分（本文）中，我们重点介绍了二次无约束二元优化（QUBO）模型，这是目前在量子计算领域应用最广泛的优化模型，它统一了丰富的组合优化问题。

**关键词** 二元无约束二元优化(QUBO) - 量子计算 - 量子桥分析 - 组合优化

---

<sup>1</sup>EECE, College of Engineering and Applied Science, University of Colorado, Boulder, CO 80302 USA [fred.glover@colorado.edu](mailto:fred.glover@colorado.edu)

<sup>2</sup>College of Business, University of Colorado at Denver, Denver, CO 80217 USA, [gary.kochenberger@ucdenver.edu](mailto:gary.kochenberger@ucdenver.edu); [yu.du@ucdenver.edu](mailto:yu.du@ucdenver.edu)

## 第1节：简介

组合优化（CO）领域是优化领域中最重要领域之一，在每个行业都有实际应用，包括私营和公共部门。它也是运筹学、计算机科学和分析学研究团体所追求的最活跃的研究领域之一，他们致力于设计和测试解决现实世界中的CO问题的新方法。

一般来说，这些问题涉及到在必须做出大量是/否决定的环境中做出明智的选择，而每一组决定都会产生一个相应的目标函数值--比如成本或利润值。在这些环境中找到好的解决方案是非常困难的。传统的方法是由分析员开发一种适合手头问题的数学结构的解决算法。虽然这种方法在某些问题的设置中产生了良好的效果，但它的缺点是在实践中产生的应用的多样性要求创造出多种多样的解决技术，每一种技术在其最初的预期用途之外的应用都是有限的。

近年来，我们发现，正如Kochenberger等人（2014年）和Anthony等人（2017年）等研究中所记载的那样，一种被称为QUBO的数学表述（Quadratic Unconstrained Binary Optimization problem的缩写）可以包容工业、科学和政府中的各种重要CO问题。通过易于应用的特殊重构技术，QUBO求解器的力量一旦被放入QUBO框架，就可以用来有效解决许多重要问题。

QUBO模型已经成为被称为量子退火和富士通数字退火的量子计算领域的基础，并成为神经形态计算的研究对象。通过这些联系，QUBO模型是D-Wave系统公司开发的量子计算机和IBM公司开发的神经形态计算机所进行的实验的核心。商业领域的IBM、谷歌、亚马逊、微软、D-Wave和洛克希德-马丁公司，公共领域的洛斯阿拉莫斯国家实验室、橡树岭国家实验室、劳伦斯-利弗莫尔国家实验室和美国宇航局艾姆斯研究中心等组织正在探索这些将QUBO模型与量子计算联系起来的新发现的结果。经典和量子计算界都在积累计算经验，这不仅突出了QUBO模型的潜力，也突出了它作为传统建模和解决方案方法的替代方案的有效性。

与量子桥分析的联系来自于在这些发展的基础上弥合经典和量子计算方法和技术之间的差距所要取得的收益。正如美国国家科学院、工程院发布的《2019年共识研究报告》中所强调的，题为《量子计算：正如美国国家科学、工程和医学研究院（<https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects>）在2019年题为《量子计算：进展与前景》的共识研究报告中所强调的那样，量子计算在未来十年内仍将处于起步阶段，而在这期间

"制定一个以开发近期量子计算商业应用为目标的研发计划，对该领域的健康发展至关重要。"该报告进一步指出，这样的计划将依赖于开发 "混合经典-量子技术"。作为这些经典-量子混合技术的基础和实现的创新是量子桥分析的重点，并在很大程度上借鉴了QUBO模型的灵感。

正如Lucas ( 2014 ) 的论文中所强调的那样，QUBO模型可以被证明等同于在物理学中发挥重要作用的Ising模型，这增强了QUBO模型在组合优化中包含许多模型的能力的意义。因此，最先进的QUBO求解方法所有有效解决的广泛的优化问题，也加入了物理学应用中出现的重要问题领域。

接下来的章节中提供的材料通过一系列明确的例子说明了将重要的优化问题重新表述为QUBO模型的过程。这些例子总的来说突出了QUBO模型的应用范围。我们揭示了以一种不同于优化界经典采用的线性模型的形式对广泛的问题进行建模的意外优势。我们展示了在实践出现的许多不同类型的约束关系是如何以一种非常自然的方式用惩罚函数体现在 "无约束 "的QUBO公式中的，产生精确的模型表示，与习惯上使用惩罚函数产生的近似表示相反。产生这种模型的每一步都通过简单的数值例子进行了详细说明，以强调在许多情况下使用QUBO模型的便利。作为其中的一部分，我们提供了一些技术，这些技术可以用来将各种起初可能不适合无约束二元优化结构的问题重塑为一个等价的QUBO模型。我们还描述了最近解决QUBO模型的创新，为整合经典和量子计算以及在机器学习中应用这些模型提供了一条肥沃的途径。

正如Kochenberger和Glover ( 2006 ) 所指出的，QUBO模型包含了以下重要的优化问题：

- 二次方作业问题
- 资本预算问题
- 多重结包问题
- 任务分配问题 ( 分布式计算机系统 )
- 最大多样性问题
- P-中位数问题
- 不对称的分配问题
- 对称分配问题
- 侧面限制性分配问题
- 二次方Knapsack问题

- 约束满足问题(CSP)
- 离散断层成像问题
- 集体分割问题
- 集合包装问题
- 仓库位置问题
- 最大夹层问题
- 最大独立集问题
- 最大切割问题
- 图形着色问题
- 数字分割问题
- 线性排序问题
- 悬崖式分割问题
- SAT问题

此类应用的细节在Kochenberger等人（2014）中得到了更充分的阐述。

在下面的发展中，我们描述了一些方法，这些方法使得在QUBO框架中对这些和许多其他类型的问题进行建模成为可能，并提供了有关将QUBO与机器学习和量子计算联系起来的最新发展的信息。

## **基本QUBO问题公式**

我们现在给出了QUBO模型的正式定义，其意义将通过数字例子更加明确，这些例子让我们了解到QUBO的各种实际应用。

定义：QUBO模型由优化问题表示：

$$\text{QUBO：最小化/最大化 } \mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i,j} Q_{ij} x_i x_j$$

其中 $\mathbf{x}$ 是一个二元决策变量的向量， $\mathbf{Q}$ 是一个常数的正方形矩阵。

通常假设 $\mathbf{Q}$ 矩阵是对称的或者是上三角形式的，这可以在不丧失一般性的情况下简单地实现，具体如下：

对称形式：对于所有的 $i$ 和 $j$ ，除了 $i=j$ ，用  $(Q_{ij} + Q_{ji})/2$  替换  $Q_{ij}$ 。

上三角形式：对于所有 $i$ 和 $j$ 的 $i > j$ ，用  $Q_{ij} + Q_{ji}$  替换  $Q_{ij}$ 。然后用0替换所有 $Q_{ij}$ ，因为 $i < j$ 。（如果矩阵已经是对称的，这只是将主对角线以上的 $Q_{ij}$ 值加倍，然后将主对角线以下的所有值设为0）。

在下面几节给出的例子中，我们将使用完整的、对称的 $\mathbf{Q}$ 矩阵，而不是采用"上三角形式"。

对QUBO模型的形式分类及其解决方案进行评论：QUBO模型属于一类已知为NP-hard的问题。其实际意义在于，旨在寻找“最优”解决方案的精确求解器（如商业化的CPLEX和Gurobi求解器）很可能是不成功的，除非是非常小的问题实例。使用这样的方法，现实大小的问题可以运行数天甚至数周而不产生高质量的解决方案。幸运的是，正如我们在接下来的章节中所披露的，通过使用现代元启发式方法，正在取得令人印象深刻的成功，这些方法被设计用来在少量的计算机时间内找到高质量但不一定是最优的解决方案。这些方法为连接经典和量子计算开辟了宝贵的可能性。

## 第二部分：说明性的例子和定义

在介绍常见的实际应用之前，我们首先给出例子和定义，以奠定基础，更好地看到这些应用如何以QUBO的形式铸造。

首先，考虑优化问题

$$\text{尽量减少 } y = -5x_1 - 3x_2 - 8x_3 - 6x_4 + 4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$$

其中的变量， $x_j$ ，是二进制的。我们可以做几个观察：

1. 要最小化的函数是一个二元变量的二次函数，有一个线性部分

$$-5x_1 - 3x_2 - 8x_3 - 6x_4 \text{ 和一个二次函数部分 } 4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$$

2. 由于二元变量满足  $x_j = x_j^2$ ，线性部分可以写成

$$-5x_1^2 - 3x_2^2 - 8x_3^2 - 6x_4^2$$

3. 然后我们可以用下面的矩阵形式重写这个模型：

$$\text{尽量减少 } y = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} \begin{pmatrix} -5 & 2 & 4 & 0 \\ 2 & -3 & 8 & 0 \\ 4 & 8 & 1 & 5 \\ 0 & 0 & 5 & -6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

4. 反过来，这可以用第1节中介绍的矩阵符号写为

$$\text{尽量减少 } y = x Q x^t$$

其中 $x$ 是一个二元变量的列向量。注意，原始线性项的系数出现在 $Q$ 矩阵的主对角线上。在这种情况下， $Q$ 是围绕主对角线对称的，不需要用第1节中的方法来修改系数。

5. 除了对决策变量的0/1限制外，QUBO是一个无约束的模型，所有问题数据都包含在 $Q$ 矩阵中。这些特点使得QUBO模型作为组合优化问题的建模框架特别有吸引力，为经典的约束性表示提供了一个新的替代方案。

6. 上述(3)中模型的解是：：
$$y = -11, \quad x_1 = x_4 = 1, \quad x_2 = x_3 = 0。$$

#### 备注：

- 如前所述，规定 $Q$ 是围绕主对角线对称的，并不限制该模型的通用性。
- 正如之前所强调的，各种优化问题都可以自然而然地被表述为QUBO模型的一个实例并得到解决。此外，许多其他看起来与QUBO问题无关的问题也可以被重新表述为QUBO模型。我们在下面的章节中说明QUBO模型的这一特殊功能。

### 第3节：天然QUBO配方

如前所述，有几个重要的问题自然属于QUBO类。为了说明这些情况，我们提供了两个重要应用的例子，这些应用的公式自然地采用了QUBO模型的形式。

#### 3.1 数字分割问题

在本说明的书目部分，引用了许多关于“数字分割”问题的应用。这个问题的一个常见版本是将一组数字划分为两个子集，使子集之和尽可能地接近对方。我们将这个问题建模为一个QUBO实例，如下所示：

考虑一组数字 $S = \{s_1, s_2, \dots, s_m\}$ 。如果 $s_j$ 被分配到子集1，则设 $x_j = 1$ ；0  
否则。那么子集1的总和为： $sum_1 = \sum_{j=1}^m s_j x_j$ ，子集2的总和为  
由 $sum_2 = \sum_{j=1}^m s_j - sum_1$ 。那么，两者之差就是

$$\sum_{i,j=1}^m s_i s_j - 2 \sum_{j=1}^m s_j x_j = c - 2 \sum_{j=1}^m s_j x_j$$

我们通过最小化这一差异来实现最小化的目标。

$$diff^2 = \left( c - 2 \sum_{j=1}^m s_j x_j \right)^2 = c^2 + 4x^t Q x$$

其中

$$Q_{ij} = s_i (s_j - c) = s_i s_j$$

抛开加法和乘法常数，我们的QUBO优化问题变成了：

$$QUBO : \min y = x^t Q x$$

其中，Q矩阵是用 $Q_{ii}$ 和 $Q_{ij}$ ，如上定义。

**数学实例：考虑八个数字的集合**

$$S = \{25, 7, 13, 31, 42, 17, 21, 10\}$$

通过上述发展，我们有 $2^8 = 27,556$ ，等效的QUBO问题是

$\min Q = x^t Q x$  with

$$Q = \begin{bmatrix} -3525 & 175 & 325 & 775 & 1050 & 425 & 525 & 250 \\ 175 & -1113 & 91 & 217 & 294 & 119 & 147 & 70 \\ 325 & 91 & -1989 & 403 & 546 & 221 & 273 & 130 \\ 775 & 217 & 403 & -4185 & 1302 & 527 & 651 & 310 \\ 1050 & 294 & 546 & 1302 & -5208 & 714 & 882 & 420 \\ 425 & 119 & 221 & 527 & 714 & -2533 & 357 & 170 \\ 525 & 147 & 273 & 651 & 882 & 357 & -3045 & 210 \\ 250 & 70 & 130 & 310 & 420 & 170 & 210 & -1560 \end{bmatrix}$$

