



**University
of Basel**

Center for
Innovative Finance

Vergleich Constant Function Market Maker

Dario Thürkauf

Betreut von Prof. Dr. Fabian Schär

Credit Suisse Asset Management Professor für DLT/FinTech

Übersicht

Aufbau der Präsentation

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

Übersicht

Einführung

- Krypto-Tauschbörsen

- Constant Function Market Makers

Protokollvergleich

- UniSwap

- Balancer

- Curve

- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen

- Slippage

- Impermanent Loss

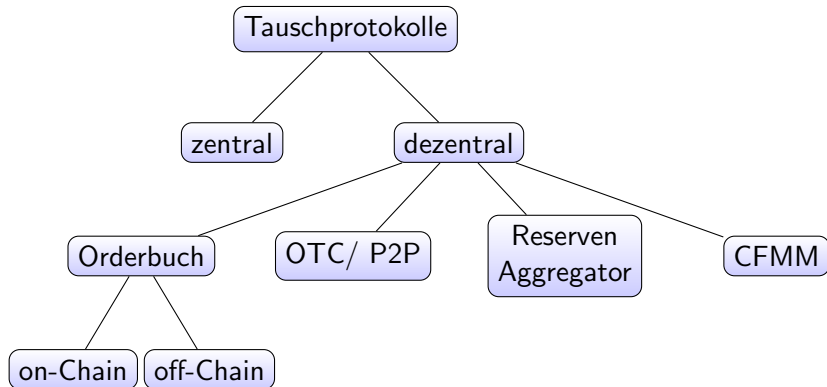
Zusammenfassung

- Front-Running Problem

- weitere Herausforderungen, Marktanteile

Krypto-Tauschbörsen

eine Übersicht



Quelle: eigene Darstellung in Anlehnung an Schär (2020, S.8-11)

Übersicht

Einführung

Krypto-Tauschbörsen

Constant Function Market Makers

Protokollvergleich

UniSwap

Balancer

Curve

Bancor

Gegenüberstellung und Diskussion

Funktionsgleichungen

Slippage

Impermanent Loss

Zusammenfassung

Front-Running Problem

weitere Herausforderungen, Marktanteile

Constant Function Market Maker

Eigenschaften

1. Smart Contract basierte Liquiditätspools: halten Reserven von 2 oder mehr Kryptoassets

Constant Function Market Maker

Eigenschaften

1. Smart Contract basierte Liquiditätspools: halten Reserven von 2 oder mehr Kryptoassets
2. von einem Code kontrolliert

Constant Function Market Maker

Eigenschaften

1. Smart Contract basierte Liquiditätspools: halten Reserven von 2 oder mehr Kryptoassets
2. von einem Code kontrolliert
3. Code führt eine Funktion aus

Constant Function Market Maker

Eigenschaften

1. Smart Contract basierte Liquiditätspools: halten Reserven von 2 oder mehr Kryptoassets
2. von einem Code kontrolliert
3. Code führt eine Funktion aus
4. kein Orderbuch: Nutzer handeln direkt gegen den Kontrakt

Constant Function Market Maker

Eigenschaften

1. Smart Contract basierte Liquiditätspools: halten Reserven von 2 oder mehr Kryptoassets
2. von einem Code kontrolliert
3. Code führt eine Funktion aus
4. kein Orderbuch: Nutzer handeln direkt gegen den Kontrakt
5. ökonomische Anreize für Liquiditätsprovider

Constant Function Market Maker

Vorteile

1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt

Constant Function Market Maker

Vorteile

1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt
2. Zensurresistenz, keine Listungsgebühren, keine Spreads

Constant Function Market Maker

Vorteile

1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt
2. Zensurresistenz, keine Listungsgebühren, keine Spreads
3. Platzsparend: Status kann über Anzahl der gepoolten Assets eindeutig repräsentiert werden

Constant Function Market Maker

Vorteile

1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt
2. Zensurresistenz, keine Listungsgebühren, keine Spreads
3. Platzsparend: Status kann über Anzahl der gepoolten Assets eindeutig repräsentiert werden
4. Offenheit: interne Logik des Smart Contracts kann beobachtet werden, durch Blockchain besichert

Constant Function Market Maker

Vorteile

1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt
2. Zensurresistenz, keine Listungsgebühren, keine Spreads
3. Platzsparend: Status kann über Anzahl der gepoolten Assets eindeutig repräsentiert werden
4. Offenheit: interne Logik des Smart Contracts kann beobachtet werden, durch Blockchain besichert
5. Integration: Smart Contracts können in weitere Applikationen integriert werden

Constant Function Market Maker

Vorteile





1. kein Gegenparteirisiko: beide Seiten des Tauschs finden in einer einzelnen Blockchain-Transaktion statt
2. Zensurresistenz, keine Listungsgebühren, keine Spreads
3. Platzsparend: Status kann über Anzahl der gepoolten Assets eindeutig repräsentiert werden
4. Offenheit: interne Logik des Smart Contracts kann beobachtet werden, durch Blockchain besichert
5. Integration: Smart Contracts können in weitere Applikationen integriert werden
6. Liquidität: Funktionen stellen stetige Liquidität sicher und determinieren die Preise

Frage: Wie sehen diese Funktionen aus?

