Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 340 (2018) 151–167

www.elsevier.com/locate/entcs

The Impact of Profit Uncertainty on Miner Decisions in Blockchain Systems

Maher Alharby[1] and Aad van Moorsel[2]

School of Computing
Newcastle University
Newcastle Upon Tyne, UK

Abstract

In blockchain systems, miners execute transactions in blocks. Since each block has a limit on the number of transactions, miners usually prioritise transactions by selecting the most profitable ones. However, miners are uncertain about the exact income and the exact cost of executing transactions. Thus, they are not able to make informed decisions of which transactions to select and execute in order to maximise their profits. The main aim of this paper is to investigate the uncertainty that miners perceive about the income and the cost of executing transactions and the impact of this uncertainty on the profit miners can gain. To achieve this aim, we design a simulation model, collect the input data for the model, implement the model and run an experiment on the model. Our simulation results show that the uncertainty miners perceive about the cost of executing transactions has a significant impact on the block profit. On the contrary, our results do not show a significant impact of the income uncertainty on the block profit.

Keywords: Blockchain, Smart contracts, Uncertainty, Profit, Simulation

1 Introduction

Blockchain, which is the underlying technology of cryptocurrencies, has gained lots of attention recently. In blockchain systems such as Ethereum, users send transactions to the blockchain network to deploy a new smart contract or to invoke an existing one. Other users (usually called miners) then execute these transactions in blocks. Each block has a limit on how many transactions it can have, and thus, miners usually prioritise transactions by selecting the most profitable ones [5]. However, the only information available to the miners before executing a transaction is the maximum income they can get from that transaction. There is uncertainty about the income as well as the cost of executing a transaction. Thus, miners are not able to make informed decisions of which transactions to select and execute in order to maximise their profits.

[1] Email: m.w.r.alharby2@newcastle.ac.uk
[2] Email: aad.vanmoorsel@newcastle.ac.uk

https://doi.org/10.1016/j.entcs.2018.09.011

The main aim of this paper is to investigate the uncertainty that miners perceive about the income and the cost of executing transactions and the impact of this uncertainty on the profits miners can get. To achieve this aim, we first design and implement a simulation model to simulate the decisions that miners take to select a subset of pending transactions in order to execute and include in their forthcoming block. Then, we collect the input data for the model from real historical data and from a controlled experiment. After that, we design and run a simulation experiment to simulate five different scenarios (two baseline and three solution scenarios). The baseline scenarios are to simulate the selection decisions of miners under the real blockchain system condition, where miners are uncertain about income and the cost of executing transactions. The solution scenarios are to simulate the selection decisions of miners when miners are certain about the income and/or the cost of executing each transaction. The objective of this experiment is to investigate the impact of the uncertainty miners perceive about the income and the cost of executing transactions by comparing the block profits that miners can get in the baseline scenarios with that for the solution scenarios. Finally, we present and discuss the experiment's results (the block income, the block cost and the block profit) of each of the five scenarios.

Our simulation results show that the uncertainty miners perceive about the cost of executing transactions has a significant impact on the block profit that miners can get. The certainty about the cost of executing transactions can help miners quadruple their block profit. On the other hand, our results do not show an impact of the income uncertainty on the block profit that miners can get. That means the certainty about the income of executing transactions does not help miners increase their block profit.

The structure of this paper is as follows. Section 2 discusses background information and related work. Section 3 describes the construction of the simulation model. Section 4 discusses the collection of the input data for the model. Section 5 describes the implementation of the simulation model. Section 6 presents the design of the simulation's experiment. Section 7 presents and discusses the experiment's results. Section 8 concludes the paper.

2 Background

2.1 Blockchain Technology Overview

A blockchain is a distributed database that records all transactions that have ever occurred in the blockchain network. This database is replicated and shared among the networks participants. The main feature of blockchain is that it allows non-trusting participants to communicate and send transactions between each other in a secure way without the need of a trusted third party. Blockchain is an ordered list of blocks, where each block is identified by its cryptographic hash. Each block references the block that came before it, resulting in a chain of blocks. Each block consists of a set of transactions [1].

Cryptocurrencies have emerged as the first generation of blockchain technology. Cryptocurrencies are digital currencies that are based on cryptographic techniques

and peer-to-peer network. The earliest and most popular example of cryptocurrencies is Bitcoin. Bitcoin [6] is an electronic payment system that allows two non-trusting parties to transact digital money with each other in a secure manner without going through a middleman (e.g., a bank). Other blockchains such as Ethereum [2,7] have emerged as the second generation of blockchain to allow building complex distributed applications beyond the cryptocurrencies. Smart contracts, which will be discussed in the following section, are considered as the main element of this generation [9].

Transactions in blockchain system can have three different forms. The first one is a transaction to move digital money from one account to another. The second one is a transaction to create and deploy a new smart contract to the blockchain. The last one is a transaction to invoke an existing contract. Each transaction needs to specify the following information:

- The gas limit which is the maximum amount of gas unit the transaction can use [3] .
- The gas price which is the money that the originator of the transaction is willing to pay for each gas unit used.

Each computation or storage operation required by the transaction costs a predefined amount of gas. For instance, a basic addition costs 3 unit of gas [8].