求解QUBO得到 $x = (0,0,0,1,1,0,0,1)$ ，其中 $Q = -6889$ ，得到完全匹配的和等于83。这里采用的发展可以扩展到解决其他形式的数字分割问题，包括数字必须被分割成三个或更多子集的问题，如Alidaee, et.al. (2005)所讨论的。

### 3.2 最大切割问题

最大切割问题是组合优化中最著名的问题之一。给定一个有顶点集V和边集E的无向图G(V, E)，最大切割问题旨在将V划分为两组，使两组之间的边数（被认为是被切割的）尽可能大。

我们可以通过引入二元变量来模拟这个问题，如果顶点j在一个集合中，则满足 $x_j = 1$ ；如果在另一个集合中，则满足 $x_j = 0$ 。将切割视为切断连接两个集合的边，使边的端点位于不同的顶点集合，数量 $x_i + x_j - 2x_{ij}$ 确定边 $(i, j)$ 是否在切割中。也就是说，如果 $(x_i + x_j - 2x_{ij})$ 等于1，那么正是 $x_i$ 和 $x_j$ 等于1，这意味着边 $(i, j)$ 是在切割中。否则 $(x_i + x_j - 2x_{ij})$ 等于零，该边不在切割范围内。

因此，切割中的边数最大化的问题可以表述为

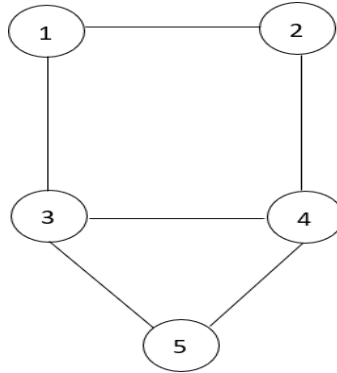
$$\text{最大化 } y = \sum_{(i,j) \in E} (x_i + x_j - 2x_{ij})$$

的一个实例，它是

$$QUBO : \max y = x Q x^t$$

线性项决定了Q的主对角线上的元素，二次项决定了非对角线上的元素。关于QUBO和Max Cut问题的进一步讨论，见Boros和Hammer（1991，2002）和Kochenberger等人（2013）。

**数学实例：**为了说明最大切割问题，请考虑以下有5个顶点和6条边的无向图。



明确考虑到图中的所有边，可以得到以下表述：

$$\begin{aligned} \text{Maximize } y = & (x_1 + x_2 - 2x_{12}) + (x_1 + x_3 - 2x_{13}) + (x_2 + x_4 - 2x_{24}) \\ & + (x_3 + x_4 - 2x_{34}) + (x_3 + x_5 - 2x_{35}) + (x_4 + x_5 - 2x_{45}) \end{aligned}$$



或

$$\max y = 2x_1 + 2x_2 + 3x_3 + 3x_4 + 2x_5 - 2x_1x_2 - 2x_1x_3 - 2x_2x_4 - 2x_3x_4 - 2x_3x_5 - 2x_4x_5$$

这采取了所需的形式

$$QUBO : \max y = x Q x^t$$

通过将对称的Q矩阵写成：

$$Q = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

解决这个QUBO模型可以得到  $x = (0, 1, 1, 0, 0)$ 。因此，顶点2和3在一组，顶点1、4和5在另一组，最大切割值为5。

在上述例子中，问题特征直接导致了QUBO形式的优化问题。如前所述，许多其他问题需要 "重铸" 来创建所需的QUBO形式。我们在下一节介绍一种广泛使用的这种重铸形式。

#### **第4节：使用已知的惩罚措施创建QUBO模型**

到目前为止，QUBO模型的 "自然形式" 除了要求变量是二进制的以外，不包含其他约束。然而，到目前为止，最大数量的相关问题包括额外的约束条件，这些约束条件在优化器寻找好的解决方案时必须得到满足。许多这样的约束模型可以有效地被重新表述为QUBO模型，方法是在目标函数中引入二次惩罚，作为经典意义上明确施加约束的替代方法。引入的惩罚是为了使原始约束对求解过程的影响可以通过优化器的自然功能来实现，因为它寻找的是避免产生惩罚的解决方案。也就是说，惩罚的制定是为了使可行的解决方案等于零，不可行的解决方案等于一些正的惩罚金额。对于最小化问题，这些惩罚被添加到创建一个要最小化的增强的目标函数。如果惩罚项可以被驱动为零，那么增强的目标函数就成为要最小化的原始函数。

对于某些类型的约束条件，对建立QUBO模型有用的二次惩罚是事先知道的，并且可以随时用于将一个给定的约束问题转化为QUBO模型。下表给出了一些经常遇到的约束条件的这种惩罚的例子。请注意，在表中，所有的变量都是二进制的，而

参数P是一个正的、标量的惩罚值。这个值必须选得足够大，以确保惩罚项确实等同于经典约束，但在实践中，可接受的P值通常很容易指定。我们将在后面更深入地讨论这个问题。

经典的制约因素	相当的处罚
$x + y \leq 1$	$P(xy)$
$x + y \geq 1$	$P(1 - x - y + xy)$
$x + y = 1$	$P(1 - x - y + 2xy)$
$x \leq y$	$P(x - xy)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$
$x = y$	$P(x + y - 2xy)$

一些已知的限制/惩罚对表

为了说明主要观点，考虑一个传统的约束问题的形式：

$$\begin{aligned} & \min f(x) \\ & \text{受制于约束条件} \\ & x_1 + x_2 \leq 1 \end{aligned}$$

其中 $x_1$ 和 $x_2$ 是二进制变量。请注意，这个约束条件允许选择任何一个或任何一个x变量。它明确地排除了两个都被选择（即两个都不能被设置为1）。

从上表的第一行，我们可以看到，与我们的约束条件相对应的二次惩罚是

$$Px_1x_2$$

其中P是一个正标量。对于选择足够大的P，无约束的问题是

$$\min f(x) + Px_1x_2$$

具有与原始约束问题相同的最优解。如果 $f(x)$ 是线性或二次的，那么这个无约束模型将是一个QUBO模型的形式。在我们现在的例子中，任何试图最小化 $f(x)$ 的优化器都会倾向于避免 $x_1$ 和 $x_2$ 都等于1的解决方案，否则会在目标函数中增加一个很大的正数。也就是说，目标函数会对不可行的解决方案产生惩罚。Pardalos和Xue(1999)已经在最大集群和相关问题上有效地使用了这种简单的惩罚。

## 4.1 最小顶点覆盖 (MVC) 问题

在第3.2节中，我们看到QUBO模型如何被用来表示著名的Max Cut问题。这里我们考虑另一个著名的图上优化问题，即最小顶点覆盖问题。给定一个具有顶点集 $V$ 和边集 $E$ 的无向图，顶点覆盖是一个顶点（节点）的子集，使得图中的每条边至少与该子集中的一个顶点发生关系。最小顶点覆盖问题旨在寻找一个子集中顶点数量最少的覆盖。

MVC的标准优化模型可以表述如下。设 $x_j = 1$ ，如果顶点 $j$ 在覆盖区（即在子集中）， $x_j = 0$ ，否则。那么这个问题的标准约束、线性0/1优化模型为：

$$\begin{aligned} & \text{尽量减} && \sum_{j \in V} x_j \\ & \text{少} && \\ & \text{受制于} && \\ & && x_i + x_j \geq 1 \text{ 对于所有 } (i, j) \in E \end{aligned}$$

请注意，约束条件确保每条边的至少一个端点在覆盖范围内，目标函数寻求使用最少的顶点数量找到覆盖。还要注意的，我们对图中的每条边都有一个约束，这意味着即使对于规模不大的图，我们也可以有很多约束。在等效的QUBO模型中，每个约束条件都可以通过给目标函数增加一个惩罚来施加。

参照我们的上表，我们可以看到，标准MVC模型中的约束可以用一个形式为 $\square (1 - x_i - x_j + x_i x_j)$ 的惩罚来表示。因此，MVC的无约束模型的替代方案是

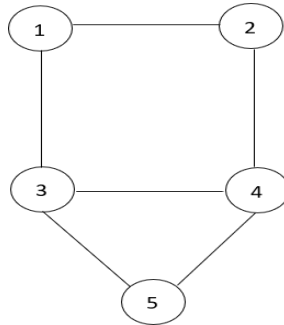
$$\text{尽量减少 } y = \sum_{j \in V} x_j + P \left( \sum_{(i, j) \in E} (1 - x_i - x_j + x_i x_j) \right)$$

其中 $P$ 再次代表一个正的标量惩罚。反过来，我们可以把它写成最小化 $\square \square \square \square$ 加上一个常数项。抛开对优化没有影响的常数，我们有一个QUBO模型形式的优化问题。

**备注：**这个问题的一个常见扩展是允许每个顶点 $j$ 都有一个权重 $w_j$ 。按照上面的发展，加权顶点覆盖问题的QUBO模型是这样的：

$$\text{尽量减少 } y = \sum_{j \in V} w_j x_j + P \left( \sum_{(i,j) \in E} (1 - x_i - x_j + x_i x_j) \right)$$

数学实例：再考虑一下第3.2节的图形，但这次我们要确定一个最小顶点覆盖。



对于这个有n=6条边和m=5个节点的图，模型变成了：

$$\begin{aligned} \text{最小化 } y = & x_1 + x_2 + x_3 + x_4 + x_5 + \\ & P(1 - x_1 - x_2 + x_1 x_2) + \\ & P(1 - x_1 - x_3 + x_1 x_3) + \\ & + P(1 - x_2 - x_4 + x_2 x_4) + \\ & + P(1 - x_3 - x_4 + x_3 x_4) + \\ & + P(1 - x_3 - x_5 + x_3 x_5) + \\ & + P(1 - x_4 - x_5 + x_4 x_5) \end{aligned}$$

可写为

$$\begin{aligned} \text{最小化 } y = & (1 - 2P)x_1 + (1 - 2P)x_2 + (1 - 3P)x_3 + (1 - 3P)x_4 + (1 - 2P)x_5 \\ & + P x_1 x_2 + P x_1 x_3 + P x_2 x_4 + P x_3 x_4 + P x_3 x_5 + P x_4 x_5 + 6P \end{aligned}$$

任意选择P等于8，并放弃加法常数（6P=48），得到我们的QUBO模型

$$QUBO : \min x Q x^t$$

Q矩阵由以下公式给出

$$Q = \begin{bmatrix} -15 & 4 & 4 & 0 & 0 \\ 4 & -15 & 0 & 4 & 0 \\ 4 & 0 & -23 & 4 & 4 \\ 0 & 4 & 4 & -23 & 4 \\ 0 & 0 & 4 & 4 & -15 \end{bmatrix}$$

请注意，我们从一个有5个变量和6个约束的约束模型变成了一个同样有5个变量的无约束的 QUBO模型。解决这个QUBO模型，可以得到： $\square\square\square\square\square\square =$

-在 $\square=(0,1,1,0,1)$ 时， $\square=48-45=3$ ，意味着节点2、3和5给出了一个最小覆盖。很容易检查出，在这个解决方案中，所有的惩罚函数都等于0。

对标量惩罚P的评论：

正如我们所指出的，许多问题的重构过程需要引入一个标量惩罚P，必须给它一个数值。这些惩罚不是唯一的，这意味着许多不同的值可以被成功地采用。对于一个特定的问题，通常要根据领域知识和需要完成的任务来设定一个可行的值。通常情况下，我们对所有的约束条件使用相同的惩罚，但如果有足够的理由对各种约束条件进行不同的处理，对不同的约束条件使用不同的惩罚也没有错。如果一个约束必须绝对满足，即一个“硬”约束，那么P必须足够大，以防止违反。然而，有些约束是“软”的，也就是说，满足它们是可取的，但可以容忍轻微的违反。对于这种情况，一个更适度的惩罚值就足够了。

