



**HAL**  
open science

## Strategic investments in distributed computing: A stochastic game perspective

Swapnil Dhamal, Walid Ben-Ameur, Tijani Chahed, Eitan Altman, Albert Sunny, Sudheer Poojary

► **To cite this version:**

Swapnil Dhamal, Walid Ben-Ameur, Tijani Chahed, Eitan Altman, Albert Sunny, et al.. Strategic investments in distributed computing: A stochastic game perspective. *Journal of Parallel and Distributed Computing*, 2022, 169, pp.317-333. 10.1016/j.jpdc.2022.07.012 . hal-03767272

**HAL Id: hal-03767272**

**<https://inria.hal.science/hal-03767272v1>**

Submitted on 1 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Strategic Investments in Distributed Computing: A Stochastic Game Perspective

Swapnil Dhamal<sup>1,\*</sup>, Walid Ben-Ameur<sup>1</sup>, Tijani Chahed<sup>1</sup>, Eitan Altman<sup>2</sup>, Albert Sunny<sup>3</sup>, Sudheer Poojary<sup>4</sup>

---

## Abstract

We study a stochastic game with a dynamic set of players, for modeling and analyzing their computational investment strategies in distributed computing. Players obtain a certain reward for solving a problem, while incurring a certain cost based on the invested time and computational power. We present our framework while considering a contemporary application of blockchain mining, and show that the framework is applicable to certain other distributed computing settings as well. For an in-depth analysis, we consider a particular yet natural scenario where the rate of solving the problem is proportional to the total computational power invested by the players. We show that, in Markov perfect equilibrium, players with cost parameters exceeding a certain threshold, do not invest; while those with cost parameters less than this threshold, invest maximal power. We arrive at an interesting conclusion that the players need not have information about the system state as well as each others' parameters, namely, cost parameters and arrival/departure rates. With extensive simulations and insights through mean field approximation, we study the effects of players' arrival/departure rates and the system parameters on the players' utilities.

*Keywords:* Stochastic games, Markov perfect equilibrium, Distributed computing, Blockchain mining

---

## 1. Introduction

Distributed computing systems comprise computers which coordinate to solve large problems. Distributed computing has gained more popularity than ever, owing to the advent of blockchain which has found applications in various fields [1], such as cryptocurrencies, smart contracts, security services, and Internet of Things. Its functioning relies on a proof-of-work (PoW) procedure [2], where miners (i.e., investors of computational power) collect block data consisting of a number of transactions, and repeatedly compute hashes on inputs from a very large search space. A miner is rewarded for mining a block, if it finds before all the other miners, one of the rare inputs that generates a hash value satisfying certain constraints (hence these rare inputs constitute the solution space). Given the cryptographic hash function, the best known method for finding such an input is randomized search (i.e., iteratively drawing elements from the search space until an element belonging to the solution space is found). Since the PoW procedure is computationally intensive, successful mining requires a miner to invest significant computational power, resulting in the miner incurring some cost. Once a block is mined, it is transmitted to all the other miners, and the process repeats for mining a new block.

In the block mining scenario, when running a randomized search algorithm, the probability of a miner being the first to solve the problem and hence winning the reward for mining a block, is proportional to its invested power at the time. A

miner's objective is to maximize its utility based on the offered reward for mining a block before others, by strategizing on the amount of power to invest. There is a natural trade-off: a higher investment increases a miner's chance of solving the problem before others, while a lower investment reduces its incurred cost. The model that we propose in this paper effectively captures the blockchain setting (and as we shall show, the model also applies to the classical distributed computing setting where the providers of computational power are rewarded in proportion to their invested power). The focal point of this paper is to study a stochastic game where players (miners) can arrive and depart during block mining, and can strategize on their investment policies so as to maximize their own utilities.

### 1.1. Preliminaries

We now provide a non-technical introduction to the concepts used in this paper. We will describe their technical details once we propose our model in the next section.

- **Stochastic Game.** A stochastic game [3] is a dynamic game with probabilistic transitions across different system states. Players' payoffs and state transitions depend on the current state and the strategies of all players. The game continues until it reaches a terminal state, if any. Stochastic games are thus a generalization of both Markov decision processes and repeated games. They naturally capture interacting adaptive players since the players' strategies depend on the system state as well as the strategies of all the players.

- **Markov Perfect Equilibrium (MPE).** MPE [4] is an adaptation of subgame perfect Nash equilibrium to stochastic games. A player's MPE policy is a function describing its strategy for each state, while ignoring history. Each player computes its strategy in each state by foreseeing the effects of its actions on

---

\*Corresponding author

Email address: [swapnil.dhamal@gmail.com](mailto:swapnil.dhamal@gmail.com) (Swapnil Dhamal)

<sup>1</sup>Télécom SudParis, Institut Polytechnique de Paris, France

<sup>2</sup>INRIA Sophia Antipolis-Méditerranée, France

<sup>3</sup>Indian Institute of Technology, Palakkad, India

<sup>4</sup>Qualcomm India Pvt. Ltd.

the state transitions and the resulting utilities, as well as the strategies of other players in each state. A player's MPE policy is a best response to the other players' MPE policies.

While solution concepts such as MPE and Nash equilibrium may seem impractical due to the common knowledge assumption, they provide a profile which can be recommended to players (e.g., by a mediator) from which no player would unilaterally deviate. Alternatively, if players play the game repeatedly while observing each other's strategies, they are likely to settle at such a profile.

## 1.2. Related Work

Stochastic games have been studied from the theoretical perspective as well as in applications such as networks, queuing systems, multiagent reinforcement learning, and complex living systems. We enlist some of the works on stochastic games, relevant to ours. Altman and Shimkin [5] consider a processor-sharing system, where an arriving customer observes the current load on the shared system and chooses whether to join it or to use a constant-cost alternative. Nahir, Orda, and Raz [6] consider a similar setup, with the difference that customers consider using the system over a long time scale and for multiple jobs. Hassin and Haviv [7] propose a version of subgame perfect Nash equilibrium for games where players are identical, and each player selects a strategy based on its private information regarding the system state. Wang and Zhang [8] investigate Nash equilibrium in a queuing system, where reentering the system is a strategic decision. Hu and Wellman [9] use the framework of general-sum stochastic games to extend Q-learning to a noncooperative multiagent context. There exist works which develop algorithms for computing reasonably good, not necessarily optimal, strategies in a state-learning setting [10, 11].

Distributed systems have been studied from the game theoretic perspective [12, 13]. Wei et al. [14] study a resource allocation game in a cloud-based network, with constraints on quality of service. Chun et al. [15] analyze a selfish caching game, where selfish server nodes incur cost, either for replicating resources or for access to a remote replica. Grosu and Chronopoulos [16] propose a game theoretic framework for obtaining a user-optimal load balancing scheme in distributed systems. Kishora, Niyogia, and Veeravalli [17] study a multiobjective load balancing problem in distributed systems by formulating it as a non-cooperative game.

Zheng and Xie [1] present a survey on the challenges in blockchain and recent advances in tackling these challenges. Liu et al. [18] present a survey reviewing game models proposed in the literature for addressing common issues in the blockchain network, with a focus on security issues (e.g., selfish mining, majority attack, denial of service attack), mining management issues (e.g., computational power allocation, reward allocation, pool selection), and issues regarding blockchain economic and energy trading.

Eyal and Sirer [19] were among the first to conduct a game analysis on blockchain miners, by introducing selfish mining wherein a miner possessing enough computational power does

not propagate a block immediately, but generates forks intentionally by propagating a block selectively only when another honest miner generates a block. Sapirshstein, Sompolinsky, and Zohar [20] study selfish mining attacks, where a miner postpones transmission of its mined blocks so as to prevent other miners from starting the mining of the next block immediately. Lewenberg et al. [21] study pooled mining, where miners form coalitions and share the obtained rewards, so as to reduce the variance of the reward received by each player. Eyal [22] models a game between two pools employing 'block withholding' attack, and hence discovers the miner's dilemma wherein the revenues of both pools diminish in Nash equilibrium. Kwon et al. [23] propose 'fork after withholding' attack, which selectively alternates between performing withholding and selfish mining attacks; the corresponding reward is greater than or equal to that using the block withholding attack. Wang et al. [24] aim to solve the security and efficiency problems of consensus protocols in blockchain with the help of a reputation scheme that incentivizes nodes to participate in network collaboration and exhibit cooperative behavior. Zappalà et al. [25] propose a game theoretic framework for characterizing the robustness of blockchain systems, as they are prone to classical fault patterns as well as various agent behaviors that exist in typical economic systems.

Pass and Shi [26] present a new blockchain protocol, which is shown to be approximately fair in terms of reward guarantee, coalition-safe with regard to coalitions controlling less than a majority of the computing power, and having a low variance of mining rewards (thus lessening the need for mining pools). Chen et al. [27] present an axiomatic theory of incentives in PoW blockchains at the time scale of a single block and a set of desirable properties that any good reward allocation rule should satisfy, and hence study the properties satisfied by Bitcoin's allocation rule as well as other reward allocation rules.

Xiong et al. [28] consider that miners can offload the mining process to an edge computing service provider; they study a Stackelberg game where the provider sets price for its services, and the miners determine the amount of services to request. Chen et al. [29] consider a Stackelberg game with multiple edge computing service providers and miners, and hence compute an equilibrium solution wherein each service provider achieves the maximum revenue and each miner obtains utility-maximized resources under budget constraints. Qiu and Li [30] attempt to prevent bid-rigging in mobile blockchain edge computing resource allocation with the help of an auction mechanism whose reserve price is obtained using a simulated annealing algorithm; it is experimentally shown to achieve individual rationality and incentive compatibility, as well as increase the edge service provider's revenue.

Altman et al. [31] model the competition over several blockchains as a non-cooperative game, where a player's strategy comprises which blockchains to mine for and which edge computing service providers to request computing power from; the existence of pure Nash equilibria is shown using a congestion game approach. When formulating a game where players aim to determine the amount of power to invest, the authors model the utility function along the lines of [32]. A critical shortcom-

ing of the model (as acknowledged in [32]) is that an explicit consideration of time is missing. In particular, the utility function therein inherently assumes that the incurred cost does not depend on the time spent for mining (or alternatively, the time required for mining a block is independent of the amounts of power invested by the players). That is, the utility function does not capture a practical scenario where the cost incurred would depend on the time spent for mining a block, which in turn may change based on the change in the combined power invested by the players. This scenario shall be captured in the utility function formulated in our paper.

Haghighat and Shajari [33] model the block withholding game as a stochastic game with finitely many states and actions, and hence analyze it using a reinforcement learning method. Kiayias et al. [34] consider a stochastic game, where each state corresponds to the mined blocks and the players who mined them; players strategize on which blocks to mine and when to transmit them. Biais et al. [35] study a stochastic game where a state contains information regarding the tree of solved blocks, and a player’s strategy comprises which chain to consider for mining the next block. The authors show that there generally exists MPE such that on the equilibrium path, players always mine the chain that they perceive to be the longest.

In general, there exist game theoretic studies for distributed systems, as well as stochastic games for applications including blockchain mining (where a state, however, signifies the state of the chain of blocks). To the best of our knowledge, this work is the first to study a stochastic game for distributed computing considering the set of players to be dynamic. We consider the most general case of heterogeneous players; the cases of homogeneous players as well as multi-type players (which also have not been studied in the literature) are special cases of this study. Moreover, in the current literature, there do not exist mathematical models which capture the game in our studied distributed computing scenario. Hence, we first present our model, followed by our analysis of MPE in the induced stochastic game.

### 1.3. Our Contributions

- We develop a stochastic game model which captures the arrival and departure of players in the studied distributed computing scenario, with particular emphasis on mining in blockchain. We hence derive a closed form expression for a player’s utility function (Section 2).
- We present a game theoretic analysis and prove that in MPE, players with cost parameters exceeding a certain threshold, do not invest; while those with cost parameters less than this threshold, invest maximal power. This shows that players are not required to have common knowledge with respect to the system state and other players’ parameters (Section 3).
- We formulate different types of expected utilities that a player can compute depending on whether the player is present, absent, or agnostic with respect to its presence in the system, using results from Engset’s system. We obtain closed-form expressions that approximate these expected utilities, using a mean field approach (Section 4).

Table 1: Notation

$r$	reward parameter
$\gamma$	rate constant of proportionality for the problem being solved
$c_i$	cost incurred by player $i$ when it invests unit power for unit time
$\lambda_i$	arrival rate corresponding to player $i$
$\mu_i$	departure rate corresponding to player $i$
$\mathcal{U}$	universal set of strategic players
$\ell$	amount of fixed power
$S$	set of strategic players currently present in the system
$x_i^{(S)}$	strategy of player $i$ in state $S$
$\mathbf{x}^{(S)}$	strategy profile of players in state $S$
$\mathbf{x}$	policy profile
$\Gamma^{(S, \mathbf{x}^{(S)})}$	rate of problem getting solved in state $S$ under strategy profile $\mathbf{x}^{(S)}$
$R_i^{(S, \mathbf{x})}$	expected utility of $i$ computed in state $S$ under policy profile $\mathbf{x}$

- We study the effects of the arrival and departure rates and other system parameters on players’ utilities, using extensive simulations and insights through mean field approximation. In particular, we study the trade-off among factors such as alterations in the competition, the likelihood of staying out of the system, and the rate of problem getting solved (Section 5).

## 2. Our Model

We now model the distributed computing scenario wherein players receive a certain reward for being the first to solve a problem, such as mining in blockchain. We later briefly discuss that the obtained utility function generalizes to certain other distributed computing scenarios.

### 2.1. Model Description

In what follows, we will use ‘miner’ and ‘player’ interchangeably for ease of exposition. Let  $r$  be the reward offered to a miner for successfully mining a block, i.e., finding a PoW solution before all the other miners.

- **Players.** We consider that there are broadly two types of players (miners) in the system, namely, (a) strategic players who can arrive and depart while a problem is being solved (e.g., during the mining of a block) and can modulate the invested power based on the system state so as to maximize their respective expected utilities and (b) fixed players who are constantly present in the system and invest a constant amount of power for large time durations (e.g., large mining firms). In blockchain mining, the universal set of players during the mining of a block consists of all those who are registered as miners at the time. We denote by  $\mathcal{U}$ , the universal set of strategic players during the mining of the block under consideration. We denote by  $\ell$ , the amount of fixed power (i.e., the constant amount of power collectively invested by the fixed players) throughout the mining of the block. We consider  $\ell > 0$  (thanks to mining firms); so the mining does not stall even if the set of strategic players is empty. Since the fixed players are constantly present in the system and invest a constant amount of power, we denote them

as a single aggregate player  $k$ , who invests a constant power of  $\ell$  irrespective of the system state.

As it may not be practically feasible for a player to manually modulate its invested power as and when the system changes its state, we consider that the power is modulated by a pre-configured automated software on the player's machine. The player can strategically devise its policy (how much to invest if the system is in a given state). In fact, we will later see that in the blockchain mining scenario, a player's MPE policy comprises a strategy that is common for all the states; so the state knowledge assumption turns out to be redundant.

• **Cost Parameters.** We consider that players are Markovian, that is, a player aims to maximize its expected utility (i.e., the expected reward it would obtain minus the expected cost it would incur) from the current time onwards; it is not specially concerned about the reward obtained or the cost incurred thus far. A rationale behind such a player is that since the past utility is unalterable, maximizing its expected utility during the past and future as a whole, is equivalent to maximizing its expected utility from the current time onwards. This is a common assumption inherently made in several game models in the literature, including those concerning mining in blockchain (e.g., [31, 32]). We denote by cost parameter  $c_i$ , the cost incurred by player  $i$  for investing unit amount of power for unit time. It is important to note that while the cost parameters of all the players are considered to be common knowledge, our analysis will prove that a player's MPE policy depends on only its own cost parameter; so the common knowledge assumption with respect to cost parameters turns out to be redundant. Nonetheless, it is an interesting future direction to design incentives for players to reveal their true costs.