Constant Function Market Maker

Protokolle

Antwort: Unterschiedlich! Gemeinsamkeit ist die Anwendung einer **konvexen** Funktion.

- ▶ UniSwap 
- ▶ Balancer 
- ▶ Curve 
- ▶ Bancor 

Vergleich: Funktionsgleichung, Slippage, Impermanent Loss und Liquiditätsbereitstellung

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

1. ermöglicht automatischen Tausch zwischen
ETH/ERC20-Token oder ERC20-/ERC20-Token

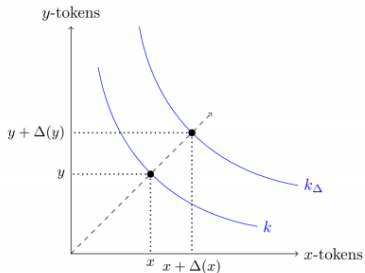
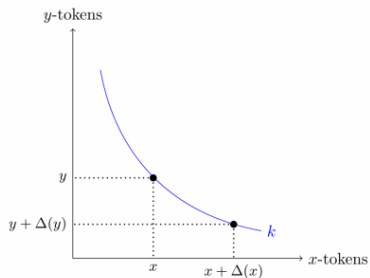
1. ermöglicht automatischen Tausch zwischen ETH/ERC20-Token oder ERC20-/ERC20-Token
2. Preise werden über Constant Product Gleichung bestimmt

1. ermöglicht automatischen Tausch zwischen ETH/ERC20-Token oder ERC20-/ERC20-Token
2. Preise werden über Constant Product Gleichung bestimmt
3. Gebühr von 0.3%

1. ermöglicht automatischen Tausch zwischen ETH/ERC20-Token oder ERC20-/ERC20-Token
2. Preise werden über Constant Product Gleichung bestimmt
3. Gebühr von 0.3%
4. Gebühr wird nach dem Tausch zum Liquiditätspool hinzugefügt

- Constant Product Funktion:

$$x \cdot y = k \quad (1)$$



Quelle: Schär (2020, S.10)

- ▶ Herleitung der Tausch-Formel:

$$k = x \cdot y$$

$$k = (x + (1 - f)\Delta x) \cdot (y + \Delta y)$$

$$\Delta y = \frac{k}{x + (1 - f)\Delta x} - y$$

- ▶ Beispiel: $x = 10$, $y = 10$, $f = 0.003$, $\Delta x = 1$

$$\frac{100}{10 + (1 - 0.003)1} - 10 = -0.9066$$

- ▶ Für das senden von 1 Einheit x erhalten wir ca. 0.907 Einheiten y

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade
- ▶ über partielle Ableitung der Constant Product Gleichung

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade
- ▶ über partielle Ableitung der Constant Product Gleichung
- ▶ Spot-Preis von x: $SP_x = \frac{y}{x}$

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade
- ▶ über partielle Ableitung der Constant Product Gleichung
- ▶ Spot-Preis von x: $SP_x = \frac{y}{x}$
- ▶ SP_x vor dem Trade: $\frac{y}{x} = 10/10 = 1$

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade
- ▶ über partielle Ableitung der Constant Product Gleichung
- ▶ Spot-Preis von x: $SP_x = \frac{y}{x}$
- ▶ SP_x vor dem Trade: $\frac{y}{x} = 10/10 = 1$
- ▶ SP_x nach dem Trade: $\frac{y + \Delta y}{x + \Delta x} = \frac{10 + (-0.9066)}{10 + 1} = 0.827$

UniSwap

effektiver Preis und Spot Preis

- ▶ Der effektive Preis für eine Einheit x-Token ist: $EP_x = \frac{\Delta y}{\Delta x}$
- ▶ Somit gilt: $EP_x = 0.9066$ y-Token
- ▶ Spot-Preis: marginaler Preis für einen Trade
- ▶ über partielle Ableitung der Constant Product Gleichung
- ▶ Spot-Preis von x: $SP_x = \frac{y}{x}$
- ▶ SP_x vor dem Trade: $\frac{y}{x} = 10/10 = 1$
- ▶ SP_x nach dem Trade: $\frac{y + \Delta y}{x + \Delta x} = \frac{10 + (-0.9066)}{10 + 1} = 0.827$
- ▶ Preis von x hat durch den Trade abgenommen, alle Trades verändern den Spot-Preis

