Automatic Smoothing for Discontinuous Regression Functions: Supporting Document

This supporting document provides additional material for the manuscript "Automatic Smoothing for Discontinuous Regression Functions". It contains two sections. Section A provides additional plots for Section 6 (Simulation Results) of the manuscript, while Section B lists the major steps of the genetic algorithms used in the manuscript.

A Graphical Outputs

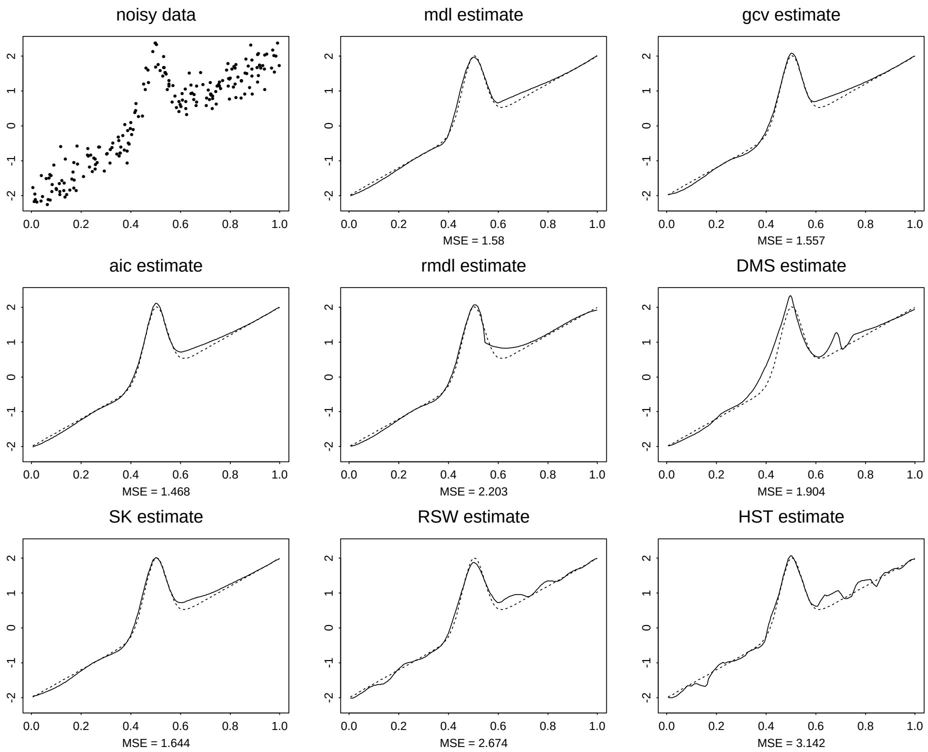

Figure 1: Test Function 1. Solid lines: estimates; broken lines: true curves.

