FlowSort parameters elicitation: the case of interval sorting

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

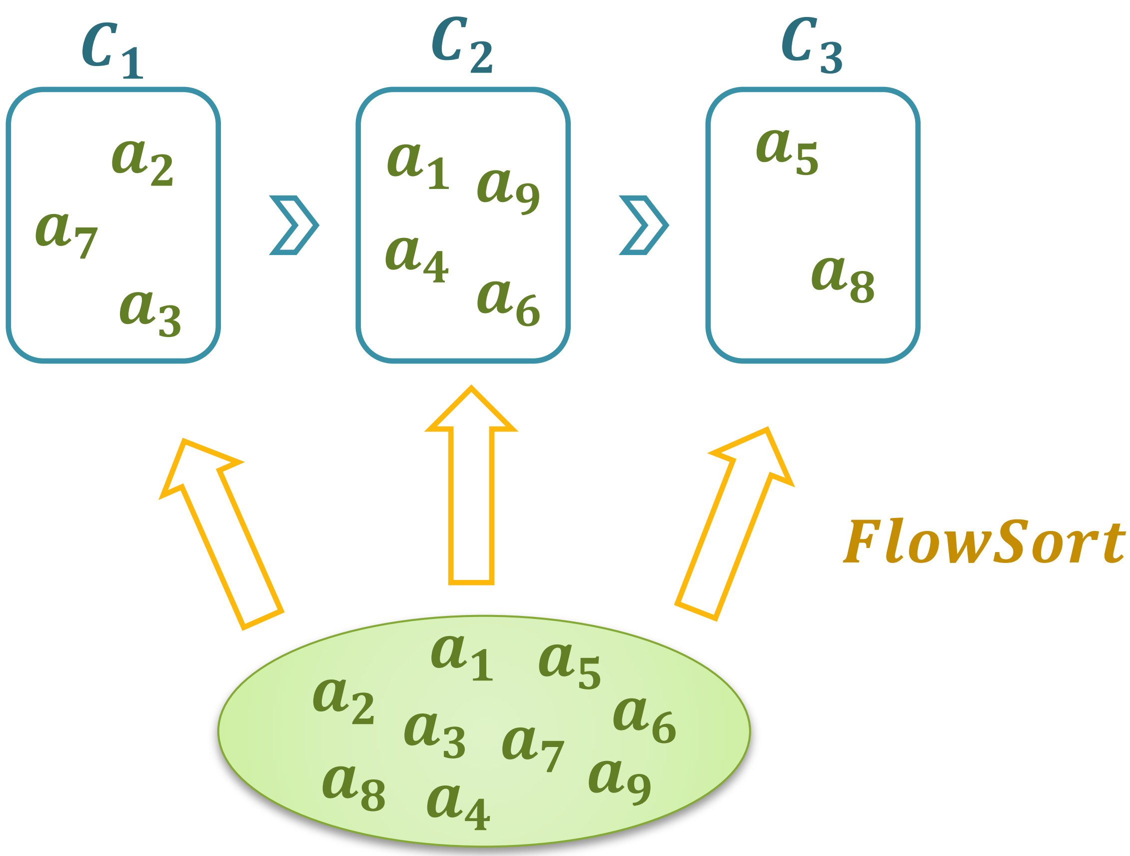
Table of content

Introduction

- 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.

