Shelley Reward Improvements Proposal

Laurent (PARIS) – 1 February 2021

This short document tries to gather, compare, and combine a few ideas of improvement for the reward formula in Shelley, in the view of improving Sybil-attack resistance, and fairness of risk/reward incentives for SPO.

1) The standard Shelley Reward function

The current effective theoretical reward per pool and per epoch is defined as:

$$R(\sigma, s) \coloneqq \frac{RwdPot}{1 + a_0}\left(\sigma' + a_0 * s' * \frac{\sigma' - s' * \frac{z_0 - \sigma'}{z_0}}{z_0}\right)$$

Where:

- $RwdPot$ is the total reward pot per epoch
- a_0 is a protocol parameter (currently valued at 0,3)
- $z_0 = 1/k$ is the saturation ratio wrt total stake (currently k=500, soon k=1000)
- σ is the pool stake divided by total global stake (incl both owner pledge and delegations)
- s is the owner pledge divided by total global stake
- $\sigma' = \min(\sigma, z_0)$
- $s' = \min(s, z_0)$

A simpler and equivalent way of expressing this formula is by using the notions of pool saturation ratio and pool pledge ratio.

The pool saturation ratio is defined as the ratio between capped pool stake, and saturation level:

$$S \coloneqq \frac{\sigma'}{z_0} = \frac{\min(\sigma, z_0)}{z_0}$$

The pool pledge ratio is defined as the ratio between capped pool pledge and capped pool stake:

$$P \coloneqq \frac{s'}{\sigma'} = \frac{\min(s, z_0)}{\min(\sigma, z_0)}$$

We also define as the pool leverage the inverse of the pool pledge ratio:

$$L \coloneqq \frac{1}{P} = \frac{\min(\sigma, z_0)}{\min(s, z_0)}$$

The reward formula can then be rewritten as:

$$\hat{R}(S,P,\sigma') = RwdPot * \sigma' * \frac{1 + a_0 * P * S * \big(1 - P * (1 - S)\big)}{1 + a_0}$$

We can then observe that for a given pool stake σ', the maximum reachable theoretical reward is equal to $RwdPot * \sigma'$ which simply represents the "perfect" reward share for the pool among the available reward pot for the epoch.

Let's then introduce the notion of pool efficiency as the ratio between the reward and the "perfect" reward:

$$E(S,P) \coloneqq \frac{\hat{R}(S,P,\sigma')}{\max\limits_{S,P} \hat{R}(S,P,\sigma')}$$

For the standard Shelley reward we get the following expression of pool efficiency:

$$E(S,P) = \frac{1 + a_0 * P * S * \big(1 - P * (1 - S)\big)}{1 + a_0}$$

It is easy to prove that the pool efficiency is maximal (i.e. 100%) if and only if the pool is fully saturated and fully pledged, i.e. when $P = S = 100\%$.

The efficiency of a fully saturated pool (but not necessarily fully pledged) is also interesting:

$$E(100\%,P) = \frac{1 + a_0 * P}{1 + a_0}$$

We see that the efficiency of a saturated pool is an affine function of the pledge ratio controlled by the parameter a_0. The higher this parameter is, the more the efficiency depends on the pledge ratio. On the contrary, when it is small the pledge ratio has no more effect on the efficiency as it is maximal regardless the pledge.

The efficiency of a fully pledged pool (but not necessarily fully saturated, i.e. the pool as no delegation outside the owner stake itself) is equal to:

$$E(S,100\%) = \frac{1 + a_0 * S^2}{1 + a_0}$$

This time, the efficiency is an affine function of the square of the saturation ratio. This nonlinearity tends to relatively advantage more the big pools vs the small pools wrt to the incentive of a high pledge ratio.

A last case is interesting, is for small pool for which saturation is far smaller than 100%.

For them, the efficiency can be approximated in a simpler form as $S << 1$:

$$E(S,P) \sim \frac{1 + a_0 * S * P * (1 - P)}{1 + a_0}$$

As evoked above, even though for such small pools, the influence of the pledge ratio is small (due to the huge amortization effect of the low saturation ratio) it is yet interesting to observe that the efficiency will be maximum for a pledge ratio equal to $P = 50\%$.

In other words, when an operator creates a pool and is not expecting a lot of delegation to start with (which is the case of most of Single Pool Operators) it is more efficient for him to split its stake 50% as a declared "pledge" and the other 50% as a "delegator" (from another wallet).

As the total pool stake will grow with new delegators, it will then be optimal to slowly increase the pledge ratio from 50% up to 100% if its personal stake allows it (otherwise, just put as much as you can in the pledge when you reach a substantial saturation level in your pool; S=50% being the threshold point).

