## Línea horizontal



Algoritmo de búsqueda armónica

30/06/2020

**─**

Héctor Eduardo Oliván González

Judith Martínez Olvera

Carlos Francisco Trejo Juárez

Isaac González García

# Algoritmos Evolutivos para Optimización

Con el actual y constante desarrollo de la tecnología, las grandes cantidades de información que se manejan actualmente son cada vez más grandes en casi todos los ámbitos conocidos. Debido a estos volúmenes grandes, los métodos de optimización exhaustiva o en programación lineal, cuadrática, mixta, etc. han cedido el paso a métodos heurísticos y evolutivos. Para estos últimos su mayor enfoque ha sido en la Ingeniería, la Medicina, la Astrofísica, las Telecomunicaciones, la Economía y en general en cualquier actividad en que la resolución de problemas complejos no lineales sea de vital importancia.

Los algoritmos evolutivos son un método robusto para la optimización de problemas complejos, es decir, cuando el número de elementos es muy elevado, hay considerables restricciones, las variables se encuentran en cantidades grandes o pueden coexistir una o varias soluciones óptimas. En tales casos, los algoritmos de programación tradicionales no tienen un desempeño satisfactorio, es por esto que se recurre a otro tipo de estrategias. Los algoritmos evolutivos parten de una población de cierta cantidad ‘’n” de individuos los cuales pueden ser cualquier clase de datos, estrellas en un cuadrante del espacio, cotizaciones en la bolsa de valores, historiales clínicos, incidencias de accidentes de tráfico, etc. sobre estos grupos de información se busca un conjunto de parámetros que maximicen o minimicen una función de adaptación. Esto se logra en un proceso de pasos o iteraciones sucesivas en las que se combinan y seleccionan los criterios que más se aproximen a los criterios deseados. Estos métodos permiten llegar a soluciones distintas recorriendo una cantidad inmensa de caminos en cada proceso de optimización.

# Introducción al algoritmo de Búsqueda Armónica.

Aunque los algoritmos actuales han solucionado gran cantidad de problemas de optimización los inconvenientes que causan o a los que se enfrentan, han llevado a diseñar nuevos algoritmos heurísticos basados en analogías con fenómenos naturales o artificiales los cuales son mucho más potentes. Para el desarrollo del concepto de búsqueda armónica o BA, (*HS* por *Harmony Search*) se imita la improvisación que realizan los músicos. La música es una de las actividades más satisfactorias generadas por los humanos. La idea para introducir este nuevo algoritmo derivado de un fenómeno que se encuentra en la interpretación musical, es por ejemplo, la búsqueda de una mejor armonía durante una presentación de un trío de jazz, que le da un gran valor estético a la pieza que se está tocando.

Desde la antigüedad, la armonía musical se ha estudiado y definido como una combinación de sonidos que son agradables al oído, además están los estándares estéticos de la música que es una rama de la filosofía ocupada en la calificación de la música con base en la naturaleza, el gusto, la creación y la apreciación de la misma. En la naturaleza, esta armonía es una relación especial entre varias ondas de sonido con diferentes frecuencias, las armonías son producidas por las tonalidades que brindan cada instrumento que se toca.

Relacionando esto con un algoritmo de optimización, la búsqueda de una mejor armonía representa la búsqueda de un óptimo global. Los sonidos para una mejor interpretación pueden mejorarse práctica tras práctica, así como los valores para una mejor evaluación de la función objetivo pueden mejorarse iteración por iteración. A continuación se muestra una tabla en la que se representa de mejor manera la analogía:

|  |  |  |
| --- | --- | --- |
| **Factor de Comparación** | **Algoritmo de optimización** | **Interpretación musical**  **(Proceso)** |
| **Mejor estado** | Óptimo Global | Mejor armonía musical |
| **Estimado en** | Función | Estándar estético |
| **Estimado con** | Valores de variables | Tonos de instrumentos |
| **Unidades del Proceso** | Cada iteración | Cada práctica |

El ahora nombrado algoritmo de búsqueda armónica BA, en forma muy general y simplificada, se logra siguiendo los siguientes pasos:

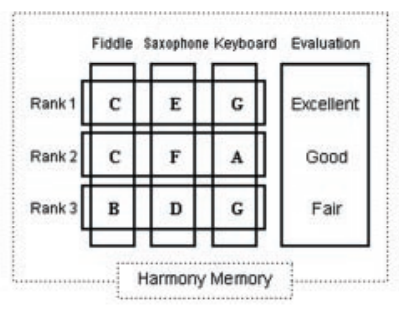
1. Inicializar una memoria de armónicos. (*Harmony Memory, HM*).
2. Improvisar una nueva HM.
3. Si la nueva armonía suena mejor que la armonía mínima en HM, incluir la nueva armonía en la memoria de armónicos y excluir la armonía mínima.
4. Si no se cumplen los criterios de detención, repetir desde el paso 2.

A continuación se muestra un ejemplo del funcionamiento del algoritmo simbolizado por un trío de jazz:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Do | Re | Mi | Fa | Sol | La | Si |
| C | D | E | F | G | A | B |

Nomenclatura de las notas musicales.

Se contará con un violín, un saxofón y un teclado para componer la memoria armónica.



* La HM se llena con notas aleatorias, (C,C,B) en el violín, (E,F,D) en el saxofón y (G,A,G) en el teclado.
* Ahora se produce una nueva improvisación en los tres instrumentos, se toca C en el violín, D en el saxofón y A en el teclado.
* Cada una de las notas tiene la misma probabilidad de ser reemplazada por el nuevo armónico.
* Si el nuevo armónico suena mejor que cualquiera de los tres, se incluye en la HM y se quita el peor armónico, en este caso el Rank 3.
* El proceso se repite hasta lograr el mejor sonido, (mejor óptimo global).

# Discusión y Aplicaciones del algoritmo

Una de las grandes ventajas de la búsqueda armónica es que a diferencia del algoritmo genético, BA crea un nuevo vector a partir de la consideración de todos los vectores existentes y no solo de dos. Además, la BA no requiere la definición de valores iniciales para variables de decisión. Estas características le permiten al algoritmo aumentar su flexibilidad y encontrar mejores soluciones.

El algoritmo se utilizó para escoger el camino más corto que tendría que recorrer un vendedor ambulante para visitar 20 ciudades.

Otra aplicación del algoritmo fue determinar el diámetro mínimo de una tubería para una red de suministro de agua, considerando una presión máxima fija en los nodos de la red.

Tres aplicaciones que tuvieron como algoritmo de solución la BA incluyendo los dos antes mencionados, demostraron que puede resolver un problema variable continuo así como problemas combinatorios, y supera a otros métodos heurísticos existentes. También se ha aplicado a otros problemas combinatorios o continuos por ejemplo, diseño óptimo de una red de tuberías, expansión óptima de una red de tuberías y calibración óptima de parámetros de un modelo hidrológico, en los que la búsqueda armónica ha superado a los métodos matemáticos y heurísticos existentes. En ejercicios similares se han comparado las iteraciones necesarias para encontrar la solución general en diversos algoritmos. Algunos de los datos más significativos para encontrar la solución fueron, 250000 iteraciones en el algoritmo genético y 70000 en el algoritmo de temple simulado, mientras que a la búsqueda armónica le llevó únicamente 1095 encontrar la misma solución, mostrando una mayor eficiencia.

# Descripción general

BA es un algoritmo metaheurístico inspirado en la manera en que músicos buscan la armonía óptima en la composición musical, el cual ha sido empleado exitosamente para resolver problemas complejos de optimización. En comparación con otros algoritmos meta-heurísticos reportados en la literatura, BA presenta varias ventajas como lo son la utilización de pocos parámetros de configuración y su rapidez.

