On the notion of compensation between number and strength of attackers in ranking-based semantics

Leila Amgoud 1  Jonathan Ben-Naim 1  Dragan Doder 2  Srdjan Vesic 3

1IRIT - CNRS, France
2University of Belgrade, Serbia
3CRIL - CNRS & Univ. Artois, France

Argument Strength Workshop@Bochum’16
Ranking-based argumentation systems

- An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  - $\mathcal{A}$ is a set of arguments
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

One successful attack has the same effect as several attacks.

In some applications, this makes sense... but not always!

Example: dialogues

a: She is the best candidate for this position
b: She does not have enough experience
c: She does not speak English

In many situations:

- One attack does not have the same effect as several attacks
- One attack does not completely destroy its target

Ranking-based semantics do not compute extensions. They assign a score to each argument.
• An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  - $\mathcal{A}$ is a set of arguments
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

• One successful attack has the same effect as several attacks
An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
- $\mathcal{A}$ is a set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

One successful attack has the same effect as several attacks

In some applications, this makes sense... but not always!
An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
- $\mathcal{A}$ is a set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

One successful attack has the same effect as several attacks
In some applications, this makes sense... but not always!
Example: dialogues
- $a:$ She is the best candidate for this position
An argumentation framework \( \mathcal{F} = (\mathcal{A}, \mathcal{R}) \)
- \( \mathcal{A} \) is a set of arguments
- \( \mathcal{R} \subseteq \mathcal{A} \times \mathcal{A} \) is an attack relation

One successful attack has the same effect as several attacks.
In some applications, this makes sense... but not always!

Example: dialogues
- \( a \): She is the best candidate for this position
- \( b \): She does not have enough experience
Ranking-based argumentation systems

• An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  - $\mathcal{A}$ is a set of arguments
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

• One successful attack has the same effect as several attacks
• In some applications, this makes sense... but not always!
• Example: dialogues
  - $a$: *She is the best candidate for this position*
  - $b$: *She does not have enough experience*
  - $c$: *She does not speak English*
An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
- $\mathcal{A}$ is a set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

One successful attack has the same effect as several attacks
In some applications, this makes sense... but not always!
Example: dialogues
- $a$: *She is the best candidate for this position*
- $b$: *She does not have enough experience*
- $c$: *She does not speak English*

In many situations:
An argumentation framework $F = (A, R)$

- $A$ is a set of arguments
- $R \subseteq A \times A$ is an attack relation

One successful attack has the same effect as several attacks

In some applications, this makes sense... but not always!

Example: dialogues

- $a : \text{She is the best candidate for this position}$
- $b : \text{She does not have enough experience}$
- $c : \text{She does not speak English}$

In many situations:

- One attack does not have the same effect as several attacks
An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
- $\mathcal{A}$ is a set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

One successful attack has the same effect as several attacks.

In some applications, this makes sense... but not always!

Example: dialogues
- $a$: She is the best candidate for this position
- $b$: She does not have enough experience
- $c$: She does not speak English

In many situations:
- One attack does not have the same effect as several attacks.
- One attack does not completely destroy its target.
• An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  ■ $\mathcal{A}$ is a set of arguments
  ■ $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

• One successful attack has the same effect as several attacks
• In some applications, this makes sense... but not always!
• Example: dialogues
  ■ $a : \text{She is the best candidate for this position}$
  ■ $b : \text{She does not have enough experience}$
  ■ $c : \text{She does not speak English}$

• In many situations:
  ■ One attack does not have the same effect as several attacks
  ■ One attack does not completely destroy its target
• Ranking-based semantics
Ranking-based argumentation systems

- An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  - $\mathcal{A}$ is a set of arguments
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

- One successful attack has the same effect as several attacks
- In some applications, this makes sense... but not always!
- Example: dialogues
  - $a$: She is the best candidate for this position
  - $b$: She does not have enough experience
  - $c$: She does not speak English

- In many situations:
  - One attack does not have the same effect as several attacks
  - One attack does not completely destroy its target

- Ranking-based semantics
  - do not compute extensions
Ranking-based argumentation systems

• An argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$
  - $\mathcal{A}$ is a set of arguments
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation

