用于约束问题的多目标变分量子优化：应用于现金管理

Pablo Díez-Valle[1], Jorge Luis-Hita[2], Senaida Hernández-Santana[2], Fernando Martínez-García[1], Álvaro Díaz-Fernández[2], Eva Andrés[2], JuanJosé García-Ripoll[1], EscolásticoSánchez-Martínez[2], 和 Diego Porras[1]

[1]基础物理研究所 IFF-CSIC，Calle Serrano 113b，马德里 28006，西班牙[2]BBVA Quantum，CalleAzul 4，28050 马德里，西班牙

arXiv:2302.04196v1 [quant-ph] 8 Feb 2023

组合优化问题在工业中无处不在。除了寻找成本最低的解之外，许多重要问题还涉及若干必须由解满足的约束。变分量子算法已成为在含噪中等规模量子阶段求解这些问题的有前景候选方案。然而，约束往往复杂到难以高效映射至量子硬件，甚至无法实现。一种常见的替代方案是将约束作为惩罚项纳入优化问题，但这种方法需要额外的超参数，而且由于存在局部最小值，无法确保满足约束。本文介绍一种使用变分量子算法求解具有复杂约束的组合优化问题的新方法。我们提出多目标变分约束优化器（MOVCO），通过遗传算法执行多目标优化，以经典方式更新变分参数。这种优化使算法能够逐步仅对约束内空间中的状态进行采样，同时优化这些状态的能量。我们在一个与金融密切相关的现实问题——现金管理问题上测试了这一方案。我们为该问题提出一种新颖的数学公式，并比较 MOVCO 与基于惩罚项的优化方法的表现优化。我们的实证结果表明，所得解的成本显著降低，尤其是避免了不满足任何强制约束的局部最小值。

1 引言

量子计算有望凭借解决复杂问题的能力，对科学和工业产生重大影响。一些潜在应用包括多体问题的量子模拟 [1, 2], 数值分析任务（如求解微分方程） [3, 4],机器学习 [5, 6, 7], 等。其中一个受到广泛研究的领域是组合优化（CO），其任务是在有限的元素集合中寻找最优解 [8]。在许多情况下，精确求解这类问题是众所周知的NP-hard问题 [9],，也就是说，无法通过经典计算在多项式时间内求解。一些典型例子包括旅行商问题 [10] 和最大割问题 [11, 12]。尽管量子计算机可能也无法在多项式时间内精确解决这类问题，但启发式量子算法仍有望在速度、所得解的质量，甚至资源节省方面，为这些问题带来更好的近似解。近期用于解决这类问题的主要量子范式之一是变分量子算法（VQA） [13]。这类算法采用量子-经典混合方法，近似求出编码问题的哈密顿量的最低能量态。量子与经典计算的结合，使得能够使用适用于含噪中等规模量子计算的浅层量子电路

Pablo Díez-Valle： pablo.diez@csic.es

（含噪声中等规模量子）时代 [14, 15]。

组合优化广泛存在于各类工业领域，例如金融 [16]，物流 [17]，人工智能 [18]，供应链 [19] 和药物发现 [20]。因此，即使所得解可能只是全局最优解的近似，更好的组合优化解仍具有重要的实际价值。现实中的组合优化问题通常不仅涉及最小化成本函数，还包含若干必须由可行解满足的等式和不等式硬约束。在量子优化领域，一种理想的方法是实现诸如量子交替算子拟设之类的受约束算法，并使用能够原生保持约束的量子电路 [21, 22]。在这方面已有一些研究，包括近期在文本摘要任务上的实验演示 [23]。然而，实际问题的现实建模中所含的约束往往过于复杂，难以高效映射到量子处理器上。在这种情况下，强制满足约束的最先进策略是通过引入惩罚项来转换原始成本函数，人为提高不可行解的能量 [24]。尽管这种方法易于实现，却无法保证算法收敛到满足所有约束的解，而且还存在一些缺点，例如需要调整依赖具体实例的惩罚超参数。尽管约束在实际应用场景中至关重要，探索变分量子算法中通用约束编码新方法的研究仍然很少。事实上，变分量子算法的性能研究通常采用经过充分研究的理论问题。这些问题虽然对于评估量子算法的潜力很有价值，却缺乏在工业领域的直接实际应用性。

在这项工作中，我们提出了多目标变分约束优化器（MOVCO），这是一种旨在提升变分量子算法收敛至满足一组限制条件的最优解的速度的方法，并在一个具有工业实际意义的问题上测试了其性能。MOVCO借助遗传算法，同时优化变分波函数向满足所有约束的解所构成子空间的投影，以及可行解的能量。据我们所知，这是首批探讨如何通过变分方法高效优化涉及大量一般性约束的现实问题的论文之一。就在我们准备本文期间，最近有一项研究发表，提出修改变分量子算法的目标函数以处理硬约束 [25]。具体而言，他们提出最小化约束内能量，而非经惩罚处理的成本函数。然而，尽管同样是少数讨论这一问题的论文之一，我们的多目标方法显然与之不同。

本文结构如下。首先，我们简要回顾理解这一新方法所需的关键概念，包括变分量子算法（第2.1节）、约束组合优化（第2.2节）以及非支配排序算法（第2.3节）。我们在第3节介绍 MOVCO，并阐述该方法的优势。随后，我们在一个名为现金管理问题的真实金融与物流问题上展示该方法的有效性。我们在第4.1节详细描述这一实际问题的具体要求，并在第4.2节提出一种以自旋为基础的新颖数学表述。在第5 节，我们展示实证结果，证明 MOVCO 在避免不可行局部最小值方面具有优势。最后，我们将总结本研究的重要性、未来研究方向以及 MOVCO 可能的改进之处。

2 背景

2.1 变分量子优化

变分量子算法（VQA）是一类量子-经典混合算法，因能够利用浅层量子电路解决复杂问题而受到广泛研究 [13]。这类算法的潜力源于强大的经典优化方法与量子电路高效探索整个希尔伯特空间的能力相结合。变分量子算法种类繁多，其特点会根据预期应用进行调整。尽管如此，它们仍有一些共同要素，并按照以下方案运行。

变分量子算法依赖量子计算机，通过依次应用一系列

量子操作 $U(\vec{\theta})$ 来构建拟设，并作用于输入态：

$$|\Psi(\vec{\theta})\rangle = U(\vec{\theta})\,|0\rangle^{\otimes N}\,, \tag{1}$$

其中 N 是量子比特的数量。这些操作取决于若干可调参数 $\vec{\theta}$。酉操作 $U(\vec{\theta})$ 可表示为若干个 L 酉算符的乘积，其中 L 是拟设的层数，它决定了量子电路的深度：

$$U(\vec{\theta}) = \prod_{l=1}^{L} U_l(\vec{\theta}_l). \tag{2}$$

然后，根据对问题进行编码的成本函数来调节拟设的参数。此调节由经典计算机执行，计算机将对拟设的测量结果作为输入，以找到最优参数，使 $\dashv|\Psi(\theta)\rangle$ 尽可能接近最佳解。

这类算法已在组合优化领域得到广泛研究 [8, 26, 27, 28, 29, 30, 31]。尽管目前尚无证据证明这些算法能够在优化问题中带来量子优势，但已有迹象表明它们具有潜力 [32, 33, 34, 35]。在这一领域得到研究的变分量子算法包括量子近似优化算法（QAOA）[36]，变分量子本征求解器（VQE）[37]，递归QAOA（RQAOA） [38]，滤波变分量子本征求解器（F-VQE） [39]，Layer VQE（L-VQE） [40],ADAPT-QAOA [41], 变分量子虚时间演化（VarQITE） [42]，等。本文将数值实验重点放在变分量子本征求解器上；其与具体问题无关的构造能够很好地适用于诸如第4节所介绍的现金管理问题这类黑箱优化问题。不过，本文提出的多目标新技术也可以轻松扩展到上述任何一种变分算法。

2.1.1 变分量子本征求解器

VQE最初被提出作为一种变分算法，用于求解由哈密顿量 H[37]，描述的化学分子的基态能量，但也已应用于量子优化 [29]领域。

在变分量子本征求解器中，通过最小化哈密顿量的期望值来训练拟设的参数$\langle\Psi(\vec{\theta})|\ \hat{}H\,|\Psi(\vec{\theta})\rangle$。

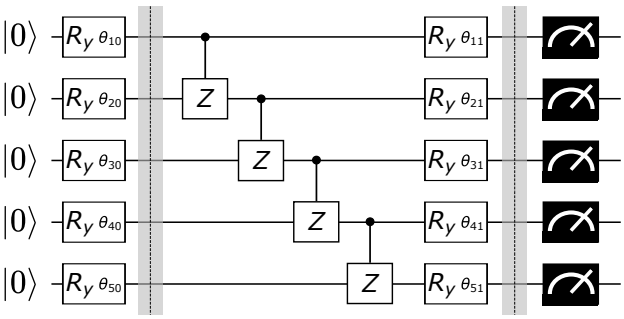


图1：适用于硬件的变分量子电路的一层（3），适用于五个量子比特。我们在最近邻之间应用参数化单量子比特 R_y 旋转和受控Z纠缠门。

尽管已有研究提出了一些针对特定问题或受问题启发的变分量子本征求解器拟设，用于解决量子化学问题 [43, 44, 45],在该算法中，变分试探态 (1) 是自由的，也就是说，它无需像量子近似优化算法那样由问题的哈密顿量决定。具体而言，一种典型的硬件高效选择是图1所示的量子门序列：

$$U(\vec{\theta}) = \prod_{l=1}^{L}\left[\prod_{n=1}^{N} R_y(\theta_{nl})U_{\text{ent}}\right]\prod_{n=1}^{N} R_y(\theta_{n0})\,, \tag{3}$$

ˆ其中$R_y\,(\theta_{nl}) \equiv \quad e^{i\theta_{nl}Y_n}$ 是围绕y轴的单量子比特旋转，$U_{\text{ent}} \equiv \prod_{n=1}^{N-1} e^{i\frac{\pi}{4}\,(\mathbb{I}-\hat{}Z_n)\,(\mathbb{I}-\hat{}Z_{n+1})}$ 是由线性量子处理器拓扑中每个量子比特与其最近邻量子比特之间的双量子比特受控Z门构成的纠缠门，ˆ ˆ并且 Y_n 和 Z_n 是作用于第n个量子比特的泡利Y算符和Z算符。在第5节所示的计算实验中，参数θ_{n0} 初始化在 $\sim\ \pi/4$ 附近，使得初始态接近完全叠加态，并加入小的随机扰动 $\theta_{nl} \sim 10^{-2}$。

硬件高效拟设的灵活性和最近邻连接使其适合在真实量子处理器上实现该算法。然而，这类拟设并非针对具体问题设计，可能会面临可训练性问题，例如贫瘠高原 [46]。

2.2 约束组合优化

与工业相关的组合优化问题通常涉及满足大量硬约束。这些

问题可以表述为最小化一个黑盒成本函数，

$$\min C(z) \text{ with } z \in \{-1,+1\}^N , \tag{4}$$

使得解 z 必须满足若干不等式约束和等式约束：

$$b_i(z) = 0 \ , \ g_j(z) \leq 0 \tag{5}$$

其中， N 是问题中的变量数量，而 $i = 1$、... ...、 I、 $j = 1$、.....、 J分别表示等式约束和不等式约束的数量。通过变量替换 $z_n \to Z_n$，可以将该成本函数ˆ转换为哈密顿量 C ，其中 Z_n 是作用于第 n 个量子比特的泡利 Z 矩阵。因此，该问题转化为寻找此哈密顿量的基态。对于变分算法而言，这等价于寻找使波函数更接近该基态的⃗参数 θ ：

$$\min_{\vec{\theta}} \left[\langle \Psi(\vec{\theta}) | \, \hat{C}(\hat{Z}) \, | \Psi(\vec{\theta}) \rangle \right] , \tag{6}$$

