



量子计算理论基础与软件系统

量子金融



01 量子金融产业现状

02 投资组合优化

03 欧式看涨期权定价

04 编程实践



大模型



大数据



频繁交易



大模型



大数据



频繁交易

量子金融：用量子算力快速解决金融领域大规模计算。



1. 关于市场风险敏感度分析的量子算法
2. 战略资产配置的量子算法
3. 开源库: pygrnd



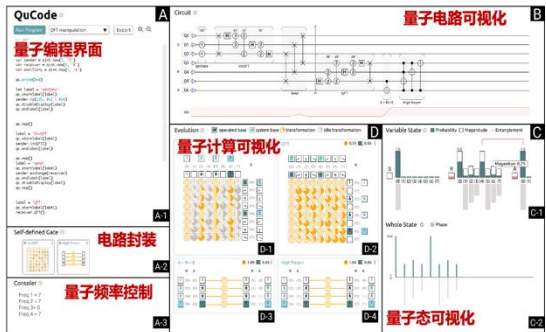
1. 投资组合优化
2. 加密货币
3. 可视化量子金融工具



金融企业	合作公司	研究方向
摩根大通	IBM	投资组合优化、期权定价
高盛集团	QC Ware	资产定价
丰业银行	Xanadu	交易产品组 合定价
BBVA	Zapata Computing、Multiverse	投资组合优化、信用评分、 套利
.....

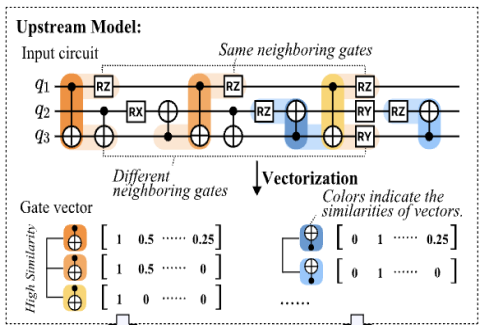
量子计算机作为新兴计算设备，研发软硬件全栈式的量子计算技术是实现弯道超车的重要方向。

云平台编程界面



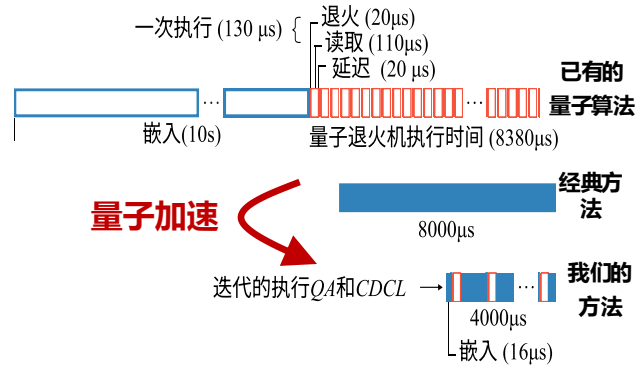
- 自主量子编程语言
- **可视化编程**环境
- 量子编译器、量子程序纠错等**高性能量子软件**

编译优化



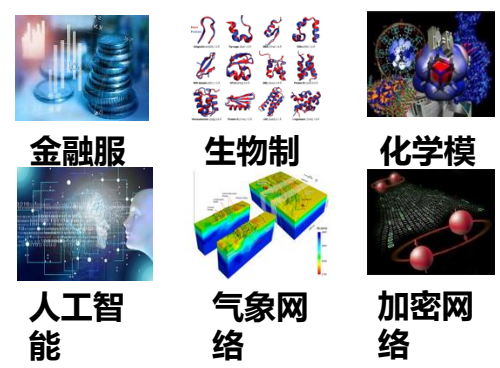
- 向量化的量子电路特征提取
- 保真度优化技术
- 提升整体编译速度
- **可扩展至上千比特编译**

可满足性问题加速



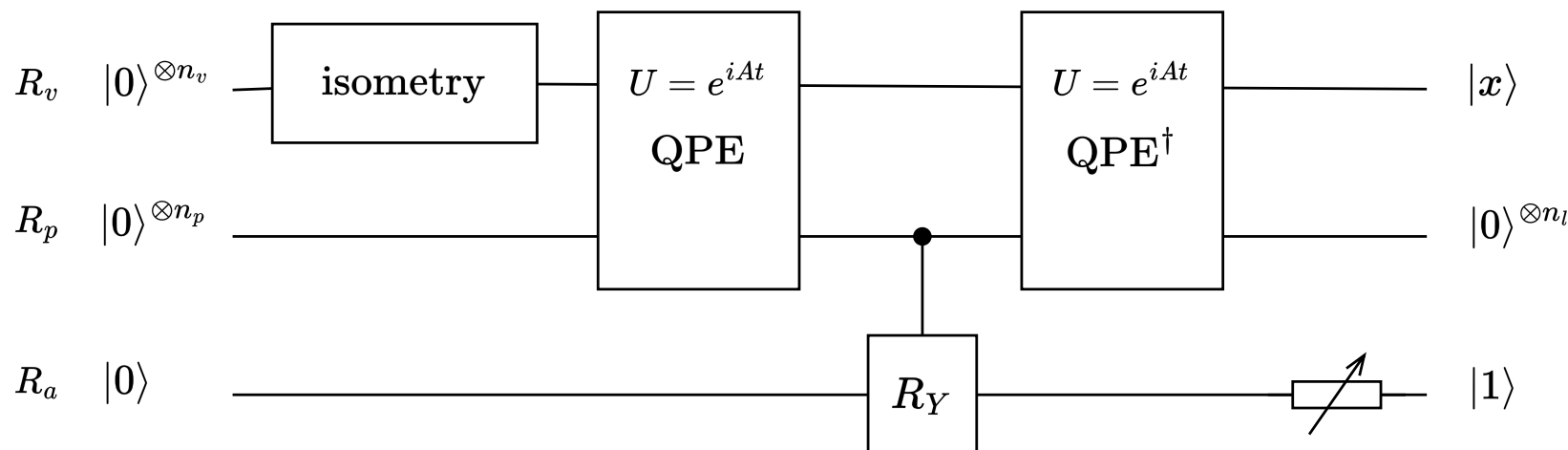
- 面向可满足性问题SAT
- 提出经典-量子混合算法
- 量子软硬件协同优化技术
- **首次实现端到端的量子加速**