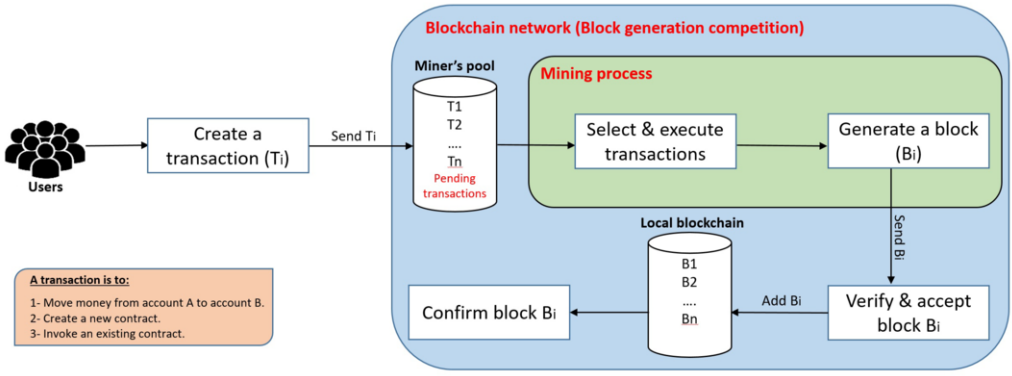

Fig. 1. How the blockchain system works

Figure 1 illustrates how the blockchain system works. In the blockchain system, any user can create a transaction and then propagate it to the blockchain network. This transaction will then be received by other users in the network. Each user has a pool of pending transactions (transactions that have not been executed yet). To create and append a new block to the blockchain system, special users (called miners) select number of transactions from their pools, execute them and then create a new block containing those transactions by solving a mathematical puzzle (called Proof of Work). This is called "mining process". Once a miner has successfully

[3] The reason for specifying the gas limit is to prevent Denial of Service attacks. If a transaction does not have a limit of gas, the transaction might take a long time (or run forever) to be executed leading to a break in the network.

generated a block, she will then propagate it to the network. Other users (including miners) in the network then verify the correctness of the generated block and only build upon it if it was generated correctly. If the majority of the users in the network accepted the block, appended it to their local blockchain copy and built upon it, the block will be confirmed and considered as a winning block and thus it will be added to the blockchain. The miner of that block will then collect a reward for the block [4] as well as the fees associated with its transactions. We call the whole process "The block generation competition" as miners compete against each other in creating a new block of transactions and the fact that there is only one winner of each block who can collect the block reward and the fees associated with its transactions. The focus of this paper and our model will be on the "mining process".

2.2 Smart Contracts Overview

A smart contract is executable code that can be deployed and run on top of the blockchain to facilitate, execute and enforce the terms of an agreement. The main aim of a smart contract is to automatically execute the terms of an agreement once the specified conditions are met. Smart contracts promise low transaction fees compared to traditional systems that require a trusted third party to enforce and execute the terms of an agreement. A smart contract can be thought of as a system that releases digital assets to all or some of the involved parties once arbitrary predefined rules have been met [2]. For instance, Alice sends X currency units to Bob, if she receives Y currency units from Carl [1].

2.3 Related Work

Most of the current research conducted on blockchain systems are about identifying and tackling security, privacy and other issues related to the development of smart contracts [1]. In addition, there are a number of studies that investigated performance issues (such throughput and latency). There are a few studies that utilised modelling and simulation techniques to investigate issues in blockchain systems. Most of these studies concern about performance issues such as in [4,3,10].

However, we are not aware of any work that investigates the impact of profit uncertainty on miners' decisions in blockchain systems. Thus, we are to the best of our knowledge the first to utilise modelling and simulation techniques to investigate the uncertainty a miner perceives about the income and the cost of executing transactions.

3 Conceptual Model

This section discusses the problem that needs to be modelled and the construction of the simulation model. In addition, assumptions and simplifications that have been considered during the construction of the model are stated.

[4] Fixed amount of money. For example, the block reward in Ethereum is 5 Ether.

3.1 Problem Formalisation

The profit a miner can get from executing a transaction takes into account the income (transaction fee) and the cost (CPU and storage costs) of the transaction [5]. The transaction fee is offered by the originator of the transaction to the miner who executed the transaction. This fee is considered as an income from a miner's perspective. The computational work required by the miner to execute a transaction is the cost of the transaction. The transaction fee is calculated as follows:

$$Transaction\ fee(in\ Ether) = used\ gas * gas\ price \tag{1}$$

Where the used gas is the total amount of gas used by the transaction. The more operations the transaction requires the more amount of gas will be used. Ether is the digital currency used in the Ethereum blockchain.

As we mentioned in Section 2.1, each miner can select any subset of transactions from her pool to execute and include in her block. Since each block has a limit of how many transactions it can have, miners usually priorities transactions by selecting the most profitable ones [5].

In blockcain systems such as Ethereum, however, the only information provided to the miner about the profit of executing a transaction is the maximum income (gas limit and gas price). Miners do not know the exact income (used gas) they can obtain from a transaction prior to executing it. Not only this but also miners do not know in advance the cost of executing the transaction. This makes miners uncertain about the profit they can gain from executing a transaction. Thus, they are not able to make informed decisions about which transactions to select and execute in order to maximise their profits.

