



# Genetic Programming for Multibiometrics

Romain Giot, Christophe Rosenberger

► **To cite this version:**

Romain Giot, Christophe Rosenberger. Genetic Programming for Multibiometrics. Expert Systems with Applications, Elsevier, 2012, 39 (2), pp.1837–1847. 10.1016/j.eswa.2011.08.066 . hal-00671952

**HAL Id: hal-00671952**

**<https://hal.archives-ouvertes.fr/hal-00671952>**

Submitted on 20 Feb 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Genetic Programming for Multibiometrics

Romain Giot\*, Christophe Rosenberger

*GREYC Laboratory  
ENSICAEN - University of Caen - CNRS  
6 Boulevard Maréchal Juin 14000 Caen Cedex - France*

---

## Abstract

Biometric systems suffer from some drawbacks: a biometric system can provide in general good performances except with some individuals as its performance depends highly on the quality of the capture... One solution to solve some of these problems is to use multibiometrics where different biometric systems are combined together (multiple captures of the same biometric modality, multiple feature extraction algorithms, multiple biometric modalities...). In this paper, we are interested in score level fusion functions application (i.e., we use a multi-biometric authentication scheme which accept or deny the claimant for using an application). In the state of the art, the weighted sum of scores (which is a linear classifier) and the use of an SVM (which is a non linear classifier) provided by different biometric systems provide one of the best performances. We present a new method based on the use of genetic programming giving similar or better performances (depending on the complexity of the database). We derive a score fusion function by assembling some classical primitives functions (+, \*, -, ...). We have validated the proposed method on three significant biometric benchmark datasets from the state of the art.

*Keywords:* Multibiometrics, Genetic Programming, Score fusion, Authentication.

---

---

\*Corresponding author

*Email addresses:* [romain.giot@ensicaen.fr](mailto:romain.giot@ensicaen.fr) (Romain Giot),  
[christophe.rosenberger@greyc.ensicaen.fr](mailto:christophe.rosenberger@greyc.ensicaen.fr) (Christophe Rosenberger)

## 1. Introduction

### 1.1. Objective

Every day, new evolutions are brought in the biometric field of research. These evolutions include the proposition of new algorithms with better performances, new approaches (cancelable biometrics, soft biometrics, ...) and even new biometric modalities (like finger knuckle recognition [1], for example). There are many different biometric modalities, each classified among three main families (even if we can find a more precise topology in the literature) :

- *biological* : recognition based on the analysis of biological data linked to an individual (e.g., DNA analysis [2], the odor [3], the analysis of the blood of different physiological signals, as well as heart beat or EEG [4]);
- *behavioural* : based on the analysis of an individual behaviour while he is performing a specific task (e.g., keystroke dynamics [5], online handwritten signature [6], the way of using the mouse of the computer [7], voice recognition [8], gait dynamics (way of walking) [9] or way of driving [10]);
- *morphological* based on the recognition of different particular physical patterns, which are, for most people, permanent and unique (e.g., face recognition [11], fingerprint recognition [12], hand shape recognition [13], or blood vessel [14], ...).

Nevertheless, there will always be some users for which a biometric modality (or method applied to this modality) gives bad results, whereas, they are better in average. These low performances can be implied by different facts: the quality of the capture, the instant of acquisition and the individual itself but they have the same implication (impostors can be accepted or user need to authenticate themselves several times on the system before being accepted). Multibiometrics allow to solve this problem while obtaining better performances (i.e., better security by accepting less impostors and better user acceptance by rejecting less genuine users) and by expecting that errors of the different modalities are not

29 correlated. In this paper, we propose a generic approach for multibiometric  
30 systems.

31 We can find different types of biometric multimodalities [15]. They use:

- 32 1. different sensors of the same biometric modality (i.e., capacitive or resistive  
33 sensors for fingerprint acquisition);
- 34 2. several different representations for the same capture (i.e., use of points  
35 of interest or texture for face or fingerprint recognition);
- 36 3. different biometric modalities (i.e., face and fingerprint recognition);
- 37 4. different instances of the same modality (i.e., left and right eye for iris  
38 recognition);
- 39 5. multiple captures (i.e., 25 images per second in a video used for face recog-  
40 nition);
- 41 6. an hybrid system composed of the association of the previous ones.

42 We are interested in the first four cases in this paper. Our objective is to  
43 automatically generate fusion functions which combine the scores provided by  
44 different biometric systems in order to obtain the most efficient multibiometrics  
45 authentication scheme.

## 46 1.2. Background

### 47 1.2.1. Performance Evaluation

48 In order to compare different multibiometrics systems, we need to present  
49 the how to evaluate them. Several works have already done on the evaluation of  
50 biometric systems [16, 17]. Evaluation is generally realized within three aspects:

- 51 • *performance*: it has for objective to measure various statistical criteria  
52 on the performance of the system (*Capacity* [18], *EER*, *Failure To En-*  
53 *roll (FTE)*, *Failure To Acquire (FTA)*, computation time, *ROC* curves,  
54 etc [17]);
- 55 • *acceptability*: it gives some information on the individuals' *perception*,  
56 *opinions* and *acceptance* regarding the system;

57     • *security*: it quantifies how well a biometric system (algorithms and de-  
58         vices) can resist to several types of logical and physical attacks such as  
59         Denial of Service (DoS) attack.

60     In this paper, we are only interested in performance evaluation (because the  
61     fusion approach is not modality dependant and perception and security depend  
62     on the used modalities). The main performance metrics are the following ones:

- 63     • *FAR (False Acceptance Rate)* which represents the ratio of impostors ac-  
64         cepted by the system;
- 65     • *FRR (False Rejection Rate)* which represents the ratio of genuine users  
66         rejected by the system;
- 67     • *EER (Error Equal Rate)* which is the error rate when the system is con-  
68         figured in order to obtain a *FAR* equal to the *FRR*;
- 69     • *ROC (Receiver Operating Characteristic)* curve which plots the *FRR* de-  
70         pending on the *FAR* and gives an overall overview of system performance;
- 71     • *AUC (Area Under the Curve)* which gives the area under the ROC curve.  
72         In our case, smaller is better. It is a way to globally compare performance  
73         of different biometric systems.

74     We can also present the *HTER* (Half Total Error Rate) which is the mean  
75     between the *FAR* and *FRR* for a given threshold (this error rate is interesting  
76     when we cannot get the *EER*).

### 77     1.2.2. *Biometric Fusion*

78     There are several studies on multibiometrics. The fusion can be operated on  
79     different points of the mechanism:

- 80     • *template fusion*: the templates captured by different biometric systems  
81         are merged together, then the learning process is realized on these new  
82         templates [19, 20]. Figure 1(a) presents this type of fusion. The fusion

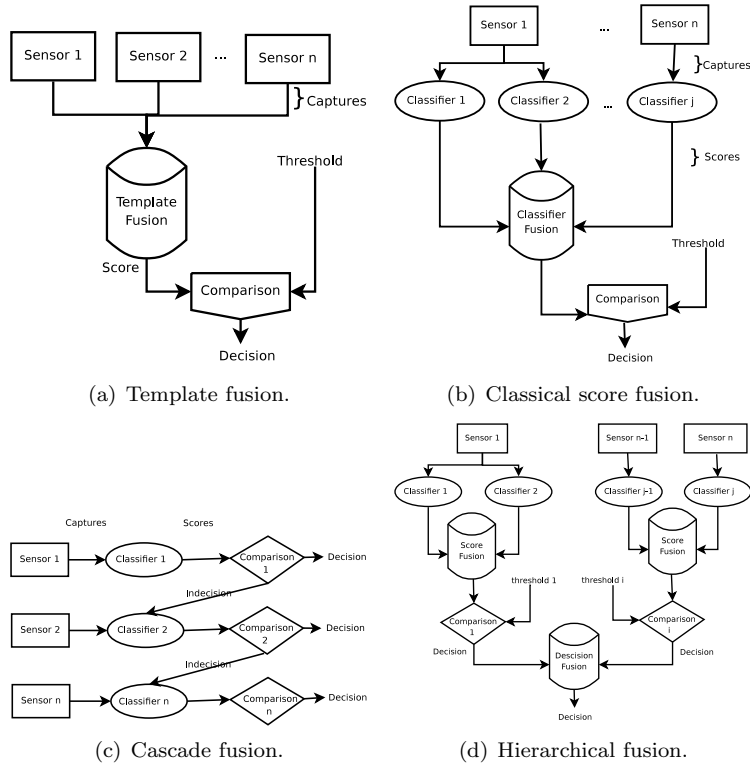


Figure 1: Illustration of different fusion mechanisms.

