Let us proceed with the pool hopping as follows. In the experiment, there will be only two pools that form the whole BitCoin network: pool_1 with power p^* and pool_2 with power $1 - p^*$. Try small values for pool_1, e.g. $p^* = 0.2$, for instance (therefore, power of pool_2 will be 0.8). Assume, that there is only one attacking miner with power $p < p^*$ (total power of pool_1 includes the power of that miner), and, it is very important that the miner starts in pool_1 at the beginning of every mining round. The miner spends time $\alpha\Delta$, in pool_1 and time $(1 - \alpha)\Delta$ in pool_2, where $\alpha = \frac{1+p^*}{2}$, time interval (or number of cycles) Δ should be quite small, but if you use cycles, $\alpha\Delta$ and $(1 - \alpha)\Delta$ should be quite small integer numbers, like 3 and 2, for example. After mining in pool_2 miner returns to pool_1 and spends there the same time as he did previously, and, so on until the full solution is found. That should work and the miner should get reward greater than p (in long run).

While shifting periodically between pool_1 and pool_2, in the long run miner allocates power αp and $(1 - \alpha)p$ for pool_1 and pool_2, respectively. This is true for any sufficiently long time. Therefore, miner should receive extra profit not only under Proportional reward scheme, but under Pay Per Last N shares too. You can try this as well, or, we can discuss it next time.