EC21 - Designing a Combinatorial Financial Options Market

Zhengyu Jin

Department of Computer Science and Technology
Zhejiang University

2025 年 11 月 26 日

Overview

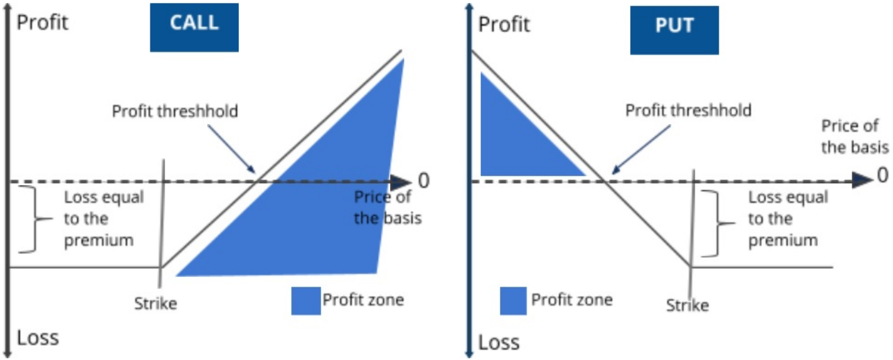
1. **Introduction**
2. **Notations**
3. **Consolidating standard financial options**
   - 3.1 Matching Orders on Standard Financial Options
   - 3.2 Quoting Prices for Standard Financial Options
4. **Combinatorial Financial Options**
   - 4.1 Matching Orders on Combinatorial Financial Options
   - 4.2 Computing Complexity for Combinatorial Financial Options
   - 4.3 A Constraint Generation Algorithm to Match Combinatorial Financial Options
5. **Experiments**
   - 5.1 Real-Market Standard Financial Market
   - 5.2 Synthetic Combinatorial Financial Market

1. Introduction

Financial Options

- 金融期权是一种合约，赋予买方在指定日期或之前以指定执行价格买入或卖出标的资产的权利（不是义务）。
- 两种主要的期权类型：看涨期权和看跌期权。
  - 看涨期权（Call Option）：赋予买方在未来某一特定时间以特定价格购买标的资产的权利。
  - 看跌期权（Put Option）：赋予买方在未来某一特定时间以特定价格出售标的资产的权利。

Financial Options



Standard Financial Options Market

对于传统的金融期权市场，交易所会把不同类型、执行价格、到期时间的期权合约拆分成独立的市场。

买方和卖方分别向交易所报告自己的价格和数量需求，交易所根据供需情况撮合买卖双方进行交易，其中我们默认交易所可以进行“部分交易”(因为都采用现金结算，所以任意分数都可以)。

从交易所的视角来看希望设计一个匹配方法，使得交易所在没有任何风险的情况下，获得一笔直接的现金流。

Standard Financial Options Market

但是，标准的金融期权市场存在一些问题：

- 市场碎片化：不同类型、执行价格、到期时间的期权合约被拆分成独立的市场，导致流动性不足，买卖双方难以找到合适的交易对手。
- 难以组合交易：投资者可能希望同时买入或卖出多种期权合约以实现特定的投资策略，但这在标准市场中会导致较高的手续费，而且对于投资者的操作难度也会比较高。
- 对于交易所来说，更多的交易合并在一个市场，也就有更多的机会实现无风险套利。

Combinatorial Financial Options Market

组合型金融期权市场允许投资者提交包含多种期权合约的组合订单，从而实现更复杂的交易策略。

例如，投资者可以购买一个形如 $\{A + B, 100\}$ 的看涨期权，即赋予买方在未来某一特定时间以 100 的价格购买一份股票 A 和 B 的权利。

但是，组合期权市场将多种股票合并在一起计算，这使得计算策略的复杂度大大增加。

The Outline and Contribution of the Article

- Matching Orders on Standard Financial Options Markets
- Computing Complexity for Standard Financial Options Markets
- Matching Orders on Combinatorial Financial Options Markets
- **Computing Complexity for Combinatorial Financial Options Markets**
- **A Constraint Generation Algorithm to Match Combinatorial Financial Options**
- Experiments

2. Notations

期权类型与符号表示

- 看涨期权（**Call**）：$C(S, K, T)$ 赋予买方在到期日 T 以执行价格 K 买入标的资产 S 的权利
- 看跌期权（**Put**）：$P(S, K, T)$ 赋予买方在到期日 T 以执行价格 K 卖出标的资产 S 的权利
- 为简化起见，后续省略 T，因为机制仅聚合同一到期日的期权

期权收益与定价

- 期权买方决定是否行权。购买期权的收益（Payoff）为：

$$\Psi := \max\{\chi(S-K),0\}$$

  其中 S 是到期时标的资产的价值，$\chi\in\{-1,1\}$（看涨期权为 1，看跌期权为-1）
- 由于买方收益始终非负，卖方需收取权利金（**Premium**）作为未来义务的补偿
- 期权系列（**Option Series**）：相同标的资产、类型、执行价格和到期日的期权合约

限价订单簿（Limit Order Book）

- 对于单个证券的期权，包含看涨和看跌期权、10 个到期日和 50 个执行价格
- 所有期权系列总共形成一千个市场，每个市场维护独立的限价订单簿
- 判断交易是否存在需要 $O(1)$ 时间（比较最优买卖价）
- 匹配一个新订单需要 $O(n_o)$ 时间，其中 n_o 是订单簿对面的订单数量

