TaoGrid 网格策略

1. 策略总体概述

TaoGrid 是一套 以主动网格（Active Grid）为核心、结合趋势 Regime 与 S/R 区间的专业做市策略。区别于传统网格依赖固定区间与均匀 spacing，TaoGrid 采用：

- 区间来自 S/R（非固定公式）
- Spacing 与仓位受波动率、趋势、自适应引擎（DGT）动态调整
- 仓位结构不对称（非平均），边缘重、中枢轻
- 严格的全局风险预算控制
- 交易员介入模式（手动指定 Regime）
- 支持 mid-shift 的动态结构调整（类似 DGT 框架）

目标不是套利，而是 在震荡区间中，以“结构性优势 + 主动交易”的方式实现高频正期望。

2. 市场 Regime 判定

TaoGrid 不依赖自动化 regime 判定，而是允许交易员通过宏观判断或盘感，明确告诉系统当前市场所属状态：

- **震荡上行（UP Range）**
- **震荡中性（Neutral Range）**
- **震荡下行（DOWN Range）**

Regime 将直接影响：

- 买卖方向配比
- 仓位结构（70/30、50/50、30/70）
- mid shift 的方向性
- spacing 的轻微偏移
- 边界触发的响应方式

这是 TaoGrid 的核心思想：**量化执行 + 人类判断（Regime override）**。

1.1 模型推荐的趋势过滤器

指标类别	用法
趋势强度（ADX、MACD slope、EMA 排列）	ADX > 25 → 禁止网格（趋势过强）
动能指标（ROC、Z-score）	动能正负明显 → 禁止网格
波动率状态（ATR、BB宽度）	过高波动率 → 降低网格密度或暂停

1.2 模型推荐的大盘的Regime 判断逻辑

1.4.1 GREEN 候选（主多环境）

1). 1 日线条件：

代码块

```
dev_D = math.abs(close_D - ema200_D) / ema200_D
slope_D = ema200_D - nz(ema200_D[20]) // 200EMA 20 日斜率
green_daily = (close_D > ema200_D)
and (ema50_D > ema200_D)
and (slope_D > 0)
and (dev_D <  max_green_dev)
```

2). 4H 条件：连续若干根 4H 收盘 > 4H 200EMA：

代码块

```
var int green_4h_count = 0
green_4h_count := close_4H > ema200_4H ? green_4h_count + 1 : 0
bool green_4h_ok = green_4h_count >= green_4h_bars_req
bool cond_green_candidate = green_daily and green_4h_ok
```

3). 4H 条件：按日线统计连续天数并做确认

代码块

```
isNewDay = ta.change(time("D")) != 0
var int green_days = 0

if isNewDay
    green_days := cond_green_candidate ? green_days + 1 : 0
```

1.4.2 RED 候选（主空 / 高风险）

1）熊市型 RED：

```
代码块
1   slope_D_down = ema200_D - nz(ema200_D[20])
2   red_bear = (close_D < ema200_D) and (ema50_D < ema200_D) and (slope_D_down <
    0)
```

2). 疯牛末端型 RED：

```
代码块
1   red_bull = (close_D > ema200_D * (1 + red_bull_dev)) and (atr_D / close_D >
    atr_vol_thresh)
2   bool cond_red_candidate = red_bear or red_bull
```

3). 按日线统计：

```
代码块
1   var int red_days = 0
2   if isNewDay
3       red_days := cond_red_candidate ? red_days + 1 : 0
```

1.4.3 锁定机制与 Regime 状态变量

使用日线 time 计算锁定（防止频繁切换）：

```
代码块
1   var string regime = "YELLOW"
2   var int last_switch_day_time = time("D")
3   can_switch = time("D") - last_switch_day_time >= lock_days * 24 * 60 * 60 *
    1000
4
5   if isNewDay and can_switch
6       if green_days >= green_confirm_days
7           regime := "GREEN"  last_switch_day_time := time("D")
8       else if red_days >= red_confirm_days
9           regime := "RED"  last_switch_day_time := time("D")
10      else  regime := "YELLOW"
11      last_switch_day_time := time("D")
```

1.3 网格的方向

1. **中性**
   - 网格可以运行，但方向必须是双向的（long + short），仓位中性，不压方向
2. **震荡做空/ 做多**

- 靠近阻力
- 关键：不是 All-in，而是 grid size + zone-based sizing。

3. 网格区间的确定（支撑/阻力 & 波动率）

网格最关键的是区间，区间选得好，策略自然收益的好。

3.1 区间来自哪?