• **Properties of Computation.** A natural theoretical model can be developed based on the nature of computation involved in block mining. As described earlier, the computation involves a randomized search over the search space for finding an element belonging to the solution space. The search space is exponentially large as compared to the solution space. For instance, as of year 2022, the odds of finding a solution in Bitcoin mining is one in tens of trillions [36, 37]. When a player randomly draws an element from the search space, it is with near-zero probability that the same element will be drawn again. So, it is immaterial whether the previously drawn elements are memorized. Hence, the search is *memoryless*, owing to which the time required to find a solution in the large search space is independent of the search space explored thus far. Consequently, this time is exponentially distributed since, if a continuous random variable has the memoryless property over the set of reals, it is necessarily exponentially distributed. This can be easily corroborated by simulating a randomized search over a given search space; it can be observed that the time to find an element belonging to a given solution space is exponentially distributed. To put forth the aforementioned argument concisely, the pseudo-random properties of cryptographic hash functions make mining, a memoryless process [38], owing to which the Nakamoto protocol [2] essentially formalizes an exponential race [18]. The memoryless property of mining and hence the

exponential distribution of the mining time, are well accepted conventions in the literature (e.g., [18, 32, 35, 38]).

• **Arrival and Departure of Players.** We consider a standard setting for modeling the arrivals and departures of players. A player  $j$ , who is not in the system, arrives after time which is exponentially distributed. Let the expected time after which player  $j$  would arrive, if it is not already in the system, be denoted by  $\frac{1}{\lambda_j}$  (hence,  $\lambda_j$  can be interpreted as the rate parameter). Such a stochastic arrival of players is natural, like in most applications. Further, a player would depart by shutting down its computer or terminating the computationally demanding mining task (by closing the automated software) so as to run other critical tasks. Similar to arrival, the departure time of a player  $j$ , who is in the system, is exponentially distributed with rate parameter  $\mu_j$  (the expected time after which player  $j$  would depart is  $\frac{1}{\mu_j}$ ). Such a stochastic departure can be attributed to the memoryless nature of computation and the Markovian nature of players. In particular, owing to the memoryless nature of computation, players have no incentive to account for the time and power they have invested thus far for mining an ongoing block, since the additional time that would be required to mine the block is independent of this information. Moreover, the Markovian players are concerned about the expected utility from the current time onwards, and not about the time and hence cost incurred thus far for mining an ongoing block. Owing to these two factors, players are oblivious to the block mining progress or the time and power they have invested thus far for mining an ongoing block, and hence depart stochastically.

Note that players could potentially have their own individual arrival and departure rate parameters. So, it is more suitable to view the arrivals/departures of different players as independent events with their individual rate parameters, rather than aggregating them like in a queueing setting. Hence, we consider the most general case where players are heterogeneous. Our analysis and results directly apply to the special cases, namely, the multi-type case where  $c_i = c_j, \lambda_i = \lambda_j, \mu_i = \mu_j$  for players  $i, j$  of the same type; and the homogeneous case where  $c_i = c_j, \lambda_i = \lambda_j, \mu_i = \mu_j$  for all  $i, j$ . Further, note that the homogeneous case is mathematically equivalent to a queueing setting.

• **State Space.** Due to the arrivals and departures of strategic players, we could view this as a continuous time multi-state process, where a state corresponds to the set of strategic players present in the system. So, if the set of strategic players in the system is  $S$  (which excludes the fixed players), we say that the system is in state  $S$ . Hence,  $S \subseteq \mathcal{U}$ , i.e.,  $S \in 2^{\mathcal{U}}$ . The players involved at any given time would influence each others' utilities, thus resulting in a game. The stochastic arrival and departure of players makes it a stochastic game. As we will see, there are also other stochastic events in addition to the arrivals and departures, and which depend on the players' strategies.

• **Players' Strategies.** Let  $x_i^{(S, \tau)}$  denote the strategy of player  $i$  (amount of power it decides to invest) at time  $\tau$  if the system is in state  $S$ . As explained earlier, the time required to find a solution is independent of the search space explored thus far. Owing to this memoryless property, a player has no incentive

to change its strategy amidst a state, if no other player changes its strategy. Hence in our analysis, we consider that no player changes its strategy within a state. So, we have  $x_i^{(S,\tau)} = x_i^{(S,\tau')}$  for any  $\tau, \tau'$ ; hence player  $i$ 's strategy could be written as a function of the state alone, that is,  $x_i^{(S)}$ . For a state  $S$  where  $j \notin S$ , we have  $x_j^{(S)} = 0$  by convention. Let  $\mathbf{x}^{(S)}$  denote the strategy profile of the players in state  $S$ . Let  $\mathbf{x} = (\mathbf{x}^{(S)})_{S \subseteq \mathcal{U}}$  denote the policy profile.

• **Rate of the Problem Getting Solved.** As explained earlier, the time required to find a solution in a large search space is independent of the search space explored thus far, and this time is exponentially distributed. Let  $\Gamma^{(S,\mathbf{x}^{(S)})}$  be the corresponding rate of problem getting solved in state  $S$ , when players' strategy profile is  $\mathbf{x}^{(S)}$ . Further, Zeng and Zuo [39] show that if the number of solutions is  $\xi$ , the distance of the probability of a player finding a solution before others, from being proportional to the player's invested power, is  $\tilde{O}(\frac{1}{\xi})$ . Since  $\xi$  is typically large in blockchain mining, this distance is practically insignificant. Hence, in practice, the probability that a player finds a solution before others at time  $\tau$  is proportional to its invested power at time  $\tau$ . This is a widely accepted convention in the literature (e.g., [31, 32, 35]).

Note that the time required for the problem to get solved is the minimum of the times required by the players to solve the problem. Now, the minimum of exponentially distributed random variables, is another exponentially distributed random variable with rate which is the sum of the rates corresponding to the original random variables. Furthermore, the probability of an original random variable being the minimum, is proportional to its rate. Let  $P_j^{(S,\mathbf{x}^{(S)})}$  be the rate (corresponding to an exponentially distributed random variable) of player  $j$  solving the problem in state  $S$ , when the strategy profile is  $\mathbf{x}^{(S)}$ . So, we have  $\sum_{j \in S \cup \{k\}} P_j^{(S,\mathbf{x}^{(S)})} = \Gamma^{(S,\mathbf{x}^{(S)})}$ . Since the probability that player  $i$  solves the problem before the other players is proportional to its invested power at that time, we have that the rate of player  $i$  solving the problem is  $P_i^{(S,\mathbf{x}^{(S)})} = \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$ , and the rate of other players solving the problem is  $Q_i^{(S,\mathbf{x}^{(S)})} = \sum_{j \in (S \setminus \{i\}) \cup \{k\}} P_j^{(S,\mathbf{x}^{(S)})} = \frac{\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$ .

• **Discounting the Future.** Consider that a player  $i$  perceives its utility to be discounted by a factor of  $\delta \in [0, 1)$  for every future block, where  $\delta = 0$  means that the utility corresponding to only the current block is valued while that corresponding to future blocks is perceived as zero. Note that while we consider a common discounting factor for all players, our analysis goes through as is even if different discounting factors are considered for different players.

• **The Continuous Time Markov Chain.** Owing to the players being Markovian, when the system transits from state  $S$  to state  $S'$ , each player  $j \in S \cap S'$  could be viewed as effectively reentering the system. So, the expected utility of player  $i$  as computed in state  $S$ , say  $R_i^{(S,\mathbf{x})}$ , could be written in a recursive form, which we now derive. Table 1 presents the notation. The possible events that can occur in a state  $S \in 2^{\mathcal{U}}$  are:

1. the problem gets solved by player  $i$  with rate  $P_i^{(S,\mathbf{x}^{(S)})}$ , player  $i$  gets a reward of  $r$ , and the system stays in state  $S$  for the mining of the next block where  $i$ 's expected utility would be perceived as  $\delta R_i^{(S,\mathbf{x})}$ ;
2. the problem gets solved by one of the players in  $(S \setminus \{i\}) \cup \{k\}$  with rate  $Q_i^{(S,\mathbf{x}^{(S)})}$ , player  $i$  gets no reward, and the system stays in state  $S$  for the mining of the next block where  $i$ 's expected utility would be perceived as  $\delta R_i^{(S,\mathbf{x})}$ ;
3. a player  $j \in \mathcal{U} \setminus S$  arrives with rate  $\lambda_j$ , and the system transits to state  $S \cup \{j\}$  where  $i$ 's expected utility would be  $R_i^{(S \cup \{j\}, \mathbf{x})}$ ;
4. a player  $j \in S$  departs with rate  $\mu_j$ , and the system transits to state  $S \setminus \{j\}$  where  $i$ 's expected utility would be  $R_i^{(S \setminus \{j\}, \mathbf{x})}$ .

In what follows, for avoiding confusion, we use the term *fixed power* when referring to the power invested by the fixed players, and we do not use the term *fixed players*. So, henceforth, players would always refer to strategic players. Also, we unambiguously write  $j \in \mathcal{U} \setminus S$  as  $j \notin S$ , for brevity. Since  $P_i^{(S,\mathbf{x}^{(S)})} + Q_i^{(S,\mathbf{x}^{(S)})} = \Gamma^{(S,\mathbf{x}^{(S)})}$ , the sojourn time in state  $S$  is  $(\Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j)^{-1}$ . Let  $B^{(S,\mathbf{x})} = \Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ . The expected cost to be incurred is calculated by multiplying the cost parameter  $c_i$  with the power to be invested and the expected time to be spent. So, the expected cost to be incurred by player  $i$  while the system is in state  $S$  is  $\frac{c_i x_i^{(S)}}{B^{(S,\mathbf{x})}}$ .

• **Utility Function.** The probability of an event occurring before any other event is equivalent to the corresponding exponentially distributed random variable being the minimum, which in turn, is proportional to its rate. So, player  $i$ 's expected utility as computed in state  $S$  is

$$R_i^{(S,\mathbf{x})} := \frac{\Gamma^{(S,\mathbf{x}^{(S)})} \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}}{B^{(S,\mathbf{x})}} \cdot (r + \delta R_i^{(S,\mathbf{x})}) + \frac{\Gamma^{(S,\mathbf{x}^{(S)})} \sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell} \cdot (\delta R_i^{(S,\mathbf{x})}) - \frac{c_i x_i^{(S)}}{B^{(S,\mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})}$$

$$\iff R_i^{(S,\mathbf{x})} := \frac{\Gamma^{(S,\mathbf{x}^{(S)})} \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}}{B^{(S,\mathbf{x})}} \cdot r + \frac{\delta \Gamma^{(S,\mathbf{x}^{(S)})}}{B^{(S,\mathbf{x})}} \cdot R_i^{(S,\mathbf{x})} - \frac{c_i x_i^{(S)}}{B^{(S,\mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} \quad (1)$$

Note that Equation (1) is defined for a state  $S$  even if  $S \not\ni i$ .

#### Other Applications

We derived Equation (1) for the expected utility by considering the application of mining in blockchain, where the probability of player  $i$  being the first to solve the problem and hence obtaining reward  $r$ , is proportional to its invested power at the time. Now, consider a classical distributed computing setting such as volunteer computing where players are rewarded in proportion to their invested power. Here, a central entity aims to

complete certain tasks, which would generally be computationally demanding [40, 41], for which it seeks computational investment from players (volunteers) in return of some reward. Define a *segment* to be section of time between consecutive instances of dispensing reward. The length of a segment may coincide with the task completion time or may be determined by the center based on the set of players present and their invested powers. One such natural distribution for segment's length is exponential when the said task involves computation with memoryless property (e.g., block mining, finding solution to NP-hard problem, etc.). Let  $\Gamma^{(S, \mathbf{x}^{(S)})}$  be the rate parameter, which can be dependent on  $S$  and  $\mathbf{x}^{(S)}$  in general.

Let  $r$  be the expected total reward that the center can dispense per segment. Consider that as a segment initiates, the center keeps a continuous-time account of the reward amount that a player should receive at each time instant, and dispenses the aggregate amount at the end of the segment. Say the reward allocated to a player at a time instant is proportional to its invested power at that instant. Consider that in state  $S$ , the total reward accounted per unit time is  $r\Gamma^{(S, \mathbf{x}^{(S)})}$ , of which the allocation to a player  $i$  is  $\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\Gamma^{(S, \mathbf{x}^{(S)})}$ . With this formulation, it can be seen that for any segment, the expected total reward dispensed would be  $r$  and that dispensed to a player would be proportional to its invested power over the segment, which is as desired. Additionally, the continuous-time accounting lets players compute the reward they would be allocated at a time instant if they invested a certain power at that instant.

Consequently, as computed in state  $S$ , player  $i$ 's profit per unit time is  $\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\Gamma^{(S, \mathbf{x}^{(S)})} - c_i x_i^{(S)}$ . Similar to mining in blockchain, the possible events that can occur in state  $S$  are: the ongoing segment terminating with rate  $\Gamma^{(S, \mathbf{x}^{(S)})}$ , a player  $j \notin S$  arriving with rate  $\lambda_j$ , and a player  $j \in S$  departing with rate  $\mu_j$ . As the sojourn time in state  $S$  for an ongoing segment is  $\frac{1}{B^{(S, \mathbf{x})}} = (\Gamma^{(S, \mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j)^{-1}$ , the net expected profit made by player  $i$  in state  $S$  before any arrival/departure or ad-

vancing to the next segment, is  $\frac{\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\Gamma^{(S, \mathbf{x}^{(S)})} - c_i x_i^{(S)}}{B^{(S, \mathbf{x})}}$ . Note that this is equal to the combination of the first and third terms in Equation (1). Thus, the mathematical form of player  $i$ 's expected utility as computed in state  $S$  (i.e.,  $R_i^{(S, \mathbf{x})}$ ) stays unchanged, and so Equation (1) is applicable here as well.

Note also that apart from mining in blockchain at individual level, our model is applicable to mining pools offering pay-per-share payouts (the argument goes on similar lines as given above for volunteer computing). In general, our model is applicable to a variety of distributed computing applications that share the same underlying concepts.

## 2.2. A Closed-form Expression for the Expected Utility

Let us define an ordering  $\mathcal{O}$  on sets, which presents a one-to-one mapping from a set  $S \subseteq \mathcal{U}$  to an integer between 1 and  $2^{|\mathcal{U}|}$ , both inclusive. Let  $\mathbf{R}_i^{(\mathbf{x})}$  be the vector whose component  $\mathcal{O}(S)$  is  $R_i^{(S, \mathbf{x})}$ . We now show that  $\mathbf{R}_i^{(\mathbf{x})}$  computed using the recursive Equation (1), converges for any policy profile  $\mathbf{x}$ .