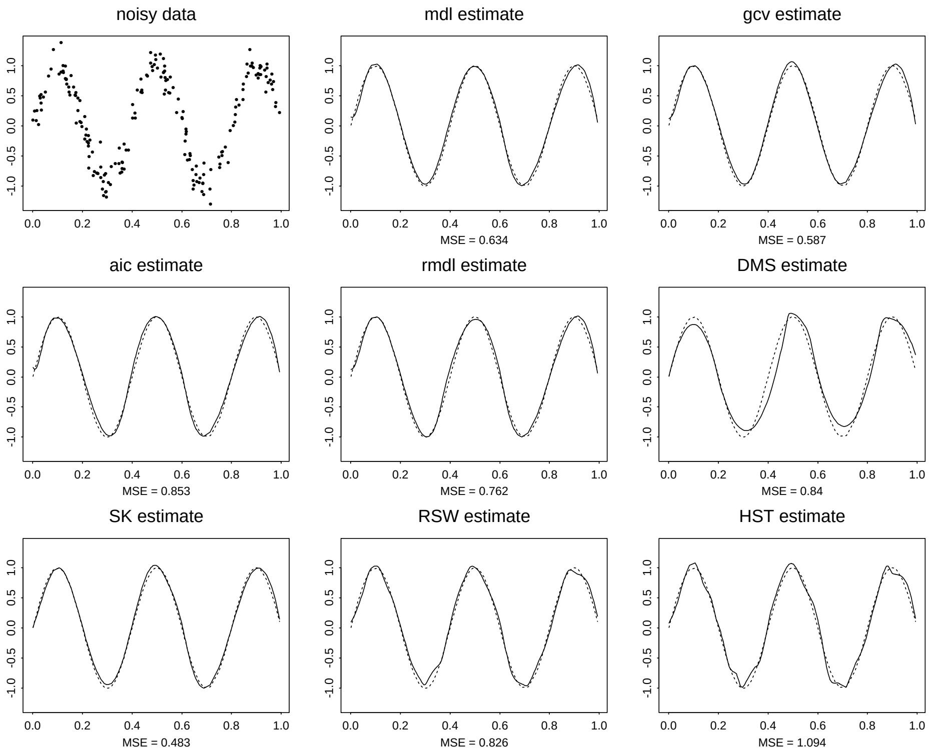

Figure 2: Test Function 3. Solid lines: estimates; broken lines: true curves.

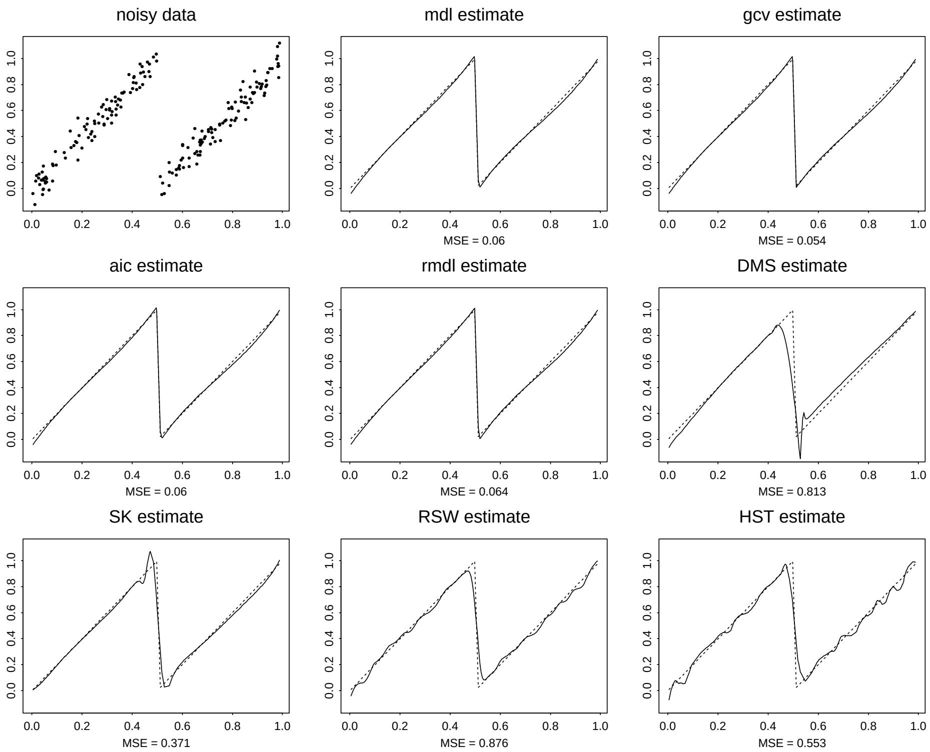

Figure 3: Test Function 4. Solid lines: estimates; broken lines: true curves.

