

Instituto Tecnológico y de Estudios Superiores de Monterrey
Carlos Bravo Marquez - A0826274
Gustavo Benitez - A00826760
Kevin Bojórquez Varela - A01176165
Ricardo Trevizo García - A01196062

Métodos numéricos en ingeniería
24 de septiembre de 2020

Primer reporte preliminar

Repositorio de GitHub:

https://github.com/CarlosBravo00/Optimizacion_Moderna

Definiciones clave:

- Optimización: Resolución de problemas eficientemente donde se espera una minimización o una maximización, recurriendo a principios matemáticos y métodos.
- Función objetivo: La función que se busca optimizar.
- Metaheurística: Algoritmo con un método de optimización diseñado para hallar una solución lo suficientemente aproximada en un tiempo de cómputo relativamente pequeño.
- Algoritmo genético: Algoritmo de optimización basado en la selección natural que busca explorar varias situaciones posibles y escoger la mejor de estas o bien, diversas soluciones igual de viables.

Listado de tareas realizadas:

(RESPONSABLE)

- Primera entrega
 - Creación de repositorio en github (**Carlos**): Crear el repositorio para control de cambios y trabajo en equipo
 - Investigación inicial (**Kevin**): investigar los conceptos básicos del proyecto
 - Planeación de actividades (**Carlos**)
 - Selección de la metaheurística (**Gustavo & Ricardo**): Con la investigación se selecciona una metaheurística para implementar
- Segunda entrega
 - Esquema de interfaz gráfica (**Carlos**): Diseñar una interfaz gráfica para simplificar el uso de el programa
 - Codificación del algoritmo genético seleccionado (**Gustavo & Ricardo**)
 - Codificación de la metaheurística seleccionada (**Kevin**)
 - Documentación (**Carlos & Gustavo & Kevin & Ricardo**): Todo el equipo documentara la funcionalidad del algoritmo, la metaheurística y la interfaz gráfica.
- Entrega final

- Implementación de la interfaz gráfica de usuario (**Carlos**)
- Implementación de programa principal (**Kevin & Ricardo**)
- Pruebas de calidad (**Gustavo**): Realizar pruebas del sistema entero y documentar los resultados

Name	Duration	Start	Finish	Perce...	Pr
Primera Entrega	14 days	9/7/20 8:00 AM	9/24/20 5:00 PM	100%	
Creación de repositorio en github	1 day	9/7/20 8:00 AM	9/7/20 5:00 PM	100%	
Investigación inicial	8 days	9/7/20 8:00 AM	9/16/20 5:00 PM	100%	
Planeación de actividades	6 days	9/7/20 8:00 AM	9/14/20 5:00 PM	100%	
Selección de la metaheurística	6 days	9/17/20 8:00 AM	9/24/20 5:00 PM	100%	3
Primer reporte preliminar	0 days	9/25/20 8:00 AM	9/25/20 8:00 AM	100%	1
Segunda entrega	22 days	9/25/20 8:00 AM	10/26/20 5:00 PM	0%	1
Esquema de interfaz gráfica	10 days	9/25/20 8:00 AM	10/8/20 5:00 PM	0%	
Codificación del algoritmo genético seleccionado	15 days	9/25/20 8:00 AM	10/15/20 5:00 PM	0%	
Codificación de la metaheurística seleccionada	15 days	9/25/20 8:00 AM	10/15/20 5:00 PM	0%	
Documentación	7 days	10/16/20 8:00 AM	10/26/20 5:00 PM	0%	8;9;10
Segundo reporte preliminar	0 days	10/26/20 5:00 PM	10/26/20 5:00 PM	0%	7
Entrega final	18 days	10/27/20 8:00 AM	11/19/20 5:00 PM	0%	7
Implementación de la interfaz gráfica de usuario	11 days	10/27/20 8:00 AM	11/10/20 5:00 PM	0%	
Implementación de programa principal	11 days	10/27/20 8:00 AM	11/10/20 5:00 PM	0%	
Pruebas de calidad	7 days	11/11/20 8:00 AM	11/19/20 5:00 PM	0%	14;15
Reporte final	0 days	11/19/20 5:00 PM	11/19/20 5:00 PM	0%	13

o Descripción de un algoritmo genético, incluyendo su pseudocódigo y una explicación de los operadores y parámetros que puede utilizar, incluyendo sus rangos.