过大的惩罚值会阻碍求解过程，因为惩罚项压倒了原始的目标函数信息，使人们难以区分一个解决方案和另一个解决方案的质量。另一方面，太小的惩罚值会危及对可行方案的搜索。一般来说，有一个相当大的“Goldilocks区域”，其中包含效果良好的惩罚值。对模型进行一点初步的思考，可以得到原始目标函数值的大致估计。将P作为这个估计值的某个百分比（75%到150%）通常是一个好的开始。最后，生成的解决方案总是可以被检查是否可行，从而改变惩罚措施，并根据需要进行进一步的求解过程，以锁定一个可接受的解决方案。

## 4.2 集合打包问题

集合问题是一个著名的二元变量优化问题，其一般（传统）表述为

$$\begin{aligned} & \text{最大 } \sum_{j=1}^n w_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \leq 1 \quad \text{对于 } i=1, \dots, m \end{aligned}$$

其中， $\square_{ij}$  是0/1系数， $\square_j$  是权重， $\square_j$  变量是二进制。使用前面表格中第一行和第五行所示的惩罚，我们可以很容易地构建一个与传统模型中每个约束条件相对应的二次惩罚。

然后从目标函数中减去惩罚，我们就有了一个以QUBO模型为形式的无约束的问题表示。

数学例子：考虑下面这个集合包装问题的小例子：

$$\begin{aligned} \max \quad & x_1 + x_2 + x_3 + x_4 \\ \text{s.t.} \quad & x_1 + x_3 + x_4 \leq 1 \\ & x_1 + x_2 \leq 1 \end{aligned}$$

这里所有的目标函数系数，即 $c_j$ 值，都等于1。使用上面提到的惩罚措施，等效的无约束问题是：

$$\max x_1 + x_2 + x_3 + x_4 - P_{x_1 x_3} - P_{x_1 x_4} - P_{x_3 x_4} - P_{x_1 x_2}$$

这有我们习惯的QUBO形式

$$QUBO : \max x^t Q x^t$$

其中Q矩阵，P任意选择为6，由以下公式给出

$$\begin{bmatrix} 1 & -3 & -3 & -3 \\ -3 & 1 & 0 & 0 \\ -3 & 0 & 1 & -3 \\ -3 & 0 & -3 & 1 \end{bmatrix}$$

解决QUBO模型，在 $x=(0,1,1,0)$ 时， $Q=2$ 。请注意，在这个解决方案中，所有四个惩罚项都等于零。

备注：Alidaee, et. al. (2008)使用本例中说明的QUBO重构法，有效地重构并解决了具有数千个变量和约束条件的集合包装问题。

### 4.3 最大的2-Sat问题

可满足性问题，以其不同的形式，在许多不同的场合都有应用。这些问题通常是用条款来表示的，以共轭正常形式，由几个真/假字元组成。挑战在于如何确定字词，使尽可能多的条款得到满足。

对于我们的优化方法，我们将把字词表示为0/1值，并制定可以重铸到QUBO框架中的模型，并用QUBO求解器来解决。为了说明这个方法，我们考虑被称为Max 2-Sat问题的可满足性问题的类别。

对于Max 2-Sat来说，每个子句由两个字组成，如果其中一个或两个字都为真，则子句得到满足。这个问题有三种可能的条款，每种条款都有一个传统的约束条件，如果该条款为真，则必须满足。反过来，这三个约束中的每一个都有一个已知的二次惩罚，在我们之前的表格中给出。

这三种条款类型以及它们的传统约束和相关惩罚措施是：

1. 没有否定：例子  $(x_i \vee x_j)$

传统的制约因素： $x_i + x_j \geq 1$

二次方处罚： $(1 - x_i - x_j + x_i x_j)$

2. 一个否定：例子  $(x_i \vee \neg x_j)$  传统

约束： $x_i + \neg x_j \geq 1$  二次罚分： $(1 - x_i - \neg x_j + x_i \neg x_j)$

3. 两个否定：例子  $(\neg x_i \vee \neg x_j)$  传统约

束： $\neg x_i + \neg x_j \geq 1$  二次惩罚： $(1 - \neg x_i - \neg x_j + \neg x_i \neg x_j)$

(请注意， $x_j = 1$ 或0，表示字面是真还是假。术语 $\neg x_j$ ，是 $x_j$ 的补充，等于 $(1 - x_j)$ 。)

对于每个条款类型，如果传统约束得到满足，相应的惩罚就等于零，而如果传统约束没有得到满足，二次惩罚就等于1。

鉴于这种一对一的对应关系，我们可以通过最小化不满足的条款数量来处理最大化满足条款数量的问题。正如我们将看到的，这种观点给了我们一个QUBO模型。

对于一个给定的Max 2-Sat实例，我们可以将与问题条款相关的二次惩罚相加，得到一个我们想要最小化的复合惩罚函数。由于惩罚都是二次的，这个惩罚函数采用QUBO模型的形式、

$\min \square = \square \square \square \square$ 。此外，如果在最小化QUBO模型时， $\square$ 变成等于零，这意味着我们有一个满足所有条款的解决方案；如果 $\square$ 变成等于5，这意味着我们有一个满足所有条款的解决方案，但不包括5；以此类推。

下面的例子说明了这种建模和求解过程，它有4个变量和12个子句，其中的惩罚措施由子句类型决定。

条款 #	条款	二次方罚金
1	$x_1 \vee x_2$	$(1 - x_1 - x_2 + x_1x_2)$
2	$x_1 \vee \bar{x}_2$	$(x_2 - x_1x_2)$
3	$\bar{x}_1 \vee x_2$	$(x_1 - x_1x_2)$
4	$\bar{x}_1 \vee \bar{x}_2$	$(x_1x_2)$
5	$\bar{x}_1 \vee x_3$	$(x_1 - x_1x_3)$
6	$\bar{x}_1 \vee \bar{x}_3$	$(x_1x_3)$
7	$x_2 \vee \bar{x}_3$	$(x_3 - x_2x_3)$
$x_3$		$(1 - x_2 - x_4 +$
8	$x_2 \vee$	$x_2x_4)$
$x_4$		$(x_2 - x_2x_3)$
9	$\bar{x}_2 \vee$	$(x_2x_3)$
$x_3$		$(1 - x_3 - x_4 + x_3x_4)$
10	$\bar{x}_2 \vee \bar{x}_3$	$(x_3x_4)$
$x_3$		
11	$x_3 \vee$	
$x_4$		
12	$\bar{x}_3 \vee \bar{x}_4$	
$x_4$		

将各个条款的处罚加在一起，就得到了我们的QUBO模型

或

$$\text{闽} \quad y = 3 + x_1^2x_4 - x_2x_3 + x_2x_4 + 2x_3x_4$$

、

$$\text{闽} \quad y = 3 + x Q x^t$$

其中Q矩阵由以下公式给出

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1/2 & 1/2 \\ 0 & -1/2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 1/2 & 1 & 1 \end{bmatrix}$$

求解QUBO可以得到： $y = 3 - 2 = 1$  在  $x_1 = x_2 = x_3 = 0, x_4 = 1$ , 意味着除了一个条款外，所有条款都得到满足。

备注：Kochenberger, et. al. (2005)成功地使用了上述的QUBO方法来解决具有数百个变量和数千个条款的Max 2-sat问题。这种解决Max 2-sat问题的方法的一个有趣的特点是，所得到的QUBO模型的大小与问题中的条款数量无关，只由手头的变量数量决定。因此，一个有200个变量和30,000个条款的Max 2-Sat问题可以被建模并解决为一个只有200个变量的QUBO模型。

## 第5节：创建QUBO模型：一个通用的方法

在这一节中，我们将说明如何在QUBO表述不自然出现（如我们在第3节中看到的）或事先不知道可用的惩罚措施（如我们在第4节中看到的）的情况下，构建一个适当的QUBO模型。事实证明，对于这些更普遍的情况，我们总是可以通过采用下面概述的程序来“发现”可用的罚则。

为此，考虑一般的0/1优化问题的形式：

$$\begin{aligned} \min y &= x^t C x \\ A x &= b, \quad x \text{ 二进制} \end{aligned}$$

这个模型同时容纳了二次和线性目标函数，因为当C是一个对角线矩阵时，线性情况会产生（观察一下，当 $x^2 = x_j$ 是一个0-1变量时）。在A和b有整数部分的假设下，不等式约束的问题总是可以通过包括松弛变量，然后用二进制扩展来表示松弛变量的形式。（例如，这将引入一个松弛变量s，将不等式 $4x_1 + 5x_2 - x_3 \leq 6$ 转换成 $4x_1 + 5x_2 - x_3 + s = 6$ ，因为显然 $s \leq 7$ （在 $\square_3 = 1$ 的情况下）， $\square$ 可以用二进制扩展 $\square_1 + 2\square_2 + 4\square_3$ ，其中 $\square_1$ ， $\square_2$ 和 $\square_3$ 是额外的二进制变量。如果另外知道 $\square_1$ 和 $\square_2$ 不能都是0，那么 $\square$ 最多可以是3，可以用扩展 $\square_1 + 2\square_2$ 表示。随后将对松弛变量进行更全面的处理）。这些有约束的二次优化模型通过将约束条件 $\square\square = \square$ （将松弛变量表示为x变量）转换为二次惩罚，添加到目标函数中，按照我们在第4节中说明的相同重铸方法，转换为等效的无约束QUBO模型。

具体来说，对于正标量P，我们在目标函数中加入一个二次惩罚 $P(Ax - b)^t (Ax - b)$ ，得到

$$\begin{aligned} y &= x^t C x + P (Ax - b)^t (Ax - b) \\ &= x^t C x + x^t D x + c \\ &= x^t Q x + c \end{aligned}$$

其中，矩阵D和加性常数c直接来自所示的矩阵乘法。抛开加性常数，约束问题的等效无约束版本成为

$$QUBO : \min x^t Qx, x \text{ binary}$$

#### 备注：

1. 正如我们前面所评论的那样，适当地选择惩罚标量P，总是可以使QUBO的最优解成为原始约束问题的最优解。可以随时检查所得到的解决方案的可行性，以确认是否做出了适当的惩罚选择。
2. 为了便于参考，前面将一般问题转化为等价的QUBO模型的过程将被称为**转化#1**。当我们需要将 $Ax=b$ 形式的线性约束转换为可用的二次惩罚时，就可以采用转化#1的机制，将一个给定的平等约束的问题重铸为QUBO形式。Boros和Hammer（2002）对这种方法进行了讨论，这是建立QUBO通用性的基础。对于现实的应用，需要写一个程序来实现转化#1，并产生QUBO模型所需的Q矩阵。任何方便的语言，如C++、Python、Matlab等，都可以用于这一目的。对于小问题，或大规模应用前的初步测试，我们通常可以手动进行，正如我们在这些笔记中所做的那样。
3. 请注意，加法常数c并不影响优化，在优化过程中可以忽略。一旦QUBO模型得到解决，常数c就可以用来恢复原来的目标函数值。另外，原始目标函数值总是可以通过使用QUBO求解时发现的最优 $\square_j$ 。

在事先不知道适当的二次惩罚函数的情况下，转化#1是"可采用"的方法。一般来说，它代表了一种可以用于任何问题的方法。由于这种通用性，转化#1已被证明是许多问题环境中的一个重要建模工具。

在继续讨论本节的应用之前，我们想单独提出另一个约束/惩罚对，以获得特别的认可，我们之前在第4节中曾与之合作：

$$(x_i + x_j \leq 1) \rightarrow P(x_i x_j)$$

这种形式的约束出现在许多重要的应用中。由于其重要性和使用频率，我们将这种特殊情况称为2号转换。在本节的后面，我们将有机会使用它和第1号变换。

## 5.1 设置分区

集合分割问题（SPP）是指将一个项目集合分割成子集，使每个项目正好出现在一个子集中，并且所选择的子集的成本最小。这个问题出现在许多环境中，包括航空公司和其他行业，传统上以二元变量形式表述为

$$\begin{aligned} \min \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = 1 \quad \text{对于 } i = 1, \dots, m \end{aligned}$$

其中 $x_j$ 表示是否选择子集 $j$ ， $c_j$ 是子集 $j$ 的成本， $a_{ij}$ 系数为0或1，表示变量 $x_j$ 是否明确出现在约束条件 $i$ 中。请注意，他的模型具有本节开头给出的一般模型的形式，在这种情况下，目标函数矩阵 $C$ 是一个对角线矩阵，所有非对角线元素都等于零，对角线元素由原来的线性目标函数系数给出。因此，我们可以通过使用转换#1将模型直接重铸为QUBO模型。我们用下面的例子来说明这一点。

