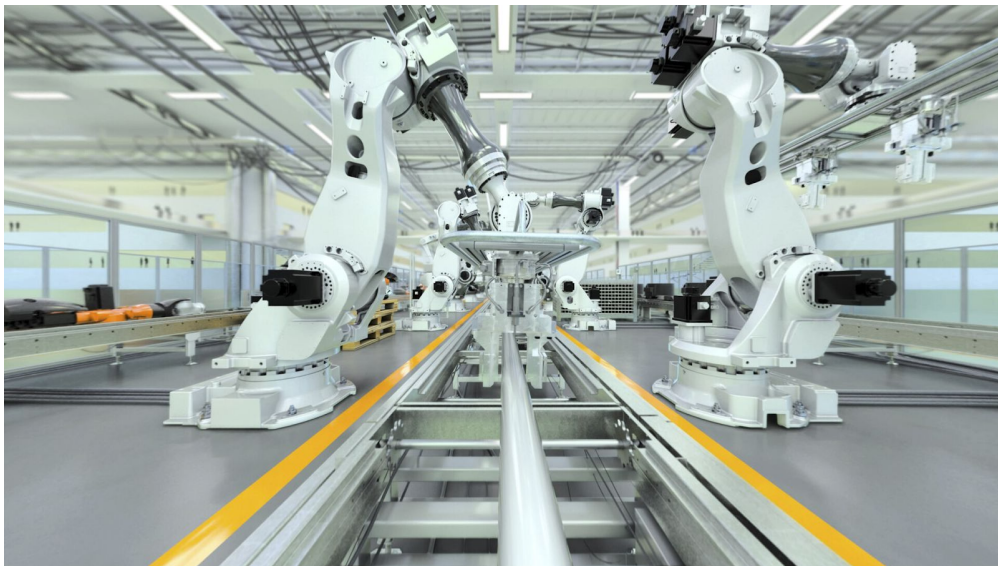


Calibración Automática de motores mediante Estrategias Evolutivas

Práctica 2

ALGORITMOS GENÉTICOS Y EVOLUTIVOS



Miguel Espinosa Miñano
100383404@alumnos.uc3m.es
Grado en Ingeniería Informática
Curso 2020-2021

Índice

Índice	2
Introducción	3
Procedimiento (explicación del código)	3
Parámetros y modelo	4
Análisis y comparativa de resultados	4
Parámetros estáticos	4
Generacional	4
Tamaño de la población	5
Elitismo puro	6
Tamaño de la población	6
Tamaño de torneo	8
Tamaño de mutación	9
Combinación ambos <code>t_size</code> y <code>m_size</code>	10
Parámetros dinámicos	11
Dynamic tournament size	11
Razonamiento	12
Experimento	12
Dynamic mutation ratio	13
Razonamiento	13
Experimento	14
Dynamic tournament size and mutation ratio	15
Mejor resultado	16
Problemas encontrados y conclusiones	17

Introducción

En esta práctica se plantea la resolución de la calibración correcta de los motores de un brazo de robot únicamente con un láser, una cámara y un objetivo de referencia con la finalidad de poder realizar soldaduras de alta precisión.

El objetivo de la práctica es resolver dicho problema con una estrategia evolutiva que permita calibrar de forma automática y con la mayor precisión posible el brazo robot. Para ello, se proporciona una función de adecuación que devuelve el error cometido en dicha aproximación de los motores. Se buscará siempre minimizar dicho valor de error.

El código se ha desarrollado en Python, y se han implementado dos estrategias evolutivas:

- $(1+1)$ -EE
- $(\mu+\lambda)$ -EE

Razonamiento general

Como toda estrategia evolutiva:

· Inicialmente...

Se buscarán valores grandes para el vector de varianzas. Esto permite mayor *exploración* en el espacio de búsqueda y poder realizar grandes saltos a lugares distantes. Esto permite encontrar soluciones mejores rápidamente y acotar rápidamente el espacio de búsqueda a zonas más prometedoras.

· Finalizando...

Se quiere realizar una *explotación* del espacio de búsqueda actual. Los valores del vector de varianzas serán cada vez más pequeños, de esta forma, afinando mejor y encontrando el subóptimo al que nos acercamos

(1 + 1) - EE

Se ha implementado la estrategia 1+1, donde la población está compuesta de un único individuo, y por lo tanto, es un método de escalada puro. La solución implementada funciona de igual manera para ambos casos: con 4 motores y con 10 motores.