3.2 Model Assumptions and Simplifications

The following assumptions and simplifications have been considered in our model:

- We assume that miners execute the most profitable transactions first after sorting out all pending transactions in their pool. If the ith transaction cannot fit in the remaining space of the block, miners can skip to the next transaction and execute it if it fits into the block.
- We assume that miners fill each block by executing as many transactions as they could in order to maximise their block profit. However, in real blockchain system, miners can generate full, non-full or even empty blocks.
- We assume that the CPU time required to execute a transaction is representative for the transaction's cost.
- We simplify the model by only considering the income and the cost of executing and including transactions in a block in order to report about the block profit. Thus, we excluded the income and the cost incurred by generating a valid block.

3.3 Model Construction

We build a model to investigate the problem discussed in Section 3.1. The model is to represent and simulate the decisions that miners take to select a subset of

pending transactions to include in their forthcoming blocks. We divide the model into three main parts, namely, model inputs, model contents and model outputs. Figure 2 illustrates the model flowchart.

Model inputs: The model takes the miner's pool as an input. The pool has a number of pending transactions that are waiting to be executed. Each transaction has the following attributes:

- Gas limit: the maximum amount of gas unit the transaction can consume. The gas limit value is proposed by the originator of the transaction.
- Used gas: the amount of gas unit consumed by the transaction. The used gas value should be less than or equal to the gas limit value.
- Gas price: the amount of money (in Ether) the originator of the transaction is willing to pay for each gas unit consumed by the transaction.
- CPU time: the CPU time (in seconds) consumed by the transaction.

Model Contents: The model contents describe all steps and formulas needed to calculate the model outputs given the model inputs. To model the decisions that miners take to select and execute transactions in a block, we first sort pending transactions and then select a subset of these transactions to execute in the block.

- Sorting transactions: The first step is to sort all pending transactions in the miner's pool based on their profits. The profit can be determined based on the values of the transactions' attributes such as gas limit, used gas, gas price and CPU time. In section 6, we will design five different sorting scenarios.
- Selecting and executing transactions: After sorting all pending transactions, the miner selects the first transaction and then check if the transaction can fit in the block or not. If the transaction does not fit, the miner can select the next transaction. Otherwise, the miner executes the transaction and then check if the block still has space for other transactions or not. If the block does have space, the miner can select the next transaction. This process is then repeated until the block is full (no more transaction can fit in the block).

After executing each transaction, the income and the cost of executing the transaction are calculated and recorded. The income of executing a transaction is the transaction fee offered by the transaction (see equation 1 in section 3.1). The cost of executing the transaction is assumed to be the CPU time required to execute the transaction.

The block income is calculated as the sum of the income gained from all transactions in the block

$$Block\ Income(in\ Ether) = F_1 + F_2 + ... + F_n \tag{2}$$

where F_{i} is the income (fee) of the ith transaction and n is the total number of transactions executed in the block.

The block cost is calculated as the sum of the CPU time required to execute all

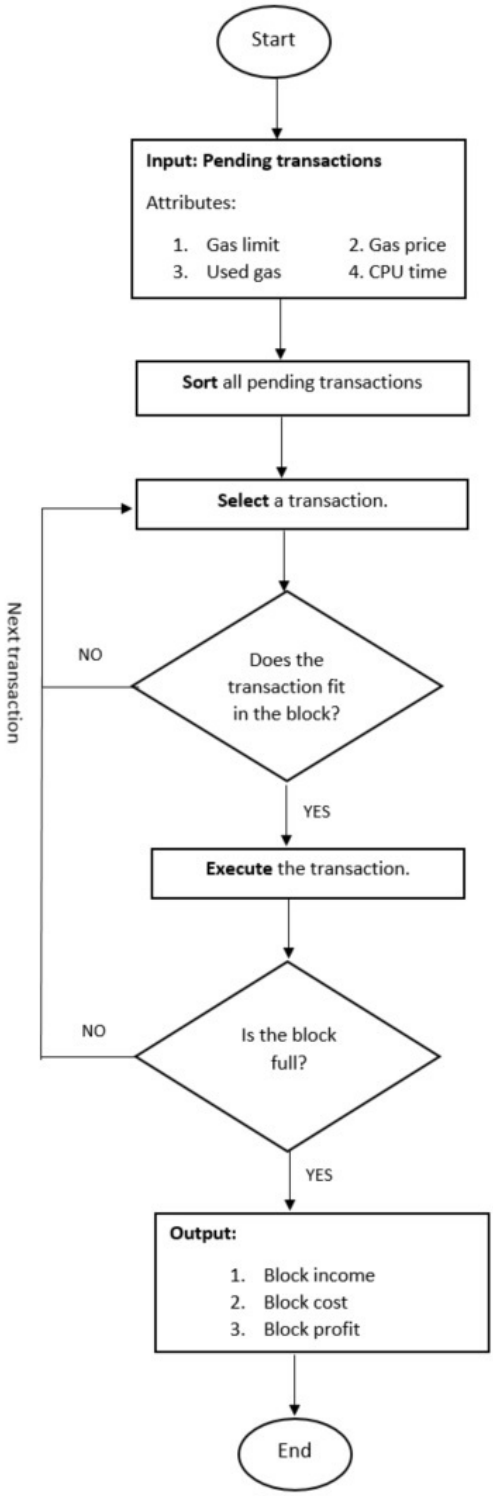

Fig. 2. Model Flowchart

transactions in the block

$$Block\ Cost(in\ second) = T_1 + T_2 + ... + T_n \tag{3}$$

where T_i is the CPU time consumed by the ith transaction and n is the total number of transactions executed in the block.