Another notion that is interesting to have in mind is the Reward relative benefit of pledging compared to not pledging at all. It is simply defined as:

$$B(P,S) \coloneqq \frac{E(S,P) - E(S,0)}{E(S,0)}$$

And for the shelley standard reward it reads:

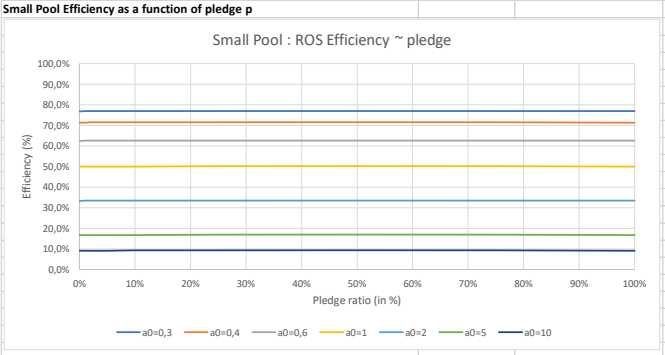
$$B(P,S) = a_0 * P * S * \big(1 - P * (1 - S)\big)$$

Below you can find some numerical illustrations of the reward formula in several situations, and the impact of the parameter a0:

Pool ROS Standard Efficiency : E(p,s)=(1+a0*p*s*(1-p*(1-s))) / (1+a0)	
a0	0,3
p=Pledge Amount / Stake Amount	
s=Saturation = Stake Amount / Saturation Level	

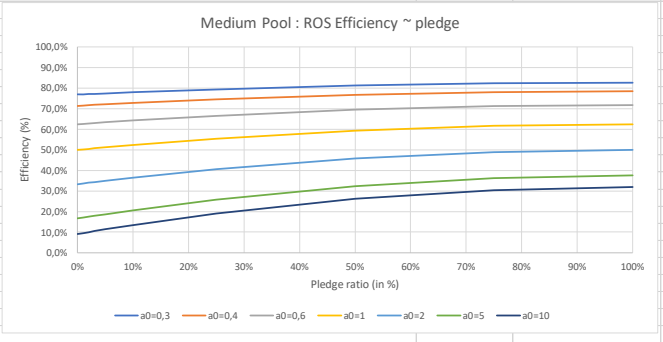
	Efficiency Table	P									
	E(p,s)	0%	1%	2%	3%	5%	10%	25%	50%	75%	100%
S	0%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%
	1%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	77,0%	77,0%	77,0%	76,9%
	2%	76,9%	76,9%	76,9%	76,9%	76,9%	77,0%	77,0%	77,0%	77,0%	76,9%
	3%	76,9%	76,9%	76,9%	76,9%	77,0%	77,0%	77,1%	77,1%	77,1%	76,9%
	5%	76,9%	76,9%	76,9%	77,0%	77,0%	77,0%	77,1%	77,2%	77,2%	77,0%
	10%	76,9%	76,9%	77,0%	77,0%	77,0%	77,1%	77,4%	77,6%	77,5%	77,2%
	25%	76,9%	77,0%	77,0%	77,1%	77,2%	77,5%	78,1%	78,7%	78,8%	78,4%
	50%	76,9%	77,0%	77,2%	77,3%	77,5%	78,0%	79,4%	81,3%	82,3%	82,7%
	75%	76,9%	77,1%	77,3%	77,4%	77,8%	78,6%	81,0%	84,5%	87,5%	89,9%
	100%	76,9%	77,2%	77,4%	77,6%	78,1%	79,2%	82,7%	88,5%	94,2%	100,0%

Small Pool Efficiency as a function of pledge p



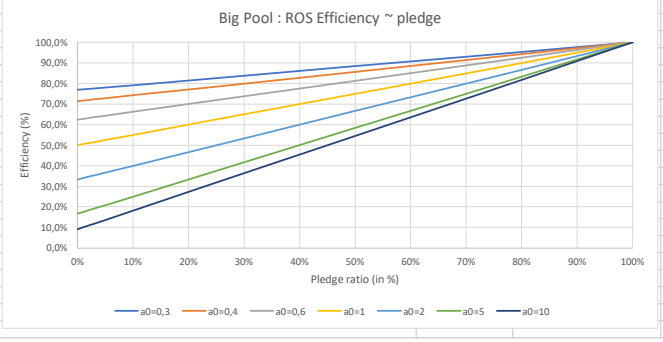
Small Pool	a0						
E(p,1%)	0,3	0,4	0,6	1	2	5	10
0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
1%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
2%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
3%	76,9%	71,4%	62,5%	50,0%	33,4%	16,7%	9,1%
5%	76,9%	71,4%	62,5%	50,0%	33,4%	16,7%	9,1%
10%	76,9%	71,5%	62,5%	50,0%	33,4%	16,7%	9,2%
25%	77,0%	71,5%	62,6%	50,1%	33,5%	16,8%	9,3%
50%	77,0%	71,5%	62,6%	50,1%	33,5%	16,9%	9,3%
75%	77,0%	71,5%	62,6%	50,1%	33,5%	16,8%	9,3%
100%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%

Medium Pool Efficiency as a function of pledge p



Medium Pool	a0						
E(p,50%)	0,3	0,4	0,6	1	2	5	10
0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
1%	77,0%	71,6%	62,7%	50,2%	33,7%	17,1%	9,5%
2%	77,2%	71,7%	62,9%	50,5%	34,0%	17,5%	10,0%
3%	77,3%	71,9%	63,1%	50,7%	34,3%	17,9%	10,4%
5%	77,5%	72,1%	63,4%	51,2%	35,0%	18,7%	11,3%
10%	78,0%	72,8%	64,3%	52,4%	36,5%	20,6%	13,4%
25%	79,4%	74,6%	66,6%	55,5%	40,6%	25,8%	19,0%
50%	81,3%	76,8%	69,5%	59,4%	45,8%	32,3%	26,1%
75%	82,3%	78,1%	71,3%	61,7%	49,0%	36,2%	30,4%
100%	82,7%	78,6%	71,9%	62,5%	50,0%	37,5%	31,8%