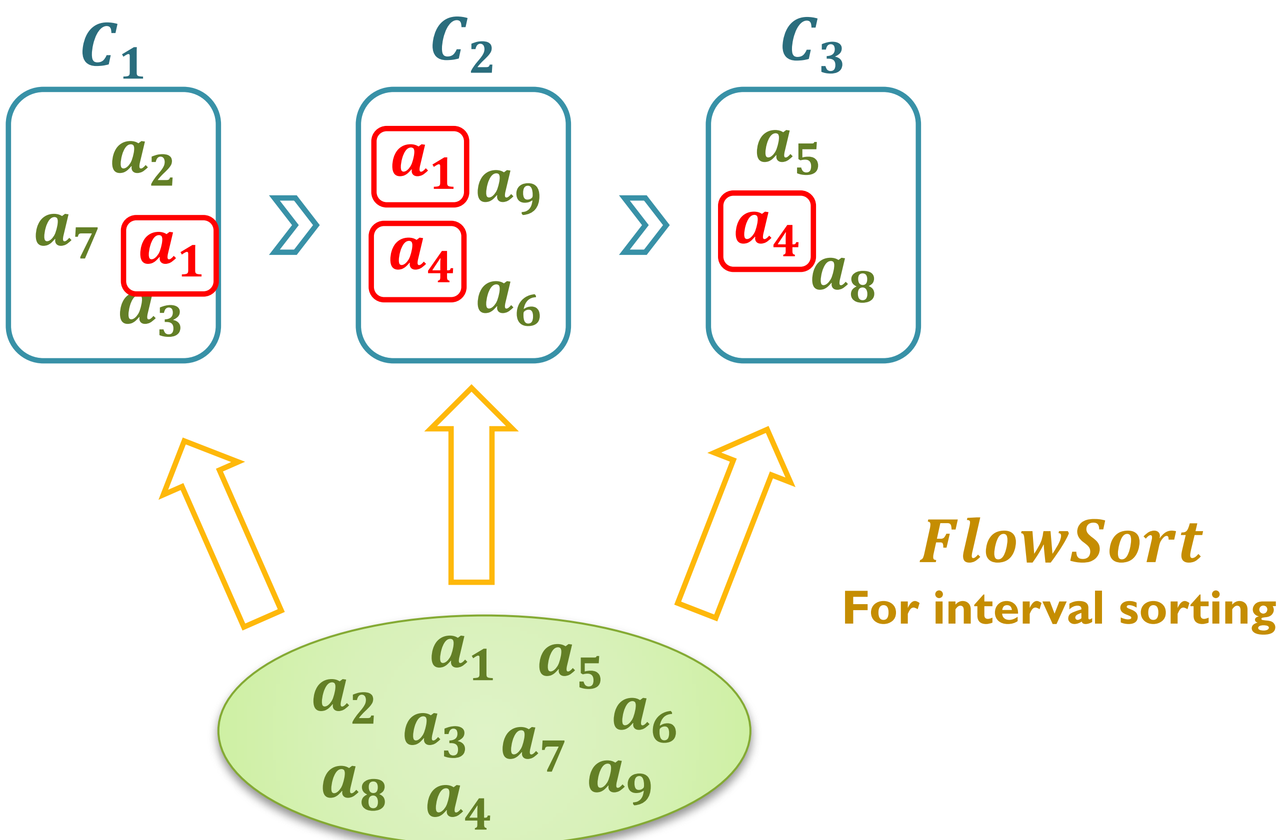
Sorting

- An alternative belongs to a single category.



Interval Sorting

- An alternative belongs to an interval of categories.



FlowSort

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

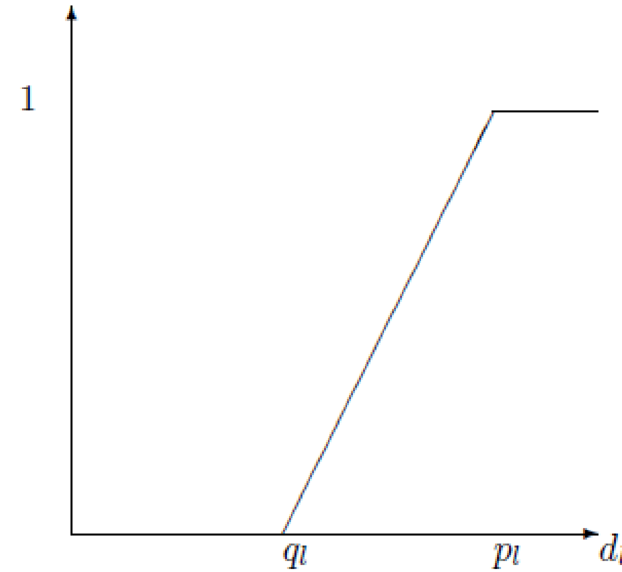
FlowSort: Sorting with Promethee

- A set of alternatives $A = \{a_1, a_2 \dots, a_n\}$;
- A set of criteria $\mathrm{F} = \{f_1, f_2 \dots, f_q\}$, $f_l(a): A \rightarrow \mathbb{R}$: 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:

$$\pi(a_i, a_j) = \sum_l w_l . \pi_l(a_i, a_j)$$

FlowSort: PROMETHEE

- Positive flow score:

$$\phi^{+}(a_i) = \frac{1}{n-1}\sum_{x \in A} \pi(a_i, x)$$

- Negative flow score:

$$\phi^{-}(a_i) = \frac{1}{n-1}\sum_{x \in A} \pi(x, a_i)$$

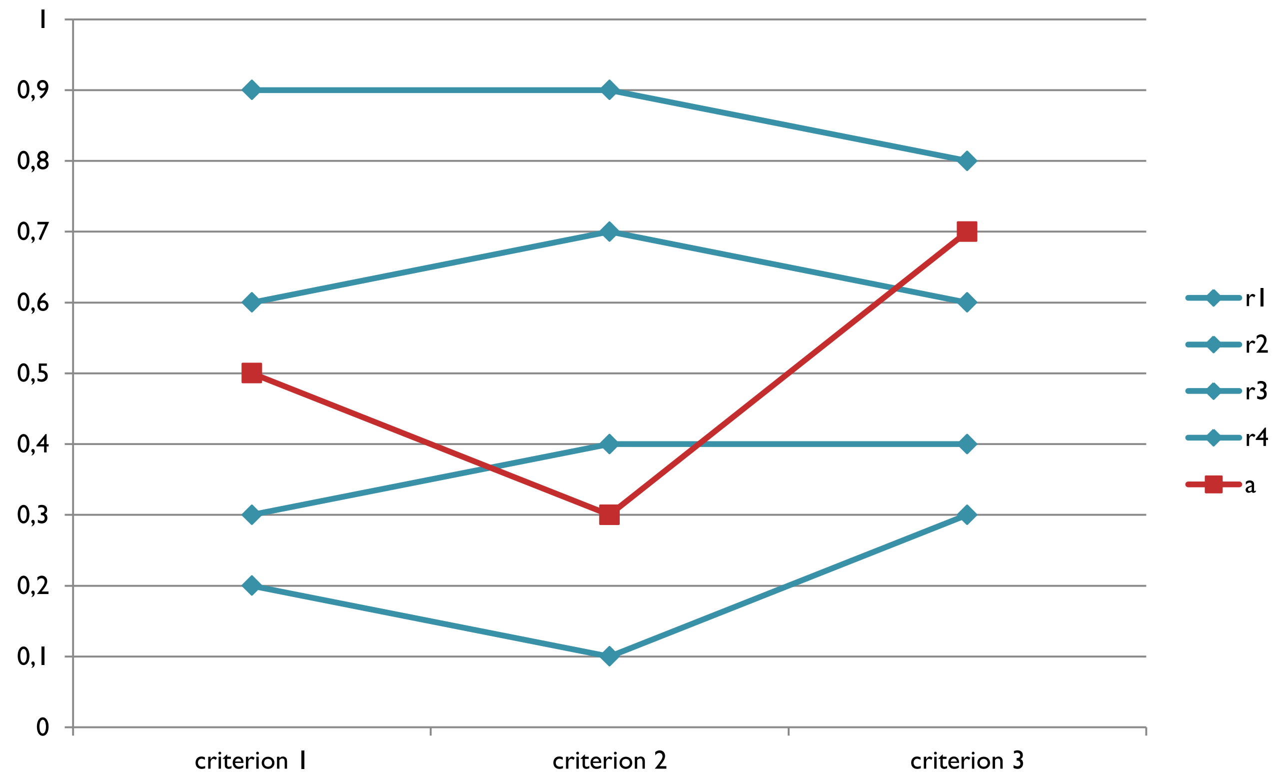
- Net flow score:

$$\phi(a_i) = \phi^{+}(a_i) - \phi^{-}(a_i)$$

FlowSort

- Extension of PROMETHEE to sorting;
- A set of predefined ordered categories $\mathrm{C} = \{c_1, c_2 \ldots, c_m\}$, $c_1 \succ c_2 \succ \cdots \succ c_m$;
- A set of central profiles $\mathrm{R} = \{r_1, r_2 \ldots, r_m\}$ representing the categories;
- Let's define $R_i = R \cup \{a_i\}$.