The block profit, that miners can get from executing all transactions in the block, is then calculated as follows:

$$Block\ Profit = \frac{Block\ Income}{Block\ Cost} \tag{4}$$

Model outputs: The model has three main outputs, namely, the block income, the block cost and the block profit. These outputs have been explained in the model contents.

4 Data Collection

The input data for the model that needs to be collected is the values of the transactions' attributes. These attributes are gas limit, gas used, gas price and CPU time. We collected this data from real historical data and from a controlled experiment.

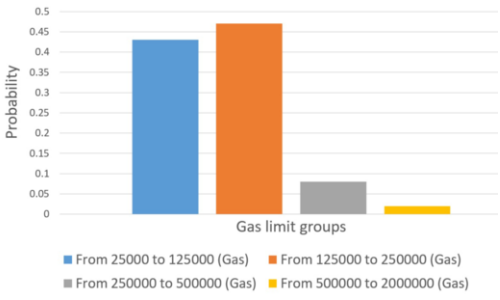

Fig. 3. Probability distribution of the gas limit values

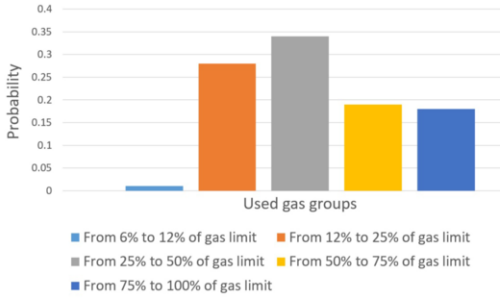

Fig. 4. Probability distribution of the used gas values

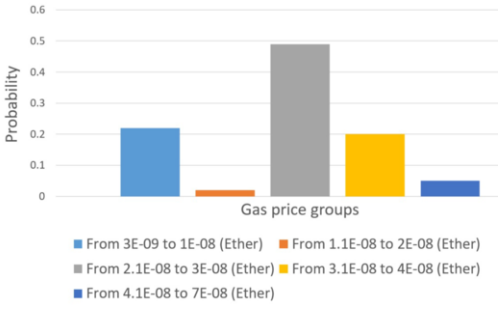

Fig. 5. Probability distribution of the gas price values

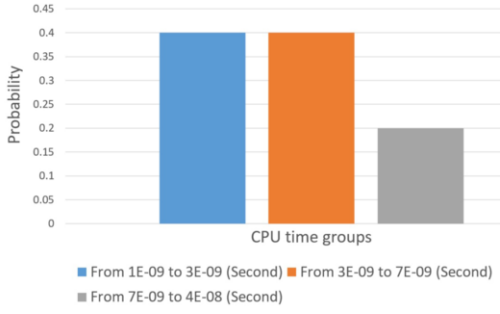

Fig. 6. Probability distribution of the CPU time values

Observation from real historical data: We relied on Etherscan [5] to collect

[5] https://etherscan.io

some historical data about gas limit, used gas and gas price. Since this is an initial study, we decided to collect the data from only 100 transactions. Those transactions were selected in a random way. Hence, we could write a separate paper about collecting all historical data. For each transaction selected, we recorded its gas limit, used gas and gas price values. Then, we built a probability distribution to describe the range of values for each input data as depicted in Figures 3-5.

For the gas limit, we found most of the transactions (90%) have a gas limit ranging from 25000 to 250000 gas. The remaining transactions have a gas limit between 250000 and 2 million gas. We have not found any transaction that has a gas limit of over 2 million gas. Thus, we classified the gas limit values into four groups as depicted in Figure 4.

For the used gas, we found that the gas used by 63% of the transactions is less than half of the proposed gas limit. Only 37% of the transactions used more than half of the proposed gas limit. Thus, we classified the used gas values into five groups as depicted in Figure 5.

For the gas price, we found that about half of the transactions offer a gas price ranging from 2.1E-08 to 3E-08 Ether per each gas unit. The remaining transactions offer a gas price either less than 2.1E-08 or more than 3E-08 Ether per each gas unit. Thus, we classified the gas price values into five groups as depicted in Figure 6.

Controlled Experiment: In order to collect data about the CPU time, we conducted a small controlled experiment on a private Ethereum network using a Laptop with Core i5 2.30 GHz CPU and 16GB RAM running Windows 10. The reason for this experiment is to generate a probability distribution to describe the range of values for the CPU time. We created and deployed five smart contracts. Then, we created twenty transactions, executed them and measured their execution time. Since each transaction uses a different amount of used gas, we calculated the CPU time required by each gas unit for all transactions.