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage
- ▶ $SL = \frac{EP_x}{SP_x} - 1 = (0.9066/1) - 1 = -0.0934$

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage
- ▶ $SL = \frac{EP_x}{SP_x} - 1 = (0.9066/1) - 1 = -0.0934$
- ▶ Der Slippage bei dem Trade ist somit ca. 9.34%

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage
- ▶ $SL = \frac{EP_x}{SP_x} - 1 = (0.9066/1) - 1 = -0.0934$
- ▶ Der Slippage bei dem Trade ist somit ca. 9.34%
- ▶ Slippage ist abhängig von der Grösse des Trades im Verhältnis zur Grösse des Pools

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage
- ▶ $SL = \frac{EP_x}{SP_x} - 1 = (0.9066/1) - 1 = -0.0934$
- ▶ Der Slippage bei dem Trade ist somit ca. 9.34%
- ▶ Slippage ist abhängig von der Grösse des Trades im Verhältnis zur Grösse des Pools
- ▶ kommt durch die Konvexität der Kurve zustande

- ▶ Die Abweichung vom Spot-Preis zum effektiven Preis nennt sich Slippage
- ▶ $SL = \frac{EP_x}{SP_x} - 1 = (0.9066/1) - 1 = -0.0934$
- ▶ Der Slippage bei dem Trade ist somit ca. 9.34%
- ▶ Slippage ist abhängig von der Grösse des Trades im Verhältnis zur Grösse des Pools
- ▶ kommt durch die Konvexität der Kurve zustande
- ▶ Arbitrage-Möglichkeit, da relativer Preis im Pool nicht mehr dem Aussenmarkt entspricht

- ▶ initialer Liquiditätsprovider kann Token-Verhältnis beliebig festlegen

- ▶ initialer Liquiditätsprovider kann Token-Verhältnis beliebig festlegen
- ▶ aber: wenn der relative Preis der Token nicht dem Aussenmarkt-Preis entspricht → Arbitrage

- ▶ initialer Liquiditätsprovider kann Token-Verhältnis beliebig festlegen
- ▶ aber: wenn der relative Preis der Token nicht dem Aussenmarkt-Preis entspricht → Arbitrage
- ▶ grundsätzlich Token im gleichen Wert bereitstellen

- ▶ initialer Liquiditätsprovider kann Token-Verhältnis beliebig festlegen
- ▶ aber: wenn der relative Preis der Token nicht dem Aussenmarkt-Preis entspricht → Arbitrage
- ▶ grundsätzlich Token im gleichen Wert bereitstellen
- ▶ Liquiditätsprovider erhält Liquiditätstoken

- ▶ initialer Liquiditätsprovider kann Token-Verhältnis beliebig festlegen
- ▶ aber: wenn der relative Preis der Token nicht dem Aussenmarkt-Preis entspricht → Arbitrage
- ▶ grundsätzlich Token im gleichen Wert bereitstellen
- ▶ Liquiditätsprovider erhält Liquiditätstoken
- ▶ Liquidität abziehen: Liquiditätstoken werden „vernichtet“, enthalten proportionalen Anteil gesammelter Tauschgebühren

- ▶ **Definition:** Verlust der durch das Poolen der Assets entsteht, verglichen mit dem Halten der Assets
 - ▶ wichtig: wird erst bei Abzug der Mittel realisiert und entsteht nur bei relativem Preisunterschied der Assets

- ▶ **Definition:** Verlust der durch das Poolen der Assets entsteht, verglichen mit dem Halten der Assets
 - ▶ wichtig: wird erst bei Abzug der Mittel realisiert und entsteht nur bei relativem Preisunterschied der Assets
- ▶ **Grund:** verändert sich der Preis eines Tokens auf dem Aussenmarkt → Arbitrage bis wieder dem Aussenmarkt entspricht → Veränderung der Token Reserven im Pool und somit Veränderung der Token bei Abzug

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage
4. Über Kombination der Spot-Preis Formel mit der Constant Product Gleichung können wir die neuen Pool Reserven bestimmen

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage
4. Über Kombination der Spot-Preis Formel mit der Constant Product Gleichung können wir die neuen Pool Reserven bestimmen
5. $x = \sqrt{k/SP_x} = \sqrt{100/2} = 7.0716$

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage
4. Über Kombination der Spot-Preis Formel mit der Constant Product Gleichung können wir die neuen Pool Reserven bestimmen
5. $x = \sqrt{k/SP_x} = \sqrt{100/2} = 7.0716$
6. $y = 100/7.0716 = 14.142$

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage
4. Über Kombination der Spot-Preis Formel mit der Constant Product Gleichung können wir die neuen Pool Reserven bestimmen
5. $x = \sqrt{k/SP_x} = \sqrt{100/2} = 7.0716$
6. $y = 100/7.0716 = 14.142$
7. Pool Wert neu = $7.0716 \cdot 2\$ + 14.142 \cdot 1\$ = 28.284\$$

