

INTELIGENCIA DE NEGOCIOS

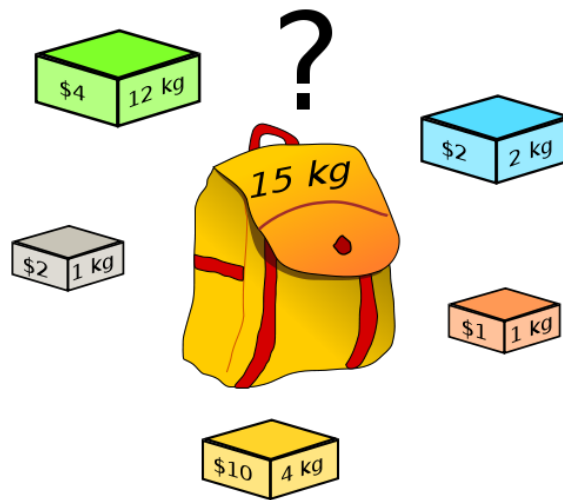
Dr. Mario Garza Fabre

Mayo – Agosto 2018

PROYECTO UNIDAD 1: Algoritmo genético para resolver el problema de la mochila

El problema de la mochila (knapsack problem) es un problema de optimización combinatoria, es decir, donde se busca la mejor solución entre un conjunto finito de posibles soluciones. Modela una situación análoga a la de llenar una mochila, incapaz de soportar más de un peso determinado, con todo o parte de un conjunto de objetos, cada uno con un peso (o costo) y valor específicos. La meta es que los objetos colocados en la mochila **maximicen el valor total sin exceder la capacidad de la mochila**.

(adaptado de: https://es.wikipedia.org/wiki/Problema_de_la_mochila)



Este problema encuentra aplicaciones importantes en diversas áreas, y algoritmos para resolver este problema pueden representar el corazón de un sistema de apoyo a la toma de decisiones.

En este proyecto el alumno diseñará e implementará un Algoritmo Genético (AG) para resolver el problema de la mochila y evaluará su desempeño con respecto a algunos enfoques menos sofisticados (generación aleatoria de soluciones y algoritmos voraces). Considere las siguientes especificaciones.

ALGORITMOS A IMPLEMENTAR

Se implementarán un total de 6 enfoques diferentes con la finalidad de investigar y evaluar las ventajas/desventajas de cada uno de ellos. Los primeros 4 enfoques serán métodos simples de solución que se utilizarán como referencia para comparar los resultados de 2 versiones diferentes del AG.

- 1) **Selección aleatoria (RND).** Este enfoque considerará todos y cada uno de los objetos en orden aleatorio y en cada paso seleccionará o descartará el objeto dependiendo de si existe o no capacidad suficiente en la mochila (enfoque no informado).
- 2) **Algoritmo voraz 1 (GR1).** Modificación del enfoque RND donde los objetos serán considerados en orden decreciente de acuerdo con su valor (enfoque ambicioso).
- 3) **Algoritmo voraz 2 (GR2).** Modificación del enfoque RND donde los objetos serán considerados en orden ascendente de acuerdo con su peso o costo (enfoque cauteloso, o tacaño).

4) **Algoritmo voraz 3 (GR3).** Modificación del enfoque RND donde los objetos serán considerados en orden decreciente de acuerdo con su relación valor/costo (enfoque astuto).

5) **Algoritmo genético 1 (AG1).** Se implementará un algoritmo genético con las siguientes características:

- **Representación:** Cada individuo (solución) se representará mediante una cadena binaria.
- **Inicialización (creación de la población inicial):** La población inicial será generada de manera aleatoria (cadenas binarias en las que el valor de cada bit se decide aleatoriamente). Debe asegurarse de que no existan individuos (cadenas binarias) duplicados en la población.
- **Selección para apareamiento/variación:** Se seleccionarán de manera aleatoria (sin reemplazo) los individuos que formarán las parejas que serán la entrada para el operador de cruza. Esto permitirá que cada individuo sea utilizado como fuente de material genético exactamente una vez.
- **Operador de cruza:** El operador de cruza recibirá como entrada 2 individuos padre y generará como salida 2 individuos hijo. Este operador se aplica con una probabilidad p_c . Se recomienda usar un valor de p_c que se encuentre en el rango $[0.8, 1.0]$. Se asignará a cada equipo de trabajo uno de los siguientes tres operadores de cruza: cruza de un punto, cruza de dos puntos, o cruza uniforme.
- **Operador de mutación:** Este operador se aplicará a cada uno de los nuevos individuos creados por el operador de cruza. Se implementará el operador de mutación uniforme. Este operador modificará cada bit de la cadena binaria con una probabilidad de $p_m = 1/n$, donde n es la longitud de la cadena binaria (en este caso, número de objetos o tamaño del problema).
- **Selección para supervivencia (estrategia de reemplazo):** Una vez que se ha generado una nueva población de individuos hijo a través de la aplicación de los operadores genéticos (cruza y mutación), se deberá decidir qué individuos padre y qué individuos hijo sobrevivirán de una generación a otra. Se asignará a cada equipo de trabajo una de las siguientes dos estrategias:
 - **Selección “+”:** Todos los individuos padre y todos los individuos hijo competirán entre ellos por un lugar en la nueva población. Es decir, de la unión de las poblaciones de padres e hijos se seleccionará la mejor mitad de los individuos (con base en su aptitud). Deberá asegurarse de que la nueva población se encuentre libre de duplicados.
 - **Selección “,” con elitismo:** Todos los individuos hijo sobreviven, excepto uno de ellos que será reemplazado por el mejor individuo padre. El individuo hijo a reemplazar puede ser el que presente menor aptitud (o puede ser seleccionado de manera aleatoria). Deberá asegurarse de que la nueva población se encuentre libre de duplicados. En este caso, reemplazar los individuos hijo duplicados por individuos padres de buena calidad (que no sean duplicados).
- **Criterio de terminación:** El criterio de terminación estará dado por un número máximo de generaciones indicado por el usuario.
- **Aptitud de los individuos (evaluación de soluciones):** El cálculo de la aptitud de los individuos es un proceso muy importante, ya que la aptitud debe reflejar la calidad (y factibilidad!) de las soluciones que representan. Recuerden que los AGs están inspirados en la “supervivencia del más

apto”. La aptitud es un criterio a maximizarse, es decir, un mayor valor de aptitud debe corresponder con una mejor calidad (y factibilidad!) de solución.

Ya que nos interesa maximizar el valor total de la mochila, la aptitud de un individuo estará definida por la **suma de los valores** de todos los objetos contenidos en ella. Sin embargo, es importante considerar que no todas las soluciones posibles son factibles. Existen individuos que no son factibles, ya que representan selecciones de objetos que exceden la capacidad de la mochila. En este proyecto, **todas las soluciones no factibles tendrán una aptitud de 0**. Esta sencilla estrategia se conoce como “pena de muerte” en la literatura.

- 6) **Algoritmo genético 2 (AG2)**. Modificación del enfoque AG1 donde se inserta en la población inicial una solución generada utilizando un método voraz (GR1, GR2 o GR3, preferentemente el que muestre el mejor desempeño global). **OPCIONALMENTE**: El alumno tendrá también la oportunidad de utilizar su creatividad o habilidades de investigación para mejorar el diseño del AG, sus operadores y/o sus parámetros (con excepción de tamaño de población y criterio de terminación) con la finalidad de obtener un mejor desempeño y, por lo tanto, incrementar su potencial para ganar la competencia (ver detalles más adelante).

PROBLEMAS DE PRUEBA

Se utilizarán un total de 24 instancias de prueba. Cada una de estas instancias se encuentra en un archivo de texto independiente con el siguiente formato:

```

N
C
Z
valor[1]    costo[1]
valor[2]    costo[2]
...
valor[N]    costo[N]

```

donde N es el tamaño del problema, C es la capacidad de la mochila y Z es el valor total de la solución óptima (o mejor solución conocida). Los detalles de las 24 instancias de prueba se muestran en la siguiente tabla:

#	Problema	N	C	Z
1	KP20_01	20	970	1428
2	KP20_02	20	13921	8242
3	KP20_03	20	1150	1360
4	KP20_04	20	13921	14039
5	KP20_05	20	873	1716
6	KP20_06	20	9286	10945
7	KP20_07	20	994	1794
8	KP20_08	20	6813	10113
9	KP20_09	20	998	1002
10	KP20_10	20	9240	9258

11	KP20_11	20	946	2296
12	KP20_12	20	7175	12319
13	KP50_01	50	970	1428
14	KP50_02	50	22924	14839
15	KP50_03	50	2991	2971
16	KP50_04	50	19583	20793
17	KP50_05	50	3064	3766
18	KP50_06	50	21690	25084
19	KP50_07	50	2996	6396
20	KP50_08	50	16892	25492
21	KP50_09	50	4497	4521
22	KP50_10	50	18550	18600
23	KP50_11	50	1001	3908
24	KP50_12	50	16243	30242

