Proyecto 1. Investigación de operaciones.

José Carlos Zamorano - 142576 Christian Alvarado - 144887

23 de septiembre de 2020

1. Problema 1

El señor Márquez trabaja su granja junto con su hijo y su tractor. Su hija, Julieta Márquez, va a hacer una política completa para el reemplazo del tractor durante los siguientes 17 años, que es cuando el señor Márquez se jubilará. La información que tiene Julieta para realizar el modelo es la siguiente:

- Actualmente el tractor tiene 2 años de antigüedad y el precio de uno nuevo es de $43,000 USD.
- No tiene sentido vender un tractor antes de los dos años, pero se debe de vender a lo más al alcanzar los 5 años de antigüedad.
- Al final de los 17 años se venderá el tractor sin importar si el tractor haya alcanzado los dos años.
- Los precios de los tractores aumentan un 5 % anual con respecto al precio anterior.
- Al comprar un tractor nuevo la depreciación inmediata es del 10 %, y cada año que pasa se deprecia un 7 % con respecto al año anterior.
- Los gastos de mantenimiento y operación actuales son de $1,300 al año. El aumento de este costo será de un 15 % anual con respecto al año anterior.

1.1. Modelo de programación lineal

Se busca minimizar el costo total de reemplazar el tractor usado durante los próximos 17 años, empezando con un tractor de dos años de antigüedad. Las decisiones se toman al principio de cada año, y se empiezan a tomar a partir del primer año, el cual contamos a partir de $i = 0$.

Sean

- n = 17 el número de años durante los cuáles se planea la política de reemplazo del tractor.
- a = 2 la antigüedad inicial del tractor al tiempo $i = 0$

- $m = 2$ el mínimo de años de antigüedad que tiene que tener un tractor antes de que se pueda vender.
- $M = 5$ el máximo de años de antigüedad que puede alcanzar un tractor.
- C_i el costo de un tractor al tiempo i, dado por:

$$C_i = 43,000(1.05)^i, \text{ con } i \in \{0, \cdots, n-1\}$$

- v_{ij} el precio de venta en el año j del tractor comprado en el año i :

$$v_{ij} = \begin{cases} 0.9C_i(0.93)^{(j-i)} & \text{si i=0,1,...,n-1; j = i+1,...,s} \\ C_{-2}(0.93)^j & \text{si i = -2; j = 0,....,M-a} \end{cases}$$

$$C_{-2} = 31,000$$

- m_j el costo de mantenimiento de un tractor en el año j:

$$m_j = 1300(1.15)^j, \text{ con } j \in \{0, \cdots, n\}$$

Suposiciones:

- Suponemos que la familia Márquez no puede tener más de un tractor al mismo tiempo
- Automáticamente después de vender un tractor se hace la compra del nuevo.
- Vamos a tomar el precio de venta del tractor actual como $31,000.

Variables de decisión:

- Vender o no vender el tractor:

$$x_{ij} = \begin{cases} 1 & \text{si vendo el tractor comprado al inicio del año i en el año j} \\ 0 & \text{si no vendo el tractor} \end{cases}$$

con $i \in \{-2, 0, 1, \cdots, 16\}$, $j \in \{i+1, \cdots, s\}$, $s = \text{mín}(i+5, 17)$

Nota: El subíndice $i = -2$ representa nuestro tractor inicial con antigüedad de dos años.

Función objetivo:

La función objetivo busca minimizar el costo total de reemplazar y mantener un tractor durante los siguientes 17 años. Los costos a tomar en cuenta son: el costo de mantenimiento de un tractor m_j, el precio de comprar un nuevo tractor C_j y el dinero que se recupera al vender un tractor en al tiempo j. Sabemos que el costo de mantenimiento se va pagar todos inicio de cada año ya que se tiene tractor todos los años. Dado que el costo de comprar es mayor que el precio de venta, nuestra función es estrictamente mayor a cero.

$$\text{mín}\left(\sum_{j=0}^{16} m_j + \sum_{i=0}^{16}\sum_{j=i+1}^{s}(C_j - v_{i,j})x_{i,j} + \sum_{j=0}^{3}(C_j - v_{-2,j})x_{-2,j}\right)$$

Restricciones:

Sabemos que sólo podemos vender una vez el tractor, es decir para $i \in \{-2, 0, 1, ..., 16\}$

$$\sum_{j} x_{i,j} \leq 1$$

Además, sólo podemos tener un tractor en cualquier momento,es decir para los tiempos $j \in \{0, ..., 16\}$:

$$\sum_{i} x_{i,j} \leq 1$$

Definimos las $x_{-2,j}$ como las variables que miden en que momento se tiene que vender el tractor inicial de antigüedad dos años. Sabemos que el tractor inicial se tiene que vender dentro de los primeros tres años, es decir:

$$\sum_{k=0}^{3} x_{-2,k} = 1$$

Si vendemos el tractor inicial en $j = 0$, tenemos que venderlo durante los siguientes cinco años. En cambio, si el tractor inicial no se vende en $j = 0$, todas la variables $x_{0,k}$ deberán ser cero. Por lo tanto tenemos la siguiente desigualdad

$$\sum_{k=1}^{5} x_{0,k} = x_{-2,0}$$

De igual forma, si el tractor inicial se vende en $j = 1$:

$$\sum_{k=2}^{6} x_{1,k} = x_{-2,1}$$

Ahora, cuando vendemos el tractor en el tiempo $j = 2$, el tractor que se puede vender tiene la opción de haberse comprado en $j = 0$ o seguir siendo el tractor inicial. Por lo que la restricción nos queda:

$$\sum_{k=3}^{7} x_{2,k} = x_{-2,2} + x_{0,2}$$

Si vendemos el tractor en $j = 3$ tenemos:

$$\sum_{j=4}^{8} x_{3,k} = x_{-2,3} + x_{0,3} + x_{1,3}$$

y si lo vendemos en $j = 4$ tenemos:

$$\sum_{k=5}^{9} x_{4,k} = x_{0,4} + x_{1,4} + x_{2,4}$$

A partir de $j > 4$ podemos generalizar las restricciones de la siguiente forma:

$$\sum_{k=j+1}^{s} xj, k = \sum_{k=2}^{5} x_{j-k,j}$$

Finalmente, sabemos que cuando se retire el señor Márquez el tractor se venderá sin importar si no se han cumplido los dos años, esto se traduce como:

$$\sum_{i=12}^{16} x_{i,17} = 1$$

1.2. Solución

La solución obtenida con la formulación anterior de Julieta es la siguiente: Vender y remplazar el traactor en los años 2, 7, 12 y 17.

La información anterior queda resumida en la siguiente tabla.

Año de remplazo	Antigüedad del tractor	Precio de venta del tractor	Costo de tractor nuevo
2	4	\$26,811.90	\$47,407.50
7	5	\$29,682.76	\$60,505.32
12	5	\$37,883.56	\$77,221.82
17	5	\$48,350.09	\$98,556.79

A continuación se presentan las tablas de los precios de venta a lo largo de los 17 años así como los costos de los tractores y los precios de mantenimiento.

Precios de venta en el año j del tractor comprado en el año i:

i,j	0	1	2	3
-2	\$31,000.00	\$28,830.00	\$26,811.90	\$24,935.07

Para $i = 0$ en adelante:

i,j	i+1	i+2	i+3	i+4	i+5
0	$35,991.00	$33,471.63	$31,128.62	$28,949.61	$26,923.14
1	$37,790.55	$35,145.21	$32,685.05	$30,397.09	$28,269.30
2	$39,680.08	$36,902.47	$34,319.30	$31,916.95	$29,682.76
3	$41,664.08	$38,747.60	$36,035.26	$33,512.80	$31,166.90
4	$43,747.29	$40,684.98	$37,837.03	$35,188.44	$32,725.24
5	$45,934.65	$42,719.22	$39,728.88	$36,947.86	$34,361.51
6	$48,231.38	$44,855.19	$41,715.32	$38,795.25	$36,079.58
7	$50,642.95	$47,097.94	$43,801.09	$40,735.01	$37,883.56
8	$53,175.10	$49,452.84	$45,991.14	$42,771.76	$39,777.74
9	$55,833.85	$51,925.48	$48,290.70	$44,910.35	$41,766.63
10	$58,625.55	$54,521.76	$50,705.24	$47,155.87	$43,854.96
11	$61,556.82	$57,247.85	$53,240.50	$49,513.66	$46,047.71
12	$64,634.67	$60,110.24	$55,902.52	$51,989.35	$48,350.09
13	$67,866.40	$63,115.75	$58,697.65	$54,588.81	-
14	$71,259.72	$66,271.54	$61,632.53	-	-
15	$74,822.70	$69,585.11	-	-	-
16	$78,563.84	-	-	-	-