使得

$$\begin{aligned} \langle \Psi(\vec{\theta}) | \, \hat{B}_i(\hat{Z}) \, | \Psi(\vec{\theta}) \rangle = 0 \, , \\ \langle \Psi(\vec{\theta}) | \, \hat{G}_j(\hat{Z}) \, | \Psi(\vec{\theta}) \rangle \leq 0 \, , \end{aligned} \tag{7}$$

其中，约束 (5) 也通过先前的变换用算符 B_i、 G_j ˆ ˆ 表示。我们将满足 (7) 的状态所构成的子空间称为约束内子空间或可行子空间 S。在现实场景中，试探态 $|\Psi(\vec{\theta})\mathbf{i}$ 在一台具有 N 个量子比特的量子计算机上制备，因此我们无法获取波函数的精确概率振幅。随后，通过在泡利 z 基中同时对所有量子比特进行 K 次测量，估计平均能量 $\mathbf{h}\Psi(\vec{\theta})| \, \hat{C}(\, \hat{Z}) \, |\Psi(\vec{\theta})\mathbf{i}$。每次测量都会产生一个经典状态 $|\Psi_k\mathbf{i} = \{-1,+1\}^N$ ，其成本函数值 $C_k = \mathbf{h}\Psi_k| \, \hat{C} \, |\Psi_k\mathbf{i}$ 可以高效计算。因此，样本均值就是我们实际最小化的估计量，

$$\langle \Psi(\vec{\theta}) | \, \hat{C}(\hat{Z}) \, | \Psi(\vec{\theta}) \rangle \approx \min_{\vec{\theta}} \left[\frac{1}{K} \sum_{k=1}^{K} \hat{C}_k(\vec{\theta}) \right] , \tag{8}$$

其中， $k = 1, ..., K$ 表示测得的每一个 K 样本。因此，受约束变分量子优化旨在近似一个波函数，使其能够采样成本较低的经典态 C_k ，满足$\langle \Psi_k| \, \hat{} \, B_i(\, \hat{Z}) \, |\Psi_k\rangle = 0$且$\langle \Psi_k| \, \hat{} \, G_j(\, \hat{Z}) \, |\Psi_k\rangle \leq 0$。

2.3 非支配排序遗传算法

非支配排序遗传算法（NSGA-II）由 [47], 提出，是一种用于执行多目标优化的进化算法，已被证明适用于大规模优化问题 [48, 49]。NSGA-II 的最终目标是为多个成本函数找到一组最优解。这组解称为非支配解集，或帕累托前沿。如果且仅当不存在另一个不同的解 $x_i \neq x_0$ ，使得 x_i 在至少一个目标函数上优于解 x_0 ，同时在所有其他正在优化的成本函数上都不劣于 x_0 ，则称解 x_0 为非支配解。

NSGA-II 属于进化算法中的遗传算法，是基于受生物学启发的算子（如自然选择、交叉和变异）的元启发式优化技术 [50]。优化过程通过迭代演化种群（一组解），利用适者生存的概念和上述算子来获得更优的个体（更优的解）。具体而言，NSGA-II 包含以下步骤（详见附录D ）：

1.种群初始化：算法以一组随机采样的解（个体）进行初始化。第一次迭代之后，种群将由上一次迭代的父代种群和子代种群组成。种群规模是模型的一个超参数。

2.非支配排序：根据帕累托支配关系对种群中的个体进行排序，并将其划分为前沿。换言之，不受任何其他元素支配的个体会被分配到前沿 1，并从后续考量中排除。然后，将剩余种群中的非支配个体归入前沿 2，以此类推，直至处理完种群中的所有元素。

3. 创建父代种群：根据前沿分类创建新的种群，使属于第一前沿的个体晋级到下一次迭代。

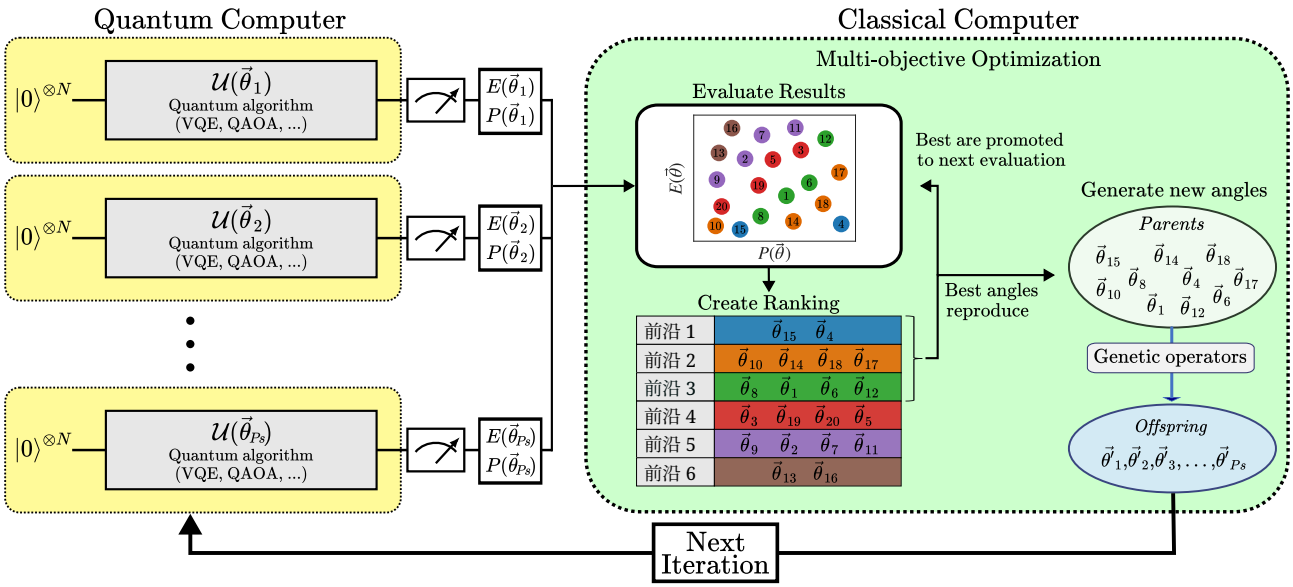


图 2：受约束多目标变分优化器（MOVCO）的示意图。该混合算法由在量子计算机上运行的变分试探态 $\mathcal{U}(\vec{\theta})$ 和经典遗传方法 NSGA-II 执行的多目标优化组成。首先，使用一组随机角度集合 $\vec{\theta}_i$ 对量子电路 (9) 和 (10)进行采样，以计算适应度函数，从约束满足和能量优化两个方面评估解的质量。根据帕累托前沿对结果进行排序，并将属于第一前沿的角度保留到下一次迭代中（父代种群）。通过对父代种群应用遗传算子（锦标赛选择、交叉和变异），生成一组新的角度（子代种群）。在下一次迭代中，仅使用子代参数对量子电路 $\mathcal{U}$ 进行采样，并将上一轮迭代的父代角度与新的子代结果结合起来执行多目标优化（彩色在线版）。

4. 拥挤距离排序：当前沿只被部分选取时，也就是说，并非需要该前沿的所有元素来填满父代种群，则根据基于密度的度量对该前沿的解进行排序。为了更广泛地探索解空间，较稀疏区域中的个体将优先晋级。

5.生成子代种群：通过遗传算子从父代种群中生成一个新种群。首先，进行二元锦标赛选择，随机抽样中只有排名最高的个体（依据前沿和拥挤距离排名）才能参与繁殖。然后，按照特定的交叉规则组合其中两个个体（父代）的变量（基因），生成一个新的解（子代）。最后，可以按照选定的概率对新个体进行变异，改变其中的一些变量。

6. 重复该过程：重新执行步骤 (ii) 至 (iv)，形成由父代种群和子代种群组成的新种群。由于采用了遗传学类比，算法的每次迭代都称为一代。

在数值实验中，我们使用开源 Python 框架 *pymoo 0.5。0* [51] 中的 NGSA-II 实现。

3 多目标代价下的变分约束优化

现在我们介绍多目标变分约束优化器(MOVCO)，这是一种用于求解带硬约束组合优化问题的新方法，它将量子变分框架与遗传多目标优化算法（如 NSGA-II）相结合，其中每个个体都是一组变分参数$\vec{\theta}$(见图2）。在该算法中，$\vec{}$变分波函数 $|\Psi(\theta)\rangle$ 的参数通过同时优化两个适应度函数进行迭代更新：其中一个根据约束满足情况评估解的质量

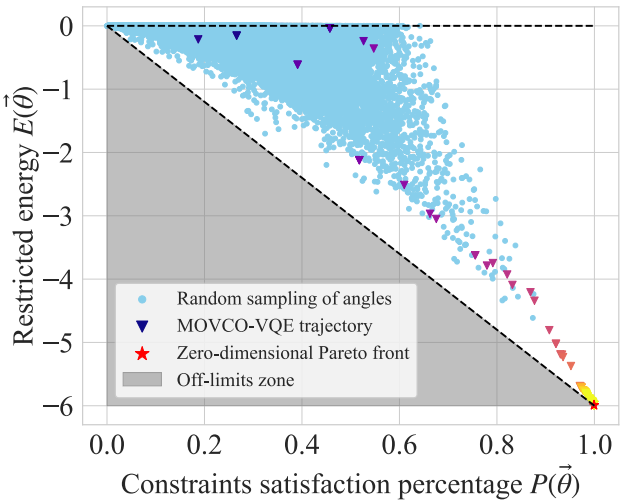


图 3：在一个包含 2 个现金点、并按第 4 节所述在 2 天周期内进行优化的 8量子比特 CMP 实例中，适应度函数空间的优化景观。交易的最高成本为 $\sum_c k_c^0 + \sum_c k_c = 18$，而满足所有约束的最低成本为 12。因此，零维帕累托前沿为 $(P,\ E)_{PF} = (1,\ -6)$。我们追踪 MOVCO-VQE 在每一代中找到的最优解轨迹。我们还展示了从随机均匀采样中抽取的 10^5组变分角所得的结果，并突出显示变分波函数无法采样的区域（在线彩色显示）。

，另一个则负责能量优化，同时避免变分算法耗费时间去优化不满足约束的区域之外的解。

- 要最大化约束满足度的适应度函数。- 理想情况下，我们希望直接最大化波函数 $|\Psi(\vec{\theta})\rangle$ 在可行空间上的投影，即满足约束的解所构成的子空间。然而，在实际场景中，这种计算并不可行，因为对于大规模系统，我们通常甚至不知道可行空间是什么。我们能高效计算的只有给定解是否满足给定约束，也就是说，我们可以判断该解是否属于这个子空间。此外，如果我们只最大化与可行空间完全重叠的解，就很容易陷入停滞，因为与整个希尔伯特空间相比，这个子空间想必很小。因此，我们最大化变分试探态所满足的约束百分比。

$$P(\vec{\theta}) = \frac{1}{K}\sum_{k=1}^{K} P_k(\vec{\theta})\ , \tag{9}$$

其中 P_k with $k = 1, ..., K$ 表示每个采样解满足的约束百分比。

- 用于最小化可行解能量的适应度函数。- 除了满足约束的状态之外，优化问题还旨在从中找到能量最低的解。为避免浪费时间从可行子空间之外的解进行采样和优化，我们不采用传统的平均能量（8），而是提出同时优化以下受限能量：

$$E(\vec{\theta}) = \sum_{k\in\mathcal{S}} \left(C_k(\vec{\theta}) - \max\left[C\right]\right)/K \tag{10}$$

