2024 开源之夏 MindSpore 项目结项报告

一、项目基本信息

- 项目名称: 浅层 Clifford 线路近似求解最大割问题
- 项目编号: 24c6d0416
- 项目主导师: 罗茂林<luoml5@mail3.sysu.edu.cn>
- 项目承接人: 杨磊<yanglei@shao.ac.cn>
- 项目难度: 基础
- 涉及技术领域标签: AI
- 项目技术要求: Python>=3.7.2, mindquantum==0.9.0
- 项目产出要求:
  1、在 MindSpore Quantum 中添加对应功能模块，并且为该功能撰写教程文档；2、利用 MindSpore Quantum 成功复现算法和论文中的主要结果图（Fig1,Fig2,Fig3,Fig5,Fig6,Fig8）； 3、提交详细的技术报告：包括对原文的详细分析、复现代码的结构分析以及对结果的分析等； 4、相关评估指标符合要求，代码需要有适当的注释并通过 clean code 标准。 5、最终项目代码需要通过审核并合入 mindquantum 代码仓。
- 项目描述:
  使用变分算法求解最大割问题面临训练成本高等问题，如果可以使用 Clifford 线路模拟求解最大割问题，就可以在经典计算机上高效求解。
- 时间规划:

阶段	时间	内容
资料文献，整理消化	6.3-6.23	精读文献，梳理思路，了解背景知识
搭建线路，实现算法	6.24-8.11	实现 Adapt QAOA、Standard QAOA、Adapt Clifford、GW 算法，并成功运行实例
应用算法，产出图像	8.12-9.14	借助有限的算力资源，设置相关参数，按照原文要求运行算法，得到产出
系统总结，提交项目	9.15-9.30	提交项目，撰写文档

二、项目实施进度

1. 算法剖析

Standard QAOA

$$|\psi(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle_p = \left[\prod_{l=1}^{p} e^{-i\beta_l H_M} e^{-i\gamma_l H_C}\right] H^{\otimes N}|0\rangle^{\otimes N}$$

$$H_M = \sum_{j=1}^{N} X_j\,, \qquad H_C = \frac{1}{2}\sum_{i<j} w_{i,j} Z_i Z_j$$

其中 p 为层数，$(\boldsymbol{\gamma},\boldsymbol{\beta})$ 为线路中所有优化参数，共 $2p$ 个参数。H_C 为系统哈密顿量，Standard QAOA cost 层线路选择单比特 X 旋转门作用于所有量子比特作为 H_B。有研究已经针对特定问题和硬件架构而将 Standard QAOA cost 线路提出了修改，这些结果也揭示了 QAOA ansatz 的潜在优势，但没有提供一个适用于广泛优化问题的 H_B 的选择方案。

Adapt QAOA

$$|\psi(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle_p^{ADAPT} = \left[\prod_{l=1}^{p} e^{-i\beta_l A_l} e^{-i\gamma_l H_C}\right] H^{\otimes N}|0\rangle^{\otimes N}$$

在 Adapt QAOA 算法中，将每层相同的 H_B 替换成层层不同的一组 A_l。

①定义 mixers pool $\{A_j\}$，满足 $A_j = A_j^\dagger$，依旧选择 $\left|\psi_{(0)}\right\rangle = |+\rangle^{\otimes n}$ 作为初始态。

②制备当前的 $\left|\psi_{(l-1)}\right\rangle$, 并计算相对于 mixers pool 中每个 mixer 的能量梯度 $-i\left\langle\psi_{(l-1)}\middle|e^{iH_C\gamma_l}[H_C,A_j]e^{-iH_C\gamma_l}\middle|\psi_{(l-1)}\right\rangle$, 其中存在一个新的变量 γ_l，我们可预先定义为 $\gamma_0 = 0.01$[1]。然后得到计算值最大的 mixer 作为当前层的 mixer: A_l.

③ $(\boldsymbol{\gamma}^{(l-1)},\boldsymbol{\beta}^{(l-1)})$ 为前层的已优化参数，将 $(\boldsymbol{\gamma}^{(l-1)},\boldsymbol{\beta}^{(l-1)},0.01,0.01)$ 作为优化初始值进行优化，优化结果使期望 $\left\langle\psi_{(k)}\middle|H_C\middle|\psi_{(k)}\right\rangle$ 最小，再返回到第二步。为保证 Adapt QAOA 与 Standard QAOA 的可比性，Standard QAOA 也需要使用相同的迭代策略。

针对不同的系统也需要选择不同的池子，我们先定义线路所需要量子比特的集合 Q, Standard QAOA 所对应的池子为: $P_{QAOA} = \{\sum_{i\in Q} X_i\}$，再定性地定义两个不同的池子，其中一个包括了所有的单比特旋转: $P_{single} = \bigcup_{i\in Q}\{X_i, Y_i\} \cup \{\sum_{i\in Q} Y_i\} \cup P_{QAOA}$，另一个包括了单比特和多比特纠缠门: $P_{multi} = \bigcup_{i,j\in Q\times Q}\{B_iC_j \middle| B_i, C_j \in \{X,Y,Z\}\} \cup P_{single}$。显然 $P_{QAOA} \subset P_{single} \subset P_{multi}$，在 Adapt QAOA 中我们选择这样的一个池子:

$$Pop = \{\sum_{i\in Q} X_i\,, \sum_{i\in Q} Y_i\} \bigcup \{X_i, Y_i\}_{i=1,\dots,n} \bigcup \{X_iX_j, Y_iY_j, Y_iZ_j, Z_iY_j\}_{i,j=1,\dots,n,i\neq j}$$

[1] *L. Zhu, H. L. Tang, G. S. Barron, F. A. Calderon-Vargas, N. J. Mayhall, E. Barnes, and S. E. Economou, Phys. Rev. Res.* ***4****, 033029(2022).*

Adapt Clifford

$$|\boldsymbol{\Psi}\rangle = \left[\prod_{r=2}^{N-1} e^{i\frac{\pi}{4}Z_{\boldsymbol{a}(r-1)}Y_{\boldsymbol{b}(r-1)}}\right] e^{i\frac{\pi}{4}Y_k Z_j} \underline{Z_k} H^{\otimes N}|0\rangle^{\otimes N}$$

其中 $\boldsymbol{a}^{(r)}, \boldsymbol{b}^{(r)}$ 分别为“激活比特”和“待激活比特”的集合，“激活”指的已被作用后的比特。k 为随机初始值，旨在作用 Z 门于一个随机比特。

