



UNIVERSIDAD  
DE GRANADA

ETSIIT

Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación



---

## Práctica 4: Propuesta de Metaheurística HALO

Metaheurísticas

Grupo A3 — Lunes de 17:30 a 19:30

---

Autor:

**Lugli, Valentino Glauco** · YB0819879 · [valentinolugli@correo.ugr.es](mailto:valentinolugli@correo.ugr.es)

# Índice

<b>1</b>	<b>Resumen</b>	<b>2</b>
1.1	Inspiración	2
1.2	Intuición	2
1.3	Funcionamiento	3
1.3.1	Método principal	4
1.3.2	Función AGG	5
1.3.3	Función Guerra	6
1.3.4	Registro Pop	7
1.3.5	Clase Chromosome	7
<b>2</b>	<b>Análisis de Rendimiento</b>	<b>7</b>
2.1	1 % de Evaluaciones	8
2.2	50 % de Evaluaciones	9
2.3	100 % de Evaluaciones	10
<b>3</b>	<b>Hibridación y Análisis de Rendimiento</b>	<b>11</b>
3.1	Procedimiento Principal Hibridizado	11
3.2	Análisis de Rendimiento	13
3.2.1	1 % de Evaluaciones	14
3.2.2	50 % de Evaluaciones	15
3.2.3	100 % de Evaluaciones	16
<b>4</b>	<b>Propuestas de Mejora de la Metaheurística</b>	<b>16</b>

---

# 1. Resumen

En esta práctica se presenta una nueva metaheurística genética con múltiples poblaciones adversarias denominada que se ha denominado “Halo”.

## 1.1. Inspiración

La metaheurística ha surgido como inspiración del autor en una serie de videojuegos de ciencia ficción y de disparos desarrollados inicialmente por la compañía *Bungie* y posteriormente por *343 Industries* para las diferentes consolas *Xbox* y ordenadores personales llamado de la misma manera que la metaheurística: Halo.

En esta saga, se sigue la historia principalmente de un supersoldado llamado John-117 y su Inteligencia Artificial Cortana, quienes forman parte de la *UNSC*, “Comando Espacial de las Naciones Unidas” – en esencia la rama militar de la exploración del espacio del Gobierno Unificado de la Tierra – en el siglo XXVI en el momento en el que la Humanidad se encuentra en una gran guerra interestelar con una alianza de alienígenas llamados *The Covenant*, “El Pacto” quienes han marcado a la Humanidad como herejes de su religión.

Luego de una fuerte pérdida para la Humanidad, la tripulación del *Pillar of Autumn* –dónde se encuentra en animación suspendida John-117– intentando escapar de El Pacto, llegan a un sistema solar en los límites de la Vía Láctea en donde se consiguen con una megaestructura: Un Mundo Anillo, *Halo*.

Sus perseguidores también llegan y la tripulación se ve forzada a realizar un aterrizaje de emergencia en este extraño mundo que posee flora y fauna, aunque es una estructura de origen artificial.

Posteriormente se descubre que esta estructura servía como un laboratorio, y entre los experimentos, se tenían almacenados los restos de una antigua especie parasitaria llamada El *Flood* que casi eliminan la vida en la galaxia hace miles de años, quienes son liberados por los enfrentamientos de la Humanidad y El Pacto; ahora, la batalla también es contra de esta especie que infecta y asimila toda materia orgánica que entra en contacto con el parásito.

## 1.2. Intuición

La propuesta se puede clasificar como una metaheurística genética de múltiples poblaciones adversarias.

Se tienen tres poblaciones distintas y en principio, disjuntas: La Humanidad ( $P_H$ ), El *Covenant* ( $P_C$ ) y El *Flood* ( $P_F$ ) quienes se encuentran en una guerra multigeneracional por la estabilidad y el control de la galaxia.

Las poblaciones, mientras no están en una guerra, se encuentran en sus planetas avanzando sus investigaciones, tecnologías, tácticas, teniendo descendientes y planificando su nueva ofensiva contra las otras poblaciones: Se definen dos estados para las poblaciones, un estado de guerra  $E_G$  y un estado de relativa paz,  $E_P$ .

En  $E_P$ , las poblaciones se reproducen y mutan de diferentes maneras acorde con su biología o cultura:

- La población  $P_H$ , representando los humanos, realizan un cruce por competición de dos en dos individuos y tienen una probabilidad de mutación intermedia, aproximadamente de 0.001 % por gen. Esta es la población balanceada: Se realiza tanto exploración como explotación en grados similares.
- La población  $P_C$ , representando alienígenas religiosos, no toleran las mutaciones de sus individuos y además,
- La población  $P_F$  al ser parásitos, tienen mutaciones muy extremas y su manera de compartir material genético.

Luego de un tiempo  $T_R$ , las poblaciones entran en  $E_G$ . En este estado, se realiza una medición del fitness de los mejores individuos de cada población y dependiendo de qué población posee el mejor fitness, se entra en el operador de guerra, aquí, la población con mayor fitness compara un trozo de sus cromosomas con un porcentaje de los cromosomas de las otras poblaciones; si el fitness de un individuo de la población “principal” es menor que de alguna de las poblaciones perdedoras, se copia ese individuo a la población reemplazando el anterior, por otro lado si es un fitness peor, se elimina ese individuo y se regenera.

Se finaliza esta etapa de guerra y las poblaciones continúan mutando hasta que se alcance el límite de evaluaciones.

### 1.3. Funcionamiento

Formalmente, se tienen tres poblaciones  $P_H$ ,  $P_C$  y  $P_F$  de 30 individuos cada una, inicializadas aleatoriamente. Se posee un bucle principal dónde comienzan a evolucionar en la etapa  $E_P$ . Se utilizó un algoritmo genético generacional con elitismo. Todas las poblaciones utilizan un torneo binario para la selección de los individuos.

Como se ha comentado anteriormente, cada población se intenta que posea diferentes características que le permitan explorar el espacio de soluciones de una manera diferente, por lo tanto, los operadores de cruce y mutación son diferentes para cada una, aún así se mantiene que la probabilidad de cruce es de 0,7 y la probabilidad de mutación es  $0,1 \times \text{tamaño de población}$ .

