**“Usando computación evolutiva para asignación de ítems a bins”**

Autor/Autores:

Cespedes Chaupis, Roberto

Chacón Mejía, José

Guzman Quintana, Deby

Hinostroza Cárdenas, Andrés

**Resumen-**

El presente trabajo aplicativo, consiste en el desarrollo, implementación y análisis de un algoritmo genético para asignar un conjunto de ítems de diferentes tamaños a un conjunto de Bins (también conocido como Bin-packing Problem, BPP). Este es un problema de optimización combinatoria bien conocido que consiste en empaquetar ortogonalmente un conjunto dado de ítems en un número bins rectangulares. En este artículo comparamos un algoritmo genético con 2 tipos de cruzamiento (one point crossover y uniform crossover) y 2 tipos de mutación (Flip y Multiflip). Todas las implementaciones están escritas en Python, se ejecutan utilizando el entorno Google Colab. Todas las implementaciones se probaron utilizando diferentes cantidades de ítems y bins.

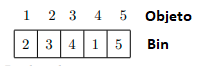
——————————◆——————————

1. **Introducción**

* Descripción del problema: Se nos da un conjunto de ítems, cada uno con un peso (o tamaño) dado (wi> 0). Los cuales se requieren asignar en bins (contenedores) de una capacidad dada C (C> wi) de modo que se minimice la suma total de los ítems que no se logran asignar, y se minimice también la suma total de áreas no utilizadas en los bins. El término bin es un nombre genérico que según el contexto puede fungir de: contenedor en el transporte, estación de trabajo en una línea de ensamblaje, un espacio en el tiempo en una programación, un área de superficie en el trabajo con metales. Como se observa, las aplicaciones de este problema son variadas y omnipresentes.
* Haciendo uso de un Algoritmo Genético (AG) y operadores genéticos tales como mutación y cruzamiento llegaremos a la solución óptima del problema planteado.

1. **Metodología**
   * **Técnicas base (baselines):**

* Como técnica base estamos considerando un algoritmo genético utilizado para el problema de bin packing, en el cual se busca minimizar la cantidad de bins. Haciendo uso de un ratio de cruzamiento del 100% y un ratio de mutación del 70% se observa que una solución mínima al problema se encuentra dentro de 1200 generaciones.
  + **Enfoque(s) propuestos:**
* El problema tiene un enfoque mono objetivo, eligiendo 10 individuos.



* Se diseñó la representación del cromosoma de modo que los genes tengan un valor entero correspondiente al número de bin al que ha sido asignado. El índice de cada gen representa un objeto del pool de items.
* Se realizarán las ejecuciones del AG realizando 26 combinaciones de los operadores de cruzamiento , mutación con el fin de encontrar la mejor agrupación de operadores.
* Dentro de la función de fitness, se añadió un hard constraint para penalizar soluciones en las que la suma total de pesos de items asignados a un bin X sobrepase la capacidad límite.

1. **Experimentación y Resultados**
   * **Setup experimental:**

Los datos y parámetros a configurar y que son introducidos manualmente son los siguientes:

- Número de bins: 5

- Peso máximo de bin: 150

- Número de ítems: 10

- Número de generaciones: 50

- Peso de los ítems:Seleccionados de manera aleatoria.

|  |  |
| --- | --- |
| **Item** | **Peso** |
| 1 | 111 |
| 2 | 45 |
| 3 | 62 |
| 4 | 7 |
| 5 | 34 |
| 6 | 115 |
| 7 | 107 |
| 8 | 90 |
| 9 | 63 |
| 10 | 97 |

La métrica de evaluación es el Fitness, que se define aquí como la diferencia de peso entre el contenedor más pesado y el contenedor más liviano. El objetivo es que se minimice la suma total del tamaño de los ítems que no pudieron ser asignados junto con la suma total de áreas no usadas en cada Bin. Se probaron dos operadores de cruzamiento (one point crossover y uniform crossover) y dos de mutación (Bit y Multiflip).

El Algoritmo genético (AG) se evaluó con los operadores de cruzamiento one point crossover y uniform crossover. En estas pruebas se deshabilitó la mutación. Luego se habilitó la mutación y se probó con tasas de mutación de 0.1, 0.3, 0.5 y 0.8

El tamaño de la población usada fue de 30 individuos. En cada configuración descrita se hicieron 10 corridas en el notebook para obtener conclusiones estadísticamente válidas. Finalmente, se probó con número de bins = 5 y Items = 10. El código y las pruebas fueron desarrollados en un notebook de Google Colab, usando Python 3.8+.

