FlowSort parameters elicitation: the case of interval sorting

Dimitri Van Assche
Yves De Smet
Computer and Decision Engineering - SMG
Université libre de Bruxelles
Table of content

- Introduction
- Interval sorting
- FlowSort, a sorting method based on PROMETHEE
- Algorithm
- Results
- Future works
Introduction

• Subject of this work:
  ◦ Find the parameters of FlowSort, a sorting method,
  ◦ Applied to interval sorting.

• This presentation follows a first contribution, available as a Technical Report, considering the case of «standard» sorting.
Sorting

- An alternative belongs to a single category.

\[ C_1 \]
\[ a_2 \\
 a_7 \\
 a_3 \]

\[ C_2 \]
\[ a_1 \]
\[ a_9 \]
\[ a_4 \]
\[ a_6 \]

\[ C_3 \]
\[ a_5 \]
\[ a_8 \]

FlowSort
Interval Sorting

- An alternative belongs to an interval of categories.

\[ C_1 \rightarrow C_2 \rightarrow C_3 \]

**FlowSort**
For interval sorting
FlowSort

- Sorting method based on PROMETHEE.
- 2 methods:
  - «Standard» sorting: based on PROMETHEE II;
  - Interval sorting: based on PROMETHEE I.
FlowSort: Sorting with Promethee

- A set of alternatives $A = \{a_1, a_2 \ldots, a_n\}$;
- A set of criteria $F = \{f_1, f_2 \ldots, f_q\}$, $f_l(a): A \rightarrow \mathbb{R}$: the evaluation of the alternative $a$ on criterion $l$.
- A preference function $P_l(x): \mathbb{R} \rightarrow [0,1]$ is assigned to each criterion $l$.  


FlowSort: Linear preference function

- 2 parameters:
  - $q_l$: indifference threshold;
  - $p_l$: preference threshold;
- $w_l$: weight of the criterion.

$$P_l(x) = \begin{cases} 
0, & x < q_l \\
\frac{x-q_l}{p_l-q_l}, & q_l \leq x < p_l \\
1, & x \geq p_l 
\end{cases}$$
FlowSort: PROMETHEE

- Preference degree on criterion $l$:
  \[ \pi_l(a_i, a_j) = P_l(f_l(a_j) - f_l(a_i)) \]

- Global preference degree of $a_i$ over $a_j$:
  \[ \pi(a_i, a_j) = \sum_l w_l \cdot \pi_l(a_i, a_j) \]
FlowSort: PROMETHEE

- Positive flow score:
  \[ \phi^+(a_i) = \frac{1}{n - 1} \sum_{x \in A} \pi(a_i, x) \]

- Negative flow score:
  \[ \phi^-(a_i) = \frac{1}{n - 1} \sum_{x \in A} \pi(x, a_i) \]

- Net flow score:
  \[ \phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \]
FlowSort

- Extension of PROMETHEE to sorting;
- A set of predefined ordered categories \( C = \{c_1, c_2 \ldots, c_m\}, c_1 > c_2 > \ldots > c_m; \)
- A set of central profiles \( R = \{r_1, r_2 \ldots, r_m\} \) representing the categories;
- Let’s define \( R_i = R \cup \{a_i\}. \)
Example
FlowSort

- Assignation rule:
  \[ h^*(a_i) = \arg\min_{h=1,2,...,m} |\phi_{R_i}(a_i) - \phi_{R_i}(r_l)| \]

- Assignation rule for interval sorting:
  \[ h^{+*}(a_i) = \arg\min_{h=1,2,...,m} |\phi^+_{R_i}(a_i) - \phi^+_{R_i}(r_l)| \]
  \[ h^{-*}(a_i) = \arg\min_{h=1,2,...,m} |\phi^-_{R_i}(a_i) - \phi^-_{R_i}(r_l)| \]

