A Lyapunov Optimization Approach to Repeated Stochastic Games

Shengkai Shi (HKU)

December 20, 2013

Overview

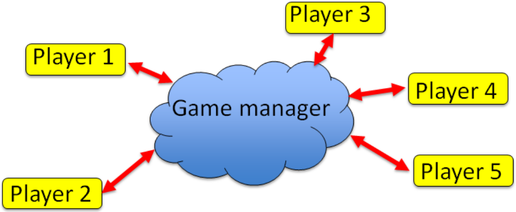


Figure: Game Structure.

Game Structure

- Slotted time t in $\{0, 1, 2, \ldots\}$.

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$
  - *Random Events* $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$
  - *Random Events* $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$
- Players maximize time average utility.

Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$
  - *Random Events* $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$
- Players maximize time average utility.
- Game manager

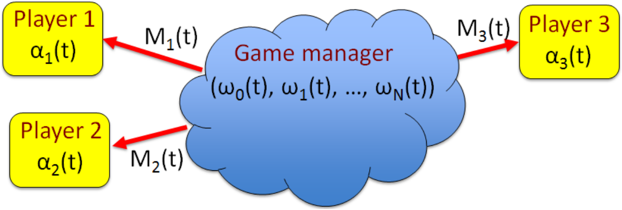
Game Structure

- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$
  - *Random Events* $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$
- Players maximize time average utility.
- Game manager
  - Provides suggestions.

Game Structure

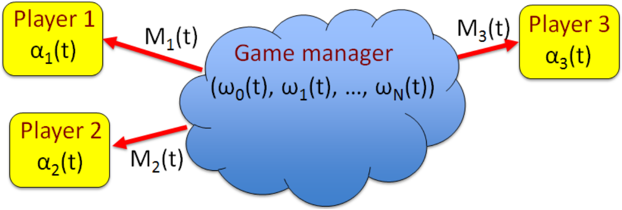
- Slotted time t in $\{0, 1, 2, ...\}$.
- N players, 1 game manager.
- Slot t utility for each player depends on:
  - *Control Actions* $\alpha(t) = (\alpha_1(t), ..., \alpha_N(t))$
  - *Random Events* $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$
- Players maximize time average utility.
- Game manager
  - Provides suggestions.
  - Maintains fairness of utilities subject to equilibrium constraints.

Actions and utilities



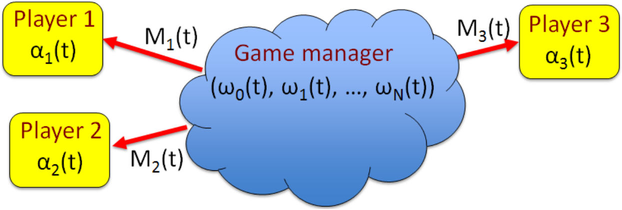
- Game manager sends suggested actions $(M_1(t), ..., M_N(t))$.

Actions and utilities



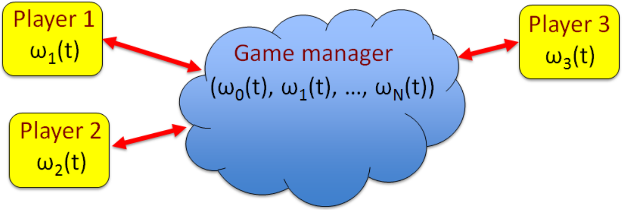
- Game manager sends suggested actions $(M_1(t), ..., M_N(t))$.
- Players take actions in A_i.

Actions and utilities



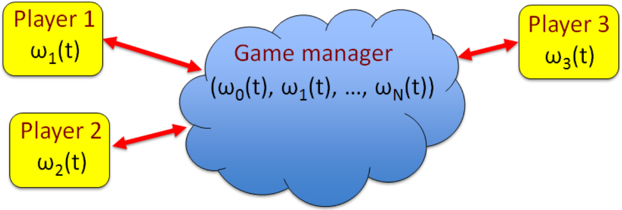
- Game manager sends suggested actions $(M_1(t), ..., M_N(t))$.
- Players take actions in A_i.
- $U_i(t) = u_i(\alpha(t), \omega(t))$.

