Strategic Investments in Distributed Computing: A Stochastic Game Perspective

SWAPNIL DHAMAL, Chalmers University of Technology, Sweden
WALID BEN-AMEUR, Télécom SudParis, France
TIJANI CHAHED, Télécom SudParis, France
EITAN ALTMAN, INRIA Sophia Antipolis Meditérranée, France
ALBERT SUNNY, Indian Institute of Technology, Palakkad, India
SUDHEER POOJARY, Qualcomm, Bangalore, India

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 the problem or for providing their computational resources, while incurring a certain cost based on the invested time and computational power. We first study a scenario where the reward is offered for solving the problem, such as in blockchain mining. 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. Here, players need not know the system state. We then consider a second scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. Here, in Markov perfect equilibrium, only players with cost parameters in a relatively low range which collectively satisfy a certain constraint in a given state, invest. With simulations in both scenarios, we study the effects of players' arrival and departure rates on the trade-off between their obtained reward and incurred cost, hence on their utilities. We conclude by showing that, if players invest as per the Markov perfect equilibrium, the total invested power in any given state increases monotonically with the offered reward, as a step function in the first scenario and as a piecewise-linear ramp function in the second scenario.

Manuscript submitted for review to the 6th World Congress of the Game Theory Society (GAMES 2020).

1 INTRODUCTION

Distributed computing systems comprise computers which coordinate to solve large problems. In a classical sense, a distributed computing system could be viewed as several providers of computational power contributing to the power of a common central entity (e.g., in volunteer computing [4, 24]). The central entity could, in turn, use the combined power for either fulfilling its own computational needs or distribute it to the next level of requesters of power (e.g., by a computing service provider to its customers in a utility computing model). The center would decide the time for which the system is to be run, and hence the compensation or reward to be given out per unit time to the providers. This compensation or reward would be distributed among the providers based on their respective contributions. A provider incurs a certain cost per unit time for investing a certain amount of power. So, in the most natural setting where the reward per unit time is distributed to the providers in proportion to their contributed power, a higher power investment by a provider is likely to fetch it a higher reward while also increasing its incurred cost, thus resulting in a trade-off.

Distributed computing has gained more popularity than ever, owing to the advent of blockchain. Blockchain has found applications in various fields [31], such as cryptocurrencies, smart contracts, security services, and Internet of Things. Its functioning relies on a proof-of-work procedure [21], where miners (providers 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 proof-of-work 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. 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.

In this paper, we study a stochastic game where players (miners or providers of computational power) can arrive and depart during block mining or during a run of volunteer computing. We consider two of the most common scenarios in distributed computing, namely, (1) in which the reward is offered for solving the problem (e.g., blockchain mining) and (2) in which the reward is offered for contributing to the computational power of a central entity (e.g., volunteer computing).

1.1 Preliminaries

• **Stochastic Game.** A stochastic game [25] 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 [19] 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 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 [3] 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 [20] 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 [11] 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 [27] investigate Nash equilibrium in a queuing system, where reentering the system is a strategic decision. Hu and Wellman [12] 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 [14, 26].

Distributed systems have been studied from the game theoretic perspective [1, 16]. Wei et al. [28] study a resource allocation game in a cloud-based network, with constraints on quality of service. Chun et al. [6] 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 [10] propose a game theoretic framework for obtaining a user-optimal load balancing scheme in distributed systems.

Zheng and Xie [31] present a survey on the challenges in blockchain and recent advances in tackling these challenges. Eyal and Sirer [9] 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. Sapirshtein et al. [23] 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. [18] 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 [8] models a game between two pools employing 'block withholding' attack, and hence discovers the miner's dilemma wherein the revenue of both pools diminishes in Nash equilibrium. Kwon et al. [17] 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.

Pass and Shi [22] 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. [5] present an axiomatic theory of incentives in proof-of-work 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. [29] 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. Altman et al. [2] model the competition over several blockchains as a non-cooperative game, and hence show the existence of pure Nash equilibria using a congestion game approach. Kiayias et al. [15] 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.

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 either of the distributed computing scenarios mentioned earlier. Hence, we first present our models for these scenarios, followed by our analysis of MPE in their induced stochastic games.

1.3 Our Contributions

- We develop stochastic game models which capture the arrival and departure of players in different scenarios of distributed computing: (1) wherein the reward is offered for solving the problem and (2) wherein the reward is offered for contributing to the computational power of a common central entity. We hence derive a closed form expression for the utility function (Section 2).
- We present a game theoretic analysis for determining Markov perfect equilibrium in the two scenarios. For the first scenario, we show 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 (Section 3). In MPE for the second scenario, we show that only players with cost parameters in a relatively low range in a given state, invest; and that players invest proportionally to the 'reward to cost' ratio if they are homogeneous (Section 4).
- We study the effects of the arrival and departure rate parameters on players' utilities, using simulations. In particular, we study the change in trade-off between the expected cost incurred and the expected reward obtained due to the resulting factors such as alterations in the competition, the likelihood of staying out of the system, and the rate of solving the problem (Section 5).
- We discuss how the offered reward would influence the total invested power in a state (Section 6).

2 OUR MODEL

Consider a distributed computing system wherein players receive a certain reward for successfully solving a problem or for providing their computational resources. We first model the scenario where the reward is offered for solving the problem, such as in blockchain mining, and explain it in detail. We then model the scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. We hence point out the similarities and differences between the utility functions of the players in the two scenarios.

2.1 Scenario 1: Model

We present our model for blockchain mining, one of the most in-demand contemporary applications of the scenario where reward is offered for solving the problem. We conclude this subsection by showing that the obtained utility function generalizes to other distributed computing applications belonging to this scenario. 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 proof-of-work solution before all other miners.

Table 1. Notation

r	reward parameter
c_i	cost incurred by player i when it invests unit power for unit time
λ_i	arrival rate corresponding to player i
μ_i	departure rate corresponding to player i
$\mathcal{U}$	universal set of strategic players
ℓ	constant amount of power invested by the fixed players
k	aggregate player accounting for all the fixed players
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 $\mathbf{x}$

• **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 expected utility 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 ℓ, 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 ℓ 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 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 (the expected reward it would obtain minus the expected cost it would incur) from the current time onwards. We denote by cost parameter c_i, the cost incurred by player i for investing unit amount of power for unit time.

In our work, we consider that the cost parameters of all the players are common knowledge. This could be integrated in a blockchain mining or volunteer computing interface where players can declare their cost parameters. This information is then made available to the interfaces of all other players (that is, to the automated software running on the players' machines). In real world, it may not be practical to make the players' cost parameters a common knowledge; moreover, players may not reveal them truthfully. To account for such limitations, a mean field approach could be used by assuming homogeneous or multi-type players (which are special cases of our analysis). Moreover, we will later see that in Scenario 1 (blockchain mining-like scenario), a player's MPE policy depends on only its own cost parameter; so the common knowledge assumption w.r.t. 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. 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. We consider this time to be 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.

• **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 $1/\lambda_j$ (hence, λ_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 μ_j (the expected time after which player j would depart is $1/\mu_j$). Such a stochastic departure can be attributed to the Markovian nature of players (they do not account for how much computation they have invested thus far for mining the current block) and the memoryless nature of computation (the time required to find a solution does not depend on the time invested thus far). Owing to these two factors, players do not monitor block mining progress or the time and power they have already invested for mining the current 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 τ 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 τ, τ'; hence player i's strategy could be written as a function of the state, 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 [30] show that if the number of solutions is ξ, the distance of the probability of a player finding a solution before others, from being proportional to the player's invested power, is $\tilde{O}(1/\xi)$. Since ξ 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 τ is proportional to its invested power at time τ.

