Yanni Yang,
Bowen Liu,
Aravindh Shanmuganathan,
Bryan Xie

Blockchain

1. Introduction

The Nash equilibrium has played a big role in making Bitcoin and by extension Blockchain unique considering that it incorporates a unique action profile in the Bitcoin transaction. Our report considers the game-theory problem with fixed cost of energy and return bonus in two situations: continuous decision variables x and discrete decision variables x. For continuous decision variable situation, we formulate and solve our mathematical models one and two through using the Karush-Kuhn-Tucker method and simplex method respectively. We then utilize the enumeration method to identify the discrete decision variables problem, listing all the possible strategies for a specific case. Moreover, from the sensitivity-analysis perspective, we want to know how changes in our linear programming parameters affect the optimal solution. To test these models, we split our models into two cases and figure out which situation still lets us maintain the optimal values. In the end, we briefly summarize all the methods and cases we use in this report.

2. Our Models

2.1. Continuous Model:

We consider a n-player contest with one prize. In this setting, if miner i has xi resources devoted to mining operations, its winning probability is $p = \frac{x_i}{\sum_j x_j}$ .In determining the resources to investing in mining operations, the miner needs to purchase computational resources and pay for energy bills. Given fees from winning the mining operation is c and energy costs for miner i is given by ei. So, our gain c is described as Gain:

$$Gain = \begin{cases} c & with \ \ p = \dfrac{x_i}{\sum_j x_j} \\ 0 & with \ \ q = 1 - p \end{cases} \tag{1}$$

Contingent upon winning or losing, the payoff for player i is a function of prizes, own resources, and resources of the rival:

$$Payoff \begin{cases} -ei & with\ p = 1 \\ 0 \end{cases} \tag{2}$$

For a given resources pair $\left(x_i, \sum_j x_j\right)$, the expected payoff for player i in contest is:

$$E\left(\pi_i\left(x_i, \sum_j x_j\right)\right) = \left(\frac{c}{\sum_j x_j} - e_i\right) x_i \tag{3}$$

Player i's best response is dervied by maximizing $E\left(\pi_i\left(x_i, \sum_j x_j\right)\right)$ with respect to xi. Differentiating Equation (3) with respect to xi yields the following first order condition:

$$\frac{dE\left(\pi_i\left(x_i, \sum_j x_j\right)\right)}{dxi} = \frac{\left(\left(c \cdot \left(\sum_j x_j\right)\right) - c \cdot xi\right)}{\left(\sum_j x_j\right)^2} - ei$$

$$= \frac{c}{\sum_j x_j} - \frac{c \cdot xi}{\left(\sum_j x_j\right)^2} - ei \tag{4}$$

From Equation (4), we know that our gain c is a constant number and xj represents resources owned by player j. So, if the resources pool(xj) increases, the expectation of the miner's gain will decrease. This phenomenon proves the key to this game， the more resources are applied, the harder it is to win, which means the relationship between xj and c is inverse. Indeed, our premise is that the payoff function for player i and constraints are all convex. After sharing our ideas with the TA, we decide to use the Karush-Kuhn-Tucker conditions to solve the conditional convex optimization problem in our model 1.

Model 1:

Karush-Kuhn-Tucker conditions are first-order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Consider our question as the following nonlinear maximization problem:

$$Min\ -E(x_i) = -\frac{cxi}{\sum_j x_j} + e_i x_i$$

$$s.t.$$

$$g: \begin{cases} x1 \le y1 \\ x2 \le y2 \\ \vdots \quad \vdots \quad \vdots \\ xi \le yi \\ \vdots \quad \vdots \quad \vdots \\ xn \le yn \end{cases} \tag{5}$$

