

# Metaheurísticas I

**Estrategia:** La materia no es muy teórica, es bueno tener apuntes de lo que se realiza a lo largo del curso y entender los procedimientos, sin embargo es más importante la aplicación de los algoritmos. Buscar y entender los algoritmos vistos en clase será de gran importancia al momento de la evaluación

## ▼ Primer Parcial

### ▼ 09-08-2021

Profesor: Francisco Javier Luna Rosas

Cambio en los criterios de evaluación:

#### NUEVOS CRITERIOS

- Dos exámenes parciales y uno final que representan un 20% del curso cada uno. Este porcentaje será evaluado considerando el examen escrito, participación en clases, realización de prácticas y tareas (60%).
- Proyecto final que representa el 20% de la calificación final. La entrega del proyecto final es el derecho a la presentación del último examen.
- Actividades y Exposiciones 20%

## Introducción al Big Data

Nueva estrategia indispensable para las empresas en cualquier sector

## 5. 3 V's (1/2).

### El significado de Big Data - 3 V's

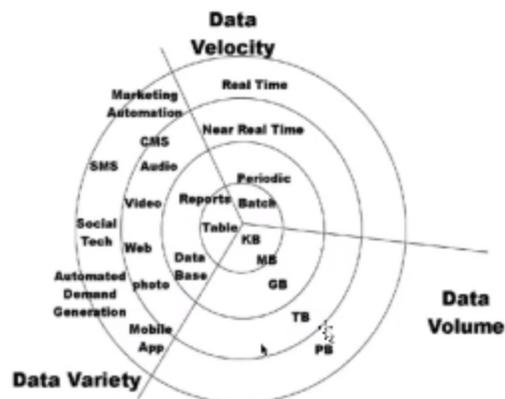
- Big Volume
- Big Velocity
- Big Variety



11

## 3 V's (2/2).

### El significado de Big Data - 3 V's



12

## Variedad (1/1)

En la actualidad, los datos se encuentran en todo tipo de formatos:

- Desde las bases de datos tradicionales hasta los almacenes de datos jerárquicos creados por los usuarios finales, pasando por los sistemas OLAP, los documentos de texto, el email, los datos de mediciones, el video, el audio, la información bursátil y las transacciones financieras, entre muchos otros.
- Según algunos cálculos, el 80% de los datos de las organizaciones no son numéricos. No obstante, estos también deben incluirse en los análisis y el proceso de toma de decisiones.



14

## 7. Plataforma de Código Abierto (1/2)

### “Hadoop”

- Open-source software framework de Apache
- Inspirada en:
  - Google Map-Reduce
  - GFS (Google File System)



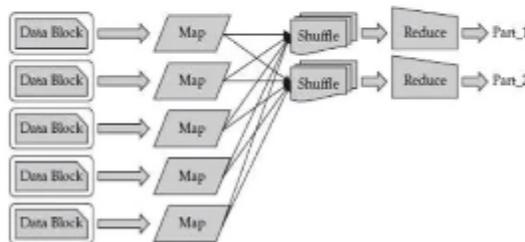
- HDFS
- Map/Reduce



23

## Hadoop Map-Reduce(2/2)

- La siguiente Figura ejemplifica un flujo de datos en un proceso sencillo de Map-Reduce:



28

## Características de «Big Data» (1/1)

¿Qué tan grande es Big Data?

**Lo que es grande hoy en día  
tal vez no lo sea mañana**

Memory unit	Size	Binary size
kilobyte (kB/KB)	$10^3$	$2^{10}$
megabyte (MB)	$10^6$	$2^{20}$
gigabyte (GB)	$10^9$	$2^{30}$
terabyte (TB)	$10^{12}$	$2^{40}$
petabyte (PB)	$10^{15}$	$2^{50}$
exabyte (EB)	$10^{18}$	$2^{60}$
zettabyte (ZB)	$10^{21}$	$2^{70}$
yottabyte (YB)	$10^{24}$	$2^{80}$

21

Los conocimientos de Aprendizaje Inteligente nos sirven para darle veracidad. Las metaheurísticas nos sirven para darle velocidad y el volumen lo obtenemos de internet

▼ 10-08-2021

Examen diagnóstico

▼ 11-08-2021

# Proyecto

Optimizando el tiempo de respuesta global en la detección de patrones relacionados con el suicidio en redes sociales

Buscar el tema en específico de detección de patrones relacionados con el suicidio

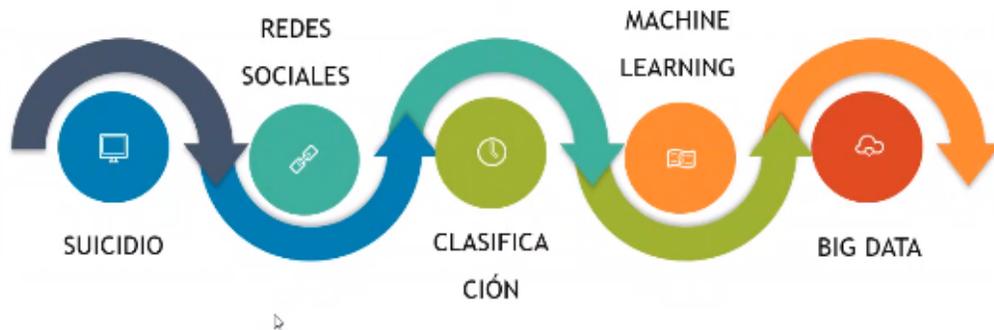
- Estrés
- Cyberbullying
- Estrés laboral
- Alcoholismo

Todo esto en redes sociales

Optimizar tiempo de respuesta

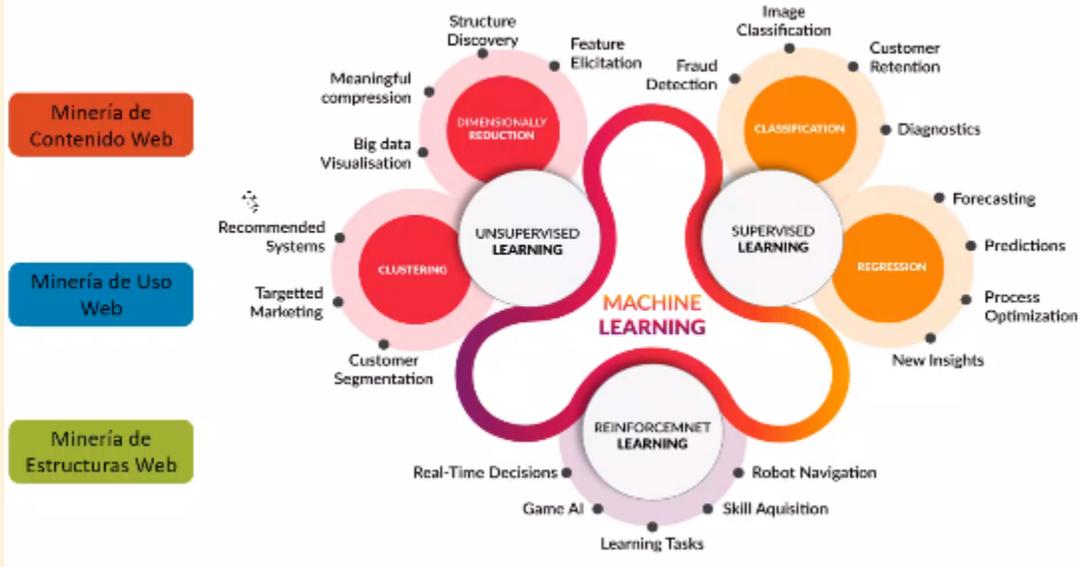
- Máquina de aprendizaje distribuida o en paralelo para hacerlo

## MARCO TEÓRICO



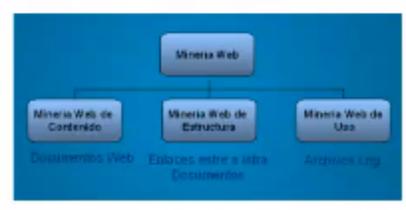


# MINERIA DE LA WEB



## Tipos de minería de la Web

1. Minería de los registros de navegación en la Web (Web Usage Mining).
2. Minería de la estructura de la Web (Web Structure Mining).
3. Minería del contenido de la Web (Web Content Mining).



