
Algorithm 1 Marie Curie Genetic Algorithm Overview

- 1: Let $NUM_GENS = 100$ represent the number of generations to evolve
 - 2: Let $POP_SIZE = 100$ represent the size of the total population
 - 3: Let $NBPG = 50$ represent the number of new candidates to breed every generation
 - 4: Let S_{min} and S_{max} represent the vectors of servo limits
 - 5: Let L_R represent the landmarks in the reference image
 - 6: Let $P = \{C_i, i \in \{0, 1, \dots, POP_SIZE\}\}$ where each C_i is a candidate expression
 - 7: **for** $i \in \{0, 1, \dots, POP_SIZE - 1\}$ **do**
 - 8: Initialize $\mathbf{C}_i.\mathbf{chromosome}$ to $\mathcal{U}(S_{min}, S_{max})$
 - 9: Score each candidate s.t. $C_i.score = mc_score()$
 - 10: **end for**
 - 11: **for** $g \in \{0, 1, \dots, NUM_GENS\}$ **do**
 - 12: Let $N = \{N_i, i \in \{0, 1, \dots, NBPG\}\} = mc_breed_new(NBPG)$ be the set of new candidates for this gen
 - 13: Score N s.t. $N_i.score = mc_score(\mathbf{N}_i.\mathbf{chromosome}, L_R)$
 - 14: Set $C = C \cup N$
 - 15: Remove the $NBPG$ elements with the lowest score from C
 - 16: **end for**
 - 17: **return** $C_{opt} = \min(C)$
-

Algorithm 2 MC Score Algorithm ($\text{mc_score}(X, L'_R)$)

- 1: Let $M = 68$ be the number of landmarks recognized in an image (constrained by dlib landmark recognition algorithm)
 - 2: Let $D = 2$ be the dimensionality of a single landmark
 - 3: Let $L \in \mathbb{R}^{D,M}$ represent the raw pixel valued landmarks
 - 4: Let L'_R be the reference landmarks to score against
 - 5: Let X be the chromosome to score
 - 6: Actuate X on the face using the Maestro Controller
 - 7: Capture an image, I_C , of the face
 - 8: Get the bounding box B of the face
 - 9: Detect the center of a rectangular bounding box, $B_C = [x, y]^T \in \mathbb{R}^2$ of the face
 - 10: Detect raw landmarks $L_C = \text{dlib_predictor}(I_C).landmarks$
 - 11: Calculate new "centered" landmarks $L_C^C = L_C - (B_C \otimes \mathbf{1}^T)$
 - 12: Calculate the horizontal scaling by $x_s = B_{Right} - B_{C,x}$
 - 13: Calculate the vertical scaling by $y_s = B_{Bottom} - B_{C,y}$
 - 14: Get normalized landmarks $L'_C = L_C^C \oslash ([x_s, y_s]^T \otimes \mathbf{1}^T)$
 - 15: Calculate difference in corresponding landmarks $L_D = L'_R - L'_C$
 - 16: **return** $\|L_D\|_F = \sqrt{\text{tr}(L_D^T L_D)}$ where $L_D^T L_D \in \mathbb{R}^{(M,M)}$ is the Gramian matrix of L_D and the diagonal elements of $L_D^T L_D$ represent the euclidean distance error of each landmark
-

Algorithm 3 MC Breed New ($\text{mc_breed_new}(N_C, N_P)$)

- 1: Let N_C be the number of new candidates to breed
 - 2: Let N_{CP} be the maximum number of candidates to breed for a single set of parents
 - 3: Calculate the number of pairs of parents N_{pP}
 - 4: Get the mating pool M which is comprised of the best (lowest) scoring $2N_{pP}$ candidates C_i
 - 5: Create N_{pP} pairs of parents, P_p by pairing the highest scoring parents together (i.e. $P_p = \{\{M_0, M_1\}, \{M_2, M_3\}, \dots\}$)
 - 6: **for** $p \in P_p$ **do**
 - 7: **for** $j \in \{0, 1, \dots, N_{CP}\}$ **do**
 - 8: Let C_N represent the new candidate
 - 9: Randomize order of p
 - 10: Randomly select location i in which to perform crossover of chromosomes
 - 11: Set $\mathbf{C}_N.\text{chromosome} = [p_0[:i]^T, p_1[i:]^T]^T$ representing the concatenation of the two parent chromosomes sliced at index i
 - 12: Set $\mathbf{C}_N.\text{chromosome} = \mathcal{N}(\mathbf{C}_N.\text{chromosome}, 50^2)$ representing a mutation of the inherited chromosome with each gene varying in a normal distribution parameterized by $\mu = \mathbf{C}_N.\text{chromosome}$ and $\sigma^2 = 50^2$ where 50 is the standard deviation in quarter microseconds
 - 13: Saturate values in $\mathbf{C}_N.\text{chromosome}$ to safe limits for servos
 - 14: **end for**
 - 15: **end for**
-