Computing Stackelberg Equilibrium

Branislav Bošanský

Artificial Intelligence Center,
Department of Computer Science,
Faculty of Electrical Engineering,
Czech Technical University in Prague

branislav.bosansky@agents.fel.cvut.cz

March 18, 2019
Stackelberg Equilibrium

Players have different roles in the Stackelberg solution concept:
Stackelberg Equilibrium

Players have different roles in the Stackelberg solution concept:

- *the leader* – publicly commits to a strategy
Stackelberg Equilibrium

Players have different roles in the Stackelberg solution concept:

- *the leader* – publicly commits to a strategy
- *the follower(s)* – play a Nash equilibrium with respect to the commitment of the leader
Stackelberg Equilibrium

Players have different roles in the Stackelberg solution concept:

- *the leader* – publicly commits to a strategy
- *the follower(s)* – play a Nash equilibrium with respect to the commitment of the leader

Stackelberg equilibrium is a strategy profile that satisfies the above conditions and maximizes the expected utility value of the leader:
Players have different roles in the Stackelberg solution concept:

- **the leader** – publicly commits to a strategy
- **the follower(s)** – play a Nash equilibrium with respect to the commitment of the leader

Stackelberg equilibrium is a strategy profile that satisfies the above conditions and maximizes the expected utility value of the leader:

$$\underset{\sigma \in \Sigma; \forall i \in N \setminus \{1\} \sigma_i \in BR_i(\sigma_{-i})}{\text{arg max}} \quad u_1(\sigma)$$
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:

- Arbitrary but fixed tie breaking rule

Strong SE
- The followers select such NE that maximizes the outcome of the leader (when the tie-braking is not specified, we mean SSE).

Weak SE
- The followers select such NE that minimizes the outcome of the leader.

Exact Weak Stackelberg equilibrium does not have to exist.
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:
- arbitrary but fixed tie breaking rule
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:

- arbitrary but fixed tie breaking rule
- *Strong SE* – the followers select such NE that maximizes the outcome of the leader (when the tie-breaking is not specified we mean SSE),
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:
- arbitrary but fixed tie breaking rule
- **Strong SE** – the followers select such NE that maximizes the outcome of the leader (when the tie-breaking is not specified we mean SSE),
- **Weak SE** – the followers select such NE that minimizes the outcome of the leader.
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:
- arbitrary but fixed tie breaking rule
- **Strong SE** – the followers select such NE that maximizes the outcome of the leader (when the tie-breaking is not specified we mean SSE),
- **Weak SE** – the followers select such NE that minimizes the outcome of the leader.

Exact Weak Stackelberg equilibrium does not have to exist.
There may be multiple Nash equilibria

The followers need to break ties in case there are multiple NE:

- arbitrary but fixed tie breaking rule
- **Strong SE** – the followers select such NE that maximizes the outcome of the leader (when the tie-breaking is not specified we mean SSE),
- **Weak SE** – the followers select such NE that minimizes the outcome of the leader.

Exact Weak Stackelberg equilibrium does not have to exist.

<table>
<thead>
<tr>
<th>1 \ 2</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
<td>(2,4)</td>
<td>(6,4)</td>
<td>(9,0)</td>
<td>(1,2)</td>
<td>(7,4)</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>(8,4)</td>
<td>(0,4)</td>
<td>(3,6)</td>
<td>(1,5)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>
There may be multiple Nash equilibria

Figure from [9].

\footnote{Figure from [9].}
There may be multiple Nash equilibria

\footnote{Figure from [9].}
Computing a Stackelberg equilibrium in NFGs

The problem is polynomial for two-players normal-form games; 1 is the leader, 2 is the follower.

Baseline polynomial algorithm requires solving $|S_2|$ linear programs:

$$\max_{\sigma_1 \in \Sigma_1} \sum_{s_1 \in S_1} \sigma_1(s_1) u_1(s_1, s_2) \geq \sum_{s_1 \in S_1} \sigma_1(s_1) u_2(s_1, s_2') \quad \forall s_2' \in S_2$$

one for each $s_2 \in S_2$ assuming $s_2$ is the best response of the follower.
The problem is polynomial for two-players normal-form games; 1 is the leader, 2 is the follower.
Computing a Stackelberg equilibrium in NFGs

The problem is polynomial for two-players normal-form games; 1 is the leader, 2 is the follower.