Where Equation (5) is the optimal function, g is the inequality constraint set. The numbers of inequality constraints are denoted by n. Suppose that the objective Equation (5) : $R^n -> R$ and constraint functions gi: $R^n -> R$ are continuously differentiable at a point x*. If x* is a local optimum and the optimization problem satisfies some regularity conditions, then there exist constants $\mu i(i = 1, ...n)$ and λ called KKT multipliers, such that, for minimizing L(x):

$$L(x_i, \lambda) = -E(x_i) + \sum_k^m \lambda_k \cdot g_k(x_i) \tag{6}$$

We then depend on Equation (6) function to derive xi:

$$0 = \nabla x\, L(x_i, \lambda) = \nabla\left(-E(x_i)\right) + \sum_k^m \lambda_k \cdot \nabla g_k(x) \tag{7}$$

Model 2:

As mentioned above, expectation payoff for player i is:

$$E = \left(\frac{c}{\sum_j x_j} - e_i\right) x_i \tag{8}$$

There are so many players in the blockchain problem, that is, j is a huge number, so that there is almost no effect of x_i to $\sum_j x_j$. Based on this fact, we simplify our model by substituting $\sum_j x_j$ with $\sum x_{-i}$. x_{-i} means the resources devoted by players other than player i. The simplified expection formula：

$$E = \left(\frac{c}{\sum x_{-i}} - e_i \right) x_i \tag{9}$$

The total resources now has nothing to do with player i, so, it is a constant when we consider xi.

$$\sum x_{-i} = d \tag{10}$$

So, the problem changes into a linear programing problem, where the object is to maximize E, and the constraints are the limited resources for every player. We learned that if other players have their secured strategies, we need to find our own strategy to get the optimal value. The number of computing resources is also a constant number. So, d, e, c and yi are all fixed parameters, and only xi changes every time.

Our model is shown below mathmaticaly,

$$\begin{aligned} &Max \left(\frac{c}{d} - e \right) x_i \\ &s.t. \quad x_i \le y_i, \ for \ i = 1, \ldots, n \end{aligned} \tag{11}$$

Linear programing standard form is:

$$\begin{aligned} Max \quad & z = C^T X \\ s.t. \quad & AX \leqslant b \\ & X \geqslant 0 \end{aligned}$$

In this model,

$$X = [x_i], A = [1], b = [y_i], C = \left[\frac{c}{d} + e_i \right] \tag{12}$$

It is obvious that, when the coefficient of xi is positive, we get optimal value of $\left(\frac{c}{d} - e \right) y_i$ at xi=yi. When the coefficient of xi is negative, we get optimal value of 0 at xi=0.

Case for model 2:

We use Matlab to implement our Equation (11). First, we assume we have 100 players. and then generate each player's resources yi randomly. Then, we substitute our mining operation fee c and energy costs for miner 3 given by ei to two constant numbers: 5*10^4 and 3. We define the variable d in Equation (11), which is the sum of xj. Here are our variables setting table and matlab code:

```
% ex: miner operation
clear
clc
n = 1000;
i = 3;
c1 = 5.*10^4;
e = 3;
d = 2000;
A = eye(1);
b = [];
sum = 0;
for j = 1:1:n
   b(j,1)= 50.*rand(1);
   sum = b(j,1)+sum;
   j=j+1;
end
b = b(i,1);
C = -c1/d + e;
linprog(C,A,b)

ans = 6.3493
```

Sensitivity:

Because variables c, d and e are all fixed, we select different player every time to identify the sensitive issue.

$$Sensitive = \begin{cases} \frac{c}{d} - e > 0 & i \\ \frac{c}{d} - e < 0 & ii \end{cases} \tag{13}$$

Indeed, if variables satisfy equation i, then the problem is still optimal.

2.2 Discrete Model:

We try to formulate this problem as a discrete model, where each Miner has to choose from a set of pure strategies. Each miner can either be a mining pool or an individual miner. If we consider two miners A and B, such that, we have the constraint $yA = 1$ and $yB = 1$so the number of pure strategies for A and B would be:

$$xA = \{0, 1\}, xB = \{0, 1\}$$

Since this is a two player non-zero-sum game, we would need two matrices of size n*m where n = No. of strategies of A and m = No. of strategies of B.

