쉘 프로토콜 백서

2020년 10월 1일

추상적인

쉘 프로토콜의 목표는 스테이블 코인을 빌딩 블록으로 사용하여 인터넷 통화 시스템을 만드는 것입니다. 첫 번째 릴리스는 스테이블코인 간 거래에 최적화된 유동성 풀입니다.
그것은 가중치, 깊은 유동성, 부러진 페그에 대한 보호 및 동적 수수료를 가지고 있습니다. 또한 cToken 및 aToken과 같은 스테이블 코인 파생 상품 간에 직접 유동성을 제공할 수도 있습니다. 이 반복의 목표는 새로운 사용 사례와 시장 조건에 적응할 수 있는 유연한 결합 곡선을 위한 프레임워크를 개발하는 것입니다. 유동성 공급자 주식은 살아있는 껍질이 내부 유기체의 용기인 것처럼 풀이 보유한 가치의 용기이기 때문에 "껍질"이라고 합니다. 쉘은 본질적으로 유동적이며 노출을 다양화하고 수익을 얻습니다. 그것들은 가치를 저장하고 거래하는 주요 수단이 될 가능성이 있습니다.

1 인터넷 머니

쉘 프로토콜의 목적은 모두가 접근할 수 있는 국경 없는 프로그래밍 가능한 교환 매체인 인터넷 통화 시스템을 만드는 것입니다. 이 꿈은 비트코인의 원래 약속이었습니다.[21] 고정된 공급으로 인해 비트코인은 화폐로 사용하기에는 너무 변동성이 큽니다. 어떻게 보면 지난 몇 년 동안 그 임무는 중도에 빠졌습니다. 그러나 우리는 포기하지 말아야 합니다.

이 백서는 스테이블 코인을 건물로 사용하여 인터넷 머니를 만드는 방법을 설명합니다.
블록.

은행은 레거시 화폐의 기초입니다. 스테이블 코인은 인터넷 머니의 기초입니다.
은행 시스템은 중복성과 그에 상응하는 수준의 비효율성으로 가득 찬 개인 원장의 임시 연합입니다. Stablecoin은 퍼블릭 블록체인에 존재하며 인터넷 네이티브이며 본질적으로 프로그래밍이 가능합니다. 스테이블 코인이 은행을 대체하는 것은 불가피하며 아마도 우리가 예상하는 것보다 훨씬 빠를 것입니다. 미국 규제 당국은 이미 USD 스테이블 코인을 수용하기 시작했습니다.[11]

단일 스테이블 코인은 그 자체로 인터넷 화폐를 구성하지 않습니다. 대신 전체 스테이블 코인 생태계가 필요합니다. 건강한 스테이블코인 생태계에는 다양하고 통합된 시장이 필요합니다. 즉, 다양한 종류의 스테이블 코인이 있어야 하고 이들 간의 가치 이전이 원활해야 합니다. 그러나 현재 스테이블 코인 시장은 고도로 집중되고 세분화되어 있습니다. 실제로 스테이블 코인은 미국 은행 시스템보다 더 집중되어 있습니다. 상위 5개 미국 은행의 규모는 거의 비슷하며 전체 가치의 40%를 넘지 않습니다.

시스템에 의해 유지됩니다(그림 1 참조). 대조적으로, 상위 5개의 스테이블 코인은 매우 불평등하며, 1위 스테이블 코인인 Tether[27]는 2위 스테이블 코인인 USD Coin[5]의 약 8배의 가치를 보유하고 있습니다. 전체적으로 상위 5개의 스테이블코인이 시장의 95% 이상을 차지합니다(그림 2 참조).

"탈중앙화" 금융(DeFi)이 레거시 금융보다 더 중앙 집중화되어 있다는 것은 큰 아이러니입니다.
우리는 스테이블 코인이 너무 커서 실패하는 것을 원하지 않습니다. 우리는 스테이블 코인을 일관되지만 다양한 생태계로 통합할 수 있는 프로토콜이 필요합니다. Stablecoin 유동성 풀은 중요한 첫 단계입니다.

유동성 풀을 사용하면 "유동성 공급자"라고 하는 사람들이 이 자본을 사용하여 풀이 보유하고 있는 예비 자산 간의 거래를 용이하게 하는 자율 시스템에 자본을 기여할 수 있습니다. 이러한 거래의 가격은 AMM(Automated Market Maker)에 의해 결정됩니다.[4]
예를 들어 Alice가 A 토큰을 풀에 제공하고 Bob이 B 토큰을 풀에 제공하면 이제 풀은 내부 논리에 따라 A와 B 간의 거래를 촉진할 수 있습니다.

스마트 계약에 기반한 유동성 풀의 힘은 유동성에 대한 사회적 확장성 문제[26]를 해결한다는 것입니다. 시스템이 자율적으로 작동할 수 있기 때문에 Alice와 Bob과 같은 유동성 공급자는 서로를 알거나 신뢰할 필요가 없으며 전문 시장 조성자를 알거나 신뢰할 필요도 없습니다. 그들은 독립적으로 확인할 수 있는 AMM 계약에 대한 신뢰만 있으면 됩니다. 이렇게 하면 이제 자산 간 유동성을 생성하는 데 사용할 수 있는 많은 양의 유휴 자본이 잠금 해제됩니다. 실제로 풀은 변동성 위험을 감수하면서 나머지 시장에 유동성 프리미엄을 부과합니다. 스테이블코인 간 풀은 스테이블코인이 페그를 영구적으로 잃지 않는 한 변동성 위험이 상대적으로 낮기 때문에 특히 효과적입니다.

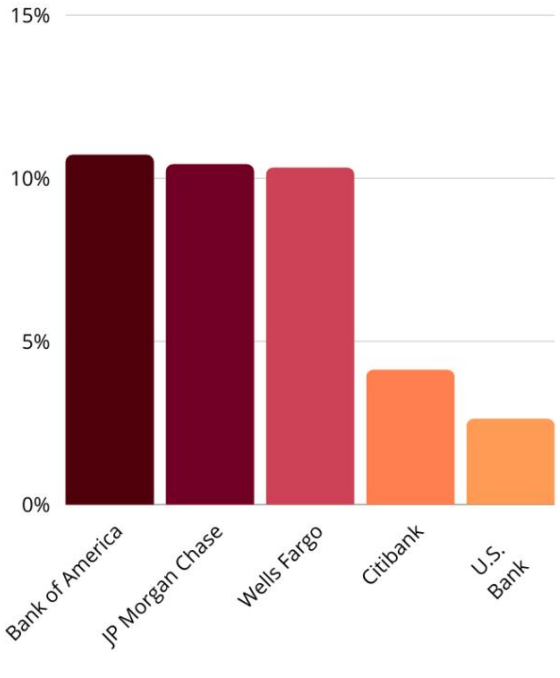

그림 1: 예금 기준 상위 5개 미국 은행의 시장 점유율 [24]

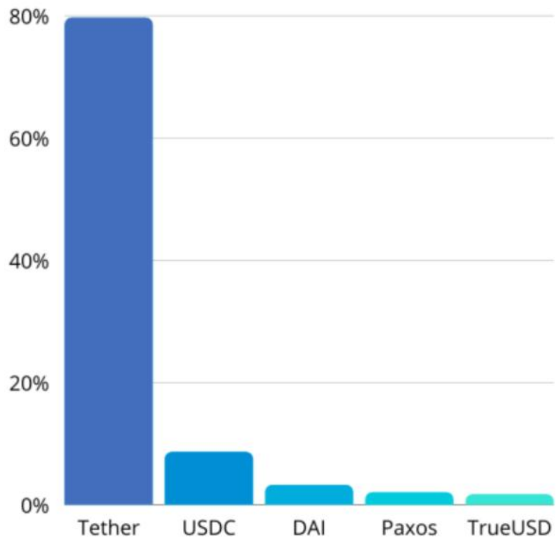

그림 2: 시가 총액 기준 상위 5개 스테이블 코인의 시장 점유율 [7]

스테이블 코인 유동성 풀의 현재 시장 리더는 Curve[13] 와 Swerve[25]입니다. 작성 당시(2020년 9월), 그들은 각각 12억 달러[9] 와 4,100만 달러[25] 상당의 스테이블 코인을 풀에 보유하고 있습니다. 그리고 그들은 일일 거래량에서 9,800만 달러와 210만 달러를 처리했습니다. 2020년 6월 4개월도 채 되지 않아 Curve는 스테이블 코인이 1,600만 달러에 불과했고 Swerve는 존재하지도 않았습니다. 지금까지 성장률이 인상적이었지만 우리는 아직 전환 주기의 초기 단계에 있습니다. 비유하자면, 미국 은행 시스템의 결제 수단인 Fedwire는 연간 약 750조 달러를 처리합니다.[14] Stablecoin은 적어도 이 크기까지 커질 것입니다. 오늘날 세상은 우편 서비스를 통해 보내는 편지보다 훨씬 더 많은 이메일을 보냅니다. Inter-stablecoin 볼륨 대 Fedwire 볼륨은 동일한 패턴을 따릅니다.

