西安交通大學
XI'AN JIAOTONG UNIVERSITY

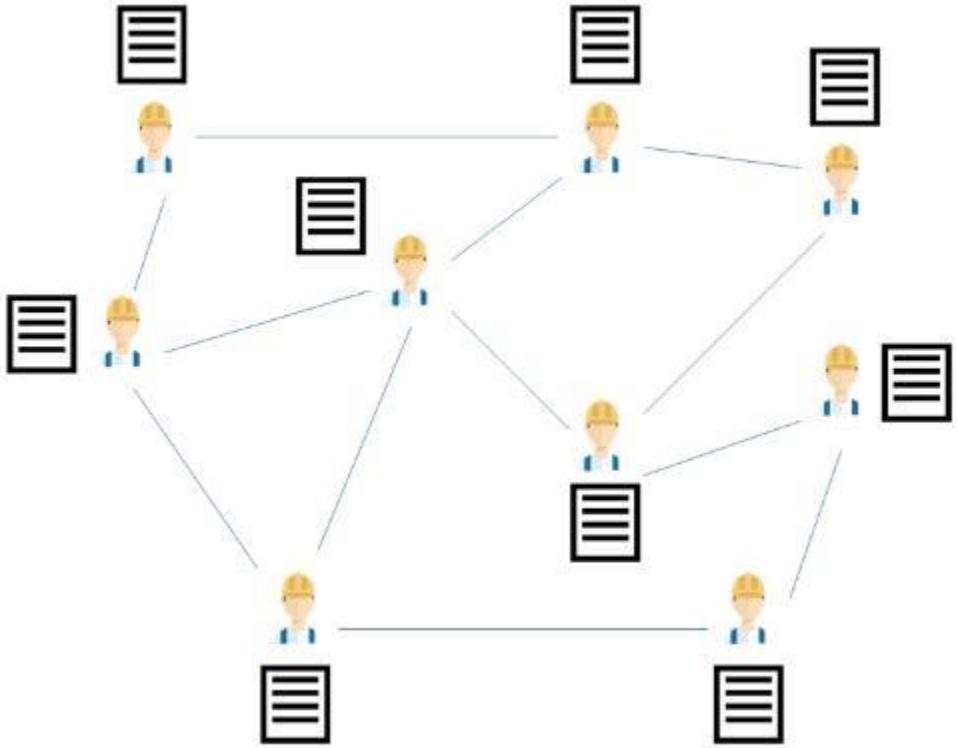
Financial Cryptography and Data Security 2014

Majority is not Enough: Bitcoin Mining is Vulnerable

2020.10.30

Background

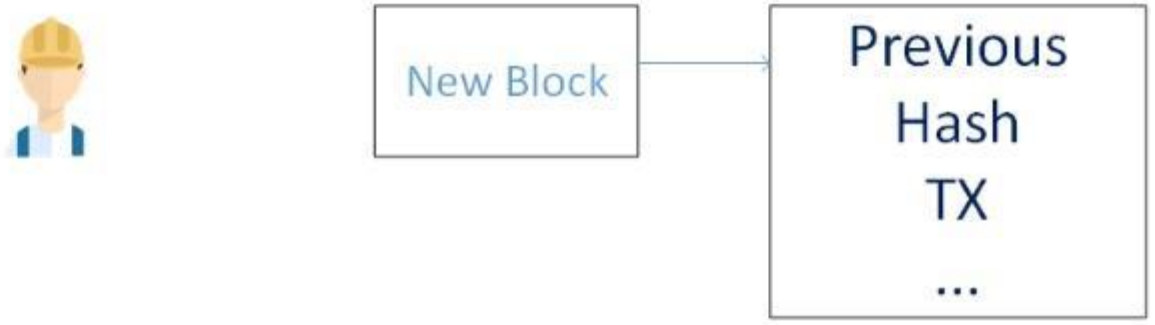
/01



- P2P Network
- Each client commands (multiple) accounts (addresses)
- Recording TXs in the Blockchain(ownership of BTC)