# Diagrama del algoritmo

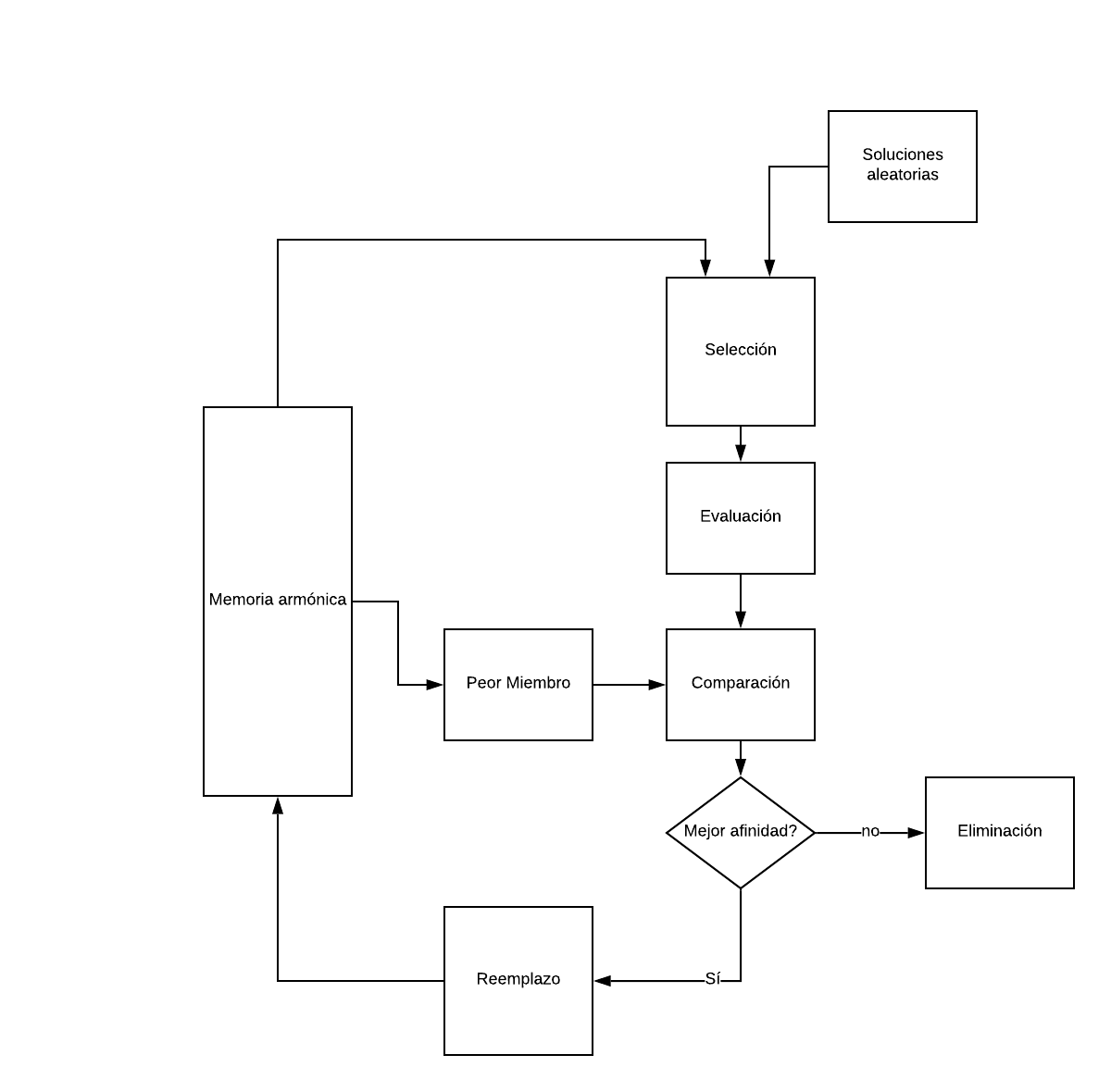


Figura 1. Método del algoritmo de búsqueda armónica.

# Procedimiento de la búsqueda armónica

Para entender mejor el procedimiento que lleva a cabo el algoritmo para encontrar los valores de los parámetros a ser optimizados primero se deben mencionar las variables con las que se trabajara en el pseudocódigo del algoritmo, las cuales son los siguientes:

· *HMS:* Tamaño de la memoria armónica, define el número de elementos cuyo mejor desempeño en el proceso de optimización pueden ser almacenados para su uso en el proceso de evolución.