Big Pool Efficiency as a function of pledge p



Big Pool	a0						
E(p,100%)	0,3	0,4	0,6	1	2	5	10
0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
1%	77,2%	71,7%	62,9%	50,5%	34,0%	17,5%	10,0%
2%	77,4%	72,0%	63,3%	51,0%	34,7%	18,3%	10,9%
3%	77,6%	72,3%	63,6%	51,5%	35,3%	19,2%	11,8%
5%	78,1%	72,9%	64,4%	52,5%	36,7%	20,8%	13,6%
10%	79,2%	74,3%	66,3%	55,0%	40,0%	25,0%	18,2%
25%	82,7%	78,6%	71,9%	62,5%	50,0%	37,5%	31,8%
50%	88,5%	85,7%	81,3%	75,0%	66,7%	58,3%	54,5%
75%	94,2%	92,9%	90,6%	87,5%	83,3%	79,2%	77,3%
100%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Even if this formula makes a lot of sense and is always quite well thought to counter Sybil attacks, we can still observe a few problems (which are not only theoretical as they tend to appear for real!)

- **Problem #1**: very low pledged (i.e. highly leveraged) big pools do not have enough efficiency penalization with a0. More precisely if increasing a0 a lot would indeed penalize them a lot, it would also penalize even more the smaller pools, despite their ability to lock a very decent pledge amount. This is due to the “scaling effect” of low saturation ratio, and even though this effect should be reduced in the future by the increase in K, it will persist very significantly. This problem today is a reality, as it incites big operators to split their total pool stakes in multiple pools, splitting by the way their initial pledge into several smaller chunks without being more penalized than any other small pool. As a consequence, smaller single pool operators tend to have more and more difficulty to engage growth and attract delegations up to an “escape velocity” level. The risk on a longer term is that only a group of a few operators will operate multiple pools and put in danger the decentralized nature of the network under the form of a “cartel”.

- **Problem #2**: an operator (or owner) with a very high personal stake is clearly incited to create a fully saturated fully pledged pool (i.e. a private pool!) as maximum reward efficiency is then naturally reached. This phenomenon will tend to be bigger and bigger with the increase of K, as it will then be easier and easier to "saturate" a pool. Even though it represents less risk than in problem #1, it is still against the idea of a "healthy" decentralized and open network as it will be equivalent to "split" the network in a public community-based subset and a fully private subset.

In next sections I will introduce two improvements proposals for each of the problem and a way to "combine" them to both address the two problems at the same time.

To solve problem #1, I propose a variant called **"Skewed Reward"** using a new additional protocol parameter $\boldsymbol{b_0}$.

To solve problem #2, I will re-expose the idea by **Shawn McMurdo** (available in details here : https://github.com/cardano-foundation/CIPs/pull/12) of a variant called **"Curved Reward Benefit"** using two new additional protocol parameter $\boldsymbol{c_r}$ and $\boldsymbol{c_f}$.

2) The skewed Shelley Reward function (Idea #1)

To solve problem #1, I propose to extend the standard Shelley reward formula as below:

$$sk\hat{R}(S,P,\sigma') = RwdPot * \sigma' * \frac{1 + (a_0 * P + b_0) * S * \left(1 - P * (1 - S)\right)}{1 + a_0 + \frac{b_0}{P}}$$

With corresponding efficiency function:

$$skE(S,P) = \frac{1 + (a_0 * P + b_0) * S * \left(1 - P * (1 - S)\right)}{1 + a_0 + \frac{b_0}{P}}$$

Where b_0 is a new protocol parameter that is non negative (and will usually be chosen as a quite low value, e.g. from 0.0001 to 0.01).

The idea of this modification is to significantly penalize the efficiency of very low pledged pool (relatively to stake) (i.e. very highly leveraged pools) in order to try to solve all the issues described in problem #1.

First, we can immediately observe that for $b_0 = 0$ the formula is equivalent to the standard version (so it is backward compatible).

Second, we can quickly see that for $b_0 > 0$ and for a low level of pledge ratio, i.e. $P << 1$, we get:

$$skE(S,P) \approx P\frac{1 + b_0 * S * \big(1 - P * (1 - S)\big)}{b_0} \leq P\frac{1 + b_0}{b_0}$$

So the efficiency collapses to zero when the pledge ratio is not high enough to counter the effect of the new parameter b_0.

This means that an operator with a finite (even big) global pledge capacity will be **discouraged to divide its pledge over a lot of multiple pools** as this would then totally ruin its reward efficiency, make them loose a lot of delegations until their pool saturation becomes low enough to "deleverage" their pools. But then, the operator will have more interest to merge back into less pools far more reasonably leveraged (most of them will land at having 2 or 3 pools max).

Moreover, for pools operated today with "almost no pledge" but an immense capacity to attract delegations (may it be an exchange or any other "big actor"), this new skew effect will force them to deleverage by pledging each pool sufficiently well, which will inevitably limit the number of pools they will be willing to maintain. With such a skew, biggest actors should be attracted then by running full private pools as they will then maximize their efficiency. (this phenomenon leads to problem #2 solved by the curved reward).