Background

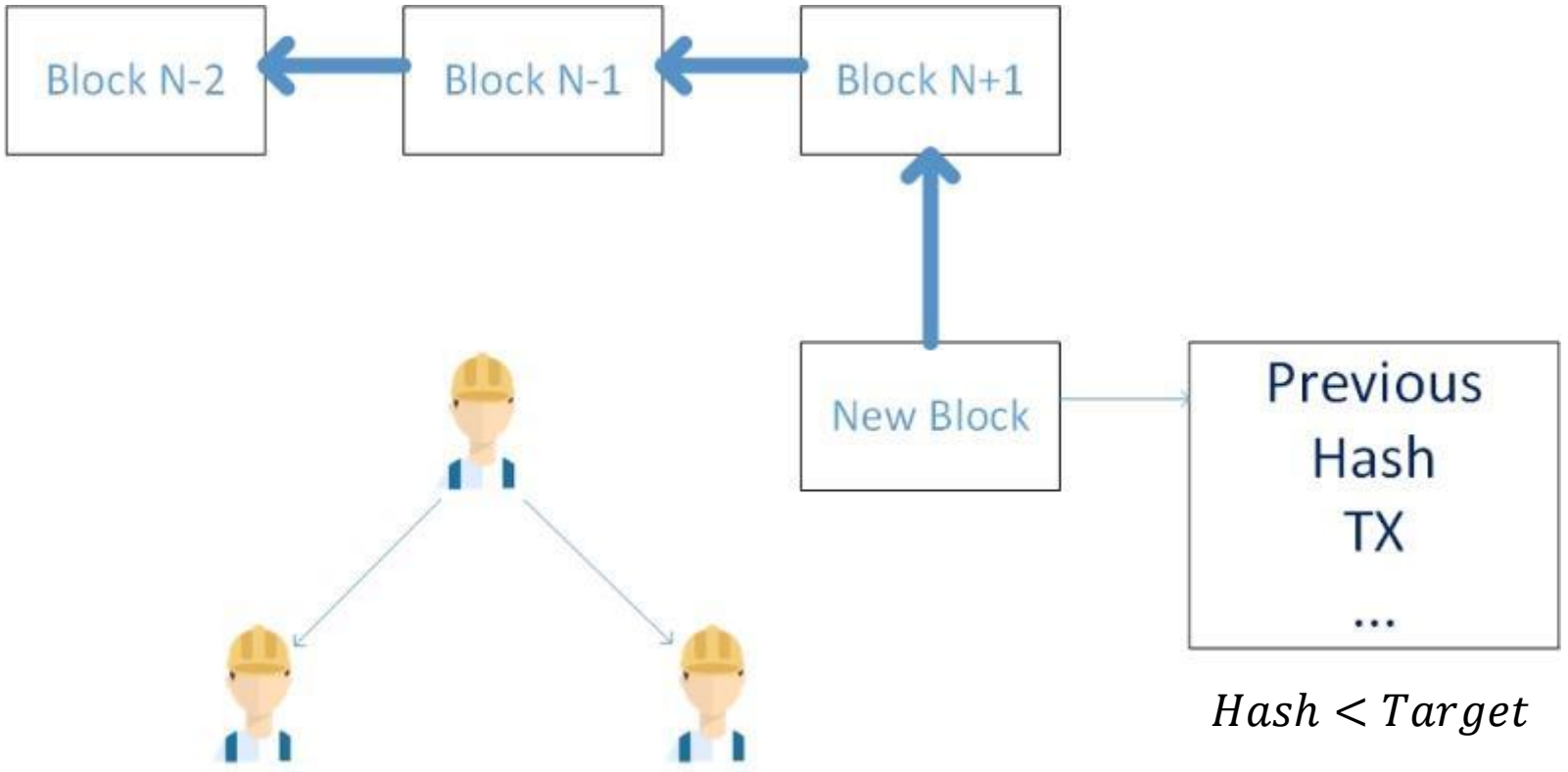
/01



$Hash < Target$

Background

/01

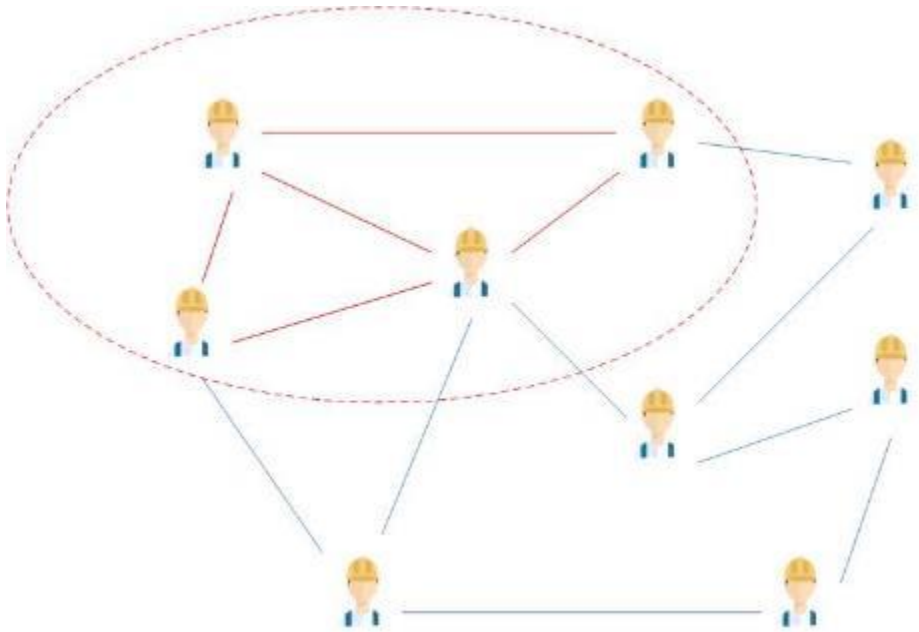


$Hash < Target$

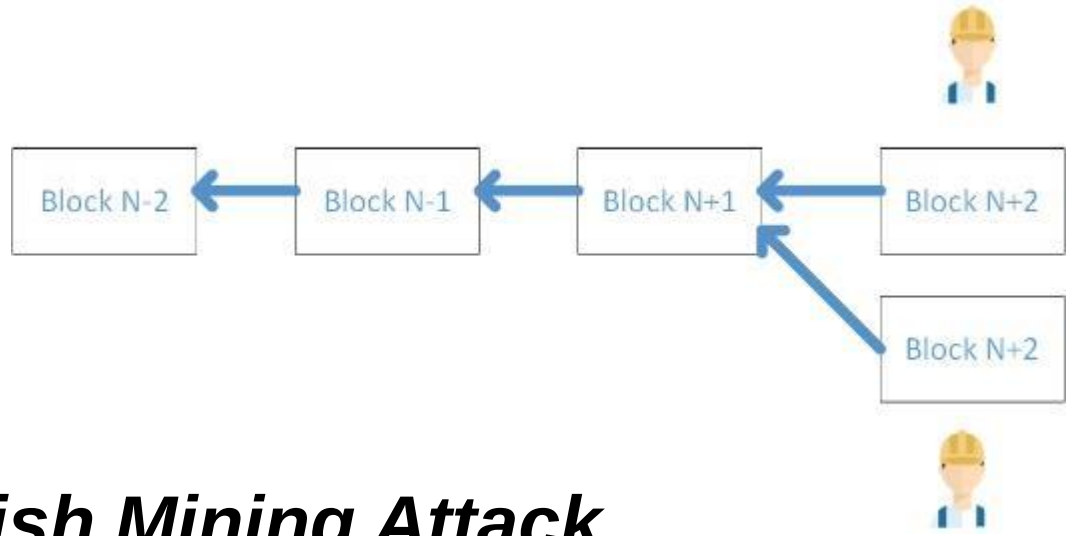
Background

/01

Miners tend to form Mining Pool