Let  $\mathbf{M}^{(\mathbf{x})}$  be the state transition matrix, among the states corresponding to the set of (strategic) players present in the system. In what follows, instead of writing  $M^{(\mathbf{x})}(\mathcal{O}(S), \mathcal{O}(S'))$ , we simply write  $M^{(\mathbf{x})}(S, S')$  since it does not introduce any ambiguity. So, the elements of  $\mathbf{M}^{(\mathbf{x})}$  are

$$\begin{aligned} M^{(\mathbf{x})}(S, S) &= \frac{\delta \Gamma^{(S, \mathbf{x}^{(S)})}}{B^{(S, \mathbf{x})}} \\ \text{for } j \notin S : M^{(\mathbf{x})}(S, S \cup \{j\}) &= \frac{\lambda_j}{B^{(S, \mathbf{x})}} \\ \text{for } j \in S : M^{(\mathbf{x})}(S, S \setminus \{j\}) &= \frac{\mu_j}{B^{(S, \mathbf{x})}} \\ \text{and all other elements of } \mathbf{M}^{(\mathbf{x})} &\text{ are } 0 \end{aligned}$$

Here,  $B^{(S, \mathbf{x})} = \Gamma^{(S, \mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ . Since  $\ell > 0$ , we have  $\Gamma^{(S, \mathbf{x}^{(S)})} > 0$ . So,  $B^{(S, \mathbf{x})} > \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ . Hence,  $\mathbf{M}^{(\mathbf{x})}$  is strictly substochastic (i.e., sum of the elements in each of its rows is less than 1).

Let  $\mathbf{F}_i^{(S, \mathbf{x})}$  be the vector whose component  $\mathcal{O}(S)$  is  $F_i^{(S, \mathbf{x})}$ , where

$$F_i^{(S, \mathbf{x})} = \left( \frac{\Gamma^{(S, \mathbf{x}^{(S)})}}{\sum_{j \in S} x_j^{(S)} + \ell} r - c_i \right) \frac{x_i^{(S)}}{B^{(S, \mathbf{x})}}$$

**Lemma 1.** *The recursive equation for  $\mathbf{R}_i^{(\mathbf{x})}$ , Equation (1), converges.*

**PROOF.** Let  $\mathbf{R}_{i(t)}^{(\mathbf{x})} = (R_{i(t)}^{(1, \mathbf{x})}, \dots, R_{i(t)}^{(2^{|\mathcal{U}|}, \mathbf{x})})^T$ , where  $t$  is the iteration number and  $(\cdot)^T$  stands for matrix transpose. The iteration for the value of  $\mathbf{R}_{i(t)}^{(\mathbf{x})}$  starts at  $t = 0$ ; we examine if it converges when  $t \rightarrow \infty$ . Now, the expression for the expected utility (Equation (1)) in all states can be written in matrix form, thus solving the recursion, as

$$\mathbf{R}_{i(t)}^{(\mathbf{x})} = \mathbf{M}^{(\mathbf{x})} \mathbf{R}_{i(t-1)}^{(\mathbf{x})} + \mathbf{F}_i^{(\mathbf{x})} = (\mathbf{M}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} + \left( \sum_{\eta=0}^{t-1} (\mathbf{M}^{(\mathbf{x})})^\eta \right) \mathbf{F}_i^{(\mathbf{x})}$$

Now, since  $\mathbf{M}^{(\mathbf{x})}$  is strictly substochastic, its spectral radius is less than 1. So when  $t \rightarrow \infty$ , we have  $\lim_{t \rightarrow \infty} (\mathbf{M}^{(\mathbf{x})})^t = \mathbf{0}$ . Since  $\mathbf{R}_{i(0)}^{(\mathbf{x})}$  is a finite constant, we have  $\lim_{t \rightarrow \infty} (\mathbf{M}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} = \mathbf{0}$ . Further,  $\lim_{t \rightarrow \infty} \sum_{\eta=0}^{t-1} (\mathbf{M}^{(\mathbf{x})})^\eta = (\mathbf{I} - \mathbf{M}^{(\mathbf{x})})^{-1}$  [42]. This implicitly means that  $(\mathbf{I} - \mathbf{M}^{(\mathbf{x})})$  is invertible. Hence,

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbf{R}_{i(t)}^{(\mathbf{x})} &= \lim_{t \rightarrow \infty} (\mathbf{M}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} + \left( \sum_{\eta=0}^{\infty} (\mathbf{M}^{(\mathbf{x})})^\eta \right) \mathbf{F}_i^{(\mathbf{x})} \\ &= \mathbf{0} + (\mathbf{I} - \mathbf{M}^{(\mathbf{x})})^{-1} \mathbf{F}_i^{(\mathbf{x})} \quad \square \end{aligned}$$

Now, since the recursive equation for  $R_i^{(S, \mathbf{x})}$  converges, the values of  $R_i^{(S, \mathbf{x})}$  on both sides of Equation (1), at convergence, would be the same. Hence, bringing all terms containing  $R_i^{(S, \mathbf{x})}$  to one side, we get

$$R_i^{(S, \mathbf{x})} = \frac{\Gamma^{(S, \mathbf{x}^{(S)})} \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}}{D^{(S, \mathbf{x})}} \cdot r - \frac{c_i x_i^{(S)}}{D^{(S, \mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} \quad (2)$$

where  $D^{(S,\mathbf{x})} = (1 - \delta)\Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ .

In order to analyze the proposed stochastic game in depth, we now consider a particular form of  $\Gamma^{(S,\mathbf{x}^{(S)})}$ . Recall that the PoW procedure is the most time and computation intensive component of the mining process. In particular, the time spent for other procedures such as gathering transactions to be included in a block, broadcasting a block post mining, and validating a block mined by other miners, are negligible as compared to that spent for the PoW procedure. As described earlier, the PoW procedure is nothing but repeated computation of hashes and moreover, the computations of different hashes are mutually independent and are completely parallelizable. Hence, the rate at which the PoW procedure runs is proportional to the number of hashes computed per unit time, which in turn, is proportional to the amount of computational power. Since the PoW procedure occupies almost the entire execution time of the whole mining process and its execution rate is proportional to the amount of power invested for it, we have that the rate of the mining process is proportional to the amount of computational power. Note that a similar reasoning would hold for other applications such as volunteer computing if almost the entirety of the concerned task can be parallelized. Hence, for such distributed computing applications, it is reasonable to assume that the rate of problem getting solved is proportional to the total power invested by the investors of computation (i.e., miners in the blockchain setting). We hence, for our detailed analysis, consider  $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)$ , where  $\gamma$  is the rate constant of proportionality determined by the problem being solved. Such a modeling of the problem solving rate can also be found in the literature [32, 35], where it is typically expressed as the total power divided by the problem difficulty (to build a correspondence,  $\gamma$  can be viewed as the reciprocal of problem difficulty).

Considering  $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)$  in Equation (2), player  $i$ 's expected utility as computed in state  $S$  is

$$R_i^{(S,\mathbf{x})} = (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S,\mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S,\mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{D^{(S,\mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} \quad (3)$$

where  $D^{(S,\mathbf{x})} = (1 - \delta)\gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right) + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ .

We now derive a closed-form expression for the expected utility. Let us define a matrix  $\mathbf{W}^{(\mathbf{x})}$  of size  $2^{|U|} \times 2^{|U|}$ , where each row/column corresponds to a state. In what follows, instead of writing  $W^{(\mathbf{x})}(\mathcal{O}(S), \mathcal{O}(S'))$ , we write  $W^{(\mathbf{x})}(S, S')$  since it does not introduce any ambiguity. Let the elements of  $\mathbf{W}^{(\mathbf{x})}$  be:

$$\begin{aligned} \text{for } j \notin S : W^{(\mathbf{x})}(S, S \cup \{j\}) &= \frac{\lambda_j}{D^{(S,\mathbf{x})}} \\ \text{for } j \in S : W^{(\mathbf{x})}(S, S \setminus \{j\}) &= \frac{\mu_j}{D^{(S,\mathbf{x})}} \\ \text{and all other elements of } \mathbf{W}^{(\mathbf{x})} &\text{ are 0} \end{aligned} \quad (4)$$

Since  $\ell > 0$ , we have  $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right) > 0$ . Also,  $\delta < 1$ . So,  $D^{(S,\mathbf{x})} > \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ . Hence,  $\mathbf{W}^{(\mathbf{x})}$  is strictly substochastic.

Let  $\mathbf{Z}_i^{(\mathbf{x})}$  be the vector whose component  $\mathcal{O}(S)$  is  $Z_i^{(S,\mathbf{x})}$ , where

$$Z_i^{(S,\mathbf{x})} = (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S,\mathbf{x})}} \quad (5)$$

It can be easily seen that when  $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)$ , we have  $\mathbf{I} - \mathbf{W}^{(\mathbf{x})} = \mathbf{G}^{(\mathbf{x})}(\mathbf{I} - \mathbf{M}^{(\mathbf{x})})$  and  $\mathbf{Z}_i^{(\mathbf{x})} = \mathbf{G}^{(\mathbf{x})}\mathbf{F}_i^{(\mathbf{x})}$ , where  $\mathbf{G}^{(\mathbf{x})}$  is the diagonal matrix with elements  $G^{(\mathbf{x})}(S, S) = \frac{B^{(S,\mathbf{x})}}{D^{(S,\mathbf{x})}}$ . Now, from the result of Lemma 1, we have  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{M}^{(\mathbf{x})})^{-1}\mathbf{F}_i^{(\mathbf{x})}$ . So, we can now write  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{M}^{(\mathbf{x})})^{-1}(\mathbf{G}^{(\mathbf{x})})^{-1}\mathbf{G}^{(\mathbf{x})}\mathbf{F}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Z}_i^{(\mathbf{x})}$ , which leads to the following proposition providing a closed-form expression for the expected utility.

**Proposition 1.**  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Z}_i^{(\mathbf{x})}$

Owing to the requirement of deriving the inverse of  $\mathbf{I} - \mathbf{W}^{(\mathbf{x})}$ , a general analysis of the concerned stochastic game when considering an arbitrary  $\mathbf{W}^{(\mathbf{x})}$  is intractable. However, for the distributed computing setting and the corresponding utility formulation that we consider in this work, we show that the analysis turns out to be tractable.

### 3. Analysis of Markov Perfect Equilibrium (MPE)

MPE is guaranteed to exist in a finite player game with a finite state space and finite action spaces, if the horizon is either finite, or infinite with the utility function being continuous at infinity [4]. Since our considered game has infinite action spaces in each state, it is not clear whether an MPE exists.

Let  $\hat{R}_i^{(S,\mathbf{x})}$  be the equilibrium utility of player  $i$  in state  $S$ , that is, when  $i$  plays its best response strategy to the equilibrium strategies of the other players  $j \in S \setminus \{i\}$  (while foreseeing effects of its actions on state transitions and resulting utilities). We can determine MPE similar to optimal policy in MDP (using policy-value iterations to reach a fixed point). Here, for maximizing  $\hat{R}_i^{(S,\mathbf{x})}$ , we could assume that we have optimized for other states and use those values to find an optimizing  $\mathbf{x}$  for maximizing  $\hat{R}_i^{(S,\mathbf{x})}$ . Since we have a closed form expression for vector  $\mathbf{R}_i^{(\mathbf{x})}$  in terms of policy  $\mathbf{x}$  (Proposition 1), we could effectively determine the fixed point directly.

Now, from Equation (3), the Bellman equations over states  $S \in 2^U$  for player  $i$  can be written as

$$\hat{R}_i^{(S,\mathbf{x})} = \max_{\mathbf{x}} \left\{ (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S,\mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S,\mathbf{x})}} \cdot \hat{R}_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{D^{(S,\mathbf{x})}} \cdot \hat{R}_i^{(S \setminus \{j\}, \mathbf{x})} \right\}$$

where  $D^{(S,\mathbf{x})} = (1 - \delta)\gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right) + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ .

**Lemma 2.** For any state  $S$  and policy profile  $\mathbf{x}$ , we have  $R_i^{(S,\mathbf{x})} < \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r > c_i$ , and  $R_i^{(S,\mathbf{x})} > \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r < c_i$ .

*Proof.* Let  $V_i^{(S,\mathbf{x}^{(S)})}$  be the expected utility of player  $i$  in state  $S$  computed without considering the arrivals and departures of players ( $\lambda_j = 0, \forall j \notin S$  and  $\mu_j = 0, \forall j \in S$ ). So, we have

$$V_i^{(S,\mathbf{x}^{(S)})} = (\gamma r - c_i) \frac{x_i^{(S)}}{(1 - \delta)\gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)} = \left( r - \frac{c_i}{\gamma} \right) \left( \frac{1}{1 - \delta} \right) \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}$$



Let  $\mathbf{V}_i^{(\mathbf{x})}$  be the vector whose component  $\mathcal{O}(S)$  is  $V_i^{(S, \mathbf{x}^{(S)})}$ . Let  $\mathbf{Z}_i^{(\mathbf{x})} = \mathbf{Y}^{(\mathbf{x})} \mathbf{V}_i^{(\mathbf{x})}$ . Note that when  $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)$ , we have that  $\mathbf{Y}^{(\mathbf{x})}$  is a diagonal matrix, with diagonal elements  $Y^{(\mathbf{x})}(S, S) = \frac{(1-\delta)\gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)}{D^{(S, \mathbf{x})}}$ . From the definition of  $\mathbf{W}^{(\mathbf{x})}$  in Equation (4) and the fact that  $D^{(S, \mathbf{x})} = (1-\delta)\gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right) + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ , we have that  $\mathbf{W}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})}$  is a stochastic matrix (the sum of elements in each of its rows is 1).

Let  $\mathbf{U}^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1}$ , where  $\mathbf{1}$  is the vector whose each element is 1. It is clear that all the elements of  $\mathbf{U}^{(\mathbf{x})}$  are non-negative. We will now show that  $\|\mathbf{U}^{(\mathbf{x})}\|_\infty \leq 1$ , that is, the maximum element of the vector  $\mathbf{U}^{(\mathbf{x})}$  is not more than 1. Let  $u_{S_0}$  be the element with the maximum value (one of the maximum, if there are multiple). Suppose  $u_{S_0}^{(\mathbf{x})} = \|\mathbf{U}^{(\mathbf{x})}\|_\infty > 1$ . So, we would have

$$\begin{aligned} \mathbf{U}^{(\mathbf{x})} &= (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1} \implies \mathbf{U}^{(\mathbf{x})} = \mathbf{W}^{(\mathbf{x})} \mathbf{U}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})} \mathbf{1} \\ \implies u_{S_0}^{(\mathbf{x})} &= \sum_{S \in 2^{\mathcal{U}}} u_S^{(\mathbf{x})} W^{(\mathbf{x})}(S_0, S) + Y^{(\mathbf{x})}(S_0, S_0) \\ \implies u_{S_0}^{(\mathbf{x})} &< u_{S_0}^{(\mathbf{x})} \sum_{S \in 2^{\mathcal{U}}} W^{(\mathbf{x})}(S_0, S) + u_{S_0}^{(\mathbf{x})} Y^{(\mathbf{x})}(S_0, S_0) \\ &\quad (\because \max_S u_S^{(\mathbf{x})} = u_{S_0}^{(\mathbf{x})} > 1) \\ \implies \sum_{S \in 2^{\mathcal{U}}} W^{(\mathbf{x})}(S_0, S) + Y^{(\mathbf{x})}(S_0, S_0) &> 1 \end{aligned}$$

This is a contradiction since  $\mathbf{W}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})}$  is a stochastic matrix. So, we have shown  $\|\mathbf{U}^{(\mathbf{x})}\|_\infty = \|(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1}\|_\infty \leq 1$ . That is,  $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})}$  is substochastic or stochastic. From Proposition 1,  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{V}_i^{(\mathbf{x})}$ . Since  $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})}$  is substochastic or stochastic,  $\mathbf{R}_i^{(S, \mathbf{x})}$  for each  $S$  is a linear combination (with weights summing to less than or equal to 1) of  $V_i^{(S, \mathbf{x}^{(S)})}$  over all  $S \in 2^{\mathcal{U}}$ .