  Interval: \([h^{-*}(a_i), h^{+*}(a_i)]\)
Algorithm

- A Genetic Algorithm has been used.
- Problem: find the set of parameters that minimizes the distance of the categorization computed with these and the real one.

\[
\sum_{a \in A} (|c_f^+(a) - c_r^+(a)| + |c_f^-(a) - c_r^-(a)|)
\]
Algorithm: parameters to find

<table>
<thead>
<tr>
<th></th>
<th>Crit₁</th>
<th>Crit₂</th>
<th>Crit₃</th>
<th>...</th>
<th>Crit₉</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>w₁</td>
<td>w₂</td>
<td>w₃</td>
<td>...</td>
<td>w₉</td>
</tr>
<tr>
<td>q (indifference)</td>
<td>q₁</td>
<td>q₂</td>
<td>q₃</td>
<td>...</td>
<td>q₉</td>
</tr>
<tr>
<td>p (preference)</td>
<td>p₁</td>
<td>p₂</td>
<td>p₃</td>
<td>...</td>
<td>p₉</td>
</tr>
<tr>
<td>r₁</td>
<td>r₁₁</td>
<td>r₁₂</td>
<td>r₁₃</td>
<td>...</td>
<td>r₁₉</td>
</tr>
<tr>
<td>r₂</td>
<td>r₂₁</td>
<td>r₂₂</td>
<td>r₂₃</td>
<td>...</td>
<td>r₂₉</td>
</tr>
<tr>
<td>r₃</td>
<td>r₃₁</td>
<td>r₃₂</td>
<td>r₃₃</td>
<td>...</td>
<td>r₃₉</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>rₘ</td>
<td>rₘ₁</td>
<td>rₘ₂</td>
<td>rₘ₃</td>
<td>...</td>
<td>rₘ₉</td>
</tr>
</tbody>
</table>

(3 + m) * q parameters to find
Genetic algorithm overview

Current population

selection

crossover

mutation

New population
Selection operator

- Random pick of 2 solutions.
- Randomly choose one of both with a probability related to its fitness. (roulette wheel selection)
Crossover operator

- 2 parameters:
  - crossover probability,
  - gene crossover probability.
- Crossover of 2 solutions with a randomly chosen $\lambda$:
  
  $$O_1(i) = \lambda \ast p_1(i) + (1 - \lambda) \ast p_2(i)$$
  
  $$O_2(i) = (1 - \lambda) \ast p_1(i) + \lambda \ast p_2(i)$$
## Solution pattern

<table>
<thead>
<tr>
<th></th>
<th>$Crit_1$</th>
<th>$Crit_2$</th>
<th>$Crit_3$</th>
<th>...</th>
<th>$Crit_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weight</strong></td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
<td>...</td>
<td>$w_q$</td>
</tr>
<tr>
<td><strong>Q (indifference)</strong></td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$q_3$</td>
<td>...</td>
<td>$q_q$</td>
</tr>
<tr>
<td><strong>P (preference)</strong></td>
<td>$p_1$</td>
<td>$p_2$</td>
<td>$p_3$</td>
<td>...</td>
<td>$p_q$</td>
</tr>
<tr>
<td>$r_1$</td>
<td>$r_{11}$</td>
<td>$r_{12}$</td>
<td>$r_{13}$</td>
<td>...</td>
<td>$r_{1q}$</td>
</tr>
<tr>
<td>$r_2$</td>
<td>$r_{21}$</td>
<td>$r_{22}$</td>
<td>$r_{23}$</td>
<td>...</td>
<td>$r_{2q}$</td>
</tr>
<tr>
<td>$r_3$</td>
<td>$r_{31}$</td>
<td>$r_{32}$</td>
<td>$r_{33}$</td>
<td>...</td>
<td>$r_{3q}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$r_m$</td>
<td>$r_{m1}$</td>
<td>$r_{m2}$</td>
<td>$r_{m3}$</td>
<td>...</td>
<td>$r_{mq}$</td>
</tr>
</tbody>
</table>
Mutation operator

- 2 parameters:
  - mutation probability,
  - gene mutation probability.