Branch may happen sometimes

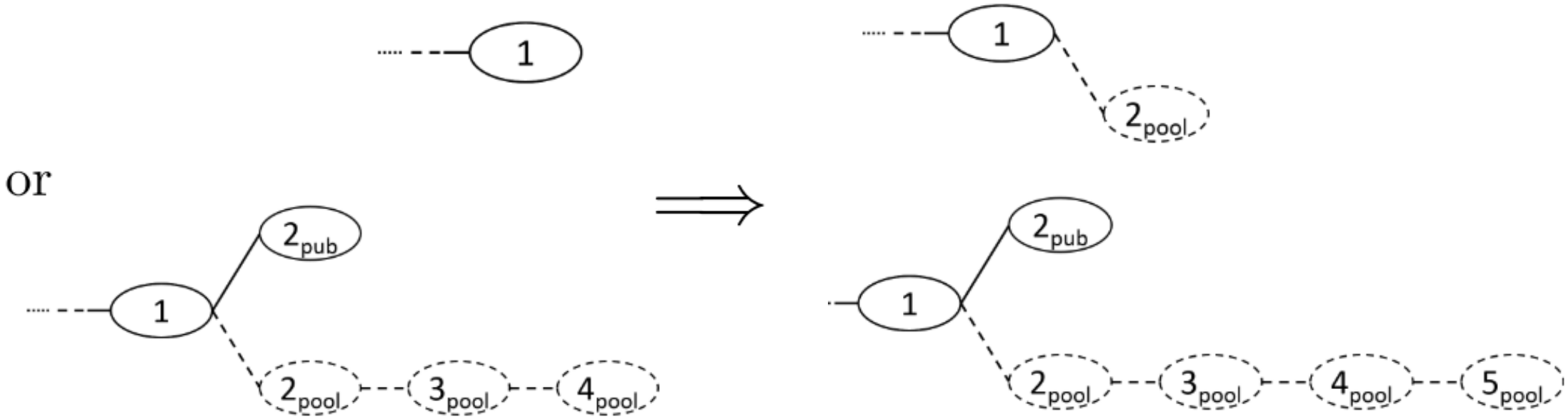


Selfish Mining Attack

Strategy

/02

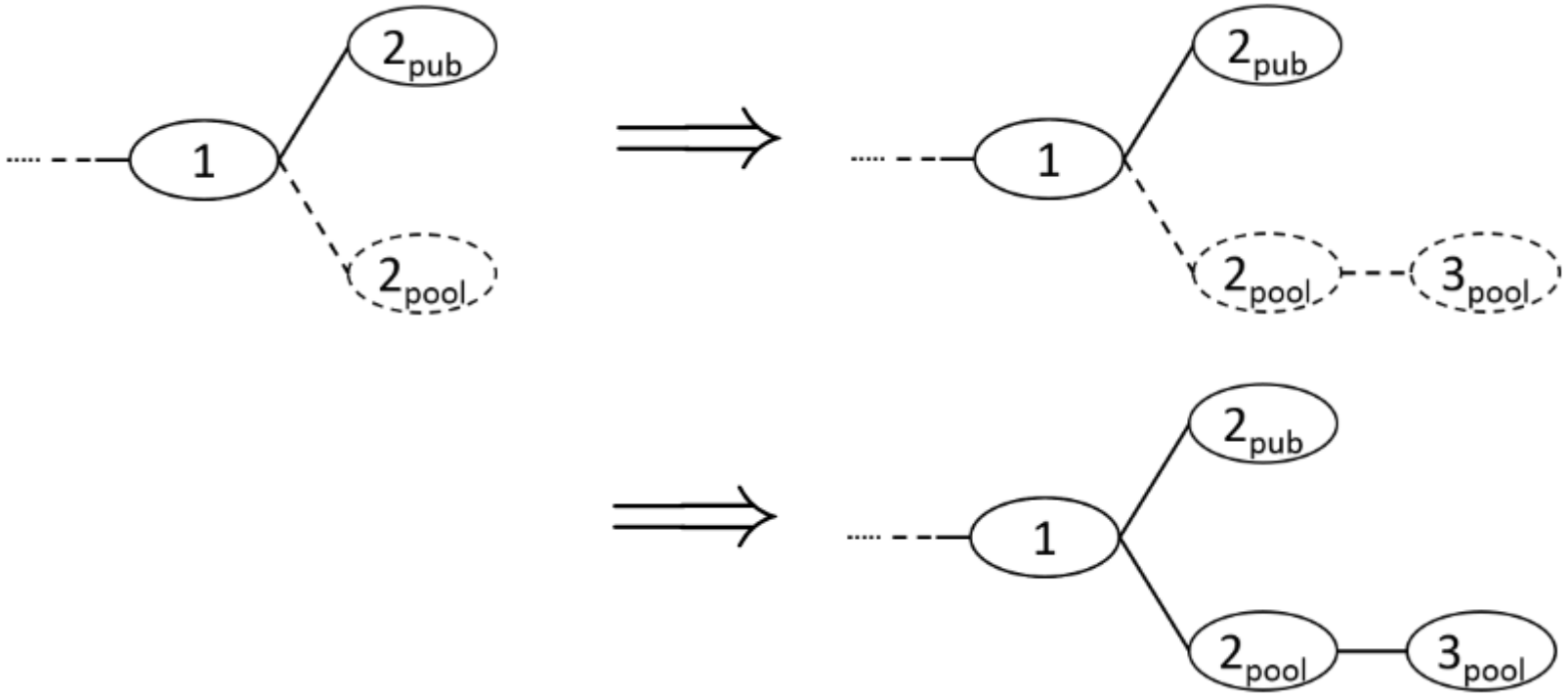
(a)Any state but two branches of length 1, pools finds a block.



Strategy