- $P_H$ :
  - Operador de Cruce: Cruce BLX- $\alpha$ , con  $\alpha = 0,5$ .
  - Operador de Mutación: Mutación No Uniforme (*Non Uniform Mutation*)
- $P_C$ :
  - Operador de Cruce: Cruce Aritmético con  $\lambda \in [0,25,0,75]$  seleccionado de manera aleatoria con probabilidad uniforme.
  - Operador de Mutación: Mutación de Mühlenbein (*Mühlenbein's Mutation*)
- $P_F$ :
  - Operador de Cruce: Cruce Lineal con 3 hijos, seleccionando los 2 mejores.
  - Operador de Mutación: Mutación Aleatoria.

Estos operadores fueron seleccionados por sus características de exploración y explotación que ofrecen<sup>1</sup>, de manera que  $P_H$  posea un balance entre exploración y explotación,  $P_C$  tenga más explotación que exploración y  $P_F$ , al contrario, tenga más exploración que explotación. En esta propuesta original, en se limita esta etapa  $E_P$  a un límite de evaluaciones de  $250 \times \text{dimensión}$  para cada una de las poblaciones.

Luego, cada dos iteraciones –no evaluaciones– de la etapa  $E_P$ , sucede la etapa  $E_G$ . Se compara el fitness de  $P_H$ ,  $P_C$  y  $P_F$  para obtener quién posee el más pequeño; una vez sucedido esto, se compara quién de las dos poblaciones restantes poseen de media un mejor fitness.

La población con el mejor fitness se le denomina  $P^\uparrow$ , a la población con segundo mejor fitness se le denomina  $P^{\leftrightarrow}$  y la tercera se denomina  $P^\downarrow$ .

Estocásticamente se determinan cuatro valores,  $s_\alpha$ ,  $f_\alpha$ ,  $s_\beta$ ,  $f_\beta$  con  $s_\alpha, s_\beta \in [0, n-1]$ ,  $f_\alpha \in [0, 25n, 0, 75n]$  y  $f_\beta \in [0, 0, 5n]$  siendo  $n$  la longitud de la población; que determinan el punto de inicio y la cantidad de cromosomas que serán comparados de las tres poblaciones; se compara en un bucle  $P^\uparrow$  con  $P^{\leftrightarrow}$  y en otro  $P^\downarrow$

Si  $P^\uparrow[i]$  es mejor que  $P^{\leftrightarrow}[i]$  o  $P^\downarrow[i]$ , entonces a estas segundas poblaciones se reemplaza ese cromosoma con uno nuevo generado a partir del mejor individuo de su población y otro individuo seleccionado aleatoriamente, se realiza un Cruce Plano (*Flat Crossover*) para ello.

En caso contrario,  $P^\uparrow[i]$  se reemplaza por  $P^{\leftrightarrow}[i]$  o  $P^\downarrow[i]$  y se termina la etapa  $E_G$ , se vuelve a la etapa  $E_P$  donde las poblaciones tendrán que reevaluar los cromosomas regenerados y/o evolucionar con el nuevo material genético que han copiado de otras poblaciones.

### 1.3.1. Método principal

Todo lo anterior se puede resumir en el siguiente pseudocódigo, debe notarse que las poblaciones están contenidas en un registro denominado `pop` la cual almacena el mejor individuo, el fitness promedio y las poblaciones como tal en otra clase llamada `cromosome` que son descritas más abajo.

```
procedimiento Halo (entero: dimension)
variables:
    entero: iteraciones<-0, evaluaciones<-0
    doble: mejorFitness
    // pop es una clase hecha específicamente para el algoritmo.
    pop: pH, pC, pF
inicio:
    inicializarPoblacion(pH, tamañoPop, dimension)
    inicializarPoblacion(pC, tamañoPop, dimension)
    inicializarPoblacion(pF, tamañoPop, dimension)

    evaluarPop(pH)
    evaluarPop(pC)
    evaluarPop(pf)

    hacer
```

<sup>1</sup>Herrera, F., Lozano, M. & Verdegay, J. "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis." *Artificial Intelligence Review* 12, 265–319 (1998). <https://doi.org/10.1023/A:1006504901164>

```

    si(iteraciones==2)entonces
        iteraciones<-0
        si(mejorFitness = pH.mejorIndividuo.fitness)entonces
            si(pC.mediaFitness < pF.mediaFitness) entonces
                guerra(pH, pF, pC)
            sino
                guerra(pH, pC, pF)
            fsi
        sino si (mejorFitness = pC.mejorIndividuo.fitness)
            si(pH.mediaFitness < pF.mediaFitness) entonces
                guerra(pC, pF, pH)
            sino
                guerra(pC, pH, pF)
            fsi
        sino
            si(pH.mediaFitness < pC.mediaFitness) entonces
                guerra(pF, pC, pH)
            sino
                guerra(pF, pH, pC)
            fsi
        fsi
    fsi
    iteraciones++

    evaluaciones += AGG(pH, dim, 0)
    evaluaciones += AGG(pC, dim, 1)
    evaluaciones += AGG(pF, dim, 2)

    mejorFitness <- min(pH.mejorIndividuo.fitness, pC.mejorIndividuo.fitness,
        pF.mejorIndividuo.fitness)
    mientras (evaluaciones < 1000 * dim)
fproc

```

### 1.3.2. Función AGG

```

procedimiento AGG(pop: &poblacion, entero: dimension, entero: tipoPoblacion)
variables:
    entero: evaluaciones <- 0
    vector de cromosome: poblacionPadre
    bestCromosome: mejorCromosoma
inicio:

    // Puede suceder que se modifiquen cromosomas en Guerra y deben reevaluarse aquí.
    evaluaciones += evaluarPoblacion(poblacion)

    mientras (evaluaciones < 250 * dimension) hacer
        seleccionPoblacion(poblacion, poblacionPadre, mejorCromosoma)
        switch(tipoPoblacion)
            caso 0: // Población  $P_H$ 
                BLX_Alpha(poblacionPadre, mejorCromosoma)
                mutacionNoUniforme(poblacionPadre, mejorCromosoma, evaluaciones,
                    250*dim)
            break

```

```

    caso 1: // Población  $P_C$ 
        cruceArimetico(poblacionPadre, mejorCromosoma)
        mutacionMuhlenbein(poblacionPadre, mejorCromosoma)
    break

    caso 2: // Población  $P_F$ 
        evaluaciones += cruceLineal(poblacionPadre, mejorCromosoma)
        mutacionAleatoria(poblacionPadre, mejorCromosoma)
    break

    reemplazarPoblacionGen(poblacion, poblacionPadre, mejorCromosoma)
    evaluaciones += evaluarPoblacion(poblacion)
fmientras
fproc