· *HMCR:* Tasa de consideración de la memoria armónica, representa la probabilidad de construir una nueva solución a partir de los elementos previamente almacenados en la memoria HM.

· *PAR:* Tasa de ajuste del tono, define la probabilidad de volver a modificar una solución previamente generada.

· *BW*: Ancho de banda de ajuste del tono, expresa la magnitud máxima de dicha modificación.

· *NI*: Número de improvisaciones, representa la cantidad de iteraciones que el algoritmo ejecuta.

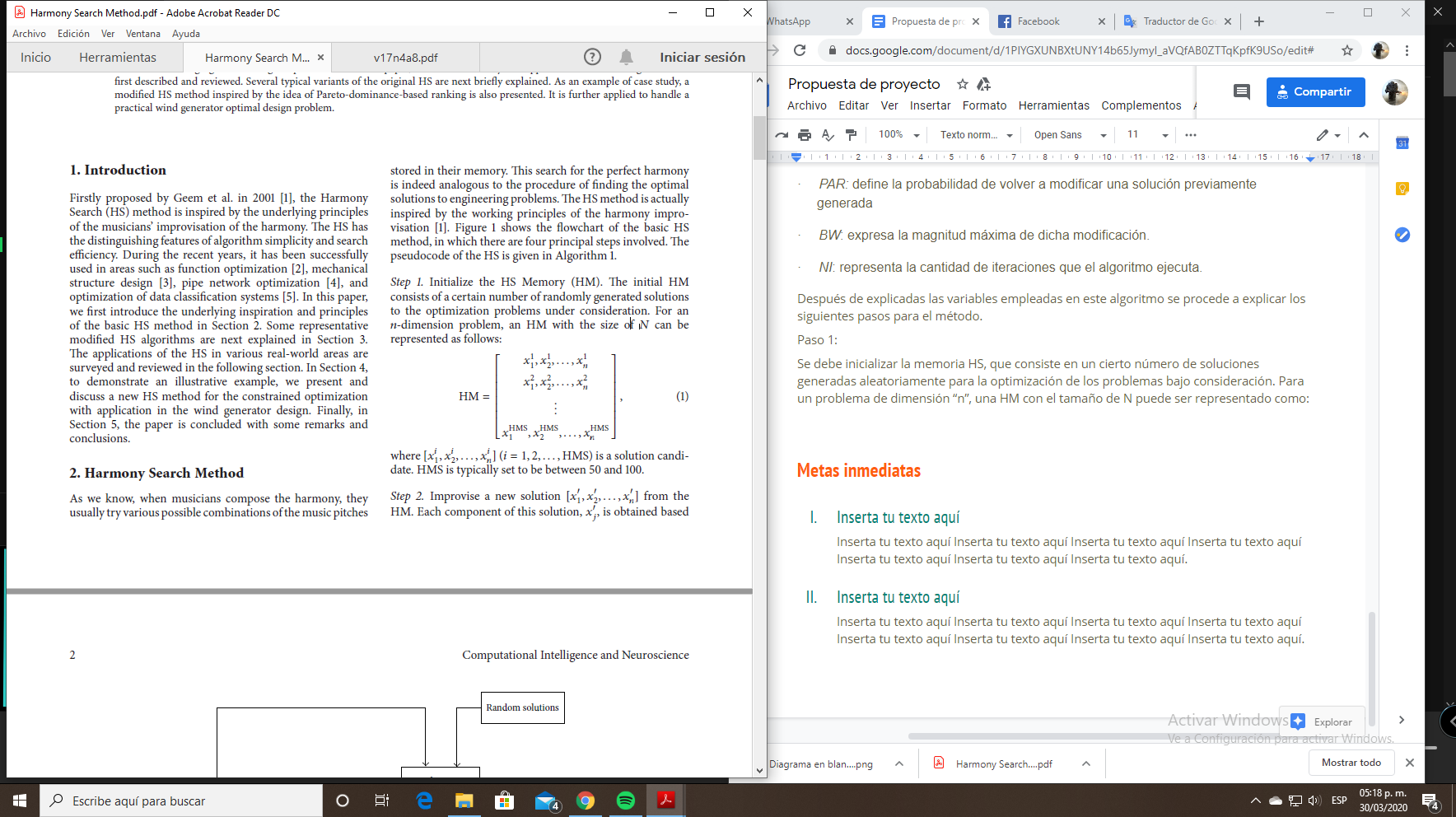
Después de explicadas las variables empleadas en este algoritmo se procede a explicar los siguientes pasos para el método.