3. Consolidating standard financial options

3.1. Matching Orders on Standard Financial Options

问题设定

- 考虑单一标的资产 S 和到期日 T 的期权市场
- 交易者可以对任意正执行价格的期权下单
- 买单集合（索引 $m \in \{1, 2, ..., M\}$）：
  - 类型向量：$\boldsymbol{\phi} \in \{-1, 1\}^M$（-1 表示看跌，1 表示看涨）
  - 执行价格向量：$\boldsymbol{p} \in \mathbb{R}_+^M$
  - 出价（bid prices）：$\boldsymbol{b} \in \mathbb{R}_+^M$
- 卖单集合（索引 $n \in \{1, 2, ..., N\}$）：
  - 类型向量：$\boldsymbol{\psi} \in \{-1, 1\}^N$
  - 执行价格向量：$\boldsymbol{q} \in \mathbb{R}_+^N$
  - 要价（ask prices）：$\boldsymbol{a} \in \mathbb{R}_+^N$

交易所的匹配决策

交易所需要决定：

- $\boldsymbol{\gamma} \in [0,1]^M$：卖给买单的比例
- $\boldsymbol{\delta} \in [0,1]^N$：从卖单购买的比例

目标：最大化交易时刻的净利润 $\boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta}$

约束：在未来所有可能的 S 状态下，收益损失为零（无套利）

无套利约束

定义交易所的权利和义务：

- 卖出期权的总价值（交易所的义务）：

$$\Psi_\Gamma := \sum_m \gamma_m \max\{\phi_m(S - p_m), 0\}$$

- 买入期权的总价值（交易所的权利）：

$$\Psi_\Delta := \sum_n \delta_n \max\{\psi_n(S - q_n), 0\}$$

无套利约束：交易所的义务不得超过其权利，对所有可能的 $S \in [0, \infty)$ 成立

$$\Psi_\Gamma - \Psi_\Delta \leq 0, \quad \forall S \in [0, \infty)$$

弱占优（Weakly Dominate）

- 记交易所买入的期权组合为投资组合 Δ
- 记交易所卖出的期权组合为投资组合 Γ
- 约束确保投资组合 Δ 弱占优投资组合 Γ：

$$\Psi_\Delta(S) \geq \Psi_\Gamma(S), \quad \forall S \in [0, \infty)$$

即：在任何未来状态下，交易所持有的期权收益都不低于其义务

扩展模型：引入损失容忍度 L

但是之前考虑的弱占优条件有些苛刻，因此我们考虑引入一个容忍度 L:

$$\max_{\gamma,\delta,L} \quad \boldsymbol{b}^\top \gamma - \boldsymbol{a}^\top \boldsymbol{\delta} - L \tag{M.1}$$

$$\text{s.t.} \quad \sum_m \gamma_m \max\{\phi_m(S - p_m), 0\} - \sum_n \delta_n \max\{\psi_n(S - q_n), 0\} \leq L, \quad \forall S \in [0, \infty) \tag{1}$$

- 约束保证：义务与收益之差不超过 L，二者差值所造成的可能风险直接从交易时刻的利润中扣除
- 此时称投资组合 Δ 以常数偏移 L 弱占优投资组合 Γ

带偏移的收益占优

定义

如果对于到期时标的变量的所有可能状态，投资组合 Δ 的收益加上常数 L 大于或等于投资组合 Γ 的收益，则称投资组合 Δ 以偏移 L **(弱) 占优**投资组合 Γ。

机制 **M.1** 的优势：

- 由于潜在盈亏已纳入匹配时的目标函数，机制 M.1 保证了每次匹配总体无损失
- 提供了额外的交易自由度
- 如果限制交易所零收益损失，下面的例子中的匹配将无法实现

Positive Offset ($L > 0$)

例

使用 M.1 整合迪士尼公司（DIS）在 2019 年 1 月 23 日定价、2019 年 6 月 21 日到期的期权。

找到以下匹配（每个订单在其独立市场中都无法成交）：

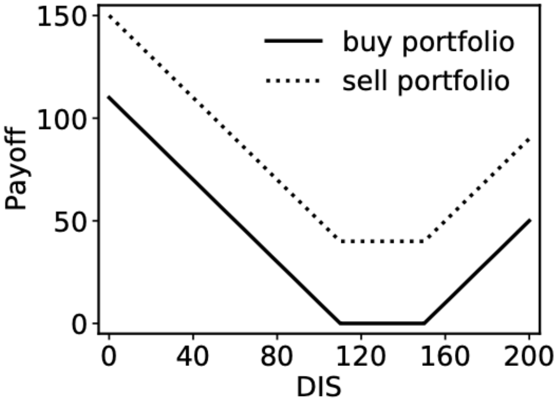
- 以出价 \$7.2 卖给买单 $C(\mathsf{DIS}, 110)$
- 以出价 \$38.75 卖给买单 $P(\mathsf{DIS}, 150)$
- 以要价 \$0.05 从卖单 $C(\mathsf{DIS}, 150)$ 购买
- 以要价 \$5.1 从卖单 $P(\mathsf{DIS}, 110)$ 购买

交易所的即时收益：$\$40.8 = 7.2 + 38.75 - 5.1 - 0.05$

无论 DIS 到期时价值如何，交易所的净负债为 \$40（即 $L = 40$）

交易所在无风险情况下获得净利润 \$0.80

匹配收益可视化



上图展示了匹配期权的收益随到期时标的资产价值的变化函数。

Negative Offset ($L < 0$)

例

使用 M.1 整合苹果公司（AAPL）在 2019 年 1 月 23 日定价、2020 年 1 月 17 日到期的期权。

找到以下匹配（每个订单在其独立市场中都无法成交）:

