frontiers | Frontiers in Quantum Science and Technology

TYPE审查
出版日期：2023年9月7日
DOI 10.3389/frqst.2023.1229471

开放存取

编辑者
胡里奥· 德维森特，
西班牙马德里卡洛斯三世大学

审查人
Nicolas Sawaya，英
特尔，美国罗明星，
中国西南交通大学

*信件
Wolfgang Lechner，
wolfgang@parityqc.com

†这些作者对这项工作做出了同等贡献

收到日期：2023年5月26日
接受日期：2023年8月15日
出版日期：2023年9月7日

引用
Dominguez F，Unger J，Traube M，Mant B，Ertler C和Lechner W（2023），量子计算的编码无关优化问题公式。*正面量子科学。Technol*。2:1229471. doi: 10.3389/frqst.2023.1229471

量子计算中与

编码无关的优化问题公式

Federico Dominguez1†，Josua Unger2†，Matthias Traube1†、Barry Mant2†、Christian Ertle1†和Wolfgang Lechner1,2,3*†

1Parity Quantum Computing Germany GmbH，德国慕尼黑，2Parity Quantum Computing GmbH，奥地利因斯布鲁克，[3]因斯布鲁克大学理论物理研究所，奥地利因斯布鲁克

我们回顾了量子计算优化问题的编码和硬件无关公式。使用这种广义方法，讨论了文献中广泛的优化问题库及其各种衍生的自旋编码。提供了用作构建这些自旋哈密顿量的构建套件的通用构建块。这种先前引入的方法为任意离散优化问题的哈密顿量的全自动构建铺平了道路，并且问题公式中的这种自由度是为不同硬件平台定制最优自旋哈密顿量的关键步骤。

关键字
量子位编码，量子优化，量子退火，QUBO， PUBO

1简介

离散优化问题在几乎任何企业中都普遍存在，其中许多问题都是NP难问题（Lenstra和Rinnooy Kan，1979）。此类问题的目标是找到一组离散变量v_k上的实值函数f（v_0，…，v_{N-1}）（*成本函数*）的最小值。搜索空间受到硬约束的限制，这些约束通常表示为等式，如g（v_0，…，v_{N-1}）=0或不等式，如h（v_0、…，v_{N-1}）>0。除了使用经典启发式方法（Dorigo和Di Caro，1999；Melnikov，2005）和机器学习方法（Mazyavkina et al.，2021）来解决这些问题外，人们对应用量子计算越来越感兴趣（Au Yeung et al.，2023）。实现这一点的常见方法包括首先将代价函数f编码在Hamiltonian H中，使得H的特征向量子集表示f的域中的元素，并且特征值是f的相应值：

$$H.\psi\square = f(v_0, \ldots, v_{N-1}).\psi\square. \tag{1}$$

在这样的编码中，H的基态是优化问题的解决方案。在获得哈密顿公式后，可以使用各种量子算法来寻找基态，包括绝热量子计算（Farhi等人，2000）和变分方法，例如量子/经典混合量子近似优化算法（QAOA）（Farhi et al.，2014）或其推广，如量子交替算子ansatz（Hadfield et al.，2019）。在硬件方面，这些算法可以在基于门的量子计算机、量子退火机或专门的伊辛机器上运行（Mohseni等人，2022）。

在当前的文献中，几乎所有用于优化的哈密顿量都被公式化为二次无约束二进制优化（QUBO）问题（Kochenberger等人，2014）。QUBO的成功反映了当前设备的强大硬件限制，其中多量子位交互不可用，必须使用ancilla量子位将其分解为两个量子位相互作用。此外，具有硬约束动 态 实现的量 子 算 法 （Hen和 Sarandy，2016；Hen和 Spedalieri，2016）需要难以在量子计算机上设计和实现的驱动项。因此，硬约束通常被包括为QUBO哈密顿量的能量惩罚。QUBO的流行也增加了一种热编码的流行，这是一种将（离散）变量映射到自旋算子本征值的特殊方式（Lucas，2014），因为这种编码允许具有低阶相互作用的哈密顿量，这特别适用于QUBO问题。

然而，QUBO和一个热的令人信服的替代方 案 旨在促进QUBO和热 编 码 之外的哈密顿公式。我们在Sawaya等人最近工作的基础上。（2022）通过使用编码独立的方法来重新审视文献中的常见问题。使用这种方法，可以使用任何自旋编码对问题进行琐碎的编码。我们还提供了最受欢迎的编码的摘要。将问题的可能约束与成本函数分开识别和呈现，从而也可以容易地探索约束的动态实现。此库中涉及的另外两个子目标是：

近年来已经提出了编码。越来越多的平台正在探索高阶相互作用（Chancellor et al.，2017；Lu et al.，2019；Schöndorf和Wilhelm，2019；Wilkinson和Hartmann，2020；Menke et al.，2021；2022；Dlaska et al.，2022；Pelegrí等 人 ，2022；Glaser et al.，2023），而 Parity架 构 （Lechner，2020；Fellner等人，2022）（LHZ架 构 的 概 括 （Lechneret al.，2015））允许将任意阶相互作用映射到只需要局部连接的量子位。还研究了约束的动态实现（Hadfield等人，2019；2017；Fuchs等人，2022；Zhu等人，2023），包括近似驱动器的设计（Wang等人，2020；Sawaya等人，2022）和奇偶性架构中约束问题的编译（Drieb-Schön等人，2022）。此外，模拟和实验结果表明，替代编码优于传统的一次性编码方法（Chancellor，2019；Sawaya等人，2020；Chen等人，2021；Plewa等人，2021年；Tamura等人，2021，Glos等人，2022；Stein等人，2023）。显然，需要探索哈密顿量的替代公式，但当哈密顿量已经使用一个热编码在QUBO中表达时，切换到其他公式并非易事。因此，探索不同配方的自动化工具将是非常有益的。

我们提供了 一个包 含 20多个 问题的库， 这些问题 是

- 元参数/选择：我们提出并回顾了在将数学公式中的优化问题映射到自旋哈密顿量的过程中做出的最重要的选择。这些主要包括编码，它可以极大地影响优化的计算成本和性能，也可以自由使用元参数或辅助变量。这些学位自由度是最优解通常仅在基态中编码的事实的结果。其他低能量本征态编码对最优解的良好近似，并且可以方便 地 进 行 近 似 ， 使 得 解 对 应 于 这 些 状 态（Montanez-Barrera等人，2022）。
- （部分）自动化：通常，每个问题都需要单独评估。由此产生的成本函数不一定是唯一的，并且不存在自动创建*H*的已知琐碎方式。通过提供成本函数的构建块和用于选择参数的启发法的集合，有助于成本函数和约束的创建。这使得能够以独立于编码的方式对问题进行一般表示，并且可以在该中间阶段进行部分参数选择和性能分析。Sawaya等人也讨论了这一目标。（2022）。

在实践中，许多优化问题不是纯粹的离散问题，而是涉及实值参数和变量。因此，在第7.4节中讨论了将实值问题编码为离散优化问题（离散化）作为中间步骤。

这篇综述的重点是优化问题，这些问题可以公式化为对角哈密顿量，写成泡利-z矩阵的和和积。这个哈密顿子集通常不适用于量子系统或量子模拟。为了介绍这些更一般的哈密顿量，我们参考了Georgescu等人的评论。（2014）关于物理问题和McArdle等人（2020）；曹等人（2019），用于量子化学模拟。

在第2节中介绍了全文中使用的符号后，我们在第3节中列出了编码列表。第4节回顾奇偶校验体系结构，第5节讨论编码约束。第6节是一本关于如何将优化问题转化为量子计算机可以解决的形式的手册。第7节包含一个优化问题库，这些优化问题分为几个类别，第8节列出了这些（以及许多其他）问题的公式中使用的构建块。第9节提供了结论。

2定义和注释

实数的离散集是一个不带累加点的可数子集$U\subset R$。离散集由大写拉丁字母表示，除了字母*G*，我们为图保留字母G。离散变量是一个范围在离散集*R*上的 变量， 用小写拉丁字母表示，主要是*v*或*w*。*R*的元素用小写希腊字母表示。

如果离散变量的范围为｛0，1｝，我们称之为*二进制*或*布尔值*。二进制变量将用字母*x*表示。类似地，范围为｛−1，1｝的变量将被称为自旋变量，字母*s*将为这些变量保留。存在从二进制变量*x*到自旋变量*s*的可逆映射：

$$x \mapsto s := 2x - 1. \tag{2}$$

对于范围为R的变量v，我们遵循Chancellor（2019）和Sawaya等人（2022），并将*值指标函数*定义为

$$\delta_\alpha^v = \begin{cases} 1 & \text{if } v = \alpha, \\ 0 & \text{if } v \neq \alpha, \end{cases} \quad (3)$$

其中$\alpha \in R$。

我们还考虑了连续变量的优化问题。一个变量v称为*连续的*，如果它的范围由$\mathbb{R}^d$给定，对于一些$d \in \mathbb{Z}_{>0}$。

优化问题O是三重（V，f，C），其中。

1. $V := \{v_i\}_{i=0,\ldots,N-1}$ is a 变量的有限集合。
2. $f := f(v_0, \ldots, v_{N-1})$ is a real-valued function, called objective or cost function.
3. $C = \{C_i\}_{i=0,\ldots,l}$ is a finite set of constraints C_i. A constraint C is 或者一个方程

$$c(v_0, \ldots, v_{N-1}) = k \quad (4)$$

对于一些$k \in R$和实值函数$c(v_0, \ldots, v_{N-1})$，或者它是一个不等式

$$c(v_0, \ldots, v_{N-1}) \leq k. \quad (5)$$

优化问题$O=(V, f, C)$的目标是找到f的极值y_{ex}，使得在y_{ex}处满足所有约束。

离散优化问题通常可以用图或超图来描述（Berge，1987）。*图*是一对（V，E），其中V是*顶点*或*节点*的有限集合，并且$E \subset \{\{v_i, v_j\} \mid v_i \neq v_j \in V\}$是这组边。元素$\{v_i, v_j\} \in E$称为之间的*边*顶点$v_i$和顶点$v_j$。注意，这样定义的图既不能有循环，即在同一个顶点开始和结束的边，也不能在同一对顶点之间有多条边。给定图$G=(V, E)$，它的*邻接矩阵*是对称的二进制$|V|\times|V|$矩阵A

$$A_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_j\} \in E, \\ 0, & \text{否则。} \end{cases} \quad (6)$$

超图是一个图的推广，其中我们允许边与两个以上的顶点相邻。也就是说，*超图*是一对$H=(V, E)$，其中V是有限的顶点集，并且

$$E \subseteq \bigcup_{i=1}^{|V|} \{\{v_1, \ldots, v_i\} \mid v_k \in V \text{ and } v_k \neq v_\ell \ \forall k, \ell = 1, \ldots, i\} \quad (7)$$

是一组超边。

构建块，Hamiltonian更紧凑，并且可以在许多不同的问题中识别重复出现的项。此外，量子算符在这个阶段并不存在：编码无关的哈密顿量 只是一个 成本函数

离散变量，这使更广泛的受众能够轻松地获得量子优化。编码变量与量子的选择算法可以稍后出现。

用Ising运算符表示构建块取决于所选择的编码。编码是将σ_z算子的特征向量与离散变量v的特定值相关联的函数：

$$|s_0, \ldots, s_{N-1}\rangle \rightarrow v, \quad s_i = \pm 1, \quad (8)$$

其中自旋变量s_i是$\sigma^{(i)}$算子的本征值。编码通常也根据二元变量x_i来定义，二元变量根据等式2与Ising变量相关。

编码的摘要如图1所示。有些编码是*密集的*，从某种意义上说，每个量子态$|s_0, \ldots, s_{N-1}\rangle$对变量$v$的一些值进行编码。其他编码是*稀疏的*，因为只有可能量子态的子集是有效态。有效子集是通过添加核心项[1]来生成的，即，需要强制执行的约束的惩罚项，以便在每个稀疏编码的变量的哈密顿量中唯一地解码变量。通常，密集编码需要更少的量子位，但稀疏编码对值指示符δ_α^v有更简单的表达式，因此有利于避免更高阶的相互作用。这是因为δ_α^v需要检查较少数量的量子位状态，以知道变量v是否具有值α，而密集编码需要寄存器中每个量子位的状态（Sawaya等人，2020；2022）。

3.1 二进制编码

二进制编码使用二进制表示进行编码

整数变量。给定一个整数变量$v \in [1, 2^D]$，我们可以使用D个二进制变量$x_i \in \{0, 1\}$来表示v：

$$v = \sum_{i=0}^{D-1} 2^i x_i + 1. \quad (9)$$

可以使用泛型编写数值指标δ_α^v表示

$$\delta_\alpha^v = \prod_{i \neq \alpha} \frac{v - i}{\alpha - i}, \quad (10)$$

我们为物理量子位保留*字量子位*。为了从编码无关的哈密顿量到量子程序，二进制或自旋变量变成作用于相应量子位的泡利-z矩阵。

3编码库

对于许多问题，成本函数和问题约束可以用两个基本的构建块来表示：整数变量v的值和等式中定义的值指标δ_α。3。当用这些术语表示时

所以我们写

这对于每个编码都是有效的。用布尔变量x_i表示的δ_α的表达式可以使用α的值编码在比特串（$x_{\alpha,0}, \ldots, x_{\alpha,D-1}$）中来编写。值指示器$\delta_\alpha$检查$D$二进制变量$x_i$是否等于$x_{\alpha,i}$，以知道变量$v$是否具有值$\alpha$。我们注意到

$$x_1(2x_2-1)-x_2+1=\begin{cases}1 & \text{if } x_1=x_2\\ 0 & \text{if } x_1\neq x_2\end{cases} \tag{11}$$

1本术语改编自Chancellor（2019）。

Encoding	Visualization example	Variable value	Value indicator	Core term
Binary	x_0 x_1 x_2 $v=2, \#v=8$	$v=\sum_{i=0}^{\log_2(N)-1} 2^i x_i + 1$	$\delta_v^\alpha = \prod_{i=0}^{D-1}\left[x_i\left(2x_{\alpha,i}-1\right) - x_{\alpha,i}+1\right]$	No core term (dense encoding)[1]
One-hot	x_0 x_1 x_2 $v=1, \#v=3$	$v=\sum_{i=0}^{N-1} i x_i$	$\delta_v^\alpha = x_\alpha$	$\left(\sum_{d=0}^{N-1} x_d - 1\right)^2$
Unary	x_0 x_1 x_2 $v=2, \#v=4$ (3 times degenerated)	$v=\sum_{i=0}^{N-1} x_i$	$\delta_v^\alpha = \prod_{i\neq\alpha}\frac{v-i}{\alpha-i}$	No core term (dense encoding)
Domain-wall	$s_0=-1$ $s_4=1$ s_1 s_2 s_3 $v=3, \#v=4$	$v=\frac{1+N}{2}-\sum_{i=1}^{N-1}\frac{s_i}{2}$	$\delta_v^\alpha = \frac{1}{2}\left(s_\alpha - s_{\alpha-1}\right)$	$\left(\sum_{\alpha=0}^{N-1} s_\alpha s_{\alpha+1} - N + 2\right)^2$
Block encoding	b_1 b_2 b_3 $v=6, \#v=9$	$v=\sum_{b=0}^{B-1}\sum_{\alpha=1}^{2^g-1} v\left(b, w_b=\alpha\right)\delta_{w_b}^\alpha$	$\delta_v^{v_0} = \delta_{w_b}^\alpha$	$\left(\sum_{b\neq b'}\sum_{d,e} x_{d,b}x_{e,b'}\right)^2$

1: Penalization term may be necessary if $\log_2 N \notin \mathbb{N}$

图1

以二进制x_i或Ising s_i变量为单位的离散变量的流行编码摘要。每个编码都有一个特定的表示

变量v的值和值指示符$\delta\alpha$。我们还包括了每个编码的可视化示例，其中红色圆圈表示激发的量子位

并且$\#v$是对于给定数量的量子位的变量v的范围。如果寄存器中的每个量子态都表示v的有效值，则编码称为*稠密*。相反，*稀疏编码*必须在哈密顿量中包括一个核心项，才能表示v的值。稀疏编码比密集编码具有更简单的值指示符表示，但核心项实现需要量子位之间的额外交互项。哈密顿量在量子位数量和相互作用项方面的成本很大程度上取决于所选择的编码，并且应该针对每个特定问题进行评估。

$$\delta_v^\alpha = \mathrm{G}_{i=0}^{D-1}[x_i(2x_{\alpha i}-1)-x_{\alpha i}+1] \tag{12}$$

通过添加核心项H来

$$= \mathrm{G}_{i=0}^{D-1} s_i(x_{\alpha,i}-\frac{1}{2})+\frac{1}{2},$$

哪里

$$s_i = 2x_i - 1 \tag{13}$$

是相应的Ising变量。因此，$\delta\alpha$中相互作用项的最大阶数与D成线性关系，其中有 $\binom{D}{v}$ 项的总数是$\sum_k\binom{D}{k}=2D$，并且二进制编码中的值指示符所需的交互项的数量与变量$N=2D$的最大值成线性比例。

如果$v\in\{1, \ldots, K\}$对于$2D<K<2D+1$, $D\in N$, 则我们需要$D+1$个二进制变量来表示v，我们将有$2D+1-K$个无效量子态，这些量子态不表示变量v的任何值。无效态的集合是

$$R = |x\rangle : \sum_{i=0} 2^i x_i + 1 > K. \tag{14}$$

核心项惩罚表示成本函数中包含的无效值的任何状态，因

对于R中的量子态，我们可以强制$v\leq K$，这可以是完成

果心 在哈密顿量中

$$Hcore = \sum_{\alpha=K+1}^{2^{D+1}} \delta_v^\alpha, \tag{15}$$

或施加总和约束

$$ccore = \sum_{\alpha=K+1}^{2^{D+1}} \delta_v^\alpha = 0 \tag{16}$$

此无需添加核心项或约束（Tamura等人，2021）（第8.2.3节）。

对于变量v, 由于核心项施加了额外的能量尺度，当$K\neq 2D$，$D\in N$时，性能可能会降低。在某些情况下，例如背包问题，无效状态的惩罚可能是

当对也可以取负值的变量进行编码时，例如，$v\in\{-K, -K+1, \ldots, K'-1, K'\}$，在经典计算中，通常使用额外的位来编码值的符号。然而，这可能不是最好的选择，因为一次旋转

翻转可能会显著改变值， 并且我们不认为完全错误

10.3389/frqst.2023.1229471

容忍对于二进制编码，有一种更适合处理负值的方法：我们可以简单地移动值

$$v = \sum_{i=0}^{D-1} 2^i x_i + 1 - K, \tag{17}$$

其中$2^{D-1} < K+K' \leq 2D$。值指示符函数的表达式保持不变，只有值α的编码必须可变−K，也可以用于其他编码。

已调整。与使用符号位相比，一个额外的优点是可以更有效地编码围绕零（$K \neq K'$）不对称的范围。转移的方法相同

3.2 灰度编码

在二进制表示中，一次自旋翻转可以导致v值的急剧变化，例如，$|1000\rangle$编码$v=9$，而$|0000\rangle$编码$v=1$。为了避免这种情况，格雷编码对二进制表示进行重新排序，使v的两个连续值在一次旋转翻转中总是不同。如果我们将势值$v \in D$排列起来，布尔变量可以描述如下：在第i个布尔变量（从右起第i列）上，序列从2^{i-1}个零开始，并以2^i 1s和2^i 0s的交替序列继续。例如，考虑[1，2^D]中的一个整数变量，该编码在

1. 0000
2. 0001
3. 0011
4. 0010
5. 0110
6. 0111

...

其中$D=4$。在每行布尔变量的左侧，我们有一个值v。例如，如果我们跟踪最右边的布尔变量，我们确实发现它以2^{1-1}=第一个值为1零开始，第二个和第三个值为2^1个1，第三个和第四个值为2^1零，依此类推。

值指示符函数和核心项保持不变，除了等式12的模拟中的α的表示也必须采用格雷编码。

这种编码相对于量子算法的一个优点是，单次自旋翻转不会导致成本函数的大变化，因此可以选择较小的系数（见第8节中的讨论）。最近，在氘的量子模拟中证明了使用格雷而不是一次热编码的优势（Di Matteo等人，2021）。

3.3 一个热编码

一种 热编码是使用N二进制的稀疏编码变量x_α 对N值变量v进行编码。编码为由其可变指标定义：

$$v = \sum_{\alpha=0}^{N-1} \alpha x_\alpha. \tag{19}$$

物理上有意义的量子态是那些在状态1中有单个量子位的量子态，因此动力学必须限制在由

$$c_{\text{core}} = \sum_{\alpha=0}^{N-1} x_\alpha - 1 = 0. \tag{20}$$

施加这种和约束的一种选择是将其编码为具有哈密顿量中的核心项的能量惩罚：

$$H_{\text{core}} = \left(1 - \sum_{\alpha=0}^{N-1} x_\alpha\right)^2, \tag{21}$$

如果只有一个x_α与零不同， 它具有最小的能量。

3.4域墙编码

这种编码使用域壁在伊辛链中的位置来编码变量v的值（Chancellor，2019；Berwald等人，2023）。如果$N+1$自旋链的端点固定在相反的状态，则该链中必须至少有一个畴壁。由于铁磁伊辛链的能量仅取决于它所具有的畴壁的数量，而不取决于它们的位置，因此具有固定相对端点的$N+1$自旋链具有N种可能的基态，这取决于单个畴壁的位置。

变量$v=l$的编码。，使用域壁编码的N需要核心Hamiltonian Chancellor（2019）：

$$H_{核心} = -\left[-s_1 + \sum_{\alpha=1}^{N-2} s_\alpha s_{\alpha+1} + s_{N-1}\right]. \tag{22}$$

由于链的固定端点不需要旋转表示（$s_0=-1$和$s_N=1$），因此$N-1$是变量$\{s_i\}^{N-1}$足以对N个值的变量进行编码。H_{core}的最小能量为$2-N$，因此可以将核心项交替编码为和约束：

$$c_{核心} = -\left[-s_1 + \sum_{\alpha=1}^{N-2} s_\alpha s_{\alpha+1} + s_{N-1}\right] = 2-N. \tag{23}$$

变量指示器证实了位置α中是否存在畴壁：

$$\delta_v^\alpha = \frac{1}{2}(s_\alpha - s_{\alpha-1}), \tag{24}$$

其中$s_0$lect−1和$s_N$lect 1，变量v可以写成

$$v = \sum_{\alpha=1}^{N} \alpha \delta_v^\alpha$$
$$= \frac{1}{2}\left((1+N) - \sum_{i=1}^{N-1} s_i\right). \tag{25}$$

$$\delta_v^{\alpha} = x_{\alpha}, \tag{18}$$

这意味着如果$x\alpha$=1，则v=α。v的值由下式给出

使用畴壁编码的量子退火实验显示，与一种热编码相比，性能显著提高（Chen等人，2021）。这部分是因为所需的搜索空间较小， 但也因为域墙

encoding generates a smoother energy landscape: in one-hot encoding, the minimum Hamming distance between two valid states is two, whereas in domain-wall, this distance is one. This implies that every valid quantum state in one-hot is a local minimum, surrounded by energy barriers generated by the core energy of Eq. 21. As a consequence, the dynamics in domain-wall encoded problems freeze later in the annealing process because only one spin-flip is required to pass from one state to the other (Berwald et al., 2023).

编码产生更平滑的能量景观：在一个热编码中，两个有效状态之间的最小汉明距离是两个，而在域壁中，这个距离是一个。这意味着一个热态中的每个有效量子态都是局部最小值，被方程21的核心能量产生的能垒包围。因此，畴壁编码问题的动力学在退火过程的后期冻结，因为从一个状态到另一个状态只需要一个自旋翻转（Berwald等人，2023）。

3.5 Unary encoding

3.5一元编码

In unary encodings, a numerical value is represented by the number of repetitions of a symbol. In the context of quantum optimization, we can use the number of qubits in excited states to represent a discrete variable (Rosenberg et al., 2015; Tamura et al., 2021)[2]. In terms of binary variables x_i, we get:

在一元编码中，数值由符号的重复次数表示。在量子优化的背景下，我们可以使用激发态中量子位的数量来表示离散变量（Rosenberg等人, 2015；Tamura等人, 2021）[2]。就二元变量x_i而言，我们得到：

$$v = \sum_{i=0}^{N-1} x_i, \tag{26}$$

3.6 Block encodings

so $N-1$ binary variables x_i are needed for encoding an N-value variable. Unary encoding does not require a core term because every quantum state is a valid state. However, this encoding is not unique in the sense that each value of v has multiple representations.