量子交叉应用



- 量子金融算法
- 量子生物制药与化学模拟
- 量子机器学习与人工智能
- **“量子+X”交叉应用探索**

相关研究引领了国际量子计算机软硬件设计、系统架构和算法开发的研究热潮，目前太元一号已有微软、悉尼大学、华为、腾讯等海内外著名高校和龙头企业注册使用，学术成果发表在计算机体系结构顶尖会议/期刊 ISCA、MICRO、HPCA、TC、TPDS上。



- 提出基于输入矩阵特征值优化HHL算法参数取值的方法；
- 将投资组合优化问题转化为线性方程组，并引入约束缩放系数，降低输入矩阵条件数以减小电路规模。



2021年4月19日，由本源量子与中国科学技术大学国际金融研究院、中国建设银行联合主办，合肥大数据资产运营有限公司协办的**量子金融行业应用生态联盟签约暨首届量子金融创新赛道班开营仪式**在合肥高新区成功举办。

相关金融产品



量子期权定价

基于量子蒙特卡罗的期权定价算法

前往体验 →



期权策略收益期望应用

市场方向和预期波动观点的常见期权策略收益期望

前往体验 →



VaR 值应用

基于量子蒙特卡罗的风险价值计算算法

前往体验 →



量子贝叶斯网络应用

反映事件的多种重要非独立因变量之间的关联

前往体验 →

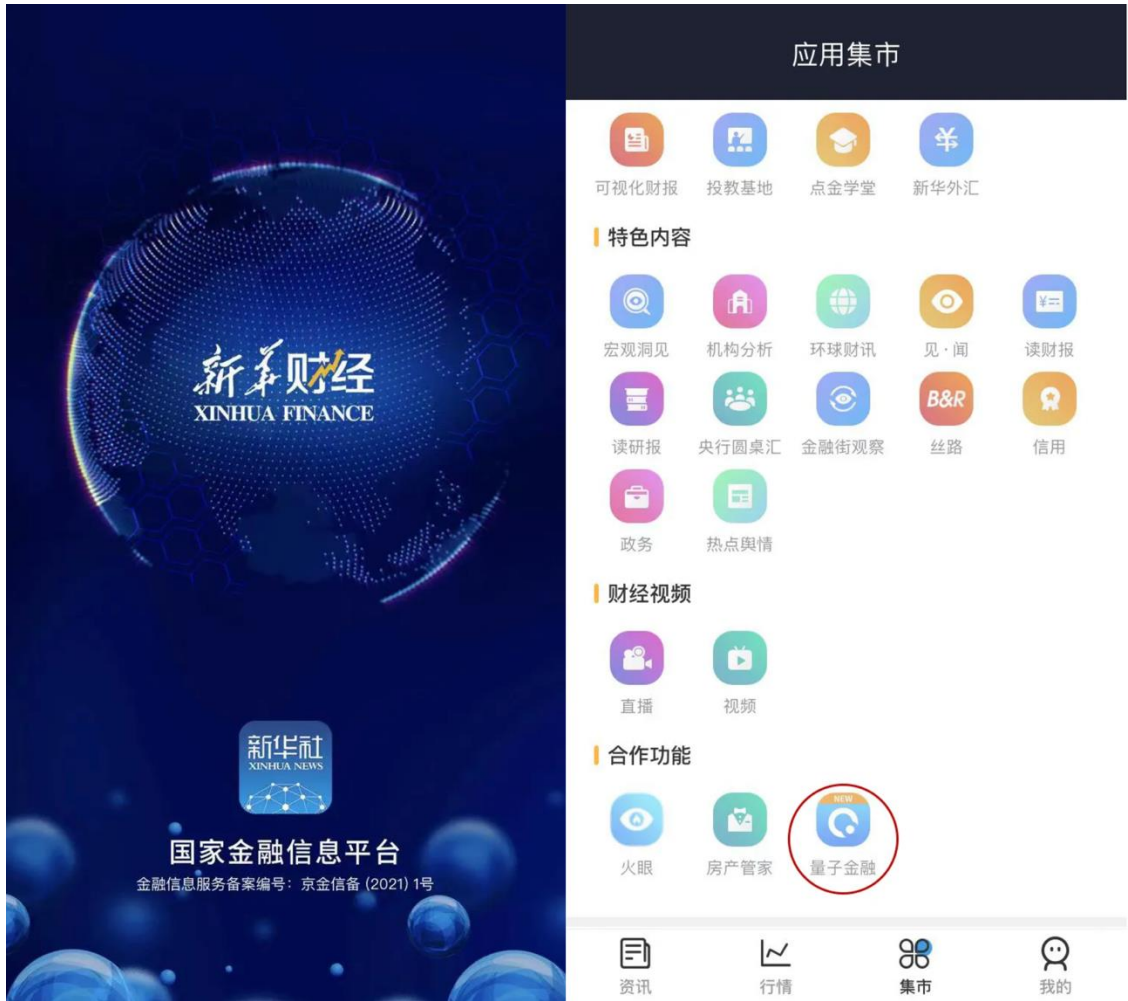


投资组合优化

基于量子算法寻找最优投资组合

前往体验 →

本源量子金融产品
体验，提供API。



2021年1月， 由本源量子联合新华社旗下中国经 济信息社新华财经共同发布的“量子金融应用” 正式在新华财经APP上线。

qiskit-community/qiskit-finance

