Helpful Behavior Based on Trust for Web Services

Ioan Alfred Letia, Radu Răzvan Slăvescu

Technical University of Cluj-Napoca
Department of Computer Science
Motivating Example

Recipe Composition

Outline

Ioan Alfred Letia, Radu Razvan Slăvescu

WoSS - SYNASC 2010, Timisoara
1 Problem statement

2 Subjective Logic Based Trust

3 Belief Receipte Tree
   - Node types
   - Functions
   - Commitment
   - BRT example

4 Conclusions and Future Work
Outline

1. Problem statement
2. Subjective Logic Based Trust
   - Node types
   - Functions
   - Commitment
   - BRT example
3. Conclusions and Future Work

Ioan Alfred Letia, Radu Razvan Slăvescu
WoSS - SYNASC 2010, Timisoara
1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipte Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work
Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost
Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost
Introduction

Issues when building composite WS

1. Decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs.
2. Select a particular WS provider for every basic task.

Criteria to be met

1. Functional: producing the intended output.
2. Non-functional: quality aspects like response time, compliance or cost.
Introduction

Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost
Introduction

Suggested Approach

Developing a composite WS = solve, with maximum likelihood, a joint goal by a set of cooperative agents like WS architect and a consultant. Targets:

1. an abstract plan (solution)
2. and a concrete list of WS providers for every step of it

Types of decisions

- architect: choose each service and its provider
- consultant: adopt (or not) a helpful behavior (suggest an alternative to maximize the chances of success, while not providing its partner with more information than paid)
Introduction

Suggested Approach

Developing a composite WS = solve, with maximum likelihood, a joint goal by a set of cooperative agents like WS architect and a consultant. Targets:

1. an abstract plan (solution)
2. and a concrete list of WS providers for every step of it

Types of decisions

- architect: choose each service and its provider
- consultant: adopt (or not) a helpful behavior (suggest an alternative to maximize the chances of success, while not providing its partner with more information than paid)
Introduction

Contributions

1. employing a "Belief Recipe Tree (BRT)" for making decisions
2. employing prior experiences-based trust, for the values required by the BRT
3. revisiting it after each experience
Opinion

Proposition $x \rightarrow$ opinion $\omega_x = < b_x, d_x, u_x, a_x >$, (belief, disbelief, uncertainty, prior probability about the truth of $x$)

- binomial opinions $\rightarrow$ beta probability distribution function
- $r$ evidence supporting $x$ and $s$ supporting $\neg x$ $\rightarrow$

\[
\begin{align*}
  b &= \frac{r}{r+s+2} \\
  d &= \frac{s}{r+s+2} \\
  u &= \frac{2}{r+s+2}
\end{align*}
\]

- probability expectation value:

\[ E(\omega_x) = b_x + a_x u_x \]
Problem statement
Subjective Logic Based Trust
Belief Receipe Tree
Conclusions and Future Work

Sentences and opinions

Opinion

Proposition $x \rightarrow \text{opinion } \omega_x = \langle b_x, d_x, u_x, a_x \rangle$, (belief, disbelief, uncertainty, prior probability about the truth of $x$)

- binomial opinions $\rightarrow$ beta probability distribution function
- $r$ evidence supporting $x$ and $s$ supporting $\neg x$ $\rightarrow$

\[
\begin{align*}
    b &= \frac{r}{r+s+2} \\
    d &= \frac{s}{r+s+2} \\
    u &= \frac{2}{r+s+2}
\end{align*}
\]

- probability expectation value:

\[
E(\omega_x) = b_x + a_x u_x
\]
Proposition $x \rightarrow \text{opinion } \omega_x = \langle b_x, d_x, u_x, a_x \rangle$, (belief, disbelief, uncertainty, prior probability about the truth of $x$)