因此需要$N-1$个二进制变量x_i来编码N值变量。一元编码不需要核心项，因为每个量子态都是有效态。然而，这种编码并不是唯一的，因为v的每个值都有多个表示。

A drawback of unary encoding (and every dense encoding) is that it requires information from all binary variables to determine the value of v. The value indicator δ^α_v is

一元编码（以及每种密集编码）的一个缺点是，它需要来自所有二进制变量的信息来确定v的值。值指示符$\delta\alpha_v$为

3.6块编码

It is also possible to combine different approaches to obtain a balance between sparse and dense encodings (Sawaya et al., 2020). Block encodings are based on B blocks, each consisting of g binary variables. Similar to one-hot encoding, the valid states for block encodings are those states where only a single block contains non-zero binary variables. The binary variables in block b, $\{x_{b,i}\}^{g-1}_{i=0}$ define a block value w_b, using a dense encoding such as binary, Gray, or unary. For example, if w_b is encoded using binary, we have

还可以结合不同的方法来获得稀疏编码和密集编码之间的平衡（Sawaya等人, 2020）。块编码基于B个块，每个块由g个二进制变量组成。与一个热编码类似，块编码的有效状态是只有单个块包含非零二进制变量的状态。块b中的二进制变量，$\{x_{b,i}\}^{g-1}_{i=0}$使用密集编码（如二进制、格雷或一元）定义块值w_b。例如，如果w_b是使用二进制编码的，我们有

$$w_b = \sum_{i=0}^{g-1} 2^i x_{b,i} + 1. \tag{28}$$

The discrete variable v is defined by the active block b and its corresponding block value w_b,

离散变量v由活动块b及其对应的块值w_b定义，

$$v = \sum_{b=0}^{B-1} \sum_{\alpha=1}^{2^g-1} v(b, w_b = \alpha)\, \delta^{\alpha}_{w_b} \tag{29}$$

where $v(b, w_b = \alpha)$ is the discrete value associated with the quantum state with active block b and block value α, and $\delta^\alpha_{w_b}$ is a value

其中v（b，$w_b=\alpha$）是与具有活性块b和块值α的量子态相关的离散值，$\delta\alpha_w$是一个值

$$\delta^\alpha_v = \prod_{i\neq\alpha} \frac{v-i}{\alpha-i}, \tag{27}$$

indicator that only needs to check the value of binary variables in block b. For each block, there are 2^{g-1} possible values (assuming Gray or binary encoding for the block), because the all-zero state is not allowed (otherwise block b is not active). If the block value w_b is encoded using unary, then g values are possible. The expression of δ^α depends on the encoding and is presented in the respective

指示符，只需要检查块b中二进制变量的值。对于每个块，都有2g−1个可能的值（假设块的格雷或二进制编码），因为不允许全零状态（否则块b不活动）。如果块值wb是使用一元编码的，那么g值是可能的。 α的表达取决于编码， 并分别以

encoding section.

which involves 2^N interaction terms. This exponential scaling in the number of terms is unfavorable, so unary encoding may be only convenient for variables that do not require value indicators δ^{α}_{v} in the problem formulation, but only the variable value v. An example of this type of variable can be found in the clustering problem, as explained in Section 6.

这涉及2N个相互作用项。这种项数的指数缩放是不利的，因此一元编码可能只对问题公式中不需要值指标δα的变量方便，而只对变量值v方便。这类变量的例子可以在聚类问题中找到，如第6节所述。

A performance comparison for the Knapsack problem using digital annealers showed that unary encoding can outperform binary and one-hot encoding and requires smaller energy scales (Tamura et al., 2021). The reasons for the high performance of unary encoding are still under investigation, but redundancy is believed to play an important role because it facilitates the annealer to find the ground state. As for domain-wall encoding (Berwald et al., 2023), the minimum Hamming distance between two valid states (i.e., the number of spin flips needed to pass from one valid state to another) could also explain the better performance of the unary encoding. Redundancy has also been pointed out as a potential problem with unary encodings since not all possible values have the same degeneracy and therefore results may be biased towards the most degenerate values (Rosenberg et al., 2015).

使用数字退火器对背包问题进行的性能比较表明，一元编码可以优于二进制和单热编码，并且需要更小的能量尺度（Tamura等人，2021）。一元编码高性能的原因仍在研究中，但冗余被认为起着重要作用，因为它有助于退火器找到基态。至于域壁编码（Berwald等人， 2023），两个有效状态之间的最小汉明距离（即从一个有效状态传递到另一个有效态所需的自旋翻转次数）也可以解释一元编码的更好性能。冗余也被指出是一 元 编码的一个潜在问题， 因为并非所有可能的值都 具有相 同 的 退化性， 因此结果可能偏向于最退化的值（Rosenberg等人，2015）。

编码部分。

The value indicator for the variable v is the corresponding block value indicator. Suppose the discrete value v_0 is encoded in the block

变量v的值指示符是对应的块值指示符。假设离散值v0被编码在块中

b with a block variable

2请 注意，Ramos Calderer等人（2021）和Sawaya等人（2020）将术语 "一元"用于一种热编码。

2 Note that Ramos-Calderer et al. (2021) and Sawaya et al. (2020) use the term "unary" for one-hot encoding.

$v_0 = v(b, w_b =$

$\alpha)$. (3

0)

then the value indicator δ^{v_0} is

则值指示符δv0为

b

$$\delta^{v_0} = \delta^{\alpha}_{b}. \quad (31)$$

v

vw

A core term is necessary so that only qubits in a single block can be in the excited state. Defining $t_b = \Sigma_i x_{i,b}$, the core terms results in

核心项是必要的，这样只有单个块中的量子位才能处于激发态。定义tb=∑ixi，b，核心项的结果为

$b \neq b'$

$$H_{\text{core}} = \sum_{b \neq b'} t_b t_{b'}, \quad (32)$$

Σ

or, as a sum constraint,

或者作为总和约束，

$b \neq b'$

$$c_{\text{core}} = \sum_{b \neq b'} t_b t_{b'} = 0. \quad (33)$$

Σ

The minimum value of H_{core} is zero. If the two blocks, b and b', have binary variables with values one, then $t_b t_{b'} \neq 0$ and the corresponding eigenstate of H_{core} is no longer the ground state.

$Hcore$的最小值为零。如果两个块b和b'具有值为1的二元变量，则$tbtb'$≠0，并且$Hcore$的相应本征态不再是基态。

4 Parity architecture

4奇偶校验体系结构

The strong hardware limitations of noisy intermediate-scale quantum (NISQ) (Preskill, 2018) devices have made sparse encodings (especially one-hot) the standard approach to problem formulation. This is mainly because the basic building blocks (value and value indicator) are of linear or quadratic order in the spin variables in these encodings. The low connectivity of qubit platforms

噪声中尺度量子（NISQ）（Preskill，2018）设备的强大硬件限制使得稀疏编码（尤其是一种热门编码）成为问题公式的标准方法。这主要是因为在这些编码中，基本构建块（值和值指示符）在自旋变量中是线性或二次阶的。量子位平台的低连接性

where the interaction strength $J_{i,j,\ldots}$ is now the local field of the Parity qubit $\sigma^{(i,j,\ldots)}$. This facilitates addressing high-order interactions and frees the problem formulation from the QUBO approach. The equivalence between the original logical problem and the Parity-transformed problem is ensured by adding three- and four-body constraints and placing them on a 2D grid such that only neighboring qubits are involved in the constraints. The mapping of a logical problem into the regular grid of a Parity chip can be realized by the Parity compiler (Ender et al., 2023). Although the Parity compilation of the problem may require a larger number of physical qubits, the locality of interactions on the grid allows for higher parallelizability of quantum algorithms. This allows constant depth algorithms (Lechner, 2020; Unger et al., 2022) to be implemented with a smaller number of gates (Fellner et al., 2023).

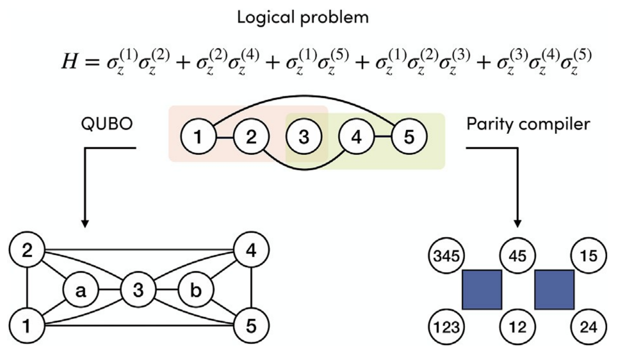

FIGURE 2
图2
Example toy problem H involving high-order interactions that shows how the Parity architecture straightforwardly handles Hamiltonians beyond the QUBO formulation. The problem is represented by the hypergraph at the top of the figure. When decomposed into QUBO form, it requires 7 qubits and 14 two-body interactions, plus additional qubit overhead depending on the embedding. In contrast, the Parity compiler can remap this problem into a 2D grid that requires six qubits with local fields and four-body interactions, represented by the blue squares. No additional embedding is necessary if the hardware is designed for the Parity Architecture.
示例玩具问题H涉及高阶相互作用，显示了奇偶校验体系结构如何直接处理QUBO公式之外的哈密顿量。该问题由图顶部的超图表示。

其中相互作用强度$J_{i,j,\ldots}$现在是奇偶校验量子位$\sigma^{(i,j,\ldots)}$的局部场。这有助于解决高阶相互作用，并将问题公式从QUBO方法中解放出来。通过添加三体和四体约束并将它们放置在2D网格上，使得只有相邻的量子位参与约束，确保了原始逻辑问题和奇偶变换问题之间的等效性。将逻辑问题映射到奇偶校验芯片的规则网格可以通过奇偶校验编译器来实现（Ender等人，2023）。尽管该问题的奇偶校验编译可能需要更多的物理量子位，但网格上相互作用的局部性允许量子算法具有更高的并行性。这允许使用较少数量的门来实现恒定深度算法（Lechner，2020；Unger等人，2022）（Fellner等人，2023）。

The following toy example, summarized in Figure 2, shows how a Parity-transformed Hamiltonian can be solved using a smaller number of qubits when the original Hamiltonian has high-order interactions. Given the logical Hamiltonian

图2中总结的以下玩具示例显示了当原始哈密顿量具有高阶相互作用时，如何使用较少数量的量子位来求解奇偶变换的哈密顿量。给定逻辑哈密顿量

$$H = \sigma_z^{(1)}\sigma_z^{(2)} + \sigma_z^{(2)}\sigma_z^{(4)} + \sigma_z^{(1)}\sigma_z^{(5)} + \sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)} + \sigma_z^{(3)}\sigma_z^{(4)}\sigma_z^{(5)}, \quad (35)$$

requires Hamiltonians in the QUBO formulation and high-order interactions are expensive when translated to QUBO Kochenberger et al. (2014). However, different choices of encodings can significantly improve the performance of quantum algorithms (Chancellor, 2019; Sawaya et al., 2020; Chen et al., 2021; Di Matteo et al., 2021; Tamura et al., 2021), by reducing the search space or generating a smoother energy landscape.

在QUBO公式中需要哈密顿量，并且当转化为QUBO Kochenberger等人时，高阶相互作用是昂贵的。（2014）。然而，不同的编码选择可以通过减少搜索空间或生成更平滑的能量景观，显著提高量子算法的性能（Chancellor，2019；Sawaya等人，2020；Chen等人，2021；Di Matteo等人，2021年；Tamura等人，2021）。

One way this difference between encodings manifests itself is in the number of spin flips of physical qubits needed to change a variable into another valid value (Berwald et al., 2023). If this number is larger than one, there are local minima separated by invalid states penalized with a high cost which can impede the performance of the optimization. On the other hand, such an energy-landscape might offer some protection against errors (Pastawski and Preskill, 2016; Fellner et al., 2022). Furthermore, other fundamental aspects of the algorithms, such as circuit depth and energy scales can be greatly improved outside QUBO (Ender et al., 2022; Drieb-Schön et al., 2023; Fellner et al., 2023; Messinger et al., 2023), prompting us to look for alternative formulations. The Parity Architecture is a paradigm for solving quantum optimization problems (Lechner et al., 2015; Ender et al., 2023) that does not rely on the QUBO formulation, allowing a wide number of options for

编码之间的这种差异表现在将一个变量更改为另一个有效值所需的物理量子位的自旋翻转次数上（Berwald等人，2023）。如果这个数字大于1，则存在由无效状态分隔的局部极小值，这些无效状态受到高成本的惩罚，这可能会阻碍优化

 10.3389/frqst.2023.1229471

的性能。另一方面，这样的能源景观可能会提供一些防止错误的保护（Pastawski和Preskill，2016；Fellner等人，2022）。此外，算法的其他基本方面，如电路深度和能量尺度，可以在QUBO之外得到极大的改进（Ender等人，2022；Drieb-Schön等人，2023；Fellner等人，2022；Messinger等人，2024），促使我们寻找替代配方。奇偶校验体系结构是解决量子优化问题的一种范式（Lechner et al.，2015；Ender et al.，2023），它不依赖于QUBO公式，允许多种选择

the corresponding QUBO formulation requires seven qubits, including two ancillas for decomposing the three-body interactions, and the total number of two-body interactions is 14. The embedding of the QUBO problem on quantum hardware may require additional qubits and interactions depending on the chosen architecture. Instead, the Parity-transformed Hamiltonian only consists of six Parity qubits with local fields and two four-body interactions between close neighbors.

formulating Hamiltonians. The architecture is based on the Parity transformation, which remaps Hamiltonians onto a 2D grid requiring only local connectivity of the qubits. The absence of

形成哈密顿量。该体系结构基于奇偶校验变换，该变换将哈密顿量重新映射到仅需要量子位局部连通性的2D网格上。的缺席

$c(v, \ldots, v) = g\, v$

long-range interactions enables high parallelizability of quantum algorithms and eliminates the need for costly and time-consuming SWAP gates, which helps to overcome two of the main obstacles of quantum computing: limited coherence time and the poor connectivity of qubits within a quantum register.

长程相互作用实现了量子算法的高并行性，并消除了对昂贵且耗时的SWAP门的需求，这有助于克服量子计算的两个主要障碍：有限的相干时间和量子寄存器内量子位的不良连接

相应的QUBO公式需要七个量子位，包括两个用于分解三体相互作用的ancilla，并且三体相互作用总数为14。在量子硬件上嵌入QUBO问题可能需要额外的量子位和相互作用，这取决于所选择的架构。相反，奇偶变换的哈密顿量仅由六个具有局部场的奇偶量子位和两个近邻之间的四体相互作用组成。

It is not yet clear what the best Hamiltonian representation is for an optimization problem. The answer will probably depend strongly on the particular use case and will take into account not only the number of qubits needed but also the smoothness of the energy landscape, which has a direct impact on the performance of quantum algorithms (King et al., 2019).

目前还不清楚优化问题的最佳哈密顿表示是什么。答案可能在很大程度上取决于特定的用例，不仅要考虑所需量子位的数量，还要考虑能量景观的平滑度，这对量子算法的性能有直接影响（King等人，2019）。

5 Encoding constraints

5编码约束

In this section, we review how to implement the hard constraints associated with the problem, assuming that the encodings of the variables have already been chosen. Hard constraints $c(v_1, \ldots, v_N) = K$ often appear in optimization problems, limiting the search space and making problems even more difficult to solve. We consider polynomial constraints of the form

在本节中，我们将回顾如何实现与问题相关的硬约束，假设变量的编码已经选择。硬约束$c(v1, \ldots, vN) = K$经常出现在优化问题中，限制了搜索空间，使问题更加难以求解。我们考虑形式的多项式约束

$$c(v_1, \ldots, v_N) = \sum_i g_i v_i + \sum g_{i,j} v_i v_j + \sum g_{i,j,k} v_i v_j v_k + \ldots, \tag{36}$$

(36)

性。

The Parity transformation creates a single Parity qubit for each interaction term in the (original) logical Hamiltonian:

奇偶性变换为（原始）逻辑哈密顿量中的每个相互作用项创建一个奇偶性量子位：

which remain polynomial after replacing the discrete variables v_i with any encoding. The coefficients $g_i, g_{i,j}, \ldots$ depend on the problem and its constraints. Even if the original problem is unconstrained, the use of sparse

 10.3389/frqst.2023.1229471

encodings such as one-hot or domain wall imposes hard constraints on the quantum variables.

其在用任何编码替换离散变量vi之后保持多项式。系数gi，gi，j。。。取决于问题及其约束条件。即使原始问题是不受约束的，使用稀疏编码（如一个热壁或域壁）也会对量子变量施加硬约束。

In general, constraints can be implemented dynamically (Hen and Sarandy, 2016; Hen and Spedalieri, 2016) (exploring only quantum states that satisfy the constraints) or as extra terms H_c

一般来说，约束可以动态实现（Hen和Sarandy，2016；Hen和Spedalieri，2016）（仅探索满足约束的量子态）或作为额外项Hc

$$\sigma^{(i)}\sigma^{(j)} \cdots \to \sigma^{(i,j,\ldots)} \quad (34)$$

$i,j,\ldots \; z \quad z$

i、j，。。。z^z

i、j，。。。z

$i,j,\ldots \; z$

in the Hamiltonian, such that eigenvectors of H are also eigenvectors

在哈密顿量中，使得H的特征向量也是特征向量

of H_c and the ground states of H_c correspond to elements in the domain of f that satisfy the constraint. These can be incorporated as a penalty term H_c into the Hamiltonian that penalizes any state outside the desired subspace:

Hc的基态与满足约束的f的域中的元素相对应。这些可以作为惩罚项Hc合并到惩罚期望子空间之外的任何状态的哈密顿量中:

$$H_c = A[c(v_1, \ldots, v_N) - K]^2, \tag{37}$$

or

或

$$H_c = A[c(v_1, \ldots, v_N) - K], \tag{38}$$

in the special case that $c(x_1, \ldots, x_N) \geq K$ is satisfied. The constant A must be large enough to ensure that the ground state of the total Hamiltonian satisfies the constraint, but the implementation of large energy scales lowers the efficiency of quantum algorithms (Lanthaler and Lechner, 2021) and additionally imposes a technical challenge. Moreover, extra terms in the Hamiltonian imply additional overhead of computational resources, especially for squared terms

在满足c（$x1$，…，xN）≥K的特殊情况下。常数A必须足够大，以确保总哈密顿量的基态满足约束，但大能量尺度的实现降低了量子算法的效率（Lanthaler和Lechner，2021），并额外带来了技术挑战。此外，哈密顿量中的额外项意味着计算资源的额外开销，尤其是平方项

such as in Eq. 37. The determination of the optimal energy scale is an important open problem. For some of the problems in the library, we provide an estimation of the energy scales (*cf.* also Section 8.4.2). Quantum algorithms for finding the ground state of Hamiltonians, such as QAOA or quantum annealing, require driver terms $U_{\mathrm{drive}} = \exp(-itH_{\mathrm{drive}})$ that spread the initial quantum state to the entire Hilbert space. Dynamical implementation of constraints employs a driver term that only explores the subspace of the Hilbert space that satisfies the constraints. Given an encoded constraint in terms of Ising

例如在等式37中。最优能量尺度的确定是一个重要的开放问题。对于库中的一些问题，我们提供了能量尺度的估计（另请参见第8.4.2节）。寻找哈密顿基态的量子算法，如QAOA或量子退火，需要将初始量子态扩展到整个希尔伯特空间的驱动项$Udrive$=exp（−itHdrive）。约束的动态实现采用了仅探索满足约束的希尔伯特空间的子空间的驱动项。给定根据Ising的编码约束

operators $\sigma_z^{(i)}$:

算子$\sigma_z^{(i)}$:

pairs of products u, w ∈ c, preserves the total magnetization and explores the complete subspace where the constraint is satisfied (Drieb-Schön et al., 2023). The decision tree for encoding constraints is presented in Figure 3.

乘积对u，w∈c，保持总磁化强度，并探索满足约束的完整子空间（Drieb-Schön等人，2023）。编码约束的决策树如图3所示。

6 Use case example

6用例示例

In this section, we present an example of the complete procedure to go from the encoding-independent formulation to the spin Hamiltonian that has to be implemented in the quantum computer, using an instance of the Clustering problem (Section 7.1.1).

在本节中，我们以聚 类 问题为例，介绍了从编码无关公式到自旋哈密顿量 的完整过程示例（第7.1.1节），该 过程必须在量子计算机中实现。

Every problem in this library includes a *Problem description*, indicating the required inputs for defining a problem instance. In the case of the clustering problem, a problem instance is defined from the number of clusters K we want to create, N objects with weights w_i, and distances $d_{i,j}$ between the objects. Two different types of discrete variables are required, variables $v_i = 1, \ldots, K$ ($i = 1, \ldots, N$) indicate to which of the K possible clusters the node i is assigned, and variables $y_j = 1, \ldots, W_{\max}$ track the weight in cluster j.

此库中的每个问题都包括一个*问题描述*，指示定义问题实例所需的输入。在聚类问题的情况下，根据我们想要创建的聚类数量K、具有权重wi的N个对象以及对象之间的距离$d_{i,j}$来定义问题实例。需要两种不同类型的离散变量，变量$v_i = 1$。，K（$i = 1, \ldots, N$）指示节点被分配给K个可能的簇中的哪一个，并且变量$y_j = 1$。，W_{max}跟踪聚类j中的权重。

The cost function of the problem only depends on $\delta^k_{v_i}$, the value indicators of variables v_i. In this case, $\delta^k_{v_i}$ indicates whether the node associated with the discrete variable v_i belongs to the cluster k or not. If two nodes i, j belong to the same cluster k, then the cost function increases by $d_{i,j}$, giving the total cost function:

该问题的代价函数仅取决于变量vi的值指标δk。在这种情况下，δk表示与离散变量v_i相关联的节点是否属于聚类k。如果两个节点i，j属于同一集群k，则成本函数增加$d_{i,j}$，得到总成本函数：

$$c(\sigma_z^{(1)}, \ldots, \sigma_z^{(N)}) = \sum_i g_i \sigma_z^{(i)} + \sum g_{i,j} \sigma_z^{(i)} \sigma_z^{(j)} + \ldots = K, \quad (39)$$

a driver Hamiltonian H_{drive} that commutes with $c(\sigma^{(1)}, \ldots, \sigma^{(N)})$

与$c(\sigma^{(1)}, \sigma^{(N)})$

will be restricted to the valid subspace provided the initial quantum state satisfies the constraint:

将被限制为有效子空间，前提是初始量子状态满足约束：

$$c|\psi\rangle = K|\psi\rangle$$

$$c|\psi_0\rangle = K|\psi_0\rangle \quad (40)$$

We can choose any encoding for the variables v_i. For example, if we decide to use binary or Gray encoding, a variable v requires $D = \log_2(K)$ qubits $\{x_u^{(i)}\}_{u=1}^{D}$ (we assume $K = 2^D$ for simplicity, but the resource estimate will not change significantly if this is not the case).

我们可以为变量v选择任何编码。例如，如果我们决定使用二进制或格雷编码，变量v需要$D=\log_2(K)$量子位（为了简单起见，我们假设$K=2D$，但是如果不是这种情况，则资源估计将不会显著改变）。

From Eq. 12 we see that $\delta^k_{v_i}$ is a polynomial of order D in variables $x_u^{(i)}$ with $\binom{D}{j}$ terms of order j and $2^D = K$ terms in total, so the product $\delta^k_{v_i}\delta^k_{v_j}$ in the cost function will have $2^D \times 2^D = 2^{2D} = K^2$ terms,

从等式12我们可以看出δk是变量中D阶的多项式x_u具有阶的$\binom{D}{j}$项和总共$2=K$项，因此成本函数中的

$$c[U_{\text{drive}}|\psi_0\rangle] = K[U_{\text{drive}}|\psi_0\rangle].$$

v_i v_j

In general, the construction of constraint-preserving drivers depends on the problem (Hen and Sarandy, 2016; Hen and Spedalieri, 2016; Hadfield et al., 2017; Chancellor, 2019; Bärtschi and Eidenbenz, 2020; Wang et al., 2020; Fuchs et al., 2021; Bakó et al., 2022). Approximate driver terms have been proposed that admit some degree of leakage and may be easier to construct (Sawaya et al., 2022). Within the Parity Architecture, each term

一般来说，约束保持驱动因素的构建取决于问题（Hen和Sarandy，2016；Hen和Spedalieri，2016；Hadfield等人，2017；Chancellor，2019；Bärtschi和Eidenbenz，2020；Wang等人，2020；Fuchs等人，2021；Bakó等人，2022）。已经提出了允许一定程度泄漏的近似驱动项，并且可能更容易构建（Sawaya等人，2022）。在奇偶校验体系结构中，每个术语

with orders between zero and $2D$. This product is summed over all v_i. Using one-hot encoding, we need K qubits per node, so NK qubits

of a polynomial constraint is a single Parity qubit (Lechner et al.,

多项式约束的是单个奇偶校验量子位（Lechner等人。，

are required for the cost function. The product $\delta^k_{v_i}\delta^k_{v_j}$

2015; Ender et al., 2023). This implies that for the Parity

2015; Ender等人，2023）。这意味着对于奇偶校验

Architecture the polynomial constraints are simply the conservation of magnetization between the qubits involved:

在体系结构中，多项式约束只是所涉及的量子位之间的磁化守恒:

terms. The K qubits associated with a variable v_i must satisfy the core constraint:

$$\sum_{i} g_{z\,i}\,\sigma_z^{(i)} + \sum_{i,j} g_{z\,z\,ij}\,\sigma_z^{(i)}\sigma_z^{(j)} + \ldots = K \rightarrow \sum_{u\in c} g_u\,\sigma_z^{(u)} = K, \tag{41}$$

where the Parity qubit $\sigma_z^{(u)}$ represents the logical qubits product

其中奇偶校验量子位 $\sigma_z^{(u)}$ 表示逻辑量子位乘积

$\sigma_z^{(u_1)}\sigma_z^{(u_2)}\ldots\sigma_z^{(u_n)}$

and we can sum over all these products that

我们可以把所有这些乘积加起来

if this constraint is implemented as an energy penalization $(c_i - 1)^2$, then $O(K^2)$ terms are added to the spin Hamiltonian for each variable, so $O(NK^2)$ terms in

v_i v_j

阶数在0和$2D$之间。这个乘积是在所有v_i上求和的。使用一个热编码，我们每个节点需要K个量子位，所以需要NK个量子位数

the $\frac{N(N-1)}{2}$ pairs i, j, so we can say that $O(N^2K^2)$ terms are required for the cost function. High-order interactions may be prohibitive for some hardware platforms, but if multiqubit gates are available, binary encoding offers an important reduction in the number of qubits.