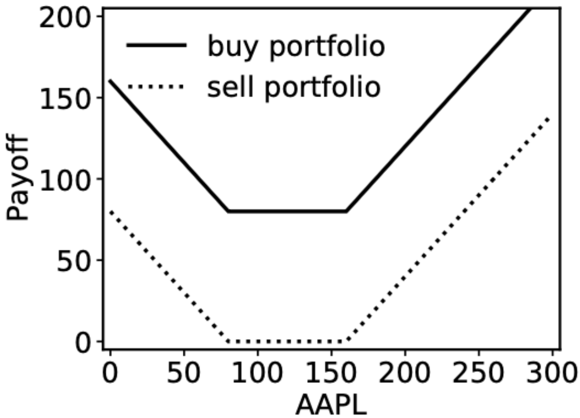
- 以出价 $14.1 卖给买单 $C(\mathsf{AAPL}, 160)$
- 以出价 $0.62 卖给买单 $P(\mathsf{AAPL}, 80)$
- 以要价 $74.2 从卖单 $C(\mathsf{AAPL}, 80)$ 购买
- 以要价 $19.1 从卖单 $P(\mathsf{AAPL}, 160)$ 购买

匹配产生即时费用：$\$78.58 = 14.1 + 0.62 - 74.2 - 19.1$

但保证到期时获得 $80 的收益（即 $L = -80$）

可以推断出隐含利率为 1.82%，由 $78.58e^{r\Delta t} = 80$ 计算得出

匹配收益可视化



上图展示了所买卖期权的各自收益。无论 AAPL 到期时价值如何，交易所的净收益恒为 \$80。

复杂度分析

$$\max_{\gamma,\delta,L} \quad \boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta} - L \tag{M.1}$$

$$\text{s.t.} \quad \sum_m \gamma_m \max\{\phi_m(S-p_m),0\} - \sum_n \delta_n \max\{\psi_n(S-q_n),0\} \leq L, \quad \forall S \in [0,\infty) \tag{1}$$

现在分析运行机制 M.1 的复杂度。约束 (1) 的左侧是 max 函数的线性组合，因此是 S 的分段线性函数。

因此，通过在每个断点处满足由 S 定义的约束来求解 M.1 就足够了。在此问题中，约束 (1) 的断点是市场中定义的执行价格，加上两个端点：$\boldsymbol{p} \cup \boldsymbol{q} \cup \{0,\infty\}$。

设 n_K 表示市场中不同执行价格的数量，其上界为 $n_{\text{orders}} = M + N$，即市场中的订单总数。因此，M.1 是一个具有 $n_K + 2$ 个收益约束的线性规划，求解所需时间关于问题实例大小是多项式的。

Remarks

Remarks:

- 可以通过在 M.1 的目标函数中乘以贴现率来纳入投资的时间价值
- M.1 返回的匹配解可能涉及股票的小数份额。理想情况下，交易所（如现金结算市场）允许这样做；如果交易所（如实物结算市场）不允许，我们需要进行标准化或取整
- M.1 识别了在没有风险的情况下可以接受的订单的最大捆绑包，但没有具体说明如何处理盈余（如果有）。盈余可以在相关交易者和交易所之间任意分配

3.2. Quoting Prices for Standard Financial Options

Price Quote Procedure

交易所扮演做市商的角色，当用户询问一个自定义期权 (χ, S, K) 的价格时，交易所需要报出两个价格：

- 买价：交易所愿意从用户手中买入该期权的最高价格。
- 卖价：交易所愿意向用户卖出该期权的最低价格。

为了确定这些价格，交易所不会凭空猜测，而是会思考："如果我进行了这笔交易，我如何通过市场上现有的其他期权来完全对冲我的风险？"这个对冲组合的成本或收益，就决定了报价。

最佳出价 b^* 的计算

1. 期权 (χ, S, K) 的最佳出价 b^* 是出售被某个常数偏移 L 弱占优的期权组合 (χ, S, K) 的最大收益。
2. 我们通过将期权 (χ, S, K) 添加到市场的卖方，将其索引为 $N+1$（因为交易所从卖方购买），将其价格 a_{N+1} 初始化为 0，并求解 M.1 来推导 b^*。最佳出价 b^* 即为返回的目标值。

最佳出价 b^* 的证明

证明.

我们证明上述过程通过返回出售弱占优期权组合的最大收益来找到最佳出价。将 (χ, S, K, T) 添加到市场后，我们有更新的 M.1 如下：

$$\max_{\boldsymbol{\gamma},\boldsymbol{\delta},L} \quad \boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta} - a_{N+1}\delta_{N+1} - L$$

$$\text{s.t.} \quad \sum_m \gamma_m \max\{\phi_m(S - p_m), 0\} - \sum_n \delta_n \max\{\psi_n(S - q_n), 0\}$$

$$-\delta_{N+1} \max\{\chi(S - K), 0\} \leq L \quad \forall S \in [0, \infty)$$

最佳出价 b^* 的证明（续）

证明（续）.

由于我们设置 $a_{N+1}=0$，购买期权 (χ, S, K, T) 且令 $\delta_{N+1}=1$ 总是最优的。因此，我们有以下从 M.1 扩展的优化问题：

$$\max_{\boldsymbol{\gamma},\boldsymbol{\delta},L} \quad z := \boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta} - L \tag{M.1-bid}$$

$$\text{s.t.} \quad \underbrace{\sum_m \gamma_m \max\{\phi_m(S-p_m),0\} - \sum_n \delta_n \max\{\psi_n(S-q_n),0\}}_{\text{组合}(*)}$$

$$\leq \max\{\chi(S-K),0\} + L \quad \forall S \in [0,\infty)$$

最佳出价 b^* 的证明（续）

证明（续）.