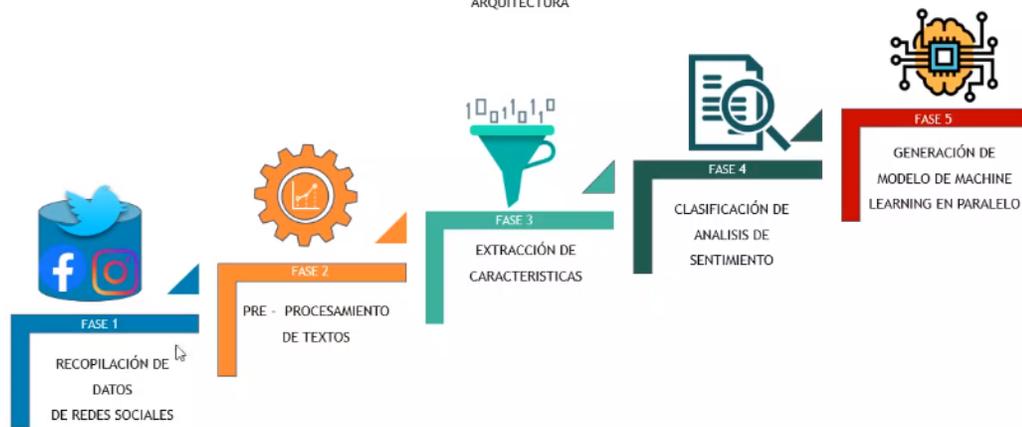
12

Se tratará Procesamiento del Lenguaje Natural

Trabajar los tweets, obtener la información de ellos, darles significado con base en el problema que estamos trabajando (suicidio pero nosotros elegiremos otro problema)

## Metodología

## METODOLOGÍA ARQUITECTURA



▼ 12-08-2021

## Formación del Equipo 10: Los Vengadores

- Meli
- Joul
- Óscar
- Dari
- Fer
- Efra

El racismo

▼ 13-08-2021

Dentro de la grabación de la clase en el minuto 15 a 20 aprox es donde está todo explicado del código

Al minuto 25 habla a fondo de todo el proyecto

▼ 16-08-2021

## Unidad 1: Antecedentes

Optimización: Buscar la mejor solución a un problema

Optimizar sirve para mejorar el funcionamiento o desarrollo de un proyecto o servicio. Se resuelven ciertos problemas y se logra mejorar el proceso

Los pasos son los siguientes:

- Formular el problema
- Modelar el problema
- Optimizar el problema
- Implementar una solución

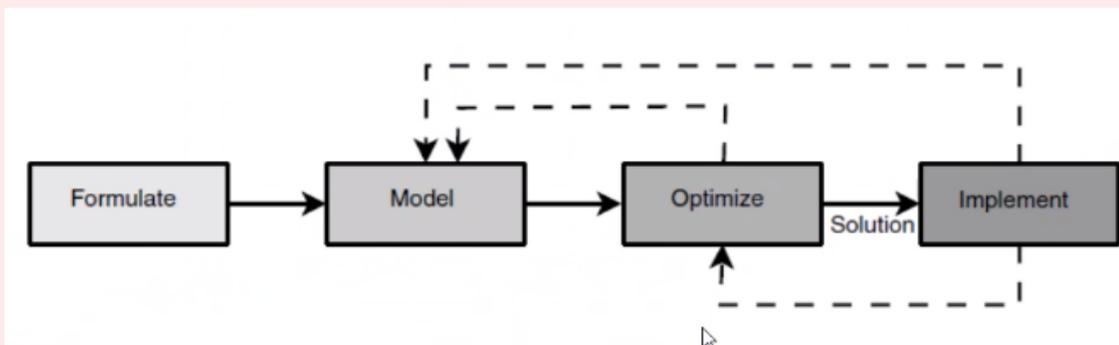


FIGURE 1.1 The classical process in decision making: formulate, model, solve, and implement.

Los métodos pueden ser iterativos hasta llegar a una solución adecuada

Las dificultades de la optimización se encuentran que al modelar un problema no se sabe si el óptimo es local o global

Heurístico: Proceso que no está bien establecido o bien definido

Por equipo hacer un escrito de qué son heurísticas, metaheurísticas, optimización, algoritmos simples, optimización lineal, combinatoria, etc.

▼ 17-08-2021

## Unidad 2: Problemas de Optimización

Cada posible solución es una heurística

Todo junto sería una metaheurística

Las bases del algoritmo genético clásico para optimizar una población binaria de 6 dígitos y tratar de que todos sean 1

### ▼ 18-08-2021

Explicación final del genético

En la grabación de la clase hay también cuestiones del proyecto

Tarea: Programar el algoritmo de la ruleta y el genético con 50, 100 y 200 cromosomas

### ▼ 19-08-2021

Un poco de algoritmo genético

Un poco de la tarea de optimización

Un poco del proyecto

### ▼ 20-08-2021

No hubo clases

### ▼ 23-08-2021

Continuamos con el genético con la probabilidad de cruzamiento

### ▼ 24-08-2021

Programar el algoritmo genético

Vistas al algoritmo de descenso de gradiente

### ▼ 25-08-2021

## Descenso de gradiente

## El Algoritmo del Descenso del Gradiente

1. Tome un punto al azar  $x_0$ .
2. Calcule el valor de la pendiente (derivada)  $f'(x_0)$ .
3. Camine en dirección opuesta a la pendiente:

$$x_1 = x_0 - \eta * f'(x_0)$$

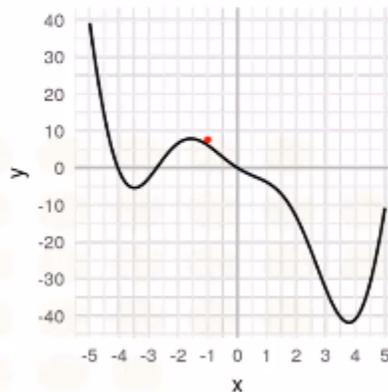
- Aquí,  $\eta$  es la tasa de aprendizaje que mencionamos anteriormente. Y el signo menos nos permite ir en la dirección opuesta.

