“Uso de computación evolutiva para hallar ubicaciones óptimas de una cadena de supermercados”

Autores: Añanca Arango, Pedro Christian. De Lama Ramírez, Diego Gonzalo. Díaz Villanueva, Julio Leonardo. Tello Rivera, Erika Elizabeth.

**Resumen-** Este proyecto busca las 10 mejores ubicaciones para nuevos supermercados basado en coordenadas geográficas y población de impacto. Para este problema de optimización se resolvió con algoritmos genéticos cuyo fitness, se creó una versión de fitness que usa un dataframe con información geoespacial, calcula distancias entre ubicaciones, y acumula estos datos en un cromosoma. Para la población, se utilizó un porcentaje de la suma total de las localizaciones, dividido entre la población total de 60 localizaciones. Se ejecutó el algoritmo ajustando hiper parámetros como operadores de mutación de cruzamiento tasa de mutación, crossover y tasa de mutación, y se usaron dos tipos de algoritmos: mono objetivo y multi objetivo. El mejor resultado para el algoritmo Mono objetivo fue el experimento con operador de cruzamiento One Point, el operador de Mutación ‘Flip’ y tasa de mutación 0.15 cuyo fitness es 92.58 y el set de localizaciones es [2, 11, 14, 15, 28, 29, 39, 50, 57, 60]. Por su parte, el mejor resultado del algoritmo Multiobjetivo fue el experimento con operador de cruzamiento ‘One Point’, el operador de Mutación ‘ Multiflip’ y tasa de mutación 0.75 cuyo fitness tiene dos valores suma del tamaño normalizado de la población igual a 0.22 y suma de distancias igual a 575.57km.

——————————◆——————————

1. **Introducción**

El problema plantea la búsqueda de las soluciones óptimas para optimizar 10 ubicaciones para supermercados. Para lo cual, se plantean 60 posibles ubicaciones distribuidas a lo largo de la provincia de Lima. Cada ubicación o punto, se cuenta con su latitud, longitud y población a 500 metros a la redonda.

Se muestra en la figura 1 con la distribución de las localizaciones en función de su longitud y latitud.

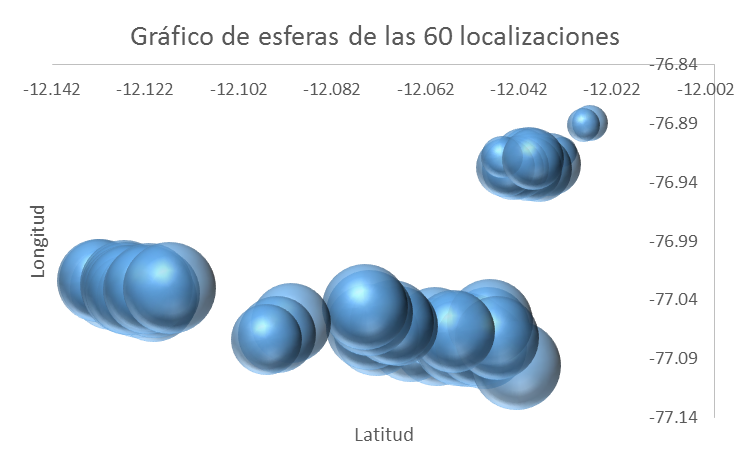


Figura 1. Representación de locaciones por esferas

Se tienen dos objetivos: maximizar la suma de la población que vive a 500 metros alrededor de los supermercados y maximizar la suma de distancias entre los supermercados escogidos.

Se propone emplear algoritmos genéticos mono-objetivo y multiobjetivo.

1. **Metodología**

* Describir la representación de estados/individuos elegida,
* Describir el comportamiento entrada/salida de los enfoques.
* Describir los operadores/funciones desarrolladas para solucionar el problema planteado. No describir métodos genéricos, sino detalles de cómo se aplicó/adaptó al problema
  1. **Primer enfoque (Algoritmo mono-objetivo)**

Tanto en el primer como en el segundo enfoque los individuos serán descritos como un cromosoma de longitud 10 cuyos alelos irán en un rango de 0 a 59. Este número indica la localización (total 60 candidatos). Por ello es necesario que no se repita ningún alelo en el cromosoma,

La obtención del fitness básicamente consistirá en el producto entre las sumas de distancias de las locaciones escogidas y la suma normalizada de las poblaciones.