上述不等式左侧描述的是被目标期权 (χ, S, K, T) 以偏移 L 弱占优的现有期权组合的收益，目标函数 z 最大化出售这样一个组合的收益。

交易所愿意以最高价格（即最佳出价）$b^* = z$ 购买 (χ, S, K, T)，因为它总可以出售弱占优的组合 $(*)$ 无风险地收回 z。 $\square$

最佳要价 a^* 的计算

1. 期权 (χ, S, K) 的最佳要价 a^* 是购买以某个常数偏移 L 弱占优 (χ, S, K) 的期权组合的最小成本。
2. 我们通过将期权 (χ, S, K) 添加到市场的买方，将其索引为 $M+1$，将其价格 b_{M+1} 初始化为一个大数（例如 10^{10}），并求解 M.1 来推导 a^*。最佳要价 a^* 即为 b_{M+1} 减去返回的目标值。

直观理解：交易所购买该期权需要支付的最小成本，同时确保其购买的期权组合能够占优该期权（加上某个偏移）。

4. Combinatorial Financial Options

组合型金融期权的定义

定义 (组合型金融期权)

考虑一组 U 个标的资产。组合型金融期权是一种衍生合约，它规定了在到期日以执行价格买入或卖出这 U 个标的资产的线性组合的权利。

每个合约指定一个看涨或看跌类型 $\chi \in \{1, -1\}$，一个权重向量 $\boldsymbol{\omega} \in \mathbb{R}^U$，一个执行价格 $K \geq 0$，以及一个到期日 T。它的收益为：

$$\max\{\chi(\boldsymbol{\omega}^\top \boldsymbol{S} - K), 0\}$$

其中 $\boldsymbol{S} \in \mathbb{R}_+^U$ 是到期时标的资产价值的向量。

组合期权示例

例

考虑一个组合期权 $C(\text{MSFT} - \text{AAPL}, 0)$，其权重向量为：MSFT 权重为 1，AAPL 权重为-1，执行价格为零。

购买该期权的投资者是在押注微软的表现会超过苹果公司，并且会在 $S_{\text{MSFT}} > S_{\text{AAPL}}$ 时行权。

因此，与只会根据单一证券价格变化支付的标准期权不同，组合期权押注的是资产或资产组之间的相对变动，从而能够表达不同标的资产之间的未来相关性。

组合期权和标准期权的区别

我们注意到关于组合型金融期权定义的几个区别和解释:

(1) 组合期权与标准期权组合的区别

考虑上述组合期权 $C(\mathsf{MSFT}-\mathsf{AAPL},0)$ 和两个标准期权的组合 $C(\mathsf{MSFT},K)$ 和 $P(\mathsf{AAPL},K)$。

- 行权组合期权时会持有投资组合，即当 $S_{\mathsf{MSFT}} > S_{\mathsf{AAPL}}$ 时，都会以 0 的代价买入一股 MSFT 并卖出一股 AAPL。
- 而在标准期权组合中，当 $S_{\mathsf{MSFT}} \geq K$ 且 $S_{\mathsf{AAPL}} \leq K$ 时才会有相同的效果。

因此，具有不同标的证券的几个标准金融期权的同时买卖无法复制组合期权的收益。

组合期权和标准期权的区别

(2) 看涨和看跌组合期权之间的区别

我们注意到，对于组合期权，看涨和看跌之间的区别取决于在合约中指定的执行价格和系数。例如，在上面的例子中，$C(\mathsf{MSFT} - \mathsf{AAPL}, 0)$ 等同于 $P(\mathsf{AAPL} - \mathsf{MSFT}, 0)$，因为它们具有相同的收益函数 $\max\{S_{\mathsf{MSFT}} - S_{\mathsf{AAPL}}, 0\}$。

尽管解释和表达不同，我们遵循标准金融期权的惯例，执行价格始终为非负。

4.1. Matching Orders on Combinatorial Financial Options

Example 3: 匹配组合期权订单

例

考虑一个包含四个订单的组合期权市场:

- o_1: 以出价 \$110 买入一个 $C(1\mathsf{AAPL} + 2\mathsf{MSFT}, 300)$
- o_2: 以出价 \$70 买入一个 $C(1\mathsf{AAPL} + 1\mathsf{MSFT}, 300)$
- o_3: 以要价 \$160 卖出一个 $C(1\mathsf{AAPL} + 3\mathsf{MSFT}, 300)$
- o_4: 以要价 \$5 卖出一个 $C(1\mathsf{AAPL}, 250)$

交易所可以同时卖给 o_1 和 o_2，并从 o_3 和 o_4 购买，获得即时收益 \$15 $(110 + 70 - 160 - 5)$。

Example 3: 收益分析

下图为到期日时交易所的收益关于 S_{AAPL} 和 S_{MSFT} 的函数：

$$\Psi := \max\{S_{\mathsf{AAPL}} + 3S_{\mathsf{MSFT}} - 300, 0\} + \max\{S_{\mathsf{AAPL}} - 250, 0\} - \max\{S_{\mathsf{AAPL}} + 2S_{\mathsf{MSFT}} - 300, 0\} - \max\{S_{\mathsf{AAPL}} + S_{\mathsf{MSFT}} - 300, 0\}$$

