A Lyapunov Optimization Approach to Repeated Stochastic Games

Michael J. Neely
University of Southern California
http://www-bcf.usc.edu/~mjneely

***Abstract*—This paper considers a time-varying game with N players. Every time slot, players observe their own random events and then take a control action. The events and control actions affect the individual utilities earned by each player. The goal is to maximize a concave function of time average utilities subject to equilibrium constraints. Specifically, participating players are provided access to a common source of randomness from which they can optimally correlate their decisions. The equilibrium constraints incentivize participation by ensuring that players cannot earn more utility if they choose not to participate. This form of equilibrium is similar to the notions of Nash equilibrium and correlated equilibrium, but is simpler to attain. A Lyapunov method is developed that solves the problem in an online *max-weight* fashion by selecting actions based on a set of time-varying weights. The algorithm does not require knowledge of the event probabilities. A similar method can be used to compute a standard correlated equilibrium, albeit with increased complexity.**

I. Introduction

Consider a repeated game with N *players* and one *game manager*. The game is played over an infinite sequence of time slots $t \in \{0, 1, 2, \ldots\}$. Every slot t there is a *random event vector* $\boldsymbol{\omega}(t) = (\omega_0(t), \omega_1(t), \ldots, \omega_N(t))$. The game manager observes the full vector $\boldsymbol{\omega}(t)$, while each player $i \in \{1, \ldots, N\}$ observes only the component $\omega_i(t)$. The value $\omega_0(t)$ represents information known only to the manager. After the slot t event is observed, the game manager sends a message to each player i. Based on this message, the players choose a *control action* $\alpha_i(t)$. The random event and the collection of all control actions for slot t determine individual *utilities* $u_i(t)$ for each player $i \in \{1, \ldots, N\}$. Each player is interested in maximizing the time average of its own utility process. The game manager is interested in providing messages that lead to a fair allocation of time average utilities across players.

Specifically, let $\overline{u}_i$ be the time average of $u_i(t)$. The *fairness* of an achieved vector of time average utilities is defined by a *concave fairness function* $\phi(\overline{u}_1, \ldots, \overline{u}_N)$. The goal is to devise strategies that maximize $\phi(\overline{u}_1, \ldots, \overline{u}_N)$ subject to certain game-theoretic equilibrium constraints. For example, suppose the fairness function is a sum of logarithms:

$$\phi(\overline{u}_1, \ldots, \overline{u}_N) = \sum_{i=1}^{N} \log(\overline{u}_i)$$

This corresponds to *proportional fair utility maximization*, a concept often studied in the context of communication networks [1]. Another natural concave fairness function is:

$$\phi(\overline{u}_1, \ldots, \overline{u}_N) = \min[\overline{u}_1, \ldots, \overline{u}_N, c]$$

for some given constant $c > 0$. This fairness function assigns no added value when the average utility of one player exceeds that of another.

Let $\boldsymbol{M}(t) = (M_1(t), \ldots, M_N(t))$ be the *message vector* provided by the game manager on slot t. The value $M_i(t)$ is an element of the set $\mathcal{A}_i$ and represents the action the manager would like player i to take. A player $i \in \{1, \ldots, N\}$ is said to *participate* if she always chooses the suggestion of the manager, that is, if $\alpha_i(t) = M_i(t)$ for all $t \in \{0, 1, 2, \ldots\}$. At the beginning of the game, each player makes a participation agreement. Participating players receive the messages $M_i(t)$, while non-participating players do not.

This paper considers the class of algorithms that deliver message vectors $\boldsymbol{M}(t)$ as a stationary and randomized function of the observed $\boldsymbol{\omega}(t)$. Assuming that all players participate, this induces a conditional probability distribution on the actions, given the current $\boldsymbol{\omega}(t)$. The conditional distribution is defined as a *coarse correlated equilibrium* (CCE) if it yields a time average utility vector $(\overline{u}_1, \ldots, \overline{u}_N)$ with the following property [2]: For each player $i \in \{1, \ldots, N\}$, the average utility $\overline{u}_i$ is at least as large as the maximum time average utility this player could achieve if she did not participate (assuming the actions of all other players do not change). Overall, the goal is to maximize $\phi(\overline{u}_1, \ldots, \overline{u}_N)$ subject to the CCE constraints.

A. Contributions and related work

The notion of *coarse correlated equilibrium* (CCE) was introduced in [2] in the static case where there is no event process $\boldsymbol{\omega}(t)$. The CCE definition is similar to a *correlated equilibrium* (CE) [3][4][5]. The difference is as follows: A correlated equilibrium (CE) is more stringent and requires the utility achieved by each player i to be at least as large as the utility she could achieve if she did not participate *but if she still knew the $M_i(t)$ messages on every slot*. It is known that both CCE and CE constraints can be written as linear programs. The concept of *Nash equilibrium* (NE) is more stringent still: The NE constraint requires all players to act independently without the aid of a message process $\boldsymbol{M}(t)$ [6][5]. Unfortunately, the problem of computing a Nash equilibrium is nonconvex.

This paper uses the NE, CE, and CCE concepts in the context of a stochastic game with random events $\boldsymbol{\omega}(t)$. When fairness functions are concave but nonlinear, the optimal action associated with a particular event can depend on whether or not the event is rare. This paper develops an online algorithm that is influenced by the event probabilities, but does not require knowledge of these probabilities. The algorithm uses the Lyapunov optimization theory of [7][8] and is of the *max-weight* type. Specifically, every slot t, the game manager observes the $\boldsymbol{\omega}(t)$ realization and chooses a suggestion vector by greedily minimizing a *drift-plus-penalty expression*. Such Lyapunov methods are used extensively in the context of queueing networks [9][10] (see also related methods in [11][12][13]). This is perhaps the first use of such techniques in a game-theoretic setting.

One reason the solution of this paper can have a simple structure is that the random event process $\boldsymbol{\omega}(t)$ is assumed to be independent of the prior control actions. Specifically, while the components $\omega_i(t)$ are allowed to be arbitrarily correlated across $i \in \{0, 1, \ldots, N\}$, the vector $\boldsymbol{\omega}(t)$ is assumed to be independent and identically distributed (i.i.d.) over slots. Prior work on stochastic games considers more complex problems where $\boldsymbol{\omega}(t+1)$ is influenced by the control action of slot t, including work in [14] which studies correlated equilibria in this context. This typically involves Markov decision theory and has complexity that grows exponentially with the dimension of the state vector $\boldsymbol{\omega}(t)$, and hence exponentially in the number of players N.

In contrast, while the current paper treats a stochastic problem with more limited structure, the resulting solution scales gracefully with N. Specifically, the algorithm uses a number of *virtual queues* that is linear in N, rather than exponential in N, resulting in polynomial time bounds on complexity and convergence time. Furthermore, the number of virtual queues is independent of the number of possible values of $\omega_0(t)$. Unfortunately, the number of virtual queues is exponential in the number of possible values of $\omega_i(t)$ for $i \in \{1, \ldots, N\}$. Hence, the algorithm works best when players observe only a small number of possible random events.