Example


1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
criterion 1
criterion 2
criterion 3
r1
r2
r3
r4
a

FlowSort

- Assignation rule:

$$h^*(a_i) = \underset{h=1,2,\dots,m}{\operatorname{argmin}} |\phi_{R_i}(a_i) - \phi_{R_i}(r_l)|$$

- Assignation rule for interval sorting:

$$h^{+*}(a_i) = \underset{h=1,2,\dots,m}{\operatorname{argmin}} |\phi^+{}_{R_i}(a_i) - \phi^+{}_{R_i}(r_l)|$$

$$h^{-*}(a_i) = \underset{h=1,2,\dots,m}{\operatorname{argmin}} |\phi^-{}_{R_i}(a_i) - \phi^-{}_{R_i}(r_l)|$$

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

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.

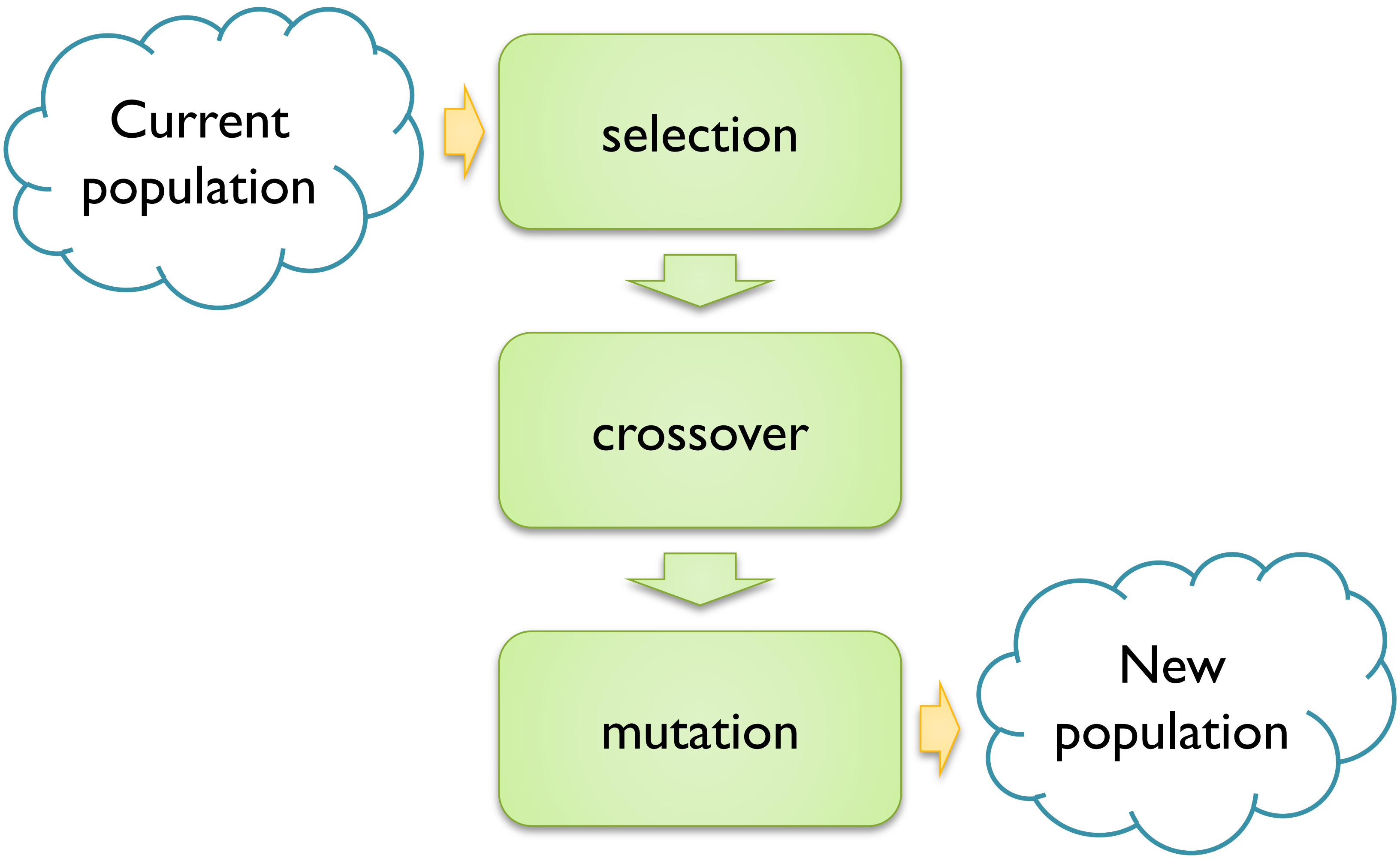
$$\sum_{a \in A} \left(\left| c_f^+(a) - c_r^+(a) \right| + \left| c_f^-(a) - c_r^-(a) \right| \right)$$