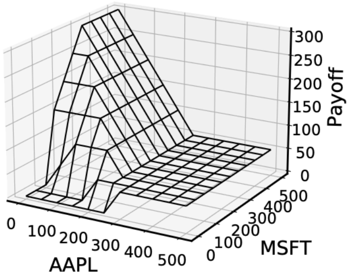


图: Example 3 中匹配的组合期权收益（$L = 0$）

Example 3: Summary

- $L = 0$，即到期日时交易所的净收益始终非负，因此可以无风险地获利 \$15
- 在上面的例子中，交易所可以考虑将每个单独的买单依次匹配到某些卖单的组合。例如，交易所可以先卖给买单 o_1 并从卖单 $\frac{2}{3}o_3$ 和 $\frac{1}{3}o_4$ 购买，然后卖给买单 o_2 并从卖单 $\frac{1}{3}o_3$ 和 $\frac{2}{3}o_4$ 购买
- 这两笔交易都是有利可图的，且不存在损失，最终匹配也相同

Example 4: 批量匹配组合期权订单

下面这个例子表明，将每个单独的买单顺序匹配到卖单的某种组合可能无法找到有效的交易。

例

考虑以下四个组合期权订单：

- o_1：以出价 \$6 买入一个 $C(A+B,10)$
- o_2：以出价 \$6 买入一个 $C(B+C,7)$
- o_3：以要价 \$10 卖出一个 $C(A+B+C,7)$
- o_4：以要价 \$2 卖出一个 $C(B,3)$

如果我们单独考虑每个买单：覆盖 o_1 或 o_2 中的已售组合期权需要购买 o_3 的相同比例，但其价格更高，会导致净损失。

Example 4: 批量匹配的优势

然而，如果同时处理所有订单，存在一个有效的匹配：同时卖给 o_1 和 o_2，并从 o_3 和 o_4 购买。

交易所的收益：

- 即时成本：\$0（因为 $6+6-10-2=0$）
- 未来可以获得正收益

交易所持有的期权满足：

$$\max\{S_A+S_B-10,0\}+\max\{S_B+S_C-7,0\} \le \max\{S_A+S_B+S_C-7,0\}+\max\{S_B-3,0\}$$

对于所有非负的 S_A, S_B, S_C 成立，这意味着负债总是不大于买入期权的收益。

4.2. Computing Complexity for Combinatorial Financial Options

组合期权市场的形式化设定

组合期权市场是一个双边市场，包含一组买单限价订单（索引 $m \in \{1, 2, ..., M\}$）和一组卖单限价订单（索引 $n \in \{1, 2, ..., N\}$）。

买单表示：

- 类型向量 $\boldsymbol{\phi} \in \{1, -1\}^M$（每个元素指定看跌或看涨组合期权）
- 权重矩阵 $\boldsymbol{\alpha} \in \mathbb{R}^{U \times M}$（指定线性组合）
- 执行价格向量 $\boldsymbol{p} \in \mathbb{R}_+^M$
- 出价向量 $\boldsymbol{b} \in \mathbb{R}_+^M$

卖单表示：类似地由 $\boldsymbol{\psi} \in \{1, -1\}^N$，$\boldsymbol{\beta} \in \mathbb{R}^{U \times N}$，$\boldsymbol{q} \in \mathbb{R}_+^N$，$\boldsymbol{a} \in \mathbb{R}_+^N$ 定义。

组合期权的匹配机制 M.2

交易所决定比例 $\boldsymbol{\gamma} \in [0,1]^M$ 卖给买单，比例 $\boldsymbol{\delta} \in [0,1]^N$ 从卖单购买，以最大化净利润。我们推广标准期权的机制 M.1 来促进组合期权交易：

$$\max_{\gamma,\boldsymbol{\delta},L} \quad \boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta} - L \tag{M.2}$$

$$\text{s.t.} \quad \sum_m \gamma_m \max\{\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m), 0\} - \sum_n \delta_n \max\{\psi_n(\boldsymbol{\beta}_n^\top \boldsymbol{S} - q_n), 0\} \le L \quad \forall \boldsymbol{S} \in \mathbb{R}_+^U \tag{2}$$

Sauer 引理 (Sauer's Lemma)

引理 (Sauer 引理)

给定 n 个半空间（*halfspaces*）在 d 维空间中，它们最多可以将空间划分为

$$\sum_{i=0}^{d} \binom{n}{i} = \mathcal{O}(n^d)$$

个不同的区域。

在我们的问题中：

- 每个期权的收益函数 $\max\{\phi(\boldsymbol{\alpha}^\top \boldsymbol{S} - p), 0\}$ 定义了一个半空间
- 有 $M + N$ 个期权，即 $M + N$ 个半空间
- 标的资产价值空间 $\boldsymbol{S} \in \mathbb{R}_+^U$ 是 U 维的
- 因此问题会变成一个 $\mathcal{O}((M+N)^U)$ 段的分段约束，每段对应一个约束变量，复杂度是指数级的（具体分析可以见算法部分的“大 M 分析法”）

组合期权的 NP 完全性

定理

考虑市场中的所有组合期权 $(\phi, \boldsymbol{\alpha}, \mathbf{p}, \psi, \boldsymbol{\beta}, \mathbf{q})$。对于任何固定的 L，判定以下问题是 *NP* 完全的：

- ***Yes:*** $\gamma = \delta = 1$ 对某个 $\boldsymbol{S}$ 违反 *M.2* 中的约束
- ***No:*** $\gamma = \delta = 1$ 对所有 $\boldsymbol{S}$ 满足 *M.2* 中的约束