We found that the CPU time required per each gas unit falls between 1E-09 and 7E-09 second in 80% of the transactions. The remaining 20% of the transactions consumes between 7E-09 and 4E-08 seconds per each gas unit. Thus, we classified the CPU time values into three groups as depicted in Figure 7.

5 Model Implementation

This section describes the implementation of the simulation model discussed in Section 3. The simulation model is implemented in Microsoft Excel by writing macros in Visual Basic. Algorithm 1 illustrates the implementation steps.

Algorithm 1 *The implementation steps of the model*
Step 1: Set the block limit = 6715126 gas, i = 1, the block income = 0, the block cost = 0, the block profit = 0, the number of pending transactions (N) = 150
Step 2: Input: Generate a random value for gas limit, used gas, gas price and CPU time for each pending transaction
Step 3: Sort pending transactions based on highest gas price value
Step 4: Select the ith transaction
Step 5: If the gas limit value of the ith transaction is more than the block limit value, then increase the value of i by one and go back to Step 4
Step 6: Otherwise, decrease the block limit by the used gas value of the ith transaction, increase the block income by multiplying the used gas value by the gas price value, increase the block cost by the CPU time value of the ith transaction
Step 7: If block limit value is more than zero, increase the value of i by one and go back to Step 4
Step 8: Calculate the block profit by dividing the block income by the block cost
Step 9: Output: print out the block income, the block cost and the block profit

Step 1: This step is to set initial values for the simulation's parameters. The block limit value is set to 6715126 gas[6]. The parameter i is just a counter to go through all transactions and it is set to 1 to start with the first transaction. The output data for the model which are the block income, the block cost and the block profit is set to zero. The number of pending transactions (N) is set to be 150 transactions, which is about the size of two blocks.

Step2: This step is to generate input data (transaction's attributes) for the model. These attributes are gas limit, used gas, gas price and PU time. Each attribute has a probability distribution of the values it can take. The probability distribution for each attribute was discussed and defined in Section 4. The value of each transaction's attribute was generated randomly from the probability distribution for that attribute. We used Rand() in Excel to draw a random value from these probability distributions for each attribute.

Step 3: This step is to sort pending transactions based on the value of the transactions' attributes. Under real blockchain system condition, only two attributes (gas limit and gas price) are available to the miners. Thus, miners can only sort transactions based on the highest gas price or based on the highest gas limit * gas price to maximise their profits (as we will discuss this in Section 6). In Algorithm 1, we sorted transactions based on highest gas price value.

Step 4: This step is to select a transaction to execute and include in a block. We started by selecting the first transaction which has the highest gas price value.

Step 5: This step is to check if the selected transaction (in Step 4) can fit in the block or not. A transaction can fit in a block if its gas limit value is less than or equal to the block limit value. If the transaction does not fit in the block, then the next transaction is selected by going back to Step 4.

[6] The block limit is dynamically adjusted by miners. As of 2nd October 2017, the average block limit was 6715126 gas.

Step 6: If the transaction fits in the block, then execute the transaction. After executing the transaction, the block limit value is decreased by the transaction's used gas value. The block income is increased by multiplying the used gas value by the gas price value. The block cost is increased by the transaction's CPU time value.

Step 7: This step is to check if the block still has space for other transactions or not. If the block limit value after executing the transaction in Step 6 is more than zero, then select the next transaction by going back to Step 4. Otherwise, go to Step 8.

Step 8: This step is to calculate the block profit after filling the block with transactions.

Step 9: This step is to print out the model outputs (the block income, the block cost and the block profit).

6 Experimental Design

This section discusses the objectives of this experiment and the design of the experiment's scenarios. Five scenarios (two baselines and three solutions) have been designed.

6.1 Objectives

In blockchain systems, miners face uncertainty about the exact income (transaction fee) they can get from executing each transaction. Not only this, but also miners face uncertainty about the cost (CPU time) of executing transactions. The objective of this experiment is to investigate the impact of the uncertainty miners perceive about the income and the cost of executing transactions on the block profit miners can get. To achieve this, we will compare the block profit that miners can get when there is uncertainty about the income and the cost of executing transactions with the case when there is certainty about the income and/or the cost of executing transactions.

6.2 Experiment Scenarios

For this experiment, we established five different scenarios to simulate the decisions that miners can take to sort pending transactions and then to select a subset of these transactions to execute and include in their forthcoming block. We classified these scenarios into two groups, namely, baseline and solution. Baseline group is to simulate miners' decisions under the real blockchain system condition, where miners are uncertain about income and the cost of executing transactions. The only information available to the miners in the baseline group is the maximum income (gas limit and gas price) they can get from executing transactions. Solution group is to simulate miners' decisions when miners are certain about the income and/or the cost of executing each transaction. Table 1 illustrates the scenarios in both the baseline and the solution groups.

