



Instituto Tecnológico de Estudios Superiores de Monterrey

ACTIVIDAD:

“Proyecto Final; Primer Reporte Preliminar”

Integrantes:

- ❖ Omar David Hernández Aguirre | A01383543
- ❖ Edgar Mauricio Sánchez Marín | A01552403
- ❖ Manuel Alejandro López Arvayo | A01252726
- ❖ Carlos Alberto Arroyo Gonzalez | A00826156

Instructor: Iván Mauricio Amaya Contreras, Ph.D.

Jueves 24 de septiembre de 2020

Monterrey, Nuevo León

Primer Reporte Preliminar

“Repositorio de Github”:

<https://github.com/davidha99/OptimizacionModerna>

“Definiciones”:

- Optimización:

Capacidad de hacer o resolver alguna cosa de la manera más eficiente posible y, en el mejor de los casos, utilizando la menor cantidad de recursos.

- *Función objetivo:*

Es la función que será optimizada dados ciertos límites y variables en donde estas serán minimizadas o maximizadas usando técnicas de programación.

- *Metaheurística:*

Son estrategias generales para resolver problemas computacionales, usando parámetros dados por el usuario y procedimientos genéricos o abstractos. Van más allá de la heurística, por ello se le denomina metaheurística.

- *Algoritmo genético:*

Es una técnica que surgió en los años 70s con el fin de explicar los procesos de adaptación en sistemas naturales, así emular por medio de softwares un sistema natural con los mecanismos de los sistemas naturales. Sus elementos son: Esquema de codificación, Función de evaluación, operadores básicos y algunos parámetros.

- *Esquema de codificación*, o sea la manera en que se representa una posible solución al problema.
- *Función de evaluación*, que indica si un individuo es apto para resolver el problema planteado.
- Tres *operadores básicos*: reproducción, cruce y mutación.
- *Parámetros* que controlan el desempeño del algoritmo genético: probabilidad de cruce, probabilidad de mutación, tamaño de la población, número de generaciones, etc.

“Algoritmo genético”

- **Descripción**

Serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular.

- **Pseudocódigo**

La estructura base de un algoritmo genético es:

```
Comenzar
  t=0
  inicializar P(t)
  evaluar P(t)
  mientras (no condición de término) hacer:
    t=t+1
    seleccionar P(t) a partir de P(t-1)
    recombinar y/o mutar P(t)
    evaluar P(t)
  fin
fin
```

- **Explicación de sus operadores y parámetros, incluyendo sus rangos.**

t es el número de iteración.

P(t) es una población de individuos para la iteración t, donde cada individuo representa un punto de búsqueda en el espacio de las soluciones del problema.

“Metaheurísticas”

Incluir el proceso en el que se inspira e incluir una referencia a algún artículo o trabajo en donde se utilice (puede ser artículo original en donde se presentó el método).

Listado de al menos 4 metaheurísticas:

1. Metaheurística 1: VMO (Variable Mesh Optimization)
2. Metaheurística 2: ACO (Ant Colony Optimization)
3. Metaheurística 3: TSO (Taboo Search Optimization)
4. Metaheurística 4: EA (Evolutionary Algorithm)
5. Metaheurística 5: Hill Climbing
6. Metaheurística 6: SA (Simulated annealing)

