

MASTER MIND

REALIZADO POR:

CAMPOS ROMERO, MIGUEL

DE TOVAR VÁZQUEZ, ERNESTO

INTRODUCCION

La entrega presenta dos documentos, AlgoritmoFinal.py y MasterMindSolBinaria.py acompañados de dos archivos: un primero, Prueba1.pynb con el pertinente test realizado a AlgoritmoFinal.py y un segundo, Prueba2.pynb, con el test que concierne a MasterMindSolBinaria.py

OPERADORES DE CRUCE

Se ha implementado la estrategia de “cruce parcial”, visto en teoría.

OPERADORES DE MUTACION

Se ha implementado la mutación uniforme.

DISEÑO DEL ALGORITMO GENÉTICO

Como se mencionó, se presenta dos soluciones distintas:

- **AlgoritmoFinal.py:**
 - Los genes son los números del **0 al 9 con repetición**.
 - Los individuos son listas de **longitud 16**.
 - Cada individuo representa la solución en la que el individuo i-ésimo es evaluado con respecto al individuo i-ésimo de cada una de las pistas.
 - El algoritmo realiza una **permutación con reemplazamiento** sobre los individuos seleccionados mediante **torneo**, genera nuevos individuos evolucionados a partir de la selección y los introduce en la población mediante la técnica de **reemplazamiento aleatorio**, dando lugar a la nueva generación.
 - Si AG encuentra la solución global, acaba.
 - Si AG encuentra a un máximo local se realiza una **búsqueda local**.
 - Una **búsqueda local**, consiste en una estrategia de optimización en la que, con centro ‘máximo local ofrecido por AG’, se

construye un espacio de vecinos virtuales y se devuelve al mejor (en caso de que no lo hubiese devolvería al máximo local).

- Las pruebas se hicieron con poblaciones de miles de individuos y dejando el algoritmo funcionar durante varias horas. Por ello, para no hacer este documento inmenso se ha optado por entregar las pruebas realizadas en el documento **Prueba1.pynb**.

- **MasterMindSolBinaria.py**

- Los genes son **0** y **1**.
- Los individuos son cromosomas de **longitud 16**.
- Cada individuo representa la solución en la que el individuo i-ésimo es evaluado con respecto al individuo i-ésimo **en binario** de cada una de las pistas.
- Para el funcionamiento del algoritmo, se han implementado una serie de traductores de **decimal a binario** y de **binario a decimal**.
- El algoritmo realiza una evolución genética de una serie de individuos seleccionados mediante **torneo** y los introduce en la población mediante la técnica de **reemplazamiento aleatorio** dando lugar a la nueva generación.
- Las pruebas se hicieron con poblaciones de miles de individuos y dejando el algoritmo funcionar durante varias horas. Por ello, para no hacer este documento inmenso se ha optado por entregar las pruebas realizadas en el documento **Prueba2.pynb**

EVALUACION DEL PROBLEMA

Ambas soluciones usan la misma función fitness, la cual evalúa el **número de dígitos dispares** de un candidato con respecto a todas las pistas, **premiando la igualdad con pistas buenas y penalizando igualdad con pistas malas**.

Para implementar esto, se ha optado por una **suma binomial**. De esta manera la fitness tan solo trabajará con valores de mismo signo y podremos estudiar qué tan bueno son los resultados en cada generación de forma más segura.

El problema se trata, por tanto, de un problema de minimización donde los individuos más cercanos al 0 son mejores, siendo el fitness == 0 la solución.