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 $\mathrm{P}_j^{(S,\mathbf{x}^{(S)})}$ 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 $\sum_{j \in S \cup \{k\}} \mathrm{P}_j^{(S,\mathbf{x}^{(S)})} = \Gamma^{(S,\mathbf{x}^{(S)})}$. 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 $\mathrm{P}_i^{(S,\mathbf{x}^{(S)})} = \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$, and the rate of other players solving the problem is $\mathrm{Q}_i^{(S,\mathbf{x}^{(S)})} = \sum_{j \in (S \setminus \{i\}) \cup \{k\}} \mathrm{P}_j^{(S,\mathbf{x}^{(S)})} = \frac{\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$.

• **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 are perceived as zero. Note that while we consider a common discounting factor for all players, the 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 $\mathrm{P}_i^{(S,\mathbf{x}^{(S)})}$, 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 $\mathrm{Q}_i^{(S,\mathbf{x}^{(S)})}$, 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 λ_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 μ_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, 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

$$
\begin{aligned}
R_i^{(S,\mathbf{x})} :=& \frac{\Gamma^{(S,\mathbf{x}^{(S)})} \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}}{B^{(S,\mathbf{x})}} \cdot (r + \delta R_i^{(S,\mathbf{x})}) + \frac{\Gamma^{(S,\mathbf{x}^{(S)})} \frac{\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell}}{B^{(S,\mathbf{x})}} \cdot (0 + \delta R_i^{(S,\mathbf{x})}) - \frac{c_i x_i^{(S)}}{B^{(S,\mathbf{x})}} \\
&+ \sum_{j \notin S} \frac{\lambda_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \cup \{j\},\mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{B^{(S,\mathbf{x})}} \cdot R_i^{(S \setminus \{j\},\mathbf{x})}.
\end{aligned} \tag{1}
$$

• **Convergence of 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 $\Gamma^{(S,\mathbf{x}^{(S)})} > 0$. Also, $\delta < 1$. 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 (sum of the elements in each of its rows is less than 1).

Let $\mathbf{F}_i^{(\mathbf{x})}$ 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 2.1. *The recursive equation for* $\mathbf{R}_i^{(\mathbf{x})}$, *Equation* (1), *converges.*

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

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

Now, since $\mathbf{M}^{(\mathbf{x})}$ is strictly substochastic, its spectral radius is less than 1. So when $t \to \infty$, we have $\lim_{t\to\infty}(\mathbf{M}^{(\mathbf{x})})^t = \mathbf{0}$. Since $\mathbf{R}_{i\langle 0\rangle}^{(\mathbf{x})}$ is a finite constant, we have $\lim_{t\to\infty}(\mathbf{M}^{(\mathbf{x})})^t\mathbf{R}_{i\langle 0\rangle}^{(\mathbf{x})} = \mathbf{0}$. Further, $\lim_{t\to\infty}\sum_{\eta=0}^{t-1}(\mathbf{M}^{(\mathbf{x})})^\eta = (\mathbf{I}-\mathbf{M}^{(\mathbf{x})})^{-1}$ [13]. This implicitly means that $(\mathbf{I}-\mathbf{M}^{(\mathbf{x})})$ is invertible. Hence,

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

□

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

It is worth pointing out the change in the denominator, from $B^{(S,\mathbf{x})} = \Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j\notin S}\lambda_j + \sum_{j\in S}\mu_j$ in Equation (1), to $D^{(S,\mathbf{x})} = (1-\delta)\Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j\notin S}\lambda_j + \sum_{j\in S}\mu_j$.

For distributed computing applications with a fixed objective such as finding a solution to a given problem, it is reasonable to assume that the rate of the problem getting solved is proportional to the total power invested by the providers of computation. We, hence, consider that $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma\left(\sum_{j\in S}x_j^{(S)}+\ell\right)$, where γ is the rate constant of proportionality determined by the problem being solved. Hence, 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 $D^{(S,\mathbf{x})} = (1-\delta)\gamma\left(\sum_{j\in S}x_j^{(S)}+\ell\right) + \sum_{j\notin S}\lambda_j + \sum_{j\in S}\mu_j$.

Other Applications of Scenario 1. We derived Expression (2) for the expected utility by considering that the probability of player i being the first to solve the problem is proportional to its invested power at the time, and hence obtains the reward r with this probability. Now, consider another type of system which aims to solve an NP-hard problem with a large search space, where players search for a solution and the system rewards the players in proportion to their invested power when the problem gets solved. In this case, the first term of Expression (2) is replaced with the term $\frac{\Gamma^{(S,\mathbf{x}^{(S)})}\left(\frac{x_i^{(S)}}{\sum_{j\in S}x_j^{(S)}+\ell}r\right)}{D^{(S,\mathbf{x})}}$. So, the mathematical form stays the same, and so when $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma\left(\sum_{j\in S}x_j^{(S)}+\ell\right)$, our analysis presented in Section 3 holds for this case too.

2.2 Scenario 2: Model

We now consider the scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. Here, the center would typically have an idea of the total budget that it is willing to invest and the time for which it wants to run the system. Hence, the reward offered per unit time is inversely proportional to the expected time for which the center decides to run the system. Considering that the time for which the center plans to run the system is exponentially distributed with rate parameter β, the reward offered per unit time is inversely proportional to $\frac{1}{\beta}$, and hence directly proportional to β. Hence, let the offered reward per unit time be $r\beta$, where r is the reward constant of proportionality. Furthermore, the reward given to a player is proportional to its computational investment. So, the revenue received by player i per unit time is $\frac{x_i^{(S)}}{\sum_{j\in S} x_j^{(S)}+\ell} r\beta$, and hence its net profit per unit time is $\frac{x_i^{(S)}}{\sum_{j\in S} x_j^{(S)}+\ell} r\beta - c_i x_i^{(S)}$. The sojourn time in state S, similar to the previous scenario, is $\frac{1}{D^{(S,\mathbf{x})}}$, where $D^{(S,\mathbf{x})} = \beta + \sum_{j\notin S} \lambda_j + \sum_{j\in S} \mu_j$ (here, we have β instead of $\Gamma^{(S,\mathbf{x}^{(S)})}$). So, the net expected profit made by player i in state S before the system transits to another state, is $\frac{\frac{x_i^{(S)}}{\sum_{j\in S} x_j^{(S)}+\ell} r\beta - c_i x_i^{(S)}}{D^{(S,\mathbf{x})}}$.

Hence, player i's expected utility as computed in state S is

$$R_i^{(S,\mathbf{x})} = \frac{\frac{x_i^{(S)}}{\sum_{j\in S} x_j^{(S)}+\ell} r\beta - c_i x_i^{(S)}}{D^{(S,\mathbf{x})}} + \sum_{j\notin S} \frac{\lambda_j}{D^{(S,\mathbf{x})}} \cdot R_i^{(S\cup\{j\},\mathbf{x})} + \sum_{j\in S} \frac{\mu_j}{D^{(S,\mathbf{x})}} \cdot R_i^{(S\setminus\{j\},\mathbf{x})}. \quad (4)$$

Note that since $D^{(S,\mathbf{x})} = \beta + \sum_{j\notin S} \lambda_j + \sum_{j\in S} \mu_j$ here, Expression (4) is obtainable from Expression (2), when $\Gamma^{(S,\mathbf{x}^{(S)})} = \beta$ and $\delta = 0$.