其中 C_k、 $k = 1,$ ....., K 分别是针对每个采样解计算得到的成本值，$\mathcal{S}$ 是满足所有约束的解所构成的子空间。因此，仅使用属于可行空间 S 的 K 采样状态子集来计算受限能量。$\max\left[C\right]$ 是成本函数值的一个非严格上界，在实际场景中可以高效计算。例如，对于一个 QUBO 问题$C = \sum_{i,j} z_i Q_{ij} z_j$ 一个直接的上界是 $\max[C] = \sum_{i,j} |Q_{ij}|$。请注意，这种选择限制了受限能量，因为

$$0 \geq E(\vec{\theta}) \geq \sum_{k\in\mathcal{S}} \frac{1}{K}\left(\min\left[C\right] - \max\left[C\right]\right)\ ,$$

以及 $\min\left[E(\vec{\theta})\right] = \min\left[C\right] - \max\left[C\right]$, (11)

其中$1 \geq P(\vec{\theta}) \geq \sum_{k\in\mathcal{S}} \frac{1}{K}$。选择适用于计算变分量子算法能量估计量的特定采样状态，与文献中的其他思路相关，例如条件风险价值变分量子本征求解器（CVaR-VQE） [52], ，其中仅使用能量最低的状态进行能量计算。

这些适应度函数的组合构建了一个有利的景观（见图3），如第5节所示，该景观增强了算法向低能量可行解收敛的能力。

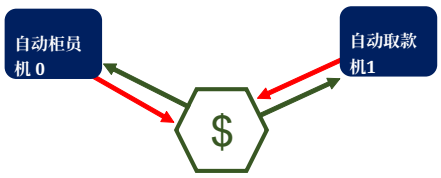
在函数空间中，使对可行空间的投影最大化并使受限能量最小化的帕累托前沿是零维的：$(P,\ E)_{PF} = (1,\ \min\ E)$。这一事实使算法能够迅速增大 P，从而经过若干次迭代后，拟设只对可行状态进行采样，避免浪费时间优化不可行解。此外，优化后的函数可以在量子处理器中同时测量。

MOVCO 实现的多目标优化具有一项优势：能够对相空间进行全局探索，而由一条最优轨迹定义的优化，其搜索范围则要狭窄得多。如第 2.3 节所述，遗传算法能够对变分参数空间中相距较远的状态进行采样，从而更高效地探索希尔伯特空间。例如，从图 3 中可以看出，即使随机采样角度，也能覆盖小规模问题的适应度函数空间中的广阔区域。这一特性与对投影 (9) 的优化相结合，能够提高收敛至满足约束的子空间的概率。

我们认为，该方法非常适用于组合优化问题，尤其是其中相当多的解至少满足部分约束的情况。如果满足任何约束的状态子空间过大，可能会形成有效的贫瘠高原：由于难以在可行空间 $\mathcal{S}$ 内采样到状态，受限能量会被困在 0。不过，许多具有工业应用价值的问题都满足这一要求。

请注意，该方法针对的是用于优化变分参数的目标函数。因此，尽管后续章节展示的结果主要围绕 VQE（MOVCO-VQE），但我们的方法与 ansatz 无关，可广泛应用于其他变分算法，例如量子近似优化算法或分层变分量子本征求解器。已知的目标函数修改形式，例如 CVaR-VQE[52]，也能从该方法中获益。在这种情况下，C_k 在（10）中的取值将不再$\mathcal{S}$中的所有状态，而只取其中能量最低的状态，这可能有助于更快地收敛到与基态重叠最小的解。

(a)



(b)

		第1天	第2天	第3	第4天
自动柜员机 0	存入/取出				-1k
	初始预测	8k	8k	9k	7k
	最终现金	8k	8k	9k	6k
自动取款机1	存入/取出	+3k	-6k		
	初始预测	4k	10k	9k	10k
	最终现金	7k	7k	6k	7k

图 4：（a）现金管理问题示意图。为确保 ATM 网络中的现金金额适当，现金会从金库中运出，并存入自动取款机或从中取出。（b）四天期间内两个经优化的自动取款机现金管理问题示例。每天每台自动取款机中的现金必须处于 [6k,9k], 范围内，且在最后一天，网络中的现金总额必须少于 13k。系统会安排最优现金转移，以满足上述要求（彩色在线版）。

4 应用于现实问题：现金管理

本节将介绍现金管理问题（现金管理问题），这是一个具有重大工业应用价值的非凸优化问题。我们还将介绍该问题的一种新颖数学公式，其规模随现金点数量和优化时间段的长度呈线性增长。

4.1 问题描述

银行日常运营任务之一，是规划向分支机构和自动取款机（ATM）配送现金，以满足其每日现金需求。此类规划基于对各分支机构和自动取款机每日可用现金的预测，并受到某些限制条件的约束。例如，若预计某台自动取款机中的现金将低于预设的某个最低金额，便会发出向该分支机构运送现金的请求。这些现金来自金库，金库中存放着未来几天可能需要的现金（见图 4）。

从金库向分支机构或自动柜员机的运输通常由一家外部公

司负责，该公司受委托向某一分支机构运送或提取现金。随后，外部公司负责运输并制定路线，而出于安全原因，路线通常会发生变化。因此，现金管理问题并不像乍看之下那样是旅行商问题的推广，因为我们无法控制具体的配送路线。相反，它是一种调度问题，与护士排班问题更为相似 [53],：需要在一个时间区间内安排一系列任务，使其满足特定的约束。此外，现金管理问题还关联着一个优化问题，因为向每个分支机构或自动柜员机配送现金都具有相应的成本，而该成本通常取决于其位置。

因此，现金管理优化问题旨在为特定地理区域内的分行和自动取款机构成的网络，确定现金配送的最优安排，以最大限度地降低交易成本，同时满足以下三项要求：

- 在所考虑的时间区间内的每一天，各分支机构和自动柜员机中的现金金额都必须高于预先设定的最低值，以确保分支机构或自动柜员机的可运营性；同时，出于安全原因，现金金额必须低于最高限额。
- 每天只能向分支机构或自动柜员机存入或取出现金一次。

此外，优化过程中还必须考虑若干硬性约束：

- 由于可用配送卡车数量有限，单日配送次数不得超过预先设定的上限。
- 现金运送和取款均以离散的现金金额为单位，从而便于管理整个运营过程。
- 出于监管要求，每个月最后三个工作日，分支机构和 ATM 网络中的现金总额不得超过预设上限。

最后还应注意，在所考虑的时间间隔的第一天安排现金配送，比其他任何一天的成本都更高，因为这相当于仅提前二十四个工作小时提出紧急请求。问题描述中还可以加入其他条件，例如某些自动柜员机在该时段存在非工作日，但本文将仅讨论本节所述的现金管理问题。

4.2 数学公式

假设有 C 家分行和自动柜员机（两者也统称为现金点），它们分布在特定的地理区域内。假设我们要规划为期 D 天的每日现金配送，覆盖 C 个现金点。我们分别用整数 $c \in [0, C)$ 和 $t \in [0, D)$ 为每个现金点和该周期内的每一天编号。现金管理问题完全由以下变量集定义：

- $k_c^0 \in \mathbb{R}^+$：在所考虑周期的第一天，于分支机构或自动柜员机汇款或取款的价格。
- $k_c \in \mathbb{R}^+, k_c < k_c^0$：在所考虑时间周期中，除第一天外的任意一天，于分支机构或自动柜员机汇款或取款的价格。
- $p_{ct} \in \mathbb{Z}$：第 t 天现金点 c 可用现金的初始预测。该预测可能超出对每个分支机构和自动柜员机每日现金金额的限制。

除了用于定义优化问题约束的变量外：

- v_l，$v_h \in \mathbb{R}^+$：分别表示在该区间内，为确保可运营性和出于安全考虑，每个分行和自动柜员机每天必须持有的现金最低值和最高值。
- $v_f \in \mathbb{R}^+$：在所考虑的时间区间的最后几天，分行和自动取款机网络中可持有的现金总额上限。为简单起见，且不失一般性，我们将时间区间的最后一天设为受此限制的日期（$t = D - 1$）。
- $l \in \mathbb{N}$：整个分支机构和自动柜员机网络中可进行的最大交易数量（现金运送和取款）

每天在整个分支机构和自动柜员机网络中完成。

这些变量的值在优化过程中无法修改，而是由各地理区域和时间区间的具体情况决定。基于这些固定变量，可以通过更改待优化变量的定义，使用不同的成本函数对现金管理问题进行编码。

在本文的公式中，待优化变量是每天各分支机构和自动柜员机可用的离散化现金 $c \quad t$, m_{ct}。为方便起见，我们定义归一化现金 M_{ct}

$$M_{ct} = \frac{m_{ct} - v_l}{\Delta m} = n\,, \tag{12}$$

其中 $n = 0$, 1, ..., h, $m_{ct} = v_l + n\Delta m$, Δm =$(v_h - v_l)/(h-1)$表示各级之间的间隔，而h是允许的离散现金值的数量。注意，编码总共需要 $V = \log_2(h) \cdot C \cdot D$ 个二进制变量$x = \{0, 1\}$或自旋变量 $z = (2x - 1)$。为简单起见，且不失一般性，我们取 $h = 4$, v_h =$h - 1$和 $v_l = 0$，因此

$$M_{ct} = 3/2 + (1/2)z^0_{ct} + z^1_{ct} \in \{0, 1, 2, 3\}\,, \tag{13}$$

其中 $z^i_{ct} \in \{-1, +1\}$ 且 $i = 0$, 1。该公式的优点在于，我们施加了所有量均处于问题约束所设定的范围内这一条件，即根据定义可满足 $M_{ct} \in [v_l, v_h]$。

我们进一步指出，资金转移和取款始终使用金额为 Δm 的离散倍数现金包进行，正如问题描述中所述。这一事实使初始预测也被离散化为 $p_{ct} \in \mathbb{Z}$，其间隔为 Δm，但与 M_{ct} （见图5c）不同，它可以取无限多个值。注意， p_{ct} 是一个非精确估计，因此将其值近似为最近的允许整数，并不会对实际应用造成明显缺陷。

为构造该问题的成本函数，定义现金点 c 在 t 日可用的现金会更方便，前提是当天该地点没有现金运送或取款，

$$\begin{aligned} W_{ct} &= p_{ct} + (M_{c,t-1} - p_{c,t-1})\ \forall\, d \neq 0, \\ W_{c0} &= p_{c0}. \end{aligned} \tag{14}$$

因此，为优化现金运送和取款的成本而需要最小化的成本函数如下：

$$\begin{aligned} C(z) = &\sum_{c\in[0,C)} k^0_c \left[1 - \delta\left(M_{c0} - p_{c0}\right)\right] \\ &+ \sum_{c\in[0,C),t\in[1,D)} k_c \left[1 - \delta\left(M_{ct} - W_{ct}\right)\right], \end{aligned} \tag{15}$$