Costo de un tractor por año:

año:	0	1	2	3	4	5	6
$	43,000.00	45,150.00	47,407.50	49,777.88	52,266.77	54,880.11	57,624.11

año:	7	8	9	10	11	12	13
$	60,505.32	63,530.58	66,707.11	70,042.47	73,544.59	77,221.82	81,082.91

año:	14	15	16
$	85,137.06	89,393.91	93,863.61

Costo de mantenimiento por año:

año:	0	1	2	3	4	5	6
$	1,300.00	1,495.00	1,719.25	1,977.14	2,273.71	2,614.76	3,006.98

año:	7	8	9	10	11	12	13
$	3,458.03	3,976.73	4,573.24	5,259.23	6,048.11	6,955.33	7,998.62

año:	14	15	16
$	9,198.42	10,578.18	12,164.91

Debido a que el costo de mantenimiento está fijo por año, el costo total de mantenimiento durante los 17 años viene dado por la suma de los costos de mantenimiento de cada año, ya que forzosamente tiene que haber un tractor por año.

1.3. El error de Julia

El hermano de Julia le comenta que cometió un error a la hora de estimar los gastos de mantenimiento y operación, ya que no consideró la edad del tractor. Tomando en cuenta la edad del tractor, el costo de mantenimiento será el siguiente:

- Se mantiene el aumento del 15 % anual.
- En los dos primeros años de vida del tractor su costo de mantenimiento tendrá un incremento del 1 % con respecto al precio de compra del tractor.
- Para los tractores de 3, 4 y 5 años de antigüedad su costo de mantenimiento aumentará en un 3 % con respecto al costo de mantenimiento del año anterior.

Por lo tanto, el costo de mantenimiento es m_{ij} donde la i representa el año en el que fue comprado el tractor, y la j seguirá siendo el año en el que se paga el mantenimiento. Vemos que $i \in \{0, 1, ..., 16\}$ y $j = i + 1, ..., s$, entonces tenemos que si $i \leq 12$:

$$m_{ij} = \begin{cases} \sum_{k=i}^{j-1}(1300(1.15)^k + 0.1C_i) & j = i+1, i+2 \\ \sum_{k=i}^{i+1}(1300(1.15)^k + 0.1C_i) + \sum_{k=i+2}^{j-1}(1300(1.15)^k + (0.1)(1.03)^{k-i-1}C_i) & j = i+3, i+4, i+5 \end{cases}$$

Cuando $i = -2$, el costo va a ser:

$$m_{ij} = \begin{cases} \sum_{k=0}^{j}(1300(1.15)^k + 0.1C_0) & j = 0, 1 \\ \sum_{k=0}^{1}(1300(1.15)^k + 0.1C_0) + \sum_{k=2}^{j}(1300(1.15)^k + (0.1)(1.03)^{k-1}C_0) & j = 2 \end{cases}$$

Este nuevo costo afecta a nuestra función objetivo, la cual cambia a:

$$\text{mín}\left(\sum_{i=0}^{16}\sum_{j=i+1}^{s}(C_j - v_{i,j} + m_{i,j})x_{i,j} + \sum_{j=0}^{3}(C_j - v_{-2,j} + m_{-2,j})x_{-2,j}\right)$$

Las restricciones de este problema coinciden con las del problema anterior. Es decir, seguimos en el mismo poliedro.

1.4. Martín continúa dirigiendo

A partir de la jubilación de su padre, Martín tiene pensado continuar dirigiendo la granja. Martín le pregunta a su hermana si hay una forma en que el cambio del tractor se realice de forma periódica en el tiempo.

Se busca que el tractor sea reemplazado cada dos, tres, cuatro o cinco años. Supongamos que a un tiempo determinado Martín compra un tractor al precio $\$p$ y costo de mantenimiento de carburantes y refracciones asociado a ese año igual a m y veamos que ocurre con el costo en los posibles años de remplazo.