/02

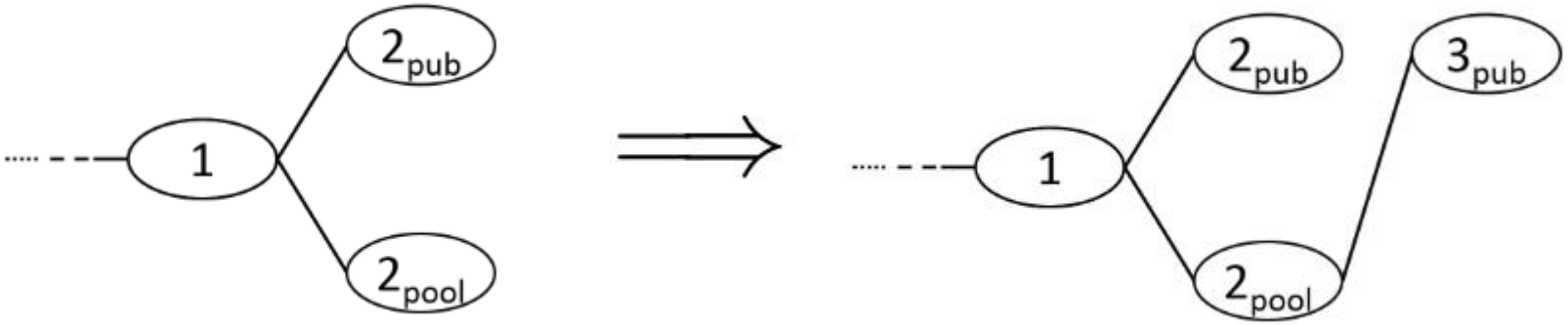
(b)Was two branches of length 1, pools finds a block.



Strategy

/02

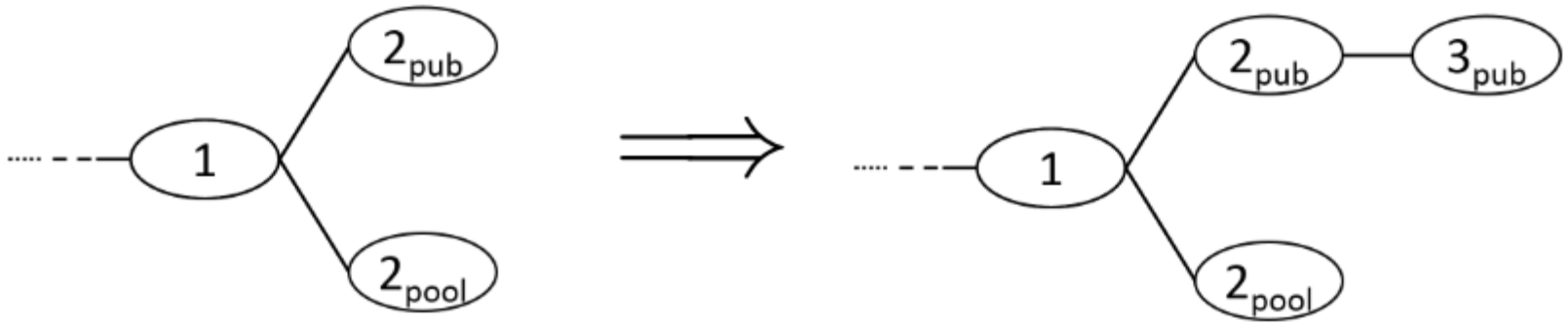
(c)Was two branches of length 1, others find a block after pool head.