- **binomial opinions** $\rightarrow$ beta probability distribution function
- $r$ evidence supporting $x$ and $s$ supporting $\neg x$ $\rightarrow$

$$\begin{cases} 
    b = \frac{r}{r+s+2} \\
    d = \frac{s}{r+s+2} \\
    u = \frac{2}{r+s+2}
\end{cases}$$

- probability expectation value:

$$E(\omega_x) = b_x + a_x u_x$$
Problem statement
Subjective Logic Based Trust
Belief Receipt Tree
Conclusions and Future Work

Sentences and opinions

Opinion

Proposition $x \rightarrow$ opinion $\omega_x = \langle b_x, d_x, u_x, a_x \rangle$, (belief, disbelief, uncertainty, prior probability about the truth of $x$)

- binomial opinions $\rightarrow$ beta probability distribution function
- $r$ evidence supporting $x$ and $s$ supporting $\neg x$ $\rightarrow$

$$
\begin{align*}
    b &= \frac{r}{r+s+2} \\
    d &= \frac{s}{r+s+2} \\
    u &= \frac{2}{r+s+2}
\end{align*}
$$

- probability expectation value:

$$
E(\omega_x) = b_x + a_x u_x
$$
Problem statement
Subjective Logic Based Trust
Belief Receipe Tree
Conclusions and Future Work

Sentences and opinions

![Belief Tree Diagram]

Uncertainty

Disbelief  a  E(ω)  Belief

ω (b, d, u, a)
Sentences and opinions

Operators

1. **conjunction** $\land$: given $\omega_x = (b_x, d_x, u_x, a_x)$ and $\omega_y = (b_y, d_y, u_y, a_y)$, their conjunction $\omega_{x \land y}$, written $\omega_x \cap \omega_y$, has the following components:

$$
\begin{align*}
 b_{x \land y} &= b_x b_y + \frac{(1-a_x)a_y b_x u_y + a_x (1-a_y) u_x b_y}{1-a_x a_y} \\
 d_{x \land y} &= d_x + d_y - d_x d_y \\
 u_{x \land y} &= u_x u_y + \frac{(1-a_y)b_x u_y + (1-a_x)u_x b_y}{1-a_x a_y} \\
 a_{x \land y} &= a_x a_y
\end{align*}
$$

2. **disjunction** $\lor$: similar
BRT Structure

Purpose

- defines a probability distribution over possible recipes for completing the action in the tree’s root
- each node in a BRT has an action attached
- allows agents to reason efficiently about the possible recipes for performing an action: $O(nm)^d$, ($n =$ number of potential recipes for an action, $m =$ average number of steps per action, $d =$ number of levels of decomposition)
- deals with opinions rather than mere probabilities as its Probabilistic counterpart
Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipte Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work
Node types

- terminal nodes: atomic tasks
- non-terminal nodes: composed tasks

Node information

- an opinion describing the trust level the evaluator has in the success of the task represented by that node
- a cost, which estimates how much the plan effector must pay in order to have the task corresponding to the node completed
- an income modeling the benefit obtained if the task is accomplished.
Structure

Node types
- terminal nodes: atomic tasks
- non-terminal nodes: composed tasks

Node information
- an opinion describing the trust level the evaluator has in the success of the task represented by that node
- a cost, which estimates how much the plan effector must pay in order to have the task corresponding to the node completed
- an income modeling the benefit obtained if the task is accomplished.
Structures

1. Leaf nodes: atomic tasks (selecting a specific atomic web service)
2. (Intermediate) AND: all its children must be completed (multi-step recipe for a composed web service)
3. (Intermediate) OR: alternatives for achieving a specific goal (making a nondeterministic choice of one alternative over the other in case one needs a web service of type $T$ and two concrete web services of that type, $Tws_1$ and $Tws_2$ are available but just one needs to be chosen)
context: all beliefs an agent bases its decision on at moment \( T \) when action \( \alpha \) is done.

