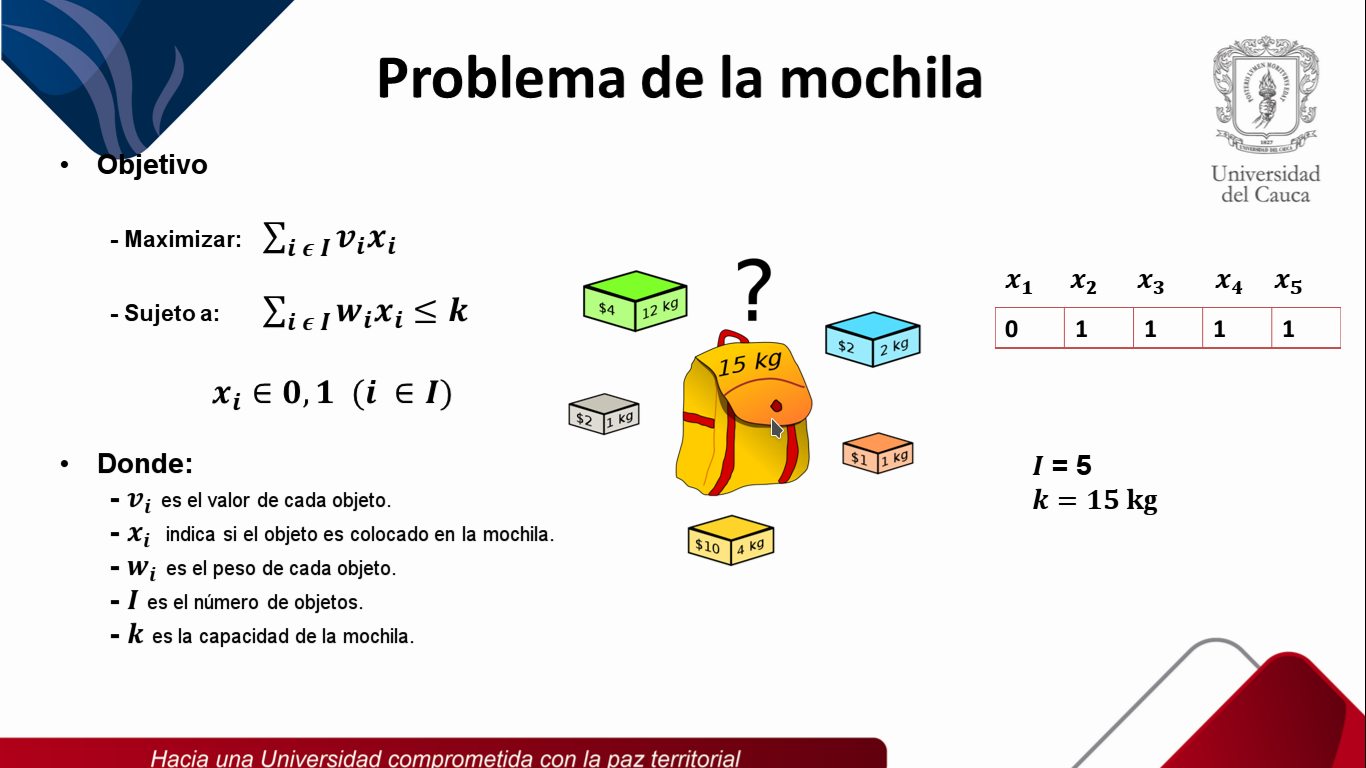
