Formal definition of weighted random mutant selection algorithm

This document describes the algorithms used to compute the correlation and perform the random mutant selection. For the sake of clarity, we first introduce the definitions used in these algorithms:

- C : Set of all classes
- M_i : Set of mutants generated for class $c_i \in C$
- $M = \bigcup\limits_{c_i \in C} M_i$ (Set of all mutants)
- $M_C = \bigcup\limits_{c_i \in C} \{M_i\}$ (A partition of M based on set of classes C)
- $(M^{(N)}$: Randomly selected sample of mutants, $(M^{(N)} \subseteq M)$, $(\left|M^{(N)}\right| = N)$
- $M_C^{(N)} = \bigcup\limits_{c_i \in C} \left\{M_i \cap M^{(N)}\right\}$ (A partition of $M^{(N)}$ based on set of classes C)
- $MC : M' \to [0, 100]$, $MC(M') = 100 \times |\forall m \in M' : \ m \ is \ killed| \div |M'|$ (Mutation coverage for set of mutants M')
- $WF : M' \to W'$, M': $WF(m) = w$ (Weight function from M' to W')

To make it more clear, we report an example in Figure 1. In this figure, M_1, M_2, and M_3 are the generated mutants for three classes, $C = \{c_1, c_2, c_3\}$. So, M, the set of all mutants, is the union of these sets, and M_C is the partition of M according to C. $M^{(N)}$ is a random sample from M. We can see that the members of $M_C^{(N)}$ can be determined using M_1, M_2, M_3, and $M^{(N)}$.

We populate $M^{(N)}$ by selecting N mutants randomly from M. Then, $M_C^{(N)}$ is a partition of $M^{(N)}$ based on set of classes C: $M_C^{(N)} = \bigcup\limits_{c_i \in C} \left\{M_i \cap M^{(N)}\right\}$. Mutation coverage for class c_i using all mutants (M) is $MC(M_i)$, and using the sampled set of mutants $(M^{(N)})$ is $MC(M_i \cap M^{(N)})$.

Correlation Calculation Algorithm. To compute the correlation we use the algorithm in Figure 2. In this algorithm, $S_{SM}(M^{(N)})$ is the set of mutation coverage values for all classes calculated using the sampled set of mutants: $S_{SM}(M^{(N)}) = \left\{MC(M_i \cap M^{(N)}) : \forall c_i \in C\right\}$. To calculate mutation coverage for all classes using all mutants, we use the formula: $S_{AM} = S_{SM}(M)$. We calculate the correlation using sampling rates from 1% to 100% to find out the

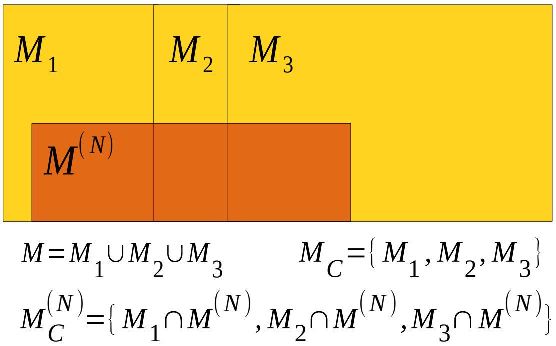

Figure 1: Example of partitioning of M and $M^{(N)}$

acceptable sampling rate. In order to minimize random noise, we repeat the random selection process 10 times and use the average of the calculated correlations between $S_{SM}(M^{(N)})$ and S_{AM} for all iterations.

The Algorithm For The Uniform And The Weighted Approaches. To select our mutants $M^{(N)}$ we use the algorithm in Figure 3. For the uniform approach, we set $W = \{1\}$ and $\forall m \in M : WF(m) = 1$. For the weighted approach, the weights are assigned according to the inverse of the size of M_i for class $c_i \in C$. Consider $MaxLength$ as $\forall M_i \in M_C : \max(|M_i|)$. Then WF is defined as:

- $WF : M \rightarrow \{\forall M_i \in M_C : |M_i|\}$
  $\forall M_i \in M_C \ \wedge \ \forall m \in M_i :$
  $WF(m) = 101 - (100 \times |M_i| \div MaxLength)$

function CalculateMutationCoverage(M_C, $M^{(N)}$)
  for all $M_i \in M_C$ **do**
    $S_{SM} = S_{SM} \cup \left\{MC(M_i \cap M^{(N)})\right\}$
  return S_{SM}
procedure CalculateCorrelationValues(M_C, M, W, WF, MC)
  $S_{AM} \leftarrow$ CalculateMutationCoverage(M_C, M)
  $SamplingRate \leftarrow 0$
  repeat
    $SamplingRate \leftarrow SamplingRate + 1$
    $N \leftarrow |M| \times SamplingRate \div 100$
    $Iteration \leftarrow 1$
    repeat
      $M^{(N)} \leftarrow$ RandomlyPopulate(M, W, WF, N)
      $S_{SM}(M^{(N)}) \leftarrow$ CalculateMutationCoverage(M_C, $M^{(N)}$)
      $PearsonSum \leftarrow PearsonSum + \rho(S_{SM}(M^{(N)}), S_{AM})$
      $KendallSum \leftarrow KendallSum + \tau_b(S_{SM}(M^{(N)}), S_{AM})$
      $Iteration \leftarrow Iteration + 1$
    until $Iteration = 10$
    $KendallAverage[SamplingRate] \leftarrow KendallSum \div 100$
    $PearsonAverage[SamplingRate] \leftarrow PearsonSum \div 100$
  until $SamplingRate = 100$
  return $KendallAverage[\,]$, $PearsonAverage[\,]$

Figure 2: Correlation calculation algorithm

function WeightedRandomSelection(M, W, WF)
  $SumOfWeights \leftarrow 0$
  for $w \in W$ **do**
    $SumOfWeights \leftarrow SumOfWeights + w$
  $r \leftarrow Random(0, w)$
  $w' \leftarrow 0$
  for $m \in M$ **do**
    $w' \leftarrow w' + WF(m)$
    if $w' >= r$ **then**
      return m
procedure RandomlyPopulate(M, W, WF, N)
  $M^{(N)} \leftarrow \emptyset$
  repeat
    $m \leftarrow$ WeightedRandomSelection(M, W, WF)
    $M^{(N)} \leftarrow M^{(N)} \cup \{m\}$
  until $\left|M^{(N)}\right| = N$

Figure 3: Weighted random selection algorithm