Random events



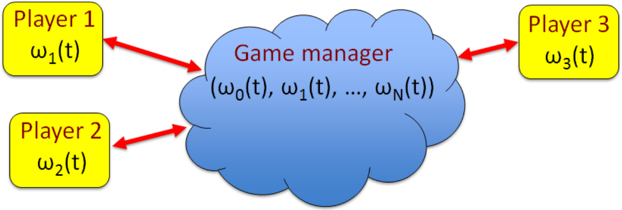
- Game manager sees $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.

Random events



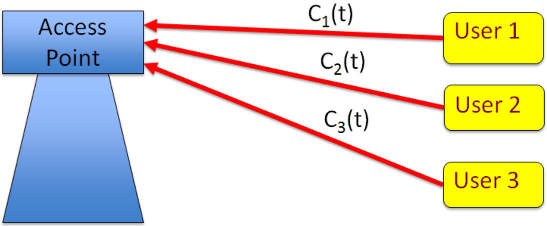
- Game manager sees $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.
- Player i sees $\omega_i(t)$.

Random events



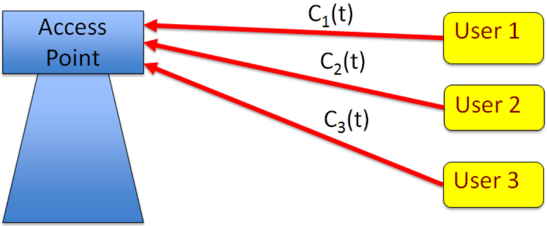
- Game manager sees $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.
- Player i sees $\omega_i(t)$.
- $\omega(t)$ is i.i.d. over slots.

Example: Wireless MAC game



- Manager knows current channel conditions: $\omega_0(t) = (C_1(t), ..., C_N(t))$.

Example: Wireless MAC game



- Manager knows current channel conditions: $\omega_0(t) = (C_1(t), ..., C_N(t))$.
- Users do not have this knowledge: $\omega_i(t) = NULL$.

Participation

- At beginning of game, players choose either:

Participation

- At beginning of game, players choose either:
  - Participate:

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:
    - Do not receive messages $M_i(t)$.

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:
    - Do not receive messages $M_i(t)$.
    - Choose $\alpha_i(t)$ however they like.

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:
    - Do not receive messages $M_i(t)$.
    - Choose $\alpha_i(t)$ however they like.
- Need incentives for participation.

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:
    - Do not receive messages $M_i(t)$.
    - Choose $\alpha_i(t)$ however they like.
- Need incentives for participation.
  - Nash Equilibrium (NE)

Participation

- At beginning of game, players choose either:
    - Participate:
        - Receive messages $M_i(t)$.
        - Always choose $\alpha_i(t) = M_i(t)$.
    - Do not participate:
        - Do not receive messages $M_i(t)$.
        - Choose $\alpha_i(t)$ however they like.
- Need incentives for participation.
    - Nash Equilibrium (NE)
    - Correlated Equilibrium (CE)

Participation

- At beginning of game, players choose either:
  - Participate:
    - Receive messages $M_i(t)$.
    - Always choose $\alpha_i(t) = M_i(t)$.
  - Do not participate:
    - Do not receive messages $M_i(t)$.
    - Choose $\alpha_i(t)$ however they like.
- Need incentives for participation.
  - Nash Equilibrium (NE)
  - Correlated Equilibrium (CE)
  - Coarse Correlated Equilibrium (CCE)

NE for static game

- Consider special case with no random process $\omega(t)$.

NE for static game

- Consider special case with no random process $\omega(t)$.
- Nash Equilibrium (NE):

NE for static game

- Consider special case with no random process $\omega(t)$.
- Nash Equilibrium (NE):
  - Players actions are independent: $Pr[\alpha] = Pr[\alpha_1]Pr[\alpha_2]...Pr[\alpha_N]$.