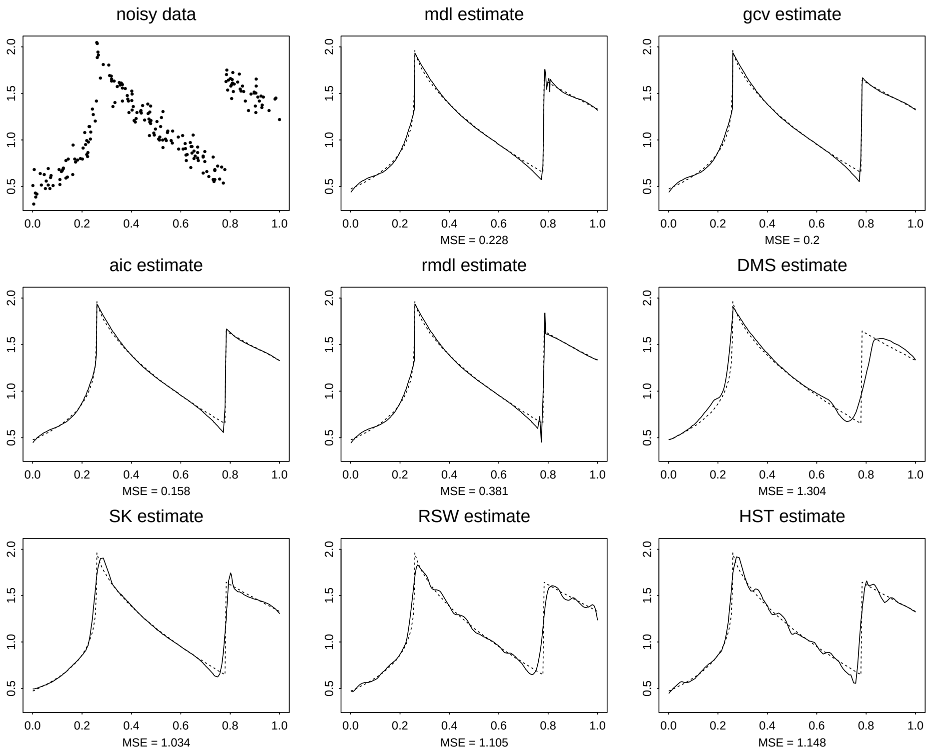

Figure 4: Test Function 6. Solid lines: estimates; broken lines: true curves.

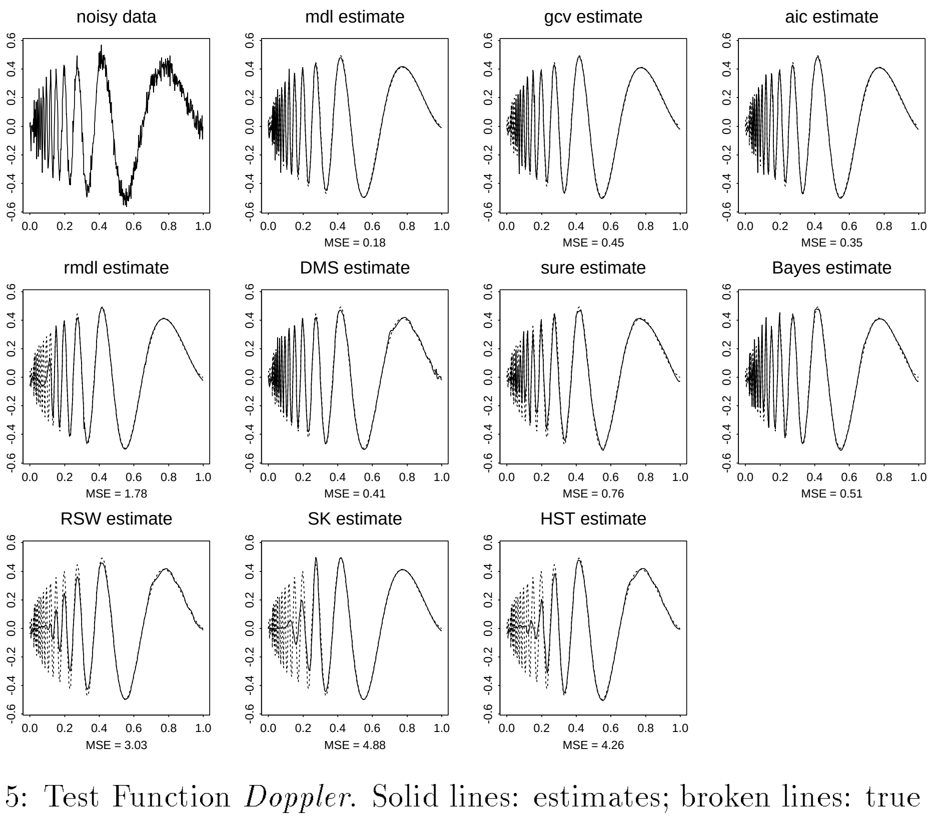

Figure 5: Test Function *Doppler*. Solid lines: estimates; broken lines: true curves.