83 process is related to a feature selection in order to determine the most  
 84 significant patterns to minimize errors.

- 85 • *decision fusion*: the decision is taken for each of the biometric authentication system, then the final decision is done by fusing the previous ones [21].
- 88 • *rank fusion*: the decision is done with the help of different ranks of biometric identification systems. The main method is the majority vote [22].
- 90 • *score fusion*: the fusion is realized considering the output of the classifiers. The Figure 1(b) presents this type of fusion.

92 BuysSENS *et al.* [23] showed the interest of biometric fusion for face recognition  
 93 combining the image in visible and infrared color spaces with convolutional

94 neural networks. In [24], Mantalva and Freire have combined keystroke dynam-  
95 ics with voice recognition, it seems it is the first time that multibiometrics has  
96 been done with keystroke dynamics and another biometric modality. In [25],  
97 Hocquet *et al.* demonstrated the interest of fusion in keystroke dynamics in  
98 order to improve the recognition rates: three different keystroke dynamics func-  
99 tions are used on the same capture. The sum operator (consisting in summing  
100 the different scores) seems to be the most powerful approach in the literature.

101 These fusion architectures are quite simple but powerful. Results can yet be  
102 improved (in term of error rate or computation time) by using different archi-  
103 tectures. A cascade fusion [26] is another interesting approach. A first test is  
104 done, if the user is correctly verified as the attended client or if it is detected  
105 as an impostor, the algorithm stops. Otherwise, another biometric authentica-  
106 tion (with another capture from another modality) is proceeded until obtaining  
107 a decision of acceptance or rejection, or reaching the end of the cascade. So,  
108 instead of using one decision threshold, each test (except the last one) needs  
109 two thresholds: one for rejection and one for acceptance. All scores between  
110 these thresholds are considered in an indecision zone. This mechanism is pre-  
111 sented in Figure 1(c). Another advantage of this method is to decrease the  
112 verification time by not using all the modalities, they are used only if necessary.  
113 This method has been successfully applied on a multibiometric system using  
114 face and fingerprint recognition in a mobile environment (where acquisition and  
115 computation times are important) [26].

116 Another kind of architecture has been proposed: it is a hierarchical fusion  
117 scheme [27] (called multiple layers by their authors). Shen *et al.* have pre-  
118 sented this method with two different keystroke dynamics methods. The fusion  
119 is done at different steps, and involves different mathematical operations on  
120 scores (sum, weighted sum, product, min, max) and logical operations decision  
121 (comparison to a threshold, or, and) on differents templates extracted from the  
122 same capture. An extended version to any multibiometric system is presented  
123 in Figure 1(d). We think our work can be seen as a generalization of this paper.

124

125 It is also possible to model the distribution of the genuine and impostor  
126 matching scores, we talk about *Density-based score fusion*. In [28], scores are  
127 modelled with a Gaussian Mixture Model and have been tested on three multi-  
128 biometric databases involving face, fingerprint, iris and speech modalities.

129

130 Concerning non linear algorithms, Support Vector Machine (SVM) can also  
131 be used in a fusion process. Each score to combine is arranged in a vector  
132 and a training set is used to learn the SVM model. In [29], the SVM fusion  
133 to improve face recognition gives slightly better performances than weighted  
134 sum. Voice and online signature have been fused with SVM in [30]. In this  
135 experiment, arithmetic mean gives best results with noise free data, while SVM  
136 gives equivalent results with noisy data.

### 137 1.3. Discussion

138 In this paper, we are interested in biometric modality independent *transformation-*  
139 *based score fusion* [28] where the matching scores are first normalized and second  
140 combined. We have previously seen that in this case, arbitrary functions are  
141 often used. Our work is based on these various fusion architectures based on  
142 score fusion in order to produce a score fusion function automatically generated  
143 with genetic programming [31].

144

145 By the way, the definition of a fusion architecture is still an open issue  
146 in the multibiometrics research field [32], because the range of possible fusion  
147 configurations is very large. We think that using automatically generated fusion  
148 functions can bring a new solution to solve this kind of problems.

## 149 2. Material and Methods

150 In this section, we present all the required information in order to allow  
151 other researchers to reproduce our experiment.



152 *2.1. Biometric databases*

153 As it is well known that results can be highly related to the database, for this  
154 study, we have used three different multibiometric databases: the first one is the  
155 BSSR1 [33] distributed by the NIST [34] (referenced as BSSR1 in the paper),  
156 the second one is a database we have created for this purpose (referenced as  
157 PRIVATE in the paper) and the third one is a subset of scores computed with  
158 the BANCA [35] database (referenced as BANCA in the text. In fact, BANCA  
159 database is composed of templates. We have used the scores available in [36]).  
160 As all these databases are multi-modal, the scores are presented with tuples:  
161 the  $i$ th tuple of scores is represented as  $s_i = (s_i^1, s_i^2, \dots, s_i^n)$  for a database having  
162  $n$  modalities (in our case,  $n \in \{4, 5\}$ ).

163 The three databases are presented in detail in the following subsections while  
164 Table 1 presents a summary of their description.

165 *2.1.1. BSSR1 database*

166 The BSSR1 [33] database consists of an ensemble of scores sets from different  
167 biometric systems. In this study, we are interested in the subset containing  
168 the scores of two facial recognition systems and the two scores of a fingerprint  
169 recognition system applied to two different fingers for 512 users. We have 512  
170 tuples of intra-scores (comparison of the capture of an individual with its model)  
171 and  $512 * 511 = 261, 632$  tuples of inter-scores (comparison of the capture of an  
172 individual with the model of another individual). Each tuple is composed of 4  
173 scores:  $s = (s_{bssr1}^1, s_{bssr1}^2, s_{bssr1}^3, s_{bssr1}^4)$ , they respectively represent the score of  
174 the algorithm A of face recognition, the score of algorithm B of face recognition  
175 (the same face image is used for the two algorithms), the score of the fingerprint  
176 recognition with left index, the score of fingerprint recognition with right index.  
177 This database has been used several times in the literature [28, 37].

178 *2.1.2. PRIVATE database*

179 The second database is a chimeric one we have created by combining two  
180 public biometric template databases: the AR [38] for the facial recognition and

181 the GREYC keystroke [39] for keystroke dynamics.

182

183 The AR database is composed of frontal facial images of 126 individuals  
184 under different facial expression, illumination conditions or occlusions. This is  
185 a quite difficult database in reason of these specificities. These images have  
186 been taken during two different sessions with 13 captures per session. The  
187 GREYC keystroke contains the captures on several session during a two months  
188 period involving 133 individuals. Users were asked to type the password "greyc  
189 laboratory" 6 times on a laptop and 6 times on an USB keyboard by interleaving  
190 the typings.