NE for static game

- Consider special case with no random process $\omega(t)$.
- Nash Equilibrium (NE):
    - Players actions are independent: $Pr[\alpha] = Pr[\alpha_1]Pr[\alpha_2]...Pr[\alpha_N]$.
    - Game manager not needed.

NE for static game

- Consider special case with no random process $\omega(t)$.
- Nash Equilibrium (NE):
  - Players actions are independent: $Pr[\alpha] = Pr[\alpha_1]Pr[\alpha_2]...Pr[\alpha_N]$.
  - Game manager not needed.

NE for static game

- Consider special case with no random process $\omega(t)$.
- Nash Equilibrium (NE):
  - Players actions are independent: $Pr[\alpha] = Pr[\alpha_1]Pr[\alpha_2]...Pr[\alpha_N]$.
  - Game manager not needed.

Definition

Distribution $Pr[\alpha]$ is a Nash Equilibrium (NE) if no player can benefit by unilaterally changing its action probabilities.

CE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.

CE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Suppose all players participate.

CE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Suppose all players participate.

CE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Suppose all players participate.

Definition

Distribution $Pr[\alpha]$ is a Correlated Equilibrium (CE) if for all players $i \in \{1, 2, ..., N\}$ and for all actions $\alpha_i, \beta_i \in A_i$:

$$\sum_{\alpha_{\bar{i}} \in A_{\bar{i}}} u_i(\alpha_i, \alpha_{\bar{i}}) Pr[\alpha_i, \alpha_{\bar{i}}] \geq \sum_{\alpha_{\bar{i}} \in A_{\bar{i}}} u_i(\beta_i, \alpha_{\bar{i}}) Pr[\alpha_i, \alpha_{\bar{i}}] \tag{1}$$

CCE for static game

- Manager suggests actions $\alpha(t) \to Pr[\alpha]$.

CCE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Gives suggestions only to participating players.

CCE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Gives suggestions only to participating players.
- Suppose all players participate.

CCE for static game

- Manager suggests actions $\alpha(t) \rightarrow Pr[\alpha]$.
- Gives suggestions only to participating players.
- Suppose all players participate.

CCE for static game

- Manager suggests actions $\alpha(t) \to Pr[\alpha]$.
- Gives suggestions only to participating players.
- Suppose all players participate.

Definition

Distribution $Pr[\alpha]$ is a Coarse Correlated Equilibrium (CCE) if for all players $i \in \{1, 2, ..., N\}$ and for all actions $\beta_i \in A_i$:

$$\sum_{\alpha \in A} u_i(\alpha) Pr[\alpha] \geq \sum_{\alpha \in A} u_i(\beta_i, \alpha_{\bar{i}}) Pr[\alpha] \quad (2)$$

Superset theorem

Theorem

$\{all\ NE\} \subseteq \{all\ CE\} \subseteq \{all\ CCE\}$

- The NE, CE, CCE definitions extend easily to the stochastic game.

Random events

- Random events $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.

Random events

- Random events $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.
- ω_i takes values in some finite set Ω_i.

Random events

- Random events $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.
- ω_i takes values in some finite set Ω_i.
- $\omega(t)$ is i.i.d. over slots with probability mass function: $\pi[\omega] = Pr[\omega(t) = \omega], \forall \omega \in \Omega_0 \times \Omega_1 \times \cdots \times \Omega_N$.

Random events

- Random events $\omega(t) = (\omega_0(t), \omega_1(t), ..., \omega_N(t))$.
- ω_i takes values in some finite set Ω_i.
- $\omega(t)$ is i.i.d. over slots with probability mass function: $\pi[\omega] = Pr[\omega(t) = \omega], \forall \omega \in \Omega_0 \times \Omega_1 \times \cdots \times \Omega_N$.
- $U_i(t) = u_i(\alpha(t), \omega(t))$.