• One successful attack has the same effect as several attacks
• In some applications, this makes sense... but not always!
• Example: dialogues
  - $a : \text{She is the best candidate for this position}$
  - $b : \text{She does not have enough experience}$
  - $c : \text{She does not speak English}$

• In many situations:
  - One attack does not have the same effect as several attacks
  - One attack does not completely destroy its target

• Ranking-based semantics
  - do not compute extensions
  - assign a score to each argument
One minute crash course in ranking-based semantics

\[ \mathcal{F}_1 \]
One minute crash course in ranking-based semantics

\[ \mathcal{F}_1 \]
One minute crash course in ranking-based semantics

\[ \mathcal{F}_1 \]

\[ p \rightarrow q \rightarrow r \rightarrow s \rightarrow a \rightarrow b \]

\[ \mathcal{F}_2 \]

\[ p \rightarrow q \rightarrow a \rightarrow b \]
One minute crash course in ranking-based semantics

\[ F_1 \]

\[ F_2 \]
One minute crash course in ranking-based semantics

$\mathcal{F}_1$

\[ p \rightarrow q \rightarrow r \rightarrow s \rightarrow a \]

$\mathcal{F}_2$

\[ p \rightarrow q \rightarrow a \rightarrow b \]

$\mathcal{F}_3$

\[ t \rightarrow v \rightarrow r \rightarrow p \rightarrow q \rightarrow a \]

\[ x \rightarrow y \rightarrow s \rightarrow z \]

\[ b \rightarrow v \rightarrow x \rightarrow y \rightarrow z \]
One minute crash course in ranking-based semantics

\[ F_1 \]

\[ F_2 \]

\[ F_3 \]
Axioms for ranking-based semantics

- Several semantics exists in the literature
Axioms for ranking-based semantics

- Several *semantics* exists in the literature
- Which *properties* should they satisfy?
Axioms for ranking-based semantics

- Several semantics exist in the literature
- Which properties should they satisfy?
- Define and study axioms for those semantics
Abstraction

$\mathcal{F}_1$

$\mathcal{F}_2$
Independence
Independence
Void Precedence
Defence Precedence
Group Comparison

\[
x_1 \quad x_2 \quad \ldots \quad x_i \quad y_1 \quad y_2 \quad \ldots \quad y_i \quad z_1 \quad z_2 \quad \ldots \quad z_k
\]

\[a \text{ is stronger than } b\]
$y_1$ stronger than $x_1$
$y_2$ stronger than $x_2$
\ldots
$y_i$ stronger than $x_i$
$y_1$ stronger than $x_1$
$y_2$ stronger than $x_2$
\ldots
$y_i$ stronger than $x_i$
$k \in \{0, 1, \ldots\}$
$y_1$ stronger than $x_1$
$y_2$ stronger than $x_2$
$\ldots$
$y_i$ stronger than $x_i$
$k \in \{0, 1, \ldots\}$
$y_1$ stronger than $x_1$
$y_2$ stronger than $x_2$
\ldots
$y_i$ stronger than $x_i$
$k \in \{0, 1, \ldots\}$

\[ a \text{ is stronger than } b \]
The notion of compensation

- A compromise between the **strength** and the **number** of attackers?
The notion of compensation

- A compromise between the strength and the number of attackers?
- No axiom specifies what to do
The notion of compensation

- A compromise between the strength and the number of attackers?
- No axiom specifies what to do
- Do we even want to decide?

```
p
  /\  
 a  r
/    \__/
|      |   
|      v   
|    x   y   z   
|   \    |     |
|    r   s    t
|   /    |
|  b
```
Our idea: A parametrised ranking-based semantics

- Define a semantics based on a parameter
Our idea: A parametrised ranking-based semantics

- Define a semantics based on a **parameter**
- Allow the user to choose to which extent to take into account...
Our idea: A parametrised ranking-based semantics

- Define a semantics based on a parameter
- Allow the user to choose to which extent to take into account
  - the strength of attackers
Our idea: A parametrised ranking-based semantics

- Define a semantics based on a parameter
- Allow the user to choose to which extent to take into account
  - the strength of attackers
  - the number of attackers
Our idea: A parametrised ranking-based semantics

- Define a semantics based on a parameter
- Allow the user to choose to which extent to take into account
  - the strength of attackers
  - the number of attackers
Compensation: a new axiom

A parametrized semantics $s$ satisfies compensation at degree $(n, k)$ if there exists a unique $\alpha$ such that $s^\alpha(a) = s^\alpha(b)$.

