

The best genetic Algorithm I

Kuri selected a set of GA's (genetic algorithms), which are structurally different from each other to compare and establish the efficiency

Variations

Different selection criteria

Cross over strategies

population size

Notation

A): set of selected optimization algorithms.

A_i): the i -th optimization algorithm (i.e. $A_i \in A$).

X): vector in R^n

F): feasibility region of the space R^n

B) Set defined as $B = \{0, 1\}$

E): iterations number such that

G): Upper bound on the number of iterations of A

R_n): Objective function [fitness function).

it's been argued that is "genetic" when it exhibits implicit parallelism but he listed a set of characteristics that an iterative algorithm must have to be considered

"genetic" which it's implicit parallelism as a consequence.

A genetic algorithm is one which satisfies the following conditions:

1. it works on an n -dimensional discrete space D defined in N_r rather than in R_n
2. it traverses D searching an approximation of the optimum vector x of CD by simultaneously analyzing a finite set $S(t) \in D$ of candidate solutions.
3. The elements of $S(t) = \{S1t, S2t, \dots, Snt\}$ are explicitly encoded in some suitable way
4. The information regarding the partial adequacy of the elements in $S(t)$ is extracted by solving the optimization problem for all $Si(t)$.
5. The qualified elements of $Sa)$ are analyzed to select an appropriate subset in order to improve the search in the problem's space
6. Selected sections of the codes of $Si(t)$ are periodically combined.
7. Selected elements of the codes of the $Si(t)$ are periodically and randomly combined.
8. A subset of the best solutions of $S(t)$ is preserved for all the future steps of the algorithms
9. The algorithm cycles through steps 4-8 until a stopping condition is met.

these are the following GA's that he compares:

- A. An elitist canonical GA-TGA

- B. A cross generational elitist selection, Heterogeneous recombination and cataclysmic mutation algorithm-CHC algorithm
- C. An eclectic Genetic Algorithm-EGA
- D. A statistical GA-SGA

The nature of code
(Daniel videos)

traditional Genetic Algorithm:

1. Create a population N elements with random genetic material (set up function)
2. A) Calculate fitness for N elements
3. Reproduction / selection
4. Pick 2 "parents"
5. Make a new element
6. Crossover
7. Mutation

Population {Array of elements}

DNA {Array of characters}

2 key places for our own creativity to make different from the code for our project

1. fitness function \leftarrow our own
2. how do you encode our DNA?
3. genotype vs. phenotype
"Data" "expression"