UniSwap

Impermanent Loss - Beispiel

1. Ausgangslage: 10 x-Token und 10 y-Token, beide Token sind jeweils 1\$ Wert
2. Annahmen: Wir haben den ganzen Pool bereitgestellt, keine Tradegebühren
3. Preis von x-Token auf dem Aussenmarkt verdoppelt sich → Arbitrage
4. Über Kombination der Spot-Preis Formel mit der Constant Product Gleichung können wir die neuen Pool Reserven bestimmen
5. $x = \sqrt{k/SP_x} = \sqrt{100/2} = 7.0716$
6. $y = 100/7.0716 = 14.142$
7. Pool Wert neu = $7.0716 \cdot 2\$ + 14.142 \cdot 1\$ = 28.284\$$
8. Haltewert neu = $10 \cdot 2\$ + 10 \cdot 1\$ = 30\$$

- ▶ allgemeine Impermanent Loss-Formel:

$$IL = \frac{PoolWert_{neu}^{\$}}{Haltewert_{neu}^{\$}} - 1$$

- ▶ allgemeine Impermanent Loss-Formel:

$$IL = \frac{PoolWert_{neu}^{\$}}{Haltewert_{neu}^{\$}} - 1$$

- ▶ Impermanent Loss im Beispiel: $(28.284 / 30) - 1 = -0.057$

Unsere Assets haben ca. 5.7% weniger Wert, als wenn wir sie einfach gehalten hätten! **Aber:** konnten Gebühren sammeln.

UniSwap

Impermanent Loss

- ▶ allgemeine Impermanent Loss-Formel:

$$IL = \frac{PoolWert_{neu}^{\$}}{Haltewert_{neu}^{\$}} - 1$$

- ▶ Impermanent Loss im Beispiel: $(28.284 / 30) - 1 = -0.057$
- ▶ UniSwap IL in Abhängigkeit des Preisunterschieds:

$$IL = \frac{2\sqrt{\Delta P_x^{\$} \Delta P_y^{\$}}}{\Delta P_x^{\$} + \Delta P_y^{\$}} - 1$$

- ▶ wobei $\Delta P = \frac{Preis_{neu}}{Preis_{alt}}$

Unsere Assets haben ca. 5.7% weniger Wert, als wenn wir sie einfach gehalten hätten! **Aber:** konnten Gebühren sammeln.

Übersicht

Einführung

Krypto-Tauschbörsen

Constant Function Market Makers

Protokollvergleich

UniSwap

Balancer

Curve

Bancor

Gegenüberstellung und Diskussion

Funktionsgleichungen

Slippage

Impermanent Loss

Zusammenfassung

Front-Running Problem

weitere Herausforderungen, Marktanteile

Balancer

Funktionsgleichung

- ▶ Erweitert das Konzept von UniSwap auf Pools mit bis zu 8 verschiedenen Token
- ▶ Constant Value Funktion

$$V = \prod_t B_t^{w_t} \quad (2)$$

- ▶ B_t Anzahl Token t
- ▶ w_t normalisiertes Gewicht von Token t

Balancer

Funktionsgleichung

- ▶ Erweitert das Konzept von UniSwap auf Pools mit bis zu 8 verschiedenen Token
- ▶ Constant Value Funktion

$$V = \prod_t B_t^{w_t} \quad (2)$$

- ▶ B_t Anzahl Token t
- ▶ w_t normalisiertes Gewicht von Token t
- ▶ normalisiertes Gewicht entspricht dem Wertanteil von t am gesamten Pool-Wert
- ▶ Summe der normalisierten Gewichte ist 1
- ▶ Gewicht und damit Wertanteil eines Tokens im Pool bleibt konstant

Balancer

Liquidität bereitstellen

- ▶ alle Token mit entsprechender Gewichtung oder nur ein einziger Token

- ▶ alle Token mit entsprechender Gewichtung oder nur ein einziger Token
- ▶ Hinzufügen eines einzelnen Tokens A ist gleichzusetzen mit dem Hinzufügen aller Token und dem sofortigen Austausch der restlichen Token zu Token A

- ▶ alle Token mit entsprechender Gewichtung oder nur ein einziger Token
- ▶ Hinzufügen eines einzelnen Tokens A ist gleichzusetzen mit dem Hinzufügen aller Token und dem sofortigen Austausch der restlichen Token zu Token A
- ▶ daher: Einzel-Deposit mit Gebühr

Balancer

Tausch-Formel, Spot-Preis und effektiver Preis

- ▶ Tausch-Formel:

$$A_o = B_o \cdot \left(1 - \left(\frac{B_i}{B_i + A_i} \right)^{\frac{w_i}{w_o}} \right)$$