Algorithm: parameters to find

	$\boldsymbol{Crit_1}$	$\boldsymbol{Crit_2}$	$\boldsymbol{Crit_3}$	…	$\boldsymbol{Crit_q}$
Weight	w_1	w_2	w_3	…	w_q
q (indifference)	q_1	q_2	q_3	…	q_q
p (preference)	p_1	p_2	p_3	…	p_q
$\boldsymbol{r_1}$	r_{11}	r_{12}	r_{13}	…	r_{1q}
$\boldsymbol{r_2}$	r_{21}	r_{22}	r_{23}	…	r_{2q}
$\boldsymbol{r_3}$	r_{31}	r_{32}	r_{33}	…	r_{3q}
…	…	…	…	…	…
$\boldsymbol{r_m}$	r_{m1}	r_{m2}	r_{m3}	…	r_{mq}

$(3 + m) * q$ parameters to find

Genetic algorithm overview



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 λ:

$$O_1(i) = \lambda * p_1(i) + (1 - \lambda) * p_2(i)$$
$$O_2(i) = (1 - \lambda) * p_1(i) + \lambda * p_2(i)$$

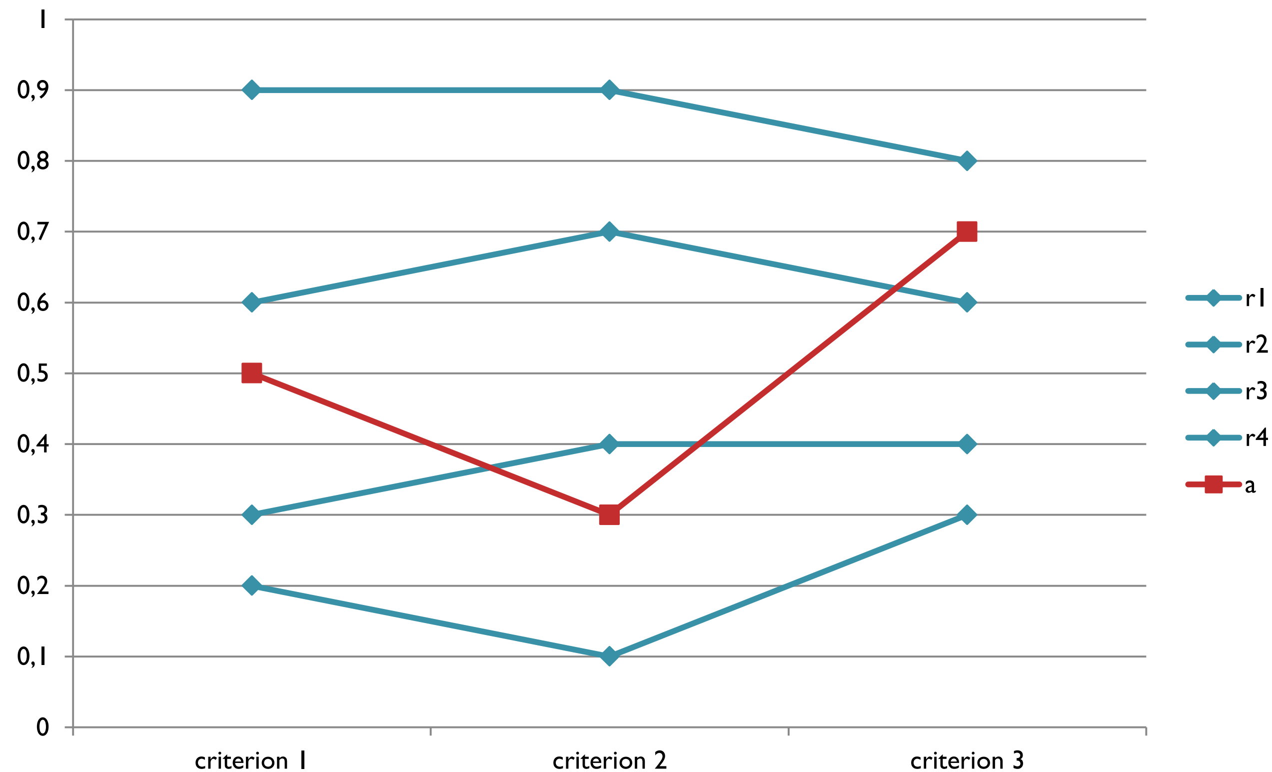
Solution pattern

	$Crit_1$	$Crit_2$	$Crit_3$	…	$Crit_q$
Weight	w_1	w_2	w_3	…	w_q
Q (indifference)	q_1	q_2	q_3	…	q_q
P (preference)	p_1	p_2	p_3	…	p_q
r_1	r_{11}	r_{12}	r_{13}	…	r_{1q}
r_2	r_{21}	r_{22}	r_{23}	…	r_{2q}
r_3	r_{31}	r_{32}	r_{33}	…	r_{3q}
…	…	…	…	…	…
r_m	r_{m1}	r_{m2}	r_{m3}	…	r_{mq}

Mutation operator

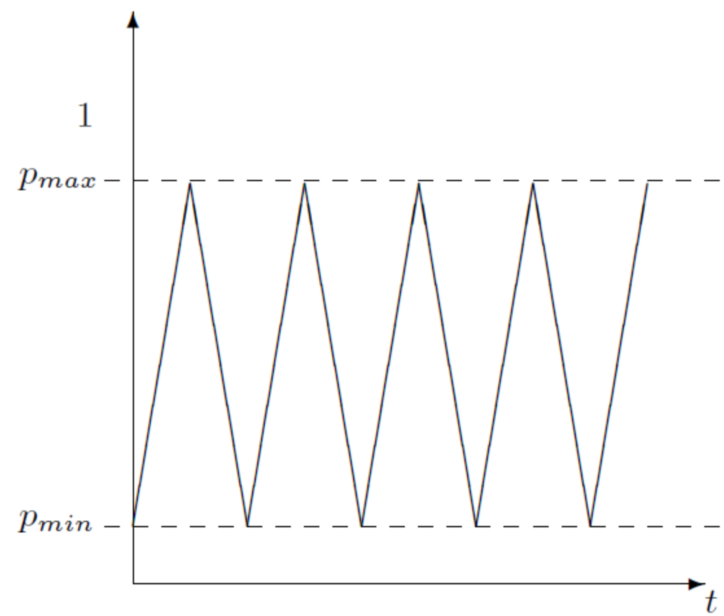
- 2 parameters:
  - mutation probability,
  - gene mutation probability.
- For each profile's values:
  - mutation depends on the percentage of overcategorization w.r.t. undercategorization of the category.
- Mutation range restricted with respect to the current correctness of the solution.