Los experimentos arriba descritos fueron planificados para poder comparar los enfoques propuestos y desarrollados. Algunas respuestas dados los experimentos:

* Los enfoques desarrollados siempre resuelven el problema. No obstante, se evidencia que dependiendo de la configuración de los parámetros y operadores se puede obtener un mejor Fitness.
* De los experimentos se puede observar que unos enfoques resuelven más eficientemente el problema, usando menor tiempo y recursos, obteniendo un mejor Fitness.
  + **Resultados y Discusión:**

Al realizar la ejecución del AG para 26 combinaciones de operadores de mutación y cruzamiento se obtuvo como 10 mejores resultados las siguientes combinaciones.

|  |  |
| --- | --- |
| **Combinación de operadores** | **AVG** |
| Crossover OnePoint y M3 flip | 234.8 |
| Crossover Uniform y M1 multiflip | 242.4 |
| Crossover OnePoint y M0 | 275.4 |
| Crossover Uniform y M3 flip | 1197.9 |
| Crossover OnePoint y M5 flip | 1201.1 |
| Crossover Uniform y M0 | 1201.3 |
| Crossover OnePoint y M1 multiflip | 1209.3 |
| Crossover Uniform y M1 flip | 1211.9 |
| Crossover Uniform y M3 multiflip | 1229.7 |

**Mejor solución**

Cromosoma con mejor fitness:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 2 | 3 | 5 | 1 | 2 | 4 | 1 | 0 | 5 | 3 |

**Calificación**

Trial 6: Fitness 199

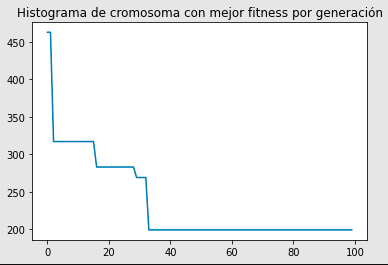
Fitness promedio de 10 trials: 234.8

**Operadores utilizados**

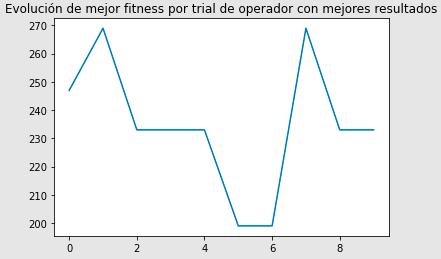
Crossover OnePoint

Mutación flip con valor de mutación de 0.3

**Gráficos de la solución**

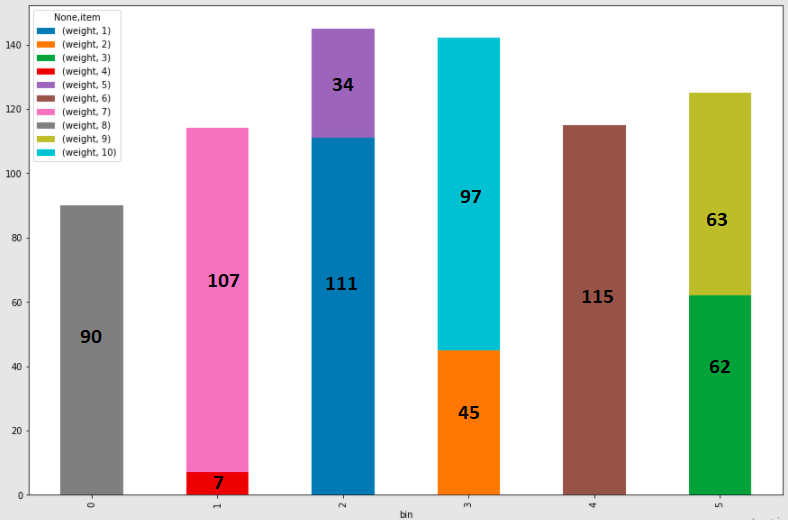


En las primeras generaciones el fitness baja rápidamente, luego la tendencia se desacelera convergiendo en el valor óptimo cerca de la generación 40.



A partir del segundo trial los valores de mejor fitness fueron disminuyendo hasta llegar al sexto, en donde se encontró el mejor fitness

**Representación de la asignación de items**.



Se observa que el item representada por la barra de color gris no fue asignado a ninguno de los 5 bins para que se pueda cumplir con la función de fitness planteada,.