$\frac{N(N-1)}{2}$对i，j，所以我们可以说$O(N^2K^2)$项是成本函数所必需的。对于一些硬件平台来说，高阶交互可能是令人望而却步的，但如果多量子位门可用，二进制编码可以显著减少量子位的数量。

Alternatively, we can choose a sparse encoding for the variables

或者，我们可以为变量选择稀疏编码

是成本函数所必需的。乘积$\delta^k_{v_i}\delta^k_{v_j}$是两个

is just a two-

只

qubit interaction $x^{(i)}x^{(j)}$, so the cost function requires $O(KN^2)$

量子位相互作用$x^{(i)}x^{(j)}$，因此代价函数需要$O(KN^2)$

条款。与变量v_i相关联的K个量子位必须满足核心约束:

$$c_i = \sum_{u=1}^{K} x_u^{(i)} = 1. \tag{43}$$

total are associated with the constraint. The advantage of sparse encodings is that low-order

10.3389/frqst.2023.1229471

如果该约束被实现为能量惩罚（$ci-1$）^{2}，则O（K2）项被添加到每个变量的自旋哈密顿量，因此总共O（$NK2$）项与

zz z z z

appear in the constraint c. A driver Hamiltonian based on

出现在约束c中。基于的 驱动器哈密顿量

exchange (or flip-flop) terms $\sigma^{(u)}\sigma^{(w)} + h.c.$, summed over all

交换（或触发器）项$\sigma^{(u)}\sigma^{(w)}+h.c.$，求和

interactions are needed for the value indicator functions (in this case, the maximum order is two).

值指示器函数需要

$+$ $-$

该约束相关联。稀疏编码的优点是低阶

交互（在这种情况下，最大阶数是两个）。

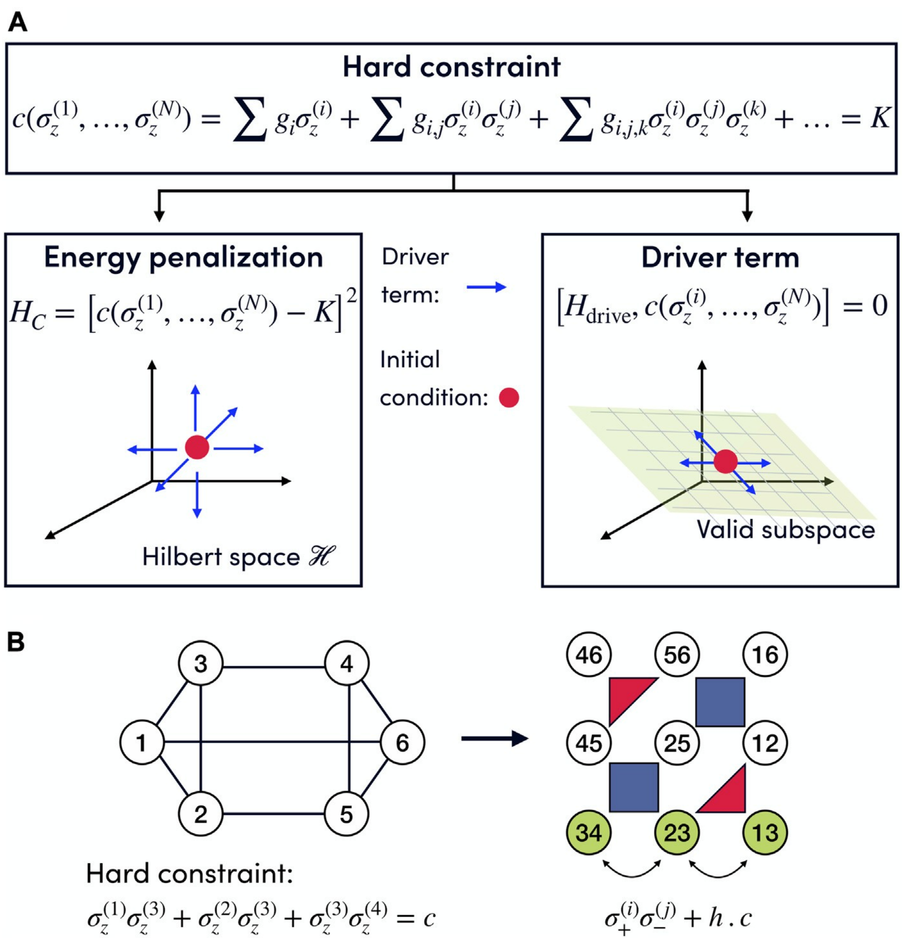

FIGURE 3
图3
(A) Decision tree to encode constraints, which can be implemented as energy penalizations in the problem Hamiltonian or dynamically by selecting
（A） 对约束进行编码的决策树，可以在问题哈密顿量中作为能量惩罚来实现，也可以通过选择
a driver Hamiltonian that preserves the desired condition. For energy penalties (left figure), the driver term needs to explore the entire Hilbert space. In contrast, the dynamic implementation of the constraints (right figure) reduces the search space to the subspace satisfying the hard constraint, thus improving the performance of the algorithms. (B) Constrained logical problem (left) and its Parity representation where each term of the polynomial constraint is represented by a single Parity qubit (green qubits), so the polynomial constraints define a subspace in which the total magnetization of the involved qubits is preserved and which can be explored with an exchange driver $\sigma_+\sigma_- + h.c.$ that preserves the total magnetization. Figure originally published in Ref. Drieb-Schön et al.

Besides the core constraints associated with the encodings, the clustering problem includes K additional constraints (one per each cluster). The total weight of nodes in any cluster cannot exceed a problem instance specific maximal value W_{max}:

除了与编码相关联的核心约束之外，聚类问题还 包括K个附加约束（每个聚类一个）。任何集群中节点的总权 重都不能超过问题实例特定的 最大值$Wmax$:

$w_i\delta_i^k \le W_{max}, \forall k \in \{1, \ldots, K\}$.(44)

$$\sum_i w_i\delta_i^k \le W_{max}, \quad \forall k \in \{1, \ldots, K\}。 \tag{44}$$

我

For the kth cluster, this constraint can be expressed in terms of auxiliary variables y_k:

对于第k个簇，该约束可以用辅助变量yk表示：

$$y_k = \sum_i w_i\delta_{v_i}^k. \tag{45}$$

我

Variables y_k are discrete variables in the range 0, . . ., W_{max}. Because the value indicators of y_k are not necessary for the constraints, we can use a dense encoding such as binary without dealing with the high-order interactions associated with value indicators of dense encodings. The variables y_k

变量yk是在0、…、…、0范围内的离散变量。，$W_{最大值}$。因为yk的值指示符对于约束是不必要的，所以我们可以使用诸如二进制的密集编码， 而不处理与密集编码的值指示符相 关 联的 高阶相 互作用。变量yk

require $K\log_2(W_{max})$ qubits if we use binary encoding, or KW_{max} if we use one-hot. These constraints can also be implemented as energy penalizations in the Hamiltonian or can be encoded in the driver term.

如果我们使用二进制编码，则 需 要 K个log2（$Wmax$）量 子

位 ， 如 果 我 们 使 用 一 个 hot， 则

需要*KWmax*。这些约束也可以被实现为哈密顿量中 的 能量惩罚， 或 者 可以被编码在驱动器项中。

The complete procedure for obtaining the spin Hamiltonian is outlined in Figure 4. We emphasize that the optimal encodings and constraint implementations depend on the details of the hardware, such as native gates, connectivity, and the number of qubits. Moreover, the efficiency of quantum algorithms is also related to the smoothness of the energy landscape, and some encodings can provide better results even though they require more qubits (see, for example, Tamura et al., 2021).

获得自旋哈密顿量的完整过程如图4所示。我们强调，最佳编码和约束实现取决于硬件的细节，如本机门、连接和量子位的数量。此外，量子算法的效率也与能量景观的平滑度有关，一些编码可以提供更好的结果，即使它们需要更多的量子位（例如，见Tamura等人，2021）。

7 Problem library

7问题库

The problems included in the library are classified into four categories: subsets, partitions, permutations, and continuous

库中包含的问题分为四类：子集、分区、排列和连续

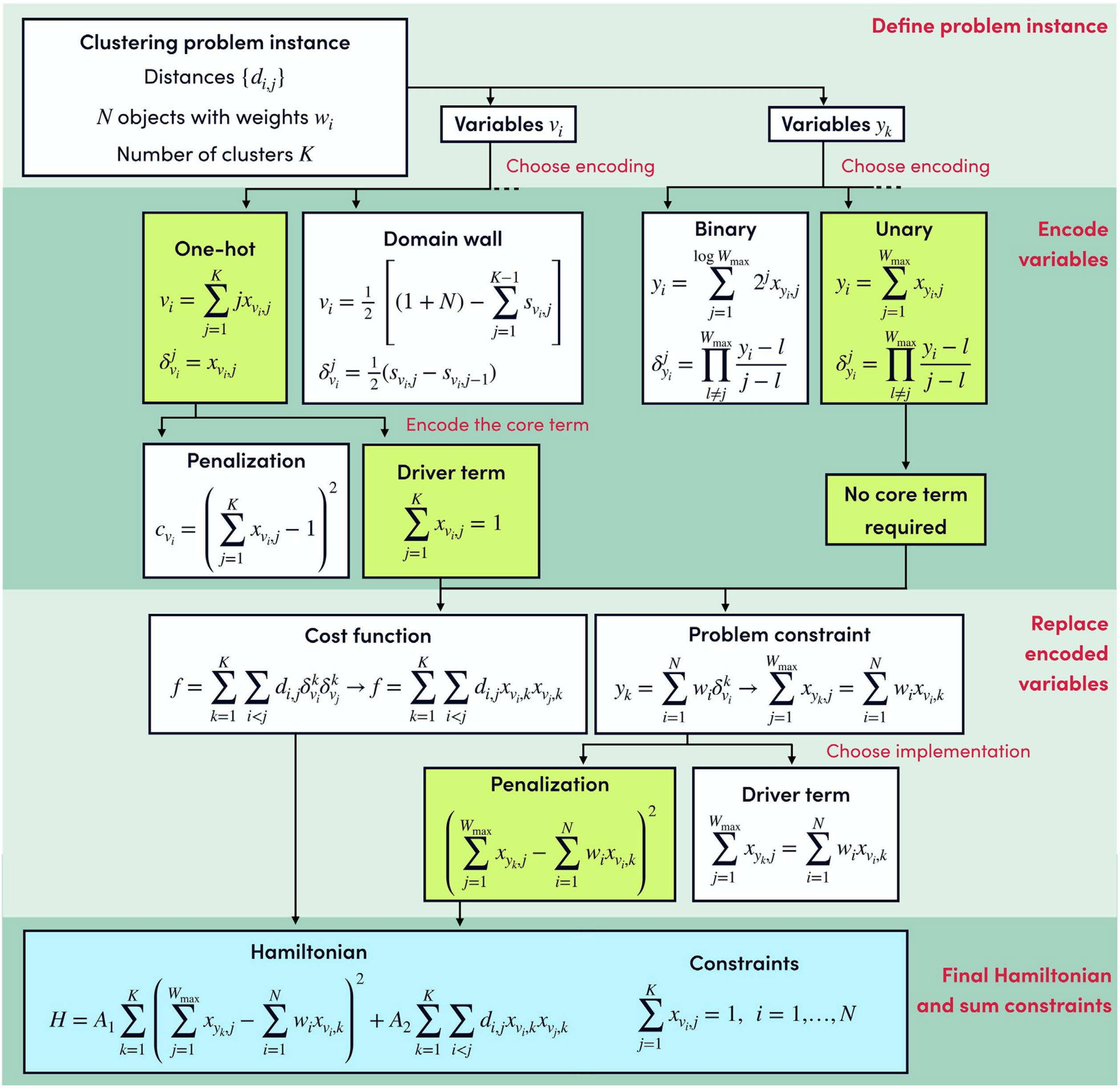

FIGURE 4
图4
Example decision tree for the clustering problem. The entire process is presented in four different blocks and the decisions taken are highlighted in green. The formulation of a problem instance requires the discrete variables v_i and y_k to be encoded in terms of qubit operators. In this example, the v_i
the encoding-independent expressions of Eqs 48, 50 the discrete variables v_i, y_k and the value indicators δ^{α}_{v} according to the chosen encoding. Core
terms associated with sparse encodings (one-hot in this case) and the problem constraints can be added to the Hamiltonian as energy penalizations or

variables. These categories are defined by the role of the discrete variable and are intended to organize the library and make it easier to find problems but also to serve as a basis for the formulation of similar use cases. An additional category in Section 7.5 contains

变量。这些类别由离散变量的作用定义，旨在组织库，使其更容易发现问题，同时也作为制定类似用例的基础。第7.5节中的附加类别包含

Partitioning problems

problems that do not fit into the previous categories but

%0.1 分区问题

The goal of partitioning problems is to look for partitions of a set U, minimizing a cost function f. A partition P of U is a set P: =

分区问题的目标是寻找集合 U的分区，使成本函数f最小化。U k=的分区P是集合P: =

may also be

不属于以前类别的问

题，但也可能是

important use cases for quantum algorithms. In Section 8 we include a summary of recurrent encoding-independent building blocks that are used throughout the library and could be useful in formulating new problems.

量子算法的重要用例。在第8节中，我们总结了整个库中使用的重复编码独立构建块，这些构建块可能有助于形成新问题。

$k \neq k'$. Partitioning problems require a discrete variable v_i for each element $u_i \in U$. The value of v_i indicates to which subset the element belongs, so v_i can take K different values. Values assigned to the subsets are arbitrary, therefore the value of v_i is usually not

$\{U_k\}_{k\in K}$ of subsets $U_k \subset U$, such that $U = \cup^K$

Uk和$UkåUk$=0如果

子集$Uk\subset U$，使得U =∪K

U_k and $U_k \cap U_{k'} = 0$ if

$k\neq k'$。分区问题要求每个元素$ui\in U$都有一个离散变量vi。vi的值表示元素属于哪个子集，因此vi可以取K个不同的值。分配给子集的 值 是任意的，因此vi的值通常不是

important in these cases, but only the value indicator δ^{α} is needed, so sparse encodings may be convenient for these variables.

在这些情况下很重要，但只需要值指示符δα，因此稀疏编码可能对这些变量很方便。

Clustering problem

7.1.1 聚类问题

Description Let $U = \{u_i\}^N$ be a set of N elements, characterized

a. 描述设U={ui}是N个元素的集合，具有特征

by weights $w_i \in \{1, \ldots, w_{max}\}$ and distances $d_{i,j}$ between them. The auxiliary variable l that can take values $1, \ldots, \Sigma u \equiv l$. Depending

可以取值1、…、…的

clustering problem looks for a partition of the set U into K non-empty subsets U_k that minimizes the distance between vertices in each subset. Partitions are subject to a weight restriction: for every subset U_k, the sum of the weights of the vertices in the subset must not exceed a given maximum weight W_{max}.

聚类问题寻找集合U到K个非空子集Uk的划分，该划分使每个子集中的顶点之间的距离最小化。分区受到权重限制：对于每个子集Uk，该子集中顶点的权重之和不得超过给定的最大权重Wmax。

Variables We can define a variable $v_i = 1, \ldots, K$ for each element in U. We also require an auxiliary variable $y_j = 0, 1, \ldots, W_{max}$ per subset U_k, that indicates the total weight of the elements in U_k:

b. 变量我们可以定义一个变量vi =1。，K。我们还需要一个辅助变量yj =0，1。，每个子集Uk的Wmax，表示Uk中元素的总重量：

on the problem instance the range of l can be restricted further. The first term in the cost function

$$y_k = \sum_{\{i:\, u_i \in U_k\}} w_i. \quad (46)$$

Constraints The weight restriction is an inequality constraint:

c. 约束权重约束是一种不等式约束：

$$y_k \leq W_{max}, \quad (47)$$

yk≤Wmax，

$$f = \sum_{i<j} (p_i - p_j)^2. \quad (52)$$

In order to minimize the maximum partial sum (maximizing the minimum partial sum is done analogously) we can introduce an

为了最小化最大部分和（类似地实现最小部分和的最大化），我们可以引入

辅助变量l。，∑u lect l 。取决于

在问题实例中，l的取值范围可以进一步限制。成本函数中的第一项

$$f = l_{max} \left[1 - \prod_i \Theta(l - p_i) \right] + l \quad (53)$$

then enforces that l is as least as large as the maximum p_i (Section 8.1.3) and the second term minimizes l. The theta step function can be expressed in terms of the value indicator functions according to

然后强制l至少与最大pi一样大（第8.1.3节），并且第二项使l最小化。θ阶跃函数可以根据

the building block Eq. 144 by either introducing auxiliary variables or expressing the value indicators directly according to the discussion in Section 8.4.1.

根据第8.4.1节中的讨论，通过引入辅助变量或直接表达值指标，构建块方程144。

which can be expressed as:

其可以表示为：

$$y_k = \sum w_i \delta^k. \quad (48)$$

Special cases For $K = 2$, the cost function that minimizes the difference between the partial sums is

d. 特殊情况对于$K=2$，使部分和之间的差最小化的代价函数是

$$f = (p_2 - p_1)^2$$

$$= \left(\sum_{i=1}^{N} \left(\delta_{v_i}^{1} - \delta_{v_i}^{2}\right) u_i\right)^2. \quad (54)$$

if the encoding for the auxiliary variables makes it necessary (e.g., a binary encoding and $W_{\max}$ not a power of 2), this constraint must be

如果辅助变量的编码是必要的（例如，二进制编码和$Wmax$不是2的幂），则该约束必须是

shifted as is described in Section 8.2.3.

如第8.2.3节所述。

d. *Cost function* The sum of the distances of the elements of a subset is:

d.成本函数子集的元素的距离之和为：

$$f_k = \sum_{\{i<j:\, u_i, u_j \in U_k\}} d_{i,j} = \sum_{i<j} d_{i,j} \delta_{v_i}^{k} \delta_{v_j}^{k}, \quad (49)$$

and the cost function results:

并且成本函数结果：

$$f = \sum_k f_k. \quad (50)$$

e. *References* Studied by Feld et al. (2019) as part of the Capacitated Vehicle Routing problem.

e.Feld等人研究的参考文献（2019），作为容量车辆路线问题的一部分。

Number partitioning

7.1.2 数字分区

Description Given a set U of N real numbers u_i, we can look for a partition of size K such that the sum of the numbers in each subset is as homogeneous as possible.

a. 描述给定一组U的N个实数ui，我们可以寻找一个大小为K的分区，使得每个子集中的数字之和尽可能是齐次的。

Variables The problem requires N variables $v_i \in [1, K]$, one per element u_i. The value of v_i indicates to which subset u_i belongs.

b. 变量这个问题需要N个变量$vi\in[1，K]$，每个元素ui。vi的值指示ui属于哪个子集。

Cost function The partial sums can be represented using the value indicators associated with the variables v_j:

The only two possible outcomes for $\delta^1 - \delta^2$ are ±1, so these factors can be trivially encoded using spin variables $s_i = \pm1$, leading to

$\delta1-\delta2$仅有的两种可能结果是±1，因此这些因素可以使用自旋变量$s_i=\pm1$进行平凡的编码，导致

$$f = \sum_{i=1}^{N} s_i u_i. \quad (55)$$

e. *References* The Hamiltonian formulation for $K = 2$ can be found in Lucas (2014).

e.参考文献K=2的哈密顿公式可以在Lucas（2014）中找到。

c. *成本函数*部分和可以使用与变量v_j相关联的值指标来表示：

$$p_j = \sum_{u_i \in v_i} u_i = \sum_{i=}^{N} u_i \delta_{j, v_i} \qquad (51)$$

Graph coloring

there are three common approaches for finding the optimal partition: maximizing the minimum partial sum, minimizing the largest partial sum, or minimizing the difference between the maximum and the minimum partitions. The latter option can be formulated as

有三种常用的方法来寻找最优分区：最大化最小部分和、最小化最大部分和或最小化最大和最小分区之间的差。后一种选择可以表述为

7.1.3 图形着色

Description The nodes of the graph $G = (V, E)$ are divided into K different subsets, each one representing a different color. We can look for a partition in which two adjacent nodes are painted with different colors and which minimizes the number of colors used.

a. *描述*图*G*=（*V*，*E*）的节点被划分为*K*个不同的子集，每个子集代表不同的颜色。我们可以寻找一个分区，其中两个相邻的节点被绘制成不同的颜色，并将使用的颜色数量最小化。

Variables We define a variable $v_i = 1...K$ for each node $i = 1, ...,$ N in the graph.

b. *变量*我们定义一个变量*vi* =*1*。*K*对于每个节点*i* =*1*。，*N*。

Constraints We must penalize solutions for which two adjacent nodes are painted with the same color. The cost function of the graph partitioning problem presented in Eq. 61 can be used for the constraint of graph coloring. In that case, $[1 - \delta(v_i - v_j)]A_{i,j}$ was used to count the number of adjacent nodes belonging to different subsets, now we can use $\delta(v_i - v_j)A_{i,j}$ to indicate if nodes i and j are painted with the same color. Therefore the constraint is (see building block Section 8.2.1)

c. *约束*我们必须惩罚两个相邻节点被涂成相同颜色的解决方案。等式61中给出的图划分问题的代价函数可用于图着色的约束。在这种情况下，[1−δ（*vi*−*vj*）]Ai，*j*用于计算属于不同子集的相邻节点的数量，现在我们可以使用δ（*vi*−*vj*）Ai，*j*来指示节点*i*和*j*是否涂有相同的颜色。因此，约束是（见构建块第8.2.1节）

$$c = \sum_{i<j} \delta(v_i - v_j)A_{i,j} = 0. \qquad (56)$$

Cost function The decision problem ("Is there a coloring that uses K colors?") can be answered by implementing the constraint as a Hamiltonian $H = c$ (note that $c \geq 0$). The existence of a solution with zero energy implies that a coloring with K colors exists.

d. *成本函数*决策问题（有没有使用*K*种颜色的着色？"）可以通过将约束实现为哈密顿量*H*=*c*（注意*c*≥*0*）来回答。零能量解的存在意味着存在*K*种颜色的着色。

Alternatively we can look for the coloring that uses the minimum number of colors. To check if a color α is used, we can use the following term (see building block Section 8.2.2):

或者，我们可以寻找使用最少颜色的着色。为了检查是否使用了颜色α，我们可以使用以下术语（见构建块第8.2.2节）：

$$u_\alpha = 1 - \prod_{i=1}^{N}\left(1 - \delta_{v_i}^{\alpha}\right) \tag{57}$$

不及物动词

must hold. If we want the two partitions, α and β, to have the same size, then the constraint is

which is one if and only if color α is not used in the coloring, and 0 if

这是一个当且仅当在着色中不使用颜色α，并且0如果

at least one node is painted with color α. The number of colors used in the coloring is the cost function of the problem:

至少有一个节点绘制有颜色α。着色中使用的颜色数量是问题的成本函数：

$$f = \sum_{\alpha=1}^{K} u_\alpha. \tag{58}$$

This objective function can be very expensive to implement since u_α includes products of δ^α of order N, so a good estimation of the minimum number $K_{\min}$ would be useful to avoid using an unnecessarily large K.

由于u_α包括N阶$\delta\alpha$的乘积，因此实现该目标函数可能非常昂贵，因此对最小数量K_{min}的良好估计将有助于避免使用不必要的大K。

References The one-hot encoded version of this problem can be found in Lucas (2014).

e. *参考文献*这个问题的一个热门编码版本可以在Lucas（2014）中找到。

Graph partitioning

7.1.4 图形分区

Description Graph partitioning divides the nodes of a graph $G = (V, E)$ into K different subsets, so that the number of edges connecting nodes in different subsets (cut edges) is minimized or maximized.

a. *描述*图划分将图G=（V，E）的节点划分为K个不同的子集，从而使连接不同子集中节点的边（切边）的数量最小化或最大化。

which is only possible if $|V|/K$ is a natural number.

Variables We can use one discrete variable $v_k = 1, \ldots, K$ per node

b. *变量*我们可以使用一个离散变量vk=1。，每个节点K

in the graph, which indicates to which partition the node belongs.

