

Routage optimal dans les écosystèmes AMM

Optimal Routing in AMM-based Ecosystems

TER (M1) — Encadrant : Sebastian Müller
sebastian.muller@univ-amu.fr

Résumé

Ce mémoire étudie la modélisation mathématique du routage optimal dans les réseaux de *Constant Function Market Makers* (CFMM). Après une introduction au modèle Uniswap V2 et aux notions de prix marginal, de prix effectif, de slippage et d'arbitrage local, le routage optimal est formulé comme un problème d'optimisation. La relaxation des contraintes de trading permet d'obtenir un programme convexe et, sous des hypothèses appropriées de croissance et de saturation, cette relaxation est exacte. On montre ensuite que la détection d'arbitrage s'interprète comme un cas particulier du même cadre d'optimisation. Enfin, une application numérique sur un sous-réseau de pools Uniswap V2 illustre l'intérêt pratique du routage optimal par rapport au choix d'une route unique.

Sommaire

1	Introduction	3
1.1	Le cas Uniswap V2 : invariant de produit constant	3
1.2	Prix marginal, prix effectif et slippage	4
1.3	Arbitrage local et efficacité des prix	4
1.4	Du pool isolé au réseau de pools	4
2	Modélisation mathématique d'un réseau de CFMM	5
2.1	Réseau de m CFMM et n tokens	5
2.2	Indexation locale/globale et matrices A_i	5
2.3	Paniers fournis et reçus	6
2.4	Fonction de trading et frais	7
2.5	Vecteur de transaction réseau	8
2.6	Fonction d'utilité du trader	8
3	Problème de routage optimal	9
3.1	Formulation avec contraintes d'égalité	9
3.2	Relaxation convexe	9
3.3	Exactitude de la relaxation	10
3.4	Exemple : liquidation d'un token vers un autre	11
3.5	Extension : coûts fixes et problème MICP	12
4	Arbitrage et équilibre du marché	12
4.1	Définition d'un arbitrage	12
4.2	Détection d'arbitrage par routage optimal	13
4.3	Conditions de non-arbitrage	13
4.4	Interprétation : prix globaux, prix locaux et bande de non-arbitrage	14
4.5	Limites économiques : frais, liquidité, gaz, exécution	15
5	Application numérique : routage optimal sur un sous-réseau de pools	15
6	Conclusion	16

1 Introduction

Les échanges décentralisés permettent d'échanger des actifs numériques à l'aide de contrats intelligents, sans intermédiaire centralisé. Sur une blockchain, la reproduction directe d'un carnet d'ordres est coûteuse, car elle exige de stocker, d'annuler et de mettre à jour de nombreux ordres. Les *Automated Market Makers* (AMM) proposent une autre architecture : les utilisateurs échangent contre des réserves de liquidité, et les prix sont déterminés par une règle explicite dépendant des réserves.

On s'intéresse ici aux *Constant Function Market Makers* (CFMM). Un CFMM est défini par une fonction de trading appliquée aux réserves du pool. Une transaction est admissible si elle conserve, à frais près, un certain niveau de cette fonction. Le cas canonique est Uniswap V2, dont l'invariant protocolaire est le produit constant [4, 2].

L'objet de ce mémoire est le problème de routage optimal. Lorsqu'un même échange peut être exécuté à travers plusieurs pools, une question naturelle apparaît : comment répartir les flux entre les différentes routes disponibles afin d'obtenir la meilleure exécution possible ? Cette question est particulièrement importante lorsque la taille de l'ordre est significative relativement aux réserves, car le prix d'exécution dépend alors de la quantité échangée.

1.1 Le cas Uniswap V2 : invariant de produit constant

Considérons un pool composé de deux actifs α et β , de réserves

$$R_\alpha > 0, \quad R_\beta > 0.$$

Dans Uniswap V2, l'invariant protocolaire est

$$R_\alpha R_\beta = k.$$

Si un utilisateur apporte une quantité $\Delta_\beta > 0$ de l'actif β pour recevoir une quantité $\Lambda_\alpha > 0$ de l'actif α , les réserves après échange deviennent

$$R_\beta + \Delta_\beta, \quad R_\alpha - \Lambda_\alpha.$$

Sans frais, la condition de produit constant est donc

$$(R_\alpha - \Lambda_\alpha)(R_\beta + \Delta_\beta) = R_\alpha R_\beta.$$

Avec frais proportionnels, on note $\gamma \in (0, 1]$ la fraction du montant entrant effectivement prise en compte dans l'invariant. Pour Uniswap V2, les frais de 0,3% correspondent à $\gamma = 0,997$. La contrainte devient

$$(R_\alpha - \Lambda_\alpha)(R_\beta + \gamma\Delta_\beta) = R_\alpha R_\beta.$$

On en déduit

$$\Lambda_\alpha = \frac{R_\alpha \gamma \Delta_\beta}{R_\beta + \gamma \Delta_\beta}.$$

La sortie obtenue n'est donc pas linéaire en la taille de l'ordre [2].

Il faut distinguer l'invariant protocolaire $R_\alpha R_\beta = k$ de la fonction de trading retenue pour l'analyse convexe. Dans la suite, le même invariant est représenté par

$$\varphi(R_\alpha, R_\beta) = \sqrt{R_\alpha R_\beta}.$$

Cette fonction a les mêmes ensembles de niveau que le produit $R_\alpha R_\beta$, car la racine carrée est strictement croissante. Contrairement au produit brut, elle est concave sur \mathbb{R}_+^2 , propriété essentielle pour obtenir une formulation convexe.