**数值实例：考虑一个集合分割的问题**

$$\text{最小 } y = 3x_1 + 2x_2 + x_3 + x_4 + 3x_5 + 2x_6$$

受制于

$$\begin{aligned} x_1 + x_3 + x_6 &= 1 \\ x_2 + x_3 + x_5 + x_6 &= 1 \\ x_3 + x_4 + x_5 &= 1 \\ x_1 + x_2 + x_4 + x_6 &= 1 \end{aligned}$$

和 $x$ 的二进制。通常情况下，转换#1将体现在一个支持性的计算机程序中，并用于将这个问题重铸为一个QUBO模型的等效实例。对于

然而，在这个小例子中，我们可以按以下方式手动进行：通过转换#1转换为等效的QUBO模型，包括形成二次惩罚并将其添加到原始目标函数中。一般来说，要添加的二次罚分（对于一个

最小化问题）由  $P \sum_i (\sum_{j=1}^n a_{ij}x_i - b_i)^2$  其中，外侧求和为  $\square\square = \square$  的所有约束。

对于我们的例子，我们有

$$\begin{aligned} \text{最小} y = & 3x_1 + 2x_2 + x_3 + x_4 + 3x_5 + 2x_6 \\ & + P(x_1 + x_3 + x_6 - 1)^2 + P(x_2 + x_3 + x_5 + x_6 - 1)^2 \\ & + P(x_3 + x_4 + x_5 - 1)^2 + P(x_1 + x_2 + x_4 + x_6 - 1)^2 \end{aligned}$$

任意取P为10，并忆及  $\square^2 = \square_j$ ，因为我们的变量是二进制的。

成为

$$\begin{aligned} \min y = & -17x_1^2 - 18x_2^2 - 29x_3^2 - 19x_4^2 - 17x_5^2 - 28x_6^2 + 20x_1x_3 + 20x_1x_5 + 20x_1x_6 + 40x_1x_6 \\ & + 20x_2x_3 + 20x_2x_4 + 20x_2x_5 + 40x_2x_6 + 20x_3x_4 + 40x_3x_5 + 40x_3x_6 + 20x_4x_5 \\ & + 20x_4x_6 + 20x_5x_6 + 40 \end{aligned}$$

放弃加性常数40，我们就有了我们的QUBO模型

$$\min x^t Q x \quad x \text{ 二进制}$$

其中Q矩阵为

$$Q = \begin{bmatrix} -17 & 10 & 10 & 0 & 20 & 0 \\ 10 & -18 & 10 & 10 & 10 & 20 \\ 10 & 10 & -29 & 10 & 20 & 20 \\ 10 & 10 & 10 & -19 & 10 & 10 \\ 0 & 10 & 20 & 10 & -17 & 10 \\ 20 & 20 & 20 & 10 & 10 & -28 \end{bmatrix}$$

解决这个QUBO公式可以得到一个最优解  $\square_1 = \square_5 = 1$ （所有其他变量都等于0），得到  $\square = 6$ 。

**备注：**

1. 在Lewis, et. al.(2008)中，解决集合划分问题的QUBO方法已经成功地应用于解决具有数千个变量和数百个约束的大型实例。

2. 集合分割模型的特殊性质使得在构建QUBO模型时，有一个替代转化#1的方法。让  $\square_j$  表示约束矩阵A的第j列中1的数量，让  $r_{ij}$  表示变量i和j出现在同一个约束中的次数。 $q_{ii} = c_i - Pk_i$ ，Q的非对角线元素由  $q_{ij} = Pr_{ij}$ 。加法常数由  $m * P$  给出。这些关系使我们很容易为任何集合分割问题制定QUBO模型，而不需要通过转化#1的明确代数。
3. 集合划分问题可以被看作是聚类问题的一种形式，并在第6节进一步阐述。

## 5.2 图形涂色

在许多应用中，转化#1和转化#2可以协同使用，以产生一个等效的QUBO模型，正如接下来在图的着色方面所展示的那样。顶点着色问题是为了给图中的节点分配颜色，使相邻的节点得到不同的颜色。K-着色问题试图用精确的K种颜色找到这样的着色。从频率分配问题到印刷电路板设计问题等广泛的应用，都可以用K-着色模型来表示。

这些问题可以被建模为可满足性问题，如下所示：设  $\square_{ij} = 1$ ，如果节点i被分配了颜色j，则为0。

由于每个节点都必须是彩色的，我们有以下约束条件

$$\sum_{j=1}^K x_{ij} = 1 \quad i = 1, \dots, n$$

其中n是图中的节点数。一个可行的着色，即相邻的节点被分配不同的颜色，是通过施加以下约束来保证的

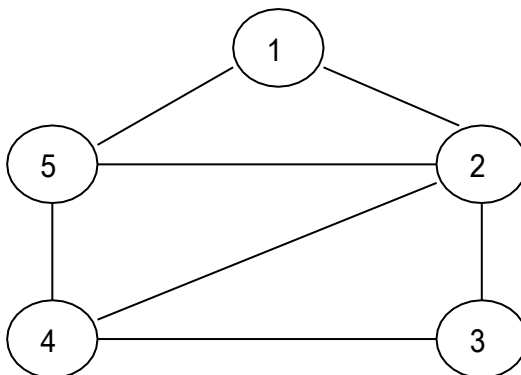
$$x_{ip} + x_{jp} \leq 1 \quad p = 1, \dots, K$$

为图中所有相邻的节点 (i, j)。

那么，这个问题可以通过在节点分配约束上使用转换1，在邻接约束上使用转换2，以QUBO模型的形式进行重构。这

问题的原始表述中没有目标函数，这意味着我们的重点是使用允许的K种颜色找到一个可行的着色。因此，任何正值的惩罚P都可以。(由此产生的QUBO模型当然有一个目标函数，即  $\sum Q$  其中Q由前述的重新表述决定)。

**数学实例：**考虑用K=3种颜色为下面的图形找一个可行的着色问题。



鉴于上面的讨论，我们看到目标是找到一个系统的解决方案：

$$\begin{aligned} x_{i1} + x_{i2} + x_{i3} &= 1 \quad i = 1, 5 \\ x_{ip} + x_{jp} &\leq 1 \quad p = 1, 3 \\ &\text{(对于所有相邻的节点 } i \text{ 和 } j) \end{aligned}$$

在这种传统形式下，该模型有15个变量和26个约束。如上所述，为了将这个问题重塑为QUBO形式，我们可以在节点分配方程上使用转换1号，在邻接不等式上使用转换2号。这里的一个方法是，从一个15乘15的Q矩阵开始，最初所有的元素都等于零，然后根据从变换#1和#2得到的惩罚重新定义适当的元素。为了澄清这个方法，我们将逐一处理这两个惩罚来源。为了便于记述，并与先前的应用相一致，我们将首先用一个下标对变量进行重新编号，从1到15，如下所示：

$$(x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, \dots, x_{52}, x_{53}) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, \dots, x_{14}, x_{15})$$

当我们开发我们的QUBO模型时，我们将使用带有单一下标的变量。首先，我们将考虑节点分配方程和我们从转换#1得到的惩罚。反过来看这些方程，我们有

$$\begin{aligned}
P(x_1 + x_2 + x_3 - 1)^2, & \text{ 成为 } P(-x_1 - x_2 - x_3 + 2x_1x_2 + 2x_1x_3 + 2x_2x_3) + P_0. \\
P(x_4 + x_5 + x_6 - 1)^2, & \text{ 变成了 } P(-x_4 - x_5 - x_6 + 2x_4x_5 + 2x_4x_6 + 2x_5x_6) + P_0. \\
P(x_7 + x_8 + x_9 - 1)^2, & \text{ 变成了 } P(-x_7 - x_8 - x_9 + 2x_7x_8 + 2x_7x_9 + 2x_8x_9) + P_0. \\
P(x_{10} + x_{11} + x_{12} - 1)^2, & \text{ 这变成了 } P(-x_{10} - x_{11} - x_{12} + 2x_{10}x_{11} + 2x_{10}x_{12} + 2x_{11}x_{12}) + P_0. \\
P(x_{13} + x_{14} + x_{15} - 1)^2, & \text{ 变成了 } P(-x_{13} - x_{14} - x_{15} + 2x_{13}x_{14} + 2x_{13}x_{15} + 2x_{14}x_{15}) + P_0.
\end{aligned}$$

将P等于4，并将这些惩罚插入"发展中的"Q矩阵中，得到以下部分完成的Q矩阵以及5P的加法常数。

$$\begin{bmatrix}
-4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & -4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & -4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & -4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 4 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -4 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & -4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 4 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -4 & 4
\end{bmatrix}$$

注意块状对角线结构。许多问题都有一些模式，可以在开发QUBO表示法所需的Q矩阵时加以利用。寻找模式通常是一个有用的去调试工具。

为了完成我们的Q矩阵，只需将代表相邻性约束的罚则插入上述矩阵即可。对于这些，我们使用转换#2的罚则，即 $Px_i x_{ij}$ ，用于每一对相邻的节点和三种允许的颜色中的每种。我们有7对相邻的节点和三种颜色，总共产生了21个相邻约束。考虑到对称性，我们将在矩阵中插入42个罚则，增加已经存在的罚则。例如，对于确保节点1和2不能同时拥有1号颜色的约束，惩罚是 $P_{11}x_1x_2$ ，这意味着我们在矩阵的第1行和第4列以及第1列和第4行插入惩罚值"2"。（记得我们已经重新标记了我们的变量，以便



原来的变量 $\square_{1,1}$ 和 $\square_{2,1}$ 现在是变量 $\square_1$ 和 $\square_4$ )。包括其他相邻约束的惩罚，完成了Q矩阵，如下所示

$$Q = \begin{bmatrix} -4 & 4 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 4 & -4 & 4 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 \\ 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & -4 & 4 \\ 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 4 & -4 \end{bmatrix}$$

上述矩阵包含了我们着色问题的所有约束，产生了等效的QUBO模型

$$QUBO : \min x Qx^t$$

解决这个模型可以得到可行的着色：

$\square_2 = \square_4 = \square_9 = \square_{11} = \square_{15} = 1$ ，其他变量都等于零。

换回我们原来的变量，这个解决方案意味着节点1和4得到2号颜色，节点2得到1号颜色，节点3和5得到3号颜色。

**备注：**正如Kochenberger, et.al. (2005)所证明的那样，这种处理图形着色问题的方法对于具有数百个节点的各种着色实例非常有效。

### 5.3 一般0/1编程

工业和政府中的许多重要问题可以被建模为具有混合约束类型的0/1线性程序。这种性质的一般问题可以表示为矩阵形式，由

$$\begin{aligned} & \max cx \\ & st \\ & Ax = b \\ & x \text{ 二进制} \end{aligned}$$

其中，根据需要引入松弛变量，将不等式约束转化为等式。给出这种形式的问题，可以用转换#1将问题重新铸成QUBO形式

$$\begin{aligned} & \max x_0 = x^t Qx \\ & st \ x \text{ binary} \end{aligned}$$

如前所述，有不等式约束的问题可以通过引入松弛变量来处理，通过二进制扩展，创建约束系统  $\square\square=\square$ 。

**数学例子：**考虑一般的0/1问题

$$\begin{aligned} & \text{最大} \quad 6x_1 + 4x_2 + 8x_3 + 5x_4 + 5x_5 \\ & \quad \quad \quad 2x_1 + 2x_2 + 4x_3 + 3x_4 + 2x_5 \leq 7 \\ & \quad \quad \quad 1x_1 + 2x_2 + 2x_3 + 1x_4 + 2x_5 = 4 \\ & \quad \quad \quad 3x_1 + 3x_2 + 2x_3 + 4x_4 + 4x_5 \geq 5 \\ & \quad \quad \quad x_i \in \{0,1\} \end{aligned}$$

由于转换1要求所有的约束都是等式而不是不等式，我们通过二进制扩展将第1和第3个约束转化为方程，包括松弛变量。要做到这一点，我们首先要估计松弛活动的上界，作为确定在二进制扩展中需要多少二进制变量来代表松弛变量的基础。通常情况下，上界是通过检查约束条件和估计松弛活动的合理值来确定的。对于目前的问题，我们可以把约束条件1和3的松弛变量称为  $\square_1$  和  $\square_3$ ，上限分别为3和6。我们的二进制扩展为：

$$\begin{aligned} 0 \leq s_1 \leq 3 & \Rightarrow s_1 = 1x_6 + 2x_7 \\ 0 \leq s_3 \leq 6 & \Rightarrow s_3 = 1x_8 + 2x_9 + 4x_{10} \end{aligned}$$