2. function \( \text{cba.basic}(G_1, \beta, C_\beta) \) gives the probability that agent \( G_1 \) can bring about the atomic action \( \beta \) within context \( C_\beta \).

3. function \( \text{cba.cost}(G_1, G_2, \beta, C_\beta) \) returns the cost for agent \( G_1 \) when the basic level action \( \beta \) is done by agent \( G_2 \) within context \( C_\beta \).

4. function \( V(G_1, \alpha, C_\alpha) \) gives the utility form \( G_1 \) if action \( \alpha \) is performed within context \( C_\alpha \).
### Functions attached

1. **context**: all beliefs an agent bases its decision on at moment $T$ when action $\alpha$ is done.

2. **function $cba.basic(G_1, \beta, C_\beta)$**: gives the probability that agent $G_1$ can bring about the atomic action $\beta$ within context $C_\beta$.

3. **function $cba.cost(G_1, G_2, \beta, C_\beta)$**: returns the cost for agent $G_1$ when the basic level action $\beta$ is done by agent $G_2$ within context $C_\beta$.

4. **function $V(G_1, \alpha, C_\alpha)$**: gives the utility form $G_1$ if action $\alpha$ is performed within context $C_\alpha$. 

---

Ioan Alfred Letia, Radu Razvan Slăvescu  
WoSS - SYNASC 2010, Timisoara
BRT structure

Functions attached

1. context: all beliefs an agent bases its decision on at moment $T$ when action $\alpha$ is done.
2. function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent $G_1$ can bring about the atomic action $\beta$ within context $C_\beta$.
3. function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent $G_1$ when the basic level action $\beta$ is done by agent $G_2$ within context $C_\beta$.
4. function $V(G_1, \alpha, C_\alpha)$ gives the utility form $G_1$ if action $\alpha$ is performed within context $C_\alpha$.
context: all beliefs an agent bases its decision on at moment $T$ when action $\alpha$ is done.

2 function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent $G_1$ can bring about the atomic action $\beta$ within context $C_\beta$.

3 function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent $G_1$ when the basic level action $\beta$ is done by agent $G_2$ within context $C_\beta$.

4 function $V(G_1, \alpha, C_\alpha)$ gives the utility form $G_1$ if action $\alpha$ is performed within context $C_\alpha$;
BRT structure

Functions attached

1. Function $p_{CBA}(BRT_\alpha, C_\alpha)$ computes the probability of action $\alpha$ to succeed and returns $cba.basic$ for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.

2. $Cost(G_i, BRT_\alpha, C_\alpha)$ computes the expected cost payed by agent $G_i$ and returns $cost.basic$ for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.

3. $Eval(BRT_\alpha)$ computes expected utility based on costs, incomes and probabilities.
Functions attached

1. Function $p_{CBA}(BRT_\alpha, C_\alpha)$ computes the probability of action $\alpha$ to succeed and returns $cba.basic$ for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.

2. $Cost(G_i, BRT_\alpha, C_\alpha)$ computes the expected cost payed by agent $G_i$ and returns $cost.basic$ for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.

3. $Eval(BRT_\alpha)$ computes expected utility based on costs, incomes and probabilities.
BRT structure

Functions attached

1. Function \( p_{CBA}(BRT_\alpha, C_\alpha) \) computes the probability of action \( \alpha \) to succeed and returns \( cba.basic \) for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.

2. \( Cost(G_i, BRT_\alpha, C_\alpha) \) computes the expected cost payed by agent \( G_i \) and returns \( cost.basic \) for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.