Quantum Finance



支持**投资组合优化**、**期权定价**

28
Contributors

32
Used by

165
Stars

115
Forks



银行存贷款

- 信用评级
- 异常交易识别

证券交易

- 选股策略
- 股票预测

衍生品

- 期权定价



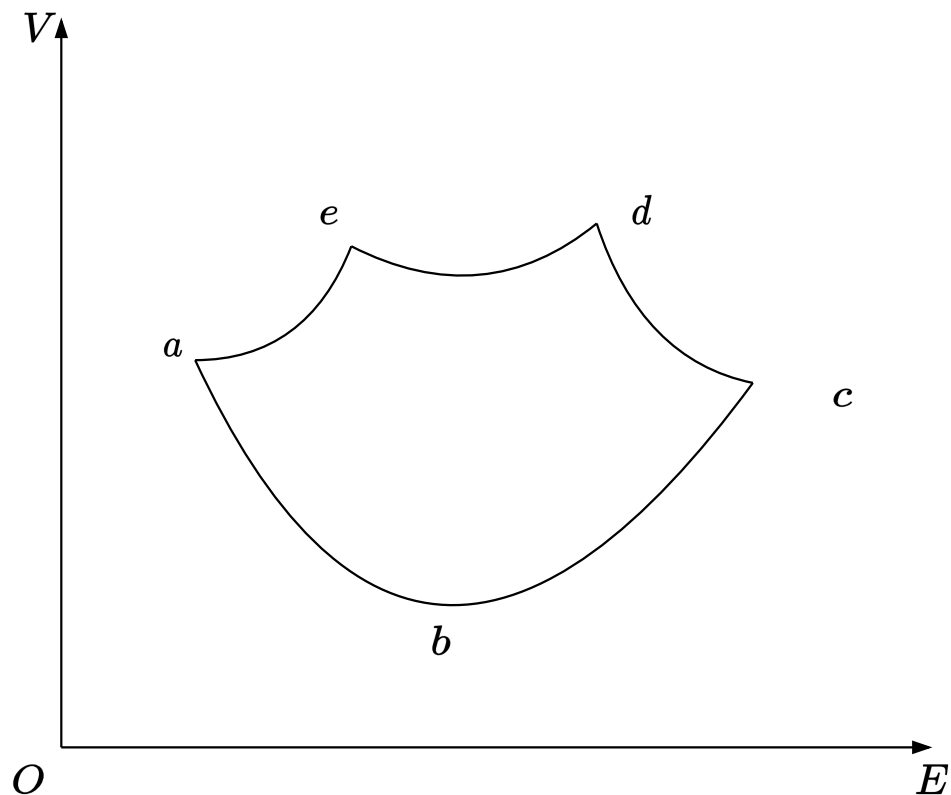
01 量子金融产业现状

02 投资组合优化

03 欧式看涨期权定价

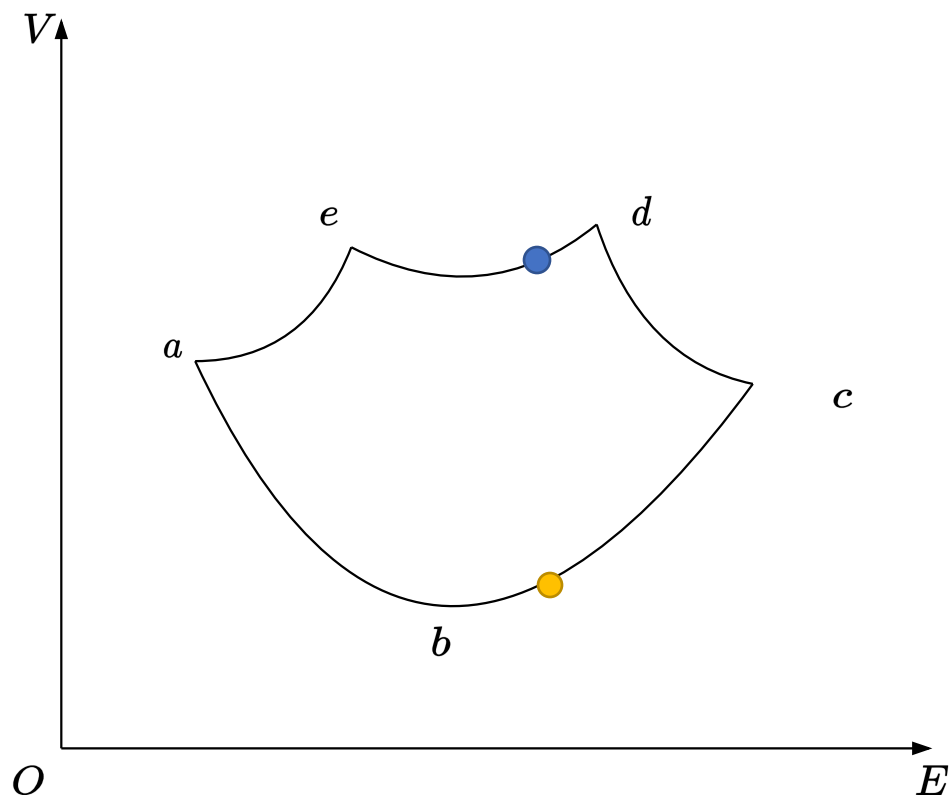
04 编程实践

投资组合优化问题建模



E 代表收益, V 代表风险。

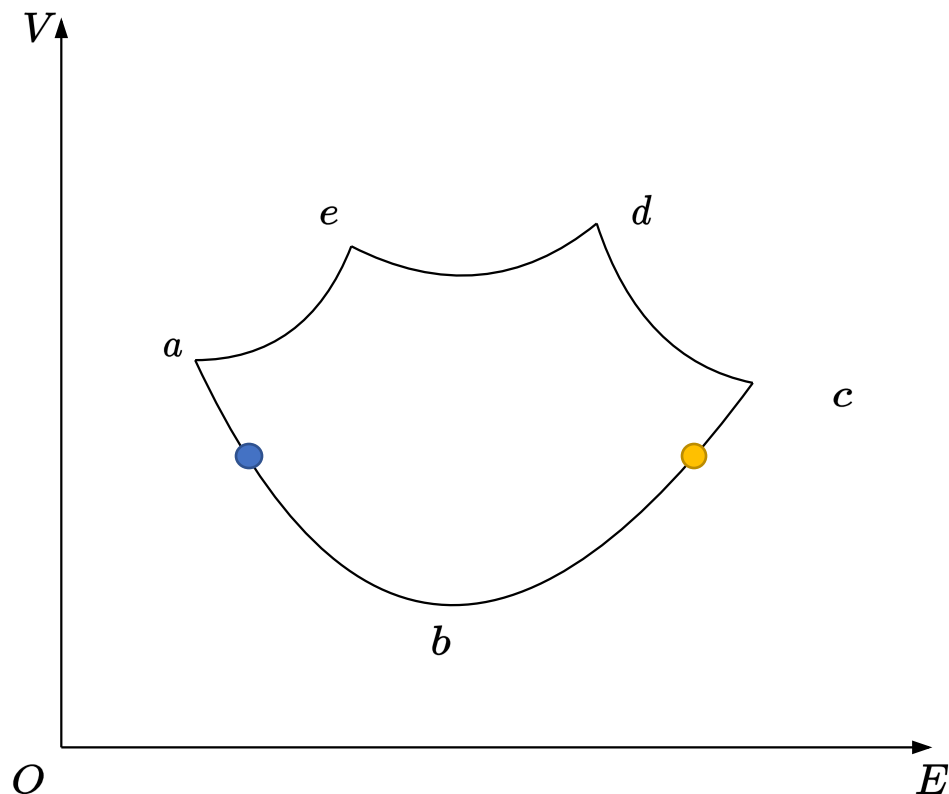
投资组合优化问题建模



E 代表收益, V 代表风险。

某一些位置, 同样的收益,
会有风险更低的点。

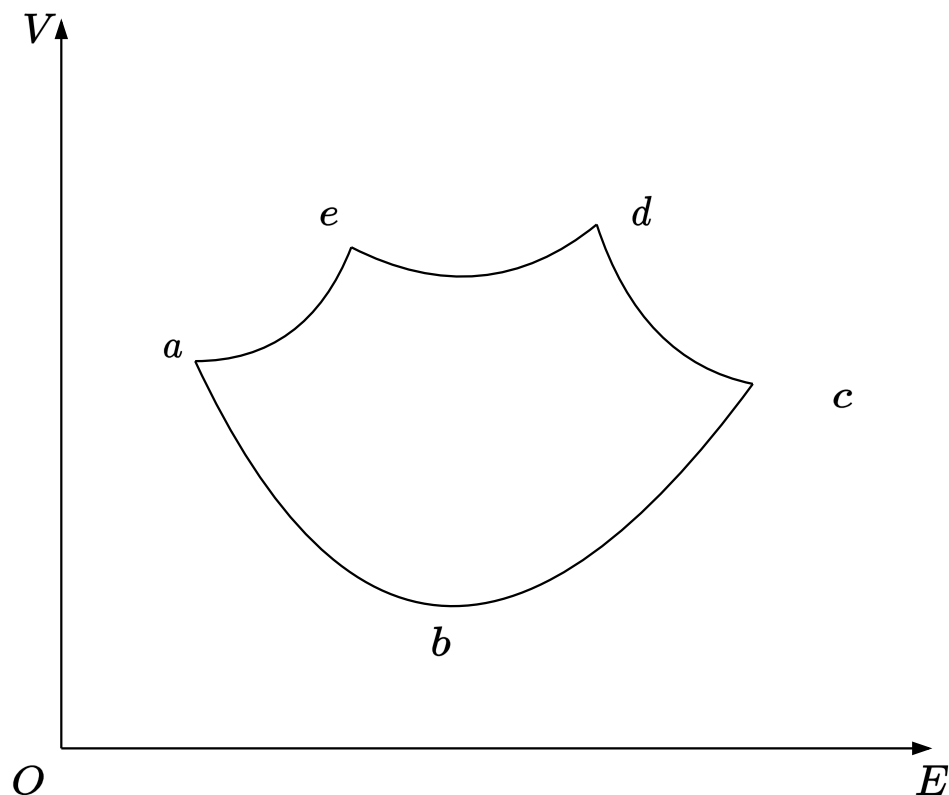
投资组合优化问题建模



E 代表收益, V 代表风险。

某一些位置, 同样的风险,
会有收益更高的点。

投资组合优化问题建模



E 代表收益, V 代表风险。

弧 bc 是有效的 EV 集合。

收益

$$E = \mathbf{w}^T \mathbf{R}$$

VS

风险

$$V = \mathbf{w}^T \Sigma \mathbf{w}$$

收益

VS

风险

$$E = \mathbf{w}^T \mathbf{R}$$

$$V = \mathbf{w}^T \Sigma \mathbf{w}$$