* 1. **Segundo enfoque (Algoritmo multi-objetivo):**

La obtención del fitness básicamente consistirá en un array de dos valores. Como primer valor se obtendrá la suma de distancias de las locaciones escogidas y como segundo valor la suma normalizada de las poblaciones.

* 1. **Operadores para solucionar el problema planteado.**

La clase Individuo tiene operadores asociados al cruzamiento de los padres y la mutación de los individuos hijos:

* **Crossover\_one\_point**: Se elige un punto aleatoriamente a lo largo del cromosoma (lista de localizaciones) y se intercambia los elementos de la lista del primer padre con índice menor igual al punto al segundo padre y los elementos de la lista del segundo padre mayores al punto elegido se intercambian al primer padre. Para adaptar a nuestro problema se parametriza para no intercambiar los elementos de cada lista que se tuvieran en común y así evitar localizaciones repetidas en un mismo cromosoma.
* **Crossover\_uniform**: Cada alelo (elemento de la lista de localizaciones) se elige aleatoriamente de los dos progenitores con una probabilidad del 50%. Si es un alelo con valores en común en ambos padres no se hace el intercambio.
* **Mutation\_flip:** Se elige un alelo del cromosoma y cambia su valor aleatoriamente en un rango específico del 0 al 59 excluyendo los valores en común de ambos padres.
* **Mutation\_multiflip:** Se elige hasta un 50% de alelos como máximo y cambia sus valores aleatoriamente en un rango específico.

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

* **Descripción de datos usados**

Para resolver el problema se nos proporcionó un mapa (1) de la ciudad en Google Earth con 60 locales candidatos donde la cadena de supermercados podría localizar sus supermercados. De ese mapa se podría extraer las coordenadas, pero optamos por usar un excel llamado ‘Candidatos supermercados’, el cual contiene una tabla con las localizaciones numeradas. Por cada localización candidata hay una coordenada asignada en formato longitud y latitud, la descripción de la dirección de dicha localización y además la población estimada que hay en 500 metros a la redonda de esta. El campo clave de esta tabla es el número de localización y es este dato que se ingresará en una lista de 10 localizaciones elegidas aleatoriamente, cada lista será un individuo de la población generada. Esta población es la que será el parámetro principal de los algoritmos genéticos Mono-objetivo y Multiobjetivo que optimizará la solución del problema.

* **Descripción de métrica de evaluación**

Cómo métrica de evaluación en los algoritmos genéticos se debe plantear un fitness. Para este problema los componentes del fitness son:

Suma de distancias de las localizaciones

Para esto se generó previamente una matriz de 60 x 60, las cuales representan el número de localizaciones, que calcula las distancias cada dos localizaciones a partir de su latitud y longitud, obtenida del excel anteriormente descrito.

Para sumarizar las distancias de las 10 localizaciones candidatas a evaluar se promedian todas las distancias entre las 10 localizaciones que podemos identificar en el triángulo superior de la matriz de distancias. Este resultado luego se convierte en kilómetros a continuación en la figura 2 enseñaremos el data-frame generado.

Imagen de la pantalla de una computadora

Descripción generada automáticamente con confianza baja

Figura 2 Data-frame generado con distancias en km.

* **Suma normalizada del tamaño de la población de las localizaciones**

Para el tamaño de población se asocia el número de localización que se encuentra en la lista candidata a evaluar con el número de fila de la tabla de datos proporcionada y se suma el valor del campo ‘Población 500m’ de las 10 localizaciones del individuo evaluado. Luego para normalizar esta variable, el resultado se divide entre la suma de las 60 localizaciones candidatas.

* **Métrica de evaluación del algoritmo genético Mono-objetivo**

Para determinar el mejor candidato el algoritmo Mono-objetivo primero calcula el fitness de la mejor generación de la población como el producto de la suma de distancias y la suma normalizada del tamaño de la población. Luego se identifica el mayor valor de fitness del set de experimentos. Si hay 2 o más generaciones que tienen el mismo fitness entonces se identifica quien tiene mayor tamaño normalizado de la población y luego qué generación tiene mayor suma de distancias.