其中  $\square_6, \square_7, \square_8, \square_9$  和  $\square_{10}$  是新的二元变量。请注意，这些新变量的目标函数系数将等于零。包括这些松弛变量后，系统

$\square\square=\square$ ， $\square$ 由：

$$A = \begin{bmatrix} 2 & 2 & 4 & 3 & 2 & 1 & 2 & 0 & 0 & 0 \\ \square & 2 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 10 \\ \square\square & 3 & 3 & 2 & 4 & 4 & 0 & 0 & -1 & -2 & -4 \end{bmatrix}$$

现在我们可以使用转换#1来重新表述我们的问题，作为一个QUBO实例。在目标函数中加入惩罚措施，就可以得到

$$\begin{aligned} \text{最大 } y = & 6x_1 + 4x_2 + 8x_3 + 5x_4 + 5x_5 \\ & - P(2x_1 + 2x_2 + 4x_3 + 3x_4 + 2x_5 + 1x_6 + 2x_7 - 7)^2 \\ & - P(1x_1 + 2x_2 + 2x_3 + 1x_4 + 2x_5 - 4)^2 \\ & - P(3x_1 + 3x_2 + 2x_3 + 4x_4 + 4x_5 - 1x_8 - 2x_9 - 4x_{10} - 5)^2 \end{aligned}$$

以P=10为例，用QUBO格式重写，可以得到

$$\max y = x'Qx$$

的加法常数为-900，Q矩阵为

$$Q = \begin{bmatrix} 526 & -150 & -160 & -190 & -180 & -20 & -40 & 30 & 60 & 120 \\ -150 & 574 & -180 & -200 & -200 & -20 & -40 & 30 & 60 & 120 \\ -160 & -180 & 688 & -220 & -200 & -40 & -80 & 20 & 40 & 80 \\ -190 & -200 & -220 & 645 & -240 & -30 & -60 & 40 & 80 & 160 \\ -180 & -200 & -200 & -240 & 605 & -20 & -40 & 40 & 80 & 160 \\ -20 & -20 & -40 & -30 & -20 & 130 & -20 & 0 & 0 & 0 \\ -40 & -40 & -80 & -60 & -40 & -20 & 240 & 0 & 0 & 0 \\ 30 & 30 & 20 & 40 & 40 & 0 & 0 & -110 & -20 & -40 \\ 60 & 60 & 40 & 80 & 80 & 0 & 0 & -20 & -240 & -80 \\ 120 & 120 & 80 & 160 & 160 & 0 & 0 & -40 & -80 & -560 \end{bmatrix}$$

求解 $\max \square = \square \square \square$  得到非零值

$$x_1 = x_4 = x_5 = x_9 = x_{10} = 1$$

为，其中 $\square=916$ 。请注意，第三个约束条件是松散的。对加法常数进行调整后，它的目标函数值为16。另外，我们可以简单地在解决方案 $\square_1 = \square_4 = \square_5 = 1$ 时评估原始目标函数，从而得到目标函数值为16。

**备注：**任何线性约束和有界整数变量的问题都可以通过二进制扩展转换为  $\max Q = \sum_{i,j} Q_{ij} x_{ij}$ ，如图所示。然而，在这种应用中，Q矩阵的元素可能会根据数据的不同而变得不可接受，可能需要适当的缩放来缓解这个问题。

## 5.4 二次方作业

二次分配问题（QAP）是组合优化中的一个著名问题，在各种场合都有应用。它也是更具挑战性的解决模式之一。问题设置如下：我们得到了n个设施和n个地点，以及一个流量矩阵（ $f_{ij}$ ），表示设施i和j之间的材料流量。距离矩阵（ $d_{ij}$ ）规定了地点i和j之间的距离。优化问题是找到一个设施与地点的分配，使整个系统的加权流量最小。成本信息可以被明确地引入，以产生一个成本最小化模型，这在某些应用中是很常见的。

决策变量为  $x_{ij} = 1$ ，如果设施i被分配到位置j；否则， $x_{ij} = 0$ 。

经典的QAP模型可以说是：

$$\begin{aligned} \text{最小化} \quad & \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \sum_{j=1}^n f_{ijkl} x_{ik} x_{jl} \\ \text{受制于} \quad & \sum_{i=1}^n x_{ij} = 1 \quad j=1, n \\ & \sum_{j=1}^n x_{ij} = 1 \quad i=1, n \\ & x_{ij} \in \{0,1\} \quad i, j=1, n \end{aligned}$$

所有的 QAP 问题都有  $n^2$  变量，这在实际环境中往往会产生很大的模型。这个模型具有本节开头提出的一般形式，因此，转换#1可以用来将任何QAP问题转换成QUBO实例。

**数值实例：**考虑一个小例子，n=3个设施和3个地点，流量和距离矩阵分别如下：

$$F = \begin{bmatrix} 0 & 5 & 2 \\ 5 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix} \quad \text{和} \quad D = \begin{bmatrix} 0 & 8 & 15 \\ 8 & 0 & 13 \\ 15 & 13 & 0 \end{bmatrix}$$

如同我们之前在图形着色问题中所做的那样，只用一个下标来重新标记变量是很方便的，因此，将其替换为

$(x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33})$  由  $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$  组成。

鉴于流量和距离矩阵，我们的QAP模型变成了：

$$\min_{x_0} = 80x_1x_5 + 150x_1x_6 + 32x_1x_8 + 60x_1x_9 + 80x_2x_4 + 130x_2x_6 + 60x_2x_7 + 52x_2x_9 \\ + 150x_3x_4 + 130x_3x_5 + 60x_3x_7 + 52x_3x_8 + 48x_4x_8 + 90x_4x_9 + 78x_5x_9 + 78x_6x_8$$

受制于

$$x_1 + x_2 + x_3 = 1$$

$$x_4 + x_5 + x_6 = 1$$

$$x_7 + x_8 + x_9 = 1$$

$$x_1 + x_4 + x_7 = 1$$

$$x_2 + x_5 + x_8 = 1$$

$$x_3 + x_6 + x_9 = 1$$

$$= 1$$

将约束条件转换为二次惩罚项，并将其加入到目标函数中，可以得到无约束的二次模型

$$\min y = 80x_1x_5 + 150x_1x_6 + 32x_1x_8 + 60x_1x_9 + 80x_2x_4 + 130x_2x_6 + 60x_2x_7 + 52x_2x_9 \\ + 150x_3x_4 + 130x_3x_5 + 60x_3x_7 + 52x_3x_8 + 48x_4x_8 + 90x_4x_9 + 78x_5x_9 + 78x_6x_8 \\ + P(x_1 + x_2 + x_3 - 1)^2 + P(x_4 + x_5 + x_6 - 1)^2 + P(x_7 + x_8 + x_9 - 1)^2 \\ + P(x_1 + x_4 + x_7 - 1)^2 + P(x_2 + x_5 + x_8 - 1)^2 + P(x_3 + x_6 + x_9 - 1)^2$$

选择一个P=200的惩罚值，这就成为标准的QUBO问题

$$\text{QUBO : } y = x Q x^t$$

有一个1200的加法常数和以下9乘9的Q矩阵：

-400	200	200	200	40	75	200		1630
200	-400	200	40	200	65	16	200	26
200	200	-400	75	65	200	30	26	200
200	40	75	-400	200	200	200	24	45
40	200	65	200	-400	200	24	200	39
75	65	200	200	200	-400	45	39	200
200	16	30	200	24	45	-400	200	200
16	200	26	24	200	39	200	-400	200
26	200	45	39	200	200	200	200	-400

求解QUBO得到 $y=-982$ ，在 $x_1 = x_5 = x_9 = 1$ ，所有其他变量 = 0。调整加法常数，我们得到原始目标函数值 $1200 - 982 = 218$ 。

**备注：**如上图所示，解决QAP问题的QUBO方法在Wang, et. al. ( 2016 ) 中已成功应用于有30多个设施和地点的问题。

## 5.5 二次元结包

Knapsack问题和本节前面介绍的其他问题一样，在组合优化领域发挥着突出的作用，在项目选择和资本预算等领域有着广泛的应用。在这种情况下，一组有吸引力的潜在项目被确定，目标是确定一个满足预算限制的最大价值（或利润）的子集。当一个项目的价值只取决于所考虑的各个项目时，经典的线性knapsack问题就适用了。当成对的项目之间存在影响所获价值的互动时，这个问题的二次方版本就会出现。

对于有 $n$ 个项目的一般情况，二次Knapsack问题（QKP）通常被建模为

$$\text{最大} \sum_{j=1}^{n-1} \sum_{i=1}^n v_{ij} x_i x_j$$

受制于预算约束

$$\sum_{j=1}^n \sigma_{aj} x_j \leq b$$

其中， $x_j = 1$ ，如果选择了项目 $j$ ；否则， $x_j = 0$ 。参数 $v_{ij}$ ， $\sigma_{aj}$ 和 $b$ 分别代表与选择项目 $i$ 和 $j$ 相关的价值，资源要求的

项目j，以及总资源预算。涉及多个knapsack约束的泛化在各种应用环境中都可以找到。

**数学实例：**考虑有四个项目的QKP模型：

$$\text{最大 } 2x_1 + 5x_2 + 2x_3 + 4x_4 + 8x_1x_2 + 6x_1x_3 + 10x_1x_4 + 2x_2x_3 + 6x_2x_4 + 4x_3x_4$$

受制于knapsack约束：

$$8x_1 + 6x_2 + 5x_3 + 3x_4 \leq 16$$

我们首先将约束条件转化为方程，然后使用转化#1中包含的思想，将其重铸为QUBO模型的形式。引入一个二进制扩展形式的松弛变量  $1x_5 + 2x_6$ ，我们得到平等约束

$$8x_1 + 6x_2 + 5x_3 + 3x_4 + 1x_5 + 2x_6 = 16$$

我们可以将其转换为刑罚，以产生我们的QUBO模型，如下所示。

在目标函数中加入惩罚项，就可以得到无约束的二次模型：

$$\begin{aligned} \text{最大 } y = & 2x_1 + 5x_2 + 2x_3 + 4x_4 + 8x_1x_2 + 6x_1x_3 \\ & + 10x_1x_4 + 2x_2x_3 + 6x_2x_4 + 4x_3x_4 \\ & - P(8x_1 + 6x_2 + 5x_3 + 3x_4 + 1x_5 + 2x_6 - 16)^2 \end{aligned}$$

选择一个惩罚性的 $P=10$ ，并对代数进行清理，得到QUBO模型

$$\text{QUBO: } \max \square = \square \square \square \square$$

的加法常数为-2560，Q矩阵为

$$\begin{bmatrix} 1922 & -476 & -397 & -235 & -80 & -160 \\ -476 & 1565 & -299 & -177 & -60 & -120 \\ -397 & -299 & 1352 & -148 & -50 & -100 \\ -235 & -177 & -148 & 874 & -30 & -60 \\ -80 & -60 & -50 & -30 & 310 & -20 \\ -160 & -120 & -100 & -60 & -20 & 600 \end{bmatrix}$$

在 $\square=(1,0,1,1,0,0)$ 时，求解QUBO得到 $\square=2588$ 。对加法常数进行调整，得到原始目标函数的值28。

备注：如Glover, et. al. (2002)所示，QKP的QUBO方法在有几百个变量的问题上被证明是成功的。

## 第6节：与量子计算和机器学习的联系

*量子计算QUBO的发展*：--如第1节所述，QUBO最重要的应用之一出现在它与物理学中著名的Ising问题的等价性。与早期证明一系列显著的NP-hard问题可以转换为QUBO形式相同，Lucas（2014）最近观察到这类问题可以转换为Ising形式，包括图和数的分割、覆盖和集合打包、可满足性、匹配和约束生成树问题等等。Pakin（2017）提出了一种寻找通过迷宫的最短路径的算法，将最短路径表达为Ising哈密顿的全局最优值而不是通过传统的回溯机制。Ising问题用 $x \in \{0, 1\}^n$ 代替 $x \in \{-1, 1\}^n$ ，并且可以通过定义 $x_j' = (x_j + 1)/2$ ，然后重新定义 $x_j$ 为 $x_j'$ ，将其置于QUBO形式。