1.2 Prix marginal, prix effectif et slippage

Le prix marginal est le taux d'échange infinitésimal. Dans le cas sans frais, le prix marginal d'une unité infinitésimale de α , exprimé en unités de β , est

$$p_\alpha^\beta = \frac{R_\beta}{R_\alpha}.$$

Cette notation signifie qu'il faut, marginalement, R_β/R_α unités de β pour obtenir une unité de α .

Pour un ordre de taille finie, le ratio d'exécution est différent. Si l'utilisateur fournit Δ_β et reçoit Λ_α , le nombre d'unités de α reçues par unité de β fournie est

$$P^{\text{eff}}(\Delta_\beta) = \frac{\Lambda_\alpha}{\Delta_\beta} = \frac{R_\alpha \gamma}{R_\beta + \gamma \Delta_\beta}.$$

Ce ratio décroît avec Δ_β . Le *slippage* désigne cette dégradation de l'exécution lorsque la taille de l'ordre devient significative relativement aux réserves.

1.3 Arbitrage local et efficacité des prix

Le prix d'un AMM dépend uniquement de ses réserves. Il peut donc s'écarter d'un prix de référence externe. Les arbitrageurs corrigent ces écarts en achetant l'actif sous-évalué et en vendant l'actif surévalué, ce qui modifie les réserves et rapproche le prix du pool du prix externe.

Notons m_p le prix externe de l'actif α exprimé en unités de β , et m_u le prix marginal implicite du pool, exprimé dans la même unité. Acheter marginalement α dans le pool en payant β coûte m_u/γ lorsque les frais sont appliqués sur le montant entrant. L'arbitrage consistant à acheter dans le pool puis à revendre au prix externe est profitable si

$$m_p > \frac{m_u}{\gamma},$$

c'est-à-dire

$$m_u < \gamma m_p.$$

Dans l'autre sens, vendre marginalement α au pool rapporte γm_u unités de β . Cette opération est profitable si

$$\gamma m_u > m_p,$$

c'est-à-dire

$$m_u > \gamma^{-1} m_p.$$

L'absence d'arbitrage local impose donc

$$\gamma m_p \leq m_u \leq \gamma^{-1} m_p.$$

Les frais créent ainsi une bande de non-arbitrage [2].

1.4 Du pool isolé au réseau de pools

L'analyse précédente concerne un pool isolé. Or un écosystème DeFi est un réseau de pools reliant plusieurs actifs. Un échange de WETH vers DAI peut être exécuté directement via un pool WETH/DAI ou indirectement via

$$\text{WETH} \longrightarrow \text{USDC} \longrightarrow \text{DAI}.$$

Il peut aussi être optimal de fractionner l'ordre entre plusieurs routes afin de réduire le slippage total.

Le problème étudié est donc le suivant : étant donné un réseau de CFMM, comment choisir les transactions locales à effectuer dans chaque pool afin de maximiser l'utilité finale du trader ? Sous des hypothèses de croissance et de concavité des fonctions de trading, ce problème admet une relaxation convexe exacte [1]. La même formulation permet également d'interpréter la détection d'arbitrage comme un cas particulier du routage optimal.

2 Modélisation mathématique d'un réseau de CFMM

2.1 Réseau de m CFMM et n tokens

On considère un réseau composé de m CFMM, indexés par

$$i = 1, \dots, m,$$

et d'un ensemble global de n tokens, indexés par

$$j = 1, \dots, n.$$

Le CFMM i ne contient qu'un sous-ensemble de n_i tokens, avec

$$2 \leq n_i \leq n.$$

Cette structure peut être représentée par un graphe biparti : un ensemble de sommets représente les tokens, l'autre les CFMM, et une arête relie un token à un pool lorsque ce token appartient aux réserves du pool [1].

Par exemple, pour l'univers

$$\{\text{WETH}, \text{USDC}, \text{DAI}, \text{WBTC}\},$$

on peut considérer les pools

$$\text{WETH/USDC}, \quad \text{WETH/DAI}, \quad \text{USDC/DAI}, \quad \text{WETH/WBTC}.$$

Un même échange WETH vers DAI peut alors utiliser la route directe WETH/DAI ou la route indirecte WETH \rightarrow USDC \rightarrow DAI. La modélisation doit donc décrire simultanément les flux dans plusieurs pools, ce qui impose de distinguer une indexation locale propre à chaque pool et une indexation globale commune au réseau.

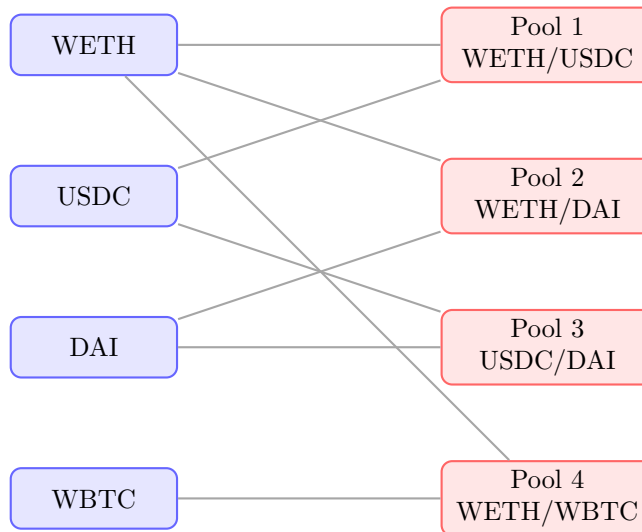


FIGURE 1 – Représentation bipartite d'un réseau de CFMM. Les sommets de gauche représentent les tokens, ceux de droite les pools. Une arête indique qu'un token appartient aux réserves du pool.