其中，若 $x = 0$，则 $\delta(x) = 1$；否则 $\delta(x) = 0$。请注意，求解这一优化问题等价于求以下哈密顿量的基态：

$$\begin{aligned} \hat{C}(\hat{Z}) = &\sum_{c\in[0,C)} k^0_c \left[1 - \delta\left(\frac{3}{2} + \frac{1}{2}\hat{Z}^0_{c0} + \hat{Z}^1_{c0} - p_{c0}\right)\right] + \\ &\sum_{c\in[0,C),t\in[1,D)} k_c \left[1 - \delta\left(\frac{1}{2}\hat{Z}^0_{ct} + \hat{Z}^1_{ct} - \frac{1}{2}\hat{Z}^0_{c,t-1} - \hat{Z}^1_{c,t-1} - p_{ct} + p_{c,t-1}\right)\right], \end{aligned} \tag{16}$$

其中，$\hat{}\, Z^i_{ct}$ 表示作用于相应量子比特的泡利Z算符。请注意，上述哈密顿量完全由泡利Z算符定义，因此其本征态是可分离的计算基态。

如上一节所述，现金管理问题的优化还受到约束，这些约束必须用不等式表示：

- 在时间区间的最后一天，网络中所有现金点的现金总额不得超过某一特定金额：

$$G_D \equiv \sum_{c\in[0,C)} M_{cD} \leq v_f\ . \tag{17}$$

- 整个网络每天的发货和取款次数都有限制：

$$N_t \equiv \sum_{c\in[0,C)} \left[1 - \delta\left(M_{ct} - W_{ct}\right)\right] \leq l\ . \tag{18}$$

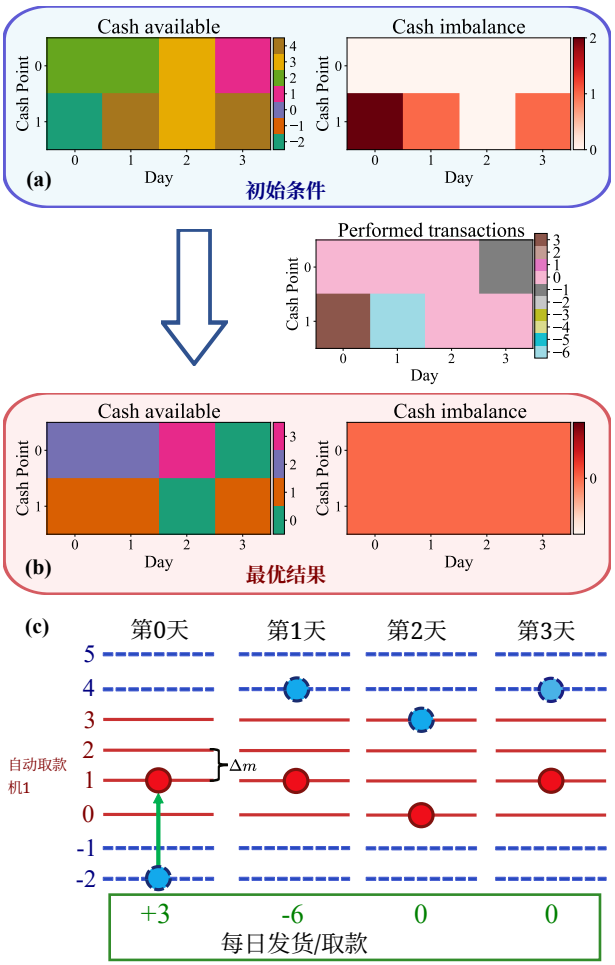


图5：一个包含两个现金点的 CMP 单实例示例，在四天期间内进行优化。(a) 初始预测 p_{cd} 给出的情形，此时未进行任何交易。(b) 最优解 m_{cd}。(c) 现金点一的情景：红色虚线和蓝色实线分别表示约束区间 $[v_l, v_h]$ 内外的离散现金水平，而红色虚线圆圈和蓝色实线圆圈分别表示初始预测和最优结果（在线彩色显示）。

因此，总共有 $D+1$ 个额外约束。将这些约束纳入优化过程的一种方法，是在成本函数中加入惩罚项；当解不满足任何约束时，这些惩罚项会提高成本函数的值。在本文中，我们探讨了对成本函数进行多目标优化的优势，这种优化能够考虑这些约束，而无需加入额外项。

4.3 单实例示例

为清晰起见，我们举例说明一个包含两个现金点的 CMP，并对四天期间内的现金交易进行优化。实际情景如图4b 所示。在离散空间中，每个分支机构或自动柜员机的可用现金初始预测为

$$p_{ct} = \begin{pmatrix} 2 & 2 & 3 & 1 \\ -2 & 4 & 3 & 4 \end{pmatrix},$$

其配送成本为 $k_0^0 = 4$，$k_1^0 = 8$，以及 $k_0 = 2$，$k_1 = 4$。问题的约束由 $v_l = 0$，$v_h = 3$，$v_f = 1$，$l = 1$ 定义。如图5a 所示，在未进行任何交易的情况下，现金点 1 违反了可用现金约束。由两个现金点的现金总额之和构成的现金总额约束，也在该区间的最后一天被违反。此实例的最优状态满足所有约束并使配送成本最小，它将

$$M_{ct} = \begin{pmatrix} 2 & 2 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$

或用自旋变量 z_{ct}^i 表示，

$$M_{ct} = \begin{pmatrix} (-1,1) & (-1,1) & (1,1) & (-1,-1) \\ (1,-1) & (1,-1) & (-1,-1) & (1,-1) \end{pmatrix},$$

这对应于执行图中所示的交易（现金运送和取款）：图4b 和图 5。最优解的成本 C 为 10。

5 数值结果

接下来，我们在一个具有重大工业价值且约束颇具挑战性的现实问题上测试所提出的变分优化方法：上一节中提出的现金管理问题。鉴于该问题的特点，我们采用算法的VQE变体（MOVCO-VQE），因为它在构造浅层且表达能力强的拟设方面具有灵活性。此外，我们还将其与使用变分量子本征求解器优化CMP、并将约束编码为惩罚项所得的结果进行比较。计算实验采用随机生成的CMP实例进行，生成方法如附录B中所述。

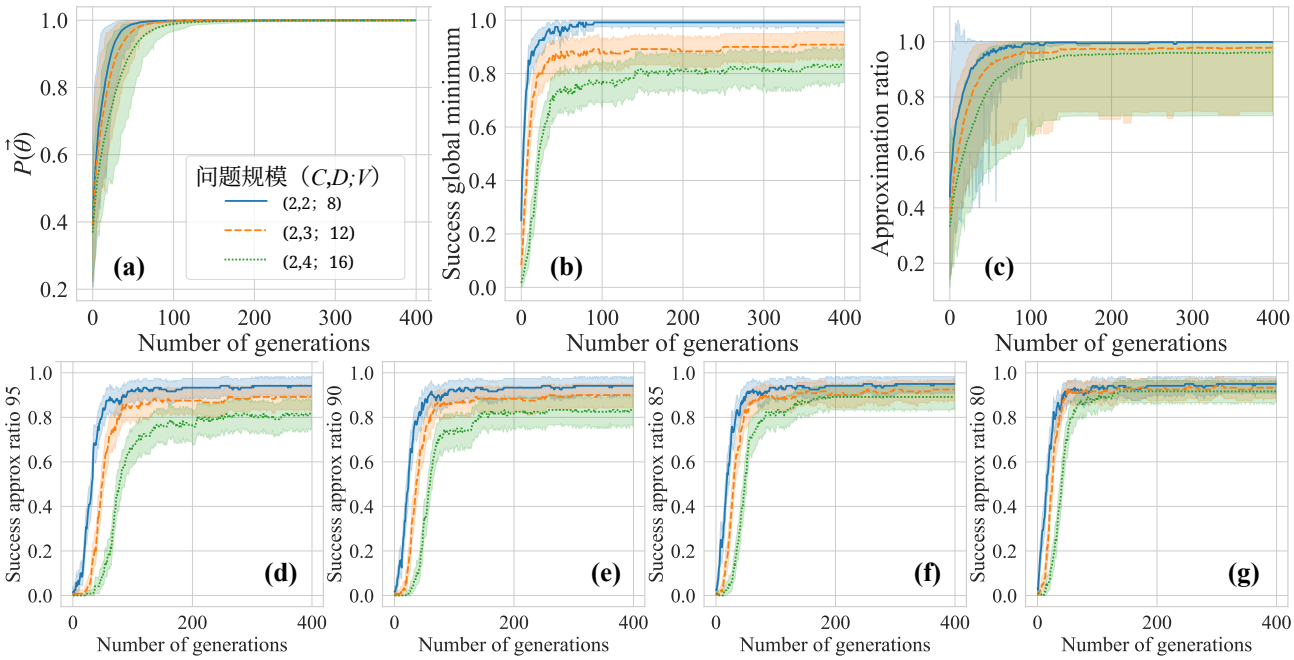


图6：使用单层拟设 $V = 2CD$ 对120个CMP实例进行求解所得的实证结果（每个周期内有2个现金点，周期分别为2、3和4天；其中3代表对问题进行编码所需的变量/量子比特数）。采用MOVCO-VQE进行经典模拟(3)。（a）8192个采样解中满足约束的比例的平均值。（b）变分波函数与全局最小值的重叠度高于0.1的实例比例。（c）近似比的平均值。（d-g）成功率；当实例的近似比分别高于0.95、0.9、0.85和0.8时，视为成功。图（a）和（c）显示围绕平均值的95% 百分位区间，范围为分布的第2.5至第97.5百分位。图（b）和（d-g）显示平均值估计的95% 置信区间（在线彩色版）。

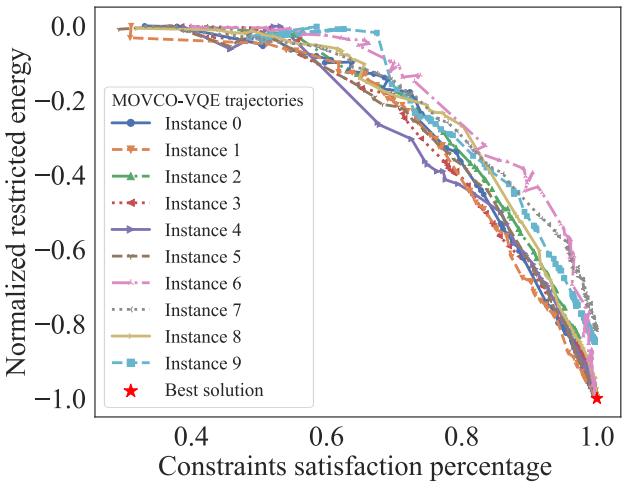


图7：使用 MOVCO-VQE 求解 10 个独立CMP实例（每个实例包含 2 个现金点，时间跨度为 4 天）时所得解的演化。每条线对应一个实例在适应度函数空间中的轨迹，展示了随着算法迭代次数增加而发生的变化。所有样本的受限能量均归一化至 0 到 -1 之间（网络版为彩色）。

5.1 多目标方法的性能

我们分析了随着代数增加，该算法在不同问题规模下的收敛性和效率（图6）。我们使用第2.1.1节所述的一层VQE拟设，变分参数按照NSGA-II流程更新，并采用二元锦标赛选择、模拟二进制交叉 [54], 和多项式变异 [55]作为遗传算子。遗传算法的种群规模设为10，每代还会产生10个新的个体。这意味着每一代中，变分算法都会使用不同的一组$\vec{}$参数 θ 对量子电路进行十次评估。在此处以及后续结果中，量子电路均在无噪声条件下进行经典模拟，通过采样 $K = 8192$ 比特串来模拟量子计算机上有限次数的测量所产生的最终概率分布。