This version of the axiom is applicable when $x_j$ and $y_i$ are not attacked.

A more general version?

⇒ see me during the coffee break.
A parametrized semantics $s_\alpha$ satisfies **compensation** at degree $(n, k)$ if there exists a unique $\alpha$ s.t. $s_\alpha(a) = s_\alpha(b)$
A parametrized semantics $s_\alpha$ satisfies compensation at degree $(n, k)$ if there exists a unique $\alpha$ s.t. $s_\alpha(a) = s_\alpha(b)$

- This version of the axiom is applicable when $x_i^j$ are not attacked
A parametrized semantics $s_\alpha$ satisfies compensation at degree $(n, k)$ if there exists a unique $\alpha$ s.t. $s_\alpha(a) = s_\alpha(b)$

- This version of the axiom is applicable when $x^i_j$ are not attacked
- A more general version? ⇒ see me during the coffee break
Let $\alpha \in (0, +\infty)$. We define $s_\alpha : \mathcal{A} \rightarrow [1, +\infty)$ such that $\forall a \in \mathcal{A}$,

$$s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha}$$
How does our semantics work?
How does our semantics work?

\[ \mathcal{F}_1 \]

\[ p \rightarrow a \]
\[ q \rightarrow b \]
\[ r \rightarrow b \]
\[ s \rightarrow b \]
How does our semantics work?
How does our semantics work?

\[ \mathcal{F}_1 \]

\[ \mathcal{F}_2 \]
How does our semantics work?

$\mathcal{F}_1$

$\mathcal{F}_2$

$\mathcal{F}_3$
How does our semantics work?

\[ F_1 \]

\[ F_2 \]

\[ F_3 \]
And what about compensation?
And what about compensation?
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
- Does $s_\alpha$ exist for every argumentation graph $\mathcal{F}$?

Theorem
For every argumentation graph, for every $\alpha \in (0, +\infty)$, $s_\alpha$ exists and is unique.

Proof: long and difficult... but very interesting!... at least we think so ;-(
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
- Does $s_\alpha$ exist for every argumentation graph $\mathcal{F}$?
- Easy case: no cycles $\Rightarrow s_\alpha$ exists and is unique
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
- Does $s_\alpha$ exist for every argumentation graph $\mathcal{F}$?
- **Easy case:** no cycles $\Rightarrow$ $s_\alpha$ exists and is unique
- But in general case?
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
- Does $s_\alpha$ exist for every argumentation graph $\mathcal{F}$?
- Easy case: no cycles $\Rightarrow$ $s_\alpha$ exists and is unique
- But in general case?

**Theorem**

*For every argumentation graph, for every $\alpha \in (0, +\infty)$, $s_\alpha$ exists and is unique.*
Existence and uniqueness of \( s_\alpha \)

- Burden number \( (s_\alpha) \) depends on the burden number of attackers
- Does \( s_\alpha \) exist for every argumentation graph \( \mathcal{F} \)?
- **Easy case:** no cycles \( \Rightarrow \) \( s_\alpha \) exists and is unique
- But in **general case**?

**Theorem**

*For every argumentation graph, for every \( \alpha \in (0, +\infty) \), \( s_\alpha \) exists and is unique.*

- Proof: long and difficult...
Existence and uniqueness of $s_\alpha$

- Burden number ($s_\alpha$) depends on the burden number of attackers
- Does $s_\alpha$ exist for every argumentation graph $\mathcal{F}$?
- Easy case: no cycles $\Rightarrow$ $s_\alpha$ exists and is unique
- But in general case?

**Theorem**

For every argumentation graph, for every $\alpha \in (0, +\infty)$, $s_\alpha$ exists and is unique.