**Segunda mejor solución**

**Cromosoma com mejor fitness:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 5 | 4 | 3 | 2 | 5 | 1 | 2 | 0 | 3 | 4 |

**Calificación**

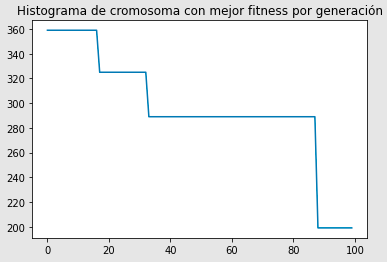
Trial 5: Fitness 199

Fitness promedio de 10 trials: 242.4

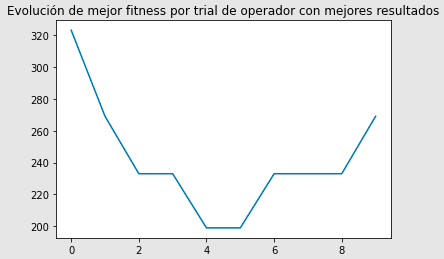
**Operadores utilizados**

Crossover Uniform

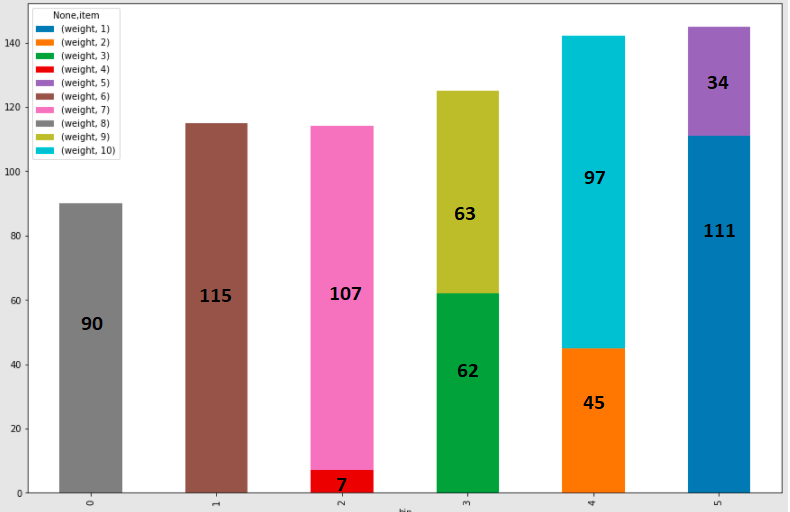
Mutación multiflip con valor de mutación de 0.1



En este caso no hay un bajón considerable en las primeras generaciones. El decaimiento del valor es gradual, llegando al mejor valor de fitness entre las últimas 20 generaciones.



Al igual que en el anterior gráfico, los trials de la mitad son los que tuvieron mejor valor de fitness.



Se observa que la distribución de los items es la misma que en la solución descrita anteriormente. Solo cambiaron los bins en los que se encuentran las agrupaciones de items.

1. **Conclusión**

Se encontraron los resultados más positivoscuando se usaron los operadores de cruzamiento de tipo Onepoint y Uniform combinados con operadores de mutación con probabilidad de mutar inferior a 0.4. El cruzamiento de tipo Onepoint encontró buenos valores de fitness, esto debido a que al combinar porciones de los cromosomas padres es más probable encontrar agrupaciones de ítems óptimas y con ello conlleva a que la población mejore.

En base a los resultados obtenidos se concluye que para este tipo de problema, la combinación de operador de mutación multiflip junto a la ausencia de operador de cruzamiento no aporta favorablemente a la búsqueda de valores óptimos. Esto debido a que las combinaciones de ítems que cumplan con la capacidad de los bins son limitadas y el cambiar más de un gen del cromosoma por un valor de bin aleatorio podría empeorar la medida de fitness de los offspring posiblemente buenos..

El enfoque propuesto al proyecto para que se reemplacen los cromosomas con menor fitness a expensas de que aumente la población, logró que la población no se vea infestada de valores subóptimos pero no evitó que parte de las soluciones caigan en el hard constraint diseñado.

Por último se comprobó lo comentado por Falkenauer [1], esto es que al usar una representación orientada a bins llevará a la larga a crear soluciones redundantes. Esto se pudo observar al detectar que las 2 mejores soluciones solo difieren en el orden de los bins.