- ▶ effektiver Preis:

$$EP_i = \frac{A_o}{A_i}$$

- ▶ Spot-Preis:

$$SP_i = \frac{\frac{B_o}{w_o}}{\frac{B_i}{w_i}}$$

- ▶ Einsetzen von EP_i und SP_i in die bekannte Slippage Gleichung ergibt:

$$SL = \frac{A_o}{A_i} \cdot \frac{B_i}{B_o} \cdot \frac{w_o}{w_i} - 1$$

- ▶ Die Höhe des Slippage ist somit abhängig vom relativen Pool-Gewicht der beiden gehandelten Token
 - ▶ Dazu in der Diskussion mehr

- ▶ Gleichung wie zuvor, nun generalisiert für n-Token in Abhängigkeit des Preisunterschieds:

$$IL = \frac{\prod_t (\Delta P_t^{\$})^{w_t}}{\sum_t (\Delta P_t^{\$} \cdot w_t)} - 1$$

- ▶ Gleichung wie zuvor, nun generalisiert für n-Token in Abhängigkeit des Preisunterschieds:

$$IL = \frac{\prod_t (\Delta P_t^{\$})^{w_t}}{\sum_t (\Delta P_t^{\$} \cdot w_t)} - 1$$

- ▶ Impermanent Loss ist somit auch von Gewichten abhängig
 - ▶ Ebenfalls in der Diskussion mehr

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve**
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

- ▶ Tausch zwischen Stablecoins

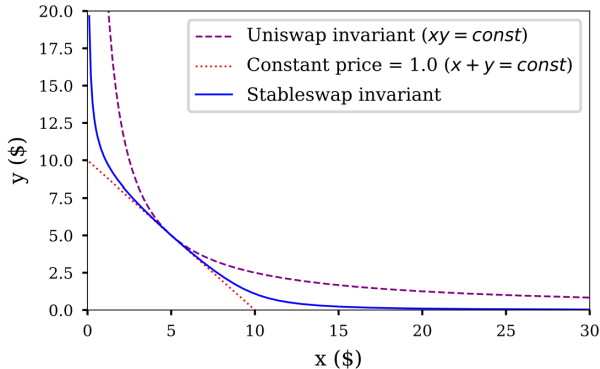
- ▶ Tausch zwischen Stablecoins
- ▶ Hintergrund: Slippage für Trade von Assets, die preisstabil bleiben sollten, nicht wünschenswert

- ▶ Tausch zwischen Stablecoins
- ▶ Hintergrund: Slippage für Trade von Assets, die preisstabil bleiben sollten, nicht wünschenswert
- ▶ Kombination von Constant Sum- und Constant Product-Function

- ▶ Tausch zwischen Stablecoins
- ▶ Hintergrund: Slippage für Trade von Assets, die preisstabil bleiben sollten, nicht wünschenswert
- ▶ Kombination von Constant Sum- und Constant Product-Function
- ▶ Hybrid Constant Function Market Maker

► Curve Funktion:

$$An^n \sum x_i + D = ADn^n + \frac{D^{n+1}}{n^n \prod x_i} \quad (3)$$



- ▶ Kurve bei Pool-Gleichgewicht nur schwach gekrümmt
- ▶ Effektiver Preis weicht nur leicht vom Spot-Preis ab → tiefer Slippage
- ▶ ausserhalb des Gleichgewichts: Slippage, dafür immernoch Liquidität

Curve

Liquidität bereitstellen und Impermanent Loss

- ▶ Flexibilität beim Hinzufügen und Abziehen der Token eines vorhandenen Pools
- ▶ Liquiditätsprovider erhalten ebenfalls Liquiditätspool-Token
- ▶ Pools, welche die Liquiditätspooltoken weiteren DeFi-Lending Protokollen (bspw. Compound/Aave) hinzufügen um zusätzliche Rendite zu erzielen

Curve

Liquidität bereitstellen und Impermanent Loss

- ▶ Flexibilität beim Hinzufügen und Abziehen der Token eines vorhandenen Pools
 - ▶ Liquiditätsprovider erhalten ebenfalls Liquiditätspool-Token
 - ▶ Pools, welche die Liquiditätspooltoken weiteren DeFi-Lending Protokollen (bspw. Compound/Aave) hinzufügen um zusätzliche Rendite zu erzielen
-
- ▶ Da es sich bei den gepoolten Assets um Stablecoins handelt, ist Impermanent Loss grundsätzlich nicht von Bedeutung

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor**

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

Bancor V1

allgemein

- ▶ Pools bestehend aus 2 Token
- ▶ Smart Token im Zentrum des Protokolls
- ▶ Brücke zwischen verschiedenen Token
- ▶ Funktionsgleichung: Konstantes Verhältnis (Gewicht) zwischen der Reserve des Connector-Tokens und dem Gesamtwert des Smart Tokens wird beibehalten