**Paso 1: Inicializar los parámetros del problema y los parámetros de HS.**

El problema de optimización se define como minimizar (o maximizar) **f(x)** tal tal que **LBi<xi<UBi** donde, **f(x)** es la función objetivo, **x** es una solución candidata que consiste de N variables de decisión ( **xi** ), y **LBi** y **UBi**, son el límite de decisión más bajo y el más alto de cada variable, respectivamente (en la programación se denotó como minim, maxim, minim2 y maxim2). Los parámetros de HS se especifican en este paso. Estos parámetros son el tamaño de la memoria armónica (**HMS**), la tasa de consideración de la memoria armónica (**HMCR**), la tasa de ajuste del tono (**PAR**), el ancho de banda de ajuste del tono (**BW**) y el número de improvisaciones (**NI**).

**Paso 2: Inicializar la memoria armónica.**

Se debe inicializar la memoria HS, que consiste en un cierto número de soluciones generadas aleatoriamente para la optimización de los problemas bajo consideración. Para un problema de dimensión “n”, una HM con el tamaño de N puede ser representado como:

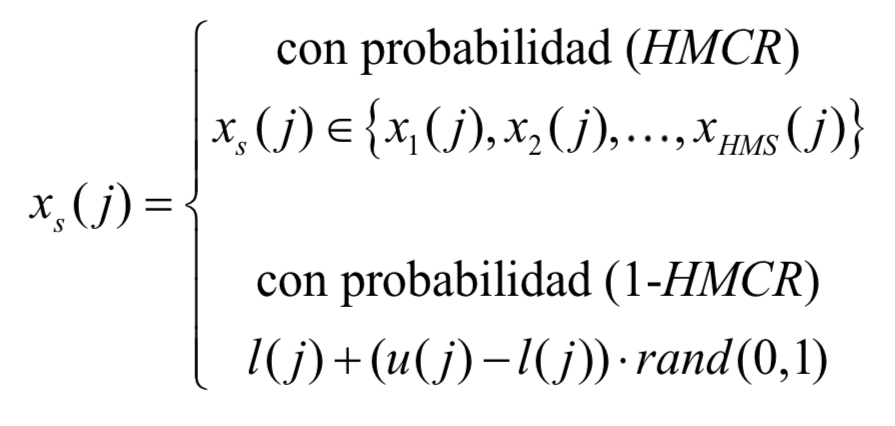


Donde cada una de las filas con (i = 1,2,...,HMS) son un candidato solución.

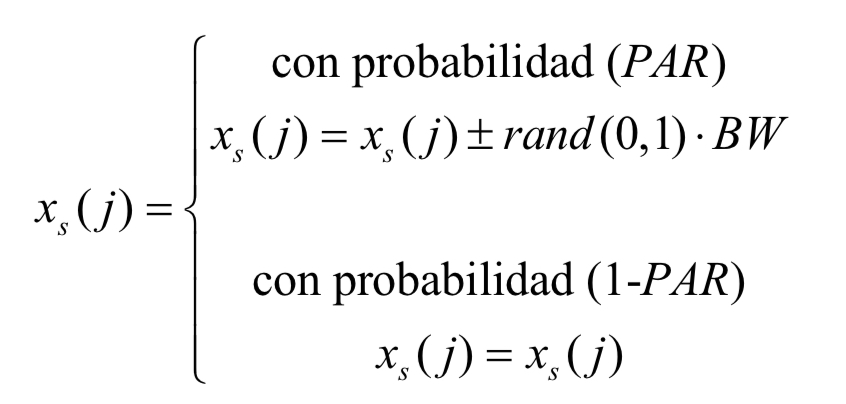
**Paso 3: Improvisar la nueva armonía.**

El proceso de generación de una nueva armonía es llamado improvisación. Se debe de improvisar una nueva solución desde la HM. Este nuevo vector armónico se genera utilizando las siguientes reglas: consideración de la memoria, ajuste del tono y selección aleatoria.