即使假设每个组合期权最多涉及两个标的资产。

证明思路：通过从 Vertex Cover 问题（顶点覆盖问题）归约来证明 NP 困难性。

4.3. A Constraint Generation Algorithm to Match Combinatorial Financial Options

Notations

- ϕ **(Phi) - 类型向量 (Type Vector):**
    - $\phi \in \{1, -1\}^M$
    - 表示每个买单是看涨（**Call**）还是看跌（**Put**）类型的期权。通常 1 代表 Call，-1 代表 Put。
- α **(Alpha) - 权重矩阵 (Weight Matrix):**
    - $\alpha \in \mathbb{R}^{U \times M}$
    - 定义了每个买单所对应的标的资产组合。例如，如果一个期权是" 买入 1 份 Apple + 2 份 Microsoft"，这些系数（1 和 2）就存储在这个矩阵中。
- p **- 执行价格向量 (Strike Price Vector):**
    - $p \in \mathbb{R}^M_{\geq 0}$
    - 表示每个买单期权的行权价（即期权到期时，触发盈亏的基准价格）。
- b **- 买入报价向量 (Bid-Price Vector):**
    - $b \in \mathbb{R}^M$
    - 表示买家愿意为购买该期权支付的最高价格（即现价/权利金）。

Notations

- ψ **(Psi) - 类型向量 (Type Vector):**
  - $\psi \in \{1, -1\}^N$
  - 表示每个卖单的期权类型（Call 或 Put）。
- β **(Beta) - 权重矩阵 (Weight Matrix):**
  - $\beta \in \mathbb{R}^{U \times N}$
  - 定义了每个卖单所对应的标的资产组合权重。
- q **- 执行价格向量 (Strike Price Vector):**
  - $q \in \mathbb{R}^N_{\geq 0}$
  - 表示每个卖单期权的行权价。
- a **- 卖出报价向量 (Ask-Price Vector):**
  - $a \in \mathbb{R}^N$
  - 表示卖家愿意出售该期权的最低价格（即要价）。

Algorithm Outline

Algorithm 1 Match orders in a combinatorial options market.

Input: A combinatorial options market defined by $(\boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{p}, \boldsymbol{b}, \boldsymbol{\psi}, \boldsymbol{\beta}, \boldsymbol{q}, \boldsymbol{a})$.
Output: An optimal clearing that matches $\boldsymbol{\gamma}^*$ buy orders to $\boldsymbol{\delta}^*$ sell orders.

Initialize $z \leftarrow \infty$, $\boldsymbol{S} \leftarrow \boldsymbol{0}$,
$\quad \boldsymbol{f} \leftarrow \max\{\boldsymbol{\phi}(\boldsymbol{\alpha}^\top \boldsymbol{S} - \boldsymbol{p}), \boldsymbol{0}\}$, $\boldsymbol{g} \leftarrow \max\{\boldsymbol{\psi}(\boldsymbol{\beta}^\top \boldsymbol{S} - \boldsymbol{q}), \boldsymbol{0}\}$, $C \leftarrow \{(\boldsymbol{f}, \boldsymbol{g})\}$.
while $z > 0$ **do**
$\quad$ Solve the following upper level LP and get the optimal $(\boldsymbol{\gamma}^*, \boldsymbol{\delta}^*, L^*)$

$$
\begin{aligned}
\max_{\boldsymbol{\gamma},\boldsymbol{\delta},L} \quad & \boldsymbol{b}^\top \boldsymbol{\gamma} - \boldsymbol{a}^\top \boldsymbol{\delta} - L \\
\text{s.t.} \quad & \boldsymbol{\gamma}^\top \boldsymbol{f} - \boldsymbol{\delta}^\top \boldsymbol{g} \le L \qquad \forall (\boldsymbol{f}, \boldsymbol{g}) \in C
\end{aligned}
\tag{M.3U}
$$

$\quad$ Given $(\boldsymbol{\gamma}^*, \boldsymbol{\delta}^*, L^*)$, solve the following lower level MILP and get the optimal $(\boldsymbol{S}^*, \boldsymbol{f}^*, \boldsymbol{g}^*, \boldsymbol{I}^*, z^*)$

$$
\begin{aligned}
\max_{\boldsymbol{S},\boldsymbol{f},\boldsymbol{g},\boldsymbol{I}} \quad & z := \boldsymbol{\gamma}^\top \boldsymbol{f} - \boldsymbol{\delta}^\top \boldsymbol{g} - L \\
\text{s.t.} \quad & \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \ge \mathcal{M}(\mathcal{I}_m - 1) \\
& \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \le \mathcal{M}\mathcal{I}_m \\
& f_m \le \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) - \mathcal{M}(\mathcal{I}_m - 1) \\
& f_m \le \mathcal{M}\mathcal{I}_m \\
& \mathcal{I}_m \in \{0, 1\} \qquad \forall m \in \{1, \ldots, M\} \\
& g_n \ge \psi_n(\boldsymbol{\beta}_n^\top \boldsymbol{S} - q_n) \\
& g_n \ge 0 \qquad \forall n \in \{1, \ldots, N\}
\end{aligned}
\tag{M.3L}
$$

$\quad C \leftarrow C \cup (\boldsymbol{f}^*, \boldsymbol{g}^*)$, $z \leftarrow z^*$
return $\boldsymbol{\gamma}^*$ and $\boldsymbol{\delta}^*$

约束生成算法概述

- 约束集 C 包含定义不同约束的 $\boldsymbol{S} \in \mathbb{R}^{U}_{\geq 0}$ 值向量
- 从 C 为零向量开始（所有标的资产到期价格为零）
- 每次迭代：
  1. 上层优化（**M.3U**）：在当前约束集 C 下计算最优匹配
  2. 下层优化（**M.3L**）：找到最违反当前匹配的 $\boldsymbol{S}^{*}$
- 当下层返回目标值为零时终止