II. Static games

This section introduces the problem in the *static case* without random processes $\omega_0(t), \omega_1(t), \ldots, \omega_N(t)$. The different forms of equilibrium are defined and compared through a simple example. The general stochastic problem is treated in Section III.

Suppose there are N players, where N is an integer larger than 1. Each player $i \in \{1, \ldots, N\}$ has an *action space* $\mathcal{A}_i$, assumed to be a finite set. The game operates in slotted time $t \in \{0, 1, 2, \ldots\}$. Every slot t, each player i chooses an action $\alpha_i(t) \in \mathcal{A}_i$. Let $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_N(t))$ be the vector of control actions on slot t. The utility $u_i(t)$ earned by player i on slot t is a real-valued function of $\boldsymbol{\alpha}(t)$:

$$u_i(t) = \hat{u}_i(\boldsymbol{\alpha}(t)) \ \forall i \in \{1, \ldots, N\}$$

The utility functions $\hat{u}_i(\boldsymbol{\alpha})$ can be different for each player i. Define $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. Consider starting with a particular vector $\boldsymbol{\alpha} \in \mathcal{A}$ and modifying it by changing a single entry i from α_i to some other action β_i. This new vector is represented by the notation $(\beta_i, \boldsymbol{\alpha}_{\bar{i}})$. Define $\mathcal{A}_{\bar{i}}$ as the set of all vectors $\boldsymbol{\alpha}_{\bar{i}}$, being the set product of $\mathcal{A}_j$ over all $j \neq i$.

The three different forms of equilibrium considered in this section are defined by probability mass functions $Pr[\boldsymbol{\alpha}]$ for $\boldsymbol{\alpha} \in \mathcal{A}$. It is assumed throughout that:

- $Pr[\boldsymbol{\alpha}] \geq 0$ for all $\boldsymbol{\alpha} \in \mathcal{A}$.
- $\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}] = 1$.

If actions $\boldsymbol{\alpha}(t)$ are chosen independently every slot according to the same probability mass function $Pr[\boldsymbol{\alpha}]$, the law of large numbers ensures that, with probability 1, the time average utility of each player $i \in \{1, \ldots, N\}$ is:

$$\overline{u}_i = \textstyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}] \hat{u}_i(\boldsymbol{\alpha})$$

A. Nash equilibrium (NE)

The standard concept of Nash equilibrium assumes players take independent actions, so that $Pr[\boldsymbol{\alpha}] = \prod_{i=1}^{N} g_i[\alpha_i]$, where $g_i[\alpha_i]$ is defined for all $i \in \{1, \ldots, N\}$ and $\alpha_i \in \mathcal{A}_i$ by:

$$g_i[\alpha_i] = Pr[\alpha_i(t) = \alpha_i]$$

A collection of such functions $g_i[\alpha_i]$ for $i \in \{1, \ldots, N\}$ defines a *mixed strategy Nash equilibrium (NE)* if [15][6]:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \prod_{j=1}^{N} g_j[\alpha_j] \hat{u}_i(\boldsymbol{\alpha}) \geq \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \prod_{j=1}^{N} g_j[\alpha_j] \hat{u}_i(\beta_i, \boldsymbol{\alpha}_{\bar{i}}) \quad \forall i \in \{1, \ldots, N\}, \forall \beta_i \in \mathcal{A}_i \tag{1}$$

B. Correlated equilibrium (CE)

The standard concept of correlated equilibrium from [3][4] can be motivated by a game manager that provides suggested actions $(\alpha_1(t), \ldots, \alpha_N(t))$ every slot t, where player 1 only sees $\alpha_1(t)$, player 2 only sees $\alpha_2(t)$, and so on. Assume the suggestion vector is independent and identically distributed (i.i.d.) over slots with some probability mass function $Pr[\boldsymbol{\alpha}]$. Assume all players participate, so that every slot their chosen actions match the suggestions. The probability mass function $Pr[\boldsymbol{\alpha}]$ is a *correlated equilibrium (CE)* if [3][4]:

$$\sum_{\boldsymbol{\alpha}_{\bar{i}} \in \mathcal{A}_{\bar{i}}} Pr[\alpha_i, \boldsymbol{\alpha}_{\bar{i}}] \hat{u}_i(\alpha_i, \boldsymbol{\alpha}_{\bar{i}}) \geq \sum_{\boldsymbol{\alpha}_{\bar{i}} \in \mathcal{A}_{\bar{i}}} Pr[\alpha_i, \boldsymbol{\alpha}_{\bar{i}}] \hat{u}_i(\beta_i, \boldsymbol{\alpha}_{\bar{i}}) \quad \forall i \in \{1, \ldots, N\}, \forall \alpha_i \in \mathcal{A}_i, \forall \beta_i \in \mathcal{A}_i \text{ with } \beta_i \neq \alpha_i \tag{2}$$

This can be understood as follows: Fix an $i \in \{1, \ldots, N\}$ and an $\alpha_i \in \mathcal{A}_i$ such that $Pr[\alpha_i(t) = \alpha_i] > 0$. Divide both sides of the above inequality by $Pr[\alpha_i(t) = \alpha_i]$. Then:

- The left-hand-side is the conditional expected utility of player i, given that all players participate and that player i sees suggestion α_i on the current slot.
- The right-hand-side is the conditional expected utility of player i, given that she sees α_i on the current slot, that all other players $j \neq i$ participate, and that player i chooses action β_i instead of α_i (so player i does *not* participate).

The correlated equilibrium constraints are linear in the $Pr[\boldsymbol{\alpha}]$ variables. Define $|\mathcal{A}_i|$ as the number of actions in set $\mathcal{A}_i$. The number of linear constraints specified by (2) is then:

$$\textstyle\sum_{i=1}^{N} |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$$

C. Coarse correlated equilibrium (CCE)

The definition of correlated equilibrium assumes that non-participating players still receive the suggestions from the game manager. As the suggestion $\alpha_i(t)$ for player i may be correlated with the suggestions $\alpha_j(t)$ of other players $j \neq i$, this can give a non-participating player i a great deal of information about the likelihood of actions from other players. The following simple modification assumes that non-participating players do not receive any suggestions from the game manager. A probability mass function $Pr[\boldsymbol{\alpha}]$ is a *coarse correlated equilibrium (CCE)* if:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}]\hat{u}_i(\boldsymbol{\alpha}) \geq \sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}]\hat{u}_i(\beta_i, \boldsymbol{\alpha}_{\bar{i}})$$
$$\forall i \in \{1, \ldots, N\}, \forall \beta_i \in \mathcal{A}_i \quad (3)$$

This CCE definition was introduced in [2]. While the above constraints are defined in terms of fixed alternatives β_i, it can be shown that these constraints imply that no player i can reach an improved payoff by using a different strategy. Intuitively, this is because any different strategy can, at a given time t, be viewed as a probabilistic mixture of fixed actions.