3. \( Eval(BRT_\alpha) \) computes expected utility based on costs, incomes and probabilities.
Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipe Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work
Agent $G_1$ is committed to $\alpha$ iff $G_1$ believes that $BRT_{\alpha}$ maximizes group utility:

$$\exists BRT_{\alpha} \in \text{BEL}(G_1, \forall BRT_{\beta} \text{ } BRT_{\beta} \neq BRT_{\alpha} \Rightarrow \text{Eval}(BRT_{\beta}) \leq \text{Eval}(BRT_{\alpha})) \land \text{Int.Th}(G_1, \text{SelectedBRT}(BRT_{\alpha}))$$
If agent is committed to $\alpha$ and believes that sending information $o$ to its partner will increase group utility, it will do so:

\begin{align*}
\text{if } & \text{Committed}(G_1, GR, \alpha) \text{ then} \\
BRT_\alpha &= PredictBRT(G_1, GR, \alpha, C_{GR}) \\
C_\beta &= ContextUpdate(C_\beta, o) \\
BRT_\beta &= PredictBRT(G_1, G_2, \beta, C_\beta) \\
BRT^o_\alpha &= BRTReplace(BRT_\alpha, BRT_\beta) \\
\text{utility} &= Eval(BRT^o_\alpha) - Eval(BRT_\alpha) \\
\text{endif} \\
\text{if } & \text{utility} \geq \text{CommunicationCost}(G_2) \text{ then} \\
& \text{Int. To}(G_1, \text{Communicate}(G_1, G_2, o)) \\
\text{endif}
\end{align*}
Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipe Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work
An example: Route generator design

1. an architect A and a consultant C want to build a Navigator WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: $Mws_1$, $Mws_2$, $Mws_3$ (map WS), $Tws_1$, $Tws_2$ (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = Route Generator + Weather Forcast.
5. Route Generator = Mapws (map info) + Trafficws (traffic info);
6. Weather Forcast = weather info
7. the customer agreed to pay $30K for the Route Generator, $2K for the Weather Forcast and separately $40K for the final delivery.
An example: Route generator design

1. an architect A and a consultant C want to build a Navigator WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forecast
3. WS offering info: $Mws_1$, $Mws_2$, $Mws_3$ (map WS), $Tws_1$, $Tws_2$ (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forecast).
4. system = Route Generator + Weather Forcast.
5. Route Generator = Mapws (map info) + Trafficws (traffic info);
6. Weather Forcast = weather info
7. the customer agreed to pay $30K for the Route Generator, $2K for the Weather Forcast and separately $40K for the final delivery
An example: Route generator design

1. an architect A and a consultant C want to build a Navigator WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forecast
3. WS offering info: Mws\textsubscript{1}, Mws\textsubscript{2}, Mws\textsubscript{3} (map WS), Tws\textsubscript{1}, Tws\textsubscript{2} (traffic info WS), Wfws\textsubscript{1}, Wfws\textsubscript{2} (weather forecast).
4. system = Route Generator + Weather Forcast.
5. Route Generator = Mapws (map info) + Trafficws (traffic info);
6. Weather Forcast = weather info
7. the customer agreed to pay $30K for the Route Generator, $2K for the Weather Forcast and separately $40K for the final delivery
An example: Route generator design

1. an architect $A$ and a consultant $C$ want to build a *Navigator* WS (which generates car routes for drivers)

2. info for route building: paths + traffic + weather forecast

3. WS offering info: $M_{ws_1}, M_{ws_2}, M_{ws_3}$ (map WS), $T_{ws_1}, T_{ws_2}$ (traffic info WS), $W_{ws_1}, W_{ws_2}$ (weather forecast).

4. system = *Route Generator* + *Weather Forcast*.

5. *Route Generator* = $Map_{ws}$ (map info) + $Traffic_{ws}$ (traffic info);

6. *Weather Forcast* = weather info

7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery.
An example: Route generator design

1. an architect A and a consultant C want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forecast
3. WS offering info: $Mws_1$, $Mws_2$, $Mws_3$ (map WS), $Tws_1$, $Tws_2$ (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forecast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = $Mapws$ (map info) + $Trafficws$ (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery.
An example: Route generator design