- For each profile’s values:
  - mutation depends on the percentage of overcategorization w.r.t. undercategorization of the category.

- Mutation range restricted with respect to the current correctness of the solution.
Mutation operator

![Graph showing the relationship between mutation operator and criteria]

- Criterion 1: Line r1 remains relatively stable.
- Criterion 2: Line r2 shows a decreasing trend.
- Criterion 3: Line r3 exhibits an increasing trend.
- Line r4 follows a pattern that is distinct from the others.
- Line a demonstrates a fluctuating behavior.
Parameters’ fine tuning

- 5 parameters (mutation probability, crossover probability, etc)
- Use of varying parameters between $p_{\text{min}}$ and $p_{\text{max}}$ with a certain angle.
- Less stuck in local optima.
Testing procedure

- 3 datasets have been chosen: CPU, BC, CEV.
- Interval sorting has been generated with a random parameters instantiation of FlowSort.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#inst.</th>
<th>#crit.</th>
<th>#cat.</th>
<th>#param.</th>
<th>%imprecise cat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>209</td>
<td>6</td>
<td>4</td>
<td>42</td>
<td>40.67</td>
</tr>
<tr>
<td>BC</td>
<td>278</td>
<td>7</td>
<td>2</td>
<td>35</td>
<td>23.02</td>
</tr>
<tr>
<td>CEV</td>
<td>1728</td>
<td>6</td>
<td>4</td>
<td>42</td>
<td>16.43</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Learning set's size</th>
<th>Dataset</th>
<th>Correctness</th>
<th>Learning set correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>CPU</td>
<td>0.7337±0.0705</td>
<td>1.0000±0.0000</td>
</tr>
<tr>
<td></td>
<td>BC</td>
<td>0.8827±0.0337</td>
<td>0.9981±0.0120</td>
</tr>
<tr>
<td></td>
<td>CEV</td>
<td>0.8498±0.0224</td>
<td>0.9227±0.0383</td>
</tr>
<tr>
<td>20%</td>
<td>CPU</td>
<td>0.8798±0.0245</td>
<td>0.9880±0.0160</td>
</tr>
<tr>
<td></td>
<td>BC</td>
<td>0.9463±0.0209</td>
<td>0.9955±0.0103</td>
</tr>
<tr>
<td></td>
<td>CEV</td>
<td>0.8809±0.0173</td>
<td>0.8554±0.0338</td>
</tr>
<tr>
<td>35%</td>
<td>CPU</td>
<td>0.9004±0.0215</td>
<td>0.9642±0.0243</td>
</tr>
<tr>
<td></td>
<td>BC</td>
<td>0.9579±0.0210</td>
<td>0.9919±0.0110</td>
</tr>
<tr>
<td></td>
<td>CEV</td>
<td>0.8868±0.0154</td>
<td>0.8395±0.0277</td>
</tr>
<tr>
<td>50%</td>
<td>CPU</td>
<td>0.9065±0.0228</td>
<td>0.9581±0.0214</td>
</tr>
<tr>
<td></td>
<td>BC</td>
<td>0.9747±0.0163</td>
<td>0.9913±0.0111</td>
</tr>
<tr>
<td></td>
<td>CEV</td>
<td>0.8944±0.0168</td>
<td>0.8309±0.0252</td>
</tr>
</tbody>
</table>
Correctness w.r.t. learning set’s size
Conclusion

- Good overall learning set correctness.
- Good prediction on the tests sets.
- Running time rather small, moreless 5 minutes for the dataset CPU on a modern computer.
- Dataset CEV seems more difficult.
Future research

- Defining benchmark datasets for interval sorting;
- Try to use an exact method for a simplified version of the model (linear version of PROMETHEE);
- Comparing different methods to investigate which one performs better on which kind of dataset;
- Deepen the idea of « partial » information.