2.2 Indexation locale/globale et matrices A_i

Chaque pool possède une indexation locale de ses tokens. L'espace local du CFMM i est donc \mathbb{R}^{n_i} , tandis que l'espace global du réseau est \mathbb{R}^n .

On introduit une matrice

$$A_i \in \mathbb{R}^{n \times n_i}$$

qui envoie les coordonnées locales du pool i vers les coordonnées globales du réseau. Si le k -ième token local du CFMM i correspond au j -ième token global, alors

$$(A_i)_{jk} = 1,$$

et les autres coefficients de la colonne correspondante sont nuls.

Par construction, chaque colonne de A_i est un vecteur de la base canonique de \mathbb{R}^n . Plus précisément, si le ℓ -ième token local du pool i correspond au token global $j(i, \ell)$, alors

$$A_i e_\ell = e_{j(i, \ell)},$$

où e_ℓ désigne le ℓ -ième vecteur de base de \mathbb{R}^{n_i} et $e_{j(i, \ell)}$ le vecteur de base correspondant de \mathbb{R}^n .

Par exemple, avec l'indexation globale

$$(\text{WETH}, \text{USDC}, \text{DAI}, \text{WBTC})$$

et l'indexation locale

$$(\text{WETH}, \text{DAI})$$

pour le pool WETH/DAI, on a

$$A_{\text{WETH/DAI}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Ainsi,

$$A_{\text{WETH/DAI}} \begin{pmatrix} x_{\text{WETH}} \\ x_{\text{DAI}} \end{pmatrix} = \begin{pmatrix} x_{\text{WETH}} \\ 0 \\ x_{\text{DAI}} \\ 0 \end{pmatrix}.$$

2.3 Paniers fournis et reçus

Une transaction locale dans le CFMM i est décrite par deux vecteurs :

$$\Delta_i \in \mathbb{R}_+^{n_i}, \quad \Lambda_i \in \mathbb{R}_+^{n_i}.$$

Le panier Δ_i est fourni au pool par le trader, tandis que Λ_i est reçu en retour. Le gain net local du trader est donc

$$\Lambda_i - \Delta_i.$$

Exprimé dans l'espace global, ce gain devient

$$A_i(\Lambda_i - \Delta_i).$$

Dans un pool WETH/DAI, un swap WETH vers DAI s'écrit par exemple

$$\Delta_i = \begin{pmatrix} \Delta_{\text{WETH}} \\ 0 \end{pmatrix}, \quad \Lambda_i = \begin{pmatrix} 0 \\ \Lambda_{\text{DAI}} \end{pmatrix}.$$

Cette notation vectorielle couvre les swaps à deux actifs et les transactions multi-actifs.

2.4 Fonction de trading et frais

Le CFMM i est défini par une fonction de trading

$$\varphi_i : \mathbb{R}_+^{n_i} \rightarrow \mathbb{R},$$

appliquée aux réserves locales

$$R_i \in \mathbb{R}_+^{n_i}.$$

Après une transaction (Δ_i, Λ_i) , les réserves pertinentes pour la contrainte de trading sont

$$R_i + \gamma_i \Delta_i - \Lambda_i,$$

où

$$\gamma_i \in (0, 1]$$

est le paramètre de frais. La contrainte relaxée s'écrit

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i).$$

L'égalité correspond au cas où la transaction se situe exactement sur la frontière admissible du CFMM. L'inégalité autorise une relaxation mathématique : le pool peut conserver une valeur de fonction de trading supérieure à son niveau initial.

Pour Uniswap V2, le protocole impose l'invariant

$$R_1 R_2 = k.$$

Pour l'analyse convexe, on retient la fonction de trading

$$\varphi(R_1, R_2) = \sqrt{R_1 R_2}.$$

Elle a les mêmes ensembles de niveau que le produit, mais elle est concave.

Vérifions-le directement. Pour $R_1, R_2 > 0$,

$$\varphi(R_1, R_2) = R_1^{1/2} R_2^{1/2}.$$

On a

$$\frac{\partial \varphi}{\partial R_1} = \frac{1}{2} R_1^{-1/2} R_2^{1/2}, \quad \frac{\partial \varphi}{\partial R_2} = \frac{1}{2} R_1^{1/2} R_2^{-1/2}.$$

La hessienne est

$$\nabla^2 \varphi(R_1, R_2) = \frac{1}{4} \begin{pmatrix} -R_1^{-3/2} R_2^{1/2} & R_1^{-1/2} R_2^{-1/2} \\ R_1^{-1/2} R_2^{-1/2} & -R_1^{1/2} R_2^{-3/2} \end{pmatrix}.$$