1. an architect A and a consultant C want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forecast
3. WS offering info: $M_{ws_1}$, $M_{ws_2}$, $M_{ws_3}$ (map WS), $T_{ws_1}$, $T_{ws_2}$ (traffic info WS), $W_{fws_1}$, $W_{fws_2}$ (weather forecast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = $Map_{ws}$ (map info) + *Traffic ws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery
An example: Route generator design

1. an architect A and a consultant C want to build a *Navigator WS* (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forecast
3. WS offering info: $Mws_1$, $Mws_2$, $Mws_3$ (map WS), $Tws_1$, $Tws_2$ (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forecast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = $Mapws$ (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery.
An example: Route generator design

1. An architect A and a consultant C want to build a Navigator WS (which generates car routes for drivers).
2. Info for route building: paths + traffic + weather forecast.
3. WS offering info: $Mws_1, Mws_2, Mws_3$ (map WS), $Tws_1, Tws_2$ (traffic info WS), $Wfws_1, Wfws_2$ (weather forecast).
4. System = Route Generator + Weather Forcast.
5. Route Generator = $Mapws$ (map info) + Trafficws (traffic info);
7. The customer agreed to pay $30K for the Route Generator, $2K for the Weather Forcast and separately $40K for the final delivery.
An example: Route generator design

Question: if consultant \( C \) has just learned that WS *Complex Route Generator* does the jobs of *Mapws* and *Trafficws* in one step and architect \( A \) is not aware of this, should \( C \) convey this new piece of knowledge to \( A \) or not?
No complex route generator
### Atomic WS evidence and opinions

<table>
<thead>
<tr>
<th>WS</th>
<th>r</th>
<th>s</th>
<th>b</th>
<th>d</th>
<th>u</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{ws_1}$</td>
<td>1</td>
<td>1</td>
<td>0.250</td>
<td>0.250</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>$M_{ws_2}$</td>
<td>3</td>
<td>1</td>
<td>0.500</td>
<td>0.167</td>
<td>0.333</td>
<td>0.500</td>
</tr>
<tr>
<td>$M_{ws_3}$</td>
<td>1</td>
<td>2</td>
<td>0.200</td>
<td>0.400</td>
<td>0.400</td>
<td>0.500</td>
</tr>
<tr>
<td>$T_{ws_1}$</td>
<td>5</td>
<td>1</td>
<td>0.625</td>
<td>0.125</td>
<td>0.250</td>
<td>0.500</td>
</tr>
<tr>
<td>$T_{ws_2}$</td>
<td>3</td>
<td>2</td>
<td>0.428</td>
<td>0.286</td>
<td>0.286</td>
<td>0.500</td>
</tr>
<tr>
<td>$W_{fws_1}$</td>
<td>78</td>
<td>22</td>
<td>0.765</td>
<td>0.216</td>
<td>0.019</td>
<td>0.500</td>
</tr>
<tr>
<td>$W_{fws_2}$</td>
<td>67</td>
<td>33</td>
<td>0.657</td>
<td>0.222</td>
<td>0.111</td>
<td>0.500</td>
</tr>
</tbody>
</table>
Deciding to cooperate

1. if, for Mapws, the descending branch opinions $Mws_1$, $Mws_2$, $Mws_3$ are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the Mapws opinion is $(0.433, 0.288, 0.279, 0.578)$.

2. Simpler: computation gives $(0.345, 0.465, 0.19, 0.253)$.

3. Navigator: $(0.227, 0.635, 0.138, 0.553)$.

4. $Eval(\text{Navigator}) = $5,202.40

5. if Complex Route Generator costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(\text{Navigator})$ is $7,919.38$

6. Conclusion: it worth telling $A$ about this, as long as the cost of communication is lower than the additional benefit.
Deciding to cooperate

1. If, for Mapws, the descending branch opinions $M_{ws_1}$, $M_{ws_2}$, $M_{ws_3}$ are (0.2, 0.4, 0.4, 0.5), (0.25, 0.25, 0.5, 0.5) and (0.4, 0.2, 0.4, 0.5), then the Mapws opinion is (0.433, 0.288, 0.279, 0.578).