Cada componente de esta solución, , es obtenida con base en el HMCR. Este es definido como la probabilidad de seleccionar un componente de los miembros de la HM, y 1-HMCR es, además, la probabilidad de generar ésta, aleatoriamente. Si viene de la HM, este es seleccionado de la jta dimensión de un miembro HM aleatorio y se muta más de acuerdo con la velocidad de ajuste de tono (PAR), la cual determina la probabilidad de un candidato de la HM de ser modificado. Por examinado de memoria, re-inicialización aleatoria y ajuste de tono. Para el nuevo vector es elegido de cualquier de los valores ya existentes dentro de HM (del conjunto {x(1),x2(1),...,xHMS (1)}). Si r1 es menor que HMCR, de otro modo, xs (1) se obtiene a partir un valor aleatorio re-inicializado entre las fronteras del problema [l(1),u(1)]. examinado de memoria y re-inicialización aleatoria.



Para cada solución generada xs ( j) se realiza además un ajuste de tono, el cual es definido por la frecuencia de ajuste (PAR) y por el factor de modificación (BW).



**Paso 3: Actualizar la memoria armónica.**

Actualizar el HM. La nueva solución del paso 2 es evaluada. Si la nueva solución ofrece una mejor respuesta que el peor miembro en la HM, se reemplazará por la nueva iteración. De no presentar una mejoría será eliminada.

Si el valor de la función objetivo de xs es mejor que xw , entonces xs lo reemplazará dentro de la memoria (HM). De lo contrario el contenido de HM permanece sin cambios.

**Paso 4: Verificar el criterio de parada.**

Repetir el paso 2 al paso 3 hasta que exista un criterio preestablecido de terminación, por ejemplo, el máximo número de iteraciones.

En general, HS es poco sensible a los parámetros [1, 2], por esto, el algoritmo no requiere una afinación exhaustiva de los parámetros para obtener buenas soluciones. A pesar de lo anterior, es preciso destacar que los parámetros HMCR y PAR ayudan al método en la búsqueda de mejores soluciones globales y locales, respectivamente. PAR y bw tienen un profundo efecto en el rendimiento del algoritmo y es por esto que el ajuste de estos dos parámetros es muy importante.

# Pseudocódigo

ALGORITMO: Búsqueda armónica

VARIABLES DECLARADAS: HMS, HMCR, PAR, BW, NI

INICIO

Inicializar HM y calcular el valor de la función objetivo de cada vector de solución.

Improvisa una nueva armonía (solución) de la siguiente manera:

for (j=1 a ) hacer lo siguiente

si( < HMCR) entonces

(j) = (j)

donde (1, 2, …, HMS)

si( < PAR) entonces

(j) = (j)\* BW

donde , , rand(0,1)

termina condición si

si (j) < l(j) entonces

(j) = l(j)

termina condición si

si (j) > u(j) entonces

(j) = u(j)

termina condición si

sino

(j) = l(j)+r\*(u(j)-l(j)

donde rand(0,1)

termina condición si

termina ciclo for

Actualizar HM,

= si f() < f()

Si NI es finalizado, el mejor vector solución en HM es localizado, de otra manera se regresa a partir de la condición si (j) < l(j).

FIN

# Resultados

Funciones para para minimizar en el algoritmo de optimización.

(1)

Con límites , .

(2)

Con límites , .