Other Variants of Scenario 2. We considered that the time for which the center decides to run the system is exponentially distributed with rate parameter β, where β is a constant. For theoretical interest, one could consider a generalization where the system may dynamically determine this parameter based on the set of players $S \cup \{k\}$ present in the system. Let such a rate parameter be given by $f(S)$. Since the fixed players and their invested power do not change, these could be encoded in $f(\cdot)$, thus making it a function of only the set of strategic players. The center could determine $f(S)$ based on the cost parameters of the players in set S, the past records of the investments of players in set S, etc. If the time for which the system is to run is independent of the set of players currently present in the system, we have the special case: $f(S) = \beta, \forall S$. It can be easily seen that the analysis presented in this paper (Section 4) goes through directly by replacing β with $f(S)$, since $\Gamma^{(S,\mathbf{x}^{(S)})} = f(S)$ is also independent of the players' investment strategies.

Further, note that if the rate parameter is not just dependent on the set of players present in the system but also proportional to their invested power, it could be written as $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma\left(\sum_{j\in S} x_j^{(S)} + \ell\right)$. This leads to the utility function being given by Equation (3) and hence its analysis is same as that of Scenario 1 (Section 3).

2.3 A Closed-form Expression for the Expected Utility

Note that Equation (2) encompasses both scenarios, where $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma\left(\sum_{j\in S} x_j^{(S)} + \ell\right)$ leads to Scenario 1, while $\Gamma^{(S,\mathbf{x}^{(S)})} = \beta$ and $\delta = 0$ leads to Scenario 2. We now derive a closed-form expression for the expected utility.

Let us define a matrix $\mathbf{W}^{(\mathbf{x})}$ of size $2^{|\mathcal{U}|}\times 2^{|\mathcal{U}|}$. Similar to matrix $\mathbf{M}^{(\mathbf{x})}$, we simply write $W^{(\mathbf{x})}(S,S')$ instead of $W^{(\mathbf{x})}(O(S),O(S'))$, since it does not introduce any ambiguity. Let the elements of $\mathbf{W}^{(\mathbf{x})}$ be:

$$\begin{aligned}
&\text{for } j\notin S:\ W^{(\mathbf{x})}(S,S\cup\{j\}) = \frac{\lambda_j}{D^{(S,\mathbf{x})}}\ ,\\
&\text{for } j\in S:\ W^{(\mathbf{x})}(S,S\setminus\{j\}) = \frac{\mu_j}{D^{(S,\mathbf{x})}}\ ,\\
&\text{and all other elements of } \mathbf{W}^{(\mathbf{x})} \text{ are } 0.
\end{aligned} \tag{5}$$

Here, $D^{(S,\mathbf{x})} = (1-\delta)\Gamma^{(S,\mathbf{x}^{(S)})} + \sum_{j\notin S}\lambda_j + \sum_{j\in S}\mu_j$. Since $\ell > 0$, we have $\Gamma^{(S,\mathbf{x}^{(S)})} > 0$. 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 (sum of the elements in each of its rows is less than 1).

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

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

Along the same line as the proof of lemma 2.1, by having $\mathbf{W}^{(\mathbf{x})}$ in place of $\mathbf{M}^{(\mathbf{x})}$ and $\mathbf{Z}_i^{(\mathbf{x})}$ in place of $\mathbf{F}_i^{(\mathbf{x})}$, we obtain the following result presenting a closed-form expression for the expected utility.

Proposition 2.2. $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I}-\mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Z}_i^{(\mathbf{x})}$.

Owing to the requirement of deriving the inverse of $\mathbf{I}-\mathbf{W}^{(\mathbf{x})}$, it is clear that a general analysis of the concerned stochastic game when considering an arbitrary $\mathbf{W}^{(\mathbf{x})}$ is intractable. In this work, we consider two special scenarios that we motivated earlier in the context of distributed computing systems, for which we show that the analysis turns out to be tractable.

3 SCENARIO 1: ANALYSIS OF 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 [19]. Since our considered game has infinite action spaces in each state, it is not clear whether an MPE exists. In this and the next section, we analyze MPE for the two considered scenarios, thus showing its existence, and hence discuss its properties.

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 2.2), we could effectively determine the fixed point directly.

Now, from Equation (3), the Bellman equations over states $S\in 2^{\mathcal{U}}$ 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 3.1. *In Scenario 1, 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 $O(S)$ 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 (5) 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 $||\mathbf{U}^{(\mathbf{x})}||_\infty \le 1$, 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

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

This is a contradiction since $\mathbf{W}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})}$ is a stochastic matrix. So, we have shown $||\mathbf{U}^{(\mathbf{x})}||_\infty = ||(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Y}^{(\mathbf{x})}\mathbf{1}||_\infty \le 1$. That is, $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Y}^{(\mathbf{x})}$ is substochastic or stochastic. From Proposition 2.2, $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Y}^{(\mathbf{x})}\mathbf{V}_i^{(\mathbf{x})}$. Since $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Y}^{(\mathbf{x})}$ is substochastic or stochastic, $R_i^{(S,\mathbf{x})}$ 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 $R_i^{(S,\mathbf{x})}$ 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 $\gamma r < c_i$. □

Lemma 3.2. *In Scenario 1, $R_i^{(S,\mathbf{x})}$ is a monotone function of $x_i^{(S)}$.*

Proof. We define the following for simplifying notation.

$$A_i^{(S,\mathbf{x})} = \sum_{j\notin S} \lambda_j \hat{R}_i^{(S\cup\{j\},\mathbf{x})} + \sum_{j\in S} \mu_j \hat{R}_i^{(S\setminus\{j\},\mathbf{x})}$$

$$\text{and} \quad E_i^{(S,\mathbf{x}^{(S)})} = \sum_{j\notin S} \lambda_j + \sum_{j\in S} \mu_j + (1-\delta)\gamma\left(\sum_{j\in S\setminus\{i\}} x_j^{(S)} + \ell\right).$$

Hence, we can write

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

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

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 $(\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S,\mathbf{x})}$. □

Proposition 3.3. *In MPE for Scenario 1, a player i invests its maximal power if $\gamma r > c_i$, no power if $\gamma r < c_i$, and any amount of power if $\gamma r = c_i$.*

Proof. Let $W^{(S,\mathbf{x})}$ be the row $O(S)$ 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 3.2, $\frac{dR_i^{(S,\mathbf{x})}}{dx_i^{(S)}}$ 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:

$$\begin{aligned}
&(\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma A_i^{(S,\mathbf{x})} \\
&= (\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)})W^{(S,\mathbf{x})}\hat{R}_i^{(\mathbf{x})} \\
&= (\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)})(\hat{R}_i^{(S,\mathbf{x})} - Z_i^{(S,\mathbf{x})}) \\
&\qquad\qquad (\because \mathbf{W}^{(\mathbf{x})}\mathbf{R}_i^{(\mathbf{x})} = \mathbf{R}_i^{(\mathbf{x})} - \mathbf{Z}_i^{(\mathbf{x})} \text{ from Proposition 2.2}) \\
&= (\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})}(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) + (1-\delta)\gamma\frac{(\gamma r - c_i)x_i^{(S)}}{E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}}(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) \\
&\qquad\qquad (\because Z_i^{(S,\mathbf{x})} = \frac{(\gamma r - c_i)x_i^{(S)}}{E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}} \text{ in Scenario 1}) \\
&= (\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})}(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}) + (1-\delta)\gamma(\gamma r - c_i)x_i^{(S)} \\
&= (\gamma r - c_i)E_i^{(S,\mathbf{x}^{(S)})} - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})}E_i^{(S,\mathbf{x}^{(S)})} + (\gamma r - c_i - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})})(1-\delta)\gamma x_i^{(S)} \\
&= (\gamma r - c_i - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})})E_i^{(S,\mathbf{x}^{(S)})} + (\gamma r - c_i - (1-\delta)\gamma\hat{R}_i^{(S,\mathbf{x})})(1-\delta)\gamma x_i^{(S)} \\
&= (1-\delta)\gamma\left(\frac{1}{1-\delta}\left(r - \frac{c_i}{\gamma}\right) - \hat{R}_i^{(S,\mathbf{x})}\right)(E_i^{(S,\mathbf{x}^{(S)})} + (1-\delta)\gamma x_i^{(S)}).
\end{aligned}$$

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 3.1, 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 2.2 when $\Gamma^{(S,\mathbf{x}^{(S)})} = \gamma\left(\sum_{j\in S} x_j^{(S)} + \ell\right)$.