Let's now look at various situations, beginning with a big (saturated) pool:

$$skE(100\%,P) = \frac{1 + b_0 + a_0 * P}{1 + a_0 + \frac{b_0}{P}}$$

So, a big pool reaches maximum efficiency with maximum pledge ratio (as in standard version) and is dramatically penalized for not ensuring a decent level of pledge wrt b_0.

For a small pool with $S << 1$, we get:

$$skE(S,P) \approx \frac{1 + S * (a_0 * P + b_0) * (1 - P)}{1 + a_0 + \frac{b_0}{P}} \approx \frac{1}{1 + a_0 + \frac{b_0}{P}}$$

And we can see another quality of the skew factor that even a small pool substantially benefits from maintaining a good pledge ratio, i.e. is incentivized to have skin in the game.

These two corrections (highly leveraged big pools penalization) + (better pledge incentive for small pools) provided by the **skew effect** allows to **adjust the balance between small and big actors in the way of improving the decentralization and the Sybil attack resistance of the whole network**.

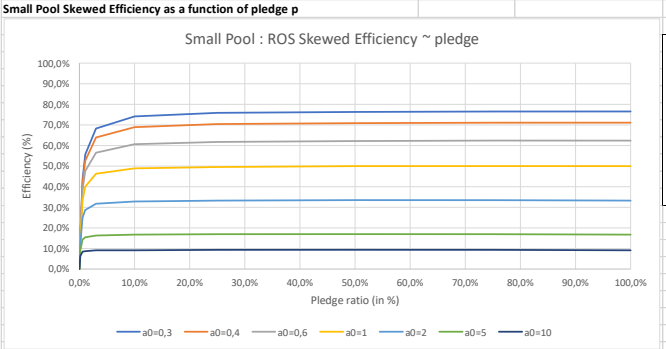
Finally, if parameter b_0 is well chosen, (not too high) it should not affect too much "medium" actors that are today running one/two pools (or one very saturated). In the view of a K increase, they should be able to divide their pledge in a reasonable way so that they keep as much stake delegations as they can without deterioring too much their reward efficiency. Anyway, thanks to the skew, a natural "equilibrium" will occur.

Below you can find some numerical illustrations of the skewed reward formula in several situations, and the impact of the parameters a0, and b0:

Pool ROS Skewed Efficiency : skE(p,s)=(1+(a0+b0/p)*p*s*(1-p*(1-s))) / (1+a0+b0/p)	
a0	0,3
b0	0,005
p=Pledge Amount / Stake Amount	
s=Saturation = Stake Amount / Saturation Level	

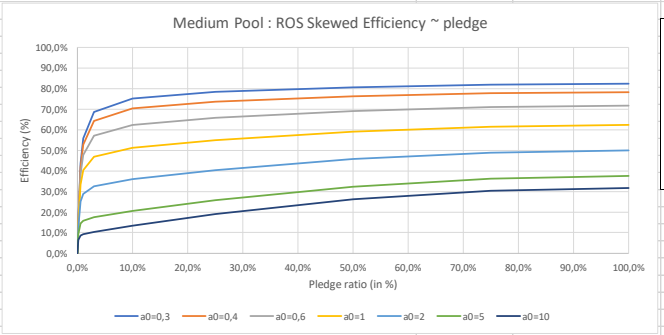
	Efficiency Table	P									
		0%	0,1%	0,5%	1%	3%	10%	25%	50%	75%	100%
S	0%	0,0%	15,9%	43,5%	55,6%	68,2%	74,1%	75,8%	76,3%	76,5%	76,6%
	1%	0,0%	15,9%	43,5%	55,6%	68,2%	74,1%	75,8%	76,4%	76,6%	76,6%
	2%	0,0%	15,9%	43,5%	55,6%	68,2%	74,1%	75,8%	76,5%	76,6%	76,6%
	3%	0,0%	15,9%	43,5%	55,6%	68,2%	74,1%	75,9%	76,5%	76,7%	76,6%
	5%	0,0%	15,9%	43,5%	55,6%	68,2%	74,2%	76,0%	76,6%	76,8%	76,7%
	10%	0,0%	15,9%	43,5%	55,6%	68,3%	74,3%	76,2%	77,0%	77,1%	76,9%
	25%	0,0%	15,9%	43,5%	55,7%	68,4%	74,7%	77,0%	78,2%	78,5%	78,1%
	50%	0,0%	15,9%	43,6%	55,8%	68,7%	75,3%	78,4%	80,8%	82,0%	82,5%
	75%	0,0%	15,9%	43,7%	55,9%	68,9%	76,0%	80,0%	84,1%	87,3%	89,8%
	100%	0,0%	16,0%	43,8%	56,0%	69,1%	76,7%	81,8%	88,2%	94,1%	100,0%