스테이블 코인은 미래의 인터넷 통화 시스템으로 성장할 씨앗이기 때문에 AMM과 유동성 풀 이상을 생각해야 합니다. 단일 스테이블 코인을 넘어. 대신 껍질에 대해 생각해야 합니다.

2 껍질이란?

풀의 스테이블 코인은 유동성 공급자가 기여한다는 점을 상기하십시오. 기여도를 추적하기 위해 풀은 토큰을 예치하는 사람에게 주식을 발행합니다. 풀에서 토큰을 인출하기 위해 유동성 공급자는 주식을 상환합니다. 우리는 유동성 풀의 주식을 "껍질"이라고 부릅니다. 살아있는 껍질이 내부의 유기체를 담는 그릇인 것처럼 유동성 풀의 기본 자산을 담는 그릇이기 때문입니다.

쉘은 비활성 스테이블 코인에 비해 세 가지 주요 이점이 있습니다.

1. 유동성
2. 다양화
3. 수율

실제로 쉘은 토큰화된 유동성 풀입니다. 따라서 그들은 스왑에 활용되는 바로 그 유동성을 물려받습니다. 풀의 스테이블 코인은 직접 쉘로 변환할 수 있으며, 그 반대의 경우 쉘은 스테이블 코인으로 직접 변환할 수 있습니다. 그런 의미에서 쉘은 기본적으로 유동 자산입니다.

쉘은 효율적인 유동성 공급원 이상을 제공합니다. 그들은 다양한 자산 포트폴리오입니다. 다각화는 단일 스테이블 코인에 비해 쉘 보유 위험을 낮출 수 있습니다. 하나의 스테이블 코인이 페그를 잃을 확률은 모든 스테이블 코인이 페그를 잃을 확률보다 높습니다.

또한 유동성 풀은 더 넓은 시장에 유동성을 제공함으로써 쉘 보유자를 대신하여 소극적 소득을 얻을 수 있습니다. 모든 스왑에 대해 수수료는 껍질에 직접 발생하여 수동 소득을 얻는 풀에서 얻습니다. 따라서 포탄은 수익률을 내는 자산입니다.

쉘 토큰은 다양하고 본질적으로 유동적이며 수익을 창출한다는 점을 감안할 때 모든 암호화 자산의 가치를 가장 잘 저장할 수 있는 잠재력을 가지고 있습니다. 또한 여러 스테이블 코인보다 단일 쉘을 관리하는 것이 더 쉽습니다. 레거시 금융 시스템에서 가장 안전하고 유동적인 이자가 발생하는 자산은 미국 국채입니다. 현재 암호화폐 시장에서 그 틈새를 충족하는 자산은 없습니다.

2.1 쉘용 프로토콜

쉘이 가치(예: "돈")를 교환하고 저장하는 기본 수단이 된다면 그러한 시스템은 어떤 모습일까요? 동일한 숫자에 고정되어 있지만 스테이블 코인은 여전히 다른 위험 프로필, 규제 제약, 지리적 분포 등과 같은 의미 있는 차이가 있습니다. 예를 들어 스테이블 코인은 USD 코인 또는 팩소스와 같은 감사 은행 계좌에 보관된 명목 화폐로 뒷받침될 수 있습니다[23]. 또는 테더와 같은 감사되지 않은 계정에 명목 화폐 준비금을 보유할 수 있습니다. Maker[18] 와 Synthetix[6] 는 각각 암호화폐로 뒷받침되는 자체 스테이블 코인을 발행합니다.

Terra에는 거래 수수료에서 생성된 미래 현금 흐름으로 페그가 뒷받침되는 스테이블 코인이 있습니다.[16] 스테이블 코인의 다양성은 시간이 지남에 따라 증가할 것입니다. 게다가 스테이블 코인이 더 광범위하게 채택됨에 따라 사용자는 점점 더 이질적이 될 것입니다. 초기 DeFi 사용자는 높은 위험 허용 범위를 가지고 있지만 주류 사용자에게는 해당되지 않습니다.

전 세계가 동일한 스테이블코인 쉘을 사용하게 될까요? 아니, 당연하지. 어떤 스테이블 코인이 그 껍질에 들어갈까요? 예비 자산의 가중치는 어떻게 됩니까? 수영장은 얼마나 위험해야 합니까? 모든 사람의 요구를 충족하는 단일 쉘에 동의하는 것은 불가능합니다. 무수한 웅덩이, 수천 개의 조개 껍질이 있을 것입니다. 아마도 중요한 것은 시장 상황이 매우 가변적일 것입니다. 스테이블코인 가격이 안정적일 때 잘 작동하는 풀 설계는 스테이블코인 가격이 변동적일 때 최적이 아닙니다(섹션 4.2 참조).

따라서 쉘을 생성하기 위한 프로토콜은 다양한 스테이블 코인 및 시장 조건에 적응할 수 있도록 고도로 구성 가능해야 합니다. 이 백서에서는 쉘과 유동성 풀을 생성하기 위한 모델을 설명하고 공식화합니다. 이 첫 번째 반복의 강조점은

매우 유연한 코어 로직을 위한 프레임워크. 즉, 매개변수를 조정하는 것만으로 풀의 동작을 의미 있게 변경할 수 있습니다.

Shell은 오픈 소스이므로 누구나 허가 없이 기술적으로 풀을 배포할 수 있습니다.
그러나 프로토콜의 참신함을 감안할 때 독립적인 배포는 프로젝트가 성숙해질 때까지 초기에 권장되거나 지원되지 않습니다.
궁극적으로 사람들이 자신의 풀을 구성, 배포 및 운영하는 것이 쉬워질 것입니다.

원칙적으로 제시된 모델은 스테이블 코인뿐만 아니라 모든 수치 내 유동성 풀을 수용할 수 있습니다. 예를 들어, BTC-to-BTC 풀은 USD-to-USD 풀만큼 실행 가능합니다.
핵심 논리의 관점에서 "안정적"은 매장량의 절대 가치가 아니라 상대적 가치를 의미합니다. Shell의 사명을 감안할 때 이 백서의 초점은 USD 스테이블코인에 맞춰질 것입니다.
나머지 논문은 다음과 같이 구성될 것이다. 섹션 3은 프로토콜에 대한 높은 수준의 설명을 제공합니다. 섹션 4에서는 접합 곡선을 통해 모델을 시각화합니다. 섹션 5에서는 Shell의 초기 사용 사례에 대해 설명합니다. 섹션 6에서는 Shell의 AMM 모델을 공식화합니다. 특히 섹션 6.1은 핵심 용어를 엄격하게 정의합니다. 마지막으로 섹션 7에서는 프로토콜의 향후 방향을 설명합니다.

3 쉘 프로토콜 설명

Shell 유동성 풀을 구성할 수 있는 다섯 가지 차원이 있습니다.

1. 예비 스테이블 코인의 이상적인 가중치(w)
2. 1:1 교환의 깊이(β)
3. 1:1 교환이 아닐 때의 가격 슬리피지(탄력성)(Δ)
4. 각 준비금의 최대 및 최소 할당(α)
5. 수수료(, λ)

3.1 예비 가중치

풀의 각 스테이블코인 보유량은 이상적인 가중치 w를 가집니다. 풀이 보유하려는 각 보유량의 백분율을 결정합니다. w = 0.5인 경우 해당 리저브의 이상적인 가중치는 풀이 보유한 총 리저브의 50%입니다. 배포되는 첫 번째 Shell 풀은 30% DAI, 30% USDC, 30% USDT 및 10% SUSD입니다. 풀이 "균형"일 때 실제 가중치는 이상적인 가중치에 가깝고 풀은 일정한 합계 시장 조성자 역할을 합니다. 모든 잔액의 합계는 스왑 전후에 변경되지 않습니다. 스테이블 코인 간의 교환은 1:1로 이루어지며 고정 수수료를 제외하고는 슬리피지가 없습니다(섹션 3.4 참조).

3.2 유동성 깊이와 가격 하락

이상적인 가중치는 다른 매개변수 β와 상호 작용하여 해당 준비금의 유동성 깊이를 결정합니다. 실제 가중치가 이상적인 가중치에서 β% 이상 벗어나면 거래

미끄러지기 시작합니다. 예를 들어 w = 0.5 및 β = 0.75인 경우 풀의 12.5%보다 낮거나 87.5%보다 크면 토큰은 점점 더 큰 슬리피지를 겪게 됩니다. 따라서 β 값이 높을수록 풀이 더 깊습니다. β를 지나서 가격 하락이 증가하는 비율(탄력성)은 매개변수 Δ에 의해 결정됩니다. Δ 값이 높을수록 미끄러짐이 더 빠르게 증가합니다. Δ = 0이면 어떤 시점에서도 가격 하락이 없습니다.