Implementación

La implementación y flujo de ejecución del primer modelo (1+1) puede resumirse de la siguiente manera:

1. Inicialización aleatoria del vector de codificación con valores entre el rango permitido $[-180, 180]$
2. Inicialización aleatoria del vector de varianzas con valores entre 0 y 10.
3. Para el número máximo de generaciones:
 - a. Creación de un hijo:
 - i. Mutación del vector de codificación del padre según una distribución normal.
 - ii. El hijo contendrá una copia exacta del vector de mutaciones del padre
 - b. Eliminación del peor individuo
 - i. Se comparan los valores de adecuación del padre y del hijo y se elimina aquel con mayor fitness o error.
 - c. Se aumenta o disminuye la frecuencia de éxitos dependiendo si el hijo a resultado ser mejor que el padre o peor.
 - d. Se muta el vector de varianzas del individuo ganador siguiendo la regla del %.
4. Se itera (Paso 3) hasta que se cumple el criterio de parada (en este caso se ha considerado un fitness de 0.0) o se llega al número máximo de generaciones.

Parámetros

Los parámetros modificables para esta estrategia evolutiva son:

- ❖ s - tamaño de la ventana: Este parámetro define cuántas iteraciones se considerarán para calcular el ratio de mejoras. Ej. $s=10$ significa que guardaremos las mejoras o no mejoras de las últimas 10 generaciones.
- ❖ c - constante de la regla del %: Indica con qué rapidez deben crecer o decrecer los valores del vector de varianzas.
- ❖ n generations: Define el número máximo de generaciones.

Observaciones y mejoras

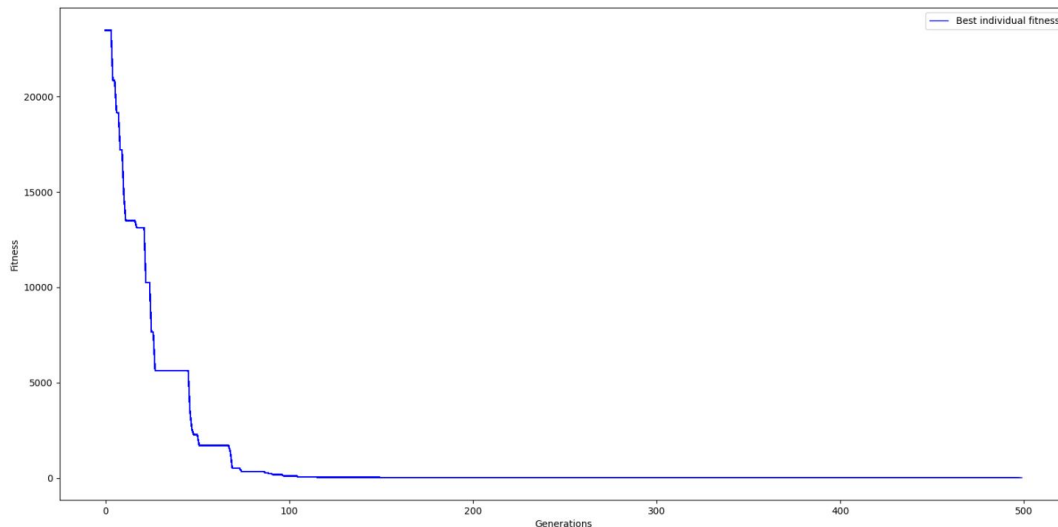
En esta sección se detallan algunas mejoras adicionales que se han llevado a cabo sobre el modelo estándar.

- Para asegurar una mejor convergencia, se ha añadido una pequeña modificación sobre el algoritmo. La intención es dar un ratio de mejoras menor al inicio (para fomentar mayores mutaciones) y un ratio de mejoras mayor al final. Esto se ha conseguido de la siguiente manera:
 - En la fase de eliminación del peor individuo se determina si esa generación ha sido una mejora o no.
 - Inicialmente, para la comparación de *fitnesses* de los individuos, la equiparación se lleva a cabo eliminando los decimales de los valores de fitness. Esto permite aceptar únicamente soluciones que mejoren sustancialmente al individuo anterior y así, avanzar más rápidamente (sin necesidad de ir generando hijos que son solamente un “poquito” mejor que el padre)
 - De esta manera, inicialmente se redondean los valores de adecuación a la decena más cercana, más tarde se redondeará al entero más cercano, después en el orden de 0.1 y finalmente en el orden de 0.01.
 - Esta modificación nos permite converger más rápidamente evitando individuos “malos” intermedios que ralentizan nuestra evolución. Y a la vez, permite afinar con mayor exactitud una vez se ha convergido a un subóptimo.
 - Todo esto repercute en el ratio de aciertos por cada s generaciones, que a su vez repercute en la mutación de las varianzas. Inicialmente, el ratio de mejoras es menor, y por lo tanto las varianzas mayores. Finalmente, el ratio de mejoras es mayor, y por lo tanto los valores de las varianzas menores.
- Cabe destacar que para las primeras s (tamaño de la ventana) iteraciones siempre se aumentará el valor de las varianzas.
- El código genera dos archivos:
 - Una gráfica del *fitness* del mejor individuo de cada generación
 - Un archivo con la evolución del mejor individuo con sus respectivos valores para los vectores de codificación y varianza y los parámetros del experimento.
- Se crea una sesión para mejorar la interacción con el servidor. También, siempre que sea posible se procurará conectarse con VPN a la red de la universidad para disminuir la latencia.