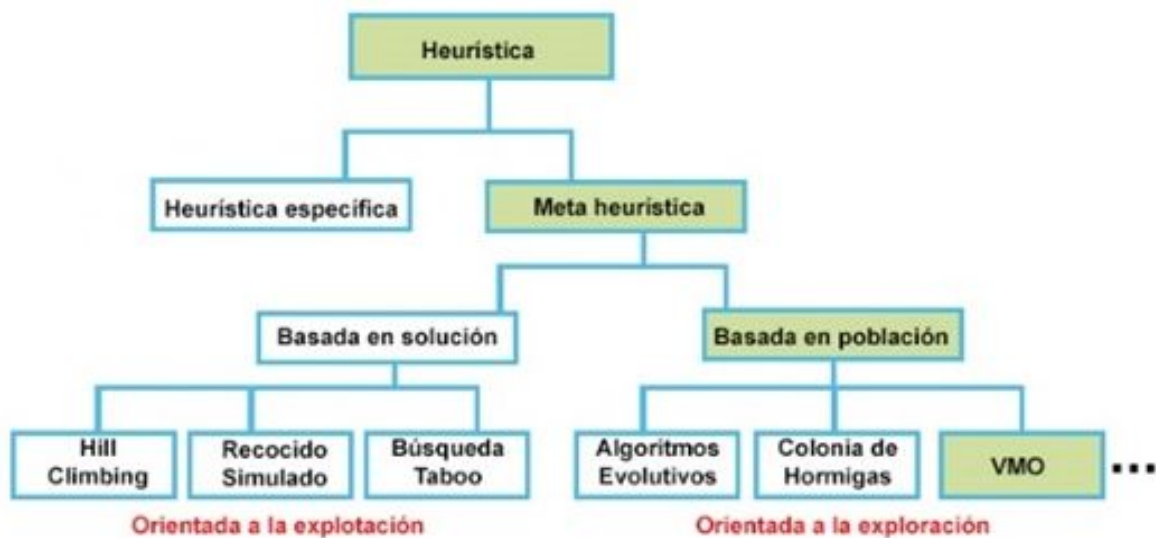


Figura 1. Taxonomía de los métodos heurísticos

Figura extraída de:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000300004

“Hill Climbing”

Similares a los algoritmos genéticos, aunque más sistemáticos y menos aleatorios. Hill climbing es una técnica que permite solo ir mejorando la solución por que aplica un mecanismo de restar o multistart tratando de mejorar la solución. En cada iteración, un nuevo punto es seleccionado de la vecindad del punto actual. Un algoritmo de ascenso a colina comienza con una solución al problema a mano, normalmente elegida al azar.

Es una técnica que permite solo ir mejorando una solución. Luego, la cadena se muta, y si la mutación proporciona una solución con mayor aptitud que la solución anterior, se conserva la nueva solución; en caso contrario, se conserva la solución actual. Si el nuevo punto es mejor, se transforma en el punto actual, sino otro punto vecino es seleccionado y evaluado. El método termina cuando no hay mejoras, o cuando alcanza un número predefinido de iteraciones. Luego el algoritmo se repite hasta que no se pueda encontrar una mutación, cuando no hay mejoras o cuando se alcanza un número predefinido de iteraciones y esta solución se devuelve como resultado.

El algoritmo de ascenso de colina es lo que se conoce como algoritmo voraz. Lo que significa que siempre hace la mejor elección disponible en cada paso, con la esperanza de que de esta manera se puede obtener el mejor resultado global. En contraste, los métodos como los algoritmos genéticos y el recocido simulado, discutido abajo, no son voraces; a veces, estos métodos hacen elecciones menos óptimas al principio con la esperanza de que conducirán hacia una solución mejor más adelante.

Se puede realizar la búsqueda de dos maneras diferentes:

Escala Simple:

- Se dirige a un estado mejor que el actual
- Función heurística de proximidad
- No se almacenan los estados anteriores
- Es un método local, sus movimientos están determinados por ser mejor al anterior.

Escala por máxima pendiente:

- Busca no solamente un estado mejor que el actual, si no el mejor de todos los estados posibles, es una máxima pendiente.

El Hill Climbing, tiene las siguientes características:

- Informado
- Utiliza información del estado para elegir un nodo u otro.
- No exhaustivo
- No explora todo el espacio de estados. Solo encuentra una solución. Encuentra buenas soluciones.
- Encuentra muy buenas soluciones, aunque muchas veces no es la mejor, puesto que no explora todos los estados.
- Es eficiente.
- Debido a que evita la exploración de una parte del espacio de estados.

Ventaja: Reduce el número de nodos a analizar

Desventaja: Puede ser que encuentre una solución, pero no es la más óptima.

“Recocido Simulado”

(Este algoritmo se ha aplicado con éxito en el campo de la optimización de estructuras de hormigón.)