Pour tout $z = (z_1, z_2) \in \mathbb{R}^2$,

$$z^\top \nabla^2 \varphi(R_1, R_2) z = -\frac{1}{4} \left(\frac{R_2^{1/4}}{R_1^{3/4}} z_1 - \frac{R_1^{1/4}}{R_2^{3/4}} z_2 \right)^2 \leq 0.$$

La hessienne est donc négative semi-définie sur \mathbb{R}_{++}^2 , ce qui prouve la concavité de φ sur l'intérieur du domaine. Comme \mathbb{R}_{++}^2 est dense dans \mathbb{R}_+^2 et que φ est continue sur \mathbb{R}_+^2 , la concavité s'étend au bord. En effet, pour $x, y \in \mathbb{R}_+^2$ et $\theta \in [0, 1]$, on peut choisir des suites $x_n, y_n \in \mathbb{R}_{++}^2$ telles que $x_n \rightarrow x$ et $y_n \rightarrow y$. La concavité sur l'intérieur donne

$$\varphi(\theta x_n + (1 - \theta) y_n) \geq \theta \varphi(x_n) + (1 - \theta) \varphi(y_n),$$

puis le passage à la limite donne la même inégalité pour x et y . Par continuité, cette concavité s'étend à \mathbb{R}_+^2 .

Un pool Balancer peut être modélisé par une moyenne géométrique pondérée [5] :

$$\varphi(R) = \prod_{k=1}^{n_i} R_k^{w_k}, \quad w_k > 0, \quad \sum_{k=1}^{n_i} w_k = 1.$$

Cette fonction est concave sur $\mathbb{R}_+^{n_i}$: c'est la concavité classique de la moyenne géométrique pondérée.

Dans la suite, les fonctions φ_i seront supposées continues, concaves et croissantes composante par composante, c'est-à-dire que

$$x \leq y \implies \varphi_i(x) \leq \varphi_i(y),$$

où l'inégalité est comprise composante par composante.

2.5 Vecteur de transaction réseau

Le vecteur de transaction réseau est

$$\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i) \in \mathbb{R}^n.$$

Sa composante Ψ_j représente la variation nette du token global j dans le portefeuille du trader : $\Psi_j > 0$ correspond à une réception nette, $\Psi_j < 0$ à une fourniture nette, et $\Psi_j = 0$ à une absence de position nette.

Par exemple, une liquidation pure de WETH vers DAI dans l'indexation

$$(\text{WETH}, \text{USDC}, \text{DAI}, \text{WBTC})$$

peut être imposée par un bilan final

$$\Psi = \begin{pmatrix} -\Delta_{\text{WETH}} \\ 0 \\ \Lambda_{\text{DAI}} \\ 0 \end{pmatrix}.$$

La composante USDC est nulle si l'on impose que l'USDC soit seulement un actif intermédiaire. Cette nullité n'est pas automatique : elle exprime une contrainte de bilan final choisie par le trader.

2.6 Fonction d'utilité du trader

Pour optimiser une transaction, on introduit une fonction d'utilité

$$U : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}.$$

Le trader choisit les transactions locales (Δ_i, Λ_i) afin de maximiser $U(\Psi)$.

Pour une liquidation d'un token a vers un token b , on prend typiquement

$$U(\Psi) = e_b^\top \Psi,$$

avec la contrainte d'entrée

$$e_a^\top \Psi = -q,$$

où $q > 0$ est la quantité du token a fournie au réseau. Plus généralement, une utilité linéaire

$$U(\Psi) = \pi^\top \Psi, \quad \pi \in \mathbb{R}_+^n,$$

représente une valorisation externe des tokens. Dans la suite, U est supposée concave et croissante composante par composante.

Remarque 2.1. *Dans le cas de liquidation pure, l'utilité $U(\Psi) = e_b^\top \Psi$ est croissante mais non strictement croissante sur \mathbb{R}^n , car elle ne dépend que de la composante b . Cette distinction sera importante pour l'exactitude de la relaxation convexe.*

3 Problème de routage optimal

On dispose maintenant des objets nécessaires : les variables locales (Δ_i, Λ_i) , les matrices A_i , le bilan global Ψ et l'utilité U .

Hypothèses générales. Pour tout CFMM i , on suppose que :

- $\varphi_i : \mathbb{R}_+^{n_i} \rightarrow \mathbb{R}$ est continue, concave et croissante composante par composante ;
- $R_i \in \mathbb{R}_{++}^{n_i}$;
- les réserves post-transaction vérifient explicitement

$$R_i + \gamma_i \Delta_i - \Lambda_i \in \mathbb{R}_+^{n_i};$$

- $U : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}$ est concave et croissante composante par composante.

Pour les résultats faisant intervenir les prix marginaux, on supposera en outre que chaque φ_i est différentiable en R_i , avec $\nabla \varphi_i(R_i) \in \mathbb{R}_{++}^{n_i}$. Pour l'exactitude stricte de la relaxation, on supposera soit que U est strictement croissante, soit que l'on sélectionne une solution optimale saturant les contraintes de trading.

3.1 Formulation avec contraintes d'égalité

Les variables sont

$$\Delta_i, \Lambda_i \in \mathbb{R}_+^{n_i}, \quad i = 1, \dots, m,$$

et le vecteur auxiliaire est

$$\Psi \in \mathbb{R}^n.$$

Le vecteur Ψ pourrait être éliminé en substituant

$$\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i)$$

dans l'objectif. On le conserve afin d'exprimer directement les contraintes globales de liquidation ou d'arbitrage.