（特注：公式中 ZY 的前后顺序并不决定作用的是 Rzy 还是 Ryz 门）

$\boldsymbol{r=0}$:

$$|\psi_0\rangle = Z_k H^{\otimes N}|0\rangle^{\otimes N}$$

$$\boldsymbol{a}^{(0)} = \{k\}, \boldsymbol{b}^{(0)} = \{1, \dots, N\}/\{k\}$$

第 0 步可被认作初始化。

$\boldsymbol{r=1}$:

$$|\psi_1\rangle = e^{i\frac{\pi}{4}Y_k Z_j}|\psi_0\rangle$$

$$\boldsymbol{a}^{(1)} = \{k, j\}, \boldsymbol{b}^{(1)} = \{1, \dots, N\}/\{k, j\}$$

j 值的确定方法：定义一组 $mixers = \left[Y_{\boldsymbol{a}^{(0)}} Z_{\boldsymbol{b}^{(0)}}\right]$，计算得到 $Y_k Z_j = \max_{M \epsilon mixers} [-i\langle\psi_0|[H_C, M]|\psi_0\rangle]$, 第 j 比特已被作用，随后添加至 $\boldsymbol{a}$ 并在 $\boldsymbol{b}$ 中删除。

$\boldsymbol{r \geq 2}$:

$$|\psi_2\rangle = e^{i\frac{\pi}{4}Z_x Y_y}|\psi_1\rangle$$

$$\boldsymbol{a}^{(2)} = \{k, j, y\}, \boldsymbol{b}^{(2)} = \{1, \dots, N\}/\{k, j, y\}$$

此后的定义不同于$r=1$，$mixers = \left[Z_{\boldsymbol{a}^{(1)}} Y_{\boldsymbol{b}^{(1)}}\right]$, 并重复上述计算过程。以此类推，直到$N-1$层填完，也就是 $\boldsymbol{b}$ 为空集。

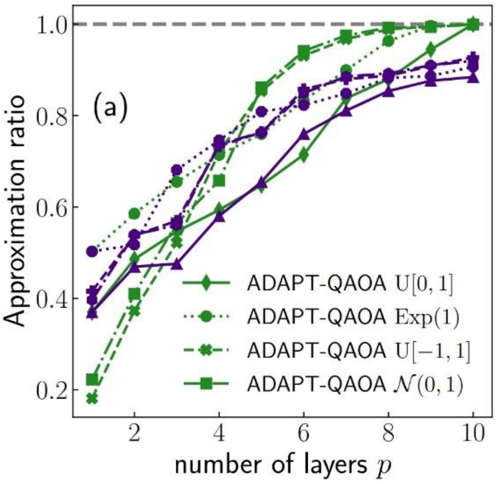
确定作用的双比特旋转门：

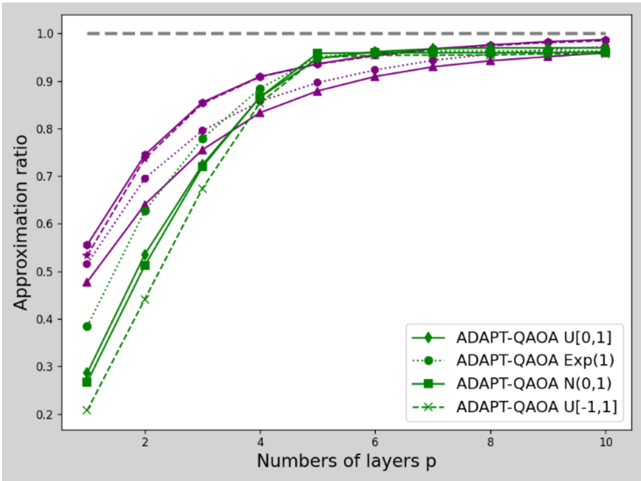
$$e^{i\frac{\pi}{4}Y_l Z_m} = \begin{cases} e^{i\frac{\pi}{4}Z\otimes Y}, & m < l \\ e^{i\frac{\pi}{4}Y\otimes Z}, & l < m \end{cases}$$

由于 $r=0$ 时的 k 为随机值，必然会产生随机误差，所以又分为 randomized 和 deterministic 方法。randomized 为随机一个 k 值；deterministic 为遍历所有 k 可能值，最后取最小的期望值。

2. 原文复现（※表示当前问题）

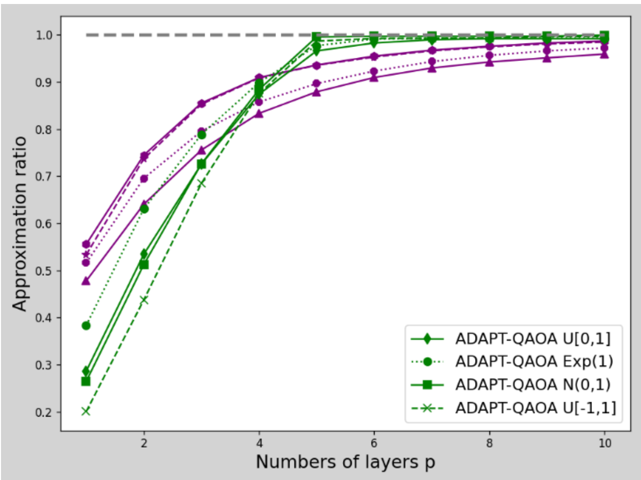
Figure 1 a





原文取 50 个节点数为 6、四种权重分布的实例，对其分别用 Adapt QAOA 和 Standard QAOA 求解，最终得到解与精确解的近似比。右图是我们随机取 instances=50 的结果，由此可见，在 layer>5 的时候 Adapt QAOA 都能够收敛，但是收敛值并不满足原文中几近 1 的结果，甚至比相同条件下的 Standard QAOA 表现更差。

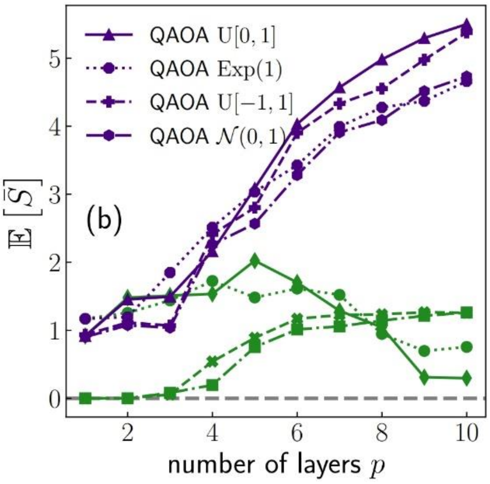
※ 为了探究这个问题，下图列出了 Adapt QAOA 求解 uniform(0,1) 分布 MaxCut 问题的部分结果数据，可以看出：在优化这一过程中，会以一定概率让结果陷入局部解的情况，并且这个局部解与精确解的偏差无法忽略地大，从而影响了结果图的平均收敛值。针对这个局部解的问题，目前尚未得到有效的解决方案，但能够断定的是这并非算法本身的原理问题、逻辑问题，而是优化过程的数据问题。如果将结果数据中小于 0.90 近似比的实例剔除，就能得到与原文相符的趋势。最后从数据的规律来看，可以得出结论： Standard QAOA 的解更稳定，更平均；Adapt QAOA 的解上限更高，下限也更低，表现不稳定。结论适用 minimize 中各种优化方法。

