Mono-Objective Genetic Algorithm in Python

October 3, 2017

Reference Manual

Contents

1 Introduction

This is an implementation of the Genetic Algorithm in Python.

The added features for faster resolution times are as follows:

1. Handling variable types differently (continuous, binary, discrete)
2. Conversion of variables into binary equivalent
3. User defined choice of selection mechanism
4. Parallel computing

1.1 Basic genetic algorithm structure

The basic structure of the genetic algorithm is as follows:

1. Initialize a population of agents
2. Evaluate the fitness of the agents in the population
3. For the predefined number of generations:
   (a) Select parents for the creation of offsprings
   (b) Use the parents to create offsprings
   (c) Apply mutation mechanism for diversity
   (d) Evaluate the fitness of this new population
4. Extract the agents with best fitness

2 Installation

At the moment, this code only works in Windows®. The folder has to be copied into the directory 'C:\' as some internal commands are hard-coded.

2.1 Required python packages

1. pandas
2. numpy
3. datetime
4. matplotlib
5. math
6. copy
7. random
8. multiprocessing

2.2 Using the package

Important files and locations to take note of:

1. Input data: `ga_mono_simplesetup.py`
   (a) Population size (e.g. population = 200).
   (b) Number of generations to run the algorithm for (e.g. generations = 1000)
   (c) Selection mechanism to determine the agents populating the parent pool
      i. Available options: 1. `roulette_wheel` and 2.`tournament_selection`
   (d) Determine the size of the parent pool
      i. The variable `crossover_perc` is used to determine the size of the parent pool
      ii. e.g. `crossover_perc` = 0.5, means that the parent pool will be 50% of the population size

3 Modules

4 References