These CCE constraints (3) are linear in the $Pr[\boldsymbol{\alpha}]$ values. The number of CCE constraints is:

$$\textstyle\sum_{i=1}^{N} |\mathcal{A}_i|$$

This number is typically much less than the number of constraints required for a CE in (2). Assuming that $|\mathcal{A}_i| \geq 2$ for each player i (so that each player has at least 2 action options), the number of CCE constraints is always less than or equal to the number of CE constraints, with equality if and only if $|\mathcal{A}_i| = 2$ for all players i.

D. A superset result

The assumption that all sets $\mathcal{A}_i$ are finite make the game a *finite game*. Fix a finite game and define $\mathcal{E}_{Nash}$, $\mathcal{E}_{CE}$, and $\mathcal{E}_{CCE}$ as the set of all probability mass functions $Pr[\boldsymbol{\alpha}]$ that define a (mixed strategy) Nash equilibrium, a correlated equilibrium, and a coarse correlated equilibrium, respectively. It is known that every such finite game has at least one mixed strategy Nash equilibrium, and so $\mathcal{E}_{Nash}$ is nonempty [15][6]. Furthermore, in [3][4] it is shown that any (mixed strategy) Nash equilibrium is also a correlated equilibrium. Similarly, it can be shown that any correlated equilibrium is also a coarse correlated equilibrium [2]. Thus:

$$\mathcal{E}_{Nash} \subseteq \mathcal{E}_{CE} \subseteq \mathcal{E}_{CCE} \quad (4)$$

Furthermore, the sets $\mathcal{E}_{CE}$ and $\mathcal{E}_{CCE}$ are closed, bounded, and convex. For example, this is true for $\mathcal{E}_{CCE}$ because this set is the intersection of the linear constraints (3) and the closed, bounded, and convex probability simplex.

E. A simple example

Consider a game where player 1 has three control options and player 2 has two control options:

$$\mathcal{A}_1 = \{\alpha, \beta, \gamma\} \ , \ \mathcal{A}_2 = \{\alpha, \beta\}$$

The utility functions $\hat{u}_1(\alpha_1, \alpha_2)$ and $\hat{u}_2(\alpha_1, \alpha_2)$ are specified in the table of Fig. 1, where player 1 actions are listed by row and player 2 actions are listed by column.

Utility 1

	α	β
α	2	5
β	4	2
γ	3	5

Utility 2

	α	β
α	50	1
β	2	4
γ	3	0

Probabilities

	α	β
α	a	b
β	c	d
γ	e	f

Fig. 1. Example utility functions $\hat{u}_1(\alpha_1, \alpha_2)$ and $\hat{u}_2(\alpha_1, \alpha_2)$.

There are six possible action vectors (α_1, α_2). Define the mass function $Pr[\boldsymbol{\alpha}]$ by values a, b, c, d, e, f associated with each of the six possibilities, as shown in Fig. 1.

The eight CE constraints for this problem are:

$$\begin{array}{ll}
\text{player 1 sees } \alpha: & 2a + 5b \geq 4a + 2b \\
\text{player 1 sees } \alpha: & 2a + 5b \geq 3a + 5b \\
\text{player 1 sees } \beta: & 4c + 2d \geq 2c + 5d \\
\text{player 1 sees } \beta: & 4c + 2d \geq 3c + 5d \\
\text{player 1 sees } \gamma: & 3e + 5f \geq 2e + 5f \\
\text{player 1 sees } \gamma: & 3e + 5f \geq 4e + 2f \\
\text{player 2 sees } \alpha: & 50a + 2c + 3e \geq a + 4c + 0e \\
\text{player 2 sees } \beta: & b + 4d + 0f \geq 50b + 2d + 3f
\end{array}$$

It can be shown that there is a single probability mass function $Pr[\boldsymbol{\alpha}]$ that satisfies all of these CE constraints:

$$a = b = 0, \ \ c = 0.45, \ \ d = 0.15, \ \ e = 0.3, \ \ f = 0.1$$

This is also the only NE. The average utility vector associated with this mass function is $(\overline{u}_1, \overline{u}_2) = (3.5, 2.4)$.

In contrast, the five CCE constraints for this problem are:

$$\begin{array}{ll}
\text{player 1 chooses } \alpha: & 2a + 5b + 4c + 2d + 3e + 5f \\
 & \quad \geq 2(a + c + e) + 5(b + d + f) \\
\text{player 1 chooses } \beta: & 2a + 5b + 4c + 2d + 3e + 5f \\
 & \quad \geq 4(a + c + e) + 2(b + d + f) \\
\text{player 1 chooses } \gamma: & 2a + 5b + 4c + 2d + 3e + 5f \\
 & \quad \geq 3(a + c + e) + 5(b + d + f) \\
\text{player 2 chooses } \alpha: & 50a + b + 2c + 4d + 3e \\
 & \quad \geq 50(a + b) + 2(c + d) + 3(e + f) \\
\text{player 2 chooses } \beta: & 50a + b + 2c + 4d + 3e \\
 & \quad \geq 1(a + b) + 4(c + d) + 0(e + f)
\end{array}$$

There are an infinite number of probability mass functions $Pr[\boldsymbol{\alpha}]$ that satisfy these CCE constraints. Three different ones are given in the table of Fig. 2, labeled *distribution 1*, *distribution 2*, and *distribution 3*. Distribution 1 corresponds to the CE and NE distribution.

The set of all utility vectors $(\overline{u}_1, \overline{u}_2)$ achievable under CCE constraints is the triangular region shown in Fig. 3. The three vertices of the triangle correspond to the three distributions in Fig. 2, and are:

$$(\overline{u}_1, \overline{u}_2) \in \{(3.5, 2.4), (3.5, 9.3), (3.8773, 3.7914)\}$$

The point $(3.5, 2.4)$ is the lower left vertex of the triangle and corresponds to the CE (and NE) distribution. It is clear that both players can significantly increase their utility by changing from CE constraints to CCE constraints. This illustrates the following general principle: *All players benefit if non-participants are denied access to the suggestions of the game manager*. This principle is justified by (4).

Distribution 1

	α	β
α	0	0
β	.45	.15
γ	.30	.10

Distribution 2

	α	β
α	.15	0
β	.60	.15
γ	0	.10

Distribution 3

	α	β
α	.0368	0
β	.9018	.0368
γ	0	.0245

Fig. 2. Three different probability distributions that satisfy the CCE constraints. The first distribution also satisfies the CE and NE constraints.