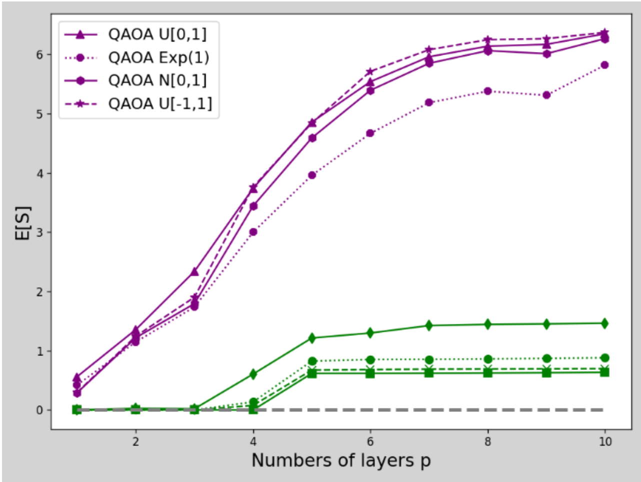


5	6	7	8	9	10
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0.763792	0.763792	0.763792	0.763792	0.763792	0.763792
1	1	1	1	1	1
1	1	1	1	1	1
0.896708	1	1	1	1	1
0.879672	0.933759	0.933759	0.933759	0.933759	0.933759
0.990672	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0.788458	0.788458	0.788458	0.788458	0.788458	0.788458
1	1	1	1	1	1
0.922121	0.922121	0.922121	0.922121	0.922121	0.922121
0.923377	0.961832	1	1	1	1
1	1	1	1	1	1
0.953364	1	1	1	1	1
1	1	1	1	1	1
0.824136	0.824136	0.824136	0.824136	0.824136	0.824136

Figure 1 b

为了衡量含参线路与 Clifford 线路的近似程度，定义线路的二阶 Renyi 熵的平均值为衡量标准。结果越接近于 0，则说明越近似于 Clifford 线路。

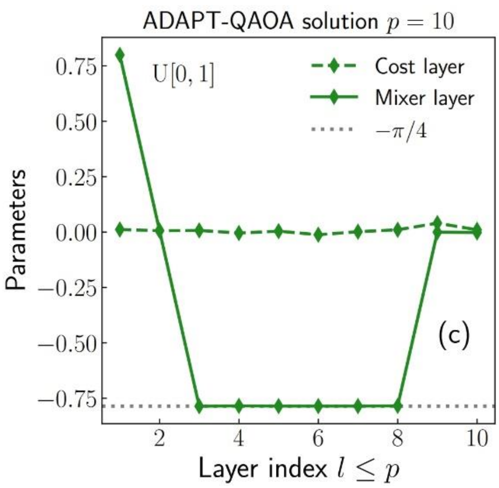


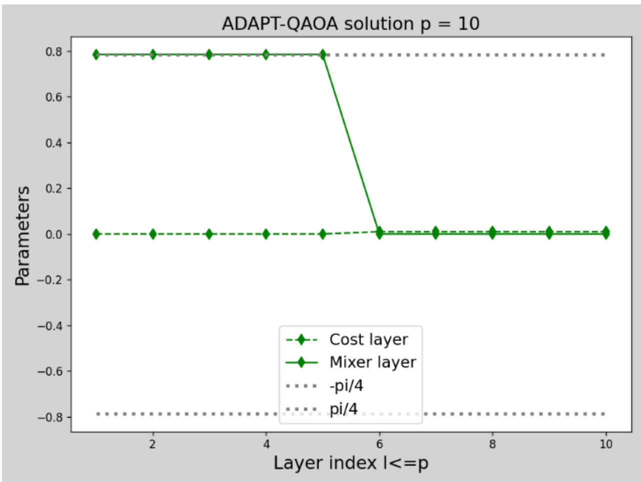


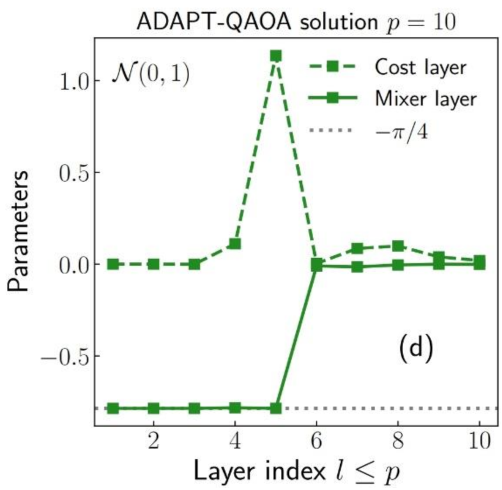
原文与我们都取 instances=50 的结果。由图可知，Adapt QAOA 相较于 Standard QAOA 更加容易转换成 Clifford 线路，但在 layer>6 时，原文中 U[0,1] 和 Exp[1] 两种分布方式会有明显的下降趋势，我们并没有得到这样的趋势结果，导致这个结果我们依然认为是局部解的原因。

原文中的说明: 即便发现线路与 Clifford 线路有较大差距，但能够说明出在小规模的最大割问题中，Adapt QAOA 与 Standard QAOA 有鲜明的对比。

Figure 1 c&d







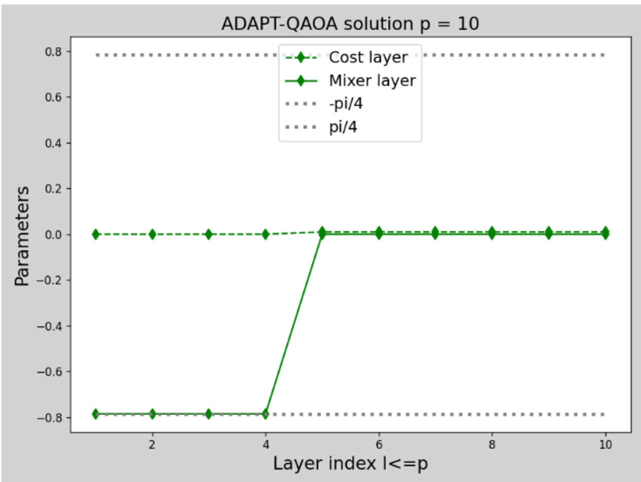


图 c&d 取自 U[0,1] 和 N(0,1) 分布一个实例中的参数分布 $(\boldsymbol{\gamma},\boldsymbol{\beta})$，原文中的总结: (1)每层的 mixer 部分都满足 Clifford 参数，也就是 0 或 $-\pi/4$，并且大多数 mixer = $Y_l Z_m$。(2)大部分层数的 cost 部分，其参数基本都为 0。(3)仅需要 N=nodes 层的旋转就能够得到收敛的近似解。与我们得到的结论基本相符。

※ 但是原文只取一个实例不具备普遍性和说服力，所以我们对产出过程进行总结。公式 $D(\boldsymbol{v}) = \sum_{v_i} \frac{min_{l\epsilon\mathbb{Z}}\left\|v_i - l\frac{\pi}{4}\right\|}{\pi/8}$ 表示所得参数与 Clifford 参数的相近程度，越接近于 0 则说明越接近 Clifford 参数。我们取 instances=50 再取平均。