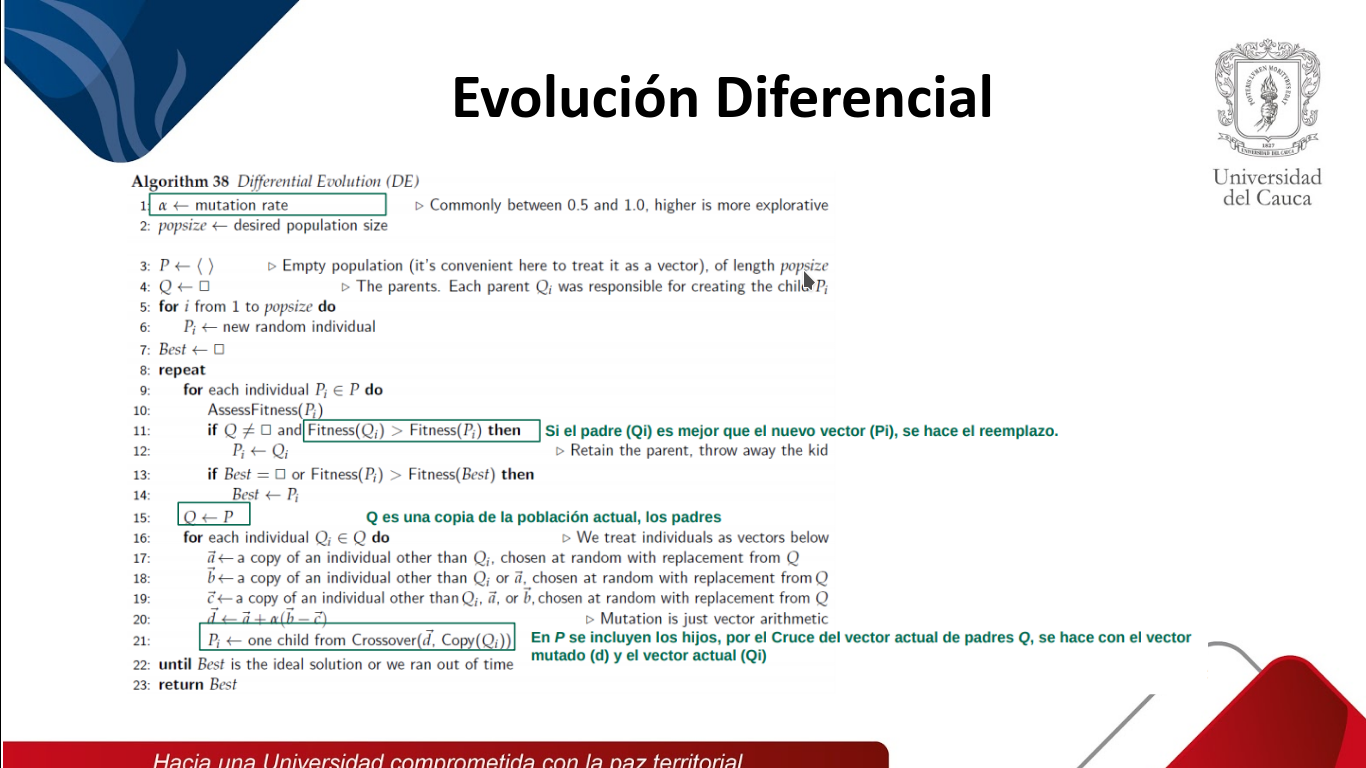
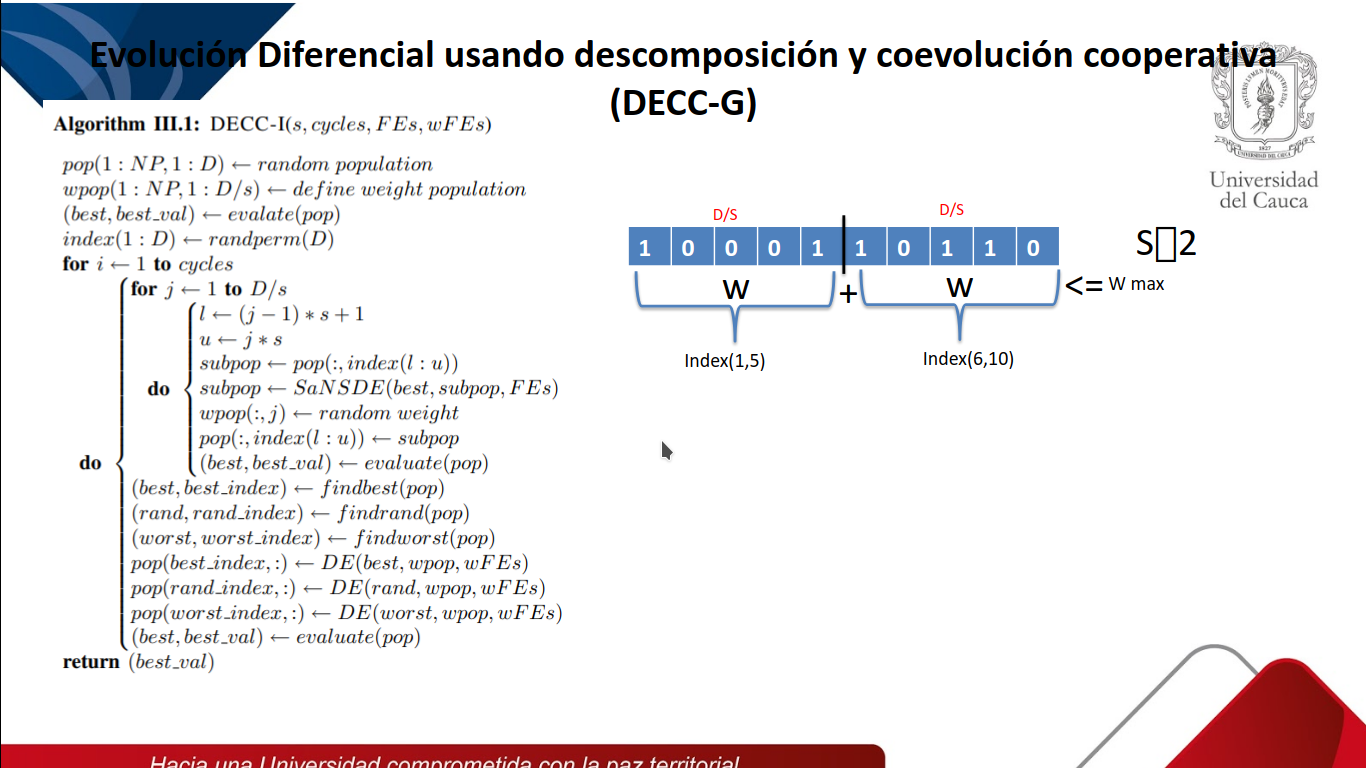
**Problema de la Mochila – Búsqueda Tabú**