Un algoritmo genético simula el proceso de evolución usando operadores de selección, cruzamiento y mutación, para este ejemplo encontramos una implementación de un algoritmo genética que permite elaborar la distribución de horarios académicos

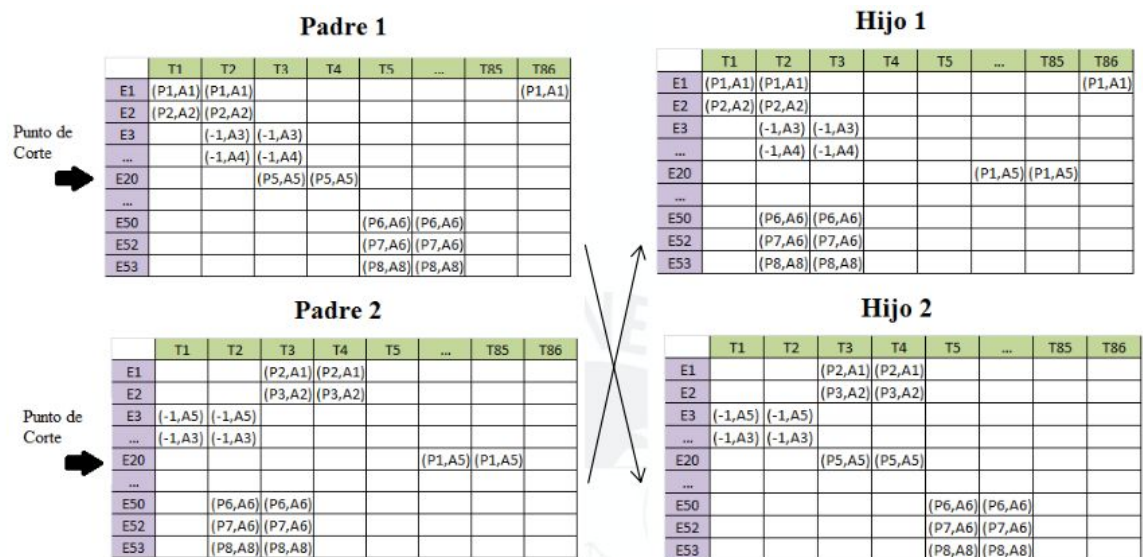
<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/6057>

Descripción: Para este algoritmo genético se busca mejorar la manera en que se distribuyeron los horarios académicos a partir de una población inicial generada por un algoritmo Grasp fase construcción. El fin de esta solución es el poder generar de manera automática los horarios de clases de una facultad de ciencias e ingeniería y así reducir el tiempo de la oración, mientras que se maximiza el uso de los recursos y la distribución de los cursos al momento de crear los horarios para cada período.

Operadores:

- **Operador de selección:** Este se utiliza para poder seleccionar 2 individuos de la población que serán después afectados por el operador de casamiento. esto para que puedan formar parte de los descendientes de la siguiente generación. En este caso se utilizó un método de ruleta para favorecer a los individuos mejor adaptados. A cada individuo se le asignará un valor de probabilidad en función a la adaptación que tenga, esto tomará en cuenta que el que posea un menor valor en la función de adaptación (minimización) tendrá una mayor probabilidad de ser elegido.
- **Operador de casamiento:** Ésta se utiliza para unir a dos individuos de la población para así lograr formar una nueva generación de individuos. En este operador se utiliza una tasa de casamiento, la cual se utilizará para calcular la probabilidad de uniones o cruces que existen dentro de la población. Cabe mencionar que no todos los individuos son afectados por este operador.

La manera de funcionar del operador de casamiento, primero toma a los dos padres seleccionados por el operador anterior y se determina una posición al azar o también llamada punto de corte, con esta posición se producirá el corte de los dos individuos para así recombinar sus extremos y generar nuevos individuos. La combinación va a generar dos hijos, el primero conformado por la parte uno del individuo 1 y la parte 2 del individuo 2; Mientras que el segundo es conformado por la parte 1 del individuo 2 y la parte 2 del individuo 1.



