

Idées d'amélioration — Optimisation des pools indexés sur halving *BTC* Date: 2025-08-21

1) Objectif

- Affiner les réglages Ichimoku + ATR par phase de cycle autour des événements de halving BTC.
- Construire des « pools » d'exploration *seeds, plages de paramètres* conditionnés par le régime spectral *lenteur/tendance vs bruit* et la phase post-halving.

2) Indexation « phase halving »

- Aligner l'axe du temps sur $t=0$ au halving pour chaque cycle; définir des fenêtres standard:
- Pré-halving $t \in [-180j, 0[$
- Post-halving I — « découverte » $t \in [0, +90j]$
- Post-halving II — « expansion » $t \in [+90j, +270j]$
- Post-halving III — « maturation » $t \in [+270j, +540j]$
- Pour chaque phase: agréger les métriques spectrales *LFP, entropie/flatness*, et en déduire des plages de paramètres cibles.

3) Heuristique Fourier → Ichimoku guidage des fenêtres

- Sur une fenêtre roulante *ex. 180–360 jours en $H2$* , calculer un spectre de puissance *PSD, Welchsidispo* et extraire:
- Période dominante $P = 1/f_*$ hors fréquence nulle
- Ratio basse fréquence: $LFP = \frac{\sum_{f < f_0} PSD(f)}{\sum_f PSD(f)}$
- Mapping *point de départ*:
- $kijun \approx P/2$, $tenkan \approx P/8 - P/6$, $senkou_b \approx P$, $shift$ inchangé (ou $\approx kijun/2$)
- Si $LFP > 0.6$ *marché lent/tendanciel*: augmenter $kijun$, $senkou_b$ et atr_mult 3–5
- Si entropie/flatness spectrale élevée *bruit*: resserrer $kijun$ 26–55, atr_mult 2–3, appliquer filtre cloud strict

4) Pools d'exploration et scheduler

- Définir des « pools » de seeds/plages par phase et régime:
- Pool Trend: $kijun \in [55, 100]$, $senkou_b \in [P*0.8, P*1.2]$, $atr_mult \in [3, 5]$
- Pool Mixte: $kijun \in [35, 70]$, $atr_mult \in [2.5, 4]$
- Pool Bruit: $kijun \in [26, 55]$, $atr_mult \in [2, 3]$, conditions d'entrée plus strictes

- Rotation des pools contrôlée par LFP/flatness *régime* et par phase halving *trelatif*.

5) Critères de sélection — « worst ou pas ? »

- Mode Robuste *worst-aware*:
- Maximiser $p5 \text{ MonteCarlofinal} \times$ sous contrainte $MDD_{\text{médiane}} \leq \text{seuil}$ et $\text{Calmar} \geq \min$
- Privilégier faible variance inter-seeds et stabilité $Lyapunov \leq 0$
- Mode Agressif *growth*:
- Maximiser p50 et CAGR sous $MDD \leq \text{plafond}$; vérifier drawdowns séquentiels et recovery
- Décision: choisir le mode par phase
ex. post-halvingI = mixte, post-halvingII = agressif, maturation = robuste

6) Accélération des backtests

- Utiliser convolution FFT pour moyennes roulantes *etlissagesATR* en $O(N \log N)$ quand beaucoup de fenêtres sont testées.
- Mutualiser les calculs d'indicateurs pour un lot de paramètres proches *reusedesTR/ATR et nuages partiels*.

7) KPI & instrumentation

- Suivre p5/p50/p95 *MC*, MDD médiane, entropie/flatness, LFP, variance inter-seeds, Latence simulée, Taux de fill, Slippage réalisé.
- Rapports par phase détachée *indexéet = 0halving* avec spectres moyens et plages retenues.

8) Roadmap proposée

1) Implémenter utilitaires Fourier *PSD*, *LFP*, *flatness* et comparaisons par phase **Misplaced &**. 2) Intégrer un « suggesteur » de paramètres *tenkan/kijun/senkou_b/atr_mult* pour générer un JSON baseline par symbole. 3) Étendre le scheduler d'exploration: pools conditionnés par *phase*, *LFP*, *flatness*. 4) Walk-forward par phase + Monte Carlo pour fixer seuils LFP et ranges finaux. 5) Bench: gains CPU via convolution FFT pour lots de fenêtres; définir les tailles de lots. 6) Décider « worst ou pas » par phase à partir des distributions *MC* et de la variance inter-seeds.

9) Limites & garde-fous

- Non-stationarité: préférer fenêtres roulantes, STFT/ondelettes si besoin de localisation temporelle.
- Échantillonnage: gérer trous *Lomb-Scargles* irrégulier.
- Fuites/sur-réglage: valider par walk-forward et variance inter-seeds; fixer les seuils via données OOS.

Ressources liées: docs/FOURIER_STRATEGIE_FR.md, docs/FORMULES_ET_EXEMPLES.md, docs/HSBC_REPORT_FR.md.