이것이 매력적으로 보일 수 있지만 가격 하락 없이 페그에서 조금만 벗어나도 모든 유동성이 고갈될 것입니다. 따라서 어떤 가격 하락도 실제로 풀의 유동성을 떨어뜨리지 않을 것입니다. 특히 가격 페그 이상에서 벗어나는 스테이블코인의 경우 더욱 그렇습니다. 이 정확한 현상은 미끄럼 방지 AMM이 있는 "메타" 스테이블 코인 바스켓인 mStable에서 발생했습니다.[20] 예비 스테이블코인 중 하나인 DAI는 일반적으로 페그 이상으로 거래되며 mStable 풀은 만성적인 DAI 부족을 겪고 있습니다.

3.3 부러진 못으로부터의 보호

스테이블코인 페그가 깨지지 않도록 보호하기 위해 풀에는 각 준비금에 대한 최소 및 최대 할당이 있습니다. 이 할당은 α에 의해 결정됩니다. α 값이 높을수록 최소 할당 비율은 낮아지고 최대 할당 비율은 높아집니다. 리저브의 이상적인 가중치가 w = 0.25 및 α = 0.8인 경우 풀은 리저브의 실제 가중치를 45% 이상으로 증가시키는 거래를 수락하지 않습니다. 이 제약 조건이 없으면 트레이더가 가치가 없을 때까지 가치 없는 스테이블 코인을 계속 추가하고 실행 가능한 스테이블 코인을 제거하기 때문에 부러진 페그는 전체 풀을 고갈시킬 것입니다. 각 스테이블코인에 대한 임계값을 설정하지 않고 예비 자산을 다양화하면 풀을 완전히 소모하는 데 하나의 부러진 페그만 필요하기 때문에 풀을 더 위험하게 만들 것입니다. Curve, Uniswap[2] 및 Balancer[19] 와 같은 AMM 프로토콜의 경우입니다 . 쉘 풀에는 최소 및 최대 할당 매개변수가 있기 때문에 다각화는 실제로 위험을 증가시키기보다는 감소시킬 수 있습니다.

3.4 수율

풀은 고정 수수료와 동적 수수료 λ를 청구하여 포탄에 대한 수익을 얻습니다. 값이 풀에서 나갈 때마다 해당 값의 비율이 유지되어 쉘에 직접 누적됩니다. 또한 풀은 가격 하락 영역의 모든 거래에 대해 동적 수수료를 부과합니다. 즉, 스테이블 코인이 가격 페그에서 더 많이 벗어날수록 매수/매도 스프레드가 높아집니다.

논의한 바와 같이 이상적인 가중치를 너무 많이 초과하는 준비금을 사용하는 트랜잭션은 균형이 맞지 않는 페널티를 받게 됩니다. 반대로, 슬리피지 존에서 시작할 때 준비금을 이상적인 가중치에 더 가깝게 만드는 거래는 재조정 보조금을 받게 됩니다. 풀의 균형이 맞지 않으면 미끄러짐이 발생합니다. 풀을 재조정하면 미끄러짐 방지가 발생합니다. 매개변수 λ는 불균형 수수료와 재조정 보조금 간의 차이를 결정합니다. 차이(1 λ)는 풀에서 동적 수수료로 유지됩니다. λ = 1이면 동적 수수료가 없습니다. λ = 0.5이면 동적 수수료는 가격 슬리피지의 절반이 됩니다.

동적 수수료는 가격 변동성이 있을 때 풀에 특히 유리할 수 있습니다. 스테이블 코인이 페그에서 벗어날 때마다 풀은 이 미끄러짐의 일부를 포착합니다. λ = 0.7 및 = 0.00035(3.5 베이시스 포인트)라고 가정합니다. 가격이 페그에서 5%만큼 벗어나면 풀은 고정 수수료보다 동적 수수료에서 약 100배 더 많은 수익을 올릴 것입니다. 무역

동적 수수료는 차익 거래자가 풀과 상호 작용하는 인센티브를 줄여 거래량을 줄입니다. 그러나 위의 예에서 거래량은 동적 수수료에서 생성된 수입을 상쇄하기 위해 100배 이상 감소해야 합니다.

이상적인 동적 수수료를 추론하기 위해서는 추가 모델링이 필요하며, 이는 시장 상황과 스테이블 코인 보유량에 따라 달라집니다.

3.5 동적 매개변수

이러한 매개변수(α, β, Δ, , λ)는 풀이 새로운 시장 조건에 동적으로 적응할 수 있도록 배치 후 조정될 수 있습니다. 풀은 변동성이 낮을 때 깊은 1:1 유동성을 가질 수 있고 변동성이 높을 때 매개변수를 재조정할 수 있습니다(섹션 4.2 참조). 첫 번째 릴리스의 경우 배포 시 가중치가 고정되지만 후속 반복에서는 동적 가중치를 사용할 수 있습니다. 매개변수 업데이트에 대한 유일한 제약은 풀이 보유한 값(유틸리티 기능으로 측정, 섹션 6.3 참조)이 새 매개변수의 결과로 감소할 수 없다는 것입니다. 이렇게 하면 중앙 집중식 풀 운영자가 풀에서 자금을 제거할 수 없습니다. Shell 모델은 스테이블코인 이상으로 작동할 수 있습니다. 모든 준비금이 동일한 숫자(예: USD, BTC 또는 ETH)로 표시되는 한 모델은 똑같이 잘 작동합니다. 따라서 풀은 이더리움의 BTC 또는 다양한 스테이킹 파생 상품 간에 유동성을 생성할 수 있습니다.

4 껍질 시각화

Shell Protocol을 시각화하고 매개변수 조정이 동작에 어떤 영향을 미치는지 확인하는 것이 도움이 됩니다. 먼저 AMM의 논리를 시각화하는 방법을 설명하겠습니다. 풀에 잔액이 있는 두 개의 예비 자산이 있다고 상상해 보십시오: [x, y]. 누군가 x를 y로 교환하려는 경우 풀의 새 예비 잔액은 [x+ x, y y]가 됩니다. XY 평면에서 [x+ x, y y]의 모든 가능한 값을 그래프로 나타낼 수 있습니다. 우리는 이 플롯을 "결합 곡선"[10]이라고 부르며 그 안에 풀의 AMM 동작이 포함되어 있습니다. Shell의 AMM은 효용함수(섹션 6.3 참조)를 사용하기 때문에 결합 곡선도 무차별 곡선입니다.[15] 임의의 주어진 점(즉, 도함수)에서 결합 곡선의 기울기는 x의 관점에서 y의 현물 가격입니다.

이 섹션에서는 세 가지 예를 시각화합니다.

1. "표준" 쉘 접합 곡선(그림 3)
2. 매개변수 세트가 다른 중첩된 본딩 곡선(그림 4)
3. 변동 수수료가 있는 결합 곡선(그림 5)

각 예 는 이상적인 가중치에 있을 때 예비 잔액이 [x, y] = [50, 50]인 균일한 가중치 풀(wx = wy = 0.5)을 특징으로 합니다.

4.1 "표준" 접합 곡선

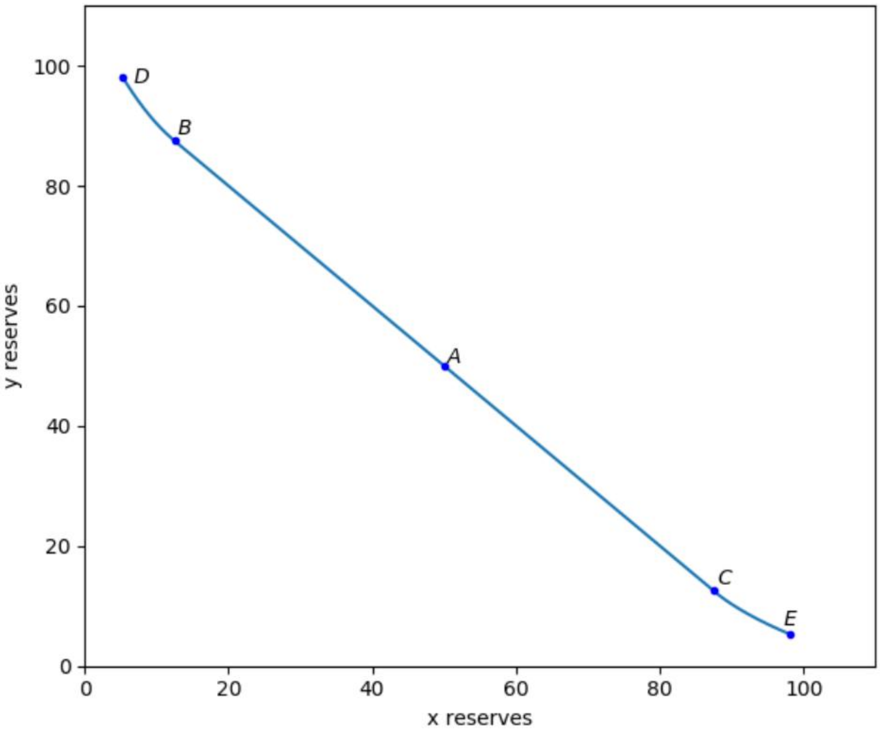

그림 3: 고르게 균형 잡힌 풀의 결합 곡선.