上层与下层优化

上层优化 **M.3U**:
给定约束集 C，求解线性规划（有 $|C|$ 个约束）:

$$\max_{\gamma,\boldsymbol{\delta},L} \ \boldsymbol{b}^\top \gamma - \boldsymbol{a}^\top \boldsymbol{\delta} - L \quad \text{s.t. } f(\boldsymbol{S}) - g(\boldsymbol{S}) \leq L, \ \forall \boldsymbol{S} \in C$$

下层优化 **M.3L**:
给定 $(\gamma^*, \boldsymbol{\delta}^*, L^*)$，求解混合整数线性规划:

$$\max_{\boldsymbol{S}\geq \mathbf{0}} \ f(\boldsymbol{S}) - g(\boldsymbol{S}) - L^*$$

生成最对抗性的 $\boldsymbol{S}^*$（最坏情况）。若目标值 > 0，将 $\boldsymbol{S}^*$ 加入 C；否则算法终止。

下层优化的线性化

引理

给定固定的 $\gamma, \boldsymbol{\delta}$ 和 L，*M.3L* 返回最违反 *M.2* 约束 *(2)* 的标的资产价值 $\boldsymbol{S}$。

证明思路：首先容易看出下面的公式返回最违反约束 (2) 的 $\boldsymbol{S}$，因为在最优解处我们将有最大可行的 f 和最小可行的 g。
即 $f_m = \max\{\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m), 0\}$ 和 $g_n = \max\{\psi_n(\boldsymbol{\beta}_n^\top \boldsymbol{S} - q_n), 0\}$。

$$\max_{\boldsymbol{S}, f, g} \quad \gamma^\top f - \boldsymbol{\delta}^\top g - L$$

$$\text{s.t.} \quad f_m \leq \max\{\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m), 0\} \qquad \forall m \in \{1, \ldots, M\}$$

$$g_n \geq \psi_n(\boldsymbol{\beta}_n^\top \boldsymbol{S} - q_n) \qquad \forall n \in \{1, \ldots, N\}$$

$$g_n \geq 0 \qquad \forall n \in \{1, \ldots, N\}$$

Big-M 技巧线性化 max 函数

剩下需要证明 M.3L 中关于任意买单 m 的约束集等价于 $f_m \leq \max\{\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m), 0\}$。

简而言之，M.3L 中的约束集通过对每个二元决策变量 $\mathcal{I}_m$ 使用 big-$\mathcal{M}$ 技巧来线性化 max 函数，其中 $\mathcal{M}$ 是一个大常数，例如 10^6。

考虑 $\mathcal{I}_m \in \{0, 1\}$ 的每种情况。前两个约束 $\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \geq \mathcal{M}(\mathcal{I}_m - 1)$ 和 $\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \leq \mathcal{M}\mathcal{I}_m$ 保证：

$$\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \geq 0 \quad \Longleftrightarrow \quad \mathcal{I}_m = 1$$

然后，遵循第三和第四个约束，即 $f_m \leq \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) - \mathcal{M}(\mathcal{I}_m - 1)$ 和 $f_m \leq \mathcal{M}\mathcal{I}_m$，我们有：

$$f_m \leq 0 \quad \text{if } \mathcal{I}_m = 0; \qquad f_m \leq \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \quad \text{if } \mathcal{I}_m = 1$$

线性化证明结论

因此，在最优解处，我们有：

$$f_m = 0 \quad \text{if } \mathcal{I}_m = 0; \qquad f_m = \phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m) \quad \text{if } \mathcal{I}_m = 1$$

这等价于 $f_m = \max\{\phi_m(\boldsymbol{\alpha}_m^\top \boldsymbol{S} - p_m), 0\}$。 □

因此，当 M.3L 返回目标值为零时，M.2 的约束 (2) 对所有 $\boldsymbol{S}$ 都满足，Algorithm 1 返回一个有效的匹配，从而优化总体利润。

算法正确性定理

定理

给定一个组合期权市场实例 $(\phi, \alpha, \mathbf{p}, \mathbf{b}, \psi, \beta, \mathbf{q}, \mathbf{a})$，*Algorithm 1* 返回 *M.2* 中定义的最优清算解。

算法保证：

- 算法通过迭代添加最违反的约束，逐步逼近真实的最优解
- 每次迭代都会找到当前解下最坏情况的 $\boldsymbol{S}^*$
- 当不存在违反约束的 $\boldsymbol{S}$ 时，算法终止并返回最优解
- 算法有限步终止，因为约束集 C 的增长受到 Sauer 引理的限制

5. Experiments

5.1. Real-Market Standard Financial Market

数据来源与实验设置

1. 数据来源:
   - 数据集：使用 OptionMetrics 提供的真实期权市场数据（通过 WRDS 获取）
   - 标的资产：选取了道琼斯工业平均指数（DJI）成分股中的 **30** 只股票
   - 时间点：2019 年 1 月 23 日
   - 规模：原始数据包含 **25,502** 个不同的期权市场（按标的资产、类型、行权价、到期日区分）
2. 实验方法:
   - 市场整合（**Consolidation**）：作者将原本分散的独立市场进行整合，不再按行权价区分市场，而是将同一标的资产、同一到期日的所有买单和卖单放在一起匹配。这样将 25,502 个市场缩减为 **366** 个整合市场。