For each  $S$ ,  $V_i^{(S, \mathbf{x}^{(S)})} = \left( r - \frac{c_i}{\gamma} \right) \left( \frac{1}{1-\delta} \right) \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}$ . So,  $V_i^{(S, \mathbf{x}^{(S)})} < \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r > c_i$ , and  $V_i^{(S, \mathbf{x}^{(S)})} > \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r < c_i$ . Since  $\mathbf{R}_i^{(S, \mathbf{x})}$  for each  $S$  is a linear combination (with weights summing to less than or equal to 1) of  $V_i^{(S, \mathbf{x}^{(S)})}$  over all  $S \in 2^{\mathcal{U}}$ , we have  $R_i^{(S, \mathbf{x})} < \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r > c_i$ , and  $R_i^{(S, \mathbf{x})} > \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right)$  if  $\gamma r < c_i$ .  $\square$

**Lemma 3.**  $R_i^{(S, \mathbf{x})}$  is a monotone function of  $x_i^{(S)}$ .

*Proof.* We define the following for simplifying notation.

$$\begin{aligned} A_i^{(S, \mathbf{x})} &= \sum_{j \notin S} \lambda_j \hat{R}_i^{(S \cup \{j, \mathbf{x}\})} + \sum_{j \in S} \mu_j \hat{R}_i^{(S \setminus \{j, \mathbf{x}\})} \\ \text{and } E_i^{(S, \mathbf{x}^{(S)})} &= \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j + (1-\delta)\gamma \left( \sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell \right) \end{aligned}$$

Hence, we can write

$$R_i^{(S, \mathbf{x})} = \frac{A_i^{(S, \mathbf{x})} + (\gamma r - c_i) x_i^{(S)}}{E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}}$$

$$\text{and hence, } \frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}} = \frac{(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S, \mathbf{x})}}{\left( E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)} \right)^2}$$

The denominator is positive, while the numerator is a constant w.r.t.  $x_i^{(S)}$ , since  $A_i^{(S, \mathbf{x})}$  and  $E_i^{(S, \mathbf{x}^{(S)})}$  do not depend on  $x_i^{(S)}$ . So,  $R_i^{(S, \mathbf{x})}$  is a monotone function of  $x_i^{(S)}$ . Whether it is increasing or decreasing, depends on the sign of  $(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S, \mathbf{x})}$ .  $\square$

In what follows, let  $\bar{x}_i$  denote the maximal power available to a player  $i$ .

**Proposition 2.** In MPE, for a state  $S \ni i$ , a player  $i$  invests its maximal power  $\bar{x}_i$  if  $\gamma r > c_i$ , no power if  $\gamma r < c_i$ , and any amount of power in the range  $[0, \bar{x}_i]$  if  $\gamma r = c_i$ .

*Proof.* Let  $W^{(S, \mathbf{x})}$  be the row  $\mathcal{O}(S)$  of  $\mathbf{W}^{(\mathbf{x})}$ . Note that  $A_i^{(S, \mathbf{x})} = (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) W^{(S, \mathbf{x})} \hat{R}_i^{(\mathbf{x})}$ . From the proof of Lemma 3,  $\frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}}$  has same sign as  $(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S, \mathbf{x})}$ , which can be written as:

$$\begin{aligned} &(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S, \mathbf{x})} \\ &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) W^{(S, \mathbf{x})} \hat{R}_i^{(\mathbf{x})} \\ &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) (\hat{R}_i^{(S, \mathbf{x})} - Z_i^{(S, \mathbf{x})}) \\ &\quad (\because \mathbf{W}^{(\mathbf{x})} \mathbf{R}_i^{(\mathbf{x})} = \mathbf{R}_i^{(\mathbf{x})} - \mathbf{Z}_i^{(\mathbf{x})} \text{ from Proposition 1}) \\ &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})} (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) \\ &\quad + (1-\delta)\gamma \frac{(\gamma r - c_i) x_i^{(S)}}{E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}} (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) \\ &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})} (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) + (1-\delta)\gamma (\gamma r - c_i) x_i^{(S)} \\ &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})} E_i^{(S, \mathbf{x}^{(S)})} + (\gamma r - c_i - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})}) (1-\delta)\gamma x_i^{(S)} \\ &= (\gamma r - c_i - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})}) E_i^{(S, \mathbf{x}^{(S)})} + (\gamma r - c_i - (1-\delta)\gamma \hat{R}_i^{(S, \mathbf{x})}) (1-\delta)\gamma x_i^{(S)} \\ &= (1-\delta)\gamma \left( \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right) - \hat{R}_i^{(S, \mathbf{x})} \right) (E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) \end{aligned}$$

Since  $E_i^{(S, \mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}$  and  $(1-\delta)\gamma$  are positive, and  $\left( \frac{1}{1-\delta} \left( r - \frac{c_i}{\gamma} \right) - \hat{R}_i^{(S, \mathbf{x})} \right)$  has the same sign as  $(\gamma r - c_i)$  from Lemma 2, we have that  $\frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}}$  has the same sign as  $(\gamma r - c_i)$ . Also, note that if  $\gamma r = c_i$ , we have  $R_i^{(S, \mathbf{x})} = 0, \forall S \in 2^{\mathcal{U}}$  from Proposition 1 when  $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left( \sum_{j \in S} x_j^{(S)} + \ell \right)$ .

So, in any state  $S \ni i$ , it is a dominant strategy for a player  $i$  to invest  $\bar{x}_i$  if  $\gamma r > c_i$ , no power if  $\gamma r < c_i$ , and any amount of power in the range  $[0, \bar{x}_i]$  if  $\gamma r = c_i$ . This provides a consistent solution to the Bellman equations.  $\square$

Thus, the MPE strategy of a player follows a threshold policy, with a threshold on its cost parameter  $c_i$  (whether it is lower than  $\gamma r$ ) or alternatively, a threshold on the offered reward  $r$

(whether it is higher than  $\frac{c_i}{\gamma}$ ). Note the difference from the result in [31, 32] where time is not explicitly considered as described earlier; a player's Nash equilibrium strategy therein follows a smooth policy depending on how its cost parameter compares with the other players' cost parameters.

Since a player's MPE strategy does not depend on  $S$  and the other players' parameters (namely, cost parameters and arrival/departure rates), the assumption of state knowledge as well as knowledge of other players' parameters can be relaxed. It is key to note that the MPE following a threshold policy profile is a very interesting result and is obtained while considering a form of  $\Gamma^{(S, \mathbf{x}^{(S)})}$  (namely,  $\gamma(\sum_{j \in S} x_j^{(S)} + \ell)$ ), which is applicable to most distributed computing settings including blockchain mining. In particular, the studied form of  $\Gamma^{(S, \mathbf{x}^{(S)})}$  results in the cancellation of the factor that captures the total amount of power ( $\sum_{j \in S} x_j^{(S)} + \ell$ ) in Equation (2), thus simplifying it to Equation (3), and eventually leading to the obtained threshold policy as MPE. The results will be inevitably very different for other forms of  $\Gamma^{(S, \mathbf{x}^{(S)})}$ .

We now provide an intuition for why the MPE strategies are independent of the arrival and departure rates. From Proposition 1,  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})}$ . For  $\gamma r > c_i$ , when power  $x_i^{(S)}$  increases,  $\mathbf{Z}_i^{(\mathbf{x})}$  increases (in particular, elements  $Z_i^{(S, \mathbf{x})}$  for  $S \ni i$ ) and the values of elements in  $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$  decrease (in particular, the non-zero elements in rows corresponding to states  $S \ni i$ ). But  $\mathbf{R}_i^{(\mathbf{x})}$  increases with  $x_i^{(S)}$  when  $\gamma r > c_i$  (shown in the proof of Proposition 2), implying that the rate of increase of  $\mathbf{Z}_i^{(\mathbf{x})}$  dominates the rate of decrease of the elements in  $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$ . So, the effect of  $\mathbf{W}^{(\mathbf{x})}$  and hence the state transitions is relatively weak, thus resulting in Markovian players playing strategies that are independent of the arrival and departure rates. Similar argument holds for  $\gamma r \leq c_i$ . It would be interesting to study scenarios where the rate of problem getting solved is a non-linear function of the players' invested powers. While a linear function is suited to most distributed computing applications, a non-linear function could possibly see  $\mathbf{W}^{(\mathbf{x})}$  having a strong effect leading to MPE being dependent on the arrival/departure rates.

To provide an intuition for the different investment strategies of a player  $i$  based on the relation between  $\gamma r$  and  $c_i$ , consider a simplistic scenario where players do not arrive or depart over time. If such a system is in state  $S$ , the expected reward obtained by player  $i$  for a block is proportional to its invested power, i.e.,  $\frac{x_i^{(S)}}{\sum_j x_j^{(S)} + \ell} r$ . Also, as the rate of problem getting solved is  $\gamma(\sum_{j \in S} x_j^{(S)} + \ell)$ , the cost incurred by the player is  $\frac{c_i}{\gamma(\sum_{j \in S} x_j^{(S)} + \ell)} x_i^{(S)}$ . Considering also discounted utilities due to future blocks, player  $i$ 's expected utility as computed in state  $S$  is  $\left( \frac{x_i^{(S)}}{\sum_j x_j^{(S)} + \ell} r - \frac{c_i}{\gamma(\sum_{j \in S} x_j^{(S)} + \ell)} x_i^{(S)} \right) \sum_{\alpha=0}^{\infty} \delta^\alpha = (\gamma r - c_i) \frac{x_i^{(S)}}{(1-\delta)\gamma(\sum_{j \in S} x_j^{(S)} + \ell)}$ . This expression can also be obtained by considering  $\lambda_j = \mu_j = 0, \forall j$ , in Equation (3). Recall that  $x_i^{(S)} = 0$  if  $i \notin S$ . Since  $\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}$  is monotonically increasing in  $x_i^{(S)}$  if  $i \in S$ , it is optimal for  $x_i^{(S)}$  to be  $\bar{x}_i$  if  $\gamma r > c_i$ , zero if  $\gamma r < c_i$ , and any value if  $\gamma r = c_i$ . This applies to all states  $S \ni i$ , which pro-

vides an intuition for the case-based investment strategies in Proposition 2, albeit this intuition ignores transitions to other states. For an intuition while accounting for transitions to other states, observe  $Z_i^{(S, \mathbf{x})}$  (Equation (5)) which can be thought of as the utility computed in state  $S$  before any player arrives or departs (here,  $\lambda_j, \mu_j \neq 0$ ). Each element of  $\mathbf{Z}_i^{(\mathbf{x})}$  is written as the product of  $\gamma r - c_i$  and a non-negative quantity. Now, from Proposition 1 and the fact that  $\mathbf{W}^{(\mathbf{x})}$  is strictly substochastic,  $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})} = \sum_{\eta=0}^{\infty} (\mathbf{W}^{(\mathbf{x})})^\eta \mathbf{Z}_i^{(\mathbf{x})}$ . As the elements of  $\mathbf{W}^{(\mathbf{x})}$  are non-negative, each element of  $\mathbf{R}_i^{(\mathbf{x})}$  can also be expressed as the product of  $\gamma r - c_i$  and a non-negative quantity that depends on  $\mathbf{x}$ . Whether or not this quantity is increasing in  $x_i^{(S)}$ , for a given  $S \ni i$ , is not readily clear. However, as one of the intuitions provided earlier, with an increase in  $x_i^{(S)}$  when  $\gamma r > c_i$ , the rate of increase of  $\mathbf{Z}_i^{(\mathbf{x})}$  dominates the rate of decrease of the elements in  $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$ , and so the said non-negative quantity is increasing. Hence, given  $\gamma r > c_i$ , when  $x_i^{(S)} = \bar{x}_i$ , each of the said quantities is maximized, thus maximizing the elements of  $\mathbf{R}_i^{(\mathbf{x})}$ . If  $\gamma r < c_i$ , the said non-negative quantity should be zero since it would be multiplied by  $\gamma r - c_i < 0$ . This is achieved when  $x_i^{(S)} = 0$ , owing to which  $\mathbf{Z}_i^{(\mathbf{x})} = \mathbf{0}$ , and hence  $\mathbf{R}_i^{(\mathbf{x})} = \mathbf{0}$ . If  $\gamma r = c_i$ , we have  $\mathbf{Z}_i^{(\mathbf{x})} = \mathbf{0}$ , leading to  $\mathbf{R}_i^{(\mathbf{x})} = \mathbf{0}$ , irrespective of the value of  $x_i^{(S)}$ .

#### 4. Analysis of Players' Expected Utilities

While we showed the players' MPE strategies to be independent of their arrival and departure rates, it is clear from Equations (2), (3) and Proposition 1 that the players' utilities would depend on these rates. We now, in this section, build a foundation for analyzing the players' utilities as a function of the arrival and departure rates, and in the next section, study the effects of these rates and the other system parameters on the utilities in MPE elaborately. As can be seen from Proposition 1, computation of expected utilities involves the inverse of  $\mathbf{I} - \mathbf{W}^{(\mathbf{x})}$ , which is arguably infeasible to obtain analytically, in general. With the aim of simplifying, we consider the (strategic) players to be homogeneous (that is, the arrival/departure rates and cost parameters corresponding to all players to be equal); let  $\lambda, \mu, c$  denote the common arrival rate, departure rate, and cost parameter, respectively. Now, the players' sets (states) can be mapped to their cardinalities from the system's perspective. From the perspective of a particular player  $i$ , the players' sets can be mapped to their cardinalities while also capturing whether they contain player  $i$ . That is, the state space comprises the empty set (which clearly does not contain player  $i$ ), the universal set (which clearly contains player  $i$ ), and two states each for all the other  $|\mathcal{U}| - 1$  cardinalities (the two states capturing whether or not the set contains player  $i$ ). This leads to a total of  $2^{|\mathcal{U}|}$  states, an exponential reduction from the general case with  $2^{|\mathcal{U}|}$  states. In order to reliably and accurately study the effects of the arrival and departure rates, we consider that the computation is dominated by the strategic players (that is, the amount of fixed power is insignificant:  $\ell \rightarrow 0$ ); we will study the effect of  $\ell$  subsequently.

Note that if  $c > \gamma r$ , no player would invest (from Proposition 2). Also, if  $c = \gamma r$ , each player's expected utility computed in any state would be zero. Hence, in what follows, we consider that the values of parameters  $\gamma, r, c$  are such that  $c < \gamma r$ . Considering that the maximum power available to all players is the same, say  $\bar{x}$ , the MPE policy profile  $\mathbf{x}$  is such that each player invests  $\bar{x}$  in each state in which it is present.