Groups	Scenarios	Available Information	Sorting/Execution Criteria	Income Certainty	Cost Certainty
Baseline	Gas price	Gas limit	Highest gas price	NO	NO
	Maximum income	Gas price	Highest gas limit * gas price	NO	NO
Solution	Exact income	Used gas Gas price	Highest Used gas * gas price	YES	NO
	Maximum profit	Gas limit Gas price CPU time	Highest (gas limit * gas price) / CPU time	NO	YES
	Exact profit	Used gas Gas price CPU time	Highest (used gas * gas price)/ CPU time	YES	YES

Table 1
The scenarios of the baseline and the solution groups. There are two baseline scenarios and three solution scenarios. The "Available Information" column is to specify which transactions' attributes are available to the miners prior to sorting and executing transactions. The "Sorting/Execution Criteria" column is to state the decisions that miners take to sort and execute transactions in each scenario. The last two columns are to show the differences between the five scenarios in terms of the income and the cost certainty. For example, miners are certain about both the income and the cost of transactions in the exact profit scenario.

6.2.1 Baseline Group

We evaluated two possible scenarios, namely, gas price scenario and maximum income scenario.

- Gas price scenario: This scenario is to sort transactions based on their gas price values.
- Maximum income scenario: This scenario is to sort transactions based on the maximum income a miner can get from executing transactions. The maximum income is calculated as follows:

$$Maximum\ income = gas\ limit * gas\ price \tag{5}$$

6.2.2 Solution Group

We evaluated three solution-based scenarios, which are as follows:

- Exact income scenario: This scenario is to see whether the certainty about the income of executing transactions would help miners increase their block profit. In this scenario, we assume miners know the exact income they can get from executing each transaction in advance. The exact income is calculated as follows:

$$Exact\ income = transaction\ fee = used\ gas * gas\ price \tag{6}$$

- Exact cost scenario: This scenario is to see whether the certainty about the cost of executing transactions would help miners increase their block profit. In this scenario, we assume miners know the exact cost (CPU time) of executing each transaction in advance. Thus, miners in this scenario sort transactions based on the maximum profit they can get from executing transactions. The maximum profit is calculated as follows:

$$Maximum\ profit = \frac{maximum\ income}{CPU\ time} \tag{7}$$

- Exact profit scenario: This scenario is to see whether the certainty about both the income and the cost of executing transactions would help miners increase their block profit. In this scenario, we assume miners know both the exact income and the exact cost of executing each transaction in advance. Thus, miners in this scenario sort transactions based on the exact profit they can get from executing transactions. The exact profit is calculated as follows:

$$Exact\ profit = \frac{Exact\ income}{CPU\ time} \tag{8}$$

7 Results and Discussion

This section presents and discusses the results of the simulation's experiment described in Section 6. We run the simulation 668 times for each of the five scenarios. Each run is to simulate the process of sorting pending transactions and the process of selecting a subset of these transactions to execute and include in a block. In each run, we calculate the output data (the block income, the block CPU time and the block profit) for each scenario. A summary that shows the average of each output data for all the five scenarios is presented in Table 2. Figures 7-9 show the output data for each simulation run for all the five scenarios.

Scenarios	Average Block Income (in Ether)	Average Block CPU time (in Second)	Average Block Profit (Income / CPU time)
Gas price scenario	0.221	0.091	2.604
Maximum income scenario	0.205	0.091	2.538
Exact income scenario	0.199	0.091	2.488
Maximum profit scenario	0.196	0.021	9.538
Exact profit scenario	0.201	0.021	10.070

Table 2
A summary of the experiment's outputs for all the five scenarios. The experiment's outputs are the average block income, the average block CPU time and the average block profit. The average value for each output is taken from 668 simulation runs. The confidence intervals (95%) for the experiment's outputs are not given here, but all intervals are within 3% of the average value.

For the baseline scenarios, our simulation results show that the average block income gained in both scenarios were almost the same (0.221 Ether for the gas price scenario and 0.205 Ether for the maximum income scenario). In addition, the average block CPU time for both scenarios was the same (0.091 second). Thus, there is no significant difference between the two scenarios in terms of the block profit that miners can get from executing and including transactions in a block.

In the first solution-based scenario (exact income scenario), miners are given the exact income they can get from executing each transaction. This is to investigate the impact of income uncertainty by comparing the block profit that miners can get in this scenario with that for the baseline scenarios. From the simulation results, we have not found any improvement in the block profit that miners can get in this scenario compared to the block profit they can get in the baseline scenarios. In other words, the certainty about the income of executing transactions does not help miners increase their block profit. This might be due to the fact that each block has a limit of how many transactions it can have, and thus, the block income is limited

by the block limit. Even if miners are uncertain about the income of transactions, they can execute transactions until the block is full to maximise their block income.

In the second solution-based scenario (maximum profit scenario), miners are given the exact cost (CPU time) consumed by each transaction. This is to investigate the impact of cost uncertainty by comparing the block profit that miners can get in this scenario with that for the baseline scenarios. From the simulation results, we have found that the block profit that miners can get in this scenario is about four times the profit they can get in the baseline scenarios. The average block CPU time for this scenario is about a quarter the average block CPU time for the baseline scenarios (0.021 seconds in the maximum profit scenario compared to 0.091 seconds in the baseline scenarios). In other words, the certainty about the cost of executing transactions can help miners to quadruple their block profit.