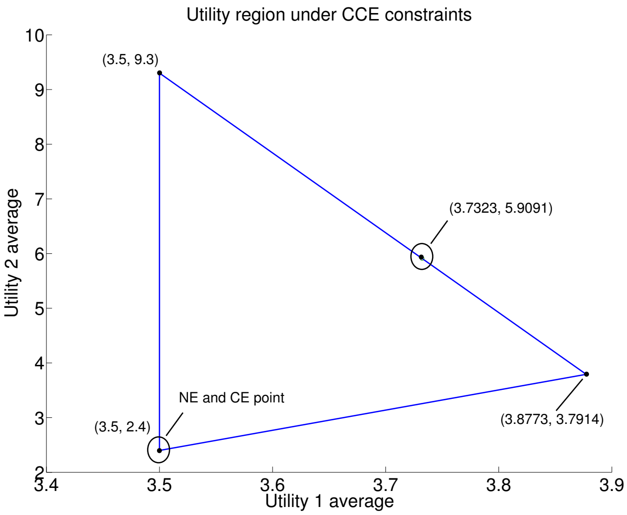

Fig. 3. The region of $(\overline{u}_1, \overline{u}_2)$ values achievable under CCE constraints. All points inside and on the triangle are achievable. The NE and CCE point is the lower left vertex. The point $(3.7323, 5.9091)$ is the solution to the convex optimization example of Section II-F.

F. Utility optimization with equilibrium constraints

For convenience, assume all utility functions are nonnegative. Define u_i^{max} as an upper bound on the utility for each player $i \in \{1, \ldots, N\}$, so that:

$$0 \leq \hat{u}_i(\boldsymbol{\alpha}) \leq u_i^{max} \quad \forall \boldsymbol{\alpha} \in \mathcal{A}$$

Define $\phi(u_1, \ldots, u_N)$ as a continuous and concave function that maps the set $\times_{i=1}^N [0, u_i^{max}]$ to the real numbers. This is called the *fairness function*. The game manager chooses a probability mass function $Pr[\boldsymbol{\alpha}]$ with the goal of maximizing $\phi(\overline{u}_1, \ldots, \overline{u}_N)$ subject to CCE constraints:

$$\text{Maximize:} \quad \phi(\overline{u}_1, \ldots, \overline{u}_N) \tag{5}$$

$$\text{Subject to:} \quad \overline{u}_i = \textstyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}]\hat{u}_i(\boldsymbol{\alpha}) \ \forall i \in \{1, \ldots, N\} \tag{6}$$

$$Pr[\boldsymbol{\alpha}] \geq 0 \ \forall \boldsymbol{\alpha} \in \mathcal{A} \tag{7}$$

$$\textstyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}] = 1 \tag{8}$$

$$\text{CCE constraints (3) are satisfied} \tag{9}$$

The above is a convex optimization problem. If the CCE constraints are replaced by the CE constraints (2), the problem remains convex but can have significantly more constraints. If the CCE constraints are replaced with the NE constraints (1), the problem becomes nonconvex.

Consider the special case example of Section II-E with fairness function given by:

$$\phi(\overline{u}_1, \overline{u}_2) = 10\log(1 + \overline{u}_1) + \log(1 + \overline{u}_2)$$

where player 1 is given a higher priority. The optimal utility is $(\overline{u}_1^*, \overline{u}_2^*) = (3.7323, 5.9091)$, plotted in Fig. 3.

III. Stochastic games

Let $\boldsymbol{\omega}(t) = (\omega_0(t), \omega_1(t), \ldots, \omega_N(t))$ be a vector of random events for slot $t \in \{0, 1, 2, \ldots\}$. Each component $\omega_i(t)$ takes values in some finite set Ω_i, for $i \in \{0, 1, \ldots, N\}$. Define $\Omega = \Omega_0 \times \Omega_1 \times \cdots \times \Omega_N$. The vector process $\boldsymbol{\omega}(t)$ is assumed to be independent and identically distributed (i.i.d.) over slots with probability mass function:

$$\pi[\boldsymbol{\omega}] \overset{\triangle}{=} Pr[\boldsymbol{\omega}(t) = \boldsymbol{\omega}] \quad \forall \boldsymbol{\omega} \in \Omega$$

where the notation "$\overset{\triangle}{=}$" means "defined to be equal to." On each slot t, the components of the vector $\boldsymbol{\omega}(t)$ can be arbitrarily correlated.

At the beginning of each slot t, each player $i \in \{1, \ldots, N\}$ observes its own random event $\omega_i(t)$. The game manager observes the full vector $\boldsymbol{\omega}(t)$, including the additional information $\omega_0(t)$. It then sends a suggested action $M_i(t)$ to each participating player $i \in \{1, \ldots, N\}$. Assume $M_i(t) \in \mathcal{A}_i$, where $\mathcal{A}_i$ is the finite set of actions available to player i. Each player i chooses an action $\alpha_i(t) \in \mathcal{A}_i$. Participating players always choose $\alpha_i(t) = M_i(t)$. Non-participating players do not receive $M_i(t)$ and choose $\alpha_i(t)$ using knowledge of only $\omega_i(t)$ and of events that occurred before slot t.

Let $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_N(t))$ be the action vector. The utility $u_i(t)$ earned by each player i on slot t is a function of $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\omega}(t)$:

$$u_i(t) = \hat{u}_i(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$$

For convenience, assume utility functions are nonnegative with maximum values u_i^{max} for $i \in \{1, \ldots, N\}$, so that:

$$0 \leq \hat{u}_i(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t)) \leq u_i^{max}$$

A. Discussion of game structures

This model can be used to treat various game structures. For example, the scenario where all players have full information can be treated by defining $\omega_i(t) = \omega_0(t)$ for all $i \in \{1, \ldots, N\}$. This is useful in games related to economic markets, where $\omega_0(t)$ can represent a commonly known vector of current prices. Alternatively, one can imagine a game with a single random event process $\omega_0(t)$ that is known to the game manager but unknown to all players. For example, consider a game defined over a wireless multiple access system. Wireless users are players in the game, and the access point is the game manager. In this example, $\omega_0(t)$ can represent a vector of current channel conditions known only to the access point. Such games can be treated by setting $\omega_i(t)$ to a default constant value for all $i \in \{1, \ldots, N\}$ and all slots t.

B. Pure strategies and the virtual static game

Assume all players participate, so that $M_i(t) = \alpha_i(t)$ for all i. Like the previous section, the goal here is to define equilibrium conditions that ensure no player can benefit by individually deviating from the suggested actions of the game manager.