La formulation naturelle du routage optimal est

$$\begin{aligned} & \text{maximiser} && U(\Psi) \\ & \text{sous contraintes} && \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\ & && \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) = \varphi_i(R_i), \quad i = 1, \dots, m, \\ & && R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\ & && \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

Cette formulation n'est pas convexe en général, car les contraintes d'égalité non linéaires définissent rarement des ensembles convexes.

3.2 Relaxation convexe

On remplace les égalités par des inégalités :

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i).$$

Le problème relaxé est

$$\begin{aligned}
& \text{maximiser} && U(\Psi) \\
& \text{sous contraintes} && \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\
& && \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \quad i = 1, \dots, m, \\
& && R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\
& && \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m.
\end{aligned}$$

Proposition 3.1 (Convexité du problème relaxé). *Sous les hypothèses précédentes, le problème relaxé est un problème d'optimisation convexe.*

Démonstration. La contrainte définissant le bilan global,

$$\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i),$$

est affine. Les contraintes

$$\Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad R_i + \gamma_i \Delta_i - \Lambda_i \geq 0$$

sont convexes.

Pour chaque i , l'application

$$(\Delta_i, \Lambda_i) \mapsto R_i + \gamma_i \Delta_i - \Lambda_i$$

est affine. Comme φ_i est concave, l'application composée

$$(\Delta_i, \Lambda_i) \mapsto \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i)$$

est concave. Par conséquent, son ensemble de sur-niveau

$$\{(\Delta_i, \Lambda_i) : \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i)\}$$

est convexe [3, Sec. 3.1].

L'ensemble admissible du problème relaxé est donc convexe. Enfin, l'objectif consiste à maximiser la fonction concave U sur cet ensemble convexe ; de manière équivalente, on minimise la fonction convexe $-U$. Le problème relaxé est donc un problème d'optimisation convexe. \square

Remarque 3.1. *La transaction nulle, définie par $\Delta_i = \Lambda_i = 0$ pour tout i , est admissible pour le problème relaxé général. En revanche, lorsque des contraintes supplémentaires de liquidation sont imposées, comme $e_a^\top \Psi = -q$, la faisabilité dépend de la connectivité du réseau, des réserves disponibles et de la taille de l'ordre.*

3.3 Exactitude de la relaxation

Proposition 3.2 (Exactitude de la relaxation). *Supposons que chaque φ_i soit continue, concave et croissante composante par composante, que U soit strictement croissante composante par composante, et que les réserves post-transaction soient strictement positives à l'optimum. Alors toute solution optimale du problème relaxé sature les contraintes de trading :*

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) = \varphi_i(R_i), \quad i = 1, \dots, m.$$

Démonstration. Soit $(\Delta_i, \Lambda_i, \Psi)_{i=1}^m$ une solution optimale du problème relaxé. Supposons par contradiction qu'il existe un pool k tel que

$$\varphi_k(R_k + \gamma_k \Delta_k - \Lambda_k) > \varphi_k(R_k).$$

Posons

$$\tilde{R}_k = R_k + \gamma_k \Delta_k - \Lambda_k.$$

Par hypothèse, $\tilde{R}_k \in \mathbb{R}_{++}^{n_k}$.

Choisissons une coordonnée $\ell \in \{1, \dots, n_k\}$. Comme \tilde{R}_k est strictement positif, il existe $\eta > 0$ tel que

$$\tilde{R}_k - \varepsilon e_\ell \in \mathbb{R}_+^{n_k} \quad \text{pour tout } \varepsilon \in [0, \eta].$$

Par continuité de φ_k , et comme l'inégalité est stricte en $\varepsilon = 0$, il existe $0 < \varepsilon \leq \eta$ tel que

$$\varphi_k(\tilde{R}_k - \varepsilon e_\ell) \geq \varphi_k(R_k).$$

Définissons alors

$$\hat{\Lambda}_k = \Lambda_k + \varepsilon e_\ell,$$

et laissons inchangées toutes les autres variables. La contrainte de trading du pool k reste vérifiée, car

$$R_k + \gamma_k \Delta_k - \hat{\Lambda}_k = \tilde{R}_k - \varepsilon e_\ell.$$

Le nouveau bilan global est

$$\hat{\Psi} = \Psi + A_k(\hat{\Lambda}_k - \Lambda_k) = \Psi + \varepsilon A_k e_\ell.$$

Comme $A_k e_\ell$ est un vecteur canonique de l'espace global, on a

$$\hat{\Psi} \geq \Psi, \quad \hat{\Psi} \neq \Psi.$$

Puisque U est strictement croissante composante par composante,

$$U(\hat{\Psi}) > U(\Psi),$$

ce qui contredit l'optimalité de $(\Delta_i, \Lambda_i, \Psi)_{i=1}^m$. Toutes les contraintes de trading sont donc saturées à l'optimum. \square

Lorsque U est seulement croissante, on peut introduire une perturbation strictement croissante

$$U_\varepsilon(\Psi) = U(\Psi) + \varepsilon \mathbf{1}^\top \Psi, \quad \varepsilon > 0.$$

Sous des hypothèses assurant l'existence d'optima et la compacité de l'ensemble des solutions pertinentes, les solutions optimales des problèmes perturbés permettent, lorsque $\varepsilon \rightarrow 0$, de sélectionner une solution optimale du problème initial qui sature les contraintes de trading. Dans ce mémoire, cette procédure sera utilisée comme principe de sélection lorsque l'utilité considérée n'est pas strictement croissante. Cette remarque est particulièrement utile pour les problèmes de liquidation pure, où l'utilité linéaire $e_b^\top \Psi$ n'est pas strictement croissante dans toutes les composantes.