In the last solution-based scenario (exact profit scenario), miners are given the exact income and the exact CPU time consumed by each transaction. This is to investigate the impact of income and cost uncertainty by comparing the block profit that miners can get in this scenario with that for the baseline scenarios. From the simulation results, we have found a significant difference between this scenario and the baseline scenarios in terms of the block profit that miners can get. Miners in this scenario can get four times as much block profit compared to miners in the baseline scenarios. However, we have not found a significant difference between this scenario and the second scenario in terms of the block profit. This is because the exact income a miner can get from executing a transaction does not have an impact on the block profit as we mentioned earlier.

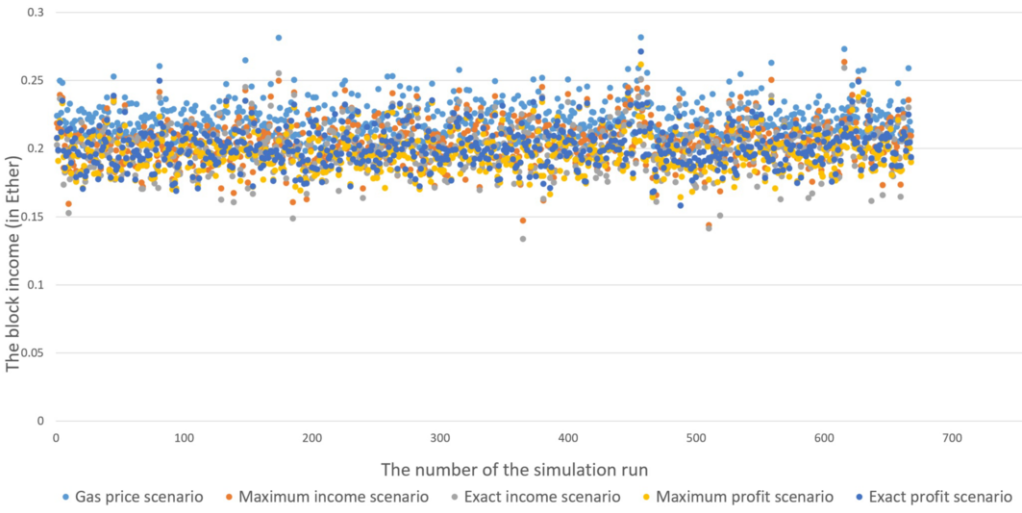

Fig. 7. The block income per each simulation run for all the five scenarios. The X-axis represents the number of the simulation run (668 runs in total) and the Y-axis represents the block income (in Ether). Each colour is to represent a different scenario. For example, the yellow colour represents the maximum profit scenario. It is obvious that there is no significant difference between the five scenarios in term of the block income. That means the uncertainty about the income of executing transactions does not have an impact on the block profit that miners can get.

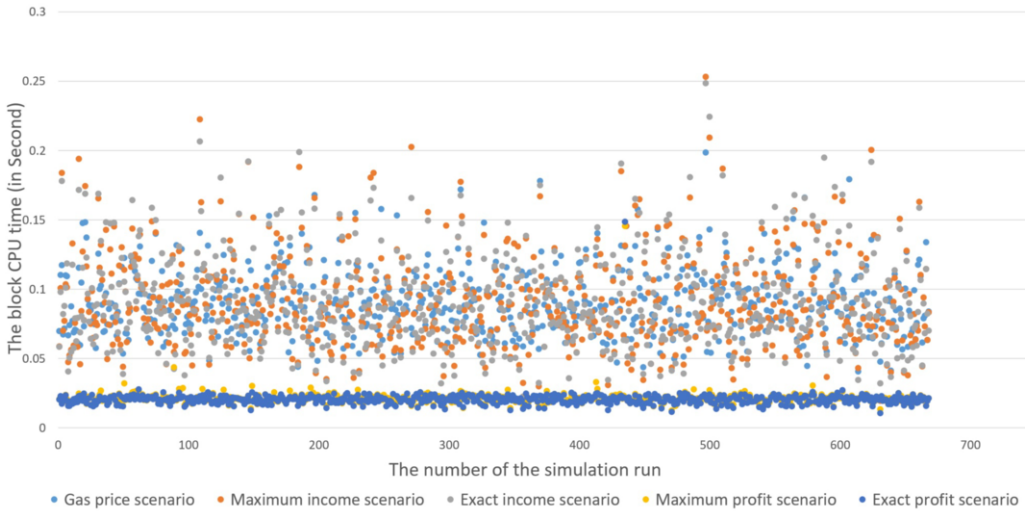

Fig. 8. The block CPU time per each simulation run for all the five scenarios. The X-axis represents the number of the simulation run (668 runs in total) and the Y-axis represents the block CPU time (in Second). It is obvious that the block CPU time for the maximum profit and the exact profit scenarios is significantly less than that for other scenarios. That means the uncertainty about the cost (CPU time) of executing transactions can have a significant impact on the block profit that miners can get.