Vemos que la política que conviene en promedio, es reemplazar el tractor cada cinco años dado que una vez que se sale de la agencia el tractor pierde 10 % de su valor. Así que si compramos y vendemos en un periodo menor a los 5 años, estamos perdiendo 10 % más el 7 % de la devaluación anual.

año:	Tractor Nuevo	Precio de venta	Costo mantenimiento
2	$p(1.05)^2$	$0.9p(.93)^2$	$m(1.15)^2 + p(0.01)$
3	$p(1.05)^3$	$0.9p(.93)^3$	$m(1.15)^3 + p(0.01)(1.03)^1$
4	$p(1.05)^4$	$0.9p(.93)^4$	$m(1.15)4 + p(0.01)(1.03)^2$
5	$p(1.05)^5$	$0.9p(.93)^5$	$m(1.15)^5 + p(0.01)(1.03)^3$

A. Problema 2

Cósmeticos Mac Bella produce en su línea de fabricación de pintalabios 10 diferentes tonos. La planeación se hace la semana anterior con base al pedido de la próxima semana. El jefe de la planta se ha dado cuenta que aun con pedidos idénticos los tiempos de entrega final de la producción varía dependiendo de cómo se haya organizado la producción. Esto se debe principalmente a que los tiempos de preparación para pasar de un color a otro cambian con base a los colores involucrados. Dichos tiempos se ven de esta forma:

Pinta Labios	Código	R1	R1	R6	RW	RR	SS	SP2	S12	T3	NG	T.C.
Rosa Pálido	R1	0	2	2	3	4	5	4	3	5	6	60
Rosa Pop	R2	2	0	1	3	3	4	6	4	3	5	31
Rosa Discreto	R6	2	1	0	4	5	3	2	4	1	7	45
Ruby Woo	RW	3	3	4	0	2	4	5	3	5	2	60
Rojo Ruso	RR	4	3	5	2	0	2	4	1	2	2	71
Saigon Summer	SS	5	4	3	4	2	0	3	2	1	3	82
Sheer Plumm	SP2	4	6	2	5	4	3	0	3	2	1	52
Sbon	S12	3	4	4	3	1	2	3	0	5	3	44
Tanarama	T3	5	3	1	5	2	1	2	5	0	4	74
Negro Gótico	NG	6	5	7	2	2	3	1	3	4	0	100

Se le contrató para analizar el problema y entregar:

A.1. Reporte técnico

Hay que darnos cuenta que se quiere optimizar el tiempo de producción. El tiempo total se da por la ecuación

$$TTP = TP + TC$$

Donde:

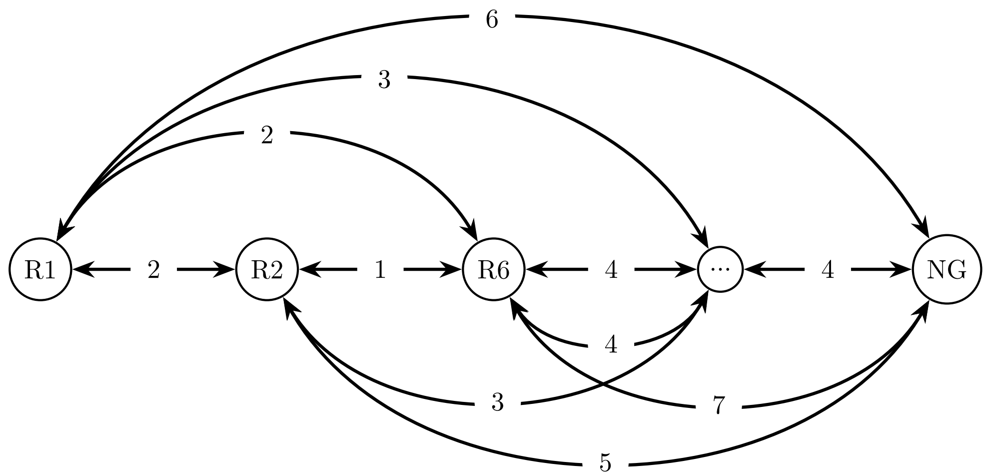
- TTP es el tiempo total de producción de la fábrica
- TP es el tiempo de producir los labiales una vez que están dentro de la linea de producción
- TC es el tiempo que tardan en preparar la linea de producción para comenzar la producción de un labial

Este proyecto solo se enfocará en el componente TC y dada la información que nos dieron, sabemos que:

- La demanda es variable semana a semana. La demanda puede incluir la producción de muchos labiales distintos o de solo un tipo de labial
- La maquina no tiene ninguna restricción en el orden en el que podemos empezar a producir los labiales, i.e., podemos escoger producir cualquier color de labial después de producir cualquier otro.