5

Estamos viéndolo con el ejemplo:

## El Algoritmo del Descenso del Gradiente

- **Paso 1:** Tome al azar  $x_0 = -1$ . Esto nos da  $f(x_0) = 6.08$ :



7

## El Algoritmo del Descenso del Gradiente

- **Paso 2:** Calculamos la derivada (pendiente), note que  $f'(x) = 4x\cos(x) - 2x^2\sin(x) - 5$ , de donde la pendiente es  $f'(x_0) = -5.478$  (pendiente negativa).
- **Paso 3:** Si tomamos  $\eta = 0.05$  entonces:
  - $x_1 = x_0 - \eta * f'(x_0) = -0.726$ ,
  - Así se tiene que  $f(x_1) = 4.419$ .
  - Graficamos el par  $(x_1, f(x_1))$  como sigue:

8

### ▼ 26-08-2021

Revisión de entregas Fase 1 del proyecto

### ▼ 27-08-2021

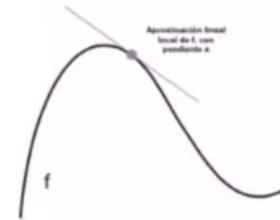
Revisión de la fase 1 del proyecto

### ▼ 30-08-2021

Continuación del Descenso del Gradiente

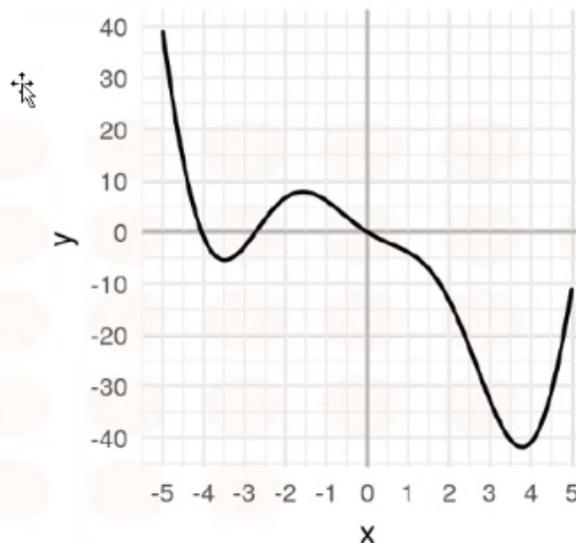
Será muy importante revisar Descenso del Gradiente

- Está pendiente es la derivada de  $f$  en  $p$ . Si  $a$  es *negativa*, esto quiere decir que un pequeño cambio en  $x$  alrededor de  $p$  resultaría en una *disminución* de  $f(x)$ . Si por otra parte,  $a$  fuera *positiva*, un pequeño cambio en  $x$  resultaría en un *aumento* de  $f(x)$ .
- Además, el valor absoluto de  $a$  (la magnitud de la derivada) indicará qué tan rápido ocurrirá este aumento o disminución.
- Cabe resaltar, que la aproximación es válida solo cuando  $x$  está lo suficientemente cerca de  $p$ .



## Ejemplo:

- Sea  $f(x) = 2x^2 \cos(x) - 5x$  en el intervalo  $[-5, 5]$



1. Tome un punto al azar  $x_0$ .
2. Calcule el valor de la pendiente (derivada)  $f'(x_0)$ .
3. Camine en dirección opuesta a la pendiente:

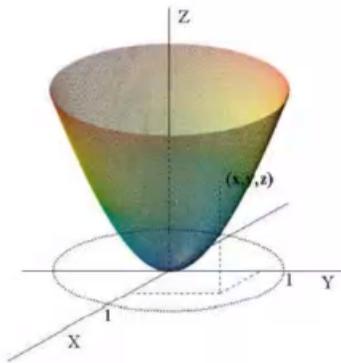
$$x_1 = x_0 - \eta * f'(x_0)$$

- Aquí,  $\eta$  es la tasa de aprendizaje que mencionamos anteriormente. Y el signo menos nos permite ir en la dirección opuesta.

Usando varias variables:

## Ejemplo en 3D

### Derivada de una operación tensorial: El Gradiente



Tome ahora el ejemplo de la función  $f(x,y) = x^2 + y^2$  y vea que su gradiente es igual a:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Para comprender esto mejor, veámoslo en el plano  $xy$  con radio igual a 1.

15



## Ejemplo en 3D

### Formalmente:

- ¿Qué es el gradiente?
- Dada una función  $f: D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciable en  $\mathbf{x}$  en  $D$ , se define el *gradiente* como el vector
$$\nabla f(\mathbf{x}) := \left( \frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right).$$
- Es un vector normal al hiperplano tangente en  $\mathbf{x}$  la hipersuperficie de  $f(\mathbf{x}) = 0$ .
- Apunta en la dirección de máximo crecimiento de la función  $f$ .

16



## Ejemplo en 3D

---

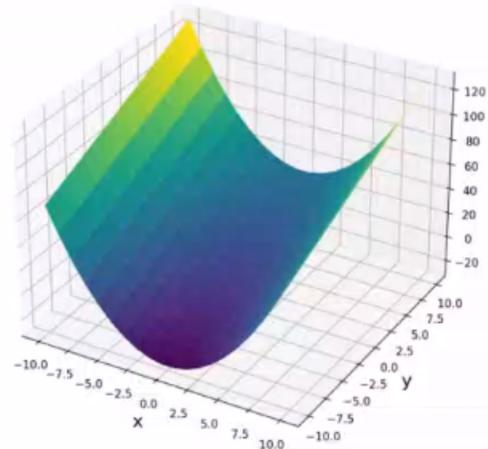
1. Tome un punto al azar  $x_0$ .
2. Calcule el valor de la pendiente (derivada)  $f'(x_0)$ .
3. Camine en dirección opuesta a la pendiente:

$$x_1 = x_0 - \eta * f'(x_0)$$

- Aquí,  $\eta$  es la tasa de aprendizaje que mencionamos anteriormente. Y el signo menos nos permite ir en la dirección opuesta.

## Ejemplo en 3D

$$f(x, y) = x^2 + 3y$$



18

## Ejemplo en 3D

Si  $f(x,y) = x^2 + 3y$ , entonces el  $\nabla f$  es el siguiente:

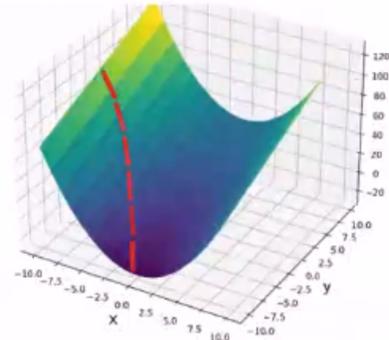
$$\nabla f = \begin{bmatrix} 2x \\ 3 \end{bmatrix}$$

Sea  $\mathbf{x}_0 = \begin{bmatrix} -10 \\ 10 \end{bmatrix}$  entonces  $f(\mathbf{x}_0) = 100 + 30 = \mathbf{130}$ .

Tomamos  $\eta = 0.1$  entonces:

$$\mathbf{x}_1 = \begin{bmatrix} -10 \\ 10 \end{bmatrix} - 0.1 * \begin{bmatrix} -20 \\ 3 \end{bmatrix} = \begin{bmatrix} -8 \\ 9.7 \end{bmatrix}$$

entonces  $f(\mathbf{x}_1) = 64 + 29.1 = \mathbf{93.1}$ .



19

▼ 31-08-2021

Granja de servidores

## Optimizar carga usando metaheurísticas

Principal:

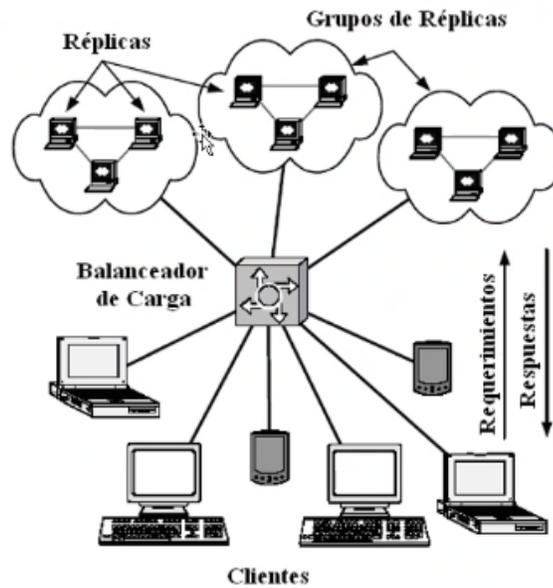
1. Proponer una nueva estrategia para optimizar carga.