**Resultado del algoritmo de búsqueda armónica con la función (1)**

X1 = 3.0006177400771383

X2 = 1.9951310751965028

Y(X1,X2) = 0.0003560795578462529

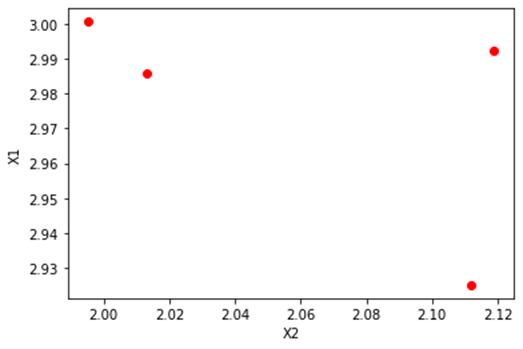


Ilustración 1. Gráfica de los puntos X1 y X2 de la función (1).

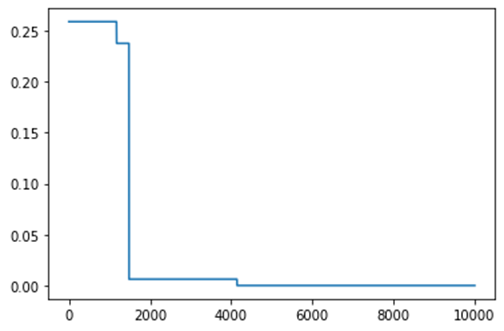


Ilustración 2. Función (1) evaluada con X1 y X2 en cada iteración

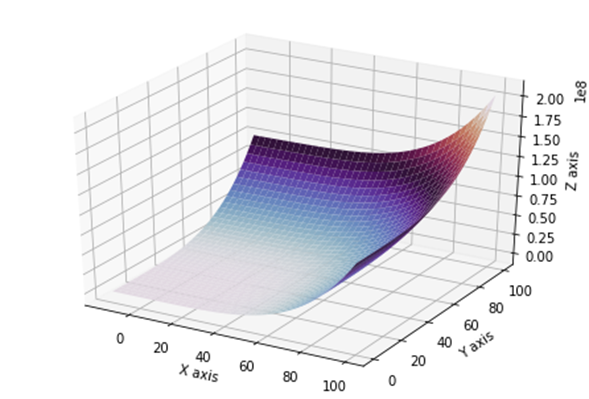


Ilustración 3. Gráfica 3D de la función (1).

**Resultado del algoritmo de búsqueda armónica con la función (2)**

X1 = 2.0009974420207803

X2 = 0.9892151049095739

Y(X1,X2) = 0.00011730885269631595

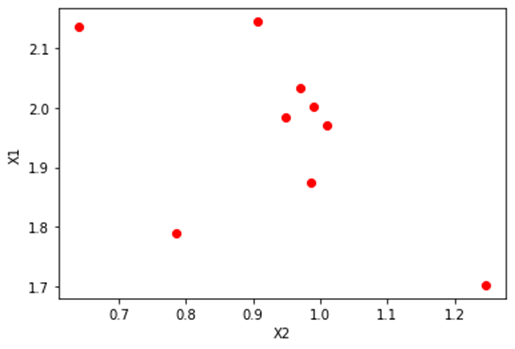


Ilustración 4. Gráfica de los puntos X1 y X2 de la función (1).

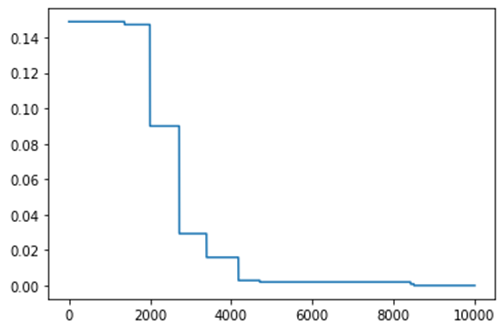


Ilustración 5. Función (2) evaluada con X1 y X2 en cada iteración.