- El tiempo para preparar la maquina varia según la secuancia de los colores de los labiales que se utilizo

Una representación gráfica de las conexiones entre los colores de los labiales es:



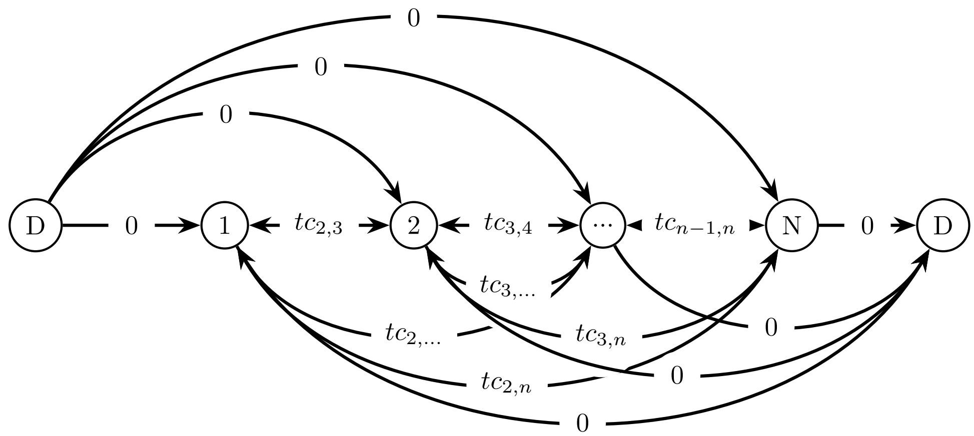
Notemos que no hace falta hacer que todos los nodos esten conectados entre si ya que basta con preparar la maquina solo una vez para producir el labial i. De esta forma podemos establecer una ecuación para analizar el TC. Proponemos:

$$TC = \sum_{j} \sum_{i} tc_{ij} b_{ij}$$

Donde:

- tc_{ij} es el tiempo que se utilza para pasar del labial i al labial j
- b_{ij} una variable binaria que se prende si se utiliza el camino del nodo i al nodo j

Es fácil notar que si elegimos un nodo inicial *dummy* y otro final para comenzar con un costo a cualquier otro nodo de 0, podemos hacer una equivalencia entre el problema del agente viajero y este problema. Adicional tenemos que notar que la demanda por los labiales es variable semanalmente por lo que existe el caso en que no tengamos que producir labiales de algún color. Esto implica, en la equivalencia del TSP, que no vamos a visitar las ciudades que no tienen demanda. De esta forma Un diagrama de esta equivalencia puede ser:



El objetivo del agente viajero es encontrar la manera de minimizar los costos de visitar distintas ciudades sin visitar ninguna de estas más de una vez y no puede faltar ninguna. En el problema de Mac Belle, tenemos el mismo objetivo. Si hacemos una equivalencia entre las ciudades y los labiales que tenemos que producir, queremos minimizar el tiempo de preparación de la linea de fabricaión, no queremos visitarlos más de una vez porque para producir la demanda de un labial nos basta con visitarlo una vez y tampoco preparar labiales para los que no nos pidieron productos.

Definamos L como el conjunto de todos los colores de los labiales y L_d el conjunto de los labiales demandados

A.2. Formulación del problema

Con esto ya podemos realizar el planteamiento del problema de programación lineal utilizando una formulación del agente viajero:

$$\begin{aligned}
\underset{X}{\text{Min}} \quad & \sum_i \sum_j tc_{ij} b_{ij} & \\
\text{S.A.} \quad & \sum_j b_{kj} = 1, \;\; k \in L_d & (i) \\
& \sum_j b_{ik} = 1, \;\; k \in L_d & (ii) \\
& \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset 2, \ldots, n, |S| \geq 2 & (iii) \\
& x_{ij} + x_{ji} \leq 1, i, j \in L_d, i \neq j & \\
& (i, j) \in L_d; b_{ij} \in 0, 1 &
\end{aligned}$$

Veamos que este es un planteamiento clásico del TSP por lo que este problema ha sido estudiado ampliamente. A continuación explicaremos cada restricción para la solución que le dimos a este problema:

(i) Esta restricción nos asegura que vamos a visitar una ciudad después de las ciudades que visitamos y solo vamos .