Secundarios:

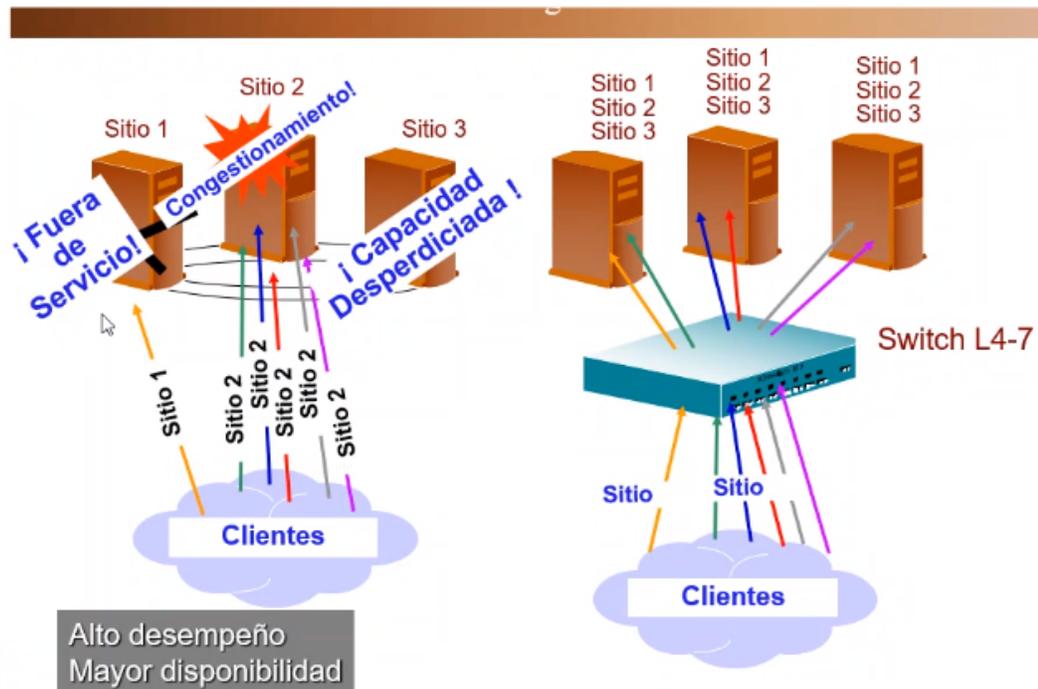
1. Proponer una arquitectura de optimización de carga robusta que soporte diferentes meta-heurísticas.
2. Determinar como diseñar una arquitectura de optimización de carga.

- Réplica
- Grupo de Replicas.
- Sesión
- Requerimiento
- Planificador de Carga.

# Conceptos Claves del Balanceo de Carga



# ¿Porque Balancear Carga?



## ▼ 01-09-2021

Gradiente descendente encuentra mínimos locales

Para ello pueden usarse algoritmos genéticos

## ▼ 02-09-2021

Continuación de lectura en AGS

## ▼ 03-09-2021

Revisión de los Algoritmos Genéticos nuestros

## ▼ 06-09-2021

Examen

Función

$$f(x) = \frac{x}{1+x^2}$$

$$L = \lceil \log_2 (1 + (X_{\max} - X_{\min} / \text{PREC})) \rceil$$

$$\text{PREC} = 0.0001$$

$$L = \lceil \log_2(1 + (20 - 0 / 0.0001)) \rceil = 18 \text{ (longitud del cromosoma)}$$

#### ▼ 07-09-2021

Cambios de fechas

## Problema del Agente Viajero



**1b. Problema del agente viajero(1/1)**

Algunas veces una optimización involucra ordenar una lista o poner las cosas en un orden correcto. El Problema más famoso es el del vendedor viajero, en el cuál un vendedor quiere visitar C ciudades viajando lo menos posible en distancia. Cada ciudad es visitada y la solución consiste en una lista de ciudades y en el orden que serán visitadas

23

#### ▼ 08-09-2021

Exposición en Word de ejemplo del Problema del Agente Viajero

#### ▼ 09-09-2021

Continuación del algoritmo

▼ **10-09-2021**

Continuación del Order Crossover del TSP

▼ **13-09-2021**

Continuación

▼ **14-09-2021**

Continuación

Explicación de tareas como la de arquitectura de procesamiento multihilos

▼ **15-09-2021**

Problema de la mochila para que lo exponga el equipo 3 (Team de Vinicio)

Último día de entrega del examen, por tanto, fin del parcial

▼ **Segundo Parcial**

▼ **20-09-2021**

Continuación de la lectura del algoritmo de balance de carga

▼ **21-09-2021**

Continuación de la lectura del algoritmo de balance de carga

▼ **22-09-2021**

Revisión final del artículo

Más proyección del proyecto

▼ **23-09-2021**

Continuación con el proyecto

▼ **24-09-2021**

No hubo clase

▼ **27-09-2021**

## ▼ Problema de la Mochila (Team de Vinicio)

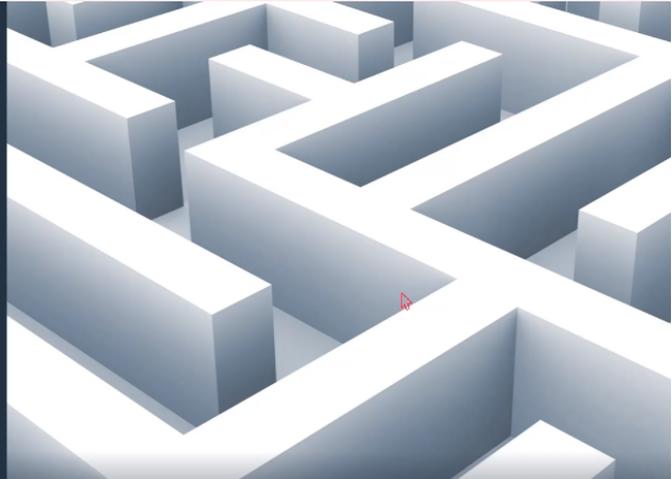
# Problema de la Mochila

Gustavo Benjamin II Pedraza Morales

Paulo Vinicio Martínez Roque

Leonardo de Jesús Ramírez Durón

Roberto Sandoval Torres



## Recordemos

- El problema de la mochila, comúnmente abreviado KP (del inglés Knapsack problem) es un problema de optimización combinatoria, es decir, que busca la mejor solución entre un conjunto finito de posibles soluciones a un problema. Modela una situación análoga al 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 y valor específicos. Los objetos colocados en la mochila deben maximizar el valor total sin exceder el peso máximo.