So, in any state S, it is a dominant strategy for a player i to invest its maximal power if $\gamma r > c_i$, no power if $\gamma r < c_i$, and any amount of power if $\gamma r = c_i$. Since the maximal power of a player i would be bounded (let the bound be $\overline{x}_i$), it would invest $\overline{x}_i$ if $\gamma r > c_i$. Hence, we have a consistent solution for the Bellman equations that a player i invests $\overline{x}_i$ if $\gamma r > c_i$, 0 if $\gamma r < c_i$, and any amount of power in the range $[0, \overline{x}_i]$ if $\gamma r = c_i$. □

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 γr) or alternatively, a threshold on the offered reward r (whether it is higher than $\frac{c_i}{\gamma}$). Note that though a player i invests maximal power when $\gamma r > c_i$, this is not inefficient since the power would be spent for less time as the problem would get solved faster. An intuition behind this result is that, when there are several players in the system, the competition drives them to invest heavily. On the other hand, when there are few players, they invest heavily so that the problem gets solved faster (before arrival of more competition). Also, since the MPE strategies do not depend on S, the assumption of state knowledge can be relaxed.

We now provide an intuition for why the MPE strategies are independent of the arrival and departure rates. From Proposition 2.2, $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Z}_i^{(\mathbf{x})}$. For $\gamma r > c_i$, when power $x_i^{(S)}$ increases, $\mathbf{Z}_i^{(\mathbf{x})}$ increases and the values of elements in $(\mathbf{I}-\mathbf{W}^{(\mathbf{x})})^{-1}$ decrease. But $\mathbf{R}_i^{(\mathbf{x})}$ increases with $x_i^{(S)}$ when $\gamma r > c_i$ (shown in the proof of Proposition 3.3), 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.

For analyzing the expected utility of a strategic player j, let us consider that the power available to it is very large, say $\overline{x}_j$. Following our result on MPE, every player j satisfying $c_j < \gamma r$ would invest $\overline{x}_j$ entirely. So, we have that $\gamma(\sum_{j\in S, c_j<\gamma r} \overline{x}_j + \ell)$ is very large, and hence $D^{(S,\mathbf{x})}$ (which now approximates to $(1-\delta)\gamma(\sum_{j\in S, c_j<\gamma r} \overline{x}_j + \ell))$ is also very large. Since we know that $R_i^{(S\cup\{j\},\mathbf{x})}$ and $R_i^{(S\setminus\{j\},\mathbf{x})}$ are bounded by a small quantity from Lemma 3.1, the limit of the expected utility $R_i^{(S,\mathbf{x})}$ computed in any state S (from Equation (3)) is $\boxed{\frac{\overline{x}_i}{(1-\delta)\left(\sum_{j\in S, c_j<\gamma r} \overline{x}_j + \ell\right)}\left(r - \frac{c_i}{\gamma}\right)}$. To get further insight into this, say ℓ is insignificant, i.e., the computation is dominated by strategic players. Further, say for every strategic player i, $c_i < \gamma r$, and let the very large amount of power available to these players be the same ($\overline{x}_i = \overline{x}_j, \forall i, j \in \mathcal{U}$). Thus, the limit of the expected utility $R_i^{(S,\mathbf{x})}$ computed in any state S simplifies to $\boxed{\frac{1}{1-\delta}\left(\frac{r}{|S|} - \frac{c_i}{\gamma|S|}\right)}$, implying that it is inversely proportional to the number of players in that state. This is intuitive, since if $\overline{x}_i = \overline{x}_j, \forall i, j \in \mathcal{U}$, the reward for mining a block would be won by the players 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

$\frac{c_i}{\gamma|S|}$). Also, owing to the perceived utility being discounted by a factor of δ for every future block, the computed expected utility is $\left(\frac{r}{|S|} - \frac{c_i}{\gamma|S|}\right)(1+\delta+\delta^2+\ldots) = \frac{1}{1-\delta}\left(\frac{r}{|S|} - \frac{c_i}{\gamma|S|}\right)$.

4 SCENARIO 2: ANALYSIS OF MPE

PROPOSITION 4.1. *In MPE for Scenario 2, a player i invests* $x_i^{(S)} = \max\left\{\psi^{(S)}\left(1 - \frac{c_i\psi^{(S)}}{r\beta}\right), 0\right\}$, *where* $\psi^{(S)} = \sum_{j\in S} x_j^{(S)} + \ell = r\beta\frac{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}}c_j}}{2\sum_{j\in\hat{S}}c_j}$. *Here,* $\hat{S}$ *is the maximal set of players* $j \in S$ *which collectively satisfy the constraints* $c_j < \frac{r\beta}{\psi^{(S)}}$. *Set* $\hat{S}$ *can be constructed iteratively by adding players j from set* $S \setminus \hat{S}$ *one at a time, in ascending order of* c_j, *until when adding a new player p to* $\hat{S}$ *violates the constraint* $c_p < \frac{2\sum_{j\in\hat{S}}c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}}c_j}}$.

PROOF. Recall that since $\mathbf{W}^{(\mathbf{x})}$ is a strictly substochastic matrix, $(\mathbf{I}-\mathbf{W}^{(\mathbf{x})})^{-1} = \lim_{t\to\infty}\sum_{\eta=0}^{t-1}(\mathbf{W}^{(\mathbf{x})})^{\eta}$. Since all the elements of $\mathbf{W}^{(\mathbf{x})}$ are non-negative, all the elements of $(\mathbf{W}^{(\mathbf{x})})^{\eta}$ also are non-negative for any natural number η, and hence all the elements of $(\mathbf{I}-\mathbf{W}^{(\mathbf{x})})^{-1}$ are non-negative. Also, since $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I}-\mathbf{W}^{(\mathbf{x})})^{-1}\mathbf{Z}_i^{(\mathbf{x})}$ (Proposition 2.2) and since $\mathbf{W}^{(\mathbf{x})}$ is independent of $x_i^{(S)}$ in Scenario 2, maximizing the components of $\mathbf{Z}_i^{(\mathbf{x})}$ (namely, $Z_i^{(S,\mathbf{x})}$) individually with respect to $x_i^{(S)}$ would essentially maximize all the elements of $\mathbf{R}_i^{(\mathbf{x})}$. Now, since $\Gamma^{(S,\mathbf{x}^{(S)})} = \beta$ in this scenario, we have

$$Z_i^{(S,\mathbf{x})} = \left(\frac{\beta}{\sum_{j\in S}x_j^{(S)}+\ell}r - c_i\right)\frac{x_i^{(S)}}{D^{(S,\mathbf{x})}}.$$