3.4 Exemple : liquidation d'un token vers un autre

Soient a le token d'entrée, b le token cible et $q > 0$ la quantité fournie. En notant e_a, e_b les vecteurs de base associés, la liquidation s'écrit

$$e_a^\top \Psi = -q, \quad U(\Psi) = e_b^\top \Psi.$$

On impose également l'absence de déficit net sur les autres tokens :

$$\Psi_j \geq 0, \quad j \neq a, b.$$

Le problème devient

$$\begin{aligned} & \text{maximiser} && e_b^\top \Psi \\ & \text{sous contraintes} && \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\ & && \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \quad i = 1, \dots, m, \\ & && R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\ & && e_a^\top \Psi = -q, \\ & && \Psi_j \geq 0, \quad j \neq a, b, \\ & && \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

Cette formulation permet de fractionner l'ordre entre plusieurs routes. Ainsi, un échange WETH vers DAI peut combiner la route directe WETH \rightarrow DAI et la route indirecte WETH \rightarrow USDC \rightarrow DAI.

Remarque 3.2. Dans ce problème, l'utilité $e_b^\top \Psi$ ne valorise que la sortie en token b . L'exactitude stricte de la relaxation ne découle donc pas directement de la proposition précédente. On peut soit ajouter une perturbation infinitésimale $\varepsilon \mathbf{1}^\top \Psi$, soit sélectionner parmi les solutions optimales une solution saturant les contraintes de trading.

3.5 Extension : coûts fixes et problème MICP

Les frais proportionnels sont intégrés par γ_i , mais les coûts fixes d'exécution, comme le gaz, ne le sont pas. On introduit un coût fixe $q_i \geq 0$ et une variable binaire

$$\eta_i \in \{0, 1\},$$

valant 1 si le pool i est utilisé.

L'objectif devient

$$U(\Psi) - q^\top \eta.$$

Avec des bornes Δ_i^{\max} et Λ_i^{\max} , on impose

$$0 \leq \Delta_i \leq \eta_i \Delta_i^{\max}, \quad 0 \leq \Lambda_i \leq \eta_i \Lambda_i^{\max}.$$

Les bornes sur les sorties peuvent être choisies à partir des réserves ; celles sur les entrées peuvent provenir du portefeuille initial ou d'une taille maximale d'ordre.

Le problème avec coûts fixes est

$$\begin{aligned} & \text{maximiser} && U(\Psi) - q^\top \eta \\ & \text{sous contraintes} && \Psi = \sum_{i=1}^m A_i (\Lambda_i - \Delta_i), \\ & && \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \quad i = 1, \dots, m, \\ & && R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\ & && 0 \leq \Delta_i \leq \eta_i \Delta_i^{\max}, \quad 0 \leq \Lambda_i \leq \eta_i \Lambda_i^{\max}, \quad i = 1, \dots, m, \\ & && \eta_i \in \{0, 1\}, \quad i = 1, \dots, m. \end{aligned}$$

C'est un programme convexe en nombres entiers mixtes, ou *Mixed-Integer Convex Program* (MICP) [1].

4 Arbitrage et équilibre du marché

4.1 Définition d'un arbitrage

Définition 4.1 (Arbitrage vectoriel). Une transaction réseau admissible est un arbitrage si son bilan global Ψ vérifie

$$\Psi \geq 0, \quad \Psi \neq 0.$$

Autrement dit, le trader ne fournit aucun token net au réseau et reçoit une quantité strictement positive d'au moins un token.

Cette définition est indépendante du choix d'un numéraire et suit le cadre d'Angeris, Chitra, Evans et Boyd [1].

Remarque 4.1. Cette définition est plus forte qu'une condition de profitabilité exprimée relativement à un vecteur de prix externe. Si un vecteur $g \in \mathbb{R}_{++}^n$ est fixé, une condition plus faible serait

$$g^\top \Psi > 0.$$

La notion retenue ici est vectorielle : elle impose un gain net non négatif dans chaque token et strictement positif dans au moins un token.

4.2 Détection d'arbitrage par routage optimal

On choisit une utilité strictement croissante sur \mathbb{R}_+^n , par exemple

$$U(\Psi) = \mathbf{1}^\top \Psi,$$

et on impose $\Psi \geq 0$. Le problème devient

$$\begin{aligned} & \text{maximiser} && U(\Psi) \\ & \text{sous contraintes} && \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\ & && \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \quad i = 1, \dots, m, \\ & && R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\ & && \Psi \geq 0, \\ & && \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

La solution nulle est admissible. Si l'optimum vérifie

$$U(\Psi^*) > U(0),$$

alors

$$\Psi^* \geq 0, \quad \Psi^* \neq 0,$$

et il existe un arbitrage. Inversement, tout arbitrage donne une valeur strictement supérieure à $U(0)$ pour toute utilité strictement croissante.