El algoritmo base, como se presenta en la descripción, está orientado hacia un conjunto finito de artículos que tienen un peso específico y que deberán guardados en un contenedor (mochila) con una capacidad limitada, teniendo como función objetivo, el minimizar el espacio utilizado. Las formulas 1 – 3 presentan el algoritmo básico del problema de la mochila (KP)

$$\text{maximize } \sum_{j=1}^n p_j x_j \quad (1)$$

$$\text{subject to } \sum_{j=1}^n w_j x_j \leq c, \quad (2)$$

$$x_j \in \{0,1\}, j = 1, \dots, n. \quad (3)$$

Donde:

$x_j$  -> Variables de decisión

$w_j$  -> Peso  $w$  del item  $j$

$c$  -> Capacidad total del contenedor (mochila)

$n$  -> número de items

La fórmula 1 hace referencia a maximizar los resultados del proyecto a partir de la integración de múltiples variables que pertenecen al proyecto actual identificado por el subíndice  $j$ . La fórmula 2 indica que es necesario estimar el peso total de los artículos que serán guardados en un contenedor cuya capacidad es determinada por la variable  $c$ . La fórmula 3 indica que las variables de decisión pertenecen al proyecto identificado por el valor  $j$ , es decir, forman parte del proyecto (valor 1) o no forman parte de dicho proyecto (valor 0).

# Restricciones

La única restricción que tenemos es que el peso de la carga transportada no puede exceder la capacidad máxima del carguero.

## Soluciones del Problema

- Técnica de programación dinámica, empleando cuatro tipos de visualización: árbol de recursión, grafo de dependencia, tabla de valores y tabla de decisiones.
- Metaheurísticas parametrizadas, mediante la combinación de parámetros.
- Ramificación y acotación como un método orientado a la solución de Problemas de Optimización Combinatoria buscando reducir el número de soluciones factible mediante la exploración sistemática del área de soluciones (árbol de soluciones), eliminando las soluciones que no son mejores que la solución actual, eliminando la rama correspondiente y los subnodos y hojas que dependen de ésta.

## Implementación de Metaheurísticas

- Algoritmos Evolutivos.
- Optimización basada en colonia de hormigas.
- Algoritmos basados en Trayectoria.

# Algoritmos evolutivos

Los AE están basados en la evolución biológica, incluyendo elementos como la reproducción, la mutación, la recombinación y la selección. Se trata de métodos de optimización y búsqueda estocásticos inspirados en la teoría de la evolución de Darwin. Estos algoritmos a menudo realizan soluciones aproximadas y utilizan la evolución simulada para explorar soluciones en problemas complejos del mundo real. Los AE son una herramienta muy popular para buscar, optimizar y proporcionar soluciones a problemas complejos.

## Optimización basada en colonia de hormigas.

- La OCH, también conocida como ACO por las siglas en inglés de Ant Colony Optimization, es una técnica que pertenece a los AIC y fue introducida como herramienta para la solución de problemas complejos. Esta técnica es utilizada principalmente para solucionar problemas que buscan los mejores caminos o rutas en grafos, aunque puede adaptarse en general para cualquier problema de optimización. Las hormigas se comunican a través de sus feromonas, las cuales son sustancias que les permiten encontrar los caminos más cortos entre su nido y la fuente de alimentos. OCH es una metaheurística basada en el comportamiento real de este insecto.

# Algoritmos basados en Trayectoria.

Los AT, también conocidos como TBM por las siglas en inglés de Trajectory-based Metaheuristics inician con una solución y buscan en el espacio de soluciones candidatas (el espacio de búsqueda) por una mejor solución. Si la encuentran, reemplaza su solución actual por la nueva y continúa con el proceso hasta que se encuentre una solución óptima. Se caracterizan por partir de un punto específico para mejora continua de la solución actual mediante la inspección de un vecindario. En general, la búsqueda finaliza cuando se alcanza un número máximo de iteraciones y se encuentra una solución con una calidad aceptable, o se detecta un estancamiento del proceso.



FIN

## Otras exposiciones

1. Team 1: Búsqueda AVARA o Breadth First
2. Team 2: Hill Climbing

3. Team 4: Recocido Simulado

4. Team 5: Búsqueda tabú

Revisión de arquitectura cliente - servidor de los equipos

▼ 28-09-2021

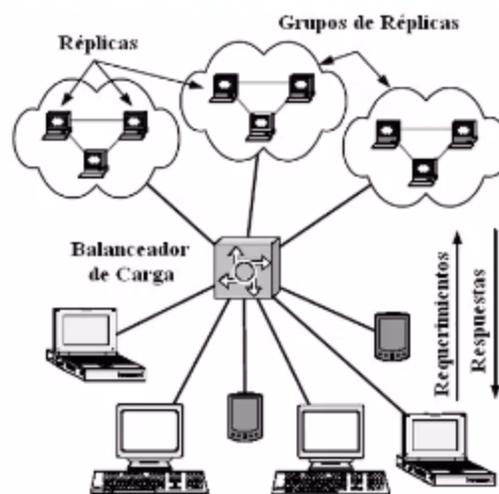
Revisión de arquitecturas

▼ 29-09-2021

Fin de revisión de arquitecturas

▼ 30-09-2021

## Conceptos Claves del Balanceo de Carga



Otros algoritmos

Breadth First Algorithm

▼ 01-10-2021

No hubo clase

▼ 04-10-2021

Aclaraciones sobre los entregables futuros

▼ **05-10-2021**

Aclaraciones sobre entregables futuros

▼ **06-10-2021**

No hubo clase por el Congreso de Ciencias Exactas

▼ **07-10-2021**

No hubo clase por el Congreso de Ciencias Exactas

▼ **08-10-2021**

No hubo clase por el Congreso de Ciencias Exactas

▼ **11-10-2021**

▼ **Búsqueda Avara (Team de Diego)**

Este algoritmo combina las ventajas de los algoritmos primero en profundidad y primero en amplitud. Sigue un sendero a la vez pero puede cambiarse a otro sendero que parece más prometedor que el que está siguiendo.

Es una de las más sencillas estrategias, consiste en reducir al mínimo el costo estimado para lograr una meta.

En otras palabras, el nodo cuyo estado se considere más cercano a la meta en términos de costo de ruta se expande primero.

Aunque casi siempre es posible calcular el costo aproximado hasta la meta, es difícil hacerlo con precisión.

La función utilizada para dicho estimado del costo se llama función heurística, simbolizada por  $h$ .

$h(n)$  = costo estimado de la ruta más barata que une el estado del nodo  $n$  con un estado meta.

$h$  puede ser cualquier función. El único requisito es que  $h(n) = 0$  cuando  $n$  es una meta.

La mejor heurística para la determinación de rutas en el mundo real es la distancia en línea recta a la meta:

$h_{DLR}(n)$  = distancia en línea recta entre  $n$  y la meta

- Se asemeja a la búsqueda preferente por profundidad, ya que se “atora” al toparse con un callejón sin salida.
- Tiene sus mismas deficiencias: no es óptima, es incompleta, puede recorrer una ruta infinita.
- Su complejidad espacial es tan grande como su temporal:  $O(b^m)$ , donde  $m$  es la profundidad máxima del espacio de búsqueda. Una buena función heurística permite disminuir notablemente la complejidad tanto de espacio como de tiempo.

# Beneficios de la búsqueda avara

- Esta búsqueda usualmente produce resultados buenos
- Tienden a producir soluciones rápidamente, aunque no siempre la solución encontrada es la Óptima.
- Best first search puede cambiar entre BFS y DFS al obtener las ventajas de ambos algoritmos.
- Este algoritmo es más eficiente que los algoritmos BFS y DFS.