Métrica de evaluación del algoritmo genético Multiobjetivo

A diferencia del algoritmo genético Mono-objetivo, este algoritmo determina un conjunto de soluciones llamadas Frontera de Pareto. Para determinar el mejor candidato se identifica qué experimento tiene mejor fitness del algoritmo Mono-objetivo y luego se plotea el valor de su suma de distancias y el valor de su tamaño normalizado de población junto con las soluciones de la Frontera de Pareto. Se revisa si la solución está dominada o no.

* **Descripción de experimentos hechos**

Para cada algoritmo se realizaron 20 experimentos con los siguientes parámetros.

* Número de elementos por cada individuo: 10 (números del 0 al 59 que no se repitan)
* Número de individuos de la población: 300
* Número de generaciones: 50
* Número de corridas: 3

En la tabla 1 podemos ver además los experimentos realizados con los diferentes hiper-parámetros seleccionados:

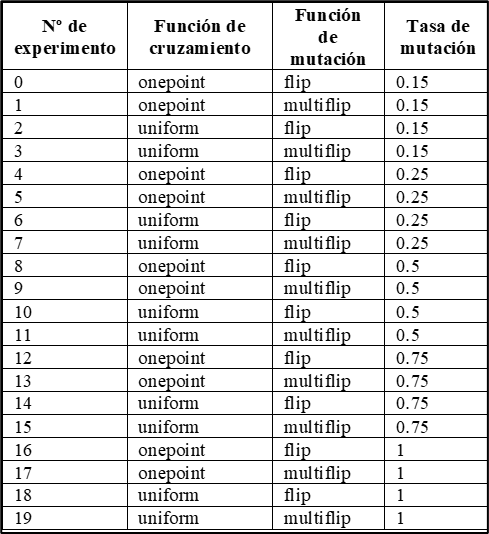


Tabla 1 Set de Experimentos realizados

* **Resultados**

**Resultados algoritmo mono-objetivo:**

Se presentan 20 resultados, en los que se presentan iteraciones para:

* Crossover: uniform y onepoint,
* Mutation: flip y multiflip
* Tasa de mutación: 0.15, 0.25, 0.50, 0.75 y 1.

En experimentos previos se identificó que los resultados convergen antes de la generación 50

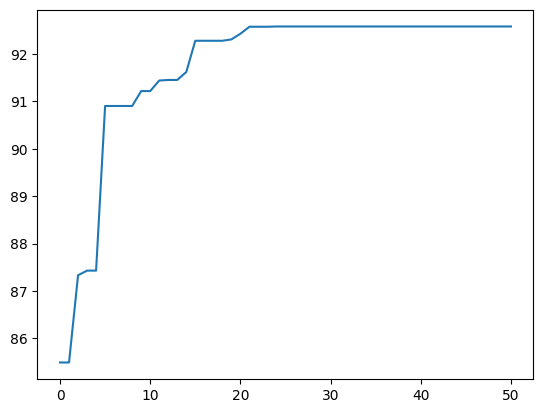


Figura 3. Nº generaciones vs Fitness

Obteniendo que la combinación de parámetros que alcanza un mayor fitness es la de Onepoint, Flip y un Pmut de 0.15. La generación mejor es la 31

Las 10 ubicaciones óptimas son los puntos 2, 11, 14, 15, 28, 29, 39, 50, 57, 60. Con un fitness de 92.58.

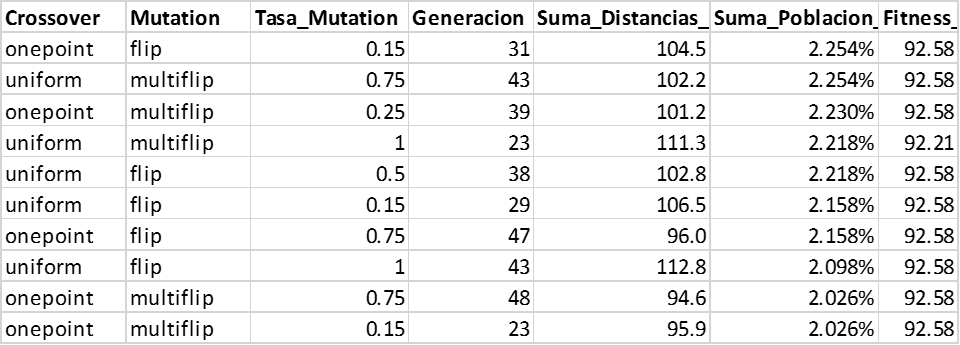


Tabla 2. Set experimentos del algoritmo mono objetivo con sus resultados.