- Proof: long and difficult...
- but very interesting!
Existence and uniqueness of $s_{\alpha}$

- Burden number ($s_{\alpha}$) depends on the burden number of attackers
- Does $s_{\alpha}$ exist for every argumentation graph $\mathcal{F}$?
- Easy case: no cycles $\Rightarrow s_{\alpha}$ exists and is unique
- But in general case?

**Theorem**

*For every argumentation graph, for every $\alpha \in (0, +\infty)$, $s_{\alpha}$ exists and is unique.*

- Proof: long and difficult...
- but very interesting!
- ...at least we think so ;-)
• Our semantics satisfies all the mandatory postulates for ranking-based semantics...
• Our semantics satisfies all the mandatory postulates for ranking-based semantics...

• ... and compensation for every $n, k \in \mathbb{N}$ such that $n > k$
Properties

- Our semantics satisfies all the mandatory postulates for ranking-based semantics...
- ... and compensation for every $n, k \in \mathbb{N}$ such that $n > k$
- (we are working on another semantics that satisfies it for every $n, k$)
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument

$$s_\alpha(a) = 1 + \left(\sum_{b \in \text{Att}(a)} 1(s_\alpha(b))^{\alpha}\right)^{1/\alpha}$$

Example for $\alpha = 2$ and $\epsilon = 0.0000001$.

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6057</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1

\[
s_\alpha(a) = 1 + \frac{\sum_{b \in \text{Att}(a)} s_\alpha(b)^{1/\alpha}}{\alpha}
\]
How to calculate $s_{\alpha}$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$

$$s_\alpha(a) = 1 + \left( \sum_{b \in Att(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha}$$
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  
  $$s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha}$$

- Example for $\alpha = 2$ and $\epsilon = 0.0000001$
How to calculate $s_{\alpha}$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  
  $s_{\alpha}(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_{\alpha}(b))^\alpha} \right)^{1/\alpha}$

- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6057</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  
  $$s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha}$$

- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  \[ s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha} \]
- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6057</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  \[ s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha} \]
- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6056</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
How to calculate $s_{\alpha}$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  $$s_{\alpha}(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_{\alpha}(b))^\alpha} \right)^{1/\alpha}$$
- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>1</td>
<td>2.414</td>
<td>2.000</td>
<td>2.000</td>
<td>2.000</td>
</tr>
<tr>
<td>2</td>
<td>1.707</td>
<td>1.414</td>
<td>1.500</td>
<td>1.500</td>
</tr>
<tr>
<td>3</td>
<td>1.942</td>
<td>1.586</td>
<td>1.707</td>
<td>1.666</td>
</tr>
<tr>
<td>4</td>
<td>1.838</td>
<td>1.518</td>
<td>1.631</td>
<td>1.586</td>
</tr>
<tr>
<td>5</td>
<td>1.879</td>
<td>1.543</td>
<td>1.660</td>
<td>1.613</td>
</tr>
<tr>
<td>6</td>
<td>1.864</td>
<td>1.532</td>
<td>1.648</td>
<td>1.602</td>
</tr>
<tr>
<td>7</td>
<td>1.870</td>
<td>1.536</td>
<td>1.653</td>
<td>1.607</td>
</tr>
<tr>
<td>8</td>
<td>1.873</td>
<td>1.534</td>
<td>1.651</td>
<td>1.605</td>
</tr>
<tr>
<td>9</td>
<td>1.869</td>
<td>1.535</td>
<td>1.651</td>
<td>1.605</td>
</tr>
<tr>
<td>10</td>
<td>1.869</td>
<td>1.535</td>
<td>1.651</td>
<td>1.605</td>
</tr>
<tr>
<td>11</td>
<td>1.869</td>
<td>1.535</td>
<td>1.651</td>
<td>1.605</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.868</td>
<td>1.535</td>
<td>1.651</td>
<td>1.605</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  
  $s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^\alpha} \right)^{1/\alpha}$

- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6057</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
How to calculate $s_\alpha$ in practice?