对于固定的收益，要求风险尽可能小。

收益

VS

风险

$$E = \mathbf{w}^T \mathbf{R}$$

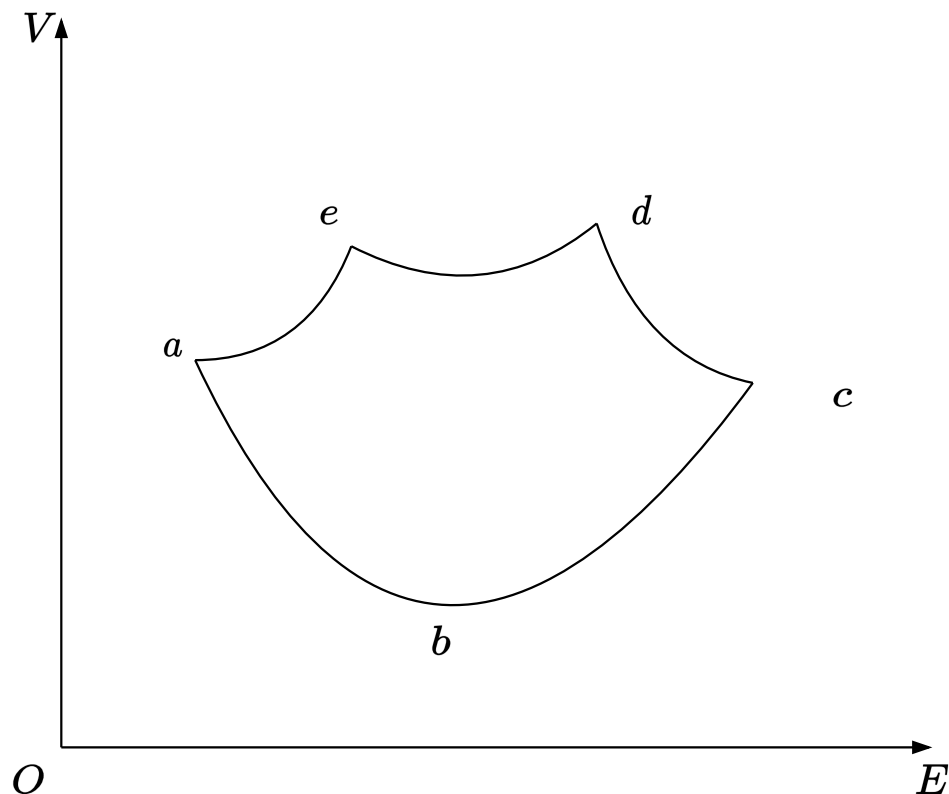
$$V = \mathbf{w}^T \Sigma \mathbf{w}$$

对于固定的收益，要求风险尽可能小。



对于固定的风险，要求收益尽可能大。

投资组合优化问题建模



E 代表收益, V 代表风险。

弧 bc 是有效的 EV 集合。

01

量子线性求解器

1. 转化为LP
2. 转化为SOCP

02

量子优化算法

1. 量子退火
2. VQE
3. QAOA

01

量子线性求解器

1. 转化为LP
2. 转化为SOCP

02

量子优化算法

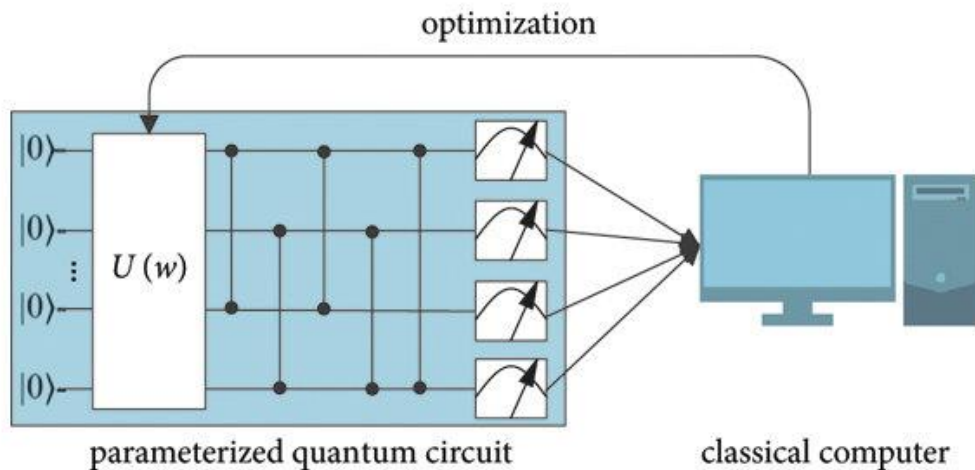
1. 量子退火
 2. **VQE**
 3. **QAOA**
- 举例**

VQE&QAOA简介



参数化电路

顾名思义，电路上有若干带参数的量子门。通过经典的优化方法（梯度下降、群体智能等）调整参数使得输出结果达到最优。



$$H_{\text{operator}} = \frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V_{(x)}$$

Hamiltonian
Operator

Operator for
Kinetic Energy

Potential
Energy

哈密顿量

哈密顿量用于表示系统能量。可简单认为哈密顿量 H 是用于量子测量的矩阵，测量得到的值可以认为是目标函数值。