- **Operador de mutación:** Se encarga de montar un gen de los individuos, esto con el fin de no dejar algún espacio de búsqueda con probabilidad 0 a ser examinado. En este operador se utiliza una tasa de mutación, la cual ayuda la probabilidad de que un individuo mute; Se basa esto en una probabilidad ya que no necesariamente se debe afectar a todos los individuos.

Primero se toman a los dos hijos generados por el operador de casamiento y se genera un número al azar entre 0 y 1. Si el número generado es mayor a la tasa de mutación se cancela la operación; mientras que si es menor, se procede con la mutación. Para la mutación se elige al azar un profesor, seguido de los cursos que pueda enseñar y así determinar a qué eventos puede ser asignado.

Lista de Profesores	0	1	2	3	4	5	6	7
---------------------	---	---	---	---	---	---	---	---

Random : 1

Lista de Cursos de Profesor 3	0	1	2	3
-------------------------------	---	---	---	---

Random : 2

Lista de Eventos Curso 1	1	4	5	7	8	10	11	12	20
--------------------------	---	---	---	---	---	----	----	----	----

Para terminar, se eligen al azar dos de los eventos del individuo que formen parte de la lista de eventos del curso y se hace el cambio de asignación.

R1		T1	T2	T3	T4	T5	...	T85	T86
➡	E1	(P1,A1)	(P1,A1)						(P1,A1)
	E2	(P2,A2)	(P2,A2)						
	E3		(-1,A3)	(-1,A3)					
R2	...		(-1,A4)	(-1,A4)					
➡	E20						(P1,A5)	(P1,A5)	
	...								
	E50		(P6,A6)	(P6,A6)					
	E52		(P7,A6)	(P7,A6)					
	E53		(P8,A8)	(P8,A8)					

	T1	T2	T3	T4	T5	...	T85	T86
E1						(P1,A5)	(P1,A5)	
E2	(P2,A2)	(P2,A2)						
E3		(-1,A3)	(-1,A3)					
...		(-1,A4)	(-1,A4)					
E20	(P1,A1)	(P1,A1)						(P1,A1)
...								
E50		(P6,A6)	(P6,A6)					
E52		(P7,A6)	(P7,A6)					
E53		(P8,A8)	(P8,A8)					

Parámetros: Se tienen tres parámetros principales en el algoritmo genético, incluyendo el tamaño de la población, la tasa de casamiento y la tasa de mutación.

Pseudocódigo:

- 1 Proceso AlgoritmoGenetico (tamaño poblacion, tasa casamiento, tasa mutacion)
- 2 Poblados <- Lista de soluciones
- 3 MejorSolucion
- 4
- 5 Poblacion <- Inicializar poblacion inicial
- 6 Evalua elementos de la poblacion: funcionFitness

```

7      Mientras no se cumpla la condicion de parada hacer
          P1,P2 <- Seleccion_Padres: Operador Seleccion
9      H1,H2 <- Casar Padres Seleccionados : Operador Casamiento
10     H1,H2 <- Mutar nueva generación : Operador Mutacion
11     Poblacion <- evaluar si ingresan a la nueva generacion
12     MejorSolucion <- evaluacion de elitismo
13     FinMientras
14     FinPnoces

```

o Listado de, al menos, cuatro metaheurísticas, el proceso en el que se inspiran, y una referencia a algún artículo o trabajo donde se utilice (puede ser el artículo original donde se presentó el método).

Metaheurísticas consideradas:

- Algoritmo de colonia de abejas artificial: se inspira en el comportamiento de búsqueda y selección de alimento de las abejas, para la producción de miel.

Referencia: Meng, T., Pan, Q.-K., & Sang, H.-Y. (2018). A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations. *International Journal of Production Research*, 56(16), 5278–5292.
<http://0-search.ebscohost.com.biblioteca-ils.tec.mx/login.aspx?direct=true&db=bsu&AN=131778688&lang=es&site=eds-live&scope=site>

- Algoritmo de búsqueda armónica: se basa en el proceso de improvisación de un músico, ya que cuando este debe improvisar tiene 3 posibles soluciones: tocar cualquier pieza famosa desde su memoria, tocar algo similar a una pieza conocida o componer nuevas notas o aleatorias. Así pues hay tres componentes en este algoritmo: memoria armónica, ajuste de tono o aleatorización.