Pure strategies for stochastic games

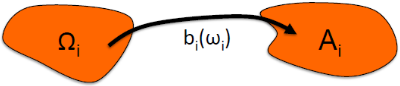
- Player i observes: $\omega_i \in \Omega_i$.

Pure strategies for stochastic games

- Player i observes: $\omega_i \in \Omega_i$.
- Player i chooses: $\alpha_i \in A_i$.

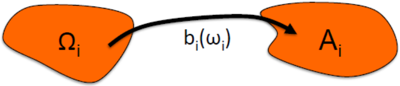
Pure strategies for stochastic games

- Player i observes: $\omega_i \in \Omega_i$.
- Player i chooses: $\alpha_i \in A_i$.
- A pure strategy for player i is a function b_i: $\Omega_i \to A_i$.



Pure strategies for stochastic games

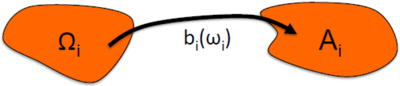
- Player i observes: $\omega_i \in \Omega_i$.
- Player i chooses: $\alpha_i \in A_i$.
- A pure strategy for player i is a function b_i: $\Omega_i \to A_i$.



- There are $|A_i|^{|\Omega_i|}$ pure strategies for player i.

Pure strategies for stochastic games

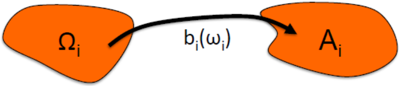
- Player i observes: $\omega_i \in \Omega_i$.
- Player i chooses: $\alpha_i \in A_i$.
- A pure strategy for player i is a function b_i: $\Omega_i \to A_i$.



- There are $|A_i|^{|\Omega_i|}$ pure strategies for player i.
- Define S_i as this set of pure strategies.

Pure strategies for stochastic games

- Player i observes: $\omega_i \in \Omega_i$.
- Player i chooses: $\alpha_i \in A_i$.
- A pure strategy for player i is a function b_i: $\Omega_i \to A_i$.



- There are $|A_i|^{|\Omega_i|}$ pure strategies for player i.
- Define S_i as this set of pure strategies.
- $\Omega = \Omega_0 \times \Omega_1 \times \cdots \times \Omega_N$ and $S = S_1 \times S_2 \times \cdots \times S_N$. For each $s \in S$ and each $\omega \in \Omega$,

$$b^{(s)}(\omega) = (b_1^{(s)}(\omega_1), b_2^{(s)}(\omega_2), \ldots, b_N^{(s_N)}(\omega_N)) \tag{3}$$

Virtual static game

- Virtual action space for player i: S_i.

Virtual static game

- Virtual action space for player i: S_i.
- Virtual utility function:

$$h_i^s = \sum_{\omega \in \Omega} \pi[\omega] u_i(b^{(s)}(\omega), \omega) \tag{4}$$

Virtual static game

- Virtual action space for player i: S_i.
- Virtual utility function:

$$h_i^s = \sum_{\omega \in \Omega} \pi[\omega] u_i(b^{(s)}(\omega), \omega) \tag{4}$$

- Probability mass function over the finite set of strategy profiles: $Pr[s], s \in S$.

CCE for virtual static game

- Suppose $Pr[s]$ is a CCE of the virtual static game, it should satisfy the following constraint:

$$\sum_{s \in S} h_i(s) Pr[s] \geq \sum_{s \in S} h_i(r_i, s_{\bar{i}}) Pr[s], \forall i \in \{1, ..., N\}, \forall r_i \in S_i \tag{5}$$

CCE for stochastic games

- Conditional probability mass function defined over all $\alpha \in A$ and $\omega \in \Omega$: $Pr[\alpha|\omega]$.

$$Pr[\alpha|\omega] = \sum_{s \in S} Pr[s] 1\{b^{(s)}(\omega) = \alpha\} \tag{6}$$

CCE for stochastic games

- Conditional probability mass function defined over all $\alpha \in A$ and $\omega \in \Omega$: $Pr[\alpha|\omega]$.