Análisis y comparativa de resultados

Cabe mencionar que se han realizado una gran cantidad de experimentos para la verificación del correcto funcionamiento del programa.

A continuación se muestran un ejemplo de las gráficas que se obtienen en tiempo real para la evaluación del valor de adecuación del individuo en cada generación.



Se han realizado múltiples experimentos con distintos tamaños de ventana, llegando a la conclusión que el valor del parámetro s deberá oscilar entre los 10 y 15. De la misma forma, se ha comprobado que la constante de la regla $\frac{1}{s}$ debe tomar valores cercanos a 0,82 (tal y como se demostró experimentalmente por la comunidad científica).

Se ha descubierto que los resultados finales de las ejecuciones son bastante variados entre sí. La razón principal es debido a que se trata de una población con un solo individuo y que por lo tanto, suele estancarse en mínimos subóptimos prematuramente debida la rápida falta de variabilidad genética. Es por esto que cada ejecución puede dar resultados bastante diferentes dada la aleatoriedad del proceso.

La media de mejores individuos con la estrategia evolutiva 1+1 cae entorno:

- 4 motores: entre 5 y 10
- 10 motores: alto rango, entre 100-1000

Mejor individuo:

Los mejores individuos que se han conseguido utilizando la estrategia evolutiva 1+1 son:

- **4 motores:** Valor de adecuación = **0.999535960834**
[$c_1 = 9.12897937$; $c_2 = 5.98947819$; $c_3 = 3.66910365$; $c_4 = 1.76911578$]

Número de generaciones: 1000

Número de evaluaciones aproximado: 2000

- **10 motores:** Valor de adecuación = **283.623112538**
[c1 = 1.23344752 ; c2 = 2.48478012 ; c3 = 3.3129611 ; c4 = 14.68599473 ;
c5 = -1.56889646 ; c6 = 4.97187625 ; c7 = 8.29871384 ; c8 = -0.06089628 ; c9 =
2.22878246 ; c10 = 0.95671166]

Número de generaciones: 1000

Número de evaluaciones aproximado: 2000

Para ambos casos, los parámetros elegidos han sido:

- s (tamaño ventana) = entre 10 y 15
- c (constante regla $\frac{1}{s}$) = 0,82
- n-max-generaciones = 1000

$(\mu + \lambda)$ - EE

En este segundo caso, la población está compuesta de μ individuos y en cada generación se crean λ hijos. De toda la población (padres e hijos) se seleccionarán a los μ mejores individuos que serán los que pasarán a la siguiente generación.

Implementación

La implementación y flujo de ejecución del segundo modelo $(\mu+\lambda)$ puede resumirse de la siguiente manera:

5. Inicialización aleatoria del vector de codificación con valores entre el rango permitido $[-180, 180]$ para toda la población
6. Inicialización aleatoria del vector de varianzas con valores entre 0 y 10.
7. Para el número máximo de generaciones:
 - a. Creación de la población de hijos λ :
 - i. Selección de los padres mediante un torneo
 - ii. Cruce entre los padres (media de sus valores del vector de codificación)
 - iii. Cruce de las varianzas de los padres (se elige una de entre los padres aleatoriamente)
 - iv. Mutación del vector de codificación del hijo según una distribución normal.
 - v. Mutación del vector de varianzas siguiendo la fórmula:

$$\sigma' = e^{N(0, \tau_0)} \cdot \sigma \cdot e^{N(0, \tau)}$$
 - vi. Clip el array por si hubiera valores que se salen del rango permitido
 - b. Eliminación de los peores λ individuos
 - i. Se comparan los valores de adecuación de la población de padres e hijos y se eliminan aquellos con mayor fitness o error.
 - c. Se muestran los resultados gráficamente.
8. Se itera (Paso 3) hasta que se cumple el criterio de parada (en este caso se ha considerado un fitness de 0.0) o se llega al número máximo de generaciones.