在图中，它指示节点属于哪个分区。

Cost function An edge connecting two nodes i and j is cut when v_i

c. *成本函数当vi时*，连接两个节点和j的边被切割

必须保持。如果我们希望两个分区，α和β，具有相同的大小，那么约束是

$$t_\alpha = t_\beta \tag{64}$$

and all the partitions will have the same size, imposing

并且所有分区都将具有相同的大小

$$\sum_{a<b} (t_a - t_b)^2 = 0 \tag{65}$$

这只有当$|V|/K$是自然数时才可能。

e. *References* The Hamiltonian for $K = 2$ can be found in Lucas (2014).

e.*参考文献*K=2的哈密顿量可以在Lucas（2014）中找到。

f. *Hypergraph partitioning* The problem formulation can be extended to hypergraphs. A hyperedge is cut when it contains vertices from at least two different subsets. Given a hyperedge e of G, the function

f.*超图划分*问题的公式可以推广到超图。当超边包含来自至少两个不同子集的顶点时，将对其进行切割。给定G的超边e，函数

$$\text{cut}(e) = \prod_{i,j\in e} \delta\left(v_i - v_j\right) \tag{66}$$

$$\text{切口}\ (e) = \prod_{i、j\in e} \delta_{vi-vj} \tag{66}$$

is only equal to 1 if all the vertices included in e belong to the same partition, and zero in any other case. The product over the vertices i, j of the edge e only needs to involve pairs such that every vertex appears at least once. The optimization objective of minimizing the cut hyperedges is implemented by the sum of penalties

仅等于1，如果e中包括的所有顶点都属于同一分区，而在任何其他情况下为零。边e的顶点i，j上的乘积只需要涉及对，使得每个顶点至少出现一次。通过惩罚的总和来实现最小化切割超边的优化目标

$$f = \sum_{e\in E}\left(1 - \text{cut}(e)\right) = \sum_{e\in E}\left(1 - \prod_{i,j\in e}\sum_{a=1}^{K}\delta_i^a\delta_j^a\right) \tag{67}$$

$\neq v_j$. It is convenient to use the symbol (see building block Section 8.1.4):

≠vj。使用该符号很方便（见构建块第8.1.4节）：

This objective function penalizes all possible cuts in the same way, regardless of the number of vertices cut or the number of partitions

$$\delta(v_i - v_j) = \sum_{\alpha=}^{K} \delta^{\alpha}_{v_i}\delta^{\alpha}_{v_j} = \begin{cases} 1 & \text{if } v_i = v_j \\ 0 & \text{if } v_i \neq v_j \end{cases}, \quad (59)$$

to which an edge belongs.

边缘所属

which is equal to 1 when nodes i and j belong to the same partition ($v_i = v_j$) and zero when not ($v_i \neq v_j$). In this way, the term:

当节点i和j属于同一分区（vi =vj）时等于1，当不属于（vi≠vj）则等于0。通过这种方式，术语：

$$(1 - \delta(v_i - v_j))A_{i,j} \quad (60)$$

（1−δ（vi−vj））Ai，j (60)

is equal to 1 if there is a cut edge connecting nodes i and j, being $A_{i,j}$ the adjacency matrix of the graph. The cost function for minimizing the number of cut edges is obtained by summing over all the nodes:

等于1，如果存在连接节点i和j的割边，Ai，j是图的邻接矩阵。通过对所有节点求和来获得用于最小化切割边的数量的成本函数：

$$f = \sum_{i<j}[1 - \delta(v_i - v_j)]A_{i,j}, \quad (61)$$

or alternatively $-f$ to maximize the number of cut edges.

或者−f以最大化切割边缘的数量。

Constraints A common constraint imposes that partitions have a

d. *约束一个常见的约束强制分区具有*

Description Given a graph $G = (V, E)$, we seek the minimum number of colors K for coloring all vertices such that the subsets W_α of vertices with the color α together with the edge set E_α restricted to edges between vertices in W_α form complete graphs. A subproblem is to decide if there is a clique cover using K colors.

specific size. The element counting building block Section 8.1.1 is defined as

具体尺寸。元素计数构建块第8.1.1节为

定义为

$$t_\alpha = \sum_{i=1}^{N} \delta^{\alpha}_{v_i}. \quad (62)$$

If we want the partition α to have L elements, then

如果我们希望分区α有L个元素，那么

该目标函数以相同的方式惩罚所有可能的切割，无论切割的顶点数量或分区数量如何

的。

7.1.5 Clique cover

a. *描述给定图G=（V，E），我们寻求对所有顶点着色的最小颜色数K，使得具有颜色α的顶点的子集Wα与限制在Wα中的顶点之间的边的边集Eα一起形成完整图。一个子问题是决定是否存在使用K颜色的集团覆盖。*

Variables For each vertex $i = 1, \ldots, N$ we define variables $v_i = 1, \ldots, K$ indicating the color that vertex is assigned to. If the number of colors is not given, one has to start with an initial guess or a minimal value for K.

b. *变量对于每个顶点i =1。，N，我们定义变量vi =1，*..，K表示顶点所分配的颜色。如果没有给出颜色的数量，则必须从K的初始猜测或最小值开始。

Constraints In this problem, $G_\alpha = (W_\alpha, E_\alpha)$ has to be a complete graph, so the maximum number of edges in E_α must be present. Using the element counting building block, we can calculate the number of vertices with color α (see bulding block Section 8.1.1):

c. *约束在这个问题中，Gα=（Wα，Eα）必须是一个完整的图，因此Eα中的最大边数必须存在。使用元素计数构建块，我们可以计算颜色为α的顶点数量（见构建块第8.1.1节）：*

$$t_\alpha = \sum_{i=1}^{N} \delta^{\alpha}_{i}. \quad (68)$$

if G_α is a complete graph, then the number of edges in E_α is $t_\alpha(t_\alpha - 1)/2$ and thus the constraint reads

因此约束读取

$$c = \sum_{\alpha=1}^{K}\left(\frac{t_\alpha(t_\alpha-1)}{2} - \sum \ldots\right) \quad (63)$$

$= 0.$

$(i,j) \in E$

$(i, j) \in E$

(69)

i

Note that we do not have to square the term as it can never be negative.

请注意，我们不必对该项求平方，因为它永远不可能是负数。

Cost function The decision problem ("Is there a clique cover using K colors?") can be answered using the constraint c as the Hamiltonian of the problem. For finding the minimum number of colors $K_{\min}$ for which a clique cover exists (the *clique cover number*), we can add a cost function for minimizing K. As in the graph coloring problem, we can minimize the number of colors using

d. *成本函数*决策问题（是否存在使用K色的团覆盖？"）可以使用约束c作为问题的哈密顿量来回答。为了找到存在集团覆盖的最小颜色数K_{min}（*集团覆盖数*），我们可以添加一个最小化K的代价函数。在图着色问题中，我们可以使用

$$f = \sum_{\alpha} (1 - u_\alpha), \tag{70}$$

where

哪里

$$u_\alpha = \prod_{i=1}^{N} (1 - \delta_{v_i}^{\alpha}) \tag{71}$$

不及物动词

of a single spin flip this is prevented as long as $a_1 \cup a_2 \Delta$, where Δ is the maximal degree of G and $a_{1,2}$ are the energy scales of the first and second constraints.

indicates if the color α is used or not (see building block Section 8.2.2).

指示是否使用了 颜色α（见构建块第8.2.2节）。

e. *References* The one-hot encoded Hamiltonian of the decision problem can be found in Lucas (2014).

e.*参考文献*决策问题的一个热门编码哈密顿量可以在Lucas (2014) 中找到。

Constrained subset problems

7.2 约束子集问题

Given a set U, we look for a non-empty subset $U_0 \subseteq U$ that minimizes a cost function f while satisfying a set of constraints c_i. In general, these problems require a binary variable x_i per element in U which indicates if the element i is included or not in the subset U_0. Although binary variables are trivially encoded in single qubits, non-binary auxiliary variables may be necessary to formulate constraints, so the encoding-independent formulation of these problems is still useful.

给定一个集合U，我们寻找一个非空子集$U0 \subseteq U$，它在满足一组约束c的同时最小化成本函数f。通常，这些问题需要U中每个元素有一个二进制变量xi，它指示元素i是否包括在子集$U0$中。尽管二进制变量在单个量子位中被平凡地编码，但非二进制辅助变量可能是制定约束所必需的，因此这些问题的独立编码公式仍然有用。

对于单个自旋翻转，只要a1Ua2Δ，这是可以防止的，其中Δ是G的最大程度，$a1,2$是第一和第二约束的能阶。

d. *Cost function* The decision problem ("Is there a clique of size K?") can be solved using the constraints as the Hamiltonian of the problem. If we want to find the largest clique of the graph G, we must encode K as a discrete variable, $K = 1, \ldots, K_{\max}$, where $K_{\max} = \Delta$ is the maximum degree of G and the largest possible size of a clique. The cost function for this case is simply the value of K:

d.*成本函数*决策问题（是否存在大小为K的团？"）可以使用约束作为问题的哈密顿量来求解。如果我们想找到图G的最大团，我们必须将K编码为离散变量，K=1。，$Kmax$，其中$Kmax$=Δ是G的最大程度和团的最大可能大小。这种情况下的成本函数只是K的值：

$$f = -K \tag{74}$$

e. *Resources* Implementing constraints as energy penalizations, the total cost function for the decision problem (fixed K),

e.*资源*作为能量惩罚的实施约束、决策问题的总成本函数（固定K），

$$f = (c_1 - K)^2 + \left(c_2 - \frac{K(K-1)}{2}\right)^2, \tag{75}$$

$U0$中。尽管二进制变量在单个量子位中被平凡地编码，但非二进制辅助变量可能是制定约束所必需的，因此这些问题的独立编码公式仍然有用。

Cliques

7.2.1 派系

Description A clique on a given graph $G = (V, E)$ is a subset of vertices $W \subseteq V$ such that W and the subset E_W of edges between vertices in W is a complete graph, i.e., the maximal possible number of edges in E_W is present. The goal is to find a clique with cardinality K. Additionally, one could ask what the largest clique of the graph is.

a. *描述*给定图G=（V，E）上的团是顶点$W \subseteq V$的子集，使得W和W中顶点之间的边的子集EW是一个完整图，即EW中存在最大可能的边数。目标是找到一个基数为K的团。此外，可以问图中最大的团是什么。

Variables We can define $|V|$ binary variables x_i that indicate whether vertex i is in the clique or not.

b. *变量*我们可以定义|V|二进制变量xi，指示顶点i是否在团中。

Constraints This problem has two constraints, namely, that the cardinality of the clique is K and that the clique indeed has the maximum number of edges. The former is enforced by

c. *约束*这个问题有两个约束，即团的基数是K，团确实有最大的边数。前者由执行

$|V|$

$$c_1 = \quad x_i = K \tag{72}$$

has interaction terms with maximum order of two and the number of terms scales with $|E| + |V|^2$. If K is encoded as a discrete variable, the resources depend on the chosen encoding.

i=1

and the latter by

后者由

$$c_2 = \sum_{(i,j)\in E} x_i x_j = \frac{K(K-1)}{2}. \tag{73}$$

2.

具有最大阶为2的相互作用项，并且项的数量与|*E*|+|*V*|²成比例。如果*K*被编码为离散变量，则资源取决于所选择的编码。

f. *References* This Hamiltonian was formulated in Lucas (2014) using one-hot encoding.

f.参考文献Lucas（2014）中使用一种热编码对该哈密顿量进行了公式化。

Maximal independent set

7.2.2 极大独立集

Description Given a hypergraph $G = (V, E)$ we look for a subset of vertices $S \subset V$ such that there are no edges in E connecting any two vertices of S. Finding the largest possible S is an NP-hard problem.

a. *描述*给定超图*G*=（*V*，*E*），我们寻找顶点*S*⊂*V*的子集，使得在*E*中没有边连接*S*的任何两个顶点。找到最大可能的*S*是一个NP难题。

Variables We use a binary variable x_i for each vertex in V.

b. *变量*我们为*V*中的每个顶点使用一个二元变量*xi*。

Cost function For maximizing the number of vertices in S, the cost function is

c. *成本函数*为了最大化*S*中的顶点数量，*成本函数*为

$$f = -\sum_{i=1}^{|V|} x_i. \tag{76}$$

Constraints Given two elements in S, there must not be any edge of hyperedge in E connecting them. The constraint

d. *约束*给定*S*中的两个元素，在*E*中不存在任何连接它们的超边。约束条件

$$c = \sum_{i,j\in V} A_{i,j} x_i x_j \tag{77}$$

$i,j\in V$

i、j∈V

counts the number of adjacent vertices in S, with A as the adjacency matrix. By setting $c = 0$, the vertices in S form an independent set.

计算S中相邻顶点的数量，*A*为邻接矩阵。通过设置*c*=0，*S*中的顶点形成独立集。

References See Lucas (2014) and Choi (2010) for graphs.

e. *参考文 献* 见图Lucas（2014）和Choi（2010）。

Set packing

7.2.3 设置填料

Description Given a set U and a family $S = \{V_i\}^N$ of subsets V_i of

a. *描述*给定一个集合*U*和一个族*S*={*Vi*}ᴺ的 子集*Vi*

U, we want to find set packings, i.e., subsets of S such that all subsets are pairwise disjoint, $V_i \cap V_j = \varnothing$. Finding the maximum packing (the maximum number of subsets V_i) is the NP-hard optimization problem called set packing.

U、我们想找到集合包装，即*S*的子集，使得所有子集成对不相交，V_i å*Vj*=∅。求最大封装（子集的最大数量*Vi*）是称为集合封装的NP难优化问题。

If constraints are implemented as energy penalization, one has to ensure that the first constraint is not violated to decrease the penalty for the second constraint. Using the cost/gain analysis (Section 8.4.2)

如果约束被实现为能量惩罚，则必须确保第一约束不被违反以减少对第二约束的惩罚。使用成本/收益分析（第8.4.2节）

Cost function Maximizing the number of subsets in the packing is achieved with the element counting building block

Variables We define N binary variables x_i that indicate whether subset V_i belongs to the packing.

b. *变量*我们定义*N*个二进制变量*xi*，表示子集*Vi*是否属于包装。

c. *成本函数*最大化包装中的子集数量是通过元素计数构建块实现的

N

$$f = -\sum_{i=1}^{N} x_i. \tag{78}$$

Constraints In order to ensure that any two subsets of the packings are disjoint we can impose a cost on overlapping sets with

d. 约束为了确保包装的任意两个子集是不相交的，我们可以对具有

Constraints We have to enforce that C is indeed a matching, i.e., that no two edges which share a vertex belong to C. Using an energy penalty, this is achieved by:

$i,j:V_i\cap V_j\neq\varnothing$

$$c = \sum_{i,\ j:V_i\cap V_j\neq\varnothing} x_i x_j = 0. \tag{79}$$

Resources The total cost function $H = H_A + H_B$ has interaction terms with maximum order of two (so it is a QUBO problem) and the number of terms scales up to N^2.

e. 资源总成本函数$H=HA+_{HB}$具有最大阶为2的交互项（因此这是一个QUBO问题），并且项的数量可扩展到$N2$。

References This problem can be found in Lucas (2014).

f. 参考文献这个问题可以在Lucas（2014）中找到。

Vertex cover

7.2.4 顶点覆盖

Description Given a hypergraph $G = (V, E)$ we want to find the smallest subset $C \subseteq V$ such that all edges contain at least one vertex in C.

a. 描述给定超图$G=$（V，E），我们想找到最小子集$C\subseteq V$，使得所有边在C中至少包含一个顶点。

Variables We define $|V|$ binary variables x_i that indicate whether vertex i belongs to the cover C.

b. 变量我们定义$|V|$二进制变量xi，表示顶点i是否属于覆盖C。

Cost function Minimizing the number of vertices in C is achieved with the element counting building block

c. 成本函数最小化C中的顶点数量是通过元素计数构建块实现的

$$f = \sum_{i=1}^{|V|} x_i. \tag{80}$$

Constraints With

d. 约束条件

k

$$c = \sum_{e=(u1,\ldots,uk)\in E} \prod_{a=1}^{k} (1 - x_{u_a}) = 0 \tag{81}$$

one can penalize all edges that do not contain vertices belonging to

可以惩罚不包含属于的顶点的所有边

C. Encoding the constraint as an energy penalization, the Hamiltonian results in

C.将 约束编码为能量惩罚，Hamiltonian结果如下

$$H = Af + Bc. \tag{82}$$

by setting $B > A$ we can avoid the constraint being traded off against the minimization of C.

$$f_A = A\sum_{i=1}^{|E|} x_i. \tag{83}$$

d. 约束我们必须强制C确实是匹配的，即没有共享一个顶点的两条边属于C。使用能量惩罚，这可以通过以下方式实现：

$$f_B = B\sum \quad \sum \quad x_i x_j = 0, \tag{84}$$

通过设置$B>A$，我们可以避免约束与C的最小化进行权衡。

Resources The maximum order of the interaction terms is the maximum rank of the hyperedges k and the number of terms scales with $|E|2^k + |V|$.

e. 资源交互项的最大阶数是超边k的最大秩，项的数量与$|E|^{2k}+|V|$成比例。

References The special case that only considers graphs can be found in Lucas (2014).

f. 参考文献只考虑图的特殊情况可以在Lucas（2014）中找到。

Minimal maximal matching

7.2.5 最小-最大匹配

Description Given a hypergraph $G = (V, E)$ with edges of maximal rank k we want to find a minimal (i.e., fewest edges) matching $C \subseteq E$ which is maximal in the sense that all edges with vertices that

a. 描述给定具有最大秩k的边的超图$G=$（V，E），我们想找到匹配$C\subseteq E$的最小（即，最小边），它在所有具有顶点的边

are not incident to edges in C have to be included in the matching.

不入射到C中的边缘必须包括在匹配中。

Variables We define $|E|$ binary variables x_i that indicate whether an edge belongs to the matching C.

b. 变量我们定义了$|E|$二进制变量xi，表示一条边是否属于匹配的C。

Cost function Minimizing the number of edges in the matching is simply done by the cost function

c. 成本函数最小化匹配中的边数简单地通过成本函数来实现

$v\in V \ (i,j)\in\partial v$

$v\in v\quad (i,\ j)\quad \in\partial v$

where ∂v is the set of edges connected to vertex v. Additionally, the matching should be maximal. For each vertex u, we define a variable $y_u = \sum_{i\in\partial u} x_i$ which is only zero if the vertex does not belong to an edge of C. If the first constraint is satisfied, this variable can only be 0 or 1. In this case, the constraint can be enforced by

其中? v是连接到顶点v的边的集合。此外，匹配应该是最大的。对于每个顶点u，我们定义了一个变量$yu=_{i\in\partial uxi}$，如果该顶点不属于C的边，则该变量仅为零。如果满足第一个约束，则此变量只能为0或1。在这种情况下，可以通过

$$f_C = C \sum_{e=(u1,\dots,uk)\in E} \prod_{a=1}^{k} \left(1 - y_{u_a}\right) = 0. \tag{85}$$

However, one has to make sure that the constraint implemented by f_B is not violated in favor of f_C which could happen if for some v, $y_v > 1$ and for m neighboring vertices $y_u = 0$. Then the contributions from v are given by

然而，必须确保由$_{fB}$实现的约束不被违反以有利于fC，如果对于一些v，yv>1并且对于m个相邻顶点yu=0，则可能发生这种情况。那么v的贡献由

$$f = By\,(y-1)^{1} + C(1-y)m \tag{86}$$

and since $m + y_v$ is bounded by the maximum degree Δ of G times the maximum rank of the hyperedges k, we need to set $B > (\Delta k - 2)C$ to ensure that the ground state of $f_B + f_C$ does not violate the first constraint. Finally, one has to prevent f_C being violated in favor of f_A which entails $C > A$.

并且由于$m+_{yv}$受G的最大度Δ乘以超边k的最大秩的限制，我们需要设置B> (Δk−2) C以确保$fB+_{fC}$的基态不违反第一约束。最后，必须防止违反fC而有利于fA，这就导致了$_C$>A。

Resources The maximum order of the interaction terms is k and the number of terms scales roughly with $(|V| + |E|2^k)\Delta(\Delta - 1)$.

e. *资源*相互作用项的最大阶数为k，项的数量大致与(|V|+|E|^{2k}) Δ (Δ−1) 成比例。

References This problem can be found in Lucas (2014).

f. *参考文献*这个问题可以在Lucas（2014）中找到。

Set cover

7.2.6 设置封面

Description Given a set $U = \{u_\alpha\}^n_{\alpha=1}$ and N subsets $V_i \subseteq U$, we look for the minimum number of V_i such that $U = \cup V_i$.

a. *描述*给定一个集U={$U\alpha$}^{n}和n个子集$Vi\subseteq U$，我们寻找Vi的最小数量，使得U=∪Vi。

Variables We define a binary variable $x_i = 0, 1$ for each subset V_i that indicates if the subset V_i is selected or not. We also define auxiliary variables $y_\alpha = 1, 2, \dots, N$ that indicate how many active subsets (V_i such that $x_i = 1$) contain the element u_α.

b. *变量*我们为每个子集Vi定义一个二进制变量xi =0，1，该变量指示子集Vi是否被选择。我们还定义了辅助变量$y\alpha$=1，2。，N表示有多少个活动子集（Vi使得xi =1）包含元素$u\alpha$。

Cost function The cost function is simply

c. *成本函数成本函数*很简单

$$f = \sum_{i=1}^{N} x_i, \tag{87}$$

which counts the number of selected subsets V_i.

其对所选子集Vi的数量进行计数。

Constraints The constraint $U = \cup_{i:\,x_i=1} V_i$ can be expressed as

d. *约束约束*U=∪i$_{:xi=1Vi}$可以表示为

$$y_\alpha > 0,\ \forall\alpha = 1, \dots, n, \tag{88}$$

which implies that every element $u_\alpha \in U$ is included at least once. These inequalities are satisfied if y_α are restricted to the valid values ($y_\alpha = 1, 2, \dots, N$, $y_\alpha \neq 0$) (Section 8.2.3). The values of y_α should be consistent with those of x_i, so the constraint is

这意味着每个元素$u\alpha\in u$至少包含一次。如果$y\alpha$被限制为有效值（$y\alpha$=1，2，...，N，$y\alpha\neq 0$），则满足这些不等式（第8.2.3节）。$y\alpha$的值应与xi的值一致，因此约束为

$ww \quad 2^v \quad -$

$\alpha=$

$$c_\alpha = y_\alpha - \sum_{i:\, u_\alpha \in V_i} x_i = 0. \tag{89}$$

Special case: exact cover If we want each element of U to appear once and only once on the cover, then $y_\alpha = 1$, for all α and the constraint of the problem reduces to

e. *特殊情况：精确覆盖*如果我们希望*U*的每个元素在覆盖上出现一次并且只出现一次，那么$y_\alpha=1$，对于所有的α，并且问题的约束减少到

$$c_\alpha = \sum_{i:\, \alpha \in V_i} x_i = 1. \tag{90}$$

c. *Cost function* For this problem, there is no cost function, so every permutation that satisfies the constraints is a solution to the problem.

f. *References* The one-hot encoded Hamiltonian can be found in Lucas (2014).

f.*参考文献*Lucas（2014）中有一个热门的编码哈密顿量。

Knapsack

7.2.7 背包

Description A set U contains N objects, each of them with a value d_i and a weight w_i. We look for a subset of U with the maximum value d_i such that the total weight of the selected objects does

a. *描述*一个集合*U*包含*N*个对象，每个对象都有一个值*d*和一个权重*wi*。我们寻找具有最大值di*的U*的子集，使得所选对象的总权重

not exceed the upper limit W.

不超过上限*W*。

Variables We define a binary variable $x_i = 0, 1$ for each element in

b. *变量*我们为中的每个元素定义一个二进制变量$x_i=0$，1

U that indicates if the element i is selected or not. We also define an auxiliary variable y that indicates the total weight of the selected objects:

U，其指示元素*i*是否被选择。我们还定义了一个辅助变量*y*，该变量表示所选对象的总重量：

$$y = \sum_{i:\, x_i=1} w_i. \tag{91}$$

if the weights are natural numbers $w_i \in \mathbb{N}$ then y is also natural, and the encoding of this auxiliary variable is greatly simplified.

如果权重是自然数$w_i\in N$，那么*y*也是自然的，并且大大简化了这个辅助变量的编码。

Cost function The cost function is given by

c. *成本函数成本函数*由

$$f = -\sum_{i=1}^{N} d_i x_i, \tag{92}$$

c.*代价函数*对于这个问题，不存在代价函数，因此每个满足约束的排列都是问题的解。

d. *Constraints* This problem requires two constraints. The first constraint is inherent to all permutation problems and imposes the $|V|$ variables $\{v_i\}$ to be a permutation of $[1, \ldots,$

d.*约束条件*这个问题需要两个约束条件。第一个约束是所有置换问题所 固有的并且将|*V*|变量$\{v_i\}$强制为[1。，

$|V|]$. This is equivalent to requiring $v_i \neq v_j$ if $i \neq j$, which can be

等价于当$i\neq j$时要求$v_i\neq v_j$，这可以是

encoded with the following constraint (Section 8.2.1):

使用以下约束进行编码（第8.2.1节）：

$$c_1 = \sum_{\alpha=1}^{N} \sum_{i \neq k} \delta^\alpha_{v_i} \delta^\alpha_{v_k} = 0. \tag{94}$$

The second constraint ensures that the path only goes through the edges of the graph. Let A be the adjacency matrix of the graph, such that $A_{i,j} = 1$ if there is an edge connecting nodes i and j and zero otherwise. To penalize invalid solutions, we use the constraint

第二个约束确保路径仅通过图形的边。设*A*是图的邻接矩阵，使得如果存在连接节点*i*和*j*的边，则$A_{i,j}=1$，否则为零。为了惩罚无效的解决方案，我们使用约束

$$c_2 = \sum_{i,j} \sum_{\alpha=1}^{|V|} (1 - A_{i,j})(\delta^\alpha_{v_i} \delta^{\alpha+1}_{v_j} + \delta^{\alpha+1}_{v_i} \delta^\alpha_{v_j}) = 0, \tag{95}$$

which counts the value of the selected elements.