$$Pr[\alpha|\omega] = \sum_{s \in S} Pr[s] 1\{b^{(s)}(\omega) = \alpha\} \tag{6}$$

CCE for stochastic games

- Conditional probability mass function defined over all $\alpha \in A$ and $\omega \in \Omega$: $Pr[\alpha|\omega]$.

$$Pr[\alpha|\omega] = \sum_{s \in S} Pr[s] 1\{b^{(s)}(\omega) = \alpha\} \tag{6}$$

Definition

$Pr[\alpha|\omega]$ is a Coarse Correlated Equilibrium for the stochastic game if :

$$\sum_{\omega \in \Omega} \sum_{\alpha \in A} \pi[\omega] Pr[\alpha|\omega] u_i(\alpha, \omega)$$
$$\geq \sum_{\omega \in \Omega} \sum_{\alpha \in A} \pi[\omega] Pr[\alpha|\omega] u_i((b_i^s(\omega_i), \alpha_{\bar{i}}),$$
$$\forall i \in \{1, ..., N\}, \forall s \in S_i.$$

Optimization Objective

$$
\begin{aligned}
Maximize \quad & \phi(\bar{u_1}, ..., \bar{u_N}) \\
s.t. \quad & \bar{u}_i = \sum_{\omega \in \Omega} \sum_{\alpha \in A} \pi[\omega] Pr[\alpha|\omega] u_i(\alpha, \omega) \\
& CCE \;\; constraints \\
& Pr[\alpha|\omega] \geq 0, \forall \alpha \in A, \omega \in \Omega \\
& \sum_{\alpha \in A} Pr[\alpha|\omega] = 1, \forall \omega \in \Omega
\end{aligned}
$$

CCE constraints

- Formally, $u_i^{(s)}(\alpha(t), \omega(t)) = u_i((b_i^{(s)}(\omega_i(t)), \alpha_{\bar{i}}(t)), \omega(t))$.

CCE constraints

- Formally, $u_i^{(s)}(\alpha(t), \omega(t)) = u_i((b_i^{(s)}(\omega_i(t)), \alpha_{\bar{i}}(t)), \omega(t))$.
- CCE constraints:

$$\bar{u}_i \geq \bar{u}_i^{(s)}, \forall i \in \{1, ..., N\}, \forall s \in S_i. \tag{7}$$

Lyapunov optimization approach

$$
\begin{aligned}
\textit{Maximize} \quad & \liminf_{t\to\infty} \phi(\bar{u_1}(t), \ldots, \bar{u_N}(t)) \\
s.t. \quad & \liminf_{t\to\infty} [\bar{u}_i - \bar{u}_i{}^{(s)}] \geq 0, \forall i \in \{1, \ldots, N\}, \forall s \in S_i \\
& \alpha(t) \in A, \forall t \in \{0, 1, 2, \ldots\}
\end{aligned}
$$

Transformation via Jensen's inequality

- Auxiliary vector, $\gamma(t) = (\gamma_1(t), ..., \gamma_N(t))$, for all t and all i, satisfies $0 \leq \gamma_i(t) \leq u_i^{max}$.

Transformation via Jensen's inequality

- Auxiliary vector, $\gamma(t) = (\gamma_1(t), ..., \gamma_N(t))$, for all t and all i, satisfies $0 \leq \gamma_i(t) \leq u_i^{max}$.
- Define $g(t) = \phi(\gamma_1(t), ..., \gamma_N(t))$.

Transformation via Jensen's inequality

- Auxiliary vector, $\gamma(t) = (\gamma_1(t), ..., \gamma_N(t))$, for all t and all i, satisfies $0 \leq \gamma_i(t) \leq u_i^{max}$.
- Define $g(t) = \phi(\gamma_1(t), ..., \gamma_N(t))$.
- Jensen's inequality implies that for all t, $\bar{g}(t) \leq \phi(\bar{\gamma_1}, ..., \bar{\gamma_N})$.