EXPERIMENTACIÓN

- Los enfoques **RND**, **AG1** y **AG2** involucran componentes estocásticos, es decir, son enfoques **no deterministas**. Por lo tanto, para cada uno de los problemas de prueba ejecutará estos algoritmos **10 veces** utilizando cada vez una semilla diferente para inicializar el generador de números aleatorios (de lo contrario, todas las ejecuciones tendrían el mismo comportamiento).
- Los enfoques **GR1**, **GR2** y **GR3** son **deterministas**, por lo que sólo **se ejecutarán una vez**.
- Con la finalidad de realizar una competencia justa entre algoritmos y equipos de trabajo (ver detalles más adelante), es necesario que en todos los casos los algoritmos **AG1** y **AG2** utilicen la misma cantidad de esfuerzo computacional (número de soluciones evaluadas). Para lograr lo anterior, considerará siempre los siguientes ajustes:
 - Tamaño de la población: **100 individuos**
 - Criterio de terminación: **100 generaciones**
- Será necesario que se almacenen los archivos de salida/resultados de cada ejecución de los diferentes algoritmos, ya que éstos resultados se emplearán para calcular estadísticas y generar el reporte solicitado (ver detalles más adelante).

REPORTE DE RESULTADOS

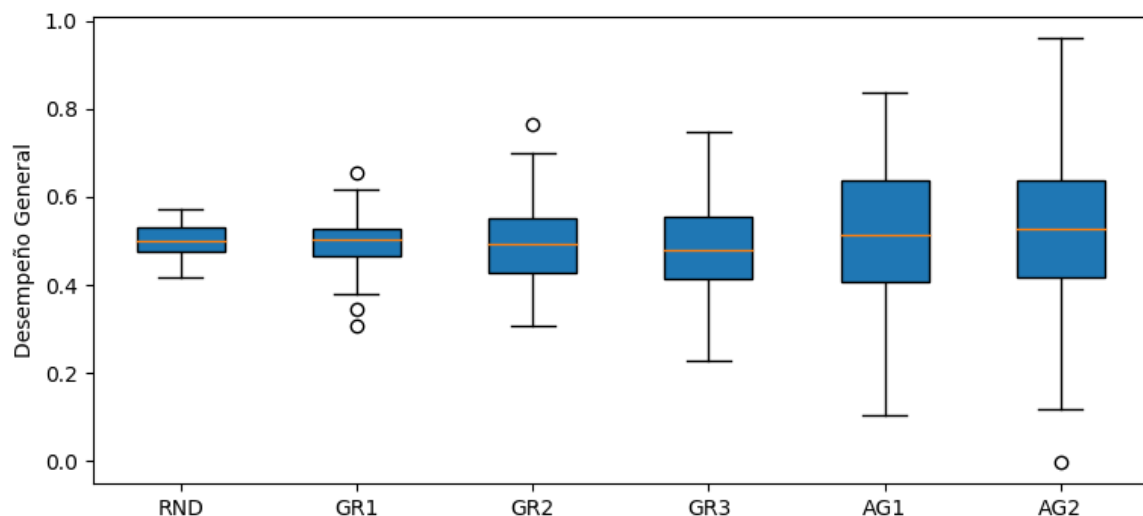
Se entregará un reporte breve con el siguiente contenido:

1. **Portada.** Importante incluir los nombres de los integrantes del equipo de trabajo.
2. **Tabla de resultados.** Para cada problema de prueba, se mostrará la media aritmética y la desviación estándar del valor total de la mejor solución encontrada por cada algoritmo en sus diferentes

ejecuciones. Para los enfoques GR1, GR2 y GR3 no es necesario calcular la desviación estándar. Para cada problema, sombrear la celda y/o resaltar el texto en negrillas para el mejor resultado obtenido.