你可以结合：

- **技术分析：支撑/阻力（4H、1H）**
- **成交密集区（Volume Profile）**
- **波动率带（布林带、Keltner Channel）**
- **历史分布（Price Distribution / Percentile Bands）**

3.2 区间的确定方式

1. S 是下界，R 是上界（硬锚点): 区间基于你的人工判断。
2. **通过 ATR 做“扩展” → 避免被扫 （Volatility Cushion）**
   - Volatility Cushion = 0.5 ~ 1.2 ATR(24H)
3. **你的 S/R 决定网格的中心和左右分布:** 如果 S/R 不对，区间整体就不对。
4. **你的 S/R 决定趋势过滤结果**
   - 趋势偏多时不在上方卖

```
代码块
1   lower_bound = S - cushion
2   upper_bound = mid          # 不在上方布局卖单
```

   - 趋势偏空时不在下方买

```
代码块
1   upper_bound = R + cushion
2   lower_bound = mid          # 不在下方布局买单
```

   - 如果中性震荡 → 两边都开

```
代码块
1   lower_bound = S - cushion
```

```
2   upper_bound = R + cushion
```

5. **你的 S/R 决定仓位密度:** 间距来自 ATR，但层数＝由 S/R 的宽度决定。

3.3 网格间距

代码块

```
1   gap_% = min_return + maker_fee + k * volatility
```

其中：

- **min_return**：trader可以手动调，默认0.5%
- **maker_fee**：0.1% (根据交易所会有一个mapping)
- **volatility**：ATR-based 波动率
- **k**：波动安全因子（经验值 0.4~1.0）
  - 如果你想网格密一点，你可以把k调小一点

最终 gap = gap_% × mid_price

3.4 止损

方法一：硬止损

代码块

```
1   ↑ 假突破区（继续运行，不止损）
2   upper_bound = R
3   cushion zone = R + ATR
4   hard_stop = R + 2 ATR   ← 真止损
5
6   ————————————————————
7   mid
8   ————————————————————
9
10  hard_stop = S - 2 ATR   ← 真止损
11  cushion zone = S - ATR
12  lower_bound = S
13  ↓ 假跌破区（继续运行，不止损）
```

4. DGT （Dynamic Grid Trading）

4.1 DGT的优势

在趋势震荡行情中非常有帮助，价格结构/波动环境发生“持续性变化”时，网格可以动态重心 / 动态间距 / 动态方向，而不是死守原区间。

静态网格与 DGT 的本质区别

- **静态网格 Static Grid**
  mid 永远是 100，不管未来价格如何变化
  → 这意味着长期趋势会上移，你会在高位不断卖空（S1~S4），变成逆势交易。
- **动态网格 DGT**
  mid 会根据“价格重心、成交密度、趋势倾斜”自动调整 → 让网格永远跟着行情的中枢移动。

  为什么？因为市场的“中枢”不是固定的。

4.2 DGT 的风险点

风险点 1：错误的 regime 判定会被市场惩罚

如果交易员判断错误：

- 系统可能在低点重锚
- 而静态网格则会在回归中赚钱（高胜率）

换言之，DGT 在“假突破 + 快速反转”环境下容易损失部分利润。

风险点 2：不适合自动化

如果用机器自动识别：

- 模型会因为噪音误判 trend
- 导致频繁 mid shift
- 变成“追涨杀跌”的系统
- 最终产生累积亏损

风险点 3：mid shift 的滞后与提前——都可能造成盈利损失

- 提前 shift → 在错误位置重锚
- 滞后 shift → 逆势累积仓位

因此 mid shift 的边界和 timing 必须人工把控。

5. 网格区间内的仓位管理

大多数业余网格策略“只靠波动”，但专业网格策略**靠仓位管理来提升收益比和降低风险**。

5.1 格子层级权重（Level-wise Weighting）

5.1.1 核心思想：权重 = 函数(层级 i, regime)

我们把每一个买/卖格子对应的 size 写成：

```
买单：q_B(i) = Q_side_buy × w_B(i)
卖单：q_S(j) = Q_side_sell × w_S(j)

其中：
Q_side_buy = 为“买方向”分配的总名义资金（≤ Side_budget）
Q_side_sell = 为“卖方向”分配的总名义资金
w_B(i), w_S(j) = 各层的权重，要求：
Σ_i w_B(i) = 1
Σ_j w_S(j) = 1
```

后面所有 “怎么加仓、哪里多一点、哪里少一点”，都转化为设计这两个权重向量的问题。

5.1.2 中性震荡（NEUTRAL_RANGE）：边缘重、中间轻

你的思路其实很自然：在中性区间里：

```
靠近 S / R 的区域，性价比更高（更靠近极值）
靠近 mid 的区域，噪音更大，不值得太大仓位
```

可以设计一个简洁的线性权重：

假设买方向有 L_B 层（B1, B2, … B_LB），其中 B1 最靠近 mid，B_LB 最靠近 S。

```
定义：
raw_w_B(i) = 1 + k × (i − 1)，i=1 表示最上层，往下权重逐层增加
再做归一化：w_B(i) = raw_w_B(i) / Σ raw_w_B

示例：L_B = 4，k = 0.5：
B1: raw=1.0
B2: raw=1.5
B3: raw=2.0
B4: raw=2.5

归一化后 w_B：
```

```
B1 ≈ 1.0 / 7.0 ≈ 14%
B2 ≈ 21%
B3 ≈ 29%
B4 ≈ 36%
```