Referencia: Yi, J., Li, X., Chu, C.-H., & Gao, L. (2019). Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization. *Journal of Intelligent Manufacturing*, 30(1), 405–428.
<http://0-search.ebscohost.com.biblioteca-ils.tec.mx/login.aspx?direct=true&db=bsu&AN=134265196&lang=es&site=eds-live&scope=site>

- Algoritmo de colonia de hormigas: está inspirado en el comportamiento de búsqueda de alimento de las hormigas, en el cual, la socialización entre ellas juega un papel importante.

Referencia: Olkhova M, Roslavitsev D, Matviichuk O, Mykhalenko A. City Delivery Routes Planning Based on the Ant Colony Algorithm. *Science & Technique*. 2020;19(4):356-362.

<http://0-search.ebscohost.com/biblioteca-ils.tec.mx/login.aspx?direct=true&db=asn&AN=145049915&lang=es&site=eds-live&scope=site>

- Recocido Simulado: imita el proceso de recocido en el procesamiento de materiales cuando un metal se enfría y se congela en un estado cristalino con una energía mínima y un tamaño de cristal más grande, para reducir la ductilidad. Tiene la ventaja de evitar quedarse atorado en un mínimo local.

Referencia: Yang, X.-S. (2008). Introduction to Mathematical Optimization : From Linear Programming to Metaheuristics. Cambridge International Science Publishing
<http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=311051&site=ehost-live>

Descripción de la metaheurística seleccionada:

Algoritmo de Búsqueda Armónica

Descripción: se basa en el proceso de improvisación de un músico, ya que cuando este debe improvisar una melodía, tiene 3 posibles soluciones:

- Puede tocar alguna melodía conocida, la cual aprendió previamente.
- Puede tocar una melodía similar a una conocida, ajustándose un poco para llegar a un tono deseado.
- Componer una melodía completamente nueva, utilizando notas aleatorias.

El músico toca cada nota dentro de un posible rango, formando un vector armónico. Este vector armónico se guarda en la memoria del músico, en caso de que considere que sea una buena armonía. La próxima vez que desee improvisar una melodía, usará estos conocimientos adquiridos para incrementar sus probabilidades de crear una mejor melodía.

De la misma manera, el algoritmo armónico toma un conjunto de variables aleatorias dentro del rango decidido, y crea un vector. Posteriormente, lo analiza y decide si este vector es una buena solución. Si es así, almacena los valores en la memoria de cada variable. Al igual que el algoritmo genético, este proceso puede ser iterado hasta llegar a la solución deseada.

Los parámetros a considerar son los siguientes:

- El número de improvisaciones: el número de iteraciones
- El tamaño de la memoria armónica: el rango que puede tener cada variable
- La tasa de consideración: el porcentaje de una solución que se usará en la siguiente iteración. Va de 0 a 1, donde entre más grande sea este número, es mayor la calidad o aproximación del resultado.
- Tasa de ajuste de tono: que tanto puede cambiar una variable en respecto a la iteración anterior
- Ancho de banda de ajuste de tono