```

### 1.3.3. Función Guerra

```

procedimiento Guerra(pop: &principal, pop: &primerPerdedor, pop: &segundoPerdedor)
variables:
    entero: sP, fP, sS, fS
inicio:
    sP <- randomUniforme(0, principal.genesis.tamaño()-1)
    sS <- randomUniforme(0, principal.genesis.tamaño()-1)
    fP <- randomUniforme(principal.genesis.tamaño() * 0.25, principal.genesis.tamaño()
        * 0.5)
    fS <- randomUniforme(principal.genesis.tamaño() * 0, principal.genesis.tamaño() *
        0.5)

    para i<-sP hasta fP hacer
        si (primerPerdedor.genesis[i].obtenerFitness() > principal.genesis[i].
            obtenerFitness()) entonces
            // Se utiliza cruce plano (Flat Crossover)
            primerPerdedor.genesis[i].regenerarHijo(primerPerdedor.genesis[
                primerPerdedor.mejorIndividuo.indice].obtenerGenes(), primerPerdedor
                    .genes[randomUniforme(0, principal.genesis.tamaño()-1).obtenerGenes()])
        sino
            principal.genesis[i].copiar(primerPerdedor.genesis[i])
        fsi
    fpara

    para i<-sS hasta fS hacer
        si (segundoPerdedor.genesis[i].obtenerFitness() > principal.genesis[i].
            obtenerFitness()) entonces
            segundoPerdedor.genesis[i].regenerarHijo(segundoPerdedor.genesis[
                segundoPerdedor.mejorIndividuo.indice].obtenerGenes(),
                segundoPerdedor.genesis[randomUniforme(0, principal.genesis.tamaño()-1).
                    obtenerGenes()])
        sino
            principal.genesis[i].copiar(segundoPerdedor.genesis[i])
        fsi
    fpara
fproc

```

---

### 1.3.4. Registro Pop

```
registro pop
  vector de cromosome: genes
  par<entero, doble>: mejorIndividuo
  doble: mediaFitness
fpop
```

### 1.3.5. Clase Cromosome

```
clase cromosome
  atributos:
    privados:
      vector de doble: genes
      doble: fitness
      booleano: tieneCambio
  metodos:
    publicos: /* Se obvian constructores y destructores */
      func obtenerGenes(): vector de doble
      func tamañoGen(): entero
      func obtenerFitness(): flotante
      func obtenerCambios(): booleano
      proc asignarGen(vector de doble: argGen)
      proc asignarFitness(doble: argFitness)
      proc asignarCambios(booleano: argCambios)
      proc cambiarGen(entero: indice, doble: info)
      /* ... y más funciones ...*/
fclase
```

## 2. Análisis de Rendimiento

Se realizaron los análisis con el paquete de Benchmarks CEC2017, con dimensión 10 y 30 y para cada una de las 30 funciones se ejecutó el algoritmo 10 veces, se ha decidido comparar la metaheurística con *Differential Evolution* y *Particle Swam Optimization* debido a que son unos algoritmos similares, puesto que DE es un algoritmo evolutivo al igual que HALO y PSO es un algoritmo basado en poblaciones, lo cual también comparte HALO, por lo tanto eran dos elecciones sensibles para comparar esta nueva metaheurística.



## 2.1. 1 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HALO	PSO		DE	HALO	PSO
F01	2.645e+10	8.753e+09	7.208e+09	F01	7.705e+10	6.040e+09	3.432e+10
F02	2.398e+17	2.056e+11	1.000e+00	F02	1.217e+60	5.184e+48	1.000e+00
F03	1.696e+04	2.374e+04	3.167e+04	F03	8.932e+04	1.654e+05	2.020e+05
F04	4.561e+03	4.728e+02	4.702e+02	F04	2.976e+04	8.015e+02	1.011e+04
F05	1.648e+02	1.059e+02	1.013e+02	F05	4.777e+02	1.648e+02	4.120e+02
F06	6.946e+01	5.680e+01	5.398e+01	F06	9.634e+01	2.014e+01	9.139e+01
F07	1.409e+02	2.007e+02	1.764e+02	F07	7.873e+02	3.340e+02	8.545e+02
F08	7.442e+01	8.155e+01	8.991e+01	F08	3.758e+02	1.609e+02	3.747e+02
F09	1.285e+03	1.302e+03	1.504e+03	F09	1.192e+04	3.080e+03	1.508e+04
F10	3.038e+03	2.195e+03	1.951e+03	F10	7.713e+03	4.839e+03	8.462e+03
F11	3.033e+07	2.133e+03	2.552e+03	F11	1.588e+08	4.925e+03	9.459e+03
F12	4.522e+09	1.861e+08	2.570e+08	F12	2.635e+10	2.777e+08	7.249e+09
F13	2.169e+09	6.886e+06	4.722e+06	F13	3.841e+10	4.896e+07	4.005e+09
F14	1.624e+09	1.889e+04	4.792e+03	F14	7.007e+08	2.512e+06	3.677e+06
F15	2.786e+08	3.257e+04	5.283e+04	F15	4.480e+09	8.075e+07	2.550e+08
F16	1.113e+03	6.408e+02	6.786e+02	F16	2.087e+04	1.662e+03	3.342e+03
F17	4.090e+02	3.083e+02	2.967e+02	F17	1.415e+05	6.171e+02	1.662e+03
F18	1.166e+10	7.192e+07	2.861e+07	F18	3.063e+09	7.734e+06	5.541e+07
F19	9.707e+09	6.292e+05	9.718e+05	F19	4.614e+09	4.209e+07	3.942e+08
F20	5.149e+02	2.615e+02	2.913e+02	F20	1.774e+03	6.972e+02	1.200e+03
F21	6.010e+02	2.791e+02	2.791e+02	F21	9.606e+02	3.369e+02	6.174e+02
F22	2.468e+03	7.166e+02	5.520e+02	F22	8.536e+03	1.597e+03	6.460e+03
F23	1.402e+03	4.378e+02	4.265e+02	F23	4.233e+03	5.766e+02	1.024e+03
F24	9.159e+02	4.832e+02	4.233e+02	F24	2.667e+03	7.090e+02	1.086e+03
F25	2.005e+03	7.738e+02	7.664e+02	F25	5.281e+03	7.358e+02	2.901e+03
F26	2.705e+03	1.551e+03	1.273e+03	F26	1.198e+04	3.535e+03	6.772e+03
F27	1.545e+03	5.425e+02	5.433e+02	F27	6.076e+03	6.457e+02	1.486e+03
F28	1.530e+03	9.025e+02	9.516e+02	F28	6.610e+03	1.271e+03	3.758e+03
F29	1.368e+04	7.160e+02	6.152e+02	F29	1.015e+05	1.354e+03	3.373e+03
F30	3.782e+08	1.753e+07	1.184e+07	F30	8.719e+09	2.351e+07	3.683e+08
Best	4	8	18	Best	1	28	1