Strategy

/02

(d)Was two branches of length 1, others find a block after others' head.

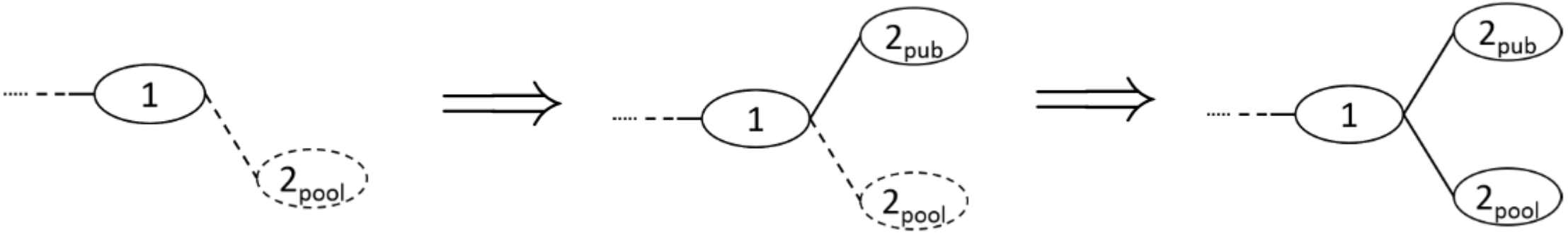


Strategy

/02

(e) No private branch, others find a block.

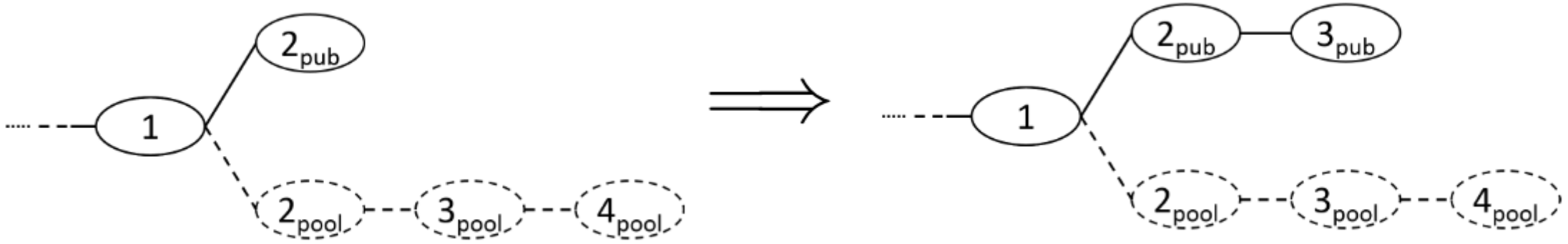
(f) Lead was 1, others find a block.



Strategy

/02

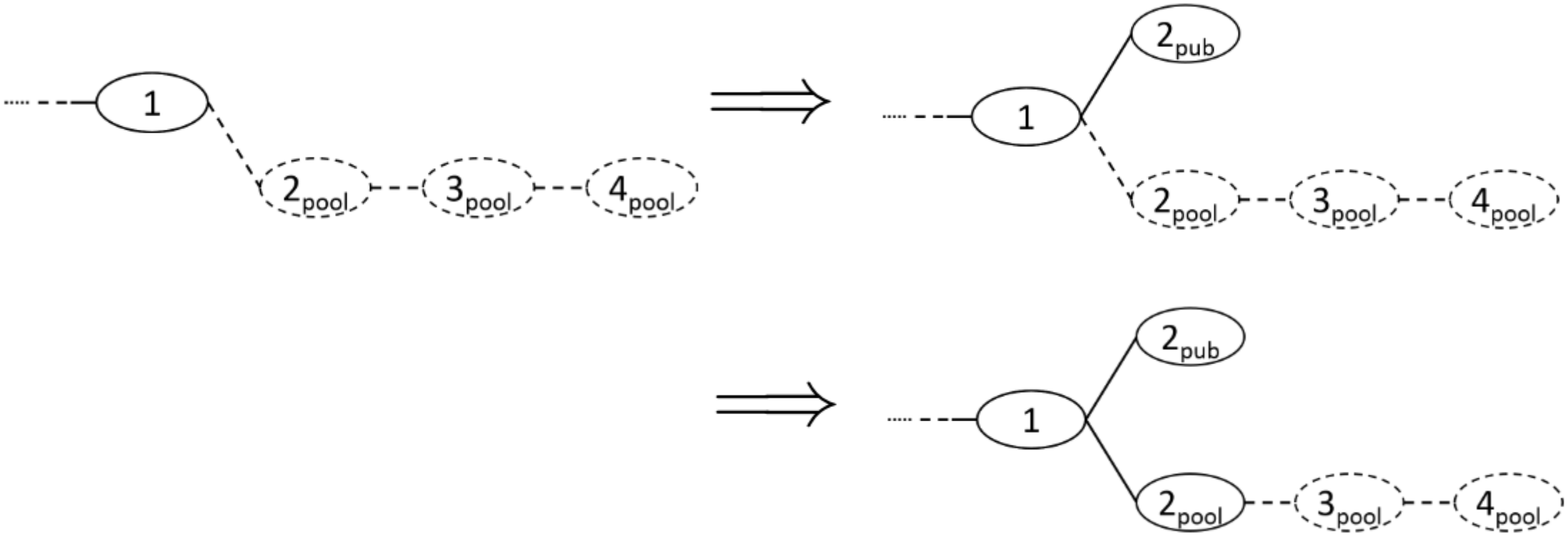
(g) Lead was 2, others find a block.