Mutation operator



Parameters' fine tuning

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



Testing procedure

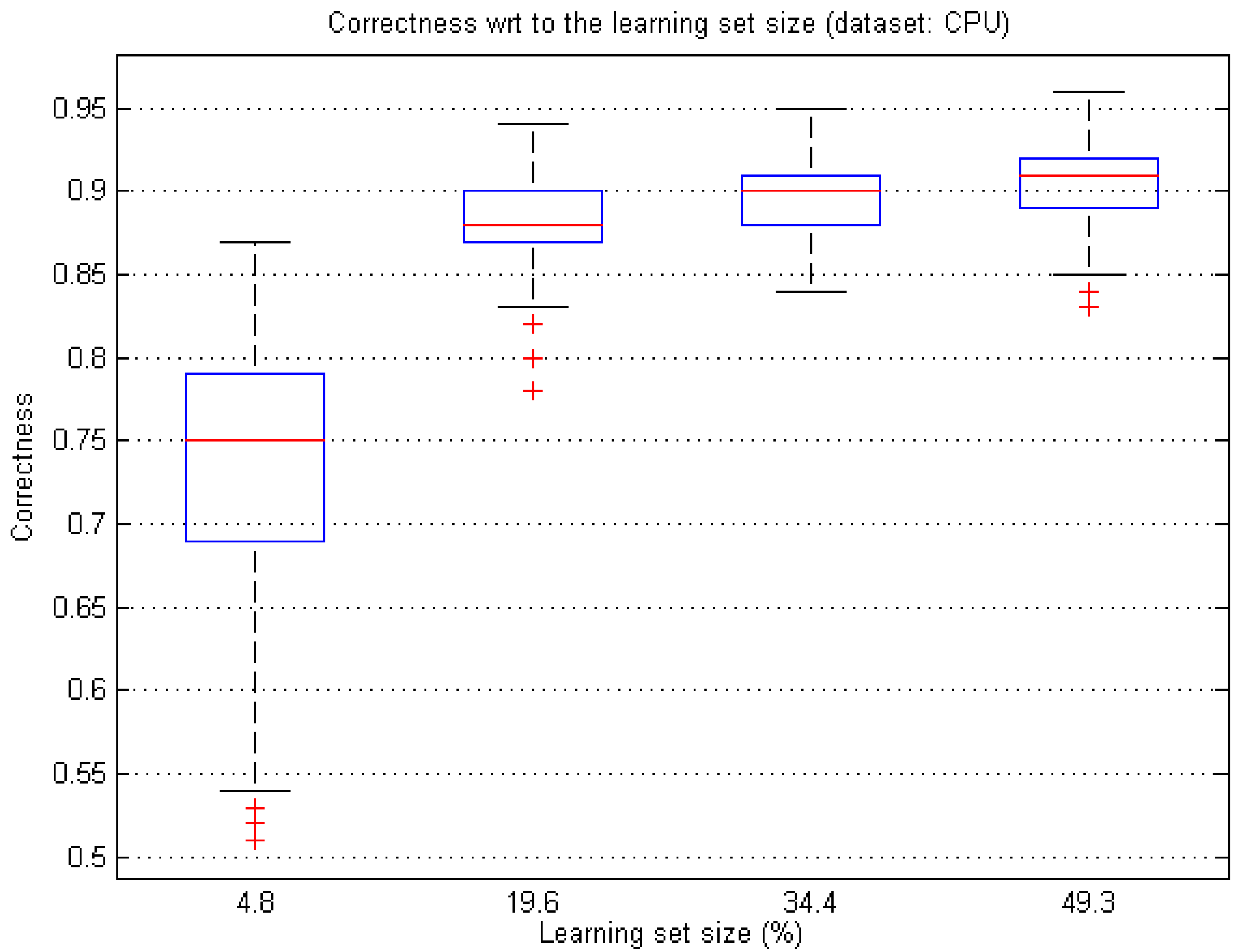
- 3 datasets have been chosen : CPU, BC, CEV.
- Interval sorting has been generated with a random parameters instantiation of FlowSort.

Dataset	#inst.	#crit.	#cat.	#param.	%imprecise cat.
CPU	209	6	4	42	40.67
BC	278	7	2	35	23.02
CEV	1728	6	4	42	16.43

Results

Learning set's size	Dataset	Correctness	Learning set correctness
5%	CPU	0,7337±0,0705	1,0000±0,0000
	BC	0,8827±0,0337	0,9981±0,0120
	CEV	0,8498±0,0224	0,9227±0,0383
20%	CPU	0,8798±0,0245	0,9880±0,0160
	BC	0,9463±0,0209	0,9955±0,0103
	CEV	0,8809±0,0173	0,8554±0,0338
35%	CPU	0,9004±0,0215	0,9642±0,0243
	BC	0,9579±0,0210	0,9919±0,0110
	CEV	0,8868±0,0154	0,8395±0,0277
50%	CPU	0,9065±0,0228	0,9581±0,0214
	BC	0,9747±0,0163	0,9913±0,0111
	CEV	0,8944±0,0168	0,8309±0,0252

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.