		Uniform[0,1]	**Normal(0,1)**
Adapt QAOA	From original	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 0.522 \pm 0.270$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 1.510 \pm 0.720$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.247 \pm 0.322$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.830 \pm 0.960$
	From us	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 0.939 \pm 0.886$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 0.318 \pm 0.447$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.399 \pm 0.789$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.029 \pm 0.199$
Standard QAOA	From original	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 3.530 \pm 0.940$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 2.770 \pm 0.780$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 3.410 \pm 0.322$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 2.750 \pm 0.640$
	From us	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 5.027 \pm 0.924$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 4.693 \pm 1.139$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 5.916 \pm 0.780$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 5.910 \pm 0.795$

由表可知，所得结果与原文相符。，我们还对剩余两种分布 Exp(1) 和 U[-1,1] 也做了计算。

		Exponential(1)	**Uniform[-1,1]**
Adapt QAOA	From us	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 0.588 \pm 0.703$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 0.368 \pm 0.624$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.144 \pm 0.565$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 0.051 \pm 0.226$
Standard QAOA	From us	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 5.471 \pm 1.489$	$\mathbb{E}[D(\boldsymbol{\gamma}^*)] = 4.857 \pm 0.872$
		$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 5.961 \pm 0.961$	$\mathbb{E}[D(\boldsymbol{\beta}^*)] = 5.806 \pm 0.709$

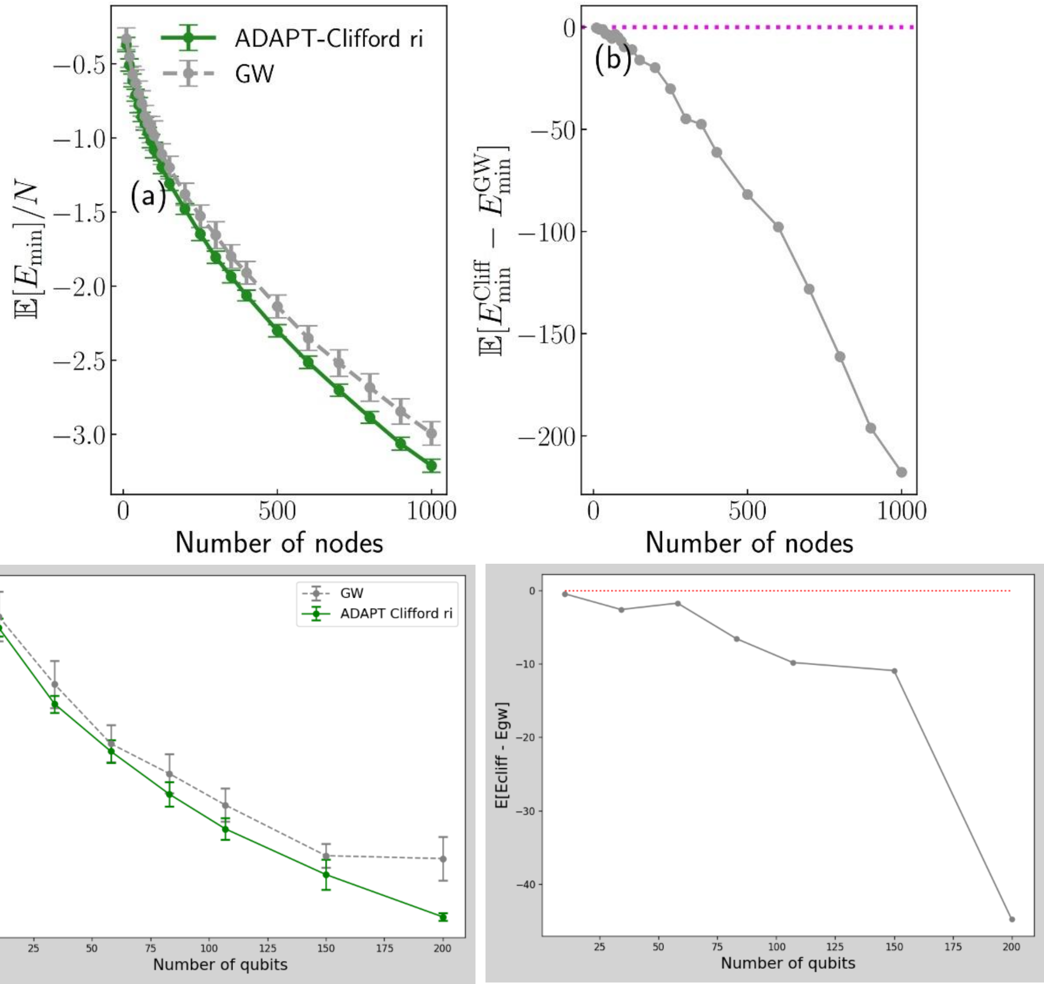
由表可知，所得结果满足规律，非常近似于 Clifford 参数。

如果我们定义：一共 20 个参数，当有>=18 个位于区间$\left(-\frac{\pi}{4} \pm 0.05,\ 0 \pm 0.05,\ \frac{\pi}{4} \pm 0.05\right)$则称之为良图。像这样的良品率我们也做了对比：

	U[0,1]	**Exp(1)**	**N(0,1)**	**U[-1,1]**
良品率	60.24%	87.72%	96.15%	100.0%

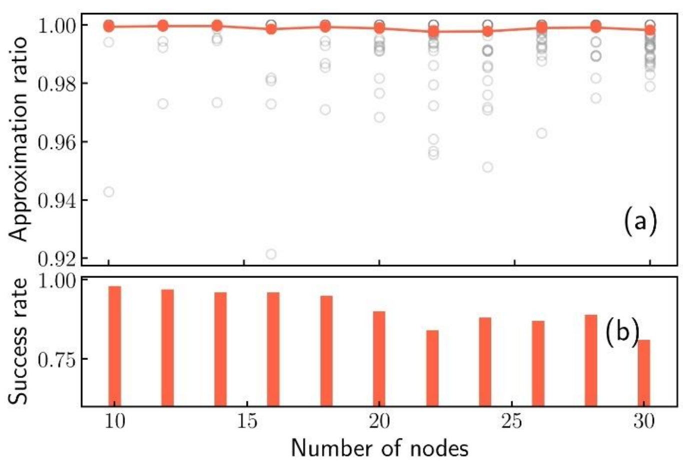
可以看出：Adapt-QAOA 算法解决以上四种权值分布的 MaxCut 问题时候所得到的含参线路，将其转化成 Clifford 线路都不难实现。