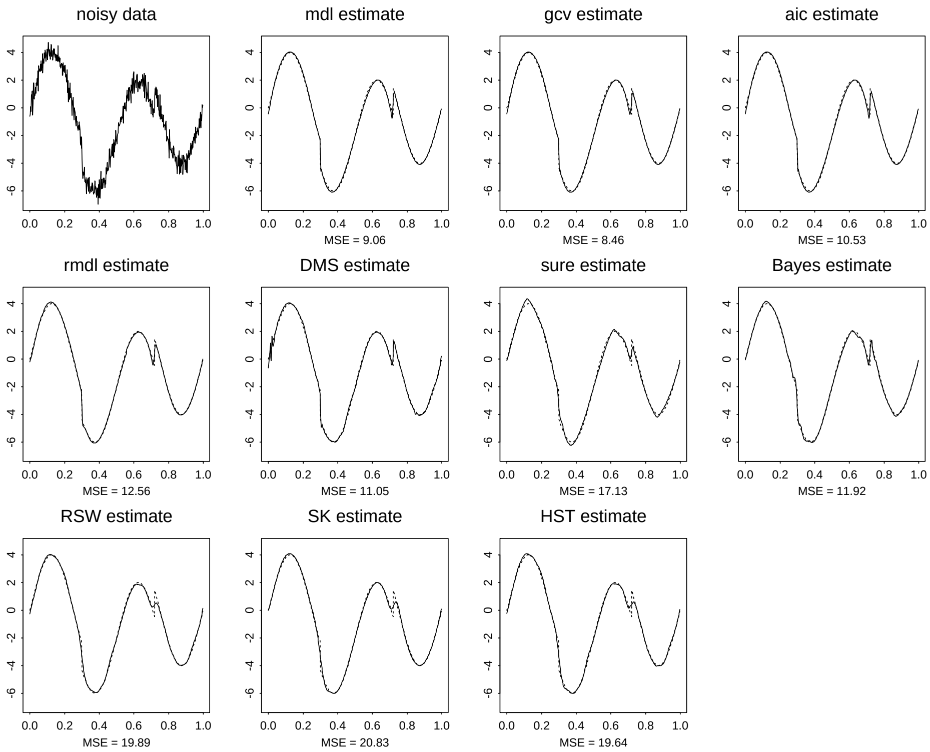

Figure 6: Test Function *Heavisine*. Solid lines: estimates; broken lines: true curves.

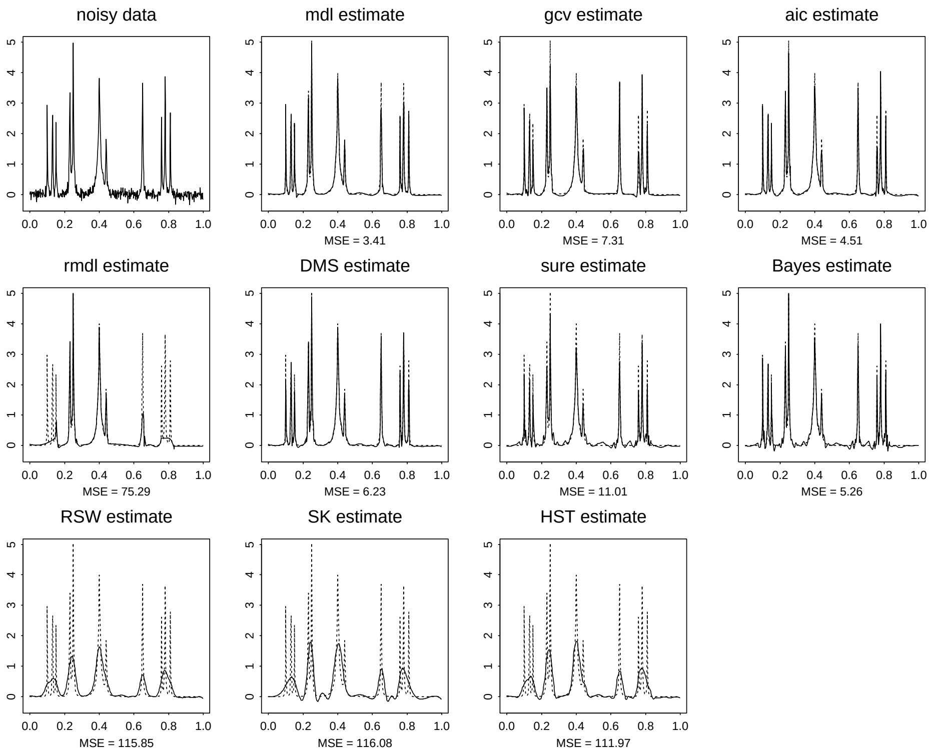

Figure 7: Test Function *Bumps*. Solid lines: estimates; broken lines: true curves.

