

Mejorando The Whale Optimization Algorithm Metaheurísticas

Ignacio Aguilera Martos

DNI: 77448262V e-mail: nacheteam@correo.ugr.es

Curso 2017-2018

Índice

1. Introducción del problema	2
1.1. Competición CEC2014	2
1.2. Módulo de funciones CEC2014	2
2. Descripción del algoritmo inicial	2
3. Desarrollo de mejoras	2
4. Versión final	2
5. Resultados	2

1. Introducción del problema

Durante este trabajo he tenido como objetivo desarrollar un algoritmo mejorado a partir de un algoritmo bioinspirado de partida. Este algoritmo ha sido 'The Whale Optimization Algorithm' desarrollado y estudiado por los investigadores Seyedal Mirjalili y Andrew Lewis ambos de la universidad de Griffith.

Para la comparativa y estudio de los resultados hemos puesto nuestro algoritmo a calcular los mínimos de 20 funciones. Estas funciones son las mismas que se utilizaron en la competición CEC2014, competición que pasaré a comentar brevemente en la siguiente sección.

1.1. Competición CEC2014

Esta competición es un referente a nivel mundial en cuanto a competiciones de algoritmos se refiere. La intención de la competición es poner todos los algoritmos a ejecutar con diferentes dimensiones: 10,30,100,... de forma que se puedan comparar a posteriori los resultados obtenidos por los mismos.

El problema consiste en minimizar 30 funciones que a priori no tienen un mínimo fácilmente localizable, como única garantía se tiene que el mínimo está localizado en el compacto $[-100, 100]^n$ donde n es la dimensión considerada.

En esta competición del año 2014 se evaluaron las funciones en dimensión 10,30,50 y 100 pero en este trabajo sólo hemos abarcado el problema de dimensión 10 y 30. Además cabe destacar que el algoritmo ganador en la competición fue L-SHADE.

1.2. Módulo de funciones CEC2014

Para poder ejecutar el algoritmo diseñado sobre las funciones de la competición CEC2014 primero tenemos que tener una implementación de las mismas. Mi decisión fue utilizar Python para el desarrollo del algoritmo, por la rapidez y facilidad en la escritura y sintaxis y por disponer de bibliotecas muy potentes en las que poder apoyar mi código.

La implementación de las funciones nos fue dada en C++, por lo que tuve que hacer un módulo en Python primero que importase las funciones de C++ para poder utilizarlas en Python. Este trabajo lo hice de forma conjunta con mi compañero Pablo Baeyens utilizando la biblioteca Cython. Tras esto implementamos dos interfaces muy sencillas que permiten ejecutar las primeras 20 funciones de la competición a través de una clase Benchmark.

Cabe destacar que esta idea se me ocurrió tras ver que el profesor de la Universidad de Granada en Ceuta Daniel Molina había desarrollado un módulo parecido para la competición de 2013 [1]. Tras contactar con el y apoyarnos en su código terminamos de desarrollar el módulo de las funciones y lo subimos a GitHub dejándolo a disposición del resto de alumnos junto con el código de otro compañero que realizó el mismo trabajo para el lenguaje Julia [2].

2. Descripción del algoritmo inicial

3. Desarrollo de mejoras

4. Versión final

5. Resultados

Referencias

- [1] <https://github.com/dmolina/cec2013lsgo>. Funciones CEC2013 de Daniel Molina.
- [2] <https://github.com/nacheteam/cec2014>. Funciones CEC2014.