



UNIVERSIDAD
DE GRANADA

ETSIIT

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación



Metaheurística HALO

Lugli, Valentino Glauco

Junio 2021

Inspiración



- La metaheurística surge como inspiración 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 de un super soldado llamado John-117 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, así mismo luego de caer en un mundo anillo, deben también combatir con una especie parasitaria denominada **The Flood** que amenaza la vida de la galaxia

Intuición



- Es fácil ver que en el juego se tienen tres grandes poblaciones las cuales están en conflicto entre sí por el destino de la galaxia, esta es la base para el algoritmo.
- Se definen tres poblaciones inicialmente disjuntas:
 - P_H es la **humanidad**, se puede pensar de ella como un balance genético entre explotación y exploración.
 - P_C es **El Pacto/The Covenant**, son adeptos religiosos por lo que no toleran quienes actúan de forma no ortodoxa, esto se puede analogar a que es una población que explota más que explora.
 - P_F es **The Flood**, siendo parásitos que toman el control de cualquier otra forma viviente, se puede analogar a que esta población explora más que explota.
- De esta manera se definen poblaciones que atraviesan el espacio de soluciones de maneras diferentes

Funcionamiento



- Dadas las tres poblaciones, se posee un bucle principal en el cual se tienen dos etapas.
 - **Etapas de Paz E_p** , las poblaciones evolucionan sin interactuar entre sí por una cantidad determinada de evaluaciones y/o generaciones.
 - **Etapas de Guerra E_g** , donde las poblaciones se enfrentan entre sí. La etapa de guerra sucede cada cierto número de iteraciones (no de evaluaciones de la función objetivo).
- E_p se encarga de que las poblaciones avancen a su ritmo por el espacio de soluciones, al tener diferentes maneras de atravesarlo, algunas quedarán mejor posicionadas que otras.
- E_g se encarga de compartir el material genético prometedor con la población que posee el mejor individuo, mientras que las poblaciones perdedoras se ven forzadas a regenerar nuevos individuos.

Funcionamiento Etapa E_p



- Las poblaciones evolucionan siguiendo un modelo genético generacional con elitismo, además de una selección por torneo entre los padres.
- Las poblaciones recurren a diferentes operadores de cruce y de mutación para moverse en el espacio de soluciones de la manera que queremos.
 - P_H hace uso del Cruce BLX- α y de la Mutación No Uniforme, ambos operadores mantienen el balance entre exploración y explotación.
 - P_C hace uso del Cruce Aritmético y de la Mutación de Mühlenbein para mantener más explotación que exploración del espacio.
 - P_F hace uso del Cruce Lineal de 3 hijos, tomando los 2 mejores y la Mutación Aleatoria para propiciar más la exploración del espacio.

Funcionamiento Etapa E_G



- Una vez transcurrido un cierto número de iteraciones, se entra en la etapa E_G en la que se define el operador **Guerra**.
- Dependiendo de la población que posea el individuo con el **mejor** fitness, con el operador Guerra se toman las otras 2 poblaciones ordenadas por el fitness **promedio** y se empiezan a realizar cambios entre ellas.
- La población con el **segundo mejor** fitness promedio puede estar sujeta a tener cambios en 25% a 50% de su población, el número es elegido estocásticamente y no necesariamente un individuo será eliminado.
- Si un individuo i de la población ganadora tiene un peor fitness que otro individuo j de la población perdedora con segundo mejor fitness, se copia la información de j a i , pero i se mantiene vivo.
- En caso contrario, i “muere” y se genera un nuevo individuo por medio de un pase del Cruce Plano entre un individuo seleccionado aleatoriamente de esa población y el mejor individuo de la misma.
- Este proceso se repite con la población que posee el peor fitness de todos, aunque esta población está sujeta tener cambios entre 0% y el 50% de sus individuos.

Pseudocódigo



```
procedimiento Halo (entero: dimension)
inicio:
    inicializarPoblaciones(pH, pC, pF)
    evaluarPoblaciones(pH, pC, pF)

    hacer
        // Etapa eG
        si(iteraciones==2)entonces
            iteraciones<-0
            guerra(pH, pC, pF)
        fsi
        iteraciones++
        // Etapa eP
        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
```


Resultados de CEC2017



- Se realizó la comparación del algoritmo con **DE** (*Differential Evolution*) y **PSO** (*Particle Swarm Optimization*) debido a que comparte características con estos algoritmos establecidos.
- Los resultados de Dimensión 10 dieron como ganador a **DE**, mientras que **HALO** quedó rezagado obteniendo resultados ligeramente peores que **PSO**
- Los resultados de Dimensión 30 dieron a relucir más el algoritmo, mientras que **DE** obtuvo 19 mejores evaluaciones ahora **HALO** obtuvo 10, quedándose más cerca que **PSO** que obtuvo a penas 1 evaluación óptima.
- La traza indica que el algoritmo está limitado por su convergencia lenta, puesto que no se observaron estancamientos que podrían deberse a óptimos locales, más que se gasta más evaluaciones al tener 3 poblaciones y esto ralentiza la búsqueda en el espacio.