191 We have selected the first 100 individual of the AR database and we have  
192 associated each of these individuals to another one in a subset of the GREYC  
193 keystroke database having 5 sessions of captures. We then used the 10 first  
194 captures to create the model of each user and the 16 remaining ones to compute  
195 the intra and inter scores.

196 These scores have been computed by using two different methods for the  
197 face recognition (the scores  $s_{private}^1$  and  $s_{private}^2$  and three different ones for the  
198 keystroke dynamics ( $s_{private}^3$ ,  $s_{private}^4$  and  $s_{private}^5$  scores). The face recognition  
199 algorithms are based on eigenfaces [11] and SIFT keypoints [40] comparisons  
200 between images from the model and the capture [41]. Keystroke dynamics scores  
201 have been computed by using different methods [42] based on SVM, statistical  
202 information and rhythm measures.

### 203 2.1.3. BANCA database

204 The latest used benchmark is a subset of scores produced by the help of  
205 the BANCA database [36]. The selected scores correspond to the following  
206 one labelled: IDIAP\_voice\_gmm\_auto\_scale\_25\_100\_pca.scores for  $s_{banca}^1$ , SUR-  
207 REY\_face\_nc\_man\_scale\_100.scores for  $s_{banca}^2$ , SURREY\_face\_svm\_man\_scale\_0.13.scores  
208 for  $s_{banca}^3$  and  
209 UC3M\_voice\_gmm\_auto\_scale\_10\_100.scores for  $s_{banca}^4$ .

210 We have empirically chosen this subset. G1 set is used as the learning set,

Table 1: Summary of the different databases used to validate the proposed method

<b>Nb of</b>	<b>BSSR1</b>	<b>PRIVATE</b>	<b>BANCA</b>
users	512	100	208
intra tuple	512	1600	467
inter tuple	261632	158400	624
items/tuples	4	5	4

211 while G2 set is used as the validation set. Users from G1 are different than users  
 212 from G2.

#### 213 *2.1.4. Discussion*

214 The main differences between these three benchmarks are:

- 215 • the biometric modalities used in BSSR1 and BANCA have better perfor-  
 216 mances than the ones in PRIVATE;
- 217 • the quantity of intra-scores is more important in PRIVATE (only one tuple  
 218 of intra-score per user in BSSR1 instead of several in PRIVATE);
- 219 • BSSR1 and BANCA are databases of scores (by the way, we do not know  
 220 the biometric systems having generated them) whereas PRIVATE is a  
 221 database of templates (we had to compute the scores);
- 222 • BSSR1 and BANCA are more adapted to physical access control appli-  
 223 cations (i.e., a building is protected by a multi-modal biometric system),  
 224 while PRIVATE is more adapted to logical access control (i.e., the au-  
 225 thentication to a Web service is protected by a multi-modal biometric  
 226 system).

227 In the following subsections, we describe the proposed methodology to auto-  
 228 matically generate a score fusion function with genetic programming. We adopt  
 229 the classical score fusion context described in Figure 1(b). Before using the  
 230 scores provided by different biometric systems, we need to normalize them.

#### 231 *2.2. Score Normalization*

232 It is necessary to normalize the various scores before operating the fusion pro-  
 233 cess: indeed, these scores come from different classifiers and their values do not

234 necessarily evolve within the same interval. We have chosen to use the *tanh* [43]  
 235 operator to normalize the scores of each modality. Equation (1) presents the  
 236 normalization method, where  $\mu_{gen}^m$  and  $\sigma_{gen}^m$  respectively represents the average  
 237 and standard deviation of the genuine scores of the modality  $m$ . The genuine  
 238 scores are obtained by comparing the model and the capture of the same user:  
 239 they are also called the *intra scores*. In opposition, the *inter scores* are obtained  
 240 by comparing the model of a user with the capture of other users.  $score'$  and  
 241  $score$  respectively represents the scores after and before normalisation.

$$score' = \frac{1}{2} \left\{ \tanh \left( \frac{1}{100} \left( \frac{score - \mu_{gen}^m}{\sigma_{gen}^m} \right) + 1 \right) \right\} \quad (1)$$

242 We have selected this normalization procedure from the state of the art  
 243 because it is known to be stable [44] and does not use impostors patterns which  
 244 can be hard or impossible to obtain in a real application. The aim of this  
 245 paper is not to analyse the performance of biometric systems depending on the  
 246 normalization procedure, but to present a new multibiometrics fusion procedure.  
 247 The scores of each modality have been normalized using this procedure.

### 248 2.3. Fusion Procedure

249 In this study, we have chosen to use genetic programming [31] in order to  
 250 generate score fusion functions. Genetic programming belongs to the family of  
 251 evolutionary algorithms and its scheme is quite similar to the one of genetic  
 252 algorithms [45]: a population of computer programs (possibly represented by a  
 253 tree) evolves during several generations; different genetic operators are used to  
 254 create the new population. Programs are evaluated by using a fitness function  
 255 which produces a value that is used for their comparisons and gives a probability  
 256 of selection during the tournaments. In a system where the computer programs  
 257 are represented by trees, their leaves mainly represent the entries of the problem,  
 258 the root gives the solution to the problem and the other nodes are the various  
 259 functions taking into arguments the values of their children nodes.

260 The leaves are called terminals and can be of several kinds: (a) pseudo-  
 261 variables containing the real entries of the problem (in our case, the list of

262 scores of each modality), (b) some constants possibly randomly generated, (c)  
263 functions without any arguments having any side effect, or (d) some ordinary  
264 variables.

265 The different genetic operators usually used during the evolution are (a)  
266 the crossover, where randomly choose sub-trees have two different trees are  
267 exchanged, (b) the mutation, where a sub-tree is destroyed and replaced by  
268 another one randomly generated, or (c) the copy, where the tree is conserved in  
269 the next generation. The different steps of a genetic programming engine are  
270 presented as following:

- 271 1. An initial population is randomly generated. This population is composed  
272 of computer programs using the available functions and terminals. The  
273 trees are built using a recursive procedure.
- 274 2. The following steps are repeated until the termination criterion is satis-  
275 fied (the fitness function has reached the right value, or we reached the  
276 maximum number of generations).
  - 277 (a) Computation of the fitness measure of each program (the program-  
278 ming is evaluated according to its input data).
  - 279 (b) Selection of programs with a probability based on their fitness to  
280 apply them the genetic operations.
  - 281 (c) Creation of the new generation of programs by applying the follow-  
282 ing genetic operations (depending on their probabilities) to the pre-  
283 viously selected programs:
    - 284 • Reproduction: the individual is copied to the new population.
    - 285 • Crossover: A new offspring program is created by recombining  
286 randomly chosen parts from two select programs. An example is  
287 provided in Figure 2.
    - 288 • Mutation: A new offspring program is created by mutating one  
289 node of the selected program at a randomly chosen place. An  
290 example is provided in Figure 3.
- 291 3. the single best program of the whole population is designated as the win-  
292 ner. This can be the solution or an approximate solution to the problem.