Baseline polynomial algorithm requires solving $|S_2|$ linear programs:

$$\max_{\sigma_1 \in \Sigma_1} \sum_{s_1 \in S_1} \sigma_1(s_1) u_1(s_1, s_2) \geq \sum_{s_1 \in S_1} \sigma_1(s_1) u_2(s_1, s_2') \quad \forall s_2' \in S_2$$

where $\sum_{s_1 \in S_1} \sigma_1(s_1) = 1$ for each $s_2 \in S_2$ assuming $s_2$ is the best response of the follower.
Computing a Stackelberg equilibrium in NFGs

The problem is polynomial for two-players normal-form games; 1 is the leader, 2 is the follower.

Baseline polynomial algorithm requires solving $|S_2|$ linear programs:

$$\max_{\sigma_1 \in \Sigma_1} \sum_{s_1 \in S_1} \sigma_1(s_1) u_1(s_1, s_2)$$

$$\sum_{s_1 \in S_1} \sigma_1(s_1) u_2(s_1, s_2) \geq \sum_{s_1 \in S_1} \sigma_1(s_1) u_2(s_1, s'_2) \quad \forall s'_2 \in S_2$$

$$\sum_{s_1 \in S_1} \sigma_1(s_1) = 1$$

one for each $s_2 \in S_2$ assuming $s_2$ is the best response of the follower.
Computing a Stackelberg equilibrium in NFGs

We can reformulate the program as a mixed-integer linear program (MILP) that is a basis for the hard cases (e.g., computing a SE in Bayesian games):

$\max_{\sigma \in \Sigma, y \in \{0, 1\}} |S_2| \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2) 0 \leq \sigma(s_1, s_2) \leq y(s_2) \forall s_1, s_2 \in S_1, S_2$

$\sum_{s_1, s_2 \in S_1, S_2} \sigma(s_1, s_2) u_2(s_1, s_2) \geq \sum_{s_1, s_2 \in S_1, S_2} \sigma(s_1, s_2) u_2(s_1, s'_2) \forall s'_2 \in S_2$

$\sum_{s_1, s_2 \in S_1, S_2} \sigma(s_1, s_2) = 1$

$\sum_{s_2 \in S_2} y(s_2) = 1$
We can reformulate the program as a mixed-integer linear program (MILP) that is a basis for the hard cases (e.g., computing a SE in Bayesian games):
Computing a Stackelberg equilibrium in NFGs

We can reformulate the program as a mixed-integer linear program (MILP) that is a basis for the hard cases (e.g., computing a SE in Bayesian games):

\[
\begin{align*}
\max_{\sigma \in \Sigma, y \in \{0,1\} | S_2|} & \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2) \\
0 & \leq \sigma(s_1, s_2) \leq y(s_2) \quad \forall s_1, s_2 \in S_{1,2} \\
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2) & \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s'_2) \quad \forall s'_2 \in S_2 \\
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) & = 1 \\
\sum_{s_2 \in S_2} y(s_2) & = 1
\end{align*}
\]
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:
two-player EFGs with chance (there exists a FPTAS for this

Main algorithms are based on the sequence-form LCP for
calculating NE:
\[ v_{\text{inf}}(\sigma_i) = s_{\sigma_i} + \sum_{I_i' \in I_i} v_{I_i'}(\sigma_i) + \sum_{\sigma^-_i \in \Sigma^-_i} g_i(\sigma_i, \sigma^-_i) \cdot r^-_i(\sigma^-_i) \]
\[ \forall i, \sigma_i \]
\[ r_i(\emptyset) = 1 = r_i(\sigma_i) \]
\[ 0 \leq r_i(\sigma_i) ; 0 \leq s_{\sigma_i} \]
\[ \forall i \in N \forall \sigma_i \in \Sigma_i \]
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:

\[
\inf_i (\sigma_i) = s_{\sigma_i} + \sum_{I_i \in I_i} \text{seq}_i (I_i) = \sigma_i v_{I_i} + \sum_{\sigma_{-i} \in \Sigma_{-i}} g_i (\sigma_i, \sigma_{-i}) \cdot r_{-i} (\sigma_{-i}) \quad \forall i, \sigma_i
\]

\[
r_i (\emptyset) = 1 = r_i (\sigma_i) \cdot s_{\sigma_i} \quad \forall i \in N, \forall \sigma_i \in \Sigma_i
\]

\[0 \leq r_i (\sigma_i) ; 0 \leq s_{\sigma_i} \quad \forall i \in N, \forall \sigma_i \in \Sigma_i\]
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:

- two-player EFGs with chance (there exists a FPTAS for this case [2]),
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:
- two-player EFGs with chance (there exists a FPTAS for this case [2]),
- two-player EFGs with imperfect information,
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:
- two-player EFGs with chance (there exists a FPTAS for this case [2]),
- two-player EFGs with imperfect information,
- two-player EFGs with perfect information but imperfect recall (games on DAGs).
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:

- two-player EFGs with chance (there exists a FPTAS for this case [2]),
- two-player EFGs with imperfect information,
- two-player EFGs with perfect information but imperfect recall (games on DAGs).