Improving Variational Quantum Optimization using CVaR

Panagiotis Kl. Barkoutsos¹, Giacomo Nannicini², Anton Robert^{1,3}, Ivano Tavernelli¹, and Stefan Woerner¹

¹IBM Research – Zurich

²IBM T.J. Watson Research Center

³École Normale Supérieure, PSL University, Paris

$$\begin{aligned} \min_{x \in \{0, 1\}^n} \quad & qx^T \Sigma x - \mu^T x \\ \text{subject to: } & 1^T x = B \end{aligned}$$

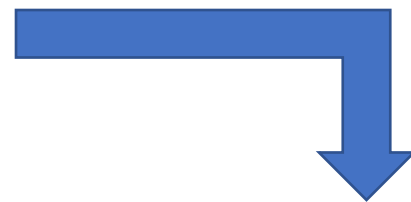
$$\min_{x \in \{0, 1\}^n} qx^T \Sigma x - \mu^T x$$
$$\text{subject to: } 1^T x = B$$

- $x \in \{0, 1\}^n$ 表示投资方案, $x_i = 1$ 表示选择第 i 个投资品, $x_i = 0$ 反之;
- $\mu \in \mathbb{R}^n$ 表示资产预期收益;
- $\Sigma \in \mathbb{R}^{n \times n}$ 表示指定资产间的协方差矩阵;
- $q > 0$ 是投资人的风险偏好;
- B 表示预算。