****

**Algoritmos base**

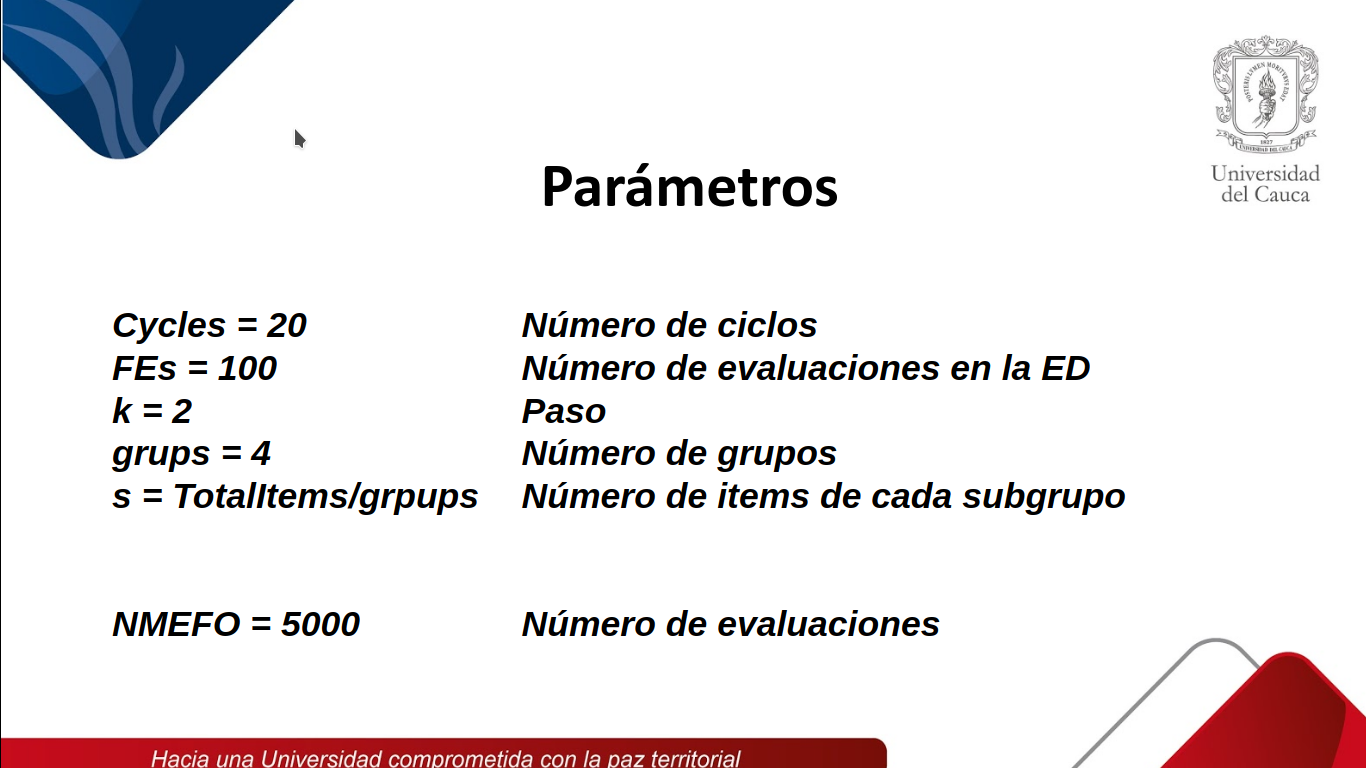
****

****

**Análisis**

Se implementó el ***EDCC-G EN*** *base a [1] y [2].*

**Parámetros:** Los parámetros de los algoritmos se escogen después de hacer varias pruebas manuales.

****

El número de ciclos **(cycles)** de escoge de 20, un número grande para este parámetro ya que el algoritmo termina en primera parte por el número de evaluaciones.