为了开展基准测试，我们将使用一些指标来评估解的质量，具体包括

它们满足的约束数量以及这些解的交易成本，无论采用哪种变分量子算法。对于约束满足情况，我们可以直接使用适应度函数 $P(\vec{\theta}_{sol})$ ，见公式 (9)，其中 $\vec{\theta}_{sol}$ 为最终的变分参数。对于交易成本，我们将使用近似比 $\epsilon(\vec{\theta})$ ，其定义如下：

$$\epsilon(\vec{\theta}) = \frac{C_{max} - \langle\Psi(\vec{\theta})|\,\hat{C}\,|\Psi(\vec{\theta})\rangle}{C_{max} - C_{min}}, \qquad (19)$$

其中，$C_{max} = \sum_c k_c^0 + (D-1)\sum_c k_c$ 是可能的最大成本，而在满足所有约束的前提下，C_{min} 是可达到的最低成本。因此，当 $P(\vec{\theta}_{sol}) = 1$ 且 $\epsilon(\vec{\theta}_{sol}) = 1$ 时，算法达到了最优解，即满足约束的全局最小值 $|\Psi_{gs}\rangle$。我们还通过成功率分析变分波函数与全局最小值的重叠程度 $\rho = |\langle\Psi(\vec{\theta})|\Psi_{gs}\rangle|^2$ 。如果 $\rho > 0.1$，我们就认为可以从波函数中轻松采样到全局最小值，因此视为成功。

在图6a中，我们可以看到该方法如何避免变分波函数陷入可行空间之外的局部最小值。事实上，经过该算法两百代后，所有实例的解都满足现金管理问题的全部约束。此外，如图6c-g所示，这些解的交易成本较低。尽管图6c显示近似比存在一定波动，但其均值较高，表明大多数实例都能得到低能量解。这一点也可以从图6d-g中看出，图中绘出了近似比达到 $\epsilon(\vec{\theta}_{sol}) > 0.95$、0。9、0。85、0。8的实例比例。即使对于规模最大的实例，大多数实例也能得到近似比高于 80%的解。至于精确最优解，从图6b中可以看到，对于约 80%的实例（在 4 天内有 2 个现金点），以及约 90% 的实例（在 3 天内有 2 个现金点），变分波函数仅经过一百代就能高效地采样到全局最小值。

我们可以将该算法的执行过程分为两个阶段（见图 7）。在第一阶段，训练会提高投影 $P(\vec{\theta})$，而能量 $E(\vec{\theta})$ 可能保持在 0 不变。当投影足够高时，波函数开始在 $\mathcal{S}$ 中采样解，并且随着投影不断增加，能量也会同时最小化，从而逐步使优化仅在满足约束的子空间中进行。这种行为不仅可以加速第二阶段的收敛，还能避免陷入局部最小值，并且由于第一阶段所进行的训练，大幅降低最终落在满足约束的子空间之外的概率。对于小规模系统，或具有较高态密度且满足所有约束的问题（例如图 6 和图 8 中实现的问题），第一阶段可以很快完成。在这种情况下，从第一代开始，能量也会得到最小化。

5.2 与惩罚项方法的比较

在确认多目标方法能够良好收敛到最优解后，我们研究 MOVCO 相较于标准方法的优势。

处理硬约束的一种传统技术是将惩罚项纳入原始哈密顿量中进行变换。当解不满足约束时，这些项会增大哈密顿量的期望值。在现金管理问题中，加入惩罚项的哈密顿量可表示为

$$\hat{C}_{pen}(\hat{Z}) = \hat{C}(\hat{Z}) + \lambda_f L\left(G_D - v_f\right) + \lambda_l \sum_{t\in[0,D)} L\left(N_t - l\right) \qquad (20)$$

ˆ其中 C、G_D 和 N_t 在第 4.2 节中定义，L 是赫维赛德阶跃函数（$L(x) = 1$ 当 $x \geq 0$，$L(x) = 0$ 其他情况，$x \in \mathbb{R}$），而 λ_f 和 λ_l 是可调的正超参数。因此，变分优化就转化为寻找使 $\langle\Psi(\vec{\theta})|\,\hat{C}_{pen}(\hat{Z})\,|\Psi(\vec{\theta})\rangle$ 最小的状态。请注意，只要 λ_f 和 λ_l 足够大，该状态就会满足所有约束。

我们在具有 2 个现金点、优化周期为 4 天的 CMP 实例中，将 MOVCO-VQE 与带惩罚项的 VQE 进行比较，即针对 16 量子比特问题进行比较。在 VQE 中，变分参数通过同时扰动随机逼近（SPSA）[56]进行优化。这种无梯度经典优化器每次迭代仅需对量子电路进行两次评估，不受自由参数数量影响，因此是变分量子算法中效率较高的方法之一。[13]。惩罚超参数的取值经过选择，使其满足以下条件：

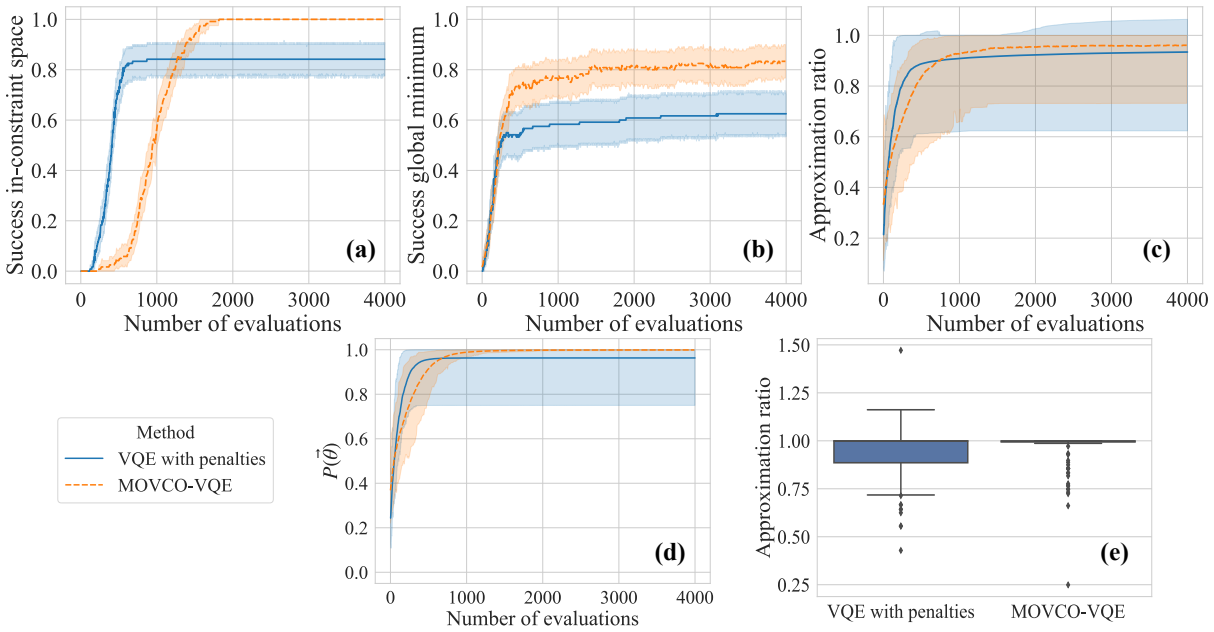


图 8：MOVCO-VQE 与带惩罚项的 VQE 性能比较。我们展示了 120 个 CMP实例（4 天周期内有 2 个现金点）的结果。我们使用单层拟设 (3) 和 8192 次测量，对两种算法进行了经典模拟。(a) 几乎所有采样解都位于可行空间内的实例百分比。(b) 变分波函数与全局最小值的重叠度高于 0.1 的实例百分比。(c、e) 近似比的平均值和分布。(d) 采样解满足的约束百分比的平均值。图 (c) 和 (d) 显示了围绕平均值的 95% 百分位区间，其范围为分布的第 2.5 至第 97.5 百分位数。图 (a) 和 (b) 显示了平均值估计的 95% 置信区间。图 (e) 中的箱体表示下四分位数和上四分位数，须线延伸至 1.5 倍四分位距（彩色在线）。

任何不满足问题任一约束的解，其成本都高于可行子空间中的解。为提高实验结果的意义，我们按照附录A中的说明，对这些取值进行调优，以最大化平均约束满足度和平均近似比 $\lambda_f = \lambda_l = 25$。对于 MOVCO-VQE 和 VQE，我们再次使用前文介绍的单层拟设(3)，并采用无噪声量子计算机模拟，测量次数为 $K = 8192$ 。

如图所示9，在所有考量的指标上，MOVCO-VQE 均优于带惩罚项的算法。为了公平比较，我们考虑到，同时扰动随机逼近每次迭代只需查询量子处理器两次，而 MOVCO 每次所需的查询次数则高达遗传算法的种群规模（在我们的设置中为 10）。因此，我们根据量子计算机上的评估次数绘制结果。MOVCO-VQE 的主要优势在于，经过若干次迭代后，它能够仅在可行空间中进行探索；而带惩罚项的 VQE 则会陷入约束内区域之外的局部最小值。这一点可从图中看出8a，其中我们展示了最终波函数与约束内子空间的重叠几乎达到完全重合的实例所占的比例，即 $P(\vec{\theta}_{sol}) > 0.99$。这一现象也很明显于图8d 中，我们在该图中直接展示了 $P(\theta)$。

达到全局最小值的实例比例也有所提高，解的成本有所降低。此外，正如图中所示，这种收敛性的改善伴随着结果离散程度的降低。8c–e。在图8e 中，近似比高于 1 的实例达到了较低的能量，但得到的解不可行。值得注意的是，MOVCO 借助遗传算法机制，可在每次迭代中并行处理各项量子电路评估，从而大幅缩短执行时间。

MOVCO 的优势源于多目标方法与遗传算法技术的结合。作为测试，我们还分析了在使用单目标遗传算法优化变分参数时，带惩罚项的变分量子本征求解器（VQE）的性能。

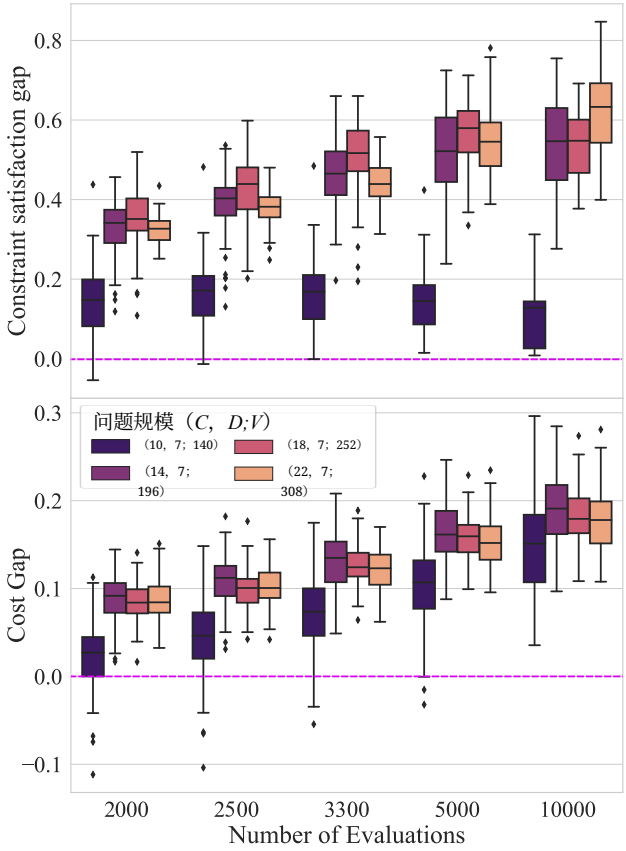


图 9：使用乘积态时，随着迭代次数增加，MOVCO 方法与惩罚方法性能的比较。两种算法均求解相同的 80 个实例，并通过以下方式比较解的质量：（a）采样态所满足约束比例的差异；（b）采样解之间交易成本的百分比差异。在粉色虚线以上的情况下，MOVCO 获得了质量更高的解。箱体表示下四分位数和上四分位数，须线延伸至 1.5 倍四分位距，其余点是该分布之外的观测数据（在线彩色显示）。