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

As $D^{(S,\mathbf{x})}$ is independent of $x_i^{(S)}$ in this scenario, it can be shown that $Z_i^{(S,\mathbf{x})}$ is a concave function w.r.t. $x_i^{(S)}$ (the second derivative is $\frac{-2r\ell\beta}{(\sum_{j\in S}x_j^{(S)}+\ell)^3 D^{(S,\mathbf{x})}}$). The first order condition $\frac{dZ_i^{(S,\mathbf{x})}}{dx_i^{(S)}} = 0$ gives

$$x_i^{(S)} = \left(\sum_{j\in S}x_j^{(S)}+\ell\right)\left(1 - \frac{c_i}{r\beta}\left(\sum_{j\in S}x_j^{(S)}+\ell\right)\right).$$

Let $\psi^{(S)} = \sum_{j\in S}x_j^{(S)} + \ell$. As $x_i^{(S)}$ is non-negative, we have

$$x_i^{(S)} = \max\left\{\psi^{(S)}\left(1 - \frac{\psi^{(S)}}{r\beta}c_i\right), 0\right\}. \tag{6}$$

Let $\hat{S} = \{j \in S : x_j^{(S)} > 0\}$. We later show how to determine set $\hat{S}$. Summing the above over all players in S and then adding ℓ on both sides, we get

$$\sum_{j\in S}x_j^{(S)} + \ell = \psi^{(S)}\left(|\hat{S}| - \frac{\psi^{(S)}}{r\beta}\sum_{j\in\hat{S}}c_j\right) + \ell.$$

Substituting $\sum_{j\in S}x_j^{(S)} + \ell$ as $\psi^{(S)}$, we get

$$\frac{1}{r\beta}\sum_{j\in\hat{S}}c_j\left(\psi^{(S)}\right)^2 - (|\hat{S}|-1)\psi^{(S)} - \ell = 0.$$

Note that if $\hat{S} = \{\}$ (that is, $x_j^{(S)} = 0, \forall j \in S$), we have $|\hat{S}| = 0$ and $\sum_{j\in\hat{S}}c_j = 0$, in which case we obtain the trivial result $\psi^{(S)} = \ell$. Hence, consider $|\hat{S}| > 0$ and $\sum_{j\in\hat{S}}c_j > 0$.
Solving this equation for positive value of $\psi^{(S)}$, we get

$$\psi^{(S)} = r\beta \frac{|\hat{S}| - 1 + \sqrt{(|\hat{S}| - 1)^2 + \frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}{2\sum_{j\in\hat{S}} c_j}.$$

Substituting this expression for $\psi^{(S)}$ in Equation (6) gives the MPE strategy of player i in state S.

So, $x_i^{(S)} > 0$ iff $c_i < \frac{2\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}$. In other words, $i \in \hat{S}$ iff $c_i < \frac{2\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}$. Now, it is mathematically possible for $\hat{S}$ to consist of players with higher cost parameters while excluding players with lower cost parameters (e.g., consider $\ell \longrightarrow 0, S = \{1, 2, 3\}, c_1 = 1, c_2 = 2, c_3 = 4$; here $\hat{S}$ could be any of $\{1, 2\}, \{1, 3\}, \{2, 3\}$). However, since we are examining MPE, given such a set $\hat{S}$, a non-investing player with a lower cost parameter could unilaterally deviate to invest, which would hence lower the threshold cost parameter, thus compelling a previously investing player with a higher cost parameter to not invest. Hence, the constraint implies that if player i invests, then player j with $c_j < c_i$ also invests. So, there exists a threshold player $\hat{i}$ such that any player j with $c_j > c_{\hat{i}}$ would not invest. Hence, set $\hat{S}$ can be constructed iteratively (initiating from an empty set) by adding players j from set $S \setminus \hat{S}$ one at a time, in ascending order of c_j, until the above constraint is violated for the cost parameter of the newly added player. □

To get a better understanding of this result, if the power ℓ invested by fixed players is considered insignificant, we have $\psi^{(S)} = r\beta\frac{|\hat{S}|-1}{\sum_{j\in\hat{S}} c_j}$ and the condition for $x_i^{(S)} > 0$ simplifies to $\boxed{c_i < \frac{\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1}}$.

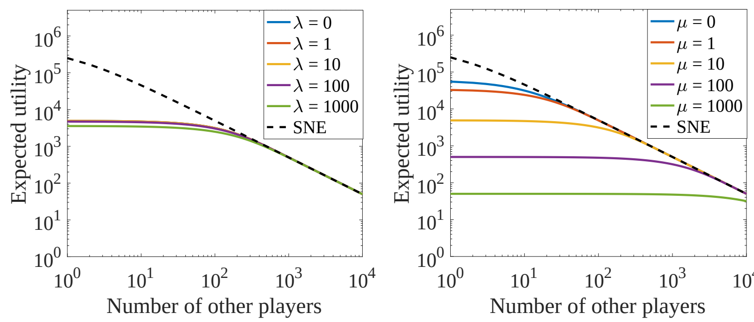
Further, if the strategic players are homogeneous ($c_i = c_j, \forall i, j \in \mathcal{U}$), the cost constraint is satisfied for all players in S $\left(\text{since } c < \frac{|S|c}{|S|-1}\right)$ and so, each of the strategic players invests $\boxed{\frac{r\beta}{c}\left(\frac{|S|-1}{|S|^2}\right)}$. That is, if the computation is dominated by strategic players and they are homogeneous, they would invest proportionally to the 'reward to cost parameter' ratio in MPE. Moreover, a player's investment in a state would be inversely proportional to $\frac{|S|^2}{|S|-1}$, that is, approximately inversely proportional to the number of players in that state.

Since the transition probabilities, and hence $\mathbf{W}^{(\mathbf{x})}$, are constant w.r.t. players' strategies in this scenario, a player's MPE utility computed in state S ($R_i^{(S,\mathbf{x})}$) is a linear combination (with constant non-negative weights) of its utilities over all states computed without accounting for state transitions. Hence, the MPE strategies are independent of the arrival and departure rates.

Note that while the decision regarding whether or not to invest was independent of the cost parameters of the other players in the system in Scenario 1, this decision highly depends on the cost parameters of other players in Scenario 2.

5 EFFECT OF ARRIVAL AND DEPARTURE RATES ON PLAYERS' UTILITIES

Throughout the paper, we determined MPE strategies, which we observed to be independent of players' arrival and departure rates. However, it is clear from Equations (2), (3), (4) and Proposition 2.2 that the players' utilities would depend on these rates. We now study the effects of these rates on the utilities in MPE. It is clear from Proposition 2.2 that computation of expected utilities involves the inverse of $\mathbf{I}-\mathbf{W}^{(\mathbf{x})}$, which is infeasible to obtain analytically, in general. If, 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), the players' sets (states) can be mapped to their cardinalities. Here, in effect, matrix $\mathbf{W}^{(\mathbf{x})}$ of size $2^{|\mathcal{U}|} \times 2^{|\mathcal{U}|}$ could be transformed into a tridiagonal matrix of size $(|\mathcal{U}| + 1) \times (|\mathcal{U}| + 1)$, say, $\overline{\mathbf{W}}^{(\mathbf{x})}$ (by replacing in Equation (5):



(a) for different λ's ($\mu = 10$) (b) for different μ's ($\lambda = 10$)

Fig. 1. Expected utility of a player in Scenario 1

$W^{(\mathbf{x})}(S, S\cup\{j\})$ for $j\notin S$, with $\overline{W}^{(\mathbf{x})}(|S|, |S|+1)$, and $W^{(\mathbf{x})}(S, S\setminus\{j\})$ for $j\in S$, with $\overline{W}^{(\mathbf{x})}(|S|, |S|-1)$), while similarly transforming other matrices. However, even with this simplification, the intricate results on tridiagonal matrices [7] make it intractable to conduct a general analysis. We hence study the effects of the arrival/departure rates on the players' utilities in MPE, by way of simulations.