El nombre e inspiración viene del proceso de recocido del acero y cerámicas, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. El calor causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía); el enfriamiento lento les da mayores probabilidades de recrystalizar en configuraciones con menor energía que la inicial (mínimo global).

En cada iteración, el método de recocido simulado evalúa algunos vecinos del estado actual s y probabilísticamente decide entre efectuar una transición a un nuevo estado s' o quedarse en el estado s . En el ejemplo de recocido de metales descrito arriba, el estado s se podría definir en función de la posición de todos los átomos del material en el momento actual; el desplazamiento de un átomo se consideraría como un estado vecino del primero en este ejemplo. Típicamente la comparación entre estados vecinos se repite hasta que se encuentre un estado óptimo que minimice la energía del sistema o hasta que se cumpla cierto tiempo computacional u otras condiciones.

El vecindario de un estado s está compuesto por todos los estados a los que se pueda llegar a partir de s mediante un cambio en la conformación del sistema. Se generan por el método de montecarlo.

Empieza en un estado s_0 y sigue hasta un máximo de k_{\max} pasos o hasta que se encuentra un estado con energía menor o igual que e_{\min} . La función vecino(s) genera aleatoriamente un vecino de un estado dado s ; la función $\text{azar}(0, 1)$ devuelve un valor aleatorio uniformemente distribuido en el intervalo $[0, 1]$; véase Distribución uniforme. El proceso de recocido (enfriado) se expresa mediante $\text{temperatura}(r)$, que da la temperatura en función de la fracción r del tiempo que ya ha transcurrido.

```
Sea  $s = s_0$ 
Para  $k = 0$  hasta  $k_{\max}$  (exclusive):
   $T \leftarrow \text{temperatura}(k/k_{\max})$ 
   $s_{\text{nue}} \leftarrow \text{vecino}(s)$ 
  Si  $P(E(s), E(s_{\text{nue}}), T) \geq \text{azar}(0, 1)$ :
     $s \leftarrow s_{\text{nue}}$ 
Salida: estado final  $s$ 
```

La probabilidad de aceptación originalmente propuesta es

- si $e' < e$, entonces $P(e, e', T) = 1$
- si $e' > e$, entonces $P(e, e', T) = \exp(-(e' - e)/T)$

“Búsqueda Tabú”

La búsqueda tabú fue desarrollada por Fred Glover en la década de los 80. La cual se basa en la premisa: “Una mala decisión tomada usando una estrategia produce más información que una buena decisión aleatoria”. Es una metaheurística que gerencia un método heurístico de búsqueda local en vecindad. Es una búsqueda de trayectoria, es decir, genera un camino hacia la mejor solución.

Una búsqueda local se puede ver como un proceso iterativo que empieza en una solución y la mejora realizando modificaciones locales.

Básicamente empieza con una solución inicial y busca en su vecindad por una mejor solución. Si la encuentra, reemplazar su solución actual por la nueva y continua con el proceso, hasta que no se pueda mejorar la solución actual.

```
Procedimiento Búsqueda Local
s = genera una solución inicial
while s no es óptimo local do
    s' ∈ N(s) con f(s) < f(s')
    (solución mejor dentro de la vecindad de s)
    s ← s'
end
return s
```

Una característica de la búsqueda tabú es que si no encuentra una solución vecina de mayor calidad de la que se tiene actualmente, pasa a la que degrada menos la función objetivo, haciéndola un método difícil de parar o estancar. Así mismo, la vecindad que maneja es dinámica, debido a que se disminuye la vecindad mediante atributos tabú (prohibidos) que no se toman en cuenta.

Una herramienta esencial de la búsqueda tabú es la exploración sensible. Básicamente, se tienen índices de sensibilidad que se conjugan con memoria adaptativa con el objetivo de evitar regresar a subespacios o soluciones ya exploradas.

1. Vecindad dinámica
2. Vecindad adaptativa
 - a. Memoria a corto plazo (prohibiciones temporales / Atributos Tabú)
 - b. Memoria a largo plazo (frecuencia, intensificación, diversificación)
3. Reencadenamiento de cadenas
4. Criterio de aceptación
5. Oscilación estratégica

“Colonia de Hormigas”

Se inspira en el comportamiento de los insectos y en la premisa: “Individuos con capacidades limitadas pueden, en grupo, resolver problemas complejos”. Un problema es que la adaptación a la solución de problemas de optimización no es trivial.