Small Pool Skewed Efficiency as a function of pledge p



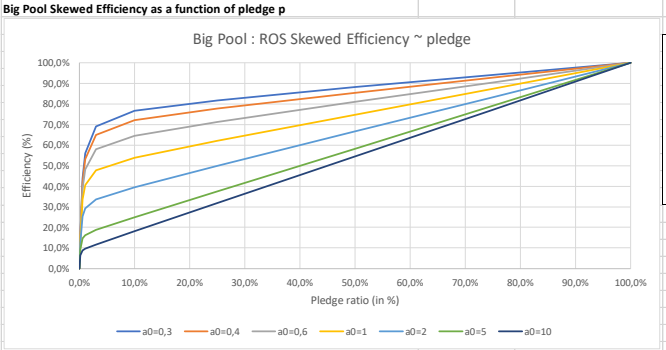
Small Pool	a0						
skE(p,1%)	0,3	0,4	0,6	1	2	5	10
P 0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,1%	15,9%	15,6%	15,2%	14,3%	12,5%	9,1%	6,3%
0,5%	43,5%	41,7%	38,5%	33,3%	25,0%	14,3%	8,3%
1,0%	55,6%	52,6%	47,6%	40,0%	28,6%	15,4%	8,7%
3,0%	68,2%	63,8%	56,6%	46,2%	31,6%	16,2%	9,0%
10,0%	74,1%	69,0%	60,6%	48,8%	32,8%	16,6%	9,1%
25,0%	75,8%	70,5%	61,8%	49,6%	33,2%	16,8%	9,2%
50,0%	76,4%	71,0%	62,2%	49,9%	33,4%	16,8%	9,3%
75,0%	76,6%	71,1%	62,3%	49,9%	33,4%	16,8%	9,3%
100,0%	76,6%	71,2%	62,3%	49,9%	33,3%	16,7%	9,1%

Medium Pool Skewed Efficiency as a function of pledge p



Medium Pool	a0						
skE(p,50%)	0,3	0,4	0,6	1	2	5	10
P 0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,1%	15,9%	15,7%	15,2%	14,3%	12,5%	9,1%	6,3%
0,5%	43,6%	41,8%	38,6%	33,5%	25,2%	14,5%	8,6%
1,0%	55,8%	52,9%	47,9%	40,3%	28,9%	15,8%	9,1%
3,0%	68,7%	64,4%	57,2%	46,9%	32,6%	17,5%	10,3%
10,0%	75,3%	70,4%	62,5%	51,2%	36,0%	20,5%	13,4%
25,0%	78,4%	73,7%	65,9%	55,0%	40,4%	25,7%	19,0%
50,0%	80,8%	76,4%	69,2%	59,2%	45,7%	32,3%	26,1%
75,0%	82,0%	77,9%	71,1%	61,6%	48,9%	36,2%	30,4%
100,0%	82,5%	78,4%	71,7%	62,4%	50,0%	37,5%	31,8%

Big Pool Skewed Efficiency as a function of pledge p



Big Pool	a0						
skE(p,100%)	0,3	0,4	0,6	1	2	5	10
P 0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,1%	16,0%	15,7%	15,2%	14,4%	12,6%	9,2%	6,3%
0,5%	43,8%	42,0%	38,8%	33,7%	25,4%	14,7%	8,8%
1,0%	56,0%	53,1%	48,1%	40,6%	29,3%	16,2%	9,6%
3,0%	69,1%	64,9%	57,9%	47,8%	33,6%	18,7%	11,7%
10,0%	76,7%	72,1%	64,5%	53,9%	39,5%	24,9%	18,1%
25,0%	81,8%	77,8%	71,3%	62,1%	49,8%	37,5%	31,8%
50,0%	88,2%	85,5%	81,1%	74,9%	66,6%	58,3%	54,5%
75,0%	94,1%	92,8%	90,6%	87,5%	83,3%	79,2%	77,3%
100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

The Reward Benefit is not defined for the skewed formula as the reward is namely zero when having no pledge at all. Nevertheless, you can see in examples above how the skew factor modifies the efficiency of the reward.

3) The curved Shelley Reward function (Idea #2)

To solve problem #2 Shawn McMurdo proposed another extension of the standard Shelley reward formula under the form of a "curved" reward function:

$$cv\hat{R}(S,P,\sigma') = RwdPot * \sigma' * \frac{1 + a_0 * c(P) * S * \big(1 - c(P) * (1 - S)\big)}{1 + a_0}$$

Where the curve subfunction is defined as:

$$c(P) := \frac{P^{\frac{1}{c_r}}}{{c_f}^{1-\frac{1}{c_r}}}$$

Where:

- c_r is a new protocol parameter (curve root) >=1
- c_f is a new protocol parameter (curve crossover factor) >=1

The curve root will have the effect to "incurve" the efficiency and reward benefit functions for lower values of P and the curve crossover factor will have the effect to decrease the efficiency benefit for higher levels of P, thus countering by the way the incentive for very highly staked actors to create private pools, which is exactly a solution to problem #2.

The corresponding efficiency function:

$$cvE(S,P) = \frac{1 + a_0 * c(P) * S * \big(1 - c(P) * (1 - S)\big)}{1 + a_0}$$

It is important to observe that **the curve function does not solve at all problem #1**, as curve will namely have almost no "effect" on small pools, and will not change at all the efficiency of very low pledged big pools, even though it will offer a better marginal incentive to increase that pledge, without penalizing the very low pledge (i.e. very high leveraged pools).