그림 3은 유동성이 풍부한 "표준" 쉘 결합 곡선을 보여줍니다. 지점 A에서 풀은 완벽하게 균형을 이룹니다. 점 B와 C는 β = 0.75인 미끄러짐 구역의 시작입니다.
따라서 세그먼트 BC의 길이는 풀의 깊이입니다. 슬리피지 존의 가격 변동률은 Δ = 1.5로 결정됩니다. 점 D와 E는 α = 0.9인 중단 경계입니다.

그래프의 기울기(도함수)가 x 측면에서 y의 가격을 결정한다는 점을 상기하십시오. 직선이 기울기가 $\frac{\partial y}{\partial x} = 1$(BC)인 직선 영역은 교환이 1:1입니다. 즉, 가격 하락이 없습니다. 풀이 x에 대해 과체중이고 y에 대해 과소평가할 때 그래프는 수평으로 구부러지고(CE) x의 가격은 y에 비해 상대적으로 감소합니다. 반대로, 풀이 x에 대해 저체중이고 y에 대해 비중이 높을 때 그래프는 수직(DB) 곡선을 그리며 x의 가격은 y에 비해 상대적으로 증가합니다.

이 그래프에서 알아야 할 중요한 점은 매개변수를 변경하기만 하면 각 지점(A, B, C, D 및 E)을 변경할 수 있다는 것입니다. 따라서 AMM은 다른 시장 조건에 따라 동작을 조정할 수 있습니다(섹션 4.2 참조).

4.2 유연한 곡선

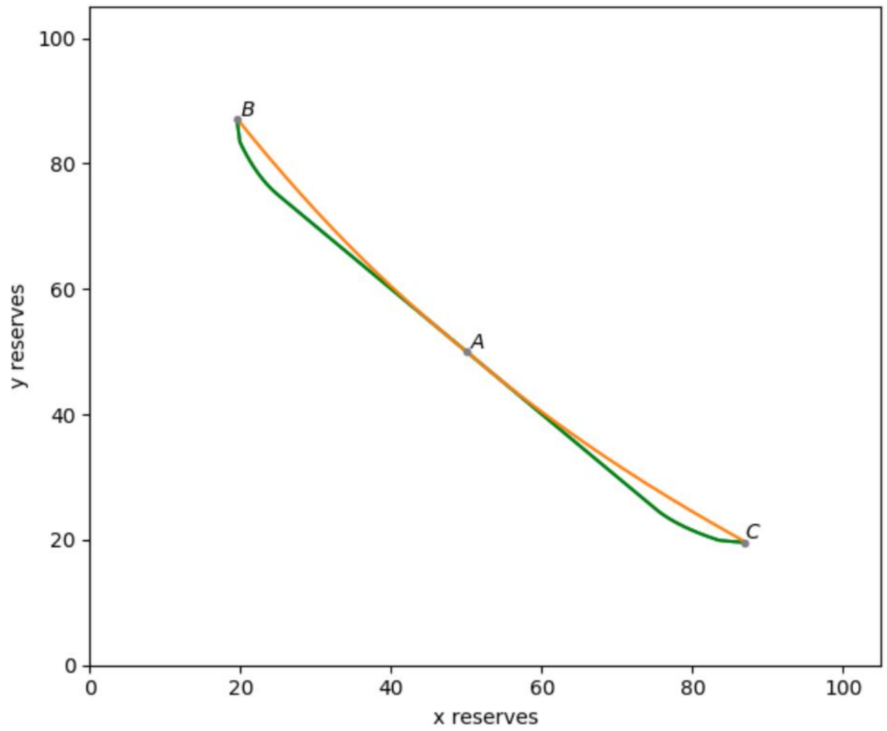

그림 4: 균형점은 같지만 매개변수가 다른 두 개의 본딩 곡선.

서로 다른 매개변수가 결합 곡선의 모양(따라서 모델의 동작)에 어떻게 영향을 미칠 수 있는지 보여주기 위해 그림 4에서 두 개의 곡선을 하나의 그래프에 오버레이할 수 있습니다. 둘 다 A의 동일한 균형점을 공유합니다. 그러나 곡선1(녹색)는 $\beta 1 = 0.5$ 및 $\beta 2 = 0.05$ 인 곡선 2(주황색) 보다 더 얕은 미끄럼 방지 영역을 가집니다 . 그러나 curve1은 $\Delta 1 = 2.5$ 및 $\Delta 2 = 0.185$ 인 curve2보다 낮은 미끄러짐 매개변수를 가집니다 . 따라서 두 곡선은 B와 C의 동일한 정지점에서 끝납니다.

스테이블코인이 페그에서 벗어나면 차익 거래자는 슬리피지 존(섹션 3.2 참조)에 도달할 때까지 풀에 대해 거래합니다. 곡선에 1:1 교환을 위한 넓은 세그먼트가 있는 경우(그림 3의 BC) 가격 변동성이 높을 때 더 많은 유동성을 잃게 됩니다.
곡선이 1:1 교환을 위한 좁은 세그먼트와 더 많은 곡률을 가지고 있다면 시장 변동성이 높을 때 유동성 손실이 적습니다. 따라서 커브2는 스테이블 코인의 변동성이 상대적으로 높은 시장에 더 적합할 것입니다. 반면 곡선1은 스테이블 코인의 변동성이 상대적으로 낮은 시장에 더 적합합니다. Shell 모델의 유연성은 동일한 풀이 최적의 거래를 선택하여 시장에 따라 결합 곡선을 변형할 수 있음을 의미합니다.

전략.

4.3 동적 수수료

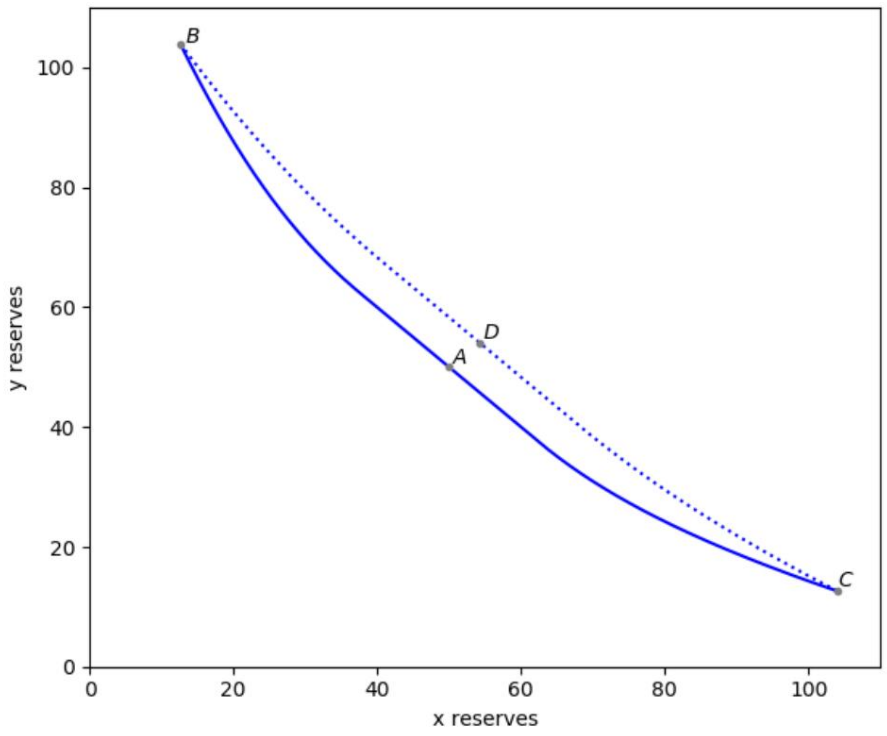

그림 5: 점선으로 표시된 동적 수수료

동적 수수료를 어떻게 시각화할 수 있습니까? 그림 5에서는 균형점 A에서 시작합니다. 그런 다음 정지 임계값 B 또는 C(실선)로 끝까지 바꿉니다. 스왑을 취소하고 균형점으로 돌아가면 미끄럼 방지를 미끄럼의 λ만큼 설정하여 동적 수수료가 평가됩니다. 이 경우 λ = 0.5이므로 미끄럼 방지는 미끄러짐의 절반입니다. 따라서 정지 경계에서 다시 새로운 균형점인 D = [54, 54]로 돌아올 때 풀의 준비금은 이전보다 높아질 것입니다(점선). A와 D 사이의 거리는 유동성 풀에 의해 포착되고 쉘 토큰에 누적되는 동적 수수료입니다.