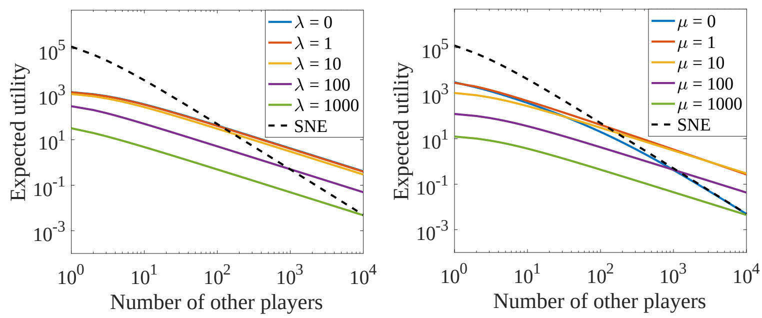
In order to reliably obtain an accurate relation between the arrival/departure rates and the expected utilities of the players, we consider that the computation is dominated by the strategic players (that is, the power invested by the fixed players is insignificant: $\ell \longrightarrow 0$) and the strategic players are homogeneous. Let λ, μ, c denote the common arrival rate, departure rate, and cost parameter, respectively. As mentioned above, if the strategic players are homogeneous, the players' sets (states) can be mapped to their cardinalities. Further, for Scenario 1, 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. We observe how the expected utility of a player changes as a function of the number of other players present in the system, for different arrival/departure rates. We consider the following values: $r = 5 \times 10^5$, $\gamma = \beta = 0.1$, $|\mathcal{U}| = 10^4$, $c = 0.003$ (a justification of these values is provided in Appendix A).

Statewise Nash Equilibrium. For a comparative study, we also look at the equilibrium strategy profile of a given set of players S, when there are no arrivals and departures ($\lambda_j = 0, \forall j \notin S$ and $\mu_j = 0, \forall j \in S$). We call this, *statewise* Nash equilibrium (SNE) in state S. Since the MPE strategies of the players are independent of the arrival and departure rates, a player's SNE strategy in a state is same as its MPE strategy corresponding to that state. Note, however, that the expected utilities in SNE would be different from those in MPE, since the expected utilities highly depend on the arrival and departure rates (Equations (2), (3), (4) and Proposition 2.2). Also, since SNE does not account for change of the set of players present in the system, the expected utilities in SNE for different values on X-axis in the plots are computed independently of each other.

5.1 Simulation Results

In Figures 1 and 2, the plots for expected utility largely follow near-linear curve (of negative slope) on log-log scale w.r.t. the number of players in the system, thus nearly following power law. So, scaling the number of players by a constant factor would lead to proportionate scaling of expected utility.

Scenario 1. Figure 1 presents plots for expected utilities with MPE policy for various values of λ and μ, and compares them with expected utilities in SNE. Following are some insights:



(a) for different λ's ($\mu = 10$)

(b) for different μ's ($\lambda = 10$)

Fig. 2. Expected utility of a player in Scenario 2

- As seen at the end of Section 3, if mining is dominated by strategic players which are homogeneous and $\delta = 0$, the expected utilities in MPE are bounded by $\frac{r}{|S|} - \frac{c}{\gamma|S|}$. It can be similarly shown that the limit of the players' expected utilities in SNE is $\frac{r}{|S|} - \frac{c}{\gamma|S|}$ (this can be seen by substituting in Equation (3): $\lambda_j = 0 \ \forall j \notin S$, $\mu_j = 0$, $c_j = c$, $x_j^{(S)} \to \infty, \forall j \in S$, and $\ell \to 0$). So, the expected utilities in MPE are bounded by the expected utilities in SNE, which is reflected in Figure 1.
- A higher λ results in a higher likelihood of the system having more players, which results in a higher rate of the problem getting solved as well as more competition. This, in turn, reduces the time spent in the system as well as the probability of winning for each player, which hence reduces the cost incurred as well as the expected reward. Figure 1(a) shows that a change in λ leads to an insignificant change in expected utility, suggesting that the change in cost incurred balances the change in expected reward.
- For a given μ, if the number of players changes, there is a balanced trade-off between the cost and the expected reward as above; so the change in expected utility is insignificant. But a higher μ results in a higher probability of player i departing from the system and staying out when the problem gets solved, thus lowering its expected utility (Figure 1(b)).

Scenario 2. Since a player's SNE strategy in a state is same as its MPE strategy corresponding to that state, a player's SNE strategy is to invest $\frac{r\beta}{c}\left(\frac{|S|-1}{|S|^2}\right)$ in state S (as explained at the end of Section 4 when computation is dominated by strategic players that are homogeneous). Further, in SNE, the expected utility of each player can be shown to be $\frac{r}{|S|^2}$ in state S (this can be seen by substituting in Equation (4): $\lambda_j = 0 \ \forall j \notin S$, $\mu_j = 0$, $c_j = c$, $x_j^{(S)} = \frac{r\beta}{c}\left(\frac{|S|-1}{|S|^2}\right), \forall j \in S$, and $\ell \to 0$). Figure 2 presents the plots for expected utilities with the analyzed MPE policy for different values of λ and μ, and compares them against SNE. Following are some insights:

- An increase in the number of players increases competition for the offered reward and hence reduces the reward per unit time received by each player, with no balancing factor (unlike in Scenario 1); so the expected utility decreases.
- For higher λ, there is higher likelihood of system having more players, thus resulting in lower expected utility owing to the aforementioned reason. Also, from Figure 2(a), if λ is not very high, an increase in μ is likely to reduce the competition to the extent that the expected MPE utility when the number of players in the system is large, can exceed the corresponding SNE utility ($\frac{r}{|S|^2}$, which would be very low when the number of players in the system is large).

- A higher μ likely results in less competition, however it also results in a higher probability of player i departing from the system and hence losing out on the reward for the time it stays out; this leads to a trade-off. Figure 2(b) shows that the effect of the probability of player i departing from the system dominates the effect of the reduction in competition. For similar reasons as above, the expected MPE utility when the number of players in the system is large, can exceed the corresponding SNE utility.

6 EFFECT OF OFFERED REWARD ON TOTAL INVESTED POWER

Throughout the paper, we assumed the reward parameter as a given. From the system's viewpoint, however, it is interesting to study how the offered reward influences the total invested power. In Scenario 1, it is clear from Proposition 3.3 that changing the reward from r to $r' > r$ would result in the set of investing players in any given state S to either (a) expand if $\exists\, i \in S : \gamma r < c_i \le \gamma r'$ or (b) stay the same if $\nexists\, i \in S : \gamma r < c_i \le \gamma r'$. Further, since an investing players would invest its maximal power if $c_i < \gamma r'$ and any amount of power if $c_i = \gamma r'$, we have the following result.

Proposition 6.1. *In Scenario 1, if players invest as per Proposition 3.3, the total invested power in any given state is a monotone increasing function of the reward parameter.*

In Scenario 2, it is not clear from Proposition 4.1 whether the total invested power would monotonically increase with r, since the expressions for determining the set of investing players in any given state as well as their invested power are more convoluted. In particular, we need to inspect whether the total invested power could decrease when the set of investing players expands owing to the increased reward. We show the following result.