Modified optimization problem

$$
\begin{aligned}
\textit{Maximize} \quad & \liminf_{t\to\infty} \bar{g}(t) \\
s.t. \quad & \liminf_{t\to\infty} |\bar{\gamma}_i(t) - \bar{u}_i(t)| = 0, \forall i \in \{1, ..., N\} \\
& \liminf_{t\to\infty} [\bar{u}_i - \bar{u}_i{}^{(s)}] \geq 0, \forall i \in \{1, ..., N\}, \forall s \in S_i \\
& \alpha(t) \in A, \forall t \in \{0, 1, 2, ...\} \\
& 0 \leq \gamma_i(t) \leq u_i^{max}, \forall t \in \{0, 1, 2, ...\}, \forall i \in \{1, ..., N\}
\end{aligned}
$$

Virtual queues

- Virtual queue $Q_i^{(s)}(t)$:

$$Q_i^{(s)}(t+1) = max[Q_i^{(s)}(t) + u_i^{(s)}(t) - u_i(t), 0] \qquad (8)$$

Virtual queues

- Virtual queue $Q_i^{(s)}(t)$:

$$Q_i^{(s)}(t+1) = max[Q_i^{(s)}(t) + u_i^{(s)}(t) - u_i(t), 0] \qquad (8)$$

- Virtual queue $Z_i(t)$:

$$Z_i(t+1) = Z_i(t) + \gamma_i(t) - u_i(t) \qquad (9)$$

Drift-plus-penalty expression

- Lyapunov function:

$$L(t) = \frac{1}{2}\sum_{i=1}^{N}\sum_{s\in S_i} Q_i^{(s)}(t)^2 + \frac{1}{2}\sum_{i=1}^{N} Z_i(t)^2 \tag{10}$$

Drift-plus-penalty expression

- Lyapunov function:

$$L(t) = \frac{1}{2}\sum_{i=1}^{N}\sum_{s\in S_i} Q_i^{(s)}(t)^2 + \frac{1}{2}\sum_{i=1}^{N} Z_i(t)^2 \tag{10}$$

- Lyapunov drift on slot t: $\Delta(t) = L(t+1) - L(t)$.

Drift-plus-penalty expression

- Lyapunov function:

$$L(t) = \frac{1}{2}\sum_{i=1}^{N}\sum_{s\in S_i} Q_i^{(s)}(t)^2 + \frac{1}{2}\sum_{i=1}^{N} Z_i(t)^2 \tag{10}$$

- Lyapunov drift on slot t: $\Delta(t) = L(t+1) - L(t)$.
- Drift-plus-penalty expression: $\Delta(t) - Vg(t)$.

Bound

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:

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

Online algorithm

Every slot t:

- Game manager observes queues and $\omega(t)$.

Online algorithm

Every slot t:

- Game manager observes queues and $\omega(t)$.
- Chooses $\alpha(t)$ in $A_1 \times A_2 \times \cdots \times A_N$to minimize:

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

Online algorithm

Every slot t:

- Game manager observes queues and $\omega(t)$.
- Chooses $\alpha(t)$ in $A_1 \times A_2 \times \cdots \times A_N$to minimize:

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

- Do an auxiliary variable selection.

Online algorithm

Every slot t:

- Game manager observes queues and $\omega(t)$.
- Chooses $\alpha(t)$ in $A_1 \times A_2 \times \cdots \times A_N$to minimize:

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

- Do an auxiliary variable selection.
- Update virtual queues.

Performance analysis

Theorem

If this online algorithm is implemented using a fixed value $V \geq 0$, then for all slots t one has :

$$\phi(\bar{\gamma_1}, ..., \bar{\gamma_N}) \geq \phi^* - \frac{B}{V} \tag{11}$$

Conclusion

- CCE constraints are simpler and lead to improved utilities.

Conclusion

- CCE constraints are simpler and lead to improved utilities.
- Online algorithm for the stochastic game.

Conclusion

- CCE constraints are simpler and lead to improved utilities.
- Online algorithm for the stochastic game.
- No knowledge of $\pi(\omega)$ required.

Thanks!