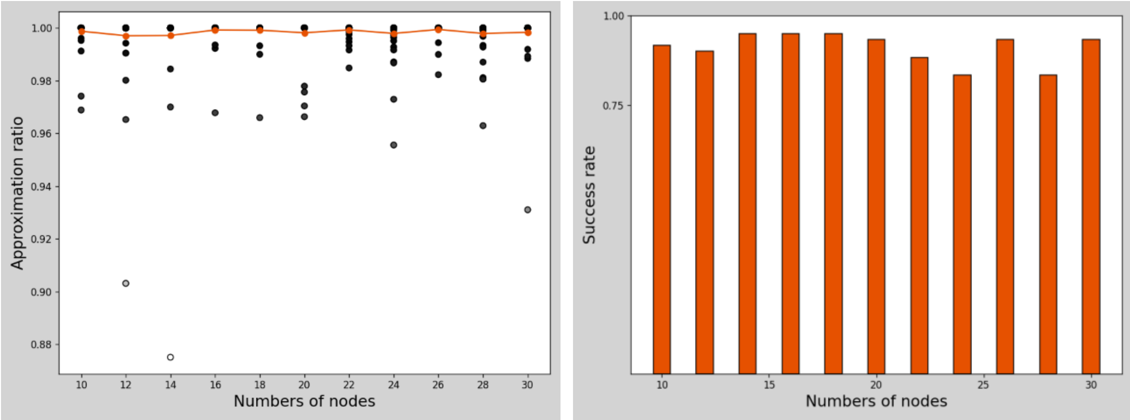
Figure 2

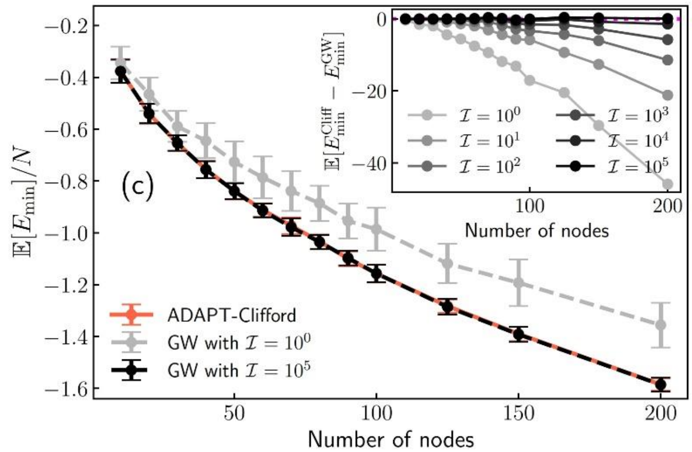


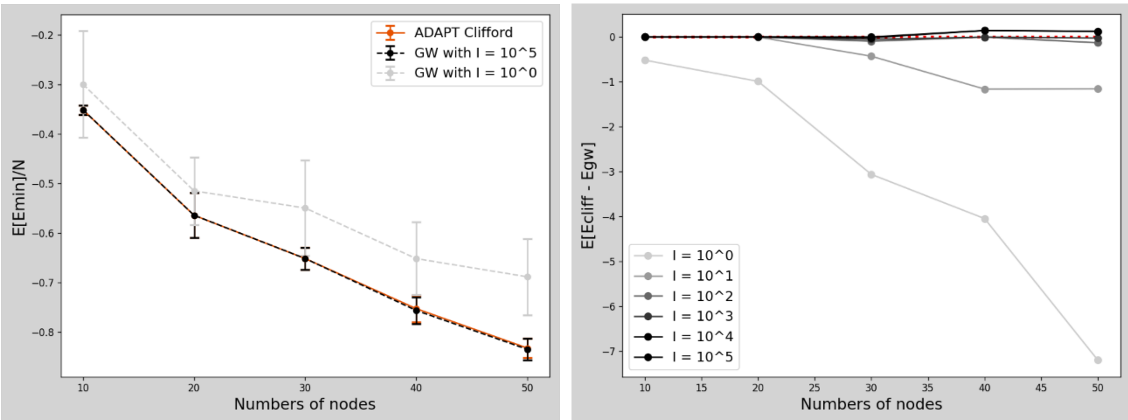
为了将 randomized 方法的 Adapt Clifford 与 standard GW 进行比较，原文取 instances=100 和 nodes(max)=1000，考虑时间成本和算力资源，我们取 instances=20 和 nodes(max)=200，并且是针对于 U[0,1] 分布的 complete 图。由图可知，randomized 方法的 Adapt Clifford 总是优于 standard GW，与原文结论相同。