Bancor V1

Funktionsgleichung und Formeln

- ▶ Bancor Funktion:

$$R = FSP \quad (4)$$

- ▶ R Connector-Reserve, F Gewicht des Connector-Token, SP Wert des Smart Token

Bancor V1

Funktionsgleichung und Formeln

- ▶ Bancor Funktion:

$$R = FSP \quad (4)$$

- ▶ R Connector-Reserve, F Gewicht des Connector-Token, SP Wert des Smart Token

- ▶ Smart Token (T) bzw. Connector-Token (E) erhalten:

$$T_{erhalten} = S_0 \cdot \left(\left(1 + \frac{E_{gesendet}}{R_0} \right)^F - 1 \right)$$

$$E_{erhalten} = R_0 \cdot \left(\left(1 + \frac{T_{gesendet}}{S_0} \right)^{1/F} - 1 \right)$$

1. Dynamic Automated Market Maker
 - ▶ Preisänderungen im Aussenmarkt werden über ein Orakel erkannt und die Ziel-Gewichte der Token entsprechend angepasst
2. Einzel-Asset Deposit
 - ▶ Liquiditätsprovider müssen nicht mehr über beide Assets verfügen, separate Liquiditätstoken pro Liquiditätspool, Gewichte passen sich ebenfalls an
3. Liquiditäts-Amplifikationsmechanismus
 - ▶ neue, flexible Funktionen sollen Slippage verringern
4. Integration der Pools mit weiteren Lending-Protokollen

- ▶ **Ziel:** relative Anzahl der Reserven soll nicht vom Anfangsinvestment abweichen
 - ▶ keine Arbitrage auf Kosten der Liquiditätsprovider und damit kein Impermanent Loss
 - ▶ Arbitrage wird nur noch bei der Veränderung der Pool-Verteilung durch Trades benötigt, um die vom Orakel definierten Ziel-Gewichte wiederherzustellen

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

Zusammenfassung

Funktionsgleichungen

Protokoll	Funktionsgleichung	Art der Funktion
UniSwap	$x \cdot y = k$	Constant Product
Balancer	$V = \prod_t B_t^{w_t}$	Constant Value
Curve	$s \cdot \sum_i x_i + \prod_i x_i = k$	Constant Sum/-Product
Bancor V1	$R = FSP$	Constant Reserve

Quelle: eigene Darstellung

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage**
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

Zusammenfassung und Diskussion

Slippage

- ▶ allgemein: je grösser der Trade im Verhältnis mit der gesamten Pool-Grösse, desto höher der Slippage

Zusammenfassung und Diskussion

Slippage

- ▶ allgemein: je grösser der Trade im Verhältnis mit der gesamten Pool-Grösse, desto höher der Slippage
- ▶ **Curve:** tiefster Slippage, aber nur für preisstabile Assets wie Stablecoins

Zusammenfassung und Diskussion

Slippage

- ▶ allgemein: je grösser der Trade im Verhältnis mit der gesamten Pool-Grösse, desto höher der Slippage
- ▶ **Curve:** tiefster Slippage, aber nur für preisstabile Assets wie Stablecoins
- ▶ **Bancor V2:** will mit Amplifikationsmechanismus tieferen Slippage als UniSwap und Balancer erreichen

Zusammenfassung und Diskussion

Slippage

- ▶ allgemein: je grösser der Trade im Verhältnis mit der gesamten Pool-Grösse, desto höher der Slippage
- ▶ **Curve:** tiefster Slippage, aber nur für preisstabile Assets wie Stablecoins
- ▶ **Bancor V2:** will mit Amplifikationsmechanismus tieferen Slippage als UniSwap und Balancer erreichen
- ▶ **UniSwap:** grundsätzlich weniger Slippage als Balancer, weil Gewichtung standardmässig 50/50

Zusammenfassung und Diskussion

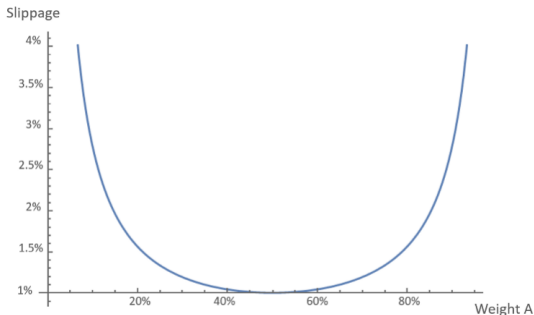
Slippage

- ▶ allgemein: je grösser der Trade im Verhältnis mit der gesamten Pool-Grösse, desto höher der Slippage
- ▶ **Curve:** tiefster Slippage, aber nur für preisstabile Assets wie Stablecoins
- ▶ **Bancor V2:** will mit Amplifikationsmechanismus tieferen Slippage als UniSwap und Balancer erreichen
- ▶ **UniSwap:** grundsätzlich weniger Slippage als Balancer, weil Gewichtung standardmässig 50/50
- ▶ **Balancer:** Trades zwischen gleichgewichteten Token haben den gleichen Slippage wie UniSwap, ansonsten höher