Proposition 6.2. *In Scenario 2, if players invest as per Proposition 4.1, the total invested power in any given state is a monotone increasing function of the reward parameter.*

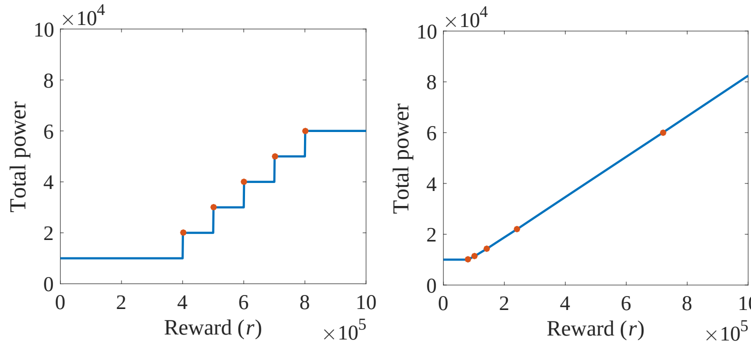
Proof. Recall that in a state S, $\psi^{(S)} = r\beta \frac{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}{2\sum_{j\in\hat{S}} c_j}$, where $\hat{S} \subseteq S$ is the set of investing players. It is clear that for a given set of investors $\hat{S}$, $\psi^{(S)}$ increases monotonically with r. As r varies, set $\hat{S}$ may change, thus changing the values of $|\hat{S}|$ as well as $\sum_{j\in\hat{S}} c_j$. In order to show a monotonic increase of $\psi^{(S)}$ with r despite any changes in set $\hat{S}$, we need to show that at any value of r where players get added to $\hat{S}$, the value of $\psi^{(S)}$ does not decrease (i.e., either increases or stays the same). Without loss of generality, consider that only one player gets added at any such value of r. In what follows, we show continuity at values of r where the set of investing players changes.

Consider a value of r such that the set of investing players is $\hat{S} \setminus \{i\}$ when the reward parameter is infinitesimally lower than r, while it is $\hat{S}$ (i.e., player i gets added to the set of investing players) when the reward parameter is infinitesimally higher than r. At this value of r, let $\underline{\psi}^{(S)}$ be the limit of $\psi^{(S)}$ from the left and $\overline{\psi}^{(S)}$ be its limit from the right. We will now show that $\underline{\psi}^{(S)} = \overline{\psi}^{(S)}$.

Since player i barely satisfies the cost constraint at this value of r, we have (the following equality is in limit): $c_i = \frac{2\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}$. So, the limit of $\psi^{(S)}$ from the right is

$$\overline{\psi}^{(S)} = r\beta \frac{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}{2\sum_{j\in\hat{S}} c_j} = \frac{r\beta}{c_i}. \tag{7}$$

Now, $c_i = \frac{2\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}$ is equivalent to



(a) Scenario 1 (b) Scenario 2

Fig. 3. Effect of the reward parameter r on the total invested power

$$r = \frac{\ell}{\beta} \cdot \frac{c_i^2}{\sum_{j \in \hat{S} \setminus \{i\}} c_j - c_i(|\hat{S}| - 2)}. \tag{8}$$

This gives us an expression for r at which the set of investing players expands from $\hat{S} \setminus \{i\}$ to $\hat{S}$. Now, the limit of $\psi^{(S)}$ from the left is $\underline{\psi}^{(S)} = r\beta \frac{|\hat{S}|-2+\sqrt{(|\hat{S}|-2)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}\setminus\{i\}} c_j}}{2\sum_{j\in\hat{S}\setminus\{i\}} c_j}$.

Let $\underline{\psi}^{(S)} = r\beta y$, where $y = \frac{|\hat{S}|-2+\sqrt{(|\hat{S}|-2)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}\setminus\{i\}} c_j}}{2\sum_{j\in\hat{S}\setminus\{i\}} c_j}$. This, in conjunction with Equation (8), gives

$$y^2 \sum_{j \in \hat{S} \setminus \{i\}} c_j - y(|\hat{S}| - 2) = \left(\frac{1}{c_i}\right)^2 \sum_{j \in \hat{S} \setminus \{i\}} c_j - \frac{1}{c_i}(|\hat{S}| - 2).$$

It can be easily seen that the above equation is satisfied when the value of y is $\frac{1}{c_i}$, and since y has a unique value from its definition, we must have $y = \frac{1}{c_i}$. Hence, from the above and Equation (7), we have $\underline{\psi}^{(S)} = r\beta y = \frac{r\beta}{c_i} = \overline{\psi}^{(S)}$. This completes the proof. □

Figure 3 presents representative plots showing the effect of the reward parameter r on the total invested power in a given state S, for both scenarios. We consider the following values for the purpose of visualization (the plots for any other values follow similar behavior): $\gamma = \beta = 0.1, \ell = 10^4, |S| = 5,$ and $\{c_i\}_{i \in S} = \{4, 5, 6, 7, 8\} \cdot 10^4$ (Scenario 1), $\{c_i\}_{i \in S} = \{0.8, 0.9, 1.0, 1.1, 1.2\}$ (Scenario 2). We vary the value of r from 0 up to 10^6 with a resolution of 10^3. Recall that as r increases, the set of investing players would expand. In the plots, the points at which a previously non-investing player turns into an investing player are marked by red dots. It can be seen that with an increase in r, the total power increases in steps for Scenario 1, as is expected; while it increases similar to a piecewise-linear ramp function for Scenario 2.

Recall that for Scenario 2, in a state S, the investing players $i \in \hat{S}$ collectively satisfy: $c_i < \frac{2\sum_{j\in\hat{S}} c_j}{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}$. For low values of r, the threshold is too low for the players' cost parameters to satisfy; hence no strategic players invest and the total power equals ℓ (this is the base of the ramp function). For values of r which attract investments, the term $\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j$ is of a similar order as $|\hat{S}|$ or lower (this can be seen from the critical value of r derived in Equation (8), which consequently results in $\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j$ being upper bounded by $4|\hat{S}|$). From Proposition 4.1, the total invested power in state S is $\psi^{(S)} = r\beta\frac{|\hat{S}|-1+\sqrt{(|\hat{S}|-1)^2+\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j}}{2\sum_{j\in\hat{S}} c_j}$. Due to the suppressed nature of the term $\frac{4\ell}{r\beta}\sum_{j\in\hat{S}} c_j$ in $\psi^{(S)}$ for values of r which attract investments, the increase in $\psi^{(S)}$ with r is close to being linear within any range of r wherein $\hat{S}$ does not change (hence piecewise-linear).

It is an interesting future direction to study a Stackelberg game with the system as the leader and providers of computation as the followers, by modeling system's utility as the difference between a practically relevant function of the received computational resources, and the offered reward.

7 CONCLUSION AND FUTURE WORK

This work modeled a stochastic game where players can arrive and depart over time in different scenarios of distributed computing, namely, (1) wherein the reward is offered for solving the problem and (2) wherein the reward is offered for contributing to the computational power of a common central entity. We formulated the utility function and derived a closed form expression for it. We then presented game theoretic analysis for determining MPE in the two scenarios.

For Scenario 1, 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. The result that the players' strategies in MPE are independent of the arrival and departure rates, can be attributed to the dominance of the effects of the investments and problem solving rate, over the effect of the state transitions. We also observed that if the computation is dominated by strategic players and they are homogeneous, the expected utility of a player computed in a state is inversely proportional to the number of players in that state, since the reward would be won by such players with equal probabilities and the cost is shared owing to the combined rate of problem solving.