具体而言，我们采用了与 MOVCO 相同的遗传算子和超参数，但每次迭代中解的排序依据是经惩罚处理后的能量的平均值 $\langle\Psi(\vec{\theta})|\,\hat{}C_{pen}(\,\hat{}Z)\,|\Psi(\vec{\theta})\rangle$。附录 C显示，结果与图 9 中使用同时扰动随机逼近（SPSA）得到的结果相似，因此，所提出的多目标优化保留了 MOVCO 的优势。

5.3 使用乘积态对更大规模系统进行基准测试

对纠缠态进行经典模拟，例如（3），需要付出指数级增长的资源成本

这使得我们无法分析变量数量较多的问题。为了在计算实验中增加现金点和天数，并比较两种算法在这种情况下的性能变化，我们需要降低变分试探态的复杂度。因此，我们现在使用完全可分拟设来分析这两种方法的性能，

$$|\Psi_{sp}(\vec{\theta})\rangle = \prod_{n}^{N} e^{i\theta_n \hat{Y}_n}\,|0\rangle = \bigotimes_{n=1}^{N} |\psi(\theta_n)\rangle\,, \qquad (21)$$

其中 $|\psi(\theta_n)\rangle = \cos\theta_n\,|0\rangle + \sin\theta_n\,|1\rangle$。这种变分形式是拟设（3）在层数为 0 时的边界情况。由于我们消除了量子电路中的纠缠，因此可以高效地采样（21）的变分波函数概率分布，只需使用一个包含 N个元素 $\{\cos^2\theta_n\}$的数组即可。请注意，这种简单的架构已经能够再现希尔伯特空间中的任意经典态。因此，始终存在一组参数 $\vec{\theta}_{sol}$，使得（21）是 CMP 哈密顿量的基态。与前述结果中测试的带纠缠层的变分量子算法相比，这一类状态可视为一种量子启发式替代方案。

我们通过优化包含 10 到 22 个现金点的自动柜员机网络来测试这些算法，优化周期为整整一周，对应的问题最多包含$V = 308$ 个变量。由于问题规模增大，在这些结果中，我们采用种群规模设为 100 的 MOVCO-VQE，以及带惩罚项的 VQE，其中 $\lambda_f = \lambda_l = 50$。每次评估采样的比特串数量仍为 K $= 8192$。具体步骤如下：我们按照前文所述生成一个随机 CMP 实例，并使用这两种算法求解该优化问题。由于这些实例规模较大，无法通过穷举搜索得知问题的精确解，因此我们通过直接比较两种方法所得的解来进行基准测试。我们使用两个指标：约束满足差距定义为

$$P_{gap} = P(\vec{\theta}_{MOVCO}) - P(\vec{\theta}_{VQEpen}) \qquad (22)$$

以及成本差距

$$C_{gap} = \frac{C_{VQEpen} - C_{MOVCO}}{C_{VQEpen}} \qquad (23)$$

其中 $C_i = \langle \Psi_{sp}(\vec{\theta_i}) | \hat{C} | \Psi_{sp}(\vec{\theta_i}) \rangle$ 中，$i = \{VQEpen, MOVCO\}$，而 $\vec{\theta}_{MOVCO}, \vec{\theta}_{VQEpen}$分别是 MOVCO-VQE 和带惩罚项的 VQE 算法的最终角度。

在图9 中，我们展示了随着代价函数评估次数增加，解的演变情况。粉色线以上的情形表明 MOVCO-VQE 得到了更好的解。在采样解的约束满足情况和交易成本方面，都可以明显看出这种改进；随着算法迭代次数增加，改进效果也愈加显著。事实上，当算法完成 10000 次评估时，我们在100% 的实例中都得到了更好的解。

6 结论

本文介绍了多目标变分约束优化器（MOVCO），这是一种用于求解带硬约束组合优化问题（CO）的变分量子方法。该方法通过遗传算法同时优化两个适应度函数，提升变分量子算法（VQAs）的性能：一个函数用于衡量约束的满足情况，另一个则用于最小化可行子空间内解的能量。

我们针对一个极具现实意义的工业问题——现金管理问题（CMP），提供了 MOVCO 性能稳健的实证证据。我们提出了现金管理问题的一种新颖二元变量表述，并使用 MOVCO 版本的变分量子本征求解器（VQE）进行求解。我们将这些结果与一种标准方法进行比较；该方法使用带惩罚项的成本函数进行 VQE 优化，人为提高不可行状态的能量。研究表明，MOVCO 有助于避免不可行极小值，同时促进收敛至能量更低的解。深入研究遗传算法超参数（如种群规模或遗传算子）对 MOVCO 性能的影响，或许能够带来更好的结果。

MOVCO 方法与 ansatz 无关，因此可应用于除 VQE 之外的广泛变分量子算法。未来可以开展新的研究，将该方法用于依赖问题的拟设，例如量子近似优化算法；在这类算法中，CO 通常表示为二次无约束二元优化，并映射到量子处理器。还可以进一步改进 MOVCO。例如，可以分析 MOVCO 的条件风险价值版本，其中仅使用采样状态中能量最低的一定比例来计算多目标优化的适应度函数。由于 CO 不要求变分波函数与满足约束的子空间完全重叠，而只要求以较高概率采样到低能量的可行解，另一种有趣的可能性是，为满足所有约束的采样解比例设定上限。

本研究进一步阐明了变分算法在具有实际意义的优化问题中的应用。据我们所知，本研究也是首次尝试使用量子计算求解现金管理问题（CMP）。因此，本研究或可为进一步研究这一问题，以及具有硬约束的大量现实世界中的组合优化问题中的其他问题开辟道路。

致谢

本研究得到西班牙 CDTI 通过科学与创新使命计划（CUCO）项目（资助编号：MIG-20211005）的支持；同时得到由 MCIN/AEI/10.13039/501100011033/FEDER,UE 资助的 PID2021-127968NB-I00 项目支持，以及 CSIC 跨学科主题平台（PTI）量子技术（PTI-QTEP）的支持。P. D.-V. 还感谢 CAM/FEDER 项目（编号：S2018/TCS-4342（QUITEMAD-CM））的支持。作者还衷心感谢 SGAI-CSIC 科学计算部（AIC）在使用 DRAGO 超级计算机进行模拟时提供的协助，并感谢加利西亚超级计算中心（CESGA）提供 FinisTerrae 超级计算机的使用权限。

作者贡献声明

P. D.-V. 制定了优化问题的数学公式，设计了量子算法并进行了数值计算。J. L.-H. 和 S. H.-S. 提出了现金管理问题。D. P.、E. S.-M. 和 J. J. G.-R. 构思并指导了这项研究，所有作者都参与了论文的撰写。

免责声明

本文纯属科学与信息性内容，并非 BBVA SA 或其任何子公司的产品。BBVA 及其子公司既不了解本文的前提、结论或总体内容，也不一定认同这些内容。因此，本文的原创性、准确性、可靠性及其他相关责任均由作者独自承担。本文不构成投资研究或投资建议，也不构成购买或出售任何证券、金融工具、金融产品或服务的建议、要约或招揽，亦不得以任何方式用于评估参与任何交易的价值。

参考文献

[1] Markus Reiher、Nathan Wiebe、Krysta M. Svore、Dave Wecker 和 Matthias Troyer。阐明量子计算机上的反应机理。美国国家科学院院刊，114(29):7555–7560，2017年7月。

[2] Vera von Burg、Guang Hao Low、Thomas Häner、Damian S. Steiger、Markus Reiher、MartinRoetteler 和 MatthiasTroyer。量子计算增强计算催化。《物理评论研究》，3(3)，2021年7月。

[3]Michael Lubasch、Jaewoo Joo、Pierre Moinier、Martin Kiffner 和 Dieter Jaksch。用于非线性问题的变分量子算法。*Physical Review A*，101(1)，2020年1月。

[4]Paula Garcí a-Molina、Javier Rodríguez- Mediavilla 和 Juan José García-Ripoll。用于多元函数的量子傅里叶分析及其在一类薛定谔型偏微分方程中的应用。*Physical Review A*，105(1)，2022年1月。

[5] Jacob Biamonte、Peter Wittek、Nicola Pancotti、Patrick Rebentrost、Nathan Wiebe 和 Seth Lloyd。量子机器学习。《自然》，549(7671):195–202，2017年9月。

[6]Marcello Benedetti、Erika Lloyd、Stefan Sack 和 Mattia Fiorentini。参数化量子电路作为机器学习模型。*QuantumScienceand Technology*，4(4):043001，2019年11月。

[7]Alejandro Perdomo-Ortiz、Marcello Benedetti、John Realpe-Gómez 和 Rupak Biswas。近期量子计算机中量子辅助机器学习的机遇与挑战。*QuantumScience andTechnology*，3(3):030502，2018年6月。

[8] Nikolaj Moll 等人。近期量子设备上使用变分算法进行量子优化。*Quantum Sci. Technol*，3 030503，2018年。[9]George L. Nemhauser 和 Laurence A. Wolsey。整数与组合优化。John Wiley & Sons, Ltd，1988年。

[10] J. K. Lenstra 和 A. H. G. Rinnooy Kan。旅行商问题的一些简单应用。运筹学会期刊，26(4):717–733，1975年。

[11]Michel X. Goemans 和 David P. Williamson。使用半定规划改进最大割和可满足性问题的近似算法。*J. ACM*，42：1115–1145，1995。

[12] P. Festa、P.M. Pardalos、M.G.C. Resende 和 C.C. Ribeiro。最大割问题的随机化启发式方法。*Optimization Methodsand Software*，17(6)：1033–1058，2002年。

[13] M. Cerezo、Andrew Arrasmith、Ryan Bab- bush、Simon C. Benjamin、Suguru Endo、Keisuke Fujii、Jarrod R. McClean、Kosuke Mitarai、Xiao Yuan、Lukasz Cincio 和 Patrick J. Coles。变分量子算法。*Nature ReviewsPhysics*，3(9)：625–644，2021年8月。

[14] John Preskill。《NISQ时代及未来的量子计算》。*Quantum*，2:79，2018年8月。[15] Kishor Bharti、Alba Cervera-Lierta、Thi Ha Kyaw、Tobias Haug、Sumner Alperin-Lea、Abhinav Anand、Matthias Degroote、Hermann Heimonen、Jakob S. Kottmann、Tim Menke、Wai-Keong Mok、Sukin Sim、Leong-Chuan Kwek 和 Alán Aspuru-Guzik。含噪中等规模量子算法。*Reviews of Modern Physics*，94(1)，2022年2月。

[16] Amparo Domínguez、Angel Juan 和 Renatas Kizys。关于元启发式算法金融应用的综述。*ACM ComputingSurveys*，50:1–23，2017年4月。