(ii) La segunda restricción nos asegura que tenemos que vamos a visitar todas las ciudades exactamente una vez.

(iii) La última restricción nos elimina tener la posibilidad de subciclos que no tienen al nodo 1. Implica que el número de arcos que pueden ser empacadas en el conjunto que de S nodos no puede ser mayor a $|S| - 1$ nodos.

Notemos que la complejidad del algoritmo es de orden de $O(2^n)$. Esto se debe a que cada nodo adicional nos agrega una cantidad importante de operaciones, por lo tanto con esta formulación del problema el tiempo de resolución va a ser un factor importante para la toma de decisiones.

A.3. Tiempo de resolución

El gerente de la planta quiere expandir el número de colores para producir por lo tanto nos pidio una opinión si es necesario comprar un programa dedicado a la resolución de problemas de programación lineal. Para poder dar una respuesta, es necesario saber: 1) el tiempo máximo que se puede tener para la resolución de este problema y 2) la potencia de la maquina, es decir, necesitamos saber cuantas operaciones puede procesar por segundo la computadora dedicada a resolver este problema. Supongamos que el número de operaciones que puede realizar la maquina son 16 operaciones por segundo, por lo que el tiempo para resolver el programa según el número de núcleos es de:

# Labiales	# de Operaciones Máximo	Tiempo Máximo en Minutos
2	4	0.00
3	8	0.01
4	16	0.02
5	32	0.03
6	64	0.07
7	128	0.13
8	256	0.27
9	512	0.53
10	1024	1.07
20	1,048,576	1,092.27
25	33,554,432	34,952.43
30	1,073,741,824	1,118,481.07

Por lo tanto el número de labiales máximo que puede tener con esta formulación es de 20 si el tiempo máximo que le puede dedicar a planear la producción es de 1 día y si es de 1 semana sería de 23 labiales.

B. Metaheurísticas

Las metaheurísticas, son estrategias inteligentes que sirven para mejorar procedimientos heurísticos muy generales, es decir, muchas veces las metaheurísticas se basan en heurísticas muy sencillas a las que se les aplica una estrategia adicional con el fin de no caer en óptimos locales y de esta manera encontrar una mejor solución al problema.

Las metaheurísticas de búsqueda establecen estrategias para recorres el espacio de soluciones del problema, transformando de forma iterativas soluciones iniciales. Las metaheurísticas pueden basarse en búsquedas monótonas, algoritmos escaladores o búsquedas locales.

B.1. Búsqueda Local

Las búsquedas locales se basan en el estudio de soluciones de vecindarios o entorno de la solución que realiza algún recorrido. La metaheurística establece como pauta elegir iterativamente la mejor de las soluciones mientras exista alguna mejora posible. Sin embargo, las búsquedas locales pueden quedarse atrapadas en un óptimo local. Para evitar quedarse en un óptimo local se puede solucionar este error mediante la aplicación de búsquedas globales que incorporan pautas para escapar de los óptimos locales. Estas pautas pueden ser: Volver a iniciar la búsqueda desde otra solución de arranque, modificar la estructura de entornos que se está aplicando y permitir movimientos o transformaciones de la solución de búsqueda que no sean de mejora mediante criterios de aceptación estocásticos o utilizando la memoria del proceso de búsqueda.

B.2. Búsqueda 2-opt

En optimización, 2-opt es un algoritmo de búsqueda local simple propuesto por primera vez por Croes en 1958 para resolver el problema del vendedor ambulante. La idea principal detrás de esto es tomar una ruta que cruce sobre sí misma y reordenarla para que no lo haga.

Una búsqueda local completa de 2 opciones (2-opt) comparará todas las posibles combinaciones válidas del mecanismo de intercambio. Esta técnica se puede aplicar al problema del vendedor ambulante, así como a muchos problemas relacionados.

El algoritmo parte a partir de una ruta ya definida y la intenta mejorar realizando cambios entre 4 nodos relacionado digamos que $A< - >B$ y $C< - >D$ los desconecta y luego verifica si se mejora la distancia con los cambios $A< - >C$ con $B< - >D$ ó $A< - >D$ con $B< - >C$ y si mediante alguno de estos cambios la distancia se mejora se aplica el cambio y se pone una bandera en estos nodos para ya no hacer otro cambio con estos nodos y se intenta este procedimiento con todos los nodos.