Las hormigas utilizan las feromonas para resolver problemas de trasladarse en la mejor ruta. Cada vez que una hormiga encuentra comida, cuando encuentra la línea recta a la colonia, recorre la misma ruta dejando feromonas a su paso. El objetivo es dejar el mayor número de feromonas en la ruta más óptima. Si no se tiene la mejor ruta, las hormigas toman caminos aleatorios depositan feromonas al pasar, la ruta más corta (es decir, la más óptima) se llena de feromonas más rápido y se solidifica a comparación de las otras, que se desvanecen con el tiempo.

Una manera de aplicar esta metaheurística es por ejemplo el algoritmo del agente viajero (véase aquí: <https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html>), en que se dispone de una hormiga en cada ciudad. Luego cada hormiga selecciona una ciudad no visitada anteriormente. Tras completar n pasos, las hormigas completan las rutas. Cada vez que se completa un arco, se deposita la feromona para su posterior optimización. La clave está en 2 aspectos, cómo se elige la ciudad y el criterio para depositar la feromona.

• ¿Cómo se elige la ciudad?

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in \mathcal{N}_i^k} \tau_{il}^\alpha \cdot \eta_{il}^\beta} \quad \text{si } j \in \mathcal{N}_i^k$$

- τ_{ij} es la feromona depositada entre las ciudades (i, j)
- η_{ij} es el valor del heurístico entre las ciudades (i, j)
- \mathcal{N}_i^k ciudades a las que puede ir la hormiga k -ésima que se encuentra en la ciudad i -ésima
- α y β son parámetros del algoritmo

Modificación de la feromona

- Disipación de la feromona $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad (0 < \rho < 1)$
- Incremento de feromona:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad \forall (i, j)$$

donde

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C^k} & \text{si el arco } (i, j) \text{ está en } T^k \\ 0 & \text{en otro caso} \end{cases}$$

siendo T^k la ruta de la hormiga k -ésima

“Algoritmos Evolutivos”

Los algoritmos evolutivos están inspirados en la teoría de la evolución de Darwin. Estos surgieron por la simulación de procesos de evolución natural y como resultado de esto es la optimización escolástica. En estos algoritmos, se tiene un conjunto de posibles soluciones en donde se mezclan y compiten entre sí, de manera que las más aptas son capaces de permanecer (sobrevivir) a lo largo del tiempo y van evolucionando a mejores soluciones.

“VMO (Variable Mesh Optimization)”

Es una meta-heurística poblacional con características evolutivas donde un conjunto de nodos que representan soluciones potenciales a un problema de optimización, forman una malla (población) que se expande y contrae dinámicamente, desplazándose por el espacio de búsqueda. La formulación general de esta meta-heurística abarca tanto los problemas de optimización continuos como los discretos (Cáceres, 2009).

Esta meta-heurística consta esencialmente de dos etapas, expansión y contracción. En la primera etapa se realiza un proceso de expansión en cada ciclo, donde se generan nuevos nodos en dirección a los extremos locales (nodos de la malla con mejor calidad en distintas vecindades) y el extremo global (nodo obtenido de mejor calidad en todo el proceso desarrollado), así como a partir de los nodos fronteras de la malla (nodos de mayor y menor norma). Luego en la última etapa, se realiza un proceso de contracción de la malla, donde los mejores nodos resultantes en cada iteración son seleccionados como malla inicial para la iteración siguiente.