Main algorithms are based on the sequence-form LCP for computing NE:
Computing a Stackelberg equilibrium in EFGs

The problem is typically NP-hard [6, 2]:

- two-player EFGs with chance (there exists a FPTAS for this case [2]),
- two-player EFGs with imperfect information,
- two-player EFGs with perfect information but imperfect recall (games on DAGs).

Main algorithms are based on the sequence-form LCP for computing NE:

\[
\begin{align*}
\nu_{inf,i}(\sigma_i) &= s_{\sigma_i} + \sum_{I_i' \in \mathcal{I}_i : \text{seq}_i(I_i') = \sigma_i} \nu_{I_i'} + \sum_{\sigma_{i^{-}} \in \Sigma_{i^{-}}} g_i(\sigma_i, \sigma_{i^{-}}) \cdot r_{i^{-}}(\sigma_{i^{-}}) \quad \forall i, \sigma_i \\
r_i(\sigma_i) &= \sum_{a \in A(I_i)} r_i(\sigma_i a) \quad \forall i \in \mathcal{N} \ \forall I_i \in \mathcal{I}_i, \ \sigma_i = \text{seq}_i(I_i) \\
r_i(\emptyset) &= 1 \quad 0 = r_i(\sigma_i) \cdot s_{\sigma_i} \quad \forall i \in \mathcal{N} \ \forall \sigma_i \in \Sigma_i \\
0 &\leq r_i(\sigma_i) \ ; \quad 0 \leq s_{\sigma_i} \quad \forall i \in \mathcal{N} \ \forall \sigma_i \in \Sigma_i
\end{align*}
\]
Computing a Stackelberg equilibrium in EFGs

\[
\max_{p,r,v,s} \sum_{z \in Z} p(z) u_1(z) C(z) v_{\inf_2} \sigma_2 = s \sigma_2 + \sum_{I' \in I_2: \text{seq}_2(I')} \sigma_2 v_{I'} + \sum_{\sigma_1 \in \Sigma_1} r_1(\sigma_1) g_2(\sigma_1, \sigma_2) \quad \forall \sigma_2 \in \Sigma_2 \\
r_i(\emptyset) = 1 \\
r_i(\sigma_i) = \sum_{a \in A_i(I_i)} r_i(\sigma_i a) \quad \forall i \in N \forall I_i \in I_i \sigma_i = \text{seq}_i(I_i) \\
0 \leq s \sigma_2 \leq (1 - r_2(\sigma_2)) \cdot M \quad \forall \sigma_2 \in \Sigma_2 \\
0 \leq p(z) \leq r_2(\text{seq}_2(z)) \quad \forall z \in Z \\
0 \leq p(z) \leq r_1(\text{seq}_1(z)) \quad \forall z \in Z \\
1 = \sum_{z \in Z} p(z) C(z) r_2(\sigma_2) \in \{0, 1\} \quad \forall \sigma_2 \in \Sigma_2 \\
0 \leq r_1(\sigma_1) \leq 1 \quad \forall \sigma_1 \in \Sigma_1
\]
Computing a Stackelberg equilibrium in EFGs

MILP for computing SE for two-player extensive-form game with perfect recall:

\[
\begin{align*}
\max_{p, r, v, s} & \sum_{z \in Z} p(z) u_1(z) C(z) v_{\text{inf}}(\sigma_2) = s \sigma_2 + \sum_{i' \in I_2': \text{seq}_2(i')} \sigma_2 v_{i'} + \sum_{\sigma_1 \in \Sigma_1} r_1(\sigma_1) g_2(\sigma_1, \sigma_2) \quad \forall \sigma_2 \in \Sigma_2 \\
r_i(\emptyset) &= 1 \quad \forall i \in N \quad \forall I_i \in I_i, \sigma_i = \text{seq}_i(I_i) \\
0 &\leq p(z) \leq r_2(\text{seq}_2(z)) \quad \forall z \in Z \\
0 &\leq r_1(\sigma_1) \leq 1 \quad \forall \sigma_1 \in \Sigma_1
\end{align*}
\]
MILP for computing SE for two-player extensive-form game with perfect recall:

$$\max_{p,r,v,s} \sum_{z \in Z} p(z)u_1(z)C(z)$$

$$v_{\inf_2}(\sigma_2) = s_{\sigma_2} + \sum_{I' \in \mathcal{I}_2: \text{seq}_2(I') = \sigma_2} v_{I'} + \sum_{\sigma_1 \in \Sigma_1} r_1(\sigma_1)g_2(\sigma_1, \sigma_2) \quad \forall \sigma_2 \in \Sigma_2$$

$$r_i(\emptyset) = 1 \quad r_i(\sigma_i) = \sum_{a \in A_i(I_i)} r_i(\sigma_ia) \quad \forall i \in \mathcal{N} \forall I_i \in \mathcal{I}_i, \sigma_i = \text{seq}_i(I_i)$$

$$0 \leq s_{\sigma_2} \leq (1 - r_2(\sigma_2)) \cdot M \quad \forall \sigma_2 \in \Sigma_2$$

$$0 \leq p(z) \leq r_2(\text{seq}_2(z)) \quad \forall z \in Z$$

$$0 \leq p(z) \leq r_1(\text{seq}_1(z)) \quad \forall z \in Z$$

$$1 = \sum_{z \in Z} p(z)C(z)$$

$$r_2(\sigma_2) \in \{0, 1\} \quad \forall \sigma_2 \in \Sigma_2$$

$$0 \leq r_1(\sigma_1) \leq 1 \quad \forall \sigma_1 \in \Sigma_1$$
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:

- We maximize the expected utility of the leader.
- We restrict the joint probability distribution so that the follower plays a pure strategy.
- There are no incentive constraints of the leader.

We can compute a Stackelberg equilibrium if we modify an algorithm for computing an optimal correlated equilibrium.
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:

- we maximize the expected utility of the leader
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:

- we maximize the expected utility of the leader
- we restrict the joint probability distribution so that the follower plays a pure strategy
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:

- we maximize the expected utility of the leader
- we restrict the joint probability distribution so that the follower plays a pure strategy
- there are no incentive constraints of the leader
Recall the MILP program Stackelberg equilibrium and compare it to the LP for correlated equilibrium:

- we maximize the expected utility of the leader
- we restrict the joint probability distribution so that the follower plays a pure strategy
- there are no incentive constraints of the leader

We can compute a Stackelberg equilibrium if we modify an algorithm for computing an optimal correlated equilibrium.
We can reformulate the MILP program as a single LP:

\[ \max_{\sigma \in \Sigma} \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2) \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2) \forall s_2' \in S_2 \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) = 1 \]

Properties:
the objective is the same as in the MILP case (or multiple LPs) case,
strategy \( \sigma \) does not necessarily correspond to Stackelberg equilibrium (the follower can receive multiple recommendations that are best responses).
We can reformulate the MILP program as a single LP:
We can reformulate the MILP program as a single LP:

$$
\max_{\sigma \in \Sigma} \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2)
$$

$$
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2) \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2') \quad \forall s_2' \in S_2
$$

$$
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) = 1
$$
We can reformulate the MILP program as a single LP:

$$\max_{\sigma \in \Sigma} \sum_{s \in S} \sigma(s_1, s_2)u_1(s_1, s_2)$$

$$\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2)u_2(s_1, s_2) \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2)u_2(s_1, s'_2) \quad \forall s'_2 \in S_2$$

$$\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) = 1$$

Properties:
Computing a Stackelberg equilibrium in NFGs (2)

We can reformulate the MILP program as a single LP:

\[
\max_{\sigma \in \Sigma} \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2)
\]

\[
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2) \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s'_2) \quad \forall s'_2 \in S_2
\]

\[
\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) = 1
\]

Properties:

- the objective is the same as in the MILP case (or multiple LPs) case,
We can reformulate the MILP program as a single LP:

$$\max_{\sigma \in \Sigma} \sum_{s \in S} \sigma(s_1, s_2) u_1(s_1, s_2)$$

$$\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s_2) \geq \sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) u_2(s_1, s'_2) \quad \forall s'_2 \in S_2$$

$$\sum_{s_1, s_2 \in S_{1,2}} \sigma(s_1, s_2) = 1$$

Properties:

- the objective is the same as in the MILP case (or multiple LPs) case,
- strategy $\sigma$ does not necessarily corresponds to Stackelberg equilibrium (the follower can receive multiple recommendations that are best responses).
Computing a Stackelberg equilibrium in EFGs (2)

How does it work in EFGs?