对于QUBO问题以及Ising问题的更有效的方法是使用现代元启发法得到的。在QUBO的最佳元启发式方法中，有Glover（1996，1997）、Glover和Laguna（1997）描述的基于Tabu搜索和路径重连的方法，Wang等人（2012，2013）对QUBO进行了调整。

2 一台基于量子退火的量子计算机，具有被称为Chimera图的量子比特综合物理网络结构，在其软件中采用了Wang等人（2012）的想法，并在D-Wave系统上实现。Boixo等人（2014）证明了该系统应用于QUBO问题获得量子加速效应的能力。

在D-Wave开源软件系统Qbsolv（2017）和Pakin（2018）的补充QMASM系统中提供了纳入Wang等人（2012，2013）的方法的额外进展。Qbsolv是一个hyrid经典/硬件加速器工具，它将一个可能比加速器更大/更密/更高精度的QUBO作为输入，并解决了

---

<sup>1</sup> 这给（1）增加了一个常数，这与优化无关。

<sup>2</sup> 提到量子计算，如果不提谷歌最近声称实现了'量子至上'，就不完整了。这一结果对这里讨论的计算考虑没有影响。例如，见Preskill(2019)。



在加速器上的子QUBO，并结合结果进行全QUBO解决。它使广泛的实验能够将优化问题映射为QUBO形式，以便在经典和D-wave计算机上执行。目前，D-Wave已经通过借鉴麻省理工学院的Kerberos系统（Kerberos，2019）对该系统进行了升级，为用户提供了许多便利功能。如下文所阐述的量子桥分析观点，正在提供额外的收益。

最近的QUBO量子计算应用，补充了早期在经典计算系统上的应用，包括Mniszewski等（2016）和Ushijima-Mwesigwa等（2017）的图分区问题；Negre等（2018，2019）的图聚类（量子社区检测问题）；Neukart等（2017）的交通流优化；Feld等（2018）、Clark等（2019）和Ohzeki等（2018）的车辆路由问题；Chapuis等（2018）的最大悬崖问题；Munch等（2018）的网络安全问题。（2018）、Clark等人（2019）和Ohzeki等人（2018）；Chapuis等人（2018）的最大集群问题；Munch等人（2018）和Reinhardt等人（2018）的网络安全问题；De Oliveira等人（2018）和Sahner等人（2018）的预测性健康分析问题；以及Elsokkary等人（2017）和Kalra等人（2018）的金融组合管理问题。在另一个最近的发展中，正如Alom等人（2017）和Aimone等人（2018）所报道的，正在使用IBM神经形态计算机研究QUBO模型。最近，Aramon等人（2019）调查和测试了富士通数字退火器方法，该方法也是为了解决完全连接的QUBO问题，在特定应用的CMOS硬件上实现，解决了1024个变量的问题。

多种量子计算范式正在成为重要的研究课题，而它们的相对优点也是一些争议的来源。最活跃的争论之一是关于量子门系统（也称为量子电路系统）与绝热或量子退火系统的前景。这场辩论的一部分涉及到绝热量子计算是否包含了量子纠缠这一关键因素的问题。经过一段时间，这场争论最终由Albash等人（2015）和Lanting等人（2015）解决，证明了这个问题可以得到肯定的回答。然而，另一个关键考虑涉及到退相干的作用。Amin等人（2008）和Albash和Lidar（2015）讨论了其中的一些主要问题。挑战在于门模型要有效地处理退相干。超导量子比特技术具有非常短的相干时间，绝热方法不需要，而门模型需要。

Yu等人（2018）的一项重要发现表明，绝热系统和门系统为实现量子计算过程中的固有收益提供了有效的相同潜力，通过数学证明，量子电路算法可以转化为量子绝热算法，时间复杂度完全相同。这对在绝热量子退火环境下实现的QUBO模型的相关性有有益的影响，披露了与QUBO模型相关的类似进展最终可能通过量子电路系统实现。

作为对这一分析的补充，Shaydulin等人（2018）首次对这两种领先的范式进行了性能比较，表明采用这两种框架的量子局部搜索方法可以实现与使用经典计算架构的最先进的局部搜索相媲美的结果，量子方法有可能超越经典的

随着硬件的发展，系统也在不断发展。然而，据一些分析家估计，实现这种潜力的时间框架在未来10年或更长时间内（Reedy，2017；Debenedictis，2019）。

无论哪种量子范式被证明是优越的（以及这种范式何时能与最好的经典计算系统竞争），Alom等人（2017）和Aimone等人（2018）在神经形态计算中的研究通过表明QUBO/Ising模型在多个框架中越来越重要，加强了基于绝热和门的模型的研究。

然而，为了给在量子计算机上解决QUBO问题创造条件，这些问题必须被嵌入（或编译）到量子计算硬件上，这本身就是一个非常困难的问题。Date等人（2019）针对这一问题，提出了一种嵌入QUBO问题的高效算法，该算法运行速度快，使用的量子比特比以前的方法少，并且得到的目标函数值接近全局最小值。在计算比较中，他们发现他们的嵌入算法优于D-Wave的嵌入算法，这也是目前的技术水平。

Vyskocil等人（2019）观察到，第5.3节中处理一般情况的转变形式为 $\sum_i x_i \leq k$ 引入了许多交叉产品的惩罚、

$\sum_i x_i^2$

$\sum_i x_i^3$

这给目前的量子退火器，如D-Wave系统公司的量子退火器带来了困难。作者给出了一个可扩展的、模块化的两级方法来处理这种情况，首先解决一个有16个二元变量和16个约束条件的小型初步混合整数优化问题，然后利用这个问题创建一个变换，增加QUBO变量的数量，但保持交叉乘积项的数量，从而帮助量子计算机的实施。

然而，其他考虑因素与评估解决QUBO问题的不同计算范式的性能有关，其中包括使用减少和预处理方法将大规模的QUBO问题实例分解成更小的实例。Hahn等人（2017）和Pelofske等人（2019）研究了这种预处理方法，这些方法利用上界和下界启发式方法与图分解、顶点和边缘提取以及持久性分析相结合。在Glover等人（2018）中介绍了额外的预处理方法，如随后在机器学习的背景下描述的。

**量子桥分析：连接经典和量子计算范式：**--正如美国国家科学院（2019）发布的题为《量子计算的共识研究报告：美国国家科学、工程和医学研究院（2019年）在题为《量子计算：进展与前景》的共识研究报告中强调，量子计算在未来若干年内仍将处于起步阶段，在此期间，"制定以开发近期量子计算商业应用为目标的研发计划，对该领域的健康发展至关重要。"正如本报告所指出的，这样的计划将依赖于开发"经典-量子混合技术"，这正是量子桥分析的重点。随着量子桥分析（QBA）的出现，这一领域致力于连接

随着Alpha-QUBO求解器（2019年）的开发，我们正在积极为这种混合系统建立有效的基础，以弥补经典和量子计算方法和技术之间的差距。这项工作正在为商业和学术研究环境中广泛的其他QUBO和QUBO相关应用铺平道路。最近，Glover和Kochenberger（2019）展示了QBA方法的力量，计算测试表明，Alpha-QUBO的一个亲戚，称为QUBO 2.0，解决100至500个变量的QUBO问题，比使用Kerberos的主流量子计算系统快三个数量级，此外还能够解决涉及数千个变量的更大问题。

另一种融合了经典和量子计算的方法，被称为量子近似优化算法（QAOA），是Farhi等人（2014）引入的一种混合变异算法，可以为组合优化问题产生近似解。QAOA方法最近在Zhou等人（2018）中被应用于MaxCut（MC）问题，包括正在进行的Max Independent Set（MIS）问题的变体，并被其作者声称有可能挑战领先的经典算法。理论上，QAOA方法可以应用于比QUBO模型所包含的更多类型的组合优化问题，但目前QAOA所研究的MC和MIS问题是QUBO家族中很小的一部分，并且没有提供获得处理其他QUBO问题实例能力的时间框架。重要的是，QAOA框架的参数必须被修改，以产生不同的算法来适当地处理不同的问题类型。这是否会限制这种方法在实际意义上的普遍性，还有待观察。

Wang和Abdullah（2018）承认，对QAOA表现出被称为“量子至上”特征的赞誉并不意味着QAOA将能够在重要的组合优化问题（如约束满足问题）上超越经典算法，而且目前QAOA的实现受到门控保真度的限制，在QAOA应用中，参数 $p$ 值较大的潜在优势可能会被解的准确性下降所抵消。

QAOA激发了许多研究人员对其潜在优点的赞美，尽管这种潜力的实际意义目前还没有得到很好的证实。Kochenberger等人（2019）目前正在进行调查，通过对一系列QUBO模型的计算测试来研究这个问题，这些模型属于目前可用的QAOA实现的范围，以确定QAOA与这些模型上的经典优化相比的前景。

现在，我们研究了正在积极研究的QUBO模型的领域，以及有效解决这些模型的替代计算框架的问题。

*使用QUBO的无监督机器学习：*--无监督机器学习最突出的形式之一是以聚类为代表。QUBO集合划分模型提供了一种非常自然的聚类形式，并使该模型与无监督机器学习有了有用的联系。正如Ailon等人（2008）和Aloise等人（2013）所观察到的，CPP（Clique分区问题）

在机器学习领域很受欢迎，因为它为相关聚类（CC）和模块化最大化（MM）提供了一个通用模型。Pudenz和Lidar（2013）进一步展示了基于QUBO的量子计算模型如何用于无监督的机器学习。O'Malley等人（2018）的一个相关应用研究了用D-Wave量子退火器进行非负/二进制矩阵分解。

Glover等人（2018）将QUBO应用于无监督的机器学习，提供了一种可以与量子计算一起使用或独立使用的方法。在一个补充性的发展中，聚类被用来促进Samorani等人（2018）的QUBO模型的解决，从而为研究聚类在此背景下的其他用途提供了基础。

*用QUBO进行监督机器学习：*--Schneidman等人（2006）提出了在监督机器学习中使用QUBO的建议。从物理学的角度来看，作者认为等效的Ising模型对于任何神经功能的表示都是有用的，基于这样的假设，应该利用最大熵的原则来选择神经活动的统计模型。因此，这个模型在有监督的机器学习的统计神经模型中具有自然的作用。Hamilton等人（2018）讨论了在自旋玻璃网络、玻尔兹曼机、卷积神经网络和约束满足问题中使用神经形态处理单元和量子退火器等先进计算的潜力。

*机器学习改善QUBO解决过程：*--制定规则和策略来学习特定模型实例的含义已经有很长的历史。今天，这种类型的机器学习渗透到混合整数编程领域，以确定各种关系，如可以分配给变量的值（或界限），或可以更严格地约束可行空间的不等式。虽然传统上没有通过机器学习的视角来看待，部分原因是被归入预处理的名下，但这些方法现在被广泛认可，构成了机器学习领域的一个可行的重要部分。

应用机器学习来揭示QUBO问题结构的含义的努力比那些致力于在混合整数编程领域识别这种含义的努力进行得更慢。在QUBO领域具有里程碑意义的论文是Boros等人（2008年）的工作，该论文使用屋顶二重性和最大流量算法来提供有用的模型推断。最近，Glover等人（2018）开发了几组逻辑测试，以学习QUBO应用中的变量之间的关系，在大约一半的测试问题中实现了45%的规模缩减，并在10个案例中成功修复了所有的变量，完全解决了这些问题。这些规则还确定了变量对之间的隐含关系，导致简单的逻辑不等式，以促进解决这些问题。

其他类型的机器学习方法也值得在未来的QUBO应用中仔细研究。其中包括Hoos（2012）的优化编程方法和Glover等人（1998）的综合群体分析方法。

## 第7节：结语

将问题重铸到QUBO框架中，使一个给定的二元优化问题能够被一个专门的QUBO求解器所解决，这种方法在它能够成功实施的各种显著的环境中得到了强烈的赞扬，正如本教程中所说明的。最后，我们强调了与QUBO建模及其在经典和量子计算中的应用有关的关键想法。