## 2.2. 50 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HALO	PSO		DE	HALO	PSO
F01	1.997e-05	6.369e+08	1.847e+08	F01	1.787e+07	1.436e+07	7.702e+09
F02	0.000e+00	9.633e+07	1.000e+00	F02	5.436e+23	2.620e+26	1.000e+00
F03	2.810e-08	5.347e+03	3.460e+03	F03	1.370e+04	8.389e+04	7.122e+04
F04	2.359e-01	6.758e+01	6.360e+01	F04	9.421e+01	1.423e+02	1.663e+03
F05	1.151e+02	5.725e+01	4.730e+01	F05	2.197e+02	9.693e+01	2.600e+02
F06	3.469e+01	2.576e+01	1.361e+01	F06	2.136e+01	1.163e+00	4.667e+01
F07	4.377e+01	8.316e+01	5.702e+01	F07	2.524e+02	1.676e+02	4.096e+02
F08	2.983e+01	4.835e+01	2.967e+01	F08	2.093e+02	1.012e+02	2.156e+02
F09	2.089e+02	3.120e+02	8.356e+01	F09	2.785e+03	1.268e+03	4.100e+03
F10	5.311e+02	1.308e+03	1.323e+03	F10	4.422e+03	3.312e+03	7.431e+03
F11	5.896e+00	1.621e+02	5.042e+01	F11	1.118e+02	1.155e+03	1.557e+03
F12	7.664e+01	1.758e+07	4.645e+06	F12	5.635e+06	9.651e+06	6.936e+08
F13	1.109e+01	6.717e+04	1.227e+04	F13	1.513e+03	1.154e+06	2.357e+08
F14	1.545e+01	2.094e+02	1.643e+02	F14	8.108e+01	6.601e+05	4.771e+05
F15	4.371e-01	1.740e+03	3.228e+03	F15	1.141e+02	5.172e+03	1.265e+06
F16	4.562e+02	2.416e+02	1.850e+02	F16	1.510e+03	1.123e+03	1.859e+03
F17	3.118e+01	1.090e+02	7.631e+01	F17	5.825e+02	3.494e+02	5.510e+02
F18	1.672e-01	3.476e+05	2.653e+04	F18	3.292e+02	6.549e+05	4.171e+06
F19	4.603e-01	3.731e+03	5.864e+03	F19	4.629e+01	1.382e+04	2.330e+06
F20	3.837e+02	1.480e+02	1.100e+02	F20	3.109e+02	4.565e+02	6.190e+02
F21	1.918e+02	1.767e+02	1.421e+02	F21	3.502e+02	2.880e+02	4.487e+02
F22	1.024e+02	1.599e+02	1.025e+02	F22	1.184e+02	1.402e+02	1.354e+03
F23	8.115e+02	3.654e+02	3.371e+02	F23	5.524e+02	4.951e+02	6.911e+02
F24	1.000e+02	3.188e+02	2.041e+02	F24	6.206e+02	6.301e+02	7.470e+02
F25	4.040e+02	4.911e+02	4.536e+02	F25	3.896e+02	4.457e+02	7.749e+02
F26	2.706e+02	5.732e+02	4.175e+02	F26	4.896e+02	2.137e+03	3.733e+03
F27	3.897e+02	4.041e+02	4.200e+02	F27	5.145e+02	5.604e+02	8.850e+02
F28	3.518e+02	5.296e+02	5.235e+02	F28	4.370e+02	5.235e+02	1.288e+03
F29	2.514e+02	3.754e+02	3.508e+02	F29	1.166e+03	1.023e+03	1.609e+03
F30	8.052e+04	1.134e+06	1.485e+06	F30	1.572e+04	2.845e+05	2.024e+07
Best	22	0	8	Best	17	12	1

## 2.3. 100 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HALO	PSO		DE	HALO	PSO
F01	0.000e+00	3.902e+08	5.255e+07	F01	4.909e+04	4.320e+05	4.175e+09
F02	0.000e+00	1.549e+07	1.000e+00	F02	1.309e+19	1.744e+25	1.000e+00
F03	0.000e+00	3.900e+03	1.989e+03	F03	3.481e+03	7.603e+04	5.453e+04
F04	1.105e-04	4.870e+01	4.684e+01	F04	8.430e+01	1.137e+02	1.183e+03
F05	1.151e+02	5.039e+01	3.212e+01	F05	2.015e+02	9.099e+01	2.170e+02
F06	3.460e+01	2.043e+01	1.001e+01	F06	6.320e+00	4.504e-01	3.694e+01
F07	3.848e+01	7.078e+01	4.275e+01	F07	2.334e+02	1.448e+02	3.596e+02
F08	2.983e+01	4.543e+01	2.203e+01	F08	1.894e+02	9.888e+01	1.745e+02
F09	1.938e+02	2.081e+02	5.686e+01	F09	6.530e+01	1.014e+03	2.842e+03
F10	3.597e+02	1.277e+03	1.077e+03	F10	3.764e+03	3.234e+03	6.938e+03
F11	1.942e-02	1.378e+02	3.843e+01	F11	7.958e+01	6.790e+02	1.207e+03
F12	4.931e+00	1.141e+07	2.517e+06	F12	3.258e+05	3.720e+06	3.586e+08
F13	5.988e+00	5.932e+04	8.409e+03	F13	1.536e+02	1.308e+04	4.508e+07
F14	5.240e-02	1.337e+02	9.993e+01	F14	7.100e+01	6.304e+05	3.059e+05
F15	6.060e-02	1.520e+03	2.066e+03	F15	6.256e+01	1.893e+03	2.737e+05
F16	4.561e+02	2.061e+02	1.415e+02	F16	1.319e+03	1.076e+03	1.568e+03
F17	2.350e+01	9.410e+01	6.497e+01	F17	4.809e+02	3.457e+02	4.725e+02
F18	3.630e-02	1.537e+05	1.484e+04	F18	6.122e+01	4.340e+05	2.170e+06
F19	5.192e-03	2.053e+03	3.220e+03	F19	3.572e+01	4.033e+03	1.260e+06
F20	3.837e+02	1.377e+02	8.444e+01	F20	2.751e+02	4.454e+02	4.621e+02
F21	1.889e+02	1.645e+02	1.320e+02	F21	3.255e+02	2.863e+02	4.113e+02
F22	1.005e+02	1.249e+02	7.740e+01	F22	1.002e+02	1.221e+02	1.027e+03
F23	8.098e+02	3.574e+02	3.304e+02	F23	5.346e+02	4.914e+02	6.404e+02
F24	1.000e+02	3.062e+02	1.810e+02	F24	6.059e+02	6.268e+02	7.095e+02
F25	4.040e+02	4.646e+02	4.478e+02	F25	3.870e+02	4.129e+02	6.864e+02
F26	2.706e+02	4.753e+02	3.729e+02	F26	4.038e+02	2.060e+03	3.369e+03
F27	3.897e+02	4.018e+02	4.134e+02	F27	4.925e+02	5.595e+02	8.068e+02
F28	3.517e+02	5.032e+02	4.698e+02	F28	3.943e+02	4.821e+02	1.105e+03
F29	2.375e+02	3.508e+02	3.193e+02	F29	1.025e+03	9.820e+02	1.409e+03
F30	8.051e+04	6.909e+05	6.352e+05	F30	3.657e+03	5.837e+04	1.359e+07
Best	21	0	9	Best	19	10	1