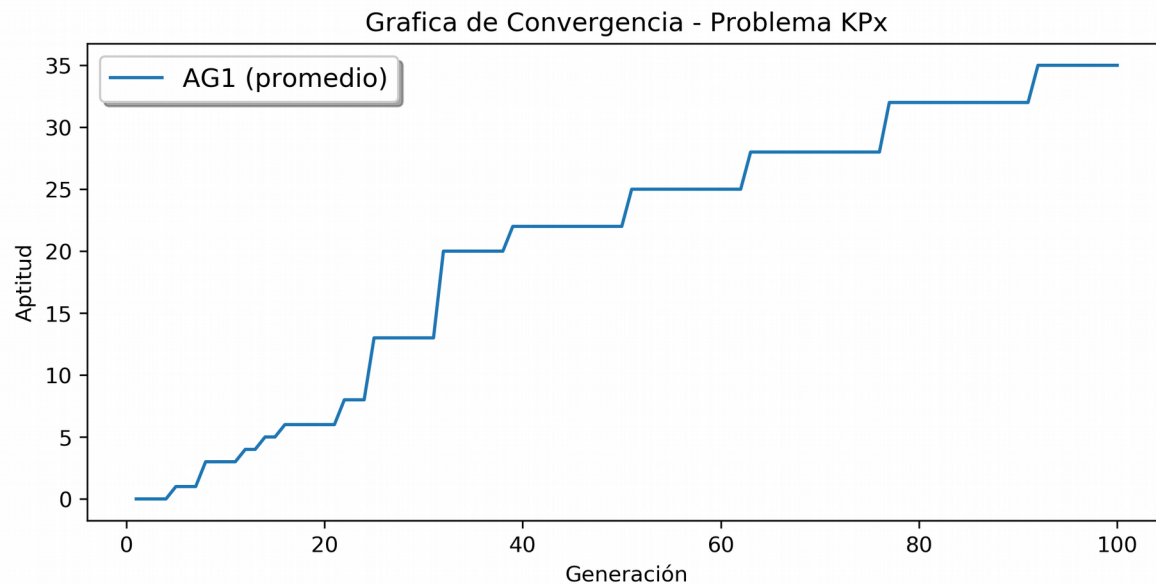
Problema	RND	GR1	GR2	GR3	AG1	AG2
KP20_01	99.8 ± 3.2	120	102	140.3	400.2 ± 11.3	436.4 ± 2.2
KP20_02						
...						
KP50_12						

3. **Gráfica de estadísticas.** Generar una gráfica de cajas (boxplot) para ilustrar el desempeño general de cada método en los diferentes problemas. **IMPORTANTE:** los valores de los resultados se encuentran en un rango diferente para cada problema. Deberá **normalizar** los resultados para que se encuentren en el mismo rango. Para cada método, sólo utilice el promedio (normalizado) de cada problema, de manera que el diagrama de caja se generará usando tantos valores como problemas se consideren.



4. **Gráfica de convergencia.** Generar una gráfica de convergencia donde se muestre el comportamiento de los algoritmos AG1 y AG2. Sólo generará 1 gráfica, en donde se mostrarán las siguientes 4 curvas de convergencia (usar diferentes colores, incluir leyenda):

1. AG1 (promedio). Aptitud promedio de la población (mejor ejecución, problema KP50_12).
2. AG1 (mejor). Aptitud más alta de la población (mejor ejecución, problema KP50_12).
3. AG2 (promedio). Aptitud promedio de la población (mejor ejecución, problema KP50_12).
4. AG2 (mejor). Aptitud más alta de la población (mejor ejecución, problema KP50_12).



COMPETENCIA

Se realizará una competencia en la que cada equipo de trabajo participará utilizando el mejor de sus algoritmos. Se considerará un problema nuevo (diferente a los que aquí se mencionan) y cada equipo tendrá 3 intentos para resolver el problema. La puntuación de cada equipo será la mejor obtenida en estos 3 intentos. El equipo con la mejor puntuación recibirá **10/100 puntos adicionales**, y cada integrante del equipo podrá utilizar estos puntos en la unidad de su preferencia.

EVALUACIÓN

El 100 % del valor de este proyecto será distribuido y evaluado de la siguiente manera:

RND	5 %
GR1	5 %
GR2	5 %
GR3	5 %
AG1	50 %
AG2	10 %
Reporte	20 %
Total	100 %

Utilice el lenguaje de programación de su elección. **Entregar:** Un único archivo (.zip o tar.gz) incluyendo código fuente y un documento o archivo de texto donde se proporcionen las instrucciones de compilación (si aplica) y funcionamiento del programa, así como el reporte de resultados.

FECHA Y HORA LÍMITE DE ENTREGA

La fecha y hora límite de entrega será: **Viernes 25 de mayo de 2018, 23:55 horas.**

NOTA: Recuerden que la puntuación máxima que puede obtener un proyecto se reducirá 10 % por cada día hábil de retraso en su entrega.