Strategy

/02

(h) Lead was more than 2, others win.



Strategy

/02

Algorithm 1: Selfish-Mine

```
1  on Init
2      public chain ← publicly known blocks
3      private chain ← publicly known blocks
4      privateBranchLen ← 0
5      Mine at the head of the private chain.

6  on My pool found a block
7      Δ_prev ← length(private chain) − length(public chain)
8      append new block to private chain
9      privateBranchLen ← privateBranchLen + 1
10     if Δ_prev = 0 and privateBranchLen = 2 then        (Was tie with branch of 1)
11         publish all of the private chain               (Pool wins due to the lead of 1)
12         privateBranchLen ← 0
13     Mine at the new head of the private chain.

14 on Others found a block
15     Δ_prev ← length(private chain) − length(public chain)
16     append new block to public chain
17     if Δ_prev = 0 then
18         private chain ← public chain                   (they win)
19         privateBranchLen ← 0
20     else if Δ_prev = 1 then
21         publish last block of the private chain        (Now same length. Try our luck)
22     else if Δ_prev = 2 then
23         publish all of the private chain               (Pool wins due to the lead of 1)
24         privateBranchLen ← 0
25     else                                               (Δ_prev > 2)
26         publish first unpublished block in private block.
27     Mine at the head of the private chain.
```

Strategy

/02

Algorithm 1: Selfish-Mine

```
1  on Init
2      public chain ← publicly known blocks
3      private chain ← publicly known blocks
4      privateBranchLen ← 0
5      Mine at the head of the private chain.

6  on My pool found a block
7      Δ_prev ← length(private chain) − length(public chain)
8      append new block to private chain
9      privateBranchLen ← privateBranchLen + 1
10     if Δ_prev = 0 and privateBranchLen = 2 then        (Was tie with branch of 1)
11         publish all of the private chain               (Pool wins due to the lead of 1)
12         privateBranchLen ← 0
13     Mine at the new head of the private chain.

14 on Others found a block
15     Δ_prev ← length(private chain) − length(public chain)
16     append new block to public chain
17     if Δ_prev = 0 then
18         private chain ← public chain                   (they win)
19         privateBranchLen ← 0
20     else if Δ_prev = 1 then
21         publish last block of the private chain        (Now same length. Try our luck)
22     else if Δ_prev = 2 then
23         publish all of the private chain               (Pool wins due to the lead of 1)
24         privateBranchLen ← 0
25     else                                               (Δ_prev > 2)
26         publish first unpublished block in private block.
27     Mine at the head of the private chain.
```

Strategy

/02

Algorithm 1: Selfish-Mine

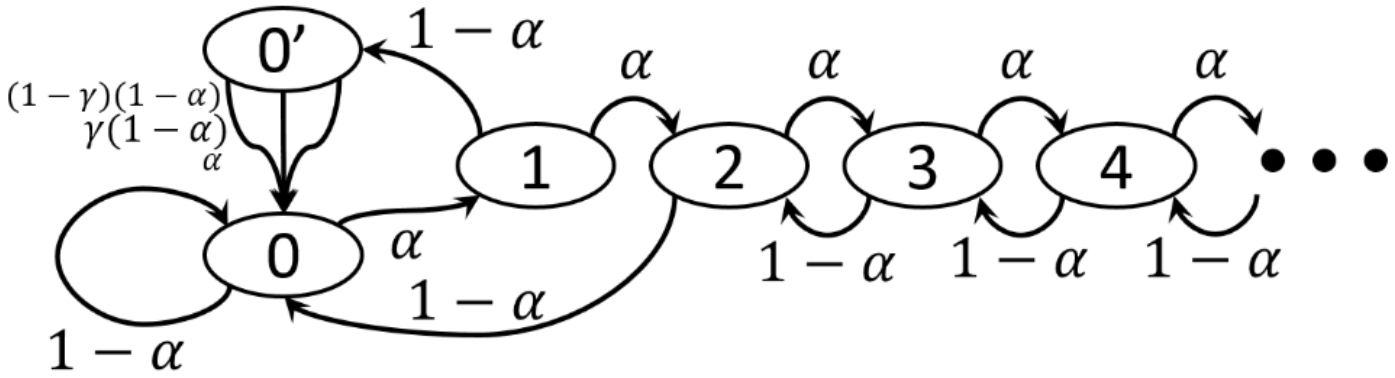
```
1  on Init
2      public chain ← publicly known blocks
3      private chain ← publicly known blocks
4      privateBranchLen ← 0
5      Mine at the head of the private chain.

6  on My pool found a block
7      Δ_prev ← length(private chain) − length(public chain)
8      append new block to private chain
9      privateBranchLen ← privateBranchLen + 1
10     if Δ_prev = 0 and privateBranchLen = 2 then        (Was tie with branch of 1)
11         publish all of the private chain               (Pool wins due to the lead of 1)
12         privateBranchLen ← 0
13     Mine at the new head of the private chain.

14 on Others found a block
15     Δ_prev ← length(private chain) − length(public chain)
16     append new block to public chain
17     if Δ_prev = 0 then
18         private chain ← public chain                   (they win)
19         privateBranchLen ← 0
20     else if Δ_prev = 1 then
21         publish last block of the private chain        (Now same length. Try our luck)
22     else if Δ_prev = 2 then
23         publish all of the private chain               (Pool wins due to the lead of 1)
24         privateBranchLen ← 0
25     else                                               (Δ_prev > 2)
26         publish first unpublished block in private block.
27     Mine at the head of the private chain.
```