“Metaheurística seleccionada:”

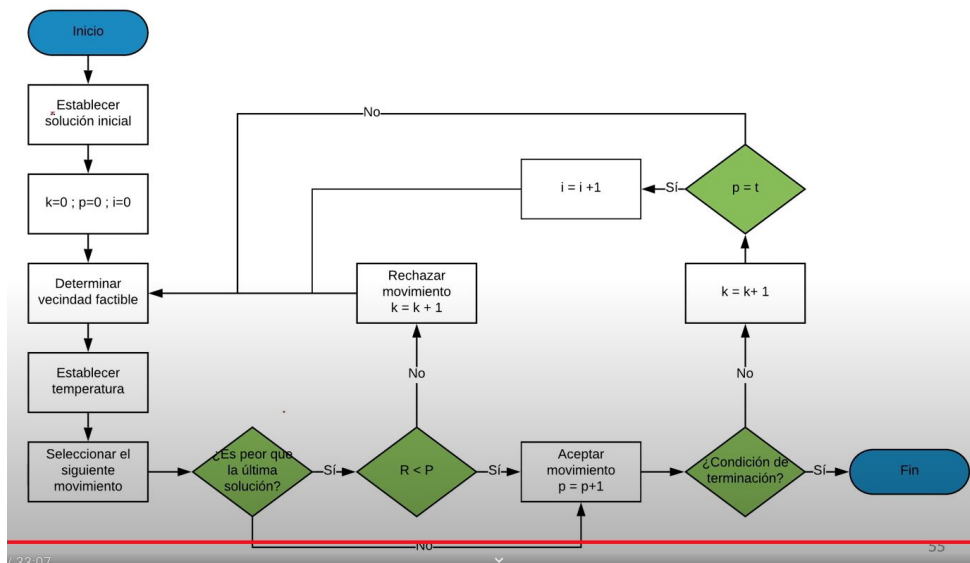
Recocido Simulado

“Descripción”

Es un algoritmo que busca la imitación de un proceso físico químico. Busca salir del óptimo local aceptando y rechazando movimientos de no mejora sujetos a la probabilidad. A medida que la cantidad de iteraciones aumenta, el método busca hacer una selección más acertada a través del parámetro $\langle T \rangle$ (temperatura) empezando en un valor muy alto y haciéndose progresivamente más pequeño.

“Pseudocódigo”

5. Recocido Simulado



“Explicación de sus operadores y parámetros”

- $\langle T \rangle$ Temperatura. Este parámetro empezará en un valor muy elevado $\langle T_i \rangle$ y empezará a disminuir conforme a sus resultados de optimización.
- $\langle P \rangle$ Probabilidad.
- $\langle s \rangle$ Solución
- $\langle r \rangle$ Factor resultante
- $\langle N \rangle$ Vecindad (factores anteriormente aceptados)

“Justificación de por qué esta metaheurística”

Las tres metaheurísticas que más llamaron nuestra atención fueron la de VMO, Tabú y Recocido simulado. Tabú la descartamos desde un principio porque no sigue un algoritmo definido como tal. VMO la descartamos debido a que nos fue difícil recabar información sobre este método. Debido a que es más reciente no es tan utilizado a comparación del Recocido simulado.

“Diagrama de Gantt”

"Semana "	Selección de equipos (10 min.).	Creación del documento compartido y el relleno con su esqueleto/formato (15 min.).	Primera junta; distribución de definiciones entre los miembros del equipo y revisión de estas entre los miembros del equipo (45 min.).	Segunda junta; Búsqueda metaheurísticas y distribución de las investigaciones (30 min.).	Tercera junta; Revisión de las Metaheurísticas (20 min.).	Discusión y selección de la Metaheurística a trabajar (10 min.).	Revisión del documento compartido y corrección.
Ago; 17-23							
Ago; 24-30							
Sep; 31-6							
Sep; 7--13							
Sep; 14-20							
Sep; 21-24							

“Tareas Realizadas”

1. Primer reporte preliminar: realizar actividades para el primer reporte. (Involucrado/s: Todo el equipo) [Responsable: Omar].
2. Repositorio de Github: crear el repositorio para el proyecto en Github. (Involucrado/s: Omar) [Responsable: Omar].
3. Definiciones: definir los conceptos de optimización, función objetivo, metaheurística y algoritmo genético. (Involucrado/s: Todo el equipo) [Responsable: Mauricio].
4. Algoritmo genético: describir un algoritmo genético e incluir su pseudocódigo, así como explicar sus operadores y parámetros. (Involucrado/s: Todo el equipo) [Responsable: Manuel].
5. Metaheurísticas: hacer listado de metaheurísticas. (Involucrado/s: Todo el equipo) [Responsable: Manuel].
6. Metaheurística seleccionada: describir la metaheurística seleccionada. (Involucrado/s: Todo el equipo) [Responsable: Todo el equipo],
7. Diagrama de Gantt: realizar diagrama de Gantt organizado por semanas. (Involucrado/s: TODOS) [Responsable: Carlos].