동적 수수료가 없으면 A → B → C로 이동하면서 얻은 소득은 약 0.0378이 됩니다(시작 준비금 100 대비). 동적 수수료로 얻은 총 수입은 8이었습니다. 따라서 이 예의 동적 수수료는 고정 수수료보다 풀에 대해 200배 이상의 수입을 창출했습니다. 그리고 결합 곡선의 모양과 마찬가지로 동적 수수료는 시장 상황에 맞게 조정될 수 있습니다.

5 사용 사례

Shell Protocol의 가장 확실한 사용 사례는 DeFi 거래자를 위한 유동성 소스와 유동성 공급자를 위한 수익 소스입니다. 유동성이 거래량을 낳기 때문에 유동성 공급자가 초기 초점입니다. Shell은 유동적인 수수료와 파손된 스테이블 코인 페그에 대한 보호 기능이 있으므로 보호 기능이 없고 고정 수수료만 있는 기존 스테이블 코인 AMM에 비해 더 적은 위험으로 더 높은 수익을 얻을 수 있는 잠재력이 있습니다. 쉘 풀은 시장에 따라 동적으로 조정될 수도 있습니다.

정황.

그러나 유동성 공급자를 위한 Shell의 진정한 이점은 모델의 유연성에 있습니다. 예를 들어 스테이블코인 준비금은 풀에서 정확한 가중치를 가질 수 있으며 이는 다른 스테이블코인 AMM과 비교할 때 고유합니다. 따라서 SUSD와 같은 상대적으로 소량의 스테이블코인은 Tether와 같은 대량의 스테이블코인에 비해 더 작은 가중치를 가질 수 있습니다. 실제로 이것은 SUSD에 10% 할당하고 Tether에 30% 할당하는 첫 번째 Shell 풀의 구조와 정확히 일치합니다.

5.1 새로운 스테이블코인을 위한 부트스트래핑 유동성

새로운 스테이블 코인의 유동성을 부트스트랩하기 위해 역으로 가중치를 사용할 수도 있습니다. 새 stablecoin에 할당된 목표 가중치가 50%인 풀을 만들 수 있습니다. 이 풀은 새로운 스테이블 코인의 상위 프로토콜에서 유동성 채굴 인센티브[3]를 받을 수 있습니다. 균등하게 가중된 풀을 사용하면 풀의 다른 모든 스테이블 코인 간에 유동성 인센티브가 희석되는 반면, 과체중 풀은 보조금 1달러당 더 많은 유동성을 생성합니다.

5.2 스테이블코인 파생상품 간의 유동성

Shell은 이더리움에서 스테이블 코인, 스테이킹 파생 상품 또는 비트코인 간의 유동성을 촉진하는 데 국한되지 않습니다. Shell은 또한 Compound[17], Aave[1] 및 yEarn[29] 과 같은 대출 풀과 스마트 수익률 프로토콜 간에 유동성을 제공할 수 있습니다 . 예를 들어 Zap[30]을 사용하지 않고 한 트랜잭션에서 cUSDC를 USDT로 직접 교환할 수 있습니다 . 풀의 보유량에도 동일한 유연성이 적용됩니다. Stablecoin은 yEarn에 입금할 수 있으며 AMM의 거래 수수료 외에 추가 수익을 얻을 수 있습니다. 풀을 변경할 필요 없이 더 많은 통합을 추가할 수 있는 아키텍처입니다. 보안을 위해 풀이 배포되면 새 통합이 해당 풀에 추가되지 않습니다. 풀의 거버넌스가 분산되면 이 제약이 완화될 수 있습니다.

5.3 DeFi 프로토콜의 결제 통화

프로토콜은 유동성과 수익률 이상의 것을 제공합니다. 쉘 토큰은 다른 DeFi 프로토콜과의 직접적인 교환 매체가 될 수 있습니다. 셸은 스테이블 코인 준비금으로 전환할 수 있으며 그 반대의 경우도 마찬가지입니다. 예를 들어, Opyn[22]와 같은 탈중앙화 옵션 거래소라면 시장은 스테이블 코인으로 정산되어야 합니다.

이상적으로는 거래소는 유동성을 파편화하지 않고 다양한 스테이블코인을 수용함으로써 스테이블코인에 구애받지 않을 수 있습니다.

단일 스테이블 코인을 통해 정산하는 대신 쉘 토큰으로 거래를 정산할 수 있습니다. Alice가 Bob에게 풋 옵션을 매도했다고 가정합니다. Alice는 DAI에서 옵션을 정산하고 Bob은 USDC에서 옵션을 정산하기를 원합니다. 스마트 계약 수준에서 옵션은 껍질에 정착할 수 있습니다. Bob이 옵션을 실행하면 쉘이 USDC로 변환됩니다. 옵션을 실행하지 않으면 쉘이 DAI로 변환되어 Alice에게 반환됩니다.

셸이 대출 풀과 다른 스테이블 코인 파생 상품 간에 전환할 수 있다는 점을 감안할 때 DeFi 프로토콜은 추가 통합을 추가할 필요 없이 플랫폼에서 방대한 자산을 수용할 수 있습니다. 또한 필요에 따라 맞춤화된 매개변수를 사용하여 자체 풀을 만들 수 있습니다. 이러한 방식으로 다른 스테이블 코인 AMM을 사용하는 것은 부러진 페그에 대한 보호 장치가 없는 경우 매우 위험합니다(섹션 3.3 참조). 이러한 보호 장치가 없으면 하나의 깨진 페그가 전체 풀을 고갈시키고 다른 DeFi 교환을 중단시킵니다.

6 Shell 프로토콜 공식 모델

이제 유동성 풀을 위한 Shell Protocol 모델을 공식화할 준비가 되었습니다. 형식주의는 먼저 풀의 주요 용어, 동작 및 매개변수를 다룹니다. 그런 다음 프로토콜이 예금, 인출 및 스왑을 계산하는 방법을 설명합니다. 모델 작동 방식에 대한 덜 수학적인 설명은 섹션 3을 참조하십시오. Shell의 모델을 시각화하려면 섹션 4를 참조하십시오.

6.1 정의

더 진행하기 전에 핵심 용어를 엄격하게 정의해야 합니다. 편의를 위해 이 정보는 목록으로 표시됩니다.

유동성 풀: ERC-20 토큰과 같은 여러 종류의 대체 가능한 자산을 보유하고 관리하는 스마트 계약입니다. 유동성 풀을 통해 사용자는 이러한 자산을 입금, 인출 및 교환할 수 있습니다.

유동성 풀 자산: 유동성 풀에서 관리하는 다양한 자산의 수입니다.

유동성 풀 준비금: 벡터 x = [x1, x2, . . . , xn], 여기서 n은 유동성 풀 자산이고 xi 는 현재 유동성 풀이 보유하고 있는 i번째 자산의 양입니다.

유동성 풀 셸(Liquidity Pool Shells): 유동성 풀에서 대체 가능한 주식과 유동성 풀이 보유하고 있는 자산을 나타내는 ERC-20 토큰입니다.

유동성 풀 상태: 쌍 [x1, x2, . . . , xn], T, 여기서 [x1, x2, . . . , xn] 은 유동성 풀 준비금이고 T는 유통되는 유동성 풀 쉘의 총량입니다.

입금: 새로 생성된 쉘과 교환하여 추가 자산을 유동성 풀에 넣는 행위.

인출: 쉘이 순환에서 제거되는 대가로 유동성 풀에서 자산을 가져오는 행위.

스왑(Swap): 한 자산의 일정량을 다른 자산과 교환하여 유동성 풀에 넣는 행위.

6.2 행동

이러한 용어가 정의되었으므로 이제 주요 동작을 정의해야 합니다. 즉, 예금, 스왑 및 인출이 있습니다. 예금에서 사용자는 풀에 토큰을 추가합니다. 그 대가로 그녀는 포탄을 받습니다. 인출에서 사용자는 쉘을 교환하여 토큰을 인출합니다. 스왑에서 사용자는 하나의 예비 토큰을 다른 토큰으로 교환합니다.

입금 기능을 다음과 같이 정의합니다.