卖方向同理（S1 靠近 mid，S4 靠近 R，S4 权重最大）。

解释：

- 在中性区间，越靠近边界（支撑/阻力），越愿意用大一点的子弹。
- 越靠近中轴（mid）越谨慎（仓位小，防抖动）。

5.1.3 震荡上行（UP_RANGE）：下重上轻，偏多

在你已经人工判断为“震荡上行”的 regime 下：

- 下方买单更有价值 → 买侧权重更高
- 上方卖单主要是止盈/套利，不是主攻方向 → 卖侧权重稍微压低

可以这样设计：

a. **买卖侧总体权重倾斜：**

```
Q_side_buy = 0.7 × Risk_budget
Q_side_sell = 0.3 × Risk_budget
代表 “我是 better buyer”。
```

b. 买方层级分布：仍用“边缘重、中间轻”的线性或凸形分布。

c. 卖方层级分布：整体总量较小，而且也可做 “中间略重、最远稍轻”，避免在极高价持续累积 paljon 空仓。

简化：

你可以设：

- 中性：买=卖=50/50
- 上行：买=70/卖=30
- 下行：买=30/卖=70

5.1.4 震荡下行（DOWN_RANGE）：上重下轻，偏空

完全镜像上行：

```
```

```
Q_side_buy = 0.3 × Risk_budget
Q_side_sell = 0.7 × Risk_budget
```

卖侧：靠近阻力的一侧用更大的 size。

买侧：更保守，主要作为回补和套利用。

5.1.5 一个具体数值例子

假设：

- 总资金：`Capital = 100,000`
- Risk_budget = 30,000
- Regime = UP_RANGE
- 所以 Q_side_buy = 21,000；Q_side_sell = 9,000

买侧 4 层（B1–B4），用 14%/21%/29%/36% 的中性权重：

- B1：21,000 × 14% ≈ 2,940
- B2：≈ 4,410
- B3：≈ 6,090
- B4：≈ 7,560

再除以各层的价格，就是每一层应挂的数量（张数 / 币数），这里就不展开。

卖侧 3–4 层，用类似但总量只有 9,000。

这样一来：

- 区间内部每一格都有量，但最贴近支撑/阻力的一两层是你真正“愿意押注”的位置。
- 净敞口风格（多 / 空）由 regime 控制。

5.2 动态节流规则（Dynamic Throttling）

仅有静态权重还不够，还要有“动态制动”的规则——防止某一侧 inventory 累积过多，或者当日 PnL 已达标还在高频折腾。可以设计 3 条简单但非常实用的规则：

5.2.1 Inventory Limit（仓位上限）

定义：

代码块

```
Max_long_units = 允许未配对多仓的最大“层数总和”或名义金额
Max_short_units = 未配对空仓最大额度

当：
```

```
E_long ≥ Max_long_units → 暂停继续挂新的 buy 单（只保留已有的卖单用来减仓）。
E_short ≥ Max_short_units → 暂停新的 sell 单。
```

这在“击穿边界之前”的网格区间内，会有效防止“单侧越积越多”。

5.2.2 Profit Lock-in（盈利锁定）

当日 PnL 达到某一目标（例如 Risk_budget 的 1–2%）：

- 新挂单规模缩小 50%（所有 w_i 乘以 0.5）。
- 或直接停止新增挂单，仅保留现有未配对单做自然出清。

> 这符合你一直强调的：
> **达到日目标以后，保护当日成果，不再拼命。**

5.2.3 波动异常时的自动降频

如果短期 ATR 急剧上升（例如 1h ATR > 某阈值）：

- 系统临时提高 gap%（你已有），同时
- `Q_side_buy` 和 `Q_side_sell` 减半 or 降到三分之一。

> 直观理解：
> **“市场疯了的时候，人要缩手，网格也要缩手。”**

6. 风险控制

6.1 全局风险预算（Global Risk Budget）

限定本策略最大占用资金 / 保证金 / 最大名义敞口、最大单侧 inventory。

- 策略资金：`Capital`
- 单一品种网格风险预算：`Risk_budget = 0.3 × Capital`（示例 30%）
- 单侧最大敞口：`Side_budget = 0.6 × Risk_budget`
  - 即做多一侧的最大名义敞口不超过策略风险预算的 60%。
- 你可以再根据自己平时说的 “每笔风险 1–2%，单日最大损失 2000” 算出一个**最大浮亏阈值**。

```
记号上：
令 E_long = 当前所有买单方向的名义敞口（已成交但未配对的多头）
令 E_short = 当前所有卖单方向的名义敞口（已成交但未配对的空头）

```

```
约束：
E_long ≤ Side_budget
E_short ≤ Side_budget
```

- 浮亏 + 已实现亏损 ⩽ Daily_loss_limit（例如 2,000 美金）