# PSEUDOCODIGO

1. Crea 2 listas vacías: ABIERTA y CERRADA
2. Comience desde el nodo inicial (diga N) y colóquelo en la lista ABIERTA "ordenada"
3. Repita los siguientes pasos hasta alcanzar el nodo OBJETIVO
  - a. Si la lista OPEN está vacía, entonces SALGA del bucle que devuelve "Falso"
  - b. Seleccione el primer / superior nodo (diga N) en la lista ABIERTO y muévelo a la lista CERRADO. También captura la información del nodo padre
  - c. Si N es un nodo OBJETIVO, mueva el nodo a la lista Cerrado y salga del ciclo devolviendo "Verdadero". La solución se puede encontrar retrocediendo el camino
4. Si N no es el nodo OBJETIVO, expanda el nodo N para generar los siguientes nodos "inmediatos" vinculados al nodo N y agregue todos esos a la lista ABIERTA
5. Reordenar los nodos en la lista ABIERTA en orden ascendente de acuerdo con una función de evaluación  $f(n)$

## EJEMPLO

Considere el siguiente problema de búsqueda. En cada iteración, cada nodo se expande usando la función heurística  $f(n)=h(n)$  dada en la tabla. En este ejemplo se usan dos listas: Abierta y Cerrada. A continuación se muestran las iteraciones durante la búsqueda del nodo G desde el nodo S

**Se expanden los nodos de S y se colocan en la lista cerrada.**

Inicialización: Open [A, B], Closed [S]

Iter 1: Open [A], Closed [S, B]

Iter 2: Open [E, F, A], Closed [S, B]

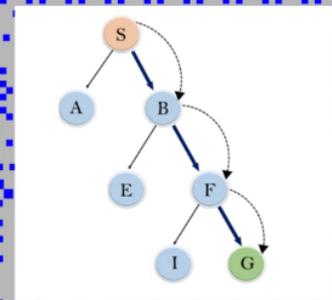
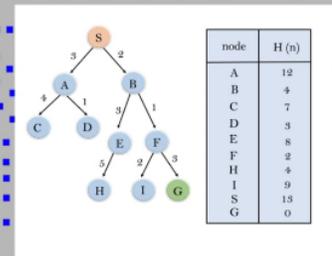
: Open [E, A], Closed [S, B, F]

Iter 3: Open [I, G, E, A], Closed [S, B, F]

: Open [I, E, A], Closed [S, B, F, G]

Así la solución final será: S----> B---->F----> G

Antonio Orduña Salas



## Resumen

- Consiste en reducir al mínimo el costo estimado para alcanzar una meta.
- Para ello se utiliza una función heurística, la cual estima el costo que implica llegar a una meta desde un estado determinado, y elige cuál es el siguiente nodo que se va a expandir aplicando esta función a cada nodo
- En esta búsqueda el tiempo y la cantidad de memoria necesaria crece exponencialmente con respecto a la profundidad. Pero la elección de una buena función heurística permite disminuir notablemente la complejidad tanto en tiempo como en espacio
- No es óptima ni completa.

### ▼ 12-10-2021

Complemento del profesor sobre la exposición de Diego

#### a) Best-First (6/6).

Para este problema particular, la búsqueda primero el mejor avara usando  $h_{DLR}$  encuentra una solución sin expandir un nodo que no este sobre el camino solución; de ahí que su costo de búsqueda es mínimo. Sin embargo no es óptimo: el camino vía Sibiu y Fagaras es 32 Kilómetros más largo que el camino por Rimnicu Vilcea y Pitesti. Esto muestra por qué se llama algoritmo <<avaro>> (en cada paso trata de ponerse tan cerca del objetivo como pueda).

54

## a) Best-First (1/6).

La **búsqueda voraz primero el mejor** trata de expandir el nodo mas cercano al objetivo, alegando que probablemente conduzca rápidamente a una solución. Así **evalúa los** nodos utilizando solamente la función heurística:  $f(n)=h(n)$ .

Veamos cómo trabaja para los problemas de encontrar una ruta en Rumanía utilizando la heurística **distancia en línea recta**, que llamaremos  $h_{DLR}$ .

49

Revisión de la Actividad 2.04

### ▼ 13-10-2021

#### ▼ Hill Climbing (Team de Pepe)

### DEFINICIÓN

- Es una técnica de optimización matemática que pertenece a la familia de los algoritmos de búsqueda local. Es un algoritmo iterativo que comienza con una solución arbitraria a un problema, luego intenta encontrar una mejor solución variando incrementalmente un único elemento de la solución. Si el cambio produce una mejor solución, otro cambio incremental se le realiza a la nueva solución, repitiendo este proceso hasta que no se puedan encontrar mejoras. Suele llamarse a esta búsqueda **algoritmo voraz local**, porque toma un estado vecino "bueno" sin pensar en la próxima acción.

- El Algoritmo Hill climbing es interesante para encontrar un óptimo local (una solución que no puede ser mejorada considerando una configuración de la vecindad) pero no garantiza encontrar la mejor solución posible (el óptimo global) de todas las posibles soluciones (el espacio de búsqueda). La característica de que sólo el óptimo local puede ser garantizado puede ser remediada utilizando reinicios (búsqueda local repetida), o esquemas más complejos basados en iteraciones, como búsqueda local iterada, en memoria, como optimización de búsqueda reactiva y búsqueda tabú, o modificaciones estocásticas, como simulated annealing.

## CARACTERISTICAS

- Informado: Utiliza información del estado por elegir un nodo u otro.
- No exhaustivo: No explora todo el espacio de estados. Como máximo, sólo encuentra una solución.
- Encuentra buenas soluciones, pero no la mejor, puesto que no es exhaustivo.
- Es eficiente, porque evita la exploración de una parte del espacio de estados.

### ▼ 14-10-2021

- ▼ Recocido simulado (Team de Cristian Pinto)

## DATOS GENERALES

- algoritmo de búsqueda metaheurística para problemas de optimización global; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande.
- El nombre e inspiración de SA 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.

## LOS VECINOS

- 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  $s$ .
- 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. Los estados vecinos son generados mediante métodos de Montecarlo.
- El método de evaluación de estados vecinos es fundamental para encontrar una solución óptima global al problema dado.

## PROTOCOLO DE RECOCIDO

- El algoritmo comienza con un valor de  $T$  muy alto, que va decreciendo en cada iteración siguiendo un cierto *protocolo de recocido*, que puede ser diferente para cada problema, pero que siempre debe terminar con  $T=0$ . Así el sistema será libre inicialmente de explorar una gran porción del espacio de búsqueda, ignorando pequeñas variaciones de la energía entre los estados vecinos evaluados, para más tarde centrarse en regiones con estados de baja energía y, al final, cambiar solo a estados con energía menor que la inicial, hasta alcanzar un mínimo.

### ▼ 15-10-2021

Revisión de nuestra propuesta de algoritmo genético

### ▼ 18-10-2021

Revisión de otros equipos sobre sus algoritmos genéticos

### ▼ 19-10-2021

#### ▼ Búsqueda Tabú (Team de Montse)

**BUSQUEDA TABU**

La búsqueda tabú toma de la **Inteligencia Artificial** el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de ésta.

El procedimiento trata de extraer información de lo sucedido y actuar en consecuencia. En este sentido puede decirse que hay un cierto aprendizaje y que la búsqueda es inteligente.

Es una metaheurística de **búsqueda agresiva**, es decir, trata de evitar que la búsqueda quede "atrapada" en un óptimo local.

**BUSQUEDA TABU**

La técnica parte de dos conceptos básicos, **vecindad** y **movimiento prohibido** o movimiento tabú.

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&lm=deeplink&lmsrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtCHQgDHS89Q...

## TERMINO TABU

De esta última definición se le da su nombre a éste método, ya que su esencia es la de **evitar el riesgo de seguir un camino no productivo**, incluyendo el de ser conducido a una trampa de la que no se puede salir (óptimo local).

09:34 a. m. 19/10/2021

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&lm=deeplink&lmsrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtCHQgDHS89Q...