For each $i \in \{1, \ldots, N\}$, denote the sizes of sets Ω_i and $\mathcal{A}_i$ by $|\Omega_i|$ and $|\mathcal{A}_i|$, respectively. Define a *pure strategy function for player i* as a function $b_i(\omega_i)$ that maps Ω_i to the set $\mathcal{A}_i$. There are $|\mathcal{A}_i|^{|\Omega_i|}$ such functions. Define:

$$\mathcal{S}_i \triangleq \{1, 2, \ldots, |\mathcal{A}_i|^{|\Omega_i|}\}$$

Enumerate the pure strategy functions for player i and represent them by $b_i^{(s)}(\omega_i)$ for $s \in \mathcal{S}_i$. Define:

$$\mathcal{S} \triangleq \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_N$$

Each vector $(s_1, s_2, \ldots, s_N) \in \mathcal{S}$ can be used to specify a profile of pure strategies used by each player. For each $\boldsymbol{s} \in \mathcal{S}$ and each $\boldsymbol{\omega} \in \Omega$, define:

$$\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = (b_1^{(s_1)}(\omega_1), b_2^{(s_2)}(\omega_2), \ldots, b_N^{(s_N)}(\omega_N))$$

The vector $\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega})$ is equal to $(\alpha_1(t), \ldots, \alpha_N(t))$ in the special case when $\boldsymbol{\omega}(t) = \boldsymbol{\omega}$ and when the action of each player i on slot t is defined by pure strategy s_i. The average utility earned by player i on such a slot t is defined:

$$h_i(\boldsymbol{s}) \triangleq \textstyle\sum_{\boldsymbol{\omega} \in \Omega} \pi[\boldsymbol{\omega}] \hat{u}_i(\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}), \boldsymbol{\omega}) \tag{10}$$

The stochastic game can be transformed into a *virtual static game* as follows: The virtual static game also has N players. The *virtual action space* of each player i is viewed as the set of pure strategies $\mathcal{S}_i$. The *virtual utility functions* are given by the functions $h_i(\boldsymbol{s})$.

As the virtual action spaces of the players remain finite, the virtual static game is indeed a finite game. Hence, the NE, CE, and CCE definitions specified for static games in the previous section can be used for this virtual static game. This provides a natural path for extending these equilibrium concepts to the stochastic context. In particular, let $Pr[\boldsymbol{s}]$ be a probability mass function over the finite set of strategy profiles $\boldsymbol{s} \in \mathcal{S}$. This generates a conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ defined over all $\boldsymbol{\alpha} \in \mathcal{A}$ and $\boldsymbol{\omega} \in \Omega$:

$$Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] = \textstyle\sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] 1\{\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}\} \tag{11}$$

where $1\{\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}\}$ is an indicator function that is 1 if $\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}$, and is 0 else. The next lemma shows that every conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ can be generated in this way (proof in [16]).

Lemma 1: Let $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ be a conditional probability mass function defined over $(\boldsymbol{\alpha}, \boldsymbol{\omega}) \in \mathcal{A} \times \Omega$. Then there exists a probability mass function $Pr[\boldsymbol{s}]$, defined over $\boldsymbol{s} \in \mathcal{S}$, for which (11) holds.

Now suppose $Pr[\boldsymbol{s}]$ is a CCE of the virtual static game. By definition of CCE:

$$\begin{aligned} \textstyle\sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] h_i(\boldsymbol{s}) \geq \textstyle\sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] h_i(r_i, \boldsymbol{s}_{\bar{i}}) \\ \forall i \in \{1, \ldots, N\}, \forall r_i \in \mathcal{S}_i \end{aligned} \tag{12}$$

Lemma 2: If $Pr[\boldsymbol{s}]$ and $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ are probability mass functions that satisfy (11), then $Pr[\boldsymbol{s}]$ satisfies the CCE constraints (12) for the virtual static game if and only if $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ satisfies the following constraints:

$$\begin{aligned} &\sum_{\boldsymbol{\omega} \in \Omega} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \pi[\boldsymbol{\omega}] Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] \hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) \\ &\quad \geq \sum_{\boldsymbol{\omega} \in \Omega} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \pi[\boldsymbol{\omega}] Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] \hat{u}_i \left((b_i^{(s)}(\omega_i), \boldsymbol{\alpha}_{\bar{i}}), \boldsymbol{\omega} \right) \\ &\quad \forall i \in \{1, \ldots, N\}, \forall s \in \mathcal{S}_i \end{aligned} \tag{13}$$

Proof: It suffices to show that the left-hand-side of (12) is equal to the left-hand-side of (13), and that the same is true of the right-hand-sides. The following two identities are useful. Fix $i \in \{1, \ldots, N\}$, $\boldsymbol{\omega} \in \Omega$, $\boldsymbol{s} \in \mathcal{S}$. Then:

$$\hat{u}_i(\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}), \boldsymbol{\omega}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} 1\{\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}\} \hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) \tag{14}$$

Fix $r_i \in \mathcal{S}_i$, and define $\boldsymbol{s}' = (r_i, \boldsymbol{s}_{\bar{i}})$, being the strategy profile that replaces the ith component of $\boldsymbol{s}$ with strategy r_i. Then:

$$\begin{aligned} &\hat{u}_i(\boldsymbol{b}^{(\boldsymbol{s}')}(\boldsymbol{\omega}), \boldsymbol{\omega}) \\ &\quad = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} 1\{\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}\} \hat{u}_i((b_i^{(r_i)}(\omega_i), \boldsymbol{\alpha}_{\bar{i}}), \boldsymbol{\omega}) \end{aligned} \tag{15}$$

Now consider the left-hand-side of (12):

$$\begin{aligned} &\sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] h_i(\boldsymbol{s}) \\ &= \sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] \sum_{\boldsymbol{\omega} \in \Omega} \pi[\boldsymbol{\omega}] \hat{u}_i(\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}), \boldsymbol{\omega}) && (16) \\ &= \sum_{\boldsymbol{s} \in \mathcal{S}} Pr[\boldsymbol{s}] \sum_{\boldsymbol{\omega} \in \Omega} \pi[\boldsymbol{\omega}] \sum_{\boldsymbol{\alpha} \in \mathcal{A}} 1\{\boldsymbol{b}^{(\boldsymbol{s})}(\boldsymbol{\omega}) = \boldsymbol{\alpha}\} \hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) && (17) \\ &= \sum_{\boldsymbol{\omega} \in \Omega} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \pi[\boldsymbol{\omega}] Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] \hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) && (18) \end{aligned}$$

where (16) follows by (10), (17) follows by (14), and (18) follows by (11). This proves the left-hand-sides are equal. A similar argument proves the right-hand-sides are equal. ■

C. Equilibrium for the stochastic game

Let $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ be a conditional probability mass function defined over $\boldsymbol{\omega} \in \Omega$, $\boldsymbol{\alpha} \in \mathcal{A}$. It is assumed throughout that:

$$Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] \geq 0 \quad \forall \boldsymbol{\alpha} \in \mathcal{A}, \forall \boldsymbol{\omega} \in \Omega \tag{19}$$

$$\textstyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}} Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] = 1 \qquad \forall \boldsymbol{\omega} \in \Omega \tag{20}$$

Lemma 2 suggests the following definition: A conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ is a *coarse correlated* *equilibrium (CCE) for the stochastic game* if constraints (13) are satisfied for all $i \in \{1, \dots, N\}$ and all $s \in \mathcal{S}_i$.