[17]Abdelkader Sbihi 和 Richard W. Eglese。组合优化与绿色物流。*Annalsof OperationsResearch*，175：159–175，2010年。

[18] Camila Bezerra、Lucas Carneiro、Elck Carvalho、Thiago Chagas、Lucas Carvalho、Ana Uetanabaro、Gervásio Da Silva、Erik Galvão Paranhos da Silva 和 Andréa Costa。人工智能作为一种组合优化策略，利用固态发酵处理桃棕榈废弃物，以生产哈茨木霉 AM7 的纤维素酶。*BioEnergy Research*，14，2021年12月。

[19]Majid Eskandarpour、Pierre Dejax、J oe Miemczyk 和 Olivier Péton。可持续供应链网络设计：面向优化的综述。*Omega*，54：11–32，2015年。

[20]J. Phillip Kennedy、Lyndsey Williams、Thomas M. Bridges、R Nathan Daniels、Dave Weaver 和 Craig W. Lindsley。组合化学科学在现代药物发现中的应用。组合化学期刊，10 3:345–54，2008。

[21]Stuart Hadfield、Zhihui Wang、Eleanor Gilbert Rieffel、Bryan O'Gorman、Davide Venturelli 和 Rupak Biswas。带硬约束和软约束的量子近似优化。第二届后摩尔时代超级计算国际研讨会论文集，2017。

[22]Stuart Hadfield、Zhihui Wang、Bryan O'Gorman、Eleanor Rieffel、Davide Venturelli 和 Rupak Biswas。从量子近似优化算法到量子交替算子拟设。算法，12(2):34，2019年2月。

[23] Pradeep Niroula, RuslanShaydulin, Romina Yalovetzky, Pierre Minssen, Dylan Herman, Shaohan Hu 和 Marco Pistoia。面向离子阱量子计算机上抽取式摘要的受约束量子优化。*Scientific Reports*，12(1):1– 14，2022年。

[24] Andrew Lucas。许多 NP 问题的伊辛表述。*Frontiers in physics*，第 5 页，2014年。[25] Tianyi Hao、Ruslan Shaydulin、Marco Pis- toia 和 Jeffrey Larson。利用受约束变分量子优化中的约束内能量，2022年。[26] Giacomo Nannicini。用于组合优化的量子-经典混合变分启发式的性能。*Phys. Rev. E*，99:013304，2019年1月。[27]Madita Willsch、Dennis Willsch、Fengping Jin、Hans De Raedt 和 Kristel Michielsen。量子近似优化算法的基准测试。量子信息处理，19(7)，2020年6月。

[28]Anton Robert、Panagiotis Kl. Barkoutsos、Stefan Woerner 和 Ivano Tavernelli。用于蛋白质折叠的高效资源量子算法。*npj Quantum Information*，7(1)，2021年2月。

[29]David Amaro、Matthias Rosenkranz、Nathan Fitzpatrick、Koji Hirano 和 Mat- ti a Fiorentini。变分量子算法在作业车间调度问题中的案例研究。*EPJ Quantum Technology*，9(1)，2022年2月。

[30] Utkarsh Azad、Bikash K. Behera、Emad A. Ahmed、Prasanta K. Panigrahi 和 Ahmed Farouk。使用量子近似优化算法求解车辆路径问题。*IEEE*智能交通系统汇刊，第1–10页，2022年。

[31] Sebastian LeonticaandDavidAmaro。使用瞬时量子多项式电路进行量子优化，2022年。

[32] Gavin E. Crooks。《量子近似优化算法在最大割问题上的性能》，2018年。

[33]Michael Streif 和 Martin Leib。用于一级量子近似优化算法和瞬时量子多项式电路的禁用子空间。《物理评论A》，102:042416，2020年10月。[34]Pablo Díez-Valle、Diego Porras 和 Juan José García-Ripoll。量子近似优化算法伪玻尔兹曼态，2022年。

[35] Edward Farhi、Jeffrey Goldstone、Sam Gutmann 和 Leo Zhou。无限规模下的量子近似优化算法与 Sherrington-Kirkpatrick 模型。《Quantum》，6:759，2022年7月。

[36] Edward Farhi、Jeffrey Goldstone 和 Sam Gutmann。《量子近似优化算法》，2014年。

[37]Alberto Peruzzo、Jarrod McClean、Peter Shadbolt、Man-Hong Yung、Xiao-Qi Zhou、Peter J. Love、Alán Aspuru-Guzik 和 Jeremy L. O'Brien。在光子量子处理器上实现变分本征值求解器。*Nature Communications*，5(1)，2014年7月。

[38]Sergey Bravyi、Alexander Kliesch、Robert Koenig 和 Eugene Tang。对称性保护带来的变分量子优化障碍。*Phys. Rev. Lett.*，125:260505，2020年12月。

[39]David Amaro、Carlo Modica、Matthias Rosenkranz、Mattia Fiorentini、Marcello Benedetti 和 Michael Lubasch。用于组合优化的过滤变分量子算法。*Quantum Science andTechnology*，7(1):015021，2022年1月。

[40]Xiaoyuan Liu、Anthony Angone、Ruslan Shaydulin、Ilya Safro、Yuri Alexeev 和 Lukasz Cincio。Layer VQE：一种在含噪量子计算机上进行组合优化的变分方法。*IEEE Transactions on Quantum Engineering*，3:1–20，2022年。

[41] Linghua Zhu、Ho Lun Tang、George S. Barron、F. A. Calderon-Vargas、Nicholas J. Mayhall、Edwin Barnes 和 Sophia E. Economou。一种用于在量子计算机上求解组合问题的自适应量子近似优化算法，2020年。[42] Sam McArdle、Tyson Jones、Suguru Endo、Ying Li、Simon C. Benjamin 和 Xiao Yuan。基于变分试探态的虚时间演化量子模拟。*npjQuantum Information*，5(1)，2019年9月。

[43]Joonho Lee、William J. Huggins、Martin Head-Gordon 和 K. Birgitta Whaley。用于量子计算的广义幺正耦合簇波函数。期刊《化学理论与计算杂志》，15(1):311–324，2018年11月。

[44]Dave Wecker、Matthew B. Hastings 和 Matthias Troyer。迈向实用变分量子算法的进展。*PhysicalReview A*，92(4)，2015年10月。

[45]Roeland Wiersema、Cunlu Zhou、Yvette de Sereville、Juan Felipe Carrasquilla、Yong Baek Kim 和 Henry Yuen。探索哈密顿量变分试探态中的纠缠与优化。*PRX Quantum*，1(2)，2020年12月。

[46] Jarrod R. McClean、Sergio Boixo、Vadim N. Smelyanskiy、Ryan Babbush 和 Hartmut Neven。量子神经网络训练景观中的贫瘠高原。《自然》通讯，9(1)，2018年11月。

[47] K. Deb、A. Pratap、S. Agarwal 和 T. Me-yarivan。一种快速且具精英策略的多目标遗传算法：NSGA-II。《*IEEE*进化计算汇刊》，6(2):182–197，2002年。[48] Kalyanmoy Deb。《使用进化算法进行多目标优化：入门》，第 3–34 页。Springer London，伦敦，2011年。[49] Antonio J. Nebro、Jesús Galeano-Brajones、Francisco Luna 和 Carlos A. Coello Coello。NSGA-II 是否已适用于大规模多目标优化？《数学与计算应用》，27(6)，2022年。[50]S. Katoch、S.S. Chauhan 和 V. Kumar。遗传算法综述：过去、现在与未来。*Multimed Tools Appl*，80:8091–8126，2021年。[51]J. Blank 和 K. Deb。pymoo：Python 中的多目标优化。*IEEE Access*，8:89497–89509，2020年。[52]Panagiotis Kl. Barkoutsos、Giacomo Nannicini、Anton Robert、Ivano Tavernelli 和 Stefan Woerner。利用条件风险价值改进变分量子优化。*Quantum*，4:256，2020年4月。[53] Edmund K. Burke、Patrick De Causmaecker、Greet Vanden Berghe 和 Hendrik Van Landeghem。护士排班的最新进展。《排程期刊》，7:441–499，2004年。

[54] Kalyanmoy Deb 和 Ram Bhushan Agrawal。用于连续搜索空间的模拟二进制交叉。*Complex Syst.*，9，1995年。[55] Kalyanmoy Deb 和 Mayank Goyal。一种用于工程设计的遗传自适应搜索（geneas）综合方法。*Computer Science andInformatics*，26(4):30–45，1996年。

[56] J.C. Spall。利用同时扰动梯度逼近进行多元随机逼近。*IEEE*自动控制汇刊，37(3):332–341，1992。

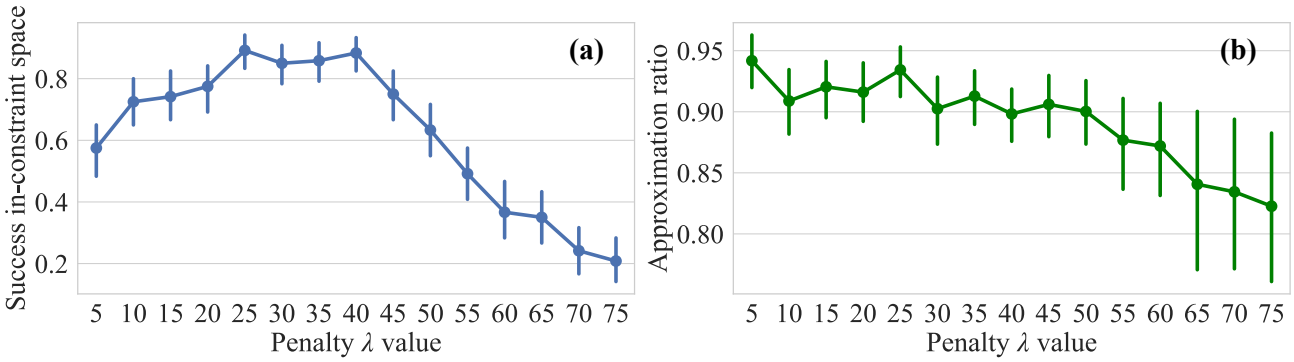


图10：采用不同超参数 $\lambda_l = \lambda_f$ 值优化现金管理问题惩罚成本函数(20) 时变分量子本征求解器的性能比较。结果显示了在四天期间内优化两个现金点的120个随机实例的平均值；采用同时扰动随机逼近算法迭代500次后得到这些结果。(a) 几乎所有采样解都位于可行空间内的实例所占百分比（$P(\vec{\theta}) > 0.99$）。(b) 近似比的平均值（仅考虑满足 $P(\vec{\theta}) > 0.99$的实例）。两幅图均显示平均估计值的95% 置信区间。（在线彩色版）。

惩罚超参数调优

如论文所述，解决带有硬约束的组合优化问题的一种典型方法，是像式(20)中那样，在成本函数中加入惩罚项。惩罚超参数 λ_l 和 λ_f 控制可行解与不可行解之间的差距，其取值会影响算法的性能。这些超参数取值较低时，可能会加快算法收敛，但也会增加陷入局部最小值的可能性，反之亦然。