- Set the burden number of every argument to 1
- Update the burden number of each argument $a_i$
  \[ s_\alpha(a) = 1 + \left( \sum_{b \in \text{Att}(a)} \frac{1}{(s_\alpha(b))^{\alpha}} \right)^{1/\alpha} \]
- Example for $\alpha = 2$ and $\epsilon = 0.0000001$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2.4142</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7071</td>
<td>1.4142</td>
<td>1.5000</td>
<td>1.5000</td>
</tr>
<tr>
<td>3</td>
<td>1.9428</td>
<td>1.5857</td>
<td>1.7071</td>
<td>1.6666</td>
</tr>
<tr>
<td>4</td>
<td>1.8385</td>
<td>1.5147</td>
<td>1.6306</td>
<td>1.5857</td>
</tr>
<tr>
<td>5</td>
<td>1.8796</td>
<td>1.5439</td>
<td>1.6601</td>
<td>1.6132</td>
</tr>
<tr>
<td>6</td>
<td>1.8643</td>
<td>1.5320</td>
<td>1.6477</td>
<td>1.6023</td>
</tr>
<tr>
<td>7</td>
<td>1.8705</td>
<td>1.5363</td>
<td>1.6527</td>
<td>1.6069</td>
</tr>
<tr>
<td>8</td>
<td>1.8679</td>
<td>1.5346</td>
<td>1.6508</td>
<td>1.6050</td>
</tr>
<tr>
<td>9</td>
<td>1.8689</td>
<td>1.5353</td>
<td>1.6516</td>
<td>1.6057</td>
</tr>
<tr>
<td>10</td>
<td>1.8685</td>
<td>1.5350</td>
<td>1.6513</td>
<td>1.6054</td>
</tr>
<tr>
<td>11</td>
<td>1.8687</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>1.8686</td>
<td>1.5351</td>
<td>1.6514</td>
<td>1.6055</td>
</tr>
</tbody>
</table>
Experimental results

- Benchmark from ICCMA 2015
  - 90 argumentation frameworks
- Less than 1.5 seconds per example on my PC
Experimental results

- Benchmark from ICCMA 2015
  - 90 argumentation frameworks
  - $\epsilon = 0.00001$

- Less than 1.5 seconds per example on my PC
Experimental results

- Benchmark from ICCMA 2015
  - 90 argumentation frameworks
  - $\epsilon = 0.00001$
  - three values: $\alpha \in \{0.3, 1, 10\}$
Experimental results

- Benchmark from ICCMA 2015
  - 90 argumentation frameworks
  - $\epsilon = 0.00001$
  - three values: $\alpha \in \{0.3, 1, 10\}$
- Less than 1.5 seconds per example in average on my PC
Experimental results

- Benchmark from ICCMA 2015
  - 90 argumentation frameworks
  - $\epsilon = 0.00001$
  - three values: $\alpha \in \{0.3, 1, 10\}$
- Less than 1.5 seconds per example in average on my PC

![Graph showing number of iterations for different alpha values](image-url)
Summary

- New axiom for ranking-based semantics
Summary

- New *axiom* for ranking-based semantics
- New *ranking-based* semantics
Summary

- New axiom for ranking-based semantics
- New ranking-based semantics
- Several weak attacks can compensate one strong attack
Summary

- New axiom for ranking-based semantics
- New ranking-based semantics
- Several weak attacks can compensate one strong attack
- Parameter $\alpha$: to which extent we value quality vs quantity of attackers
Summary

- New axiom for ranking-based semantics
- New ranking-based semantics
- Several weak attacks can compensate one strong attack
- Parameter $\alpha$: to which extent we value quality vs quantity of attackers
  - notion of compensation
Summary

• New axiom for ranking-based semantics
• New ranking-based semantics
• Several weak attacks can compensate one strong attack
• Parameter $\alpha$: to which extent we value quality vs quantity of attackers
  □ notion of compensation
• Implementation
Summary

• New axiom for ranking-based semantics
• New ranking-based semantics
• Several weak attacks can compensate one strong attack
• Parameter $\alpha$: to which extent we value quality vs quantity of attackers
  ■ notion of compensation
• Implementation
• Promising experimental results
Summary

- New axiom for ranking-based semantics
- New ranking-based semantics
- Several weak attacks can compensate one strong attack
- Parameter $\alpha$: to which extent we value quality vs quantity of attackers
  - notion of compensation
- Implementation
- Promising experimental results

Thank you