Definitions of NE and CE can be similarly extended to this stochastic context. A probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ is a *mixed strategy Nash equilibrium (NE) for the stochastic game* if it has the product form:

$$Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] = \prod_{i=1}^{N} Pr[\alpha_i|\omega_i]$$

and if the following constraints are satisfied:

$$\begin{aligned}&\sum_{\boldsymbol{\omega}\in\Omega}\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \pi[\boldsymbol{\omega}] \prod_{i=1}^{N} Pr[\alpha_i|\omega_i]\hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) \\ &\quad \geq \sum_{\boldsymbol{\omega}\in\Omega}\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \pi[\boldsymbol{\omega}] \prod_{i=1}^{N} Pr[\alpha_i|\omega_i]\hat{u}_i\left((b_i^{(s)}(\omega_i), \boldsymbol{\alpha}_{\bar{i}}), \boldsymbol{\omega}\right) \\ &\quad \forall i \in \{1, \dots, N\}, \forall s \in \mathcal{S}_i \end{aligned} \tag{21}$$

A conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ is a *correlated equilibrium (CE) for the stochastic game* if:

$$\begin{aligned}&\sum_{\boldsymbol{\omega}\in\Omega}\sum_{\boldsymbol{\alpha}\in\mathcal{A}|\alpha_i=c_i} \pi[\boldsymbol{\omega}] Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]\hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) \\ &\geq \sum_{\boldsymbol{\omega}\in\Omega}\sum_{\boldsymbol{\alpha}\in\mathcal{A}|\alpha_i=c_i} \pi[\boldsymbol{\omega}] Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]\hat{u}_i\left((b_i^{(s)}(\omega_i), \boldsymbol{\alpha}_{\bar{i}}), \boldsymbol{\omega}\right) \\ &\quad \forall i \in \{1, \dots, N\}, \forall s \in \mathcal{S}_i, \forall c_i \in \mathcal{A}_i \end{aligned} \tag{22}$$

Let $\mathcal{E}_{NE}^{stoc}$, $\mathcal{E}_{CE}^{stoc}$, $\mathcal{E}_{CCE}^{stoc}$ be the set of all conditional probability mass functions $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ that are NE, CE, and CCE, respectively, for the stochastic game.

Lemma 3: For the general stochastic game defined above:
(a) The set $\mathcal{E}_{NE}^{stoc}$ is nonempty.
(b) $\mathcal{E}_{NE}^{stoc} \subseteq \mathcal{E}_{CE}^{stoc} \subseteq \mathcal{E}_{CCE}^{stoc}$.
(c) Sets $\mathcal{E}_{CE}^{stoc}$ and $\mathcal{E}_{CCE}^{stoc}$ are closed, bounded, and convex.

Proof: See [16]. ■

As in the static case, it can be shown that the CCE constraints (13) imply that no player can benefit by individually choosing not to participate (assuming non-participants do not receive the messages from the game manager) [16]. Likewise, the CE constraints (22) imply that no player can benefit by individually choosing not to participate (assuming non-participants still receive the messages).

D. Optimization objective

As before, define $\phi(u_1, \dots, u_N)$ as a continuous and concave function that maps $\times_{i=1}^{N}[0, u_i^{max}]$ to the set of real numbers. The goal is to choose messages $\boldsymbol{M}(t) = \boldsymbol{\alpha}(t)$ according to a conditional probability mass function $Pr[\boldsymbol{\alpha}(t)|\boldsymbol{\omega}(t)]$ that solves the problem below:

$$\text{Maximize:} \quad \phi(\overline{u}_1, \dots, \overline{u}_N) \tag{23}$$

$$\text{Subject to:} \quad \overline{u}_i = \textstyle\sum_{\boldsymbol{\omega}\in\Omega}\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \pi[\boldsymbol{\omega}]Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]\hat{u}_i(\boldsymbol{\alpha}, \boldsymbol{\omega}) \quad \forall i \in \{1, \dots, N\} \tag{24}$$

$$\text{CCE constraints (13) are satisfied} \tag{25}$$

$$Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] \geq 0 \ \forall \boldsymbol{\alpha} \in \mathcal{A}, \boldsymbol{\omega} \in \Omega \tag{26}$$

$$\textstyle\sum_{\boldsymbol{\alpha}\in\mathcal{A}} Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}] = 1 \ \forall \boldsymbol{\omega} \in \Omega \tag{27}$$

This is a convex program in the unknowns $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$. Section IV presents an online solution technique that does not require knowledge of the probabilities $\pi[\boldsymbol{\omega}]$.

IV. Lyapunov optimization

For a real-valued stochastic process $u(t)$ defined over slots $t \in \{0, 1, 2, \dots\}$, define:

$$\overline{u}(t) \triangleq \tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1} \mathbb{E}\left[u(\tau)\right]$$

Recall that $u_i(t) \triangleq \hat{u}_i(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$. For each $i \in \{1, \dots, N\}$, define $u_i^{(s)}(t) \triangleq \hat{u}_i^{(s)}(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$, where $\hat{u}_i^{(s)}(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$ is the corresponding utility when the player i action is replaced by the action of the pure strategy $b_i^{(s)}(\omega_i)$:

$$\hat{u}_i^{(s)}(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t)) \triangleq \hat{u}_i\left(\left(b_i^{(s)}(\omega_i(t)), \boldsymbol{\alpha}_{\bar{i}}(t)\right), \boldsymbol{\omega}(t)\right)$$

Consider the following modification of the problem (23)-(27): Every slot $t \in \{0, 1, 2, \dots\}$ the game manager observes $\boldsymbol{\omega}(t)$ and chooses an action vector $\boldsymbol{\alpha}(t) \in \mathcal{A}$ to solve:

Maximize:

$$\liminf_{t\rightarrow\infty} \phi(\overline{u}_1(t), \dots, \overline{u}_N(t)) \tag{28}$$

Subject to:

$$\liminf_{t\rightarrow\infty} [\overline{u}_i(t) - \overline{u}_i^{(s)}(t)] \geq 0 \ \forall i \in \{1, \dots, N\}, \forall s \in \mathcal{S}_i \tag{29}$$

$$\boldsymbol{\alpha}(t) \in \mathcal{A} \ \ \forall t \in \{0, 1, 2, \dots\} \tag{30}$$

It can be shown that optimality can be achieved by a stationary and randomized algorithm that observes $\boldsymbol{\omega}(t)$ and independently chooses $\boldsymbol{\alpha}(t)$ according to the same conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ every slot. Such algorithms yield well defined limits. Any probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ that solves (23)-(27) must also solve (28)-(30). Moreover, any solution to (28)-(30) must have time average expectations that are arbitrarily close to conditional probability mass functions $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ that solve (23)-(27).