Pseudocódigo:

```

Definir función  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)^T$ 
Generar armonías iniciales (generación de arreglos)
Ingresar tasa de ajuste de tono (tTono), tamaño de memoria armónica (tamMA) y
ancho de banda (abMin, abMax)
Ingresar tasa de consideración if( $tCnsdr < 0$  AND  $tCnsdr > 1$ ), pedir un valor válido
Ingresar número de improvisaciones (numImp)

while(  $t < numImp$ )
    Generar nueva memoria armónica con las mejores armonías
    Ajustar tono para tener nuevas armonías
    if( $rand > tCnsdr$ ), escoger nueva armonía aleatoria
    else if ( $rand > tTono$ ), ajustar el tono con un valor aleatorio respetando límites
    else generar nuevas armonías aleatoriamente
    Reemplazar en la memoria a las nuevas armonías si son mejores
     $numImp = numImp + 1$ 

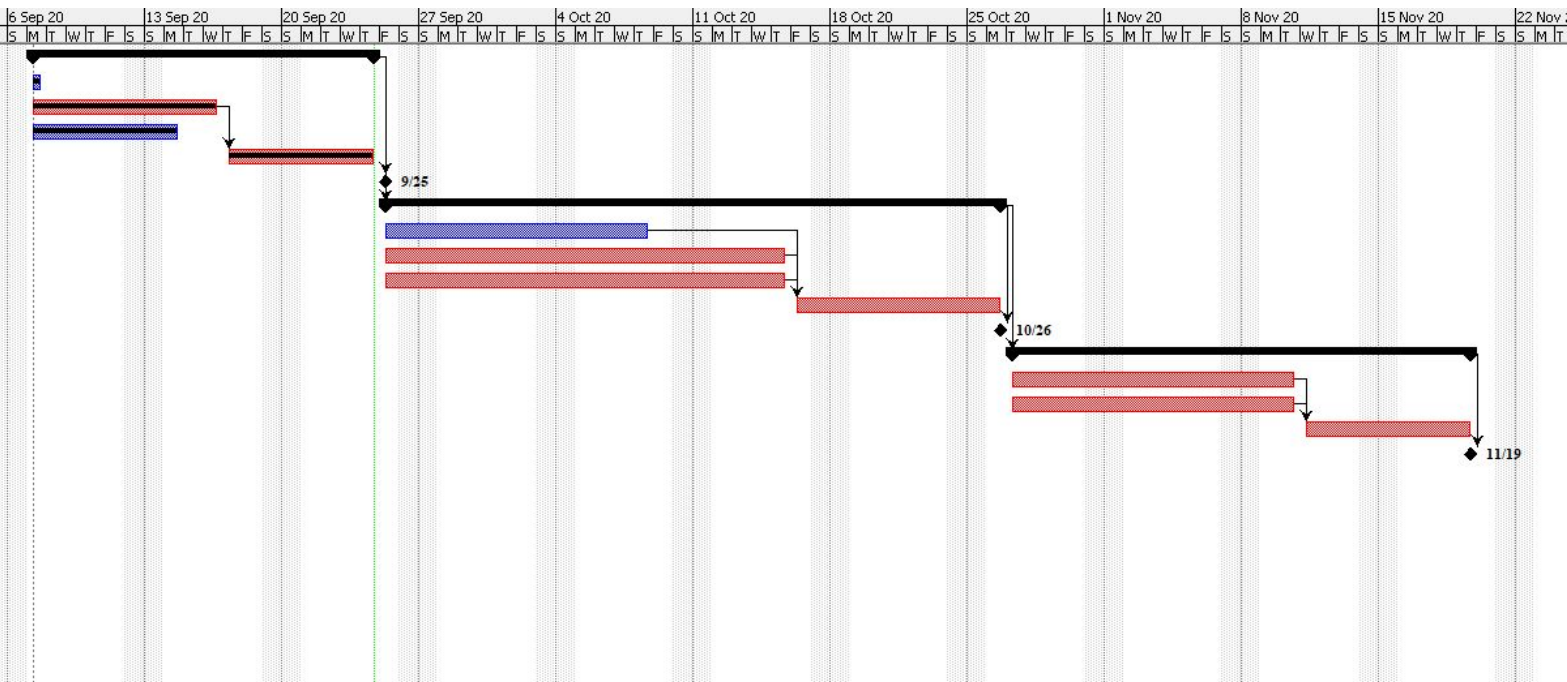
Desplegar armonías guardadas en la memoria armónica

```

Justificación: Se escogió el algoritmo de búsqueda armónica como opción primaria para la resolución de la problemática debido a que, dentro de las metaheurísticas indagadas, esta es la que en la mayoría de los casos se comportará de una manera más amigable con los datos ingresados y los procesa para arrojar una solución viable más aproximada, debido a su cualidad de memoria armónica; de igual manera, el equipo se siente con más confianza de usar este algoritmo debido a la antigüedad de este mismo y la existencia de una amplia oferta de recursos de ayuda (que sería muy útil ante problemáticas que se encuentren durante el proceso de su implementación). Se consideran a los algoritmos de recocido simulado y colonia de abejas artificial como segunda y tercera opción alternativa, respectivamente.

