Definitions

1. Genes: Basic elements represented as unique strings or symbols.

2. Encodings: A mapping mechanism that associates genes with unique hash keys generated through a hashing function.

3. Captured Segments: Subsets of genes that are grouped together and treated as a single entity in the encoding and decoding process.

4. Start and End Delimiters: Special genes used to mark the boundaries of captured segments within the encoding structure.

Mathematical Formalization

Encodings

Let $G = \{ g_1, g_2, ..., g_n \}$ be a set of genes. Each gene g is associated with a unique hash key k through a function h:

$h: G \rightarrow K$

where K is the set of all possible hash keys.

The hash key generation uses the xxhash function, ensuring a unique and consistent mapping:

$k = xxhash(g)$

Start and End Delimiters

Two special genes, "Start" and "End", are used to demarcate the boundaries of captured segments. These are predefined and mapped to unique identifiers:

$h("Start") = k_s$

$h("End") = k_e$

Captured Segments

A captured segment S is defined as a tuple of genes:

$S = (g_{i1}, g_{i2}, ..., g_{im})$

Each segment S is also associated with a unique hash key k_s obtained similarly through the hashing function. This key k_s serves as a reference in the encodings map.

$h(S) = k_s$

Encoding and Decoding Functions

Encoding function E maps a sequence of genes to a sequence of hash keys:

$E(g_1, g_2, ..., g_n) = (h(g_1), h(g_2), ..., h(g_n))$

Decoding function D reverses the process, mapping a sequence of hash keys back to the corresponding gene sequence:

$D(k_1, k_2, ..., k_n) = (h^{-1}(k_1), h^{-1}(k_2), ..., h^{-1}(k_n))$

Integration and Segmentation

During integration of external encoding data or when modifying the gene pool, the system incorporates and validates new genes and captured segments against existing keys. If new, they are added; if duplicates or conflicts arise, the system manages these based on predefined logic.

Operations

- Add Gene: Addition of a new gene involves generating a hash key and updating the mappings.

- Capture Segment: Grouping a series of genes into a segment and assigning it a unique identifier.

- Open Segment: Decompressing a captured segment, potentially including the application of start and end delimiters unless specified not to do so.

Usage in Genetic Algorithms

In the context of a genetic algorithm, these encodings facilitate complex manipulations of genetic material, allowing for advanced operations such as crossover and mutation at a more abstract level, where entire segments of genetic information can be operated upon as units, enhancing feature extraction and transfer learning capabilities through this mutable encoding system.

Definitions and Initialization

- Let P represent the population of organisms, where each organism is a sequence of genes encoded as hash keys.

- Let f denote the fitness function, f: $G^n \rightarrow \mathbb{R}$, mapping a sequence of genes (an organism) to a real number indicating its fitness.

Genetic Algorithm Processes

Selection

Elitism:

$P\_elite = select\_top(P, elitism\_ratio)$

where select_top selects the top-performing individuals based on fitness, determined by the parameter elitism_ratio.

Parent Selection:

$P\_parents = tournament\_select(P, num\_parents)$

where tournament_select picks parents based on a tournament selection process.

Crossover

Single-Point Crossover:

$(O_1, O_2) = crossover(P_1, P_2)$

where P_1 and P_2 are parent organisms and O_1, O_2 are the offspring produced. The crossover point is randomly selected, except within delimited segments unless explicitly allowed.

Mutation

Mutation Types include insertion, deletion, point mutation, and special mutations within delimited segments (capture and open mutations).

$O' = mutate(O, mutation\_prob, mutation\_types)$

where O' is the mutated organism and mutation_types are context-dependent based on the presence of special markers like start and end delimiters.

Fitness Evaluation

The fitness of each organism in the population is evaluated using:

$\text{fitness\_scores} = f(P)$

where f is applied to each organism in the population.

Algorithm Execution Flow

Initialization: Set up an initial population of encoded organisms.

Generational Loop:

Evaluate fitness of the current population.

Select the elite to carry forward to the next generation.

Select parents and generate offspring via crossover and mutation.

Evaluate new population's fitness and repeat until the termination condition (e.g., max generations) is met.

Mathematical Formalization of Mutable Encoding Operations

Encode and Decode

$E(g_1, g_2, \ldots, g_n) = (h(g_1), h(g_2), \ldots, h(g_n))$

$D(k_1, k_2, \ldots, k_n) = (h^{-1}(k_1), h^{-1}(k_2), \ldots, h^{-1}(k_n))$

where h is the hashing function mapping genes to their hash keys and h^{-1} is the inverse mapping hash keys back to genes.

Capture Segment

- Segments of genes are captured and encoded as a single unit, which can later be decoded or

manipulated as a group.

Delimited Operations

- Operations within segments marked by start and end delimiters are handled specially, such as not allowing crossover points within these segments unless specified.

This mathematical formalization captures the essential functionality of the M_E_GA_Base class, focusing on the integration of mutable encoding within a genetic algorithm framework, optimizing genetic operations based on encoded representations.

Mutations

Definitions

Base Genes (G): Fundamental units of genetic material, each represented by a unique string or symbol, excluding special delimiters.

Captured Codons (C): Encoded segments of base genes that have been previously delimited and are treated as individual units during the genetic algorithm process.

Organism (O): A sequence of genes, which may include base genes, captured codons, and delimiters, representing an individual in the population.

Delimited Region (D): A section within an organism explicitly marked by start and end delimiters, enclosing one or more genes or codons. This region can be treated as a segment for specialized mutations or operations.

Mathematical Formalization

Mutation Probabilities and Selection Function

Define mutation probabilities and a selection function distinguishing between base genes, captured codons, and handling within delimited regions.

Mutation Probabilities

p_mutation: Probability of any mutation occurring.

p_insertion: Probability of an insertion mutation.

p_deletion: Probability of a deletion mutation.

p_point: Probability of a point mutation.

p_swap: Probability of a swap mutation.

p_capture: Probability of capturing a new segment within a delimited region.

p_open: Probability of opening a captured segment.

Selection Function

The selection function S(O) chooses between selecting a base gene or a captured codon based on their available proportions and defined probabilities:

$S(O) =$

{

g_i if $X < p_{base}$, $g_i \in G$

c_j if $X \geq p_{base}$, $c_j \in C$

}

Where

X is a random variable uniformly distributed over [0, 1].

p_base is the probability of choosing a base gene over a captured codon.

Mutation Operations:

Mutations are applied based on the context (delimited or undelimited regions) and the type of gene or codon:

Insertion

$O' = O[1:i-1] + S(O) + O[i:len(O)]$, if $X < p\_insertion$

Deletion

$O' = O[1:i-1] + O[i+1:len(O)]$, if $X < p\_deletion$

Point Mutation

$O' = O[1:i-1] + S(O) + O[i+1:len(O)]$, if $X < p\_point$

Swap (within bounds and not involving delimiters)

$O'[i] = O[j]$, $O'[j] = O[i]$, if $X < p\_swap$

Capture (only within delimited regions)

$O' = O[1:s-1] + capture(O[s:e]) + O[e+1:len(O)]$, if $X < p\_capture$

Open (only applies to captured codons)

$O' = O[1:i-1] + open(c\_j) + O[i+1:len(O)]$, if $X < p\_open$, $c\_j \in C$

These mathematical formulations capture the essential functionality of mutation selection and application, emphasizing the differential treatment of base genes and captured codons within a genetic algorithm context, ensuring fidelity to the encoded genetic structure while promoting genetic diversity and adaptability.