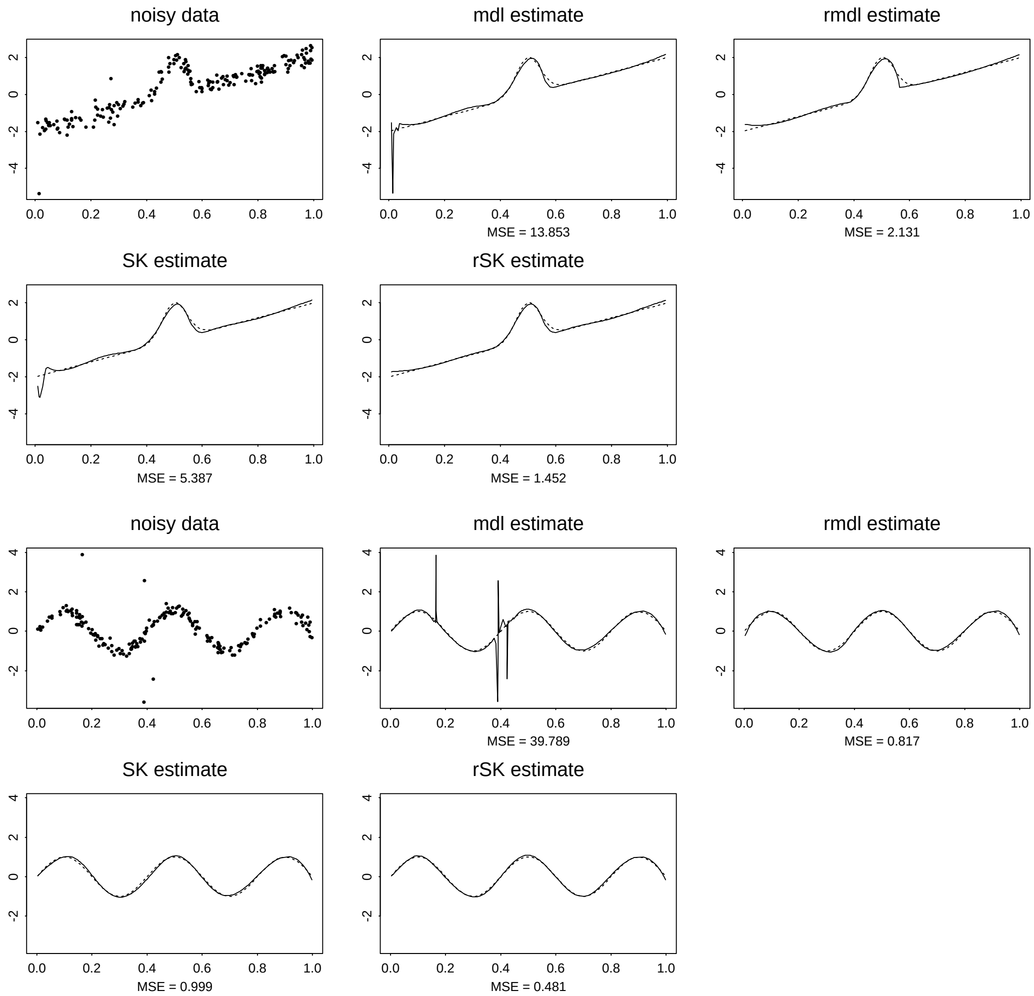

Figure 8: Various estimates of Test Functions 1 (top two rows) and 3 (bottom two rows) when data were contaminated by outliers. Solid lines: estimates; broken lines: true curves.