Zusammenfassung und Diskussion

Slippage

- ▶ Gleichverteilung der Token-Gewichte minimiert den Slippage
 - ▶ Beispiel: Pool mit Gewichten von (0.4/0.4/0.1/0.1)



Quelle: Martinelli (2020), <https://medium.com/balancer-protocol/80-20-balancer-pools-ad7fed816c8d>

Übersicht

Einführung

Krypto-Tauschbörsen

Constant Function Market Makers

Protokollvergleich

UniSwap

Balancer

Curve

Bancor

Gegenüberstellung und Diskussion

Funktionsgleichungen

Slippage

Impermanent Loss

Zusammenfassung

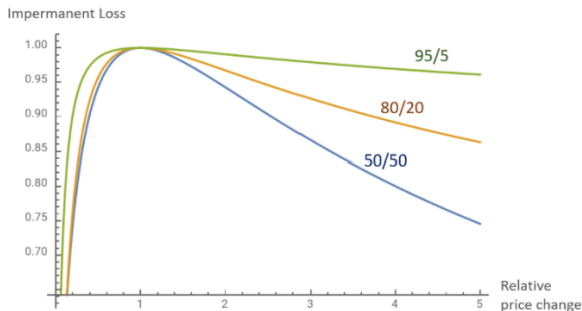
Front-Running Problem

weitere Herausforderungen, Marktanteile

Impermanent Loss

Zusammenfassung und Diskussion

- ▶ einseitiger Preisunterschied führt immer zu Impermanent Loss
- ▶ Liquiditätspool mit einer Gewichtung von 50/50 maximiert den Impermanent Loss



Quelle: Martinelli (2020), <https://medium.com/balancer-protocol/80-20-balancer-pools-ad7fed816c8d>

Zusammenfassung und Diskussion

Impermanent Loss

- ▶ **UniSwap:** Constant Product Formel (Gewichtung von 50/50) maximiert den IL

Zusammenfassung und Diskussion

Impermanent Loss

- ▶ **UniSwap:** Constant Product Formel (Gewichtung von 50/50) maximiert den IL
- ▶ **Balancer:** LP können die Gewichte beliebig festlegen und damit den Upside gegenüber einem Asset beibehalten, IL wird verringert
 - ▶ dafür höherer Slippage → weniger Handelsvolumen → weniger Handelsgebühren

Zusammenfassung und Diskussion

Impermanent Loss

- ▶ **UniSwap:** Constant Product Formel (Gewichtung von 50/50) maximiert den IL
- ▶ **Balancer:** LP können die Gewichte beliebig festlegen und damit den Upside gegenüber einem Asset beibehalten, IL wird verringert
 - ▶ dafür höherer Slippage → weniger Handelsvolumen → weniger Handelsgebühren
- ▶ **Curve:** Stablecoins sollten keinen grossen relativen Preisunterschied aufweisen, IL dadurch nicht wirklich von Bedeutung

Zusammenfassung und Diskussion

Impermanent Loss

- ▶ **UniSwap:** Constant Product Formel (Gewichtung von 50/50) maximiert den IL
- ▶ **Balancer:** LP können die Gewichte beliebig festlegen und damit den Upside gegenüber einem Asset beibehalten, IL wird verringert
 - ▶ dafür höherer Slippage → weniger Handelsvolumen → weniger Handelsgebühren
- ▶ **Curve:** Stablecoins sollten keinen grossen relativen Preisunterschied aufweisen, IL dadurch nicht wirklich von Bedeutung
- ▶ **Bancor V2:** Anpassung der Pool-Gewichte mittels Orakel, sollte Impermanent Loss eliminieren
 - ▶ Problem: Orakel aktualisieren zu langsam, gleichzeitig Front-Running möglich, IL also (noch) nicht lösbar

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

- ▶ „Ausnützen einer privaten Information, welche den Preis eines Assets ändern könnte, für finanziellen Gewinn.“

(vgl. Zhou et. al., 2020, S.1)

- ▶ „Ausnützen einer privaten Information, welche den Preis eines Assets ändern könnte, für finanziellen Gewinn.“

(vgl. Zhou et. al., 2020, S.1)

- ▶ im Kontext von CFMMs: Sandwich-Attacke

Sandwich-Attacke

ein Beispiel

1. **Ausgangslage:** UniSwap Liquiditätspool bestehend aus 10 Einheiten Token A und 10 Einheiten Token B. Ein Nutzer will eine Einheit A gegen B tauschen. Miner sieht diese Transaktionsnachricht in seinem Mempool und verfasst zwei eigene Transaktionsnachrichten, die er vor und nach der ursprünglichen Transaktion im Block platziert.