其对所选元素的值进行计数。

Constraints The constraint $y < W$ is implemented by forcing y to take one of the possible values $y = 1, \ldots W - 1$ (see Section 8.2.3). The value of y must be consistent with the selected items from U:

d. *约束条件*$y<W$是通过强制*y*取一个可能的值$y=1$来实现的。$W-1$（见第8.2.3节）。*y*的值必须与从*U*中选择的项目一致：

$$c = y - \sum_i w_i x_i = 0. \tag{93}$$

References The one-hot encoded Hamiltonian can be found in

e. *参考文献*一个热编码的哈密顿量可以在

which counts how many adjacent nodes in the solution are not connected by an edge in the graph. $\alpha = |V| + 1$ represents $\alpha = 1$ since we are looking for a closed path.

它统计解决方案中有多少相邻节点没有通过图中的边连接。

$\alpha=|V|+1$表示$\alpha=1$，因为我们正在寻找一条闭合路径。

e. *References* The one-hot encoded Hamiltonian can be found in Lucas (2014).

e.*参考文献*Lucas（2014）中有一个热门的编码哈密顿量。

7.3.2 Traveling salesperson problem (TSP)

7.3.2旅行销售人员问题（TSP）

Description The TSP is a trivial extension of the Hamiltonian cycles problem. In this case, the nodes represent cities and the edges are the possible roads connecting the cities, although in general it is assumed that all cities are connected (the graph is complete). For each edge connecting cities i and j there is a cost $w_{i,j}$. The solution of the TSP is the Hamiltonian cycle that minimizes the total cost $w_{i,j}$.

a. *描述*TSP是哈密顿循环问题的一个平凡扩展。在这种情况下，节点代表城市，边缘是连接城市的可能道路，尽管通常假设所有城市都是连接的（图是完整的）。对于连接城市i和j的每个边缘，都有一个成本$w_{i,j}$。TSP的解是最小化总成本$w_{i,j}$的哈密顿循环。

Variables We define a variable $v_i = 1, \ldots, |V|$ for each node in the graph, indicating the position of the node in the permutation.

b. *变量*我们定义一个变量$v_i = 1, \ldots, |V|$用于图中的每个节点，指示节点在排列中的位置。

Cost function If the traveler goes from city i at position α to city j in the next step, then

c. *成本函数*如果旅行者从位置α的城市i到城市j在下一步中，然后

$$\delta^{\alpha}_{v_i}\delta^{\alpha+1}_{v_j} = 1, \tag{96}$$

otherwise, that expression would be zero. Therefore the total cost of the travel is codified in the function:

否则，该表达式将为零。因此，差旅的总成本被编入函数：

$$f_w = \sum_{i,j} \sum_{\alpha\, i<j} w_{i,j} \left(\delta^{\alpha}_{v_i}\delta^{\alpha+1}_{v_j} + \delta^{\alpha}_{v_j}\delta^{\alpha+1}_{v_i}\right). \tag{97}$$

对于哈密顿循环，$\alpha=|V|+1$表示$\alpha=1$，因为我们正在寻找一条闭合路径。

Constraints The constraints are the same as those used in the Hamiltonian cycles problem (see paragraph 7.3.1). If the graph is complete (all the cities are connected) then constraint c_2 is not necessary.

d. *约束*约束与哈密顿循环问题中使用的约束相同（见第7.3.1段）。如果该图是完整的（所有城市都是连接的），则约束c_2是不必要的。

References The one-hot encoded Hamiltonian can be found in Lucas (2014).

e. *参考文献*Lucas（2014）中有一个热门的编码哈密顿量。

7.3.3 Machine scheduling

7.3.3机器调度

a. *Description* Machine scheduling problems seek the best way to distribute a number of jobs over a finite number of machines.

a.*描述*机器调度问题寻求在有限数量的机器上分配多个作业的最佳方式。

Permutation problems

7.3 置换问题

In permutation problems, we need to find a permutation of N elements that minimizes a cost function while satisfying a given set of constraints. In general, we will use a discrete variable $v_i \in [1, N]$ that indicates the position of the element i in the permutation.

在置换问题中，我们需要找到一个N个元素的置换，使成本函数最小化，同时满足给定的一组约束。通常，我们将使用离散变量$v_i \in [1, N]$，它指示元素i在排列中的位置。

Hamiltonian cycles

7.3.1 哈密顿循环

Description For a graph $G = (V, E)$, we ask if a Hamiltonian cycle, i.e., a closed path that connects all nodes in the graph through the existing edges without visiting the same node twice, exists.

a. *描述*对于图$G=(V, E)$，我们问是否存在哈密顿循环，即通过现有边连接图中所有节点而不访问同一节点两次的闭合路径。

Variables We define a variable $v_i = 1, \ldots, |V|$ for each node in the graph, that indicates the position of the node in the permutation.

b. *变量*我们定义一个变量$v_i = 1, \ldots, |V|$对于图中的每个节点，表示节点在排列中的位置。

as for Hamiltonian cycles, $\alpha = |V| + 1$ represents $\alpha = 1$ since we are looking for a closed path.

These problems explore permutations of the job list, where the position of a job in the permutation indicates on which machine and at what time the job is executed. Many variants of the problem exist, including formulations in terms of spin Hamiltonians (Venturelli et al., 2015; Kurowski et al., 2020; Amaro et al., 2022). Here we consider the problem of M machines and N jobs, where all jobs take the same amount of time to complete, so the time can be divided into time slots of equal duration t. It is possible to include jobs of duration nt

这些问题探讨了作业列表的排列，其中作业在排列中的位置指示在哪台机器上以及在什么时间执行作业。该问题存在许多变体，包括自旋哈密顿量的公式（Venturelli等人，2015；Kurowski等人，2020；Amaro等人，2022）。在这里，我们考虑M台机器和N个作业的问题，其中所有作业都需要相同的时间来完成，因此时间可以划分为相等持续时间t的时隙。可以包括持续时间nt的作业

($n \in \mathbb{N}$) by using appropriate constraints that force some jobs to

（$n \in n$），通过使用适当的约束来强制一些作业

run in consecutive time slots on the same machine. In this way,

在同一台机器上的连续时隙中运行。通过这种方式，

problems with jobs of different duration can be solved by

不同持 续时间的工作问题可以通过以下方式解决

that reduces the energy of any solution in which job j_2 is performed immediately after job j_1 on the same machine m. This constraint allows the encoding of problems with different job durations since j_1 and j_2 can be considered part of the same job of duration $2t$. choosing a sufficiently short time t.

选择足够短的时间t。

b. *Variables* We define variables $v_{m,t} = 0, \ldots N$, where the subindex

b.变量我们定义变量$v_{m,t}$=0。N、 其中子索引

$m = 1, \ldots M$ indicates the machine and $t = 1, \ldots T$ the time slot. When $v_{m,t} = 0$, the machine m is unoccupied in time slot t, and if $v_{m,t} = j \neq 0$ then the job j is done in the machine m, in the time slot t.

m=1。M表示机器，$t=1$。T时隙。当$v_{m,t}$=0时，机器m在时隙t中未被占用，并且如果$v_{m,t}=j\neq 0$，则作业j在机器m中在时隙t内完成。

c. *Constraints* There are many possible constraints depending on the use case we want to run. As in every permutation problem, we require that no pair of variables have the same value, $v_{m,t} \neq v_{m',t'}$,

c.约束根据我们想要运行的用例，有许多可能的约束。正如在每个置换问题中一样，我们要求没有一对变量具有相同的值，

v≠v，

These functions can be generated from $v_{m,t}$:

for all m, t.

这减少了在同一机器m上紧接在作业$j1$之后执行作业$j2$的任何解决方案的能量。由于$j1$和$j2$可以被视为具有持续时间$2t$的相同作业的一部分，因此该约束允许对具有不同作业持续时间的问题进行编码。

d. *Cost function* Different objective functions can be chosen for this problem, such as minimizing machine idle time or early and late deliveries. A common option is to minimize the makespan, i.e., the time slot of the last scheduled job. To do this, we first slots in which a job has been scheduled:

d.*成本函数*对于这个问题可以选择不同的目标函数，例如最小化机器空闲时间或提前和延迟交货。一个常见的选项是最小化完工时间，即最后一个计划作业的时间段。要做到这一点，我们首先要在其中安排作业：

introduce an auxiliary function $\tau(v_{m,t})$ that indicates the time

引入一个表示时间的辅助函数τ（$v_{m,t}$）

$$\tau(v_{m,t}) = \begin{cases} t & \text{if } v_{m,t} \neq 0 \\ 0 & \text{if } v_{m,t} = 0. \end{cases} \tag{103}$$

$\tau(v_{m,t})$ 0if $v_{m,t}$=0。(103)

这些函数可以从$v_{m,t}$:

$$\tau(v_{m,t}) = (1 - \delta^0_{v_{m,t}}). \tag{104}$$

Note that the maximum value of τ corresponds to the makespan of the problem, which is to be minimized. To do this, we introduce the extra variable $\tau_{\max}$ and penalize configurations where $\tau_{\max} < \tau(v_{m,t})$

注意，τ的最大值对应于问题的完成时间，该完成时间要最小化。为此，我们引入了额外变量τmax和惩罚配置，其中$\tau max<\tau$（$v_{m,t}$）

对 j 所有m，t

otherwise, some jobs would be performed twice. We also require each job to be complete so there must be exactly one $v_{m,t} = j$ for each job j. This constraint is explained in and holds for every job $j \neq 0$:

否则，某些作业将执行两次。我们还要求每个作业都是完整的，因此每个作业必须恰好有一个$vm，t=j$。这个约束在中进行了解释，并适用于每个作业$j\neq 0$:

$$c_1 = \sum_{j\neq 0}\left[\sum_{m,t}\delta^j_{v_{m,t}} - 1\right]^2 = 0. \tag{98}$$

Note that if job j is not assigned (i.e., there are no m, t such that $v_{m,t} = j$) then $c_1 > 0$. Also, if there is more than one variable $v_{m,t} = j$, then again $c_1 > 0$. The constraint will be satisfied ($c_1 = 0$) if and only if every job is assigned to a single time slot on a single machine.

注意，如果作业j没有被分配（即，没有m，t使得$v_{m,t}=j$），则$c1 > 0$。此外，如果存在多个变量$v_{m,t}=j$，则再次$c1>0$。当且仅当每个作业被分配到单个机器上的单个时隙时，将满足约束（$c1 =0$）。

Suppose job k can only be started if another job, j, has been done previously. This constraint can be implemented as

假设只有在先前已经完成了另一个作业j的情况下才能开始作业k。此约束可以实现为

For details on how $\Theta(x)$ can be expressed see Section 8.1.2. By minimizing $f_{\tau_{\max}}$, we ensure that $\tau_{\max} \geq \tau(v_{m,t})$. Then, the cost function is used to simply minimize $\tau_{\max}$, the latest time a job can be scheduled, via

$$c_{2,k>j} = \sum_{m,m',t\geq t'} \delta^j_{v_{m,t}}\delta^k_{v_{m',t'}} = 0, \tag{99}$$

which precludes any solution in which job j is done after job k. Alternatively, it can be codified as

这排除了作业j在作业k之后完成的任何解决方案。或者，它可以被编码为

$$c'_{2,k>j} = \sum_{m,m',t<t'} \delta^j_{v_{m,t}}\delta^k_{v_{m',t'}} = 1. \tag{100}$$

These constraints can be used to encode problems that consider jobs with different operations $O_1, \ldots O_q$ that must be performed in sequential order. Note that constraints c_2, c_2' allow the use of different machines for different operations. If we want job k to be done immediately after work j, we can substitute t' by $t+1$ in c_2'.

这些约束可用于对考虑具有不同操作$O1$、…的作业的问题进行编码。Oq，必须按顺序执行。注意，约束条件$c2$，$c2'$允许使用不同的机器进行不同的操作。如果我们希望工作k在工作j之后立即完成，我们可以用$c2'$中的$t+1$代替t'。

$$f_{\tau_{\max}} = \sum_{m,t}\Theta[\tau(v_{m,t}) - \tau_{\max}], \tag{105}$$

with

具有

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{106}$$

关于如何表达$\Theta(x)$的详细信息，请参见第8.1.2节。通过最小化$f\tau max$，我们确保$\tau max\geq\tau(vm, t)$。然后，使用成本函数来简单地最小化τmax，即可以调度作业的最新时间，通过

$$f = \tau_{\max}. \tag{107}$$

An alternative cost function for this problem is

这个问题的另一个成本函数是

$$f' = \sum_{m,t}\tau(v_{m,t}), \tag{108}$$

which forces all jobs to be scheduled as early as possible.

这迫使所有作业尽可能早地被安排。

e. *References* Quantum formulations of this and related problems can be found in Refs. Venturelli et al. (2015); Kurowski et al.

e.参考文献这一问题和相关问题的量子公式可以在参考文献中找到。Venturelli等人（2015）；Kurowski等人。

If we want two jobs j_1 and j_2 to run on the same machine in consecutive time slots, the constraint can be encoded as

如果我们希望两个作业$j1$和$j2$在连续的时隙中在同一台机器

上运行，则约束可以编码为

(2020); Amaro et al. (2022) and Carugno et al. (2022).

(2020); Amaro等人（2022）和Carugno等人（2022年）。

$$c_3 = \sum_{m,t} \delta^{j1}_{m,t} \, \delta^{j2}_{v_{m,t+1}} = 1, \tag{101}$$

or as a reward term in the Hamiltonian,

或者作为哈密顿量中的奖励项，

$$c_3' = -\sum_{m,t} \delta^{j1}_{m,t} \, \delta^{j2}_{v_{m,t+1}}, \tag{102}$$

7.3.4 Nurse scheduling problem

7.3.4护士排班问题

Description In this problem we have N nurses and D working shifts. Nurses must be scheduled with minimal workload following hard and soft constraints, such as minimum workload $n_{\min,t}$ of nurses i (where nurse i contributes workload p_i) in a given shift t, and balancing the number of shifts to be as equal as possible. Furthermore, no nurse should have to work on more than $d_{\max}$ consecutive days.

a. *描述*在这个问题中，我们有N名护士和D个轮班。护士必须按照硬约束和软约束安排最小工作量，例如给定班次t中护士i的最小工作量 $n_{min、t}$（其中护士i贡献工作量p_i），并平衡

偏移尽可能相等。此外，任何护士都不应该必须连续工作超过d_{max}天。

Variables We define ND binary variables $v_{i,t}$ indicating whether nurse i is scheduled for shift t.

b. *变量*我们定义ND二进制变量$v_{i,t}$，表示是否护士i被安排上班。

Cost function The cost function, whose minimum corresponds to the minimal number of overall shifts, is given by

c. *成本函数*成本函数的最小值对应于总班次的最小数量，由下式给出

D NFor the discrete variables $\tilde{v}_i$ one can use the encodings in Section

$$f = \sum_{t=1} \sum_{i=1} v_{i,t} \quad (109)$$

Constraints Balancing of the shifts is expressed by the constraint

d. *约束偏移的平衡*由约束表示

$$c = \sum_{i<j} \left(\sum_t v_{i,t} - \sum_{t'} v_{j,t'} \right)^2 . \quad (110)$$

3. In the final step after an intermediate solution, $\tilde{v}^*$, is found, one has to map it back to $\mathbb{R}^d$ by uniformly sampling the components of v^* from the hypercuboids corresponding to $\tilde{v}^*$. Any intermediate solution that is valid for the discrete encoding can also be decoded and thus there are no core terms in the cost function besides those from the discrete encoding.

In order to get a minimal workload per shift we introduce the

为了使每次轮班的工作量最小，我们引入了

b. *Random subspace coding* A further possibility to encode a vector

auxiliary variables y_t

将其与价值观绑定

辅助变量yt

which we bind to the values

$$y_t = \sum_{i=1}^{N} p_i v_{i,t} \quad (111)$$

我们

with the penalty terms

带有处罚条款

coordinate direction in $\mathbb{R}^d$ (Devroye et al., 1993). We denote

Rd中的

$$\left(y_t - \sum_{i=1}^{N} p_i v_{i,t} \right)^2 . \quad (112)$$

(116)

Now the constraint takes the form

现在约束采用以下形式

$$c = \sum_{t=1}^{D} \sum \delta_{q_t} = 0. \quad (113)$$

for the projection to the ith coordinate.

A *hyperrectangle* $R \subset \mathbb{R}^d$ is defined as

Note that we can also combine the cost function for minimizing the number of shifts and this constraint by using the cost function

注意，我们也可以通过使用成本函数来组合成本函数，以最小

对于离散变量$v\sim_i$，可以使用第节中的编码

3.在找到中间解$v\sim$*后的最后一步，必须通过从对应于$v\sim$*的超立方体中均匀采样v*的分量，将其映射回$\mathbb{R}^d$。对于离散编码有效的任何中间解也可以被解码，因此在成本函数中除了来自离散编码的核心项之外没有核心项。

b.*随机子空间编码*对矢量进行编码的另一种可能性

$v \in \mathbb{R}^d$ into discrete variables is random subspace coding

离散变量是随机子 空间编码

(Rachkovskii et al., 2005). One starts by randomly choosing a set of coordinates $D_n := \{1, 2, \ldots, d_n\} \subset \{1, 2, \ldots, d\}$. For each $i \in D_n$, a Dirichlet process is used to pick an interval $[a_i, b_i]$ in the i-

（Rachkovskii等人，2005年）。首先随机选择一组坐标Dn:={1,2，...，Dn}⊂{1,2...，d}。对于每个$i\in Dn$，使用狄利克雷过程来选取i中的区间[ai，bi]-

坐标方向（Devroye等人，1993）。我们表示

$$\pi_i : \mathbb{R}^d \to \mathbb{R}$$

$$x = (x_1, \ldots, x_d) \mapsto \pi_i(x) := x_i \quad (116)$$

用于投影到第i个坐标。*超矩形$R\subset Rd$*定义为

$$R := \{x \in \mathbb{R}^d \mid \pi_i(x) \in [a_i, b_i], \forall i \in D_n\}. \quad (117)$$

化班次数量和该约束

For the fixed set of chosen coordinates D_n, the Dirichlet processes are run m times to get m hyperrectangles $\{R_1, \ldots, R_m\}$. For $k = 1, \ldots, m$, we define binary projection maps as

对于所选坐标Dn的固定集合，狄利克雷过程运行m次以

$D N^2$

$$f = \sum p_i v_{i,t} \quad (114)$$

$- n_{\min,t}$

Finally, the constraint that a nurse i should work in maximally $d_{\max}$ consecutive shifts reads

最后，护士我应该在最大限度的dmax连续轮班中工作的约束条件如下

$$c(i) = \sum_{t_1 - t_2 > d_{\max}} \sum_{t'=t_1}^{t_2} v_{i,t'} = 0. \quad (115)$$

References This problem can be found in Ikeda et al. (2019).

e. 参考文献这个问题可以在Ikeda等人（2019）中找到。

7.4 Real variables problems

7.4 实变量问题

Some problems are defined by a set of real variables $\{v_i \in \mathbb{R},\ i = 1, \ldots, d\}$, and a cost function $f(v)$ that has to be minimized. We again want to express these problems as Hamiltonians whose ground state represents an optimal solution. This involves three steps. First, encode the continuous variables into discrete ones. Second, choose an encoding of these discrete variables to spin variables exactly as before, and third, a decoding step is necessary that maps the solution to our discrete problem back to the continuous domain. In the following, we investigate two possible methods one could use for the first step.

一些问题是由一组实变量定义的{y∈R, i=1，...，d}，和一个代价函数f（y），它必须是最小化。我们再次希望将这些问题表示为哈密顿量，其基态表示最优解。这包括三个步骤。首先，将连续变量编码为离散变量。第二，选择这些离散变量的编码，使变量像以前一样旋转。第三，需要一个解码步骤，将离散问题的解决方案映射回连续域。在下文中，我们研究了两种可能用于第一步的方法。

获得m个超矩形$\{R_1, \ldots, R_m\}$。对于k=1。，m、我们将二进制投影映射定义为

$$z_k : \mathbb{R}^d \to \mathbb{Z}_2$$

$$x \mapsto z_k(x) := \begin{cases} 1, & \text{if } x \in R_k \\ 0, & \text{else.} \end{cases} \quad (118)$$

如果x∈Rk，否则为0。

then *Random subspace coding* is defined as a map

然后将随机子空间编码定义为映射

$$z : \mathbb{R}^d \to \mathbb{Z}_2^m$$

$$x \mapsto z(x) := (z_1(x), \ldots, z_m(x)). \quad (119)$$

Depending on the set of hyperrectangles $\{R_1, \ldots, R_n\}$, random subspace coding can be a sparse encoding. Let $M := \{1, \ldots, m\}$ and $K \subset M$. We define sets

根据超矩形$\{R_1, \ldots, R_n\}$的集合，随机子空间编码可以是稀疏编码。设M:={1，...，M}和K⊂M。我们定义集合

$$U(K) := \bigcap_{k \in K} R_k - \bigcup_{i \in M-K} R_i \neq \emptyset. \quad (120)$$

For $z \in \mathbb{Z}_2^m$, let $K(z) = \{k \in \{1, \ldots, m\} \mid z_k = 1\}$. The binary vector z is in the image of the random subspace encoding if and only if

对于z∈Zm，设K（z）={K∈{1，...，m}|zk=1}。二进制矢量z在随机子空间编码的图像中当且仅当

$$U(K(z)) \neq \emptyset \quad (121)$$

holds. From here one can simply use the discrete encodings discussed in Section 3 to map the components $z_i(x)$ of $z(x)$ to spin variables. Due to the potential sparse encoding, a core term has to be added to the cost function. Let $N = \{L \subset M \mid U(L) = \emptyset\}$, then the core term reads

持有。从这里可以简单地使用第3节中讨论的离散编码来将z（x）的分量zi（x）映射到自旋变量。由于潜在的稀疏编码，必须在成本函数中添加一个核心项。设N={L⊂M|U（L）=∅}，则核心项为

1

$$c \stackrel{z_\ell}{=} \sum_{L\in N} \bigotimes_{\ell\in L} \delta_{z_\ell}^{1.} \bigotimes_{p\in M-L} \delta^{0}. \qquad (122)$$

$L \in N \ell \in L$ $\bigotimes_{p\in M-L}$ δ^0 .(122)

a. *Standard discretization* Given a vector $v \in \mathbb{R}^d$ one could simply discretize it by partitioning the axis into intervals with the length of the aspired precision q, i.e., the components of $v = (v_i)$, $i = 1$,

a.*标准离散化*给定向量$v\in Rd$，可以简单地通过将轴划分为具有期望精度q的长度的区间来对其进行离散化，即，$v=$（vi），i=1的分量，

. . ., d are mapped to $v_i \to \tilde{v}_i \in \mathbb{N}$ such that $(\tilde{v}_i - 1)q \le v_i < \tilde{v} q$.

. . ., d映射到vi→ v~i∈N使得（$v\sim_{i-1}$）$q\le vi < v\sim iq$。

Depending on the problem it might also be useful to have different resolutions for different axes or a non-uniform discretization, e.g., logarithmic scaling of the interval length.

根据问题的不同，对于不同的轴或非均匀离散化（例如，间隔长度的对数缩放）具有不同的分辨率也可能是有用的。

Despite this drawback, random subspace coding might be preferable due to its simplicity and high resolution with relatively few hyperrectangles compared to the hypercubes of the standard discretization.

尽管有这个缺点，但由于随机子空间编码的简单性和高分辨率，与标准离散化的超立方体相比，它具有相对较少的超矩形，因此可能是优选的。

Financial crash problem

7.4.1 金融危机问题

Description We calculate the financial equilibrium of market values v_i, $i = 1, \ldots, n$ of n institutions according to a simple

a. *描述*我们计算市场价值的金融均衡vi，i=1。，n个机构中的n个根据简单

model following Elliott et al. (2014) and Orús et al. (2019). In this model, the prices of m assets are labeled by p_k, $k = 1, \ldots, m$.

模型遵循Elliott等人（2014）和Orús等人（2019）。在该模型中，m个资产的价格由p_k标记，k=1。，m。

Furthermore we define the ownership matrix D, where D_{ij} denotes the percentage of asset j owned by i; the cross-holdings C, where C_{ij} denotes the percentage of institution j owned by i (except for self-holdings); and the self-ownership matrix $\tilde{C}$. The model postulates that without crashes the equity values V (such that $v = \tilde{C}V$) in equilibrium satisfy

此外，我们定义了所有权矩阵D，其中D_{ij}表示i拥有的资产j的百分比；交叉持股C，其中C_{ij}表示i拥有的机构j的百分比（自我持股除外）；以及自拥有矩阵$\tilde{C}$。该模型假定在没有崩溃的情况下，均衡中的权益值V（使得$V=\tilde{C}V$）满足

$$V = Dp + CV \rightarrow v = \tilde{C}(1 - C)^{-1}Dp. \tag{123}$$

Crashes are then modeled as abrupt changes in the prices of assets held by an institution, i.e., via

然后，崩溃被建模为价格的突然变化

$$v = \tilde{C}(1 - C)^{-1}(Dp - b(v, p)), \tag{124}$$

where $b_i(v, p) = \beta_i(p)(1 - \Theta(v_i - v^c))$ results in the problem being highly non-linear.