“Referencias”

- Alan Mora. Función objetivo. (2020). Retrieved September 25, 2020, from prezi.com website:
<https://prezi.com/ql3aagwj9-ve/funcion-objetivo/#:~:text=La%20funci%C3%B3n%20o%20objetivo%20es%20la,programaci%C3%B3n%20lineal%20o%20no%20lineal>
- Alejandro Fuentes. (2013). Problema del agente viajero. Retrieved September 25, 2020, from Uaeh.edu.mx website:
<https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html>
- Monardes, C. (2020). Clase 03, Capítulo 4, Metaheurísticas [YouTube Video]. Retrieved from <https://www.youtube.com/watch?v=taIGbP2o8cQ&t=460s>
- MANUEL CENTENO ROMERO, & VELÁSQUEZ, R. (2016). UN ALGORITMO METAHEURÍSTICO DE RECOCIDO SIMULADO PARA EL 3AP-AXIAL. SABER. Revista Multidisciplinaria Del Consejo de Investigación de La Universidad de Oriente, 28(3), 566–573. Retrieved from <https://www.redalyc.org/jatsRepo/4277/427750771013/html/index.html>
- EcuRed. Algoritmo genético (2020). Retrieved September 25, 2020, from Ecured.cu website: https://www.ecured.cu/Algoritmo_gen%C3%A9tico
- Significados. (2019, May 13). Significado de Optimización. Retrieved September 25, 2020, from Significados website: <https://www.significados.com/optimizacion/>
- Daniel Molina Cabrera. (2013, October). Algoritmo NVMO para problemas multimodales. Retrieved September 25, 2020, from Slideshare.net website: https://es.slideshare.net/dmolina_/algoritmo-nvmo-para-problemas-multimodales
- Pérez, J. A. M. (2004). Metaheurísticas: Concepto y Propiedades. *Santa Cruz de Tenerife*.
- Díaz López, Ernesto, Martínez Prieto, Adrian, & Gálvez Lio, Daniel. (2016). Una implementación de la meta-heurística “Optimización en Mallas Variables” en la arquitectura CUDA. *Revista Cubana de Ciencias Informáticas*, 10(3), 42-56. Recuperado en 24 de septiembre de 2020, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000300004&lng=es&tlng=es.
- Valencia, P. E. (1997, August). Optimización mediante algoritmos genéticos. In *Anales del Instituto de Ingenieros de Chile* (Vol. 109, No. 2, pp. 83-92).
- Anderson Benavides, Y.S, Hernández Ribota, P.S, Ramírez Saavedra, A., Reyes Chavez, K.R & Velásquez Flores, E. (2005). Algoritmo Hill Climbing. Retrieved September 20, 2020, from [http://www.itnuevolaredo.edu.mx/takeyas/apuntes/InteligenciaArtificial/Apuntes/tareas_alumnos/Hill_Climbing/Hill_Climbing\(2005-II-B\).pdf](http://www.itnuevolaredo.edu.mx/takeyas/apuntes/InteligenciaArtificial/Apuntes/tareas_alumnos/Hill_Climbing/Hill_Climbing(2005-II-B).pdf)
- Lozano, J.A (2015). Búsqueda Tabu [Youtube video] retrieved from <https://youtu.be/NGo9P33aEV8>
- Glover, F., & Melián, B. (2003). Búsqueda tabú. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 7(19), 0.
- Lozano, J.A (2015). Colonias de Hormigas [Youtube video] retrieved from <https://youtu.be/H3cl6Ytu0cl?t=310>