Figure 3





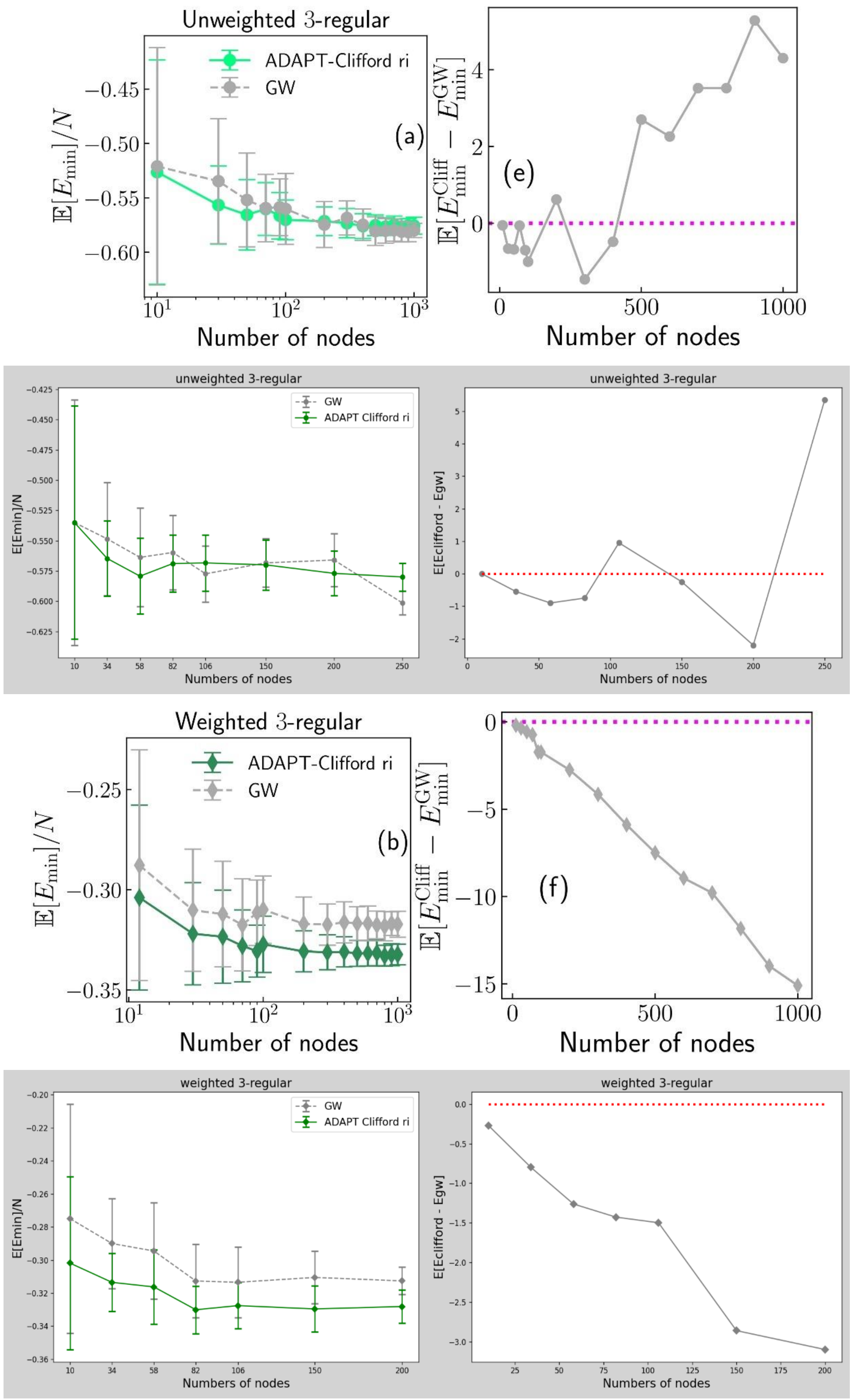




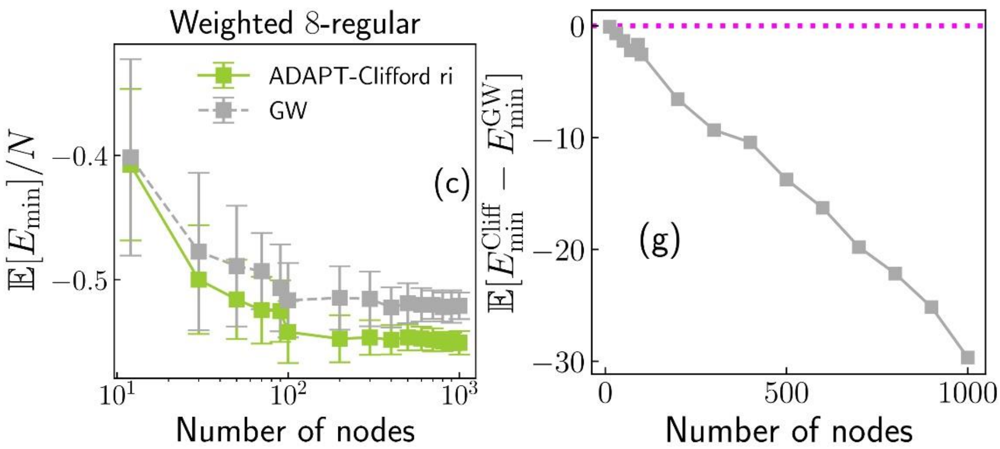
专注于 deterministic 方法的 Adapt Clifford 和 U[0,1] 分布的 complete 图。图 a&b 原文取 instances=100，我们取 instances=60。原文描述: 平均近似比为 0.997 (我们的结果为 0.9984)，并且在 N=30 时有大约 80% 的成功率。由图可知，结论与原文相符。

图 c 是将 deterministic 方法的 Adapt Clifford 与不同循环次数的 GW 相比较。原文取 instances=60，我们取 instances=5。原文得到的平均近似比为 0.9686，我们的结果为 0.9988。由图可知，结论与原文相符。

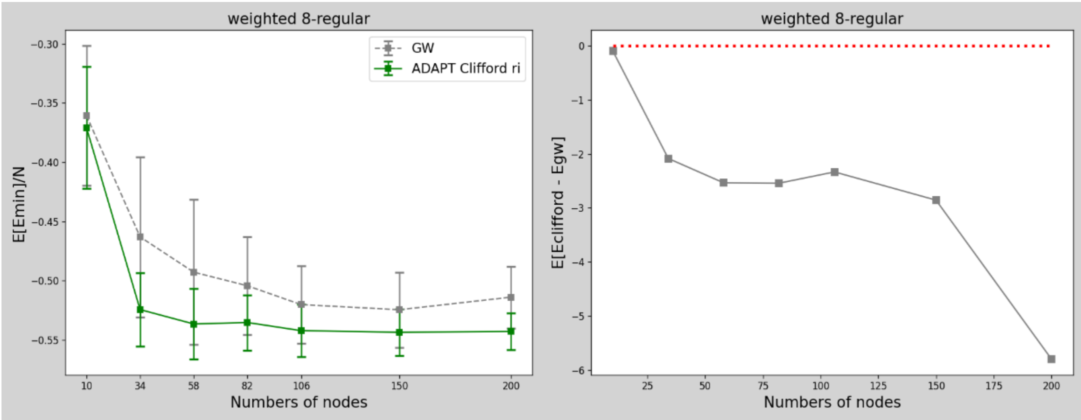
Figure 5



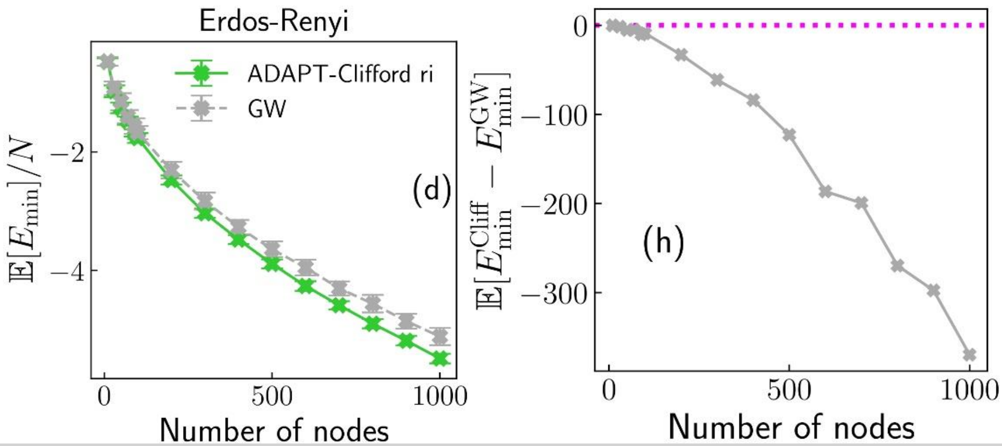
Unweighted 3-regular
ADAPT-Clifford ri
GW
(a)
$\mathbb{E}[E_{\min}]/N$
Number of nodes
(e)
$\mathbb{E}[E_{\min}^{\mathrm{Cliff}} - E_{\min}^{\mathrm{GW}}]$
Number of nodes
unweighted 3-regular
GW
ADAPT Clifford ri
E[Emin]/N
Numbers of nodes
unweighted 3-regular
E[Eclifford - Egw]
Numbers of nodes
Weighted 3-regular
ADAPT-Clifford ri
GW
(b)
$\mathbb{E}[E_{\min}]/N$
Number of nodes
(f)
$\mathbb{E}[E_{\min}^{\mathrm{Cliff}} - E_{\min}^{\mathrm{GW}}]$
Number of nodes
weighted 3-regular
GW
ADAPT Clifford ri
E[Emin]/N
Numbers of nodes
weighted 3-regular
E[Eclifford - Egw]
Numbers of nodes