A primeros rasgos puede notarse una diferencia considerable en la calidad de la solución obtenida por el algoritmo en dimensión 10 y dimensión 30: Al finalizar, con el 100 % de las evaluaciones realizadas se nota que el algoritmo HALO no llega a tener ninguna solución mejor que *Divergent Evolution* o *Particle Swarm Optimization* sin embargo una mirada más detallada a los resultados indica que HALO ha obtenido resultados que se encuentran entre DE y PSO en varios de los casos pero en su mayoría la calidad está ligeramente por detrás de lo que obtiene PSO.

Mientras que en dimensión 30, cambia el panorama favorablemente, HALO posee una ventaja mayor sobre PSO logrando en  $\frac{1}{3}$  de las ejecuciones ganarle a DE, si bien DE posee una mayor

---

ventaja teniendo prácticamente el resto de las mejores soluciones; dejando a PSO con una única solución mejor, de momento se puede ver que existe una correlación entre la dimensión a explorar y la calidad de la solución.

Con solo los resultados finales el algoritmo HALO se le pueden atribuir soluciones decentes pero mejorables frente a los algoritmos ya establecidos, de igual manera la calidad que obtiene para ser la primera iteración de una nueva metaheurística es notable.

De momento, y también, mirando como evoluciona la calidad de la solución de 1%, a 50% y últimamente hasta 100% puede verse que el algoritmo HALO no se ha quedado estancado en óptimos locales, se puede evidenciar si se obtiene la traza completa del error de fitness por medio de la herramienta TACOLab (<https://tacolab.org/bench>) de igual manera, sí es cierto que tarda más en converger –aunque también se debe tener en cuenta que para Dimensión 30, sucedió lo contrario y convergió más rápido al principio que DE y PSO, poseyendo 28 de 30 mejores evaluaciones– y esto debe tomarse en cuenta ya que se están gastando 3 veces más evaluaciones para que otros algoritmos debido a la manera en que funciona, esto soporta la hipótesis de que con mayores ajustes podría obtenerse una metaheurística potente.

Sin embargo, una de las cosas que hay que tomarse en cuenta es que, esta metaheurística posee muchos parámetros –si esto ya era verdad con un algoritmo genético, ahora son básicamente 3 en 1 y además de operadores adicionales– y también que debe balancearse las poblaciones y que tantas evaluaciones deberían realizarse en cada etapa  $E_P$ , puesto que esto limita su competitividad con otros algoritmos para una misma cantidad de evaluaciones.

Se puede concluir, entonces, que para ser una metaheurística nueva, los resultados obtenidos son prometedores aunque también se tendrá que tomar en cuenta que posee muchos parámetros que deben estudiarse para poder hacer brillar esta idea que ahora mismo se encuentra en su primera iteración, pero por las razones explicadas anteriormente, sería válido un estudio posterior del algoritmo para verificar si las hipótesis que se proponen se cumplen en la realidad.

### 3. Hibridación y Análisis de Rendimiento

Se ha realizado la hibridación del algoritmo HALO con el algoritmo de Búsqueda Local en de parámetros reales Solis-Wets, de esta manera se obtiene el algoritmo memético HALO  $\times$  Solis-Wets.

En la etapa  $E_G$  se añade una llamada para cada población ( $P_H, P_C, P_F$ ) a Solis-Wets, donde se envía el mejor individuo de cada población para que sea mejorado por dicho método.

Solis-Wets posee  $150 \times$  dimensión evaluaciones de la función objetivo y para compensar eso se reduce el tamaño de la población de 30 individuos a 20, reduciendo el tamaño total de 90 individuos a 60, así balanceando las evaluaciones que realiza la BL y las que realiza el algoritmo HALO propiamente.

