Analysing the Applicability of a Multi-Criteria Decision Method in Fog Computing Placement Problem

Isaac Lera, Carlos Guerrero and Carlos Juiz
Dept. of Mathematics and Computer Science
University of the Balearic Islands (Palma, Majorca, Spain)
isaac.lera@uib.es

The Fourth IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2019)
Fog Computing

Architecture that distributes computing, storage, and networking closer to users, and anywhere along the Cloud.

OpenFog Consortium - Definition

Where do we allocate services along the continuum?

- Not trivial
- Dynamic
- Complex Decision

Cloud

Network

- Core
- Edge

Things

Users & IoT end-devices
Fog Computing Placement Problem (FCPP) is a challenging problem:

* **Influences** on the functional and non-functional requirements.

* **Depends** on multi-criteria satisfaction, i.e. latency, temporal availability, power consumption, costs, application packaging, resource utilisation, scaling possibilities, network congestions, software compatibility, user preferences, licenses, redundant links, geographical distribution, migrations, application composition, & so on.

NP-hard problem
Our approach

Other approaches try to find the most optimal allocation: a lot of resources, slow process, and global- &- static vision of the ecosystem

To address this problem:

* Our objective is to find the placement of a service that satisfies the higher number of criteria,
* This decision concerning diverse and, often, opposite criteria
* This process can be applied
  * multiple-times / dynamically
  * performed on fog-nodes

We use multi-criteria decision aiding discipline (MCDA) method...
MCDA

- MCDA is an activity which helps **making decisions** mainly in terms of choosing, ranking or sorting the actions.

- The ingredients of **MCDA** are a finite or infinite set of actions (alternatives, solutions, courses of action, ...), at least two criteria, and, obviously, at least one decision-maker (DM).

- (MCDA) A decision "is a binary relation $S$ defined on the set of potential actions $a$ such that $aSb$ if there are enough arguments to decide that $a$ is at least as good as $b$, whereas there is no essential argument to refute that statement."

- Main outranking family methods are: PROMETHEE and ELECTRE: I, IV, IS, II, III, IV, SS, TRI
To “aSb” two possible allocations, we need to evaluate them in terms of $m$ criteria ($g_1$, $g_2$, ..., $g_m$), i.e. latency.

Given a criterion ($g_i$), the alternative $a$ is considered better than alternative $b$ when $g_i(a) > g_i(b)$

\[
\text{if } \text{eval}_\text{latency}(\text{alloc}_a) > \text{eval}_\text{latency}(\text{alloc}_b) \text{ then alloc}_a \text{ better than alloc}_b \text{ in terms of latency}
\]

but in this decision, we need to manage with some uncertain threshold of preference and indifference values for each criterion.
ELECTRE III ingredients \( (2^0) \)

**1º Concordance Matrix:**

\[
C_i(a, b) = \begin{cases} 
1 & \text{if } g_i(a) \geq g_i(b) - q_i(g_i(a)) \\
0 & \text{if } g_i(a) \leq g_i(b) - p_i(g_i(a)) \\
\frac{p_i(g_i(a)) + g_i(a) - g_i(b)}{p_i(g_i(a)) - q_i(g_i(a))} & \text{otherwise}
\end{cases}
\]

**2º Discordance Matrix:**

\[
d_i(a, b) = \begin{cases} 
1 & \text{if } g_i(a) \geq g_i(b) - v_i(g_i(a)) \\
0 & \text{if } g_i(a) > g_i(b) - p_i(g_i(a)) \\
\frac{g_i(b) - g_i(a) - p_i(g_i(a))}{v_i(g_i(a)) - p_i(g_i(a))} & \text{otherwise}
\end{cases}
\]

**3º “\( a \preceq b \) = a outranks b”:**

\[
S(a, b) = \begin{cases} 
C(a, b) & \text{if } g_i(a) \geq g_i(b) - v_i(g_i(a)) \\
C(a, b) \prod_{i \in K} \frac{1 - d_i(a, b)}{1 - C(a, b)} & \text{otherwise}
\end{cases}
\]
Experiment

We compare our **Electre III**-based approach with a simple weighted average in six different cases according with the necessities of a DM.