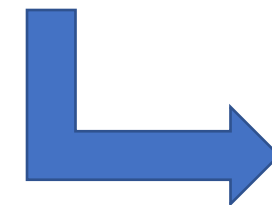
$$\min_{x \in \{0, 1\}^n} qx^T \Sigma x - \mu^T x$$

$$\text{subject to: } 1^T x = B$$

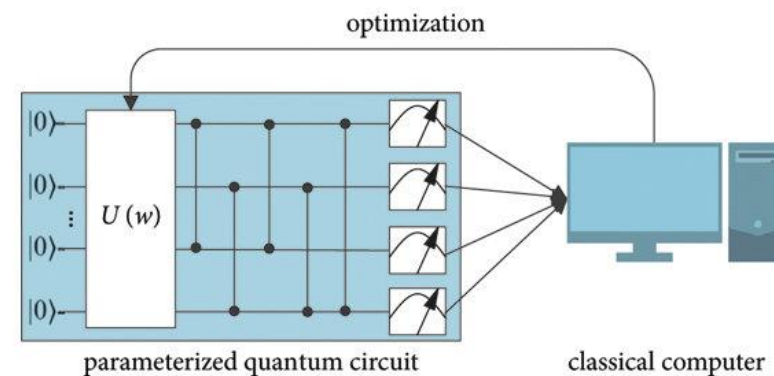
转二次规划问题



$$ax_1 + bx_2 + cx_1^2 + bx_2^2 + \dots$$



VQE&QAOA





01 量子金融产业现状

02 投资组合优化

03 欧式看涨期权定价

04 编程实践



期权

赋予持有人在某一特定日期或该日之前的任何时间以固定价格购进或售出一种资产的权利。



欧式期权

欧式期权是指买入期权的一方必须在期权到期日当天才能行使的期权。



看涨期权

期权的购买者拥有在期权合约有效期内按执行价格买进一定数量标的物的权利。

定价问题



已知 当前价格 S_0 。

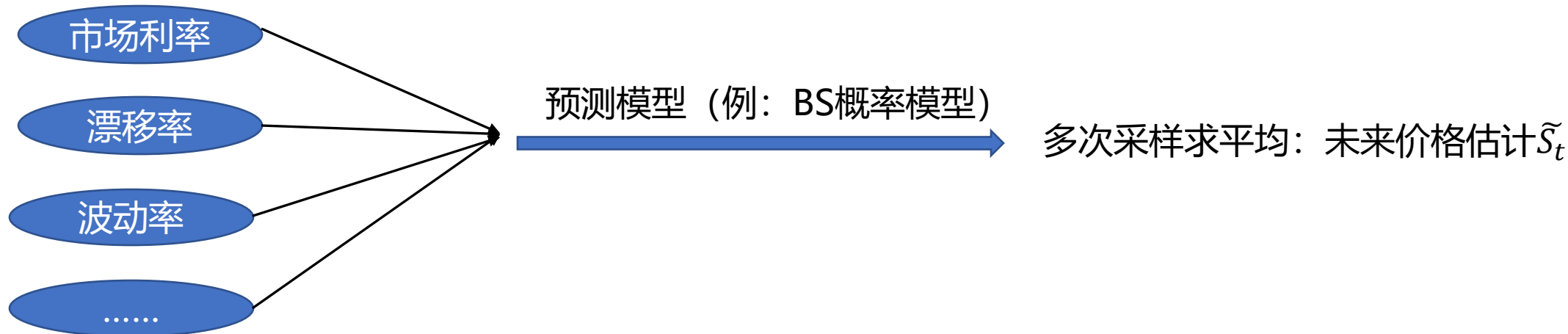
未知 未来价格 S_t 。

定价问题



已知 当前价格 S_0 。

未知 未来价格 S_t 。



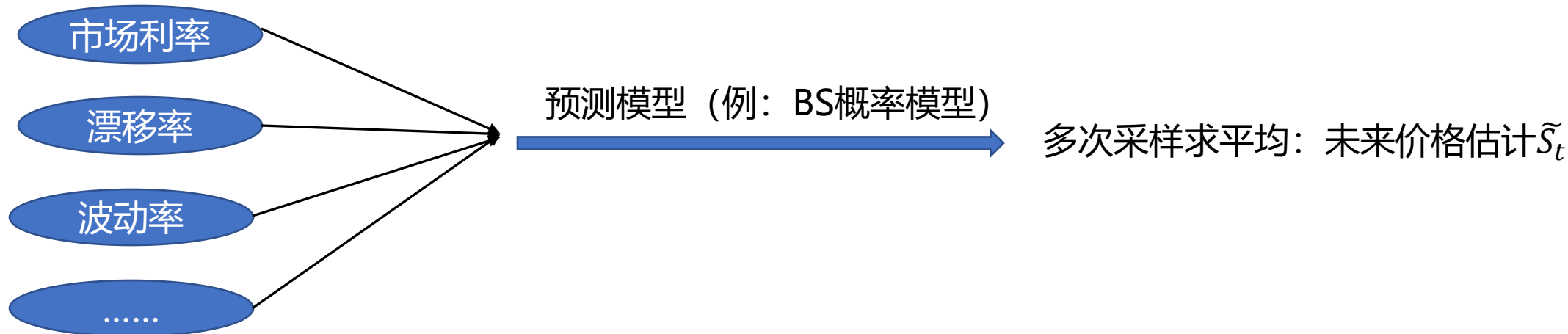
定价问题

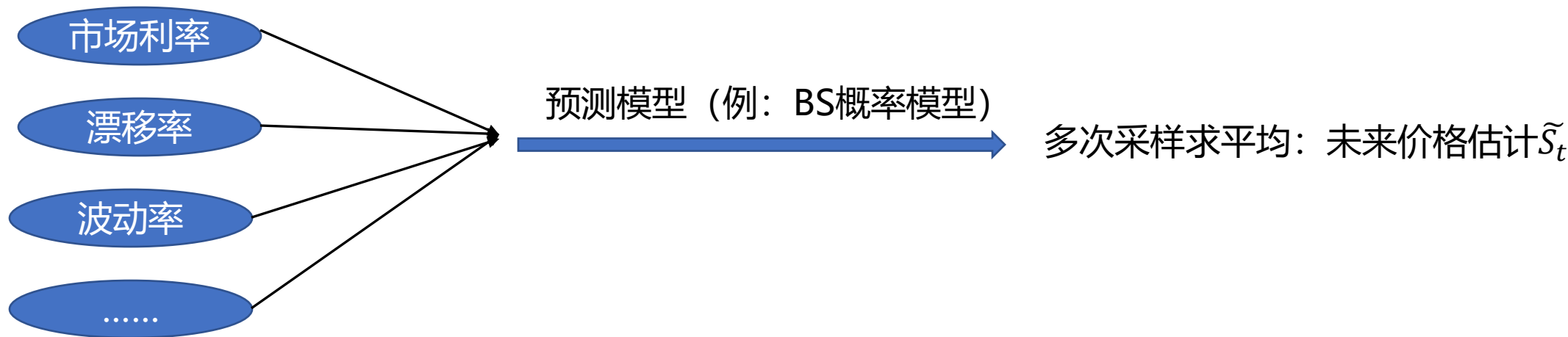


已知 当前价格 S_0 。

未知 未来价格 S_t 。

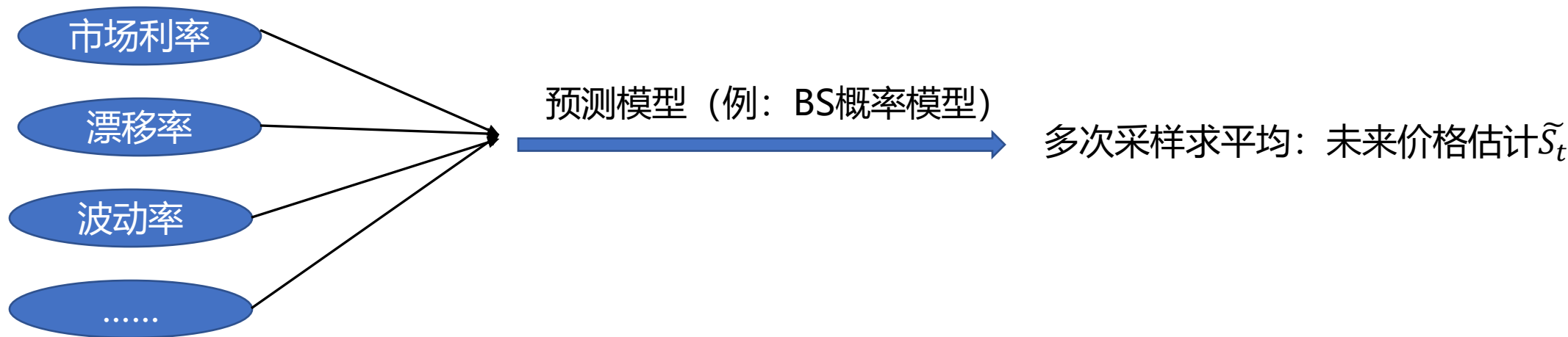
$\tilde{S}_t > S_0$: 有盈利潜力
 $\tilde{S}_t \leq S_0$: 有亏损风险





采样10次，得10个值：[1,2,2,3,3,3,20,20,40,50]。

$$\tilde{S}_t = \frac{1 + 2 + 2 + 3 + 3 + 3 + 20 + 20 + 40 + 50}{10} = 14.4$$