Before diving into the expected utilities when there are transitions among the different states, let us understand the expected utility of a player  $i$  in the homogeneous case when there are no state transitions (i.e.,  $\lambda = \mu = 0$ ) and  $\ell \rightarrow 0$ . It can be seen that for the state corresponding to the players' set having cardinality  $s$  and containing player  $i$ , the expected utility of player  $i$  given by Equation (3) simplifies to  $\frac{(\gamma r - c)\bar{x}}{(1 - \delta)\gamma s \bar{x}} = \frac{1}{1 - \delta} \left( r - \frac{c}{\gamma s} \right)$ , implying that it is inversely proportional to the number of players in that state. This is intuitive, since in the homogeneous case, the reward for mining a block would be won by the players that are present in the system with equal probability (hence the term  $\frac{r}{s}$ ), and the cost is reduced owing to the reduced time due to the combined rate of the problem getting solved (hence the term  $\frac{c}{\gamma s}$ ). Also, owing to the perceived utility being discounted by a factor of  $\delta \in [0, 1]$  for every future block, the computed expected utility is  $\left( r - \frac{c}{\gamma s} \right) \sum_{\alpha=0}^{\infty} \delta^\alpha$ . On the other hand, for the state corresponding to the players' set of cardinality  $s$  and not containing player  $i$ , the expected utility of player  $i$  is zero. Now that we have an understanding of the expected utility without state transitions, we proceed to analyzing the expected utility for the general homogeneous case in the presence of state transitions. In what follows, we consider  $\delta = 0$ , that is, players consider the expected utility corresponding to only the current block. The results for other values of  $\delta \in (0, 1)$  are just scaled versions of the results for  $\delta = 0$  and are qualitatively very similar. When  $\delta = 0$ , the expected utility of a player  $i$  if the problem gets solved when there are  $s$  players present in the system without transiting to another state, is

$$\begin{aligned} & \frac{1}{s} \left( r - \frac{c}{\gamma} \right), \text{ if player } i \text{ is present} \\ & \text{and } 0, \text{ if player } i \text{ is absent} \end{aligned} \quad (6)$$

We highlight the above expression as we will recall it when we study a mean field approximation.

#### 4.1. Types of Expected Utilities and their Formulation

In principle, a player can compute its expected utility in each system state (recall that a state captures the total number of players present and whether or not the given player is present). It is, however, impractical for a player to exactly know the state that the system is in. Nevertheless, a player can compute three types of expected utilities, namely, (a) the conditional expected utility given that it is present when the utility is being computed, denoted by  $R_{\ni}$ , (b) the conditional expected utility given that it is absent when the utility is being computed, denoted by  $R_{\not\ni}$ , and (c) the overall expected utility without having to know whether or not it is present when the utility is being computed, denoted by  $\langle R \rangle$ . Note that we drop player  $i$ 's specification in the notation since it does not introduce ambiguity in the homogeneous

case. We now proceed to formulating these different types of expected utilities. In what follows, let  $N = |\mathcal{U}|$ .

Let  $R_{\ni}^s$  be the expected utility as computed by the given player when in the state that corresponds to the player being present and a total of  $s$  players being present. Similarly, let  $R_{\not\ni}^s$  be the expected utility as computed when the player is absent and a total of  $s$  players are present. We can now write Equation (3) for two types of states given the number of players present, namely, that containing the given player and that not containing the given player. In order to segregate the terms based on whether or not the given player  $i$  is present, the last summation in Equation (3), namely,  $\sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} R_i^{(S \setminus \{j\}, \mathbf{x})}$ , can be written as  $\sum_{j \in S \setminus \{i\}} \frac{\mu_j}{D^{(S, \mathbf{x})}} R_i^{(S \setminus \{j\}, \mathbf{x})} + \frac{\mu_i}{D^{(S, \mathbf{x})}} R_i^{(S \setminus \{i\}, \mathbf{x})}$ , for  $S \ni i$ . That is, Equation (3) can be rewritten as

$$R_i^{(S, \mathbf{x})} = (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S, \mathbf{x})}} + \sum_{j \in S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S \setminus \{i\}} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} + \frac{\mu_i}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{i\}, \mathbf{x})}$$

Hence, for computing a player's utility  $R_{\ni}^s$  in the state that consists of  $s$  players and contains the given player (where  $s \in \{1, \dots, N\}$ ), the following events resulting from arrivals/departures of players need to be accounted for: (a) any of the  $N - s$  players which are not in the system, can arrive with rate  $\lambda$  each, thus transiting the system to state  $R_{\ni}^{s+1}$  consisting of  $s + 1$  players and containing the given player, (b) any of the  $s - 1$  players which are present in the system excluding the given player, can depart with rate  $\mu$  each, thus transiting the system to state  $R_{\ni}^{s-1}$  consisting of  $s - 1$  players and containing the given player, (c) the given player can depart with rate  $\mu$ , thus transiting the system to state  $R_{\not\ni}^{s-1}$  consisting of  $s - 1$  players and not containing the given player. So, we have

$$R_{\ni}^s = (\gamma r - c) \frac{\bar{x}}{D^s} + \frac{(N - s)\lambda}{D^s} R_{\ni}^{s+1} + \frac{(s - 1)\mu}{D^s} R_{\ni}^{s-1} + \frac{\mu}{D^s} R_{\not\ni}^{s-1} \quad (7)$$

where  $D^s = \gamma s \bar{x} + (N - s)\lambda + s\mu$ . Note that for  $s = N$ , the second term vanishes.

For states that do not contain the given player, the first term disappears because the given player is not a part of the system and hence does not invest. On similar lines as above, for the state that consists of  $s$  players and does not contain the given player (where  $s \in \{0, \dots, N - 1\}$ ), Equation (3) transforms into

$$R_{\not\ni}^s = \frac{(N - s - 1)\lambda}{D^s} R_{\not\ni}^{s+1} + \frac{\lambda}{D^s} R_{\ni}^{s+1} + \frac{s\mu}{D^s} R_{\not\ni}^{s-1} \quad (8)$$

Note that for  $s = 0$ , the last term vanishes.

Now, for formulating the different types of expected utilities introduced earlier (namely,  $R_{\ni}$ ,  $R_{\not\ni}$  and  $\langle R \rangle$ ), the minimal information that the given player should have is of the probability distribution over the number of players that are present. This distribution can be obtained by exploiting the fact that the underlying stochastic arrival and departure process here resembles an *Engset's* system [43]. Given the homogeneous player arrival rate  $\lambda$ , departure rate  $\mu$ , and population size  $N$ , let  $\mathbb{P}_{\lambda, \mu}^N(s)$  be the

probability that the number of players present in the system is  $s$ . It is known that

$$\mathbb{P}_{\lambda,\mu}^N(s) = \binom{N}{s} \left( \frac{\lambda}{\lambda+\mu} \right)^s \left( \frac{\mu}{\lambda+\mu} \right)^{N-s}$$

It is also known that the probability of a player being present (or the fraction of time during which a player is present) is  $\frac{\lambda}{\lambda+\mu}$ , and the expected number of players present is  $\frac{\lambda}{\lambda+\mu}N$ . The mean duration of a full cycle of the stochastic process is  $\frac{1}{\lambda} + \frac{1}{\mu}$ .

In absence of additional information, since all players are equally likely to be present or absent, we have that if the number of players present is  $s$ , a given player is present with probability  $\frac{s}{N}$  and absent with probability  $\frac{N-s}{N}$ . So, the probability that a total of  $s$  players are present, with the given player present is  $\mathbb{P}_{\lambda,\mu}^N(s) \frac{s}{N}$ , and that with the player absent is  $\mathbb{P}_{\lambda,\mu}^N(s) \frac{N-s}{N}$ . Clearly, the overall probability of the player being present is  $\sum_s \mathbb{P}_{\lambda,\mu}^N(s) \frac{s}{N}$ , and that of being absent is  $\sum_s \mathbb{P}_{\lambda,\mu}^N(s) \frac{N-s}{N}$ . Hence, we get

$$R_{\exists} = \frac{\sum_s R_{\exists}^s \mathbb{P}_{\lambda,\mu}^N(s) \frac{s}{N}}{\sum_s \mathbb{P}_{\lambda,\mu}^N(s) \frac{s}{N}} \quad (9)$$

$$R_{\nexists} = \frac{\sum_s R_{\nexists}^s \mathbb{P}_{\lambda,\mu}^N(s) \frac{N-s}{N}}{\sum_s \mathbb{P}_{\lambda,\mu}^N(s) \frac{N-s}{N}} \quad (10)$$

$$\langle R \rangle = \frac{\lambda}{\lambda+\mu} R_{\exists} + \frac{\mu}{\lambda+\mu} R_{\nexists} \quad (11)$$

Since Equation (11) is not easy to analyze or get insights into, we present an alternative derivation for  $\langle R \rangle$  that would aid our understanding of the overall expected utility. Let  $\langle R^s \rangle$  denote the expected utility of the given player computed when the number of players present is  $s$ . As earlier, if the number of players present is  $s$ , a given player is present with probability  $\frac{s}{N}$  and absent with probability  $\frac{N-s}{N}$ . On similar lines as Equation (3), we can write  $\langle R^s \rangle$  recursively in terms of  $\langle R^{s+1} \rangle$  and  $\langle R^{s-1} \rangle$  as

$$\langle R^s \rangle = \frac{s}{N} \frac{(\gamma r - c)\bar{x}}{D^s} + \frac{(N-s)\lambda}{D^s} \langle R^{s+1} \rangle + \frac{s\mu}{D^s} \langle R^{s-1} \rangle \quad (12)$$

where  $D^s = \gamma s\bar{x} + (N-s)\lambda + s\mu$ . Note that for  $s = N$ , the second term vanishes while for  $s = 0$ , the last term vanishes. Multiplying the above equation throughout by  $D^s$ , we get

$$\gamma s\bar{x}\langle R^s \rangle + (N-s)\lambda\langle R^s \rangle + s\mu\langle R^s \rangle = \frac{s}{N}(\gamma r - c)\bar{x} + (N-s)\lambda\langle R^{s+1} \rangle + s\mu\langle R^{s-1} \rangle$$

Now,  $\langle R^s \rangle = \langle R^{s+1} \rangle = \langle R^{s-1} \rangle = \frac{1}{N} \left( r - \frac{c}{\gamma} \right)$  satisfies the above equation. Owing to uniqueness of the solution and since  $\frac{1}{N} \left( r - \frac{c}{\gamma} \right)$  is independent of  $s$ , we have that

$$\forall s : \langle R^s \rangle = \frac{1}{N} \left( r - \frac{c}{\gamma} \right) \quad (13)$$

So, the expected utility of the player is a constant with respect to the number of players present when the utility is being computed. Hence, the overall expected utility is

$$\langle R \rangle = \frac{1}{N} \left( r - \frac{c}{\gamma} \right) \quad (14)$$

Note that for obtaining the above closed-form expression for  $\langle R \rangle$ , it is critical that the maximum powers available to all players are equal and the amount of fixed power  $\ell \rightarrow 0$ . For instance, if  $\ell \rightarrow 0$ , we would have  $D^s = \gamma(s\bar{x} + \ell) + (N-s)\lambda + s\mu$  and so, a simplified expression cannot be readily obtained. Now, with the aim of obtaining simplified expressions for  $R_{\exists}$  and  $R_{\nexists}$  as well, albeit approximate, we take a mean field approach.

#### 4.2. Mean Field Approximation

Obtaining general closed-form expressions for  $R_{\exists}$  and  $R_{\nexists}$  that could be analyzed theoretically, are arguably intractable, as can be seen from Equations (9), (10) and the terms involved therein (namely, Equations (7) and (8)). So, we proceed to obtain closed-form expressions that can approximate their values with reasonable accuracy, by way of a mean field approach. In this approach, instead of the  $2N$  states that captured the number of players present and whether or not the given player is present in the system, we consider only two states, namely, (a)  $S_{\exists}$  that corresponds to the given player being present and (b)  $S_{\nexists}$  that corresponds to the given player being absent. So, from the given player's perspective, when it arrives, the system transits from state  $S_{\nexists}$  to  $S_{\exists}$ , while when it departs, the system transits from state  $S_{\exists}$  to  $S_{\nexists}$ . The mean field approximation will consider that the system invariably consists of an average number of players (say  $n$ ), and the arrival/departure of the given player does not affect this average number and so, also does not affect the rate of problem getting solved. In particular, irrespective of whether we are in state  $S_{\exists}$  or  $S_{\nexists}$ , the number of players present in the system is considered to be a constant,  $n = \frac{\lambda}{\lambda+\mu}N$ , which is the expected number of players present in an Engset's system, and hence the rate of problem getting solved is consistently  $\gamma n\bar{x}$ .

Recall from Equation (6) that the expected utility of a player if the problem gets solved when there are  $s$  players present in the system without transiting to another state, is  $\frac{1}{s} \left( r - \frac{c}{\gamma} \right)$  if the player is present, and 0 if the player is absent when the utility is being computed. Since in the mean field approach, we consider the number of players being present in the system to be  $n = \frac{\lambda}{\lambda+\mu}N$ , we have that the given player would get a utility of  $\frac{1}{n} \left( r - \frac{c}{\gamma} \right)$  if the problem gets solved while in state  $S_{\exists}$ , and a utility of 0 if the problem gets solved while in state  $S_{\nexists}$ .

For computing the expected utility in state  $S_{\exists}$ , which is nothing but  $R_{\exists}$ , the following events need to be accounted for: (a) the player can depart with rate  $\mu$ , thus transiting to state  $S_{\nexists}$ , wherein the expected utility computed would be  $R_{\nexists}$ , and (b) the problem can get completely solved with rate  $\gamma n\bar{x}$ , in which case the given player obtains a utility of  $\frac{1}{n} \left( r - \frac{c}{\gamma} \right)$ . On the other hand, for computing  $R_{\nexists}$ , the expected utility in state  $S_{\nexists}$ , the following events need to be accounted for: (a) the player can arrive with

rate  $\lambda$ , thus transiting to state  $S_\ominus$ , wherein the expected utility computed would be  $R_\ominus$ , and (b) the problem can get completely solved with rate  $\gamma n\bar{x}$ , in which case the given player obtains a utility of 0 since the player is not present in the system. So, we have

$$\begin{aligned} R_\ominus &= \frac{\mu}{\mu + \gamma n\bar{x}} \cdot R_\oplus + \frac{\gamma n\bar{x}}{\mu + \gamma n\bar{x}} \cdot \frac{1}{n} \left( r - \frac{c}{\gamma} \right) \\ R_\oplus &= \frac{\lambda}{\lambda + \gamma n\bar{x}} \cdot R_\ominus + \frac{\gamma n\bar{x}}{\lambda + \gamma n\bar{x}} \cdot 0 \end{aligned} \quad (15)$$

where  $n = \frac{\lambda}{\lambda + \mu} N$ . Solving the above two equations, we get the following closed-form expressions:

$$R_\ominus = \frac{1}{N} \left( r - \frac{c}{\gamma} \right) \left( 1 + \frac{\mu \gamma N \bar{x}}{(\lambda + \mu)^2 + \lambda \gamma N \bar{x}} \right) \quad (16)$$

$$R_\oplus = \frac{1}{N} \left( r - \frac{c}{\gamma} \right) \left( 1 - \frac{\lambda \gamma N \bar{x}}{(\lambda + \mu)^2 + \lambda \gamma N \bar{x}} \right) \quad (17)$$

It can be easily verified that  $\langle R \rangle = \frac{\lambda}{\lambda + \mu} R_\ominus + \frac{\mu}{\lambda + \mu} R_\oplus = \frac{1}{N} \left( r - \frac{c}{\gamma} \right)$ . It is important to note that while the above equations provide closed-form expressions and are far easier and faster to compute and analyze, these are essentially approximations to the actual values of  $R_\ominus$  and  $R_\oplus$ . Larger values of  $N$  would likely result in better approximations, owing to the underlying assumption that the average number of players present and the rate of problem getting solved would not be affected by the arrival/departure of the given player. For small values of  $N$ , for instance when  $N = 1$ , the exact expressions give  $R_\ominus = R_\oplus = r - \frac{c}{\gamma}$ , while the mean field approximation gives  $R_\ominus > r - \frac{c}{\gamma}$  and  $R_\oplus < r - \frac{c}{\gamma}$ . So, the mean field approximate expressions should not be viewed as a replacement to the actual expressions, but rather an aid to intuitively understand how  $R_\ominus$  and  $R_\oplus$  are sensitive towards the values of the different parameters.