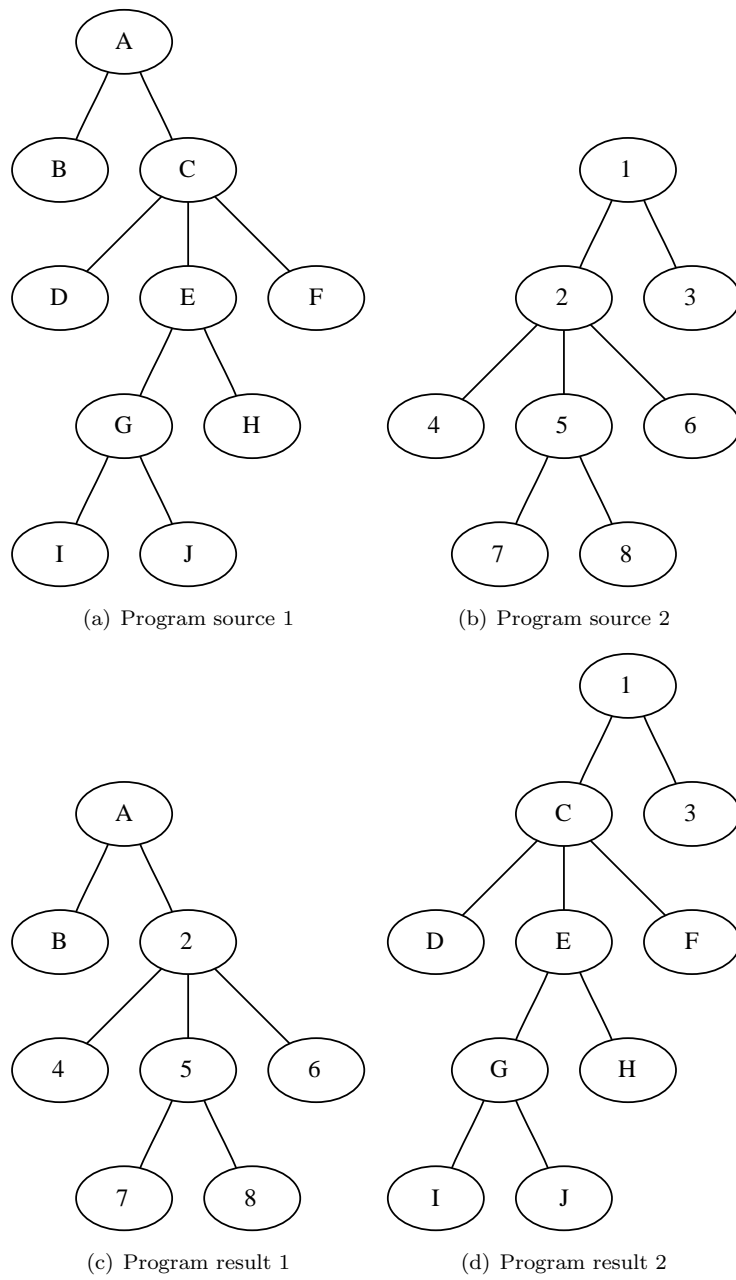


Figure 2: Crossover in genetic programming: node C from tree 1 is exchanged with node 2 from tree 2. Program result 1 is the new individual to add to the new generation.

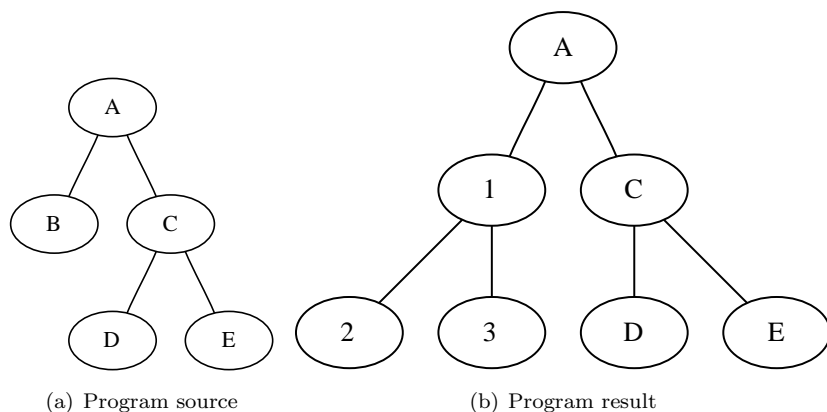


Figure 3: Mutation in genetic programming: node B is replaced by another sub-tree.

294 Different applications to genetic programming are presented in [46] as well  
 295 as their bibliographic references. The fields of these applications can be listed  
 296 in curve fitting, data modelling, symbolic regression, image and signal process-  
 297 ing, economics, industrial process control, medicine, biology, bioinformatics,  
 298 compression... but, it seems, so far of our knowledge, that it has not been  
 299 yet applied to multibiometrics. We only found one reference on genetic pro-  
 300 gramming in the biometrics field. In this paper [47], authors have used genetic  
 301 programming to learn speaker recognition programs. They have used an island  
 302 model where different islands operate their genetic programming evolution, and,  
 303 after each generation some individuals are able to leave to another island. The  
 304 obtained performance was similar to the state of the art in speaker recognition  
 305 in normal conditions, but, the generated systems performed better in degraded  
 306 conditions.

307 More information about the configuration of the genetic programming sys-  
 308 tem is presented in the next section.

#### 309 2.4. Parameters of the Genetic Programming

310 We want to use a score fusion function that returns a score related to the  
 311 performance of a multibiometric system. This score has to be compared with a  
 312 threshold in order to make the decision of acceptance or rejection of the user.

313 In this case, none logical operation is required in the generated programs and  
314 different information can be extracted from the result of the fusion function (we  
315 can compute the ROC curve, the EER, ...).

#### 316 2.4.1. Fitness Function

317 The EER (Error Equal Rate) is usually used to compare the performance  
318 of different biometric systems together. A low EER means that FAR and FRR  
319 are both low and the system has a good performance if its threshold is config-  
320 ured accordingly to obtain this value. For this reason, we have chosen to use  
321 this running point to evaluate the performance of the generated score fusion  
322 functions.

323 To compute the EER, we consider the highest and lowest values in the final  
324 scores generated by the genetic programming. Then, we set a threshold at the  
325 lowest score and linearly increment it until obtaining the highest score value in  
326 1000 steps. For each of these steps, we compute the FAR (comparison between  
327 the threshold and the inter scores) and FRR (comparison between the threshold  
328 and the intra scores). The ROC curve can be obtained by plotting all these  
329 couples of (FAR, FRR), while the EER is the mean of FAR and FRR for the  
330 couple having the lowest absolute difference. So, the fitness function is  $fitness =$   
331  $(FAR_i + FRR_i)/2$ , where  $i$  is the threshold for which  $abs(FAR_i - FRR_i)$  is  
332 minimal.

#### 333 2.4.2. Genetic Programming Parameters

334 In this section, we present the various parameters used in the genetic pro-  
335 gramming algorithm. Table 2 presents the various parameters of the evolution-  
336 ary algorithm.

337 To achieve this experiment, we used the PySTEP [48] library. The generated  
338 programs contain basic functions (+, -, \*, /, *min*, *max*, *avg*). The terminals  
339 are the scores of the biometric systems and random constants between 0 and 1.

340 The whole fitness cases are completed with a single tree evaluation, thanks to  
341 the numpy [49] library. Each fitness case is a tuple of scores (where each score



Table 2: Summary of the configuration of the genetic programming iterations. Numbers used in function set can be scores or constants.