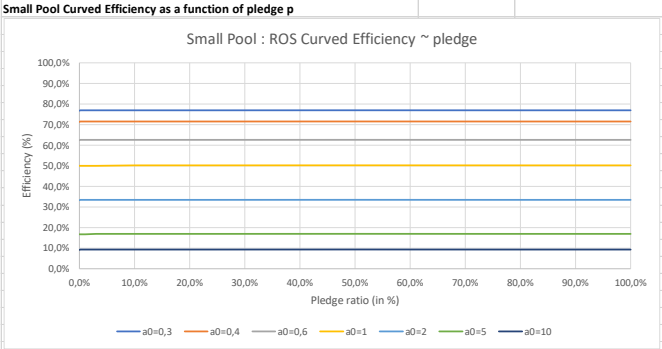
I will not discuss more all the advantages of the curve effect as it has already been well documented by its author.

You'll find below numerical examples as well for the curve reward:

Pool ROS Curved Efficiency : cvE(p,s)=(1+a0*c(p)*s*(1-c(p)*(1-s))) / (1+a0)	
a0	0,3
cr	3
cf	2
curve : c(p)=p^(1/cr) / cf^(1-1/cr)	
p=Pledge Amount / Stake Amount	
s=Saturation = Stake Amount / Saturation Level	

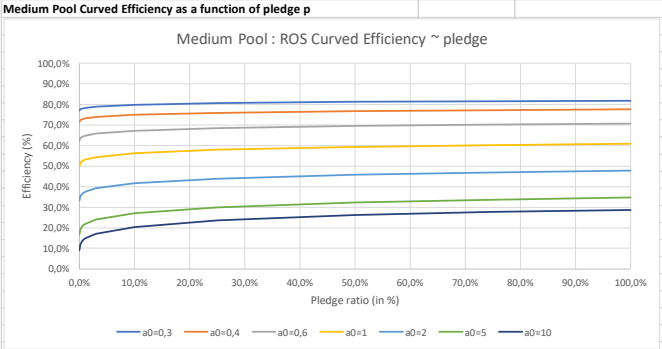
	Efficiency Table	P									
	cvE(p,s)	0%	0,1%	0,5%	1%	3%	10%	25%	50%	75%	100%
S	0%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%	76,9%
	1%	76,9%	76,9%	76,9%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%
	2%	76,9%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%	77,0%
	3%	76,9%	77,0%	77,0%	77,0%	77,0%	77,1%	77,1%	77,1%	77,1%	77,1%
	5%	76,9%	77,0%	77,0%	77,1%	77,1%	77,2%	77,2%	77,2%	77,2%	77,2%
	10%	76,9%	77,1%	77,1%	77,2%	77,3%	77,4%	77,5%	77,6%	77,6%	77,6%
	25%	76,9%	77,3%	77,5%	77,6%	77,9%	78,2%	78,5%	78,7%	78,8%	78,8%
	50%	76,9%	77,6%	78,1%	78,4%	79,0%	79,8%	80,6%	81,3%	81,6%	81,9%
	75%	76,9%	78,0%	78,7%	79,2%	80,1%	81,6%	83,1%	84,5%	85,4%	86,1%
	100%	76,9%	78,4%	79,4%	80,1%	81,4%	83,7%	86,1%	88,5%	90,1%	91,5%

Small Pool Curved Efficiency as a function of pledge p



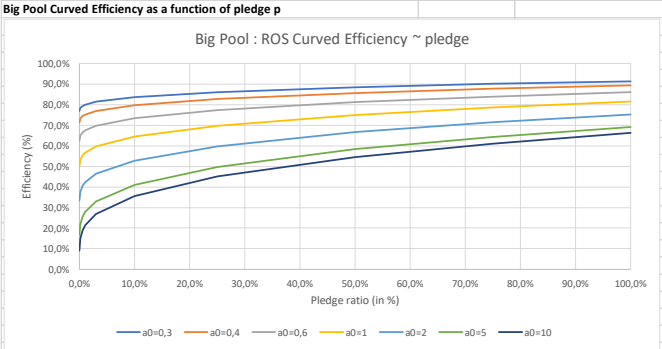
Small Pool	a0							
cvE(p,1%)	0,3	0,4	0,6	1	2	5	10
0,0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
0,1%	76,9%	71,4%	62,5%	50,0%	33,4%	16,7%	9,1%
0,5%	76,9%	71,5%	62,5%	50,0%	33,4%	16,7%	9,2%
1,0%	77,0%	71,5%	62,5%	50,1%	33,4%	16,8%	9,2%
3,0%	77,0%	71,5%	62,6%	50,1%	33,4%	16,8%	9,2%
10,0%	77,0%	71,5%	62,6%	50,1%	33,5%	16,8%	9,3%
25,0%	77,0%	71,5%	62,6%	50,1%	33,5%	16,9%	9,3%
50,0%	77,0%	71,5%	62,6%	50,1%	33,5%	16,9%	9,3%
75,0%	77,0%	71,5%	62,6%	50,1%	33,5%	16,9%	9,3%
100,0%	77,0%	71,5%	62,6%	50,1%	33,5%	16,9%	9,3%