采样10次，得10个值：[1,2,2,3,3,3,20,20,40,50]。

$$\tilde{S}_t = \frac{1}{10} \times 1 + \frac{1}{5} \times 2 + \frac{3}{10} \times 3 + \frac{1}{5} \times 20 + \frac{1}{10} \times 40 + \frac{1}{10} \times 50 = 14.4$$

欧式看涨期权定价



- S_t : 到期日价格
- K : 成交价

到期收益: $w = \max\{S_t - K, 0\}$

估计到期收益 $\mathbb{E}[\max\{S_T - K, 0\}]$

假设 $S_t \in [a, b]$, 可离散化为 2^n 个可能的取值, 量子态 $|i\rangle$ 表示一个可能的取值:

$$\phi(i) = \frac{i}{2^n - 1}(b - a) + a$$

对应的收益为:

$$W(i) = \max\{\phi(i) - K, 0\}$$

假设 $S_t = \phi(i)$ 的概率为 p_i , 制备量子态:

$$|0\rangle_n \mapsto |\psi\rangle_n = \sum_{i=0}^{2^n-1} \sqrt{p_i} |i\rangle_n$$

引入辅助量子比特和算子 $L(W(i)) \in [0,1]$, 执行酉变换:

$$|i\rangle_n |0\rangle \mapsto \sqrt{1 - L(W(i))} |i\rangle_n |0\rangle + \sqrt{L(W(i))} |i\rangle_n |1\rangle$$

$$|i\rangle_n |0\rangle \mapsto \sqrt{1 - L(W(i))} |i\rangle_n |0\rangle + \sqrt{L(W(i))} |i\rangle_n |1\rangle$$