Here we are assuming a reward of c = $5

The electricity costs $ej = xj$ (i.e. 1 processor consumes $1 of electricity).

The payout for a miner would be:

$$P = \begin{cases} c - ei & if\ the\ miner\ wins \\ -ei & if\ the\ miner\ loses \end{cases} \tag{14}$$

The payout matrix would be:

		B			
A		**0**		**1**	
		A	B	A	B
	0	0	0	0	4
	1	4	0	**1.5**	**1.5**

We find that the Nash equilibrium is at A=1 and B=1. The two strategies that we have here are to either mine or not mine. When both players decide to mine, each player wins the reward 50% of the time given an n number of rounds. Thus, in a setting where all miners have the same computational power, it is always better to mine.

When we increase the upper limit to, $yA = 5$ and $yB = 5$, $c = \$\,5$ and $ej = yj$,

$$xA = \{0,\ 1,\ 2,\ 3,\ 4,\ 5\}$$
$$xB = \{0,\ 1,\ 2,\ 3,\ 4,\ 5\}$$

If we choose $yj > c$, we will always get a negative payout given our electricity costs, so we ignore cases where $yj > c$ in this setting. The payout matrix is:

		B											
		0		1		2		3		4		5	
		A	B	A	B	A	B	A	B	A	B	A	B
A	0	0	0	0	4	0	3	0	2	0	1	0	0
	1	4	0	1.5	1.5	-1	3	-1	2	-1	1	-1	0
	2	3	0	3	-1	0.5	0.5	-2	2	-2	1	-2	0
	3	2	0	2	-1	2	-2	-0.5	-0.5	-3	1	-3	0
	4	1	0	1	-1	1	-2	1	-3	-1.5	-1.5	-4	0
	5	0	0	0	-1	0	-2	0	-3	0	-4	-2.5	-2.5

Green- Best Strategy for A given B's Strategy; Yellow- Best Strategy for B given A's Strategy

There is no Nash equilibrium for this setting. Thus, in a discrete model formulation there is no pure strategy for the miners that will give the maximum payout. We must use a mixed strategy for each miner.

3. Conclusion

Therefore, it can be asserted that when using continuous model, player should put resources in the game when game bonus is in large amount, otherwise, do not join the game is wise so that player can avoid paying for computation resources. On the other hand, when using a discrete model, where each player has a fixed number of pure strategies there was no Nash equilibrium point. This indicates that in order for each miner to obtain the maximum payout we must use a mixed strategy. It is for this

reason that the Nash equilibrium has played a big role in making Bitcoin and by extension Blockchain unique considering that it incorporates a unique action profile in the Bitcoin transaction that cannot be ruled out by the common knowledge of rationality-maximization of he expected payoff.

4. Potential Improvement

Due to the limited time, we only completed the solving processes of our two models. For model 1, we believe that we may implement the gradient descent algorithm to get the optimal solution in future.

5. References

[1] Altman, E., Reiffers, F. A., Menasché, D. S., Matar, M., Dhamal, S., & Touati, C. (2018, December). Mining competition in a multi-cryptocurrency ecosystem at the network edge: a congestion game approach. In 1st Symposium on Cryptocurrency Analysis (SOCCA 2018).

[2] Cheng, Y., Du, D., & Han, Q. (2018). A Hashing Power Allocation Game in Cryptocurrencies. In International Symposium on Algorithmic Game Theory (pp. 226-238). Springer, Cham.

[3] Chiu, J., & Koeppl, T. V. (2017). The economics of cryptocurrencies–bitcoin and beyond.

[4] Ma, J., Gans, J. S., & Tourky, R. (2018). Market structure in bitcoin mining (No.w24242). National Bureau of Economic Research.

[5] Morisse, M. (2015). Cryptocurrencies and bitcoin: Charting the research landscape.