为了提升第5.2节中MOVCO-VQE与惩罚版变分量子本征求解器之间比较的说服力，我们通过数值研究这些超参数的取值如何影响惩罚版变分量子本征求解器的平均性能，并选出最佳取值。我们针对现金管理问题的120个随机生成实例（每个实例在四天期间内包含两个现金点），采用不同的超参数取值，对经典模拟的算法进行了500次迭代。与第5.2节一样，我们使用了拟设(3)、同时扰动随机逼近优化器和8192次测量。由于 λ_l 与 λ_f 在定性方面没有差异，我们选择了 $\lambda_l = \lambda_f$。结果如图10所示，其中比较了几乎所有采样解都位于可行空间内的实例所占百分比，以及得到的近似比。我们注意到， $\lambda_l = \lambda_f = 25$ 取得了最佳结果。

B 现金管理实例生成

第5 节所示的计算实验中，包含 C 个现金点的现金管理实例是按照以下流程随机生成的：

- 在分支机构或自动柜员机 c 发送或提取现金的价格 k_c 是从区间 $[1, 4]$ 内的离散均匀分布中抽取的一个整数。第一天的价格是其两倍， $k_c^0 = 2k_c$。
- 每个现金点 c 在每一天 d 可用现金的初始预测 p_{cd} 是从区间 $[-2, 5]$ 内的离散均匀分布中抽取的一个整数。
- 时间区间最后一天分行和自动取款机网络中的现金总额上限设定为 $v_f = C$。
- 每天最多可进行的交易（现金运送和取款）次数为：如果 $C = 2$，则为 $l = 1$； $l = \lfloor \frac{3}{4}C \rceil$ 否则（其中 $\lfloor \cdot \rceil$ 表示取整到最接近整数值的取整函数）。

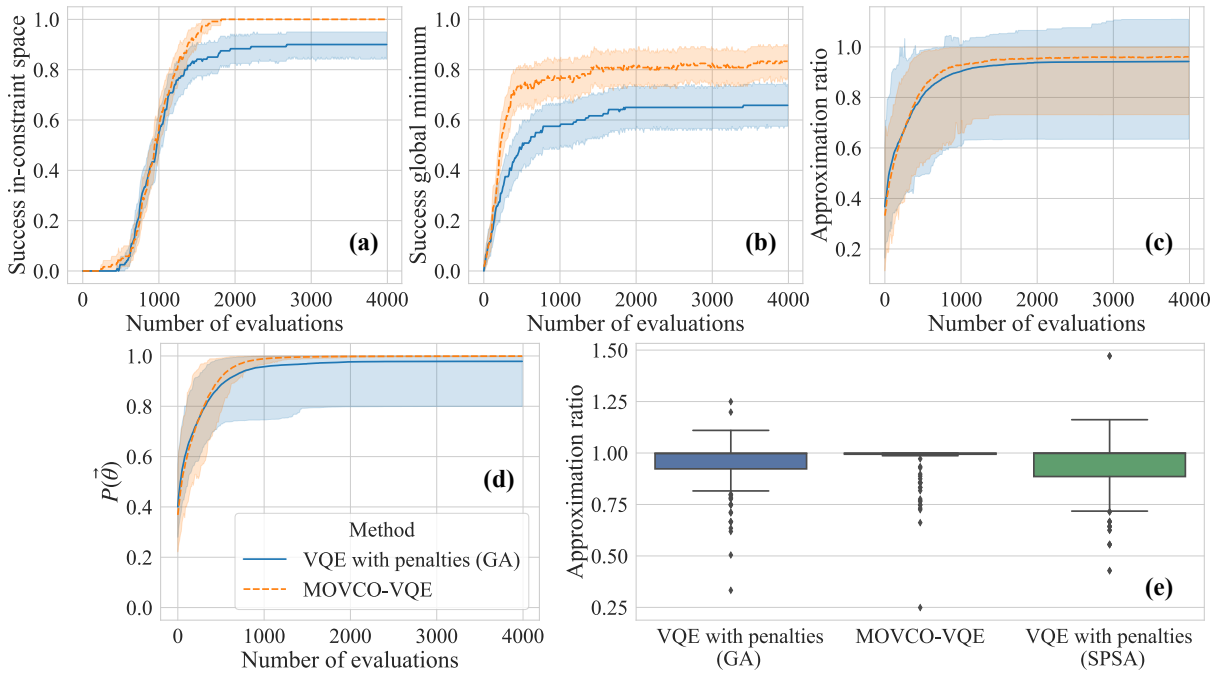


图11：MOVCO-VQE与带惩罚项的VQE性能对比。我们展示了120个CMP实例的结果（在4天的周期内有2个现金点）。在带惩罚项的VQE中，使用与MOVCO具有相同遗传算子和超参数的单目标遗传算法作为经典优化器。图（c）和（d）显示了围绕平均值的95%百分位区间，范围为分布的第2.5至第97.5百分位数。图（a）和（b）显示了平均值估计的95%置信区间。图（e）中的箱体表示下四分位数和上四分位数，须线延伸至1.5倍四分位距（彩色在线）。

如第 4 节所述，硬约束总数为 $D+1$，不包括根据问题的公式设定在任何状态下都能满足的约束（每天每台自动柜员机可用现金的上下限）。然而，对于第 5.1 节和 5.2 节中生成的小规模问题，可能不存在满足所有约束的状态。在这种情况下，会计算至少有一个解能够满足的约束最大数量，并据此计算受限能量和约束满足结果。

C 单目标遗传算法优化

观察MOVCO的性能后，人们可能会想知道，它的优势究竟是否源于约束多目标方法本身，还是仅仅是遗传算法优化机制的结果。

在这方面，在图11中，我们将其性能与一种变分量子本征求解器进行比较；在该求解器中，变分参数的经典优化由单目标遗传算法和含惩罚项的成本函数完成。如第3节所述，MOVCO-VQE 使用 NSGA-II 算法作为优化子程序。在所示数值结果中，NSGA-II 采用模拟二进制交叉和多项式变异作为遗传算子，种群规模为 10。为了公平比较这两种方法，在图11 中，带惩罚项的变分量子本征求解器使用的遗传算法与 MOVCO-VQE 中的 NSGA-II 采用相同的遗传算子和超参数；但该算法并非依据第3节介绍的两个适应度函数进行多目标优化来提升解的质量，而是仅以式(20)中定义的惩罚哈密顿量的能量作为质量指标：$\langle\Psi(\vec{\theta})|\,\hat{}C_{pen}(\,\hat{}Z)\,|\Psi(\vec{\theta})\rangle$。因此，基于支配前沿的 MOVCO 排名被基于平均惩罚能量值的评分所取代，其中 $\lambda_l=\lambda_f=25$。

我们可以看到，使用单目标遗传算法得到的现金管理问题结果与使用SPSA优化器得到的结果相似（见图8），这表明MOVCO在以下方面具有优势：

所得解的成本、结果的离散程度，尤其是收敛到满足所有硬约束的解的能力。这些实证结果表明，MOVCO的优势在于将多目标方法与遗传算法的有效探索能力相结合：多目标方法在优化过程中同时考虑解的能量和复杂约束的满足情况，而遗传算法则有效探索希尔伯特空间。

D NSGA-II 算法

本节将详细介绍非支配排序遗传算法（NSGA-II），[47],包括算法的一般步骤（算法1）、将解分类到各个前沿的方法（算法 3），以及拥挤距离的计算方法（算法 2）。

算法 1 NSGA-II 算法

输入： 种群规模 N、代数 T、两个适应度函数、遗传算子（选择、交叉、变异）

输出： 构成帕累托前沿的一组解

1: $P(0) \leftarrow$ 随机种群(N) ▷ 生成初始种群
2: 评估($P(0)$) ▷ 使用非支配排序分配适应度或等级（参见算法 3），并然后使用拥挤距离算子（参见算法2）
3: $Q \leftarrow \emptyset$ ▷ 使用空集初始化子代种群
4: **对** $t = 0$ 至 $T - 1$ **执行循环**
5: $W(t) \leftarrow$ 锦标赛选择($P(t)$) ▷ 将个体两两比较并选出获胜者
6: $M(t) \leftarrow$ SBX交叉($W(t)$) ▷ 通过创建新的解来探索搜索空间解 [54]
7: $Q(t) \leftarrow$ 多项式变异($M(t)$) 。开发 [55]
8: 评估($Q(t)$) 。评估新的种群成员
9: $P_{new}(t) \leftarrow P(t) \cup Q(t)$ 。合并种群并计算适应度
10: $P(t+1) \leftarrow$ 幸存者($P_{new}(t)$) ▷ 为下一代选择更优的 N 个解
11: **结束循环**

算法 2 拥挤距离计算 输入：构成前沿 F 的一组解，目标函数 f_m输出：根据基于密度的距离 d_i 对这些解进行排序 1: $r \leftarrow |F|$ ▷ 基数，即前沿 F 中解的数量 F 2: $M \leftarrow$ 目标函数的数量 3: 对 $i = 1$ 到 r 执行 4: $d_i \leftarrow 0$ ▷ 对于前沿 F 中的每个元素，将其距离 d_i 设为零 5: 结束循环 6: 对 $m = 0$ 到 M 执行 7: $F \leftarrow$ 排序(F, f_m) . 根据目标函数 $f_m(i)$ 的值，对前沿 F 中的元素 i 进行排序 8: $d_1 \leftarrow d_r \leftarrow \infty$ ▷ 将最极端解的距离 d 设为较大的值 9: $f_{m,max} \leftarrow \max f_m(i)$ 10: $f_{m,min} \leftarrow \min f_m(i)$ 11: 对 $i = 2$ 到 r - 1 执行 12: $d_i \leftarrow d_i + |f_m(i+1) - f_m(i-1)|/(f_{m,max} - f_{m,min})$. 将前沿 F 中每个解的距离计算为排序后前一个元素和下一个元素的目标函数 f_m 值之差的总和 13: 结束循环 14: 结束循环 . 优先保留位于较稀疏空间中的解，即拥挤距离 d 较大的解。

算法 3 前沿分类

输入： 一组解或种群 P，适应度函数

输出： 按帕累托支配关系对这些解进行排序

$F_1 \leftarrow \emptyset$ ▷ 将第一前沿初始化为空集

对于每个 p 属于 P **执行**

  $S_p \leftarrow \emptyset, \quad n_p \leftarrow 0$

  对于每个 $q \neq p$ 在 P **执行**

    如果 $p < q$ **则** 。 如果 p 根据适应度函数支配 q

      $S_p \leftarrow S_p \cup \{q\}$ 。 被 p支配的空间

    else

      如果 $q < p$，**则** ▷如果 q 支配 p

        $n_p \leftarrow n_p + 1$ ▷ 扩大支配 p 的空间

      结束条件判断

    结束条件判断

  结束循环

  if $n_p = 0$ **然后**

    $p_{rank} \leftarrow 1$

    $F_1 \leftarrow F_1 \cup \{p\}$ ▷ 第一前沿的构建

  结束 if

结束 for

$i \leftarrow 1$

当 $F_i \neq \emptyset$**时，执行**

  $F_{i+1} \leftarrow \emptyset$ ▷ 其余前沿的构造

  对每个 p 属于 F_i **执行**

    对 每个 q 属于 S_p **执行** ▷ 每个被 p支配的元素

      $n_q \leftarrow n_q$ - 1 。 支配 q 的空间减少了一个元素

      如果 $n_q = 0$ **那么** ▷ q 现在是一个非支配解，因此它属于下一个前沿

        $q_{rank} \leftarrow i + 1$

        $F_{i+1} \leftarrow F_{i+1} \cup \{q\}$

      结束 if

    结束循环

  结束循环

  $i \leftarrow i + 1$

结束 while 循环 ▷ 较低排名的解，即属于第一前沿的解，会得到提升。