**Resultados algoritmo multiobjetivo:**

Se presentan 20 resultados, en los que se presentan iteraciones para Crossover: uniform y onepoint, Mutation: flip y multiflip, Tasa de mutación: 0.15, 0.25, 0.50, 0.75 y 1.

Obteniendo que la combinación de parámetros que alcanza un mayor fitness es la de Onepoint, multiflip y un Pmut de 0.75.

Para este algoritmo, se hallaron los valores máximos de suma de distancias y suma de población. Obteniendo que los valores máximos son: Suma de distancias = 0.2225 y Suma de población = 575.5784. Las 10 ubicaciones óptimas son los puntos: Los puntos óptimos son: 11, 14, 15, 28, 29, 38, 45, 50, 56 y 60.

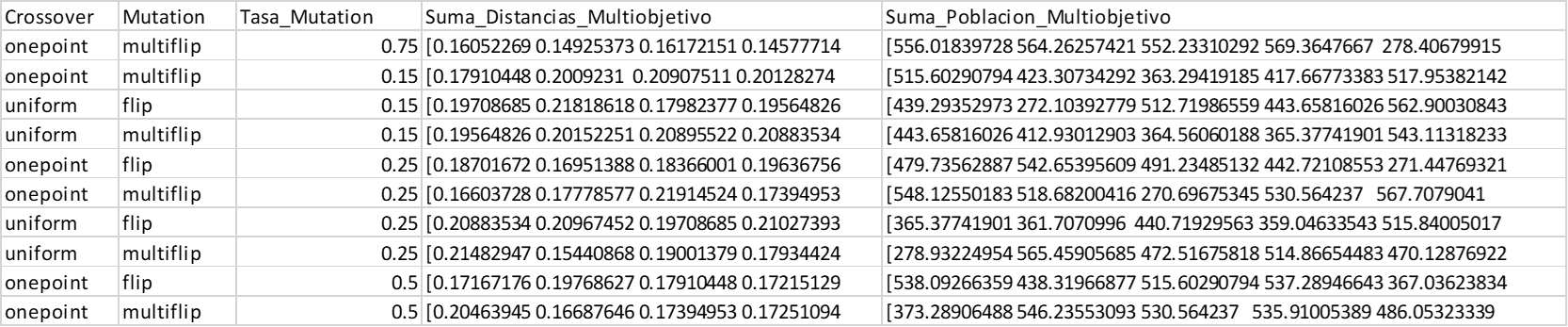


Tabla 3 Set experimentos del algoritmo multiobjetivo con sus resultados.

En la figura 4 se puede observar el experimento del mejor fitness obtenido con el algoritmo mono objetivo y el pareto resultante con el algoritmo multiobjetivo del mismo experimento:

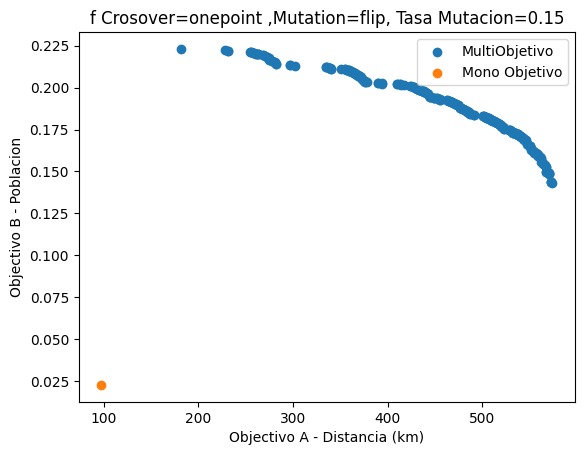


Figura 4 Pareto con los mejores resultados del experimento mono objetivo.

Incluyendo a eso se compara los resultados del pareto obtenidos con los mejores parámetros de la experimentación del enfoque multi-objetivo.