1. 如前所述，美国国家科学、工程和医学研究院发布了一份关于量子计算进展和前景的共识研究报告（2019年），披露了量子计算和经典计算结合的相关性，指出“制定一个研发计划，旨在开发近期量子计算的商业应用，对该领域的健康发展至关重要。这样的计划将包括……确定使用中等大小的量子子系统的经典-量子混合技术可以提供显著的速度。”洛斯阿拉莫斯国家实验室目前正在进行专门针对这一挑战的研究，以调查通过将量子计算举措与经典计算方法（如Alpha-QUBO系统（2019年）中嵌入的方法）相结合来实现这种加速的可能性。
2. Glover等人（2017）的工作中识别变量之间关系的逻辑分析可以在量子计算的环境中实现，以应对应用当前量子计算方法有效扩展解决大型问题的困难。基于这种分析的近似方法可用于分解和分割大型QUBO问题以解决大型问题，并提供与广泛的量子计算应用相关的策略。
3. 在经典和量子环境中，首先采用改变变量的方法，有时可以极大地帮助向QUBO的转化。这在原模型是基于边的图模型的情况下特别有用，如在悬崖分割中，由于图中的边的数量，标准模型可能有数百万个变量。一个有用的替代方法是引入基于节点的变量，用两个节点变量的乘积来代替每个边缘变量。这样的改变将线性模型转化为具有更少变量的二次模型，因为图形的节点数量通常比边的数量少得多。然后，由此产生的二次模型可以通过前面说明的方法转换为QUBO模型。
4. 在某些应用中会出现涉及高阶多项式的问题，可以按照Rosenberg（1975）、Rodrigues-Heck（2018）和Verma等人（2019）的思路，通过采用还原技术重新投射到QUBO框架中。例如，考虑一个具有二进制变量的三维项 $x_{123}$ 的问题。替换产品

$\square \square_{12}$  的二进制变量,  $\square_1$ , 并在目标函数中加入一个惩罚, 其形式为  $P(x \times_{12} - 2x y_{11} - 2x y_{21} + 3y_1)$ 。通过这个过程, 当优化驱动惩罚项为0时 (仅当  $y_1 = \square \square_{12}$ ), 我们已经将三次项减少为等价的二次项  $y y_{13}$ 。这个过程可以递归使用, 将高阶多项式转换为QUBO形式的二次模型。

5. 变革#1的一般程序与经典优化的拉格朗日乘数方法有相似之处。关键区别在于, 我们的标量惩罚 (P) 不是由优化决定的 "对偶" 变量。相反, 它们是预先设定的参数, 以鼓励搜索过程中避免不可行的候选解决方案。此外, 除了在凸优化的特殊情况下, 拉格朗日乘数方法并不能保证产生一个满足问题约束的解决方案, 这与QUBO模型的情况不同。为了确定拉格朗日乘数的美好值 (一般来说, 它只能产生一个下限, 而不是问题目标的最佳值), 必须求助于另外一种称为子梯度优化的优化, 而QUBO模型并不依赖于此。
6. 解决QUBO模型: 在解决QUBO模型的方法的设计和实施方面的持续进展将对优化和机器学习的广泛实际应用产生影响。下面的参考书目给出了解决这些模型的一些比较突出的方法。

## 参考文献:

N.Ailon, M. Charikar, A. Newman (2008) "聚集不一致的信息：排名和聚类", *ACM 杂志 (JACM)* , 55(5), 23。

J.B. Aimone, K.E.Hamilton, S. Mniszewsk, L. Reeder, C.D. Schuman, W.M.Severa ( 2018 ) "Spiking Neuromorphic硬件的非神经网络应用" , *PMES 研讨会*。

T.Albash, I. Hen, F. M. Spedalieri, D. A. Lidar (2015) "Reexamination of the evidence for entanglement in the D-Wave processor, " *Phys. Rev. A* 92, 62328, arXiv:1506.03539v2.

T.Albash, D. A. Lidar (2015) "绝热量子计算中的退相干 , " *Phys. Rev. A* 91, 062320, arXiv:1503.08767v2。

B.Alidaee, F. Glover, G. Kochenberger and C. Rego (2005) " A New Modeling and Solution Approach for the Number Partitioning Problem", *Journal of Applied Mathematics and Decision Sciences* 9 (2), pp.135-145.

B.Alidaee, G. Kochenberger, K. Lewis, M. Lewis, H, Wang (2008) "A New Approach for Modeling and Solving Set Packing Problems," *European Journal of Operational Research*, 186 ( 2 ) :504-512.

D.Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron, L. Liberti (2010) "Column generation algorithms for exact modularity maximization in networks", *Physical Review E*, 82 ( 4 ) , 046112.

Alpha-QUBO (2019) <http://meta-analytics.net/Home/AlphaQUBO>

M.H. S. Amin, C. J. S. Truncik, D. V. Averin (2008) "Role of Single Qubit Decoherence Time in Adiabatic Quantum Computation, " *Phys. Rev. A* 80, 022303, arXiv:0803.1196v2.

M.Anthony, E. Boros, Y. Crama, A. Gruber (2017) "Quadratic Reformulations of Nonlinear Binary Optimization Problems," *Mathematical Programming*, 162(1-2): 115-144.

M.Z. Alom, B. Van Essen, A. T. Moody, D. P. Widemann, T. M. Taha (2017) "Quadratic Unconstrained Binary Optimization (QUBO) on neuromorphic computing system," *IEEE 2017 International Joint Conference on Neural Networks ( IJCNN )* , doi 10.1109/ijcnn.2017.7966350.

M.Aramon, G. Rosenberger, E. Valiante, T. Miyazawa, H. Tamura and H. G. Katzgraber (2019) "Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer", *Frontiers in Physics*, Volumn 7, pages 48.

J.J. Berwald, J. M. Gottlieb, E. Munch (2018) "Computing Wasserstein Distance for Persistence Diagrams on a Quantum Computer", *arXiv: 1809.06433*

S.Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, M. Troyer (2014) "Evidence for quantum annealing with more than one hundred qubits, " *Nature Physics*, vol. 10, pp.

M.Booth, S. P. Reinhardt, A. Roy (2017) "Partitioning Optimization Problems for Hybrid Classical/Quantum Execution," *D-Wave Technical Report Series 14-1006A-A*, [github.com/wavesystems/qbsolv/qbsolv\\_techReport.pdf](https://github.com/wavesystems/qbsolv/qbsolv_techReport.pdf).

E.Boros, P. Hammer (1991) "The Max-cut Problem and Quadratic 0-1 Optimization : 多面体方面 , 放松和界限" , 《运筹学年鉴》 33 ( 3 ) : 151-180。

E.Boros, P. Hammer (2002) "Pseudo-Boolean Optimization," *Discrete Applied Mathematics*, 123(1): 155-225.

E.Boros, P. Hammer, G. Tavares (2007) "Local Search Heuristics for Quadratic Unconstrained Binary Optimization (QUBO)," *Journal of Heuristics*, 13(2): 99-132.

E.Boros, P. Hammer, X. Sun (1989) "用于二次0-1最小化的DDT方法 , " *RUTCOR研究中心* , RRR : 39-89。

E.Boros, P. L. Hammer,, R. Sun,, G. Tavares (2008) "A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (QUBO)," *Discrete Optimization*, Volume 5, Issue 2, pp.501-529.

G. Chapuis, H. Djidjev, G. Hahn, G. Rizk (2018) "Finding Maximum Cliques on the D-Wave Quantum Annealer," To be published in : *Journal of Signal Processing Systems*, DOI 10.1007/s11265-018-1357-8.

J.Clark, T. West, J. Zammit, X. Guo, L. Mason, D. Russell (2019) "Towards Real Time Multi-robot Routing using Quantum Computing Technologies", *HPC Asia 2019 Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, pages 111-119.

P.Date, R. Patton, C. Schuman, T. Potok (2019) "Efficiently embedding QUBO problems on adiabatic quantum computers", *Quantum Information Processing 2019*, 18:117 <https://doi.org/10.1007/s11128-019-2236-3>.

E.P. DeBenedictis (2019) "A Future with Quantum Machine Learning," *IEEE Computing Edge*, Vol.5, No.3, pp.24 - 27.

N.Elsokkary, F.S. Khan, T. S. Humble, D. L. Torre, J. Gottlieb (2017) " Financial Portfolio Management using D-Wave's Quantum Optimizer : The Case of Abu Dhabi Securities Exchange", *2017 IEEE High-performance Extreme Computing (HPEC)*.



E.Farhi和J. Goldstone (2014) "A Quantum Approximate Optimization Algorithm",  
*arXiv:1411.4028*.

S.Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Maurer, C. Linnhoff-Popien  
(2018) "A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a  
Quantum Annealer", *arXiv: 1811.07403*

F.Glover (1977) "Heuristics for Integer Programming Using Surrogate Constraints," *Decision  
Sciences*, Vol.8, No.1, pp.156-166.

F.Glover (1996) "Tabu Search and Adaptive Memory Programming - Advances, Applications  
and Challenges," in *Interfaces in Computer Science and Operations Research*, Barr,  
Helgason and Kennington (eds.) Kluwer Academic Publishers, Springer, pp.1-75.

F.Glover (1997) "A Template for Scatter Search and Path Relinking," in *Artificial Evolution,  
Lecture Notes in Computer Science*, 1363 , J.-K.Hao, E. Lutton, E. Ronald, M. Schoenauer  
and  
D.Snyers, Eds.Springer, pp.13-54.

F.Glover和G.Kochenberger, eds. (2003) *Handbook of Metaheuristics (International Series in  
Operations Research & Management Science)* Volume 1, Kluwer Academic Publishers,  
Springer, Boston.

F.Glover和G.Kochenberger ( 2019年 ) "量子桥分析与QUBO 2.0" , 2019年量子洞察会议 ,  
邀请演讲10/04/19 , LHOFT - 卢森堡金融科技之家 , 9 , rue du Laboratoire , 卢森堡。

F.Glover and M. Laguna (1997) *Tabu Search*, Kluwer Academic Publishers, Springer.  
F.Glover, B. Alidaee, C. Rego, G. Kochenberger (2002) "One-Pass Heuristics for Large Scale  
Unconstrained Binary Quadratic Problems," *European Journal of Operational Research*,  
137 ( 2 ) :272-287.

F.Glover, G. Kochenberger, B. Alidaee (1998) "Adaptive Memory Tabu Search for Binary  
Quadratic Programs," *Management Science*, 44(3): 336-345.

F.Glover, G. Kochenberger, B. Alidaee, M. Amini (1999) "Tabu Search with Critical Event  
Memory : 二元二次方程的强化应用" , 在 : *Meta-Heuristics*, Springer, Berlin, pp.93-109.

F.Glover, G. Kochenberger, B. Alidaee, M. Amini (2002) "Solving Quadratic  
Knapsack Problems by Reformulation and Tabu Search," *Combinatorial and Global  
Optimization*: (eds.

P.M. Pardalos, A. Megados, R. Burkard, 世界科学出版公司 , 第272-287页。

F.Glover, G. Kochenberger, Y Wang (2018) "A new QUBO model for unsupervised machine learning, " Research in progress.

F.Glover, J. Mulvey, D. Bai, and M. Tapia (1998) "Integrative Population Analysis for Better Solutions to Large-Scale Mathematical Programs," in *Industrial Applications of Combinatorial Optimization*, G. Yu, Ed.Kluwer Academic Publishers, Springer, Boston, MA, pp.212-237.

F.Glover, M. Lewis, G. Kochenberger (2018) "Logical and Inequality Implications for Reducing Size and Difficulty of Unconstrained Binary Optimization Problems," *European Journal of Operational Research*, 265 ( 2018 ) 829-842

F.Glover, Y. Tao, A. Punnen, G. Kochenberger (2015) "整合Tabu搜索和VLSN搜索以开发增强算法 : A Case Study Using Bipartite Boolean Quadratic Programs," *European journal of Operational Research*, 241(20;697-707.

G. Hahn和H. Djidjev ( 2017 ) "Reducing Binary Quadratic Forms for More Scalable Quantum Annealing" , *2017 IEEE International Conference on Rebooting Computing*、  
DOI: 10.1109/ICRC.2017.8123654.

K.Hamilton, C.D. Schuman, S. R. Young, N. Imam and T. S. Humble (2018) "Neural Networks and Graph Algorithms with Next-Generation Processors", *2018 IEEE International Parallel and Distributed Processing Symposium Workshops ( IPDPSW )* 。 doi: 10.1109/ipdpsw.2018.00184

P.Hammer, P. Hansen, B. Simeone (1984) "Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization," *Mathematical Programming*, 28(2): 121-155.

H.H. Hoos (2012) "Programming by Optimization," *Communications of the ACM*, Vol. 55, Issue 2, pp.70-80.