Parámetros

Los parámetros modificables para esta estrategia evolutiva son:

- ❖ El tamaño de la población
- ❖ El número de hijos que se generan en cada población
- ❖ El tamaño de los torneos a utilizar

❖ El número de generaciones máximas

Observaciones y mejoras

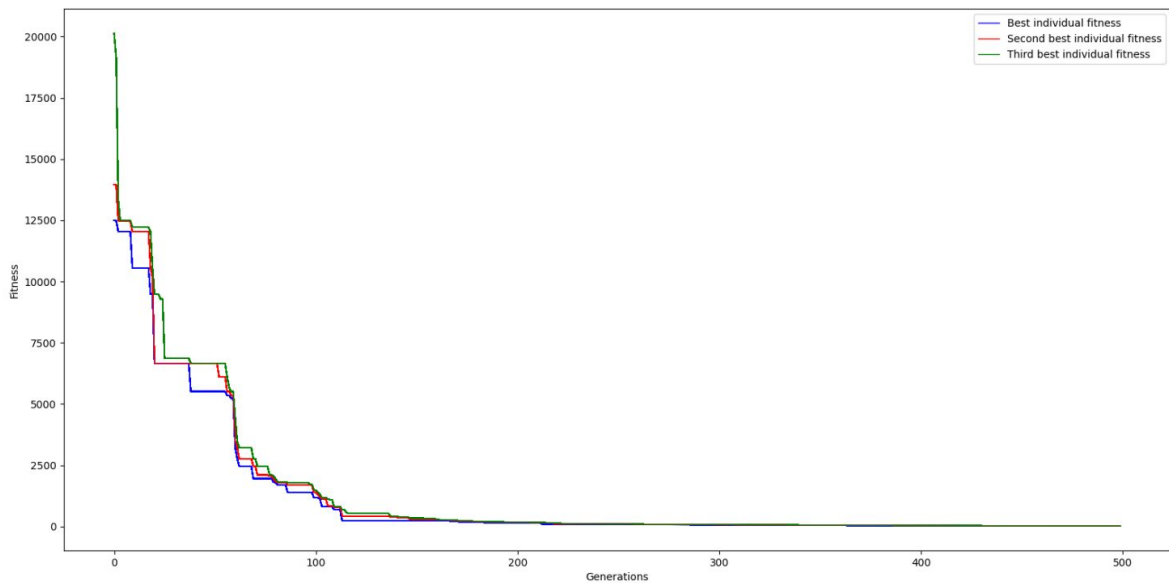
En esta sección se detallan algunas mejoras adicionales que se han llevado a cabo sobre el modelo estándar.

- A menudo se puede observar como se pierde rápidamente la variabilidad genética. Para evitar dicha declinación se ha optado por multiplicar las varianzas por 10 cuando la desviación estándar de estas para la población es menor a 0,01. Se han visto mejoras de forma experimental con esta nueva regla, evitando convergencias prematuras y añadiendo así más variabilidad genética para nuevas optimizaciones.
- El código genera dos archivos:
 - Una gráfica del *fitness* del mejor individuo de cada generación
 - Un archivo con la evolución del mejor individuo con sus respectivos valores para los vectores de codificación y varianza y los parámetros del experimento.
- Se crea una sesión para mejorar la interacción con el servidor. También, siempre que sea posible se procurará conectarse con VPN a la red de la universidad para disminuir la latencia.

Análisis y comparativa de resultados

Cabe mencionar que se han realizado múltiples experimentos para la verificación del correcto funcionamiento del programa.

A continuación se muestran un ejemplo de las gráficas que se obtienen en tiempo real para la evaluación del valor de adecuación de los tres mejores individuos en cada generación.



Mejor individuo:

Los mejores individuos que se han conseguido utilizando la estrategia evolutiva 1+1 son:

- **4 motores:** Valor de adecuación = **0.0** → VALOR ÓPTIMO

[c1=10.12345679, c2= 5.987654 , c3= 3.666666 , c4= 1.7654321]

Número de generaciones: 68

Número de evaluaciones aproximado: 20400

Tamaño de la población: 100

Número de hijos por generación: 200

Tamaño del torneo: 3

- **6 motores:** Valor de adecuación = **0.0** → VALOR ÓPTIMO

[c1=3.22345679; c2= 2.487654 ; c3= 2.324663 ; c4 = 1.7654321 ; c5 = 1.43244343 ; c6 = 3.54354565]

Número de generaciones: 119

Número de evaluaciones aproximado: 124950

Tamaño de la población: 600

Número de hijos por generación: 450

Tamaño del torneo: 4

- 10 motores: Valor de adecuación = 0.00747024793554

Número de generaciones: 190

Número de evaluaciones aproximado: 199500

Tamaño de la población: 600

Número de hijos por generación: 450

Tamaño del torneo: 4

- 10 motores version B: Valor de adecuación = 2.68641626818e-05
[15.30456875, 25.96792993, -7.98476624, -24.47909557,
1.39439924, 7.39929318, 14.25053738, -22.20521646, 30.68908963,
-27.71919779]

Número de generaciones: 10

Número de evaluaciones aproximado: 10500

Tamaño de la población: 600

Número de hijos por generación: 450

Tamaño del torneo: 4