B.2.1. PseudoCódigo

Funcion 2opt(r) *r es una ruta que se obtiene con anterioridad*

Sea r = rutaexistente

Mientras no haya mejora

mejordistancia= dist(r)

Para i = 1 hasta num. de nodos a intercambiar - 1

Para k=i+1 hasta num. de nodos a intercambiar

nuevaruta = 2optCambio(r,i,k)

nuevadist = dist(nuevaruta)

Si nuevadist¡mejordistancia

r = nuevaruta

mejordistancia = nuevadistancia

Regresar r y mejordistancia

Funcion 2optCambio(ruta, i,k) *donde i y k son indices de los nodo de la ruta r*

nuevaruta = null (rutavacia);

Tomar ruta de 0 a i-1 de la ruta r y agregar trayecto a nuevaruta

Tomar ruta de i a k y agregar en forma inversa a la nuevaruta

Tomar ruta de k a fin de la ruta r y agregar trayecto a nuevaruta

Regresar nuevaruta

B.3. Búsqueda Tabú

La búsqueda tabú es una metaheurística que usa un procedimiento de tipo iterativo para resolver problemas de optimización combinatoria discretos.

El principal objetivo de la búsqueda tabú es escapar de óptimos locales, para ello emplea algunas metodologías como el uso de memorias flexibles o cambio de la estructura del problema creando o relajando las restricciones con la finalidad de realizar la búsqueda en áreas no factibles pero que de alguna manera nos puede llevar a una solución de buena calidad.

La búsqueda tabú se basa en tres principios:

1. Uso de estructuras de memoria basadas en atributos diseñados para permitir criterios de evaluación e información de búsqueda histórica, la cual se explota más a fondo que las estructuras de memoria rígida o por sistemas de pérdida de memoria (como recocido simulado)

2. Un mecanismo asociado de control, mediante el empleo de estructuras de memoria, basado en la modificación de las condiciones que restringen y liberan al proceso de búsqueda

3. La incorporación de funciones de memoria de diferentes lapsos de tiempo desde corto plazo hasta largo plazo.

El tipo de memorias que se usan en la búsqueda tabú se clasifica en:

- Memoria a corto plazo: También llamada lista tabú, en esta memoria se almacena el historial de los últimos movimientos realizados a través de una lista FIFO, prohibiendo que los movimientos se repitan en las próximas iteraciones, así evita ciclos y escapada de mínimos locales.

- Memoria a medio plazo: En esta se registran los atributos más comunes de un conjunto de soluciones.

- Memoria a largo plazo: La memoria guarda un registro de las zonas que no han sido exploradas aún, diversificando de esta manera la búsqueda.

La estrategia de diversificación trata de llevar la búsqueda a zonas del espacio de soluciones no visitadas antes, generando de esta manera soluciones que difieren significativamente de las ya encontradas. Además de emplear estas estrategias para la búsqueda del óptimo global emplea un método llamado nivel de aspiración. El objetivo de aplicar este criterio es determinar en que condiciones se puede admitir un movimiento clasificado como tabú, cambiando su clasificación de tabú cuando su memoria a corto plazo expire.

B.4. Algoritmos Genéticos

Los algoritmos genéticos son un tipo de metaheurísticas que se inspiran en el proceso de evolución que tienen los organismos a lo largo de las generaciones. Biologicamente, el proceso de evolución, es una combinación de los cromosomas al momento de la concepción de un organismo y de mutaciones de estos genes. Los organismos que sobreviven y se van reproduciendo cada generación son los más aptos para la forma de vida. Estos organismos aptos se van reproduciendo entre si para ir creando organismos cada vez más aptos a través de una combinación de sus cromosomas. Es muy raro, pero suele suceder, que algún cromosoma haga una mutación sin tener que ver con los cromosomas de sus antecesores.