The idea is to determine if all the cases reflect the importance of each decision.

Preference, Indifference and Veto thresholds are chosen dynamically according with all possible values in each allocation for each criterion.

<table>
<thead>
<tr>
<th>Case</th>
<th>Hop count</th>
<th>Latency</th>
<th>Power</th>
<th>Cost</th>
<th>D.Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**10th percentile / 3**  **20th percentile**  **40th percentile**
Modelling a dynamic Scenario

* **5 Criteria**: Latency, Hop count, Energy consumption, Cost, and Deployment penalty

* **10 Applications** set up by a composition of 10 avg. services (modeled as a Directed Acyclic Graph model)

* **40 Users**, where their arrivals follow an exponential distribution

  * In each User's service invocation, -> **ALLOCATION PROCESS**

* **Network infrastructure**: a graph

  * Generalized Linear Preference (GLP) (aSHIIP tool): 200 nodes

  * All nodes are Fog nodes

  * IPS = random (50: 1000)
    PR links = r(10: 90)
    BW links = r(100: 1000)

  * Power min = r(30: 50) & max = r(400: 1000)

  * Cost depends on degree level, 0–4 degree value are “cheaper” & >4 degree value are “expensive”
Yet Another Fog Simulator

Python & Free software

Infrastructure -> Complex Network theory

Dynamic movement of "things/users", services and other events.

https://github.com/acsicuib/YAFS
Results

<table>
<thead>
<tr>
<th>Case</th>
<th>Hop count</th>
<th>Latency</th>
<th>Power</th>
<th>Cost</th>
<th>D.Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

hop count

latency

Users or Gateways - Case A

(a) Case A
Results

<table>
<thead>
<tr>
<th>Case</th>
<th>Hop count</th>
<th>Latency</th>
<th>Power</th>
<th>Cost</th>
<th>D.Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

### Latency

**App:0**
- Electre III: 169 ms
- Weighted avg.: 196 ms

**App:1**
- Electre III: 172 ms
- Weighted avg.: 178 ms

**App:2**
- Electre III: 194 ms
- Weighted avg.: 196 ms

**App:3**
- Electre III: 169 ms

**App:4**
- Electre III: 197 ms

**App:5**
- Electre III: 187 ms

**App:6**
- Electre III: 194 ms

**App:7**
- Electre III: 198 ms

**App:8**
- Electre III: 199 ms

**App:9**
- Electre III: 197 ms

### Hop Count

**Case C**

### Power

**Case C**

Users or Gateways - Case C
Discussion & Conclusions

EXPLORATORY WORK -> Results are limited!

The WA gives more importance to a specific criteria without considering the rest (obvious)

ELECTRE III method preserves the decision maker.

Not significative results due to a uniform definition of the infrastructure. there are no polarized situations

A WA is computationally less cost than this ELECTRE III method.

Conclusion:

MCDA methods can aid to find allocation with multiple and contradictory criteria; a powerful and flexible tool to incorporate more detailed criteria: hardware configurations, budgets, user preferences, and so on.

As future work, we need to do MORE EXPERIMENTATION: from 2 to n-criteria, and comparing with more optimisation solutions
Analysing the Applicability of a Multi-Criteria Decision Method in Fog Computing Placement Problem

Isaac Lera, Carlos Guerrero and Carlos Juiz
Dept. of Mathematics and Computer Science
University of Balearic Islands (Palma, Spain)
isaac.lera@uib.es

Thank you for your attention

Any question?

This research was supported by the Spanish Government and the European Commision (Fondo Europeo de Desarrollo Regional) through grant number TIN2017-88547-P (MINECO/AEI/FEDER, UE)

http://ordcot.uib.es/

https://github.com/acsicuib/YAFS