其中b_i（v，p）=β_i（p）（1−Θ（v_i-v^c））导致问题高度非线性。

Variables It is useful to shift the market values and we find a variable $v - v^c = v' \in \mathbb{R}^n$. Since the crash functions $b_i(v, p)$ explicitly depend on the components of v' it is not convenient

b. 变量改变市场价值是有用的，我们发现变量$v-v^c=v'\in\mathbb{R}^n$。由于碰撞函数b_i（v，p）显式地依赖于v'的分量，这是不方便的

to use the random subspace coding (or only for the components individually) and so we use the standard discretization, i.e., each component of v' takes discrete values $0, \ldots, K$ such that with desired resolution r, a cut-off value rK is reached.

使用随机子空间编码（或仅针对单独的分量），因此我们使用标准离散化，即v'的每个分量取离散值0。，使得以期望的分辨率r达到截止值rK。

Cost function In order to enforce that the system is in financial equilibrium we simply square Eq. 124

c. *成本函数*为了使系统处于财务平衡，我们简单地对等式进行平方。124

$$f = (v' + v^c - \tilde{C}(1 - C)^{-1}(Dp - b(v, p)))^2, \tag{125}$$

where for the theta functions in $b(v, p)$ we use

$$\Theta(v'_i) = \sum_{\alpha \geq 0} \delta^{\alpha}_{v'_i} \tag{126}$$

$$y(v, w) = \sum_{r=1}^{k} \sum_{i_1,\ldots,i_r=0,\ldots,\Delta} w_{i_1,\ldots,i_r} v_{i_1} \ldots v_{i_r}, \tag{127}$$

where k is the highest order of the ansatz and the variables v_i depend on the continuous to discrete encoding. In terms of the indicator functions, they are expressed as

其中k是ansatz的最高阶，并且变量v_i取决于连续到离散编码。就指标函数而言，它们表示为

$$v_i = \sum_{\alpha} \alpha \delta^{\alpha}_{v_i}, \tag{128}$$

alternatively, one could write the ansatz as a function of the indicator functions alone instead of the variables

或者，可以将ansatz单独写成指标函数的函数，而不是变量

$$y'(v, w) = \sum_{r=1}^{k} \sum_{i_1,\ldots,i_r=0,\ldots,\Delta} w'_{i_1,\ldots,i_r} \delta^{i_1}_{v_{i_1}} \ldots \delta^{i_r}_{v_{i_r}}. \tag{129}$$

其中，对于b（v，p）中的Θ函数，我们使用

as suggested in Section 8.1.2.

如第8.1.2节所述。

7.4.2 Continuous black box optimization

7.4.2 持续的黑盒优化

Description Given a function $f: [0, 1]^d \rightarrow \mathbb{R}$, which can be evaluated at individual points a finite number of times but

a. *描述*给定函数f:[0，1]^d→R、 其可以在单个点处被评估有限次数， 但是

where no closed form is available, we want to find the global minimum. In classical optimization, the general strategy is to use machine learning to learn an analytical acquisition function $g(x)$ in closed form from some sample evaluations, optimize it to generate the next point on which to evaluate f, and repeat as long as resources are available.

在没有闭合形式可用的情况下，我们希望找到全局最小值。在经典优化中，一般策略是使用机器学习从一些样本评估中学习闭合形式的分析获取函数g（x），对其进行优化以生成下一个评估f的点，并在资源可用的情况下重复。

Variables The number and range of variables depends on the

 10.3389/frqst.2023.1229471

b. *变量变量*的数量和范围取决于

continuous to discrete encoding. In the case of the standard discretization, we have d variables v_a taking values in $\{0, .., \lceil 1/q \rceil\}$, where q is the precision.

连续到离散编码。在标准离散化的情况下，我们有d个变量va，取值为{0，...，⌈1/q}，其中q是精度。

With the random subspace encoding, we have Δ variables v_a taking values in $\{0, .., d_s\}$, where Δ is the maximal overlap of the rectangles and d_s is the number of rectangles.

对于随机子空间编码，我们有Δ变量va，取值为{0，...，ds}，其中Δ是矩形的最大重叠，ds是矩形的数量。

Cost function Similar to the classical strategy, we first fit/learn an acquisition function with an ansatz. Such an ansatz could take the form

c. *成本函数*与经典策略类似，我们首先用模拟来拟合/学习获取函数。这样的仿制品可以采用以下形式

While the number of terms is roughly the same as before, this has the advantage that the energy scales in the cost function can be much lower. A downside is that one has to consider that usually for an optimization to be better than random sampling one needs the assumption that the function f is well-behaved in some way (e.g., analytic). The formulation of the ansatz in Eq. 129 might not take advantage of this assumption in the same way as the first.

虽然项的数量与以前大致相同，但这具有成本函数中的能量尺度可以低得多的优点。缺点是，必须考虑到，通常为了使优化比随机采样更好，需要假设函数f在某些方面表现良好（例如，分析）。公式129中的模拟公式可能不会以与第一个相同的方式利用这一假设。

Let $w^* \equiv \{w_{i_1}{}^*{}_{,...,i_r}\}$ denote the fitted parameters of the ansatz.

设$w^*\neq\{wi1^*, ..., {}_{ir}\}$表示拟子的拟合参数。

Then the cost function is simply $y(v, w^*)$ or $y'(v, w^*)$.

则成本函数简单地为y（v，w^*）或y'（v，w^*）。

Constraints In this problem, the only constraints that can appear are core terms. For the standard discretization, no such term is necessary and for the random subspace coding, we add Eq. 122.

d. *约束*在这个问题中，唯一可以出现的约束是核心项。对于标准离散化，不需要这样的项，并且对于随机子空间编码，我们添加等式。122。

References This problem can be found in Izawa et al. (2022).

e. *参考文献*这个问题可以在Izawa等人（2022）中找到。

Other problems

7.5 其他问题

In this final category, we include problems that do not fit into the previous classifications but constitute important use cases for quantum optimization.

在最后一类中，我们包括了不属于以前分类的问题，但这些问题构成了量子优化的重要用例。

Syndrome decoding problem

7.5.1 综合症解码问题

Description For an $[n, k]$ classical linear code (a code where k logical bits are encoded in n physical bits) the parity check matrix H indicates whether a state y of physical bits is a code word or has a non-vanishing error syndrome $\eta = yH^T$. Given such a syndrome, we want to decode it, i.e., find the most likely error with that syndrome, which is equivalent to solving

a. *描述*对于[n，k]经典线性码（其中k个逻辑位编码在n个物理位中的码），奇偶校验矩阵H指示物理位的状态y是码字还是具有非消失误差综合征$\eta=yHT$。给定这样一个综合征，我们想对其进行解码，即找到该综合征最有可能的 错误，这相当于 求解

$$\underset{e\in\{0,1\}^n, eH^T=\eta}{\operatorname{argmin}} wt(e), \quad (130)$$

$$\underset{e\in\{0,1\}^n,\ eHT=\eta}{\operatorname{argmin}} wt(e), \quad (130)$$

where wt denotes the Hamming weight and all arithmetic is mod 2.

其中wt表示汉明权重，并且所有算术都是mod 2。

Cost function There are two distinct ways to formulate the problem: check-based and generator-based. In the generator-based approach, we note that the generator matrix G of the code satisfies $GH^T = 0$ and thus any logical word u yields a solution to $eH^T = \eta$ via $e = uG + v$ where v is any state such that $vH^T = 0$ can be found efficiently. Minimizing the weight of $uG + v$ leads to the cost function

b. *成本函数*有两种不同的方法来表述问题：基于检查和基于生成器。在基于生成器的方法中，我们注意到代码的生成器矩阵G满足$GHT=0$，因此任何逻辑字u通过$e=uG+v$产生eHT=η的解，其中v是任何状态，使得$vHT=0$可以有效地找到。最小化$uG+v$的权重导致成本函数

$$f_G = \sum_{j=1}^{n}(1-2v_j)\left(1-2\sum_{l=1}^{k}\delta_{u_l} G\right). \quad (131)$$

note that the summation over l is mod 2 but the rest of the equation is over N. We can rewrite this as

注意，l上的总和是mod 2，但方程的其余部分是N上的。我们可以将其重写为

$$f_G = \sum_{j=1}^{n}(1-2v_j)\, G\, s_l^{j} \quad (132)$$

according to Section 8.3.1.

根据第8.3.1节。

In the check-based approach, we can directly minimize deviations from $eH^T = \eta$ with the cost function

在基于检查的 方法中，我们可以使用成本函数直接最小化与eHT=η的偏差

$$f_1 = \sum_{j=1}^{n}(1-2\eta_j)\, G\, s_l^{H_j} \quad (133)$$

but we additionally have to penalize higher weight errors with the term

但我们还必须用该项惩罚更高权重的错误

$$f_2 = \sum_{j=1}^{n} \delta_{e_j} \quad (134)$$

so we have

所以我们有

$$f_H \equiv c_1 f_1 + c_2 f_2 \quad (135)$$

$$\sigma_i(x_{i_1}, \ldots, x_{i_k}) = l_{i,1} \vee \cdots \vee l_{i,k}, \quad (137)$$

where a literal $l_{i,k}$ is a variable $x_{i,k}$ or its negation $\neg x_{i,k}$. We

如果存在满足公式的变量赋值。

if there exists an assignment of the variables that satisfies the formula.

b. *Variables* There are two strategies to express a k-SAT problem in a Hamiltonian formulation. First, one can use a Hamiltonian cost function based on violated clauses. In this case, the variables are the assignments $x \in \{0,1\}^n$. For the second method, a graph is constructed from a k-SAT instance in CNF as follows. Each clause σ_j will be a fully connected graph of k variables $x_{i_1}, \ldots, x_{i_k}$. Connect two vertices from different clauses if they are negations of each other, i.e., $y_{i_l} = \neg y_{j_p}$ and solve the maximum independent set (MIS) problem for this graph (Choi, 2010). If, and only if, this set has cardinality m is the SAT instance satisfiable. The formulation are the $m \times k$ boolean variables x_i which are 1 if the vertex is part of the independent set and 0 otherwise.

b. 变量有两种策略可以在Hamiltonian公式。首先，可以使用Hamiltonian代价基于违反条款的功能。在这种情况下，变量为赋值$x \in \{0,1\}^n$.对于第二种方法，根据CNF中的k-SAT实例构建图，如下所示。每个条款σ将是k个变量$xi1, xik$。连接来自不同子句的两个顶点，如果它们是彼此的否定，求解最大独立集（MIS）这个图的问题（Choi，2010）。如果且仅当，此集合具有公式是$m \times k$布尔变量xi，如果顶点是独立集的一部分，则为1，否则为0。

c. *Cost function* A clause-violation-based cost function for a problem in CNF can be written as

c. 成本函数基于子句违规的成本函数CNF中的问题可以写成

with (Section 8.3.2)

与（第8.3.2节）

$f_C = \Sigma(1$ −

with positive parameters c_1/c_2.

具有正参数$c1/c2$。

Variables The variables in the check-based formulation are the n

c. 变量基于检查的公式中的变量是n

bits e_i of the error e. In the generator-based formulation, k bits u_i of the logical word u are defined. If we use the reformulation Eq. , these are replaced by the k spin variables s_l. Note that the state v is assumed to be given by an efficient classical calculation.

在基于生成器的公式中，定义了逻辑字u的k个比特ui。如果我们使用公式。，这些被k个自旋变量sl代替。注意，假设状态v是由有效的经典计算给出的。

Constraints There are no hard constraints, any logical word u and any physical state e are valid.

d. *约束*没有硬约束，任何逻辑字u和任何物理状态e都是有效的。

Resources A number of interesting tradeoffs can be found by analyzing the resources needed by both approaches. The check-based approach features a cost function with up to

e. *资源*通过分析这两种方法所需的资源，可以找到许多有趣的折衷方案。基于检 查 的方法具有高达的成本函数

$(n - k) + n$ terms (for a non-degenerate check matrix with $n - k$ rows, there are $(n - k)$ terms from f_1 and n terms from f_2), whereas in f_G, there are at most n terms (the number of rows of

（$n-k$） +n项（对于具有$n-k$行的非退化校验矩阵，从$f1$有

（$n-k$）项，从$f2$有n项），而在fG中，最多有n项（的行数

G). The highest order of the variables that appear in these terms is for the generator-based formulation bounded by the number of rows of G which is k. In general, this order can be up to n (number of columns of H) for f_H but for an important class of linear codes (low-density parity check codes), the order would be bounded by the constant weight of the parity checks. This weight can be quite low but there is again a tradeoff because higher weights of the checks result in better encoding rates (i.e., for good low-density parity check codes, they increase the constant encoding rate $n/k \sim \alpha$).

G在这些项中出现的变量的最高阶是基于生成器的公式，该公式由G的行数k限定。通常，对于fH，该阶可以高达n（H的列数），但对于一类重要的线性码（低密度奇偶校验码），该阶将由奇偶校验的恒定权重限定。该权重可能相当低，但也存在折衷，因为较高的检查权重会产生更好的编码率（即，对于良好的低密度奇偶校验码，它们会 增加恒定的 编码率$n/k \sim \alpha$）。

f. *References* This problem was originally presented in Lai et al. (2022).

f.*参考文献*这个问题最初出现在Lai等人（2022）中。

k-SAT

$\sigma_j(x))$ (138)

$$\sigma_j = l_{j1} \vee l_{j2} \ldots \vee l_{jk} \quad (139)$$

$$= 1 - \prod_{n=1}^{k} (1 - l_{jn}). \quad (139)$$

A cost function for the MIS problem can be constructed from a term encouraging a higher cardinality of the independent set:

MIS问题的成本函数可以由鼓励独立集具有更高基数的项来构建：

$$f_B = -b \sum_{j=1}^{m\times k} x_j. \quad (140)$$

Constraints In the MIS formulation, one has to enforce that there are no connections between members of the maximally independent set:

d. *约束*在MIS公式中，必须强制最大独立集合的成员之间没有连接：

$$c = \sum_{(i,j)\in E} x_i x_j = 0. \quad (141)$$

7.5.2 k-SAT

a. *Description* A k-SAT problem instance consists of a boolean formula

a.*描述k-SAT问题实例由布尔公式组成*

$$f(x_1, \ldots, x_n) = \sigma_1(x_{1_1}, \ldots, x_{1_k}) \ldots \wedge \sigma_m(x_{m_1}, \ldots, x_{m_k}) \quad (136)$$

If this constraint is implemented as an energy penalty $f_A = ac$, it should have a higher priority. The minimal cost of a spin flip (*cf.* Section 8.4.2) from f_A is $a(m - 1)$ and in f_B, a spin flip could result in a maximal gain of b. Thus, $a/b \gtrsim 1/(m - 1)$.

如果这个约束被实现为能量惩罚$fA=_{ac}$，那么它应该具有更高的优先级。fA的自旋翻转的最小成本（参见第8.4.2节）为a（$m-1$），而在fB中，自旋翻转可能导致b的最大增益。因此，$a/b \gtrsim 1/(m-2)$。

Resources The cost function f_C consists of up to $O(2^k mk)$ terms with a maximal order of k in the spin variables. In the alternative

e. 资源成本函数fC由旋转变量中最大阶数为k的O（2kmk）项组成。在备选方案中

approach, the order is only quadratic so it would naturally be in a QUBO formulation. However, the number of terms in f_{MIS} can be as high as $O(m^2k^2)$ and thus scales worse in the number of clauses.

方法，阶数只是二次的，所以它自然会在QUBO公式中。

然而，$fMIS$中的项的数量可以高达O（$m2k2$），因此在子句

in the conjunctive normal form (CNF), that is, σ_i are disjunction clauses over k literals l:

在连接范式（CNF）中，即σ是k个文字l上的析取子句：

problems in this library. Similar building blocks are also discussed by Sawaya et al. (2022).

这个库中的

的数量上比例更差。

References This problem can be found in Choi (2010) which

f. 参考文献这个问题可以在Choi（2010）中找到

specifically focuses on the 3-SAT implementation.

特别关注3-SAT的实施。

Summary of building blocks

8 构建块摘要

Here we summarize the parts of the cost functions and

在这里， 我们总结了 成本函数和

techniques that are used as reoccurring building

问题。Sawaya等人也讨论了类似的构建块。（2022）。

Auxiliary functions

8.1 辅助功能

All (connected) variables are different

If we have a set of variables $\{v_i \in [1, K]\}^N$

and two variables v_i,

The simplest class of building blocks are auxiliary scalar functions of multiple variables that can be used directly in cost functions or constraints via penalties.

最简单的构建块类是多个变量的辅助标量函数，可以通过惩罚直接用于成本函数或约束。

v_j are connected when the entry A_{ij} of the adjacency matrix A is 1, thefollowing term has a minimum when $v_i \neq v_j$ for all connected $i \neq j$:

当邻接矩阵A的入口A_{ij}为1时，

Element counting

8.1.1 元素计数

One of the most common building blocks is the function t_a that

最常见的构建块之一是函数t

simply counts the number of variables v_i with a given value a. In

简单地计算具有给定值a的变量vi的数量。In

terms of the value indicator functions, we have

就价值指标函数而言，我们有

$$t_a = \sum_i \delta^a_{v_i}. \tag{142}$$

It might be useful to introduce t_a as additional variables. In that case, one has to bind it to its desired value with the constraints

引入ta作为附加变量可能会很有用。在这种情况下，必须通过约束将其绑定到所需的值

$$c = \sum_a \left(t_a - \sum_i \delta^a_{v_i}\right)^2. \tag{143}$$

The minimum value of c is zero, and it is only possible if and only if there is no connected pair i, j such that $v_i = v_j$. For all-to-all connectivity, one can use this building block to enforce that all variables are different. If $K = N$, the condition $v_i \neq v_j$ for all $i \neq j$ is then equivalent to asking that each of the K possible values is reached by a variable.

Step function

8.1.2 梯级功能

Step functions $\Theta(v - w)$ can be constructed as

阶跃函数Θ（v−w）可以构造为

$$\Theta(v-w) = \sum_{\beta}^{K} \sum_{\alpha} \delta^{\beta}_{v} \delta^{\alpha}_{w} \quad \text{if } v \geq w$$

8.2.1 所有（连接的）变量都不同

如果我们有一组变量$\{vi\in[1，K]\}^N_{i=}$以及两个变量vi，

vj是连通的，当所有连通的$i\neq j$的$vi\neq vj$时，以下项具有最小值：

$$c = \sum_{i \neq j} \delta(v_i - v_j) A_{ij} = \sum_{i\neq j} \sum_{\alpha=1}^{K} \delta^{\alpha}_{v_i} \delta^{\alpha}_{v_j} A_{ij} \tag{147}$$

的最小值为零，并且仅当且仅当不存在连通对i，j使得vi =vj时才有可能。对于所有到所有的连接，可以使用此构建块来强制所有变量都是不同的。如果$K=N$，则所有$i\neq j$的条件$_{vi\neq vj}$等价于要求K个可能值中的每一个都由一个变量达到。

Value α is used

8.2.2 使用值α

Given a set of N variables v_i, we want to know if at least one variable is taking the value α. This is done by the term

给定一组N个变量vi，我们想知道是否至少有一个变量取α值。这是由术语完成的

$$u_\alpha = \prod_i \left(1 - \delta^{\alpha}_{v_i}\right) = \begin{cases} 0 & \text{if } \exists v_i = \alpha \\ 1 & \text{if } \nexists v_i = \alpha. \end{cases} \tag{148}$$

$\Theta(v-w) = \delta_v \delta_w = \quad 0 \quad \text{if } v < w, (144)$

$\theta\ (v-w) = \delta v \delta w = 0 \text{ if } v < w'$ (144)

with K being the maximum value that the v, w can take on. Step functions can be used to penalize configurations where $v \geq w$.

其中K是v，w可以取的 最大值。阶跃函数可以用于惩罚$v \geq w$的配置。

Inequalities

Minimizing the maximum element of a set

8.1.3 最小化集合的最大元素

Step functions are particularly useful for minimizing the

阶跃函数对于最小化

maximum value of a set $\{v_i\}$. Given an auxiliary variable l, we can guarantee that $l \geq v_i$, for all i with the penalization

集合$\{vi\}$的最大值。给定一个辅助变量l，我们可以保证$l \geq vi$，对于所有带有惩罚的i

$$f = 1 - \Theta(l - v_i) \neq \{ \quad \forall i \qquad (145)$$

(145)

$0 \quad \text{if } \exists\, i: v_i > l,$

i0if∃i:

which increases the energy if l is smaller than any v_i. The maximum value of $\{v_i\}$ can be minimized by adding to the cost function of the problem the value of l. In that case, we also have to multiply the term from Eq. 145 by the maximum value that l can take in order to avoid trading off the penalty.

如果l小于任何vi，则会增加能量。通过将l的值添加到问题的成本函数中，可以最小化$\{vi\}$的最大值。在这种情况下，我们还必须将等式145中的项乘以l可以取的最大值，以避免抵消惩罚。

Compare variables

8.1.4 比较变量

Given two variables $v, w \in [1, K]$, the following term indicates if v and w are equal:

给定两个变量v，$w \in [1，K]$，下列项表示v和w是否相等：

$K \quad \alpha\ \alpha 1 \quad \text{if } v = w$

$K\alpha 1$ 如果$v=w$

more severe violations should be penalized more. That option comes with the drawback of introducing in general higher energy scales in the system, especially if K is large, which might decrease the relative energy gap.

$\delta(v-w) = \sum \delta_v \delta_w = \{ \ 0 \quad \text{if } v \neq w$ (146)

$\sum_{\alpha=1} \delta v \delta w$ 0， 如果$v \neq w$ (146)

If we want to check if $v > w$, then we can use the Eq. 144.

8.2.3 不平等

Inequalities of a single variable If a discrete variable v_i, which can take values in $1, \ldots, K$, is subject to an inequality $v_i \leq a'$ one can enforce this with a energy penalization for all values that do not satisfy the inequality

a. *单个变量的不等式*如果离散变量vi可以取1，…中的值。，K、 受到不等式$vi \leq a'$的约束，可以对所有不满足不等式的值进行能量惩罚

$$c = \sum_{K > a > a'} \delta_a^i. \qquad (149)$$

$c = \sum_{K>a>a'} \delta a$。 (149)

it is also possible to have a weighted penalty, e.g.,

也可以具有加权惩罚。，

$$c = \sum_{K \geq a > a'} a \delta_a^{v_i}, \qquad (150)$$

$c = \sum_{K \geq a > a'} a\delta a$， (150)

which might be useful if the inequality is not a hard constraint and

如果不等式不是硬约束，并且

更严重的违规行为应该受到更多的惩罚。这种选择的缺点是在系统中通常引入更高的能量尺度，特别是如果K很大，这可能会减少相对能隙。

Inequality constraints If the problem is restricted by an inequality constraint:

b. *不等式约束*如果问题受到不等式约束的限制：

$c(v_1, \ldots, v_N) < K, (151)$

$c\ (v1, \ldots, vN) < K,$ (151)

it is convenient to define an auxiliary variable $y < K$ and impose the constraint:

定义辅助变量$y<K$并施加约束是方便的：

$(y - c(v, \ldots, v))^2 = 0, (152)$

$(y-c\ (v, \ldots, v))^2 = 0,$ (152)

1 N

等式144。

Constraints

8.2 约束

Here we present the special case of functions of variables where the groundstate fulfills useful constraints. These naturally serve as building blocks for enforcing constraints via penalties.

在这里，我们给出了变量函数的特殊情况，其中基态满足有用的约束。这些自然是通过惩罚来强制执行约束的构建块。

so c is bound to be equal to some value of y, and the only possible values for y are those that satisfy the inequality.

所以c必然等于y的某个值，y的唯一可能值是满足不等式的值。

If the auxiliary variable y is expressed in binary or Gray encoding, then Eq. 152 must be modified when $2^n < K < 2^{n+1}$ for some $n \in \mathbb{N}$,

如果辅助变量y用二进制或格雷编码表示，则当某些$n \in n$的$2n < K < 2n+1$时，必须修改等式152，

$$(y - c(v_1, \ldots, v_N) - 2^{n+1} + K)^2 = 0, \tag{153}$$

$$(y - c(v1, \ldots, v_N) - 2n^{+1+}K)^2 = 0, \tag{153}$$

which ensures $c(v_1, \ldots, v_N) < K$ if $y = 0, \ldots, 2^{n+1} - 1$.

这确保了如果$y = 0$，…，则$c(v1, \ldots, vN) < K$。，$2n+^{1-1}$。

8.2.4 Constraint preserving driver

8.2.4约束保留驱动程序

As explained in Section 5, the driver Hamiltonian must commute with the operator generating the constraints so that it reaches the entire valid search space. Arbitrary polynomial constraints c can for example, be handled by using the Parity

如第5节所述，驾驶员哈密顿量必须与生成约束的算子进行交换，以便到达整个有效搜索空间。例如，任意多项式约束c可以通过使用奇偶校验来处理

mapping. It brings constraints to the form $\sum_{u\in c} g_u \sigma^{(u)}$. Starting in a constraint-fulfilling state and employing constraint depended

映射。它给形式$_{u\in c} g_u\sigma^{(u)}$带来了约束。从满足约束的状态开始并使用依赖于约束的

flip-flop terms constructed from σ_+, σ_- operators as driver terms on the mapped spins (Drieb-Schön et al., 2023) automatically enforces the constraints.