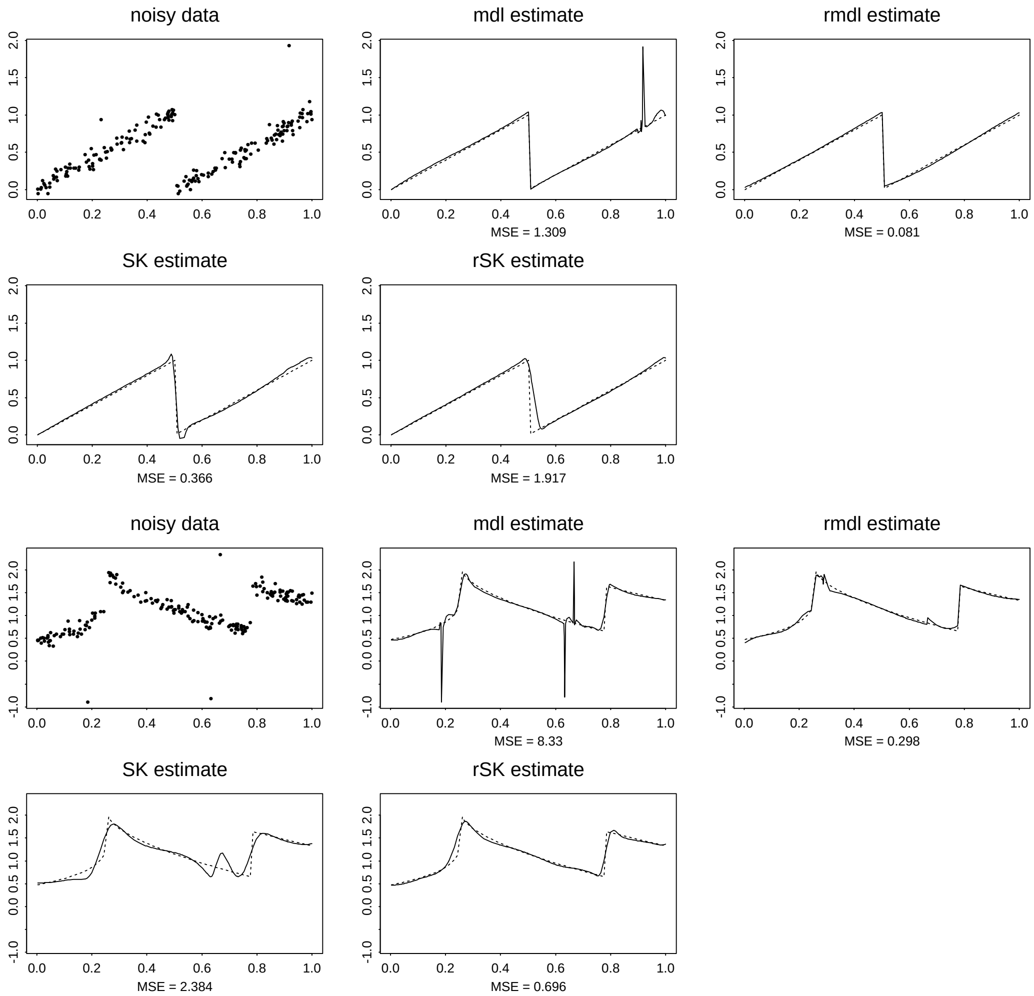

Figure 9: Various estimates of Test Functions 4 (top two rows) and 6 (bottom two rows) when data were contaminated by outliers. Solid lines: estimates; broken lines: true curves.

B Implementation of Genetic Algorithms

This section lists the major steps in the implementation of the genetic algorithm that was used to minimize $\mathrm{RMDL}(\hat{f})$ in the numerical experiments. Genetic algorithms for minimizing $\mathrm{MDL}(\hat{f})$, $\mathrm{GCV}(\hat{f})$ and $\mathrm{AIC}(\hat{f})$ are similar and hence omitted. The steps are as follows.

1. Randomly generate an initial population of chromosomes of size S.

2. Compute the $\mathrm{RMDL}(\hat{f})$ value for each of the S chromosomes.

3. Sort the $\mathrm{RMDL}(\hat{f})$ values in descending order. Assign rank 1 to the chromosome with the largest $\mathrm{RMDL}(\hat{f})$ value, rank 2 to the one with the second largest $\mathrm{RMDL}(\hat{f})$ value, and so on. Denote the rank of chromosome i as r_i.

4. With probability P_{CROSS}, perform a crossover operation. Otherwise, perform a mutation operation.

   - *Crossover* — one child is produced from two parents: select two parent chromosomes from the initial population. The probability that chromosome i will be selected is $r_i/(r_1 + \ldots + r_S)$. Then for each gene location of the child chromosome, with equal probability, assign it either the corresponding father gene or the corresponding mother gene.

   - *Mutation* — one child is produced from one parent: select one parent using the same probability law as in crossover. For each child gene location, assign it either a knot gene with probability P_{K}, a discontinuity gene with probability P_{D}, an outlier gene with probability P_{O}, a normal gene with probability P_{N}, or the corresponding father gene with probability $1 - P_{\text{K}} - P_{\text{D}} - P_{\text{O}} - P_{\text{N}}$.

5. Repeat Step 4 until S child chromosomes are produced. That is, until a whole new generation is obtained.

6. Repeat Steps 2 to 5 using the new generation as the initial population. Then repeat the whole process until the RMDL($\hat{f}$) value of the best chromosome does not change for N_{SAME} generations.

7. The best chromosome of the youngest generation is taken as the minimizer of RMDL($\hat{f}$).

In our implementation $S = 300$, $P_{\text{CROSS}} = 0.9$, $P_{\text{K}} = 0.1$, $P_{\text{D}} = 0.02$, $P_{\text{O}} = 0.3$, $P_{\text{N}} = 0.1$ and $N_{\text{SAME}} = 20$.