We can define a Stackelberg extension of EFCE [2] – the leader (1) controls the correlation device, (2) sends signals to the follower, (3) maximizes her expected utility.
How does it work in EFGs?
How does it work in EFGs?

We can define a Stackelberg extension of EFCE [2] – the leader (1) controls the correlation device, (2) sends signals to the follower, (3) maximizes her expected utility.
How does it work in EFGs?

We can define a Stackelberg extension of EFCE [2] – the leader (1) controls the correlation device, (2) sends signals to the follower, (3) maximizes her expected utility.
Computing a Stackelberg equilibrium in EFGs (2)

How does it work in EFGs?

We can define a Stackelberg extension of EFCE [2] – the leader (1) controls the correlation device, (2) sends signals to the follower, (3) maximizes her expected utility.
Computing a Stackelberg equilibrium in EFGs (2)

We can follow the same steps [3]:

1. Consider an algorithm for computing an optimal EFCE in an EFG.
2. Remove the incentives constraints of the leader.
3. Add an objective to maximize the expected value of the leader.
4. Restrict the recommendations to the follower so that only a unique action in an information set is considered.
We can follow the same steps [3]:
We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
Computing a Stackelberg equilibrium in EFGs (2)

We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
- remove the incentives constraints of the leader
We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
- remove the incentives constraints of the leader
- add objective to maximize the expected value of the leader
Computing a Stackelberg equilibrium in EFGs (2)

We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
- remove the incentives constraints of the leader
- add objective to maximize the expected value of the leader
- restrict the recommendations to the follower so that only a unique action in an information set
We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
- remove the incentives constraints of the leader
- add objective to maximize the expected value of the leader
- restrict the recommendations to the follower so that only a unique action in an information set

![Graph showing runtime vs. number of realization plans for different algorithms]
Computing a Stackelberg equilibrium in EFGs (2)

We can follow the same steps [3]:

- consider an algorithm for computing an optimal EFCE in an EFGs
- remove the incentives constraints of the leader
- add objective to maximize the expected value of the leader
- restrict the recommendations to the follower so that only a unique action in an information set
Computing a Stackelberg equilibrium in EFGs (3)
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]
Computing a Stackelberg equilibrium in EFGs (3)

- incremental strategy generation [4]

<table>
<thead>
<tr>
<th>Instance (\epsilon)</th>
<th>0.01</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4a All-Points</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.9%</td>
<td>2.42%</td>
<td>3.01%</td>
<td>3.23%</td>
</tr>
<tr>
<td>4a No-Info</td>
<td>0%</td>
<td>0.35%</td>
<td>0.72%</td>
<td>1.29%</td>
<td>1.77%</td>
<td>2.5%</td>
<td>2.55%</td>
</tr>
<tr>
<td>4b All-Points</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.56%</td>
<td>0.8%</td>
<td>2.39%</td>
<td>2.48%</td>
</tr>
<tr>
<td>4b No-Info</td>
<td>0%</td>
<td>0.16%</td>
<td>0.67%</td>
<td>1.27%</td>
<td>2.15%</td>
<td>2.42%</td>
<td>2.86%</td>
</tr>
<tr>
<td>4c All-Points</td>
<td>0%</td>
<td>0%</td>
<td>0.033%</td>
<td>0.79%</td>
<td>3.47%</td>
<td>4.8%</td>
<td>6.38%</td>
</tr>
<tr>
<td>4c No-Info</td>
<td>0%</td>
<td>0%</td>
<td>0.24%</td>
<td>0.89%</td>
<td>1.75%</td>
<td>1.75%</td>
<td>1.75%</td>
</tr>
</tbody>
</table>
Computing a Stackelberg equilibrium in EFGs (4)

Using Finite State Machines for Computing SE (under review for EC '19)
we can restrict the set of pure strategies that we consider for
the follower in an EFG, these restrictions can be described using (for
example) Finite State Machines
Using Finite State Machines for Computing SE (under review for EC '19)
Using Finite State Machines for Computing SE (under review for EC ’19)
- we can restrict the set of pure strategies that we consider for the follower
Using Finite State Machines for Computing SE (under review for EC ’19)
- we can restrict the set of pure strategies that we consider for the follower
- in an EFG, these restrictions can be described using (for example) Finite State Machines
Computing a Stackelberg equilibrium in EFGs (4)

- Using Finite State Machines for Computing SE (under review for EC ’19)
  - we can restrict the set of pure strategies that we consider for the follower
  - in an EFG, these restrictions can be described using (for example) Finite State Machines
References I

(besides the books)