#### 3.1. Procedimiento Principal Hibridizado

Se detalla dónde se ha añadido el código adicional, el resto del procedimiento se mantiene igual a su variante genética.

```

procedimiento HaloxSolisWets (entero: dimension)
inicio:

    ...

    hacer
        si(iteraciones==2)entonces
            iteraciones<-0
            si(mejorFitness = pH.mejorIndividuo.fitness)entonces
                si(pC.mediaFitness < pF.mediaFitness) entonces
                    guerra(pH, pF, pC)
                sino
                    guerra(pH, pC, pF)
            fsi
            sino si (mejorFitness = pC.mejorIndividuo.fitness)
                si(pH.mediaFitness < pF.mediaFitness) entonces
                    guerra(pC, pF, pH)
                sino
                    guerra(pC, pH, pF)
            fsi
            sino
                si(pH.mediaFitness < pC.mediaFitness) entonces
                    guerra(pF, pC, pH)
                sino
                    guerra(pF, pH, pC)
            fsi
        fsi

        auxGenes<-pH.genes[pH.mejorIndividuo.indice].obtenerGenes()
        auxFitness<-pH.mejorIndividuo.fitness

        soliswets(auxGenes, auxFitness, 0.2, 150 * dim, -100, 100)

        si(auxFitness < pH.mejorIndividuo.fitness)entonces
            pH.genes[pH.mejorIndividuo.indice].actualizar(auxGenes,auxFitness)
            pH.mejorIndividuo.fitness = auxFitness
        fsi

        auxGenes<-pC.genes[pC.mejorIndividuo.indice].obtenerGenes()
        auxFitness<-pC.mejorIndividuo.fitness

        soliswets(auxGenes, auxFitness, 0.2, 150 * dim, -100, 100)

        si(auxFitness < pC.mejorIndividuo.fitness)entonces
            pC.genes[pC.mejorIndividuo.indice].actualizar(auxGenes,auxFitness)
            pC.mejorIndividuo.fitness = auxFitness
        fsi

        auxGenes<-pF.genes[pF.mejorIndividuo.indice].obtenerGenes()
        auxFitness<-pF.mejorIndividuo.fitness

        soliswets(auxGenes, auxFitness, 0.2, 150 * dim, -100, 100)

        si(auxFitness < pF.mejorIndividuo.fitness)entonces

```

```
pF.genes[pF.mejorIndividuo.indice].actualizar(auxGenes,auxFitness)
pF.mejorIndividuo.fitness = auxFitness
    fsi
    fsi
    iteraciones++
    ...
fproc
```

### 3.2. Análisis de Rendimiento

La adición de la Búsqueda Local, como era de esperarse, ha mejorado el algoritmo. Si se observa la tabla del 100 % de evaluaciones puede observarse que ahora la metaheurística HALO× Solis-Wets ha podido reclamar 7 soluciones de mejor calidad, a diferencia de las 0 que obtuvo en la propuesta original –si bien *Differential Evolution* lleva la ventaja, y de igual manera, en dimensión 30 se aumentó de 10 a 12 soluciones con mejor calidad, esto da confianza de que esta propuesta si bien, como se comentó antes, posee muchos parámetros que habría que controlar es capaz de obtener resultados muy buenos, en este caso con un cambio relativamente simple que es añadir Solis-Wets sin cambiar mucho el resto de la estructura del algoritmo ya ha provisto muchas mejoras.

Nuevamente, vale la pena volver a mencionar que siguiendo la traza se puede ver que el algoritmo si bien avanza, lo hace lentamente, puesto que no se ha detallado un caso donde claramente pueda verse que ha llegado a un óptimo local, puesto que las diferencias en la evaluación de la función objetivo siempre están decreciendo, pero nunca se nota que se forme una meseta donde no hay mejora alguna de la solución.

### 3.2.1. 1 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HxSOLIS	PSO		DE	HxSOLIS	PSO
F01	2.645e+10	3.216e+09	7.208e+09	F01	4.909e+04	2.107e+03	4.175e+09
F02	2.398e+17	4.161e+09	1.000e+00	F02	1.309e+19	7.672e+25	1.000e+00
F03	1.696e+04	3.233e+04	3.167e+04	F03	3.481e+03	5.696e+04	5.453e+04
F04	4.561e+03	9.596e+01	4.702e+02	F04	8.430e+01	8.067e+01	1.183e+03
F05	1.648e+02	3.874e+01	1.013e+02	F05	2.015e+02	1.110e+02	2.170e+02
F06	6.946e+01	8.763e+00	5.398e+01	F06	6.320e+00	7.861e-01	3.694e+01
F07	1.409e+02	6.775e+01	1.764e+02	F07	2.334e+02	1.550e+02	3.596e+02
F08	7.442e+01	4.692e+01	8.991e+01	F08	1.894e+02	1.024e+02	1.745e+02
F09	1.285e+03	2.381e+02	1.504e+03	F09	6.530e+01	1.142e+03	2.842e+03
F10	3.038e+03	1.264e+03	1.951e+03	F10	3.764e+03	3.075e+03	6.938e+03
F11	3.033e+07	3.300e+03	2.552e+03	F11	7.958e+01	9.387e+01	1.207e+03
F12	4.522e+09	9.396e+06	2.570e+08	F12	3.258e+05	1.076e+06	3.586e+08
F13	2.169e+09	7.204e+05	4.722e+06	F13	1.536e+02	4.001e+03	4.508e+07
F14	1.624e+09	7.742e+03	4.792e+03	F14	7.100e+01	9.199e+04	3.059e+05
F15	2.786e+08	9.611e+03	5.283e+04	F15	6.256e+01	1.388e+03	2.737e+05
F16	1.113e+03	4.832e+02	6.786e+02	F16	1.319e+03	1.138e+03	1.568e+03
F17	4.090e+02	1.392e+02	2.967e+02	F17	4.809e+02	6.097e+02	4.725e+02
F18	1.166e+10	1.166e+06	2.861e+07	F18	6.122e+01	1.636e+05	2.170e+06
F19	9.707e+09	4.768e+05	9.718e+05	F19	3.572e+01	7.567e+02	1.260e+06
F20	5.149e+02	1.538e+02	2.913e+02	F20	2.751e+02	4.976e+02	4.621e+02
F21	6.010e+02	2.371e+02	2.791e+02	F21	3.255e+02	3.165e+02	4.113e+02
F22	2.468e+03	4.052e+02	5.520e+02	F22	1.002e+02	1.510e+03	1.027e+03
F23	1.402e+03	3.565e+02	4.265e+02	F23	5.346e+02	4.794e+02	6.404e+02
F24	9.159e+02	3.809e+02	4.233e+02	F24	6.059e+02	7.150e+02	7.095e+02
F25	2.005e+03	5.546e+02	7.664e+02	F25	3.870e+02	3.924e+02	6.864e+02
F26	2.705e+03	9.005e+02	1.273e+03	F26	4.038e+02	1.626e+03	3.369e+03
F27	1.545e+03	4.619e+02	5.433e+02	F27	4.925e+02	5.638e+02	8.068e+02
F28	1.530e+03	8.049e+02	9.516e+02	F28	3.943e+02	3.925e+02	1.105e+03
F29	1.368e+04	3.990e+02	6.152e+02	F29	1.025e+03	9.509e+02	1.409e+03
F30	3.782e+08	1.167e+06	1.184e+07	F30	3.657e+03	1.829e+04	1.359e+07
Best	1	26	3	Best	16	12	2