Configuration	Values						
<b>Objective</b>	Generates a function producing a multibiometrics score.						
<b>Functions set</b>	<ul style="list-style-type: none"> <li>• +: addition of two numbers, two numbers,</li> <li>• -: subtraction of two numbers,</li> <li>• *: multiplication of two numbers, • <i>max</i>: returns the maximum of two numbers,</li> <li>• /: division of two numbers, • <i>avg</i>: returns the mean of two numbers</li> <li>• <i>min</i>: returns the minimum of</li> </ul>						
<b>Fitness function</b>	Computes the EER of the multibiometric system						
<b>Terminal set</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: left;">BSSR1</th> <th style="width: 33%; text-align: left;">PRIVATE</th> <th style="width: 33%; text-align: left;">BANCA</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{bssr1}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{bssr1}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{bssr1}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{bssr1}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• <i>a, b, c</i>: keystroke dynamics scores (<math>s_{private}^3, s_{private}^4, s_{private}^5</math>),</li> <li>• <i>d, e</i>: face recognition scores (<math>s_{private}^1, s_{private}^2</math>),</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{banca}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{banca}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{banca}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{banca}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul> </td> </tr> </tbody> </table>	BSSR1	PRIVATE	BANCA	<ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{bssr1}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{bssr1}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{bssr1}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{bssr1}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>a, b, c</i>: keystroke dynamics scores (<math>s_{private}^3, s_{private}^4, s_{private}^5</math>),</li> <li>• <i>d, e</i>: face recognition scores (<math>s_{private}^1, s_{private}^2</math>),</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{banca}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{banca}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{banca}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{banca}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>
BSSR1	PRIVATE	BANCA					
<ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{bssr1}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{bssr1}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{bssr1}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{bssr1}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>a, b, c</i>: keystroke dynamics scores (<math>s_{private}^3, s_{private}^4, s_{private}^5</math>),</li> <li>• <i>d, e</i>: face recognition scores (<math>s_{private}^1, s_{private}^2</math>),</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>a</i>: scores from <math>s_{banca}^1</math>,</li> <li>• <i>b</i>: scores from <math>s_{banca}^2</math>,</li> <li>• <i>c</i>: scores from <math>s_{banca}^3</math>,</li> <li>• <i>d</i>: scores from <math>s_{banca}^4</math>,</li> <li>• 50 constants linearly distributed between 0 and 1.</li> </ul>					
<b>Initial population</b>	500 random trees with a depth between 2 and 8 built with the ramped half and half method.						
<b>Evolution parameters</b>	<ul style="list-style-type: none"> <li>• Number of individuals: 500,</li> <li>• Maximal number of generations: 50,</li> <li>• Depth limited to: 8,</li> <li>• Probability of crossover: 45%,</li> <li>• Probability of mutation: 50%</li> <li>• Probability of reproduction: 5% (with elitism),</li> <li>• Selection: tournament of size 10 with a selection probability of 80%.</li> </ul>						
<b>Termination criterion</b>	Best individual has a fitness inferior at 0.001 (by the way, this value would never be met ...) or maximal number of generations reached.						
<b>Learning set</b>	First half of the intra-scores tuples and first half of the inter-scores tuples.						
<b>Validating set</b>	Second half of the intra-scores tuples and second half of the inter-scores tuples.						

342 comes from a different biometric modality) and its result value is the score  
343 returned by the generated multimodal system. The global fitness value of a tree  
344 is the EER value computed with the previously generated scores (computation  
345 of the ROC curve, then reading of the EER value from it).

346 PySTEP is a strongly typed genetic programming engine, but, in our case,  
347 we do not use any particular constraints: the root node can only have a function  
348 as child (no terminal in order to avoid an unimodal system, and any function of  
349 the set), while the other function nodes can have any of the functions as children  
350 as well as any of the terminals.

351 The maximal depth of the generated trees is set to 8. In order to avoid  
352 to stay in a local minimal solution, the mutation probability is set to 50%.  
353 500 individuals evolve during 50 generations. We have set this few quantities,  
354 because during our investigations, using a population of 5000 individuals on  
355 100 generations did not give so much better results (gain not interesting in  
356 comparison to the computation time). Each database has been splitted in two  
357 sets of equal size: the first half is the learning set and the second half is the  
358 validation set.

359 The mutation rate is set to 50%, the cross-over rate to 45% and the repro-  
360 duction rate to 5%. For mutation and cross-over the individuals are selected  
361 with a tournament of size 10 with a probability of 80% to select the best individ-  
362 ual. The same individual can be selected several times. For the reproduction,  
363 the individuals are selected with an elitism scheme: the 5% best individuals are  
364 copied from generation  $n - 1$  to generation  $n$ . During a crossover, only the first  
365 offspring (of the two generated ones) is kept.

### 366 **3. Results**

367 In this section, we present the results of the generated fusion programs on  
368 the three benchmark data sets.

369 The results are compared to other functions from the state of the art: (a)  
370 the *min* rule which returns the minimum score value, (b) the *mul* rule which

371 returns the product of all the scores, (c) the *sum* rule which returns the sum  
 372 of the scores, (c) the *weight* rule which returns a weighted sum, and (d) an  
 373 SVM implementation. The weights of the weighted sum have been configured by  
 374 using genetic algorithm on the training sets [50, 51] (in order to give the best  
 375 results as possible). The fitness function is the value of the EER and the genetic  
 376 algorithm engine must lower this value. Table 3 presents the configuration of  
 377 the genetic algorithm.

Table 3: Configuration of the genetic algorithm to set the weights of the weighted sum

Parameter	Value
Population	5000
Generations	500
Chromosome signification	weights of the fusion functions
Chromosome values interval	$[-10; 10]$
Fitness	EER on the generated function
Selection	normalized genetic selection (probability of 0.9)
Elitism	True

378 For the SVM, we have computed the best parameters (i.e., search the  $C$   
 379 and  $\gamma$  parameter giving the lowest error rate) using the learning database on  
 380 a 5-fold cross validation scheme. We have used the *easy.py* script provided  
 381 with libSVM [52] for this purpose. We have then tested the performance on the  
 382 validation set. We only obtain on functional point (and not a curve) when using  
 383 an SVM. That’s why we have used the HTER instead of the EER.

384 Table 4 presents the performances, for the three databases, of each biometric  
 385 systems, fusion mechanisms from the state of the art, and our contribution.

386 Concerning the state of the art performances, can see that the simple fusion  
 387 functions *sum* and *mul* tend to give better performances compared to the best  
 388 biometric method of each database, but they are outperform by the *weight* rule.  
 389 The *min* operator gives quite bad results (it does not improve the best biometric

390 system). The *SVM* method gives good results but is outperform by the *weight*  
391 method.

392 Table 5 presents the gain of performance against the *weight* operator (which  
393 gives the best results in Table 4) in term of EER and AUC.

This gain is computed as following:

$$gain = 100 \frac{(EER_{weight} - EER_{gpfunc})}{EER_{weight}} \quad (2)$$

394 where  $EER_{weight}$  and  $EER_{gpfunc}$  are respectively the EER values of the weighted  
395 fusion and the generated score fusion function (the same procedure is used for  
396 the AUC). Better values than the weighted sum are represented in bold. The  
397 EER gives a local performance for one running point (system configured in or-  
398 der to obtain an FAR equal to the FRR), while the AUC gives a gives a global  
399 performance of the whole system. These two information are really interesting  
400 to use when comparing biometric systems. Figure 4 presents the ROC curves  
401 of the generated programs against the weighted sum. Performance of the initial  
402 biometric systems are not represented, because we have already seen that they  
403 are worst than the weighted sum (same remark for the other fusion functions).  
404 Logarithmic scales are used, because error rates are quite small.

405 We can see from Table 5 and Figure 4 that most of the time, the automati-  
406 cally generated functions with genetic programming give slightly better results  
407 than the weighted sum. These improvements can be local and global and vary  
408 between 16% and 59% for the EER and 0.05% and 76% for the area under  
409 the curve. When there is no improvement, the results are equal or (in one  
410 case) slightly inferior. Even if there is some difference between training (not  
411 represented in this paper) and validating sets, we cannot observe overfitting  
412 problem. The BSSR1 dataset presents the largest difference of performance  
413 between training and validation sets, but, the results are still better than the  
414 ones from the state of the art (and the same problem can be observe with the  
415 weighted sum). By the way, the fitness criterion has never been met, we did  
416 not achieve to obtain fusion functions doing no error. So, the evolution always

Table 4: Performance (HTER in %) of the initial methods ( $s_*^1, s_*^2, s_*^3, s_*^4, s_*^5$ ), the state of the art fusion functions ( $sum, min, mul, weight$ ) and our proposal on the three databases. Bold values represent better performance than the initial biometric systems, and \* represents fusion results better than state of the art.