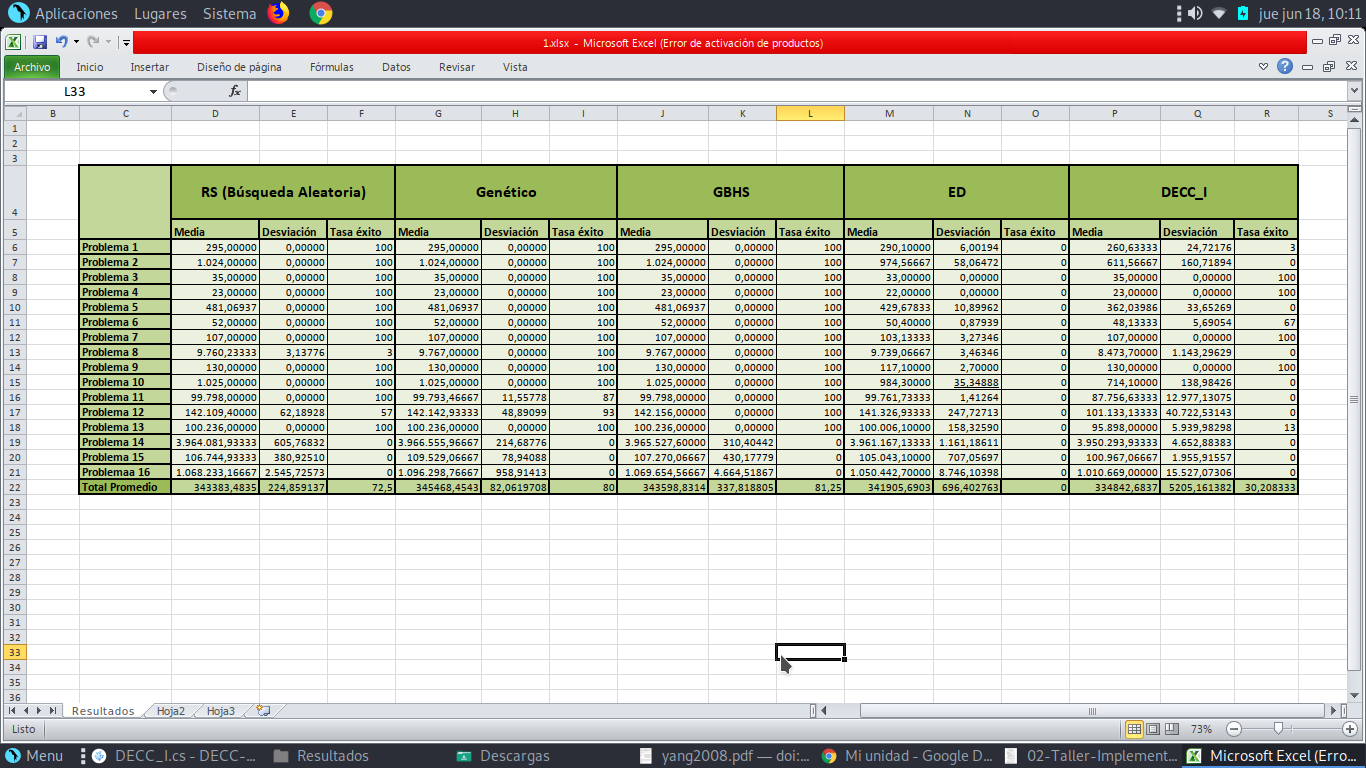
Las **FEs** que son el número de evaluaciones en el algoritmo de **ED** se escoge de 100 para que toda la población compita por lo menos una vez con un padre. Cabe aclarar que se está trabajando con una población de 100

No se se escoge un valor mayor de **FEs** ya que se tendría que trabajar con un número mayor de evaluaciones para tener ciclos completos del algoritmo.

El número **k** es el paso del algoritmo y se utiliza en la mutación diferencial que se está utilizando. Se escoge un valor de 2 ya que fue el que ayudó a arrojar mejores resultados.

El algoritmo se puede iniciar con un número de grupos que define el número de subgrupos en los que se partirá cada solución o asignando a **s** directamente el número de items que tendrá cada grupo, este último es mejor en problemas grandes.

**Resultados:**



Los resultados no fueron los esperados, ya que se tuvo problemas en la implementación de el algoritmo de **ED**. La **ED** no nos arrojó resultados buenos ya que a pesar de arrojar resultados cercanos a la solución su tasa de éxito fue de 0%.

EL **DECCI** usa la **ED** por tal motivo se identifica como la culpable de los resultados.

**Conclusiones:**

* Como se espera el algoritmo Evolución Diferencial usando descomposición y coevolución cooperativa (DECC-G) tiene mejores resultados en relación a los otros algoritmos en problemas de que contienen muchas más dimensiones.
* Los algoritmos poblacionales tiene un mejor rendimiento que los algoritmos de estado simple.
* El tiempo de ejecución de los algoritmos de ED y DECC-G es mucho mejor que los algoritmos restantes..

**Referencias**

* [1] Z. Yang, K. Tang, X. Yao, “Large scale evolutionary optimization using cooperative coevolution”, Information Sciences, 2008, pp 2985-2999
* [2] Z. Yang, K. Tang, X. Yao, “Differential Evolution for High-Dimensional Function Optimization”, Septiembre de 2007.