Analysis

/03

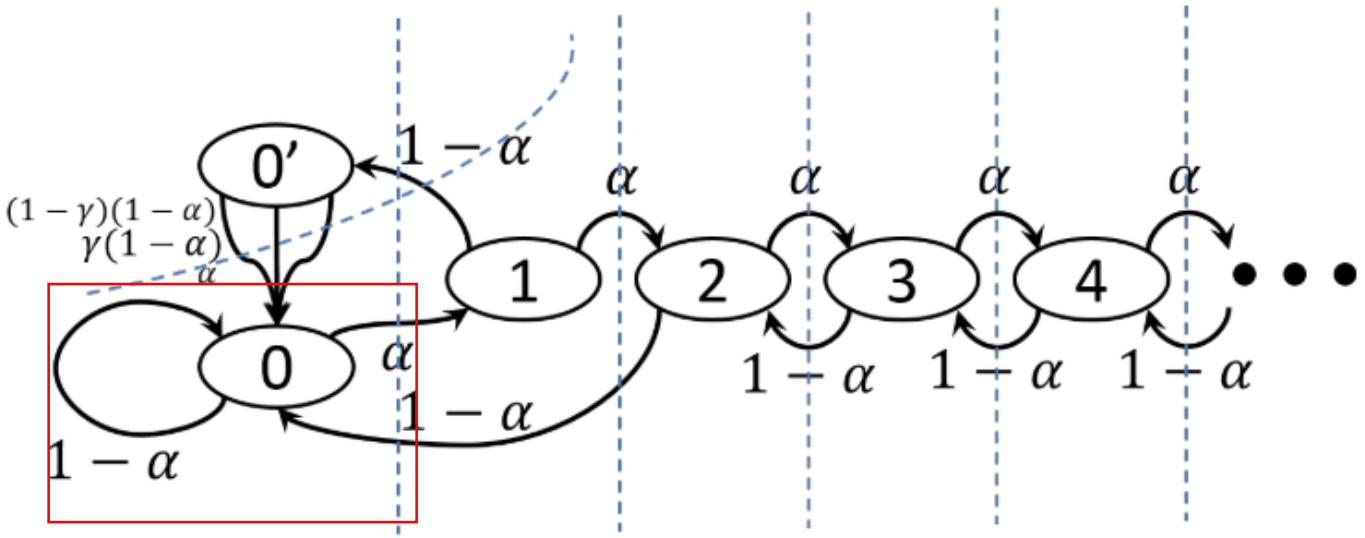


State 0 : no branch
State 0': two branches of length 1
State n : private chain lead n blocks

α : selfish pool mining power
γ : ratio of honest miner mine on private chain

Analysis

/03



$$\begin{cases} \alpha p_0 = (1-\alpha)p_1 + (1-\alpha)p_2 \\ p_{0'} = (1-\alpha)p_1 \\ \alpha p_1 = (1-\alpha)p_2 \\ \forall k \geq 2 : \alpha p_k = (1-\alpha)p_{k+1} \\ \sum_{k=0}^{\infty} p_k + p_{0'} = 1 \end{cases}$$

$$p_0 = p_{0'} + (1-\alpha)p_2 = (1-\alpha)p_1 + (1-\alpha)p_2$$

$$p_0 = \frac{\alpha - 2\alpha^2}{\alpha(2\alpha^3 - 4\alpha^2 + 1)}$$

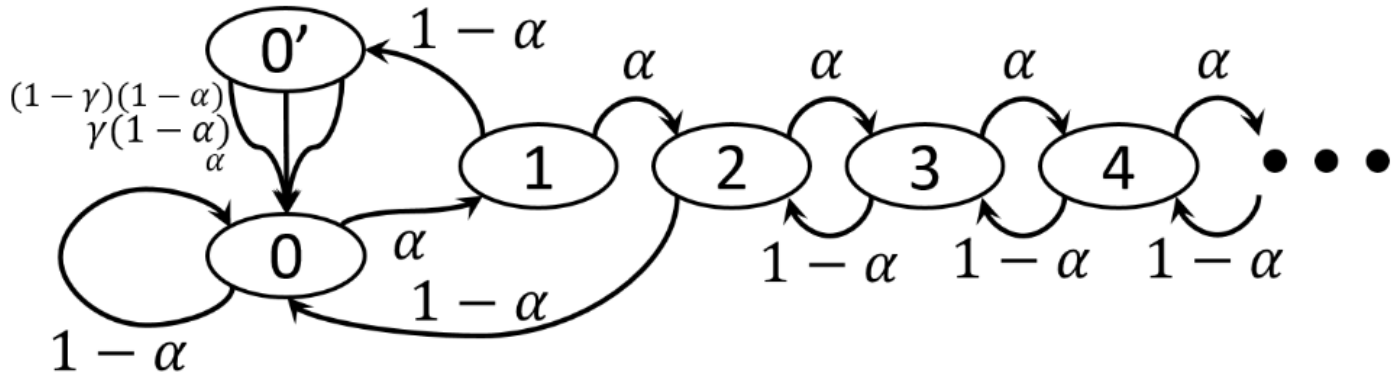
$$p_{0'} = \frac{(1-\alpha)(\alpha - 2\alpha^2)}{1 - 4\alpha^2 + 2\alpha^3}$$