## LISTA TABU

La lista tabú es una lista (linked list), donde se registran aquellas soluciones o atributos de soluciones que **no deben ser elegidas**.

**Puede contener:**

- Soluciones visitadas recientemente
- Movimientos realizados recientemente o
- Atributos o características que tenían las soluciones visitadas.

09:34 a. m. 19/10/2021

**LISTA TABU**

**Forma sencilla de realizarla**  
Cada vez que se realiza un movimiento, se introduce su inverso  
(si se pasó de  $x0$  a  $x1$ , el inverso es  $x1$  a  $x0$ )  
en una lista circular, de forma que los elementos de dicha lista están penalizados durante un cierto tiempo. 😞

Por tanto, si un movimiento está en la lista tabú no será aceptado, aunque aparentemente sea mejor solución que la solución actual.

The slide features a pink and orange sunburst background with a lightning bolt icon in the top right corner. The Microsoft Teams interface is visible, including a search bar and a participant list at the bottom.

**TABU TENURE**

**Tiempo o número de iteraciones que un elemento (movimiento o atributo) permanece en la lista tabú.**

Es posible que no todos los elementos de la lista TABÚ tengan el mismo *tenure*. 😊

The slide features a pink and orange sunburst background with a lightning bolt icon in the top right corner and smiley face icons in the bottom left and right corners. The Microsoft Teams interface is visible, including a search bar and a participant list at the bottom.

**CRITERIO DE ASPIRACION**

Es posible quitar un elemento de la lista tabú antes de que se complete su tabú tenure si se llega a cumplir algún criterio de aspiración.

**Las aspiraciones son de dos clases:**

- Aspiraciones de movimiento: revoca la condición tabú del movimiento.
- Aspiraciones de atributo: revoca el status tabú del atributo.

The slide features a background with red and pink sunburst rays, a yellow lightning bolt icon in the top right, and a yellow smiley face icon in the bottom left.

**CRITERIOS DE ASPIRACION**

**Aspiración por Default:** Si todos los movimientos posibles son clasificados como tabú, entonces se selecciona el movimiento "menos tabú".

The slide features a background with red and pink sunburst rays and a row of seven lightning bolt icons at the bottom, alternating in color between blue and yellow.

Microsoft Teams

## CRITERIOS DE ASPIRACION

**Aspiración por Objetivo:** Una aspiración de movimiento se satisface, permitiendo que un movimiento  $x$  sea un candidato para seleccionarse.

por ejemplo:  
 $F(x) < \text{mejor costo.}$   
(en un problema de minimización)

09:36 a. m. 19/10/2021

Microsoft Teams

La grabación se ha iniciado. Esta reunión se está grabando. Al unirse, da su consentimiento para que se grabe. [Política de privacidad](#) Descartar

## CRITERIOS DE ASPIRACION

**Aspiración por Dirección de Búsqueda:** Un atributo de aspiración para la solución "s" se satisface si la dirección en "s" proporciona un mejoramiento y el actual movimiento es un movimiento de mejora.

Entonces "s" se considera un candidato

20:51

Montserrat Alejandra Ulloa Rivera

09:36 a. m. 19/10/2021

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&Im=deeplink&msrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtGgDHS89Q...

Microsoft Teams

MEMORIA

opera en relación a cuatro dimensiones principales

<p><b>calidad</b></p> <p>Habilidad para diferenciar el mérito de las soluciones.</p> <p>Un movimiento puede ser de mejor <b>calidad</b> si, por ejemplo, su frecuencia de ocurrencia en el pasado es baja o no ha ocurrido antes y nos permite explorar nuevas regiones.</p>	<p><b>influencia</b></p> <p>Considera el impacto de las elecciones hechas durante la búsqueda.</p> <p>En ausencia de movimientos de mejora, los criterios de aspiración cambian para dar a los movimientos influyentes un rango mayor y estos pueden salir de la lista tabú antes del plazo establecido en su tabú tenure.</p>	<p><b>corto plazo</b></p> 	<p><b>largo plazo</b></p> 
--	--	--	---

09:37 a. m. 19/10/2021

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&Im=deeplink&msrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtGgDHS89Q...

Microsoft Teams

MEMORIA

opera en relación a cuatro dimensiones principales

**Memoria basada en lo reciente (corto plazo)**

Puede ser utilizada para "recordar" aquellos movimientos que hacen caer de nuevo en soluciones ya exploradas.

Objetivo: es penalizar la búsqueda para evitar el ciclo.

09:37 a. m. 19/10/2021

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&Im=deeplink&msrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtCHGgDHS89Q...

Microsoft Teams

MEMORIA

opera en relación a cuatro dimensiones principales

**Memoria basada en frecuencia (largo plazo)**  
 En esta estructura de memoria se registra la frecuencia de ocurrencias de los movimientos, las soluciones o sus atributos y puede ser:

- Frecuencia de transiciones: Cantidad de veces que una solución es la mejor o cantidad de veces que un atributo pertenece a una solución generada
- Frecuencia de residencia: Cantidad de iteraciones durante la cual un atributo pertenece a la solución generada

09:38 a. m. 19/10/2021

MHI (Reunión) | Microsoft Teams | Metaheurísticas I

teams.microsoft.com/?culture=es-mx&country=MX&Im=deeplink&msrc=homePageWeb&cmpid=WebSignIn#/pre-join-calling/19:BqewiNEK62SPtCHGgDHS89Q...

Microsoft Teams

MEMORIA

opera en relación a cuatro dimensiones principales

**Memoria basada en frecuencia (largo plazo)**

**Intensificación**  
 Consiste en regresar a regiones ya exploradas para estudiarlas más a fondo. Para ello se favorece la aparición de aquellos atributos asociados a buenas soluciones encontradas.

**Diversificación**  
 Consiste en visitar nuevas áreas no exploradas del espacio de soluciones. Para ello se modifican las reglas de elección para incorporar a las soluciones atributos que no han sido usados frecuentemente.

09:38 a. m. 19/10/2021

Microsoft Teams

### Algoritmo de la BÚSQUEDA TABÚ SIMPLE

Generar solución inicial  $x_0$   
 $k := 1$ .  
 $x := x_0$ . (x es la solución actual)

**MIENTRAS** la condición de finalización no se encuentre

**HACER:**

- Identificar  $N(x)$ . (Vecindario de x)
- Identificar  $T(x,k)$ . (Lista Tabú)
- Identificar  $A(s,k)$ . (Conjunto de Aspirantes)
- Determinar  $N^*(x,k) = (N(x) - T(x,k)) \cup A(x,k)$ . (Vecindario reducido)
- Escoger la mejor  $x \in N^*(x,k)$
- "Guardar" x si mejora la mejor solución conocida  $x_k := x$ .
- Actualizar la lista tabú
- $k := k+1$ .

**FIN MIENTRAS**

09:39 a. m. 19/10/2021

Microsoft Teams

### Elementos a considerar... al implementar este método

Componente	Símbolo	Descripción
Función Objetivo	$f(x)$	La función que nos permite identificar la calidad de la solución, generalmente es maximizar o minimizar una expresión lineal o no lineal
Espacio de Soluciones	S	El conjunto de soluciones posibles, puede estar dado explícitamente o estar dada la estructura de la solución, generalmente en $R^n$
Vecindario	$N(x)$	Las posibles soluciones que pueden seguir a x. En el método simple serían todos los arreglos tal que una variable actualmente básica pasa a tener un valor cero y una no básica pasa a ser básica $N(x) \subseteq S$
Lista Tabú	$T(x,k)$	Lista de las posibles soluciones que no pueden ser elegidas en la actual iteración. Pueden ser las "L" soluciones anteriores o los L últimos movimientos
Conjunto de aspirantes o candidatos	$A(x,k)$	Soluciones que tienen algún atributo que las hace elegibles aún si estuvieran en la lista tabú. El conocimiento del "experto" es importante para determinar qué soluciones pueden aspirar a ser solución en la iteración k.