D([x1, x2, . . , xn], T, [x1 , xx 2 , . . . , n]) = 및 (1)

여기서 [x1, x2, . . . , xn], T는 입금 전 유동성 풀 상태, [x 1 , xx 2 , . . . , n]는 풀에 추가된 자산 금액의 벡터이고 y는 생성되어 자산과 교환하여 예금자에게 제공되는 쉘의 양입니다. 입금 후 상태는 다음과 같이 나타낼 수 있습니다.

[x1 + 엑스 x2 + x xn + x 1 , 2 , . . . , N], 티 + 와이

입금 기능과 유사하게 인출 기능을 다음과 같이 정의합니다.

W([x1, x2, . . , xn], T, [x 1 , xx 2 , . . . , N]) = 및 (2)

여기서 [x1, x2, . . . , xn], T는 출금 전 유동성 풀 상태, [x 1 , xx 2 , . . . , n]는 풀에서 인출된 자산 금액의 벡터이고 y는 인출자가 자산과 교환한 껍질의 양입니다. 이 쉘은 순환에서 영구적으로 제거됩니다.

인출 후 상태는 다음과 같이 나타낼 수 있습니다.

[x1 - x 1 , x2 - x 2 , . . . , xn - x N], 티 - 와이

스왑 기능의 경우 다음과 같이 정의합니다. 모든 i ∈ {1 에 대해. . . n} 및 j ∈ {1 . . . 엔} \ {i}
i번째 자산을 j번째 자산으로 교환하기 위한 스왑 기능을 다음과 같이 정의합니다.

Si,j ([x1, x2, ..., xn], T, x) = y (삼)

여기서 [x1, x2, . . . , xn], T는 스왑 전 유동성 풀 상태, x는 풀에 부여된 i번째 자산의 양, y는 스왑 개시자에게 부여된 j번째 자산의 양입니다. 우리는 x를 스왑 "원본 금액"으로, i를 "원본 토큰"이라고 합니다. 그리고 y는 스왑 "목표 금액"이고 j는 "목표 토큰"입니다. 이는 스왑 개시자가 거래의 기원인 x로 시작하여 거래의 대상인 y로 끝나기 때문입니다. 스왑 후 상태는 다음과 같이 나타낼 수 있습니다. [x 1 ,

xx 2 , . . . , n], T, 여기서:

$$엑스 케이 = \begin{cases} xk + x, k = i인 경우 \\ xk \quad y, k = j인 경우 \\ xk, 그렇지 않으면 \end{cases}$$

6.3 유틸리티 기능

유동성 풀은 준비금과 껍질 사이의 상대적 가치를 계산할 수 있어야 합니다. 본질적으로 주관적인 개념인 가치를 평가하는 프로그래밍 가능한 규칙을 어떻게 만들 수 있습니까? 쉘 프로토콜은 유틸리티 기능인 U를 사용합니다. 유틸리티 기능은 대안 세트의 요소에 값 점수를 할당합니다. 에이전트(예: 유동성 풀)가 두 가지 대안 중에서 선택해야 하는 경우 효용 점수가 더 높은 것을 선택합니다. Shell의 경우 유틸리티 함수에 대한 입력은 토큰 준비금의 잔액, x = [x1, x2, . . . , xn]이고 출력은 스칼라 값 점수입니다. Shell 풀의 효용 함수에는 이득 함수 G와 수수료 함수 F의 두 가지 구성 요소가 있습니다.

$$유(엑스) = 지(엑스) \quad F(엑스)$$

$$지(엑스) = xi$$

$$에프(엑스) = 파이(엑스)$$

효용함수는 이익함수와 수수료함수의 차이다. 이득 함수는 풀에 보관된 모든 예비 잔액의 합계입니다. 수수료 함수는 각 개별 토큰 fi 에 대한 소액 수수료 함수의 합입니다 . 풀의 효용 기능의 복잡성과 풀의 논리의 대부분은 소액 수수료 기능에 내재되어 있습니다. 높은 수준에서 fi는 풀에 있는 예비 토큰의 실제 무게가 이상적인 무게이거나 그 근처에 있을 때 0입니다. 그러나 예비 토큰의 실제 가중치가 특정 임계값을 벗어나면 fi는 2차적으로 증가하기 시작합니다. 모든 준비금 토큰이 이상적인 가중치에 충분히 근접하면 효용 함수는 모든 준비금 잔액의 합계인 G가 됩니다. 따라서 풀이 균형 상태에 있는 동안 스왑 중에 잔액 합계를 일정하게 유지하는 일정한 합계의 자동 시장 조성자처럼 작동합니다. 상수 합계 시장 조성자는 기본적으로 1:1 가격으로 토큰을 거래합니다.

소액 수수료 기능을 해체하려면 몇 가지 새로운 변수를 정의해야 합니다. 그만큼
수영장의 각 보호 구역에 대한 이상적인 무게, 위스콘신 , 다음과 같은 제약 조건이 있습니다.

$$wi \in (0, 1), \qquad wi = 1$$

wi 임계값 Li , 우리는 각 토큰에 대한 이상적인 균형을 구축할 수 있습니다 . , 하한 및 상한 수수료
및 Vi 각각 사용 :

$$Ii = G \cdot wi$$

$$Li = Ii \cdot (1 - \beta)$$

$$Vi = Ii \cdot (1 + \beta)$$

어디:

$$\beta \in (0, 1)$$

낮은 수수료 임계값은 이상적인 잔액보다 낮은 β퍼센트이고 높은 수수료 임계값은 이상적인 잔액보다 높은 β퍼센트입니다. "no-fee zone" 또는 "no-slippage zone"은 거리가 2 · β · wi · G 단위 길이 인 Ii 에 중심점이 있는 밴드이다 . 준비금 잔액이 이 범위에 속하면 fi = 0 입니다. 이 논리를 캡슐화하기 위해 수수료 마진, Mi fee, mi를 정의하겠습니다 .

, 그리고 중간체

$$Mi = 최대(Li - xi \quad , xi - Vi , 0)$$

$$미 = \Delta \cdot \frac{미}{이이}$$

$$fi = 미분 (mi , \gamma)$$

$$\Delta > 0$$

매개변수 Δ는 수수료 함수의 증가율로 생각할 수 있습니다. Δ가 높을수록 준비금 잔고가 수수료 구역으로 깊숙이 들어갈수록 fi가 더 빠르게 증가합니다. 매개변수 γ는 최대 허용 중개 수수료로 생각할 수 있습니다. 현재 구현에서는 γ = 4 로 설정합니다 . 이 제약 조건을 포함하면 F < G가 보장됩니다. 이제 fi 에 대한 완전한 표현을 찾기 위해 몇 가지 축소 작업을 수행할 수 있습니다 .

1

$$Mi = 최대(Li - xi \quad , \quad , 0) = 최대(Ii \cdot (1 - \beta) - xi\ xi - Vi \quad , xi - II \cdot (1 + \beta), 0) =$$

$$= Ii \cdot 최대\ 1 - \beta - \quad \frac{xi}{이이}, \frac{xi}{이이} \quad 1 \quad \beta, 0 =$$

$$= ii \cdot 최대\ 1 \quad \frac{xi}{이이} \quad - b, \quad \frac{xi}{이이} \quad 1 \quad \beta, 0 =$$

$$= Ii \cdot 최대\ 최대\ 1 \quad \frac{xi}{이이}, \frac{xi}{이이} \quad 1 \quad \beta, 0 =$$

$$= ii \cdot 최대 \quad \frac{xi}{이이} \quad 1 \quad b, 0$$

그리고 몇 가지 추가 감소:

$$fi = Mi\ min\ (mi , \gamma) = Mi\ min\ \Delta , \gamma = \quad \frac{Mi}{이이}$$

$$= ii \cdot 최대 \quad \frac{xi}{이이} \quad 1 \quad \beta, 0 \cdot 분 \quad \frac{이 \cdot 최대 \quad \frac{xi}{이이} \quad 1 \quad b, 0}{이이}, 씨 \quad = \quad \Delta \cdot$$

$$= ii \cdot 최대 \quad \frac{xi}{이이} \quad 1 \quad \beta, 0 \cdot 최소\ \Delta \cdot 최대 \quad \frac{xi}{이이} \quad 1 \quad b, 0 \quad , 씨$$

이러한 감소를 종합하면 소액 수수료에 대한 완전한 표현에 도달합니다.
기능:

$$
\text{있다} = \begin{cases} 0, & \frac{xi}{\text{이이}} \quad 1\beta \text{ 인 경우} \\ \Delta \cdot ii \cdot \quad \frac{xi}{\text{이이}} - 1 - \text{비} \quad ^2, & \beta < \text{인 경우} \quad \frac{xi}{\text{이이}} - 1b + \quad \overline{CD} \\ \text{씨} \cdot \text{이} \cdot \quad \frac{xi}{\text{이이}} \quad 1 \quad b, & \text{만약 } \beta + \overline{CD} < \frac{xi}{\text{이이}} \quad 1 \end{cases} \tag{4}
$$

6.4 정지 임계값

예금, 인출 및 스왑을 공식화하기 전에 효용 기능과 다소 관련이 없는 메커니즘인 정지 임계값을 설명해야 합니다. 유동성 풀은 토큰의 일반적인 시장 가격을 직접적으로 인식하지 못합니다. 보유하고 있는 예비 토큰 수와 사용자가 입금, 출금 또는 교환을 시도하는 양만 알 수 있습니다. 따라서 스테이블코인 예비금이 영구적으로 페그를 잃으면 풀은 이 스테이블코인이 이제 가치가 없다는 것을 알지 못할 것입니다. 보조 논리가 없으면 풀은 다른 모든 준비금이 고갈될 때까지 가치 없는 토큰을 계속 수락합니다. 이러한 상황을 방지하기 위해 풀에는 최소 및 최대 할당이 있습니다.