Medium Pool Curved Efficiency as a function of pledge p



Medium Pool	a0							
cvE(p,50%)	0,3	0,4	0,6	1	2	5	10
0,0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
0,1%	77,6%	72,3%	63,6%	51,5%	35,4%	19,2%	11,9%
0,5%	78,1%	72,9%	64,4%	52,5%	36,7%	20,9%	13,7%
1,0%	78,4%	73,2%	64,9%	53,2%	37,6%	21,9%	14,8%
3,0%	79,0%	74,0%	65,8%	54,4%	39,2%	24,0%	17,1%
10,0%	79,8%	75,0%	67,2%	56,2%	41,7%	27,1%	20,4%
25,0%	80,6%	76,0%	68,5%	58,0%	43,9%	29,9%	23,6%
50,0%	81,3%	76,8%	69,5%	59,4%	45,8%	32,3%	26,1%
75,0%	81,6%	77,3%	70,2%	60,2%	47,0%	33,7%	27,7%
100,0%	81,9%	77,6%	70,6%	60,8%	47,7%	34,6%	28,7%

Big Pool Curved Efficiency as a function of pledge p



Big Pool	a0							
cvE(p,100%)	0,3	0,4	0,6	1	2	5	10
0,0%	76,9%	71,4%	62,5%	50,0%	33,3%	16,7%	9,1%
0,1%	78,4%	73,2%	64,9%	53,1%	37,5%	21,9%	14,8%
0,5%	79,4%	74,5%	66,5%	55,4%	40,5%	25,6%	18,9%
1,0%	80,1%	75,3%	67,6%	56,8%	42,4%	28,0%	21,4%
3,0%	81,4%	77,0%	69,8%	59,8%	46,4%	33,0%	26,9%
10,0%	83,7%	79,8%	73,5%	64,6%	52,8%	41,0%	35,7%
25,0%	86,1%	82,8%	77,4%	69,8%	59,8%	49,7%	45,2%
50,0%	88,5%	85,7%	81,3%	75,0%	66,7%	58,3%	54,5%
75,0%	90,1%	87,8%	84,0%	78,6%	71,5%	64,4%	61,1%
100,0%	91,5%	89,4%	86,1%	81,5%	75,3%	69,2%	66,4%

4) The Skewed & Curved Shelley Reward function (Improvement #3)

To solve both problem #1 and problem #2 it is possible to combine both solutions and have a hybrid variant defined as below:

$$skcv\hat{R}(S,P,\sigma') = RwdPot * \sigma' * \frac{1 + (a_0 * c(P) + b_0) * S * \big(1 - c(P) * (1 - S)\big)}{1 + a_0 + \frac{b_0}{c(P)}}$$

With corresponding efficiency function:

$$skcvE(S,P) = \frac{1 + (a_0 * c(P) + b_0) * S * \big(1 - c(P) * (1 - S)\big)}{1 + a_0 + \frac{b_0}{c(P)}}$$

And curve subfunction

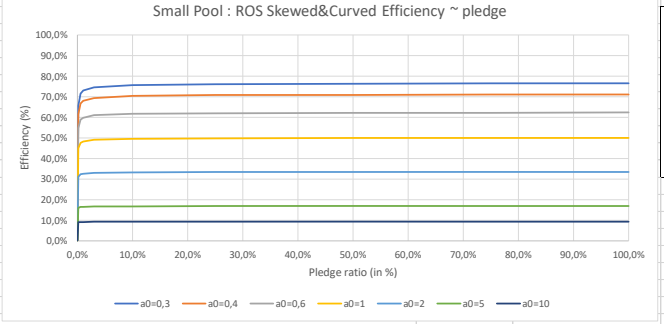
$$c(P) := \frac{P^{\frac{1}{c_r}}}{{c_f}^{1-\frac{1}{c_r}}}$$

With the help of the three new parameters (skew factor b_0, curve root c_r, and curve crossover factor c_f) we can see in numerical illustrations below that both problems are indeed solved.

Pool ROS Skewed&Curved Efficiency : skcvE(p,s)=(1+(a0+b0/p)*p*s*(1-p*(1-s))) / (1+a0+b0/p)	
a0	0,3
b0	0,005
cr	2
cf	2
curve : c(p)=p^(1/cr) / cf^(1-1/cr)	
p=Pledge Amount / Stake Amount	
s=Saturation = Stake Amount / Saturation Level	