A. Transformation via Jensen's inequality

Using the auxiliary variable technique of [7], the problem (28)-(30), which maximizes a nonlinear function of a time average, can be transformed into a maximization of the time average of a nonlinear function. To this end, let $\boldsymbol{\gamma}(t) = (\gamma_1(t), \dots, \gamma_N(t))$ be an *auxiliary vector* that the game manager chooses on slot t, assumed to satisfy $0 \leq \gamma_i(t) \leq u_i^{max}$ for all t and all i. Define:

$$g(t) \triangleq \phi(\gamma_1(t), \dots, \gamma_N(t))$$

Jensen's inequality implies that for all slots $t > 0$:

$$\overline{g}(t) \leq \phi(\overline{\gamma}_1(t), \dots, \overline{\gamma}_N(t)) \tag{31}$$

Now consider the following problem: Every slot $t \in \{0, 1, 2, \dots\}$ the game manager observes $\boldsymbol{\omega}(t)$ and chooses both an action vector $\boldsymbol{\alpha}(t) \in \mathcal{A}$ and an auxiliary vector $\boldsymbol{\gamma}(t)$

to solve:

Maximize:

$$\liminf_{t\rightarrow\infty} \overline{g}(t) \tag{32}$$

Subject to:

$$\lim_{t\rightarrow\infty} |\overline{\gamma}_i(t) - \overline{u}_i(t)| = 0 \ \forall i \in \{1,\ldots,N\} \tag{33}$$

$$\liminf_{t\rightarrow\infty}[\overline{u}_i(t) - \overline{u}_i^{(s)}(t)] \geq 0 \ \forall i \in \{1,\ldots,N\}, \forall s \in \mathcal{S}_i \tag{34}$$

$$\boldsymbol{\alpha}(t) \in \mathcal{A} \ \forall t \tag{35}$$

$$0 \leq \gamma_i(t) \leq u_i^{max} \ \ \forall t, \forall i \in \{1,\ldots,N\} \tag{36}$$

For intuition, suppose all limits exist, so that constraint (33) is equivalent to $\overline{\gamma}_i = \overline{u}_i$. This together with Jensen's inequality (31) ensures the optimal value of the objective function in the above problem is less than or equal to that of the problem (28)-(30). On the other hand, the optimal value of (28)-(30) can be *achieved* by choosing $\gamma_i(t) = \overline{u}_i^*$ for all t, where $(\overline{u}_1^*, \ldots, \overline{u}_N^*)$ are optimal time average utilities for problem (28)-(30). The problems (28)-(30) and (32)-(36) are equivalent.

B. The drift-plus-penalty algorithm

The problem (32)-(36) can be solved via the *drift-plus-penalty algorithm* of [7]. To enforce the constraints (34), define a *virtual queue* $Q_i^{(s)}(t)$ for all $i \in \{1,\ldots,N\}$ and all $s \in \mathcal{S}$, with update equation:

$$Q_i^{(s)}(t+1) = \max[Q_i^{(s)}(t) + u_i^{(s)}(t) - u_i(t), 0] \tag{37}$$

The above looks like a slotted time queueing equation with arrival process $u_i^{(s)}(t)$ and service process $u_i(t)$. The intuition is that if a control algorithm is constructed that makes these queues *mean rate stable*, so that:

$$\lim_{t\rightarrow\infty} \frac{\mathbb{E}\left[Q_i^{(s)}(t)\right]}{t} = 0$$

then constraint (34) is satisfied [7]. Likewise, to enforce the constraints (33), define a virtual queue $Z_i(t)$ for all $i \in \{1,\ldots,N\}$, with update equation:

$$Z_i(t+1) = Z_i(t) + \gamma_i(t) - u_i(t) \tag{38}$$

For simplicity, assume all virtual queues are initialized to 0.

Define $L(t)$ as a sum of squares of all virtual queues (divided by 2 for convenience):

$$L(t) \triangleq \tfrac{1}{2}\textstyle\sum_{i=1}^N \sum_{s\in\mathcal{S}_i} Q_i^{(s)}(t)^2 + \tfrac{1}{2}\sum_{i=1}^N Z_i(t)^2$$

This is called a *Lyapunov function*. Define $\Delta(t) \triangleq L(t+1) - L(t)$, called the *Lyapunov drift on slot* t. The drift-plus-penalty algorithm is defined by choosing control actions greedily every slot to minimize a bound on the *drift-plus-penalty expression* $\Delta(t) - Vg(t)$. Here, $-g(t)$ is a "penalty" and V is a non-negative constant that affects a tradeoff between convergence time and proximity to the optimal solution.

Lemma 4: For all slots t one has:

$$\begin{aligned}\Delta(t) - Vg(t) \ \ \leq \ \ & B - Vg(t) \\ & + \sum_{i=1}^N \sum_{s\in\mathcal{S}_i} Q_i^{(s)}(t)[u_i^{(s)}(t) - u_i(t)] \\ & + \sum_{i=1}^N Z_i(t)[\gamma_i(t) - u_i(t)]\end{aligned} \tag{39}$$

where:

$$B \triangleq \tfrac{1}{2}\textstyle\sum_{i=1}^N \sum_{s\in\mathcal{S}_i}(u_i^{max})^2 + \tfrac{1}{2}\sum_{i=1}^N (u_i^{max})^2$$

Proof: The result follows immediately from the fact that $\max[x,0]^2 \leq x^2$ (see details in [16]). ■

Greedily minimizing the right-hand-side of (39) every slot leads to the following algorithm: Every slot t, the game manager observes the queues and the current $\boldsymbol{\omega}(t)$. Then:

- Auxiliary variables: Choose $\gamma_i(t) \in [0, u_i^{max}]$ for all $i \in \{1,\ldots,N\}$ to maximize:

  $$V\phi(\gamma_1(t),\ldots,\gamma_N(t)) - \textstyle\sum_{i=1}^N Z_i(t)\gamma_i(t)$$

- Suggested actions: Choose $\boldsymbol{\alpha}(t) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ to minimize:

  $$\begin{aligned}& -\sum_{i=1}^N Z_i(t)\hat{u}_i(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) \\ & + \sum_{i=1}^N \sum_{s\in\mathcal{S}_i} Q_i^{(s)}(t)[\hat{u}_i^{(s)}(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) - \hat{u}_i(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t))]\end{aligned}$$

  Then send suggested actions $\alpha_i(t)$ to each (participating) player $i \in \{1,\ldots,N\}$.
- Update virtual queues via (37) and (38).

This is an online algorithm that does not require knowledge of the probabilities $\pi[\boldsymbol{\omega}]$.

C. Performance analysis

Define ϕ^* as the optimal value of the objective function for problem (23)-(27), and note by equivalence of the transformations that this is also the optimal value for the problem (32)-(36). Define $\boldsymbol{\theta}(t)$ as the vector of all virtual queues $Q_i^{(s)}(t)$ and $Z_i(t)$, and define $||\boldsymbol{\theta}|| \triangleq \sqrt{2L(t)}$.