(a) BSSR1

Method	HTER	
<b>BSSR1</b>		
Biometric systems	$s_{bssr1}^1$	04.30%
	$s_{bssr1}^2$	06.19%
	$s_{bssr1}^3$	08.41%
	$s_{bssr1}^4$	04.54%
	$s_{bssr1}^5$	04.54%
Fusion functions	$sum$	<b>00.70%</b>
	$min$	05.04%
	$mul$	<b>00.70%</b>
	$weight$	<b>00.38%</b>
	$SVM$	<b>0.77%</b> (FAR=1.16%, FRR=0.39%)
Proposal	gpI	<b>0.40%</b>

(b) PRIVATE

Method	HTER	
<b>PRIVATE</b>		
Biometric systems	$s_{private}^1$	8.92%
	$s_{private}^2$	11.53%
	$s_{private}^3$	15.69%
	$s_{private}^4$	06.21%
	$s_{private}^5$	31.43%
	$s_{private}^6$	31.43%
Fusion functions	$sum$	<b>02.70%</b>
	$min$	13.72%
	$mul$	<b>02.67%</b>
	$weight$	<b>02.26%</b>
	$SVM$	<b>05.47%</b> (FAR=10.87, FRR= 0.07%)
Proposal	gpA	<b>01.57%*</b>

(c) BANCA

Method	HTER	
<b>BANCA</b>		
Biometric systems	$s_{banca}^1$	04.38%
	$s_{banca}^2$	11.54%
	$s_{banca}^3$	08.97%
	$s_{banca}^4$	07.32%
	$s_{banca}^5$	07.32%
Fusion functions	$sum$	<b>01.28%</b>
	$min$	04.38%
	$mul$	<b>01.28%</b>
	$weight$	<b>00.91%</b>
	$SVM$	<b>01.01%</b> (FAR= 1.71 %, FRR=0.32%)
Proposal	gpΦ	<b>00.75%*</b>

Table 5: Performance gain between our proposal and the weighted sum (which gives the best results in the methods of the state of the art).

Database	EER	AUC
BSSR1	-5.26%	<b>0.05%</b>
PRIVATE	<b>34.85%</b>	<b>23.85%</b>
BANCA	<b>17.58%</b>	<b>76.74%</b>

417 ended when reaching the 50<sup>th</sup> generation.

418 Figure 5 represents the fitness evolution during all the generations of one  
 419 genetic programming run on the BSSR1 database. A logarithmic scale has been  
 420 used to give more importance to the low values and track easier the fitness  
 421 evolution of the best individual of each generation. We can observe the same  
 422 kind of results with the other databases. The fitness convergence appears several  
 423 generations before the end of the computation. The worst program of each  
 424 generation is always very bad which implies that the standard deviation of the  
 425 fitness is also always quite huge. This can be explained by the high quantity of  
 426 mutation probability and the low quantity of good programs kept for the next  
 427 generation. When running the experiment several times, we obtain the same  
 428 convergence value. We can say that we reach the maximum performance of the  
 429 system.

#### 430 4. Discussion

431 The score fusion functions generated by the proposed approach give a slightly  
 432 better performance than the fusion functions used in the state of the art in multi-  
 433 biometrics. We can argue that genetic programming is adapted to automatically  
 434 define score fusion functions returning a score. The tradeoff of this performance  
 435 gain is the need of training patterns which are not necessary for *sum*, *mul* or  
 436 *min* (but this requirement is already present for the weighted sum or the use  
 437 of an SVM). By the way, this is not really a problem, because we already need  
 438 training patterns to configure the threshold of decision (if we do not want to do  
 439 it empirically) or if we need to normalize the scores before doing the fusion.

440 Another problem inherent to genetic programming is the complexity of the

441 generated programs. It is probable that some subtrees could be pruned or sim-  
442 plified without losing performance. Another trail would be to add regulariza-  
443 tion parameter to the fitness function (for example, the number of nodes or the  
444 depth of the tree). Generated programs would be more readable by a human  
445 and quicker to interpret. Figure 6 presents a simple generated tree (depend-  
446 ing on the database, they can be more or less complex). Even if the program  
447 is quite short (comparing to the other generated functions), it includes useless  
448 code (e.g., the subtree  $avg(a, a - 1/12)$  could be simplified by  $a - 1/24$ ). Some  
449 generated trees include preprocessing steps by not using all the modalities in  
450 the terminal set.

451 Genetic programming generated score fusion functions give performance  
452 slightly equal or better than genetic algorithm configured weighted sum. Even  
453 if computation time is more important than for genetic algorithm, we can think  
454 that the gain is not really important between the two methods, but, to obtain  
455 these results, genetic programming needed a population ten times smaller and  
456 ten times less of generations.

## 457 5. Conclusion

458 We propose in this paper a new approach for multibiometrics based on the  
459 automatic generation of score fusion functions. We have seen interesting ap-  
460 proaches in the state of the art and decided to improve them by automatically  
461 generated score fusion programs by the help of genetic programming.

462 Our contribution concerns the designing of multibiometric systems while  
463 using a generic approach based on genetic programming (and is inspired from the  
464 state of the art architectures). The proposed method returns a multibiometrics  
465 score to be compared with a defined threshold. The proposed multibiometric  
466 system has been heavily tested on three different multibiometric databases. We  
467 obtained great improvements compared to classical fusion functions used in the  
468 state of the art. We hope to have opened a new path in the fusion of biometric  
469 systems thanks to genetic programming.

470 Results could surely be improved by using different parameters in the genetic  
471 programming engine (i.e., more individuals and generations, different range of  
472 constants, different functions, . . . ). It could be interesting to test other perfor-  
473 mance metrics could be improved by adding quality measures of the capture,  
474 and if genetic programming could produce template fusion programs.

## 475 **6. Acknowledgment**

476 The authors would like to thank: the author of pySTEP [48], the library  
477 used during the experiment, for his helpfull help when encountering problems  
478 with it, the authors of the various biometric databases used in this experiment,  
479 as well as the French Basse-Normandie region for its financial support.

## 480 **References**

- 481 [1] A. Kumar, Y. Zhou, Human Identification Using KnuckleCodes, in: IEEE  
482 International Conference on Biometrics: Theory, Applications and Systems  
483 (BTAS 2009), 2009.
- 484 [2] M. Hashiyada, Developement of biometric dna ink for authentication secu-  
485 rity, *Tohoku J. Exp. Med.* 204 (2004) 109–117.
- 486 [3] Z. Korotkaya, Biometric person authentication: Odor, Tech. rep., De-  
487 partment of Information Technology, Laboratory of Applied Mathematics,  
488 Lappeenranta University of Technology (2003).
- 489 [4] A. Riera, A. Soria-Frisch, M. Caparrini, C. Grau, G. Ruffini, Unobtru-  
490 sive biometric system based on electroencephalogram analysis, *EURASIP*  
491 *Journal on Advances in Signal Processing* 2008 (2008) 8.
- 492 [5] R. Gaines, W. Lisowski, S. Press, N. Shapiro, Authentication by keystroke  
493 timing: some preliminary results, Tech. rep., Rand Corporation (1980).
- 494 [6] J. Fierrez, J. Ortega-Garcia, *On-line signature*, Springer US, 2008, pp.  
495 189–209.