H.Huang, P. Pardalos, O. Prokopyev (2006) "下限改进和强制规则的 二元编程 , " *Comput Optim Applied*, 33(2-3):187-208.

A.Kalra, F. Qureshi, M. Tisi (2018) "Portfolio Asset Identification using Graph Algorithms on a Quantum Annealer", <http://www.henryyuen.net/fall2018/projects/qfinance.pdf>

G. Kochenberger, A. Badgett, R. Chawla, F. Glover, Y. Wang and Y. Du (2019) "QAOA和Alpha QUBO算法的比较", in process.

G. Kochenberger和F. Glover (2006) "A Unified Framework for Modeling and Solving Combinatorial Optimization Problems : A Tutorial," In : *多尺度优化方法及应用* , 编辑 : W.Hager, S-J Huang, P.P.A.。 W. Hager, S-J Huang, P. Pardalos, and O. Prokopyev, Springer,

pp.101-124.

G. Kochenberger, B. Alidaee, F. Glover, H. Wang (2007) "An Effective Modeling and Solution Approach for the Generalized Independent Set Problem," *Optimization Letters*, 1(1): 111-117.

G. Kochenberger, F. Glover, B. Alidaee, C. Rego (2005) "An Unconstrained Quadratic Binary Programming Approach to the Vertex Coloring Problem," *Annals of OR*, 139(1-4): 229-241.

G. Kochenberger, F. Glover, B. Alidaee, H. Wang (2005) "Clustering of Micro Array Data via Clique Partitioning," *Journal of Combinatorial Optimization*, 10(1): 77-92.

G. Kochenberger, F. Glover, B. Alidaee, K. Lewis (2005) "使用无约束的二次方程来建模和解决最大2-Sat问题", *国际OR杂志*, 1(1):89-100.

G. Kochenberger, J-K Hao, S. Lu, H. Wang, F. Glover (2013) "Solving Large Scale Max Cut Problems via Tabu Search," *Journal of Heuristics*, 19(4): 565-571.

G. Kochenberger, J-K.Hao, F. Glover, M. Lewis, Z. Lu, H. Wang, Y. Wang (2014) "The Unconstrained Binary Quadratic Programming Problem: A Survey," *Journal of Combinatorial Optimization*, Vol. 28, Issue 1, pp.58-81.

G. Kochenberger和M. Ma ( 2019 ) "QUBO模型在投资组合优化中的量子计算应用", 白皮书, 科罗拉多大学, 丹佛, 2019年9月。

Lanting, A.J. Przybysz, A. Yu.Smirnov, F.M. Spedalieri, M.H. Amin, A.J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J.P. Hilton, E. Hoskinson, M.W.

Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M.C. Thom, E. Tolkacheva, S. Uchaikin, A.B. Wilson, G. Rose (2014) "Entanglement in a quantum annealing processor, " *Phys. Rev. X* 4, 021041, arXiv:1401.3500v1。

M.Lewis, B. Alidaee, F. Glover, G. Kochenberger (2009) "A Note on xQx as a Modeling and Solution Framework for the Linear Ordering Problem," *International Journal of OR*, 5(2): 152- 162.

M.Lewis, B. Alidaee, G. Kochenberger (2005) "Using xQx to Model and Solve the Uncapacitated Task Allocation Problem," *Operations Research Letters*, 33(2): 176-182.

M.Lewis, G. Kochenberger, B. Alidaee (2008) "A New Modeling and Solution Approach for the Set Partitioning Problem".*Computers and OR*, 35(3):807-813.

A.Lucas (2014) "Ising Formulations of Many NP Problems," *Frontiers in Physics*, vol. 5, no. arXiv:1302.5843, p. 2.

Meta-Analytcs(2019) <http://meta-analytcs.net/>

- A.Martin, E. Boros, Y. Crama, A. Gruber (2016) "Quadratization of Symmetric Pseudo-Boolean Functions," *Discrete Applied Mathematics* 203:1-12.
- S.Mniszewski, C. Negre, H. Ushijima-Mwesigwa ( 2016 ) 。 "Graph Partitioning using the D-Wave for Electronic Structure Problems," *LA-UR-16-27873*, 1-21.
- S.M. Mniszewski, C. F. A. Negre, Ushijima-Mwesigwa (2018) "Graph Clustering Approaches using Nearterm Quantum Computing", *Argonne Quantum Computing Workshop*.
- C.F. A. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewsk (2019) "D-Wave系统上使用量子退火检测多个社区"。 *arXiv:1901.09756*
- F.Neukart, G. Compostella, C. Seidel, D. Dollen, S. Yarkoni, B. Parney (2017) "Traffic flow optimization using a quantum annealer", *arXiv: 1708.01625*
- M.Ohzeki, A. Miki, M.J. Miyama, M.Terabe (2018) "Control of automated guided vehicles without collision by quantum annealer and digital devices", *arXiv: 1812.01532*
- D.O'Malley, V.V. Vesselinov, B. S. Alexandrov, L.B. Alexandrov (2018) "Nonnegative/Binary matrix factorization with a D-Wave quantum annealer" 。 *PLoS ONE 13(12): e0206653*.  
<https://doi.org/10.1371/journal.pone.0206653>
- Kerberos ( 2019 ) *Kerberos : The Network Authentication Protocol*,  
<https://web.mit.edu/kerberos/>.
- S.Pakin ( 2017 ) "Navigating a Maze using a Quantum Annealer", *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, pages 30-36.
- S.Pakin ( 2018 ) "QMASM-量子宏汇编" , <https://ccsweb.lanl.gov/~pakin/software/>和  
<https://github.com/lanl/qmasm>
- G. Palubeckis (2006) "Iterated Tabu Search for the Unconstrained Binary Quadratic Optimization Problem," *Informatica*, 17(2) : 279-296
- P.Pardalos, G. Rodgers (1990) "Quadratic zero-one Programming的Branch and Bound Algorithm的计算方面" 。 *计算* , 45 ( 2 ) : 131-144。
- P.Pardalos, J. Xue (1999) "The Maximum Clique Problem," *Journal of Global Optimization*, 4(3): 301-328.
- P.Pardalos, O. Prokopyev, O. Shylo, V. Shylo (2008) "Global Equilibrium Search Applied to the Unconstrained Binary Quadratic Optimization Problem," *Optimization Methods and*

*Software*, 23(1): 129-140.

E.Pelofske, G. Hahn, and H. Djidjev (2019) "Solving large Maximum Clique problems on a quantum annealer", *arXiv: 1901.07657*

J.Preskill (2019) "为什么我称它为'量子至上'", *Quanta* 杂志 ,  
<https://www.quantamagazine.org/john-preskill-explains-quantum-supremacy-20191002/>

K.L. Pudenz和D.A. Lidar ( 2013 ) 。"量子绝热机器学习" , 《量子信息处理》 , 12 ( 5 ) , 2027-2070。

Qbsolv ( 2017 ) 。 D-Wave启动开放量子软件环境。 [www.dwavesys.com/press-releases/d-wave-initiates-open-quantum-software-environment](http://www.dwavesys.com/press-releases/d-wave-initiates-open-quantum-software-environment)。

C.Reedy ( 2017 ) "量子计算机何时能成为消费产品?" *Futurism*,  
<https://futurism.com/when-will-quantum-computers-be-consumer-products>

S.Reinhardt (2018) "用计算密集的图核检测横向运动",  
<http://www.clsac.org/uploads/5/0/6/3/50633811/reinhardt-clsac-2018.pdf>

E.Rodriguez-Heck (2018) "二元变量中非线性优化问题的线性和二次重构" , *列日大学博士论文*

I.Rosenberg (1975) "Reduction of Bivalent Maximization to the Quadratic Case," *Cahiers du Centre d'Etudes de Recherche Operationnelle*," 17:71-74.

D.Sahner (2018) "A Potential Role for Quantum Annealing in the Enhancement of Patient Outcomes?", <https://www.dwavesys.com/sites/default/files/Sahner.2018.pdf>

M.Samorani, Y. Wang, Y. Wang, Z. Lu, F. Glover (2018) "Clustering-Driven Evolutionary Algorithms : An Application of Path Relinking to the Quadratic Unconstrained Binary Optimization Problem," 将发表在《启发式算法杂志》的学习、强化和多样化特刊上。

E.Schneidman, M. J. Berry, R. Segev; W. Bialek (2006), "Weak pairwise correlations imply strongly correlated network states in a neural population," *Nature*, 440 (7087): pp.1007-1012。

R.Shaydulin, H. Ushijima-Mwesigwa, I. Safro, S. Mniszewski, Y. Alexeev (2018) "Community Detection Across Emerging Quantum Architectures, " *PMES Workshop*.

V.Shylo, O. Shylo (2011) "Systems Analysis Solving Unconstrained Binary Quadratic Programming Problems by Global Equilibrium Search," *Cybern Syst Anal*, 47(6): 889-897.

T.Simonite ( 2018a ) "量子计算的有线指南" , 《商业》 8.24.18 ,  
<https://www.wired.com/story/wired-guide-to-quantum-computing/>。

T.Simonite ( 2018b ) "是时候让你了解量子计算了", 《商业》, 6.25.28 ,  
[www.wired.com/story/time-you-learned-about-quantum-computing/](http://www.wired.com/story/time-you-learned-about-quantum-computing/)

美国国家科学、工程和医学研究院的共识研究报告 ( 2019年 ) , 《量子计算 : 进展与前景》 <https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects>。

H.Ushijima-Mwesigwa, C. F. A. Negre, and S. M. Mniszewsk (2017) "Graph Partitioning using Quantum Annealing on the D-Wave System", *arXiv: 1705.03082*.

A.Verma, M. Lewis (2019) "四度伪布尔函数的最优二次重述", 《优化通讯》,  
<https://doi.org/10.1007/s11590-019-01460-7>

T.Vyskocil, S. Pakin, and H. N. Djidjev (2019) " Embedding Inequality Constraints for Quantum Annealing Optimization, " *Quantum Technology and Optimization Problems.QTOP 2019.Lecture Notes in Computer Science*, vol 11413.Springer, Cham

H.Wang, B. Alidaee, F. Glover, G. Kochenberger (2006) "Solving Group Technology Problems via Clique Partitioning," *International Journal of Flexible Manufacturing Systems*, 18(2): 77-87.

H Wang, Y. Wang, M. Resende, and G. Kochenberger (2016) "A QUBO Approach to Solving QAP Problems," Unpublished manuscript.

Q.Wang和T. Abdullah ( 2018 ) "量子优化近似算法介绍",  
[https://www.cs.umd.edu/class/fall2018/cmsc657/projects/group\\_16.pdf](https://www.cs.umd.edu/class/fall2018/cmsc657/projects/group_16.pdf)

Y.Wang, Z. Lu, F. Glover和J-K.Hao (2012) "Path relinking for unconstrained binary quadratic programming," *European Journal of Operational Research* 223(3): pp.595-604.

Y.Wang, Z. Lu, F. Glover and J-K.Hao (2013) "Backbone guided tabu search for solving the UBQP problem," *Journal of Heuristics*, 19 ( 4 ) : 679-695.

Z.Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel (2017) "The Quantum Approximation Optimization Algorithm for MaxCut : A Fermionic View," *arXiv:1706.02998*.

H.Yu, Y. Huang and B. Wu (2018) "Exact Equivalence between Quantum Adiabatic Algorithm and Quantum Circuit Algorithm," *arXiv: 1706.07646v3 [quant-ph]*, DOI: 10.1088/0256-307X/35/11/110303.

L.Zhou, S. Wang, S. Choi, H.Pichler, and M. D. Lukin (2018) " Quantum Approximate Optimization Algorithm : 性能、机制和在近代设备上的实现" , *arXiv:1812.01041*



## **鸣谢：**

本教程受到了我们近年来与几位同事在许多论文上的合作的影响，我们对他们表示了极大的感谢。这些同事，按字母顺序排列，是：：Bahram Alidaee, Dick Barr, Andy Badgett, Rajesh Chawla, Yu Du, Jin-Kao Hao, Mark Lewis, Karen Lewis, Zhipeng Lu, Abraham Punnen, Cesar Rego, Yang Wang, Haibo Wang and Qinghua Wu.其他合作者的工作给了我们很大的启发，不胜枚举。他们的名字可以在我们的主页上找到，作为我们的共同作者。