## 5. Sensitivity Analysis: A Simulation Study

With the foundation for analyzing the players' utilities in place, we now conduct a simulation study for investigating the effects of the arrival and departure rates as well as other system parameters.

### 5.1. Parameters' Values for Simulations

We take cues from Bitcoin mining for the parametric values to be used in our numerical simulations. The currently offered reward for successfully mining a block is 6.25 Bitcoins. Assuming 1 Bitcoin  $\approx$  \$48,000, the reward translates to  $\$3 \times 10^5$ . The Bitcoin problem complexity is set such that it takes around 10 minutes on average for a block to get mined. That is, the rate of problem getting solved is 6 hour<sup>-1</sup> on average. One of the most powerful ASIC (application-specific integrated circuit) currently available in market is Antminer S9, which performs computations of up to 13 TeraHashes per sec; let us call this amount of computational power as 1 unit, without loss of generality. So, we consider the maximum power available to each player to be  $\bar{x} = 1$  unit. As per BitNode (bitnodes.earn.com), a crawler developed to estimate the size of Bitcoin

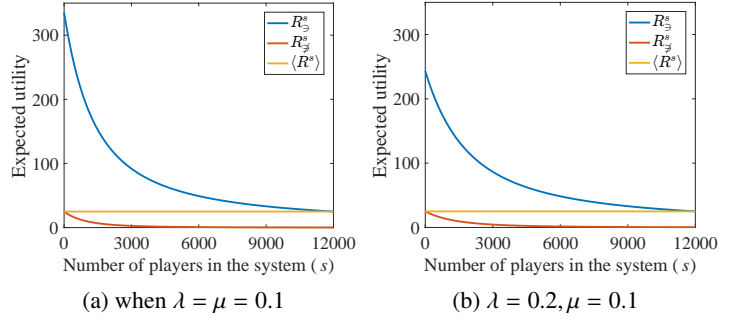


Figure 1: Expected utility of a player as computed when the system contains a certain number of players

network, the number of Bitcoin miners is around  $1.2 \times 10^4$ , providing a realistic estimate for  $N$  (i.e.,  $|\mathcal{U}|$ ). Considering a preliminary scenario wherein  $\lambda = \mu$ , we would have the average number of players present in the system, say  $\hat{N}$ , to be  $\frac{\lambda}{\lambda + \mu} N = \frac{N}{2} = 6 \times 10^3$ . Hence, we should have that  $\gamma \bar{x} \hat{N} = 6$  hour<sup>-1</sup>. As we consider  $\bar{x} = 1$  and  $\hat{N} = 6 \times 10^3$ , we should have  $\gamma = 10^{-3}$  hour<sup>-1</sup>. The aforementioned ASIC consumes about 1.5 kWh in 1 hour, which translates to  $\$0.18$  hour<sup>-1</sup> (at the rate of  $\$0.12$  per kWh). Hence, we consider  $r = 3 \times 10^5, \gamma = 10^{-3}, c = 0.18, \bar{x} = 1, N = |\mathcal{U}| = 1.2 \times 10^4$ . Note that for these values of parameters, we have that the overall expected utility  $\langle R \rangle = \frac{1}{N} \left( r - \frac{c}{\gamma} \right) = 24.985$ .

### 5.2. Effect of Arrival and Departure Rates on Players' Utilities

We now explicitly study the effects of the critical stochastic components, namely, the players' arrival and departure rates, while considering the values of other parameters to be as described earlier ( $r = 3 \times 10^5, \gamma = 10^{-3}, c = 0.18, \bar{x} = 1, N = 1.2 \times 10^4$ ). In particular, we observe how the expected utilities  $R_\ominus$  and  $R_\oplus$  change with respect to the values of  $\lambda$  and  $\mu$ .

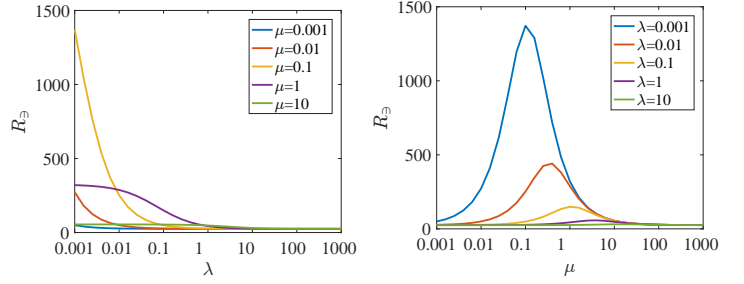
Before proceeding, let us obtain a fundamental understanding of how a player's expected utility is influenced by the number of players present in the system ( $s$ ) at the time of its utility being computed, which we explain with the help of Figure 1. It can be seen that the values of  $R_\ominus^s$  (where  $s \in \{1, \dots, N\}$ ) and  $R_\oplus^s$  (where  $s \in \{0, \dots, N-1\}$ ) decrease as  $s$  increases; this is owing to an increase in competition in the system as the number of players increases. In particular, for a higher value of  $s$ , we have that  $R_\ominus^s$  is lower owing to a higher competition when the given player is present in the system, and  $R_\oplus^s$  is lower owing to a higher competition if and when the given player arrives. Note also that irrespective of the number of players present in the system, if  $c \leq \gamma r$ , it is at least as advantageous for a player to be present in the system and investing than being absent. So, the utility computed when the given player is present is at least as high as that computed when the player is absent, which is why, the plot shows  $\forall s : R_\ominus^s \geq R_\oplus^s$ . As discussed earlier, if the number of players present in the system is  $s$ , a given player is present with probability  $\frac{s}{N}$  and absent with probability  $\frac{N-s}{N}$  and so, we can write  $\langle R^s \rangle = \frac{s}{N} R_\ominus^s + \frac{N-s}{N} R_\oplus^s$ . Clearly, a higher value of  $s$  results in a higher weightage to  $R_\ominus^s$  and hence, the graph of  $\langle R^s \rangle$  approaching that of  $R_\ominus^s$ . Figure 1 validates numerically that the value of  $\langle R^s \rangle$  should be the same for all values of  $s$

(as shown in Equation (13)). Furthermore, if the given player is present in the system, a higher  $\lambda$  results in a higher probability of more players arriving before the problem gets solved, and hence higher level of competition (if departure rate  $\mu$  is unchanged). So, for all values of  $s$ , the values of  $R_{\exists}^s$  are lower for a higher  $\lambda$ . This reasons why the graph of  $R_{\exists}^s$  is lower when  $\lambda = 0.2, \mu = 0.1$  as compared to when  $\lambda = \mu = 0.1$ .

Now that we have an understanding of how  $R_{\exists}^s$  and  $R_{\not\exists}^s$  are influenced by the number of players present, let us continue to studying the effect of  $\lambda$  and  $\mu$  on  $R_{\exists}$  and  $R_{\not\exists}$ . Note that these could be illustrated by way of 3D plots, while showing  $R_{\exists}$  and  $R_{\not\exists}$  as functions of  $\lambda$  and  $\mu$ ; however we present two plots for better clarity and understanding. In particular, we observe how  $R_{\exists}$  changes as a function of  $\lambda$  for different values of  $\mu$ , and as a function of  $\mu$  for different values of  $\lambda$ ; we do the same for  $R_{\not\exists}$ . In order to consider a wide range of values, our plots have a logarithmic X-axis. For instance, in Figure 2(a), while plotting the change in  $R_{\exists}$  as a function of  $\lambda$ , we consider the values of  $\lambda$  to be powers of 10 with the powers ranging from  $-3$  to  $3$  with increments of  $0.2$ . While values more than  $10$  (i.e., arriving at the rate of more than  $10$  times per hour, if not in the system) may seem impractical, they are interesting for understanding the limiting behavior. For this particular plot, the different values of  $\mu$  that we consider are integral powers of  $10$  with the values ranging from  $10^{-3}$  to  $10$ . It is worth highlighting that the mean field approach described earlier, for approximating the values of  $R_{\exists}$  and  $R_{\not\exists}$ , gives a root mean squared relative error of less than  $0.05\%$  and a mean absolute relative error of less than  $0.01\%$ .

As explained in the context of Figure 1, we have  $\forall s : R_{\exists}^s \geq R_{\not\exists}^s$  and so, the expected utility computed when the player is present in the system ( $R_{\exists}$ ) is always at least as high as that computed when the player is absent ( $R_{\not\exists}$ ) for any given set of parameters' values. Also, if all the parameters other than  $\lambda$  and  $\mu$  are unchanged, the overall expected utility  $\langle R \rangle$  (which is a weighted average of  $R_{\exists}$  and  $R_{\not\exists}$ ) stays unchanged and its value is  $\frac{1}{N}(r - \frac{c}{\gamma})$ . Since  $R_{\exists} \geq R_{\not\exists}$  for any given values of  $\lambda, \mu$  and the weighted average of  $R_{\exists}$  and  $R_{\not\exists}$  is a constant across different values of  $\lambda, \mu$ , we have that  $R_{\exists} \geq \langle R \rangle = \frac{1}{N}(r - \frac{c}{\gamma}) \geq R_{\not\exists}$  across different values of  $\lambda, \mu$ . Note that the best possible utility  $R_{\exists}^s$  that the given player can obtain is  $r - \frac{c}{\gamma}$ , which is when the player is the only one present in the system (i.e., no competition) throughout the problem-solving process. So, we have from Equation (9) that  $R_{\exists} \leq r - \frac{c}{\gamma}$ . Also, it is clear that  $R_{\not\exists} \geq 0$  when  $c < \gamma r$ . So, we have that  $0 \leq R_{\not\exists} \leq \langle R \rangle \leq R_{\exists} \leq r - \frac{c}{\gamma}$ , across different values of  $\lambda, \mu$ . This is what we also observe in all the plots (Figures 2 and 3).

Figure 2 illustrates the effect of arrival and departure rates on the given player's expected utility given that the player is present at the time of computing its utility. Figure 2(a) shows the change in  $R_{\exists}$  with respect to  $\lambda$  for different values of  $\mu$ . It can be seen that  $R_{\exists}$  decreases monotonically with  $\lambda$ , and its decrease with respect to the logarithm of  $\lambda$  is a reverse sigmoid-like function (for very low values of  $\mu$ , the lower end of the graph is outside the range of  $\lambda$  shown). To get an intuition, note that for a particular value of  $\mu$ , if  $\lambda$  increases, the probability of more players being present in the system increases, thus lead-



(a) w.r.t.  $\lambda$  for different values of  $\mu$  (b) w.r.t.  $\mu$  for different values of  $\lambda$

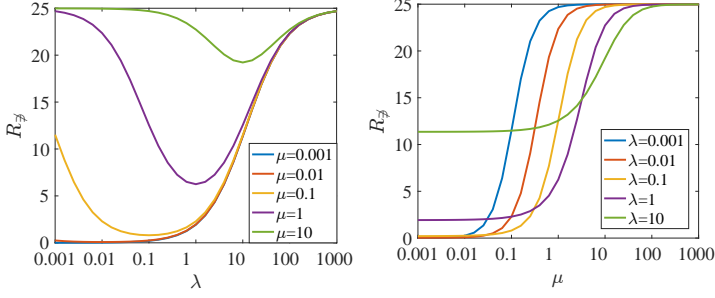
Figure 2: Effect of arrival and departure rates on a player's expected utility if the player is present in the system when the utility is being computed

ing to a higher competition and hence a reduced value of  $R_{\exists}$ . Figure 2(b) shows the change in  $R_{\exists}$  with respect to  $\mu$  for different values of  $\lambda$ . The initial increase and subsequent decrease in the utility as  $\mu$  increases, exhibits a bell-shaped-like nature with respect to the logarithm of  $\mu$ . For any particular  $\lambda$ , if  $\mu$  increases, it is with a higher probability that less players are present, thus leading to a lower competition and an increase in  $R_{\exists}$ . However, if  $\mu$  increases beyond a certain threshold, it results in a higher probability of the given player departing and staying out of the system when the problem gets solved, thus resulting in a decrease in  $R_{\exists}$ . We can estimate the peak of the graph by equating the derivative of  $R_{\exists}$  with respect to  $\mu$  to zero, using its approximate expression from Equation (16); we get

$$\frac{dR_{\exists}}{d\mu} = 0 \implies \mu - \lambda = \gamma \bar{x} \frac{\lambda}{\lambda + \mu} N \implies \mu = \sqrt{\lambda(\lambda + \gamma N \bar{x})}$$

We now provide an intuition for why the critical point is when  $\mu - \lambda = \gamma \bar{x} n$ , where  $n = \frac{\lambda}{\lambda + \mu} N$  is the average number of players present. Note that if the given player is present, it is advantageous if the problem gets solved faster, in particular, before the player departs, and also if it arrives soon after in case it departs. That is, it is advantageous to have the effective rate of departure ( $\mu - \lambda$ ) to be lower than the average rate of problem getting solved ( $\gamma \bar{x} n$ ). Note that for a lower  $\lambda$ , the rate  $\gamma \bar{x} n$  is lower. So, for  $\mu - \lambda$  to be lower than  $\gamma \bar{x} n$ , the optimal value of  $\mu$  should be lower, which is what we observe from the plots.

Figure 3 illustrates the effect of arrival and departure rates on the given player's expected utility if the player is absent when its utility is being computed. Figure 3(a) shows the change in  $R_{\not\exists}$  with respect to  $\lambda$  for different values of  $\mu$ . The initial decrease and subsequent increase in the utility as  $\lambda$  increases, exhibits an inverted bell-shaped-like nature with respect to the logarithm of  $\lambda$ . For a given  $\mu$ , if  $\lambda$  increases, it is with a higher probability that more players are present, thus resulting in a higher competition if and when the given player arrives in the system, and hence a decrease in  $R_{\not\exists}$ . However, if  $\lambda$  increases beyond a certain threshold, it results in a higher probability of the given player arriving and staying in the system when the problem gets solved, thus resulting in an increase in  $R_{\not\exists}$ . We can estimate the trough of the graph by equating the derivative of  $R_{\not\exists}$  with respect to  $\lambda$  to zero, using its approximate expression from Equation (17); we get



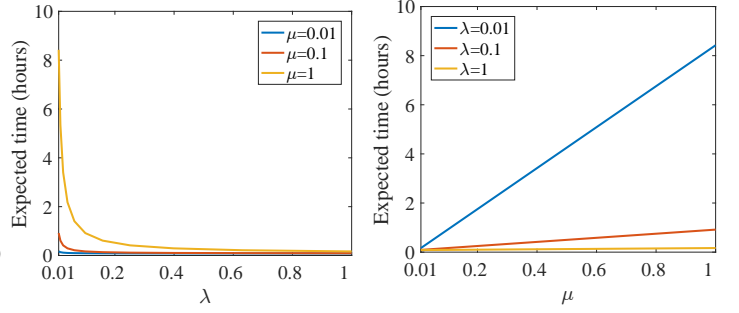
(a) w.r.t.  $\lambda$  for different values of  $\mu$  (b) w.r.t.  $\mu$  for different values of  $\lambda$

Figure 3: Effect of arrival and departure rates on a player's expected utility if the player is absent from the system when the utility is being computed

$$\frac{dR_{\bar{z}}}{d\lambda} = 0 \implies \lambda - \mu = 0 \implies \lambda = \mu$$

To provide an intuition for why the critical point is at  $\lambda - \mu = 0$ , note that if the given player is absent, a first and essential condition for it to obtain a positive utility is that it arrives and stays in the system. Hence, it is advantageous to have a higher probability of arriving while not departing soon after, i.e., to have the effective rate of arrival ( $\lambda - \mu$ ) to be positive, which explains the critical point. Figure 3(b) shows the change in  $R_{\bar{z}}$  with respect to  $\mu$  for different values of  $\lambda$ .  $R_{\bar{z}}$  increases monotonically with  $\mu$ , and its increase with respect to the logarithm of  $\mu$  is a sigmoid-like function. To get an understanding of this observation, note that for a particular  $\lambda$ , if  $\mu$  increases, it is more likely that there are less players present, thus resulting in a lower competition if and when the given player arrives in the system, and hence a higher value of  $R_{\bar{z}}$ .