In MPE for Scenario 2, only players with cost parameters in a relatively low range (collectively satisfying a certain constraint) in a given state, invest. If the strategic players are homogeneous and dominate the computation, their investment in a state is proportional to the 'reward to cost' ratio and approximately inversely proportional to the number of players in that state. The players' strategies in MPE were observed to be independent of the arrival and departure rates, since a player's MPE utility computed in a state turned out to be a linear combination with constant non-negative weights, of its utilities over all states computed without accounting for state transitions.

Using simulations, we studied the effects of the arrival and departure rate parameters on the players' utilities. In Scenario 1, a higher arrival rate likely leads to more competition but also a higher problem solving rate, thus balancing the expected reward and cost, and resulting in insignificant change in a player's expected utility. A higher departure rate, however, lowers its expected utility owing to a higher probability of the player departing from the system and staying out when the problem gets solved. In Scenario 2, a higher arrival rate likely leads to more competition with no balancing factor, unlike in Scenario 1, thus lowering a player's expected utility. Though a higher departure rate likely results in less competition, its effect is dominated by the increased probability of the player departing and losing out on the reward for the time it stays out, thus lowering its utility. We concluded by showing for both scenarios that, if players invest as per their MPE strategies, the total power in any given state is a monotone increasing function of the reward parameter. The increase is in steps for Scenario 1, while it is a piecewise-linear ramp function for Scenario 2.

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. 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. As mentioned earlier, a Stackelberg game could be studied, where the system decides the amount of reward to offer, and the computational providers decide how much power to invest based on the offered reward. Among other future directions, one is to study a variant of Scenario 1 where the rate of problem getting solved (and perhaps also the cost) increases non-linearly with the invested power.

REFERENCES

[1] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. 2006. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *ACM Symposium on Principles of Distributed Computing*. ACM, 53–62.

[2] Eitan Altman, Alexandre Reiffers, Daniel S Menasché, Mandar Datar, Swapnil Dhamal, Corinne Touati, and Rachid El-Azouzi. 2020. Blockchain competition between miners: A game theoretic perspective. *Frontiers in Blockchain* 2 (2020), 26.

[3] Eitan Altman and Nahum Shimkin. 1998. Individual equilibrium and learning in processor sharing systems. *Operations Research* 46, 6 (1998), 776–784.

[4] David P Anderson and Gilles Fedak. 2006. The computational and storage potential of volunteer computing. In *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, Vol. 1. IEEE, 73–80.

[5] Xi Chen, Christos Papadimitriou, and Tim Roughgarden. 2019. An axiomatic approach to block rewards. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 124–131.

[6] Byung-Gon Chun, Kamalika Chaudhuri, Hoeteck Wee, Marco Barreno, Christos H Papadimitriou, and John Kubiatowicz. 2004. Selfish caching in distributed systems: A game-theoretic analysis. In *ACM Symposium on Principles of Distributed Computing*. ACM, 21–30.

[7] Mohamed Elouafi and Ahmed Driss Aiat Hadj. 2009. On the powers and the inverse of a tridiagonal matrix. *Appl. Math. Comput.* 211, 1 (2009), 137–141.

[8] Ittay Eyal. 2015. The miner's dilemma. In *2015 IEEE Symposium on Security and Privacy*. IEEE, 89–103.

[9] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*. Springer, 436–454.

[10] Daniel Grosu and Anthony T Chronopoulos. 2005. Noncooperative load balancing in distributed systems. *J. Parallel and Distrib. Comput.* 65, 9 (2005), 1022–1034.

[11] Refael Hassin and Moshe Haviv. 2002. Nash equilibrium and subgame perfection in observable queues. *Annals of Operations Research* 113, 1-4 (2002), 15–26.

[12] Junling Hu and Michael Wellman. 2003. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research* 4, Nov (2003), 1039–1069.

[13] John H Hubbard and Barbara Burke Hubbard. 2015. *Vector calculus, linear algebra, and differential forms: A unified approach.* Matrix Editions.

[14] Chunxiao Jiang, Yan Chen, Yu-Han Yang, Chih-Yu Wang, and KJ Ray Liu. 2014. Dynamic Chinese restaurant game: Theory and application to cognitive radio networks. *IEEE Transactions on Wireless Communications* 13, 4 (2014), 1960–1973.

[15] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. 2016. Blockchain mining games. In *ACM Conference on Economics and Computation (EC)*. ACM, 365–382.

[16] Yu-Kwong Kwok, ShanShan Song, and Kai Hwang. 2005. Selfish grid computing: Game-theoretic modeling and NAS performance results. In *IEEE International Symposium on Cluster Computing and the Grid*, Vol. 2. IEEE, 1143–1150.

[17] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. 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*. 195–209.

[18] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S Rosenschein. 2015. Bitcoin mining pools: A cooperative game theoretic analysis. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 919–927.

[19] Eric Maskin and Jean Tirole. 2001. Markov perfect equilibrium: I. Observable actions. *Journal of Economic Theory* 100, 2 (2001), 191–219.

[20] Amir Nahir, Ariel Orda, and Danny Raz. 2012. Workload factoring with the cloud: A game-theoretic perspective. In *IEEE International Conference on Computer Communications (INFOCOM)*, Vol. 12. IEEE, 2566–2570.

[21] Satoshi Nakamoto. 2008. A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

[22] Rafael Pass and Elaine Shi. 2017. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*. 315–324.

[23] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal selfish mining strategies in Bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 515–532.

[24] Luis Francisco Gumaru Sarmenta. 2001. *Volunteer computing*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[25] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the National Academy of Sciences* 39, 10 (1953), 1095–1100.

[26] Chih-Yu Wang, Yan Chen, and KJ Ray Liu. 2018. 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.

[27] Jinting Wang and Feng Zhang. 2013. Strategic joining in M/M/1 retrial queues. *European Journal of Operational Research* 230, 1 (2013), 76–87.

[28] Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong. 2010. A game-theoretic method of fair resource allocation for cloud computing services. *The Journal of Supercomputing* 54, 2 (2010), 252–269.

[29] Zehui Xiong, Shaohan Feng, Dusit Niyato, Ping Wang, and Zhu Han. 2018. Optimal pricing-based edge computing resource management in mobile blockchain. In *IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[30] Yulong Zeng and Song Zuo. 2019. The Matthew Effect in Computation Contests: High Difficulty May Lead to 51% Dominance?. In *The World Wide Web Conference*. ACM, 2281–2289.

[31] Zibin Zheng and Shaoan Xie. 2018. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14, 4 (2018), 352–375.

A JUSTIFICATION FOR THE VALUES USED IN SIMULATIONS

We take cues from Bitcoin mining for our numerical simulations. The currently offered reward for successfully mining a block is 12.5 Bitcoins. Assuming 1 Bitcoin ≈ \$40,000, the reward translates to $\$5 \times 10^5$. 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 0.1 per minute 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, while consuming about 1.5 kWh in 1 hour, which translates to \$0.18 per hour (at the rate of \$0.12 per kWh), equivalently \$0.003 per minute. As per BitNode (bitnodes.earn.com), a crawler developed to estimate the size of Bitcoin network, the number of Bitcoin miners is around 10^4. Hence, we consider $r = 5 \times 10^5, \gamma = \beta = 0.1, c = 0.003, |\mathcal{U}| = 10^4$.