2. Simpler: computation gives (0.345, 0.465, 0.19, 0.253).

3. Navigator: (0.227, 0.635, 0.138, 0.553).

4. $\text{Eval(Navigator)} = \$5,202.40$

5. If Complex Route Generator costs $\$10,000$ and has opinions of (0.666, 0.167, 0.167, 0.500) and (0.333, 0.0, 0.667, 0.500), the new value for $\text{Eval(Navigator)}$ is $\$7,919.38$

6. Conclusion: it worth telling $A$ about this, as long as the cost of communication is lower than the additional benefit.
Deciding to cooperate

1. if, for \(\text{Mapws}\), the descending branch opinions \(\text{Mws}_1\), \(\text{Mws}_2\), \(\text{Mws}_3\) are \((0.2, 0.4, 0.4, 0.5), (0.25, 0.25, 0.5, 0.5)\) and \((0.4, 0.2, 0.4, 0.5)\), then the \(\text{Mapws}\) opinion is \((0.433, 0.288, 0.279, 0.578)\).

2. \texttt{Simplerg}: computation gives \((0.345, 0.465, 0.19, 0.253)\).

3. \texttt{Navigator}: \((0.227, 0.635, 0.138, 0.553)\).

4. \(\text{Eval(}\text{Navigator}\text{)} = 5,202.40\)

5. if \texttt{Complex Route Generator} costs \$10,000 and has opinions of \((0.666, 0.167, 0.167, 0.500)\) and \((0.333, 0.0, 0.667, 0.500)\), the new value for \(\text{Eval(}\text{Navigator}\text{)}\) is \$7,919.38.

Conclusion: it is worth telling \(A\) about this, as long as the cost of communication is lower than the additional benefit.
Deciding to cooperate

1. if, for Mapws, the descending branch opinions $Mws_1$, $Mws_2$, $Mws_3$ are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the Mapws opinion is $(0.433, 0.288, 0.279, 0.578)$.

2. Simplerg: computation gives $(0.345, 0.465, 0.19, 0.253)$.

3. Navigator: $(0.227, 0.635, 0.138, 0.553)$.

4. $Eval(Navigator) = 5,202.40$

5. if Complex Route Generator costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is $7,919.38$

6. Conclusion: it worth telling $A$ about this, as long as the cost of communication is lower than the additional benefit.
Deciding to cooperate

1. if, for \( \text{Mapws} \), the descending branch opinions \( \text{Mws}_1, \text{Mws}_2, \text{Mws}_3 \) are \((0.2, 0.4, 0.4, 0.5), (0.25, 0.25, 0.5, 0.5)\) and \((0.4, 0.2, 0.4, 0.5)\), then the \( \text{Mapws} \) opinion is \((0.433, 0.288, 0.279, 0.578)\).

2. \textbf{Simplerg}: computation gives \((0.345, 0.465, 0.19, 0.253)\).

3. \textbf{Navigator}: \((0.227, 0.635, 0.138, 0.553)\).

4. \[ \text{Eval}(\text{Navigator}) = \$5,202.40 \]

5. if \textbf{Complex Route Generator} costs \$10,000 and has opinions of \((0.666, 0.167, 0.167, 0.500)\) and \((0.333, 0.0, 0.667, 0.500)\), the new value for \[ \text{Eval}(\text{Navigator}) \] is \$7,919.38

6. Conclusion: it worth telling A about this, as long as the cost of communication is lower than the additional benefit.
Deciding to cooperate

1. If, for Mapws, the descending branch opinions $M_{ws_1}$, $M_{ws_2}$, $M_{ws_3}$ are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the Mapws opinion is $(0.433, 0.288, 0.279, 0.578)$.