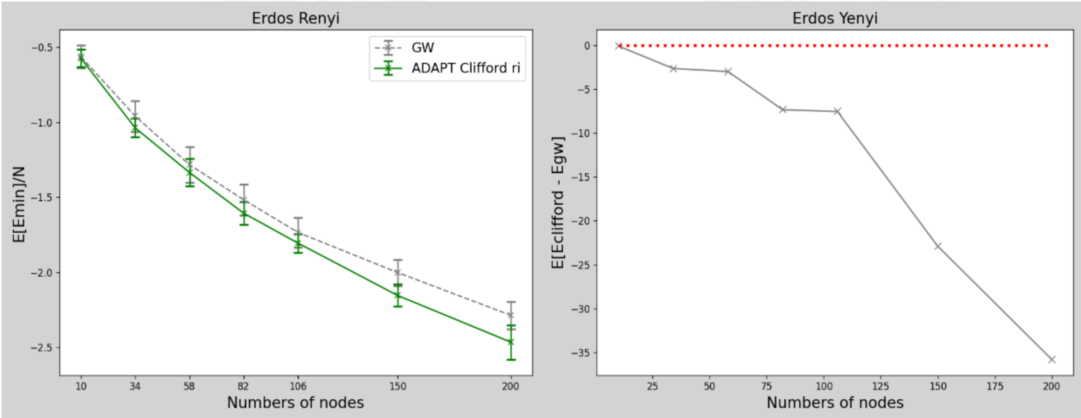
Weighted 8-regular
ADAPT-Clifford ri
GW
(c)
$\mathbb{E}[E_{\min}]/N$
Number of nodes
(g)
$\mathbb{E}[E_{\min}^{\mathrm{Cliff}} - E_{\min}^{\mathrm{GW}}]$
Number of nodes



weighted 8-regular
GW
ADAPT Clifford ri
E[Emin]/N
Numbers of nodes
weighted 8-regular
E[Eclifford - Egw]
Numbers of nodes



Erdos-Renyi
ADAPT-Clifford ri
GW
(d)
$\mathbb{E}[E_{\min}]/N$
Number of nodes
(h)
$\mathbb{E}[E_{\min}^{\mathrm{Cliff}} - E_{\min}^{\mathrm{GW}}]$
Number of nodes



Erdos Renyi
GW
ADAPT Clifford ri
E[Emin]/N
Numbers of nodes
Erdos Yenyi
E[Eclifford - Egw]
Numbers of nodes

图 5 是将 randomized 方法的 Adapt Clifford 与 standard GW 进行比较，分别取 Unweighted 3-regular、Weighted 3-regular、Weighted 8-regular 和 Erdos-Renyi 四种图进行求解，其中含权重的都为 U[0,1] 分布，Erdos-Renyi 的 edge 概率为 0.5。原文取 instances=100，nodes(max)=1000，我们取 instances=20，nodes(max)=200。由图可知，对于 N>200 的 Unweighted 3-regular 图，standard GW 有更优解。但随着 edge 的数量增多，权重的概率分布更复杂，Adapt Clifford 会有更优解，Erdos-Renyi 图也是如此。我们的结果与原文相符。

Figure 6

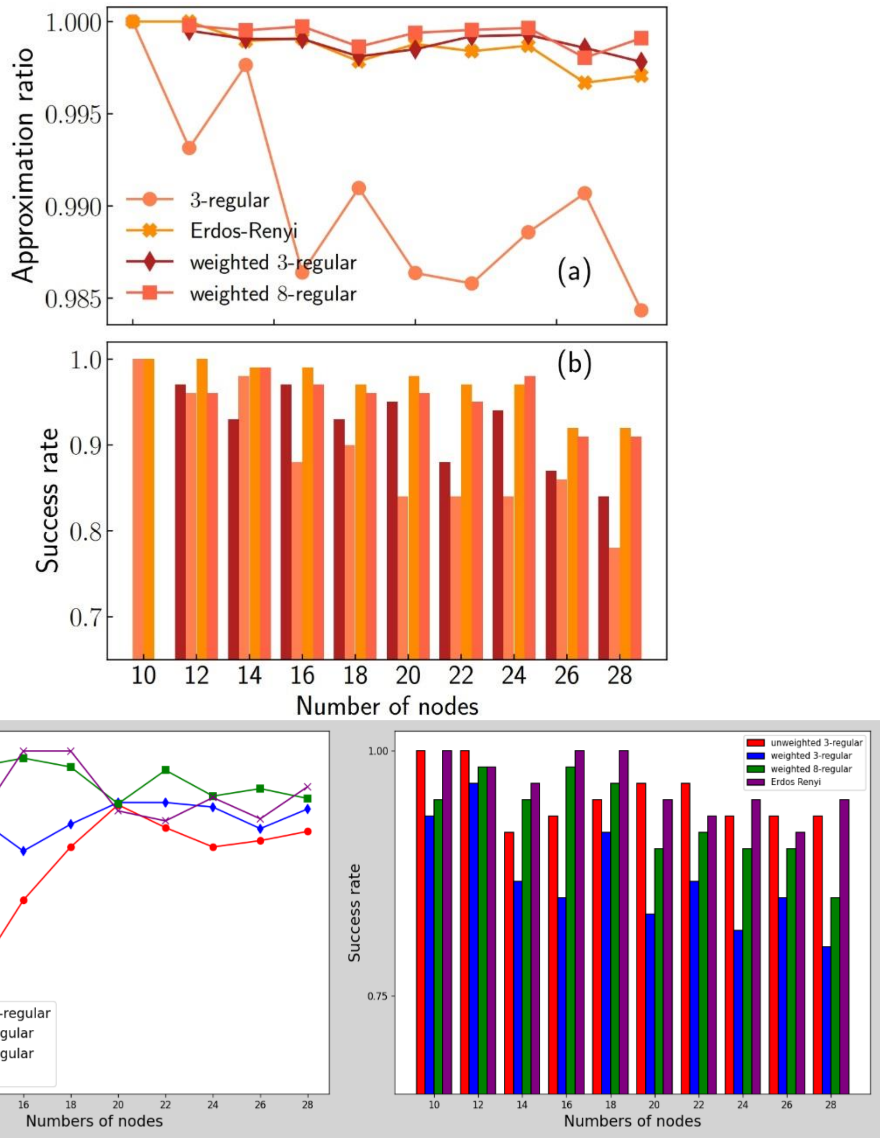
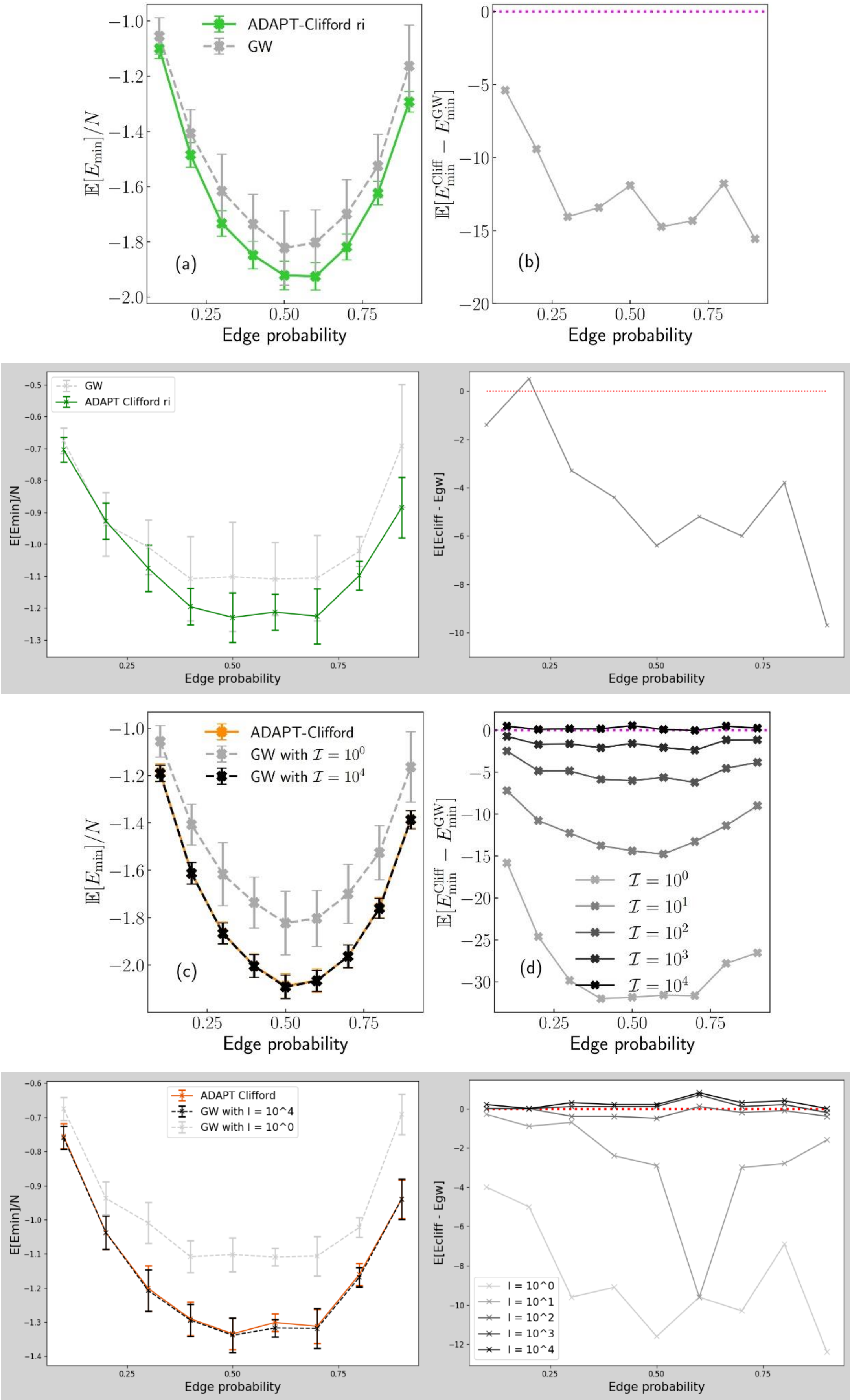