$$p_1 = \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1}$$

$$\forall k \geq 2 : p_k = \left(\frac{\alpha}{1-\alpha}\right)^{k-1} \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1}$$

Analysis

/03



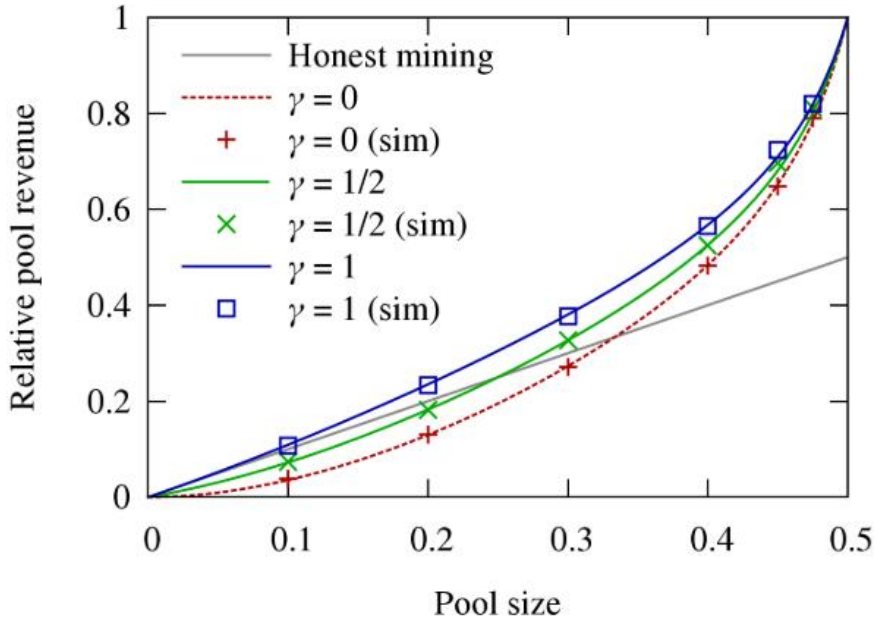
$$r_{\text{others}} = \overbrace{p_{0'} \cdot \gamma(1-\alpha) \cdot 1}^{\text{Case (c)}} + \overbrace{p_{0'} \cdot (1-\gamma)(1-\alpha) \cdot 2}^{\text{Case (d)}} + \overbrace{p_0 \cdot (1-\alpha) \cdot 1}^{\text{Case (e)}}$$

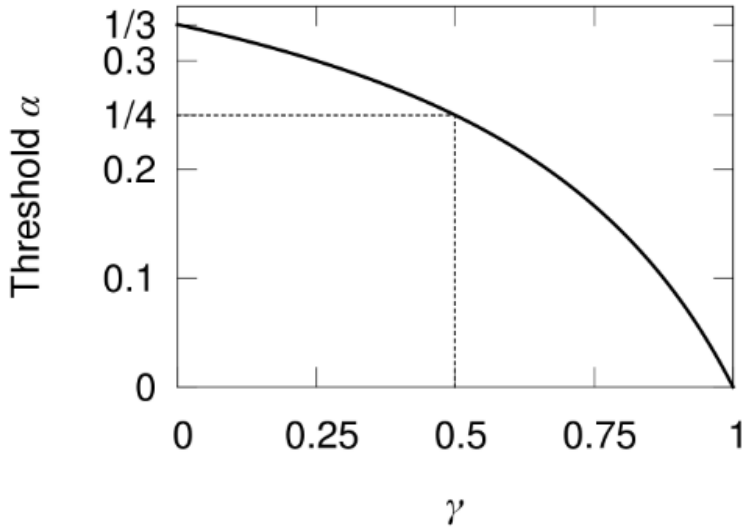
$$r_{\text{pool}} = \overbrace{p_{0'} \cdot \alpha \cdot 2}^{\text{Case (b)}} + \overbrace{p_{0'} \cdot \gamma(1-\alpha) \cdot 1}^{\text{Case (c)}} + \overbrace{p_2 \cdot (1-\alpha) \cdot 2}^{\text{Case (g)}} + \overbrace{P[i>2](1-\alpha) \cdot 1}^{\text{Case (h)}}$$

$$R_{\text{pool}} = \frac{r_{\text{pool}}}{r_{\text{pool}} + r_{\text{others}}} = \cdots = \frac{\alpha(1-\alpha)^2(4\alpha + \gamma(1-2\alpha)) - \alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$

Simulate & Experiment

/03





$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2}$$