4.3 Conditions de non-arbitrage

Pour chaque CFMM i , on suppose ici que φ_i est différentiable en R_i , et l'on note

$$P_i = \nabla \varphi_i(R_i).$$

Ce vecteur donne les prix marginaux locaux à un facteur d'échelle près. Le vecteur $A_i^\top g$ correspond aux prix globaux g restreints aux tokens du pool i .

Proposition 4.1 (Condition de non-arbitrage). *Le résultat de non-arbitrage d'Angeris, Chitra, Evans et Boyd [1] s'énonce ainsi. Un réseau de CFMM ne contient pas d'arbitrage si et seulement s'il existe un vecteur $g \in \mathbb{R}_+^n$ et des scalaires $\lambda_i > 0$ tels que*

$$\gamma_i \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i, \quad i = 1, \dots, m,$$

où $P_i = \nabla \varphi_i(R_i)$.

Esquisse de preuve du sens suffisant. On montre le sens utile pour l'interprétation économique : si un tel système de prix existe, alors il n'existe pas d'arbitrage vectoriel.

Soit $(\Delta_i, \Lambda_i)_{i=1}^m$ une transaction admissible, et soit

$$\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i)$$

son bilan global. Par admissibilité,

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i).$$

Comme φ_i est concave et différentiable en R_i , on a aussi

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \leq \varphi_i(R_i) + P_i^\top (\gamma_i \Delta_i - \Lambda_i).$$

En combinant ces deux inégalités, il vient

$$P_i^\top \Lambda_i \leq \gamma_i P_i^\top \Delta_i.$$

D'autre part, la condition de compatibilité des prix donne

$$A_i^\top g \leq \lambda_i P_i \quad \text{et} \quad A_i^\top g \geq \gamma_i \lambda_i P_i.$$

Ainsi,

$$g^\top A_i (\Lambda_i - \Delta_i) = (A_i^\top g)^\top \Lambda_i - (A_i^\top g)^\top \Delta_i \leq \lambda_i P_i^\top \Lambda_i - \gamma_i \lambda_i P_i^\top \Delta_i \leq 0.$$

En sommant sur i , on obtient donc

$$g^\top \Psi \leq 0.$$

Si Ψ était un arbitrage vectoriel, on aurait $\Psi \geq 0$ et $\Psi \neq 0$. Comme $g \in \mathbb{R}_{++}^n$, cela impliquerait $g^\top \Psi > 0$, contradiction. Il n'existe donc pas d'arbitrage.

La réciproque repose sur un argument de dualité convexe, ou de théorème de l'alternative, et constitue le contenu principal du résultat cité. \square

Ce résultat provient de l'analyse duale du problème de routage : l'absence d'arbitrage impose l'existence d'un système de prix global compatible avec les prix marginaux locaux. Les scalaires λ_i reflètent le fait que les prix d'un CFMM sont définis à un facteur d'échelle près ; les frais transforment l'égalité de prix en bande d'inégalités.

Sans frais, $\gamma_i = 1$, et la condition devient

$$A_i^\top g = \lambda_i P_i, \quad i = 1, \dots, m.$$

Remarque 4.2. Cette écriture dépend de la convention retenue pour exprimer les prix marginaux. Ici, $P_i = \nabla \varphi_i(R_i)$ est un vecteur de prix implicites défini à un facteur d'échelle près. Dans le cas bidimensionnel, le rapport de deux composantes de P_i redonne le prix marginal local, et l'inégalité vectorielle ci-dessus se traduit par une bande de non-arbitrage sur ce rapport.

Dans le cas d'Uniswap V2,

$$\varphi(R_1, R_2) = \sqrt{R_1 R_2}.$$

On a donc

$$\nabla \varphi(R_1, R_2) = \left(\frac{1}{2} \sqrt{\frac{R_2}{R_1}}, \frac{1}{2} \sqrt{\frac{R_1}{R_2}} \right).$$

Par conséquent,

$$\frac{\partial_{R_1} \varphi(R_1, R_2)}{\partial_{R_2} \varphi(R_1, R_2)} = \frac{R_2}{R_1}.$$

Le rapport des composantes du gradient retrouve donc le prix marginal local introduit dans la section consacrée à Uniswap V2.

4.4 Interprétation : prix globaux, prix locaux et bande de non-arbitrage

Dans un pool isolé, la bande de non-arbitrage peut s'écrire

$$\gamma m_p \leq m_u \leq \gamma^{-1} m_p,$$

où m_p désigne le prix externe et m_u le prix marginal implicite du pool, exprimés dans la même convention.

Dans un réseau de CFMM, le rôle du prix de référence est joué par le vecteur global g . La condition

$$\gamma_i \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i$$

doit être satisfaite simultanément pour tous les pools du réseau. Cette condition doit être satisfaite simultanément pour tous les pools du réseau : chaque pool impose une condition locale, mais toutes ces conditions doivent être compatibles avec un même vecteur de prix global g .

4.5 Limites économiques : frais, liquidité, gaz, exécution

La condition

$$\Psi \geq 0, \quad \Psi \neq 0$$

définit un arbitrage dans le modèle statique. Elle ne garantit pas sa profitabilité après coûts d'exécution.

Les frais proportionnels créent une bande de non-arbitrage. Les coûts fixes, notamment le gaz, réduisent encore l'ensemble des opportunités exploitables et justifient l'extension MICP. La liquidité limite aussi la taille des arbitrages : dans un CFMM, le prix marginal se déplace avec les réserves, donc un arbitrage optimal est généralement de taille finie.