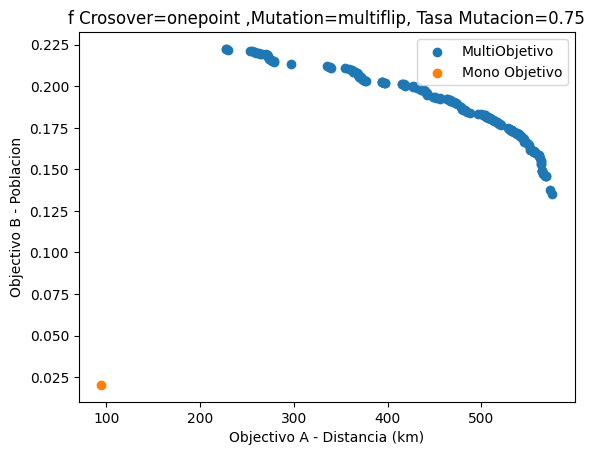


Figura 5 Pareto con los mejores resultados del experimento del multi objetivo.

Al comparar los mejores experimentos entre ambos algoritmos vemos que en el resultado del algoritmo mono objetivo con el fitness planteado hace este dominado por las soluciones de la frontera de Pareto del algoritmo multiobjetivo.

Revisando los gráficos 4 y 5 podemos ver que ambos algoritmos tienden a optimizar más la variable distancia que la variable tamaño normalizado de la población.

1. **Conclusión**

* Se pudo optimizar la localización de supermercados utilizando un enfoque de algoritmos genéticos mono objetivo y multi objetivo pudiendo manipular sus hiper-parámetros.
* De lo obtenido podemos identificar que el resultado converge en menos de la generación 50 limitando el nivel exploratorio del algoritmo mono objetivo.
* La definición del fitness planteado para el algoritmo monoobjetivo hace que el tamaño normalizado de la población castigue al tamaño de distancias como resultado de la productoria de ambos.
* Es cierto que la forma de obtener el fitness dio resultados positivos pero este puede ser definido de otra manera más óptima con la cual se pueden tener mejores resultados.
* El algoritmo multiobjetivo presenta un mejor performance que el algoritmo monoobjetivo dado que su frontera de soluciones domina la solucion monoobjetivo.

1. **Sugerencias de trabajos futuros**

* Se sugiere mejorar el algoritmo de iniciación de la población para evitar que los individuos tengan todos sus elementos comunes.
* Para trabajos futuros, para el algoritmo multiobjetivo, se plantea hallar otros métodos para obtener las soluciones con resultados máximos (para cada objetivo).
* Se puede mejorar el enfoque empleando una selección de padres por torneo, en lugar de por ruleta; lo cual sería un método de selección más elitista, probablemente se cuente con mejores descendientes.
* Probar con distintas poblaciones para visualizar el efecto e impacto a lo largo del tiempo de los operadores de mutación, crossover y obtener hiper parámetros más generalizables.
* Probar con otros operadores de cruzamiento y mutación podría mejorar el resultado del algoritmo monoobjetivo y buscar acercar su resultado a la frontera de pareto del multiobjetivo.
* Se podría retar el algoritmo aplicado con otros algoritmos de maximización.
* Se podría buscar otro tipo de operador entre las distancias y la población en vez de la multiplicación.

1. **Link del repositorio del trabajo**

Github: <https://github.com/OSOMASTER123/Uso-de-computaci-n-evolutiva-para-hallar-ubicaciones-ptimas-de-una-cadena-de-supermercados-PUCP>

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

Añanca Arango,Pedro Christian.

Adaptación de algoritmo multiobjetivo y creación de la función e informe.

De Lama Ramírez, Diego Gonzalo:

Adaptación de algoritmos de mutación y cruzamiento, tabla de resultados e informe

Díaz Villanueva, Julio Leonardo :

Planteamiento de variables de entrada, función fitness, notebook e informe.

Tello Rivera, Erika Elizabeth.

Funcion fitness, adaptación de algoritmos de mutación y cruzamiento, set de experimentos e informe.

1. **Referencias**

[1] Mapa con las 60 loclizaciones <https://drive.google.com/open?id=1w7n77ByWK6TrX74lOBTwID719LRj_UQz&usp=sharing>.

[2] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing. Springer, 2nd Edition, 2015.

[3] Gen, M. and Cheng, R. (1997) Genetic Algorithms & Engineering Design. John Wiley & Sons, Inc., New York.

[4] Miller, B. (2017). Introduction to Genetic Algorithms Including Example Code. Towards Data Science. Recuperado el 20 de mayo de 2023, de <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>