각 토큰에 대해. 각 토큰 i에 대한 임계값을 다음과 같이 정의할 수 있습니다.

$$\text{엘}_{\text{나}} = Ii \quad (1 \quad \alpha)$$

$$V_{\text{나}} = Ii \quad (1 + \alpha)$$

어디:

$$0 < \beta < \alpha < 1$$

그리고

$$\text{우리} \cdot (1 + \alpha) <$$

$\text{엘}_{\text{나}}$ 및 V i 는 각각 하한 및 상한 정지 임계값입니다. 매개변수 α는 예비 토큰 잔액에 허용되는 이상적인 포인트 위아래의 비율입니다. 풀은 결과 상태가 다음과 같은 경우에만 트랜잭션(예: 예금, 인출 및 스왑)을 실행합니다.

$$\text{오류 } i \leq xi \; VO_{\text{나}}$$

6.5 입금 및 출금

효용 기능 측면에서 풀의 논리를 정의할 때의 이점은 예금, 인출 및 스왑의 세 가지 주요 동작을 구성하는 것이 상대적으로 간단해진다는 것입니다. 이 섹션에서는 입금 및 출금에 대해 중점적으로 설명합니다. 예금의 경우 예금자는 예비 토큰을 풀에 추가하고 그 대가로 쉘을 받습니다. 따라서 풀 논리는 추가된 토큰과 생성된 셸 양 사이의 전환율을 결정해야 합니다. 인출의 경우 풀은 인출자가 상환한 쉘과 풀에서 제거된 토큰 양 사이의 전환율을 결정해야 합니다.

x = [x1, x2, . . . , xn] 은 초기 준비금이고 x = [x 1 , 예치 또는 출금 중입니다. 그러면 xx 2 , . . . , N] 토큰의 양
다음과 같은 부등식이 성립해야 합니다.

$$\frac{유(엑스)}{티} \quad T \quad \frac{U(x + x) \leq}{+ y \text{ 여기서 } T} \tag{5}$$

는 포탄의 총 공급량이며, 이 경우 y는 입출금 후 추가된 포탄의 양입니다. 이러한 불평등으로 인해 풀의 효용과 쉘 공급 사이의 비율이 항상 증가하거나 동일하게 유지되어야 합니다. 예금의 경우 새 껍질이 생성되고 y > 0입니다. 인출의 경우 껍질이 순환에서 제거되고 y < 0입니다.

방정식 5는 풀의 불변량입니다. 즉, 유효한 모든 트랜잭션에 대해 이 관계는 참으로 유지되어야 합니다. 이 불평등이 의미하는 바는 풀의 효용이 10% 증가하면 포탄 공급량은 10% 이상 증가해서는 안 된다는 것입니다. 풀의 효용이 10% 감소하면 포탄 공급량은 최소 10% 감소해야 합니다. 이 제약 조건을 준수하여 입금 및 출금 기능을 구성할 수 있습니다.

$$D(x, T, x) = \frac{티}{지(엑스) - 에프(엑스)} (G(x) - ρ (F(x + x) - F(x))) \tag{6}$$

$$W(x, T, x) = \frac{티}{지(엑스) - 에프(엑스)} (G(x) + ρ(F(x - x) - F(x))) \tag{7}$$

어디

$$p(엑스) = \begin{cases} 엑스, & x > 0 \text{인 경우} \\ λx, & \text{그렇지 않으면} \end{cases}$$

그리고

$$λ ∈ [0, 1]$$

ρ를 무시하면 방정식 6과 7은 방정식 1과 2를 방정식 5의 불변량과 결합한 결과입니다. ρ는 어디에 적합합니까? 동적 수수료라고 생각하시면 됩니다. 트랜잭션이 풀의 균형을 깨뜨리고 수수료 함수 F가 트랜잭션의 결과로 증가하면 발행(또는 상환)된 쉘은 단순히 효용 증가에 비례할 것입니다. 즉, ρ(x) = x.

그러나 트랜잭션이 풀을 재조정하고 F가 감소하면 쉘 공급의 변화가 λ만큼 상쇄됩니다. 따라서 풀 균형 조정으로 인한 페널티는 풀 균형 조정에 대한 보상보다 적습니다. 이 차이는 λ퍼센트가 됩니다. 따라서 λ = 0.5이면 동적 수수료는 가격 슬리피지의 절반이 됩니다. λ = 1이면 동적 수수료가 없습니다. 섹션 3.4에서 동적 수수료의 의미와 메커니즘에 대해 논의합니다.

6.6 스왑

수학적으로 스왑은 풀의 효용이 변경되지 않은 채 입금과 출금을 동시에 하는 것입니다(그림 ?? 참조). 직관적으로 이것은 스왑 중에 쉘 공급이 변경되지 않으므로 풀의 유틸리티도 변경되지 않기 때문에 이치에 맞습니다. 스왑을 유도하려면

공식, 먼저 음수 금액을 입금하는 것이 양수 금액을 인출하는 것과 같다는 것을 깨달으십시오.
양:

$$D(x, T, -x) = \frac{티}{지(엑스) - 에프(엑스)} (G(\quad x) \quad \rho (F(x \quad x) \quad F(x))) =$$

$$= - \frac{티}{지(엑스) - 에프(엑스)} (G(x) + \rho (F(x + x) \quad F(x))) = W(x, T, x)$$

어디:

$$x = [0, . . . , 0, x \quad ᄖ = x, 0, . . . , 0, xj \quad = -y, 0, . . . , 0]$$

따라서 셸 공급을 수정하지 않는 스왑 작업은 0개의 셸이 발행되는 입금 작업과 동일해야 합니다.

$$D(x, T, x) = 0$$

$$G(x) - \rho (F(x + x) - F(x)) = 0$$

$$y = x \quad \rho (F(x + x) \quad F(x))$$

여기서 y는 목표 금액이고 x는 원래 금액입니다. 위의 공식을 공식 3과 결합하면 스왑에 대한 다음 공식에 도달합니다.

$$Si,j (x, T, x) = x \quad \rho (F(x + x) \quad F(x)) \tag{8}$$

방정식 8은 간단하게 설명할 수 있지만 F가 y의 함수이기 때문에 분석적으로 풀기가 어렵습니다. 따라서 프로토콜은 스왑에 대한 올바른 값을 찾는 반복적인 접근 방식에 의존합니다. 방정식 8을 y의 함수 h로 다시 설명할 수 있습니다.

$$Si,j (x, T, x) = h(y)$$

그런 다음 다음과 같이 반복합니다.

$$y0 = x$$

$$yk = h(yk-1)$$

알고리즘이 수렴하면(예: yk = yk-1) 올바른 값을 찾은 것입니다.

7 향후 방향

기술적 관점에서 Shell의 최종 목표는 복잡한 스테이블 코인 시장 위에 추상화 계층을 만드는 것입니다. 이러한 프로토콜을 사용하면 최종 사용자와 개발자는 수돗물을 켤 때 배관에 대해 생각하지 않는 것과 유사하게 돈을 사용하는 것처럼 느낄 것입니다. 사용 사례와 시장 조건의 다양성을 지원하려면 일반화된 유동성 풀 모델이 필요합니다. Ethereum의 부상을 고려하십시오. 비트코인은 누가 얼마나 많은 비트코인을 소유하고 있는지 추적하는 특정한 목적을 위해 만들어진 블록체인입니다. 대조적으로 Ethereum은 범용입니다.

운영 체제 역할을 하는 블록체인. 이더리움의 일반적인 특성은 DeFi가 비트코인 대신 이더리움을 기반으로 구축되는 이유입니다. 2015년 이더리움이 출시되었을 때 DeFi가 어떤 모습일지 아무도 몰랐습니다(이 용어는 아직 발명되지도 않았습니다). Ethereum은 개발자에게 실험을 위한 플랫폼을 제공했습니다. 결과적으로 개발자는 이더리움 생태계가 유기적으로 성장할 수 있는 새롭고 예상치 못한 사용 사례를 발견했습니다.