Sandwich-Attacke

ein Beispiel

1. **Ausgangslage:** UniSwap Liquiditätspool bestehend aus 10 Einheiten Token A und 10 Einheiten Token B. Ein Nutzer will eine Einheit A gegen B tauschen. Miner sieht diese Transaktionsnachricht in seinem Mempool und verfasst zwei eigene Transaktionsnachrichten, die er vor und nach der ursprünglichen Transaktion im Block platziert.
2. **Front-Run Transaktion:** Miner sendet eine Einheit A und erhält dafür 0.9091 Einheiten von B.
3. **Ursprungstransaktion:** Nutzer sendet eine Einheit A, erhält dafür 0.757576 Einheiten B.
4. **Back-Run Transaktion:** Miner sendet 0.7576 Einheiten von B und erhält 1 Einheit A.

Sandwich-Attacke

ein Beispiel

1. **Ausgangslage:** UniSwap Liquiditätspool bestehend aus 10 Einheiten Token A und 10 Einheiten Token B. Ein Nutzer will eine Einheit A gegen B tauschen. Miner sieht diese Transaktionsnachricht in seinem Mempool und verfasst zwei eigene Transaktionsnachrichten, die er vor und nach der ursprünglichen Transaktion im Block platziert.
2. **Front-Run Transaktion:** Miner sendet eine Einheit A und erhält dafür 0.9091 Einheiten von B.
3. **Ursprungstransaktion:** Nutzer sendet eine Einheit A, erhält dafür 0.757576 Einheiten B.
4. **Back-Run Transaktion:** Miner sendet 0.7576 Einheiten von B und erhält 1 Einheit A.
5. **Profit für den Miner:** $0.9090 \text{ B} - 0.7575 \text{ B} = 0.1515$ B-Token auf Kosten des Nutzers, der anstatt 0.9090 nur 0.7575 Einheiten von B erhält.

Sandwich-Attacke

ein Beispiel

Aktion	Anzahl A	Anzahl B	Konstante	gesendet → erhalten
1.	10	10	100	-
2.	11	9.0909	100	1 A → 0.9090 B
3.	12	8.3333	100	1 A → 0.7575 B
4.	11	9.0909	100	0.7575 B → 1 A

Quelle: eigene Darstellung in Anlehnung an Buterin(2018),
<https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers/1281>

- ▶ Lösung: maximaler Slippage von Nutzer definierbar, wenn darüber liegt wird Trade nicht ausgeführt
- ▶ bessere Lösungen?

Übersicht

Einführung

- Krypto-Tauschbörsen
- Constant Function Market Makers

Protokollvergleich

- UniSwap
- Balancer
- Curve
- Bancor

Gegenüberstellung und Diskussion

- Funktionsgleichungen
- Slippage
- Impermanent Loss

Zusammenfassung

- Front-Running Problem
- weitere Herausforderungen, Marktanteile

Constant Function Market Makers

Herausforderungen und aktueller Marktanteil

- ▶ weitere Herausforderungen
 1. Sicherheitsrisiken von Smart Contracts, v.a. beim Zusammenspiel mehrerer Protokolle
 2. Skalierung
 3. Open-Source Protokolle: können kopiert werden, siehe SushiSwap

- ▶ aktuelle Übersicht Marktvolumen:
<https://coinmarketcap.com/rankings/exchanges/dex/>

Constant Function Market Makers

Herausforderungen und aktueller Marktanteil

- ▶ weitere Herausforderungen
 1. Sicherheitsrisiken von Smart Contracts, v.a. beim Zusammenspiel mehrerer Protokolle
 2. Skalierung
 3. Open-Source Protokolle: können kopiert werden, siehe SushiSwap
- ▶ aktuelle Übersicht Marktvolumen:
<https://coinmarketcap.com/rankings/exchanges/dex/>
- ▶ Vielen Dank für Ihre Aufmerksamkeit!

Egorov, M. (2019) ,Stableswap - efficient mechanism for stablecoin liquidity.'

URL: <https://www.curve.fi/stableswap-paper.pdf>

Schär, F. (2020) ,Decentralized finance: On blockchain- and smart contract-based financial markets.'

URL: <https://dx.doi.org/10.13140/RG.2.2.18469.65764>

Martinelli, F. (2020) ,80/20 balancer pools, Medium.'

URL: <https://medium.com/balancer-protocol/80-20-balancer-pools-ad7fed816c8d>

Zhou, L.; Qin, K.; Torres, C.F.; Le, D.V. & Gervais, A (2020) ,High-frequency Trading on Decentralized On-Chain Exchanges.'

URL: <https://arxiv.org/abs/2009.14021>