2. Simpler: computation gives $(0.345, 0.465, 0.19, 0.253)$.

3. Navigator: $(0.227, 0.635, 0.138, 0.553)$.

4. $\text{Eval}(\text{Navigator}) = 5,202.40$

5. If Complex Route Generator costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $\text{Eval}(\text{Navigator})$ is $7,919.38$

6. Conclusion: it worth telling $A$ about this, as long as the cost of communication is lower than the additional benefit.
With complex route generator

![Belief Receipt Tree Diagram]

- NAVIGATOR
- ROUTE GENERATOR
- WEATHER FORECAST
- SIMPLE ROUTE GENERATOR
- CRG₁
- MAPWS
- TRAFFICWS
- Mws₁, Mws₂, Mws₃, Tws₁, Tws₂

© AND NODE
○ OR NODE
Deciding not to cooperate

1. If $CRG_1$ has failed twice, after adjusting the beliefs, we get the new values for $Eval(Navigator)$: $5,202.40$ and $4,180.70$ respectively.

2. Conclusion: in this case, $C$ should not suggest the alternative.

Observation

This is not feasible in the classical approach when the values of probabilities remain the same, thus the flexibility of this new approach.
Deciding not to cooperate

1. if $CRG_1$ has failed twice, after adjusting the beliefs, we get the new values for $Eval(Navigator)$: $5,202.40$ and $4,180.70$ respectively.

2. Conclusion: in this case, $C$ should not suggest the alternative

Observation

This is not feasible in the classical approach when the values of probabilities remain the same, thus the flexibility of this new approach
BRT versus PRT

In the same scenario, let us consider 2 situations:

1. \( WFws_1 \) has performance record \( r = 300, s = 1000 \)
2. \( WFws_1 \) has performance record \( r = 3, s = 1 \)

If a maximum threshold of uncertainty of 0.01 is considered:

1. situation 1 \( (u = 0.001) \): commit to weather forecast WS
2. situation 2 \( (u = 0.333) \): ask for more evidence

Observation

The classical approach gives the same solution since the same ratio \( r/s \). This also illustrates the flexibility of this new approach.
BRT versus PRT

In the same scenario, let us consider 2 situations:

1. $WF_{ws_1}$ has performance record $r = 300$, $s = 1000$
2. $WF_{ws_1}$ has performance record $r = 3$, $s = 1$

If a maximum threshold of uncertainty of 0.01 is considered:

1. situation 1 ($u = 0.001$): commit to weather forecast WS
2. situation 2 ($u = 0.333$): ask for more evidence

Observation

The classical approach gives the same solution since the same ratio $r/s$. This also illustrates the flexibility of this new approach.
Conclusions

1. BRT extended with trust in form of subjective opinions to accommodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.
Conclusions

1. BRT extended with trust in form of subjective opinions to accommodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed).
2. This allows agents to decide whether to pay for more information or simply to rely on the available one.
3. It also allows agents to adjust beliefs as further evidence becomes available.
4. It offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.
Conclusions

1. BRT extended with trust in form of subjective opinions to accommodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.
Conclusions

1. BRT extended with trust in form of subjective opinions to accommodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.
Future work

1. extensive experiments with this formalism
2. extend the formalism to multinominal (complex, mutually exclusive) opinions like "The level of success is poor/average/good/excellent" as opposite to a mere failure/success
3. estimate base rates by learning class stereotypes
Future work

1. extensive experiments with this formalism
2. extend the formalism to multinomial (complex, mutually exclusive) opinions like "The level of success is poor/average/good/excellent" as opposite to a mere failure/success
3. estimate base rates by learning class stereotypes
Future work

1. extensive experiments with this formalism
2. extend the formalism to multinomial (complex, mutually exclusive) opinions like "The level of success is poor/average/good/excellent" as opposite to a mere failure/success
3. estimate base rates by learning class stereotypes
Thanks for your attention!

Questions?
Thanks for your attention!
Questions?