由σ_+、σ_-算子构造的触发器项作为映射自旋上的驱动项（Drieb-Schön et al., 2023）自动执行约束。

theiruniquegroundstate;namely,$f = -G^n\ x_i$ and

$(1 - x_i)$. The two corresponding spin Hamiltonians will

Problem specific representations

8.3 特定于问题的表述

For selected problems that are widely applicable, we demonstrate useful techniques for mapping them to cost functions.

对于广泛适用的选定问题，我们展示了将它们映射到成本函数的有用技术。

Modulo 2 linear programming

8.3.1 模2线性规划

For the set of linear equations

对于线性方程组

$xA = y$,(154)

$$xA=y, \tag{154}$$

where $A \in \mathbb{F}^{l\times n}, y \in \mathbb{F}^n$ are given, we want to solve for $x \in \mathbb{F}^l$. The

当$A\in Fl\times n$, $y\in Fn$时，我们要求解$x\in Fl$。这个

have vastly different properties, as the first one will have, when expressed in spin variables, 2^n terms with up to nth order interactions, whereas the second one has n linear terms. Additionally, configurations that are closer to a fulfilling assignment, i.e., that have more of the x_i equal to one, have a lower cost in the second option which is not the case for the first option. Note that for $x_1 \vee \ldots \vee x_n$ there is no similar trick as we want to penalize exactly one configuration.

22

cost function

成本函数

$$f = \sum_{i=1}^{n} (1 - 2y_i) \left(1 - \sum_{j=1}^{l} 2x_j A_{ji}\right) \tag{155}$$

(155)

It is always possible to convert a k-SAT instance to 3-SAT and, more generally, any boolean formula to a 3-SAT in conjunctive normal form with the Tseytin transformation (Tseitin, 1983) with only a linear overhead in the size of the formula.

通过Tseitin变换（Tseitin，1983），总是可以将k-SAT实例转换为3-SAT，并且更一般地，将任何布尔公式转换为结合正规形式的3-SAT，其中公式的大小只有线性开销。

Note that for the purpose of encoding optimization problems, one might use different expressions that only need to coincide (up to a constant shift) with those shown here for the ground state/solution to the problem at hand. For example, if we are interested in a function in two different ways that both have $x_1 = \ldots = x_n = 1$ as

请注意，为了对优化问题进行编码，可以使用不同的表达式，这些表达式只需要与此处所示的基态/手头问题的解相一致（直到恒定偏移）。例如，如果我们以两种不同的方式对一个函数感兴趣，它们都有$x1=\ldots=xn=1$作为

satisfying assignment for $x_1 \wedge \ldots \wedge x_n$ we might formulate the cost

满足$x1\wedge\ldots\wedge$的赋值xn我们可以计算成本

他们的独特国情；即 $f=-\prod_{i=1}^{n}$ 和 $\sum_{i=1}^{n}(1-x_i)$。两个相应的自旋哈密顿量将

具有大不相同的性质，因为当用自旋变量表示时，第一个具有2n个具有高达n阶相互作用的项，而第二个具有n个线性项。此外，更接近完成任务的配置，即x_i中的更多个等于1的配置，在第二选项中具有较低的成本，而第一选项的情况并非如此。注意，对于$x1\wedge\ldots\wedge xn$没有类似的技巧，因为我们只想惩罚一种配置。

Meta optimization

8.4 元优化

There are several options for choices that arise in the construction of cost functions. These include meta parameters such as coefficients of building blocks but also reoccurring

中出现的选项有几个构建成本函数。其中包括几参数

例如 构建块

minimizes the Hamming distance between xA and y and thus the ground state represents a solution. If we consider the second factor for fixed i, we notice that it counts the number of 1s in x (mod 2) where A_{ji} does not vanish at the corresponding index. When acting with

最小化xA和y之间的汉明距离，因此基态表示解。如果我们考虑固定的第二个因子，我们注意到它计算x（mod 2）中的1的数量，其中A_{ji}不会在相应的索引处消失。与合作时

methods and techniques in dealing with optimization problems.

处理优化问题的

(156)

or the maximal value of a set of variables where we showed a way to bind the value of an auxiliary variable to this function of the set of

on $|x\rangle$ we find the same result and thus the cost function

在|x上，我们得到了相同的结果，从而得到了代价函数

variables. If the function f can be expressed algebraically in terms of the variable values/indicator functions (as a finite polynomial), the

$$f = \sum_{i=1}^{n}(1-2y_i)\prod_{j=1}^{l} s_j^{A_{ji}}, \quad (157)$$

natural way to do this is via adding the constraint term

$$c(y - f(v_i))^2 \quad (163)$$

with spin variables s_j has the solution to Eq. 154 as its ground state.

其中自旋变量s_j具有方程154的解作为其基态。

8.3.2 Representation of boolean functions

8.3.2布尔函数的表示

Given a boolean function f: $\{0,1\}^n \to \{0, 1\}$ we want to express it as a cost function in terms of the n boolean variables. This is hard in general (Hadfield, 2021), but for (combinations of) local boolean functions there are simple expressions. In particular, we have, e.g.,

给定布尔函数f: {0,1}^{n}→｛0，1｝我们希望将其表示为n个布尔变量的成本函数。这通常很难（Hadfield，2021），但对于局部布尔函数的（组合），有简单的表达式。特别地，例如。，

$$\neg x_1 = 1 - x_1 \quad (158)$$

with an appropriate coefficient c. To avoid the downsides of introducing constraints, there is an alternative route that might be preferred in certain cases. This alternative consists of simply using

$$x_1 \wedge x_2 \wedge \ldots \wedge x_n = \prod_{i=1}^{n} x_i \quad (159)$$

的系数，但也重复出现

方法和技术。

Auxiliary variables

8.4.1 辅助变量

It is often useful to combine information about a set of variables v_i into one auxiliary variable y that is then used in other parts of the cost function. Examples include the element counting building block

将关于一组变量v_i的信息组合成一个辅助变量y通常是有用的，该辅助变量y然后用于成本函数的其他部分。示例包括元素计数构建块

或者一组变量的最大值，其中我们展示了一种将辅助变量的值绑定到

变量。如果函数f可以用变量值/指标函数（作为有限多项式）代数表示

自然的方法是添加约束项

$$c\left(y - f(v_i)\right)^2 \quad (163)$$

具有适当的系数c。为了避免引入约束的不利影响，在某些情况下可能会首选一种替代路线。此替代方案包括简单地使用

the variable indicator $\delta^{y(v_i)}_{\beta}$ and expressing the variable indicator function in terms of spin/binary variables according to a chosen encoding (one does not even have to use the same encoding for y and the v_i). As an example, let us consider the auxiliary variable $\tau_{m,t}$ in the machine scheduling problem 7.3.3 for which we need to express

变量指示符$\delta_{\alpha\beta}$，并根据 chosenencoding用$y(v_i, \delta v_i)$变量表示变量指示符函数（甚至不必对y和v_i使用相同的编码）。作为一个例子，让我们考虑机器调度问题7.3.3中的辅助变量$\tau_{m,t}$，我们需要表达它$\delta^{\alpha}_{\tau_{m,t}}$

$\delta^{\alpha}_{\tau_{m,t}}$ encoding for $v_{m,t}$ and (159). For simplicity, we can choose the one-hot

。为了简单起见，我们可 以为vm、t和

$\tau m, t$，导致

12n

i

1 2 n i=1 我 $\pounds(1-x_v$

n

n

$\tau_{m,t}$ leading to

$$x_1 \vee x_2 \vee \ldots \vee x_n = 1 - (1 - x_i) \quad (160)$$

$x1 \wedge x2 \wedge \ldots \wedge xn = 1-$ (1−xi)

(160)

i=1

$$(x_1 \rightarrow x_2) = 1 - x_1 + x_1 x_2 \quad (161)$$

α

$\tau_{m,t m,t}$

α m,t m、t

(164)

$\delta_{\tau m, t} = x_{\tau_{m,t},\alpha} =$ $x_{v_{m,t},0}$ if $t = 0$

$xvm, t,$ 0if$=0$

0 else,

0 其他，

(164)

$\mathrm{XOR}(x_1, x_2) \equiv x_1 + x_2 \mod 2 = x_1 + x_2 - 2x_1x_2$. (162) where the $x_{v_{m,t},\beta}$ are the binary variables of the one-hot encoding of $v_{m,t}$.

XOR（$x1$，$x2$）Select $x1+x2 \bmod 2 = x1+x2-2x1x2$。 (162) 其中$xvm, t, \beta$是$vm, t$的一个热编码的二进制变量。

Prioritization of cost function terms

8.4.2 成本函数项的优先级

If a cost function is constructed out of multiple terms,

如果成本函数由多个项构成，

$$f = a_1 f_1 + a_2 f_2, \tag{165}$$

one often wants to prioritize one term over another, e.g., if the first term encodes a hard constraint. One strategy to ensure that f_1 is not "traded off" against f_2 is to evaluate the minimal cost Δ_1 in f_1 and the maximal gain Δ_2 in f_2 from flipping one spin. It can be more efficient to do this independently for both terms and assume a state close to an optimum. The coefficients can then be set according to $c_1\Delta_1 \gtrsim c_2\Delta_2$. In general, this will depend on the encoding for two reasons. First, for some encodings, one has to add core terms to the cost function which have to be taken into account for the prioritization (they usually have the highest priority). Furthermore, in some encodings, a single spin flip can cause the value of a variable to change by more than one. Nonetheless, it can be an efficient heuristic to compare the "costs" and "gains" introduced above for pairs of cost function terms already in the encoding independent formulation by evaluating them for single variable changes by one and thereby fixing their relative coefficients. Then one only has to fix the coefficient for the core term after the encoding is chosen.

人们通常想要将一个术语优先于另一个术语，例如，如果第一个术语编码硬约束。确保$f1$不与$f2$ 交换"的一种策略是评估$f1$中翻转一个自旋的最小成本Δ1和$f2$中翻转一次自旋的最大增益Δ2。对于这两个项，独立地进行这项操作并假设接近最佳状态可能更有效。然后可以根据$c1\Delta 1\gtrsim c2\Delta 2$设置系数。一般来说，这将取决于编码，原因有两个。首先，对于一些编码，必须将核心项添加到成本函数中，这些项必须在优先级排序时考虑在内（它们通常具有最高优先级）。此外，在某些编码中，一次旋转翻转可能会导致变量的值变化不止一次。尽管如此，通过对单个变量的变化进行逐一评估，从而固定它们的相对系数，来比较上面为独立编码公式中已经存在的成本函数项对引入的 成本"和 收益"，这可能是一种有效的启发式方法。然后，只需要在选择编码之后固定核心项的系数。

Problem conversion

8.4.3 转换问题

Many decision problems associated with discrete optimization problems are NP-complete: one can always map them to any other NP-complete problem with only the polynomial overhead of classical runtime in the system size. Therefore, a new problem without a cost function formulation could be classically mapped to another problem where such a formulation is at hand. However, since quantum algorithms are hoped to deliver at most a polynomial advantage in the run time for general NP-hard problems, it is advisable to carefully analyze the overhead. This hope is based mainly on heuristic arguments related to the usefulness of quantum tunneling (Mandra et al., 2016) or empiric scaling studies (Guerreschi and Matsuura, 2019; Boulebnane and Montanaro, 2022). It is possible that there are trade-offs (as for k-SAT in Section 7.5.2), where in the original formulation the order of terms in the cost function is k while in the MIS formulation, we naturally have a QUBO problem where the number of terms scales worse in the number of clauses.

许多与离散优化问题相关的决策问题都是NP完全的：人们总是可以将它们映射到任何其他NP完全问题，在系统大小中只有经典运行时的多项式开销。因此，没有成本函数公式的新问题可以经典地映射到手头有这种公式的另一个问题。然而，由于量子算法希望在一般NP难题的运行时间内最多提供多项式优势，因此建议仔细分析开销。这种希望主要基于与量子隧道有用性相关的启发式论点（Mandra等人, 2016）或经验标度研究（Guerreschi和Matsuura，2019；Boulebnane和Montanaro，2022）。可能存在权衡（如第7.5.2节中的k-SAT），其中在原始公式中，成本函数中的项的顺序是k，而在MIS公式中，我们自然会遇到QUBO问题，其中项的数量随着条款的数量而变差。

9 Conclusion and outlook

9结论与展望

In this review, we have collected and elaborated on a wide variety of optimization problems that are formulated in terms of discrete variables. By selecting an appropriate qubit encoding for these variables, we can obtain a spin Hamiltonian suitable for quantum algorithms such as quantum annealing or QAOA. The choice of the qubits encoding leads to distinct Hamiltonians, influencing important factors such as the required number of qubits, the order of interactions, and the smoothness of the energy landscape. Consequently, the encoding decisions directly impact the performance of quantum algorithms.

在这篇综述中，我们收集并阐述了各种各样的优化问题，这些问题是用离散变量表示的。通过为这些变量选择合适的量子位编码，我们可以获得适用于量子退火或QAOA等量子算法的自旋哈密顿量。量子位编码的选择导致不同的哈密顿量，影响重要因素，如所需的量子位数量、相互作用的顺序和能量景观的平滑度。因此，编码决策直接影响量子算法的性能。

The encoding-independent formulation (Sawaya et al., 2022) employed in this review offers a significant advantage by enabling

本综述中采用的编码独立配方（Sawaya等人，2022）通过使

3 E.g., in the binary encoding a spin flip can cause a variable shift of up to $K/2$ if the variables take values 1, . . ., K. This is the main motivation to use modifications like the Gray encoding.

3例如，在二进制编码中，如果变量取值1、...、。，K.这是使用格雷编码等修改的主要动机。

the utilization of automated tools to explore diverse encodings, ultimately optimizing the problem formulation. This approach allows for a comprehensive examination and comparison of various encoding strategies, facilitating the identification of the most efficient configuration for a given optimization problem. In addition, the identification of recurring blocks facilitates the formulation of new optimization problems, also constituting a valuable automation tool. By leveraging these automatic tools, we can enhance the efficiency of quantum algorithms in solving optimization problems.

利用自动化工具探索不同的编码，最终优化问题的公式。这种方法允许对各种编码策略进行全面的检查和比较，有助于识别给定优化问题的最有效配置。此外，重复块的识别有助于制定新的优化问题，也构成了一个有价值的自动化工具。通过利用这些自动化工具，我们可以提高量子算法解决优化问题的效率。

Finding the optimal spin Hamiltonian for a given quantum computer hardware platform is an important problem in itself. As pointed out by Sawaya et al. (2022), optimal spin Hamiltonians are likely to vary across different hardware platforms, underscoring the need for a procedure capable of tailoring a problem to a specific platform. The hardware-agnostic approach made use of in this review represents a further step in that direction.

为给定的量子计算机硬件平台寻找最优自旋哈密顿量本身就是一个重要问题。正如Sawaya等人所指出的，(2022)，最佳自旋哈密顿量可能在不同的硬件平台上有所不同，这突出了对能够根据特定平台定制问题的程序的需求。在这篇综述中使用的硬件不可知方法代表着朝着这个方向迈出的又一步。

Author contributions

作者贡献

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

所有列出的作者都对这部作品做出了实质性的、直接的和智力上的贡献，并批准出版。

Funding

基金

Work was supported by the Austrian Science Fund (FWF) through a START grant under Project No. Y1067-N27, the SFB BeyondC Project No. F7108-N38, QuantERA II Programme under Grant Agreement No. 101017733, the Federal Ministry for Economic Affairs and Climate Action through project QuaST, and the Federal Ministry of Education and Research on the basis of a decision by the German Bundestag.

根据德国联邦议院的决定，奥地利科学基金会（FWF）通过项目编号Y1067-N27下的START拨款、SFB BeyondC项目编号F7108-N38、赠款协议编号101017733下的QuantERA II计划、联邦经济事务和气候行动部通过项目QuaST以及联邦教育和研究部支持了这项工作。

Acknowledgments

鸣谢

The authors thank Dr. Kaonan Micadei for fruitful discussions.

作者感谢Kaonan Micadei博士富有成果的讨论。

Conflict of interest

利益冲突

Authors FD, MT, CE, and WL were employed by Parity Quantum Computing Germany GmbH. Authors JU, BM, and WL were employed by Parity Quantum Computing GmbH.

作者FD、MT、CE和WL受雇于Parity Quantum Computing Germany GmbH。作者JU、BM和WL受聘于Parity Quantum Computing GmbH。

The author BM declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

提交人BM宣称，在提交时，他们是《前沿》杂志的编辑委员会成员。这对同行审议进程和最终决定没有影响。

Publisher's note

发布者备注

References

工具书类

Amaro, D., Rosenkranz, M., Fitzpatrick, N., Hirano, K., and Fiorentini, M. (2022). A case study of variational quantum algorithms for a job shop scheduling problem. *EPJ Quantum Technol.* 9, 5. doi:10.1140/epjqt/s40507-022-00123-4

Amaro，D.、Rosenkranz，M.、Fitzpatrick，N.、Hirano，K.和Fiorentini，M.（2022）。一个车间调度问题的变分量子算法的案例研究。*EPJ量子技术*。9, 5. doi:10.1140/epjqt/s40507-022-00123-4

Au-Yeung, R., Chancellor, N., and Halffmann, P. (2023). NP-Hard but no longer hard to solve? Using quantum computing to tackle optimization problems. *Front. Quantum Sci. Technol.* 2, 1128576. doi:10.3389/frqst.2023.1128576

Au Yeung，R.，Chancellor，N.和Halffmann，P.（2023）。NP难，但不再难解决？使用量子计算来解决优化问题。*正面量子科学*。*Technol*。2, 1128576. doi:10.3389/frqst.2023.1128576

Bakó, B., Glos, O., Salehi, A., and Zimborás, Z. (2022). *Near-optimal circuit design for variational quantum optimization*. arXiv:2209.03386. doi:10.48550/arXiv.2209.03386

Bakó，B.、Glos，O.、Salehi，A.和Zimborás，Z.（2022）。*变分量子优化的近似最优电路设计*。arXiv：2209.03386。doi:10.44850/arXiv.2209.03386

Bärtschi, A., and Eidenbenz, S. (2020). "Grover mixers for QAOA: shifting complexity from mixer design to state preparation," in *2020 IEEE international conference on quantum computing and engineering (QCE)*, 72–82. doi:10.1109/QCE49297.2020.00020

Bärtschi，A.和Eidenbenz，S.（2020）。用于QAOA的Grover混频器：将复杂性从混频器设计转移到状态准备"，*2020年IEEE量子计算与工程国际会议(QCE)*，72–82。doi:10.10109/QCE4929.7020.00020

Berge, C. (1987). *Hypergraphs*. Amersterdam: Elsevier.

Berge，C.（1987）。*Hypergraph*。艾姆斯特丹：爱思唯尔。

Berwald, J., Chancellor, N., and Dridi, R. (2023). Understanding domain-wall encoding theoretically and experimentally. *Phil. Trans. R. Soc. A* 381, 20210410. doi:10.1098/rsta.2021.0410

Berwald，J.，Chancellor，N.和Dridi，R.（2023）。从理论和实验上理解畴壁编码。*菲尔*，*Trans*。*R.Soc.A* 38120210410。doi:10.1098/rsta.20210.410

Boulebnane, S., and Montanaro, A. (2022). *Solving boolean satisfiability problems with the quantum approximate optimization algorithm*. arXiv preprint arXiv:2208.06909. doi:10.48550/arXiv.2208.06909

Boulebnane，S.和Montanaro，A.（2022）。*用量子近似优化算法求解布尔可满足性问题*。arXiv预印本arXiv：2208.06909。doi:10.44850/arXiv.2208.06909

Cao, Y., Romero, J., Olson, J. P., Degroote, M., Johnson, P. D., Kieferová, M., et al. (2019). Quantum chemistry in the age of quantum computing. *Chem. Rev.* 119, 10856–10915. doi:10.1021/acs.chemrev.8b00803

Cao，Y.、Romero，J.、Olson，J.P.、Degroune，M.、Johnson，P.D.、Kieferová，M.等人（2019）。量子计算时代的量子化学。*化学*。*修订版* 11910856-10915。doi:10.1021/acs.chemrev.8b00803

Carugno, C., Ferrari Dacrema, M., and Cremonesi, P. (2022). Evaluating the job shop scheduling problem on a D-wave quantum annealer. *Sci. Rep.* 12, 6539. doi:10.1038/ s41598-022-10169-0

Carugno，C.、Ferrari Dacrema，M.和Cremonesi，P.（2022）。在D波量子退火器上评估作业车间调度问题。*Sci*。*众议员* 12，6539。doi:10.1038/s41598-022-10169-0

Chancellor, N. (2019). Domain wall encoding of discrete variables for quantum annealing and QAOA. *Quantum Sci. Technol.* 4, 045004. doi:10.1088/2058-9565/ab33c2

财政大臣N（2019）。用于量子退火和QAOA的离散变量的域壁编码。*量子科学*。*Technol*。4, 045004. doi:10.1088/2058-9565/ab33c2

Chancellor, N., Zohren, S., and Warburton, P. A. (2017). Circuit design for multi-body interactions in superconducting quantum annealing systems with applications to a scalable architecture. *npj Quantum Inf.* 3, 21. doi:10.1038/s41534-017-0022-6

Chancellor，N.、Zohren，S.和Warburton，P.A.（2017）。超导量子退火系统中多体相互作用的电路设计及其在可扩展架构中的应用。*npj量子信息*3，21。doi:10.1038/s41534-017-0022-6

Chen, J., Stollenwerk, T., and Chancellor, N. (2021). Performance of domain-wall encoding for quantum annealing. *IEEE Trans. Quantum Eng.* 2, 1–14. doi:10.1109/TQE. 2021.3094280

Chen，J.，Stollenwerk，T.和Chancellor，N.（2021）。量子退火的畴壁编码性能。*IEEE Trans*。*量子工程*2，1-14。doi:10.109/TQE。2021.3094280

Choi, V. (2010). *Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3SAT problems*. arXiv:1004.2226. doi:10.48550/ARXIV. 1004.2226

Choi，V.（2010）。*NP完全最大权独立集的绝热量子算法，精确覆盖和3SAT问题*。arXiv：1004.2226。doi:10.44850/ARXIV。1004.2226

Devroye, L., Epstein, P., and Sack, J.-R. (1993). On generating random intervals and hyperrectangles. *J. Comput. Graph. Stat.* 2, 291–307. doi:10.2307/1390647

Devroye，L.、Epstein，P.和Sack，J.-R.（1993）。关于生成随机区间和超矩形。*J.计算*。*图表Stat*.2，291–307。doi:10.307/1390647

Di Matteo, O., McCoy, A., Gysbers, P., Miyagi, T., Woloshyn, R. M., and Navrátil, P. (2021). Improving Hamiltonian encodings with the Gray code. *Phys. Rev. A* 103, 042405. doi:10.1103/PhysRevA.103.042405

Di Matteo，O.、McCoy，A.、Gysbers，P.、Miyagi，T.、Woloshyn，R.M.和Navrátil，P.（2021）。用格雷码改进哈密顿编码。*Phys*。*修订版A* 103、042405。 doi:10.10103/PhysRevA.103.042405

Dlaska, C., Ender, K., Mbeng, G. B., Kruckenhauser, A., Lechner, W., and van Bijnen, R. (2022). Quantum optimization via four-body rydberg gates. *Phys. Rev. Lett.* 128, 120503. doi:

Dlaska，C.、Ender，K.、Mbeng，G.B.、Kruckenhauser，A.、Lechner，W.和van Bijnen，R.（2022）。通过四体里德堡门进行量子优化。*Phys*。*Rev.Lett*。128, 120503. doi：

Dorigo, M., and Di Caro, G. (1999). "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (IEEE), 2, 1470–1477. doi:10.1109/CEC.1999.782657

Dorigo，M.和Di Caro，G.（1999）。 蚁群优化：一种新的元启发式方法"，载于*1999年进化计算大会论文集-CEC99（目录号99TH8406）*（IEEE），21470–1477。doi:10.10109/CEC.1999.782657

Drieb-Schön, M., Javanmard, Y., Ender, K., and Lechner, W. (2023). Parity quantum optimization: encoding constraints. *Quantum* 7, 951. doi:10.22331/q-2023-03-17-951

Drieb Schön，M.、Javanmard，Y.、Ender，K.和Lechner，W.（2023）。奇偶量子优化：编码约束。*量子*7951。doi:10.22331/q-2023-03-17-951

Elliott, M., Golub, B., and Jackson, M. O. (2014).

Financial networks and contagion.