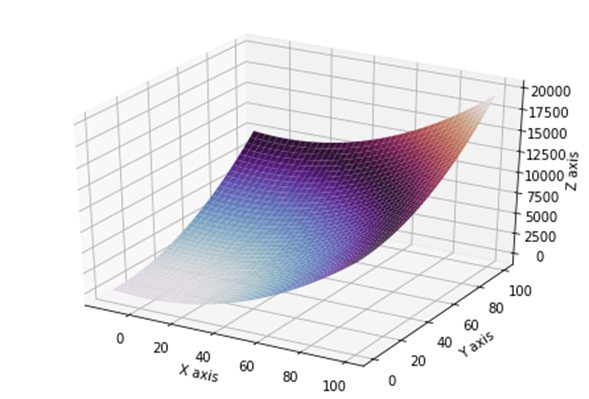


Ilustración 6. Gráfica 3D de la función (2).

# Análisis de resultados

Las gráficas más importantes son las mostradas en las ilustraciones 3 y 5, ahí se muestra en cada iteración la función evaluada con los valores de X1 y X2, ambas gráficas convergen al óptimo global, o bien al mínimo de la función. El número de iteraciones que se configuraron para ambas funciones fue de 10000 pero en cuanto a la función (1) a partir de la iteración 4000 ya se minimizó la función al máximo, por otro lado la función (2) necesitó un poco más de 8000 iteraciones para encontrar el valor mínimo.

**Conclusiones**

El algoritmo de búsqueda armónica (HS) es una poderosa herramienta de optimización, la cual no necesita de cálculo matemáticos complejos para obtener soluciones óptimas a un problema determinado. HS es una buena elección si lo que se busca es una poderosa herramienta de programación que obtenga un alto desempeño en la búsqueda de soluciones, un bajo consumo de tiempo, energía y costos computacionales. Es un algoritmo clave para la optimización general que permite adaptar su estructura y parámetros dependiendo del campo de aplicación.

El algoritmo HS frente a otras estrategias de optimización es exitoso porque emplea mecanismo que no requieren de un conocimiento complejo para que un programador o desarrollador lo entienda. En los algoritmos meta heurísticos (como lo es HS) se destacan dos características principales: la diversificación y la intensificación. También conocidos como exploración y explotación. La diversificación o exploración, es el proceso por el cual el algoritmo explora en muchos lugares y regiones del espacio de soluciones, tanto como sea posible de una manera eficiente y efectiva. Esta característica pretende que el algoritmo no caiga en óptimos locales sin tener que recorrer todo el espacio de búsqueda. En HS, la diversificación es controlada por la tase de ajuste de tono (PAR) y la aleatoriedad (1-HMCR), con lo cual se buscó refinar los valores obtenidos anteriormente, es decir encontrar nuevos valores cercanos a los mejores valores ya obtenidos. Esto asegura que las buenas soluciones locales ya obtenidas sean guardadas en memoria mientras que la aleatoriedad hace que el algoritmo explore el espacio de búsqueda local de manera efectiva. En cuanto a la característica de intensificación o exploración pretende aprovechar la historia y la experiencia del proceso de búsqueda, asegurando la velocidad de convergencia, cuando sea necesaria haciendo uso de la reducción de la aleatoriedad y limitando la diversificación. Este proceso consiste en volver a explorar zonas del espacio prometedoras (en las cuales se hallaron buenas soluciones), ya exploradas parcialmente. La intensificación en HS es representada por la tasa de consideración de la memoria armónica (HMCR). Un valor alto para la tasa de consideración de la memoria indica que se reutilizaran más valores almacenados en la memoria armónica. Si es demasiado bajo, la convergencia se tornará demasiado lenta. Teniendo en cuenta estas características, se puede observar que la interacción entre estos dos componentes asegura el éxito del algoritmo HS frente a otros algoritmos.

# Link del repositorio

<https://github.com/olvjudith/sistemas_embebidos.git>

**Bibliografía**

[El algoritmo de búsqueda armónica y sus usos en el procesamiento digital de imágenes](http://www.scielo.org.mx/pdf/cys/v17n4/v17n4a8.pdf)

[Harmony Search Method: Theory and Applications](https://www.hindawi.com/journals/cin/2015/258491/)

Zong Woo Geem, Joong Hoon Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. Simulation, 76(2):60–68, 2001

<https://revistas.unal.edu.co/index.php/avances/article/view/26727/27036>