图 6 是 deterministic 方法的 Adapt Clifford 针对图 5 的四种图的求解。原文取 instances=100，我们取 instances=20。原文描述: 从 Unweighted 3-regular 到 Weighted 3-regular 再到 Weighted 8-regular，近似比得到逐步改善。3-regular 图的成功率随着节点数增加而减小，对比之下，Weighted 8-regular 的成功率却能在 N=28 时达到近 90%。考虑到随机误差，我们的结果与原文基本符合。

Figure 8


ADAPT-Clifford ri
GW
(a)
(b)
$\mathbb{E}[E_{\min}]/N$
$\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$
Edge probability
GW
ADAPT Clifford ri
E[Emin]/N
E[Ecliff - Egw]
ADAPT-Clifford
GW with $\mathcal{I} = 10^0$
GW with $\mathcal{I} = 10^4$
(c)
(d)
$\mathcal{I} = 10^0$
$\mathcal{I} = 10^1$
$\mathcal{I} = 10^2$
$\mathcal{I} = 10^3$
$\mathcal{I} = 10^4$
ADAPT Clifford
GW with I = 10^4
GW with I = 10^0
I = 10^0
I = 10^1
I = 10^2
I = 10^3
I = 10^4

为了单独研究 Erdos-Renyi 图，分别用 randomized 和 deterministic 方法进行求解。原文取 N=120，instances=100，考虑到时间成本与计算资源，N=12 0 的 deterministic 方法无法实现，所以我们取 N=50，instances=10。

图 a&b 为 randomized 方法与 standard GW 的对比，由图可知，此方法下的 Adapt Clifford 总是会优于 GW。结论与原文相符。

图 c&d 为 deterministic 方法与 不同循环次数的 GW 的对比。原文描述: 此方法下的 Adapt Clifford 总是会优于 GW，仅且当 $I = 10^4$ 时，GW 才能与 Adapt Clifford 有微乎其微的差距。由图可知，结论与原文相符。

3. 问题总结

目前为止，综上为项目要求所有需要复现的图，总体来看，我们的结果与原文基本一致，较为成功。但依旧遇到暂时无法查明的问题，问题集中于 Adapt QAOA 中，也就是 Figure 1 中。

问题一:

在获取 Adapt QAOA 求解线路的所有参数时，也就是 Figure 1 c&d 过程中。从四种权值分布的良品率来看，也直接反应了不同 MaxCut 问题对局部解的敏感程度，其中 uniform[0,1] 最容易产生局部解。

问题二:

在应用 Adapt Clifford 算法时，由于执行过程中有限的时间成本以及不足的算力资源，在确保能够验证原有规律的情况下，对部分图像的 nodes 和 instances 有所缩减。所得所有数据均为计算机计算的实际结果，无人为编造。

问题三:

在编写 Adapt Clifford 算法过程中，由于项目要求使用的 mindquantum =0.9.0 以及 0.10.0 中的 stabilizer 模拟器功能尚不完善，部分功能无法直接实现，所以本次项目沿用的是原文中所用的 stim 工具。线路实际为 stim 的 clifford 线路，使用 stim 的 peek_observable_expectation 接口对线路求期望。但是 stim 仅支持单个算符组成的哈密顿量进行求期望，所以本项目使用 mindquantum 的 commutator 求对易式，然后将结果 split() 成数个部分逐个求期望，最后求和。

4. 后续计划

对算法代码进行优化简化，使其满足 clean code 标准。根据提交结果的审核意见进一步修改。

根据 Adapt 算法求解最大割问题的启发，研究各类算法的区别并比较。