Elliott，M.、Golub，B.和Jackson，M.O.（2014）。金融网络和传染病。*Am. Econ. Rev.* 104, 3115–3153. doi:10.1257/aer.104.10.3115
Am.Econ。修订版1043115–3153。doi:10.1257/aer.104.10.13115

Ender, K., Messinger, A., Fellner, M., Dlaska, C., and Lechner, W. (2022). Modular parity quantum approximate optimization. *PRX Quantum* 3, 030304. doi:10.1103/ prxquantum.3.030304

Ender，K.、Messinger，A.、Fellner，M.、Dlaska，C.和 Lechner，W.（2022）。模块奇偶量子近似优化。PRX量子3，030304。doi:10.10103/prxquantum.030304

Ender, K., ter Hoeven, R., Niehoff, B. E., Drieb-Schön, M., and Lechner, W. (2023). Parity quantum optimization: compiler. *Quantum* 7, 950. doi:10.22331/q-2023-03-17-950

Ender，K.、ter Hoeven，R.、Niehoff，B.E.、Drieb Schön，M.和Lechner，W.（2023）。奇偶校验量子优化：编译器。量子7950。doi:10.22331/q-2023-03-17-950

Farhi, E., Goldstone, J., and Gutmann, S. (2014). *A quantum approximate optimization algorithm*, 4028. arXiv:1411. doi:10.48550/ARXIV.1411.4028

Farhi，E.、Goldstone，J.和Gutmann，S.（2014）。量子近似优化算法，4028。arXiv:1411。doi:10.44850/ARXIV.1411.4028

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). *Quantum computation by adiabatic evolution*. arXiv:quant-ph/0001106. doi:10.48550/ARXIV.QUANT-PH/0001106

Farhi，E.、Goldstone，J.、Gutmann，S.和Sipser，M.（2000）。绝热演化的量子计算。arXiv：定量ph/0001106。doi:10.44850/ARXIV。数量-PH/0 0 0 1 1 0 6

Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., et al. (2019). A hybrid solution method for the capacitated Vehicle routing problem using a quantum annealer. *Front. ICT* 6. doi:10.3389/fict.2019.00013

Feld，S.、Roch，C.、Gabor，T.、Seidel，C.、Neukart，F.、Galter，I.等人（2019）。一种使用量子退火器的电容车辆路径问题的混合求解方法。正面ICT6。doi:10.3389/fict.2019.00013

Fellner, M., Ender, K., ter Hoeven, R., and Lechner, W. (2023). Parity quantum optimization: benchmarks. *Quantum* 7, 952. doi:10.22331/q-2023-03-17-952

Fellner，M.、Ender，K.、ter Hoeven，R.和Lechner，W.（2023）。奇偶量子优化：基准。量子7952。doi:10.22331/q-2023-03-17-952

Fellner, M., Messinger, A., Ender, K., and Lechner, W. (2022). Universal parity quantum computing. *Phys. Rev. Lett.* 129, 180503. doi:10.1103/PhysRevLett.129.180503

Fellner，M.、Messinger，A.、Ender，K.和Lechner，W.（2022）。通用奇偶量子计算。Phys。Rev.Lett。129，180503。doi:10.10103/PhysRevLett.129180503

Fuchs, F. G., Kolden, H. O., Aase, N. H., and Sartor, G. (2021). Efficient encoding of the weighted MAX *k*-CUT on a quantum computer using QAOA. *SN Comput. Sci.* 2, 89. doi:10.1007/s42979-020-00437-z

Fuchs，F.G.、Kolden，H.O.、Aase，N.H.和Sartor，G.（2021）。在量子计算机上使用QAOA对加权的最大*k-CUT*进行有效编码。SN计算。Sci。2, 89. doi:10.1007/s42979-020-00437-z

Fuchs, F. G., Lye, K. O., Nilsen, H. M., Stasik, A. J., and Sartor, G. (2022). Constraint preserving mixers for the quantum approximate optimization algorithm. *Algorithms* 15, 202. doi:10.3390/a15060202

Fuchs，F.G.、Lye，K.O.、Nilsen，H.M.、Stasik，A.J.和Sartor，G.（2022）。量子近似优化算法的约束保持混合器。算法15，202.doi:10.3390/a15060202

Georgescu, I. M., Ashhab, S., and Nori, F. (2014). Quantum simulation. *Rev. Mod. Phys.* 86, 153–185. doi:10.1103/RevModPhys.86.153

Georgescu，I.M.、Ashhab，S.和Nori，F.（2014）。量子模拟。修订版Mod。Phys。86, 153–185. doi:10.1103/RevModPhys.86.153

Glaser, N. J., Roy, F., and Filipp, S. (2023). Controlled-controlled-phase gates for superconducting qubits mediated by a shared tunable coupler. *Phys. Rev. Appl.* 19, 044001. doi:10.1103/PhysRevApplied.19.044001

Glaser，N.J.、Roy，F.和Filipp，S.（2023）。由共享可调谐耦合器介导的超导量子位受控控制相位门。Phys。修订申请。19, 044001. doi:10.10103/PhysRevApplied.19.044001

Glos, A., Krawiec, A., and Zimborás, Z. (2022). Space-efficient binary optimization for variational quantum computing. *npj Quantum Inf.* 8, 39. doi:10.1038/s41534-022-00546-y

Glos，A.、Krawiec，A.和Zimborás，Z.（2022）。变分量子计算的空间有效二进制优化。npj量子信息8，39。doi:10.1038/s41534-022-00546-y

Guerreschi, G. G., and Matsuura, A. Y. (2019). Qaoa for max-cut requires hundreds of qubits for quantum speed-up. *Sci. Rep.* 9, 6903. doi:10.1038/s41598-019-43176-9

Guerreschi，G.G.和Matsuura，A.Y.（2019）。用于最大切割的Qaoa需要数百个量子位用于量子加速。Sci。众议员9，6903。doi:10.1038/s41598-019-43176-9

Hadfield, S. (2021). On the representation of Boolean and real functions as Hamiltonians for quantum computing. *ACM Trans. Quant. Comput.* 2, 1–21. doi:10. 1145/3478519

Hadfield，S.（2021）。关于量子计算中布尔函数和实函数作为哈密顿量的表示。ACM Trans。数量。Comput。2, 1–21. doi:10.1145/3478519

Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E. G., Venturelli, D., and Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 12, 34. doi:10.3390/a12020034

Hadfield，S.、Wang，Z.、O'Gorman，B.、Rieffel，E.G.、Venturelli，D.和Biswas，R.（2019）。从量子近似优化算法到量子交替算子变换。算法12，34。doi:10.3390/a12020034

Hadfield, S., Wang, Z., Rieffel, E. G., O'Gorman, B., Venturelli, D., and Biswas, R. (2017). *Quantum approximate optimization with hard and soft constraints. PMES' 17*. New York, NY, USA: Association for Computing Machinery. doi:10.1145/3149526. 3149530

Hadfield，S.、Wang，Z.、Rieffel，E.G.、O'Gorman，B.、Venturelli，D.和Biswas，R.（2017）。具有硬约束和软约束的量子近似优化。设备17。美国纽约：计算机协会。doi:10.145/3149526。3149530

Hen, I., and Sarandy, M. S. (2016). Driver Hamiltonians for constrained optimization in quantum annealing. *Phys. Rev. A* 93, 062312. doi:10.1103/ PhysRevA.93.062312

Hen，I.和Sarandy，M.S.（2016）。量子退火中约束优化的驱动哈密顿量。Phys。修订版A 9306312。doi:10.10103/PhysRevA.93.062132

Hen, I., and Spedalieri, F. M. (2016). Quantum annealing for constrained optimization. *Phys. Rev. Appl.*

5, 034007. doi:10.1103/PhysRevApplied.5.034007

Hen，I.和Spedalieri，F.M.（2016）。用于约束优化的量子退火。*Phys。修订申请*。5, 034007. doi:10.1003/PhysRevApplied.5.034007

Ikeda, K., Nakamura, Y., and Humble, T. S. (2019). Application of quantum annealing to nurse scheduling problem. *Sci. Rep.* 9, 12837. doi:10.1038/s41598-019-49172-3

Ikeda，K.、Nakamura，Y.和Humble，T.S.（2019）。量子退火在护理调度问题中的应用。*Sci。众议员* 9，12837。doi:10.1038/s41598-019-49172-3

Izawa, S., Kitai, K., Tanaka, S., Tamura, R., and Tsuda, K. (2022). Continuous black-box optimization with an ising machine and random subspace coding. *Phys. Rev. Res.* 4, 023062. doi:10.1103/PhysRevResearch.4.023062

Izawa，S.、Kitai，K.、Tanaka，S.、Tamura，R.和Tsuda，K.（2022）。连续黑盒优化与ising机器和随机子空间编码。*Phys。Rev.Res.*4023062。doi:10.10103/PhysRevResearch.4.023062

King, J., Yarkoni, S., Raymond, J., Ozfidan, I., King, A. D., Nevisi, M. M., et al. (2019). Quantum annealing amid local ruggedness and global frustration. *J. Phys. Soc. Jpn.* 88, 061007. doi:10.7566/JPSJ.88.061007

King，J.、Yarkoni，S.、Raymond，J.、Ozfidan，I.、King，A.D.、Nevisi，M.等人（2019）。局部崎岖和全球挫折中的量子退火。*J.Phys。Soc.Jpn*。88，061007. doi:10.7566/JPSJ.88.061007

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., et al. (2014). The unconstrained binary quadratic programming problem: a survey. *J. Comb. Optim.* 28, 58–81. doi:10.1007/s10878-014-9734-0

Kochenberger，G.、Hao，J.-K、Glover，F.、Lewis，M.、Lü，Z.、Wang，H.等人（2014）。无约束二次规划问题综述。*J.Comb。Optim。*28, 58–81. doi:10.1007/s10878-014-9734-0

Kurowski, K., Weglarz, J., Subocz, M., Różycki, R., and Waligóra, G. (2020). "Hybrid quantum annealing heuristic method for solving job shop scheduling problem," in *Computational science – ICCS 2020*. Editors V. V. Krzhizhanovskaya, G. Závodszky,

Kurowski，K.、Weglarz，J.、Subocz，M.、Różycki，R.和 Waligóra，G.（2020）。 求解作业车间调度问题的混合量子退火启发式方法"，载于《*计算科学-ICCS 2020*》。编辑V.V.Krzhizhanovskaya，G.Závodsky，

M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, et al. (Springer International Publishing), 502–515.

M.H.Lees、J.J.Dongarra、P.M.A.Sloot、S.Brissos等人（施普林格国际出版社），502–515。

Lai, C.-Y., Kuo, K.-Y., and Liao, B.-J. (2022). *Syndrome decoding by quantum approximate optimization*. arXiv:2207.05942. doi:10.48550/ARXIV.2207.05942

赖、郭、廖（2022）。*量子近似优化的综合症解码*。arXiv：2207.05942。doi:10.44850/ARXIV.2207.05942

Lanthaler, M., and Lechner, W. (2021). Minimal constraints in the parity formulation of optimization problems. *New J. Phys.* 23, 083039. doi:10.1088/1367-2630/ac1897

Lanthaler，M.和Lechner，W.（2021）。优化问题奇偶性公式中的最小约束。*New J.Phys*。23, 083039. doi:10.1088/1367-2630/ac1897

Lechner, W., Hauke, P., and Zoller, P. (2015). A quantum annealing architecture with all-to-all connectivity from local interactions. *Sci. Adv.* 1, e1500838. doi:10.1126/sciadv. 1500838

Lechner，W.、Hauke，P.和Zoller，P.（2015）。一种量子退火架构，具有来自局部相互作用的所有到所有连接。*Sci。Adv.*1，e1500838。doi:10.126/ciadv。1500838

Lechner, W. (2020). Quantum approximate optimization with parallelizable gates.

Lechner，W.（2020）。具有可并行门的量子近似优化。

IEEE Trans. Quantum Eng. 1, 1–6. doi:10.1109/TQE.2020.3034798

*IEEE Trans。量子工程*1，1–6。doi:10.109/TQE.2020.3034798

Lenstra, J. K., and Rinnooy Kan, A. H. G. (1979). Computational complexity of discrete optimization problems. *AODM* 4, 121–140. doi:10.1016/S0167-5060(08) 70821-5

Lenstra，J.K.和Rinnooy Kan，A.H.G.（1979）。离散优化问题的计算复杂性。*AODM*4，121–140。doi:10.1016/S0167-5060（08）70821-5

Lu, Y., Zhang, S., Zhang, K., Chen, W., Shen, Y., Zhang, J., et al. (2019). Global entangling gates on arbitrary ion qubits. *Nature* 572, 363–367. doi:10.1038/s41586-019-1428-4

鲁、张、S、张、K、陈、沈、张等（2019）。任意离子量子位上的全局纠缠门。《*自然*》572363-367。doi:10.1038/s41586-019-1428-4

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2, 5. doi:10. 3389/fphy.2014.00005

Lucas，A.（2014）。许多NP问题的Ising公式。*正面Phys*。2, 5. doi:10。3389/fphy.2014.00005

Mandra, S., Zhu, Z., Wang, W., Perdomo-Ortiz, A., and Katzgraber, H. G. (2016). Strengths and weaknesses of weak-strong cluster problems: a detailed overview of state-of-the-art classical heuristics versus quantum approaches. *Phys. Rev. A* 94, 022337. doi:10.1103/physreva.94.022337

Mandra，S.、Zhu，Z.、Wang，W.、Perdomo Ortiz，A.和Katzgraber，H.G.（2016）。弱-强簇问题的优势和劣势：最先进的经典启发式与量子方法的详细概述。*Phys。修订版A*94022337。doi:10.10103/physrev.94.022337

Mazyavkina, N., Sviridov, S., Ivanov, S., and Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: a survey. *Comput. Oper. Res.* 134, 105400. doi:10.1016/j.cor.2021.105400

Mazyavkina，N.、Sviridov，S.、Ivanov，S.和Burnaev，E.（2021）。组合优化的强化学习：一项调查。*Comput。Oper。第*134105400*号决议*。doi:10.1016/j.cor.2021.105400

McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C., and Yuan, X. (2020). Quantum computational chemistry. *Rev. Mod. Phys.* 92, 015003. doi:10.1103/RevModPhys.92.015003

McArdle，S.、Endo，S.、Aspuru Guzik，A.、Benjamin，S.C.和Yuan，X.（2020）。量子计算化学。*修订版Mod。Phys。*92，015003. doi:10.10103/RevModPhys.92.015003

Melnikov, B. (2005). “Discrete optimization problems-some new heuristic approaches,” in *Eighth international conference on high-performance computing in asia-pacific region (HPCASIA'05)* (IEEE), 73–82. doi:10.1109/HPCASIA.2005.34

Melnikov，B.（2005）。离散优化问题——一些新的启发式方法”，*第八届亚太地区高性能计算国际会议（HPCASIA'05）*（IEEE），73–82。doi:10.1009/HPCASIA.2005.34

Menke, T., Banner, W. P., Bergamaschi, T. R., Di Paolo, A., Vepsäläinen, A., Weber, S. J., et al. (2022). Demonstration of tunable three-body interactions between superconducting qubits. *Phys. Rev. Lett.* 129, 220501. doi:10.1103/PhysRevLett.129. 220501

Menke，T.、Banner，W.P.、Bergamaschi，T.R.、Di Paolo，A.、Vepsäläinen，A.、Weber，S.J.等人（2022）。超导量子位之间可调谐三体相互作用的演示。*Phys。Rev.Lett。*129，220501. doi:10.103/PhysRevLett.129。220501

Menke, T., Häse, F., Gustavsson, S., Kerman, A. J., Oliver, W. D., and Aspuru-Guzik,

Menke，T.、Häse，F.、Gustavsson，S.、Kerman，A.J.、Oliver，W.D.和Aspuru Guzik，

A. (2021). Automated design of superconducting circuits and its application to 4-local couplers. *npj Quantum Inf.* 7, 49. doi:10.1038/s41534-021-00382-6

A.（2021）。超导电路的自动化设计及其在四局域耦合器中的应用。*npj量子信息*7，49。doi:10.1038/s41534-021-00382-6

Messinger, A., Fellner, M., and Lechner, W. (2023). *Constant depth code deformations in the parity architecture*. arXiv:2303.08602. doi:10.48550/arXiv.2303.08602

Messinger，A.、Fellner，M.和Lechner，W.（2023）。*奇偶校验体系结构中的恒定深度代码变形*。arXiv：2303.08602。doi:10.44850/arXiv.2303.08602

Mohseni, N., McMahon, P. L., and Byrnes, T. (2022). Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* 4, 363–379. doi:10. 1038/s42254-022-00440-8

Mohseni，N.、McMahon，P.L.和Byrnes，T.（2022）。将机器作为组合优化问题的硬件求解器。*自然科学版*。4，363–379. doi:10。1038/s42254-022-00440-8

Montanez-Barrera, A., Maldonado-Romo, A., Willsch, D., and Michielsen, K. (2022). *Unbalanced penalization: a new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms*. arXiv:2211.13914. doi:10.48550/arXiv.2211.13914

Montanez-Barrera，A.、Maldonado Romo，A.、Willsch，D.和Michielsen，K.（2022）。*不平衡惩罚：为量子优化算法编码组合问题不等式约束的一种新方法*。arXiv：2211.13914。doi:10.44850/arXiv.2211.13914

Orús, R., Mugel, S., and Lizaso, E. (2019). Forecasting financial crashes with quantum computing. *Phys. Rev. A* 99, 060301. doi:10.1103/PhysRevA.99.060301

Orús，R.、Mugel，s.和Lizaso，E.（2019）。用量子计算预测金融崩溃。*Phys。修订版A* 99060301。doi:10.1003/PhysRevA.99.060301

Pastawski, F., and Preskill, J. (2016). Error correction for encoded quantum annealing.

Pastawski，F.和Preskill，J.（2016）。编码量子退火的纠错。

Phys. Rev. A 93, 052325. doi:10.1103/PhysRevA.93.052325

Phys。修订版A 93052325。doi:10.1003/PhysRevA.93.052325

Pelegrí, G., Daley, A. J., and Pritchard, J. D. (2022). High-fidelity multiqubit Rydberg gates via two-photon adiabatic rapid passage. *Quantum Sci. Technol.* 7, 045020. doi:10. 1088/2058-9565/ac823a

Pelegrí，G.、Daley，A.J.和Pritchard，J.D.（2022）。通过双光子绝热快速通道实现高保真多量子比特里德堡门。*量子科学*。*Technol*。7，045020. doi:10。1088/2058-9565/ac823a

Plewa, J., Sieńko, J., and Rycerz, K. (2021). Variational algorithms for workflow scheduling problem in gate-based quantum devices. *Comput. Inf.* 40, 897–929. doi:10. 31577/cai_2021_4_897

Plewa，J.、Sieńko，J.和Rycerz，K.（2021）。基于门的量子器件中工作流调度问题的变分算法。*Comput*。第40页，897–929页。doi:10。31577/蔡_2021_4_897

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum* 2, 79. doi:10.22331/q-2018-08-06-79

Preskill，J.（2018）。NISQ时代及以后的量子计算。*量子*2，79。doi:10.22331/q-2018-08-06-79

Rachkovskii, D. A., Slipchenko, S. V., Kussul, E. M., and Baidyk, T. N. (2005). Properties of numeric codes for the scheme of random subspaces rsc. *Cybern. Syst. Anal.* 41, 509–520. doi:10.1007/s10559-005-0086-8

Rachkovskii，D.A.、Slipchenko，S.V.、Kussul，E.M.和Baidyk，T.N.（2005）。随机子空间rsc格式的数字码的性质。*Cybern。系统。分析*。41，509–520. doi:10.1007/s10559-005-0086-8

Ramos-Calderer, S., Pérez-Salinas, A., García-Martín, D., Bravo-Prieto, C., Cortada, J., Planagumà, J., et al. (2021). Quantum unary approach to option pricing. *Phys. Rev. A* 103, 032414. doi:10.1103/PhysRevA.103.032414

Ramos Calderer，S.、Pérez Salinas，A.、GarcíA-Martín，D.、Bravo Prieto，C.、Cortada，J.、Planagumà，J.等人（2021）。期权定价的量子一元方法。*Phys。修订版A* 103，032414。doi:10.10103/PhysRevA.103.032414

Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., and de Prado, M. L. (2015). "Solving the optimal trading trajectory problem using a quantum annealer," in *Proceedings of the 8th workshop on high performance computational finance* (New York, NY, USA: Association for Computing Machinery). doi:10.1145/2830556.2830563

Rosenberg，G.、Haghnegahdar，P.、Goddard，P.、Carr，P.、Wu，K.和de Prado，M.L.（2015）。 使用量子退火器解决最佳交易轨迹问题"，载于《 第八届高性能计算金融研讨会论文集》（美国纽约：计算机械协会）。doi:10.145/2830556.2830563

Sawaya, N. P. D., Menke, T., Kyaw, T. H., Johri, S., Aspuru-Guzik, A., and Guerreschi, G. G. (2020). Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians. *npj Quantum Inf.* 6, 49. doi:

Sawaya，N.P.D.、Menke，T.、Kyaw，T.H.、Johri，S.、AspuruGuzik，A.和Guerreschi，G.（2020）。光子、振动和自旋s哈密顿的d级系统的资源高效数字量子模拟。*npj量子信息*6，49。doi:

Sawaya, N. P. D., Schmitz, A. T., and Hadfield, S. (2022). *Encoding trade-offs and design toolkits in quantum algorithms for discrete optimization: coloring, routing, scheduling, and other problems*, 14432. arXiv:2203. doi:10.48550/arXiv.2203.14432

Sawaya，N.P.D.、Schmitz，A.T.和Hadfield，S.（2022）。*离散优化量子算法中的编码权衡和设计工具包：着色、路由、调度和其他问题*，14432。arXiv:2203。doi:10.44850/arXiv.2203.14432

Schöndorf, M., and Wilhelm, F. K. (2019). Nonpairwise interactions induced by virtual transitions in four coupled artificial atoms. *Phys. Rev. Appl.* 12, 064026. doi:10. 1103/PhysRevApplied.12.064026

Schöndorf，M.和Wilhelm，F.K.（2019）。由四个耦合的人造原子中的虚跃迁引起的非空气相互作用。*Phys*。*修订申请*。12, 064026. doi:10。1103/PhysRevApplied.12.064026

Stein, J., Chamanian, F., Zorn, M., Nüßlein, J., Zielinski, S., Kölle, M., et al. (2023). *Evidence that PUBO outperforms QUBO when solving continuous optimization problems with the QAOA*. arXiv:2305.03390. doi:10.48550/arXiv.2305.03390

Stein，J.、Chamanian，F.、Zorn，M.、Nüßlein，J.、Zielinski，S.、Kölle，M.等人（2023）。证明 *PUBO在使用QAOA解决连续优化问题时优于QUBO的证据*。arXiv：2305.03390。doi:10.44850/arXiv.2305.03390

Tamura, K., Shirai, T., Katsura, H., Tanaka, S., and Togawa, N. (2021). Performance comparison of typical binary-integer encodings in an ising machine. *IEEE Access* 9, 81032–81039. doi:10.1109/ACCESS.2021.3081685

Tamura，K.、Shirai，T.、Katsura，H.、Tanaka，S.和 Togawa，N.（2021）。ising机中典型二进制整数编码的性能比较。*IEEE Access* 9，81032–81039。doi:10.10109/ACCESS.2021.3081685

Tseitin, G. S. (1983). "On the complexity of derivation in propositional calculus," in

Tseitin，G.S.（1983）。 关于命题演算中推导的复杂性"

Automation of reasoning (Springer), 466–483. doi:10.1007/978-3-642-81955-1_28

推理自动化（施普林格），466–483。doi:10.1007/978-3-642-81955-1_28

Unger, J., Messinger, A., Niehoff, B. E., Fellner, M., and Lechner, W. (2022). *Low-depth circuit implementation of parity constraints for quantum optimization*, 11287. arXiv:2211. doi:10.48550/ARXIV.2211.11287

Unger，J.、Messinger，A.、Niehoff，B.E.、Fellner，M.和Lechner，W.（2022）。*量子优化奇偶校验约束的低深度电路实现*，11287。arXiv:2211。doi:10.44850/ARXIV.2211.11287

Venturelli, D., Marchand, D. J. J., and Rojo, G. (2015). Quantum annealing implementation of job-shop scheduling. arXiv:1506.08479. doi:10.48550/ARXIV. 1506.08479

Venturelli，D.、Marchand，D.J.和Rojo，G.（2015）。作业车间调度的量子退火实现。arXiv：1506.08479。doi:10.44850/ARXIV。1506.08479

Wang, Z., Rubin, N. C., Dominy, J. M., and Rieffel, E. G. (2020). XY mixers: analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* 101, 012320. doi:10.1103/PhysRevA.101.012320

王，Z.，鲁宾，N.C.，多明尼，J.M.和里菲尔，E.G.（2020）。XY混合器：量子交替算子模拟的分析和数值结果。*Phys*。*修订版A* 101，012320。doi:10.10103/PhysRevA.101.012320

Wilkinson, S. A., and Hartmann, M. J. (2020). Superconducting quantum many-body circuits for quantum simulation and computing. *Appl. Phys. Lett.* 116, 230501. doi:10. 1063/5.0008202

Wilkinson，S.A.和Hartmann，M.J.（2020）。用于量子模拟和计算的超导量子多体电路。*Appl*。*Phys*。*Lett*。116, 230501. doi:10。1063/5.0008202

Zhu, Y., Zhang, Z., Sundar, B., Green, A. M., Alderete, C. H., Nguyen, N. H., et al. (2023). Multi-round QAOA and advanced mixers on a trapped-ion quantum computer. *Quantum Sci. Technol.* 8, 015007. doi:10.1088/2058-9565/ac91ef

Zhu，Y.，Zhang，Z.，Sundar，B.，Green，A.M.，Alderete，C.H.，Nguyen，N.H.等人（2023）。捕获离子量子计算机上的多轮QAOA和高级混合器。*量子科学*。*Technol*。8, 015007. doi:10.1088/2058-9565/ac91ef