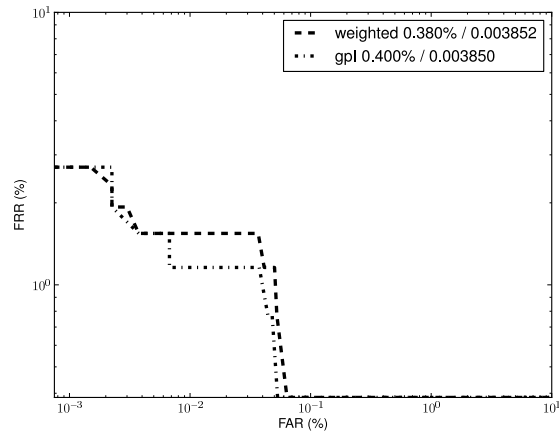
- 496 [7] A. Weiss, A. Ramapanicker, S. Pranav, S. Noble, L. Immohr, Mouse move-  
497 ments biometric identification: A feasibility study, in: Proceedings of Stu-  
498 dent/Faculty Research Day, CSIS, Pace University,, 2007.
- 499 [8] D. Petrovska-Delacretaz, A. El Hannani, G. Chollet, Text-independent  
500 speaker verification: State of the art and challenges, Lecture Notes In Com-  
501 puter Science 4391 (2007) 135.
- 502 [9] C. Nandini, C. Kumar, Comprehensive framework to gait recognition, In-  
503 ternational Journal of Biometrics 1 (1) (2008) 129–137.
- 504 [10] K. Benli, R. Duzagac, M. Eskil, Driver recognition using gaussian mixture  
505 models and decision fusion techniques, in: ISICA 2008, 2008.
- 506 [11] M. Turk, A. Pentland, Face recognition using eigenfaces, in: Proc. IEEE  
507 Conf. on Computer Vision and Pattern Recognition, Vol. 591, 1991.
- 508 [12] D. Maltoni, A. Jain, S. Prabhakar, Handbook of fingerprint recognition,  
509 Springer, 2009.
- 510 [13] A. Kumar, D. Zhang, Personal recognition using hand shape and texture,  
511 IEEE Transactions on Image Processing 15 (8) (2006) 2454.
- 512 [14] Z. Xu, X. Guo, X. Hu, X. Cheng, The blood vessel recognition of ocular  
513 fundus, in: Proceedings of the 4th International Conference on Machine  
514 Learning and Cybernetics (ICMLC'05), 2005, pp. 4493–4498.
- 515 [15] A. Ross, K. Nandakumar, A. Jain, Handbook of multibiometrics, Springer,  
516 2006.
- 517 [16] M. Theofanos, B. Stanton, C. A. Wolfson, Usability & Biometrics: En-  
518 suring Successful Biometric Systems, National Institute of Standards and  
519 Technology (NIST), 2008.
- 520 [17] ISO, Biometric performance testing and reporting, Tech. rep., ISO/IEC  
521 1975-1:2006(E) (2006).

- 522 [18] J. Bhatnagar, A. Kumar, On estimating performance indices for biometric  
523 identification, *Pattern Recognition* 42 (2009) 1803 – 1815.
- 524 [19] R. Raghavendra, B. Dorizzi, A. Rao, G. Hemantha Kumar, Pso versus  
525 adaboost for feature selection in multimodal biometrics, in: *IEEE 3rd In-*  
526 *ternational Conference on Biometrics: Theory, Applications and Systems,*  
527 *BTAS 2009, 2009.*
- 528 [20] A. Rattani, M. Tistarelli, Robust multi-modal and multi-unit feature level  
529 fusion of face and iris biometrics, in: *International Conference on biometrics*  
530 *(ICB2009), 2009.*
- 531 [21] A. Ross, A. Jain, Multimodal biometrics: An overview, in: *Proceedings*  
532 *of 12th European Signal Processing Conference, Citeseer, 2004, pp. 1221–*  
533 *1224.*
- 534 [22] Y. Zuev, S. Ivanov, The voting as a way to increase the decision reliability,  
535 *Journal of the Franklin Institute* 336 (2) (1999) 361–378.
- 536 [23] P. Buysens, M. Revenu, O. Lepetit, Fusion of ir and visible light modali-  
537 ties for face recognition, in: *IEEE International Conference on Biometrics:*  
538 *Theory, Applications and Systems (BTAS 2009), 2009.*
- 539 [24] J. Montalvao Filho, E. Freire, Multimodal biometric fusion—joint typist  
540 (keystroke) and speaker verification, in: *Telecommunications Symposium,*  
541 *2006 International, 2006, pp. 609–614.*
- 542 [25] S. Hocquet, Authentification biométrique adaptative application à la dy-  
543 namique de frappe et à la signature manuscrite, Ph.D. thesis, Université  
544 de Tours (2007).
- 545 [26] L. Allano, La biométrie multimodale : stratégies de fusion de scores et  
546 mesures de dépendance appliquées aux bases de personnes virtuelles, Ph.D.  
547 thesis, Institut National des Télécommunications (2009).

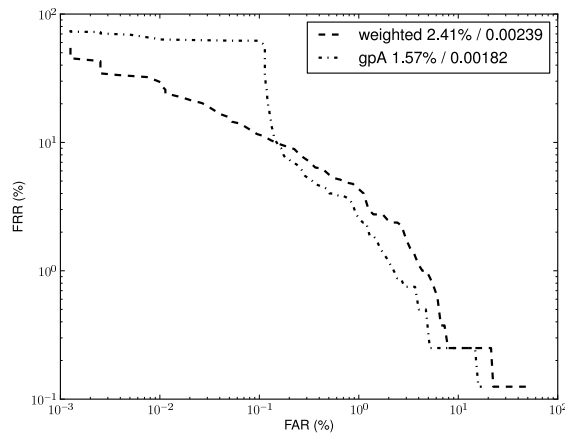
- 548 [27] P. S. Teh, A. B. J. Teoh, C. Tee, T. S. Ong, A multiple layer fusion approach  
549 on keystroke dynamics, *Pattern Analysis & Applications* (2009) 14.
- 550 [28] K. Nandakumar, Y. Chen, S. Dass, A. Jain, Likelihood ratio-based bio-  
551 metric score fusion, *IEEE Transactions on Pattern Analysis and Machine*  
552 *Intelligence* 30 (2) (2008) 342.
- 553 [29] J. Czyz, M. Sadeghi, J. Kittler, L. Vandendorpe, Decision fusion for face  
554 authentication 7.
- 555 [30] S. Garcia-Salicetti, M. Mellakh, L. Allano, B. Dorizzi, Multimodal bio-  
556 metric score fusion: the mean rule vs. support vector classifiers, in: *Proc.*  
557 *EUSIPCO*, 2005.
- 558 [31] J. Koza, J. Rice, *Genetic programming*, Springer, 1992.
- 559 [32] A. Ross, N. Poh, *Handbook of Remote Biometrics*, Springer, Ch. Multibio-  
560 metric Systems: Overview, Case Studies, and Open Issues.
- 561 [33] NIST, Nist biometric score set (2006).  
562 URL <http://www.itl.nist.gov/iad/894.03/biometricscores/>
- 563 [34] N. I. of Standards, Technology, Nist biometric score set (2006).  
564 URL <http://www.itl.nist.gov/iad/894.03/biometricscores/>
- 565 [35] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler,  
566 J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, et al., The  
567 BANCA database and evaluation protocol, *Lecture Notes in Computer*  
568 *Science* (2003) 625–638.
- 569 [36] N. Poh, Banca score database.  
570 URL [http://info.ee.surrey.ac.uk/Personal/Norman.Poh/web/](http://info.ee.surrey.ac.uk/Personal/Norman.Poh/web/banca_multi/main.php?bodyfile=entry_page.html)  
571 [banca\\_multi/main.php?bodyfile=entry\\_page.html](http://info.ee.surrey.ac.uk/Personal/Norman.Poh/web/banca_multi/main.php?bodyfile=entry_page.html)
- 572 [37] N. Sedgwick, C. Limited, Preliminary Report on Development and Evalua-  
573 tion of Multi-Biometric Fusion using the NIST BSSR1 517-Subject Dataset,  
574 Cambridge Algorithmica Limited.