In all the plots in Figures 2 and 3, it can be seen that as  $\lambda$  or  $\mu$  increases to a very high value, both  $R_{\bar{z}}$  and  $R_{\bar{z}}$  converge to  $\langle R \rangle$ . While this is mathematically clear from Equations (16) and (17), this phenomenon has a natural reasoning. If the values of both  $\lambda$  and  $\mu$  are very high, particularly very high as compared to the average rate of problem getting solved, the mean duration of a full Engset's cycle ( $\frac{1}{\lambda} + \frac{1}{\mu}$ ) would be very low as compared to the time that would be taken for the problem to get solved. So, it is very likely that the given player would depart if it is present, and arrive if it is absent. This would diminish the effect of the given player being present or absent in the beginning when the utility is being computed, thus diminishing the difference between  $R_{\bar{z}}$  and  $R_{\bar{z}}$ . On the other hand, if the value of  $\lambda$  is very high but not that of  $\mu$ , it is very likely that each player would be present for a significant amount of time while the problem is being solved, irrespective of whether it is present when the utility is being computed. Similarly, if the value of  $\mu$  is very high but not that of  $\lambda$ , it is with high probability that each player would be absent for a significant amount of time while the problem is being solved, irrespective of whether it is absent when the utility is being computed. So, both  $R_{\bar{z}}$  and  $R_{\bar{z}}$  approach  $\langle R \rangle$  in these cases, owing to the diminished effect of being present or absent at the time of computing the utility.



(a) w.r.t.  $\lambda$  for different values of  $\mu$  (b) w.r.t.  $\mu$  for different values of  $\lambda$

Figure 4: Effect of arrival and departure rates on the expected time taken to solve the problem

### 5.3. Effect of Arrival and Departure Rates on Time Taken to Solve the Problem

We now discuss how the arrival and departure rates influence the expected time taken to solve the problem, say  $\langle T \rangle$ . Let  $T^s$  be the expected time to solve the problem if the number of players currently present in the system is  $s$ . Recall that the rate of the problem getting solved when there are  $s$  players present, is  $\gamma s \bar{x}$ . Similar to the recursive equations for expected utilities, we can write recursive equation for the time to be taken for solving the problem while accounting for the change in the number of players present in the system, as follows:

$$T^s = \begin{cases} \frac{1}{B^s} + \frac{(N-s)\lambda}{B^s} T^{s+1} + \frac{s\mu}{B^s} T^{s-1} & \text{if } 1 \leq s \leq N-1 \\ \frac{1}{B^0} + T^1 & \text{if } s = 0 \\ \frac{1}{B^N} + \frac{N\mu}{B^N} T^{N-1} & \text{if } s = N \end{cases}$$

where  $B^s = \gamma s \bar{x} + (N-s)\lambda + s\mu$ .

Multiplying the equation corresponding to  $T^s$  by  $B^s$  and bringing all terms containing  $\{T^s\}_{s'=0}^N$  to one side, the above system of equations can be written in matrix form as  $\mathbf{HT} = \mathbf{1}$ , where  $\mathbf{T}$  is a vector whose components are:  $T(s) = T^s$ ,  $\mathbf{1}$  is a vector of all ones, and  $\mathbf{H}$  is a matrix of size  $(N+1) \times (N+1)$  whose elements are:

$$\begin{aligned} H(s, s) &= B^s \\ H(s, s-1) &= -s\mu \text{ (if } s \geq 1) \\ H(s, s+1) &= -(N-s)\lambda \text{ (if } s \leq N) \\ &\text{and all other elements are 0} \end{aligned}$$

We can thus obtain the expected time  $T^s$  to solve the problem, given that there are  $s$  players in the system, by computing  $\mathbf{T} = \mathbf{H}^{-1}\mathbf{1}$ . Hence, the average time to solve the problem  $\langle T \rangle$  can be computed (just as we computed the overall expected utility  $\langle R \rangle$  from  $\langle R^s \rangle$ ), by taking an expectation over the number of players present in the system, based on the steady-state probabilities corresponding to the Engset's system. So, we have  $\langle T \rangle = \sum_{s=0}^N \mathbb{P}_{\lambda, \mu}(s) T^s$ .

Figure 4 presents the effect of arrival and departure rates on the expected time taken to solve the problem. It can be seen that the expected time decreases with  $\lambda$  and increases with  $\mu$ . The decrease of  $\langle T \rangle$  with  $\lambda$  is convex in nature, while its increase with  $\mu$  is linear. We can get an understanding of these

observations using a mean field approximation of the time that would be taken to solve the problem. Since the average number of players present in the system is  $\frac{\lambda}{\lambda+\mu}N$ , the expected time taken to solve the problem would be  $\frac{1}{\gamma\bar{x}(\frac{\lambda}{\lambda+\mu}N)} = \frac{1}{\gamma N\bar{x}}(1 + \frac{\mu}{\lambda})$ . This says that the decrease with respect to  $\lambda$  would be hyperbolic and converging to  $\frac{1}{\gamma N\bar{x}}$ , while the increase with respect to  $\mu$  would be linear with a slope of  $\frac{1}{\lambda}(\frac{1}{\gamma N\bar{x}})$ . It is to be noted again that the mean field approach provides an approximation and cannot replace the actual expression for  $\langle T \rangle$  exactly. As an example, when  $N = 1$ , we have  $T^0 = \frac{\gamma\bar{x} + \lambda + \mu}{\lambda\gamma\bar{x}}$ ,  $T^1 = \frac{\lambda + \mu}{\lambda\gamma\bar{x}}$ , and hence  $\langle T \rangle = \frac{\mu}{\lambda + \mu}T^0 + \frac{\lambda}{\lambda + \mu}T^1 = \frac{1}{\gamma\bar{x}(\frac{\lambda}{\lambda + \mu})} + \frac{\mu}{\lambda(\lambda + \mu)}$ . This is clearly not equal to  $\frac{1}{\gamma\bar{x}(\frac{\lambda}{\lambda + \mu})}$  that is given by the mean field approach.

#### A Note on the Fraction of Time for which a Given Player is Present in the System during the Problem being Solved

In an Engset's system, the fraction of time for which a given player is present is  $\frac{\lambda}{\lambda + \mu}$ . However, this fraction holds valid over a long enough period of time, and does not hold in general. For instance, if we aim to obtain this fraction during the process of a given problem being solved, it would depend on whether or not the player is present at the starting time of solving the problem, and also on how the mean arrival/departure times (and hence the mean duration of a full Engset's cycle) compare with the average time that would be taken for solving the problem. Let  $\mathcal{F}$  denote the fraction of time for which a given player is present while the problem is being solved. We know that the player's overall expected utility while the problem is being solved is  $\langle R \rangle = \frac{1}{N}(r - \frac{c}{\gamma})$ , where  $\frac{r}{N}$  is the expected reward obtained and  $\frac{c}{\gamma N}$  is the expected cost incurred. Note that the expected cost incurred could be alternatively obtained as  $c\bar{x}\langle T \rangle\mathcal{F}$ , where  $\langle T \rangle\mathcal{F}$  gives the expected time for which the given player is present (and hence invests power  $\bar{x}$ ) while the problem is being solved. Equating the two expressions for the expected cost incurred, we get  $c\bar{x}\langle T \rangle\mathcal{F} = \frac{c}{\gamma N}$ , and hence  $\mathcal{F} = \frac{1}{\gamma N\bar{x}\langle T \rangle}$ .

In order to verify that the obtained fraction  $\mathcal{F}$  is indeed different from  $\frac{\lambda}{\lambda + \mu}$ , let us consider the example of  $N = 1$ . Here, as discussed above, we have  $\langle T \rangle = \frac{1}{\gamma\bar{x}(\frac{\lambda}{\lambda + \mu})} + \frac{\mu}{\lambda(\lambda + \mu)}$ , and hence  $\mathcal{F} = \frac{\lambda}{(\lambda + \mu)(1 + \frac{\mu\bar{x}}{(\lambda + \mu)\bar{x}})}$ , which is different from  $\frac{\lambda}{\lambda + \mu}$ . However, it is intuitive that if the mean duration of a full Engset's cycle is significantly lower than the average time that would be taken for solving the problem, the fraction of time for which a given player is present in the system while the problem is being solved, would be approximately equal to  $\frac{\lambda}{\lambda + \mu}$ .

#### 5.4. Effects of Other Parameters on Players' Utilities

We now study the effect of other parameters, namely,  $r, \gamma, \bar{x}$  on the different types of expected utilities of a player. While studying the effect of a parameter, we consider the values of the other parameters to be as mentioned earlier. Further, in order to observe the asymmetry of  $R_{\geq}$  and  $R_{\leq}$  around  $\langle R \rangle$ , we consider  $\lambda \neq \mu$ , in particular,  $\lambda = 0.2, \mu = 0.1$ .

Figure 5 presents the effect of the offered reward on the given player's expected utility. It can be easily seen from the

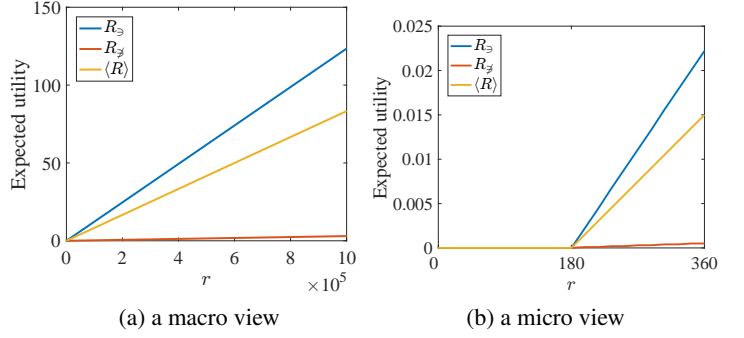


Figure 5: Effect of the offered reward on a player's expected utility

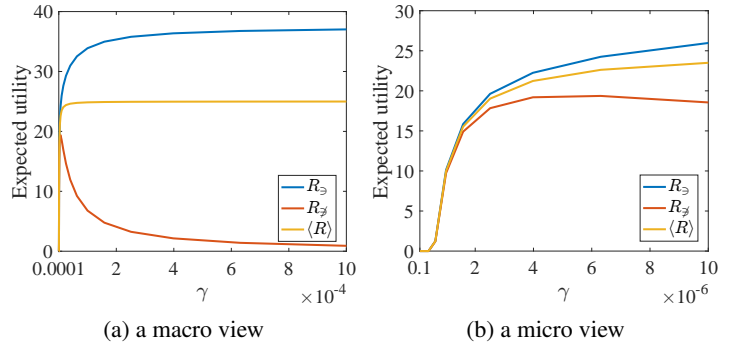


Figure 6: Effect of the rate constant of the problem getting solved on a player's expected utility

micro view that when  $r < \frac{c}{\gamma} = \frac{0.18}{10^{-3}} = 180$ , all types of expected utilities are zero, since the player does not invest (Proposition 2). As the value of  $r$  crosses the threshold of  $\frac{c}{\gamma}$ , it is intuitive that the expected utilities would increase monotonically with the offered reward. We observe that the increase in the overall expected utility is linear with respect to  $r$ , which is as expected since  $\langle R \rangle = \frac{1}{N}(r - \frac{c}{\gamma})$ . Moreover, the increase in  $R_{\geq}$  and  $R_{\leq}$  are also linear as can be understood from their approximate expressions given by Equations (16) and (17), respectively. Further, since we considered  $\lambda = 0.2, \mu = 0.1$ , we have that  $\langle R \rangle = \frac{2}{3}R_{\geq} + \frac{1}{3}R_{\leq}$ , which is why the graph of  $\langle R \rangle$  is closer to that of  $R_{\geq}$  than to  $R_{\leq}$ .

We know that a higher  $\gamma$  leads to the problem getting solved faster, thus resulting in a lower cost to the given player. Figure 6 presents the effect of  $\gamma$  on the given player's expected utility. It can be seen from the micro view that when  $\gamma < \frac{c}{r} = \frac{0.18}{3 \times 10^5} = 0.6 \times 10^{-6}$ , all types of expected utilities are zero, as the player does not invest. As the value of  $\gamma$  crosses the threshold of  $\frac{c}{r}$ , there is a steady increase in all types of expected utilities. It can be understood that the overall expected utility increases monotonically with  $\gamma$  since  $\langle R \rangle = \frac{1}{N}(r - \frac{c}{\gamma})$ . Moreover, the monotone increasing graph of  $R_{\geq}$  as well as the subsequent decrease in the value of  $R_{\leq}$  can also be understood from Equations (16) and (17), respectively. Intuitively, if the given player is present at the time of computing its utility for solving the current problem, a higher value of  $\gamma$  would result in a higher probability of the problem getting solved before the player departs, thus leading to a higher utility. On the other hand, if the given player is absent at the time of computing its utility, a higher value of



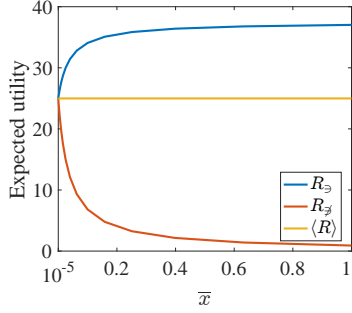


Figure 7: Effect of the maximum power available to players on a player's expected utility

$\gamma$  would result in a higher probability of the problem getting solved before the player arrives, thus leading to a lower utility. Equations (16) and (17) and the expression for  $\langle R \rangle$  also explain why the effect of  $\gamma$  on the expected utilities diminishes beyond a certain value. In particular,  $\langle R \rangle$  converges to  $\frac{r}{N}$ ,  $R_3$  approaches the value  $\frac{r}{\lambda + \mu N}$  (recall that  $\frac{\lambda}{\lambda + \mu}$  is the expected number of players present in the Engset's system), and  $R_\beta$  converges to 0.

While the overall expected utility is not affected by the maximum power available to players ( $\bar{x}$ ) since  $\langle R \rangle = \frac{1}{N} \left( r - \frac{c}{\gamma} \right)$ , its influence on  $R_3$  and  $R_\beta$  is very similar to that of  $\gamma$  (as can be seen from Figure 7). However, there are a couple of subtle differences, which can be understood from Equations (16) and (17). First, while the expected utilities rise from 0 gradually when  $\gamma$  crosses the threshold  $\frac{c}{r}$ , they jump from 0 to  $\frac{1}{N} \left( r - \frac{c}{\gamma} \right)$  when  $\bar{x}$  goes from 0 to an infinitesimally small value, with a discontinuity at  $\bar{x} = 0$ . Another difference from the effect of  $\gamma$  is that, for a very large value of  $\gamma$ , we have  $R_3$  converging to  $\frac{r}{\lambda + \mu N}$ , while for a very large value of  $\bar{x}$ , we have that  $R_3$  converges to  $\frac{1}{N} \left( r - \frac{c}{\gamma} \right)$ . It can be seen that  $R_\beta$  converges to 0 for a very large value of  $\bar{x}$  (same as that for a very large value of  $\gamma$ ).