7.1 유동성 풀 "운영 체제"

Shell의 기본 설계 목표는 유동성 풀을 위한 운영 체제를 구축하는 것입니다. 스스로 혁신적인 AMM 전략을 내놓는 것보다 다른 사람들이 자신의 창의적인 아이디어를 구현할 수 있는 플랫폼을 만드는 것이 좋습니다. 이것은 Ethereum이 혁신적인 DeFi 애플리케이션을 위한 플랫폼을 만든 방법과 유사합니다. 유동성 풀에는 많은 측면이 있지만 가장 중요한 것은 효용 기능입니다(섹션 6.3 참조). 다가오는 Shell 반복은 풀의 동작을 진정으로 사용자 정의하기 위해 점점 더 유연한 유틸리티 기능을 제공할 것입니다.

7.2 거버넌스

두 번째 설계 목표는 프로토콜 자체에 대한 거버넌스 계층입니다. 셸 프로토콜 거버넌스에 대한 심도 있는 논의는 핵심 프로토콜에 초점을 맞춘 이 백서의 범위를 벗어납니다.
Shell이 토큰 보유자 커뮤니티에 의해 관리될 것이라고 말하는 것으로 충분합니다. 이 토큰은 또한 프로토콜의 성장을 장려하는 역할을 합니다. 또한 풀은 핵심 프로토콜이 아니라 풀을 배포하고 이후에 운영하는 사람이 제어하고 관리하기 위한 것임을 언급해야 합니다. 거버넌스에서 핵심 프로토콜의 역할은 프로토콜 전체의 성장 인센티브와 같은 단일 풀의 범위를 넘어서는 조치를 조정하는 것입니다.

7.3 풀 집계

셸 프로토콜의 세 번째 설계 목표는 풀을 함께 연결하기 위한 집계 계층(온체인)을 만드는 것입니다. 모든 셸을 다른 셸로 변환할 수 있습니다. 또한 모든 풀의 유동성을 유동성 바다로 결합합니다. Shell Protocol의 진정한 가치는 이 집계 계층에 있습니다. 유연한 풀 모델을 개발하는 것은 그 자체로 중요한 단계이자 도전입니다. 그러나 궁극적인 목적지는 유동성의 바다입니다.

8 결론

이 백서에 제시된 모델은 많은 반복 중 첫 번째 모델입니다. 그것은 가중치, 깊은 유동성, 부러진 페그에 대한 보호 및 동적 수수료를 가지고 있습니다. 매개변수를 변경하여 모델의 동작을 동적으로 조정할 수 있습니다. 이를 통해 풀은 새로운 사용 사례 및 시장 조건에 적응할 수 있습니다. Shell은 궁극적으로 스테이블 코인 AMM의 운영 체제 역할을 할 수 있는 유동성 풀 모델을 만들 것입니다. 개별 스테이블 코인에 비해 쉘 토큰의 우월성을 감안할 때 쉘은 결국 가치를 저장하고 거래하는 주요 수단이 될 것입니다. 를 위한 임무

쉘 프로토콜은 인터넷 통화 시스템을 만드는 것입니다. 이 비전을 실현하려면 무수한 풀의 유동성을 결합할 집계 계층이 필요합니다. Shell의 거버넌스는 개별 풀을 관리하기 위해 존재하는 것이 아니라 유동성 바다를 감독하기 위해 존재합니다.

참조

[1] Aave(2020년 9월 4일). "Aave FAQ." https://docs.aave.com/faq/

[2] 아담스, 헤이든 (2019년 7월 5일). "Uniswap 백서". https://hackmd.io/C-DvwDSfSxuh-Gd4WKE ig#DEX-inside-a-백서

[3] 바이낸스 아카데미 (2020년 9월 4일). "(DeFi)에서 수확량 농업이란 무엇입니까?" https://academy.binance.com/blockchain/ 탈중앙화 재원 what-is-yield-farming-in-decentralized-finance-defi _

[4] 부테린, 비탈릭(2018년 3월). "x*y=k 시장 메이커의 선행 저항 개선". ETH 리서치. 2019 년 11월 24일 검색: https://ethresear.ch/t/proving-front-running-resistance-of-xyk-market-maker/1281

[5] 서클(2020). "USDC: 가장 빠르게 성장하고 있으며 완전히 예약된 디지털 달러 스테이블 코인입니다." 회수됨 2020년 9월 27일. https://www.circle.com/en/usdc

[6] 마이클 코헨(2020년 4월). "sUSD의 부상." https://synthetix.community/blog/2020/04/13/the-rise-of-susd _

[7] 코인마켓캡(2020년 8월). https://coinmarketcap.com/

[8] 디파이 펄스(2020년 9월 4일). "DeFi 펄스: 컴파운드". 2020년 9월 4일에 확인함. https://defipulse.com/compound

[9] 디파이 펄스(2020년 9월 27일). "DeFi 펄스: 커브 파이낸스". 2020년 9월에 확인함 27. https://defipulse.com/curve-finance

[10] De la Rouviere, Simon (2017년 11월 21일). "토큰 2.0: 큐레이션 시장의 곡선형 토큰 본딩". 검색됨, 2020년 9월 27 일 .

[11] 드 니킬레쉬(2020년 9월 21일). "SEC, OCC, Stablecoins에 대한 첫 번째 규제 설명을 발행합니다." https://www.coindesk.com/occ-banks-can-hold-some-stablecoin-reserves

[12] 도겐(1240). 산수경 (山水經). Arnold Kotler가 번역하고 타나하시 카즈아키(2008). 해켓 출판사.

[13] 마이클 에고로프(2019년 11월 10일). "StableSwap - Stablecoin을 위한 효율적인 메커니즘 유동성." https://www.curve.fi/stableswap-paper.pdf

[14] 페드와이어(2020). "Fedwire 자금 서비스 - 월별 통계." https://www.frbservices.org/resources/financial-services/wires/volume-value-stats/monthly-stats.html

[15] 엠마 후틴슨(2016). 미시경제학의 원리. 6.2장: 무차별 곡선 빅토리아 대학교. https://pressbooks.bccampus.ca/uvicecon103/chapter/ 6-3-how-changes-in-income-and-prices-affect-consumption-choices/

[16] Kereiakes 외(2019년 4월). "테라머니: 안정성과 채택". https://s3. ap-northeast-2.amazonaws.com/terra.money.home/static/Terra White paper.pdf?201904 _ _

[17] Leshner, Robert 및 Geoffrey Hayes(2019년 2월). “화합물: 화폐 시장 규약". https://compound.finance/documents/Compound.Whitepaper.pdf

[18] 메이커다오(2020). “메이커 프로토콜: MakerDAO의 다중 담보 다이(MCD) 시스템 그는 가지고있다." https://makerdao.com/en/whitepaper/#abstract

[19] Martinelli, Fernando 및 Nikolai Mushegian (2019년 9월 19일). "밸런서: 비수탁 포트폴리오 관리자, 유동성 공급자 및 가격 센서." https://balancer.finance/whitepaper.html _

[20] mStable(2020). 2020년 9월 29일 액세스. https://mstable.org/

[21] 사토시 나카모토(2008년 8월). "비트코인: P2P 전자 현금 시스템". https://bitcoin.org/bitcoin.pdf

[22] 오핀(2020년 9월). “탈중앙화 금융 확보.” https://opyn.co/

[23] 팍소스(2020). “새로운 디지털 달러.” 2020년 9월 27일에 확인함. https://www.paxos.com/pax/

[24] Peter, Bianca (2019년 9월). “예금 및 자산별 은행 시장점유율.” https://wallethub.com/edu/sa/bank-market-share-by-deposits/25587/

[25] 스워브 파이낸스(2020). 2020년 9월 27일에 확인함. https://swerve.fi/

[26] 닉 사보(2017년 2월). "돈, 블록체인, 사회적 확장성." http://unnumerated.blogspot.com/2017/02/money-blockchains-and-social-scalability.html _

[27] 테더(2016년 6월). "테더 백서". https://tether.to/wp-content/uploads/2016/06/테더백서.pdf

[28] 미국 국무부: 역사가 사무실(2018). “닉슨과 브레튼우즈 체제의 종말, 1971-1973”. 2019 년 7월 21일 검색: https://history.state.gov/milestones/1969-1976/nixon-shock

[29] yEarn 홈페이지(2020년 9월 4일). https://yearn.finance/

[30] 재퍼 홈페이지(2020년 9월 30일). “DeFi 자산과 부채를 하나로 관리 간단한 인터페이스.”. https://zapper.fi/