Theorem 1: If the above algorithm is implemented using a fixed value $V > 0$, then:

(a) For all slots $t > 0$ one has:

$$\phi\left(\overline{\gamma}_1(t),\ldots,\overline{\gamma}_N(t)\right) \geq \phi^* - B/V$$

(b) All virtual queues $Q_i^{(s)}(t)$ and $Z_i(t)$ are mean rate stable, so that the constraints (33)-(36) are satisfied.

(c) $\frac{\mathbb{E}[||\boldsymbol{\theta}(t)||]}{t} \leq \sqrt{\frac{2B+2V(g_{max}-\phi^*)}{t}}$ for all slots $t > 0$, where g_{max} is the maximum possible value for $g(t)$, being the maximum of $\phi(\gamma_1,\ldots,\gamma_N)$ over $\gamma_i \in [0, u_i^{max}]$ for all $i \in \{1,\ldots,N\}$.

Proof: The algorithm minimizes the right-hand-side of (39) every slot t, and so:

$$\begin{aligned}\Delta(t) - Vg(t) \quad &\leq \quad B - V\phi(\gamma_1^*(t), \ldots, \gamma_N^*(t)) \\ &\quad + \sum_{i=1}^{N} \sum_{s\in\mathcal{S}_i} Q_i^{(s)}(t)[u_i^{*(s)}(t) - u_i^*(t)] \\ &\quad + \sum_{i=1}^{N} Z_i(t)[\gamma_i^*(t) - u_i^*(t)] \qquad (40)\end{aligned}$$

for all alternative decisions $\boldsymbol{\alpha}^*(t) \in \mathcal{A}$ and $\boldsymbol{\gamma}^*(t)$ that satisfy $0 \leq \gamma_i^*(t) \leq u_i^{max}$, where:

$$\begin{aligned} u_i^*(t) \quad &\triangleq \quad \hat{u}_i(\boldsymbol{\alpha}^*(t), \boldsymbol{\omega}(t)) \\ u_i^{*(s)}(t) \quad &\triangleq \quad \hat{u}_i\left(\left(b_i^{(s)}(\omega_i(t)), \boldsymbol{\alpha}_{\bar{i}}^*(t)\right), \boldsymbol{\omega}(t)\right)\end{aligned}$$

Now randomly choose $\boldsymbol{\alpha}^*(t)$ as a function of $\boldsymbol{\omega}(t)$ (and independently of queue backlog) according to the probabilities $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$ that solve (23)-(27). Let u_i^* be the expected utility of player i under this distribution, and note that:

$$\phi(u_1^*, \ldots, u_N^*) = \phi^*$$

Choose $\gamma_i(t) = u_i^*$ for $i \in \{1, \ldots, N\}$. Taking expectations of (40) then gives:

$$\mathbb{E}\left[\Delta(t) - Vg(t)\right] \quad \leq \quad B - V\phi^*$$

Fix a slot $T > 0$. Summing the above over slots $t \in \{0, 1, 2, \ldots, T-1\}$ and using $L(0) = 0$ gives:

$$\mathbb{E}\left[L(T)\right] - V\sum_{t=0}^{T-1} \mathbb{E}\left[g(t)\right] \leq BT - V\phi^* T \qquad (41)$$

Rearranging (41) and using the definition of $g(t)$ gives:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\phi(\gamma_1(t), \ldots, \gamma_N(t))\right] \geq \phi^* - \frac{B}{V} + \frac{\mathbb{E}\left[L(T)\right]}{VT}$$

Using Jensen's inequality and $\mathbb{E}\left[L(T)\right] \geq 0$ proves part (a).

Again rearranging (41) gives:

$$\mathbb{E}\left[||\boldsymbol{\theta}(T)||^2\right] \leq 2BT + 2VT(g_{max} - \phi^*)$$

Using the fact that $\mathbb{E}\left[||\boldsymbol{\theta}(T)||\right]^2 \leq \mathbb{E}\left[||\boldsymbol{\theta}(T)||^2\right]$, dividing by T^2, and taking square roots proves part (c). Part (b) follows immediately from part (c). ■

Define $\epsilon = 1/V$. Theorem 1 shows that average utility is within $O(\epsilon)$ of optimality. Part (c) of the theorem implies that constraint violation is within $O(\epsilon)$ after time $O(1/\epsilon^3)$. If a *Slater condition* holds, this convergence time is improved to $O(1/\epsilon^2)$ [7]. Similar bounds can be shown for infinite horizon time averages (rather than time average expectations) [7].

D. Discussion

The online algorithm ensures the constraints (34) are satisfied. This shows that average utility of each player i is greater than or equal to the achievable utility if the player were to constantly use some other pure strategy. This corresponds to the constraint in (13). If an algorithm makes random decisions independently every slot according to a conditional probability mass function $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$, then constraint (13) implies player i cannot do better under *any* alternative decisions, possibly those that mix pure strategies with different mixing probabilities every slot. A subtlety is that the online algorithm does *not* make stationary and randomized decisions. Thus, it is not clear if a player with knowledge of the algorithm could improve average utility by making alternative decisions that do not correspond to a pure strategy. Of course, the online algorithm yields time averages that correspond to a desired $Pr[\boldsymbol{\alpha}|\boldsymbol{\omega}]$. Thus, a potential fix is to run the online algorithm in the background and make $\boldsymbol{\alpha}(t)$ decisions according to the time averages that emerge. A faster method might be to modify the action choice selection by using time averages of the $Z_i(t)$ and $Q_i^{(s)}(t)$ values, rather than their instantaneous values.

Acknowledgement

This work is supported in part by: NSF grant CCF-0747525, the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Lab W911NF-09-2-0053.

References

[1] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, vol. 8, no. 1 pp. 33-37, Jan.-Feb. 1997.

[2] H. Moulin and J. P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, vol. 7, no. 3/4, pp. 201-221, 1978.

[3] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, vol. 1, pp. 67-96, 1974.

[4] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, vol. 55, pp. 1-18, 1987.

[5] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[6] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, vol. 54, pp. 286-295, 1951.

[7] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[8] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[9] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466-478, March 1993.

[10] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396-409, April 2008.

[11] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1333-1344, Dec. 2007.

[12] A. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, vol. 54, no. 3, pp. 203-220, 2006.

[13] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 14, no. 8, Aug. 2006.

[14] E. Solan and N. Vieille. Correlated equilibrium in stochastic games. *Games and Economic Behavior*, vol. 38, pp. 362-399, 2002.

[15] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, pp. 48-49, 1950.

[16] M. J. Neely. A Lyapunov optimization approach to repeated stochastic games. *ArXiv technical report*, Oct. 2013.