Este es el comportamiento que se trata de replicar en los algoritmos geneticos. Siguiendo con la misma idea, los elementos de un algoritmo genetico son:

- Una función objetivo, que en los algoritmos geneticos se le llama una función de adaptabilidad. La función de adaptabilidad es la función que nos dice que tan bueno es cada una de las combinaciones de los cromosomas para saber si es un organismo apto o si no lo es
- Una población de cromosomas, que es una muestra de posibles resultados del problema de programación lineal. Comienza siendo una muestra aleatoria de las posibles solucionas del problema y se representan a través de una combinación númerica. Esta combinación usualmente es de 1's y 0's y se hace un vector de caracteres pegados, aunque tambien se utilizan otras formas de vector incluyendo que sus elementos sean números reales
- A partir de la población de cromosomas, necesitamos una forma de elegir cuales son los cromosomas más optimos para reproducirse. Usualmente se define una función de probabilidad dada la adaptanilidad del organismo y después se elige con remplazo una cantidad "X"de ellos. A estp se le conoce como *crossover*
- Una técnica de reproducción entre estos cromosomas aceptados y
- Una definición para la mutación de los cromosomas de recien progenitores

Con estos elementos y un número máximo de iteraciones se define el algoritmo genético. El procedimiento estandar del algoritmo es escoger aleatoriamente la población de cromosomas inicial y correr el procedimiento un número máximo de veces esperando que el algoritmo converga a la solución optima o una solución muy cercana a la optima. Adicional se corre el algoritmo completo varias veces ya que la población inicial pudo haber sesgado el experimento. Al final de esto se reporta el cromosoma con la mejor adaptabilidad.

B.4.1. Planteamiento del TSP

Definamos los elementos del algoritmo genético que utilizaremos para el TSP. Sea n el número de ciudades a visitar y la secuencia $d_1, d_2, \ldots, d_k, \ldots d_n$ es la distancia entre las ciudades de la ruta elegida

- **Función de Adaptabilidad:** $\sum_{k=1}^{n} d_{k,k+1}$
- **Población de Cromosomas:** Cada cromosoma tendrá n elementos en el vector y nuestra población siempre será de 20 cromosomas
- **Creación de la siguinente generación:** Nos quedaremos con la mitad de los cromosomas progenitores que sean más aptos y la otra mitad será una combinación de los progenitores en pareja.
- **Crossover:** Las parejas se elegiran aleatoriamente usando una función de probabilidad que le de una mayor probabilidad de ser elegido al más apto y una probabilidad menor al menos apto.
  Para formar los nuevos cromosomas utilizaremos un elemento aleatoriamente del primer progenitor y a partir de ahí intercambiaremos el resto del vector. Es muy probable que ambos vectores se queden con elementos duplicados por lo que aletaoriamente empezaremos a elegir un elemento del vector para intercambiarlo hasta lograr que no haya duplicados en cada uno de los dos vectores.
- **Mutación:** La mutación de un cromosoma será de 1 % de probabilidad y si muta, se elegiran dos ciudades aleatoriamente y se permutarán

B.4.2. Pseudocódigo del TSP

Algorithm 1 Algoritmos Geneticos Adaptado al TSP

Require: $n > 0$ Número de ciudades a visitar
$maxit > 0$ Número de iteraciones máximo
$A \in R^{n \times n}$ Una matriz de distancias entre ciudades
$Pob > 0$ Número de habitantes en la población en todo momento
$0 \leq mut \leq 1$ La probablidad de mutación de un elemento de la población
$0 \leq select \leq 1$ Porcentaje que avanza a la siguiente generación
$function$ La función a optimizar
Ensure: POB Se crea aleatoriamente la población inicial
Dist ← function(POB) Obtienes la distancia de cada cromosoma de la población
sort(POB, Dist) Ordenas los cromosomas con menor distancia a mayor
Hdist[1] ← min(Dist) El mejor resultado de la población actual
MinDist[1] ← POB[1] El recorrido de la mejor distancia
probas ← La probabilidad de elegir los cromosoma para reproducirse
Nprobas ← length(probas)
for $i = 1$ to maxit **do**
(i) Obtienes las parejas de progenitores aleatoriamente
(ii) Combinas a los progenitores con la regla de combinación
(iii) Creas el vector de población incluyendo los 10 más aptos y los progenitores
(iv) Haces la mutación del vector de población nuevo
Dist ← function(POB) Obtienes la distancia de cada cromosoma de la población
sort(POB, Dist) Ordenas la población con menor distancia a mayor
Hdist[i+1] ← min(Dist) El mejor resultado de la población actual
MinDist[i+1] ← POB[1] El recorrido de la mejor distancia
end for