Diagrama de Gantt:

Name	Duration	Start	Finish	Perce...	Pr
Primera Entrega	14 days	9/7/20 8:00 AM	9/24/20 5:00 PM	100%	
Creación de repositorio en github	1 day	9/7/20 8:00 AM	9/7/20 5:00 PM	100%	
Investigación inicial	8 days	9/7/20 8:00 AM	9/16/20 5:00 PM	100%	
Planeación de actividades	6 days	9/7/20 8:00 AM	9/14/20 5:00 PM	100%	
Selección de la metaheurística	6 days	9/17/20 8:00 AM	9/24/20 5:00 PM	100%	3
Primer reporte preliminar	0 days	9/25/20 8:00 AM	9/25/20 8:00 AM	100%	1
Segunda entrega	22 days	9/25/20 8:00 AM	10/26/20 5:00 PM	0%	1
Esquema de interfaz gráfica	10 days	9/25/20 8:00 AM	10/8/20 5:00 PM	0%	
Codificación del algoritmo genético seleccionado	15 days	9/25/20 8:00 AM	10/15/20 5:00 PM	0%	
Codificación de la metaheurística seleccionada	15 days	9/25/20 8:00 AM	10/15/20 5:00 PM	0%	
Documentación	7 days	10/16/20 8:00 AM	10/26/20 5:00 PM	0%	8;9;10
Segundo reporte preliminar	0 days	10/26/20 5:00 PM	10/26/20 5:00 PM	0%	7
Entrega final	18 days	10/27/20 8:00 AM	11/19/20 5:00 PM	0%	7
Implementación de la interfaz gráfica de usuario	11 days	10/27/20 8:00 AM	11/10/20 5:00 PM	0%	
Implementación de programa principal	11 days	10/27/20 8:00 AM	11/10/20 5:00 PM	0%	
Pruebas de calidad	7 days	11/11/20 8:00 AM	11/19/20 5:00 PM	0%	14;15
Reporte final	0 days	11/19/20 5:00 PM	11/19/20 5:00 PM	0%	13



Referencias:

- Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, 5(1), 1-35.
- Meng, T., Pan, Q.-K., & Sang, H.-Y. (2018). A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations. *International Journal of Production Research*, 56(16), 5278–5292.
<http://0-search.ebscohost.com/biblioteca-ils.tec.mx/login.aspx?direct=true&db=bsu&AN=131778688&lang=es&site=eds-live&scope=site>
- Yi, J., Li, X., Chu, C.-H., & Gao, L. (2019). Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization. *Journal of Intelligent Manufacturing*, 30(1), 405–428.
<http://0-search.ebscohost.com/biblioteca-ils.tec.mx/login.aspx?direct=true&db=bsu&AN=134265196&lang=es&site=eds-live&scope=site>
- Olkhova M, Roslavytsev D, Matviichuk O, Mykhalenko A. City Delivery Routes Planning Based on the Ant Colony Algorithm. *Science & Technique*. 2020;19(4):356-362.
<http://0-search.ebscohost.com/biblioteca-ils.tec.mx/login.aspx?direct=true&db=asn&AN=145049915&lang=es&site=eds-live&scope=site>
- Yang, X.-S. (2008). *Introduction to Mathematical Optimization : From Linear Programming to Metaheuristics*. Cambridge International Science Publishing
<http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=311051&site=ehost-live>
- Cobos., C., Pérez., J., & Estupiñan., D. (2011). Una revisión de la búsqueda armónica. *Avances en Sistemas e Informática*, 8(2), 67-80. Recuperado de <https://revistas.unal.edu.co/index.php/avances/article/view/26727/27036>
- A., & A. (2015, August 06). Elaboración de una solución metaheurística usando un algoritmo genético que permita elaborar la distribución de los horarios académicos. Recuperado de <http://hdl.handle.net/20.500.12404/6057>