### 3.2.2. 50 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HxSOLIS	PSO		DE	HxSOLIS	PSO
F01	1.997e-05	2.345e+02	1.847e+08	F01	1.787e+07	2.109e+03	7.702e+09
F02	0.000e+00	1.090e+07	1.000e+00	F02	5.436e+23	1.269e+26	1.000e+00
F03	2.810e-08	5.375e+02	3.460e+03	F03	1.370e+04	9.029e+04	7.122e+04
F04	2.359e-01	1.959e+00	6.360e+01	F04	9.421e+01	9.439e+01	1.663e+03
F05	1.151e+02	1.669e+01	4.730e+01	F05	2.197e+02	1.114e+02	2.600e+02
F06	3.469e+01	1.052e+00	1.361e+01	F06	2.136e+01	1.894e+00	4.667e+01
F07	4.377e+01	2.950e+01	5.702e+01	F07	2.524e+02	1.632e+02	4.096e+02
F08	2.983e+01	1.864e+01	2.967e+01	F08	2.093e+02	1.045e+02	2.156e+02
F09	2.089e+02	2.232e+01	8.356e+01	F09	2.785e+03	1.190e+03	4.100e+03
F10	5.311e+02	6.187e+02	1.323e+03	F10	4.422e+03	3.101e+03	7.431e+03
F11	5.896e+00	2.005e+01	5.042e+01	F11	1.118e+02	1.103e+02	1.557e+03
F12	7.664e+01	8.170e+05	4.645e+06	F12	5.635e+06	2.618e+06	6.936e+08
F13	1.109e+01	2.445e+03	1.227e+04	F13	1.513e+03	8.863e+03	2.357e+08
F14	1.545e+01	6.265e+02	1.643e+02	F14	8.108e+01	2.493e+05	4.771e+05
F15	4.371e-01	9.136e+02	3.228e+03	F15	1.141e+02	2.060e+03	1.265e+06
F16	4.562e+02	2.053e+02	1.850e+02	F16	1.510e+03	1.157e+03	1.859e+03
F17	3.118e+01	3.503e+01	7.631e+01	F17	5.825e+02	6.148e+02	5.510e+02
F18	1.672e-01	3.481e+03	2.653e+04	F18	3.292e+02	3.007e+05	4.171e+06
F19	4.603e-01	2.021e+03	5.864e+03	F19	4.629e+01	1.294e+03	2.330e+06
F20	3.837e+02	1.987e+01	1.100e+02	F20	3.109e+02	5.059e+02	6.190e+02
F21	1.918e+02	1.747e+02	1.421e+02	F21	3.502e+02	3.171e+02	4.487e+02
F22	1.024e+02	1.082e+02	1.025e+02	F22	1.184e+02	1.575e+03	1.354e+03
F23	8.115e+02	3.293e+02	3.371e+02	F23	5.524e+02	4.879e+02	6.911e+02
F24	1.000e+02	2.332e+02	2.041e+02	F24	6.206e+02	7.164e+02	7.470e+02
F25	4.040e+02	4.265e+02	4.536e+02	F25	3.896e+02	3.997e+02	7.749e+02
F26	2.706e+02	4.307e+02	4.175e+02	F26	4.896e+02	1.658e+03	3.733e+03
F27	3.897e+02	4.099e+02	4.200e+02	F27	5.145e+02	5.654e+02	8.850e+02
F28	3.518e+02	4.533e+02	5.235e+02	F28	4.370e+02	4.236e+02	1.288e+03
F29	2.514e+02	3.441e+02	3.508e+02	F29	1.166e+03	9.627e+02	1.609e+03
F30	8.052e+04	2.558e+05	1.485e+06	F30	1.572e+04	2.929e+04	2.024e+07
Best	21	7	2	Best	14	14	2



### 3.2.3. 100 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HxSOLIS	PSO		DE	HxSOLIS	PSO
F01	0.000e+00	2.345e+02	5.255e+07	F01	4.909e+04	2.107e+03	4.175e+09
F02	0.000e+00	6.430e+05	1.000e+00	F02	1.309e+19	7.672e+25	1.000e+00
F03	0.000e+00	1.304e-02	1.989e+03	F03	3.481e+03	5.696e+04	5.453e+04
F04	1.105e-04	1.471e+00	4.684e+01	F04	8.430e+01	8.067e+01	1.183e+03
F05	1.151e+02	1.638e+01	3.212e+01	F05	2.015e+02	1.110e+02	2.170e+02
F06	3.460e+01	3.185e-01	1.001e+01	F06	6.320e+00	7.861e-01	3.694e+01
F07	3.848e+01	2.779e+01	4.275e+01	F07	2.334e+02	1.550e+02	3.596e+02
F08	2.983e+01	1.771e+01	2.203e+01	F08	1.894e+02	1.024e+02	1.745e+02
F09	1.938e+02	1.633e+01	5.686e+01	F09	6.530e+01	1.142e+03	2.842e+03
F10	3.597e+02	5.813e+02	1.077e+03	F10	3.764e+03	3.075e+03	6.938e+03
F11	1.942e-02	1.785e+01	3.843e+01	F11	7.958e+01	9.387e+01	1.207e+03
F12	4.931e+00	5.014e+05	2.517e+06	F12	3.258e+05	1.076e+06	3.586e+08
F13	5.988e+00	2.419e+03	8.409e+03	F13	1.536e+02	4.001e+03	4.508e+07
F14	5.240e-02	6.032e+02	9.993e+01	F14	7.100e+01	9.199e+04	3.059e+05
F15	6.060e-02	8.235e+02	2.066e+03	F15	6.256e+01	1.388e+03	2.737e+05
F16	4.561e+02	2.051e+02	1.415e+02	F16	1.319e+03	1.138e+03	1.568e+03
F17	2.350e+01	3.221e+01	6.497e+01	F17	4.809e+02	6.097e+02	4.725e+02
F18	3.630e-02	3.450e+03	1.484e+04	F18	6.122e+01	1.636e+05	2.170e+06
F19	5.192e-03	2.010e+03	3.220e+03	F19	3.572e+01	7.567e+02	1.260e+06
F20	3.837e+02	1.338e+01	8.444e+01	F20	2.751e+02	4.976e+02	4.621e+02
F21	1.889e+02	1.731e+02	1.320e+02	F21	3.255e+02	3.165e+02	4.113e+02
F22	1.005e+02	1.078e+02	7.740e+01	F22	1.002e+02	1.510e+03	1.027e+03
F23	8.098e+02	3.284e+02	3.304e+02	F23	5.346e+02	4.794e+02	6.404e+02
F24	1.000e+02	2.332e+02	1.810e+02	F24	6.059e+02	7.150e+02	7.095e+02
F25	4.040e+02	4.192e+02	4.478e+02	F25	3.870e+02	3.924e+02	6.864e+02
F26	2.706e+02	4.160e+02	3.729e+02	F26	4.038e+02	1.626e+03	3.369e+03
F27	3.897e+02	4.098e+02	4.134e+02	F27	4.925e+02	5.638e+02	8.068e+02
F28	3.517e+02	4.284e+02	4.698e+02	F28	3.943e+02	3.925e+02	1.105e+03
F29	2.375e+02	3.382e+02	3.193e+02	F29	1.025e+03	9.509e+02	1.409e+03
F30	8.051e+04	1.764e+05	6.352e+05	F30	3.657e+03	1.829e+04	1.359e+07
Best	20	7	3	Best	16	12	2