### 5.5. Effect of the Fixed Power ( $\ell$ )

Throughout our simulation study, we considered the amount of fixed power (i.e., power invested by the fixed players)  $\ell$  to be practically zero. It is to be noted that the nature of the plots studied thus far are similar when we consider  $\ell \rightarrow 0$ , albeit with decreased utility values, and an important difference being that the overall expected utility  $\langle R \rangle$  is no longer a constant (which was  $\frac{1}{N} \left( r - \frac{c}{\gamma} \right)$ ). We hence present an interesting aspect here regarding how the fixed power influences a player's expected utility that is computed when a certain number of players are present in the system. For this set of simulations, we consider  $\lambda = 0.2, \mu = 0.1$  and the values of the other parameters to be as described earlier.

Recall that Figure 1 illustrated the expected utility of a player as computed when a certain number of players are present in the system, when the amount of fixed power  $\ell \rightarrow 0$ . Figure 8 presents the plots corresponding to when the fixed power amounts to (a) one-tenth of the total maximum power available to the players ( $\frac{N}{10}\bar{x}$ ) and (b) exactly the total maximum power available to the players ( $N\bar{x}$ ). It is clear as we compare Figure 1 and Figure 8(a)-(b) that all types of expected utilities (namely,

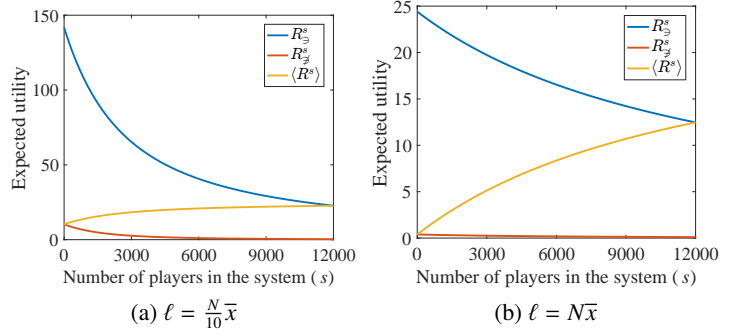


Figure 8: Effect of the fixed power on the expected utility of a player as computed when the system contains a certain number of players ( $\lambda = 0.2, \mu = 0.1$ )

the one computed when the given player is present in the system, the one computed when it is absent, and the overall expected utility) decrease with an increase in the amount of fixed power. This is owing to the increase in competition and the resulting decrease in the probability of the given player winning the reward, irrespective of which state the system is in (which depends on the number of players present in the system and whether the given player is present in the system).

It can also be seen that, while the overall expected utility was a constant in Figure 1 when  $\ell \rightarrow 0$ , it increases with the number of players that are present in the system in Figure 8 ( $\ell \rightarrow 0$ ). Further, a higher  $\ell$  results in a higher relative increase in the overall expected utility. This can be explained with the relative increase in competition in the two cases. Note that as the number of players in the system increases, the relative increase in the total computational power (and hence the relative increase in competition) is lower for a higher value of  $\ell$ . For instance, the relative increase in the total computational power when the number of players in the system increases from  $s$  to  $s + 1$  is  $\frac{((s+1)\bar{x} + \ell) - (s\bar{x} + \ell)}{s\bar{x} + \ell} = \frac{\bar{x}}{s\bar{x} + \ell}$ , which is lower for a higher value of  $\ell$ . When  $\ell \rightarrow 0$ , we observed that the computed overall expected utility did not change with an increase in the number of players that are present in the system; the advantage of an increased probability of the given player being present in the system was balanced out by the disadvantage of an increased competition. However, when  $\ell \rightarrow 0$ , the relative increase in the competition with an increase in the number of players in the system, is lower as compared to the case where  $\ell \rightarrow 0$ . So, the advantage of an increased probability of the given player being present in the system does not balance out, thus resulting in an increase in the overall expected utility as the number of players in the system increases. Further, a higher value of  $\ell$  results in a higher relative increase in the overall expected utility as the number of players in the system increases.

## 6. Conclusion

This work modeled a stochastic game where players can arrive and depart over time in a distributed computing setting, wherein players obtain a certain reward for solving a problem, while incurring a certain cost based on the invested time and computational power. We formulated the utility function and derived a closed form expression for it. We then presented a

game theoretic analysis for determining Markov perfect equilibrium (MPE). We showed that in MPE, players with cost parameters exceeding a certain threshold, do not invest; while those with cost parameters less than this threshold, invest maximal power. Thus, the state knowledge as well as the common knowledge assumptions are shown to be redundant. Using theory of Engset's system, mean field approximation, and extensive simulations, we studied the effects of the arrival and departure rates and other system parameters on players' utilities. In our study, we considered three types of expected utilities that a player can compute depending on whether the player is present, absent, or agnostic with respect to its presence in the system. Following is a summary of the findings of our simulations:

- When studying the effects of arrival and departure rates on the different types of expected utilities of a player, we typically observed a trade-off between the level of competition and the player's likelihood of being present in the system when the problem gets solved. A higher arrival rate led to a higher competition but also a higher likelihood of the player arriving if it was absent at the outset, while a higher departure rate led to a lower competition but with a compromise that the player could depart if it was present.
- The expected utility computed at a time when the player is present, was at least as high as that computed at a time when it is absent. Further, if the fixed power is practically zero, the overall expected utility was a constant with respect to the arrival/departure rates and the number of players present.
- Concerning the expected time taken to solve the problem, its decrease with respect to the arrival rate was hyperbolic, while its increase with respect to the departure rate was linear.
- An increase in the offered reward (beyond our derived threshold) led to a linear increase in all types of expected utilities.
- An increase in the rate constant of the problem getting solved resulted in a steady increase followed by a forking of the different types of expected utilities (that computed when present, increased; that computed when absent, decreased; and the overall expected utility stayed constant), which eventually converged to distinct peculiar values.
- The effect of the maximum power available to players was similar to that of the rate constant of the problem getting solved; an exception being that the different types of expected utilities forked at an infinitesimally small value itself.
- While the overall expected utility was a constant with respect to the number of players present when the fixed power was practically zero, it visibly increased with the number of players present when the amount of fixed power was not ignorable; the increase depended on how the fixed power compared with the total power available to the (strategic) players.

#### *Future Work*

We believe that our model enables us to lay a theoretical foundation for analyzing strategic investments in distributed computing and take a first step towards solving a very challenging

problem, which leaves ample scope for it to be developed further. In order to develop a more sophisticated stochastic model, one could obtain real data concerning the arrivals and departures of players and their investment strategies. We considered the rate of problem getting solved to be proportional to the total computational power being invested, which is applicable to most distributed computing settings including blockchain mining; one could motivate and study other scenarios wherein the rate of problem getting solved has different forms. From the perspective of mechanism design, it would be interesting to design incentives so as to elicit the true cost parameters of the players. Alternatively, one could devise a method for deducing these latent variables (namely, cost parameters) from the observed players' actions and game situations. It would be interesting to analyze the game under bounded rationality. Another promising possibility is to incorporate state-learning in our model. One could study the game by accounting for possibility of players forming coalitions. It is an interesting future direction to study a Stackelberg game with the system as the leader (which decides the amount of reward to offer) and investors of computation as the followers (which decide how much power to invest based on the offered reward), by modeling system's utility as the difference between a practically relevant function of the received computational resources, and the offered reward.

#### **Acknowledgments**

This work was partly supported by CEFIPRA grant No. IFC/DST-Inria-2016-01/448 "Machine Learning for Network Analytics". It was also partly supported by ANR MAESTRO5G (ANR-18-CE25-0012). We thank Monika Dhok (Google, India) for discussion on the effects of computational resources on program execution times. We are thankful to the anonymous reviewers for their useful and insightful comments, which led to improvements across all sections of the paper.

#### **References**

- [1] Z. Zheng, S. Xie, Blockchain challenges and opportunities: A survey, *International Journal of Web and Grid Services* 14 (4) (2018) 352–375.
- [2] S. Nakamoto, A peer-to-peer electronic cash system, <https://bitcoin.org/bitcoin.pdf> (2008).
- [3] L. S. Shapley, Stochastic games, *Proceedings of the National Academy of Sciences* 39 (10) (1953) 1095–1100.
- [4] E. Maskin, J. Tirole, Markov perfect equilibrium: I. observable actions, *Journal of Economic Theory* 100 (2) (2001) 191–219.
- [5] E. Altman, N. Shimkin, Individual equilibrium and learning in processor sharing systems, *Operations Research* 46 (6) (1998) 776–784.
- [6] A. Nahir, A. Orda, D. Raz, Workload factoring with the cloud: A game-theoretic perspective, in: *IEEE International Conference on Computer Communications (INFOCOM)*, Vol. 12, IEEE, 2012, pp. 2566–2570.
- [7] R. Hassin, M. Haviv, Nash equilibrium and subgame perfection in observable queues, *Annals of Operations Research* 113 (1-4) (2002) 15–26.
- [8] J. Wang, F. Zhang, Strategic joining in M/M/1 retrial queues, *European Journal of Operational Research* 230 (1) (2013) 76–87.
- [9] J. Hu, M. Wellman, Nash Q-learning for general-sum stochastic games, *Journal of Machine Learning Research* 4 (Nov) (2003) 1039–1069.
- [10] C. Jiang, Y. Chen, Y.-H. Yang, C.-Y. Wang, K. R. Liu, Dynamic Chinese restaurant game: Theory and application to cognitive radio networks, *IEEE Transactions on Wireless Communications* 13 (4) (2014) 1960–1973.

- [11] C.-Y. Wang, Y. Chen, K. R. Liu, Game-theoretic cross social media analytic: How Yelp ratings affect deal selection on Groupon?, *IEEE Transactions on Knowledge and Data Engineering* 30 (5) (2018) 908–921.
- [12] I. Abraham, D. Dolev, R. Gonen, J. Halpern, Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation, in: *ACM Symposium on Principles of Distributed Computing*, ACM, 2006, pp. 53–62.
- [13] Y.-K. Kwok, S. Song, K. Hwang, Selfish grid computing: Game-theoretic modeling and NAS performance results, in: *IEEE International Symposium on Cluster Computing and the Grid*, Vol. 2, IEEE, 2005, pp. 1143–1150.
- [14] G. Wei, A. V. Vasilakos, Y. Zheng, N. Xiong, A game-theoretic method of fair resource allocation for cloud computing services, *The Journal of Supercomputing* 54 (2) (2010) 252–269.
- [15] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, J. Kubiatowicz, Selfish caching in distributed systems: A game-theoretic analysis, in: *ACM Symposium on Principles of Distributed Computing*, ACM, 2004, pp. 21–30.
- [16] D. Grosu, A. T. Chronopoulos, Noncooperative load balancing in distributed systems, *Journal of Parallel and Distributed Computing* 65 (9) (2005) 1022–1034.
- [17] A. Kishor, R. Niyogi, B. Veeravalli, A game-theoretic approach for cost-aware load balancing in distributed systems, *Future Generation Computer Systems* 109 (2020) 29–44.
- [18] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, D. I. Kim, A survey on blockchain: A game theoretical perspective, *IEEE Access* 7 (2019) 47615–47643.
- [19] I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: *International conference on financial cryptography and data security*, Springer, 2014, pp. 436–454.
- [20] A. Sapirshtein, Y. Sompolinsky, A. Zohar, Optimal selfish mining strategies in Bitcoin, in: *International Conference on Financial Cryptography and Data Security*, Springer, 2016, pp. 515–532.
- [21] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, J. S. Rosenschein, Bitcoin mining pools: A cooperative game theoretic analysis, in: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, IFAAMAS, 2015, pp. 919–927.
- [22] I. Eyal, The miner’s dilemma, in: *2015 IEEE Symposium on Security and Privacy*, IEEE, 2015, pp. 89–103.
- [23] Y. Kwon, D. Kim, Y. Son, E. Vasserman, Y. Kim, Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin, in: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 195–209.
- [24] E. K. Wang, Z. Liang, C.-M. Chen, S. Kumari, M. K. Khan, PoRX: A reputation incentive scheme for blockchain consensus of IIoT, *Future Generation Computer Systems* 102 (2020) 140–151.
- [25] P. Zappalà, M. Belotti, M. Potop-Butucaru, S. Secci, Game theoretical framework for analyzing blockchains robustness, in: *35th International Symposium on Distributed Computing*, Vol. 209, Schloss Dagstuhl, 2021, pp. 42:1–42:18.
- [26] R. Pass, E. Shi, Fruitchains: A fair blockchain, in: *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2017, pp. 315–324.
- [27] X. Chen, C. Papadimitriou, T. Roughgarden, An axiomatic approach to block rewards, in: *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 124–131.
- [28] Z. Xiong, S. Feng, D. Niyato, P. Wang, Z. Han, Optimal pricing-based edge computing resource management in mobile blockchain, in: *IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–6.
- [29] Y. Chen, Z. Li, B. Yang, K. Nai, K. Li, A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing, *Future Generation Computer Systems* 108 (2020) 273–287.
- [30] H. Qiu, T. Li, Auction method to prevent bid-rigging strategies in mobile blockchain edge computing resource allocation, *Future Generation Computer Systems* (2021).
- [31] E. Altman, A. Reiffers, D. S. Menasché, M. Datar, S. Dhamal, C. Touati, R. El-Azouzi, Blockchain competition between miners: A game theoretic perspective, *Frontiers in Blockchain* 2 (2020) 26.
- [32] N. Dimitri, Bitcoin mining as a contest, *Ledger* 2 (2017) 31–37.
- [33] A. T. Haghighat, M. Shajari, Block withholding game among bitcoin mining pools, *Future Generation Computer Systems* 97 (2019) 482–491.
- [34] A. Kiayias, E. Koutsoupias, M. Kyropoulou, Y. Tselekounis, Blockchain mining games, in: *ACM Conference on Economics and Computation (EC)*, ACM, 2016, pp. 365–382.
- [35] B. Biais, C. Bisiere, M. Bouvard, C. Casamatta, The blockchain folk theorem, *The Review of Financial Studies* 32 (5) (2019) 1662–1715.
- [36] E. Hong, How does Bitcoin mining work?, *Investopedia*, <https://www.investopedia.com/tech/how-does-bitcoin-mining-work/#toc-the-mining-process> (2022).
- [37] Blockchain.com, Blockchain explorer: Network difficulty, <https://www.blockchain.com/charts/difficulty> (2022).
- [38] C. Grunspan, R. Pérez-Marco, The mathematics of Bitcoin, *EMS Newsletter* 115 (2020) 31–37.
- [39] Y. Zeng, S. Zuo, The Matthew effect in computation contests: High difficulty may lead to 51% dominance?, in: *The World Wide Web Conference*, ACM, 2019, pp. 2281–2289.
- [40] L. F. G. Sarmenta, Volunteer computing, Ph.D. thesis, Massachusetts Institute of Technology (2001).
- [41] D. P. Anderson, G. Fedak, The computational and storage potential of volunteer computing, in: *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID’06)*, Vol. 1, IEEE, 2006, pp. 73–80.
- [42] J. H. Hubbard, B. B. Hubbard, Vector calculus, linear algebra, and differential forms: A unified approach, Matrix Editions, 2015.
- [43] J. Cohen, The generalized Engset formulae, *Philips Telecommunication Review* 18 (4) (1957) 158–170.