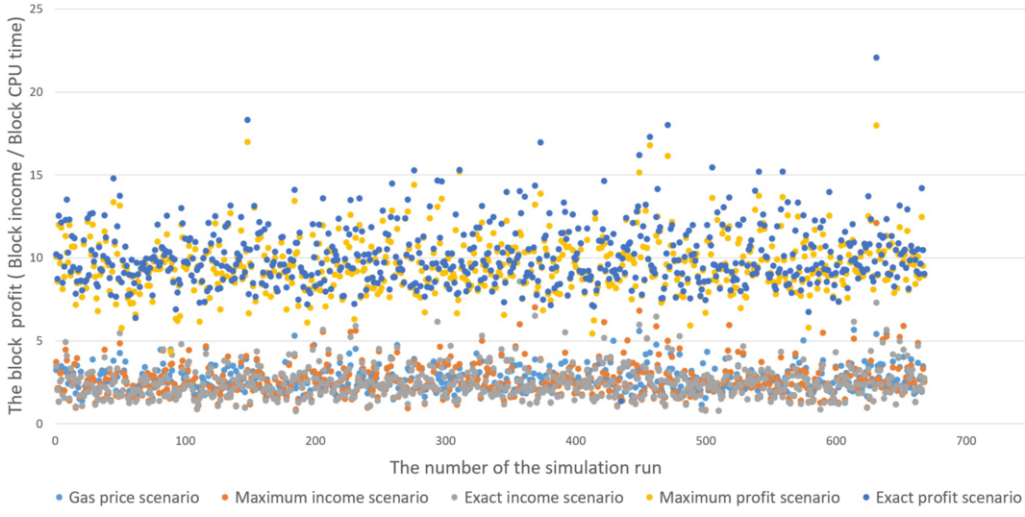

Fig. 9. The block profit per each simulation run for all the five scenarios. The X-axis represents the number of the simulation run (668 runs in total) and the Y-axis represents the block profit (the block income / the block CPU time). It is obvious that the block profit for the maximum profit and the exact profit scenarios is significantly more than that for other scenarios. This is because miners in these two scenarios are certain about the cost (CPU time) of executing transactions.

8 Conclusion

This paper has investigated the uncertainty miners perceive about the income and the cost of executing transactions and the impact of this uncertainty on the profit that miners can gain. We designed a simulation model to simulate the decisions that

miners take to select and execute transactions in a block, collected the input data for the model from real historical data and from a controlled experiment and then implemented the model. After that, we designed and run a simulation experiment to compare the block profits that miners can get when there is uncertainty about both the income and the cost of executing transactions with the case when there is certainty about the income and/or the cost of executing transactions.

Our simulation results show that the uncertainty miners perceive about the income of executing and including a transaction in a block does not have an impact on the block profit. However, the uncertainty miners perceive about the cost of executing and including a transaction in a block has a significant impact on the block profit. The block profit miners can get when there is uncertainty about the cost of executing transactions is about quarter the profit that they would get when there is certainty about the cost of executing transactions.

The limitations of this work that will be addressed in our future work are as follows. First, we assumed that the CPU time required to execute a transaction is representative for the cost of executing the transaction. We will extend this work by finding a way to report about the cost of executing transactions (e.g., considering storage costs, energy costs and hardware costs). Second, we simplified this model by only considering the income and the cost of executing and including transactions in a block. We have not considered the income and the cost incurred by generating a block of transactions. Also, we have not considered "The block generation competition" that we explained in Section 2.1. We will extend this model to account for the income and the cost of generating a block as well as considering "The block generation competition" to report about the block profit.

References

[1] Alharby, M. and A. van Moorsel, *Blockchain-based smart contracts: A systematic mapping study*, in: *Proceedings of the 4th International Conference on Computer Science and Information Technology (CSIT-2017)*, AIRCC Publishing Corporation, 2017, pp. 125–140.

[2] Buterin, V., *A next-generation smart contract and decentralized application platform.* Available online at: https://github.com/ethereum/wiki/wiki/White-Paper/ [Accessed 19/10/2017].

[3] Gervais, A., G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf and S. Capkun, *On the security and performance of proof of work blockchains*, in: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 3–16.

[4] Kasahara, S. and J. Kawahara, *Priority mechanism of bitcoin and its effect on transaction-confirmation process*, arXiv preprint arXiv:1604.00103 (2016).

[5] Miller, A., *Gas economics* Available online at: https://github.com/LeastAuthority/ethereum-analyses/blob/master/GasEcon.md [Accessed 17/10/2017].

[6] Nakamoto, S., *Bitcoin: A peer-to-peer electronic cash system* (2008).

[7] Wood, G., *Ethereum: A secure decentralised generalised transaction ledger*, Ethereum Project Yellow Paper **151** (2014).

[8] Wood, G., *Ethereum: A secure decentralised generalised transaction ledger*, Ethereum Project Yellow Paper **151** (2014).

[9] Xu, X., C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran and S. Chen, *The blockchain as a software connector*, in: *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, IEEE, 2016, pp. 182–191.

[10] Yasaweerasinghelage, R., M. Staples and I. Weber, *Predicting latency of blockchain-based systems using architectural modelling and simulation*, in: *2017 IEEE International Conference on Software Architecture (ICSA)*, IEEE, 2017, pp. 253–256.