## 4. Propuestas de Mejora de la Metaheurística

Como se ha mencionado anteriormente en los análisis de HALO, esta posee una cantidad de parámetros los cuales pueden afectar la calidad de la solución obtenida; pero por lo general se puede pensar en unas cuantas mejoras y/o estudios que sería válido realizar para el algoritmo; en primer lugar, variar la población puede tener efectos sobre la calidad puesto que a menor cantidad de individuos se gastan menos evaluaciones por cada iteración lo que permitiría que convergiera más cerca de la función objetivo.

Se probó cambiar la población a 16 individuos, haciendo un total de 48, los resultados indican que efectivamente, realizar este cambio mejora a un nivel cercano pero no superior que el de realizar la hibridación con la Búsqueda Local.

### 100 % de Evaluaciones

Dimensión 10				Dimensión 30			
	DE	HALO	PSO		DE	HxSOLIS	PSO
F01	0.000e+00	1.607e+06	5.255e+07	F01	4.909e+04	7.201e+06	4.175e+09
F02	0.000e+00	3.904e+05	1.000e+00	F02	1.309e+19	1.951e+24	1.000e+00
F03	0.000e+00	9.333e+03	1.989e+03	F03	3.481e+03	1.126e+05	5.453e+04
F04	1.105e-04	7.155e+00	4.684e+01	F04	8.430e+01	1.464e+02	1.183e+03
F05	1.151e+02	1.913e+01	3.212e+01	F05	2.015e+02	1.227e+02	2.170e+02
F06	3.460e+01	8.601e-01	1.001e+01	F06	6.320e+00	1.229e+00	3.694e+01
F07	3.848e+01	3.749e+01	4.275e+01	F07	2.334e+02	1.773e+02	3.596e+02
F08	2.983e+01	1.337e+01	2.203e+01	F08	1.894e+02	1.095e+02	1.745e+02
F09	1.938e+02	3.351e+01	5.686e+01	F09	6.530e+01	1.585e+03	2.842e+03
F10	3.597e+02	6.072e+02	1.077e+03	F10	3.764e+03	3.174e+03	6.938e+03
F11	1.942e-02	3.639e+01	3.843e+01	F11	7.958e+01	1.465e+03	1.207e+03
F12	4.931e+00	5.902e+05	2.517e+06	F12	3.258e+05	8.959e+06	3.586e+08
F13	5.988e+00	5.702e+03	8.409e+03	F13	1.536e+02	1.950e+06	4.508e+07
F14	5.240e-02	1.026e+03	9.993e+01	F14	7.100e+01	6.391e+05	3.059e+05
F15	6.060e-02	1.229e+03	2.066e+03	F15	6.256e+01	3.108e+03	2.737e+05
F16	4.561e+02	1.999e+02	1.415e+02	F16	1.319e+03	1.132e+03	1.568e+03
F17	2.350e+01	4.573e+01	6.497e+01	F17	4.809e+02	5.806e+02	4.725e+02
F18	3.630e-02	3.054e+03	1.484e+04	F18	6.122e+01	1.301e+06	2.170e+06
F19	5.192e-03	2.832e+03	3.220e+03	F19	3.572e+01	5.726e+03	1.260e+06
F20	3.837e+02	1.827e+01	8.444e+01	F20	2.751e+02	4.440e+02	4.621e+02
F21	1.889e+02	2.111e+02	1.320e+02	F21	3.255e+02	3.168e+02	4.113e+02
F22	1.005e+02	1.087e+02	7.740e+01	F22	1.002e+02	2.411e+03	1.027e+03
F23	8.098e+02	3.356e+02	3.304e+02	F23	5.346e+02	5.126e+02	6.404e+02
F24	1.000e+02	2.931e+02	1.810e+02	F24	6.059e+02	7.210e+02	7.095e+02
F25	4.040e+02	4.405e+02	4.478e+02	F25	3.870e+02	4.519e+02	6.864e+02
F26	2.706e+02	5.609e+02	3.729e+02	F26	4.038e+02	1.622e+03	3.369e+03
F27	3.897e+02	4.167e+02	4.134e+02	F27	4.925e+02	5.633e+02	8.068e+02
F28	3.517e+02	4.841e+02	4.698e+02	F28	3.943e+02	5.095e+02	1.105e+03
F29	2.375e+02	3.353e+02	3.193e+02	F29	1.025e+03	9.735e+02	1.409e+03
F30	8.051e+04	1.247e+06	6.352e+05	F30	3.657e+03	2.455e+05	1.359e+07
Best	20	6	4	Best	19	9	2

Por otro lado, se pueden empezar a variar los operadores de selección, cruce y mutación, esto ya requeriría un estudio de mayor amplitud por ser algoritmos genéticos, no se puede confiar en el resultado de una única ejecución; también se podrían variar los operadores nuevos añadidos en la etapa  $E_G$ , por ejemplo, hacer que cada población tenga un operador único que cuando esta sea la que tiene la solución de mejor calidad, se apliquen diferentes efectos a las otras poblaciones.