- 575 [38] A. Martinez, R. Benavente, The ar face database, Tech. rep., CVC Techni-  
576 cal report (1998).
- 577 [39] R. Giot, M. El-Abed, R. Christophe, Greyc keystroke: a benchmark for  
578 keystroke dynamics biometric systems, in: IEEE International Conference  
579 on Biometrics: Theory, Applications and Systems (BTAS 2009), 2009.
- 580 [40] D. Lowe, Distinctive image features from scale-invariant keypoints, Inter-  
581 national journal of computer vision 60 (2) (2004) 91–110.
- 582 [41] C. Rosenberger, L. Brun, Similarity-based matching for face authentication,  
583 in: Proceedings of the International Conference on Pattern Recognition  
584 (ICPR'2008), Tampa, Florida, USA, 2008.
- 585 [42] R. Giot, M. El-Abed, C. Rosenberger, Keystroke dynamics with low con-  
586 straints svm based passphrase enrollment, in: IEEE Third International  
587 Conference on Biometrics : Theory, Applications and Systems (BTAS),  
588 2009.
- 589 [43] F. Hampel, E. Ronchetti, P. Rousseeuw, W. Stahel, Robust statistics: the  
590 approach based on influence functions, John Wiley & Sons New York, 1986.
- 591 [44] A. Jain, K. Nandakumar, A. Ross, Score normalization in multimodal  
592 biometric systems, Pattern Recognition 38 (12) (2005) 2270 – 2285.  
593 URL [http://www.sciencedirect.com/science/article/  
594 B6V14-4G0DDW4-1/2/d922960ee7ed8928744113dd9494d37a](http://www.sciencedirect.com/science/article/B6V14-4G0DDW4-1/2/d922960ee7ed8928744113dd9494d37a)
- 595 [45] M. Mitchell, An introduction to genetic algorithms, The MIT press, 1998.
- 596 [46] R. Poli, W. Langdon, N. McPhee, A field guide to genetic programming,  
597 Lulu Enterprises Uk Ltd, 2008, freely available at [http://www.gp-filed-  
598 guide.org.uk](http://www.gp-filed-guide.org.uk).
- 599 [47] P. Day, A. K. Nandi, Robust text-independent speaker verification using ge-  
600 netic programming, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND  
601 LANGUAGE PROCESSING 15 (2007) 285–295.

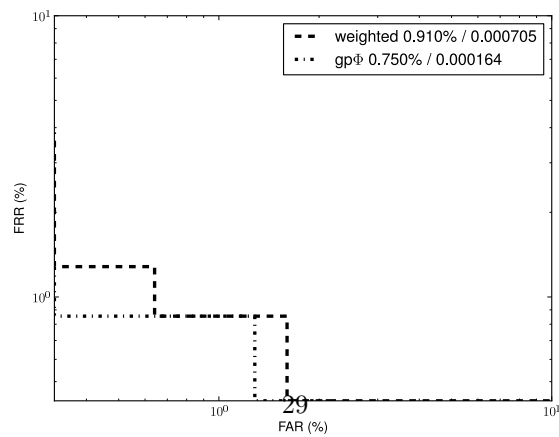
- 602 [48] M. Khoury, Python strongly typed genetic programming.  
603 URL <http://pystep.sourceforge.net>
- 604 [49] T. Oliphant, Guide to NumPy, Spanish Fork, UT, Trelgol Publishing.
- 605 [50] R. Giot, M. El-Abed, C. Rosenberger, Fast learning for multibiometrics sys-  
606 tems using genetic algorithms, in: The International Conference on High  
607 Performance Computing & Simulation (HPCS 2010), IEEE Computer So-  
608 ciety, Caen, France, 2010, p. 8.
- 609 [51] R. Giot, B. Hemery, C. Rosenberger, Low cost and usable multimodal bio-  
610 metric system based on keystroke dynamics and 2d face recognition, in:  
611 IAPR International Conference on Pattern Recognition (ICPR), IAPR, Is-  
612 tanbul, Turkey, 2010.
- 613 [52] C. Chang, C. Lin, LIBSVM: a library for support vector machines (2001).



(a) Validation with BSSR1



(b) Validation with PRIVATE



(c) Validation with BANCA

Figure 4: ROC curves of the fusion systems from the state of the art and with genetic programming. The EER of each fusion function is presented in the legend. Note the use of a logarithmic scale.

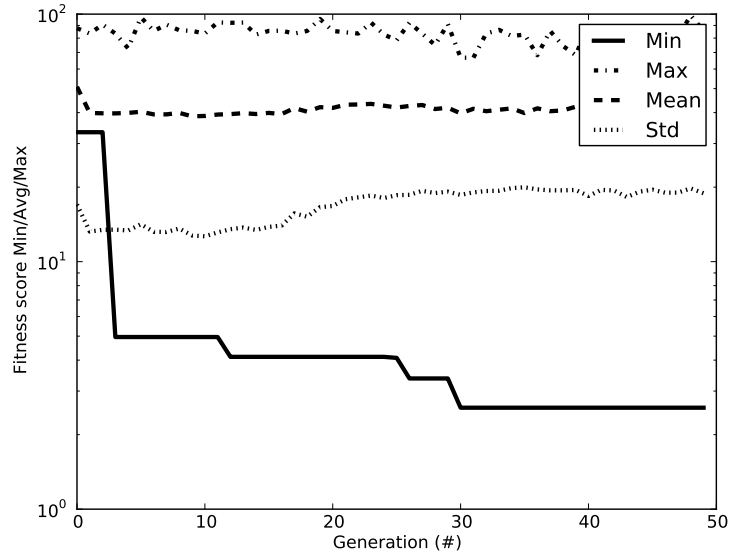


Figure 5: Fitness evolution of one run of the genetic programming evolution. The max, min, mean and std values of the fitness are represented. We want to minimize the fitness value, so lower is better.

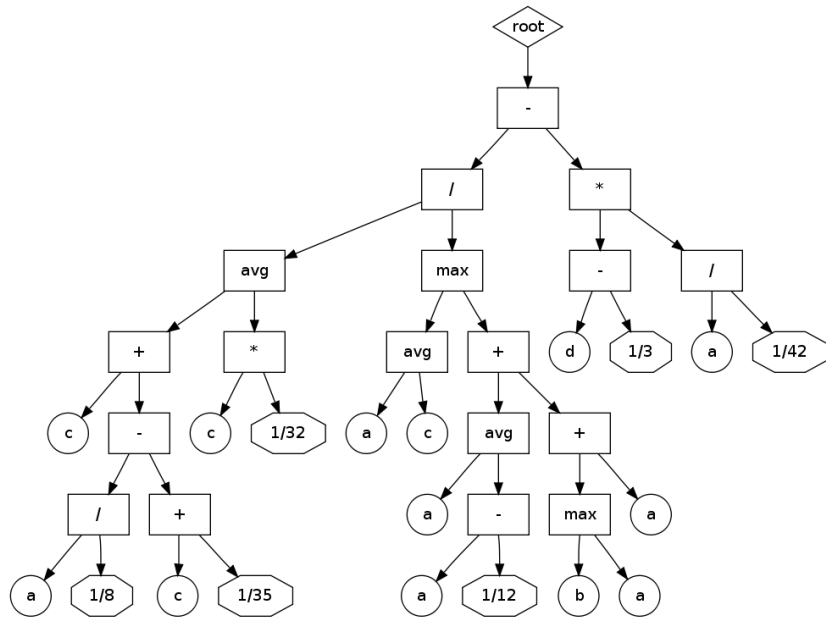


Figure 6: Sample of a "simple" generated program. We can observe the complexity of the generated fusion function.