实验方法与对比基准

运行机制：

- 在这些整合市场上运行机制 **M.1**（这是作者为标准期权设计的机制，是组合期权机制的特例）

对比基准：

- 对比了限制 $L = 0$（即不允许通过借贷/负债来获利）和不限制 L 的情况

3. 关键结果：

- 发现新交易：在 366 个整合市场中，有 **150** 个市场发现了原本独立市场设计下无法达成的盈利匹配
- 其中 **94** 个市场的 $L \geq 0$（即无风险纯套利），平均净利润 **$1.03**，最高 **$9.64**
- 剩余 **56** 个市场的 $L < 0$（即此时现金流为负，但未来收益能覆盖成本，相当于赚利息），平均利率 **0.7%**

引入 L 的影响

限制 $L = 0$ 的影响：

- 如果强制要求 $L = 0$，只能在 **74** 个市场中找到匹配（远少于 150 个）
- 证明了引入 L（允许跨期现金流管理）能显著增加交易机会

价差缩窄（**Bid-Ask Spread Reduction**）：

- 对于那些没有直接成交的市场，作者利用该机制计算了" 最具竞争力的买卖报价"
- 结果显示，买卖价差平均减少了 **73%**（从 80 美分降至 21 美分）
- 即便在限制 $L = 0$ 的情况下，价差也减少了 **52%**

意义：

- 市场整合机制能够显著提高流动性
- 降低交易成本，为投资者创造更好的交易环境

5.2. Synthetic Combinatorial Financial Market

合成订单生成流程

为了让数据逼真，作者并没有完全随机生成，而是基于真实数据校准：

- 组合结构：每个组合期权由 **2** 只股票 (S_i, S_j) 组成，从 U 个标的资产中随机选取。
- 权重 (w)：在 $\{\pm 1, ..., \pm 9\}$ 中随机选取互质整数。
- 行权价 (K)：基于真实股票的行权价 K_i, K_j 生成，公式为 $K = w_i K_i + w_j K_j$。
- 价格生成 (b, a)：
  - 先计算" 能覆盖该组合期权收益" 的一组标准期权的成本/收益，作为基准价格。
  - 加入噪声 (η)：为了模拟市场观点的分歧，在基准价格上加入噪声参数 η。噪声越大，买卖双方对价格的分歧越大，匹配的可能性通常越高。

评估指标

评估指标：

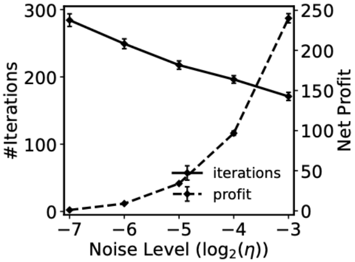
- 迭代次数 **(#Iterations)**： 即 Algorithm 1 生成约束的数量，代表计算复杂度。
- 净利润 **(Net Profit)**： 衡量机制的经济效益。

实验 A：改变价格噪声 (η)

设置：4 种标的资产 $(U = 4)$，150 个订单。

结果：

- 随着噪声 η 增加（市场分歧变大），净利润显著增加
- 同时算法需要的迭代次数也增加了（因为需要处理更多的潜在违约场景）



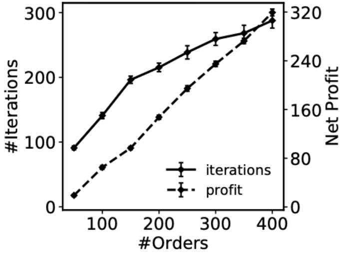
(a) Vary noise η added to order prices in markets with $U = 4$ and $n_{\text{orders}} = 150$.

实验 B：改变订单数量（n_{orders}）

设置：$U = 4$，噪声固定。订单数从 50 增加到 400。

结果：

- 净利润随订单数量线性增长（更多订单 = 更多机会）
- 迭代次数增长呈亚线性（**Sub-linear**）趋势。这说明算法具有良好的扩展性，生成的约束数量并没有随着订单量爆炸式增长。



(b) Vary number of orders n_{orders} in markets with $U = 4$ and $\eta = 2^{-4}$.

实验 C：改变标的资产数量 (U) 与"薄市场"问题

设置：订单数固定 150。U 从 4 增加到 20。

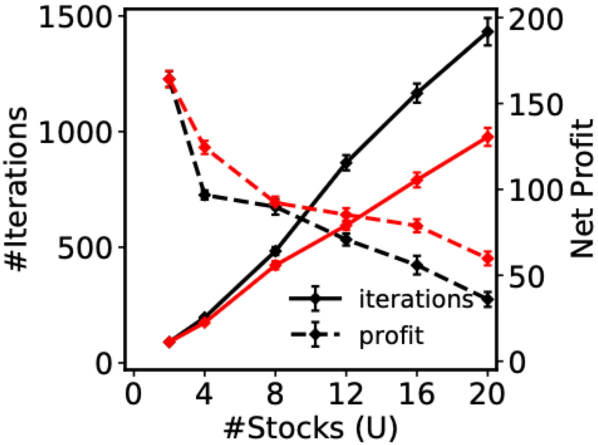
问题（黑色线条）：随着 U 增加，可能的资产组合对数 $\binom{U}{2}$ 指数级增长。这导致市场变"薄"（Thin Market），即买卖双方很难匹配到同一资产对进行交易。

结果：当 U 变大，利润下降，且迭代次数急剧上升（算法收敛难度增大）。

解决方案（红色线条 - **Prescriptive Set**）：

- 作者提出一种"规定集合"设计：不再允许交易任意资产组合，而是预先规定一组覆盖所有资产的组合对 $\mathcal{P}$（数量 $|\mathcal{P}| = U$）。
- 效果：如图中红线所示，这种设计显著缓解了薄市场问题，收敛速度更快，且利润更高。

实验 C 结果图示



(c) Vary size of underlying assets U in markets with $n_{\text{orders}} = 150$ and $\eta = 2^{-4}$.

The End