I've been thinking a lot after the recent Maker drama about how no algorithmic stablecoins have ever gained much (any?) traction. I believe that part of the reason for this is all attempts/proposals for an algorithmic model start out as 0% backed stablecoin that's doomed to instill any kind of confidence in users on day 1.

So I thought of a hybrid design that I think has a good chance of potential success and wanted the community's feedback. It starts as standard shares and coins 2 token system like Robert Sams' original idea but at first backed by either Dai/USDC/USDT.

1. The system starts 1 to 1 backed by Dai/USDT/USDC. Put in 1 USDC in the contract, get 1 stablecoin. This can always be redeemed back from the contract 1 to 1 at any time. Some share tokens are distributed/airdropped. Maybe share tokens can even be given away to people who bootstrap the system by minting the first stablecoins.

2. After a certain threshold market cap is reached, the system slowly moves to the seigniorage shares model in increments.

3. Every X blocks, Y coins are minted and auctioned for shares, increasing the supply of coins compared to the collateral in the contract.

4. If the price of coins remains stable, then step 3 repeats.

5. If the price of coins drop, then shares are minted to buy back Y coins and recover the price.

6. Step 3-5 repeat until the system is sufficiently algorithmic and virtually seigniorage shares so that the amount of collateral left in the contract is trivial. Or essentially the system remains at whatever fractional-collateral ratio the market supports to keep the price of coins at $1. Perhaps the market only has sufficient confidence that 30% of the coin supply can be stabilized algorithmically and the rest requires collateral. The system would remain in that band through steps 3-5 repeating.

Some problems with this setup:

- Requires a price feed for the coin and share tokens still. The onchain auctions aren't sufficient enough to discover the price of each token. Perhaps share token holders can vote for price feeds a la MKR-type governance

- The collateral must be non-volatile crypto (Dai, USDC, USDT, but not ETH)

- There are some attack vectors in the redemption process whereby once the system is part algorithmic, a large actor could redeem a lot of coins for all of the collateral at 1 to 1 rates. A fix for this would be to change redemption during the algorithmic phase so that you can redeem partly for collateral and partly for newly minted shares. Ex: If the system is only 70% backed, then 1 coin is redeemable for 70% collateral and 30% shares to keep the ratio from deviating.

Thoughts and feedback? Improvements?