估计 $|1\rangle$ 上的概率

$$\mathbb{E} \approx \sum p_i W(i) = L^{-1} \left(\sum p_i L(W(i)) \right)$$



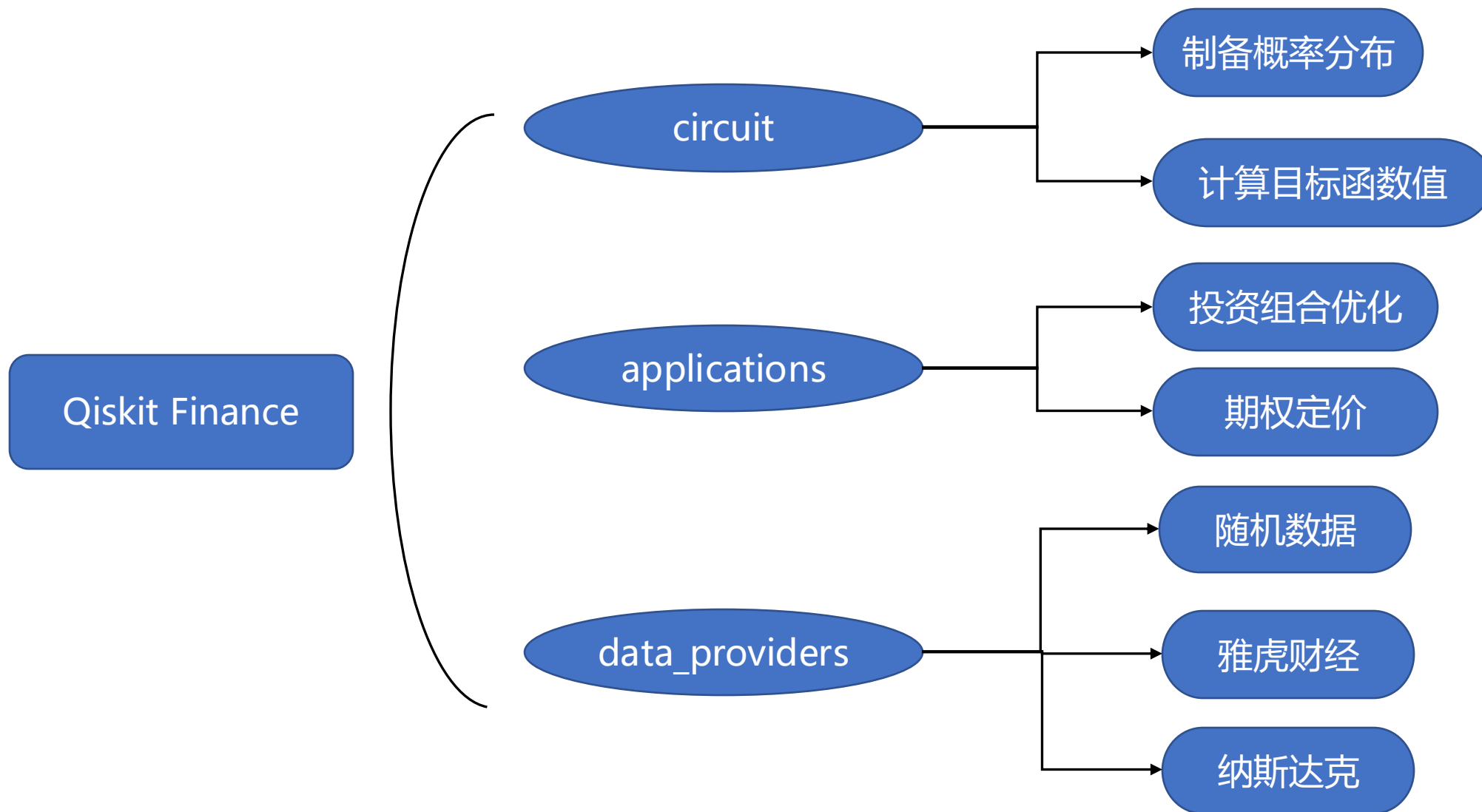
01 量子金融产业现状

02 投资组合优化

03 欧式看涨期权定价

04 编程实践

Qiskit Finance包结构



编程实例：投资组合优化



```
import datetime
from qiskit_finance.data_providers import RandomDataProvider

# 股票数
num_stocks = 4
data_provider = RandomDataProvider(
    # 股票名称
    tickers=[f"TICKER{i}" for i in range(num_stocks)],
    # 起始时间
    start=datetime.datetime(2023, 1, 1),
    # 终止时间
    end=datetime.datetime(2023, 6, 1),
)
# 造数据
data_provider.run()
```

获取金融数据

编程实例：投资组合优化



```
import datetime
from qiskit.finance.providers import RandomDataProvider

# 股票数据
num_stocks = 4
data_provider = RandomDataProvider(
    # 股票名称
    tickers=[TICKER[i] for i in range(num_stocks)],
    # 起始时间
    start=datetime.datetime(2023, 1, 1),
    # 终止时间
    end=datetime.datetime(2023, 6, 1),
)
# 造数据
data_provider.run()
```

data_provider是获取经融数据的模块，这里使用RandomProvider生成随机数据，其它Provider如YahooDataProvider可爬取雅虎财经的真实金融数据。所有Provider都继承自BaseProvider。

1. RandomProvider: 随机数据
2. YahooDataProvider: 雅虎财经
3. WikipediaDataProvider: 调用Nasdaq Data Link的Restful API
4.

参考: https://qiskit.org/ecosystem/finance/tutorials/11_time_series.html

编程实例：投资组合优化



```
from qiskit_finance.applications import PortfolioOptimization

portfolio = PortfolioOptimization(
    expected_returns=data_provider.get_period_return_mean_vector(),
    covariances=data_provider.get_period_return_covariance_matrix(),
    risk_factor=1,
    budget=3,
)
```

**设置风险系数和预算，
构造投资组合问题。**

applications模块下有诸多量子金融应用，主要是投资组合优化和期权定价两类：

1. PortfolioOptimization
2. PortfolioDiversification
3. EuropeanCallPricing
4. EuropeanCallDelta
5. FixedIncomePricing

编程实例：投资组合优化



```
qp = portfolio.to_quadratic_program()
```

将原问题转为二次优化问题，
打印出来的形式为最小化：

```
0.07561071946848313*x_0^2 +  
0.0007399823698913228*x_0*x_1 -  
0.0015050238660271349*x_0*x_2 +  
0.0003335366346480111*x_0*x_3 +  
0.0005162538317453201*x_1^2 -  
2.444858496773193e-06*x_1*x_2 -  
0.0002080850319349677*x_1*x_3 +  
0.00017834703639235414*x_2^2 -  
1.699300347337927e-05*x_2*x_3 +  
0.00127594810153012*x_3^2 +  
0.00438612319576254*x_0 +  
0.002299712822593353*x_1 -  
0.001277482315067536*x_2 +  
0.0068806182646217766*x_3
```

编程实例：投资组合优化



```
from qiskit.algorithms.optimizers import COBYLA
from qiskit.algorithms.minimum_eigensolvers import QAOA
from qiskit.circuit.library import TwoLocal
from qiskit_optimization.algorithms import MinimumEigenOptimizer
from qiskit_aer.primitives import Sampler
```

```
result = MinimumEigenOptimizer(
    QAOA(
        sampler=Sampler(),
        optimizer=COBYLA(maxiter=500),
    )
).solve(qp)
```

用QAOA解决二次规划问题。