Enfin, l'état du réseau peut changer entre la résolution et l'exécution. Si les réserves passent de R_i à $R_i + \varepsilon_i$, une solution admissible pour l'état initial peut devenir inadmissible. Cette question de *stale state* pourrait être étudiée par une analyse de sensibilité fondée sur les multiplicateurs de KKT. Pour obtenir une stratégie exécutable, il faudrait aussi intégrer la latence, la concurrence entre arbitrageurs et le MEV.

5 Application numérique : routage optimal sur un sous-réseau de pools

On considère un snapshot réel de trois pools Uniswap V2 :

$$\text{USDC/DAI}, \quad \text{USDC/USDT}, \quad \text{USDT/DAI}.$$

Les réserves sont récupérées directement depuis les contrats Uniswap V2 via les fonctions `getPair` et `getReserves`. L'objectif est de comparer l'exécution d'un swap de 10 000 USDC vers DAI selon trois méthodes :

$$\text{USDC} \rightarrow \text{DAI}, \quad \text{USDC} \rightarrow \text{USDT} \rightarrow \text{DAI}, \quad \text{routage convexe optimal}.$$

Méthode	DAI reçus	Gain vs direct
Route directe USDC \rightarrow DAI	9784.0661	0
Route indirecte USDC \rightarrow USDT \rightarrow DAI	6279.1081	-3504.9580
Routage convexe optimal	9788.9398	4.8737

TABLE 1 – Comparaison des sorties obtenues pour un swap de 10 000 USDC vers DAI sur un snapshot réel Uniswap V2.

Le gain relatif du routage optimal par rapport à la route directe est

$$10\,000 \left(\frac{9788.9398}{9784.0661} - 1 \right) \simeq 4.98 \text{ bps.}$$

La solution optimale ne choisit pas une route unique. Elle envoie environ 9709.92 USDC dans le pool USDC/DAI et environ 290.08 USDC dans la route indirecte USDC \rightarrow USDT \rightarrow DAI. Le bilan global est

$$\Psi_{\text{USDC}} = -10000, \quad \Psi_{\text{DAI}} = 9788.9398, \quad \Psi_{\text{USDT}} \simeq 0.$$

Le token USDT est donc utilisé uniquement comme actif intermédiaire.

Lorsque la taille de l'ordre augmente, le gain du routage optimal croît. La figure ci-dessous, obtenue en répétant l'expérience pour plusieurs tailles d'ordre, mesure le gain en points de base par rapport à la route directe.

Cette expérience illustre le point central du mémoire : le routage optimal n'est pas un problème de choix d'un chemin unique, mais un problème d'allocation de flux dans un réseau de liquidité. Le résultat numérique est cohérent avec l'intuition économique : la route indirecte est peu performante lorsqu'elle est utilisée seule, mais elle devient utile à la marge pour absorber une petite partie de l'ordre et réduire le slippage sur la route directe.

6 Conclusion

Ce mémoire a présenté une modélisation du routage optimal dans les réseaux de CFMM. À partir du cas élémentaire d'Uniswap V2, on a introduit les notions de prix marginal, de prix effectif, de slippage et de bande locale de non-arbitrage. Le passage d'un pool isolé à un réseau de pools impose ensuite de distinguer les coordonnées locales propres à chaque CFMM et les coordonnées globales du réseau.

Le coeur mathématique du mémoire repose sur la formulation du routage optimal comme un problème d'optimisation. Les contraintes d'égalité issues des fonctions de trading ne sont pas convexes en général. En les relaxant en contraintes d'inégalité, on obtient un problème convexe lorsque les fonctions de trading sont concaves et croissantes. Sous une hypothèse de stricte croissance de l'utilité, ou par sélection d'une solution saturée, cette relaxation est exacte : l'optimum du problème relaxé se situe sur la frontière admissible des CFMM.

Le même cadre permet de formuler la détection d'arbitrage. Un arbitrage vectoriel correspond à une transaction admissible dont le bilan global est composante par composante non négatif et non nul. L'absence d'arbitrage est alors liée à l'existence d'un vecteur de prix global compatible avec les prix marginaux locaux de tous les pools, les frais transformant l'égalité des prix en bande de non-arbitrage.

Enfin, l'application numérique montre que le routage optimal peut améliorer l'exécution par rapport à une route unique. Même lorsque le gain en points de base semble modéré, il devient économiquement significatif pour des tailles d'ordre importantes. Les limites principales du modèle concernent les coûts de gaz, la latence, l'évolution des réserves entre résolution et exécution, ainsi que la concurrence entre arbitragistes. Ces aspects ouvrent naturellement vers des extensions intégrant des coûts fixes, une analyse de sensibilité ou une modélisation dynamique de l'exécution.

Bibliographie

- [1] G. Angeris, T. Chitra, A. Evans, S. Boyd, *Optimal Routing for Constant Function Market Makers*, Proceedings of the ACM Conference on Economics and Computation, 2022.
- [2] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, T. Chitra, *An Analysis of Uniswap Markets*, arXiv :1911.03380, 2019.
- [3] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [4] H. Adams, N. Zinsmeister, D. Robinson, *Uniswap v2 Core*, 2020.
- [5] F. Martinelli, N. Mushegian, *Balancer : A non-custodial portfolio manager, liquidity provider, and price sensor*, 2019.