1. **Sugerencias de trabajos futuros**

Para continuar analizando la performance del algoritmo genético, se puede experimentar con representaciones de encoding del cromosoma diferentes a las que se implementó (representación basada en bins) . Una de las representaciones a tomar en cuenta es la representación basada en grupo, propuesta por Falkenauer [2] que viene a ser un algoritmo genético orientado a la agrupación híbrida. Así mismo el enfoque de asignación de ítems en bins de estructura 1D puede ser ampliado para abarcar bins 2D o 3D ,esto ampliará también los posibles casos de uso. Con el enfoque utilizado , se pueden abordar más problemas de la vida real como el llenado de contenedores, llenado de carga de un camión con mayor cantidad de variables con las que interactuar [3] , así como en otros relacionados a la cadena de suministros.

1. **Link del repositorio del trabajo**

El trabajo realizado se puede consultar en el siguiente repositorio:

<https://github.com/jchaconm/PUCP-Proyecto-Computacion-Evolutiva>

1. **Declaración de contribución de cada integrante**

José Chacón aportó con la exploración de fuentes para el proyecto, el diseño de la representación del cromosoma y codificación de las ejecuciones del AG.

Deby Guzman aportó en la revisión de las fuentes utilizadas, diseño del cálculo del fitness y análisis de resultados obtenidos.

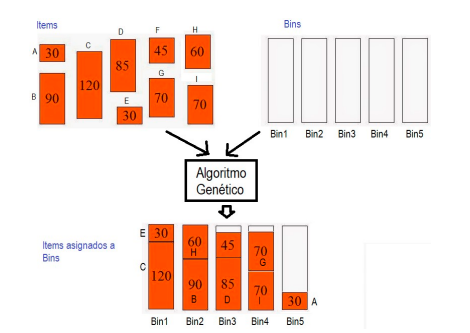
Roberto Cespedes aportó en el diseño e implementación de la representación del cromosoma, cálculo del fitness, algoritmo genético y ejecuciones de las combinaciones.

Andrés Hinostroza aportó en el análisis de los resultados, observando el comportamiento del fitness en base a los parámetros y operadores genéticos en prueba.

1. **Referencias**
2. Falkenauer R E. (1998) ”Genetic algorithms and grouping problems, ” J. Wiley & Sons, 25-103
3. Falkenauer R E., “A new representation and operators for genetic algorithms applied to grouping problems,” Evolutionary Computation, vol. 2, no. 2, pp. 123–144, 1994. [Online]. Available: <https://doi.org/10.1162/evco.1994.2.2.123>
4. Ramesh R (2001) A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. Comput Aided Des 33(12):879–891

**Figuras y tablas**

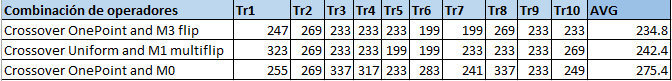
**Caso de uso a optimizar en el proyecto con AG**

****

**Tabla completa de ranking de combinaciones de operadores según valor promedio tras 10 trials**

|  |  |
| --- | --- |
| **Combinación** | **AVG** |
| Crossover OnePoint y M3 flip | 234.8 |
| Crossover Uniform y M1 multiflip | 242.4 |
| Crossover OnePoint y M0 | 275.4 |
| Crossover Uniform y M3 flip | 1197.9 |
| Crossover OnePoint y M5 flip | 1201.1 |
| Crossover Uniform y M0 | 1201.3 |
| Crossover OnePoint y M1 multiflip | 1209.3 |
| Crossover Uniform y M1 flip | 1211.9 |
| Crossover Uniform y M3 multiflip | 1229.7 |
| Crossover OnePoint y M1 flip | 1247.1 |
| Crossover Uniform y M5 flip | 2178 |
| Crossover Uniform y M5 multiflip | 2181.4 |
| Crossover Uniform y M8 flip | 2206 |
| No Crossover y M8 flip | 2334.4 |
| Crossover OnePoint y M5 multiflip | 3162.7 |
| Crossover OnePoint y M8 flip | 3163.1 |
| Crossover OnePoint y M3 multiflip | 3177.1 |
| No Crossover y M5 flip | 3304.3 |
| No Crossover y M3 flip | 3305.9 |
| Crossover OnePoint y M8 multiflip | 4143 |
| Crossover Uniform y M8 multiflip | 4144 |
| No Crossover y M1 flip | 4268.8 |
| No Crossover y M3 multiflip | 4302.8 |
| No Crossover y M8 multiflip | 4305.6 |
| No Crossover y M1 multiflip | 4326.6 |
| No Crossover y M5 multiflip | 4326.6 |

**Tabla Podio de combinación de operadores de acuerdo a su mejor valor de fitness en 10 trials**

****