23:54

09:39 a. m. 19/10/2021

Microsoft Teams Meeting: MH1 (Reunión) | Microsoft Teams | Metaheurísticas I

### ALGUNAS APLICACIONES...

- Se puede aplicar a cualquier problema de optimización tratado previamente con un método tradicional de paso descendente.
- Existen problemas de tipo **combinatorio** en los cuales la técnica se aplica en forma natural.
- Problema de recargas de combustible nuclear.
- Diseño de co...
- Problemas de pegado de proteínas.
- Diseño axial de combustible nuclear.
- Aplicación de colorantes en la industria textil.
- Problemas de rutas de vehículos.

24:06

Montserrat Alejandra Ulloa Rosas

Aplicaciones: +11

Montserrat Alejandra Ulloa... Giancarlo Ramirez D... Leonardo De Jesus R... Francisco Javier Luna Rosas

Escribe aquí para buscar

09:40 a. m. 19/10/2021

Microsoft Teams Meeting: MH1 (Reunión) | Microsoft Teams | Metaheurísticas I

### Gracias x su atención, muy amables



24:34

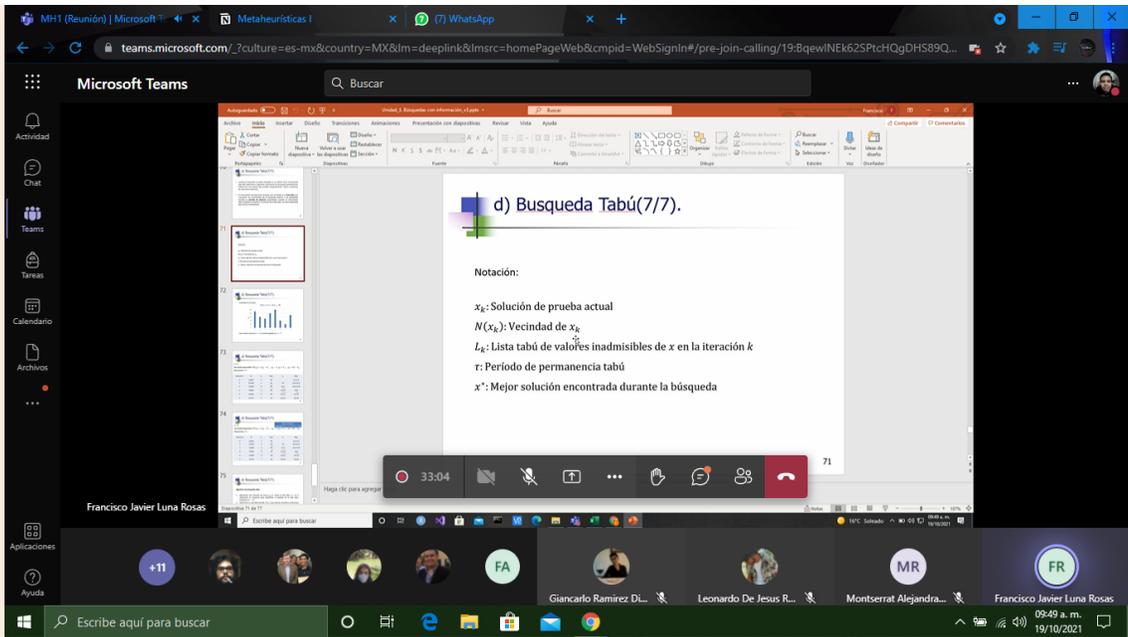
Montserrat Alejandra Ulloa Rosas

Aplicaciones: +13

Montserrat Alejandra Ulloa... Giancarlo Ramirez D... Leonardo De Jesus R... Francisco Javier Luna Rosas

Escribe aquí para buscar

09:40 a. m. 19/10/2021



▼ 20-10-2021

Búsqueda tabú por parte del profesor

▼ 21-10-2021

Revisión de algoritmos genéticos

▼ 22-10-2021

Dejó y explicó la segunda evaluación parcial

▼ 25-10-2021

Mayor explicación del examen y revisión de actividades

▼ 26-10-2021

Algoritmo genético simple

Examen

▼ 27-10-2021

Explicación de algoritmo genético vs selección natural

▼ 28-10-2021

Explicación del segundo parcial

▼ **29-10-2021**

No hubo clase

▼ **Tercer Parcial**

▼ **01-11-2021**

Revisión de exámenes de Diego y Gustavo II

▼ **02-11-2021**

No hubo clase. Asueto

▼ **03-11-2021**

Chequeo de exámenes de Chuy y Marco

▼ **04-11-2021**

Teorema del esquema

▼ **05-11-2021**

No hubo clase

▼ **08-11-2021**

Teorema del esquema

▼ **09-11-2021**

## **Algoritmos bioinspirados**

Inteligencia en comportamiento colectivo de diversas especies

Sistemas autoorganizados

Ejemplos:

- Parvadas de pájaros
- Bancos de peces
- Colonias de abejas

- Colonias de hormigas
- GWO (Grey Wolf Optimization)
- Artificial Immune System

Basados en inteligencia colectiva

Toca exponer sobre Ant Colony Optimization con el team de Marco

#### ▼ 10-11-2021

##### **Fase 4. Agrupación de los datos.**

En esta fase el análisis de sentimiento puede proporcionarnos información valiosa acerca del suicidio de los comentarios compartidos en Twitter por los usuarios, al detectar estas emociones u opiniones a partir del gran volumen de datos recolectados desde las fases anteriores. Este análisis incluye tres clases principales de polaridad, que son la positivas, negativas y neutrales. La polaridad de cada tweet se determina asignando una puntuación de  $[-1.0, 1.0]$  que hace referencia a como el texto puede ser medido en positivo o negativo dependiendo del tono del tweet, para lo cual  $-1$  nos indicara que son más negativos y  $+1$  es más positiva, mientras que el valor de cero se considerara como un sentimiento neutral. Una puntuación de subjetividad  $[0.0, 1.0]$  hace referencia a la representación de un significado subjetivo u objetivo, donde el valor cercano a cero representa un comentario objetivo y cercano al 1 es un comentario subjetivo [56].

Para detectar la polaridad y subjetividad en los tweets recolectados para esta investigación, se hizo uso de la biblioteca de TextBlob [57] de Python [58], la cual utiliza internamente un diccionario con un total de 2920 palabras (Figura 4), puede ser una ventaja el utilizar un enfoque basado en un diccionario para extraer el sentimiento del tweet, ya que se puede encontrar rápidamente un gran número de palabras con sus orientaciones, pero se puede convertirse en una

#### ▼ 11-11-2021

Exposición nuestra de Colonia de Hormigas

#### ▼ 12-11-2021

Exposición de algoritmo de inteligencia colectiva del sistema inmune humano

#### ▼ 15-11-2021

No hubo clase. Asueto

▼ **16-11-2021**

???

▼ **17-11-2021**

Exposición de Colonia de Abejas

▼ **18-11-2021**

Exposición PSO del Team de Horacio

▼ **19-11-2021**

Explicación del proyecto

▼ **22-11-2021**

Explicaciones finales sobre el examen y el proyecto

▼ **23-11-2021**

Explicación del proyecto final

▼ **24-11-2021**

▼ **25-11-2021**

▼ **26-11-2021**