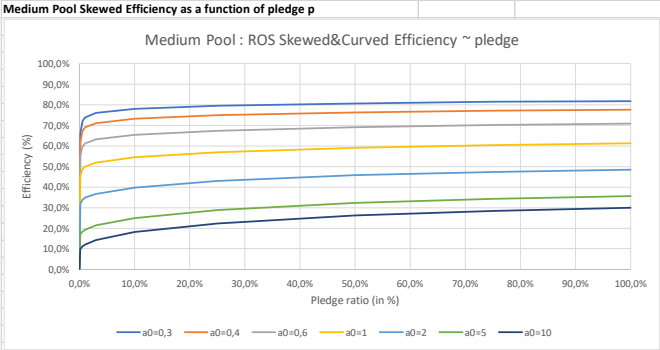
	Efficiency Table	P									
		0%	0,1%	0,5%	1%	3%	10%	25%	50%	75%	100%
S	0%	0,0%	65,6%	71,4%	73,0%	74,6%	75,6%	76,1%	76,3%	76,4%	76,5%
	1%	0,0%	65,6%	71,4%	73,0%	74,6%	75,7%	76,2%	76,4%	76,5%	76,6%
	2%	0,0%	65,6%	71,5%	73,0%	74,6%	75,7%	76,2%	76,5%	76,6%	76,6%
	3%	0,0%	65,7%	71,5%	73,0%	74,7%	75,8%	76,3%	76,5%	76,6%	76,7%
	5%	0,0%	65,7%	71,5%	73,0%	74,7%	75,8%	76,4%	76,6%	76,7%	76,8%
	10%	0,0%	65,7%	71,6%	73,1%	74,9%	76,1%	76,7%	77,0%	77,1%	77,1%
	25%	0,0%	65,8%	71,8%	73,4%	75,3%	76,8%	77,6%	78,2%	78,4%	78,5%
	50%	0,0%	66,0%	72,1%	73,9%	76,0%	78,0%	79,6%	80,8%	81,4%	81,9%
	75%	0,0%	66,2%	72,5%	74,4%	76,8%	79,5%	81,9%	84,1%	85,6%	86,8%
	100%	0,0%	66,4%	72,9%	74,9%	77,7%	81,1%	84,5%	88,2%	90,9%	93,1%

Small Pool Skewed Efficiency as a function of pledge p



Small Pool	a0							
skcvE(p,1%)	0,3	0,4	0,6	1	2	5	10
P	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
	0,1%	65,6%	61,6%	54,8%	45,0%	31,0%	16,1%	8,9%
	0,5%	71,4%	66,7%	58,8%	47,6%	32,3%	16,4%	9,1%
	1,0%	73,0%	68,0%	59,9%	48,3%	32,6%	16,5%	9,1%
	3,0%	74,6%	69,4%	61,0%	49,1%	33,0%	16,6%	9,2%
	10,0%	75,7%	70,4%	61,7%	49,5%	33,2%	16,7%	9,2%
	25,0%	76,2%	70,8%	62,0%	49,8%	33,3%	16,8%	9,3%
	50,0%	76,4%	71,0%	62,2%	49,9%	33,4%	16,8%	9,3%
	75,0%	76,5%	71,1%	62,3%	49,9%	33,4%	16,8%	9,3%
	100,0%	76,6%	71,1%	62,3%	49,9%	33,4%	16,8%	9,3%

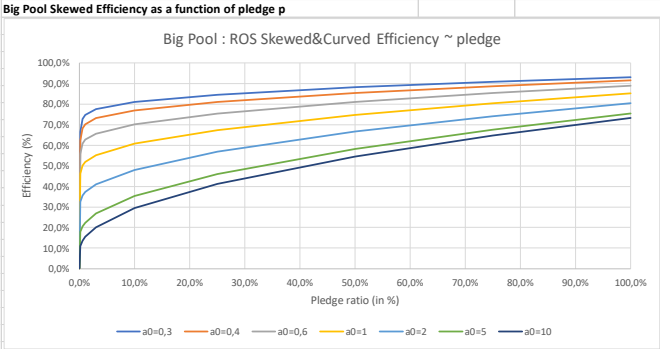
Medium Pool Skewed Efficiency as a function of pledge p



Medium Pool	a0						
skcvE(p,50%)	0,3	0,4	0,6	1	2	5	10
0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,1%	66,0%	62,0%	55,3%	45,6%	31,8%	17,0%	9,9%
0,5%	72,1%	67,5%	59,8%	48,9%	33,9%	18,4%	11,2%
1,0%	73,9%	69,1%	61,2%	50,1%	34,9%	19,3%	12,1%
3,0%	76,0%	71,2%	63,2%	51,9%	36,7%	21,4%	14,3%
10,0%	78,0%	73,3%	65,4%	54,5%	39,7%	24,9%	18,1%
25,0%	79,6%	75,0%	67,5%	57,0%	42,9%	28,8%	22,3%
50,0%	80,8%	76,4%	69,2%	59,2%	45,7%	32,3%	26,1%
75,0%	81,4%	77,2%	70,2%	60,5%	47,4%	34,4%	28,4%
100,0%	81,9%	77,7%	70,9%	61,3%	48,5%	35,7%	29,9%

P

Big Pool Skewed Efficiency as a function of pledge p



Big Pool	a0						
skcvE(p,100%)	0,3	0,4	0,6	1	2	5	10
0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,1%	66,4%	62,5%	55,8%	46,2%	32,6%	17,9%	10,9%
0,5%	72,9%	68,3%	60,9%	50,2%	35,6%	20,6%	13,6%
1,0%	74,9%	70,3%	62,7%	51,9%	37,3%	22,4%	15,5%
3,0%	77,7%	73,2%	65,7%	55,2%	41,1%	26,8%	20,2%
10,0%	81,1%	76,9%	70,2%	60,8%	48,0%	35,3%	29,4%
25,0%	84,5%	81,1%	75,4%	67,5%	56,8%	46,1%	41,2%
50,0%	88,2%	85,5%	81,1%	74,9%	66,6%	58,3%	54,5%
75,0%	90,9%	88,8%	85,3%	80,5%	74,1%	67,7%	64,8%
100,0%	93,1%	91,5%	88,9%	85,3%	80,5%	75,6%	73,4%

P